Succinct Functional Commitments for Circuits from k-Lin

Hoeteck Wee
NTT Research and ENS, Paris
wee@di.ens.fr

David J. Wu*
UT Austin
dwu4@cs.utexas.edu

Abstract

A functional commitment allows a user to commit to an input $\mathbf{x}$ and later, open the commitment to an arbitrary function $\mathbf{y} = f(\mathbf{x})$. The size of the commitment and the opening should be sublinear in $|\mathbf{x}|$ and $|f|$.

In this work, we give the first pairing-based functional commitment for arbitrary circuits where the size of the commitment *and* the size of the opening consist of a *constant* number of group elements. Security relies on the standard bilateral k-Lin assumption. This is the first scheme with this level of succinctness from falsifiable bilinear map assumptions (previous approaches required SNARKs for NP). This is also the first functional commitment scheme for general circuits with $\mathsf{poly}(\lambda)$-size commitments and openings from *any* assumption that makes fully black-box use of cryptographic primitives and algorithms. As an immediate consequence, we also obtain a succinct non-interactive argument for arithmetic circuits (i.e., a SNARG for P/poly) with a *universal* setup and where the proofs consist of a constant number of group elements. In particular, the CRS in our SNARG only depends on the size of the arithmetic circuit $|C|$ rather than the circuit C itself; the same CRS can be used to verify computations with respect to different circuits. Our construction relies on a new notion of projective chainable commitments which may be of independent interest.

1 Introduction

A functional commitment scheme [IKO07, BC12, LRY16] allows a user to commit to an input $\mathbf{x}$ and later on, open the commitment to an arbitrary function f evaluated on the committed value (i.e., open to the value $f(\mathbf{x})$). Moreover, we require that both the size of the commitment *and* the size of the opening be short; they should be sublinear in the size of the input $\mathbf{x}$ and the description length of f. The security requirement is *evaluation binding*, which states that given a commitment σ, an efficient adversary should not be able to open σ to two different values $\mathbf{y} \neq \mathbf{y}'$ with respect to the same function f.

Functional commitments generalize notions like vector commitments [CFM08, LY10, CF13, LM19, GRWZ20] and polynomial commitments [KZG10, PST13, LRY16, Lee21], and have found numerous applications to verifiable outsourcing of storage [BGV11], authenticated data structures [PSTY13], and new constructions of homomorphic signatures and verifiable databases [CFT22]. As a primitive, functional commitments can be viewed as a particular case of succinct non-interactive arguments (SNARGs) for "commit-and-prove" languages, albeit satisfying a *weaker* security notion of evaluation binding rather than soundness. In many cases, functional commitments are a building block in many constructions of succinct arguments [MBKM19, GWC19, CHM+20, BDFG21, BFS20, COS20, Lee21, ACL+22, CLM23] (where the stronger security requirement of soundness is obtained by relying either on the random oracle model or making a stronger knowledge assumption on the underlying commitment scheme).

Recently, there has been significant progress on constructing functional commitments that can support *arbitrary* circuits from both pairing-based [BCFL23, KLVW23] and lattice-based assumptions [dCP23, WW23b, KLVW23, BCFL23, WW23a]. With the exception of the RAM delegation scheme of [KLVW23], the size of the commitments or the openings (or both) in the other constructions scale with the depth of the circuit. The RAM delegation scheme of [KLVW23] gives a functional commitment where the size of the commitments and openings scale polylogarithmically with the length of the input and the size of the circuit, but relies on extensive non-black-use of cryptography.

*Part of this work was done while visiting NTT Research.

Scheme	Functions	$|\text{crs}|$	$|\sigma|$	$|\pi|$	BB	Assumption
[LRY16, Gro16]	arithmetic circuits	$O(s)$	$O(1)$	$O(1)$	✗	generic group
[LRY16]	linear functions	$O(\ell)$	$O(1)$	$O(m)$	✓	subgroup decision
[LM19]	linear functions	$O(\ell m)$	$O(1)$	$O(1)$	✓	generic group
[LP20]	sparse polynomials	$O(\mu)^*$	$O(m)$	$O(1)$	✓	uber assumption
[CFT22]	degree-d polynomials	$O(\ell^d m)$	$O(d)$	$O(d)$	✓	ℓ^d-DHE
[BCFL23]†	arithmetic circuits	$O(s^5)$	$O(1)$	$O(d)$	✓	ℓ-HiKer
[KLVW23]§	arithmetic circuits	$\text{poly}(\lambda)$	$O(1)$	$\text{poly}(\lambda)$	✗	k-Lin
This work	arithmetic circuits	$O(s^5)$	$O(1)$	$O(1)$	✓	bilateral k-Lin

*The parameter μ is a sparsity parameter for the polynomials (c.f., [LP20]).
†The authors of [BCFL23] also give a scheme that supports bounded-width arithmetic circuits where the CRS contains $O(w^5)$ group elements and the openings contain $O(d^2)$ group elements. Our techniques also yield a construction with these parameters (and from the standard k-Lin assumption as opposed to the non-standard q-type assumption); see Remark 5.18.
§ While [KLVW23] construct delegation for RAM programs, their construction can be adapted to obtain a functional commitments for general Boolean and arithmetic circuits. We consider the instantiation of their scheme with pairing-based batch arguments [WW22].

Table 1: Summary of pairing-based non-interactive functional commitments. For each scheme, we report the class of functions they support, the number of *group elements* in the common reference string crs, the commitment σ, and the opening π as a function of the input length ℓ and the output length m. For the constructions that support arithmetic circuits, we write s to denote the size of the circuit and d to denote the depth. We say that a scheme is "black-box" (**BB**) if it only makes black-box use of the group and any cryptographic primitives.

This work. In this work, we study functional commitments for general arithmetic circuits from pairings. Our goal in this work is to minimize the size of the commitments and the openings in a functional commitment scheme. Towards that end, we construct the first pairing-based functional commitment scheme that supports arbitrary circuits where the commitment and the openings consist of a *constant* number of group elements, irrespective of the input length or the circuit size. The security of our construction relies on the standard bilateral k-Lin assumption[1] for any constant $k > 1$. We summarize our main theorem below:

Theorem 1.1 (Informal)**.** *Let $k > 1$ be a constant. Assuming the bilateral k-*Lin *assumption over a pairing group of prime order p, there exists a (non-interactive) functional commitment scheme for arithmetic circuits (over $\mathbb{Z}_p$) of a priori bounded size with the following features:*

- *The commitment consists of $2k$ group elements.*
- *The opening consists of $O(k^2)$ group elements. (For $k = 2$, the number is 54).*
- *The scheme requires a structured common reference string (CRS) with $O(k^3 s^5)$ group elements, where s is the size of the circuit.*
- *If the circuit C in the opening is known in advance, then we can preprocess it into a short verification key. Then, the online verification of the commitment only requires computing $O(m)$ bilinear map operations, where m is the output length of the circuit C. We refer to Remark 5.16 for more details.*

We provide a comparison with other pairing-based constructions in Table 1. Notably, Theorem 1.1 is first functional commitment scheme for circuits with the following efficiency features:

- The first scheme based on falsifiable bilinear map assumptions (e.g., bilateral k-Lin or q-type assumptions) where the commitment and the opening consists of a *constant* number of group elements. The only previous

[1]The bilateral k-Lin assumption is a variant of k-Lin where the challenge is encoded in both $\mathbb{G}_1$ and $\mathbb{G}_2$.

constructions that support constant-size openings rely on the generic group model or on knowledge assumptions (due to the use of pairing-based SNARKs for NP).

- The first functional commitment scheme that makes fully *black-box* use of cryptographic primitives and algorithms where the size of the commitment and the opening is $\mathsf{poly}(\lambda)$ bits, regardless of the underlying assumptions. The recent lattice-based and pairing-based schemes in [dCP23, WW23b, BCFL23, WW23a] are also black-box, but the size of the opening all scale with the depth of the circuit. Even for the special case of constant-degree polynomials, our result improves upon the state of the art in [BCFL23] in that we rely on k-Lin instead of q-type assumptions. Constructions based on generic approaches (SNARKs or non-interactive batch arguments) do achieve $\mathsf{poly}(\lambda)$ size, but requires non-black-box access to the underlying primitives and algorithms. We provide more discussion on this below.

Moreover, our functional commitment scheme is additively-homomorphic, so using the results from [CFT22], we obtain homomorphic signatures for all (bounded-size) arithmetic circuits from the bilateral k-Lin assumption. This is the first homomorphic signature scheme for general circuits based on falsifiable pairing-based assumptions where the signature consists of a *constant* number of group elements. The number of group elements in previous pairing-based constructions either grow with the depth of the circuit [BCFL23] or require a $\mathsf{poly}(\lambda)$ number of group elements due to non-black-box use of cryptography [KLVW23].

SNARG for $\mathsf{P/poly}$ with universal setup. Our functional commitment scheme immediately gives a succinct non-interactive argument (SNARG) for $\mathsf{P/poly}$ with a universal setup. In this setting, the prover has an input $\mathbf{x} \in \mathbb{Z}_p^\ell$, and seeks to convince the verifier that $\mathbf{y} = C(\mathbf{x})$, where C is an arithmetic circuit. Moreover, the length of the proof should be much shorter than the size of the arithmetic circuit $|C|$ as well as the input length $|\mathbf{x}|$ and output length $|\mathbf{y}|$. In a SNARG with universal setup [GKM+18], the common reference string should only depend on a bound on the size of the circuit $|C|$ rather than the circuit C itself. Moreover, there is then an algorithm that takes as input the CRS and the circuit C and outputs a succinct verification key vk_C for C. Given the preprocessed verification key vk_C, checking a proof that $\mathbf{y} = C(\mathbf{x})$ should require time that is sublinear in the size of $|C|$.

A functional commitment scheme for arithmetic circuits directly implies a SNARG for $\mathsf{P/poly}$. The proof is a commitment σ to $\mathbf{x}$ together with an opening of σ to $\mathbf{y}$ with respect to the circuit C. The SNARG verifier can check that the commitment σ was *honestly* computed (since it knows the input $\mathbf{x}$). Soundness now follows from evaluation binding of the functional commitment scheme. If the functional commitment scheme supports fast verification, then the resulting SNARG has a universal setup algorithm, where the same CRS can be used to check different computations. Thus, Theorem 1.1 gives a SNARG for $\mathsf{P/poly}$ from bilateral k-Lin with a universal setup and where the proof consists of a constant number of group elements. Previously, the work of [GZ21] showed how to construct a SNARG for $\mathsf{P/poly}$ from the bilateral k-Lin assumption where the proof consists of a constant number of group elements. The construction in [GZ21] relies on a *circuit-dependent* CRS where the circuit C is embedded into the CRS. It is possible to use universal circuits and have the description of C be part of the statement itself; the question then is whether the resulting construction supports fast verification (given a precomputed verification key vk_C). Recent RAM delegation schemes (i.e., SNARGs for P) [CJJ21, KVZ21, KLVW23] also imply a SNARG for $\mathsf{P/poly}$ with universal setup by treating the description of the circuit C as part of the initial contents of the memory of the RAM program. Due to the non-black-box use of cryptography, the proofs in these constructions (when instantiated over groups with bilinear maps) contain a super-constant number of group elements.

Comparison to generic approaches. Generic approaches based on SNARKs [LRY16] and non-interactive batch arguments (BARGs) [KLVW23] provide an alternative route for constructing functional commitments for general circuits. Here, we discuss some limitations of these approaches beyond their non-black-box use of cryptography:

- The SNARK-based approach [LRY16] instantiated using a pairing-based SNARKs for NP with constant-size proofs (e.g., [Gro10, Lip12, GGPR13, BCI+13, DFGK14, Gro16]) yields a functional commitment where the commitment and openings contain $O(1)$ group elements. However, the reliance on SNARKs for NP brings in strong, non-falsifiable assumptions or requires working in the generic bilinear map model to argue security. Moreover, constructing SNARKs for NP from simple falsifiable assumptions over bilinear maps is likely to be difficult [GW11]. The functional commitments we build in this work rely solely on the falsifiable (bilateral) k-Lin assumption.

- The authors of [CJJ21, KLVW23] shows how to use non-interactive batch arguments (BARGs) for NP to obtain a RAM delegation scheme. In particular, the approach from [KLVW23] can be adapted to obtain a functional commitment for general circuits; we refer to [WW23a, §1.3] for a sketch of the adaptation. Combined with the pairing-based BARG from [WW22], this yields a functional commitments for all circuits from the standard k-Lin assumption.[2] While the commitments in the resulting construction consist of a constant number of group elements, the opening are longer. Specifically, the opening consists of a BARG proof. When the BARG is instantiated with [WW22], the size of the BARG proof scales linearly with the size of the verification circuit for the underlying NP relation. In [KLVW23], this NP relation includes the verification algorithm of a somewhere extractable hash function. This is a cryptographic primitive, so the size of this circuit scales polynomially with the security parameter. Correspondingly, the size of the opening contains $\mathsf{poly}(\lambda)$ group elements. It is unclear how to adapt this approach to obtain a functional commitment where the opening consists of a *constant* number of group elements. In this case, the non-black-box use of cryptography translates to an asymptotic loss in succinctness.

On the flip side, these non-black-box approaches have the advantage that they require a short CRS. Notably, the BARG-based approach of [KLVW23] only requires a CRS that grows polylogarithmically with the circuit size. Their scheme thus supports circuits of unbounded size, but do not have constant-size openings.

Open problems. An interesting question is to construct functional commitments from k-Lin (or q-type assumptions) with constant-size commitments and openings (measured in terms of the number of group elements) with a shorter CRS (e.g., a quadratic-size CRS or linear-size CRS). The CRS size in our current construction scales with $O(s^5)$. Existing approaches that have constant-size commitment and openings all rely on pairing-based SNARKs, which requires strong non-falsifiable assumptions. We note that in this setting, there has been a long and successful line of work focused on constructing and optimizing pairing-based SNARGs with constant-size proofs [Gro10, Lip12, GGPR13, BCI+13, DFGK14, Gro16]. Similarly, in the related setting of batch arguments for NP, recursive composition has proven useful for reducing the size of the CRS [KPY19, CJJ21, WW22, KLVW23]. It is an interesting to see if similar techniques are applicable to obtain functional commitments with a shorter CRS (while retaining commitments and openings that are only a constant number of group elements).

2 Technical Overview

The starting point of our construction is a new *chainable functional commitment* scheme for quadratic functions from the k-Lin assumption. In a chainable functional commitment [BCFL23], the user can commit to an input $\mathbf{x} \in \mathbb{Z}_p^\ell$ (with commitment $\sigma_\mathbf{x}$) and then compute an opening π to a new *commitment* $\sigma_\mathbf{y}$ of the output vector $\mathbf{y} = f(\mathbf{x})$ where $f \colon \mathbb{Z}_p^\ell \to \mathbb{Z}_p^\ell$ is a vector-valued function. The key difference between chainable functional commitments and standard functional commitments is that the user opens to a succinct commitment of the output rather than the (possibly long) output itself. The security requirement is evaluation binding, which says that an efficient adversary should not be able to open the commitment $\sigma_\mathbf{x}$ to two different output commitments $\sigma_\mathbf{y}, \sigma'_\mathbf{y}$. The authors of [BCFL23] show that a chainable commitment scheme directly implies a functional commitment scheme for arithmetic circuits. Here, we describe their approach for the simpler setting of *layered* arithmetic circuits:

- The commitment itself is a commitment σ_1 to the input.
- To construct an opening to a (layered) arithmetic circuit C where the value of layer i is a quadratic function of the values in layer $i-1$, the user first commits to the wires at each layer. If there are d layers, then the user constructs d commitments $\sigma_2, \ldots, \sigma_d$ (note that the original commitment σ_1 corresponds to the inputs). Finally, the user provides a chaining proof $\pi_{i,i+1}$ that each pair (σ_i, σ_{i+1}) is correctly computed (with respect to the quadratic function that implements the mapping from the layer-i wires to the layer-$(i+1)$ wires). This step is implemented using a chainable commitment for quadratic functions.

[2] This construction can also be instantiated in pairing-free groups by relying on the (subexponential) DDH assumption [CGJ+23].

The above construction provides a general blueprint for constructing functional commitments for layered arithmetic circuits where the size of the opening grows with the depth of the circuit. The authors of [BCFL23] then describe how to construct chainable functional commitments for quadratic functions using a non-standard q-type assumption on bilinear maps (the ℓ-HiKER assumption, where ℓ denotes the input length). We note that a similar approach was also used for constructing succinct arguments in [GR19].

Overview of our approach. Our goal is to implement the [BCFL23] approach, but with only a constant number of group elements in the opening. A natural approach is to commit to *all* of the wires in the circuit *twice*: once as an input commitment σ_1 and once as an output commitment σ_2. Suppose we number the wires in topological order. Then, to argue evaluation binding, we could try to argue that the first $i + 1$ wires committed in σ_2 are consistent with the first i wires committed in σ_1. The problem with this strategy is the evaluation binding property for a chainable commitment only allows us to reason *globally* about the input and output commitments, whereas this "wire-by-wire" consistency property pertains to reasoning about prefixes of the committed vectors (i.e., analyzing relationships between the first i components of the input vector and the first $i + 1$ components of the output vector). In this work, we introduce the notion of a "projective chainable commitment" that allows us to reason about properties on prefixes of the committed vectors. Our overall construction then has the following high-level structure:

- The commitment is a commitment σ_{in} to the input $\mathbf{x}$.
- The opening for a circuit $C\colon \mathbb{Z}_p^\ell \to \mathbb{Z}_p^m$ contains 3 commitments: σ_1, σ_2 are commitments to all s wire values (where s is the number of wires in C), and σ_{out} is a commitment to the m output wires.

In addition, the opening contain "proofs" that enforce the following prefix-based constraints:

- **Input consistency:** The first ℓ committed values in σ_1 are equal to the committed values in the input commitment σ_{in}.
- **Gate consistency:** For all $j = \ell + 1, \ldots, s$, the first $j + 1$ committed values in σ_2 are consistent with the first j committed values in σ_1 as determined by the circuit's "next wire" function (i.e., the function corresponding to the gate computing wire j). The "next wire" function can be described by a quadratic function.
- **Internal consistency:** For all $j = \ell+1, \ldots, s$, the first j committed values in σ_1 are equal to the first j committed values in σ_2.
- **Output consistency:** The last m committed values in σ_1 are equal to the committed values in σ_{out}

If all of these constraints are satisfied, then a straightforward iterative argument suffices to show evaluation binding (several recent constructions of delegation follow this type of approach [GZ21, CJJ21, KLVW23]). To formalize this approach, we need to first define what we mean when we say the "first j committed values in a commitment σ." We formalize this by defining a trapdoor setup algorithm that takes as input an index j and generates the public parameters together with a trapdoor $\mathsf{td}^{(j)}$. Then, given a commitment σ, we can use the trapdoor to extract from σ a commitment to the first j committed values in σ; we denote this latter commitment by $\mathsf{Project}(\mathsf{td}^{(j)}, \sigma)$. In particular, we can now restate the gate consistency and internal consistency constraints as follows:

- **Gate consistency:** For all $j = \ell + 1, \ldots, s$, the output of $\mathsf{Project}(\mathsf{td}^{(j+1)}, \sigma_2)$ is consistent with $\mathsf{Project}(\mathsf{td}^{(j)}, \sigma_1)$ with respect to the circuit "next-wire" function.
- **Internal consistency:** For all $j = \ell+1, \ldots, s$, the output of $\mathsf{Project}(\mathsf{td}^{(j)}, \sigma_1)$ is consistent with $\mathsf{Project}(\mathsf{td}^{(j)}, \sigma_2)$ with respect to the identity map.

Here, the "consistency requirement" corresponds to a chain-binding security property. In the actual construction, the commitments σ_1 and σ_2 will have different "types" and a different projection trapdoor will be used to project σ_1 and σ_2. The added flexibility will allow us to carry out the full proof of evaluation binding (see Sections 2.3 and 5) We refer to chainable commitments with this projective property as "projective chainable commitments."

2.1 Chainable Commitments for Quadratic Functions from Bilateral k-Lin

The starting point of our construction is a new construction of chainable commitments for quadratic functions. To simplify the description in the overview, we start by describing a "designated-verifier" variant of the construction, where a secret key is needed to check the opening. The secret-key version is simpler to describe, and readily extends to the setting of public verifiability using the techniques of Kiltz and Wee [KW15]. In the technical sections (Section 4), we only describe the version with public verification.

Notation. Throughout this work, we will use the implicit notation of group elements introduced in [EHK⁺13]. Our construction operates over a prime-order pairing group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ of order p with an efficiently-computable non-degenerate pairing $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. We let g_1 denote a generator for $\mathbb{G}_1$ and analogously for g_2 and g_T. For a matrix $\mathbf{M} \in \mathbb{Z}_p^{n \times m}$, we write $[\mathbf{M}]_1 \in \mathbb{G}_1^{n \times m}$ to denote the matrix of group elements $g_1^{\mathbf{M}}$ (when exponentiation is defined component-wise). Similarly, we write $[\mathbf{M}]_2$ to denote $g_2^{\mathbf{M}}$ and $[\mathbf{M}]_T$ to denote $g_T^{\mathbf{M}}$. For matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ over $\mathbb{Z}_p$ with compatible dimensions, we write $\mathbf{A}[\mathbf{B}]_1 + \mathbf{C}[\mathbf{D}]_1 \coloneqq [\mathbf{AB} + \mathbf{CD}]_1$, which can be computed using the group operation over $\mathbb{G}_1$. We define linear operations over $\mathbb{G}_2$ and $\mathbb{G}_T$ analogously. For two scalars $a, b \in \mathbb{Z}_p$, the pairing satisfies $e([a]_1, [b]_2) \coloneqq [ab]_T$. We extend this to matrix and tensor products[3] by writing $[\mathbf{A}]_1[\mathbf{B}]_2 \coloneqq [\mathbf{AB}]_T$ and $[\mathbf{A}]_1 \otimes [\mathbf{B}]_2 \coloneqq [\mathbf{A} \otimes \mathbf{B}]_T$. In more detail, the individual components of the matrix and tensor products are computed by applying the pairing to the corresponding elements of $\mathbf{A}$ and $\mathbf{B}$ and then, in the case of matrix multiplication, applying the group operation over $\mathbb{G}_T$. Finally, in the following description, we write $\mathbf{I}_d$ to denote the d-by-d identity matrix.

Warm-up: a scheme for *fixed* linear functions. We first describe a functional commitment that supports a single *fixed* linear function $\mathbf{x} \mapsto \mathbf{Mx}$. In this scheme, a user can commit to an input $\mathbf{x}$ and open to $\mathbf{y} = \mathbf{Mx}$. The construction is an adaptation of the Kiltz-Wee proof system [KW15] for proving membership in linear spaces:

- The public parameters contain two encoding matrices $[\mathbf{T}]_2, [\hat{\mathbf{T}}]_2 \in \mathbb{G}_2^{k \times \ell}$, where k is a constant (the parameter in the k-Lin assumption) and ℓ is the input length. We sample $\mathbf{T}, \hat{\mathbf{T}} \xleftarrow{\text{R}} \mathbb{Z}_p^{k \times \ell}$.
- A commitment to $\mathbf{x} \in \mathbb{Z}_p^\ell$ with respect to $[\mathbf{T}]_2$ is $[\mathbf{Tx}]_2$. We define commitments with respect to $\hat{\mathbf{T}}$ analogously.
- In the overview (and the rest of this paper), we refer to commitments with respect to $\hat{\mathbf{T}}$ as "Type-I commitments" and those with respect to $\mathbf{T}$ as "Type-II commitments." Our goal is to prove relationships between Type-I and Type-II commitments. For the setting of linear functions, the input commitment $[\mathbf{c}]_2$ might be a Type-II commitment to $\mathbf{x}$ and the goal is to construct an opening to a Type-I commitment $[\hat{\mathbf{c}}]_2$ of the vector $\mathbf{y} = \mathbf{Mx}$. We will also consider relations where the input is a Type-I commitment and the output is a Type-II commitment.

We now describe how to support linear openings for Type-II commitments. Specifically, starting from a Type-II commitment $[\mathbf{Tx}]_2$ of $\mathbf{x}$, we want to construct an opening to the Type-I commitment $[\hat{\mathbf{T}}\mathbf{Mx}]_2$ of the vector $\mathbf{Mx}$. To do so, we sample two vectors $\mathbf{r}, \mathbf{w} \xleftarrow{\text{R}} \mathbb{Z}_p^k$ and publish $[\mathbf{z}]_2$ in the public parameters where

$$\mathbf{z}^{\mathsf{T}} = \mathbf{w}^{\mathsf{T}}\mathbf{T} - \mathbf{r}^{\mathsf{T}}\hat{\mathbf{T}}\mathbf{M} \in \mathbb{Z}_p^\ell.$$

For now, we consider the *designated-verifier* setting where a secret key is needed to verify the openings. In this case, the vectors $(\mathbf{r}, \mathbf{w})$ are the secret verification key. Observe now that

$$\mathbf{z}^{\mathsf{T}}\mathbf{x} = \mathbf{w}^{\mathsf{T}}\mathbf{Tx} - \mathbf{r}^{\mathsf{T}}\hat{\mathbf{T}}\mathbf{Mx}.$$

We define the opening to be $v = \mathbf{z}^{\mathsf{T}}\mathbf{x}$. Then, the verification relation takes the Type-II commitment $[\mathbf{c}]_2 = [\mathbf{Tx}]_2$, the Type-I commitment $[\hat{\mathbf{c}}]_2 = [\hat{\mathbf{T}}\mathbf{Mx}]_2$ and checks that

$$[v]_2 \stackrel{?}{=} \mathbf{w}^{\mathsf{T}}[\mathbf{c}]_2 - \mathbf{r}^{\mathsf{T}}[\hat{\mathbf{c}}]_2.$$

[3] We recall some basic properties of the tensor product in Section 3.

Security of the basic construction. The security requirement says that it should be computationally difficult to construct a Type-II commitment $[\mathbf{c}]_2$ and a pair of distinct Type-I commitments $[\hat{\mathbf{c}}]_2 \neq [\hat{\mathbf{c}}']_2$ along with accepting openings $[v]_2, [v']_2$. In other words, it should be difficult for the adversary to output $[\mathbf{c}]_2$, $[\hat{\mathbf{c}}]_2$, $[\hat{\mathbf{c}}']_2$, $[v]_2$, and $[v']_2$ such that

$$\mathbf{r}^{\mathsf{T}}\hat{\mathbf{c}} = \mathbf{w}^{\mathsf{T}}\mathbf{c} - v \quad \text{and} \quad \mathbf{r}^{\mathsf{T}}\hat{\mathbf{c}}' = \mathbf{w}^{\mathsf{T}}\mathbf{c} - v'.$$

Equivalently, the adversary must be able to come up with $\hat{\mathbf{c}}^* = \hat{\mathbf{c}} - \hat{\mathbf{c}}' \neq \mathbf{0}$ and $v^* = v' - v$ such that $\mathbf{r}^{\mathsf{T}}\hat{\mathbf{c}}^* = v^*$. To argue that this is difficult, we first claim that the vector $\mathbf{r}$ (in the secret verification key) is computationally hidden from the view of the adversary. This follows via the k-Lin assumption. Under k-Lin, $[\mathbf{w}^{\mathsf{T}}\mathbf{T}]_2$ is pseudorandom given $[\mathbf{T}]_2$ and $[\hat{\mathbf{T}}]_2$. Thus $[\mathbf{z}]_2$ computationally hides the vector $\mathbf{r}$. Since $\mathbf{r}$ is computationally hidden and $\mathbf{r}$ is sampled uniformly from $\mathbb{Z}_p^k$, whenever $\hat{\mathbf{c}}^* \neq \mathbf{0}$, the distribution of $\mathbf{r}^{\mathsf{T}}\hat{\mathbf{c}}^*$ is uniform over $\mathbb{Z}_p$. In this case, for any fixed v^* chosen independently of $\mathbf{r}$, the probability that $\mathbf{r}^{\mathsf{T}}\hat{\mathbf{c}}^* = v^*$ is $1/p$, which is negligible.

Chainable commitments for linear functions. The basic scheme above supports a *fixed* function $\mathbf{M}$, which was programmed into the public parameters $[\mathbf{z}]_2$. To support *arbitrary* functions (as in the case of a functional commitment) from $\mathbb{Z}_p^\ell \to \mathbb{Z}_p^\ell$, we instantiate ℓ^2 copies of the basic scheme. The ℓ^2 schemes can be viewed as functional commitment schemes for the fixed functions $\mathbf{M}_{i,j}$ that is 0 everywhere and 1 in component (i, j). The opening to an arbitrary linear mapping $\mathbf{x} \mapsto \mathbf{M}\mathbf{x}$ then corresponds to taking a linear combination of ℓ^2 openings where the coefficients are defined by the elements of $\mathbf{M}$. To describe the construction more compactly, we start with the following identity: for all $\mathbf{M} \in \mathbb{Z}_p^{\ell\times\ell}$,

$$\mathbf{r}^{\mathsf{T}}\hat{\mathbf{T}}\mathbf{M} = \mathrm{vec}(\mathbf{M})^{\mathsf{T}}(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{r}^{\mathsf{T}}\hat{\mathbf{T}})), \tag{2.1}$$

where $\mathrm{vec}(\mathbf{M})$ is the vectorization operation that takes as input a matrix $\mathbf{M}$ and outputs the vector formed by concatenating the columns of $\mathbf{M}$ from left to right (see Section 3). This means

$$\mathbf{r}^{\mathsf{T}}\hat{\mathbf{T}}\mathbf{M}\mathbf{x} = \mathrm{vec}(\mathbf{M})^{\mathsf{T}}(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{r}^{\mathsf{T}}\hat{\mathbf{T}}))\mathbf{x}.$$

We now sample $\mathbf{W} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^2\times k}$ and publish $[\mathbf{Z}]_2$ in the public parameters where $\mathbf{Z} = \mathbf{W}\mathbf{T} - \mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{r}^{\mathsf{T}}\hat{\mathbf{T}})$. Now,

$$\begin{aligned}\mathrm{vec}(\mathbf{M})^{\mathsf{T}} \cdot \mathbf{Z} \cdot \mathbf{x} &= \mathrm{vec}(\mathbf{M})^{\mathsf{T}}\mathbf{W} \cdot \mathbf{T}\mathbf{x} - \mathrm{vec}(\mathbf{M})^{\mathsf{T}}(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{r}^{\mathsf{T}}\hat{\mathbf{T}}))\mathbf{x} \\ &= \mathrm{vec}(\mathbf{M})^{\mathsf{T}}\mathbf{W} \cdot \mathbf{T}\mathbf{x} - \mathbf{r}^{\mathsf{T}}\hat{\mathbf{T}}\mathbf{M}\mathbf{x}.\end{aligned}$$

We define the opening to be $[v]_2$ where $v = \mathrm{vec}(\mathbf{M})^{\mathsf{T}}\mathbf{Z}\mathbf{x}$. Then, given a Type-II commitment $[\mathbf{c}]_2 = [\mathbf{T}\mathbf{x}]_2$ and an opening $[v]_2$ to a Type-I commitment $[\hat{\mathbf{c}}]_2 = [\hat{\mathbf{T}}\mathbf{M}\mathbf{x}]_2$, the verification algorithm uses the (secret) verification keys $\mathbf{W}$ and $\mathbf{r}$ to check that

$$[v]_2 \stackrel{?}{=} \mathrm{vec}(\mathbf{M})^{\mathsf{T}}\mathbf{W} \cdot [\mathbf{c}]_2 - \mathbf{r}^{\mathsf{T}}[\hat{\mathbf{c}}]_2.$$

Security of the chainable commitment. The chain binding proof for this construction follows exactly as that for the basic construction. Namely, suppose an adversary is able to output $[\mathbf{c}]_2$, $[\hat{\mathbf{c}}]_2$, $[\hat{\mathbf{c}}']_2$, $[v]_2$, and $[v']_2$ such that

$$\mathbf{r}^{\mathsf{T}}\hat{\mathbf{c}} = \mathrm{vec}(\mathbf{M})^{\mathsf{T}}\mathbf{W}\mathbf{c} - v \quad \text{and} \quad \mathbf{r}^{\mathsf{T}}\hat{\mathbf{c}}' = \mathrm{vec}(\mathbf{M})^{\mathsf{T}}\mathbf{W}\mathbf{c} - v'.$$

Just as in the basic case, the adversary in this case is able to come up with $\hat{\mathbf{c}}^* = \hat{\mathbf{c}} - \hat{\mathbf{c}}' \neq \mathbf{0}$ and $v^* = v' - v$ such that $\mathbf{r}^{\mathsf{T}}\hat{\mathbf{c}}^* = v^*$. Similar to the basic case, we can argue via k-Lin that $[\mathbf{W}\mathbf{T}]_2$ is pseudorandom given $[\mathbf{T}]_2$ and $[\hat{\mathbf{T}}]_2$. As such, the vector $\mathbf{r}$ is computationally hidden from the view of the adversary. Then, when $\hat{\mathbf{c}}^* \neq \mathbf{0}$, the distribution of $\mathbf{r}^{\mathsf{T}}\hat{\mathbf{c}}^*$ is uniform over $\mathbb{Z}_p$ and the claim follows exactly as before.

Chainable commitments for quadratic functions. Next, we extend the above construction to obtain a chainable commitment for quadratic functions. In this setting, our goal is to support openings to (homogeneous)[4] quadratic functions $\mathbf{x} \mapsto \mathbf{M}(\mathbf{x} \otimes \mathbf{x})$ where $\mathbf{M} \in \mathbb{Z}_p^{\ell\times\ell^2}$. A basic approach is to linearize the quadratic system and have the user

[4]It suffices to consider homogeneous quadratic functions. We can support arbitrary quadratic functions by having the user commit to the vector $\mathbf{x}' = \left[\begin{smallmatrix}1\\\mathbf{x}\end{smallmatrix}\right]$. A quadratic function on $\mathbf{x}$ then corresponds to a homogeneous quadratic function on $\mathbf{x}'$.

commit to $\mathbf{x} \otimes \mathbf{x}$, and then use the functional commitment for linear functions to open to $\mathbf{M}(\mathbf{x} \otimes \mathbf{x})$. However, this basic approach is not *chainable*: the input is a commitment to a tensored value $\mathbf{x} \otimes \mathbf{x}$, while the output is a commitment to the *untensored* value $\mathbf{y} = \mathbf{M}(\mathbf{x} \otimes \mathbf{x})$. We do not have a way to evaluate a quadratic function on the commitment to $\mathbf{y}$.

We take an alternative approach and replace the Type-II encoding matrix $\mathbf{T}$ with a *pair* of encoding matrices $\mathbf{T}_1, \mathbf{T}_2 \xleftarrow{\text{R}} \mathbb{Z}_p^{k \times \ell}$. A Type-II commitment to $\mathbf{x}$ is now a pair $([\mathbf{T}_1\mathbf{x}]_1, [\mathbf{T}_2\mathbf{x}]_2)$. To construct an opening, the client first computes a tensored commitment $[(\mathbf{T}_1 \otimes \mathbf{T}_2)(\mathbf{x} \otimes \mathbf{x})]_2$ and then applies the chainable commitment for linear functions with $\mathbf{T}_1 \otimes \mathbf{T}_2$ as the input encoding matrix and $\hat{\mathbf{T}}$ as the output encoding matrix. The yields an opening to a Type-I commitment $\hat{\mathbf{T}}\mathbf{M}(\mathbf{x} \otimes \mathbf{x})$ of the output $\mathbf{y} = \mathbf{M}(\mathbf{x} \otimes \mathbf{x})$. We describe our construction below:

- The secret verification key is $\mathbf{r} \xleftarrow{\text{R}} \mathbb{Z}_p^k$ and a matrix $\mathbf{W} \xleftarrow{\text{R}} \mathbb{Z}_p^{\ell^3 \times k^2}$.
- The public key consists of encoding matrices $[\mathbf{T}_1]_1$, $[\mathbf{T}_2]_2$, $[\mathbf{T}_1 \otimes \mathbf{T}_2]_2$, $[\hat{\mathbf{T}}]_2$, and $[\mathbf{Z}]_2$ where $\mathbf{T}_1, \mathbf{T}_2, \hat{\mathbf{T}} \xleftarrow{\text{R}} \mathbb{Z}_p^{k \times \ell}$ and
$$\mathbf{Z} = \mathbf{W}(\mathbf{T}_1 \otimes \mathbf{T}_2) - \mathbf{I}_{\ell^2} \otimes \operatorname{vec}(\mathbf{r}^{\mathsf{T}}\hat{\mathbf{T}}) \in \mathbb{Z}_p^{\ell^3 \times \ell^2}.$$
- A Type-II commitment to a vector $\mathbf{x} \in \mathbb{Z}_p^{\ell}$ is a pair $([\mathbf{c}_1]_1, [\mathbf{c}_2]_2)$ where $\mathbf{c}_1 = \mathbf{T}_1\mathbf{x} \in \mathbb{Z}_p^k$ and $\mathbf{c}_2 = \mathbf{T}_2\mathbf{x} \in \mathbb{Z}_p^k$. A Type-I commitment to a vector $\mathbf{y} \in \mathbb{Z}_p^{\ell}$ is $[\hat{\mathbf{c}}]_2$ where $\hat{\mathbf{c}} = \hat{\mathbf{T}}\mathbf{y} \in \mathbb{Z}_p^k$.
- An opening for the quadratic function $\mathbf{x} \mapsto \mathbf{M}(\mathbf{x} \otimes \mathbf{x})$ where $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell^2}$ consists of the tensored commitment $[\mathbf{c}_*]_2 = [(\mathbf{T}_1 \otimes \mathbf{T}_2)(\mathbf{x} \otimes \mathbf{x})]_2$ and the opening $[v]_2 = [\operatorname{vec}(\mathbf{M})^{\mathsf{T}}\mathbf{Z}(\mathbf{x} \otimes \mathbf{x})]_2$.
- Given a Type-II commitment $\big([\mathbf{c}_1]_1, [\mathbf{c}_2]_2\big)$, a homogeneous quadratic function $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell^2}$, a Type-I commitment $[\hat{\mathbf{c}}]_2$, and an opening $\big([\mathbf{c}_*]_2, [v]_2\big)$, the verification algorithm checks the following two conditions:
$$[\mathbf{c}_1]_1 \otimes [\mathbf{c}_2]_2 \stackrel{?}{=} [1]_1 \cdot [\mathbf{c}_*]_2 \quad \text{and} \quad \mathbf{r}^{\mathsf{T}}[\hat{\mathbf{c}}]_2 \stackrel{?}{=} \operatorname{vec}(\mathbf{M})^{\mathsf{T}}\mathbf{W}[\mathbf{c}_*]_2 - [v]_2.$$
The first verification relation uses the pairing to check that the tensored commitment was correctly computed from the Type-II commitment $([\mathbf{c}_1]_1, [\mathbf{c}_2]_2)$ while the second relation is checking validity of the linearized system.

Both correctness and security follow analogously to that of the linear system. For correctness, we observe the following. If $\mathbf{c}_1 = \mathbf{T}_1\mathbf{x}$, $\mathbf{c}_2 = \mathbf{T}_2\mathbf{x}$ and $\hat{\mathbf{c}} = \hat{\mathbf{T}}\mathbf{y}$, where $\mathbf{y} = \mathbf{M}(\mathbf{x} \otimes \mathbf{x})$, then we have
$$\mathbf{c}_1 \otimes \mathbf{c}_2 = (\mathbf{T}_1\mathbf{x}) \otimes (\mathbf{T}_2\mathbf{x}) = (\mathbf{T}_1 \otimes \mathbf{T}_2)(\mathbf{x} \otimes \mathbf{x}),$$
so the first verification relation passes. For the second verification relation, we appeal to Eq. (2.1) adapted to the case where $\mathbf{M} \in \mathbb{Z}_p^{\ell^2 \times \ell}$:
$$\mathbf{r}^{\mathsf{T}}\hat{\mathbf{T}}\mathbf{M} = \operatorname{vec}(\mathbf{M})^{\mathsf{T}}(\mathbf{I}_{\ell^2} \otimes \operatorname{vec}(\mathbf{r}^{\mathsf{T}}\hat{\mathbf{T}})),$$
Then,
$$\mathbf{r}^{\mathsf{T}}\hat{\mathbf{c}} = \mathbf{r}^{\mathsf{T}}\hat{\mathbf{T}}\mathbf{M}(\mathbf{x} \otimes \mathbf{x}) = \operatorname{vec}(\mathbf{M})^{\mathsf{T}}(\mathbf{I}_{\ell^2} \otimes \operatorname{vec}(\mathbf{r}^{\mathsf{T}}\hat{\mathbf{T}}))(\mathbf{x} \otimes \mathbf{x}) = \operatorname{vec}(\mathbf{M})^{\mathsf{T}}(\mathbf{W}(\mathbf{T}_1 \otimes \mathbf{T}_2) - \mathbf{Z})(\mathbf{x} \otimes \mathbf{x}) = \operatorname{vec}(\mathbf{M})^{\mathsf{T}}\mathbf{W}\mathbf{c}_* - v.$$
To argue evaluation binding, we use a similar strategy and argue that $[\mathbf{W}(\mathbf{T}_1 \otimes \mathbf{T}_2)]_2$ is pseudorandom given $[\mathbf{T}_1]_1$, $[\mathbf{T}_2]_2$, and $[\mathbf{T}_1 \otimes \mathbf{T}_2]_2$. This follows from the *bilateral* k-Lin assumption (since the matrix $\mathbf{T}_1$ is encoded in *both* $\mathbb{G}_1$ and $\mathbb{G}_2$); we provide a formal proof of this in Lemma 3.10. If $[\mathbf{W}(\mathbf{T}_1 \otimes \mathbf{T}_2)]_2$ is pseudorandom, then once again, the vector $\mathbf{r}$ is computationally hidden from the view of the adversary. The analysis then proceeds exactly as in the case for linear functions.

Public verification via k-KerLin. We now show how to lift the designated-verifier constructions described above to the public verification setting. We exploit the fact that the above verification relation is *linear*. As such, we can use the technique from [KW15] of giving out a *partial* encoding of $\mathbf{r}$ and $\mathbf{W}$ and then implementing the verification relation "in the exponent" via the pairing. Specifically, our scheme for quadratic functions now works as follows:

- We first sample a matrix $\mathbf{A} \xleftarrow{\text{R}} \mathbb{Z}_p^{k\times(k+1)}$ and sample $\mathbf{W} \xleftarrow{\text{R}} \mathbb{Z}_p^{\ell^3(k+1)\times k^2}$ and $\mathbf{R} \xleftarrow{\text{R}} \mathbb{Z}_p^{(k+1)\times k}$. The common reference string now contains

$$\mathsf{crs} = \big([\mathbf{A}]_1, [(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}]_1, [\mathbf{AR}]_1, [\mathbf{T}_1]_1, [\mathbf{T}_2]_2, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1 \otimes \mathbf{T}_2]_2, [\mathbf{Z}]_2\big),$$

  where $\mathbf{T}_1, \mathbf{T}_2, \hat{\mathbf{T}} \xleftarrow{\text{R}} \mathbb{Z}_p^{k\times\ell}$ and $\mathbf{Z} = \mathbf{W}(\mathbf{T}_1 \otimes \mathbf{T}_2) - \mathbf{I}_{\ell^2} \otimes \text{vec}(\mathbf{R}\hat{\mathbf{T}})$. In particular $[(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}]_1$ and $[\mathbf{AR}]_1$ are the public encodings of the secret verification keys. The key point is that $\mathbf{A}$ is *compressing* and loses information about $\mathbf{W}$ and $\mathbf{R}$. The reduction then embeds the private key of the designated-verifier scheme into the components of $\mathbf{W}, \mathbf{R}$ that are hidden given $(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}$ and $\mathbf{AR}$.

- The commitments are constructed exactly as in the designated-verifier scheme. Since $\mathbf{r}^{\mathsf{T}}$ has been replaced by a matrix, the analogous opening relation is now $[\mathbf{v}]_2 = [(\text{vec}(\mathbf{M})^{\mathsf{T}} \otimes \mathbf{I}_{k+1})\mathbf{Z}(\mathbf{x} \otimes \mathbf{x})]_2$.

- Given an input commitment $\big([\mathbf{c}_1]_1, [\mathbf{c}_2]_2\big)$, a homogeneous quadratic function $\mathbf{M} \in \mathbb{Z}_p^{\ell\times\ell^2}$, and an opening $\big([\mathbf{c}_*]_2, [\mathbf{v}]_2\big)$, the *public* verification algorithm now checks the following:

$$[\mathbf{c}_1]_1 \otimes [\mathbf{c}_2]_2 \stackrel{?}{=} [1]_1 \cdot [\mathbf{c}_*]_2 \quad \text{and} \quad (\text{vec}(\mathbf{M})^{\mathsf{T}} \otimes \mathbf{I}_k)[(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}]_1[\mathbf{c}_*]_2 \stackrel{?}{=} [\mathbf{AR}]_1[\hat{\mathbf{c}}]_2 + [\mathbf{A}]_1[\mathbf{v}]_2.$$

We refer to Section 4.4 (Construction 4.38) for the full description (which describes the projective variant of this construction). Correctness of this scheme follows by a similar calculation as in the designated-verifier case; we refer to Theorem 4.39 for the exact details. We now provide a brief sketch of the security analysis for this construction.

Consider an adversary for the evaluation binding game. Given the public parameters, the adversary outputs an input commitment $([\mathbf{c}_1]_1, [\mathbf{c}_2]_2)$, a homogeneous quadratic function $\mathbf{M} \in \mathbb{Z}_p^{\ell\times\ell^2}$, two output vectors $[\hat{\mathbf{c}}]_2, [\hat{\mathbf{c}}']_2$ along with two openings $\pi = ([\mathbf{c}_*]_2, [\mathbf{v}]_2)$ and $\pi' = ([\mathbf{c}'_*]_2, [\mathbf{v}']_2)$. If the adversary is successful, then $\hat{\mathbf{c}} \neq \hat{\mathbf{c}}'$ and π and π' are valid openings. If the openings are valid, then $\mathbf{c}_* = \mathbf{c}'_*$ and the verification relation now implies that

$$\mathbf{AR}(\hat{\mathbf{c}} - \hat{\mathbf{c}}') + \mathbf{A}(\mathbf{v} - \mathbf{v}') = \mathbf{0}.$$

Equivalently, we observe that any adversary that breaks evaluation binding must be able to compute $\hat{\mathbf{c}}^* \coloneqq \hat{\mathbf{c}} - \hat{\mathbf{c}}' \neq \mathbf{0}$ and $\mathbf{v}^* \coloneqq \mathbf{v} - \mathbf{v}'$ such that

$$\mathbf{A}(\mathbf{R}\hat{\mathbf{c}}^* + \mathbf{v}^*) = \mathbf{0}. \tag{2.2}$$

Our security proof now proceeds as follows:

- **Step 1:** First we rely on the kernel assumption (k-KerLin), which is a search version of the k-Lin assumption [MRV15] (and thus, implied by k-Lin). The assumption states that given $[\mathbf{A}]_1$ where $\mathbf{A} \xleftarrow{\text{R}} \mathbb{Z}_p^{k\times(k+1)}$, it is difficult to find $[\mathbf{x}]_2$ such that $\mathbf{x} \neq \mathbf{0}$ and $\mathbf{Ax} = \mathbf{0}$. Under the k-KerLin assumption, if an efficient adversary can find $\hat{\mathbf{c}}^*$ and $\mathbf{v}^*$ that satisfies Eq. (2.2), then it must be the case that $\mathbf{R}\hat{\mathbf{c}}^* + \mathbf{v}^* = \mathbf{0}$. Otherwise, the adversary found a non-trivial vector in the kernel of $\mathbf{A}$.

- **Step 2:** Next, we use the fact that $\mathbf{A}$ is *compressing*. Let $\mathbf{a}^\perp \in \mathbb{Z}_p^{k+1}$ be an arbitrary non-zero vector in the kernel of $\mathbf{A}$ (i.e., $\mathbf{A} \cdot \mathbf{a}^\perp = \mathbf{0}$). Suppose we now sample $\mathbf{W}$ and $\mathbf{R}$ as

$$\mathbf{W} = \mathbf{W}_1 + (\mathbf{I}_{\ell^3} \otimes \mathbf{a}^\perp)\mathbf{W}_2$$
$$\mathbf{R} = \mathbf{R}_1 + \mathbf{a}^\perp\mathbf{r}_2^{\mathsf{T}},$$

  where $\mathbf{W}_1 \xleftarrow{\text{R}} \mathbb{Z}_p^{\ell^3(k+1)\times k^2}$, $\mathbf{W}_2 \xleftarrow{\text{R}} \mathbb{Z}_p^{\ell^3\times k^2}$, $\mathbf{R}_1 \xleftarrow{\text{R}} \mathbb{Z}_p^{(k+1)\times k}$, and $\mathbf{r} \xleftarrow{\text{R}} \mathbb{Z}_p^k$. Since $\mathbf{W}_1$ and $\mathbf{R}_1$ are uniform, $\mathbf{W}$ and $\mathbf{R}$ are distributed exactly as in the real public parameters. However, the components $(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}$ and $\mathbf{AR}$ in the public parameters information-theoretically hide the components $\mathbf{W}_2, \mathbf{r}_2$. In particular, since $\mathbf{Aa}^\perp = \mathbf{0}$, we have

$$(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W} = (\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}_1 + (\mathbf{I}_{\ell^3} \otimes \mathbf{Aa}^\perp)\mathbf{W}_2 = (\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}_1$$
$$\mathbf{AR} = \mathbf{AR}_1 + \mathbf{Aa}^\perp\mathbf{r}_2^{\mathsf{T}} = \mathbf{AR}_1.$$

Consider now the verification relation. If $\mathbf{R}\hat{\mathbf{c}}^* + \mathbf{v}^* = \mathbf{0}$, then it must be the case that

$$\mathbf{R}\hat{\mathbf{c}}^* + \mathbf{v}^* = \mathbf{0} \implies \mathbf{a}^{\perp}\mathbf{r}_2^{\mathsf{T}}\hat{\mathbf{c}}^* = -\mathbf{v}^* - \mathbf{R}_1\hat{\mathbf{c}}^*.$$

This is essentially the same type of verification relation as in the designated-verifier setting where $\mathbf{r}_2$ is the secret key. Like in the basic scheme, what remains is to analyze the leakage on $\mathbf{r}_2$ from $\mathbf{Z}$.

- **Step 3:** By a similar argument as in the designated verifier case, we can argue that under bilateral k-Lin, $\mathbf{Z}$ computationally hides $\mathbf{r}_2$. Specifically, we can decompose

$$\mathbf{Z} = \mathbf{W}(\mathbf{T}_1 \otimes \mathbf{T}_2) - \mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{R}\hat{\mathbf{T}}) = \mathbf{Z}_1 + (\mathbf{I}_{\ell^3} \otimes \mathbf{a}^{\perp})(\mathbf{W}_2(\mathbf{T}_1 \otimes \mathbf{T}_2) - \mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_2^{\mathsf{T}}\hat{\mathbf{T}})),$$

where $\mathbf{Z}_1$ does *not* depend on $\mathbf{W}_2$ and $\mathbf{r}_2$. By the bilateral k-Lin assumption, we can show that $[\mathbf{W}_2(\mathbf{T}_1 \otimes \mathbf{T}_2)]_2$ is pseudorandom even given the other components in the public parameters, and thus, computationally hides $\mathbf{r}_2$. The claim now follows exactly as in the designated-verifier case.

We give the formal proof of evaluation binding for quadratic functions in Section 4.4 (Theorem 4.40). The proof of Theorem 4.40 is more involved since it is for the projective variant (see Section 2.2). That notwithstanding, the key steps described here correspond to Lemma 4.43 (Step 1), Lemmas 4.44 and 4.45 (Step 2), and Lemma 4.46 (Step 3).

2.2 Projective Commitments

To go from a chainable commitment for quadratic functions to a functional commitment for general circuits, we introduce the notion of a "projective" commitment. As described at the beginning of Section 2, in a projective commitment, the goal is to take a commitment σ to a vector $\mathbf{x} = (x_1, \ldots, x_\ell)$ and "project" it onto a commitment to a subvector (e.g., the vector $\mathbf{x}' = (x_1, \ldots, x_j)$ for some $j \in [\ell]$). In this work, we will only consider projecting a commitment onto its first j components (i.e., a prefix of length j). Specifically, the syntax of a projective commitment is defined as follows:

- The CRS for a projective commitment can be sampled either in a normal mode or in a projective mode. In this work, we refer to the projective mode as a "semi-functional mode."[5]
- The semi-functional setup algorithm takes as input a Type-I index j_1 and a Type-II index j_2, and outputs a CRS along with two trapdoors td_1 and td_2. The trapdoor td_1 can be used to project Type-I commitments onto a commitment to the first j_1 components. Similarly, the trapdoor td_2 can be used to efficiently project a Type-II commitment onto a commitments to the first j_2 components. We refer to the CRS output by the semi-functional setup algorithm with indices (j_1, j_2) as a (j_1, j_2)-semi-functional CRS. We write $\mathsf{Project}^{(1)}$ and $\mathsf{Project}^{(2)}$ to denote the projection algorithms for Type-I and Type-II commitments, respectively.

The chain binding security requirement now says the following:

- First, suppose $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell^2}$ is the matrix associated with a (homogeneous) quadratic function with the property that the first j_2 components of the output $\mathbf{M}(\mathbf{x} \otimes \mathbf{x})$ only depends on the first j_1 components of $\mathbf{x}$. We say such functions are (j_1, j_2)-local. In other words, given just the first j_1 components of the input vector $\mathbf{x}$, we can compute the first j_2 outputs of $\mathbf{M}(\mathbf{x} \otimes \mathbf{x})$.
- Now, suppose we sample a (j_1, j_2)-semi-functional CRS. Let σ_1 and σ_1' be a pair of Type-I commitments whose projections onto their first j_1 components are equal: $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$. Let σ_2 and σ_2' be a pair of Type-II commitments. Suppose the adversary comes up with valid openings for σ_2 and σ_2' with respect to σ_1 and σ_1', respectively, and with respect to the same (j_1, j_2)-local function $\mathbf{M}$. Projective chain binding security then requires that $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) = \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2')$. Unlike standard evaluation binding, we allow two *different* input commitments σ_1 and σ_1'; the only stipulation is that their projections match. Note that we can define an analogous notion where the inputs are Type-II commitments while the outputs are Type-I commitments.

[5] Specifically, our realization of the projective mode will introduce a "shadow" subspace into the commitments and we embed a copy of the chainable commitment within this shadow subspace. This type of approach is commonly used in dual-system proofs [Wat09, LW10], where a shadow subspace is introduced when constructing the "semi-functional" keys and ciphertexts.

Intuitively, the projective chain binding enforces *local* consistency on the committed values. If a quadratic function is (j_1, j_2)-local, then the adversary should not be able to open two input commitments that "agree" on their first j_1 values to two output commitments that disagree on their first j_2 outputs (since the first j_2 output values are completely determined by the first j_1 input values). We require a few additional properties on the projective commitment:

- For all $j_1, j_2 \in [\ell]$, a (j_1, j_2)-semi-functional CRS should be computationally indistinguishable from a normal CRS.
- For all $j_1, j_2, j_2' \in [\ell]$, a (j_1, j_2)-semi-functional CRS should be computationally indistinguishable from a (j_1, j_2')-semi-functional CRS even given the trapdoor td_1. Likewise, for all $j_1, j_1', j_2 \in [\ell]$, a (j_1, j_2)-semi-functional CRS should be computationally indistinguishable from a (j_1', j_2)-semi-functional CRS even given the trapdoor td_2. Essentially, the first property is saying that if we keep the Type-I index associated with a semi-functional CRS fixed, but change the Type-II index, the projections of a Type-I commitment (i.e., the output of $\mathsf{Project}^{(1)}(\mathsf{td}_1, \cdot)$) do *not* change. This stronger notion of CRS indistinguishability is often referred to as a "no-signaling extraction" property [PR17, KPY19, GZ21, KVZ21, CJJ21].
- Finally, we require a semi-functional collision-resistance property, which essentially says that under a (ℓ, ℓ)-semi-functional CRS (i.e., we are projecting onto all ℓ components of the vector), it should be difficult to find two distinct vector $\mathbf{y} \neq \mathbf{y}'$ whose *honestly-generated* commitments have identical projections.

We provide the formal abstraction as well as the security requirements in Section 4.1.

Constructing projective commitments. To construct a projective commitment scheme, we expand the commitment space. In the basic chainable commitment from Section 2.1, the commitments live in a k-dimensional space. Our projective commitments will live in a $2k$-dimensional vector space where the normal commitments inhabit a k-dimensional space while the "semi-functional" commitments inhabit a k-dimensional shadow subspace. A similar projection approach was used in the delegation scheme from [GZ21]. Concretely, we proceed as follows:

- Let $[\mathbf{B}_1^* \mid \mathbf{B}_2^*] \in \mathbb{Z}_p^{2k \times 2k}$ be a basis for $\mathbb{Z}_p^{2k}$ where $\mathbf{B}_1^*, \mathbf{B}_2^* \in \mathbb{Z}_p^{2k \times k}$. To sample a semi-functional encoding matrix $\mathbf{T}$ that supports projection onto the first j_1 components, we set
$$\mathbf{T} = \mathbf{B}_1^* \mathbf{S}_1 + \mathbf{B}_2^* \mathbf{S}_2,$$
where $\mathbf{S}_1 \xleftarrow{\text{R}} \mathbb{Z}_p^{k \times \ell}$, $\mathbf{S}_2 = [\tilde{\mathbf{S}}_2 \mid \mathbf{0}^{k \times (\ell - j_1)}]$, and $\tilde{\mathbf{S}}_2 \xleftarrow{\text{R}} \mathbb{Z}_p^{k \times j_1}$. In particular, $\mathbf{S}_2$ is random in the first j_1 columns and zero in the remaining $\ell - j_1$ columns.
- Let $\mathbf{B}_2 \in \mathbb{Z}_p^{k \times 2k}$ be the (unique) matrix where $\mathbf{B}_2 \mathbf{B}_1^* = \mathbf{0}$ and $\mathbf{B}_2 \mathbf{B}_2^* = \mathbf{I}_k$. Consider a commitment to a vector $\mathbf{x} \in \mathbb{Z}_p^{\ell}$. A commitment is an encoding of $\mathbf{T}\mathbf{x}$. Then,
$$\mathbf{B}_2 \mathbf{T} \mathbf{x} = \mathbf{B}_2 \big(\mathbf{B}_1^* \mathbf{S}_1 + \mathbf{B}_2^* \mathbf{S}_2 \big) \mathbf{x} = \mathbf{S}_2 \mathbf{x}.$$
Observe that this is essentially a commitment to $\mathbf{x}$ with respect to the new encoding matrix $\mathbf{S}_2$. Moreover, $\mathbf{S}_2$ is *zero* in all but the first j_1 columns. This means that $\mathbf{S}_2$ is a commitment to the first j_1 components of $\mathbf{x}$. Thus, we have successfully projected a commitment $\mathbf{T}\mathbf{x}$ of $\mathbf{x}$ onto a commitment $\mathbf{S}_2\mathbf{x}$ to the first j_1 components of $\mathbf{x}$. In this case, the projection trapdoor is the matrix $\mathbf{B}_2$.

In the actual construction (Construction 4.8), we use a different and independent choice of basis $[\mathbf{B}_1^* \mid \mathbf{B}_2^*]$ for the Type-I and Type-II encoding matrices $\mathbf{T}_1, \mathbf{T}_2, \hat{\mathbf{T}}$. This allows us change the distribution of the Type-I encoding matrix $\hat{\mathbf{T}}$ while retaining the ability to project Type-II commitments (and vice versa).

Arguing projective chain binding. When the CRS is (j_1, j_2)-semi-functional, a Type-II commitment to $\mathbf{x}$ can be viewed as two commitments: a normal commitment to $\mathbf{x}$ in the "normal" subspace, and a semi-functional commitment to the first j_2 components of $\mathbf{x}$ in the "semi-functional" subspace. Our goal is to argue that the scheme satisfies chain binding. This essentially follows by a similar argument as the proof of chain binding security for quadratic functions, except we now implement it in the semi-functional subspace. There is, however, one important difference. Recall from Section 2.1 that the binding analysis critically relied on the fact that $[\mathbf{W}(\mathbf{T}_1 \otimes \mathbf{T}_2)]_2$ computationally hid the value of $[\mathbf{R}]_2$ in $[\mathbf{Z}]_2$ where $\mathbf{Z} = \mathbf{W}(\mathbf{T}_1 \otimes \mathbf{T}_2) - \mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{R}\hat{\mathbf{T}})$. Previously, when $\mathbf{W}, \mathbf{T}_1, \mathbf{T}_2$ were all uniform, we were able to appeal to the k-Lin assumption. If we consider this relation in the semi-functional space, we run into a potential problem. Namely, the input encoding matrices $\mathbf{T}_1$ and $\mathbf{T}_2$ are no longer fully random in the semi-functional space: they are only random in the first j_2 components. As such, our previous proof strategy no longer applies.

Relying on locality. To complete the proof of projective chain binding, we rely on the fact that when the quadratic relation $\mathbf{M}$ is (j_2, j_1)-local,[6] correctness does *not* require giving out all of $\mathbf{Z}$. In particular, we only need to give out a subset of the components of $\mathbf{Z}$ to ensure correctness. Towards this end, we define a projection matrix $\mathbf{P}_{\mathsf{quad}} \in \{0,1\}^{\ell^3 \times \ell^3}$ (a square diagonal matrix) with the following two properties:

- For every (j_2, j_1)-local function $\mathbf{M}$, it holds that $\mathrm{vec}(\mathbf{M})^\intercal \mathbf{P}_{\mathsf{quad}} = \mathrm{vec}(\mathbf{M})^\intercal$. This property ensures correctness for the scheme.
- If we now define $\mathbf{Z}$ to be $\mathbf{W}(\mathbf{T}_1 \otimes \mathbf{T}_2) - (\mathbf{P}_{\mathsf{quad}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{R}\hat{\mathbf{T}}))$, it holds that the non-zero columns of $(\mathbf{P}_{\mathsf{quad}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{R}\hat{\mathbf{T}}))$ in the semi-functional space precisely coincide with the non-zero columns of $\mathbf{W}(\mathbf{T}_1 \otimes \mathbf{T}_2)$ in the semi-functional space. Now, we can rely on the k-Lin assumption to argue that $\mathbf{W}(\mathbf{T}_1 \otimes \mathbf{T}_2)$ hides $\mathbf{R}$ in the semi-functional space. This allows us to essentially implement the original proof strategy of chain binding for quadratic functions described in Section 2.1.

We provide the specific details (including the exact definition of the necessary projection matrix $\mathbf{P}_{\mathsf{quad}}$) in Section 4.4. The proof of projective chain binding for the overall scheme is described in Theorem 4.40.

Additional proof systems. In addition to arguing projective chain binding for quadratic functions, our functional commitment scheme for general circuits relies on two additional systems for proving relations on commitments. These constructions rely on a similar (and simpler) set of techniques as that used to argue security of the projective quadratic commitment. We state the properties we require (since these are needed for our functional commitments scheme in Section 2.3), but defer the details of their construction and analysis to the relevant technical section.

- **Projective commitment for linear functions.** We require a (slimmed-down) version of our projective chainable commitment for quadratic functions that just supports linear functions. While technically this is subsumed by our above construction for quadratic functions, having a scheme for linear functions reduces the size of the openings since it avoids the extra burden of needing to encode the output of the quadratic commitment in both $\mathbb{G}_1$ and $\mathbb{G}_2$. We describe this construction in Section 4.3.
- **Prefix matching.** We require a proof system to show that two commitments σ and σ' share a common prefix (of fixed length k). This will be used to argue consistency between a commitment to the input and a commitment to all of the wires in an arithmetic circuit (which includes the input). The security property essentially says that when the CRS is (k,k)-semi-functional and the prefix-matching proof verifies, then $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma')$. We describe this construction in Section 4.2.

2.3 Functional Commitments for Circuits

Using the projective commitments from Section 2.2, we are now ready to construct our functional commitment for general circuits. We start with a more detailed version of the general overview from the beginning of Section 2:

- To commit to an input $\mathbf{x} \in \mathbb{Z}_p^\ell$, the input commitment consists of a Type-I commitment σ_{in} to $\mathbf{x}$.
- To open σ to a value $\mathbf{y} = C(\mathbf{x})$ where $C \colon \mathbb{Z}_p^\ell \to \mathbb{Z}_p^m$ is a circuit of size s, the user first defines the vector $\mathbf{z} \in \mathbb{Z}_p^s$ to be the vector of *all* of the wire values of $C(\mathbf{x})$, arranged in topological order (i.e., the value of wire i is a function of only the first $i-1$ wires). The user prepares a Type-I commitment σ_1 and a Type-II commitment σ_2 to $\mathbf{z}$.

[6] The relation is (j_2, j_1)-local since the inputs are Type-II commitments while the output is a Type-I commitment.

- The user now constructs the following openings:
  - First, the user uses the prefix-matching proof system to construct a proof π_{pre} that σ_{in} and σ_1 share a common prefix of length ℓ (i.e., they agree on the input).
  - The user gives a chainable linear opening π_{lin} that applying the *identity mapping* $\mathbf{I}_s$ to the Type-I commitment σ_1 yields the Type-II commitment σ_2 (recall that σ_1, σ_2 are both commitments to the wire values $C(\mathbf{x})$).
  - The user gives a chainable quadratic opening π_{quad} that applying the "next-wire" function $\mathbf{M}_C$ to the Type-II commitment σ_2 yields the Type-I commitment σ_1. Here, $\mathbf{M}_C$ is the circuit's "next wire" function whose i^{th} output corresponds to the i^{th} wire of $C(\mathbf{x})$. By construction, $\mathbf{M}_C$ implements the identity function on the first ℓ wires (corresponding to the input), and a quadratic function for the remaining wires. Since the wires are arranged topologically, for all $i \geq \ell$, the function $\mathbf{M}_C$ is $(i, i+1)$-local (i.e., the value of wire $i+1$ is a function of the first i wires only).
  - Finally, the user computes a Type-II commitment σ_{out} to the output $\mathbf{y} = C(\mathbf{x})$, together with a chainable linear opening π_{out} that σ_{out} is consistent with σ_1 under the *linear* projection operator that simply selects for the output wires.

  The opening consists of the commitments to the wires σ_1, σ_2 along with the openings $\pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}$, and π_{out}.

- To verify the opening $\pi = (\sigma_1, \sigma_2, \pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}})$, the verifier first computes the Type-II commitment σ_{out} to the purported output $\mathbf{y}$ itself and checks that each of the underlying openings are valid.

Using the projective commitment schemes described in Section 2.2 (see also Section 4), each of the commitments and openings consists of a constant number of group elements, so we obtain a functional commitment for circuits with constant-size commitments and openings.

Security analysis. We now describe how to leverage the security properties of our projective commitment scheme to argue evaluation binding of the above construction. We provide the formal proof in Section 5. Suppose an adversary comes up with an input commitment σ_{in} along with two openings $\pi = (\sigma_1, \sigma_2, \pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}})$ and $\pi' = (\sigma_1', \sigma_2', \pi_{\mathsf{pre}}', \pi_{\mathsf{lin}}', \pi_{\mathsf{quad}}', \pi_{\mathsf{out}}')$ for vectors $\mathbf{y} \neq \mathbf{y}'$ and with respect to the same circuit C. Our proof shares many similarities with the iterative approaches from [GZ21, CJJ21, KLVW23] for constructing delegation schemes. Specifically, our argument proceeds as follows:

- We start by switching the CRS to be (ℓ, ℓ)-semi-functional. If π_{pre} and π_{pre}' verify, then security of the prefix matching construction now says that

$$\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_{\mathsf{in}}) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1').$$

- Since $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$, the identity function $\mathbf{I}_s$ is (ℓ, ℓ)-local, and $\pi_{\mathsf{lin}}, \pi_{\mathsf{lin}}'$ verify, linear chain-binding (from Type-I to Type-II) then says that $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) = \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2')$.

- Now we switch the CRS to be $(\ell+1, \ell)$-semi-functional. Since only the Type-I index changed, it must be the case that $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) = \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2')$ still holds. This step critically relies on the fact that in the CRS indistinguishability game, the reduction algorithm is given the projection trapdoor, and thus, can project the Type-II commitments and check for equality. Note that because the Type-I index of the CRS has changed, it may no longer be the case that $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$ anymore.

- Since the $\mathbf{M}_C$ circuit is $(\ell, \ell+1)$-local by construction, $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) = \mathsf{Project}^{(1)}(\mathsf{td}_2, \sigma_2')$, and $\pi_{\mathsf{quad}}, \pi_{\mathsf{quad}}'$ verify, quadratic chain-binding (from Type-II to Type-I) now re-establishes the property that $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$.

- Now we switch the CRS to be $(\ell+1, \ell+1)$-semi-functional. Since only the Type-II index changed, this means that $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$ still holds.

- The above sequence of steps allowed us to move the CRS from (ℓ, ℓ)-semi-functional to $(\ell+1, \ell+1)$-semi-functional while maintaining the invariant that $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$. We iterate this same sequence of transitions to conclude that when the CRS is (s, s)-semi-functional, it is still the case that $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$.
- When the CRS is (s, s)-semi-functional, $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$, and $\pi_{\mathrm{out}}, \pi_{\mathrm{out}}'$ verify, we can appeal to linear chain binding to show that the output commitments $\sigma_{\mathrm{out}}, \sigma_{\mathrm{out}}'$ satisfy $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_{\mathrm{out}}) = \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_{\mathrm{out}}')$. However, the verifier computes the output commitments $\sigma_{\mathrm{out}}, \sigma_{\mathrm{out}'}$ from $\mathbf{y}$ and $\mathbf{y}'$ *honestly*. If $\mathbf{y} \neq \mathbf{y}'$, but σ_{out} and σ_{out}' are equal in the semi-functional space, then this breaks the collision resistance property of the projective commitment scheme.

We provide the formal argument in Section 5 (Theorem 5.4). We also refer to Table 2 for a quick overview of the formal hybrid structure. Taken together, this yields the construction in Theorem 1.1.

3 Preliminaries

We write λ to denote the security parameter. For a positive integer $n \in \mathbb{N}$, we write $[n]$ to denote the set $\{1, \ldots, n\}$. For a positive integer $p \in \mathbb{N}$, we write $\mathbb{Z}_p$ to denote the integers modulo p. We use bold uppercase letters to denote matrices (e.g., $\mathbf{A}, \mathbf{B}$) and bold lowercase letters to denote vectors (e.g., $\mathbf{u}, \mathbf{v}$). We use non-boldface letters to refer to their components: $\mathbf{v} = (v_1, \ldots, v_n)$. For a vector $\mathbf{v} = (v_1, \ldots, v_n)$, we write $\mathrm{diag}(\mathbf{v})$ to denote the n-by-n diagonal matrix whose diagonal entries are $(v_1, \ldots, v_n)$. We write $\mathbf{I}_\ell$ to denote the ℓ-by-ℓ identity matrix.

We write $\mathsf{poly}(\lambda)$ to denote a function that is $O(\lambda^c)$ for some constant $c \in \mathbb{N}$ and $\mathsf{negl}(\lambda)$ to denote a function that is $o(\lambda^{-c})$ for all $c \in \mathbb{N}$. We say an algorithm is efficient if it runs in probabilistic polynomial time in the length of its input. We say that two families of distributions $\mathcal{D}_1 = \{\mathcal{D}_{1,\lambda}\}_{\lambda \in \mathbb{N}}$ and $\mathcal{D}_2 = \{\mathcal{D}_{2,\lambda}\}_{\lambda \in \mathbb{N}}$ are computationally indistinguishable if no efficient algorithm can distinguish them with non-negligible probability, and we denote this by writing $\mathcal{D}_1 \stackrel{c}{\approx} \mathcal{D}_2$. We say that $\mathcal{D}_1$ and $\mathcal{D}_2$ are statistically indistinguishable if the statistical distance $\Delta(\mathcal{D}_1, \mathcal{D}_2)$ between the two distributions is bounded by a negligible function $\mathsf{negl}(\lambda)$.

Tensor products and vectorization. For matrices $\mathbf{A} \in \mathbb{Z}_p^{n \times m}$ and $\mathbf{B} \in \mathbb{Z}_p^{k \times \ell}$, we write $\mathbf{A} \otimes \mathbf{B}$ to denote the tensor (Kronecker) product of $\mathbf{A}$ and $\mathbf{B}$. For matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ where the products $\mathbf{AC}$ and $\mathbf{BD}$ are well-defined, the tensor product satisfies the following mixed-product property:

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD}). \tag{3.1}$$

We now state two useful corollaries of the mixed-product property. For a vector $\mathbf{x}$ and a matrix $\mathbf{A}$,

$$(\mathbf{x} \otimes \mathbf{I})\mathbf{A} = (\mathbf{x} \otimes \mathbf{I})(1 \otimes \mathbf{A}) = \mathbf{x} \otimes \mathbf{A}. \tag{3.2}$$

For matrices $\mathbf{A} \in \mathbb{Z}_p^{n \times m}$ and $\mathbf{B} \in \mathbb{Z}_p^{k \times \ell}$,

$$\mathbf{A} \otimes \mathbf{B} = (\mathbf{I}_n \otimes \mathbf{B})(\mathbf{A} \otimes \mathbf{I}_\ell) = (\mathbf{A} \otimes \mathbf{I}_k)(\mathbf{I}_m \otimes \mathbf{B}). \tag{3.3}$$

For a matrix $\mathbf{A} \in \mathbb{Z}_p^{n \times m}$, we write $\mathrm{vec}(\mathbf{A})$ to denote its vectorization (i.e., the vector formed by vertically stacking the columns of $\mathbf{A}$ from leftmost to rightmost). We will use the following useful identity: for matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ where the product $\mathbf{ABC}$ is well-defined, then

$$\mathrm{vec}(\mathbf{ABC}) = (\mathbf{C}^{\mathsf{T}} \otimes \mathbf{A}) \cdot \mathrm{vec}(\mathbf{B}) \quad \text{and} \quad \mathrm{vec}(\mathbf{ABC})^{\mathsf{T}} = \mathrm{vec}(\mathbf{B})^{\mathsf{T}}(\mathbf{C} \otimes \mathbf{A}^{\mathsf{T}}) \tag{3.4}$$

Functional commitments. We now give the formal definition of a fully succinct functional commitment scheme for arithmetic circuits:

Definition 3.1 (Succinct Functional Commitment)**.** Let λ be a security parameter. A succinct functional commitment for arithmetic circuits (over a ring) is a tuple of efficient algorithms $\mathsf{FC} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Eval}, \mathsf{Verify})$ with the following properties:

- $\mathsf{Setup}(1^\lambda, 1^\ell, 1^s) \to \mathsf{crs}$: On input the security parameter λ, the input length ℓ, and the circuit size s, the setup algorithm outputs a common reference string crs. We assume that crs implicitly specifies the input space $\mathcal{R}^\ell$, where $\mathcal{R}$ is a finite ring.

- $\mathsf{Commit}(\mathsf{crs}, \mathbf{x}) \to (\sigma, \mathsf{st})$: On input the common reference string crs and an input $\mathbf{x} \in \mathcal{R}^\ell$, the commitment algorithm outputs a commitment σ and a state st.

- $\mathsf{Eval}(\mathsf{st}, C) \to \pi$: On input a commitment state st, an arithmetic circuit $C \colon \mathcal{R}^\ell \to \mathcal{R}^m$, the evaluation algorithm outputs an opening π.

- $\mathsf{Verify}(\mathsf{crs}, \sigma, C, \mathbf{y}, \pi) \to \{0,1\}$: On input the common reference string crs, a commitment σ, an arithmetic circuit $C \colon \mathcal{R}^\ell \to \mathcal{R}^m$, a value $\mathbf{y} \in \mathcal{R}^m$, and an opening π, the verification algorithm outputs a bit $b \in \{0,1\}$.

We now define several correctness and security properties on the functional commitment scheme:

- **Correctness:** For all $\lambda, \ell, s \in \mathbb{N}$, all crs in the support of $\mathsf{Setup}(1^\lambda, 1^\ell, 1^s)$, all arithmetic circuits $C \colon \mathcal{R}^\ell \to \mathcal{R}^m$ (where $\mathcal{R}$ is the ring determined by crs), all inputs $\mathbf{x} \in \mathcal{R}^\ell$,

$$\Pr\left[\mathsf{Verify}\big(\mathsf{crs}, \sigma, C, C(\mathbf{x}), \pi\big) = 1 : \begin{array}{c} (\sigma, \mathsf{st}) \leftarrow \mathsf{Commit}(\mathsf{crs}, \mathbf{x}); \\ \pi \leftarrow \mathsf{Eval}(\mathsf{st}, C) \end{array}\right] = 1.$$

- **Binding:** For a security parameter λ and an adversary $\mathcal{A}$, we define the binding security game as follows:

  1. On input the security parameter λ, the adversary $\mathcal{A}$ outputs the input length 1^ℓ and the circuit size 1^s.
  2. The challenger samples $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\ell, 1^s)$ and gives crs to $\mathcal{A}$. Let $\mathcal{R}$ be the ring associated with crs.
  3. The adversary outputs a commitment σ, an arithmetic circuit $C \colon \mathcal{R}^\ell \to \mathcal{R}^m$ of size at most s, and vectors $\mathbf{y}, \mathbf{y}' \in \mathcal{R}^m$ along with openings π, π'.
  4. The challenger outputs $b = 1$ if $\mathbf{y} \neq \mathbf{y}'$ and $\mathsf{Verify}(\mathsf{crs}, \sigma, C, \mathbf{y}, \pi) = 1 = \mathsf{Verify}(\mathsf{crs}, \sigma, C, \mathbf{y}', \pi')$. Otherwise, the challenger outputs $b = 0$.

  The functional commitment scheme is binding if for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that $\Pr[b = 1] = \mathsf{negl}(\lambda)$ in the binding security game.

- **Succinctness:** There exists a universal polynomial $\mathsf{poly}(\cdot)$ such that for all $\lambda, \ell, s \in \mathbb{N}$, all crs in the support of $\mathsf{Setup}(1^\lambda, 1^\ell, 1^s)$, all vectors $\mathbf{x} \in \mathcal{R}^\ell$ (where $\mathcal{R}$ is the ring associated with crs), all arithmetic circuits $C \colon \mathcal{R}^\ell \to \mathcal{R}^m$, all (σ, st) in the support of $\mathsf{Commit}(\mathsf{crs}, \mathbf{x})$, and all π in the support of $\mathsf{Eval}(\mathsf{st}, C)$,

$$|\sigma| \leq \mathsf{poly}(\lambda + \log \ell + \log s) \quad \text{and} \quad |\pi| \leq \mathsf{poly}(\lambda + \log \ell + \log s).$$

3.1 Prime-Order Pairing Groups

We start by recalling the definition of a prime-order pairing group and the matrix decision Diffie-Hellman assumption and kernel Diffie-Hellman assumptions we use in this work [EHK+13, MRV15].

Definition 3.2 (Prime-Order Bilinear Group)**.** A prime-order asymmetric pairing group generator GroupGen is an efficient algorithm that takes as input the security parameter 1^λ and outputs a description $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e)$ of two base groups $\mathbb{G}_1$ and $\mathbb{G}_2$ with generators g_1, g_2, respectively, a target group $\mathbb{G}_T$, all of prime order $p = 2^{\Theta(\lambda)}$, and a non-degenerate bilinear map $e \colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. We write $g_T = e(g_1, g_2)$ to denote a generator of $\mathbb{G}_T$. We require that the group operation in $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ and the pairing operations be efficiently computable.

Notation. Let $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e)$ be a prime-order group. As described in Section 2.1, we use the implicit representation of group elements [EHK+13] throughout this work. Namely, for matrices $\mathbf{A}, \mathbf{B}$, we write $[\mathbf{A}]_1$ to denote $g_1^{\mathbf{A}}$ and $[\mathbf{A}]_1 [\mathbf{B}]_2 \coloneqq [\mathbf{AB}]_T$ as well as $[\mathbf{A}]_1 \otimes [\mathbf{B}]_2 \coloneqq [\mathbf{A} \otimes \mathbf{B}]_T$.

Matrix Diffie-Hellman assumptions. We now recall the matrix Diffie-Hellman and kernel Diffie-Hellman assumptions we use in this work. Our presentation is adapted from [EHK+13, MRV15].

Definition 3.3 (k-Lin Assumption)**.** Let GroupGen be a group generator and $k \in \mathbb{N}$ be a positive integer. The k-Lin assumption holds in $\mathbb{G}_2$ with respect to GroupGen if for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$|\Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}]_2, [\mathbf{s}^\mathsf{T}\mathbf{A}]_2) = 1] - \Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}]_2, [\mathbf{u}^\mathsf{T}]_2) = 1]| = \mathsf{negl}(\lambda),$$

where $\mathbf{A} = [\mathbf{1}^k \mid \mathrm{diag}(a_1, \dots, a_k)] \in \mathbb{Z}_p^{k \times (k+1)}$ and the probability is taken over $\mathcal{G} \leftarrow \mathsf{GroupGen}(1^\lambda)$, $a_1, \dots, a_k \xleftarrow{\mathrm{R}} \mathbb{Z}_p$, $\mathbf{s} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$, and $\mathbf{u} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k+1}$.

Definition 3.4 (Matrix Diffie-Hellman Assumption)**.** Let GroupGen be a group generator, and let $k, \ell, d \in \mathbb{N}$ be positive integers. We say that the matrix Diffie-Hellman assumption with parameters k, ℓ, d ($\mathsf{MDDH}_{k,\ell,d}$) holds in $\mathbb{G}_2$ with respect to GroupGen if for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$|\Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}]_2, [\mathbf{SA}]_2) = 1] - \Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}]_2, [\mathbf{U}]_2) = 1]| = \mathsf{negl}(\lambda),$$

where the probability is taken over $\mathcal{G} \leftarrow \mathsf{GroupGen}(1^\lambda)$, $\mathbf{A} \leftarrow \mathbb{Z}_p^{k \times \ell}$, $\mathbf{S} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{d \times k}$, and $\mathbf{U} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{d \times \ell}$.

Definition 3.5 (Kernel Diffie-Hellman Assumption)**.** Let GroupGen be a group generator. We say that the kernel Diffie-Hellman assumption ($\mathsf{KerDH}_{k,\ell}$) holds in $\mathbb{G}_1$ with respect to GroupGen if for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\Pr\left[\mathbf{Ax} = \mathbf{0} \land \mathbf{x} \neq \mathbf{0} : \begin{array}{c} \mathcal{G} \leftarrow \mathsf{GroupGen}(1^\lambda), \mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}, \\ [\mathbf{x}]_2 \leftarrow \mathcal{A}(\mathcal{G}, [\mathbf{A}]_1) \end{array}\right] = \mathsf{negl}(\lambda).$$

We define the $\mathsf{KerDH}_{k,\ell}$ assumption in $\mathbb{G}_2$ analogously (where the challenge $\mathbf{A}$ is encoded in $\mathbb{G}_2$ and the adversary's output is in $\mathbb{G}_1$). Finally, we define the k-KerLin assumption to be an instance of the $\mathsf{KerDH}_{k,k+1}$ assumption where the challenge matrix $\mathbf{A}$ is given by $\mathbf{A} = [\mathbf{1}^k \mid \mathrm{diag}(a_1, \dots, a_k)] \in \mathbb{Z}_p^{k \times (k+1)}$ and $a_1, \dots, a_k \xleftarrow{\mathrm{R}} \mathbb{Z}_p$.

Bilateral MDDH assumptions. Similar to [GZ21], we rely on a bilateral Diffie-Hellman assumption in this work where the challenge is encoded in both $\mathbb{G}_1$ and $\mathbb{G}_2$. We recall the assumptions below:

Definition 3.6 (Bilateral k-Lin Assumption)**.** Let GroupGen be a group generator and $k \in \mathbb{N}$ be a positive integer. The bilateral k-Lin assumption holds with respect to GroupGen if for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$|\Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{s}^\mathsf{T}\mathbf{A}]_1, [\mathbf{s}^\mathsf{T}\mathbf{A}]_2) = 1] - \Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{u}^\mathsf{T}]_1, [\mathbf{u}^\mathsf{T}]_2) = 1]| = \mathsf{negl}(\lambda),$$

where $\mathbf{A} = [\mathbf{1}^k \mid \mathrm{diag}(a_1, \dots, a_k)] \in \mathbb{Z}_p^{k \times (k+1)}$ and the probability is taken over $\mathcal{G} \leftarrow \mathsf{GroupGen}(1^\lambda)$, $a_1, \dots, a_k \xleftarrow{\mathrm{R}} \mathbb{Z}_p$, $\mathbf{s} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$, and $\mathbf{u} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k+1}$.

Definition 3.7 (Bilateral Matrix Diffie-Hellman Assumption)**.** Let GroupGen be a group generator, and let $k, \ell, d \in \mathbb{N}$ be positive integers. We say that the bilateral matrix Diffie-Hellman assumption with parameters k, ℓ, d (bilateral $\mathsf{MDDH}_{k,\ell,d}$) holds with respect to GroupGen if for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$|\Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{SA}]_1, [\mathbf{SA}]_2) = 1] - \Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{U}]_1, [\mathbf{U}]_2) = 1]| = \mathsf{negl}(\lambda),$$

where the probability is taken over $\mathcal{G} \leftarrow \mathsf{GroupGen}(1^\lambda)$, $\mathbf{A} \leftarrow \mathbb{Z}_p^{k \times \ell}$, $\mathbf{S} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{d \times k}$, and $\mathbf{U} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{d \times \ell}$.

Remark 3.8 (Relationship to k-Lin)**.** The analysis of Escala et al. [EHK+13] extends to show that for all $k \geq 1$, the k-Lin assumption implies the $\mathsf{MDDH}_{k,\ell,d}$ assumption for all polynomially-bounded ℓ and d. An analogous result applies for k-KerLin and $\mathsf{KerDH}_{k,\ell}$. This analysis directly extends to the bilateral case when $k > 1$. Finally, Morillo et al. [MRV15] showed that the (standard) $\mathsf{MDDH}_{k,\ell,d}$ in $\mathbb{G}_1$ (resp., $\mathbb{G}_2$) assumption implies the $\mathsf{KerDH}_{k,\ell}$ assumption in $\mathbb{G}_1$ (resp., $\mathbb{G}_2$). Thus, for all $k > 1$ and assuming the bilateral k-Lin assumption holds with respect to GroupGen, both bilateral $\mathsf{MDDH}_{k,\ell,d}$ and $\mathsf{KerDH}_{k,\ell}$ hold with respect to GroupGen.

Tensored MDDH. The security analysis of our functional commitment scheme will rely on a tensored version of the bilateral MDDH assumption. We define this below and show that it is implied by the standard bilateral MDDH assumption (Definition 3.7).

Definition 3.9 (Tensored MDDH). Let GroupGen be a group generator and let $k, \ell_1, \ell_2, d \in \mathbb{N}$ be positive integers. We say the tensored matrix Diffie-Hellman assumption with parameters k, ℓ, d (tensored $\mathsf{MDDH}_{k,\ell_1,\ell_2,d}$) holds in $\mathbb{G}_2$ with respect to GroupGen if for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$|\Pr[\mathcal{A}(\mathcal{G}, X, [\mathbf{S}(\mathbf{A} \otimes \mathbf{B})]_2) = 1] - \Pr[\mathcal{A}(\mathcal{G}, X, [\mathbf{U}]_2) = 1]| = \mathsf{negl}(\lambda),$$

where $X = \big([\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{B}]_1, [\mathbf{B}]_2, [\mathbf{A} \otimes \mathbf{B}]_2\big)$ and the probability is taken over $\mathcal{G} \leftarrow \mathsf{GroupGen}(1^\lambda)$, $\mathbf{A} \leftarrow \mathbb{Z}_p^{k \times \ell_1}$, $\mathbf{B} \leftarrow \mathbb{Z}_p^{k \times \ell_2}$, $\mathbf{S} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{d \times k^2}$, and $\mathbf{U} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{d \times \ell_1 \ell_2}$.

Lemma 3.10. *Let $k, \ell_1, \ell_2, d \in \mathbb{N}$ be positive integers and* GroupGen *be a group generator. If the bilateral* $\mathsf{MDDH}_{k,\ell_1,k}$ *and bilateral* $\mathsf{MDDH}_{k,\ell_2,\ell_1}$ *assumptions hold with respect to* GroupGen*, then for all polynomials $d = d(\lambda)$, the tensored* $\mathsf{MDDH}_{k,\ell_1,\ell_2,d}$ *assumption holds in $\mathbb{G}_2$ with respect to* GroupGen.

Proof. We first show the claim for $d = 1$. The general case then follows by a hybrid argument. When $d = 1$, the goal is to show that the following two distributions are computationally indistinguishable:

$$\big(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{B}]_1, [\mathbf{B}]_2, [\mathbf{A} \otimes \mathbf{B}]_2, [\mathbf{s}^\intercal(\mathbf{A} \otimes \mathbf{B})]_2\big) \overset{c}{\approx} \big(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{B}]_1, [\mathbf{B}]_2, [\mathbf{A} \otimes \mathbf{B}]_2, [\mathbf{u}^\intercal]_2\big), \tag{3.5}$$

where $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell_1}$, $\mathbf{B} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell_2}$, $\mathbf{s} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k^2}$ and $\mathbf{u} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell_1 \ell_2}$. To argue this, we first define $\mathbf{T} \in \mathbb{Z}_p^{k \times k}$ to be the matrix where $\mathsf{vec}(\mathbf{T}) = \mathbf{s}$. Then, by Eq. (3.4),

$$\mathbf{s}^\intercal(\mathbf{A} \otimes \mathbf{B}) = \mathsf{vec}(\mathbf{T})^\intercal(\mathbf{A} \otimes \mathbf{B}) = \mathsf{vec}(\mathbf{B}^\intercal \mathbf{T} \mathbf{A}).$$

Thus, it suffices to show that

$$\big(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{B}]_1, [\mathbf{B}]_2, [\mathbf{A} \otimes \mathbf{B}]_2, [\mathbf{B}^\intercal \mathbf{T} \mathbf{A}]_2\big) \overset{c}{\approx} \big(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{B}]_1, [\mathbf{B}]_2, [\mathbf{A} \otimes \mathbf{B}]_2, [\mathbf{V}]_2\big),$$

where $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell_1}$, $\mathbf{B} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell_2}$, $\mathbf{T} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times k}$, and $\mathbf{V} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell_2 \times \ell_1}$. This follows by applying bilateral MDDH twice (once on the left and once on the right). Formally, we define the following sequence of hybrid experiments:

- Hyb_0: Sample $\mathcal{G} \leftarrow \mathsf{GroupGen}(1^\lambda)$, $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell_1}$, $\mathbf{B} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell_2}$, $\mathbf{T} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times k}$. Output
$$\big(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{B}]_1, [\mathbf{B}]_2, [\mathbf{A} \otimes \mathbf{B}]_2, [\mathbf{B}^\intercal \mathbf{T} \mathbf{A}]_2\big).$$
- Hyb_1: Sample $\mathcal{G} \leftarrow \mathsf{GroupGen}(1^\lambda)$, $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell_1}$, $\mathbf{B} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell_2}$, $\mathbf{T} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times k}$, $\mathbf{U} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell_1}$. Output
$$\big(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{B}]_1, [\mathbf{B}]_2, [\mathbf{A} \otimes \mathbf{B}]_2, [\mathbf{B}^\intercal \mathbf{U}]_2\big).$$
- Hyb_2: Sample $\mathcal{G} \leftarrow \mathsf{GroupGen}(1^\lambda)$, $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell_1}$, $\mathbf{B} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell_2}$, $\mathbf{T} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times k}$, $\mathbf{V} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell_2 \times \ell_1}$. Output
$$\big(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{B}]_1, [\mathbf{B}]_2, [\mathbf{A} \otimes \mathbf{B}]_2, [\mathbf{V}]_2\big).$$

We now argue that each adjacent pair of distributions are computationally indistinguishable under the bilateral MDDH assumption:

- Hyb_0 and Hyb_1 are computationally indistinguishable under bilateral $\mathsf{MDDH}_{k,\ell_1,k}$. Specifically, on input a bilateral $\mathsf{MDDH}_{k,\ell_1,k}$ challenge $(\mathcal{G}, [\tilde{\mathbf{A}}]_1, [\tilde{\mathbf{A}}]_2, [\tilde{\mathbf{Z}}]_1, [\tilde{\mathbf{Z}}]_2)$, the reduction algorithm samples $\mathbf{B} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell_2}$ and constructs the challenge
$$\big(\mathcal{G}, [\tilde{\mathbf{A}}]_1, [\tilde{\mathbf{A}}]_2, [\mathbf{B}]_1, [\mathbf{B}]_2, [\tilde{\mathbf{A}}]_2 \otimes \mathbf{B}, \mathbf{B}^\intercal[\tilde{\mathbf{Z}}]_2\big) = \big(\mathcal{G}, [\tilde{\mathbf{A}}]_1, [\tilde{\mathbf{A}}]_2, [\mathbf{B}]_1, [\mathbf{B}]_2, [\tilde{\mathbf{A}} \otimes \mathbf{B}]_2, [\mathbf{B}^\intercal \tilde{\mathbf{Z}}]_2\big).$$
When $\tilde{\mathbf{Z}} = \mathbf{T}\tilde{\mathbf{A}}$ for $\mathbf{T} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times k}$, this corresponds to Hyb_0 and if $\tilde{\mathbf{Z}} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell_1}$, then this corresponds to Hyb_1.

- Hyb_1 and Hyb_2 are computationally indistinguishable under bilateral $\mathsf{MDDH}_{k,\ell_2,\ell_1}$. Specifically, on input a bilateral $\mathsf{MDDH}_{k,\ell_2,\ell_1}$ challenge $(\mathcal{G}, [\tilde{\mathbf{B}}]_1, [\tilde{\mathbf{B}}]_2, [\tilde{\mathbf{Z}}]_1, [\tilde{\mathbf{Z}}]_2)$, the reduction algorithm samples $\mathbf{A} \xleftarrow{\text{R}} \mathbb{Z}_p^{k \times \ell_1}$ and constructs the challenge

$$\left(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\tilde{\mathbf{B}}]_1, [\tilde{\mathbf{B}}]_2, \mathbf{A} \otimes [\tilde{\mathbf{B}}]_2, [\tilde{\mathbf{Z}}^\intercal]_2\right) = \left(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\tilde{\mathbf{B}}]_1, [\tilde{\mathbf{B}}]_2, [\mathbf{A} \otimes \tilde{\mathbf{B}}]_2, [\tilde{\mathbf{Z}}^\intercal]_2\right).$$

When $\tilde{\mathbf{Z}} = \mathbf{U}\tilde{\mathbf{B}}$ for $\mathbf{U} \xleftarrow{\text{R}} \mathbb{Z}_p^{\ell_1 \times k}$, this corresponds to Hyb_1 and if $\tilde{\mathbf{Z}} \xleftarrow{\text{R}} \mathbb{Z}_p^{\ell_1 \times \ell_2}$, then this corresponds to Hyb_2.

For the general case (i.e., $d > 1$), we proceed via a hybrid argument. For each $i \in \{0, \ldots, d\}$, we define experiment Hyb_i as follows:

- Hyb_i for $i \in \{0, \ldots, d\}$: Sample $\mathcal{G} \leftarrow \mathsf{GroupGen}(1^\lambda)$, $\mathbf{A} \xleftarrow{\text{R}} \mathbb{Z}_p^{k \times \ell_1}$, $\mathbf{B} \xleftarrow{\text{R}} \mathbb{Z}_p^{k \times \ell_2}$, $\mathbf{S} \xleftarrow{\text{R}} \mathbb{Z}_p^{d \times k^2}$. Parse $\mathbf{S} = \begin{bmatrix} \mathbf{S}_1 \\ \mathbf{S}_2 \end{bmatrix}$ where $\mathbf{S}_1 \in \mathbb{Z}_p^{i \times k^2}$ and $\mathbf{S}_2 \in \mathbb{Z}_p^{(d-i) \times k^2}$. Let $\mathbf{V} \xleftarrow{\text{R}} \mathbb{Z}_p^{i \times \ell_1 \ell_2}$. Output

$$\left(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{B}]_1, [\mathbf{B}]_2, [\mathbf{A} \otimes \mathbf{B}]_2, \left[\begin{smallmatrix} \mathbf{V} \\ \mathbf{S}_2(\mathbf{A} \otimes \mathbf{B}) \end{smallmatrix}\right]\right).$$

By construction, the distributions in the bilateral $\mathsf{MDDH}_{k,\ell_1,\ell_2,d}$ assumption correspond to Hyb_0 and Hyb_d. It suffices to show that for all $i \in [d]$, Hyb_{i-1} and Hyb_i are computationally indistinguishable. This reduces to the 1-dimensional case. The reduction algorithm receives a 1-dimensional challenge

$$\left(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{B}]_1, [\mathbf{B}]_2, [\mathbf{A} \otimes \mathbf{B}]_2, [\mathbf{z}^\intercal]_2\right),$$

where $\mathbf{A} \xleftarrow{\text{R}} \mathbb{Z}_p^{k \times \ell_1}$, $\mathbf{B} \xleftarrow{\text{R}} \mathbb{Z}_p^{k \times \ell_2}$ and samples $\mathbf{V} \xleftarrow{\text{R}} \mathbb{Z}_p^{(i-1) \times \ell_1 \ell_2}$ and $\mathbf{S}_2 \xleftarrow{\text{R}} \mathbb{Z}_p^{(d-i) \times k^2}$. It then constructs the challenge

$$\left(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{B}]_1, [\mathbf{B}]_2, [\mathbf{A} \otimes \mathbf{B}]_2, \left[\begin{smallmatrix} [\mathbf{V}]_2 \\ [\mathbf{z}^\intercal]_2 \\ \mathbf{S}_2[\mathbf{A} \otimes \mathbf{B}]_2 \end{smallmatrix}\right]\right).$$

If $\mathbf{z}^\intercal = \mathbf{s}^\intercal(\mathbf{A} \otimes \mathbf{B})$ where $\mathbf{s} \xleftarrow{\text{R}} \mathbb{Z}_p^{k^2}$, then this challenge is distributed according to Hyb_{i-1} whereas if $\mathbf{z} \xleftarrow{\text{R}} \mathbb{Z}_p^{\ell_1 \ell_2}$, then it is distributed according to Hyb_i. Finally, since $d = \mathrm{poly}(\lambda)$, the claim now follows by a hybrid argument. □

4 Projective Commitments from k-Lin

In this section, we introduce and construct the main building blocks that we use for constructing a succinct functional commitment for general circuits from the bilateral k-Lin assumption. Our main construction relies on the ability to project a committed vector onto a subset of its components and argue properties on the projected subset. We start by defining the basic projection matrix we use throughout this section.

Definition 4.1 (Projection Matrix)**.** Let ℓ be a vector dimension. For an index $j \in [\ell]$, define the projection matrix $\mathbf{P}_j \in \{0,1\}^{\ell \times \ell}$ as follows:

$$\mathbf{P}_j := \mathrm{diag}\left([\mathbf{1}^{1 \times j} \mid \mathbf{0}^{1 \times (\ell - j)}]\right) \in \{0,1\}^{\ell \times \ell} \tag{4.1}$$

Namely, for every vector $\mathbf{x} = [x_1, \ldots, x_\ell]^\intercal$, we have $\mathbf{P}_j \mathbf{x} = [x_1, \ldots, x_j, 0, \ldots, 0]^\intercal$.

Local functions. Our constructions in the subsequent sections will also consider local functions, which are functions where some of the outputs only depend on a subset of the inputs.

Definition 4.2 (Local Function)**.** Let $f : \mathcal{X}^\ell \to \mathcal{Y}^m$ be a vector-valued function. For parameters $j_1 \in [\ell]$ and $j_2 \in [m]$, we say that f is (j_1, j_2)-local if the first j_2 outputs of f only depend the first j_1 inputs to f. In other words, if $f_i : \mathcal{X}^\ell \to \mathcal{Y}$ is the function that computes the i^{th} output of f, then for all $i \leq j_2$, the function $f_i(\mathbf{x})$ only depends on the values of $x_1, \ldots, x_{j_1}$. For a set $S \subseteq [\ell] \times [m]$, we say that f is S-local if for all $(j_1, j_2) \in S$, the function f is (j_1, j_2)-local. We refer to S as a "locality set."

4.1 The Base Projective Commitment Scheme

We now define the syntax of our base projective commitment scheme. The base scheme supports two types of commitments (which we refer to as Type I and Type II). The base commitment scheme does *not* provide any useful functionality. However, in the subsequent sections, we augment the base scheme with succinct proof systems for demonstrating relations on Type I and Type II commitments. These proof systems will be used as the main building blocks for our (fully) succinct functional commitment scheme in Section 5.

Projective commitments. In a projective commitment, the CRS for the base scheme can be sampled in a "normal" mode which is used for the real scheme, and a "semi-functional" mode which will be used for the security analysis. When the CRS is sampled in the semi-functional mode, it will be possible to "project" a commitment to a vector $\mathbf{x}$ onto a commitment to the first j components of $\mathbf{x}' = (x_1, \ldots, x_j)$. There are two different projection modes: one for projecting Type-I commitments and one for projecting Type-II commitments. Essentially, the projection operators allow us to "embed" a chainable commitment scheme within the semi-functional space of the projective commitment. We can then leverage a proof strategy similar to [GZ21, CJJ21, KLVW23] in the semi-functional space of the projective commitment scheme to obtain a functional commitment for general arithmetic circuits. We refer to Section 2 for a high-level description and Section 5 for the formal description and analysis. We now describe the syntax and primary security properties we require on our base projective commitment scheme.

Definition 4.3 (Projective Commitment Scheme)**.** A (base) projective commitment scheme $\mathsf{FC} = \big(\mathsf{SetupBase}, \mathsf{SetupSF}, \mathsf{Commit}^{(1)}, \mathsf{Commit}^{(2)}, \mathsf{Project}^{(1)}, \mathsf{Project}^{(2)}\big)$ is a tuple of efficient algorithms with the following syntax:

- $\mathsf{SetupBase}(1^\lambda, 1^\ell) \to \mathsf{crs}_{\mathsf{base}}$: On input the security parameter λ and a vector dimension ℓ, the normal setup algorithm outputs a common reference string $\mathsf{crs}_{\mathsf{base}}$. We assume that $\mathsf{crs}_{\mathsf{base}}$ implicitly contains a description of the input space $\mathcal{R}^\ell$ of the commitment scheme. We require that the input space $\mathcal{R}$ is a ring.
- $\mathsf{SetupSF}(1^\lambda, 1^\ell, j_1, j_2) \to (\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2)$: On input the security parameter λ, a vector dimension ℓ, a Type-I index $j_1 \in [\ell]$, and a Type-II index $j_2 \in [\ell]$, the semi-functional setup algorithm outputs a common reference string $\mathsf{crs}_{\mathsf{base}}$ and projection trapdoors td_1 and td_2.
- $\mathsf{Commit}^{(1)}(\mathsf{crs}_{\mathsf{base}}, \mathbf{x}) \to \sigma_1$: On input the common reference string $\mathsf{crs}_{\mathsf{base}}$ and a vector $\mathbf{x} \in \mathcal{R}^\ell$, the Type-I commitment algorithm outputs a Type-I commitment σ_1. This algorithm is deterministic.
- $\mathsf{Commit}^{(2)}(\mathsf{crs}_{\mathsf{base}}, \mathbf{y}) \to \sigma_2$: On input the common reference string $\mathsf{crs}_{\mathsf{base}}$ and a vector $\mathbf{y} \in \mathcal{R}^\ell$, the Type-II commitment algorithm outputs a Type-II commitment σ_2. This algorithm is deterministic.
- $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) \to \sigma_1'$: On input a Type-I projection trapdoor td_1 and a Type-I commitment σ_1, the Type-I projection algorithm outputs a projected commitment σ_1'. This algorithm is deterministic.
- $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) \to \sigma_2'$: On input a Type-II projection trapdoor td_2 and a commitment σ_2, the Type-I projection algorithm outputs a projected commitment σ_2'. This algorithm is deterministic.

Roadmap. In the remainder of this section, we define the primary security properties we require of the base projective commitment scheme. We summarize these below and follow with the formal definitions:

- **Mode indistinguishability:** The normal CRS (output by Setup) should be computationally indistinguishable from a semi-functional CRS (output by SetupSF).
- **Type-I indistinguishability:** Semi-functional common reference strings with the same Type-II index j_2, but different Type-I indices j_1, j_1', should be computationally indistinguishable even given the Type-II trapdoor td_2.
- **Type-II indistinguishability:** Semi-functional common reference strings with the same Type-I index j_1, but different Type-II indices j_2, j_2', should be computationally indistinguishable even given the Type-I trapdoor td_1.

- **Type-II collision resistance:** When the Type-II index $j_2 = \ell$ is the vector length, then it should be computationally infeasible to find distinct vectors $\mathbf{y} \neq \mathbf{y}'$ whose Type-II commitments are equal in their *semi-functional* components.

In the subsequent sections, we design proof systems for arguing certain properties on the commitments in Construction 4.8:

- **Prefix checking.** If σ_1 and σ_1' are Type-I commitments to vectors $\mathbf{x}, \mathbf{x}'$, respectively, we describe a proof system to argue that $\mathbf{x}$ and $\mathbf{x}'$ share a common prefix. We describe this in Section 4.2.
- **Type-I to Type-II linear mapping.** If σ_1 is a Type-I commitment to a vector $\mathbf{x}$, we describe a proof system to demonstrate that σ_2 is a Type-II commitment on a vector $\mathbf{y} = f(\mathbf{x})$, where f is a *linear* function. We describe this in Section 4.3.
- **Type-II to Type-I quadratic mapping.** If σ_2 is a Type-II commitment to a vector $\mathbf{y}$, we describe a proof system to demonstrate that σ_1 is a Type-I commitment to a vector $\mathbf{x} = f(\mathbf{y})$, where f is a *quadratic* function. We describe this in Section 4.4.

Finally, in Section 5, we show how to use the projective commitment from Construction 4.8 in conjunction with these three proof systems to obtain a functional commitment for arbitrary circuits.

Security properties. We now give the formal definitions of the security properties outlined above.

Definition 4.4 (Mode Indistinguishability)**.** Let FC be a projective commitment scheme where $\mathsf{FC} = \big(\mathsf{SetupBase}, \mathsf{SetupSF}, \mathsf{Commit}^{(1)}, \mathsf{Commit}^{(2)}, \mathsf{Project}^{(1)}, \mathsf{Project}^{(2)}\big)$. For a bit $b \in \{0,1\}$ and an adversary $\mathcal{A}$, we define the mode indistinguishability game $\mathsf{ExptMI}_{\mathcal{A}}[\lambda, b]$ as follows:

1. On input the security parameter λ, algorithm $\mathcal{A}$ outputs the input length 1^ℓ, and indices $j_1, j_2 \in [\ell]$.
2. The challenger samples the CRS as follows:
   - If $b = 0$, $\mathsf{crs}_{\mathsf{base}} \leftarrow \mathsf{SetupBase}(1^\lambda, 1^\ell)$.
   - If $b = 1$, $(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^\lambda, 1^\ell, j_1, j_2)$.

   The challenger gives $\mathsf{crs}_{\mathsf{base}}$ to $\mathcal{A}$.
3. Algorithm $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ which is the output of the experiment.

The projective commitment scheme FC satisfies mode indistinguishability if for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\big|\Pr[\mathsf{ExptMI}_{\mathcal{A}}[\lambda, 0] = 1] - \Pr[\mathsf{ExptMI}_{\mathcal{A}}[\lambda, 0] = 1]\big| = \mathsf{negl}(\lambda).$$

Definition 4.5 (Type-I Indistinguishability)**.** Let FC be a projective commitment scheme where $\mathsf{FC} = \big(\mathsf{SetupBase}, \mathsf{SetupSF}, \mathsf{Commit}^{(1)}, \mathsf{Commit}^{(2)}, \mathsf{Project}^{(1)}, \mathsf{Project}^{(2)}\big)$. For a bit $b \in \{0,1\}$ and an adversary $\mathcal{A}$, we define the Type-I indistinguishability game $\mathsf{ExptTI}_{\mathcal{A}}[\lambda, b]$ as follows:

1. On input the security parameter λ, algorithm $\mathcal{A}$ outputs the input length 1^ℓ, two Type-I indices $j_1, j_1' \in [\ell]$, and a Type-II index $j_2 \in [\ell]$,
2. The challenger samples the CRS as follows:
   - If $b = 0$, $(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^\lambda, 1^\ell, j_1, j_2)$.
   - If $b = 1$, $(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^\lambda, 1^\ell, j_1', j_2)$.

   The challenger gives $\mathsf{crs}_{\mathsf{base}}$ and td_2 to $\mathcal{A}$.

3. Algorithm $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ which is the output of the experiment.

The projective commitment scheme FC satisfies Type-I indistinguishability if for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\left|\Pr[\mathsf{ExptTI}_{\mathcal{A}}[\lambda,0]=1]-\Pr[\mathsf{ExptTI}_{\mathcal{A}}[\lambda,0]=1]\right| = \mathsf{negl}(\lambda).$$

Definition 4.6 (Type-II Indistinguishability)**.** Let FC be a projective commitment scheme where $\mathsf{FC} = (\mathsf{SetupBase}, \mathsf{SetupSF}, \mathsf{Commit}^{(1)}, \mathsf{Commit}^{(2)}, \mathsf{Project}^{(1)}, \mathsf{Project}^{(2)})$. For a bit $b \in \{0,1\}$ and an adversary $\mathcal{A}$, we define the Type-II indistinguishability game $\mathsf{ExptTII}_{\mathcal{A}}[\lambda, b]$ as follows:

1. On input the security parameter λ, algorithm $\mathcal{A}$ outputs the input length 1^{ℓ}, a Type-I index $j_1 \in [\ell]$, and two Type-II indices $j_2, j_2' \in [\ell]$.
2. The challenger samples the CRS as follows:
   - If $b = 0$, $(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^{\lambda}, 1^{\ell}, j_1, j_2)$.
   - If $b = 1$, $(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^{\lambda}, 1^{\ell}, j_1, j_2')$.

   The challenger gives $\mathsf{crs}_{\mathsf{base}}$ and td_1 to $\mathcal{A}$.
3. Algorithm $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ which is the output of the experiment.

The projective commitment scheme FC satisfies Type-II indistinguishability if for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\left|\Pr[\mathsf{ExptTII}_{\mathcal{A}}[\lambda,0]=1]-\Pr[\mathsf{ExptTII}_{\mathcal{A}}[\lambda,0]=1]\right| = \mathsf{negl}(\lambda).$$

Definition 4.7 (Type-II Collision Resistance)**.** Let FC be a projective commitment scheme where $\mathsf{FC} = (\mathsf{SetupBase}, \mathsf{SetupSF}, \mathsf{Commit}^{(1)}, \mathsf{Commit}^{(2)}, \mathsf{Project}^{(1)}, \mathsf{Project}^{(2)})$. For an adversary $\mathcal{A}$, we define the Type-II collision resistance game as follows:

1. On input the security parameter λ, algorithm $\mathcal{A}$ outputs the input length 1^{ℓ} and a Type-I index $j_1 \in [\ell]$.
2. The challenger samples $(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^{\lambda}, 1^{\ell}, j_1, \ell)$ and gives $\mathsf{crs}_{\mathsf{base}}$ to $\mathcal{A}$.
3. Algorithm $\mathcal{A}$ outputs two vectors $\mathbf{y}, \mathbf{y}' \in \mathcal{R}^{\ell}$, where $\mathcal{R}^{\ell}$ is the input space associated with $\mathsf{crs}_{\mathsf{base}}$.
4. The challenger then computes $\sigma_2 = \mathsf{Commit}^{(2)}(\mathsf{crs}_{\mathsf{base}}, \mathbf{y})$ and $\sigma_2' = \mathsf{Commit}^{(2)}(\mathsf{crs}_{\mathsf{base}}, \mathbf{y}')$. The output of the experiment is $b = 1$ if
$$\mathbf{y} \neq \mathbf{y}' \quad \text{and} \quad \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) = \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2').$$
   Otherwise, the experiment outputs $b = 0$.

We say FC satisfies Type-II collision resistance if for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that $\Pr[b = 1] = \mathsf{negl}(\lambda)$ in the Type-II collision resistance security game.

Constructing projective commitments from pairings. We now describe our base projective commitment scheme from pairings and then show that it satisfies the security properties listed above (under the bilateral k-Lin assumption).

Construction 4.8 (Projective Commitment Scheme)**.** Let $k \in \mathbb{N}$ be a constant and GroupGen be a prime-order pairing group generator. Our base projective commitment scheme $\mathsf{FC} = (\mathsf{SetupBase}, \mathsf{SetupSF}, \mathsf{Commit}^{(1)}, \mathsf{Commit}^{(2)}, \mathsf{Project}^{(1)}, \mathsf{Project}^{(2)})$ is defined as follows:

- $\mathsf{SetupBase}(1^\lambda, 1^\ell)$: On input the security parameter λ and the input length ℓ, the setup algorithm samples $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e) \leftarrow \mathsf{GroupGen}(1^\lambda)$. Then, it samples $\hat{\mathbf{T}}, \mathbf{T}_1, \mathbf{T}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k\times\ell}$ and sets $\mathbf{T}_* = \mathbf{T}_1 \otimes \mathbf{T}_2$. It outputs the common reference string

$$\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big).$$

  The input space associated with $\mathsf{crs}_{\mathsf{base}}$ is the ring $\mathbb{Z}_p$.

- $\mathsf{SetupSF}(1^\lambda, 1^\ell, j_1, j_2)$: On input the security parameter λ, the input length ℓ, the Type-I index $j_1 \in [\ell]$, and the Type-II index $j_2 \in [\ell]$, the semi-functional setup algorithm samples the following components:
  - Sample $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e) \leftarrow \mathsf{GroupGen}(1^\lambda)$.
  - Sample full-rank matrices $\hat{\mathbf{B}}, \mathbf{B}_1, \mathbf{B}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k\times 2k}$ and define $\hat{\mathbf{B}}^* = \hat{\mathbf{B}}^{-1}$, $\mathbf{B}_1^* = \mathbf{B}_1^{-1}$, and $\mathbf{B}_2^* = \mathbf{B}_2^{-1}$. It parses the matrices as
$$\hat{\mathbf{B}} = \begin{bmatrix} \hat{\mathbf{B}}_1 \\ \hat{\mathbf{B}}_2 \end{bmatrix} \quad \text{and} \quad \mathbf{B}_1 = \begin{bmatrix} \mathbf{B}_{1,1} \\ \mathbf{B}_{1,2} \end{bmatrix} \quad \text{and} \quad \mathbf{B}_2 = \begin{bmatrix} \mathbf{B}_{2,1} \\ \mathbf{B}_{2,2} \end{bmatrix}, \tag{4.2}$$
    where $\hat{\mathbf{B}}_1, \hat{\mathbf{B}}_2, \mathbf{B}_{1,1}, \mathbf{B}_{1,2}, \mathbf{B}_{2,1}, \mathbf{B}_{2,2} \in \mathbb{Z}_p^{k\times 2k}$. Similarly, it parses
$$\hat{\mathbf{B}}^* = \big[\hat{\mathbf{B}}_1^*, \mid \hat{\mathbf{B}}_2^*\big] \quad \text{and} \quad \mathbf{B}_1^* = \big[\mathbf{B}_{1,1}^* \mid \mathbf{B}_{1,2}^*\big] \quad \text{and} \quad \mathbf{B}_2^* = \big[\mathbf{B}_{2,1}^* \mid \mathbf{B}_{2,2}^*\big], \tag{4.3}$$
    where $\hat{\mathbf{B}}_1^*, \hat{\mathbf{B}}_2^*, \mathbf{B}_{1,1}^*, \mathbf{B}_{1,2}^*, \mathbf{B}_{2,1}^*, \mathbf{B}_{2,2}^* \in \mathbb{Z}_p^{2k\times k}$.
  - Construct the encoding matrices $\hat{\mathbf{T}}, \mathbf{T}_1, \mathbf{T}_2$ as follows:
    - **Type-I encodings:** Sample $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k\times\ell}$ and let $\hat{\mathbf{T}} = \hat{\mathbf{B}}_1^*\hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^*\hat{\mathbf{S}}_2\mathbf{P}_{j_1} \in \mathbb{Z}_p^{2k\times\ell}$.
    - **Type-II encodings:** For $\alpha \in \{1, 2\}$, sample $\mathbf{S}_{\alpha,1}, \mathbf{S}_{\alpha,2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k\times\ell}$. Let $\mathbf{T}_\alpha = \mathbf{B}_{\alpha,1}^*\mathbf{S}_{\alpha,1} + \mathbf{B}_{\alpha,2}^*\mathbf{S}_{\alpha,2}\mathbf{P}_{j_2} \in \mathbb{Z}_p^{2k\times\ell}$,

    where $\mathbf{P}_{j_1}, \mathbf{P}_{j_2}$ are the projection matrices from Definition 4.1. Then, let $\mathbf{T}_* = \mathbf{T}_1 \otimes \mathbf{T}_2$.

  The setup algorithm outputs the common reference string $\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big)$ and the projection trapdoors $\mathsf{td}_1 = \hat{\mathbf{B}}_2$ and $\mathsf{td}_2 = (\mathbf{B}_{1,2}, \mathbf{B}_{2,2})$. The message space associated with $\mathsf{crs}_{\mathsf{base}}$ is the ring $\mathbb{Z}_p$.

- $\mathsf{Commit}^{(1)}(\mathsf{crs}_{\mathsf{base}}, \mathbf{x})$: On input the common reference string $\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big)$ and a vector $\mathbf{x} \in \mathbb{Z}_p^\ell$, the Type-I commitment algorithm computes $[\hat{\mathbf{c}}]_2 \leftarrow [\hat{\mathbf{T}}]_2\mathbf{x} = [\hat{\mathbf{T}}\mathbf{x}]_2$. It outputs $\sigma_1 = [\hat{\mathbf{c}}]_2 \in \mathbb{G}_2^{2k}$.

- $\mathsf{Commit}^{(2)}(\mathsf{crs}_{\mathsf{base}}, \mathbf{y})$: On input the common reference string $\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big)$ and a vector $\mathbf{y} \in \mathbb{Z}_p^\ell$, the Type-II commitment algorithm computes $[\mathbf{c}_1]_1 \leftarrow [\mathbf{T}_1]_1\mathbf{y} = [\mathbf{T}_1\mathbf{y}]_1 \in \mathbb{G}_1^{2k}$ and $[\mathbf{c}_2]_2 \leftarrow [\mathbf{T}_2]_2\mathbf{y} = [\mathbf{T}_2\mathbf{y}]_2 \in \mathbb{G}_2^{2k}$. It outputs the commitment $\sigma_2 = ([\mathbf{c}_1]_1, [\mathbf{c}_2]_2)$.

- $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1)$: On input a Type-I projection trapdoor $\mathsf{td}_1 = \hat{\mathbf{B}}_2$, and a commitment $\sigma_1 = [\hat{\mathbf{c}}]_2$, output $\hat{\mathbf{B}}_2[\sigma_1]_2$.

- $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2)$: On input a Type-II projection trapdoor $\mathsf{td}_2 = (\mathbf{B}_{1,2}, \mathbf{B}_{2,2})$ and a commitment $\sigma_2 = ([\mathbf{c}_1]_1, [\mathbf{c}_2]_2)$, output $(\mathbf{B}_{1,2}[\mathbf{c}_1]_1, \mathbf{B}_{2,2}[\mathbf{c}_2]_2)$.

Theorem 4.9 (Mode Indistinguishability). *If the bilateral k-*Lin *assumption holds with respect to* GroupGen*, then Construction 4.8 satisfies mode indistinguishability.*

Proof. Take any adversary $\mathcal{A}$ for the mode indistinguishability game, and let ℓ, j_1, j_2 be the values chosen by the adversary $\mathcal{A}$. We define a sequence of hybrid experiments:

- Hyb_0: This is experiment $\mathsf{ExptMI}_{\mathcal{A}}[\lambda, 0]$. In this experiment, the challenger samples $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e) \leftarrow \mathsf{GroupGen}(1^\lambda)$. It also samples $\hat{\mathbf{T}}, \mathbf{T}_1, \mathbf{T}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k\times\ell}$, computes $\mathbf{T}_* \leftarrow \mathbf{T}_1 \otimes \mathbf{T}_2$ and outputs

$$\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big)$$

- Hyb_1: Same as Hyb_0, except the challenger samples $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$ and $\hat{\mathbf{B}}_1^*, \hat{\mathbf{B}}_2^* \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times k}$. It then sets

$$\hat{\mathbf{T}} = \hat{\mathbf{B}}_1^* \hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^* \hat{\mathbf{S}}_2 \mathbf{P}_{j_1} \in \mathbb{Z}_p^{2k \times \ell}.$$

- Hyb_2: Same as Hyb_1, except the challenger samples $\mathbf{S}_{1,1}, \mathbf{S}_{1,2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$ and $\mathbf{B}_{1,1}^*, \mathbf{B}_{1,2}^* \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times k}$. It then sets

$$\mathbf{T}_1 = \mathbf{B}_{1,1}^* \mathbf{S}_{1,1} + \mathbf{B}_{1,2}^* \mathbf{S}_{1,2} \mathbf{P}_{j_2} \in \mathbb{Z}_p^{2k \times \ell}.$$

- Hyb_3: Same as Hyb_2, except the challenger samples $\mathbf{S}_{2,1}, \mathbf{S}_{2,2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$ and $\mathbf{B}_{2,1}^*, \mathbf{B}_{2,2}^* \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times k}$. It then sets

$$\mathbf{T}_2 = \mathbf{B}_{2,1}^* \mathbf{S}_{2,1} + \mathbf{B}_{2,2}^* \mathbf{S}_{2,2} \mathbf{P}_{j_2} \in \mathbb{Z}_p^{2k \times \ell}.$$

  This is $\mathsf{ExptMI}_{\mathcal{A}}[\lambda, 1]$.

We now argue that each adjacent pair of hybrid experiments is computationally indistinguishable. In the following, we implicitly use the fact that sampling $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times 2k}$ is statistically indistinguishable from sampling a *full rank* $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times 2k}$.

- Hybrids Hyb_0 and Hyb_1 are computationally indistinguishable under the $\mathsf{MDDH}_{k,\ell,2k}$ assumption in $\mathbb{G}_2$. Given the $\mathsf{MDDH}_{k,\ell,2k}$ challenge $(\mathcal{G}, [\mathbf{A}]_2, [\mathbf{V}]_2)$ where $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$ and $\mathbf{V} \in \mathbb{Z}_p^{2k \times \ell}$, the reduction algorithm samples $\mathbf{T}_1, \mathbf{T}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times \ell}$ and $\hat{\mathbf{S}}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$, $\hat{\mathbf{B}}_2^* \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times k}$. It creates the CRS

$$\mathsf{crs}_{\mathsf{base}} = \left(\mathcal{G}, [\mathbf{V}]_2 + \hat{\mathbf{B}}_2^* \hat{\mathbf{S}}_2 \mathbf{P}_{j_1}, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_1 \otimes \mathbf{T}_2]_2\right).$$

  When $\mathbf{V} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times \ell}$, this corresponds to the distribution in Hyb_0 and when $\mathbf{V} = \mathbf{S}\mathbf{A}$ where $\mathbf{S} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times k}$ and $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$, this corresponds to the distribution in Hyb_1.

- Hybrids Hyb_1 and Hyb_2 are computationally indistinguishable under the bilateral $\mathsf{MDDH}_{k,\ell,2k}$ assumption. Given the bilateral $\mathsf{MDDH}_{k,\ell,2k}$ challenge $(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{V}]_1, [\mathbf{V}]_2)$, the reduction algorithm samples $\mathbf{T}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times \ell}$. It also samples $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2, \mathbf{S}_{1,2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$ and $\hat{\mathbf{B}}_1^*, \hat{\mathbf{B}}_2^*, \mathbf{B}_{1,2}^* \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times k}$. It sets $\hat{\mathbf{T}} = \hat{\mathbf{B}}_1^* \hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^* \hat{\mathbf{S}}_2 \mathbf{P}_{j_1}$. It creates the CRS

$$\mathsf{crs}_{\mathsf{base}} = \left(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{V}]_1 + \mathbf{B}_{1,2}^* \mathbf{S}_{1,2} \mathbf{P}_{j_2}, [\mathbf{V}]_2 + \mathbf{B}_{1,2}^* \mathbf{S}_{1,2} \mathbf{P}_{j_2}, [\mathbf{T}_2]_2, \left([\mathbf{V}]_2 + \mathbf{B}_{1,2}^* \mathbf{S}_{1,2} \mathbf{P}_{j_2}\right) \otimes \mathbf{T}_2\right).$$

  When $\mathbf{V} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times \ell}$, this corresponds to the distribution in Hyb_1 and when $\mathbf{V} = \mathbf{S}\mathbf{A}$ where $\mathbf{S} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times k}$ and $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$, this corresponds to the distribution in Hyb_2.

- Hybrids Hyb_2 and Hyb_3 are computationally indistinguishable under the $\mathsf{MDDH}_{k,\ell,2k}$ assumption in $\mathbb{G}_2$. Given the $\mathsf{MDDH}_{k,\ell,2k}$ challenge $(\mathcal{G}, [\mathbf{A}]_2, [\mathbf{V}]_2)$ where $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$ and $\mathbf{V} \in \mathbb{Z}_p^{2k \times \ell}$, the reduction algorithm samples $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2, \mathbf{S}_{1,1}, \mathbf{S}_{1,2}, \mathbf{S}_{2,2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$ and $\hat{\mathbf{B}}_1^*, \hat{\mathbf{B}}_2^*, \mathbf{B}_{1,1}^* \mathbf{B}_{1,2}^*, \mathbf{B}_{2,2}^* \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times k}$. It sets $\hat{\mathbf{T}} = \hat{\mathbf{B}}_1^* \hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^* \hat{\mathbf{S}}_2 \mathbf{P}_{j_1}$ and $\mathbf{T}_1 = \mathbf{B}_{1,1}^* \mathbf{S}_{1,1} + \mathbf{B}_{1,2}^* \mathbf{S}_{1,2} \mathbf{P}_{j_2}$. It creates the CRS

$$\mathsf{crs}_{\mathsf{base}} = \left(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{V}]_2 + \mathbf{B}_{2,2}^* \mathbf{S}_{2,2} \mathbf{P}_{j_2}, \mathbf{T}_1 \otimes \left([\mathbf{V}]_2 + \mathbf{B}_{2,2}^* \mathbf{S}_{2,2} \mathbf{P}_{j_2}\right)\right).$$

  When $\mathbf{V} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times \ell}$, this corresponds to the distribution in Hyb_2 and when $\mathbf{V} = \mathbf{S}\mathbf{A}$ where $\mathbf{S} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times k}$ and $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$, this corresponds to the distribution in Hyb_3.

Since $\ell = \mathrm{poly}(\lambda)$, the bilateral k-Lin assumption implies each of the underlying MDDH assumption we use in the above analysis (Remark 3.8), the theorem now follows by a hybrid argument. $\square$

Theorem 4.10 (Type-I Indistinguishability). *If the k-Lin assumption holds in $\mathbb{G}_2$ with respect to* GroupGen*, then Construction 4.8 satisfies Type-I indistinguishability.*

Proof. Let $\mathcal{A}$ be an adversary and let ℓ, j_1, j_1', j_2 be the values it chooses. We proceed via a hybrid argument:

- Hyb_0: This is $\mathsf{ExptTI}_{\mathcal{A}}[\lambda, 0]$. In this experiment, the challenger samples $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e) \leftarrow \mathsf{GroupGen}(1^\lambda)$. It samples $\hat{\mathbf{B}}_1^*, \hat{\mathbf{B}}_2^* \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times k}$, $\mathbf{B}_1, \mathbf{B}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times 2k}$, and defines $\mathbf{B}_1^* = \mathbf{B}_1^{-1}$ and $\mathbf{B}_2^* = \mathbf{B}_2^{-1}$. It parses $\mathbf{B}_1, \mathbf{B}_2$ into matrices $\mathbf{B}_{1,1}, \mathbf{B}_{1,2}, \mathbf{B}_{2,1}, \mathbf{B}_{2,2} \in \mathbb{Z}_p^{k \times 2k}$ according to Eq. (4.2) and $\mathbf{B}_1^*, \mathbf{B}_2^*$ into $\mathbf{B}_{1,1}^*, \mathbf{B}_{1,2}^*, \mathbf{B}_{2,1}^*, \mathbf{B}_{2,2}^* \in \mathbb{Z}_p^{2k \times k}$ according to Eq. (4.3). Next, it samples $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2, \mathbf{S}_{1,1}, \mathbf{S}_{1,2}, \mathbf{S}_{2,1}, \mathbf{S}_{2,2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$. It sets
$$\hat{\mathbf{T}} = \hat{\mathbf{B}}_1^* \hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^* \hat{\mathbf{S}}_2 \mathbf{P}_{j_1} \quad \text{and} \quad \mathbf{T}_1 = \mathbf{B}_{1,1}^* \mathbf{S}_{1,1} + \mathbf{B}_{1,2}^* \mathbf{S}_{1,2} \mathbf{P}_{j_2} \quad \text{and} \quad \mathbf{T}_2 = \mathbf{B}_{2,1}^* \mathbf{S}_{2,1} + \mathbf{B}_{2,2}^* \mathbf{S}_{2,2} \mathbf{P}_{j_2}.$$
Finally, it computes $\mathbf{T}_* = \mathbf{T}_1 \otimes \mathbf{T}_2$ and outputs
$$\mathsf{crs}_{\mathsf{base}} = \left(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\right)$$
along with $\mathsf{td}_2 = (\mathbf{B}_{1,2}, \mathbf{B}_{2,2})$.
- Hyb_1: Same as Hyb_0 except the challenger samples $\hat{\mathbf{T}} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times \ell}$.
- Hyb_2: Same as Hyb_0 except the challenger samples $\hat{\mathbf{T}} = \hat{\mathbf{B}}_1^* \hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^* \hat{\mathbf{S}}_2 \mathbf{P}_{j_1'}$. This is $\mathsf{ExptTI}_{\mathcal{A}}[\lambda, 1]$.

We now show that each adjacent pair of hybrid experiments is computationally indistinguishable. As before, we implicitly use the fact that sampling $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times 2k}$ is statistically indistinguishable from sampling a *full rank* $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times 2k}$.

- Hybrids Hyb_0 and Hyb_1 are computationally indistinguishable under the $\mathsf{MDDH}_{k,\ell,2k}$ assumption in $\mathbb{G}_2$. Given the $\mathsf{MDDH}_{k,\ell,2k}$ challenge $(\mathcal{G}, [\mathbf{A}]_2, [\mathbf{V}]_2)$ where $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$ and $\mathbf{V} \in \mathbb{Z}_p^{2k \times \ell}$, the reduction algorithm samples $\mathbf{B}_1, \mathbf{B}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times 2k}$ and defines $\mathbf{B}_1^* = \mathbf{B}_1^{-1}$, $\mathbf{B}_2^* = \mathbf{B}_2^{-1}$. Then it samples $\mathbf{S}_{1,1}, \mathbf{S}_{1,2}, \mathbf{S}_{2,1}, \mathbf{S}_{2,2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$ and constructs $\mathbf{T}_1 = \mathbf{B}_{1,1}^* \mathbf{S}_{1,1} + \mathbf{B}_{1,2}^* \mathbf{S}_{1,2} \mathbf{P}_{j_2}$, and $\mathbf{T}_2 = \mathbf{B}_{2,1}^* \mathbf{S}_{2,1} + \mathbf{B}_{2,2}^* \mathbf{S}_{2,2} \mathbf{P}_{j_2}$, where the components $\mathbf{B}_{1,1}^*, \mathbf{B}_{1,2}^*, \mathbf{B}_{2,1}^*, \mathbf{B}_{2,2}^*$ are obtained from $\mathbf{B}_1^*, \mathbf{B}_2^*$ according to Eq. (4.3). Finally, it samples $\hat{\mathbf{S}}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$, $\hat{\mathbf{B}}_2^* \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times k}$ and creates the CRS
$$\mathsf{crs}_{\mathsf{base}} = \left(\mathcal{G}, [\mathbf{V}]_2 + \hat{\mathbf{B}}_2^* \hat{\mathbf{S}}_2 \mathbf{P}_{j_1}, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_1 \otimes \mathbf{T}_2]_2\right)$$
and the trapdoor $\mathsf{td}_2 = (\mathbf{B}_{1,2}, \mathbf{B}_{2,2})$ where $\mathbf{B}_{1,2}$ and $\mathbf{B}_{2,2}$ are derived from $\mathbf{B}_1, \mathbf{B}_2$ as in Eq. (4.2). When $\mathbf{V} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times \ell}$, this corresponds to the distribution in Hyb_1 and when $\mathbf{V} = \mathbf{S}\mathbf{A}$ where $\mathbf{S} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times k}$ and $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$, this corresponds to the distribution in Hyb_0.
- Hybrids Hyb_1 and Hyb_2 are computationally indistinguishable under $\mathsf{MDDH}_{k,\ell,2k}$ by an analogous argument.

Since $\ell = \mathrm{poly}(\lambda)$, the k-Lin assumption in $\mathbb{G}_2$ implies the $\mathsf{MDDH}_{k,\ell,2k}$ assumption in $\mathbb{G}_2$. The theorem now follows by a hybrid argument. □

Theorem 4.11 (Type-II Indistinguishability)**.** *If the bilateral* k-Lin *assumption holds with respect to* $\mathsf{GroupGen}$*, then Construction 4.8 satisfies Type-II indistinguishability.*

Proof. Let $\mathcal{A}$ be an adversary and let ℓ, j_1, j_2, j_2' be the values it chooses. We proceed via a hybrid argument:

- Hyb_0: This is $\mathsf{ExptTII}_{\mathcal{A}}[\lambda, 0]$. In this experiment, the challenger samples $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e) \leftarrow \mathsf{GroupGen}(1^\lambda)$. It samples $\hat{\mathbf{B}} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times 2k}$ and defines $\hat{\mathbf{B}}^* = \hat{\mathbf{B}}^{-1}$. Then it parses $\hat{\mathbf{B}}$ into matrices $\hat{\mathbf{B}}_1, \hat{\mathbf{B}}_2 \in \mathbb{Z}_p^{k \times 2k}$ according to Eq. (4.2) and $\hat{\mathbf{B}}^*$ into matrices $\hat{\mathbf{B}}_1^*, \hat{\mathbf{B}}_2^* \in \mathbb{Z}_p^{2k \times k}$ according to Eq. (4.3). It also samples $\mathbf{B}_{1,1}^*, \mathbf{B}_{1,2}^*, \mathbf{B}_{2,1}^*, \mathbf{B}_{2,2}^* \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times 2k}$ and $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2, \mathbf{S}_{1,1}, \mathbf{S}_{1,2}, \mathbf{S}_{2,1}, \mathbf{S}_{2,2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$. Finally, it sets
$$\hat{\mathbf{T}} = \hat{\mathbf{B}}_1^* \hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^* \hat{\mathbf{S}}_2 \mathbf{P}_{j_1} \quad \text{and} \quad \mathbf{T}_1 = \mathbf{B}_{1,1}^* \mathbf{S}_{1,1} + \mathbf{B}_{1,2}^* \mathbf{S}_{1,2} \mathbf{P}_{j_2} \quad \text{and} \quad \mathbf{T}_2 = \mathbf{B}_{2,1}^* \mathbf{S}_{2,1} + \mathbf{B}_{2,2}^* \mathbf{S}_{2,2} \mathbf{P}_{j_2}.$$
Finally, it computes $\mathbf{T}_* = \mathbf{T}_1 \otimes \mathbf{T}_2$ and outputs
$$\mathsf{crs}_{\mathsf{base}} = \left(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\right)$$
along with $\mathsf{td}_1 = \hat{\mathbf{B}}_2$.

- Hyb_1: Same as Hyb_0 except the challenger samples $\mathbf{T}_1 \xleftarrow{R} \mathbb{Z}_p^{2k\times\ell}$.
- Hyb_2: Same as Hyb_1 except the challenger samples $\mathbf{T}_2 \xleftarrow{R} \mathbb{Z}_p^{2k\times\ell}$.
- Hyb_3: Same as Hyb_2 except the challenger sets $\mathbf{T}_2 = \mathbf{B}^*_{2,1}\mathbf{S}_{2,1} + \mathbf{B}^*_{2,2}\mathbf{S}_{2,2}\mathbf{P}_{j_2'}$
- Hyb_4: Same as Hyb_3 except the challenger sets $\mathbf{T}_1 = \mathbf{B}^*_{1,1}\mathbf{S}_{1,1} + \mathbf{B}^*_{1,2}\mathbf{S}_{1,2}\mathbf{P}_{j_2'}$. This is $\mathsf{ExptTII}_{\mathcal{A}}[\lambda, 1]$

We now show that each adjacent pair of hybrid experiments is computationally indistinguishable. As before, we implicitly use the fact that sampling $\mathbf{A} \xleftarrow{R} \mathbb{Z}_p^{2k\times 2k}$ is statistically indistinguishable from sampling a *full rank* $\mathbf{A} \xleftarrow{R} \mathbb{Z}_p^{2k\times 2k}$.

- Hybrids Hyb_0 and Hyb_1 are computationally indistinguishable under the bilateral $\mathsf{MDDH}_{k,\ell,2k}$ assumption. Specifically, given the bilateral $\mathsf{MDDH}_{k,\ell,2k}$ challenge $(\mathcal{G}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{V}]_1, [\mathbf{V}]_2)$, the reduction algorithm samples $\hat{\mathbf{B}} \xleftarrow{R} \mathbb{Z}_p^{2k\times 2k}$ and defines $\hat{\mathbf{B}}^* = \hat{\mathbf{B}}^{-1}$. Then it parses $\hat{\mathbf{B}}$ into matrices $\hat{\mathbf{B}}_1, \hat{\mathbf{B}}_2 \in \mathbb{Z}_p^{k\times 2k}$ according to Eq. (4.2) and $\hat{\mathbf{B}}^*$ into matrices $\hat{\mathbf{B}}^*_1, \hat{\mathbf{B}}^*_2 \in \mathbb{Z}_p^{2k\times k}$ according to Eq. (4.3). It also samples $\mathbf{B}^*_{1,2}, \mathbf{B}^*_{2,1}, \mathbf{B}^*_{2,2} \xleftarrow{R} \mathbb{Z}_p^{k\times 2k}$ and $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2, \mathbf{S}_{1,2}, \mathbf{S}_{2,1}, \mathbf{S}_{2,2} \xleftarrow{R} \mathbb{Z}_p^{k\times\ell}$. Next, it sets
$$\hat{\mathbf{T}} = \hat{\mathbf{B}}^*_1\hat{\mathbf{S}}_1 + \hat{\mathbf{B}}^*_2\hat{\mathbf{S}}_2\mathbf{P}_{j_1} \quad \text{and} \quad \mathbf{T}_2 = \mathbf{B}^*_{2,1}\mathbf{S}_{2,1} + \mathbf{B}^*_{2,2}\mathbf{S}_{2,2}\mathbf{P}_{j_2}.$$
It gives $\mathcal{A}$ the CRS
$$\mathsf{crs}_{\mathsf{base}} = \left(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{V}]_1 + \mathbf{B}^*_{1,2}\mathbf{S}_{1,2}\mathbf{P}_{j_2}, [\mathbf{V}]_2 + \mathbf{B}^*_{1,2}\mathbf{S}_{1,2}\mathbf{P}_{j_2}, [\mathbf{T}_2]_2, \left([\mathbf{V}]_2 + \mathbf{B}^*_{1,2}\mathbf{S}_{1,2}\mathbf{P}_{j_2}\right) \otimes \mathbf{T}_2\right)$$
and the projection trapdoor $\mathsf{td}_1 = \hat{\mathbf{B}}_2$. When $\mathbf{V} \xleftarrow{R} \mathbb{Z}_p^{2k\times\ell}$, this corresponds to the distribution in Hyb_1 and when $\mathbf{V} = \mathbf{S}\mathbf{A}$ where $\mathbf{S} \xleftarrow{R} \mathbb{Z}_p^{2k\times k}$ and $\mathbf{A} \xleftarrow{R} \mathbb{Z}_p^{k\times\ell}$, this corresponds to the distribution in Hyb_0.
- Hybrids Hyb_1 and Hyb_2 are computationally indistinguishable under the $\mathsf{MDDH}_{k,\ell,2k}$ assumption in $\mathbb{G}_2$. Specifically, given the $\mathsf{MDDH}_{k,\ell,2k}$ challenge $(\mathcal{G}, [\mathbf{A}]_2, [\mathbf{V}]_2)$, the reduction algorithm samples $\hat{\mathbf{B}} \xleftarrow{R} \mathbb{Z}_p^{2k\times 2k}$ and defines $\hat{\mathbf{B}}^* = \hat{\mathbf{B}}^{-1}$. Then it parses $\hat{\mathbf{B}}$ into matrices $\hat{\mathbf{B}}_1, \hat{\mathbf{B}}_2 \in \mathbb{Z}_p^{k\times 2k}$ according to Eq. (4.2) and $\hat{\mathbf{B}}^*$ into matrices $\hat{\mathbf{B}}^*_1, \hat{\mathbf{B}}^*_2 \in \mathbb{Z}_p^{2k\times k}$ according to Eq. (4.3). It also samples $\mathbf{B}^*_{2,2} \xleftarrow{R} \mathbb{Z}_p^{k\times 2k}$ and $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2, \mathbf{S}_{2,2} \xleftarrow{R} \mathbb{Z}_p^{k\times\ell}$. Next, it sets
$$\hat{\mathbf{T}} = \hat{\mathbf{B}}^*_1\hat{\mathbf{S}}_1 + \hat{\mathbf{B}}^*_2\hat{\mathbf{S}}_2\mathbf{P}_{j_1} \quad \text{and} \quad \mathbf{T}_1 \xleftarrow{R} \mathbb{Z}_p^{2k\times\ell}.$$
It gives $\mathcal{A}$ the CRS
$$\mathsf{crs}_{\mathsf{base}} = \left(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{V}]_2 + \mathbf{B}^*_{2,2}\mathbf{S}_{2,2}\mathbf{P}_{j_2}, \left(\mathbf{T}_1 \otimes \left([\mathbf{V}]_2 + \mathbf{B}^*_{2,2}\mathbf{S}_{2,2}\mathbf{P}_{j_2}\right)\right)\right)$$
and the projection trapdoor $\mathsf{td}_1 = \hat{\mathbf{B}}_2$. When $\mathbf{V} \xleftarrow{R} \mathbb{Z}_p^{2k\times\ell}$, this corresponds to the distribution in Hyb_2 and when $\mathbf{V} = \mathbf{S}\mathbf{A}$ where $\mathbf{S} \xleftarrow{R} \mathbb{Z}_p^{2k\times k}$ and $\mathbf{A} \xleftarrow{R} \mathbb{Z}_p^{k\times\ell}$, this corresponds to the distribution in Hyb_1.
- Hybrids Hyb_2 and Hyb_3 are computationally indistinguishable under the $\mathsf{MDDH}_{k,\ell,2k}$ assumption in $\mathbb{G}_2$. This follows by an analogous argument as that used to argue indistinguishability of Hyb_1 and Hyb_2.
- Hybrids Hyb_3 and Hyb_4 are computationally indistinguishable under the bilateral $\mathsf{MDDH}_{k,\ell,2k}$ assumption. This follows by an analogous argument as that used to argue indistinguishability of Hyb_0 and Hyb_1.

Since $\ell = \mathsf{poly}(\lambda)$, the bilateral k-Lin assumption implies the bilateral $\mathsf{MDDH}_{k,\ell,2k}$ assumption (Remark 3.8). The theorem now follows by a hybrid argument. □

Theorem 4.12 (Type-II Collision Resistance). *Suppose the k-KerLin assumption holds in $\mathbb{G}_2$ with respect to $\mathsf{GroupGen}$. Then, Construction 4.8 satisfies Type-II collision resistance.*

Proof. Take any adversary $\mathcal{A}$ that breaks the Type-II collision resistance of Construction 4.8 with non-negligible probability ε. Let ℓ and j_1 be the input length and Type-I index chosen by $\mathcal{A}$. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ that breaks the $\mathsf{KerDH}_{k,\ell}$ assumption in $\mathbb{G}_2$ with respect to $\mathsf{GroupGen}$:

1. On input the $\mathsf{KerDH}_{k,\ell}$ challenge $(\mathcal{G}, [\mathbf{A}]_2)$, algorithm $\mathcal{B}$ samples full-rank matrices $\hat{\mathbf{B}}, \mathbf{B}_1, \mathbf{B}_2 \xleftarrow{R} \mathbb{Z}_p^{2k \times 2k}$ and defines $\hat{\mathbf{B}}^* = \hat{\mathbf{B}}^{-1}$, $\mathbf{B}_1^* = \mathbf{B}_1^{-1}$, and $\mathbf{B}_2^* = \mathbf{B}_2^{-1}$. Then it samples $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2, \mathbf{S}_{1,1}, \mathbf{S}_{1,2}, \mathbf{S}_{2,1}, \xleftarrow{R} \mathbb{Z}_p^{k \times \ell}$ and constructs

$$\hat{\mathbf{T}} = \hat{\mathbf{B}}_1^* \hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^* \hat{\mathbf{S}}_2 \mathbf{P}_{j_1} \quad \text{and} \quad \mathbf{T}_1 = \mathbf{B}_{1,1}^* \mathbf{S}_{1,1} + \mathbf{B}_{1,2}^* \mathbf{S}_{1,2} \quad \text{and} \quad [\mathbf{T}_2]_2 = \mathbf{B}_{2,1}^* \mathbf{S}_{2,1} + \mathbf{B}_{2,2}^* [\mathbf{A}]_2,$$

where the components $\hat{\mathbf{B}}_1^*, \hat{\mathbf{B}}_2^*, \mathbf{B}_{1,1}^*, \mathbf{B}_{1,2}^*, \mathbf{B}_{2,1}^*, \mathbf{B}_{2,2}^*$ are obtained from $\hat{\mathbf{B}}^*, \mathbf{B}_1^*, \mathbf{B}_2^*$ according to Eq. (4.3). Algorithm $\mathcal{B}$ gives $\mathsf{crs}_{\mathsf{base}}$ to $\mathcal{A}$ where

$$\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, \mathbf{T}_1 \otimes [\mathbf{T}_2]_2\big).$$

2. At the end of the game, algorithm $\mathcal{A}$ outputs two vectors $\mathbf{y}, \mathbf{y}' \in \mathbb{Z}_p^\ell$. Algorithm $\mathcal{B}$ outputs $[\mathbf{y} - \mathbf{y}']_1$.

Since the KerDH challenger samples $\mathbf{A} \xleftarrow{R} \mathbb{Z}_p^{k \times \ell}$ and $\mathbf{P}_\ell = \mathbf{I}_\ell$, algorithm $\mathcal{B}$ perfectly simulates an execution of the Type-II collision resistance game for $\mathcal{A}$. Thus, with probability at least ε, algorithm $\mathcal{A}$ outputs $\mathbf{y} \neq \mathbf{y}'$ such that $\mathbf{B}_{2,2}\mathbf{T}_2\mathbf{y} = \mathbf{B}_{2,2}\mathbf{T}_2\mathbf{y}'$ (and $\mathbf{B}_{1,2}\mathbf{T}_1\mathbf{y} = \mathbf{B}_{1,2}\mathbf{T}_1\mathbf{y}'$). This means that

$$\begin{aligned} \mathbf{B}_{2,2}\mathbf{T}_2\mathbf{y} &= \mathbf{B}_{2,2}(\mathbf{B}_{2,1}^*\mathbf{S}_{2,1} + \mathbf{B}_{2,2}^*\mathbf{A})\mathbf{y} = \mathbf{A}\mathbf{y} \\ \mathbf{B}_{2,2}\mathbf{T}_2\mathbf{y}' &= \mathbf{B}_{2,2}(\mathbf{B}_{2,1}^*\mathbf{S}_{2,1} + \mathbf{B}_{2,2}^*\mathbf{A})\mathbf{y}' = \mathbf{A}\mathbf{y}' \end{aligned}$$

We conclude that $\mathbf{A}\mathbf{y} = \mathbf{A}\mathbf{y}'$, so $\mathbf{A}(\mathbf{y} - \mathbf{y}') = \mathbf{0}$, but $\mathbf{y} \neq \mathbf{y}'$. Correspondingly, algorithm $\mathcal{B}$ breaks $\mathsf{KerDH}_{k,\ell}$ with advantage ε. Finally, since $\ell = \mathsf{poly}(\lambda)$, the $\mathsf{KerDH}_{k,\ell}$ assumption follows from k-KerLin, as required. □

4.2 Prefix Checking on Committed Values

The first proof system we design for the base projective commitment scheme in Section 4.1 is to argue that two Type-I commitments share a common prefix (i.e., that σ_1, σ_1' are commitments to $\mathbf{x}$ and $\mathbf{x}'$ where $x_i = x_i'$ for all $i \leq j$). In the broader context of constructing functional commitments (Section 5), the prefix-checking proof system is used to check consistency between a commitment to an input $\mathbf{x}$ and a commitment to *all* of the wires in an arithmetic circuit evaluation $C(\mathbf{x})$. The security requirement is enforced in the *semi-functional space*. We start by defining the syntax of the prefix-checking proof system as well as its correctness and security requirements:

Definition 4.13 (Prefix Checking for Projective Commitments)**.** Let $\mathsf{FC}_{\mathsf{base}} = \big(\mathsf{SetupBase}, \mathsf{SetupSF}, \mathsf{Commit}^{(1)}, \mathsf{Commit}^{(2)}, \mathsf{Project}^{(1)}, \mathsf{Project}^{(2)}\big)$ be a projective commitment scheme. A prefix-checking proof system for $\mathsf{FC}_{\mathsf{base}}$ is a triple of efficient algorithms $\mathsf{FC}_{\mathsf{pre}} = \big(\mathsf{SetupPre}, \mathsf{OpenPre}, \mathsf{VerifyPre}\big)$ with the following properties:

- $\mathsf{SetupPre}(\mathsf{crs}_{\mathsf{base}}, j) \to \mathsf{crs}$: On input the common reference string $\mathsf{crs}_{\mathsf{base}}$ (defining the associated input space $\mathcal{R}^\ell$) and a prefix length $j \in [\ell]$, the setup algorithm outputs a common reference string crs.
- $\mathsf{OpenPre}(\mathsf{crs}, \mathbf{x}, \mathbf{x}') \to \pi$: On input a common reference string crs and two vectors $\mathbf{x}, \mathbf{x}' \in \mathcal{R}^\ell$, the opening algorithm outputs a proof π.
- $\mathsf{VerifyPre}(\mathsf{crs}, \sigma_1, \sigma_1', \pi) \to b$: On input the common reference string crs, two Type-I commitments σ_1, σ_1', and an opening π, the verification algorithm outputs a bit $b \in \{0, 1\}$.

The prefix-checking proof system $\mathsf{FC}_{\mathsf{pre}}$ should satisfy the following two properties:

- **Correctness:** For all security parameters $\lambda \in \mathbb{N}$, all vector lengths $\ell \in \mathbb{N}$, all prefix lengths $j \in [\ell]$, all $\mathsf{crs}_{\mathsf{base}}$ in the support of $\mathsf{SetupBase}(1^\lambda, 1^\ell)$, all vectors $\mathbf{x}, \mathbf{x}' \in \mathcal{R}^\ell$ (where $\mathcal{R}^\ell$ is the message space associated with $\mathsf{crs}_{\mathsf{base}}$) where $x_i = x_i'$ for all $i \leq j$,

$$\Pr\left[\mathsf{VerifyPre}(\mathsf{crs}, \sigma_1, \sigma_1', \pi) = 1 : \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{SetupPre}(\mathsf{crs}_{\mathsf{base}}, j) \\ \sigma_1 \leftarrow \mathsf{Commit}^{(1)}(\mathsf{crs}_{\mathsf{base}}, \mathbf{x}) \\ \sigma_1' \leftarrow \mathsf{Commit}^{(1)}(\mathsf{crs}_{\mathsf{base}}, \mathbf{x}') \\ \pi \leftarrow \mathsf{OpenPre}(\mathsf{crs}, \mathbf{x}, \mathbf{x}') \end{array}\right] = 1.$$

- **Prefix-matching security:** For a security parameter λ and an adversary $\mathcal{A}$, we define the prefix-matching security game as follows:
  1. On input the security parameter λ, the adversary outputs the dimension 1^ℓ and the prefix length $j \in [\ell]$.
  2. The challenger samples $(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^\lambda, 1^\ell, j, j)$ and $\mathsf{crs} \leftarrow \mathsf{SetupPre}(\mathsf{crs}_{\mathsf{base}}, j)$. It gives $(\mathsf{crs}_{\mathsf{base}}, \mathsf{crs})$ to $\mathcal{A}$.
  3. The adversary outputs two Type-I commitments (σ_1, σ_1') and an opening π.
  4. The output of the experiment is $b = 1$ if the following properties hold:
     - **Mismatching prefix:** $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) \neq \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$.
     - **Validity of opening:** $\mathsf{VerifyPre}(\mathsf{crs}, \sigma_1, \sigma_1', \pi) = 1$.

     Otherwise, the challenger outputs $b = 0$.

  We say that that $\mathsf{FC}_{\mathsf{pre}}$ satisfies prefix-matching security if for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that $\Pr[b = 1] = \mathsf{negl}(\lambda)$ in the prefix-matching security game.

Constructing a prefix-checking proof system. We now show how to construct a prefix-checking proof system for the base projective commitment scheme from Section 4 (Construction 4.8).

Construction 4.14 (Prefix Checking for Projective Commitments)**.** Let $\mathsf{FC}_{\mathsf{base}} = \big(\mathsf{SetupBase}, \mathsf{SetupSF}, \mathsf{Commit}^{(1)}, \mathsf{Commit}^{(2)}, \mathsf{Project}^{(1)}, \mathsf{Project}^{(2)}\big)$ be the projective commitment scheme from Construction 4.8. We construct a prefix-checking proof system $\mathsf{FC}_{\mathsf{pre}} = \big(\mathsf{SetupPre}, \mathsf{OpenPre}, \mathsf{VerifyPre}\big)$ for $\mathsf{FC}_{\mathsf{base}}$ as follows:

- $\mathsf{SetupPre}(\mathsf{crs}_{\mathsf{base}}, j)$: On input the common reference string $\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big)$ for the base projective commitment scheme, and a prefix length $j \in [\ell]$, the setup algorithm samples $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times (k+1)}$ and $\mathbf{W} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(k+1) \times 2k}$. Then, it computes
$$[\mathbf{Z}]_2 = \mathbf{W}[\hat{\mathbf{T}}]_2 \begin{bmatrix} \mathbf{0}^{j \times (\ell - j)} \\ \mathbf{I}_{\ell - j} \end{bmatrix} \in \mathbb{G}_2^{(k+1) \times (\ell - j)}, \tag{4.4}$$
  Output the common reference string
$$\mathsf{crs} = (\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, [\mathbf{AW}]_1, [\mathbf{Z}]_2)\,. \tag{4.5}$$
- $\mathsf{OpenPre}(\mathsf{crs}, \mathbf{x}, \mathbf{x}')$: On input the common reference string $\mathsf{crs} = (\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, [\mathbf{AW}]_1, [\mathbf{Z}]_2)$ and two vectors $\mathbf{x}, \mathbf{x}' \in \mathbb{Z}_p^\ell$, the opening algorithm computes and outputs
$$\pi = [\mathbf{v}]_2 = [\mathbf{Z}]_2 \cdot [\mathbf{0}^{(\ell - j) \times j} \mid \mathbf{I}_{\ell - j}](\mathbf{x} - \mathbf{x}') \in \mathbb{G}_2^{k+1}.$$
- $\mathsf{VerifyPre}(\mathsf{crs}, \sigma_1, \sigma_1', \pi)$: On input the common reference string $\mathsf{crs} = (\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, [\mathbf{AW}]_1, [\mathbf{Z}]_2)$, two Type-I commitments $\sigma_1 = [\hat{\mathbf{c}}]_2$, $\sigma_1' = [\hat{\mathbf{c}}']_2$, and an opening $\pi = [\mathbf{v}]_2$, the verification algorithm outputs 1 if
$$[\mathbf{AW}]_1([\hat{\mathbf{c}}]_2 - [\hat{\mathbf{c}}']_2) = [\mathbf{A}]_1[\mathbf{v}]_2.$$

Theorem 4.15 (Correctness)**.** *Construction 4.14 is correct.*

Proof. Take any $\lambda, \ell \in \mathbb{N}$ and $j \in [\ell]$. Let $\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big) \leftarrow \mathsf{SetupBase}(1^\lambda, 1^\ell)$. Let $\mathsf{crs} = (\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, [\mathbf{AW}]_1, [\mathbf{Z}]_2) \leftarrow \mathsf{SetupPre}(\mathsf{crs}_{\mathsf{base}}, j)$. Take any two vectors $\mathbf{x}, \mathbf{x}' \in \mathbb{Z}_p^\ell$ with a common prefix of length j. This means that
$$\begin{bmatrix} \mathbf{0}^{j \times (\ell - j)} \\ \mathbf{I}_{\ell - j} \end{bmatrix} [\mathbf{0}^{(\ell - j) \times j} \mid \mathbf{I}_{\ell - j}](\mathbf{x} - \mathbf{x}') = \mathbf{x} - \mathbf{x}'.$$
Suppose $\sigma_1 \leftarrow \mathsf{Commit}^{(1)}(\mathsf{crs}_{\mathsf{base}}, \mathbf{x})$ and $\sigma_1' \leftarrow \mathsf{Commit}^{(1)}(\mathsf{crs}_{\mathsf{base}}, \mathbf{x}')$, and $\pi \leftarrow \mathsf{OpenPre}(\mathsf{crs}, \mathbf{x}, \mathbf{x}')$. By construction, $\sigma_1 = [\hat{\mathbf{c}}]_2 = [\hat{\mathbf{T}}\mathbf{x}]_2$, $\sigma_1' = [\hat{\mathbf{c}}']_2 = [\hat{\mathbf{T}}\mathbf{x}']_2$, and $\pi = [\mathbf{v}]_2$ where
$$\mathbf{Av} = \mathbf{AZ}[\mathbf{0}^{(\ell - j) \times j} \mid \mathbf{I}_{\ell - j}](\mathbf{x} - \mathbf{x}') = \mathbf{AW}\hat{\mathbf{T}} \begin{bmatrix} \mathbf{0}^{j \times (\ell - j)} \\ \mathbf{I}_{\ell - j} \end{bmatrix} [\mathbf{0}^{(\ell - j) \times j} \mid \mathbf{I}_{\ell - j}](\mathbf{x} - \mathbf{x}') = \mathbf{AW}\hat{\mathbf{T}}(\mathbf{x} - \mathbf{x}') = \mathbf{AW}(\hat{\mathbf{c}} - \hat{\mathbf{c}}'). \quad \square$$

Theorem 4.16 (Prefix-Matching Security)**.** *Suppose the* $\mathsf{KerLin}_{k,k+1}$ *assumption holds in* $\mathbb{G}_1$ *with respect to* $\mathsf{GroupGen}$. *Then, Construction 4.14 satisfies prefix-matching security.*

Proof. Take any efficient adversary $\mathcal{A}$ for the prefix-matching security game. We start by defining a sequence of hybrid experiments.

- Hyb_0: This is the prefix-checking security experiment. We provide the full specification here:
  - At the beginning of the game, the adversary $\mathcal{A}$ outputs 1^ℓ and $j \in [\ell]$.
  - The challenger samples $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e) \leftarrow \mathsf{GroupGen}(1^\lambda)$. It samples full-rank matrices $\hat{\mathbf{B}}, \mathbf{B}_1, \mathbf{B}_2 \xleftarrow{\textsc{r}} \mathbb{Z}_p^{2k \times 2k}$ and defines $\hat{\mathbf{B}}^* = \hat{\mathbf{B}}^{-1}$, $\mathbf{B}_1^* = \mathbf{B}_1^{-1}$, and $\mathbf{B}_2^* = \mathbf{B}_2^{-1}$. It parses $\hat{\mathbf{B}}, \mathbf{B}_1, \mathbf{B}_2$ as in Eq. (4.2) and $\hat{\mathbf{B}}^*, \mathbf{B}_1^*, \mathbf{B}_2^*$ as in Eq. (4.3).
  - The challenger constructs the encoding matrices $\hat{\mathbf{T}}, \mathbf{T}_1, \mathbf{T}_2$ as follows:
    - **Type-I encodings:** Sample $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2 \xleftarrow{\textsc{r}} \mathbb{Z}_p^{k \times \ell}$ and let $\hat{\mathbf{T}} = \hat{\mathbf{B}}_1^* \hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^* \hat{\mathbf{S}}_2 \mathbf{P}_j \in \mathbb{Z}_p^{2k \times \ell}$.
    - **Type-II encodings:** For $\alpha \in \{1, 2\}$, sample $\mathbf{S}_{\alpha,1}, \mathbf{S}_{\alpha,2} \xleftarrow{\textsc{r}} \mathbb{Z}_p^{k \times \ell}$. Let $\mathbf{T}_\alpha = \mathbf{B}_{\alpha,1}^* \mathbf{S}_{\alpha,1} + \mathbf{B}_{\alpha,2}^* \mathbf{S}_{\alpha,2} \mathbf{P}_j \in \mathbb{Z}_p^{2k \times \ell}$.

    Finally, the challenger sets $\mathbf{T}_* = \mathbf{T}_1 \otimes \mathbf{T}_2$ and sets $\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big)$.
  - The challenger samples $\mathbf{A} \xleftarrow{\textsc{r}} \mathbb{Z}_p^{k \times (k+1)}$ and $\mathbf{W} \xleftarrow{\textsc{r}} \mathbb{Z}_p^{(k+1) \times 2k}$. It computes

    $$\mathbf{Z} = \mathbf{W} \hat{\mathbf{T}} \begin{bmatrix} \mathbf{0}^{j \times (\ell - j)} \\ \mathbf{I}_{\ell - j} \end{bmatrix}.$$

    The challenger gives the common reference string crs to $\mathcal{A}$ where

    $$\mathsf{crs} = (\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, [\mathbf{A}\mathbf{W}]_1, [\mathbf{Z}]_2).$$
  - The adversary outputs two commitments $\sigma_1 = [\hat{\mathbf{c}}]_2$, $\sigma_1' = [\hat{\mathbf{c}}']_2$ and an opening $\pi = [\mathbf{v}]_2$.

  The output of the experiment is 1 if $\hat{\mathbf{B}}_2 \hat{\mathbf{c}} \neq \hat{\mathbf{B}}_2 \hat{\mathbf{c}}'$ (i.e., $\hat{\mathbf{B}}_2(\hat{\mathbf{c}} - \hat{\mathbf{c}}') \neq \mathbf{0}$) and $\mathbf{A}\mathbf{W}(\hat{\mathbf{c}} - \hat{\mathbf{c}}') = \mathbf{A}\mathbf{v}$.
- Hyb_1: Same as Hyb_0, except the challenger outputs 1 if $\mathbf{W}(\hat{\mathbf{c}} - \hat{\mathbf{c}}') = \mathbf{v}$ and $\hat{\mathbf{B}}_2(\hat{\mathbf{c}} - \hat{\mathbf{c}}') \neq \mathbf{0}$.
- Hyb_2: Same as Hyb_1, except when constructing the CRS, the challenger samples a random nonzero vector $\mathbf{a}^\perp \in \mathbb{Z}_p^{k+1}$ in the kernel of $\mathbf{A}$. Then, it samples $\mathbf{W}_{\mathsf{norm}} \xleftarrow{\textsc{r}} \mathbb{Z}_p^{(k+1) \times k}$, $\mathbf{W}_{\mathsf{sf},1} \xleftarrow{\textsc{r}} \mathbb{Z}_p^{(k+1) \times k}$, $\mathbf{w}_{\mathsf{sf},2} \xleftarrow{\textsc{r}} \mathbb{Z}_p^k$. It sets

  $$\mathbf{W}_{\mathsf{sf}} = \mathbf{W}_{\mathsf{sf},1} + \mathbf{a}^\perp \mathbf{w}_{\mathsf{sf},2}^{\mathsf{T}} \quad \text{and} \quad \mathbf{W} = \mathbf{W}_{\mathsf{norm}} \hat{\mathbf{B}}_1 + \mathbf{W}_{\mathsf{sf}} \hat{\mathbf{B}}_2.$$

  The challenger then sets $\mathbf{Z}$ as

  $$\mathbf{Z} = \mathbf{W}_{\mathsf{norm}} \hat{\mathbf{S}}_1 \begin{bmatrix} \mathbf{0}^{j \times (\ell - j)} \\ \mathbf{I}_{\ell - j} \end{bmatrix}.$$

  Finally, the challenger sets the CRS to be

  $$\mathsf{crs} = \Big(\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, \big[\mathbf{A}\big(\mathbf{W}_{\mathsf{norm}} \hat{\mathbf{B}}_1 + \mathbf{W}_{\mathsf{sf},1} \hat{\mathbf{B}}_2\big)\big]_1, [\mathbf{Z}]_2\Big).$$

We write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output distribution of an execution of hybrid Hyb_i with adversary $\mathcal{A}$. We now show that the output distribution of each pair of hybrids is indistinguishable.

Lemma 4.17. *Suppose the* $\mathsf{KerDH}_{k,k+1}$ *assumption holds in* $\mathbb{G}_1$ *with respect to* $\mathsf{GroupGen}$. *Then, it follows that* $\big|\Pr[\mathsf{Hyb}_0(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_1(\mathcal{A}) = 1]\big| = \mathsf{negl}(\lambda)$.

Proof. Suppose $|\Pr[\mathsf{Hyb}_0(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_1(\mathcal{A}) = 1]| \geq \varepsilon$ for some non-negligible ε. The only difference between Hyb_0 and Hyb_1 is the verification relation. Let $[\hat{\mathbf{c}}]_2, [\hat{\mathbf{c}}']_2, [\mathbf{v}]_2$ be the output of $\mathcal{A}$ in an execution of Hyb_0 or Hyb_1. If the outputs of Hyb_0 and Hyb_1 differ, then it must be the case that

$$\mathbf{A}\mathbf{W}(\hat{\mathbf{c}} - \hat{\mathbf{c}}') = \mathbf{A}\mathbf{v} \quad \text{and} \quad \mathbf{W}(\hat{\mathbf{c}} - \hat{\mathbf{c}}') \neq \mathbf{v}. \tag{4.6}$$

In all other cases, the output in Hyb_0 and Hyb_1 is identical. We use $\mathcal{A}$ to construct an efficient adversary $\mathcal{B}$ for $\mathsf{KerDH}_{k,k+1}$:

1. On input the KerDH challenge $(\mathcal{G}, [\mathbf{A}]_1)$, algorithm $\mathcal{B}$ starts by running algorithm $\mathcal{A}$. Algorithm $\mathcal{A}$ outputs the input dimension 1^ℓ and $j \in [\ell]$.
2. Next, algorithm $\mathcal{B}$ samples full-rank matrices $\hat{\mathbf{B}}, \mathbf{B}_1, \mathbf{B}_2 \xleftarrow{\text{R}} \mathbb{Z}_p^{2k \times 2k}$ and defines $\hat{\mathbf{B}}^* = \hat{\mathbf{B}}^{-1}$, $\mathbf{B}_1^* = \mathbf{B}_1^{-1}$, and $\mathbf{B}_2^* = \mathbf{B}_2^{-1}$. It parses the components of $\hat{\mathbf{B}}, \mathbf{B}_1, \mathbf{B}_2$ as in Eq. (4.2) and $\mathbf{B}_1^*, \mathbf{B}_2^*, \hat{\mathbf{B}}^*$ as in Eq. (4.3).
3. Algorithm $\mathcal{B}$ then constructs the encoding matrices $\hat{\mathbf{T}}, \mathbf{T}_1, \mathbf{T}_2$ as in Hyb_0 and Hyb_1:
   - **Type-I encodings:** Sample $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2 \xleftarrow{\text{R}} \mathbb{Z}_p^{k \times \ell}$ and let $\hat{\mathbf{T}} = \hat{\mathbf{B}}_1^* \hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^* \hat{\mathbf{S}}_2 \mathbf{P}_j \in \mathbb{Z}_p^{2k \times \ell}$.
   - **Type-II encodings:** For $\alpha \in \{1, 2\}$, sample $\mathbf{S}_{\alpha,1}, \mathbf{S}_{\alpha,2} \xleftarrow{\text{R}} \mathbb{Z}_p^{k \times \ell}$ and let $\mathbf{T}_\alpha = \mathbf{B}_{\alpha,1}^* \mathbf{S}_{\alpha,1} + \mathbf{B}_{\alpha,2}^* \mathbf{S}_{\alpha,2} \mathbf{P}_j$.

   Algorithm $\mathcal{B}$ computes $\mathbf{T}_* = \mathbf{T}_1 \otimes \mathbf{T}_2$ and sets $\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big)$.
4. Algorithm $\mathcal{B}$ samples $\mathbf{W} \xleftarrow{\text{R}} \mathbb{Z}_p^{(k+1) \times 2k}$ and computes

   $$\mathbf{Z} = \mathbf{W}\hat{\mathbf{T}} \begin{bmatrix} \mathbf{0}^{j \times (\ell - j)} \\ \mathbf{I}_{\ell - j} \end{bmatrix}.$$

   The challenger gives the common reference string crs to $\mathcal{A}$ where

   $$\mathsf{crs} = (\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, [\mathbf{A}]_1 \mathbf{W}, [\mathbf{Z}]_2) = (\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, [\mathbf{A}\mathbf{W}]_1, [\mathbf{Z}]_2).$$
5. Algorithm $\mathcal{A}$ outputs commitments $\sigma_1 = [\hat{\mathbf{c}}]_2$, $\sigma_1' = [\hat{\mathbf{c}}']_2$ and an opening $\pi = [\mathbf{v}]_2$. Algorithm $\mathcal{B}$ outputs $\mathbf{W}([\hat{\mathbf{c}}]_2 - [\hat{\mathbf{c}}']_2) - [\mathbf{v}]_2$.

Since the KerDH challenger samples $\mathbf{A} \xleftarrow{\text{R}} \mathbb{Z}_p^{(k+1) \times k}$, the common reference string crs constructed by $\mathcal{B}$ is distributed exactly as required in Hyb_0 and Hyb_1. By the above analysis, this means that with probability at least ε, algorithm $\mathcal{A}$ outputs $[\hat{\mathbf{c}}]_2$, $[\hat{\mathbf{c}}']_2$, and $[\mathbf{v}]_2$ such that Eq. (4.6) holds. This means $\mathbf{A}\big(\mathbf{W}(\hat{\mathbf{c}} - \hat{\mathbf{c}}') - \mathbf{v}\big) = \mathbf{0}$ but $\mathbf{W}(\hat{\mathbf{c}} - \hat{\mathbf{c}}') - \mathbf{v} \neq \mathbf{0}$. Correspondingly, algorithm $\mathcal{B}$ breaks the KerDH assumption with the same advantage ε. ◻

Lemma 4.18. $\Pr[\mathsf{Hyb}_1(\mathcal{A}) = 1] = \Pr[\mathsf{Hyb}_2(\mathcal{A}) = 1]$.

Proof. Consider the distribution of $\mathbf{W}$ in Hyb_2. In Hyb_2, both $\mathbf{W}_{\mathsf{norm}}$ and $\mathbf{W}_{\mathsf{sf}}$ are sampled uniformly at random from $\mathbb{Z}_p^{(k+1) \times k}$. Since $\hat{\mathbf{B}} = [\hat{\mathbf{B}}_1 \mid \hat{\mathbf{B}}_2]$ is a basis for $\mathbb{Z}_p^{2k}$, the distribution of $\mathbf{W}$ is uniform over $\mathbb{Z}_p^{(k+1) \times 2k}$, which matches the distribution in Hyb_1. Next,

$$\begin{aligned} \mathbf{W}\hat{\mathbf{T}} &= \big(\mathbf{W}_{\mathsf{norm}} \hat{\mathbf{B}}_1 + \mathbf{W}_{\mathsf{sf},1} \hat{\mathbf{B}}_2 + \mathbf{a}^\perp \mathbf{w}_{\mathsf{sf},2}^\intercal \hat{\mathbf{B}}_2\big)\big(\hat{\mathbf{B}}_1^* \hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^* \hat{\mathbf{S}}_2 \mathbf{P}_j\big) \\ &= \mathbf{W}_{\mathsf{norm}} \hat{\mathbf{S}}_1 + \mathbf{W}_{\mathsf{sf},1} \hat{\mathbf{S}}_2 \mathbf{P}_j + \mathbf{a}^\perp \mathbf{w}_{\mathsf{sf},2}^\intercal \hat{\mathbf{S}}_2 \mathbf{P}_j. \end{aligned}$$

From Eq. (4.1), $\mathbf{P}_j = \operatorname{diag}\big([\mathbf{1}^{1 \times j} \mid \mathbf{0}^{1 \times (\ell - j)}]\big)$, so

$$\mathbf{P}_j \begin{bmatrix} \mathbf{0}^{j \times (\ell - j)} \\ \mathbf{I}_{\ell - j} \end{bmatrix} = \mathbf{0}.$$

Correspondingly, by Eq. (4.4),

$$\mathbf{Z} = \mathbf{W}\hat{\mathbf{T}} \begin{bmatrix} \mathbf{0}^{j \times (\ell - j)} \\ \mathbf{I}_{\ell - j} \end{bmatrix} = \mathbf{W}_{\mathsf{norm}} \hat{\mathbf{S}}_1 \begin{bmatrix} \mathbf{0}^{j \times (\ell - j)} \\ \mathbf{I}_{\ell - j} \end{bmatrix}.$$

We conclude that the distribution of $\mathbf{Z}$ is identical in Hyb_1 and Hyb_2. Finally, we consider the remaining components in the CRS. Again, using the fact that $\mathbf{A}\mathbf{a}^{\perp} = \mathbf{0}$, we have that

$$\mathbf{A}\mathbf{W} = \mathbf{A}\big(\mathbf{W}_{\mathsf{norm}}\hat{\mathbf{B}}_1 + \mathbf{W}_{\mathsf{sf},1}\hat{\mathbf{B}}_2 + \mathbf{a}^{\perp}\mathbf{w}_{\mathsf{sf},2}^{\mathsf{T}}\hat{\mathbf{B}}_2)\big) = \mathbf{A}\big(\mathbf{W}_{\mathsf{norm}}\hat{\mathbf{B}}_1 + \mathbf{W}_{\mathsf{sf},1}\hat{\mathbf{B}}_2\big).$$

We conclude that the components of the CRS are distributed identically in Hyb_1 and Hyb_2. ◻

Lemma 4.19. $\Pr[\mathsf{Hyb}_2(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$.

Proof. By construction in Hyb_2, the components of crs are *independent* of the vector $\mathbf{w}_{\mathsf{sf},2}$. This means that the challenger in Hyb_2 can defer the sampling of $\mathbf{w}_{\mathsf{sf},2}$ until *after* the adversary outputs $[\hat{\mathbf{c}}]_2$, $[\hat{\mathbf{c}}']_2$, and $[\mathbf{v}]_2$. For the challenger to output 1 in Hyb_2, it must be the case that $\hat{\mathbf{B}}_2(\hat{\mathbf{c}} - \hat{\mathbf{c}}') \neq \mathbf{0}$ and $\mathbf{W}(\hat{\mathbf{c}} - \hat{\mathbf{c}}') = \mathbf{v}$. We argue that over the choice of $\mathbf{w}_{\mathsf{sf},2}$, the probability that $\mathbf{W}(\hat{\mathbf{c}} - \hat{\mathbf{c}}') = \mathbf{v}$ is negligible. Since $\mathbf{W} = \big(\mathbf{W}_{\mathsf{norm}}\hat{\mathbf{B}}_1 + \mathbf{W}_{\mathsf{sf},1}\hat{\mathbf{B}}_2 + \mathbf{a}^{\perp}\mathbf{w}_{\mathsf{sf},2}^{\mathsf{T}}\hat{\mathbf{B}}_2\big)$, this means that

$$\mathbf{a}^{\perp} \cdot \mathbf{w}_{\mathsf{sf},2}^{\mathsf{T}}\hat{\mathbf{B}}_2(\hat{\mathbf{c}} - \hat{\mathbf{c}}') = \mathbf{v} - \big(\mathbf{W}_{\mathsf{norm}}\hat{\mathbf{B}}_1 + \mathbf{W}_{\mathsf{sf},1}\hat{\mathbf{B}}_2\big)(\hat{\mathbf{c}} - \hat{\mathbf{c}}') \in \mathbb{Z}_p^{k+1}.$$

Since $\hat{\mathbf{B}}_2(\hat{\mathbf{c}} - \hat{\mathbf{c}}') \neq \mathbf{0}$ and $\mathbf{w}_{\mathsf{sf},2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$, the distribution of $\mathbf{w}_{\mathsf{sf},2}^{\mathsf{T}}\hat{\mathbf{B}}_2(\hat{\mathbf{c}} - \hat{\mathbf{c}}')$ is uniform over $\mathbb{Z}_p$. Finally, since $\mathbf{a}^{\perp} \neq \mathbf{0}$ and the challenger samples $\mathbf{w}_{\mathsf{sf},2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$ *after* all other quantities have been fixed, we conclude that

$$\Pr\left[\mathbf{a}^{\perp} \cdot \mathbf{w}_{\mathsf{sf},2}^{\mathsf{T}}\hat{\mathbf{B}}_2(\hat{\mathbf{c}} - \hat{\mathbf{c}}') = \mathbf{v} - \big(\mathbf{W}_{\mathsf{norm}}\hat{\mathbf{B}}_1 + \mathbf{W}_{\mathsf{sf},1}\hat{\mathbf{B}}_2\big)(\hat{\mathbf{c}} - \hat{\mathbf{c}}') : \mathbf{w}_{\mathsf{sf},2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k\right] \leq \frac{1}{p} = \mathsf{negl}(\lambda). \quad ◻$$

By Lemmas 4.17 to 4.19, we conclude that $\Pr[\mathsf{Hyb}_0(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$. Thus, Construction 4.14 satisfies prefix-matching security. ◻

4.3 Proving Linear Relations on Committed Values

The second proof system we design is to argue that a Type-II commitment is consistent with a *linear* function applied to a Type-I commitment. Specifically, we describe a succinct proof system for statements of the following flavor: for a linear function $f : \mathbb{Z}_p^{\ell} \to \mathbb{Z}_p^{\ell}$:,

if σ_1 is a Type-I commitment to a vector $\mathbf{x} \in \mathbb{Z}_p^{\ell}$, then σ_2 is a Type-II commitment to the vector $\mathbf{y} = f(\mathbf{x})$.

Specifically, the "binding" requirement is that the adversary cannot open an input commitment σ_1 to two different output commitments σ_2, σ_2' with respect to the same linear function f. Following [BCFL23], we refer to this property as a linear chain binding property (also called arguments of knowledge transfer in [GR19, GZ21]). Similar to our prefix-checking proof system from Section 4.2, the chaining property is enforced in the *semi-functional* space (i.e., if σ_1 and σ_1' agree in their semi-functional space, then σ_2, σ_2' must also agree in their semi-functional space).

Projective chain binding for *local* functions. The security analysis of our functional commitment scheme in Section 5 relies on a stronger notion of chain binding tailored to S-local linear functions (Definition 4.2). At a high level, our security requirement captures the following idea:

- Let $\mathbf{x}_{j_1}$ denote the first j_1 components of a vector $\mathbf{x}$ and let $\mathbf{y}_{j_2}$ denote the first j_2 components of a vector $\mathbf{y}$. If $(j_1, j_2) \in S$ and the function f is S-local, then the value of $\mathbf{y}_{j_2}$ is *entirely* determined by the value of $\mathbf{x}_{j_1}$.
- Our notion of S-local chain binding then says that given two Type-I commitments σ_1, σ_1' whose Type-I projections are identical on the first j_1 components, then the adversary should not be able to open σ_1, σ_1' to Type-II commitments σ_2, σ_2' whose Type-II projections disagree in the first j_2 components with respect to the function f. Observe that unlike standard chain binding, the adversary chooses *two* input commitments and two output commitments (in standard chain binding, the adversary only chooses a single input commitment and must open it two different ways).

We now give the formal definition.

Definition 4.20 (Projective Chainable Commitments for Linear Functions)**.** Let $\mathsf{FC}_{\mathsf{base}} = \big(\mathsf{SetupBase}, \mathsf{SetupSF}, \mathsf{Commit}^{(1)}, \mathsf{Commit}^{(2)}, \mathsf{Project}^{(1)}, \mathsf{Project}^{(2)}\big)$ be a projective commitment scheme. In the following description, we represent linear functions $f(\mathbf{x}) := \mathbf{M}\mathbf{x}$ by a matrix $\mathbf{M}$. A chainable proof system for linear functions is a triple of efficient algorithms $\mathsf{FC}_{\mathsf{lin}} = \big(\mathsf{SetupLin}, \mathsf{OpenLin}, \mathsf{VerifyLin}\big)$ with the following properties:

- $\mathsf{SetupLin}(\mathsf{crs}_{\mathsf{base}}, S) \to \mathsf{crs}$: On input the common reference string $\mathsf{crs}_{\mathsf{base}}$ (which defines the input space $\mathcal{R}^\ell$) and a locality set $S \subseteq [\ell] \times [\ell]$, the setup algorithm outputs a common reference string crs.
- $\mathsf{OpenLin}(\mathsf{crs}, \mathbf{x}, \mathbf{M}) \to \pi$: On input a common reference string crs, an input vector $\mathbf{x} \in \mathcal{R}^\ell$, and a linear function $\mathbf{M} \in \mathcal{R}^{\ell \times \ell}$, the opening algorithm outputs a proof π.
- $\mathsf{VerifyLin}(\mathsf{crs}, \sigma_1, \mathbf{M}, \sigma_2, \pi) \to b$: On input the common reference string crs, a Type-I commitment σ_1, a linear function $\mathbf{M} \in \mathcal{R}^{\ell \times \ell}$, a Type-II commitment σ_2, and a proof π, the verification algorithm outputs a bit $b \in \{0, 1\}$.

The proof system should satisfy the following two properties:

- **Correctness:** For all security parameters $\lambda \in \mathbb{N}$, all vector lengths $\ell \in \mathbb{N}$, all locality sets $S \subseteq [\ell] \times [\ell]$, all $\mathsf{crs}_{\mathsf{base}}$ in the support of $\mathsf{SetupBase}(1^\lambda, 1^\ell)$, all vectors $\mathbf{x} \in \mathcal{R}^\ell$ (where $\mathcal{R}^\ell$ is the message space associated with $\mathsf{crs}_{\mathsf{base}}$), and all S-local linear functions $\mathbf{M} \in \mathcal{R}^{\ell \times \ell}$,

$$\Pr\left[\mathsf{VerifyLin}(\mathsf{crs}, \sigma_1, \mathbf{M}, \sigma_2, \pi) = 1 : \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{SetupLin}(\mathsf{crs}_{\mathsf{base}}, S) \\ \sigma_1 \leftarrow \mathsf{Commit}^{(1)}(\mathsf{crs}_{\mathsf{base}}, \mathbf{x}) \\ \sigma_2 \leftarrow \mathsf{Commit}^{(2)}(\mathsf{crs}_{\mathsf{base}}, \mathbf{M}\mathbf{x}) \\ \pi \leftarrow \mathsf{OpenLin}(\mathsf{crs}, \mathbf{x}, \mathbf{M}) \end{array}\right] = 1.$$

- **Chain binding for linear functions:** For a security parameter λ and an adversary $\mathcal{A}$, we define the chain binding for linear functions security game as follows:

  1. On input the security parameter λ, the adversary outputs the dimension 1^ℓ, a locality set $S \subseteq [\ell] \times [\ell]$, and a pair $(j_1, j_2) \in S$.
  2. The challenger samples $(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^\lambda, 1^\ell, j_1, j_2)$ and $\mathsf{crs} \leftarrow \mathsf{SetupLin}(\mathsf{crs}_{\mathsf{base}}, S)$. It gives $(\mathsf{crs}_{\mathsf{base}}, \mathsf{crs})$ to $\mathcal{A}$.
  3. The adversary outputs an S-local function $\mathbf{M} \in \mathcal{R}^{\ell \times \ell}$, two Type-I commitments (σ_1, σ_1'), two Type-II commitments (σ_2, σ_2'), and two openings π, π'.
  4. The challenger outputs $b = 1$ if all the following properties hold:
     - **Matching inputs:** $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$.
     - **Mismatching outputs:** $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) \neq \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2')$.
     - **Validity of openings:** $\mathsf{VerifyLin}(\mathsf{crs}, \sigma_1, \mathbf{M}, \sigma_2, \pi) = 1 = \mathsf{VerifyLin}(\mathsf{crs}, \sigma_1', \mathbf{M}, \sigma_2', \pi')$.

     Otherwise, the challenger outputs $b = 0$.

  We say that $\mathsf{FC}_{\mathsf{lin}}$ satisfies chain binding for linear functions if for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that $\Pr[b = 1] = \mathsf{negl}(\lambda)$ in the chain binding for linear functions security game.

Constructing projective chainable commitments. We now show how to construct a projective chainable commitment for local linear functions on top of the base projective commitment scheme from Section 4.1 (Construction 4.8). Before describing our construction, we define the projection matrix for a local linear function.

Definition 4.21 (Projection Matrix for a Local Linear Function)**.** Let $\ell \in \mathbb{N}$ be an input length. For indices $j_1, j_2 \in [\ell]$, we define the projection matrix $\mathbf{P}_{\mathsf{lin}}^{(j_1,j_2)}$ to be

$$\mathbf{P}_{\mathsf{lin}}^{(j_1,j_2)} := \mathbf{I}_{\ell^2} - \left(\mathbf{I}_\ell - \mathbf{P}_{j_1}\right) \otimes \mathbf{P}_{j_2} \in \{0,1\}^{\ell^2 \times \ell^2}, \tag{4.7}$$

where $\mathbf{P}_{j_1}, \mathbf{P}_{j_2} \in \{0,1\}^{\ell \times \ell}$ are the projection matrices from Definition 4.1. For a locality set $S \subseteq [\ell] \times [\ell]$, we define the projection matrix for S to be

$$\mathbf{P}_{\mathsf{lin}}^{(S)} := \prod_{(j_1,j_2) \in S} \mathbf{P}_{\mathsf{lin}}^{(j_1,j_2)} \in \{0,1\}^{\ell^2 \times \ell^2}. \tag{4.8}$$

Lemma 4.22 (Projection Matrix for a Local Linear Function)**.** *Let $\ell \in \mathbb{N}$ be an input length and $S \subseteq [\ell] \times [\ell]$ be a locality set. Suppose $f \colon \mathbb{Z}_p^\ell \to \mathbb{Z}_p^\ell$ is an S-local linear function $f(\mathbf{x}) := \mathbf{M}\mathbf{x}$ where $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell}$. Let $\mathbf{P}_{\mathsf{lin}} := \mathbf{P}_{\mathsf{lin}}^{(S)}$ be the projection matrix associated with S from Definition 4.21. Then the following properties hold:*

- $\mathrm{vec}(\mathbf{M})^\intercal \mathbf{P}_{\mathsf{lin}} = \mathrm{vec}(\mathbf{M})^\intercal$.
- *For all $(j_1, j_2) \in S$ and all vectors $\mathbf{r} \in \mathbb{Z}_p^\ell$, $\mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{r}^\intercal \mathbf{P}_{j_2})\right)(\mathbf{I}_\ell - \mathbf{P}_{j_1}) = \mathbf{0}$, where $\mathbf{P}_{j_1}, \mathbf{P}_{j_2} \in \{0,1\}^{\ell \times \ell}$ are the projection matrices from Definition 4.1.*

Proof. We show each claim individually:

- For the first claim, we start by observing that if f is (j_1, j_2)-local, then the first j_2 components of $\mathbf{M}\mathbf{e}_i$ are zero for all $i > j_1$ and where $\mathbf{e}_i \in \{0,1\}^\ell$ is the i^{th} basis vector. In other words,

$$\mathbf{P}_{j_2} \cdot \mathbf{M} \cdot (\mathbf{I}_\ell - \mathbf{P}_{j_1}) = \mathbf{0}. \tag{4.9}$$

  Then, for all $(j_1, j_2) \in S$,

$$\begin{aligned}
\mathrm{vec}(\mathbf{M})^\intercal \mathbf{P}_{\mathsf{lin}}^{(j_1,j_2)} &= \mathrm{vec}(\mathbf{M})^\intercal \left[\mathbf{I}_{\ell^2} - (\mathbf{I}_\ell - \mathbf{P}_{j_1}) \otimes \mathbf{P}_{j_2}\right] \\
&= \mathrm{vec}(\mathbf{M})^\intercal - \mathrm{vec}(\mathbf{M})^\intercal\left((\mathbf{I}_\ell - \mathbf{P}_{j_1}) \otimes \mathbf{P}_{j_2}\right) \\
&= \mathrm{vec}(\mathbf{M})^\intercal - \mathrm{vec}\left(\mathbf{P}_{j_2}^\intercal \mathbf{M}(\mathbf{I}_\ell - \mathbf{P}_{j_1})\right) && \text{by Eq. (3.4)} \\
&= \mathrm{vec}(\mathbf{M})^\intercal && \text{by Eq. (4.9) and since } \mathbf{P}_{j_2} = \mathbf{P}_{j_2}^\intercal.
\end{aligned}$$

  Since f is (j_1, j_2)-local for all $(j_1, j_2) \in S$, we have that

$$\mathrm{vec}(\mathbf{M})^\intercal \mathbf{P}_{\mathsf{lin}} = \mathrm{vec}(\mathbf{M})^\intercal \prod_{(j_1,j_2)\in S} \mathbf{P}_{\mathsf{lin}}^{(j_1,j_2)} = \mathrm{vec}(\mathbf{M})^\intercal.$$

- For the second claim, take any $(j_1, j_2) \in S$, and let $\mathbf{Q}_{j_1} = \mathbf{I}_\ell - \mathbf{P}_{j_1} \in \{0,1\}^{\ell \times \ell}$. Then,

$$(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{r}^\intercal \mathbf{P}_{j_2}))\mathbf{Q}_{j_1} = (\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{r}^\intercal \mathbf{P}_{j_2}))(\mathbf{Q}_{j_1} \otimes 1) = \mathbf{Q}_{j_1} \otimes \mathrm{vec}(\mathbf{r}^\intercal \mathbf{P}_{j_2}).$$

  Since $\mathbf{Q}_{j_1}$ is a diagonal matrix and its entries are in $\{0,1\}$, it follows that $\mathbf{Q}_{j_1}^2 = \mathbf{Q}_{j_1}$. Similarly, since $\mathbf{P}_{j_2}$ is a diagonal matrix with entries in $\{0,1\}$, we have $\mathbf{P}_{j_2}\mathbf{P}_{j_2}^\intercal = \mathbf{P}_{j_2}^2 = \mathbf{P}_{j_2}$. Then,

$$\begin{aligned}
(\mathbf{Q}_{j_1} \otimes \mathbf{P}_{j_2})(\mathbf{Q}_{j_1} \otimes \mathrm{vec}(\mathbf{r}^\intercal \mathbf{P}_{j_2})) &= \mathbf{Q}_{j_1}^2 \otimes \left((\mathbf{P}_{j_2} \otimes 1) \cdot \mathrm{vec}(\mathbf{r}^\intercal \mathbf{P}_{j_2})\right) && \text{by Eq. (3.1)} \\
&= \mathbf{Q}_{j_1} \otimes \mathrm{vec}(\mathbf{r}^\intercal \mathbf{P}_{j_2} \mathbf{P}_{j_2}^\intercal) && \text{by Eq. (3.4)} \\
&= \mathbf{Q}_{j_1} \otimes \mathrm{vec}(\mathbf{r}^\intercal \mathbf{P}_{j_2}) && \text{since } \mathbf{P}_{j_2}\mathbf{P}_{j_2}^\intercal = \mathbf{P}_{j_2}.
\end{aligned} \tag{4.10}$$

  Combining the above two relations and using the fact that $\mathbf{P}_{\mathsf{lin}}^{(j_1,j_2)} = \mathbf{I}_{\ell^2} - (\mathbf{I}_\ell - \mathbf{P}_{j_1}) \otimes \mathbf{P}_{j_2} = \mathbf{I}_{\ell^2} - \mathbf{Q}_{j_1} \otimes \mathbf{P}_{j_2}$,

$$\begin{aligned}
\mathbf{P}_{\mathsf{lin}}^{(j_1,j_2)}\left(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{r}^\intercal \mathbf{P}_{j_2})\right)(\mathbf{I}_\ell - \mathbf{P}_{j_1}) &= \mathbf{P}_{\mathsf{lin}}^{(j_1,j_2)}\left(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{r}^\intercal \mathbf{P}_{j_2})\right)\mathbf{Q}_{j_1} \\
&= \mathbf{P}_{\mathsf{lin}}^{(j_1,j_2)}\left(\mathbf{Q}_{j_1} \otimes \mathrm{vec}(\mathbf{r}^\intercal \mathbf{P}_{j_2})\right) && \text{by Eq. (3.1)} \\
&= (\mathbf{I}_{\ell^2} - (\mathbf{Q}_{j_1} \otimes \mathbf{P}_{j_2}))\left(\mathbf{Q}_{j_1} \otimes \mathrm{vec}(\mathbf{r}^\intercal \mathbf{P}_{j_2})\right) && \text{by definition of } \mathbf{P}_{\mathsf{lin}}^{(j_1,j_2)} \\
&= \left(\mathbf{Q}_{j_1} \otimes \mathrm{vec}(\mathbf{r}^\intercal \mathbf{P}_{j_2})\right) - \left(\mathbf{Q}_{j_1} \otimes \mathrm{vec}(\mathbf{r}^\intercal \mathbf{P}_{j_2})\right) && \text{by Eq. (4.10)} \\
&= \mathbf{0}.
\end{aligned}$$

Finally, since the matrices $\mathbf{P}_{\mathsf{lin}}^{(j_1,j_2)}$ are diagonal for all $j_1, j_2 \in [\ell]$, they commute so we can write

$$\mathbf{P}_{\mathsf{lin}} = \prod_{(s,t)\in S} \mathbf{P}_{\mathsf{lin}}^{(s,t)} = \left(\prod_{(s,t)\in S\setminus\{(j_1,j_2)\}} \mathbf{P}_{\mathsf{lin}}^{(s,t)} \right) \cdot \mathbf{P}_{\mathsf{lin}}^{(j_1,j_2)}.$$

Correspondingly,

$$\mathbf{P}_{\mathsf{lin}} \left(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{r}^\mathsf{T} \mathbf{P}_{j_2}) \right) (\mathbf{I}_\ell - \mathbf{P}_{j_1}) = \left(\prod_{(s,t)\in S\setminus\{(j_1,j_2)\}} \mathbf{P}_{\mathsf{lin}}^{(s,t)} \right) \cdot \mathbf{P}_{\mathsf{lin}}^{(j_1,j_2)} \left(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{r}^\mathsf{T} \mathbf{P}_{j_2}) \right) (\mathbf{I}_\ell - \mathbf{P}_{j_1}) = \mathbf{0}. \qquad \square$$

Construction 4.23 (Projective Chainable Commitments for Local Linear Functions)**.** Let $\mathsf{FC}_{\mathsf{base}} = \big(\mathsf{SetupBase}, \mathsf{SetupSF}, \mathsf{Commit}^{(1)}, \mathsf{Commit}^{(2)}, \mathsf{Project}^{(1)}, \mathsf{Project}^{(2)}\big)$ be the projective commitment scheme from Construction 4.8. We build a projective chainable commitment for local linear functions $\mathsf{FC}_{\mathsf{lin}} = \big(\mathsf{SetupLin}, \mathsf{OpenLin}, \mathsf{VerifyLin}\big)$ over $\mathsf{FC}_{\mathsf{base}}$ as follows:

- $\mathsf{SetupLin}(\mathsf{crs}_{\mathsf{base}}, S)$: On input the common reference string $\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big)$ for the base projective commitment scheme (which defines the input space $\mathbb{Z}_p^\ell$) and a locality set $S \subseteq [\ell] \times [\ell]$, the setup algorithm samples $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k\times(k+1)}$. Then, for $\alpha \in \{1,2\}$, it samples $\mathbf{R}_\alpha \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(k+1)\times 2k}$ and $\mathbf{W}_\alpha \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^2(k+1)\times 2k}$. It computes
$$\begin{aligned}[][\mathbf{Z}_\alpha]_2 &= \mathbf{W}_\alpha [\hat{\mathbf{T}}]_2 - (\mathbf{P}_{\mathsf{lin}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{R}_\alpha [\mathbf{T}_\alpha]_2)) \\ &= [\mathbf{W}_\alpha \hat{\mathbf{T}} - (\mathbf{P}_{\mathsf{lin}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{R}_\alpha \mathbf{T}_\alpha))]_2 \in \mathbb{G}_2^{\ell^2(k+1)\times\ell},\end{aligned} \tag{4.11}$$
where $\mathbf{P}_{\mathsf{lin}} \coloneqq \mathbf{P}_{\mathsf{lin}}^{(S)}$ is the projection matrix from Eq. (4.8). Output the common reference string
$$\mathsf{crs} = \Big(\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, \big\{[(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_\alpha]_1, [\mathbf{A}\mathbf{R}_\alpha]_1, [\mathbf{Z}_\alpha]_2\big\}_{\alpha\in\{1,2\}}\Big). \tag{4.12}$$

- $\mathsf{OpenLin}(\mathsf{crs}, \mathbf{x}, \mathbf{M})$: On input the common reference string crs (parsed as in Eq. (4.12)), the vector $\mathbf{x} \in \mathbb{Z}_p^\ell$, and the matrix $\mathbf{M} \in \mathbb{Z}_p^{\ell\times\ell}$, the opening algorithm computes for each $\alpha \in \{1,2\}$,
$$[\mathbf{v}_\alpha]_2 = (\mathrm{vec}(\mathbf{M})^\mathsf{T} \otimes \mathbf{I}_{k+1})[\mathbf{Z}_\alpha]_2 \mathbf{x} \in \mathbb{G}_2^{k+1}$$
along with $[\mathbf{c}_1']_2 = [\mathbf{T}_1]_2 \mathbf{M}\mathbf{x} = [\mathbf{T}_1 \mathbf{M}\mathbf{x}]_2 \in \mathbb{G}_2^{2k}$. It outputs the opening $\pi = ([\mathbf{c}_1']_2, [\mathbf{v}_1]_2, [\mathbf{v}_2]_2)$.

- $\mathsf{VerifyLin}(\mathsf{crs}, \sigma_1, \mathbf{M}, \sigma_2, \pi)$: On input the common reference string crs (parsed as in Eq. (4.12)), a Type-I commitment $\sigma_1 = [\hat{\mathbf{c}}]_2$, a matrix $\mathbf{M} \in \mathbb{Z}_p^{\ell\times\ell}$, a Type-II commitment $\sigma_2 = ([\mathbf{c}_1]_1, [\mathbf{c}_2]_2)$, and a proof $\pi = ([\mathbf{c}_1']_2, [\mathbf{v}_1]_2, [\mathbf{v}_2]_2)$, the verification algorithm outputs 1 if the following conditions hold:
  - $[\mathbf{c}_1]_1 [1]_2 = [1]_1 [\mathbf{c}_1']_2$.
  - $(\mathrm{vec}(\mathbf{M})^\mathsf{T} \otimes \mathbf{I}_k)[(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_1]_1 [\hat{\mathbf{c}}]_2 = [\mathbf{A}\mathbf{R}_1]_1 [\mathbf{c}_1']_2 + [\mathbf{A}]_1 [\mathbf{v}_1]_2$.
  - $(\mathrm{vec}(\mathbf{M})^\mathsf{T} \otimes \mathbf{I}_k)[(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_2]_1 [\hat{\mathbf{c}}]_2 = [\mathbf{A}\mathbf{R}_2]_1 [\mathbf{c}_2]_2 + [\mathbf{A}]_1 [\mathbf{v}_2]_2$.

Theorem 4.24 (Correctness)**.** *Construction 4.23 is correct.*

Proof. Take any $\lambda, \ell \in \mathbb{N}$ and let $S \subseteq [\ell] \times [\ell]$ be an arbitrary locality set. Let $\mathsf{crs}_{\mathsf{base}} \leftarrow \mathsf{SetupBase}(1^\lambda, 1^\ell)$ and parse $\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big)$ Let $\mathsf{crs} \leftarrow \mathsf{SetupLin}(\mathsf{crs}_{\mathsf{base}}, S)$, and parse

$$\mathsf{crs} = \Big(\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, \big\{[(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_\alpha]_1, [\mathbf{A}\mathbf{R}_\alpha]_1, [\mathbf{Z}_\alpha]_2\big\}_{\alpha\in\{1,2\}}\Big).$$

Take any vector $\mathbf{x} \in \mathbb{Z}_p^\ell$ and any S-local linear function $f(\mathbf{x}) \coloneqq \mathbf{M}\mathbf{x}$ where $\mathbf{M} \in \mathbb{Z}_p^{\ell\times\ell}$. Let $\mathbf{y} = \mathbf{M}\mathbf{x}$. Let $\sigma_1 \leftarrow \mathsf{Commit}^{(1)}(\mathsf{crs}_{\mathsf{base}}, \mathbf{x})$, $\sigma_2 \leftarrow \mathsf{Commit}^{(2)}(\mathsf{crs}_{\mathsf{base}}, \mathbf{y})$, and $\pi \leftarrow \mathsf{OpenLin}(\mathsf{crs}, \mathbf{x}, \mathbf{M})$. We parse $\sigma_1 = [\hat{\mathbf{c}}]_2$, $\sigma_2 = ([\mathbf{c}_1]_1, [\mathbf{c}_2]_2)$ and $\pi = ([\mathbf{c}_1']_1, [\mathbf{v}_1]_2, [\mathbf{v}_2]_2)$. Consider now $\mathsf{VerifyLin}(\mathsf{crs}, \sigma_1, \mathbf{M}, \sigma_2, \pi)$. By construction of the underlying algorithms, we now have the following:

- First, the commitments satisfy $\hat{\mathbf{c}} = \hat{\mathbf{T}}\mathbf{x}$, $\mathbf{c}_1 = \mathbf{T}_1\mathbf{y}$, and $\mathbf{c}_2 = \mathbf{T}_2\mathbf{y}$. In addition, $\mathbf{c}_1' = \mathbf{T}_1\mathbf{M}\mathbf{x} = \mathbf{T}_1\mathbf{y} = \mathbf{c}_1$, and the first verification relation holds.
- For the second verification relation, for $\alpha \in \{1,2\}$, we have

$$\begin{aligned}(\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_\alpha\hat{\mathbf{c}} &= (\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_\alpha\hat{\mathbf{T}}\mathbf{x} \\ &= (\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{A})\mathbf{W}_\alpha\hat{\mathbf{T}}\mathbf{x} && \text{by Eq. (3.1)} \\ &= \mathbf{A}(\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})\mathbf{W}_\alpha\hat{\mathbf{T}}\mathbf{x} && \text{by Eq. (3.3).}\end{aligned} \tag{4.13}$$

Since f is S-local, by Lemma 4.22, we have that $\mathrm{vec}(\mathbf{M})^\top\mathbf{P}_{\mathsf{lin}} = \mathrm{vec}(\mathbf{M})^\top$. Then, we can write

$$\begin{aligned}(\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})\mathbf{Z}_\alpha &= (\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})\mathbf{W}_\alpha\hat{\mathbf{T}} - (\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})(\mathbf{P}_{\mathsf{lin}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{R}_\alpha\mathbf{T}_\alpha)) \\ &= (\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})\mathbf{W}_\alpha\hat{\mathbf{T}} - (\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{R}_\alpha\mathbf{T}_\alpha)).\end{aligned}$$

Thus, we have

$$(\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})\mathbf{W}_\alpha\hat{\mathbf{T}} = (\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})\mathbf{Z}_\alpha + (\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{R}_\alpha\mathbf{T}_\alpha)).$$

Substituting back into Eq. (4.13), and using the fact that $\mathbf{v}_\alpha = (\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})\mathbf{Z}_\alpha\mathbf{x}$, we have

$$\begin{aligned}(\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_\alpha\hat{\mathbf{c}} &= \mathbf{A}(\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})\mathbf{W}_\alpha\hat{\mathbf{T}}\mathbf{x} \\ &= \mathbf{A}(\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})\big(\mathbf{Z}_\alpha\mathbf{x} + (\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{R}_\alpha\mathbf{T}_\alpha))\mathbf{x}\big) \\ &= \mathbf{A}\mathbf{v}_\alpha + \mathbf{A}(\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{R}_\alpha\mathbf{T}_\alpha))\mathbf{x} \\ &= \mathbf{A}\mathbf{v}_\alpha + \mathbf{A}(\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})(\mathbf{x} \otimes \mathrm{vec}(\mathbf{R}_\alpha\mathbf{T}_\alpha)).\end{aligned} \tag{4.14}$$

To complete the proof, we now have

$$\begin{aligned}(\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})(\mathbf{x} \otimes \mathrm{vec}(\mathbf{R}_\alpha\mathbf{T}_\alpha)) &= (\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})(\mathbf{x} \otimes \mathbf{I}_\ell \otimes \mathbf{I}_{k+1})\mathrm{vec}(\mathbf{R}_\alpha\mathbf{T}_\alpha) && \text{by Eq. (3.2)} \\ &= \big((\mathrm{vec}(\mathbf{M})^\top(\mathbf{x} \otimes \mathbf{I}_\ell)) \otimes \mathbf{I}_{k+1}\big)\mathrm{vec}(\mathbf{R}_\alpha\mathbf{T}_\alpha) && \text{by Eq. (3.1)} \\ &= \big((\mathbf{M}\mathbf{x})^\top \otimes \mathbf{I}_{k+1}\big)\mathrm{vec}(\mathbf{R}_\alpha\mathbf{T}_\alpha) && \text{by Eq. (3.4)} \\ &= \mathbf{R}_\alpha\mathbf{T}_\alpha\mathbf{M}\mathbf{x} = \mathbf{R}_\alpha\mathbf{T}_\alpha\mathbf{y} = \mathbf{R}_\alpha\mathbf{c}_\alpha && \text{by Eq. (3.4).}\end{aligned}$$

Substituting back into Eq. (4.14), we have Since $\mathbf{v}_\alpha = (\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})\mathbf{Z}_\alpha\mathbf{x}$, we can now write

$$\begin{aligned}(\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_\alpha\hat{\mathbf{c}} &= \mathbf{A}\mathbf{v}_\alpha + \mathbf{A}(\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})(\mathbf{x} \otimes \mathrm{vec}(\mathbf{R}_\alpha\mathbf{T}_\alpha)) \\ &= \mathbf{A}\mathbf{v}_\alpha + \mathbf{A}\mathbf{R}_\alpha\mathbf{c}_\alpha.\end{aligned}$$

Since $\mathbf{c}_1' = \mathbf{c}_1$, this means the second and third verification relations hold. ☐

Theorem 4.25 (Chain Binding for Linear Functions)**.** *Suppose the* k-KerLin *assumption holds in* $\mathbb{G}_1$ *with respect to* GroupGen *and the* k-Lin *assumption holds in* $\mathbb{G}_2$ *with respect to* GroupGen. *Then, Construction 4.23 satisfies chain binding for linear functions.*

Proof. To simplify the proof, we start by defining a "homogeneous" version of the chain binding for linear functions security game for Construction 4.23. We define the game below:

1. On input the security parameter λ, the adversary outputs the dimension 1^ℓ, a locality set $S \subseteq [\ell] \times [\ell]$, and a pair $(j_1, j_2) \in S$.
2. The challenger samples $(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^\lambda, 1^\ell, j_1, j_2)$ and $\mathsf{crs} \leftarrow \mathsf{SetupLin}(\mathsf{crs}_{\mathsf{base}}, S)$. Then, $\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big)$, $\mathsf{td}_1 = \hat{\mathbf{B}}_2$, $\mathsf{td}_2 = (\mathbf{B}_{1,2}, \mathbf{B}_{2,2})$, and

$$\mathsf{crs} = \Big(\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, \big\{[(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_\alpha]_1, [\mathbf{A}\mathbf{R}_\alpha]_1, [\mathbf{Z}_\alpha]_2\big\}_{\alpha \in \{1,2\}}\Big).$$

The challenger gives crs to $\mathcal{A}$.

3. The adversary outputs an S-local function $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell}$ and a tuple $\big([\hat{\mathbf{c}}]_2, [\mathbf{c}_1]_2, [\mathbf{c}_2]_2, [\mathbf{v}_1]_2, [\mathbf{v}_2]_2\big)$.

4. The challenger outputs 1 if the following properties hold:

   - **Matching inputs:** $\hat{\mathbf{B}}_2 \hat{\mathbf{c}} = \mathbf{0}$.
   - **Mismatching outputs:** either $\mathbf{B}_{1,2}\mathbf{c}_1 \neq \mathbf{0}$ or $\mathbf{B}_{2,2}\mathbf{c}_2 \neq \mathbf{0}$.
   - **Validity of openings:** for each $\alpha \in \{1, 2\}$, $(\mathrm{vec}(\mathbf{M})^{\intercal} \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_\alpha \hat{\mathbf{c}} = \mathbf{A}\mathbf{R}_\alpha \mathbf{c}_\alpha + \mathbf{A}\mathbf{v}_\alpha$.

We now show that any adversary that can win the homogeneous chain binding security game (i.e., cause the above experiment to output 1) implies an adversary that can win the standard chain binding security game (Definition 4.20). The claim essentially follows by linearity of the verification relation. We give the formal statement below:

Lemma 4.26. *Suppose for all efficient adversaries $\mathcal{B}$, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that* $\Pr[b = 1] = \mathsf{negl}(\lambda)$ *in the homogeneous chain binding experiment for linear functions. Then, Construction 4.23 satisfies chain binding security for linear functions.*

Proof. Suppose there exists an adversary $\mathcal{A}$ that breaks chain binding security for linear functions (Definition 4.20) with advantage ε. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ that wins the homogeneous chain binding game:

1. Algorithm $\mathcal{B}$ starts running algorithm $\mathcal{A}$ to obtain the input length 1^ℓ, the locality set $S \subseteq [\ell] \times [\ell]$, and a pair $(j_1, j_2) \in S$. It gives 1^ℓ, S, and (j_1, j_2) to the challenger to obtain the common reference string crs.

2. Algorithm $\mathcal{B}$ forwards crs to $\mathcal{A}$ and receives a function $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell}$, two Type-I commitments $\sigma_1 = [\hat{\mathbf{c}}]_2$, $\sigma_1' = [\hat{\mathbf{c}}']_2$, two Type-II commitments $\sigma_2 = ([\mathbf{c}_1]_1, [\mathbf{c}_2]_2)$, $\sigma_2' = ([\mathbf{c}_1']_1, [\mathbf{c}_2']_2)$, and two openings $\pi = ([\tilde{\mathbf{c}}_1]_2, [\mathbf{v}_1]_2, [\mathbf{v}_2]_2)$, $\pi' = ([\tilde{\mathbf{c}}_1']_2, [\mathbf{v}_1']_2, [\mathbf{v}_2']_2)$.

3. Algorithm $\mathcal{B}$ outputs the same function $\mathbf{M}$ together with the tuple

$$\big([\hat{\mathbf{c}}]_2 - [\hat{\mathbf{c}}']_2, [\tilde{\mathbf{c}}_1]_2 - [\tilde{\mathbf{c}}_1']_2, [\mathbf{c}_2]_2 - [\mathbf{c}_2']_2, [\mathbf{v}_1]_2 - [\mathbf{v}_1']_2, [\mathbf{v}_2]_2 - [\mathbf{v}_2']_2\big).$$

In the homogeneous chain binding game, the challenger samples $(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^\lambda, 1^\ell, j_1, j_2)$ and $\mathsf{crs} \leftarrow \mathsf{SetupLin}(\mathsf{crs}_{\mathsf{base}}, S)$. Thus algorithm $\mathcal{B}$ perfectly simulates an execution of the chain binding security game for $\mathcal{A}$. Thus, with probability ε, the outputs of algorithm $\mathcal{A}$ satisfies the following properties:

- **Matching inputs:** $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$.
- **Mismatching outputs:** $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) \neq \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2')$.
- **Validity of openings:** $\mathsf{VerifyLin}(\mathsf{crs}, \sigma_1, \mathbf{M}, \sigma_2, \pi) = 1 = \mathsf{VerifyLin}(\mathsf{crs}, \sigma_1', \mathbf{M}, \sigma_2', \pi')$.

We claim that in this case, the output in the homogeneous chain binding game is also 1:

- Parse $\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big)$ and

$$\mathsf{crs} = \Big(\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, \big\{[(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_\alpha]_1, [\mathbf{A}\mathbf{R}_\alpha]_1, [\mathbf{Z}_\alpha]_2\big\}_{\alpha \in \{1,2\}}\Big).$$

  In addition, parse $\mathsf{td}_1 = \hat{\mathbf{B}}_2$, $\mathsf{td}_2 = (\mathbf{B}_{1,2}, \mathbf{B}_{2,2})$.

- Since $\mathsf{VerifyLin}(\mathsf{crs}, \sigma_1, \mathbf{M}, \sigma_2, \pi) = 1 = \mathsf{VerifyLin}(\mathsf{crs}, \sigma_1', \mathbf{M}, \sigma_2', \pi')$, the following conditions hold:

  - $\mathbf{c}_1 = \tilde{\mathbf{c}}_1$ and $\mathbf{c}_1' = \tilde{\mathbf{c}}_1'$.

– For $\alpha \in \{1, 2\}$, we have that

$$(\mathrm{vec}(\mathbf{M})^{\intercal} \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_\alpha \hat{\mathbf{c}} = \mathbf{A}\mathbf{R}_\alpha \mathbf{c}_\alpha + \mathbf{A}\mathbf{v}_\alpha$$
$$(\mathrm{vec}(\mathbf{M})^{\intercal} \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_\alpha \hat{\mathbf{c}}' = \mathbf{A}\mathbf{R}_\alpha \mathbf{c}'_\alpha + \mathbf{A}\mathbf{v}'_\alpha,$$

where we have used the fact that $\mathbf{c}_1 = \tilde{\mathbf{c}}_1$ and $\mathbf{c}'_1 = \tilde{\mathbf{c}}'_1$. Taking the difference of these two relations, we have for each $\alpha \in \{1, 2\}$,

$$(\mathrm{vec}(\mathbf{M})^{\intercal} \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_\alpha (\hat{\mathbf{c}} - \hat{\mathbf{c}}') = \mathbf{A}\mathbf{R}_\alpha (\mathbf{c}_\alpha - \mathbf{c}'_\alpha) + \mathbf{A}(\mathbf{v}_\alpha - \mathbf{v}'_\alpha).$$

This is precisely the third requirement in the homogeneous game.

- First, $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma'_1)$ means that $\hat{\mathbf{B}}_2 \hat{\mathbf{c}} = \hat{\mathbf{B}}_2 \hat{\mathbf{c}}'$. Thus, $\hat{\mathbf{B}}_2(\hat{\mathbf{c}} - \hat{\mathbf{c}}') = \mathbf{0}$, so the first requirement in the homogeneous game is satisfied.
- Next $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) \neq \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma'_2)$ means that either $\mathbf{B}_{1,2}\tilde{\mathbf{c}}_1 \neq \mathbf{B}_{1,2}\tilde{\mathbf{c}}'_1$ or $\mathbf{B}_{2,2}\mathbf{c}_2 \neq \mathbf{B}_{2,2}\mathbf{c}'_2$. Since $\mathbf{c}_1 = \tilde{\mathbf{c}}_1$ and $\mathbf{c}'_1 = \tilde{\mathbf{c}}'_1$, this means that either $\mathbf{B}_{1,2}(\mathbf{c}_1 - \mathbf{c}'_1) \neq \mathbf{0}$ or $\mathbf{B}_{2,2}(\mathbf{c}_2 - \mathbf{c}'_2) \neq \mathbf{0}$, so the second requirement in the homogeneous game holds.

Correspondingly, the output is 1 in the homogeneous evaluation binding game, and the claim follows. □

Proof of Theorem 4.25. We now return to the proof of Theorem 4.25. Let $\mathcal{A}$ be an efficient adversary for the homogeneous chain binding experiment. Let $\ell \in \mathbb{N}$ be the vector dimension that $\mathcal{A}$ chooses (which will determine the size of the MDDH assumption in Lemma 4.31). We now define a sequence of hybrid experiments:

- Hyb_0: This is the homogeneous chain binding experiment. We recall the full specification here:
  – At the beginning of the game, the adversary $\mathcal{A}$ outputs the dimension ℓ, a locality set $S \subseteq [\ell] \times [\ell]$, and a pair $(j_1, j_2) \in S$.
  – The challenger samples $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e) \leftarrow \mathsf{GroupGen}(1^\lambda)$.
  – The challenger samples full-rank matrices $\hat{\mathbf{B}}, \mathbf{B}_1, \mathbf{B}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times 2k}$ and defines $\hat{\mathbf{B}}^* = \hat{\mathbf{B}}^{-1}$, $\mathbf{B}_1^* = \mathbf{B}_1^{-1}$, $\mathbf{B}_2^* = \mathbf{B}_2^{-1}$. It parses $\hat{\mathbf{B}}, \mathbf{B}_1, \mathbf{B}_2$ as in Eq. (4.2) and $\hat{\mathbf{B}}^*, \mathbf{B}_1^*, \mathbf{B}_2^*$ as in Eq. (4.3).
  – The challenger constructs the encoding matrices $\hat{\mathbf{T}}, \mathbf{T}_1, \mathbf{T}_2$ as follows:
    * **Type-I encodings:** Sample $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$ and let $\hat{\mathbf{T}} = \hat{\mathbf{B}}_1^* \hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^* \hat{\mathbf{S}}_2 \mathbf{P}_{j_1} \in \mathbb{Z}_p^{2k \times \ell}$.
    * **Type-II encodings:** For $\alpha \in \{1, 2\}$, sample $\mathbf{S}_{\alpha,1}, \mathbf{S}_{\alpha,2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$. Let $\mathbf{T}_\alpha = \mathbf{B}_{\alpha,1}^* \mathbf{S}_{\alpha,1} + \mathbf{B}_{\alpha,2}^* \mathbf{S}_{\alpha,2} \mathbf{P}_{j_2} \in \mathbb{Z}_p^{2k \times \ell}$.

    Let $\mathbf{T}_* = \mathbf{T}_1 \otimes \mathbf{T}_2$ and set $\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big)$.
  – The challenger samples $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times (k+1)}$. Then, for $\alpha \in \{1, 2\}$, it samples $\mathbf{R}_\alpha \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(k+1) \times 2k}$, $\mathbf{W}_\alpha \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^2 (k+1) \times 2k}$, and computes for each $\alpha \in \{1, 2\}$,

$$\mathbf{Z}_\alpha = \mathbf{W}_\alpha \hat{\mathbf{T}} - (\mathbf{P}_{\mathsf{lin}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{R}_\alpha \mathbf{T}_\alpha)), \tag{4.15}$$

    where $\mathbf{P}_{\mathsf{lin}} = \mathbf{P}_{\mathsf{lin}}^{(S)}$ is projection matrix from Eq. (4.8). The challenger gives the common reference string crs to $\mathcal{A}$ where

$$\mathsf{crs} = \Big(\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, \big\{[(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_\alpha]_1, [\mathbf{A}\mathbf{R}_\alpha]_1, [\mathbf{Z}_\alpha]_2\big\}_{\alpha \in \{1,2\}}\Big).$$

  – The adversary outputs an S-local function $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell}$ and a tuple $\big([\hat{\mathbf{c}}]_2, [\mathbf{c}_1]_2, [\mathbf{c}_2]_2, [\mathbf{v}_1]_2, [\mathbf{v}_2]_2\big)$.

  The output of the experiment is 1 if the following conditions hold:

$$\hat{\mathbf{B}}_2 \hat{\mathbf{c}} = \mathbf{0} \quad \text{and} \quad \forall \alpha \in \{1, 2\} : (\mathrm{vec}(\mathbf{M})^{\intercal} \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_\alpha \hat{\mathbf{c}} = \mathbf{A}\mathbf{R}_\alpha \mathbf{c}_\alpha + \mathbf{A}\mathbf{v}_\alpha \quad \text{and} \quad \mathbf{B}_{1,2}\mathbf{c}_1 \neq \mathbf{0} \text{ or } \mathbf{B}_{2,2}\mathbf{c}_2 \neq \mathbf{0}.$$

- Hyb_1: Same as Hyb_0, except the challenger samples $\mathbf{W}_{\mathsf{norm}}^{(\alpha)}, \mathbf{W}_{\mathsf{sf}}^{(\alpha)} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^2(k+1)\times k}$ for each $\alpha \in \{1,2\}$. It then sets $\mathbf{W}_\alpha = \mathbf{W}_{\mathsf{norm}}^{(\alpha)}\hat{\mathbf{B}}_1 + \mathbf{W}_{\mathsf{sf}}^{(\alpha)}\hat{\mathbf{B}}_2$ when setting up the CRS. After the adversary outputs $(\mathbf{M}, [\hat{\mathbf{c}}]_2, [\mathbf{c}_1]_2, [\mathbf{c}_2]_2, [\mathbf{v}_1]_2, [\mathbf{v}_2]_2)$, the challenger computes
$$\mathbf{v}'_\alpha = \mathbf{v}_\alpha - (\mathsf{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})\mathbf{W}_{\mathsf{norm}}^{(\alpha)}\hat{\mathbf{B}}_1\hat{\mathbf{c}}. \tag{4.16}$$
The output of the experiment is 1 if the following conditions hold:
$$\hat{\mathbf{B}}_2\hat{\mathbf{c}} = \mathbf{0} \quad \text{and} \quad \forall \alpha \in \{1,2\} : \mathbf{A}\mathbf{R}_\alpha\mathbf{c}_\alpha + \mathbf{A}\mathbf{v}'_\alpha = \mathbf{0} \quad \text{and} \quad \mathbf{B}_{1,2}\mathbf{c}_1 \neq \mathbf{0} \text{ or } \mathbf{B}_{2,2}\mathbf{c}_2 \neq \mathbf{0}.$$

- Hyb_2: Same as Hyb_1 except the output of the experiment is 1 if the following conditions hold:
$$\hat{\mathbf{B}}_2\hat{\mathbf{c}} = \mathbf{0} \quad \text{and} \quad \forall \alpha \in \{1,2\} : \mathbf{R}_\alpha\mathbf{c}_\alpha + \mathbf{v}'_\alpha = \mathbf{0} \quad \text{and} \quad \mathbf{B}_{1,2}\mathbf{c}_1 \neq \mathbf{0} \text{ or } \mathbf{B}_{2,2}\mathbf{c}_2 \neq \mathbf{0}.$$

- Hyb_3: Same as Hyb_2 except when constructing the CRS, the challenger samples a random nonzero vector $\mathbf{a}^\perp \in \mathbb{Z}_p^{k+1}$ in the kernel of $\mathbf{A}$ (i.e., $\mathbf{A}\mathbf{a}^\perp = \mathbf{0}$). Then, for each $\alpha \in \{1,2\}$, it samples $\mathbf{W}_{\mathsf{sf},1}^{(\alpha)} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^2(k+1)\times k}$, $\mathbf{W}_{\mathsf{sf},2}^{(\alpha)} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^2\times k}$. It also samples $\mathbf{R}_{\alpha,1} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(k+1)\times 2k}$ and $\mathbf{r}_{\alpha,2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k}$, and sets
$$\mathbf{W}_{\mathsf{sf}}^{(\alpha)} = \mathbf{W}_{\mathsf{sf},1}^{(\alpha)} + \left(\mathbf{W}_{\mathsf{sf},2}^{(\alpha)} \otimes \mathbf{a}^\perp\right) \quad \text{and} \quad \mathbf{R}_\alpha = \mathbf{R}_{\alpha,1} + \left(\mathbf{r}_{\alpha,2}^\intercal \otimes \mathbf{a}^\perp\right) = \mathbf{R}_{\alpha,1} + \mathbf{a}^\perp\mathbf{r}_{\alpha,2}^\intercal.$$
The challenger then computes
$$\begin{aligned}\mathbf{Z}_{\alpha,1} &= \left(\mathbf{W}_{\mathsf{norm}}^{(\alpha)}\hat{\mathbf{B}}_1 + \mathbf{W}_{\mathsf{sf},1}^{(\alpha)}\hat{\mathbf{B}}_2\right)\hat{\mathbf{T}} - (\mathbf{P}_{\mathsf{lin}} \otimes \mathbf{I}_{k+1})\left(\mathbf{I}_\ell \otimes \mathsf{vec}(\mathbf{R}_{\alpha,1}\mathbf{T}_\alpha)\right)\\ \mathbf{Z}_{\alpha,2} &= \mathbf{W}_{\mathsf{sf},2}^{(\alpha)}\hat{\mathbf{S}}_2\mathbf{P}_{j_1} - \mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathsf{vec}(\mathbf{r}_{\alpha,2}^\intercal\mathbf{T}_\alpha)\right)\end{aligned}$$
and sets $\mathbf{Z}_\alpha = \mathbf{Z}_{\alpha,1} + (\mathbf{I}_{\ell^2} \otimes \mathbf{a}^\perp)\mathbf{Z}_{\alpha,2}$.

- Hyb_4: Same as Hyb_3 except when constructing the CRS, the challenger sets
$$\mathsf{crs} = \left(\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, \left\{\left[(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\left(\mathbf{W}_{\mathsf{norm}}^{(\alpha)}\hat{\mathbf{B}}_1 + \mathbf{W}_{\mathsf{sf},1}^{(\alpha)}\hat{\mathbf{B}}_2\right)\right]_1, [\mathbf{A}\mathbf{R}_{\alpha,1}]_1, [\mathbf{Z}_\alpha]_2\right\}\right).$$

- Hyb_5: Same as Hyb_4, except for each $\alpha \in \{1,2\}$, the challenger samples $\mathbf{U}_\alpha \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^2\times\ell}$ and sets
$$\mathbf{Z}_{\alpha,2} = \mathbf{U}_\alpha\mathbf{P}_{j_1} - \mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathsf{vec}(\mathbf{r}_{\alpha,2}^\intercal\mathbf{T}_\alpha)\right).$$

- Hyb_6: Same as Hyb_5, except for each $\alpha \in \{1,2\}$ the challenger samples $\mathbf{r}_{\alpha,2,\mathsf{norm}}, \mathbf{r}_{\alpha,2,\mathsf{sf}} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$ and sets
$$\mathbf{r}_{\alpha,2}^\intercal = \mathbf{r}_{\alpha,2,\mathsf{norm}}^\intercal\mathbf{B}_{\alpha,1} + \mathbf{r}_{\alpha,2,\mathsf{sf}}^\intercal\mathbf{B}_{\alpha,2}.$$
Then, it sets
$$\mathbf{Z}_{\alpha,2} = \mathbf{U}_\alpha\mathbf{P}_{j_1} - \mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathsf{vec}(\mathbf{r}_{\alpha,2,\mathsf{norm}}^\intercal\mathbf{S}_{\alpha,1})\right) - \mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathsf{vec}(\mathbf{r}_{\alpha,2,\mathsf{sf}}^\intercal\mathbf{S}_{\alpha,2}\mathbf{P}_{j_2})\right).$$

- Hyb_7: Same as Hyb_6, except the challenger sets
$$\mathbf{Z}_{\alpha,2} = \mathbf{U}_\alpha\mathbf{P}_{j_1} - \mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathsf{vec}(\mathbf{r}_{\alpha,2,\mathsf{norm}}^\intercal\mathbf{S}_{\alpha,1})\right).$$
Recall that in this experiment, the challenger still samples $\mathbf{U}_\alpha \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^2\times\ell}$.

We write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output distribution of an execution of hybrid Hyb_i with adversary $\mathcal{A}$. We now show that the output distribution of each adjacent pair of hybrids is indistinguishable.

Lemma 4.27. $\Pr[\mathsf{Hyb}_0(\mathcal{A}) = 1] = \Pr[\mathsf{Hyb}_1(\mathcal{A}) = 1]$.

Proof. Since $\hat{\mathbf{B}}$ is a basis for $\mathbb{Z}_p^{2k}$ and the matrices $\mathbf{W}_{\mathsf{norm}}^{(\alpha)}$ and $\mathbf{W}_{\mathsf{sf}}^{(\alpha)}$ are uniform, the distribution of $\mathbf{W}^{(\alpha)}$ is also uniform in Hyb_1, and thus, is identical to the distribution in Hyb_0. It suffices to consider the outputs of the two experiments. Suppose $\mathcal{A}$ outputs $\big(\mathbf{M}, [\hat{\mathbf{c}}]_2, [\mathbf{c}_1]_2, [\mathbf{c}_2]_2, [\mathbf{v}_1]_2, [\mathbf{v}_2]_2\big)$. First, if $\hat{\mathbf{B}}_2\hat{\mathbf{c}} \neq \mathbf{0}$, then the output in both experiments is identical. Suppose then that $\hat{\mathbf{B}}_2\hat{\mathbf{c}} = \mathbf{0}$. This means that

$$\mathbf{W}_\alpha\hat{\mathbf{c}} = \mathbf{W}_{\mathsf{norm}}^{(\alpha)}\hat{\mathbf{B}}_1\hat{\mathbf{c}} + \mathbf{W}_{\mathsf{sf}}\hat{\mathbf{B}}_2\hat{\mathbf{c}} = \mathbf{W}_{\mathsf{norm}}^{(\alpha)}\hat{\mathbf{B}}_1\hat{\mathbf{c}}. \tag{4.17}$$

Consider the value of $\mathbf{A}\mathbf{R}_\alpha\mathbf{c}_\alpha + \mathbf{A}\mathbf{v}'_\alpha$ in Hyb_1:

$$\begin{aligned}
\mathbf{A}\mathbf{R}_\alpha\mathbf{c}_\alpha + \mathbf{A}\mathbf{v}'_\alpha &= \mathbf{A}\mathbf{R}_\alpha\mathbf{c}_\alpha + \mathbf{A}\mathbf{v}_\alpha - \mathbf{A}(\mathrm{vec}(\mathbf{M})^{\mathsf{T}} \otimes \mathbf{I}_{k+1})\mathbf{W}_{\mathsf{norm}}^{(\alpha)}\hat{\mathbf{B}}_1\hat{\mathbf{c}} && \text{by Eq. (4.16)} \\
&= \mathbf{A}\mathbf{R}_\alpha\mathbf{c}_\alpha + \mathbf{A}\mathbf{v}_\alpha - (\mathrm{vec}(\mathbf{M})^{\mathsf{T}} \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_{\mathsf{norm}}^{(\alpha)}\hat{\mathbf{B}}_1\hat{\mathbf{c}} && \text{by Eq. (3.3)} \\
&= \mathbf{A}\mathbf{R}_\alpha\mathbf{c}_\alpha + \mathbf{A}\mathbf{v}_\alpha - (\mathrm{vec}(\mathbf{M})^{\mathsf{T}} \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_\alpha\hat{\mathbf{c}} && \text{by Eq. (4.17).}
\end{aligned}$$

Thus, in Hyb_1, if $\hat{\mathbf{B}}_2\hat{\mathbf{c}} = \mathbf{0}$, then $\mathbf{A}\mathbf{R}_\alpha\mathbf{c}_\alpha + \mathbf{A}\mathbf{v}'_\alpha = \mathbf{0}$ if and only if $(\mathrm{vec}(\mathbf{M})^{\mathsf{T}} \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_\alpha\hat{\mathbf{c}} = \mathbf{A}\mathbf{R}_\alpha\mathbf{c}_\alpha + \mathbf{A}\mathbf{v}_\alpha$. Correspondingly, the output distribution of $\mathsf{Hyb}_1(\mathcal{A})$ is identical to the output distribution of $\mathsf{Hyb}_0(\mathcal{A})$. ☐

Lemma 4.28. *Suppose the* $\mathsf{KerDH}_{k,k+1}$ *assumption holds in* $\mathbb{G}_1$ *with respect to* $\mathsf{GroupGen}$. *Then, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that* $|\Pr[\mathsf{Hyb}_1(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_2(\mathcal{A}) = 1]| \le \mathsf{negl}(\lambda)$.

Proof. Suppose $|\Pr[\mathsf{Hyb}_1(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_2(\mathcal{A}) = 1]| \ge \varepsilon$ for some non-negligible ε. Suppose the output of $\mathcal{A}$ is $\big(\mathbf{M}, [\hat{\mathbf{c}}]_2, [\mathbf{c}_1]_2, [\mathbf{c}_2]_2, [\mathbf{v}_1]_2, [\mathbf{v}_2]_2\big)$ in an execution of Hyb_1 or Hyb_2. If the outputs of Hyb_1 and Hyb_2 differ, then it must be the case that that for some $\alpha \in \{1, 2\}$,

$$\mathbf{A}(\mathbf{R}_\alpha\mathbf{c}_\alpha + \mathbf{v}'_\alpha) = \mathbf{0} \quad \text{and} \quad \mathbf{R}_\alpha\mathbf{c}_\alpha + \mathbf{v}'_\alpha \neq \mathbf{0}. \tag{4.18}$$

In all other cases, the output in Hyb_1 and Hyb_2 is identical. We use $\mathcal{A}$ to construct an efficient adversary $\mathcal{B}$ for $\mathsf{KerDH}_{k,k+1}$:

1. On input the KerDH challenge $(\mathcal{G}, [\mathbf{A}]_1)$, algorithm $\mathcal{B}$ starts by running algorithm $\mathcal{A}$. Algorithm $\mathcal{A}$ outputs the input dimension ℓ, the locality set $S \subseteq [\ell] \times [\ell]$, and a pair $(j_1, j_2) \in S$.
2. Next, algorithm $\mathcal{B}$ samples full-rank matrices $\hat{\mathbf{B}}, \mathbf{B}_1, \mathbf{B}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k\times 2k}$ and defines $\hat{\mathbf{B}}^* = \hat{\mathbf{B}}^{-1}$, $\mathbf{B}_1^* = \mathbf{B}_1^{-1}$, $\mathbf{B}_2^* = \mathbf{B}_2^{-1}$. It parses the components of $\hat{\mathbf{B}}, \mathbf{B}_1, \mathbf{B}_2$ as in Eq. (4.2) and $\hat{\mathbf{B}}^*, \mathbf{B}_1^*, \mathbf{B}_2^*$ as in Eq. (4.3).
3. Algorithm $\mathcal{B}$ then constructs the encoding matrices $\hat{\mathbf{T}}, \mathbf{T}_1, \mathbf{T}_2$ as in Hyb_1 and Hyb_2:
   - **Type-I encodings:** Sample $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k\times\ell}$ and let $\hat{\mathbf{T}} = \hat{\mathbf{B}}_1^*\hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^*\hat{\mathbf{S}}_2\mathbf{P}_{j_1} \in \mathbb{Z}_p^{2k\times\ell}$.
   - **Type-II encodings:** For $\alpha \in \{1, 2\}$, sample $\mathbf{S}_{\alpha,1}, \mathbf{S}_{\alpha,2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k\times\ell}$ and let $\mathbf{T}_\alpha = \mathbf{B}_{\alpha,1}^*\mathbf{S}_{\alpha,1} + \mathbf{B}_{\alpha,2}^*\mathbf{S}_{\alpha,2}\mathbf{P}_{j_2}$.

   Let $\mathbf{T}_* = \mathbf{T}_1 \otimes \mathbf{T}_2$ and set $\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big)$.
4. For each $\alpha \in \{1, 2\}$, algorithm $\mathcal{B}$ samples $\mathbf{W}_{\mathsf{norm}}^{(\alpha)}, \mathbf{W}_{\mathsf{sf}}^{(\alpha)} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^2(k+1)\times k}$ and sets $\mathbf{W}_\alpha = \mathbf{W}_{\mathsf{norm}}^{(\alpha)}\hat{\mathbf{B}}_1 + \mathbf{W}_{\mathsf{sf}}^{(\alpha)}\hat{\mathbf{B}}_2$. It also samples $\mathbf{R}_\alpha \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(k+1)\times 2k}$. Then, for $\alpha \in \{1, 2\}$, it computes

   $$\mathbf{Z}_\alpha = \mathbf{W}_\alpha\hat{\mathbf{T}} - (\mathbf{P}_{\mathsf{lin}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{R}_\alpha\mathbf{T}_\alpha)),$$

   where $\mathbf{P}_{\mathsf{lin}} = \mathbf{P}_{\mathsf{lin}}^{(S)}$. The challenger gives the common reference string crs to $\mathcal{A}$ where

   $$\begin{aligned}
   \mathsf{crs} &= \Big(\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, \big\{(\mathbf{I}_{\ell^2} \otimes [\mathbf{A}]_1)\mathbf{W}_\alpha, [\mathbf{A}]_1\mathbf{R}_\alpha, [\mathbf{Z}_\alpha]_2\big\}_{\alpha\in\{1,2\}}\Big) \\
   &= \Big(\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, \big\{[(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_\alpha]_1, [\mathbf{A}\mathbf{R}_\alpha]_1, [\mathbf{Z}_\alpha]_2\big\}_{\alpha\in\{1,2\}}\Big).
   \end{aligned}$$

5. After algorithm $\mathcal{A}$ outputs $(\mathbf{M}, [\hat{\mathbf{c}}]_2, [\mathbf{c}_1]_2, [\mathbf{c}_2]_2, [\mathbf{v}_1]_2, [\mathbf{v}_2]_2)$ algorithm $\mathcal{B}$ computes for each $\alpha \in \{1, 2\}$,

$$[\mathbf{v}'_\alpha]_2 = [\mathbf{v}_\alpha]_2 - (\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})\mathbf{W}^{(\alpha)}_{\mathsf{norm}}\hat{\mathbf{B}}_1[\hat{\mathbf{c}}]_2.$$

It then checks if there exist $\alpha \in \{1, 2\}$ where

$$[\mathbf{A}\mathbf{R}_\alpha]_1[\mathbf{c}_\alpha]_2 + [\mathbf{A}]_1[\mathbf{v}'_\alpha]_2 = [\mathbf{0}]_T \quad \text{and} \quad \mathbf{R}_\alpha[\mathbf{c}_\alpha]_2 + [\mathbf{v}'_\alpha]_2 \neq [\mathbf{0}]_2.$$

If so, it outputs $\mathbf{R}_\alpha[\mathbf{c}_\alpha]_2 + [\mathbf{v}'_\alpha]_2 = [\mathbf{R}_\alpha\mathbf{c}_\alpha + \mathbf{v}'_\alpha]_2$.

Since the KerDH challenger samples $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(k+1)\times k}$, the common reference string crs constructed by $\mathcal{B}$ is distributed exactly as required in Hyb_1 and Hyb_2. By the above analysis, this means that with probability ε, algorithm $\mathcal{A}$ outputs $(\mathbf{M}, [\hat{\mathbf{c}}]_2, [\mathbf{c}_1]_2, [\mathbf{c}_2]_2, [\mathbf{v}_1]_2, [\mathbf{v}_2]_2)$ which satisfies Eq. (4.18). Correspondingly, algorithm $\mathcal{B}$ breaks KerDH with the same advantage ε. □

Lemma 4.29. $\Pr[\mathsf{Hyb}_2(\mathcal{A}) = 1] = \Pr[\mathsf{Hyb}_3(\mathcal{A}) = 1]$.

Proof. We argue that Hyb_2 and Hyb_3 are identically distributed. Since $\mathbf{W}^{(\alpha)}_{\mathsf{sf},1}$ and $\mathbf{R}_{\alpha,1}$ are uniform over their respective domains, it follows that $\mathbf{W}^{(\alpha)}_{\mathsf{sf}}$ and $\mathbf{R}_\alpha$ are identically distributed as in Hyb_2 and Hyb_3. To complete the proof, we show that the distribution of $\mathbf{Z}_\alpha$ in Hyb_3 is identical to that in Hyb_2. Suppose we construct $\mathbf{Z}_\alpha$ according to Eq. (4.15). Then,

$$\begin{aligned}
\mathbf{Z}_\alpha &= \mathbf{W}_\alpha\hat{\mathbf{T}} - (\mathbf{P}_{\mathsf{lin}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{R}_\alpha\mathbf{T}_\alpha)) \\
&= \left(\mathbf{W}^{(\alpha)}_{\mathsf{norm}}\hat{\mathbf{B}}_1 + \mathbf{W}^{(\alpha)}_{\mathsf{sf},1}\hat{\mathbf{B}}_2 + \left(\mathbf{W}^{(\alpha)}_{\mathsf{sf},2} \otimes \mathbf{a}^\perp\right)\hat{\mathbf{B}}_2\right)\hat{\mathbf{T}} - (\mathbf{P}_{\mathsf{lin}} \otimes \mathbf{I}_{k+1})\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left((\mathbf{R}_{\alpha,1} + \mathbf{a}^\perp\mathbf{r}^\intercal_{\alpha,2})\mathbf{T}_\alpha\right)\right) \\
&= \mathbf{Z}_{\alpha,1} + \left(\mathbf{W}^{(\alpha)}_{\mathsf{sf},2} \otimes \mathbf{a}^\perp\right)\hat{\mathbf{B}}_2\hat{\mathbf{T}} - (\mathbf{P}_{\mathsf{lin}} \otimes \mathbf{I}_{k+1})\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{a}^\perp\mathbf{r}^\intercal_{\alpha,2}\mathbf{T}_\alpha\right)\right).
\end{aligned} \tag{4.19}$$

We analyze the components of $\mathbf{Z}_\alpha$ in the subspace spanned by $\mathbf{a}^\perp$. First, using Eq. (3.3), we can write

$$\left(\mathbf{W}^{(\alpha)}_{\mathsf{sf},2} \otimes \mathbf{a}^\perp\right)\hat{\mathbf{B}}_2\hat{\mathbf{T}} = (\mathbf{I}_{\ell^2} \otimes \mathbf{a}^\perp)\mathbf{W}^{(\alpha)}_{\mathsf{sf},2}\hat{\mathbf{B}}_2\hat{\mathbf{T}} = (\mathbf{I}_{\ell^2} \otimes \mathbf{a}^\perp)\mathbf{W}^{(\alpha)}_{\mathsf{sf},2}\hat{\mathbf{B}}_2(\hat{\mathbf{B}}_1^*\hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^*\hat{\mathbf{S}}_2\mathbf{P}_{j_1}) = (\mathbf{I}_{\ell^2} \otimes \mathbf{a}^\perp)\mathbf{W}^{(\alpha)}_{\mathsf{sf},2}\hat{\mathbf{S}}_2\mathbf{P}_{j_1}. \tag{4.20}$$

For the remaining component in Eq. (4.19),

$$\begin{aligned}
\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{a}^\perp\mathbf{r}^\intercal_{\alpha,2}\mathbf{T}_\alpha\right) &= \mathbf{I}_\ell \otimes \left[\left(\mathbf{I}_\ell \otimes \mathbf{a}^\perp\mathbf{r}^\intercal_{\alpha,2}\right)\mathrm{vec}(\mathbf{T}_\alpha)\right] && \text{by Eq. (3.4)} \\
&= \mathbf{I}_\ell \otimes \left[\left(\mathbf{I}_\ell \otimes \mathbf{a}^\perp\right)\left(\mathbf{I}_\ell \otimes \mathbf{r}^\intercal_{\alpha,2}\right)\mathrm{vec}(\mathbf{T}_\alpha)\right] && \text{by Eq. (3.1)} \\
&= \mathbf{I}_\ell \otimes \left[\left(\mathbf{I}_\ell \otimes \mathbf{a}^\perp\right)\mathrm{vec}\left(\mathbf{r}^\intercal_{\alpha,2}\mathbf{T}_\alpha\right)\right] && \text{by Eq. (3.4)} \\
&= \left(\mathbf{I}_\ell \otimes (\mathbf{I}_\ell \otimes \mathbf{a}^\perp)\right)\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{r}^\intercal_{\alpha,2}\mathbf{T}_\alpha\right)\right) && \text{by Eq. (3.1)} \\
&= \left(\mathbf{I}_{\ell^2} \otimes \mathbf{a}^\perp\right)\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{r}^\intercal_{\alpha,2}\mathbf{T}_\alpha\right)\right).
\end{aligned}$$

Finally, by Eq. (3.3),

$$\begin{aligned}
(\mathbf{P}_{\mathsf{lin}} \otimes \mathbf{I}_{k+1})\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{a}^\perp\mathbf{r}^\intercal_{\alpha,2}\mathbf{T}_\alpha\right)\right) &= (\mathbf{P}_{\mathsf{lin}} \otimes \mathbf{I}_{k+1})\left(\mathbf{I}_{\ell^2} \otimes \mathbf{a}^\perp\right)\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{r}^\intercal_{\alpha,2}\mathbf{T}_\alpha\right)\right) \\
&= (\mathbf{I}_{\ell^2} \otimes \mathbf{a}^\perp)\mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{r}^\intercal_{\alpha,2}\mathbf{T}_\alpha\right)\right).
\end{aligned} \tag{4.21}$$

Combining Eq. (4.21), (4.20), and (4.19), we have

$$\mathbf{Z}_\alpha = \mathbf{Z}_{\alpha,1} + \left(\mathbf{I}_{\ell^2} \otimes \mathbf{a}^\perp\right)\left(\mathbf{W}^{(\alpha)}_{\mathsf{sf},2}\hat{\mathbf{S}}_2\mathbf{P}_{j_1} - \mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{r}^\intercal_{\alpha,2}\mathbf{T}_\alpha\right)\right)\right) = \mathbf{Z}_{\alpha,1} + (\mathbf{I}_{\ell^2} \otimes \mathbf{a}^\perp)\mathbf{Z}_{\alpha,2},$$

which is precisely how the challenger constructs $\mathbf{Z}_\alpha$ in Hyb_3. □

Lemma 4.30. $\Pr[\mathsf{Hyb}_3(\mathcal{A}) = 1] = \Pr[\mathsf{Hyb}_4(\mathcal{A}) = 1]$.

Proof. The distribution of crs in the two experiments are identical. In particular, in Hyb_3, for $\alpha \in \{1,2\}$,

$$\begin{aligned}(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_\alpha &= (\mathbf{I}_{\ell^2} \otimes \mathbf{A})\big(\mathbf{W}^{(\alpha)}_{\mathsf{norm}}\hat{\mathbf{B}}_1 + \mathbf{W}^{(\alpha)}_{\mathsf{sf}}\hat{\mathbf{B}}_2\big) \\ &= (\mathbf{I}_{\ell^2} \otimes \mathbf{A})\big(\mathbf{W}^{(\alpha)}_{\mathsf{norm}}\hat{\mathbf{B}}_1 + \mathbf{W}^{(\alpha)}_{\mathsf{sf},1}\hat{\mathbf{B}}_2 + \big(\mathbf{W}^{(\alpha)}_{\mathsf{sf},2} \otimes \mathbf{a}^\perp\big)\hat{\mathbf{B}}_2\big) \\ &= (\mathbf{I}_{\ell^2} \otimes \mathbf{A})\big(\mathbf{W}^{(\alpha)}_{\mathsf{norm}}\hat{\mathbf{B}}_1 + \mathbf{W}^{(\alpha)}_{\mathsf{sf},1}\hat{\mathbf{B}}_2\big)\end{aligned}$$

since $\mathbf{A}\mathbf{a}^\perp = \mathbf{0}$. Similarly,

$$\mathbf{A}\mathbf{R}_\alpha = \mathbf{A}\big(\mathbf{R}_{\alpha,1} + \mathbf{a}^\perp \mathbf{r}_{\alpha,2}^{\mathsf{T}}\big) = \mathbf{A}\mathbf{R}_{\alpha,1}.$$

This coincides with the distribution of crs in Hyb_4. □

Lemma 4.31. *Suppose the* $\mathsf{MDDH}_{k,\ell,2\ell^2}$ *assumption holds in* $\mathbb{G}_2$ *with respect to* $\mathsf{GroupGen}$. *Then, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that* $|\Pr[\mathsf{Hyb}_4(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_5(\mathcal{A}) = 1]| = \mathsf{negl}(\lambda)$.

Proof. Suppose $|\Pr[\mathsf{Hyb}_4(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_5(\mathcal{A}) = 1]| \geq \varepsilon$ for some non-negligible ε. We use $\mathcal{A}$ to construct an efficient adversary $\mathcal{B}$ for $\mathsf{MDDH}_{k,\ell,\ell^2}$:

1. On input the MDDH challenge $(\mathcal{G}, [\hat{\mathbf{S}}_2]_2, [\mathbf{V}]_2)$, algorithm $\mathcal{A}$ starts by parsing $[\mathbf{V}_2] = \left[\begin{smallmatrix}\mathbf{V}_1 \\ \mathbf{V}_2\end{smallmatrix}\right]_2$, where $\mathbf{V}_1, \mathbf{V}_2 \in \mathbb{Z}_p^{\ell^2 \times \ell}$. Then, it samples full-rank matrices $\hat{\mathbf{B}}, \mathbf{B}_1, \mathbf{B}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times 2k}$ and defines $\hat{\mathbf{B}}^* = \hat{\mathbf{B}}^{-1}$, $\mathbf{B}_1^* = \mathbf{B}_1^{-1}$, $\mathbf{B}_2^* = \mathbf{B}_2^{-1}$. It parses $\hat{\mathbf{B}}, \mathbf{B}_1, \mathbf{B}_2$ as in Eq. (4.2) and $\hat{\mathbf{B}}^*, \mathbf{B}_1^*, \mathbf{B}_2^*$ as in Eq. (4.3).
2. Algorithm $\mathcal{A}$ constructs the Type-I and Type-II encoding matrices $\hat{\mathbf{T}}, \mathbf{T}_1, \mathbf{T}_2$ as follows:
   - **Type-I encodings:** Sample $\hat{\mathbf{S}}_1 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$ and let $[\hat{\mathbf{T}}]_2 = \hat{\mathbf{B}}_1^*\hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^*[\hat{\mathbf{S}}_2]_2\mathbf{P}_{j_1} \in \mathbb{Z}_p^{2k \times \ell}$.
   - **Type-II encodings:** For $\alpha \in \{1,2\}$, sample $\mathbf{S}_{\alpha,1}, \mathbf{S}_{\alpha,2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$. Let $\mathbf{T}_\alpha = \mathbf{B}_{\alpha,1}^*\mathbf{S}_{\alpha,1} + \mathbf{B}_{\alpha,2}^*\mathbf{S}_{\alpha,2}\mathbf{P}_{j_2} \in \mathbb{Z}_p^{2k \times \ell}$.

   Let $\mathbf{T}_* = \mathbf{T}_1 \otimes \mathbf{T}_2$ and set $\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big)$.
3. Sample $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times (k+1)}$ and a random nonzero vector $\mathbf{a}^\perp \in \mathbb{Z}_p^{k+1}$ in the kernel of $\mathbf{A}$.
4. For $\alpha \in \{1,2\}$, sample $\mathbf{W}^{(\alpha)}_{\mathsf{norm}} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^2(k+1) \times k}$, $\mathbf{W}^{(\alpha)}_{\mathsf{sf},1} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^2(k+1) \times k}$, $\mathbf{R}_{\alpha,1} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(k+1) \times 2k}$, and $\mathbf{r}_{\alpha,2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k}$. Set $\mathbf{R}_\alpha = \mathbf{R}_{\alpha,1} + \mathbf{a}^\perp\mathbf{r}_{\alpha,2}^{\mathsf{T}}$. It then computes

   $$\begin{aligned}[\mathbf{Z}_{\alpha,1}]_2 &= \big(\mathbf{W}^{(\alpha)}_{\mathsf{norm}}\hat{\mathbf{B}}_1 + \mathbf{W}^{(\alpha)}_{\mathsf{sf},1}\hat{\mathbf{B}}_2\big)[\hat{\mathbf{T}}]_2 - (\mathbf{P}_{\mathsf{lin}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{R}_{\alpha,1}\mathbf{T}_\alpha)) \\ [\mathbf{Z}_{\alpha,2}]_2 &= [\mathbf{V}_\alpha]_2\mathbf{P}_{j_1} - \mathbf{P}_{\mathsf{lin}}\big(\mathbf{I}_\ell \otimes \mathrm{vec}(\mathbf{r}_{2,\alpha}^{\mathsf{T}}\mathbf{T}_\alpha)\big),\end{aligned}$$

   and $[\mathbf{Z}_\alpha]_2 = [\mathbf{Z}_{\alpha,1}]_2 + (\mathbf{I}_{\ell^2} \otimes \mathbf{a}^\perp)[\mathbf{Z}_{\alpha,2}]_2$.
5. Finally, algorithm $\mathcal{B}$ gives crs to $\mathcal{A}$ where

   $$\mathsf{crs} = \left(\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, \left\{\left[(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\big(\mathbf{W}^{(\alpha)}_{\mathsf{norm}}\hat{\mathbf{B}}_1 + \mathbf{W}^{(\alpha)}_{\mathsf{sf},1}\hat{\mathbf{B}}_2\big)\right]_1, [\mathbf{A}\mathbf{R}_{\alpha,1}]_1, [\mathbf{Z}_\alpha]_2\right\}\right).$$

6. After algorithm $\mathcal{A}$ outputs $\big(\mathbf{M}, [\hat{\mathbf{c}}]_2, [\mathbf{c}_1]_2, [\mathbf{c}_2]_2, [\mathbf{v}_1]_2, [\mathbf{v}_2]_2\big)$, algorithm $\mathcal{B}$ computes for each $\alpha \in \{1,2\}$,

   $$[\mathbf{v}'_\alpha]_2 = [\mathbf{v}_\alpha]_2 - (\mathrm{vec}(\mathbf{M})^{\mathsf{T}} \otimes \mathbf{I}_{k+1})\mathbf{W}^{(\alpha)}_{\mathsf{norm}}\hat{\mathbf{B}}_1[\hat{\mathbf{c}}]_2$$

   Then, it outputs 1 if the following hold:

   $$\hat{\mathbf{B}}_2[\hat{\mathbf{c}}]_2 = [\mathbf{0}]_2 \quad \text{and} \quad \forall \alpha \in \{1,2\} : \mathbf{R}_\alpha[\mathbf{c}_\alpha]_2 + [\mathbf{v}'_\alpha]_2 = [\mathbf{0}]_2 \quad \text{and} \quad \mathbf{B}_{1,2}[\mathbf{c}_1]_2 \neq [\mathbf{0}]_2 \text{ or } \mathbf{B}_{2,2}[\mathbf{c}_2]_2 \neq [\mathbf{0}]_2.$$

By definition, the MDDH challenger samples $\hat{\mathbf{S}}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$. Thus, algorithm $\mathcal{B}$ perfectly simulates the distribution of every component other than $[\mathbf{Z}_\alpha]_2$ in the common reference string according to the specification of Hyb_4 and Hyb_5. Thus it suffices to consider the distribution of $\mathbf{Z}_\alpha$ in the two cases:

- Suppose $\mathbf{V}_\alpha = \mathbf{W}_{\mathsf{sf},2}^{(\alpha)} \hat{\mathbf{S}}_2$ where the challenger samples $\mathbf{W}_{\mathsf{sf},2}^{(\alpha)} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^2 \times k}$. Then algorithm $\mathcal{B}$ perfectly simulates the distribution of crs in Hyb_4. In this case, algorithm $\mathcal{B}$ outputs 1 with probability $\Pr[\mathsf{Hyb}_4(\mathcal{A}) = 1]$.

- Suppose $\mathbf{V} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2\ell^2 \times \ell}$, in which case $\mathbf{V}_1, \mathbf{V}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^2 \times \ell}$. This corresponds to the distribution of $\mathbf{Z}_\alpha$ in Hyb_5, so in this case, algorithm $\mathcal{B}$ outputs 1 with probability $\Pr[\mathsf{Hyb}_5(\mathcal{A}) = 1]$.

We conclude that the distinguishing advantage of $\mathcal{B}$ is exactly ε and the claim follows. □

Lemma 4.32. $\Pr[\mathsf{Hyb}_5(\mathcal{A}) = 1] = \Pr[\mathsf{Hyb}_6(\mathcal{A}) = 1]$.

Proof. For each $\alpha \in \{1, 2\}$, $\mathbf{B}_\alpha = \begin{bmatrix} \mathbf{B}_{\alpha,1} \\ \mathbf{B}_{\alpha,2} \end{bmatrix}$ is a basis for $\mathbb{Z}_p^{2k}$, the distribution of $\mathbf{r}_{\alpha,2}$ in Hyb_6 is uniform over $\mathbb{Z}_p^{2k}$, which is identical to the distribution of $\mathbf{r}_{\alpha,2}$ in Hyb_5. It suffices to argue that $\mathbf{Z}_{\alpha,2}$ is correctly distributed. This follows by the fact that $\mathbf{B}_\alpha \mathbf{B}_\alpha^* = \mathbf{I}_{2k}$ and the fact that $\mathbf{T}_\alpha = \mathbf{B}_{\alpha,1}^* \mathbf{S}_{\alpha,1} + \mathbf{B}_{\alpha,2}^* \mathbf{S}_{\alpha,2} \mathbf{P}_{j_2}$. In particular, we can write

$$\begin{aligned}
\mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{r}_{\alpha,2}^{\mathsf{T}} \mathbf{T}_\alpha\right)\right) &= \mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\left(\mathbf{r}_{\alpha,2,\mathsf{norm}}^{\mathsf{T}} \mathbf{B}_{\alpha,1} + \mathbf{r}_{\alpha,2,\mathsf{sf}}^{\mathsf{T}} \mathbf{B}_{\alpha,2}\right)\left(\mathbf{B}_{\alpha,1}^* \mathbf{S}_{\alpha,1} + \mathbf{B}_{\alpha,2}^* \mathbf{S}_{\alpha,2} \mathbf{P}_{j_2}\right)\right)\right) \\
&= \mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{r}_{\alpha,2,\mathsf{norm}}^{\mathsf{T}} \mathbf{S}_{\alpha,1} + \mathbf{r}_{\alpha,2,\mathsf{sf}}^{\mathsf{T}} \mathbf{S}_{\alpha,2} \mathbf{P}_{j_2}\right)\right) \\
&= \mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{r}_{\alpha,2,\mathsf{norm}}^{\mathsf{T}} \mathbf{S}_{\alpha,1}\right)\right) + \mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{r}_{\alpha,2,\mathsf{sf}}^{\mathsf{T}} \mathbf{S}_{\alpha,2} \mathbf{P}_{j_2}\right)\right),
\end{aligned}$$

which matches the distribution in Hyb_6. □

Lemma 4.33. $\Pr[\mathsf{Hyb}_6(\mathcal{A}) = 1] = \Pr[\mathsf{Hyb}_7(\mathcal{A}) = 1]$.

Proof. The claim follows by properties of the projection matrix (Lemma 4.22). Specifically, we will show that for $\alpha \in \{1, 2\}$, the following two distributions are identically distributed over the choice of $\mathbf{U}$:

$$\left\{\mathbf{U}_\alpha \mathbf{P}_{j_1} - \mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{r}_{\alpha,2,\mathsf{sf}}^{\mathsf{T}} \mathbf{S}_{\alpha,2} \mathbf{P}_{j_2}\right)\right) : \mathbf{U} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^2 \times \ell}\right\} \equiv \left\{\mathbf{U}_\alpha \mathbf{P}_{j_1} : \mathbf{U}_\alpha \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^2 \times \ell}\right\}. \tag{4.22}$$

Since $(j_1, j_2) \in S$ and moreover, $\mathbf{P}_{\mathsf{lin}} = \mathbf{P}_{\mathsf{lin}}^{(S)}$, we can appeal to Lemma 4.22 (applied to the vector $\mathbf{r}_{\alpha,2,\mathsf{sf}}^{\mathsf{T}} \mathbf{S}_{\alpha,2}$) to conclude that

$$\mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{r}_{\alpha,2,\mathsf{sf}}^{\mathsf{T}} \mathbf{S}_{\alpha,2} \mathbf{P}_{j_2}\right)\right)\left(\mathbf{I}_\ell - \mathbf{P}_{j_1}\right) = \mathbf{0}.$$

Now, we can write

$$\begin{aligned}
\mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{r}_{\alpha,2,\mathsf{sf}}^{\mathsf{T}} \mathbf{S}_{\alpha,2} \mathbf{P}_{j_2}\right)\right) &= \mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{r}_{\alpha,2,\mathsf{sf}}^{\mathsf{T}} \mathbf{S}_{\alpha,2} \mathbf{P}_{j,2}\right)\right)\left(\mathbf{P}_{j_1} + \mathbf{I}_\ell - \mathbf{P}_{j_1}\right) \\
&= \mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{r}_{\alpha,2,\mathsf{sf}}^{\mathsf{T}} \mathbf{S}_{\alpha,2} \mathbf{P}_{j_2}\right)\right) \mathbf{P}_{j_1}.
\end{aligned}$$

This means that

$$\mathbf{U}_\alpha \mathbf{P}_{j_1} - \mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{r}_{\alpha,2,\mathsf{sf}}^{\mathsf{T}} \mathbf{S}_{\alpha,2} \mathbf{P}_{j_2}\right)\right) = \left(\mathbf{U}_\alpha - \mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{r}_{\alpha,2,\mathsf{sf}}^{\mathsf{T}} \mathbf{S}_{\alpha,2} \mathbf{P}_{j_2}\right)\right)\right) \mathbf{P}_{j_1}. \tag{4.23}$$

Since $\mathbf{U}_\alpha$ is uniform over $\mathbb{Z}_p^{\ell^2 \times \ell}$ and independent of $\mathbf{P}_{\mathsf{lin}}$, $\mathbf{r}_{\alpha,2,\mathsf{sf}}$, $\mathbf{S}_{\alpha,2}$, and $\mathbf{P}_{j_2}$, it follows that

$$\left\{\mathbf{U}_\alpha - \mathbf{P}_{\mathsf{lin}}\left(\mathbf{I}_\ell \otimes \mathrm{vec}\left(\mathbf{r}_{\alpha,2,\mathsf{sf}}^{\mathsf{T}} \mathbf{S}_{\alpha,2} \mathbf{P}_{j_2}\right)\right) : \mathbf{U}_\alpha \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^2 \times \ell}\right\} \equiv \left\{\mathbf{U}_\alpha : \mathbf{U}_\alpha \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^2 \times \ell}\right\}. \tag{4.24}$$

Eq. (4.22) now follows by combining Eqs. (4.23) and (4.24). □

Lemma 4.34. *There exists a negligible function* $\mathsf{negl}(\cdot)$ *such that* $\Pr[\mathsf{Hyb}_7(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$.

Proof. In Hyb_7, the components of crs are *independent* of the vector $\mathbf{r}_{\alpha,2,\mathsf{sf}}$ for each $\alpha \in \{1, 2\}$. This means the challenger in Hyb_7 can defer the sampling of $\mathbf{r}_{\alpha,2,\mathsf{sf}}$ until *after* the adversary outputs $(\mathbf{M}, [\hat{\mathbf{c}}]_2, [\mathbf{c}_1]_2, [\mathbf{c}_2]_2, [\mathbf{v}_1]_2, [\mathbf{v}_2]_2)$. For the challenger to output 1 in Hyb_7, it must be the case that there exists $\alpha \in \{1, 2\}$ where

$$\mathbf{R}_\alpha \mathbf{c}_\alpha + \mathbf{v}'_\alpha = \mathbf{0} \quad \text{and} \quad \mathbf{B}_{\alpha,2} \mathbf{c}_\alpha \neq \mathbf{0},$$

where $\mathbf{v}'_\alpha = \mathbf{v}_\alpha - (\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})\mathbf{W}^{(\alpha)}_{\mathsf{norm}}\hat{\mathbf{B}}_1\hat{\mathbf{c}}$. We argue that when $\mathbf{B}_{\alpha,2}\mathbf{c}_\alpha \neq \mathbf{0}$, the probability that $\mathbf{R}_\alpha\mathbf{c}_\alpha + \mathbf{v}'_\alpha = \mathbf{0}$ is negligible when taken over the choice of $\mathbf{r}_{\alpha,2,\mathsf{sf}}$. Since

$$\mathbf{R}_\alpha = \mathbf{R}_{\alpha,1} + \mathbf{a}^\perp \mathbf{r}^\intercal_{\alpha,2} = \mathbf{R}_{\alpha,1} + \mathbf{a}^\perp \mathbf{r}_{\alpha,2,\mathsf{norm}}\mathbf{B}_{\alpha,1} + \mathbf{a}^\perp \mathbf{r}^\intercal_{\alpha,2,\mathsf{sf}}\mathbf{B}_{\alpha,2},$$

the equation $\mathbf{R}_\alpha\mathbf{c}_\alpha + \mathbf{v}'_\alpha = \mathbf{0}$ holds only if

$$\mathbf{a}^\perp \cdot \mathbf{r}^\intercal_{\alpha,2,\mathsf{sf}}\mathbf{B}_{\alpha,2}\mathbf{c}_\alpha = -\mathbf{v}'_\alpha - \mathbf{R}_{\alpha,1}\mathbf{c}_\alpha - \mathbf{a}^\perp \cdot \mathbf{r}^\intercal_{\alpha,2,\mathsf{norm}}\mathbf{B}_{\alpha,1}\mathbf{c}_\alpha \in \mathbb{Z}_p^{k+1}.$$

Since $\mathbf{B}_{\alpha,2}\mathbf{c}_\alpha \neq \mathbf{0}$ and $\mathbf{r}_{\alpha,2,\mathsf{sf}} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$, the distribution of $\mathbf{r}^\intercal_{\alpha,2,\mathsf{sf}}\mathbf{B}_{\alpha,2}\mathbf{c}_\alpha$ is uniform over $\mathbb{Z}_p$. Finally, since $\mathbf{a}^\perp \neq \mathbf{0}$ and the challenger samples $\mathbf{r}_{\alpha,2,\mathsf{sf}} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$ *after* all other quantities have been fixed, we conclude that

$$\Pr\left[\mathbf{a}^\perp \cdot \mathbf{r}^\intercal_{\alpha,2,\mathsf{sf}}\mathbf{B}_{\alpha,2}\mathbf{c}_\alpha = -\mathbf{v}'_\alpha - \mathbf{R}_{\alpha,1}\mathbf{c}_\alpha - \mathbf{a}^\perp \cdot \mathbf{r}^\intercal_{\alpha,2,\mathsf{norm}}\mathbf{B}_{\alpha,1}\mathbf{c}_\alpha : \mathbf{r}_{\alpha,2,\mathsf{sf}} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k\right] \leq \frac{1}{p} = \mathsf{negl}(\lambda). \qquad \square$$

By Lemmas 4.27 to 4.34, we conclude that $\Pr[\mathsf{Hyb}_0(\mathcal{A}) = 1] \leq \mathsf{negl}(\lambda)$. This means that Construction 4.23 satisfies homogeneous chain binding for linear functions. Finally, since the vector dimension $\ell = \mathsf{poly}(\lambda)$, the k-Lin assumption in $\mathbb{G}_2$ implies the $\mathsf{MDDH}_{k,\ell,\ell^2}$ assumption in $\mathbb{G}_2$ (Remark 3.8); similarly, the k-KerLin assumption in $\mathbb{G}_1$ implies the $\mathsf{KerDH}_{k,k+1}$ assumption in $\mathbb{G}_1$. Theorem 4.25 now follows from Lemma 4.26. $\square$

4.4 Proving Quadratic Relations on Committed Values

The final proof system we require is a way to argue that a Type-I commitment is consistent with a quadratic function applied to a Type-II commitment. Specifically, we describe a succinct proof system for statements of the following form: for a quadratic function $f\colon \mathbb{Z}_p^\ell \to \mathbb{Z}_p^\ell$,

if σ_2 is a Type-II commitment to a vector $\mathbf{x} \in \mathbb{Z}_p^\ell$, then σ_1 is a Type-I commitment to a vector $\mathbf{y} = f(\mathbf{x})$.

In contrast to the proof system for linear functions from Section 4.3, the inputs to this proof system are *Type-II* commitments while the outputs are *Type-I* commitments. Similar to Section 4.3, we require chain binding for *local* quadratic functions. We give the formal syntax and security requirement below:

Definition 4.35 (Projective Chainable Commitments for Quadratic Functions)**.** Let $\mathsf{FC}_{\mathsf{base}} = (\mathsf{SetupBase}, \mathsf{SetupSF}, \mathsf{Commit}^{(1)}, \mathsf{Commit}^{(2)}, \mathsf{Project}^{(1)}, \mathsf{Project}^{(2)})$ be a projective commitment scheme. In the following description, we represent (homogeneous) quadratic functions $f(\mathbf{x}) \coloneqq \mathbf{M}(\mathbf{x} \otimes \mathbf{x})$ by a matrix $\mathbf{M}$. A chainable proof system for quadratic functions is a triple of efficient algorithms $\mathsf{FC}_{\mathsf{quad}} = (\mathsf{SetupQuad}, \mathsf{OpenQuad}, \mathsf{VerifyQuad})$ with the following properties:

- $\mathsf{SetupQuad}(\mathsf{crs}_{\mathsf{base}}, S) \to \mathsf{crs}$: On input the common reference string $\mathsf{crs}_{\mathsf{base}}$ (which defines the input space $\mathcal{R}^\ell$) and a locality set $S \subseteq [\ell] \times [\ell]$, the setup algorithm outputs a common reference string crs.
- $\mathsf{OpenQuad}(\mathsf{crs}, \mathbf{x}, \mathbf{M}) \to \pi$: On input a common reference string crs, an input vector $\mathbf{x} \in \mathcal{R}^\ell$, and a homogeneous quadratic function $\mathbf{M} \in \mathcal{R}^{\ell \times \ell^2}$, the opening algorithm outputs a proof π.
- $\mathsf{VerifyQuad}(\mathsf{crs}, \sigma_2, \mathbf{M}, \sigma_1, \pi) \to b$: On input the common reference string crs, a Type-II commitment σ_2, a linear function $\mathbf{M} \in \mathcal{R}^{\ell \times \ell}$, a Type-I commitment σ_1, and a proof π, the verification algorithm outputs a bit $b \in \{0, 1\}$.

The proof system should satisfy the following two properties:

- **Correctness:** For all security parameters $\lambda \in \mathbb{N}$, all vector lengths $\ell \in \mathbb{N}$, all locality sets $S \subseteq [\ell] \times [\ell]$, all $\mathsf{crs}_{\mathsf{base}}$ in the support of $\mathsf{SetupBase}(1^\lambda, 1^\ell)$, all vectors $\mathbf{x} \in \mathcal{R}^\ell$ (where $\mathcal{R}^\ell$ is the message space associated with $\mathsf{crs}_{\mathsf{base}}$), and all S-local homogeneous quadratic functions $\mathbf{M} \in \mathcal{R}^{\ell \times \ell^2}$,

$$\Pr\left[\mathsf{VerifyQuad}(\mathsf{crs}, \sigma_2, \mathbf{M}, \sigma_1, \pi) = 1 : \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{SetupQuad}(\mathsf{crs}_{\mathsf{base}}, S) \\ \sigma_2 \leftarrow \mathsf{Commit}^{(2)}(\mathsf{crs}_{\mathsf{base}}, \mathbf{x}) \\ \sigma_1 \leftarrow \mathsf{Commit}^{(1)}(\mathsf{crs}_{\mathsf{base}}, \mathbf{M}(\mathbf{x} \otimes \mathbf{x})) \\ \pi \leftarrow \mathsf{OpenQuad}(\mathsf{crs}, \mathbf{x}, \mathbf{M}) \end{array}\right] = 1.$$

- **Chain binding for quadratic functions:** For a security parameter λ and an adversary $\mathcal{A}$, we define the chain binding for quadratic functions security experiment as follows:
  1. On input the security parameter λ, the adversary outputs the dimension 1^ℓ, a locality set $S \subseteq [\ell] \times [\ell]$ and a pair $(j_1, j_2) \in S$. Note here that j_1 denotes the length of the prefix for the *input* (i.e., a *Type-II* index) and j_2 denotes the length of the prefix for the *output* (i.e., a *Type-I* index).
  2. The challenger samples $(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^\lambda, 1^\ell, j_2, j_1)$ and $\mathsf{crs} \leftarrow \mathsf{SetupQuad}(\mathsf{crs}_{\mathsf{base}}, S)$. It gives $(\mathsf{crs}_{\mathsf{base}}, \mathsf{crs})$ to $\mathcal{A}$.
  3. The adversary outputs an S-local quadratic function $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell^2}$, two Type-II commitments (σ_2, σ_2'), two Type-I commitments (σ_1, σ_1'), and two openings π, π'.
  4. The challenger outputs $b = 1$ if all the following properties hold:
     - **Matching inputs:** $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) = \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2')$.
     - **Mismatching outputs:** $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) \neq \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$.
     - **Validity of openings:** $\mathsf{VerifyQuad}(\mathsf{crs}, \sigma_2, \mathbf{M}, \sigma_1, \pi) = 1 = \mathsf{VerifyQuad}(\mathsf{crs}, \sigma_2', \mathbf{M}, \sigma_1', \pi')$.

     Otherwise, the challenger outputs $b = 0$.

  We say that $\mathsf{FC}_{\mathsf{quad}}$ satisfies chain binding for quadratic functions if for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that $\Pr[b = 1] = \mathsf{negl}(\lambda)$ in the chain binding for quadratic functions security game.

Constructing projective chainable commitments. Similar to the construction of chainable commitments for linear functions from Section 4.3, we start by defining the projection matrix for a local quadratic function; this is the analog of Definition 4.21. We then prove the analog of Lemma 4.22 for the case of (homogeneous) quadratic functions.

Definition 4.36 (Projection Matrix for a Local Quadratic Function)**.** Let $\ell \in \mathbb{N}$ be an input length. For indices $j_1, j_2 \in [\ell]$, we define the projection matrix $\mathbf{P}_{\mathsf{quad}}^{(j_1, j_2)}$ to be

$$\mathbf{P}_{\mathsf{quad}}^{(j_1, j_2)} := \mathbf{I}_{\ell^3} - \big(\mathbf{I}_{\ell^2} - \big(\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1}\big)\big) \otimes \mathbf{P}_{j_2} \in \{0,1\}^{\ell^3 \times \ell^3},$$

where $\mathbf{P}_{j_1}, \mathbf{P}_{j,2} \in \{0,1\}^{\ell \times \ell}$ are the projection matrices from Definition 4.1. For a locality set $S \subseteq [\ell] \times [\ell]$, we define the projection matrix for S to be

$$\mathbf{P}_{\mathsf{quad}}^{(S)} := \prod_{(j_1, j_2) \in S} \mathbf{P}_{\mathsf{quad}}^{(j_1, j_2)} \in \{0,1\}^{\ell^3 \times \ell^3}. \tag{4.25}$$

Lemma 4.37 (Projection Matrix for a Local Quadratic Function)**.** *Let $\ell \in \mathbb{N}$ be an input length and $S \subseteq [\ell] \times [\ell]$ be a locality set. Suppose $f \colon \mathbb{Z}_p^\ell \to \mathbb{Z}_p^\ell$ is an S-local homogeneous quadratic function $f(\mathbf{x}) := \mathbf{M}(\mathbf{x} \otimes \mathbf{x})$ where $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell^2}$. Let $\mathbf{P}_{\mathsf{quad}} := \mathbf{P}_{\mathsf{quad}}^{((S))}$ be the projection matrix associated with S from Definition 4.36. Then the following properties hold:*

- $\mathrm{vec}(\mathbf{M})^\intercal \mathbf{P}_{\mathsf{quad}} = \mathrm{vec}(\mathbf{M})^\intercal$.
- *For all $(j_1, j_2) \in S$ and all vectors $\mathbf{r} \in \mathbb{Z}_p^\ell$, $\mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}^\intercal \mathbf{P}_{j_2})\big)\big(\mathbf{I}_{\ell^2} - (\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1})\big) = \mathbf{0}$, where $\mathbf{P}_{j_1}, \mathbf{P}_{j_2} \in \{0,1\}^{\ell \times \ell}$ are the projection matrices from Definition 4.1.*

Proof. The proof follows a similar strategy as the proof of Lemma 4.22. We show each claim separately:

- For the first claim, we start by observing that if f is (j_1, j_2)-local, then the first j_2 components of $\mathbf{M}(\mathbf{e}_i \otimes \mathbf{e}_{i'})$ are zero whenever $i > j_1$ or $i' > j_1$, where $\mathbf{e}_i \in \{0,1\}^\ell$ is the i^{th} basis vector. This means that

$$\mathbf{P}_{j_2} \cdot \mathbf{M} \cdot \big(\mathbf{I}_{\ell^2} - \big(\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1}\big)\big) = \mathbf{0}, \tag{4.26}$$

Then, for all $(j_1, j_2) \in S$,

$$\begin{aligned}
\mathrm{vec}(\mathbf{M})^\mathsf{T}\mathbf{P}_{\mathrm{quad}}^{(j_1,j_2)} &= \mathrm{vec}(\mathbf{M})^\mathsf{T}\left[\mathbf{I}_{\ell^3} - \left(\mathbf{I}_{\ell^2} - (\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1})\right) \otimes \mathbf{P}_{j_2}\right] \\
&= \mathrm{vec}(\mathbf{M})^\mathsf{T} - \mathrm{vec}(\mathbf{M})^\mathsf{T}\left(\left(\mathbf{I}_{\ell^2} - (\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1})\right) \otimes \mathbf{P}_{j_2}\right) \\
&= \mathrm{vec}(\mathbf{M})^\mathsf{T} - \mathrm{vec}\left(\mathbf{P}_{j_2}^\mathsf{T}\mathbf{M}\left(\mathbf{I}_{\ell^2} - (\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1})\right)\right) && \text{by Eq. (3.4)} \\
&= \mathrm{vec}(\mathbf{M})^\mathsf{T} && \text{by Eq. (4.26) and since } \mathbf{P}_{j_2} = \mathbf{P}_{j_2}^\mathsf{T}.
\end{aligned}$$

- For the second claim, take any $(j_1, j_2) \in S$. Let $\mathbf{Q}_{j_1} = \mathbf{I}_{\ell^2} - (\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1}) \in \{0,1\}^{\ell^2 \times \ell^2}$. Then,

$$(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}^\mathsf{T}\mathbf{P}_{j_2}))\mathbf{Q}_{j_1} = (\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}^\mathsf{T}\mathbf{P}_{j_2}))(\mathbf{Q}_{j_1} \otimes 1) = \mathbf{Q}_{j_1} \otimes \mathrm{vec}(\mathbf{r}^\mathsf{T}\mathbf{P}_{j_2}).$$

Since $\mathbf{Q}_{j_1}$ is a diagonal matrix and its entries are in $\{0, 1\}$, it follows that $\mathbf{Q}_{j_1}^2 = \mathbf{Q}_{j_1}$. Similarly, since $\mathbf{P}_{j_2}$ is a diagonal matrix with entries in $\{0, 1\}$, it follows that $\mathbf{P}_{j_2}\mathbf{P}_{j_2}^\mathsf{T} = \mathbf{P}_{j_2}^2 = \mathbf{P}_{j_2}$. Then,

$$\begin{aligned}
(\mathbf{Q}_{j_1} \otimes \mathbf{P}_{j_2})(\mathbf{Q}_{j_1} \otimes \mathrm{vec}(\mathbf{r}^\mathsf{T}\mathbf{P}_{j_2})) &= \mathbf{Q}_{j_1}^2 \otimes \left((\mathbf{P}_{j_2} \otimes 1) \cdot \mathrm{vec}(\mathbf{r}^\mathsf{T}\mathbf{P}_{j_2})\right) && \text{by Eq. (3.1)} \\
&= \mathbf{Q}_{j_1} \otimes \mathrm{vec}(\mathbf{r}^\mathsf{T}\mathbf{P}_{j_2}\mathbf{P}_{j_2}^\mathsf{T}) && \text{by Eq. (3.4)} \\
&= \mathbf{Q}_{j_1} \otimes \mathrm{vec}(\mathbf{r}^\mathsf{T}\mathbf{P}_{j_2}) && \text{since } \mathbf{P}_{j_2}\mathbf{P}_{j_2}^\mathsf{T} = \mathbf{P}_{j_2}.
\end{aligned} \tag{4.27}$$

Combining the above two relations and using the fact that $\mathbf{P}_{\mathrm{quad}}^{(j_1,j_2)} = \mathbf{I}_{\ell^3} - \mathbf{Q}_{j_1} \otimes \mathbf{P}_{j_2}$, we now have

$$\begin{aligned}
\mathbf{P}_{\mathrm{quad}}^{(j_1,j_2)}\left(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}^\mathsf{T}\mathbf{P}_{j_2})\right)\left(\mathbf{I}_{\ell^2} - (\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1})\right) &= \mathbf{P}_{\mathrm{quad}}^{(j_1,j_2)}\left(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}^\mathsf{T}\mathbf{P}_{j_2})\right)\mathbf{Q}_{j_1} \\
&= \mathbf{P}_{\mathrm{quad}}^{(j_1,j_2)}\left(\mathbf{Q}_{j_1} \otimes \mathrm{vec}(\mathbf{r}^\mathsf{T}\mathbf{P}_{j_2})\right) && \text{by Eq. (3.1)} \\
&= \left(\mathbf{I}_{\ell^3} - (\mathbf{Q}_{j_1} \otimes \mathbf{P}_{j_2})\right)\left(\mathbf{Q}_{j_1} \otimes \mathrm{vec}(\mathbf{r}^\mathsf{T}\mathbf{P}_{j_2})\right) && \text{by definition of } \mathbf{P}_{\mathrm{quad}}^{(j_1,j_2)} \\
&= \left(\mathbf{Q}_{j_1} \otimes \mathrm{vec}(\mathbf{r}^\mathsf{T}\mathbf{P}_{j_2})\right) - \left(\mathbf{Q}_{j_1} \otimes \mathrm{vec}(\mathbf{r}^\mathsf{T}\mathbf{P}_{j_2})\right) && \text{by Eq. (4.27)} \\
&= \mathbf{0}.
\end{aligned}$$

Next, the matrices $\mathbf{P}_{\mathrm{quad}}^{(j_1,j_2)}$ are diagonal for all $j_1, j_2 \in [\ell]$, so they commute. Thus,

$$\mathbf{P}_{\mathrm{quad}} = \prod_{(j_1,j_2)\in S} \mathbf{P}_{\mathrm{quad}}^{(j_1,j_2)} = \left(\prod_{(s,t)\in S\setminus\{(j_1,j_2)\}} \mathbf{P}_{\mathrm{quad}}^{(s,t)}\right) \cdot \mathbf{P}_{\mathrm{quad}}^{(j_1,j_2)}.$$

This means

$$\mathbf{P}_{\mathrm{quad}}\left(\mathbf{I}_{\ell^2}\otimes\mathrm{vec}(\mathbf{r}^\mathsf{T}\mathbf{P}_{j_2})\right)\left(\mathbf{I}_{\ell^2}-(\mathbf{P}_{j_1}\otimes\mathbf{P}_{j_1})\right) = \left(\prod_{(s,t)\in S\setminus\{(j_1,j_2)\}} \mathbf{P}_{\mathrm{quad}}^{(s,t)}\right)\cdot\mathbf{P}_{\mathrm{quad}}^{(j_1,j_2)}\left(\mathbf{I}_{\ell^2}\otimes\mathrm{vec}(\mathbf{r}^\mathsf{T}\mathbf{P}_{j_2})\right)\left(\mathbf{I}_{\ell^2}-(\mathbf{P}_{j_1}\otimes\mathbf{P}_{j_1})\right) = \mathbf{0}. \quad \square$$

Construction 4.38 (Projective Chainable Commitments for Local Quadratic Functions)**.** Let $\mathsf{FC}_{\mathsf{base}} = (\mathsf{SetupBase}, \mathsf{SetupSF}, \mathsf{Commit}^{(1)}, \mathsf{Commit}^{(2)}, \mathsf{Project}^{(1)}, \mathsf{Project}^{(2)})$ be the projective commitment scheme from Construction 4.8. We build a projective chainable commitment for local linear functions $\mathsf{FC}_{\mathsf{quad}} = (\mathsf{SetupQuad}, \mathsf{OpenQuad}, \mathsf{VerifyQuad})$ over $\mathsf{FC}_{\mathsf{base}}$ as follows:

- $\mathsf{SetupQuad}(\mathsf{crs}_{\mathsf{base}}, S)$: On input the common reference string $\mathsf{crs}_{\mathsf{base}} = (\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2)$ for the base projective commitment scheme (which defines the input space $\mathbb{Z}_p^\ell$) and a locality set $S \subseteq [\ell] \times [\ell]$, the setup algorithm samples $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k\times(k+1)}$, $\mathbf{R} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(k+1)\times 2k}$ and $\mathbf{W} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^3(k+1)\times 4k^2}$. It then computes

$$\begin{aligned}
[\mathbf{Z}]_2 &= \mathbf{W}[\mathbf{T}_*]_2 - (\mathbf{P}_{\mathrm{quad}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{R}[\hat{\mathbf{T}}]_2)) \\
&= [\mathbf{W}\mathbf{T}_* - (\mathbf{P}_{\mathrm{quad}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{R}\hat{\mathbf{T}}))]_2 \in \mathbb{G}_2^{\ell^3(k+1)\times\ell^2},
\end{aligned} \tag{4.28}$$

where $\mathbf{P}_{\mathrm{quad}} = \mathbf{P}^{(S)}_{\mathrm{quad}} \in \mathbb{Z}_p^{\ell^3 \times \ell^3}$ is the projection matrix from Eq. (4.25). Output the common reference string

$$\mathsf{crs} = (\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, [(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}]_1, [\mathbf{AR}]_1, [\mathbf{Z}]_2) \,. \tag{4.29}$$

- $\mathsf{OpenQuad}(\mathsf{crs}, \mathbf{x}, \mathbf{M})$: On input the common reference string (parsed as in Eq. (4.29)), the vector $\mathbf{x} \in \mathbb{Z}_p^\ell$, and a matrix $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell^2}$, the evaluation algorithm computes $[\mathbf{c}_*]_2 \leftarrow [\mathbf{T}_*]_2(\mathbf{x} \otimes \mathbf{x}) \in \mathbb{G}_2^{4k^2}$ and

$$[\mathbf{v}]_2 = (\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})[\mathbf{Z}]_2(\mathbf{x} \otimes \mathbf{x}) \in \mathbb{G}_2^{k+1}.$$

It outputs the opening $\pi = ([\mathbf{c}_*]_2, [\mathbf{v}]_2)$.

- $\mathsf{VerifyQuad}(\mathsf{crs}, \sigma_2, \mathbf{M}, \sigma_1, \pi)$: On input the common reference string crs (parsed as in Eq. (4.29)), a Type-II commitment $\sigma_2 = ([\mathbf{c}_1]_1, [\mathbf{c}_2]_2)$, a matrix $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell^2}$, a Type-I commitment $\sigma_1 = [\hat{\mathbf{c}}]_2$ and a proof $\pi = ([\mathbf{c}_*]_2, [\mathbf{v}]_2)$, the verification algorithm outputs 1 if

$$[\mathbf{c}_1]_1 \otimes [\mathbf{c}_2]_2 = [1]_1[\mathbf{c}_*]_2 \quad \text{and} \quad (\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_k)[(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}]_1[\mathbf{c}_*]_2 = [\mathbf{AR}]_1[\hat{\mathbf{c}}]_2 + [\mathbf{A}]_1[\mathbf{v}]_2.$$

Theorem 4.39 (Correctness). *Construction 4.38 is correct.*

Proof. Take any $\lambda, \ell \in \mathbb{N}$ and let $S \subseteq [\ell] \times [\ell]$ be an arbitrary locality set. Let $\mathsf{crs}_{\mathsf{base}} \leftarrow \mathsf{SetupBase}(1^\lambda, 1^\ell)$ and $\mathsf{crs} \leftarrow \mathsf{SetupQuad}(\mathsf{crs}_{\mathsf{base}}, S)$. Then $\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big)$ and

$$\mathsf{crs} = (\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, [(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}]_1, [\mathbf{AR}]_1, [\mathbf{Z}]_2).$$

Take any input $\mathbf{x} \in \mathbb{Z}_p^\ell$ and any S-local homogeneous quadratic function $f(\mathbf{x}) \coloneqq \mathbf{M}(\mathbf{x} \otimes \mathbf{x})$ where $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell^2}$. Let $\mathbf{y} = \mathbf{M}(\mathbf{x} \otimes \mathbf{x})$. Suppose $\sigma_2 \leftarrow \mathsf{Commit}^{(2)}(\mathsf{crs}_{\mathsf{base}}, \mathbf{x})$, $\sigma_1 \leftarrow \mathsf{Commit}^{(1)}(\mathsf{crs}_{\mathsf{base}}, \mathbf{y})$, and $\pi \leftarrow \mathsf{OpenQuad}(\mathsf{crs}, \mathbf{x}, \mathbf{M})$. We parse $\sigma_2 = ([\mathbf{c}_1]_1, [\mathbf{c}_2]_2)$, $\sigma_1 = [\hat{\mathbf{c}}_2]_2$, and $\pi = ([\mathbf{c}_*]_2, [\mathbf{v}]_2)$. Consider $\mathsf{VerifyQuad}(\mathsf{crs}, \sigma_2, \mathbf{M}, \sigma_1, \pi)$. By construction of the underlying algorithms, $\mathbf{c}_* = \mathbf{T}_*(\mathbf{x} \otimes \mathbf{x})$, $\mathbf{c}_1 = \mathbf{T}_1\mathbf{x}$, $\mathbf{c}_2 = \mathbf{T}_2\mathbf{x}$, $\hat{\mathbf{c}} = \hat{\mathbf{T}}\mathbf{y}$, and $\mathbf{v} = (\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})\mathbf{Z}(\mathbf{x} \otimes \mathbf{x})$. Consider now the verification relation $\mathsf{VerifyQuad}(\mathsf{crs}, \sigma_2, \mathbf{M}, \sigma_1, \pi)$:

- The first verification relation follows from Eq. (3.1):

$$\mathbf{c}_1 \otimes \mathbf{c}_2 = (\mathbf{T}_1\mathbf{x}) \otimes (\mathbf{T}_2\mathbf{x}) = (\mathbf{T}_1 \otimes \mathbf{T}_2)(\mathbf{x} \otimes \mathbf{x}) = \mathbf{T}_*(\mathbf{x} \otimes \mathbf{x}) = \mathbf{c}_*.$$

- For the second verification relation, we first compute

$$\begin{aligned} (\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}\mathbf{c}_* &= (\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}\mathbf{T}_*(\mathbf{x} \otimes \mathbf{x}) \\ &= (\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{A})\mathbf{W}\mathbf{T}_*(\mathbf{x} \otimes \mathbf{x}) && \text{by Eq. (3.1)} \\ &= \mathbf{A}(\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})\mathbf{W}\mathbf{T}_*(\mathbf{x} \otimes \mathbf{x}) && \text{by Eq. (3.3).} \end{aligned} \tag{4.30}$$

Next, since f is S-local, by Lemma 4.37, we have that $\mathrm{vec}(\mathbf{M})^\intercal\mathbf{P}_{\mathrm{quad}} = \mathrm{vec}(\mathbf{M})^\intercal$. This means

$$\begin{aligned} (\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})\mathbf{Z} &= (\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})\mathbf{W}\mathbf{T}_* - (\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})(\mathbf{P}_{\mathrm{quad}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{R}\hat{\mathbf{T}})) \\ &= (\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})\mathbf{W}\mathbf{T}_* - (\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{R}\hat{\mathbf{T}})). \end{aligned}$$

Thus, we have

$$(\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})\mathbf{W}\mathbf{T}_* = (\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})\mathbf{Z} + (\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{R}\hat{\mathbf{T}})).$$

Substituting into Eq. (4.30), and using the fact that $\mathbf{v} = (\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})\mathbf{Z}(\mathbf{x} \otimes \mathbf{x})$, we have

$$\begin{aligned} (\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}\mathbf{c}_* &= \mathbf{A}(\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})\mathbf{W}\mathbf{T}_*(\mathbf{x} \otimes \mathbf{x}) \\ &= \mathbf{A}(\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})\big(\mathbf{Z}(\mathbf{x} \otimes \mathbf{x}) + (\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{R}\hat{\mathbf{T}}))(\mathbf{x} \otimes \mathbf{x})\big) \\ &= \mathbf{A}\mathbf{v} + \mathbf{A}(\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{R}\hat{\mathbf{T}}))(\mathbf{x} \otimes \mathbf{x}) \\ &= \mathbf{A}\mathbf{v} + \mathbf{A}(\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})(\mathbf{x} \otimes \mathbf{x} \otimes \mathrm{vec}(\mathbf{R}\hat{\mathbf{T}})). \end{aligned} \tag{4.31}$$

To complete the proof, we now have

$$\begin{aligned}
(\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})(\mathbf{x} \otimes \mathbf{x} \otimes \mathrm{vec}(\mathbf{R}\hat{\mathbf{T}})) &= (\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})(\mathbf{x} \otimes \mathbf{x} \otimes \mathbf{I}_\ell \otimes \mathbf{I}_{k+1})\mathrm{vec}(\mathbf{R}\hat{\mathbf{T}}) && \text{by Eq. (3.2)} \\
&= \big((\mathrm{vec}(\mathbf{M})^\top(\mathbf{x} \otimes \mathbf{x} \otimes \mathbf{I}_\ell)) \otimes \mathbf{I}_{k+1}\big)\mathrm{vec}(\mathbf{R}\hat{\mathbf{T}}) && \text{by Eq. (3.1)} \\
&= \big((\mathbf{M}(\mathbf{x} \otimes \mathbf{x}))^\top \otimes \mathbf{I}_{k+1}\big)\mathrm{vec}(\mathbf{R}\hat{\mathbf{T}}) && \text{by Eq. (3.4)} \\
&= (\mathbf{y}^\top \otimes \mathbf{I}_{k+1})\mathrm{vec}(\mathbf{R}\hat{\mathbf{T}}) && \text{since } \mathbf{y} = \mathbf{M}(\mathbf{x} \otimes \mathbf{x}) \\
&= \mathbf{R}\hat{\mathbf{T}}\mathbf{y} = \mathbf{R}\hat{\mathbf{c}} && \text{by Eq. (3.4).}
\end{aligned}$$

Substituting back into Eq. (4.31), we have

$$\begin{aligned}
(\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}\mathbf{c}_* &= \mathbf{A}\mathbf{v} + \mathbf{A}(\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_{k+1})(\mathbf{x} \otimes \mathbf{x} \otimes \mathrm{vec}(\mathbf{R}\hat{\mathbf{T}})) \\
&= \mathbf{A}\mathbf{v} + \mathbf{A}\mathbf{R}\hat{\mathbf{c}}.
\end{aligned}$$

and the verification relation holds.

Since both verification relations pass, the output of Verify is 1 and the claim follows. ◻

Theorem 4.40 (Chain Binding for Quadratic Functions)**.** *Suppose the bilateral* k-Lin *assumption holds with respect to* GroupGen. *Then, Construction 4.38 satisfies chain binding for quadratic functions.*

Proof. Similar to the proof of Theorem 4.25, we start by defining a "homogeneous" version of the chain binding for quadratic functions security game for Construction 4.38. We define the game below:

1. On input the security parameter λ, the adversary outputs the dimension ℓ, a locality set $S \subseteq [\ell] \times [\ell]$, and a pair $(j_1, j_2) \in S$.
2. The challenger samples $(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupBase}(1^\lambda, 1^\ell, j_2, j_1)$ and $\mathsf{crs} \leftarrow \mathsf{Setup}(\mathsf{crs}_{\mathsf{base}})$. Then $\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big)$, $\mathsf{td}_1 = \hat{\mathbf{B}}_2$, $\mathsf{td}_2 = (\mathbf{B}_{1,2}, \mathbf{B}_{2,2})$, and
$$\mathsf{crs} = (\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, [(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}]_1, [\mathbf{A}\mathbf{R}]_1, [\mathbf{Z}]_2)\,.$$
The challenger gives crs to $\mathcal{A}$.
3. The adversary outputs an S-local homogeneous quadratic function $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell^2}$ and a triple $([\mathbf{c}_*]_2, [\hat{\mathbf{c}}]_2, [\mathbf{v}]_2)$.
4. The challenger outputs 1 if the following properties hold:
   - **Matching inputs:** $(\mathbf{B}_{1,2} \otimes \mathbf{B}_{2,2})\mathbf{c}_* = \mathbf{0}$.
   - **Mismatching outputs:** $\hat{\mathbf{B}}_2\hat{\mathbf{c}} \neq \mathbf{0}$.
   - **Validity of opening:** $(\mathrm{vec}(\mathbf{M})^\top \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}\mathbf{c}_* = \mathbf{A}\mathbf{R}\hat{\mathbf{c}} + \mathbf{A}\mathbf{v}$.

We now show that any adversary that can win the homogeneous chain binding security game (i.e., cause the above experiment to output 1) implies an adversary that can win the standard chain binding security game (Definition 4.35). Like the proof of Lemma 4.26, the claim essentially follows by linearity of the verification relation. We give the formal statement below:

Lemma 4.41. *Suppose for all efficient adversaries* $\mathcal{B}$*, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that* $\Pr[b = 1] = \mathsf{negl}(\lambda)$ *in the homogeneous chain binding experiment for quadratic functions. Then, Construction 4.38 satisfies chain binding security for quadratic functions.*

Proof. Suppose there exists an adversary $\mathcal{A}$ that breaks chain binding security for quadratic functions (Definition 4.35) with advantage ε. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ for the homogeneous chain binding game:

1. Algorithm $\mathcal{B}$ starts running algorithm $\mathcal{A}$ to obtain the input length 1^ℓ, the locality set $S \subseteq [\ell] \times [\ell]$, and a pair $(j_1, j_2) \in S$. It gives 1^ℓ, S, and (j_1, j_2) to the challenger to obtain the common reference string crs.

2. Algorithm $\mathcal{B}$ forwards crs to $\mathcal{A}$ and receives a matrix $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell^2}$, two Type-II commitments $\sigma_2 = ([\mathbf{c}_1]_1, [\mathbf{c}_2]_2)$, $\sigma_2' = ([\mathbf{c}_1']_1, [\mathbf{c}_2']_2)$, two Type-I commitments $\sigma_1 = [\hat{\mathbf{c}}]_2$, $\sigma_1' = [\hat{\mathbf{c}}']_2$, and two openings $\pi = ([\mathbf{c}_*]_2, [\mathbf{v}]_2)$, $\pi' = ([\mathbf{c}_*']_2, [\mathbf{v}']_2)$.

3. Algorithm $\mathcal{B}$ outputs the same function $\mathbf{M}$ together with the triple

$$\big([\mathbf{c}_*]_2 - [\mathbf{c}_*']_2, [\hat{\mathbf{c}}]_2 - [\hat{\mathbf{c}}']_2, [\mathbf{v}]_2 - [\mathbf{v}']_2\big).$$

In the homogeneous chain binding game, the challenger samples $(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^\lambda, 1^\ell, j_2, j_1)$ and $\mathsf{crs} \leftarrow \mathsf{SetupQuad}(\mathsf{crs}_{\mathsf{base}}, S)$. Thus algorithm $\mathcal{B}$ perfectly simulates an execution of the chain binding security game for $\mathcal{A}$. Thus, with probability ε, the outputs of algorithm $\mathcal{A}$ satisfies the following properties:

- **Matching inputs:** $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) = \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2')$.
- **Mismatching outputs:** $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) \neq \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$.
- **Validity of openings:** $\mathsf{VerifyQuad}(\mathsf{crs}, \sigma_2, \mathbf{M}, \sigma_1, \pi) = 1 = \mathsf{VerifyQuad}(\mathsf{crs}, \sigma_2', \mathbf{M}, \sigma_1', \pi')$.

We claim that in this case, the output in the homogeneous chain binding game is also 1:

- Parse $\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big)$ and $\mathsf{crs} = \big(\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, [(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}]_1, [\mathbf{AR}]_1, [\mathbf{Z}]_2\big)$. In addition, parse $\mathsf{td}_1 = \hat{\mathbf{B}}_2$, $\mathsf{td}_2 = (\mathbf{B}_{1,2}, \mathbf{B}_{2,2})$.
- Since $\mathsf{VerifyQuad}(\mathsf{crs}, \sigma_2, \mathbf{M}, \sigma_1, \pi) = 1 = \mathsf{VerifyQuad}(\mathsf{crs}, \sigma_2', \mathbf{M}, \sigma_1', \pi')$, the following two conditions hold:
  - $\mathbf{c}_1 \otimes \mathbf{c}_2 = \mathbf{c}_*$ and $\mathbf{c}_1' \otimes \mathbf{c}_2' = \mathbf{c}_*'$.
  - $(\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}\mathbf{c}_* = \mathbf{AR}\hat{\mathbf{c}} + \mathbf{Av}$ and $(\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}\mathbf{c}_*' = \mathbf{AR}\hat{\mathbf{c}}' + \mathbf{Av}'$.

  This means that
  $$(\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}(\mathbf{c}_* - \mathbf{c}_*') = \mathbf{AR}(\hat{\mathbf{c}} - \hat{\mathbf{c}}') + \mathbf{A}(\mathbf{v} - \mathbf{v}'),$$
  and the third requirement in the homogeneous chain binding experiment is satisfied.
- Since $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) = \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2')$, this means $\mathbf{B}_{1,2}\mathbf{c}_1 = \mathbf{B}_{1,2}\mathbf{c}_1'$ and $\mathbf{B}_{2,2}\mathbf{c}_2 = \mathbf{B}_{2,2}\mathbf{c}_2'$. This means that
  $$\begin{aligned}(\mathbf{B}_{1,2} \otimes \mathbf{B}_{2,2})\mathbf{c}_* = (\mathbf{B}_{1,2} \otimes \mathbf{B}_{2,2})(\mathbf{c}_1 \otimes \mathbf{c}_2) &= (\mathbf{B}_{1,2}\mathbf{c}_1) \otimes (\mathbf{B}_{2,2}\mathbf{c}_2) \\ &= (\mathbf{B}_{1,2}\mathbf{c}_1') \otimes (\mathbf{B}_{2,2}\mathbf{c}_2') = (\mathbf{B}_{1,2} \otimes \mathbf{B}_{2,2})(\mathbf{c}_1' \otimes \mathbf{c}_2') = (\mathbf{B}_{1,2} \otimes \mathbf{B}_{2,2})\mathbf{c}_*'.\end{aligned}$$
  Correspondingly, this means that $(\mathbf{B}_{1,2} \otimes \mathbf{B}_{2,2})(\mathbf{c}_* - \mathbf{c}_*') = \mathbf{0}$, and the first requirement of the homogeneous chain binding experiment is satisfied.
- Finally, if $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) \neq \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$, then $\hat{\mathbf{B}}_2\hat{\mathbf{c}} \neq \hat{\mathbf{B}}_2\hat{\mathbf{c}}'$. Thus, $\hat{\mathbf{B}}_2(\hat{\mathbf{c}} - \hat{\mathbf{c}}') \neq \mathbf{0}$, and the second requirement in the homogeneous game is satisfied.

Correspondingly, the output is 1 in the homogeneous evaluation binding game, and the claim follows. □

Proof of Theorem 4.40. We now return to the proof of Theorem 4.40. Let $\mathcal{A}$ be an efficient adversary for the homogeneous chain binding experiment for quadratic functions. Let $\ell \in \mathbb{N}$ be the vector dimension that $\mathcal{A}$ chooses at the beginning of the security experiment. This will determine the size of the tensor MDDH assumption in Lemma 4.46. We now define a sequence of hybrid experiments. The sequence of experiments closely parallels those in the proof of Theorem 4.25.

- Hyb_0: This is the homogeneous chain binding experiment for quadratic functions. We give the full specification here:
  - At the beginning of the game, the adversary $\mathcal{A}$ outputs the input dimension ℓ, a locality set $S \subseteq [\ell] \times [\ell]$, and a pair $(j_1, j_2) \in S$.

- The challenger samples $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e) \leftarrow \mathsf{GroupGen}(1^\lambda)$.
- The challenger samples full-rank matrices $\hat{\mathbf{B}}, \mathbf{B}_1, \mathbf{B}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times 2k}$ and defines $\hat{\mathbf{B}}^* = \hat{\mathbf{B}}^{-1}, \mathbf{B}_1^* = \mathbf{B}_1^{-1}, \mathbf{B}_2^* = \mathbf{B}_2^{-1}$. It parses $\hat{\mathbf{B}}, \mathbf{B}_1, \mathbf{B}_2$ as in Eq. (4.2) and $\hat{\mathbf{B}}^*, \mathbf{B}_1^*, \mathbf{B}_2^*$, as in Eq. (4.3).
- The challenger constructs the encoding matrices $\hat{\mathbf{T}}, \mathbf{T}_1, \mathbf{T}_2$ as follows:
  * **Type-I encodings:** Sample $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$ and let $\hat{\mathbf{T}} = \hat{\mathbf{B}}_1^* \hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^* \hat{\mathbf{S}}_2 \mathbf{P}_{j_2} \in \mathbb{Z}_p^{2k \times \ell}$.
  * **Type-II encodings:** For $\alpha \in \{1, 2\}$, sample $\mathbf{S}_{\alpha,1}, \mathbf{S}_{\alpha,2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$. Let $\mathbf{T}_\alpha = \mathbf{B}_{\alpha,1}^* \mathbf{S}_{\alpha,1} + \mathbf{B}_{\alpha,2}^* \mathbf{S}_{\alpha,2} \mathbf{P}_{j_1} \in \mathbb{Z}_p^{2k \times \ell}$.

  Let $\mathbf{T}_* = \mathbf{T}_1 \otimes \mathbf{T}_2$ and set $\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big)$.
- The challenger samples $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times (k+1)}$, $\mathbf{R} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(k+1) \times 2k}$ and $\mathbf{W} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^3 (k+1) \times 4k^2}$. Let

$$\mathbf{Z} = \mathbf{W}\mathbf{T}_* - (\mathbf{P}_{\mathsf{quad}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{R}\hat{\mathbf{T}})) \in \mathbb{Z}_p^{\ell^3(k+1) \times \ell^2}, \tag{4.32}$$

  where $\mathbf{P}_{\mathsf{quad}} = \mathbf{P}_{\mathsf{quad}}^{(S)}$ is the projection matrix from Eq. (4.25). The challenger gives the common reference string $\mathsf{crs} = (\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, [(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}]_1, [\mathbf{A}\mathbf{R}]_1, [\mathbf{Z}]_2)$ to $\mathcal{A}$.
- Algorithm $\mathcal{A}$ outputs an S-local function $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell^2}$, and a triple $([\mathbf{c}_*]_2, [\hat{\mathbf{c}}]_2, [\mathbf{v}]_2)$.

The output of the experiment is 1 if

$$(\mathbf{B}_{1,2} \otimes \mathbf{B}_{2,2})\mathbf{c}_* = \mathbf{0} \quad \text{and} \quad \hat{\mathbf{B}}_2 \hat{\mathbf{c}} \neq \mathbf{0} \quad \text{and} \quad (\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}\mathbf{c}_* = \mathbf{A}\mathbf{R}\hat{\mathbf{c}} + \mathbf{A}\mathbf{v}.$$

- Hyb_1: Same as Hyb_0, except the challenger samples $\mathbf{W}$ as follows:
  - Define matrices $\mathbf{D}_{\mathsf{norm}}$ and $\mathbf{D}_{\mathsf{sf}}$ as follows:

$$\mathbf{D}_{\mathsf{norm}} = \begin{bmatrix} \mathbf{B}_{1,1} \otimes \mathbf{B}_{2,1} \\ \mathbf{B}_{1,1} \otimes \mathbf{B}_{2,2} \\ \mathbf{B}_{1,2} \otimes \mathbf{B}_{2,1} \end{bmatrix} \in \mathbb{Z}_p^{3k^2 \times 4k^2} \quad \text{and} \quad \mathbf{D}_{\mathsf{sf}} = \mathbf{B}_{1,2} \otimes \mathbf{B}_{2,2} \in \mathbb{Z}_p^{k^2 \times 4k^2}. \tag{4.33}$$

  - Sample $\mathbf{W}_{\mathsf{norm}} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^3(k+1) \times 3k^2}$ and $\mathbf{W}_{\mathsf{sf}} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^3(k+1) \times k^2}$ and let $\mathbf{W} = \mathbf{W}_{\mathsf{norm}}\mathbf{D}_{\mathsf{norm}} + \mathbf{W}_{\mathsf{sf}}\mathbf{D}_{\mathsf{sf}}$.

  Then, after the adversary outputs $(\mathbf{M}, [\mathbf{c}_*]_2, [\hat{\mathbf{c}}]_2, [\mathbf{v}]_2)$, the challenger first computes

$$\mathbf{v}' = \mathbf{v} - (\mathrm{vec}(\mathbf{M})^\intercal \otimes \mathbf{I}_{k+1})\mathbf{W}_{\mathsf{norm}}\mathbf{D}_{\mathsf{norm}}\mathbf{c}_*. \tag{4.34}$$

  The output of the experiment is 1 if

$$\mathbf{D}_{\mathsf{sf}}\mathbf{c}_* = \mathbf{0} \quad \text{and} \quad \hat{\mathbf{B}}_2 \hat{\mathbf{c}} \neq \mathbf{0} \quad \text{and} \quad \mathbf{A}\mathbf{R}\hat{\mathbf{c}} + \mathbf{A}\mathbf{v}' = \mathbf{0}.$$

- Hyb_2: Same as Hyb_1 except the challenger outputs 1 if

$$\mathbf{D}_{\mathsf{sf}}\mathbf{c}_* = \mathbf{0} \quad \text{and} \quad \hat{\mathbf{B}}_2 \hat{\mathbf{c}} \neq \mathbf{0} \quad \text{and} \quad \mathbf{R}\hat{\mathbf{c}} + \mathbf{v}' = \mathbf{0}.$$

- Hyb_3: Same as Hyb_2 except when constructing the CRS, the challenger samples a random nonzero vector $\mathbf{a}^\perp \in \mathbb{Z}_p^{k+1}$ in the kernel of $\mathbf{A}$ (i.e., $\mathbf{A}\mathbf{a}^\perp = \mathbf{0}$). Then, it samples $\mathbf{W}_{\mathsf{sf},1} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^3(k+1) \times k^2}$, $\mathbf{W}_{\mathsf{sf},2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^3 \times k^2}$, $\mathbf{R}_1 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(k+1) \times 2k}$, and $\mathbf{r}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k}$. It sets

$$\mathbf{W}_{\mathsf{sf}} = \mathbf{W}_{\mathsf{sf},1} + (\mathbf{W}_{\mathsf{sf},2} \otimes \mathbf{a}^\perp) \quad \text{and} \quad \mathbf{R} = \mathbf{R}_1 + (\mathbf{r}_2^\intercal \otimes \mathbf{a}^\perp) = \mathbf{R}_1 + \mathbf{a}^\perp \mathbf{r}_2^\intercal.$$

  The challenger then computes

$$\begin{aligned} \mathbf{Z}_1 &= (\mathbf{W}_{\mathsf{norm}}\mathbf{D}_{\mathsf{norm}} + \mathbf{W}_{\mathsf{sf},1}\mathbf{D}_{\mathsf{sf}})\mathbf{T}_* - (\mathbf{P}_{\mathsf{quad}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{R}_1\hat{\mathbf{T}})) \\ \mathbf{Z}_2 &= \mathbf{W}_{\mathsf{sf},2}(\mathbf{S}_{1,2} \otimes \mathbf{S}_{2,2})(\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1}) - \mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_2^\intercal \hat{\mathbf{T}})\big) \end{aligned} \tag{4.35}$$

  and sets $\mathbf{Z} = \mathbf{Z}_1 + (\mathbf{I}_{\ell^3} \otimes \mathbf{a}^\perp)\mathbf{Z}_2$.

- Hyb_4: Same as Hyb_3 except when constructing the CRS, the challenger sets

$$\mathsf{crs} = (\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, [(\mathbf{I}_{\ell^3} \otimes \mathbf{A})(\mathbf{W}_{\mathsf{norm}}\mathbf{D}_{\mathsf{norm}} + \mathbf{W}_{\mathsf{sf},1}\mathbf{D}_{\mathsf{sf}})]_1, [\mathbf{A}\mathbf{R}_1]_1, [\mathbf{Z}]_2)$$

- Hyb_5: Same as Hyb_4, except the challenger samples $\mathbf{U} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^3 \times \ell^2}$ and sets

$$\mathbf{Z}_2 = \mathbf{U}(\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1}) - \mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_2^{\mathsf{T}}\hat{\mathbf{T}})\big).$$

- Hyb_6: Same as Hyb_5, except the challenger samples $\mathbf{r}_{2,\mathsf{norm}}, \mathbf{r}_{2,\mathsf{sf}} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$ and sets

$$\mathbf{r}_2^{\mathsf{T}} = \mathbf{r}_{2,\mathsf{norm}}^{\mathsf{T}}\hat{\mathbf{B}}_1 + \mathbf{r}_{2,\mathsf{sf}}^{\mathsf{T}}\hat{\mathbf{B}}_2.$$

  Then, it sets

$$\mathbf{Z}_2 = \mathbf{U}(\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1}) - \mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_{2,\mathsf{norm}}^{\mathsf{T}}\hat{\mathbf{S}}_1)\big) - \mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_{2,\mathsf{sf}}^{\mathsf{T}}\hat{\mathbf{S}}_2\mathbf{P}_{j_1})\big).$$

- Hyb_7: Same as Hyb_6, except the challenger sets

$$\mathbf{Z}_2 = \mathbf{U}(\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1}) - \mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_{2,\mathsf{norm}}^{\mathsf{T}}\hat{\mathbf{S}}_1)\big).$$

  Recall that in this experiment, the challenger still samples $\mathbf{U} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^3 \times \ell^2}$.

We write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output distribution of an execution of hybrid Hyb_i with adversary $\mathcal{A}$. We now show that the output distribution of each adjacent pair of hybrids is indistinguishable.

Lemma 4.42. $\Pr[\mathsf{Hyb}_0(\mathcal{A}) = 1] = \Pr[\mathsf{Hyb}_1(\mathcal{A}) = 1]$.

Proof. Since $\mathbf{B}_1$ and $\mathbf{B}_2$ are each a basis for $\mathbb{Z}_p^{2k}$, it follows that $\mathbf{B}_1 \otimes \mathbf{B}_2$ is a basis for $\mathbb{Z}_p^{4k^2}$. Moreover,

$$\mathbf{B}_1 \otimes \mathbf{B}_2 = \begin{bmatrix} \mathbf{B}_{1,1} \otimes \mathbf{B}_{2,1} \\ \mathbf{B}_{1,1} \otimes \mathbf{B}_{2,2} \\ \mathbf{B}_{1,2} \otimes \mathbf{B}_{2,1} \\ \mathbf{B}_{1,2} \otimes \mathbf{B}_{2,2} \end{bmatrix} = \begin{bmatrix} \mathbf{D}_{\mathsf{norm}} \\ \mathbf{D}_{\mathsf{sf}} \end{bmatrix}.$$

This means that the distribution of $\mathbf{W}$ is identically distributed in Hyb_0 and Hyb_1. It suffices to consider the outputs of the two experiments. Suppose $\mathcal{A}$ outputs $(\mathbf{M}, [\mathbf{c}_*]_2, [\hat{\mathbf{c}}]_2, [\mathbf{v}]_2)$. Suppose $\mathbf{D}_{\mathsf{sf}}\mathbf{c}^* \neq \mathbf{0}$. Then, the output in both experiments is 0. Consider the case where $\mathbf{D}_{\mathsf{sf}}\mathbf{c}^* = \mathbf{0}$. In this case,

$$\mathbf{W}\mathbf{c}_* = \mathbf{W}_{\mathsf{norm}}\mathbf{D}_{\mathsf{norm}}\mathbf{c}_* + \mathbf{W}_{\mathsf{sf}}\mathbf{D}_{\mathsf{sf}}\mathbf{c}_* = \mathbf{W}_{\mathsf{norm}}\mathbf{D}_{\mathsf{norm}}\mathbf{c}_*. \tag{4.36}$$

Now, in Hyb_1, we have

$$\begin{aligned} \mathbf{A}\mathbf{R}\hat{\mathbf{c}} + \mathbf{A}\mathbf{v}' &= \mathbf{A}\mathbf{R}\hat{\mathbf{c}} + \mathbf{A}\mathbf{v} - \mathbf{A}(\mathrm{vec}(\mathbf{M})^{\mathsf{T}} \otimes \mathbf{I}_{k+1})\mathbf{W}_{\mathsf{norm}}\mathbf{D}_{\mathsf{norm}}\mathbf{c}_* && \text{by Eq. (4.34)} \\ &= \mathbf{A}\mathbf{R}\hat{\mathbf{c}} + \mathbf{A}\mathbf{v} - (\mathrm{vec}(\mathbf{M})^{\mathsf{T}} \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}_{\mathsf{norm}}\mathbf{D}_{\mathsf{norm}}\mathbf{c}_* && \text{by Eq. (3.3)} \\ &= \mathbf{A}\mathbf{R}\hat{\mathbf{c}} + \mathbf{A}\mathbf{v} - (\mathrm{vec}(\mathbf{M})^{\mathsf{T}} \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}\mathbf{c}_* && \text{by Eq. (4.36)}. \end{aligned}$$

Thus, in Hyb_1, if $\mathbf{D}_{\mathsf{sf}}\mathbf{c}_* = \mathbf{0}$, then $\mathbf{A}\mathbf{R}\hat{\mathbf{c}}+\mathbf{A}\mathbf{v}' = \mathbf{0}$ if and only if $\mathbf{A}\mathbf{R}\hat{\mathbf{c}}+\mathbf{A}\mathbf{v} = (\mathrm{vec}(\mathbf{M})^{\mathsf{T}} \otimes \mathbf{I}_k)(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}\mathbf{c}_*$. Correspondingly, the output distribution of $\mathsf{Hyb}_1(\mathcal{A})$ is identical to the output distribution of $\mathsf{Hyb}_0(\mathcal{A})$. ☐

Lemma 4.43. *Suppose the* $\mathsf{KerDH}_{k,k+1}$ *assumption holds in* $\mathbb{G}_1$ *with respect to* $\mathsf{GroupGen}$. *Then, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that* $|\Pr[\mathsf{Hyb}_1(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_2(\mathcal{A}) = 1]| \leq \mathsf{negl}(\lambda)$.

Proof. Suppose $|\Pr[\mathsf{Hyb}_1(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_2(\mathcal{A}) = 1]| \geq \varepsilon$ for some non-negligible ε. The only difference between Hyb_1 and Hyb_2 is the verification relation. Let $(\mathbf{M}, [\mathbf{c}_*]_2, [\hat{\mathbf{c}}]_2, [\mathbf{v}]_2)$ be the output of $\mathcal{A}$ in an execution of Hyb_1 or Hyb_2. If the outputs of Hyb_1 and Hyb_2 differ, then it must be the case that

$$\mathbf{A}(\mathbf{R}\hat{\mathbf{c}} + \mathbf{v}') = \mathbf{0} \quad \text{and} \quad \mathbf{R}\hat{\mathbf{c}} + \mathbf{v}' \neq \mathbf{0}. \tag{4.37}$$

In all other cases, the output in Hyb_1 and Hyb_2 is identical. We use $\mathcal{A}$ to construct an efficient adversary $\mathcal{B}$ for $\mathsf{KerDH}_{k,k+1}$:

1. On input the KerDH challenge $(\mathcal{G}, [\mathbf{A}]_1)$, algorithm $\mathcal{B}$ starts by running algorithm $\mathcal{A}$. Algorithm $\mathcal{A}$ outputs the input dimension ℓ, the locality set $S \subseteq [\ell] \times [\ell]$, and a pair $(j_1, j_2) \in S$.
2. Next, algorithm $\mathcal{B}$ samples full-rank matrices $\hat{\mathbf{B}}, \mathbf{B}_1, \mathbf{B}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times 2k}$ and defines $\hat{\mathbf{B}}^* = \hat{\mathbf{B}}^{-1}$, $\mathbf{B}_1^* = \mathbf{B}_1^{-1}$, and $\mathbf{B}_2^* = \mathbf{B}_2^{-1}$. It parses the components of $\hat{\mathbf{B}}, \mathbf{B}_1, \mathbf{B}_2$ as in Eq. (4.2) and $\hat{\mathbf{B}}^*, \mathbf{B}_1^*, \mathbf{B}_2^*$ as in Eq. (4.3).
3. Algorithm $\mathcal{B}$ then constructs the encoding matrices $\hat{\mathbf{T}}, \mathbf{T}_1, \mathbf{T}_2$ as in Hyb_1 and Hyb_2:
   - **Type-I encodings:** Sample $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$ and let $\hat{\mathbf{T}} = \hat{\mathbf{B}}_1^* \hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^* \hat{\mathbf{S}}_2 \mathbf{P}_{j_2} \in \mathbb{Z}_p^{2k \times \ell}$.
   - **Type-II encodings:** For $\alpha \in \{1, 2\}$, sample $\mathbf{S}_{\alpha,1}, \mathbf{S}_{\alpha,2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$ and let $\mathbf{T}_\alpha = \mathbf{B}_{\alpha,1}^* \mathbf{S}_{\alpha,1} + \mathbf{B}_{\alpha,2}^* \mathbf{S}_{\alpha,2} \mathbf{P}_{j_1}$.

   Let $\mathbf{T}_* = \mathbf{T}_1 \otimes \mathbf{T}_2$ and set $\mathsf{crs}_{\mathsf{base}} = \big(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\big)$.
4. Algorithm $\mathcal{B}$ defines $\mathbf{D}_{\mathsf{norm}}$ and $\mathbf{D}_{\mathsf{sf}}$ according to Eq. (4.33). It samples $\mathbf{W}_{\mathsf{norm}} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^3(k+1) \times 3k^2}$ and $\mathbf{W}_{\mathsf{sf}} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^3(k+1) \times k^2}$ and sets $\mathbf{W} = \mathbf{W}_{\mathsf{norm}} \mathbf{D}_{\mathsf{norm}} + \mathbf{W}_{\mathsf{sf}} \mathbf{D}_{\mathsf{sf}}$. It also samples $\mathbf{R} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(k+1) \times 2k}$ and constructs

   $$\mathbf{Z} = \mathbf{W}\mathbf{T}_* - (\mathbf{P}_{\mathsf{quad}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{R}\hat{\mathbf{T}})) \in \mathbb{Z}_p^{\ell^3(k+1) \times \ell^2},$$

   where $\mathbf{P}_{\mathsf{quad}} = \mathbf{P}_{\mathsf{quad}}^{(S)}$. The challenger gives the common reference string crs to $\mathcal{A}$ where

   $$\mathsf{crs} = (\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, (\mathbf{I}_{\ell^3} \otimes [\mathbf{A}]_1)\mathbf{W}, [\mathbf{A}]_1 \mathbf{R}, [\mathbf{Z}]_2) = (\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, [(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}]_1, [\mathbf{A}\mathbf{R}]_1, [\mathbf{Z}]_2)$$
5. After algorithm $\mathcal{A}$ outputs $(\mathbf{M}, [\mathbf{c}_*]_2, [\hat{\mathbf{c}}]_2, [\mathbf{v}]_2)$ algorithm $\mathcal{B}$ computes

   $$[\mathbf{v}']_2 = [\mathbf{v}]_2 - (\mathrm{vec}(\mathbf{M})^{\mathsf{T}} \otimes \mathbf{I}_{k+1}) \mathbf{W}_{\mathsf{norm}} \mathbf{D}_{\mathsf{norm}} [\mathbf{c}_*]_2$$

   and outputs $\mathbf{R}[\hat{\mathbf{c}}]_2 + [\mathbf{v}']_2 = [\mathbf{R}\mathbf{c} + \mathbf{v}']_2$.

Since the KerDH challenger samples $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(k+1) \times k}$, the common reference string crs constructed by $\mathcal{B}$ is distributed exactly as required in Hyb_1 and Hyb_2. By the above analysis, this means that with probability ε, algorithm $\mathcal{A}$ outputs $(\mathbf{M}, [\mathbf{c}_*]_2, [\hat{\mathbf{c}}]_2, [\mathbf{v}]_2)$ that satisfies Eq. (4.37). This means $\mathbf{R}\hat{\mathbf{c}} + \mathbf{v}' \neq \mathbf{0}$ but $\mathbf{A}(\mathbf{R}\hat{\mathbf{c}} + \mathbf{v}') = \mathbf{0}$, where $\mathbf{v}' = \mathbf{v} - (\mathrm{vec}(\mathbf{M})^{\mathsf{T}} \otimes \mathbf{I}_{k+1}) \mathbf{W}_{\mathsf{norm}} \mathbf{D}_{\mathsf{norm}} \mathbf{c}_*$. Correspondingly, algorithm $\mathcal{B}$ breaks KerDH with the same advantage ε. ☐

Lemma 4.44. $\Pr[\mathsf{Hyb}_2(\mathcal{A}) = 1] = \Pr[\mathsf{Hyb}_3(\mathcal{A}) = 1]$.

Proof. We argue that Hyb_2 and Hyb_3 are identically distributed. Since $\mathbf{W}_{\mathsf{sf},1}$ and $\mathbf{R}_1$ are uniform over their respective domains, it follows that $\mathbf{W}_{\mathsf{sf}}$ and $\mathbf{R}$ are identically distributed as in Hyb_2 and Hyb_3. To complete the proof, we show that the distribution of $\mathbf{Z}$ in Hyb_3 is identical to that in Hyb_2. Suppose we construct $\mathbf{Z}$ according to Eq. (4.32). Then, we have

$$\begin{aligned}
\mathbf{Z} &= \mathbf{W}\mathbf{T}_* - (\mathbf{P}_{\mathsf{quad}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{R}\hat{\mathbf{T}})) \\
&= (\mathbf{W}_{\mathsf{norm}} \mathbf{D}_{\mathsf{norm}} + \mathbf{W}_{\mathsf{sf},1} \mathbf{D}_{\mathsf{sf}} + (\mathbf{W}_{\mathsf{sf},2} \otimes \mathbf{a}^{\perp}) \mathbf{D}_{\mathsf{sf}}) \mathbf{T}_* - (\mathbf{P}_{\mathsf{quad}} \otimes \mathbf{I}_{k+1}) \big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}\big((\mathbf{R}_1 + \mathbf{a}^{\perp} \mathbf{r}_2^{\mathsf{T}}) \hat{\mathbf{T}}\big)\big) \\
&= \mathbf{Z}_1 + (\mathbf{W}_{\mathsf{sf},2} \otimes \mathbf{a}^{\perp}) \mathbf{D}_{\mathsf{sf}} \mathbf{T}_* - (\mathbf{P}_{\mathsf{quad}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{a}^{\perp} \mathbf{r}_2^{\mathsf{T}} \hat{\mathbf{T}}))
\end{aligned} \tag{4.38}$$

We analyze the components of $\mathbf{Z}$ in the subspace spanned by $\mathbf{a}^\perp$. First, using Eq. (3.3), we can write

$$(\mathbf{W}_{\mathsf{sf},2} \otimes \mathbf{a}^\perp)\mathbf{D}_{\mathsf{sf}}\mathbf{T}_* = (\mathbf{I}_{\ell^3} \otimes \mathbf{a}^\perp)\mathbf{W}_{\mathsf{sf},2}\mathbf{D}_{\mathsf{sf}}\mathbf{T}_*. \tag{4.39}$$

By definition, $\mathbf{D}_{\mathsf{sf}} = \mathbf{B}_{1,2} \otimes \mathbf{B}_{2,2}$ and $\mathbf{T}_* = \mathbf{T}_1 \otimes \mathbf{T}_2$. By orthogonality, we can write

$$\begin{aligned}
\mathbf{D}_{\mathsf{sf}}\mathbf{T}_* &= (\mathbf{B}_{1,2} \otimes \mathbf{B}_{2,2})(\mathbf{T}_1 \otimes \mathbf{T}_2) \\
&= \mathbf{B}_{1,2}\big(\mathbf{B}_{1,1}^*\mathbf{S}_{1,1} + \mathbf{B}_{1,2}^*\mathbf{S}_{1,2}\mathbf{P}_{j_1}\big) \otimes \mathbf{B}_{2,2}\big(\mathbf{B}_{2,1}^*\mathbf{S}_{2,1} + \mathbf{B}_{2,2}^*\mathbf{S}_{2,2}\mathbf{P}_{j_1}\big) \\
&= (\mathbf{S}_{1,2} \otimes \mathbf{S}_{2,2})(\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1}).
\end{aligned}$$

Substituting back into Eq. (4.39), we have

$$(\mathbf{W}_{\mathsf{sf},2} \otimes \mathbf{a}^\perp)\mathbf{D}_{\mathsf{sf}}\mathbf{T}_* = (\mathbf{I}_{\ell^3} \otimes \mathbf{a}^\perp)\mathbf{W}_{\mathsf{sf},2}\mathbf{D}_{\mathsf{sf}}\mathbf{T}_* = (\mathbf{I}_{\ell^3} \otimes \mathbf{a}^\perp)\mathbf{W}_{\mathsf{sf},2}(\mathbf{S}_{1,2} \otimes \mathbf{S}_{2,2})(\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1}). \tag{4.40}$$

For the remaining component in Eq. (4.38),

$$\begin{aligned}
\mathbf{I}_{\ell^2} \otimes \mathrm{vec}\big(\mathbf{a}^\perp\mathbf{r}_2^\intercal\hat{\mathbf{T}}\big) &= \mathbf{I}_{\ell^2} \otimes \big[(\mathbf{I}_\ell \otimes \mathbf{a}^\perp\mathbf{r}_2^\intercal)\mathrm{vec}(\hat{\mathbf{T}})\big] && \text{by Eq. (3.4)} \\
&= \mathbf{I}_{\ell^2} \otimes \big[(\mathbf{I}_\ell \otimes \mathbf{a}^\perp)(\mathbf{I}_\ell \otimes \mathbf{r}_2^\intercal)\mathrm{vec}(\hat{\mathbf{T}})\big] && \text{by Eq. (3.1)} \\
&= \mathbf{I}_{\ell^2} \otimes \big[(\mathbf{I}_\ell \otimes \mathbf{a}^\perp)\mathrm{vec}(\mathbf{r}_2^\intercal\hat{\mathbf{T}})\big] && \text{by Eq. (3.4)} \\
&= \big(\mathbf{I}_{\ell^2} \otimes (\mathbf{I}_\ell \otimes \mathbf{a}^\perp)\big)\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_2^\intercal\hat{\mathbf{T}})\big) && \text{by Eq. (3.1)} \\
&= \big(\mathbf{I}_{\ell^3} \otimes \mathbf{a}^\perp\big)\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_2^\intercal\hat{\mathbf{T}})\big).
\end{aligned}$$

Combined with Eq. (3.3),

$$\begin{aligned}
(\mathbf{P}_{\mathsf{quad}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{a}^\perp\mathbf{r}_2^\intercal\hat{\mathbf{T}})) &= (\mathbf{P}_{\mathsf{quad}} \otimes \mathbf{I}_{k+1})\big(\mathbf{I}_{\ell^3} \otimes \mathbf{a}^\perp\big)\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_2^\intercal\hat{\mathbf{T}})\big) \\
&= (\mathbf{I}_{\ell^3} \otimes \mathbf{a}^\perp)\mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_2^\intercal\hat{\mathbf{T}})\big).
\end{aligned} \tag{4.41}$$

Combining Eqs. (4.38), (4.40), and (4.41), we have the desired result:

$$\begin{aligned}
\mathbf{Z} &= \mathbf{Z}_1 + (\mathbf{W}_{\mathsf{sf},2} \otimes \mathbf{a}^\perp)\mathbf{D}_{\mathsf{sf}}\mathbf{T}_* - (\mathbf{P}_{\mathsf{quad}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{a}^\perp\mathbf{r}_2^\intercal\hat{\mathbf{T}})) && \text{by Eq. (4.38)} \\
&= \mathbf{Z}_1 + \big(\mathbf{I}_{\ell^3} \otimes \mathbf{a}^\perp\big)\big(\mathbf{W}_{\mathsf{sf},2}(\mathbf{S}_{1,2} \otimes \mathbf{S}_{2,2})(\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1}) - \mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_2^\intercal\hat{\mathbf{T}})\big)\big) && \text{by Eqs. (4.40) and (4.41)} \\
&= \mathbf{Z}_1 + (\mathbf{I}_{\ell^3} \otimes \mathbf{a}^\perp)\mathbf{Z}_2 && \text{by definition of } \mathbf{Z}_2 \text{ from Eq. (4.35)},
\end{aligned}$$

which is precisely how the challenger constructs $\mathbf{Z}$ in Hyb_3. We conclude that the common reference string in Hyb_2 and Hyb_3 are identically distributed. ◻

Lemma 4.45. $\Pr[\mathsf{Hyb}_3(\mathcal{A}) = 1] = \Pr[\mathsf{Hyb}_4(\mathcal{A}) = 1]$.

Proof. The distribution of crs in the two experiments are identical. In particular, in Hyb_3,

$$\begin{aligned}
(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W} &= (\mathbf{I}_{\ell^3} \otimes \mathbf{A})(\mathbf{W}_{\mathsf{norm}}\mathbf{D}_{\mathsf{norm}} + \mathbf{W}_{\mathsf{sf}}\mathbf{D}_{\mathsf{sf}}) \\
&= (\mathbf{I}_{\ell^3} \otimes \mathbf{A})(\mathbf{W}_{\mathsf{norm}}\mathbf{D}_{\mathsf{norm}} + \mathbf{W}_{\mathsf{sf},1}\mathbf{D}_{\mathsf{sf}} + (\mathbf{W}_{\mathsf{sf},2} \otimes \mathbf{a}^\perp)\mathbf{D}_{\mathsf{sf}}) \\
&= (\mathbf{I}_{\ell^3} \otimes \mathbf{A})(\mathbf{W}_{\mathsf{norm}}\mathbf{D}_{\mathsf{norm}} + \mathbf{W}_{\mathsf{sf},1}\mathbf{D}_{\mathsf{sf}})
\end{aligned}$$

since $(\mathbf{I}_{\ell^3} \otimes \mathbf{A})(\mathbf{W}_{\mathsf{sf},2} \otimes \mathbf{a}^\perp) = \mathbf{W}_{\mathsf{sf},2} \otimes \mathbf{A}\mathbf{a}^\perp = \mathbf{0}$. Similarly,

$$\mathbf{A}\mathbf{R} = \mathbf{A}(\mathbf{R}_1 + \mathbf{a}^\perp\mathbf{r}_2^\intercal) = \mathbf{A}\mathbf{R}_1 + \mathbf{A}\mathbf{a}^\perp\mathbf{r}_2^\intercal = \mathbf{A}\mathbf{R}_1.$$

This coincides with the distribution of crs in Hyb_4. ◻

Lemma 4.46. *Suppose the tensor* $\mathsf{MDDH}_{k,\ell,\ell,\ell^3}$ *assumption holds with respect to* $\mathsf{GroupGen}$. *Then, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that* $|\Pr[\mathsf{Hyb}_4(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_5(\mathcal{A}) = 1]| = \mathsf{negl}(\lambda)$.

Proof. Suppose $|\Pr[\mathsf{Hyb}_4(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_5(\mathcal{A}) = 1]| \geq \varepsilon$ for some non-negligible ε. We use $\mathcal{A}$ to construct an efficient adversary $\mathcal{B}$ for $\mathsf{MDDH}_{k,\ell,\ell,\ell^3}$:

1. On input the tensor MDDH challenge $(\mathcal{G}, [\mathbf{S}_{1,2}]_1, [\mathbf{S}_{1,2}]_2, [\mathbf{S}_{2,2}]_1, [\mathbf{S}_{2,2}]_2, [\mathbf{S}_{1,2} \otimes \mathbf{S}_{2,2}]_2, [\mathbf{V}]_2)$, algorithm $\mathcal{A}$ samples full-rank matrices $\hat{\mathbf{B}}, \mathbf{B}_1, \mathbf{B}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k \times 2k}$ and defines $\hat{\mathbf{B}}^* = \hat{\mathbf{B}}^{-1}$, $\mathbf{B}_1^* = \mathbf{B}_1^{-1}$, and $\mathbf{B}_2^* = \mathbf{B}_2^{-1}$. It parses $\hat{\mathbf{B}}, \mathbf{B}_1, \mathbf{B}_2$ as in Eq. (4.2) and $\hat{\mathbf{B}}^*, \mathbf{B}_1^*, \mathbf{B}_2^*$ as in Eq. (4.3). Define the matrices $\mathbf{D}_{\mathsf{norm}}$ and $\mathbf{D}_{\mathsf{sf}}$ as in Eq. (4.33).

2. Algorithm $\mathcal{A}$ constructs the encoding matrices $\hat{\mathbf{T}}, \mathbf{T}_1, \mathbf{T}_2$ as follows:

   - **Type-I encodings:** Sample $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$ and let $\hat{\mathbf{T}} = \hat{\mathbf{B}}_1^* \hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^* \hat{\mathbf{S}}_2 \mathbf{P}_{j_2} \in \mathbb{Z}_p^{2k \times \ell}$.
   - **Type-II encodings:** Sample $\mathbf{S}_{1,1}, \mathbf{S}_{2,1} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$. It constructs the encodings

$$\begin{aligned}
[\mathbf{T}_1]_1 &= \mathbf{B}_{1,1}^* \mathbf{S}_{1,1} + \mathbf{B}_{1,2}^* [\mathbf{S}_{1,2}]_1 \mathbf{P}_{j_1} = \left[\mathbf{B}_{1,1}^* \mathbf{S}_{1,1} + \mathbf{B}_{1,2}^* \mathbf{S}_{1,2} \mathbf{P}_{j_1}\right]_1 \\
[\mathbf{T}_1]_2 &= \mathbf{B}_{1,1}^* \mathbf{S}_{1,1} + \mathbf{B}_{1,2}^* [\mathbf{S}_{1,2}]_2 \mathbf{P}_{j_1} = \left[\mathbf{B}_{1,1}^* \mathbf{S}_{1,1} + \mathbf{B}_{1,2}^* \mathbf{S}_{1,2} \mathbf{P}_{j_1}\right]_2 \\
[\mathbf{T}_2]_2 &= \mathbf{B}_{2,1}^* \mathbf{S}_{2,1} + \mathbf{B}_{2,2}^* [\mathbf{S}_{2,2}]_2 \mathbf{P}_{j_1} = \left[\mathbf{B}_{2,1}^* \mathbf{S}_{2,1} + \mathbf{B}_{2,2}^* \mathbf{S}_{2,2} \mathbf{P}_{j_1}\right]_2.
\end{aligned}$$

   - **Tensor encoding:** Compute

$$\begin{aligned}
[\mathbf{T}_*]_2 = &(\mathbf{B}_{1,1}^* \otimes \mathbf{B}_{2,1}^*)(\mathbf{S}_{1,1} \otimes \mathbf{S}_{2,1}) + (\mathbf{B}_{1,1}^* \otimes \mathbf{B}_{2,2}^*)(\mathbf{S}_{1,1} \otimes [\mathbf{S}_{2,2}]_2)(\mathbf{I}_\ell \otimes \mathbf{P}_{j_1}) \\
&+ (\mathbf{B}_{1,2}^* \otimes \mathbf{B}_{2,1}^*)([\mathbf{S}_{1,2}]_2 \otimes \mathbf{S}_{2,1})(\mathbf{P}_{j_1} \otimes \mathbf{I}_\ell) + (\mathbf{B}_{1,2}^* \otimes \mathbf{B}_{2,2}^*)[\mathbf{S}_{1,2} \otimes \mathbf{S}_{2,2}]_2 (\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1}).
\end{aligned}$$

   Let $\mathsf{crs}_{\mathsf{base}} = \left(\mathcal{G}, [\hat{\mathbf{T}}]_2, [\mathbf{T}_1]_1, [\mathbf{T}_1]_2, [\mathbf{T}_2]_2, [\mathbf{T}_*]_2\right)$.

3. Sample $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times (k+1)}$ and a random nonzero vector $\mathbf{a}^\perp \in \mathbb{Z}_p^{k+1}$ in the kernel of $\mathbf{A}$.

4. Sample $\mathbf{W}_{\mathsf{norm}} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^3(k+1) \times 3k^2}$, $\mathbf{W}_{\mathsf{sf},1} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^3(k+1) \times k^2}$, $\mathbf{R}_1 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{(k+1) \times 2k}$, and $\mathbf{r}_2 \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{2k}$. It sets $\mathbf{R} = \mathbf{R}_1 + \mathbf{a}^\perp \mathbf{r}_2^{\mathsf{T}}$. It then computes

$$\begin{aligned}
[\mathbf{Z}_1]_2 &= (\mathbf{W}_{\mathsf{norm}} \mathbf{D}_{\mathsf{norm}} + \mathbf{W}_{\mathsf{sf},1} \mathbf{D}_{\mathsf{sf}})[\mathbf{T}_*]_2 - (\mathbf{P}_{\mathsf{quad}} \otimes \mathbf{I}_{k+1})(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{R}_1 \hat{\mathbf{T}})) \\
[\mathbf{Z}_2]_2 &= [\mathbf{V}]_2 (\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1}) - \mathbf{P}_{\mathsf{quad}} \left(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_2^{\mathsf{T}} \hat{\mathbf{T}})\right),
\end{aligned}$$

   and $[\mathbf{Z}]_2 = [\mathbf{Z}_1]_2 + (\mathbf{I}_{\ell^3} \otimes \mathbf{a}^\perp)[\mathbf{Z}_2]_2$.

5. Finally, algorithm $\mathcal{B}$ gives $\mathsf{crs} = (\mathsf{crs}_{\mathsf{base}}, [\mathbf{A}]_1, [(\mathbf{I}_{\ell^3} \otimes \mathbf{A})(\mathbf{W}_{\mathsf{norm}} \mathbf{D}_{\mathsf{norm}} + \mathbf{W}_{\mathsf{sf},1} \mathbf{D}_{\mathsf{sf}})]_1, [\mathbf{A}\mathbf{R}_1]_1, [\mathbf{Z}]_2)$ to $\mathcal{A}$.

6. After algorithm $\mathcal{A}$ outputs $(\mathbf{M}, [\mathbf{c}_*]_2, [\hat{\mathbf{c}}]_2, [\mathbf{v}]_2)$, algorithm $\mathcal{B}$ outputs 1 if the following hold

$$\mathbf{D}_{\mathsf{sf}}[\mathbf{c}_*]_2 = [\mathbf{0}]_2 \quad \text{and} \quad \hat{\mathbf{B}}_2 [\hat{\mathbf{c}}]_2 \neq \mathbf{0} \quad \text{and} \quad \mathbf{R}[\hat{\mathbf{c}}]_2 + [\mathbf{v}]_2 - (\mathrm{vec}(\mathbf{M})^{\mathsf{T}} \otimes \mathbf{I}_{k+1}) \mathbf{W}_{\mathsf{norm}} \mathbf{D}_{\mathsf{norm}} [\mathbf{c}_*]_2 = [\mathbf{0}]_2.$$

By definition, the tensor MDDH challenger samples $\mathbf{S}_{1,2}, \mathbf{S}_{2,2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{k \times \ell}$. Thus, algorithm $\mathcal{B}$ perfectly simulates the distribution of every component other than $[\mathbf{Z}]_2$ in the common reference string according to the specification of Hyb_4 and Hyb_5. Thus it suffices to consider the distribution of $\mathbf{Z}$ in the two cases:

- Suppose $\mathbf{V} = \mathbf{W}_{\mathsf{sf},2}(\mathbf{S}_{1,2} \otimes \mathbf{S}_{2,2})$ where the challenger samples $\mathbf{W}_{\mathsf{sf},2} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^3 \times k^2}$. Then algorithm $\mathcal{B}$ perfectly simulates the distribution of crs in Hyb_4. In this case, algorithm $\mathcal{B}$ outputs 1 with probability $\Pr[\mathsf{Hyb}_4(\mathcal{A}) = 1]$.

- Suppose $\mathbf{V} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^3 \times \ell^2}$. This corresponds to the distribution of $\mathbf{Z}$ in Hyb_5, so in this case, algorithm $\mathcal{B}$ outputs 1 with probability $\Pr[\mathsf{Hyb}_5(\mathcal{A}) = 1]$.

We conclude that the distinguishing advantage of $\mathcal{B}$ is exactly ε and the claim follows. □

Lemma 4.47. $\Pr[\mathsf{Hyb}_5(\mathcal{A}) = 1] = \Pr[\mathsf{Hyb}_6(\mathcal{A}) = 1]$.

Proof. Since $\hat{\mathbf{B}}$ is a basis for $\mathbb{Z}_p^{2k}$, the distribution of $\mathbf{r}_2$ in Hyb_6 is uniform over $\mathbb{Z}_p^{2k}$, which is identical to the distribution of $\mathbf{r}_2$ in Hyb_5. It suffices to argue that $\mathbf{Z}_2$ is computed identically. This follows by the fact that $\hat{\mathbf{B}}\hat{\mathbf{B}}^* = \mathbf{I}_{2k}$ and the fact that $\hat{\mathbf{T}} = \hat{\mathbf{B}}_1^*\hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^*\hat{\mathbf{S}}_2\mathbf{P}_{j_2}$. In particular, we can write

$$\begin{aligned}\mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_2^{\mathsf{T}}\hat{\mathbf{T}})\big) &= \mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}\big((\mathbf{r}_{2,\mathsf{norm}}^{\mathsf{T}}\hat{\mathbf{B}}_1 + \mathbf{r}_{2,\mathsf{sf}}^{\mathsf{T}}\hat{\mathbf{B}}_2)(\hat{\mathbf{B}}_1^*\hat{\mathbf{S}}_1 + \hat{\mathbf{B}}_2^*\hat{\mathbf{S}}_2\mathbf{P}_{j_2})\big)\big)\\ &= \mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_{2,\mathsf{norm}}^{\mathsf{T}}\hat{\mathbf{S}}_1 + \mathbf{r}_{2,\mathsf{sf}}^{\mathsf{T}}\hat{\mathbf{S}}_2\mathbf{P}_{j_2})\big)\\ &= \mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_{2,\mathsf{norm}}^{\mathsf{T}}\hat{\mathbf{S}}_1)\big) + \mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_{2,\mathsf{sf}}^{\mathsf{T}}\hat{\mathbf{S}}_2\mathbf{P}_{j_2})\big),\end{aligned}$$

which matches the distribution in Hyb_6. □

Lemma 4.48. $\Pr[\mathsf{Hyb}_6(\mathcal{A}) = 1] = \Pr[\mathsf{Hyb}_7(\mathcal{A}) = 1]$.

Proof. The claim follows by properties of the projection matrix (Lemma 4.37). Specifically, we will show that the following two distributions are identically distributed over the choice of $\mathbf{U}$:

$$\left\{\mathbf{U}(\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1}) - \mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_{2,\mathsf{sf}}^{\mathsf{T}}\hat{\mathbf{S}}_2\mathbf{P}_{j_1})\big) : \mathbf{U} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^3\times\ell^2}\right\} \equiv \left\{\mathbf{U}(\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1}) : \mathbf{U} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^3\times\ell^2}\right\}. \tag{4.42}$$

Since $(j_1, j_2) \in S$ and moreover, $\mathbf{P}_{\mathsf{quad}} = \mathbf{P}_{\mathsf{quad}}^{(S)}$, we can appeal to Lemma 4.37 (applied to the vector $\mathbf{r}_{2,\mathsf{sf}}^{\mathsf{T}}\hat{\mathbf{S}}_2$) to conclude that

$$\mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_{2,\mathsf{sf}}^{\mathsf{T}}\hat{\mathbf{S}}_2\mathbf{P}_{j_2})\big)\big(\mathbf{I}_{\ell^2} - (\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1})\big) = \mathbf{0}.$$

Now, we can write

$$\begin{aligned}\mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_{2,\mathsf{sf}}^{\mathsf{T}}\hat{\mathbf{S}}_2\mathbf{P}_{j_2})\big) &= \mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_{2,\mathsf{sf}}^{\mathsf{T}}\hat{\mathbf{S}}_2\mathbf{P}_{j_2})\big)\big((\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1}) + \mathbf{I}_{\ell^2} - (\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1})\big)\\ &= \mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_{2,\mathsf{sf}}^{\mathsf{T}}\hat{\mathbf{S}}_2\mathbf{P}_{j_2})\big)(\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1}).\end{aligned}$$

This means that

$$\mathbf{U}(\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1}) - \mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_{2,\mathsf{sf}}^{\mathsf{T}}\hat{\mathbf{S}}_2\mathbf{P}_{j_2})\big) = \Big(\mathbf{U} - \mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_{2,\mathsf{sf}}^{\mathsf{T}}\hat{\mathbf{S}}_2\mathbf{P}_{j_2})\big)\Big)(\mathbf{P}_{j_1} \otimes \mathbf{P}_{j_1}). \tag{4.43}$$

Since $\mathbf{U}$ is uniform over $\mathbb{Z}_p^{\ell^3\times\ell^2}$ and independent of $\mathbf{P}_{\mathsf{quad}}$, $\mathbf{r}_{2,\mathsf{sf}}$, $\hat{\mathbf{S}}_2$, and $\mathbf{P}_{j_2}$, it follows that

$$\left\{\mathbf{U} - \mathbf{P}_{\mathsf{quad}}\big(\mathbf{I}_{\ell^2} \otimes \mathrm{vec}(\mathbf{r}_{2,\mathsf{sf}}^{\mathsf{T}}\hat{\mathbf{S}}_2\mathbf{P}_{j_2})\big) : \mathbf{U} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^3\times\ell^2}\right\} \equiv \left\{\mathbf{U} : \mathbf{U} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^{\ell^3\times\ell^2}\right\}. \tag{4.44}$$

Eq. (4.42) now follows by combining Eqs. (4.43) and (4.44). □

Lemma 4.49. *There exists a negligible function* $\mathsf{negl}(\cdot)$ *such that* $\Pr[\mathsf{Hyb}_7(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$.

Proof. By construction in Hyb_7, the components of crs are *independent* of the vector $\mathbf{r}_{2,\mathsf{sf}}$. This means that the challenger in Hyb_7 can defer the sampling of $\mathbf{r}_{2,\mathsf{sf}}$ until *after* the adversary outputs $(\mathbf{M}, [\mathbf{c}_*]_2, [\hat{\mathbf{c}}]_2, [\mathbf{v}]_2)$. For the challenger to output 1 in Hyb_7, it must be the case that $\hat{\mathbf{B}}_2\hat{\mathbf{c}} \neq \mathbf{0}$ and $\mathbf{R}\hat{\mathbf{c}} + \mathbf{v}' = \mathbf{0}$, where $\mathbf{v}' = \mathbf{v} - (\mathrm{vec}(\mathbf{M})^{\mathsf{T}} \otimes \mathbf{I}_{k+1})\mathbf{W}_{\mathsf{norm}}\mathbf{D}_{\mathsf{norm}}\mathbf{c}_*$. We argue that when $\hat{\mathbf{B}}_2\hat{\mathbf{c}} \neq \mathbf{0}$, the probability that $\mathbf{R}\hat{\mathbf{c}} + \mathbf{v}' = \mathbf{0}$ is negligible when taken over the choice of $\mathbf{r}_{2,\mathsf{sf}}$. Since $\mathbf{R} = \mathbf{R}_1 + \mathbf{a}^{\perp}\mathbf{r}_2^{\mathsf{T}} = \mathbf{R}_1 + \mathbf{a}^{\perp}\mathbf{r}_{2,\mathsf{norm}}\hat{\mathbf{B}}_1 + \mathbf{a}^{\perp}\mathbf{r}_{2,\mathsf{sf}}^{\mathsf{T}}\hat{\mathbf{B}}_2$, the equation $\mathbf{R}\hat{\mathbf{c}} + \mathbf{v}' = \mathbf{0}$ holds only if

$$\mathbf{a}^{\perp} \cdot \mathbf{r}_{2,\mathsf{sf}}^{\mathsf{T}}\hat{\mathbf{B}}_2\hat{\mathbf{c}} = -\mathbf{v}' - \mathbf{R}_1\hat{\mathbf{c}} - \mathbf{a}^{\perp} \cdot \mathbf{r}_{2,\mathsf{norm}}^{\mathsf{T}}\hat{\mathbf{B}}_1\hat{\mathbf{c}} \in \mathbb{Z}_p^{k+1}.$$

Since $\hat{\mathbf{B}}_2\hat{\mathbf{c}} \neq \mathbf{0}$ and $\mathbf{r}_{2,\mathsf{sf}} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$, the distribution of $\mathbf{r}_{2,\mathsf{sf}}^{\mathsf{T}}\hat{\mathbf{B}}_2\hat{\mathbf{c}}$ is uniform over $\mathbb{Z}_p$. Finally, since $\mathbf{a}^{\perp} \neq \mathbf{0}$ and the challenger samples $\mathbf{r}_{2,\mathsf{sf}} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k$ *after* all other quantities have been fixed, we conclude that

$$\Pr\left[\mathbf{a}^{\perp} \cdot \mathbf{r}_{2,\mathsf{sf}}^{\mathsf{T}}\hat{\mathbf{B}}_2\hat{\mathbf{c}} = -\mathbf{v}' - \mathbf{R}_1\hat{\mathbf{c}} - \mathbf{a}^{\perp} \cdot \mathbf{r}_{2,\mathsf{norm}}^{\mathsf{T}}\hat{\mathbf{B}}_1\hat{\mathbf{c}} : \mathbf{r}_{2,\mathsf{sf}} \xleftarrow{\mathrm{R}} \mathbb{Z}_p^k\right] \le \frac{1}{p} = \mathsf{negl}(\lambda).$$ □

By Lemmas 4.42 to 4.49, we conclude that for all efficient adversaries $\mathcal{A}$, $\Pr[\mathsf{Hyb}_0(\mathcal{A}) = 1] \le \mathsf{negl}(\lambda)$. This means that Construction 4.38 satisfies homogeneous chain binding for quadratic functions. Finally, since the vector dimension $\ell = \mathrm{poly}(\lambda)$, the bilateral k-Lin assumption implies the $\mathsf{MDDH}_{k,\ell,\ell,\ell^3}$ assumption in $\mathbb{G}_2$ (Lemma 3.10 and Remark 3.8) as well as the k-KerLin assumption in $\mathbb{G}_1$. Theorem 4.40 now follows from Lemma 4.41. □

5 Functional Commitments for all Circuits

In this section, we describe how to use our projective chainable commitments for to obtain a functional commitment for arithmetic circuits. In our construction, we will use the following representation for arithmetic circuits:

Definition 5.1 (Arithmetic Circuit Representation)**.** Let $\mathcal{R}$ be a ring, and let $C \colon \mathcal{R}^\ell \to \mathcal{R}^m$ be an arithmetic circuit (consisting of binary addition and multiplication gates) with s wires. We define the "next-wire" matrix $\mathbf{M}_C \in \mathcal{R}^{(s+1)\times(s+1)^2}$ associated with C as follows:

- Index each wire in C in topological order. Specifically, the input wires are associated with the indices $1, \ldots, \ell$, and the output wires are associated with indices $s - m + 1, \ldots, s$. The value of each intermediate wire i is a (quadratic) function of the values of the wires indexed $\{1, \ldots, i - 1\}$. We assume that there is a canonical topological ordering for the wires of C.
- For an input $\mathbf{x} \in \mathcal{R}^\ell$, let $\mathbf{z} \in \mathcal{R}^s$ be the vector of wire values associated with $C(\mathbf{x})$ under the canonical wire ordering. Let $\hat{\mathbf{z}} = \left[\begin{smallmatrix} 1 \\ \mathbf{z} \end{smallmatrix}\right]$. In the following description, we write $\hat{z}_0 = 1$ to refer to the first entry of $\hat{\mathbf{z}}$ and $\hat{z}_1, \ldots, \hat{z}_s$ to refer to the remaining entries.
- Let $S = \{(j, j + 1) : j \in \{\ell + 1, \ldots, s\}\}$. We define $\mathbf{M}_C \in \mathcal{R}^{(s+1)\times(s+1)}$ to be an S-local homogeneous quadratic mapping that satisfies $\mathbf{M}_C(\hat{\mathbf{z}} \otimes \hat{\mathbf{z}}) = \hat{\mathbf{z}}$:
  - For $i \in \{0, \ldots, \ell\}$, the i^{th} row of $\mathbf{M}_C$ implements the identity mapping $\hat{z}_i \mapsto \hat{z}_0\hat{z}_i$.
  - For $i \in \{\ell + 1, \ldots, s\}$, the i^{th} row of $\mathbf{M}_C$ implements the quadratic function associated with the gate computing the i^{th} wire of C. Since we index the wires of C in topological order, the value of the i^{th} wire is a quadratic function of the values of wires $1, \ldots, i - 1$, or equivalently, the variables $\hat{z}_1, \ldots, \hat{z}_{i-1}$. Finally, since we defined $\hat{z}_0 = 1$, we can express $\hat{z}_i$ as a *homogeneous* quadratic function of $\hat{z}_0, \ldots, \hat{z}_{i-1}$.

  By construction, for all $j \geq \ell + 1$, the first $j + 1$ outputs of $\mathbf{M}_C$ only depend on the first j values of $\hat{\mathbf{z}}$, so the function $\mathbf{M}_C$ is S-local, as desired.

Construction 5.2 (Functional Commitment for Arbitrary Functions)**.** Our functional commitment scheme will rely on the projective commitments and the associated proof systems from Section 4:

- Let $\mathsf{FC}_{\mathsf{base}} = \big(\mathsf{SetupBase}, \mathsf{SetupSF}, \mathsf{Commit}^{(1)}, \mathsf{Commit}^{(2)}, \mathsf{Project}^{(1)}, \mathsf{Project}^{(2)}\big)$ be the base projective commitment scheme (Definition 4.3).
- Let $\mathsf{FC}_{\mathsf{pre}} = \big(\mathsf{SetupPre}, \mathsf{OpenPre}, \mathsf{VerifyPre}\big)$ be a prefix-checking proof system for $\mathsf{FC}_{\mathsf{base}}$ (Definition 4.13).
- Let $\mathsf{FC}_{\mathsf{lin}} = \big(\mathsf{SetupLin}, \mathsf{OpenLin}, \mathsf{VerifyLin}\big)$ be a chainable proof system for local linear functions over $\mathsf{FC}_{\mathsf{base}}$ (Definition 4.20).
- Let $\mathsf{FC}_{\mathsf{quad}} = \big(\mathsf{SetupQuad}, \mathsf{OpenQuad}, \mathsf{VerifyQuad}\big)$ be a chainable proof system for local quadratic functions over $\mathsf{FC}_{\mathsf{base}}$ (Definition 4.35).

We construct our functional commitment scheme $\mathsf{FC} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Eval}, \mathsf{Verify})$ for arithmetic circuits as follows:

- $\mathsf{Setup}(1^\lambda, 1^\ell, 1^s)$: On input the security parameter λ, the input length ℓ, and the circuit size s, the setup algorithm starts by sampling a CRS for the base projective commitment scheme $\mathsf{crs}_{\mathsf{base}} \leftarrow \mathsf{SetupBase}(1^\lambda, 1^{s+1})$. It samples parameters for each of the underlying proof systems:
  - $\mathsf{crs}_{\mathsf{pre}} \leftarrow \mathsf{SetupPre}(\mathsf{crs}_{\mathsf{base}}, \ell + 1)$.
  - $\mathsf{crs}_{\mathsf{lin}} \leftarrow \mathsf{SetupLin}(\mathsf{crs}_{\mathsf{base}}, S_{\mathsf{lin}})$ where $S_{\mathsf{lin}} = \{(j, j) : j \in [s + 1]\}$.
  - $\mathsf{crs}_{\mathsf{quad}} \leftarrow \mathsf{SetupQuad}(\mathsf{crs}_{\mathsf{base}}, S_{\mathsf{quad}})$ where $S_{\mathsf{quad}} = \{(j, j + 1) : j \in \{\ell + 1, \ldots, s\}\}$.

  The setup algorithm outputs
  $$\mathsf{crs} = \big(1^s, \mathsf{crs}_{\mathsf{base}}, \mathsf{crs}_{\mathsf{pre}}, \mathsf{crs}_{\mathsf{lin}}, \mathsf{crs}_{\mathsf{quad}}\big).$$
  The input ring associated with crs is the same as that associated with $\mathsf{crs}_{\mathsf{base}}$.

- $\mathsf{Commit}(\mathsf{crs}, \mathbf{x})$: On input the common reference string $\mathsf{crs} = (1^s, \mathsf{crs}_{\mathsf{base}}, \mathsf{crs}_{\mathsf{pre}}, \mathsf{crs}_{\mathsf{lin}}, \mathsf{crs}_{\mathsf{quad}})$, an input $\mathbf{x} \in \mathcal{R}^\ell$ (where $\ell \leq s$), the commit algorithm outputs the commitment

$$\sigma_{\mathsf{in}} \leftarrow \mathsf{Commit}^{(1)}_{\mathsf{base}}(\mathsf{crs}_{\mathsf{base}}, \hat{\mathbf{x}}) \quad \text{where} \quad \hat{\mathbf{x}} = \begin{bmatrix} 1 \\ \mathbf{x} \\ \mathbf{0}^{s-\ell} \end{bmatrix} \in \mathcal{R}^{s+1} \tag{5.1}$$

and the state $\mathsf{st} = \hat{\mathbf{x}}$.

- $\mathsf{Eval}(\mathsf{crs}, \mathsf{st}, C)$: On input the common reference string $\mathsf{crs} = (1^s, \mathsf{crs}_{\mathsf{base}}, \mathsf{crs}_{\mathsf{pre}}, \mathsf{crs}_{\mathsf{lin}}, \mathsf{crs}_{\mathsf{quad}})$, the state $\mathsf{st} = \hat{\mathbf{x}}$ (parsed into $\mathbf{x} \in \mathcal{R}^\ell$ according to Eq. (5.1)), and an arithmetic circuit $C \colon \mathcal{R}^\ell \to \mathcal{R}^m$ of size s, the evaluation algorithm starts by computing the following quantities:
  - Let $\mathbf{z} \in \mathcal{R}^s$ be the vector of wire values associated with $C(\mathbf{x})$ (as defined in Definition 5.1), and let $\hat{\mathbf{z}} = \begin{bmatrix} 1 \\ \mathbf{z} \end{bmatrix}$.
  - Compute commitments $\sigma_1 \leftarrow \mathsf{Commit}^{(1)}(\mathsf{crs}_{\mathsf{base}}, \hat{\mathbf{z}})$ and $\sigma_2 \leftarrow \mathsf{Commit}^{(2)}(\mathsf{crs}_{\mathsf{base}}, \hat{\mathbf{z}})$ to the wire values $\hat{\mathbf{z}}$.

  Then, it prepares the following openings:
  - **Input consistency:** Compute $\pi_{\mathsf{pre}} \leftarrow \mathsf{OpenPre}(\mathsf{crs}_{\mathsf{pre}}, \hat{\mathbf{x}}, \hat{\mathbf{z}})$.
  - **Internal consistency**: Compute $\pi_{\mathsf{lin}} \leftarrow \mathsf{OpenLin}(\mathsf{crs}_{\mathsf{lin}}, \hat{\mathbf{z}}, \mathbf{I}_{s+1})$.
  - **Gate consistency:** Compute $\pi_{\mathsf{quad}} \leftarrow \mathsf{OpenQuad}(\mathsf{crs}_{\mathsf{quad}}, \hat{\mathbf{z}}, \mathbf{M}_C)$, where $\mathbf{M}_C$ is the "next-wire" matrix associated with C (Definition 5.1).
  - **Output consistency**: Let $\mathbf{P}_{\mathsf{out}} = \mathrm{diag}([\mathbf{0}^{1\times(s+1-m)} \mid \mathbf{1}^{1\times m}]) \in \{0,1\}^{(s+1)\times(s+1)}$ be the matrix that projects onto the last m components. Compute the opening $\pi_{\mathsf{out}} \leftarrow \mathsf{OpenLin}(\mathsf{crs}_{\mathsf{lin}}, \hat{\mathbf{z}}, \mathbf{P}_{\mathsf{out}})$.

  Finally, it outputs the proof $\pi = (\sigma_1, \sigma_2, \pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}})$.

- $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}, \pi)$: On input the common reference string $\mathsf{crs} = (1^s, \mathsf{crs}_{\mathsf{base}}, \mathsf{crs}_{\mathsf{pre}}, \mathsf{crs}_{\mathsf{lin}}, \mathsf{crs}_{\mathsf{quad}})$, the input commitment σ_{in}, a function $f \colon \mathcal{R}^\ell \to \mathcal{R}^m$, an output $\mathbf{y} \in \mathcal{R}^m$, and a proof $\pi = (\sigma_1, \sigma_2, \pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}})$, the verification algorithm computes $\sigma_{\mathsf{out}} \leftarrow \mathsf{Commit}^{(2)}(\mathsf{crs}_{\mathsf{base}}, \begin{bmatrix} \mathbf{0} \\ \mathbf{y} \end{bmatrix})$ and checks each of the following properties:
  - **Input consistency:** $\mathsf{VerifyPre}(\mathsf{crs}_{\mathsf{pre}}, \sigma_{\mathsf{in}}, \sigma_1, \pi_{\mathsf{pre}}) = 1$.
  - **Internal consistency**: $\mathsf{VerifyLin}(\mathsf{crs}_{\mathsf{lin}}, \sigma_1, \mathbf{I}_{s+1}, \sigma_2, \pi_{\mathsf{lin}}) = 1$.
  - **Gate consistency:** $\mathsf{VerifyQuad}(\mathsf{crs}_{\mathsf{quad}}, \sigma_2, \mathbf{M}_C, \sigma_1, \pi_{\mathsf{quad}}) = 1$, where $\mathbf{M}_C$ is the next-wire matrix associated with C (Definition 5.1).
  - **Output consistency**: $\mathsf{VerifyLin}(\mathsf{crs}_{\mathsf{lin}}, \sigma_1, \mathbf{P}_{\mathsf{out}}, \sigma_{\mathsf{out}}, \pi_{\mathsf{out}}) = 1$, where $\mathbf{P}_{\mathsf{out}} = \mathrm{diag}([\mathbf{0}^{1\times(s+1-m)} \mid \mathbf{1}^{1\times m}])$.

  The verification algorithm outputs 1 if all of the above checks pass and outputs 0 otherwise.

Theorem 5.3 (Correctness). *If $\mathsf{FC}_{\mathsf{pre}}$, $\mathsf{FC}_{\mathsf{lin}}$, and $\mathsf{FC}_{\mathsf{quad}}$ are correct, then Construction 5.2 is correct.*

Proof. Let $\lambda, \ell, s \in \mathbb{N}$. Let $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\ell, 1^s)$. Let $\mathcal{R}$ be the input ring associated with crs and let $C \colon \mathcal{R}^\ell \to \mathcal{R}^m$ be an arbitrary arithmetic circuit of size s. Take any $\mathbf{x} \in \mathcal{R}^\ell$. Suppose $(\sigma_{\mathsf{in}}, \mathsf{st}) \leftarrow \mathsf{Commit}(\mathsf{crs}, \mathbf{x})$ and $\pi \leftarrow \mathsf{Eval}(\mathsf{crs}, \mathsf{st}, C)$. By construction, $\mathsf{crs} = (1^s, \mathsf{crs}_{\mathsf{base}}, \mathsf{crs}_{\mathsf{pre}}, \mathsf{crs}_{\mathsf{lin}}, \mathsf{crs}_{\mathsf{quad}})$, $\sigma_{\mathsf{in}} \leftarrow \mathsf{Commit}^{(1)}(\mathsf{crs}_{\mathsf{base}}, \hat{\mathbf{x}})$ where $\hat{\mathbf{x}} = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix}$, and $\pi = (\sigma_1, \sigma_2, \pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}})$. In addition, $\sigma_1 \leftarrow \mathsf{Commit}^{(1)}(\mathsf{crs}_{\mathsf{base}}, \hat{\mathbf{z}})$ and $\sigma_2 \leftarrow \mathsf{Commit}^{(2)}(\mathsf{crs}_{\mathsf{base}}, \hat{\mathbf{z}})$. Consider the output of $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, C(\mathbf{x}), \pi)$.

- **Input consistency:** By definition, $\hat{\mathbf{z}} = \begin{bmatrix} 1 \\ \mathbf{z} \end{bmatrix}$, where $\mathbf{z}$ is the vector of wire values associated with $C(\mathbf{x})$. By definition, the first ℓ components of $\mathbf{z}$ is exactly $\mathbf{x}$. This means $\hat{\mathbf{x}}$ and $\hat{\mathbf{z}}$ share a common prefix of length $\ell + 1$. Since $\pi_{\mathsf{pre}} \leftarrow \mathsf{OpenPre}(\mathsf{crs}_{\mathsf{pre}}, \hat{\mathbf{x}}, \hat{\mathbf{z}})$, correctness of $\mathsf{FC}_{\mathsf{pre}}$ now says that $\mathsf{VerifyPre}(\mathsf{crs}_{\mathsf{pre}}, \sigma_{\mathsf{in}}, \sigma_1, \pi_{\mathsf{pre}}) = 1$.

- **Internal consistency:** Since σ_1 and σ_2 are both commitments to $\hat{\mathbf{z}}$, the identity mapping $\hat{\mathbf{z}} \mapsto \mathbf{I}_{s+1}\hat{\mathbf{z}}$ is S_{lin}-local, and $\pi_{\mathsf{lin}} \leftarrow \mathsf{OpenLin}(\mathsf{crs}_{\mathsf{lin}}, \hat{\mathbf{z}}, \mathbf{I}_{s+1})$, correctness of $\mathsf{FC}_{\mathsf{lin}}$ implies $\mathsf{VerifyLin}(\mathsf{crs}_{\mathsf{lin}}, \sigma_1, \mathbf{I}_{s+1}, \sigma_2, \pi_{\mathsf{lin}}) = 1$.

- **Gate consistency:** From Definition 5.1, the mapping $\mathbf{M}_C$ is S_{quad}-local and moreover, $\mathbf{M}_C\hat{\mathbf{z}} = \hat{\mathbf{z}}$. Since $\pi_{\mathsf{quad}} \leftarrow \mathsf{OpenQuad}(\mathsf{crs}_{\mathsf{quad}}, \hat{\mathbf{z}}, \mathbf{M}_C)$, correctness of $\mathsf{FC}_{\mathsf{quad}}$ implies $\mathsf{VerifyQuad}(\mathsf{crs}_{\mathsf{quad}}, \sigma_2, \mathbf{M}_C, \sigma_1, \pi_{\mathsf{quad}}) = 1$.

Hybrid	(j_1, j_2)	$\mathsf{Project}^{(1)}$	$\mathsf{Project}^{(2)}$	**Justification**	
$\mathsf{Hyb}_{\mathsf{real}}$	—	✗	✗		
$\mathsf{Hyb}_{\mathsf{sf}}$	$(\ell+1, \ell+1)$	✗	✗	Mode Indistinguishability	(Definition 4.4)
$\mathsf{Hyb}_{\ell+1,0}$	$(\ell+1, \ell+1)$	✓	✗	Prefix Matching	(Definition 4.13)
$\mathsf{Hyb}_{i,0}$	(i, i)	✓	✗		
$\mathsf{Hyb}_{i,1}$	(i, i)	✓	✓	Linear Chain Binding	(Definition 4.20)
$\mathsf{Hyb}_{i,2}$	(i, i)	✗	✓	Dropping Verification Condition	
$\mathsf{Hyb}_{i,3}$	$(i+1, i)$	✗	✓	Type-I Indistinguishability	(Definition 4.5)
$\mathsf{Hyb}_{i,4}$	$(i+1, i)$	✓	✓	Quadratic Chain Binding	(Definition 4.35)
$\mathsf{Hyb}_{i,5}$	$(i+1, i)$	✓	✗	Dropping Verification Condition	
$\mathsf{Hyb}_{i+1,0}$	$(i+1, i+1)$	✓	✗	Type-II Indistinguishability	(Definition 4.6)

Table 2: Overview of main hybrid experiments in the proof of Theorem 5.4. For each hybrid, we provide the Type-I projection index j_1 and the Type-II projection index j_2 associated with the (semi-functional) common reference string. We also indicate whether each experiment is checking the consistency of the Type-I commitments using $\mathsf{Project}^{(1)}$ (which requires knowledge of td_1) and the consistency of the Type-II commitments using $\mathsf{Project}^{(2)}$ (which requires knowledge of td_2). The justification column lists the reason why the adversary's advantage from one experiment to the next cannot *decrease* by a non-negligible amount.

- **Output consistency:** By the convention in Definition 5.1, the last m components of $\hat{\mathbf{z}}$ correspond to the outputs of $C(\mathbf{x})$. This means that $\mathbf{P}_{\mathsf{out}}\hat{\mathbf{z}} = \left[\begin{smallmatrix} \mathbf{0} \\ \mathbf{y} \end{smallmatrix}\right]$, where $\mathbf{y} = C(\mathbf{x})$. Next, the verification algorithm computes $\sigma_{\mathsf{out}} \leftarrow \mathsf{Commit}^{(2)}(\mathsf{crs}_{\mathsf{base}}, \left[\begin{smallmatrix} \mathbf{0} \\ \mathbf{y} \end{smallmatrix}\right])$. In addition, $\mathbf{P}_{\mathsf{out}}$ is diagonal so it is also S_{lin}-local. Since $\pi_{\mathsf{out}} \leftarrow \mathsf{OpenLin}(\mathsf{crs}_{\mathsf{lin}}, \sigma_1, \mathbf{P}_{\mathsf{out}}, \sigma_{\mathsf{out}}, \pi_{\mathsf{out}})$, correctness of $\mathsf{FC}_{\mathsf{lin}}$ implies that $\mathsf{VerifyLin}(\mathsf{crs}_{\mathsf{lin}}, \sigma_1, \mathbf{P}_{\mathsf{out}}, \sigma_{\mathsf{out}}, \pi_{\mathsf{out}}) = 1$.

Since each of the checks pass, the verification algorithm outputs 1 and correctness holds. □

Theorem 5.4 (Binding). *Suppose* $\mathsf{FC}_{\mathsf{base}}$ *satisfies mode indistinguishability, Type-I indistinguishability, Type-II indistinguishability, and Type-II collision resistance. Suppose also that* $\mathsf{FC}_{\mathsf{pre}}$, $\mathsf{FC}_{\mathsf{lin}}$, *and* $\mathsf{FC}_{\mathsf{quad}}$ *are secure. Then, Construction 5.2 is binding.*

Proof. We start by defining a sequence of hybrid experiments:

- $\mathsf{Hyb}_{\mathsf{real}}$: This is the real binding experiment.
  1. Algorithm $\mathcal{A}$ starts by outputting the input length 1^ℓ and the circuit size 1^s.
  2. The challenger samples the base common reference string $\mathsf{crs}_{\mathsf{base}} \leftarrow \mathsf{SetupBase}(1^\lambda, 1^{s+1})$ and
  $$\begin{aligned} \mathsf{crs}_{\mathsf{pre}} &\leftarrow \mathsf{SetupPre}(\mathsf{crs}_{\mathsf{base}}, \ell+1) \\ \mathsf{crs}_{\mathsf{lin}} &\leftarrow \mathsf{SetupLin}(\mathsf{crs}_{\mathsf{base}}, S_{\mathsf{lin}}) \\ \mathsf{crs}_{\mathsf{quad}} &\leftarrow \mathsf{SetupQuad}(\mathsf{crs}_{\mathsf{base}}, S_{\mathsf{quad}}), \end{aligned}$$
  where the locality sets S_{lin} and S_{quad} are defined as in Construction 5.2. The challenger replies to the adversary with $\mathsf{crs} = (1^s, \mathsf{crs}_{\mathsf{base}}, \mathsf{crs}_{\mathsf{pre}}, \mathsf{crs}_{\mathsf{lin}}, \mathsf{crs}_{\mathsf{quad}})$. Let $\mathcal{R}$ be the input ring associated with $\mathsf{crs}_{\mathsf{base}}$.
  3. The adversary $\mathcal{A}$ outputs an input commitment σ_{in}, an arithmetic circuit $C \colon \mathcal{R}^\ell \to \mathcal{R}^m$, vectors $\mathbf{y}, \mathbf{y}' \in \mathcal{R}^m$, and openings $\pi = (\sigma_1, \sigma_2, \pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}})$ and $\pi' = (\sigma_1', \sigma_2', \pi_{\mathsf{pre}}', \pi_{\mathsf{lin}}', \pi_{\mathsf{quad}}', \pi_{\mathsf{out}}')$.
  4. The output of the experiment is 1 if $\mathbf{y} \neq \mathbf{y}'$, $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}, \pi) = 1$ and $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}', \pi') = 1$.
- $\mathsf{Hyb}_{\mathsf{sf}}$: Same as $\mathsf{Hyb}_{\mathsf{real}}$, except the challenger samples $\mathsf{crs}_{\mathsf{base}}$ in semi-functional mode:
$$(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^\lambda, 1^{s+1}, \ell+1, \ell+1).$$

- $\mathsf{Hyb}_{i,0}$ for $i \in \{\ell+1, \ldots, s+1\}$: Same as $\mathsf{Hyb}_{\mathsf{sf}}$ except when setting up the CRS, the challenger samples
$$(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^\lambda, 1^{s+1}, i, i).$$
Moreover, the output of the experiment is 1 only if the following hold:
  - $\mathbf{y} \neq \mathbf{y}'$ and $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}, \pi) = 1$ and $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}', \pi') = 1$.
  - $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$.
- $\mathsf{Hyb}_{i,1}$ for $i \in \{\ell+1, \ldots, s\}$: Same as $\mathsf{Hyb}_{i,0}$ except the output of the experiment is 1 only if the following hold:
  - $\mathbf{y} \neq \mathbf{y}'$ and $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}, \pi) = 1$ and $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}', \pi') = 1$.
  - $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$.
  - $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) = \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2')$.
- $\mathsf{Hyb}_{i,2}$ for $i \in \{\ell+1, \ldots, s\}$: Same as $\mathsf{Hyb}_{i,1}$ except the output of the experiment is 1 only if the following hold:
  - $\mathbf{y} \neq \mathbf{y}'$ and $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}, \pi) = 1$ and $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}', \pi') = 1$.
  - $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) = \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2')$.

  In particular, the challenger no longer checks the projection on σ_1, σ_1'.
- $\mathsf{Hyb}_{i,3}$ for $i \in \{\ell+1, \ldots, s\}$: Same as $\mathsf{Hyb}_{i,2}$ except when setting up the CRS, the challenger samples
$$(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^\lambda, 1^{s+1}, i+1, i).$$
- $\mathsf{Hyb}_{i,4}$ for $i \in \{\ell+1, \ldots, s\}$: Same as $\mathsf{Hyb}_{i,3}$ except the output of the experiment is 1 only if the following hold:
  - $\mathbf{y} \neq \mathbf{y}'$ and $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}, \pi) = 1$ and $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}', \pi') = 1$.
  - $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$.
  - $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) = \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2')$.
- $\mathsf{Hyb}_{i,5}$ for $i \in \{\ell+1, \ldots, s\}$: Same as $\mathsf{Hyb}_{i,4}$ except the output of the experiment is 1 only if the following hold:
  - $\mathbf{y} \neq \mathbf{y}'$ and $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}, \pi) = 1$ and $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}', \pi') = 1$.
  - $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$.

  In particular, the challenger no longer checks the projection on σ_2, σ_2'.
- $\mathsf{Hyb}_{\mathsf{final}}$: Same as $\mathsf{Hyb}_{s+1,0}$, where the challenger samples
$$(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^\lambda, 1^{s+1}, s+1, s+1).$$
At the end of the experiment, after the adversary outputs $\sigma_2, C, \mathbf{y}, \mathbf{y}'$ and π, π', the challenger computes
$$\sigma_{\mathsf{out}} \leftarrow \mathsf{Commit}^{(2)}(\mathsf{crs}_{\mathsf{base}}, \left[\begin{smallmatrix}\mathbf{0}\\ \mathbf{y}\end{smallmatrix}\right]) \quad \text{and} \quad \sigma_{\mathsf{out}}' \leftarrow \mathsf{Commit}^{(2)}(\mathsf{crs}_{\mathsf{base}}, \left[\begin{smallmatrix}\mathbf{0}\\ \mathbf{y}'\end{smallmatrix}\right]).$$
The output of the experiment is 1 only if the following hold:
  - $\mathbf{y} \neq \mathbf{y}'$ and $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}, \pi) = 1$ and $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}', \pi') = 1$.
  - $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$.
  - $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_{\mathsf{out}}) = \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_{\mathsf{out}}')$.

Take any efficient adversary $\mathcal{A}$ for the binding game. Let ℓ be the input length and s be the circuit size chosen by $\mathcal{A}$. We write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output distribution of an execution of Hyb_i with adversary $\mathcal{A}$. We now show that the probability of a hybrid outputting 1 *cannot* decrease by a non-negligible amount as we move from one hybrid to the next. Then, we show that in the final hybrid $\mathsf{Hyb}_{\mathsf{final}}$, the probability that the challenger outputs 1 is negligible by Type-II collision-resistance of the underlying projective commitment (Definition 4.7). We summarize the key sequence of hybrid transitions in Table 2.

Lemma 5.5. *Suppose* $\mathsf{FC}_{\mathsf{base}}$ *satisfies mode indistinguishability (Definition 4.4). Then there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that* $|\Pr[\mathsf{Hyb}_{\mathsf{real}}(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_{\mathsf{sf}}(\mathcal{A}) = 1]| = \mathsf{negl}(\lambda)$.

Proof. Suppose $|\Pr[\mathsf{Hyb}_{\mathsf{real}}(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_{\mathsf{sf}}(\mathcal{A}) = 1]| \geq \varepsilon$ for some non-negligible ε. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ for the mode indistinguishability game:

1. Algorithm $\mathcal{B}$ starts running algorithm $\mathcal{A}$ which starts by outputting the input length 1^ℓ and the circuit size 1^s. Algorithm $\mathcal{B}$ sends $(1^{s+1}, \ell + 1, \ell + 1)$ to the mode indistinguishability challenger and receives $\mathsf{crs}_{\mathsf{base}}$.

2. Algorithm $\mathcal{B}$ samples

$$\begin{aligned} \mathsf{crs}_{\mathsf{pre}} &\leftarrow \mathsf{SetupPre}(\mathsf{crs}_{\mathsf{base}}, \ell + 1) \\ \mathsf{crs}_{\mathsf{lin}} &\leftarrow \mathsf{SetupLin}(\mathsf{crs}_{\mathsf{base}}, S_{\mathsf{lin}}) \\ \mathsf{crs}_{\mathsf{quad}} &\leftarrow \mathsf{SetupQuad}(\mathsf{crs}_{\mathsf{base}}, S_{\mathsf{quad}}), \end{aligned}$$

   It give $\mathsf{crs} = \left(1^s, \mathsf{crs}_{\mathsf{base}}, \mathsf{crs}_{\mathsf{pre}}, \mathsf{crs}_{\mathsf{lin}}, \mathsf{crs}_{\mathsf{quad}}\right)$ to $\mathcal{A}$.

3. Algorithm $\mathcal{A}$ outputs an input commitment σ_{in}, an arithmetic circuit $C \colon \mathcal{R}^\ell \to \mathcal{R}^m$, vectors $\mathbf{y}, \mathbf{y}' \in \mathcal{R}^m$, and openings $\pi = (\sigma_1, \sigma_2, \pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}})$ and $\pi' = (\sigma_1', \sigma_2', \pi_{\mathsf{pre}}', \pi_{\mathsf{lin}}', \pi_{\mathsf{quad}}', \pi_{\mathsf{out}}')$.

4. Algorithm $\mathcal{B}$ outputs 1 if $\mathbf{y} \neq \mathbf{y}'$, $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}, \pi) = 1$ and $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}', \pi') = 1$. Otherwise, it outputs 0.

By construction, if the challenger sampled $\mathsf{crs}_{\mathsf{base}} \leftarrow \mathsf{SetupBase}(1^\lambda, 1^{s+1})$, the algorithm $\mathcal{B}$ perfectly simulates an execution of $\mathsf{Hyb}_{\mathsf{real}}$ for $\mathcal{A}$ and outputs 1 with probability $\Pr[\mathsf{Hyb}_{\mathsf{real}}(\mathcal{A}) = 1]$. If the challenger sampled $\mathsf{crs}_{\mathsf{base}} \leftarrow \mathsf{SetupSF}(1^\lambda, 1^{s+1}, \ell+1, \ell+1)$, then algorithm $\mathcal{B}$ perfectly simulates an execution of $\mathsf{Hyb}_{\mathsf{sf}}$ for $\mathcal{A}$ and outputs 1 with probability $\Pr[\mathsf{Hyb}_{\mathsf{sf}}(\mathcal{A}) = 1]$. Thus, algorithm $\mathcal{B}$ breaks mode indistinguishability with advantage ε. ◻

Lemma 5.6. *Suppose* $\mathsf{FC}_{\mathsf{pre}}$ *satisfies prefix-matching security (Definition 4.13). Then there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that* $|\Pr[\mathsf{Hyb}_{\mathsf{sf}}(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_{\ell+1,0}(\mathcal{A}) = 1]| = \mathsf{negl}(\lambda)$.

Proof. Suppose $|\Pr[\mathsf{Hyb}_{\mathsf{sf}}(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_{\ell+1,0}(\mathcal{A}) = 1]| \geq \varepsilon$ for some non-negligible ε. By construction, the common reference string crs in the two experiments is identically distributed. Thus, it must be the case that with probability at least ε, algorithm $\mathcal{A}$ will output σ_{in}, C, $\mathbf{y}, \mathbf{y}'$, $\pi = (\sigma_1, \sigma_2, \pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}})$ and $\pi' = (\sigma_1', \sigma_2', \pi_{\mathsf{pre}}', \pi_{\mathsf{lin}}', \pi_{\mathsf{quad}}', \pi_{\mathsf{out}}')$ such that

$$\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}, \pi) = 1 = \mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}', \pi') \quad \text{and} \quad \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) \neq \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1'). \tag{5.2}$$

In all other cases, the outputs of $\mathsf{Hyb}_{\mathsf{sf}}$ and $\mathsf{Hyb}_{\ell+1,0}$ are identical. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ for the prefix matching security game:

1. Algorithm $\mathcal{B}$ runs algorithm $\mathcal{A}$, which starts by outputting the input length 1^ℓ and the circuit size 1^s. Algorithm $\mathcal{B}$ forwards $(1^{s+1}, \ell + 1)$ to the prefix matching challenger and receives $(\mathsf{crs}_{\mathsf{base}}, \mathsf{crs}_{\mathsf{pre}})$.

2. Algorithm $\mathcal{B}$ samples $\mathsf{crs}_{\mathsf{lin}} \leftarrow \mathsf{SetupLin}(\mathsf{crs}_{\mathsf{base}}, S_{\mathsf{lin}})$ and $\mathsf{crs}_{\mathsf{quad}} \leftarrow \mathsf{SetupQuad}(\mathsf{crs}_{\mathsf{base}}, S_{\mathsf{quad}})$. It gives the common reference string $\mathsf{crs} = \left(1^s, \mathsf{crs}_{\mathsf{base}}, \mathsf{crs}_{\mathsf{pre}}, \mathsf{crs}_{\mathsf{lin}}, \mathsf{crs}_{\mathsf{quad}}\right)$ to $\mathcal{A}$.

3. Algorithm $\mathcal{A}$ outputs an input commitment σ_{in}, an arithmetic circuit $C \colon \mathcal{R}^\ell \to \mathcal{R}^m$, vectors $\mathbf{y}, \mathbf{y}' \in \mathcal{R}^m$, and openings $\pi = (\sigma_1, \sigma_2, \pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}})$ and $\pi' = (\sigma_1', \sigma_2', \pi_{\mathsf{pre}}', \pi_{\mathsf{lin}}', \pi_{\mathsf{quad}}', \pi_{\mathsf{out}}')$.

4. Algorithm $\mathcal{B}$ samples a bit $b \xleftarrow{\mathrm{R}} \{0,1\}$. If $b = 0$, it outputs $(\sigma_{\mathsf{in}}, \sigma_1)$ and the opening π_{pre}. If $b = 1$, it outputs $(\sigma_{\mathsf{in}}, \sigma_1')$ and the opening π_{pre}'.

The prefix-matching security challenger samples $(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^\lambda, 1^{s+1}, \ell+1, \ell+1)$, so algorithm $\mathcal{B}$ perfectly simulates an execution of $\mathsf{Hyb}_{\mathsf{sf}}$ and $\mathsf{Hyb}_{\ell,0}$ for $\mathcal{A}$. Thus, with probability at least ε, the quantities output by $\mathcal{A}$ satisfy Eq. (5.2). Then, the following hold:

- If $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}, \pi) = 1 = \mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}', \pi')$, then we have that $\mathsf{VerifyPre}(\mathsf{crs}_{\mathsf{pre}}, \sigma_{\mathsf{in}}, \sigma_1, \pi_{\mathsf{pre}}) = 1$ and $\mathsf{VerifyPre}(\mathsf{crs}_{\mathsf{pre}}, \sigma_{\mathsf{in}}, \sigma_1', \pi_{\mathsf{pre}}') = 1$.
- If $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) \neq \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$, then it must be the case that

$$\text{either}\quad \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_{\mathsf{in}}) \neq \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) \quad\text{or}\quad \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_{\mathsf{in}}) \neq \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1').$$

Since algorithm $\mathcal{B}$ samples the bit b uniformly at random, it breaks prefix matching with probability at least $\varepsilon/2$. □

Lemma 5.7. *Suppose* $\mathsf{FC}_{\mathsf{lin}}$ *satisfies linear chain binding (Definition 4.20). Then there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that for all* $i \in \{\ell+1, \ldots, s\}$, $|\Pr[\mathsf{Hyb}_{i,0}(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_{i,1}(\mathcal{A}) = 1]| = \mathsf{negl}(\lambda)$.

Proof. Suppose there exists an index $i \in \{\ell+1, \ldots, s\}$ where $|\Pr[\mathsf{Hyb}_{i,0}(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_{i,1}(\mathcal{A}) = 1]| \geq \varepsilon$ for some non-negligible ε. By construction, the common reference string in the two experiments is identically distributed. Thus, it must be the case that with probability at least ε, algorithm $\mathcal{A}$ will output $\sigma_{\mathsf{in}}, C, \mathbf{y}, \mathbf{y}', \pi = (\sigma_1, \sigma_2, \pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}})$ and $\pi' = (\sigma_1', \sigma_2', \pi_{\mathsf{pre}}', \pi_{\mathsf{lin}}', \pi_{\mathsf{quad}}', \pi_{\mathsf{out}}')$ such that

- $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}, \pi) = 1 = \mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}', \pi')$.
- $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$.
- $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) \neq \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2')$.

In all other cases, the outputs of $\mathsf{Hyb}_{i,0}$ and $\mathsf{Hyb}_{i,1}$ are identical. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ for the linear chain binding game:

1. Algorithm $\mathcal{B}$ runs algorithm $\mathcal{A}$, which starts by outputting the input length 1^ℓ and the circuit size 1^s. Algorithm $\mathcal{B}$ provides 1^{s+1}, the locality set S_{lin} and indices (i,i) to the linear chain binding adversary. It receives $(\mathsf{crs}_{\mathsf{base}}, \mathsf{crs}_{\mathsf{lin}})$.
2. Algorithm $\mathcal{B}$ samples $\mathsf{crs}_{\mathsf{quad}} \leftarrow \mathsf{SetupQuad}(\mathsf{crs}_{\mathsf{base}})$ and $\mathsf{crs}_{\mathsf{pre}} \leftarrow \mathsf{SetupPre}(\mathsf{crs}_{\mathsf{base}}, \ell+1)$. It gives the common reference string $\mathsf{crs} = \left(1^s, \mathsf{crs}_{\mathsf{base}}, \mathsf{crs}_{\mathsf{pre}}, \mathsf{crs}_{\mathsf{lin}}, \mathsf{crs}_{\mathsf{quad}}\right)$ to $\mathcal{A}$.
3. Algorithm $\mathcal{A}$ outputs an input commitment σ_{in}, an arithmetic circuit $C \colon \mathcal{R}^\ell \to \mathcal{R}^m$, vectors $\mathbf{y}, \mathbf{y}' \in \mathcal{R}^m$, and openings $\pi = (\sigma_1, \sigma_2, \pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}})$ and $\pi' = (\sigma_1', \sigma_2', \pi_{\mathsf{pre}}', \pi_{\mathsf{lin}}', \pi_{\mathsf{quad}}', \pi_{\mathsf{out}}')$.
4. Algorithm $\mathcal{B}$ outputs the matrix $\mathbf{I}_{s+1}$, the Type-I commitments σ_1, σ_1', the Type-II commitments σ_2, σ_2', and the openings $\pi_{\mathsf{lin}}, \pi_{\mathsf{lin}}'$.

First, we note that $\mathcal{B}$ is a valid adversary for the chain binding security game. Namely, $(i,i) \in S_{\mathsf{lin}}$, and moreover, $\mathbf{I}_{s+1}$ is S_{lin}-local. Then, the challenger samples $(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^\lambda, 1^{s+1}, i, i)$, so algorithm $\mathcal{B}$ perfectly simulates an execution of $\mathsf{Hyb}_{i,0}$ and $\mathsf{Hyb}_{i,1}$ for $\mathcal{A}$. Thus, with probability at least ε, the quantities output by $\mathcal{A}$ satisfy the properties enumerated above. Then, the following hold:

- If $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}, \pi) = 1 = \mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}', \pi')$, then we have that $\mathsf{VerifyLin}(\mathsf{crs}_{\mathsf{lin}}, \sigma_1, \mathbf{I}_{s+1}, \sigma_2, \pi_{\mathsf{lin}}) = 1$ and $\mathsf{VerifyLin}(\mathsf{crs}_{\mathsf{lin}}, \sigma_1', \mathbf{I}_{s+1}, \sigma_2', \pi_{\mathsf{lin}}') = 1$.
- $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$.
- $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) \neq \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2')$.

These conditions precisely coincide with the requirements of the linear chain binding game, so we conclude that algorithm $\mathcal{B}$ succeeds with advantage ε. ◻

Lemma 5.8. *For all $i \in \{\ell+1,\ldots,s\}$, $\Pr[\mathsf{Hyb}_{i,1}(\mathcal{A})=1] \le \Pr[\mathsf{Hyb}_{i,2}(\mathcal{A})=1]$.*

Proof. The verification conditions in $\mathsf{Hyb}_{i,1}$ is a strict superset of those in $\mathsf{Hyb}_{i,2}$. Correspondingly, if $\mathsf{Hyb}_{i,1}(\mathcal{A})$ outputs 1, then the same is true for $\mathsf{Hyb}_{i,2}(\mathcal{A})$ and the claim holds. ◻

Lemma 5.9. *Suppose $\mathsf{FC}_{\mathsf{base}}$ satisfies Type-I indistinguishability (Definition 4.5). Then there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $i \in \{\ell+1,\ldots,s\}$, $|\Pr[\mathsf{Hyb}_{i,2}(\mathcal{A})=1] - \Pr[\mathsf{Hyb}_{i,3}(\mathcal{A})=1]| = \mathsf{negl}(\lambda)$.*

Proof. Suppose there exists an index $i \in \{\ell+1,\ldots,s\}$ where $|\Pr[\mathsf{Hyb}_{i,2}(\mathcal{A})=1] - \Pr[\mathsf{Hyb}_{i,3}(\mathcal{A})=1]| \ge \varepsilon$ for some non-negligible ε. We use $\mathcal{A}$ to construct an efficient adversary $\mathcal{B}$ that breaks Type-I indistinguishability of Construction 4.8:

1. Algorithm $\mathcal{B}$ runs algorithm $\mathcal{A}$, which starts by outputting the input length 1^ℓ and the circuit size 1^s. Algorithm $\mathcal{B}$ forwards 1^{s+1}, the Type-I indices $(i, i+1)$, and the Type-II index i to its challenger. It receives the base common reference string $\mathsf{crs}_{\mathsf{base}}$ and the Type-II projection trapdoor td_2.

2. Algorithm $\mathcal{B}$ samples

$$\begin{aligned}\mathsf{crs}_{\mathsf{pre}} &\leftarrow \mathsf{SetupPre}(\mathsf{crs}_{\mathsf{base}}, \ell+1)\\ \mathsf{crs}_{\mathsf{lin}} &\leftarrow \mathsf{SetupLin}(\mathsf{crs}_{\mathsf{base}}, S_{\mathsf{lin}})\\ \mathsf{crs}_{\mathsf{quad}} &\leftarrow \mathsf{SetupQuad}(\mathsf{crs}_{\mathsf{base}}, S_{\mathsf{quad}}),\end{aligned}$$

   It give $\mathsf{crs} = \left(1^s, \mathsf{crs}_{\mathsf{base}}, \mathsf{crs}_{\mathsf{pre}}, \mathsf{crs}_{\mathsf{lin}}, \mathsf{crs}_{\mathsf{quad}}\right)$ to $\mathcal{A}$.

3. Algorithm $\mathcal{A}$ outputs an input commitment σ_{in}, an arithmetic circuit $C\colon \mathcal{R}^\ell \to \mathcal{R}^m$, vectors $\mathbf{y}, \mathbf{y}' \in \mathcal{R}^m$, and openings $\pi = (\sigma_1, \sigma_2, \pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}})$ and $\pi' = (\sigma_1', \sigma_2', \pi_{\mathsf{pre}}', \pi_{\mathsf{lin}}', \pi_{\mathsf{quad}}', \pi_{\mathsf{out}}')$.

4. Algorithm $\mathcal{B}$ outputs 1 if $\mathbf{y} \ne \mathbf{y}'$, $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}, \pi) = 1$, $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}', \pi') = 1$, and $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) = \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2')$. Otherwise, it outputs 0.

If the challenger sampled $\mathsf{crs}_{\mathsf{base}} \leftarrow \mathsf{SetupSF}(1^\lambda, 1^{s+1}, i, i)$, the algorithm $\mathcal{B}$ perfectly simulates an execution of $\mathsf{Hyb}_{i,2}$ for $\mathcal{A}$ and outputs 1 with probability $\Pr[\mathsf{Hyb}_{i,2}(\mathcal{A})=1]$. If the challenger sampled $\mathsf{crs}_{\mathsf{base}} \leftarrow \mathsf{SetupSF}(1^\lambda, 1^{s+1}, i+1, i)$, then algorithm $\mathcal{B}$ perfectly simulates an execution of $\mathsf{Hyb}_{i,3}$ for $\mathcal{A}$ and outputs 1 with probability $\Pr[\mathsf{Hyb}_{i,3}(\mathcal{A})=1]$. Correspondingly, algorithm $\mathcal{B}$ breaks Type-I indistinguishability with advantage ε. ◻

Lemma 5.10. *Suppose $\mathsf{FC}_{\mathsf{quad}}$ satisfies quadratic chain binding (Definition 4.35). Then there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $i \in \{\ell+1,\ldots,s\}$, $|\Pr[\mathsf{Hyb}_{i,3}(\mathcal{A})=1] - \Pr[\mathsf{Hyb}_{i,4}(\mathcal{A})=1]| = \mathsf{negl}(\lambda)$.*

Proof. Suppose there exists an index $i \in \{\ell+1,\ldots,s\}$ where $|\Pr[\mathsf{Hyb}_{i,3}(\mathcal{A})=1] - \Pr[\mathsf{Hyb}_{i,4}(\mathcal{A})=1]| \ge \varepsilon$ for some non-negligible ε. By construction, the common reference string in the two experiments is identically distributed. Thus, it must be the case that with probability at least ε, algorithm $\mathcal{A}$ will output $\sigma_{\mathsf{in}}, C, \mathbf{y}, \mathbf{y}', \pi = (\sigma_1, \sigma_2, \pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}})$ and $\pi' = (\sigma_1', \sigma_2', \pi_{\mathsf{pre}}', \pi_{\mathsf{lin}}', \pi_{\mathsf{quad}}', \pi_{\mathsf{out}}')$ such that

- $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}, \pi) = 1 = \mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}', \pi')$.
- $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) \ne \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$.
- $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) = \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2')$.

In all other cases, the outputs of $\mathsf{Hyb}_{i,3}$ and $\mathsf{Hyb}_{i,4}$ are identical. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ for the quadratic chain binding game:

1. Algorithm $\mathcal{B}$ runs algorithm $\mathcal{A}$, which starts by outputting the input length 1^ℓ and the circuit size 1^s. Algorithm $\mathcal{B}$ sends 1^{s+1}, the locality set S_{quad}, and indices $(i, i+1)$ to the quadratic chain binding adversary. It receives $(\mathsf{crs}_{\mathsf{base}}, \mathsf{crs}_{\mathsf{quad}})$.

2. Algorithm $\mathcal{B}$ samples $\mathsf{crs}_{\mathsf{lin}} \leftarrow \mathsf{SetupLin}(\mathsf{crs}_{\mathsf{base}})$ and $\mathsf{crs}_{\mathsf{pre}} \leftarrow \mathsf{SetupPre}(\mathsf{crs}_{\mathsf{base}}, \ell+1)$. It gives the common reference string $\mathsf{crs} = \left(1^s, \mathsf{crs}_{\mathsf{base}}, \mathsf{crs}_{\mathsf{pre}}, \mathsf{crs}_{\mathsf{lin}}, \mathsf{crs}_{\mathsf{quad}}\right)$ to $\mathcal{A}$.

3. Algorithm $\mathcal{A}$ outputs an input commitment σ_{in}, an arithmetic circuit $C \colon \mathcal{R}^\ell \to \mathcal{R}^m$, vectors $\mathbf{y}, \mathbf{y}' \in \mathcal{R}^m$, and openings $\pi = (\sigma_1, \sigma_2, \pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}})$ and $\pi' = (\sigma_1', \sigma_2', \pi_{\mathsf{pre}}', \pi_{\mathsf{lin}}', \pi_{\mathsf{quad}}', \pi_{\mathsf{out}}')$.

4. Algorithm $\mathcal{B}$ outputs the matrix $\mathbf{M}_C$, the Type-II commitments σ_2, σ_2', the Type-I commitments σ_1, σ_1', and the openings $\pi_{\mathsf{quad}}, \pi_{\mathsf{quad}}'$.

First, we note that $\mathcal{B}$ is a valid adversary for the chain binding security game. From Definition 5.1, the "next-wire" matrix $\mathbf{M}_C$ is $(j, j+1)$-local for all $j \geq \ell + 1$. In particular, this means that $\mathbf{M}_C$ is S_{quad}-local and moreover, that $(i, i+1) \in S_{\mathsf{quad}}$. Then, the chain-binding challenger samples $(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^\lambda, 1^{s+1}, i+1, i)$ Thus, algorithm $\mathcal{B}$ perfectly simulates an execution of $\mathsf{Hyb}_{i,3}$ and $\mathsf{Hyb}_{i,4}$ for $\mathcal{A}$, so with probability at least ε, the quantities output by $\mathcal{A}$ satisfy the properties enumerated above. Then, the following hold:

- If $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}, \pi) = 1 = \mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}', \pi')$, then $\mathsf{VerifyQuad}(\mathsf{crs}_{\mathsf{quad}}, \sigma_2, \mathbf{M}_C, \sigma_1, \pi_{\mathsf{quad}}) = 1$ and $\mathsf{VerifyQuad}(\mathsf{crs}_{\mathsf{quad}}, \sigma_2', \mathbf{M}_C, \sigma_1', \pi_{\mathsf{quad}}') = 1$.

- $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) \neq \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$.

- $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2) = \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_2')$.

These conditions precisely coincide with the requirements of the quadratic chain binding game, so we conclude that algorithm $\mathcal{B}$ succeeds with advantage ε. ◻

Lemma 5.11. *For all $i \in \{\ell+1, \ldots, s\}$, $\Pr[\mathsf{Hyb}_{i,4}(\mathcal{A}) = 1] \leq \Pr[\mathsf{Hyb}_{i,5}(\mathcal{A}) = 1]$.*

Proof. The verification conditions in $\mathsf{Hyb}_{i,4}$ is a strict superset of those in $\mathsf{Hyb}_{i,5}$. Correspondingly, if $\mathsf{Hyb}_{i,4}(\mathcal{A})$ outputs 1, then the same is true for $\mathsf{Hyb}_{i,5}(\mathcal{A})$ and the claim holds. ◻

Lemma 5.12. *Suppose $\mathsf{FC}_{\mathsf{base}}$ satisfies Type-II indistinguishability (Definition 4.6). Then there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $i \in \{\ell+1, \ldots, s\}$, $|\Pr[\mathsf{Hyb}_{i,5}(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_{i+1,0}(\mathcal{A}) = 1]| = \mathsf{negl}(\lambda)$.*

Proof. Suppose there exists an index $i \in \{\ell+1, \ldots, s\}$ where $|\Pr[\mathsf{Hyb}_{i,5}(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_{i+1,0}(\mathcal{A}) = 1]| \geq \varepsilon$ for some non-negligible ε. We use $\mathcal{A}$ to construct an efficient adversary $\mathcal{B}$ that breaks Type-II indistinguishability of Construction 4.8:

1. Algorithm $\mathcal{B}$ runs algorithm $\mathcal{A}$, which starts by outputting the input length 1^ℓ and the circuit size 1^s. Algorithm $\mathcal{B}$ forwards 1^{s+1}, the Type-I index $i+1$, and two Type-II indices $(i, i+1)$ to its challenger. It receives the base common reference string $\mathsf{crs}_{\mathsf{base}}$ and the Type-I projection trapdoor td_1.

2. Algorithm $\mathcal{B}$ samples

$$\begin{aligned} \mathsf{crs}_{\mathsf{pre}} &\leftarrow \mathsf{SetupPre}(\mathsf{crs}_{\mathsf{base}}, \ell+1) \\ \mathsf{crs}_{\mathsf{lin}} &\leftarrow \mathsf{SetupLin}(\mathsf{crs}_{\mathsf{base}}, S_{\mathsf{lin}}) \\ \mathsf{crs}_{\mathsf{quad}} &\leftarrow \mathsf{SetupQuad}(\mathsf{crs}_{\mathsf{base}}, S_{\mathsf{quad}}), \end{aligned}$$

   It give $\mathsf{crs} = \left(1^s, \mathsf{crs}_{\mathsf{base}}, \mathsf{crs}_{\mathsf{pre}}, \mathsf{crs}_{\mathsf{lin}}, \mathsf{crs}_{\mathsf{quad}}\right)$ to $\mathcal{A}$.

3. Algorithm $\mathcal{A}$ outputs an input commitment σ_{in}, an arithmetic circuit $C \colon \mathcal{R}^\ell \to \mathcal{R}^m$, vectors $\mathbf{y}, \mathbf{y}' \in \mathcal{R}^m$, and openings $\pi = (\sigma_1, \sigma_2, \pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}})$ and $\pi' = (\sigma_1', \sigma_2', \pi_{\mathsf{pre}}', \pi_{\mathsf{lin}}', \pi_{\mathsf{quad}}', \pi_{\mathsf{out}}')$.

4. Algorithm $\mathcal{B}$ outputs 1 if $\mathbf{y} \neq \mathbf{y}'$, $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}, \pi) = 1$, $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}', \pi') = 1$, and $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$. Otherwise, it outputs 0.

If the challenger sampled $\mathsf{crs}_{\mathsf{base}} \leftarrow \mathsf{SetupSF}(1^\lambda, 1^{s+1}, i+1, i)$, the algorithm $\mathcal{B}$ perfectly simulates an execution of $\mathsf{Hyb}_{i,5}$ for $\mathcal{A}$ and outputs 1 with probability $\Pr[\mathsf{Hyb}_{i,5}(\mathcal{A}) = 1]$. If the challenger sampled $\mathsf{crs}_{\mathsf{base}} \leftarrow \mathsf{SetupSF}(1^\lambda, 1^{s+1}, i+1, i+1)$, then algorithm $\mathcal{B}$ perfectly simulates an execution of $\mathsf{Hyb}_{i+1,0}$ for $\mathcal{A}$ and outputs 1 with probability $\Pr[\mathsf{Hyb}_{i+1,0}(\mathcal{A}) = 1]$. Correspondingly, algorithm $\mathcal{B}$ breaks Type-II indistinguishability with advantage ε. ◻

Lemma 5.13. *Suppose* $\mathsf{FC}_{\mathsf{lin}}$ *satisfies satisfies linear chain binding (Definition 4.20). Then there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that* $|\Pr[\mathsf{Hyb}_{s+1,0}(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_{\mathsf{final}}(\mathcal{A}) = 1]| = \mathsf{negl}(\lambda)$.

Proof. This proof is similar to the proof of Lemma 5.7. Suppose $|\Pr[\mathsf{Hyb}_{s+1,0}(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_{\mathsf{final}}(\mathcal{A}) = 1]| \geq \varepsilon$ for some non-negligible ε. By construction, the common reference string in the two experiments is identically distributed. Thus, it must be the case that with probability at least ε, algorithm $\mathcal{A}$ will output σ_{in}, C, $\mathbf{y}, \mathbf{y}'$, $\pi = (\sigma_1, \sigma_2, \pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}})$ and $\pi' = (\sigma_1', \sigma_2', \pi_{\mathsf{pre}}', \pi_{\mathsf{lin}}', \pi_{\mathsf{quad}}', \pi_{\mathsf{out}}')$ such that

- $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}, \pi) = 1 = \mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}', \pi')$.
- $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1)$.
- $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_{\mathsf{out}}) \neq \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_{\mathsf{out}}')$, where

$$\sigma_{\mathsf{out}} \leftarrow \mathsf{Commit}^{(2)}\left(\mathsf{crs}_{\mathsf{base}}, \begin{bmatrix} \mathbf{0} \\ \mathbf{y} \end{bmatrix}\right) \quad \text{and} \quad \sigma_{\mathsf{out}}' \leftarrow \mathsf{Commit}^{(2)}\left(\mathsf{crs}_{\mathsf{base}}, \begin{bmatrix} \mathbf{0} \\ \mathbf{y}' \end{bmatrix}\right).$$

In all other cases, the outputs of $\mathsf{Hyb}_{s+1,0}$ and $\mathsf{Hyb}_{\mathsf{final}}$ are identical. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ for the linear chain binding game:

1. Algorithm $\mathcal{B}$ runs algorithm $\mathcal{A}$, which starts by outputting the input length 1^ℓ and the circuit size 1^s. Algorithm $\mathcal{B}$ forwards 1^{s+1}, the locality set S_{lin}, and indices $(s+1, s+1)$ to the linear chain binding challenger. It receives $(\mathsf{crs}_{\mathsf{base}}, \mathsf{crs}_{\mathsf{lin}})$.
2. Algorithm $\mathcal{B}$ samples $\mathsf{crs}_{\mathsf{quad}} \leftarrow \mathsf{SetupQuad}(\mathsf{crs}_{\mathsf{base}})$ and $\mathsf{crs}_{\mathsf{pre}} \leftarrow \mathsf{SetupPre}(\mathsf{crs}_{\mathsf{base}}, \ell+1)$. It gives the common reference string $\mathsf{crs} = \left(1^s, \mathsf{crs}_{\mathsf{base}}, \mathsf{crs}_{\mathsf{pre}}, \mathsf{crs}_{\mathsf{lin}}, \mathsf{crs}_{\mathsf{quad}}\right)$ to $\mathcal{A}$.
3. Algorithm $\mathcal{A}$ outputs an input commitment σ_{in}, an arithmetic circuit $C \colon \mathcal{R}^\ell \to \mathcal{R}^m$, vectors $\mathbf{y}, \mathbf{y}' \in \mathcal{R}^m$, and openings $\pi = (\sigma_1, \sigma_2, \pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}})$ and $\pi' = (\sigma_1', \sigma_2', \pi_{\mathsf{pre}}', \pi_{\mathsf{lin}}', \pi_{\mathsf{quad}}', \pi_{\mathsf{out}}')$.
4. Algorithm $\mathcal{B}$ outputs the matrix $\mathbf{P}_{\mathsf{out}}$, the Type-I commitments σ_1, σ_1', the Type-II commitments $\sigma_{\mathsf{out}}, \sigma_{\mathsf{out}}'$ (computed as in Section 5), and the openings $\pi_{\mathsf{out}}, \pi_{\mathsf{out}}'$.

First, we note that $\mathcal{B}$ is a valid adversary for the chain binding security game. Since $\mathbf{P}_{\mathsf{out}}$ is a diagonal matrix, it is S_{lin}-local, and moreover, $(s+1, s+1) \in S_{\mathsf{lin}}$. Thus, the challenger samples $(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^\lambda, 1^{s+1}, s+1, s+1)$, and algorithm $\mathcal{B}$ perfectly simulates an execution of $\mathsf{Hyb}_{s+1,0}$ and $\mathsf{Hyb}_{\mathsf{final}}$ for $\mathcal{A}$. Thus, with probability at least ε, the quantities output by $\mathcal{A}$ satisfy the properties enumerated above. Then, the following hold:

- If $\mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}, \pi) = 1 = \mathsf{Verify}(\mathsf{crs}, \sigma_{\mathsf{in}}, C, \mathbf{y}', \pi')$, then we have that $\mathsf{VerifyLin}(\mathsf{crs}_{\mathsf{lin}}, \sigma_1, \mathbf{P}_{\mathsf{out}}, \sigma_{\mathsf{out}}, \pi_{\mathsf{out}}) = 1$ and $\mathsf{VerifyLin}(\mathsf{crs}_{\mathsf{lin}}, \sigma_1', \mathbf{I}_s, \sigma_{\mathsf{out}}', \pi_{\mathsf{out}}') = 1$.
- $\mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1) = \mathsf{Project}^{(1)}(\mathsf{td}_1, \sigma_1')$.
- $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_{\mathsf{out}}) \neq \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_{\mathsf{out}}')$.

These conditions precisely coincide with the requirements of the linear chain binding game, so we conclude that algorithm $\mathcal{B}$ succeeds with advantage ε. ◻

Lemma 5.14. *Suppose* $\mathsf{FC}_{\mathsf{base}}$ *satisfies Type-II collision resistance (Definition 4.7). Then there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that* $\Pr[\mathsf{Hyb}_{\mathsf{final}}(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$.

Proof. Suppose $\Pr[\mathsf{Hyb}_{\mathsf{final}}(\mathcal{A}) = 1] \geq \varepsilon$ for some non-negligible ε. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ that breaks Type-II collision resistance:

1. Algorithm $\mathcal{B}$ runs algorithm $\mathcal{A}$, which starts by outputting the input length 1^ℓ and the circuit size 1^s. Algorithm $\mathcal{B}$ forwards 1^{s+1} and the Type-I index $s+1$ to the challenger. It receives $\mathsf{crs}_{\mathsf{base}}$.

2. Algorithm $\mathcal{B}$ samples

$$\begin{aligned} \mathsf{crs}_{\mathsf{pre}} &\leftarrow \mathsf{SetupPre}(\mathsf{crs}_{\mathsf{base}}, \ell + 1) \\ \mathsf{crs}_{\mathsf{lin}} &\leftarrow \mathsf{SetupLin}(\mathsf{crs}_{\mathsf{base}}, \mathcal{S}_{\mathsf{lin}}) \\ \mathsf{crs}_{\mathsf{quad}} &\leftarrow \mathsf{SetupQuad}(\mathsf{crs}_{\mathsf{base}}, \mathcal{S}_{\mathsf{quad}}), \end{aligned}$$

   It give $\mathsf{crs} = \left(1^s, \mathsf{crs}_{\mathsf{base}}, \mathsf{crs}_{\mathsf{pre}}, \mathsf{crs}_{\mathsf{lin}}, \mathsf{crs}_{\mathsf{quad}}\right)$ to $\mathcal{A}$.

3. Algorithm $\mathcal{A}$ outputs an input commitment σ_{in}, an arithmetic circuit $C \colon \mathcal{R}^\ell \to \mathcal{R}^m$, vectors $\mathbf{y}, \mathbf{y}' \in \mathcal{R}^m$, and openings $\pi = (\sigma_1, \sigma_2, \pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}})$ and $\pi' = (\sigma_1', \sigma_2', \pi_{\mathsf{pre}}', \pi_{\mathsf{lin}}', \pi_{\mathsf{quad}}', \pi_{\mathsf{out}}')$.

4. Algorithm $\mathcal{B}$ outputs the vectors $\mathbf{y}, \mathbf{y}'$.

The challenger samples $(\mathsf{crs}_{\mathsf{base}}, \mathsf{td}_1, \mathsf{td}_2) \leftarrow \mathsf{SetupSF}(1^\lambda, 1^{s+1}, s+1, s+1)$, so algorithm $\mathcal{B}$ perfectly simulates an execution of $\mathsf{Hyb}_{\mathsf{final}}$ for $\mathcal{A}$. Thus, with probability at least ε, it holds that $\mathbf{y} \neq \mathbf{y}'$ and $\mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_{\mathsf{out}}) = \mathsf{Project}^{(2)}(\mathsf{td}_2, \sigma_{\mathsf{out}}')$, where $\sigma_{\mathsf{out}} = \mathsf{Commit}^{(2)}\left(\mathsf{crs}_{\mathsf{base}}, \left[\begin{smallmatrix} \mathbf{0} \\ \mathbf{y} \end{smallmatrix}\right]\right)$ and $\sigma_{\mathsf{out}}' = \mathsf{Commit}^{(2)}\left(\mathsf{crs}_{\mathsf{base}}, \left[\begin{smallmatrix} \mathbf{0} \\ \mathbf{y}' \end{smallmatrix}\right]\right)$. These conditions precisely coincide with the requirements of the Type-II collision resistance game, so algorithm $\mathcal{B}$ succeeds with advantage ε. ◻

Since $s = \mathsf{poly}(\lambda)$, we conclude via Lemmas 5.5 to 5.13 that

$$\Pr[\mathsf{Hyb}_{\mathsf{final}}(\mathcal{A}) = 1] \geq \Pr[\mathsf{Hyb}_{\mathsf{real}}(\mathcal{A}) = 1] - \mathsf{negl}(\lambda).$$

By Lemma 5.14, we have that $\Pr[\mathsf{Hyb}_{\mathsf{final}}(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$, so we conclude that $\Pr[\mathsf{Hyb}_0(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$, and binding holds. ◻

Succinct functional commitments from bilateral k-Lin. Combining the construction from Construction 5.2 with our projective commitments (and associated proof systems) from Section 4, we obtain a functional commitment for all arithmetic circuits from the bilateral k-Lin assumption. Notably, both the commitments and the openings in our construction consist of a *constant* number of group elements. We summarize our instantiation in the following corollary.

Corollary 5.15 (Functional Commitments from k-Lin)**.** *Let $k > 1$ be a constant and* $\mathsf{GroupGen}$ *be a prime-order pairing group generator. If the bilateral k-Lin assumption holds with respect to* $\mathsf{GroupGen}$*, then there exists a succinct functional commitment that supports openings to arbitrary arithmetic circuits of size s (over the ring $\mathbb{Z}_p$ associated with* $\mathsf{GroupGen}$*) with the following properties:*

- **Commitment size:** *A commitment to an input $\mathbf{x} \in \mathbb{Z}_p^\ell$ consists of $2k$ elements in the group $\mathbb{G}_2$.*
- **Opening size:** *An opening to an arithmetic circuit $C \colon \mathbb{Z}_p^\ell \to \mathbb{Z}_p^m$ consists of $2k$ elements in $\mathbb{G}_1$ and $4k^2 + 14k + 6$ elements in $\mathbb{G}_2$.*
- **CRS size:** *The CRS is a structured reference string containing $O(k^3 s^5)$ group elements.*

For the particular case of $k = 2$, a commitment consists of 4 group elements and an opening consists of 54 group elements (specifically, 4 $\mathbb{G}_1$ and 50 $\mathbb{G}_2$ elements).

Proof. We instantiate the base scheme $\mathsf{FC}_{\mathsf{base}}$ with Construction 4.8 and the proof systems $\mathsf{FC}_{\mathsf{pre}}, \mathsf{FC}_{\mathsf{lin}}, \mathsf{FC}_{\mathsf{quad}}$ with Constructions 4.8, 4.14, and 4.23. Then, we have the following:

- **Commitment size:** A commitment to an input $\mathbf{x} \in \mathbb{Z}_p^\ell$ is a Type-I commitment (output by $\mathsf{Commit}^{(1)}$), which is a vector in $\mathbb{G}_2^{2k}$.

- **Opening size:** An opening consists of a tuple $(\sigma_1, \sigma_2, \pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}})$. We consider each component:

  - σ_1 is a Type-I commitment so $\sigma_1 \in \mathbb{G}_2^{2k}$.
  - σ_2 is a Type-II commitment so $\sigma_2 \in \mathbb{G}_1^{2k} \times \mathbb{G}_2^{2k}$.
  - π_{pre} is an opening for $\mathsf{FC}_{\mathsf{pre}}$ so $\pi_{\mathsf{pre}} \in \mathbb{G}_2^{k+1}$.
  - π_{lin} is an opening for $\mathsf{FC}_{\mathsf{lin}}$, so $\pi_{\mathsf{lin}} \in \mathbb{G}_2^{4k+2}$.
  - π_{quad} is an opening for $\mathsf{FC}_{\mathsf{quad}}$, so $\pi_{\mathsf{quad}} \in \mathbb{G}_2^{4k^2+k+1}$.
  - π_{out} is an opening for $\mathsf{FC}_{\mathsf{lin}}$ so $\pi_{\mathsf{out}} \in \mathbb{G}_2^{4k+2}$.

  Taken altogether, the opening consists of $2k$ elements in $\mathbb{G}_1$ and $4k^2 + 14k + 6$ elements in $\mathbb{G}_2$.

- **CRS size:** The CRS in Construction 5.2 is a tuple $\mathsf{crs} = (1^s, \mathsf{crs}_{\mathsf{base}}, \mathsf{crs}_{\mathsf{pre}}, \mathsf{crs}_{\mathsf{lin}}, \mathsf{crs}_{\mathsf{quad}})$. The base CRS consists of $O(s^2k^2)$ group elements. Next, $\mathsf{crs}_{\mathsf{pre}}$ contains an additional $O(k^2s)$ group elements, $\mathsf{crs}_{\mathsf{lin}}$ contains an additional $O(k^2s^3)$ and $\mathsf{crs}_{\mathsf{quad}}$ contains an additional $O(k^3s^5)$ group elements. Taken together, the CRS size contains $O(k^3s^5)$ group elements. □

Extensions and applications. We now describe several simple extensions and corollaries of our new functional commitment scheme.

Remark 5.16 (Fast Verification)**.** The running time of the verification algorithm for the functional commitment scheme in Corollary 5.15 scales with $O(s^3)$, where s is the size of the arithmetic circuit. This is the time needed to implement the verification algorithm for the chainable proof system for quadratic functions (Construction 4.38). However, when the circuit C is known in advance, we can preprocess the circuit C so that verification requires only $O(m)$ bilinear map operations, where m is the output size. Specifically, we can precompute the following quantities to reduce the online cost of checking $\pi_{\mathsf{pre}}, \pi_{\mathsf{lin}}, \pi_{\mathsf{quad}}, \pi_{\mathsf{out}}$ in Construction 5.2:

- Checking π_{pre}: The $\mathsf{VerifyPre}$ algorithm is already fast (only requires $O(k)$ number of bilinear map operations), so no preprocessing is needed for checking π_{pre}.

- Checking π_{lin}: We precompute $(\mathrm{vec}(\mathbf{I}_s)^\intercal \otimes \mathbf{I}_k)[(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_1]_2 \in \mathbb{G}_1^{k \times 2k}$ and $(\mathrm{vec}(\mathbf{I}_s)^\intercal \otimes \mathbf{I}_k)[(\mathbf{I}_{\ell^2} \otimes \mathbf{A})\mathbf{W}_2]_2 \in \mathbb{G}_1^{k \times 2k}$. Then, evaluating $\mathsf{VerifyLin}$ in Construction 4.23 only requires $O(k^2)$ group operations. The precomputed key in this case only depends on the size of the circuit C and *not* on the actual description of C.

- Checking π_{quad}: We precompute the circuit-dependent verification key $(\mathrm{vec}(\mathbf{M}_C)^\intercal \otimes \mathbf{I}_k)[(\mathbf{I}_{\ell^3} \otimes \mathbf{A})\mathbf{W}]_1 \in \mathbb{G}_1^{k \times 4k^2}$. Then, evaluating $\mathsf{VerifyQuad}$ in Construction 4.38 only requires $O(k^3)$ group operations.

- Checking π_{out}: Similar to the case for π_{lin}, we precompute $(\mathrm{vec}(\mathbf{P}_{\mathsf{out}})^\intercal \otimes \mathbf{I}_k)[(\mathbf{I}_{\ell^2} \otimes \mathbf{A}]\mathbf{W}_1]_2 \in \mathbb{G}_1^{k \times 2k}$ and $(\mathrm{vec}(\mathbf{P}_{\mathsf{out}})^\intercal \otimes \mathbf{I}_k)[(\mathbf{I}_{\ell^2} \otimes \mathbf{A}]\mathbf{W}_2]_2 \in \mathbb{G}_1^{k \times 2k}$. With the precomputed key, evaluating $\mathsf{VerifyLin}$ only takes $O(k^2)$ group operations.

Since $k = O(1)$, these operations only require a constant number of bilinear group operations. The online cost of the verification is then just the cost of computing the commitment σ_{out} to the output $\mathbf{y}$, which requires $O(m)$ group operations. Note that if the target value $\mathbf{y}$ is also known in advance, then we can also precompute σ_{out}. In this case, the online verification would only require a constant number of bilinear map operations.

Remark 5.17 (Application to Homomorphic Signatures)**.** Previously, the authors of [CFT22] described a generic approach for constructing a homomorphic signature from any additively-homomorphic functional commitment scheme. The class of functions supported by the homomorphic signature scheme coincides with the class of functions associated with the functional commitment scheme. Our functional commitment scheme (Corollary 5.15) satisfies the required additive homomorphism property. Namely, the commitments in our scheme consist of a single Type-I commitment for the base projective commitment scheme (Construction 4.8). In Construction 4.8, a commitment to $\mathbf{x} \in \mathbb{Z}_p^\ell$ is $[\hat{\mathbf{T}}\mathbf{x}]_2$. The base commitment scheme is clearly additively homomorphic. Thus, we can apply the [CFT22] approach to obtain a homomorphic signature for all bounded-size arithmetic circuits. The resulting homomorphic signature scheme inherits the efficiency properties of the underlying functional commitment in this case. In our setting, this gives a homomorphic signature for general circuits where the size of the signature is always a *constant* number of group elements. Previous pairing-based approaches for homomorphic signatures either required knowledge assumptions (through the use of general-purpose SNARKs), had signatures whose size grew with the depth of the computation [BCFL23], or had signatures who size consisted of a super-constant number of group elements [KLVW23] (specifically, the number of group elements is proportional to the size of a circuit implementing a cryptographic hash function, which has size $\mathsf{poly}(\lambda)$).

Remark 5.18 (Chainable Commitment for Arbitrary Circuits)**.** In Construction 5.2, the input commitments are Type-I commitments while the output commitments are Type-II. It is easy to construct a chainable commitment where the input and outputs have the same type; namely, where the output commitment is also a Type-I commitment

$$\sigma_{\mathsf{out}} \coloneqq \mathsf{Commit}_{\mathsf{base}}^{(1)}\left(\mathsf{crs}_{\mathsf{base}}, \left[\begin{smallmatrix} 1 \\ C(\mathbf{x}) \\ \mathbf{0} \end{smallmatrix}\right]\right).$$

To support this, we simply include an additional opening for the projection function that maps

$$\begin{bmatrix} 1 \\ \mathbf{z} \\ C(\mathbf{x}) \end{bmatrix} \mapsto \begin{bmatrix} 1 \\ C(\mathbf{x}) \\ \mathbf{0} \end{bmatrix},$$

where $\mathbf{z}$ denotes the input and intermediate wires of $C(\mathbf{x})$.[7] Clearly, this is a linear mapping, and thus can be handled using our techniques; technically, we will use the quadratic system here since we are converting from a Type-II commitment to a Type-I commitment. In this way, we obtain a chainable commitment for arbitrary circuits. In particular this allows a user to take a commitment σ_1 to an input $\mathbf{x}$, apply a circuit C_1 to $\mathbf{x}$ to obtain a commitment σ_2 to the value $C_1(\mathbf{x})$. The user can then *apply* a new circuit C_1 to obtain a commitment σ_3 to the value $C_2(C_1(\mathbf{x}))$, and so on. As shown by the authors of [BCFL23], a chainable commitment can be used to obtain a functional commitment for circuits of a priori unbounded depth, so long as we allow the size of the opening to scale with the depth of the circuit. Our approach directly supports this setting.

Acknowledgments

We thank Daniel Wichs for helpful pointers on RAM delegation and homomorphic signatures and Valerio Cini for helpful discussions about [BCFL23]. David J. Wu is supported in part by NSF CNS-2151131, CNS-2140975, CNS-2318701, a Microsoft Research Faculty Fellowship, and a Google Research Scholar award.

References

[ACL+22] Martin R. Albrecht, Valerio Cini, Russell W. F. Lai, Giulio Malavolta, and Sri AravindaKrishnan Thyagarajan. Lattice-based SNARKs: Publicly verifiable, preprocessing, and recursively composable. In *CRYPTO*, 2022.

[BC12] Nir Bitansky and Alessandro Chiesa. Succinct arguments from multi-prover interactive proofs and their efficiency benefits. In *CRYPTO*, 2012.

[BCFL23] David Balbás, Dario Catalano, Dario Fiore, and Russell W. F. Lai. Chainable functional commitments for unbounded-depth circuits. In *TCC*, 2023.

[7] Strictly speaking, we replace the existing opening π_{out} with an opening for the modified projection function described here.

[BCI+13] Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Omer Paneth. Succinct non-interactive arguments via linear interactive proofs. In *TCC*, 2013.

[BDFG21] Dan Boneh, Justin Drake, Ben Fisch, and Ariel Gabizon. Halo infinite: Proof-carrying data from additive polynomial commitments. In *CRYPTO*, 2021.

[BFS20] Benedikt Bünz, Ben Fisch, and Alan Szepieniec. Transparent snarks from DARK compilers. In *EUROCRYPT*, 2020.

[BGV11] Siavosh Benabbas, Rosario Gennaro, and Yevgeniy Vahlis. Verifiable delegation of computation over large datasets. In *CRYPTO*, 2011.

[CF13] Dario Catalano and Dario Fiore. Vector commitments and their applications. In *PKC*, 2013.

[CFM08] Dario Catalano, Dario Fiore, and Mariagrazia Messina. Zero-knowledge sets with short proofs. In *EUROCRYPT*, 2008.

[CFT22] Dario Catalano, Dario Fiore, and Ida Tucker. Additive-homomorphic functional commitments and applications to homomorphic signatures. In *ASIACRYPT*, 2022.

[CGJ+23] Arka Rai Choudhuri, Sanjam Garg, Abhishek Jain, Zhengzhong Jin, and Jiaheng Zhang. Correlation intractability and SNARGs from sub-exponential DDH. In *CRYPTO*, 2023.

[CHM+20] Alessandro Chiesa, Yuncong Hu, Mary Maller, Pratyush Mishra, Noah Vesely, and Nicholas P. Ward. Marlin: Preprocessing zkSNARKs with universal and updatable SRS. In *EUROCRYPT*, 2020.

[CJJ21] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. SNARGs for P from LWE. In *FOCS*, 2021.

[CLM23] Valerio Cini, Russell W. F. Lai, and Giulio Malavolta. Lattice-based succinct arguments from vanishing polynomials - (extended abstract). In *CRYPTO*, 2023.

[COS20] Alessandro Chiesa, Dev Ojha, and Nicholas Spooner. Fractal: Post-quantum and transparent recursive proofs from holography. In *EUROCRYPT*, 2020.

[dCP23] Leo de Castro and Chris Peikert. Functional commitments for all functions, with transparent setup. In *EUROCRYPT*, 2023.

[DFGK14] George Danezis, Cédric Fournet, Jens Groth, and Markulf Kohlweiss. Square span programs with applications to succinct NIZK arguments. In *ASIACRYPT*, 2014.

[EHK+13] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge L. Villar. An algebraic framework for Diffie-Hellman assumptions. In *CRYPTO*, 2013.

[GGPR13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In *EUROCRYPT*, 2013.

[GKM+18] Jens Groth, Markulf Kohlweiss, Mary Maller, Sarah Meiklejohn, and Ian Miers. Updatable and universal common reference strings with applications to zk-SNARKs. In *CRYPTO*, 2018.

[GR19] Alonso González and Carla Ràfols. Shorter pairing-based arguments under standard assumptions. In *ASIACRYPT*, 2019.

[Gro10] Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In *ASIACRYPT*, 2010.

[Gro16] Jens Groth. On the size of pairing-based non-interactive arguments. In *EUROCRYPT*, 2016.

[GRWZ20] Sergey Gorbunov, Leonid Reyzin, Hoeteck Wee, and Zhenfei Zhang. Pointproofs: Aggregating proofs for multiple vector commitments. In *ACM CCS*, 2020.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *STOC*, 2011.

[GWC19] Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. PLONK: permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. *IACR Cryptol. ePrint Arch.*, 2019.

[GZ21] Alonso González and Alexandros Zacharakis. Fully-succinct publicly verifiable delegation from constant-size assumptions. In *TCC*, 2021.

[IKO07] Yuval Ishai, Eyal Kushilevitz, and Rafail Ostrovsky. Efficient arguments without short PCPs. In *CCC*, 2007.

[KLVW23] Yael Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs. Boosting batch arguments and RAM delegation. In *STOC*, 2023.

[KPY19] Yael Tauman Kalai, Omer Paneth, and Lisa Yang. How to delegate computations publicly. In *STOC*, 2019.

[KVZ21] Yael Tauman Kalai, Vinod Vaikuntanathan, and Rachel Yun Zhang. Somewhere statistical soundness, post-quantum security, and SNARGs. In *TCC*, 2021.

[KW15] Eike Kiltz and Hoeteck Wee. Quasi-adaptive NIZK for linear subspaces revisited. In *EUROCRYPT*, 2015.

[KZG10] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In *ASIACRYPT*, 2010.

[Lee21] Jonathan Lee. Dory: Efficient, transparent arguments for generalised inner products and polynomial commitments. In *TCC*, 2021.

[Lip12] Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In *TCC*, 2012.

[LM19] Russell W. F. Lai and Giulio Malavolta. Subvector commitments with application to succinct arguments. In *CRYPTO*, 2019.

[LP20] Helger Lipmaa and Kateryna Pavlyk. Succinct functional commitment for a large class of arithmetic circuits. In *ASIACRYPT*, 2020.

[LRY16] Benoît Libert, Somindu C. Ramanna, and Moti Yung. Functional commitment schemes: From polynomial commitments to pairing-based accumulators from simple assumptions. In *ICALP*, 2016.

[LW10] Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In *TCC*, 2010.

[LY10] Benoît Libert and Moti Yung. Concise mercurial vector commitments and independent zero-knowledge sets with short proofs. In *TCC*, 2010.

[MBKM19] Mary Maller, Sean Bowe, Markulf Kohlweiss, and Sarah Meiklejohn. Sonic: Zero-knowledge snarks from linear-size universal and updatable structured reference strings. In *ACM CCS*, 2019.

[MRV15] Paz Morillo, Carla Ràfols, and Jorge L. Villar. Matrix computational assumptions in multilinear groups. *IACR Cryptol. ePrint Arch.*, 2015.

[PR17] Omer Paneth and Guy N. Rothblum. On zero-testable homomorphic encryption and publicly verifiable non-interactive arguments. In *TCC*, 2017.

[PST13] Charalampos Papamanthou, Elaine Shi, and Roberto Tamassia. Signatures of correct computation. In *TCC*, 2013.

[PSTY13] Charalampos Papamanthou, Elaine Shi, Roberto Tamassia, and Ke Yi. Streaming authenticated data structures. In *EUROCRYPT*, 2013.

[Wat09] Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *CRYPTO*, 2009.

[WW22] Brent Waters and David J. Wu. Batch arguments for NP and more from standard bilinear group assumptions. In *CRYPTO*, 2022.

[WW23a] Hoeteck Wee and David J. Wu. Lattice-based functional commitments: Fast verification and cryptanalysis. In *ASIACRYPT*, 2023.

[WW23b] Hoeteck Wee and David J. Wu. Succinct vector, polynomial, and functional commitments from lattices. In *EUROCRYPT*, 2023.