Lattice-Based Functional Commitments: Fast Verification and Cryptanalysis

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 $\mathbf{x} \in \{0,1\}^\ell$ and later open up the commitment to a value $y = f(\mathbf{x})$ with respect to some function f. In this work, we focus on schemes that support fast verification. Specifically, after a preprocessing step that depends only on f, the verification time as well as the size of the commitment and opening should be *sublinear* in the input length ℓ, We also consider the dual setting where the user commits to the function f and later, opens up the commitment at an input $\mathbf{x}$.

In this work, we develop two (non-interactive) functional commitments that support fast verification. The first construction supports openings to constant-degree polynomials and has a shorter CRS for a broad range of settings compared to previous constructions. Our second construction is a dual functional commitment for arbitrary bounded-depth Boolean circuits. Both schemes are lattice-based and avoid non-black-box use of cryptographic primitives or lattice sampling algorithms. Security of both constructions rely on the ℓ-succinct short integer solutions (SIS) assumption, a *falsifiable* q-type generalization of the SIS assumption (Preprint 2023).

In addition, we study the challenges of extending lattice-based functional commitments to extractable functional commitments, a notion that is equivalent to succinct non-interactive arguments (when considering openings to quadratic relations). We describe a general methodology that heuristically breaks the extractability of our construction and provides evidence for the implausibility of the knowledge k-R-ISIS assumption of Albrecht et al. (CRYPTO 2022) that was used in several constructions of lattice-based succinct arguments. If we additionally assume hardness of the standard inhomogeneous SIS assumption, we obtain a direct attack on a variant of the extractable linear functional commitment of Albrecht et al.

1 Introduction

In a functional commitment scheme [IKO07, BC12, LRY16], a user can commit to a vector $\mathbf{x}$ and at a later point in time, provide a *short* opening to a value $y = f(\mathbf{x})$ with respect to an (arbitrary) function f. We also consider a *dual* notion where a user commits to the function f and opens to an evaluation at a point $\mathbf{x}$ [BNO21, dCP23]. The efficiency requirement on a functional commitment is both the commitment and the openings are short (i.e., have size that is sublinear or polylogarithmic in the length of $\mathbf{x}$ and the size of the function f). The security requirement is that an adversary cannot open up a commitment σ to two distinct values $y_0 \neq y_1$ with respect to any function f (or in the dual formulation, with respect to an input $\mathbf{x}$). In this work, we focus exclusively on *non-interactive* functional commitments [LRY16, LP20, PPS21, BNO21, ACL+22, BCFL23, dCP23, WW23] in the *standard model* (with a common reference string). Functional commitments generalize notions like vector commitments [LY10, CF13] and polynomial commitments [KZG10, PSTY13] and have found numerous applications to cryptography, most notably, to efficient constructions of succinct non-interactive arguments (SNARGs).

Functional commitments with fast verification. Our focus in this work is on lattice-based functional commitments for general functions. We are specifically interested in constructions that support *fast verification* in the preprocessing model. In this setting, we allow for an initial preprocessing stage that can depend *only* on the function f (which operates on inputs of length ℓ) and outputs a short verification key vk_f. Given the preprocessed verification key vk_f, we then require that the verifier running time (and by extension, the size of the commitment and opening)

to be sublinear in the input length ℓ. We can define a similar property in the dual setting where we preprocess the input $\mathbf{x}$ instead of the function f. Note that having succinct commitments and openings alone does not imply fast verification. For instance, the verification time in [WW23] is linear in the size of the function f even though the size of the commitment and the opening only depend on the depth of f.

In applications where the function of interest is known in advance, preprocessing can significantly reduce verification costs. This is common in settings like delegation and outsourcing computation. Specifically, for the closely-related problem of succinct arguments, working in the "preprocessing" model yields the most succinct constructions [GGPR13, BCI+13, PHGR13, Gro16].

Lattice-based functional commitments. Functional commitments from lattice-based assumptions have received extensive study in the last few years. Several works [PPS21, ACL+22, BCFL23, WW23] gave constructions of functional commitments for broad classes of functions from lattice-based assumptions with a structured CRS. De Castro and Peikert [dCP23] gave a dual functional commitment for all circuits from the standard short integer solutions (SIS) problem in the *uniform* random string model. The authors of [KLVW23] consider a closely-related problem of delegation for RAM programs; their techniques can be adapted to obtain a functional commitments scheme for Boolean circuits from the learning with errors (LWE) assumption in the uniform random string model; see Section 1.3 for more details. Their construction relies on non-black-box use of cryptographic hash functions (and lattice sampling algorithms). Our focus in this work is on constructions that only make black-box use of cryptographic algorithms.

If we restrict our attention to lattice-based functional commitments that only make black-box use of cryptography, the existing constructions with fast verification either support constant-degree polynomials [ACL+22] or bounded-width Boolean circuits [BCFL23]. In the dual setting, we do not have any constructions with fast verification. We refer to Table 1 for a summary of the current state of the art.

1.1 Our Contributions

In this work, we give two constructions of functional commitments that support fast verification. Security of both construction rely on the ℓ-succinct SIS assumption, a falsifiable "q-type" generalization of the SIS assumption introduced by Wee [Wee23]. Notably, this is a weaker assumption than the more structured $\mathsf{BASIS}_{\mathsf{struct}}$ assumption from [WW23]. Our first construction supports constant-degree polynomials. Our second construction is the first dual functional commitment for (bounded-depth) Boolean circuits with fast verification and which only makes black-box use of cryptographic algorithms. We provide a more detailed comparison to previous constructions in Table 1 and summarize the main results here.

Functional commitment for constant-degree polynomials. Our first construction (Construction 3.2) is a functional commitment for constant-degree polynomials where the size of the CRS scales with $\ell^{d+1} \cdot \mathsf{poly}(\lambda, d, \log \ell)$, where d is a bound on the degree of the polynomial, λ is the security parameter, and ℓ is the input length.

For the specific case of opening to quadratic polynomials (an important special case for delegating computations due to the NP-hardness of deciding satisfiability of a system of quadratic functions), our construction has a CRS size of ℓ^3. Previous approaches required a CRS that scale with ℓ^4 [ACL+22] or ℓ^5 [BCFL23]. More generally, for opening to polynomials of constant-degree $d \in \mathbb{N}$, our scheme compares to previous schemes as follows:

- To support degree-d polynomials, the [ACL+22] construction has a CRS of size $\ell^{2d} \cdot \mathsf{poly}(\lambda, d, \log \ell)$. Our construction reduces the exponent from $2d$ to $d+1$.
- The lattice-based construction from [BCFL23] supports Boolean circuits of width w and depth t with a CRS of size w^5 and openings of size $t \log^2 w$. For *sparse* polynomials where the width w of the circuit computing the polynomial is roughly the input length ℓ, then the size of the CRS in the [BCFL23] construction is $O(\ell^5)$, which is shorter than our construction. Conversely, for dense polynomials with roughly ℓ^d monomials, the basic instantiation of [BCFL23] would require a CRS of size ℓ^{5d} (corresponding to a Boolean circuit of width ℓ^d and constant depth d). Alternatively, we could first rebalance the circuit computing the polynomial to have width ℓ^d/t and depth t. This yields a construction with a CRS of size $(\ell^d/t)^5$ and openings of size $\Omega(t)$. Note

Scheme	Functions	$\lvert crs \rvert$	$\lvert \sigma \rvert$	$\lvert \pi \rvert$	FV	BB	Assumption
[KLVW23]*	Boolean circuits	1	1	1	✓	✗	LWE
[BCFL23]	width-w, depth-d circuits	w^5	1	1	✓	✓	twin-k-M-ISIS
[WW23]	linear functions	ℓ^2	1	1	✓	✓	$\mathsf{BASIS}_{\mathsf{struct}}$
[WW23]	depth-d Boolean circuits	ℓ^2	1	1	✗	✓	$\mathsf{BASIS}_{\mathsf{struct}}$
Construction B.1	depth-d Boolean circuits	ℓ^2	1	1	✗	✓	ℓ-succinct SIS
[ACL+22]	degree-d polynomials	ℓ^{2d}	1	1	✓	✓	k-R-ISIS
[BCFL23]	degree-d polynomials†	$(\ell^d/t)^5$	1	t	✓	✓	twin-k-M-ISIS
Construction 3.2	degree-d polynomials	ℓ^{d+1}	1	1	✓	✓	$O(\ell^d)$-succinct SIS
[KLVW23]*	Boolean circuits	1	1	1	✓	✗	LWE
[dCP23]	depth-d Boolean circuits	ℓ	1	ℓ	✗‡	✓	SIS
Construction 3.19	depth-d Boolean circuits	ℓ^2	1	1	✓	✓	ℓ-succinct SIS

*While [KLVW23] construct delegation for RAM programs, their construction can be adapted to obtain a functional commitments for all Boolean circuits. We provide more details in Section 1.3.
†We consider the general case of a dense polynomial with ℓ^d monomials. We instantiate [BCFL23] with a Boolean circuit of width ℓ^d/t and depth $t \leq \ell$ that computes the polynomial. Note that the size of the opening t should be smaller than the input length ℓ (otherwise, the opening can just be the input itself).
‡The [dCP23] construction supports fast verification for certain special cases (e.g., vector commitments and polynomial commitments).

Table 1: Summary of succinct *lattice-based* functional commitments. For each scheme, we report the class of functions it supports, the size of the common reference string crs, the size of the commitment σ, and the size of an opening π in terms of the associated function class and the input length ℓ. We assume functions with a single output. For simplicity, we suppress $\mathsf{poly}(\lambda, d, \log \ell)$ terms throughout the comparison (where d refers to either the degree of the polynomial or the depth of the circuit). The first set of constructions (above the solid purple line) are standard functional commitments where one commits to an input $\mathbf{x}$ and opens to a function f while the second set (below the solid purple line) are dual functional commitments where one commits to a function f and opens to an input $\mathbf{x}$. We say that a scheme supports "fast verification" (**FV**) if after an *input-independent* preprocessing step, the verification time is *sublinear* in ℓ and that it is "black-box" (**BB**) if it only makes black-box use of cryptographic algorithms. Note that $\mathsf{BASIS}_{\mathsf{struct}}$ implies ℓ-succinct SIS [Wee23]. In all constructions, the running time of the commitment algorithm is *linear* in the input length.

that for the setting to be non-trivial, it should be the case that $t \leq \ell$ (otherwise, the opening can just be the input itself). Thus, if we demand sublinear-size openings, the size of the [BCFL23] CRS for supporting general dense polynomials is at least $\ell^{5(d-1)}$, which is worse than both [ACL+22] and our construction.

- Finally, the [WW23] construction supports (bounded-depth) Boolean circuits with a CRS of size ℓ^2. This has a shorter CRS than our construction; however, [WW23] does not support fast verification except in the special case of linear functions.

On the assumption front, the security of Construction 3.2 follows from the L-succinct SIS assumption (with $L = O(\ell^d)$), a falsifiable "q-type" generalization of the SIS assumption introduced by [Wee23]. This is a weaker assumption than the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption used in [WW23] (i.e., is implied by the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption), and is a less structured generalizations of SIS compared to the k-R-ISIS and twin-k-M-ISIS assumptions used in [ACL+22, BCFL23]. We refer to Section 1.2 and Section 3 for an overview of the assumption and construction.

Dual functional commitment for Boolean circuits. Our second construction is a dual functional commitment for arbitrary (bounded-depth) Boolean circuits (Construction 3.19). This is the first dual functional commitment scheme based on falsifiable assumptions that supports *succinct* openings, fast verification, and which does not make non-black-box use of cryptography. Previously, [dCP23] constructed a dual functional commitment from the standard SIS assumption with short commitments but *long* openings and thus, slow verification. Specifically, in their scheme, the size of the opening and the running time of the verification algorithm scale linearly with the input

length ℓ. In our construction, the size of the opening is polylogarithmic in the input length, as is verification (after an initial preprocessing step). On the flip side, the [dCP23] construction has a *transparent* CRS whose size scales linearly with ℓ while our construction has a *structured* CRS whose size scales quadratically with ℓ. The structured CRS is used to "compress" the openings (see Section 1.2 and Construction 3.19). Similar to our functional commitment scheme for constant-degree polynomials, security of our dual functional commitment also relies on the ℓ-succinct SIS assumption.

Extractable commitments and cryptanalysis. The authors of [ACL+22] showed that if the binding property on a functional commitment for *quadratic* functions was replaced by a stronger extractability property, then it can be used to obtain a succinct non-interactive argument for NP. A functional commitment is extractable if for any efficient adversary that outputs a commitment σ and an opening π to the value y with respect to a function f, there exists an extractor that outputs an input x such that $f(x) = y$. Extractable functional commitments for quadratic functions can be used to obtain a succinct non-interactive argument (SNARG) for NP (using the fact that satisfiability of quadratic systems is NP-complete). In this work, we describe a general methodology for cryptanalyzing existing approaches for constructing extractable functional commitments. Notably, we show heuristically that our functional commitment for constant-degree polynomials is unlikely to satisfy extractability. We then describe a similar attack on an adaptation of the [ACL+22] functional commitment for linear functions. Here, we show that assuming (non-uniform) hardness of the *standard* inhomogeneous SIS problem, the variant of [ACL+22] we consider is *not* extractable. Along the way, we also give an oblivious sampling algorithm on a matrix version of the k-R-ISIS knowledge assumption from [ACL+22]. We provide an overview in Section 1.2 and the details in Section 4.

1.2 Technical Overview

In this section, we provide a high-level overview of our approach for constructing functional commitments with fast verification in the preprocessing model as well as the challenges in extending these constructions to satisfy the stronger extractability notion needed to construct preprocessing succinct non-interactive arguments.

Notation. We start with some basic notation. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a target vector $\mathbf{t} \in \mathbb{Z}_q^n$, we write $\mathbf{A}^{-1}(\mathbf{t})$ to denote a random variable $\mathbf{x} \in \mathbb{Z}_q^m$ whose entries are distributed according to a discrete Gaussian distribution conditioned on $\mathbf{Ax} = \mathbf{t}$. We can efficiently sample from $\mathbf{A}^{-1}(\mathbf{t})$ given a trapdoor for the matrix $\mathbf{A}$. We write $\mathbf{I}_n$ to denote the identity matrix of dimension n. We let $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ denote the standard gadget matrix (i.e., $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}^\intercal$, where $\mathbf{g}^\intercal = [1, 2, \ldots, 2^{\lfloor \log q \rfloor}]$) [MP12], and $\mathbf{G}^{-1}(\cdot) \colon \mathbb{Z}_q^n \to \mathbb{Z}_q^m$ denote the usual binary-decomposition operator.

The ℓ-succinct SIS assumption. Our constructions rely on the ℓ-succinct short integer solutions (SIS) assumption [Wee23]. For a matrix $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, the standard SIS problem [Ajt96] is to find a short non-zero solution $\mathbf{x} \in \mathbb{Z}_q^m$ such that $\mathbf{Ax} = \mathbf{0}$. The ℓ-succinct SIS assumption states that SIS is hard with respect to $\mathbf{A}$ even given a trapdoor for $[\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}]$ where $\mathbf{W} \xleftarrow{R} \mathbb{Z}_q^{\ell n \times m}$ is a random *narrow* matrix. Note that if $\mathbf{W} \in \mathbb{Z}_q^{\ell n \times \ell m}$ is *wide*, then hardness of ℓ-succinct SIS can be reduced to the hardness of SIS using lattice trapdoor extension techniques [Wee23].

The ℓ-succinct SIS assumption is a *weaker* assumption that the structured $\mathsf{BASIS}_{\mathsf{struct}}$ assumption used in [WW23] for constructing functional commitments; notably, the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption from [WW23] is an instance of the ℓ-succinct SIS assumption with a *structured* $\mathbf{W}$. While ℓ-succinct SIS is a new and non-standard assumption, it is a falsifiable assumption, and can be viewed as a "q-type" analog of the SIS assumption. We note that it is also implied by the "evasive LWE" assumption [Wee22, Tsa22], which is an assumption that has been used successfully in several other recent works [WWW22, VWW22].

1.2.1 A Functional Commitment Scheme for Quadratic Polynomials

Here, we describe our approach for constructing a functional commitment for constant-degree polynomials on ℓ-dimensional inputs. Specifically, the committer should be able to commit to an input $\mathbf{x} \in \mathbb{Z}_q^\ell$ and then subsequently open up the commitment to $f(\mathbf{x})$ where f is a constant-degree polynomial. For simplicity of exposition, we will focus on the case of quadratic polynomials, and defer the generalization to higher-degree polynomials to Section 3.

The Wee-Wu scheme. We start with a quick recap of the functional commitment for circuits from [WW23] based on the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption (c.f., [WW23, Remark 4.13]), adapted to the ℓ-succinct SIS assumption.[1] As we explain below, although the [WW23] construction shares a similar verification relation as our construction, it does *not* appear to support fast verification. To describe the construction, we first parse the matrix $\mathbf{W} \in \mathbb{Z}_q^{\ell n \times m}$ from the ℓ-succinct SIS assumption as the vertical concatenation of matrices $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(\ell)} \in \mathbb{Z}_q^{n \times m}$. A commitment to a (short) input vector $\mathbf{x} \in \mathbb{Z}_q^{\ell}$ consists of a short matrix $\mathbf{C} \in \mathbb{Z}^{m \times m}$ along with short matrices $\mathbf{V}_i$ satisfying the following relation:

$$\mathbf{W}^{(i)}\mathbf{C} = x_i\mathbf{G} - \mathbf{A}\mathbf{V}_i$$

Then, for all $i, j \in [\ell]$,

$$\begin{aligned}(\mathbf{W}^{(i)}\mathbf{C}) \cdot \mathbf{G}^{-1}(\mathbf{W}^{(j)}\mathbf{C}) &= x_i\mathbf{W}^{(j)}\mathbf{C} - \mathbf{A}\mathbf{V}_i\mathbf{G}^{-1}(\mathbf{W}^{(j)}\mathbf{C}) \\ &= x_i x_j \cdot \mathbf{G} - \mathbf{A} \cdot \underbrace{(x_i\mathbf{V}_j + \mathbf{V}_i\mathbf{G}^{-1}(\mathbf{W}^{(j)}\mathbf{C}))}_{\tilde{\mathbf{V}}_{ij}}\end{aligned}$$

Observe that $\tilde{\mathbf{V}}_{i,j} = x_i\mathbf{V}_j + \mathbf{V}_i\mathbf{G}^{-1}(\mathbf{W}^{(j)}\mathbf{C})$ is small since x_i, $\mathbf{V}_i$, and $\mathbf{V}_j$ are all small. We now view $\tilde{\mathbf{V}}_{ij}$ as the opening for $\mathbf{C}$ to the quadratic relation $x_i x_j$. Furthermore, this extends readily to circuits following [BGG+14, GVW15b]. For the specific case of a general quadratic polynomial $f(\mathbf{x}) = \sum_{i,j\in[\ell]} \gamma_{ij} x_i x_j$, the left-hand side of the verification relation becomes

$$\sum_{i,j\in[\ell]} \gamma_{ij}(\mathbf{W}^{(i)}\mathbf{C}) \cdot \mathbf{G}^{-1}(\mathbf{W}^{(j)}\mathbf{C}).$$

We do not know how to decompose this computation into a slow preprocessing phase that is *independent* of $\mathbf{C}$, followed by a fast computation on $\mathbf{C}$. The analogous expression in the functional commitment scheme of [ACL+22] is given by $\sum_{i,j\in[\ell]} \gamma_{ij} w^{(i)} c \cdot w^{(j)} c$ where $w^{(i)}, w^{(j)}, c$ are *ring* elements. Since ring multiplication is commutative (unlike matrix multiplication), this can be rewritten as $(\sum \gamma_{i,j\in[\ell]} w^{(i)} w^{(j)}) \cdot c^2$. By precomputing the quantity $(\sum \gamma_{i,j\in[\ell]} w^{(i)} w^{(j)})$, which is *independent* of the commitment, the [ACL+22] construction supports fast verification in the preprocessing model.

Our approach. To construct a functional commitment scheme that supports fast verification (with preprocessing), we introduce *additional* structure. For the case of quadratic functions, we rely on the $(\ell+\ell^2)$-succinct SIS assumption; contrast this with the [WW23] construction described above which can rely on the *smaller* ℓ-succinct SIS assumption. We parse the matrix $\mathbf{W} \in \mathbb{Z}_q^{(\ell+\ell^2)n \times m}$ from the $(\ell + \ell^2)$-succinct SIS assumption as

$$\mathbf{W} = \begin{bmatrix}\mathbf{W}_1 \\ \mathbf{W}_2\end{bmatrix} \quad \text{where} \quad \mathbf{W}_1 = \begin{bmatrix}\mathbf{W}_1^{(1)} \\ \vdots \\ \mathbf{W}_1^{(\ell)}\end{bmatrix} \in \mathbb{Z}_q^{n\ell \times m} \quad \text{and} \quad \mathbf{W}_2 = \begin{bmatrix}\mathbf{W}_2^{(1,1)} \\ \vdots \\ \mathbf{W}_1^{(\ell,\ell)}\end{bmatrix} \in \mathbb{Z}_q^{n\ell^2 \times m},$$

where $\mathbf{W}_1^{(i)}, \mathbf{W}_2^{(i,j)} \in \mathbb{Z}_q^{n \times m}$. A commitment to a (short) input vector $\mathbf{x} \in \mathbb{Z}_q^{\ell}$ consists of a *short* matrix $\mathbf{C} \in \mathbb{Z}^{m \times m}$ along with short matrices $\mathbf{V}_i, \mathbf{V}_{ij} \in \mathbb{Z}_q^{m \times m}$ satisfying the following relation:

$$\mathbf{W}_1^{(i)}\mathbf{C} = x_i\mathbf{G} - \mathbf{A}\mathbf{V}_i \tag{1.1}$$

$$\mathbf{W}_2^{(i,j)}\mathbf{C} = x_i\mathbf{W}_1^{(j)} - \mathbf{A}\mathbf{V}_{ij} \tag{1.2}$$

Then, for all $i, j \in [\ell]$,

$$\begin{aligned}\mathbf{W}_2^{(i,j)}\mathbf{C}^2 &= x_i\mathbf{W}_1^{(j)}\mathbf{C} - \mathbf{A}\mathbf{V}_{ij}\mathbf{C} \\ &= x_i x_j \cdot \mathbf{G} - \mathbf{A} \cdot \underbrace{(x_i\mathbf{V}_j + \mathbf{V}_{ij}\mathbf{C})}_{\tilde{\mathbf{V}}_{ij}}.\end{aligned}$$

[1] In Appendix B, we provide the formal description and analysis of [WW23] using the ℓ-succinct SIS assumption.

Observe that $\tilde{\mathbf{V}}_{i,j} = x_i\mathbf{V}_j + \mathbf{V}_{ij}\mathbf{C}$ is small since $\mathbf{x}, \mathbf{V}_j, \mathbf{V}_{ij}$, and $\mathbf{C}$ are all small. We now take $\tilde{\mathbf{V}}_{ij}$ to be the opening for $\mathbf{C}$ to the quadratic relation $x_i x_j$. More generally, an opening for a general quadratic polynomial $f(\mathbf{x}) = \sum_{i,j\in[\ell]} \gamma_{ij} x_i x_j$ to the value $y = f(\mathbf{x})$ is a short matrix $\tilde{\mathbf{V}}$ where

$$\underbrace{\left(\sum_{i,j\in[\ell]} \gamma_{ij}\mathbf{W}_2^{(i,j)}\right)}_{\mathbf{W}_f}\cdot\mathbf{C}^2 = y\cdot\mathbf{G} - \mathbf{A}\cdot\tilde{\mathbf{V}}. \tag{1.3}$$

Our scheme. To complete the description, we publish the following components in the CRS:

$$\begin{bmatrix}\mathbf{T}_{\mathsf{open}}\\ \mathbf{T}_{\mathsf{com}}\end{bmatrix} \leftarrow \left[\begin{array}{cc|c} \mathbf{I}_\ell\otimes\mathbf{A} & & \mathbf{W}_1\\ & \mathbf{I}_{\ell^2}\otimes\mathbf{A} & \mathbf{W}_2\end{array}\right]^{-1}\left(\begin{bmatrix}\mathbf{I}_\ell\otimes\mathbf{G}\\ \mathbf{I}_\ell\otimes\mathbf{W}_1\end{bmatrix}\right), \tag{1.4}$$

where $\mathbf{T}_{\mathsf{open}} \in \mathbb{Z}_q^{(\ell+\ell^2)m\times m\ell}$ and $\mathbf{T}_{\mathsf{com}} \in \mathbb{Z}_q^{m\times m\ell}$. Note that the CRS has size $O(\ell^3)$, improving upon the $O(\ell^4)$-sized CRS in [ACL+22].

To commit to a short $\mathbf{x}\in\mathbb{Z}_q^\ell$, the committer computes $\mathbf{C}\leftarrow\mathbf{T}_{\mathsf{com}}(\mathbf{x}\otimes\mathbf{I}_m)$. By construction this means that

$$\begin{aligned}\mathbf{W}_1\mathbf{C} &= \mathbf{W}_1\mathbf{T}_{\mathsf{com}}(\mathbf{x}\otimes\mathbf{I}_m) = (\mathbf{I}_\ell\otimes\mathbf{G})(\mathbf{x}\otimes\mathbf{I}_m) - (\mathbf{I}_\ell\otimes\mathbf{A})\mathbf{T}_{\mathsf{open}}(\mathbf{x}\otimes\mathbf{I}_m)\\ &= \mathbf{x}\otimes\mathbf{G} - (\mathbf{I}_\ell\otimes\mathbf{A})\mathbf{T}_{\mathsf{open}}(\mathbf{x}\otimes\mathbf{I}_m)\\ \mathbf{W}_2\mathbf{C} &= \mathbf{W}_2\mathbf{T}_{\mathsf{com}}(\mathbf{x}\otimes\mathbf{I}_m) = (\mathbf{I}_\ell\otimes\mathbf{W}_1)(\mathbf{x}\otimes\mathbf{I}_m) - (\mathbf{I}_{\ell^2}\otimes\mathbf{A})\mathbf{T}_{\mathsf{open}}(\mathbf{x}\otimes\mathbf{I}_m)\\ &= \mathbf{x}\otimes\mathbf{W}_1 - (\mathbf{I}_{\ell^2}\otimes\mathbf{A})\mathbf{T}_{\mathsf{open}}(\mathbf{x}\otimes\mathbf{I}_m).\end{aligned}$$

Observe that taking $\mathbf{V}_i$ and $\mathbf{V}_{ij}$ to be the blocks of $\mathbf{T}_{\mathsf{open}}(\mathbf{x}\otimes\mathbf{I}_m)$, we satisfy Eqs. (1.1) and (1.2). To argue binding from the $(\ell^2+\ell)$-succinct SIS assumption, observe that $\mathbf{T}_{\mathsf{open}}$ and $\mathbf{T}_{\mathsf{com}}$ can be sampled using the trapdoor provided by the $(\ell^2+\ell)$-succinct SIS assumption. Suppose now that an adversary outputs two possible openings $\tilde{\mathbf{V}}_0, \tilde{\mathbf{V}}_1$ to values $y_0, y_1 \in \mathbb{Z}_q$ with respect to the *same* quadratic function f. From Eq. (1.3), this means that

$$\mathbf{W}_f\mathbf{C}^2 = y_0\mathbf{G} - \mathbf{A}\tilde{\mathbf{V}}_0 = y_1\mathbf{G} - \mathbf{A}\tilde{\mathbf{V}}_1,$$

or equivalently, that $\mathbf{A}(\tilde{\mathbf{V}}_1 - \tilde{\mathbf{V}}_0) = (y_1 - y_0)\mathbf{G}$. When $y_1 \neq y_0$ and q is prime (so that $y_1 - y_0$ is invertible), this yields a gadget trapdoor [MP12] for $\mathbf{A}$, which the reduction can use to sample a short non-zero SIS solution from $\mathbf{A}^{-1}(\mathbf{0})$. We provide the full details (and extension to higher-degree polynomials) in Section 3.

Fast verification with preprocessing. It is easy to see that the above construction supports fast verification given preprocessing. For instance, consider the verification relation in Eq. (1.3). If the function f is known in advance, we can precompute the matrix $\mathbf{W}_f = \sum_{i,j\in[\ell]}\gamma_{ij}\mathbf{W}_2^{(i,j)}$. If we do so, then the verification relation simply checks $\mathbf{W}_f\mathbf{C}^2 = f(\mathbf{x})\cdot\mathbf{G} - \mathbf{A}\tilde{\mathbf{V}}$, which can be computed in time that depends only *polylogarithmically* on ℓ.

Extending to multiple outputs. Using a similar technique as [WW23], we can also extend our construction above to functions with multiple outputs. To illustrate, suppose we have a commitment $\mathbf{C}$ and a collection of T openings $\tilde{\mathbf{V}}_1,\ldots,\tilde{\mathbf{V}}_T$ to values $y_1,\ldots,y_T$ and with respect to functions $f_1,\ldots,f_T$. Then, for all $i\in[T]$, we have from Eq. (1.3) that $\mathbf{W}_{f_i}\mathbf{C}^2 = y_i\mathbf{G} - \mathbf{A}\tilde{\mathbf{V}}_i$. To support openings to multiple outputs, we publish random vectors $\mathbf{u}_1,\ldots,\mathbf{u}_T \xleftarrow{\mathrm{R}} \mathbb{Z}_q^n$ in the CRS, and define the "multi-output" verification relation to be

$$\sum_{i\in[T]}\mathbf{W}_{f_i}\mathbf{C}^2\mathbf{G}^{-1}(\mathbf{u}_i) \stackrel{?}{=} \sum_{i\in[T]} y_i\mathbf{u}_i - \sum_{i\in[T]}\mathbf{A}\tilde{\mathbf{V}}_i\mathbf{G}^{-1}(\mathbf{u}_i).$$

The new opening is now $\sum_{i\in[T]}\tilde{\mathbf{V}}_i\mathbf{G}^{-1}(\mathbf{u}_i)$ which remains short. Moreover, the multi-output scheme still supports preprocessing. This is because the left-hand-side of the verification relation is still a linear function in $\mathbf{C}^2$ and can

be preprocessed; formally, this is done by "vectorizing" the verification relation (see Remark 3.10). In this case, the verification time with preprocessing is independent of the input length ℓ, but still dependent on the output dimension T (this is anyhow necessary since the verification algorithm needs to read the opened values). In the setting where the target values $y_1, \ldots, y_T$ are also known in advance, we can also precompute the target value $\sum_{i \in [T]} y_i \mathbf{u}_i$. When both the functions and the outputs are preprocessed, the running time of the verification algorithm is polylogarithmic in *both* the input length ℓ and the output dimension T. Finally, security of the multi-output version still reduces to $(\ell^2 + \ell)$-succinct SIS. We provide the full details in Section 3.1. Taken together, we obtain a functional commitment for constant-degree polynomials of degree d where the size of the CRS is $\ell^{d+1} \cdot \mathsf{poly}(\lambda, d, \log \ell, \log T)$ and the proof/opening sizes are $\mathsf{poly}(\lambda, d, \log \ell, \log T)$. Compared to [ACL+22], our construction achieves a shorter CRS (reducing from ℓ^{2d} to ℓ^{d+1}) and relies on a less-structured assumption.

Generalizing to module lattices. Our functional commitment scheme described here generalizes directly to module lattices and ideal lattices. Security in turn relies on the hardness of ℓ-succinct SIS assumption over module lattices (as opposed to integer lattices). We describe the generalization in Appendix A. For a security parameter λ and using module lattices (along with a z-ary gadget matrix where $z \geq q^{1/c}$ for some constant $c \in \mathbb{N}$), we obtain a functional commitment scheme for constant-degree polynomials where the commitment and the opening for an input of length ℓ (and single output) is $\tilde{O}(\lambda \log \ell)$; this relies on $2^{\tilde{\Omega}(\lambda)}$ hardness of $O(\ell^d)$-succinct module SIS. This matches the commitment size and the opening size of the functional commitment from [ACL+22] which relies on ideal lattices. As noted above, compared to [ACL+22], our construction reduces the CRS size from $\ell^{2d} \cdot \mathsf{poly}(\lambda, d, \log \ell)$ to $\ell^{d+1} \cdot \mathsf{poly}(\lambda, d, \log \ell)$.

1.2.2 A Dual Functional Commitment for Boolean Circuits

Next, we turn our attention to the dual setting where the user commits to a function f and opens to an input $\mathbf{x}$. This is the setting studied in [BNO21, dCP23]. While a functional commitment that supports general functions (e.g., [WW23, BCFL23]) can be used to obtain a dual functional commitment for general functions through the use of universal circuits, the generic transformation necessarily both imposes an *a priori* bound on the size (or description length) of the function. Here, we opt for a more direct construction that avoids the need for universal circuits. Our approach is essentially a hybrid of the dual functional commitment for bounded-depth Boolean circuits from [dCP23] (which has short commitments but openings whose size scales with the input length) and the succinct attribute-based encryption (ABE) scheme from [Wee23]. We show how to combine these techniques to obtain a dual functional commitment for bounded-depth Boolean circuits with short commitments *and* openings. As before, our starting point is the ℓ-succinct SIS assumption, where we are given a trapdoor $\mathbf{T}$ satisfying

$$[\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}] \cdot \mathbf{T} = \mathbf{I}_\ell \otimes \mathbf{G}. \tag{1.5}$$

We again parse the trapdoor $\mathbf{T}$ as $\mathbf{T} = \begin{bmatrix} \mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix}$ where $\mathbf{T}_{\mathsf{open}} \in \mathbb{Z}_q^{\ell m \times \ell m}$ and $\mathbf{T}_{\mathsf{com}} \in \mathbb{Z}_q^{m \times \ell m}$. If we use the fact that $(\mathbf{x}^\intercal \otimes \mathbf{I}_n)(\mathbf{I}_\ell \otimes \mathbf{A}) = (1 \otimes \mathbf{A})(\mathbf{x}^\intercal \otimes \mathbf{I}_m) = \mathbf{A}(\mathbf{x}^\intercal \otimes \mathbf{I}_m)$, we obtain

$$\mathbf{x}^\intercal \otimes \mathbf{G} = (\mathbf{x}^\intercal \otimes \mathbf{I}_n)(\mathbf{I}_\ell \otimes \mathbf{G}) = (\mathbf{x}^\intercal \otimes \mathbf{I}_n)[\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}] \cdot \mathbf{T} = [\mathbf{A}(\mathbf{x}^\intercal \otimes \mathbf{I}_m) \mid (\mathbf{x}^\intercal \otimes \mathbf{I}_n)\mathbf{W}] \cdot \begin{bmatrix} \mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix}.$$

Take any matrix $\mathbf{W}_0 \in \mathbb{Z}_q^{n \times m}$. Then, we can write

$$[\mathbf{A} \mid \mathbf{W}_0 + (\mathbf{x}^\intercal \otimes \mathbf{I}_n)\mathbf{W}] \cdot \begin{bmatrix} -(\mathbf{x}^\intercal \otimes \mathbf{I}_m)\mathbf{T}_{\mathsf{open}} \\ -\mathbf{T}_{\mathsf{com}} \end{bmatrix} = -\mathbf{W}_0\mathbf{T}_{\mathsf{com}} - \mathbf{x}^\intercal \otimes \mathbf{G}. \tag{1.6}$$

Let us define $\mathbf{B} := -\mathbf{W}_0\mathbf{T}_{\mathsf{com}} \in \mathbb{Z}_q^{n \times \ell m}$. The CRS will contain the elements $(\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{W}_0, \mathbf{B})$. Now, Eq. (1.6) essentially says we can "recode" the matrix $[\mathbf{A} \mid \mathbf{W}_0 + (\mathbf{x}^\intercal \otimes \mathbf{I}_n)\mathbf{W}]$ to $\mathbf{B} - \mathbf{x}^\intercal \otimes \mathbf{G}$. Following [dCP23], we now define the commitment to a function $f\colon \{0,1\}^\ell \to \{0,1\}$ as the matrix $\mathbf{B}_f$ obtained by homomorphically evaluating f on $\mathbf{B}$ using the lattice-based homomorphic evaluation machinery from [GSW13, BGG+14].[2] To recall, for every matrix

[2] In the ABE scheme from [Wee23], the ciphertext is essentially $\mathbf{s}^\intercal[\mathbf{A} \mid \mathbf{W}_0 + (\mathbf{x} \otimes \mathbf{I}_n)\mathbf{W}]$ + error and the secret key is a short Gaussian pre-image of $[\mathbf{A} \mid \mathbf{B}_f]$ where $\mathbf{B}_f$ is derived from $\mathbf{B}$ via homomorphic evaluation [GSW13, BGG+14] of f on $\mathbf{B}$.

$\mathbf{B} \in \mathbb{Z}_q^{n \times \ell m}$, every function $f\colon \{0,1\}^\ell \to \{0,1\}$, and every input $\mathbf{x} \in \{0,1\}^\ell$, there exist a matrix $\mathbf{B}_f \in \mathbb{Z}_q^{n \times m}$ that depends only on $\mathbf{B}$ and f, and a short matrix $\mathbf{H}_{\mathbf{B},f,\mathbf{x}} \in \mathbb{Z}_q^{\ell m \times m}$ such that

$$(\mathbf{B} - \mathbf{x}^\intercal \otimes \mathbf{G}) \cdot \mathbf{H}_{\mathbf{B},f,\mathbf{x}} = \mathbf{B}_f - f(\mathbf{x}) \cdot \mathbf{G} \in \mathbb{Z}_q^{n \times m}.$$

To open at a point $\mathbf{x} \in \{0,1\}^\ell$ to the value $z = f(\mathbf{x})$, the committer then computes

$$\mathbf{V} = \begin{bmatrix} -(\mathbf{x} \otimes \mathbf{I}_m)\mathbf{T}_{\mathsf{open}} \\ -\mathbf{T}_{\mathsf{com}} \end{bmatrix} \cdot \mathbf{H}_{\mathbf{B},f,\mathbf{x}} \in \mathbb{Z}_q^{2m \times m}.$$

Observe that the size of the opening is essentially independent of the input length ℓ.[3] In [dCP23], the opening is the full matrix $\mathbf{H}_{\mathbf{B},f,\mathbf{x}}$. Here, the trapdoor $\mathbf{T}$ from the ℓ-succinct SIS assumption allows us to "compress" the opening. The verification relation is then

$$\mathbf{B}_f - z\mathbf{G} \stackrel{?}{=} [\mathbf{A} \mid \mathbf{W}_0 + (\mathbf{x} \otimes \mathbf{I}_n)\mathbf{W}]\mathbf{V}. \tag{1.7}$$

From Eq. (1.6), we see that

$$\begin{aligned}
[\mathbf{A} \mid \mathbf{W}_0 + (\mathbf{x}^\intercal \otimes \mathbf{I}_n)\mathbf{W}]\mathbf{V} &= [\mathbf{A} \mid \mathbf{W}_0 + (\mathbf{x}^\intercal \otimes \mathbf{I}_n)\mathbf{W}] \begin{bmatrix} -(\mathbf{x}^\intercal \otimes \mathbf{I}_m)\mathbf{T}_{\mathsf{open}} \\ -\mathbf{T}_{\mathsf{com}} \end{bmatrix} \cdot \mathbf{H}_{\mathbf{B},f,\mathbf{x}} \\
&= (-\mathbf{W}_0\mathbf{T}_{\mathsf{com}} - \mathbf{x}^\intercal \otimes \mathbf{G}) \cdot \mathbf{H}_{\mathbf{B},f,\mathbf{x}} \\
&= (\mathbf{B} - \mathbf{x}^\intercal \otimes \mathbf{G}) \cdot \mathbf{H}_{\mathbf{B},f,\mathbf{x}} \\
&= \mathbf{B}_f - f(\mathbf{x}) \cdot \mathbf{G}.
\end{aligned}$$

This yields a dual functional commitment for all (bounded-depth) Boolean circuits with inputs of length ℓ where the size of the commitment and the opening are both $\mathrm{poly}(\lambda, d^{1/\varepsilon}, \log \ell)$, where d is the bound on the depth of the function and $\varepsilon > 0$ is a constant (see below). The CRS in our construction has size $\ell^2 \cdot \mathrm{poly}(\lambda, d^{1/\varepsilon}, \log \ell)$. This construction also supports preprocessing; namely, if the input $\mathbf{x}$ is known in advance, we can precompute the matrix $[\mathbf{A} \mid \mathbf{W}_0 + (\mathbf{x} \otimes \mathbf{I}_n)\mathbf{W}]$ in Eq. (1.7). Security relies on the ℓ-succinct SIS with a *sub-exponential* noise bound $2^{\tilde{O}(n^\varepsilon)}$, where $\varepsilon > 0$ is a constant and n is the lattice dimension. We refer to Section 3.2 for the full construction and analysis.

1.2.3 Knowledge Assumptions, Extractable Functional Commitments, and Cryptanalysis

The authors of [ACL+22] showed that if we strengthen the binding property on a functional commitment for *quadratic* functions to an extractability property, then it can be used to obtain a succinct non-interactive argument (SNARG) for NP. More specifically, in an extractable functional commitment, the binding property is replaced by a stronger extractability requirement which roughly says that for any efficient adversary that outputs a commitment σ and an opening π to the value y with respect to a function f, there exists an extractor that outputs an input x such that $f(x) = y$. Extractable functional commitments for quadratic functions can be used to obtain a SNARG for NP (using the fact that satisfiability of quadratic systems is NP-complete).

In Section 4, we highlight some of the difficulties in constructing extractable functional commitments from lattices, and more generally, the challenges of formulating lattice-based knowledge assumptions. The difficulties stem from the following fundamental phenomenon about lattices, which has no analog in the pairing world: given sufficiently many independent short vectors in the kernel of a lattice $\mathbf{A}$, we can recover a trapdoor for $\mathbf{A}$ and efficiently sample short pre-images for any coset of $\mathbf{A}$. (The pairing analogue would be recovering a trapdoor that allows computing discrete logs). In our attacks, we invoke this basic fact for a carefully crafted matrix $\mathbf{A}$ derived from the verification equation of the functional commitment scheme.

Attack on knowledge k-R-ISIS. As a warm-up, we describe a candidate attack on a matrix variant of the knowledge k-R-ISIS assumption from [ACL+22].[4] Here, the adversary is given

$$\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{t \times m}, \quad \mathbf{D} \xleftarrow{R} \mathbb{Z}_q^{t \times n}, \quad \forall i \in [\ell] : \mathbf{t}_i \xleftarrow{R} \mathbb{Z}_q^n,\ \mathbf{z}_i \leftarrow \mathbf{A}^{-1}(\mathbf{D}\mathbf{t}_i)$$

[3] Technically, there is a polylogarithmic dependence on ℓ since $\log q$ scales with $\mathrm{poly}(\log \ell)$.
[4] After communicating the attack to the authors of [ACL+22], Albrecht implemented and confirmed the attack [Alb23].

where $\ell \gg m+n$ and $t \geq n+1$. The goal of the adversary is to sample $\mathbf{c} \in \mathbb{Z}_q^t$ along with a low-norm $\mathbf{v} \in \mathbb{Z}^m$ so that

$$\mathbf{Av} = \mathbf{Dc}.$$

One way to do this is to sample small integers x_i, and then compute $\mathbf{v} = \sum_{i\in[\ell]} x_i \mathbf{z}_i$ and $\mathbf{c} = \sum_{i\in[\ell]} x_i \mathbf{t}_i$. The knowledge assumption basically asserts that this is the only way to sample $(\mathbf{c}, \mathbf{v})$. In particular, if an adversary samples a *random* low-norm $\mathbf{v}$, then $\mathbf{Av}$ will lie outside the column span of $\mathbf{D}$ with high probability.

Our candidate attack uses Babai's rounding algorithm to sample small *fractional* $x_i \in \mathbb{Q}$ such that $\mathbf{v} = \sum_{i\in[\ell]} x_i \mathbf{z}_i \in \mathbb{Z}^m$ and $\mathbf{c} = \sum_{i\in[\ell]} x_i \mathbf{t}_i \in \mathbb{Z}_q^t$ and satisfies $\mathbf{Av} = \mathbf{Dc}$. It is a candidate attack in the sense that we do not know how to rule out an extractor that outputs the same distribution for $\mathbf{v}, \mathbf{c}$ using small *integer* x_i's. The attack is fairly simple (in hindsight): we first construct a basis for the lattice $\mathbf{B} = [\mathbf{A} \mid \mathbf{DG}]$ as follows:

$$[\mathbf{A} \mid \mathbf{DG}] \cdot \underbrace{\begin{bmatrix} \mathbf{z}_1 & \cdots & \mathbf{z}_\ell \\ -\mathbf{G}^{-1}(\mathbf{t}_1) & \cdots & -\mathbf{G}^{-1}(\mathbf{t}_\ell) \end{bmatrix}}_{\mathbf{T}} = \mathbf{0} \bmod q.$$

Since the $\mathbf{z}_i$'s are independent Gaussians and the $\mathbf{t}_i$'s are uniformly random, we (heuristically) assume that $\mathbf{T} \in \mathbb{Z}^{(m+n)\times\ell}$ is full rank over the *reals*.[5] Now, an adversary can start with an arbitrary (non-zero) solution $\mathbf{y} \in \mathbb{Z}^{m+n}$ where $\mathbf{By} = \mathbf{0} \bmod q$, solves for the unique $\mathbf{z} \in \mathbb{Q}^{m+n}$ where $\mathbf{Tz} = \mathbf{y} \in \mathbb{Q}^{m+n}$, and then outputs the integer vector $\mathbf{y}^* = \mathbf{y} - \mathbf{T} \cdot \lfloor \mathbf{z} \rceil$. By construction $\mathbf{By}^* = \mathbf{0} \bmod q$ and moreover, $\|\mathbf{y}^*\| \leq \|\mathbf{T}(\mathbf{z} - \lfloor \mathbf{z} \rceil)\|$, which is small. From $\mathbf{y}^*$, we can compute $\mathbf{v}, \mathbf{c}$ as desired.

Attacks on extractable functional commitments. Using a similar methodology, we obtain heuristic attacks on the extractability of our functional commitment for constant-degree polynomials described above as well as on a version of the [ACL+22] functional commitment for the particular case of linear functions. We note that [ACL+22] define their commitment over module and ideal lattices, so when describing our attack, we consider a specific translation of their scheme to the integer case. Our methodology for analyzing the extractability of functional commitments follows the general blueprint:

1. We start by writing down the key verification relation. In all lattice-based functional commitment constructions [ACL+22, WW23, dCP23, BCFL23], the verification relation consists of checking that the opening is a short solution to a linear system. We re-express the verification relation as finding a short non-zero vector in the kernel of some related lattice.

2. Using the components published in the CRS, we derive a basis for this related lattice. We now use the basis to *jointly* sample a (possibly short) commitment and a (short) opening that satisfies the main verification relation.

Importantly, the commitment and the opening are sampled without explicit knowledge of a specific input. We can apply this strategy both to our functional commitment for constant-degree polynomials as well as to an integer variant of the [ACL+22] construction:

- In the case of our functional commitment for quadratic functions, we can use the above procedure to sample a commitment and a set of valid openings that correspond to an *unsatisfiable* constraint system. For instance, we show that the attacker can efficiently come up with a commitment $\mathbf{C}$ together with valid openings asserting that $x_1^2 = 0$ and $x_1 x_2 = 1$.

- When applied to our integer-variant of the [ACL+22] functional commitment for linear functions, we can use this strategy to efficiently sample a commitment together with an opening for an *arbitrary* linear function to an arbitrary vector $\mathbf{y}$. In other words, for *any* (short) matrix $\mathbf{M}$, we can construct an efficient algorithm that samples a commitment $\mathbf{C}$ and an opening $\mathbf{V}$ to *any* target vector $\mathbf{y}$ under the linear function $\mathbf{x} \mapsto \mathbf{Mx}$. Note that this sampler does *not* need an explicit $\mathbf{x}$ to sample $(\mathbf{C}, \mathbf{V})$. If the commitment scheme is extractable, then there would exist an extractor that can output a short $\mathbf{x}$ such that $\mathbf{Mx} = \mathbf{y}$. But this is precisely solving

[5] Note that $\mathbf{T}$ does *not* (and cannot) have full rank over $\mathbb{Z}_q$.

the inhomogeneous SIS problem (with respect to a short matrix $\mathbf{M}$; hardness of inhomogeneous SIS with low-norm matrices follows from the standard setting with uniform $\mathbf{M}$ via the mapping $\mathbf{M} \mapsto \mathbf{G}^{-1}(\mathbf{M})$). Thus, our attacks demonstrates that assuming (non-uniform) hardness of the *standard* inhomogeneous SIS assumption, the variant of [ACL+22] defined over the integers does *not* satisfy extractability (i.e., the existence of an efficient extractor for our adversarial strategy implies a non-uniform polynomial-time algorithm for inhomogeneous SIS). Note that due to the way we construct the basis for the related lattice, our approach can be used to (heuristically) break inhomogeneous SIS, but not necessarily SIS. We refer to Section 4.2 for more details.

We describe our methodology and attack algorithms in Section 4. We stress that our oblivious sampling attacks only apply to the *extractability* of lattice-based functional commitments. All of the aforementioned schemes still plausibly satisfy the standard notion of binding security for functional commitments. We hope that our techniques will encourage further cryptanalysis of lattice-based knowledge assumptions (and also of the new falsifiable assumptions such as ℓ-succinct SIS) that underlie succinct commitments and arguments from lattices.

1.3 Related Work

Interactive functional commitments were first introduced in [IKO07] (for linear functions) and extended to general functions in [BC12] for realizing (interactive) succinct arguments without relying on traditional probabilistically-checkable proofs. In the interactive setting, we can also obtain a functional commitment from any collision-resistant hash function via Kilian's interactive succinct argument [Kil92]. This can be made non-interactive in the random oracle model [Mic00] through the Fiat-Shamir heuristic. Functional commitments are also generically implied by succinct non-interactive arguments of knowledge (SNARKs) for NP and collision-resistant hash functions [LRY16], but all existing constructions of SNARKs for NP either rely on strong non-falsifiable assumptions or rely on idealized models (e.g., the random oracle model or the generic group model). Our focus in this work is on non-interactive functional commitments in the plain model from *falsifiable* assumptions.

There have also been numerous constructions of functional commitments (and its specialization to vector and polynomial commitments) from standard pairing-based assumptions [LY10, KZG10, CF13, LRY16, LM19, TAB+20, GRWZ20, BCFL23] as well as assumptions over groups of unknown order such as RSA groups or class groups [CF13, LM19, CFG+20, AR20, TXN20]. We refer to [Nit21] for a survey of recent constructions. Our focus in this work is on functional commitments from lattice assumptions (similar to [PPS21, ACL+22, BCFL23, dCP23, WW23]). The work of [GVW15b] construct *non-succinct* functional commitments for arbitrary functions and fast verification from SIS; non-succinct functional commitments are often referred to as *homomorphic commitments*.

RAM delegation. A RAM delegation scheme [KP16, BHK17, KPY19, CJJ21, KVZ21, KLVW23] allows a prover to compute a short digest of an input x and later on, convince the verifier that $M(x) = y$ for an arbitrary RAM program M with a proof whose size scales with $\mathsf{poly}(\lambda, \log|x|, \log T)$, where T is the running time of the RAM computation. A RAM delegation scheme can be used to obtain a functional commitment for circuits by having the digest be over the pair (x, C), where x is the input and C is the circuit, and taking M to be the RAM program that evaluates C gate-by-gate. There is a slight syntactic mismatch here because in a functional commitment scheme, the user should be able to commit to the input x (resp., in the dual case, the circuit C) separately, and later on, open the commitment to the circuit C (resp., at the input x). However, if the underlying digest-computation algorithm has the property that the digest for the pair (x, C) can be derived from independent digests for x and C separately, then it is possible to obtain a functional commitment scheme for circuits. In recent RAM delegation schemes [CJJ21, KVZ21, KLVW23], the digest is just a Merkle hash of the inputs [Mer87], which satisfies this requirement.

Taken together, the RAM delegation schemes from [CJJ21, KVZ21] yields a functional commitments from circuits that satisfy the weaker notion of *target binding* security (where binding is only required to hold for *honestly-generated* commitments). The construction of Kalai et al. [KLVW23] yields a functional commitment for general circuits satisfies the standard notion of evaluation binding for functional commitments.[6] This yields a functional commitment scheme for all circuits from the plain LWE assumption. This scheme has a transparent setup and the size of the common reference string, commitment, and opening scale with $\mathsf{poly}(\lambda, \log|x|, \log|C|)$. While the basic approach

[6]The difference in target binding vs. evaluation binding is due to the soundness properties of the underlying RAM delegation scheme. We refer to [KLVW23, Remark 6.1] for more discussion on the different security definitions for RAM delegation.

supports openings to functions with a single output bit, it is straightforward to extend to multiple output bits by first composing with a collision-resistant hash function (i.e., instead of opening to a vector-valued $\mathbf{y} = C(\mathbf{x})$, we open to the bits of the hash output $H(\mathbf{y}) = H(C(\mathbf{x}))$; this only incurs $\mathsf{poly}(\lambda)$ overhead). The main limitation of the RAM delegation approaches is their heavy *non-black-box* use of cryptography. Namely, the constructions require the circuit description of cryptographic hash functions and lattice sampling algorithms. In this work, we focus on constructions that only need *black-box* use of cryptographic algorithms (and lattice sampling algorithms).

Relation to [Wee23]. The ℓ-succinct SIS assumption we rely on in this work was recently introduced by [Wee23], who showed how to use it (specifically, its extension to ℓ-succinct LWE) to construct succinct attribute-based encryption, reusable garbled circuits, and laconic functional encryption. The main technical result there is an attribute-based encryption scheme that achieves ciphertext overhead and key size $\mathsf{poly}(\lambda, d)$ (independent of both the attribute length and circuit size) for circuits of depth d under the ℓ-succinct LWE assumption. These aforementioned applications exploit the fact that the trapdoor $[\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}]$ can be used to "compress" the homomorphic evaluation matrix $\mathbf{H}_{\mathbf{B},f,\mathbf{x}}$, which is also the approach we take for compressing our openings in our dual functional commitment scheme.

We refer to [Wee23] for more discussion on the ℓ-succinct SIS and LWE assumptions, including reductions basing these assumptions on the evasive LWE assumption [Wee22, Tsa22]. In particular, ℓ-succinct SIS is implied by both the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption from [WW23] (the latter is in turn implied by matrix variants of k-R-ISIS, as shown in [WW23, §6]) and the evasive LWE assumption (plus LWE). In other words, ℓ-succinct SIS constitutes the "weakest" of recent non-standard lattice assumptions used in functional commitments as well as other advanced lattice-based cryptosystems.

Concurrent work. Concurrent to this work, [FLV23, CLM23] gave new constructions of lattice-based SNARKs with a linear-size CRS based on the knowledge k-R-ISIS assumption from [ACL+22]. The construction of [FLV23] leverage the k-R-ISIS assumption to construct a polynomial commitment with a linear-size CRS; in conjunction with the knowledge variant of the k-R-ISIS assumption, they obtain a lattice-based preprocessing SNARK for NP with a linear-size CRS and quasilinear prover complexity. The work of [CLM23] introduces the vanishing SIS problem and uses it to construct functional commitments for quadratic functions (and correspondingly, a preprocessing SNARK for NP). They provide two ways to instantiate their SNARK: in the plain model under the knowledge variant of the k-R-ISIS assumption, or in the random oracle model under the new, but falsifiable, vanishing SIS assumption. The results we show in this work provide strong evidence against the plausibility of the knowledge k-R-ISIS assumption, but they do not appear to directly break soundness of the SNARKs themselves. It is an interesting question to study whether our approach can be extended to break soundness of these new SNARK candidates.

2 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 $q \in \mathbb{N}$, we write $\mathbb{Z}_q$ to denote the integers modulo q. 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: $\mathbf{v} = (v_1, \ldots, v_n)$.

We write $\mathsf{poly}(\lambda)$ to denote a fixed function that is $O(\lambda^c)$ for some $c \in \mathbb{N}$ and $\mathsf{negl}(\lambda)$ to denote a function that is $o(\lambda^{-c})$ for all $c \in \mathbb{N}$. For functions $f = f(\lambda), g = g(\lambda)$, we write $g \geq O(f)$ to denote that there exists a fixed function $f'(\lambda) = O(f)$ such that $g(\lambda) > f'(\lambda)$ for all $\lambda \in \mathbb{N}$. We say an event occurs with overwhelming probability if its complement occurs with negligible probability. An algorithm is efficient if it runs in probabilistic polynomial time in its input length. 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 $\mathcal{D}_1 \overset{c}{\approx} \mathcal{D}_2$. We say that $\mathcal{D}_1$ and $\mathcal{D}_2$ are statistically indistinguishable if the statistical distance $\Delta(\mathcal{D}_1, \mathcal{D}_2)$ is bounded by a negligible function $\mathsf{negl}(\lambda)$.

Tensor products. For matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{B} \in \mathbb{Z}_q^{k \times \ell}$, we write $\mathbf{A} \otimes \mathbf{B}$ to denote the tensor (Kronecker) product of $\mathbf{A}$ and $\mathbf{B}$. For a positive integer $i \in \mathbb{N}$, we write $\mathbf{A}^{\otimes i}$ to denote tensoring $\mathbf{A}$ with itself i times. 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:

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD}). \tag{2.1}$$

The following is a useful consequence of the mixed-product property. For a vector $\mathbf{x}$ and a matrix $\mathbf{A}$,

$$(\mathbf{x} \otimes \mathbf{I})\mathbf{A} = (\mathbf{x} \otimes \mathbf{I})(1 \otimes \mathbf{A}) = \mathbf{x} \otimes \mathbf{A}. \tag{2.2}$$

Vectorization. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, 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

$$\mathrm{vec}(\mathbf{ABC}) = (\mathbf{C}^\intercal \otimes \mathbf{A}) \cdot \mathrm{vec}(\mathbf{B}).$$

2.1 Functional Commitments

In this section, we recall the formal definition of a (succinct) functional commitment. Our definition is adapted from that of [WW23].

Definition 2.1 (Succinct Functional Commitment [WW23, Definition 4.1])**.** Let λ be a security parameter. Let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ be a family of efficiently-computable functions $f : \mathcal{X}^\ell \to \mathcal{Y}^T$ with domain $\mathcal{X}^\ell$ and range $\mathcal{Y}^T$; here $\ell = \ell(\lambda)$ and $T = T(\lambda)$ denote the input dimension and the output dimension, respectively. A succinct functional commitment for $\mathcal{F}$ is a tuple of efficient algorithms $\Pi_{\mathsf{FC}} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Eval}, \mathsf{Verify})$ with the following properties:

- $\mathsf{Setup}(1^\lambda) \to \mathsf{crs}$: On input the security parameter λ, the setup algorithm outputs a common reference string crs.
- $\mathsf{Commit}(\mathsf{crs}, \mathbf{x}) \to (\sigma, \mathsf{st})$: On input the common reference string crs and an input $\mathbf{x} \in \mathcal{X}^\ell$, the commitment algorithm outputs a commitment σ and a state st.
- $\mathsf{Eval}(\mathsf{st}, f) \to \pi_f$: On input a commitment state st and a function $f \in \mathcal{F}$, the evaluation algorithm outputs an opening π_f.
- $\mathsf{Verify}(\mathsf{crs}, \sigma, f, \mathbf{y}, \pi) \to \{0, 1\}$: On input the common reference string crs, a commitment σ, a function $f \in \mathcal{F}$, a value $\mathbf{y} \in \mathcal{Y}^T$, 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 security parameters λ, all functions $f \in \mathcal{F}$, and all inputs $\mathbf{x} \in \mathcal{X}^\ell$,

$$\Pr\left[\mathsf{Verify}\big(\mathsf{crs}, \sigma, f, f(\mathbf{x}), \pi_f\big) = 1 : \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda); \\ (\sigma, \mathsf{st}) \leftarrow \mathsf{Commit}(\mathsf{crs}, \mathbf{x}); \\ \pi_f \leftarrow \mathsf{Eval}(\mathsf{st}, f) \end{array}\right] = 1 - \mathsf{negl}(\lambda).$$

- **Succinctness:** There exists a universal polynomial $\mathrm{poly}(\cdot)$ such that for all $\lambda \in \mathbb{N}$, $|\sigma| = \mathrm{poly}(\lambda + \log \ell)$ and $|\pi_f| = \mathrm{poly}(\lambda + \log \ell + T)$ in the correctness definition.
- **Binding:** We say Π_{FC} satisfies computational binding if for all efficient adversaries $\mathcal{A}$,

$$\Pr\left[\mathsf{Verify}(\mathsf{crs}, \sigma, f, \mathbf{y}_0, \pi_0) = 1 = \mathsf{Verify}(\mathsf{crs}, \sigma, f, \mathbf{y}_1, \pi_1) : \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda); \\ (\sigma, f, \mathbf{y}_0, \mathbf{y}_1, \pi_0, \pi_1) \leftarrow \mathcal{A}(1^\lambda, \mathsf{crs}) \end{array}\right] = \mathsf{negl}(\lambda).$$

Functional commitments with preprocessing. In many constructions of functional commitments, verifying an opening with respect to a function f requires time that scales with the running time of f and the size of the opening often *scales* with the output dimension T. In settings where the function f and the target $\mathbf{y}$ are known in advance (e.g., f could encode a list of predicates and the output $\mathbf{y}$ could be the all-ones vector, indicating that every predicate should be satisfied by the committed input), it is sometimes possible to decompose the verification algorithm into a "slow" offline step that takes as input the function f and the target output $\mathbf{y}$ and outputs a verification key $\mathsf{vk}_{f,\mathbf{y}}$. Importantly, $\mathsf{vk}_{f,\mathbf{y}}$ is independent of the commitment and the opening. Then, there is a fast online verification algorithm that uses the preprocessed verification key to validate the commitment and opening in time that is sublinear in the size of f and the number of outputs T. This setting is very similar to that of a preprocessing SNARK; here, the list of predicates (which may be associated with the gates of a Boolean circuit) is fixed, and the goal is to prove knowledge of a witness that satisfies all of the predicates (i.e., gate constraints).

In Remark 3.3, we note that it is also possible to preprocess the verification key when only the function f is known in advance. In this case, the online verification algorithm will need to run in time that grows with the output dimension T (since the verifier necessarily has to read the output in this case). Several recent schemes support fast verification with preprocessing [ACL+22, dCP23, BCFL23]. We define this below:

Definition 2.2 (Functional Commitment with Full Preprocessing)**.** Let λ be a security parameter. Let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of efficiently-computable functions $f : \mathcal{X}^\ell \to \mathcal{Y}^T$ where each function f can be computed by a Boolean circuit of size at most $s = s(\lambda)$. Let $\Pi_{\mathsf{FC}} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Eval}, \mathsf{Verify})$ be a succinct functional commitment for $\mathcal{F}$. We say that $\mathcal{F}$ supports preprocessing if the verification algorithm can be decomposed into two efficient algorithms $(\mathsf{Preprocess}, \mathsf{OnlineVerify})$ with the following syntax:

- $\mathsf{Preprocess}(\mathsf{crs}, f, \mathbf{y}) \to \mathsf{vk}_{f,\mathbf{y}}$: On input the common reference string crs, a function $f \in \mathcal{F}$, and an output $\mathbf{y} \in \mathcal{Y}^T$, the preprocess algorithm outputs a verification key $\mathsf{vk}_{f,\mathbf{y}}$.
- $\mathsf{OnlineVerify}(\mathsf{vk}, \sigma, \pi) \to \{0,1\}$: On input a verification key vk, a commitment σ, and an opening π, the online verification algorithm outputs a bit $b \in \{0,1\}$.

We require that

$$\mathsf{Verify}(\mathsf{crs}, \sigma, f, \mathbf{y}, \pi) := \mathsf{OnlineVerify}(\mathsf{Preprocess}(\mathsf{crs}, f, \mathbf{y}), \sigma, \pi).$$

In addition, we require the additional succinctness property:

- **Fast online verification:** There exists a universal polynomial $\mathsf{poly}(\cdot)$ such that for all $\lambda \in \mathbb{N}$, for $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$, all functions $f \in \mathcal{F}$, and all outputs $\mathbf{y} \in \mathcal{Y}^T$, the verification key $\mathsf{vk}_{f,\mathbf{y}}$ output by $\mathsf{Preprocess}(\mathsf{crs}, f, \mathbf{y})$ satisfies $|\mathsf{vk}_{f,\mathbf{y}}| = \mathsf{poly}(\lambda + \log s + \log T)$, and moreover, the running time of $\mathsf{OnlineVerify}$ is $\mathsf{poly}(\lambda + \log s + \log T)$.

Remark 2.3 (Function-Only Preprocessing)**.** We can also consider functional commitments with a weaker function-only preprocessing where the preprocessing algorithm Preprocess only takes the crs and the function f as input (but *not* the output $\mathbf{y}$) and outputs a preprocessed function key vk_f. Then, the online verification algorithm OnlineVerify takes the verification key vk_f, the output $\mathbf{y} \in \mathcal{Y}^T$, the commitment σ, and the opening π as input. In this case, we require that the size of the verification key satisfy $|\mathsf{vk}_f| = \mathsf{poly}(\lambda + \log s)$, and the verification time to be $\mathsf{poly}(\lambda + \log s + T)$. Notably, the online verification algorithm can now depend on the output dimension T (and this is required since the verification algorithm must read the output).

2.2 Lattice Preliminaries

Some parts are taken verbatim from [WW23, §2.1]. Throughout this work, we use the ℓ_∞-norm for vectors and matrices. For a vector $\mathbf{u}$, we write $\|\mathbf{u}\| := \max_i |x_i|$, and for a matrix $\mathbf{A}$, we write $\|\mathbf{A}\| = \max_{i,j} |A_{i,j}|$.

Min-entropy. We recall some basic definitions on min-entropy. Our definitions are taken from [DRS04]. For a (discrete) random variable X, we write $\mathbf{H}_\infty(X) = -\log(\max_x \Pr[X = x])$ to denote its min-entropy. For two (possibly correlated) discrete random variables X and Y, we define the average min-entropy of X given Y to be $\mathbf{H}_\infty(X \mid Y) = -\log(\mathbb{E}_{y\leftarrow Y} \max_x \Pr[X = x \mid Y = y])$. We now state the (generalized) leftover hash lemma:

Lemma 2.4 (Leftover Hash Lemma [HILL99, DRS04]). *Let n, m, q be lattice parameters. Let $\mathbf{x} \in \{0,1\}^m$, $\mathsf{aux} \in \{0,1\}^*$ be (arbitrarily correlated) random variables where $\mathbf{H}_\infty(\mathbf{x} \mid \mathsf{aux}) \geq 2\lambda + n \log q$. Then, the statistical distance between the following distributions is at most $2^{-\lambda}$:*

$$\{(\mathbf{A}, \mathbf{Ax}, \mathsf{aux}) : \mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n \times m}\} \quad \text{and} \quad \{(\mathbf{A}, \mathbf{u}, \mathsf{aux}) : \mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n \times m}, \mathbf{u} \xleftarrow{R} \mathbb{Z}_q^n\}.$$

Discrete Gaussians. We write $D_{\mathbb{Z},\chi}$ to denote the discrete Gaussian distribution over $\mathbb{Z}$ with width parameter $\chi > 0$. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times t}$, and a vector $\mathbf{v} \in \mathbb{Z}_q^n$, we write $\mathbf{A}_\chi^{-1}(\mathbf{v})$ to denote the random variable $\mathbf{x} \leftarrow D_{\mathbb{Z},\chi}^m$ conditioned on $\mathbf{Ax} = \mathbf{v} \bmod q$. We extend $\mathbf{A}_\chi^{-1}$ to matrices by applying $\mathbf{A}_\chi^{-1}$ to each column of the input. We now recall some useful properties on discrete Gaussian distributions over lattices:

Lemma 2.5 (Gaussian Tail Bound [MP12, Lemma 2.6, adapted]). *Let n, m, q be lattice parameters where $m \geq O(n \log q)$. Sample $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n \times m}$. Then, for all $\chi \geq \omega(\sqrt{\log m})$ and all vectors $\mathbf{v} \in \mathbb{Z}_q^n$ in the span of $\mathbf{A}$,*

$$\Pr[\|\mathbf{u}\| > \sqrt{m}\chi : \mathbf{u} \leftarrow \mathbf{A}_\chi^{-1}(\mathbf{v})] = \mathsf{negl}(n).$$

For the particular case of the discrete Gaussian distribution over the integers,

$$\Pr[|x| > \sqrt{\lambda}\chi : x \leftarrow D_{\mathbb{Z},\chi}] = \mathsf{negl}(\lambda).$$

Lemma 2.6 (Min-Entropy of a Discrete Gaussian [PR06, Lemma 2.11, adapted]). *Let n, q be lattice parameters and suppose $m \geq 2n \log q$. Let $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n \times m}$. Then, with $1 - \mathsf{negl}(n)$ probability over the choice of $\mathbf{A}$, for all $\chi \geq \omega(\sqrt{\log m})$, and any $\mathbf{x}^* \in \mathbb{Z}_q^n$, $\mathbf{y} \in \mathbb{Z}_q^m$,*

$$\Pr[\mathbf{x} = \mathbf{x}^* : \mathbf{x} \leftarrow \mathbf{A}_\chi^{-1}(\mathbf{y})] \leq \mathsf{negl}(n).$$

Lemma 2.7 (Discrete Gaussian Preimages [WW23, Lemma 2.7, adapted]). *Let n, q be lattice parameters with q prime and suppose $m \geq 2n \log q$. Take any $t = \mathsf{poly}(n, \log q)$ and $\ell = \mathsf{poly}(n, \log q)$. Sample $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n \times m}$. Then, for all matrices $\mathbf{B} \in \mathbb{Z}_q^{nt \times \ell}$, all target vectors $\mathbf{t} \in \mathbb{Z}_q^{nt}$, all width parameters $\chi \geq \omega(\sqrt{\log m})$, and setting $\mathbf{C} = [\mathbf{I}_t \otimes \mathbf{A} \mid \mathbf{B}] \in \mathbb{Z}_q^{nt \times (mt+\ell)}$, the statistical distance between the following distributions is $\mathsf{negl}(n)$:*

$$\{\mathbf{v} : \mathbf{v} \leftarrow \mathbf{C}_\chi^{-1}(\mathbf{t})\} \quad \text{and} \quad \left\{ \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} : \mathbf{v}_2 \leftarrow D_{\mathbb{Z},\chi}^\ell, \mathbf{v}_1 \leftarrow (\mathbf{I}_t \otimes \mathbf{A})_\chi^{-1}(\mathbf{t} - \mathbf{B}\mathbf{v}_2) \right\}.$$

The gadget matrix. We recall the definition of the gadget matrix [MP12]. For positive integers $n, q \in \mathbb{N}$, let $\mathbf{G}_n = \mathbf{I}_n \otimes \mathbf{g}^\intercal \in \mathbb{Z}_q^{n \times m'}$ be the gadget matrix where $\mathbf{g}^\intercal = [1, 2, \ldots, 2^{\lfloor \log q \rfloor}]$ and $m' = n(\lfloor \log q \rfloor + 1)$. For dimensions $m \geq m'$, we overload the notation and write $\mathbf{G}_n \in \mathbb{Z}_q^{n \times m}$ to denote the "padded gadget matrix" $[\mathbf{I}_n \otimes \mathbf{g}^\intercal \mid \mathbf{0}^{n \times (m-m')}]$. The inverse function $\mathbf{G}_n^{-1} : \mathbb{Z}_q^{n \times t} \to \mathbb{Z}_q^{m' \times t}$ expands each entry $x \in \mathbb{Z}_q$ into a column of size $\lfloor \log q \rfloor + 1$ consisting of the bits in the binary representation of x. Similarly, when $\mathbf{G}_n \in \mathbb{Z}_q^{n \times m}$ is a padded gadget matrix with dimension $m \geq m'$, we extend the output of $\mathbf{G}_n^{-1} : \mathbb{Z}_q^{n \times t} \to \mathbb{Z}_q^{m \times t}$ by zero-padding each column. For every matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times t}$, it follows that $\mathbf{G}_n \cdot \mathbf{G}_n^{-1}(\mathbf{A}) = \mathbf{A} \bmod q$. When the dimension n is clear, we omit the subscript and simply write $\mathbf{G}$ and $\mathbf{G}^{-1}(\cdot)$ to denote $\mathbf{G}_n$ and $\mathbf{G}_n^{-1}(\cdot)$, respectively.

Gadget trapdoors. Our constructions will use the gadget trapdoors from [MP12], which builds on a long sequence of works on constructing lattice trapdoors [Ajt96, GPV08, AP09, ABB10a, ABB10b, CHKP10].

Theorem 2.8 (Gadget Trapdoor [MP12, adapted]). *Let n, m, q be lattice parameters with $m \geq O(n \log q)$. Then there exist efficient algorithms* $(\mathsf{TrapGen}, \mathsf{SamplePre})$ *with the following syntax:*

- $\mathsf{TrapGen}(1^n, q, m) \to (\mathbf{A}, \mathbf{R})$*: On input the lattice dimension n, the modulus q, and the number of samples m, the trapdoor-generation algorithm outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ together with a trapdoor $\mathbf{R} \in \mathbb{Z}_q^{m \times m}$.*
- $\mathsf{SamplePre}(\mathbf{A}, \mathbf{R}, \mathbf{v}, \chi) \to \mathbf{u}$*: On input a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a trapdoor $\mathbf{R} \in \mathbb{Z}_q^{m \times m}$, a target vector $\mathbf{v} \in \mathbb{Z}_q^n$, and a Gaussian width parameter χ, the preimage-sampling algorithm outputs a vector $\mathbf{u} \in \mathbb{Z}_q^m$.*

Moreover, the above algorithms satisfy the following properties:

- **Trapdoor distribution:** *If* $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$ *and* $\mathbf{A}' \xleftarrow{R} \mathbb{Z}_q^{n \times m}$*, then* $\Delta(\mathbf{A}, \mathbf{A}') \leq 2^{-n}$*. Moreover,* $\mathbf{AR} = \mathbf{G}$ *and* $\|\mathbf{R}\| = 1$.
- **Preimage sampling:** *For all matrices* $\mathbf{R} \in \mathbb{Z}_q^{m \times m}$*, parameters* $\chi > 0$*, and all target vectors* $\mathbf{v} \in \mathbb{Z}_q^n$ *in the column span of* $\mathbf{A}$*, the output* $\mathbf{u} \leftarrow \mathsf{SamplePre}(\mathbf{A}, \mathbf{R}, \mathbf{v}, \chi)$ *of* SamplePre *satisfies* $\mathbf{Au} = \mathbf{v}$.
- **Preimage distribution:** *Suppose* $\mathbf{R}$ *is a gadget trapdoor for* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *(i.e.,* $\mathbf{AR} = \mathbf{G}$*). Then, for all* $\chi \geq m \|\mathbf{R}\| \cdot \omega(\sqrt{\log n})$*, and all target vectors* $\mathbf{v} \in \mathbb{Z}_q^n$*, the statistical distance between the following distributions is at most* 2^{-n}*:*

  $$\{\mathbf{u} \leftarrow \mathsf{SamplePre}(\mathbf{A}, \mathbf{R}, \mathbf{v}, \chi)\} \quad \text{and} \quad \{\mathbf{u} \leftarrow \mathbf{A}_\chi^{-1}(\mathbf{v})\}.$$

  More generally, the above properties also hold if $\mathbf{AR} = \mathbf{HG}$ *for some invertible matrix* $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$*. In addition, the* SamplePre *algorithm extends naturally to block-diagonal matrices. Namely, for any* ℓ *and any* $\mathbf{v} \in \mathbb{Z}_q^{\ell n}$ *and under the same conditions as above, the statistical distance between the following distributions is at most* $\ell \cdot 2^{-n}$*:*

  $$\{\mathbf{u} \leftarrow \mathsf{SamplePre}(\mathbf{I}_\ell \otimes \mathbf{A}, \mathbf{I}_\ell \otimes \mathbf{R}, \mathbf{v}, \chi)\} \quad \text{and} \quad \{\mathbf{u} \leftarrow (\mathbf{I} \otimes \mathbf{A})_\chi^{-1}(\mathbf{v})\}.$$

For a matrix $\mathbf{V} \in \mathbb{Z}_q^{n \times \ell}$*, we define* $\mathsf{SamplePre}(\mathbf{A}, \mathbf{R}, \mathbf{V}, \chi)$ *to be the algorithm that outputs the matrix where the* i^{th} *column is* $\mathsf{SamplePre}(\mathbf{A}, \mathbf{R}, \mathbf{v}_i, \chi)$ *and* $\mathbf{v}_i$ *denotes the* i^{th} *column of* $\mathbf{V}$.

Remark 2.9 (Trapdoor Extension [ABB10b, CHKP10, MP12])**.** Suppose $\mathbf{R} \in \mathbb{Z}_q^{m \times m'}$ is a gadget trapdoor for a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. Then, for every matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times t}$, $\mathbf{R}' = \left[\begin{smallmatrix} \mathbf{R} \\ \mathbf{0} \end{smallmatrix}\right]$ is a gadget trapdoor for $[\mathbf{A} \mid \mathbf{B}]$ and $\|\mathbf{R}\| = \|\mathbf{R}'\|$. To simplify notation, we will overload the preimage sampling algorithm SamplePre from Theorem 2.8 and write $\mathsf{SamplePre}([\mathbf{A} \mid \mathbf{B}], \mathbf{R}, \mathbf{v}, \chi)$ to denote $\mathsf{SamplePre}([\mathbf{A} \mid \mathbf{B}], \mathbf{R}', \mathbf{v}, \chi)$.

Ajtai trapdoors. When analyzing our new hardness assumptions, it will also be convenient to use the more traditional lattice trapdoors introduced by Ajtai [Ajt96] and subsequently expanded by a number of subsequent works [GPV08, AP09, ABB10b, ABB10a, CHKP10, LW15]. We recall the main property we need:

Definition 2.10 (Ajtai Trapdoor [Ajt96])**.** Let n, m, q be lattice parameters and $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be a matrix. We say that a matrix $\mathbf{R} \in \mathbb{Z}^{m \times m}$ is an Ajtai-trapdoor for $\mathbf{A}$ if $\mathbf{AR} = \mathbf{0} \bmod q$ and $\mathbf{R}$ is full rank over $\mathbb{R}$. We write $\tilde{\mathbf{R}} \in \mathbb{R}^{m \times m}$ to denote the Gram-Schmidt orthogonalization of the columns of $\mathbf{R}$ (from left to right).

Theorem 2.11 (Preimage Sampling [GPV08])**.** *There exists an efficient algorithm* $\mathsf{SamplePre}'(\mathbf{A}, \mathbf{R}, \mathbf{v}, \chi)$ *that takes as input a matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$*, an Ajtai-trapdoor* $\mathbf{R} \in \mathbb{Z}^{m \times m}$ *for* $\mathbf{A}$*, a target vector* $\mathbf{v} \in \mathbb{Z}_q^n$ *in the column-span of* $\mathbf{A}$*, and a Gaussian width parameter* χ*, and outputs a vector* $\mathbf{u} \in \mathbb{Z}_q^m$ *such that* $\mathbf{Au} = \mathbf{v}$*. Moreover, if* $\chi \geq \|\tilde{\mathbf{R}}\| \cdot \omega\left(\sqrt{m \log m}\right)$*, then the statistical distance between the following distributions is* $\mathsf{negl}(n)$*:*

$$\{\mathbf{u} \leftarrow \mathsf{SamplePre}'(\mathbf{A}, \mathbf{R}, \mathbf{v}, \chi)\} \quad \text{and} \quad \{\mathbf{u} \leftarrow \mathbf{A}_\chi^{-1}(\mathbf{v})\}$$

Short integer solutions. We now recall the short integer solution (SIS) problem [Ajt96].

Assumption 2.12 (Short Integer Solution [Ajt96])**.** Let λ be a security parameter and $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$, and $\beta = \beta(\lambda)$ be lattice parameters. We say the short integer solution problem $\mathsf{SIS}_{n,m,q,\beta}$ holds if for all efficient adversaries $\mathcal{A}$,

$$\Pr\left[\mathbf{Ax} = \mathbf{0} \text{ and } 0 < \|\mathbf{x}\| \leq \beta : \begin{array}{c} \mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n \times m}; \\ \mathbf{x} \leftarrow \mathcal{A}(1^\lambda, \mathbf{A}) \end{array}\right] = \mathsf{negl}(\lambda).$$

We also define the *inhomogeneous* SIS assumption where the target $\mathbf{0}$ in the above assumption is replaced by a uniform random vector $\mathbf{y} \xleftarrow{R} \mathbb{Z}_q^n$.

When $m = \mathsf{poly}(n, \log q)$ and $q \geq \beta \cdot \mathsf{poly}(n)$, hardness of SIS and inhomogeneous SIS can be based on the hardness of approximating worst-case lattice problems on n-dimensional lattices to within a $\beta \cdot \mathsf{poly}(n)$ factor [Ajt96, MR04, GPV08].

Homomorphic evaluation. Our construction of succinct functional commitments will rely on the lattice homomorphic evaluation procedure developed in [GSW13, BGG+14]. Our presentation is adapted from that in [BV15, BCTW16, BTVW17].

Theorem 2.13 (Homomorphic Encodings [GSW13, BGG+14])**.** *Let λ be a security parameter and $n = n(\lambda)$, $q = q(\lambda)$ be lattice parameters. Take any $m \geq n(\lfloor \log q \rfloor + 1)$, and let $\ell = \ell(\lambda)$ be an input length. Let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ be a family of functions $f : \{0,1\}^\ell \to \{0,1\}$ that can be computed by a Boolean circuit of depth at most $d = d(\lambda)$. Then, there exist a pair of efficient algorithms* $(\mathsf{EvalF}, \mathsf{EvalFX})$ *with the following properties:*

- $\mathsf{EvalF}(\mathbf{A}, f) \to \mathbf{A}_f$*: On input a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times \ell m}$ and a function $f \in \mathcal{F}$, the input-independent evaluation algorithm outputs a matrix $\mathbf{A}_f \in \mathbb{Z}_q^{n \times m}$.*
- $\mathsf{EvalFX}(\mathbf{A}, f, \mathbf{x}) \to \mathbf{H}_{\mathbf{A},f,\mathbf{x}}$*: On input a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times \ell m}$, a function $f \in \mathcal{F}$, and an input $\mathbf{x} \in \{0,1\}^\ell$, the input-dependent evaluation algorithm outputs a matrix $\mathbf{H}_{\mathbf{A},f,\mathbf{x}} \in \mathbb{Z}_q^{\ell m \times m}$.*

Moreover for all security parameters $\lambda \in \mathbb{N}$, matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times \ell m}$, all functions $f \in \mathcal{F}$, and all inputs $\mathbf{x} \in \{0,1\}^\ell$, the matrices $\mathbf{A}_f \leftarrow \mathsf{EvalF}(\mathbf{A}, f)$ and $\mathbf{H}_{\mathbf{A},f,\mathbf{x}} \leftarrow \mathsf{EvalFX}(\mathbf{A}, f, \mathbf{x})$ satisfy the following properties:

- $\|\mathbf{H}_{\mathbf{A},f,\mathbf{x}}\| \leq (n \log q)^{O(d)}$.
- $(\mathbf{A} - \mathbf{x}^\intercal \otimes \mathbf{G}) \cdot \mathbf{H}_{\mathbf{A},f,\mathbf{x}} = \mathbf{A}_f - f(\mathbf{x}) \cdot \mathbf{G}$.

3 Functional Commitments with Fast Verification

In this section, we show how to construct a functional commitment for constant-degree polynomials that support fast verification. Security of our construction relies on the ℓ-succinct short integer solutions problem from [Wee23], which we recall below:

Assumption 3.1 (ℓ-Succinct SIS [Wee23])**.** Let λ be a security parameter and $n = n(\lambda), m = m(\lambda), q = q(\lambda), \chi = \chi(\lambda)$, and $\beta = \beta(\lambda)$ be lattice parameters. We say that the ℓ-succinct SIS assumption with parameters (n, m, q, χ, β) holds if for all efficient adversaries $\mathcal{A}$,

$$\Pr\left[\mathbf{A}\mathbf{x} = \mathbf{0} \text{ and } 0 < \|\mathbf{x}\| \leq \beta : \begin{array}{c} \mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n \times m}, \mathbf{W} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n\ell \times m}, \\ \mathbf{R} \leftarrow [\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}]_\chi^{-1}(\mathbf{G}_{n\ell}) \\ \mathbf{x} \leftarrow \mathcal{A}(1^\lambda, \mathbf{A}, \mathbf{W}, \mathbf{R}) \end{array}\right] = \mathsf{negl}(\lambda).$$

As suggested in [Wee23], we consider parameter settings for (n, m, q, β) where $\mathsf{SIS}_{n,m,q,\beta}$ hold and where $\chi = \mathsf{poly}(\lambda, m, \ell)$.

Construction 3.2 (Functional Commitment for Constant-Degree Polynomials)**.** Let λ be a security parameter and $n = n(\lambda), m = m(\lambda), q = q(\lambda), \chi = \chi(\lambda)$ be lattice parameters. We define the following additional parameters:

- Let $\ell = \ell(\lambda)$ be an input length parameter, $d_{\max} = O(1)$ be a *constant* degree bound, $B_{\mathsf{in}} = B_{\mathsf{in}}(\lambda)$ be a bound on the magnitude of the inputs, and $B_{\mathsf{out}} = B_{\mathsf{out}}(\lambda)$ be a bound on the magnitude of the outputs.
- Let $L = \sum_{i \in [d_{\max}]} \ell^i$ and $B = B(\lambda)$ be a verification bound.
- Let $\mathcal{F}_\lambda$ be the set of functions $f : [-B_{\mathsf{in}}, B_{\mathsf{in}}]^\ell \to [-B_{\mathsf{out}}, B_{\mathsf{out}}]$ where f can be computed by a *homogeneous* polynomial[7] with B_{in}-bounded coefficients and degree at most $d_{\max}$. For each function $f \in \mathcal{F}_\lambda$, we associate a vector $\mathbf{f} \in [-B_{\mathsf{in}}, B_{\mathsf{in}}]^{\ell^d}$ for some $d \leq d_{\max}$ and define $f(\mathbf{x}) := \mathbf{f}^\intercal \mathbf{x}^{\otimes d}$.

We construct a functional commitment $\Pi_{\mathsf{FC}} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Eval}, \mathsf{Verify})$ for $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ as follows:

[7] A functional commitment scheme for homogeneous polynomials implies one for non-homogeneous polynomial by padding the input with a constant-value 1. See also Remark 3.4.

- $\mathsf{Setup}(1^\lambda)$: On input the security parameter λ, the setup algorithm samples $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$ and $\mathbf{W} \xleftarrow{R} \mathbb{Z}_q^{Ln \times m}$. Next, define the target matrix

$$\mathbf{P} = \begin{bmatrix} \mathbf{I}_\ell \otimes \mathbf{G} \\ \mathbf{I}_\ell \otimes \mathbf{W}_1 \\ \vdots \\ \mathbf{I}_\ell \otimes \mathbf{W}_{d_{\max}-1} \end{bmatrix} \in \mathbb{Z}_q^{Ln \times \ell m} \quad \text{where} \quad \mathbf{W} = \begin{bmatrix} \mathbf{W}_1 \\ \vdots \\ \mathbf{W}_{d_{\max}} \end{bmatrix} \in \mathbb{Z}_q^{Ln \times m}, \tag{3.1}$$

  where $\mathbf{W}_i \in \mathbb{Z}_q^{\ell^i n \times m}$. Then, compute $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_L \otimes \mathbf{A} \mid \mathbf{W}], \mathbf{I}_L \otimes \mathbf{R}, \mathbf{P}, \chi) \in \mathbb{Z}_q^{(Lm+m) \times \ell m}$. Parse $\mathbf{T} = \begin{bmatrix} \mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix}$ where $\mathbf{T}_{\mathsf{open}} \in \mathbb{Z}_q^{Lm \times \ell m}$ and $\mathbf{T}_{\mathsf{com}} \in \mathbb{Z}_q^{m \times \ell m}$. Output the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}})$.

- $\mathsf{Commit}(\mathsf{crs}, \mathbf{x})$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}})$ and an input $\mathbf{x} \in [-B_{\mathsf{in}}, B_{\mathsf{in}}]^\ell$, the commit algorithm outputs the commitment $\sigma = \mathbf{C} = \mathbf{T}_{\mathsf{com}}(\mathbf{x} \otimes \mathbf{I}_m) \in \mathbb{Z}_q^{m \times m}$ and the state $\mathsf{st} = \mathbf{x}$.

- $\mathsf{Eval}(\mathsf{crs}, \mathsf{st}, f)$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}})$, the state $\mathsf{st} = \mathbf{x}$, and a function $f = \mathbf{f} \in \mathbb{Z}_q^{\ell^d}$ (for some $d \le d_{\max}$) with B_{in}-bounded coefficients, the evaluation algorithm first computes $\mathbf{V} = \mathbf{T}_{\mathsf{open}}(\mathbf{x} \otimes \mathbf{I}_m)$. It then parses

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1 \\ \vdots \\ \mathbf{V}_{d_{\max}} \end{bmatrix} \in \mathbb{Z}_q^{Lm \times m} \tag{3.2}$$

  where $\mathbf{V}_i \in \mathbb{Z}_q^{\ell^i m \times m}$. Let $\mathbf{V}'_1 \leftarrow \mathbf{V}_1$ and for $i \in [d]$, let $\mathbf{V}'_i \leftarrow (\mathbf{x} \otimes \mathbf{I}_{\ell^{i-1}m})\mathbf{V}'_{i-1} + \mathbf{V}_i \mathbf{C}^{i-1} \in \mathbb{Z}_q^{\ell^i m \times m}$. Equivalently, in expanded form, we can write

$$\begin{aligned} \mathbf{V}'_i &= \mathbf{V}_i \mathbf{C}^{i-1} + (\mathbf{x} \otimes \mathbf{I}_{\ell^{i-1}m})\mathbf{V}_{i-1}\mathbf{C} + (\mathbf{x}^{\otimes 2} \otimes \mathbf{I}_{\ell^{i-2}m})\mathbf{V}_{i-2}\mathbf{C}^2 + \cdots + (\mathbf{x}^{\otimes i-1} \otimes \mathbf{I}_{\ell m})\mathbf{V}_1 \\ &= \sum_{j \in [i]} (\mathbf{x}^{\otimes i-j} \otimes \mathbf{I}_{\ell^j m})\mathbf{V}_j \mathbf{C}^{j-1} \end{aligned}$$

  Output the opening $\pi_f = \mathbf{V}_f = (\mathbf{f}^\intercal \otimes \mathbf{I}_m)\mathbf{V}'_d \in \mathbb{Z}_q^{m \times m}$.

- $\mathsf{Verify}(\mathsf{crs}, \sigma, f, y, \pi)$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}})$, the commitment $\sigma = \mathbf{C} \in \mathbb{Z}_q^{m \times m}$, the output $y \in [-B_{\mathsf{out}}, B_{\mathsf{out}}]$, a function $f = \mathbf{f} \in \mathbb{Z}_q^{\ell^d}$ (for some $d \le d_{\max}$) with B_{in}-bounded coefficients, and the proof $\pi = \mathbf{V} \in \mathbb{Z}_q^{m \times m}$, the verification algorithm first parses $\mathbf{W}$ into $\mathbf{W}_1, \ldots, \mathbf{W}_{d_{\max}}$ as in Eq. (3.1) and outputs 1 if

$$\|\mathbf{V}\| \le B \quad \text{and} \quad (\mathbf{f}^\intercal \otimes \mathbf{I}_m)\mathbf{W}_d \mathbf{C}^d = y \cdot \mathbf{G} - \mathbf{A}\mathbf{V}. \tag{3.3}$$

Remark 3.3 (Supporting Preprocessing)**.** Similar to previous (non-succinct) homomorphic commitments [GVW15b] and succinct functional commitments [ACL+22, dCP23, BCFL23], our functional commitment (Construction 3.2) supports fast verification in the preprocessing model. Note that since the output dimension is 1, we do not distinguish between function-only preprocessing (Remark 2.3) and full preprocessing (Definition 2.2). We define the preprocessing and online verification algorithms as follows:

- $\mathsf{Preprocess}(\mathsf{crs}, f)$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}})$ and the function $f = \mathbf{f} \in \mathbb{Z}_q^{\ell^d}$ for some $d \le d_{\max}$, the preprocess algorithm outputs $\mathsf{vk}_f = \mathbf{F}_d = (\mathbf{f}^\intercal \otimes \mathbf{I}_m)\mathbf{W}_d \in \mathbb{Z}_q^{n \times m}$.

- $\mathsf{OnlineVerify}(\mathsf{vk}, \sigma, y, \pi)$: On input the verification key $\mathsf{vk} = \mathbf{F}_d$, the commitment $\sigma = \mathbf{C} \in \mathbb{Z}_q^{m \times m}$, the value $y \in [-B_{\mathsf{out}}, B_{\mathsf{out}}]$, and the opening $\pi = \mathbf{V} \in \mathbb{Z}_q^{m \times m}$, the online verification algorithm outputs 1 if

$$\|\mathbf{V}\| \le B \quad \text{and} \quad \mathbf{F}_d \cdot \mathbf{C}^d = y \cdot \mathbf{G} - \mathbf{A}\mathbf{V}.$$

By construction, $|\mathbf{F}_d| = nm \log q$ and similarly, the online verification algorithm runs in time $\mathsf{poly}(n, m, d_{\max}, \log q)$. We can set the parameters for Construction 3.2, so $n, m, \log q$ scale polylogarithmically with the input dimension ℓ.

Remark 3.4 (Supporting Non-Homogeneous Polynomials)**.** It is straightforward to extend a functional commitment for homogeneous polynomials (i.e., polynomials where every monomial has the same degree) to a functional commitment for inhomogeneous polynomials. Specifically, to support openings to inhomogeneous polynomials over inputs of dimension ℓ, we instantiate a scheme that supports homogeneous polynomials over inputs of dimension $\ell + 1$. Then to commit to an input $\mathbf{x} \in \mathbb{Z}_q^\ell$, the committer commits to the extended vector $\mathbf{x}' = \left[\begin{smallmatrix} 1 \\ \mathbf{x} \end{smallmatrix}\right]$. Now, every inhomogeneous polynomial $f : \mathbb{Z}_q^\ell \to \mathbb{Z}_q$ of degree at most d can be described by a *homogeneous* polynomial $f' : \mathbb{Z}_q^{\ell+1} \to \mathbb{Z}_q$ of degree d where $f'(\mathbf{x}') = f(\mathbf{x})$. Now, to open to an inhomogeneous polynomial f, the committer instead open to f'.

Theorem 3.5 (Correctness)**.** *Suppose* $n \geq \lambda, m \geq O(n \log q), \chi \geq O(m \log n)$, *and* $B \geq O\big(B_{\mathsf{in}}^{d_{\max}+1} \ell^{2d_{\max}} \chi^{d_{\max}} m^{(3d_{\max}-2)/2}\big)$. *Then, Construction 3.2 is correct.*

Proof. Take any input $\mathbf{x} \in [-B_{\mathsf{in}}, B_{\mathsf{in}}]^\ell$ and function $f = \mathbf{f} \in \mathbb{Z}_q^{\ell^d}$ for some $d \leq d_{\max}$. Let $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}) \leftarrow \mathsf{Setup}(1^\lambda)$. Suppose $(\mathbf{C}, \mathsf{st}) \leftarrow \mathsf{Commit}(\mathsf{crs}, \mathbf{x})$ and $\mathbf{V}_f \leftarrow \mathsf{Eval}(\mathsf{crs}, \mathsf{st}, f)$. Then

$$\begin{bmatrix} \mathbf{V} \\ \mathbf{C} \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{\mathsf{open}}(\mathbf{x} \otimes \mathbf{I}_m) \\ \mathbf{T}_{\mathsf{com}}(\mathbf{x} \otimes \mathbf{I}_m) \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix} (\mathbf{x} \otimes \mathbf{I}_m).$$

Parse $\mathbf{W}$ into $\mathbf{W}_1, \ldots, \mathbf{W}_{d_{\max}}$ according to Eq. (3.1). By construction of Setup and using Eq. (2.2),

$$\left[\begin{array}{ccc|c} \mathbf{I}_\ell \otimes \mathbf{A} & & & \mathbf{W}_1 \\ & \mathbf{I}_{\ell^2} \otimes \mathbf{A} & & \mathbf{W}_2 \\ & & \ddots & \vdots \\ & & \mathbf{I}_{\ell^{d_{\max}}} \otimes \mathbf{A} & \mathbf{W}_{d_{\max}} \end{array}\right] \cdot \begin{bmatrix} \mathbf{V} \\ \mathbf{C} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_\ell \otimes \mathbf{G} \\ \mathbf{I}_\ell \otimes \mathbf{W}_1 \\ \vdots \\ \mathbf{I}_\ell \otimes \mathbf{W}_{d_{\max}-1} \end{bmatrix} (\mathbf{x} \otimes \mathbf{I}_m) = \begin{bmatrix} (\mathbf{x} \otimes \mathbf{I}_n)\mathbf{G} \\ (\mathbf{x} \otimes \mathbf{I}_{\ell n})\mathbf{W}_1 \\ \vdots \\ (\mathbf{x} \otimes \mathbf{I}_{\ell^{d_{\max}-1} n})\mathbf{W}_{d_{\max}-1} \end{bmatrix}.$$

Thus, for all $i \in [d_{\max}]$, we have that

$$\mathbf{W}_i \mathbf{C} = (\mathbf{x} \otimes \mathbf{I}_{\ell^{i-1} n})\mathbf{W}_{i-1} - (\mathbf{I}_{\ell^i} \otimes \mathbf{A})\mathbf{V}_i, \tag{3.4}$$

where $\mathbf{V}_i \in \mathbb{Z}_q^{\ell^i m \times m}$ is defined according to Eq. (3.2) and $\mathbf{W}_0 = \mathbf{G}$. Now, we claim that for all $i \in [d_{\max}]$,

$$\mathbf{W}_i \mathbf{C}^i = \mathbf{x}^{\otimes i} \otimes \mathbf{G} - (\mathbf{I}_{\ell^i} \otimes \mathbf{A})\mathbf{V}'_i. \tag{3.5}$$

For the base case where $i = 1$, $\mathbf{V}'_1 = \mathbf{V}_1$ and the claim follows by Eq. (3.4). For the general case, we have from Eq. (3.4)

$$\begin{aligned} \mathbf{W}_{i+1}\mathbf{C}^{i+1} = \mathbf{W}_{i+1}\mathbf{C}\mathbf{C}^i &= (\mathbf{x} \otimes \mathbf{I}_{\ell^i n})\mathbf{W}_i \mathbf{C}^i - (\mathbf{I}_{\ell^{i+1}} \otimes \mathbf{A})\mathbf{V}_{i+1}\mathbf{C}^i \\ &= (\mathbf{x} \otimes \mathbf{I}_{\ell^i n})\big((\mathbf{x}^{\otimes i} \otimes \mathbf{G}) - (\mathbf{I}_{\ell^i} \otimes \mathbf{A})\mathbf{V}'_i\big) - (\mathbf{I}_{\ell^{i+1}} \otimes \mathbf{A})\mathbf{V}_{i+1}\mathbf{C}^i \\ &= (\mathbf{x} \otimes \mathbf{I}_{\ell^i n})(\mathbf{x}^{\otimes i} \otimes \mathbf{G}) - (\mathbf{x} \otimes \mathbf{I}_{\ell^i n})(\mathbf{I}_{\ell^i} \otimes \mathbf{A})\mathbf{V}'_i - (\mathbf{I}_{\ell^{i+1}} \otimes \mathbf{A})\mathbf{V}_{i+1}\mathbf{C}^i. \end{aligned} \tag{3.6}$$

Using Eqs. (2.1) and (2.2), we can write

$$\begin{aligned} (\mathbf{x} \otimes \mathbf{I}_{\ell^i n})(\mathbf{x}^{\otimes i} \otimes \mathbf{G}) &= \mathbf{x} \otimes \mathbf{x}^{\otimes i} \otimes \mathbf{G} = \mathbf{x}^{\otimes i+1} \otimes \mathbf{G} \\ (\mathbf{x} \otimes \mathbf{I}_{\ell^i n})(\mathbf{I}_{\ell^i} \otimes \mathbf{A}) &= \mathbf{x} \otimes (\mathbf{I}_{\ell^i} \otimes \mathbf{A}) = (\mathbf{I}_\ell \otimes (\mathbf{I}_{\ell^i} \otimes \mathbf{A}))(\mathbf{x} \otimes \mathbf{I}_{\ell^i m}) = (\mathbf{I}_{\ell^{i+1}} \otimes \mathbf{A})(\mathbf{x} \otimes \mathbf{I}_{\ell^i m}). \end{aligned}$$

Substituting back into Eq. (3.6), we have

$$\begin{aligned} \mathbf{W}_{i+1}\mathbf{C}^{i+1} &= (\mathbf{x} \otimes \mathbf{I}_{\ell^i n})(\mathbf{x}^{\otimes i} \otimes \mathbf{G}) - (\mathbf{x} \otimes \mathbf{I}_{\ell^i n})(\mathbf{I}_{\ell^i} \otimes \mathbf{A})\mathbf{V}'_i - (\mathbf{I}_{\ell^{i+1}} \otimes \mathbf{A})\mathbf{V}_{i+1}\mathbf{C}^i \\ &= \mathbf{x}^{\otimes i+1} \otimes \mathbf{G} - (\mathbf{I}_{\ell^{i+1}} \otimes \mathbf{A})\left((\mathbf{x} \otimes \mathbf{I}_{\ell^i m})\mathbf{V}'_i + \mathbf{V}_{i+1}\mathbf{C}^i\right) \\ &= \mathbf{x}^{\otimes i+1} \otimes \mathbf{G} - (\mathbf{I}_{\ell^{i+1}} \otimes \mathbf{A})\mathbf{V}'_{i+1}. \end{aligned}$$

Thus, Eq. (3.5) holds and we can write

$$\begin{aligned} (\mathbf{f}^\intercal \otimes \mathbf{I}_n)\mathbf{W}_d \mathbf{C}^d &= (\mathbf{f}^\intercal \otimes \mathbf{I}_n)\big(\mathbf{x}^{\otimes d} \otimes \mathbf{G}\big) - (\mathbf{f}^\intercal \otimes \mathbf{I}_n)(\mathbf{I}_{\ell^d} \otimes \mathbf{A})\mathbf{V}'_d \\ &= \mathbf{f}^\intercal \mathbf{x}^{\otimes d} \cdot \mathbf{G} - (1 \otimes \mathbf{A})(\mathbf{f}^\intercal \otimes \mathbf{I}_m)\mathbf{V}'_d \\ &= \mathbf{f}^\intercal \mathbf{x}^{\otimes d} \cdot \mathbf{G} - \mathbf{A} \cdot (\mathbf{f}^\intercal \otimes \mathbf{I}_m)\mathbf{V}'_d \\ &= f(\mathbf{x}) \cdot \mathbf{G} - \mathbf{A}\mathbf{V}_f, \end{aligned}$$

by the mixed-product property (Eq. (2.1)). It suffices to bound $\|\mathbf{V}_f\|$. By Lemma 2.5 and Theorem 2.8, with all but negligible probability, $\|\mathbf{T}_{\mathsf{com}}\|, \|\mathbf{T}_{\mathsf{open}}\| \le \chi \cdot \sqrt{m}$. Thus, $\|\mathbf{C}\| \le \ell \|\mathbf{T}_{\mathsf{com}}\| \|\mathbf{x}\| \le \ell \chi B_{\mathsf{in}} \sqrt{m}$ and similarly, $\|\mathbf{V}\| \le \ell \|\mathbf{T}_{\mathsf{open}}\| \|\mathbf{x}\| \le \ell \chi B_{\mathsf{in}} \sqrt{m}$. We now show inductively that

$$\left\|\mathbf{V}'_i\right\| \le O\big((\ell \chi B_{\mathsf{in}})^i m^{(3i-2)/2}\big).$$

First, $\mathbf{V}'_1 = \mathbf{V}_1$ and the claim holds. For the inductive step, we have

$$\begin{aligned}\left\|\mathbf{V}'_{i+1}\right\| \le \|\mathbf{x}\| \|\mathbf{V}'_i\| \ell + \|\mathbf{V}_{i+1}\| \|\mathbf{C}\|^i m^i &\le B_{\mathsf{in}} \ell \cdot O\big((\ell \chi B_{\mathsf{in}})^i m^{(3i-2)/2}\big) + (\ell \chi B_{\mathsf{in}} \sqrt{m})^{i+1} m^i \\ &= O\big((\ell \chi B_{\mathsf{in}})^{i+1} m^{(3i+1)/2}\big).\end{aligned}$$

Since $d \le d_{\max}$, we can bound

$$\left\|\mathbf{V}_f\right\| \le \|\mathbf{f}\| \left\|\mathbf{V}'_d\right\| \ell^d \le \|\mathbf{f}\| \left\|\mathbf{V}'_{d_{\max}}\right\| \ell^{d_{\max}} \le O\big(B_{\mathsf{in}}^{d_{\max}+1} \ell^{2d_{\max}} \chi^{d_{\max}} m^{(3d_{\max}-2)/2}\big).$$

As long as $B \ge O\big(B_{\mathsf{in}}^{d_{\max}+1} \ell^{2d_{\max}} \chi^{d_{\max}} m^{(3d_{\max}-2)/2}\big)$, we see that Eq. (3.3) is satisfied. □

Theorem 3.6 (Computational Binding). *Suppose q is prime, $n \ge O(\lambda)$, $m \ge O(n \log q)$, $\chi_0 \ge \omega(\sqrt{\log m})$, $\chi \ge O(m^{3/2} L \chi_0 \log n)$, and $\beta \ge 2Bm^{3/2} \log n$. Then, under the L-succinct SIS assumption with parameters (n, m, q, χ_0, β) assumption, Construction 3.2 satisfies computational binding.*

Proof. We define an intermediate hybrid experiment:

- Hyb_0: This is the real computational binding experiment, where the challenger starts by sampling $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$ and gives $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}})$ to $\mathcal{A}$. Specifically, it samples $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$, $\mathbf{W} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{Ln \times m}$, and
$$\begin{bmatrix}\mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}}\end{bmatrix} \leftarrow \mathsf{SamplePre}([\mathbf{I}_L \otimes \mathbf{A} \mid \mathbf{W}], \mathbf{I}_L \otimes \mathbf{R}, \mathbf{P}, \chi),$$
where $\mathbf{P}$ is the target matrix in Eq. (3.1). Algorithm $\mathcal{A}$ outputs a commitment $\sigma = \mathbf{C} \in \mathbb{Z}_q^{m \times m}$, a function $f = \mathbf{f} \in \mathbb{Z}_q^{\ell^d}$ (for some $d \le d_{\max}$), distinct values $y_0, y_1 \in [-B_{\mathsf{out}}, B_{\mathsf{out}}]$, and openings $\mathbf{V}_0, \mathbf{V}_1 \in \mathbb{Z}_q^{m \times m}$. The output of the experiment is 1 if the following conditions hold:
  - $\|\mathbf{V}_0\|, \|\mathbf{V}_1\| \le B$; and
  - $(\mathbf{f}^\intercal \otimes \mathbf{I}_m)\mathbf{W}_d \mathbf{C}^d = y_0 \mathbf{G} - \mathbf{A}\mathbf{V}_0 = y_1 \mathbf{G} - \mathbf{A}\mathbf{V}_1$, where $\mathbf{W}_d \in \mathbb{Z}_q^{\ell^d n \times m}$ is derived from $\mathbf{W}$ according to Eq. (3.1).
- Hyb_1: Same as Hyb_0 except the challenger samples $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n \times m}$, $\mathbf{R} \leftarrow [\mathbf{I}_L \otimes \mathbf{A} \mid \mathbf{W}]_{\chi_0}^{-1}(\mathbf{G}_{nL})$ and
$$\begin{bmatrix}\mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}}\end{bmatrix} \leftarrow \mathsf{SamplePre}([\mathbf{I}_L \otimes \mathbf{A} \mid \mathbf{W}], \mathbf{R}, \mathbf{P}, \chi).$$

For an adversary $\mathcal{A}$, we write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output of an execution of Hyb_i with adversary $\mathcal{A}$. We now reason about the output distributions $\mathsf{Hyb}_0(\mathcal{A})$ and $\mathsf{Hyb}_1(\mathcal{A})$.

Lemma 3.7. *Suppose q is prime, $n \ge O(\lambda)$, $m \ge O(n \log q)$, $\chi_0 \ge \omega(\sqrt{\log m})$, and $\chi \ge O(m^{3/2} L \chi_0 \log n)$. Then, for all adversaries $\mathcal{A}$, $\mathsf{Hyb}_0(\mathcal{A}) \overset{s}{\approx} \mathsf{Hyb}_1(\mathcal{A})$.*

Proof. We introduce some additional intermediary distributions for $\mathbf{A}$, $\mathbf{T}_{\mathsf{open}}$, and $\mathbf{T}_{\mathsf{com}}$.

- Hyb_0: $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$ and $\left[\begin{smallmatrix}\mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}}\end{smallmatrix}\right] \leftarrow \mathsf{SamplePre}([\mathbf{I}_L \otimes \mathbf{A} \mid \mathbf{W}], \mathbf{I}_L \otimes \mathbf{R}, \mathbf{P}, \chi)$.
- $\mathsf{Hyb}_{0,1}$: $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$ and $\left[\begin{smallmatrix}\mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}}\end{smallmatrix}\right] \leftarrow [\mathbf{I}_L \otimes \mathbf{A} \mid \mathbf{W}]_{\chi}^{-1}(\mathbf{P})$.

  Indistinguishability: $\mathsf{Hyb}_0(\mathcal{A})$ and $\mathsf{Hyb}_{0,1}(\mathcal{A})$ are statistically indistinguishable when $n \ge O(\lambda)$, $m \ge O(n \log q)$ and $\chi \ge O(m \log n)$ by Theorem 2.8.

- $\mathsf{Hyb}_{0,2}$: $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}$ and $\begin{bmatrix}\mathbf{T}_{\mathsf{open}}\\ \mathbf{T}_{\mathsf{com}}\end{bmatrix} \leftarrow [\mathbf{I}_L \otimes \mathbf{A} \mid \mathbf{W}]_\chi^{-1}(\mathbf{P})$.

  Indistinguishability: $\mathsf{Hyb}_{0,1}(\mathcal{A})$ and $\mathsf{Hyb}_{0,2}(\mathcal{A})$ are statistically indistinguishable when $n \geq O(\lambda), m \geq O(n \log q)$ by Theorem 2.8.

- Hyb_1: $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}$, $\mathbf{R} \leftarrow [\mathbf{I}_L \otimes \mathbf{A} \mid \mathbf{W}]_{\chi_0}^{-1}(\mathbf{G}_{nL})$, and $\begin{bmatrix}\mathbf{T}_{\mathsf{open}}\\ \mathbf{T}_{\mathsf{com}}\end{bmatrix} \leftarrow \mathsf{SamplePre}([\mathbf{I}_L \otimes \mathbf{A} \mid \mathbf{W}], \mathbf{R}, \mathbf{P}, \chi)$.

  Indistinguishability: $\mathsf{Hyb}_{0,2}(\mathcal{A})$ and $\mathsf{Hyb}_1(\mathcal{A})$ are statistically indistinguishable when $n \geq O(\lambda)$, $\chi_0 \geq \omega(\sqrt{\log m})$ and $\chi \geq O(m^{3/2} L \chi_0 \log n)$. This follows by first applying Lemmas 2.5 and 2.7 to conclude that with overwhelming probability, $\|\mathbf{R}\| \leq \sqrt{m}\chi_0$. We can then invoke Theorem 2.8.

The claim then follows by a hybrid argument. □

Lemma 3.8. *Suppose q is prime and $\beta \geq 2Bm^{3/2} \log n$. Under the L-succinct SIS assumption with parameters (n, m, q, χ_0, β), for all efficient adversaries $\mathcal{A}$, $\Pr[\mathsf{Hyb}_1(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$.*

Proof. Suppose there exists an efficient adversary $\mathcal{A}$ where $\Pr[\mathsf{Hyb}_1(\mathcal{A}) = 1] = \varepsilon$ for some non-negligible ε. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ that breaks the L-succinct SIS assumption:

- At the beginning of the game, algorithm $\mathcal{B}$ receives $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{W} \in \mathbb{Z}_q^{nL\times m}$, and a trapdoor $\mathbf{R}$. Algorithm $\mathcal{B}$ forms the target matrix $\mathbf{P} \in \mathbb{Z}_q^{Ln\times \ell m}$ according to Eq. (3.1) and computes

$$\begin{bmatrix}\mathbf{T}_{\mathsf{open}}\\ \mathbf{T}_{\mathsf{com}}\end{bmatrix} \leftarrow \mathsf{SamplePre}([\mathbf{I}_L \otimes \mathbf{A} \mid \mathbf{W}], \mathbf{R}, \mathbf{P}, \chi).$$

  It gives $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}})$ to $\mathcal{A}$.

- Algorithm $\mathcal{A}$ outputs a commitment $\mathbf{C} \in \mathbb{Z}_q^{m\times m}$, a function $f = \mathbf{f} \in \mathbb{Z}_q^{\ell^d}$, distinct values $y_0, y_1 \in [-B_{\mathsf{out}}, B_{\mathsf{out}}]$, and openings $\mathbf{V}_0, \mathbf{V}_1 \in \mathbb{Z}_q^{m\times m}$.

- Algorithm $\mathcal{B}$ outputs $\mathbf{x} \leftarrow \mathsf{SamplePre}(\mathbf{A}, \mathbf{V}_0 - \mathbf{V}_1, \mathbf{0}, \chi')$ where $\chi' = 2Bm \log n$.

In the L-succinct SIS assumption, $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}$, $\mathbf{W} \xleftarrow{R} \mathbb{Z}_q^{nL\times m}$, and $\mathbf{R} \leftarrow [\mathbf{I}_L \otimes \mathbf{A} \mid \mathbf{W}]_{\chi_0}^{-1}(\mathbf{G}_{nL})$. Thus, algorithm $\mathcal{A}$ perfectly simulates the common reference string in Hyb_1. Thus, with probability at least ε, the output of algorithm $\mathcal{A}$ satisfies the following properties:

$$y_0 \neq y_1 \quad \text{and} \quad \|\mathbf{V}_0\|, \|\mathbf{V}_1\| \leq B \quad \text{and} \quad (\mathbf{f}^\intercal \otimes \mathbf{I}_m)\mathbf{W}_d\mathbf{C}^d = y_0\mathbf{G} - \mathbf{A}\mathbf{V}_0 = y_1\mathbf{G} - \mathbf{A}\mathbf{V}_1.$$

This means $\mathbf{A}(\mathbf{V}_0 - \mathbf{V}_1) = (y_0 - y_1)\mathbf{G}$. Since $y_0 \neq y_1$ and q is prime, $\mathbf{V}_0 - \mathbf{V}_1$ is a gadget trapdoor for $\mathbf{A}$. Since $\|\mathbf{V}_0 - \mathbf{V}_1\| \leq 2B$ and $\chi' \geq 2Bm \log n$, by Theorem 2.8, the distribution of $\mathbf{x}$ is statistically close to $\mathbf{A}_{\chi'}^{-1}(\mathbf{0})$, which is non-zero with overwhelming probability. Moreover, by Lemma 2.5, $\|\mathbf{x}\| \leq \sqrt{m}\chi'$ and the claim follows. □

The claim now follows by combining Lemmas 3.7 and 3.8. □

3.1 Opening to Multiple Outputs

In this section, we describe how to extend Construction 3.2 to obtain a functional commitment scheme that supports succinct openings to *multiple* outputs (i.e., the size of the opening scales sub-linearly with the number of functions we open to). Our approach follows the the approach from [WW23] for aggregating openings.

Construction 3.9 (Multi-Output Functional Commitment for Constant-Degree Polynomials)**.** Let λ be a security parameter. Let $n, m, q, \chi, \ell, d_{\max}, B_{\mathsf{in}}, B_{\mathsf{out}}, B$ be the same parameters as in Construction 3.2. Let $T = T(\lambda)$ be a bound on the number of outputs. Let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ be the set of functions $f\colon [-B_{\mathsf{in}}, B_{\mathsf{in}}]^\ell \to [-B_{\mathsf{out}}, B_{\mathsf{out}}]^T$, where each

function f can be described by a vector of homogeneous polynomials $(\mathbf{f}_1, \ldots, \mathbf{f}_T)$ with B_{in}-bounded coefficients and of the same degree $d \le d_{\max}$:[8]

$$f(\mathbf{x}) := \left(\mathbf{f}_1^\intercal \mathbf{x}^{\otimes d}, \ldots, \mathbf{f}_T^\intercal \mathbf{x}^{\otimes d}\right).$$

We construct a functional commitment $\Pi_{\mathsf{FC}} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Eval}, \mathsf{Verify})$ for $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ as follows:

- $\mathsf{Setup}(1^\lambda)$: Sample $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{W} \in \mathbb{Z}_q^{Ln \times m}$, $\mathbf{T}_{\mathsf{open}} \in \mathbb{Z}_q^{Lm \times \ell m}$, and $\mathbf{T}_{\mathsf{com}} \in \mathbb{Z}_q^{m \times \ell m}$ using the same procedure as Setup in Construction 3.2. Sample $\mathbf{D} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n \times T}$, and output the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{D})$.
- $\mathsf{Commit}(\mathsf{crs}, \mathbf{x})$: Same as in Construction 3.2.
- $\mathsf{Eval}(\mathsf{crs}, \mathsf{st}, f)$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{D})$, the state $\mathsf{st} = \mathbf{x}$, and a function $f = (\mathbf{f}_1, \ldots, \mathbf{f}_T)$ where each $\mathbf{f}_i \in \mathbb{Z}_q^{\ell^d}$ is B_{in}-bounded and $d \le d_{\max}$, the evaluation algorithm first computes an opening $\mathbf{V}_{\mathbf{f}_i} \in \mathbb{Z}_q^{m \times m}$ for $\mathbf{f}_i$ using the same procedure as in Construction 3.2. Then, it outputs the opening $\pi_f = \mathbf{v}_f$ where

  $$\mathbf{v}_f = \sum_{i \in [T]} \mathbf{V}_{\mathbf{f}_i} \mathbf{G}^{-1}(\mathbf{d}_i) \in \mathbb{Z}_q^m,$$

  and $\mathbf{d}_i \in \mathbb{Z}_q^n$ denotes the i^{th} column of $\mathbf{D}$.
- $\mathsf{Verify}(\mathsf{crs}, \sigma, f, \mathbf{y}, \pi)$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{D})$, the commitment $\sigma = \mathbf{C} \in \mathbb{Z}_q^{m \times m}$, the function $f = (\mathbf{f}_1, \ldots, \mathbf{f}_T)$ where each $\mathbf{f}_i \in \mathbb{Z}_q^{\ell^d}$ is B_{in}-bounded and $d \le d_{\max}$, the output $\mathbf{y} \in [-B_{\mathsf{out}}, B_{\mathsf{out}}]^T$, and the proof $\pi = \mathbf{v} \in \mathbb{Z}_q^m$, the verification algorithm parses $\mathbf{W}$ as in Eq. (3.1) and outputs 1 if

  $$\|\mathbf{v}\| \le B \quad \text{and} \quad \sum_{i \in [T]} (\mathbf{f}_i^\intercal \otimes \mathbf{I}_m) \mathbf{W}_d \mathbf{C}^d \mathbf{G}^{-1}(\mathbf{d}_i) = \mathbf{D}\mathbf{y} - \mathbf{A}\mathbf{v}, \tag{3.7}$$

  where $\mathbf{d}_i \in \mathbb{Z}_q^n$ is the i^{th} column of $\mathbf{D}$.

Remark 3.10 (Supporting Preprocessing)**.** Like Construction 3.2, Construction 3.9 supports full preprocessing (Definition 2.2) and function-only preprocessing (Remark 2.3). Here, we describe the approach for full preprocessing.

- $\mathsf{Preprocess}(\mathsf{crs}, f, \mathbf{y})$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{D})$, the function $f = (\mathbf{f}_1, \ldots, \mathbf{f}_T)$ where each $\mathbf{f}_i \in \mathbb{Z}_q^{\ell^d}$ is B_{in}-bounded and $d \le d_{\max}$, and the output $\mathbf{y} \in [-B_{\mathsf{out}}, B_{\mathsf{out}}]^T$, the preprocessing algorithm computes

  $$\mathbf{F} = \sum_{i \in [T]} \left(\left(\mathbf{G}^{-1}(\mathbf{d}_i)\right)^\intercal \otimes (\mathbf{f}_i^\intercal \otimes \mathbf{I}_m) \mathbf{W}_d \right) \in \mathbb{Z}_q^{n \times m^2} \tag{3.8}$$

  $$\mathbf{y}^* = \mathbf{D}\mathbf{y} \in \mathbb{Z}_q^n, \tag{3.9}$$

  and outputs the verification key $\mathsf{vk}_{f,\mathbf{y}} = (\mathbf{F}, \mathbf{y}^*)$.
- $\mathsf{OnlineVerify}(\mathsf{vk}, \sigma, \pi)$: On input the verification key $\mathsf{vk} = (\mathbf{F}, \mathbf{y}^*)$, the commitment $\sigma = \mathbf{C} \in \mathbb{Z}_q^{m \times m}$, and the opening $\pi = \mathbf{v} \in \mathbb{Z}_q^m$, the online verification algorithm outputs 1 if

  $$\|\mathbf{v}\| \le B \quad \text{and} \quad \mathbf{F} \cdot \mathrm{vec}(\mathbf{C}^d) = \mathbf{y}^* - \mathbf{A}\mathbf{v}.$$

To show that this is correct, we apply vectorization to the main verification relation in Eq. (3.7):

$$\mathrm{vec}\left(\sum_{i \in [T]} (\mathbf{f}_i^\intercal \otimes \mathbf{I}_m) \mathbf{W}_d \mathbf{C}^d \mathbf{G}^{-1}(\mathbf{d}_i) \right) = \underbrace{\sum_{i \in [T]} \left(\left(\mathbf{G}^{-1}(\mathbf{d}_i)\right)^\intercal \otimes (\mathbf{f}_i^\intercal \otimes \mathbf{I}_m) \mathbf{W}_d \right)}_{\mathbf{F}} \mathrm{vec}(\mathbf{C}^d).$$

[8] Our construction also supports the setting where $\mathbf{f}_1, \ldots, \mathbf{f}_T$ have different degrees $d_1, \ldots, d_T \le d_{\max}$. For simplicity of exposition, we just describe the case where they have equal degree $d \le d_{\max}$.

Then, the main verification relation in Eq. (3.7) becomes

$$\mathbf{F} \cdot \mathrm{vec}(\mathbf{C}^d) = \mathrm{vec}(\mathbf{D}\mathbf{y} - \mathbf{A}\mathbf{v}) = \mathbf{D}\mathbf{y} - \mathbf{A}\mathbf{v} = \mathbf{y}^* - \mathbf{A}\mathbf{v},$$

and correctness reduces to that of Construction 3.9. By construction, $|\mathsf{vk}_{f,\mathbf{y}}| = (nm^2 + n)\log q$ and the running time of OnlineVerify is $\mathrm{poly}(n, m, d_{\max}, \log q)$. As we show below, we can instantiate our scheme so that $n, m, \log q = \mathrm{poly}(\lambda, \log \ell, \log T)$, and so the construction satisfies the required efficiency properties. Finally, the above analysis also applies to function-only preprocessing: namely, the preprocessed function key for a function $f = (\mathbf{f}_1, \ldots, \mathbf{f}_T)$ is the matrix $\mathbf{F}$ from Eq. (3.8). In this case, the running time of verification becomes $\mathrm{poly}(n, m, \log q, T)$.

Theorem 3.11 (Correctness). *Suppose $n \geq \lambda$, $m \geq O(n \log q)$, $\chi \geq O(m \log n)$, and $B \geq O\big(TB_{\mathsf{in}}^{d_{\max}+1} \ell^{2d_{\max}} \chi^{d_{\max}} m^{3d_{\max}}\big)$. Then, Construction 3.9 is correct.*

Proof. Take any input $\mathbf{x} \in [-B_{\mathsf{in}}, B_{\mathsf{in}}]^\ell$ and function $f = (\mathbf{f}_1, \ldots, \mathbf{f}_T)$ where $\mathbf{f}_i \in \mathbb{Z}_q^{\ell^d}$ for some $d \leq d_{\max}$. Let $\mathbf{y} = f(\mathbf{x}) = (\mathbf{f}_1^\intercal \mathbf{x}^{\otimes d}, \ldots, \mathbf{f}_T^\intercal \mathbf{x}^{\otimes d})$. Let $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{D}) \leftarrow \mathsf{Setup}(1^\lambda)$, $(\mathbf{C}, \mathsf{st}) \leftarrow \mathsf{Commit}(\mathsf{crs}, \mathbf{x})$, and $\pi_f = \mathbf{v}_f \leftarrow \mathsf{Eval}(\mathsf{crs}, \mathsf{st}, f)$. Then, $\mathbf{C} = \mathbf{T}_{\mathsf{com}}(\mathbf{x} \otimes \mathbf{I}_m)$. For $i \in [T]$, let $\mathbf{V}_{\mathbf{f}_i} \in \mathbb{Z}_q^{m \times m}$ be the matrices computed by $\mathsf{Eval}(\mathsf{crs}, \mathsf{st}, f)$. By the same analysis as in the proof of Theorem 3.5, we have that for all $i \in [T]$,

$$(\mathbf{f}_i^\intercal \otimes \mathbf{I}_n)\mathbf{W}_d \mathbf{C}^d = \mathbf{f}_i^\intercal \mathbf{x}^{\otimes d} \cdot \mathbf{G} - \mathbf{A}\mathbf{V}_{\mathbf{f}_i} = y_i \mathbf{G} - \mathbf{A}\mathbf{V}_{\mathbf{f}_i},$$

and $\|\mathbf{V}_{\mathbf{f}_i}\| \leq O\big(B_{\mathsf{in}}^{d+1} \ell^{2d} \chi^d m^{(3d-2)/2}\big)$. Then,

$$\sum_{i \in [T]} (\mathbf{f}_i^\intercal \otimes \mathbf{I}_m)\mathbf{W}_d \mathbf{C}^d \mathbf{G}^{-1}(\mathbf{d}_i) = \sum_{i \in [T]} y_i \mathbf{d}_i - \mathbf{A} \cdot \sum_{i \in [T]} \mathbf{V}_{\mathbf{f}_i} \mathbf{G}^{-1}(\mathbf{p}_i) = \mathbf{D}\mathbf{y} - \mathbf{A}\mathbf{v}_f.$$

Moreover,

$$\|\mathbf{v}_f\| \leq \sum_{i \in [T]} \|\mathbf{V}_{\mathbf{f}_i}\| m \leq O\big(TB_{\mathsf{in}}^{d+1} \ell^{2d} \chi^d m^{3d}\big) \leq B$$

since $d \leq d_{\max}$. □

Theorem 3.12 (Computational Binding). *Suppose q is prime, $n \geq O(\lambda)$, $m \geq O(n \log q)$, $\chi_0 \geq \omega(\sqrt{\log m})$, $\chi \geq O(m^{3/2} L \chi_0 \log n)$, and $\beta \geq 2B + 2TB_{\mathsf{out}}$. Then, under the L-succinct SIS assumption with parameters (n, m, q, χ_0, β) assumption, Construction 3.9 satisfies computational binding.*

Proof. We use a similar sequence of hybrids as in the proof of Theorem 3.6:

- Hyb_0: This is the real computational binding experiment, where the challenger starts by sampling $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$ and gives $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{D})$ to $\mathcal{A}$. Specifically, it samples $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$, $\mathbf{W} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{Ln \times m}$, $\mathbf{D} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n \times T}$, and

  $$\begin{bmatrix} \mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix} \leftarrow \mathsf{SamplePre}([\mathbf{I}_L \otimes \mathbf{A} \mid \mathbf{W}], \mathbf{I}_L \otimes \mathbf{R}, \mathbf{P}, \chi),$$

  where $\mathbf{P}$ is the target matrix in Eq. (3.1). Algorithm $\mathcal{A}$ outputs a commitment $\sigma = \mathbf{C} \in \mathbb{Z}_q^{m \times m}$, a function $f = (\mathbf{f}_1, \ldots, \mathbf{f}_T)$ where each $\mathbf{f}_i \in \mathbb{Z}_q^{\ell^d}$ (for some $d \leq d_{\max}$), values $\mathbf{y}_0, \mathbf{y}_1 \in [-B_{\mathsf{out}}, B_{\mathsf{out}}]^T$, and openings $\mathbf{v}_0, \mathbf{v}_1 \in \mathbb{Z}_q^m$. The output of the experiment is 1 if the following conditions hold:

  - $\|\mathbf{v}_0\|, \|\mathbf{v}_1\| \leq B$; and
  - $\sum_{i \in [T]} (\mathbf{f}_i^\intercal \otimes \mathbf{I}_m)\mathbf{W}_d \mathbf{C}^d \mathbf{G}^{-1}(\mathbf{d}_i) = \mathbf{D}\mathbf{y}_0 - \mathbf{A}\mathbf{v}_0 = \mathbf{D}\mathbf{y}_1 - \mathbf{A}\mathbf{v}_1$, where $\mathbf{d}_i \in \mathbb{Z}_q^n$ is the i^{th} column of $\mathbf{D}$ and $\mathbf{W}_d \in \mathbb{Z}_q^{\ell^d n \times m}$ is the d^{th} block of $\mathbf{W}$ (see Eq. (3.1)).

- Hyb_1: Same as Hyb_0 except the challenger samples $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n \times m}$, $\mathbf{R} \leftarrow [\mathbf{I}_L \otimes \mathbf{A} \mid \mathbf{W}]_{\chi_0}^{-1}(\mathbf{G}_{nL})$ and

  $$\begin{bmatrix} \mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix} \leftarrow \mathsf{SamplePre}([\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}], \mathbf{R}, \mathbf{P}, \chi).$$

- Hyb_2: Same as Hyb_1 except the challenger samples $\mathbf{R}' \xleftarrow{R} \{0,1\}^{m \times T}$ and sets $\mathbf{D} \leftarrow \mathbf{A}\mathbf{R}'$.

For an adversary $\mathcal{A}$, we write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output of an execution of Hyb_i with adversary $\mathcal{A}$.

Lemma 3.13. *Suppose q is prime, $n \geq O(\lambda)$, $m \geq O(n \log q)$, $\chi_0 \geq \omega(\sqrt{\log m})$, and $\chi \geq O(m^{3/2} L \chi_0 \log n)$. Then, for all adversaries $\mathcal{A}$, $\mathsf{Hyb}_0(\mathcal{A}) \overset{s}{\approx} \mathsf{Hyb}_1(\mathcal{A})$.*

Proof. Follows by the same argument as the proof of Lemma 3.8 □

Lemma 3.14. *Suppose $m > 2\lambda + n \log q$. Then, for all adversaries $\mathcal{A}$, $\mathsf{Hyb}_1 \overset{s}{\approx} \mathsf{Hyb}_2(\mathcal{A})$.*

Proof. Follows by the leftover hash lemma (Lemma 2.4) and a standard hybrid argument. □

Lemma 3.15. *Suppose $m \geq 2\lambda + (\lambda + n) \log q$ and $\beta \geq 2B + 2TB_{\mathsf{out}}$. Under the L-succinct SIS assumption with parameters (n, m, q, χ_0, β), for all efficient adversaries $\mathcal{A}$, $\Pr[\mathsf{Hyb}_2(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$.*

Proof. Suppose there exists an efficient adversary $\mathcal{A}$ such that $\Pr[\mathsf{Hyb}_1(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ that breaks the L-succinct SIS assumption:

- At the beginning of the game, algorithm $\mathcal{B}$ receives $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{W} \in \mathbb{Z}_q^{nL \times m}$, and a trapdoor $\mathbf{R}$. Algorithm $\mathcal{B}$ forms the target matrix $\mathbf{P} \in \mathbb{Z}_q^{Ln \times \ell m}$ according to Eq. (3.1) and computes

$$\begin{bmatrix} \mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix} \leftarrow \mathsf{SamplePre}([\mathbf{I}_L \otimes \mathbf{A} \mid \mathbf{W}], \mathbf{R}, \mathbf{P}, \chi).$$

  It then samples $\mathbf{R}' \xleftarrow{R} \{0,1\}^{m \times T}$ and sets $\mathbf{D} \leftarrow \mathbf{A}\mathbf{R}'$. It gives $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{D})$ to $\mathcal{A}$.

- Algorithm $\mathcal{A}$ outputs a commitment $\mathbf{C} \in \mathbb{Z}_q^{m \times m}$, a function $f = (\mathbf{f}_1, \ldots, \mathbf{f}_T)$ where each $\mathbf{f}_i \in \mathbb{Z}_q^{\ell d}$, distinct values $\mathbf{y}_0, \mathbf{y}_1 \in [-B_{\mathsf{out}}, B_{\mathsf{out}}]^T$, and openings $\mathbf{v}_0, \mathbf{v}_1 \in \mathbb{Z}_q^m$.
- Algorithm $\mathcal{B}$ outputs $\mathbf{x} = \mathbf{v}_0 - \mathbf{v}_1 + \mathbf{R}'(\mathbf{y}_1 - \mathbf{y}_0)$.

In the L-succinct SIS assumption, $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, $\mathbf{W} \xleftarrow{R} \mathbb{Z}_q^{nL \times m}$, and $\mathbf{R} \leftarrow [\mathbf{I}_L \otimes \mathbf{A} \mid \mathbf{W}]_{\chi_0}^{-1}(\mathbf{G}_{nL})$. Thus, algorithm $\mathcal{A}$ perfectly simulates the common reference string in Hyb_2. Thus, with probability at least ε, the output of algorithm $\mathcal{A}$ satisfies the following properties:

$$\mathbf{y}_0 \neq \mathbf{y}_1 \quad \text{and} \quad \|\mathbf{v}_0\|, \|\mathbf{v}_1\| \leq B \quad \text{and} \quad \mathbf{D}\mathbf{y}_0 - \mathbf{A}\mathbf{v}_0 = \mathbf{D}\mathbf{y}_1 - \mathbf{A}\mathbf{v}_1.$$

We can re write this as

$$\mathbf{0} = \mathbf{A}(\mathbf{v}_0 - \mathbf{v}_1) + \mathbf{D}(\mathbf{y}_1 - \mathbf{y}_0) = \mathbf{A}(\mathbf{v}_0 - \mathbf{v}_1 + \mathbf{R}'(\mathbf{y}_1 - \mathbf{y}_0)) = \mathbf{A}\mathbf{x}.$$

By construction, $\|\mathbf{x}\| \leq 2B + 2TB_{\mathsf{out}} \leq \beta$, so it suffices to argue that $\mathbf{x} \neq \mathbf{0}$, or equivalently, that $\mathbf{R}'(\mathbf{y}_0 - \mathbf{y}_1) \neq \mathbf{v}_0 - \mathbf{v}_1$. We appeal to an entropy argument and the generalized leftover hash lemma. By construction, $\mathbf{v}_0, \mathbf{v}_1, \mathbf{y}_0, \mathbf{y}_1$ are functions of $\mathbf{D} \in \mathbb{Z}_q^{n \times T}$ (and other quantities that are independent of $\mathbf{R}'$). Moreover, $\mathbf{D}$ contains at most $nT \log q$ bits of information about $\mathbf{R}'$. This means that

$$\mathbf{H}_\infty(\mathbf{R}' \mid \mathbf{v}_0, \mathbf{v}_1, \mathbf{y}_0, \mathbf{y}_1) \geq \mathbf{H}_\infty(\mathbf{R}' \mid \mathbf{D}) \geq mT - nT \log q \geq (2\lambda + \lambda \log q)T.$$

By the (generalized) leftover hash lemma (Lemma 2.4), the following distributions are statistically indistinguishable:

$$D_0 \coloneqq \left\{ (\mathbf{C}, \mathbf{C}\mathbf{R}', (\mathbf{v}_0, \mathbf{v}_1, \mathbf{y}_0, \mathbf{y}_1)) : \begin{array}{c} \mathbf{C} \xleftarrow{R} \mathbb{Z}_q^{\lambda \times m} \\ \mathbf{R}' \xleftarrow{R} \{0,1\}^{m \times T} \end{array} \right\} \text{ and } D_1 \coloneqq \left\{ (\mathbf{C}, \mathbf{Z}, (\mathbf{v}_0, \mathbf{v}_1, \mathbf{y}_0, \mathbf{y}_1)) : \begin{array}{c} \mathbf{C} \xleftarrow{R} \mathbb{Z}_q^{\lambda \times m} \\ \mathbf{Z} \xleftarrow{R} \mathbb{Z}_q^{\lambda \times T} \end{array} \right\}. \tag{3.10}$$

Since $\mathbf{y}_0 - \mathbf{y}_1 \neq \mathbf{0}$, in distribution D_1, with overwhelming probability over the choice of $\mathbf{C} \xleftarrow{R} \mathbb{Z}_q^{\lambda \times m}$ and $\mathbf{Z} \xleftarrow{R} \mathbb{Z}_q^{\lambda \times t}$, we have that

$$[\mathbf{C} \mid \mathbf{Z}] \begin{bmatrix} \mathbf{v}_1 - \mathbf{v}_0 \\ \mathbf{y}_0 - \mathbf{y}_1 \end{bmatrix} \neq \mathbf{0}.$$

Since the distributions D_0 and D_1 in Eq. (3.10) are statistically close, this means that with overwhelming probability,

$$\mathbf{0} \neq [\mathbf{C} \mid \mathbf{C}\mathbf{R}'] \begin{bmatrix} \mathbf{v}_1 - \mathbf{v}_0 \\ \mathbf{y}_0 - \mathbf{y}_1 \end{bmatrix} = \mathbf{C}[\mathbf{I}_m \mid \mathbf{R}'] \begin{bmatrix} \mathbf{v}_1 - \mathbf{v}_0 \\ \mathbf{y}_0 - \mathbf{y}_1 \end{bmatrix}.$$

This correspondingly means that with overwhelming probability

$$[\mathbf{I}_m \mid \mathbf{R}'] \begin{bmatrix} \mathbf{v}_1 - \mathbf{v}_0 \\ \mathbf{y}_0 - \mathbf{y}_1 \end{bmatrix} \neq \mathbf{0},$$

or equivalently, that $\mathbf{v}_0 - \mathbf{v}_1 \neq \mathbf{R}'(\mathbf{y}_0 - \mathbf{y}_1)$, as required. We conclude that algorithm $\mathcal{A}$ succeeds with probability $\varepsilon - \mathsf{negl}(\lambda)$ and the claim follows. □

The claim now follows by combining Lemmas 3.13 to 3.15. □

Parameter instantiation. Let λ be a security parameter, ℓ be the input dimension, T be the output dimension, and $d_{\max} = O(1)$ be a degree bound. Let B_{in} be a bound on the input magnitude and B_{out} be a bound on the output magnitude. We instantiate the lattice parameters in Construction 3.9 to satisfy Theorems 3.11 and 3.12 as follows:

- We set the lattice dimension $n = \lambda$ and $m = O(n \log q)$.
- We set $\chi_0 = \mathsf{poly}(\lambda, m, L)$ and $\chi \geq O(m^{3/2} L \chi_0 \log n) = \mathsf{poly}(\lambda, m, L)$. By definition, $L = O(\ell^{d_{\max}})$.
- We set the bound $B = O(T B_{\mathsf{in}}^{d_{\max}+1} \ell^{2d_{\max}} \chi^{d_{\max}} m^{3d_{\max}})$.
- Let $\beta = 2B + 2TB_{\mathsf{out}} = O\big(T(B_{\mathsf{in}}^{d_{\max}+1} \ell^{2d_{\max}} \chi^{d_{\max}} m^{3d_{\max}} + B_{\mathsf{out}})\big)$. We choose the modulus $q = \beta \cdot \mathsf{poly}(n)$ so that the L-succinct SIS assumption with parameters (n, m, q, χ_0, β) holds. In this case,

  $$\log q = \mathsf{poly}(d_{\max}, \log \lambda, \log \ell, \log T, \log B_{\mathsf{in}}, \log B_{\mathsf{out}}).$$

  When $B_{\mathsf{in}}, B_{\mathsf{out}} = \mathsf{poly}(\lambda)$ and $d_{\max} = O(1)$, the noise bound β and the modulus q are both $\mathsf{poly}(n)$.

For simplicity of exposition, we consider the case where the input magnitude B_{in} and output magnitudes B_{out} are both $\mathsf{poly}(\lambda)$. Then, $\log q = \mathsf{poly}(d_{\max}, \log \lambda, \log \ell, \log T)$. With this setting of parameters, we obtain a functional commitment scheme for constant-degree polynomials of degree up to $d_{\max}$ with the following parameter sizes:

- **Commitment size:** A commitment σ to an input $\mathbf{x} \in \{0,1\}^\ell$ consists of a matrix $\sigma = \mathbf{C} \in \mathbb{Z}_q^{m \times m}$, so

  $$|\sigma| = m^2 \log q = \mathsf{poly}(n, \log q) = \mathsf{poly}(\lambda, d_{\max}, \log \ell, \log T).$$

- **Opening size:** An opening π to a function f consists of a vector $\pi = \mathbf{v}_f \in \mathbb{Z}_q^m$ so

  $$|\pi| = m \log q = \mathsf{poly}(\lambda, d_{\max}, \log \ell, \log T).$$

- **CRS size:** The CRS consists of $(\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{D})$ where $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{W} \in \mathbb{Z}_q^{Ln \times m}$, $\mathbf{T}_{\mathsf{open}} \in \mathbb{Z}_q^{Lm \times \ell m}$, and $\mathbf{T}_{\mathsf{com}} \in \mathbb{Z}_q^{m \times \ell m}$. Thus the total size of the CRS is

  $$|\mathsf{crs}| = O(L \ell m^2) \cdot \log q = \ell^{d_{\max}+1} \cdot \mathsf{poly}(\lambda, d_{\max}, \log \ell, \log T).$$

We summarize the instantiation in the following corollary:

Corollary 3.16 (Succinct Functional Commitment for Constant-Degree Polynomials)**.** *Let λ be a security parameter, and let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of functions $f\colon [-B_{\mathsf{in}}, B_{\mathsf{in}}]^\ell \to [-B_{\mathsf{out}}, B_{\mathsf{out}}]^T$ on inputs of length $\ell = \ell(\lambda)$ and magnitude $B_{\mathsf{in}} = \mathsf{poly}(\lambda)$, and outputs of length $T = T(\lambda)$ and magnitude $B_{\mathsf{out}} = \mathsf{poly}(\lambda)$, and where each function f can be described by a vector of T homogeneous polynomials with B_{in}-bounded coefficients and degree $d \le d_{\max} = O(1)$. Then, under the L-succinct SIS assumption (with $L = O(\ell^{d_{\max}})$) and a polynomial norm bound, there exists a succinct functional commitment for $\mathcal{F}$. The commitment and opening have size $\mathsf{poly}(\lambda, d_{\max}, \log\ell, \log T)$ and the CRS has size $\ell^{d_{\max}+1}\cdot\mathsf{poly}(\lambda, d_{\max}, \log\ell, \log T)$. The functional commitment supports full preprocessing (Definition 2.2) and function-only preprocessing (Remark 2.3). With full preprocessing, the running time of the online verification algorithm is $\mathsf{poly}(\lambda, d_{\max}, \log\ell, \log T)$.*

Remark 3.17 (Shorter Commitment and Openings)**.** We can reduce the commitment size to $O(n^2\log q)$ and the opening size to $O(n\log q)$ in the above construction by using a gadget matrix with a larger decomposition base (specifically, instead of considering a binary decomposition, we consider a z-ary gadget matrix where $z = q^{1/c}$ for a large constant $c \in \mathbb{N}$). This coincides with the approach taken in [ACL+22]. In addition, we can further reduce the size of the commitment by using module lattices instead of integer lattices. We provide the details on extending to modules and using a z-ary gadget decomposition in Appendix A.

3.2 A Dual Functional Commitment for Committing to Functions

In this section, we construct a functional commitment that supports committing to a *function* $f\colon \{0,1\}^\ell \to \{0,1\}$ and then opening the commitment at a particular input $\mathbf{x} \in \{0,1\}^\ell$. This is a dual notion of Definition 2.1, where the Commit algorithm takes as input the function f and the Eval algorithm takes as input an input vector $\mathbf{x}$. We often refer to this variant of functional commitment as a "dual functional commitment."

Here, we consider a construction for general Boolean functions f on inputs of length $\ell = \ell(\lambda)$ and computable by Boolean circuits with bounded depth $d = d(\lambda)$. Similar to [dCP23, WW23], we allow the length of the commitment and the openings to scale with $\mathsf{poly}(\lambda, d, \log\ell)$. We can view our construction as a hybrid of the dual functional commitment from [dCP23] and the attribute-based encryption (ABE) scheme from [Wee23].

Like the construction of [dCP23], our functional commitment scheme satisfies a weaker notion of binding called "selective-input security" where the adversary is required to first *commit* to the point $\mathbf{x} \in \{0,1\}^\ell$ to which it will construct an opening. The adversary has to commit to this input *before* seeing the public parameters. The security reduction will then program $\mathbf{x}$ into the public parameters itself. This limitation to a selective notion of security is common to many related lattice-based primitives such as attribute-based encryption [GVW13, BGG+14, GVW15a, Wee23] and constrained PRFs [BV15, BTVW17]. We now give the formal definition of selective-input binding and then show how to use the ℓ-succinct SIS assumption to construct a succinct dual functional commitment for Boolean circuits with succinct commitments, openings, and fast verification (in the preprocessing model).

Definition 3.18 (Selective-Input Binding Security)**.** Let λ be a security parameter, and let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of efficiently-computable functions $f\colon \mathcal{X}^\ell \to \mathcal{Y}$. Let $\Pi_{\mathsf{FC}} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Eval}, \mathsf{Verify})$ be a (dual) functional commitment scheme for $\mathcal{F}$. We now define the selective-input binding game between an adversary $\mathcal{A}$ and a challenger:

1. At the beginning of the game, the adversary chooses an input $\mathbf{x} \in \mathcal{X}^\ell$ and sends $\mathbf{x}$ to the challenger.
2. The challenger samples $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$ and gives crs to $\mathcal{A}$.
3. The adversary outputs a commitment σ, values $y_0, y_1 \in \mathcal{Y}$, and openings π_0, π_1.
4. The output of the experiment is $b = 1$ if $y_0 \neq y_1$ and $\mathsf{Verify}(\mathsf{crs}, \sigma, \mathbf{x}, y_0, \pi_0) = 1 = \mathsf{Verify}(\mathsf{crs}, \sigma, \mathbf{x}, y_1, \pi_1)$. Otherwise, the output of the experiment is $b = 0$.

The functional commitment scheme satisfies computational selective-input binding if for all efficient adversaries $\mathcal{A}$, $\Pr[b = 1] = \mathsf{negl}(\lambda)$ in the above security game.

Construction 3.19 (Dual Functional Commitment for Boolean Circuits)**.** Let λ be a security parameter and $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$, and $\chi = \chi(\lambda)$ be lattice parameters. Let $\ell = \ell(\lambda)$ be an input length parameter, and $B = B(\lambda)$ be a bound. Let $\mathcal{F}_\lambda$ be a collection of functions $f\colon \{0,1\}^\ell \to \{0,1\}$ that can be computed by a Boolean circuit of depth at most $d = d(\lambda)$. We construct a dual functional commitment $\Pi_{\mathsf{FC}} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Eval}, \mathsf{Verify})$ for $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ as follows:

- $\mathsf{Setup}(1^\lambda)$: On input the security parameter λ, the setup algorithm samples $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$ and $\mathbf{W} \xleftarrow{R} \mathbb{Z}_q^{\ell n \times m}$. Sample $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}], \mathbf{I}_\ell \otimes \mathbf{R}, \mathbf{G}_{n\ell}, \chi) \in \mathbb{Z}_q^{(\ell m + m) \times \ell m}$. Parse $\mathbf{T} = \begin{bmatrix} \mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix}$ where $\mathbf{T}_{\mathsf{open}} \in \mathbb{Z}_q^{\ell m \times \ell m}$ and $\mathbf{T}_{\mathsf{com}} \in \mathbb{Z}_q^{m \times \ell m}$. Finally, it samples $\mathbf{W}_0 \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, computes $\mathbf{B} = -\mathbf{W}_0 \mathbf{T}_{\mathsf{com}} \in \mathbb{Z}_q^{n \times \ell m}$ and outputs the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{W}_0, \mathbf{B})$.
- $\mathsf{Commit}(\mathsf{crs}, f)$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{W}_0, \mathbf{B})$ and a function $f\colon \{0,1\}^\ell \to \{0,1\}$, the commit algorithm computes $\mathbf{B}_f \leftarrow \mathsf{EvalF}(\mathbf{B}, f)$ and outputs the commitment $\sigma = \mathbf{B}_f \in \mathbb{Z}_q^{n \times m}$ along with the state $\mathsf{st} = f$.
- $\mathsf{Eval}(\mathsf{crs}, \mathsf{st}, \mathbf{x})$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{W}_0, \mathbf{B})$, the state $\mathsf{st} = f$, and the input $\mathbf{x} \in \{0,1\}^\ell$, the evaluation algorithm computes $\mathbf{H}_{\mathbf{B},f,\mathbf{x}} \leftarrow \mathsf{EvalFX}(\mathbf{B}, f, \mathbf{x}) \in \mathbb{Z}_q^{\ell m \times m}$ and outputs

$$\pi = \mathbf{V} = \begin{bmatrix} -(\mathbf{x}^\intercal \otimes \mathbf{I}_m)\mathbf{T}_{\mathsf{open}} \\ -\mathbf{T}_{\mathsf{com}} \end{bmatrix} \cdot \mathbf{H}_{\mathbf{B},f,\mathbf{x}} \in \mathbb{Z}_q^{2m \times m}. \tag{3.11}$$

- $\mathsf{Verify}(\mathsf{crs}, \sigma, \mathbf{x}, y, \pi)$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{W}_0, \mathbf{B})$, a commitment $\sigma = \mathbf{B}_f \in \mathbb{Z}_q^{n \times m}$, an input $\mathbf{x} \in \{0,1\}^\ell$, an output $y \in \{0,1\}$, and an opening $\pi = \mathbf{V} \in \mathbb{Z}_q^{2m \times m}$, the verification algorithm outputs 1 if

$$\|\mathbf{V}\| \le B \quad \text{and} \quad \mathbf{B}_f - y\mathbf{G} = [\mathbf{A} \mid \mathbf{W}_0 + (\mathbf{x}^\intercal \otimes \mathbf{I}_n)\mathbf{W}]\mathbf{V}. \tag{3.12}$$

Remark 3.20 (Supporting Preprocessing)**.** Similar to Constructions 3.2 and 3.9, Construction 3.19 also supports fast verification in the preprocessing model. Note that in the dual setting, we preprocess with respect to an *input* $\mathbf{x}$ rather than a function f.

- $\mathsf{Preprocess}(\mathsf{crs}, \mathbf{x})$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{W}_0, \mathbf{B})$ and the input $\mathbf{x} \in \{0,1\}^\ell$, the preprocess algorithm outputs $\mathsf{vk}_{\mathbf{x}} = \mathbf{F}_{\mathbf{x}} = [\mathbf{A} \mid \mathbf{W}_0 + (\mathbf{x}^\intercal \otimes \mathbf{I}_n)\mathbf{W}] \in \mathbb{Z}_q^{n \times 2m}$.
- $\mathsf{OnlineVerify}(\mathsf{vk}, \sigma, y, \pi)$: On input the verification key $\mathsf{vk} = \mathbf{F}_{\mathbf{x}} \in \mathbb{Z}_q^{n \times 2m}$, the commitment $\sigma = \mathbf{B}_f \in \mathbb{Z}_q^{n \times 2m}$, a value $y \in \{0,1\}$, and an opening $\pi = \mathbf{V} \in \mathbb{Z}_q^{2m \times m}$, the online verification algorithm outputs 1 if

$$\|\mathbf{V}\| \le B \quad \text{and} \quad \mathbf{B}_f - y\mathbf{G} = \mathbf{F}_{\mathbf{x}}\mathbf{V}.$$

Theorem 3.21 (Correctness)**.** *Suppose $n \ge O(\lambda)$, $m \ge O(n \log q)$, and $B \ge \ell m^{5/2} \chi (n \log q)^{O(d)}$. Then, Construction 3.19 is correct.*

Proof. Take any function $f\colon \{0,1\}^\ell \to \{0,1\}$ which can be computed by a function of depth at most d and any input $\mathbf{x} \in \{0,1\}^\ell$. Let $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{W}_0, \mathbf{B}) \leftarrow \mathsf{Setup}(1^\lambda)$, $(\mathbf{B}_f, \mathsf{st}) \leftarrow \mathsf{Commit}(\mathsf{crs}, f)$, and $\pi = \mathbf{V} \leftarrow \mathsf{Eval}(\mathsf{crs}, \mathsf{st}, \mathbf{x})$. By construction, this means $[\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}]\mathbf{T} = \mathbf{G}$, $\mathbf{B} = -\mathbf{W}_0 \mathbf{T}_{\mathsf{com}}$, $\mathbf{B}_f = \mathsf{EvalF}(\mathbf{B}, f)$, and $\mathbf{V}$ satisfies Eq. (3.11). The key equation is

$$\begin{aligned} [\mathbf{A} \mid (\mathbf{x}^\intercal \otimes \mathbf{I}_n)\mathbf{W}] \begin{bmatrix} (\mathbf{x} \otimes \mathbf{I}_m)\mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix} &= \mathbf{A}(\mathbf{x}^\intercal \otimes \mathbf{I}_m)\mathbf{T}_{\mathsf{open}} + (\mathbf{x}^\intercal \otimes \mathbf{I}_n)\mathbf{W}\mathbf{T}_{\mathsf{com}} \\ &= (\mathbf{x}^\intercal \otimes \mathbf{I}_n)(\mathbf{I}_\ell \otimes \mathbf{A})\mathbf{T}_{\mathsf{open}} + (\mathbf{x}^\intercal \otimes \mathbf{I}_n)\mathbf{W}\mathbf{T}_{\mathsf{com}} \\ &= (\mathbf{x}^\intercal \otimes \mathbf{I}_n)[\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}]\mathbf{T} \\ &= \mathbf{x}^\intercal \otimes \mathbf{G}, \end{aligned} \tag{3.13}$$

where the second line follows from the mixed product property (i.e., $\mathbf{A}(\mathbf{x}^\intercal \otimes \mathbf{I}_m) = (1 \otimes \mathbf{A})(\mathbf{x}^\intercal \otimes \mathbf{I}_m) = (\mathbf{x}^\intercal \otimes \mathbf{I}_n)(\mathbf{I}_\ell \otimes \mathbf{A})$) and the final line by Eq. (2.2). Consider now the main verification relation (Eq. (3.12)):

$$\begin{aligned} [\mathbf{A} \mid \mathbf{W}_0 + (\mathbf{x}^\intercal \otimes \mathbf{I}_n)\mathbf{W}]\mathbf{V} &= [\mathbf{A} \mid \mathbf{W}_0 + (\mathbf{x}^\intercal \otimes \mathbf{I}_n)\mathbf{W}] \begin{bmatrix} -(\mathbf{x}^\intercal \otimes \mathbf{I}_m)\mathbf{T}_{\mathsf{open}} \\ -\mathbf{T}_{\mathsf{com}} \end{bmatrix} \cdot \mathbf{H}_{\mathbf{B},f,\mathbf{x}} \\ &= (-\mathbf{W}_0 \mathbf{T}_{\mathsf{com}} - \mathbf{x}^\intercal \otimes \mathbf{G}) \cdot \mathbf{H}_{\mathbf{B},f,\mathbf{x}} \\ &= (\mathbf{B} - \mathbf{x}^\intercal \otimes \mathbf{G}) \cdot \mathbf{H}_{\mathbf{B},f,\mathbf{x}} \\ &= \mathbf{B}_f - f(\mathbf{x}) \cdot \mathbf{G}, \end{aligned}$$

using Eq. (3.13) and Theorem 2.13. To complete the proof, it suffices to bound $\|\mathbf{V}\|$. By Theorem 2.8, $\|\mathbf{T}\| \le \sqrt{m}\chi$. By Theorem 2.13, $\|\mathbf{H}_{\mathbf{B},f,\mathbf{x}}\| \le (n \log q)^{O(d)}$, so

$$\|\mathbf{V}\| \le (m^{3/2}\chi)(\ell m)(n \log q)^{O(d)} = \ell m^{5/2}\chi(n \log q)^{O(d)} \le B. \qquad \square$$

Theorem 3.22 (Selective-Input Binding)**.** *Suppose $n \ge O(\lambda)$, $m \ge O(n \log q)$, and $\chi \ge O(m \log n)$. Set $\beta \ge O(m^{5/2} B \log n)$. Then, under the ℓ-succinct SIS assumption with parameters (n, m, q, χ, β), Construction 3.19 satisfies computational selective-input binding.*

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

- Hyb_0: This is the selective-input computational binding experiment, where the adversary starts by committing to the input $\mathbf{x} \in \{0,1\}^\ell$. Then, the challenger samples $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$ and gives $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{W}_0, \mathbf{B})$ to $\mathcal{A}$. Specifically, it samples $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^\lambda, q, m)$, $\mathbf{W} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{\ell n \times m}$ and

$$\begin{bmatrix} \mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix} \leftarrow \mathsf{SamplePre}([\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}], \mathbf{I}_\ell \otimes \mathbf{R}, \mathbf{G}_{n\ell}, \chi),$$

  $\mathbf{W}_0 \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n \times m}$ and $\mathbf{B} = -\mathbf{W}_0 \mathbf{T}_{\mathsf{com}} \in \mathbb{Z}_q^{n \times \ell m}$. Algorithm $\mathcal{A}$ outputs a commitment $\sigma = \mathbf{B}_f \in \mathbb{Z}_q^{n \times m}$ and two openings $\mathbf{V}_0, \mathbf{V}_1 \in \mathbb{Z}_q^{2m \times m}$ for values 0 and 1, respectively.[9] The output of the experiment is 1 if the following conditions hold:

  - $\|\mathbf{V}_0\|, \|\mathbf{V}_1\| \le B$; and
  - $\mathbf{B}_f = [\mathbf{A} \mid \mathbf{W}_0 + (\mathbf{x}^\intercal \otimes \mathbf{I}_n)\mathbf{W}]\mathbf{V}_0$ and $\mathbf{B}_f - \mathbf{G} = [\mathbf{A} \mid \mathbf{W}_0 + (\mathbf{x}^\intercal \otimes \mathbf{I}_n)\mathbf{W}]\mathbf{V}_1$.

- Hyb_1: Same as Hyb_0 except the challenger samples $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n \times m}$, $\mathbf{W} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{\ell n \times m}$, and $\mathbf{T} \leftarrow [\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}]_\chi^{-1}(\mathbf{G}_{n\ell})$.
- Hyb_2: Same as Hyb_1 except the challenger samples $\mathbf{R}' \xleftarrow{\mathrm{R}} \{0,1\}^{m \times m}$ and $\mathbf{W}_0 \leftarrow \mathbf{A}\mathbf{R}' - (\mathbf{x}^\intercal \otimes \mathbf{I}_n)\mathbf{W}$.

For an adversary $\mathcal{A}$, we write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output of an execution of Hyb_i with adversary $\mathcal{A}$. We now reason about the output distributions of each adjacent pair of experiments:

Lemma 3.23. *Suppose $n \ge O(\lambda)$, $m \ge O(n \log q)$, and $\chi \ge O(m \log n)$. Then, for all adversaries $\mathcal{A}$, $\mathsf{Hyb}_0(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_1(\mathcal{A})$.*

Proof. This follows by a similar sequence of hybrid arguments as the proof of Lemma 3.7. Specifically, we introduce some additional intermediary distributions for $\mathbf{A}$, $\mathbf{T}_{\mathsf{open}}$, and $\mathbf{T}_{\mathsf{com}}$.

- Hyb_0: $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$ and $\begin{bmatrix} \mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix} \leftarrow \mathsf{SamplePre}([\mathbf{I}_L \otimes \mathbf{A} \mid \mathbf{W}], \mathbf{I}_L \otimes \mathbf{R}, \mathbf{G}_{n\ell}, \chi)$.
- $\mathsf{Hyb}_{0,1}$: $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$ and $\begin{bmatrix} \mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix} \leftarrow [\mathbf{I}_L \otimes \mathbf{A} \mid \mathbf{W}]_\chi^{-1}(\mathbf{G}_{n\ell})$.

  Indistinguishability: $\mathsf{Hyb}_0(\mathcal{A})$ and $\mathsf{Hyb}_{0,1}(\mathcal{A})$ are statistically indistinguishable when $n \ge O(\lambda)$, $m \ge O(n \log q)$ and $\chi \ge O(m \log n)$ by Theorem 2.8.

- $\mathsf{Hyb}_{0,2}$: $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n \times m}$ and $\begin{bmatrix} \mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix} \leftarrow [\mathbf{I}_L \otimes \mathbf{A} \mid \mathbf{W}]_\chi^{-1}(\mathbf{G}_{n\ell})$.

  Indistinguishability: $\mathsf{Hyb}_{0,1}(\mathcal{A})$ and $\mathsf{Hyb}_{0,2}(\mathcal{A})$ are statistically indistinguishable when $n \ge O(\lambda)$, $m \ge O(n \log q)$ by Theorem 2.8.

The claim now follows by a hybrid argument. $\square$

Lemma 3.24. *Suppose $m \ge 2\lambda + n \log q$. Then, for all adversaries $\mathcal{A}$, $\mathsf{Hyb}_1(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_2(\mathcal{A})$.*

[9] Note that we are considering single-bit outputs so without loss of generality, we require the adversary to output openings to 0 and 1, respectively, in the selective-input binding game.

Proof. Since $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}$ and $\mathbf{R}' \xleftarrow{R} \mathbb{Z}_q^{m\times m}$ where $m \geq 2\lambda + n\log q$ the distribution of $\mathbf{AR}'$ is statistically close to uniform over $\mathbb{Z}_q^{n\times m}$ by the leftover hash lemma (Lemma 2.4). Since $\mathbf{A}, \mathbf{R}'$ are sampled independently of $\mathbf{x}$ and $\mathbf{W}$, the distribution of $\mathbf{W}_0$ in Hyb_2 is statistically close to uniform, which coincides with its distribution in Hyb_1. □

Lemma 3.25. *Suppose $n \geq O(\lambda)$, $m \geq O(n\log q)$, and $\beta \geq O(m^{5/2}B\log n)$. Then, under the ℓ-succinct SIS assumption with parameters (n, m, q, χ, β), for all efficient adversaries $\mathcal{A}$, $\Pr[\mathsf{Hyb}_2(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$.*

Proof. Suppose there exists an efficient adversary $\mathcal{A}$ where $\Pr[\mathsf{Hyb}_2(\mathcal{A}) = 1] = \varepsilon$ for some non-negligible ε. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ for the ℓ-succinct SIS assumption:

1. At the beginning of the game, algorithm $\mathcal{B}$ receives $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{W} \in \mathbb{Z}_q^{n\ell\times m}$, and a trapdoor $\mathbf{T} \in \mathbb{Z}_q^{(\ell+1)m\times\ell m}$. Algorithm $\mathcal{B}$ parses $\mathbf{T} = \begin{bmatrix} \mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix}$ where $\mathbf{T}_{\mathsf{open}} \in \mathbb{Z}_q^{\ell m\times\ell m}$ and $\mathbf{T}_{\mathsf{com}} \in \mathbb{Z}_q^{m\times\ell m}$.

2. Algorithm $\mathcal{B}$ runs algorithm $\mathcal{A}$ to obtain $\mathbf{x} \in \{0,1\}^\ell$. Then, algorithm $\mathcal{B}$ samples $\mathbf{R}' \xleftarrow{R} \{0,1\}^{m\times m}$ and sets $\mathbf{W}_0 \leftarrow \mathbf{AR}' - (\mathbf{x}^\intercal \otimes \mathbf{I}_n)\mathbf{W}$ and $\mathbf{B} \leftarrow -\mathbf{W}_0\mathbf{T}_{\mathsf{com}}$. It gives $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{W}_0, \mathbf{B})$ to $\mathcal{A}$.

3. Algorithm $\mathcal{A}$ outputs a commitment $\mathbf{B}_f \in \mathbb{Z}_q^{n\times m}$ along with openings $\mathbf{V}_0, \mathbf{V}_1 \in \mathbb{Z}_q^{2m\times m}$. Algorithm $\mathcal{A}$ samples and outputs $\mathbf{x} \leftarrow \mathsf{SamplePre}(\mathbf{A}, [\mathbf{I}_m \mid \mathbf{R}'](\mathbf{V}_0 - \mathbf{V}_1), \mathbf{0}, \chi')$, where $\chi' = m(2m+1)B\log n = O(m^2B\log n)$.

In the ℓ-succinct SIS assumption, the challenger samples $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}$, $\mathbf{W} \xleftarrow{R} \mathbb{Z}_q^{n\ell\times m}$ and $\mathbf{T} \leftarrow [\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}]_\chi^{-1}(\mathbf{G}_{n\ell})$, so algorithm $\mathcal{B}$ perfectly simulates an execution of Hyb_2 for algorithm $\mathcal{B}$. Thus, with probability at least ε, algorithm $\mathcal{B}$ outputs a commitment $\mathbf{B}_f \in \mathbb{Z}_q^{n\times m}$ and openings $\mathbf{V}_0, \mathbf{V}_1 \in \mathbb{Z}_q^{2m\times m}$ where $\|\mathbf{V}_0\|, \|\mathbf{V}_1\| \leq B$ and

$$\mathbf{B}_f = [\mathbf{A} \mid \mathbf{W}_0 + (\mathbf{x}^\intercal \otimes \mathbf{I}_n)\mathbf{W}]\mathbf{V}_0 \quad \text{and} \quad \mathbf{B}_f - \mathbf{G} = [\mathbf{A} \mid \mathbf{W}_0 + (\mathbf{x}^\intercal \otimes \mathbf{I}_n)\mathbf{W}]\mathbf{V}_1.$$

By construction $\mathbf{W}_0 + (\mathbf{x}^\intercal \otimes \mathbf{I}_n)\mathbf{W} = \mathbf{AR}'$, so this means that

$$\mathbf{G} = [\mathbf{A} \mid \mathbf{AR}'](\mathbf{V}_0 - \mathbf{V}_1) = \mathbf{A}[\mathbf{I}_m \mid \mathbf{R}'](\mathbf{V}_0 - \mathbf{V}_1).$$

Thus, $[\mathbf{I}_m \mid \mathbf{R}'](\mathbf{V}_0 - \mathbf{V}_1)$ is a gadget trapdoor for $\mathbf{A}$ and $\|[\mathbf{I}_m \mid \mathbf{R}'](\mathbf{V}_0 - \mathbf{V}_1)\| \leq (2m+1)B$. When $\chi' \geq m(2m+1)B\log n$, the distribution of $\mathbf{x}$ is statistically close to $\mathbf{A}_{\chi'}^{-1}(\mathbf{0})$ by Theorem 2.8. By Lemmas 2.6 and 2.7,

$$0 < \|\mathbf{x}\| \leq \sqrt{m}\chi' = O(m^{5/2}B\log n) \leq \beta,$$

and the claim follows. □

The claim now follows by combining Lemmas 3.23 to 3.25. □

Parameter instantiation. Let λ be a security parameter and $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of functions $f\colon \{0,1\}^\ell \to \{0,1\}$ on inputs of length $\ell = \ell(\lambda)$ and which can be computed by Boolean circuits of depth $d = d(\lambda)$. We instantiate the lattice parameters in Construction 3.19 to satisfy Theorems 3.21 and 3.22 as follows:

- Let $\varepsilon > 0$ be a constant. We set the lattice dimension $n = d^{1/\varepsilon} \cdot \mathsf{poly}(\lambda)$ and $m = O(n\log q)$.
- We set $\chi = \mathsf{poly}(\lambda, m, \ell)$ and $B = \ell m^{5/2}\chi(n\log q)^{O(d)} = (n\log q)^{O(d)} \cdot \mathsf{poly}(\lambda, m, \ell)$.
- We choose the modulus $q = \beta \cdot \mathsf{poly}(n)$ so that the ℓ-succinct SIS assumption with parameters (n, m, q, χ, β) holds, where
$$\beta = O(m^{5/2}B\log n) = (n\log q)^{O(d)} \cdot \mathsf{poly}(\lambda, m, \ell) = 2^{\tilde{O}(d)} = 2^{\tilde{O}(n^\varepsilon)},$$
where we write $\tilde{O}(\cdot)$ to suppress polylogarithmic factors in λ, d, and ℓ. With this instantiation, $\log q = \mathsf{poly}(d^{1/\varepsilon}, \log\lambda, \log\ell)$, and we are relying on ℓ-succinct SIS with a *sub-exponential* noise bound.

With this setting of parameters, we obtain a dual functional commitment for $\mathcal{F}$ with the following parameter sizes:

- **Commitment size:** A commitment σ to a function $f\colon \{0,1\}^\ell \to \{0,1\}$ consists of a matrix $\sigma = \mathbf{B}_f \in \mathbb{Z}_q^{n\times m}$, so

$$|\sigma| = nm \log q = \mathsf{poly}(\lambda, d^{1/\varepsilon}, \log \ell).$$

- **Opening size:** An opening π to a function f consists of a matrix $\pi = \mathbf{V}_f \in \mathbb{Z}_q^{2m\times m}$ so

$$|\pi| = 2m^2 \log q = \mathsf{poly}(\lambda, d^{1/\varepsilon}, \log \ell).$$

- **CRS size:** The CRS consists of $(\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{W}_0, \mathbf{B})$ where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{W} \in \mathbb{Z}_q^{\ell n\times m}$, $\mathbf{T}_{\mathsf{open}} \in \mathbb{Z}_q^{\ell m\times \ell m}$, and $\mathbf{T}_{\mathsf{com}} \in \mathbb{Z}_q^{m\times \ell m}$, $\mathbf{W}_0 \in \mathbb{Z}_q^{n\times m}$, and $\mathbf{B} \in \mathbb{Z}_q^{n\times \ell m}$. Thus the total size of the CRS is

$$|\mathsf{crs}| = O(\ell^2 m^2) \cdot \log q = \ell^2 \cdot \mathsf{poly}(\lambda, d^{1/\varepsilon}, \log \ell).$$

We summarize the instantiation in the following corollary:

Corollary 3.26 (Dual Functional Commitment for Bounded-Depth Boolean Circuits)**.** *Let λ be a security parameter and let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of functions $f\colon \{0,1\}^\ell \to \{0,1\}$ on inputs of length $\ell = \ell(\lambda)$ and which can be computed by Boolean circuits of depth at most $d = d(\lambda)$. Under the ℓ-succinct SIS assumption with a sub-exponential norm bound $\beta = 2^{\tilde{O}(n^\varepsilon)}$ for some constant $\varepsilon > 0$ and lattice dimension $n = n(\lambda)$, there exists a dual functional commitment for $\mathcal{F}$. The functional commitment satisfies computational* selective-input *binding and supports preprocessing for fast verification (Definition 2.2). The size of the commitment and the opening have size* $\mathsf{poly}(\lambda, d^{1/\varepsilon}, \log \ell)$ *and the CRS has size $\ell^2 \cdot \mathsf{poly}(\lambda, d^{1/\varepsilon}, \log \ell)$.*

4 Cryptanalysis of Extractable Commitments

In this section, we describe some of the challenges in constructing extractable lattice-based functional commitments. First, we show that Construction 3.2 is not an extractable functional commitment for quadratic functions. We then show that assuming inhomogeneous SIS, the [ACL+22] approach does not yield an extractable functional commitment for linear functions. The attacks we develop work by using the components in the CRS to derive a basis for a lattice defined by the scheme's verification relation. We then use the basis to *obliviously* sample a solution that satisfies the schemes' verification relation *without* knowledge of a corresponding input. In one case (Section 4.1), this can be used to sample a valid opening to an unsatisfiable set of quadratic constraints, while in the other case (Section 4.2), we can embed a SIS instance that the extractor must solve in order to output a valid input. We start with a basic definition of a extractable functional commitment.

Definition 4.1 (Extractability)**.** Let λ be a security parameter. We say that a functional commitment $\Pi_{\mathsf{FC}} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Eval}, \mathsf{Verify})$ for a function family $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ is extractable if for all efficient adversaries $\mathcal{A}$, there exists an efficient extractor $\mathcal{E}$ such that

$$\Pr\left[\begin{array}{c} \exists i \in [T] : \mathsf{Verify}(\mathsf{crs}, \sigma, f_i, y_i, \pi_i) = 1 \text{ and} \\ f_i(\mathbf{x}) \neq y_i \end{array} : \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda) \\ \left((\sigma, \{(f_i, y_i, \pi_i)\}_{i\in[T]}), \mathbf{x}\right) \leftarrow (\mathcal{A}\|\mathcal{E})(1^\lambda, \mathsf{crs}) \end{array}\right] = \mathsf{negl}(\lambda).$$

Here, we write $(\mathcal{A}\|\mathcal{E})(\cdot)$ to denote invoking algorithm $\mathcal{A}$ and the extractor $\mathcal{E}$ on the same input *and* randomness. The output of $\mathcal{A}$ is a commitment σ along with a list of openings (f_i, y_i, π_i) for σ (for value y_i with respect to function f_i), and the output of $\mathcal{E}$ is $\mathbf{x}$.

4.1 An Attack on the Extractability of Construction 3.2

We begin by describing a (heuristic) attack on extractability for Construction 3.2. Here, we will just focus on the case of extraction for degree-2 polynomials over (a subset of) $\mathbb{Z}_q$ (for prime q).

Attack strategy. Suppose the input dimension satisfies $\ell > 6$. Let $\mathsf{crs} = (\mathbf{A}, \mathbf{W}_1, \mathbf{W}_2, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}) \leftarrow \mathsf{Setup}(1^\lambda)$ In the following we will view $\mathbf{W}_1$ as the vertical concatenation of ℓ matrices $\mathbf{W}_1^{(1)}, \ldots, \mathbf{W}_1^{(\ell)} \in \mathbb{Z}_q^{n \times m}$, one associated with each variable x_i. Similarly, we will view $\mathbf{W}_2$ as the vertical concatenation of ℓ^2 matrices $\mathbf{W}_2^{(i,j)} \in \mathbb{Z}_q^{n \times m}$, where $\mathbf{W}_2^{(i,j)}$ is associated with the product $x_i x_j$. Consider the following system of quadratic constraints:

$$x_1^2 = 0 \quad \text{and} \quad x_1 x_2 = 1. \tag{4.1}$$

Over $\mathbb{Z}_q$, this system of constraints is unsatisfiable since $x_1^2 = 0$ implies that $x_1 = 0$, and correspondingly, $x_1 x_2 = 0$. Then, to break extraction, it suffices for the adversary to construct a commitment $\mathbf{C} \in \mathbb{Z}_q^{m \times m}$ along with short openings $\mathbf{V}_{11}, \mathbf{V}_{12} \in \mathbb{Z}_q^{m \times m}$ such that

$$\mathbf{W}_2^{(1,1)} \mathbf{C}^2 = -\mathbf{A}\mathbf{V}_{11} \tag{4.2}$$

$$\mathbf{W}_2^{(1,2)} \mathbf{C}^2 = \mathbf{G} - \mathbf{A}\mathbf{V}_{12} \tag{4.3}$$

Suppose first that the adversary has a trapdoor for the following matrix $\mathbf{B}$:

$$\mathbf{B} = \begin{bmatrix} \mathbf{A} & & & \mathbf{W}_1^{(2)} \\ & \mathbf{A} & & \mathbf{W}_2^{(1,1)} \\ & & \mathbf{A} & \mathbf{W}_2^{(1,2)} \end{bmatrix} \in \mathbb{Z}_q^{3n \times 4m} \tag{4.4}$$

The adversary can use the trapdoor for $\mathbf{B}$ to efficiently sample a short solution $(\mathbf{V}_2', \mathbf{V}_{11}', \mathbf{V}_{12}', \mathbf{C})$ for the linear system

$$\begin{bmatrix} \mathbf{A} & & & \mathbf{W}_1^{(2)} \\ & \mathbf{A} & & \mathbf{W}_2^{(1,1)} \\ & & \mathbf{A} & \mathbf{W}_2^{(1,2)} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{V}_2' \\ \mathbf{V}_{11}' \\ \mathbf{V}_{12}' \\ \mathbf{C} \end{bmatrix} = \begin{bmatrix} \mathbf{G} \\ \mathbf{0} \\ \mathbf{W}_1^{(2)} \end{bmatrix}. \tag{4.5}$$

We now define $\mathbf{V}_{11}$ and $\mathbf{V}_{12}$ as follows:[10]

- From Eq. (4.5), $\mathbf{W}_2^{(1,1)} \mathbf{C} = -\mathbf{A}\mathbf{V}_{11}'$, so $\mathbf{W}_2^{(1,1)} \mathbf{C}^2 = -\mathbf{A}\mathbf{V}_{11}'\mathbf{C}$. Since $\mathbf{V}_{11}'$ and $\mathbf{C}$ are short, setting $\mathbf{V}_{11} = \mathbf{V}_{11}'\mathbf{C}$ yields a short solution to Eq. (4.2).
- Similarly Eq. (4.5) implies that $\mathbf{W}_2^{(1,2)} \mathbf{C} = \mathbf{W}_1^{(2)} - \mathbf{A}\mathbf{V}_{12}'$ and $\mathbf{W}_1^{(2)} \mathbf{C} = \mathbf{G} - \mathbf{A}\mathbf{V}_2'$. Thus,

  $$\mathbf{W}_2^{(1,2)} \mathbf{C}^2 = \mathbf{W}_1^{(2)} \mathbf{C} - \mathbf{A}\mathbf{V}_{12}'\mathbf{C} = \mathbf{G} - \mathbf{A}(\mathbf{V}_2' + \mathbf{V}_{12}'\mathbf{C}).$$

  Setting $\mathbf{V}_{12} = \mathbf{V}_2' + \mathbf{V}_{12}'\mathbf{C}$ yields a short solution to Eq. (4.3).

Thus, given a trapdoor for $\mathbf{B}$ in Eq. (4.4), it is straightforward to sample a commitment $\mathbf{C}$ and short openings $\mathbf{V}_{11}, \mathbf{V}_{12}$ that satisfy Eqs. (4.2) and (4.3). To complete the attack description, we show how the adversary can construct a trapdoor for $\mathbf{B}$ using the components in the CRS. This is immediate from the construction. Namely, the components $\mathbf{W}_1, \mathbf{W}_2$ and $\mathbf{T}_{\mathsf{open}}, \mathbf{T}_{\mathsf{com}}$ in the CRS satisfy

$$\left[\begin{array}{cc|c} \mathbf{I}_\ell \otimes \mathbf{A} & & \mathbf{W}_1 \\ & \mathbf{I}_{\ell^2} \otimes \mathbf{A} & \mathbf{W}_2 \end{array}\right] \cdot \begin{bmatrix} \mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_\ell \otimes \mathbf{G} \\ \mathbf{I}_\ell \otimes \mathbf{W}_1. \end{bmatrix} \in \mathbb{Z}_q^{(\ell+\ell^2)n \times \ell m}$$

Consider the subset of $3n$ rows corresponding to the blocks $\mathbf{W}_1^{(2)}$, $\mathbf{W}_2^{(1,1)}$, and $\mathbf{W}_2^{(1,2)}$:

$$\underbrace{\left[\begin{array}{cc|c} \mathbf{e}_2^\mathsf{T} \otimes \mathbf{A} & & \mathbf{W}_1^{(2)} \\ & \mathbf{e}_{11}^\mathsf{T} \otimes \mathbf{A} & \mathbf{W}_2^{(1,1)} \\ & \mathbf{e}_{12}^\mathsf{T} \otimes \mathbf{A} & \mathbf{W}_2^{(1,2)} \end{array}\right]}_{\mathbf{B}'} \cdot \begin{bmatrix} \mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{e}_2^\mathsf{T} \otimes \mathbf{G} \\ \mathbf{e}_1^\mathsf{T} \otimes \mathbf{W}_1^{(1)} \\ \mathbf{e}_1^\mathsf{T} \otimes \mathbf{W}_1^{(2)} \end{bmatrix}}_{\mathbf{U}} \in \mathbb{Z}_q^{3n \times \ell m},$$

[10] If the evaluator was *honest*, instead of targeting $\begin{bmatrix} \mathbf{G} \\ \mathbf{0} \\ \mathbf{W}_1^{(2)} \end{bmatrix}$ in Eq. (4.5), it would have targeted the matrix $\begin{bmatrix} x_2\mathbf{G} \\ x_1\mathbf{W}_1^{(1)} \\ x_1\mathbf{W}_1^{(2)} \end{bmatrix}$ for some $x_1, x_2 \in \{0,1\}$. Essentially, in this attack, the adversary is targeting a matrix that corresponds to an *inconsistent* assignment for x_1.

where $\mathbf{e}_i \in \mathbb{Z}_q^\ell$ denotes the i^{th} canonical basis vector. If we remove the columns of $\mathbf{B}'$ that are all-zeroes, and remove the corresponding rows from $\mathbf{T}_{\mathsf{open}}$ (call the resulting matrix $\mathbf{T}'_{\mathsf{open}} \in \mathbb{Z}_q^{3m \times \ell m}$), we see that

$$\mathbf{B} \cdot \begin{bmatrix} \mathbf{T}'_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix} = \mathbf{U} = [\mathbf{U}' \mid \mathbf{0}],$$

where $\mathbf{U}' \in \mathbb{Z}_q^{3n \times 2m}$. This means that $\begin{bmatrix} \mathbf{T}'_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix}$ contains $(\ell - 2)m$ short vectors in the kernel of $\mathbf{B}$. Since the dimension ℓ satisfies $\ell > 6$, the adversary obtains at least $5m > 4m$ short vectors in the kernel of $\mathbf{B}$. These vectors are sampled independently from a discrete Gaussian distribution, so *heuristically* we assume that there is a set of $4m$ linearly independent vectors over the *rationals*. If this holds, then we obtain an Ajtai trapdoor for $\mathbf{B}$ (Definition 2.10), which suffices to carry out the above attack. Putting everything together, we construct an adversary that breaks extractability of Construction 3.2 as follows:

1. Using the components $\mathbf{T}_{\mathsf{open}}$ and $\mathbf{T}_{\mathsf{com}}$ from the CRS, construct an Ajtai trapdoor for the matrix $\mathbf{B}$ in Eq. (4.4). The trapdoor is formed by taking a subset of the rows of $\begin{bmatrix} \mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix}$.
2. Using the Ajtai trapdoor for $\mathbf{B}$, sample a short solution $(\mathbf{V}'_2, \mathbf{V}'_{11}, \mathbf{V}'_{12}, \mathbf{C})$ to Eq. (4.5).
3. Output the commitment $\mathbf{C}$ and openings $\mathbf{V}_{11} = \mathbf{V}'_{11}\mathbf{C}$ and $\mathbf{V}_{12} = \mathbf{V}'_2 + \mathbf{V}'_{12}\mathbf{C}$. This is a valid opening for the constraints in Eq. (4.1), which is an *unsatisfiable* quadratic system.

Cryptanalysis of a candidate defense based on sparsification. We briefly remark that the above oblivious sampling attack still works even if we adopt the [ACL+22] strategy of "sparsification." In the [ACL+22] approach (see also Section 4.2), a commitment would only be considered valid if the adversary can additionally output a short opening $\mathbf{V}_{\mathsf{ext}}$ such that $\mathbf{A}_{\mathsf{ext}}\mathbf{V}_{\mathsf{ext}} = \mathbf{W}_{\mathsf{ext}}\mathbf{C}$, where $\mathbf{A}_{\mathsf{ext}} \in \mathbb{Z}_q^{tm \times tm \log q}$ is a random matrix and $\mathbf{W}_{\mathsf{ext}} \in \mathbb{Z}_q^{t \times m}$ where $t \gg m$. To facilitate this, the CRS would now contain short matrices $(\mathbf{T}_{\mathsf{open}}, \mathbf{T}_{\mathsf{ext}}, \mathbf{T}_{\mathsf{com}})$ where

$$\begin{bmatrix} \mathbf{I}_{\ell+\ell^2} \otimes \mathbf{A} & & \mathbf{W} \\ & \mathbf{A}_{\mathsf{ext}} & \mathbf{W}_{\mathsf{ext}} \end{bmatrix} \begin{bmatrix} \mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{ext}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix} = \begin{bmatrix} \mathbf{P} \\ \mathbf{0} \end{bmatrix} \quad \text{where} \quad \mathbf{P} = \begin{bmatrix} \mathbf{I}_\ell \otimes \mathbf{G} \\ \mathbf{I}_\ell \otimes \mathbf{W}_1 \end{bmatrix}.$$

The intuition in [ACL+22] (see also Section 4.2) is that the only way to sample low-norm $\mathbf{V}_{\mathsf{ext}}, \mathbf{C}$ that satisfies this verification relation is to multiply $\mathbf{T}_{\mathsf{ext}}, \mathbf{T}_{\mathsf{com}}$ by the same low-norm matrix, from which we can "extract" $\mathbf{x}$. However, the attack strategy described above naturally extends to this setting, except we now use the components in the CRS to derive a trapdoor for the extended matrix

$$\mathbf{B} = \begin{bmatrix} \mathbf{A} & & & & \mathbf{W}_1^{(2)} \\ & \mathbf{A} & & & \mathbf{W}_2^{(1,1)} \\ & & \mathbf{A} & & \mathbf{W}_2^{(1,2)} \\ & & & \mathbf{A}_{\mathsf{ext}} & \mathbf{W}_{\mathsf{ext}} \end{bmatrix}$$

In the next section, we apply a similar methodology to analyze a variant of the linear functional commitment scheme from [ACL+22].

4.2 Analyzing the [ACL+22] Knowledge Assumption

In this section, we analyze one version of the k-ISIS and knowledge k-ISIS family of assumptions from [ACL+22]. While the original assumptions from [ACL+22] were defined over polynomial rings (and module/ideal lattices), we consider the analogous assumptions over the integers. Since ring multiplication is commutative whereas matrix multiplication is not, there are multiple (and similar) ways to translate the [ACL+22] family of assumptions to the integers. We describe one adaptation here, where we "sparsify by left multiplication." We refer to this adaptation as the MatrixACLMT construction.

Assumption 4.2 (MatrixACLMT k-ISIS Assumption for Linear Functions)**.** Let λ be a security parameter and let $(n, m, q, \chi, \ell, \beta)$ be lattice parameters. The MatrixACLMT k-SIS assumption says that for every efficient adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\Pr\left[\begin{array}{c} \mathbf{A}\mathbf{x} = \alpha\mathbf{u} \bmod q \\ \text{and} \\ 0 < |\alpha|, \|\mathbf{x}\| \leq \beta \end{array} : \begin{array}{c} \mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m},\ \mathbf{u} \xleftarrow{R} \mathbb{Z}_q^n, \\ \forall i \in [\ell] : \mathbf{W}_i \xleftarrow{R} \mathbb{Z}_q^{n\times n},\ \mathbf{t}_i \leftarrow \mathbf{W}_i^{-1}\mathbf{u}, \\ \forall i \neq j : \mathbf{z}_{i,j} \leftarrow \mathbf{A}_\chi^{-1}(\mathbf{W}_i\mathbf{t}_j), \\ (\alpha, \mathbf{x}) \leftarrow \mathcal{A}\big(1^\lambda, \mathbf{A}, \mathbf{u}, \{\mathbf{W}_i\}_{i\in[\ell]}, \{\mathbf{z}_{i,j}\}_{i\neq j}\big) \end{array}\right] = \mathsf{negl}(\lambda).$$

Assumption 4.3 (MatrixACLMT Knowledge Assumption)**.** Let λ be a security parameter and let $(n, m, q, \chi, t, \ell, \alpha, \beta)$ be lattice parameters where $q^{n-t} = \mathsf{negl}(\lambda)$ and $m \geq O(t \log q)$. The MatrixACLMT knowledge assumption says that for every efficient adversary $\mathcal{A}$, there exists an efficient extractor $\mathcal{E}$ such that

$$\Pr\left[\begin{array}{c} \mathbf{A}\mathbf{v} = \mathbf{D}\mathbf{c} \bmod q \text{ and } \|\mathbf{v}\| \leq \beta \text{ and} \\ (\|\mathbf{x}\| \geq \alpha \text{ or } \mathbf{c} \neq \sum_{i\in[\ell]} x_i\mathbf{t}_i \bmod q) \end{array} : \begin{array}{c} \mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{t\times m}, \mathbf{D} \xleftarrow{R} \mathbb{Z}_q^{t\times n}, \\ \forall i \in [\ell] : \mathbf{t}_i \xleftarrow{R} \mathbb{Z}_q^n,\ \mathbf{z}_i \leftarrow \mathbf{A}_\chi^{-1}(\mathbf{D}\mathbf{t}_i) \\ \big((\mathbf{c}, \mathbf{v}), \mathbf{x}\big) \leftarrow (\mathcal{A}\|\mathcal{E})\big(1^\lambda, \mathbf{A}, \mathbf{D}, \{(\mathbf{t}_i, \mathbf{z}_i)\}_{i\in[\ell]}\big) \end{array}\right] = \mathsf{negl}(\lambda),$$

where $((\mathbf{c}, \mathbf{v}), \mathbf{x}) \leftarrow (\mathcal{A}\|\mathcal{E})(1^\lambda, \mathbf{A}, \mathbf{D}, \{(\mathbf{t}_i, \mathbf{z}_i)\}_{i\in[\ell]})$ denotes that $\mathcal{A}$ and $\mathcal{E}$ are invoked on the same input *and* randomness, and $(\mathbf{c}, \mathbf{v})$ is the output of $\mathcal{A}$ while $\mathbf{x}$ is the output of $\mathcal{E}$.

The MatrixACLMT knowledge assumption essentially says that any efficient adversarial strategy that produces a short $\mathbf{v} \in \mathbb{Z}_q^m$ where $\mathbf{A}\mathbf{v} \in \mathbb{Z}_q^t$ lies in the image of $\mathbf{D}$ (i.e., $\mathbf{A}\mathbf{v} = \mathbf{D}\mathbf{c}$) can be explained as taking a short linear combination of the given preimages $\mathbf{z}_1, \ldots, \mathbf{z}_\ell$. Indeed, if $\mathbf{c} = \sum_{i\in[\ell]} x_i\mathbf{t}_i$, then we can write $\mathbf{v} = \sum_{i\in[\ell]} x_i\mathbf{z}_i$:

$$\mathbf{D}\mathbf{c} = \mathbf{D}\left(\sum_{i\in[\ell]} x_t\mathbf{t}_i\right) = \mathbf{A}\left(\sum_{i\in[\ell]} x_i\mathbf{z}_i\right) = \mathbf{A}\mathbf{v}.$$

The requirement $q^{n-t} = \mathsf{negl}(\lambda)$ is necessary to prevent the basic oblivious sampling attack where the adversary samples a random short vector $\mathbf{v} \in \mathbb{Z}_q^m$ and solves for a $\mathbf{c} \in \mathbb{Z}_q^n$ satisfying $\mathbf{A}\mathbf{v} = \mathbf{D}\mathbf{c}$. Since the image of $\mathbf{A}$ has q^t elements and the image of $\mathbf{D}$ has q^n elements, only a negligible fraction of the elements in the image of $\mathbf{A}$ are also in the image of $\mathbf{D}$ (i.e., for most vectors $\mathbf{v} \in \mathbb{Z}_q^m$, there will not exist a vector $\mathbf{c} \in \mathbb{Z}_q^n$ where $\mathbf{A}\mathbf{v} = \mathbf{D}\mathbf{c}$).

A heuristic oblivious sampling algorithm for Assumption 4.3. We start by describing an adversary for Assumption 4.3 that *obliviously samples* a short vector $\mathbf{v} \in \mathbb{Z}_q^m$ such that $\mathbf{A}\mathbf{v}$ is in the image of $\mathbf{D}$. While this by itself does not necessarily falsify Assumption 4.3, we will subsequently show that Assumptions 4.2 and 4.3 cannot simultaneously hold for a broad range of parameter settings (i.e., at least one of Assumption 4.2 or Assumption 4.3 is false).

Algorithm 4.4 (Candidate Oblivious Sampler for MatrixACLMT)**.** Suppose $\ell \gg m + n$ in Assumption 4.3. Our heuristic oblivious sampling algorithm $\mathcal{A}$ for Assumption 4.3 works as follows:

1. Let $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{t\times m}, \mathbf{D} \xleftarrow{R} \mathbb{Z}_q^{t\times n}, \mathbf{t}_i \xleftarrow{R} \mathbb{Z}_q^n$ and $\mathbf{z}_i \leftarrow \mathbf{A}_\chi^{-1}(\mathbf{D}\mathbf{t}_i)$ be the challenge from Assumption 4.3. By construction,

$$[\mathbf{A} \mid \mathbf{D}\mathbf{G}] \cdot \underbrace{\begin{bmatrix} \mathbf{z}_1 & \cdots & \mathbf{z}_\ell \\ -\mathbf{G}^{-1}(\mathbf{t}_1) & \cdots & -\mathbf{G}^{-1}(\mathbf{t}_\ell) \end{bmatrix}}_{\bar{\mathbf{T}}} = \mathbf{0} \bmod q.$$

   Since $\mathbf{t}_i$ and $\mathbf{z}_i$ are sampled independently and assuming that $\ell \gg m + n$ is sufficiently large (e.g., setting $\ell = 2(m + n)$ should suffice), we can heuristically assume that $\bar{\mathbf{T}} \in \mathbb{Z}^{(m+n)\times\ell}$ is full rank over the *reals*.[11] Thus, we can use $\bar{\mathbf{T}}$ to derive an Ajtai-trapdoor $\mathbf{T}$ (Definition 2.10) for the matrix $\mathbf{B} = [\mathbf{A} \mid \mathbf{D}\mathbf{G}]$ (e.g., by taking a subset of $m + n$ columns of $\bar{\mathbf{T}}$ that are linearly independent over the reals).

[11] Note that $\bar{\mathbf{T}}$ does *not* (and cannot) have full rank over $\mathbb{Z}_q$.

2. Using $\mathbf{T}$, the algorithm samples a short $\left[\begin{smallmatrix}\mathbf{v}\\ \mathbf{c}\end{smallmatrix}\right]$ where $\mathbf{B}\cdot\left[\begin{smallmatrix}\mathbf{v}\\ \mathbf{c}\end{smallmatrix}\right]=\mathbf{0}$. The commitment is then $\mathbf{Gc}$ and the opening is $\mathbf{v}$. For instance, the algorithm might implement Babai's rounding algorithm. Specifically, it starts with an arbitrary (non-zero) solution $\mathbf{y}\in\mathbb{Z}^{m+n}$ where $\mathbf{By}=\mathbf{0} \bmod q$, solves for the unique $\mathbf{z}\in\mathbb{Q}^{m+n}$ where $\mathbf{Tz}=\mathbf{y}\in\mathbb{Q}^{m+n}$ and then outputs $\mathbf{x}=\mathbf{y}-\mathbf{T}\cdot\lfloor\mathbf{z}\rceil$. By construction $\mathbf{Bx}=\mathbf{0} \bmod q$ and moreover $\|\mathbf{x}\|\leq\|\mathbf{T}(\mathbf{z}-\lfloor\mathbf{z}\rceil)\|$, which is small.

The basic question is whether the solution $\mathbf{x}$ derived by rounding off a long solution as in Algorithm 4.4 (or sampled through some alternative trapdoor sampling algorithm) can *always* be explained by a short linear combination of the basis vectors $\mathbf{T}$. We note that in the particular case of Assumption 4.3, if $\ell\gg m+n$, then the adversary actually has ℓ short vectors in the kernel of $\mathbf{B}$, so the extractor does have more flexibility in coming up with a linear strategy. It is an interesting challenge to either write down an explicit extractor for this oblivious sampling strategy or prove that no such strategy is possible (say, under a standard computational assumption). In the following, we show that assuming (non-uniform) hardness of inhomogeneous SIS and the matrix-ACLMT assumption for linear functions (Assumption 4.2), then no such extractor exists. One implication of this is that this particular adaptation of [ACL+22] to the integers is not an extractable functional commitment for linear functions.

Attacking the Matrix-ACLMT commitment for linear functions. We now show how we can apply the approach in Algorithm 4.4 to break extractability for the linear functional commitment from [ACL+22] (when instantiated over the integers). We start by recalling their construction (over the integers):

Construction 4.5 (Functional Commitment for Linear Functions)**.** Let λ be a security parameter and n, m, m', q, t, B, χ be lattice parameters. Let $\ell=\ell(\lambda)$ be the input length. For a matrix $\mathbf{M}\in\mathbb{Z}_q^{k\times\ell}$, let $f_{\mathbf{M}}\colon\mathbb{Z}_q^{k\times\ell}\to\mathbb{Z}_q^k$ be the linear function $\mathbf{x}\mapsto\mathbf{Mx}$. Let $\mathcal{F}_\lambda=\{f_{\mathbf{M}}\mid\mathbf{M}\in\{0,1\}^{k\times\ell}\}$. We construct a functional commitment $\Pi_{\mathsf{FC}}=(\mathsf{Setup},\mathsf{Commit},\mathsf{Eval},\mathsf{Verify})$ for $\mathcal{F}=\{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ as follows:

- $\mathsf{Setup}(1^\lambda,1^\ell)$: Sample matrices $(\mathbf{A},\mathbf{R_A})\leftarrow\mathsf{TrapGen}(1^\lambda,n,m)$, $\mathbf{W}_1,\ldots,\mathbf{W}_\ell\xleftarrow{\mathrm{R}}\mathbb{Z}_q^{n\times n}$, $\mathbf{u}\leftarrow\mathbb{Z}_q^n$, and let $\mathbf{t}_i\leftarrow\mathbf{W}_i^{-1}\mathbf{u}\in\mathbb{Z}_q^n$ for each $i\in[\ell]$. For each $i\neq j$, sample $\mathbf{z}_{i,j}\leftarrow\mathsf{SamplePre}(\mathbf{A},\mathbf{R_A},\mathbf{W}_i\mathbf{t}_j,\chi)$. Let $\widehat{\mathbf{W}}\in\mathbb{Z}_q^{\ell n\times n}$ be the vertical stacking of the matrices $\mathbf{W}_1,\ldots,\mathbf{W}_\ell$:

$$\widehat{\mathbf{W}}=\begin{bmatrix}\mathbf{W}_1\\ \vdots\\ \mathbf{W}_\ell\end{bmatrix}\in\mathbb{Z}_q^{\ell n\times n}.$$

  Next, sample $(\mathbf{B},\mathbf{R_B})\leftarrow\mathsf{TrapGen}(1^\lambda,t,m')$ and a matrix $\mathbf{D}\xleftarrow{\mathrm{R}}\mathbb{Z}_q^{t\times n}$. Sample $\mathbf{z}_i'\leftarrow\mathsf{SamplePre}(\mathbf{B},\mathbf{R_B},\mathbf{Dt}_i,\chi)$ for each $i\in[\ell]$. Output the common reference string $\mathsf{crs}=\big(\mathbf{A},\mathbf{B},\mathbf{D},\mathbf{u},\{\mathbf{W}_i\}_{i\in[\ell]},\{\mathbf{z}_{i,j}\}_{i\neq j},\{\mathbf{z}_i'\}_{i\in[\ell]}\big)$.

- $\mathsf{Commit}(\mathsf{crs},\mathbf{x})$: On input the common reference string $\mathsf{crs}=\big(\mathbf{A},\mathbf{B},\mathbf{D},\mathbf{u},\{\mathbf{W}_i\}_{i\in[\ell]},\{\mathbf{z}_{i,j}\}_{i\neq j},\{\mathbf{z}_i'\}_{i\in[\ell]}\big)$ and an input vector $\mathbf{x}\in\mathbb{Z}_q^\ell$, the commit algorithm outputs the commitment $\mathbf{c}=\sum_{i\in[\ell]}x_i\mathbf{t}_i\in\mathbb{Z}_q^n$ and the state $\mathsf{st}=\mathbf{x}$.

- $\mathsf{Eval}(\mathsf{crs},\mathsf{st},f_{\mathbf{M}})$: On input the common reference string $\mathsf{crs}=\big(\mathbf{A},\mathbf{B},\mathbf{D},\mathbf{u},\{\mathbf{W}_i\}_{i\in[\ell]},\{\mathbf{z}_{i,j}\}_{i\neq j},\{\mathbf{z}_i'\}_{i\in[\ell]}\big)$, a commitment state $\mathsf{st}=\mathbf{x}$, and a function $f_{\mathbf{M}}$ for some matrix $\mathbf{M}\in\mathbb{Z}_q^{k\times\ell}$, the evaluation algorithm computes $\widehat{\mathbf{v}}_i\leftarrow\sum_{j\neq i}x_j\mathbf{z}_{i,j}$ for each $i\in[\ell]$ and defines $\widehat{\mathbf{v}}\in\mathbb{Z}_q^{\ell m}$ and $\hat{\mathbf{z}}\in\mathbb{Z}_q^{\ell m'}$ as follows:

$$\widehat{\mathbf{v}}=\begin{bmatrix}\widehat{\mathbf{v}}_1\\ \vdots\\ \widehat{\mathbf{v}}_\ell\end{bmatrix}\in\mathbb{Z}_q^{\ell m}\quad\text{and}\quad\hat{\mathbf{z}}=\begin{bmatrix}\mathbf{z}_1'\\ \vdots\\ \mathbf{z}_\ell'\end{bmatrix}.$$

  It outputs the opening

$$\mathbf{v}=\begin{bmatrix}(\mathbf{M}\otimes\mathbf{I}_m)\widehat{\mathbf{v}}\\ (\mathbf{x}^\intercal\otimes\mathbf{I}_{m'})\mathbf{z}_i'\end{bmatrix}\in\mathbb{Z}_q^{km+m'}.$$

- $\mathsf{Verify}(\mathsf{crs}, \sigma, f_{\mathbf{M}}, y, \pi)$: On input the common reference string $\mathsf{crs} = \left(\mathbf{A}, \mathbf{B}, \mathbf{D}, \mathbf{u}, \{\mathbf{W}_i\}_{i\in[\ell]}, \{\mathbf{z}_{i,j}\}_{i\neq j}, \{\mathbf{z}'_i\}_{i\in[\ell]}\right)$, a commitment $\sigma = \mathbf{c} \in \mathbb{Z}_q^n$, a function $f_{\mathbf{M}} \colon \mathbb{Z}_q^{k\times\ell} \to \mathbb{Z}_q^k$ where $\mathbf{M} \in \mathbb{Z}_q^{k\times\ell}$, a value $\mathbf{y} \in \mathbb{Z}_q^k$, and an opening $\pi = \mathbf{v} \in \mathbb{Z}_q^{(km+m')\times m}$, the verification algorithm outputs 1 if

$$\|\mathbf{v}\| \leq B \quad \text{and} \quad \begin{bmatrix} \mathbf{I}_k \otimes \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix} \cdot \mathbf{v} = \begin{bmatrix} (\mathbf{M} \otimes \mathbf{I}_n)\widehat{\mathbf{W}} \\ \mathbf{D} \end{bmatrix} \cdot \mathbf{c} - \begin{bmatrix} \mathbf{y} \otimes \mathbf{u} \\ \mathbf{0} \end{bmatrix}. \tag{4.6}$$

Correctness. Correctness follows by the same argument as in [ACL+22], adapted to operate over the integers. We give a sketch here and refer to [ACL+22] for more details. Let $\mathsf{crs} = \left(\mathbf{A}, \mathbf{B}, \mathbf{D}, \mathbf{u}, \{\mathbf{W}_i\}_{i\in[\ell]}, \{\mathbf{z}_{i,j}\}_{i\neq j}, \{\mathbf{z}'_i\}_{i\in[\ell]}\right)$ be a CRS sampled via the Setup algorithm. Suppose $\mathbf{c} = \sum_{i\in[\ell]} x_i \mathbf{t}_i$ is a commitment to a *short* input $\mathbf{x} \in \mathbb{Z}_q^\ell$. Suppose $\mathbf{v}$ is an opening to a function $f_{\mathbf{M}}$ where $\mathbf{M} \in \mathbb{Z}_q^{k\times\ell}$ is a matrix with small entries. By construction, if the entries of $\mathbf{M}$ and $\mathbf{x}$ are short, then so is $\mathbf{v}$. Consider now the main verification relation. First, for each $i \in [\ell]$,

$$\mathbf{W}_i\mathbf{c} = \sum_{j\in[\ell]} x_j \mathbf{W}_i \mathbf{t}_j = x_i\mathbf{u} + \sum_{j\neq i} x_j \mathbf{A}\mathbf{z}_{i,j} = x_i\mathbf{u} + \mathbf{A}\widehat{\mathbf{v}}_i.$$

Equivalently, this means $\widehat{\mathbf{W}}\mathbf{c} = \mathbf{x} \otimes \mathbf{u} + (\mathbf{I}_\ell \otimes \mathbf{A})\widehat{\mathbf{v}}$. Consider now the main verification relation:

$$\begin{aligned}(\mathbf{M} \otimes \mathbf{I}_n)\widehat{\mathbf{W}}\mathbf{c} &= (\mathbf{M} \otimes \mathbf{I}_n)(\mathbf{x} \otimes \mathbf{u}) + (\mathbf{M} \otimes \mathbf{I}_n)(\mathbf{I}_\ell \otimes \mathbf{A})\widehat{\mathbf{v}} \\ &= (\mathbf{M}\mathbf{x} \otimes \mathbf{u}) + (\mathbf{I}_k \otimes \mathbf{A})(\mathbf{M} \otimes \mathbf{I}_m)\widehat{\mathbf{v}} \\ \mathbf{D}\mathbf{c} = \sum_{i\in[\ell]} x_i \mathbf{D}\mathbf{t}_i &= \mathbf{B} \cdot \left(\sum_{i\in[\ell]} x_i \mathbf{z}'_i\right) = \mathbf{B} \cdot (\mathbf{x}^\intercal \otimes \mathbf{I}_{m'})\hat{\mathbf{z}}.\end{aligned}$$

For a sufficiently-large bound B, the verification relations hold and correctness follows.

Extractability. By an analogous argument as in [ACL+22], we can show that under Assumptions 4.2 and 4.3 (with suitable parameter instantiations), if an efficient adversary can produce a commitment $\sigma = \mathbf{c}$ along with a valid opening $\pi = \mathbf{v}$ to a short value $\mathbf{y} \in \mathbb{Z}_q^t$ with respect to a linear function $f_{\mathbf{M}}$ with short $\mathbf{M} \in \mathbb{Z}_q^{k\times\ell}$, then there exists an efficient extractor that outputs a *short* $\mathbf{x} \in \mathbb{Z}_q^\ell$ where $\mathbf{M}\mathbf{x} = \mathbf{y}$. We give a sketch of the general approach here and refer to [ACL+22] for a formal argument:

- Suppose there exists an efficient adversary $\mathcal{A}$ is able to come up with a commitment $\mathbf{c} \in \mathbb{Z}_q^n$ and a short opening $\mathbf{v} = \left[\begin{smallmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{smallmatrix}\right]$ that satisfies Eq. (4.6). This means that $\mathbf{B}\mathbf{v}_2 = \mathbf{D}\mathbf{c}$. By Assumption 4.3, there exists an efficient extractor $\mathcal{E}$ that outputs a short $\mathbf{x} \in \mathbb{Z}_q^\ell$ such that $\mathbf{c} = \sum_{i\in[\ell]} x_i \mathbf{t}_i$.
- If the extracted $\mathbf{x}$ satisfies $\mathbf{M}\mathbf{x} = \mathbf{y}$, then the extractor is successful. Consider the case where $\mathbf{M}\mathbf{x} \neq \mathbf{y}$. If this happens with non-negligible probability, we can construct an adversary $\mathcal{B}$ that uses the extractor $\mathcal{E}$ to break Assumption 4.2:
  1. Algorithm $\mathcal{B}$ receives $\left(\mathbf{A}, \mathbf{u}, \{\mathbf{W}_i\}_{i\in[\ell]}, \{\mathbf{z}_{i,j}\}_{i\neq j}\right)$ from the challenger for Assumption 4.2.
  2. It samples $(\mathbf{B}, \mathbf{R}_\mathbf{B}) \leftarrow \mathsf{TrapGen}(1^\lambda, t, m')$, $\mathbf{D} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{t\times n}$, and $\mathbf{z}'_i \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathbf{R}_\mathbf{B}, \mathbf{D}\mathbf{t}_i, \chi)$ for each $i \in [\ell]$ as in the real scheme. The reduction algorithm constructs the common reference string $\mathsf{crs} = \left(\mathbf{A}, \mathbf{B}, \mathbf{D}, \mathbf{u}, \{\mathbf{W}_i\}_{i\in[\ell]}, \{\mathbf{z}_{i,j}\}_{i\neq j}, \{\mathbf{z}'_i\}_{i\in[\ell]}\right)$ and gives crs to $\mathcal{A}$.
  3. After $\mathcal{A}$ outputs a commitment $\mathbf{c}$ and an opening $\mathbf{v} = \left[\begin{smallmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{smallmatrix}\right]$ to a value $\mathbf{y}$, algorithm $\mathcal{B}$ runs the extractor $\mathcal{E}$ on the same input as $\mathcal{A}$ to obtain a short input $\mathbf{x} \in \mathbb{Z}_q^\ell$. Suppose $\mathbf{M}\mathbf{x} = \mathbf{y}' \neq \mathbf{y}$. Then algorithm $\mathcal{A}$ computes an opening $\mathbf{v}' = \left[\begin{smallmatrix} \mathbf{v}'_1 \\ \mathbf{v}'_2 \end{smallmatrix}\right]$ by computing $\mathsf{Eval}(\mathsf{crs}, \mathbf{x}, f_\mathbf{M})$. By correctness, $\mathbf{v}'$ is short and moreover satisfies the following verification relation from Eq. (4.6):

$$(\mathbf{I}_k \otimes \mathbf{A})\mathbf{v}'_1 = (\mathbf{M} \otimes \mathbf{I}_n)\widehat{\mathbf{W}}\mathbf{c} - \mathbf{M}\mathbf{x} \otimes \mathbf{u} \tag{4.7}$$

Since $\mathbf{v}$ is also a valid opening, we have that

$$(\mathbf{I}_k \otimes \mathbf{A})(\mathbf{v}_1 - \mathbf{v}_1') = (\mathbf{y}' - \mathbf{y}) \otimes \mathbf{u}.$$

Since $\mathbf{y} - \mathbf{y}' \neq \mathbf{0}$, there is at least one non-zero "block" where $\mathbf{A}(\mathbf{v}_{1,i} - \mathbf{v}_{1,i}') = (y_i' - y_i)\mathbf{u}$ and $y_i' \neq y_i$. Since $\mathbf{y}, \mathbf{y}'$ are both short, this yields a valid solution to Assumption 4.2.

An attack on Construction 4.5. To conclude, we describe a (heuristic) attack that breaks extractability of Construction 4.5. Our approach takes the following template:

1. Given the CRS for the functional commitment scheme, we construct an efficient adversary $\mathcal{A}$ that can obliviously sample an opening to an arbitrary vector $\mathbf{y} \in \mathbb{Z}_q^k$ with respect to a function $f_{\mathbf{M}}$ where $\mathbf{M} = [\mathbf{M}_{\mathrm{L}} \mid \mathbf{0}^{k \times \ell_1}]$ and $\mathbf{M}_{\mathrm{L}} \in \mathbb{Z}_q^{k \times \ell_2}$ is short.
2. Extractability of the functional commitment now says that there exists an efficient extractor that outputs a short $\mathbf{x} \in \mathbb{Z}_q^{\ell_1 + \ell_2}$ such that $\mathbf{M}\mathbf{x} = \mathbf{y}$.
3. Since the oblivious sampler is agnostic to the choice of $\mathbf{M}_{\mathrm{L}}$ (as long as it is short), we can embed an (inhomogeneous) SIS instance into $\mathbf{M}_{\mathrm{L}}$. In this case, an extractor for algorithm $\mathcal{A}$ is able to solve inhomogeneous SIS with respect to $\mathbf{M}$, and by extension, $\mathbf{M}_{\mathrm{L}}$.

We now describe the first step in more detail. Here, we consider an instance of the vector commitment scheme for inputs of length $\ell_1 + \ell_2$:

- Let $\mathsf{crs} = \left(\mathbf{A}, \mathbf{B}, \mathbf{D}, \mathbf{u}, \{\mathbf{W}_i\}_{i \in [2\ell]}, \{\mathbf{z}_{i,j}\}_{i \neq j}, \{\mathbf{z}_i'\}_{i \in [2\ell]}\right)$. Suppose we want to open a function $f_{\mathbf{M}}$ for some matrix $\mathbf{M} = [\mathbf{M}_{\mathrm{L}} \mid \mathbf{0}^{k \times \ell}]$ to $\mathbf{y}$. The goal is to sample a commitment $\mathbf{c} \in \mathbb{Z}_q^n$ and short openings $\mathbf{v} = \left[\begin{smallmatrix} \mathbf{v}_{\mathsf{fc}} \\ \mathbf{v}_{\mathsf{ext}} \end{smallmatrix}\right] \in \mathbb{Z}_q^{km+m'}$ where

$$\begin{bmatrix} \mathbf{I}_k \otimes \mathbf{A} & \mathbf{0} & -(\mathbf{M} \otimes \mathbf{I}_n)\widehat{\mathbf{W}} \\ \mathbf{0} & \mathbf{B} & -\mathbf{D} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v}_{\mathsf{fc}} \\ \mathbf{v}_{\mathsf{ext}} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{y} \otimes \mathbf{u} \\ \mathbf{0} \end{bmatrix}.$$

  Since $\mathbf{M} = [\mathbf{M}_{\mathrm{L}} \mid \mathbf{0}^{k \times \ell_2}]$, this is equivalent to

$$\begin{bmatrix} \mathbf{I}_k \otimes \mathbf{A} & \mathbf{0} & -(\mathbf{M}_{\mathrm{L}} \otimes \mathbf{I}_n)\widehat{\mathbf{W}}_{\mathrm{T}} \\ \mathbf{0} & \mathbf{B} & -\mathbf{D} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v}_{\mathsf{fc}} \\ \mathbf{v}_{\mathsf{ext}} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{y} \otimes \mathbf{u} \\ \mathbf{0} \end{bmatrix} \quad \text{where} \quad \widehat{\mathbf{W}}_{\mathrm{T}} = \begin{bmatrix} \mathbf{W}_1 \\ \vdots \\ \mathbf{W}_{\ell_1} \end{bmatrix} \in \mathbb{Z}_q^{\ell_1 n \times n}$$

  Define the related matrix

$$\tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{I}_k \otimes \mathbf{A} & \mathbf{0} & -(\mathbf{M}_{\mathrm{L}} \otimes \mathbf{I}_n)\widehat{\mathbf{W}}_{\mathrm{T}}\mathbf{G} \\ \mathbf{0} & \mathbf{B} & -\mathbf{D}\mathbf{G} \end{bmatrix} \in \mathbb{Z}_q^{(kn+\ell) \times (km+m'+m)}. \tag{4.8}$$

- By construction, for all $i \neq j \in [\ell_1 + \ell_2]$, we have that $\mathbf{A}\mathbf{z}_{i,j} = \mathbf{W}_i \mathbf{t}_j$ and $\mathbf{B}\mathbf{z}_i' = \mathbf{D}\mathbf{t}_i$. For $i \in \{\ell_1 + 1, \ldots, \ell_1 + \ell_2\}$, define the vector

$$\hat{\mathbf{z}}_i = \begin{bmatrix} \mathbf{z}_{1,i} \\ \vdots \\ \mathbf{z}_{\ell_1,i} \end{bmatrix} \in \mathbb{Z}_q^{m\ell_1}.$$

  Then, $(\mathbf{I}_{\ell_1} \otimes \mathbf{A})\hat{\mathbf{z}}_i = \widehat{\mathbf{W}}_{\mathrm{T}}\mathbf{t}_i$. Now we can write

$$(\mathbf{M}_{\mathrm{L}} \otimes \mathbf{I}_n)\widehat{\mathbf{W}}_{\mathrm{T}}\mathbf{t}_i = (\mathbf{M}_{\mathrm{L}} \otimes \mathbf{I}_n)(\mathbf{I}_{\ell_1} \otimes \mathbf{A})\hat{\mathbf{z}}_i = (\mathbf{I}_k \otimes \mathbf{A})(\mathbf{M}_{\mathrm{L}} \otimes \mathbf{I}_m)\hat{\mathbf{z}}_i.$$

  We can now write

$$\underbrace{\begin{bmatrix} \mathbf{I}_k \otimes \mathbf{A} & \mathbf{0} & -(\mathbf{M}_{\mathrm{L}} \otimes \mathbf{I}_n)\widehat{\mathbf{W}}_{\mathrm{T}}\mathbf{G} \\ \mathbf{0} & \mathbf{B} & -\mathbf{D}\mathbf{G} \end{bmatrix}}_{\tilde{\mathbf{B}}} \cdot \underbrace{\begin{bmatrix} (\mathbf{M}_{\mathrm{L}} \otimes \mathbf{I}_m)\hat{\mathbf{z}}_i \\ \mathbf{z}_i' \\ \mathbf{G}^{-1}(\mathbf{t}_i) \end{bmatrix}}_{\tilde{\mathbf{v}}_i} = \mathbf{0}.$$

- When $\mathbf{M}_{\mathrm{L}}$ has small coefficients, $\tilde{\mathbf{v}}_i \in \mathbb{Z}_q^{km'+m'+m}$ is a short vector in the kernel of $\tilde{\mathbf{B}}$. Suppose $\ell_2 \gg km + m' + m$ (e.g., setting $\ell_2 > 2(km + m' + m)$ seems sufficient). We now heuristically assume that the matrix $[\tilde{\mathbf{v}}_{\ell_1+1} \mid \cdots \mid \tilde{\mathbf{v}}_{\ell_1+\ell_2}]$ has full rank over the rationals (but *not* mod q). This seems plausible since the vectors $\hat{\mathbf{z}}_i$, $\mathbf{z}'_i$, and $\mathbf{t}_i$ are all independent. This yields an Ajtai trapdoor (Definition 2.10) for $\tilde{\mathbf{B}}$.
- Using the trapdoor for $\tilde{\mathbf{B}}$, we can sample short $\mathbf{v}_{\mathrm{fc}}, \mathbf{v}_{\mathrm{ext}}, \mathbf{c}'$ such that

$$\tilde{\mathbf{B}} \cdot \begin{bmatrix} \mathbf{v}_{\mathrm{fc}} \\ \mathbf{v}_{\mathrm{ext}} \\ \mathbf{c}' \end{bmatrix} = \begin{bmatrix} \mathbf{I}_k \otimes \mathbf{A} & \mathbf{0} & -(\mathbf{M}_{\mathrm{L}} \otimes \mathbf{I}_n)\widehat{\mathbf{W}}_{\mathrm{T}}\mathbf{G} \\ \mathbf{0} & \mathbf{B} & -\mathbf{D}\mathbf{G} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v}_{\mathrm{fc}} \\ \mathbf{v}_{\mathrm{ext}} \\ \mathbf{c}' \end{bmatrix} = \begin{bmatrix} \mathbf{y} \otimes \mathbf{u} \\ \mathbf{0} \end{bmatrix}.$$

  Then $\mathbf{v} = \left[\begin{smallmatrix} \mathbf{v}_{\mathrm{fc}} \\ \mathbf{v}_{\mathrm{ext}} \end{smallmatrix}\right]$ is a valid opening for the commitment $\mathbf{c} = \mathbf{G}\mathbf{c}' \in \mathbb{Z}_q^n$ to the value $\mathbf{y}$ with respect to $f_{\mathbf{M}}$.

To complete the attack on extractability, we show that if Construction 4.5 satisfies extractability (for sufficiently long inputs ℓ), then the extractor breaks SIS:

- Let $(\mathbf{K}, \mathbf{y})$ where $\mathbf{K} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n \times m}$ and $\mathbf{y} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^n$ be an inhomogeneous SIS challenge. Let $\mathbf{M} = [\mathbf{G}^{-1}(\mathbf{K}) \mid \mathbf{0}^{m \times \ell_2}]$.
- Let $\mathcal{A}$ be the adversary that takes as input $\mathrm{crs} = \left(\mathbf{A}, \mathbf{B}, \mathbf{D}, \mathbf{u}, \{\mathbf{W}_i\}_{i \in [2\ell]}, \{\mathbf{z}_{i,j}\}_{i \neq j}, \{\mathbf{z}'_i\}_{i \in [2\ell]}\right)$ for Construction 4.5 and runs the above algorithm to obtain a commitment $\mathbf{c}$ along with a short opening $\mathbf{v}$ to the value $\mathbf{G}^{-1}(\mathbf{y})$ with respect to the function $\mathbf{M}$. Here, the matrix $\mathbf{M}$ is hard-wired in the description of $\mathcal{A}$.
- If Construction 4.5 is extractable, then there exists an efficient extractor $\mathcal{E}$ that on input crs and outputs a *short* $\mathbf{x} = \left[\begin{smallmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{smallmatrix}\right]$ where $\mathbf{x}_1 \in \mathbb{Z}_q^m$ and $\mathbf{z}_2 \in \mathbb{Z}_q^{\ell_2}$ such that

$$\mathbf{G}^{-1}(\mathbf{y}) = \mathbf{M} \cdot \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = [\mathbf{G}^{-1}(\mathbf{K}) \mid \mathbf{0}^{m \times \ell_2}] \cdot \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \mathbf{G}^{-1}(\mathbf{K}) \cdot \mathbf{x}_1.$$

  Correspondingly, this means that $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{y}) = \mathbf{y} = \mathbf{K}\mathbf{x}_1$, and $\mathcal{E}$ has successfully outputted a solution to the inhomogeneous SIS problem $(\mathbf{K}, \mathbf{y})$.

The approach described above shows that as long as the vector dimension is sufficiently large (i.e., $\ell = \ell_1 + \ell_2 \gg m^2 + m' + 2m$), then the existence of an extractor for Construction 4.5 implies a non-uniform adversary for the inhomogeneous SIS assumption. Note that our approach does rely on a heuristic assumption that the short preimages provided in the CRS span the full space (over the *reals*). This seems like a relatively mild assumption in practice. Assuming this heuristic holds, our analysis shows that under the inhomogeneous SIS assumption, either Assumption 4.2 or Assumption 4.3 must be false, and correspondingly, the functional commitment scheme in Construction 4.5 is *not* extractable.

Acknowledgments

We thank Martin Albrecht for helpful discussions about the cryptanalysis of the k-R-ISIS assumption and Daniel Wichs for helpful insights on functional commitments and RAM delegation. We thank the anonymous reviewers for helpful comments on the presentation. David J. Wu is supported in part by NSF CNS-2151131, CNS-2140975, CNS-2318701, a Microsoft Research Faculty Fellowship, a Google Research Scholar award, and a grant from the Ethereum Foundation.

References

[ABB10a] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, 2010.

[ABB10b] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *CRYPTO*, 2010.

[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.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, 1996.

[Alb23] Martin Albrecht. Knowledge K-M-ISIS is false, 2023. https://gist.github.com/malb/7c8b86520c675560be62eda98dab2a6f.

[AP09] Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. In *STACS*, 2009.

[AR20] Shashank Agrawal and Srinivasan Raghuraman. KVaC: Key-value commitments for blockchains and beyond. In *ASIACRYPT*, 2020.

[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.

[BCI+13] Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Omer Paneth. Succinct non-interactive arguments via linear interactive proofs. In *TCC*, 2013.

[BCTW16] Zvika Brakerski, David Cash, Rotem Tsabary, and Hoeteck Wee. Targeted homomorphic attribute-based encryption. In *TCC*, 2016.

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In *EUROCRYPT*, 2014.

[BHK17] Zvika Brakerski, Justin Holmgren, and Yael Tauman Kalai. Non-interactive delegation and batch NP verification from standard computational assumptions. In *STOC*, 2017.

[BNO21] Dan Boneh, Wilson Nguyen, and Alex Ozdemir. Efficient functional commitments: How to commit to private functions. *IACR Cryptol. ePrint Arch.*, 2021.

[BTVW17] Zvika Brakerski, Rotem Tsabary, Vinod Vaikuntanathan, and Hoeteck Wee. Private constrained PRFs (and more) from LWE. In *TCC*, 2017.

[BV15] Zvika Brakerski and Vinod Vaikuntanathan. Constrained key-homomorphic PRFs from standard lattice assumptions - or: How to secretly embed a circuit in your PRF. In *TCC*, 2015.

[CF13] Dario Catalano and Dario Fiore. Vector commitments and their applications. In *PKC*, 2013.

[CFG+20] Matteo Campanelli, Dario Fiore, Nicola Greco, Dimitris Kolonelos, and Luca Nizzardo. Incrementally aggregatable vector commitments and applications to verifiable decentralized storage. In *ASIACRYPT*, 2020.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT*, 2010.

[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.

[dCP23] Leo de Castro and Chris Peikert. Functional commitments for all functions, with transparent setup and from SIS. In *EUROCRYPT*, 2023.

[DRS04] Yevgeniy Dodis, Leonid Reyzin, and Adam D. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *EUROCRYPT*, 2004.

[FLV23] Ben Fisch, Zeyu Liu, and Psi Vesely. Orbweaver: Succinct linear functional commitments from lattices. In *CRYPTO*, 2023.

[GGPR13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In *EUROCRYPT*, 2013.

[GM18] Nicholas Genise and Daniele Micciancio. Faster gaussian sampling for trapdoor lattices with arbitrary modulus. In *EUROCRYPT*, 2018.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008.

[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.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO*, 2013.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In *STOC*, 2013.

[GVW15a] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In *CRYPTO*, 2015.

[GVW15b] Sergey Gorbunov, Vinod Vaikuntanathan, and Daniel Wichs. Leveled fully homomorphic signatures from standard lattices. In *STOC*, 2015.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4), 1999.

[IKO07] Yuval Ishai, Eyal Kushilevitz, and Rafail Ostrovsky. Efficient arguments without short PCPs. In *CCC*, 2007.

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *STOC*, 1992.

[KLVW23] Yael Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs. Boosting batch arguments and RAM delegation. In *STOC*, 2023.

[KP16] Yael Tauman Kalai and Omer Paneth. Delegating RAM computations. In *TCC*, 2016.

[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*, pages 330–368, 2021.

[KZG10] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In *ASIACRYPT*, 2010.

[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 *Advances in Cryptology - ASIACRYPT 2020, Part III*, 2020.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, 2010.

[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.

[LS15] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.*, 75(3), 2015.

[LW15] Vadim Lyubashevsky and Daniel Wichs. Simple lattice trapdoor sampling from a broad class of distributions. In *PKC*, 2015.

[LY10] Benoît Libert and Moti Yung. Concise mercurial vector commitments and independent zero-knowledge sets with short proofs. In *TCC*, 2010.

[Mer87] Ralph C. Merkle. A digital signature based on a conventional encryption function. In *CRYPTO*, 1987.

[Mic00] Silvio Micali. Computationally sound proofs. *SIAM J. Comput.*, 30(4), 2000.

[Mic02] Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. In *FOCS*, 2002.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, 2012.

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. In *FOCS*, 2004.

[Nit21] Anca Nitulescu. SoK: Vector commitments, 2021. https://www.di.ens.fr/~nitulesc/files/vc-sok.pdf.

[PHGR13] Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. In *IEEE Symposium on Security and Privacy*, 2013.

[PPS21] Chris Peikert, Zachary Pepin, and Chad Sharp. Vector and functional commitments from lattices. In *TCC*, 2021.

[PR06] Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *TCC*, 2006.

[PSTY13] Charalampos Papamanthou, Elaine Shi, Roberto Tamassia, and Ke Yi. Streaming authenticated data structures. In *EUROCRYPT*, 2013.

[TAB+20] Alin Tomescu, Ittai Abraham, Vitalik Buterin, Justin Drake, Dankrad Feist, and Dmitry Khovratovich. Aggregatable subvector commitments for stateless cryptocurrencies. In *SCN*, 2020.

[Tsa22] Rotem Tsabary. Candidate witness encryption from lattice techniques. In *CRYPTO*, 2022.

[TXN20] Alin Tomescu, Yu Xia, and Zachary Newman. Authenticated dictionaries with cross-incremental proof (dis)aggregation. *IACR Cryptol. ePrint Arch.*, 2020.

[VWW22] Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Witness encryption and null-IO from evasive LWE. In *ASIACRYPT*, 2022.

[Wee22] Hoeteck Wee. Optimal broadcast encryption and CP-ABE from evasive lattice assumptions. In *EUROCRYPT*, 2022.

[Wee23] Hoeteck Wee. Circuit ABE with $\mathsf{poly}(\mathsf{depth}, \lambda)$-sized ciphertexts and keys from lattices, 2023. Manuscript.

[WW23] Hoeteck Wee and David J. Wu. Succinct vector, polynomial, and functional commitments from lattices. In *EUROCRYPT*, 2023.

[WWW22] Brent Waters, Hoeteck Wee, and David J. Wu. Multi-authority ABE from lattices without random oracles. In *TCC*, 2022.

A Extending to Module Lattices

Both of our functional commitments (Constructions 3.2 and 3.19) readily translate to work over module lattices [Mic02, LPR10, LS15]. For some parameter settings, this confers asymptotic improvements in the size of the commitment and opening. Specifically, in this section, we show how to adapt Construction 3.2 to obtain a functional commitment for constant-degree polynomials where the size of the commitment and opening are both $O(n \log q)$, where n is the lattice dimension, $q = \mathsf{poly}(\lambda, \ell)$ is the modulus, and ℓ is the input length. As described in Section 3, the size of the commitment in Construction 3.2 is $O(n^2 \log^3 q)$ and the size of the opening is $O(n \log^2 q)$. By working over module lattices and using a larger decomposition base in the gadget matrix (as done also in [ACL+22]), the (asymptotic) size of the commitment and openings of our construction matches those from [ACL+22].

Background on module lattices. Let R be a $\mathbb{Z}$-module of rank t. A common choice for R is to take $t = 2^k$ to be a power-of-two and $R = \mathbb{Z}[x]/(x^t + 1)$ to be the $(2^{k+1})^{\text{th}}$-cyclotomic ring. For an element $r \in R$, we write $\|r\|$ to denote the ℓ_∞-norm of the components of r (viewed as a t-dimensional vector over $\mathbb{Z}$). When $\mathbf{r} = (r_1, \ldots, r_n) \in R^n$ is a vector, we define $\|\mathbf{r}\|$ as the ℓ_∞-norm of the (nd)-dimensional vector formed by concatenating the components of $r_1, \ldots, r_n \in R$. For a modulus $q \in \mathbb{N}$, we write R_q to denote the quotient module $R_q := R/qR$. For parameters $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$, $\beta = \beta(\lambda)$, the module SIS problem over R says that for all efficient adversaries $\mathcal{A}$,

$$\Pr\left[\mathbf{A}\mathbf{x} = \mathbf{0} \text{ and } 0 < \|\mathbf{x}\| \le \beta : \begin{array}{c} \mathbf{A} \xleftarrow{R} R_q^{n \times m}; \\ \mathbf{x} \leftarrow \mathcal{A}(1^\lambda, \mathbf{A}) \end{array}\right] = \mathsf{negl}(\lambda).$$

Observe that the standard SIS problem (Assumption 2.12) corresponds to the case where $t = 1$ (i.e., where $R = \mathbb{Z}$). The ring SIS problem over R corresponds to the case where $n = 1$.

Gadget trapdoors for module lattices. The trapdoor sampling techniques from [MP12] directly extend to the setting of rings (c.f., [MP12, §4.3] and [GM18]). We summarize the relevant results (i.e., the analog of Theorem 2.8) when $R = \mathbb{Z}[x]/(x^t + 1)$ is a power-of-two cyclotomic ring (i.e., $t = 2^k$ for some $k \in \mathbb{N}$) and with respect to an *arbitrary* decomposition base $z \in \mathbb{N}$:

- Let $\mathbf{g}_z = \left[1, z, z^2, \ldots, z^{\lfloor \log_z q \rfloor}\right]$ be the z-ary gadget vector.
- Let λ be a security parameter and let $t = t(\lambda)$ be the module rank, $n = n(\lambda)$ be the dimension, and $q = q(\lambda)$ be a modulus. We require that for all $\lambda \in \mathbb{N}$, $t(\lambda)$ is a power-of-two; then, let $R = \{R_\lambda\}_{\lambda \in \mathbb{N}}$ where R_λ is the power-of-two cyclotomic ring $R_\lambda := \mathbb{Z}[x]/(x^t + 1)$. There exist efficient algorithms $\mathsf{TrapGen}_R$ and $\mathsf{SamplePre}_R$ with the following syntax:
  - $\mathsf{TrapGen}_R(1^\lambda, m) \to (\mathbf{A}, \mathbf{R})$: On input the security parameter λ (which defines the module rank t, the dimension n, and the modulus q), and the number of samples m, the trapdoor-generation algorithm outputs a matrix $\mathbf{A} \in R_q^{n \times m}$ and a trapdoor $\mathbf{R} \in R_q^{m \times m}$.
  - $\mathsf{SamplePre}_R(\mathbf{A}, \mathbf{R}, \mathbf{v}, \chi) \to \mathbf{u}$: On input a matrix $\mathbf{A} \in R_q^{n \times m}$, a trapdoor $\mathbf{R} \in R_q^{m \times m}$, a target vector $\mathbf{v} \in R_q^n$, and a Gaussian width parameter χ, the preimage-sampling algorithm outputs a vector $\mathbf{u} \in R_q^m$.

  Suppose $m \ge O(n \log_z q)$. Then, these algorithms satisfy a similar set of properties as in Theorem 2.8:
  - **Trapdoor distribution:** If $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^\lambda, m)$ and $\mathbf{A}' \xleftarrow{R} R_q^{n \times m}$, then $\Delta(\mathbf{A}, \mathbf{A}') \le \mathsf{negl}(\lambda)$. Moreover, $\mathbf{A}\mathbf{R} = \mathbf{G}_z = (\mathbf{I}_n \otimes \mathbf{g}_z)$ and $\|\mathbf{R}\| = O(z)$.

– **Preimage sampling:** For all matrices $R \in R_q^{m\times m}$, parameters $\chi > 0$, and all target vectors $\mathbf{v} \in R_q^n$ in the column span of $\mathbf{A}$, the output $\mathbf{u} \leftarrow \mathsf{SamplePre}_R(\mathbf{A}, \mathbf{R}, \mathbf{v}, \chi)$ of SamplePre satisfies $\mathbf{Au} = \mathbf{v}$.

– **Preimage distribution:** Suppose $\mathbf{AR} = \mathbf{G}_z$. There exists a fixed polynomial $\mathsf{poly}(\cdot,\cdot,\cdot)$ such that for all $\chi \geq \mathsf{poly}(\lambda, m, \|\mathbf{R}\|)$ and all target vectors $\mathbf{v} \in R_q^n$, the statistical distance between the following distributions is $\mathsf{negl}(\lambda)$:

$$\{\mathbf{u} \leftarrow \mathsf{SamplePre}_R(\mathbf{A}, \mathbf{R}, \mathbf{v}, \chi)\} \quad \text{and} \quad \{\mathbf{u} \leftarrow \mathbf{A}_\chi^{-1}(\mathbf{v})\},$$

where $\mathbf{A}_\chi^{-1}(\mathbf{v})$ denotes sampling a vector $\mathbf{u}$ according to discrete Gaussian distribution over R^m with parameter χ subject to the condition that $\mathbf{Au} = \mathbf{v} \in R_q^n$.

Functional commitment over module lattices. We now describe our adaptation of Construction 3.2 over module lattices. We essentially replace the ring $\mathbb{Z}$ with the $\mathbb{Z}$-module R in Construction 3.2. For completeness, we provide the full description here.

Construction A.1 (Functional Commitment over Module Lattices)**.** Let λ be a security parameter. We now define the following scheme parameters:

- Let $t = t(\lambda)$, $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$, $\chi = \chi(\lambda)$ be lattice parameters. We assume that for all $\lambda \in \mathbb{N}$, $t(\lambda)$ is a power of two. The scheme operates over a $\mathbb{Z}$-module $R = \{R_\lambda\}_{\lambda\in\mathbb{N}}$ where each $R_\lambda = \mathbb{Z}[x]/(x^t + 1)$ is a power-of-two cyclotomic ring.
- Let $z = z(\lambda)$ be a decomposition base.
- Let $\ell = \ell(\lambda)$ be an input length parameter, $d_{\max} = O(1)$ be a *constant* degree bound, $B_{\mathsf{in}} = B_{\mathsf{in}}(\lambda)$ be a bound on the magnitude of the inputs, and $B_{\mathsf{out}} = B_{\mathsf{out}}(\lambda)$ be a bound on the magnitude of the outputs.
- Let $L = \sum_{i\in[d_{\max}]} \ell^i$ and $B = B(\lambda)$ be a verification bound.
- Let $\mathcal{F}_\lambda$ be the set of functions $f\colon [-B_{\mathsf{in}}, B_{\mathsf{in}}]^\ell \to [-B_{\mathsf{out}}, B_{\mathsf{out}}]$ where f can be computed by a *homogeneous* polynomial with B_{in}-bounded coefficients and degree at most $d_{\max}$. As in Construction 3.2, we associate a function $f \in \mathcal{F}_\lambda$ with a vector $\mathbf{f} \in [-B_{\mathsf{in}}, B_{\mathsf{in}}]^{\ell^d}$ for some $d \leq d_{\max}$ and define $f(\mathbf{x}) := \mathbf{f}^\intercal \mathbf{x}^{\otimes d}$.

We construct a functional commitment $\Pi_{\mathsf{FC}} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Eval}, \mathsf{Verify})$ for $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ as follows:

- $\mathsf{Setup}(1^\lambda)$: On input the security parameter λ, the setup algorithm samples $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}_R(1^\lambda, m)$ and $\mathbf{W} \xleftarrow{\mathrm{R}} R_q^{Ln\times m}$. Next, define the target matrix

$$\mathbf{P} = \begin{bmatrix} \mathbf{I}_\ell \otimes \mathbf{G}_z \\ \mathbf{I}_\ell \otimes \mathbf{W}_1 \\ \vdots \\ \mathbf{I}_\ell \otimes \mathbf{W}_{d_{\max}-1} \end{bmatrix} \in R_q^{Ln\times \ell m} \quad \text{where} \quad \mathbf{W} = \begin{bmatrix} \mathbf{W}_1 \\ \vdots \\ \mathbf{W}_{d_{\max}} \end{bmatrix} \in R_q^{Ln\times m}, \tag{A.1}$$

  where $\mathbf{W}_i \in R_q^{\ell^i n\times m}$. Then, compute $\mathbf{T} \leftarrow \mathsf{SamplePre}_R([\mathbf{I}_L \otimes \mathbf{A} \mid \mathbf{W}], \mathbf{I}_L \otimes \mathbf{R}, \mathbf{P}, \chi) \in R_q^{(Lm+m)\times \ell m}$. Parse $\mathbf{T} = \begin{bmatrix} \mathbf{T}_{\mathsf{open}} \\ \mathbf{T}_{\mathsf{com}} \end{bmatrix}$ where $\mathbf{T}_{\mathsf{open}} \in R_q^{Lm\times \ell m}$ and $\mathbf{T}_{\mathsf{com}} \in R_q^{m\times \ell m}$. Output the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}})$.
- $\mathsf{Commit}(\mathsf{crs}, \mathbf{x})$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}})$ and an input $\mathbf{x} \in [-B_{\mathsf{in}}, B_{\mathsf{in}}]^\ell$, the commit algorithm outputs the commitment $\sigma = \mathbf{C} = \mathbf{T}_{\mathsf{com}}(\mathbf{x} \otimes \mathbf{I}_m) \in R_q^{m\times m}$ and the state $\mathsf{st} = \mathbf{x}$.
- $\mathsf{Eval}(\mathsf{crs}, \mathsf{st}, f)$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}})$, the state $\mathsf{st} = \mathbf{x}$, and a function $f = \mathbf{f} \in \mathbb{Z}_q^{\ell^d}$ (for some $d \leq d_{\max}$) with B_{in}-bounded coefficients, the evaluation algorithm first computes $\mathbf{V} = \mathbf{T}_{\mathsf{open}}(\mathbf{x} \otimes \mathbf{I}_m)$. It then parses

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1 \\ \vdots \\ \mathbf{V}_{d_{\max}} \end{bmatrix} \in R_q^{Lm\times m} \tag{A.2}$$

where $\mathbf{V}_i \in R_q^{\ell^i m \times m}$. Let $\mathbf{V}'_1 \leftarrow \mathbf{V}_1$ and for $i \in [d]$, let $\mathbf{V}'_i \leftarrow (\mathbf{x} \otimes \mathbf{I}_{\ell^{i-1}m})\mathbf{V}'_{i-1} + \mathbf{V}_i \mathbf{C}^{i-1} \in R_q^{\ell^i m \times m}$. Output the opening $\pi_f = \mathbf{V}_f = (\mathbf{f}^\intercal \otimes \mathbf{I}_m)\mathbf{V}'_d \in R_q^{m \times m}$.

- $\mathsf{Verify}(\mathsf{crs}, \sigma, f, y, \pi)$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}})$, the commitment $\sigma = \mathbf{C} \in R_q^{m \times m}$, the output $y \in [-B_{\mathsf{out}}, B_{\mathsf{out}}]$, a function $f = \mathbf{f} \in \mathbb{Z}_q^{\ell^d}$ (for some $d \le d_{\max}$) with B_{in}-bounded coefficients, and the proof $\pi = \mathbf{V} \in R_q^{m \times m}$, the verification algorithm first parses $\mathbf{W}$ into $\mathbf{W}_1, \ldots, \mathbf{W}_{d_{\max}}$ as in Eq. (3.1) and outputs 1 if
$$\|\mathbf{V}\| \le B \quad \text{and} \quad (\mathbf{f}^\intercal \otimes \mathbf{I}_m)\mathbf{W}_d \mathbf{C}^d = y \cdot \mathbf{G} - \mathbf{A}\mathbf{V}. \tag{A.3}$$

Correctness and security analysis. Correctness and security follow via a similar analysis as in the proofs of Theorems 3.5 and 3.6. Binding in this case relies on the L-succinct SIS assumption over the module $R = \{R_\lambda\}_{\lambda \in \mathbb{N}}$: namely, for all efficient adversaries $\mathcal{A}$,

$$\Pr\left[\mathbf{A}\mathbf{x} = \mathbf{0} \text{ and } 0 < \|\mathbf{x}\| \le \beta : \begin{array}{c} \mathbf{A} \xleftarrow{\mathrm{R}} R_q^{n \times m}, \mathbf{W} \xleftarrow{\mathrm{R}} R_q^{n\ell \times m}, \\ \mathbf{R} \leftarrow [\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}]_\chi^{-1}(\mathbf{G}_{n\ell}) \\ \mathbf{x} \leftarrow \mathcal{A}(1^\lambda, \mathbf{A}, \mathbf{W}, \mathbf{R}) \end{array}\right] = \mathsf{negl}(\lambda).$$

Similar to the case over the integers, we consider instantiations of the lattice parameters (R, n, m, q, β) where the module SIS assumption over R holds. To satisfy correctness and security for constant-degree polynomials and polynomially-bounded inputs and outputs (i.e., $d_{\max} = O(1)$ and $B_{\mathsf{in}}, B_{\mathsf{out}} = \mathsf{poly}(\lambda)$), it suffices to set $m \ge O(n \log_z q)$, $\chi_0 \ge \mathsf{poly}(\lambda, m, z)$, $\chi \ge \mathsf{poly}(\lambda, m, z)$, $B \ge \mathsf{poly}(\lambda, m, \ell, z)$, and $\beta \ge \mathsf{poly}(\lambda, m, \ell, z)$.

Parameter instantiation. We now describe one way to instantiate the above construction using module lattices. Let λ be a security parameter, ℓ be the input dimension, and $d_{\max} = O(1)$ be a degree bound. For simplicity, we consider the setting where the input and output magnitudes are both polynomially-bounded: namely, $B_{\mathsf{in}} = \mathsf{poly}(\lambda)$ and $B_{\mathsf{out}} = \mathsf{poly}(\lambda)$. Consider the following instantiation of the the lattice parameters in the above construction:

- Let $R = \mathbb{Z}[x]/(x^{2^k} + 1)$ be a power-of-two cyclotomic ring. Let $t = 2^k$ be the rank of R when viewed as a $\mathbb{Z}$-module. We set the dimension n such that $nt = \tilde{O}(\lambda)$.
- We set $m = O(n \log_z q)$.
- We set $\chi_0, \chi = \mathsf{poly}(\lambda, m, L, z) = \mathsf{poly}(\lambda, m, \ell, z)$ since by definition, $L = O(\ell^{d_{\max}}) = \mathsf{poly}(\ell)$ for constant $d_{\max}$.
- We set the bound $B = \mathsf{poly}(\lambda, m, \ell, z)$ and $\beta = \mathsf{poly}(\lambda, m, \ell, z)$.
- We choose the decomposition base so that $z \ge q^{1/c}$ for a (sufficiently-large) constant $c \in \mathbb{N}$. In particular, the constant c must be larger than the exponent on z in β.
- We set the modulus $q \ge \beta \cdot \mathsf{poly}(n, t) = \beta \cdot \mathsf{poly}(\lambda)$ so that the L-succinct module SIS assumption (over R) holds. In particular, we can set $q = \mathsf{poly}(\lambda, m, \ell)$.

With this choice of parameters, $m = O(n \log_z q) = O(n \log q / \log z) = O(n)$, since $c \in \mathbb{N}$ is a constant. Hardness relies on (quasi)-exponential hardness of L-succinct SIS (i.e., for $n = n(\lambda)$ and $t = t(\lambda)$ where $nt \ge \lambda$ and all adversaries running in time at most $2^{\tilde{\Omega}(\lambda)}$, the advantage is bounded by a negligible function $\mathsf{negl}(\lambda)$). With this setting of parameters, we obtain a functional commitment scheme for constant-degree polynomials with the following parameter sizes:

- **Commitment size:** A commitment σ to an input $\mathbf{x} \in [-B_{\mathsf{in}}, B_{\mathsf{in}}]^\ell$ is a matrix $\sigma = \mathbf{C} \in R_q^{m \times m}$, so
$$|\sigma| = m^2 \cdot (t \log q) = O(n^2 t \log q).$$
- **Opening size:** An opening π to a function f consists of a vector $\pi = \mathbf{v}_f \in R_q^m$ so
$$|\pi| = m \cdot (t \log q) = O(nt \log q).$$

We now obtain the following instantiations:

- **Integer lattices:** If we consider integer lattices (i.e., $R = \mathbb{Z}$), then $t = 1$ and $n = \tilde{O}(\lambda)$. In this case, the above parameter instantiations yield commitments of size $\tilde{O}(\lambda^2 \log \ell)$ and openings of size $\tilde{O}(\lambda \log \ell)$.
- **Module lattices:** If we consider module lattices, we set $t = \tilde{O}(\lambda)$ and $n = O(1)$. This yields commitments and openings with size $\tilde{O}(\lambda \log \ell)$.

B Functional Commitments for Circuits from ℓ-Succinct SIS

In this section, we show how to adapt the functional commitment scheme from [WW23] to rely on the ℓ-succinct SIS assumption (Assumption 3.1) instead of the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption. We start with an informal overview of the construction from [WW23] and then describe the simple tweak that allows us to base security on the less-structured ℓ-succinct SIS assumption. We specifically consider the "alternative" version described in [WW23, Remark 4.13]:

- The CRS contains $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, a matrix $\tilde{\mathbf{W}} \xleftarrow{R} \mathbb{Z}_q^{n\ell \times n}$ and a trapdoor $\mathbf{T}$ for $\mathbf{B}_\ell := [\mathbf{I}_\ell \otimes \mathbf{A} \mid \tilde{\mathbf{W}}\mathbf{G}]$. We will often parse
$$\tilde{\mathbf{W}} = \begin{bmatrix} \tilde{\mathbf{W}}_1 \\ \vdots \\ \tilde{\mathbf{W}}_\ell \end{bmatrix} \quad \text{where} \quad \tilde{\mathbf{W}}_i \in \mathbb{Z}_q^{n \times n}.$$
- To commit to an input $\mathbf{x}$, the committer samples
$$\begin{bmatrix} \mathbf{V}_1 \\ \vdots \\ \mathbf{V}_\ell \\ \hat{\mathbf{C}} \end{bmatrix} \leftarrow \mathbf{B}_\ell^{-1}(-\mathbf{x} \otimes \mathbf{G}).$$
The commitment is $\mathbf{C} = \mathbf{G}\hat{\mathbf{C}}$. By construction, for all $i \in [\ell]$, $\mathbf{A}\mathbf{V}_i = -\tilde{\mathbf{W}}_i\mathbf{C} - x_i\mathbf{G}$. Let $\tilde{\mathbf{C}} = [-\tilde{\mathbf{W}}_1\mathbf{C} \mid \cdots \mid -\tilde{\mathbf{W}}_\ell\mathbf{C}]$ and $\tilde{\mathbf{V}} = [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_\ell]$. Then,
$$\mathbf{A}\tilde{\mathbf{V}} = \tilde{\mathbf{C}} - \mathbf{x}^\intercal \otimes \mathbf{G}.$$
- The opening for a function f is $\tilde{\mathbf{V}}_f := \tilde{\mathbf{V}}\mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}}$. To check the opening, the verifier computes $\tilde{\mathbf{C}}_f$ from $\tilde{\mathbf{C}}$ and checks that $\tilde{\mathbf{V}}_f$ is short and moreover, $\mathbf{A}\tilde{\mathbf{V}}_f = \tilde{\mathbf{C}}_f - f(\mathbf{x}) \cdot \mathbf{G}$. Correctness now follows from Theorem 2.13:
$$\mathbf{A}\tilde{\mathbf{V}}_f = \mathbf{A}\tilde{\mathbf{V}}\mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}} = (\tilde{\mathbf{C}} - \mathbf{x}^\intercal \otimes \mathbf{G}) \cdot \mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}} = \tilde{\mathbf{C}}_f - f(\mathbf{x}) \cdot \mathbf{G}.$$

Binding then follows from the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption. In particular, in this setting, the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption essentially asserts that SIS is hard with respect to $\mathbf{A}$ even given a trapdoor for the structured matrix $[\mathbf{I}_\ell \otimes \mathbf{A} \mid \tilde{\mathbf{W}}\mathbf{G}]$.

Basing security on ℓ-succinct SIS. The observation in this work is we can replace $\tilde{\mathbf{W}}\mathbf{G}$ in the above construction with a *uniform* matrix $\mathbf{W} \xleftarrow{R} \mathbb{Z}_q^{n\ell \times m}$; that is, $\mathbf{B}_\ell := [\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}]$. To commit to an input $\mathbf{x}$, the committer still samples $\mathbf{B}_\ell^{-1}(-\mathbf{x} \otimes \mathbf{G})$. The only difference is now the commitment $\mathbf{C}$ is the full preimage (instead of $\mathbf{G}\hat{\mathbf{C}}$ as before). This leads to slight larger commitments (by a $\log q$ factor); see Remark B.8. We now give the full construction and analysis (which closely follows the corresponding analysis in [WW23]):

Construction B.1 (Succinct Functional Commitment)**.** Let λ be a security parameter and $n = n(\lambda)$, $m = m(\lambda)$, and $q = q(\lambda)$ be lattice parameters where q is prime. Let $B = B(\lambda)$ be a bound. Let $\chi_0 = \chi_0(\lambda)$, $\chi_1 = \chi_1(\lambda)$ be Gaussian width parameters. Let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ be a family of Boolean valued functions $f\colon \{0,1\}^\ell \to \{0,1\}$ where each function $f\colon \{0,1\}^\ell \to \{0,1\}$ is a function on inputs of length $\ell = \ell(\lambda)$ and which can be computed by a Boolean circuit of depth at most $d = d(\lambda)$. We construct a functional commitment $\Pi_{\mathsf{VC}} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Verify})$ for $\mathcal{F}$ as follows:

- $\mathsf{Setup}(1^\lambda)$: On input the security parameter λ, the setup algorithm samples $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$, $\mathbf{W} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{\ell n \times m}$ and $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}], \mathbf{I}_\ell \otimes \mathbf{R}, \mathbf{G}_{n\ell}, \chi_0)$. Finally, it outputs $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T})$.
- $\mathsf{Commit}(\mathsf{crs}, \mathbf{x})$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T})$ and a vector $\mathbf{x} \in \{0,1\}^\ell$, the commit algorithm samples a preimage

$$\begin{bmatrix} \mathbf{V}_1 \\ \vdots \\ \mathbf{V}_\ell \\ \mathbf{C} \end{bmatrix} \leftarrow \mathsf{SamplePre}([\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}], \mathbf{T}, -\mathbf{x} \otimes \mathbf{G}_n, \chi_1). \tag{B.1}$$

  It outputs the commitment $\sigma = \mathbf{C} = \mathbb{Z}_q^{m \times m}$ and the state $\mathsf{st} = (\mathbf{x}, \mathbf{C}, \mathbf{V}_1, \ldots, \mathbf{V}_\ell)$.
- $\mathsf{Eval}(\mathsf{crs}, \mathsf{st}, f)$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T})$, a commitment state $\mathsf{st} = (\mathbf{x}, \mathbf{C}, \mathbf{V}_1, \ldots, \mathbf{V}_\ell)$, and a function $f\colon \{0,1\}^\ell \to \{0,1\}$, the evaluation algorithm parses

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 \\ \vdots \\ \mathbf{W}_\ell \end{bmatrix} \quad \text{where } \mathbf{W}_i \in \mathbb{Z}_q^{n \times m} \tag{B.2}$$

  and sets $\tilde{\mathbf{C}} \leftarrow [-\mathbf{W}_1\mathbf{C} \mid \cdots \mid -\mathbf{W}_\ell\mathbf{C}]$. It then computes $\mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}} \leftarrow \mathsf{EvalFX}(\tilde{\mathbf{C}}, f, \mathbf{x})$, and outputs the opening $\pi_f = \mathbf{V}_f \leftarrow [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_\ell] \cdot \mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}} \in \mathbb{Z}_q^{m \times m}$.
- $\mathsf{Verify}(\mathsf{crs}, \sigma, f, y, \pi)$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T})$, a commitment $\sigma = \mathbf{C} \in \mathbb{Z}_q^{m \times m}$, a function $f\colon \{0,1\}^\ell \to \{0,1\}$, a value $y \in \{0,1\}$, and an opening $\pi = \mathbf{V}_f \in \mathbb{Z}_q^{m \times m}$, the verification algorithm parses $\mathbf{W}$ into $\mathbf{W}_1, \ldots, \mathbf{W}_\ell$ according to Eq. (B.2), computes $\tilde{\mathbf{C}} \leftarrow [-\mathbf{W}_1\mathbf{C} \mid \cdots \mid -\mathbf{W}_\ell\mathbf{C}]$, $\tilde{\mathbf{C}}_f \leftarrow \mathsf{EvalF}(\tilde{\mathbf{C}}, f)$, and outputs 1 if

$$\left\|\mathbf{V}_f\right\| \le B \quad \text{and} \quad \mathbf{A}\mathbf{V}_f = \tilde{\mathbf{C}}_f - y\mathbf{G}. \tag{B.3}$$

Correctness and security. The correctness and security analysis of Construction B.1 follow via the same template as in [WW23]. We include the analysis here for completeness:

Theorem B.2 (Correctness). *Suppose $n \ge \lambda$, $m \ge O(n \log q)$, $\chi_0 \ge O(\ell m \log(n\ell))$, $\chi_1 \ge O(\ell^{3/2} m^{3/2} \log(n\ell) \cdot \chi_0)$ and $B \ge m\sqrt{m(\ell+1)} \cdot (n \log q)^{O(d)} \cdot \chi_1$. Then, Construction B.1 is correct.*

Proof. Take a security parameter λ, a function $f \in \mathcal{F}_\lambda$, and an input $\mathbf{x} \in \{0,1\}^\ell$. Let $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}) \leftarrow \mathsf{Setup}(1^\lambda)$ and $(\sigma, \mathsf{st}) \leftarrow \mathsf{Commit}(\mathsf{crs}, \mathbf{x})$ where $\sigma = \mathbf{C} \in \mathbb{Z}_q^{m \times m}$ and $\mathsf{st} = (\mathbf{x}, \mathbf{C}, \mathbf{V}_1, \ldots, \mathbf{V}_\ell)$. Let $\pi = \mathbf{V}_f \leftarrow \mathsf{Eval}(\mathsf{crs}, \mathsf{st}, f)$ and consider $\mathsf{Verify}(\mathsf{crs}, \sigma, f, f(\mathbf{x}), \pi)$:

- By Theorem 2.8 and Lemma 2.5, for $m \ge O(n \log q)$ and $\chi_0 \ge m(\ell+1) \cdot \omega(\sqrt{\log n\ell}) = O(\ell m \log(n\ell))$. then $[\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}]\mathbf{T} = \mathbf{G}_{n\ell}$ and $\|\mathbf{T}\| \le \sqrt{m(\ell+1)}\chi_0$ with overwhelming probability.
- Suppose $\chi_1 \ge m(\ell+1)\|\mathbf{T}\| \cdot \omega(\sqrt{\log(n\ell)}) = O(\ell^{3/2} m^{3/2} \log(n\ell) \cdot \chi_0)$. By construction of $(\mathbf{V}_1, \ldots, \mathbf{V}_\ell, \mathbf{C})$,

$$\mathbf{A}\mathbf{V}_i + \mathbf{W}_i\mathbf{C} = -x_i\mathbf{G}.$$

  Let $\tilde{\mathbf{C}} = [-\mathbf{W}_1\mathbf{C} \mid \cdots \mid -\mathbf{W}_\ell\mathbf{C}]$ and $\tilde{\mathbf{V}} = [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_\ell]$. Then,

$$\tilde{\mathbf{C}} - \mathbf{x}^\intercal \otimes \mathbf{G} = \mathbf{A}[\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_\ell] = \mathbf{A}\tilde{\mathbf{V}}. \tag{B.4}$$

  Let $B_0 = \sqrt{m(\ell+1)} \cdot \chi_1$ be the "initial" noise bound. By Lemma 2.5, $\|\mathbf{V}_i\| \le \sqrt{m(\ell+1)}\chi_1 = B_0$ and so $\|\tilde{\mathbf{V}}\| \le B_0$.
- By construction of Eval, we have that $\mathbf{V}_f = \tilde{\mathbf{V}} \cdot \mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}}$ where $\mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}} \leftarrow \mathsf{EvalFX}(\tilde{\mathbf{C}}, f, \mathbf{x})$. By Theorem 2.13, $\left\|\mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}}\right\| \le (n \log q)^{O(d)}$, so $\|\mathbf{V}_f\| \le m \cdot B_0 \cdot (n \log q)^{O(d)} \le \chi_1 \cdot m\sqrt{m(\ell+1)} \cdot (n \log q)^{O(d)}$.

- Again appealing to Theorem 2.13 and Eq. (B.4), we can write

$$\mathbf{A}\mathbf{V}_f = \mathbf{A}\tilde{\mathbf{V}}\mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}} = (\tilde{\mathbf{C}} - \mathbf{x}^\intercal \otimes \mathbf{G}) \cdot \mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}} = \tilde{\mathbf{C}}_f - f(\mathbf{x}) \cdot \mathbf{G},$$

  where $\tilde{\mathbf{C}} = \mathsf{EvalF}(\tilde{\mathbf{C}}, f)$. Correspondingly, $\mathsf{Verify}(\mathsf{crs}, \sigma, f, f(\mathbf{x}), \pi)$ outputs 1. □

Theorem B.3 (Binding). *Suppose $n \geq \lambda$, $m \geq O(n \log q)$, $\chi_0 \geq O(\ell m \log(n\ell))$, and $\beta \geq 2Bm^{3/2} \log n$. Then, under the ℓ-succinct SIS assumption with parameters (n, m, q, χ_0, β), Construction B.1 is computationally binding.*

Proof. We proceed via a hybrid argument:

- Hyb_0: This is the real binding experiment:
  - The challenger samples $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$, $\mathbf{W} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{\ell n \times m}$, and $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}], \mathbf{I}_\ell \otimes \mathbf{R}, \mathbf{G}_{n\ell}, \chi_0)$ and gives $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T})$ to the adversary $\mathcal{A}$. Let $\mathbf{W}_1, \ldots, \mathbf{W}_\ell \in \mathbb{Z}_q^{n \times m}$ be the components of $\mathbf{W}$ according to Eq. (B.2).
  - Algorithm $\mathcal{A}$ outputs a commitment $\mathbf{C} \in \mathbb{Z}_q^{m \times m}$, a function $f \in \mathcal{F}_\lambda$, and openings $\mathbf{V}_0, \mathbf{V}_1 \in \mathbb{Z}_q^{m \times m}$.
  - The output of the experiment is 1 if $\|\mathbf{V}_0\|, \|\mathbf{V}_1\| \leq B$, $\mathbf{A}\mathbf{V}_0 = \tilde{\mathbf{C}}_f$, and $\mathbf{A}\mathbf{V}_1 = \tilde{\mathbf{C}}_f - \mathbf{G}$, where $\tilde{\mathbf{C}}_f \leftarrow \mathsf{EvalF}(\tilde{\mathbf{C}}, f)$, and $\tilde{\mathbf{C}} = [-\mathbf{W}_1\mathbf{C} \mid \cdots \mid -\mathbf{W}_\ell\mathbf{C}]$. Otherwise, the experiments outputs 0.
- Hyb_1: Same as Hyb_0 except after constructing the matrix the challenger samples $\mathbf{T} \leftarrow [\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}]_{\chi_0}^{-1}(\mathbf{G}_{n\ell})$ *without* using the trapdoor $\mathbf{R}$. The CRS is now independent of $\mathbf{R}$.
- Hyb_2: Same as Hyb_1 except the challenger samples $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n \times m}$.

For an adversary $\mathcal{A}$, we write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output of an execution of Hyb_i with adversary $\mathcal{A}$. We now show that each adjacent pair of experiments are computationally indistinguishable.

Lemma B.4. *Suppose $n \geq \lambda$, $m \geq O(n \log q)$, and $\chi_0 \geq O(\ell m \log(n\ell))$. Then, for all adversaries $\mathcal{A}$, $\mathsf{Hyb}_0(\mathcal{A}) \overset{s}{\approx} \mathsf{Hyb}_1(\mathcal{A})$.*

Proof. The only difference between Hyb_0 and Hyb_1 is the distribution of $\mathbf{T}$. In Hyb_0, the challenger samples $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}], \mathbf{I}_\ell \otimes \mathbf{R}, \mathbf{G}_{n\ell}, \chi_0)$. By Theorem 2.8, for $m \geq O(n \log q)$ and $\chi_0 \geq m(\ell + 1) \cdot \omega(\sqrt{\log n\ell}) = O(\ell m \log(n\ell))$, the distribution of $\mathbf{T}$ is statistically close to the distribution $[\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}]_{\chi_0}^{-1}(\mathbf{G}_{n\ell})$, which is the distribution of $\mathbf{T}$ in Hyb_1. □

Lemma B.5. *Suppose $n \geq \lambda$ and $m \geq O(n \log q)$. Then, for all adversaries $\mathcal{A}$, $\mathsf{Hyb}_1(\mathcal{A}) \overset{s}{\approx} \mathsf{Hyb}_2(\mathcal{A})$.*

Proof. The only difference between Hyb_1 and Hyb_2 is the distribution of $\mathbf{A}$. In Hyb_1, the challenger samples $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$. By Theorem 2.8, the distribution of $\mathbf{A}$ is statistically close to $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n \times m}$. □

Lemma B.6. *Suppose $\beta \geq 2Bm^{3/2} \log n$. Under the ℓ-succinct SIS assumption with parameters (n, m, q, χ_0, β), for all efficient adversaries $\mathcal{A}$, $\Pr[\mathsf{Hyb}_2(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$.*

Proof. Suppose there exists an efficient adversary $\mathcal{A}$ where $\Pr[\mathsf{Hyb}_2(\mathcal{A}) = 1] = \varepsilon$ for some non-negligible ε. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ for the ℓ-succinct SIS assumption:

1. Algorithm $\mathcal{B}$ receives a challenge $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{W} \in \mathbb{Z}_q^{n\ell \times m}$, and $\mathbf{T} \in \mathbb{Z}_q^{m(\ell+1) \times \ell m}$. Algorithm $\mathcal{B}$ gives $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T})$ to $\mathcal{A}$.
2. Algorithm $\mathcal{A}$ outputs a commitment $\mathbf{C} \in \mathbb{Z}_q^{n \times m}$, a function $f \in \mathcal{F}_\lambda$, and openings $\mathbf{V}_0, \mathbf{V}_1 \in \mathbb{Z}_q^{m \times m}$.
3. Algorithm $\mathcal{B}$ outputs $\mathbf{x} \leftarrow \mathsf{SamplePre}(\mathbf{A}, \mathbf{V}_0 - \mathbf{V}_1, \mathbf{0}, s')$ where $s' = 2Bm \log n$.

By construction, algorithm $\mathcal{B}$ perfectly simulates the common reference string according to the specification of Hyb_2. Thus, with probability ε, $\|\mathbf{V}_0\|, \|\mathbf{V}_1\| \leq B$, $\mathbf{A}\mathbf{V}_0 = \tilde{\mathbf{C}}_f$, $\mathbf{A}\mathbf{V}_1 = \tilde{\mathbf{C}}_f - \mathbf{G}$. This means that $\mathbf{A}(\mathbf{V}_0 - \mathbf{V}_1) = \mathbf{G}$, so $\mathbf{V}_0 - \mathbf{V}_1$ is a trapdoor for $\mathbf{A}$. By Theorem 2.8, the distribution of $\mathbf{x}$ is statistically close to $\mathbf{A}_{s'}^{-1}(\mathbf{0})$. By Lemma 2.6, $\mathbf{x}$ is non-zero with probability $1 - \mathsf{negl}(n)$. Finally, by Lemma 2.5, $\|\mathbf{x}\| \leq \sqrt{m}s' = \beta$, and the claim holds. □

Combining Lemmas B.4 to B.6, the functional commitment scheme is computationally binding. □

Parameter instantiations. We can instantiate the parameters for Construction B.1 as in [WW23]. Specifically, let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of functions $f\colon \{0,1\}^\ell \to \{0,1\}$ on inputs of length $\ell = \ell(\lambda)$ and which can be computed by Boolean circuits of depth at most $d = d(\lambda)$. We instantiate the parameters in Construction B.1 as follows:

- Let $\varepsilon > 0$ be a constant. We set the lattice dimension $n = d^{1/\varepsilon} \cdot \mathsf{poly}(\lambda)$ and $m = O(n \log q)$.
- We set $\chi_0 = O(\ell m \log(n\ell))$ and $\chi_1 = O(\ell^{3/2} m^{3/2} \log(n\ell) \cdot \chi_0) = O(\ell^{5/2} m^{5/2} \log^2(n\ell))$.
- We set the bound $B = \chi_1 \cdot m\sqrt{m(\ell+1)} \cdot (n \log q)^{O(d)} = \ell^3 \log^2 \ell \cdot (n \log q)^{O(d)}$.
- We set the modulus q so that the ℓ-succinct SIS assumption holds with parameters (n, m, q, χ_0, β), where

$$\beta = 2Bm^{3/2} \log n = \ell^3 \log^2 \ell \cdot (n \log q)^{O(d)} = 2^{\tilde{O}(d)} = 2^{\tilde{O}(n^\varepsilon)},$$

  where we write $\tilde{O}(\cdot)$ to suppress polylogarithmic factors in λ, d, and ℓ. With this instantiation, $\log q = \mathsf{poly}(d^{1/\varepsilon}, \log\lambda, \log\ell)$, and we are relying on ℓ-succinct SIS with *sub-exponential* noise bound.

With this setting of parameters, we obtain a functional commitment scheme for $\mathcal{F}$ with the following properties:

Corollary B.7 (Succinct Vector Commitment from ℓ-Succinct SIS)**.** *Let λ be a security parameter, and let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of functions $f\colon \{0,1\}^\ell \to \{0,1\}$ on inputs of length $\ell = \ell(\lambda)$ and which can be computed by Boolean circuits of depth at most $d = d(\lambda)$. Under the ℓ-succinct SIS assumption with a sub-exponential norm bound $\beta = 2^{\tilde{O}(1/\varepsilon)}$ for some constant $\varepsilon > 0$ and lattice dimension $n = n(\lambda)$, there exists a succinct functional commitment scheme for $\mathcal{F}$. Both the size of the commitment and the opening are $\mathsf{poly}(\lambda, d^{1/\varepsilon}, \log\ell)$, and the CRS has size $\ell^2 \cdot \mathsf{poly}(\lambda, d^{1/\varepsilon}, \log\ell)$.*

Remark B.8 (Comparison with [WW23])**.** As noted above, Construction B.1 is identical to [WW23, Remark 4.13] except the CRS contains a trapdoor for the matrix $[\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{W}]$ where $\mathbf{W} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n\ell\times m}$ is uniform. In [WW23], the corresponding matrix $\mathbf{W} \coloneqq \tilde{\mathbf{W}}\mathbf{G}$ is structured, where $\tilde{\mathbf{W}} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n\ell\times n}$. Replacing $\mathbf{W}$ with a uniform matrix yields a construction with essentially the same efficiency as the construction of [WW23] while enabling a reduction to the weaker ℓ-succinct SIS assumption rather than the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption. Previously, [Wee23] showed that ℓ-succinct SIS implies the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption. The drawback of Construction B.1 is that the commitments are longer than those in [WW23] by a $\log q$ factor. In [WW23], because $\mathbf{W} \coloneqq \tilde{\mathbf{W}}\mathbf{G}$, we can take the commitment $\tilde{\mathbf{C}} \in \mathbb{Z}_q^{m\times m}$ and "pre-multiply" by $\mathbf{G}$. Namely, the commitment in [WW23] is $\mathbf{C} \coloneqq \mathbf{G}\tilde{\mathbf{C}} \in \mathbb{Z}_q^{n\times m}$. In contrast, when $\mathbf{W}$ is uniform, we cannot compress $\tilde{\mathbf{C}}$ anymore. Thus, in Construction B.1, the commitments are m-by-m matrices over $\mathbb{Z}_q$. Since $m = n \log q$, the commitments in Construction B.1 are larger by a factor $\log q = \mathsf{poly}(d^{1/\varepsilon}, \log\lambda, \log\ell)$.