Lattice-Based Functional Commitments: Fast Verification and Cryptanalysis

Hoeteck Wee[1,2] and David J. Wu[3(✉)]

[1] NTT Research, Sunnyvale, CA, USA
[2] ENS, Paris, France
[3] University of Texas at Austin, Austin, TX, USA
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 that supports fast verification with security from falsifiable assumptions. 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

J. Guo and R. Steinfeld (Eds.): ASIACRYPT 2023, LNCS 14442, pp. 201–235, 2023.
https://doi.org/10.1007/978-981-99-8733-7_7

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,BCFL22,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,BCFL22,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 random string model; see Sect. 1.3 for more details. Their construction relies on non-black-box use of cryptographic hash functions (as well as 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 [BCFL22]. 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.

Table 1. Summary of succinct *lattice-based* functional commitments. For each scheme, we report the class of functions they support, the size of the common reference string crs, the size of the commitment σ, and the size of an opening π as a function of the function f 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.

Scheme	**Functions**	$\vert\mathsf{crs}\vert$	$\vert\sigma\vert$	$\vert\pi\vert$	**FV**	**BB**	**Assumption**
[KLVW23][*]	Boolean circuits	1	1	1	✓	✗	LWE
[BCFL22]	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	✗	✓	$\mathsf{BASIS}_{\mathsf{struct}}$
[ACL+22]	degree-d polynomials[†]	ℓ^{2d}	1	1	✓	✓	k-R-ISIS
[BCFL22]	degree-d polynomials[§]	ℓ^{5d}	1	1	✓	✓	twin-k-M-ISIS
Cons. 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
Cons. 3.10	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 Sect. 1.3.

[§] Only supports commitments and openings to *small* values.

[†] The width of the circuit w is always at least the input length ℓ. In the case of an arbitrary *dense* polynomial of degree d (e.g., a polynomial with ℓ^d distinct monomials), then the width of the circuit computing it is ℓ^d.

[‡] The [dCP23] construction supports fast verification for certain special cases (e.g., vector commitments and polynomial commitments).

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 [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 and the second is the first dual functional commitment for (bounded-depth) Boolean circuits with fast verification and only making black-box use of cryptography. 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 the degree of the polynomial, λ is the security parameter, and ℓ is the input length.

For the particular 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 [BCFL22]. More broadly, when considering openings to polynomials of constant degree d, we achieve a factor of 2 reduction in the *exponent* for the CRS size compared to [ACL+22]. Namely, the [ACL+22] construction has a CRS of size $\ell^{2d} \cdot \mathsf{poly}(\lambda, d, \log \ell)$, so our construction reduces the exponent from $2d$ to $d + 1$. The [BCFL22] scheme has a smaller CRS for the case of sparse polynomials (e.g., when the width w of the circuit computing the polynomial f is roughly the input length $w \approx \ell$). Conversely, for dense polynomials with $\approx \ell^d$ monomials, and which corresponds to a circuit of width ℓ^d, the size of the CRS is significantly worse for their scheme. While the CRS size of our construction is worse than that of [WW23], the latter does not support fast verification (except in the 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 less structured generalizations of SIS compared to the k-R-ISIS and twin-k-M-ISIS assumptions used in [ACL+22, BCFL22]. We refer to Sect. 1.2 and Sect. 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 3.10). This is the first dual functional commitment scheme based on falsifiable assumptions that supports *succinct* openings and 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 scaled linearly with the input size 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 Sect. 1.2 and Construction 3.10). Security of our construction also relies on the falsifiable 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. Alone 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 Sect. 1.2 and the details in Sect. 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 $\mathbf{A}\mathbf{x} = \mathbf{t}$. 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 $\mathbf{g}^\intercal = [1, 2, \dots, 2^{\lfloor \log q \rfloor}]$) [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 $\mathbf{A} \xleftarrow{\text{R}} \mathbb{Z}_q^{n \times m}$, 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 $\mathbf{W} \xleftarrow{\text{R}} \mathbb{Z}_q^{\ell n \times m}$ 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 Sect. 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:

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

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

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

[1] In the full version of this paper, we provide the formal description and analysis of [WW23] using the ℓ-succinct SIS assumption.

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

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

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_ix_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 Sect. 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 $\mathbf{W}_{f_i}\mathbf{C}^2 = y_i\mathbf{G} - \mathbf{A}\tilde{\mathbf{V}}_1$. To support openings to multiple outputs, we publish random vectors $\mathbf{u}_1, \ldots, \mathbf{u}_T \xleftarrow{\text{R}} \mathbb{Z}_q^n$ in the CRS, and define the "multi-output" verification relation to be

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

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 3.6). 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 Sect. 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 over module lattices (as opposed to integer lattices). We describe the generalization in the full version of this paper. For a security parameter λ and using module lattices (along with a z-ary gadget matrix), 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, BCFL22]) 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 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 multiply both sides of Eq. (1.5) by $(\mathbf{x}^\mathsf{T} \otimes \mathbf{I}_n)$ and use the fact that $(\mathbf{x}^\mathsf{T} \otimes \mathbf{I}_n)(\mathbf{I}_\ell \otimes \mathbf{A}) = (1 \otimes \mathbf{A})(\mathbf{x}^\mathsf{T} \otimes \mathbf{I}_m) = \mathbf{A}(\mathbf{x}^\mathsf{T} \otimes \mathbf{I}_m)$, we have that

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

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

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

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 $[\mathbf{A} \mid \mathbf{W}_0 + (\mathbf{x}^\mathsf{T} \otimes \mathbf{I}_n)\mathbf{W}]$ to $\mathbf{B} - \mathbf{x}^\mathsf{T} \otimes \mathbf{G}$. 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 $\mathbf{B} \in \mathbb{Z}_q^{n \times \ell m}$, 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

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

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

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

[2] In the syntax of [Wee23], the ABE ciphertext is essentially $\mathbf{s}^\mathsf{T}[\mathbf{A} \mid \mathbf{W}_0 + (\mathbf{x} \otimes \mathbf{I}_n)\mathbf{W}] + \mathsf{error}$ 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 $\mathbf{B}$.

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

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

This yields a dual functional commitment for all (bounded-depth) Boolean circuits on ℓ-length inputs where the size of the commitment and the opening are both $\mathsf{poly}(\lambda, d^{1/\varepsilon}, \log \ell)$, where d is the bound on the depth of the function. The CRS in our construction has size $\ell^2 \cdot \mathsf{poly}(\lambda, d^{1/\varepsilon}, \log \ell)$. We note that 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 reduces to 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 Sect. 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 for NP. More specifically, in an extractable functional commitment, the binding property is replaced by a stronger extractability requirement which 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 succinct non-interactive argument (SNARG) for NP (using the fact that satisfiability of quadratic systems is NP-complete).

In Sect. 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 preimages for any coset of $\mathbf{A}$. (The pairing analogue would be recovering a trapdoor

[3] Technically, there is a polylogarithmic dependence on ℓ since $\log q$ scales with $\mathsf{poly}(\log \ell)$.

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

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

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

$$\mathbf{A}\mathbf{v} = \mathbf{D}\mathbf{c}.$$

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 $\mathbf{A}\mathbf{v}$ 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's 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 $\mathbf{A}\mathbf{v} = \mathbf{D}\mathbf{c}$. 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 $\mathbf{B} = [\mathbf{A} \mid \mathbf{D}\mathbf{G}]$ as follows:

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

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 $\mathbf{B}\mathbf{y} = \mathbf{0} \bmod q$, solves for the unique $\mathbf{z} \in \mathbb{Q}^{m+n}$ where $\mathbf{T}\mathbf{z} = \mathbf{y} \in \mathbb{Q}^{m+n}$, and then outputs the integer vector $\mathbf{y}^* = \mathbf{y} - \mathbf{T} \cdot \lfloor \mathbf{z} \rceil$. By construction $\mathbf{B}\mathbf{y}^* = \mathbf{0} \bmod q$ 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

[4] After communicating the attack to the authors of [ACL+22], Albrecht implemented and confirmed the attack [Alb23].

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

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, BCFL22], 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 $\mathbf{x} \mapsto \mathbf{M}\mathbf{x}$. 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 $\mathbf{M}\mathbf{x} = \mathbf{y}$. But this is precisely solving 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 Sect. 4.1 for more details.

We describe our methodology and attack algorithms in Sect. 4. We stress that our oblivious sampling attacks only apply to 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 (SNARKs), but constructions of SNARKs either rely on strong non-falsifiable assumptions [GW11] 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, BCFL22] 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, BCFL22, 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; notably, this scheme has a transparent setup and $\mathsf{poly}(\lambda, \log |x|, \log |C|)$ common reference string, commitment, and opening. 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 make 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). That is, 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

[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.

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. It is an interesting question to study whether our approach can be used to directly 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 $\mathbf{I}_\ell$ to denote the ℓ-by-ℓ identity matrix.

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

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

Lattice Preliminaries. Throughout this work, we let χ denote a Gaussian width parameter. We review some preliminaries on lattice-based cryptography in the full version of this paper.

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 $f\colon \mathcal{X}^\ell \to \mathcal{Y}^T$ 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 $\mathbf{x} \in \mathcal{X}^\ell$,*

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

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

$$\Pr\left[\mathsf{Verify}(\mathsf{crs}, \sigma, f, y_0, \pi_0) = 1 = \mathsf{Verify}(\mathsf{crs}, \sigma, f, y_1, \pi_1)\right] = \mathsf{negl}(\lambda),$$

where $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\ell, 1^d)$ *and* $(\sigma, f, (y_0, \pi_0), (y_1, \pi_1)) \leftarrow \mathcal{A}(1^\lambda, 1^\ell, 1^d, \mathsf{crs})$.

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.

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, BCFL22]. 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 $f \colon \mathcal{X}^\ell \to \mathcal{Y}^T$ 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 $\mathsf{Preprocess}$ 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 $\mathsf{OnlineVerify}$ 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 $\mathsf{vk}_f = \mathsf{poly}(\lambda, \log s)$, 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).

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. 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[7] with B_{in}-bounded coefficients and degree at most $d_{\max}$. 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 $f(\mathbf{x}) := \mathbf{f}^\mathsf{T} \mathbf{x}^{\otimes d}$. 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 $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$ and $\mathbf{W} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{Ln\times m}$. Next, define the target matrix

[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.

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

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 $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}})$, 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, BCFL22], 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 $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}})$ and the function $f = \mathbf{f} \in \mathbb{Z}_q^{\ell^d}$ for some $d \leq d_{\max}$, the preprocess algorithm outputs $\mathsf{vk}_f = \mathbf{F}_d = (\mathbf{f}^{\mathsf{T}} \otimes \mathbf{I}_m)\mathbf{W}_d \in \mathbb{Z}_q^{n \times m}$.
- $\mathsf{OnlineVerify}(\mathsf{vk}, \sigma, y, \pi)$: 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

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

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 $f \colon \mathbb{Z}_q^\ell \to \mathbb{Z}_q$ of degree at most d can be described by a *homogeneous* polynomial $f' \colon \mathbb{Z}_q^{\ell+1} \to \mathbb{Z}_q$ of degree d where $f'(\mathbf{x}') = f(\mathbf{x})$. Now, to open to an inhomogeneous polynomial f, the committer instead open to f'.

Correctness and Security Analysis. We provide the correctness and security analysis of Construction 3.2 in the full version of this paper.

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 approach from [WW23] for aggregating openings.

Construction 3.5 (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 $d \leq d_{\max}$:[8]

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

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

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 $\mathbf{D} \xleftarrow{\text{R}} \mathbb{Z}_q^{n \times T}$, 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 $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{D})$, 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 $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{D})$, 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
$$\|\mathbf{v}\| \le B \quad \text{and} \quad \sum_{i \in [T]} (\mathbf{f}_i^{\mathsf{T}} \otimes \mathbf{I}_m) \mathbf{W}_d \mathbf{C}^d \mathbf{G}^{-1}(\mathbf{d}_i) = \mathbf{D}\mathbf{y} - \mathbf{A}\mathbf{v}, \tag{3.4}$$
where $\mathbf{d}_i \in \mathbb{Z}_q^n$ is the i^{th} column of $\mathbf{D}$.

Remark 3.6 (Supporting Preprocessing). Like Construction 3.2, Construction 3.5 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 $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{D})$, 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
$$\mathbf{F} = \sum_{i \in [T]} \left((\mathbf{G}^{-1}(\mathbf{d}_i))^{\mathsf{T}} \otimes (\mathbf{f}_i^{\mathsf{T}} \otimes \mathbf{I}_m) \mathbf{W}_d \right) \in \mathbb{Z}_q^{n \times m^2} \tag{3.5}$$
$$\mathbf{y}^* = \mathbf{D}\mathbf{y} \in \mathbb{Z}_q^n, \tag{3.6}$$
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
$$\|\mathbf{v}\| \le B \quad \text{and} \quad \mathbf{F} \cdot \operatorname{vec}(\mathbf{C}^d) = \mathbf{y}^* - \mathbf{A}\mathbf{v}.$$

To show that this is correct, we apply vectorization to the main verification relation in Eq. (3.4):

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

Then, the main verification relation in Eq. (3.4) becomes

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

and correctness reduces to that of Construction 3.5. By construction, $|\mathsf{vk}_{f,\mathbf{y}}|=(nm^2+n)\log q$ and the running time of $\mathsf{OnlineVerify}$ is $\mathsf{poly}(n,m,d_{\max},\log q)$. As we show below, we can instantiate our scheme so that $n,m,\log q=\mathsf{poly}(\lambda,\log\ell,\log T)$, 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. (3.5). In this case, the running time of verification becomes $\mathsf{poly}(n,m,\log q,T)$.

Correctness and Security Analysis. We provide the correctness and security analysis as well as the parameter instantiation in the full version of this paper. We summarize the results in the following corollary:

Corollary 3.7 (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 $d\leq d_{\max}=O(1)$. 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 3.8 (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 the full version of this paper.

3.2 A Dual Construction 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 3.9 (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 3.10 (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 $\mathbf{W} \xleftarrow{\text{R}} \mathbb{Z}_q^{\ell n\times m}$. 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 $\mathbf{W}_0 \xleftarrow{\text{R}} \mathbb{Z}_q^{n\times m}$, 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 $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{W}_0, \mathbf{B})$ 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 $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{W}_0, \mathbf{B})$, 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
$$\pi = \mathbf{V} = \begin{bmatrix} -(\mathbf{x}^{\mathsf{T}} \otimes \mathbf{I}_m)\mathbf{T}_{\mathsf{open}} \\ -\mathbf{T}_{\mathsf{com}} \end{bmatrix} \cdot \mathbf{H}_{\mathbf{B},f,\mathbf{x}} \in \mathbb{Z}_q^{2m\times m}. \tag{3.7}$$
- $\mathsf{Verify}(\mathsf{crs}, \sigma, \mathbf{x}, y, \pi)$: On input $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{W}_0, \mathbf{B})$, 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
$$\|\mathbf{V}\| \le B \quad \text{and} \quad \mathbf{B}_f - y\mathbf{G} = [\mathbf{A} \mid \mathbf{W}_0 + (\mathbf{x}^{\mathsf{T}} \otimes \mathbf{I}_n)\mathbf{W}]\mathbf{V}. \tag{3.8}$$

Remark 3.11 (Supporting Preprocessing). Similar to Constructions 3.2 and 3.5, Construction 3.10 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 $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T}_{\mathsf{com}}, \mathbf{T}_{\mathsf{open}}, \mathbf{W}_0, \mathbf{B})$ and the input $\mathbf{x} \in \{0,1\}^\ell$, the preprocess algorithm outputs $\mathsf{vk}_{\mathbf{x}} = \mathbf{F}_{\mathbf{x}} = [\mathbf{A} \mid \mathbf{W}_0 + (\mathbf{x}^{\mathsf{T}} \otimes \mathbf{I}_n)\mathbf{W}] \in \mathbb{Z}_q^{n\times 2m}$.
- $\mathsf{OnlineVerify}(\mathsf{vk}, \sigma, y, \pi)$: 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
$$\|\mathbf{V}\| \le B \quad \text{and} \quad \mathbf{B}_f - y\mathbf{G} = \mathbf{F}_{\mathbf{x}}\mathbf{V}.$$

Correctness and Security Analysis. We provide the correctness, security analysis, and parameter instantiation for Construction 3.10 in the full version of this paper. We summarize the instantiation in the following corollary:

Corollary 3.12 (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. In the full version of this paper, we show that Construction 3.2 is not an extractable functional commitment for quadratic functions. In this section, we 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, this can be used to sample a valid opening to an unsatisfiable set of quadratic constraints, while in the other case (Sect. 4.1), we can embed a SIS instance that the extractor must solve in order to output a valid input. We start with the 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*

$$\Pr\left[\begin{array}{c} \mathsf{Verify}(\mathsf{crs}, \sigma, f, y, \pi) = 1 \ \textit{and} \\ f(\mathbf{x}) \neq y \end{array} : \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda) \\ \big((\sigma, f, y, \pi), \mathbf{x}\big) \leftarrow (\mathcal{A} \| \mathcal{E})(1^\lambda, \mathsf{crs}) \end{array}\right] = \mathsf{negl}(\lambda).$$

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 (σ, f, y, π) and the output of $\mathcal{E}$ is $\mathbf{x}$.*

4.1 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 $\mathsf{MatrixACLMT}$ 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

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

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 $m \ge O(t \log q)$. The MatrixACLMT knowledge assumption says that for every efficient adversary $\mathcal{A}$, there exists an efficient extractor $\mathcal{E}$ such that $\Pr[b = 1] = \mathsf{negl}(\lambda)$, where $b \in \{0, 1\}$ is the output of the following experiment:

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

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 $\mathbf{A}\left(\sum_{i\in[\ell]} x_i\mathbf{z}_i\right) = \mathbf{D}\left(\sum_{i\in[\ell]} x_t\mathbf{t}_i\right) = \mathbf{D}\mathbf{c}$. 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, all but a negligible fraction of the elements in the image of $\mathbf{A}$ are contained in the image of $\mathbf{D}$.

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* $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{t \times m}$, $\mathbf{D} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{t \times n}$, $\mathbf{t}_i \xleftarrow{\mathrm{R}} \mathbb{Z}_q^n$ *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.[9] *Thus, we can use* $\bar{\mathbf{T}}$ *to derive an Ajtai-trapdoor* $\mathbf{T}$ *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).*
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* $\mathbf{G}\mathbf{c}$ *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* $\mathbf{B}\mathbf{y} = \mathbf{0} \bmod q$*, solves for the unique* $\mathbf{z} \in \mathbb{Q}^{m+n}$ *where* $\mathbf{T}\mathbf{z} = \mathbf{y} \in \mathbb{Q}^{m+n}$ *and then outputs* $\mathbf{x} = \mathbf{y} - \mathbf{T} \cdot \lfloor \mathbf{z} \rceil$*. By construction* $\mathbf{B}\mathbf{x} = \mathbf{0} \bmod q$ *and moreover* $\|\mathbf{x}\| \le \|\mathbf{T}(\mathbf{z} - \lfloor \mathbf{z} \rceil)\|$*, 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 $\mathbf{T}$. 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 $\mathbf{x} \mapsto \mathbf{M}\mathbf{x}$. 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 $(\mathbf{A}, \mathbf{R}_{\mathbf{A}}) \leftarrow \mathsf{TrapGen}(1^\lambda, n, m)$, $\mathbf{W}_1, \ldots, \mathbf{W}_\ell \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n\times n}$, $\mathbf{u} \leftarrow \mathbb{Z}_q^n$, and let $\mathbf{t}_i \leftarrow \mathbf{W}_i^{-1}\mathbf{u} \in \mathbb{Z}_q^n$ for each $i \in [\ell]$. For

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

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 $\widehat{\mathbf{W}} \in \mathbb{Z}_q^{\ell n \times n}$ to 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 $\mathbf{D} \xleftarrow{\text{R}} \mathbb{Z}_q^{t \times n}$. Sample $\mathbf{z}'_i \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathbf{R_B}, \mathbf{D}\mathbf{t}_i, \chi)$ 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 $\mathsf{crs} = \big(\mathbf{A}, \mathbf{B}, \mathbf{D}, \mathbf{u}, \{\mathbf{W}_i\}_{i\in[\ell]}, \{\mathbf{z}_{i,j}\}_{i\neq j}, \{\mathbf{z}'_i\}_{i\in[\ell]}\big)$ 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 $\mathsf{crs} = \big(\mathbf{A}, \mathbf{B}, \mathbf{D}, \mathbf{u}, \{\mathbf{W}_i\}_{i\in[\ell]}, \{\mathbf{z}_{i,j}\}_{i\neq j}, \{\mathbf{z}'_i\}_{i\in[\ell]}\big)$, 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

$$\mathbf{v} = \begin{bmatrix} (\mathbf{M} \otimes \mathbf{I}_m)\widehat{\mathbf{v}} \\ (\mathbf{x}^\mathsf{T} \otimes \mathbf{I}_{m'})\mathbf{z}'_i \end{bmatrix} \in \mathbb{Z}_q^{km+m'}.$$

- $\mathsf{Verify}(\mathsf{crs}, \sigma, f_\mathbf{M}, y, \pi)$: On input $\mathsf{crs} = \big(\mathbf{A}, \mathbf{B}, \mathbf{D}, \mathbf{u}, \{\mathbf{W}_i\}_{i\in[\ell]}, \{\mathbf{z}_{i,j}\}_{i\neq j}, \{\mathbf{z}'_i\}_{i\in[\ell]}\big)$, a commitment $\sigma = \mathbf{c} \in \mathbb{Z}_q^n$, 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

$$\|\mathbf{v}\| \leq B \quad \text{and} \quad \begin{bmatrix} \mathbf{I}_k \otimes \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix} \cdot \mathbf{v} = \begin{bmatrix} (\mathbf{M} \otimes \mathbf{I}_n)\widehat{\mathbf{W}} \\ \mathbf{D} \end{bmatrix} \cdot \mathbf{c} - \begin{bmatrix} \mathbf{y} \otimes \mathbf{u} \\ \mathbf{0} \end{bmatrix}. \tag{4.1}$$

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 $\mathsf{crs} = \big(\mathbf{A}, \mathbf{B}, \mathbf{D}, \mathbf{u}, \{\mathbf{W}_i\}_{i\in[\ell]}, \{\mathbf{z}_{i,j}\}_{i\neq j}, \{\mathbf{z}'_i\}_{i\in[\ell]}\big)$ 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:

$$\begin{aligned}(\mathbf{M} \otimes \mathbf{I}_n)\widehat{\mathbf{W}}\mathbf{c} &= (\mathbf{M} \otimes \mathbf{I}_n)(\mathbf{x} \otimes \mathbf{u}) + (\mathbf{M} \otimes \mathbf{I}_n)(\mathbf{I}_\ell \otimes \mathbf{A})\widehat{\mathbf{v}} \\ &= (\mathbf{M}\mathbf{x} \otimes \mathbf{u}) + (\mathbf{I}_k \otimes \mathbf{A})(\mathbf{M} \otimes \mathbf{I}_m)\widehat{\mathbf{v}}\end{aligned}$$

$$\mathbf{D}\mathbf{c} = \sum_{i \in [\ell]} x_i \mathbf{D}\mathbf{t}_i = \mathbf{B} \cdot \left(\sum_{i \in [\ell]} x_i \mathbf{z}'_i \right) = \mathbf{B} \cdot (\mathbf{x}^{\mathsf{T}} \otimes \mathbf{I}_{m'})\hat{\mathbf{z}}.$$

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. (4.1). 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 $(\mathbf{B}, \mathbf{R}_{\mathbf{B}}) \leftarrow \mathsf{TrapGen}(1^\lambda, t, m')$, $\mathbf{D} \xleftarrow{\text{R}} \mathbb{Z}_q^{t \times n}$, 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 opening $\mathbf{v} = \left[\begin{smallmatrix}\mathbf{v}_1 \\ \mathbf{v}_2\end{smallmatrix}\right]$, 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 $\mathbf{v}' = \begin{bmatrix}\mathbf{v}'_1 \\ \mathbf{v}'_2\end{bmatrix}$ 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. (4.1):

$$(\mathbf{I}_k \otimes \mathbf{A})\mathbf{v}'_1 = (\mathbf{M} \otimes \mathbf{I}_n)\widehat{\mathbf{W}}\mathbf{c} - \mathbf{M}\mathbf{x} \otimes \mathbf{u} \tag{4.2}$$

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 $\mathbf{M} = [\mathbf{M}_{\text{L}} \mid \mathbf{0}^{k \times \ell_1}]$ and $\mathbf{M}_{\text{L}} \in \mathbb{Z}_q^{k \times \ell_2}$ 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 $\mathbf{Mx} = \mathbf{y}$.
3. Since the oblivious sampler is agnostic to the choice of $\mathbf{M}_{\text{L}}$ (as long as it is short), we can embed an (inhomogeneous) SIS instance into $\mathbf{M}_{\text{L}}$. In this case, an extractor for algorithm $\mathcal{A}$ is able to solve inhomogeneous SIS with respect to $\mathbf{M}$, and by extension, $\mathbf{M}_{\text{L}}$.

We defer the details to the the full version of this paper. Taken together, 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. David J. Wu is supported in part by NSF CNS-2151131, CNS-2140975, CNS-2318701, a Microsoft Research Faculty Fellowship, and a Google Research Scholar award.

References

ACL+22. Albrecht, M.R., Cini, V., Lai, R.W.F., Malavolta, G., Thyagarajan, S.A.: Lattice-based SNARKs: publicly verifiable, preprocessing, and recursively composable. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022. LNCS, vol. 13508, pp. 102–132. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-15979-4_4

Ajt96. Ajtai, M.. Generating hard instances of lattice problems (extended abstract). In: STOC (1996)

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

AR20. Agrawal, S., Raghuraman, S.: KVaC: key-value commitments for blockchains and beyond. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020. LNCS, vol. 12493, pp. 839–869. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64840-4_28

BC12. Bitansky, N., Chiesa, A.: Succinct arguments from multi-prover interactive proofs and their efficiency benefits. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 255–272. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_16

BCFL22. Balbás, D., Catalano, D., Fiore, D., Lai, R.W.F.: Functional commitments for circuits from falsifiable assumptions. IACR Cryptol. ePrint Arch. (2022)

BCI+13. Bitansky, N., Chiesa, A., Ishai, Y., Paneth, O., Ostrovsky, R.: Succinct non-interactive arguments via linear interactive proofs. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 315–333. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36594-2_18

BGG+14. Boneh, D., et al.: Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 533–556. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_30

BHK17. Brakerski, Z., Holmgren, J., Kalai, Y.T.: Non-interactive delegation and batch NP verification from standard computational assumptions. In: STOC (2017)

BNO21. Boneh, D., Nguyen, W., Ozdemir, A.: How to commit to private functions. In: IACR Cryptol. ePrint Arch, Efficient Functional Commitments (2021)

BTVW17. Brakerski, Z., Tsabary, R., Vaikuntanathan, V., Wee, H.: Private constrained PRFs (and more) from LWE. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017. LNCS, vol. 10677, pp. 264–302. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70500-2_10

BV15. Brakerski, Z., Vaikuntanathan, V.: Constrained key-homomorphic PRFs from standard lattice assumptions. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9015, pp. 1–30. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46497-7_1

CF13. Catalano, D., Fiore, D.: Vector commitments and their applications. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 55–72. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36362-7_5

CFG+20. Campanelli, M., Fiore, D., Greco, N., Kolonelos, D., Nizzardo, L.: Incrementally aggregatable vector commitments and applications to verifiable decentralized storage. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020. LNCS, vol. 12492, pp. 3–35. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64834-3_1

CJJ21. Choudhuri, A.R., Jain, A., Jin, Z.: SNARGs for $\mathcal{P}$ from LWE. In: FOCS (2021)

CLM23. Cini, V., Lai, R.W.F., Malavolta, G.: Lattice-based succinct arguments from vanishing polynomials: (extended abstract). In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023. LNCS, vol. 14082, pp. 72–105. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-38545-2_3

dCP23. de Castro, L., Peikert, C.: Functional commitments for all functions, with transparent setup and from SIS. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023. LNCS, vol. 14006, pp. 287–320. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30620-4_10

FLV23. Fisch, B., Liu, Z., Vesely, P.: Orbweaver: succinct linear functional commitments from lattices. In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023. LNCS, vol. 14082, pp. 106–131. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-38545-2_4

GGPR13. Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic span programs and succinct NIZKs without PCPs. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 626–645. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_37

Gro16. Groth, J.: On the size of pairing-based non-interactive arguments. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 305–326. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_11

GRWZ20. Gorbunov, S., Reyzin, L., Wee, H., Zhang, Z.: Aggregating proofs for multiple vector commitments. In: ACM CCS, Pointproofs (2020)

GSW13. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. CRYPTO 2013. LNCS, vol. 8042, pp. 75–92. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_5

GVW13. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute-based encryption for circuits. In: STOC (2013)

GVW15a. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Predicate encryption for circuits from LWE. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 503–523. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_25

GVW15b. Gorbunov, S., Vaikuntanathan, V., Wichs, D.: Leveled fully homomorphic signatures from standard lattices. In: STOC (2015)

GW11. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: STOC (2011)

IKO07. Ishai, Y., Kushilevitz, E., Ostrovsky, R.: Efficient arguments without short PCPs. In: CCC (2007)

Kil92. Kilian, J.: A note on efficient zero-knowledge proofs and arguments (extended abstract). In: STOC (1992)

KLVW23. Kalai, Y., Lombardi, A., Vaikuntanathan, V., Wichs, D.: Boosting batch arguments and RAM delegation. In: STOC (2023)

KP16. Kalai, Y., Paneth, O.: Delegating RAM computations. In: Hirt, M., Smith, A. (eds.) TCC 2016. LNCS, vol. 9986, pp. 91–118. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53644-5_4

KPY19. Kalai, Y.T., Paneth, O., Yang, L.: How to delegate computations publicly. In: STOC (2019)

KVZ21. Kalai, Y.T., Vaikuntanathan, V., Zhang, R.Y.: Somewhere statistical soundness, post-quantum security, and SNARGs. In: Nissim, K., Waters, B. (eds.) TCC 2021. LNCS, vol. 13042, pp. 330–368. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-90459-3_12

KZG10. Kate, A., Zaverucha, G.M., Goldberg, I.: Constant-size commitments to polynomials and their applications. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 177–194. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17373-8_11

LM19. Lai, R.W.F., Malavolta, G.: Subvector commitments with application to succinct arguments. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11692, pp. 530–560. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26948-7_19

LP20. Lipmaa, H., Pavlyk, K.: Succinct functional commitment for a large class of arithmetic circuits. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020. LNCS, vol. 12493, pp. 686–716. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64840-4_23

LRY16. Libert, B., Ramanna, S.C., Yung, M.: From polynomial commitments to pairing-based accumulators from simple assumptions. In: ICALP, Functional Commitment Schemes (2016)

LY10. Libert, B., Yung, M.: Concise mercurial vector commitments and independent zero-knowledge sets with short proofs. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 499–517. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11799-2_30

Mer87. Merkle, R.C.: A digital signature based on a conventional encryption function. In: Pomerance, C. (ed.) CRYPTO 1987. LNCS, vol. 293, pp. 369–378. Springer, Heidelberg (1988). https://doi.org/10.1007/3-540-48184-2_32

Mic00. Micali, S.: Computationally sound proofs. SIAM J. Comput. **30**(4), 1253–1298 (2000)

MP12. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_41

Nit21. Nitulescu, A.: SoK: Vector Commitments (2021). https://www.di.ens.fr/~nitulesc/files/vc-sok.pdf

PHGR13. Parno, B., Howell, J., Gentry, C., Raykova, M: Nearly practical verifiable computation. In: IEEE Symposium on Security and Privacy, Pinocchio (2013)

PPS21. Peikert, C., Pepin, Z., Sharp, C.: Vector and functional commitments from lattices. In: Nissim, K., Waters, B. (eds.) TCC 2021. LNCS, vol. 13044, pp. 480–511. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-90456-2_16

PSTY13. Papamanthou, C., Shi, E., Tamassia, R., Yi, K.: Streaming authenticated data structures. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 353–370. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_22

TAB+20. Tomescu, A., et al.: Aggregatable subvector commitments for stateless cryptocurrencies. In: Galdi, C., Kolesnikov, V. (eds.) SCN 2020. LNCS, vol. 12238, pp. 45–64. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-57990-6_3

Tsa22. Tsabary, R.: Candidate witness encryption from lattice techniques. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022. LNCS, vol. 13507, pp. 535–559. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-15802-5_19

TXN20. Tomescu, A., Xia, Y., Newman, Z.: Authenticated dictionaries with cross-incremental proof (dis)aggregation. IACR Cryptol. ePrint Arch. (2020)

VWW22. Vaikuntanathan, V., Wee, H., Wichs, D.: Witness encryption and null-IO from evasive LWE. In: Agrawal, S., Lin, D. (eds.) ASIACRYPT 2022. LNCS, vol. 13791, pp. 195–221. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-22963-3_7

Wee22. Wee, H.: Optimal broadcast encryption and CP-ABE from evasive lattice assumptions. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022. LNCS, vol. 13276, pp. 217–241. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-07085-3_8

Wee23. Wee, H.: Circuit ABE with small ciphertexts and keys from lattices (2023). Manuscript

WW23. Wee, H., Wu, D.J.: Succinct vector, polynomial, and functional commitments from lattices. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023. LNCS, vol. 14006, pp. 385–416. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30620-4_13

WWW22. Waters, B., Wee, H., Wu, D.J.: Multi-authority ABE from lattices without random oracles. In: Kiltz, E., Vaikuntanathan, V. (eds.) TCC 2022, vol. 13747, pp. 651–679. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-22318-1_23