Succinct Vector, Polynomial, and Functional Commitments from Lattices

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

Abstract. Vector commitment schemes allow a user to commit to a vector of values $\mathbf{x} \in \{0,1\}^\ell$ and later, open up the commitment to a specific set of positions. Both the size of the commitment and the size of the opening should be succinct (i.e., polylogarithmic in the length ℓ of the vector). Vector commitments and their generalizations to polynomial commitments and functional commitments are key building blocks for many cryptographic protocols.

We introduce a new framework for constructing non-interactive lattice-based vector commitments and their generalizations. A simple instantiation of our framework yields a new vector commitment scheme from the standard short integer solution (SIS) assumption that supports *private* openings and large messages. We then show how to use our framework to obtain the first succinct *functional* commitment scheme that supports openings with respect to arbitrary bounded-depth Boolean circuits. In this scheme, a user commits to a vector $\mathbf{x} \in \{0,1\}^\ell$, and later on, open the commitment to any function $f(\mathbf{x})$. Both the commitment *and* the opening are non-interactive and succinct: namely, they have size $\mathsf{poly}(\lambda, d, \log \ell)$, where λ is the security parameter and d is the *depth* of the Boolean circuit computing f. Previous constructions of functional commitments could only support constant-degree polynomials, or require a trusted *online* authority, or rely on non-falsifiable assumptions. The security of our functional commitment scheme is based on a new falsifiable family of "basis-augmented" SIS assumptions (BASIS) we introduce in this work.

We also show how to use our vector commitment framework to obtain (1) a polynomial commitment scheme where the user can commit to a polynomial $f \in \mathbb{Z}_q[x]$ and subsequently open the commitment to an evaluation $f(x) \in \mathbb{Z}_q$; and (2) an aggregatable vector (resp., functional) commitment where a user can take a set of openings to multiple indices (resp., function evaluations) and aggregate them into a *single* short opening. Both of these extensions rely on the same BASIS assumption we use to obtain our succinct functional commitment scheme.

1 Introduction

Vector commitment schemes [Mer87, CFM08, LY10, CF13] allow a user to commit to a vector of values $\mathbf{x} \in \{0,1\}^\ell$ and subsequently, open up the commitment to a specific set of positions. Both the commitment and the openings

should be *succinct* (i.e., have size that scales *polylogarithmically* with the vector length ℓ) and *non-interactive*.[4] There has recently been tremendous interest and progress in the design and application of vector commitments, and even a "Vector Commitment Research Day" [Res22]. Starting from the classic vector commitment scheme of Merkle [Mer87] based on collision-resistant hash functions, we now have a broad range of algebraic constructions from pairing-based assumptions [LY10, KZG10, CF13, LRY16, LM19, TAB+20, GRWZ20] as well as assumptions over groups of unknown order (e.g., RSA groups or class groups) [CF13, LM19, CFG+20, AR20, TXN20]. We refer to [Nit21] for a survey of recent schemes. As a primitive, vector commitment schemes have found numerous applications to verifiable outsourced databases [BGV11, CF13], cryptographic accumulators [CF13], pseudonymous credentials [KZG10], and to blockchain protocols [RMCI17, CPSZ18, BBF19]. Moreover, the generalization of vector commitments to polynomial commitments [KZG10] has emerged as a key building block in many recent (random-oracle) constructions of succinct non-interactive arguments of knowledge (SNARKs) [MBKM19, CHM+20, GWC19, BDFG21, BFS20, COS20] having various appealing properties (e.g., universal or transparent setup, recursive composability, and more).

In this work, we focus on two themes in the study of vector commitments where progress has been more limited: (1) post-quantum constructions based on lattices [PSTY13, LLNW16, PPS21, ACL+22, FSZ22]; and (2) functional commitments, a generalization of vector commitments that supports openings to various functions on the committed values [LRY16, LP20, PPS21, BNO21, ACL+22]. There are good technical reasons for the limited progress on these two fronts. First, many of the techniques developed for vector commitments crucially exploit algebraic structure in pairing and RSA/class groups that do not naturally extend to the lattice setting. Second, pairing and RSA/class groups only support limited homomorphic capabilities.

1.1 Our Results

In this work, we introduce a general framework for constructing lattice-based vector commitments that simultaneously encapsulates recent lattice-based vector commitment schemes [PPS21, ACL+22] and enables us to achieve stronger functionality and security properties. As we describe below, our framework readily generalizes to also yield *polynomial commitments*, *functional commitments*, and *aggregatable commitments* from (falsifiable) lattice-based assumptions.

A new family of SIS assumptions. The security of our schemes relies on a new "basis-augmented" family of short integer solution (SIS) assumptions we introduce in this work. We refer to our basis-augmented SIS assumption as the BASIS assumption (Assumption 3.2). The basic version of our assumption (denoted $\mathsf{BASIS}_{\mathsf{rand}}$) suffices for constructing standard vector commitments and is implied

[4] We discuss interactive commitments (as well as constructions in the random oracle model) in Section 1.3.

by the *standard* SIS assumption. The structured version of the assumption (denoted $\mathsf{BASIS}_{\mathsf{struct}}$) we need for our extensions has a similar flavor as the k-SIS-like assumptions introduced in [ACL+22] for constructing lattice-based succinct arguments (c.f., Section 6). While the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption is not a standard lattice-based assumption, it is a *falsifiable* assumption [Nao03]. We view our assumption as a "q-type" lattice assumption and at a conceptual level, it shares a similar flavor as the q-type assumptions used in the pairing-based world for constructing vector commitments [CF13] and polynomial commitments [KZG10].

Vector commitments with private opening. An immediate consequence of our framework is a vector commitment scheme that supports *private* openings. In this setting, a user can commit to a vector $\mathbf{x} \in \{0,1\}^\ell$ with a short commitment σ and then open σ to an index-value pair (i, x_i) with a short opening π_i. We say the vector commitment scheme supports private openings if the commitment σ and any collection of openings $\{(i, x_i, \pi_i)\}_{i \in S}$ reveal no additional information about x_j for any $j \notin S$. Notably and in contrast to previous lattice-based vector commitment schemes [PPS21, ACL+22], our scheme also does *not* impose any restrictions on the magnitude of the entries of $\mathbf{x}$ (the vectors can be arbitrary elements of $\mathbb{Z}_q^\ell$ and the commitment as well as the opening are vectors over $\mathbb{Z}_q$). Previous lattice-based schemes [PPS21, ACL+22] require that the components of $\mathbf{x}$ be small and this property was essential for *both* correctness and security.

Our vector commitment scheme has the same efficiency properties as the earlier scheme of Peikert et al. [PPS21] which did not support private openings and was limited to a small message space. Our scheme provides the same functionality (e.g., support for "stateless updates") and like the scheme of [PPS21], security can be based on the *standard* SIS assumption. Thus, relative to [PPS21], our framework achieves private openings and supports a large message space with essentially no overhead.

We could alternatively obtain a lattice-based vector commitment by instantiating Merkle's classic construction [Mer87] with a lattice-based collision-resistant hash function (e.g., Ajtai's hash function from SIS [Ajt96, GGH96]). Our vector commitment scheme improves upon this generic approach in two main ways: (1) we support (bounded) stateless updates like [PPS21] (where a user can update a commitment to a vector $\mathbf{x}$ into a commitment to a vector $\mathbf{x}'$ given only knowledge of the difference $\mathbf{x}' - \mathbf{x}$ and not the entirety of $\mathbf{x}$ or $\mathbf{x}'$); and (2) we can support private openings directly. It is possible to extend Merkle hashing to support private openings via zero-knowledge proofs, but this would either need non-black-box use of cryptography or require interaction, random oracles, or correlation-intractable hash functions [CCH+19, PS19]. More broadly, as we illustrate below, our algebraic scheme serves as a stepping stone for realizing polynomial and functional commitment schemes (for which we crucially exploit *algebraic* structure).

Functional commitments. A functional commitment [GVW15, LRY16] is a generalization of a vector commitment with the property that given a commitment to an input $\mathbf{x} \in \{0,1\}^\ell$, one can then construct an opening π_f to $y = f(\mathbf{x})$,

for some function f. The basic binding property of the commitment scheme says that the adversary cannot come up with openings π_f and π'_f that open σ to different values $y \neq y'$ with respect to the same function f. The efficiency requirements are that the size of the commitment and the opening should be sublinear in both the size of the function f and the length of the input $\mathbf{x}$. Previously, Peikert et al. [PPS21] showed how to construct functional commitments for bounded-depth Boolean circuits in an *online* model where a central *trusted* authority issues opening keys for functions f, with security based on the standard SIS assumption. Albrecht et al. [ACL+22] subsequently showed how to construct functional commitments for constant-degree polynomials from new variants of the SIS assumption in the standard setting without an online authority. Earlier pairing-based functional commitments could only support linear functions [LRY16] or sparse polynomials [LP20]. Functional commitments can also be obtained generically by combining a vanilla vector commitment (e.g., a Merkle tree [Mer87]) with a succinct non-interactive argument of knowledge (for NP). However, existing constructions of SNARKs (for NP) either rely on making non-falsifiable assumptions [GW11] or working in idealized models.

Our vector commitment framework directly yields a succinct functional commitment scheme for all bounded-depth Boolean circuits in the standard offline model without an authority and from falsifiable assumptions. The size of the commitment and the openings are $\mathsf{poly}(\lambda, d, \log \ell)$, where λ is a security parameter, d is the *depth* of the Boolean circuit computing $f\colon \{0,1\}^\ell \to \{0,1\}$, and ℓ is the length of the input. Security relies on the new non-standard, but falsifiable, $\mathsf{BASIS}_{\mathsf{struct}}$ assumption we introduce in this work (with a *sub-exponential* noise bound). Notably, this is the first succinct functional commitment scheme for general circuits from a falsifiable assumption, and answers an open question posed by Peikert et al. [PPS21]. Our construction can be viewed as a *succinct* analog of the homomorphic commitments and signatures introduced by [GSW13, GVW15].[5]

Polynomial commitments. In a polynomial commitment [KZG10], a user can commit to a polynomial $f \in \mathbb{Z}_q[x]$ over $\mathbb{Z}_q$ and later open to an evaluation $f(x)$ at any point $x \in \mathbb{Z}_q$. A polynomial commitment can be viewed as a succinct commitment to the vector of evaluations of f on all inputs $x \in \mathbb{Z}_q$. While a polynomial commitment can be built from a succinct functional commitment for Boolean circuits, this incurs a $\mathsf{poly}(\log q)$ overhead to encode the polynomial evaluation over $\mathbb{Z}_q$ as a Boolean circuit and also relies on the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption with a *sub-exponential* noise bound. In this work, we show that a simple adaptation of our succinct functional commitments to the setting of linear functions directly gives a polynomial commitment over $\mathbb{Z}_q$. Notably, this construction can be based on our $\mathsf{BASIS}_{\mathsf{struct}}$ assumption with only a *polynomial* noise bound. This is the first polynomial commitment scheme from lattices based on falsifiable assumptions.

An important feature of our framework that enables the direct construction of polynomial commitments is that it natively supports values over $\mathbb{Z}_q$. Previous

[5] The homomorphic commitments from [GSW13, GVW15] are *non-succinct*; in particular, the size of the commitment scales *linearly* with the input length ℓ.

lattice-based vector commitments [PPS21, ACL+22] required that the committed value x and the opened value $f(x)$ be "small," and moreover, that the modulus q scale with the norm of the output (i.e., $f(x)$) when computed over the *integers*. This is not suitable when constructing polynomial commitments directly, as the size of $f(x)$ computed over the integers scales with the degree of f. Correspondingly, if the modulus q scales linearly with the degree of f, then the resulting scheme is no longer succinct. The ability to directly work over the entirety of $\mathbb{Z}_q$ is an appealing property of our new framework.

Aggregatable commitments. A simple modification to our basic vector commitment scheme yields a scheme that supports *aggregation*. We say a vector commitment scheme is aggregatable [BBF19, CFG+20] if given a commitment σ along with a set of openings $\pi_1, \ldots, \pi_t$ to indices $i_1, \ldots, i_t \in [\ell]$ and values $x_{i_1}, \ldots, x_{i_t}$, there is an efficient aggregation algorithm that outputs a short aggregate opening π that validates the full set of values $\{(i_j, x_{i_j})\}_{j \in [t]}$. The requirement is that the size of π scale sublinearly, or better yet, *polylogarithmically* with t. Aggregatable commitments immediately imply subvector commitments [LM19] (i.e., a vector commitment scheme that supports batch openings to a set of indices $S \subseteq [\ell]$). Our framework yields an aggregatable commitment scheme for short messages from the same falsifiable $\mathsf{BASIS}_{\mathsf{struct}}$ assumption used to construct succinct functional commitments. This is the first aggregatable commitment scheme from lattice assumptions *without* relying on general-purpose succinct arguments [ACL+22] or batch arguments [CJJ21, DGKV22], and answers another open question posed by Peikert et al. [PPS21].

A limitation of our aggregatable commitment is that it only satisfies *same-set binding*, which guarantees that for every subset of indices $S \subseteq [\ell]$, the adversary can only open to a single set of values. However, there is still the possibility that an adversary could open the commitment to different sets S and T that are *inconsistent* (i.e., $x_i = 0$ with respect to S while $x_i = 1$ with respect to T).[6] Constructing aggregatable commitments that satisfy the stronger notion of *different-set* binding directly from falsifiable lattice-based assumptions is an interesting open problem.

The same techniques we use to construct aggregatable vector commitments also applies to our succinct functional commitment scheme, and we obtain an aggregatable functional commitment scheme from the same underlying hardness assumption. In this setting, a user can take openings $\pi_1, \ldots, \pi_t$ for function-value pairs $(f_1, y_1), \ldots, (f_t, y_t)$ and aggregate the openings into a single short opening π that validates all t function-value pairs and where the size of the aggregated opening scales polylogarithmically with t.

Summary. Similar to previous lattice-based vector commitments [PPS21, ACL+22], we rely on a *structured* reference string in all of our constructions. We refer to the structured reference string as a common reference string (CRS). To summa-

[6] Note though that if the commitment is honestly-generated, then same-set binding implies different-set binding; see the full version of this paper [WW22].

rize, our new lattice-based vector commitment framework yields the following constructions:

- A vector commitment scheme with private openings based on the *standard* SIS assumption with polynomial noise bound (Corollary 3.6). For vectors of dimension ℓ, the size of the commitment is $O(\lambda(\log \lambda + \log \ell))$ and the size of an opening is $O(\lambda(\log^2 \lambda + \log^2 \ell))$.[7] The size of the CRS is $\ell^2 \cdot \mathsf{poly}(\lambda, \log \ell)$.
- A succinct functional commitment scheme supporting all bounded-depth Boolean circuits from the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption with a sub-exponential noise bound (Corollary 4.3). A variant of this construction supports private openings under a weaker notion of target binding. For both constructions, to support functions on ℓ-bit inputs and computable by Boolean circuits of depth d, the sizes of the commitment and openings are $\mathsf{poly}(\lambda, d, \log \ell)$. The size of the CRS is $\ell^2 \cdot \mathsf{poly}(\lambda, d, \log \ell)$.
- A polynomial commitment (for polynomials of a priori bounded degree) under the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption with a polynomial noise bound. To support polynomials of degree up to d over $\mathbb{Z}_q$ (where $q = \mathsf{poly}(\lambda)$), the sizes of the commitment and openings are $\mathsf{poly}(\lambda, \log d)$. The size of the CRS is $d^2 \cdot \mathsf{poly}(\lambda, \log d)$.
- An aggregatable vector commitment scheme (over a small message space) based on the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption with polynomial noise bound. The sizes of the commitment, openings, and CRS match those above for our vanilla vector commitment.
- An aggregatable functional commitment scheme for all bounded-depth Boolean circuits from the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption used to obtain succinct functional commitments. To support aggregating T openings for functions on ℓ-bit inputs and computable by Boolean circuits of depth d, the sizes of the commitment and opening are $\mathsf{poly}(\lambda, d, \log \ell, \log T)$. The size of the CRS is $(\ell^2 + T) \cdot \mathsf{poly}(\lambda, d, \log \ell, \log T)$. In the random oracle model, we can reduce the CRS size to $\ell^2 \cdot \mathsf{poly}(\lambda, d, \log \ell)$ and support an *arbitrary* polynomial number of aggregations.

1.2 Technical Overview

In this section, we provide a general overview of our new framework for constructing vector commitments from lattices as well as the family of basis-augmented SIS assumptions (BASIS) we use to prove hardness. In the following description, 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 conditioned on $\mathbf{A}\mathbf{x} = \mathbf{t}$. Sampling $\mathbf{x} \leftarrow \mathbf{A}^{-1}(\mathbf{t})$ can be done efficiently given a trapdoor for $\mathbf{A}$ [Ajt96, GPV08, AP09, ABB10a, ABB10b, CHKP10, MP12]. Here, we will use the Micciancio-Peikert gadget trapdoors [MP12]; namely, a matrix $\mathbf{R}$

[7] We note that these bounds match the base construction of Peikert et al. [PPS21]. While [PPS21, Figure 1] reports that their scheme has $O(\log \ell)$-size openings (ignoring the security parameter λ), the construction itself [PPS21, Construction 1] has $O(\log^2 \ell)$-size openings.

is a gadget trapdoor for $\mathbf{A}$ if $\mathbf{R}$ is short and $\mathbf{AR} = \mathbf{G}$, where $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}^\intercal$ is the gadget matrix and $\mathbf{g}^\intercal = [1, 2, \ldots, 2^{\lfloor \log q \rfloor}]$.

A general framework for constructing vector commitments. We begin by describing a general framework for constructing lattice-based vector commitments that encapsulates the recent schemes from [PPS21, ACL+22]:

- The common reference string (CRS) specifies a collection of ℓ matrices $\mathbf{A}_1, \ldots, \mathbf{A}_\ell \in \mathbb{Z}_q^{n \times m}$ and ℓ vectors $\mathbf{t}_1, \ldots, \mathbf{t}_\ell \in \mathbb{Z}_q^n$ along with some auxiliary input $\mathsf{aux}_\ell := \{\mathbf{A}_i^{-1}(\mathbf{t}_j)\}_{i \neq j}$.
- The commitment to a vector $\mathbf{x} = (x_1, \ldots, x_\ell) \in \{0,1\}^\ell$ is a vector $\mathbf{c} \leftarrow \sum_{i \in [\ell]} x_i \mathbf{t}_i \in \mathbb{Z}_q^n$.
- An opening to index $i \in [\ell]$ and value $x_i \in \{0,1\}$ is a short vector $\mathbf{v}_i \in \mathbb{Z}_q^m$ such that
$$\mathbf{c} = \mathbf{A}_i \mathbf{v}_i + x_i \mathbf{t}_i. \tag{1.1}$$
The honest opening is computed as $\mathbf{v}_i \leftarrow \sum_{j \neq i} x_j \mathbf{A}_i^{-1}(\mathbf{t}_j)$.

Correctness follows by inspection:

$$\mathbf{A}_i \mathbf{v}_i + x_i \mathbf{t}_i = \sum_{j \neq i} x_j \mathbf{A}_i \cdot \mathbf{A}_i^{-1}(\mathbf{t}_j) + x_i \mathbf{t}_i = \sum_{i \in [\ell]} x_i \mathbf{t}_i = \mathbf{c}.$$

For binding, we require that it is hard to find a short vector $\mathbf{z} \in \mathbb{Z}_q^m$ such that $\mathbf{A}_i \mathbf{z} = \mathbf{t}_i$ for any $i \in [\ell]$ given the components in the CRS. Next, we explain how the schemes PPS_1 from [PPS21][8] and MatrixACLMT from [ACL+22][9] fall into this framework.

- In PPS_1, the matrices $\mathbf{A}_i \xleftarrow{R} \mathbb{Z}_q^{n \times m}$ and vectors $\mathbf{t}_i \xleftarrow{R} \mathbb{Z}_q^n$ are independent and uniformly random for all $i \in [\ell]$. Binding in turn is based on the standard SIS assumption.
- In MatrixACLMT, they sample uniformly random vectors $\mathbf{u}_i \xleftarrow{R} \mathbb{Z}_q^n$, a matrix $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, and invertible matrices $\mathbf{W}_i \xleftarrow{R} \mathbb{Z}_q^{n \times n}$ for each $i \in [\ell]$. Then, they set $\mathbf{A}_i \leftarrow \mathbf{W}_i \mathbf{A}$, $\mathbf{t}_i \leftarrow \mathbf{W}_i \mathbf{u}_i$. In this case, $\mathbf{A}_i^{-1}(\mathbf{t}_j) = \mathbf{A}^{-1}(\mathbf{W}_i^{-1} \mathbf{W}_j \mathbf{u}_j)$. Binding is based on a new assumption which stipulates that it is hard to find a short vector $\mathbf{z} \in \mathbb{Z}_q^m$ where $\mathbf{Az} = \mathbf{u}_i$ for any $i \in [\ell]$ given the CRS. The authors of [ACL+22] then show how to leverage the extra structure arising from the correlated $\mathbf{A}_i$'s to obtain a functional commitment scheme for constant-degree polynomials as well as a preprocessing succinct non-interactive argument (SNARG) for NP.

Before describing our approach, we describe two limitations of these instantiations:

[8] By PPS_1, we refer to the the base scheme from [PPS21, Construction 1]; they also present a second tree-based scheme that uses PPS_1 as a building block.

[9] The authors of [ACL+22] describe their scheme in the ring setting. We write MatrixACLMT to denote one possible translation from the ring setting to the integer setting. Note that there are other ways to translate their scheme to the integer setting such as sampling $\mathbf{W}_i \xleftarrow{R} \mathbb{Z}_q^{m \times m}$ and then setting $\mathbf{A}_i \leftarrow \mathbf{AW}_i$.

– **Small message space:** In both the PPS_1 and the MatrixACLMT instantiations of this framework, both correctness *and* security require that the entries of the vector $\mathbf{x} = [x_1, \ldots, x_\ell]$ be small. This is because the verification relation is checking that the opening $\mathbf{v}_i = \sum_{j \neq i} x_j \mathbf{A}_i^{-1}(\mathbf{t}_j)$ is small. Thus, correctness requires that each x_j be small. Moreover, in the proof of binding, the reduction algorithm takes a commitment $\mathbf{c}$ along with two openings $(x_i, \mathbf{v}_i)$, $(x_i', \mathbf{v}_i')$ to derive a solution to SIS or a related problem. The existing reductions require that the difference $(x_i - x_i')$ be small (in order to derive a *short* solution).
– **Uniform target vectors.** In both the PPS_1 and MatrixACLMT constructions, the target vectors $\mathbf{t}_i$ are essentially random vectors. This is important for ensuring that $\mathbf{A}_i^{-1}(\mathbf{t}_j)$ does not leak a trapdoor for $\mathbf{A}_i$, which would immediately break binding. Using structured target vectors could enable additional functionality. For instance, in Remark 6.1, we show that instantiating MatrixACLMT with structured targets can be used to support functional openings. Unfortunately, this instantiation *also* leaks a trapdoor for $\mathbf{A}_i$, and is insecure.

The approach we take in this work avoids these limitations and allows us to construct vector commitments with a large message space as well as support new capabilities like polynomial and functional openings.

Our approach. We consider the same verification relation $\mathbf{c} = \mathbf{A}_i \mathbf{v}_i + x_i \mathbf{t}_i$ from Eq. (1.1), but take a completely different approach for computing the commitment $\mathbf{c}$ and the openings $\mathbf{v}_i$: we sample a *random* tuple $(\mathbf{v}_1, \ldots, \mathbf{v}_\ell, \mathbf{c})$ that simultaneously satisfies the verification relation for all $i \in [\ell]$. As in the previous verification relation, we require that the openings $\mathbf{v}_1, \ldots, \mathbf{v}_\ell$ are short. The commitment $\mathbf{c}$ can have large entries. In our particular setting, we write $\mathbf{c}$ as $\mathbf{c} = \mathbf{G}\hat{\mathbf{c}}$ where $\hat{\mathbf{c}} \in \mathbb{Z}_q^m$ is a short vector. Using the gadget matrix $\mathbf{G}$ will be important in the security analysis. Then, Eq. (1.1) corresponds to the relation $\mathbf{G}\hat{\mathbf{c}} = \mathbf{A}_i \mathbf{v}_i + x_i \mathbf{t}_i$, or equivalently, $\mathbf{A}_i \mathbf{v}_i - \mathbf{G}\hat{\mathbf{c}} = -x_i \mathbf{t}_i$ for all $i \in [\ell]$. We can express these ℓ relations as a linear system:

$$\underbrace{\left[\begin{array}{ccc|c} \mathbf{A}_1 & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{A}_\ell & -\mathbf{G} \end{array}\right]}_{\mathbf{B}_\ell} \cdot \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_\ell \\ \hat{\mathbf{c}} \end{bmatrix} = \begin{bmatrix} -x_1 \mathbf{t}_1 \\ \vdots \\ -x_\ell \mathbf{t}_\ell \end{bmatrix}. \tag{1.2}$$

Our goal now is to sample a random *short* tuple $(\mathbf{v}_1, \ldots, \mathbf{v}_\ell, \hat{\mathbf{c}})$ that satisfies Eq. (1.2). This can be done by giving out a random trapdoor for the matrix $\mathbf{B}_\ell$:

$$\mathbf{B}_\ell :- \left[\begin{array}{ccc|c} \mathbf{A}_1 & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{A}_\ell & -\mathbf{G} \end{array}\right]. \tag{1.3}$$

Using $\mathbf{B}_\ell$, we can sample a random short preimage $(\mathbf{v}_1, \ldots, \mathbf{v}_\ell, \hat{\mathbf{c}})$ satisfying Eq. (1.2). This yields the commitment $\mathbf{c} = \mathbf{G}\hat{\mathbf{c}}$ and the openings $\mathbf{v}_1, \ldots, \mathbf{v}_\ell$. In our

construction, we set the target vector $\mathbf{t}_i$ to the first basis vector $\mathbf{e}_1 = [1, 0, \ldots, 0]^\intercal$ for all $i \in [\ell]$. We now make the following observations:

- **Binding:** To argue that the scheme is binding, we require that it is hard to find a short vector $\mathbf{z}$ where $\underline{\mathbf{A}_i}\mathbf{z} = \mathbf{0}$ for any $i \in [\ell]$ even given the (related) matrix $\mathbf{B}_\ell$ and a trapdoor for $\mathbf{B}_\ell$. Here, $\underline{\mathbf{A}_i}$ denotes $\mathbf{A}_i$ with the first row removed. We refer to assumptions of this type as "basis-augmented SIS" (BASIS) assumptions (Assumption 3.2). As we sketch below (and show formally in Theorem 3.4), when $\mathbf{A}_1, \ldots, \mathbf{A}_\ell \xleftarrow{R} \mathbb{Z}_q^{n \times m}$ are *random*, this instantiation of the BASIS assumption holds under the *standard* SIS assumption. We refer to this instance of the BASIS assumption with random matrices as $\mathsf{BASIS}_{\mathsf{rand}}$. Now, to argue binding, we observe that an adversary that breaks binding is able to come up with an index $i \in [\ell]$, short vectors $\mathbf{v}, \mathbf{v}' \in \mathbb{Z}_q^m$ and values $x, x' \in \mathbb{Z}_q$ such that $\mathbf{c} = \mathbf{A}_i\mathbf{v} + x\mathbf{e}_1 = \mathbf{A}_i\mathbf{v}' + x'\mathbf{e}_1$. This means that

$$\mathbf{A}_i(\mathbf{v} - \mathbf{v}') = (x' - x)\mathbf{e}_1.$$

  As long as $x' - x \neq 0$, $\mathbf{v} - \mathbf{v}' \neq \mathbf{0}$, and so $\mathbf{v} - \mathbf{v}'$ is a SIS solution to $\underline{\mathbf{A}_i}$. Observe that this analysis does not impose *any* restriction on the magnitude of $x' - x$. This means our construction naturally supports committing to arbitrary vectors over $\mathbb{Z}_q$ as opposed to vectors with small entries.[10] We give the formal reduction to the $\mathsf{BASIS}_{\mathsf{rand}}$ assumption in the full version of this paper [WW22].
- **Private openings.** A vector commitment scheme supports private openings if the commitment $\mathbf{c}$ and any collections of openings $\{(i, x_i, \mathbf{v}_i)\}_{i \in S}$ completely hide the values x_j for $j \notin S$. Since we sample the commitment $\mathbf{c}$ and the openings $\mathbf{v}_i$ *jointly* in our approach, it is straightforward to argue (by appealing to properties of discrete Gaussians) that the commitment $\mathbf{c}$ is statistically close to uniform over $\mathbb{Z}_q^n$ and each opening $\mathbf{v}_i$ is statistically close to the distribution $\mathbf{A}_i^{-1}(\mathbf{c} - x_i\mathbf{t}_i)$. Thus our scheme provides *statistically* private openings out of the box.

Taken together, this yields a vector commitment from standard SIS that supports statistically private openings and commitments to arbitrary vectors over $\mathbb{Z}_q^\ell$. We give the full description and analysis in Section 3.

Reducing $\mathsf{BASIS}_{\mathsf{rand}}$ *to standard SIS.* As described above, the binding property of our vector commitment relies on the BASIS assumption, which says that SIS with respect to $\underline{\mathbf{A}_i}$ (i.e., $\mathbf{A}_i$ with the first row removed) is hard even given the related matrix $\mathbf{B}_\ell$ from Eq. (1.3) and a trapdoor for $\mathbf{B}_\ell$. As we show in Theorem 3.4, when the matrices $\mathbf{A}_1, \ldots, \mathbf{A}_\ell \xleftarrow{R} \mathbb{Z}_q^{n \times m}$ are uniform and independent, this assumption ($\mathsf{BASIS}_{\mathsf{rand}}$) reduces to the *standard* SIS assumption in a straightforward way. Here, we provide a sketch of the reduction. For ease of exposition, we show that SIS with respect to $\mathbf{A}_i$ (as opposed to $\underline{\mathbf{A}_i}$) is hard given a trapdoor for $\mathbf{B}_\ell$. We

[10] As discussed earlier, previous vector commitments [PPS21, ACL+22] based on SIS or its generalizations needed to assume small inputs for *both* correctness and security.

also describe the approach for the case $i = 1$, and refer to Theorem 3.4 for the full analysis.

The idea is simple: we set $\mathbf{A}_1$ to be the SIS challenge and sample matrices $\mathbf{A}_2, \ldots, \mathbf{A}_\ell$ together with trapdoors $\mathbf{R}_2, \ldots, \mathbf{R}_\ell$ (i.e., $\mathbf{A}_i\mathbf{R}_i = \mathbf{G}$). Let $\tilde{\mathbf{B}}_\ell$ be $\mathbf{B}_\ell$ with the first column block removed (i.e., the column block containing $\mathbf{A}_1$). Then, using $\mathbf{R}_2, \ldots, \mathbf{R}_\ell$ we can construct a trapdoor $\tilde{\mathbf{R}}_\ell$ for $\tilde{\mathbf{B}}_\ell$ (i.e., $\tilde{\mathbf{B}}_\ell\tilde{\mathbf{R}}_\ell = \mathbf{G}_{n\ell} = \mathbf{I}_{n\ell} \otimes \mathbf{g}^\intercal$):

$$\tilde{\mathbf{B}}_\ell = \left[\begin{array}{ccc|c} \mathbf{0} & \cdots & \mathbf{0} & -\mathbf{G} \\ \mathbf{A}_2 & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{A}_\ell & -\mathbf{G} \end{array}\right] \quad \text{and} \quad \tilde{\mathbf{R}}_\ell = \begin{bmatrix} -\mathbf{R}_2 & \mathbf{R}_2 & & \\ \vdots & & \ddots & \\ -\mathbf{R}_\ell & & & \mathbf{R}_\ell \\ -\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}$$

Using standard trapdoor extension techniques [ABB10a, ABB10b, CHKP10, MP12], we can *extend* $\tilde{\mathbf{R}}_\ell$ to obtain a trapdoor $\mathbf{R}_\ell$ for $\mathbf{B}_\ell$. This yields a $\mathsf{BASIS}_{\mathsf{rand}}$ instance (i.e., comprised of the matrix $\mathbf{A}_1$, the matrix $\mathbf{B}_\ell$, and the trapdoor $\mathbf{R}_\ell$). Thus, an adversary that breaks the $\mathsf{BASIS}_{\mathsf{rand}}$ assumption implies an adversary that breaks SIS (with comparable parameters). We give the formal analysis in Theorem 3.4.

Functional commitments using structured $\mathbf{A}_i$. Instantiating our framework with uniform $\mathbf{A}_i \xleftarrow{R} \mathbb{Z}_q^{n\times m}$ (as in PPS_1), we obtain a vector commitment scheme with private openings and supporting large messages from the standard SIS assumption. If we instead use a structured set of matrices $\mathbf{A}_i$ as in $\mathsf{MatrixACLMT}$, we obtain functional commitments, polynomial commitments, and aggregatable commitments.

We start by describing our functional commitment scheme. Our starting point is to consider the main verification relation from Eq. (1.1) and generalize it in two ways: (1) we replace the matrices $\mathbf{A}_1, \ldots, \mathbf{A}_\ell \in \mathbb{Z}_q^{n\times m}$ with *structured* matrices; and (2) we consider a *matrix* extension of the verification relation. In particular, we first sample $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}$. Then, for each $i \in [\ell]$, we sample an invertible matrix $\mathbf{W}_i \xleftarrow{R} \mathbb{Z}_q^{n\times n}$ and set $\mathbf{A}_i \leftarrow \mathbf{W}_i\mathbf{A}$. We now consider a matrix analog of the verification relation from Eq. (1.1) where each target vector $\mathbf{t}_i$ is replaced with the matrix $\mathbf{W}_i\mathbf{G}$ (this choice will be helpful for supporting *functional* openings). Our matrix verification relation is now

$$\mathbf{C} = \mathbf{A}_i\mathbf{V}_i + x_i\mathbf{W}_i\mathbf{G}. \tag{1.4}$$

Our goal now is to sample a tuple $(\mathbf{V}_1, \ldots, \mathbf{V}_\ell, \mathbf{C})$ that satisfy Eq. (1.4) for all $i \in [\ell]$ and where $\mathbf{V}_1, \ldots, \mathbf{V}_\ell \in \mathbb{Z}_q^{m\times m}$ are short. As before, the commitment $\mathbf{C}$ can be large and we specifically define it to be $\mathbf{C} = \mathbf{G}\hat{\mathbf{C}}$, where $\hat{\mathbf{C}} \in \mathbb{Z}_q^{m\times m}$ is short. This way, we can sample $\hat{\mathbf{C}}$ using an analogous trapdoor sampling procedure as before. Specifically, the trapdoor for the same matrix $\mathbf{B}_\ell$ from Eq. (1.3) allows us to jointly sample short openings $\mathbf{V}_1, \ldots, \mathbf{V}_\ell$ along with a

matrix $\hat{\mathbf{C}}$ that satisfy Eq. (1.4):

$$\mathbf{B}_\ell \begin{bmatrix} \mathbf{V}_1 \\ \vdots \\ \mathbf{V}_\ell \\ \hat{\mathbf{C}} \end{bmatrix} = \left[\begin{array}{ccc|c} \mathbf{A}_1 & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{A}_\ell & -\mathbf{G} \end{array}\right] \cdot \begin{bmatrix} \mathbf{V}_1 \\ \vdots \\ \mathbf{V}_\ell \\ \hat{\mathbf{C}} \end{bmatrix} = \begin{bmatrix} -x_1\mathbf{W}_1\mathbf{G} \\ \vdots \\ -x_\ell\mathbf{W}_\ell\mathbf{G} \end{bmatrix}. \tag{1.5}$$

By construction, for all $i \in [\ell]$, we have that $\mathbf{A}_i\mathbf{V}_i - \mathbf{G}\hat{\mathbf{C}} = -x_i\mathbf{W}_i\mathbf{G}$, or equivalently, $\mathbf{C} = \mathbf{G}\hat{\mathbf{C}} = \mathbf{A}_i\mathbf{V}_i + x_i\mathbf{W}_i\mathbf{G}$ and Eq. (1.4) holds. We now show that this directly extends to yield a succinct functional commitment. Since $\mathbf{A}_i = \mathbf{W}_i\mathbf{A}$ and $\mathbf{W}_i$ is invertible, we can rewrite Eq. (1.4) as

$$\mathbf{W}_i^{-1}\mathbf{C} = \mathbf{A}\mathbf{V}_i + x_i\mathbf{G},$$

where $\mathbf{V}_i$ is *short*. Readers familiar with the homomorphic encryption scheme of Gentry et al. [GSW13] or the homomorphic signature scheme of Gorbunov et al. [GVW15] may recognize that $\mathbf{W}_i^{-1}\mathbf{C}$ is an encryption of x_i under randomness $\mathbf{V}_i$ or that $\mathbf{V}_i$ is a signature on x_i under the verification key $\mathbf{W}_i^{-1}\mathbf{C}$. Thus, we can use the same lattice-based homomorphic evaluation machinery [GSW13, BGG+14] to homomorphically compute an opening to $f(\mathbf{x})$ for an arbitrary Boolean circuit $f\colon \{0,1\}^\ell \to \{0,1\}$.

In slightly more detail, let $\tilde{\mathbf{C}} = [\mathbf{W}_1^{-1}\mathbf{C} \mid \cdots \mid \mathbf{W}_\ell^{-1}\mathbf{C}]$ and $\tilde{\mathbf{V}} = [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_\ell]$. Then,

$$\mathbf{A}\tilde{\mathbf{V}} = \mathbf{A}[\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_\ell] = [\mathbf{W}_1^{-1}\mathbf{C} - x_1\mathbf{G} \mid \cdots \mid \mathbf{W}_\ell^{-1}\mathbf{C} - x_\ell\mathbf{G}] = \tilde{\mathbf{C}} - \mathbf{x}^\intercal \otimes \mathbf{G}.$$

Using the homomorphic evaluation techniques from [GSW13, BGG+14], there exists a short matrix $\mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}}$ that depends on $\tilde{\mathbf{C}}$, f, and $\mathbf{x}$ such that

$$(\tilde{\mathbf{C}} - \mathbf{x}^\intercal \otimes \mathbf{G}) \cdot \mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}} = \tilde{\mathbf{C}}_f - f(\mathbf{x}) \cdot \mathbf{G}, \tag{1.6}$$

where $\tilde{\mathbf{C}}_f$ is a matrix that can be efficiently computed from $\tilde{\mathbf{C}}$ and f. To open $\mathbf{C}$ to a function f, the user computes $\tilde{\mathbf{V}}_f \leftarrow \tilde{\mathbf{V}} \cdot \mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}}$. To verify a candidate value $y \in \{0,1\}$ with respect to a function f and commitment $\mathbf{C}$, the verifier first computes $\tilde{\mathbf{C}}_f$ from $(\mathbf{C}, \mathbf{W}_1, \ldots, \mathbf{W}_\ell, f)$ and then checks that $\tilde{\mathbf{V}}_f$ is short and moreover,

$$\mathbf{A}\tilde{\mathbf{V}}_f = \tilde{\mathbf{C}}_f - y \cdot \mathbf{G}.$$

For correctness, observe that

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

For binding, we require that SIS is hard with respect to $\mathbf{A}$ even given the matrix $\mathbf{B}_\ell$ and a trapdoor for $\mathbf{B}_\ell$. We refer to this instance of the BASIS assumption with structured $\mathbf{A}_i$'s as $\mathsf{BASIS}_{\mathsf{struct}}$. Since the matrices $\mathbf{A}_i$ that comprise $\mathbf{B}_\ell$ are now correlated, we do not know how to reduce $\mathsf{BASIS}_{\mathsf{struct}}$ to the standard SIS assumption. Nonetheless, $\mathsf{BASIS}_{\mathsf{struct}}$ is a falsifiable assumption under which we obtain a succinct functional commitment for all bounded-depth Boolean circuits. This is the first succinct functional commitment for general circuits from a falsifiable assumption. We provide the full description in Section 4 and a comparison to previous succinct functional commitments in Table 1.

Functional commitments with private openings. Using the approach from [GVW15] for constructing context-hiding homomorphic signatures [GVW15], we can easily extend our functional commitment scheme above to support *private* openings (i.e., where the commitment $\mathbf{C}$ and the opening $\tilde{\mathbf{V}}_f$ reveals nothing more about the input $\mathbf{x}$ other than the value $f(\mathbf{x})$). We sketch the approach here. Let $\mathbf{C}$ be a commitment to $\mathbf{x}$ and let $\tilde{\mathbf{V}}_f = \tilde{\mathbf{V}} \cdot \mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}}$ be the opening computed as described above. Then, define the matrix $\mathbf{D}_f$ to be

$$\mathbf{D}_f = [\mathbf{A} \mid \tilde{\mathbf{C}}_f + (f(\mathbf{x}) - 1) \cdot \mathbf{G}] = [\mathbf{A} \mid \mathbf{A}\tilde{\mathbf{V}}_f + (2f(\mathbf{x}) - 1) \cdot \mathbf{G}].$$

Since $\tilde{\mathbf{V}}_f$ is short and $2f(\mathbf{x}) - 1 \in \{-1, 1\}$, the matrix $\begin{bmatrix} \tilde{\mathbf{V}}_f \\ -\mathbf{I}_n \end{bmatrix}$ is a trapdoor for $\mathbf{D}_f$. We now include a random target $\mathbf{u} \in \mathbb{Z}_q^n$ as part of the CRS, and define the opening to be a *random* short vector $\mathbf{v}_f$ where $\mathbf{D}_f \mathbf{v}_f = \mathbf{u}$. The honest prover samples $\mathbf{v}_f$ using the trapdoor for $\mathbf{D}_f$ (derived from $\tilde{\mathbf{V}}_f$). To check an opening $\mathbf{v}_f$ is a valid opening to a value $y \in \{0, 1\}$ with respect to a function f and commitment $\mathbf{C}$, the verifier computes $\tilde{\mathbf{C}}_f$ from $(\mathbf{C}, \mathbf{W}_1, \ldots, \mathbf{W}_\ell, f)$ as before, defines the matrix $\mathbf{D}_f = [\mathbf{A} \mid \tilde{\mathbf{C}}_f + (y-1) \cdot \mathbf{G}]$, and finally, checks that $\mathbf{D}_f \mathbf{v}_f = \mathbf{u}$. To argue that $\mathbf{v}_f$ hides all information about $\mathbf{x}$ other than what is revealed by $f(\mathbf{x})$, observe that the matrix $\mathbf{D}_f$ depends only on $f(\mathbf{x})$ and *not* $\mathbf{x}$. Thus, given a trapdoor for $\mathbf{A}$ (which can be extended into a trapdoor for $\mathbf{D}_f$ for all f), and the value $f(\mathbf{x})$, we can sample a short $\mathbf{v}_f$ such that $\mathbf{D}_f \mathbf{v}_f = \mathbf{u}$ whose distribution is statistically close to the real opening. This latter procedure only depends on $f(\mathbf{x})$ and not $\mathbf{x}$, so hiding follows. While this construction is hiding, we do not know how to show that it is binding; however, it does satisfy the *weaker* notion of *target binding* where binding holds for all honestly-generated commitments. We provide the full details and analysis in the full version of this paper [WW22].

Polynomial commitments. We can obtain a polynomial commitment over $\mathbb{Z}_q$ via a simple adaptation of our functional commitment. The starting point is to construct a functional commitment scheme for linear functions on $\mathbb{Z}_q^\ell$ (as opposed to a function on the binary domain $\{0,1\}^\ell$). We first consider linear functions with small coefficients. Let $\mathbf{z} \in \{0,1\}^\ell$ be a vector and define the linear function $f_{\mathbf{z}}(\mathbf{x}) \coloneqq \mathbf{z}^\intercal \mathbf{x}$. We use the same commitment and opening structure as in our functional commitment. Namely, a commitment $\mathbf{C}$ and the openings $\mathbf{V}_1, \ldots, \mathbf{V}_\ell$ for an input $\mathbf{x}$ satisfy $\mathbf{A}\mathbf{V}_i = \mathbf{W}_i^{-1}\mathbf{C} - x_i\mathbf{G}$, where $x_i \in \mathbb{Z}_q$ now. Observe that

$$\underbrace{\sum_{i \in [\ell]} z_i \mathbf{A}\mathbf{V}_i}_{\mathbf{A}\mathbf{V}_{\mathbf{z}}} = \sum_{i \in [\ell]} z_i \mathbf{W}_i^{-1}\mathbf{C} - \sum_{i \in [\ell]} z_i x_i \mathbf{G} = \underbrace{\sum_{i \in [\ell]} z_i \mathbf{W}_i^{-1}\mathbf{C}}_{\tilde{\mathbf{C}}_{\mathbf{z}}} - \underbrace{(\mathbf{z}^\intercal \mathbf{x}) \cdot \mathbf{G}}_{f_{\mathbf{z}}(\mathbf{x}) \cdot \mathbf{G}}.$$

Thus, $\mathbf{V}_{\mathbf{z}} = \sum_{i \in [\ell]} z_i \mathbf{V}_i$ is an opening to the function $f_{\mathbf{z}}$. Here, we need $\mathbf{z} \in \{0,1\}^\ell$ to be short so $\mathbf{V}_{\mathbf{z}}$ is short. To extend to arbitrary linear functions over $\mathbb{Z}_q^\ell$ (rather than short ones), we rely on standard binary decomposition and blow up the vector dimension by a factor of $O(\log q)$. Namely, to commit to a vector $\mathbf{x}$, the

Scheme	CRS Size	Function Class	Assumption	Fast Verification
Folklore	$\mathsf{poly}(\lambda, \log \ell)$	Boolean circuits	CRHF + SNARK*	✓
[LRY16]	$O(\ell)$	linear functions	bilinear maps	✓
[PPS21]†	$s \cdot \mathsf{poly}(\lambda, d)$	depth d Boolean circuits‡	SIS	✗
[ACL+22]	$\ell^{2d} \cdot \mathsf{poly}(\lambda)$	degree d polynomials§	k-R-ISIS	✓
This work	$\ell^2 \cdot \mathrm{poly}(\lambda, d, \log \ell)$	depth d Boolean circuits	$\mathsf{BASIS}_{\mathsf{struct}}$	✗
This work	$\ell^2 \cdot \mathrm{poly}(\lambda, \log \ell)$	linear functions	$\mathsf{BASIS}_{\mathsf{struct}}$	✓

* Collision-resistant hash functions (CRHFs) together with a succinct non-interactive argument of knowledge (SNARK).
† This scheme is in a significantly weaker model that requires an *online trusted* authority to issue opening keys.
‡ The Boolean circuit has size at most s.
§ Only supports commitments and openings to *small* values.

Table 1: Summary of *succinct* functional commitments. For each scheme, we report the size of the CRS as a function of the security parameter λ and the input length ℓ. We say that a scheme supports "fast verification" if after an *input-independent* preprocessing step, the verification running time is sublinear in ℓ. In all schemes, the size of the commitment and the openings are polylogarithmic in the input length ℓ.

user now commits to $\mathbf{x} \otimes \mathbf{g}^\intercal$, and to open to a linear function $f_{\mathbf{z}}$ where $\mathbf{z} \in \mathbb{Z}_q^n$, the user constructs an opening with respect to $f_{\mathbf{g}^{-1}(\mathbf{z})}$. This yields a functional commitment scheme for linear functions over $\mathbb{Z}_q^\ell$.

As observed by Libert et al. [LRY16], a functional commitment scheme for linear functions over $\mathbb{Z}_q^\ell$ directly implies a polynomial commitment over $\mathbb{Z}_q$ for polynomials of degree up to $d = \ell - 1$. Namely, a commitment to a polynomial $f \in \mathbb{Z}_q[x]$ of degree d is a vector commitment to the coefficients of f. To open the commitment to a point $x \in \mathbb{Z}_q$, the user constructs a linear opening with respect to the evaluation vector $[1, x, x^2, \ldots, x^d]$. For this to work, it is critical that our functional commitment for linear functions over $\mathbb{Z}_q^\ell$ supports committing to and opening to *arbitrary* $\mathbb{Z}_q$ values (and not just *small* values). Thus, we obtain a polynomial commitment scheme where the commitment size and the opening size is $\mathsf{poly}(\lambda, \log d)$. We provide the full details in the full version of this paper [WW22].

Aggregatable commitments. Another application of using structured matrices $\mathbf{A}_i$ is it immediately gives an aggregatable commitment. As before, we instantiate our framework with $\mathbf{A}_i = \mathbf{W}_i \mathbf{A}$. We also sample target vectors $\mathbf{u}_1, \ldots, \mathbf{u}_\ell \xleftarrow{R} \mathbb{Z}_q^n$ and include them as part of the CRS. To commit to an input $\mathbf{x} \in \mathbb{Z}_q^\ell$, we sample

$(\mathbf{v}_1, \ldots, \mathbf{v}_\ell, \mathbf{c})$ where

$$\mathbf{B}_\ell \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_\ell \\ \hat{\mathbf{c}} \end{bmatrix} = \left[\begin{array}{ccc|c} \mathbf{A}_1 & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{A}_\ell & -\mathbf{G} \end{array}\right] \cdot \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_\ell \\ \hat{\mathbf{c}} \end{bmatrix} = \begin{bmatrix} -x_1 \mathbf{W}_1 \mathbf{u}_1 \\ \vdots \\ -x_\ell \mathbf{W}_\ell \mathbf{u}_\ell \end{bmatrix}.$$

Let $\mathbf{c} = \mathbf{G}\hat{\mathbf{c}}$. Then, for all $i \in [\ell]$, $\mathbf{A}_i \mathbf{v}_i - \mathbf{c} = -x_i \mathbf{W}_i \mathbf{u}_i$, or equivalently, $\mathbf{W}_i^{-1}\mathbf{c} = \mathbf{A}\mathbf{v}_i + x_i \mathbf{u}_i$. Observe now that this scheme immediately supports aggregation: for any set $S \subseteq [\ell]$,

$$\sum_{i \in S} \mathbf{W}_i^{-1}\mathbf{c} = \mathbf{A} \sum_{i \in S} \mathbf{v}_i - \sum_{i \in S} x_i \mathbf{u}_i.$$

Thus, $\sum_{i \in S} \mathbf{v}_i$ is an opening to all of the indices in S. We show in the full version of this paper [WW22] that under the same $\mathsf{BASIS}_{\mathsf{struct}}$ assumption (i.e., SIS is hard with respect to $\mathbf{A}$ given $\mathbf{B}_\ell$ and a trapdoor for $\mathbf{B}_\ell$), this scheme satisfies "same-set binding." This means no efficient adversary can open a commitment $\mathbf{c}$ to different sets of values $\{(i, x_i)\}_{i \in S}$ and $\{(i, x_i')\}_{i \in S}$ for the *same* set S. Unlike our vector commitment construction, the security of our aggregatable construction only holds when the input vector $\mathbf{x}$ is short (i.e., our reduction to the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption in the full version of this paper [WW22] constructs an SIS solution whose norm scales with the magnitude of the opened values).

Our aggregatable vector commitment scheme does not satisfy the stronger notion of "different-set binding." This means an efficient adversary may be able to construct a commitment $\mathbf{c}$ along with valid openings $\{(i, x_i)\}_{i \in S}$ and $\{(i, x_i')\}_{i \in T}$ to (distinct) sets S and T, respectively, such that $x_i = 0$ and $x_i' = 1$. Indeed in the full version of this paper [WW22], we describe an explicit attack where an adversary can use the trapdoor for $\mathbf{B}_\ell$ to (heuristically) obtain a trapdoor for the matrix $[\mathbf{W}_S^{-1}\mathbf{A} \mid \mathbf{W}_T^{-1}\mathbf{A}]$ whenever $S \neq T$ and where $\mathbf{W}_S = \sum_{i \in S} \mathbf{W}_i^{-1}$ and $\mathbf{W}_T = \sum_{i \in T} \mathbf{W}_i^{-1}$. Knowledge of this trapdoor allows an adversary to construct a valid opening to $\{(i, x_i)\}_{i \in S}$ and $\{(i, x_i')\}_{i \in T}$ for any choice of x_i, x_i'.

Conceptually, our approach for constructing an aggregatable vector commitment scheme is to replace the *fixed* target value $x_i \mathbf{e}_1$ from our basic vector commitment with *random* linear combinations of $\{x_i\}_{i \in S}$ (where the coefficients of the random linear combination are the vectors $\{\mathbf{u}_i\}_{i \in S}$). A similar approach was used for aggregating pairing-based signatures in [BDN18] and for aggregating openings (to constant-degree polynomials) in [ACL+22].

Aggregating functional commitments. The same aggregation technique applies to our succinct functional commitment scheme. Recall the functional commitment verification relation from Eq. (1.6): $\mathbf{A}\tilde{\mathbf{V}}_f = \tilde{\mathbf{C}}_f - y \cdot \mathbf{G}$. Here $\tilde{\mathbf{V}}_f$ is the opening, $\tilde{\mathbf{C}}_f$ is a function of the commitment $\mathbf{C}$ and the function f, and y is the value. To support aggregating up to t openings, we include random vectors $\mathbf{u}_1, \ldots, \mathbf{u}_t \xleftarrow{R} \mathbb{Z}_q^n$ in the CRS. Then, given a collection of openings $(f_1, y_1, \tilde{\mathbf{V}}_1), \ldots, (f_t, y_t, \tilde{\mathbf{V}}_t)$ where

the functions $f_1, \ldots, f_t$ are sorted in lexicographic order, we define the aggregate opening to be $\mathbf{v} = \sum_{i \in [t]} \tilde{\mathbf{V}}_i \cdot \mathbf{G}^{-1}(\mathbf{u}_i)$. The new verification relation is then

$$\sum_{i \in [t]} \tilde{\mathbf{C}}_{f_i} \cdot \mathbf{G}^{-1}(\mathbf{u}_i) = \mathbf{A}\mathbf{v} - \sum_{i \in [t]} y_i \mathbf{u}_i.$$

Similar to the case with aggregatable vector commitments, we can argue "same-function binding," where no efficient adversary can open a commitment $\mathbf{C}$ to two different sets of values $(y_1, \ldots, y_t) \neq (y'_1, \ldots, y'_t)$ with respect to the same set of functions $(f_1, \ldots, f_t)$. We provide the full analysis in the full version of this paper [WW22].

Understanding the BASIS *assumption.* The BASIS assumptions we introduce in this work enable a number of new constructions of vector commitments and their generalizations. While the basic version $\mathsf{BASIS}_{\mathsf{rand}}$ that suffices for vector commitments can be reduced to the standard SIS assumption (Theorem 3.4), the more general version $\mathsf{BASIS}_{\mathsf{struct}}$ with structured matrices does not. Nonetheless, the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption is still falsifiable and thus, yields the first succinct functional commitments and polynomial commitments from falsifiable lattice assumptions. We invite cryptanalysis of our new family of SIS assumptions and are also optimistic that the assumption as well as our general methodology will be helpful for realizing new lattice-based cryptographic primitives.

In Section 6, we compare the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption to similar assumptions made in [ACL+22]. We show a close connection between the two families of assumptions. We can also view the BASIS assumptions as a new type of "q-type" assumption in the lattice-based setting (where the size of the assumption grows with the input dimension).

1.3 Related Work and Concurrent Work

Functional commitments have also been extensively studied in the *interactive* model. In these settings, there is typically an *interactive* opening procedure between the committer and the verifier. Ishai et al. [IKO07] introduced interactive functional commitments for linear function, and subsequently, Bitansky and Chiesa [BC12] extended it to interactive functional commitments. In both cases, these were used to construct (interactive) succinct arguments without relying on probabilistically-checkable proofs (PCPs). Alternatively, using PCPs or their generalization to interactive oracle proofs [BCS16], we can also construct a functional commitment from any collision-resistant hash function via Kilian's interactive succinct argument [Kil92], which can then be made non-interactive in the random oracle model [Mic00]. Our focus in this work is on *non-interactive* vector and functional commitments in the plain model (i.e., *without* random oracles).

Concurrent works. Recently, two concurrent works [dCP23, BCFL22] introduced new constructions of functional commitments. We provide a more detailed comparison with these works in the full version of this paper [WW22].

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 integers $a, b \in \mathbb{N}$, we write $[a, b]$ to denote the set $\{a, a+1, \ldots, b\}$. 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: $\mathbf{v} = (v_1, \ldots, v_n)$. For matrices $\mathbf{A}_1, \ldots, \mathbf{A}_\ell \in \mathbb{Z}_q^{n \times m}$, we write $\mathsf{diag}(\mathbf{A}_1, \ldots, \mathbf{A}_\ell) \in \mathbb{Z}_q^{n\ell \times m\ell}$ to denote the block diagonal matrix with blocks $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$ along the main diagonal (and 0 elsewhere).

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 say they are statistically indistinguishable if the statistical distance $\Delta(\mathcal{D}_1, \mathcal{D}_2)$ is bounded by a negligible function $\mathsf{negl}(\lambda)$.

We review additional preliminaries, especially on lattice-based cryptography in the full version of this paper [WW22].

3 Vector Commitments with Private Opening from SIS

In this section, we show how to construct a vector commitment with private openings from the standard SIS assumption. We start by recalling the definition of a vector commitment:

Definition 3.1 (Vector Commitment). *A vector commitment scheme with succinct local openings over a message space $\mathcal{M}$ consists of a tuple of efficient algorithms $\Pi_{\mathsf{VC}} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Verify})$ with the following properties:*

- $\mathsf{Setup}(1^\lambda, 1^\ell) \to \mathsf{crs}$*: On input the security parameter λ and the vector length ℓ, 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 a vector $\mathbf{x} \in \mathcal{M}^\ell$, the commit algorithm outputs a commitment σ and a state* st.
- $\mathsf{Open}(\mathsf{st}, i) \to \pi$*: On input a commitment state* st *and an index $i \in [\ell]$, the open algorithm outputs an opening π.*
- $\mathsf{Verify}(\mathsf{crs}, \sigma, i, x_i, \pi) \to \{0, 1\}$*: On input the common reference string* crs*, a commitment σ, an index i, a message $x_i \in \mathcal{M}$, and an opening π, the verification algorithm outputs a bit $b \in \{0, 1\}$.*

We now define several standard properties on vector commitment schemes:

– **Correctness:** *For all security parameters λ, vector dimensions ℓ, and inputs $\mathbf{x} = (x_1, \ldots, x_\ell) \in \mathcal{M}^\ell$,*

$$\Pr\left[\mathsf{Verify}(\mathsf{crs}, \sigma, i, m_i, \pi) = 1 : \begin{array}{c}\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\ell);\\ (\sigma, \mathsf{st}) \leftarrow \mathsf{Commit}(\mathsf{crs}, \mathbf{x});\\ \pi \leftarrow \mathsf{Open}(\mathsf{st}, i)\end{array}\right] = 1 - \mathsf{negl}(\lambda).$$

– **Succinctness:** *The vector commitment scheme is succinct if there exists a universal polynomial $\mathsf{poly}(\cdot)$ such that for all $\lambda \in \mathbb{N}$, $|\sigma| = \mathsf{poly}(\lambda, \log \ell)$ and $|\pi| = \mathsf{poly}(\lambda, \log \ell)$ in the correctness definition.*

– **Binding:** *We say the commitment scheme is statistically binding (resp., computationally binding) if for all polynomials $\ell = \ell(\lambda)$ and all adversaries $\mathcal{A}$ (resp., efficient adversaries $\mathcal{A}$),*

$$\Pr\left[\begin{array}{c}\mathsf{Verify}(\mathsf{crs}, \sigma, i, x_i, \pi) = 1\\ \text{and } x_i \neq x_i' \text{ and}\\ \mathsf{Verify}(\mathsf{crs}, \sigma, i, x_i', \pi') = 1\end{array} : \begin{array}{c}\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\ell);\\ \big(\sigma, i, (x_i, \pi), (x_i', \pi')\big) \leftarrow \mathcal{A}(1^\lambda, 1^\ell, \mathsf{crs})\end{array}\right] = \mathsf{negl}(\lambda).$$

– **Private openings:** *For a vector dimension ℓ, an adversary $\mathcal{A}$, and a simulator $\mathcal{S} = (\mathcal{S}_0, \mathcal{S}_1)$, we define two distributions $\mathsf{Real}_{\mathcal{A}}(1^\lambda, 1^\ell)$ and $\mathsf{Ideal}_{\mathcal{A},\mathcal{S}}(1^\lambda, 1^\ell)$ as follows:*

$\mathsf{Real}_{\mathcal{A}}(1^\lambda)$*:*

1. *Give* $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\ell)$ *to* $\mathcal{A}$.
2. *Algorithm* $\mathcal{A}$ *outputs an input* $\mathbf{x} \in \mathcal{M}^\ell$.
3. *Compute* $(\sigma, \mathsf{st}) \leftarrow \mathsf{Commit}(\mathsf{crs}, \mathbf{x})$ *and give* σ *to* $\mathcal{A}$.
4. *Algorithm* $\mathcal{A}$ *outputs an index* $i \in [\ell]$.
5. *Give* $\pi_i \leftarrow \mathsf{Open}(\mathsf{st}, i)$ *to* $\mathcal{A}$.
6. *Algorithm* $\mathcal{A}$ *outputs a bit* $b \in \{0, 1\}$ *which is the output of the experiment.*

$\mathsf{Ideal}_{\mathcal{A},\mathcal{S}}(1^\lambda)$*:*

1. *Sample* $(\mathsf{crs}, \sigma, \mathsf{st}) \leftarrow \mathcal{S}_0(1^\lambda, 1^\ell)$ *and give* crs *to* $\mathcal{A}$.
2. *Algorithm* $\mathcal{A}$ *outputs an input* $\mathbf{x} \in \mathcal{M}^\ell$.
3. *Give* σ *to* $\mathcal{A}$.
4. *Algorithm* $\mathcal{A}$ *outputs an index* $i \in [\ell]$.
5. *Compute* $\pi_i \leftarrow \mathcal{S}_1(\mathsf{st}, i, x_i)$ *and give* π_i *to* $\mathcal{A}$.
6. *Algorithm* $\mathcal{A}$ *outputs a bit* $b \in \{0, 1\}$ *which is the output of the experiment.*

We say that the vector commitment scheme has statistically (resp., computationally) private openings if for all polynomials $\ell = \ell(\lambda)$ and adversaries $\mathcal{A}$ (resp., efficient adversaries $\mathcal{A}$), there exists an efficient simulator $\mathcal{S} = (\mathcal{S}_0, \mathcal{S}_1)$ such that $\mathsf{Real}_{\mathcal{A}}(1^\lambda, 1^\ell)$ and $\mathsf{Ideal}_{\mathcal{A},\mathcal{S}}(1^\lambda, 1^\ell)$ are statistically (resp., computationally) indistinguishable.

3.1 The Basis-Augmented SIS (BASIS) Assumption

In this section, we introduce the family of SIS assumptions that we use to build our vector commitment schemes. At a high level, our assumptions assert that the SIS problem is hard with respect to a random matrix $\mathbf{A}$ even given a trapdoor for a matrix $\mathbf{B}$ that is correlated with $\mathbf{A}$. We refer to our family of assumptions as the "basis-augmented SIS" (BASIS) assumption. As we discuss below (Theorem 3.4), some versions of the BASIS assumption can be reduced

to the standard SIS assumption. For instance, our first construction of a vector commitments with *private* openings (Construction 3.5) relies on a version that reduces to the standard SIS assumption.

Assumption 3.2 (BASIS Assumption). Let λ be a security parameter and $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$, and $\beta = \beta(\lambda)$ be lattice parameters. Let $s = s(\lambda)$ be a Gaussian width parameter. Let Samp be an efficient sampling algorithm that takes as input a security parameter λ and a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and outputs a matrix $\mathbf{B} \in \mathbb{Z}_q^{n' \times m'}$ along with auxiliary input aux. We say that the basis-augmented SIS (BASIS) assumption holds with respect to Samp if for all efficient adversaries $\mathcal{A}$,

$$\Pr\left[\mathbf{A}\mathbf{x} = \mathbf{0} \text{ and } 0 < \|\mathbf{x}\| \leq \beta : \begin{array}{c} \mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n \times m}; \\ (\mathbf{B}, \mathsf{aux}) \leftarrow \mathsf{Samp}(1^\lambda, \mathbf{A}), \mathbf{T} \leftarrow \mathbf{B}_s^{-1}(\mathbf{G}_{n'}); \\ \mathbf{x} \leftarrow \mathcal{A}(1^\lambda, \mathbf{A}, \mathbf{B}, \mathbf{T}, \mathsf{aux}) \end{array}\right] = \mathsf{negl}(\lambda).$$

In other words, we require that SIS is hard with respect to $\mathbf{A}$ even given a trapdoor $\mathbf{T}$ for the related matrix $\mathbf{B}$.

Assumption 3.3 (BASIS Assumption Instantiations). Let λ be a security parameter and $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$, and $\beta = \beta(\lambda)$ be lattice parameters. Let $s = s(\lambda)$ be a Gaussian width parameter and $\ell = \ell(\lambda)$ be a dimension. We consider two concrete instantiations of the BASIS assumption:

- $\mathsf{BASIS}_{\mathsf{rand}}$: The sampling algorithm $\mathsf{Samp}(1^\lambda, \mathbf{A})$ samples $i^* \xleftarrow{R} [\ell]$, $\mathbf{A}_i \xleftarrow{R} \mathbb{Z}_q^{(n+1) \times m}$ for all $i \neq i^*$, $\mathbf{a} \xleftarrow{R} \mathbb{Z}_q^m$, sets $\mathbf{A}_{i^*} \leftarrow \left[\begin{smallmatrix} \mathbf{a}^\mathsf{T} \\ \mathbf{A} \end{smallmatrix}\right]$, and outputs

$$\mathbf{B}_\ell = \left[\begin{array}{ccc|c} \mathbf{A}_1 & & & -\mathbf{G}_{n+1} \\ & \ddots & & \vdots \\ & & \mathbf{A}_\ell & -\mathbf{G}_{n+1} \end{array}\right] \quad \text{and} \quad \mathsf{aux} = i^*.$$

  We refer to this assumption as "the BASIS assumption with random matrices."
- $\mathsf{BASIS}_{\mathsf{struct}}$: The sampling algorithm $\mathsf{Samp}(1^\lambda, \mathbf{A})$ samples $\mathbf{W}_i \xleftarrow{R} \mathbb{Z}_q^{n \times n}$ for all $i \in [\ell]$ and outputs

$$\mathbf{B}_\ell = \left[\begin{array}{ccc|c} \mathbf{W}_1\mathbf{A} & & & -\mathbf{G}_n \\ & \ddots & & \vdots \\ & & \mathbf{W}_\ell\mathbf{A} & -\mathbf{G}_n \end{array}\right] \quad \text{and} \quad \mathsf{aux} = (\mathbf{W}_1, \ldots, \mathbf{W}_\ell).$$

  This is essentially $\mathsf{BASIS}_{\mathsf{rand}}$ with *structured* matrices $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$. We refer to this assumption as "the BASIS assumption with structured matrices."

Each of the above assumptions is parameterized by the tuple of parameters $(n, m, q, \beta, s, \ell)$. Strictly speaking, in both cases above, the auxiliary information aux can be efficiently computed directly from $\mathbf{A}$ and $\mathbf{B}_\ell$, and thus, can be safely omitted. We include them here for notational convenience.

Hardness of $\mathsf{BASIS}_{\mathsf{rand}}$. The $\mathsf{BASIS}_{\mathsf{rand}}$ assumption can be reduced to SIS. We state the theorem below and give the proof in the full version of this paper [WW22].

Theorem 3.4 (Hardness of $\mathsf{BASIS}_{\mathsf{rand}}$). *Let λ be a security parameter and $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$, $\beta = \beta(\lambda)$ be lattice parameters. Take any polynomial $\ell = \ell(\lambda)$ and suppose $n \geq \lambda$, $m \geq O(n \log q)$, and $s \geq O(\ell m \log(n\ell))$. Then, under the $\mathsf{SIS}_{n,m,q,\beta}$ assumption, the $\mathsf{BASIS}_{\mathsf{rand}}$ assumption with parameters $(n, m, q, \beta, s, \ell)$ holds.*

3.2 Vector Commitments with Private Opening from SIS

We now show how to construct a vector commitment scheme with statistically private openings from the $\mathsf{BASIS}_{\mathsf{rand}}$ assumption. By Theorem 3.4, we can in turn base hardness on the standard SIS assumption (with polynomial modulus).

Construction 3.5 (Vector Commitments from SIS). Let λ be a security parameter and $n = n(\lambda)$, $m = m(\lambda)$, and $q = q(\lambda)$ be lattice parameters. Let $m' = n(\lceil \log q \rceil + 1)$ and $B = B(\lambda)$ be a bound. Let $s_0 = s_0(\lambda)$, $s_1 = s_1(\lambda)$ be Gaussian width parameters. Let ℓ be the vector dimension. We construct a vector commitment scheme $\Pi_{\mathsf{VC}} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Verify})$ for $\mathbb{Z}_q^\ell$ as follows:

- $\mathsf{Setup}(1^\lambda, 1^\ell)$: On input the security parameter λ and the vector dimension ℓ, the setup algorithm samples $(\mathbf{A}_i, \mathbf{R}_i) \leftarrow \mathsf{TrapGen}(1^n, q, m)$ for each $i \in [\ell]$. Then, it constructs matrices $\mathbf{B}_\ell$ and $\mathbf{R}$ where

$$\mathbf{B}_\ell = \left[\begin{array}{ccc|c} \mathbf{A}_1 & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{A}_\ell & -\mathbf{G} \end{array}\right] \quad \text{and} \quad \mathbf{R} = \left[\begin{array}{c} \mathrm{diag}(\mathbf{R}_1, \ldots, \mathbf{R}_\ell) \\ \hline \mathbf{0}^{m' \times \ell m'} \end{array}\right]. \tag{3.1}$$

  Finally, the setup algorithm samples $\mathbf{T} \leftarrow \mathsf{SamplePre}(\mathbf{B}_\ell, \mathbf{R}, \mathbf{G}_{n\ell}, s_0)$ and outputs the common reference string $\mathsf{crs} = (\mathbf{A}_1, \ldots, \mathbf{A}_\ell, \mathbf{T})$.
- $\mathsf{Commit}(\mathsf{crs}, \mathbf{x})$: On input the common reference string $\mathsf{crs} = (\mathbf{A}_1, \ldots, \mathbf{A}_\ell, \mathbf{T})$ and a vector $\mathbf{x} \in \mathbb{Z}_q^\ell$, the commit algorithm constructs $\mathbf{B}_\ell$ from $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$ according to Eq. (3.1) and then uses $\mathbf{T}$ to sample

$$\begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_\ell \\ \hat{\mathbf{c}} \end{bmatrix} \leftarrow \mathsf{SamplePre}(\mathbf{B}_\ell, \mathbf{T}, -\mathbf{x} \otimes \mathbf{e}_1, s_1), \tag{3.2}$$

  where $\mathbf{e}_1 = [1, 0, \ldots, 0]^\intercal \in \mathbb{Z}_q^m$ is the first standard basis vector. It computes $\mathbf{c} \leftarrow \mathbf{G}\hat{\mathbf{c}} \in \mathbb{Z}_q^n$ and outputs the commitment $\sigma = \mathbf{c}$ and the state $\mathsf{st} = (\mathbf{v}_1, \ldots, \mathbf{v}_\ell)$.
- $\mathsf{Open}(\mathsf{st}, i)$: On input the state $\mathsf{st} = (\mathbf{v}_1, \ldots, \mathbf{v}_\ell)$ and the index $i \in [\ell]$, the opening algorithm outputs $\mathbf{v}_i$.

– $\mathsf{Verify}(\mathsf{crs}, \sigma, i, x_i, \pi)$: On input the common reference string $\mathsf{crs} = (\mathbf{A}_1, \ldots, \mathbf{A}_\ell, \mathbf{T})$, a commitment $\sigma = \mathbf{c} \in \mathbb{Z}_q^n$, an index $i \in [\ell]$, a message $x_i \in \mathbb{Z}_q$, and an opening $\pi = \mathbf{v}_i$, the verification algorithm outputs 1 if

$$\|\mathbf{v}_i\| \leq B \quad \text{and} \quad \mathbf{c} = \mathbf{A}_i \mathbf{v}_i + x_i \mathbf{e}_1.$$

Due to space limitations, we defer the formal analysis of Construction 3.5 to the full version of this paper [WW22] and simply state the main result below:

Corollary 3.6 (Vector Commitments with Private Openings from SIS). *Let λ be a security parameter. Then, for all polynomials $\ell = \ell(\lambda)$, under the SIS assumption with a polynomial norm bound $\beta = \mathsf{poly}(\lambda, \ell)$ and a polynomial modulus $q = \mathsf{poly}(\lambda, \ell)$, there exists a vector commitment scheme over $\mathbb{Z}_q^\ell$ that is computationally binding and has statistically private openings. The size of a commitment to a vector $\mathbf{x} \in \mathbb{Z}_q^\ell$ has size $O(\lambda(\log \lambda + \log \ell))$ and the openings have size $O(\lambda(\log^2 \lambda + \log^2 \ell))$. The size of the CRS is $\ell^2 \cdot \mathsf{poly}(\lambda, \log \ell)$.*

Extensions: linear homomorphism and updatability. Similar to the non-private scheme of Peikert et al. [PPS21], our vector commitment scheme is linearly homomorphic and supports stateless updates. We provide more details in the full version of this paper [WW22].

4 Succinct Functional Commitments for Circuits

In this section, we show how to obtain a succinct functional commitment for general circuits from the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption. We consider schemes where the parameters scale with the *depth* of the Boolean circuit. We start with the formal definition:

Definition 4.1 (Succinct Functional Commitment). *Let λ be a security parameter. Let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ be a family of functions $f : \{0,1\}^\ell \to \{0,1\}$ on inputs of length $\ell = \ell(\lambda)$ and which can be computed by Boolean circuits of depth at most $d = d(\lambda)$. 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:*

– $\mathsf{Setup}(1^\lambda, 1^\ell, 1^d) \to \mathsf{crs}$*: On input the security parameter λ, the input length ℓ, and the bound on the circuit depth d, 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 $\mathbf{x} \in \{0,1\}^\ell$, 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.*
– $\mathsf{Verify}(\mathsf{crs}, \sigma, f, y, \pi) \to \{0,1\}$*: On input the common reference string* crs*, a commitment σ, a function $f \in \mathcal{F}$, a value $y \in \{0,1\}$, 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 $\mathbf{x} \in \{0,1\}^\ell$,*

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

- **Succinctness:** *The functional commitment scheme is succinct if there exists a universal polynomial $\mathsf{poly}(\cdot,\cdot,\cdot)$ such that for all $\lambda \in \mathbb{N}$, $|\sigma| = \mathsf{poly}(\lambda, d, \log \ell)$ and $|\pi_f| = \mathsf{poly}(\lambda, d, \log \ell)$ in the correctness definition.*[11]
- **Binding:** *We say Π_{FC} satisfies statistical (resp., computational) binding if for all adversaries $\mathcal{A}$ (resp., efficient adversaries $\mathcal{A}$),*

$$\Pr\left[\mathsf{Verify}(\mathsf{crs}, \sigma, f, 0, \pi_0) = 1 = \mathsf{Verify}(\mathsf{crs}, \sigma, f, 1, \pi_1)\right] = \mathsf{negl}(\lambda),$$

  where $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\ell, 1^d)$ and $(\sigma, f, \pi_0, \pi_1) \leftarrow \mathcal{A}(1^\lambda, 1^\ell, 1^d, \mathsf{crs})$.
- **Private openings:** *For an adversary $\mathcal{A}$ and a simulator $\mathcal{S} = (\mathcal{S}_0, \mathcal{S}_1)$, we start by defining two distributions $\mathsf{Real}_{\mathcal{A}}(1^\lambda, 1^\ell, 1^d)$ and $\mathsf{Ideal}_{\mathcal{A},\mathcal{S}}(1^\lambda, 1^\ell, 1^d)$:*

$\mathsf{Real}_{\mathcal{A}}(1^\lambda, 1^\ell, 1^d)$:	$\mathsf{Ideal}_{\mathcal{A},\mathcal{S}}(1^\lambda, 1^\ell, 1^d)$:
1. *Give* $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\ell, 1^d)$ *to* $\mathcal{A}$. 2. *Algorithm* $\mathcal{A}$ *outputs an input* $\mathbf{x} \in \{0,1\}^\ell$. 3. *Compute* $(\sigma, \mathsf{st}) \leftarrow \mathsf{Commit}(\mathsf{crs}, \mathbf{x})$ *and give* σ *to* $\mathcal{A}$. 4. *Algorithm* $\mathcal{A}$ *outputs a function* $f \in \mathcal{F}_\lambda$. 5. *Give* $\pi_f \leftarrow \mathsf{Eval}(\mathsf{st}, f)$ *to* $\mathcal{A}$. 6. *Algorithm* $\mathcal{A}$ *outputs a bit* $b \in \{0,1\}$ *which is the output of the experiment.*	1. *Sample* $(\mathsf{crs}, \sigma, \mathsf{st}) \leftarrow \mathcal{S}_0(1^\lambda, 1^\ell, 1^d)$ *and give* crs *to* $\mathcal{A}$. 2. *Algorithm* $\mathcal{A}$ *outputs an input* $\mathbf{x} \in \{0,1\}^\ell$. 3. *Give* σ *to* $\mathcal{A}$. 4. *Algorithm* $\mathcal{A}$ *outputs a function* $f \in \mathcal{F}_\lambda$. 5. *Compute* $\pi_f \leftarrow \mathcal{S}_1(\mathsf{st}, f, f(\mathbf{x}))$ *and give* π_f *to* $\mathcal{A}$. 6. *Algorithm* $\mathcal{A}$ *outputs a bit* $b \in \{0,1\}$ *which is the output of the experiment.*

  We say that Π_{FC} has statistical (resp., computational) private openings if for all adversaries $\mathcal{A}$ (resp., efficient adversaries $\mathcal{A}$), there exists an efficient simulator $\mathcal{S} = (\mathcal{S}_0, \mathcal{S}_1)$ such that $\mathsf{Real}_{\mathcal{A}}(1^\lambda, 1^\ell, 1^d)$ and $\mathsf{Ideal}_{\mathcal{A},\mathcal{S}}(1^\lambda, 1^\ell, 1^d)$ are statistically (resp., computationally) indistinguishable.

Construction 4.2 (Succinct Functional Commitment). Let λ be a security parameter and $n = n(\lambda)$, $m = m(\lambda)$, and $q = q(\lambda)$ be lattice parameters where q is prime. Let $m' = n(\lceil \log q \rceil + 1)$ and $B = B(\lambda)$ be a bound. Let

[11] We could consider an even stronger notion of succinctness where the size of the commitment and the opening depends polylogarithmically on the *size* of the Boolean circuits computing $\mathcal{F}$. However, like existing (non-succinct) lattice-based homomorphic commitments and signatures [GVW15], the size of the commitment and openings in our construction scale with the *depth* of the computation.

$s_0 = s_0(\lambda)$, $s_1 = s_1(\lambda)$ be Gaussian width parameters. Let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of Boolean valued functions $f\colon \{0,1\}^\ell \to \{0,1\}$ where each function $f\colon \{0,1\}^\ell \to \{0,1\}$ is a function on inputs of length $\ell = \ell(\lambda)$ and which can be computed by a Boolean circuit of depth at most $d = d(\lambda)$. We construct a functional commitment $\Pi_{\mathsf{VC}} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Verify})$ for $\mathcal{F}$ as follows:

- $\mathsf{Setup}(1^\lambda, 1^\ell, 1^d)$: On input the security parameter λ, the input length ℓ, and the bound d on the circuit depth, the setup algorithm samples $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$ and for each $i \in [\ell]$, samples an *invertible* matrix $\mathbf{W}_i \xleftarrow{R} \mathbb{Z}_q^{n\times n}$. Next, it computes $\tilde{\mathbf{R}}_i \leftarrow \mathbf{R}\mathbf{G}^{-1}(\mathbf{W}_i^{-1}\mathbf{G}) \in \mathbb{Z}_q^{m\times m'}$ for each $i \in [\ell]$ and constructs matrices $\mathbf{B}_\ell$ and $\mathbf{R}$ as follows:

$$\mathbf{B}_\ell = \left[\begin{array}{ccc|c} \mathbf{W}_1\mathbf{A} & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{W}_\ell\mathbf{A} & -\mathbf{G} \end{array}\right] \quad \text{and} \quad \tilde{\mathbf{R}} = \left[\frac{\mathrm{diag}(\tilde{\mathbf{R}}_1, \ldots, \tilde{\mathbf{R}}_\ell)}{\mathbf{0}^{m'\times \ell m'}}\right]. \tag{4.1}$$

  Finally, the setup algorithm samples $\mathbf{T} \leftarrow \mathsf{SamplePre}(\mathbf{B}_\ell, \tilde{\mathbf{R}}, \mathbf{G}_{n\ell}, s_0)$ and outputs the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}_1, \ldots, \mathbf{W}_\ell, \mathbf{T})$.
- $\mathsf{Commit}(\mathsf{crs}, \mathbf{x})$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}_1, \ldots, \mathbf{W}_\ell, \mathbf{T})$ and a vector $\mathbf{x} \in \{0,1\}^\ell$, the commit algorithm constructs $\mathbf{B}_\ell$ from $\mathbf{A}, \mathbf{W}_1, \ldots, \mathbf{W}_\ell$ according to Eq. (4.1). It then constructs a target matrix

$$\mathbf{U}_\mathbf{x} = \begin{bmatrix} -x_1\mathbf{W}_1\mathbf{G} \\ \vdots \\ -x_\ell\mathbf{W}_\ell\mathbf{G} \end{bmatrix} \in \mathbb{Z}_q^{n\ell\times m'}. \tag{4.2}$$

  It then uses $\mathbf{T}$ to sample a preimage

$$\begin{bmatrix} \mathbf{V}_1 \\ \vdots \\ \mathbf{V}_\ell \\ \hat{\mathbf{C}} \end{bmatrix} \leftarrow \mathsf{SamplePre}(\mathbf{B}_\ell, \mathbf{T}, \mathbf{U}_\mathbf{x}, s_1). \tag{4.3}$$

  It outputs the commitment $\sigma = \mathbf{C} = \mathbf{G}\hat{\mathbf{C}} \in \mathbb{Z}_q^{n\times m'}$ and the state $\mathsf{st} = (\mathbf{x}, \mathbf{C}, \mathbf{V}_1, \ldots, \mathbf{V}_\ell)$.
- $\mathsf{Eval}(\mathsf{crs}, \mathsf{st}, f)$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}_1, \ldots, \mathbf{W}_\ell, \mathbf{T})$, a commitment state $\mathsf{st} = (\mathbf{x}, \mathbf{C}, \mathbf{V}_1, \ldots, \mathbf{V}_\ell)$, and a function $f\colon \{0,1\}^\ell \to \{0,1\}$, the evaluation algorithm sets $\tilde{\mathbf{C}} \leftarrow [\mathbf{W}_1^{-1}\mathbf{C} \mid \cdots \mid \mathbf{W}_\ell^{-1}\mathbf{C}]$, computes $\mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}} \leftarrow \mathsf{EvalFX}(\tilde{\mathbf{C}}, f, \mathbf{x})$, and outputs the opening $\pi_f = \mathbf{V}_f \leftarrow [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_\ell] \cdot \mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}}$.
- $\mathsf{Verify}(\mathsf{crs}, \sigma, f, y, \pi)$: On input the common reference string $\mathsf{crs} = (\mathbf{A}, \mathbf{W}_1, \ldots, \mathbf{W}_\ell, \mathbf{T})$, a commitment $\sigma = \mathbf{C} \in \mathbb{Z}_q^{n\times m'}$, a function $f\colon \{0,1\}^\ell \to \{0,1\}$, a value $y \in \{0,1\}$, and an opening $\pi = \mathbf{V}_f \in \mathbb{Z}_q^{m\times m'}$, the verification algorithm sets $\tilde{\mathbf{C}} \leftarrow [\mathbf{W}_1^{-1}\mathbf{C} \mid \cdots \mid \mathbf{W}_\ell^{-1}\mathbf{C}]$, computes $\tilde{\mathbf{C}}_f \leftarrow \mathsf{EvalF}(\tilde{\mathbf{C}}, f)$ and outputs 1 if

$$\|\mathbf{V}_f\| \le B \quad \text{and} \quad \mathbf{A}\mathbf{V}_f = \tilde{\mathbf{C}}_f - y\mathbf{G}. \tag{4.4}$$

Due to space limitations, we defer the analysis of Construction 4.2 to the full version of this paper [WW22]. We summarize the parameter instantiation below:

Parameter instantiation. Let λ be a security parameter and $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of functions $f\colon \{0,1\}^\ell \to \{0,1\}$ on inputs of length $\ell = \ell(\lambda)$ and which can be computed by Boolean circuits of depth at most $d = d(\lambda)$.

- Let $\varepsilon > 0$ be a constant. We set the lattice dimension $n = d^{1/\varepsilon} \cdot \mathsf{poly}(\lambda)$ and $m = O(n \log q)$.
- We set $s_0 = O(\ell m^2 \log(n\ell))$ and

$$s_1 = O(\ell^{3/2} m^{3/2} \log(n\ell) \cdot s_0) = O(\ell^{5/2} m^{7/2} \log^2(n\ell)) = O(\ell^{5/2} n^{7/2} \log^2(n\ell) \log^{7/2} q).$$

- We set the bound $B = s_1 \cdot \sqrt{\ell m + m'} \cdot (n \log q)^{O(d)} = \ell^3 \log^2 \ell \cdot (n \log q)^{O(d)}$.
- We set the modulus q so that the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption holds with parameters $(n, m, q, \beta, s_0, \ell)$, where

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

  where we write $\tilde{O}(\cdot)$ to suppress polylogarithmic factors in λ, d, ℓ. Note that this also requires that $\mathsf{SIS}_{n,m,q,\beta}$ hold. For instance, we set $q = \beta \cdot \mathsf{poly}(n)$. Then, $\log q = \mathsf{poly}(d, \log\lambda, \log\ell)$. Note that the underlying SIS assumption now relies on a *sub-exponential* noise bound.

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

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

$$|\sigma| = nm' \log q = O(n^2 \log^2 q) = \mathsf{poly}(\lambda, d, \log\ell).$$

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

$$|\pi| = mm' \log q = \mathsf{poly}(\lambda, d, \log\ell).$$

  In Remark 4.4, we describe a simple approach to compress the opening to a *vector* instead of a matrix.
- **CRS size:** The CRS consists of $(\mathbf{A}, \mathbf{W}_1, \ldots, \mathbf{W}_\ell, \mathbf{T})$, where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{W}_i \in \mathbb{Z}_q^{n\times n}$, and $\mathbf{T} \in \mathbb{Z}_q^{(\ell m+m')\times \ell m'}$. Thus, the total size of the CRS is

$$|\mathsf{crs}| = nm \log q + \ell n^2 \log q + (\ell m + m')(\ell m') \log q = \ell^2 \cdot \mathsf{poly}(\lambda, d, \log\ell).$$

Thus, Construction 4.2 is a succinct functional commitment scheme for bounded-depth circuits. We summarize the instantiation in the following corollary:

Corollary 4.3 (Succinct Vector Commitment from $\mathsf{BASIS}_{\mathsf{struct}}$). *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 $\mathsf{BASIS}_{\mathsf{struct}}$ assumption with a norm bound $\beta = 2^{\tilde{O}(d)}$ and modulus $q = 2^{\tilde{O}(d)}$, there exists a computationally-binding succinct functional commitment scheme for $\mathcal{F}$. Both the size of the commitment and the opening are $\mathsf{poly}(\lambda, d, \log\ell)$, and the CRS has size $\ell^2 \cdot \mathsf{poly}(\lambda, d, \log\ell)$. Here, $\tilde{O}(\cdot)$ suppresses polylogarithmic factors in λ, d, and ℓ.*

Remark 4.4 (Reducing the Opening Size). An opening π_f to a function f in Construction 4.2 consists of a matrix $\pi_f = \mathbf{V}_f \in \mathbb{Z}_q^{m \times m'}$ where $m, m' = O(n \log q)$. It is easy to adapt Construction 4.2 to obtain slightly shorter openings (i.e., $\pi_f = \mathbf{v}_f \in \mathbb{Z}_q^m$). The idea is simple: we publish a random target vector $\mathbf{u} \xleftarrow{R} \mathbb{Z}_q^n$ in the CRS and define the new opening to be $\mathbf{v}_f \leftarrow \mathbf{V}_f \mathbf{G}^{-1}(\mathbf{u})$, where $\mathbf{V}_f$ is the original opening from Construction 4.2. The updated verification relation then checks that $\|\mathbf{v}_f\|$ is small and that $\mathbf{A}\mathbf{v}_f = \tilde{\mathbf{C}}\mathbf{G}^{-1}(\mathbf{u}) - y \cdot \mathbf{u}$. We also use this approach to aggregate openings in the full version of this paper [WW22]. However, giving out the "matrix" opening is convenient when specializing our construction to obtain polynomial commitments.

Remark 4.5 (Comparison with [ACL+22]). The authors of [ACL+22] showed how to construct a functional commitment for constant-degree polynomials where the size of the CRS scales exponentially with the *degree* of the polynomial. Our functional commitment scheme (Construction 4.2) supports arbitrary Boolean circuits of bounded depth, and the size of our CRS scales polynomially with the depth of the circuit family. Moreover, security of our construction can be reduced to a function-independent assumption (the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption) whereas the scheme in [ACL+22] relied on a function-dependent assumption. We compare the two types of assumptions in more detail in Section 6.

An advantage of the [ACL+22] construction is that if supports *fast* verification with preprocessing. Namely, in their scheme, the verifier can precompute a verification key for a function f, and subsequently, verify openings with respect to f in time that is *polylogarithmic* in the running time of f. In contrast, with our scheme, the verifier has to first homomorphically compute f on the commitment in order to verify. In the full version of this paper [WW22], we show that for the special case of linear functions, we can adapt Construction 4.2 to support fast verification in the preprocessing model.

Remark 4.6 (A Candidate SNARG with Expensive Verification). The authors of [ACL+22] show how to boost their functional commitment for quadratic polynomials to a preprocessing SNARG for NP as follows:

1. First, [ACL+22] applying "sparsification" to their functional commitment scheme. Over the integers, one analog of sparsification is to require the adversary to output a short $\tilde{\mathbf{V}}$ such that $\tilde{\mathbf{A}}\tilde{\mathbf{V}} = \mathbf{D}\mathbf{C}$, where $\mathbf{D} \in \mathbb{Z}_q^{2m \times n}$, where $\tilde{\mathbf{A}} \xleftarrow{R} \mathbb{Z}_q^{2m \times 2m \log q}$ is a sparsification matrix. The CRS includes short preimages of $\tilde{\mathbf{A}}$ to enable sampling of $\tilde{\mathbf{V}}$.
2. Next, [ACL+22] introduce a knowledge assumption that says that the only way an adversary can produce $\mathbf{C}$ and $\tilde{\mathbf{V}}$ is by computing a short linear combination of the preimages in the CRS (where the coefficients correspond to the committed vector $\mathbf{x}$).
3. To support openings to multiple quadratic polynomials with a succinct opening (i.e., sublinear in the number of openings), [ACL+22] introduces a novel SIS-based technique.

Taken together, [ACL+22] show how to obtain an "extractable" commitment scheme, a notion that is equivalent to a succinct argument of knowledge for

satisfiability of quadratic systems. This yields a publicly-verifiable preprocessing SNARG for NP since satisfiability of degree-2 polynomials is NP-complete. Specifically, a proof for a statement x consists of a commitment σ to a satisfying witness w and an opening π of σ to a satisfying assignment to the quadratic constraint system representing the NP relation. By relying on preprocessing (Remark 4.5), the [ACL+22] SNARG has short proofs and fast verification.

We can apply an analogous approach to our functional commitment scheme (Construction 4.2) to obtain a candidate SNARG for NP; our SNARG would have short proofs but an *expensive* verification step (since our functional commitment does not support fast verification in the preprocessing model). We also note that even without sparsification, our construction is still a candidate SNARG: we do not know how to prove soundness of our construction, but at the same time, are not aware of any attacks either. An attack on our candidate SNARG (without sparsification) would be interesting, and we invite cryptanalysis of our candidate.

4.1 Opening to Linear Functions and Applications to Polynomial Commitments

In the full version of this paper [WW22], we describe a variant of Construction 4.2 for the setting of *linear* functions that supports fast verification in the preprocessing model (see Remark 6.1). Moreover, the full version of this paper [WW22] naturally supports linear functions over $\mathbb{Z}_q^\ell$ (as opposed to $\{0,1\}^\ell$) and generalizes to yield a *polynomial commitment* [KZG10]. Unlike [ACL+22], we do *not* require the values in the committed vector or the output of the linear function to be short. Supporting *large* values is necessary for obtaining a succinct polynomial commitment.

4.2 Supporting Private Openings

In the full version of this paper [WW22], we show how to extend Construction 4.2 to additionally support *private* openings. Recall from Definition 4.1 that a functional commitment supports private openings if the commitment σ to an input $\mathbf{x}$ together with an opening π_f with respect to a function f leaks no additional information about $\mathbf{x}$ other than the value $f(\mathbf{x})$. In the context of homomorphic signatures [GVW15], this property is called *context hiding*. We show that the same approach used to achieve context hiding in the setting of homomorphic signatures applies to our setting and yields a succinct functional commitment that supports private openings. However, the transformation does not preserve the binding property on the functional commitments scheme. Nonetheless, we can still show that the scheme satisfies a *weaker* notion of binding called *target binding*, which says that any *honestly-generated* commitment on $\mathbf{x}$ can only be opened to $f(\mathbf{x})$ for any function f. We defer the details to the full version of this paper [WW22].

5 Aggregatable Vector Commitments

In the full version of this paper [WW22], we show how to use the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption to obtain a variant of our SIS-based vector commitment (Construction 3.5) that supports aggregation. Recall that in an aggregatable commitment, one can take a collection of openings $\{(i, \pi_i)\}_{i \in S}$ for a set of $S \subseteq [\ell]$ of indices and aggregate them into a *single* opening π (for the set S) whose size scales sublinearly with the size of S. Aggregatable commitments imply subvector commitments [LM19] which are vector commitments that allow for succinct openings to a set of indices (but do *not* necessarily support aggregating openings).

6 New SIS Assumptions: Relations and Discussion

In this section, we compare our approach of publishing a full trapdoor in the common reference string with the approach of Albrecht et al. [ACL+22] of publishing short preimages in the CRS. While Albrecht et al. formulate their assumption over polynomial rings, their ideas apply equally well in the integer setting. We describe everything over the integers to enable a more direct comparison. We start by recalling the general paradigm for constructing vector commitments from Section 1.2 common to our approach and their approach:

- The CRS consists of ℓ matrices $\mathbf{A}_i \in \mathbb{Z}_q^{n \times m}$ and a set of target vectors $\mathbf{t}_i \in \mathbb{Z}_q^n$ for $i \in [\ell]$. The CRS also contains some auxiliary information aux_ℓ that is used to construct commitments and openings.
- An opening $\mathbf{v}_i$ to value x_i at index i with respect to a commitment $\mathbf{c} \in \mathbb{Z}_q^n$ is a short vector $\mathbf{v}_i$ that satisfies $\mathbf{c} = \mathbf{A}_i \mathbf{v}_i - x_i \mathbf{t}_i$.

We now compare the two types of auxiliary information aux_ℓ in our approach (based on the BASIS assumption) and the Albrecht et al. approach (based on variants of the k-ISIS assumption):

(I) **Our approach:** In our approach based on the BASIS assumption, $\mathsf{aux}_\ell = \mathbf{T}$ is a trapdoor $\mathbf{T} \leftarrow \mathbf{B}_\ell^{-1}(\mathbf{G}_{n\ell})$ for the matrix $\mathbf{B}_\ell = [\mathrm{diag}(\mathbf{A}_1, \ldots, \mathbf{A}_\ell) \mid -\mathbf{1}^\ell \otimes \mathbf{G}]$. As shown in Sections 3 and 4, the trapdoor $\mathbf{T}$ suffices to *jointly* sample commitments $\mathbf{c}$ and openings $\mathbf{v}_1, \ldots, \mathbf{v}_\ell$ that satisfy the verification relation.

(II) **The Albrecht et al. approach:** In the Albrecht et al. [ACL+22] approach, the auxiliary information $\mathsf{aux}_\ell = \{\mathbf{z}_{j,i}\}_{i \neq j}$ consists of a collection of short vectors $\mathbf{z}_{j,i} \leftarrow \mathbf{A}_i^{-1}(\mathbf{t}_j)$. The commitment to a vector $\mathbf{x} \in \{0,1\}^\ell$ is the vector $\mathbf{c} = \sum_{i \in [\ell]} -x_i \mathbf{t}_i$ and the openings are $\mathbf{v}_i = \sum_{j \neq i} -x_j \mathbf{z}_{j,i}$.

We now compare the relative power of these two types of auxiliary information. We refer to the above auxiliary data as "Type I" auxiliary data and "Type II" auxiliary data, respectively.

- When the target vectors $\mathbf{t}_1, \ldots, \mathbf{t}_\ell$ are uniform, we can simulate a CRS with Type II auxiliary data from a CRS with Type I auxiliary data. Namely, given

matrices $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$ and the trapdoor $\mathsf{aux}_\ell = \mathbf{T}$ for $\mathbf{B}$, we can sample

$$\begin{bmatrix} \mathbf{z}_{j,1} \\ \vdots \\ \mathbf{z}_{j,\ell} \\ \hat{\mathbf{c}}_j \end{bmatrix} \leftarrow \mathbf{B}_\ell^{-1}(\mathbf{0}),$$

for each $j \in [\ell]$. By construction of $\mathbf{B}_\ell$, for all $j \in [\ell]$, $\mathbf{z}_{j,i} \in \mathbb{Z}_q^m$ is a short vector satisfying $\mathbf{A}_i \mathbf{z}_{j,i} = \mathbf{G}\hat{\mathbf{c}}_j$. The marginal distribution of each $\hat{\mathbf{c}}_j$ is a discrete Gaussian, and $\mathbf{G}\hat{\mathbf{c}}_j$ is uniform over $\mathbb{Z}_q^n$. Thus, we obtain a Type II CRS with matrices $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$, target vectors $\mathbf{t}_1 = \mathbf{G}\hat{\mathbf{c}}_1, \ldots, \mathbf{t}_\ell = \mathbf{G}\hat{\mathbf{c}}_\ell$, and auxiliary data $\mathsf{aux}_\ell = \{\mathbf{z}_{j,i}\}_{i \neq j}$.

– Next, we show that we can also use Type II auxiliary data to obtain a trapdoor for *sub-matrices* of $\mathbf{B}$. We illustrate this with a concrete example. Suppose we want to obtain a trapdoor for the matrix $\mathbf{B}_k$ (where $k < \ell/m$):

$$\mathbf{B}_k = \left[\begin{array}{ccc|c} \mathbf{A}_1 & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{A}_k & -\mathbf{G} \end{array}\right] \in \mathbb{Z}_q^{kn \times (km+m')},$$

where $m' = n(\lfloor \log q \rfloor + 1)$. For $j \neq i$, let $\mathbf{z}_{j,i} \in \mathbb{Z}_q^m$ be short vectors where $\mathbf{A}_i \mathbf{z}_{j,i} = \mathbf{t}_j$ be the vectors in the Type II auxiliary data. For any $j > k$, consider the vector

$$\mathbf{v}_j = \begin{bmatrix} \mathbf{z}_{j,1} \\ \vdots \\ \mathbf{z}_{j,k} \\ \mathbf{G}^{-1}(\mathbf{t}_j) \end{bmatrix} \in \mathbb{Z}_q^{km+m'},$$

Observe that $\mathbf{v}_j$ is short, and moreover $\mathbf{B}_k \mathbf{v}_j = \mathbf{0}$. If $\ell - k > km + m'$, then we can collect $km + m'$ such vectors $\mathbf{v}_j$. Heuristically, if these vectors are linearly independent (over the integers), then this yields a Ajtai-basis for $\mathbf{B}_k$. Thus Type II auxiliary data implies Type I auxiliary data for a slightly smaller dimension $k \approx \ell/m$.

While asking for security given a full trapdoor for the related matrix $\mathbf{B}_\ell$ might seem like a stronger assumption than giving our many short preimages under $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$, the above analysis shows that these these two types of auxiliary data have comparable power. Hardness of SIS/ISIS with respect to one type of auxiliary data is comparable to hardness with respect to the other (up to an $O(n \log q)$ loss in the vector dimension ℓ). In fact, the above analysis shows that Type II auxiliary data (with essentially arbitrary target vectors $\mathbf{u}_i$) is already sufficient to construct a trapdoor that yields a Type I auxiliary data for a smaller input dimension. However, the converse is not true, as the trapdoor for $\mathbf{B}_\ell$ seem to only allow sampling preimages of $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$ with respect to random target vectors $\mathbf{t}_1, \ldots, \mathbf{t}_\ell$. This distinction is important, and as we discuss in Remark 6.1, Type I auxiliary data seem essential to realizing the functional commitment

scheme from Section 4 (as well as its aggregatable analog in the full version of this paper [WW22]). Other advantages to using a Type I auxiliary data include supporting private openings (Construction 3.5) and commitments to large inputs.

Remark 6.1 (Structured Targets and Functional Commitments). The main verification relation of our functional commitment scheme (Construction 4.2) is $\mathbf{C} = \mathbf{A}_i\mathbf{V}_i + x_i\mathbf{G}$ where $\mathbf{A}_i = \mathbf{W}_i\mathbf{A}$. If we consider a Type II auxiliary data for this verification relation, the auxiliary data would contain $\mathbf{A}_i^{-1}(\mathbf{G})$, or equivalently, $\mathbf{A}^{-1}(\mathbf{W}_i^{-1}\mathbf{G})$. However, $\mathbf{A}^{-1}(\mathbf{W}_i^{-1}\mathbf{G})$ is a trapdoor for $\mathbf{A}$ (with tag $\mathbf{W}_i^{-1}$), which trivially breaks security. In contrast, using Type I auxiliary data does not appear to yield a trapdoor for $\mathbf{A}$, and plausibly yields a succinct functional commitment scheme.

6.1 Another View of the $\mathsf{BASIS}_{\mathsf{struct}}$ Assumption

To facilitate cryptanalysis of our new assumption, we provide an equivalent formulation of the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption (Assumption 3.3) underlying our functional, polynomial, and aggregatable commitments. Consider a variant of the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption where $\mathbf{T}$ is an Ajtai trapdoor for $\mathbf{B}$ (i.e., $\mathbf{T} \leftarrow \mathbf{B}_s^{-1}(\mathbf{0}^{m\times 2m})$). Note that we can efficiently convert between gadget trapdoors and Ajtai trapdoors, up to small polynomial losses in the quality of the trapdoor. It is easy to see that we can re-express $\mathbf{B}^{-1}(\mathbf{0}^{m\times 2m})$ as $\mathbf{A}^{-1}(\mathbf{W}_i^{-1}\mathbf{R})$ for all $i \in [\ell]$, and $\mathbf{R} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n\times 2m}$. Therefore, the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption is equivalent to:

$$\begin{gathered}\textit{SIS is hard with respect to } \mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n\times m} \textit{ given } \mathbf{A}^{-1}(\mathbf{W}_i^{-1}\mathbf{R}) \\ \textit{for all } i \in [\ell], \textit{ where } \mathbf{W}_i \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n\times n} \textit{ and } \mathbf{R} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n\times 2m}.\end{gathered}$$

Remark 6.2 (Parameter Choices for the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption). While hardness of the $\mathsf{BASIS}_{\mathsf{rand}}$ assumption can be based on the hardness of the standard SIS assumption, we do not know of an analogous reduction for the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption. When setting parameters for the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption, we use Theorem 3.4 as a guide and consider instantiations where $n \geq \lambda$, $m \geq O(n \log q)$ and $s \geq O(\ell m \log n) = \mathsf{poly}(\lambda, \ell)$. Note that this means the quality of the basis *decreases* with the dimension. For this parameter setting, we are not aware of any concrete attacks on the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption and conjecture that its security is comparable with the hardness of $\mathsf{SIS}_{n,m,q,\beta}$, with a noise bound bound $\beta = \mathsf{poly}(\lambda, \ell)$ that scales with the dimension of the vector (as in Theorem 3.4). In particular, the hardness of the SIS instance decreases with the dimension ℓ (similar to the case with q-type assumptions over groups [Che06]).

Acknowledgments

We thank Jonathan Bootle for helpful conversations about lattice-based SNARKs. D. J. Wu is supported by NSF CNS-2151131, CNS-2140975, a Microsoft Research Faculty Fellowship, and a Google Research Scholar award.

References

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

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

ACL+22. Martin R. Albrecht, Valerio Cini, Russell W. F. Lai, Giulio Malavolta, and Sri AravindaKrishnan Thyagarajan. Lattice-based SNARKs: Publicly verifiable, preprocessing, and recursively composable. In *CRYPTO*, 2022.

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

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

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

BBF19. Dan Boneh, Benedikt Bünz, and Ben Fisch. Batching techniques for accumulators with applications to iops and stateless blockchains. In *CRYPTO*, 2019.

BC12. Nir Bitansky and Alessandro Chiesa. Succinct arguments from multi-prover interactive proofs and their efficiency benefits. In *CRYPTO*, 2012.

BCFL22. David Balbás, Dario Catalano, Dario Fiore, and Russell W. F. Lai. Functional commitments for circuits from falsifiable assumptions. *IACR Cryptol. ePrint Arch.*, 2022.

BCS16. Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In *TCC*, 2016.

BDFG21. Dan Boneh, Justin Drake, Ben Fisch, and Ariel Gabizon. Halo infinite: Proof-carrying data from additive polynomial commitments. In *CRYPTO*, 2021.

BDN18. Dan Boneh, Manu Drijvers, and Gregory Neven. Compact multi-signatures for smaller blockchains. In *ASIACRYPT*, 2018.

BFS20. Benedikt Bünz, Ben Fisch, and Alan Szepieniec. Transparent snarks from DARK compilers. In *EUROCRYPT*, 2020.

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

BGV11. Siavosh Benabbas, Rosario Gennaro, and Yevgeniy Vahlis. Verifiable delegation of computation over large datasets. In *CRYPTO*, 2011.

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

CCH+19. Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-shamir: from practice to theory. In *STOC*, pages 1082–1090, 2019.

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

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

CFM08. Dario Catalano, Dario Fiore, and Mariagrazia Messina. Zero-knowledge sets with short proofs. In *EUROCRYPT*, 2008.

Che06. Jung Hee Cheon. Security analysis of the strong diffie-hellman problem. In *EUROCRYPT*, 2006.

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

CHM+20. Alessandro Chiesa, Yuncong Hu, Mary Maller, Pratyush Mishra, Noah Vesely, and Nicholas P. Ward. Marlin: Preprocessing zksnarks with universal and updatable SRS. In *EUROCRYPT*, 2020.

CJJ21. Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. SNARGs for $\mathcal{P}$ from LWE. In *FOCS*, 2021.

COS20. Alessandro Chiesa, Dev Ojha, and Nicholas Spooner. Fractal: Post-quantum and transparent recursive proofs from holography. In *EUROCRYPT*, 2020.

CPSZ18. Alexander Chepurnoy, Charalampos Papamanthou, Shravan Srinivasan, and Yupeng Zhang. Edrax: A cryptocurrency with stateless transaction validation. *IACR Cryptol. ePrint Arch.*, 2018.

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

DGKV22. Lalita Devadas, Rishab Goyal, Yael Kalai, and Vinod Vaikuntanathan. Rate-1 non-interactive arguments for batch-np and applications. *IACR Cryptol. ePrint Arch.*, 2022.

FSZ22. Nils Fleischhacker, Mark Simkin, and Zhenfei Zhang. Squirrel: Efficient synchronized multi-signatures from lattices. In *ACM CCS*, 2022.

GGH96. Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Collision-free hashing from lattice problems. *IACR Cryptol. ePrint Arch.*, page 9, 1996.

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

GRWZ20. Sergey Gorbunov, Leonid Reyzin, Hoeteck Wee, and Zhenfei Zhang. Pointproofs: Aggregating proofs for multiple vector commitments. In *ACM CCS*, 2020.

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

GVW15. Sergey Gorbunov, Vinod Vaikuntanathan, and Daniel Wichs. Leveled fully homomorphic signatures from standard lattices. In *STOC*, 2015.

GW11. Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *STOC*, 2011.

GWC19. Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. PLONK: permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. *IACR Cryptol. ePrint Arch.*, 2019.

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

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

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

LLNW16. Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In *EUROCRYPT*, 2016.

LM19. Russell W. F. Lai and Giulio Malavolta. Subvector commitments with application to succinct arguments. In *CRYPTO*, 2019.

LP20. Helger Lipmaa and Kateryna Pavlyk. Succinct functional commitment for a large class of arithmetic circuits. In *Advances in Cryptology - ASIACRYPT 2020, Part III*, 2020.

LRY16. Benoît Libert, Somindu C. Ramanna, and Moti Yung. Functional commitment schemes: From polynomial commitments to pairing-based accumulators from simple assumptions. In *ICALP*, 2016.

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

MBKM19. Mary Maller, Sean Bowe, Markulf Kohlweiss, and Sarah Meiklejohn. Sonic: Zero-knowledge snarks from linear-size universal and updatable structured reference strings. In *ACM CCS*, 2019.

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

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

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

Nao03. Moni Naor. On cryptographic assumptions and challenges. In *CRYPTO*, 2003.

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

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

PS19. Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In *CRYPTO*, pages 89–114, 2019.

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

Res22. Protocol Labs Research. Vector commitment research day, 2022. https://cryptonet.vercel.app/.

RMCI17. Leonid Reyzin, Dmitry Meshkov, Alexander Chepurnoy, and Sasha Ivanov. Improving authenticated dynamic dictionaries, with applications to cryptocurrencies. In *FC*, 2017.

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

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

WW22. Hoeteck Wee and David J. Wu. Succinct vector, polynomial, and functional commitments from lattices. *IACR Cryptol. ePrint Arch.*, page 1515, 2022.