Sumcheck Arguments and their Applications

Jonathan Bootle
jbt@zurich.ibm.com
IBM Research – Zurich

Alessandro Chiesa
alexch@berkeley.edu
UC Berkeley

Katerina Sotiraki
katesot@berkeley.edu
UC Berkeley

June 9, 2021

Abstract

We introduce a class of interactive protocols, which we call *sumcheck arguments*, that establishes a novel connection between the sumcheck protocol (Lund et al. JACM 1992) and folding techniques for Pedersen commitments (Bootle et al. EUROCRYPT 2016).

We define a class of sumcheck-friendly commitment schemes over modules that captures many examples of interest, and show that the sumcheck protocol applied to a polynomial associated with the commitment scheme yields a succinct argument of knowledge for openings of the commitment. Building on this, we additionally obtain succinct arguments for the NP-complete language R1CS over certain rings.

Sumcheck arguments enable us to recover as a special case numerous prior works in disparate cryptographic settings (discrete logarithms, pairings, groups of unknown order, lattices), providing one framework to understand them all. Further, we answer open questions raised in prior works, such as obtaining a lattice-based succinct argument from the SIS assumption for satisfiability problems over rings.

Keywords: sumcheck protocol; succinct arguments; scalar-product protocol

Contents

1 Introduction

Sumcheck protocols. The sumcheck protocol is an interactive proof introduced in [LFKN92] that has played a fundamental role in the theory of probabilistic proofs in complexity theory (e.g., [BFL91; BFLS91; GKR08]) and, more recently, in cryptography. The sumcheck protocol has been used widely in a line of works on *succinct arguments* [CMT12; VSBW13; Wah+17; ZGKPP17; WTSTW18; XZZPS19; Set20]. One of the main benefits of the sumcheck protocol is that, in certain settings, the prover can be implemented in a linear number of operations [Tha13] or as a streaming algorithm [CMT12]; this avoids operations such as the Fast Fourier Transform (common in other succinct arguments) that are costly in time and in memory. The sumcheck protocol also satisfies strong soundness properties that facilitate arguing the security of the Fiat–Shamir transformation in the plain model [CCHLRR18], which is notoriously hard to analyze for other interactive proofs. Moreover, variants of the sumcheck protocol have spawned lines of research: the univariate sumcheck [BCRSVW19] was used in numerous succinct arguments [BCGGRS19; ZXZS20; CHMMVW20; COS20; CFFQR20; BFHVXZ20]; and the sumcheck protocol for tensor codes [Mei13] was used to obtain probabilistic proofs with linear-size proofs [BCGRS17; RR20] and linear-time provers [BCG20; BCL20].

Folding techniques. Separately, a line of works starting with [BCCGP16] constructs succinct arguments based on *folding techniques* for Pedersen commitments in the discrete logarithm setting. Informally, to prove knowledge of a long message opening a given Pedersen commitment, the prover engages with the verifier in a reduction that halves the message length by folding the message "around" a verifier challenge. This can be repeatedly applied until the message length is small enough to send the message directly. Beyond commitment openings, [BCCGP16] give protocols for scalar-product relations, which lead to succinct arguments for NP languages such as arithmetic circuit satisfiability. These succinct arguments can be realized via a linear number of group scalar multiplications, or alternatively as streaming algorithms [BHRRS20].

Folding techniques, subsequently improved in [BBBPWM18], have been deployed in cryptocurrencies (Monero [Mon] and PIVX [Piv]) and are widely used thanks to popular open-source libraries [dalek18; Adj]. These practical applications have motivated careful analyses of concrete security [JT20], which facilitates setting security parameters in applications.

Folding techniques have been adapted to work in other cryptographic settings, such as bilinear groups [LMR19], unknown-order groups [BFS20], and lattices [BLNS20]. They have also been formulated in more abstract settings: [BMMTV19] study sufficient properties of commitment schemes that enable folding techniques; and [AC20; ACF20; ACR20; BDFG20] study folding techniques for general group homomorphisms.

Folding techniques for Pedersen (and related) commitments are arguably not fully understood, despite the numerous works and applications mentioned above. For example, they are typically used as non-interactive arguments after the Fiat–Shamir transformation is applied to the (public-coin) interactive argument. Yet the security of this non-interactive argument, even in the random oracle model, has only been proven via a superpolynomial-time extractor [BMMTV19] or in the algebraic group model [GT20]. Moreover, almost all succinct arguments are obtained via some type of probabilistic proof (and there are settings where this is inherent [RV09; CY20]) but no such probabilistic proof is evident in folding techniques.

A connection? The sumcheck protocol and folding techniques seem rather different protocols but they share several common features. Both protocols have a prover that can be realized via a linear number of operations [Tha13; BCCGP16], or alternatively as a streaming algorithm [CMT12; BHRRS20]; moreover, both protocols satisfy similar notions of strong soundness [CCHLRR18; GT20], which facilitate proving useful security properties. Are these similarities mere coincidences?

1.1 Our results

We introduce a class of interactive protocols, *sumcheck arguments*, that establishes a novel connection between the sumcheck protocol and folding techniques for Pedersen commitments. This provides a single framework to understand numerous prior works in disparate cryptographic settings (prime-order groups, bilinear groups, unknown-order groups, lattices) and also enables us to answer open questions raised in prior works. We elaborate on these contributions below, and summarize the underlying technical ideas in Section 2.

(1) Sumcheck arguments. Recall that the sumcheck protocol is an interactive proof for statements of the form $\sum_{\underline{\omega} \in H^\ell} p(\underline{\omega}) = \tau$ for a given summation domain H, ℓ-variate polynomial p, and claimed sum τ. While typically stated for polynomials over finite fields, the sumcheck protocol works for polynomials over any module M over a ring R (given certain mild conditions).[1] Let $\Sigma[R, M, H, \ell, \tau, \mathcal{C}, p]$ denote the sumcheck protocol for the statement $\sum_{\underline{\omega} \in H^\ell} p(\underline{\omega}) = \tau$ when $H \subseteq R$, $\tau \in M$, and $p \in M[X_1, \ldots, X_\ell]$, and the verifier uses the challenge set $\mathcal{C} \subseteq R$ to sample each round's challenge. (We explain later on in Section 2.1 why the sumncheck protocol over modules involves a given challenge set for the verifier.)

A *sumcheck argument* is, informally, a sumcheck protocol used to succinctly prove knowledge of openings for certain commitments (you run the sumcheck reduction followed by a cryptographic analogue of the consistency check). We say that a commitment scheme CM is *sumcheck-friendly* if the statement "I know m of length n such that $\mathsf{CM.Commit}(\mathsf{ck}, \mathsf{m}) = \mathsf{cm}$" can be rewritten as the statement "I know m of length n such that $\sum_{\underline{\omega} \in \{-1,1\}^{\log n}} f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega})) = \mathsf{cm}$" where the message polynomial $p_{\mathsf{m}}(\underline{X})$ is over an R-module $\mathbb{M}$, the key polynomial $p_{\mathsf{ck}}(\underline{X})$ is over an R-module $\mathbb{K}$, and the combiner function f_{CM} maps $\mathbb{M} \times \mathbb{K}$ to an R-module $\mathbb{C}$ (and is such that $f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{X}), p_{\mathsf{ck}}(\underline{X}))$ is a polynomial over $\mathbb{C}$). We observe that commitment schemes of interest are sumcheck-friendly, including various forms of Pedersen commitments (we elaborate on this later). Our main result is to construct a knowledge extractor for the sumcheck protocol applied to such statements, provided CM is *invertible* (a certain property that we discuss later on).

Theorem 1 (informal)**.** *Let* CM *be a sumcheck-friendly commitment scheme that is invertible. Let* cm *be a commitment to a message* m *using a commitment key* ck*. Then (a straightforward extension of)*

$$\Sigma\big[R, M = \mathbb{C}, H = \{-1, 1\}, \ell = \log n, \tau = \mathsf{cm}, \mathcal{C}, p = f_{\mathsf{CM}}(p_{\mathsf{m}}, p_{\mathsf{ck}})\big]$$

is an interactive argument of knowledge for an opening to cm *with respect to* ck *with knowledge error* $O(\frac{\log n}{|\mathcal{C}|})$*, where the polynomial in the numerator depends on* CM*. The round complexity is* $O(\log n)$ *and the communication complexity is* $O(\log n)$ *elements in* $\mathbb{C}$*. Moreover, if* f_{CM} *is a bilinear function, then the prover and verifier complexity is dominated by* $O(n)$ *operations in* $\mathbb{C}$*.*

The above informal statement omits many technical details, such as commitment randomness and relaxed notions of commitment opening necessary to express settings over lattices. Moreover, the informal statement fixes certain choices (such as choosing the summation domain $H = \{-1, 1\}$ and $\ell = \log n$ variables).

As we elaborate in Appendix A, well-known folding techniques from prior works can be viewed, perhaps surprisingly, as special cases of a sumcheck argument. We remark that while the usual security notion of the sumcheck protocol is an unconditional soundness guarantee, the security notion that we establish for a sumcheck argument is a knowledge guarantee, proved from CM's invertibility. In turn invertibility may hold unconditionally or under certain hardness assumptions (we give examples of this in Section 2.3.2).[2]

(2) Succinct arguments for R1CS over rings. Building on sumcheck arguments, we obtain zero-knowledge succinct arguments for satisfiability problems defined over *rings*. This is in contrast to most prior succinct

[1] A module is a mathematical structure the extends a vector space by allowing scalars to be from a ring rather than a field.

[2] Thus sumcheck arguments are distinct from direct algebraic generalizations of the sumcheck protocol to rings [CCKP19].

arguments, which support satisfiability problems defined over prime-order fields (which are the "scalar fields" associated to underlying cryptographic prime-order groups). This extension is motivated by the fact that certain computations are more efficiently expressed over certain rings (e.g., approximate arithmetic [CCKP19]), and parallels prior lines of work for secret-sharing schemes and multiparty computation protocols [CFIK03; CDESX18; ACDEY19; Abs+20] for supporting computations defined over rings.

We focus on the ring variant of the NP-complete problem known as *rank-1 constraint satisfiability* (R1CS), which is a widely used generalization of arithmetic circuit satisfiability. We obtain a zero-knowledge succinct argument for R1CS over any ring $R_\bullet$ with suitable algebraic properties, assuming the hardness of the *bilinear relation assumption* over a related ring, which is a natural generalization of assumptions such as the DL assumption, the SIS assumption, and others.

Definition 1 (informal)**.** *The R1CS problem asks: given a ring $R_\bullet$, coefficient matrices $A, B, C \in R_\bullet^{n \times n}$ each containing at most $m = \Omega(n)$ non-zero entries, and an instance vector $\underline{x}$ over $R_\bullet$, is there a witness vector $\underline{w}$ over $R_\bullet$ such that $\underline{z} := (\underline{x}, \underline{w}) \in R_\bullet^n$ and $A\underline{z} \circ B\underline{z} = C\underline{z}$? (Here "$\circ$" denotes the entry-wise product of vectors over $R_\bullet$.)*

Theorem 2 (informal)**.** *Let R be a ring, M be an R-module, $\mathcal{C} \subseteq R$ a challenge space, and $I \subseteq R$ an ideal. If pairwise differences in $\mathcal{C}$ have suitable pseudoinverses in $R_\bullet := R/I$ and the bilinear relation assumption holds over M, then there is a zero-knowledge succinct argument of knowledge for the R1CS problem over $R_\bullet$. For $n \times n$ coefficient matrices with at most m non-zero entries, the argument has knowledge error $O(\frac{\log n}{|\mathcal{C}|})$, round complexity $O(\log n)$, communication complexity $O(\log n)$ elements of M and $O(1)$ elements of R, and prover and verifier complexity dominated by $O(m)$ operations in R and $O(n)$ operations in M.*

One immediate application of our result is to lattice cryptography. Prior work used folding techniques to obtain (zero-knowledge) succinct arguments of knowledge for lattice commitments [BLNS20], but left open the question of obtaining succinct arguments for NP-complete problems relevant to lattices.[3]

Our Theorem 2 directly implies a solution to this open question. This may be surprising because the knowledge extractor for a sumcheck argument over lattices finds only a relaxed opening of a (sumcheck-friendly and invertible) commitment; this relaxed extraction occurs in many other lattice-based arguments of knowledge. This notwithstanding we still derive from it a knowledge extractor for the R1CS problem.

Corollary 1 (informal)**.** *Let $R := \mathbb{Z}[X]/\langle X^d + 1 \rangle$ for d a power of 2. Let p and q be primes with q sufficiently larger than p. Assuming hardness of the SIS problem over R/qR, there is an argument of knowledge for R1CS over $R_\bullet := R/pR$ with knowledge-soundness error $O(\frac{\log n}{d})$, round complexity $O(\log n)$, communication complexity dominated by $O(\log n)$ elements of R/qR, and prover and verifier complexity dominated by $O(m)$ operations in $R_\bullet$ and $O(n)$ operations in R/qR.*

Our new lattice-based argument system shows that one can succinctly prove general relations over rings pertinent to lattice cryptography, despite the fact that most lattice-based proofs of knowledge suffer from relaxed soundness properties. This allows users to prove statements about lattice-based encryption and signature schemes directly over their native rings rather than having to convert them into statements tractable for other proof systems, which often leads to computational overheads in practical schemes [BCOS20].

Moreover, Corollary 1 contributes a new succinct argument that is plausibly post-quantum, adding to a surprisingly short list of such candidates. (Prior constructions of post-quantum succinct arguments are from hash functions [CMS19; CMSZ21] or lattice knowledge assumptions [BISW17; BISW18; GMNO18].) An

[3] This differs from using lattices to instantiate the collision-resistant hash function in Kilian's PCP-based protocol [Kil92], because this would not lead to a succinct argument for computations expressed over relevant rings.

intriguing question left open by our work is whether the *security reduction* of the construction in Corollary 1 can be carried out against an efficient quantum adversary.

Finally, returning to Theorem 2, having a single construction of a zero-knowledge succinct argument over general rings may simplify future practical applications. Our theorem enables having a single abstract implementation that can be debugged and audited once and for all, and can then be instantiated over disparate algebraic settings depending on an application's needs, by simply specifying the desired ring.

(3) On instantiations. By instantiating the sumcheck-friendly commitment CM in Theorem 1 we obtain succinct arguments of knowledge for different relations of interest, as we now explain.

As a simple example, the Pedersen commitment scheme can be formulated in an abstract setting where messages and group generators are replaced by elements of appropriate rings or modules. This *generalized Pedersen commitment scheme* satisfies the conditions in Theorem 1, either unconditionally or under the same assumptions that imply its binding properties. Our sumcheck argument for the generalized Pedersen commitment scheme thus yields succinct protocols for opening Pedersen commitments in different settings, such as prime-order groups, bilinear groups, unknown-order groups, and lattices.

We also study instantiations that capture richer functionalities.

- *Linear-function commitments:* the commitment includes a commitment to the scalar product of a public (query) message and a secret message. This draws inspiration from [AC20] which considers linear-function commitments in the prime-order group setting, bilinear group setting, and strong RSA setting.
- *Scalar-product commitments:* the commitment includes a commitment to the scalar product of two secret parts of the message. This draws inspiration from [BCCGP16; BBBPWM18; BMMTV19] which consider bilinear commitment schemes for prime-order or bilinear groups. Proving knowledge of an opening implies that the commitment was correctly computed, and therefore in this case that a scalar-product relation is satisfied. These scalar-product commitments in fact underlie our proof of Theorem 2 based on Theorem 1.

In Figure 1 we provide a comparison between succinct arguments with comparable efficiency in prior works, classified by type of relation and algebraic setting. The table demonstrates that our sumcheck arguments recover *all prior types of relations and all algebraic settings* as special cases, and additionally contribute *new combinations that were not achieved before*.

<table>
<tr><th></th><th>prime-order groups
(DL assumption)</th><th>bilinear groups
(double-pairing assumption)</th><th>unknown-order groups
(order assumption)</th><th>ideal lattices
(SIS assumption)</th></tr>
<tr><td>basic commitment</td><td></td><td></td><td></td><td>[BLNS20]</td></tr>
<tr><td>linear-function commitment
or polynomial commitment</td><td colspan="2">[ACR20; AC20]</td><td>[BFS20]</td><td>previously open</td></tr>
<tr><td>scalar-product commitment</td><td>[BCCGP16]</td><td>[LMR19]</td><td colspan="2">previously open</td></tr>
<tr><td>bilinear commitment</td><td colspan="2">[BMMTV19]</td><td colspan="2">previously open</td></tr>
<tr><td>sumcheck-friendly commitment</td><td colspan="4">sumcheck arguments from this work</td></tr>
</table>

Figure 1: Comparison of prior works that use folding techniques to achieve succinct arguments of knowledge, and also our sumcheck arguments. The rows from top to bottom indicate increasingly more general types of commitment (and so a result in a row directly implies a result in all rows above it). The columns indicate different cryptographic settings in which the commitments are constructed (along with corresponding sufficient cryptographic assumptions). Results spanning multiple columns indicate an abstraction that simultaneously captures all those settings. We see that our work captures all prior settings and types of commitments, and also achieves functionalities and settings that were left open by prior works.

1.2 New connections and new opportunities

The novel connection between folding techniques and the sumcheck protocol, captured by our sumcheck arguments, casts many aspects of prior works in a new light. Below we provide several examples.

- [BCCGP16] describes folding techniques for splitting a long vector into more than two pieces before folding, to allow trading argument size for round complexity. This corresponds to running a sumcheck argument using *polynomials of fewer variables and higher individual degree*.
- [BBBPWM18] improves the efficiency of folding techniques via a more complicated use of verifier challenges. This corresponds to running a sumcheck argument using a *different evaluation domain*, and where the sumcheck prover sends polynomials expressed in a *different monomial basis*.
- [CHJKS20] gives weighted inner-product arguments to improve concrete efficiency. This corresponds to a sumcheck argument for *weighted-sums of polynomial evaluations* (see Remark 4.2).
- [PLS19] gives a zero-knowledge version of folding techniques that achieves better concrete efficiency by using less prover randomness. This relates to *derandomizing a zero-knowledge sumcheck argument*.
- [BMMTV19; BFS20] consider subprotocols for delegating expensive verifier computation to the prover. This corresponds to *delegating polynomial evaluation*, to help the verifier outsource evaluating the commitment key polynomial. Sumcheck arguments neatly conceptualize the role of polynomials in folding protocols and simplify the task of applying delegation protocols in other settings (see Remark 4.8).
- Like [BMMTV19; ACF20; BDFG20], sumcheck arguments capture optimizations of folding techniques that compress several target commitment values into one (e.g., the optimization from [BCCGP16] to [BBBPWM18]) as sumcheck arguments applied to alternative commitment schemes.

We expect that other folding techniques such as [ACR20; Lee20] can also be viewed as sumcheck arguments.

Looking ahead, the new perspective offered by sumcheck arguments, with the sumcheck protocol at their core, makes it easier to explore new design options and optimizations for succinct arguments, especially so for those that have already been studied for the (information-theoretic) sumcheck protocol.

Existing analyses of the sumcheck protocol may also inspire analogous ones for sumcheck arguments. For example, the sumcheck protocol can be made non-interactive via the Fiat–Shamir transformation, where the verifier's messages are replaced by the outputs of a hash function. Jawale et al. [JKKZ20] show that the result is a non-interactive argument provided the hash function is *lossy correlation-intractable* (and construct such hash functions based on the LWE assumption). This seems to provide a starting point for studying the security of sumcheck arguments under the Fiat–Shamir transformation.

1.3 Related work

Folding techniques. Figure 1 summarizes the main relationship between sumcheck arguments for sumcheck-friendly commitments and prior work that uses folding techniques. Below we additionally discuss the prior works that have studied folding techniques for abstract commitment schemes and homomorphisms.

Bünz et al. [BMMTV19] present folding techniques for doubly-homomorphic commitments over prime-order groups, which are both key-homomorphic and message homomorphic. These can capture non-linear relations such as scalar-product relations under computational assumptions.

Attema, Cramer, and Fehr [ACF20] present folding techniques for pre-images of general group homomorphisms over prime-order groups. These were extended from prime-order groups to $\mathbb{Z}$-modules in [BDFG20], who also noted that a $\mathbb{Z}$-module homomorphism could be phrased as a Pedersen-like function. These techniques give *proofs* for homomorphisms and linear relations, without using computational assumptions.

Both general group homomorphisms and doubly-homomorphic commitment schemes are special cases of sumcheck-friendly commitment schemes. Our work also finds the same distinction between proofs and arguments: our sumcheck argument for "linear" commitment schemes such as the generalized Pedersen commitment scheme (and linear-function commitments) do not require computational assumptions, whereas our sumcheck argument for "quadratic" commitment schemes require computational assumptions.

Reductions from NP-complete problems. Attema and Cramer [AC20] construct zero-knowledge succinct arguments for NP-complete relations by (i) using secret-sharing techniques to interactively reduce NP statements to linear relations (under computational assumptions), and then (ii) relying on succinct arguments for linear relations. This "linearization" requires the prover to perform polynomial arithmetic on high-degree polynomials, and hence an efficient realization would likely rely on FFTs. FFTs require linear space-complexity for the prover, and prevent the prover from being implemented in logarithmic space as in the sumcheck protocol [CMT12] or other succinct arguments based on folding protocols [BHRRS20]. In contrast, we reduce NP statements to bilinear relations such as scalar-product relations, and then rely on succinct arguments for scalar products; this reduction can be performed via a linear number of cryptographic operations, and without relying on FFTs.

1.4 Concurrent work

Attema, Cramer, and Kohl [ACK21] construct zero-knowledge succinct arguments for NP based on the SIS assumption, using folding techniques for lattices. As with [AC20], their construction uses secret-sharing techniques which are likely to rely on FFTs and lead to a prover with large space complexity. Moreover, the techniques in [ACK21] are for lattices, while our techniques based on sumcheck arguments provide a general framework in which lattices are a special case. Additionally, [ACK21] give a detailed analysis of the knowledge error of their lattice-based folding techniques, which was not present in [BLNS20], and establish that the knowledge error can be reduced using parallel repetition.

Albrecht and Lai [AL21] study a variant of the folding techniques in [BLNS20], instantiated in a different choice of ring which offers exact proofs (rather than proofs with relaxed knowledge extraction) and various efficiency advantages. Like [ACK21], they also analyze the knowledge error of their folding techniques, and prove results relating relaxed extraction to ring structure. We are optimistic that their ideas can be incorporated into our sumcheck-based framework.

Ganesh, Nitulescu, and Soria-Vazquez [GNS21] model NP relations over rings and give a generic construction of designated-verifier zero-knowledge SNARKs using techniques related to prior lattice-based SNARK constructions [BISW17; BISW18; GMNO18].

Block et al. [BHRRS21] study a variant of the commitment scheme of [BFS20] in groups of unknown order that is compatible with a streaming formalism, and give space-efficient arguments for NP languages. We are optimistic that their ideas can be incorporated into our sumcheck-based framework.

2 Techniques

We summarize the main ideas behind our results. The first few subsections are dedicated to explaining sumcheck arguments (Theorem 1) in several steps of progressive generality. In Section 2.1 we describe the sumcheck protocol for polynomials over modules. Then in Section 2.2 we present a succinct zero-knowledge argument for Pedersen commitments based on the sumcheck protocol. In Section 2.3 we show how to lift this protocol to any "sumcheck-friendly" commitment, but still in the setting of prime-order groups. Finally in Section 2.4 we explain the main considerations in generalizing further to commitments over rings, and in Section 2.5 we give an example of how commitments can be formulated in this framework. After that we turn our attention to our other contributions. In Section 2.6 we discuss a generic scalar-product protocol built from sumcheck arguments, and then in Section 2.7 we explain how it enables us to obtain zero-knowledge succinct arguments for R1CS over rings (Theorem 2 and in particular Corollary 1). Finally, in Section 2.8 we discuss how we also obtain polynomial commitment schemes over rings from sumcheck arguments.

2.1 Sumcheck protocol over modules

The sumcheck protocol [LFKN92] directly extends to work with polynomials over *modules*. The prover P_{SC} and verifier V_{SC} receive a sumcheck instance $\mathbb{x}_{\mathrm{SC}} = (R, M, H, \ell, \tau, \mathcal{C})$, where R is a ring, M is a module over R, H is a subset of R, ℓ is a number of variables, $\tau \in M$ is a claimed sum, and $\mathcal{C} \subseteq R$ is a sampling set (more about this below). The prover P_{SC} additionally receives a polynomial $p \in M[X_1, \ldots, X_\ell]$ such that $\sum_{\underline{\omega} \in H^\ell} p(\underline{\omega}) = \tau$. The protocol has ℓ rounds and works as follows.

1. For $i = 1, \ldots, \ell$:
   (a) P_{SC} sends to V_{SC} the polynomial $q_i(X) := \sum_{\omega_{i+1}, \ldots, \omega_\ell \in H} p(r_1, \ldots, r_{i-1}, X, \omega_{i+1}, \ldots, \omega_\ell) \in M[X]$;
   (b) V_{SC} sends to P_{SC} a random challenge $r_i \leftarrow \mathcal{C}$.
2. V_{SC} checks that $\sum_{\omega_1 \in H} q_1(\omega_1) = \tau$ and, for $i \in \{2, \ldots, \ell\}$, that $\sum_{\omega_i \in H} q_i(\omega_i) = q_{i-1}(r_{i-1})$.
3. If the checks pass then V_{SC} sets $v := q_\ell(r_\ell) \in M$ and outputs the tuple $((r_1, \ldots, r_\ell), v)$.

The security guarantee of the sumcheck protocol, which requires $\mathcal{C}$ to be a sampling set, is given below.

Definition 2.1. *We say that $\mathcal{C} \subseteq R$ is a* **sampling set** *for the R-module M if for every distinct $c_1, c_2 \in \mathcal{C}$ the map that sends $m \in M$ to $(c_1 - c_2) \cdot m \in M$ is injective.*

Lemma 2.2. *Let $\mathbb{x}_{\mathrm{SC}} = (R, M, H, \ell, \tau, \mathcal{C})$ be a sumcheck instance and a polynomial $p \in M[X_1, \ldots, X_\ell]$ of total degree d. If $\mathcal{C}$ is a sampling set for M then the following holds.*
• Completeness. *If $\sum_{\underline{\omega} \in H^\ell} p(\underline{\omega}) = \tau$ then $\mathrm{Pr}_{\underline{r} \leftarrow \mathcal{C}^\ell}[\langle P_{\mathrm{SC}}(\mathbb{x}_{\mathrm{SC}}, p), V_{\mathrm{SC}}(\mathbb{x}_{\mathrm{SC}}; \underline{r})\rangle = (\underline{r}, p(\underline{r}))] = 1$.*
• Soundness. *If $\sum_{\underline{\omega} \in H^\ell} p(\underline{\omega}) \neq \tau$ then, for every $\tilde{P}_{\mathrm{SC}}$, $\mathrm{Pr}_{\underline{r} \leftarrow \mathcal{C}^\ell}[\langle \tilde{P}_{\mathrm{SC}}, V_{\mathrm{SC}}(\mathbb{x}_{\mathrm{SC}}; \underline{r})\rangle = (\underline{r}, p(\underline{r}))] < \frac{\ell d}{|\mathcal{C}|}$.*
Above $\langle A, V_{\mathrm{SC}}(\mathbb{x}_{\mathrm{SC}}; \underline{r})\rangle$ is the output of $V_{\mathrm{SC}}(\mathbb{x}_{\mathrm{SC}})$ when interacting with algorithm A using randomness $\underline{r}$.

The lemma directly follows from a generalization of the Schwartz–Zippel lemma over modules.

Lemma 2.3. *Let R be a ring, M an R-module, and $f \in M[X_1, \ldots, X_\ell]$ a non-zero polynomial of total degree D. If $\mathcal{C}$ is a sampling set for M then $\mathrm{Pr}_{\underline{r} \leftarrow \mathcal{C}^\ell}[f(\underline{r}) = 0] \leq \frac{D}{|\mathcal{C}|}$.*

The proof of Lemma 2.3 follows the same approach as the usual inductive proof of the standard Schwartz–Zippel lemma. The properties of $\mathcal{C}$ are used to establish that a polynomial $f \in M[X]$ of degree D has at most D roots in $\mathcal{C}$, which in turn is used in the base case and in the inductive step.

The sumcheck protocol in the special case when $M = R$ has been used before, e.g., in [CCKP19].

2.2 Sumcheck argument for Pedersen commitments

We describe a cryptographic protocol for proving knowledge of an opening of a Pedersen commitment, whose main subroutine is the sumcheck protocol. We refer to such a protocol as a *sumcheck argument*. Note that for now we ignore the goal of zero knowledge, and instead focus on achieving communication complexity that is much smaller than (indeed, logarithmic in) the message whose knowledge is being proved.

Definition 2. *We index the entries of a vector $\underline{v}$ of length $n = 2^\ell$ via binary strings $(i_1, \ldots, i_\ell) \in \{0,1\}^\ell$, and define the corresponding polynomial $p_{\underline{v}}(X_1, \ldots, X_\ell) := \sum_{i_1,\ldots,i_\ell \in \{0,1\}} v_{i_1,\ldots,i_\ell} X_1^{i_1} \cdots X_\ell^{i_\ell}$.*

Protocol 1: sumcheck argument for Pedersen commitments

For $n = 2^\ell$, the prover and verifier receive as input a commitment key $\underline{\mathsf{G}} \in \mathbb{G}^n$ and commitment $\mathsf{C} \in \mathbb{G}$. The prover also receives as input an opening $\underline{a} \in \mathbb{F}^n$ such that $\mathsf{C} = \langle \underline{a}, \underline{\mathsf{G}} \rangle$.

The prover and verifier engage in a sumcheck protocol for the instance

$$\mathbb{x}_{\mathrm{SC}} := (R = \mathbb{F},\ M = \mathbb{G},\ H = \{-1, 1\},\ \ell = \log n,\ \tau = 2^\ell \mathsf{C},\ \mathcal{C} = \mathbb{F})$$

where the prover uses the polynomial $p(\underline{X}) := p_{\underline{a}}(\underline{X}) \cdot p_{\underline{\mathsf{G}}}(\underline{X})$. After the end of the sumcheck protocol, the prover learns $\underline{r} \in \mathbb{F}^\ell$ and the verifier learns $(\underline{r}, v) \in \mathbb{F}^\ell \times \mathbb{G}$. Then the prover computes and sends $p_{\underline{a}}(\underline{r}) \in \mathbb{F}$ to the verifier, and the verifier computes $p_{\underline{\mathsf{G}}}(\underline{r}) \in \mathbb{G}$ and checks that $p_{\underline{a}}(\underline{r}) \cdot p_{\underline{\mathsf{G}}}(\underline{r}) = v$.

We begin by explaining why Protocol 1 is mathematically well-defined. The "multiplication" operation implicit in the expression $p_{\underline{a}}(\underline{X}) \cdot p_{\underline{\mathsf{G}}}(\underline{X})$, which maps $\mathbb{F}[X_1, \ldots, X_\ell] \times \mathbb{G}[X_1, \ldots, X_\ell] \to \mathbb{G}[X_1, \ldots, X_\ell]$, is a natural extension of the scalar multiplication operation $a \cdot \mathsf{G}$ which maps $\mathbb{F} \times \mathbb{G} \to \mathbb{G}$. For example, consider the polynomials $p_1(X) = a + a' \cdot X \in \mathbb{F}[X]$ and $p_2(X) = \mathsf{G} + X \cdot \mathsf{G}' \in \mathbb{G}[X]$, and let $r \in \mathbb{F}$. The product of $p_1(r)$ and $p_2(r)$ can be written as follows:

$$\begin{aligned} p_1(r) \cdot p_2(r) &= (a + a'r) \cdot (\mathsf{G} + r \cdot \mathsf{G}') \\ &= a \cdot (\mathsf{G} + r \cdot \mathsf{G}') + a'r \cdot (\mathsf{G} + r \cdot \mathsf{G}') \\ &= a \cdot \mathsf{G} + ar \cdot \mathsf{G}' + a'r \cdot \mathsf{G} + a'r^2 \cdot \mathsf{G}' \\ &= a \cdot \mathsf{G} + r \cdot (a \cdot \mathsf{G}' + a' \cdot \mathsf{G}) + r^2 \cdot (a' \cdot \mathsf{G}') \ , \end{aligned}$$

where the second and third equalities follow from the *bilinear properties* of scalar multiplication.[4] This holds for any $r \in \mathbb{F}$, and so it makes sense to define the "scalar multiplication" of $p_1(X)$ and $p_2(X)$:

$$p_1(X) \cdot p_2(X) = (a + a'X) \cdot (\mathsf{G} + X \cdot \mathsf{G}') := a \cdot \mathsf{G} + X \cdot (a \cdot \mathsf{G}' + a' \cdot \mathsf{G}) + X^2 \cdot (a' \cdot \mathsf{G}') \ .$$

The polynomial $p_{\underline{a}}(\underline{X}) \cdot p_{\underline{\mathsf{G}}}(\underline{X})$, whose coefficients lie in $\mathbb{G}$, is defined this way.

Completeness of Protocol 1 follows from the fact that $\sum_{\underline{\omega} \in \{-1,1\}^n} p_{\underline{a}}(\underline{\omega}) \cdot p_{\underline{\mathsf{G}}}(\underline{\omega}) = 2^\ell \langle \underline{a}, \underline{\mathsf{G}} \rangle$. Indeed, each contribution to $\sum_{\underline{\omega} \in \{-1,1\}^\ell} p_{\underline{a}}(\underline{\omega}) \cdot p_{\underline{\mathsf{G}}}(\underline{\omega})$ corresponds to the monomials of $p_{\underline{a}}(\underline{X}) \cdot p_{\underline{\mathsf{G}}}(\underline{X})$ of the form $X_1^{2i_1} \cdots X_\ell^{2i_\ell}$. The coefficient of $X_1^{2i_1} \cdots X_\ell^{2i_\ell}$ in $p_{\underline{a}}(\underline{X}) \cdot p_{\underline{\mathsf{G}}}(\underline{X})$ arises from a multiplication of the monomials in the terms $a_{i_1,\ldots,i_\ell} X_1^{i_1} \cdots X_\ell^{i_\ell}$ and $\mathsf{G}_{i_1,\ldots,i_\ell} X_1^{i_1} \cdots X_\ell^{i_\ell}$, which multiply to give $a_{i_1,\ldots,i_\ell} \cdot \mathsf{G}_{i_1,\ldots,i_\ell} \cdot X_1^{2i_1} \cdots X_\ell^{2i_\ell}$. Thus, $\sum_{\underline{\omega} \in \{-1,1\}^\ell} p_{\underline{a}}(\underline{\omega}) \cdot p_{\underline{\mathsf{G}}}(\underline{\omega}) = 2^\ell \langle \underline{a}, \underline{\mathsf{G}} \rangle$.

[4] For any $a, a' \in \mathbb{F}$ and $\mathsf{G}, \mathsf{G}' \in \mathbb{G}$ we have $(a + a') \cdot \mathsf{G} = a \cdot \mathsf{G} + a' \cdot \mathsf{G}$ and $a \cdot (\mathsf{G} + \mathsf{G}') = a \cdot \mathsf{G} + a \cdot \mathsf{G}'$.

The security guarantee of Protocol 1 is different from that of the sumcheck protocol. The sumcheck protocol has a soundness guarantee: if the polynomial p does not have the claimed sum τ then the verifier accepts with small probability. In contrast, Protocol 1 has a **knowledge soundness** guarantee: there exists an extractor that, given a suitable collection of accepting transcripts for a given commitment key $\underline{\mathsf{G}}$ and commitment C, efficiently finds an opening $\underline{a}$ such that $\mathsf{C} = \langle \underline{a}, \underline{\mathsf{G}} \rangle$.

This difference makes sense: any given Pedersen commitment C can *always* be expressed as a scalar product of some opening $\underline{a}$ and the commitment key generators $\underline{\mathsf{G}}$; in fact, there are many different possible openings $\underline{a}$ for which this is true! Therefore, soundness is not a meaningful notion for Protocol 1.

The security guarantee is summarized by the following lemma, whose proof we sketch in Section 2.2.1.

Lemma 2.4 (informal)**.** *Protocol 1 satisfies the following for every key $\underline{\mathsf{G}} \in \mathbb{G}^n$ and commitment $\mathsf{C} \in \mathbb{G}$.*

- Completeness. *For every $\underline{a} \in \mathbb{F}^n$ such that $\mathsf{C} = \langle \underline{a}, \underline{\mathsf{G}} \rangle$, $\Pr[\langle \mathbf{P}(\underline{\mathsf{G}}, \mathsf{C}, \underline{a}), \mathbf{V}(\underline{\mathsf{G}}, \mathsf{C}) \rangle = 1] = 1$.*
- Knowledge soundness. *Given a suitable tree of accepting transcripts for $\mathbf{V}(\underline{\mathsf{G}}, \mathsf{C})$, one can efficiently extract an opening $\underline{a} \in \mathbb{F}^n$ such that $\mathsf{C} = \langle \underline{a}, \underline{\mathsf{G}} \rangle$.*

Perhaps surprisingly, Protocol 1 is *equivalent* to the "split-and-fold" knowledge protocol for Pedersen commitments introduced in [BCCGP16] (we describe this equivalence in Appendix A). Moreover, knowledge soundness can be established without relying on any computational assumptions, a fact that was noted for the "split-and-fold" knowledge protocol in [ACF20; BDFG20].

2.2.1 Proof sketch of Lemma 2.4

We discuss knowledge soundness. The extractor takes as input 3^ℓ accepting transcripts arranged in a 3-ary tree of depth ℓ, with each path from the root to the leaf identified by a choice of verifier randomness $r_1, \ldots, r_\ell \in \mathbb{F}$. For $i \in [\ell]$, the node at layer $i-1$ corresponding to path $r_1, \ldots, r_{i-1} \in \mathbb{F}$ is labeled with the message sent by the prover given challenges $r_1, \ldots, r_{i-1}$ and has three children nodes each corresponding to a distinct challenge $r_i^{(j)} \in \mathbb{F}$. For $i \in [\ell]$, a prover message for the layer $i-1$ is a quadratic polynomial $q_i[r_1, \ldots, r_{i-1}] \in \mathbb{G}[X]$ sent by the prover in the sumcheck protocol given challenges $r_1, \ldots, r_{i-1}$; and a prover message for the layer ℓ is an opening $w[r_1, \ldots, r_\ell] \in \mathbb{F}$ sent by the prover after the sumcheck protocol. Since transcripts are accepting, we know that: $\sum_{\omega_1 \in \{-1,1\}} q_1(\omega_1) = 2^\ell \mathsf{C}$; for $i \in \{2, \ldots, \ell\}$, $\sum_{\omega \in \{-1,1\}} q_i[r_1, \ldots, r_{i-1}](\omega) = q_{i-1}[r_1, \ldots, r_{i-2}](r_{i-1})$; and $w[r_1, \ldots, r_\ell] \cdot p_{\underline{\mathsf{G}}}(r_1, \ldots, r_\ell) = q_\ell[r_1, \ldots, r_{\ell-1}](r_\ell)$.

The extractor works inductively, processing each layer of the tree starting from the ℓ-th layer and moving upwards towards the root. For $i = \ell, \ldots, 1$ and for every path $(r_1, \ldots, r_{i-1}) \in \mathbb{F}^{i-1}$ in the transcript tree with children $\{r_i^{(j)}\}_{j \in [3]}$, the extractor works as follows.

1. Let $\underline{\mathsf{G}}[r_1, \ldots, r_{i-1}] \in \mathbb{G}^{n/2^{i-1}}$ be the coefficients of $p_{\underline{\mathsf{G}}}(r_1, \ldots, r_{i-1}, X_i, \ldots, X_\ell)$, and let $\underline{\mathsf{G}}_0[r_1, \ldots, r_{i-1}]$ and $\underline{\mathsf{G}}_1[r_1, \ldots, r_{i-1}]$ be the coefficients for monomials without X_i and with X_i respectively. For $j \in [3]$, let $\underline{\mathsf{G}}'[r_1, \ldots, r_{i-1}, r_i^{(j)}] := \underline{\mathsf{G}}[r_1, \ldots, r_{i-1}] + r_i^{(j)} \cdot \underline{\mathsf{G}}_1[r_1, \ldots, r_{i-1}] \in \mathbb{G}^{n/2^i}$ be the coefficients of $p_{\underline{\mathsf{G}}}(r_1, \ldots, r_{i-1}, r_i^{(j)}, X_{i+1}, \ldots, X_\ell)$.
2. We inductively know, for each $j \in [3]$, an opening $w[r_1, \ldots, r_{i-1}, r_i^{(j)}] \in \mathbb{F}^{n/2^i}$ to the commitment $q_i[r_1, \ldots, r_{i-1}](r_i^{(j)}) \in \mathbb{G}$ with respect to the key $\underline{\mathsf{G}}'[r_1, \ldots, r_{i-1}, r_i^{(j)}]$:

$$\begin{aligned}
\langle w[r_1, \ldots, r_{i-1}, r_i^{(1)}], \underline{\mathsf{G}}'[r_1, \ldots, r_{i-1}, r_i^{(1)}] \rangle &= q_i[r_1, \ldots, r_{i-1}](r_i^{(1)}) \ , \\
\langle w[r_1, \ldots, r_{i-1}, r_i^{(2)}], \underline{\mathsf{G}}'[r_1, \ldots, r_{i-1}, r_i^{(2)}] \rangle &= q_i[r_1, \ldots, r_{i-1}](r_i^{(2)}) \ , \\
\langle w[r_1, \ldots, r_{i-1}, r_i^{(3)}], \underline{\mathsf{G}}'[r_1, \ldots, r_{i-1}, r_i^{(3)}] \rangle &= q_i[r_1, \ldots, r_{i-1}](r_i^{(3)}) \ .
\end{aligned}$$

3. Since the polynomial $q_i[r_1, \ldots, r_{i-1}]$ is quadratic, we can use linear algebra on the above three equations to compute a quadratic polynomial $\pi[r_1, \ldots, r_{i-1}] \in \mathbb{F}^{n/2^{i-1}}[X]$ such that

$$\langle \pi[r_1, \ldots, r_{i-1}](X), \underline{\mathsf{G}}[r_1, \ldots, r_{i-1}] \rangle = q_i[r_1, \ldots, r_{i-1}](X) \ .$$

Then we can obtain an opening $w[r_1, \ldots, r_{i-1}] \in \mathbb{F}^{n/2^{i-1}}$ such that

$$\langle w[r_1, \ldots, r_{i-1}], \underline{\mathsf{G}}[r_1, \ldots, r_{i-1}] \rangle = \textstyle\sum_{\omega \in \{-1,1\}} q_i[r_1, \ldots, r_{i-1}](\omega) \ .$$

Observe that:

- If $i > 1$, the verifier's checks imply that $\sum_{\omega \in \{-1,1\}} q_i[r_1, \ldots, r_{i-1}](\omega) = q_{i-1}[r_1, \ldots, r_{i-2}](r_{i-1})$, and so $w[r_1, \ldots, r_{i-1}]$ is an opening to the commitment $q_{i-1}[r_1, \ldots, r_{i-2}](r_{i-1})$ under the key $\underline{\mathsf{G}}[r_1, \ldots, r_{i-1}]$.
- If $i = 1$ (this is the last iteration) then the verifier's checks imply that $\sum_{\omega_1 \in \{-1,1\}} q_1(\omega_1) = 2^\ell \mathsf{C}$, and so w is an opening to the commitment $2^\ell \mathsf{C}$ under the key $\underline{\mathsf{G}}$. Dividing by 2^ℓ yields the desired opening.

A key ingredient of the knowledge extractor is the ability to double the length of known openings by manipulating multiple transcripts for a given recursion round. The Pedersen commitment, being a homomorphism into $\mathbb{G}$, allows this unconditionally. Jumping ahead, this property of a commitment scheme, which we call **invertibility**, may require computational assumptions, and is a central component of our sumcheck argument for the general setting of sumcheck-friendly commitments (see Sections 2.3 and 2.4).

2.3 Sumcheck argument for sumcheck-friendly commitments

We explain how to formulate a sumcheck argument for proving knowledge of an opening for any commitment scheme that satisfies certain functionality and security properties. We proceed in two steps: in Section 2.3.1 we focus on the special case of scalar product protocols under Pedersen commitments to gain intuition, and then in Section 2.3.2 we extend this to apply to a *sumcheck-friendly* commitment.

2.3.1 Scalar-products under Pedersen commitments

In Section 2.2 we have seen how to construct a sumcheck argument for Pedersen commitments. We now write a sumcheck argument that proves knowledge of openings of *two* Pedersen commitments such that the scalar product of the two openings is a publicly-known value. That is, we obtain a knowledge protocol for the commitment scheme CM that, given a commitment key $(\underline{G}, \underline{H})$, maps a message $(\underline{a}, \underline{b})$ to

$$\mathsf{CM.Commit}\Big((\underline{G}, \underline{H}), (\underline{a}, \underline{b})\Big) := (\langle \underline{a}, \underline{G} \rangle, \langle \underline{b}, \underline{H} \rangle, \langle \underline{a}, \underline{b} \rangle) \ .$$

Protocol 2: sumcheck argument for scalar-products under Pedersen commitments

For $n = 2^\ell$, the prover and verifier receive as input commitment keys $\underline{G}, \underline{H} \in \mathbb{G}^n$, commitments $\mathsf{C}_a, \mathsf{C}_b \in \mathbb{G}$ and target value $t \in \mathbb{F}$. The prover also receives as input openings $\underline{a}, \underline{b} \in \mathbb{F}^n$ such that $\mathsf{C}_a = \langle \underline{a}, \underline{G} \rangle$, $\mathsf{C}_b = \langle \underline{b}, \underline{H} \rangle$ and $t = \langle \underline{a}, \underline{b} \rangle$. (I.e., such that $\mathsf{CM.Commit}\big((\underline{G}, \underline{H}), (\underline{a}, \underline{b})\big) = (\mathsf{C}_a, \mathsf{C}_b, t)$.)

The prover and verifier engage in a sumcheck protocol for the instance

$$\mathbb{x}_{\mathrm{SC}} := (R = \mathbb{F}, M = \mathbb{G} \times \mathbb{G} \times \mathbb{F}, H = \{-1, 1\}, \ell = \log n, \tau = (2^\ell \mathsf{C}_a, 2^\ell \mathsf{C}_b, 2^\ell t), \mathcal{C} = \mathbb{F})$$

where the prover uses the polynomial

$$p(\underline{X}) := \Big(p_{\underline{a}}(\underline{X}) \cdot p_{\underline{G}}(\underline{X}),\ p_{\underline{b}}(\underline{X}) \cdot p_{\underline{H}}(\underline{X}),\ p_{\underline{a}}(\underline{X}) \cdot p_{\underline{b}}(\underline{X})\Big) \in (\mathbb{G} \times \mathbb{G} \times \mathbb{F})[X_1, \ldots, X_\ell] \ .$$

After the end of the sumcheck protocol, the prover learns $\underline{r} \in \mathbb{F}^\ell$ and the verifier learns $(\underline{r}, v) \in \mathbb{F}^\ell \times (\mathbb{G} \times \mathbb{G} \times \mathbb{F})$. Then the prover computes and sends $p_{\underline{a}}(\underline{r}), p_{\underline{b}}(\underline{r}) \in \mathbb{F}$ to the verifier, and the verifier computes $p_{\underline{G}}(\underline{r}), p_{\underline{H}}(\underline{r}) \in \mathbb{G}$ and checks that $(p_{\underline{a}}(\underline{r}) \cdot p_{\underline{G}}(\underline{r}),\ p_{\underline{b}}(\underline{r}) \cdot p_{\underline{H}}(\underline{r}),\ p_{\underline{a}}(\underline{r}) \cdot p_{\underline{b}}(\underline{r})) = v$.

Similarly to Section 2.2, the first and second components of the polynomial $p(\underline{X})$ are well-defined because of the bilinearity of scalar multiplication from $\mathbb{F} \times \mathbb{G}$ to $\mathbb{G}$; moreover, the third component of $p(\underline{X})$ is well-defined because it involves the product of two polynomials over $\mathbb{F}$.

Protocol 2 is complete because

$$\sum_{\underline{\omega} \in \{-1,1\}^\ell} p_{\underline{a}}(\underline{\omega}) p_{\underline{G}}(\underline{\omega}) = 2^\ell \langle \underline{a}, \underline{G} \rangle \ , \quad \sum_{\underline{\omega} \in \{-1,1\}^\ell} p_{\underline{b}}(\underline{\omega}) p_{\underline{H}}(\underline{\omega}) = 2^\ell \langle \underline{b}, \underline{H} \rangle \ , \quad \sum_{\underline{\omega} \in \{-1,1\}^\ell} p_{\underline{a}}(\underline{\omega}) p_{\underline{b}}(\underline{\omega}) = 2^\ell \langle \underline{a}, \underline{b} \rangle \ .$$

Moreover, one can show that Protocol 2 satisfies the following knowledge-soundness property: there exists an extractor that, given a suitable collection of accepting transcripts for a given commitment key $(\underline{G}, \underline{H})$ and commitment $\mathsf{C} = (\mathsf{C}_a, \mathsf{C}_b, t)$, efficiently finds an opening $(\underline{a}, \underline{b})$ such that $\mathsf{C} = \mathsf{CM.Commit}((\underline{G}, \underline{H}), (\underline{a}, \underline{b}))$, assuming that the discrete logarithm problem is hard over $\mathbb{G}$. Proving knowledge soundness follows a similar approach to that for Protocol 1 sketched in Section 2.2.1. The main difference is that "inverting" from a level to the previous one involves not only solving linear equations to find openings of commitments corresponding to the first two components of the polynomial $p(\underline{X})$, but also arguing that the scalar-product of these openings equals the third component of the polynomial $p(\underline{X})$. This step relies on the hardness of the discrete logarithm problem over $\mathbb{G}$ (which one may have assumed anyway to make the commitment binding). This is different from Protocol 1, where no assumptions were necessary to establish knowledge soundness, and intuitively is because the commitment scheme involves a quadratic, rather than linear, computation on the message.

2.3.2 Extending to any sumcheck-friendly commitment

The commitments used in Protocols 1 and 2 are examples of a *sumcheck-friendly* commitment scheme. Below we give an informal definition (which omits technicalities such as how commitment randomness is handled).

Definition 3 (informal)**.** *Let $\mathbb{F}$ be a prime-order field and let $\mathbb{M}, \mathbb{K}, \mathbb{C}$ be $\mathbb{F}$-linear spaces. A commitment scheme CM is* **sumcheck-friendly** *if there exists an efficient function $f_{\mathsf{CM}} \colon \mathbb{M} \times \mathbb{K} \to \mathbb{C}$ such that for every commitment key ck and message m it holds that $\mathsf{CM.Commit}(\mathsf{ck}, \mathsf{m}) = \sum_{\underline{\omega} \in H^\ell} f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}))$ where: (i) $H \subseteq \mathbb{F}$ is a domain and $\ell \in \mathbb{N}$ a number of variables; (ii) $p_{\mathsf{m}}(\underline{X}) \in \mathbb{M}[\underline{X}]$ can be efficiently obtained from the message m (and, conversely, m can be efficiently obtained from $p_{\mathsf{m}}(\underline{X})$); (iii) $p_{\mathsf{ck}}(\underline{X}) \in \mathbb{K}[\underline{X}]$ can be efficiently obtained from the commitment key ck; (iv) $f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{X}), p_{\mathsf{ck}}(\underline{X})) \in \mathbb{C}[\underline{X}]$ is a polynomial.*

We can obtain an opening protocol for CM via a sumcheck argument.

Protocol 3: sumcheck argument for sumcheck-friendly commitments

For $n = 2^\ell$, the prover and verifier receive as input commitment key ck and commitment cm. The prover also receives as input an opening m such that $\mathsf{cm} = \mathsf{CM.Commit}(\mathsf{ck}, \mathsf{m})$.

The prover and verifier engage in a sumcheck protocol for the instance

$$\mathbb{x}_{\mathrm{SC}} := (R = \mathbb{F}, M = \mathbb{C}, H, \ell, \tau = \mathsf{cm}, \mathcal{C} = \mathbb{F})$$

where the prover uses the polynomial $p_{\mathsf{m},\mathsf{ck}}(\underline{X}) := f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{X}), p_{\mathsf{ck}}(\underline{X}))$. At the end of the sumcheck protocol, the prover learns $\underline{r} \in \mathbb{F}^\ell$ and the verifier learns $(\underline{r}, v) \in \mathbb{F}^\ell \times \mathbb{C}$. Then the prover computes and sends $p_{\mathsf{m}}(\underline{r})$ to the verifier, and the verifier computes $p_{\mathsf{ck}}(\underline{r})$ and checks that $f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{r}), p_{\mathsf{ck}}(\underline{r})) = v$.

The above opening protocol for the sumcheck-friendly commitment scheme CM has perfect completeness, and also has knowledge soundness if CM is *invertible* (a property that we discuss shortly).

Theorem 3 (informal)**.** *Let* CM *be a sumcheck-friendly commitment scheme. If* CM *is invertible then Protocol 3 is an opening protocol for* CM*: there exists an extractor that given a key* ck*, commitment* cm*, and a suitable tree of accepting transcripts for* $(\mathsf{ck}, \mathsf{cm})$*, finds an opening* m *such that* $\mathsf{cm} = \mathsf{CM.Commit}(\mathsf{ck}, \mathsf{m})$.

Completeness. The sumcheck-friendly property tells us that $\mathsf{cm} = \sum_{\underline{\omega} \in H^\ell} f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}))$, so the completeness of Protocol 3 follows from the completeness of the sumcheck protocol.

Knowledge soundness. Since m can be efficiently obtained from $p_{\mathsf{m}}(\underline{X})$, it suffices for the extractor to recover, from the tree of transcripts, a polynomial $p_{\mathsf{m}}(\underline{X})$ such that $\mathsf{cm} = \sum_{\underline{\omega} \in H^\ell} f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}))$.

The proof strategy is similar to the one described in Section 2.2.1: the extractor proceeds layer by layer, starting from the leaf layer of the tree of transcripts and continuing to the root; for each node in a particular layer, the extractor computes a polynomial obtained from the polynomials associated to the node's children. The desired polynomial $p_{\mathsf{m}}(\underline{X})$ is the polynomial associated to the root of the tree.

The invertibility property facilitates progress from children to parents, and states that given enough openings for a commitment of a layer one can find an opening of a commitment of the previous layer.

Definition 2.5 (informal)**.** CM *is* K***-invertible*** *if there exists an efficient algorithm* $\mathcal{I}$ *satisfying the following. Suppose that* $\mathcal{I}$ *receives* $i \in [\ell]$*, challenge vector* $(r_1, \ldots, r_{i-1}) \in \mathbb{F}^{i-1}$*, distinct challenges* $r_i^{(1)}, \ldots, r_i^{(K)} \in \mathbb{F}$*, opening polynomials* $p_1, \ldots, p_K \in \mathbb{M}[X_{i+1}, \ldots, X_\ell]$*, and commitment polynomial* $q(X) \in \mathbb{C}[X]$ *such that*

$$\forall\, j \in [K]\,,\ q(r_i^{(j)}) = \sum_{\omega_{i+1}, \ldots, \omega_\ell \in H} f_{\mathsf{CM}}\Big(p_j(\omega_{i+1}, \ldots, \omega_\ell), p_{\mathsf{ck}}(r_1, \ldots, r_{i-1}, r_i^{(j)}, \omega_{i+1}, \ldots, \omega_\ell)\Big)\ . \tag{1}$$

Then $\mathcal{I}$ *outputs an opening polynomial* $p \in \mathbb{M}[X_i, \ldots, X_\ell]$ *such that*

$$\sum_{\omega_i \in H} q(\omega_i) = \sum_{\omega_i, \ldots, \omega_\ell \in H} f_{\mathsf{CM}}\Big(p(\omega_i, \ldots, \omega_\ell), p_{\mathsf{ck}}(r_1, \ldots, r_{i-1}, \omega_i, \ldots, \omega_\ell)\Big)\ . \tag{2}$$

The above definition omits technicalities such as the fact that the inputs to the inverter should be restricted to be efficiently generated by an adversary (given the commitment key ck) and the fact that input and output opening polynomials should be restricted to be "admissible" (partial evaluations of p_{m} for some m).

The extractor receives a K-ary tree of accepting transcripts for $(\mathsf{ck}, \mathsf{cm})$. In more detail, for every $i \in [\ell]$ and $(r_1, \ldots, r_{i-1}) \in \mathbb{F}^{i-1}$, $q_i[r_1, \ldots, r_{i-1}] \in \mathbb{C}[X]$ is the polynomial corresponding to the path $(r_1, \ldots, r_{i-1})$ in the transcript tree (the prover's polynomial in the i-th round of the sumcheck protocol for these challenges); moreover, for every $(r_1, \ldots, r_\ell) \in \mathbb{F}^\ell$, $w[r_1, \ldots, r_\ell] \in \mathbb{C}$ is the opening corresponding to

the path $(r_1, \ldots, r_\ell)$ in the transcript tree (sent by the prover after the sumcheck protocol for these challenges). Every transcript is accepting, so we know that for every $(r_1, \ldots, r_\ell) \in \mathbb{F}^\ell$ it holds that

$$\sum_{\omega_1 \in H} q_1(\omega_1) = \mathsf{cm} \quad , \quad \left\{ \sum_{\omega \in H} q_i[r_1, \ldots, r_{i-1}](\omega) = q_{i-1}[r_1, \ldots, r_{i-2}](r_{i-1}) \right\}_{i \in \{2,\ldots,\ell\}} ,$$
$$\text{and} \quad f_{\mathsf{CM}}(w[r_1, \ldots, r_\ell], p_{\mathsf{ck}}(r_1, \ldots, r_\ell), 1) = q_\ell(r_\ell) \ .$$

The extractor iterates over the whole tree, proceeding with $i = \ell, \ldots, 1$. In the iteration for a path $(r_1, \ldots, r_{i-1}) \in \mathcal{C}^{i-1}$ with children $\{r_i^{(j)}\}_{j \in [K]}$, the extractor uses the inverter $\mathcal{I}$ to transform polynomials $\{p[r_1, \ldots, r_{i-1}, r_i^{(j)}]\}_{j \in [K]}$ in $\mathbb{M}[X_{i+1}, \ldots, X_\ell]$ that satisfy Equation (1) into a new polynomial $p[r_1, \ldots, r_{i-1}]$ in $\mathbb{M}[X_i, \ldots, X_\ell]$ that satisfies Equation (2). The initial polynomials $\{p[r_1, \ldots, r_\ell]\}_{(r_1,\ldots,r_\ell) \in \mathcal{C}^\ell}$ are the constant polynomials corresponding to the opening values $\{w[r_1, \ldots, r_\ell]\}_{(r_1,\ldots,r_\ell) \in \mathcal{C}^\ell}$. The fact that transcripts are accepting ensures that the initial polynomials satisfy the required condition, and that each produced polynomial satisfies the invertibility condition for the prior layer.

After all these iterations the extractor has found a polynomial p in $\mathbb{M}[X_1, \ldots, X_\ell]$ such that $\sum_{\omega_1 \in H} q_1(\omega_1) = \sum_{\underline{\omega} \in H^\ell} f_{\mathsf{CM}}(p(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}))$; again by the accepting condition we know that $\sum_{\omega_1 \in H} q_1(\omega_1) = \mathsf{cm}$ so we deduce that $\mathsf{cm} = \sum_{\underline{\omega} \in H^\ell} f_{\mathsf{CM}}(p(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}))$, and the desired polynomial is p.

Whence invertibility? Invertibility is incomparable to the commitment's binding property. For example, the Pedersen commitment scheme is unconditionally invertible (see Section 2.2.1), whereas invertibility for the scalar-product commitment scheme in Protocol 2 relies on the hardness of the discrete logarithm problem. In Section 2.5 we elaborate on how to establish invertibility for different choices of commitment schemes.

Examples. Protocol 3 captures sumcheck arguments for several commitment schemes.

- The Pedersen commitment scheme (used in Protocol 1) is sumcheck-friendly because, for the function $f_{\mathsf{CM}}(a, \mathsf{G}) := 2^{-\ell} a \cdot \mathsf{G}$, for every commitment key $\underline{\mathsf{G}} \in \mathbb{G}^n$ and message $\underline{a} \in \mathbb{F}^n$ it holds that $\mathsf{CM.Commit}(\underline{\mathsf{G}}, \underline{a}) = \sum_{\underline{\omega} \in H^\ell} f_{\mathsf{CM}}(p_{\underline{a}}(\underline{\omega}), p_{\underline{\mathsf{G}}}(\underline{\omega}))$, where $H := \{-1, 1\}$, $\ell := \log n$, and $p_{\underline{a}}(\underline{X}), p_{\underline{\mathsf{G}}}(\underline{X})$ are the multilinear polynomials induced by $\underline{a}, \underline{\mathsf{G}}$ respectively. (See Definition 2.)
- The scalar-product commitment scheme (used in Protocol 2) is sumcheck-friendly because, for the function $f_{\mathsf{CM}}((a, b), (\mathsf{G}, \mathsf{H})) := 2^{-\ell}(a \cdot \mathsf{G}, b \cdot \mathsf{H}, a \cdot b)$, for every commitment key $(\underline{G}, \underline{H}) \in \mathbb{G}^n \times \mathbb{G}^n$ and message $(\underline{a}, \underline{b}) \in \mathbb{F}^n \times \mathbb{F}^n$ it holds that $\mathsf{CM.Commit}((\underline{G}, \underline{H}), (\underline{a}, \underline{b})) = \sum_{\underline{\omega} \in H^\ell} f_{\mathsf{CM}}((p_{\underline{a}}(\underline{\omega}), p_{\underline{b}}(\underline{\omega})), (p_{\underline{G}}(\underline{\omega}), p_{\underline{H}}(\underline{\omega})))$, where $H := \{-1, 1\}$, $\ell := \log n$, and $p_{\underline{a}}(\underline{X}), p_{\underline{b}}(\underline{X}), p_{\underline{G}}(\underline{X}), p_{\underline{\mathsf{H}}_0}(\underline{X})$ are the multilinear polynomials induced by $\underline{a}, \underline{b}, \underline{G}, \underline{H}$ respectively. (See Definition 2.)

More generally, all inner-product commitments in [BMMTV19] are sumcheck-friendly; this includes pairing-based commitment schemes appearing in works such as [LMR19]. Below we describe inner-product commitments via the notion of *sum-bilinear* commitments, which is easier to work with in our setting.

Definition 4. *A commitment scheme* CM *is* **sum-bilinear** *over a finite field* $\mathbb{F}$ *if the key, message, and commitment spaces are* $\mathbb{F}$*-linear spaces and the following properties hold for all commitment keys* $\mathsf{ckL}, \mathsf{ckR} \in \mathbb{K}^n$*, and messages* $\mathsf{mL}, \mathsf{mR} \in \mathbb{M}^n$*:*

$$\begin{aligned} \mathsf{CM.Commit}\,(\mathsf{ckL} + \mathsf{ckR}, \mathsf{mL} + \mathsf{mR}) &= \mathsf{CM.Commit}\,(\mathsf{ckL}, \mathsf{mL}) + \mathsf{CM.Commit}\,(\mathsf{ckR}, \mathsf{mL}) \\ &\quad + \mathsf{CM.Commit}\,(\mathsf{ckL}, \mathsf{mR}) + \mathsf{CM.Commit}\,(\mathsf{ckR}, \mathsf{mR}) \quad \textit{and} \\ \mathsf{CM.Commit}\,(\mathsf{ckL} \| \mathsf{ckR}, \mathsf{mL} \| \mathsf{mR}) &= \mathsf{CM.Commit}\,(\mathsf{ckL}, \mathsf{mL}) + \mathsf{CM.Commit}\,(\mathsf{ckR}, \mathsf{mR}) \ . \end{aligned}$$

Claim (informal)**.** *If* CM *is sum-bilinear then* CM *is sumcheck-friendly.*

Proof sketch. The first property allows us to "lift" the commitment function to a polynomial. For the function $f_{\mathsf{CM}}(\underline{a}, \underline{\mathsf{G}}) = 2^{-\ell}\mathsf{CM.Commit}(\underline{\mathsf{G}}; \underline{a})$, it holds that for every message $\underline{a} \in \mathbb{M}^n$ and commitment key $\underline{\mathsf{G}} \in \mathbb{K}^n$

$$\begin{aligned} f_{\mathsf{CM}}(p_{\underline{a}}(\underline{X}), p_{\underline{\mathsf{G}}}(\underline{X})) &= 2^{-\ell}\mathsf{CM.Commit}\left(p_{\underline{\mathsf{G}}}(\underline{X}), p_{\underline{a}}(\underline{X})\right) \\ &= 2^{-\ell} \sum_{\underline{i},\underline{j} \in \{0,1\}^\ell} \mathsf{CM.Commit}\left(\underline{\mathsf{G}}_{\underline{j}}, \underline{a}_{\underline{i}}\right) \cdot X_1^{i_1+j_1} \cdots X_\ell^{i_\ell+j_\ell} \end{aligned}$$

where $\ell := \log n$, and $p_{\underline{a}}(\underline{X}), p_{\underline{\mathsf{G}}}(\underline{X})$ are the multilinear polynomials induced by $\underline{a}, \underline{\mathsf{G}}$ via Definition 2. The second property implies that $\sum_{\underline{\omega} \in H^\ell} f_{\mathsf{CM}}(p_{\underline{a}}(\underline{\omega}), p_{\underline{\mathsf{G}}}(\underline{\omega})) = \mathsf{CM.Commit}\left(\mathsf{ck}, \mathsf{m}\right)$ for $H = \{-1, 1\}$. □

2.4 Extending sumcheck arguments to modules

We have so far discussed sumcheck arguments for sumcheck-friendly commitment schemes involving a prime-order group and its scalar field. Yet sumcheck arguments can be formulated more generally to capture commitments in other settings, such as groups of unknown order [BFS20] and lattices [BLNS20]. We explain the changes for this generalization, and how they affect completeness and knowledge soundness.

Modules, norms, slackness. To motivate the considerations that arise, we find it helpful to first recall the Pedersen commitment scheme in other cryptographic settings (ignoring for now randomness for hiding).

- *Pedersen over groups of unknown order.* Let $\mathbb{G}$ be a group of unknown order and let $q, p > 2$ be primes that satisfy certain conditions (determined by the type of instantiation of $\mathbb{G}$). A Pedersen commitment is computed as $\mathsf{CM.Commit}(\underline{\mathsf{G}}, \underline{a}) = \langle \underline{a}, \underline{\mathsf{G}} \rangle \in \mathbb{G}$ where the commitment key $\underline{\mathsf{G}}$ equals $(1 \cdot \mathsf{G}, q \cdot \mathsf{G}, \ldots, q^{n-1} \cdot \mathsf{G})$ for a random group element $\mathsf{G} \in \mathbb{G}$ and the message $\underline{a}$ is a vector in $\left((-\frac{p-1}{2}, \frac{p-1}{2}) \cap \mathbb{Z}\right)^n$.
- *Pedersen over lattices.* Let R be a normed ring and let B_{SIS} be a norm bound of "short" ring elements; a popular choice is $R = \mathbb{Z}_q[X]/\langle X^d + 1 \rangle$ and short ring elements in $R(B_{\mathsf{SIS}})$, i.e. elements of R with norm at most B_{SIS}, for a suitable B_{SIS}. A Pedersen commitment is computed as $\mathsf{CM.Commit}(\underline{\mathsf{G}}, \underline{a}) = \langle \underline{a}, \underline{\mathsf{G}} \rangle$ where $\underline{\mathsf{G}}$ is a matrix of random ring elements and $\underline{a}$ is a vector of short ring elements.

These examples suggest that we need to consider algebraic structures that are not necessarily rings but whose scalars are over a ring, and so we rely on the notion of modules over a ring. Moreover, we need to take into account the norms of openings. Finally, we will only be able to extract a "relaxed" opening for a given commitment, which differs from a regular opening in two ways: (i) the opening might have larger norm than an honestly committed value; (ii) the opening might not satisfy the commitment equation but only a related equation parametrized by a *slackness* c, which we model via an opening algorithm $\mathsf{CM.Open}$ that additionally takes c as input. This is similar to what happens for Schnorr protocols in these settings, as we explain in Section 2.5.

Extending the sumcheck-friendly property. We extend the definition of a sumcheck-friendly commitment scheme (Definition 3) as follows: (i) the spaces $\mathbb{M}, \mathbb{K}, \mathbb{C}$ are modules over the same ring R; (ii) the summation domain is a subset H of R; (iii) a message polynomial $p_{\mathsf{m}}(\underline{X})$ is over the module $\mathbb{M}$; (iv) a key polynomial $p_{\mathsf{ck}}(\underline{X})$ is over the module $\mathbb{K}$; (v) the combiner function f_{CM} maps $\mathbb{M} \times \mathbb{K}$ (and a slackness factor) to the module $\mathbb{C}$; (vi) the summation condition now involves an efficient predicate ϕ_{sc} and is as follows:

$$\begin{aligned} &\mathsf{CM.Commit}\left(\mathsf{ck}, \mathsf{m}\right) = \textstyle\sum_{\underline{\omega} \in H^\ell} f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}), 1) \quad \text{and for every slackness } c \\ &\mathsf{CM.Open}\left(\mathsf{ck}, \mathsf{m}, \mathsf{cm}, c\right) = 1 \;\Leftrightarrow\; \phi_{\text{sc}}\big(\mathsf{cm}, \textstyle\sum_{\underline{\omega} \in H^\ell} f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}), c), c\big) = 1 \;. \end{aligned}$$

(Thus Definition 3 is the special case where $\mathbb{M}, \mathbb{K}, \mathbb{C}$ are $\mathbb{F}$-linear, $R = \mathbb{F}$, ϕ_{sc} checks equality of cm and the sum, and there are no slackness factors.)

Extending sumcheck arguments. In the sumcheck argument for a commitment scheme that is sumcheck-friendly according to the extended definition, we must additionally ensure that: (i) we use a challenge set $\mathcal{C} \subseteq R$ for the sumcheck protocol that satisfies certain properties (discussed further below) that facilitate proving knowledge soundness; (ii) we use norm bounds for commitment openings, so the underlying ring R and the module $\mathbb{M}$ must be equipped with a norm. With these in mind, we now rewrite Protocol 3 for the more general setting (differences in blue), which will allow us to capture the different cryptographic settings.

Protocol 4: sumcheck argument for sumcheck-friendly commitments (over modules)

For $n = 2^\ell$, the prover and verifier receive as input commitment key ck and commitment cm. The prover also receives as input an opening m such that $\|p_{\mathsf{m}}(\underline{X})\| \leq B_{\mathsf{C}}$ and $\mathsf{cm} = \mathsf{CM.Commit}(\mathsf{ck}, \mathsf{m})$.

The prover and verifier engage in a sumcheck protocol for the instance

$$\mathbb{x}_{\text{SC}} := (R, M = \mathbb{C}, H, \ell, \tau = \mathsf{cm}, \mathcal{C})$$

where the prover uses the polynomial $p_{\mathsf{m},\mathsf{ck}}(\underline{X}) := f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{X}), p_{\mathsf{ck}}(\underline{X}), 1)$. At the end of the sumcheck protocol, the prover learns $\underline{r} \in R^\ell$ and the verifier learns $(\underline{r}, v) \in R^\ell \times \mathbb{C}$. Then the prover computes and sends $w := p_{\mathsf{m}}(\underline{r})$ to the verifier. Finally the verifier computes $p_{\mathsf{ck}}(\underline{r})$, checks that $\|w\| \leq B_{\mathsf{SA}}$ (for B_{SA} discussed in the completeness property below), and checks that $f_{\mathsf{CM}}(w, p_{\mathsf{ck}}(\underline{r}), 1) = v$.

Completeness. This follows similarly as in the special case considered in Section 2.3.2, with the main difference that the norm bounds must be set so that they hold for any valid execution of the protocol. We need that for any message m (in the message space of the given commitment key ck) such that $\|p_{\mathsf{m}}(\underline{X})\| \leq B_{\mathsf{C}}$ and challenge vector $\underline{r} \in \mathcal{C}^\ell$ it holds that $\|p_{\mathsf{m}}(\underline{r})\| \leq B_{\mathsf{SA}}$. An explicit expression for B_{SA} can be computed in a straightforward way from the maximum norm of a challenge in $\mathcal{C}$, the number of variables ℓ of $p_{\mathsf{m}}(\underline{X})$, the degree of $p_{\mathsf{m}}(\underline{X})$, and B_{C} (a bound on the maximum norm of a coefficient in $p_{\mathsf{m}}(\underline{X})$).

Knowledge soundness. We wish to prove that Protocol 4 is an opening protocol for CM: given a tree of accepting transcripts for the commitment key ck and commitment cm, we can extract a corresponding (relaxed) opening m. Similarly to Section 2.3.2, we argue knowledge soundness based on an invertibility property that generalizes the prior one (Definition 4.12); the challenge set $\mathcal{C}$ is now part of the property.

Definition 2.6 (informal)**.** CM *is* (K, N, ξ)***-invertible*** *if there exists an efficient algorithm* $\mathcal{I}$ *satisfying the following. Suppose that* $\mathcal{I}$ *receives* $i \in [\ell]$*, challenge vector* $(r_1, \ldots, r_{i-1}) \in \mathcal{C}^{i-1}$*, distinct challenges* $r_i^{(1)}, \ldots, r_i^{(K)} \in \mathcal{C}$*, opening polynomials* $p_1, \ldots, p_K \in \mathbb{M}[X_{i+1}, \ldots, X_\ell]$*, commitment polynomial* $q(X) \in \mathbb{C}[X]$*, and slackness* c *such that*

$$\forall\, j \in [K]\,,\ \phi_{\text{sc}}\left(q(r_i^{(j)}),\ \sum_{\omega_{i+1},\ldots,\omega_\ell \in H} f_{\mathsf{CM}}\Big(p_j(\omega_{i+1}, \ldots, \omega_\ell), p_{\mathsf{ck}}(r_1, \ldots, r_{i-1}, r_i^{(j)}, \omega_{i+1}, \ldots, \omega_\ell), c\Big)\right) = 1\ .$$

Then $\mathcal{I}$ *outputs an opening polynomial* $p \in \mathbb{M}[X_i, \ldots, X_\ell]$ *of norm at most* $N \cdot \max_{j \in [K]} \|p_j\|$ *such that*

$$\phi_{\text{sc}}\left(\sum_{\omega_i \in H} q(\omega_i),\ \sum_{\omega_i,\ldots,\omega_\ell \in H} f_{\mathsf{CM}}\Big(p(\omega_i, \ldots, \omega_\ell), p_{\mathsf{ck}}(r_1, \ldots, r_{i-1}, \omega_i, \ldots, \omega_\ell), \xi \cdot c\Big)\right) = 1\ .$$

Theorem 4. *If the sumcheck-friendly commitment scheme* CM *is* (K, N, ξ)*-invertible then Protocol 4 is an opening protocol for* CM*: there exists an extractor that given a commitment key* ck*, commitment* cm *for a message with norm bound* B_{C}*, and a* K*-ary tree of accepting transcripts for* $(\mathsf{ck}, \mathsf{cm})$*, finds an opening* m *with norm* $\|p_{\mathsf{m}}(\underline{X})\| \leq N^{\ell} B_{\mathsf{SA}}$ *such that* $\mathsf{CM.Open}\left(\mathsf{ck}, \mathsf{cm}, \mathsf{m}, \xi^{\ell}\right) = 1$.

Note that since the extractor works over a tree of depth ℓ, the final loss in norm and slackness involves ℓ factors of N and ξ respectively. Technical details for our sumcheck argument are in Section 4. The definition of invertibility there (Definition 4.12) has an extra parameter B_{INV}, which is an absolute upper bound on the norm of a relaxed opening for which invertibility can hold.

The slackness loss ξ depends on the cryptographic setting, and in the settings that we consider, $\xi \neq 1$ in the lattice and in the GUO setting.

2.5 Instantiations of sumcheck-friendly commitments

Our main theorem on sumcheck arguments (Theorem 1) applies to any sumcheck-friendly commitment that is invertible. Below, we summarize how to construct such commitment schemes; details are in Section 5.

- In Section 2.5.1 we introduce secure bilinear modules.
- In Section 2.5.2 we explain how to construct a (generalized) Pedersen commitment scheme from a secure bilinear module, and give intuition for why it is sumcheck-friendly and invertible. In the technical sections, we also discuss other commitment schemes, which capture linear functions and scalar products.
- In Section 2.5.3 we outline how to instantiate secure bilinear modules in different cryptographic settings: (i) prime-order groups; (ii) bilinear groups; (iii) unknown-order groups; and (iv) lattices.

2.5.1 Secure bilinear modules

A *bilinear module* $\mathcal{M} = (R, M_{\mathrm{L}}, M_{\mathrm{R}}, M_{\mathrm{T}}, e)$ consists of a ring R, three modules $M_{\mathrm{L}}, M_{\mathrm{R}}, M_{\mathrm{T}}$ over R, and a non-degenerate bilinear map $e\colon M_{\mathrm{L}} \times M_{\mathrm{R}} \to M_{\mathrm{T}}$; moreover, R and M_{L} are equipped with norms. For notational simplicity we denote $e(\underline{a}, \underline{\mathsf{G}})$ as $\langle \underline{a}, \underline{\mathsf{G}} \rangle$ and define $M(B) := \{m \in M \text{ such that } \|m\| \leq B\}$.

A *bilinear-module generator* is a tuple $\mathsf{BM} = (\mathsf{Setup}, \mathsf{KeyGen})$ where: $\mathsf{BM.Setup}$ (given a security parameter and length parameter n) samples a bilinear module $\mathcal{M}$, integer $h \in \mathbb{N}$, and auxiliary string aux; and $\mathsf{BM.KeyGen}$ (given $\mathsf{BM.Setup}$'s output) samples a vector $\underline{\mathsf{G}} = (\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1)$ in M_{R}^{n+h}.

A bilinear-module generator BM is *secure* if it satisfies the following.

- It satisfies the *bilinear relation assumption*: for a norm bound B_{BRA} specified in aux and given $\underline{\mathsf{G}} \leftarrow \mathsf{BM.KeyGen}$, it is hard to find a non-zero $\underline{a} \in M_{\mathrm{L}}^{n+h}(B_{\mathsf{BRA}})$ such that $\langle \underline{a}, \underline{\mathsf{G}} \rangle = 0$. (This is a natural generalization of the discrete logarithm assumption, the SIS assumption, and others.)

- The integer h is hiding: there is a distribution $\mathcal{U}_{M_{\mathrm{L}}}$ on M_{L}^{h} such that, for every $\underline{a} \in M_{\mathrm{L}}^{n}$, the following two random variables are statistically close:

$$\left\{ (\underline{\mathsf{G}}, \langle \underline{a}, \underline{\mathsf{G}}_0 \rangle + \langle \underline{r}, \underline{\mathsf{G}}_1 \rangle) \middle| \begin{array}{r} \underline{\mathsf{G}} \leftarrow \mathsf{BM.KeyGen} \\ \underline{r} \leftarrow \mathcal{U}_{M_{\mathrm{L}}} \end{array} \right\} \text{ and } \left\{ (\underline{\mathsf{G}}, \langle \underline{r}, \underline{\mathsf{G}}_1 \rangle) \middle| \begin{array}{r} \underline{\mathsf{G}} \leftarrow \mathsf{BM.KeyGen} \\ \underline{r} \leftarrow \mathcal{U}_{M_{\mathrm{L}}} \end{array} \right\} .$$

- The string aux specifies a norm bound B_{C} such that $B_{\mathsf{C}} \leq B_{\mathsf{BRA}}$.

- The string aux specifies pseudoinverse parameters $(\mathcal{C}, \xi, N)$ for (R, M_{T}): for every $m, m^* \in M_{\mathrm{T}}$, $a \in R$, and distinct $c_1, c_2 \in \mathcal{C}$, if $(c_1 - c_2)m = am^*$ then there exists (and one can efficiently find) $r \in R$ such that $\xi m = rm^*$ and $\|r\| \leq N\|a\|$.

2.5.2 Sumcheck-friendly commitments over bilinear modules

We use secure bilinear-module generators to construct several sumcheck-friendly commitment schemes that are invertible: a generalized Pedersen commitment scheme (Section 5.2), as well as commitment schemes that capture linear functions (Section 5.3) and scalar products (Sections 5.4 and 5.5). Below we restrict our technical overview to discuss the Pedersen commitment scheme.

Definition 2.7 (informal)**.** *Let* $\mathsf{BM} = (\mathsf{Setup}, \mathsf{KeyGen})$ *be a secure bilinear-module generator and consider an output* $(\mathcal{M}, h, \mathsf{aux})$ *of* $\mathsf{BM.Setup}$ *(for a message length* n*) and an output* $\underline{\mathsf{G}} = (\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1) \in M_{\mathrm{R}}^n \times M_{\mathrm{R}}^h$ *of* $\mathsf{BM.KeyGen}$*. The* **(generalized) Pedersen commitment scheme** *for messages of length* n *has messages of the form* $\underline{a} \in M_{\mathrm{L}}^n(B_{\mathsf{C}})$*, and a commitment is computed as* $\mathsf{C} := \langle \underline{a}, \underline{\mathsf{G}}_0 \rangle + \langle \underline{\rho}, \underline{\mathsf{G}}_1 \rangle$*, where* ρ *is sampled appropriately from* $M_{\mathrm{L}}^h(B_{\mathsf{C}})$*. An opening with slackness* $c \in R$ *for a commitment* $\mathsf{C} \in M_{\mathrm{T}}$ *under the commitment key* $(\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1) \in M_{\mathrm{R}}^n \times M_{\mathrm{R}}^h$ *is a vector* $(\underline{a}, \underline{\rho}) \in M_{\mathrm{L}}^n(B_{\mathsf{BRA}}) \times M_{\mathrm{L}}^h(B_{\mathsf{BRA}})$ *such that*

$$c \cdot \mathsf{C} = \langle \underline{a}, \underline{\mathsf{G}}_0 \rangle + c \cdot \langle \underline{\rho}, \underline{\mathsf{G}}_1 \rangle \ .$$

The Pedersen commitment scheme is binding under the bilinear relation assumption (which holds because BM is secure) and is hiding by the property of h (which also holds because BM is secure). Moreover, the Pedersen commitment scheme is (unconditionally) sumcheck-friendly; this can be argued in a similar way as for the usual Pedersen commitment scheme (over prime-order groups).

Establishing invertibility, however, is more challenging. Rather than specifically discussing invertibility of the Pedersen commitment, in this informal overview we describe how the fact that BM is secure enables us to (straightforwardly) obtain an extraction algorithm for the (suitably generalized) Schnorr protocol. This protocol is a simple zero-knowledge argument of knowledge for a commitment opening of a given Pedersen commitment, and the extractor is asked to produce a (possibly relaxed) opening for the commitment given two accepting transcripts sharing the same first message. The considerations that arise when establishing knowledge soundness of the (non-succinct) Schnorr protocol are loosely related to, though technically simpler than, those that arise when establishing invertibility for the Pedersen commitment scheme (which in turns leads to succinct arguments of knowledge via our sumcheck arguments).

Definition 2.8 (informal)**.** *In the* **Schnorr protocol** *for the (generalized) Pedersen commitment scheme, the prover and verifier receive a key* $\underline{\mathsf{G}} = (\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1) \in M_{\mathrm{R}}^{n+h}$*, commitment* $\mathsf{C} \in M_{\mathrm{T}}$ *and norm bound* B_{C}*; and the prover additionally receives as witness a message* $\underline{a} \in M_{\mathrm{L}}^n(B_{\mathsf{C}})$ *and randomness* $\underline{\rho} \in M_{\mathrm{L}}^h(B_{\mathsf{C}})$ *such that* $\langle \underline{a}, \underline{\mathsf{G}}_0 \rangle + \langle \underline{\rho}, \underline{\mathsf{G}}_1 \rangle = \mathsf{C}$*. The prover and verifier interact as follows:*

- *the prover samples* $\underline{b} \in M_{\mathrm{L}}^{n+h}(\kappa \|\mathcal{C}\| B_{\mathsf{C}})$*, where* $\|\mathcal{C}\| := \max_{r \in \mathcal{C}} \|r\|$*, and sends* $t := \langle \underline{b}, \underline{\mathsf{G}} \rangle \in M_{\mathrm{T}}$*;*
- *the verifier sends a random challenge* $r \in \mathcal{C}$*;*
- *the prover sends the response* $\underline{s} := r \cdot (\underline{a}, \underline{\rho}) + \underline{b} \in M_{\mathrm{L}}^{n+h}$ *if* $\|\underline{s}\| \leq (\kappa - 1)\|\mathcal{C}\| B_{\mathsf{C}}$ *(otherwise aborts);*
- *the verifier accepts if* $\langle \underline{s}, \underline{\mathsf{G}} \rangle = r \cdot \mathsf{C} + t$ *and* $\|\underline{s}\| \leq (\kappa - 1)\|\mathcal{C}\| B_{\mathsf{C}}$*.*

The parameter κ is chosen such that $\underline{b}$ "masks" $(\underline{a}, \underline{\rho})$. We discuss how to choose κ in Section 2.6, where similar considerations arise in other protocols; here, instead, we focus on discussing knowledge extraction. The extractor recovers an opening of C from two accepting transcripts $(t, r_1, \underline{s}_1)$ and $(t, r_2, \underline{s}_2)$ sharing the same first message t but with distinct challenges r_1 and r_2. First, subtracting the verification equation for one transcript from that of the other transcript shows that $\langle \underline{s}_1 - \underline{s}_2, \underline{\mathsf{G}} \rangle = (r_1 - r_2) \cdot \mathsf{C}$. The fact that BM is secure implies that $(\mathcal{C}, \xi, N)$ are pseudoinverse parameters for (R, M_{T}), so we can compute an $r \in R$ such that $\xi \cdot \mathsf{C} = r\langle \underline{s}_1 - \underline{s}_2, \underline{\mathsf{G}} \rangle$ with $\|r\| \leq N$. Therefore, the extractor has found a relaxed opening $(\underline{a}', \underline{\rho}') := r(\underline{s}_1 - \underline{s}_2)$ such that $\langle \underline{a}', \underline{\mathsf{G}}_0 \rangle + \langle \underline{\rho}', \underline{\mathsf{G}}_1 \rangle = \xi \cdot \mathsf{C}$ with $\|(\underline{a}', \underline{\rho}')\| \leq N2(\kappa - 1)\|\mathcal{C}\| B_{\mathsf{C}}$. (And we see that the norm B_{C} must satisfy $N2(\kappa - 1)\|\mathcal{C}\| B_{\mathsf{C}} \leq B_{\mathsf{BRA}}$.)

The norm computations above ignore *expansion factors* that appear when computing the norms of expressions that involve the multiplication of ring and module elements (see Definitions 3.1 and 3.2).

2.5.3 Instantiations of secure bilinear modules

We summarize how to instantiate secure bilinear-module generators $\mathsf{BM} = (\mathsf{Setup}, \mathsf{KeyGen})$ in different cryptographic settings. Technical details can be found in Section 5.6.

Prime-order groups. $\mathsf{BM.Setup}$ samples a group $\mathbb{G}$ of prime order q and outputs: (i) the bilinear module $(R, M_{\mathrm{L}}, M_{\mathrm{R}}, M_{\mathrm{T}}, e) := (\mathbb{F}_q, \mathbb{F}_q, \mathbb{G}, \mathbb{G}, e)$ where e is scalar multiplication (e sends $a \in \mathbb{F}_q$ and $\mathsf{G} \in \mathbb{G}$ to $a \cdot \mathsf{G} \in \mathbb{G}$) and $\mathbb{F}_q$ is equipped with the trivial norm (zero has norm 0 and all non-zero elements have norm 1); (ii) the norm bound $B_{\mathsf{BRA}} := \infty$ (every element in $\mathbb{F}_q$ has trivial norm at most ∞); (iii) the norm bound $B_{\mathsf{C}} := 1$ (which is at most $B_{\mathsf{BRA}} = \infty$); (iv) the hiding constant $h := 1$ ($\mathcal{U}_{M_{\mathrm{L}}}$ is the uniform distribution over $\mathbb{F}_q$ as discussed below); (v) the pseudoinverse parameters $(\mathcal{C}, \xi, N) := (\mathbb{F}_q, 1, 1)$. $\mathsf{BM.KeyGen}$ samples a random element $\underline{\mathsf{G}} \in M_{\mathrm{R}}^{n+h} = \mathbb{G}^{n+h}$.

The bilinear-module generator BM above is secure, as outlined below.

- The bilinear relation assumption corresponds to the discrete logarithm assumption. (Indeed, since $\mathbb{F}_q$ is equipped with the trivial norm, $M_{\mathrm{L}}(B_{\mathsf{BRA}}) = \mathbb{F}_q(1) = \mathbb{F}_q$, which means that the assumption states that, given $\underline{\mathsf{G}} \in \mathbb{G}^{n+h}$ sampled by $\mathsf{BM.KeyGen}$, it is hard to find a non-zero $\underline{a} \in \mathbb{F}_q^{n+h}$ such that $\langle \underline{a}, \underline{\mathsf{G}} \rangle = 0$.)
- $h = 1$ is a hiding constant because for every $\underline{a} \in \mathbb{F}_q^n$ the random variables $(\underline{\mathsf{G}}, \langle \underline{a}, \underline{\mathsf{G}}_0 \rangle + r \cdot \mathsf{G}_1)$ and $(\underline{\mathsf{G}}, r \cdot \mathsf{G}_1)$ are identical for random $\underline{\mathsf{G}} = (\underline{\mathsf{G}}_0, \mathsf{G}_1) \in \mathbb{G}^{n+1}$ (as sampled by $\mathsf{BM.KeyGen}$) and secret random $r \in \mathbb{F}_q$ (as sampled by $\mathcal{U}_{M_{\mathrm{L}}}$).
- $(\mathcal{C}, \xi, N) = (\mathbb{F}_q, 1, 1)$ are pseudoinverse parameters because, for every distinct $c_1, c_2 \in \mathbb{F}_q$, $m, m^* \in \mathbb{G}$ and $a \in \mathbb{F}_q$, if $(c_1 - c_2)m = am^*$, then $m = (c_1 - c_2)^{-1}am^*$ and $\|(c_1 - c_2)^{-1}\| = 1$.

Bilinear groups. $\mathsf{BM.Setup}$ samples groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_{\mathrm{T}}$ of prime order q equipped with a non-degenerate bilinear map $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_{\mathrm{T}}$ and outputs: (i) the bilinear module $(R, M_{\mathrm{L}}, M_{\mathrm{R}}, M_{\mathrm{T}}, e) := (\mathbb{F}_q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_{\mathrm{T}}, e)$ with $\mathbb{F}_q, \mathbb{G}_1$ equipped with the trivial norm (zero has norm 0 and all non-zero elements have norm 1); (ii) the norm bound $B_{\mathsf{BRA}} = \infty$ (every element in $\mathbb{F}_q$ and $\mathbb{G}_1$ has trivial norm at most ∞); (iii) the norm bound $B_{\mathsf{C}} := 1$ (which is at most $B_{\mathsf{BRA}} = \infty$); (iv) the hiding constant $h := 1$ ($\mathcal{U}_{M_{\mathrm{L}}}$ is the uniform distributions over $\mathbb{G}_1$); (v) the pseudoinverse parameters $(\mathcal{C}, \xi, N) = (\mathbb{F}_q, 1, 1)$. $\mathsf{BM.KeyGen}$ samples a random element $\underline{\mathsf{G}} \in M_{\mathrm{R}}^{n+h} = \mathbb{G}_2^{n+h}$.

The bilinear-module generator BM above is secure via similar arguments as in the discrete logarithm setting (with the bilinear relation assumption corresponding to the double-pairing assumption [AFGHO16]).

GUO setting. There are two instantiations of groups of unknown order (which are related to the commitment schemes in [BFS20]): (a) RSA groups, and (b) class groups of an imaginary quadratic order. $\mathsf{BM.Setup}$ selects a group $\mathbb{G}$ and primes $q, p > 2$, and outputs: (i) the bilinear module

$$(R, M_{\mathrm{L}}, M_{\mathrm{R}}, M_{\mathrm{T}}, e) := (\mathbb{Z}, \mathbb{Z}, \mathbb{G}, \mathbb{G}, e)$$

where e corresponds to group exponentiation; (ii) a norm bound B_{BRA} which is equal to $\frac{q-1}{2}$ for RSA groups and $\frac{q-1}{4}$ for class groups (see below); (iii) the norm bound B_{C}; (iv) the hiding constant $h := \log \frac{2^\lambda |\mathbb{G}|}{B_{\mathsf{C}}}$ ($\mathcal{U}_{M_{\mathrm{L}}}$ is the uniform distributions over the elements of $((-B_{\mathsf{C}}, B_{\mathsf{C}}) \cap \mathbb{Z})^h$ with norm less than B_{C}); (v) the pseudoinverse parameters $(\mathcal{C}, \xi, N) := ((-B_{\mathsf{C}}, B_{\mathsf{C}}) \cap \mathbb{Z}, \mathrm{lcm}([p-1]), \mathrm{lcm}([p-1]))$. $\mathsf{BM.KeyGen}$ samples a random element G in $\mathbb{G}$ and outputs $(\mathsf{G}, q\mathsf{G}, \ldots, q^{n+h-1}\mathsf{G}) \in M_{\mathrm{R}}^{n+h}$.

The bilinear-module generator BM above is secure, as outlined below.

- The bilinear relation assumption with norm bound B_{BRA} is implied by the Order Assumption [BFS20] for the sampled group $\mathbb{G}$. The Order Assumption is implied by the Adaptive Root Assumption [BBBPWM18; Wes19]. The difference in B_{BRA} for RSA groups and class groups is related to the fact that computing square roots is easy in class groups.
- $h = \log \frac{2^\lambda |\mathbb{G}|}{B_{\mathsf{C}}}$ is a hiding constant as proved in [BFS20; BDFG20].
- $(\mathcal{C}, \xi, N) = ((-B_{\mathsf{C}}, B_{\mathsf{C}}) \cap \mathbb{Z}, \mathrm{lcm}([p-1]), \mathrm{lcm}([p-1]))$ are pseudoinverse parameters.

Lattice setting. $\mathsf{BM.Setup}$ selects a degree d, prime q, and dimension r and outputs: (i) the bilinear module

$$(R, M_{\mathrm{L}}, M_{\mathrm{R}}, M_{\mathrm{T}}, e) := \Big(\mathbb{Z}[X]/\langle X^d+1\rangle, \mathbb{Z}[X]/\langle X^d+1\rangle, (\mathbb{Z}_q[X]/\langle X^d+1\rangle)^r, (\mathbb{Z}_q[X]/\langle X^d+1\rangle)^r, e\Big)$$

where $\mathbb{Z}[X]/\langle X^d+1\rangle$ is equipped with the infinity norm and e maps $x \in \mathbb{Z}[X]/\langle X^d+1\rangle$ and $A \in (\mathbb{Z}_q[X]/\langle X^d+1\rangle)^r$ to $Ax \in (\mathbb{Z}_q[X]/\langle X^d+1\rangle)^r$ via ring multiplication; (ii) a norm bound B_{BRA} related to the SIS assumption (see below); (iii) a norm bound B_{C} that is at most B_{BRA}; (iv) the hiding constant $h := 2r \log \frac{q}{B_{\mathsf{C}}}$ ($\mathcal{U}_{M_{\mathrm{L}}}$ is the uniform distributions over the elements of $(\mathbb{Z}_q[X]/\langle X^d+1\rangle)^h$ with norm less than B_{C}); (v) the pseudoinverse parameters $(\mathcal{C}, \xi, N) := (\{X^i \in \mathbb{Z}[X]/\langle X^d+1\rangle\}_{0 \le i \le 2d-1}, 2, 1)$. $\mathsf{BM.KeyGen}$ samples a random element in $M_{\mathrm{R}}^{n+h} = \big(\mathbb{Z}_q[X]/\langle X^d+1\rangle\big)^{r \times (n+h)}$.

The bilinear-module generator BM above is secure, as outlined below.

- The bilinear relation assumption with norm bound B_{BRA} corresponds to the SIS assumption with norm bound B_{BRA}.
- $h = 2r \log \frac{q}{B_{\mathsf{C}}}$ is a hiding constant as proved in [Mic07; SSTX09].
- $(\mathcal{C}, \xi, N) = (\{X^i \in \mathbb{Z}[X]/\langle X^d+1\rangle\}_{0 \le i \le 2d-1}, 2, 1)$ are pseudoinverse parameters. From [BCKLN14], for every $X^i, X^j \in \mathbb{Z}[X]/\langle X^d+1\rangle$ with $i \neq j$, it holds that the inverse $(X^i - X^j)^{-1}$ over $\mathbb{Z}_q[X]/\langle X^d+1\rangle$ is such that $\|2 \cdot (X^i - X^j)^{-1}\| \le 1$. Hence, for every $m \in M_{\mathrm{T}} = (\mathbb{Z}_q[X]/\langle X^d+1\rangle)^r$, distinct $c_1, c_2 \in \mathcal{C}$, and $m^* \in M_{\mathrm{T}}$, if $(X^i - X^j)m = am^*$, then $2m = 2(X^i - X^j)^{-1} m^*$, where the inverse is over $\mathbb{Z}_q[X]/\langle X^d+1\rangle$, and $\|2(X^i - X^j)^{-1} a\| \le d$.

2.6 Succinct argument for scalar products over rings

We explain how to use sumcheck arguments to obtain zero-knowledge succinct arguments of knowledge for scalar-product relations over rings. This involves choosing a specific sumcheck-friendly commitment to plug in to Theorem 1, and also carefully using randomness to achieve zero knowledge (which is not a guarantee of Theorem 1). Afterwards, in Section 2.7 we explain how to build on this to prove Theorem 2.

We first introduce the notion of *protocol-friendly* bilinear-module generator. A bilinear-module generator BM is *protocol-friendly* if it satisfies the following.

- BM is secure (see Section 2.5.1).
- M_{L} is not merely an R-module but also a ring itself (so that scalar products over M_{L} are defined).[5]
- The string aux specifies $\kappa \in \mathbb{N}$ such that BM is *masking-friendly* (i.e., for every $B \in \mathbb{N}$ with $B_{\mathsf{C}} \le B \le B_{\mathsf{BRA}}/\kappa$ and $\underline{a} \in M_{\mathrm{L}}^n(B)$, $\{\underline{a} + \underline{b}\}_{\underline{b} \leftarrow M_{\mathrm{L}}^n(\kappa B)}$ is close to uniform).
- The string aux specifies an ideal I such that multiplication by ξ (which is part of the pseudoinverse parameters $(\mathcal{C}, \xi, N)$ also in aux) is invertible modulo I.

The instantiations of bilinear-module generators of Section 2.5.3 are also protocol-friendly. Technical details can be found in Section 5.6.

[5] In the pairing setting where M_{L} is not a ring, we define scalar-product commitments differently. See Section 5 for details.

- *Prime-order groups:* $\mathsf{BM.Setup}$ additionally outputs $\kappa := \infty$ and $I := \{0\}$. This means that the argument supports scalar products over $M_{\mathrm{L}}/I = \mathbb{F}_q$, the scalar field of a prime-order group $\mathbb{G}$.
- *Bilinear groups:* $\mathsf{BM.Setup}$ additionally outputs $\kappa := \infty$ and $I := \{0\}$. This means that the argument supports scalar products over $M_{\mathrm{L}}/I = \mathbb{G}_1$ (alternatively, $\mathbb{G}_2$), a source group in the bilinear group.
- *GUO setting:* $\mathsf{BM.Setup}$ additionally outputs $\kappa := O(2^\lambda)$ and $I := n\mathbb{Z}$ for $n \in \mathbb{Z}$ whose prime factors are greater than or equal to p. This means that the argument supports scalar products over $M_{\mathrm{L}}/I = \mathbb{Z}/n\mathbb{Z}$ for any n satisfying these conditions.
- *Lattice setting:* $\mathsf{BM.Setup}$ additionally outputs $\kappa := O(dn)$ and $I := n\mathbb{Z}$ for odd $n \neq -1, 1$. This means that the argument supports scalar products over $M_{\mathrm{L}}/I = \mathbb{Z}/n\mathbb{Z}$ for any n satisfying these conditions.

The commitment scheme that we consider has two-part messages and includes a commitment to their scalar-product; it is the extension of the scalar-product commitment from Section 2.3.1 to bilinear modules.

Definition 2.9 (informal)**.** *Let* $\mathsf{BM} = (\mathsf{Setup}, \mathsf{KeyGen})$ *be a protocol-friendly bilinear-module generator. The* **(generalized) scalar-product commitment scheme** *for messages of length* n *has messages of the form* $(\underline{a}, \underline{b}) \in M_{\mathrm{L}}^n \times M_{\mathrm{L}}^n$ *such that* $\|\underline{a}\|, \|\underline{b}\| \leq B_{\mathsf{C}}$*, and commitment keys of the form* $(\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1, \underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1, \mathsf{U}_0, \underline{\mathsf{U}}_1) \in M_{\mathrm{R}}^{n+h} \times M_{\mathrm{R}}^{n+h} \times M_{\mathrm{R}}^{1+h}$*. A commitment is computed by sampling* $\underline{\rho}_a, \underline{\rho}_b, \underline{\rho}_t \in M_{\mathrm{L}}^h(B_{\mathsf{C}})$ *and computing*

$$\Big(\langle \underline{a}, \underline{\mathsf{G}}_0 \rangle + \langle \underline{\rho}_a, \underline{\mathsf{G}}_1 \rangle,\ \langle \underline{b}, \underline{\mathsf{H}}_0 \rangle + \langle \underline{\rho}_b, \underline{\mathsf{H}}_1 \rangle,\ \langle \underline{a}, \underline{b} \rangle \cdot \mathsf{U}_0 + \langle \underline{\rho}_t, \underline{\mathsf{U}}_1 \rangle\Big)$$

In other words, a commitment is the tuple consisting of three generalized Pedersen commitments: for the first part of the message $\underline{a}$*, for the second part of the message* $\underline{b}$*, and for their scalar product* $\langle \underline{a}, \underline{b} \rangle \in M_{\mathrm{L}}$*.*

A valid opening for a commitment $(\mathsf{C}_a, \mathsf{C}_b, \mathsf{C}_t) \in M_{\mathrm{T}}^3$ *with keys* $(\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1, \underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1\mathsf{U}_0, \underline{\mathsf{U}}_1) \in M_{\mathrm{R}}^{n+h} \times M_{\mathrm{R}}^{n+h} \times M_{\mathrm{R}}^{1+h}$ *and slackness* $c \in R$ *is a vector* $(\underline{a}, \underline{b}, \underline{\rho}_a, \underline{\rho}_b, \underline{\rho}_t) \in M_{\mathrm{L}}^n(B_{\mathsf{BRA}}) \times M_{\mathrm{L}}^n(B_{\mathsf{BRA}}) \times M_{\mathrm{L}}^{3h}(B_{\mathsf{BRA}})$ *such that*

$$c^2 \cdot \mathsf{C} = \big(c \cdot \langle \underline{a}, \underline{\mathsf{G}}_0 \rangle + c^2 \cdot \langle \underline{\rho}_a, \underline{\mathsf{G}}_1 \rangle,\ c \cdot \langle \underline{b}, \underline{\mathsf{H}}_0 \rangle + c^2 \cdot \langle \underline{\rho}_b, \underline{\mathsf{H}}_1 \rangle,\ \langle \underline{a}, \underline{b} \rangle \cdot \mathsf{U}_0 + c^2 \langle \underline{\rho}_t, \underline{\mathsf{U}}_1 \rangle\big) \quad .$$

The generalized scalar-product commitment scheme is binding under the bilinear relation assumption. Moreover, it is sumcheck-friendly (unconditionally). The proof of invertibility follows from algebraic manipulations analogous to the case of generalized Pedersen commitments discussed in Section 2.5; though note that establishing invertibility in this case requires computational assumptions (even in the discrete logarithm setting as discussed in Section 2.3.1).

We give a zero-knowledge succinct argument of knowledge for the following relation related to the scalar-product of committed messages, which we denote by $\mathcal{R}_{\mathsf{CMSP}}$.

Definition 2.10 (informal)**.** *The committed scalar-product relation* $\mathcal{R}_{\mathsf{CMSP}}(c, B_{\mathsf{C}})$ *are the pairs* $(\mathbb{x}, \mathbb{w})$ *where:*

- *The instance* $\mathbb{x}$ *contains*
  - *a protocol-friendly bilinear-module generator* BM*;*
  - *commitment keys* $(\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1, \underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1, \mathsf{U}_0, \underline{\mathsf{U}}_1) \in M_{\mathrm{R}}^n \times M_{\mathrm{R}}^h \times M_{\mathrm{R}}^n \times M_{\mathrm{R}}^h \times M_{\mathrm{R}} \times M_{\mathrm{R}}^h$*;*
  - *commitments* $\mathsf{C}_a, \mathsf{C}_b, \mathsf{C}_t \in M_{\mathrm{T}}$*.*
- *The witness* $\mathbb{w} = (\underline{a}, \underline{\rho}_a, \underline{b}, \underline{\rho}_b, t, \underline{\rho}_t) \in M_{\mathrm{L}}^{2n+1+3h}$ *is such that* $\|\underline{a}\|, \|\underline{\rho}_a\|, \|\underline{b}\|, \|\underline{\rho}_b\|, \|t\|, \|\underline{\rho}\| \leq B_{\mathsf{C}}$ *and*
  - $(\underline{a}, \underline{\rho}_a)$ *is a valid opening of the Pedersen commitments* C_a *with slackness* c*;*
  - $(\underline{b}, \underline{\rho}_b)$ *is a valid opening of the Pedersen commitments* C_b *with slackness* c*;*
  - $(t, \underline{\rho}_t)$ *is a valid opening of the Pedersen commitment* C_t *with slackness* c^2 *and* $t = \langle \underline{a}, \underline{b} \rangle \bmod I$*.*

The relation reasons about scalar-product relations over the quotient ring $R_\bullet = M_{\mathrm{L}}/I$ (M_{L} modulo I) for the ideal $I \subseteq M_{\mathrm{L}}$ specified in aux. In certain settings, such as the lattice and GUO setting, we only extract openings to commitments with slackness $c \neq 1$, we choose I so that we can "cancel out" the slackness c modulo I as part of knowledge extraction algorithms and prove exact scalar-product relations over $R_\bullet$. We now summarize the scalar-product argument; details can be found in Section 6.

The prover begins by computing a commitment $\mathsf{C} \in M_{\mathrm{T}}$ to $\langle \underline{a}, \underline{b} \rangle \in M_{\mathrm{L}}$, which may not be equal to $t \in M_{\mathrm{L}}$. Then the prover and verifier engage in the these sub-protocols: (i) an interactive reduction masking the three Pedersen commitments to $\underline{a}, \underline{b}, t$, converting them into a single scalar-product commitment; (ii) a sumcheck argument to prove knowledge of an opening to the scalar-product commitment; and (iii) a consistency check that the committed values $\langle \underline{a}, \underline{b} \rangle$ and t equal modulo I.

Reduction to a sumcheck argument. The prover samples masking values $\underline{y}_a$ and $\underline{y}_b$ to rerandomize the commitments to $\underline{a}, \underline{b}, \langle \underline{a}, \underline{b} \rangle$: the prover sends commitments to $\underline{y}_a$ and $\underline{y}_b$, and also to $v_1 := \langle \underline{a}, \underline{y}_b \rangle + \langle \underline{b}, \underline{y}_a \rangle$ and $v_0 := \langle \underline{y}_a, \underline{y}_b \rangle$ (which depend only on $\underline{a}, \underline{b}, \underline{y}_a, \underline{y}_b$). Then the verifier sends to the prover a random challenge $\alpha \in \mathcal{C}$. Then the prover computes $\underline{e}_a := \alpha \underline{a} + \underline{y}_a$, $\underline{e}_b := \alpha \underline{b} + \underline{y}_b$, and $\langle \underline{e}_a, \underline{e}_b \rangle = \alpha^2 \langle \underline{a}, \underline{b} \rangle + \alpha v_1 + v_0$. The openings of the rerandomized commitments do not leak any information about $\underline{a}$ or $\underline{b}$, and so the prover can safely send the corresponding commitment randomness to the verifier. Finally, the prover and verifier engage in a sumcheck argument on the scalar-product commitment consisting of the commitments to $\underline{e}_a$, $\underline{e}_b$, and $\langle \underline{e}_a, \underline{e}_b \rangle$. Since the sumcheck argument is invoked on inputs that have been masked, zero knowledge is ensured (i.e., no information about the witness $\mathbb{w} = (\underline{a}, \underline{\rho}_a, \underline{b}, \underline{\rho}_b, t, \underline{\rho}_t)$ is revealed) even though the sumcheck argument itself is not zero knowledge.

Checking consistency modulo I. The sumcheck argument merely convinces the verifier that the prover knows a witness for the scalar-product commitment $(\mathsf{C}_a, \mathsf{C}_b, \mathsf{C})$, while the verifier additionally wants to know that the openings of C and C_t are equal modulo I. For this, we rely on a protocol on the commitments to $\langle \underline{a}, \underline{b} \rangle$ and t to check that they are equivalent modulo I. First, before receiving the verifier's challenge α, the prover samples a masking value ζ, and sends to the verifier its Pedersen commitment C_ζ and its reduction $\zeta \bmod I$ (in the clear); after receiving α the prover sends to the verifier the value $\bar{v} := \alpha \cdot (\langle \underline{a}, \underline{b} \rangle - t) + \zeta$. The verifier then checks that $\bar{v} = \zeta \bmod I$, and that $\bar{v}$ is a valid opening for the commitment to $\alpha \cdot (\langle \underline{a}, \underline{b} \rangle - t) + \zeta$ (for appropriate commitment randomness). Intuitively, if $\bar{v} = \alpha \cdot (\langle \underline{a}, \underline{b} \rangle - t) + \zeta$ for two distinct values of α, then one can solve linear equations to deduce that $\xi \cdot (\langle \underline{a}, \underline{b} \rangle - t) = 0 \bmod I$. Then, since multiplication by the constant ξ from the pseudoinverse parameters $(\mathcal{C}, \xi, N)$ is invertible modulo I (this is required by the protocol-friendly property), we conclude that $\langle \underline{a}, \underline{b} \rangle = t \bmod I$.

2.7 Succinct argument for R1CS over rings

We explain the main ideas behind Theorem 2, which provides a zero-knowledge succinct argument of knowledge for R1CS over rings. Recall that the R1CS problem over a ring $R_\bullet$ asks: given coefficient matrices $A, B, C \in R_\bullet^{n \times n}$ and an instance vector $\underline{x}$ over $R_\bullet$, is there a witness vector $\underline{w}$ over $R_\bullet$ such that $\underline{z} := (\underline{x}, \underline{w}) \in R_\bullet^n$ satisfies $A\underline{z} \circ B\underline{z} = C\underline{z}$? To a first order, Theorem 2 is proved by reducing the R1CS problem over $R_\bullet$ to several scalar-product sub-problems over $R_\bullet$, and then relying on the zero-knowledge succinct argument for scalar products in Section 2.6. This implies that we support R1CS over the rings supported in that section: $R_\bullet = M_{\mathrm{L}}/I$, where M_{L} is the left module of a protocol-friendly bilinear module, and $I \subseteq M_{\mathrm{L}}$ is an ideal. As with our scalar-product arguments, I is used to cancel out slackness factors and prove exact relations. Below we summarize the reduction from R1CS to scalar products.

The prover $\mathbf{P}$ sends commitments to the full assignment $\underline{z} \in R_\bullet^n$ and to its linear combinations $\underline{z}_A, \underline{z}_B \in$

$R_\bullet^n$. Then $\mathbf{P}$ is left to convince the verifier $\mathbf{V}$ that the committed information satisfies these conditions:

$$\underline{z}_A = A\underline{z} \quad , \quad \underline{z}_B = B\underline{z} \quad , \quad \underline{z}_A \circ \underline{z}_B = C\underline{z} \quad , \quad \underline{x} \text{ is a prefix of } \underline{z} \ .$$

To reduce the first three conditions, the verifier $\mathbf{V}$ sends a structured challenge vector $\underline{r}$. Multiplying on the left by $\underline{r}^\intercal$ reduces the first three conditions to $\langle \underline{r}, \underline{z}_A \rangle = \langle \underline{r}_A, \underline{z} \rangle$, $\langle \underline{r}, \underline{z}_B \rangle = \langle \underline{r}_B, \underline{z} \rangle$, $\langle \underline{r} \circ \underline{z}_A, \underline{z}_B \rangle = \langle \underline{r}_C, \underline{z} \rangle$; here we defined $\underline{r}_A := \underline{r}^\intercal A$, $\underline{r}_B := \underline{r}^\intercal B$, and $\underline{r}_C := \underline{r}^\intercal C$. Moreover, to reduce the last condition, the verifier $\mathbf{V}$ sends a random challenge vector $\underline{s}$ of the same length as $\underline{x}$; padding $\underline{s}$ with zeroes to get $\underline{s}'$ of the same length as $\underline{z}$, we have $\langle \underline{s}', \underline{z} \rangle = \langle \underline{s}, \underline{x} \rangle$. Note that both parties can each individually compute $\underline{r}_A, \underline{r}_B, \underline{r}_C$ by right-multiplying $\underline{r}$ by A, B, C respectively, and also both parties can each individually compute $\langle \underline{s}, \underline{x} \rangle$.

Next, the prover $\mathbf{P}$ sends a commitment to $\underline{z}'_A := \underline{r} \circ \underline{z}_A$, and also commitments to $\alpha := \langle \underline{r}_A, \underline{z} \rangle$, $\beta := \langle \underline{r}_B, \underline{z} \rangle$, and $\gamma := \langle \underline{r}_C, \underline{z} \rangle$. Then the prover and verifier engage in scalar-product sub-protocols (described in Section 2.6) to verify these 7 scalar products (recall each party can compute $\langle \underline{s}, \underline{x} \rangle$):

$$\begin{matrix} \langle \underline{r}, \underline{z}_A \rangle = \alpha \\ \langle \underline{r}_A, \underline{z} \rangle = \alpha \end{matrix} \quad , \quad \begin{matrix} \langle \underline{r}, \underline{z}_B \rangle = \beta \\ \langle \underline{r}_B, \underline{z} \rangle = \beta \end{matrix} \quad , \quad \begin{matrix} \langle \underline{z}'_A, \underline{z}_B \rangle = \gamma \\ \langle \underline{r}_C, \underline{z} \rangle = \gamma \end{matrix} \quad , \quad \langle \underline{s}', \underline{z} \rangle = \langle \underline{s}, \underline{x} \rangle \ .$$

The prover and verifier use an additional challenge vector $\underline{y}$ and 2 further scalar-product sub-protocols to check that $\langle \underline{z}'_A, \underline{y} \rangle = \langle \underline{z}_A, \underline{r} \circ \underline{y} \rangle$, which shows that $\underline{z}'_A$ was correctly computed from $\underline{z}_A$ and $\underline{r}$.

All commitments in the protocol are hiding, and hence do not leak any information about the witness vector $\underline{w}$. Hence the zero-knowledge property of the above protocol directly reduces to the zero-knowledge property of the scalar-product sub-protocols.

We conclude by noting that if we instantiate the bilinear module with lattices then Theorem 2 gives Corollary 1: a zero-knowledge succinct argument of knowledge for R1CS based on the SIS assumption.

Technical details can be found in Section 7.

Remark 2.11 (on succinct verification)**.** The verifier complexity in the above succinct argument for R1CS is proportional to the description of the R1CS instance. Nevertheless, one could aim for a succinct verifier in appropriate settings: when the R1CS matrices have structure [BCGGRS19] or in the preprocessing setting [CHMMVW20; COS20; Set20; BCG20; BCL20]. We believe that the tools developed in this paper can be extended to these settings, and leave additionally achieving a succinct verifier to future work.

Remark 2.12 (using polynomial commitments instead)**.** The above protocol is built atop scalar-product sub-protocols, similar to [BCG20; BCL20]. We could have also built the protocol atop a polynomial commitment scheme (such as the one we discuss in Section 2.8), following the approaches in [CHMMVW20; COS20; Set20]. This alternate route offers several tradeoffs, and we leave exploring these tradeoffs to future work.

2.8 Commitments to linear functions and polynomials over modules

As another application of sumcheck arguments, we construct commitment schemes for linear functions over modules. This directly leads to polynomial commitment schemes over modules, as we explain.

Linear-function commitments. The first step is to extend the (generalized) Pedersen commitment scheme to linear functions, similar to prior works [AC20; BDFG20]. Let $\mathcal{M} = (R, M_{\mathrm{L}}, M_{\mathrm{R}}, M_{\mathrm{T}}, e)$ be a bilinear module. Let $\mathsf{mS} \in M_{\mathrm{L}}^n$, and consider the R-linear function f over M_{L} that maps $\underline{z} \in R^n$ to $\langle \underline{z}, \mathsf{mS} \rangle \in M_{\mathrm{L}}$. A linear-function commitment (over modules) consists of a Pedersen commitment $\mathsf{C} := \langle \mathsf{mS}, \underline{\mathsf{G}}_0 \rangle + \langle \rho, \underline{\mathsf{G}}_1 \rangle \in M_{\mathrm{T}}$ to the coefficients of f (with a key $(\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1) \in M_{\mathrm{R}}^n \times M_{\mathrm{R}}^h$ and randomness $\rho \in M_{\mathrm{L}}^h$), a public vector $\mathsf{mP} \in R^n$ specifying the evaluation point for f, and the evaluation $v := \langle \mathsf{mP}, \mathsf{mS} \rangle \in M_{\mathrm{L}}$ of f at mP.

We show, in a similar way to the generalized Pedersen commitment scheme, that this linear-function commitment scheme is sumcheck-friendly and, if $\mathcal{M}$ is sampled by a secure bilinear-module generator, invertible. Therefore we can use our sumcheck argument, and thereby obtain an *argument of knowledge for succinctly opening the linear function commitment*. In more detail, the communication complexity is logarithmic in the length n of the vector describing the linear-function, and the knowledge property has the following form: there exists an extractor that given a commitment key $(\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1) \in M_{\mathrm{R}}^n \times M_{\mathrm{R}}^h$, commitment $\mathsf{C} \in M_{\mathrm{T}}$ for a message with norm bound B_{C}, and a 3-ary tree of accepting transcripts for $(\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1, \mathsf{C})$, finds an opening $\mathsf{mS} \in M_{\mathrm{L}}^n$ with norm at most $(2N\|\mathcal{C}\|)^{\log n} \cdot B_{\mathsf{C}}$ and randomness $\rho \in M_{\mathrm{L}}^h$ with norm at most B_{C} such that $\xi^{\log n} \cdot \mathsf{C} = \langle \mathsf{mS}, \underline{\mathsf{G}}_0 \rangle + \xi^{\log n} \cdot \langle \rho, \underline{\mathsf{G}}_1 \rangle$ and $\xi^{\log n} \cdot v = \langle \mathsf{mP}, \mathsf{mS} \rangle$. Here $(\mathcal{C}, \xi, N)$ are the pseudoinverse parameters for (R, M_{T}) output by the bilinear-module generator.

Polynomial commitments. As a direct application of the above, we construct a polynomial commitment scheme for polynomials (over modules). In a polynomial commitment, the prover commits to a polynomial p with coefficients in M_{L} and then later proves the correct evaluation of the polynomial at a desired point. It is easy to see that this is a special case of a linear function commitment. For example, if p is a univariate polynomial of degree d described by its coefficients in the monomial basis, then we can take $\mathsf{mS} \in M_{\mathrm{L}}^{d+1}$ to be the vector of coefficients, and $\mathsf{mP} = (1, z, \ldots, z^d)$ where $z \in R$ is the desired evaluation point, so that $p(z) = \langle \mathsf{mP}, \mathsf{mS} \rangle$. This directly extends to more variables and other polynomial bases (see Remark 5.14).

Therefore, the aforementioned succinct argument of knowledge for linear-function commitments yields a polynomial commitment scheme whose knowledge guarantee has the following form: there exists an extractor that given a commitment key $(\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1) \in M_{\mathrm{R}}^{d+1} \times M_{\mathrm{R}}^h$, commitment $\mathsf{C} \in M_{\mathrm{T}}$ for a message with norm bound B_{C}, and a 3-ary tree of accepting transcripts for $(\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1, \mathsf{C})$, finds a polynomial $p \in M_{\mathrm{L}}[X]$ of degree at most d and with norm at most $(2N\|\mathcal{C}\|)^{\log n} \cdot B_{\mathsf{C}}$ and randomness $\rho \in M_{\mathrm{L}}^h$ with norm at most B_{C} such that $\xi^{\log n} \cdot \mathsf{C} = \langle p, \underline{\mathsf{G}}_0 \rangle + \xi^{\log n} \cdot \langle \rho, \underline{\mathsf{G}}_1 \rangle$ and $\xi^{\log n} \cdot v = p(z)$.

When $\xi = 1$, this is the usual guarantee of a polynomial commitment scheme. However, when $\xi \neq 1$, this relaxed guarantee does not immediately give an opening protocol. This is because $p(z) = \xi^{\log n} \cdot v$ which is different from the claimed evaluation v. Unfortunately, multiplication of elements of M_{L} by ξ might not be invertible, so the knowledge extractor cannot give "$\xi^{-\log n} p$" as its final output.

To address this problem, we require an additional property of $\mathcal{M}$: we require that $\mathcal{M}$ is *quotient-friendly*, and remove the extra factors of ξ in a quotient module. When $\mathcal{M}$ is quotient-friendly, there is a submodule $I \subseteq M_{\mathrm{L}}$ included in the auxiliary information for the bilinear module, such that multiplication by ξ is invertible modulo I. Then, we can view the commitment scheme as a commitment to polynomials over M_{L}/I, and the polynomial "$\xi^{-\log n} p \bmod I$" evaluates to $v \bmod I$ at z.

Theorem 2.13 (informal)**.** *Let $\mathcal{M}$ be a quotient-friendly secure bilinear-module generator which produces R, $\mathcal{C} \subseteq R$, M_{L} and $I \subseteq M_{\mathrm{L}}$. Then, there is a polynomial commitment scheme (e.g., for univariate polynomials in the monomial basis) with coefficients in M_{L}/I where the evaluation protocol is a sumcheck argument. If n is the number of coefficients of the polynomial (e.g., its degree plus one if a univariate polynomial), the evaluation protocol has knowledge error $O(\frac{\log n}{|\mathcal{C}|})$, round complexity $O(\log n)$, communication $O(\log n)$ elements of M_{T}, and prover and verifier complexity $O(n)$ operations in $M_{\mathrm{L}}, M_{\mathrm{R}}$ and $O(n)$ applications of e.*

3 Preliminaries

3.1 Rings and modules

A ring R is a mathematical structure that generalizes a field: R is equipped with addition and multiplication operations, but, unlike in a field, multiplicative inverses need not exist. We will use only commutative rings, where the multiplication operation commutes. A module M over a ring R extends the notion of vector space over a field, where the scalars are elements of a ring.

Norms. We use rings and modules equipped with norms. The definitions below are slightly different than the ones in standard algebra textbooks due to the expansion factors used.

Definition 3.1. *Let R be a ring. A* **norm for** *R is a map $\|\cdot\|_R\colon R \to \mathbb{R}_{\geq 0}$ that satisfies the following properties: (i) $\|0\|_R = 0$ and $\|1\|_R = 1$; (ii) for all $a \in R$, $\|a\|_R = \|-a\|_R$; (iii) for all $a, b \in R$, $\|a+b\|_R \leq \|a\|_R + \|b\|_R$; (iv) there exists a constant "expansion factor" $\gamma_R \in \mathbb{R}_{>0}$ such that, for all $a, b \in R$, $\|ab\|_R \leq \gamma_R \|a\|_R \|b\|_R$.*

Definition 3.2. *Let R be a ring with norm $\|\cdot\|_R$, and let M be an R-module. A* **norm for** *M is a map $\|\cdot\|_M\colon R \to \mathbb{R}_{\geq 0}$ that satisfies the following properties: (i) $\|0\|_M = 0$; (ii) for all $a \in M$, $\|a\|_M = \|-a\|_M$; (iii) for all $a, b \in M$, $\|a+b\|_M \leq \|a\|_M + \|b\|_M$; (iv) there exists a constant "expansion factor" $\gamma_M \in \mathbb{R}_{>0}$ such that, for all $a \in R$ and $b \in M$, $\|ab\|_M \leq \gamma_M \|a\|_R \|b\|_M$.*

Remark 3.3. To simplify notation in later analysis, although multiplication of elements of M and R may cause norm expansion by different factors γ_R and γ_M, we will only use the notation γ_R, which will represent the maximum of these quantities.

Definition 3.4. *For a ring R with norm $\|\cdot\|_R$, $R(B) := \{r \in R : \|r\|_R \leq B\}$ is the set of ring elements with norm at most B; and similarly for a module M and set $M(B)$. For a set $\mathcal{C} \subseteq R$, $\|\mathcal{C}\|_R := \max_{x \in \mathcal{C}} \|x\|_R$.*

For a normed module M, the norm of a vector $\underline{v} \in M^n$ is $\|\underline{v}\|_M := \max_{i \in [n]} \|v_i\|_M$ (the maximum of the norms of all entries of $\underline{v}$).

Polynomials over modules. Let R be a ring and M an R-module. We denote by $M[X_1, \ldots, X_\ell]$ the set of polynomials in variables $X_1, \ldots, X_\ell$ with coefficients in M. A polynomial $p \in M[X_1, \ldots, X_\ell]$ defines a function from R^ℓ to M. Similarly to the case of vectors, if M is normed then the norm of p is defined to be the maximum of the norms of its coefficients.

We often use multilinear polynomials whose coefficients are defined by a vector $\underline{v}$ as follows.

Definition 3.5. *Let R be a ring and M an R-module. For $n \in \mathbb{N}$ a power of 2, set $\ell := \log n$ and let $\underline{v} \in M^n$ be vector whose entries we index via binary strings $(i_1, \ldots, i_\ell) \in \{0,1\}^\ell$. The ℓ-variate polynomial $p_{\underline{v}} \in M[X_1, \ldots, X_\ell]$ is defined as follows:*

$$p_{\underline{v}}(X_1, \ldots, X_\ell) := \sum_{i_1, \ldots, i_\ell \in \{0,1\}} v_{i_1, \ldots, i_\ell} X_1^{i_1} \cdots X_\ell^{i_\ell} \ .$$

We state a straightforward generalization of a lemma from [BCG20] concerning sums of polynomials; we rely on this several times in this paper.

Lemma 3.6. *Let H be a cyclic subgroup (of finite order) of the multiplicative group of a ring R, such that $1-h$ is not a zero-divisor for any $h \in H \setminus \{1\}$. Let M be an R-module and let $p(X_1, \ldots, X_\ell) \in M[X_1, \ldots, X_\ell]$*

be a polynomial. If we denote by $p_{i_1,\ldots,i_\ell} \in M$ the coefficient of $X_1^{i_1} \cdots X_\ell^{i_\ell}$ in the polynomial $p(X_1, \ldots, X_\ell)$, then

$$\sum_{\underline{\omega} \in H^\ell} p(\underline{\omega}) = \left(\sum_{\underline{i} \equiv \underline{0} \bmod |H|} p_{\underline{i}} \right) \cdot |H|^\ell \ . \tag{3}$$

Proof. We prove the case $\ell = 1$; the general case follows by induction on ℓ. Let h_0 be a generator of H. When $\ell = 1$, $p(X) = \sum_i p_i X^i$, so that

$$\sum_{\omega \in H} p(\omega) = \sum_{\omega \in H} \sum_i p_i \omega^i = \sum_i p_i \sum_{j=0}^{|H|-1} (h^j)^i \ .$$

When $i \equiv 0 \bmod |H|$, then $h^i = 1$ so $\sum_{j=0}^{|H|-1} (h^i)^j = |H|$. When $i \not\equiv 0 \bmod |H|$, then $h^i \neq 1$ and $(h^i - 1) \sum_{j=0}^{|H|-1} (h^i)^j = h^{|H|} - 1 = 0$; since $(h^i - 1)$ is not a zero-divisor, $\sum_{j=0}^{|H|-1} (h^i)^j = 0$. □

We apply Lemma 3.6 with subgroups H with $|H| = 2$ in this paper. An analogue holds for the case of *additive* subgroups of finite fields, which implies that our results also hold for fields of characteristic 2.

Lemma 3.7. *Let H be an additive subgroup of $\mathbb{F}$. Let a_0 be the (formal) linear term of the subspace polynomial $\prod_{h \in H}(X - h)$. Then*

$$\sum_{\underline{\omega} \in H^\ell} p(\underline{\omega}) = \left(\sum_{\underline{i} \equiv -\underline{1} \bmod |H|} p_{\underline{i}} \right) \cdot |a_0|^\ell \ . \tag{4}$$

Remark 3.8. Lemma 3.7 does not hold if $\mathbb{F}$ is replaced with a ring R. Taking $R = \mathbb{Z}/4\mathbb{Z}$, $H = \{0, 2\}$, and $p(X) = 1$ gives a counterexample: $\sum_{\underline{\omega} \in H^\ell} p(\underline{\omega}) = 2$ but $(\sum_{\underline{i} \equiv -\underline{1} \bmod |H|} p_{\underline{i}}) \cdot |a_0|^\ell = 0$. Nevertheless, to apply the techniques in this paper to rings such as $\mathbb{Z}/2^k\mathbb{Z}$, one can explore the use of Galois extensions as in [ACDEY19; GNS21], to obtain suitable subgroups of size 3 for Lemma 3.6 (though the original motivation of [ACDEY19; GNS21] was to ensure the existence of large *sampling sets* of the type that we discuss next).

Sampling sets. The verifier in protocols will sample challenges from a designated subset $\mathcal{C} \subseteq R$ with certain properties, where R is the ring associated to a certain module M.

Definition 3.9. *A* **sampling set $\mathcal{C}$ for an R-module** *M is a subset of R such that for all $c_1, c_2 \in \mathcal{C}$ with $c_1 \neq c_2$, the mapping $M \to M$ that sends $m \mapsto (c_1 - c_2)m$ is injective.*

Definition 3.10. *A* **strong sampling set $\mathcal{C}$ for an R-module** *M is a subset of R such that for all $c_1, c_2 \in \mathcal{C}$ with $c_1 \neq c_2$, there exists $r \in R$ (depending on c_1 and c_2) such that $r(c_1 - c_2)m = m$ for all $m \in M$.*

The special case $M = R$ recovers the notions of sampling sets for rings in [CCKP19].

One can verify that a strong sampling set for M is also a sampling set for M. Conversely, for many rings, a sampling set is also a strong sampling set, as shown in the following lemma.

Lemma 3.11. *Let R be a finite commutative ring. If $r \in R$ is not a zero-divisor then r is invertible.*

Proof. The function $f\colon R \to R$ defined as $f(x) := r \cdot x$ is a ring homomorphism. The kernel of f contains only 0, as otherwise r would be a zero divisor. By the first Isomorphism Theorem for rings, we know that $R/\ker(f) \simeq \mathrm{Im}(f)$, and we can write $|R| = |\ker(f)| \cdot |\mathrm{Im}(f)|$. But $|\ker(f)| = 1$, so $|R| = |\mathrm{Im}(f)|$. Since R is finite and $\mathrm{Im}(f) \subseteq R$, we deduce that $\mathrm{Im}(f) = R$. Hence, there exists $s \in R$ such that $rs = 1$. □

Note that Lemma 3.11 does not hold over infinite rings. For example, take $R = \mathbb{Z}$ and $r = 2$. (The proof breaks down because $\mathrm{Im}(f) = 2\mathbb{Z}$ has the same cardinality as $\mathbb{Z}$, but is a subset of $\mathbb{Z}$.)

Constants. We associate certain constants to a given sampling set.

Definition 3.12. *Let R be a ring with norm $\|\cdot\|_R$ and $\mathcal{C} \subseteq R$ a set. Let $V_{c_1,\ldots,c_K}$ be the Vandermonde matrix with respect to distinct $c_1, \ldots, c_K \in \mathcal{C}$:*

$$V_{c_1,\ldots,c_K} := \begin{bmatrix} 1 & c_1 & \cdots & c_1^{K-1} \\ 1 & c_2 & \cdots & c_2^{K-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & c_K & \cdots & c_K^{K-1} \end{bmatrix}$$

and let $A_{c_1,\ldots,c_K}$ be the adjugate of $V_{c_1,\ldots,c_K}$ (which satisfies $A_{c_1,\ldots,c_K} \cdot V_{c_1,\ldots,c_K} = \det(V_{c_1,\ldots,c_K}) \cdot I_K$).

For $K \in \mathbb{N}$, we define the K-th inversion constant associated to $\mathcal{C}$ to be

$$\iota(\mathcal{C}, K) := \max_{c_1,\ldots,c_K \in \mathcal{C}} \max_{i,j \in [K]} \|A_{c_1,\ldots,c_K}[i,j]\|_R \ .$$

Definition 3.13. *Let R be a ring and M an R-module. For $\xi \in R$ and $N \in \mathbb{N}$ and $\mathcal{C} \subseteq R$ a set, we say that $(\mathcal{C}, \xi, N)$ are* **pseudoinverse parameters** *for (R, M) if for every $a \in R$, $m, m^* \in M$, and distinct $c_1, c_2 \in \mathcal{C}$ it holds that if $(c_1 - c_2)m = a \cdot m^*$ then there exists $r \in R$ such that $\xi \cdot m = r \cdot m^*$ and $\|r\|_R \leq N\|a\|_R$.*

When $M = R$, Definition 3.13 is related to [ACK21, Definition 16] and [AL21, Definition 6].

3.2 Commitments

A (non-interactive) *commitment scheme* is a tuple of algorithms $\mathsf{CM} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Commit}, \mathsf{Open})$ with the following syntax.

- $\mathsf{CM.Setup}(1^\lambda, n) \to \mathsf{pp}$: samples public parameters given a security parameter and a message length.
- $\mathsf{CM.KeyGen}(\mathsf{pp}) \to \mathsf{ck}$: samples a commitment key, which in particular determines a commitment space $\mathbb{C}_{\mathsf{ck}}$, message space $\mathbb{M}_{\mathsf{ck}}$, randomness space $\mathbb{R}_{\mathsf{ck}}$, and slackness space $\mathbb{S}_{\mathsf{ck}}$.
- $\mathsf{CM.Commit}(\mathsf{ck}, \mathsf{m}; \rho) \to \mathsf{cm}$: use the commitment key ck to commit to $\mathsf{m} \in \mathbb{M}_{\mathsf{ck}}$ by sampling $\rho \in \mathbb{R}_{\mathsf{ck}}$ according to some (possibly not uniform) distribution and computing a commitment $\mathsf{cm} \in \mathbb{C}_{\mathsf{ck}}$.
- $\mathsf{CM.Open}(\mathsf{ck}, \mathsf{m}, \rho, \mathsf{cm}, c) \to b \in \{0, 1\}$: checks that $\mathsf{cm} \in \mathbb{C}_{\mathsf{ck}}$ is a commitment to the message $\mathsf{m} \in \mathbb{M}_{\mathsf{ck}}$ with randomness $\rho \in \mathbb{R}_{\mathsf{ck}}$ and slackness value $c \in \mathbb{S}_{\mathsf{ck}}$, relative to the commitment key ck.

We require CM to satisfy completeness and binding, and sometimes also hiding, as specified below.

Definition 3.14. CM *is* **complete** *if for every $n \in \mathbb{N}$ and adversary $\mathcal{A}$,*

$$\Pr\left[\mathsf{CM.Open}(\mathsf{ck}, \mathsf{m}, \rho, \mathsf{cm}, 1) = 1 \;\middle|\; \begin{array}{r} \mathsf{pp} \leftarrow \mathsf{CM.Setup}(1^\lambda, n) \\ \mathsf{ck} \leftarrow \mathsf{CM.KeyGen}(\mathsf{pp}) \\ (\mathsf{m} \in \mathbb{M}_{\mathsf{ck}}, \rho \in \mathbb{R}_{\mathsf{ck}}) \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{ck}) \\ \mathsf{cm} \leftarrow \mathsf{CM.Commit}(\mathsf{ck}, \mathsf{m}; \rho) \end{array}\right] = 1 \ .$$

Definition 3.15. CM *is (computationally)* **binding** *if for every* $n \in \mathbb{N}$ *and polynomial-size adversary* $\mathcal{A}$*,*

$$\Pr\left[\begin{array}{c|r} \mathsf{m}_0 \neq \mathsf{m}_1 & \mathsf{pp} \leftarrow \mathsf{CM.Setup}(1^\lambda, n) \\ \mathsf{CM.Open}(\mathsf{ck}, \mathsf{m}_0, \rho_0, \mathsf{cm}, c) = 1 & \mathsf{ck} \leftarrow \mathsf{CM.KeyGen}(\mathsf{pp}) \\ \mathsf{CM.Open}(\mathsf{ck}, \mathsf{m}_1, \rho_1, \mathsf{cm}, c) = 1 & (\mathsf{cm}, \mathsf{m}_0, \mathsf{m}_1, \rho_0, \rho_1, c) \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{ck}) \end{array}\right] = \mathrm{negl}(\lambda) \ .$$

Definition 3.16. CM *is (statistically)* **hiding** *if for every* $n \in \mathbb{N}$ *and adversary* $\mathcal{A}$*,*

$$\Pr\left[\begin{array}{c|r} & \mathsf{pp} \leftarrow \mathsf{CM.Setup}(1^\lambda, n) \\ & \mathsf{ck} \leftarrow \mathsf{CM.KeyGen}(\mathsf{pp}) \\ \mathcal{A}(\mathsf{pp}, \mathsf{ck}, \mathsf{cm}) = b & (\mathsf{m}_0, \mathsf{m}_1) \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{ck}) \\ & b \leftarrow \{0,1\} \\ & \rho \leftarrow \mathbb{R}_{\mathsf{ck}} \\ & \mathsf{cm} \leftarrow \mathsf{CM.Commit}(\mathsf{ck}, \mathsf{m}_b; \rho) \end{array}\right] = \frac{1}{2} + \mathrm{negl}(\lambda) \ .$$

If we the above probability equals $1/2$ *then* CM *is* **perfectly hiding**.

3.3 Interactive arguments

We say that $\mathsf{ARG} = (\mathbf{G}, \mathbf{P}, \mathbf{V})$ is an *interactive argument* of knowledge for a relation $\mathcal{R}$ if it satisfies the following completeness and knowledge properties.

- **Completeness.** For every adversary $\mathcal{A}$,

$$\Pr\left[\begin{array}{c|r} (\mathbb{x}, \mathbb{w}) \notin \mathcal{R} \text{ or} & \mathsf{pp} \leftarrow \mathbf{G}(1^\lambda) \\ \langle \mathbf{P}(\mathsf{pp}, \mathbb{x}, \mathbb{w}), \mathbf{V}(\mathsf{pp}, \mathbb{x}) \rangle = 1 & (\mathbb{x}, \mathbb{w}) \leftarrow \mathcal{A}(\mathsf{pp}) \end{array}\right] = 1 \ .$$

  We also consider constructions with a *completeness error* ϵ, where the above probability is at least $1 - \epsilon(\lambda)$.

- **Witness-extended emulation.** ARG has witness-extended emulation with knowledge error κ if there exists an expected polynomial-time algorithm $\mathcal{E}$ such that for every polynomial-size adversary $\mathcal{A}$ it holds that

$$\left|\Pr\left[\mathcal{A}(\mathsf{aux}, \mathrm{tr}) = 1 \,\middle|\, \begin{array}{r} \mathsf{pp} \leftarrow \mathbf{G}(1^\lambda) \\ (\mathbb{x}, \mathsf{aux}) \leftarrow \mathcal{A}(\mathsf{pp}) \\ \mathrm{tr} \leftarrow \langle \mathcal{A}(\mathsf{aux}), \mathbf{V}(\mathsf{pp}, \mathbb{x}) \rangle \end{array}\right]\right.$$
$$\left. - \Pr\left[\begin{array}{c|r} \mathcal{A}(\mathsf{aux}, \mathrm{tr}) = 1 & \mathsf{pp} \leftarrow \mathbf{G}(1^\lambda) \\ \text{and} & (\mathbb{x}, \mathsf{aux}) \leftarrow \mathcal{A}(\mathsf{pp}) \\ \text{if } \mathrm{tr} \text{ is accepting then } (\mathbb{x}, \mathbb{w}) \in \mathcal{R} & (\mathrm{tr}, \mathbb{w}) \leftarrow \mathcal{E}^{\mathcal{A}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x}) \end{array}\right]\right| \leq \kappa(\lambda) \ .$$

  Above $\mathcal{E}$ has oracle access to (the next-message functions of) $\mathcal{A}(\mathsf{aux})$.

We also consider argument systems with a zero-knowledge property.

- **Semi-honest-verifier (statistical) zero knowledge.** There exists a probabilistic polynomial-time simulator $\mathcal{S}$ such that for every stateful adversary $\mathcal{A}$ the following probabilities are $\mathrm{negl}(\lambda)$-close:

$$\Pr\left[\begin{array}{c|r} (\mathbb{x}, \mathbb{w}) \in \mathcal{R} & \mathsf{pp} \leftarrow \mathbf{G}(1^\lambda) \\ \mathcal{A}(\mathrm{tr}) = 1 & (\mathbb{x}, \mathbb{w}, \rho) \leftarrow \mathcal{A}(\mathsf{pp}) \\ & \mathrm{tr} \leftarrow \langle \mathbf{P}(\mathsf{pp}, \mathbb{x}, \mathbb{w}), \mathbf{V}(\mathsf{pp}, \mathbb{x}; \rho) \rangle \end{array}\right] \text{ and } \Pr\left[\begin{array}{c|r} (\mathbb{x}, \mathbb{w}) \in \mathcal{R} & \mathsf{pp} \leftarrow \mathbf{G}(1^\lambda) \\ \mathcal{A}(\mathrm{tr}) = 1 & (\mathbb{x}, \mathbb{w}, \rho) \leftarrow \mathcal{A}(\mathsf{pp}) \\ & \mathrm{tr} \leftarrow \mathcal{S}(\mathsf{pp}, \mathbb{x}, \rho) \end{array}\right] .$$

  Above, ρ is the randomness used by the verifier $\mathbf{V}$ (and chosen by the adversary $\mathcal{A}$).

3.3.1 Extraction from trees

We say that ARG is *public coin* if each verifier message is a uniform random string (of a prescribed length). The public-coin interactive arguments in this paper have the property that a witness can be extracted from an appropriate tree of accepting transcripts. The definition below is a natural generalization of special-soundness for sigma-protocols (where $m = 1$ and $n_1 = 2$).

Definition 3.17. *Let* ARG *be a public-coin interactive argument for a relation $\mathcal{R}$ where the verifier sends m messages. For $n_1, \ldots, n_m \in \mathbb{Z}$, we say that T is a $(n_1, \ldots, n_m)$-***tree of accepting transcripts for*** $\mathbb{x}$ *if (1) T is a tree of depth m where, for each $i \in [m]$, each vertex at layer i has n_i children (so the tree has $\prod_{i\in[m]} n_i$ leaves); (2) the n_i outgoing edges of every vertex in layer i are labeled with n_i different choices of randomness for the verifier's i-th message; (3) each vertex in layer i is labeled with a prover message; (4) every path from the root to a leaf in the tree is an accepting transcript for the interactive argument.*

Definition 3.18. ARG *has $(n_1, \ldots, n_m)$-***tree extraction** *if there exists an efficient algorithm χ such that*

$$\Pr\left[\begin{array}{c|c} T \text{ is a } (n_1,\ldots,n_m)\text{-tree of accepting transcripts for } \mathbb{x} & \mathsf{pp} \leftarrow \mathbf{G}(1^\lambda) \\ (\mathbb{x}, \mathbb{w}) \notin \mathcal{R} & (\mathbb{x}, T) \leftarrow \mathcal{A}(\mathsf{pp}) \\ & \mathbb{w} \leftarrow \chi(\mathsf{pp}, \mathbb{x}, T) \end{array}\right] = \mathrm{negl}(\lambda) \ .$$

The following lemma from [ACK21] proves the existence of a tree-finding algorithm, which shows that tree extraction implies witness-extended emulation. Throughout this paper we rely on this generic implication in that it will suffice for our technical statements to establish tree extraction for the protocols that we study.

Lemma 3.19 ([ACK21, Lemma 5], adapted)**.** *Let $n_1, \ldots, n_m \colon \mathbb{N} \to \mathbb{N}$ be functions such that $K := \prod_{i=1}^{m} n_i$ is upper bounded by a polynomial. Let* ARG *be a $(2m+1)$-message public-coin interactive argument in which the verifier $\mathbf{V}$ samples each challenge uniformly at random from a challenge set of size $N \geq \max_{i\in[m]} n_i$. There is an algorithm* Tree *such that for any malicious prover $\tilde{\mathbf{P}}$ for* ARG *that makes $\mathbf{V}$ accept with probability at least ϵ, $\mathsf{Tree}^{\tilde{\mathbf{P}}}(\mathsf{pp}, \mathbb{x})$ runs in expected time at most K (where running $\tilde{\mathbf{P}}$ takes unit time) and produces a $(n_1, \ldots, n_m)$-tree of accepting transcripts for $\mathbb{x}$ with probability at least $\epsilon - \frac{\sum_{i=1}^{m}(n_i-1)}{N}$ (otherwise producing a tree that is incomplete or contains non-accepting transcripts). Further, the first transcript produced by* Tree *(corresponding to the first leaf in the tree) is distributed according to $\langle \tilde{\mathbf{P}}, \mathbf{V} \rangle$.*

Lemma 3.20. *If* ARG *has $(n_1, \ldots, n_m)$-tree extraction, and satisfies the conditions in Lemma 3.19, then* ARG *has witness-extended emulation with knowledge error $\kappa = \frac{\sum_{i=1}^{m}(n_i-1)}{N} + \mathrm{negl}(\lambda)$.*

Proof. Let $\mathcal{A}$ be an adversary against witness-extended emulation such that $\langle \mathcal{A}(\mathsf{aux}), \mathbf{V}(\mathsf{pp}, \mathbb{x}) \rangle = 1$ with probability at least ϵ. We construct an algorithm $\mathcal{E}$ for witness-extended emulation:

1. Run $\mathsf{Tree}^{\mathcal{A}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x})$ from Lemma 3.19.
2. If $\mathsf{Tree}^{\mathcal{A}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x})$ finds a $(n_1, \ldots, n_m)$-tree of accepting transcripts for $\mathbb{x}$, then run χ from Definition 3.18 to obtain $\mathbb{w}$; otherwise set $\mathbb{w} := \perp$.
3. Outputs the first transcript tr produced by Tree, along with $\mathbb{w}$.

We argue that $\mathcal{E}$ satisfies witness-extended emulation.

- *Running time.* Tree runs in expected polynomial time and χ runs in polynomial time, so $\mathcal{E}$ runs in expected polynomial time.

- *Failure probability.* Despite producing an accepting transcript, $\mathcal{E}$ may fail to produce a witness $\mathtt{w}$ such that $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$ if Tree fails to produce an $(n_1, \ldots, n_m)$-tree of accepting transcripts, or χ fails to produce a valid witness from an accepting tree. The probability that this occurs is at most $\frac{\sum_{i=1}^{m}(n_i-1)}{N} + \mathrm{negl}(\lambda)$.
- *Distribution.* Consider the two sampling procedures in the definition of witness-extended emulation. Since tr is distributed according to $\langle \mathcal{A}(\mathsf{aux}), \mathbf{V}(\mathsf{pp}, \mathtt{x}) \rangle$ in either case, the probability that $\mathcal{A}(\mathsf{aux}, \mathrm{tr}) = 1$ is the same for either procedure. The probability that $\mathcal{E}$ produces accepting tr with an invalid witness is at most $\frac{\sum_{i=1}^{m}(n_i-1)}{N} + \mathrm{negl}(\lambda)$, which implies the result.

□

4 Sumcheck argument for opening a commitment

We define the notion of a sumcheck-friendly commitment scheme, and then describe a protocol for proving knowledge of openings for such commitment schemes. We call our protocol a sumcheck argument because it builds directly on top of the sumcheck protocol.

Definition 4.1. *A commitment scheme* $\mathsf{CM} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Commit}, \mathsf{Open})$ *is* **sumcheck-friendly** *if for every security parameter* $\lambda \in \mathbb{N}$*, message length* $n \in \mathbb{N}$*, and public parameters* $\mathsf{pp} \in \mathsf{CM.Setup}(1^\lambda, n)$ *there exist a ring* R*, domain* $H \subseteq R$*, challenge set* $\mathcal{C} \subseteq R$*, number of variables* $\ell \in \mathbb{N}$*, modules* $\mathbb{M}, \mathbb{K}, \mathbb{C}$ *over* R *with* $\mathbb{M}$ *having a norm, and efficient functions* $f_{\mathsf{CM}}, g_{\mathsf{CM}}, \phi_{\text{sc}}, \alpha_{\text{sc}}$ *such that for every commitment key* $\mathsf{ck} \in \mathsf{CM.KeyGen}(\mathsf{pp})$*, message* $\mathsf{m} \in \mathbb{M}_{\mathsf{ck}}$*, randomness* $\rho \in \mathbb{R}_{\mathsf{ck}}$*, and slackness* $c \in \mathbb{S}_{\mathsf{ck}}$*:*

- $\mathsf{CM.Commit}(\mathsf{ck}, \mathsf{m}, \rho) - g_{\mathsf{CM}}(\mathsf{ck}, \rho) = \sum_{\underline{\omega} \in H^\ell} f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}), 1)$*; and*
- $\phi_{\text{sc}}\big(\mathsf{cm} - g_{\mathsf{CM}}(\mathsf{ck}, \rho), \sum_{\underline{\omega} \in H^\ell} f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}), c), c\big) = 1$ *if and only if* $\mathsf{CM.Open}(\mathsf{ck}, \mathsf{m}, \rho, \mathsf{cm}, c) = 1$*;*
- *when* $c = 1$*,* ϕ_{sc} *is simply an equality check on its first two inputs;*
- *for every* $i \in \{0, 1, \ldots, \ell\}$*,* $p \in \mathbb{M}[X_{i+1}, \ldots, X_\ell]$*, and* $(r_1, \ldots, r_i) \in \mathcal{C}^i$*,* $\alpha_{\text{sc}}(\mathsf{ck}, p, r_1, \ldots, r_i) = 1$ *if and only if there exists a message* $\mathsf{m} \in \mathbb{M}_{\mathsf{ck}}$ *such that* $p(X_{i+1}, \ldots, X_\ell) = p_{\mathsf{m}}(r_1, \ldots, r_i, X_{i+1}, \ldots, X_\ell)$*.*

Here:

- $p_{\mathsf{m}}(X_1, \ldots, X_\ell)$ *is a polynomial over* $\mathbb{M}$ *that can be efficiently obtained from the message* m *(and, conversely,* m *can be efficiently obtained from* p_{m}*);*
- $p_{\mathsf{ck}}(X_1, \ldots, X_\ell)$ *is a polynomial over* $\mathbb{K}$ *that can be efficiently obtained from the commitment key* ck*;*
- $p_{\text{sc}}(X_1, \ldots, X_\ell) := f_{\mathsf{CM}}(p_{\mathsf{m}}(X_1, \ldots, X_\ell), p_{\mathsf{ck}}(X_1, \ldots, X_\ell), c)$ *is a polynomial over* $\mathbb{C}$*.*

Letting ideg *denote the maximum individual degree of a polynomial, we also define the following degrees:*

$$d_{\mathsf{ck}} := \max_{\mathsf{m} \in \mathbb{M}_{\mathsf{ck}}} \text{ideg}\Big(p_{\mathsf{m}}(\underline{X})\Big) \ ,$$

$$d^\star_{\mathsf{ck}} := \max_{\mathsf{m} \in \mathbb{M}_{\mathsf{ck}}} \max_{c} \text{ideg}\Big(f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{X}), p_{\mathsf{ck}}(\underline{X}), c)\Big) \ .$$

At the end of every sumcheck argument, the verifier will use α_{sc} to check that the opening message w is admissible. We use this to prove the security property of the sumcheck argument. For most of our instantiations, *every* scalar $w \in \mathbb{M}$ will be admissible and this check will be trivial.

Remark 4.2 (weights)**.** Definition 4.1 extends, analogously to [Mei13], to *weighted sums* of the form

$$\sum_{\underline{\omega} \in H^\ell} \mu_1(\omega_1) \cdots \mu_\ell(\omega_\ell) \ f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}), c)$$

with coefficients $\mu_1, \ldots, \mu_\ell \in R^H$. Our sumcheck argument extends to support weighted sums as well.

Remark 4.3 (existence of p_{sc})**.** Given a strong sampling set $S \subseteq R$ for $\mathbb{C}$ that is sufficiently large, one can find a polynomial p_{sc} satisfying the conditions above by *interpolating* $\{f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{a}), p_{\mathsf{ck}}(\underline{a}), 1)\}_{\underline{a} \in S^\ell}$. Alternatively, when f_{CM} satisfies certain homomorphic properties (f_{CM} is R-linear or R-bilinear), it is possible to "lift" f_{CM} from a function $f_{\mathsf{CM}} \colon \mathbb{M} \times \mathbb{K} \times R \to \mathbb{C}$ to a function on $f_{\mathsf{CM}} \colon \mathbb{M}[\underline{X}] \times \mathbb{K}[\underline{X}] \times R \to \mathbb{C}[\underline{X}]$ to show the existence of a suitable f_{CM}, without any additional assumptions on $\mathbb{C}$. We explain this in Appendix B.

Definition 4.4. *The relation $\mathcal{R}_{\text{SC}}(c, B_{\mathsf{C}})$ is the set of tuples*

$$(\mathbb{x}, \mathbb{w}) = \Big((\mathsf{CM}, \mathsf{pp}, \mathcal{C}, \mathsf{ck}, \mathsf{cm}), (\mathsf{m}, \rho)\Big)$$

where CM *is a sumcheck-friendly commitment scheme,* $\mathsf{pp} \in \mathsf{CM.Setup}(1^\lambda, n)$ *(that in particular specifies* $R, H, \ell, \mathbb{M}, \mathbb{K}, \mathbb{C}, f_{\mathsf{CM}}, g_{\mathsf{CM}}, \phi_{\text{sc}}$*),* $\mathcal{C} \subseteq R$*,* $\mathsf{ck} \in \mathsf{CM.KeyGen}(\mathsf{pp})$*,* $\mathsf{cm} \in \mathbb{C}$*,* $\mathsf{m} \in \mathbb{M}_{\mathsf{ck}}$*,* $\|p_{\mathsf{m}}(\underline{X})\|_{\mathbb{M}} \leq B_{\mathsf{C}}$*,* $\rho \in \mathbb{R}_{\mathsf{ck}}$*,* $c \in \mathbb{S}_{\mathsf{ck}}$ *and* $\mathsf{CM.Open}(\mathsf{ck}, \mathsf{m}, \rho, \mathsf{cm}, c) = 1$*.*

Construction 4.5 (sumcheck argument)**.** We describe a public-coin interactive argument $\mathsf{SCA} = (\mathbf{P}, \mathbf{V})$ for the relation in Definition 4.4. The prover $\mathbf{P}$ and verifier $\mathbf{V}$ take as input an instance $\mathbb{x} = (\mathsf{CM}, \mathsf{pp}, \mathcal{C}, \mathsf{ck}, \mathsf{cm})$; the prover $\mathbf{P}$ additionally takes as input a witness $\mathbb{w} = (\mathsf{m}, \rho)$. Here, CM is sumcheck-friendly with respect to the ring R and subset H.

The prover sends the randomness ρ to the verifier. The prover $\mathbf{P}$ and verifier $\mathbf{V}$ engage in a sumcheck protocol for the instance

$$\mathbb{x}_{\text{SC}} := \Big(R, M = \mathbb{C}, H, \ell, \tau = \mathsf{cm} - g_{\mathsf{CM}}(\mathsf{ck}, \rho), \mathcal{C}\Big)$$

where the prover $\mathbf{P}$ uses the polynomial $p_{\text{sc}}(\underline{X}) := f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{X}), p_{\mathsf{ck}}(\underline{X}), 1)$ induced by $\mathbb{x}$ and $\mathbb{w}$.

After the end of the sumcheck protocol, the prover $\mathbf{P}$ learns $\underline{r} \in \mathcal{C}^\ell$ and the verifier $\mathbf{V}$ learns $(\underline{r}, v) \in \mathcal{C}^\ell \times \mathbb{C}$. (If the sumcheck verifier rejects, then $\mathbf{V}$ rejects.) Then the prover $\mathbf{P}$ computes and sends $w := p_{\mathsf{m}}(\underline{r}) \in \mathbb{M}$ to the verifier $\mathbf{V}$. The verifier $\mathbf{V}$ checks that $\|w\|_{\mathbb{M}} \leq B_{\mathsf{C}} \cdot (d_{\mathsf{ck}} + 1)^\ell \, \gamma_R^{\ell d_{\mathsf{ck}}} \, \|\mathcal{C}\|_R^{\ell d_{\mathsf{ck}}}$, checks that $\rho \in \mathbb{R}_{\mathsf{ck}}$, computes $p_{\mathsf{ck}}(\underline{r}) \in \mathbb{K}$, checks that $f_{\mathsf{CM}}(w, p_{\mathsf{ck}}(\underline{r}), 1) = v$, and checks that $\alpha_{\text{sc}}(\mathsf{ck}, w, \underline{r}) = 1$.

Theorem 4.6. *The sumcheck argument* SCA *in Construction 4.5 satisfies the following properties:*

- *Communication: the prover sends $d^\star_{\mathsf{ck}}\ell$ elements of $\mathbb{C}$, an element of $\mathbb{M}$ with norm at most $B_{\mathsf{C}} \cdot (d_{\mathsf{ck}} + 1)^\ell \, \gamma_R^{\ell d_{\mathsf{ck}}} \, \|\mathcal{C}\|_R^{\ell d_{\mathsf{ck}}}$, and an element of $\mathbb{R}$, and the verifier sends ℓ elements of $\mathcal{C}$.*
- *The prover performs the following operations: computing $p_{\text{sc}}(\underline{X})$, and partially evaluating it $O(|H|^{\ell-1})$ times; and $O(d^\star_{\mathsf{ck}}|H|^{\ell-1})$ additions and scalar-multiplications in $\mathbb{C}$.*
- *The verifier performs the following operations: $O(d^\star_{\mathsf{ck}}|H|\ell)$ additions and scalar multiplications in $\mathbb{C}$; 1 evaluation of the polynomial $p_{\mathsf{ck}}(\underline{X})$; 1 evaluation of f_{CM}; 1 evaluation of g_{CM}; one evaluation of α_{sc}; and a norm-check on w.*
- *If* CM *is sumcheck-friendly (see Definition 4.1), then Construction 4.5 has perfect completeness.*
- *If* CM *is invertible with parameter K (see Definition 4.12), then Construction 4.5 has K^ℓ-tree extraction.*

Proof. We prove the theorem via several lemmas. In Lemma 4.9 and Lemma 4.10, we discuss the arithmetic complexity of the prover and of the verifier. In Lemma 4.11 we prove perfect completeness. In Definition 4.12 we define invertibility, and in Lemma 4.13 we prove tree extraction. □

Remark 4.7 (on prover efficiency)**.** The prover costs in Theorem 4.6 are based on a generic prover implementation that computes p_{sc} and then naively evaluates all sums in the sumcheck protocol (Lemma 2.2). This generic method is more expensive than many instantiations of interest because efficiency improvements are possible when given additional information about H or f_{CM}. For example, if f_{CM} is R-linear, or R-bilinear, then prover efficiency can be improved considerably, by avoiding handling p_{sc} directly. This will lead to a linear prover-time in our instantiations. We explain this in Appendix B.

Remark 4.8 (on verifier efficiency)**.** Typically, the most expensive operation for the verifier is computing the polynomial evaluation $p_{\mathsf{ck}}(\underline{r}) \in \mathbb{K}$. For example, in many instantiations, $p_{\mathsf{ck}}(\underline{X})$ is a multilinear polynomial

in $\log n$ variables and computing $p_{\mathsf{ck}}(\underline{r})$ naively (say, via Horner's method) uses $O(n)$ operations in $\mathbb{K}$. In some cases this cost can be avoided (and ultimately lead to an overall succinct verifier time), e.g., by relying on a sub-protocol to outsource the computation of $p_{\mathsf{ck}}(\underline{r})$ to the prover, or by ensuring that $p_{\mathsf{ck}}(\underline{X})$ has a special structure that allows a more efficient evaluation. Both ideas are exploited in [BMMTV19] (they ensure that $p_{\mathsf{ck}}(r)$ is itself a commitment to a polynomial using the scheme of [KZG10], and that this polynomial has a special factorization allowing it to be evaluated in $O(\log n)$ operations).

Separately, computing each of the ℓ verification equations in the sumcheck protocol involves $O(|H|)$ additions and scalar-multiplications in $\mathbb{C}$. This is typically cheap for the verifier, but can be expensive when using weighted sums (Remark 4.2) where the coefficients $\mu_1, \ldots, \mu_\ell$ have large representation. This is the case in [BFS20], where the commitment space $\mathbb{C}$ is an unknown-order group and the scalar multiplications in the verification equations involve large integer scalars; nevertheless, this cost is avoided by outsourcing to the prover the expensive scalar multiplications via the protocol of [Wes19].

4.1 Efficiency

Lemma 4.9. *The prover in Construction 4.5 performs the following operations: computing $p_{\mathrm{sc}}(\underline{X})$, and partially evaluating it $O(|H|^{\ell-1})$ times; and $O(d^\star_{\mathsf{ck}}|H|^{\ell-1})$ additions and scalar-multiplications in $\mathbb{C}$.*

Proof. The prover computes the coefficients of the polynomial $p_{\mathrm{sc}}(\underline{X})$. In the i-th round of the sumcheck protocol, the prover computes $q_i(X) = \sum_{\omega_{i+1},\ldots,\omega_\ell \in H} p_{\mathrm{sc}}(r_1, \ldots, r_{i-1}, X, \omega_{i+1}, \ldots, \omega_\ell) \in \mathbb{C}[X]$, which involves partially evaluating $p_{\mathrm{sc}}(\underline{X})$ at $|H|^{\ell-i}$ points. Each partial evaluation of $p_{\mathrm{sc}}(\underline{X})$ in the sum is a univariate polynomial of degree at most $d^\star_{\mathsf{ck}}$, so computing the sum costs $O(d^\star_{\mathsf{ck}}|H|^{\ell-1})$ additions and scalar-multiplications in $\mathbb{C}$. Summing over i gives the result. □

Lemma 4.10. *The verifier in Construction 4.5 performs the following operations: $O(d^\star_{\mathsf{ck}}|H|\ell)$ additions and scalar multiplications in $\mathbb{C}$; 1 evaluation of the polynomial $p_{\mathsf{ck}}(\underline{X})$; 1 evaluation of the function f_{CM}; 1 evaluation of the function g_{CM}; one evaluation of the function α_{sc}, and a norm check on w.*

Proof. The sumcheck verifier checks that $\sum_{\omega_1 \in H} q_1(\omega_1) = \mathsf{cm}$ and, for $i \in \{2, \ldots, \ell\}$, that $\sum_{\omega_i \in H} q_i(\omega_i) = q_{i-1}(r_{i-1})$. For each round i, the polynomial $q_i(X)$ has degree at most $d^\star_{\mathsf{ck}}$ and must be evaluated at each $\omega \in H$ and at r_i, which is $|H| + 1$ points. Each evaluation costs $d^\star_{\mathsf{ck}}$ additions and scalar-multiplications in $\mathbb{C}$. These evaluations contribute the dominant cost for the sumcheck verifier, which is $O(d^\star_{\mathsf{ck}}|H|\ell)$ additions and scalar multiplications in $\mathbb{C}$. All other verifier operations are easily read off from Construction 4.5. □

4.2 Completeness

Lemma 4.11. *If CM is sumcheck-friendly, then Construction 4.5 has perfect completeness.*

Proof. Let $\big((\mathsf{CM}, \mathsf{pp}, \mathcal{C}, \mathsf{ck}, \mathsf{cm}), (\mathsf{m}, \rho)\big) \in \mathcal{R}_{\mathrm{SC}}(1, B_{\mathsf{C}})$, so that $\mathsf{CM.Open}(\mathsf{ck}, \mathsf{m}, \rho, \mathsf{cm}, c) = 1$. Fix any choice of verifier challenges $\underline{r} \in \mathcal{C}^\ell$. We need to show that the (honest) prover makes the verifier accept. The verifier checks whether $f_{\mathsf{CM}}(w, p_{\mathsf{ck}}(\underline{r})) = v$, whether $\|w\|_{\mathbb{M}} \leq B_{\mathsf{C}} \cdot (d_{\mathsf{ck}} + 1)^\ell \, \gamma_R^{\ell d_{\mathsf{ck}}} \, \|\mathcal{C}\|_R^{\ell d_{\mathsf{ck}}}$, where $w := p_{\mathsf{m}}(\underline{r})$, and whether $\alpha_{\mathrm{sc}}(\mathsf{ck}, w, \underline{r}) = 1$.

Since CM is sumcheck-friendly, and $p_{\mathrm{sc}}(\underline{r})$ is equal to $f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{r}), p_{\mathsf{ck}}(\underline{r}), 1)$ for every choice of $\underline{r}$, we have $\mathsf{cm} = \sum_{\underline{\omega} \in H^\ell} f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}), 1) + g_{\mathsf{CM}}(\mathsf{ck}, \rho) = \sum_{\underline{\omega} \in H^\ell} p_{\mathrm{sc}}(\underline{\omega}) + g_{\mathsf{CM}}(\mathsf{ck}, \rho)$. Thus, $\mathsf{cm} - g_{\mathsf{CM}}(\mathsf{ck}, \rho)$ and $p_{\mathrm{sc}}(\underline{X})$ define a valid sumcheck instance. By completeness of the sumcheck protocol (Lemma 2.2), the sumcheck verifier does not reject and outputs the claim $(\underline{r}, v)$, where $v = p_{\mathrm{sc}}(\underline{r}) = f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{r}), p_{\mathsf{ck}}(\underline{r}), 1)$. The honest prover sends $w := p_{\mathsf{m}}(\underline{r})$, so the verifier's check that $v = f_{\mathsf{CM}}(w, p_{\mathsf{ck}}(\underline{r}), 1)$ succeeds. Further, w is clearly admissible, so $\alpha_{\mathrm{sc}}(\mathsf{ck}, w, \underline{r}) = 1$.

Finally, we show that $\|w\|_{\mathbb{M}} \leq B_{\mathsf{C}} \cdot (d_{\mathsf{ck}}+1)^{\ell}\, \gamma_R^{\ell d_{\mathsf{ck}}}\, \|\mathcal{C}\|_R^{\ell d_{\mathsf{ck}}}$. Since $p_{\mathsf{m}}(\underline{X})$ has ℓ variables and individual degree d_{ck}, we can write $p_{\mathsf{m}}(\underline{X}) = \sum_{i_1,\ldots,i_\ell=0}^{d_{\mathsf{ck}}} p_{i_1,\ldots,i_\ell} X_1^{i_1} \cdots X_\ell^{i_\ell}$. We can then bound the norm of $p_{\mathsf{m}}(\underline{r})$:

$$\begin{aligned}
\|p_{\mathsf{m}}(\underline{r})\|_{\mathbb{M}} &\leq \sum_{i_1,\ldots,i_\ell=0}^{d_{\mathsf{ck}}} \|p_{i_1,\ldots,i_\ell} r_1^{i_1} \cdots r_\ell^{i_\ell}\|_{\mathbb{M}} \\
&\leq \sum_{i_1,\ldots,i_\ell=0}^{d_{\mathsf{ck}}} \gamma_R^{i_1+\cdots+i_\ell} \|p_{i_1,\ldots,i_\ell}\|_{\mathbb{M}} \|r_1\|_{\mathbb{M}}^{i_1} \cdots \|r_\ell\|_{\mathbb{M}}^{i_\ell} \\
&\leq B_{\mathsf{C}} \cdot (d_{\mathsf{ck}}+1)^{\ell}\, \gamma_R^{\ell d_{\mathsf{ck}}}\, \|\mathcal{C}\|_R^{\ell d_{\mathsf{ck}}} \quad .
\end{aligned}$$

The first inequality follows from the triangle inequality. The second follows from the multiplicative property of the norm on $\mathbb{M}$. The final inequality follows from the fact that $\|p_{\mathsf{m}}(\underline{X})\|_{\mathbb{M}} \leq B_{\mathsf{C}}$, and the fact that norms of challenges are bounded by $\|\mathcal{C}\|_R$. □

4.3 Knowledge soundness

We define invertibility and prove knowledge soundness of our protocol via tree extraction (see Lemma 3.20).

Definition 4.12. *A sumcheck-friendly commitment scheme* CM *is* $(K, B_{\mathsf{INV}}, N, \xi)$**-invertible** *if there exists a polynomial-time inverter algorithm* $\mathcal{I}$ *such that for every security parameter* $\lambda \in \mathbb{N}$*, message length* $n \in \mathbb{N}$*, and polynomial-time algorithm* $\mathcal{A}$*, the following experiment outputs* 1 *with probability* $1 - \mathrm{negl}(\lambda)$.

1. *Sample* $\mathsf{pp} \leftarrow \mathsf{CM.Setup}(1^\lambda, n)$ *and* $\mathsf{ck} \leftarrow \mathsf{CM.KeyGen}(\mathsf{pp})$.
2. $\mathcal{A}(\mathsf{pp}, \mathsf{ck})$ *outputs*
   - *an index* $i \in [\ell]$*;*
   - *a challenge vector* $(r_1, \ldots, r_{i-1}) \in \mathcal{C}^{i-1}$*;*
   - *distinct challenges* $r_i^{(1)}, \ldots, r_i^{(K)} \in \mathcal{C}$*;*
   - *polynomials* $p_1, \ldots, p_K \in \mathbb{M}[X_{i+1}, \ldots, X_\ell]$*;*
   - *a polynomial* $q(X)$ *in* $\mathbb{C}[X]$ *of degree at most* $d^\star_{\mathsf{ck}}$*; and*
   - *a slackness* $c \in \mathbb{S}_{\mathsf{ck}}$.
3. *The experiment outputs* 1 *if and only if one of the following conditions hold:*
   - *there exists* $j \in [K]$ *such that* $N \cdot \|p_j\|_{\mathbb{M}} > B_{\mathsf{INV}}$*;*
   - *there exists* $j \in [K]$ *such that* p_j *is not* ck*-admissible for* $(r_1, \ldots, r_{i-1}, r_i^{(j)})$*;*
   - *there exists* $j \in [K]$ *such that*

$$\phi_{\mathrm{sc}}\left(q(r_i^{(j)}), \sum_{\omega_{i+1},\ldots,\omega_\ell \in H} f_{\mathsf{CM}}\Big(p_j(\omega_{i+1}, \ldots, \omega_\ell), p_{\mathsf{ck}}(r_1, \ldots, r_{i-1}, r_i^{(j)}, \omega_{i+1}, \ldots, \omega_\ell), c \Big), c \right) = 1 \ ;$$

   - $\mathcal{I}(\mathsf{pp}, \mathsf{ck}, \mathcal{A}(\mathsf{pp}, \mathsf{ck}))$ *outputs* ck*-admissible* $p \in \mathbb{M}[X_i, \ldots, X_\ell]$ *with* $\|p\|_{\mathbb{M}} \leq N \cdot \max_{j \in [K]} \|p_j\|_{\mathbb{M}}$ *such that*

$$\phi_{\mathrm{sc}}\left(\sum_{\omega_i \in H} q(\omega_i), \sum_{\omega_i,\ldots,\omega_\ell \in H} f_{\mathsf{CM}}\Big(p(\omega_i, \ldots, \omega_\ell), p_{\mathsf{ck}}(r_1, \ldots, r_{i-1}, \omega_i, \ldots, \omega_\ell), \xi \cdot c \Big), \xi \cdot c \right) = 1 \ .$$

Lemma 4.13. *Suppose that* CM *is* $(K, B_{\mathsf{INV}}, N, \xi)$*-invertible and* $B' := N^\ell \cdot (B_{\mathsf{C}} \cdot (d_{\mathsf{ck}}+1)^\ell \, \gamma_R^{\ell d_{\mathsf{ck}}} \, \|\mathcal{C}\|_R^{\ell d_{\mathsf{ck}}})$ *satisfies* $B' \leq B_{\mathsf{INV}}$*. Then there exists an efficient algorithm* χ *such that, given an instance* $\mathbb{x} = (\mathsf{CM}, \mathsf{pp}, \mathcal{C}, \mathsf{ck}, \mathsf{cm})$ *and a* K^ℓ*-tree of accepting transcripts for* $\mathbb{x}$*, outputs a witness* $\mathbb{w} = (\mathsf{m}, \rho)$ *for the instance* $\mathbb{x}$ *relative to the relation* $\mathcal{R}_{\mathrm{SC}}(\xi^\ell, B')$*.*

Proof. First we describe the extractor algorithm, then show that it runs in polynomial time, and finally prove that it produces the required output.

The extractor. The inputs and output of the extractor χ are as follows:

- *Input:* An instance $\mathbb{x} = (\mathsf{CM}, \mathsf{pp}, \mathcal{C}, \mathsf{ck}, \mathsf{cm})$ and a K^ℓ-tree of accepting transcripts for Construction 4.5.
- *Output:* A witness $\mathbb{w} = (\mathsf{m}, \rho)$ such that $\|p_{\mathsf{m}}(\underline{X})\|_{\mathbb{M}}$, $\rho \in \mathbb{R}_{\mathsf{ck}}$ and $\mathsf{CM.Open}\left(\mathsf{ck}, \mathsf{m}, \rho, \mathsf{cm}, \xi^\ell\right) = 1$.

For every $i \in [\ell]$ and $(r_1, \ldots, r_{i-1}) \in \mathcal{C}^{i-1}$, let $q_i[r_1, \ldots, r_{i-1}]$ be the polynomial corresponding to the path $(r_1, \ldots, r_{i-1})$ in the transcript tree; this polynomial represents the prover's polynomial in the i-th round of the sumcheck protocol given challenges $(r_1, \ldots, r_{i-1})$. Acceptance in the sumcheck protocol implies that $\sum_{\omega_1 \in H} q_1(\omega_1) = \mathsf{cm} + g_{\mathsf{CM}}(\mathsf{ck}, \rho)$ and, for $i \in \{2, \ldots, \ell\}$, that $\sum_{\omega \in H} q_i[r_1, \ldots, r_{i-1}](\omega) = q_{i-1}[r_1, \ldots, r_{i-2}](r_{i-1})$. Moreover, for every $(r_1, \ldots, r_\ell) \in \mathcal{C}^\ell$, let $w[r_1, \ldots, r_\ell]$ be the opening corresponding to the path $(r_1, \ldots, r_\ell)$ in the transcript tree (i.e., sent by the prover in the sumcheck argument given challenges $(r_1, \ldots, r_\ell)$). Acceptance in the sumcheck argument implies that $\|w[r_1, \ldots, r_\ell]\|_{\mathbb{M}} \leq B_{\mathsf{C}} \cdot (d_{\mathsf{ck}} + 1)^\ell \, \gamma_R^{\ell d_{\mathsf{ck}}} \, \|\mathcal{C}\|_R^{\ell d_{\mathsf{ck}}}$ and $f_{\mathsf{CM}}(w[r_1, \ldots, r_\ell], p_{\mathsf{ck}}(r_1, \ldots, r_\ell), 1) = q_\ell(r_\ell)$ (and that $w[r_1, \ldots, r_\ell]$ is ck-admissible).

By the sumcheck-friendly property, it holds that $\phi_{\mathrm{sc}}(q_\ell(r_\ell), f_{\mathsf{CM}}(w[r_1, \ldots, r_\ell], p_{\mathsf{ck}}(r_1, \ldots, r_\ell), 1), 1) = 1$.

In the iteration for a path $(r_1, \ldots, r_{i-1}) \in \mathcal{C}^{i-1}$ with children $\{r_i^{(j)}\}_{j \in [K]}$, the extractor will use ck-admissible polynomials $\{p[r_1, \ldots, r_{i-1}, r_i^{(j)}]\}_{j \in [K]}$ in $\mathbb{M}[X_{i+1}, \ldots, X_\ell]$ of $\mathbb{M}$-norm at most $N^{\ell-i} \cdot B_{\mathsf{C}} \cdot (d_{\mathsf{ck}} + 1)^\ell \, \gamma_R^{\ell d_{\mathsf{ck}}} \, \|\mathcal{C}\|_R^{\ell d_{\mathsf{ck}}}$ (that satisfy certain properties) to construct a new ck-admissible polynomial $p[r_1, \ldots, r_{i-1}]$ of $\mathbb{M}$-norm at most $N^{\ell-(i-1)} \cdot B_{\mathsf{C}} \cdot (d_{\mathsf{ck}} + 1)^\ell \, \gamma_R^{\ell d_{\mathsf{ck}}} \, \|\mathcal{C}\|_R^{\ell d_{\mathsf{ck}}}$ (that satisfies certain properties). We set the initial polynomials $\{p[r_1, \ldots, r_\ell]\}_{(r_1, \ldots, r_\ell) \in \mathcal{C}^\ell}$ to be the constant polynomials corresponding to the opening values $\{w[r_1, \ldots, r_\ell]\}_{(r_1, \ldots, r_\ell) \in \mathcal{C}^\ell}$.

The extractor χ works as follows.

For $i = \ell, \ldots, 1$:

- For every path $(r_1, \ldots, r_{i-1}) \in \mathcal{C}^{i-1}$ in the transcript tree with children $\{r_i^{(j)}\}_{j \in [K]}$:
  - For each $j \in [K]$, we have that

$$\phi_{\mathrm{sc}}\left(q_i[r_1, \ldots, r_{i-1}](r_i^{(j)}), \sum_{\underline{\omega} \in H^{\ell-i}} f_{\mathsf{CM}}\Big(p[r_1, \ldots, r_{i-1}, r_i^{(j)}](\underline{\omega}), p_{\mathsf{ck}}(r_1, \ldots, r_{i-1}, r_i^{(j)}, \underline{\omega}), \xi^{\ell-i}\Big), \xi^{\ell-i}\right) = 1 \ ,$$

  - Run the inverter $\mathcal{I}$ on input $(\mathsf{pp}, \mathsf{ck})$ and
    - the index $i \in [\ell]$,
    - the challenge vector $(r_1, \ldots, r_{i-1}) \in \mathcal{C}^{i-1}$,
    - the distinct challenges $\{r_i^{(j)}\}_{j \in [K]}$ in $\mathcal{C}$,
    - the polynomials $\{p[r_1, \ldots, r_{i-1}, r_i^{(j)}]\}_{j \in [K]}$,
    - the polynomial $q_i[r_1, \ldots, r_{i-1}]$,
    - the slackness $\xi^{\ell-i}$

to produce new polynomial $p[r_1, \ldots, r_{i-1}] \in \mathbb{M}[X_i, \ldots, X_\ell]$ with norm at most $N^{\ell-(i-1)} \cdot B_{\mathsf{C}} \cdot (d_{\mathsf{ck}} + 1)^\ell \, \gamma_R^{\ell d_{\mathsf{ck}}} \, \|\mathcal{C}\|_R^{\ell d_{\mathsf{ck}}}$ such that

$$\phi_{\mathrm{sc}}\left(\sum_{\omega \in H} q_i[r_1, \ldots, r_{i-1}](\omega), \sum_{\underline{\omega} \in H^{\ell-(i-1)}} f_{\mathsf{CM}}\Big(p[r_1, \ldots, r_{i-1}](\underline{\omega}), p_{\mathsf{ck}}(r_1, \ldots, r_{i-1}, \underline{\omega}), \xi^{\ell-(i-1)}\Big), \xi^{\ell-(i-1)}\right) = 1 \ .$$

If the inverter fails to produce a valid output, then output $\perp$.

The inverter preserves admissibility and increases the norms of extracted openings by N on each of its ℓ invocations. After the final step, the extractor has found a ck-admissible polynomial $p \in \mathbb{M}[X_1, \ldots, X_\ell]$ with norm at most $B' = N^\ell \cdot B_{\mathsf{C}} \cdot (d_{\mathsf{ck}} + 1)^\ell \, \gamma_R^{\ell d_{\mathsf{ck}}} \, \|\mathcal{C}\|_R^{\ell d_{\mathsf{ck}}}$ such that

$$\phi_{\mathrm{sc}}\left(\sum_{\omega \in H} q_1(\omega), \sum_{\underline{\omega} \in H^\ell} f_{\mathsf{CM}}(p(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}), \xi^\ell), \xi^\ell\right) = \phi_{\mathrm{sc}}\left(\mathsf{cm} - g_{\mathsf{CM}}(\mathsf{ck}, \rho), \sum_{\underline{\omega} \in H^\ell} f_{\mathsf{CM}}(p(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}), \xi^\ell), \xi^\ell\right) = 1 \ .$$

The extractor then computes the message m encoded by p. By the sumcheck-friendly property of CM, it follows that $\mathsf{CM.Open}\left(\mathsf{ck}, \mathsf{m}, \rho, \xi^\ell\right) = 1$, which means that the extractor has found a witness for the instance $\mathbb{x} = (\mathsf{CM}, \mathsf{pp}, \mathcal{C}, \mathsf{ck}, \mathsf{cm})$ relative to the relation $\mathcal{R}_{\mathrm{SC}}(\xi^\ell, B')$.

Running time. Let $T(i)$ denote the running time of the inverter $\mathcal{I}$ on inputs with parameter $i \in [\ell]$. At step i, the extractor runs the inverter $\mathcal{I}$ a total of K^{i-1} times with parameter i. Therefore, the running time of the extractor is $\sum_{i \in [\ell]} K^{i-1} T(i)$.

Success probability. Let $\epsilon(i)$ denote the failure probability of the inverter $\mathcal{I}$ on properly-distributed inputs with parameter i. If any execution of the inverter fails for any i, then the extractor will terminate in failure rather than producing an opening of cm. Therefore, by a union bound, the failure probability of the extractor is at most $\sum_{i \in [\ell]} K^{i-1} \epsilon(i)$. □

5 Instantiations of sumcheck-friendly commitments

We describe instantiations of sumcheck-friendly commitments suitable for a sumcheck argument (Section 4). In Section 5.1 we define bilinear modules and describe properties that we use to construct sumcheck-friendly commitments. Then we provide examples of sumcheck-friendly commitments:
- in Section 5.2 we describe a generalization of the Pedersen commitment;
- in Section 5.3 we describe a linear-function commitment;
- in Section 5.4 we describe a scalar-product commitment; and
- in Section 5.5 we describe a compressed version of the scalar-product commitment.

Finally, in Section 5.6 we describe how to instantiate the bilinear modules in different cryptographic settings.

Remark 5.1 (simplifying assumption)**.** For simplicity, throughout this section *we restrict our attention to working with rings R where 2 is not a zero-divisor*. This is because we use Lemma 3.6 with a multiplicative subgroup H with $|H| = 2$, which introduces factors of 2 into various algebraic expressions. One way to avoid this is to use multiplicative subgroups H with $|H| = 3$ (or other invertible constants), and generalizations of Lemma 3.6 and Definition 4.1 to suitable weighted sums. One might work with ring extensions of rings of interest as in [Abs+20; GNS21], to ensure the existence of a suitable subgroup. Another way, in certain settings, is to use an *additive* subgroup H and rely on Lemma 3.7 instead of Lemma 3.6.

5.1 Bilinear modules

We define bilinear modules and describe properties that we use to construct sumcheck-friendly commitments.

Definition 5.2. *A* **bilinear module** *is a tuple $\mathcal{M} = (R, M_{\mathrm{L}}, M_{\mathrm{R}}, M_{\mathrm{T}}, e)$ where R is a ring, $M_{\mathrm{L}}, M_{\mathrm{R}}, M_{\mathrm{T}}$ are R-modules, and $e\colon M_{\mathrm{L}} \times M_{\mathrm{R}} \to M_{\mathrm{T}}$ is a non-degenerate bilinear map; moreover, R and M_{L} are equipped with norms $\|\cdot\|_R$ and $\|\cdot\|_{M_{\mathrm{L}}}$. We use arithmetic notation as a shorthand for e: for $a \in M_{\mathrm{L}}$ and $\mathsf{G} \in M_{\mathrm{R}}$, "$a \cdot \mathsf{G}$" denotes $e(a, \mathsf{G}) \in M_{\mathrm{T}}$; similarly, for $\underline{a} \in M_{\mathrm{L}}^n$ and $\in M_{\mathrm{R}}^n$, "$\langle \underline{a}, \underline{\mathsf{G}} \rangle$" denotes $\sum_{i\in[n]} e(a_i, \mathsf{G}_i) \in M_{\mathrm{T}}$.*

Definition 5.3. *Let $\mathcal{M} = (R, M_{\mathrm{L}}, M_{\mathrm{R}}, M_{\mathrm{T}}, e)$ be bilinear module. An integer $h \in \mathbb{N}$ is $(\mathcal{M}, \mathcal{U}_{M_{\mathrm{L}}}, \mathcal{U}_{M_{\mathrm{R}}}, \epsilon)$-*
hiding*, where $\mathcal{U}_{M_{\mathrm{L}}}, \mathcal{U}_{M_{\mathrm{R}}}$ are distributions over $M_{\mathrm{L}}^h, M_{\mathrm{R}}^{n+h}$ respectively, if for every $\underline{a} \in M_{\mathrm{L}}^n$ the following random variables are ϵ-close:*

$$\left\{ (\underline{\mathsf{G}}, \langle \underline{a}, \underline{\mathsf{G}}_0 \rangle + \langle \underline{r}, \underline{\mathsf{G}}_1 \rangle) \middle| \begin{array}{c} \underline{\mathsf{G}} \leftarrow \mathcal{U}_{M_{\mathrm{R}}} \\ \underline{r} \leftarrow \mathcal{U}_{M_{\mathrm{L}}} \end{array} \right\} \text{ and } \left\{ (\underline{\mathsf{G}}, \langle \underline{r}, \underline{\mathsf{G}}_1 \rangle) \middle| \begin{array}{c} \underline{\mathsf{G}} \leftarrow \mathcal{U}_{M_{\mathrm{R}}} \\ \underline{r} \leftarrow \mathcal{U}_{M_{\mathrm{L}}} \end{array} \right\} .$$

Definition 5.4. *A* **bilinear-module generator** *is a tuple $\mathsf{BM} = (\mathsf{Setup}, \mathsf{KeyGen})$ with the following syntax:*
- $\mathsf{BM.Setup}$*, given 1^λ and $n \in \mathbb{N}$, outputs a bilinear module $\mathcal{M}$, integer $h \in \mathbb{N}$, and auxiliary string aux;*
- $\mathsf{BM.KeyGen}$*, given $(\mathcal{M}, h, \mathsf{aux})$, outputs a vector in M_{R}^{n+h}.*

We assume without loss of generality that the parameters λ and n are part of aux*.*

Definition 5.5. BM *is* **scalar-product compatible** *if M_{L} output by* $\mathsf{BM.Setup}$ *is itself a ring.*

The bilinear-module generators that we consider output auxiliary strings that contain several pieces of information: $\mathsf{aux} = (B_{\mathsf{BRA}}, \mathcal{U}_{M_{\mathrm{L}}}, \mathcal{C}, \xi, N, B_{\mathsf{C}})$ where $B_{\mathsf{BRA}} \in \mathbb{Z}$, $\mathcal{U}_{M_{\mathrm{L}}}$ is a distribution over M_{L}^h, $\mathcal{C} \subseteq R$, $\xi \in R$, $N \in \mathbb{Z}$, and $B_{\mathsf{C}} \in \mathbb{Z}$ with $B_{\mathsf{C}} \leq B_{\mathsf{BRA}}$.

Definition 5.6. *A bilinear-module generator* BM *is* **secure** *if it satisfies the following properties.*

- BM *satisfies the* **bilinear relation assumption (BRA)***: for every* $n \in \mathbb{N}$*, algorithm* Check*, and polynomial-size adversary* $\mathcal{A}$*,*

$$\Pr\left[\begin{array}{c|c} \mathsf{Check}(\mathcal{M}, h, \mathsf{aux}) = 1 & (\mathcal{M}, h, \mathsf{aux}) \leftarrow \mathsf{BM.Setup}(1^\lambda, n) \\ \underline{a} \in M_{\mathrm{L}}^{n+h}(B_{\mathsf{BRA}}) \setminus \{0^{n+h}\} & \underline{\mathsf{G}} \leftarrow \mathsf{BM.KeyGen}(\mathcal{M}, h, \mathsf{aux}) \\ \langle \underline{a}, \underline{\mathsf{G}} \rangle = 0 & \underline{a} \leftarrow \mathcal{A}(\mathcal{M}, h, \mathsf{aux}, \underline{\mathsf{G}}) \end{array}\right] = \mathrm{negl}(\lambda) \ ;$$

- h *is* $(\mathcal{M}, \mathcal{U}_{M_{\mathrm{L}}}, \mathcal{U}_{M_{\mathrm{R}}}, \mathrm{negl}(\lambda))$*-hiding, where* $\mathcal{U}_{M_{\mathrm{L}}}$ *is in* aux *and* $\mathcal{U}_{M_{\mathrm{R}}}$ *is the distribution of* $\mathsf{BM.KeyGen}$ *(Definition 5.3);*

- $(\mathcal{C}, \xi, N)$ *in* aux *are pseudoinverse parameters for* (R, M_{T}) *output by* $\mathsf{BM.Setup}$ *(Definition 3.13).*

We use secure bilinear-module generators to define invertible sumcheck-friendly commitments. In Section 6 and Section 7, we use bilinear-module generators with additional properties, which we call protocol-friendly bilinear-module generators.

Definition 5.7. BM *is* **masking-friendly** *if* aux *output by* BM *contains* $\kappa \in \mathbb{N}$ *such that for every* $B \in \mathbb{Z}$ *with* $B_{\mathsf{C}} \leq B \leq B_{\mathsf{BRA}}/\kappa$ *and* $\underline{a} \in M_{\mathrm{L}}^n(B)$ *the procedure "sample* $\underline{b} \leftarrow M_{\mathrm{L}}^n(\kappa B)$ *and output* $\underline{a} + \underline{b}$ *if* $\|\underline{a} + \underline{b}\|_{M_{\mathrm{L}}} \leq (\kappa - 1)B$ *and* $\perp$ *otherwise" has the following properties:*
- *The procedure outputs* $\perp$ *with probability at most* n/κ*.*
- *Conditioned on not outputting* $\perp$*, the distribution of* $\underline{a} + \underline{b}$ *is uniform in* $M_{\mathrm{L}}^n((\kappa - 1)B)$*.*

Definition 5.8. BM *is* **quotient-friendly** *if* M_{L} *output by* $\mathsf{BM.Setup}$ *contains an* R*-submodule* $I \subseteq M_{\mathrm{L}}$ *such that* $(\mathcal{C}, \xi, N)$ *in* aux *output by* $\mathsf{BM.Setup}$ *are pseudoinverse parameters for* (R, M_{T}) *such that multiplication by* ξ *is invertible in* M_{L}/I*. (Definition 3.13).*

Definition 5.9. *A bilinear-module generator* BM *is* **protocol-friendly** *if it satisfies the following properties.*

- BM *is secure for any* $\mathcal{U}_{M_{\mathrm{L}}}$ *that is the uniform distribution over* $M_{\mathrm{L}}^h(B)$ *with* $B_{\mathsf{C}} \leq B \leq B_{\mathsf{BRA}}$*;*

- BM *is scalar-product compatible (Definition 5.5);*

- BM *is masking-friendly (Definition 5.7);*

- BM *is quotient-friendly (Definition 5.8).*

5.2 Pedersen commitment

The Pedersen commitment scheme is an example of a sumcheck-friendly commitment scheme.

Definition 5.10. *Let* $\mathsf{BM} = (\mathsf{Setup}, \mathsf{KeyGen})$ *be a bilinear-module generator. The* **(generalized) Pedersen commitment scheme** *is defined via the following algorithms.*

- $\mathsf{Ped.Setup}(1^\lambda, n)$: *sample* $(\mathcal{M}, h, \mathsf{aux}) \leftarrow \mathsf{BM.Setup}(1^\lambda, n)$ *and output* $\mathsf{pp} := (\mathcal{M}, h, \mathsf{aux})$.
- $\mathsf{Ped.KeyGen}(\mathsf{pp})$: *sample* $\mathsf{ck} \leftarrow \mathsf{BM.KeyGen}(\mathcal{M}, h, \mathsf{aux})$ *and output* $\mathsf{ck} = (\mathsf{ck}_0, \mathsf{ck}_1) \in M_{\mathrm{R}}^n \times M_{\mathrm{R}}^h$.
- $\mathsf{Ped.Commit}(\mathsf{ck}, \mathsf{m}; \rho)$: *given* $\mathsf{m} \in M_{\mathrm{L}}^n(B_{\mathsf{C}})$ *and* $\rho \leftarrow \mathcal{U}_{M_{\mathrm{L}}}$*, output* $\mathsf{cm} := \langle \mathsf{m}, \mathsf{ck}_0 \rangle + \langle \rho, \mathsf{ck}_1 \rangle \in M_{\mathrm{T}}$.
- $\mathsf{Ped.Open}(\mathsf{ck}, \mathsf{m}, \rho, \mathsf{cm}, c)$: *check that* $\mathsf{m} \in M_{\mathrm{L}}^n(B_{\mathsf{BRA}})$*,* $\rho \in M_{\mathrm{L}}^h(B_{\mathsf{BRA}})$*,* $c \cdot \mathsf{cm} = \langle \mathsf{m}, \mathsf{ck}_0 \rangle + c \cdot \langle \rho, \mathsf{ck}_1 \rangle$.

Lemma 5.11. *If* BM *is secure then* Ped *is a sumcheck-friendly invertible commitment scheme. In more detail:*

1. if BM *satisfies the BRA then* Ped *is binding;*

2. *if h is $(\mathcal{M}, \mathcal{U}_{M_{\mathrm{L}}}, \mathcal{U}_{M_{\mathrm{R}}}, \mathrm{negl}(\lambda))$-hiding then* Ped *is computationally hiding;*

3. Ped *is sumcheck-friendly;*

4. *if $(\mathcal{C}, \xi, N)$ are pseudoinverse parameters for (R, M_{T}) then, for every $B \in \mathbb{N}$ (regardless of B_{BRA} in* aux*),* Ped *is $(3, B, \xi^3, N_{\mathsf{Ped}})$-invertible for*

$$N_{\mathsf{Ped}} := 6\gamma_R^2 N^3 \iota(\mathcal{C}, 3) \|\mathcal{C}\|_R \ . \tag{5}$$

The proofs of Items 1 and 2 follow directly from the definition of the BRA and the hiding constant of BM. We now prove the other two items.

Proof of Item 3. Let $H := \{-1, 1\}$ and $\ell := \log n$. We define p_{m} and p_{ck} to be the multilinear polynomials corresponding to the message $\mathsf{m} \in M_{\mathrm{L}}^n$ and to the subkey $\mathsf{ck}_0 \in M_{\mathrm{R}}^n$ in the key ck respectively (see Definition 3.5). We also define:

- $f_{\mathsf{CM}} \colon M_{\mathrm{L}} \times M_{\mathrm{R}} \times R \to M_{\mathrm{T}}$ as $f_{\mathsf{CM}}(a, \mathsf{G}, c) := 2^{-\ell} a \cdot \mathsf{G}$;
- $g_{\mathsf{CM}} \colon M_{\mathrm{R}}^{n+h} \times M_{\mathrm{L}}^h \to M_{\mathrm{T}}$ as $g_{\mathsf{CM}}(\mathsf{ck}, \rho) := \langle \rho, \mathsf{ck}_1 \rangle$;
- $\phi_{\mathrm{sc}} \colon M_{\mathrm{T}} \times M_{\mathrm{T}} \times R \to \{0, 1\}$ as $\phi_{\mathrm{sc}}(\mathsf{cm}, \mathsf{cm}', c) := \mathbb{I}[c \cdot \mathsf{cm} \stackrel{?}{=} \mathsf{cm}']$, which checks equality when $c = 1$.
- $\alpha_{\mathrm{sc}} \equiv 1$. (For every $i \in \{0, 1, \ldots, \ell\}$, $p \in M_{\mathrm{L}}[X_{i+1}, \ldots, X_\ell]$, and $(r_1, \ldots, r_i) \in \mathcal{C}^i$, there exists a message $\mathsf{m} \in M_{\mathrm{L}}^n$ such that $p(X_{i+1}, \ldots, X_\ell) = p_{\mathsf{m}}(r_1, \ldots, r_i, X_{i+1}, \ldots, X_\ell)$.)

For a key $\mathsf{ck} = (\mathsf{ck}_0, \mathsf{ck}_1) \in M_{\mathrm{R}}^n \times M_{\mathrm{R}}^h$, message $\mathsf{m} \in M_{\mathrm{L}}^n$, randomness $\rho \in M_{\mathrm{L}}^h$, and slackness $c \in R$, it holds that

$$\sum_{\underline{\omega} \in \{-1,1\}^\ell} f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}), c) = \sum_{\underline{\omega} \in \{-1,1\}^\ell} 2^{-\ell} p_{\mathsf{m}}(\underline{\omega}) \cdot p_{\mathsf{ck}}(\underline{\omega}) \ .$$

Since p_{m} and p_{ck} are multilinear polynomials, their product has individual degree at most 2 in each variable. By Lemma 3.6, $\sum_{\underline{\omega} \in \{-1,1\}^\ell} 2^{-\ell} p_{\mathsf{m}}(\underline{\omega}) \cdot p_{\mathsf{ck}}(\underline{\omega})$ is the sum of the coefficients of $X_1^{i_1} \cdots X_\ell^{i_\ell}$ such that $i_1, \ldots, i_\ell \equiv 0 \bmod 2$. Note that for $i, j \in \{0, 1\}$, $i + j \equiv 0 \bmod 2$ if and only if $i = j$. This means that each of the coefficients of $p_{\underline{a}}(\underline{X}) \cdot p_{\mathrm{rv}(\underline{\mathsf{G}})}(\underline{X})$ with $i_1, \ldots, i_\ell \equiv 0 \bmod 2$ arise from a multiplication of the monomials in the terms $\mathsf{m}_{\underline{i}} X_1^{i_1} \cdots X_\ell^{i_\ell}$ and $\mathsf{ck}_{\underline{i}} X_1^{i_1} \cdots X_\ell^{i_\ell}$. This implies that

$$\sum_{\underline{\omega} \in \{-1,1\}^\ell} f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}), c) = \sum_{\underline{\omega} \in \{-1,1\}^\ell} 2^{-\ell} p_{\mathsf{m}}(\underline{\omega}) \cdot p_{\mathsf{ck}}(\underline{\omega}) = \langle \mathsf{m}, \mathsf{ck}_0 \rangle \ .$$

Therefore

$$\mathsf{Ped.Commit}(\mathsf{ck}, \mathsf{m}; \rho) = \langle \mathsf{m}, \mathsf{ck}_0 \rangle + \langle \rho, \mathsf{ck}_1 \rangle = \sum_{\underline{\omega} \in \{-1,1\}^\ell} f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}), 1) + g_{\mathsf{CM}}(\mathsf{ck}_0, \mathsf{ck}_1, \rho) \ .$$

We have $\mathsf{Ped.Open}(\mathsf{ck}, \mathsf{m}, \rho, \mathsf{cm}, c) = 1$ if and only if $c \cdot \mathsf{cm} = \langle \mathsf{m}, \mathsf{ck}_0 \rangle + c \cdot \langle \rho, \mathsf{ck}_1 \rangle$, which is true if and only if $c \cdot (\mathsf{cm} - g_{\mathsf{CM}}(\mathsf{ck}_0, \mathsf{ck}_1, \rho)) = \sum_{\underline{\omega} \in \{-1,1\}^\ell} f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}), c)$, implying the required property of ϕ_{sc}. Finally, we have $d_{\mathsf{ck}} = 1$ and $d^\star_{\mathsf{ck}} = 2$. □

Proof of Item 4. Let $\mathsf{ck} \in M_{\mathrm{R}}^{n+h}$ be the commitment key sampled in the first step of the invertibility definition. Also, fix an output of $\mathcal{A}$: an index $i \in [\ell]$; a challenge vector $(r_1, \ldots, r_{i-1}) \in \mathcal{C}^{i-1}$; distinct challenges $r_i^{(1)}, r_i^{(2)}, r_i^{(3)} \in \mathcal{C}$; polynomials $p_1, p_2, p_3 \in M_{\mathrm{L}}[X_{i+1}, \ldots, X_\ell]$; a polynomial $q(X) = q_0 + q_1 X + q_2 X^2$ in $M_{\mathrm{T}}[X]$; and a slackness $c \in R$.

Moreover, suppose that $\mathcal{A}$'s outputs satisfy the experiment's checks (or else the experiment just outputs 1): for every $j \in [3]$ it holds that $N_{\mathsf{Ped}} \cdot \|p_j\|_{M_{\mathrm{L}}} \leq B$, p_j is ck-admissible for $(r_1, \ldots, r_{i-1}, r_i^{(j)})$, and

$$q(r_i^{(j)}) = \sum_{\omega_{i+1},\ldots,\omega_\ell \in H} f_{\mathsf{CM}}\Big(p_j(\omega_{i+1}, \ldots, \omega_\ell), p_{\mathsf{ck}}(r_1, \ldots, r_{i-1}, r_i^{(j)}, \omega_{i+1}, \ldots, \omega_\ell), c\Big) \ .$$

We have to specify an inverter $\mathcal{I}$ that, given pp, ck, and $\mathcal{A}$'s outputs, outputs a ck-admissible $p \in M_{\mathrm{L}}[X_i, \ldots, X_\ell]$ with $\|p\|_{M_{\mathrm{L}}} \leq N_{\mathsf{Ped}} \cdot \max_{j\in[3]} \|p_j\|_{M_{\mathrm{L}}}$ such that

$$\sum_{\omega_i \in H} q(\omega_i) = \sum_{\omega_i,\ldots,\omega_\ell \in H} f_{\mathsf{CM}}\Big(p(\omega_i, \ldots, \omega_\ell), p_{\mathsf{ck}}(r_1, \ldots, r_{i-1}, \omega_i, \ldots, \omega_\ell), \xi \cdot c\Big) \ .$$

Denoting by $\underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0]$ and $\underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1]$ the coefficients in $p_{\mathsf{ck}}(r_1, \ldots, r_{i-1}, X_i, \ldots, X_\ell)$ for monomials without X_i and with X_i respectively, for every $j \in [3]$ it holds that:

$$\begin{aligned}
&c \cdot (q_0 + q_1 r_i^{(j)} + q_2 {r_i^{(j)}}^2) \\
&= c \cdot q(r_i^{(j)}) \\
&= \sum_{\underline{\omega} \in \{-1,1\}^{\ell-i}} 2^{-\ell} p_j(\underline{\omega}) \cdot p_{\mathsf{ck}}(r_1, \ldots, r_{i-1}, r_i^{(j)}, \underline{\omega}) \\
&= 2^{-i} \langle \underline{p_j}, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0] + r_i^{(j)} \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1] \rangle \ ,
\end{aligned}$$

where $\underline{p_j}$ are the coefficients of p_j, and the last equality follows from Lemma 3.6.

Define the following Vandermonde matrix and its adjugate:

$$V := \begin{bmatrix} 1 & r_i^{(1)} & {r_i^{(1)}}^2 \\ 1 & r_i^{(2)} & {r_i^{(2)}}^2 \\ 1 & r_i^{(3)} & {r_i^{(3)}}^2 \end{bmatrix} \quad \text{and} \quad \mathrm{adj}(V) := \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \ .$$

We can find multiples of q_0, q_1, q_2 in terms of $\underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0]$ and $\underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1]$ by computing

$$\begin{aligned}
\begin{bmatrix} \det(V) c \cdot q_0 \\ \det(V) c \cdot q_1 \\ \det(V) c \cdot q_2 \end{bmatrix} &:= \mathrm{adj}(V) \cdot \begin{bmatrix} 2^{-i} \langle \underline{p_1}, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0] + r_i^{(1)} \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1] \rangle \\ 2^{-i} \langle \underline{p_2}, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0] + r_i^{(2)} \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1] \rangle \\ 2^{-i} \langle \underline{p_3}, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0] + r_i^{(3)} \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1] \rangle \end{bmatrix} \\
&= 2^{-i} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \cdot \begin{bmatrix} \langle \underline{p_1}, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0] \rangle + \langle r_i^{(1)} \underline{p_1}, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1] \rangle \\ \langle \underline{p_2}, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0] \rangle + \langle r_i^{(2)} \underline{p_2}, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1] \rangle \\ \langle \underline{p_3}, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0] \rangle + \langle r_i^{(3)} \underline{p_3}, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1] \rangle \end{bmatrix} \\
&= 2^{-i} \begin{bmatrix} \langle \sum_{j=1}^3 a_{1j} \underline{p_j}, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0] \rangle + \langle \sum_{j=1}^3 a_{1j} r_i^{(j)} \underline{p_j}, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1] \rangle \\ \langle \sum_{j=1}^3 a_{2j} \underline{p_j}, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0] \rangle + \langle \sum_{j=1}^3 a_{2j} r_i^{(j)} \underline{p_j}, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1] \rangle \\ \langle \sum_{j=1}^3 a_{3j} \underline{p_j}, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0] \rangle + \langle \sum_{j=1}^3 a_{3j} r_i^{(j)} \underline{p_j}, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1] \rangle \end{bmatrix} \\
&= 2^{-i} \begin{bmatrix} \langle \sum_{j=1}^3 (a_{1j} \underline{p_j}, a_{1j} r_i^{(j)} \underline{p_j}), (\underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0], \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1]) \rangle \\ \langle \sum_{j=1}^3 (a_{2j} \underline{p_j}, a_{2j} r_i^{(j)} \underline{p_j}), (\underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0], \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1]) \rangle \\ \langle \sum_{j=1}^3 (a_{3j} \underline{p_j}, a_{3j} r_i^{(j)} \underline{p_j}), (\underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0], \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1]) \rangle \end{bmatrix} \ .
\end{aligned}$$

Note that, for each $k \in [3]$, it holds that

$$\|\sum_{j=1}^{3}(a_{kj}\underline{p}_j, a_{kj}r_i^{(j)}\underline{p}_j)\|_{M_{\mathrm{L}}} \leq 3\gamma_R^2\iota(\mathcal{C},3)\|\mathcal{C}\|_R \max_{j\in[3]}\|p_j\|_{M_{\mathrm{L}}}$$

because the sum contains three terms, and each of these terms is a multiplication of at most three elements: an entry of $\mathrm{adj}(V)$, an element of $\mathcal{C}$, and a coordinate of $\underline{p}_j$ (for $j \in [3]$). See Definition 3.12 for the definition of $\iota(\mathcal{C},3)$, and Definition 3.4 for the definition of $\|\mathcal{C}\|_R$; the term γ_R^2 arises from the expansion factor of norms when multiplying (in this case at most three) elements (see Remark 3.3).

In other words, we can find a quadratic polynomial $\pi \in M_{\mathrm{L}}^{2^{\ell-i+1}}[X]$ such that

$$\det(V)c \cdot q(X) = 2^{-i}\langle \pi(X), (\underline{p}_{\mathsf{ck}}[r_1,\ldots,r_{i-1};0], \underline{p}_{\mathsf{ck}}[r_1,\ldots,r_{i-1};1])\rangle$$

where each coefficient of π has M_{L}-norm at most $3\gamma_R^2\iota(\mathcal{C},3)\|\mathcal{C}\|_R \max_{j\in[3]}\|p_j\|_{M_{\mathrm{L}}}$.

Note that $\det(V) = (r_i^{(1)} - r_i^{(2)})(r_i^{(2)} - r_i^{(3)})(r_i^{(3)} - r_i^{(1)})$, and the constants (ξ, N) satisfy Definition 3.13. So, we can find $a_0, a_1, a_2, b_0, b_1, b_2 \in M_{\mathrm{L}}^{2^{\ell-i}}$ with M_{L}-norm at most $N^3(3\gamma_R^2\iota(\mathcal{C},3)\|\mathcal{C}\|_R \max_{j\in[3]}\|p_j\|_{M_{\mathrm{L}}}) = 3\gamma_R^2 N^3\iota(\mathcal{C},3)\|\mathcal{C}\|_R \max_{j\in[3]}\|p_j\|_{M_{\mathrm{L}}}$ such that

$$\xi^3 c\cdot q(X) = 2^{-i}\langle a_0+a_1X+a_2X^2, \underline{p}_{\mathsf{ck}}[r_1,\ldots,r_{i-1};0]\rangle + 2^{-i}\langle b_0+b_1X+b_2X^2, \underline{p}_{\mathsf{ck}}[r_1,\ldots,r_{i-1};1]\rangle \ . \quad (6)$$

By summing the above equation over $H = \{-1, 1\}$, we obtain that

$$\begin{aligned}
&\xi^3 c \cdot \big(q(1) + q(-1)\big) \\
&= 2^{-i}\langle a_0 + a_1 + a_2, \underline{p}_{\mathsf{ck}}[r_1,\ldots,r_{i-1};0]\rangle + 2^{-i}\langle b_0 + b_1 + b_2, \underline{p}_{\mathsf{ck}}[r_1,\ldots,r_{i-1};1]\rangle \\
&\quad + 2^{-i}\langle a_0 - a_1 + a_2, \underline{p}_{\mathsf{ck}}[r_1,\ldots,r_{i-1};0]\rangle + 2^{-i}\langle b_0 - b_1 + b_2, \underline{p}_{\mathsf{ck}}[r_1,\ldots,r_{i-1};1]\rangle \\
&= 2^{-i}\langle 2a_0 + 2a_2, \underline{p}_{\mathsf{ck}}[r_1,\ldots,r_{i-1};0]\rangle + 2^{-i}\langle 2a_0 + 2a_2, \underline{p}_{\mathsf{ck}}[r_1,\ldots,r_{i-1};1]\rangle \\
&= 2^{-i}\Big\langle \Big(2a_0 + 2a_2, 2b_0 + 2b_2\Big), \Big(\underline{p}_{\mathsf{ck}}[r_1,\ldots,r_{i-1};0], \underline{p}_{\mathsf{ck}}[r_1,\ldots,r_{i-1};1]\Big)\Big\rangle \\
&= 2^{-i+1}\Big\langle \Big(a_0 + a_2, b_0 + b_2\Big), \Big(\underline{p}_{\mathsf{ck}}[r_1,\ldots,r_{i-1};0], \underline{p}_{\mathsf{ck}}[r_1,\ldots,r_{i-1};1]\Big)\Big\rangle \ .
\end{aligned}$$

Now consider the vector $\underline{v}_i := (a_0 + a_2, b_0 + b_2) \in M_{\mathrm{L}}^{2^{\ell-i+1}}$, and its corresponding multilinear polynomial $p_{\underline{v}_i} \in M_{\mathrm{L}}[X_i,\ldots,X_\ell]$ obtained according to Definition 3.5. Note that $\underline{v}_i$, and thus $p_{\underline{v}_i}$, has M_{L}-norm at most

$$6\gamma_R^2 N^3\iota(\mathcal{C},3)\|\mathcal{C}\|_R \max_{j\in[3]}\|p_j\|_{M_{\mathrm{L}}} = N_{\mathsf{Ped}} \max_{j\in[3]}\|p_j\|_{M_{\mathrm{L}}}$$

and satisfies the following equation:

$$\xi^3 c \cdot (q(1) + q(-1)) = \sum_{\omega_i,\ldots,\omega_\ell\in\{-1,1\}} f_{\mathsf{CM}}\Big(p_{\underline{v}_i}(\omega_i,\ldots,\omega_\ell), p_{\mathsf{ck}}(r_1,\ldots,r_{i-1},\omega_i,\ldots,\omega_\ell), \xi^3\cdot c\Big) \ .$$

Moreover, $p_{\underline{v}_i}$ is ck-admissible and all the derivations to obtain $p_{\underline{v}_i}$, the desired polynomial, are efficient. □

5.3 Linear-function commitment

We define a commitment scheme that includes the scalar-product of two parts of a message, where one part is public and the other part is in a Pedersen commitment. This commitment scheme can be viewed as a *linear-function commitment*, where the linear function is the committed part of message, the "query" to the linear function is the public part of the message, and query answer is the scalar product of the two; the committed part of the message can be created in an online phase and re-used across different queries.

Definition 5.12. *Let* $\mathsf{BM} = (\mathsf{Setup}, \mathsf{KeyGen})$ *be a bilinear-module generator. The* **linear-function commitment scheme** *is defined via the following algorithms.*

- $\mathsf{LF.Setup}(1^\lambda, n)$: *sample* $(\mathcal{M}, h, \mathsf{aux}) \leftarrow \mathsf{BM.Setup}(1^\lambda, n)$ *and output* $\mathsf{pp} := (\mathcal{M}, h, \mathsf{aux})$.
- $\mathsf{LF.KeyGen}(\mathsf{pp})$: *sample* $\mathsf{ck} \leftarrow \mathsf{BM.KeyGen}(\mathcal{M}, h, \mathsf{aux})$ *and output* $\mathsf{ck} = (\mathsf{ck}_0, \mathsf{ck}_1) \in M_{\mathrm{R}}^n \times M_{\mathrm{R}}^h$.
- $\mathsf{LF.Commit}(\mathsf{ck}, \mathsf{m}; \rho)$: *given* $\mathsf{m} = (\mathsf{mP}, \mathsf{mS}) \in R^n \times M_{\mathrm{L}}^n(B_{\mathsf{C}})$ *and* $\rho \leftarrow \mathcal{U}_{M_{\mathrm{L}}}$ *, output*

$$\mathsf{cm} := \Big(\mathsf{mP}, \langle \mathsf{mS}, \mathsf{ck}_0\rangle + \langle \rho, \mathsf{ck}_1\rangle, \langle \mathsf{mP}, \mathsf{mS}\rangle\Big) \in R \times M_{\mathrm{T}} \times M_{\mathrm{L}} \ .$$

- $\mathsf{LF.Open}(\mathsf{ck}, \mathsf{m}, \rho, \mathsf{cm}, c)$: *check that* $\mathsf{m} = (\mathsf{mP}, \mathsf{mS}) \in R^n \times M_{\mathrm{L}}^n(B_{\mathsf{BRA}})$*, and* $\rho \in M_{\mathrm{L}}^h(B_{\mathsf{BRA}})$*,* $c \cdot \mathsf{cm} = (c \cdot \mathsf{mP}, \langle \mathsf{mS}, \mathsf{ck}_0\rangle + c\langle \rho, \mathsf{ck}_1\rangle, \langle \mathsf{mP}, \mathsf{mS}\rangle)$.

Lemma 5.13. *If* BM *is secure then* LF *is a sumcheck-friendly invertible commitment scheme. In more detail:*

1. *if* BM *satisfies the BRA then* LF *is binding;*
2. LF *is sumcheck-friendly;*
3. *if* $(\mathcal{C}, \xi, N)$ *are pseudoinverse parameters for* (R, M_{T}) *then, for every* $B \in \mathbb{N}$ *(regardless of* B_{BRA} *in* aux*),* LF *is* $(3, B, \xi^3, N_{\mathsf{LF}})$*-invertible for*

$$N_{\mathsf{LF}} := 6\gamma_R^2 N^3 \iota(\mathcal{C}, 3)\|\mathcal{C}\|_R \ . \tag{7}$$

Note that LF is not hiding because the message part mP is included in the commitment cm (but note that the other message part mS is hidden because it is under a Pedersen commitment). The proof of Item 1 follows directly from the definition of the BRA. We now prove the other two items.

Proof of Item 2. Let $H := \{-1, 1\}$ and $\ell := \log n$. We define $p_{\mathsf{m}} := (\mathsf{mP}, p_{\mathsf{mP}}, p_{\mathsf{mS}})$, where p_{mP} and p_{mS} are the multilinear polynomials corresponding to the messages $\mathsf{mP} \in M_{\mathrm{L}}^n$ and $\mathsf{mS} \in M_{\mathrm{L}}^n$ respectively (see Definition 3.5). We also define p_{ck} to be the multilinear polynomial corresponding to $\mathsf{ck}_0 \in M_{\mathrm{R}}^n$ (see Definition 3.5). We define the functions $f_{\mathsf{CM}}, g_{\mathsf{CM}}, \phi_{\mathrm{sc}}$ as

- $f_{\mathsf{CM}}((\underline{aP}_0, aP_1, aS), \mathsf{G}, c) := 2^{-\ell}(c \cdot \underline{aP}_0, aS \cdot \mathsf{G}, aP_1 \cdot aS)$;
- $g_{\mathsf{CM}}(\mathsf{ck}, \rho) := (0, \langle \rho, \mathsf{ck}_1\rangle, 0)$;
- $\phi_{\mathrm{sc}}(\mathsf{cm}, \mathsf{cm}', c) := \mathbb{I}[c \cdot \mathsf{cm} \stackrel{?}{=} \mathsf{cm}']$;
- for every $i \in \{0, 1, \ldots, \ell\}$, $p = (\underline{c}, p_1, p_2) \in R^n \times (R \times M_{\mathrm{L}})[X_{i+1}, \ldots, X_\ell]$, and $(r_1, \ldots, r_i) \in \mathcal{C}^i$, $\alpha_{\mathrm{sc}}(\mathsf{ck}, p, r_1, \ldots, r_i) := 1$ if only if $p_1 = p_{\underline{c}}(r_1, \ldots, r_i, X_{i+1}, \ldots, X_\ell)$.

For a key $\mathsf{ck} \in M_{\mathrm{R}}^n \times M_{\mathrm{R}}^h$, message $\mathsf{m} \in M_{\mathrm{L}}^n$, randomness $\rho \in M_{\mathrm{L}}^h$, and slackness $c \in R$, it holds that

$$\begin{aligned}\sum_{\underline{\omega} \in \{-1,1\}^\ell} f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}), c) &= \sum_{\underline{\omega} \in \{-1,1\}^\ell} 2^{-\ell}(\mathsf{mP}, c \cdot p_{\mathsf{mS}}(\underline{\omega}) \cdot p_{\mathsf{ck}}(\underline{\omega}), c \cdot p_{\mathsf{mS}}(\underline{\omega}) \cdot p_{\mathsf{mP}}) \\ &= (\mathsf{mP}, c\langle \mathsf{mS}, \mathsf{ck}_0\rangle, c\langle \mathsf{mP}, \mathsf{mS}\rangle) \ .\end{aligned}$$

The last equality follows from Lemma 3.6. This implies that

$$\begin{aligned}\mathsf{LF.Commit}(\mathsf{ck}, \mathsf{m}; \rho) &= (\mathsf{mP}, \langle \mathsf{mS}, \mathsf{ck}_0\rangle + \langle \rho, \mathsf{ck}_1\rangle, \langle \mathsf{mP}, \mathsf{mS}\rangle) \\ &= \sum_{\underline{\omega} \in \{-1,1\}^\ell} f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}), 1) + g_{\mathsf{CM}}(\mathsf{ck}_0, \mathsf{ck}_1, \rho) \ .\end{aligned}$$

We have $\mathsf{LF.Open}(\mathsf{ck}, \mathsf{m}, \rho, \mathsf{cm}, c) = 1$ if and only if $c \cdot \mathsf{cm} = (c \cdot \mathsf{mP}, \langle \mathsf{mS}, \mathsf{ck}_0 \rangle + c\langle \rho, \mathsf{ck}_1 \rangle, \langle \mathsf{mP}, \mathsf{mS} \rangle)$, which is true if and only if $c \cdot (\mathsf{cm} - g_{\mathsf{CM}}(\mathsf{ck}_0, \mathsf{ck}_1, \rho)) = \sum_{\underline{\omega} \in \{-1,1\}^\ell} f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}), c)$, implying the required property of ϕ_{sc}. Also, we have $d_{\mathsf{ck}} = 1$ and $d^\star_{\mathsf{ck}} = 2$. □

Proof of Item 4. Let $\mathsf{ck} = (\mathsf{ck}_0, \mathsf{ck}_1) \in M_{\mathrm{R}}^{n+h}$ be the commitment key sampled in the first step of the invertibility definition. Also, fix an output of $\mathcal{A}$: an index $i \in [\ell]$; a challenge vector $(r_1, \ldots, r_{i-1}) \in \mathcal{C}^{i-1}$; distinct challenges $r_i^{(1)}, r_i^{(2)}, r_i^{(3)} \in \mathcal{C}$; polynomials $p_1, p_2, p_3 \in R^n \times (R \times M_{\mathrm{L}})[X_{i+1}, \ldots, X_\ell]$; a polynomial $q(X) = q_0 + q_1 X + q_2 X^2$ in $R^n \times (M_{\mathrm{T}} \times M_{\mathrm{L}})[X]$; and a slackness $c \in R$. Moreover, suppose that $\mathcal{A}$'s outputs satisfy the experiment's checks (or else the experiment just outputs 1).

We introduce some notation:

- $\underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0]$ and $\underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1]$ are the coefficients in $p_{\mathsf{ck}}(r_1, \ldots, r_{i-1}, X_i, \ldots, X_\ell)$ for monomials without X_i and with X_i respectively;
- $\underline{qC} \in R^n$ is the first part of q, which is a vector in R^n;
- $\underline{qC}[r_1, \ldots, r_{i-1}; 0]$ and $\underline{qC}[r_1, \ldots, r_{i-1}; 1]$ are the coefficients in $p_{\underline{qC}}(r_1, \ldots, r_{i-1}, X_i, \ldots, X_\ell)$ for monomials without X_i and with X_i respectively;
- writing $p_j = (\underline{p}_{j,P,0}, p_{j,P,1}, p_{j,S})$, $\underline{p}_{j,S}$ are the coefficients of $p_{j,S}$

For every $j \in [3]$, it holds that:

$$
\begin{aligned}
&c \cdot (q_0 + q_1 r_i^{(j)} + q_2 {r_i^{(j)}}^2) \\
&= c \cdot q(r_i^{(j)}) \\
&= c \sum_{\underline{\omega} \in \{-1,1\}^{\ell - i}} f_{\mathsf{CM}}(p_j(\underline{\omega}), p_{\mathsf{ck}}(r_1, \ldots, r_{i-1}, r_i^{(j)}, \underline{\omega})) \\
&= 2^{-i} \sum_{\underline{\omega} \in \{-1,1\}^{\ell - i}} \Big(c \cdot \underline{p}_{j,P,0}, p_{j,S}(\underline{\omega}) p_{\mathsf{ck}}(r_1, \ldots, r_{i-1}, r_i^{(j)}, \underline{\omega}), p_{j,P,1}(\underline{\omega}) p_{j,S}(\underline{\omega}) \Big) \\
&= 2^{-i} \big(c \cdot \underline{qC}, \langle \underline{p}_{j,S}, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0] + r_i^{(j)} \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1] \rangle, \langle \underline{p}_{j,P,1}, \underline{p}_{j,S} \rangle \big) \\
&= 2^{-i} \big(c \cdot \underline{qC}, \langle \underline{p}_{j,S}, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0] + r_i^{(j)} \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1] \rangle, \\
&\qquad\qquad \langle \underline{qC}[r_1, \ldots, r_{i-1}; 0] + r_i^{(j)} \underline{qC}[r_1, \ldots, r_{i-1}; 1], \underline{p}_{j,S} \rangle \big) \ .
\end{aligned}
$$

The second to last equality follows from Lemma 3.6 and the fact that the first coordinate of $q(X)$ is a constant in R^n ($\underline{p}_{j,P,0} = \underline{qC}$ for every j). The last equality follows because p_j is admissible.

Similarly to the proof of Lemma 5.11, we can find representations of $\xi^3 c \cdot q_0, \xi^3 c \cdot q_1, \xi^3 c \cdot q_2$ in terms of $\underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0]$, $\underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1]$, $\underline{qC}[r_1, \ldots, r_{i-1}; 0]$, and $\underline{qC}[r_1, \ldots, r_{i-1}; 1]$. Namely, we can find $a_0, a_1, a_2, b_0, b_1, b_2 \in M_{\mathrm{L}}^{2^{\ell - i}}$ with M_{L}-norm at most $3\gamma_R^2 N^3 \iota(\mathcal{C}, 3) \|\mathcal{C}\|_R \max_{j \in [3]} \|p_j\|_{M_{\mathrm{L}}}$ such that

$$
\begin{aligned}
&\xi^3 c \cdot q(X) \\
&= 2^{-i} \Big(\xi^3 c \cdot \underline{qC}, \\
&\qquad \langle a_0 + a_1 X + a_2 X^2, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0] \rangle + \langle b_0 + b_1 X + b_2 X^2, \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1] \rangle, \\
&\qquad \langle \underline{qC}[r_1, \ldots, r_{i-1}; 0], a_0 + a_1 X + a_2 X^2 \rangle + \langle \underline{qC}[r_1, \ldots, r_{i-1}; 1], b_0 + b_1 X + b_2 X^2 \rangle \Big) \ .
\end{aligned}
$$

By summing the above equation over $H = \{-1, 1\}$, we obtain that

$$
\xi^3 c \cdot \big(q(1) + q(-1) \big)
$$

$$= 2^{-i+1}\Big(\xi^3 c \cdot \underline{qC},$$
$$\langle (a_0 + a_2, b_0 + b_2), (\underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 0], \underline{p_{\mathsf{ck}}}[r_1, \ldots, r_{i-1}; 1])\rangle,$$
$$\langle (p_{\underline{qC}}[r_1, \ldots, r_{i-1}; 0], p_{\underline{qC}}[r_1, \ldots, r_{i-1}; 1]), (a_0 + a_2, b_0 + b_2)\rangle\Big) \ .$$

Now consider the vector $\underline{v_i} := (a_0 + a_2, b_0 + b_2) \in M_{\mathrm{L}}^{2^{\ell-i+1}}$ and the polynomial

$$p(\underline{X}) := (\underline{qC}, p_{\underline{qC}}(r_1, \ldots, r_{i-1}, \underline{X}), p_{\underline{v_i}}(\underline{X})) \in (R^n \times R \times M_{\mathrm{L}})[X_i, \ldots, X_\ell]$$

obtained using Definition 3.5. Note that $\underline{v_i}$, and thus p, has M_{L}-norm at most

$$6\gamma_R^2 N^3 \iota(\mathcal{C}, 3)\|\mathcal{C}\|_R \max_{j\in[3]} \|p_j\|_{M_{\mathrm{L}}} = N_{\mathsf{LF}} \max_{j\in[3]} \|p_j\|_{M_{\mathrm{L}}}$$

and satisfies the following equation:

$$\xi^3 c \cdot (q(1) + q(-1)) = \sum_{\omega_i,\ldots,\omega_\ell \in \{-1,1\}} f_{\mathsf{cM}}\Big((\underline{qC}, p_{\underline{qC}}(r_1, \ldots, r_{i-1}, \underline{\omega}), p(\omega_i, \ldots, \omega_\ell)),$$
$$p_{\mathsf{ck}}(r_1, \ldots, r_{i-1}, \omega_i, \ldots, \omega_\ell), \xi^3 \cdot c\Big) \ .$$

which implies that ϕ_{sc} evaluates to 1 as required. Moreover, p is ck-admissible and all the derivations to obtain p, the desired polynomial, are efficient. □

Remark 5.14 (polynomial evaluation)**.** Linear functions capture polynomial evaluation, in any basis (whether in monomial basis, in Lagrange basis, or other bases). We give some examples below.

- *Univariate polynomial in monomial basis.* The evaluation of $p(X) = \sum_{i=0}^{d} c_i X^i \in M_{\mathrm{L}}[X]$ at $z \in R$ is $\langle \mathsf{mP}, \mathsf{mS}\rangle \in M_{\mathrm{L}}$ for $\mathsf{mP} := (1, z, z^2, \ldots, z^d) \in R^{d+1}$ and $\mathsf{mS} := (c_0, \ldots, c_d) \in M_{\mathrm{L}}^{d+1}$.
- *Univariate polynomial in Lagrange basis.* Let $S \subseteq R$ be a strong sampling set for R (Definition 3.10), and for every $a \in S$ let $\ell_{S,a}(X) := \prod_{b\in S\setminus\{a\}}(a-b)^{-1}(X-b) \in R[X]$ be the a-th Lagrange polynomial over S. The evaluation of $p(X) = \sum_{a\in S} c_a \ell_{S,a}(X) \in M_{\mathrm{L}}[X]$ at $z \in R$ is $\langle \mathsf{mP}, \mathsf{mS}\rangle \in M_{\mathrm{L}}$ for $\mathsf{mP} := (\ell_{S,a}(z))_{a\in S} \in R^{|S|}$ and $\mathsf{mS} := (c_a)_{a\in S} \in M_{\mathrm{L}}^{|S|}$.
- *Multilinear polynomial in monomial basis.* The evaluation of $p(X_1, \ldots, X_m) = \sum_{i_1,\ldots,i_m\in\{0,1\}} c_{i_1,\ldots,i_m} X_1^{i_1} \cdots X_m^{i_m}$ at $(z_1, \ldots, z_m) \in R^m$ is $\langle \mathsf{mP}, \mathsf{mS}\rangle \in M_{\mathrm{L}}$ for $\mathsf{mP} := (\prod_{j\in[m]} z_j^{i_j})_{i_1,\ldots,i_m\in\{0,1\}} \in R^{2^m}$ and $\mathsf{mS} := (c_{i_1,\ldots,i_m})_{i_1,\ldots,i_m\in\{0,1\}} \in M_{\mathrm{L}}^{2^m}$.
- *Multilinear polynomial in Lagrange basis.* As above, $\ell_{\{0,1\},b}(X) := bX + (1-b)X$ is the b-th Lagrange polynomial over $\{0,1\}$. The evaluation of $p(X_1, \ldots, X_m) = \sum_{i_1,\ldots,i_m\in\{0,1\}} c_{i_1,\ldots,i_m} \prod_{j\in[m]} \ell_{\{0,1\},i_j}(X)$ at $(z_1, \ldots, z_m) \in R^m$ is $\langle \mathsf{mP}, \mathsf{mS}\rangle \in M_{\mathrm{L}}$ for $\mathsf{mP} := (\prod_{j\in[m]} \ell_{\{0,1\},i_j}(z_j))_{i_1,\ldots,i_m\in\{0,1\}} \in R^{2^m}$ and $\mathsf{mS} := (c_{i_1,\ldots,i_m})_{i_1,\ldots,i_m\in\{0,1\}} \in M_{\mathrm{L}}^{2^m}$.

Remark 5.15 (polynomial commitments over rings)**.** We have explained that linear functions capture polynomial evaluation (Remark 5.14), and this implies that we can use our sumcheck argument to obtain polynomial commitment schemes for polynomials over rings, as we now explain.

In the commit phase, the sender commits to a polynomial p via a Pedersen commitment: $\mathsf{C} := \langle p, \mathsf{ck}_0\rangle + \langle \rho, \mathsf{ck}_1\rangle$ for randomness ρ. Subsequently, in the reveal phase (which can be repeated any number of times for this commitment), the sender can reveal evaluations of p by applying our sumcheck argument to appropriate statements about the commitment scheme LF. In more detail, suppose that the sender wishes to prove the

statement "given a commitment C, evaluation point z, and claimed value v, I know a polynomial p and randomness ρ such that $\mathsf{C} = \langle p, \mathsf{ck}_0 \rangle + \langle \rho, \mathsf{ck}_1 \rangle$ and $p(z) = v$". Then the sender and receiver engage in a sumcheck argument for the instance $\mathbb{x} = (\mathsf{CM}, \mathsf{pp}, \mathcal{C}, \mathsf{ck}, \mathsf{cm})$ and witness $\mathbb{w} = (\mathsf{m}, \rho)$ where

$$\mathsf{CM} := \mathsf{LF} \quad , \quad \mathsf{ck} := (\mathsf{ck}_0, \mathsf{ck}_1) \quad , \quad \mathsf{cm} := (f(z), \mathsf{C}, v) \quad , \quad \mathsf{m} := (\mathsf{mP}, \mathsf{mS}) = (f(z), p) \ ,$$

where f is a function depending on the representation of p. Tree extraction with slackness factor c implies that the extractor can find a polynomial p and randomness ρ such that $c \cdot \mathsf{C} = \langle p, \mathsf{ck}_0 \rangle + c\langle \rho, \mathsf{ck}_1 \rangle$ and $p(z) = cv$. This may be enough for applications, or else one can eliminate the slackness factor by "quotienting out".

This "quotienting" strategy, which we do use later in Section 6, uses bilinear module generators that are quotient-friendly (Definition 5.8), which means that there is a submodule $I \subseteq M_{\mathrm{L}}$[6] such that multiplication by the slackness factor $c \in R$ can be inverted in the quotient module M_{L}/I. Then, while computing commitments over M_{L} as before, we can view C as a commitment to $p \bmod I$ in M_{L}/I rather than a commitment to p in M_{L}. The knowledge extractor can find a polynomial p satisfying $p(z) = cv$ over M_{L}, which implies that $c^{-1}p(z) = v \mod I$. This method yields openings without slackness in the case where multiplication by c may not be invertible in M_{L}.

Note that our sumcheck argument is not zero knowledge so the opening algorithm of the polynomial commitment scheme outlined above is also not zero knowledge. There are several ways to achieve a zero-knowledge opening algorithm.

- One option is to devise a zero-knowledge sumcheck argument (say, by leveraging ideas for zero-knowledge sumcheck protocols [BCFGRS17] or by adding randomness in each round of the sumcheck argument) and apply that to LF as above; we leave this interesting direction to future work.
- Another (and somewhat overkill) option is to rely on the zero-knowledge succinct argument for *scalar products* that we design in Section 6. This latter, which is based on applying our sumcheck argument to the scalar-product commitment scheme SP from Section 5.4 and then adding zero knowledge, captures polynomial evaluation because scalar products generalize linear functions (both parts of the message can be committed rather than just one) and also takes care of quotienting out slackness factors.

A key difference between the two options is that the sumcheck argument is a *proof* of knowledge for LF but merely an *argument* of knowledge for SP. This is because in the former invertibility holds unconditionally (Lemma 5.13) while in the latter invertibility holds if SP is binding (Lemma 5.17).

5.4 Scalar-product commitment

We define a commitment scheme that includes the scalar-product of two committed messages. For this commitment scheme, we assume that M_{L} in the bilinear module is a ring, so that the scalar-product of two message vectors with entries in M_{L} is well-defined (which is not the case if M_{L} is merely a module).

Definition 5.16. *Let* $\mathsf{BM} = (\mathsf{Setup}, \mathsf{KeyGen})$ *be a bilinear-module generator that is scalar-product compatible. The* **scalar-product commitment scheme** *is defined via the following algorithms.*

- $\mathsf{SP.Setup}(1^\lambda, n)$: *sample* $(\mathcal{M}, h, \mathsf{aux}) \leftarrow \mathsf{BM.Setup}(1^\lambda, n)$ *and output* $\mathsf{pp} := (\mathcal{M}, h, \mathsf{aux})$.
- $\mathsf{SP.KeyGen}(\mathsf{pp})$: *sample*

$$\mathsf{ckL} = (\mathsf{ckL}_0, \mathsf{ckL}_1) \leftarrow \mathsf{BM.KeyGen}(\mathcal{M}, h, \mathsf{aux}) \ ,$$

[6] If BM is scalar-product compatible and M_{L} is a ring, then I will be an ideal of M_{L}.

$$\mathsf{ckR} = (\mathsf{ckR}_0, \mathsf{ckR}_1) \leftarrow \mathsf{BM.KeyGen}(\mathcal{M}, h, \mathsf{aux}) \ ,$$
$$(\mathsf{ckP}^\star, \mathsf{ckH}) \leftarrow \mathsf{BM.KeyGen}(\mathcal{M}, h, \mathsf{aux}) \ ,$$

set $\mathsf{ckP} := \mathsf{ckP}^\star[1] \in M_{\mathrm{R}}$*, and output* $\mathsf{ck} := (\mathsf{ckL}_0, \mathsf{ckR}_0, \mathsf{ckP}, \mathsf{ckL}_1, \mathsf{ckR}_1, \mathsf{ckH}) \in M_{\mathrm{R}}^{2n+1} \times M_{\mathrm{R}}^{3h}$.

- $\mathsf{SP.Commit}(\mathsf{ck}, \mathsf{m}; \rho)$: *given* $\mathsf{m} = (\mathsf{mL}, \mathsf{mR}) \in M_{\mathrm{L}}^{2n}(B_{\mathsf{C}})$ *and* $\rho = (\rho\mathsf{L}, \rho\mathsf{R}, \rho\mathsf{H}) \in M_{\mathrm{L}}^{3h}$ *where* $\rho\mathsf{L}, \rho\mathsf{R}, \rho\mathsf{H} \leftarrow \mathcal{U}_{M_{\mathrm{L}}}$*, output* $\mathsf{cm} := (\mathsf{cmL}, \mathsf{cmR}, \mathsf{cmP}) \in M_{\mathrm{T}}^3$ *where*

$$\mathsf{cmL} := \langle \mathsf{mL}, \mathsf{ckL}_0 \rangle + \langle \rho\mathsf{L}, \mathsf{ckL}_1 \rangle \ ,$$
$$\mathsf{cmR} := \langle \mathsf{mR}, \mathsf{ckR}_0 \rangle + \langle \rho\mathsf{R}, \mathsf{ckR}_1 \rangle \ ,$$
$$\mathsf{cmP} := \langle \mathsf{mL}, \mathsf{mR} \rangle \cdot \mathsf{ckP} + \langle \rho\mathsf{H}, \mathsf{ckH} \rangle \ .$$

- $\mathsf{SP.Open}(\mathsf{ck}, \mathsf{m}, \rho, \mathsf{cm}, c)$: *check that* $\mathsf{m} = (\mathsf{mL}, \mathsf{mR}) \in M_{\mathrm{L}}^{2n}(B_{\mathsf{BRA}})$*,* $\rho = (\rho\mathsf{L}, \rho\mathsf{R}, \rho\mathsf{H}) \in M_{\mathrm{L}}^{3h}(B_{\mathsf{BRA}})$*, and, parsing* cm *and* $(\mathsf{cmL}, \mathsf{cmR}, \mathsf{cmP})$*, also that*

$$c \cdot \mathsf{cmL} = \langle \mathsf{mL}, \mathsf{ckL}_0 \rangle + c\langle \rho\mathsf{L}, \mathsf{ckL}_1 \rangle \ ,$$
$$c \cdot \mathsf{cmR} = \langle \mathsf{mR}, \mathsf{ckR}_0 \rangle + c\langle \rho\mathsf{R}, \mathsf{ckR}_1 \rangle \ ,$$
$$c^2 \cdot \mathsf{cmP} = \langle \mathsf{mL}, \mathsf{mR} \rangle \cdot \mathsf{ckP} + c^2\langle \rho\mathsf{H}, \mathsf{ckH} \rangle \ .$$

Lemma 5.17. *If* BM *is secure then* SP *is a sumcheck-friendly invertible commitment scheme. In more detail:*

1. *if* BM *satisfies the BRA then* SP *is binding;*
2. *if* h *is* $(\mathcal{M}, \mathcal{U}_{M_{\mathrm{L}}}, \mathcal{U}_{M_{\mathrm{R}}}, \mathrm{negl}(\lambda))$*-hiding then* SP *is computationally hiding;*
3. SP *is sumcheck-friendly;*
4. *if* SP *is binding and* $(\mathcal{C}, \xi, N)$ *are pseudoinverse parameters for* (R, M_{T}) *then* SP *is* $(4, B_{\mathsf{SP}}, \xi^3, N_{\mathsf{SP}})$*-invertible for*

$$B_{\mathsf{SP}} := \frac{B_{\mathsf{BRA}}}{3\gamma_R^2 \|\mathcal{C}\|_R^2 \|\xi^3\|_R} \ , \tag{8}$$
$$N_{\mathsf{SP}} := 3\gamma_R^2 N^3 \iota(\mathcal{C}, 3) \|\mathcal{C}\|_R \ . \tag{9}$$

The proofs of Items 1 and 2 follow directly from the definition of the BRA and the hiding constant of BM. We now prove the other two items.

Proof of Item 3. Let $H := \{-1, 1\}$ and $\ell := \log n$. For a message $\mathsf{m} = (\mathsf{mL}, \mathsf{mR}) \in M_{\mathrm{L}}^{2n}$, we define $p_{\mathsf{m}} := (p_{\mathsf{mL}}, p_{\mathsf{mR}}) \in M_{\mathrm{L}}^2[X_1, \ldots, X_\ell]$ where $p_{\mathsf{mL}}, p_{\mathsf{mR}}$ are the multilinear polynomials respectively obtained from $\mathsf{mL}, \mathsf{mR} \in M_{\mathrm{L}}^n$ via Definition 3.5. For a key $\mathsf{ck} = (\mathsf{ckL}_0, \mathsf{ckR}_0, \mathsf{ckP}, \mathsf{ckL}_1, \mathsf{ckR}_1, \mathsf{ckH})$, we define $p_{\mathsf{ck}} := (p_{\mathsf{ckL}_0}, p_{\mathsf{ckR}_0}, p_{\mathsf{ckP}}) \in M_{\mathrm{L}}^3[X_1, \ldots, X_\ell]$ where $p_{\mathsf{ckL}_0}, p_{\mathsf{ckR}_0}$ are the multilinear polynomials respectively obtained from $\mathsf{ckL}_0, \mathsf{ckR}_0 \in M_{\mathrm{R}}^n$ via Definition 3.5 and ckP is a degenerate 0-degree polynomial equal to the constant $\mathsf{ckP} \in M_{\mathrm{R}}$. We define:

- $f_{\mathsf{CM}} \colon M_{\mathrm{L}}^2 \times M_{\mathrm{R}}^3 \times R \to M_{\mathrm{T}}^3$ as $f_{\mathsf{CM}}((a_{\mathrm{L}}, a_{\mathrm{R}}), (\mathsf{G}_{\mathrm{L}}, \mathsf{G}_{\mathrm{R}}, \mathsf{G}_{SC}), c) := 2^{-\ell}(\ a_{\mathrm{L}} \cdot \mathsf{G}_{\mathrm{L}},\ a_{\mathrm{R}} \cdot \mathsf{G}_{\mathrm{R}}, a_{\mathrm{L}} a_{\mathrm{R}} \mathsf{G}_{SC})$;
- $g_{\mathsf{CM}} \colon M_{\mathrm{R}}^{2n+1+3h} \times M_{\mathrm{L}}^{3h} \to M_{\mathrm{T}}^3$ as $g_{\mathsf{CM}}(\mathsf{ck}, \rho) := (\langle \rho\mathsf{L}, \mathsf{ckL}_1 \rangle, \langle \rho\mathsf{R}, \mathsf{ckR}_1 \rangle, \langle \rho\mathsf{H}, \mathsf{ckH} \rangle)$;
- $\phi_{\mathrm{sc}} \colon M_{\mathrm{T}} \times M_{\mathrm{T}} \times R \to \{0, 1\}$ as $\phi_{\mathrm{sc}}(\mathsf{cm}, \mathsf{cm}', c) = \mathbb{I}[(c \cdot \mathsf{cmL}, c \cdot \mathsf{cmR}, c^2\mathsf{cmP}) \stackrel{?}{=} \mathsf{cm}']$, where $\mathsf{cm} = (\mathsf{cmL}, \mathsf{cmR}, \mathsf{cmP})$, which checks equality when $c = 1$.

- $\alpha_{\mathrm{sc}} \equiv 1$. (For every $i \in \{0,1,\ldots,\ell\}$, $p \in M_{\mathrm{L}}^2[X_{i+1},\ldots,X_\ell]$, and $(r_1,\ldots,r_i) \in \mathcal{C}^i$, there exists a message $\mathsf{m} = (\mathsf{mL},\mathsf{mR}) \in M_{\mathrm{L}}^{2n}$ such that $p(X_{i+1},\ldots,X_\ell) = (p_{\mathsf{mL}}(r_1,\ldots,r_i,X_{i+1},\ldots,X_\ell), p_{\mathsf{mR}}(r_1,\ldots,r_i,X_{i+1},\ldots,X_\ell))$.)

For a key $\mathsf{ck} \in M_{\mathrm{R}}^{2n+1+3h}$, message $\mathsf{m} \in M_{\mathrm{L}}^{2n}$, randomness $\rho \in M_{\mathrm{L}}^{3h}$, and slackness $c \in R$, one can show, using Lemma 3.6 and arguments similar to the proof of Lemma 5.11, that

$$\begin{aligned}
&\sum_{\underline{\omega}\in\{-1,1\}^\ell} f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}), c) \\
&= \sum_{\underline{\omega}\in\{-1,1\}^\ell} 2^{-\ell}\left(p_{\mathsf{mL}}(\underline{\omega})\cdot p_{\mathsf{ckL}_0}(\underline{\omega}), p_{\mathsf{mR}}(\underline{\omega})\cdot p_{\mathsf{ckR}_0}(\underline{\omega}), p_{\mathsf{mL}}(\underline{\omega})\cdot p_{\mathsf{mR}}(\underline{\omega})\cdot p_{\mathsf{ckP}}\right) \\
&= \left(\langle \mathsf{mL},\mathsf{ckL}_0\rangle, \langle \mathsf{mR},\mathsf{ckR}_0\rangle, \langle \mathsf{mL},\mathsf{mR}\rangle\cdot \mathsf{ckP}\right) \ .
\end{aligned}$$

This implies that

$$\begin{aligned}
&\mathsf{SP.Commit}(\mathsf{ck},\mathsf{m};\rho) \\
&= \left(\langle \mathsf{mL},\mathsf{ckL}_0\rangle, \langle \mathsf{mR},\mathsf{ckR}_0\rangle, \langle \mathsf{mL},\mathsf{mR}\rangle\cdot \mathsf{ckP}\right) + \left(\langle \rho\mathsf{L},\mathsf{ckL}_1\rangle, \langle \rho\mathsf{R},\mathsf{ckR}_1\rangle, \langle \rho\mathsf{H},\mathsf{ckH}\rangle\right) \\
&= \sum_{\underline{\omega}\in\{-1,1\}^\ell} f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}), 1) + g_{\mathsf{CM}}(\mathsf{ck},\rho) \ .
\end{aligned}$$

We have $\mathsf{SP.Open}(\mathsf{ck},\mathsf{m},\rho,\mathsf{cm},c) = 1$ if and only if

$$(c{\cdot}\mathsf{cmL}, c{\cdot}\mathsf{cmR}, c^2{\cdot}\mathsf{cmP}) = \left(\langle \mathsf{mL},\mathsf{ckL}_0\rangle + c\langle \rho\mathsf{L},\mathsf{ckL}_1\rangle, \langle \mathsf{mR},\mathsf{ckR}_0\rangle + c\langle \rho\mathsf{R},\mathsf{ckR}_1\rangle, \langle \mathsf{mL},\mathsf{mR}\rangle{\cdot}\mathsf{ckP} + c^2\langle \rho\mathsf{H},\mathsf{ckH}\rangle\right) \ ,$$

which is true if and only if $(c,c,c^2)\cdot((\mathsf{cmL},\mathsf{cmR},\mathsf{cmP}) - g_{\mathsf{CM}}(\mathsf{ck}_0,\mathsf{ck}_1,\rho)) = \sum_{\underline{\omega}\in\{-1,1\}^\ell} f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{\omega}), p_{\mathsf{ck}}(\underline{\omega}), c)$, implying the required property of ϕ_{sc}. Finally, we have $d_{\mathsf{ck}} = 1$ and $d^\star_{\mathsf{ck}} = 2$.

□

Proof of Item 4. Let $\mathsf{ck} \in M_{\mathrm{R}}^{2n+1} \times M_{\mathrm{R}}^{3h}$ be the commitment key sampled in the first step of the invertibility definition. Also, fix an output of $\mathcal{A}$: an index $i \in [\ell]$; a challenge vector $(r_1,\ldots,r_{i-1}) \in \mathcal{C}^{i-1}$; distinct challenges $r_i^{(1)}, r_i^{(2)}, r_i^{(3)}, r_i^{(4)} \in \mathcal{C}$; polynomials $p_1,p_2,p_3,p_4 \in M_{\mathrm{L}}^2[X_{i+1},\ldots,X_\ell]$; a polynomial $q(X) = (qL,qR,qP)(X) = (qL_0,qR_0,qP_0) + (qL_1,qR_1,qP_1)X + (qL_2,qR_2,qP_2)X^2$ in $M_{\mathrm{T}}^3[X]$; and a slackness $c \in R$. Moreover, suppose that $\mathcal{A}$'s outputs satisfy the experiment's checks (or else the experiment just outputs 1).

We introduce some notation:

- $\underline{p_{\mathsf{ckL}}}[r_1,\ldots,r_{i-1};0]$ and $\underline{p_{\mathsf{ckL}}}[r_1,\ldots,r_{i-1};1]$ are the coefficients in $p_{\mathsf{ckL}}(r_1,\ldots,r_{i-1},X_i,\ldots,X_\ell)$ for monomials without X_i and with X_i respectively;
- $\underline{p_{\mathsf{ckR}}}[r_1,\ldots,r_{i-1};0]$ and $\underline{p_{\mathsf{ckR}}}[r_1,\ldots,r_{i-1};1]$ are the coefficients in $p_{\mathsf{ckR}}(r_1,\ldots,r_{i-1},X_i,\ldots,X_\ell)$ for monomials without X_i and with X_i;
- for $j \in [4]$, $\underline{p}_{j,L}$ and $\underline{p}_{j,R}$ are the coefficients of the two coordinates of p_j.

For every $j \in [4]$ it holds that:

$$\begin{aligned}
&(c\cdot qL_0, c\cdot qR_0, c^2\cdot qP_0) + (c\cdot qL_1, c\cdot qR_1, c^2\cdot qP_1)r_i^{(j)} + (c\cdot qL_2, c\cdot qR_2, c^2\cdot qP_2){r_i^{(j)}}^2 \\
&= (c\cdot qL, c\cdot qR, c^2 qP)(r_i^{(j)}) \\
&= \sum_{\omega_{i+1},\ldots,\omega_\ell\in H} f_{\mathsf{CM}}\Big(p_j(\omega_{i+1},\ldots,\omega_\ell), p_{\mathsf{ck}}(r_1,\ldots,r_{i-1},r_i^{(j)},\omega_{i+1},\ldots,\omega_\ell), c\Big)
\end{aligned}$$

$$
\begin{aligned}
&= 2^{-\ell} \sum_{\omega_{i+1},\ldots,\omega_\ell} \Big(\underline{p}_{j,L}(\omega_{i+1},\ldots,\omega_\ell) \cdot p_{\mathsf{ckL}}(r_1,\ldots,r_{i-1},r_i^{(j)},\omega_{i+1},\ldots,\omega_\ell), \\
&\qquad\qquad \underline{p}_{j,R}(\omega_{i+1},\ldots,\omega_\ell) \cdot p_{\mathsf{ckR}}(r_1,\ldots,r_{i-1},r_i^{(j)},\omega_{i+1},\ldots,\omega_\ell), \\
&\qquad\qquad \underline{p}_{j,L}(\omega_{i+1},\ldots,\omega_\ell)\underline{p}_{j,R}(\omega_{i+1},\ldots,\omega_\ell) \cdot \mathsf{ckP} \Big) \\
&= 2^{-i} \Big(\langle \underline{p}_{j,L}, \underline{p}_{\mathsf{ckL}}[r_1,\ldots,r_{i-1};0] + r_i^{(j)} \underline{p}_{\mathsf{ckL}}[r_1,\ldots,r_{i-1};1] \rangle, \\
&\qquad \langle \underline{p}_{j,R}, \underline{p}_{\mathsf{ckR}}[r_1,\ldots,r_{i-1};0] + r_i^{(j)} \underline{p}_{\mathsf{ckR}}[r_1,\ldots,r_{i-1};1] \rangle, \\
&\qquad \langle \underline{p}_{j,L}, \underline{p}_{j,R} \rangle \cdot \mathsf{ckP} \Big) \ ,
\end{aligned} \tag{10}
$$

where the last equality follows from Lemma 3.6.

Similarly to the proof of Lemma 5.11, we can find representations of $(\xi^3 c \cdot qL, \xi^3 c \cdot qR, \xi^6 c^2 \cdot qP)$ in terms of $\underline{p}_{\mathsf{ckL}}[r_1,\ldots,r_{i-1};0]$, $\underline{p}_{\mathsf{ckL}}[r_1,\ldots,r_{i-1};1]$, $\underline{p}_{\mathsf{ckR}}[r_1,\ldots,r_{i-1};0]$, $\underline{p}_{\mathsf{ckR}}[r_1,\ldots,r_{i-1};1]$ and ckP. Namely, we can find $a_{L,0}, a_{L,1}, a_{L,2}, b_{L,0}, b_{L,1}, b_{L,2} \in M_{\mathrm{L}}^{2^{\ell-i}}$, $a_{R,0}, a_{R,1}, a_{R,2}, b_{R,0}, b_{R,1}, b_{R,2} \in M_{\mathrm{L}}^{2^{\ell-i}}$ with M_{L}-norm at most $3\gamma_R^2 N^3 \iota(\mathcal{C},3) \|\mathcal{C}\|_R \max_{j\in[4]} \|p_j\|_{M_{\mathrm{L}}} = N_{\mathsf{SP}} \max_{j\in[4]} \|p_j\|_{M_{\mathrm{L}}}$ and $a_{\bullet,0}, a_{\bullet,1}, a_{\bullet,2} \in M_{\mathrm{L}}^{2^{\ell-i}}$ such that

$$
\begin{aligned}
&(\xi^3 c \cdot qL, \xi^3 c \cdot qR, \xi^6 c^2 \cdot qP)(X) \\
&= 2^{-i} \Big((\langle a_{L,0} + a_{L,1}X + a_{L,2}X^2, \underline{p}_{\mathsf{ckL}}[r_1,\ldots,r_{i-1};0] \rangle + \langle b_{L,0} + b_{L,1}X + b_{L,2}X^2, \underline{p}_{\mathsf{ckL}}[r_1,\ldots,r_{i-1};1] \rangle), \\
&\qquad (\langle a_{R,0} + a_{R,1}X + a_{R,2}X^2, \underline{p}_{\mathsf{ckR}}[r_1,\ldots,r_{i-1};0] \rangle + \langle b_{R,0} + b_{R,1}X + b_{R,2}X^2, \underline{p}_{\mathsf{ckR}}[r_1,\ldots,r_{i-1};1] \rangle), \\
&\qquad (a_{\bullet,0} + a_{\bullet,1}X + a_{\bullet,2}X^2) \cdot \mathsf{ckP} \Big) \ .
\end{aligned} \tag{11}
$$

This step uses Equation (10) for three choices of $j \in [4]$; the fourth one is used for the following claim.

Claim 5.18.

$$
\begin{aligned}
&(\xi^3 c \cdot qL, \xi^3 c \cdot qR, \xi^6 c^2 \cdot qP)(X) \\
&= 2^{-i} \Big((\langle a_{L,0} + a_{L,1}X, \underline{p}_{\mathsf{ckL}}[r_1,\ldots,r_{i-1};0] \rangle + \langle (a_{L,0} + a_{L,1}X)X, \underline{p}_{\mathsf{ckL}}[r_1,\ldots,r_{i-1};1] \rangle), \\
&\qquad (\langle a_{R,0} + a_{R,1}X, \underline{p}_{\mathsf{ckR}}[r_1,\ldots,r_{i-1};0] \rangle + \langle (a_{R,0} + a_{R,1}X)X, \underline{p}_{\mathsf{ckR}}[r_1,\ldots,r_{i-1};1] \rangle), \\
&\qquad (\langle a_{L,0} + a_{L,1}X), a_{R,0} + a_{R,1}X \rangle) \cdot \mathsf{ckP} \Big) \ .
\end{aligned}
$$

Proof. The claimed equation follows from Equation (11) and the following equalities.

- $a_{L,0} = b_{L,1}, a_{L,1} = b_{L,2}, a_{L,2} = b_{L,0} = 0$. By comparing coefficients of $\underline{p}_{\mathsf{ckL}}[r_1,\ldots,r_{i-1};0]$ and $\underline{p}_{\mathsf{ckL}}[r_1,\ldots,r_{i-1};1]$ in Equation (11), we now argue that for every $j \in [4]$

$$
(a_{L,0} + a_{L,1} r_i^{(j)} + a_{L,2} {r_i^{(j)}}^2) r_i^{(j)} = b_{L,0} + b_{L,1} r_i^{(j)} + b_{L,2} {r_i^{(j)}}^2 \ . \tag{12}
$$

  This in turn implies that $a_{L,0} = b_{L,1}$, $a_{L,1} = b_{L,2}$, and $a_{L,2} = b_{L,0} = 0$.

  Suppose by way of contradiction that Equation (12) does not hold for some $j \in [4]$. Define the messages

$$
\begin{aligned}
m_1^{(j)} &:= (a_{L,0} + a_{L,1} r_i^{(j)} + a_{L,2} {r_i^{(j)}}^2, b_{L,0} + b_{L,1} r_i^{(j)} + b_{L,2} {r_i^{(j)}}^2) \ , \\
m_2^{(j)} &:= \xi^3 (\underline{p}_{j,L}, r_i^{(j)} \underline{p}_{j,L}) \ .
\end{aligned}
$$

Note that

$$\begin{aligned}\|m_1^{(j)}\|_{M_{\mathrm{L}}} &\leq 3\gamma_R^2\|\mathcal{C}\|_R^2 \max\{\|a_{L,0}\|_{M_{\mathrm{L}}}, \|a_{L,1}\|_{M_{\mathrm{L}}}, \|a_{L,2}\|_{M_{\mathrm{L}}}, \|b_{L,0}\|_{M_{\mathrm{L}}}, \|b_{L,1}\|_{M_{\mathrm{L}}}, \|b_{L,2}\|_{M_{\mathrm{L}}}\} \\ &\leq 3\gamma_R^2\|\mathcal{C}\|_R^2 N_{\mathsf{SP}} \max_{j\in[4]} \|p_j\|_{M_{\mathrm{L}}} \leq 3\gamma_R^2\|\mathcal{C}\|_R^2 B_{\mathsf{SP}} \leq 3\gamma_R^2\|\mathcal{C}\|_R^2 \frac{B_{\mathsf{BRA}}}{3\gamma_R^2\|\mathcal{C}\|_R^2\|\xi^3\|_R} \leq B_{\mathsf{BRA}} \ , \\ \|m_2^{(j)}\|_{M_{\mathrm{L}}} &\leq \gamma_R^2\|\xi^3\|_R\|\mathcal{C}\|_R \max_{j\in[4]} \|p_j\|_{M_{\mathrm{L}}} \leq \gamma_R^2\|\xi^3\|_R\|\mathcal{C}\|_R B_{\mathsf{SP}} \leq \gamma_R^2\|\xi^3\|_R\|\mathcal{C}\|_R \frac{B_{\mathsf{BRA}}}{3\gamma_R^2\|\mathcal{C}\|_R^2\|\xi^3\|_R} \leq B_{\mathsf{BRA}} \ .\end{aligned}$$

Moreover, from Equation (10) we deduce that

$$\langle m_1^{(j)}, (\underline{p}_{\mathsf{ckL}}[r_1,\ldots,r_{i-1};0], \underline{p}_{\mathsf{ckL}}[r_1,\ldots,r_{i-1};1]0\rangle = \langle m_2^{(j)}, (\underline{p}_{\mathsf{ckL}}[r_1,\ldots,r_{i-1};0], \underline{p}_{\mathsf{ckL}}[r_1,\ldots,r_{i-1};1])\rangle$$

which for $\mathsf{cm} := \xi^3\langle(\underline{p}_{j,L}, r_i^{(j)}\underline{p}_{j,L}), (\underline{p}_{\mathsf{ckL}}[r_1,\ldots,r_{i-1};0], \underline{p}_{\mathsf{ckL}}[r_1,\ldots,r_{i-1};1])\rangle$ implies that

$$\mathsf{SP.Open}(\mathsf{ck}, (m_1^{(j)}, 0), 0, \mathsf{cm}, \xi^3 c) = 1 \quad \text{and} \quad \mathsf{SP.Open}(\mathsf{ck}, (m_2^{(j)}, 0), 0, \mathsf{cm}, \xi^3 c) = 1 \ ,$$

which breaks the binding property of SP. Above we set the right part of the message $\mathsf{mR} := 0$ and the commitment randomness $\rho := 0$.

- $a_{R,0} = b_{R,1}, a_{R,1} = b_{R,2}, a_{R,2} = b_{R,0} = 0$. We apply the same argument as above for

$$\begin{aligned} m_1^{(j)} &:= (a_{R,0} + a_{R,1}r_i^{(j)} + a_{R,2}{r_i^{(j)}}^2, b_{R,0} + b_{R,1}r_i^{(j)} + b_{R,2}{r_i^{(j)}}^2) \ , \\ m_2^{(j)} &:= \xi^3(\underline{p}_{j,R}, r_i^{(j)}\underline{p}_{j,R}) \ . \end{aligned}$$

- $a_{\bullet,0} + a_{\bullet,1}X + a_{\bullet,2}X^2 = \langle a_{L,0} + a_{L,1}X, a_{R,0} + a_{R,1}X\rangle$. Below we argue that for each $j \in [4]$

$$\xi^3\underline{p}_{j,L} = a_{L,0} + a_{L,1}r_i^{(j)} \quad \text{and} \quad \xi^3\underline{p}_{j,R} = a_{R,0} + a_{R,1}r_i^{(j)} \ .$$

This in turn implies that for each $j \in [4]$

$$a_{\bullet,0} + a_{\bullet,1}r_i^{(j)} + a_{\bullet,2}{r_i^{(j)}}^2 = \xi^6\langle\underline{p}_{j,L}, \underline{p}_{j,R}\rangle = \langle a_{L,0} + a_{L,1}r_i^{(j)}, a_{R,0} + a_{R,1}r_i^{(j)}\rangle$$

where the first equation follows by combining Equation (10) and Equation (11). So, $a_{\bullet,0}+a_{\bullet,1}X+a_{\bullet,2}X^2 = \langle a_{L,0} + a_{L,1}X, a_{R,0} + a_{R,1}X\rangle$ as claimed.

Suppose by way of contradiction that there exists $j \in [4]$ such that $\xi^3\underline{p}_{j,L} \neq a_{L,0} + a_{L,1}r_i^{(j)}$. Then the two distinct messages $m_1^{(j)} := (a_{L,0} + a_{L,1}r_i^{(j)}, r_i^{(j)}(a_{L,0} + a_{L,1}r_i^{(j)}))$ and $m_2^{(j)} := (\xi^3\underline{p}_{j,L}, r_i^{(j)}\xi^3\underline{p}_{j,L})$ satisfy

$$\mathsf{SP.Open}(\mathsf{ck}, (m_1^{(j)}, 0), 0, \mathsf{cm}, \xi^3 c) = 1 \quad \text{and} \quad \mathsf{SP.Open}(\mathsf{ck}, (m_2^{(j)}, 0), 0, \mathsf{cm}, \xi^3 c) = 1 \ ,$$

which breaks the binding property of SP, since $\|m_1^{(j)}\|_{M_{\mathrm{L}}}, \|m_2^{(j)}\|_{M_{\mathrm{L}}} \leq B_{\mathsf{BRA}}$. A similar argument holds if there exists $j \in [4]$ such that $\xi^3\underline{p}_{j,R} \neq a_{R,0} + a_{R,1}r_i^{(j)}$.

□

By summing the equation of Claim 5.18 over $H = \{-1, 1\}$, we obtain that

$$\begin{aligned}&(\xi^3 c \cdot qL, \xi^3 c \cdot qR, \xi^6 c^2 \cdot qP)(1) + (\xi^3 c \cdot qL, \xi^3 c \cdot qR, \xi^6 c^2 \cdot qP)(-1) = \\ &2^{-i+1}\Big(\langle (a_{L,0}, a_{L,1}), (\underline{p_{\mathsf{ckL}}}[r_1, \ldots, r_{i-1}; 0], \underline{p_{\mathsf{ckL}}}[r_1, \ldots, r_{i-1}; 1])\rangle, \\ &\langle (a_{R,0}, a_{R,1}), (\underline{p_{\mathsf{ckR}}}[r_1, \ldots, r_{i-1}; 0], \underline{p_{\mathsf{ckR}}}[r_1, \ldots, r_{i-1}; 1])\rangle, \\ &\langle (a_{L,0}, a_{L,1}), (a_{R,0}, a_{R,1})\rangle \mathsf{ck}_0\Big) \ .\end{aligned}$$

Now consider the vector $\underline{v}_i := \big((a_{L,0}, a_{L,1}), (a_{R,0}, a_{R,1})\big) \in (M_{\mathrm{L}}^{2^{\ell-i+1}})^2$ and its corresponding polynomial $p_{\underline{v}_i} \in M_{\mathrm{L}}^2[X_i, \ldots, X_\ell]$ whose two coordinates are the multilinear polynomials obtained from the two vectors $(a_{L,0}, a_{L,1})$ and $(a_{R,0}, a_{R,1})$ in $M_{\mathrm{L}}^{2^{\ell-i+1}}$. Note that $\underline{v}_i$, and thus $p_{\underline{v}_i}$, has M_{L}-norm at most $N_{\mathsf{SP}} \max_{j\in[4]} \|p_j\|_{M_{\mathrm{L}}}$ and satisfies

$$(\xi^3 c, \xi^3 c, \xi^6 c^2) \cdot (q(1) + q(-1)) = \sum_{\omega_i, \ldots, \omega_\ell \in \{-1,1\}} f_{\mathsf{CM}}\Big(p_{\underline{v}_i}(\omega_i, \ldots, \omega_\ell), p_{\mathsf{ck}}(r_1, \ldots, r_{i-1}, \omega_i, \ldots, \omega_\ell), \xi^3 \cdot c\Big) \ ,$$

which implies that ϕ_{sc} evaluates to 1 as required. Moreover, $p_{\underline{v}_i}$ is ck-admissible and all the derivations to obtain $p_{\underline{v}_i}$, the desired polynomial, are efficient. □

Remark 5.19 (scalar products without scalar-product compatibility)**.** If BM is not scalar-product compatible, we can still define a scalar-product commitment as follows. Sample commitment keys $(\mathsf{ckL}_0, \mathsf{ckL}_1) \in M_{\mathrm{R}}^{n+h}$, $(\mathsf{ckR}_0, \mathsf{ckR}_1) \in M_{\mathrm{L}}^{n+h}$, and $\mathsf{ckH} \in M_{\mathrm{R}}^h$. The message and randomness are $(\mathsf{mL}, \mathsf{mR}) \in M_{\mathrm{L}}^n \times M_{\mathrm{R}}^n$ and $(\rho\mathsf{L}, \rho\mathsf{R}, \rho\mathsf{H}) \in M_{\mathrm{L}}^h \times M_{\mathrm{R}}^h \times M_{\mathrm{L}}^h$. A commitment is computed as three Pedersen commitments:

$$\Big(\langle \mathsf{mL}, \mathsf{ckL}_0\rangle + \langle \rho\mathsf{L}, \mathsf{ckL}_1\rangle, \langle \mathsf{mR}, \mathsf{ckR}_0\rangle + \langle \rho\mathsf{R}, \mathsf{ckR}_1\rangle, \langle \mathsf{mL}, \mathsf{mR}\rangle + \langle \rho\mathsf{H}, \mathsf{ckH}\rangle\Big) \in M_{\mathrm{T}}^3 \ .$$

To prove Lemma 5.17 for this construction we require that BM is secure, and *also* that the bilinear-module generator that outputs the bilinear module $\mathcal{M} = (R, M_{\mathrm{R}}, M_{\mathrm{L}}, M_{\mathrm{T}}, e)$ with M_{L} and M_{R} swapped is secure. The binding property follows from the bilinear-relation assumption for both of the two generators. The hiding property of both bilinear-module generators implies that the commitment scheme is hiding.

5.5 Compressed scalar-product commitment

One can "compress" the three parts of the commitment in Section 5.4 into one. This leads to a commitment scheme that is similar to a Pedersen commitment, except that the last coordinate corresponds to the scalar-product of the "first" and "second" half of the message.

Definition 5.20. *Let* $\mathsf{BM} = (\mathsf{Setup}, \mathsf{KeyGen})$ *be a bilinear-module generator that is scalar-product compatible. The* **compressed scalar-product commitment scheme** *is defined via the following algorithms.*

- $\mathsf{CSP.Setup}(1^\lambda, n)$: *sample* $(\mathcal{M}, h, \mathsf{aux}) \leftarrow \mathsf{BM.Setup}(1^\lambda, 2n+1)$ *and output* $\mathsf{pp} := (\mathcal{M}, h, \mathsf{aux})$.
- $\mathsf{CSP.KeyGen}(\mathsf{pp})$: *sample* $\mathsf{ck} \leftarrow \mathsf{BM.KeyGen}(\mathcal{M}, h, \mathsf{aux})$ *and output*

$$\mathsf{ck} := (\mathsf{ckL}, \mathsf{ckR}, \mathsf{ckP}, \mathsf{ckH}) \in M_{\mathrm{R}}^{2n+1} \times M_{\mathrm{R}}^h \ .$$

- $\mathsf{CSP.Commit}(\mathsf{ck}, \mathsf{m}; \rho)$: *given* $\mathsf{m} = (\mathsf{mL}, \mathsf{mR}) \in M_{\mathrm{L}}^{2n}(B_{\mathsf{C}})$ *and* $\rho \leftarrow \mathcal{U}_{M_{\mathrm{L}}}$*, output*

$$\mathsf{cm} := \langle \mathsf{mL}, \mathsf{ckL}\rangle + \langle \mathsf{mR}, \mathsf{ckR}\rangle + \langle \mathsf{mL}, \mathsf{mR}\rangle \cdot \mathsf{ckP} + \langle \rho, \mathsf{ckH}\rangle \in M_{\mathrm{T}} \ .$$

- $\mathsf{CSP.Open}(\mathsf{ck}, \mathsf{m}, \rho, \mathsf{cm}, c)$: *check that* $\mathsf{m} = (\mathsf{mL}, \mathsf{mR}) \in M_{\mathrm{L}}^{2n}(B_{\mathsf{BRA}})$, $\rho \in M_{\mathrm{L}}^{h}(B_{\mathsf{BRA}})$, *and* $c^2 \cdot \mathsf{cm} = c\langle \mathsf{mL}, \mathsf{ckL}\rangle + c\langle \mathsf{mR}, \mathsf{ckR}\rangle + \langle \mathsf{mL}, \mathsf{mR}\rangle \cdot \mathsf{ckP} + c^2 \cdot \langle \rho, \mathsf{ckH}\rangle$.

Lemma 5.21. *If* BM *is secure then* CSP *is a sumcheck-friendly invertible commitment scheme. In more detail:*

1. *if* BM *satisfies the BRA then* CSP *is binding;*
2. *if h is $(\mathcal{M}, \mathcal{U}_{M_{\mathrm{L}}}, \mathcal{U}_{M_{\mathrm{R}}}, \mathrm{negl}(\lambda))$-hiding then* CSP *is computationally hiding;*
3. CSP *is sumcheck-friendly;*
4. *if the BRA holds and $(\mathcal{C}, \xi, N)$ are pseudoinverse parameters for (R, M_{T}) then* CSP *is $(4, B_{\mathsf{CSP}}, \xi^3, N_{\mathsf{CSP}})$-invertible for*

$$B_{\mathsf{CSP}} := \sqrt{\frac{B_{\mathsf{BRA}}}{n\gamma_R^2 \|\mathcal{C}\|_R^2 \|\xi^6\|_R}} \ , \tag{13}$$

$$N_{\mathsf{CSP}} := 3\gamma_R^2 N^3 \iota(\mathcal{C}, 3)\|\mathcal{C}\|_R \ . \tag{14}$$

The proofs of Items 1 and 2 follow directly from the definition of the BRA and the hiding constant of BM. The proof of Item 3 is similar to the corresponding argument in Lemma 5.17, so we simply list the functions involved. We only provide a proof sketch for the last item, highlighting the main differences in the proof.

Functions for Item 3. Let $H := \{-1, 1\}$ and $\ell := \log n$. For a message $\mathsf{m} = (\mathsf{mL}, \mathsf{mR}) \in M_{\mathrm{L}}^{2n}$, we define $p_{\mathsf{m}} := (p_{\mathsf{mL}}, p_{\mathsf{mR}}) \in M_{\mathrm{L}}^2[X_1, \ldots, X_\ell]$ where $p_{\mathsf{mL}}, p_{\mathsf{mR}}$ are the multilinear polynomials respectively obtained from $\mathsf{mL}, \mathsf{mR} \in M_{\mathrm{L}}^n$ via Definition 3.5. For a key $(\mathsf{ckL}, \mathsf{ckR}, \mathsf{ckP}, \mathsf{ckH})$, we define $p_{\mathsf{ck}} := (p_{\mathsf{ckL}}, p_{\mathsf{ckR}}, p_{\mathsf{ckP}}) \in M_{\mathrm{L}}^3[X_1, \ldots, X_\ell]$ where $p_{\mathsf{ckL}}, p_{\mathsf{ckR}}$ are the multilinear polynomials respectively obtained from $\mathsf{ckL}_0, \mathsf{ckR}_0 \in M_{\mathrm{R}}^n$ via Definition 3.5 and ckP is a degenerate 0-degree polynomial equal to the constant $\mathsf{ckP} \in M_{\mathrm{R}}$. We define

- $f_{\mathsf{CM}} \colon M_{\mathrm{L}}^2 \times M_{\mathrm{R}}^3 \times R \to M_{\mathrm{T}}$ as $f_{\mathsf{CM}}((a_{\mathrm{L}}, a_{\mathrm{R}}), (\mathsf{G}_{\mathrm{L}}, \mathsf{G}_{\mathrm{R}}, \mathsf{G}_{SC}), c) := 2^{-\ell} c\, a_{\mathrm{L}} \cdot \mathsf{G}_{\mathrm{L}} + c\, a_{\mathrm{R}} \cdot \mathsf{G}_{\mathrm{R}} + a_{\mathrm{L}} a_{\mathrm{R}} \mathsf{G}_{SC}$;
- $g_{\mathsf{CM}} \colon M_{\mathrm{R}}^{2n+1+h} \times M_{\mathrm{L}}^h \to M_{\mathrm{T}}^3$ as $g_{\mathsf{CM}}(\mathsf{ck}, \rho) := \langle \rho\mathsf{H}, \mathsf{ckH}\rangle$;
- $\phi_{\mathrm{sc}} \colon M_{\mathrm{T}} \times M_{\mathrm{T}} \times R \to \{0, 1\}$ as $\phi_{\mathrm{sc}}(\mathsf{cm}, \mathsf{cm}', c) := \mathbb{I}[c^2 \cdot \mathsf{cm} \stackrel{?}{=} \mathsf{cm}']$, which checks equality when $c = 1$.
- $\alpha_{\mathrm{sc}} \equiv 1$.

□

Proof sketch for Item 4. The proof follows similar steps to the proof of Lemma 5.17. We sketch the differences between the two proofs.

First, we find $a_{L,0}, a_{L,1}, a_{L,2}, b_{L,0}, b_{L,1}, b_{L,2} \in M_{\mathrm{L}}^{2^{\ell-i}}$, $a_{R,0}, a_{R,1}, a_{R,2}, b_{R,0}, b_{R,1}, b_{R,2} \in M_{\mathrm{L}}^{2^{\ell-i}}$ with M_{L}-norm at most $3\gamma_R^2 N^3 \iota(\mathcal{C}, 3)\|\mathcal{C}\|_R \max_{j\in[4]} \|p_j\|_{M_{\mathrm{L}}} = N_{\mathsf{CSP}} \max_{j\in[4]} \|p_j\|_{M_{\mathrm{L}}}$ and $a_{\bullet,0}, a_{\bullet,1}, a_{\bullet,2} \in M_{\mathrm{L}}^{2^{\ell-i}}$ with M_{L}-norm at most $2^{\ell-i} N_{\mathsf{CSP}} (\max_{j\in[4]} \|p_j\|_{M_{\mathrm{L}}})^2$ such that

$$\begin{aligned}
&\xi^6 c^2 \cdot q(X) \\
&= 2^{-i}\Big(\xi^3 c(\langle a_{L,0} + a_{L,1}X + a_{L,2}X^2, \underline{p_{\mathsf{ckL}}}[r_1, \ldots, r_{i-1}; 0]\rangle + \langle b_{L,0} + b_{L,1}X + b_{L,2}X^2, \underline{p_{\mathsf{ckL}}}[r_1, \ldots, r_{i-1}; 1]\rangle), \\
&\qquad \xi^3 c(\langle a_{R,0} + a_{R,1}X + a_{R,2}X^2, \underline{p_{\mathsf{ckR}}}[r_1, \ldots, r_{i-1}; 0]\rangle + \langle b_{R,0} + b_{R,1}X + b_{R,2}X^2, \underline{p_{\mathsf{ckR}}}[r_1, \ldots, r_{i-1}; 1]\rangle),
\end{aligned} \tag{15}$$

$$(a_{\bullet,0} + a_{\bullet,1}X + a_{\bullet,2}X^2) \cdot \mathsf{ckP}\Big) \ .$$

Next, we show that the following holds.

Claim 5.22.

$$\begin{aligned}
&\xi^6 c^2 \cdot q(X) \\
&= 2^{-i}\Big(\xi^3 c(\langle a_{L,0} + a_{L,1}X, \underline{p_{\mathsf{ckL}}}[r_1, \ldots, r_{i-1}; 0]\rangle + \langle (a_{L,0} + a_{L,1}X)X, \underline{p_{\mathsf{ckL}}}[r_1, \ldots, r_{i-1}; 1]\rangle) + \\
&\quad \xi^3 c(\langle a_{R,0} + a_{R,1}X, \underline{p_{\mathsf{ckR}}}[r_1, \ldots, r_{i-1}; 0]\rangle + \langle (a_{R,0} + a_{R,1}X)X, \underline{p_{\mathsf{ckR}}}[r_1, \ldots, r_{i-1}; 1]\rangle) + \\
&\quad (\langle a_{L,0} + a_{L,1}X, a_{R,0} + a_{R,1}X\rangle) \cdot \mathsf{ckP}\Big) \ .
\end{aligned}$$

Proof. The claimed equation follows from Equation (15) and the following equalities.

- $a_{L,0} = b_{L,1}, a_{L,1} = b_{L,2}, a_{L,2} = b_{L,0} = 0, a_{R,0} = b_{R,1}, a_{R,1} = b_{R,2}, a_{R,2} = b_{R,0} = 0$. Unlike the corresponding proof of Lemma 5.17, we argue about all coefficients of the polynomials $a_{L,0} + a_{L,1}X + a_{L,2}X^2$, $b_{L,0} + b_{L,1}X + b_{L,2}X^2$, $a_{R,0} + a_{R,1}X + a_{R,2}X^2$, and $b_{R,0} + b_{R,1}X + b_{R,2}X^2$ simultaneously. For each $j \in [4]$, we define messages $m_1^{(j)} := (m_{1,L}^{(j)}, m_{1,R}^{(j)}, m_{1,\bullet}^{(j)})$ and $m_2^{(j)} := (m_{2,L}^{(j)}, m_{2,R}^{(j)}, m_{2,\bullet}^{(j)})$ such that

$$\begin{aligned}
m_{1,L}^{(j)} &:= (a_{L,0} + a_{L,1}r_i^{(j)} + a_{L,2}{r_i^{(j)}}^2, b_{L,0} + b_{L,1}r_i^{(j)} + b_{L,2}{r_i^{(j)}}^2) \ , \\
m_{1,R}^{(j)} &:= (a_{R,0} + a_{R,1}r_i^{(j)} + a_{R,2}{r_i^{(j)}}^2, b_{R,0} + b_{R,1}r_i^{(j)} + b_{R,2}{r_i^{(j)}}^2) \ , \\
m_{1,\bullet}^{(j)} &:= a_{\bullet,0} + a_{\bullet,1}r_i^{(j)} + a_{\bullet,2}{r_i^{(j)}}^2 \ , \\
m_{2,L}^{(j)} &:= \xi^3(\underline{p}_{j,L}, r_i^{(j)}\underline{p}_{j,L}) \ , \\
m_{2,R}^{(j)} &:= \xi^3(\underline{p}_{j,R}, r_i^{(j)}\underline{p}_{j,R}) \ , \\
m_{2,\bullet}^{(j)} &:= \xi^6\langle(\underline{p}_{j,L}, r_i^{(j)}\underline{p}_{j,L}), (\underline{p}_{j,R}, r_i^{(j)}\underline{p}_{j,R})\rangle \ .
\end{aligned}$$

  One can verify that $\|m_1^{(j)}\|_{M_{\mathrm{L}}}, \|m_2^{(j)}\|_{M_{\mathrm{L}}} \leq n\gamma_R^2\|\mathcal{C}\|_R^2\|\xi^6\|_R B_{\mathsf{CSP}}^2 \leq B_{\mathsf{BRA}}$, so the BRA implies that $m_1^{(j)} = m_2^{(j)}$.

- $a_{\bullet,0} + a_{\bullet,1}X + a_{\bullet,2}X^2 = \langle a_{L,0} + a_{L,1}X, a_{R,0} + a_{R,1}X\rangle$. From the last coordinate of $m_1^{(j)}$ and $m_2^{(j)}$,

$$a_{\bullet,0} + a_{\bullet,1}r_i^{(j)} + a_{\bullet,2}{r_i^{(j)}}^2 = \xi^6\langle(\underline{p}_{j,L}, r_i^{(j)}\underline{p}_{j,L}), (\underline{p}_{j,R}, r_i^{(j)}\underline{p}_{j,R})\rangle. \tag{16}$$

  Similarly, we argue that for each $j \in [4]$

$$\begin{aligned}
\xi^3\underline{p}_{j,L} &= a_{L,0} + a_{L,1}r_i^{(j)} \ , \\
\xi^3\underline{p}_{j,R} &= a_{R,0} + a_{R,1}r_i^{(j)} \ , \\
\xi^6\langle\underline{p}_{j,L}, \underline{p}_{j,R}\rangle &= \langle a_{L,0} + a_{L,1}r_i^{(j)}, a_{R,0} + a_{R,1}r_i^{(j)}\rangle \ .
\end{aligned}$$

  Combining the above with Equation (16) shows that $a_{\bullet,0}+a_{\bullet,1}X+a_{\bullet,2}X^2 = \langle a_{L,0}+a_{L,1}X, a_{R,0}+a_{R,1}X\rangle$.

□

By summing the equation of Claim 5.22 over $H = \{-1, 1\}$, we obtain that

$$\begin{aligned}
&\xi^6 \cdot c^2(q(1) + q(-1)) \\
&= 2^{-i+1}\Big(\langle \xi^3 \cdot c\,(a_{L,0}, a_{L,1}), (\underline{p_{\mathsf{ckL}}}[r_1, \ldots, r_{i-1}; 0], \underline{p_{\mathsf{ckL}}}[r_1, \ldots, r_{i-1}; 1])\rangle \\
&\qquad + \langle \xi^3 \cdot c\,(a_{R,0}, a_{R,1}), (\underline{p_{\mathsf{ckL}}}[r_1, \ldots, r_{i-1}; 0], \underline{p_{\mathsf{ckL}}}[r_1, \ldots, r_{i-1}; 1])\rangle \\
&\qquad + \langle (a_{L,0}, a_{L,1}), (a_{R,0}, a_{R,1})\rangle \mathsf{ckP}\Big) \ .
\end{aligned}$$

Now consider the vector $\underline{v}_i := \big((a_{L,0}, a_{L,1}), (a_{R,0}, a_{R,1})\big) \in (M_{\mathrm{L}}^{2^{\ell-i+1}})^2$ and its corresponding polynomial $p_{\underline{v}_i} \in M_{\mathrm{L}}^2[X_i, \ldots, X_\ell]$ whose two coordinates are the multilinear polynomials obtained from the two vectors $(a_{L,0}, a_{L,1})$ and $(a_{R,0}, a_{R,1})$ in $M_{\mathrm{L}}^{2^{\ell-i+1}}$. Note that $\underline{v}_i$, and thus $p_{\underline{v}_i}$, has M_{L}-norm at most $N_{\mathsf{SP}} \max_{j\in[4]} \|p_j\|_{M_{\mathrm{L}}}$ and satisfies the following equation:

$$\xi^6 \cdot c^2 \cdot (q(1) + q(-1)) = \sum_{\omega_i, \ldots, \omega_\ell \in \{-1,1\}} f_{\mathsf{CM}}\Big(p_{\underline{v}_i}(\omega_i, \ldots, \omega_\ell), p_{\mathsf{ck}}(r_1, \ldots, r_{i-1}, \omega_i, \ldots, \omega_\ell), \xi^3 \cdot c\Big) \ ,$$

which implies that ϕ_{sc} evaluates to 1 as required. Moreover, $p_{\underline{v}_i}$ is ck-admissible and all the derivations to obtain $p_{\underline{v}_i}$, the desired polynomial, are efficient. □

5.6 Instantiations of bilinear-module generators

We describe instantiations of bilinear-module generators $\mathsf{BM} = (\mathsf{Setup}, \mathsf{KeyGen})$ for several cryptographic settings; in each case the BRA follows from corresponding well-known cryptographic assumptions (DL assumption, SIS assumption, and so on). For each setting, we specify the algorithms $\mathsf{BM.Setup}$ and $\mathsf{BM.KeyGen}$, give the corresponding output parameters in Figure 2, and argue the required properties of BM.

Discrete logarithms. The algorithm $\mathsf{BM.Setup}(1^\lambda, n)$ samples a group $\mathbb{G}$ of prime order $q = \exp(\lambda)$ and outputs $(\mathcal{M}, h, \mathsf{aux})$ as in Figure 2. The algorithm $\mathsf{BM.KeyGen}(\mathcal{M}, h, \mathsf{aux})$ samples a uniformly random element $\underline{\mathsf{G}} \in M_{\mathrm{R}}^{n+h} = \mathbb{G}^{n+h}$. Since $M_{\mathrm{L}} = \mathbb{F}_q$ is a ring, BM is scalar-product compatible.

Moreover BM is protocol-friendly. First, it is secure:

- Assuming the hardness of the discrete logarithm problem in $\mathbb{G}$, the BRA holds for $\mathsf{BM} = (\mathsf{Setup}, \mathsf{KeyGen})$.
- The integer $h = 1$ is $(\mathcal{M}, \mathcal{U}_{M_{\mathrm{L}}}, \mathcal{U}_{M_{\mathrm{R}}}, \epsilon)$-hiding where $\mathcal{U}_{M_{\mathrm{L}}}$ is the uniform distribution on $M_{\mathrm{L}} = \mathbb{F}_q$, $\mathcal{U}_{M_{\mathrm{R}}}$ the uniform distribution on $M_{\mathrm{R}}^{n+1} = \mathbb{G}^{n+1}$, and $\epsilon = 0$.
- $(\mathcal{C}, \xi, N) = (\mathbb{F}_q, 1, 1)$ are pseudoinverse parameters for $(R, M_{\mathrm{T}}) = (\mathbb{F}_q, \mathbb{G})$.

Next, BM is masking-friendly for $\kappa = \infty$. For any B with $1 \le B \le \infty$, $M_{\mathrm{L}}^n(B) = \mathbb{F}_q^n$, and hence, for any $\underline{a} \in \mathbb{F}_q^n$ and $\underline{b} \leftarrow \mathbb{F}_q^n$, $\underline{a} + \underline{b}$ is distributed uniformly in $\mathbb{F}_q$; the rejection probability is 0.

Finally, we investigate choices of I for which BM is quotient-friendly. Since $M_{\mathrm{L}} = \mathbb{F}_q$ is a prime-order field, its only submodules are $I = \{0\}$ and $I = \mathbb{F}_q$. If $I = \mathbb{F}_q$ then ξ is 0 in M_{L}/I, so $I = \{0\}$ is the only ideal for which BM is quotient-friendly (in which case $M_{\mathrm{L}}/I = \mathbb{F}_q$ and so $\xi = 1$ is invertible).

Instantiating the commitment schemes in the previous sections with the above choice of BM gives $N_{\mathsf{Ped}} = 1$, $N_{\mathsf{LF}} = 1$, $B_{\mathsf{SP}} = \infty$, $N_{\mathsf{SP}} = 1$, $B_{\mathsf{CSP}} = \infty$, and $N_{\mathsf{CSP}} = 1$ (since we consider the trivial norm for R and M_{L} all constants greater than 0 can be thought as equal to 1). Additionally, $B_{\mathsf{C}} = 1$, $\gamma_R = 1$, and $\|\mathcal{C}\|_R = 1$ and in all settings $d_{\mathsf{ck}} = 1$, so we can apply Lemma 4.13 to the corresponding commitment schemes for $\mathcal{R}_{\mathrm{SC}}(1, 1)$.

	DL	pairings	GUO	ideal lattices
R	$\mathbb{F}_q$	$\mathbb{F}_q$	$\mathbb{Z}$	$\mathbb{Z}[X]/\langle X^d+1\rangle$
M_{L}	$\mathbb{F}_q$	$\mathbb{G}_1$	$\mathbb{Z}$	$\mathbb{Z}[X]/\langle X^d+1\rangle$
M_{R}	$\mathbb{G}$	$\mathbb{G}_2$	$\mathbb{G}$	$\big(\mathbb{Z}_q[X]/\langle X^d+1\rangle\big)^r$
M_{T}	$\mathbb{G}$	$\mathbb{G}_{\mathrm{T}}$	$\mathbb{G}$	$\big(\mathbb{Z}_q[X]/\langle X^d+1\rangle\big)^r$
e	group exp	bilinear map	group exp	poly multiplication mod q and X^d+1
R-norm	trivial †	trivial †	absolute value	ℓ_∞
M_{L}-norm	trivial †	trivial †	absolute value	ℓ_∞
h	1	1	$\log \frac{2^\lambda\lvert\mathbb{G}\rvert}{B_{\mathsf{C}}}$	$2r\log\frac{q}{B_{\mathsf{C}}}$
B_{BRA}	∞	∞	$B_{\mathsf{RSA}}=\frac{q-1}{2}$, $B_{\mathsf{CL}}=\frac{q-1}{4}$ $\star$	B_{SIS}
$\mathcal{U}_{M_{\mathrm{L}}}$	uniform over $\mathbb{G}$	uniform over $\mathbb{G}_1$	uniform over $(-B,B)\cap\mathbb{Z}$ where $B_{\mathsf{C}}\le B\le B_{\mathsf{BRA}}$	uniform over $M_{\mathrm{L}}(B)$ where $B_{\mathsf{C}}\le B\le B_{\mathsf{BRA}}$
$\mathcal{C}$	$\mathbb{F}_q$	$\mathbb{F}_q$	$(-\frac{p-1}{2},\frac{p-1}{2})\cap\mathbb{Z}$	$\{X^i\in\mathbb{Z}[X]/\langle X^d+1\rangle : 0\le i\le 2d-1\}$
ξ	1	1	$\mathrm{lcm}([p-1])$	2
N	1	1	$\mathrm{lcm}([p-1])$	d
B_{C}	1	1	B	B
κ	∞	∞	$O(2^\lambda)$	$O(dn)$
I	$\{0\}$	$\{0\}$	$I=n\mathbb{Z}$ for $n\in\mathbb{Z}$ with prime factors $\ge p$	$I=n\mathbb{Z}$ for odd $n\ne -1,1$ ‡

Figure 2: Output $(\mathcal{M},h,\mathsf{aux})$ of a bilinear-module generator in the different cryptographic settings, where $\mathcal{M}=(R,M_{\mathrm{L}},M_{\mathrm{R}},M_{\mathrm{T}},e)$ and $\mathsf{aux}=(B_{\mathsf{BRA}},\mathcal{U}_{M_{\mathrm{L}}},\mathcal{C},\xi,N,B_{\mathsf{C}},\kappa,I)$. (†: Equals 1 for any non-zero element of R or M_{L} and equals 0 otherwise. $\star$: The difference in B_{BRA} between RSA groups and class groups is related to the fact that computing square roots is easy in class groups. ‡: There are more choices of I related to Dedekind's Factorization Criterion as discussed further below.)

Pairings. The algorithm $\mathsf{BM.Setup}(1^\lambda,n)$ samples groups $\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_{\mathrm{T}}$ of prime order $q=\exp(\lambda)$ equipped with a pairing $e\colon \mathbb{G}_1\times\mathbb{G}_2\to\mathbb{G}_{\mathrm{T}}$ and outputs $(\mathcal{M},h,\mathsf{aux})$ as in Figure 2. The algorithm $\mathsf{BM.KeyGen}(\mathcal{M},h,\mathsf{aux})$ samples a uniformly random element $\underline{\mathsf{G}}\in M_{\mathrm{R}}^{n+h}=\mathbb{G}_2^{n+h}$.

Moreover BM is protocol-friendly. First, it is secure:

- Assuming the double-pairing assumption,[7] the BRA holds for $\mathsf{BM}=(\mathsf{Setup},\mathsf{KeyGen})$.
- The integer $h=1$ is $(\mathcal{M},\mathcal{U}_{M_{\mathrm{L}}},\mathcal{U}_{M_{\mathrm{R}}},\epsilon)$-hiding where $\mathcal{U}_{M_{\mathrm{L}}}$ is the uniform distribution on $M_{\mathrm{L}}=\mathbb{F}_q$, $\mathcal{U}_{M_{\mathrm{R}}}$ the uniform distribution on $M_{\mathrm{R}}^{n+1}=\mathbb{G}_2^{n+1}$, and $\epsilon=0$.
- $(\mathcal{C},\xi,N)=(\mathbb{F}_q,1,1)$ are pseudoinverse parameters for $(R,M_{\mathrm{T}})=(\mathbb{F}_q,\mathbb{G}_{\mathrm{T}})$.

Next, BM is masking-friendly for $\kappa=\infty$. For any B with $1\le B\le\infty$, $M_{\mathrm{L}}^n(B)=\mathbb{G}_1^n$, and hence, for any $\underline{a}\in\mathbb{G}_1^n$ and $\underline{b}\leftarrow\mathbb{G}_1^n$, $\underline{a}+\underline{b}$ is distributed uniformly in $\mathbb{G}_1$; the rejection probability is 0.

Finally, we investigate choices of I for which BM is quotient-friendly. Since $M_{\mathrm{L}}=\mathbb{G}_1$ (a vector space of dimension 1 over a field of prime order), its only submodules are $I=\{0\}$ and $I=\mathbb{G}_1$. If $I=\mathbb{G}_1$ then ξ is 0 in M_{L}/I, so $I=\{0\}$ is the only ideal for which BM is quotient-friendly (in which case $M_{\mathrm{L}}/I=\mathbb{G}_1$ and multiplication by $\xi=1$ is invertible).

Note that we could also have chosen $M_{\mathrm{L}}:=\mathbb{G}_2$ and $M_{\mathrm{R}}:=\mathbb{G}_1$, and then switched the inputs of e.

BM described above is *not* scalar-product compatible, because M_{L} is merely a group but not a ring. Nevertheless one can still construct commitment schemes for scalar products, as described in Remark 5.19.

[7] This assumption is implied by the decisional Diffie–Hellman assumption in $\mathbb{G}_2$, which in turn implies binding for the commitment scheme in [AFGHO16]

Instantiating the commitment schemes in the previous sections with the above choice of BM gives $N_{\mathsf{Ped}} = 1$, $N_{\mathsf{LF}} = 1$, $B_{\mathsf{SP}} = \infty$ and $N_{\mathsf{SP}} = 1$. Note that we cannot instantiate the commitment scheme CSP, since BM is not scalar-product compatible. Additionally, $B_{\mathsf{C}} = 1$, $\gamma_R = 1$, and $\|\mathcal{C}\|_R = 1$ and in all settings $d_{\mathsf{ck}} = 1$, so we can apply Lemma 4.13 to the corresponding commitment schemes for $\mathcal{R}_{\mathrm{SC}}(1,1)$.

Groups of unknown order. There are two instantiations of groups of unknown order (which are related to the commitment schemes in [BFS20]): RSA groups and class groups of an imaginary quadratic order.

The algorithm $\mathsf{BM.Setup}(1^\lambda, n)$ samples a group of unknown order $\mathbb{G}$ and primes $q, p > 2$ and outputs $(\mathcal{M}, h, \mathsf{aux})$ as in Figure 2. Observe that B_{BRA} differs in the RSA and class group settings (and, in the RSA case, the primes q, p are unrelated to the primes that determine the order of $\mathbb{G}$). The algorithm $\mathsf{BM.KeyGen}(\mathcal{M}, h, \mathsf{aux})$ samples a random element $\mathsf{G} \in M_{\mathrm{R}} = \mathbb{G}$, computes $\underline{\mathsf{G}}_0 = (q^h\mathsf{G}, q^{h+1}\mathsf{G}, \ldots, q^{n+h-1}\mathsf{G})$, $\underline{\mathsf{G}}_1 = (\mathsf{G}, \ldots, q^{h-1}\mathsf{G})$, and outputs the vector $(\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1) \in M_{\mathrm{R}}^{n+h} = \mathbb{G}^{n+h}$. Since $M_{\mathrm{L}} = \mathbb{Z}$ is a ring, BM is scalar-product compatible.

Moreover BM is protocol-friendly. First, it is secure:

- Assuming the Order Assumption [BFS20] for the sampled group $\mathbb{G}$, the BRA holds for $\mathsf{BM} = (\mathsf{Setup}, \mathsf{KeyGen})$. The Order Assumption is implied by the Adaptive Root Assumption [BBBPWM18; Wes19].
- Any integer $h \geq \log \frac{2^\lambda |\mathbb{G}|}{B_{\mathsf{C}}}$ is $(\mathcal{M}, \mathcal{U}_{M_{\mathrm{L}}}, \mathcal{U}_{M_{\mathrm{R}}}, \epsilon)$-hiding where $\mathcal{U}_{M_{\mathrm{L}}}$ is the uniform distribution over $((-B, B) \cap \mathbb{Z})^h$ for any B such that $B_{\mathsf{C}} \leq B \leq B_{\mathsf{BRA}}$, $\mathcal{U}_{M_{\mathrm{R}}}$ samples a random element $\mathsf{G} \in \mathbb{G}$ and outputs $(q^h\mathsf{G}, \ldots, q^{n+h-1}\mathsf{G}, \ldots, \mathsf{G}, q^{h-1}\mathsf{G})$ (this matches the output of $\mathsf{BM.KeyGen}$), and $\epsilon = \mathrm{negl}(\lambda)$. (See [BFS20; BDFG20] for the proof.)
- $(\mathcal{C}, \xi, N) = ((-\frac{p-1}{2}, \frac{p-1}{2}) \cap \mathbb{Z}, \mathrm{lcm}([p-1]), \mathrm{lcm}([p-1]))$ are pseudoinverse parameters for $(R, M_{\mathrm{T}}) = (\mathbb{Z}, \mathbb{G})$. If $(c_1 - c_2)m = a \cdot m^*$ for distinct challenges $c_1, c_2 \in \mathcal{C}$, then $(c_1 - c_2) \in (-(p-1), p-1) \cup \mathbb{Z}$, so we can set $r = \mathrm{lcm}([p-1])/(c_1 - c_2)$.

Next, if $q = O(2^\lambda)$, then BM is masking-friendly for $\kappa = O(2^\lambda)$. For every $B \in \mathbb{Z}$ with $B_{\mathsf{C}} \leq B \leq B_{\mathsf{BRA}}/\kappa$ and $\underline{a} \in \mathbb{Z}^n(B)$, if $\underline{b} \leftarrow \mathbb{Z}^n(\kappa B)$, then $\underline{a} + \underline{b}$ has norm more than $(\kappa - 1)B$ with probability less than $n\frac{2B}{2\kappa B} = \frac{n}{\kappa} = \mathrm{negl}(\lambda)$. Also, conditioned that the masking procedure does not output $\perp$, the distribution of $\underline{a} + \underline{b}$ is uniform in $\mathbb{Z}^n((\kappa - 1)B)$; the rejection probability is $\mathrm{negl}(\lambda)$.

Finally, we investigate choices of I for which BM is quotient-friendly. Since $M_{\mathrm{L}} = \mathbb{Z}$ (a ring), its submodules are the ideals $I = n\mathbb{Z}$ for $n \in \mathbb{N} \cup \{0\}$. As with previous examples, it is clear that $I = M_{\mathrm{L}}$ (when $n = 1$) does not lead to a quotient-friendly bilinear module. We have $\xi = \mathrm{lcm}([p-1])$, so it is clear that ξ is invertible modulo $M_{\mathrm{L}}/I = \mathbb{Z}/n\mathbb{Z}$ if and only if all prime factors of n are at least p. This means that M_{L} is quotient friendly if and only if $I = n\mathbb{Z}$ for $n \in \mathbb{Z}$ whose prime factors are all at least p.

Instantiating the commitment schemes in the previous sections with BM and observing that $\iota(\mathcal{C}, 3) = 2\gamma_R^2 \|\mathcal{C}\|_R^3 = 2(\frac{p-1}{2})^3 = \frac{(p-1)^3}{4}$ gives

$$
\begin{aligned}
N_{\mathsf{Ped}} &= 6\gamma_R^2 N^3 \iota(\mathcal{C}, 3)\|\mathcal{C}\|_R = 3/4 \cdot (p-1)^4 \cdot \mathrm{lcm}([p-1])^3 \ , \\
N_{\mathsf{LF}} &= 6\gamma_R^2 N^3 \iota(\mathcal{C}, 3)\|\mathcal{C}\|_R = 3/4 \cdot (p-1)^4 \cdot \mathrm{lcm}([p-1])^3 \ , \\
B_{\mathsf{SP}} &= \frac{B_{\mathsf{BRA}}}{3\gamma_R^2 \|\mathcal{C}\|_R^2 \|\xi^3\|_R} = \frac{4B_{\mathsf{BRA}}}{3(p-1)^2 \mathrm{lcm}([p-1])^3} \ , \\
N_{\mathsf{SP}} &= 3\gamma_R^2 B_{\mathsf{BRA}}^3 \iota(\mathcal{C}, 3)\|\mathcal{C}\|_R = 3/8 \cdot (p-1)^4 \cdot \mathrm{lcm}([p-1])^3 \ , \\
B_{\mathsf{CSP}} &= \sqrt{\frac{B_{\mathsf{BRA}}}{n\gamma_R^2 \|\mathcal{C}\|_R^2 \|\xi^6\|_R}} = \frac{2\sqrt{B_{\mathsf{BRA}}}}{\sqrt{n}(p-1)\mathrm{lcm}([p-1])^3} \ , \\
N_{\mathsf{CSP}} &= 3\gamma_R^2 N^3 \iota(\mathcal{C}, 3)\|\mathcal{C}\|_R = 3/8 \cdot (p-1)^4 \cdot \mathrm{lcm}([p-1])^3 \ .
\end{aligned}
$$

Additionally, $B_{\mathsf{C}} = B$, $\gamma_R = 1$ and $\|\mathcal{C}\|_R = \frac{p-1}{2}$, so we can apply Lemma 4.13 as follows:

- to the Pedersen commitment scheme for $\mathcal{R}_{\mathrm{SC}}(\mathrm{lcm}([p-1])^\ell, B' = B \cdot (3/4)^\ell \cdot (p-1)^{5\ell} \cdot \mathrm{lcm}([p-1])^{3\ell})$,
- to the linear-function commitment scheme for $\mathcal{R}_{\mathrm{SC}}(\mathrm{lcm}([p-1])^\ell, B' = B \cdot (3/4)^\ell \cdot (p-1)^{5\ell} \cdot \mathrm{lcm}([p-1])^{3\ell})$,
- to the scalar-product commitment scheme for $\mathcal{R}_{\mathrm{SC}}(\mathrm{lcm}([p-1])^\ell, B' = B \cdot (3/8)^\ell \cdot (p-1)^{5\ell} \cdot \mathrm{lcm}([p-1])^{3\ell})$ if
$$B' \leq B_{\mathsf{SP}} \implies 3B \cdot (3/8)^\ell \cdot (p-1)^{5\ell+2} \cdot \mathrm{lcm}([p-1])^{3\ell+3} \leq 4B_{\mathsf{BRA}} \ ,$$
- to the compressed scalar-product commitment scheme for $\mathcal{R}_{\mathrm{SC}}(\mathrm{lcm}([p-1])^\ell, B' = B \cdot (3/8)^\ell \cdot (p-1)^{5\ell} \cdot \mathrm{lcm}([p-1])^{3\ell})$ if
$$B' \leq B_{\mathsf{CSP}} \implies nB^2(3/8)^{2\ell}(p-1)^{10\ell+2}\mathrm{lcm}([p-1])^{6\ell+6} \leq 4B_{\mathsf{BRA}} \ .$$

Remark 5.23. The values ξ, N have size $e^{O(p)}$. This means that a polynomial-sized challenge space leads to exponentially-large slackness. Reducing the size of the slackness will directly lead to asymptotic performance improvements by allowing a smaller choice of q, or a larger challenge space. Prior work [BFS20] in this setting used exponential challenge spaces and $\xi = 1$, but the security argument was later found to be flawed [BHRRS21]. This latter works repairs the argument by using a different configuration of rings and modules with $\xi = 1$, but the argument requires amortization over multiple committed values. It is plausible that a more refined security analysis of our approach would lead to much better slackness.

Lattices. The algorithm $\mathsf{BM.Setup}(1^\lambda, n)$ samples a prime number q, an integer d that is a power of 2, and norm bounds $B_{\mathsf{SIS}}, B \in \mathbb{Z}$, and outputs $(\mathcal{M}, h, \mathsf{aux})$ as in Figure 2; note that the challenge set $\mathcal{C}$ consists of $2d$ elements. The algorithm $\mathsf{BM.KeyGen}(\mathcal{M}, h, \mathsf{aux})$ outputs a uniformly random vector in $M_{\mathrm{R}}^{n+h} = \left(\mathbb{Z}_q[X]/\langle X^d+1\rangle\right)^{r\times(n+h)}$. Since $M_{\mathrm{L}} = \mathbb{Z}[X]/\langle X^d+1\rangle$ is a ring, BM is scalar-product compatible.

Moreover BM is protocol-friendly. First, it is secure:

- Assuming the hardness of the SIS assumption for norm bound B_{SIS}, the BRA holds for $\mathsf{BM} = (\mathsf{Setup}, \mathsf{KeyGen})$. In turn, for the SIS assumption to hold, B_{SIS} should be less than $\min\{q, 2^{2\sqrt{r\log q \log \delta}}\}$ [GN08] (δ is related to the optimal block size in the BKZ algorithm applied to the SIS problem and is typically set to $\delta \approx 1.005$).
- Any integer $h \geq 2r\log\frac{q}{B_{\mathsf{C}}}$ is $(\mathcal{M}, \mathcal{U}_{M_{\mathrm{L}}}, \mathcal{U}_{M_{\mathrm{R}}}, \epsilon)$-hiding where $\mathcal{U}_{M_{\mathrm{L}}}$ is the uniform distribution over $M_{\mathrm{L}}(B)$ for any B such that $B_{\mathsf{C}} \leq B \leq B_{\mathsf{BRA}}$, $\mathcal{U}_{M_{\mathrm{R}}}$ is the uniform distribution on $\left(\mathbb{Z}_q[X]/\langle X^d+1\rangle\right)^{r\times(n+h)}$, and $\epsilon = \mathrm{negl}(\lambda)$. (This holds by [Mic07; SSTX09].) We note that, alternatively, $\mathcal{U}_{M_{\mathrm{L}}}$ could also be a discrete Gaussian [LPR10; SS11].
- The lemma below states that the difference of any two elements in the challenge space has a short "pseudoinverse", and implies that $(\mathcal{C}, \xi, N) = (\{X^i\}_{i\in\{0,\ldots,2d-1\}}, 2, d)$ are pseudoinverse parameters for $(R, M_{\mathrm{T}}) = (\mathbb{Z}[X]/\langle X^d+1\rangle, \left(\mathbb{Z}_q[X]/\langle X^d+1\rangle\right)^r)$.

Lemma 5.24 ([BCKLN14, Lemma 3.1])**.** *Fix $i, j \in \{0, \ldots, 2d-1\}$ with $i < j$. In the ring $\mathbb{Z}[X]/(X^d+1)$, there exists an efficiently computable $r \in R$ with coefficients in $\{-1, 0, 1\}$ such that $r \cdot (X^i - X^j) = 2$.*

Hence, if $(c_1 - c_2)m = a \cdot m^*$ for distinct challenges $c_1, c_2 \in \mathcal{C}$, then there exists an $r \in R$ such that $2m = r \cdot a \cdot m^*$, and $\|r \cdot a\|_R \leq \gamma_R \|r\|_R \|a\|_R = d\|a\|_R$.

Furthermore, BM is masking-friendly with $\kappa = O(dn)$. For every $B \in \mathbb{Z}$ with $B_{\mathsf{C}} \leq B \leq B_{\mathsf{BRA}}/\kappa$ and $\underline{a} \in (\mathbb{Z}[X]/\langle X^d + 1\rangle)^n(B)$, if $\underline{b} \leftarrow (\mathbb{Z}[X]/\langle X^d + 1\rangle)^n(\kappa B)$, then $\underline{a} + \underline{b}$ has norm more than $(\kappa - 1)B$ with probability less than $nd\frac{2B}{2\kappa B} = \frac{nd}{\kappa} = O(1)$. Also, conditioned that the masking procedure does not output $\perp$, the distribution of $\underline{a} + \underline{b}$ is uniform in $(\mathbb{Z}[X]/\langle X^d + 1\rangle)^n((\kappa - 1)B)$.

Finally, we investigate choices of I for which BM is quotient-friendly. Since $M_{\mathrm{L}} = \mathbb{Z}[X]/\langle X^d + 1\rangle$, a ring, its submodules are its ideals. In this setting, $\xi = 2$, and it is clear that setting $I = nM_{\mathrm{L}}$ for even n or $n \in \{-1, 1\}$ does not give a quotient-friendly bilinear module, since in these cases, multiplication by ξ will not be invertible in M_{L}/I. Otherwise, if n is odd, 2 is invertible in M_{L}/I, and BM is quotient friendly.

These are not the only ideals in M_{L}. According to Dedekind's Factorization Criterion, the prime ideals of M_{L} are of the form $\langle p, f(X)\rangle$, where $p \in \mathbb{Z}$ is a prime number and $f(X)$ is an irreducible factor of $X^d + 1$ modulo p. Furthermore, since M_{L} is a Dedekind domain, every non-zero ideal of M_{L} has a unique factorization into these prime ideas. We leave investigations into which of these ideals are quotient-friendly or have other desirable properties to future work.

Instantiating the commitment schemes in the previous sections with the above choice of BM and observing that $\iota(\mathcal{C}, 3) = 2\gamma_R^2\|\mathcal{C}\|_R = 2d^2$ gives

$$
\begin{aligned}
N_{\mathsf{Ped}} &= 6\gamma_R^2 N^3 \iota(\mathcal{C}, 3)\|\mathcal{C}\|_R = 12d^7 \ , \\
N_{\mathsf{LF}} &= 6\gamma_R^2 N^3 \iota(\mathcal{C}, 3)\|\mathcal{C}\|_R = 12d^7 \ , \\
B_{\mathsf{SP}} &= \frac{B_{\mathsf{BRA}}}{3\gamma_R^2\|\mathcal{C}\|_R^2\|\xi^3\|_R} = \frac{B_{\mathsf{SIS}}}{24d^2} \ , \\
N_{\mathsf{SP}} &= 3\gamma_R^2 N^3 \iota(\mathcal{C}, 3)\|\mathcal{C}\|_R = 48d^7 \ , \\
B_{\mathsf{CSP}} &= \sqrt{\frac{B_{\mathsf{BRA}}}{n\gamma_R^2\|\mathcal{C}\|_R^2\|\xi^6\|_R}} = \sqrt{\frac{B_{\mathsf{SIS}}}{64nd^2}} \ , \\
N_{\mathsf{CSP}} &= 3\gamma_R^2 N^3 \iota(\mathcal{C}, 3)\|\mathcal{C}\|_R = 6d^7 \ .
\end{aligned}
$$

Additionally, $B_{\mathsf{C}} = B$, $\gamma_R = d$ and $\|\mathcal{C}\|_R = 1$, so we can apply Lemma 4.13 as follows

- to the Pedersen commitment scheme for $\mathcal{R}_{\mathrm{SC}}(8^\ell, B' = 24^\ell d^{8\ell} B)$,
- to the linear-function commitment scheme for $\mathcal{R}_{\mathrm{SC}}(8^\ell, B' = 23^\ell d^{8\ell} B)$,
- to the scalar-product commitment scheme for $\mathcal{R}_{\mathrm{SC}}(8^\ell, B' = 96^\ell d^{8\ell} B)$ if

$$B' \leq B_{\mathsf{SP}} \implies 24d^2 96^\ell d^{8\ell} B \leq B_{\mathsf{SIS}} \ ,$$

- to the compressed scalar-product commitment scheme for $\mathcal{R}_{\mathrm{SC}}(8^\ell, B' = 12^\ell d^{8\ell} B)$ if

$$B' \leq B_{\mathsf{CSP}} \implies 64nd^2(12^\ell d^{8\ell} B)^2 \leq B_{\mathsf{SIS}} \ .$$

6 Succinct argument for scalar products over rings

We give zero-knowledge succinct arguments for committed scalar-products for secure bilinear-module generators, which are scalar-product compatible, quotient-friendly and masking.

Definition 6.1. *For $c \in R$, the* **committed scalar product** *relation $\mathcal{R}_{\mathsf{CMSP}}(c, B_{\mathsf{C}})$ is the set of tuples*

$$(\mathbb{x}, \mathbb{w}) = \Big(\big(\mathsf{BM}, (\mathcal{M}, h, \mathsf{aux}), (\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1, \underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1, \mathsf{U}_0, \underline{\mathsf{U}}_1), (\mathsf{C}_a, \mathsf{C}_b, \mathsf{C}_t)\big), \big(\underline{a}, \underline{b}, t, \underline{\varrho}_a, \underline{\varrho}_b, \underline{\varrho}_t\big)\Big)$$

where BM *is a protocol-friendly bilinear-module generator, $(\mathcal{M}, h, \mathsf{aux}) \in \mathsf{BM.Setup}(1^\lambda, n)$, with $\mathsf{aux} = (B_{\mathsf{BRA}}, \mathcal{U}_{M_{\mathrm{L}}}, \mathcal{C}, \xi, N, B_{\mathsf{C}}, \kappa, I)$, where $\mathcal{U}_{M_{\mathrm{L}}}$ is the uniform distribution over $M_{\mathrm{L}}^h(B_{\mathsf{C}})$, such that*

- $\mathsf{Ped.Open}((\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1), \underline{a}, \underline{\varrho}_a, \mathsf{C}_a, c) = 1$,
- $\mathsf{Ped.Open}((\underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1), \underline{b}, \underline{\varrho}_b, \mathsf{C}_b, c) = 1$,
- $\mathsf{Ped.Open}((\mathsf{U}_0, \underline{\mathsf{U}}_1), t, \underline{\varrho}_t, \mathsf{C}_t, c^2) = 1$,

where $t = \langle \underline{a}, \underline{b} \rangle \bmod I$, $(\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1), (\underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1) \in M_{\mathrm{R}}^n \times M_{\mathrm{R}}^h$, $(\mathsf{U}_0, \underline{\mathsf{U}}_1) \in M_{\mathrm{R}} \times M_{\mathrm{R}}^h$, $\mathsf{C}_a, \mathsf{C}_b, \mathsf{C}_t \in M_{\mathrm{T}}$, $\underline{a}, \underline{b} \in M_{\mathrm{L}}^n(B_{\mathsf{C}})$, $t \in M_{\mathrm{L}}(B_{\mathsf{C}})$, $\underline{\varrho}_a, \underline{\varrho}_b, \underline{\varrho}_t \in M_{\mathrm{L}}^h(B_{\mathsf{C}})$.

Theorem 6.2. *If* BM *is a protocol-friendly bilinear-module generator then Construction 6.3 is an interactive argument for $\mathcal{R}_{\mathsf{CMSP}}(1, B_{\mathsf{C}})$ supporting instances with $n = 2^\ell$ that satisfies the following properties:*
- *(Lemma 6.5) it has completeness error $e_{\mathrm{SP}} = O(n/\kappa)$;*
- *(Lemma 6.6) it has $(3, 4^\ell)$-tree extraction;*
- *(Lemma 6.7) it has semi-honest-verifier statistical zero-knowledge;*
- *the round complexity is $O(\log n)$;*
- *the communication complexity is dominated by $O(\log n)$ elements of M_{T};*
- *the prover performs $O(n)$ applications of e and $O(n)$ operations in M_{L}, M_{R}, and M_{T};*
- *the verifier performs $O(h)$ applications of e; $O(n)$ operations in M_{R}; and $O(\log n)$ additions in M_{T}.*

We state various bounds that we use in our construction.

$$\begin{aligned}
m_1 &:= \gamma_R \|\mathcal{C}\|_R B_{\mathsf{C}} \ , \\
m_2 &:= \gamma_R n B_{\mathsf{C}}^2 \ , \\
m_3 &:= \gamma_R \|\mathcal{C}\|_R (m_2 + B_{\mathsf{C}}) \ , \\
m_4 &:= 2\kappa\gamma_R n B_{\mathsf{C}} m_1 \ , \\
m_5 &:= \gamma_R^2 \|\mathcal{C}\|_R^2 m_2 + \gamma_R \|\mathcal{C}\|_R m_4 \ .
\end{aligned}$$

Construction 6.3. We construct an interactive argument for the relation in Definition 6.1. The prover $\mathbf{P}$ and verifier $\mathbf{V}$ take as input an instance $\mathbb{x} = \big(\mathsf{BM}, (\mathcal{M}, h, \mathsf{aux}), (\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1, \underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1, \mathsf{U}_0, \underline{\mathsf{U}}_1), (\mathsf{C}_a, \mathsf{C}_b, \mathsf{C}_t)\big)$, while the prover $\mathbf{P}$ additionally takes as input a witness $\mathbb{w} = \big(\underline{a}, \underline{b}, t, \underline{\varrho}_a, \underline{\varrho}_b, \underline{\varrho}_t\big)$.

- The prover $\mathbf{P}$ samples random masks $\underline{y}_a, \underline{y}_b \leftarrow M_{\mathrm{L}}^n(\kappa m_1)$, $\zeta \leftarrow M_{\mathrm{L}}^n(\kappa m_3)$, and computes:
  - $\mathsf{C}_{y_a} := \mathsf{Ped.Commit}((\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1), \underline{y}_a; \underline{\sigma}_{y_a})$ for $\underline{\sigma}_{y_a} \leftarrow M_{\mathrm{L}}^h(\kappa m_1)$,
  - $\mathsf{C}_{y_b} := \mathsf{Ped.Commit}((\underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1), \underline{y}_b; \underline{\sigma}_{y_b})$ for $\underline{\sigma}_{y_b} \leftarrow M_{\mathrm{L}}^h(\kappa m_1)$,
  - $\mathsf{C}_\zeta := \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \zeta; \underline{\sigma}_\zeta)$ for $\underline{\sigma}_\zeta \leftarrow M_{\mathrm{L}}^h(\kappa m_3)$,
  - $v_0 := \langle \underline{y}_a, \underline{y}_b \rangle$,
  - $v_1 := \langle \underline{a}, \underline{y}_b \rangle + \langle \underline{y}_a, \underline{b} \rangle$,

– $v_2 := \langle \underline{a}, \underline{b} \rangle$,
– $\mathsf{C}_{v_0} := \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), v_0; \underline{\sigma}_{v_0})$ for $\underline{\sigma}_{v_0} \leftarrow M_{\mathrm{L}}^h(\kappa m_5)$,
– $\mathsf{C}_{v_1} := \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), v_1; \underline{\sigma}_{v_1})$ for $\underline{\sigma}_{v_1} \leftarrow M_{\mathrm{L}}^h(\kappa m_4)$,
– $\mathsf{C}_{v_2} := \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), v_2; \underline{\sigma}_{v_2})$ for $\underline{\sigma}_{v_2} \leftarrow M_{\mathrm{L}}^h(\kappa m_2)$.

The prover $\mathbf{P}$ sends $\mathsf{C}_{y_a}, \mathsf{C}_{y_b}, \mathsf{C}_\zeta, \mathsf{C}_{v_0}, \mathsf{C}_{v_1}, \mathsf{C}_{v_2} \in M_{\mathrm{T}}$, and $\zeta' := \zeta \bmod I$ to the verifier $\mathbf{V}$.

- The verifier $\mathbf{V}$ sends a random challenge $\alpha \leftarrow \mathcal{C}$ to the prover $\mathbf{P}$.

- The prover $\mathbf{P}$ computes the masked values

– $\underline{e}_a := \alpha \underline{a} + \underline{y}_a \in M_{\mathrm{L}}^n$,
– $\underline{e}_b := \alpha \underline{b} + \underline{y}_b \in M_{\mathrm{L}}^n$,
– $\underline{\rho}'_a := \alpha \underline{\rho}_a + \underline{\sigma}_{y_a} \in M_{\mathrm{L}}^h$,
– $\underline{\rho}'_b := \alpha \underline{\rho}_b + \underline{\sigma}_{y_b} \in M_{\mathrm{L}}^h$,
– $\underline{\rho}'_t := \alpha^2 \underline{\sigma}_{v_2} + \alpha \underline{\sigma}_{v_1} + \underline{\sigma}_{v_0} \in M_{\mathrm{L}}^h$,
– $\bar{v} := \alpha(\langle \underline{a}, \underline{b} \rangle - t) + \zeta \in M_{\mathrm{L}}$,
– $\underline{\sigma}' := \alpha(\underline{\sigma}_{v_2} - \underline{\rho}_t) + \underline{\sigma}_\zeta \in M_{\mathrm{L}}^h$.

The prover $\mathbf{P}$ aborts if any of the following conditions are not satisfied:

$$\begin{aligned} \|\underline{e}_a\|_{M_{\mathrm{L}}} &\le (\kappa-1)m_1 \ , & \|\underline{e}_b\|_{M_{\mathrm{L}}} &\le (\kappa-1)m_1 \ , \\ \|\underline{\rho}'_a\|_{M_{\mathrm{L}}} &\le (\kappa-1)m_1 \ , & \|\underline{\rho}'_b\|_{M_{\mathrm{L}}} &\le (\kappa-1)m_1 \ , \\ \|\underline{\rho}'_t\|_{M_{\mathrm{L}}} &\le (\kappa-1)m_5 \ , & \|\bar{v}\|_{M_{\mathrm{L}}} &\le (\kappa-1)m_3 \ , \\ \|\underline{\sigma}'\|_{M_{\mathrm{L}}} &\le (\kappa-1)m_3 \ . \end{aligned} \tag{17}$$

The prover $\mathbf{P}$ sends $\bar{v}$ and $\underline{\sigma}'$ to the verifier $\mathbf{V}$.

- The verifier $\mathbf{V}$ checks that

$$\|\bar{v}\|_{M_{\mathrm{L}}} \le (\kappa-1)m_3 \ , \text{ and } \qquad \|\underline{\sigma}'\|_{M_{\mathrm{L}}} \le (\kappa-1)m_3 \ . \tag{18}$$

The verifier $\mathbf{V}$ computes

$$\begin{aligned} \mathsf{T} &:= \big(\alpha \mathsf{C}_a + \mathsf{C}_{y_a},\ \alpha \mathsf{C}_b + \mathsf{C}_{y_b},\ \alpha^2 \mathsf{C}_t + \alpha \mathsf{C}_{v_1} + \mathsf{C}_{v_0}\big) \\ &= \mathsf{SP.Commit}((\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1, \underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1, \mathsf{U}_0, \underline{\mathsf{U}}_1), (\underline{e}_a, \underline{e}_b); (\underline{\rho}'_a, \underline{\rho}'_b, \underline{\rho}'_t)) \ . \end{aligned}$$

- The prover $\mathbf{P}$ and the verifier $\mathbf{V}$ run the opening protocol of Construction 4.5 for $\mathcal{R}_{\mathrm{SC}}(1, (\kappa-1)m_1)$ with instance $\mathbb{x} = (\mathsf{SP}, \mathsf{pp}_{\mathsf{SP}}, \mathcal{C}, (\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1, \underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1, \mathsf{U}_0, \underline{\mathsf{U}}_1), \mathsf{T})$ and witness $\mathbb{w} = (\underline{e}_a, \underline{e}_b)$ to show that

$$\mathsf{SP.Open}((\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1, \underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1, \mathsf{U}_0, \underline{\mathsf{U}}_1), (\underline{e}_a, \underline{e}_b), (\underline{\rho}'_a, \underline{\rho}'_b, \underline{\rho}'_t), \mathsf{T}, 1) = 1 \ .$$

Since $\underline{\rho}'_a, \underline{\rho}'_b \in M_{\mathrm{R}}((\kappa-1)m_1)$, and $\underline{\rho}'_t \in M_{\mathrm{R}}((\kappa-1)m_5)$, the randomness space $\mathbb{R}_{\mathsf{ck}}$ for the sumcheck argument is equal to $M_{\mathrm{R}}((\kappa-1)m_1) \times M_{\mathrm{R}}((\kappa-1)m_1) \times M_{\mathrm{R}}((\kappa-1)m_5)$.

- The verifier $\mathbf{V}$ checks that $\mathsf{Ped.Open}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \bar{v}, \underline{\sigma}', \alpha(\mathsf{C}_{v_2} - \mathsf{C}_t) + \mathsf{C}_\zeta, 1) = 1$ and $\bar{v} \bmod I = \zeta'$.

6.1 Proof of Theorem 6.2

The prover and verifier efficiency of Construction 6.3 are directly inherited from the efficiency of Construction 4.5 for the scalar-product commitment scheme, which is analyzed in Section 4 and Appendix B.

Lemma 6.4. *The prover in Construction 6.3 performs $O(n)$ applications of e; $O(n)$ operations in M_{L}; $O(n)$ operations in M_{R}; and $O(n)$ additions in M_{T}. The verifier in Construction 6.3 performs $O(h)$ applications of e; $O(n)$ operations in M_{R}; and $O(\log n)$ additions in M_{T}.*

Lemma 6.5 (completeness)**.** *Construction 6.3 is complete for $\mathcal{R}_{\mathsf{CMSP}}(1, B_{\mathsf{C}})$ with error $O(n/\kappa)$.*

Proof. Suppose that

$$\Big(\big(\mathsf{BM}, (\mathcal{M}, h, \mathsf{aux}), (\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1, \underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1, \mathsf{U}_0, \underline{\mathsf{U}}_1), (\mathsf{C}_a, \mathsf{C}_b, \mathsf{C}_t)\big), \big(\underline{a}, \underline{b}, t, \underline{\rho}_a, \underline{\rho}_b, \underline{\rho}_t\big)\Big) \in \mathcal{R}_{\mathsf{CMSP}}(1, B_{\mathsf{C}}) \ .$$

First, by the masking property of BM, the probability that any of the inequalities of Equation (17) are not satisfied is at most n/κ for each inequality. So, the prover aborts with probability at most $7n/\kappa$. In this case, the verifier's checks of Equation (18) are all satisfied, and we have $\|\underline{e}_a\|_{M_{\mathrm{L}}}, \|\underline{e}_b\|_{M_{\mathrm{L}}} \leq (\kappa - 1)m_1$.

Next, we argue that

$$(\mathbb{x}', \mathbb{w}') = \Big(\mathsf{BM}, (\mathcal{M}, h, \mathsf{aux}), (\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1, \underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1, \mathsf{U}_0, \underline{\mathsf{U}}_1), \mathsf{T}), (\underline{e}_a, \underline{e}_b, \underline{\rho}'_a, \underline{\rho}'_b, \underline{\rho}'_t)\Big) \in \mathcal{R}_{\mathrm{SC}}(1, (\kappa - 1)m_1) \ .$$

By construction, we have $\langle \underline{e}_a, \underline{e}_b \rangle = \langle \alpha\underline{a} + \underline{y}_a, \alpha\underline{b} + \underline{y}_b \rangle = \alpha^2 \langle \underline{a}, \underline{b} \rangle + \alpha(\langle \underline{a}, \underline{y}_b \rangle + \langle \underline{y}_a, \underline{b} \rangle) + \langle \underline{y}_a, \underline{y}_b \rangle = \alpha^2 v_2 + \alpha v_1 + v_0$. Thus,

$$\begin{aligned}
\mathsf{T} &= \big(\alpha\mathsf{C}_a + \mathsf{C}_{y_a},\ \alpha\mathsf{C}_b + \mathsf{C}_{y_b},\ \alpha^2\mathsf{C}_t + \alpha\mathsf{C}_{v_1} + \mathsf{C}_{v_0}\big) \\
&= \big(\langle \underline{e}_a, \underline{\mathsf{G}}_0 \rangle + \langle \underline{\rho}'_a, \underline{\mathsf{G}}_1 \rangle,\ \langle \underline{e}_b, \underline{\mathsf{H}}_0 \rangle + \langle \underline{\rho}'_b, \underline{\mathsf{H}}_1 \rangle,\ \langle \underline{e}_a, \underline{e}_b \rangle \cdot \mathsf{U}_0 + \langle \underline{\rho}'_t, \underline{\mathsf{U}}_1 \rangle\big) \\
&= \mathsf{SP.Commit}((\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1, \underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1, \mathsf{U}_0, \underline{\mathsf{U}}_1), (\underline{e}_a, \underline{e}_b); (\underline{\rho}'_a, \underline{\rho}'_b, \underline{\rho}'_t)) \ .
\end{aligned}$$

The vector $(\underline{e}_a, \underline{e}_b)$ is a valid opening of T with respect to the appropriate bases. By completeness of Construction 4.5 for the scalar-product commitment scheme, the verifier of the subprotocol for $\mathcal{R}_{\mathrm{SC}}(1, (\kappa - 1)m_1)$ will accept.

It remains to show that the verifier checks that $\mathsf{Ped.Open}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \bar{v}, \underline{\sigma}', \alpha(\mathsf{C}_{v_2} - \mathsf{C}_t) + \mathsf{C}_\zeta, 1) = 1$ and $\bar{v} \bmod I = \zeta$ succeed. Since

$$\begin{aligned}
\mathsf{C}_{v_2} &= \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \langle \underline{a}, \underline{b} \rangle; \underline{\sigma}_{v_2}) \ , \\
\mathsf{C}_t &= \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), t; \underline{\rho}_t) \ , \\
\mathsf{C}_\zeta &= \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \zeta; \underline{\sigma}_\zeta) \ , \\
\bar{v} &= \alpha(\langle \underline{a}, \underline{b} \rangle - t) + \zeta \ , \\
\underline{\sigma}' &= \alpha(\underline{\sigma}_{v_2} - \underline{\rho}_t) + \underline{\sigma}_\zeta \ ,
\end{aligned}$$

and by the homomorphic property of the Pedersen commitment scheme, it follows that $\alpha(\mathsf{C}_{v_2} - \mathsf{C}_t) + \mathsf{C}_\zeta = \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \bar{v}; \underline{\sigma}')$. Therefore, the verifier's checks succeed by the completeness property of the generalized Pedersen commitment scheme, and by reducing $\bar{v}$ modulo I. □

Lemma 6.6 (tree extraction)**.** *Let N_{SP} be as in Lemma 5.17 and*

$$B' = n \cdot (\gamma_R \|\mathcal{C}\|_R N_{\mathsf{SP}})^\ell \cdot (\kappa - 1)m_1$$

$$B_{\mathsf{CMSP}} = 4\|\xi^4\|_R\gamma_R N^2 B'^2 + 2\|\xi^{2(\ell+3)}\|_R N m_3 + 3N^3\|\xi\|_R\iota(\mathcal{C},3)(\kappa-1)m_5 \ .$$

Suppose that BM *is protocol-friendly and that*

$$\max(2\gamma_R N\|\mathcal{C}\|_R B', B_{\mathsf{CMSP}}) \leq B_{\mathsf{BRA}} \ ,$$

then there exists an efficient algorithm that, given an instance

$$\mathbb{x} = \big(\mathsf{BM}, (\mathcal{M}, h, \mathsf{aux}), (\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1, \underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1, \mathsf{U}_0, \underline{\mathsf{U}}_1), (\mathsf{C}_a, \mathsf{C}_b, \mathsf{C}_t)\big) \ ,$$

and a $(3, 4^\ell)$*-tree of accepting transcripts, extracts a witness to* $\mathcal{R}_{\mathsf{CMSP}}(c_{\mathsf{CMSP}}, B_{\mathsf{CMSP}})$ *for* $c_{\mathsf{CMSP}} = \xi^{\ell+3}$.

Note that since BM *is protocol-friendly, multiplication by* c_{CMSP} *is invertible in* M_{L}/I.

Proof. By Lemma 4.13, there is an efficient algorithm which takes the 4^ℓ-subtree of accepting transcripts for Construction 4.5 applied to $\mathcal{R}_{\mathrm{SC}}(1, (\kappa-1)m_1)$ and produces a witness $(\underline{e}_a, \underline{e}_b)$ to $\mathcal{R}_{\mathrm{SC}}(c_*, B')$ with $c_* = \xi^\ell$ and $B' := n \cdot (\gamma_R\|\mathcal{C}\|_R N_{\mathsf{SP}})^\ell \cdot (\kappa-1)m_1$ satisfying

$$(c_* \cdot \mathsf{T}_a, c_* \cdot \mathsf{T}_b, c_*^2 \cdot \mathsf{T}_t) = \big(\langle \underline{e}_a, \underline{\mathsf{G}}_0\rangle + c_*\langle \underline{\varrho}'_a, \underline{\mathsf{G}}_1\rangle,\ \langle \underline{e}_b, \underline{\mathsf{H}}_0\rangle + c_*\langle \underline{\varrho}'_b, \underline{\mathsf{H}}_1\rangle,\ \langle \underline{e}_a, \underline{e}_b\rangle \cdot \mathsf{U}_0 + c_*^2\langle \underline{\varrho}'_t, \underline{\mathsf{U}}_1\rangle\big) \ ,$$

where $\mathsf{T} = (\mathsf{T}_a, \mathsf{T}_b, \mathsf{T}_t)$, $\underline{\varrho}'_a \in M_{\mathrm{R}}((\kappa-1)m_1)$, $\underline{\varrho}'_b \in M_{\mathrm{R}}((\kappa-1)m_1)$, and $\underline{\varrho}'_t \in M_{\mathrm{R}}((\kappa-1)m_5)$. This is done for each value of $\alpha^{(j)}$, $j \in [3]$, in the $(3, 4^\ell)$-tree of accepting transcripts. Hence, we have that for each $j \in [3]$

$$(c_* \cdot \mathsf{T}_a^{(j)}, c_* \cdot \mathsf{T}_b^{(j)}, c_*^2 \cdot \mathsf{T}_t^{(j)}) = \Big(c_*(\alpha^{(j)}\mathsf{C}_a + \mathsf{C}_{y_a}),\ c_*(\alpha^{(j)}\mathsf{C}_b + \mathsf{C}_{y_b}),\ c_*^2(\alpha^{(j)^2}\mathsf{C}_{v_2} + \alpha^{(j)}\mathsf{C}_{v_1} + \mathsf{C}_{v_0})\Big) \tag{19}$$

$$= \Big(\mathsf{Ped.Commit}((\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1), \underline{e}_a^{(j)}; c_*\underline{\varrho}_a'^{(j)}),$$
$$\mathsf{Ped.Commit}((\underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1), \underline{e}_b^{(j)}; c_*\underline{\varrho}_b'^{(j)}),$$
$$\mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \langle \underline{e}_a^{(j)}, \underline{e}_b^{(j)}\rangle; c_*^2\underline{\varrho}_t'^{(j)})\Big) \ .$$

Examining each component of Equation (19) gives

$$c_*(\alpha^{(j)}\mathsf{C}_a + \mathsf{C}_{y_a}) = \mathsf{Ped.Commit}((\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1), \underline{e}_a^{(j)}; c_*\underline{\varrho}_a'^{(j)}) \ , \tag{20}$$

$$c_*(\alpha^{(j)}\mathsf{C}_b + \mathsf{C}_{y_b}) = \mathsf{Ped.Commit}((\underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1), \underline{e}_b^{(j)}; c_*\underline{\varrho}_b'^{(j)}) \ , \tag{21}$$

$$\text{and} \quad c_*^2(\alpha^{(j)^2}\mathsf{C}_{v_2} + \alpha^{(j)}\mathsf{C}_{v_1} + \mathsf{C}_{v_0}) = \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \langle \underline{e}_a^{(j)}, \underline{e}_b^{(j)}\rangle; c_*^2\underline{\varrho}_t'^{(j)}) \ . \tag{22}$$

Now, we consider Equation (20) and Equation (21) for two distinct challenge values $\alpha^{(1)}$ and $\alpha^{(2)}$ and set $c := \alpha^{(1)} - \alpha^{(2)}$. By subtracting Equation (20) and Equation (21) for the two distinct challenges $\alpha^{(1)}$ and $\alpha^{(2)}$ and using the pseudoinverse parameters $(\mathcal{C}, \xi, N)$ (Definition 3.13), we obtain $\underline{a}$, $\underline{b}$, $\underline{\varrho}_a$, $\underline{\varrho}_b$ with norms at most $2NB'$ such that

$$\xi^{\ell+1}\mathsf{C}_a = \langle \underline{a}, \underline{\mathsf{G}}_0\rangle + \xi^{\ell+1}\langle \underline{\varrho}_a, \underline{\mathsf{G}}_1\rangle$$
$$\xi^{\ell+1}\mathsf{C}_b = \langle \underline{b}, \underline{\mathsf{H}}_0\rangle + \xi^{\ell+1}\langle \underline{\varrho}_b, \underline{\mathsf{H}}_1\rangle \ .$$

It is left to show how to recover an opening of C_t. We first recover openings to C_{v_2} and C_ζ, and then use the verification check $\mathsf{Ped.Open}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \bar{v}, \underline{\sigma}', \alpha(\mathsf{C}_{v_2} - \mathsf{C}_t) + \mathsf{C}_\zeta, 1) = 1$ to recover a relaxed opening of C_t.

By multiplying Equation (20) and Equation (21) for challenges $\alpha^{(1)}$ and $\alpha^{(2)}$ with $\alpha^{(2)}$ and $\alpha^{(1)}$, respectively, and then subtracting them, we obtain $\underline{y}_a$, $\underline{y}_b$, $\underline{\sigma}_{y_a}$, $\underline{\sigma}_{y_b}$ with norms at most $2\gamma_R N\|\mathcal{C}\|_R B' \leq B_{\mathsf{BRA}}$ such that

$$\begin{aligned}\xi^{\ell+1}\mathsf{C}_{y_a} &= \langle \underline{y}_a, \underline{\mathsf{G}}_0\rangle + \xi^{\ell+1}\langle \underline{\sigma}_{y_a}, \underline{\mathsf{G}}_1\rangle \\ \xi^{\ell+1}\mathsf{C}_{y_b} &= \langle \underline{y}_b, \underline{\mathsf{H}}_0\rangle + \xi^{\ell+1}\langle \underline{\sigma}_{y_b}, \underline{\mathsf{H}}_1\rangle \ .\end{aligned}$$

It must be that for each $j \in [3]$, $\xi\underline{e}_a^{(j)} = \alpha^{(j)}\underline{a} + \underline{y}_a$, $\xi\underline{e}_b^{(j)} = \alpha^{(j)}\underline{b} + \underline{y}_b$, $\xi^{\ell+1}\underline{\rho}_a'^{(j)} = \xi^{\ell+1}(\alpha^{(j)}\underline{\rho}_a + \underline{\sigma}_{y_a})$ and $\xi^{\ell+1}\underline{\rho}_b'^{(j)} = \xi^{\ell+1}(\alpha^{(j)}\underline{\rho}_b + \underline{\sigma}_{y_b})$, otherwise we can use Equation (20) or Equation (21) to break the binding property of the generalized Pedersen commitment scheme. Substituting in Equation (22) these values gives

$$\xi^{2(\ell+1)}({\alpha^{(j)}}^2\mathsf{C}_{v_2} + \alpha^{(j)}\mathsf{C}_{v_1} + \mathsf{C}_{v_0}) = {\alpha^{(j)}}^2\langle\underline{a},\underline{b}\rangle\cdot\mathsf{U}_0 + \alpha^{(j)}(\langle\underline{a},\underline{y}_b\rangle + \langle\underline{y}_a,\underline{b}\rangle)\cdot\mathsf{U}_0 + \langle\underline{y}_a,\underline{y}_b\rangle\cdot\mathsf{U}_0 + \xi^{2(\ell+1)}\langle\underline{\rho}_t', \underline{\mathsf{U}}_1\rangle \tag{23}$$

We define the Vandermonde matrix $V := \begin{pmatrix} {\alpha^{(1)}}^2 & \alpha^{(1)} & 1 \\ {\alpha^{(2)}}^2 & \alpha^{(2)} & 1 \\ {\alpha^{(3)}}^2 & \alpha^{(3)} & 1 \end{pmatrix}$, and its adjugate $\mathrm{adj}(V)$.

Multiplying the first row of $\mathrm{adj}(V)$ with $({\alpha^{(j)}}^2\mathsf{C}_{v_2} + \alpha^{(j)}\mathsf{C}_{v_1} + \mathsf{C}_{v_0})_{j\in[3]}$ gives $\det(V)\mathsf{C}_{v_2}$. Hence, from Equation (23) multiplied by the first row of $\mathrm{adj}(V)$, and the pseudoinverse parameters $(\mathcal{C}, \xi, N)$, we get $\underline{\sigma}_{v_2}$ with norm at most $3N^3\|\xi\|_R\iota(\mathcal{C},3)(\kappa-1)m_5 \leq B_{\mathsf{BRA}}$ such that

$$\xi^{2(\ell+3)}\mathsf{C}_{v_2} = \xi^4\langle\underline{a},\underline{b}\rangle\cdot\mathsf{U}_0 + \xi^{2(\ell+3)}\langle\underline{\sigma}_{v_2}, \underline{\mathsf{U}}_1\rangle \tag{24}$$

This means that the message $\langle\xi^2\underline{a}, \xi^2\underline{b}\rangle$ and randomness $\underline{\sigma}_{v_2}$ are a relaxed opening to C_{v_2} with slackness $\xi^{2(\ell+3)}$.

Finally, we show that C_t has a relaxed opening which is related to the opening of C_{v_2} in M_{L}, considered modulo I. The verifier's check that $\mathsf{Ped.Open}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \bar{v}^{(j)}, \underline{\sigma}'^{(j)}, \alpha^{(j)}(\mathsf{C}_{v_2} - \mathsf{C}_t) + \mathsf{C}_\zeta, 1) = 1$ for each $j \in [3]$ implies that

$$\alpha^{(j)}(\mathsf{C}_{v_2} - \mathsf{C}_t) + \mathsf{C}_\zeta = \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \bar{v}^{(j)}; \underline{\sigma}'^{(j)}) \ . \tag{25}$$

We also have that $\bar{v}^{(j)} \bmod I = \zeta'$. Considering Equation (25) for $j = 1$ and $j = 2$, subtracting one from the other, and using the definition of pseudoinverse parameters $(\mathcal{C}, \xi, N)$ we find openings O and ζ, and randomness ϕ and $\underline{\sigma}_\zeta$ such that

$$\xi(\mathsf{C}_{v_2} - \mathsf{C}_t) = \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \xi O; \xi\phi) \ , \tag{26}$$

$$\xi\mathsf{C}_\zeta = \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \xi\zeta; \xi\underline{\sigma}_\zeta) \ , \tag{27}$$

where the norms of O, ζ, ϕ and $\underline{\sigma}_\zeta$ are at most $2Nm_3 \leq B_{\mathsf{BRA}}$. Furthermore, multiplying Equation (25) by ξ and substituting the openings of Equation (26) and Equation (27), we have that either $\xi\bar{v}^{(j)} = \alpha^{(j)}O + \zeta$ and $\xi\underline{\sigma}'^{(j)} = \alpha^{(j)}\phi + \underline{\sigma}_\zeta$, or we can use Equation (25) to break the binding property of Pedersen commitment.

The verifier's checks guarantee that $\bar{v}^{(1)} = \bar{v}^{(2)} = \zeta' \bmod I$. Hence, we have $\alpha^{(1)}O + \zeta = \alpha^{(2)}O + \zeta \bmod I \implies (\alpha^{(1)} - \alpha^{(2)})O = 0 \bmod I$. Since multiplication by ξ is invertible in M_{L}/I, it follows that $O = 0 \bmod I$.

Rearranging Equation (26), multiplying by $\xi^{2\ell+5}$, and using Equation (24) shows that

$$\xi^{2(\ell+3)}\mathsf{C}_t = \xi^{2(\ell+3)}\mathsf{C}_{v_2} - \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \xi^{2(\ell+3)}O; \xi^{2(\ell+3)}\phi)$$

$$= \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \langle \xi^2 \underline{a}, \xi^2 \underline{b} \rangle; \xi^{2(\ell+3)} \underline{\sigma}_{v_2}) - \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \xi^{2(\ell+3)} O; \xi^{2(\ell+3)} \phi)$$

This means that message $\langle \xi^2 \underline{a}, \xi^2 \underline{b} \rangle - \xi^{2(\ell+3)} O$ and randomness $\underline{\sigma}_{v_2} - \phi$ give a relaxed opening to C_t with slackness $\xi^{2(\ell+3)}$. Since $O = 0 \bmod I$, $\langle \xi^2 \underline{a}, \xi^2 \underline{b} \rangle$, which is a relaxed opening of C_{v_2}, and $\langle \xi^2 \underline{a}, \xi^2 \underline{b} \rangle - \xi^{2(\ell+3)} O$, which is the opening of C_t, are equal modulo I.

The norms of $\langle \xi^2 \underline{a}, \xi^2 \underline{b} \rangle - \xi^{2(\ell+3)} O$ and $\underline{\sigma}_{v_2} - \phi$ are at most

$$B_{\mathsf{CMSP}} = 4\|\xi^4\|_R \gamma_R N^2 B'^2 + 2\|\xi^{2(\ell+3)}\|_R N m_3 + 3N^3 \|\xi\|_R \iota(\mathcal{C}, 3)(\kappa - 1) m_5 \ .$$

Finally, $(\xi^2 \underline{a}, \xi^2 \underline{b}, \langle \xi^2 \underline{a}, \xi^2 \underline{b} \rangle - \xi^{2(\ell+3)} O, \xi^2 \underline{\rho}_a, \xi^2 \underline{\rho}_b, \underline{\sigma}_{v_2} - \phi)$ is a witness to $\mathcal{R}_{\mathsf{CMSP}}(c_{\mathsf{CMSP}}, B_{\mathsf{CMSP}})$, with $c_{\mathsf{CMSP}} = \xi^{\ell+3}$. □

Lemma 6.7 (zero knowledge)**.** *Construction 6.3 is semi-honest-verifier statistical zero-knowledge.*

Proof. We give a simulator for Construction 6.3, which takes an input the verifier's message α and the verifier's randomness r_{SC} for the opening protocol of Construction 4.5.

1. Sample openings $\underline{e}_a, \underline{e}_b \leftarrow M_{\mathrm{L}}^n((\kappa - 1)m_1)$ and $\bar{v} \leftarrow M_{\mathrm{L}}^n((\kappa - 1)m_1)$. Set $\zeta' = \bar{v} \bmod I$.
2. Sample commitment randomness $\underline{\rho}'_a, \underline{\rho}'_b \leftarrow M_{\mathrm{L}}^h((\kappa - 1)m_1)$, $\underline{\sigma}' \leftarrow M_{\mathrm{L}}^h((\kappa - 1)m_3)$ and $\underline{\rho}'_t \leftarrow M_{\mathrm{L}}^h((\kappa - 1)m_5)$.
3. Compute $\mathsf{T} := (\langle \underline{e}_a, \underline{\mathsf{G}}_0 \rangle,\ \langle \underline{e}_b, \underline{\mathsf{H}}_0 \rangle,\ \langle \underline{e}_a, \underline{e}_b \rangle \cdot \mathsf{U}_0)$ and $\mathsf{T}' := \mathsf{T} + (\langle \underline{\rho}'_a, \underline{\mathsf{G}}_1 \rangle,\ \langle \underline{\rho}'_b, \underline{\mathsf{H}}_1 \rangle,\ \langle \underline{\rho}'_t, \underline{\mathsf{U}}_1 \rangle)$.
4. Compute the following commitments:

$$\begin{aligned}
\mathsf{C}_{y_a} &:= \mathsf{Ped.Commit}((\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1), \underline{0}; \underline{\sigma}_{y_a}) \quad \text{for} \quad \underline{\sigma}_{y_a} \leftarrow M_{\mathrm{L}}^h(\kappa m_1) \ , \\
\mathsf{C}_{y_b} &:= \mathsf{Ped.Commit}((\underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1), \underline{0}; \underline{\sigma}_{y_b}) \quad \text{for} \quad \underline{\sigma}_{y_b} \leftarrow M_{\mathrm{L}}^h(\kappa m_1) \ , \\
\mathsf{C}_{v_2} &:= \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \underline{0}; \underline{\sigma}_{v_2}) \quad \text{for} \quad \underline{\sigma}_{v_2} \leftarrow M_{\mathrm{L}}^h(\kappa m_2) \ , \\
\mathsf{C}_{v_1} &:= \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \underline{0}; \underline{\sigma}_{v_1}) \quad \text{for} \quad \underline{\sigma}_{v_1} \leftarrow M_{\mathrm{L}}^h(\kappa m_4) \ , \\
\mathsf{C}_{v_0} &:= \mathsf{T}' - \left(\alpha \mathsf{C}_a + \mathsf{C}_{y_a},\ \alpha \mathsf{C}_b + \mathsf{C}_{y_b},\ \alpha^2 \mathsf{C}_t - \alpha \mathsf{C}_{v_1}\right) \ , \\
\mathsf{C}_\zeta &:= \mathsf{Ped.Commit}(\mathsf{U}_0, \underline{\mathsf{U}}_1; \bar{v}, \underline{\sigma}') - \alpha(\mathsf{C}_{v_2} - \mathsf{C}_t) \ .
\end{aligned}$$

5. Run the opening protocol of Construction 4.5 for $\mathcal{R}_{\mathrm{SC}}(1, (\kappa - 1)m_1)$ with instance $\mathbb{x} = (\mathsf{SP}, \mathsf{pp}_{\mathsf{SP}}, \mathcal{C}, (\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1, \underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1, \mathsf{U}_0, \underline{\mathsf{U}}_1), \mathsf{T})$ and witness $\mathbb{w} = (\underline{e}_a, \underline{e}_b)$ using the verifier's randomness r_{SC}.
6. Abort with probability $7n/\kappa$, equal to that of the honest prover, outputting only the commitments and ζ' in that case.

We argue that the simulated transcript is statistically indistinguishable from a transcript generated by an honest prover. Firstly, note that the simulator aborts with the same probability as the honest prover.

The openings $\underline{e}_a, \underline{e}_b \in M_{\mathrm{L}}^n((\kappa - 1)m_1)$, and $\bar{v} \in M_{\mathrm{L}}^n((\kappa - 1)m_1)$, and randomness $\underline{\rho}'_a, \underline{\rho}'_b \leftarrow M_{\mathrm{L}}^h((\kappa - 1)m_1)$, $\underline{\sigma}' \leftarrow M_{\mathrm{L}}^h((\kappa - 1)m_3)$ and $\underline{\rho}'_t \leftarrow M_{\mathrm{L}}^h((\kappa - 1)m_5)$, are uniformly distributed in a real protocol

execution when the prover does not abort. This implies that the simulator produces distributions of these values which are statistically indistinguishable from those in a real protocol execution.

By the hiding property of the generalized Pedersen commitment scheme, $\mathsf{C}_{y_a}, \mathsf{C}_{y_b}, \mathsf{C}_{v_2}, \mathsf{C}_{v_1}$ are statistically indistinguishable from honestly generated commitments.

In both a real or simulated execution, all other simulated values are now fully determined: C_ζ and C_{v_0} by linear relations used in the protocol, ζ' by reducing $\bar{v}$ modulo I, and the rest of the transcript by executing Construction 4.5. The result follows. □

7 Succinct argument for R1CS over rings

We construct a zero-knowledge succinct argument for R1CS over rings. We define the R1CS relation, we formally re-state Theorem 2, and then describe the protocol. In Lemmas 7.5 to 7.7 we show that the protocol is complete, knowledge sound, and zero-knowledge respectively. In Lemma 7.8, we provide bounds on the number of operations performed by the prover and verifier.

Definition 7.1 (R1CS)**.** *Given a ring $R_\bullet$, the relation $\mathcal{R}_{\mathrm{R1CS}}(R_\bullet)$ is the set of all triples*

$$(\mathbb{x}, \mathbb{w}) = \big((A, B, C, m, n_{\mathrm{row}}, n_{\mathrm{col}}, n_{\mathrm{in}}, \underline{x}), \underline{w}\big)$$

where A, B, C are matrices in $R_\bullet^{n_{\mathrm{row}} \times n_{\mathrm{col}}}$ each having at most m non-zero entries, $\underline{x} \in R_\bullet^{n_{\mathrm{in}}}$, $\underline{w} \in R_\bullet^{n_{\mathrm{col}} - n_{\mathrm{in}}}$, and $\underline{z} := (\underline{x}, \underline{w}) \in R_\bullet^{n_{\mathrm{col}}}$ is such that $A\underline{z} \circ B\underline{z} = C\underline{z}$. (Here "$\circ$" is the entry-wise product between vectors.)

Our results supports R1CS over rings $R_\bullet$ induced by suitable bilinear modules: $R_\bullet = M_{\mathrm{L}}/I$ where M_{L} is itself a ring and I is a suitable ideal of M_{L}. For simplicity we state the lemma for $n_{\mathrm{row}} = n_{\mathrm{col}} = 2^\ell$.

Theorem 7.2. *If* BM *is a protocol-friendly bilinear-module generator then Construction 7.4 is an interactive argument for $\mathcal{R}_{\mathrm{R1CS}}(R_\bullet)$ supporting instances with $n_{\mathrm{row}} = n_{\mathrm{col}} = 2^\ell$ that satisfies the following properties:*

- *(Lemma 7.5) it has completeness error $O(e_{\mathrm{SP}}) = O(n_{\mathrm{row}}/\kappa)$;*
- *(Lemma 7.6) it has $(2^\ell, 2^\ell, 2^{\log n_{\mathrm{in}}}, 3, 2, 2, 4^\ell)$-tree extraction;*
- *(Lemma 7.7) it has semi-honest-verifier statistical zero-knowledge;*
- *the round complexity is $O(\log n_{\mathrm{row}})$;*
- *the communication complexity is dominated by $O(\log n_{\mathrm{row}})$ elements of M_{T};*
- *(Lemma 7.8) the prover and verifier each perform $O(n_{\mathrm{row}})$ applications of e, $O(n_{\mathrm{row}} + m)$ operations in M_{L}, and $O(n_{\mathrm{row}})$ additions in M_{T}.*

Remark 7.3 (computing on $\mathcal{R}_{\mathrm{R1CS}}$)**.** The R1CS instance and its witness are defined over $R_\bullet = M_{\mathrm{L}}/I$. However, the commitment scheme used in the protocol commits to messages over M_{L}. Therefore, during the protocol, we will consider $\underline{w}$ and other values defined over $R_\bullet$ as elements of M_{L}, with each element in $R_\bullet$ represented by a fixed representative of M_{L} that reduces to $\underline{w}$ modulo I. We define $\|R_\bullet\|_{M_{\mathrm{L}}, I}$ to be the maximum over the norms of representatives of elements of $R_\bullet$ in M_{L}. In principle, one could use any set of representatives. We pick representatives with minimal norm.

Construction 7.4. The prover $\mathbf{P}$ and verifier $\mathbf{V}$ take as input an instance $\mathbb{x} = (A, B, C, m, n_{\mathrm{row}}, n_{\mathrm{col}}, n_{\mathrm{in}}, \underline{x})$, while the prover $\mathbf{P}$ additionally takes as input a witness $\mathbb{w} = \underline{w}$.

- The prover $\mathbf{P}$ assembles the satisfying assignment $\underline{z} := (\underline{x}, \underline{w}) \in R_\bullet^{n_{\mathrm{col}}}$, and computes the vectors $\underline{z}_A := A\underline{z} \bmod I$, $\underline{z}_B := B\underline{z} \bmod I$, $\underline{z}_C := C\underline{z} \bmod I$ in $R_\bullet^{n_{\mathrm{row}}}$. Then $\mathbf{P}$ computes commitments to $\underline{z}, \underline{z}_A, \underline{z}_B$ as follows.

$$\begin{aligned}
\mathsf{C}_{\underline{z}} &:= \mathsf{Ped.Commit}((\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1), \underline{z}; \varrho) \quad \text{for} \quad \varrho \leftarrow M_{\mathrm{L}}^h(\|R_\bullet\|_{M_{\mathrm{L}}, I}) \ , \\
\mathsf{C}_{\underline{z}_A} &:= \mathsf{Ped.Commit}((\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1), \underline{z}_A; \varrho_A) \quad \text{for} \quad \varrho_A \leftarrow M_{\mathrm{L}}^h(\|R_\bullet\|_{M_{\mathrm{L}}, I}) \ , \\
\mathsf{C}_{\underline{z}_B} &:= \mathsf{Ped.Commit}((\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1), \underline{z}_B; \varrho_B) \quad \text{for} \quad \varrho_B \leftarrow M_{\mathrm{L}}^h(\|R_\bullet\|_{M_{\mathrm{L}}, I}) \ .
\end{aligned}$$

  The prover $\mathbf{P}$ sends $\mathsf{C}_{\underline{z}}, \mathsf{C}_{\underline{z}_A}, \mathsf{C}_{\underline{z}_B} \in M_{\mathrm{T}}$ to the verifier $\mathbf{V}$.

- The verifier $\mathbf{V}$ sends random challenge vectors $(r_1, \ldots, r_{\log n_{\mathrm{row}}}) \in \mathcal{C}^{\log n_{\mathrm{row}}}$ and $(y_1, \ldots, y_{\log n_{\mathrm{row}}}) \in \mathcal{C}^{\log n_{\mathrm{row}}}$ to the prover $\mathbf{P}$.

- The prover $\mathbf{P}$ and verifier $\mathbf{V}$ compute:
  - the vectors $\underline{r} := \bigotimes_{i=1}^{\log n_{\text{row}}}(1, r_i)$ and $\underline{y} := \bigotimes_{i=1}^{\log n_{\text{row}}}(1, y_i)$ in $R^{n_{\text{row}}}$,
  - the vectors $\underline{r}_A := \underline{r}^{\intercal} A \bmod I$, $\underline{r}_B := \underline{r}^{\intercal} B \bmod I$, $\underline{r}_C := \underline{r}^{\intercal} C \bmod I$ in $R_{\bullet}^{n_{\text{col}}}$.

- The prover $\mathbf{P}$ computes the following elements and commitments

$$\begin{aligned}
\alpha &:= \langle \underline{z}_A, \underline{r} \rangle \bmod I & \mathsf{C}_\alpha &:= \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \alpha; \varrho_\alpha) \quad \text{for} \quad \varrho_\alpha \leftarrow M_{\text{L}}^h(\|R_\bullet\|_{M_{\text{L}},I}) \\
\beta &:= \langle \underline{z}_B, \underline{r} \rangle \bmod I & \mathsf{C}_\beta &:= \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \beta; \varrho_\beta) \quad \text{for} \quad \varrho_\beta \leftarrow M_{\text{L}}^h(\|R_\bullet\|_{M_{\text{L}},I}) \\
\gamma &:= \langle \underline{z}_C, \underline{r} \rangle \bmod I & \mathsf{C}_\gamma &:= \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \gamma; \varrho_\gamma) \quad \text{for} \quad \varrho_\gamma \leftarrow M_{\text{L}}^h(\|R_\bullet\|_{M_{\text{L}},I}) \\
\alpha' &:= \langle \underline{z}_A, \underline{r} \circ \underline{y} \rangle \bmod I & \mathsf{C}'_\alpha &:= \mathsf{Ped.Commit}((\mathsf{U}_0, \underline{\mathsf{U}}_1), \alpha'; \varrho''_\alpha) \quad \text{for} \quad \varrho''_\alpha \leftarrow M_{\text{L}}^h(\|R_\bullet\|_{M_{\text{L}},I}) \\
\underline{z}'_A &:= \underline{r} \circ \underline{z}_A \bmod I & \mathsf{C}'_{\underline{z}_A} &:= \mathsf{Ped.Commit}((\underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1), \underline{z}'_A; \varrho'_A) \quad \text{for} \quad \varrho'_A \leftarrow M_{\text{L}}^h(\|R_\bullet\|_{M_{\text{L}},I})
\end{aligned}$$

  and sends $\mathsf{C}_\alpha, \mathsf{C}_\beta, \mathsf{C}_\gamma, \mathsf{C}'_\alpha, \mathsf{C}'_{\underline{z}_A} \in M_{\text{T}}$ to the verifier $\mathbf{V}$.

- The verifier $\mathbf{V}$ samples a random challenge vector $(s_1, \ldots, s_{\log n_{\text{in}}}) \in \mathcal{C}^{\log n_{\text{in}}}$ and sends it to the prover $\mathbf{P}$. The prover $\mathbf{P}$ and the verifier $\mathbf{V}$ compute the vectors $\underline{s} := \bigotimes_{i=1}^{\log n_{\text{in}}}(1, s_i) \in R^{n_{\text{in}}}$ and $\underline{s}' \in R^{n_{\text{row}}}$ obtained by padding $\underline{s}$ with zeroes. The verifier $\mathbf{V}$ computes $\sigma := \langle \underline{x}, \underline{s} \rangle \bmod I \in R_\bullet$.

- The prover $\mathbf{P}$ and verifier $\mathbf{V}$ engage in several scalar-product sub-protocols, in parallel. Each scalar-product sub-protocol has instance $\mathbb{x}$ of the form $(\mathcal{M}, h, I, \underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1, \underline{\mathsf{H}}_0, \underline{\mathsf{H}}_1, \mathsf{U}_0, \underline{\mathsf{U}}_1, X, Y, Z)$. The values of X, Y, Z, the witnesses, and the purpose of each sub-protocol are specified in the table below; for each protocol, the witnesses are reduced modulo I and have norm at most $\|R_\bullet\|_{M_{\text{L}},I}$.

  Note that $\underline{r}, \underline{y}, \underline{r}_A, \underline{r}_B, \underline{r}_C, \underline{s}', \sigma$ are known to $\mathbf{V}$, who will compute commitments to these values for itself, with respect to bases $\underline{\mathsf{G}}_0$, $\underline{\mathsf{H}}_0$, or U_0 as required. This is signified in the table using a "#" symbol.

  Also note that the challenge vectors, such as $\underline{r}$, have elements in R rather than M_{L}, but are easy to map into M_{L} simply by multiplying each element by $1_{M_{\text{L}}}$. We do not do this explicitly in order to keep notation simple.

Witness	X, Y, Z	Checks	Purpose
$(\underline{z}_B, \underline{z}'_A, \varrho_B, \varrho'_A, \varrho_\gamma)$	$\mathsf{C}_{\underline{z}_B}, \mathsf{C}'_{\underline{z}_A}, \mathsf{C}_\gamma$	$\langle \underline{z}_B, \underline{z}'_A \rangle = \gamma \bmod I$	Hadamard check
$(\underline{y}, \underline{z}'_A, 0, \varrho'_A, \varrho''_\alpha)$	$\#, \mathsf{C}'_{\underline{z}_A}, \mathsf{C}'_\alpha$	$\langle \underline{y}, \underline{z}'_A \rangle = \alpha' \bmod I$	Consistency check on $\underline{z}_A, \underline{z}'_A$
$(\underline{z}_A, \underline{r} \circ \underline{y}, \varrho_A, 0, \varrho''_\alpha)$	$\mathsf{C}_{\underline{z}_A}, \#, \mathsf{C}'_\alpha$	$\langle \underline{z}_A, \underline{r} \circ \underline{y} \rangle = \alpha' \bmod I$	Consistency check on $\underline{z}_A, \underline{z}'_A$
$(\underline{z}, \underline{r}_A, \varrho, 0, \varrho_\alpha)$	$\mathsf{C}_{\underline{z}}, \#, \mathsf{C}_\alpha$	$\langle \underline{z}, \underline{r}_A \rangle = \alpha \bmod I$	Lincheck for A
$(\underline{z}_A, \underline{r}, \varrho_A, 0, \varrho_\alpha)$	$\mathsf{C}_{\underline{z}_A}, \#, \mathsf{C}_\alpha$	$\langle \underline{z}_A, \underline{r} \rangle = \alpha \bmod I$	Lincheck for A
$(\underline{z}, \underline{r}_B, \varrho, 0, \varrho_\beta)$	$\mathsf{C}_{\underline{z}}, \#, \mathsf{C}_\beta$	$\langle \underline{z}, \underline{r}_B \rangle = \beta \bmod I$	Lincheck for B
$(\underline{z}_B, \underline{r}, \varrho_B, 0, \varrho_\beta)$	$\mathsf{C}_{\underline{z}_B}, \#, \mathsf{C}_\beta$	$\langle \underline{z}_B, \underline{r} \rangle = \beta \bmod I$	Lincheck for B
$(\underline{z}, \underline{r}_C, \varrho, 0, \varrho_\gamma)$	$\mathsf{C}_{\underline{z}}, \#, \mathsf{C}_\gamma$	$\langle \underline{z}, \underline{r}_C \rangle = \gamma \bmod I$	Lincheck for C
$(\underline{z}, \underline{s}', \varrho, 0, 0)$	$\mathsf{C}_{\underline{z}}, \#, \#$	$\langle \underline{z}, \underline{s}' \rangle = \sigma \bmod I$	Partial assignment

Lemma 7.5 (completeness)**.** *Let e_{SP} be the completeness error for the scalar-product sub-protocol (Construction 6.3) obtained in Lemma 6.5. Then Construction 7.4 is complete with completeness error $O(e_{\text{SP}})$.*

Proof. Let $(\mathbb{x}, \mathbb{w}) = \big((A, B, C, m, n_{\text{row}}, n_{\text{col}}, n_{\text{in}}, \underline{x}), \underline{w}\big) \in \mathcal{R}_{\text{R1CS}}$. Then $\underline{z} := (\underline{x}, \underline{w})$ satisfies $A\underline{z} \circ B\underline{z} = C\underline{z} \bmod I$. The prover computes $\underline{z}_A = A\underline{z} \bmod I$ and similarly for B and C. Hence $\underline{z}_A \circ \underline{z}_B = \underline{z}_C \bmod I$.

We argue that each scalar-product sub-protocol is invoked on a valid instance-witness pair. As there are nine such sub-protocols, the claimed completeness error follows directly.

- *Hadamard and consistency checks.* For any choice of $\underline{r}$, we have

$$\langle \underline{z}_B, \underline{z}'_A \rangle = \langle \underline{z}_B, \underline{r} \circ \underline{z}_A \rangle = \langle \underline{r}, \underline{z}_A \circ \underline{z}_B \rangle = \langle \underline{z}_C, \underline{r} \rangle = \gamma \bmod I \ .$$

  Thus, $\underline{z}_B$, $\underline{z}'_A$ and γ and their commitments give a member of the scalar-product relation $\mathcal{R}_{\mathsf{CMSP}}(1, \|R_\bullet\|_{M_{\mathrm{L}},I})$. Further, for every choice of $\underline{y}$, we have $\langle \underline{y}, \underline{z}'_A \rangle = \langle \underline{y}, \underline{r} \circ \underline{z}_A \rangle = \alpha' \bmod I$. Therefore, the second and third scalar-product protocols succeed, except with probability $O(e_{\mathrm{SP}})$.

- *Linchecks.* For every choice of $\underline{r}$ it holds that $\langle \underline{r}, \underline{z}_A \rangle = \langle \underline{r}, A\underline{z} \rangle = \langle \underline{r}^\intercal A, \underline{z} \rangle = \langle \underline{r}_A, \underline{z} \rangle \bmod I$, and therefore, the fourth and fifth scalar-product protocols succeed (the Lincheck for A), except with probability $O(e_{\mathrm{SP}})$. Similar reasoning applies to the Linchecks for B and C.

- *Partial assignment consistency.* For every choice of $\underline{s}$, it holds that $\langle \underline{s}', \underline{z} \rangle = \langle \underline{s}, \underline{x} \rangle = \sigma \bmod I$, which is the equation checked by the verifier. This means that $\underline{z}$, $\underline{s}'$, σ and their commitments are in the relation $\mathcal{R}_{\mathsf{CMSP}}$, so the final scalar product sub-protocol succeeds.

□

Lemma 7.6 (tree extraction)**.** *There exists an efficient algorithm such that given an instance* $\mathbb{x} = (A, B, C, m, n_{\mathrm{row}}, n_{\mathrm{col}}, n_{\mathrm{in}}, \underline{x})$*, and a* $(2^\ell, 2^\ell, 2^{\log n_{\mathrm{in}}}, 3, 2, 2, 4^\ell)$*-tree of accepting transcripts, either extracts a valid witness* $\mathbb{w} = \underline{w}$ *for the relation* $\mathcal{R}_{\mathrm{R1CS}}(R_\bullet)$ *or a non-trivial bilinear relation, whenever* $\gamma_R n B^2_{\mathsf{CMSP}} < B_{\mathsf{BRA}}$*. Here* B_{CMSP} *is the norm bound derived in Lemma 6.6, with* $B_{\mathsf{C}} = \|R_\bullet\|_{M_{\mathrm{L}},I}$*.*

Proof. Fix values of $\underline{r}$, $\underline{y}$ and $\underline{s}$ (this corresponds to following some path from the root down to the $(2\ell + \log n_{\mathrm{in}})$-th level of the tree of transcripts). By Lemma 6.6 there exists a polynomial-time algorithm which takes as input the $(3, 2, 2, 4^\ell)$-subtree of accepting transcripts (the full tree restricted to each scalar-product subprotocol) and outputs a witness to $\mathcal{R}_{\mathsf{CMSP}}(c_{\mathsf{CMSP}}, B_{\mathsf{CMSP}})$ (see Definition 6.1) for some c_{CMSP} that is invertible modulo I since BM is quotient-friendly.

Consider the first scalar-product subprotocol, run on commitments $\mathsf{C}_{\underline{z}_B}, \mathsf{C}'_{\underline{z}_A}, \mathsf{C}_\gamma$. By applying the knowledge extractor for the scalar-product subprotocol, we can extract vectors $\underline{z}_B, \underline{z}'_A \in M_{\mathrm{L}}(B_{\mathsf{CMSP}})^{n_{\mathrm{col}}}$ and a scalar γ such that $c_{\mathsf{CMSP}} \cdot \mathsf{C}_{\underline{z}_B} = \langle \underline{z}_B, \underline{\mathsf{G}}_0 \rangle + \langle \underline{\rho}_B, \underline{\mathsf{G}}_1 \rangle$, $c_{\mathsf{CMSP}} \cdot \mathsf{C}'_{\underline{z}_A} = \langle \underline{z}'_A, \underline{\mathsf{H}}_0 \rangle + \langle \underline{\rho}'_A, \underline{\mathsf{H}}_1 \rangle$, and $c^2_{\mathsf{CMSP}} \cdot \mathsf{C}_\gamma = \gamma \cdot \mathsf{U}_0 + \langle \underline{\rho}_\gamma, \underline{\mathsf{U}}_1 \rangle$, for suitable randomness values $\underline{\rho}_B$, $\underline{\rho}'_A$ and $\underline{\rho}_\gamma$, with $\langle \underline{z}_B, \underline{z}'_A \rangle = \gamma \bmod I$.

Now consider the second scalar-product protocol, run on an honestly-made commitment to $\underline{y}$, and commitments C'_α, $\mathsf{C}'_{\underline{z}_A}$. We can extract vectors $\underline{z}''_A \in M_{\mathrm{L}}(B_{\mathsf{CMSP}})^{n_{\mathrm{col}}}$ and a scalar α' such that $\langle c_{\mathsf{CMSP}} \cdot \underline{y}, \underline{z}''_A \rangle = \alpha' \bmod I$ which open C'_α and $\mathsf{C}'_{\underline{z}_A}$ with suitable randomness values. Note that $\|c_{\mathsf{CMSP}} \cdot \underline{y}\|_R \leq \gamma_R \|c_{\mathsf{CMSP}}\|_R \|\mathcal{C}\|_R \leq B_{\mathsf{CMSP}} < B_{\mathsf{BRA}}$. Since the norms of all the openings are less than B_{BRA}, by the binding property of the generalized Pedersen commitment scheme, $\underline{z}''_A = \underline{z}'_A$, so $\langle c_{\mathsf{CMSP}} \cdot \underline{y}, \underline{z}'_A \rangle = \alpha' \bmod I$.

Similarly, by applying the knowledge extractor for each scalar-product protocol and using the binding property to show that relaxed openings for commitments made by the verifier are multiples of the honestly committed values, we can extract vectors $\underline{z}, \underline{z}_A, \underline{z}_B$ and scalars $\alpha, \alpha', \beta, \gamma$ with norms at most B_{CMSP}, satisfying the following equations:

$$\begin{aligned}
\langle \underline{z}_B, \underline{z}'_A \rangle &= \gamma \bmod I \\
\langle c_{\mathsf{CMSP}} \cdot \underline{y}, \underline{z}'_A \rangle &= \alpha' \bmod I \\
\langle \underline{z}_A, c_{\mathsf{CMSP}} \cdot \underline{r} \circ \underline{y} \rangle &= \alpha' \bmod I \\
\langle \underline{z}, c_{\mathsf{CMSP}} \cdot \underline{r}_A \rangle &= \alpha \bmod I \\
\langle \underline{z}_A, c_{\mathsf{CMSP}} \cdot \underline{r} \rangle &= \alpha \bmod I
\end{aligned}$$

$$\langle \underline{z}, c_{\mathsf{CMSP}} \cdot \underline{r}_B \rangle = \beta \bmod I$$
$$\langle \underline{z}_B, c_{\mathsf{CMSP}} \cdot \underline{r} \rangle = \beta \bmod I$$
$$\langle \underline{z}, c_{\mathsf{CMSP}} \cdot \underline{r}_C \rangle = \gamma \bmod I$$
$$\langle \underline{z}, c_{\mathsf{CMSP}} \cdot \underline{s}' \rangle = c_{\mathsf{CMSP}}^2 \cdot \sigma \bmod I$$

We also know that $\langle c_{\mathsf{CMSP}} \cdot \underline{s}, c_{\mathsf{CMSP}} \cdot \underline{x} \rangle = c_{\mathsf{CMSP}}^2 \cdot \sigma \bmod I$.

Equating the left-hand sides of the equations above which have related right-hand sides, and dividing by $c_{\mathsf{CMSP}}^2 \bmod I$ (c_{CMSP} is invertible modulo I by Lemma 6.6), we can write

$$\langle c_{\mathsf{CMSP}}^{-1} \cdot \underline{z}_B, c_{\mathsf{CMSP}}^{-1} \cdot \underline{z}'_A \rangle = \langle c_{\mathsf{CMSP}}^{-1} \cdot \underline{z}, \underline{r}_C \rangle \bmod I \ , \tag{28}$$
$$\langle \underline{y}, c_{\mathsf{CMSP}}^{-1} \cdot \underline{z}'_A \rangle = \langle c_{\mathsf{CMSP}}^{-1} \cdot \underline{z}_A, \underline{r} \circ \underline{y} \rangle \bmod I \ , \tag{29}$$
$$\langle c_{\mathsf{CMSP}}^{-1} \cdot \underline{z}, \underline{r}_A \rangle = \langle c_{\mathsf{CMSP}}^{-1} \cdot \underline{z}_A, \underline{r} \rangle \bmod I \ , \tag{30}$$
$$\langle c_{\mathsf{CMSP}}^{-1} \cdot \underline{z}, \underline{r}_B \rangle = \langle c_{\mathsf{CMSP}}^{-1} \cdot \underline{z}_B, \underline{r} \rangle \bmod I \ , \tag{31}$$
$$\langle c_{\mathsf{CMSP}}^{-1} \cdot \underline{z}, \underline{s}' \rangle = \langle \underline{s}, \underline{x} \rangle \bmod I \ . \tag{32}$$

Rearranging Equation (29) gives

$$\langle c_{\mathsf{CMSP}}^{-1} \cdot \underline{z}'_A - c_{\mathsf{CMSP}}^{-1} \cdot \underline{z}_A \circ \underline{r}, \underline{y} \rangle = 0 \bmod I \ . \tag{33}$$

Now, we show that $c_{\mathsf{CMSP}}^{-1} \cdot \underline{z}$ is a witness to the R1CS instance modulo I.

Consider Equation (33) for each accepting transcript at the $(2\ell + \log n_{\text{in}})$-th level. The entries of $\underline{y}$ are distinct monomials in $y_1, \ldots, y_\ell$, and Equation (33) is a multilinear polynomial in $(\underline{y}_1, \ldots, \underline{y}_{\log n_{\text{row}}})$ where each coefficient is an entry of $c_{\mathsf{CMSP}}^{-1} \cdot \underline{z}'_A - c_{\mathsf{CMSP}}^{-1} \cdot \underline{z}_A \circ \underline{r}$. Equation (33) holds for a $(2^\ell, 2^\ell)$-tree of values of $(r_1, \ldots, r_\ell), (y_1, \ldots, y_\ell)$, and gives a system of linear equations with coefficients in terms of the y_i for each choice of $(r_1, \ldots, r_\ell)$. We can solve the linear equations to deduce that each entry of $c_{\mathsf{CMSP}}^{-1} \cdot \underline{z}'_A - c_{\mathsf{CMSP}}^{-1} \cdot \underline{z}_A \circ \underline{r}$ is equal to zero. This implies that for every choice of $(r_1, \ldots, r_\ell)$ in the tree of accepting transcripts, $c_{\mathsf{CMSP}}^{-1} \cdot \underline{z}'_A = c_{\mathsf{CMSP}}^{-1} \cdot \underline{r} \circ \underline{z}_A \bmod I$. It is always possible to solve the linear system because Equation (33) is a multilinear polynomial, and solving for the coefficients amounts to solving linear equations in $\underline{y}_{\log n_{\text{row}}}$, then $\underline{y}_{\log n_{\text{row}}-1}$, and so on, recursively. Each linear equation is solvable up to factors of ξ. Since BM is protocol-friendly, multiplication by ξ is invertible modulo I, so each linear equation is completely solvable modulo I.

Substituting $c_{\mathsf{CMSP}}^{-1} \cdot \underline{z}'_A = c_{\mathsf{CMSP}}^{-1} \cdot \underline{r} \circ \underline{z}_A$ into Equation (28) and applying the same technique shows that $c_{\mathsf{CMSP}}^{-1}\underline{z}_A \circ c_{\mathsf{CMSP}}^{-1}\underline{z}_B = C \cdot c_{\mathsf{CMSP}}^{-1}\underline{z} \bmod I$. Applying the same technique to Equation (30), Equation (31) and Equation (32) implies that $c_{\mathsf{CMSP}}^{-1}\underline{z}_A = A \cdot c_{\mathsf{CMSP}}^{-1}\underline{z} \bmod I$, $c_{\mathsf{CMSP}}^{-1}\underline{z}_B = B \cdot c_{\mathsf{CMSP}}^{-1}\underline{z} \bmod I$, and $c_{\mathsf{CMSP}}^{-1}\underline{z} = (\underline{x}, \underline{w})$ for some vector $\underline{w}$, and thus $c_{\mathsf{CMSP}}^{-1}\underline{z}$ is a witness to the R1CS relation modulo I. □

Lemma 7.7 (zero-knowledge)**.** *Suppose that*

- *the scalar-product sub-protocol (Construction 6.3) is statistically semi-honest-verifier zero-knowledge, with statistical distance at most δ_{SP} between simulated transcripts and real transcripts; and*
- *the generalized Pedersen commitment is statistically hiding with statistical distance at most ϵ between commitments to different messages.*

Then Construction 7.4 is statistically semi-honest-verifier zero-knowledge, with statistical distance at most $O(\delta_{\text{SP}} + \epsilon)$ between simulated transcripts and real transcripts.

Proof. We give a simulator for Construction 7.4, which takes as input the verifier's messages $(r_1, \ldots, r_{\log n_{\text{row}}}) \in \mathcal{C}^{\log n_{\text{row}}}$, $(y_1, \ldots, y_{\log n_{\text{row}}}) \in \mathcal{C}^{\log n_{\text{row}}}$ and $(s_1, \ldots, s_{\log n_{\text{in}}}) \in \mathcal{C}^{\log n_{\text{in}}}$, and the verifier's randomness $r_{\mathsf{CMSP},1}, \ldots, r_{\mathsf{CMSP},9}$ for each of the 9 scalar-product sub-protocols, described by Construction 6.3.

The simulator runs as follows:

- The simulator computes random commitments $\mathsf{C}_{\underline{z}}, \mathsf{C}_{\underline{z}_A}, \mathsf{C}_{\underline{z}_B}, \mathsf{C}_\alpha, \mathsf{C}_\beta, \mathsf{C}_\gamma, \mathsf{C}'_\alpha, \mathsf{C}'_{\underline{z}_A}$ to zero messages, using randomness sampled uniformly from $M_{\text{L}}^h(\|R_\bullet\|_{M_{\text{L}},I})$.
- For $i = 1, \ldots, 9$, the simulator invokes the simulator of the scalar-product protocol (given by Lemma 6.7) for the i-th scalar-product subprotocol, using verifier randomness $r_{\mathsf{CMSP},i}$.

Zero-knowledge follows from the hiding property of the generalized Pedersen commitment scheme, and the zero-knowledge property of the scalar-product protocol. □

Lemma 7.8. *The prover and verifier in Construction 7.4 each perform $O(n_{\text{row}})$ applications of e; $O(n_{\text{row}}+m)$ operations in M_{L}; and $O(n_{\text{row}})$ additions in M_{T}.*

Proof. The costs of Construction 7.4 are mostly inherited from the costs of Construction 6.3, which are analyzed in Lemma 6.4. Both the prover and the verifier must also compute $\underline{r}_A, \underline{r}_B, \underline{r}_C$ in M_{L}, which costs $O(m)$ operations in M_{L}. □

A Folding techniques as a sumcheck argument

We review the folding technique introduced in [BCCGP16], which is an interactive protocol for proving knowledge of an opening of a given Pedersen commitment. Then we explain how that interactive protocol can be reformulated as a sumcheck argument. Similar reformulations can be done for other folding techniques in the literature (including for scalar-product commitments, and different cryptographic settings).

The folding technique in [BCCGP16] is an interactive reduction that halves the message length and can be applied recursively. Below $\mathbb{G}$ is a group of prime order and $\mathbb{F}$ is the finite field of size $|\mathbb{G}|$.

Protocol 5: folding technique for Pedersen commitments

For $n = 2^\ell$, the prover and verifier receive as input a commitment key $\underline{\mathsf{G}} \in \mathbb{G}^n$ and commitment $\mathsf{C} \in \mathbb{G}$. The prover also receives as input an opening $\underline{a} \in \mathbb{F}^n$ such that $\mathsf{C} = \langle \underline{a}, \underline{\mathsf{G}} \rangle$.

If $n = 1$ then the prover sends $a \in \mathbb{F}$ to the verifier, and the verifier checks if $a \cdot \mathsf{G} = \mathsf{C}$ as claimed. If $n > 1$, the interactive reduction works as follows.

1. Parse $\underline{\mathsf{G}} \in \mathbb{G}^n$ as $(\underline{\mathsf{G}}_0, \underline{\mathsf{G}}_1) \in \mathbb{G}^{n/2} \times \mathbb{G}^{n/2}$, and $\underline{a} \in \mathbb{F}^n$ as $(\underline{a}_0, \underline{a}_1) \in \mathbb{F}^{n/2} \times \mathbb{F}^{n/2}$.
2. The prover computes cross terms $\mathsf{C}_- := \langle \underline{a}_0, \underline{\mathsf{G}}_1 \rangle$ and $\mathsf{C}_+ := \langle \underline{a}_1, \underline{\mathsf{G}}_0 \rangle$ and sends them to the verifier.
3. The verifier samples $r \leftarrow \mathbb{F}$ and sends r to the prover.
4. The verifier outputs the new commitment key $\underline{\mathsf{G}}' := r \cdot \underline{\mathsf{G}}_0 + \underline{\mathsf{G}}_1 \in \mathbb{G}^{n/2}$ and the new commitment $\mathsf{C}' := \mathsf{C}_- + r \cdot \mathsf{C} + r^2 \cdot \mathsf{C}_+$. The prover outputs the new opening $\underline{a}' := \underline{a}_0 + r \cdot \underline{a}_1 \in \mathbb{F}^{n/2}$.

The reduction preserves completeness because if we expand the new commitment C' then the original commitment C appears as the middle coefficient of the polynomial in r with C_+ and C_- as the other terms:

$$\langle \underline{a}', \underline{\mathsf{G}}' \rangle = \langle \underline{a}_0 + r \cdot \underline{a}_1, r \cdot \underline{\mathsf{G}}_0 + \underline{\mathsf{G}}_1 \rangle = \langle \underline{a}_0, \underline{\mathsf{G}}_1 \rangle + r \cdot \langle \underline{a}, \underline{\mathsf{G}} \rangle + r^2 \cdot \langle \underline{a}_1, \underline{\mathsf{G}}_0 \rangle = \mathsf{C}_- + r \cdot \mathsf{C} + r^2 \cdot \mathsf{C}_+ = \mathsf{C}' \ .$$

Thus $\underline{a}'$ is a new opening for the new commitment C' under the new commitment key $\underline{\mathsf{G}}'$. Intuitively, the reduction is secure because the prover sends the cross terms C_+ and C_- before receiving the challenge r; this intuition can be formalized via an extraction argument.

After the reduction, the prover may send the new opening $\underline{a}' \in \mathbb{F}^{n/2}$ to the verifier, who checks that $\mathsf{C}' = \langle \underline{a}', \underline{\mathsf{G}}' \rangle$. Alternatively, the interactive reduction can be applied recursively until the final opening is a single field element a, the final Pedersen commitment key is a single group element G, and the verifier checks that the final commitment C^* satisfies $\mathsf{C}^* = a \cdot \mathsf{G}$. In this case the total number of recursions is $\ell = \log_2 n$.

Next, we describe a sumcheck argument that also proves knowledge of an opening for a given Pedersen commitment. For a vector $\underline{v}$ of length $n = 2^\ell$, whose entries we index via binary strings $(i_1, \ldots, i_\ell) \in \{0,1\}^\ell$, we consider the multilinear polynomial $p_{\underline{v}}$ from Definition 2; we use $\mathrm{rv}(\underline{v})$ to denote $\underline{v}$ in reverse order.

We use a generalization of the sumcheck protocol over modules (Section 2.1) that works for *weighted sums* of the form

$$\tau = \sum_{\underline{\omega} \in H^\ell} \mu_1(\omega_1) \cdots \mu_\ell(\omega_\ell)\ p(\underline{\omega})$$

with coefficients $\mu_1, \ldots, \mu_\ell \in R^H$ as in [Mei13]. In this protocol, the i-th prover message is the polynomial

$$q_i(X) := \sum_{\omega_{i+1}, \ldots, \omega_\ell \in H} \mu_{i+1}(\omega_{i+1}) \cdots \mu_\ell(\omega_\ell)\ p(r_1, \ldots, r_{i-1}, X, \omega_{i+1}, \ldots, \omega_\ell) \ ,$$

and the verifier checks that $\tau = \sum_{\omega_1 \in H} \mu_1(\omega_1)\ q_1(\omega_1)$ and that $\wedge_{i=2}^{\ell} q_{i-1}(r_{i-1}) = \sum_{\omega_i \in H} \mu_i(\omega_i)\ q_i(\omega_i)$.

Protocol 6: sumcheck argument for Pedersen commitments (variant of Protocol 1)

For $n = 2^\ell$, the prover and verifier receive as input a commitment key $\underline{\mathsf{G}} \in \mathbb{G}^n$ and commitment $\mathsf{C} \in \mathbb{G}$. The prover also receives as input an opening $\underline{a} \in \mathbb{F}^n$ such that $\mathsf{C} = \langle \underline{a}, \underline{\mathsf{G}} \rangle$.

The prover and verifier engage in a sumcheck protocol for the instance

$$\mathbb{x}_{\mathrm{SC}} := (R = \mathbb{F},\ M = \mathbb{G},\ H = \{-1, 1\},\ \ell = \log n,\ \tau = \mathsf{C},\ \mathcal{C} = \mathbb{F},\ \{\mu_i(\omega_i)\}_{i=1}^{\ell} = \{\tfrac{\omega_i}{2}\}_{i=1}^{\ell})$$

where the prover uses the polynomial $p(\underline{X}) := p_{\underline{a}}(\underline{X}) \cdot p_{\mathrm{rv}(\underline{\mathsf{G}})}(\underline{X})$. After the end of the sumcheck protocol, the prover learns $\underline{r} \in \mathbb{F}^\ell$ and the verifier learns $(\underline{r}, v) \in \mathbb{F}^\ell \times \mathbb{G}$. Then the prover computes and sends $p_{\underline{a}}(\underline{r}) \in \mathbb{F}$ to the verifier, and the verifier computes $p_{\mathrm{rv}(\underline{\mathsf{G}})}(\underline{r}) \in \mathbb{G}$ and checks that $p_{\underline{a}}(\underline{r}) \cdot p_{\mathrm{rv}(\underline{\mathsf{G}})}(\underline{r}) = v$.

Finally, we explain how Protocol 6 is essentially mathematically equivalent to Protocol 5. Note that Protocol 6 is slightly different from Protocol 1 (another knowledge protocol for Pedersen commitments) in order to facilitate the equivalence. We establish the equivalence via a sequence of observations.

(1) Pedersen commitment as polynomial summation. One can express the Pedersen commitment $\mathsf{C} = \langle \underline{a}, \underline{\mathsf{G}} \rangle \in \mathbb{G}$ (the scalar product of a vector over $\mathbb{F}$ and a vector over $\mathbb{G}$) as a sum of evaluations of $p_{\underline{a}} \cdot p_{\mathrm{rv}(\underline{\mathsf{G}})}$:

$$\sum_{\omega_1,\ldots,\omega_\ell \in \{-1,1\}} \frac{\omega_1}{2} \cdots \frac{\omega_\ell}{2} \cdot p_{\underline{a}}(\omega_1, \ldots, \omega_\ell) \cdot p_{\mathrm{rv}(\underline{\mathsf{G}})}(\omega_1, \ldots, \omega_\ell) = \langle \underline{a}, \underline{\mathsf{G}} \rangle \ . \tag{34}$$

Below we follow [BCG20] which explains the same for the scalar product of two vectors over $\mathbb{F}$.

Each contribution to the coefficient of $X_1 \cdots X_\ell$ in $p_{\underline{a}}(\underline{X}) \cdot p_{\mathrm{rv}(\underline{\mathsf{G}})}(\underline{X})$ arises from a multiplication of the monomials in the terms $a_{i_1,\ldots,i_\ell} X_1^{i_1} \cdots X_\ell^{i_\ell}$ and $\mathsf{G}_{i_1,\ldots,i_\ell} X_1^{1-i_1} \cdots X_\ell^{1-i_\ell}$, which multiply to give $a_{i_1,\ldots,i_\ell} \cdot \mathsf{G}_{i_1,\ldots,i_\ell} \cdot X_1 \cdots X_\ell$. Thus, the coefficient of $X_1 \cdots X_\ell$ in $p_{\underline{a}}(\underline{X}) \cdot p_{\mathrm{rv}(\underline{\mathsf{G}})}(\underline{X})$ is equal to $\langle \underline{a}, \underline{\mathsf{G}} \rangle = \mathsf{C}$.

Next, for any univariate polynomial $P(X)$, computing $\frac{1}{2}P(1) - \frac{1}{2}P(-1) = \sum_{\omega \in \{-1,1\}} \frac{\omega}{2} \cdot P(\omega)$ gives the sum of the odd coefficients of P. Applying the same idea to $p(\underline{X}) = p_{\underline{a}}(\underline{X}) \cdot p_{\mathrm{rv}(\underline{\mathsf{G}})}(\underline{X})$ and summing $\frac{\omega_1}{2} \cdots \frac{\omega_\ell}{2} \cdot p(\underline{\omega})$ over $\underline{\omega} \in \{-1, 1\}^\ell$ returns the sum of the coefficients $p_{\underline{i}}$ such that $\underline{i} \equiv 1 \bmod 2$. The only such non-zero coefficient is the coefficient of $X_1 \cdots X_\ell$, which is $\langle \underline{a}, \underline{\mathsf{G}} \rangle$, as claimed in Equation (34).

(2) New commitment key and new opening. Each iteration of Protocol 5 produces a new commitment key and a new opening that are the *same* as the coefficients of partial evaluations of $p_{\underline{\mathsf{G}}}(\underline{X})$ and $p_{\underline{a}}(\underline{X})$ in each round of Protocol 6. We focus on the first iteration of Protocol 5, as commitment keys and openings for other iterations are computed similarly.

- The new commitment key in Protocol 5 is $\underline{\mathsf{G}}' = r \cdot \underline{\mathsf{G}}_0 + \underline{\mathsf{G}}_1 \in \mathbb{G}^{n/2}$ and its entries are $\mathsf{G}'_{i_2,\ldots,i_\ell} = r \cdot \mathsf{G}_{0,\ldots,i_\ell} + \mathsf{G}_{1,\ldots,i_\ell} = \sum_{i_1 \in \{0,1\}} \mathsf{G}_{i_1,\ldots,i_\ell} r^{1-i_1}$. In Protocol 6 after the first round, we have

  $$p_{\mathrm{rv}(\underline{\mathsf{G}})}(r, X_2, \ldots, X_\ell) = \sum_{i_1,\ldots,i_\ell \in \{0,1\}} \mathsf{G}_{1-i_1,\ldots,1-i_\ell} r^{i_1} X_2^{i_2} \cdots X_\ell^{i_\ell}$$

  and so $\mathrm{rv}(\underline{\mathsf{G}}')$ are the coefficients of $p_{\mathrm{rv}(\underline{\mathsf{G}})}(r, X_2, \ldots, X_\ell)$.

- The new opening in Protocol 5 is $\underline{a}' = \underline{a}_0 + r \cdot \underline{a}_1 \in \mathbb{F}^{n/2}$ and its entries are $a'_{i_2,\ldots,i_\ell} = a_{0,\ldots,i_\ell} + r \cdot a_{1,\ldots,i_\ell} = \sum_{i_1 \in \{0,1\}} a_{i_1,\ldots,i_\ell} r^{i_1}$. In Protocol 6 after the first round, we have

  $$p_{\underline{a}}(r, X_2, \ldots, X_\ell) = \sum_{i_1,\ldots,i_\ell \in \{0,1\}} a_{i_1,\ldots,i_\ell} r^{i_1} X_2^{i_2} \cdots X_\ell^{i_\ell}$$

and so $\underline{a}'$ are the coefficients of $p_{\underline{a}}(r, X_2, \ldots, X_\ell)$.

(3) The cross terms as coefficients of q_i. The cross terms $\mathsf{C}_+, \mathsf{C}_-$ in the i-th iteration of Protocol 5 are the coefficients of X^2, X^0 of the polynomial $q_i(X)$ sent by the prover in the i-th round in Protocol 6. We again focus on the first iteration of Protocol 5, and argue that $q_1(X) = \mathsf{C}_- + X \cdot \mathsf{C} + X^2 \cdot \mathsf{C}_+$. In Protocol 6 the coefficients of $1, X, X^2$ in

$$q_1(X) = \sum_{\omega_2,\ldots,\omega_\ell \in \{-1,1\}} \frac{\omega_2}{2} \cdots \frac{\omega_\ell}{2} \cdot p(X, \omega_2, \ldots, \omega_\ell)$$

are equal to the coefficients of $X_2 \cdots X_\ell, X_1 X_2 \cdots X_\ell, X_1^2 X_2 \cdots X_\ell$ in the polynomial $p_{\underline{a}}(\underline{X}) \cdot p_{\mathrm{rv}(\underline{\mathsf{G}})}(\underline{X})$, because the sum leaves only the coefficients $p_{\underline{i}}$ such that $(i_2, \ldots, i_\ell) \equiv 1 \bmod 2$. We have already established that the coefficient of $X_1 X_2 \cdots X_\ell$ is $\mathsf{C} = \langle \underline{a}, \underline{\mathsf{G}} \rangle$. Next, each contribution to the $X_2 \cdots X_\ell$ term of $p_{\underline{a}}(\underline{X}) \cdot p_{\mathrm{rv}(\underline{\mathsf{G}})}(\underline{X})$ arises from a multiplication of the monomials in the terms $a_{0,i_2,\ldots,i_\ell} X_2^{i_2} \cdots X_\ell^{i_\ell}$ and $\mathsf{G}_{1,i_2,\ldots,i_\ell} X_2^{1-i_2} \cdots X_\ell^{1-i_\ell}$, which multiply to give $a_{0,i_2,\ldots,i_\ell} \cdot \mathsf{G}_{1,i_2,\ldots,i_\ell} \cdot X_2 \cdots X_\ell$. Thus, the coefficient of $X_2 \cdots X_\ell$ in $p_{\underline{a}}(\underline{X}) \cdot p_{\mathrm{rv}(\underline{\mathsf{G}})}(\underline{X})$ is equal to $\langle \underline{a}_0, \underline{\mathsf{G}}_1 \rangle$, which is equal to C_-. Similar reasoning applies to C_+.

(4) The verification equations. In Protocol 6 the prover in the first round sends $q_1(X) = c_0 + X \cdot c_1 + X^2 \cdot c_2$ to the verifier, and the verifier checks that $\mathsf{C} = \frac{1}{2} q_1(1) - \frac{1}{2} q_1(-1)$. One can then view the next round's commitment C' to be $q_1(r) = c_0 + r \cdot c_1 + r^2 \cdot c_2$. The verification equations in Protocol 6 simplify to give the calculation of the new commitment key in Protocol 5, as we now explain. We have already argued that for the honest prover in Protocol 6 it holds that $(c_0, c_1, c_2) = (\mathsf{C}_-, \mathsf{C}, \mathsf{C}_+)$, and so the prover could send only c_0 and c_2 to the verifier, and the verifier could avoid the verification check and simply compute $q_1(r)$ as $c_0 + r \cdot \mathsf{C} + r^2 \cdot c_2$ (as c_1 and C are supposed to be equal). Similarly, we can avoid the verification equations for all rounds of Protocol 6. Therefore, Protocol 5 incorporates a straightforward optimization to Protocol 6.

B Sumcheck arguments with additional homomorphism

We show that if a sumcheck-friendly commitment scheme has linear or bilinear properties then the polynomial $p_{\mathrm{sc}}(\underline{X}) := f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{X}), p_{\mathsf{ck}}(\underline{X}), 1)$ has a straightforward representation in terms of the coefficients of $p_{\mathsf{m}}(\underline{X})$ and $p_{\mathsf{ck}}(\underline{X})$, and the prover of Construction 4.5 can be implemented more efficiently.

Definition B.1. *Let* $f_{\mathsf{CM}} \colon \mathbb{M} \times \mathbb{K} \times \mathbb{S}_{\mathsf{ck}} \to \mathbb{C}$*,* $p_{\mathsf{m}}(\underline{X}) \in \mathbb{M}[\underline{X}]$*, and* $p_{\mathsf{ck}}(\underline{X}) \in \mathbb{K}[\underline{X}]$*.*

- $(f_{\mathsf{CM}}, p_{\mathsf{m}}, p_{\mathsf{ck}})$ *is* R**-linear** *if* $p_{\mathsf{m}}(\underline{X}) = \sum_{\underline{a} \in I^\ell} p_{\mathsf{m},\underline{a}}\, \underline{X}^{\underline{a}} \in \mathbb{M}[\underline{X}]$ *for some* $I \subseteq \mathbb{Z}$*,* $p_{\mathsf{ck}}(\underline{X})$ *is equal to a constant* $p_{\mathsf{ck}} \in \mathbb{K}$*, and for all* $r \in R$*,* $\mathsf{m}, \mathsf{m}' \in \mathbb{M}$*,* $\mathsf{ck} \in \mathbb{K}$*,*

$$\begin{aligned} f_{\mathsf{CM}}(\mathsf{m} + \mathsf{m}', \mathsf{ck}, 1) &= f_{\mathsf{CM}}(\mathsf{m}, \mathsf{ck}, 1) + f_{\mathsf{CM}}(\mathsf{m}', \mathsf{ck}, 1) \ , \\ f_{\mathsf{CM}}(r \cdot \mathsf{m}, \mathsf{ck}, 1) &= r \cdot f_{\mathsf{CM}}(\mathsf{m}, \mathsf{ck}, 1) \ . \end{aligned}$$

- $(f_{\mathsf{CM}}, p_{\mathsf{m}}, p_{\mathsf{ck}})$ *is* R**-bilinear** *if* $p_{\mathsf{m}}(\underline{X}) = \sum_{\underline{a} \in I^\ell} p_{\mathsf{m},\underline{a}}\, \underline{X}^{\underline{a}} \in \mathbb{M}[\underline{X}]$ *and* $p_{\mathsf{ck}}(\underline{X}) = \sum_{\underline{b} \in I^\ell} p_{\mathsf{ck},\underline{b}}\, \underline{X}^{\underline{b}} \in \mathbb{K}[\underline{X}]$ *for some* $I \subseteq \mathbb{Z}$*, and for all* $r \in R$*,* $\mathsf{m}, \mathsf{m}' \in \mathbb{M}$*,* $\mathsf{ck}, \mathsf{ck}' \in \mathbb{K}$*,*

$$\begin{aligned} f_{\mathsf{CM}}(\mathsf{m} + \mathsf{m}', \mathsf{ck}, 1) &= f_{\mathsf{CM}}(\mathsf{m}, \mathsf{ck}, 1) + f_{\mathsf{CM}}(\mathsf{m}', \mathsf{ck}, 1) \ , \\ f_{\mathsf{CM}}(\mathsf{m}, \mathsf{ck} + \mathsf{ck}', 1) &= f_{\mathsf{CM}}(\mathsf{m}, \mathsf{ck}, 1) + f_{\mathsf{CM}}(\mathsf{m}, \mathsf{ck}', 1) \ , \\ r \cdot f_{\mathsf{CM}}(\mathsf{m}, \mathsf{ck}, 1) &= f_{\mathsf{CM}}(r \cdot \mathsf{m}, \mathsf{ck}, 1) = f_{\mathsf{CM}}(\mathsf{m}, r \cdot \mathsf{ck}, 1) \ . \end{aligned}$$

- $(f_{\mathsf{CM}}, p_{\mathsf{m}}, p_{\mathsf{ck}})$ *is* R**-cross-bilinear** *if* $\mathbb{M} = \mathbb{M}_{\mathrm{L}} \times \mathbb{M}_{\mathrm{R}}$*,* $\mathbb{K} = \mathbb{K}_{\mathrm{L}} \times \mathbb{K}_{\mathrm{R}} \times \mathbb{K}_{\mathrm{U}}$*,*

$$p_{\mathsf{m}}(\underline{X}) = \left(\sum_{\underline{a} \in I^\ell} p_{\mathsf{m},\mathrm{L},\underline{a}}\, \underline{X}^{\underline{a}}, \sum_{\underline{a} \in I^\ell} p_{\mathsf{m},\mathrm{R},\underline{a}}\, \underline{X}^{\underline{a}} \right) \in \mathbb{M}_{\mathrm{L}}[\underline{X}] \times \mathbb{M}_{\mathrm{R}}[\underline{X}] \quad \textit{and}$$

$$p_{\mathsf{ck}}(\underline{X}) = \left(\sum_{\underline{b} \in I^\ell} p_{\mathsf{ck},\mathrm{L},\underline{b}}\, \underline{X}^{\underline{b}}, \sum_{\underline{b} \in I^\ell} p_{\mathsf{ck},\mathrm{R},\underline{b}}\, \underline{X}^{\underline{b}}, p_{\mathsf{ck},\mathrm{U}} \right) \in \mathbb{K}_{\mathrm{L}}[\underline{X}] \times \mathbb{K}_{\mathrm{R}}[\underline{X}] \times \mathbb{K}_{\mathrm{U}} \ ,$$

and for every $r \in R$*,* $(\mathsf{m}_{\mathrm{L}}, \mathsf{m}_{\mathrm{R}}), (\mathsf{m}'_{\mathrm{L}}, \mathsf{m}'_{\mathrm{R}}) \in \mathbb{M}_{\mathrm{L}} \times \mathbb{M}_{\mathrm{R}}$*,* $(\mathsf{ck}_{\mathrm{L}}, \mathsf{ck}_{\mathrm{R}}, \mathsf{ck}_{\mathrm{U}}), (\mathsf{ck}'_{\mathrm{L}}, \mathsf{ck}'_{\mathrm{R}}, \mathsf{ck}'_{\mathrm{U}}) \in \mathbb{K}_{\mathrm{L}} \times \mathbb{K}_{\mathrm{R}} \times \mathbb{K}_{\mathrm{U}}$*,*

$$\begin{aligned} f_{\mathsf{CM}}(\mathsf{m}_{\mathrm{L}} + \mathsf{m}'_{\mathrm{L}}, \mathsf{m}_{\mathrm{R}}, \mathsf{ck}_{\mathrm{L}} + \mathsf{ck}'_{\mathrm{L}}, \mathsf{ck}_{\mathrm{R}}, \mathsf{ck}_{\mathrm{U}}, 1) &= f_{\mathsf{CM}}(\mathsf{m}_{\mathrm{L}}, \mathsf{m}_{\mathrm{R}}, \mathsf{ck}_{\mathrm{L}}, \mathsf{ck}_{\mathrm{R}}, \mathsf{ck}_{\mathrm{U}}, 1) + f_{\mathsf{CM}}(\mathsf{m}'_{\mathrm{L}}, \mathsf{m}_{\mathrm{R}}, \mathsf{ck}'_{\mathrm{L}}, \mathsf{ck}_{\mathrm{R}}, \mathsf{ck}_{\mathrm{U}}, 1) \ , \\ f_{\mathsf{CM}}(\mathsf{m}_{\mathrm{L}}, \mathsf{m}_{\mathrm{R}} + \mathsf{m}'_{\mathrm{R}}, \mathsf{ck}_{\mathrm{L}}, \mathsf{ck}_{\mathrm{R}} + \mathsf{ck}'_{\mathrm{R}}, \mathsf{ck}_{\mathrm{U}}, 1) &= f_{\mathsf{CM}}(\mathsf{m}_{\mathrm{L}}, \mathsf{m}_{\mathrm{R}}, \mathsf{ck}_{\mathrm{L}}, \mathsf{ck}_{\mathrm{R}}, \mathsf{ck}_{\mathrm{U}}, 1) + f_{\mathsf{CM}}(\mathsf{m}_{\mathrm{L}}, \mathsf{m}'_{\mathrm{R}}, \mathsf{ck}_{\mathrm{L}}, \mathsf{ck}'_{\mathrm{R}}, \mathsf{ck}_{\mathrm{U}}, 1) \ , \\ r \cdot f_{\mathsf{CM}}(\mathsf{m}_{\mathrm{L}}, \mathsf{m}_{\mathrm{R}}, \mathsf{ck}_{\mathrm{L}}, \mathsf{ck}_{\mathrm{R}}, \mathsf{ck}_{\mathrm{U}}, 1) &= f_{\mathsf{CM}}(r \cdot \mathsf{m}_{\mathrm{L}}, \mathsf{m}_{\mathrm{R}}, \mathsf{ck}_{\mathrm{L}}, r \cdot \mathsf{ck}_{\mathrm{R}}, \mathsf{ck}_{\mathrm{U}}, 1) = f_{\mathsf{CM}}(\mathsf{m}_{\mathrm{L}}, r \cdot \mathsf{m}_{\mathrm{R}}, r \cdot \mathsf{ck}_{\mathrm{L}}, \mathsf{ck}_{\mathrm{R}}, \mathsf{ck}_{\mathrm{U}}, 1) \ . \end{aligned}$$

Some of the conditions on p_{m} and p_{ck} in Definition B.1 are vacuously true, since p_{m} and p_{ck} can always be written using an indexing set $I \subseteq \mathbb{Z}$. However, p_{ck} must be a constant in the R-linear case, and the $\mathbb{K}_{\mathrm{U}}$-component of p_{ck} must be a constant in the R-cross-bilinear case. These latter are meaningful restrictions that are used to provide expressions for p_{sc} in Lemma B.3 below.

The generalized Pedersen commitment scheme (Section 5.2) is an example of a sumcheck-friendly commitment scheme where $(f_{\mathsf{CM}}, p_{\mathsf{m}}, p_{\mathsf{ck}})$ is R-bilinear. The commitment schemes in Sections 5.3 to 5.5 are examples where $(f_{\mathsf{CM}}, p_{\mathsf{m}}, p_{\mathsf{ck}})$ is R-cross-bilinear. While we do not work out the details, the commitment scheme in [BFS20] is a sumcheck-friendly commitment scheme where $(f_{\mathsf{CM}}, p_{\mathsf{m}}, p_{\mathsf{ck}})$ is R-linear.

Next, we describe simple representations of p_{sc} when a sumcheck-friendly commitment scheme is R-linear, R-bilinear, or R-cross-bilinear.

Definition B.2. *The polynomial $p_{\mathrm{sc}}(\underline{X})$ is defined as below for each case of Definition B.1.*

- *If $(f_{\mathsf{CM}}, p_{\mathsf{m}}, p_{\mathsf{ck}})$ is R-linear, $p_{\mathrm{sc}}(\underline{X}) := \sum_{\underline{a} \in I^\ell} f_{\mathsf{CM}}(p_{\mathsf{m},\underline{a}}, p_{\mathsf{ck}}, 1) \cdot \underline{X}^{\underline{a}}$.*
- *If $(f_{\mathsf{CM}}, p_{\mathsf{m}}, p_{\mathsf{ck}})$ is R-bilinear, $p_{\mathrm{sc}}(\underline{X}) := \sum_{\underline{a},\underline{b} \in I^\ell} f_{\mathsf{CM}}(p_{\mathsf{m},\underline{a}}, p_{\mathsf{ck},\underline{b}}, 1) \cdot \underline{X}^{\underline{a}+\underline{b}}$.*
- *If $(f_{\mathsf{CM}}, p_{\mathsf{m}}, p_{\mathsf{ck}})$ is R-cross-bilinear, $p_{\mathrm{sc}}(\underline{X}) := \sum_{\underline{a},\underline{b} \in I^\ell} f_{\mathsf{CM}}(p_{\mathsf{m},\mathrm{L},\underline{a}}, p_{\mathsf{m},\mathrm{R},\underline{b}}, p_{\mathsf{ck},\mathrm{L},\underline{a}}, p_{\mathsf{ck},\mathrm{R},\underline{b}}, p_{\mathsf{ck},\mathrm{U}}, 1) \cdot \underline{X}^{\underline{a}+\underline{b}}$.*

Lemma B.3. *In each case of Definition B.2, for every $\underline{r} \in R^\ell$ it holds that $p_{\mathrm{sc}}(\underline{r}) = f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{r}), p_{\mathsf{ck}}(\underline{r}), 1)$.*

Proof. We prove the bilinear case, as the linear and cross-bilinear cases can be proved in a similar way. Using the properties of R-bilinearity,

$$f_{\mathsf{CM}}(p_{\mathsf{m}}(\underline{r}), p_{\mathsf{ck}}(\underline{r}), 1) = \sum_{\underline{a} \in I^\ell} f_{\mathsf{CM}}(p_{\mathsf{m},\underline{a}}, p_{\mathsf{ck}}(\underline{r}), 1) \cdot \underline{r}^{\,\underline{a}} = \sum_{\underline{a},\underline{b} \in I^\ell} f_{\mathsf{CM}}(p_{\mathsf{m},\underline{a}}, p_{\mathsf{ck},\underline{b}}, 1) \cdot \underline{r}^{\,\underline{a}+\underline{b}} = p_{\mathrm{sc}}(\underline{r}) \ .$$

□

One can view the definitions of $p_{\mathrm{sc}}(X)$ in Definition B.2 as the result of "lifting" f_{CM} so that it takes as input *polynomials* over $\mathbb{M}[X_1, \ldots, X_\ell]$ and $\mathbb{K}[X_1, \ldots, X_\ell]$ instead of *scalars* over $\mathbb{M}$ and $\mathbb{K}$, by acting on the coefficients of the polynomials individually. Other work [ACR20] uses a similar idea and shows that operations such as multi-exponentiation and elliptic-curve pairings, which can described by "bilinear gates", can be made to take polynomials of group and field elements as inputs instead of group and field elements.

Next, we describe how the honest prover, given $p_{\mathsf{m}} \in \mathbb{M}[\underline{X}]$ and $p_{\mathsf{ck}} \in \mathbb{K}[\underline{X}]$, and challenges $r_1, \ldots, r_\ell \in \mathcal{C}$, can compute the coefficients of the polynomials $q_1(X_1), \ldots, q_\ell(X_\ell)$ to be sent in the sumcheck protocol by exploiting the homomorphic properties of f_{CM}. In many cases, this is more efficient than the "generic" prover complexity stated in Theorem 4.6 (and in more detail in Lemma 4.9).

Lemma B.4. *Suppose that H is a multiplicative subgroup of R, and $d := \max\{\mathrm{ideg}\big(p_{\mathsf{m}}(\underline{X})\big), \mathrm{ideg}\big(p_{\mathsf{ck}}(\underline{X})\big)\}$ is at most $|H|$.*

- *If f_{CM} is R-linear, the prover in Construction 4.5 can be implemented in $O(d^\ell)$ scalar multiplications in $\mathbb{K}$; $O(d^\ell)$ scalar multiplications in $\mathbb{M}$; $O(d\ell)$ applications of f_{CM}; and $O(d^{\ell+1})$ additions in $\mathbb{C}$.*
- *If f_{CM} is R-bilinear, the prover in Construction 4.5 can be implemented in $O(d^\ell)$ scalar multiplications in $\mathbb{K}$; $O(d^\ell)$ scalar multiplications in $\mathbb{M}$; $O(d^{\ell+1})$ applications of f_{CM}; and $O(d^{\ell+1})$ additions in $\mathbb{C}$.*
- *If f_{CM} is R-cross-bilinear, the prover in Construction 4.5 can be implemented in $O(d^\ell)$ scalar multiplications in each of $\mathbb{K}_{\mathrm{L}}$ and $\mathbb{K}_{\mathrm{R}}$; $O(d^\ell)$ scalar multiplications in each of $\mathbb{M}_{\mathrm{L}}$ and $\mathbb{M}_{\mathrm{R}}$; $O(d^{\ell+1})$ applications of f_{CM}; and $O(d^{\ell+1})$ additions in $\mathbb{C}$.*

As stated in Theorem 4.6, a generic implementation of the prover in Construction 4.5 computes $p_{\mathrm{sc}}(\underline{X})$, partially evaluates it $O(|H|^{\ell-1})$ times, and performs $O(d^\star_{\mathsf{ck}}|H|^{\ell-1})$ additions and scalar-multiplications over $\mathbb{K}$. We now compare this against the costs from Lemma B.4 for the bilinear case. By Definition B.2, p_{sc} has degree $\mathrm{ideg}\big(p_{\mathsf{m}}(\underline{X})\big) + \mathrm{ideg}\big(p_{\mathsf{ck}}(\underline{X})\big)$, which is at most $2d$. Hence p_{sc} has $O(2^\ell d^\ell)$ coefficients, meaning that a *single* partial evaluation would cost $O(2^\ell d^\ell)$ additions and scalar-multiplications in $\mathbb{C}$. Since $O(|H|^{\ell-1})$ partial evaluations are required of the prover from Theorem 4.6, this could lead to a prover algorithm which performs $O(|H|^{\ell-1}2^\ell d^\ell)$ operations in $\mathbb{C}$ and $O(d^\star_{\mathsf{ck}}|H|^{\ell-1})$ operations in $\mathbb{K}$. By contrast, Lemma B.4 shows that, in the bilinear case, the prover need only perform $O(d^{\ell+1})$ operations over $\mathbb{C}$, as well as $O(d^\ell)$ scalar multiplications in $\mathbb{K}$, $O(d^\ell)$ scalar multiplications in $\mathbb{M}$, and $O(d^{\ell+1})$ applications of f_{CM}. Since $d \leq |H|$, the prover algorithm of Theorem 4.6 might perform up to $2^\ell |H|^{\ell-2}$ times as many operations over $\mathbb{C}$ in the worst case. This can be a significant improvement in complexity. For example, when instantiating sumcheck arguments in the prime-order group setting, operations over $\mathbb{K}$ and applications of f_{CM} have the same complexity as operations over $\mathbb{C}$, and dominate the cost of operations over $\mathbb{M}$.

Proof. Fix verifier challenges $\underline{r} = (r_1, \ldots, r_\ell) \in \mathcal{C}^\ell$. At round $i \in [\ell]$, the prover must compute

$$q_i(X_i) = \sum_{\omega_{i+1},\ldots,\omega_\ell \in H} p_{\mathrm{sc}}(r_1, \ldots, r_{i-1}, X_i, \omega_{i+1}, \ldots, \omega_\ell) \ .$$

Each of the polynomials $q_1(X_1), \ldots, q_\ell(X_\ell)$ can be evaluated and the evaluation points sent to the verifier as and when required. We explain how to do this efficiently in the bilinear case, and then explain simplifications that apply in the linear case. The argument for the cross-bilinear case is similar to the bilinear case.

First, we express each $q_i(X_i)$ in terms of partial evaluations of $p_{\mathsf{m}}(\underline{X})$ and $p_{\mathsf{ck}}(\underline{X})$. Then, we give an algorithm that computes the coefficients of $q_1(X_1), \ldots, q_\ell(X_\ell)$ in $O(d^{\ell+1})$ arithmetic operations.

Define coefficients $\mathsf{ck}^{(i)}_{\underline{a}}$ and $\mathsf{m}^{(i)}_{\underline{a}}$ by $p_{\mathsf{ck}}(r_1, \ldots, r_i, X_{i+1}, \ldots, X_\ell) = \sum_{\underline{a} \in I^{\ell-i}} \mathsf{ck}^{(i)}_{\underline{a}} X_{i+1}^{a_{i+1}} \cdots X_\ell^{a_\ell}$ and $p_{\mathsf{m}}(r_1, \ldots, r_i, X_{i+1}, \ldots, X_\ell) = \sum_{\underline{a} \in I^{\ell-i}} \mathsf{m}^{(i)}_{\underline{a}} X_{i+1}^{a_{i+1}} \cdots X_\ell^{a_\ell}$. Note that $\mathsf{ck}^{(i)}_{\underline{a}}$ and $\mathsf{m}^{(i)}_{\underline{a}}$ satisfy the following recurrence relations.

$$\mathsf{ck}^{(i)}_{a_{i+1}\cdots a_\ell} = \sum_{a_i \in I} \mathsf{ck}^{(i-1)}_{a_i \cdots a_\ell} \ r_i^{a_i} \ ,$$
$$\mathsf{m}^{(i)}_{a_{i+1}\cdots a_\ell} = \sum_{a_i \in I} \mathsf{m}^{(i-1)}_{a_i \cdots a_\ell} \ r_i^{a_i} \ .$$

Using the bilinear properties of f_{CM}, we have

$$p_{\mathrm{sc}}(r_1, \ldots, r_{\ell-1}, X_\ell) = q_\ell(X_\ell) = \sum_{a_\ell, b_\ell \in I} f_{\mathsf{CM}}(\mathsf{m}^{(\ell-1)}_{a_\ell}, \mathsf{ck}^{(\ell-1)}_{b_\ell}, 1) \ X_\ell^{a_\ell + b_\ell} \ . \tag{35}$$

Since $q_{\ell-1}(r_{\ell-1}) = \sum_{\omega_\ell \in H} q_\ell(\omega_\ell)$, we have $q_{\ell-1}(r_{\ell-1}) = \sum_{\omega_\ell \in H} \sum_{a,b \in I} f_{\mathsf{CM}}(\mathsf{m}^{(\ell-1)}_a, \mathsf{ck}^{(\ell-1)}_b, 1) \ \omega_\ell^{a+b}$. The set H is a multiplicative subgroup of R, and so we can apply Lemma 3.6 (which is about sums of polynomial evaluations over multiplicative subgroups) and then expand using the recurrence relations. We thus obtain that:

$$\begin{aligned}
q_{\ell-1}(r_{\ell-1}) &= \sum_{\substack{a_\ell, b_\ell \in I \text{ s.t.} \\ a_\ell + b_\ell \equiv 0 \bmod |H|}} f_{\mathsf{CM}}(\mathsf{m}^{(\ell-1)}_{a_\ell}, \mathsf{ck}^{(\ell-1)}_{b_\ell}, 1) \\
&= \sum_{\substack{a_\ell, b_\ell \in I \text{ s.t.} \\ a_\ell + b_\ell \equiv 0 \bmod |H|}} f_{\mathsf{CM}} \left(\sum_{a_{\ell-1} \in I} \mathsf{m}^{(\ell-2)}_{a_{\ell-1} a_\ell} r_{\ell-1}^{a_{\ell-1}}, \sum_{b_{\ell-1} \in I} \mathsf{ck}^{(\ell-2)}_{b_{\ell-1} b_\ell} r_{\ell-1}^{b_{\ell-1}}, 1 \right) \\
&= \sum_{a_{\ell-1}, b_{\ell-1} \in I} r_{\ell-1}^{a_{\ell-1} + b_{\ell-1}} \sum_{\substack{a_\ell, b_\ell \in I \text{ s.t.} \\ a_\ell + b_\ell \equiv 0 \bmod |H|}} f_{\mathsf{CM}} \left(\mathsf{m}^{(\ell-2)}_{a_{\ell-1} a_\ell}, \mathsf{ck}^{(\ell-2)}_{b_{\ell-1} b_\ell}, 1 \right) \ .
\end{aligned}$$

Thus, $q_{\ell-1}(X_{\ell-1}) = \sum_{a_{\ell-1}, b_{\ell-1} \in I} X_{\ell-1}^{a_{\ell-1} + b_{\ell-1}} \sum_{a_\ell, b_\ell \in I \colon a_\ell + b_\ell \equiv 0 \bmod |H|} f_{\mathsf{CM}} \left(\mathsf{m}^{(\ell-2)}_{a_{\ell-1} a_\ell}, \mathsf{ck}^{(\ell-2)}_{b_{\ell-1} b_\ell}, 1 \right)$.

Next, we see that for each $i \in [\ell]$, we can express $q_i(X_i)$ in terms of the coefficients $\mathsf{ck}^{(i-1)}_{a_{i+1} \ldots a_\ell}$ and $\mathsf{m}^{(i-1)}_{a_{i+1} \ldots a_\ell}$:

$$q_i(X_i) = \sum_{a_i, b_i \in I} X_i^{a_i + b_i} \sum_{\substack{a_{i+1}, \ldots, a_\ell, b_{i+1}, \ldots, b_\ell \in I \text{ s.t.} \\ \forall j \in [i+1, \ldots, \ell], \ a_j + b_j \equiv 0 \bmod |H|}} f_{\mathsf{CM}} \left(\mathsf{m}^{(i-1)}_{a_i a_{i+1} \ldots a_\ell}, \mathsf{ck}^{(i-1)}_{b_i b_{i+1} \ldots b_\ell}, 1 \right) \ . \tag{36}$$

We now give an algorithm that computes the coefficients of $q_1(X_1), \ldots, q_\ell(X_\ell)$ in $O(d^{\ell+1})$ arithmetic operations. For $i = 0$, the prover already knows the coefficients $\mathsf{ck}^{(0)}_{a_1 \ldots a_\ell}$ and $\mathsf{m}^{(0)}_{a_1 \ldots a_\ell}$. Then, for each $i \in [\ell]$:

- The prover has the coefficients $\mathsf{ck}^{(i)}_{a_{i+1}\ldots a_\ell}, \mathsf{m}^{(i)}_{a_{i+1}\ldots a_\ell}$ for every $(a_{i+1}, \ldots, a_\ell) \in I^{\ell-i}$.
- The sums in Equation (36) giving the coefficients of $q_i(X_i)$ contain $d^2 \cdot d^{\ell-i}$ terms. Given the values of $\mathsf{ck}^{(i)}_{a_{i+1}\ldots a_\ell}, \mathsf{m}^{(i)}_{a_{i+1}\ldots a_\ell}$ for every $(a_{i+1}, \ldots, a_\ell) \in I^{\ell-i}$ we can compute all of the terms using $d^2 \cdot d^{\ell-i}$ applications of f_{CM} and add them together in $d^2 \cdot d^{\ell-i}$ additions over $\mathbb{C}$ to find the coefficients of $q_i(X_i)$.
- On receiving r_i from the verifier, the prover computes the coefficients $\mathsf{ck}^{(i)}_{a_{i+1}\ldots a_\ell}, \mathsf{m}^{(i)}_{a_{i+1}\ldots a_\ell}$ for every $(a_{i+1}, \ldots, a_\ell) \in I^{\ell-i}$ via the recurrence relations. This requires $d^{\ell-(i-1)}$ scalar multiplications and $d^{\ell-(i-1)}$ additions in $\mathbb{K}$ and $\mathbb{M}$. The prover need not compute $\mathsf{ck}^{(\ell)}$.

The total cost of computing the polynomials $q_1(X_1), \ldots, q_\ell(X_\ell)$ is the sum of a geometric series and is $O(d^{\ell+1})$ applications of f_{CM} and additions in $\mathbb{C}$, and $O(d^\ell)$ additions and scalar multiplications in $\mathbb{K}$ and $\mathbb{M}$.

Further improvements can be made when $(f_{\mathsf{CM}}, p_{\mathsf{m}}, p_{\mathsf{ck}})$ is R-linear. Since $p_{\mathsf{ck}}(\underline{X})$ is a constant in this case, the expression for $q_i(X_i)$ in Equation (36) becomes

$$q_i(X_i) = \sum_{a_i \in I} X_i^{a_i} \sum_{\substack{a_{i+1},\ldots,a_\ell \in I^{\ell-i} \text{ s.t.} \\ \forall j \in [i+1,\ldots,\ell],\ a_j \equiv 0 \bmod |H|}} f_{\mathsf{CM}}\left(\mathsf{m}^{(i-1)}_{a_i a_{i+1}\ldots a_\ell}, p_{\mathsf{ck}}, 1\right) \ .$$

This simplifies to $q_i(X_i) = \sum_{a_i \in I} X_i^{a_i} f_{\mathsf{CM}}\left(\mathsf{m}^{(i-1)}_{a_i 0^{\ell-i}}, p_{\mathsf{ck}}, 1\right)$ since $d \leq |H|$, so that only $d\ell$ applications of f_{CM} are required. □

Acknowledgments

This research was supported in part by a donation from the Ethereum Foundation. Part of the work was conducted while the first author was employed by UC Berkeley.

References

[AC20] Thomas Attema and Ronald Cramer. "Compressed Σ-Protocol Theory and Practical Application to Plug & Play Secure Algorithmics". In: *Proceedings of the 40th Annual International Cryptology Conference*. CRYPTO '20. 2020, pp. 513–543.

[ACDEY19] Mark Abspoel, Ronald Cramer, Ivan Damgård, Daniel Escudero, and Chen Yuan. "Efficient Information-Theoretic Secure Multiparty Computation over $\mathbb{Z}/p^k\mathbb{Z}$ via Galois Rings". In: *Proceedings of the 17th Theory of Cryptography Conference*. TCC '19. 2019, pp. 471–501.

[ACF20] Thomas Attema, Ronald Cramer, and Serge Fehr. *Compressing Proofs of k-Out-Of-n Partial Knowledge*. IACR Cryptology ePrint Archive, Report 2020/753. 2020.

[ACK21] Thomas Attema, Ronald Cramer, and Lisa Kohl. *A Compressed Σ-Protocol Theory for Lattices*. Cryptology ePrint Archive, Report 2021/307. 2021.

[ACR20] Thomas Attema, Ronald Cramer, and Matthieu Rambaud. *Compressed Sigma-Protocols for Bilinear Circuits and Applications to Logarithmic-Sized Transparent Threshold Signature Schemes*. IACR Cryptology ePrint Archive, Report 2020/1447. 2020.

[AFGHO16] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. "Structure-Preserving Signatures and Commitments to Group Elements". In: *Journal of Cryptology* 29 (2 2016), pp. 363–421.

[AL21] Martin R. Albrecht and Russell W. F. Lai. *Subtractive Sets over Cyclotomic Rings: Limits of Schnorr-like Arguments over Lattices*. Cryptology ePrint Archive, Report 2021/202. 2021.

[Abs+20] Mark Abspoel, Ronald Cramer, Ivan Damgård, Daniel Escudero, Matthieu Rambaud, Chaoping Xing, and Chen Yuan. "Asymptotically Good Multiplicative LSSS over Galois Rings and Applications to MPC over $\mathbb{Z}/p^k\mathbb{Z}$". In: *Proceedings of the 26th International Conference on the Theory and Application of Cryptology and Information Security*. ASIACRYPT '20. 2020, pp. 151–180.

[Adj] URL: `https://github.com/adjoint-io/bulletproofs`.

[BBBPWM18] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. "Bulletproofs: Short Proofs for Confidential Transactions and More". In: *Proceedings of the 39th IEEE Symposium on Security and Privacy*. S&P '18. 2018, pp. 315–334.

[BCCGP16] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. "Efficient Zero-Knowledge Arguments for Arithmetic Circuits in the Discrete Log Setting". In: *Proceedings of the 35th Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT '16. 2016, pp. 327–357.

[BCFGRS17] Eli Ben-Sasson, Alessandro Chiesa, Michael A. Forbes, Ariel Gabizon, Michael Riabzev, and Nicholas Spooner. "Zero Knowledge Protocols from Succinct Constraint Detection". In: *Proceedings of the 15th Theory of Cryptography Conference*. TCC '17. 2017, pp. 172–206.

[BCG20] Jonathan Bootle, Alessandro Chiesa, and Jens Groth. "Linear-Time Arguments with Sublinear Verification from Tensor Codes". In: *Proceedings of the 18th Theory of Cryptography Conference*. TCC '20. 2020, pp. 19–46.

[BCGGRS19] Eli Ben-Sasson, Alessandro Chiesa, Lior Goldberg, Tom Gur, Michael Riabzev, and Nicholas Spooner. "Linear-Size Constant-Query IOPs for Delegating Computation". In: *Proceedings of the 17th Theory of Cryptography Conference*. TCC '19. 2019, pp. 494–521.

[BCGRS17] Eli Ben-Sasson, Alessandro Chiesa, Ariel Gabizon, Michael Riabzev, and Nicholas Spooner. "Interactive Oracle Proofs with Constant Rate and Query Complexity". In: *Proceedings of the 44th International Colloquium on Automata, Languages and Programming*. ICALP '17. 2017, 40:1–40:15.

[BCKLN14] Fabrice Benhamouda, Jan Camenisch, Stephan Krenn, Vadim Lyubashevsky, and Gregory Neven. "Better Zero-Knowledge Proofs for Lattice Encryption and Their Application to Group Signatures". In: *Proceedings of the 20th International Conference on the Theory and Application of Cryptology and Information Security*. ASIACRYPT '14. 2014, pp. 551–572.

[BCL20] Jonathan Bootle, Alessandro Chiesa, and Siqi Liu. *Zero-Knowledge Succinct Arguments with a Linear-Time Prover*. IACR Cryptology ePrint Archive, Report 2020/1527. 2020.

[BCOS20] Cecilia Boschini, Jan Camenisch, Max Ovsiankin, and Nicholas Spooner. "Efficient Post-quantum SNARKs for RSIS and RLWE and Their Applications to Privacy". In: *Proceedings of the 11th International Conference on Post-Quantum Cryptography*. PQCrypto '20. 2020, pp. 247–267.

[BCRSVW19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. "Aurora: Transparent Succinct Arguments for R1CS". In: *Proceedings of the 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT '19. Full version available at `https://eprint.iacr.org/2018/828`. 2019, pp. 103–128.

[BDFG20] Dan Boneh, Justin Drake, Ben Fisch, and Ariel Gabizon. *Halo Infinite: Recursive zk-SNARKs from any Additive Polynomial Commitment Scheme*. IACR Cryptology ePrint Archive, Report 2020/1536. 2020.

[BFHVXZ20] Rishabh Bhadauria, Zhiyong Fang, Carmit Hazay, Muthuramakrishnan Venkitasubramaniam, Tiancheng Xie, and Yupeng Zhang. "Ligero++: A New Optimized Sublinear IOP". In: *Proceedings of the 27th ACM Conference on Computer and Communications Security*. CCS '20. 2020, pp. 2025–2038.

[BFL91] László Babai, Lance Fortnow, and Carsten Lund. "Non-Deterministic Exponential Time has Two-Prover Interactive Protocols". In: *Computational Complexity* 1 (1991). Preliminary version appeared in FOCS '90., pp. 3–40.

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. "Checking computations in polylogarithmic time". In: *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*. STOC '91. 1991, pp. 21–32.

[BFS20] Benedikt Bünz, Ben Fisch, and Alan Szepieniec. "Transparent SNARKs from DARK Compilers". In: *Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT '20. 2020, pp. 677–706.

[BHRRS20] Alexander R. Block, Justin Holmgren, Alon Rosen, Ron D. Rothblum, and Pratik Soni. "Public-Coin Zero-Knowledge Arguments with (almost) Minimal Time and Space Overheads". In: *Proceedings of the 18th Theory of Cryptography Conference*. TCC '20. 2020, pp. 168–197.

[BHRRS21] Alexander R. Block, Justin Holmgren, Alon Rosen, Ron D. Rothblum, and Pratik Soni. "Time- and Space-Efficient Arguments from Groups of Unknown Order". In: *Proceedings of the 41st Annual International Cryptology Conference*. CRYPTO '21. 2021, ???–???

[BISW17] Dan Boneh, Yuval Ishai, Amit Sahai, and David J. Wu. "Lattice-Based SNARGs and Their Application to More Efficient Obfuscation". In: *Proceedings of the 36th Annual International Conference on Theory and Applications of Cryptographic Techniques*. EUROCRYPT '17. 2017, pp. 247–277.

[BISW18] Dan Boneh, Yuval Ishai, Amit Sahai, and David J. Wu. "Quasi-Optimal SNARGs via Linear Multi-Prover Interactive Proofs". In: *Proceedings of the 37th Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT '18. 2018, pp. 222–255.

[BLNS20] Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. "A Non-PCP Approach to Succinct Quantum-Safe Zero-Knowledge". In: *Proceedings of the 40th Annual International Cryptology Conference*. CRYPTO '20. 2020, pp. 441–469.

[BMMTV19] Benedikt Bünz, Mary Maller, Pratyush Mishra, Nirvan Tyagi, and Psi Vesely. *Proofs for Inner Pairing Products and Applications*. Cryptology ePrint Archive, Report 2019/1177. 2019.

[CCHLRR18] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, and Ron D. Rothblum. *Fiat–Shamir From Simpler Assumptions*. Cryptology ePrint Archive, Report 2018/1004. 2018.

[CCKP19] Shuo Chen, Jung Hee Cheon, Dongwoo Kim, and Daejun Park. *Verifiable Computing for Approximate Computation*. IACR Cryptology ePrint Archive, Report 2019/762. 2019.

[CDESX18] Ronald Cramer, Ivan Damgård, Daniel Escudero, Peter Scholl, and Chaoping Xing. "SPD$\mathbb{Z}_{2^k}$: Efficient MPC mod 2^k for Dishonest Majority". In: *Proceedings of the 38th Annual International Cryptology Conference*. CRYPTO '18. 2018, pp. 769–798.

[CFFQR20] Matteo Campanelli, Antonio Faonio, Dario Fiore, Anaïs Querol, and Hadrián Rodríguez. *Lunar: a Toolbox for More Efficient Universal and Updatable zkSNARKs and Commit-and-Prove Extensions*. Cryptology ePrint Archive, Report 2020/1069. 2020.

[CFIK03] Ronald Cramer, Serge Fehr, Yuval Ishai, and Eyal Kushilevitz. "Efficient Multi-party Computation over Rings". In: *Proceedings of the 22nd Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT '03. 2003, pp. 596–613.

[CHJKS20] Heewon Chung, Kyoohyung Han, Chanyang Ju, Myungsun Kim, and Jae Hong Seo. *Bulletproofs+: Shorter Proofs for Privacy-Enhanced Distributed Ledger*. Cryptology ePrint Archive, Report 2020/735. 2020.

[CHMMVW20] Alessandro Chiesa, Yuncong Hu, Mary Maller, Pratyush Mishra, Noah Vesely, and Nicholas Ward. "Marlin: Preprocessing zkSNARKs with Universal and Updatable SRS". In: *Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT '20. 2020, pp. 738–768.

[CMS19] Alessandro Chiesa, Peter Manohar, and Nicholas Spooner. "Succinct Arguments in the Quantum Random Oracle Model". In: *Proceedings of the 17th Theory of Cryptography Conference*. TCC '19. 2019, pp. 1–29.

[CMSZ21] Alessandro Chiesa, Fermi Ma, Nicholas Spooner, and Mark Zhandry. *Post-Quantum Succinct Arguments*. Cryptology ePrint Archive, Report 2021/334. 2021.

[CMT12] Graham Cormode, Michael Mitzenmacher, and Justin Thaler. "Practical Verified Computation with Streaming Interactive Proofs". In: *Proceedings of the 4th Symposium on Innovations in Theoretical Computer Science*. ITCS '12. 2012, pp. 90–112.

[COS20] Alessandro Chiesa, Dev Ojha, and Nicholas Spooner. "Fractal: Post-Quantum and Transparent Recursive Proofs from Holography". In: *Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT '20. 2020, pp. 769–793.

[CY20] Alessandro Chiesa and Eylon Yogev. "Barriers for Succinct Arguments in the Random Oracle Model". In: *Proceedings of the 18th Theory of Cryptography Conference*. TCC '20. 2020, pp. 47–76.

[GKR08] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. "Delegating Computation: Interactive Proofs for Muggles". In: *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*. STOC '08. 2008, pp. 113–122.

[GMNO18] Rosario Gennaro, Michele Minelli, Anca Nitulescu, and Michele Orrù. "Lattice-Based zk-SNARKs from Square Span Programs". In: *Proceedings of the 25th ACM Conference on Computer and Communications Security*. CCS '18. 2018, pp. 556–573.

[GN08] Nicolas Gama and Phong Q. Nguyen. "Predicting Lattice Reduction". In: *Proceedings of the 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT '08. 2008, pp. 31–51.

[GNS21] Chaya Ganesh, Anca Nitulescu, and Eduardo Soria-Vazquez. *Rinocchio: SNARKs for Ring Arithmetic*. Cryptology ePrint Archive, Report 2021/322. 2021.

[GT20] Ashrujit Ghoshal and Stefano Tessaro. *Tight State-Restoration Soundness in the Algebraic Group Model*. Cryptology ePrint Archive, Report 2020/1351. 2020.

[JKKZ20] Ruta Jawale, Yael Tauman Kalai, Dakshita Khurana, and Rachel Zhang. *SNARGs for Bounded Depth Computations and PPAD Hardness from Sub-Exponential LWE*. IACR Cryptology ePrint Archive, Report 2020/980. 2020.

[JT20] Joseph Jaeger and Stefano Tessaro. "Expected-Time Cryptography: Generic Techniques and Applications to Concrete Soundness". In: *Proceedings of the 18th Theory of Cryptography Conference*. TCC '20. 2020, pp. 414–443.

[KZG10] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. "Constant-Size Commitments to Polynomials and Their Applications". In: *Proceedings of the 16th International Conference on the Theory and Application of Cryptology and Information Security*. ASIACRYPT '10. 2010, pp. 177–194.

[Kil92] Joe Kilian. "A note on efficient zero-knowledge proofs and arguments". In: *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*. STOC '92. 1992, pp. 723–732.

[LFKN92] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. "Algebraic Methods for Interactive Proof Systems". In: *Journal of the ACM* 39.4 (1992), pp. 859–868.

[LMR19] Russell W. F. Lai, Giulio Malavolta, and Viktoria Ronge. "Succinct Arguments for Bilinear Group Arithmetic: Practical Structure-Preserving Cryptography". In: *Proceedings of the 26th ACM Conference on Computer and Communications Security*. CCS '19. 2019, pp. 2057–2074.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On ideal lattices and learning with errors over rings". In: *Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT '10. 2010, pp. 1–23.

[Lee20] Jonathan Lee. *Dory: Efficient, Transparent arguments for Generalised Inner Products and Polynomial Commitments*. Cryptology ePrint Archive, Report 2020/1274. 2020.

[Mei13] Or Meir. "IP = PSPACE Using Error-Correcting Codes". In: *SIAM Journal on Computing* 42.1 (2013), pp. 380–403.

[Mic07] Daniele Micciancio. "Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-Way Functions". In: *Computational Complexity* 16.4 (2007), pp. 365–411.

[Mon] URL: `https://github.com/monero-project/monero/tree/master/src/ringct`.

[PLS19] Rafaël del Pino, Vadim Lyubashevsky, and Gregor Seiler. "Short Discrete Log Proofs for FHE and Ring-LWE Ciphertexts". In: *Proceedings of the 22nd International Conference on Practice and Theory of Public-Key Cryptography*. PKC '19. 2019, pp. 344–373.

[Piv] *PIVX Implementation of Bulletproofs*. `https://github.com/PIVX-Project/PIVX/tree/Bulletproofs/src/libzerocoin`.

[RR20] Noga Ron-Zewi and Ron Rothblum. "Local Proofs Approaching the Witness Length". In: *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science*. FOCS '20. 2020.

[RV09] Guy N. Rothblum and Salil Vadhan. "Are PCPs Inherent in Efficient Arguments?" In: *Proceedings of the 24th IEEE Annual Conference on Computational Complexity*. CCC '09. 2009, pp. 81–92.

[SS11] Damien Stehlé and Ron Steinfeld. "Making NTRU as Secure as Worst-Case Problems over Ideal Lattices". In: *Proceedings of the 30th Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT '11. 2011, pp. 27–47.

[SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. "Efficient Public Key Encryption Based on Ideal Lattices". In: *Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security*. ASIACRYPT '09. 2009, pp. 617–635.

[Set20] Srinath Setty. "Spartan: Efficient and General-Purpose zkSNARKs Without Trusted Setup". In: *Proceedings of the 40th Annual International Cryptology Conference*. CRYPTO '20. 2020, pp. 704–737.

[Tha13] Justin Thaler. "Time-Optimal Interactive Proofs for Circuit Evaluation". In: *Proceedings of the 33rd Annual International Cryptology Conference*. CRYPTO '13. 2013, pp. 71–89.

[VSBW13] Victor Vu, Srinath Setty, Andrew J. Blumberg, and Michael Walfish. "A hybrid architecture for interactive verifiable computation". In: *Proceedings of the 34th IEEE Symposium on Security and Privacy*. Oakland '13. 2013, pp. 223–237.

[WTSTW18] Riad S. Wahby, Ioanna Tzialla, Abhi Shelat, Justin Thaler, and Michael Walfish. "Doubly-efficient zkSNARKs without trusted setup". In: *Proceedings of the 39th IEEE Symposium on Security and Privacy*. S&P '18. 2018, pp. 926–943.

[Wah+17] Riad S. Wahby, Ye Ji, Andrew J. Blumberg, Abhi Shelat, Justin Thaler, Michael Walfish, and Thomas Wies. "Full Accounting for Verifiable Outsourcing". In: *Proceedings of the 24th ACM Conference on Computer and Communications Security*. CCS '17. 2017, pp. 2071–2086.

[Wes19] Benjamin Wesolowski. "Efficient Verifiable Delay Functions". In: *Proceedings of the 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT '19. 2019, pp. 379–407.

[XZZPS19] Tiancheng Xie, Jiaheng Zhang, Yupeng Zhang, Charalampos Papamanthou, and Dawn Song. "Libra: Succinct Zero-Knowledge Proofs with Optimal Prover Computation". In: *Proceedings of the 39th Annual International Cryptology Conference*. CRYPTO '19. 2019, pp. 733–764.

[ZGKPP17] Yupeng Zhang, Daniel Genkin, Jonathan Katz, Dimitrios Papadopoulos, and Charalampos Papamanthou. "vSQL: Verifying Arbitrary SQL Queries over Dynamic Outsourced Databases". In: *Proceedings of the 38th IEEE Symposium on Security and Privacy*. S&P '17. 2017, pp. 863–880.

[ZXZS20] Jiaheng Zhang, Tiancheng Xie, Yupeng Zhang, and Dawn Song. "Transparent Polynomial Delegation and Its Applications to Zero Knowledge Proof". In: *Proceedings of the 41st IEEE Symposium on Security and Privacy*. S&P '20. 2020, pp. 859–876.

[dalek18] dalek cryptography. *A pure-Rust implementation of Bulletproofs using Ristretto*. 2018. URL: `https://github.com/dalek-cryptography/bulletproofs`.