Multilinear Schwartz-Zippel mod N
and Lattice-Based Succinct Arguments

Benedikt Bünz[*], Ben Fisch[†]

Stanford University, Yale University

June 14, 2023

Abstract

We show that for $\mathbf{x} \xleftarrow{\$} [0, 2^\lambda)^\mu$ and any integer N the probability that $f(\mathbf{x}) \equiv 0 \bmod N$ for any non-zero multilinear polynomial $f \in \mathbb{Z}[X_1, \ldots, X_\mu]$, co-prime to N is inversely proportional to N. As a corollary we show that if $\log_2 N \geq \log_2(2\mu)\lambda + 8\mu^2$ then the probability is bounded by $\frac{\mu+1}{2^\lambda}$. We also give tighter numerically derived bounds, showing that if $\log_2 N \geq 418$, and $\mu \leq 20$ the probability is bounded by $\frac{\mu}{2^\lambda} + 2^{-120}$.

We then apply this Multilinear Composite Schwartz-Zippel Lemma (LCSZ) to resolve an open problem in the literature on succinct arguments: that the *Bulletproofs* protocol for linear relations over classical Pedersen commitments in prime-order groups remains knowledge sound when generalized to commitment schemes that are binding only over short integer vectors. In particular, this means that the Bulletproofs protocol can be instantiated with plausibly post-quantum commitments from lattice hardness assumptions (SIS/R-SIS/M-SIS). It can also be instantiated with commitments based on groups of unknown order (GUOs), in which case the verification time becomes logarithmic instead of linear time.[1]

Prior work on lattice-based Bulletproofs (Crypto 2020) and its extensions required modifying the protocol to sample challenges from special sets of polynomial size. This results in a non-negligible knowledge error, necessitating parallel repetition to amplify soundness, which not only impacts efficiency but also poses issues for the Fiat-Shamir transform. Our analysis shows knowledge soundness for the original Bulletproofs protocol with the exponential-size integer challenge set $[0, 2^\lambda]$ and thus achieves a negligible soundness error without repetition, circumventing a previous impossibility result (Crypto 2021). Our analysis also closes a critical gap in the original security proof of DARK, a GUO-based polynomial commitment scheme (Eurocrypt 2020). Along the way to achieving our result we also define *Almost Special Soundness* (AMSS), a generalization of Special-Soundness. Our main result is divided into two parts: (1) that the Bulletproofs protocol over generalized commitments is AMSS, and (2) that AMSS implies knowledge soundness. This framework serves to simplify the application of our analytical techniques to protocols beyond Bulletproofs in the future.

1 Introduction

The famous DeMillo-Lipton-Schwartz–Zippel (DLSZ) lemma [DL77; Zip79; Sch80] states that for any field $\mathbb{F}$, non-empty finite subset $S \subseteq \mathbb{F}$, and non-zero μ-variate polynomial f over $\mathbb{F}$ of total degree d, the number of zeros of f contained in S^μ is bounded by $d \cdot |S|^{\mu-1}$ (or equivalently, the probability that $f(\mathbf{x}) = 0$ for $\mathbf{x}$ sampled uniformly from S^μ is bounded by $\frac{d}{|S|}$). For $\mu = 1$ this simply follows from the Fundamental Theorem of Algebra, but for multivariate polynomials, the number of zeros over the whole field could be unbounded. The computational significance of this lemma is that sampling an element from S only takes $n \cdot \log_2(|S|)$ random bits but the probability of randomly sampling a zero of f from S^μ is inversely proportional to $|S|$, which is exponential in the number of random bits. One of its original motivations was an efficient randomized algorithm for polynomial identity testing, but it has since found widespread application in computer science [KI04].

The classical lemma applies more broadly to integral domains, but not to more general commutative rings such as the ring $\mathbb{Z}_N$. As a simple counterexample, over the ring of integers modulo $N = 2p$ the polynomial $f(X) = pX \bmod N$ vanishes on half of the points in $[0, N)$. This counterexample exploits the fact that f is of the form $f(X) = u \cdot g(X)$ where u is a zero-divisor. There are also simple counterexamples for f co-prime to N: setting $N = 2^\lambda$ the polynomial $f(X) = X^\lambda \bmod N$ vanishes on half of the points in $[0, N)$. However, there are no such counterexamples when f is both multilinear and co-prime to N. In fact, we will show in this work the probability a random vector from $S^\mu = [0, m)^\mu$ is a zero of a μ-linear polynomial (multilinear with μ variables) co-prime to N is negligible in the minimum of $\log m$ and $\log N$. As we will show in our main result, this special case of f and N still has a surprisingly powerful

---

[1]This paper supersedes an earlier pre-print on the LCSZ [BF22] and also incorporates content published in the updated EPRINT of DARK [BFS19].

application to cryptography that resolves multiple recent open questions in the area of succinct arguments.

The DLSZ lemma has previously been extended to commutative rings by restricting the set S to special subsets in which the difference of any two elements is not a zero divisor [BCPS15]. For example, in the case of $\mathbb{Z}_N$ this would require the difference of any two elements in S to be co-prime to N. All examples of such sets have $O(\log N)$ size. Our present work explores the setting where S is the contiguous interval $[0, m)$ and thus does not have this restriction.

As a warmup, it is easy to see that any univariate linear polynomial $f(X) = c \cdot X + b$ co-prime to N has at most one root modulo N. If there were two such roots $x_1 \not\equiv x_2 \bmod N$ then $c(x_1 - x_2) \equiv 0 \bmod N$ implies c is a zero divisor (i.e., $gcd(c, N) \neq 1$). Furthermore, $c \cdot x_1 \equiv -b \bmod N$ implies $-b = c \cdot x_1 + q \cdot N$ for some $q \in \mathbb{Z}$, and thus, $gcd(c, N)$ also divides b. This would contradict the co-primality of f and N. So for x uniformly distributed in $S = [0, m)$ the probability of $f(x) \equiv 0 \bmod N$ in this case is indeed at most $\frac{1}{|S|}$. Unfortunately, this does not generalize nicely to polynomials of arbitrary degree as illustrated by the counterexample above. On the other hand, we are able to generalize the lemma in a meaningful way to multivariate *linear* polynomials (i.e, at most degree 1 in each variable). We bound the probability of sampling a zero from $S^\mu = [0, m)^\mu$ of a μ-linear polynomial co-prime to N by $\epsilon + \frac{\mu}{|S|}$, where ϵ is tightly bounded by a product of regularized beta functions.

We also formulate an inverse lemma showing that for all sufficiently large N, ϵ is negligibly small. In particular, for $\log N \geq 8\mu^2 + (1 + \log \mu)\lambda$, ϵ is at most $2^{-\lambda}$, showing that the probability decays exponentially. Our technique for deriving this threshold lower bound $t(\lambda, \mu)$ on N for a target λ formulates $t(\lambda, \mu)$ as the objective function of a knapsack problem. We derive an analytical solution by deriving bounds on the regularized beta function. We also apply a knapsack approximation algorithm to find tighter values of $t(\lambda, \mu)$ for specific values of μ and λ.

1.1 Bulletproofs for short pre-images

Using the multi-linear composite Schwartz-Zippel lemma (LCSZ), we can prove that a generalization of the Bulletproofs Polynomial Commitment [BCCGP16; BBBPWM18; WTsTW18] is secure even with large challenge sets. The generalization allows for commitments to "short" (i.e., bounded norm) integer vectors. This includes groups of unknown order, such as the RSA group or class groups, as well as lattice-based commitments (i.e. Ajtai commitments based on the Integer SIS or Ring-SIS assumptions). The instantiation using commitments based on groups of unknown order is essentially a variation of DARK [BFS20], and our analysis closes a vital gap in the security proof that was first discovered by [BHRRS21].[2] Unlike the fix proposed by [BHRRS21], our analysis covers the original DARK protocol and enables the use of a large challenge space instead of relying on binary challenges. Our analysis is also the first to show that lattice-based Bulletproofs [BLNS20] (i.e., Bulletproofs instantiated with Ajtai commitmenets) with a challenge space of exponential size (e.g., $[0, 2^\lambda)$) has a negligible knowledge error (without parallel repetition). All previous attempts [BLNS20; ACK21b; AL21] had analyzed small, specially constructed challenge sets, which result in a knowledge error $o(\frac{1}{\mathsf{poly}(\lambda)})$, and thus these protocols used parallel repetition to amplify soundness. In fact, [AL21] give an impossibility result, showing that the approach of specially constructing such sets is limited and unlikely to result in a negligible soundness error. Furthermore, parallel-repetition is not always compatible with the Fiat-Shamir transform [AFK21; Wik21].

Both DARK, as well as the previous analysis of lattice-based Bulletproofs, consider the special-soundness of the protocol. Informally, special-soundness requires that an extractor can extract an opening to a vector commitment from *any* tree of accepting transcripts with distinct challenges at any branch. By the forking lemma, which shows how to generate such trees, special-soundness implies knowledge soundness. The special-soundness requirement includes trees where the differences of challenges are arbitrary, and thus it is difficult to tightly bound the size of the extracted opening. Bulletproofs over commitments binding only to short vectors fails to satisfy special-soundness with large challenges. However, using the LCSZ, we are able to show that Bulletproofs satisfies a more general property we call *almost-special soundness* (AMSS), which we show also implies knowledge soundness.

Almost Special Soundness We introduce Almost special soundness (AMSS) as a generalization of Special-Soundness. AMSS protocols are multi-round protocols where every round is associated with a commitment that is binding over openings to messages in a set $\mathcal{W}$. This does not have to be an explicit message sent to the verifier but roughly represents the prover's state at a round of the protocol. At a very high level, protocols are AMSS if there exists an algorithm that extracts from any forking transcript tree an opening to these commitments, and if the opening is not inside a subset $\mathcal{W}' \subset \mathcal{W}$ then re-running (or completing) the protocol starting from this extracted state on fresh challenges would fail (with overwhelmingly high probability) to result in a transcript accepted by the verifier. One additional key requirement, stated informally here, is that re-running the protocol on the *same* challenges would either result in the same transcript or a break of the commitment scheme. We leverage these combined properties to show that AMSS protocols are knowledge-sound. We then show that Bulletproofs with commitments to short pre-images are almost-special sound, which relies on the inverse LCSZ.

---

[2]The same analysis we provide in this paper also applies to DARK in its original form, which is now included in an updated appendix of the original DARK paper.

As a brief overview, we relax the commitment scheme to allow for *rational openings* of a commitment C to a rational $\mathbf{f}/N$ such that $\mathsf{com}(\mathbf{f}) = N \cdot C$. The Bulletproofs verifier accepts the protocol transcript if the final message is a "small" integer. This suggests that if the prover were to run the honest protocol starting with $\mathbf{f}/N$ as its private state, then its success would imply $f(\mathbf{r}) \equiv 0 \bmod N$ where f is a multilinear polynomial with the coefficients defined by $\mathbf{f}$ and $\mathbf{r}$ are the verifier challenges. We can use the inverse LCSZ to show that if N is too "large" and $\mathbf{r}$ is sampled randomly then this probability is negligible. Making this analysis formal is non-trivial, and we present a summary of the ideas in the technical overview below.

Along the way to showing that AMSS implies knowledge soundness we also introduce a variant of the standard forking lemma, which we call the *path predicate forking lemma*. This lemma shows the existence of a PPT algorithm to generate a transcript tree satisfying additional properties for AMSS protocols that enable the efficient extraction of a witness.

Fiat-Shamir transform The Fiat-Shamir transform is a method for transforming an interactive protocol with a public coin verifier into a non-interactive publicly verifiable protocol by replacing the verifier public-coin challenges with hashes of the prover's messages. Recent work [ACK21a; Wik21] has shown that the Fiat-Shamir transform is secure for multi-round special-sound protocols. However, the security proof does not translate immediately to almost special-sound (AMSS) protocols. We prove security of the Fiat-Shamir transform for AMSS protocols with *computationally unique commitments*, where it is infeasible to open two distinct commitments to the same message. The deterministic variants of DARK and Ajtai commitments have this property.

1.2   Related work

Lattice-based Bulletproofs Most practical lattice-based succinct proof systems have focused on single-round protocols. These protocols [LNS21; ALS20; ENS20; NS22] have $o(\sqrt{(m)})$ communication complexity. Bulletproofs [BCCGP16; BBBPWM18] is a multiround argument of knowledge for the opening of discrete-logarithm-based vector commitments. Using a folding technique, one can prove knowledge of a vector of size m using only $2\log_2(m)$ λ-bit sized messages. Bootle et al. [BLNS20] ported the protocol to the lattice setting using Ring-SIS commitments. They rely on a small challenge set (the challenges are monomials with binary coefficients), such that the differences of any two challenges divide 2 in the ring. This allows them to extract an opening to the lattice-based vector commitment from a tree of transcripts. However, the small challenge space as well as the blow-up in the opening size, increase the total opening size to $O(\lambda^2 \log^2(m))$, instead of $O(\lambda \log(m))$ for normal Bulletproofs. [ACK21b] generalize the techniques of [BLNS20] for more general challenge sets, i.e. special sets where differences in challenges are invertible. This also enables a generalization from Ring-SIS to other modules. [AL21] further generalizes the challenge set to so-called subtractive sets. These are sets such that for any t-sized subset T and $c \in T$, $\prod_{c' \in T \setminus \{c\}}(c - c')$ divides a ring-element s. They construct a $(2, 3)$ sized set of size $O(m)$ for a 2^m extension of the integers. This means that the challenge set is only linear in the bits of the message space as opposed to being exponential, necessitating parallel repetition to amplify soundness. Beyond increasing the communication this also poses difficulties for the security of the Fiat-Shamir transform [AFK21; Wik21]. Moreover, [AL21] show that the subtractive sets they construct are nearly optimal, implying the limits of this approach. [AL21] state that "unless fundamentally new techniques are discovered" the small challenge sets and thus, the resulting $\Omega(\frac{1}{n})$ knowledge error, "represents a barrier to practically efficient lattice-based succinct arguments in the Bulletproof framework".

Our work overcomes this barrier, showing that lattice-based Bulletproofs can be instantiated with an exponential-sized challenge set and a negligible knowledge error. The analysis is based on our variant of the DLSZ lemma for multilinear polynomials mod composite N (i.e., our LCSZ lemma). It also demonstrates compatibility with the Fiat-Shamir transform. Specifically, we use the integer set $[0, 2^\lambda)$ as in the original Bulletproofs protocol. To achieve the same soundness level prior variants of lattice-Bulletproofs [BLNS20; ACK21b; AL21] required $O(\lambda/\log(m))$ parallel repetitions where m is the degree of the cyclotomic polynomial of the ring used for M-SIS commitments. On the other hand, we require a larger modulus for the M-SIS commitment. With vectors of length n over $\mathbb{Z}_p$ the modulus q in our case is $O(\lambda \log n + \log p)$ bits, whereas when using a smaller challenge set of size m the modulus q can be $O(\log m \log n + \log p)$. Our performance benefit is thus optimized when $\log p \approx \lambda \log n$. Unfortunately, this means that for $p \ll n^\lambda$ there isn't yet a clear performance benefit over prior work based on subtractive sets. An interesting direction for future work is to look at ways to pack the coefficients of vectors over a smaller modulus $p' \ll p$ into vectors over the larger modulus and still make use of the linear form opening. Another direction is to use an exponential-size challenge set of smaller norm elements (i.e., over the polynomial ring rather than integers), but this would require further generalizations of the LCSZ lemma.

DARK and Groups of Unknown Order The DARK Polynomial Commitment [BFS20] is a polynomial commitment with succinct verification using groups of unknown order. If instantiated with class groups (see [BH01]) the protocol does not require a trusted setup. The scheme is particularly interesting because of the short proof sizes. Unfortunately, the original scheme had a gap in the security proof that was first discovered by [BHRRS21]. [BHRRS21] provide a fix by using binary challenges but this blows up the proof size. [AGLMS22] provide another

fix which allows for large challenges but requires significantly larger parameters, resulting in a proof system that has a setup and prover runtime that are quadratic in the length of the prover's witness.

2 Technical overview

The regular DLSZ is relatively simple to prove. Consider the special case of a multilinear polynomial over a field. As a base case, a univariate linear polynomial has at most one root over the field. For the induction step, express $f(X_1, ..., X_{\mu+1}) = g(X_1, ..., X_\mu) + X_{\mu+1}h(X_1, ..., X_\mu)$ for random variables $X_1, \ldots, X_{\mu+1}$. The probability that $h(x_1, ..., x_n) = 0$ over random x_i sampled from S is at most $\mu/|S|$ by the inductive hypothesis, and if $h(x_1, ..., x_\mu) = w \neq 0$ and $g(x_1, \ldots, x_\mu) = u$, then $u + X_{\mu+1}w$ has at most one root (base case). By the union bound, the overall probability is at most $\mu/|S| + 1/|S| = (\mu + 1)/|S|$. This simple proof does not work for multilinear polynomials modulo a composite integer. The base case is the same for f coprime to N, which has at most one root. However, in the induction step, it isn't enough that $h(x_1, ..., x_\mu) \neq 0$ as it still may be a zero divisor, in which case the polynomial $u + X_{\mu+1}w$ is not necessarily coprime to N and the base case no longer applies. The number of roots depends on $\gcd(u + X_{\mu+1}w, N)$ and our new analysis takes into account its distribution. For each prime divisor p_i of N, the highest power of p_i that divides $u + X_{\mu+1}w$ follows a geometric distribution. Using a modified inductive argument, we are able to show that the probability $f(x_1, \ldots, x_\mu) \equiv 0 \bmod p^r$ is bounded by the probability that $\sum_{i=1}^n Z_i \geq r$ for $i.i.d.$ geometric variables with success parameter $1 - \frac{1}{p}$. This probability is equal to a $I_{\frac{1}{p}}(r, \mu)$ where I is the regularized beta function. Furthermore, by CRT this probability is independent for each prime factor of N, and thus, the overall probability can be bounded by a product of regularized beta functions.

Inverse lemma While our main theorem gives a tight bound on the probability for particular values of N, μ, and m, cryptographic applications require finding concrete parameters such that the probability is exponentially small in a security parameter λ. Concretely, we want to find a value N^* such that for all $N \geq N^*$ the probability that $f(X_1, \ldots, X_\mu) \equiv 0 \bmod N$ is bounded by $2^{-\lambda}$. To do this, we first derive simple and useful bounds for the regularized beta function:

- $I_{\frac{1}{p}}(r, \mu) \leq \left(\frac{n}{p}\right)^r$ for $p \geq 2\mu$

- $I_{\frac{1}{p}}(r, \mu) \leq \frac{r^n}{p^r}$ for $r \geq 2\mu$

- $\log(I_{1/p}(r - 1, \mu)) - \log(I_{1/p}(r, \mu))$ is non-increasing in r for any $p > \mu$ and for $r = 1$ in p.

We then formulate finding N^* as an optimization problem. N^* is the maximum value of N such that the probability of $f(\mathbf{x}) \equiv 0 \bmod N$ is greater than $2^{-\lambda}$. For any N let $S(N)$ denote the set of pairs (p, r) where p is a prime divisor of N with multiplicity r. Taking the logarithm of both the objective and the constraint yields a knapsack-like constraint maximization problem where the objective is $\log(N^*) = \sum_{(p_i, r_i) \in S(N^*)} r_i \cdot \log(p_i)$ and the constraint is $\sum_{(p_i, r_i) \in S(N^*)} -\log(I_{\frac{1}{p_i}}(r, \mu)) \leq \lambda$. Using the bounds on $I_{\frac{1}{p_i}}$ and several transformations of the problem we show that any optimal solution to this problem must be bounded by $t = 8\mu^2 + \log_2(2\mu)\lambda$, which in turn implies that $N^* \leq 2^t$.

Tighter computational solution We further show that a simple greedy knapsack algorithm computes an upper bound to the knapsack problem. The algorithm uses the fact that $\frac{\log(p)}{\log(I_{1/p}(r-1,\mu)) - \log(I_{1/p}(r,\mu))}$ the so-called marginal density of each item is non increasing over certain regions. Adding the densest items to the knapsack computes an upper bound to the objective. We run the algorithm on a large number of values for μ and λ and report the result.

2.1 Bulletproofs for Short Pre-Images and Almost Special Soundness

In the Bulletproofs Inner Product Argument a prover convinces a verifier that it knows the opening $\mathbf{f} \in \mathbb{F}_p^n$ to a homomorphic commitment $C = \mathsf{com}(\mathbf{f}) \in \mathbb{G}$ where $\mathbb{G}$ is a prime-order group. At a very high level, it does this by iteratively computing $\vec{f'} = \mathbf{f}_L + x \cdot \mathbf{f}_R \in \mathbb{F}^{n/2}$, where $\mathbf{f}_L, \mathbf{f}_R$ are the left and right half of $\mathbf{f}$ respectively and $x \in \mathbb{F}$ is a verifier generated challenge, and sending a commitment to the new $\mathbf{f}'$ to the verifier. After $\log_2(n)$ rounds, the prover sends a single field element as the final message. This naturally generalizes to homomorphic commitments that map vectors in $\mathbb{Z}^n$ to a group $\mathbb{G}$, where $\mathbb{G}$ is either prime or unknown. We consider schemes that may only be binding to *short* vectors in $\mathbb{Z}^n$, or more generally binding under what we call *short rational openings* that open C to $\mathbf{h} = \mathbf{f}/N$ by showing $N \cdot C = \mathsf{com}(\mathbf{f})$, where the numerator and denominator of $\mathbf{h}$ have bounded norms in reduced form. The Bulletproofs protocol operates in exactly the same way over integer vectors but using these more general instantiations of the commitment scheme. Additionally, the verifier checks that the final message is a "small" integer. In the soundness analysis, we leverage the fact that the final integer is small in order to bound the numerator/denominators of the prover's witness. This is where we invoke the new LCSZ lemma for multilinear polynomials. Intuitively, if the prover's private state at the start of protocol is a rational vector $\mathbf{f}/N$, i.e. $\mathsf{com}(\mathbf{f}) = N \cdot C$, then running protocol will result in a final integer y that is the evaluation of a multi-linear polynomial f with coefficients $\vec{f}$ at the $\log_2(n)$ challenges

$\mathbf{x} \in \mathbb{Z}^{\log_2(n)}$: $y = \frac{f(\mathbf{x})}{N}$. Note, that since y is an integer, we can deduce that $f(\mathbf{x}) \equiv 0 \mod N$. We LCSZ to show that if N is too 'large', then this probability is small.

For commitments binding over $\mathbb{Z}_p$, Bulletproofs satisfies special-soundness: there exists an efficient extractor that can extract a witness (i.e., an opening to the input commitment) from *any* forking tree of transcripts. Special-soundness implies knowledge-soundness by the classic *forking lemma*, which shows how to generate a transcript tree in polynomial time. Unfortunately, Bulletproofs with commitments that are only binding over bounded norm openings fails to satisfy special-soundness because the opening extracted from a forking tree of transcripts may have a very large norm. On the other hand, it turns out that we can leverage the intuition above in order to bound the size of extracted openings. Along the way, we introduce a new notion called almost-special soundness.

Almost Special Soundness   A k-ary transcript tree for a μ-round public-coin interactive proof labels each node of a k-ary μ-depth tree with a prover message and each edge with a verifier public-coin challenge so that the labels along any root-to-leaf path in the tree form a valid transcript between the prover and verifier that would cause the verifier to accept. An interactive proof for a relation $\mathcal{R}$ is $k^{(\mu)}$-special-sound if there exists an extractor that can efficiently extract a witness w from any k-ary tree of protocol transcripts for input x so that $(\mathsf{x}, \mathsf{w}) \in \mathcal{R}$. Bulletproofs for standard Pedersen vector commitments has this property for 3-ary trees, but not for vector commitments with bounded-norm openings because the extractor may obtain an opening that has a too large norm. On the other hand, if a prover running the protocol honestly were to start in its head with an opening of the commitment that has a too large norm there is a negligible probability over the random challenges that it would result in a valid transcript. The probability analysis relies on LCSZ.

This observation suggests the following strawman extraction analysis: show that one of the valid transcripts in the tree corresponds to running the honest prover on the extracted opening/witness, and conclude that if the extracted opening had large norm then it would have a negligible probability of resulting in a valid transcript. A fallacy in this argument is that the extracted witness is computed from the transcripts, and is thus dependent on the challenges appearing in the transcripts, whereas running the prover on a large norm witness only results in an invalid transcript with high probability over challenges sampled independently from the witness. To address this we could attempt the following: generate a 4-ary tree T via rejection sampling from polynomially many random simulations, extract an opening w^* from its 3-ary left subtree T_L, show that some transcript tr in $T \setminus T_L$ is the result of running the prover on w^*. The extracted witness is now independent of the challenges appearing in tr and was well-defined during the generation of T, after T_L was created and before tr was added. Given that T was generated via polynomially many random simulations we can argue this event had a negligible probability of occurring.

The remaining challenge, however, is to show that some transcript tr in $T \setminus T_L$ is consistent with running the honest prover on the extracted witness w^*. It turns out that for protocols like Bulletproofs this is only true when the extracted witness is already a valid opening of the input commitment within the space over which the scheme is binding, which seems to bring us back to square one. For example, the transcript for a single round Bulletproofs protocol over an input commitment C to a vector in $\mathbb{Z}^2$ and commitment basis $\mathbf{g} = (g_L, g_R) \in \mathbb{G}^2$ consists of $(C, C_L, C_R, r, f') \in \mathbb{G}^3 \times \mathbb{Z}$ such that $f' \cdot (g_R + rg_L) = C_R + r^2 C_L + rC$. Given an opening $\mathbf{f} = (f_L, f_R)$ such that $\mathsf{com}(\mathbf{f}) = \langle \mathbf{f}, \mathbf{g} \rangle = C$ and $\langle (0, f_L), \mathbf{g} \rangle = C_L$ and $\langle (f_R, 0), \mathbf{g} \rangle = C_R$ then "re-running" the protocol on the openings with challenge r gives $f^* = f_L + r \cdot f_R$ such that $f^* \cdot (g_R + rg_L) = C_R + r^2 C_L + rC = C^*$. If f^* is sufficiently small then this implies $f^* = f'$, otherwise it is a break of the commitment scheme as it provides conflicting openings to C^*. However, f^* is only guaranteed to be small if the extracted opening $\mathbf{f}$ has low norm.

To get around this issue, we can increase the parameters of the commitment scheme so that it is binding over slightly larger openings, and we will argue level by level that the extracted openings remain small (using the independent path and the fact that re-generation of transcripts either returns the same commitments/messages or a break of the commitment scheme). In other words, while we want to show that every extracted value remains below some norm bound A, each extraction step produces a value that might be as large as some bound $B > A$, but for which the scheme is still binding, and thus re-running the protocol on this value will conflict with some path in the transcript tree with overwhelmingly high probability if too much larger than A. This allows us to bound the norm growth at each level by some sufficiently small value $C \in (A, B)$. Crucially, while the growth from A to B is at least quadratic in A, the growth from A to C will be constant.

We generalize this to the notion of *Almost Special Sound* (AMSS) protocols and replace the bounds A and B with arbitrary predicates ϕ_a and ϕ_b. We prove that all protocols with this structure are knowledge sound, just like special-sound protocols, where the knowledge error is dependent on the probability that a random completion of a transcript starting from a message that fails predicate ϕ_a results in a valid transcript. Intuitively, this captures the fact that once the adversary has a private state that fails the desired extraction predicate, it will fail with overwhelming probability over fresh challenges to complete the proof transcript successfully.

Our proof that AMSS implies knowledge-soundness relies on a lemma that we call the *path predicate forking lemma* (Lemma 9). The usual forking lemma shows how to generate forking transcript tree, which in special-sound protocols can be passed directly to the extractor. In our case, we need to generate transcript tree that satisfies additional predicates on each node. In

the standard forking lemma [BCCGP16], the predicate would simply be that challenges on each child of a node in the transcript tree are distinct from previous challenges. Our new lemma considers more general predicates, which may depend on partial transcripts that have already been generated in the course of the transcript generation algorithm. The analysis is similar and uses a union bound over all polynomial steps of the transcript tree generation process.

Fiat-Shamir transform The analysis showing that AMSS protocols are knowledge-sound critically relies on the fact that in the transcript tree generation process for an interactive protocol, the challenges on any given branch are sampled uniformly and independently. This is used to show that the transcript tree generated satisfies a certain property with overwhelming probability. The Fiat-Shamir transform converts an interactive public-coin protocol into a non-interactive protocol by replacing the verifier's messages with a transcript hash. The problem with applying the Fiat-Shamir transform to an AMSS protocol is that the adversary can now grind the challenges in each round when generating a transcript, breaking uniformity and independence of challenges. Using a union bound, we could still bound the probability that the transcript tree does not have the desired property, but this would result in a factor Q^μ loss where Q is the number of queries an adversary performs, and μ is the number of rounds in the protocol. However, we can instead focus on protocols where grinding challenges is impossible for the adversary. To do this, we introduce the notion of *computationally unique* commitments. In a computationally unique commitment scheme, it is infeasible to open two distinct commitments to the same message. We prove that this property is held by a large class of deterministic homomorphic commitment schemes, which include those from groups of unknown order and lattice assumptions. We prove security of the Fiat-Shamir transform for AMSS protocols in the random oracle model with computationally unique commitments. This analysis is in Appendix D.

3   Main theorem statement (LCSZ)

Theorem 1 (Multilinear Composite Schwartz-Zippel (LCSZ))**.** Let $N = \prod_{i=1}^{\ell} p_i^{r_i}$ for distinct primes $p_1, ..., p_\ell$. Let f be any μ-linear integer polynomial co-prime to N. For any integer $m > 1$ and $\mathbf{x}$ sampled uniformly from $[0, m)^\mu$:

$$\mathbb{P}_{\mathbf{x} \leftarrow [0,m)^\mu}[f(\mathbf{x}) \equiv 0 \bmod N] \leq \frac{\mu}{m} + \prod_{i=1}^{\ell} I_{\frac{1}{p_i}}(r_i, \mu),$$

where $I_{\frac{1}{p}}(r, \mu) = (1 - \frac{1}{p})^\mu \sum_{j=r}^{\infty} \binom{\mu+r-1}{r} \left(\frac{1}{p}\right)^j$ is the regularized beta function.

Remark 1. The regularized beta function characterizes the tail distribution of the sum of independent geometric random variables. If $Y = \sum_{i=1}^{\mu} Z_i$ where each Z_i is an independent geometric random variable with parameter ϵ then $P[Y \geq r] = I_{1-\epsilon}(r, \mu)$. Y is a negative binomial variable with parameters ϵ, μ.

Remark 2. A close reading of the proof reveals that if $m = N$, then the theorem statement simplifies to $\mathbb{P}_{\mathbf{x} \leftarrow [0,N)^\mu}[f(\mathbf{x}) \equiv 0 \bmod N] \leq \prod_{i=1}^{\ell} I_{\frac{1}{p_i}}(r_i, \mu)$. This is because $\mathbf{x}$ is uniform mod N and thus uniform mod any $N^*|N$.

Remark 3. The theorem is nearly tight for all N. Setting $f(\mathbf{x}) = \prod_{i=1}^{\mu} x_i$ and $m = N$ gives $P_{\mathbf{x} \leftarrow [0,m)^\mu}[f(\mathbf{x}) \equiv 0 \bmod N] = \mathbb{P}_{\mathbf{x} \leftarrow [0,N)^\mu}[f(\mathbf{x}) \equiv 0 \bmod N] = \prod_{i=1}^{\ell} I_{\frac{1}{p_i}}(r_i, \mu)$.

Remark 4. $1 - e^{-\mu/p_i} \leq I_{\frac{1}{p_i}}(1, \mu) = 1 - (1 - \frac{1}{p_i})^\mu \leq \frac{\mu}{p_i}$. Hence, for square-free N the probability in Theorem 1 is upper bounded by $\frac{\mu}{m} + \frac{\mu^\ell}{N}$, but for $\ell > 1$ this is a loose upper bound unless $\mu \ll p_i$ for all $p_i|N$. For $\ell = 1$ (i.e., prime N), Theorem 1 coincides with the Schwartz-Zippel lemma.

Remark 5. $I_{\frac{1}{p_i}}(r_i, 1) = \left(\frac{1}{p_i}\right)^{r_i}$. Hence, for $\mu = 1$, the bound in Theorem 1 is $\frac{1}{N} + \frac{1}{m}$.

We defer the proof of Theorem 1 to Appendix A.1.

4   Inverse LCSZ

Theorem 1 (LCSZ) bounds the probability $\mathbb{P}_{\mathbf{x} \leftarrow [0,m)^\mu}[f(\mathbf{x} \equiv 0 \bmod N]$ for given values of μ, N, and m, which has the form $\frac{\mu}{m} + \delta_{N,\mu}$. In the case that N is prime, $\delta_{N,\mu} = \frac{\mu}{N}$, which agrees with the standard Schwartz-Zippel lemma applied to μ-linear polynomials. The term $\delta_{N,\mu}$ for composite N, which is dependent on both μ and the factorization of N, has a complicated closed form expression in terms of a product of regularized beta functions.

This section analyzes the inverse: for a given $\mu, \lambda \in \mathbb{N}$ what size threshold $t(\lambda, \mu) \in \mathbb{N}$ is sufficient such that $\delta_{N,\mu} \leq 2^{-\lambda}$ for all $N \geq t(\lambda, \mu)$? In other words:

$$t(\lambda, \mu) := \sup\{N \in \mathbb{N} : \prod_{(p,r) \in S(N)} I_{\frac{1}{p}}(r, \mu) \geq 2^{-\lambda}\}. \qquad (t(\lambda, \mu) \text{ def})$$

For $\mu = 1$, since $I_{1/p}(r, 1) = \frac{1}{p^r}$ and $\prod_{(p,r) \in S(N)} I_{\frac{1}{p}}(r, \mu) = \frac{1}{N}$, it is easy to see that $t(\lambda, \mu) = 2^\lambda$. For $\mu \geq 2$, the value of $t(\lambda, \mu)$ (or even an upper bound) is not nearly as easy to derive. For

the rest of this section we will focus on this $\mu \geq 2$ case. We will analytically derive an upper bound to $t(\lambda, \mu)$, showing that $\log t(\lambda, \mu) \in O(\mu^{2+\epsilon} + \frac{\lambda}{\epsilon})$ for any $\epsilon \geq \log_\mu(2)$.

Theorem 2 (Inverse LCSZ). For all $\mu \geq 2$, $\epsilon \geq \log_\mu(2)$, and all N such that

$$\log N \geq 4\mu^{2+\epsilon} + (1 + \frac{1}{\epsilon}) \cdot \lambda.$$

we have that for any μ-linear polynomial f that is coprime with N

$$\mathbb{P}_{x \leftarrow [0,m)^\mu}[f(x) \equiv 0 \bmod N] \leq 2^{-\lambda} + \frac{\mu}{m}.$$

By setting $\epsilon = \log_\mu(2)$ we get:

Corollary 1. For all N such that

$$\log N \geq 8\mu^2 + \log_2(2\mu) \cdot \lambda.$$

we have that for any n-linear polynomial f that is coprime with N

$$P_{x \leftarrow [0,m)^\mu}[f(x) \equiv 0 \bmod N] \leq 2^{-\lambda} + \frac{\mu}{m}.$$

We defer the proof of Theorem 2 to Appendix A.3

4.1 Computational Inverse LCSZ

Theorem 2 provides an analytical upper bound on $t(\lambda, \mu)$ for any $\mu, \lambda \in \mathbb{N}$. However, the analytical bound does not appear to be tight for $\mu \geq 2$ (for $\mu = 1$ it is tight). This next section provides an algorithm to derive an upper bound on $t(\lambda, \mu)$ for any specific values of λ, μ. The algorithm gives tighter bounds than Theorem 2 for a table of tested values (Table 1). This is useful in practice, e.g. for deriving concrete cryptographic security parameters in cryptographic protocols that rely on LCSZ.

Algorithm 1 Greedy algorithm that returns an upper bound to Eq. (Constrained Max 2)

Input $\mu \in \mathbb{N}, \lambda \in \mathbb{N}$

1. Initialize a max heap H that stores tuples $(p, r, d) \in \mathbb{P} \times \mathbb{Z} \times \mathbb{R}$ and sorts them by the third value.

2. Initialize $w \leftarrow 0$ and $v \leftarrow 0$.

3. Push $(\mathsf{density}(2,1), 2, 1)$ onto the heap and set $\mathsf{pmax} = 2$.

4. While $w < \lambda$

5. (a) Pop (p, r, d) from H.
   (b) Push $(p, r+1, (\mathsf{density}(p, r+1))$ onto H
   (c) Set $v \leftarrow v + \mathsf{val}(p, r)$
   (d) Set $w \leftarrow w + \mathsf{weight}(p, r)$
   (e) If $p = \mathsf{pmax}$ then set $\mathsf{pmax} \leftarrow \mathsf{next\_prime}(p)$ and push $(\mathsf{density}(\mathsf{pmax}, 1), \mathsf{pmax}, 1)$ onto H

6. Output v

Theorem 3 (Computational bound). Let k be the output of algorithm Algorithm 1 on input λ, μ. Then for all $m \in \mathbb{N}$, all $N \geq 2^k$ and all μ-linear polynomials f, coprime with N, $\log_2 N \geq t(\lambda, \mu)$ and

$$P_{\mathbf{x} \leftarrow [0,m)^\mu}[f(\mathbf{x}) \equiv 0 \bmod N] \leq 2^{-\lambda} + \frac{\mu}{m}.$$

The proof of the Theorem is in Appendix A.4.

4.2 Computational Results

Using Algorithm 1 we computed analytical bounds for all $\mu \in (1, 50)$ for different values of λ. The precise bound for $\mu = 20$ and $\lambda = 120$ is:

$$2^v \geq 2^{36} \cdot 3^{20} \cdot 5^{11} \cdot 7^8 \cdot 11^5 \cdot 13^5 \cdot 17^4 \cdot 19^3 \cdot 23^3 \cdot 29^2 \cdot 31^2 \cdot 37^2 \cdot 41^2 \cdot 43^2 \cdot 47^2 \cdot 53^2 \cdot 59 \cdot 61 \cdot 67$$
$$\cdot 71 \cdot 73 \cdot 79 \cdot 83 \cdot 89 \cdot 97 \cdot 101 \cdot 103 \cdot 107 \cdot 109 \cdot 113 \cdot 127 \cdot 131 \cdot 137 \cdot 139 \cdot 149 \cdot 151 \cdot 157 \cdot 163$$

Other results are presented in their logarithmic form in Table 1. The results are significantly tighter than the analytical Theorem 2. For $n = 20$ and $\lambda = 120$ the analytical theorem gives a value for $\log_2(N)$ of 3839 vs the computational which is 416. We also provide the open-source Python implementation of the algorithm[3].

---

[3] `https://github.com/bbuenz/Composite-Schwartz-Zippel`

μ	$\lambda = 40$	$\lambda = 120$	$\lambda = 240$	μ	$\lambda = 40$	$\lambda = 120$	$\lambda = 240$
1	40	120	240	25	241	472	758
10	133	289	500	26	248	481	772
20	207	416	679	27	256	492	792
21	216	429	695	28	264	506	806
22	222	437	718	29	275	516	820
23	228	448	732	30	278	527	831
24	233	464	749	50	419	736	1105

Table 1: Computationally determined values of $t(\lambda, \mu)$ such that for all $N \geq t(\lambda, \mu)$, $\mathbb{P}_{\mathbf{x} \leftarrow [0,m)^\mu}[f(\mathbf{x}) \equiv 0 \bmod N] \leq 2^{-\lambda} + \frac{\mu}{m}$ for different μ and different λ

5 Almost-Special-Soundness

We first define deterministic (non-hiding) commitment scheme over a message space $\mathcal{M}$ and opening space $\mathcal{W} \supseteq \mathcal{M}$. When $\mathcal{M} = \mathcal{W}$ it is identical to collision-resistant hash functions. More generally, the commitment function is a collision-resistant hash function $H : \mathcal{M} \to \{0,1\}^\lambda$, but the algorithm that verifies an opening of C to m is not restricted to checking $H(m) = C$ (e.g., this may not be possible when $m \notin \mathcal{M}$).

These schemes do not provide a way to commit to $\mathbf{x} \in \mathcal{W} \setminus \mathcal{M}$, but is nonetheless useful to define in the context of arguments of knowledge. Suppose a party commits to a message $m \in \mathcal{M}$ as $C = H(m)$ and is asked to prove knowledge of an opening of C using an argument system for which the knowledge extractor is only guaranteed to extract an opening to a message in the superset $\mathcal{W}$. There are applications where it doesn't matter whether the prover knows an actual input $m \in \mathcal{M}$ to H such that $H(m) = C$ as long as it is committed in a binding way to some message in $\mathcal{W}$ that it knows. In fact, commitment schemes where $\mathcal{M} = \mathbb{Z}^n$ and $\mathcal{W} \subseteq \mathbb{Q}^n$ together with arguments of knowledge that extract openings to $\mathcal{W}$ suffice to construct linear-map vector commitments over prime fields $\mathbb{F}_p$, with polynomial commitment schemes as a special case. These have very powerful applications including the construction of generic succinct non-interactive argument (SNARK) systems for all of NP.

Definition 1 (Deterministic Commitment Scheme). A deterministic commitment scheme Γ is a tuple $\Gamma = (\mathsf{Setup}, \mathcal{G}, \mathsf{Commit}, \mathsf{Open})$ where:

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$ is a PPT algorithm that generates public parameters pp, which define a finite set of indices $\mathcal{I}$, a message space $\mathcal{M}$, and an opening space $\mathcal{W} \supseteq \mathcal{M}$;

- $\mathcal{G}(\mathsf{pp}, \iota) \to \mathsf{pp}^*$ is a PPT algorithm that generates parameters for the index $\iota \in \mathcal{I}$;

- $\mathsf{Commit}(\mathsf{pp}^*, m) \to C$ is a polynomial time computable function that takes a secret message m and returns a public commitment C.

- $\mathsf{Open}(\mathsf{pp}^*, C, w, \sigma) \to b \in \{0,1\}$ is a PPT algorithm that verifies the opening of commitment C to the message $w \in \mathcal{W}$ provided with an opening hint $\sigma \in \{0,1\}^*$.

A commitment scheme Γ is **binding** if for all PPT adversaries $\mathcal{A}$:

$$\Pr\left[b_0 = b_1 \neq 0 \wedge x_0 \neq x_1 \; : \; \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda) \\ (\iota, C, m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pp}) \\ \mathsf{pp}^* \leftarrow \mathcal{G}(\mathsf{pp}, \iota) \\ b_0 \leftarrow \mathsf{Open}(\mathsf{pp}^*, C, x_0, r_0) \\ b_1 \leftarrow \mathsf{Open}(\mathsf{pp}^*, C, x_1, r_1) \end{array} \right] \leq \mathsf{negl}(\lambda)$$

Remark on $\mathcal{I}$: The standard definition of commitment schemes does not have an indexing set $\mathcal{I}$, or equivalently has $|\mathcal{I}| = 1$. The more general definition we presented is important for our applications, and it is important for the scheme to be binding even for an adversarially sampled index. As a simple example, Setup might determine a prime-order group $\mathcal{G}$ in which the discrete logarithm is hard and $\mathcal{I} \subset \mathcal{G}$ is the set of $p-1$ generators. The commitment function at index $g \in \mathcal{I}$ computes $x \mapsto g^x$ for $x \in \mathbb{F}_p$.

Definition 2 (Almost-Special-Soundness). Let χ denote any set of size 2^λ. Let $\Gamma = (\mathsf{Setup}, \mathcal{G}, \mathsf{Commit}, \mathsf{Open})$ denote a deterministic commitment scheme with message space $\mathcal{M}$, opening space $\mathcal{W}$, and indexing set $\mathcal{I}$. A μ-round public-coin interactive proof Π for a relation $\mathcal{R}_{\mathsf{pp}}$, where the relation is possibly dependent on $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, with challenge space χ is $(k^{(\mu)}, \delta(\cdot), \Gamma, \phi)$-**almost-special-sound** if it satisfies the following conditions with respect to some pair of predicates $\phi = (\phi_a, \phi_b)$ where $\phi_a, \phi_b : [\mu] \times \mathcal{M} \to \{0,1\}$, and a negligible function $\delta : \mathbb{N} \to \mathbb{R}$:

1. The setup for the interactive proof Π includes generation of the public parameters for the commitment scheme $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, which determine sets $\mathcal{I}, \mathcal{M}$, and $\mathcal{W}$ among other parameters (including possibly the relation $\mathcal{R}_{\mathsf{pp}}$). For each index $\iota \in \mathcal{I}$ let $\mathsf{pp}_\iota = \mathcal{G}(\mathsf{pp}, \iota)$.

2. There is a localized transcript tree labeling[4] F that assigns to each node ν of a valid transcript tree for Π a commitment label $C^{(\nu)}$ for the scheme Γ at an index $\iota_\nu \in \mathcal{I}$.[5] Additionally, any leaf node ν of the transcript tree is labelled with $C^{(\nu)}$ for some index $\iota_\nu \in \mathcal{I}$ together with an opening (m_ν, o_ν) such that $\mathsf{Open}(\mathsf{pp}_{\iota_\nu}, C^{(\nu)}, m_\nu, o_\nu) = 1$ and $\phi_a(\mu, m_\mu) = 1$.

3. There is a $poly(\lambda)$ time algorithm $\mathsf{Extract}(\nu, C^{(\nu)}, \mathsf{openSubtree}) \to (m, o)$ where $C^{(\nu)}$ is a commitment label assigned by F to node ν at level ℓ_ν, and $\mathsf{openSubtree}$ contains a list of openings for the commitment labels that F assigns to all internal nodes (excluding the root) of a k-ary subtree of a transcript tree rooted at a node ν. The corresponding commitment labels and indices in $\mathcal{I}$ for each internal node of the subtree is also an implicit input. If the challenge labels on the k children of any node in the subtree are distinct, and the openings for all internal (non-root) nodes of the subtree are valid with respect to their corresponding indices in $\mathcal{I}$ and satisfy predicate ϕ_a (i.e., for any $j > \ell_\nu$ and node u on the jth level of the transcript tree that is a member of this subtree with label C^u as a commitment with respect to index ι_u, its opening (m_u, o_u) in $\mathsf{openSubtree}$ satisfies $\mathsf{Open}(\mathsf{pp}_{\iota_u}, C^{(u)}, m_u, o_u)$ and $\phi_a(j, m_u) = 1$) then the algorithm returns an opening (m, o) for the commitment label $C^{(\nu)}$ with respect to its corresponding index ι_ν, such that $\phi_b(\ell_\nu, m) = 1$ and $\mathsf{Open}(\mathsf{pp}_{\iota_\nu}, C^{(\nu)}, m_\nu, o_\nu)$ or it returns a break of the commitment $C^{(u)}$ of any label of the subtree. That is, $\mathsf{break} = (\nu, m', o', m_\nu, o_\nu, C^{(\nu)})$, such that both (m', o') and (m, o) are valid openings of $C^{(\nu)}$ to distinct messages in $\mathcal{W}$.

4. $\mathsf{Extract}(\mathsf{x}, \mathsf{openTree}) \to \mathsf{w}$ takes as inputs openings for the commitments assigned by F to all nodes in an entire transcript tree satisfying predicate ϕ_a (same condition as above for subtrees) and returns a witness w for the public input x such that $\mathcal{R}(\mathsf{x}, \mathsf{w}) = 1$ or it returns a break of a commitment label for node ν, i.e. $\mathsf{break} = (\nu, m', o', m_\nu, o_\nu, C^{(\nu)})$, such that both (m', o') and (m, o) are valid openings of $C^{(\nu)}$ to distinct messages in $\mathcal{W}$.

5. $\mathsf{Extend}(i, m, \alpha_1, .., \alpha_{\mu-i})$ is a deterministic $\mathsf{poly}(\lambda)$-time algorithm that is given an index $i \in [\mu-1]$, a message m in the opening space of the commitment scheme com, $\mu - i$ challenges from $\mathcal{X}$, and outputs $\mu - i$ messages $m'_1, ..., m'_{\mu-i}$ in the opening space of the commitment scheme.

   For any $i \in [\mu - 1]$ and m where $\phi_a(i, m) = 0$, the probability over $\alpha_i, ..., \alpha_\mu$ sampled uniformly i.i.d. from $\mathcal{X}$ that the last message $m'_{\mu-i}$ in the list returned by $\mathsf{Extend}(i, m)$ satisfies $\phi_a(\mu, m'_{\mu-i}) = 1$ is bounded by $\delta(\lambda)$.

6. $\mathsf{Break}(i, m, \alpha_1, ..., \alpha_\mu, \iota_0, ..., \iota_\mu, C_0, ..., C_\mu, (m_i, o_i), ..., (m_\mu, o_\mu))$ first runs the algorithm $\mathsf{Extend}(i, m, \alpha_i, .., \alpha_\mu)$, which returns messages $m'_1, ..., m'_{\mu-i}$. If $\phi_b(i, m_i) = 0$ or $\forall_j \ m'_j = m_{i+j-1}$ then it outputs $\perp$. Otherwise, for some $j \geq i$, it outputs an attempted opening (m', o') of C_j for index ι_j where $m' \neq m_j$.

   For any $i \in [\mu - 1]$, given a valid (accepting) transcript with commitments $\mathbf{C} = (C_0, ..., C_\mu)$ for the commitment scheme indices $\vec{\iota} = (\iota_0, ..., \iota_\mu)$, round challenges $\vec{\alpha} = (\alpha_1, ..., \alpha_\mu)$, and openings $\mathbf{open} = ((m_i, o_i), ..., (m_\mu, o_\mu))$ to the last $\mu - i + 1$ commitments, where $\phi_b(i, m_i) = 1$ and $\phi_a(j, m_j) = 1$ for all $j \in [i + 1, \mu]$, either $\mathsf{Extend}(i, m_i, \alpha_{i+1}, ..., \alpha_\mu)$ returns $m_{i+1}, ..., m_\mu$ or $\mathsf{Break}(i, m_i, \vec{\alpha}, \vec{\iota}, \mathbf{C}, \mathbf{open})$ returns the output $\mathsf{break} = (m', o', m_j, o_j, C_j)$, where (m', o') is an opening of some C_j to a conflicting message $m' \neq m_j \in \mathcal{W}$ such that $\mathsf{Open}(\mathsf{pp}_{\iota_j}, C_j, m', o') = 1$, which breaks the binding of the commitment scheme over $\mathcal{W}$.

Short-hand notation: An interactive proof is $(k^{(\mu)}, \delta)$-almost-special-sound if it is $(k^{(\mu)}, \delta, \mathsf{com}, \phi)$-almost-special-sound for some commitment scheme com and some predicate pair ϕ. We may omit δ and simply write $k^{(\mu)}$-almost-special-soundness if this holds for some negligible function $\delta : \mathbb{N} \to \mathbb{R}$.

---

[4] Recall that a localized transcript tree labeling is a function F that given the transcript tree L as input, and thus its assignments of message/challenge labels to each tree node, induces a post-order tree labeling $F(L) : [1, N] \to \{0, 1\}^*$, which computes a new label on each node ν that is a function of its predecessor labels $L(\mathsf{pred}(\nu))$. As a special case F may be the identity function, i.e. the induced labeling is the same as L.

[5] Crucially, the definition does not require the ith position of all transcripts to use the same commitment index. The commitment index μ_i used in a particular transcript for the ith commitment $C^{(i)}$ might be a function of the transcript prefix preceding $C^{(i)}$.

Remark 6. Any special sound protocol satisfies almost-special-soundness as 3) essentially captures the special soundness definition. More precisely a $k^{(\mu)}$-special sound satisfies $k^{(\mu)}$-almost-special-soundness by setting the commitment scheme to be trivial (i.e., identity function) and the ith round commitment $C^{(i)}$ to the prover's ith round message and setting the predicates $\phi_a = 1, \phi_b = 0$ to be trivial as well (i.e., always return 1 and 0 respectively). The algorithm Extend can output an arbitrary set of messages because the condition on the algorithm is vacuously true as $\phi_a(i, m) \neq 0$ for any (i, m). The algorithm $\mathsf{Extract}(\nu, C^{(\nu)}, *)$ is trivial because $C^{(\nu)}$ is the message itself. The algorithm Break is also trivial as ϕ_b is always 0. The algorithm $\mathsf{Extract}(x, \mathsf{openTree}) \to w$ exists by the definition of $k^{(\mu)}$-special soundness.

Theorem 4 (AMSS implies Knowledge Soundness). If a μ-round interactive proof for a relation $\mathcal{R}$ with λ-bit challenges, $\mu \in O(\log(\lambda + |\mathsf{x}|))$, and verifier decision algorithm runtime $t_V \in \mathsf{poly}(|\mathsf{pp}|, |\mathsf{x}|, \lambda)$ on input $\mathsf{x} \in \mathcal{L}_{\mathcal{R}}$ and parameters $\mathsf{pp} \leftarrow \Gamma.\mathsf{Setup}(1^\lambda)$ is $(k^{(\mu)}, \delta, \Gamma, \phi)$-almost-special-sound then for $\delta'(\lambda) = 2\lambda(k+1)^\mu(\mu + t_V) \cdot \max(\delta(\lambda), k \cdot 2^{-\lambda}) + 2^{-\lambda}$ it is δ'-knowledge sound for the modified relation:

$$\mathcal{R}'(\mathsf{pp}) = \{(\mathsf{x}, \mathsf{w}) : \mathcal{R}(\mathsf{x}, \mathsf{w}) = 1 \ \vee \mathsf{w} \in \mathcal{L}_{\mathsf{break}}(\mathsf{pp})\}$$

where

$$\mathcal{L}_{\mathsf{break}}(\mathsf{pp}) = \{(\mathcal{C}, \sigma_1, \sigma_2, \iota) : \iota \in \mathcal{I} \ \wedge \ \sigma_1 \neq \sigma_2 \ \wedge \ \mathsf{Open}(\mathsf{pp}_\iota, \mathcal{C}, \sigma_1) = \mathsf{Open}(\mathsf{pp}_\iota, \mathcal{C}, \sigma_2) = 1\}$$

Remark 7. $\delta'(\lambda)$ is a negligible function if $\delta(\lambda)$ is negligible, assuming $k \in O(1)$, $\mu \in O(\log(\lambda + |\mathsf{x}|))$, $t_V \in \mathsf{poly}(|\mathsf{x}|, \lambda)$, and $|\mathsf{x}| \in \mathsf{poly}(\lambda)$.

The proof of Theorem 4 is in Appendix C. By Lemma 11, this theorem has the following corollary:

Corollary 2. An interactive proof with λ-bit challenges that is $k^{(\mu)}$-almost-special-sound for a relation $\mathcal{R}$ and has at most $\mu \in O(\log(\lambda + |\mathsf{x}|))$ rounds on any instance $\mathsf{x} \in \mathcal{L}_{\mathcal{R}}$ has witness-extended emulation for $\mathcal{R}$.

6 Argument of knowledge of "short" rational opening

For any deterministic homomorphic commitment to vectors in $\mathbb{Z}_p^n$, where the commitment function is a homomorphism $\mathsf{com} : \mathbb{Z}_p^n \to \mathbb{G}$, there is a generic succinct argument of knowledge of a commitment opening (i.e., a homomorphism preimage) with $O(\log n)$ communication complexity [BDFG21; AC20; BCS21]. This same protocol can also be used to argue knowledge of any linear form of the committed vector. This is also known as a linear-map vector commitment (LMVC) [LM19; CNRZZ22], and captures polynomial commitments as a special case.

We generalize this result to work with deterministic homomorphic integer vector commitments that are only binding over *short* vectors in $\mathbb{Z}^n$, which includes vector commitment schemes based on lattices and groups of unknown order. Furthermore, the protocol can still be used to argue knowledge of any linear form $h : \mathbb{Z}^n \to \mathbb{G}$ of the committed vector, where $\mathbb{G}$ is a prime order group. The protocol is essentially the same as the succinct homomorphism preimage protocol [BDFG21; AC20], but where the verifier additionally checks a bound on the prover's final message. This can also be viewed as a special case of the sumcheck argument with a bound check, as described in [BCS21], but we provide a much tighter soundness analysis.

A bit more precisely, the protocol is not an argument of knowledge of a short integer vector pre-image per se, but rather a *short rational* opening of the commitment $C = \mathsf{com}(\mathsf{x})$, which is some $\mathsf{x}' \in \mathbb{Z}^n$ and $z \in \mathbb{Z}$ such that $\mathsf{com}(\mathsf{x}') = z \cdot C$ and $\mathsf{x}'/z \in \mathcal{M} \subseteq \mathbb{Q}^n$ where $\mathcal{M} = \{\mathbf{x}/z \in \mathbb{Q}^n : z \in \mathbb{Z}, gcd(\mathbf{x}, z) = 1, ||\mathbf{x}|| \leq \beta_\mathsf{n}, |z| \leq \beta_\mathsf{d}\}$ is a subset of rational vectors with bounded norm denominators and numerators in reduced form. We call this an opening of the commitment C to the rational vector $\mathbf{x}'/z \in \mathcal{M}$. If the commitment scheme is binding over such rational openings to vectors in $\mathcal{M}$ and $[0, p)^n \subseteq \mathcal{M}$, then it also functions as a binding commitment scheme for vectors in $\mathbb{Z}_p^n$ where an opening to $m \in \mathcal{M}$ is also an opening to $m \bmod p \in \mathbb{Z}_p^n$ and the protocol is thus an argument of knowledge of an opening to a unique vector in $\mathbb{Z}_p^n$. Finally, for any linear form $h : \mathbb{Z}_p^n \to \mathbb{G}$ the modified commitment $\mathsf{com}^*(\mathbf{x}) = (\mathsf{com}(\mathbf{x}), h(\mathbf{x}))$ preserves binding over $\mathcal{M}$ and running the same protocol for com^* becomes an argument of knowledge for an opening of the commitment (C, y) to a unique $\mathbf{x}^* \in \mathbb{Z}_p^n$ such that $h(\mathbf{x}^*) = y$.

Remark on zero-knowledge Deterministic commitment schemes are non-hiding and our construction is focussed on producing a succinct argument of knowledge without concern for zero-knowledge. To obtain a zero-knowledge argument, the succinct argument can be composed with (i.e., applied as the pivot to) a sigma protocol for pre-images of ϕ. See [ACK21b] for further details.

Definition 3 ($\mathbb{Z}$-linear commitment scheme). Let $\Gamma = (\mathsf{Setup}, \mathcal{G}, \mathsf{Commit}, \mathsf{Open})$ be a deterministic commitment scheme such that for any $\mathsf{pp} = (\mathcal{I}, \mathcal{M}, \mathcal{W}) \leftarrow \mathsf{Setup}(1^\lambda)$ where $\mathcal{M} \subseteq \mathcal{W} \subseteq \mathbb{Z}^n$ and $\iota \in \mathcal{I}$ the commitment function $\mathsf{Commit}(\mathsf{pp}_\iota, \cdot) : \mathcal{M} \to \mathbb{G}$ is the restriction $h|_{\mathcal{M}}$ of a group homomorphism $h : \mathbb{Z}^n \to \mathbb{G}$. We say that Γ is a $\mathbb{Z}$-linear deterministic commitment scheme.

Definition 4 (Rational openings). Let $\Gamma = (\mathsf{Setup}, \mathcal{G}, \mathsf{Commit}, \mathsf{Open})$ be $\mathbb{Z}$-linear deterministic commitment scheme where commitments are contained within a finite group $\mathbb{G}$. Let $\mathcal{W}^* \subseteq \mathbb{Q}^n$ and let $\mathsf{Setup}^*(1^\lambda)$ denote a new setup algorithm that runs $\mathsf{Setup}(1^\lambda)$ but replaces $\mathcal{W}$ with $\mathcal{W}^*$. Let $\mathsf{Open}^*(\mathsf{pp}, C, \mathbf{x}/z, (z, \mathbf{x}, \sigma))$ denote a new opening verification algorithm which on inputs $z \in \mathbb{Z}$, $\mathbf{x} \in \mathbb{Z}^n$ such that $\mathbf{x}/z \in \mathcal{W}^*$ and $C \in \mathbb{G}$ returns 1 if and only if $z \neq 0 \bmod |\mathbb{G}|$, and $\mathsf{Open}(\mathsf{pp}, z \cdot C, \mathbf{x}, \sigma) = 1$.[6] We call the opening hint $(z, \mathbf{x}, \sigma)$ a *rational opening* of C to $\mathbf{x}/z \in \mathcal{W}^*$. We say Γ is *binding over rational openings to* $\mathcal{W}^* \subseteq \mathbb{Q}^n$ if $\Gamma^* = (\mathsf{Setup}^*, \mathcal{G}, \mathsf{Commit}, \mathsf{Open}^*)$ satisfies the usual definition of commitment binding (See Definition 1).

Lemma 1 (trivial). If $(\mathsf{Setup}, \mathcal{G}, \mathsf{Commit}, \mathsf{Open})$ is a $\mathbb{Z}$-linear deterministic commitment scheme binding over $\mathcal{W} \subseteq \mathbb{Q}^n$ where $\mathsf{Commit}(\mathsf{pp}_\iota, \cdot) : \mathbb{Z}^n \to \mathbb{G}_{\mathsf{com}}$ and $h : \mathbb{Z}^n \to \mathbb{G}$ is any homomorphism, then the modified scheme $(\mathsf{Setup}, \mathcal{G}, \mathsf{Commit}, \mathsf{Open}^*)$ where:

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$ and $\mathcal{G}(\iota, \mathsf{pp}) \to \mathsf{pp}_\iota$
- $\mathsf{Commit}^*(\mathsf{pp}_\iota, \mathbf{x} \in \mathbb{Z}^n) \to (\mathsf{Commit}(\mathsf{pp}_\iota, \mathbf{x}), h(\mathbf{x})) \in \mathbb{G}_{\mathsf{com}} \times \mathbb{G}$
- $\mathsf{Open}^*(\mathsf{pp}_\iota, (C, y) \in \mathbb{G}_{\mathsf{com}} \times \mathbb{G}, \mathbf{x} \in \mathbb{Z}^n, s) \to \mathsf{Open}(\mathsf{pp}_\iota, \mathbf{x}, s, C)$

is also a $\mathbb{Z}$-linear deterministic commitment scheme binding over $\mathcal{W}$.

Proof. Any break of the modified scheme includes a break of the original scheme, i.e. tuples $(\mathbf{x}, \iota, s, (C, y))$ and $(\mathbf{x}', \iota, s', (C, y))$ such that $\mathbf{x} \neq \mathbf{x}' \in \mathcal{W}$ and $\mathsf{Open}(\mathsf{pp}_\iota, C, \mathbf{x}, s) = 1$ and $\mathsf{Open}(\mathsf{pp}_\iota, C, \mathbf{x}', s') = 1$. □

Commitment schemes binding over bounded rationals Let $\beta = \beta_\mathsf{n} \cdot \beta_\mathsf{d}$ for some $\beta_\mathsf{n}, \beta_\mathsf{d} > 0$. Let $\mathcal{W}(\beta_\mathsf{n}, \beta_\mathsf{d}) := \{\mathbf{x}/z \in \mathbb{Q}^n : z \in \mathbb{Z}, ||\mathbf{x}|| \leq \beta_\mathsf{n}, |z| \leq \beta_\mathsf{d}\}$. The following lemmas establish several families of integer vector commitment schemes that are binding over $\mathcal{W}(\beta_\mathsf{n}, \beta_\mathsf{d})$.

Lemma 2. Let $\Gamma = (\mathsf{Setup}, \mathcal{G}, \mathsf{Commit}, \mathsf{Open})$ denote any $\mathbb{Z}$-linear deterministic commitment scheme where for any $\iota \in \mathcal{I}$ and $\mathsf{pp}_\iota = \mathcal{G}(\mathsf{pp}, \iota)$ the algorithm $\mathsf{Open}(\mathsf{pp}_\iota, C, \mathbf{x})$ receives no hint and simply checks $C = \mathsf{Commit}(\mathsf{pp}_\iota, \mathbf{x}) = \langle \mathbf{x}, \mathbf{g}_{\mathsf{pp}_\iota} \rangle$ where $\mathbf{g}_{\mathsf{pp}_\iota} \in \mathbb{G}^n$ is a basis for the homomorphism. Let $\mathcal{A}(\mathsf{pp})$ denote a polynomial time algorithm that on input $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ returns $\iota \in \mathcal{I}$, $(z, \mathbf{x}), (z', \mathbf{x}') \in \mathbb{Z} \times \mathbb{Z}^n$ and $Y \in \mathbb{G}$ such that with probability non-negligible in λ:

(a) $z'\mathbf{x} \neq z\mathbf{x}'$
(b) $\frac{\mathbf{x}}{z}, \frac{\mathbf{x}'}{z'} \in \mathcal{W}(\beta_\mathsf{n}, \beta_\mathsf{d})$
(c) $z \cdot Y = \mathsf{Commit}(\mathsf{pp}_\iota, \mathbf{x})$ and $z' \cdot Y = \mathsf{Commit}(\mathsf{pp}_\iota, \mathbf{x}')$
(d) $z, z' \neq 0 \bmod |\mathbb{G}|$ and either $\frac{\mathbf{x}}{z} \neq 0 \bmod |\mathbb{G}|$ or $\frac{\mathbf{x}'}{z'} \neq 0 \bmod |\mathbb{G}|$

If $|\mathbb{G}|$ is prime then $\mathcal{A}$ breaks the binding of Γ over $\mathcal{W}_\beta = \{\mathbf{x} \in \mathbb{Z}^n : ||\mathbf{x}|| \leq \beta\}$. Otherwise, $\mathcal{A}$ either breaks the binding over $\mathcal{W}_\beta$ or outputs an element u in the span of $\mathbf{g}_{\mathsf{pp}_\iota}$ and an integer multiple of its order.

Proof. Let $\mathsf{com} := \mathsf{Commit}(\mathsf{pp}_\iota, \cdot)$ for shorthand notation. Condition (c) implies $\mathsf{com}(z' \cdot \mathbf{x}) = \mathsf{com}(z \cdot \mathbf{x}')$. Condition (b) implies $\frac{\mathbf{x}}{z} = \frac{\mathbf{x}_1}{z_1}$ for $||\mathbf{x}_1|| \leq \beta_\mathsf{n}$ and $|z_1| \leq \beta_\mathsf{d}$ where $gcd(\mathbf{x}_1, z_1) = 1$, and likewise $\frac{\mathbf{x}'}{z'} = \frac{\mathbf{x}_2}{z_2}$ for $||\mathbf{x}_2|| \leq \beta_\mathsf{n}$ and $|z_2| \leq \beta_\mathsf{d}$ where $gcd(\mathbf{x}_2, z_2) = 1$. Let $\Delta_1 = z/z_1 \in \mathbb{Z}$ and $\Delta_2 = z/z_2 \in \mathbb{Z}$ and let $\Delta = \Delta_1 \cdot \Delta_2$. These relations imply that $\Delta \cdot \mathsf{com}(z_2 \cdot \mathbf{x}_1) = \Delta \cdot \mathsf{com}(z_1 \cdot \mathbf{x}_2)$ where, by condition (b), $||z_1 \cdot \mathbf{x}_2|| \leq \beta$ and

---

[6]In a group of unknown order it may be difficult to check that $z \neq 0 \bmod |\mathbb{G}|$. If g is a generator (e.g., the generator of a subgroup of unknown order) then it suffices to check $z \cdot g = 0$. In any case, GUOs are typically used for commitments under the hardness assumption that it is difficult to compute any multiple of the order of $\mathbb{G}$, in which case checking $z \neq 0 \bmod |\mathbb{G}|$ can be dropped from verification.

$||z_2 \cdot \mathbf{x}_1|| \leq \beta$. Finally, $\Delta \neq 0 \bmod |\mathbb{G}|$ by condition (d) and $z_2 \cdot \mathbf{x}_1 \neq z_1 \cdot \mathbf{x}_2$ by condition (a). Let $\mathbf{x}^* = z_2 \cdot \mathbf{x}_1 - z_1 \cdot \mathbf{x}_2$ and $u = \mathsf{com}(\mathbf{x}^*)$. We have established that $\mathbf{x}^* \neq 0$ and $\Delta \cdot u = 0$ and $\Delta \neq 0 \bmod |\mathbb{G}|$. If $|\mathbb{G}|$ is prime, then this implies $\mathsf{com}(z_2 \cdot \mathbf{x}_1) = \mathsf{com}(z_1 \cdot \mathbf{x}_2)$, which breaks the binding of Π_{cm} over $\mathcal{W}_\beta$. If $|\mathbb{G}|$ is not prime, then either $u = 0$, which breaks binding over $\mathcal{W}_\beta$, or otherwise Δ must be a multiple of the order of u, which is in the span of $\mathbf{g}_{\mathsf{pp}_\iota}$. $\qquad\square$

Corollary 3 (Homomorphic prime-order commitments)**.** Let $\Gamma = (\mathsf{Setup}, \mathcal{G}, \mathsf{Commit}, \mathsf{Open})$ denote any $\mathbb{Z}$-linear deterministic commitment scheme where the commitments are contained within a group $\mathbb{G}$ of prime-order, for $C \neq 0$ the algorithm $\mathsf{Open}(\mathsf{pp}_\iota, C, \mathbf{x})$ checks $\mathsf{Commit}(\mathsf{pp}_\iota, \mathbf{x}) = C$, and the element $0 \in \mathbb{G}$ is the unique valid commitment to $\mathbf{0} \in \mathbb{Z}^n$. If Γ is binding over $\mathcal{W}_\beta = \{\mathbf{x} \in \mathbb{Z}^n : ||\mathbf{x}|| \leq \beta\}$ then Γ is also binding over rational openings to $\mathcal{W}(\beta_{\mathsf{n}}, \beta_{\mathsf{d}})$.

Proof. Suppose there exists a polynomial time algorithm $\mathcal{A}(\mathsf{pp})$ that on input $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ returns $\iota \in \mathcal{I}$ and $C \in \mathbb{G}$ and conflicting valid openings $(z, \mathbf{x}), (z', \mathbf{x}') \in \mathbb{Z} \times \mathbb{Z}^n$ of C to $\mathbf{x}/z \neq \mathbf{x}'/z' \in \mathcal{W}(\beta_{\mathsf{n}}, \beta_{\mathsf{d}})$ with probability non-negligible in λ. By validity of the openings $z, z' \neq 0 \bmod |\mathbb{G}|$. If $\mathbf{x} = 0 \bmod |\mathbb{G}|$ then $z \cdot C = \mathsf{Commit}(\mathsf{pp}_\iota, \mathbf{x}) = 0$, which implies $C = 0$ because $z \neq 0 \bmod |\mathbb{G}|$ and $\mathbb{G}$ has prime order. If $C = 0$ then only $\mathbf{x} = \mathbf{x}' = 0$ would be accepted as valid openings of $z \cdot C$ and $z' \cdot C$ respectively, contradicting $\mathbf{x}/z \neq \mathbf{x}'/z'$. This shows that $\mathbf{x} \neq 0 \bmod |\mathbb{G}|$ and the same argument shows that $\mathbf{x}' \neq 0 \bmod |\mathbb{G}|$. Therefore, by Lemma 2, $\mathcal{A}(\mathsf{pp})$ breaks the binding of Π over $\mathcal{W}_\beta = \{\mathbf{x} \in \mathbb{Z}^n : ||\mathbf{x}|| \leq \beta\}$, a contradiction. $\qquad\square$

Homomorphic prime-order examples. Examples of commitments to integer vectors that satisfy the conditions in Corollary 3 trivially include Pedersen commitments but also Ajtai commitments based on Integer-SIS/Ring-SIS/Module-SIS. Let q be a prime, let $n, \kappa \in \mathbb{N}$ denote security parameters, Let $\Phi_n(x)$ be the n-th cyclotomic polynomial[7] (i.e., its roots are the primitive n-th roots of unity). Let $\mathcal{R} = \mathbb{Z}[x]/\Phi_n$ and $\mathcal{R}_q = \mathbb{Z}_q[x]/\Phi_n$, and let $m = m(n)$ and $B = B(n)$ be functions.

- **Integer-SIS.** The problem $\mathrm{SIS}_{n,q,m,B}$ is: given $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, find $\mathbf{z} \in \mathbb{Z}^m$ such that $\mathbf{A} \cdot \mathbf{z} = 0^n \bmod q$ and $0 < ||\mathbf{z}|| \leq B$.

- **Ring-SIS.** The problem $\mathrm{R\text{-}SIS}_{n,q,m,B}$ is: given $\mathbf{a} \xleftarrow{\$} \mathcal{R}_q^m$, find $\mathbf{z} \in \mathcal{R}^m$ such that $\mathbf{a} \cdot \mathbf{z} = 0 \bmod q$ and $0 < ||\mathbf{z}|| \leq B$.

- **Module-SIS.** For some $\kappa > 0$, the problem $\mathrm{M\text{-}SIS}_{\kappa,n,q,m,B}$ is: given $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{\kappa \times m}$, find $\mathbf{z} \in \mathcal{R}^m$ such that $\mathbf{A} \cdot \mathbf{z} = 0^\kappa \bmod q$ and $0 < ||\mathbf{z}|| \leq B$.

Note that SIS and R-SIS are both special cases of M-SIS, with appropriate setting of the parameters. For example, Ring-SIS is the same as Module-SIS when $\kappa = 1$. All three problems are related to certain worst case lattice problems, and are believed to be hard for appropriate settings of the parameters n, q, m, B, and κ. In particular, these problems are believed to be post-quantum secure when the following conditions holds:

- **Integer-SIS [MP13; GPV08]** n sufficiently large and $m, \log(q) \leq poly(n)$ and $q \geq B\sqrt{n} \cdot \omega(\log n)$.

- **Module-SIS [LS15, Th. 3.6]:** $n\kappa$ sufficiently large and $q > B \cdot (n\kappa)^{1/2} \omega(\sqrt{\log n\kappa})$, and both m and $\log q$ are less than $\mathrm{poly}(n\kappa)$.

Corollary 4 (Homomorphic GUO commitments)**.** Let $\Gamma = (\mathsf{Setup}, \mathcal{G}, \mathsf{Commit}, \mathsf{Open})$ denote any $\mathbb{Z}$-linear deterministic commitment scheme where for any pp, ι, and $\mathsf{pp}_\iota = \mathcal{G}(\mathsf{pp}, \iota)$ the function $\mathsf{Commit}(\mathsf{pp}_\iota) : \mathbb{Z}^n \to \mathbb{G}$ is a homomorphism with basis $\mathbf{g}_{\mathsf{pp}_\iota}$. Suppose that the subgroup order assumption for $\mathbb{G}$ holds[8] with respect to the distribution over subgroups $\langle \mathbf{g}_{\mathsf{pp}_\iota} \rangle \subseteq \mathbb{G}$ induced by sampling $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ and any adversarially sampled $\iota \in \mathcal{I}$. If Γ is binding over $\mathcal{W}_\beta = \{\mathbf{x} \in \mathbb{Z}^n : ||\mathbf{x}|| \leq \beta\}$ then Γ is also binding over rational openings to $\mathcal{W}(\beta_{\mathsf{n}}, \beta_{\mathsf{d}})$.

---

[7]When n is a power of two we have that $\Phi_n(x) = x^n + 1$.

[8]If the generators $\mathbf{g}_{\mathsf{pp}_\iota}$ are uniformly distributed for any ι and $\mathbb{G}$ is cyclic (or more generally a random element in the span of $\mathbf{g}_{\mathsf{pp}_\iota}$ is statistically close to uniform over $\mathbb{G}$) then this is equivalent to the assumption that it is hard to compute a multiple of the order of a random element, which is implied by the difficulty of taking square roots of random elements (the RSA assumption) for $\mathbb{G}$. In certain groups where it is difficult to compute any integer that shares a common factor with $|\mathbb{G}|$ then the assumption holds regardless of how $\mathbf{g}_{\mathsf{pp}_\iota}$ is sampled. One such group is the multiplicative subgroup $\mathbb{H} = \{x^4 : x \in \mathbb{Z}_N^\times\}$ for $N = p \cdot q$, where p and q are unknown safe primes and thus $|\mathbb{H}| = (p-1)(q-1)/4 = p' \cdot q'$ for unknown primes p', q'.

Corollary 5 (*DARK* commitments). The integer vector commitment scheme known as DARK [BFS20] uses a group of unknown order $\mathbb{G}$ and for $\mathbf{x} \in \mathbb{Z}^n$ computes the homomorphism $\mathbf{x} \mapsto g^{f_\mathbf{x}(q)}$, where $f_\mathbf{x}(q) = \sum_{i=1}^n \mathbf{x}_i q^{i-1}$, given setup parameters $q \in \mathbb{N}$ and $g \in \mathbb{G}$ sampled uniformly at random. Setting $q = \lceil 2\sqrt{n}\beta \rceil$, DARK is binding over rational openings to $\mathcal{W}(\beta_\mathsf{n}, \beta_\mathsf{d})$ based on the order assumption in $\mathbb{G}$.

Proof. For $\mathbf{x} \in \mathbb{Z}^n$ let $f_\mathbf{x}$ denote the univariate polynomial with coefficient vector $\mathbf{x}$. By Fact 1, $f_\mathbf{x}(q) \neq f_{\mathbf{x}'}(q)$ for all $\mathbf{x} \neq \mathbf{x}'$ of L_∞ norm at most $q/2$ which is guaranteed if the L_2 norm is at most $q/(2\sqrt{n})$. By the order assumption (Assumption 1) no polynomial-time adversary can compute integers $x \neq y$ such that $g^x = g^y$ for $g \xleftarrow{\$} \mathbb{G}$. DARK is therefore binding over $\mathcal{W}_\beta = \{\mathbf{x} \in \mathbb{Z}^n : ||\mathbf{x}|| \leq \beta\}$ for $q \geq 2\sqrt{n}\beta$ by the order assumption, and as a corollary to Lemma 2 it is also binding over $\mathcal{W}(\beta_\mathsf{n}, \beta_\mathsf{d})$ under the order assumption. $\qquad\square$

Definition 5. We define a *integer matrix norm family* $\mathcal{F}$ as an infinite collection $\mathcal{F} = \{\mathcal{N}_{m,n}\}$ of matrix norms $\mathcal{N}_{m,n} : \mathbb{Z}^{m \times n} \to \mathbb{R}$. We say that the $\mathcal{F}$ is $\alpha(m, n, k)$-submultiplicative if for all $m, n, k \in \mathbb{N}$, $M \in \mathbb{Z}^{m \times n}$, and $X \in \mathbb{Z}^{n \times k}$:

$$\mathcal{N}_{m,k}(M \cdot X) \leq \alpha(m, n, k) \cdot \mathcal{N}_{m,n}(M) \cdot \mathcal{N}_{n,k}(X)$$

Lemma 3. Let $(\mathsf{Setup}, \mathcal{G}, \mathsf{Commit}, \mathsf{Open})$ be any $\mathbb{Z}$-linear deterministic commitment scheme that is binding over $\mathcal{W}_\beta = \{\mathbf{x} \in \mathbb{Z}^n : ||\mathbf{x}|| \leq \beta\}$, where the norm $|| \cdot ||$ belongs to an α-submultiplicative norm family. Let $U : \mathbb{Z}^m \to \mathbb{Z}^n$ denote any injective linear map, i.e. $U \in \mathbb{Z}^{m \times n}$ is a full row rank matrix. Then the modified scheme $(\mathsf{Setup}, \mathcal{G}, \mathsf{Commit}^*, \mathsf{Open}^*)$ over $\mathbb{Z}^m$ where:

- $\mathsf{Commit}^*(\mathsf{pp}, \mathbf{x} \in \mathbb{Z}^m) \to \mathsf{Commit}(\mathsf{pp}, \mathbf{x} \cdot U)$
- $\mathsf{Open}^*(\mathsf{pp}, C, \mathbf{x} \in \mathbb{Z}^m, s) \to \mathsf{Open}(\mathsf{pp}, C, \mathbf{x} \cdot U, s)$

is binding over $\mathcal{W}^* = \{\mathbf{x} \in \mathbb{Z}^m : ||\mathbf{x}|| \leq \frac{\beta}{\alpha \cdot ||U||}\}$.

Proof. Suppose $(C, \mathbf{x} \in \mathbb{Z}^m, s)$ and $(C, \mathbf{x}' \in \mathbb{Z}^m, s')$ are both accepted by Open^* for distinct $\mathbf{x}, \mathbf{x}' \in \mathcal{W}^*$, then both $(C, \mathbf{x} \cdot U, s)$ and $(\mathbf{x}' \cdot U, s', C)$ are accepted by Open. Since U is injective, $\mathbf{x} \cdot U \neq \mathbf{x}' \cdot U$. Furthermore, since the norm is α-submultiplicative, $||\mathbf{x} \cdot U|| \leq \alpha \cdot ||\mathbf{x}|| \cdot ||U|| \leq \beta$. Similarly, $||\mathbf{x}' \cdot U|| \leq \beta$. This contradicts the binding of $(\mathsf{Setup}, \mathcal{G}, \mathsf{Commit}, \mathsf{Open})$. $\qquad\square$

Lemma 4. Let $(\mathsf{Setup}, \mathcal{G}, \mathsf{Commit}, \mathsf{Open})$ be any $\mathbb{Z}$-linear deterministic commitment scheme that is binding over rational openings to $\mathcal{W}(\beta_\mathsf{n}, \beta_\mathsf{d})$, where the norm $|| \cdot ||$ belongs to an α-submultiplicative norm family. Let $U : \mathbb{Z}^m \to \mathbb{Z}^n$ denote any injective linear map, i.e. $U \in \mathbb{Z}^{m \times n}$ is a full row rank matrix. Then the modified scheme $(\mathsf{Setup}, \mathcal{G}, \mathsf{Commit}^*, \mathsf{Open}^*)$ over $\mathbb{Z}^m$ where:

- $\mathsf{Commit}^*(\mathsf{pp}, \mathbf{x} \in \mathbb{Z}^m) \to \mathsf{Commit}(\mathsf{pp}, \mathbf{x} \cdot U)$
- $\mathsf{Open}^*(\mathsf{pp}, C, \mathbf{x} \in \mathbb{Z}^m, s) \to \mathsf{Open}(\mathsf{pp}, C, \mathbf{x} \cdot U, s)$

is binding over $\mathcal{W}(\frac{\beta_\mathsf{n}}{\alpha ||U||}, \beta_\mathsf{d})$.

Proof. Let $\beta_\mathsf{n}^* = \frac{\beta_\mathsf{n}}{\alpha ||U||}$. Suppose $(C, \mathbf{x} \in \mathbb{Z}^m, z \in \mathbb{Z}, s)$ and $(C, \mathbf{x}' \in \mathbb{Z}^m, z' \in \mathbb{Z}, s')$ are both accepted by Open^* for distinct $\mathbf{x}/z, \mathbf{x}'/z \in \mathcal{W}(\beta_\mathsf{n}^*, \beta_\mathsf{d})$, then both $(C, \mathbf{x} \cdot U, z, s)$ and $(\mathbf{x}' \cdot U, z', s', C)$ are accepted by Open. Let $\mathbf{x}/z = \mathbf{x}_1/z_1$ in reduced form so that $||\mathbf{x}_1|| \leq \beta_\mathsf{n}^*$ and $|z_1| \leq \beta_\mathsf{d}$. Similarly, let $\mathbf{x}'/z' = \mathbf{x}_1'/z_1'$ in reduced form so that $||\mathbf{x}_1'|| \leq \beta_\mathsf{n}^*$ and $|z_1'| \leq \beta_\mathsf{d}$. Since the norm is α-submultiplicative, $||\mathbf{x}_1 \cdot U|| \leq \alpha \cdot ||\mathbf{x}_1|| \cdot ||U|| \leq \beta_\mathsf{n}$. Similarly, $||\mathbf{x}_1' \cdot U|| \leq \beta_\mathsf{n}$. This shows that $\mathbf{x} \cdot U/z \in \mathcal{W}(\beta_\mathsf{n}, \beta_\mathsf{d})$ and $\mathbf{x}' \cdot U/z' \in \mathcal{W}(\beta_\mathsf{n}, \beta_\mathsf{d})$. Furthermore, since $z'\mathbf{x} \neq z\mathbf{x}'$ and U is injective, $z'\mathbf{x} \cdot U \neq z\mathbf{x}' \cdot U$. This contradicts the binding of $(\mathsf{Setup}, \mathcal{G}, \mathsf{Commit}, \mathsf{Open})$. $\qquad\square$
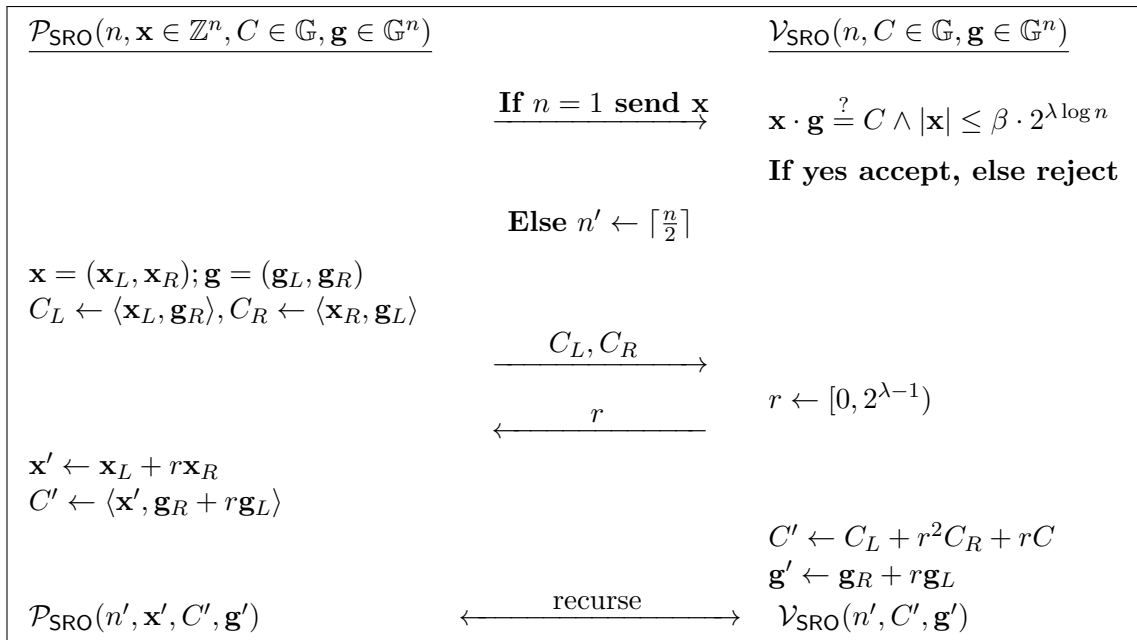
6.1 Interactive protocol for short rational openings

The public input of the protocol is a homomorphism $\mathsf{com} : \mathbb{Z}^n \to \mathbb{G}$, a target value $C \in \mathbb{G}$, and a bound $\beta \in \mathbb{R}$. The honest prover has a witness $\mathbf{x}$ of L2 norm at most β such that $\mathsf{com}(\mathbf{x}) = C$. The prover's claim is that it knows some witness $(z, \mathbf{x})$ such that $\mathsf{com}(\mathbf{x}) = z \cdot C$ and $||\mathbf{x}|| \leq 2^{\lambda \log n}\sqrt{\xi} \cdot \beta$ and $|z| \leq \sqrt{\xi}/2^{\lambda \log n}$ for a *soundness slack* parameter ξ. In other words, setting $\beta_\mathsf{n} = 2^{\lambda \log n}\sqrt{\xi} \cdot \beta$ and $\beta_\mathsf{d} = \sqrt{\xi}/2^{\lambda \log n}$ so that $\beta_\mathsf{n} \cdot \beta_\mathsf{d} = \xi \cdot \beta$, the knowledge extractor for this protocol opens a valid rational opening for C as long as the commitment scheme is binding over $\mathcal{W}(\beta_\mathsf{n}, \beta_\mathsf{d}) := \{\mathbf{x}/z \in \mathbb{Q}^n : z \in \mathbb{Z}, ||\mathbf{x}||_2 \leq \beta_\mathsf{n}, |z| \leq \beta_\mathsf{d}\}$. We prove that it suffices to set $\xi = 2^{(16+2\mu)\lambda + 6\mathsf{CSZ}_{\mu,\lambda} + 2\mu + 6}$,

where $\mathsf{CSZ}_{\log n, \lambda} \in O(\log^2 n + \lambda \log \log n)$ is derived from the Multilinear Composite Schwartz Zippel (Theorem 2). Hence $\xi \in 2^{O(\lambda \log n + \log^2 n)}$.

Let $\mathbf{g} = (\mathbf{g}_1, ..., \mathbf{g}_n)$ denote a basis of com such that $\mathsf{com}(\mathbf{x}) = \langle \mathbf{x}, \mathbf{g} \rangle$. The interactive protocol is essentially the succinct argument for homomorphism preimages from [BDFG21; AC20], but where the verifier additionally checks a bound on the prover's final message. This is also a special case of a sumcheck argument with a bound check, as described in [BCS21]. The prover's final message is an integer $x^* \in \mathbb{Z}$ and the verifier immediately rejects the proof unless $|x|^* \leq \beta \cdot 2^{\lambda \log n}$. We include the full protocol in the figure below for completeness.

Figure 1: *A succinct interactive protocol for short rational openings.* The honest prover's input must have norm at most β. For simplicity n is a power of 2.

$\underline{\mathcal{P}_{\mathsf{SRO}}(n, \mathbf{x} \in \mathbb{Z}^n, C \in \mathbb{G}, \mathbf{g} \in \mathbb{G}^n)}$		$\underline{\mathcal{V}_{\mathsf{SRO}}(n, C \in \mathbb{G}, \mathbf{g} \in \mathbb{G}^n)}$
	If $n = 1$ send $\underline{\mathbf{x}}$ $\xrightarrow{\hspace{2cm}}$	$\mathbf{x} \cdot \mathbf{g} \stackrel{?}{=} C \wedge |\mathbf{x}| \leq \beta \cdot 2^{\lambda \log n}$ **If yes accept, else reject**
	Else $n' \leftarrow \lceil \frac{n}{2} \rceil$	
$\mathbf{x} = (\mathbf{x}_L, \mathbf{x}_R); \mathbf{g} = (\mathbf{g}_L, \mathbf{g}_R)$ $C_L \leftarrow \langle \mathbf{x}_L, \mathbf{g}_R \rangle, C_R \leftarrow \langle \mathbf{x}_R, \mathbf{g}_L \rangle$	$\xrightarrow{\quad C_L, C_R \quad}$	
	$\xleftarrow{\quad r \quad}$	$r \leftarrow [0, 2^{\lambda - 1})$
$\mathbf{x}' \leftarrow \mathbf{x}_L + r\mathbf{x}_R$ $C' \leftarrow \langle \mathbf{x}', \mathbf{g}_R + r\mathbf{g}_L \rangle$		$C' \leftarrow C_L + r^2 C_R + rC$ $\mathbf{g}' \leftarrow \mathbf{g}_R + r\mathbf{g}_L$
$\mathcal{P}_{\mathsf{SRO}}(n', \mathbf{x}', C', \mathbf{g}')$	$\xleftarrow{\quad \text{recurse} \quad}$	$\mathcal{V}_{\mathsf{SRO}}(n', C', \mathbf{g}')$

An important observation, as first noted in [BBBPWM18] and leveraged in Halo [BGH19], is that the verifier does not use its input $\mathbf{g}$ unless $n = 1$. Thus, the verifier does not need to compute $\mathbf{g} \in \mathbb{G}^{2^i}$ at each round of recursion. Instead, it only needs to compute the final $g' \in \mathbb{G}$ at the final round, which can be computed as $g' = \sum_{i=1}^{n} u_i \mathbf{g}_i$ where u_i is the ith coefficient of the polynomial $u(X) = \prod_{j=1}^{\log n}(r_j + X^{2^{j-1}})$ where r_j is the verifier's challenge at the jth round.

Application: polynomial commitment scheme Given any prime p of size $|p| \geq \beta$ and any deterministic homomorphic commitment scheme $(\mathsf{Setup}, \mathsf{com}, \mathsf{Vf})$ that is binding over $D = \{\mathbf{x} \in \mathbb{Z}^n : ||\mathbf{x}|| \leq \xi \cdot \beta\}$, where $\mathsf{com} : \mathbb{Z}^n \to \mathbb{G}_{\mathsf{com}}$ is a homomorphism and ξ is the slack parameter of the homomorphism-preimage (HPI) protocol, we can commit to the coefficient vector $\mathbf{x} \in [0, p)^n$ of a degree-n polynomial $f_{\mathbf{x}}(Z) = \sum_i \mathbf{x}_i Z^i$ as $C = \mathsf{com}(\mathbf{x})$ and open evaluations of the committed polynomial at any point $z \in \mathbb{Z}_p$ to $t = f_{\mathbf{x}}(z)$ by running the HPI protocol for the homomorphism $\mathbf{x} \mapsto (\mathsf{com}(\mathbf{x}), f_{\mathbf{x}}(z))$. Setting $\beta_1 = 2^{\lambda \log n} \sqrt{\xi} \cdot \beta$ and $\beta_2 = \sqrt{\xi}/2^{\lambda \log n}$, this protocol proves knowledge of a pair $(a, \mathbf{x}^*)$ such that $\mathsf{com}(\mathbf{x}^*) = a \cdot C$ and $f_{\mathbf{x}^*}(z) = a \cdot t \mod p$ where $||\mathbf{x}|| \leq \beta_1$ and $|a| \leq \beta_2$. This is a valid *rational opening* of C to $\mathbf{x}^*/a$, and by Lemma 2 $|a| \cdot ||\mathbf{x}^*|| \leq \xi\beta$. to open the polynomial evaluation $f_{\mathbf{x}}(z) = \sum_i \mathbf{x}_i z^i \mod p$ at a point $z \in \mathbb{Z}_p$. The protocol has logarithmic communication complexity but requires a linear verifier in order to compute the final g' value.

Special case: DARK rational openings with log verifier In the special case that $\mathbb{G}$ is a group of unknown order and $\mathsf{com}(\mathbf{x}) = g^{f_{\mathbf{x}}(q)}$ for $q \in \mathbb{N}$ and g a random element in $\mathbb{G}$, where $f_{\mathbf{x}}(q) = \sum_{i=1}^{n} \mathbf{x}_i q^{i-1}$, the verifier complexity can be made logarithmic by adding a *proof-of-exponentiation* (PoE)[Wes19] step. By Corollary 5, for $q = \lceil \sqrt{n} 2\xi\beta \rceil$ this commitment scheme is binding over vectors in $\{\mathbf{x} \in \mathbb{Z}^n : ||\mathbf{x}||_2 \leq \xi\beta\}$ and for any $\beta_{\mathsf{n}}, \beta_{\mathsf{d}} > 0$ such that $\beta_{\mathsf{n}} \cdot \beta_{\mathsf{d}} = \xi\beta$, is binding under rational openings to $\mathcal{W}(\beta_{\mathsf{n}}, \beta_{\mathsf{d}}) := \{\mathbf{x}/z \in \mathbb{Q}^n : z \in \mathbb{Z}, ||\mathbf{x}||_2 \leq \beta_{\mathsf{n}}, |z| \leq \beta_{\mathsf{d}}\}$.

When using this as a linear-map vector commitment scheme over $\mathbb{F}_p$, where the honest prover may commit to any $\mathbf{x} \in [0, p)^n$ which has L2 norm at most $\beta = \sqrt{n} \cdot p$,

we would set $q = \lceil n2\xi p \rceil$. The resulting interactive argument in this case is a slight variation of the DARK evaluation protocol.

The verifier complexity is made logarithmic using a PoE as follows. Recall that the verifier only needs to compute the final round $g' = \prod \mathbf{g}_i^{u_i}$ where u_i are the coefficients of $u(X) = \prod_{j=1}^{\log n}(r_j + X^{2^{j-1}})$. In this case, $\mathbf{g}_i = g^{(q^i)}$, and thus $g' = g^{(\sum_i u_i q^i)} = g^{u(q)}$. The Wesolowski PoE[Wes19] enables a Prover to convince a Verifier that $Y = X^a \in \mathbb{G}$ for $X, Y \in \mathbb{G}$ and $a \in \mathbb{Z}$. In the protocol, the verifier sends a random 2λ-bit prime ℓ, and the prover computes $Q \leftarrow g^{(\lfloor \frac{a}{\ell} \rfloor)}$ and sends it to the verifier. The verifier then computes $r \leftarrow a \bmod \ell$ and checks that $Y = X^r + Q^\ell$. The protocol is secure under the adaptive root assumption in $\mathbb{G}$. Note that the verifier just has to compute $a \bmod \ell$ and do two $O(\lambda)$-bit scalar multiplications. The prover runs the PoE protocol on input $(g, g', u(q))$ in order to show that g' was constructed correctly. Note that the verifier can evaluate $u(q) \bmod \ell$ in $O(\log(n))$ field multiplications in $\mathbb{F}_\ell$. To do this the prover computes $q^{(2^j)} \bmod \ell$ for $j \in [0, \log_2(n) - 1]$ using $\log_2(n)$ multiplications. And then evaluates $u(q) \bmod \ell = \prod_{j=1}^{\log n}(r_j + X^{2^{j-1}}) \bmod \ell$ using another $\log_2(n)$ multiplications. Alternatively, one can use Pietrzak's proof of exponentiation[Pie19]. The protocol only works for powers of 2. The prover would send the $g_i \leftarrow g^{q^{2^i}}$ for all $i \in [0, \log_2(n))$. The verifier can efficiently compute $g^{u(q)}$ given the g_i values. The proof size would be $O(\log_2(n))$, but importantly, it can be instantiated without any assumptions in all groups[HHKKP22]. Using preprocessing of q, it also has prover $O(\sqrt{(n)})$, independent of $|q|$.

6.2 Almost special soundness analysis

Theorem 5 (Bulletproofs for short openings is AMSS). Let $n = 2^\mu$ for $\mu \geq 1$. Let $\mathsf{CSZ}_{\mu,\lambda} = 8\mu^2 + \log_2(2\mu)\lambda$.[9] Let $\Gamma = (\mathsf{Setup}, \mathcal{G}, \mathsf{Commit}, \mathsf{Open})$ denote any $\mathbb{Z}$-linear deterministic commitment scheme satisfying the following conditions:

- For $\beta_\mathsf{n} = 2^{\lambda\mu}\sqrt{\xi} \cdot \beta$ and $\beta_\mathsf{d} = \sqrt{\xi}/2^{\lambda\mu}$ where $\log \xi = (14 + 2\mu)\lambda + 6\mathsf{CSZ}_{\mu,\lambda} + 2(\mu + 4)$ it is binding to $\mathcal{W}(\beta_\mathsf{n}, \beta_\mathsf{d}) := \{\mathbf{x}/z \in \mathbb{Q}^n : z \in \mathbb{Z}, ||\mathbf{x}||_2 \leq \beta_\mathsf{n}, |z| \leq \beta_\mathsf{d}\}$ under rational openings.
- $\mathsf{Open}(\mathsf{pp}_\iota, C, \mathbf{x})$ receives no hint, i.e. checks $\mathsf{Commit}(\mathsf{pp}_\iota, \mathbf{x}) = C$.
- The group $\mathbb{G}$ defined for any $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ either has prime-order, or is a GUO (i.e., the order assumption (Assumption 1) holds in $\mathbb{G}$).

Let Π denote the interactive protocol which runs the protocol in Figure 1 for parameters β and $\mathbf{g} \in \mathbb{G}^n$ defined by $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, an index $\iota \in \mathcal{I}$, $\mathsf{pp}_\iota = \mathcal{G}(\mathsf{pp}, \iota)$, and $\mathsf{Commit}(\mathsf{pp}_\iota, \mathbf{x}) = \langle \mathbf{x}, \mathbf{g} \rangle$ for $\mathbf{x} \in \mathbb{Z}^n$. There exists a predicate pair $\phi = (\phi_a, \phi_b)$ and commitment scheme Γ^* such that Π is an $(4^{(\mu)}, \frac{3\mu}{2^\lambda}, \Gamma^*, \phi)$-almost-special-sound interactive argument for the relation:

$$\mathsf{SRO}_{\beta,\mathsf{pp}_\iota} = \{(C, \mathsf{w} = (z, \mathbf{x})) : \langle \mathbf{x}, \mathbf{g} \rangle = z \cdot C \text{ and } \mathbf{x}/z \in \mathcal{W}(\beta_\mathsf{n}, \beta_\mathsf{d})\}$$

Remark 8. Note that $\xi \in 2^{O(\lambda \log n + \log^2 n)}$. $\mathsf{CSZ}_{\mu,\lambda}$ is derived from the Multilinear Composite Schwartz Zippel Theorem (Theorem 2). We can also substitute any value for $\mathsf{CSZ}_{\mu,\lambda}$ using the table of concrete bounds in Theorem 3 for fixed 120-bit security in place of the analytical bound from Theorem 2).

Proof. The commitment scheme $\Gamma^* = (\mathsf{Setup}^*, \mathcal{G}^*, \mathsf{Commit}^*, \mathsf{Open}^*)$ is defined as follows:

- $\mathsf{Setup}^*(1^\lambda)$ first runs $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, which determines $\mathbb{G}$ and $\mathcal{I}$ among other parameters. The new indexing set is defined as $\mathcal{I}^* = \mathcal{I} \cup \bigcup_{i=1}^\mu \mathcal{I} \times \chi^i$.
- $\mathcal{G}^*(\mathsf{pp}, \iota \in \mathcal{I}, \mathbf{r} \in \chi^i)$ for $i \in [0, \mu]$ first computes $\mathsf{pp}_\iota = \mathcal{G}(\mathsf{pp}, \iota)$ to determine the basis $\mathbf{g} \in \mathbb{G}^n$ such that $\mathsf{Commit}(\mathsf{pp}_\iota, \mathbf{x}) = \langle \mathbf{x}, \mathbf{g} \rangle$ for all $\mathbf{x} \in \mathbb{Z}^n$, and then computes $\mathbf{g}^{(i)} = U_i \cdot \mathbf{g}$ where $U_i \in \mathbb{Z}^{2^{\mu-i} \times n}$ is defined recursively as: $U_0 = I_{2^\mu}$ and $U_i = (r_i \cdot I_{2^{\mu-i}} \quad I_{2^{\mu-i}}) \cdot U_{i-1}$ for $i \geq 1$. The notation I_k for $k \in \mathbb{N}$ denotes the $k \times k$ identity matrix. Let $\beta_\mathsf{n}^{(i)} = \beta_\mathsf{n} \cdot 2^{-(i\lambda + \frac{\mu}{2})}$ for $i \geq 1$ and $\beta_\mathsf{n}^{(0)} = \beta_\mathsf{n}$.

  The output $\mathsf{pp}_{\iota,\mathbf{r}}$ determines the message space $\mathcal{M}_i = \{\mathbf{x} \in \mathbb{Z}^{2^{\mu-i}} : ||\mathbf{x}|| \leq \beta_\mathsf{n}^{(i)}\}$ and opening space $\mathcal{W}_i(\beta_\mathsf{n}^{(i)}, \beta_\mathsf{d}) = \{\mathbf{x}/z \in \mathbb{Q}^{2^{\mu-i}} : z \in \mathbb{Z}, ||\mathbf{x}||_2 \leq \beta_\mathsf{n}^{(i)}, |z| \leq \beta_\mathsf{d}\}$ and includes $\mathbf{g}^{(i)} \in \mathbb{G}^{2^{\mu-i}}$.
- $\mathsf{Commit}^*(\mathsf{pp}_{\iota,\mathbf{r}}, \mathbf{x} \in \mathbb{Z}^{2^{\mu-i}})$ returns $C = \langle \mathbf{x}, \mathbf{g}^{(i)} \rangle$, as well as $C_L = \langle \mathbf{x}_L, \mathbf{g}_R^{(i)} \rangle$ and $\langle \mathbf{x}_R, \mathbf{g}_L^{(i)} \rangle$ for $i \in [1, \mu]$. Here $\mathbf{x}_L, \mathbf{g}_L$ denote the left half of $\mathbf{x}$ and $\mathbf{g}$ respectively and $\mathbf{x}_R, \mathbf{g}_R$ the right. For $i = 0$, it just returns $C = \langle \mathbf{x}, \mathbf{g} \rangle$.

---
[9]The $\mathsf{CSZ}_{\mu,\lambda}$ value can be replaced with values from the table in Theorem 3

- $\mathsf{Open}^*(\mathsf{pp}_{\iota,\mathbf{r}}, (C, C_L, C_R), \mathbf{x} \in \mathbb{Z}^{2^{\mu-i}}, z \in \mathbb{Z})$ returns 1 iff $\mathbf{x}/z \in \mathcal{W}_i(\beta_{\mathsf{n}}^{(i)}, \beta_{\mathsf{d}})$, and $\langle \mathbf{x}, \mathbf{g}^{(i)} \rangle = z \cdot C$, $\langle \mathbf{x}_L, \mathbf{g}_R^{(i)} \rangle = z \cdot C_L$, $\langle \mathbf{x}_R, \mathbf{g}_L^{(i)} \rangle = z \cdot C_R$, and $z \neq 0 \bmod |\mathbb{G}|$ for $i \in [1, \mu-1]$. For $i = 0$, the C_L and C_R checks are omitted. (If $\mathbb{G}$ is a GUO, $z \neq 0$ suffices). For $i = \mu$ it returns 1 iff $|\mathbf{x}| \leq \beta_{\mathsf{n}}^{(\mu)}$ and $\mathbf{x} \cdot \mathbf{g}^{(\mu)} = C$.

The setup of the proof system determines the bound $\beta \in \mathbb{R}$, $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, and an index ι and $\mathsf{pp}_\iota = \mathcal{G}(\mathsf{pp}, \iota)$ so that the protocol in Figure 1 is run on shared inputs $\beta, C \in \mathbb{G}$, and $\mathbf{g} \in \mathbb{G}^n$ to the prover/verifier, where $\mathsf{Commit}(\mathsf{pp}_\iota, \mathbf{x}) = \langle \mathbf{x}, \mathbf{g} \rangle$ for $\mathbf{x} \in \mathbb{Z}^n$. The parameters β, pp_ι thus determine the relation $\mathsf{SRO}_{\beta, \mathsf{pp}_\iota}$. This setup can also be viewed as running $\mathsf{Setup}^*(1^\lambda)$ as an extension of $\mathsf{Setup}(1^\lambda)$, which determines the same parameters. The public input C to the relation $\mathsf{SRO}_{\beta, \mathsf{pp}_\iota}$ is thus a commitment for Γ^* at index $\iota \in \mathcal{I}$. By construction, the ith round prover message of the protocol for $i \in [1, \mu]$ is a commitment for Γ^* at index $(\iota, \mathbf{r}_i) \in \mathcal{I} \times \chi^i$ where $\mathbf{r}_i = (r_1, ..., r_i)$ are the first i round challenges in the protocol transcript.

Let $E_i = \begin{pmatrix} r_i \cdot I_{2^{\mu-i}} & I_{2^{\mu-i}} \end{pmatrix}$ so that $U_i = E_i \cdot U_{i-1}$ for all $i \geq 1$. Right-multiplication by U_0 is injective. For $i \geq 1$, right-multiplication by E_i is an injective map from $\mathbb{Z}^{2^{\mu-i}} \to \mathbb{Z}^{2^{\mu-i+1}}$. Thus, by induction, right-multiplication by U_i is an injective map from $\mathbb{Z}^{2^{\mu-i}} \to \mathbb{Z}^{2^\mu}$. The number of non-zero entries in U_1 is exactly 2^μ, and for each $i > 1$ the number of non-zero entries in U_i is no greater than the number of non-zero entries in U_{i-1}. The entries of U_i are subset products of $\{r_1, ..., r_i\}$ and thus each have absolute value at most $2^{i\lambda}$. Therefore, we can bound the Frobenius norm $||U_i|| \leq 2^{i\lambda + \frac{\mu}{2}}$. Let $\beta_{\mathsf{n}}^{(i)} = \beta_{\mathsf{n}} \cdot 2^{-(i\lambda + \frac{\mu}{2})}$. Since the Frobenius norm is 1-submultiplicative, by Lemma 4, the commitment scheme Γ^* at index $(\iota, \mathbf{r}_i)$ is binding over rational openings of C_L, C_R to $\mathcal{W}_i(\beta_{\mathsf{n}}^{(i)}/, \beta_{\mathsf{d}})$.

For each round of the protocol, let r_i denote the ith round challenge and $(C_L^{(i)}, C_R^{(i)})$ denote the ith round prover messages for $i \in [0, \mu-1]$. Let $x_\mu \in \mathbb{Z}$ denote the prover's μth round message. We define $C_0 = C$ (the input) and "virtual" messages $C_{i+1} = C_L^{(i)} + r_{i+1}^2 C_R^{(i)} + r_{i+1} C^{(i)}$ for each $i \in [0, \mu-1]$. In transcript trees we will denote by $C^{(\nu)}$ the virtual commitment corresponding to a node index $\nu \in [0, N]$ (i.e., the virtual commitment defined by the partial transcript ending at node ν). These virtual commitments will be included in the localized transcript tree labeling F.

Transcript tree labeling F In any transcript the ith round prover message is a triple of commitments in $\mathbb{G}^3$. Given a transcript tree defined by $L : [1, N] \to \chi \times \mathbb{G}^3$, the function $F(L)(\nu) = (C^{(\nu)}, C_R^{(\nu)}, C_L^{(\nu)})$ assigns a Γ^* commitment label to node $\nu \in [1, N]$, computed as follows. The root of a transcript tree, which is at index N in a post-order labelling and level 0, is assigned the commitment label $F(L)(N) = C^{(N)} = C$, where C is the input commitment to the protocol. Each internal node of a transcript tree with index ν is assigned $(C^{(\nu)} = C_L^{(\omega)} + r_\nu^2 C_R^{(\omega)} + r_\nu C_\omega, C_L^{(\omega)}, C_R^{(\omega)})$. F is localized because $C^{(\nu)}$ is computed from the partial transcript that L assigns to the predecessor nodes of ν.

Finally, we show that $C^{(\nu)}$ is a correct Γ^* commitment for the index ι_ν. Define $\mathbf{g}_N = \mathbf{g} = \mathcal{G}^*(\mathsf{pp}, \iota)$ for the root, and $\mathbf{g}_\nu = \mathcal{G}^*(\mathsf{pp}, \iota, \mathbf{r}^{(\nu)})$ for a node on level $i \geq 1$ whose transcript prefix has the challenge sequence $\mathbf{r}^{(\nu)} = (r_1^{(\nu)}, ..., r_i^{(\nu)})$. Note that on the ith level $\mathbf{g}_\nu \in \mathbb{G}^{2^{\mu-i}}$. In a correct transcript tree, for any node ν at the ℓ_νth level, $C^{(\nu)} = \langle \mathbf{x}', \mathbf{g}_\nu \rangle$ for some $\mathbf{x}' \in \mathbb{Z}^{2^{\mu-\ell_\nu}}$. Additionally, $C_L^{(\nu)} = \langle (\mathbf{0}, \mathbf{x}'_L), \mathbf{g}_\nu \rangle$ and $C_R^{(\nu)} = \langle (\mathbf{x}'_R, \mathbf{0}), \mathbf{g}_\nu \rangle$ where $\mathbf{x}'_L$ and $\mathbf{x}'_R$ are the left and right half of $\mathbf{x}'$, respectively.

The predicates ϕ_a and ϕ_b We define the numerator bounds $B_0 \geq \cdots \geq B_\mu$ such that $\log B_\mu = \mu\lambda + \log \beta$ (corresponding to the numerator verification bound on the prover's final round message) and $\log B_i = \mathsf{CSZ}_{\mu-i,\lambda} + (2\mu - i)\lambda + \frac{\mu}{2} + \log \beta$ for $i \geq 1$. We define denominator bounds $D_0 \geq \cdots \geq D_\mu$ such that $D_\mu = 1$ (corresponding to the denominator verification bound on the prover's final round message, i.e. that it is an integer) and $\log D_i = \mathsf{CSZ}_{\mu-i,\lambda}$ for $i \geq 1$.

For any $i \in [0, \mu]$ and $\mathbf{m} = \mathbf{x}/z \in \mathbb{Q}^{2^{\mu-i}}$ we define:

- $\phi_a(i, \mathbf{m}) = 1$ iff $\mathbf{m} \in \mathcal{W}_i(B_i, D_i)$

- $\phi_b(i, \mathbf{m}) = 1$ iff $\mathbf{m} \in \mathcal{W}_i(\beta_{\mathsf{n}}^{(i)}/2^{2\lambda+1}, \beta_{\mathsf{d}})$

Note that $\log \beta_{\mathsf{n}}^{(i)} = (2\mu - i)\lambda + \log \beta + 2(\lambda + 1 + \mathsf{CSZ}_{\mu,\lambda}) + \frac{1}{2} \geq \log B_i$ and $\beta_{\mathsf{d}} \geq D_i$ for all $i \in [0, \mu]$, thus $\mathcal{W}_i(B_i, D_i) \subseteq \mathcal{W}_i(\beta_{\mathsf{n}}^{(i)}, \beta_{\mathsf{d}})$. This shows that ith round commitments (i.e., commitment labels assigned by F to nodes on the ith level) are binding under

rational openings to $\mathcal{W}_i(B_i, D_i)$ as a subset of $\mathcal{W}_i(\beta_n^{(i)}, \beta_d)$ and also that $\phi_a(i, \mathbf{m}) = 1$ implies $\phi_b(i, \mathbf{m}) = 1$.

The following facts will be needed later on in the proof:

- $\beta_d = \sqrt{\xi}/2^{\mu\lambda} \geq 2^{3\lambda} \cdot D_0^3$ for $\frac{1}{2}\log\xi \geq (\mu+3)\lambda + 3\mathsf{CSZ}_{\mu,\lambda}$

- $\beta_n^{(i)}/2^{2\lambda+1} = 2^{(\mu-i-2)\lambda-\frac{\mu}{2}-1}\sqrt{\xi}\cdot\beta \geq 2^{6\lambda+2}B_{i+1}D_{i+1}^2$ for $\frac{1}{2}\log\xi \geq (\mu+8)\lambda + 3\mathsf{CSZ}_{\mu,\lambda} + \mu + 3$

Both conditions are satisfied by the theorem hypothesis.

The Extract algorithm The $\mathsf{Extract}(i, \nu, (C^{(\nu)}, C_R^{(\nu)}, C_L^{(\nu)}), \mathsf{openSubtree})$ algorithm operates as follows to extract an opening of $(C^{(\nu)}, C_R^{(\nu)}, C_L^{(\nu)})$ for a node ν on the ith tree level. The input $\mathsf{openSubtree}$ contains openings $(\mathbf{x}_{u_j}, t_{u_j})$ for all three children nodes $u_1, u_2,$ and u_3 of node ν. Let $C_j = F(L)(u_j)$ and let r_j denote the challenge labels that L assigns to u_j where $r_1 \neq r_2 \neq r_3 \neq r_4 \neq 0$. By construction, $C_j = C_L^{(\nu)} + r_j^2 C_R^{(\nu)} + r_j C^{(\nu)}$ for $j \in \{1, 2, 3, 4\}$. Define $\mathbf{R} \in \mathbb{Z}^{3\times 3}$ to be the matrix with rows $(r_j, r_j^2, 1)$ for $j \in \{1, 2, 3\}$, $\mathbf{T}$ the diagonal matrix with diagonal entries $t_{u_1}, t_{u_2}, t_{u_3} \neq 0$ (also non-zero mod prime $|\mathbb{G}|$), and let $E_j = \begin{pmatrix} r_j \cdot I_{2^{\mu-i-1}} & I_{2^{\mu-i-1}} \end{pmatrix}$. We can summarize the relations as:

$$\mathbf{R} \cdot \begin{bmatrix} C^{(\nu)} \\ C_R^{(\nu)} \\ C_L^{(\nu)} \end{bmatrix} = \begin{bmatrix} C^{(u_1)} \\ C^{(u_2)} \\ C^{(u_3)} \end{bmatrix} \qquad \mathbf{T} \cdot \begin{bmatrix} C^{(u_1)} \\ C^{(u_2)} \\ C^{(u_3)} \end{bmatrix} = \begin{bmatrix} \langle \mathbf{x}_{u_1}, \mathbf{g}_{u_1} \rangle \\ \langle \mathbf{x}_{u_2}, \mathbf{g}_{u_2} \rangle \\ \langle \mathbf{x}_{u_3}, \mathbf{g}_{u_3} \rangle \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{u_1} \cdot E_1 \\ \mathbf{x}_{u_2} \cdot E_2 \\ \mathbf{x}_{u_3} \cdot E_3 \end{bmatrix} \cdot \mathbf{g}_\nu$$

$$\mathbf{T} \cdot \mathbf{R} \cdot \begin{bmatrix} C^{(\nu)} \\ C_R^{(\nu)} \\ C_L^{(\nu)} \end{bmatrix} = \mathbf{T} \cdot \begin{bmatrix} C^{(u_1)} \\ C^{(u_2)} \\ C^{(u_3)} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{u_1} \cdot E_1 \\ \mathbf{x}_{u_2} \cdot E_2 \\ \mathbf{x}_{u_3} \cdot E_3 \end{bmatrix} \cdot \mathbf{g}_\nu = \mathbf{X} \cdot \mathbf{g}_\nu.$$

Let $\mathbf{P} = adj(\mathbf{T} \cdot \mathbf{R}) \in \mathbb{Z}^{3\times 3}$, the adjugate matrix so that $\mathbf{P} \cdot \mathbf{T} \cdot \mathbf{R} = \det(\mathbf{T}) \cdot \det(\mathbf{R}) \cdot \mathbf{I}_3$. Let $t_\nu = \det(\mathbf{T}) \cdot \det(\mathbf{R}) = t_{u_1} \cdot t_{u_2} \cdot t_{u_3} \cdot \det(\mathbf{R})$. Since $\mathbf{R}$ is Vandermonde it has a non-zero integer determinant, which is also non-zero mod prime $|\mathbb{G}|$. Therefore $t_\nu \neq 0$, and also $t_\nu \neq 0 \bmod |\mathbb{G}|$ in the case that $|\mathbb{G}|$ is prime.

Finally:

$$\mathbf{P} \cdot \mathbf{T} \cdot \mathbf{R} \cdot \begin{bmatrix} C^{(\nu)} \\ C_R^{(\nu)} \\ C_L^{(\nu)} \end{bmatrix} = t_\nu \cdot \begin{bmatrix} C^{(\nu)} \\ C_R^{(\nu)} \\ C_L^{(\nu)} \end{bmatrix} = \mathbf{P} \cdot \mathbf{X} \cdot \mathbf{g}_\nu.$$

Letting $\mathbf{P}_1$ denote the first row of $\mathbf{P}$ we obtain $t_\nu \cdot C_\nu = \langle \mathbf{P}_1, \mathbf{X} \cdot \mathbf{g}_\nu \rangle$. The extractor sets $\mathbf{x}^{(\nu)} := \langle \mathbf{P}_1, \mathbf{X} \rangle$, which now satisfies $\langle \mathbf{x}^{(\nu)}, \mathbf{g}_\nu \rangle = t_\nu \cdot C_\nu$. Similarly, it sets $\mathbf{x}_R^{(\nu)} = \langle \mathbf{P}_2, \mathbf{X} \rangle$ and $\mathbf{x}_L^{(\nu)} = \langle \mathbf{P}_3, \mathbf{X} \rangle$, which satisfy $\langle \mathbf{x}_R^{(\nu)}, g_\nu \rangle = t_\nu \cdot C_R^{(\nu)}$ and $\langle \mathbf{x}_L^{(\nu)}, g_\nu \rangle = t_\nu \cdot C_L^{(\nu)}$.

If all entries of $\mathbf{T}$ have absolute value bounded by D_{i+1} then $|t_\nu| \leq 2^{3\lambda} D_{i+1}^3$. We will next bound the norm of $\mathbf{x}_\nu$ assuming that all $\mathbf{x}_{u_i}$ have norm bounded by B_{i+1}, first note that all entries of the adjugate matrix are determinants of 2×2 submatrices, thus $||adj(\mathbf{T})|| \leq D_{i+1}^2\sqrt{3}$ and $||adj(\mathbf{R})|| \leq 2^{3\lambda}\sqrt{3}$. Since $\mathbf{P} = adj(\mathbf{T} \cdot \mathbf{R}) = adj(\mathbf{R}) \cdot adj(\mathbf{T})$, and since the Frobenius norm is 1-submultiplicative, $||\mathbf{P}|| \leq 3 \cdot 2^{3\lambda} D_{i+1}^2$. For each j, $||\mathbf{x}_{u_j} \cdot E_j||^2 = (r_j^2 + 1) \cdot ||\mathbf{x}_{u_j}||^2$, thus $||\mathbf{X}||^2 = \sum_{j=1}^{3} ||\mathbf{x}_{u_j} \cdot E_j||^2 \leq 3 \cdot (2^{2\lambda} + 1) \cdot B_{i+1}^2$. Finally, for any $j \in \{1, 2, 3\}$, $||\langle \mathbf{P}_j, \mathbf{X} \rangle|| \leq ||\mathbf{P} \cdot \mathbf{X}|| \leq 2^{3\lambda} D_{i+1}^2 \cdot \sqrt{3}(2^\lambda + 1)B_{i+1} \leq 2^{4\lambda+1}B_{i+1}D_{i+1}^2$. As shown above, this implies $\mathbf{x}^{(\nu)}/t_\nu, \mathbf{x}_R^{(\nu)}/t_\nu, \mathbf{x}_L^{(\nu)}/t_\nu \in \mathcal{W}_i(\beta_n^{(i)}, \beta_d)$.

Note that for all $i \in \{1, 2, 3, 4\}$ we have that:

$$t_{u_i} \cdot \langle (r_i, r_i^2, 1), (C^{(\nu)}, C_R^{(\nu)}, C_L^{(\nu)}) \rangle = \langle (r_i \mathbf{x}_{u_i}, \mathbf{x}_{u_i}), \mathbf{g}_\nu \rangle$$

Multiplying both sides by t_ν we get:

$$t_{u_i} t_\nu \cdot \langle (r_i, r_i^2, 1), (C^{(\nu)}, C_R^{(\nu)}, C_L^{(\nu)}) \rangle = t_\nu \langle (r_i \mathbf{x}_{u_i}, \mathbf{x}_{u_i}), \mathbf{g}_\nu \rangle$$

We can now replace $(C^{(\nu)}, C_R^{(\nu)}, C_L^{(\nu)})$ with the extracted openings:

$$t_{u_i} \cdot \langle r_i \mathbf{x}^{(\nu)} + r_i^2 \mathbf{x}_R^{(\nu)} + \mathbf{x}_L^{(\nu)}, \mathbf{g}_\nu \rangle = t_\nu \langle (r_i \mathbf{x}_{u_i}, \mathbf{x}_{u_i}), \mathbf{g}_\nu \rangle$$

Let $\mathbf{x}^\nu = (\mathbf{a}, \mathbf{b})$, $\mathbf{x}_L^\nu = (\mathbf{a}_L, \mathbf{b}_L)$ and $\mathbf{x}_R^\nu = (\mathbf{a}_R, \mathbf{b}_R)$ denote the subdivision of each vector into two equal length halves. Two cases ensue. Either

- $t_{u_i} \cdot (r_i \cdot \mathbf{a} + r_i^2 \mathbf{a}_R + \mathbf{a}_L) = t_\nu r_i \mathbf{x}_{u_i}$ and $t_{u_i} \cdot (r_i \cdot \mathbf{b} + r_i^2 \mathbf{b}_R + \mathbf{b}_L) = t_\nu \mathbf{x}_{u_i}$; or

17

- they are distinct openings of the commitment $C^{(u_i)} = \langle (r_i, r_i^2, 1), (C^{(\nu)}, C_R^{(\nu)}, C_L^{(\nu)}) \rangle$ to the rationals $\mathbf{h}_1 = \frac{(r_i \mathbf{x}_{u_i}, \mathbf{x}_{u_i})}{t_\nu}$ and $\mathbf{h}_2 = \frac{r_i \mathbf{x}^{(\nu)} + r_i^2 \mathbf{x}_R^{(\nu)} + \mathbf{x}_L^{(\nu)}}{t_{u_i}}$.

Given that $||\mathbf{x}^{(\nu)}||, ||\mathbf{x}_L^\nu||$ and $||\mathbf{x}^{(\nu)}||$ are all bounded by $B' = 2^{4\lambda+1} B_{i+1} D_{i+1}^2$ we have that $||(r_i \mathbf{x}^{(\nu)} + r_i^2 \mathbf{x}_R^{(\nu)} + \mathbf{x}_L^{(\nu)})||$ is bounded by $3 \cdot 2^{2\lambda} B' \leq 2^{6\lambda+3} B_{i+1} D_{i+1}^2$ and thus $\mathbf{h}_1, \mathbf{h}_2 \in \mathcal{W}_i(\beta_{\mathsf{n}}^{(i)}, \beta_{\mathsf{d}})$. Additionally, $||(\mathbf{x}_{u_i}, \mathbf{x}_{u_i})|| \leq 2^{\lambda+1} \cdot B_{i+1} \leq B'$ and thus $\mathbf{h}_1, \mathbf{h}_2 \in \mathcal{W}_i(\beta_{\mathsf{n}}^{(i)}, \beta_{\mathsf{d}})$. In this case, the algorithm returns these openings as a break of the commitment scheme Γ^*.

Otherwise:

$$r_i \cdot \mathbf{a} + r_i^2 \mathbf{a}_R + r_i \mathbf{a}_L = r_i^2 \cdot \mathbf{b} + r_i^3 \mathbf{b}_R + r_i \mathbf{b}_L \forall i \in \{1, 2, 3, 4\}$$

Since the 4×4 Vandermonde matrix with rows $(1, r_i, r_i^2, r_i^3)$ has a trivial kernel, this equation implies:

$$\mathbf{a} = \mathbf{b}_L \quad \mathbf{b} = \mathbf{a}_R$$
$$\mathbf{b}_R = 0 \quad \mathbf{a}_L = 0$$

Therefore, $(\mathbf{x}^{(\nu)}, t_\nu)$ is a valid opening to the commitment $(C^{(\nu)}, C_R^{(\nu)}, C_L^{(\nu)})$.

At the root, $\mathsf{Extract}(C, \mathsf{openTree})$ runs $\mathsf{Extract}(0, N, C, \mathsf{openTree})$ to either obtain a witness $(\mathbf{x}, t)$ such that $\langle \mathbf{x}, \mathbf{g} \rangle = t \cdot C$ where $\mathbf{x}/t \in \mathcal{W}(\beta_{\mathsf{n}}, \beta_{\mathsf{d}})$ or a break of the commitment scheme Γ^*.

The Extend algorithm For $\mathbf{h} \in \mathbb{Q}^{2^{\mu-i}}$ and $\mu - i$ challenges in χ, the algorithm $\mathsf{Extend}(i, \mathbf{h}, r_{i+1}, ..., r_\mu)$ sets $\mathbf{h}_i := \mathbf{h}$ and runs the following iterative algorithm: for $j = i$ to $\mu - 1$ set $\mathbf{h}_{j+1} := \mathbf{h}_{j,L} + r_{j+1} \cdot \mathbf{h}_{j,R}$ where $\mathbf{h}_{j,L}$ and $\mathbf{h}_{j,R}$ are the left/right coefficient split of $\mathbf{h}_j$. It returns $\mathbf{h}_i, ..., \mathbf{h}_\mu$.

Note that if $\mathbf{h}_i$ is the honest prover's committed state in the ith round of the protocol Π and $r_{i+1}, ..., r_\mu$ are the last $\mu - i$ round challenges then the value $\mathbf{h}_\mu \in \mathbb{Z}$ returned by $\mathsf{Extend}(i, \mathbf{h}_i, r_{i+1}, ..., r_\mu)$ returns is also the honest prover's last message to the verifier.

The Break algorithm Given virtual commitments $(C^{(1)}, C_L^{(1)}, C_R^{(1)}), ..., (C^{(\mu)}, C_L^{(\mu)}, C_R^{(\mu)})$, openings for the last $\mu - i$ commitments, and round challenges $r_1, ..., r_\mu$, the algorithm first runs $\mathsf{Extend}(i, \mathbf{h}, r_i, .., r_\mu)$, which returns rational vectors $\mathbf{h}'_{i+1}, ..., \mathbf{h}'_\mu$. If $\forall_{j \geq i} \mathbf{h}'_j = \mathbf{x}_j / z_j$ then it outputs $\perp$. Otherwise, let $j \in [i, \mu)$ be the *first* index where $\mathbf{h}'_{j+1} \neq \mathbf{x}_{j+1} / z_{j+1}$ and output $(\mathbf{x}_{j,L} + r_{j+1} \cdot \mathbf{x}_{j,R}, z_j)$, where $\mathbf{x}_{j,L}$ and $\mathbf{x}_{j,R}$ are the left/right halves of $\mathbf{x}_j$, as the attempted opening for $C^{(j+1)}$.

Subclaim 1 (Application of New Schwartz-Zippel Lemma). For any $i \in [\mu]$ and $\mathbf{h} \in \mathbb{Q}^{2^{\mu-i}}$, if $\phi_a(i, \mathbf{h}) = 0$ then the probability over uniform i.i.d. $r_{i+1}, ..., r_\mu$ that $\mathbf{h}_\mu$ returned by $\mathsf{Extend}(i, \mathbf{h}, r_{i+1}, ..., r_\mu)$ satisfies $\phi_a(\mu, \mathbf{h}_\mu) = 1$ is at most $\frac{3(\mu-i)}{2^\lambda}$.

Let $\mathbf{x}/N = \mathbf{h}$ for $\gcd(\mathbf{x}, N) = 1$ denote the reduced form of $\mathbf{h} \in \mathbb{Q}^{2^{\mu-i}}$. If $\phi_a(i, \mathbf{h}) = 0$ then either $z > D_i$ or $||\mathbf{x}|| > B_i$ while $\phi_a(\mu, \mathbf{h}_\mu) = 1$ implies $\mathbf{h}_\mu \in \mathbb{Z}$ and $|\mathbf{h}_\mu| \leq B_\mu$. Let $\mu' = \mu - i$. Observe that $\mathbf{h}_\mu = \frac{1}{z} \cdot f(r_\mu, ..., r_{i+1})$ where f is a μ'-linear polynomial with coefficient vector $\mathbf{x}$.

Case 1 $N > D_i$:
If $|N| > D_i$ then since $\log D_i = \mathsf{CSZ}_{\mu-i}$, the probability that $\mathbf{h}_\mu \in \mathbb{Z}$ is:

$$\mathbb{P}_{(r_{i+1}, ..., r_\mu) \leftarrow [0, 2^\lambda)^{\mu-i}}[f(r_\mu, ..., r_{i+1}) \equiv 0 \bmod N] \leq \frac{\mu - i + 1}{2^\lambda}.$$

by the Multilinear Composite Schwartz-Zippel Lemma (Theorem 2).

Case 2 $N \leq D_i \ \wedge \ ||\mathbf{x}|| > B_i$:
In this case, if $|\mathbf{h}_\mu| \leq B_\mu$ then $|f(r_\mu, ..., r_{i+1})| \leq N \cdot B_\mu \leq 2^{\mathsf{CSZ}_{\mu-i,\lambda}} \cdot B_\mu$. Furthermore, the fact that $||\mathbf{x}|| > B_i$ and $\log B_i = \mathsf{CSZ}_{\mu-i,\lambda} + (\mu - i)\lambda + \frac{\mu}{2} + \log B_\mu$ imply:

$$\log(\mathsf{CSZ}_{\mu-i,\lambda} \cdot B_\mu) \leq \log B_i - (\mu - i)\lambda - \frac{\mu}{2} < \log ||\mathbf{x}|| - (\mu - i)\lambda - \frac{\mu}{2}.$$

Since $||\mathbf{x}|| \leq 2^{\mu/2} \cdot ||f||_\infty$, the Evaluation Bound Lemma (Lemma 7) implies:

$$\mathbb{P}[\mathbf{h}_\mu \leq B_\mu] \leq \mathbb{P}[|f(r_\mu, ..., r_{i+1})| \leq D_i \cdot B_\mu] \leq \mathbb{P}[|f(r_\mu, ..., r_{i+1})|$$
$$\leq \frac{1}{2^{(\mu-i)\lambda}} \cdot ||f||_\infty \leq \frac{3(\mu - i)}{2^\lambda}.$$

18

Together these imply that if $\phi_a(i, \mathbf{x}/N) = 0$ then the probability over the random challenges that the final element $\mathbf{h}_\mu$ of the list returned by Extend satisfies $\phi_a(\mu, \mathbf{h}_\mu) = 1$ is negligible.

$\square$

Subclaim 2. For any $i \in [\mu - 1]$, given a valid transcript together with the "virtual" commitments $(C^{(0)}, ..., C^{(\mu)})$, round challenges $(r_1, ..., r_\mu)$, and openings $(o_i, ..., o_\mu)$ of the $\mu - i + 1$ last Γ^*-commitment triples $\mathcal{C}_i, ..., \mathcal{C}_\mu$ where for $j \in [i, \mu]$:

- $\mathcal{C}_j = (C^{(j)}, C_L^{(j)}, C_R^{(j)})$
- o_j contains an opening of $C^{(j)}$ to a rational vector $\mathbf{h}_j$, such that $\phi_b(j, \mathbf{h}_j) = 1$
- $\phi_a(j, \mathbf{h}_j) = 1$ if $j \geq i + 1$

then either $\mathsf{Extend}(i, \mathbf{h}_i, r_{i+1}, ..., r_\mu)$ returns $\mathbf{h}_{i+1}, ..., \mathbf{h}_\mu$ or $\mathsf{Break}(i, \mathbf{h}_i, (o_i, ..., o_\mu))$ returns for some $j \geq i$ a valid opening of $\mathcal{C}^{(j)}$ to $\mathbf{h}'_j \neq \mathbf{h}_j$.

Proof. Let $(\mathbf{h}'_{i+1}, ..., \mathbf{h}'_\mu)$ denote the output of $\mathsf{Extend}(i, \mathbf{h}_i, r_{i+1}, ..., r_\mu)$ and suppose it is not equal to $(\mathbf{h}_{i+1}, ..., \mathbf{h}_\mu)$. Let $j \in [i, \mu)$ denote the *first* index for which $\mathbf{h}_{j+1} \neq \mathbf{h}'_{j+1}$. This means that $\mathbf{h}_j = \mathbf{h}'_j$ and thus $\mathbf{h}'_{j+1} = \mathbf{h}_{j,L} + r_{j+1} \cdot \mathbf{h}_{j,R}$ where $\mathbf{h}_{j,L}$ and $\mathbf{h}_{j,R}$ are the left/right halves of $\mathbf{h}_j$. Let $\mathbf{g}^{(j)} = \mathcal{G}^*(\mathsf{pp}, \iota, \mathbf{r}^{(j)})$ denote the commitment basis determined by $r_1, ..., r_j$, let $\mathbf{g}^{(j)} = (\mathbf{g}_L^{(j)}, \mathbf{g}_R^{(j)})$, and let $\mathbf{g}^{(j+1)} = r_{j+1} \cdot \mathbf{g}_L^{(j)} + \mathbf{g}_R^{(j)}$.

Let $\mathbf{h}_j = \mathbf{x}_j/z_j$ so that $\langle \mathbf{x}_j, \mathbf{g}_j \rangle = z_j \cdot C^{(j)}$ by validity of the opening of $\mathcal{C}_j$. Let $\mathbf{x}_j = (\mathbf{x}_{j,L}, \mathbf{x}_{j,R})$ so that $\mathbf{x}'_{j+1} = \mathbf{x}_{j,L} + r_{j+1}\mathbf{x}_{j,R}$ and $\mathbf{h}'_{j+1} = \mathbf{x}'_{j+1}/z_j$. Validity of the opening of $\mathcal{C}_j$ additionally implies $z_j \cdot C_L^{(j)} = \langle \mathbf{x}_{j,L}, \mathbf{g}_R^{(j)} \rangle$ and $z_j \cdot C_R^{(j)} = \langle \mathbf{x}_{j,R}, \mathbf{g}_L^{(j)} \rangle$. Furthermore, $\phi_b(j, \mathbf{h}_j) = 1$ implies $\|\mathbf{x}'_{j+1}\| \leq 2^{\lambda+1}\|\mathbf{x}_j\| \leq \beta_\mathsf{n}^{(j)}/2^\lambda = \beta_\mathsf{n}^{(j+1)}$ and thus $\mathbf{h}'_{j+1} \in \mathcal{W}(\beta_\mathsf{n}^{(i)}, \beta_\mathsf{d})$.

In a correct transcript:

$$\langle \mathbf{x}_{j,L}, \mathbf{g}_R^{(j)} \rangle + r_{j+1}^2 \langle \mathbf{x}_{j,L}, \mathbf{g}_R^{(j)} \rangle + r_{j+1}\langle x_j, \mathbf{g}^{(j)} \rangle = z_j \cdot C^{(j+1)}$$

Thus $\langle \mathbf{x}_{j,L} + r_{j+1} \cdot \mathbf{x}_{j,R}, \mathbf{g}_R^{(j)} + r_{j+1}\mathbf{g}_L^{(j)} \rangle = \langle \mathbf{x}'_{j+1}, \mathbf{g}^{(j+1)} \rangle = z_j \cdot C^{(j+1)}$. This shows that $\mathbf{h}'_{j+1} \neq \mathbf{h}_{j+1}$ is a conflicting valid opening to $C^{(j+1)}$. $\square$

References

[AC20]   Thomas Attema and Ronald Cramer. "Compressed Σ-Protocol Theory and Practical Application to Plug & Play Secure Algorithmics". In: *CRYPTO 2020, Part III*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12172. LNCS. Springer, Heidelberg, Aug. 2020, pp. 513–543. DOI: 10.1007/978-3-030-56877-1_18.

[ACK21a]   Thomas Attema, Ronald Cramer, and Lisa Kohl. *A Compressed Σ-Protocol Theory for Lattices*. Cryptology ePrint Archive, Report 2021/307. https://eprint.iacr.org/2021/307. 2021.

[ACK21b]   Thomas Attema, Ronald Cramer, and Lisa Kohl. "A Compressed Σ-Protocol Theory for Lattices". In: *CRYPTO 2021, Part II*. Ed. by Tal Malkin and Chris Peikert. Vol. 12826. LNCS. Virtual Event: Springer, Heidelberg, Aug. 2021, pp. 549–579. DOI: 10.1007/978-3-030-84245-1_19.

[AFK21]   Thomas Attema, Serge Fehr, and Michael Klooß. *Fiat-Shamir Transformation of Multi-Round Interactive Proofs*. Cryptology ePrint Archive, Report 2021/1377. https://eprint.iacr.org/2021/1377. 2021.

[AGLMS22]   Arasu Arun, Chaya Ganesh, Satya Lokam, Tushar Mopuri, and Sriram Sridhar. *Dew: Transparent Constant-sized zkSNARKs*. Cryptology ePrint Archive, Report 2022/419. https://eprint.iacr.org/2022/419. 2022.

[AL21]   Martin R. Albrecht and Russell W. F. Lai. "Subtractive Sets over Cyclotomic Rings - Limits of Schnorr-Like Arguments over Lattices". In: *CRYPTO 2021, Part II*. Ed. by Tal Malkin and Chris Peikert. Vol. 12826. LNCS. Virtual Event: Springer, Heidelberg, Aug. 2021, pp. 519–548. DOI: 10.1007/978-3-030-84245-1_18.

[ALS20]   Thomas Attema, Vadim Lyubashevsky, and Gregor Seiler. "Practical Product Proofs for Lattice Commitments". In: *CRYPTO 2020, Part II.* Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12171. LNCS. Springer, Heidelberg, Aug. 2020, pp. 470–499. DOI: `10.1007/978-3-030-56880-1_17`.

[BBBPWM18]   Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. "Bulletproofs: Short Proofs for Confidential Transactions and More". In: *2018 IEEE Symposium on Security and Privacy.* IEEE Computer Society Press, May 2018, pp. 315–334. DOI: `10.1109/SP.2018.00020`.

[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: *EUROCRYPT 2016, Part II.* Ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9666. LNCS. Springer, Heidelberg, May 2016, pp. 327–357. DOI: `10.1007/978-3-662-49896-5_12`.

[BCPS15]   Anurag Bishnoi, Pete L. Clark, Aditya Potukuchi, and John R. Schmitt. *On zeros of a polynomial in a finite grid.* 2015. DOI: `10.48550/ARXIV.1508.06020`. URL: `https://arxiv.org/abs/1508.06020`.

[BCS21]   Jonathan Bootle, Alessandro Chiesa, and Katerina Sotiraki. "Sumcheck Arguments and Their Applications". In: *CRYPTO 2021, Part I.* Ed. by Tal Malkin and Chris Peikert. Vol. 12825. LNCS. Virtual Event: Springer, Heidelberg, Aug. 2021, pp. 742–773. DOI: `10.1007/978-3-030-84242-0_26`.

[BDFG21]   Dan Boneh, Justin Drake, Ben Fisch, and Ariel Gabizon. "Halo Infinite: Proof-Carrying Data from Additive Polynomial Commitments". In: *CRYPTO 2021, Part I.* Ed. by Tal Malkin and Chris Peikert. Vol. 12825. LNCS. Virtual Event: Springer, Heidelberg, Aug. 2021, pp. 649–680. DOI: `10.1007/978-3-030-84242-0_23`.

[BF22]   Benedikt Bünz and Ben Fisch. *Schwartz-Zippel for multilinear polynomials mod N.* Cryptology ePrint Archive, Report 2022/458. `https://eprint.iacr.org/2022/458`. 2022.

[BFS19]   Benedikt Bünz, Ben Fisch, and Alan Szepieniec. *Transparent SNARKs from DARK Compilers.* Cryptology ePrint Archive, Report 2019/1229. `https://eprint.iacr.org/2019/1229`. 2019.

[BFS20]   Benedikt Bünz, Ben Fisch, and Alan Szepieniec. "Transparent SNARKs from DARK Compilers". In: *EUROCRYPT 2020, Part I.* Ed. by Anne Canteaut and Yuval Ishai. Vol. 12105. LNCS. Springer, Heidelberg, May 2020, pp. 677–706. DOI: `10.1007/978-3-030-45721-1_24`.

[BG93]   Mihir Bellare and Oded Goldreich. "On Defining Proofs of Knowledge". In: *CRYPTO'92.* Ed. by Ernest F. Brickell. Vol. 740. LNCS. Springer, Heidelberg, Aug. 1993, pp. 390–420. DOI: `10.1007/3-540-48071-4_28`.

[BGH19]   Sean Bowe, Jack Grigg, and Daira Hopwood. *Halo: Recursive Proof Composition without a Trusted Setup.* Cryptology ePrint Archive, Report 2019/1021. `https://eprint.iacr.org/2019/1021`. 2019.

[BH01]   Johannes Buchmann and Safuat Hamdy. "A survey on IQ cryptography". In: *Public-Key Cryptography and Computational Number Theory.* 2001, pp. 1–15.

[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: *CRYPTO 2021, Part IV.* Ed. by Tal Malkin and Chris Peikert. Vol. 12828. LNCS. Virtual Event: Springer, Heidelberg, Aug. 2021, pp. 123–152. DOI: `10.1007/978-3-030-84259-8_5`.

[BLNS20]   Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. "A Non-PCP Approach to Succinct Quantum-Safe Zero-Knowledge". In: *CRYPTO 2020, Part II.* Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12171. LNCS. Springer, Heidelberg, Aug. 2020, pp. 441–469. DOI: `10.1007/978-3-030-56880-1_16`.

[CNRZZ22]   Matteo Campanelli, Anca Nitulescu, Carla Ràfols, Alexandros Zacharakis, and Arantxa Zapico. *Linear-map Vector Commitments and their Practical Applications*. Cryptology ePrint Archive, Report 2022/705. https://eprint.iacr.org/2022/705. 2022.

[CPP17]   Geoffroy Couteau, Thomas Peters, and David Pointcheval. "Removing the Strong RSA Assumption from Arguments over the Integers". In: *EUROCRYPT 2017, Part II*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10211. LNCS. Springer, Heidelberg, Apr. 2017, pp. 321–350. DOI: 10.1007/978-3-319-56614-6_11.

[DF02]   Ivan Damgård and Eiichiro Fujisaki. "A Statistically-Hiding Integer Commitment Scheme Based on Groups with Hidden Order". In: *ASIACRYPT 2002*. Ed. by Yuliang Zheng. Vol. 2501. LNCS. Springer, Heidelberg, Dec. 2002, pp. 125–142. DOI: 10.1007/3-540-36178-2_8.

[DL77]   Richard A DeMillo and Richard J Lipton. *A Probabilistic Remark on Algebraic Program Testing*. Tech. rep. GEORGIA INST OF TECH ATLANTA SCHOOL OF INFORMATION and COMPUTER SCIENCE, 1977.

[ENS20]   Muhammed F. Esgin, Ngoc Khanh Nguyen, and Gregor Seiler. "Practical Exact Proofs from Lattices: New Techniques to Exploit Fully-Splitting Rings". In: *ASIACRYPT 2020, Part II*. Ed. by Shiho Moriai and Huaxiong Wang. Vol. 12492. LNCS. Springer, Heidelberg, Dec. 2020, pp. 259–288. DOI: 10.1007/978-3-030-64834-3_9.

[GI08]   Jens Groth and Yuval Ishai. "Sub-linear Zero-Knowledge Argument for Correctness of a Shuffle". In: *EUROCRYPT 2008*. Ed. by Nigel P. Smart. Vol. 4965. LNCS. Springer, Heidelberg, Apr. 2008, pp. 379–396. DOI: 10.1007/978-3-540-78967-3_22.

[GPV08]   Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. "Trapdoors for hard lattices and new cryptographic constructions". In: *40th ACM STOC*. Ed. by Richard E. Ladner and Cynthia Dwork. ACM Press, May 2008, pp. 197–206. DOI: 10.1145/1374376.1374407.

[HHKKP22]   Charlotte Hoffmann, Pavel Hubáček, Chethan Kamath, Karen Klein, and Krzysztof Pietrzak. "Practical Statistically-Sound Proofs of Exponentiation in Any Group". In: *CRYPTO 2022, Part II*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13508. LNCS. Springer, Heidelberg, Aug. 2022, pp. 370–399. DOI: 10.1007/978-3-031-15979-4_13.

[Kar15]   Dmitrii Karp. *Normalized incomplete beta function: log-concavity in parameters and other properties*. 2015. DOI: 10.48550/ARXIV.1509.05120. URL: https://arxiv.org/abs/1509.05120.

[KI04]   Valentine Kabanets and Russell Impagliazzo. "Derandomizing Polynomial Identity Tests Means Proving Circuit Lower Bounds". In: *Comput. Complex.* 13.1-2 (2004), pp. 1–46. DOI: 10.1007/s00037-004-0182-6. URL: https://doi.org/10.1007/s00037-004-0182-6.

[Lin01]   Yehuda Lindell. "Parallel Coin-Tossing and Constant-Round Secure Two-Party Computation". In: *CRYPTO 2001*. Ed. by Joe Kilian. Vol. 2139. LNCS. Springer, Heidelberg, Aug. 2001, pp. 171–189. DOI: 10.1007/3-540-44647-8_10.

[LM19]   Russell W. F. Lai and Giulio Malavolta. "Subvector Commitments with Application to Succinct Arguments". In: *CRYPTO 2019, Part I*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Vol. 11692. LNCS. Springer, Heidelberg, Aug. 2019, pp. 530–560. DOI: 10.1007/978-3-030-26948-7_19.

[LNS21]   Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. "Shorter Lattice-Based Zero-Knowledge Proofs via One-Time Commitments". In: *PKC 2021, Part I*. Ed. by Juan Garay. Vol. 12710. LNCS. Springer, Heidelberg, May 2021, pp. 215–241. DOI: 10.1007/978-3-030-75245-3_9.

[LS15]   Adeline Langlois and Damien Stehlé. "Worst-case to average-case reductions for module lattices". In: *Des. Codes Cryptogr.* 75.3 (2015), pp. 565–599.

[MP13]   Daniele Micciancio and Chris Peikert. "Hardness of SIS and LWE with Small Parameters". In: *CRYPTO 2013, Part I*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8042. LNCS. Springer, Heidelberg, Aug. 2013, pp. 21–39. DOI: `10.1007/978-3-642-40041-4_2`.

[NS22]   Ngoc Khanh Nguyen and Gregor Seiler. "Practical Sublinear Proofs for R1CS from Lattices". In: *CRYPTO 2022, Part II*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13508. LNCS. Springer, Heidelberg, Aug. 2022, pp. 133–162. DOI: `10.1007/978-3-031-15979-4_5`.

[Pie19]   Krzysztof Pietrzak. "Proofs of Catalytic Space". In: *ITCS 2019*. Ed. by Avrim Blum. Vol. 124. LIPIcs, Jan. 2019, 59:1–59:25. DOI: `10.4230/LIPIcs.ITCS.2019.59`.

[RS62]   J Barkley Rosser and Lowell Schoenfeld. "Approximate formulas for some functions of prime numbers". In: *Illinois Journal of Mathematics* 6.1 (1962), pp. 64–94.

[RSA78]   Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". In: *Communications of the Association for Computing Machinery* 21.2 (1978), pp. 120–126.

[Sch80]   Jacob T Schwartz. "Fast probabilistic algorithms for verification of polynomial identities". In: *Journal of the ACM (JACM)* 27.4 (1980), pp. 701–717.

[Wes19]   Benjamin Wesolowski. "Efficient Verifiable Delay Functions". In: *EUROCRYPT 2019, Part III*. Ed. by Yuval Ishai and Vincent Rijmen. Vol. 11478. LNCS. Springer, Heidelberg, May 2019, pp. 379–407. DOI: `10.1007/978-3-030-17659-4_13`.

[Wik21]   Douglas Wikström. *Special Soundness in the Random Oracle Model*. Cryptology ePrint Archive, Report 2021/1265. `https://eprint.iacr.org/2021/1265`. 2021.

[WTsTW18] Riad S. Wahby, Ioanna Tzialla, abhi shelat, Justin Thaler, and Michael Walfish. "Doubly-Efficient zkSNARKS Without Trusted Setup". In: *2018 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2018, pp. 926–943. DOI: `10.1109/SP.2018.00060`.

[Zip79]   Richard Zippel. "Probabilistic algorithms for sparse polynomials". In: *International symposium on symbolic and algebraic manipulation*. Springer. 1979, pp. 216–226.

A   Proofs for the Multilinear Composite Schwartz Zippel Lemma

A.1   Proof of main LCSZ theorem (Theorem 1)

Theorem 1 (Multilinear Composite Schwartz-Zippel (LCSZ)). Let $N = \prod_{i=1}^{\ell} p_i^{r_i}$ for distinct primes $p_1, ..., p_\ell$. Let f be any μ-linear integer polynomial co-prime to N. For any integer $m > 1$ and $\mathbf{x}$ sampled uniformly from $[0, m)^\mu$:

$$\mathbb{P}_{\mathbf{x} \leftarrow [0,m)^\mu}[f(\mathbf{x}) \equiv 0 \bmod N] \leq \frac{\mu}{m} + \prod_{i=1}^{\ell} I_{\frac{1}{p_i}}(r_i, \mu),$$

where $I_{\frac{1}{p}}(r, \mu) = (1 - \frac{1}{p})^\mu \sum_{j=r}^{\infty} \binom{\mu+r-1}{r} \left(\frac{1}{p}\right)^j$ is the regularized beta function.

Proof. We begin by introducing some notations.

- For a polynomial $f \in \mathbb{Z}[X_1, ..., X_\mu]$ let $\vec{f}$ denote the coefficients of f and let $\mathsf{cont}(f)$ denote the greatest integer divisor of f, i.e. the *content*.

- $\vec{\beta} = (\beta_1, ..., \beta_\mu) \in [0, m)^\mu$ is a random variable distributed uniformly over $[0, m)^\mu$. For any $i \geq 1$, let $\vec{\beta}_i = (\beta_1, ..., \beta_i)$, let $f_0 = f$, and let $f_i(\vec{\beta}_i) := f(\beta_1, \ldots, \beta_i, X_{i+1}, \ldots, X_\mu)$.

- Given the random variable $\vec{\beta}$ distributed uniformly over $[0, m)^\mu$, define for each $j \in [1, \ell]$ and $i \in [1, \mu]$ the random variable $Y_{j,i}$ (as a function of $\vec{\beta}$) representing the multiplicity of p_j in $\mathsf{cont}(f_i(\vec{\beta}_i))$. Naturally, we set $Y_{j,0} = 0$ for all j because $\forall_j f_0 \not\equiv 0 \bmod p_j$. (**Note:** $Y_{j,i}$ are *not* independent). For any $i \in [\mu]$ the event

that $\forall_j Y_{j,i} \geq r_j$ is equivalent to the event that $f_i(\vec{\beta}_i) = 0 \bmod N$ and the event that $\forall_j Y_{j,i} = r_i$ is equivalent to $\mathsf{cont}(f_i(\vec{\beta}_i)) = N$. The event $\forall_j Y_{j,\mu} \geq r_j$ is thus equivalent to $f(\vec{\beta}) = 0 \bmod N$.

- Let $\{Z_{j,i}\}$ for $i \in [\mu]$ and $j \in [\ell]$ be a set of independent random variables, where $Z_{j,i}$ is geometric with parameter $1 - \frac{1}{p_j}$.

From the CCDF (complementary CDF) of geometric random variables we have that $P[Z_{j,i} \geq k] = (\frac{1}{p_j})^k$. Setting $Z_j := \sum_{i=1}^{\mu} Z_{j,i}$, from the CCDF of the negative binomial distribution (i.e., tail distribution of the sum of independent geometric random variables) it follows that $\forall_j P[Z_j \geq r] = I_{\frac{1}{p_j}}(r, \mu)$.

Next, we establish an important subclaim:

Claim 1. For any $i \geq 2$ and $\vec{k}, \vec{k}' \in \mathbb{N}^\ell$ where $\forall_j k_j \geq 0$:

$$P[\forall_j Y_{j,i} \geq k_j + k_j' | \forall_j Y_{j,i-1} = k_j'] \leq \frac{1}{m} + P[\forall_j Z_{j,i} \geq k_j].$$

Furthermore, for all $i \geq 1$, $P[\forall_j Y_{j,i} \geq Y_{j,i-1} + k_j] \leq \frac{1}{m} + P[\forall_j Z_{j,i} \geq k_j]$.

Proof. Since the order of the variables does not matter, w.l.o.g. assume that $\forall_{j \in [1,\ell']} k_j \geq 1$ and $\forall_{j > \ell'} k_j = 0$. Let $N^* = \prod_{j=1}^{\ell'} p_j^{k_j}$ and $N' = \prod_{j=1}^{\ell'} p_j^{k_j'}$. For any $i \geq 2$ and any $\mathbf{x} \in [0, m)^{i-1}$, in the event that $\vec{\beta}_{i-1} = \mathbf{x}$ and $\forall j\, Y_{j,i-1} = k_j'$, then by definition $f_{i-1}(\mathbf{x}) \equiv 0 \bmod N'$ and $\forall_j f_{i-1}(\mathbf{x})/N' \not\equiv 0 \bmod p_j$. In case $i = 1$, we have that $\forall_j Y_{j,0} = 0$, $N' = 1$, and $\forall_j f_0 = f \neq 0 \bmod p_j$. Thus, for all $i \geq 1$, conditioned on the events that $\vec{\beta}_{i-1} = \mathbf{x}$ and $\forall j\, Y_{j,i-1} = k_j'$, there exist multilinear $\mu - i$ variate polynomials h_i, g_i such that $f_{i-1}(\mathbf{x})/N' = h_i(X_{i+1}, \ldots, X_\mu) + X_i \cdot g_i(X_{i+1}, \ldots, X_\mu)$ where for all j at least one of h_i or g_i is nonzero modulo p_j.

Furthermore, for any $i \geq 1$, conditioned on the events $\forall j\, Y_{j,i-1} = k_j'$ and $\vec{\beta}_{i-1} = \mathbf{x}$, the event that $\forall_j Y_{j,i} \geq k_j + k_j'$ is equivalent to the event that $h_i + \beta_i g_i \equiv 0 \bmod N^*$.

For each index $t \in [1, 2^{\mu-i}]$ let $h_i[t]$ and $g_i[t]$ denote the tth coefficients of h_i and g_i respectively (i.e., the coefficients on the tth monomial in a canonical ordering of the $2^{\mu-i}$ monomials over the $\mu - i$ variables $X_{i+1}, ..., X_\mu$). Given that for every $j \in [\ell']$ the polynomial $h_i + X_i \cdot g_i$ is non-zero modulo p_j, for each $j \in [\ell']$ there exists at least one index t_j for which the univariate linear polynomial $h_i[t_j] + X_i g_i[t_j]$ is non-zero modulo p_j. We now have that:

$$P[\forall_j Y_{j,i} \geq k_j + k_j' | \vec{\beta}_{i-1} = \mathbf{x} \wedge \forall_j Y_{j,i-1} = k_j'] \tag{1}$$

$$= P[\forall_j\ h_i + \beta_i \cdot g_i \equiv 0 \bmod p_j^{k_j}] \tag{2}$$

$$\leq P[\forall_j\ h_i[t_j] + \beta_i \cdot g_i[t_j] \equiv 0 \bmod p_j^{k_j}]. \tag{3}$$

For each t_j there is *at most* one solution to the equation $h_i[t_j] + X_i \cdot g_i[t_j] \equiv 0 \bmod p_j$. Consequently, by CRT, there is at most one integer solution $x^* \in [0, N^*)$ to the system of equations $\forall_j h_i[t_j] + X_i \cdot g_i[t_j] \equiv 0 \bmod\ p_j$[10]. If x^* was uniform mod N^* then the probability would be $\frac{1}{N^*}$. However, we must also consider the case that x^* is not uniform mod N^*.

Let E denote the event that the system of equations $\forall_j\ h_i[t_j] + \beta_i \cdot g_i[t_j] \equiv 0 \bmod p_j^{k_j}$ from equation (3) is satisfied. There is at most one integer equivalence class modulo N^* satisfying this system of equations and the random variable β_i is uniformly distributed over $[0, m)$. Let U denote the event that $\beta_i \in [0, m - m \bmod N^*)$ and let $\bar{U}$ denote the event that $\beta_i \in [m - m \bmod N^*, m)$. Conditioned on U, β_i is uniformly distributed modulo N^*, and thus $P[E|U] \leq 1/N^*$. Conditioned on $\bar{U}$, β_i is uniformly distributed over the set $[m - m \bmod N^*, m)$. As this set is a contiguous interval of less than N^* integers it contains at most one solution to the systems of equations, and thus, $P[E|\bar{U}] \leq 1/(m \bmod N^*)$. The probability of $\bar{U}$ is exactly $\frac{m \bmod N^*}{m}$. Therefore:

$$P[E] = P[E|U] \cdot P[U] + P[E|\bar{U}] \cdot P[\bar{U}] \tag{4}$$

$$\leq \frac{1}{N^*} \cdot (1 - \frac{m \bmod N^*}{m}) + \frac{1}{m \bmod N^*} \cdot \frac{m \bmod N^*}{m} \tag{5}$$

$$\leq \frac{1}{N^*} + \frac{1}{m}. \tag{6}$$

---

[10]This is where the proof falls apart for higher degree polynomials. For example if the polynomial is quadratic in each variable then there could be up to 2^ℓ solutions where ℓ is the number of prime factors.

We also have that:

$$P[\forall_{j\in[\ell]}Z_{j,i} \geq k_j] = \prod_{j=1}^{\ell'} P[Z_{j,i} \geq k_j] = \prod_{j=1}^{\ell'} \left(\frac{1}{p_j}\right)^{k_j} = \frac{1}{N^*}.$$

Thus, putting it all together, for any $i \geq 1$ and any $\mathbf{x} \in [0,m)^{i-1}$:

$$P[\forall_j Y_{j,i} \geq k_j + k_j' | \vec{\beta}_{i-1} = \mathbf{x} \wedge \forall_j Y_{j,i-1} = k_j'] \leq \frac{1}{m} + \frac{1}{N^*} = \frac{1}{m} + P[\forall_j Z_{j,i} \geq k_j].$$

Since the probability bound is independent of $\mathbf{x}$, this implies:

$$P[\forall_j Y_{j,i} \geq k_j + k_j' | \forall_j Y_{j,i-1} = k_j'] \leq \frac{1}{m} + P[\forall_j Z_{j,i} \geq k_j].$$

Consequently, for any $i \geq 2$ and $\vec{k} \in \mathbb{N}^\ell$ where $\forall_j k_j > 0$:

$$P[\forall_j Y_{j,i} \geq Y_{j,i-1} + k_j] \leq \max_{\vec{k'}} P[\forall_j Y_{j,i} \geq k_j + k_j' | \forall_j Y_{j,i-1} = k_j'] \leq \frac{1}{m} + P[\forall_j Z_{j,i} \geq k_j].$$

Similarly, for $i = 1$ and $\vec{k} \in \mathbb{N}^\ell$ where $\forall_j k_j > 0$:

$$P[\forall_j Y_{j,i} \geq k_j] \leq \frac{1}{m} + P[\forall_j Z_{j,i} \geq k_j].$$

$\square$

We now prove the full theorem by induction over $i \in [1,\mu]$. Specifically, we will prove the following inductive hypothesis for $i \in [1,\mu]$:

$$P[f_i(\vec{\beta}_i) = 0 \bmod N] = P[\forall_j Y_{j,i} \geq r_j] \leq \frac{i}{m} + \prod_{j=1}^{\ell} P[\sum_{k=1}^{i} Z_{j,k} \geq r_j].$$

Setting $i = \mu$ this is equivalent to the theorem statement.
Base Case: The base case follows directly from the subclaim for the case $i = 1$.

$$P[\forall_j Y_{j,1} \geq r_j] \leq \frac{1}{m} + P[\forall_j Z_{j,1} \geq r_j] = \frac{1}{m} + \prod_{j=1}^{\ell} P[Z_{j,1} \geq r_j].$$

Induction Step: Assume the inductive hypothesis holds for some $1 \leq i < \mu$, i.e.:

$$P[\forall_j Y_{j,i} \geq r_j] \leq \frac{i}{m} + \prod_{j=1}^{\ell} P[\sum_{k=1}^{i} Z_{j,k} \geq r_j].$$

We will show this implies the hypothesis holds for $i+1$:

$$P[\forall_j Y_{j,i+1} \geq r_j] = \sum_{\vec{k}\in\mathbb{N}^\ell} P[\forall_j Y_{j,i+1} - Y_{j,i} \geq r_j - k_j | \forall_j Y_{j,i} = k_j] \cdot P[\forall_j Y_{j,i} = k_j]$$

$$\leq \sum_{\vec{k}} (P[\forall_j Z_{j,i+1} \geq r_j - k_j] + \frac{1}{m}) \cdot P[\forall_j Y_{j,i} = k_j] \quad \textbf{(by subclaim)}$$

$$= \frac{1}{m} + \sum_{\vec{k}} P[\forall_j Z_{j,i+1} \geq r_j - k_j] \cdot P[\forall_j Y_{j,i} = k_j]$$

$$= \frac{1}{m} + \sum_{\vec{\Delta}\in\mathbb{Z}^\ell : \forall_j \Delta_j \leq r_j} P[\forall_j Z_{j,i+1} \geq \Delta_j] \cdot P[\forall_j Y_{j,i} = r_j - \Delta_j] \quad \textbf{(change of variables)}$$

$$= \frac{1}{m} + \sum_{\vec{\Delta}\in\mathbb{Z}^\ell : \forall_j \Delta_j \leq r_j} \sum_{\vec{k}\in\mathbb{N}^\ell} P[\forall_j Z_{j,i+1} = \Delta_j + k_j] \cdot P[\forall_j Y_{j,i} = r_j - \Delta_j]$$

$$= \frac{1}{m} + \sum_{\vec{\Delta'}\in\mathbb{N}^\ell} P[\forall_j Z_{j,i+1} = \Delta_j'] \cdot \sum_{\vec{k}\in\mathbb{N}^\ell} P[\forall_j Y_{j,i} = r_j - (\Delta_j' - k_j)] \quad \textbf{(c.o.v.)}$$

$$= \frac{1}{m} + \sum_{\vec{\Delta'}\in\mathbb{N}^\ell} P[\forall_j Z_{j,i+1} = \Delta_j'] \cdot P[\forall_j Y_{j,i} \geq r_j - \Delta_j']$$

$$\leq \frac{1}{m} + \sum_{\vec{\Delta'}} P[\forall_j Z_{j,i+1} = \Delta_j'] \cdot (P[\forall_j \sum_{i'=1}^{i} Z_{j,i'} \geq r_j - \Delta_j'] + \frac{i}{m}) \quad \textbf{(inductive hyp.)}$$

$$= \frac{i+1}{m} + \sum_{\vec{\Delta'}} P[\forall_j Z_{j,i+1} = \Delta_j'] \cdot P[\forall_j \sum_{k=1}^{i} Z_{j,k} \geq r_j - \Delta_j']$$

$$= \frac{i+1}{m} + P[\forall_j \sum_{k=1}^{i} Z_{j,k} \geq r_j - Z_{j,i+1}] \quad \textbf{(independence of variables)}$$

$$= \frac{i+1}{m} + \prod_{j=1}^{\ell} P[\sum_{k=1}^{i+1} Z_{j,k} \geq k_j]. \quad \textbf{(independence of variables)}$$

$\square$

A.2 Bounds on the regularized beta function

We provide useful bounds of the regularized beta function that are necessary for the proofs of Theorems 2 and 3.

The regularized incomplete beta function is defined for $k, \mu \in \mathbb{N}$ as:

$$I_\epsilon(k, \mu) = (1 - \epsilon)^\mu \sum_{j=k}^{\infty} \binom{\mu + j - 1}{j} \epsilon^j.$$

Special values are $I_\epsilon(k, 1) = \epsilon^k$, which matches the tail distribution of a geometric variable with parameter $1 - \epsilon$, and $I_\epsilon(1, \mu) = 1 - (1 - \epsilon)^\mu$, which is the probability that at least one of μ geometric variables of parameter $1 - \epsilon$ is positive.

Lemma 5. $I_\epsilon(k, \mu) \leq (\epsilon\mu)^k$ for all $\mu, k \in \mathbb{N}$ and $\epsilon \in (0, 1)$ where $\epsilon\mu \leq 1/2$.

Proof. For $k = 0$ the statement holds because $I_\epsilon(0, \mu) = 1$. For $\mu = 1$ we have $I_\epsilon(k, 1) = \epsilon^k$. It remains to prove the inequality for $\mu \geq 2$ and $k \geq 1$. We will use the following ordinary generating function identity for binomial coefficients: $\sum_{j=0}^{\infty} \binom{a+j}{a} x^j = \frac{1}{(1-x)^{a+1}}$.

This allows us to write $I_\epsilon(k, \mu)$ as:

$$I_\epsilon(k, \mu) = (1 - \epsilon)^\mu \cdot \left(\sum_{j=0}^{\infty} \binom{\mu + j - 1}{j} \epsilon^j - \sum_{j=0}^{k-1} \binom{\mu + j - 1}{j} \epsilon^j \right)$$

$$= (1 - \epsilon)^\mu \cdot \left(\frac{1}{(1 - \epsilon)^\mu} - \sum_{j=0}^{k-1} \binom{\mu + j - 1}{j} \epsilon^j \right) = 1 - (1 - \epsilon)^\mu \sum_{j=0}^{k-1} \binom{\mu + j - 1}{j} \epsilon^j$$

Using the geometric series identity $(\epsilon\mu)^k = 1 - (1 - \epsilon\mu) \cdot \sum_{j=0}^{k-1} (\epsilon\mu)^j$, we obtain:

$$(\epsilon\mu)^k - I_\epsilon(k, \mu) = \sum_{j=0}^{k-1} (1 - \epsilon)^\mu \cdot \binom{\mu + j - 1}{j} \epsilon^j - (1 - \epsilon\mu)(\epsilon\mu)^j$$

$$= \sum_{j=0}^{k-1} \left((1 - \epsilon)^\mu \cdot \binom{\mu + j - 1}{j} \right) - (1 - \epsilon\mu)\mu^j \right) \epsilon^j.$$

Let $\Phi_{\epsilon,\mu}(k) = (\epsilon\mu)^k - I_\epsilon(k, \mu)$ so that the goal is to show $\Phi_{\epsilon,\mu}(k) \geq 0$ for $k, \mu \in \mathbb{N}$ where $\mu \geq 2$ and $\epsilon\mu \leq 1/2$. Observe that:

$$\Phi_{\epsilon,\mu}(1) = \epsilon\mu - 1 + (1 - \epsilon)^\mu = (1 - \epsilon)^\mu - (1 - \epsilon\mu) \geq 0.$$

$$\lim_{k \to \infty} \Phi_{\epsilon,\mu}(k) = 0.$$

Thus, it suffices to show that $\Phi_{\epsilon,\mu}(k)$ is non-increasing on the interval $k \in [2, \infty)$ when $\epsilon\mu \leq 1/2$ as this implies that $\Phi_{\epsilon,\mu}(k) \geq 0$ for all $k \geq 1$, $\mu \geq 2$, and $\epsilon\mu \leq 1/2$. Moreover, we can easily show this by showing that for all $\mu, j \geq 2$ and $\epsilon\mu \leq 1/2$:

$$(1 - \epsilon)^\mu \cdot \binom{\mu + j - 1}{j} \leq (1 - \epsilon\mu) \cdot \mu^j.$$

Letting $R(\mu, j) = \frac{\binom{\mu + j - 1}{j}}{\mu^j}$, observe that:

$$R(\mu, j) = \frac{\binom{\mu + j - 1}{j}}{\mu^j} = \frac{\prod_{i=0}^{j-1} (\mu + i)}{j! \, \mu^j} = \prod_{i=0}^{j-1} \frac{\mu + i}{\mu \cdot (i + 1)}.$$

Since $\mu + i \leq \mu \cdot (i + 1)$ for all $\mu \geq 2$ and $i \geq 0$, it follows that $R(\mu, j) \leq R(\mu, 2) = \frac{1}{2} \cdot (1 + \frac{1}{\mu})$ for $j \geq 2$. Furthermore, for $\epsilon \in (0, 1/\mu)$:

$$\frac{d}{d\epsilon} \frac{(1 - \epsilon)^\mu}{1 - \epsilon\mu} = \frac{(\mu - 1)\epsilon\mu(1 - \epsilon)^{\mu-1}}{(1 - \epsilon\mu)^2} \geq 0.$$

Thus, for $\epsilon \in (0, \frac{1}{2\mu}]$ and $\mu \geq 2$:

$$\frac{(1 - \epsilon)^\mu}{(1 - \epsilon\mu)} \cdot R(\mu, j) \leq \frac{(1 - \frac{1}{2\mu})^\mu}{1/2} \cdot R(\mu, 2) \leq \frac{(1 + \frac{1}{\mu})}{\sqrt{e}} \leq e^{\frac{1}{\mu} - 1/2} \leq 1.$$

This completes the proof. $\qquad\square$

25

Lemma 6. $I_\epsilon(k, \mu) \leq \epsilon^k \cdot k^\mu$ for $k \geq 2\mu$ and $\epsilon \leq 1/2$.

This is tighter than Bound 1 for larger k, i.e. when $k^\mu < \mu^k$.

Proof. Similar to the analysis in Bound 1, define $\Psi_{\epsilon,\mu}(k) = \epsilon^k k^\mu - I_\epsilon(k, \mu)$ so that:

$$\Psi_{\epsilon,\mu}(k) = \epsilon^k k^\mu - 1 + (1 - \epsilon)^\mu \sum_{j=0}^{k-1} \binom{\mu + j - 1}{j} \epsilon^j.$$

Bound 2 holds iff $\Psi_{\epsilon,\mu}(k) \geq 0$ for all $k \geq 2\mu$ and $\epsilon \leq 1/2$. For $\mu = 1$ we have $I_\epsilon(k, 1) = \epsilon^k$ so

$$\Psi_{\epsilon,1}(k) = \epsilon^k \cdot k - \epsilon^k \geq 0.$$

Furthermore, $\lim_{k\to\infty} \Psi_{\epsilon,\mu}(k) = 0$. Thus, it suffices to show that $\Psi_{\epsilon,\mu}$ is non-increasing on the interval $[2\mu, \infty)$ for $\mu \geq 2$ and $\epsilon \leq 1/2$.

Observe that:

$$\Psi_{\epsilon,\mu}(k + 1) - \Psi_{\epsilon,\mu}(k) = \epsilon^{k+1}(k + 1)^\mu - \epsilon^k k^\mu + (1 - \epsilon)^\mu \binom{\mu + k - 1}{k} \epsilon^k$$

$$= \epsilon^k \left[\epsilon(k + 1)^\mu - k^\mu + (1 - \epsilon)^\mu \binom{\mu + k - 1}{k} \right].$$

Defining:

$$\Delta_\epsilon(k, \mu) := (1 - \epsilon)^\mu \frac{\binom{\mu+k-1}{k}}{k^\mu} + \epsilon(1 + \frac{1}{k})^\mu.$$

$\Psi_{\epsilon,\mu}(k)$ is non-increasing on $k \in [2\mu, \infty]$ iff $\Delta_\epsilon(k, \mu) \leq 1$ for all $k \geq 2\mu$. We will prove this for $\mu \geq 2$ and $\epsilon \leq 1/2$. Using the inequality $(1 + \frac{1}{k})^\mu \leq \sqrt{e}$ for $k \geq 2\mu$:

$$\Delta_\epsilon(2\mu, \mu) \leq (1 - \epsilon)^\mu \frac{\binom{3n-1}{2\mu}}{(2\mu)^\mu} + \epsilon\sqrt{e}.$$

The right hand side is decreasing as $\mu \to \infty$ and thus for $\mu \geq 2$ and $\epsilon \leq 1/2$:

$$\Delta_\epsilon(2\mu, \mu) \leq (1 - \epsilon)^2 \cdot \frac{\binom{5}{4}}{4^2} + \epsilon\sqrt{e} = (1 - \epsilon)^2 \cdot \frac{5}{16} + \epsilon\sqrt{e} < 1.$$

To see why this is less than 1 for $\epsilon \leq 1/2$, note that $\frac{d}{d\epsilon}(1-\epsilon)^2 \cdot c_1 + \epsilon \cdot c_2 = 2c_1\epsilon + c_2 - 2c_1$ is positive when $\epsilon \geq 0$ and $c_2 \geq 2c_1$, and $\sqrt{e} > \frac{5}{8}$. Thus, on the interval $\epsilon \in [0, \frac{1}{2}]$, $(1 - \epsilon)^2 \frac{5}{16} + \epsilon\sqrt{e} \leq \frac{1}{4} \cdot \frac{5}{16} + \frac{1}{2}\sqrt{e} = 0.902...$

Finally, since $\Delta_\epsilon(k, \mu)$ is decreasing as $k \to \infty$:

$$\forall_{k \geq 2\mu, \mu \geq 2, \epsilon \leq 1/2} \; \Delta_\epsilon(k, \mu) \leq \Delta_\epsilon(2\mu, \mu) < 1.$$

$\square$

Corollary 6. For any prime p and any positive integer μ

$$P[\sum_{i=1}^{\mu} X_i \geq r] = I_{\frac{1}{p}}(r, \mu) \leq \begin{cases} \frac{r^\mu}{p^r} & \text{if } r \geq 2\mu \\ (\frac{\mu}{p})^r & \text{if } p \geq 2\mu \\ 1 & \text{otherwise} \end{cases},$$

where X_i are independent geometric variables with parameter $(\frac{1}{p})$ and $P[X_i \geq r] = \left(\frac{1}{p}\right)^r$.

A.3 Proof of Inverse LCSZ (Theorem 2)

Theorem 2 (Inverse LCSZ)**.** For all $\mu \geq 2$, $\epsilon \geq \log_\mu(2)$, and all N such that

$$\log N \geq 4\mu^{2+\epsilon} + (1 + \frac{1}{\epsilon}) \cdot \lambda.$$

we have that for any μ-linear polynomial f that is coprime with N

$$\mathbb{P}_{x \leftarrow [0,m)^\mu}[f(x) \equiv 0 \bmod N] \leq 2^{-\lambda} + \frac{\mu}{m}.$$

26

By Theorem 1 (CSZ) we have that for $N = \prod_i p_i^{r_i}$:

$$\mathbb{P}_{\mathbf{x} \leftarrow [0,m)^\mu}[f(\mathbf{x}) \equiv 0 \bmod N] \leq \prod_i I_{\frac{1}{p_i}}(r_i, \mu) + \frac{\mu}{m}.$$

For $\mu = 1$ and $\log_2(N) \geq \lambda$, Theorem 1 shows that $\mathbb{P}_{x \leftarrow [0,m)}[f(x) \equiv 0 \bmod N] \leq 2^{-\lambda} + \frac{1}{m}$. This is derived by substituting $I_{1/p}(r, 1) = \frac{1}{p^r}$, which gives $\mathbb{P}_{x \leftarrow [0,m)}[f(x) \equiv 0 \bmod N] \leq \frac{1}{N} + \frac{1}{m}$. The case $\mu \geq 2$ is more complicated. This is the focus of the rest of the proof..

For a given $N \in \mathbb{N}$, let $S(N)$ denote the set of pairs (p, r) where p is a prime divisor of N and r is its multiplicity, i.e. $N = \prod_{(p,r) \in S} p^r$.

It follows from Theorem 1 (CSZ) that if $N \geq t(\lambda, \mu) = \sup_{N \in \mathbb{N}}\{\prod_{(p,r) \in S(N)} I_{\frac{1}{p}}(r, \mu) \geq 2^{-\lambda}\}$ then

$$\mathbb{P}_{\mathbf{x} \leftarrow [0,m)^\mu}[f(\mathbf{x}) \equiv 0 \bmod N] \leq 2^{-\lambda} + \frac{\mu}{m}.$$

Assuming $t(\lambda, \mu) < \infty$, we obtain the following constrained maximization problem:

$$\log_2 t(\lambda, \mu) := \max_{N \in \mathbb{N}} \log_2 N$$
$$\text{subject to} \sum_{(p,r) \in S(N)} -\log_2 I_{\frac{1}{p}}(r, \mu) \leq \lambda. \qquad \text{(Constrained Max 1)}$$

In order to derive an upper bound on $\log_2 t(\lambda, \mu)$, we construct a sequence of modified maximization problems, each of which is an upper bound to the prior. The last in this sequence is a knapsack problem for which we analytically derive an upper bound.

Definition 6. Let $\mathsf{val}(p, r) := r \log_2 p$ and let $\mathsf{weight}_\mu(p, r) = -\log_2 I_{\frac{1}{p}}(r, \mu)$. Additionally, for all $\epsilon \geq \log_\mu(2)$, let:

$$\mathsf{weight}_{\mu,\epsilon}(p, r) := \begin{cases} r \cdot (\log_2(p) - \log_2(\mu)) & \text{if } p \geq \mu^{1+\epsilon} \\ r \cdot \log_2(p) - \mu \cdot \log_2(r) & \text{if } p < \mu^{1+\epsilon} \wedge r > 2(1+\epsilon)\frac{\mu \ln \mu}{\ln p} \\ 0 & \text{otherwise} \end{cases} .$$

Claim 2. For any prime p and $r \in \mathbb{N}$, if $\epsilon \geq \log_\mu(2)$ then $\mathsf{weight}_\mu(p, r) \leq \mathsf{weight}_{\mu,\epsilon}(p, r)$.

Proof. If $\epsilon \geq \log_\mu(2)$ then $\mu^{1+\epsilon} \geq 2\mu$ and the claim follows from Corollary 6 (to Lemma 5 and Lemma 6). □

Claim 3. For any prime p, $\mathsf{weight}_{\mu,\epsilon}(p, r)$ is non-decreasing over $r \geq 1$ and increasing for $r > 2(1+\epsilon)\frac{\mu \log \mu}{\log p}$.

Proof. We first show that the function $\mathsf{weight}_{\mu,\epsilon}(p, r)$ is non-decreasing in r in each of the three cases, which comprise three subdivisions of the plane $\mathsf{Primes} \times \mathbb{N}$, which we denote S_A, S_B, and S_C respectively. S_A contains all (p, r) where $p \geq \mu^{1+\epsilon}$, in which case $\frac{d}{dr}\mathsf{weight}_{\mu,\epsilon}(p, r) = \log_2(p) - \log_2(\mu) > 0$. S_B contains all (p, r) where $r > 2(1+\epsilon)\frac{\mu \ln \mu}{\ln p}$ and $p < \mu^{1+\epsilon}$, in which case $\frac{d}{dr}\mathsf{weight}_{\mu,\epsilon}(p, r) = \log_2(p) - \frac{\mu}{r \ln 2}$, and $\log_2(p) - \frac{\mu}{r \ln 2} \geq \log_2 p - \frac{1}{2 \ln 2} > 0$. The weight function is constant at 0 for all remaining pairs, which comprise set S_C.

It remains to show that $\mathsf{weight}_{\mu,\epsilon}(p, r)$ increases in r across the boundary between S_B and S_C, for which it suffices to show that the weight is positive for all $(p, r) \in S_B$. Suppose, towards contradiction, that $r \log_2 p \leq \mu \log_2 r$ and $r > 2(1+\epsilon)\frac{\mu \log_2 \mu}{\log_2 p}$. This would imply that both $\frac{r}{\log_2 r} \leq \frac{\mu}{\log_2 p}$ and $\frac{r}{2(1+\epsilon) \log \mu} > \frac{\mu}{\log_2 p}$, which implies that $\log_2 r > 2(1+\epsilon) \log \mu$. Since $\frac{r}{\log_2 r}$ is monotonic increasing in r, this in turn implies that $\frac{r}{\log_2 r} > \frac{\mu^2}{4 \log_2 \mu} \geq \mu$ for all $\mu \geq 1$. Finally, the implication that $\frac{r}{\log_2 r} > \mu$ contradicts the assumption that $r \log_2 p \leq \mu \log_2 r$. □

The first modified maximization problem is:

$$\max_{N \in \mathbb{N}} \sum_{(p,r) \in S(N)} \mathsf{val}(p, r)$$
$$\text{subject to} \sum_{(p,r) \in S(N)} \mathsf{weight}_{\mu,\epsilon}(p, r) \leq \lambda. \qquad \text{(Constrained Max 2)}$$

27

Claim 4. Eq. (Constrained Max 2) is an upper bound to Eq. (Constrained Max 1) for any μ and $\epsilon \geq \log_\mu(2)$:

Proof. For any $N \in \mathbb{N}$, by definition $\sum_{(p,r) \in S(N)} \mathsf{val}(p,r) = \log_2 N$. Thus the only difference between the two maximization problems are the constraints. Furthermore, if $\epsilon \geq \log_\mu(2)$ then, by Claim 2, Eq. (Constrained Max 2) is simply a relaxation of the constraints in Eq. (Constrained Max 1). $\qquad \square$

Let Primes denote the infinite set of prime numbers.

Definition 7 (Marginal Value/Weight). Using $\mathsf{val}(p,r)$ and $\mathsf{weight}(p,r)$ as defined in Definition 6, $r \geq 1$ and $p \in \mathsf{Primes}$, let $\Delta\mathsf{val}(p,r) := \mathsf{val}(p,r) - \mathsf{val}(p, r-1) = \log_2 p$ and $\Delta\mathsf{val}(p,0) := \mathsf{val}(p,0) = 0$. Similarly, for $r \geq 1$ let $\Delta\mathsf{weight}_{\mu,\epsilon}(p,r) := \mathsf{weight}_{\mu,\epsilon}(p,r) - \mathsf{weight}_{\mu,\epsilon}(p, r-1)$ and $\Delta\mathsf{weight}_{\mu,\epsilon}(p,0) := \mathsf{weight}_{\mu,\epsilon}(p,0) = 1$.

By Claim 3, $\Delta\mathsf{weight}_{\mu,\epsilon}(p,r)$ is non-negative for all prime p and $r \in \mathbb{N}$, and positive for $r > 2(1 + \epsilon)\frac{\mu \log \mu}{\log p}$. The following knapsack problem gives an upper bound to Eq. (Constrained Max 2) :

$$
\begin{aligned}
& \max_{S \subseteq \mathsf{Primes} \times \mathbb{N}} \sum_{(p,r) \in S} \Delta\mathsf{val}(p,r) \\
& \text{subject to } \sum_{(p,r) \in S} \Delta\mathsf{weight}_{\mu,\epsilon}(p,r) \leq \lambda.
\end{aligned}
\qquad \text{(Knapsack Problem)}
$$

Claim 5. Eq. (Knapsack Problem) is an upper bound to Eq. (Constrained Max 2).

Proof. Le S^* denote argmax of Eq. (Knapsack Problem) with $v^* = \sum_{(p,r) \in S^*} \Delta\mathsf{val}(p,r)$ and suppose (towards contradiction) that there exists $N \in \mathbb{N}$ such that

$$
\sum_{(p,r) \in S(N)} \mathsf{weight}_{\mu,\epsilon}(p,r) \leq \lambda
$$

and

$$
\sum_{(p,r) \in S(N)} \mathsf{val}(p,r) > v^*.
$$

Consider the set S', which includes $S(N)$ and adds for each $(p,r) \in S(N)$ the pairs (p,j) for each $j \leq r$, i.e:

$$
S' = \bigcup_{(p,r) \in S(N)} \bigcup_{j=0}^{r} \{(p,j)\}.
$$

Observe that:

$$
\sum_{(p,r) \in S'} \Delta\mathsf{val}(p,r) = \sum_{(p,r) \in S(N)} \sum_{j=0}^{r} \Delta\mathsf{val}(p,j) = \sum_{(p,r) \in S(N)} \mathsf{val}(p,r) > v^*.
$$

and

$$
\sum_{(p,r) \in S'} \Delta\mathsf{weight}_{\mu,\epsilon}(p,r) = \sum_{(p,r) \in S(N)} \sum_{j=0}^{r} \Delta\mathsf{weight}_{\mu,\epsilon}(p,j) = \sum_{(p,r) \in S(N)} \mathsf{weight}_{\mu,\epsilon}(p,r) \leq \lambda.
$$

This is a contradiction to the assumption that v^* is the solution to Eq. (Knapsack Problem). $\qquad \square$

Finally, we prove a series of claims that will enable us to derive an upper bound on Eq. (Knapsack Problem). First, we define:

Definition 8 (Density). $\mathsf{density}_{\mu,\epsilon}(p,r) = \frac{\Delta\mathsf{val}(p,r)}{\Delta\mathsf{weight}_{\mu,\epsilon}(p,r)}$ where $\Delta\mathsf{val}(p,r)$ and $\Delta\mathsf{weight}(p,r)$ are defined in Definition 7.

Claim 6. For all $S \subseteq \mathsf{Primes} \times \mathbb{N}$:

$$
\sum_{(p,r) \in S} \Delta\mathsf{val}(p,r) \leq \sum_{(p,r) \in S} \Delta\mathsf{weight}(p,r) \cdot \max_{(p,r) \in S} \{\mathsf{density}_{\mu,\epsilon}(p,r)\}.
$$

28

Proof.

$$\sum_{(p,r)\in S} \Delta\mathsf{weight}(p,r) \cdot \max_{(p,r)\in S}\{\mathsf{density}_{\mu,\epsilon}(p,r)\}$$

$$\geq \sum_{(p,r)\in S} \mathsf{weight}_{\mu}(p,r) \cdot \mathsf{density}_{\mu,\epsilon}(p,r)$$

$$= \sum_{(p,r)\in S} \mathsf{val}(p,r).$$

$\square$

Claim 7. If $\mu \geq 2$, $\epsilon \geq \log_{\mu}(2)$, $r \geq 1$, and $p \geq \mu^{1+\epsilon}$ then $\mathsf{density}_{\mu,\epsilon}(p,r) \leq 1 + \frac{1}{\epsilon}$

Proof. If $\epsilon \geq \log_{\mu}(2)$ and $p \geq \mu^{1+\epsilon}$ then $p \geq 2\mu$, and thus:

$$\mathsf{density}_{\mu,\epsilon}(p,r) = \frac{\Delta\mathsf{val}(p,r)}{\Delta\mathsf{weight}_{\mu,\epsilon}(p,r)} = \frac{\log p}{\log_2(p) - \log_2(\mu)}.$$

Furthermore, $p \geq \mu^{1+\epsilon}$ implies that:

$$\log_2(p) - \log_2(\mu) \geq \log_2(p) - \frac{1}{1+\epsilon}\log_2(p) = \log_2(p) \cdot \frac{\epsilon}{1+\epsilon}.$$

Thus $\mathsf{density}_{\mu,\epsilon}(p,r) \leq \frac{\log_2(p)}{\log_2(p)\cdot\frac{\epsilon}{1+\epsilon}} = 1 + \frac{1}{\epsilon}$. $\square$

Claim 8. If $\mu \geq 2$, $\epsilon \geq \log_{\mu}(2)$, $r > 2(1+\epsilon) \cdot \frac{\mu\cdot\ln\mu}{\ln p}$, and $p < \mu^{1+\epsilon}$ then $\mathsf{density}_{\mu,\epsilon}(p,r) \leq 1 + \frac{1}{\epsilon}$

Proof. Since $p < \mu^{1+\epsilon}$, the conditions on r imply $r > 2(1+\epsilon)\frac{\mu\ln\mu}{\ln p} > 2\mu$ and thus:

$$\mathsf{density}_{\mu,\epsilon}(p,r) = \frac{\Delta\mathsf{val}(p,r)}{\Delta\mathsf{weight}_{\mu,\epsilon}(p,r)} = \frac{\ln p}{\ln p + \mu\ln\frac{r-1}{r}} = \frac{1}{1 - \frac{\mu}{\ln p}\ln\frac{r}{r-1}}.$$

$\mathsf{density}_{\mu,\epsilon}(p,r)$ is non-negative because $\Delta\mathsf{val}(p,r)$ is non-negative and $\Delta\mathsf{weight}(p,r)$ is positive over $r > 2(1+\epsilon)\frac{\mu\log\mu}{\log p}$. Thus, for p and r satisfying these conditions, it must be the case that $0 \leq \frac{\mu}{\ln p}\ln\frac{r}{r-1} < 1$. Furthermore, this term is decreasing (approaching zero) as r increases, which shows that for r and p subject to these conditions $\mathsf{density}_{\mu,\epsilon}(p,r)$ is also decreasing in r. Combining this with the fact that $\ln\frac{r}{r-1} = \ln(1 + \frac{1}{r-1}) \leq \frac{1}{r-1}$:

$$\mathsf{density}_{\mu,\epsilon}(p,r) \leq \mathsf{density}_{\mu,\epsilon}(p, 2(1+\epsilon)\frac{\mu\ln\mu}{\ln p}+1) \leq \frac{1}{1 - \frac{\mu}{\ln p}\frac{\ln p}{2(1+\epsilon)\mu\ln\mu}} = \frac{1}{1 - \frac{1}{2(1+\epsilon)\ln\mu}} \leq 1 + \frac{1}{\epsilon}.$$

$\square$

Claim 9. For $\alpha \in \mathbb{R}$ let $\mathsf{Primes}(\alpha)$ denote the set of prime numbers strictly less than α. For $\mu \in \mathbb{N}$ and $\epsilon \in (0,1)$ define:

$$B_{\mu,\epsilon} := \{(p,r) : p \in \mathsf{Primes}(\mu^{1+\epsilon}), r \leq 2(1+\epsilon)\frac{\mu\ln\mu}{\ln p}\}.$$

Then :

$$\sum_{(p,r)\in B_{\mu,\epsilon}} \Delta\mathsf{val}(p,r) \leq 4\mu^{2+\epsilon}.$$

Proof. Let $\pi(X)$ denote the prime counting function. We use the fact that $\pi(x) \leq 1.3 \cdot \frac{x}{\ln(x)}$ for all $x > 1$ [RS62].

$$\sum_{(p,r)\in B_{\mu,\epsilon}} \Delta\mathsf{val}(p,r) = \sum_{p\in\mathsf{Primes}(\mu^{1+\epsilon})} \sum_{r=0}^{\lfloor 2(1+\epsilon)\frac{\mu\ln\mu}{\ln p}\rfloor} \log p = \sum_{\mathsf{Primes}(\mu^{1+\epsilon})} 2(1+\epsilon)\mu\log_2\mu$$

$$\leq 1.3 \cdot \frac{\mu^{1+\epsilon}}{\ln(\mu^{1+\epsilon})} \cdot \frac{2\cdot(1+\epsilon)\mu\ln(\mu)}{\ln(2)} \leq 4\mu^{2+\epsilon}.$$

$\square$

29

Putting together these claims, we obtain the bound:

Claim 10. For all $\mu \geq 2$, $\epsilon \geq \log_\mu(2)$ and $S \subseteq \mathsf{Primes} \times \mathbb{N}$:

$$\sum_{(p,r) \in S} \Delta\mathsf{val}(p,r) \leq 4n^{2+\epsilon} + \sum_{(p,r) \in S} \Delta\mathsf{weight}_{\mu,\epsilon}(p,r) \cdot (1 + \frac{1}{\epsilon}).$$

Proof. Partition S into disjoint sets S_1 and S_2 such that S_2 contains all the pairs $(p,r) \in S$ for which either $p \geq \mu^{1+\epsilon}$ or $r > 2(1+\epsilon)\frac{\mu \ln \mu}{\ln p}$, and S_1 contains the remaining pairs. $S_1 \subseteq B_{\mu,\epsilon}$ from Claim 9 and thus $\sum_{(p,r) \in S_1} \Delta\mathsf{val}(p,r) \leq 4\mu^{2+\epsilon}$. Furthermore, by Claim 7, if $(p,r) \in S_2$ then $\mathsf{density}_{\mu,\epsilon}(p,r) \leq 1 + \frac{1}{\epsilon}$ and by Claim 6:

$$\sum_{(p,r) \in S_2} \Delta\mathsf{val}(p,r) \leq \sum_{(p,r) \in S_2} \Delta\mathsf{weight}_{\mu,\epsilon}(p,r) \cdot (1 + \frac{1}{\epsilon}).$$

Putting everything together:

$$\sum_{(p,r) \in S} \Delta\mathsf{val}(p,r) = \sum_{(p,r) \in S_1} \Delta\mathsf{val}(p,r) + \sum_{(p,r) \in S_2} \Delta\mathsf{val}(p,r)$$
$$\leq 4n^{2+\epsilon} + \sum_{(p,r) \in S} \Delta\mathsf{weight}_{\mu,\epsilon}(p,r) \cdot (1 + \frac{1}{\epsilon}).$$

$\square$

Finally, we can conclude from Claim 10 that for any $S \in \mathsf{Primes} \times \mathbb{N}$, $\mu \geq 2$, and $\epsilon \geq \log_\mu(2)$, if $\sum_{(p,r) \in S} \Delta\mathsf{weight}_{\mu,\epsilon}(p,r) \leq \lambda$, i.e., if S satisfies the constraints of Eq. (Knapsack Problem) then:

$$\sum_{(p,r) \in S} \Delta\mathsf{weight}_{\mu,\epsilon}(p,r) \leq 4n^{2+\epsilon} + \lambda \cdot (1 + \frac{1}{\epsilon}).$$

The right hand side of the equation is therefore an upper bound for the solution to Eq. (Knapsack Problem), and consequently (by Claim 4 and Claim 5), an upper bound for the solution $t(\lambda, \mu)$ to Eq. (Constrained Max 1) when $\mu \geq 2$. In conclusion, for any $N \in \mathbb{N}$, $\mu \geq 2$, and $\epsilon \geq \log_\mu(2)$, if $\log_2 N \geq 4n^{2+\epsilon} + \lambda \cdot (1 + \frac{1}{\epsilon})$ then $\log_2 N \geq t(\lambda, \mu)$ and:

$$\mathbb{P}_{\mathbf{x} \leftarrow [0,m)^\mu}[f(\mathbf{x}) \equiv 0 \bmod N] \leq 2^{-\lambda} + \frac{\mu}{m}$$

A.4 Proof of Computational Inverse LCSZ (Theorem 3)

Theorem 3 (Computational bound). Let k be the output of algorithm Algorithm 1 on input λ, μ. Then for all $m \in \mathbb{N}$, all $N \geq 2^k$ and all μ-linear polynomials f, coprime with N, $\log_2 N \geq t(\lambda, \mu)$ and

$$P_{\mathbf{x} \leftarrow [0,m)^\mu}[f(\mathbf{x}) \equiv 0 \bmod N] \leq 2^{-\lambda} + \frac{\mu}{m}.$$

As shown in the prior section, $t(\lambda, \mu)$ is upper bounded by a solution to a knapsack problem, Eq. (Knapsack Problem), over the infinite set of items $\mathsf{Primes} \times \mathbb{N}$. There is a simple well-known greedy algorithm that returns an upper bound to the optimal value for the knapsack problem over a *finite* set of items. This algorithm greedily adds items to the knapsack in order of decreasing density until the knapsack overflows the weight bound, and returns the total value of the added items at this point. Over an infinite set of items, it is not generally possible to sort by decreasing density. However, by leveraging monotonicity properties of the density function in our case, we are able to adapt the greedy approximation algorithm to work for Eq. (Knapsack Problem). In particular, we are able to enumerate over pairs in $\mathsf{Primes} \times \mathbb{N}$ in an order of non-increasing density.

Claim 11. Let A and B be any pair of discrete strictly ordered sets which contain minimum elements $a_0 \in A$ and $b_0 \in B$. Let $a+$ denote the next element of A after a and likewise $b+$ the next element of B after b. If $f : A \times B \to \mathbb{R}^+$ is monotonically non-increasing over pairs (a, b_0) as $a \in A$ increases, and for any fixed a, monotonically non-increasing over pairs (a, b) as $b \in B$ increases, then the following algorithm enumerates the pairs $(a, b) \in A \times B$ in order of decreasing $f(a, b)$. The algorithm initializes the set $C = \{(a_0, b_0)\}$ and also a variable max_A to keep track of the highest order element in

A seen so far. At each iteration, it removes a pair $(a, b) \in C$ of lowest $f(a, b)$ value and appends (a, b) to the output enumeration list. Before proceeding to the next iteration, it adds $(a, b+)$ to C, and if $a = \mathsf{max}_A$ then it also adds $(a+, b_0)$ to C and updates $\mathsf{max}_A := a+$.

Proof. Suppose, towards contradiction, that the algorithm appends (a, b) to the output list, and there exists at least one pair (a', b') not yet in the list at this iteration such that $f(a', b') > f(a, b)$. If $b' \neq b_0$ and (a', b_0) appeared in the output list already, then each (a', b^*) for $b^* \in [b_0, b']$ would also have been added to C and removed before (a, b) because $f(a', b^*) \geq f(a', b') > f(a, b)$. This would be a contradiction, so it remains to consider the case that $b' = b_0$, i..e. that (a', b_0) did not appear in the list and $f(a', b_0) > f(a, b)$.

First, this implies that $a' > a$. Otherwise, If $a' \leq a$, then (a', b_0) would have been added to C before (a, b) and thus removed before (a, b). Second, $b \neq b_0$ and $f(a, b) < f(a, b_0)$, as otherwise this would imply that $f(a', b_0) > f(a, b) \geq f(a, b_0)$, contradicting the monotonicity property. Thus (a, b_0) must already appear in the output list because it is added to C before (a, b) and removed before (a, b). Furthermore, for all $a^* \in [a, a']$, $f(a^*, b_0) \geq f(a', b_0) > f(a, b)$, thus each such pair (a^*, b_0) would have been added and removed before (a, b). This is a contradiction. $\square$

We use the enumeration algorithm of Claim 11 to implement the greedy algorithm that obtains an upper bound to a generic knapsack problem over the infinite set of items $\mathsf{Primes} \times \mathbb{N}$:

$$\max_{S \subseteq \mathsf{Primes} \times \mathbb{N}} \sum_{(p,r) \in S} \mathsf{val}(p, r)$$

$$\text{subject to} \sum_{(p,r) \in S} \mathsf{weight}(p, r) \leq \lambda. \qquad \text{(Generic Knapsack Problem)}$$

for any $\mathsf{val}, \mathsf{weight} : \mathsf{Primes} \times \mathbb{N} \to \mathbb{R}^+$ where $\mathsf{density}(p, r) = \frac{\mathsf{val}(p,r)}{\mathsf{weight}(p,r)}$ is monotonic non-increasing over r for any fixed p, and also over p for fixed $r = 1$. This is presented below as Algorithm 1.

Moreover, we will show that the density function defined in terms of $\mathsf{val}(p, r) = \log p$, $\mathsf{weight}(p, 1) = -\log I_{1/p}(1, \mu)$, and $\mathsf{weight}(p, r) = \log I_{1/p}(r-1,) - \log I_{1/p}(r, \mu)$ for $r > 1$ satisfies this monotonicity property. The density function in Eq. (Knapsack Problem) also has this monotonicity property, but we are able to obtain a tighter bound on Eq. (Constrained Max 1) by defining regularized beta function directly instead of the simpler form upper bounds on the regularized beta function that were more useful for deriving the analytical result in Theorem 2.

Claim 12. Eq. (Generic Knapsack Problem) with $\mathsf{density}(p, r) = \frac{\mathsf{val}(p,r)}{\mathsf{weight}(p,r)}$, $\mathsf{val}(p, r) = \log p$, and $\mathsf{weight}(p, r) = \log I_{1/p}(r - 1, \mu) - \log I_{1/p}(r, \mu)$ is an upper bound to Eq. (Constrained Max 1).

Proof. The analysis is the same as in Claim 5, replacing $\Delta\mathsf{weight}_{\mu,\epsilon}$ with the weights defined here. $\square$

Claim 13. For $p \in \mathbb{P}, r \in \mathbb{N}$ and $\mu \geq 2 \in \mathbb{N}$ let $\mathsf{density}(p, r) = \frac{\mathsf{val}(p,r)}{\mathsf{weight}(p,r)}$ where $\mathsf{val}(p, r) = \log p$, and $\mathsf{weight}(p, r) = \log I_{1/p}(r - 1, \mu) - \log I_{1/p}(r, \mu)$ for $r \geq 1$. Then $\mathsf{density}(p, r)$ is decreasing in r and $\mathsf{density}(p, 1)$ is non-increasing in p.

Proof. **Part 1: $\mathsf{density}(p, r)$ is decreasing in r**

$\mathsf{density}(p, r) = \frac{\log p}{\mathsf{weight}(p,r)}$ is decreasing in r iff $\mathsf{weight}(p, r)$ is increasing in r.

$$\mathsf{weight}(p, r) = -\log I_{1/p}(r, \mu) + \log I_{1/p}(r - 1, \mu) = \log \frac{I_{1/p}(r - 1, \mu)}{I_{1/p}(r, \mu)}.$$

is decreasing in r if $\frac{I_{1/p}(r-1,\mu)}{I_{1/p}(r,\mu)}$ is decreasing in r. This is the case if for all r

$$\frac{I_{1/p}(r, \mu)}{I_{1/p}(r + 1, \mu)} - \frac{I_{1/p}(r - 1, \mu)}{I_{1/p}(r, \mu)} > 0.$$

Which is equivalent to showing that

$$I_{1/p}(r, \mu) \cdot I_{1/p}(r, \mu) - I_{1/p}(r - 1, \mu) \cdot I_{1/p}(r - 1, \mu) > 0. \qquad (7)$$

The regularized beta function $I_x(a, b)$ is log convave for all $b > 1$ as shown in [Kar15]. This implies that for $b > 1$ and all $\alpha, \beta > 0$ $I_x(a + \alpha, b) \cdot I_x(a + \beta, b) - I_x(a, b) \cdot I_x(a + \alpha + \beta, b) > 0$. Setting $a = r - 1, b = \mu, \alpha = 1, \beta = 1$ this shows that Eq. (7) holds and weight is incresing in r, and density is decreasing in r for all $\mu > 1$.

Part 2: density$(p, 1)$ is non-increasing in p.

$$\mathsf{density}(p, 1) = \frac{\log p}{\mathsf{weight}(p, 1)} = \frac{\ln p}{-\ln(1 - (1 - p^{-1})^\mu)}.$$

The derivative $\frac{d}{dp}\mathsf{density}(p, 1)$ is non-negative iff:

$$-p^{-1} \cdot \ln(1 - (1 - p^{-1})^\mu) - \frac{\ln p \cdot \mu (1 - p^{-1})^{\mu-1} \cdot p^{-2}}{1 - (1 - p^{-1})^\mu} \geq 0.$$

Equivalently:

$$-\ln(1 - (1 - p^{-1})^\mu) - \frac{\ln p \cdot \mu (1 - p^{-1})^{\mu-1} p^{-1}}{1 - (1 - p^{-1})^\mu} \geq 0.$$

Set $x = 1 - \frac{1}{p}$, which increases over the range $(1/2, 1)$ as p increases in the range $[2, \infty)$. The requisite inequality for $x \in (1/2, 1)$ becomes:

$$-\ln(1 - x^\mu) + \frac{\ln(1 - x) \cdot \mu \cdot x^{\mu-1}(1 - x)}{1 - x^\mu} \geq 0.$$

which, rearranging terms, holds true iff for $x \in (1/2, 1)$:

$$\frac{\ln(1 - x^\mu)(1 - x^\mu)}{\ln(1 - x)(1 - x)x^{\mu-1}} \leq \mu.$$

In fact we can show this inequality holds true over all $x \in (0, 1)$. Using the inequalities $\ln(1 - x^\mu) \leq -x^\mu$ and $-\ln(1 - x) \geq x$ we obtain:

$$\frac{\ln(1 - x^\mu)(1 - x^\mu)}{\ln(1 - x)(1 - x)x^{\mu-1}} \leq \frac{x^\mu(1 - x^\mu)}{x(1 - x)x^{\mu-1}} = \frac{1 - x^\mu}{1 - x} \leq \mu.$$

$\square$

Proof of Theorem 3

Proof. Algorithm 1 is a greedy enumeration algorithm over pairs (p, r) following the enumeration strategy in Claim 11. By Claim 13, density satisfies the monotonicity conditions required for Claim 11 and thus the enumeration algorithm enumerates pairs in order of non-increasing density. Thus, the algorithm outputs an upper bound to (Generic Knapsack Problem) with density $\mathsf{density}(p, r) = \frac{\mathsf{val}(p,r)}{\mathsf{weight}(p,r)}$, $\mathsf{val}(p, r) = \log p$, and $\mathsf{weight}(p, r) = \log I_{1/p}(r - 1, \mu) - \log I_{1/p}(r, \mu)$. By Claim 12 this is an upper bound to Eq. (Constrained Max 1) which in turn by Theorem 1 gives a bound on $\log_2 t(\lambda, \mu)$. $\square$

B Definitions and Notations

B.1 Integer Polynomials

If f is a multivariate polynomial, then $||f||_\infty$ denotes the maximum over the absolute values of all coefficients of f.

Lemma 7 (Evaluation Bound). For any μ-linear integer polynomial f and $m \geq 2$:

$$\mathbb{P}_{\mathbf{x} \leftarrow [0,m)^\mu}[|f(\mathbf{x})| \leq \frac{1}{m^\mu} \cdot ||f||_\infty] \leq \frac{3\mu}{m}$$

Proof. Let $f^{(0)} := f$. Given a vector $\mathbf{x} = (x_1, .., x_\mu)$, for each $j \in [1, \mu]$ define $f_{\mathbf{x}}^{(j)}$ to be the $\mu - j$-variate partial evaluation $f_{\mathbf{x}}^{(j)} := f(x_1, ..., x_j, X_{j+1}, ..., X_\mu)$. Then we can rewrite the lemma statement as:

$$P_{\mathbf{x} \leftarrow [0,m)^\mu}[||f_{\mathbf{x}}^{(\mu)}||_\infty \leq \frac{1}{m^\mu} \cdot ||f^{(0)}||_\infty] \leq \frac{3\mu}{m}$$

We will bound the probability for random $\mathbf{x}$ that there exists any j for which $||f_{\mathbf{x}}^{(j)}|| < \frac{1}{m} \cdot ||f_{\mathbf{x}}^{(j-1)}||$. If no such j exists, then $||f^{(\mu)}|| \geq \frac{1}{m^\mu} \cdot ||f^{(0)}||$.

For any j, we can write $f_{\mathbf{x}}^{(j)} = g(X_{j+1}, ..., X_\mu) + x_j \cdot h(X_{j+1}, ..., X_\mu)$ where g, h are $\mu - j$ variate multilinear integer polynomials and $||f_{\mathbf{x}}^{(j-1)}|| = \max(||g||, ||h||)$ because the coefficients of g and h are a partition of the coefficients of $f_{\mathbf{x}}^{(j-1)}$. Suppose now that $||f_{\mathbf{x}}^{(j)}|| < \frac{1}{m} \cdot ||f_{\mathbf{x}}^{(j-1)}||$, i.e. that $||g + x_j \cdot h|| < \frac{1}{m} \cdot \max(||g||, ||h||)$ and consider two cases:

Case 1: $||h|| = \max(||g||, ||h||)$. For any integer $\Delta \neq 0$, using the triangle inequality:

$$||g + (x_j + \Delta)h|| = ||g + x_j h + \Delta h|| \geq ||\Delta h|| - ||g + x_j h|| > (1 - \frac{1}{m}) \cdot ||h|| \geq \frac{1}{m} \cdot ||h||$$

The last part of the inequality holds because $1 - \frac{1}{m} \geq \frac{1}{m}$ for any $m \geq 2$.

Case2: $||g|| = \max(||g||, ||h||)$. Using the triangle inequality,

$$\frac{1}{m}||g|| > ||g + x_j \cdot h|| \geq ||g|| - ||x_j \cdot h||$$

This implies, for $m \geq 2$, that $||h|| > \frac{1}{m} \cdot ||g||$ because:

$$||x_j \cdot h|| > (1 - \frac{1}{m}) \cdot ||g|| \implies ||h|| > \frac{m-1}{x_j \cdot m} \cdot ||g|| \geq \frac{1}{m}||g||$$

The last step uses that $x_j \in [1, m)$. For $x_j = 0$, $||g + x_j h|| = ||g||$. Finally, for any integer Δ, by the triangle inequality:

$$||g + (x_j + \Delta) \cdot h|| \geq ||\Delta h|| - ||g + x_j \cdot h|| > \frac{|\Delta|}{m} \cdot ||g|| - \frac{1}{m} \cdot ||g|| = \frac{|\Delta| - 1}{m} \cdot ||g||$$

When $|\Delta| \geq 2$ this implies that $||g + (x_j + \Delta) \cdot h|| > \frac{1}{m} \cdot ||g||$.

In both cases, we conclude that for any choice of $(x_1, ..., x_{j-1})$ for the first $j - 1$ components of the random $\mathbf{x}$, which define g and h, there are at most three choices of x_j such that the event $||f_{\mathbf{x}}^{(j)}|| < \frac{1}{m} \cdot ||f_{\mathbf{x}}^{(j-1)}||$ holds true (i.e., if true for x_j, then it is also true for at most $x_j + 1$ and $x_j - 1$). Thus this event occurs with probability at most $\frac{3}{m}$. Finally, by a union bound over j, the probability this event occurs for some index j is at most $\frac{3\mu}{m}$.

$\square$

Fact 1. Let $q \in \mathbb{Z}$ be any positive integer. For any integer $E \in \mathbb{Z}$ such that $|E| \leq \frac{q^{d+2} - q}{2(q-1)}$ there exists a unique degree d integer polynomial $f \in \mathbb{Z}[X]$ with $||f||_\infty \leq q/2$ such that $f(q) = E$.

Lemma 8 (Rational Encoding of multi-linear polynomials). Let $q \in \mathbb{Z}$ be any positive integer. Let $\vec{q} = [q^{2^{i-1}}]_{i=1}^\mu \in \mathbb{Z}^\mu$. Consider any $\beta_d, \beta_n \in \mathbb{N}$ such that $\beta_d \cdot \beta_n \leq \frac{q}{2}$. Let $Z = \{z \in \mathbb{Z} : |z| \leq \beta_d\}$, let $\mathcal{F} = \{f \in \mathbb{Z}[X_1, ..., X_\mu] : ||f||_\infty \leq \beta_n\}$ be a μ-linear polynomial, and let $\mathcal{H} = \{f/z \in \mathbb{Q}[X_1, ... X_\mu] : f \in \mathcal{F} \wedge z \in Z\}$. Then for any $h_1, h_2 \in \mathcal{H}$, if $h_1(\vec{q}) = h_2(\vec{q})$ then $h_1 = h_2$.

Proof. Let $h_1 = \frac{f_1}{z_1}$ and $h_2 = \frac{f_2}{z_2}$. If $h_1(\vec{q}) = h_2(\vec{q})$ then $z_1 f_2(\vec{q}) = z_2 f_1(\vec{q})$. Since $||z_2 \cdot f_1||_\infty \leq \beta_d \cdot \beta_n \leq \frac{q}{2}$ and likewise $||z_1 \cdot f_2||_\infty \leq \frac{q}{2}$. Note that there exist a unique univariate degree $2^\mu - 1$ polynomial $\hat{f}_1$ that has the same coefficients as f_1 such that for all q $f_1(\vec{q}) = \hat{f}_1(q)$. Let f_2 be the univariate degree $2^\mu - 1$ polynomial with the same coefficients as $\hat{f}_2$. It then follows from Fact 1 that if $z_1 f_2(\vec{q}) = z_1 \hat{f}_2(q) = z_2 \hat{f}_1(q) = z_2 f_1(\vec{q})$ then $z_1 f_2 = z_2 f_1$, or equivalently, $h_1 = h_2$. $\square$

B.2 Groups of Unknown Order

A group of unknown order is a group where the order is computationally hard to compute. It is defined by an algorithm *GGen* that on input security parameter, samples a group $\mathbb{G}$, along with size bounds on the group that depend on the security parameter (we omit the size bounds for simplicity). We define three assumptions in these groups. The Order assumption which is the most basic, minimal assumption, saying that it is hard to compute the order of random group elements. The stronger sub-group order assumption states that it is hard to compute any information about the order of a sampled subgroup, e.g., a subgroup generated by a commitment key. And finally, the famous RSA assumption which states that it is hard to compute roots of random elements in the group and implies the order assumption.

Assumption 1 (Order Assumption). The order assumption holds for a group sampling algorithm $GGen$ if for any probabilistic polynomial time adversary $\mathcal{A}$:

$$\Pr\left[a \cdot \mathsf{G} = 0 : \begin{array}{c} \mathbb{G} \leftarrow GGen(1^\lambda) \\ \mathsf{G} \xleftarrow{\$} \mathbb{G} \\ a \in \mathbb{Z} \leftarrow \mathcal{A}(\mathbb{G}, \mathsf{G}) \end{array} \right] \leq \mathsf{negl}(\lambda) \ .$$

Assumption 2 (Subgroup Order Assumption). The subgroup order assumption is a generalization of the order assumption. It says that it is difficult to compute a multiple of the order of any element in a subgroup sampled according to some distribution. It holds for a group sampling algorithm $GGen$ and subgroup sampling[11] algorithm $SGGen$ if for any probabilistic polynomial time adversary $\mathcal{A}$:

$$\Pr\left[\gcd(a, |\mathbb{H}|) \neq 1 : \begin{array}{c} \mathbb{G} \leftarrow GGen(1^\lambda) \\ \mathbb{H} \leftarrow SGGen(\mathbb{G}) \\ a \in \mathbb{Z} \leftarrow \mathcal{A}(\mathbb{G}, \mathbb{H}) \end{array} \right] \leq \mathsf{negl}(\lambda) \ .$$

Assumption 3 (RSA assumption, [RSA78; CPP17]). The RSA assumption holds for $GGen$ if for any probabilistic polynomial time adversary $\mathcal{A}$:

$$\Pr\left[\ell \cdot \mathsf{U} = \mathsf{G} \ : \ \begin{array}{c} \mathbb{G}, N \leftarrow GGen(1^\lambda) \\ \mathsf{G} \xleftarrow{\$} \mathbb{G}, \ell \xleftarrow{\$} [N] \\ \mathsf{U} \in \mathbb{G} \leftarrow \mathcal{A}(\mathbb{G}, \mathsf{G}) \end{array} \right] \leq \mathsf{negl}(\lambda) \ .$$

The RSA Assumption implies Assumption 1[BFS19].

B.3   IP Transcript Trees

Let (P, V) be a μ-round public coin interactive protocol. A μ-round public-coin protocol (P, V) consists of μ-rounds of messages between the prover and verifier, where in each round the prover sends a message to the verifier and the verifier responds with $x \leftarrow \mathcal{X}$ sampled uniformly from the challenge space $\mathcal{X}$. At the end of the protocol, the verifier outputs either *accept* or *reject*. By convention, the protocol starts with the prover's first message and ends with the prover's last message. A *transcript* thus contains $\mu + 1$ prover messages and μ challenges. We will denote transcripts by a $\mu \times 2$ matrix A such that $A(0, 0)$ is the protocol input x, $A(0, 1)$ is the prover's first message, and for all $i \geq 1$, $A(i, 0)$ is the verifier's *ith* round challenge and $A(i, 1)$ is the prover's *ith* round response. We restrict our attention to protocols in which the verifier's decision is a deterministic function D_V of the transcript, but is also without loss of generality. An *accepting transcript* is an array A such that $D_V(A) = accept$.

A k-ary *transcript tree* for (P, V) is a labeling of a μ-depth k-ary tree such that the labels on every root-to-leaf path forms an accepting (P, V) transcript. It will be convenient to order the nodes of the tree according to a depth-first reverse topological sort (aka post-order tree traversal). This is a topological sorting of the tree with directed edges flowing from leaves to root which places left subtrees before right subtrees. This ordering associates each node with an index in $[1, N]$ where $N = \mathsf{size}(\mu, k) = \frac{k^{\mu+1}-1}{k-1}$. A *post-order labeling* of the tree is a function $L : [1, N] \to \{0, 1\}^*$. We may refer to the *level* of a node in the tree. The root of a tree is always at *level 0* and the leaves of a depth μ tree are at *level μ*. The *height* of node at level ℓ within a μ-depth tree is $\mu - \ell$. For each $v \in [1, N]$ let $S_v \subseteq [1, N]$ denote the indices of all nodes in the subtree rooted at node v. For each $v \in [1, N]$ let v^* denote the largest index $v^* < v$ such that the node at index v^* does not belong to the subtree S_v. Note that for nodes on the leftmost path of the tree v^* does not exist so we denote it by $\perp$. For any subset $S \subseteq [1, N]$ et $\mathsf{pred}(S)$ denote a superset of S which include the indices of all predecessors of nodes S. Finally, for each $v \in [1, N]$ let $L_v : [1, v] \to \{0, 1\}^*$ denote the restriction of L to the subset $\mathsf{pred}([1, v])$. More generally, for any $S \subseteq [1, N]$ let $L_S : S \to \{0, 1\}^*$ denote the restriction of the labeling L to the subset of node indices S. L_{S_v} thus denotes the labeling of the subtree S_v. Note that L_v is not the same as L_{S_v}.

A transcript tree is thus a special post-order labeling of the tree $L : [1, N] \to \mathcal{X} \times \mathcal{M}$ where $\mathcal{X}$ is the verifier's challenge set and $\mathcal{M}$ is the space of prover messages. We can think of the first component (i.e., the verifier challenge) as a label on the node's incoming edge and the second component (i.e., prover's response) as a label on the node itself. The root has no incoming edge, but the root label's first component is the

---

[11]The subgroup sampling algorithm takes $\mathbb{G}$ as input, which is interpreted as a succinct description of $\mathbb{G}$, such as a list of generators, not necessarily the list of all elements in $\mathbb{G}$.
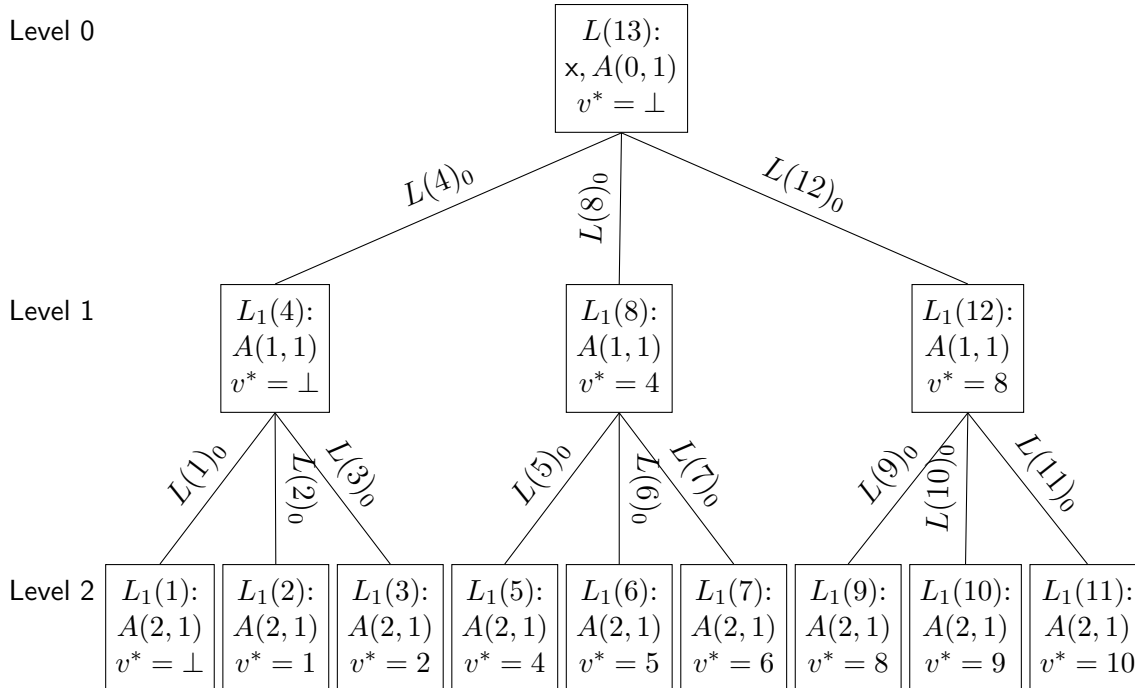
Figure 2: IP transcript tree for $\mu = k = 3$. Nodes and edges are labeled using post-order labeling. We also indicate v^* for every node.

protocol input. For any root-to-leaf path of nodes with indices $\{v_0, ..., v_\mu\}$ the labeling L defines the matrix A such that $L(v_i) = (A(i,0), A(i,1))$ and A is an accepting transcript. Given a label $L(v)$ for $v < N$ (non-roots) we will use the notation $L(v)_0$ to denote the first component of the label containing the verifier's challenge and $L(v)_1$ the second component containing the prover's response. We define a k-ary *forking transcript tree* to be a k-ary transcript tree in which the challenge labels on all edges sharing a common parent are distinct.

Finally, we define a *transcript tree labeling* as a function of transcript trees that returns a derived labeling of the tree, i.e. a function $f : (\mathcal{X} \times \mathcal{M})^N \to \mathcal{C}^N$ that given the labeling L corresponding to a k-ary transcript tree induces the labeling $f(L) : [1, N] \to \mathcal{C}$. For any $\nu \in [1, N]$ let $\mathsf{pred}(\nu)$ denote the indices of the path starting at the root and ending at node ν so that $L_{\mathsf{pred}(\nu)}$ denotes the labels assigned to the nodes along this path by L, i.e. in a transcript tree this is the prefix of all transcripts that pass through ν. A *localized transcript tree labeling* further restricts f so that the label $c_\nu \in \mathcal{C}$ that $f(L)$ assigned to node ν is a function of $L_{\mathsf{pred}(\nu)}$.

B.4 Path Predicate Forking Lemma

The standard forking lemma for μ-round public coin interactive protocols characterizes the efficiency of generating a k-ary μ-depth transcript tree for which the challenges labeling the children within the tree *fork*, i.e. are distinct. More precisely, the forking lemma says that given any adversarial prover $\mathcal{A}$ that may deviate from the honest protocol but causes the verifier to accept with probability ϵ, there is a tree generation algorithm that has only black-box access to $\mathcal{A}$, runs in time $t \in O(\frac{\lambda}{\epsilon} \cdot k^\mu \cdot (\mu + t_V))$, where t_V is the running time of the verifier's decision algorithm, and succeeds with probability $1 - t \cdot \mathsf{negl}(\lambda)$ in producing a transcript tree with the forking property.

Our *path predicate forking lemma* generalizes the property of the transcript tree that can be generated by considering arbitrary predicates on partial labelings of the tree. In the standard forking lemma, the predicate would simply be that new challenges are distinct from previous challenges. The lemma considers more general predicates for each node $v \in [1, N]$, which may depend on L_v. Recall that for each $v \in [1, N]$, the node v^* is the largest index $v^* < v$ such that the node at this index is not contained in the subtree S_v rooted at v and $\mathsf{pred}([1, v^*])$ is the superset of $[1, v^*]$ that includes the indices of all predecessors of nodes $[1, v^*]$, which by definition also do not intersect with S_v. Our lemma considers predicates at level ℓ_v of the form $\pi_v : (\mathcal{X} \times \mathcal{M})^{\mathsf{pred}([1, v^*])} \times \mathcal{X}^{\mu - \ell_v} \to \{0, 1\}$, i.e. each predicate π_v takes as input a labeling function L_{v^*} and a vector of challenges $\mathbf{x} \in \mathcal{X}^{\mu - \ell_v}$.

The vector of challenges will represent *the leftmost* path down the tree starting from v, which by definition is independent of the partial labeling L_{v^*}. We denote the indices of the leftmost path from v to the leaves as lpath_v and the challenge labels along this path assigned by L as $L(\mathsf{lpath}_v)_0$. For example in Figure 2 the predicate π_8 for node

8 would take as input the labels on the subtree spanned by node 4, the root label, and the challenge $L(5)_0$. The lemma says that if $\pi_v(L_{v^*}, \mathbf{x}) = 1$ with overwhelming probability $1 - \mathsf{negl}(\lambda)$ for any post-order labeling $L : [1, N] \to \mathcal{X} \times \mathcal{M}$ of the k-ary μ-depth tree, any node v in the tree, and $\mathbf{x}$ sampled randomly, then the transcript generation algorithm produces a transcript tree represented by some post-order labeling L for which $\pi_v(L_{v^*}, L(\mathsf{lpath}_v)_0) = 1$ for all v in the tree. In fact, the lemma is even more general as it has a weaker requirement that $\pi_v(L_v, \mathbf{x}) = 1$ with overwhelming probability conditioned on $\pi_u(L_u, L(\mathsf{lpath}_u)_0) = 1$ for all $u \le v^*$. The standard forking lemma is a special case where π_v checks that the challenge label on v is distinct from the challenge labels on any of its left siblings. The challenge label $L(v)_0$ on v is the first component of $L(\mathsf{lpath}_v)_0$. Assuming v is not a first child, the labels on the left sibling(s) of v as well as its parent are included in L_{v^*}.

Proof Overview   We will begin with a high level overview of the proof. The algorithm is exactly the same as the recursive tree generation algorithm for the standard forking lemma. The difference is only in the analysis. The standard forking lemma considers predicates $\pi_v(L_{v^*}, x)$ that are functions only of the challenges assigned by L_{v^*} to left sibling nodes of v and a single (fresh) $x \in \mathcal{X}$ rather than a vector, and are independently true with overwhelming probability.

Just as in the standard forking lemma, the analysis is a simple union bound. First, the tree generation algorithm is transformed to a Monte Carlo algorithm that runs for $t \in \mathsf{poly}(\lambda)$ steps and succeeds with overwhelming probability. The standard forking lemma is based on the observation that a t-step algorithm makes at most t samples from $\mathcal{X}$ and thus the predicates hold true for all sampled challenges with probability at least $1 - t \cdot \mathsf{negl}(\lambda)$. In our case, the analysis is very similar. Let L denote the labeling returned by the Monte Carlo tree generation algorithm. We begin with the observation that this tree generation algorithm constructs the labeling in depth-first post-order. In particular, when the transcript tree generation algorithm visits a node v at heigh h_v it has already derived a partial labeling L_{v^*}. It samples a random vector $\mathbf{x} \in \mathcal{X}^{h_v}$ and attempts to derive a valid transcript for lpath_v using this challenge vector $\mathbf{x}$. If it succeeds then it sets $L(\mathsf{lpath}_v)_0 = \mathbf{x}$, otherwise the entire vector $\mathbf{x}$ is discarded and it tries again starting from v. Suppose there exists some v such that $\pi_v(L_{v^*}, L(\mathsf{lpath}_v)_0) = 0$ and let v be the lowest index node with this property. This would imply that there occurred an event where the algorithm had already constructed L_{v^*} satisfying $\pi_u(L_{u^*}, L(\mathsf{lpath}_u)_0) = 1$ for all $u \le v^*$ and then sampled $\mathbf{x} \leftarrow \mathcal{X}^{h_v}$, setting $L(\mathsf{lpath}_v)_0 = \mathbf{x}$, such that $\pi_v(L_{v^*}, \mathbf{x}) = 0$. However, by hypothesis this event occurs with probability $\mathsf{negl}(\lambda)$ over random $\mathbf{x}$. Since the algorithm runs for only $t \in \mathsf{poly}(\lambda)$ steps, an event of this kind occurs with probability at most $t \cdot \mathsf{negl}(\lambda)$.

Thus, we obtain a Monte Carlo algorithm that returns a transcript tree where all the predicates are satisfied with overwhelming probability.

Lemma 9 (Path Predicate Forking Lemma)**.** Let (P, V) be a μ-round public-coin protocol with prover message space $\mathcal{M}$ and verifier challenge space $\mathcal{X}$. For each node $v \in [1, N]$ of a μ-depth k-ary balanced tree on $N = \mathsf{size}(\mu, k)$ nodes, let h_v denote the height of v. Let $\{\pi_v : v \in [1, N]\}$ denote a set of predicates, where $\pi_v(L_{v^*}, \mathbf{x})$ is a function of the partial labeling L_{v^*} and challenge vector $\mathbf{x} \in \mathcal{X}^{h_v}$, with the property that for any post-order labeling function $L : [1, N] \to \mathcal{X} \times \mathcal{M}$ and any $v \in [1, N]$:

$$Pr_{\mathbf{x} \leftarrow \mathcal{X}^{h_v}}[\pi_v(L_{v^*}, \mathbf{x}) = 1 \mid \forall_{u \le v^*} \pi_u(L_{u^*}, L(\mathsf{lpath}_u)_0) = 1] \ge 1 - \delta$$

Let t_V denote the worst-case running time of the verifier's decision algorithm D_V. There is an algorithm $\mathsf{Tree}^{\mathcal{A}}(\mathsf{z})$ that, given a security parameter $\lambda \in \mathbb{N}$ and oracle access to an adversarial prover $\mathcal{A}$ that causes V to accept with probability ϵ on public input z, runs in time at most $t = 2\lambda \cdot \frac{k^\mu}{\epsilon} \cdot (\mu + t_V)$ and with probability at least $1 - t \cdot \delta - 2^{-\lambda}$ outputs a k-ary transcript tree with post-order labeling $L : [1, N] \to \mathcal{X} \times \mathcal{M}$ such that $\pi_v(L_{v^*}, L(\mathsf{lpath}_v)_0) = 1$ for all $v \in [1, N]$.

Proof. We will first describe a Las Vegas tree-finding algorithm that runs in expected polynomial time as we can then transform it to a Monte Carlo algorithm with a finite runtime and overwhelming success probability.

Tree finding algorithm   The tree-finding algorithm $\mathsf{Tree}(k, \mathsf{z})$ begins by sampling a random tape σ for the adversary. Let $A(\sigma)$ denote the *deterministic*[12] adversary with

---

[12]Any probabilistic adversarial algorithm can be represented by a deterministic algorithm that takes as input a random tape.

fixed random tape σ. For all $i \in [0, \mu]$ define $T_i(\sigma, k, \mathsf{z}, x_1, ..., x_i)$ as follows:

Algorithm $T_i(\sigma, k, \mathsf{z}, x_1, ..., x_i)$**:**

- **If $i = \mu$:** Simulate the protocol with $\mathcal{A}(\sigma)$ as the prover and fixing the verifier's challenges μ ordered challenges to the values $x_1, ..., x_\mu$. If the verifier outputs 1 during this simulation then return the protocol transcript tr, and otherwise return fail.

- **Else if $0 \leq i < \mu$:** Sample $x_{i+1} \leftarrow \mathcal{X}$ and run $T_i(\sigma, k, y, x_1, ..., x_{i+1})$. This either returns fail or a transcript tree denoted tree. If it returns fail, then output fail. Otherwise, save the pair (x_{i+1}, tree). If $i < \mu - 1$ then tree is a tree of accepting transcripts that share a common prefix for the first $i+1$ rounds, which includes the challenges $x_1, ..., x_{i+1}$. If $i+1 = \mu$ then tree is a single accepting transcript. Repeat this process as many times as needed, each time sampling a fresh x'_{i+1}, running $T_i(\sigma, y, x_1, ..., x'_{i+1})$, ignoring the runs that fail, saving the succesful challenge/tree pairs until k pairs have been recorded. Together the transcripts in all k recorded trees form one larger tree of accepting transcripts that share a commmon prefix $\mathsf{tr_{pre}}$ for the first i rounds of messages with fixed challenges $x_1, ..., x_i$.

$\mathsf{Tree}(k, \mathsf{z})$ repeatedly samples σ and runs $T_0(\sigma, k, \mathsf{z})$ until it outputs a tree of accepting transcripts.

We now analyze the expected runtime of $\mathsf{Tree}(k, \mathsf{z})$ and success probability of returning an k-ary tree of accepting transcripts given that $\mathcal{A}$ succeeds with probability ϵ. $T_0(\sigma, k, \mathsf{z})$ returns fail iff the first iteration of each subroutine T_i returns fail for $i = 1$ to μ. The probability this happens is equal to the probability that $T_\mu(\sigma, y, x_1, ..., x_\mu)$ outputs fail for a uniformly distributed challenge tuple $(x_1, ..., x_\mu)$. This is equal to the failure probability of $\mathcal{A}(\sigma)$, i.e. $1 - \epsilon$. Thus, $\mathsf{Tree}(k, \mathsf{z})$ calls T_0 in expectation $1/\epsilon$ times. Letting t_0 be a random variable for the runtime of $T_0(\sigma, \mathsf{z})$ over random σ, the expected runtime of $\mathsf{Tree}(k, \mathsf{z})$ is t_0/ϵ.

It remains to analyze the expected runtime $\mathbb{E}[t_0]$ of $T_0(\sigma, \mathsf{z})$. Each call to $T_i(\sigma, k, \mathsf{z}, x_1, ..., x_i)$ for $i \in [1, \mu]$ that occurs in the execution trace of $T_0(\sigma, k, \mathsf{z})$ is on i.i.d. uniformly distributed challenges $x_1, ..., x_i$. Let t_i be a random variable denoting the runtime of $T_i(\sigma, k, \mathsf{z}, x_1, ..., x_i)$ over a uniformly distributed challenge prefix $\mathbf{x}_i = (x_1, ..., x_i)$ and uniformly distributed σ. We omit the time to sample a random challenge from the runtime analysis as this will only affect the runtime up to a constant factor. Since $T_\mu(\sigma, k, \mathsf{z}, \mathbf{x}_\mu)$ makes μ calls to the oracle $\mathcal{A}$ and one call to the verifier's decision algorithm D_V its runtime is at most $\mu + t_V$, where t_V is the worst case running time of D_V.

For $i < \mu$, $T_i(\sigma, k, y, \mathbf{x}_i)$ outputs fail iff the first call to each T_j subroutine for $j \in [i+1, \mu]$ returns fail, in which case the runtime is $t_\mathcal{A}$. The probability $T_i(\sigma, k, \mathsf{z}, \mathbf{x}_i)$ outputs fail for random σ and $\mathbf{x}_i$ is again equal to the failure probability of $\mathcal{A}(\sigma)$, i.e. $1 - \epsilon$. If it does not output fail, then in expectation it runs an additional $(k-1)/\epsilon$ iterations of $T_{i+1}(\sigma, k, \mathsf{z}, \mathbf{x}_i, x_{i+1})$ sampling a fresh x_{i+1} for each iteration. Thus, the expected runtime $\mathbb{E}(t_i)$ is:

$$\mathbb{E}[(1 - \epsilon) \cdot t_\mathcal{A} + (k-1) \cdot t_{i+1}] \leq \mathbb{E}[t_{i+1} \cdot k]$$

This recurrence relation shows:

$$\mathbb{E}[t_0] \leq \mathbb{E}[t_\mu \cdot k^\mu]$$

Thus, we have shown that the expected runtime of $\mathsf{Tree}(k, \mathsf{z})$ is $\mathbb{E}[t] \leq \frac{k^\mu}{\epsilon} \cdot (\mu + t_V)$.

By standard techniques[13], the Las Vegas algorithm $\mathsf{Tree}(k, \mathsf{z})$ may be transformed to a Monte Carlo algorithm that runs for $2\lambda \cdot \mathbb{E}[t])$ steps and succeeds except with probability $1 - 2^{-\lambda}$.

Transcript tree property analysis The transcript tree labels returned by $\mathsf{Tree}(k, \mathsf{z})$ are computed in depth-first post-order by the Monte Carlo tree generation algorithm. Let L denote this post-order labeling. Consider any node v that is labeled with challenge $L(v)_0 = x$ in the tree. Let i denote the level of v within the tree and let $\mathbf{x} = (x_1, .., x_{i-1}, x, x_{i+1}, ..., x_\mu)$ denote the vector of challenge labels assigned to the path starting from the root to v and following the left-most path down the tree from v. During the execution of $\mathsf{Tree}(k, \mathsf{z})$ the following event occurred: immediately after x was

---

[13]Run λ independent instances in parallel for $2 \cdot \mathbb{E}[t]$ steps. By Markov, each instance terminates (i.e., succeeds) with probability at least $1/2$. The probability none succeed is at most $2^{-\lambda}$

sampled as a candidate label for v, the challenges $x_{i+1}, ..., x_\mu$ were sampled uniformly and independently such that $\mathcal{A}(\sigma, k, \mathsf{z}, \mathbf{x})$ succeeded (i.e., produced a valid transcript). If this event had not occurred (i.e., $\mathcal{A}(\sigma, k, \mathsf{z}, \mathbf{x})$ failed) then x would have been discarded and the process would have been repeated.

In other words, when the transcript tree generation algorithm visits a node v it has already derived a partial labeling L_{v^*} for the nodes at indices $\mathsf{pred}([1, v^*])$, which are not in any subtree rooted at v. It samples a random vector $\mathbf{x} \in \mathcal{X}^{h_v}$ and attempts to derive a valid transcript for lpath_v using this challenge vector $\mathbf{x}$. If it succeeds, then it sets $L(\mathsf{lpath}_v)_0 = \mathbf{x}$, otherwise the entire vector $\mathbf{x}$ is discarded, and it tries again starting from v. Suppose there exists some v such that $\pi_v(L_{v^*}, L(\mathsf{lpath}_v)_0) = 0$ and let v be the lowest index node with this property. This would imply that there occurred an event where the algorithm had already partially constructed L such that $\pi_u(L_{u^*}, L(\mathsf{lpath}_u)_0) = 1$ for all $u \leq v^*$ and then subsequently sampled $\mathbf{x} \leftarrow \mathcal{X}^{h_v}$, setting $L(\mathsf{lpath}_v)_0 = \mathbf{x}$, such that $\pi_v(L_{v^*}, \mathbf{x}) = 0$. However, by hypothesis this event occurs with probability δ over random $\mathbf{x}$. The algorithm runs for at most $t = \frac{2\lambda}{\epsilon} k^\mu \cdot (\mu + t_V)$ steps in total, hence by a union bound the probability that an event of this kind occurs at all is at most $t \cdot \delta$.

Thus, we obtain a Monte Carlo extraction algorithm that returns a transcript tree where all the predicates are satisfied with overwhelming probability (for appropriate setting of the parameters). More precisely, for any security parameter $\lambda \in \mathbb{N}$ and for $t = \frac{2\lambda}{\epsilon} \cdot k^\mu \cdot (\mu + t_V)$ the extraction algorithm runs in time at most t and (by a union bound) succeeds in returning a transcript tree labeling L where, for all v, $\pi_v(L_{v^*}, L(\mathsf{lpath}_v)_0) = 1$ with probability at least $1 - t \cdot \delta - 2^{-\lambda}$.

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\square$

B.5 Knowledge Soundness

An NP relation $\mathcal{R}$ is a subset of strings $x, w \in \{0, 1\}^*$ such that there is a decision algorithm to decide $(x, w) \in \mathcal{R}$ that runs in time polynomial in $|x|$ and $|w|$. The language of $\mathcal{R}$, denoted $\mathcal{L}_R$, is the set $\{x \in \{0, 1\}^* : \exists w \in \{0, 1\}^* \ s.t. \ (x, w) \in \mathcal{R}\}$. The string w is called the *witness* and x the *instance*. An **interactive proof of knowledge** for an NP relation $\mathcal{R}$ is a special kind of two-party interactive protocol between a prover denoted $\mathcal{P}$ and a verifier denoted $\mathcal{V}$, where $\mathcal{P}$ has a private input w and both parties have a common public input x such that $(x, w) \in \mathcal{R}$. Informally, the protocol is *complete* if $\mathcal{P}(x, w)$ always causes $\mathcal{V}(x)$ to output 1 for any $(x, w) \in \mathcal{R}$. The protocol is *knowledge sound* if there exists an extraction algorithm $\mathcal{E}$ called the *extractor* such that for every x and adversarial prover $\mathcal{A}$ that causes $\mathcal{V}(x)$ to output 1 with non-negligible probability, $\mathcal{E}$ outputs w such that $(x, w) \in \mathcal{R}$ with overwhelming probability given access[14] to $\mathcal{A}$.

Definition 9 (Interactive Proof of Knowledge). An interactive protocol $\Pi = (\mathcal{P}, \mathcal{V})$ between a prover $\mathcal{P}$ and verifier $\mathcal{V}$ is a proof of knowledge for a relation $\mathcal{R}$ with knowledge error $\delta : \mathbb{N} \to [0, 1]$ if the following properties hold, where on common input x and prover witness w the output of the verifier is denoted by the random variable $\langle \mathcal{P}(x, w), \mathcal{V}(x) \rangle$:

- Perfect Completeness: for all $(x, w) \in \mathcal{R}$

$$\Pr\left[\langle \mathcal{P}(x, w), \mathcal{V}(x) \rangle = 1 \right] = 1$$

- δ-Knowledge Soundness: There exists a polynomial $\mathsf{poly}(\cdot)$ and a probabilistic oracle machine $\mathcal{E}$ called the *extractor* such that given oracle access to any adversarial interactive prover algorithm $\mathcal{A}$ and any input $x \in \mathcal{L}_R$ the following holds: if

$$\mathbb{P}\left[\langle \mathcal{A}(x), \mathcal{V}(x) \rangle = 1\right] = \epsilon(x)$$

then $\mathcal{E}^\mathcal{A}(x)$ with oracle access to $\mathcal{A}$ runs in time $\frac{\mathsf{poly}(|x|)}{\epsilon(x)}$ and outputs w such that $(x, w) \in R$ with probability at least $1 - \frac{\delta(|x|)}{\epsilon(x)}$.

An interactive proof is "knowledge sound", or simply a "proof of knowledge", if has negligible knowledge error δ.

Remark 9. Definition 9 places no restriction on the runtime of the adversary, however, it does not guarantee extraction from an adversary that succeeds with sufficiently small $\epsilon(x)$ such that $\epsilon(x) \leq \delta(|x|)$. For $\mathcal{R}$ in NP, this definition of knowledge soundness implies the alternative formulation of Bellare and Goldreich [BG93], which says that

---

[14]The extractor can run $\mathcal{A}$ for any specified number of steps, inspect the internal state of $\mathcal{A}$, and even rewind $\mathcal{A}$ to a previous state.

the protocol has knowledge error $\delta(|x|)$ if there exists an extractor that succeeds in expected time $\frac{\mathsf{poly}(|x|)}{\epsilon(x)-\delta(|x|)}$. An extractor which succeeds with probability $p = 1 - \frac{\delta(|x|)}{\epsilon(x)}$ in $t = \frac{\mathsf{poly}(|x|)}{\epsilon(x)}$ steps can run repeatedly (for t steps per iteration) on fresh randomness until it obtains a witness for the relation, which it can verify efficiently. It will succeed in an expected $\frac{t}{p} = \frac{\mathsf{poly}(|x|)}{\epsilon(x)-\delta(|x|)}$ steps. Finally, this has been shown to imply another equivalent formulation which requires the extractor to run in $O(\mathsf{poly}(|x|))$ steps and succeed with probability $\frac{\epsilon(x)-\delta(|x|)}{q(|x|)}$ for some polynomial q. It is easy to see this implies the former because such an extractor can be repeated, succeeding in expected time $\frac{q \cdot \mathsf{poly}(|x|)}{\epsilon(x)-\delta(|x|)}$.

Interactive arguments Knowledge soundness holds against unbounded provers. The DARK protocol does not satisfy knowledge soundness because it relies on the computational binding property of cryptographic commitments. Interactive proofs that are only secure against computationally bounded adversaries are called *interactive arguments*. Adapting Definition 9 for arguments is more subtle than simply restricting the runtime of the adversary. The issue comes from the fact that the knowledge soundness definition quantifies the success of the extractor over all inputs x. For example, there could exist an input x that encodes the factorization of an RSA modulus which allows the adversarial prover to break the binding property of commitments that are based on the difficulty of factoring. For this input, the adversarial prover could succeed while the extractor would fail. This particular problem is fixed by requiring the adversary to generate the input x. If the trapdoor is exponentially hard to compute the polynomial time adversary will not be able to embed the trapdoor in x with non-negligible probability. (See Damgård and Fujisaki [DF02] for a broader discussion of these issues).

Witness-extended emulation A property called witness-extended emulation [Lin01] strengthens the knowledge-soundness definition so that the extractor outputs not only a witness but also a simulated transcript of the messages between the prover and verifier. This property is helpful for composability. In particular, if the interactive proof is used as a subprotocol within a larger protocol, it may be necessary in the security analysis to construct a simulator that needs to both obtain the adversary's witness as well as simulate its view in the subprotocol. Fortunately, Lindell [Lin01] proved that every knowledge sound protocol also satisfies witness-extended emulation. Groth and Ishai [GI08] further adapt the definition of witness-extended emulation for interactive arguments with setup (i.e., SRS model). This is the definition we will use in the present work.

Before presenting the definition we will introduce some useful notations. In the SRS model, there is an additional setup algorithm Setup that generates public parameters pp that are common inputs to the prover $\mathcal{P}$ and verifier $\mathcal{V}$. The setup, which may or may not require a trusted party to sample trapdoor secrets, typically generates these parameters based on a security parameter λ necessary for computational security. Without loss of generality, the length of $\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda)$ is at least λ bits. The relation $\mathcal{R}$ being proved may also depend on the setup parameters, denote $\mathcal{R}_{\mathsf{pp}}$. For any prover algorithm $\mathcal{P}^*$ interacting with a verifier algorithm $\mathcal{V}$, which may deviate arbitrarily from the honest prover algorithm $\mathcal{P}$, let $\mathsf{Record}(\mathcal{P}^*, \mathsf{pp}, x, \mathsf{st})$ denote the message transcript between $\mathcal{P}^*$ and $\mathcal{V}$ on shared inputs x and pp and initial prover state st. For $\mathsf{tr} \leftarrow \mathsf{Record}(\mathcal{P}^*, \mathsf{pp}, x, \mathsf{st})$ let $V_{\mathsf{check}}(\mathsf{tr})$ denote the verifier's decision algorithm to accept or reject the transcript. Furthermore, let $\mathcal{E}^{\mathsf{Record}(\mathcal{P}^*, \mathsf{pp}, x, \mathsf{st})}$ denote a machine $\mathcal{E}$ with a transcript oracle for this interaction that can be rewound to any round and run again on fresh verifier randomness.

Definition 10 (Witness-extended emulation [GI08; Lin01])**.** An interactive proof in the SRS model $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ satisfies witness-extended emulation for relation $\mathcal{R}$ if for every deterministic polynomial time $\mathcal{P}^*$ there exists an expected polynomial time emulator $\mathcal{E}$ such that for any non-uniform[15] adversary $\mathcal{A}$ and distinguisher $\mathcal{D}$ that runs in time $\mathsf{poly}(\lambda)$ the following condition holds:

$$\Pr\left[\mathcal{D}(\mathsf{tr}) = 1 : \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda) \\ (x, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{pp}) \\ \mathsf{tr} \leftarrow \mathsf{Record}(\mathcal{P}^*, \mathsf{pp}, x, \mathsf{st}) \end{array}\right] \approx_\lambda$$

$$\Pr\left[\begin{array}{c} \mathcal{D}(\mathsf{tr}) = 1 \text{ and} \\ V_{\mathsf{check}}(\mathsf{tr}) = 1 \Rightarrow (x, w) \in \mathcal{R}_{\mathsf{pp}} \end{array} : \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda) \\ (x, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{pp}) \\ (\mathsf{tr}, w) \leftarrow \mathcal{E}^{\mathsf{Record}(\mathcal{P}^*, \mathsf{pp}, x, \mathsf{st})}(\mathsf{pp}, x) \end{array}\right]$$

where $X \approx_\lambda Y$ denotes that $|X - Y| \le \mathsf{negl}(\lambda)$.

---

[15]A non-uniform adversary may run a different algorithm for each input length.

Lemma 10 (Lindell [Lin01]). *Any proof of knowledge for relation $\mathcal{R}$ also satisfies witness-extended emulation for $\mathcal{R}$.*

Lemma 11. *Let $\mathcal{R}$ denote any NP relation. Given a commitment scheme $\mathsf{com} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open})$, for any $\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda)$ let $\mathcal{R}'(\mathsf{pp})$ denote the relation:*

$$
\mathcal{R}'(\mathsf{pp}) = \left\{
\begin{array}{c}
(\mathsf{x}, \mathsf{w}) : \mathcal{R}(\mathsf{x}, \mathsf{w}) = 1 \\
\vee \\
[\mathsf{w} = (C, \sigma_1, \sigma_2) \ \wedge \ \sigma_1 \neq \sigma_2 \ \wedge \ \mathsf{Open}(\mathsf{pp}, C, \sigma_1) = \mathsf{Open}(\mathsf{pp}, C, \sigma_2) = 1]
\end{array}
\right\}
$$

Let $\Pi(\mathsf{pp})$ denote the interactive protocol between $\mathcal{P}$ and $\mathcal{V}$ parameterized by the setup parameter pp. If for all $\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda)$ the protocol $\Pi(\mathsf{pp})$ is a proof of knowledge (Definition 9) for $\mathcal{R}'(\mathsf{pp})$ then the tuple $(\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ as an interactive proof with setup satisfies witness-extended emulation (Definition 10) for $\mathcal{R}$.

Proof. By Lemma 10 a knowledge sound interactive proof for $\mathcal{R}'(\mathsf{pp})$ also satisfies witness-extended emulation for $\mathcal{R}'(\mathsf{pp})$. It remains to show that this implies witness-extended emulation for $\mathcal{R}$. It suffices to show that:

$$
\Pr\left[
\begin{array}{c}
\mathcal{D}(\mathsf{tr}) = 1 \text{ and} \\
V_{\mathsf{check}}(\mathsf{tr}) = 1 \Rightarrow (x, w) \in \mathcal{R}
\end{array}
:
\begin{array}{c}
\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda) \\
(x, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{pp}) \\
(\mathsf{tr}, w) \leftarrow \mathcal{E}^{\mathsf{Record}(\mathcal{P}^*, \mathsf{pp}, x, \mathsf{st})}(\mathsf{pp}, x)
\end{array}
\right] \approx_\lambda
$$

$$
\Pr\left[
\begin{array}{c}
\mathcal{D}(\mathsf{tr}) = 1 \text{ and} \\
V_{\mathsf{check}}(\mathsf{tr}) \Rightarrow (x, w) \in \mathcal{R}'(\mathsf{pp})
\end{array}
:
\begin{array}{c}
\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda) \\
(x, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{pp}) \\
(\mathsf{tr}, w) \leftarrow \mathcal{E}^{\mathsf{Record}(\mathcal{P}^*, \mathsf{pp}, x, \mathsf{st})}(\mathsf{pp}, x)
\end{array}
\right]
$$

The difference between these two probabilities is bounded by the probability, over the distribution on the right side of the equation, that $(\mathsf{x}, \mathsf{w}) \in \mathcal{R}'(\mathsf{pp})$ but $(\mathsf{x}, \mathsf{w}) \notin \mathcal{R}$. This event implies that w encodes a break to the commitment scheme with parameters pp. Since $\mathcal{A}$, P^*, Setup and $\mathcal{E}$ all run in time $\mathsf{poly}(\lambda)$ this occurs with probability at most $\mathsf{negl}(\lambda)$ over randomly sampled $\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda)$ by the computational binding property of the commitment scheme. More precisely, supposing that the difference between these two probabilities is $\epsilon(\lambda)$, then we can use $\mathcal{A}$, P^*, and $\mathcal{E}$ to construct an algorithm $\mathcal{A}'$ which on input $\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda)$ simulates $(x, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{pp})$ and $(\mathsf{tr}, w) \leftarrow \mathcal{E}^{\mathsf{Record}(\mathcal{P}^*, \mathsf{pp}, x, \mathsf{st})}(\mathsf{pp}, x)$ returning w such that:

$$
\Pr\left[
\begin{array}{c}
w = (C, \sigma_1, \sigma_2) \\
\wedge \\
\sigma_1 \neq \sigma_2 \\
\wedge \\
\mathsf{Vf}_{\mathsf{pp}}(C, \sigma_1) = \mathsf{Vf}_{\mathsf{pp}}(C, \sigma_2) = 1
\end{array}
:
\begin{array}{c}
\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda) \\
w \leftarrow \mathcal{A}'(\mathsf{pp})
\end{array}
\right] = \epsilon(\lambda)
$$

If $\epsilon(\lambda)$ is non-negligible this contradicts the binding property of the commitment scheme. $\square$

Zero knowledge We recall the definition of *honest verifier zero-knowledge* (HVZK) for interactive proofs. HVZK only considers simulating the view of a verifier that follows the protocol honestly. The Fiat-Shamir transform compiles public-coin proofs that have HVZK into non-interactive proofs that have statistical zero-knowledge (for malicious verifiers).

Definition 11 (HVZK for interactive arguments). *Let $\mathsf{View}_{\langle \mathcal{P}(x, w), \mathcal{V}(x) \rangle}$ denote the view of the verifier in an interactive protocol on common input x and prover witness input w. The interactive protocol has δ-statistical honest verifier zero-knowledge if there exists a probabilistic polynomial time algorithm $\mathcal{S}$ such that for every $(x, w) \in \mathcal{R}$, the distribution $\mathcal{S}(x)$ is δ-close to $\mathsf{View}_{\langle \mathcal{P}(x, w), \mathcal{V}(x) \rangle}$ (as distributions over the randomness of $\mathcal{P}$ and $\mathcal{V}$).*

C   Proof of Almost Special Soundness Theorems

Theorem 4 (AMSS implies Knowledge Soundness). *If a μ-round interactive proof for a relation $\mathcal{R}$ with λ-bit challenges, $\mu \in O(\log(\lambda + |\mathsf{x}|))$, and verifier decision algorithm runtime $t_V \in \mathsf{poly}(|\mathsf{pp}|, |\mathsf{x}|, \lambda)$ on input $\mathsf{x} \in \mathcal{L}_{\mathcal{R}}$ and parameters $\mathsf{pp} \leftarrow \Gamma.\mathsf{Setup}(1^\lambda)$ is $(k^{(\mu)}, \delta, \Gamma, \phi)$-almost-special-sound then for $\delta'(\lambda) = 2\lambda(k+1)^\mu(\mu + t_V) \cdot \max(\delta(\lambda), k \cdot 2^{-\lambda}) + 2^{-\lambda}$ it is δ'-knowledge sound for the modified relation:*

$$
\mathcal{R}'(\mathsf{pp}) = \{(\mathsf{x}, \mathsf{w}) : \mathcal{R}(\mathsf{x}, \mathsf{w}) = 1 \ \vee \mathsf{w} \in \mathcal{L}_{\mathsf{break}}(\mathsf{pp})\}
$$

where

$$\mathcal{L}_{\mathsf{break}}(\mathsf{pp}) = \{(\mathcal{C}, \sigma_1, \sigma_2, \iota) : \iota \in \mathcal{I} \wedge \sigma_1 \neq \sigma_2 \wedge \mathsf{Open}(\mathsf{pp}_\iota, \mathcal{C}, \sigma_1) = \mathsf{Open}(\mathsf{pp}_\iota, \mathcal{C}, \sigma_2) = 1\}$$

Proof. Suppose we have a protocol that is $(k^{(\mu)}, \delta)$-almost-special-sound with challenge space $\mathcal{X}$ of size 2^λ for some negligible function $\delta : \mathbb{N} \to \mathbb{R}$. We will make use of algorithms Extract, Extend, and Break and their properties that are guaranteed to exist by the definition of almost-special-soundness (Definition 2).

For any node ν of a $(k+1)$-ary transcript tree let S_ν^* denote the *left k-ary subtree* rooted at ν defined by a breadth first search from ν that visits only the first k children of each node reached (i.e., prunes the rightmost branch from each node of the complete $(k+1)$-ary subtree S_ν).

In Definition 12, we define an algorithm $\mathsf{TreeExtract}(\ell_\nu, \nu, L_{S_\nu}, F(L)_{S_\nu})$ that operates on a labeled subtree of a $(k+1)$-ary transcript tree that has depth μ, where ℓ_ν is the level of ν, L defines a $(k+1)$-ary transcript tree and $F(L)$ is a localized transcript tree labeling, while L_{S_ν} and $F(L)_{S_\nu}$ are their respective restrictions to the subtree S_ν rooted at ν.[16] This algorithm uses the algorithm Extract as a subroutine. There are three potential outcomes of running $\mathsf{TreeExtract}(k, \ell_\nu, \nu, L_{S_\nu}, F(L)_{S_\nu})$:

1. If any of the internal calls to Extract return break, i.e. a break of a commitment scheme, the algorithm immediately outputs that break as a valid witness.

2. It returns openSubtree, which contains valid openings of all the commitment labels $F(L)$ assigned to nodes in S_ν valid for their corresponding commitment scheme indices, including the label $\mathcal{C}_\nu$ on node ν with respect to index ι_ν. In this case we say it "succeeds".

3. It returns $\bot$.

The TreeExtract algorithm is not guaranteed to succeed. In particular, the internal calls to Extract are only guaranteed to either succeed or break the commitment scheme when the openings of subtrees satisfy predicate ϕ_a and the challenge labels are distinct within the pruned subtrees S_ν^*. Definition 12 also defines $\mathsf{TreeExtract}^*(k, \ell_\nu, \nu, L_{S_\nu^*}), F(L)_{S_\nu^*})$, an algorithm that only extracts openings of the commitments in $F(L)_{S_\nu^*}$ and returns an opening of C_ν as a commitment for index ι_ν. This runs similarly to TreeExtract, but it is only a function of nodes present in the left k-ary subtree S_ν^*. While it is possible that TreeExtract fails and TreeExtract* succeeds, they will always output the same opening of C_ν in the event that both succeed.

Let $\mathsf{size}(k, \mu) = \frac{k^{\mu+1} - 1}{k-1}$, which is the number of nodes is a k-ary depth μ tree. Given any μ-round protocol that satisfies $(k^{(\mu)}, \delta)$-almost-special-soundness, setting $N = \mathsf{size}(k+1, \mu)$ we will define a collection of predicates $\{\pi_\nu : \nu \in [1, N]\}$ for the nodes of a k-ary transcript tree with post-order labeling L, such that each π_ν is a function of the partial labeling L_{ν^*} and a $\mu - \ell_\nu$-length challenge vector $\mathbf{r} \in \mathcal{X}^{\mu - \ell_\nu}$, where ℓ_ν is the level of node ν in the tree. Recall that $\nu^* < \nu$ is the node of highest index smaller than ν that is not a member of the subtree of ν, and L_{ν^*} are the labels L assigned to $\mathsf{pred}([1, \nu^*])$, the superset of $[1, \nu^*]$ that includes all its predecessors. Let ω denote the parent node of ν. Note that if ν is not a first child then $\omega \in \mathsf{pred}([1, \nu^*])$. Similarly, if ν' is a left-sibling of ν then $\nu' \in [1, \nu^*]$. So the labels on both ν' and ω would be included in L_{ν^*}. Finally, for any *localized* transcript tree labeling F, the induced labeling $F(L)_{S_\omega^*}$ is a function of L_{ν^*}. This is because the labels that $F(L)$ assigns to S_ω^* are computed from the labels that L assigns to S_ω^* and its predecessors $\mathsf{pred}(S_\omega^*) \subseteq \mathsf{pred}([1, \nu^*])$.

The predicate $\pi_\nu(L_{\nu^*}, \mathbf{r})$ is defined as follows:

- If ν has no left-sibling, then π_ν always returns 1.

- If ν has $0 < i < k$ left-siblings (i.e., it is neither the first nor last child) then $\pi_\nu(L_{\nu^*}, \mathbf{r}) = 1$ iff the challenge label $L(\nu)_0$ assigned to ν is distinct from the challenge labels assigned to its i left-siblings.

- If ν has no right-sibling (i.e., is rightmost child) then let (m_ω, o_ω) denote the output returned by $\mathsf{TreeExtract}^*(k, \ell_\omega, C_\omega, L_{S_\omega^*}, F(L)_{S_\omega^*})$ if successful in returning a valid opening of the commitment C_ω with respect to its index ι_ω. We say it "fails" otherwise, including potentially returning a break of the commitment scheme. Let m' denote the last message in the output list of $\mathsf{Extend}(\ell_\omega, m_\omega, \mathbf{r})$ and finally:

---

[16]In addition to commitment/challenge labels, we assume information about the commitment scheme indices used by $F(L)$ for each node in S_ν are also implicit inputs.

$$\pi_\nu(L_{\nu^*}, \mathbf{r}) = \begin{cases} 1 & \text{if } \mathsf{TreeExtract}^*(k, \ell_\omega, \omega, L_{S_\omega^*}, F(L)_{S_\omega^*}) \text{ fails} \\ 1 & \text{if } \phi_a(\ell_\omega, m_\omega) = 1 \\ 1 & \text{if } \phi_a(\ell_\omega, m_\omega) = 0 \wedge \phi_a(\mu, m') = 0 \\ 0 & \text{otherwise} \end{cases}$$

As remarked above, while $\mathsf{TreeExtract}$ operates on the entire $(k{+}1)$-ary subtree of labels rooted at ω, the algorithm $\mathsf{TreeExtract}^*$ takes as input only the labelings of the left k-ary subtree S_ω^*, i.e. $L_{S_\omega^*} \subseteq L_{\nu^*}$, and $F(L)_{S_\omega^*}$, which can all be derived from L_{ν^*}. By the definition of $(k^{(\mu)}, \delta)$-almost-special-soundness (Definition 2, pt. 5), for any $\nu \in [0, N)$ that has no right-sibling and any L_{ν^*}:

$$\mathbb{P}_{\mathbf{r} \leftarrow \mathcal{X}^{\mu - \ell_\nu}}[\pi_v(L_{\nu^*}, \mathbf{r}) = 1] \geq 1 - \delta(\lambda)$$

If ν has $0 < i < k$ left-siblings then by a union bound:

$$\mathbb{P}_{\mathbf{r} \leftarrow \mathcal{X}^{\mu - \ell_\nu}}[\pi_v(L_{\nu^*}, \mathbf{r}) = 1] \geq 1 - \frac{i}{2^\lambda}$$

For any $\nu \in [0, N]$ let $\mathsf{lpath}(\nu)_0$ denote the challenge labels $L(\cdot)_0$ along the leftmost branch from ν to a leaf starting with the label $L(\nu)_0$ on ν. By Lemma 9 (Path Predicate Forking Lemma) there is an algorithm $\mathsf{Tree}^{\mathcal{A}}(\mathsf{z})$ that, given a security parameter $\lambda \in \mathbb{N}$, an input $\mathsf{x} \in \mathcal{L}_{\mathcal{R}}$, and oracle access to an adversarial prover $\mathcal{A}$ that causes V to accept on input x with probability ϵ, runs in time at most $t = 2\lambda \cdot \frac{(k+1)^\mu}{\epsilon} \cdot (\mu + t_V)$, where t_V is the worst-case running time of verifier's decision algorithm, and returns with probability at least $1 - t \cdot \max(\delta(\lambda), \frac{k}{2^\lambda}) - 2^{-\lambda}$ a $(k+1)$-ary transcript tree with post-order labeling $L : [1, N] \to \mathcal{X} \times \mathcal{M}$ such that $\pi_v(L_{\nu^*}, \mathsf{lpath}(v)_0) = 1$ for all $v \in [1, N]$.

In particular, L defines a $(k + 1)$-ary transcript tree with the properties:

1. The challenge labels on the first k children of any node are distinct, i.e., if ω has children $\nu_1, ..., \nu_{k+1}$ ordered from left-to-right, then for any $i, j \in [1, k]$ if $i \neq j$ then $L_0(\nu_i) \neq L_0(\nu_j)$.

2. If ν is the $(k + 1)$th child of ω and running $\mathsf{TreeExtract}(k, \ell_\omega, \omega, L_{S_\omega^*}, F(L)_{S_\omega^*})$ at level ℓ_ω returns an opening of the commitment label $\mathcal{C}_\omega = F(L)(\omega)$ to m_ω with respect to index ι_ω such that $\phi_a(\ell_\omega, m_\omega) \neq 1$, then the final output of $\mathsf{Extend}(\ell_\omega, m_\omega, \mathsf{lpath}(\nu)_0)$ is a message m' such that $\phi_a(\mu, m') \neq 1$.

By Lemma 12 there is a deterministic extraction algorithm that takes any L with the above properties and computes a witness w such that $(\mathsf{x}, \mathsf{w}) \in \mathcal{R}'$.

In conclusion, for any adversarial prover that succeeds on input x with probability $\epsilon(x)$, there is a probabilistic extractor that runs in time at most $t = 2\lambda \frac{(k+1)^\mu}{\epsilon(x)}(\mu + t_V)$ and with probability at least $1 - t \cdot \max(\delta(\lambda), \frac{k}{2^\lambda}) - 2^{-\lambda}$ returns a witness for $\mathcal{R}'$. Since $t \in \frac{\mathsf{poly}(|x|, \lambda)}{\epsilon(x)}$ assuming $\mu \in O(\log(\lambda + |x|))$ and $t_V \in \mathsf{poly}(|x|, \lambda)$, this satisfies the definition of δ'-knowledge soundness with $\delta'(\lambda) = 2\lambda(k + 1)^\mu(\mu + t_V) \cdot \max(\delta(\lambda), k \cdot 2^{-\lambda}) + 2^{-\lambda}$, which is a negligible function of λ as long as $\delta(\lambda)$ is negligible.

$\square$

Definition 12 (Tree Extractor). We define an algorithm $\mathsf{TreeExtract}(k, \ell, \nu, C_\nu, L_{S_\nu}, F(L)_{S_\nu})$ where ℓ_ν is the level of ν in a $(k{+}1)$-ary transcript tree with depth μ, L is a labeling that defines a valid transcript tree and $F(L)$ is a localized induced labeling, $\mathcal{C}_\nu = F(L)(\nu)$ is the commitment label on ν for commitment index ι_ν, and and L_{S_ν} and $F(L)_{S_\nu}$ are their restrictions to the $(k{+}1)$-ary subtree S_ν rooted at ν. L_{S_ν} also contains the commitment indices in $\mathcal{I}$ for each node of the subtree S_ν. If $\mathsf{TreeExtract}(k, \ell_\nu, \nu, L_{S_\nu}, F(L)_{S_\nu})$ fails it returns $\bot$. Otherwise, it succeeds and returns $\mathsf{openSubtree}$, which contains openings of all the commitment labels $F(L)$ assigned to nodes in S_ν with respect to their corresponding commitment indices in $\mathcal{I}$, and also an opening (m, o) for the root. The algorithms runs as follows:

- For all leaf nodes u, if the opening of (m_u, o_u) of $\mathcal{C}_u = F(L)(u)$ with respect to index ι_u does not satisfy $\phi_a(\mu, m_u) = 1$ then abort and output $\bot$, otherwise output (m_u, o_u) and continue.

- For each node $\omega \in S_\nu$ on the second to last level with commitment label $\mathcal{C}_\omega = F(L)(\omega)$ with respect to index ι_ω, set $\mathsf{openLeaves}$ to include the first k leaves of ω along with their openings and commitment indices. If the challenge labels on these k leaves given by L are not all distinct then abort and return $\bot$. Otherwise run $\mathsf{Extract}(\mu - 1, \omega, C_\omega, \mathsf{openLeaves})$ to get an opening (m_ω, o_ω) of $\mathcal{C}_\omega$ with respect to ι_ω. If $\phi(\mu - 1, m_\omega) \neq 1$ then abort and return $\bot$.

- Continue iteratively: once openings for all commitment labels of all subtrees rooted at the ith level have been computed, for each node ω on level $i - 1$ with label $C_\omega = F(L)(\omega)$ with respect to ι_ω, if its first k children do not have distinct challenge labels in L abort and return $\bot$, otherwise run $\mathsf{Extract}(i, \omega, C_\omega, \mathsf{openSubtree}_\omega^*)$ on the commitment label openings $\mathsf{openSubtree}_\omega^*$ of the left k-ary subtree S_ω^* (excluding node ω), which were computed in prior iterations, to get (m_ω, o_ω). If $\phi_a(i, m_\omega) \neq 1$ abort and return $\bot$.

Finally, $\mathsf{TreeExtract}^*(k, \ell_\nu, \nu, C_\nu, L_{S_\nu^*})$ denotes the algorithm that only extracts openings of the commitments in $L_{S_\nu^*}$ and returns an opening of C_ν. This runs exactly like $\mathsf{TreeExtract}$ except that it only iterates over nodes that are present in the left k-ary subtree S_ν^*.

The $\mathsf{TreeExtract}$ algorithm is not guaranteed to succeed. In particular, the internal calls to $\mathsf{Extract}$ are only guaranteed to succeed when the openings of subtrees satisfy predicate ϕ_a and the challenge labels are distinct within the pruned subtrees S_ν^*. If any internal step fails then $\mathsf{TreeExtract}$ outputs $\bot$. While it is possible that $\mathsf{TreeExtract}$ fails and $\mathsf{TreeExtract}^*$ succeeds, it is easy to see that they output the same opening of the commitment label C_ν on node ν assuming both succeed. Furthermore, while $\mathsf{TreeExtract}$ operates on a labeling of a $(k + 1)$-ary transcript tree, the internal calls to $\mathsf{Extract}$ run on labelings of k-ary transcript trees because it is defined for a protocol that is $(k^{(\mu)}, \delta)$-almost-special-sound. The reason we always pass the labeling/opening of the left k-ary subtree (as opposed to an arbitrary k-ary subtree) to $\mathsf{Extract}$ is to ensure that the opening of C_ν included in the output of $\mathsf{TreeExtract}(k, \ell_\nu, \nu, L_{S_\nu}, F(L)_{S_\nu})$ is a function of only the labels on the left k-ary subtree S_ν^* and its predecessors, and in particular is computed independently from any of the labels assigned to the (rightmost) subtree rooted at the $(k + 1)$th (rightmost) child of ν. This fact is used in the proof of Theorem 4.

Definition 13 (Predicate Special Soundness). Let ρ denote any binary predicate that takes as input any k-ary μ-depth transcript tree. A μ-round public coin interactive proof for a relation $\mathcal{R}$ with λ-bit challenges is $(k^{(\mu)}, \rho)$-**special sound** if there exists a deterministic extraction algorithm $\mathcal{E}$ that takes as input an instance $\mathsf{x} \in \mathcal{L}_\mathcal{R}$, any k-ary forking transcript tree rooted at x with labeling L such that $\rho(L) = 1$, and returns a witness w such that $(\mathsf{x}, \mathsf{w}) \in \mathcal{R}$ in time $\mathsf{poly}(\lambda, k^\mu)$.

Setting $\rho = 1$, i.e. the trivial predicate that is always true, recovers the standard definition of $k^{(\mu)}$-special soundness. Recall that we defined a *forking* transcript tree (Section B.3) as a transcript tree in which the challenge labels on edges that share the same parent node are distinct.[17]

Lemma 12. Let $\Pi(\mathsf{pp})$ denote a $(k^{(\mu)}, \delta, \Gamma, \phi)$-almost-special-sound protocol for a relation $\mathcal{R}$ and any $\delta \in [0, 1]$, parametrized by $\mathsf{pp} \leftarrow \Gamma.\mathsf{Setup}(1^\lambda)$. Define the binary predicate ρ as a function of a $(k + 1)$-ary μ-depth forking transcript tree given by labeling L, which uses the algorithm $\mathsf{TreeExtract}$ from Definition 12, the algorithm Extend and localized transcript tree labelling F from Definition 2 and returns 1 iff the following condition holds:

> For any node ω with $(k + 1)$th child ν, if $\mathsf{TreeExtract}(k, \ell_\omega, \omega, L_{S_\omega^*}, F(L)_{S_\omega^*})$ returns an opening (m_ω, o_ω) of the label $C_\omega = F(L)(\omega)$ on node ω (at level ℓ_ω) with respect to commitment index $\iota_\omega \in \mathcal{I}$ such that $\phi_a(\ell_\omega, m_\omega) \neq 1$, then the final output of $\mathsf{Extend}(\ell_\omega, m_\omega, \mathsf{lpath}(\nu)_0)$ is a message m' such that $\phi_a(\mu, m') \neq 1$.

$\Pi(\mathsf{pp})$ is $((k + 1)^{(\mu)}, \rho)$-special sound for the relation $\mathcal{R}'(\mathsf{pp})$ defined in Theorem 4.

Remark 10. The value of δ does not affect $((k + 1)^\mu, \rho)$-special soundness. The value of δ affects the runtime of the extraction algorithm that is able to generate a transcript tree satisfying the predicate ρ (in Theorem 4).

Proof. We will argue that, assuming the $(k+1)$-ary forking transcript tree has property ρ, for any $\omega \in [1, N]$, either $\mathsf{TreeExtract}(k, \ell_\omega, \omega, L_{S_\omega}, F(L)_{S_\omega})$:

1. returns a break of the commitment scheme

---

[17] We could have defined predicate special soundness in an even more general way such that the forking property of the tree is not required, yet can be encapsulated in the predicate. However, this would not be useful for our present work and less convenient for notational purposes.

2. returns a subtree openSubtree of openings of the commitment labels in $F(L)_{S_\omega}$ satisfying ϕ_a (i.e., each opening of a label C_ω to m_ω for a node ω on level ℓ_ω satisfies $\phi_a(\ell_\omega, m_\omega) = 1$)

3. or it returns openSubtree and there is an efficient algorithm that uses openSubtree and L to break the commitment scheme.

Step 1: Suppose that ω is a node of highest level ℓ_ω for which TreeExtract fails . In other words, for any node of higher level than ℓ_ω, the output of TreeExtract is a subtree opening that satisfies ϕ_a. This means that all the openings of internal (non-root) nodes of the subtree labeling $F(L)_{S_\omega}$ computed while running TreeExtract on ω satisfy ϕ_a. Furthermore, L has the property that all challenge labels on the first k siblings are distinct.

Step 2: By the definition of almost-special-soundness and the hypothesis in **Step 1**, the algorithm TreeExtract$(k, \ell_\omega, \omega, L_{S_\omega}, F(L)_{S_\omega})$ either returns a break of the commitment scheme or succeeds in returning openSubtree consisting of the openings of $F(L)_{S_\omega}$ such that the openings of all internal (non-root) nodes satisfy ϕ_a, and the opening (m_ω, o_ω) of the subtree root ω satisfies $\phi_b(\ell_\omega, m_\omega) = 1$. The opening (m_ω, o_ω) is also identical to the output of TreeExtract$^*(k, \ell_\omega, \omega, L_{S_\omega^*}, F(L)_{S_\omega^*})$.

Step 3: Suppose that TreeExtract$(k, \ell_\omega, \omega, L_{S_\omega}, F(L)_{S_\omega})$ does not return a break of the commitment scheme (i.e., returns a subtree opening as described in Step 2). Let ν denote the rightmost child of ω. By hypothesis, if $\phi_a(\ell_\omega, m_\omega) = 0$ then the final output of Extend$(\ell_\omega, m_\omega, \mathsf{lpath}(\nu))$ is a message m' such that $\phi_a(\mu, m') = 0$. However, this implies that m' must be distinct from the label $F(L)$ assigns to the leaf node of the rightmost branch extending from ν. Let $v_1, ..., v_\mu$ denote the nodes along the root-to-leaf path passing through node ω and ending with its leftmost branch so that $\mathsf{lpath}(\nu) = (F(L)(v_{\ell_\nu}), ..., F(L)(v_\mu))$. For each $i \in [1, \mu]$ let $\hat{C}_i = L(v_i)_1$ and $\hat{\mathbf{C}} = (\hat{C}_1, ..., \hat{C}_\mu)$ with corresponding commitment indices $\vec{\iota} = (\iota_1, ..., \iota_\mu)$. Finally, since $\phi_b(\ell_\omega, m_\omega) = 1$ and openSubtree contains openings $(m_{\ell_\nu}, o_{\ell_\nu}), ... (m_\mu, o_\mu)$ of the commitments $\hat{C}_{\ell_\nu}, ... \hat{C}_\mu$ with respect to their corresponding indices $\vec{\iota}$ that all satisfy predicate $\phi_a(i, m_i) = 1$, if $m' \neq m_\mu$ then by the definition of almost-special-soundness this implies that Break$(\ell_\omega, m_\omega, \mathsf{lpath}(\nu), \vec{\iota}, \hat{\mathbf{C}}, (m_{\ell_\nu}, o_{\ell_\nu}), ... (m_\mu, o_\mu))$ outputs a conflicting opening of some commitment label in $\hat{\mathbf{C}}$.

Let $\mathcal{C} = F(L)(N)$ denote the transcript tree root commitment label. The extractor uses TreeExtract to either obtain a break of the commitment scheme or openTree, which includes an opening for $\mathcal{C}$. If every opening in openTree satisfies predicate ϕ_a then it runs Extract$(\mathsf{x}, \mathsf{openTree})$ to obtain witness w satisfying $R(\mathsf{x}, \mathsf{w}) = 1$. Otherwise, it uses the Break algorithm (as described in the previous step) to output an index $\iota \in \mathcal{I}$ and conflicting openings of a commitment with respect to ι, which is a witness for $R'(\mathsf{pp})$.

$\square$

D   Fiat-Shamir Transform of Almost-Special-Sound Protocols

Recently, [Wik21; AFK21] showed that for μ-round special sound protocols the non-interactive Fiat-Shamir transform of these protocols only suffers a security loss that is linear in the number of queries the adversary makes to the random oracle. These proofs do not directly apply to almost-special-sound protocols. In particular, in a non-interactive protocol, we cannot guarantee that the challenges on $\mathsf{lpath}(\nu)$ are mutually independent. With the FS transform each challenge is a deterministic function of the partial transcript of prover messages preceding the challenge. If the adversary has some degree of freedom in choosing each prover message without altering prior round messages, then it may be able to sample each challenge conditional on previous challenges, called *grinding*. In the case of analyzing DARK, we were able to bound the probability that $f(x_1, \ldots, x_\mu) \equiv 0 \bmod N$ for *independently* sampled $x_1, \ldots, x_\mu$. If the adversary can grind challenges then it can for each challenge x_i ensure that $f(x_1, \ldots, x_i, X_{i+1}, \ldots, X_\mu) \equiv 0 \bmod N_i$ where N_i is a factor of N of size roughly $N^{\frac{1}{\mu}}$. The proof of Theorem 5 relies on the fact that $\mathbb{P}_{(\alpha_{i+1}, \ldots, \alpha_\mu) \leftarrow [0, 2^\lambda)^{\mu - i}}[g(\alpha_{i+1}, ..., \alpha_\mu) \equiv 0 \bmod N] = \delta$ is negligible for sufficiently large N. Analyzing a grinding adversary of the non-interactive protocol that makes at most T queries to the random oracle corresponds

to analyzing the probability that for any set of T values

$$\mathbb{P}_{S=\{\alpha_{i,j}\}_{i\in[\mu],j\in[T]}\leftarrow[0,2^\lambda)^{\mu\cdot T}}\exists(j_1,\ldots,j_\mu) \text{ s.t. } [g(\alpha_{1,j_1},...,\alpha_{\mu,j_\mu}) \equiv 0 \bmod N]$$

which by a union bound is less than $T^\mu \cdot \delta$.

However, what conditions might be sufficient to prevent an adversary from grinding challenges? If for every partial transcript, there is a unique next prover message that leads to an accepting transcript then the adversary would not have the opportunity to grind. In fact, it suffices for this to be computationally hard. We will show that this is the case for any protocol satisfying the definition of almost-special soundness with respect to a *computationally unique* commitment scheme, which we define. This implies that the messages in each round are commitments from a scheme where it is computationally difficult to come up with two distinct commitments to the same message (note this implies the scheme cannot be hiding). The DARK protocol has precisely this property. Finally, we prove that the Fiat-Shamir transform of these almost-special-sound protocols is secure.

Definition 14 (Random Oracle)**.** In our version of the random oracle model, all random oracle algorithms have black-box access to a shared random function $\mathcal{H} : \mathcal{M}^{\leq u} \to \mathcal{X}$ where $\mathcal{M}^{\leq u}$ consists of all vectors $(m_1,...,m_i) \in \mathcal{M}^i$ for each $i \leq u$. $\mathcal{H}$ assigns to each of the $\frac{|\mathcal{M}|^{u+1}-1}{|\mathcal{M}|-1}$ unique elements of $\mathcal{M}^{\leq u}$ an output independently and uniformly distributed in $\mathcal{X}$. Random oracles may be assigned indices from a set $\mathcal{I}$, where for any distinct $i,j \in \mathcal{I}$ the oracles $\mathcal{H}_i$ and $\mathcal{H}_j$ are independently distributed random functions. For any $k < u$ and fixed $\mathbf{m} = (m_1,...,m_k)$ we will use the notation $\mathcal{H}_{\mathbf{m}} : \mathcal{M}^{\leq u-k} \to \mathcal{X}$ where $\mathcal{H}_{\mathbf{m}}(m'_1,...,m'_i) = \mathcal{H}(\mathbf{m},m'_1,...,m'_i)$ for any $i \leq u-k$. This is equivalent to a random oracle family indexed by $\mathcal{M}^k$.

A *Q-query random oracle algorithm* is an algorithm that makes at most Q queries to the random oracle.

Definition 15 (Non-interactive Proof of Knowledge in RO)**.** An non-interactive protocol $\Pi = (\mathcal{P},\mathcal{V})$ between a prover $\mathcal{P}$ and verifier $\mathcal{V}$ in the random oracle model (i.e., with shared oracle access to the random function $\mathcal{H}$) is a proof of knowledge for a relation $\mathcal{R}$ with knowledge error $\delta : \mathbb{N}^2 \to [0,1]$ if the following properties hold:

- Perfect Completeness: for all $(x,w) \in \mathcal{R}$

$$\mathbb{P}\left[\mathcal{V}^{\mathcal{H}}(x,\pi) = 1 : \pi \leftarrow \mathcal{P}^{\mathcal{H}}(x,w)\right] = 1$$

- δ-Knowledge Soundness: There exists a polynomial $\mathsf{poly}(\cdot)$ and a probabilistic oracle machine $\mathcal{E}$ called the *extractor* such that given oracle access to any Q-query random oracle algorithm $\mathcal{A}$ and any input $x \in \mathcal{L}_R$ the following holds[18]: if

$$\mathbb{P}\left[\mathcal{V}^{\mathcal{H}}(x,\pi) = 1 : \pi \leftarrow \mathcal{A}^{\mathcal{H}}(x)\right] = \epsilon(x)$$

then $\mathcal{E}^{\mathcal{A}}(x)$ outputs w such that $(x,w) \in \mathcal{R}$ in an expected $\frac{\mathsf{poly}(|x|)}{\epsilon(x)-\delta(|x|,Q)}$ number of steps. In the course of running with black-box access to $\mathcal{A}$, $\mathcal{E}^{\mathcal{A}}(x)$ implements the random oracle for $\mathcal{A}$, i.e. it intercepts all queries that $\mathcal{A}$ makes to $\mathcal{H}$ and simulates the response.

Π is called "knowledge sound" or a "proof of knowledge" for $\mathcal{R}$ if for all Q polynomial in $|x|$ the knowledge error $\delta(|x|,Q)$ is a negligible function of $|x|$.

Definition 16 (Non-interactive argument of knowledge in SRS/RO)**.** A non-interactive proof system $\Pi = (\mathsf{Setup}, \mathcal{P}^{\mathcal{H}}, \mathcal{V}^{\mathcal{H}})$ with setup procedure $\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda)$ in the random oracle model, where $\mathcal{P}^{\mathcal{H}}$ and $\mathcal{V}^{\mathcal{H}}$ are given shared access to both the random oracle $\mathcal{H}$ and the parameters pp (where $|\mathsf{pp}| \geq \lambda$), is an argument of knowledge for relation $\mathcal{R}$ with knowledge error $\mathsf{err} : \mathbb{N}^2 \to [0,1]$ if there exists a polynomial time extractor $\mathcal{E}$ such that for any non-uniform polynomial time adversary $\mathcal{A}$ and deterministic Q-query polynomial time prover P^* the following holds:[19]

$$\mathbb{P}\left[\begin{array}{c} (x,w) \in \mathcal{R} \text{ and} \\ \mathcal{V}^{\mathcal{H}}(\mathsf{pp},x,\pi) = 1 \end{array} : \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{Setup}(\lambda) \\ (x,\mathsf{st}) \leftarrow \mathcal{A}(\mathsf{pp}) \\ \pi \leftarrow P^*(\mathsf{st}) \\ w \leftarrow \mathcal{E}^{P^*(\mathsf{st})}(\mathsf{pp},x) \end{array}\right] \geq \mathbb{P}\left[\mathcal{V}^{\mathcal{H}}(\mathsf{pp},x,\pi) = 1 : \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{Setup}(\lambda) \\ (x,\mathsf{st}) \leftarrow \mathcal{A}(\mathsf{pp}) \\ \pi \leftarrow P^*(\mathsf{st}) \end{array}\right] - \mathsf{err}(\lambda,Q)$$

---

[18]The algorithm $\mathcal{A}$ may explicitly hardcode a witness or may not have one, so no witness is given to $\mathcal{A}$ as input.

[19]This definition says that there is overwhelming intersection between the event where the adversary generates an input x and corresponding proof π that convinces the verifier, and the event where the extractor succeeds in obtaining a witness from the input x generated by the adversary. This not only ensures that extraction succeeds with close to the same probability of the adversary's success over randomly sampled parameters, but also excludes the pathological case that both the adversary and extractor succeed with noticeable probability on disjoint sets of inputs. This definition is also equivalent to fixing the transcript distinguisher in the definition of witness-extended emulation (Definition 10) to be the verifier decision algorithm. In WEE the transcript distinguisher could be arbitrary, which is a stronger property important for simulation analysis.

Definition 17 (FS Transform). For any μ-round public-coin interactive proof Π the FS transform $\Pi_{FS}^{\mathcal{H}}$ of Π with respect to the random oracle $\mathcal{H}$ is a non-interactive proof in the random oracle model which on public input x simulates the interactive protocol Π by replacing each ith round public-coin challenge, for $i \in [1, \mu]$, with $\mathcal{H}(\mathsf{x}, m_1, ..., m_i)$, where $m_1, ..., m_i$ denote the first i prover messages.

Lemma 13 (FS for special-sound multiround protocols [AFK21; Wik21]). For any μ-round interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ for relation $\mathcal{R}$ and its FS transform $\Pi_{FS} = (\mathcal{P}^{\mathcal{H}}, \mathcal{V}^{\mathcal{H}})$ with random oracle $\mathcal{H}$, there exists a random oracle algorithm Tree which given blackbox access to any Q-query deterministic prover algorithm $\mathcal{P}^*$, input $x \in \mathcal{L}_R$, and $k \in \mathbb{N}$ makes at most $Q + \mu$ queries to $\mathcal{H}$ and returns a k-ary forking transcript tree for Π in expected time $k^{\mu} + Q \cdot (k^{\mu} - 1)$ and succeeds with probability $\frac{\epsilon(x) - (Q+1) \cdot \kappa}{(1-\kappa)}$ where $\kappa = 1 - (1 - \frac{k-1}{2^{\lambda}})^{\mu}$ and $\epsilon(x)$ is the probability (over $\mathcal{H}$) that $\mathcal{P}^*$ outputs a non-interactive proof for x that $\mathcal{V}^{\mathcal{H}}$ accepts. Moreover, the transcript tree satisfies additional properties:

- Every root-to-leaf labelled path (i.e., transcript included in the tree) matches the output of $\mathcal{P}^{*\mathcal{H}'}(x)$ with a partially fresh random oracle $\mathcal{H}'$, and thus has the format of a valid Π_{FS} proof with respect to $\mathcal{H}'$.

- For any node ν, the labels L_{S_ν} on the subtree S_ν are generated by a $(Q + \mu)$-query random oracle algorithm independently from all labels L_{ν^*}, i.e. labels computed on lower indexed nodes that do not belong to S_ν. In particular, if ν is the kth child of ω then L_{S_ν} is independent of the labels on the $(k-1)$-ary left subtree of ω.

Lemma 13 immediately implies that the FS transform of any $k^{(\mu)}$-special sound protocol for $\mathcal{R}$ is knowledge sound in the RO model. However, we need to work a bit harder to apply this lemma to almost-special-sound protocols. We will only be able to show computational knowledge soundness for protocols that are almost-special-sound with respect to a computationally-unique commitment scheme. The DARK proof system has this property.

Definition 18 (RO relation hardness). Let $\mathcal{R}^{\mathcal{H}}(\mathsf{pp})$ denote a family of relations parametrized by a random oracle $\mathcal{H}$ and setup $\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda)$ with security parameter λ where $|\mathsf{pp}| \geq \lambda$. $\mathcal{R}^{\mathcal{H}}(\mathsf{pp})$ is $(Q, \epsilon(\lambda))$-hard in the RO model if for any pair of polynomial time random oracle algorithms $\mathcal{A}_1, \mathcal{A}_2$ where $\mathcal{A}_1$ makes at most Q queries to a random oracle $\mathcal{H}'$ (possibly distinct from $\mathcal{H}$) and $\mathcal{A}_2$ makes at most Q queries to random oracle $\mathcal{H}$:

$$\mathbb{P}\left[(\mathsf{x}, \mathsf{w}) \in \mathcal{R}^{\mathcal{H}}(\mathsf{pp}) : \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Setup}(\lambda) \\ \mathsf{x} \leftarrow \mathcal{A}_1^{\mathcal{H}'}(\mathsf{pp}) \\ \mathsf{w} \leftarrow \mathcal{A}_2^{\mathcal{H}}(\mathsf{pp}, \mathsf{x}) \end{array}\right] \leq \epsilon(\lambda)$$

Definition 19 (Computationally Unique Commitments). A deterministic commitment scheme $\Gamma = (\mathsf{Setup}, \mathcal{G}.\mathsf{Commit}, \mathsf{Open})$ is computationally-unique if for any polynomial-time adversary $\mathcal{A}$

$$\Pr\left[b_0 = b_1 = 1 \wedge C_0 \neq C_1 : \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Setup}(1^{\lambda}) \\ (C_0, C_1, x, r_0, r_1, \iota) \leftarrow \mathcal{A}(\mathsf{pp}) \\ \mathsf{pp}_{\iota} = \mathcal{G}(\mathsf{pp}, \iota) \\ b_0 \leftarrow \mathsf{Open}(\mathsf{pp}_{\iota}, C_0, x, r_0) \\ b_1 \leftarrow \mathsf{Open}(\mathsf{pp}_{\iota}, C_1, x, r_1) \end{array}\right] \leq \mathsf{negl}(\lambda)$$

Lemma 14. Any deterministic homomorphic commitment scheme Γ that is binding under rational openings to $\mathcal{W} \subseteq \mathbb{Q}^n$ is computationally unique if the commitment group $\mathbb{G}$ is either prime or is a group of unknown order in which the subgroup order assumption holds for all of $\mathbb{G}$.

Proof. Suppose $C' \neq C \in \mathbb{G}$ and $\mathbf{x}/z \in \mathcal{W}$ and $\mathsf{com}(\mathbf{x}) = z \cdot C$ and $\mathsf{com}(\mathbf{x} \cdot \Delta) = z \cdot \Delta \cdot C$. This implies that $z\Delta \cdot (C' - C) = 0$. In a prime-order group this is impossible because $C' - C \neq 0$. In a group of unknown order, $z\Delta$ is a multiple of the order of $C' - C$, which violates the subgroup order assumption for $\mathbb{G}$. $\square$

Lemma 15. For any $\mu, Q = \mathsf{poly}(\lambda)$, and computationally-unique commitment scheme $\Gamma = (\mathsf{Setup}, \mathcal{G}, \mathsf{Commit}, \mathsf{Open})$, predicate $\rho : \mathcal{M}^2 \times \mathcal{X}^{\mu} \to \{0, 1\}$, and $\mathsf{ext} : \mathcal{M} \times \mathcal{X}^{\leq \mu} \to \mathcal{M}$ where $\forall i, m \in \mathcal{M}$:

$$\mathbb{P}_{\alpha_1, ..., \alpha_{\mu}}[\rho_i(m, m_{\mu}, \alpha_1, ..., \alpha_{\mu}) = 1 : \mathsf{ext}(m, \alpha_1, ..., \alpha_{\mu}) = m_{\mu}] \leq \delta$$

the relation

$$\mathcal{R}^{\mathcal{H}}_{\text{ext}}(\text{pp}, \mu, \rho) = \left\{ (\text{x} = (C, m, o), \text{w} = (\text{tr}, M)) : \begin{array}{c} \text{tr} = (C_1, ..., C_\mu) \\ \forall_{i \in [\mu]} \; \alpha_i = \mathcal{H}(C, C_1, ..., C_{i-1}) \\ M = ((m_1, o_1), ..., (m_\mu, o_\mu)) \\ \text{Open}(\text{pp}, C, m, o) = 1 \\ \text{pp}_i = \mathcal{G}(\text{pp}, \alpha_1, ..., \alpha_i) \\ \forall_{i \in [\mu]} \text{Open}(\text{pp}_i, C_i, m_i, o_i) = 1 \\ \forall_{i \in [\mu]} \text{ext}(m, \alpha_1, ..., \alpha_i) = m_i \\ \rho(m, m_\mu, \alpha_1, ..., \alpha_\mu) = 1 \end{array} \right\}$$

is $(Q, \delta + \text{negl}(\lambda))$-hard in the RO model.

Proof. Consider any pair of polynomial time random oracle algorithms $\text{x} \leftarrow \mathcal{A}_1^{\mathcal{H}'}(\text{pp})$ and $\text{w} \leftarrow \mathcal{A}_2^{\mathcal{H}}(\text{pp}, \text{x})$ that for setup parameter λ make at most $Q = \text{poly}(\lambda)$ queries to their respective oracles $\mathcal{H}'$ and $\mathcal{H}$. For fixed x let $T_{\mathcal{H}}(\text{x}) = (T_1, ..., T_\mu)$ denote a random variable representing the output tr included in w *conditioned on* the event that w satisfies at least all validity criteria other than possibly the last, i.e. $\rho(m, m_\mu, \alpha_1, ..., \alpha_\mu) = 1$. Note that $T_{\mathcal{H}}(\text{x})$ is also dependent on the randomness of the oracle $\mathcal{H}$. Presuming this event occurs with probability at least $\epsilon(\text{x})$, we can repeat $\mathcal{A}_2^{\mathcal{H}}(\text{pp}, \text{x})$ on fresh internal randomness (not changing $\mathcal{H}$) in expectation $1/\epsilon(\text{x})$ times until it outputs w satisfying this event with a particular assignment $\text{tr} = (C_1, ..., C_\mu)$ to the random variable $T_{\mathcal{H}}(\text{x})$. Suppose that for any $i \in [\mu]$ we then reprogrammed $\mathcal{H}$ to an oracle $\mathcal{H}^*$ by sampling new answers to any subset of the queries $\{q_j = (C, C_1, ..., C_j)\}_{i \leq j < \mu}$ but keeping all other queries consistent with $\mathcal{H}$. For $i = \mu$ we do not change the oracle and $\mathcal{H}^* = \mathcal{H}$. Define the random variable $T_{\mathcal{H}^*}(\text{x})$ in the same way for the new oracle $\mathcal{H}^*$. Suppose further that repeating the same experiment with $\mathcal{A}_2^{\mathcal{H}^*}(\text{pp}, \text{x})$ were to return with probability greater than $\delta'(\text{x})$ a vector $\text{tr}' = (C_1', ..., C_\mu') \neq \text{tr}$ where the first distinct index between tr' and tr is some $k \leq i$. For $i = \mu$ we just repeat the same experiment with $\mathcal{H}$. For all $i \in [\mu]$ let $\alpha_i = \mathcal{H}(C, C_1, ..., C_{i-1})$, let $m_i = \text{ext}(m, \alpha_1, ..., \alpha_i)$, let $\alpha_i' = \mathcal{H}^*(C, C_1', ..., C_{i-1}')$, and let $m_i' = \text{ext}(m, \alpha_1', ..., \alpha_i')$. Since the oracle answers to queries C and $\{q_j = (C, C_1, ..., C_\ell)\}_{1 \leq \ell < k}$ have not changed and $(C_1, ..., C_{k-1}) = (C_1', ..., C_{k-1}')$ it follows that $(\alpha_1, ..., \alpha_k) = (\alpha_1', ..., \alpha_k')$ and $(m_1, ..., m_k) = (m_1', ..., m_k')$. The result of these two experiments would thus include openings of the distinct commitments $C_k \neq C_k'$ to the same message $m_k = m_k'$. By computational uniqueness of the commitment scheme, for $\text{pp} \leftarrow \text{Setup}(\lambda)$ and $\text{x} \leftarrow \mathcal{A}_1^{\mathcal{H}'}(\text{pp})$ either $\delta'(\text{x})$ or $\epsilon(\text{x})$ is negligible in λ, as we have shown that it is possible to construct and adversary using $\mathcal{A}_1$ and $\mathcal{A}_2$ that on input pp breaks the uniqueness of the commitment scheme (Definition 19) in expected time $(\frac{1}{\epsilon(\text{x})} + \frac{1}{\delta'(\text{x})}) \cdot \text{poly}(\lambda)$.

We draw two conclusions from this. For $\text{pp} \leftarrow \text{Setup}(\lambda)$ and $\text{x} \leftarrow \mathcal{A}_1^{\mathcal{H}'}(\text{pp})$, if $\mathcal{A}_2^{\mathcal{H}}(\text{pp}, \text{x})$ succeeds in returning w satisfying the aforementioned event (i.e., all criteria except the last predicate) with non-negligible probability then there is a unique vector $(C_1, ..., C_\mu)$ such that:

(a) $\mathbb{P}[T_{\mathcal{H}}(\text{x}) = (C_1, ..., C_\mu)] \geq 1 - \text{negl}(\lambda)$

(b) While this unique vector of high support may depend on $\mathcal{H}$, for all $i \in [\mu]$ the first i components $(C_1, ..., C_i)$ are independent of the answers to the values $\mathcal{H}(q)$ for $q \in \{(C, C_1, ..., C_j)\}_{i \leq j < \mu}$.

If (a) were false then running the experiment above for case $i = \mu$ would succeed with non-negligible probability δ' in returning a distinct assignment to $\mathcal{T}_{\mathcal{H}}(\text{x})$, which contradicts the computational uniqueness of the commitment scheme as shown above. If (b) were false, then for some $i < \mu$ the experiment would succeed with non-negligible probability δ' in returning a distinct assignment to $\mathcal{T}_{\mathcal{H}}(\text{x})$ where the first index of distinction is $k \leq i$, again contradicting the computational uniqueness of the commitment scheme as shown above.

Finally, (b) implies that $(\alpha_1, ..., \alpha_\mu)$ where $\alpha_i = \mathcal{H}(C, C_1, ..., C_i)$ is uniformly distributed and hence $\mathbb{P}[\rho(m, m_\mu, \alpha_1, ..., \alpha_\mu) = 1] \leq \delta$ as stated in the hypothesis. Thus, conditioned on the event that w satisfies at least all validity criteria except the predicate, then its first component is $\text{tr} = (C_1, ..., C_\mu)$ with probability $1 - \text{negl}(\lambda)$, in which case it fails the last criteria (i.e., the predicate) with probability $1 - \delta$. In conclusion, by a union bound it satisfies all criteria with probability at most $\delta + \text{negl}(\lambda)$. $\square$

Theorem 6 (Fiat Shamir for AMSS protocols with unique commitments). If Π is a $(k^{(\mu)}, \delta, \text{com}, \phi)$-almost-special-sound protocol for a relation $\mathcal{R}$ and a computationally-unique commitment scheme com (Definition 19) whose setup runs $\text{pp} \leftarrow \text{com.Setup}(\lambda)$

then its FS transform Π_{FS} is an argument of knowledge for $\mathcal{R}$ in the RO model (Definition 16) with knowledge error:

$$\mathsf{err}(\lambda, Q) = \frac{(Q+1)\kappa}{1-\kappa} + 2\lambda(k^\mu + Q \cdot (k^\mu - 1)) \cdot \delta + \mathsf{negl}(\lambda)$$

where $\kappa = 1 - (1 - \frac{k}{2^\lambda})^\mu$.

Proof. Let $\mathcal{V}^{\mathcal{H}}$ denote the resulting verifier for Π_{FS}. We will construct an extractor $\mathcal{E}$ which is given black-box access to any *deterministic* Q-query prover algorithm $\mathcal{P}^*$, where Q is assumed to be polynomial in λ. $\mathcal{E}$ has the power to intercept and respond to the queries $\mathcal{P}^*$ makes to the random oracle, simulating (i.e., reprogramming) the oracle responses. On input x and parameters pp, $\mathcal{E}$ first tests that $\mathcal{P}^*$ outputs a proof π such that $\mathcal{V}^{\mathcal{H}}(\mathsf{pp}, x, \pi) = 1$ and otherwise aborts. If this first step succeeds, then $\mathcal{E}$ continues by running the tree generation algorithm from Lemma 13 to generate a $(k+1)$-ary forking transcript tree. Given black-box access to a deterministic Q-query prover algorithm, this tree generation algorithm runs in expected polynomial time $k^\mu + Q \cdot (k^\mu - 1)$ succeeding with probability $\frac{\epsilon(x) - (Q+1) \cdot \kappa}{(1-\kappa)}$ where $\kappa = 1 - (1 - \frac{k}{2^\lambda})^\mu$ and $\epsilon(x)$ is the probability over the randomness of $\mathcal{H}$ that $\mathcal{P}^*$ outputs a non-interactive proof that $\mathcal{V}^{\mathcal{H}}$ accepts. $\mathcal{E}$ will repeat λ iterations of running this tree generation algorithm for $2(k^\mu + Q \cdot (k^\mu - 1))$ steps each time, and returns the first trial that succeeds. By Markov, this results in a new tree generation algorithm that runs in strict polynomial time $2\lambda(k^\mu + Q \cdot (k^\mu - 1))$ with negligible loss $2^{-\lambda}$ in its probability of success.

For any $(k+1)$-ary transcript tree, there is a polynomial time procedure (Definition 12) which as shown in Lemma 12 either:

(a) Extracts a witness w such that $(x, w) \in \mathcal{R}$

(b) Extracts a break to the binding of the commitment scheme, i.e. an element of $\mathcal{L}_{\mathsf{break}}(\mathsf{pp})$ defined in Theorem 4.

(c) Extracts an opening (m_ω, o_ω) for the commitment label on some node ω at some level i with rightmost child ν at level $i+1$ such that $\phi_b(i, m_\omega) = 1$, $\phi_a(i, m_\omega) = 0$, openings of all commitment labels on the leftmost path $\mathsf{lpath}(\nu)$ extending down from ν to messages equal to $\mathsf{Extend}(i, m, \alpha_i, ..., \alpha_\mu) = (m_i, ..., m_\mu)$ where $(\alpha_i, ..., \alpha_\mu)$ are the verifier challenges along this path such that $\forall_{j \geq i} \phi_a(j, m_j) = 1$.

The third extraction event was ruled out (with overwhelming probability) from any transcript tree generated via the Path Predicate Forking Lemma, see Theorem 4 and Lemma 9. In particular, the transcript tree generated there was shown to satisfy a predicate (with overwhelming probability) that eliminates the possibility that $\phi_a(i, m_\omega) = 0$ yet $\phi_a(\mu, m_\mu) = 1$. The analysis leveraged the way that transcripts are sampled by that tree generation algorithm. However, we will need to use a slightly different analysis this time.

First, we define the relation for all $i \in [\mu]$, commitment scheme parameters pp, and random oracle $\mathcal{H}^*$:

$$\mathcal{R}_{\mathsf{ext}}^{\mathcal{H}^*}(\mathsf{pp}, \mu - i) = \left\{ (\mathsf{x} = (C, m, o), \mathsf{w} = (\mathsf{tr}, M)) : \begin{array}{l} \mathsf{tr} = (\mathsf{x}, C_1, ..., C_{\mu - i}) \\ \forall_{j \in [\mu - i]} \, \alpha_j = \mathcal{H}^*(C, C_1, ..., C_{j-1}) \\ M = ((m_1, o_1), ..., (m_{\mu-i}, o_{\mu-i})) \\ \forall_{j \in [\mu - i]} \mathsf{com.Open}(\mathsf{pp}, C_j, m_j, o_j) = 1 \\ \mathsf{Extend}(i, m, \alpha_1, ..., \alpha_{\mu-i}) = (m_1, ..., m_{\mu-i}) \\ \phi_b(i, m) = 1, \phi_a(i, m) = 0, \phi_a(\mu, m_{\mu-i}) = 1 \end{array} \right\}$$

We note that by Lemma 13, in case (c) occurs, there is a $(Q + \mu)$-query polynomial time algorithm $\mathcal{A}_1$ that generates (C, m, o) and transcript prefix $\mathbf{y}$, and an independent $(Q + \mu)$-query adversary $\mathcal{A}_2^{\mathcal{H}^*}$ which generates the witness (tr, M) such that $((C, m, o), (\mathsf{tr}, M)) \in \mathcal{R}_{\mathsf{ext}}^{\mathcal{H}_\mathbf{y}^*}(\mathsf{pp}, \mu - i)$ for some i. $\mathcal{A}_1$ represents the algorithm that ran the partial tree generation that created all labels on the left k-ary subtree of ω and also the prefix $\mathbf{y}$ labeling the trunk (i.e., from root to ω) of subtree S_ν, and then also ran the tree extraction algorithm (Definition 12) on this left k-ary subtree of ω. Note that conditioned on event (c), ω is the first node and index i for which this tree extraction succeeds in producing an opening (m_ω, o_ω) such that $\phi_b(i, m_\omega) = 1$ but $\phi_a(i, m_\omega) = 0$. By Lemma 13 the root-to-leaf path that includes the prefix $\mathbf{y}$ and $\mathsf{lpath}(\nu)$ matches the output of some $\mathcal{A}_*^{\mathcal{H}^*}(x)$ with partially fresh random oracle $\mathcal{H}^*$, and there is also a subtree generation algorithm that is a $(Q + \mu)$-query algorithm which generated the labels on subtree S_ν. $\mathcal{A}_2^{\mathcal{H}^*}$ represents the combination of these two algorithms and also the subtree extractor that opens the commitments on these labels. Conditioned on (c),

the openings of the commitment labels within this subtree all satisfy predicate ϕ_a and the opened messages of the commitment labels $\mathsf{lpath}(\nu)_0$ along the leftmost path from ν match the output of $\mathsf{Extend}(i, m, \mathsf{lpath}(v)_1)$ where $\mathsf{lpath}(\nu)_1$ are the challenge labels along this path.

Let ρ_i denote the predicate such that $\rho_i(m, m', \alpha_1, ..., \alpha_{\mu-i}) = 1$ iff $\phi_b(i, m) = 1$, $\phi_a(i, m) = 0$, and $\phi_a(\mu, m') = 1$. By the definition of $(k^{(\mu)}, \delta, \mathsf{com}, \phi)$-almost-special-soundness, for uniform random $\beta_1, ..., \beta_{\mu-i}$:

$$\mathbb{P}_{\beta_1, ..., \beta_{\mu-i}}[\rho(m, m_{\mu-i}, \beta_1, ..., \beta_{\mu-i}) = 1 : \mathsf{Extend}(i, m, \beta_1, ..., \beta_{\mu-i}) = (m_1, ..., m_{\mu-i})] \leq \delta$$

Thus, since $Q + \mu$ is polynomial in λ, by Lemma 15 the relation $\mathcal{R}_{\mathsf{ext}}^{\mathcal{H}^*}(\mathsf{pp}, \mu - i)$ is $(Q + \mu, \delta + \mathsf{negl}(\lambda))$-hard in the RO model for $Q = \mathsf{poly}(\lambda)$. This shows that for any particular index in the transcript tree, the event of type (c) occurs with probability at most $\delta + \mathsf{negl}(\lambda)$ when running the extractor with polynomial time provers making a polynomial number of queries to the RO. This experiment may effectively occur times over the course of the tree generation algorithm, but we can loosely union bound the probability that event (c) ever occurs by the runtime of the tree generation algorithm. Similarly, we can eliminate event (b) as occurring with $\mathsf{negl}(\lambda)$ by the computational binding property of the commitment scheme, which does not require an additional union bound.

Finally, letting $\epsilon(x, \mathsf{st})$ denote the probability over the parameters and random oracle that the verifier accepts the proof π output by $P^*(\mathsf{st})$ for public input x, we conclude that:

$$\mathbb{P}\left[\begin{array}{c} (x,w) \in \mathcal{R} \ and \\ \mathcal{V}^{\mathcal{H}}(\mathsf{pp}, x, \pi) = 1 \end{array} : \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{Setup}(\lambda) \\ (x, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{pp}) \\ \pi \leftarrow P^*(\mathsf{st}) \\ w \leftarrow \mathcal{E}^{P^*(\mathsf{st})}(\mathsf{pp}, x) \end{array}\right] = \mathbb{P}\left[(x,w) \in \mathcal{R} : \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{Setup}(\lambda) \\ (x, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{pp}) \\ \pi \leftarrow P^*(\mathsf{st}) \\ w \leftarrow \mathcal{E}^{P^*(\mathsf{st})}(\mathsf{pp}, x) \end{array}\right]$$

$$\geq \sum_{x, \mathsf{st}} \left(\frac{\epsilon(x, \mathsf{st}) - (Q+1)\kappa}{1 - \kappa} - 2\lambda(k^\mu + Q \cdot (k^\mu - 1)) \cdot \delta - \mathsf{negl}(\lambda) \right) \cdot \mathbb{P}\left[\mathcal{A}(\mathsf{pp}) = (x, \mathsf{st}) : \mathsf{pp} \leftarrow \mathsf{Setup}(\lambda)\right]$$

$$\geq \mathbb{P}\left[\mathcal{V}^{\mathcal{H}}(\mathsf{pp}, x, \pi) = 1 : \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{Setup}(\lambda) \\ (x, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{pp}) \\ \pi \leftarrow P^*(\mathsf{st}) \end{array}\right] - \frac{(Q+1)\kappa}{1-\kappa} - 2\lambda(k^\mu + Q \cdot (k^\mu - 1)) \cdot \delta - \mathsf{negl}(\lambda)$$

The first equality holds because $\mathcal{E}^{P^*(\mathsf{st})}(\mathsf{pp}, x)$ aborts in its first step if the deterministic $\mathcal{P}^*(\mathsf{st})$ outputs π such that $\mathcal{V}^{\mathcal{H}}(\mathsf{pp}, x, \pi) \neq 1$. $\qquad \square$