Sub-Linear Lattice-Based Zero-Knowledge Arguments for Arithmetic Circuits*

Carsten Baum[1], Jonathan Bootle[2], Andrea Cerulli[2],
Rafael del Pino[3], Jens Groth[2], and Vadim Lyubashevsky[3]

[1] Bar-Ilan University
carsten.baum@biu.ac.il
[2] University College London, London, UK
{jonathan.bootle.14, andrea.cerulli.13, j.groth}@ucl.ac.uk
[3] IBM Research - Zurich

Abstract. We propose the first zero-knowledge argument with sub-linear communication complexity for arithmetic circuit satisfiability over a prime p whose security is based on the hardness of the short integer solution (SIS) problem. For a circuit with N gates, the communication complexity of our protocol is $O\left(\sqrt{N\lambda\log^3 N}\right)$, where λ is the security parameter. A key component of our construction is a surprisingly simple zero-knowledge proof for pre-images of linear relations whose amortized communication complexity depends only logarithmically on the number of relations being proved. This latter protocol is a substantial improvement, both theoretically and in practice, over the previous results in this line of research of Damgård et al. (CRYPTO 2012), Baum et al. (CRYPTO 2016), Cramer et al. (EUROCRYPT 2017) and del Pino and Lyubashevsky (CRYPTO 2017), and we believe it to be of independent interest.

Keywords: Sigma-protocol, zero-knowledge argument, arithmetic circuit, SIS assumption.

1 Introduction

Zero-knowledge proofs and arguments are used throughout cryptography as a key ingredient to ensure security in complex protocols. They form an important part of applications such as authentication protocols, electronic voting systems, encryption primitives, multi-party computation schemes, and verifiable computation protocols. Therefore, designing zero-knowledge protocols with strong security and high efficiency is of the utmost importance.

* Jonathan Bootle, Andrea Cerulli and Jens Groth were supported by funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 307937. Rafael del Pino and Vadim Lyubashevsky were supported in part by the SNSF ERC Transfer Starting Grant CRETP2-166734-FELICITY. Carsten Baum acknowledges support by the BIU Center for Research in Applied Cryptography and Cyber Security in conjunction with the Israel National Cyber Bureau in the Prime Ministers Office.

A zero-knowledge argument allows a prover to convince a verifier that a particular statement is true, without the prover revealing any other information that she knows about the statement. Statements are of the form $u \in \mathcal{L}$, where $\mathcal{L}$ is a language in NP. We call w a witness for statement u if $(u, w) \in R$, where R is a polynomial time decidable binary relation associated with $\mathcal{L}$. Zero-knowledge arguments must be complete, sound and zero-knowledge.

Completeness: A prover with witness w for $u \in \mathcal{L}$ can convince the verifier.
Soundness: A prover cannot convince the verifier when $u \notin \mathcal{L}$.
Zero-knowledge: The interaction should not reveal anything to the verifier except that $u \in \mathcal{L}$. In particular, it should not reveal the prover's witness w.

We wish to design a zero-knowledge argument based on the short integer solution (SIS) assumption. Lattice problems appear to resist quantum attacks, and possess attractive worst-case to average-case reductions, in stark contrast with number theoretic assumptions such as the hardness of factoring or computing discrete logarithms. Moreover, using SIS (and the even more efficient Ring-SIS) yields better computational efficiency, which is a significant bottleneck in many zero-knowledge arguments.

1.1 Our Contributions

We provide an honest verifier zero-knowledge argument for arithmetic circuit satisfiability over $\mathbb{Z}_p$, for an arbitrary prime p. Our argument is based on the SIS assumption [Ajt96, MR04], which is conjectured to be secure even against a quantum adversary. Our argument has an expected constant number of moves and sub-linear communication complexity, as shown in Table 1. Moreover, it achieves small soundness error in a single protocol execution. Moreover, both the prover and verifier have quasi-linear computational complexity in the amount of computation it would require to evaluate the arithmetic circuit directly. The argument therefore improves on the state-of-the-art in communication complexity for lattice proof systems and is efficient on all performance parameters.

Expected # Moves	Communication (bits)	Prover Complexity (bit ops)	Verifier Complexity (bit ops)
$O(1)$	$O(\sqrt{N\lambda \log^3 N})$	$O(N \log N (\log \lambda)^2)$	$O(N(\log \lambda)^3)$

Table 1: Performance of our zero-knowledge argument for arithmetic circuit satisfiability. Here N is the number of gates in the arithmetic circuit, and λ is the security parameter.

Techniques. We draw inspiration from the discrete logarithm based arithmetic circuit satisfiability argument of Bootle et al. [BCC$^+$16], which requires 5 moves and has square root communication complexity in the number of multiplication gates. In their argument the prover commits to all the wires using homomorphic commitments, and embeds the wire values into a polynomial that verifies products and linear relations simultaneously, avoiding the cost for addition gates.

Almost all parts of the original arguments adapt seamlessly to the SIS setting, except for two important issues:

- To achieve sub-linear communication, we need a technique for proving knowledge of commitment openings in sub-linear space.
- Due to the new algebraic setting, we require new techniques for achieving negligible soundness in a single run of the protocol.

The first of these issues has been an open problem in a fairly active area of research, and we sketch our solution below.

Proof of Knowledge. Suppose that we have a linear relation

$$\boldsymbol{A}\boldsymbol{s} = \boldsymbol{t} \bmod q, \tag{1}$$

where $\boldsymbol{A} \in \mathbb{Z}_q^{r \times v}, \boldsymbol{t} \in \mathbb{Z}_q^r$ are public and $\boldsymbol{s} \in \mathbb{Z}_q^v$ is a vector with small coefficients, and we want to give a zero-knowledge proof of knowledge of an $\bar{\boldsymbol{s}}$ with small coefficients (the coefficients of $\bar{\boldsymbol{s}}$ may be larger than those of $\boldsymbol{s}$) that satisfies

$$\boldsymbol{A}\bar{\boldsymbol{s}} = \boldsymbol{t} \bmod q. \tag{2}$$

We do not currently know of any an efficient linear-communication protocol for proving knowledge of a single relation of the above form in a direct way. There are protocols, however, that allow for proofs of many such relations for the same $\boldsymbol{A}$ but different $\boldsymbol{s}_i$ (and thus different $\boldsymbol{t}_i$) in linear amortized complexity. We will mention these previous works in more detail in Section 1.2.

In this work, we give a protocol for proving (1) where the proof length is a factor $\frac{\lambda}{\ell} \cdot O(\log v\ell\lambda)$ larger than the total bit-length of ℓ pre-images $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_\ell$ of the relations, where λ is the security parameter. More specifically, to prove knowledge of ℓ pre-images $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_\ell$ whose coefficients have $\log s$ bits each, the prover needs to send λ vectors in $\mathbb{Z}_q^v$ whose coefficients require $O(\log v\ell\lambda s)$ bits to represent. Ignoring logarithmic terms, our proof essentially requires a fixed-size proof regardless of the number of relations being proved. The previously best results had proofs that were at least linear in the total size of the pre-images.

Surprisingly, the proof of knowledge protocol turns out to just be a parallel repetition of λ copies of the ZKPoK implicit in the signing protocol from [Lyu12]. In particular, if we write the ℓ relations as $\boldsymbol{A}\boldsymbol{S} = \boldsymbol{T} \bmod q$, where $\boldsymbol{S} \in \mathbb{Z}_q^{v \times \ell}$, then the protocol begins with the prover selecting a "masking" value $\boldsymbol{Y}$ with small coefficients and sending $\boldsymbol{W} = \boldsymbol{A}\boldsymbol{Y} \bmod q$. The verifier then picks a random challenge matrix $\boldsymbol{C} \in \{0,1\}^{\ell \times (\lambda+2)}$, and sends it to the prover. The prover computes $\boldsymbol{Z} = \boldsymbol{S}\boldsymbol{C} + \boldsymbol{Y}$ and performs a rejection sampling step in order to make the distribution of $\boldsymbol{Z}$ independent from $\boldsymbol{S}$, and if it passes, sends $\boldsymbol{Z}$ to the verifier. The verifier checks that all columns comprising $\boldsymbol{Z}$ have small norms and that $\boldsymbol{A}\boldsymbol{Z} = \boldsymbol{T}\boldsymbol{C} + \boldsymbol{W} \bmod q$. This protocol can be shown to be zero-knowledge using exactly the same techniques as in [Lyu09, Lyu12].

To show that the protocol is a proof of knowledge, we make the following observation: if the prover succeeds with probability $\epsilon > 2^{-\lambda}$, and she succeeded for a random $\boldsymbol{C}$, then there is a probability of $\epsilon - 2^{-\lambda-2}$ that she would successfully answer another challenge $\boldsymbol{C}' \neq \boldsymbol{C}$ that is constructed such that all rows except the i^{th} are the same as that of $\boldsymbol{C}$, and the i^{th} row is picked uniformly at random. This property follows from an averaging (or "heavy row") type argument. The implication is that if the prover succeeds in time t with probability ϵ, then the extractor can extract responses to two such commitments $\boldsymbol{C}, \boldsymbol{C}'$ in expected time $O(t/\epsilon)$. Obtaining two responses $\boldsymbol{Z}, \boldsymbol{Z}'$ for two such challenges allows the extractor to compute $\boldsymbol{A}(\boldsymbol{Z} - \boldsymbol{Z}') = \boldsymbol{T}(\boldsymbol{C} - \boldsymbol{C}')$ where $\boldsymbol{C} - \boldsymbol{C}'$ is 0 everywhere except in row i. Since $\boldsymbol{C} \neq \boldsymbol{C}'$, this implies that some position in row i is ± 1. If $\boldsymbol{t}_i$ is the i^{th} column of $\boldsymbol{T}$ and $\boldsymbol{z}_i$ is the i^{th} column of $\boldsymbol{Z} - \boldsymbol{Z}'$, then we have a solution $\boldsymbol{A}\boldsymbol{z}_i = \pm \boldsymbol{t}_i$. Repeating this extraction ℓ times, each time rewinding by fixing all rows in the challenge except for the i^{th}, results in an algorithm that runs in expected time $O(\ell \cdot t/\epsilon)$, which is only a factor of ℓ larger than the expected running time of a successful prover.

In the case that we are proving (1) over the polynomial ring $\mathbb{Z}[X]/(X^d + 1)$, the proof can be even shorter, as we can reduce the number of columns in $\boldsymbol{C}$ to $\approx \lambda / \log 2d$ because we can use challenges of the form $\pm X^i$ and prove the knowledge of $\bar{\boldsymbol{s}}$ such that $\boldsymbol{A}\bar{\boldsymbol{s}} = 2\boldsymbol{t}$ using the observation from [BCK+14].

Commitment Scheme. Central to the main proof of proving circuit satisfiability is being able to commit to N values in $\mathbb{Z}_p$ and giving a ZKPoK for the values such that the total size of the commitments and the proofs is sub-linear in N. For this, it is necessary to use a compressing commitment scheme – i.e. one in which we can commit to n elements of $\mathbb{Z}_p$ in space less than n elements. The scheme that we will use is the "classic" statistically-hiding commitment scheme based on the hardness of SIS that was already implicit in the original work of Ajtai [Ajt96]. The public randomness consists of two matrices $\boldsymbol{A} \in \mathbb{Z}_q^{r \times 2r \log_p q}, \boldsymbol{B} \in \mathbb{Z}_q^{r \times n}$, and committing to a message string $\boldsymbol{s} \in \mathbb{Z}_p^n$ where $p < q$ involves picking a random vector $\boldsymbol{r} \in \mathbb{Z}_p^{2r \log_p q}$ and outputting the commitment $\boldsymbol{t} = \boldsymbol{A}\boldsymbol{r} + \boldsymbol{B}\boldsymbol{s} \bmod q$. Thus the commitment of n elements of $\mathbb{Z}_p$ requires $r \log q$ bits. One can set the parameters such that $n = \mathsf{poly}(r)$ and the commitment scheme will still be computationally binding based on the worst-case hardness of approximating SIVP for all lattices of dimension r.

We now explain the intuition for putting together this commitment scheme with the zero-knowledge proof system we described above to produce a commitment to N values in $\mathbb{Z}_p$ such that the total size of the commitments and the ZKPoK of the committed values is $O(\sqrt{N\lambda \log N})$. The idea is to create N/n commitments (for some choice of n which will be optimized later), with each one committing to n values. Our motivation is that an arithmetic circuit over $\mathbb{Z}_p$ with N gates has $3N$ wire values in $\mathbb{Z}_p$. Now, we can arrange all of the wire values in the circuit into, for example, a $3N/n \times n$ matrix over $\mathbb{Z}_p$, and make one homomorphic commitment to all of the elements in each row of the matrix. Then, we can employ techniques from [Gro09a, BCC+16], where checking arithmetic circuit satisfiability is reduced to checking linear-algebraic statements over committed matrices, using a homomorphic commitment scheme.

The total space requirement for these commitments is therefore $\frac{N}{n} \cdot r \log_p q$. We now have a linear equation of the form $\begin{bmatrix} \boldsymbol{A} \; \boldsymbol{B} \end{bmatrix} \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{S} \end{bmatrix} = \boldsymbol{T} \bmod q$. Using our new zero-knowledge proof, the communication complexity of proving the knowledge of a short $\begin{bmatrix} \bar{\boldsymbol{R}} \\ \bar{\boldsymbol{S}} \end{bmatrix} \in \mathbb{Z}_q^{(r \log_p q + n) \times N/n}$ such that $\begin{bmatrix} \boldsymbol{A} \; \boldsymbol{B} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{R}} \\ \bar{\boldsymbol{S}} \end{bmatrix} = \boldsymbol{T} \bmod q$ requires sending λ vectors of length $r \log_p q + n$ with coefficients requiring $O(\log N\lambda p)$ bits, for a total bit-length of $n \cdot \lambda \cdot O(\log N\lambda p)$. Combining the proof size with the commitment size results in a total bit-size of

$$\frac{N}{n} \cdot r \log q + n \cdot \lambda \cdot O(\log N\lambda p).$$

We minimize the above by setting $n = \sqrt{\frac{Nr \log_p q}{\lambda \log N\lambda p}}$, which makes the size

$$O\left(\sqrt{Nr\lambda(\log_p q)(\log N\lambda p)}\right).$$

Based on the complexity of the best known algorithm against the SIS problem, one can set $\log q, r = O(\log N)$, thus making the proof size of order $O(\sqrt{N\lambda \log^3 N})$.

1.2 Related Work

Zero-knowledge proofs were invented by Goldwasser et al. [GMR85]. It is useful to distinguish between zero-knowledge *proofs*, with statistical soundness, and zero-knowledge *arguments* with computational soundness. The most efficient proofs have communication proportional to the size of the witness [IKOS07, KR08, GGI$^+$15] and proofs cannot in general have communication that is smaller than the witness size unless surprising results about the complexity of solving SAT instances hold [GH98, GVW02]. Kilian [Kil92] showed that in contrast to proofs, zero-knowledge arguments can have very low communication complexity. His construction relied on the PCP theorem, and thus incurred a large computational cost.

Group theoretic zero-knowledge arguments. Schnorr [Sch91] and Guillou and Quisquater [GQ88] gave early examples of practical zero-knowledge arguments for concrete number theoretic problems. Extending Schnorr's protocols, there have been many constructions of zero-knowledge arguments based on the discrete logarithm assumption, for instance [CD97, Gro09b]. The most efficient discrete logarithm based zero-knowledge arguments for arithmetic circuits are by Bootle et al. [BCC$^+$16] and later optimised in [BBB$^+$17], which have logarithmic communication complexity and require a linear number of exponentiations.

An exciting line of research [Gro10a, Lip12, BCCT12, GGPR13, BCCT13, PHGR13, Gro16] on succinct non-interactive arguments (SNARGs) has yielded pairing-based constructions where the arguments consist of a constant number of group elements. However, it can be shown that all SNARKs must rely on non-falsifiable knowledge extractor assumptions [GW11]. In contrast, since our argument is interactive, we do not need to rely on these strong assumptions.

Lattice-based zero-knowledge arguments. The first zero-knowledge proofs from lattice-based assumptions were aimed at lattice problems themselves. Goldreich and Goldwasser [GG98] presented constant round interactive zero knowledge proofs for the complements of the approximate Shortest Vector Problem (SVP) and the approximate Closest Vector Problem (CVP). Micciancio and Vadhan [MV03] later constructed statistical zero knowledge proofs for these problems which had efficient provers.

Stern's protocol [Ste94] was one of the first zero-knowledge identification protocols to be based on a post-quantum assumption, namely, on the hardness of syndrome decoding for a random linear code, which is essentially proving (1) where $q = 2$ and $\|\boldsymbol{s}\| \ll \sqrt{v}$. The protocol achieves constant soundness error, and thus requires many parallel repetitions. Stern's work prompted many variants and similar protocols. For example, [LNSW13] adapts the protocol for larger q, which implies proving knowledge of SIS solutions.

Another technique for creating zero-knowledge proofs is the "Fiat-Shamir with Aborts" approach [Lyu09, Gro10b, Lyu12]. When working over polynomial rings R, it gives a proof of knowledge of a vector $\bar{\boldsymbol{s}}$ with small coefficients (though larger than those in $\boldsymbol{s}$) and a ring element $\bar{c}$ with very small coefficients satisfying $\boldsymbol{A}\bar{\boldsymbol{s}} = \bar{c}\boldsymbol{t}$. As long as the ring R has many elements with small coefficients, such proofs are very efficient, producing soundness of $1 - 2^{-128}$ with just one iteration. While these proofs are good enough for constructing practical digital signatures (e.g. [GLP12, DDLL13, BG14]), commitment schemes with proofs of knowledge [BKLP15, BDOP16], and certain variants of verifiable encryption schemes [LN17], they prove less than what the honest prover knows. In many applications where zero-knowledge proofs are used, in particular those that need to take advantage of additive homomorphisms, the presence of the element $\bar{c}$ makes these kinds of "approximate" proofs too weak to be useful. As of today, we do not have any truly practical zero-knowledge proof systems that give a proof of (1).

The situation is more promising when one considers *amortized* proofs. The work of [BD10] uses MPC-in-the-head to prove knowledge of plaintexts for multiple Regev [Reg05] ciphertexts. Damgård and López-Alt [DL12] extend the [BD10] results to prove knowledge of plaintext in $\mathbb{Z}_p$, rather than bits, and provide a proof for the correctness of multiplications. Combining these together gives a zero-knowledge proof for the satisfiability of arithmetic circuits with linear communication in the circuit size.

Another idea for proving the relation in (1) is to use the above-mentioned "Fiat-Shamir with Aborts" protocol, but with challenges that come from the set $\{0, 1\}$. The works of [BDLN16, CDXY17, dPL17] gave a series of improved protocols that were able to employ this technique in the amortized setting. Their proofs had a small polynomial "slack" (i.e. the ratio between the original committed $\boldsymbol{s}$ and the extracted $\bar{\boldsymbol{s}}$) and were of approximate linear size when the number of commitments was a couple of thousand. The schemes are considerably less efficient when one is proving fewer relations.

The amortized zero-knowledge proof in the current work improves on the above series of papers in two important ways. First, the number of relations necessary before the size of our proof is linear only in λ. But more importantly, if we have more than λ relations, the communication complexity does not increase except for small logarithmic factors (i.e. the proof size becomes sub-linear).

Hash-based zero-knowledge arguments. Recently Bootle et al [BCG$^+$17] used error-correcting codes and linear-time collision-resistant hash functions to give proof systems for the satisfiability of an arithmetic circuit where the prover uses a linear number of field multiplications. Verification is even more efficient, requiring only a linear number of additions. While their proofs and arguments are asymptotically very efficient, they are not quite practical as their choices of error-correcting codes and hash functions involves very large constants.

An another effective way to construct efficient zero-knowledge proofs is to follow the so-called MPC-in-the-head paradigm of [IKOS07]. This approach proved itself to give very efficient constructions both theoretically and practically. Most notably, ZKBOO [GMO16] and subsequent optimisation ZKB++ [CDG$^+$17] use hash functions to construct zero-knowledge arguments for the satisfiability of boolean circuits. Their communication complexity is linear in the circuit size, but the use of symmetric primitives gives good performances in practice. Ligero [AHIV17] provides another implementation of the MPC-in-the-head paradigm and used techniques similar to [BCG$^+$17] to construct sublinear arguments for arithmetic circuits.

2 Preliminaries

Algorithms in our schemes receive a security parameter λ as input (sometimes implicitly) written in unary. The intuition is that the higher the security parameter, the lower the risk of the scheme being broken. Given two functions $f, g : \mathbb{N} \to [0, 1]$ we write $f(\lambda) \approx g(\lambda)$ when $|f(\lambda) - g(\lambda)| = \lambda^{-\omega(1)}$. We say that f is *negligible* when $f(\lambda) \approx 0$ and that f is *overwhelming* when $f(\lambda) \approx 1$. For any integer N, $[N]$ denotes the set $\{0, 1, \ldots, N-1\}$ of integers.

2.1 Notation

Throughout this paper we will consider a ring $\mathcal{R}$, which will be either $\mathbb{Z}$ or the polynomial ring $\mathbb{Z}[X]/(X^d + 1)$ for d some power of 2. We will denote elements of $\mathcal{R}$ by lowercase letters, (column) vectors over $\mathcal{R}$ in bold lowercase and matrices over $\mathcal{R}$ in bold uppercase. e.g. $\boldsymbol{A} = \big[\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k\big] \in \mathcal{R}^{l \times k}$ with $\boldsymbol{a}_i = (a_{i1}, \ldots, a_{im})^T \in \mathcal{R}^l$. We will consider the norm of elements in $\mathcal{R}$ to be $\|a\|_2 = |a|$ if $a \in \mathbb{Z}$, and $\|a\|_2 = \sqrt{\sum a_i^2}$ if $a = \sum a_i X^i \in \mathbb{Z}[X]/(X^d + 1)$. We extend the notation to vectors and matrices $\|\boldsymbol{a}\|_2 = \sqrt{\sum \|a_i\|_2^{\,2}}$, $\|\boldsymbol{A}\|_2 = \sqrt{\sum \|\boldsymbol{a}_i\|_2^{\,2}}$. We

will also consider the quotient ring $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$ for odd q. In the quotient ring, the norm of an element $\mathcal{R}_q$ will be the norm of its unique representative $\mathcal{R}$ with coefficients in $\left[-\frac{q-1}{2}, \frac{q-1}{2}\right]$.
We will also consider the operator norm of matrices over $\mathcal{R}$ defined as $s_1(\boldsymbol{A}) = \underset{\|x\|_2 \neq 0}{max} \left(\frac{\|\boldsymbol{A}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}\right)$.

Probability Distributions. Let $\mathcal{D}$ denote a distribution over some set. Then, $d \leftarrow \mathcal{D}$ means that d was sampled from the distribution $\mathcal{D}$. If we write $d \overset{\$}{\leftarrow} S$ for some finite set S without a specified distribution this means that d was sampled uniformly random from S. We let $\Delta(X, Y)$ indicate the statistical distance between two distributions X, Y. Define the function $\rho_\sigma(x) = \exp\left(\frac{-x^2}{2\sigma^2}\right)$ and the discrete Gaussian distribution over the integers, D_σ, as

$$D_\sigma(x) = \frac{\rho(x)}{\rho(\mathbb{Z})} \text{ where } \rho(\mathbb{Z}) = \sum_{v \in \mathbb{Z}} \rho(v).$$

We will write $\boldsymbol{X} \leftarrow D_\sigma^{r \times m}$ to mean that every coefficient of the matrix $\boldsymbol{X}$ is distributed according to D_σ.

Using the tail bounds for the 0-centered discrete Gaussian distribution (cf. [Ban93]), we can show that for any $\sigma > 0$ the norm of $x \leftarrow D_\sigma$ can be upper-bounded using σ. Namely, for any $k > 0$ it holds that

$$\Pr_{x \leftarrow D_\sigma}[|x| > k\sigma] \leq 2e^{-k^2/2}, \tag{3}$$

and when $\boldsymbol{x}$ is drawn from D_σ^r, we have

$$\Pr_{\boldsymbol{x} \leftarrow D_\sigma^r}[\|\boldsymbol{x}\|_2 > \sqrt{2r} \cdot \sigma] < 2^{-r/4}. \tag{4}$$

We will abuse the notation $x \leftarrow D_\sigma$ when $x \in \mathbb{Z}[X]/(X^d + 1)$ to denote the distribution in which each coefficient of x is taken from D_σ. It is clear that in this case $\|x\|_2$ can be bounded using equation 4 with d instead of r.

2.2 Lattice-based Commitment Schemes

A commitment scheme allows a sender to create commitments to secret values, which she might then decide to reveal later. The main properties of commitment schemes are hiding and binding. Hiding guarantees that commitments do not leak information about the committed values, while binding guarantees that the sender cannot change her mind and open commitments to different values.

Formally, a non-interactive commitment scheme is a pair of probabilistic polynomial-time algorithms (Gen, Com). The setup algorithm $ck \leftarrow \text{Gen}(1^\lambda)$ generates a commitment key ck, which specifies message, randomness and commitment spaces $\mathsf{M}_{ck}, \mathsf{R}_{ck}, \mathsf{C}_{ck}$. It also specifies an efficiently sampleable probability distribution $D_{\mathsf{R}_{ck}}$ over R_{ck} and a binding set $\mathsf{B}_{ck} \subset \mathsf{M}_{ck} \times \mathsf{R}_{ck}$. The commitment key also specifies a deterministic polynomial-time commmitment function $\mathrm{Com}_{ck} : \mathsf{M}_{ck} \times \mathsf{R}_{ck} \to \mathsf{C}_{ck}$. We define $\mathrm{Com}_{ck}(\boldsymbol{m})$ to be the probabilistic algorithm that given $\boldsymbol{m} \in \mathsf{M}_{ck}$ samples $\boldsymbol{r} \leftarrow D_{\mathsf{R}_{ck}}$ and returns $\boldsymbol{c} = \mathrm{Com}_{ck}(\boldsymbol{m}; \boldsymbol{r})$.

The commitment scheme is homomorphic, if the message, randomness and commitment spaces are abelian groups (written additively) and we have for all $\lambda \in \mathbb{N}$, and for all $ck \leftarrow \mathrm{Gen}(1^\lambda)$, for all $\boldsymbol{m}_0, \boldsymbol{m}_1 \in \mathsf{M}_{ck}$ and for all $\boldsymbol{r}_0, \boldsymbol{r}_1 \in \mathsf{R}_{ck}$

$$\mathrm{Com}_{ck}(\boldsymbol{m}_0; \boldsymbol{r}_0) + \mathrm{Com}_{ck}(\boldsymbol{m}_1; \boldsymbol{r}_1) = \mathrm{Com}_{ck}(\boldsymbol{m}_0 + \boldsymbol{m}_1; \boldsymbol{r}_0 + \boldsymbol{r}_1).$$

Definition 1 (Hiding). *The commitment scheme is computationally hiding if a commitment does not reveal the committed value. Formally, we say the commitment scheme is hiding if for all probabilistic polynomial time stateful interactive adversaries $\mathcal{A}$*

$$\Pr\left[\begin{array}{l} ck \leftarrow \mathrm{Gen}(1^\lambda); (\boldsymbol{m}_0, \boldsymbol{m}_1) \leftarrow \mathcal{A}(ck); b \leftarrow \{0,1\}; \\ \boldsymbol{r} \leftarrow D_{\mathsf{R}_{ck}}; \boldsymbol{c} \leftarrow \mathrm{Com}_{ck}(\boldsymbol{m}_b; \boldsymbol{r}) : \mathcal{A}(\boldsymbol{c}) = b \end{array}\right] \approx \frac{1}{2},$$

where $\mathcal{A}$ outputs $\boldsymbol{m}_0, \boldsymbol{m}_1 \in \mathsf{M}_{ck}$.

Definition 2 (Binding). *The commitment scheme is computationally binding if a commitment can only be opened to one value within the binding set B_{ck}. For all probabilistic polynomial time adversaries $\mathcal{A}$*

$$\Pr\left[\begin{array}{c} ck \leftarrow \mathrm{Gen}(1^\lambda); (\boldsymbol{m}_0, \boldsymbol{r}_0, \boldsymbol{m}_1, \boldsymbol{r}_1) \leftarrow \mathcal{A}(ck) : \\ \boldsymbol{m}_0 \neq \boldsymbol{m}_1 \textit{ and } \mathrm{Com}_{ck}(\boldsymbol{m}_0; \boldsymbol{r}_0) = \mathrm{Com}_{ck}(\boldsymbol{m}_1; \boldsymbol{r}_1) \end{array}\right] \approx 0,$$

where $\mathcal{A}$ outputs $(\boldsymbol{m}_0, \boldsymbol{r}_0), (\boldsymbol{m}_1, \boldsymbol{r}_1) \in \mathsf{B}_{ck}$.

The commitment scheme is compressing if the sizes of commitments are smaller than the sizes of the committed values.

Ajtai's One-Way Function. The standard one-way function used in lattice cryptography maps a vector $\mathcal{R}^n$ to $\mathcal{R}^r$ via the function

$$f_{\boldsymbol{A}}(\boldsymbol{s}) = \boldsymbol{A}\boldsymbol{s},$$

where $\boldsymbol{A}$ is a fixed, randomly-chosen matrix in $\mathcal{R}^{r \times n}$. Ajtai's seminal result [Ajt96] stated that when $\mathcal{R} = \mathbb{Z}_q$, it is as hard to find elements $\boldsymbol{s}$ with some bounded norm $\|\boldsymbol{s}\| \leq B$ such that $f_{\boldsymbol{A}}(\boldsymbol{s}) = 0$ for random $\boldsymbol{A}$, as it is to find short vectors in any lattice of dimension r. This is called the short integer solution (SIS) problem and its hardness increases as r, q increase and B decreases; but somewhat surprisingly, the hardness of SIS is essentially unaffected by n as soon as n is large enough. The independence of the hardness from n holds both theoretically and in practice.

When solving SIS, one can ignore, if one wishes, any columns of $\boldsymbol{A}$ by setting the corresponding coefficient of $\boldsymbol{s}$ to 0, and solving SIS over the remaining columns. It was computed in [MR08] that if n is very large, then one should solve SIS for a submatrix where the number of columns is $n' = \sqrt{r \log q / \log \delta}$ for some constant δ.[4] With such a setting of n', one expects to find a vector of length approximately

$$\min\{q, 2^{\sqrt{r \log q \log \delta}}\}. \tag{5}$$

Compressing Commitments Based on SIS. The fact that a larger n (after a certain point) does not decrease the security of the scheme allows one to construct simple compressing commitment schemes where the messages are elements in $\mathbb{Z}_p$ for $p < q$. The commitment scheme, which was already implicit in the aforementioned work of Ajtai [Ajt96], uses uniformly-random matrices $\boldsymbol{A}_1 \in \mathbb{Z}_q^{r \times 2r \log_p q}$ and $\boldsymbol{A}_2 \in \mathbb{Z}_q^{r \times n}$ as a commitment key, where n is the number of elements that one wishes to commit to. A commitment to a vector $\boldsymbol{m} \in \mathbb{Z}_p^n$ involves choosing a random vector $\boldsymbol{r} \in \mathbb{Z}_p^{2r \log_p q}$ and outputting the commitment vector $\boldsymbol{v} = \boldsymbol{A}_1\boldsymbol{r} + \boldsymbol{A}_2\boldsymbol{m} \bmod q$. By the leftover hash lemma, $(\boldsymbol{A}_1, \boldsymbol{A}_1\boldsymbol{r} \bmod q)$ is statistically close to uniform, and so the commitment scheme is statistically hiding.[5]

To prove binding, note that if there are two different $(\boldsymbol{r}, \boldsymbol{m}) \neq (\boldsymbol{r}', \boldsymbol{m}')$ such that $\boldsymbol{v} = \boldsymbol{A}_1\boldsymbol{r} + \boldsymbol{A}_2\boldsymbol{m} = \boldsymbol{A}_1\boldsymbol{r}' + \boldsymbol{A}_2\boldsymbol{m}' \bmod q$, then $\boldsymbol{A}_1(\boldsymbol{r} - \boldsymbol{r}') + \boldsymbol{A}_2(\boldsymbol{m} - \boldsymbol{m}') = \boldsymbol{0} \bmod q$, and the non-zero vector $\boldsymbol{s} = \begin{bmatrix} \boldsymbol{r} - \boldsymbol{r}' \\ \boldsymbol{m} - \boldsymbol{m}' \end{bmatrix}$ is a solution to the SIS problem for the matrix $\boldsymbol{A} = [\boldsymbol{A}_1 \ \boldsymbol{A}_2]$. As long as the parameters are set such that $\|\boldsymbol{s}\|$ is smaller than the value in (5), the binding property of the commitment is based on an intractable version of the SIS problem.

The commitment scheme we will be working with in this paper works as follows:

Gen$(1^\lambda) \to ck$: Select a ring $\mathcal{R}$ (either $\mathbb{Z}$ or $\mathbb{Z}[X]/(X^d + 1)$), and parameter p, q, r, v, N, B, σ according to Table 2, and let $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$.
Pick uniformly at random matrices $\boldsymbol{A}_1 \leftarrow \mathcal{R}_q^{r \times r \log_p q}$ and $\boldsymbol{A}_2 \leftarrow \mathcal{R}_q^{r \times n}$.
Return $ck = (p, q, r, v, \ell, N, B, \mathcal{R}_q, A_1, A_2)$.
The commitment key defines message, randomness, commitment and binding spaces and distribution $\mathsf{M}_{ck} = \mathcal{R}_q^n \quad \mathsf{R}_{ck} = \mathcal{R}_q^{2r \log_p q}, \mathsf{C}_{ck} = \mathcal{R}_q^r$, $\mathsf{B}_{ck} = \left\{ \boldsymbol{s} = \begin{bmatrix} \boldsymbol{m} \\ \boldsymbol{r} \end{bmatrix} \in \mathcal{R}_q^{n + 2r \log_p q} \;\middle|\; ||\boldsymbol{s}|| < B \right\}, D_{\mathsf{R}_{ck}} = D_\sigma^r$.

Com$_{ck}(\boldsymbol{m}; \boldsymbol{r})$: Given $\boldsymbol{m} \in \mathcal{R}_q^n$ and $\boldsymbol{r} \in \mathcal{R}_q^{2r \log_p q}$ return $\boldsymbol{c} = \boldsymbol{A}_1\boldsymbol{r} + \boldsymbol{A}_2\boldsymbol{s}$.

In the following, when we make multiple commitments to vectors $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_\ell \in \mathsf{M}_{ck}$ we write $\boldsymbol{C} = \mathrm{Com}_{ck}(\boldsymbol{M}; \boldsymbol{R})$ when concatenating the commitment vectors as $\boldsymbol{C} = [\boldsymbol{c}_1, \cdots, \boldsymbol{c}_\ell]$. It corresponds to computing $\boldsymbol{C} = \boldsymbol{A}_1\boldsymbol{R} + \boldsymbol{A}_2\boldsymbol{M}$ with $\boldsymbol{M} = [\boldsymbol{m}_1, \cdots, \boldsymbol{m}_\ell]$ and randomness $\boldsymbol{R} = [\boldsymbol{r}_1, \cdots, \boldsymbol{r}_\ell]$.

[4] This constant δ is related to the optimal block-size in BKZ reduction [GN08], which is the currently best way of solving the SIS problem. Presently, the optimal lattice reductions set $\delta \approx 1.005$.

[5] For improved efficiency, one could reduce the number of columns in $\boldsymbol{A}_1$ and make the commitment scheme computationally-hiding based on the hardness of the LWE problem.

2.3 Arguments of Knowledge

We aim to give efficient lattice-based proofs for arithmetic circuit satisfiability over $\mathbb{Z}_p$. The strategy we will employ is to commit to the values of a satisfying assignment to the wires, execute a range proof to demonstrate the committed values are within a suitable range, and to prove the committed values satisfy the constraints imposed by the arithmetic circuit. We will now formally define arguments of knowledge.

Let R be a polynomial time decidable ternary relation. The first input will contain some public parameters (aka common reference string) pp. We define the corresponding language L_{pp} indexed by the public parameters that consists of elements u with a witness w such that $(pp, u, w) \in R$. This is a natural generalisation of standard NP languages, which can be cast as the special case of relations that ignore the first input.

A proof system consists of a PPT parameter generator PGen, and interactive and stateful PPT algorithms $\mathcal{P}$ and $\mathcal{V}$ used by the prover and verifier. We write $(tr, b) \leftarrow \langle \mathcal{P}(pp), \mathcal{V}(pp, t)\rangle$ for running $\mathcal{P}$ and $\mathcal{V}$ on inputs pp, s, and t and getting communication transcript tr and the verifier's decision bit b. Our convetion is $b = 0$ means reject and $b = 1$ means accept.

Definition 3 (Argument of knowledge). *The proof system* (PGen, $\mathcal{P}, \mathcal{V}$) *is called an* argument of knowledge *for the relation R if it is complete and knowledge sound as defined below.*

Definition 4 (Statistical completeness). (PGen, $\mathcal{P}, \mathcal{V}$) *has statistical completeness with completeness error $\rho : \mathbb{N} \to [0; 1]$ if for all adversaries $\mathcal{A}$*

$$\Pr\begin{bmatrix} pp \leftarrow \mathrm{PGen}(1^\lambda); (u, w) \leftarrow \mathcal{A}(pp); (tr, b) \leftarrow \langle \mathcal{P}(pp, u, w), \mathcal{V}(pp, u)\rangle : \\ (pp, u, w) \in R \textit{ and } b = 0 \end{bmatrix} \leq \rho(\lambda).$$

Definition 5 (Computational knowledge soundness). $(\mathcal{K}, \mathcal{P}, \mathcal{V})$ *is knowledge sound with knowledge soundness error $\epsilon : \mathbb{N} \to [0; 1]$ if for all deterministic polynomial time $\mathcal{P}^*$ there exists an expected polynomial time extractor $\mathcal{E}$ such that for all PPT adversaries $\mathcal{A}$*

$$\Pr\begin{bmatrix} pp \leftarrow \mathrm{PGen}(1^\lambda); (u, s) \leftarrow \mathcal{A}(pp); (tr, b) \leftarrow \langle \mathcal{P}^*(pp, u, s), \mathcal{V}(pp, u)\rangle; \\ w \leftarrow \mathcal{E}^{P^*(pp,u,s)}(pp, u, tr, b) : (pp, u, w) \notin R \textit{ and } b = 1 \end{bmatrix} \leq \epsilon(\lambda).$$

It is sometimes useful to relax the definition of knowledge soundness to hold only for a larger relation $\bar{R}$ such that $R \subset \bar{R}$. In this work, our zero-knowledge proofs of pre-images will for instance have "slack". Thus, even though $\boldsymbol{v}$ is constructed using $\boldsymbol{r}, \boldsymbol{m}$ with coefficients in $\mathbb{Z}_p$, we will only be able to prove knowledge of vectors $\bar{\boldsymbol{r}}, \bar{\boldsymbol{m}}$ with larger norms. This extracted commitment is still binding as long as the parameters are set such that the vector $\bar{\boldsymbol{s}} = \begin{bmatrix} \bar{\boldsymbol{r}} - \bar{\boldsymbol{r}}' \\ \bar{\boldsymbol{m}} - \bar{\boldsymbol{m}}' \end{bmatrix}$ has norm smaller than the bound in (5).[6]

[6] Commitments over other rings, such as $\mathbb{Z}_q[X]/(X^d + 1)$ can be done in the same manner as above based on the hardness of the Ring-SIS problem [PR06, LM06] for which the bound in (5) still appears to hold in practice.

Concretely, if we would like to make a commitment to N values in $\mathbb{Z}_p$, then to satisfy (5) we need to make sure that $q > \|\bar{\boldsymbol{s}}\|$ and $\sqrt{r \log q \log \delta} > \log \|\bar{\boldsymbol{s}}\|$. In the protocols in our paper, we will have $\|\bar{\boldsymbol{s}}\| < N^2 p^2$ and $p < N$, which implies that $r = O(\log N)$.

We say the proof system is *public coin* if the verifier's challenges are chosen uniformly at random independently of the prover's messages. A proof system is special honest verifier zero-knowledge if it is possible to simulate the proof without knowing the witness whenever the verifier's challenges are known in advance.

Definition 6 (Special honest-verifier zero-knowledge). *A public-coin argument of knowledge* $(\mathrm{PGen}, \mathcal{P}, \mathcal{V})$ *is said to be* statistical special honest-verifier zero-knowledge (SHVZK) *if there exists a PPT simulator* $\mathcal{S}$ *such that for all interactive and stateful adversaries* $\mathcal{A}$

$$\Pr\left[\begin{matrix} pp \leftarrow \mathrm{PGen}(1^\lambda); (u, w, \varrho) \leftarrow \mathcal{A}(pp); (tr, b) \leftarrow \langle \mathcal{P}(pp, u, w), \mathcal{V}(\sigma, u; \varrho)\rangle : \\ (pp, u, w) \in R \text{ and } \mathcal{A}(tr) = 1 \end{matrix}\right]$$

$$\approx \Pr\left[\begin{matrix} pp \leftarrow \mathrm{PGen}(1^\lambda); (u, w, \varrho) \leftarrow \mathcal{A}(pp); (tr, b) \leftarrow \mathcal{S}(pp, u, \varrho) : \\ (pp, u, w) \in R \text{ and } \mathcal{A}(tr) = 1 \end{matrix}\right],$$

where ϱ *is the randomness used by the verifier.*

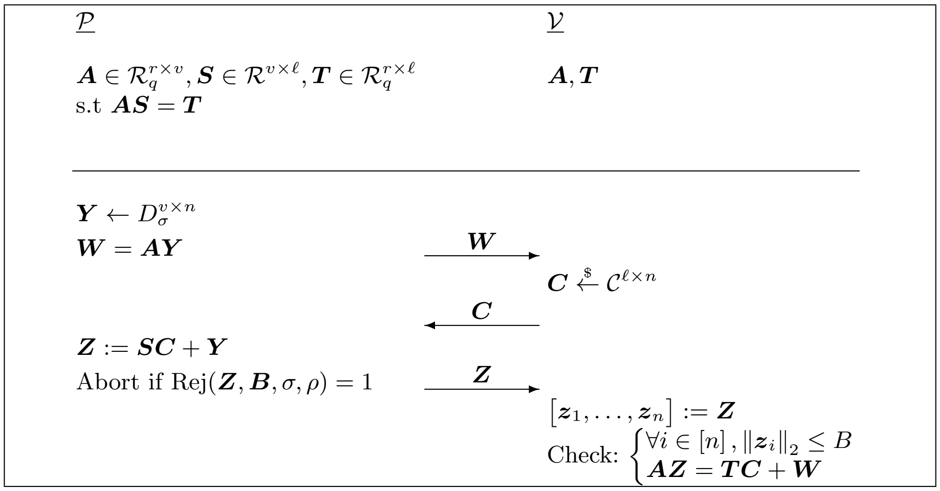
Full Zero-Knowledge. In real life applications special honest verifier zero-knowledge may not suffice since a malicious verifier may give non-random challenges. However, it is easy to convert an SHVZK argument into a full zero-knowledge argument secure against *arbitrary* verifiers in the common reference string model using standard techniques, and when using the Fiat-Shamir heuristic to make the argument non-interactive SHVZK suffices to get zero-knowledge in the random oracle model.

3 Amortized Proofs of Knowledge

We will consider amortized proofs of knowledge for preimages of the Ajtai one-way function. Formally, given a matrix $\boldsymbol{A} \in \mathcal{R}_q^{r \times v}$ the relation we want to give a zero-knowledge proof of knowledge for is

$$R = \left\{ \begin{array}{c|} (pp, u, w) = ((q, \ell, \beta, \mathcal{R}, \boldsymbol{A}, c), \boldsymbol{T}, \boldsymbol{S}) \\ \\ (\boldsymbol{A}, \boldsymbol{S}, \boldsymbol{T}) \in \mathcal{R}_q^{r \times v} \times \mathcal{R}^{v \times \ell} \times \mathcal{R}_q^{r \times \ell} \wedge \ \boldsymbol{A}\boldsymbol{S} = c \cdot \boldsymbol{T} \wedge [\|\boldsymbol{s}_i\|_2 \leq \beta]_{i \in [\ell]} \end{array} \right\}$$

with $\boldsymbol{S} = [\boldsymbol{s}_1, \cdots, \boldsymbol{s}_\ell]$ where $\mathcal{R}$ is implicitly fixed in advance. The multiplier c depends on the instantiation of the proof: for $\mathcal{R} = \mathbb{Z}$ our proof achieves $c = 1$ and is exact, while for $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$ it only guarantees that $c = 2$.



$\underline{\mathcal{P}}$		$\underline{\mathcal{V}}$
$\boldsymbol{A} \in \mathcal{R}_q^{r\times v}, \boldsymbol{S} \in \mathcal{R}^{v\times \ell}, \boldsymbol{T} \in \mathcal{R}_q^{r\times \ell}$ s.t $\boldsymbol{AS} = \boldsymbol{T}$		$\boldsymbol{A}, \boldsymbol{T}$
$\boldsymbol{Y} \leftarrow D_\sigma^{v\times n}$		
$\boldsymbol{W} = \boldsymbol{AY}$	$\xrightarrow{\boldsymbol{W}}$	
		$\boldsymbol{C} \overset{\$}{\leftarrow} \mathcal{C}^{\ell\times n}$
	$\xleftarrow{\boldsymbol{C}}$	
$\boldsymbol{Z} := \boldsymbol{SC} + \boldsymbol{Y}$		
Abort if $\mathrm{Rej}(\boldsymbol{Z}, \boldsymbol{B}, \sigma, \rho) = 1$	$\xrightarrow{\boldsymbol{Z}}$	
		$[\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n] := \boldsymbol{Z}$
		Check: $\begin{cases} \forall i \in [n], \|\boldsymbol{z}_i\|_2 \leq B \\ \boldsymbol{AZ} = \boldsymbol{TC} + \boldsymbol{W} \end{cases}$

Fig. 1: Amortized proof for ℓ equations. The ring $\mathcal{R}$ can be either $\mathbb{Z}$ or $\mathbb{Z}[X]/(X^d+1)$, the challenge set $\mathcal{C}$ will be respectively $\{0,1\}$ or $\{0\} \bigcup \{\pm X^j\}_{j<d}$

We consider a generalization of Σ-Protocols in which honest instances only complete with some constant probability $1/\rho$, this is to accommodate the fact that the rejection sampling step described in Lemma 1 only outputs 1 with probability $1/\rho$. In practice such a restriction is not too inconvenient: though the interactive protocol has to be repeated an average of ρ times to terminate, what we are interested in is usually the non-interactive protocol obtained by using the Fiat-Shamir transform, in which case the prover only has to output a proof when she obtains a challenge which passes the rejection step.

In our zero-knowledge proof, the prover will want to output a matrix $\boldsymbol{Z}$ whose distribution should be independent of the secret matrix $\boldsymbol{S}$. During the protocol, the prover obtains $\boldsymbol{Z}' = \boldsymbol{B} + \boldsymbol{Y}$ where $\boldsymbol{B}$ depends on the secret $\boldsymbol{S}$ and $\boldsymbol{Y}$ is a "masking" matrix each of whose coefficients is a discrete Gaussian with standard deviation σ. To remove the dependency of $\boldsymbol{Z}'$ on $\boldsymbol{B}$, we use the rejection sampling procedure from [Lyu12] in Algorithm 1, which has the properties described in Lemma 1.

Algorithm 1 $\mathrm{Rej}(\boldsymbol{Z}, \boldsymbol{B}, \sigma, \rho)$

```
$u \leftarrow [0,1)$
if $u > \frac{1}{\rho} \cdot \exp\left(\frac{-2\langle \boldsymbol{Z}, \boldsymbol{B}\rangle + \|\boldsymbol{B}\|^2}{2\sigma^2}\right)$ then
    return 0
else
    return 1
end if
```

Lemma 1 ([Lyu12]). *Let $\boldsymbol{B} \in \mathcal{R}^{r \times n}$ be any matrix. Consider a procedure that samples a $\boldsymbol{Y} \leftarrow D_\sigma^{r \times n}$ and then returns the output of* $\mathrm{Rej}(\boldsymbol{Z} := \boldsymbol{Y} + \boldsymbol{B}, \boldsymbol{B}, \sigma, \rho)$ *where $\sigma \geq \frac{12}{\ln \rho} \cdot \|\boldsymbol{B}\|$. The probability that this procedure outputs* 1 *is within* 2^{-100} *of* $1/\rho$*. The distribution of* $\boldsymbol{Z}$*, conditioned on the output being* 1*, is within statistical distance of* 2^{-100} *of* $D_\sigma^{r \times n}$*.*

We give a useful lemma for knowledge extraction. In essence this lemma will be used to show that a prover who can output a verifying output for a challenge $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_\ell$ has a high probability of also being able to answer a challenge $\boldsymbol{c}_1', \boldsymbol{c}_2, \ldots, \boldsymbol{c}_\ell$ in which only $\boldsymbol{c}_1' \neq \boldsymbol{c}_1$.

Lemma 2 ([Dam10]). *Let $\boldsymbol{H} \in \{0,1\}^{\ell \times n}$ for some $n, \ell > 1$, such that a fraction ε of the inputs of $\boldsymbol{H}$ are 1. We say that a row of $\boldsymbol{H}$ is "heavy" if it contains a fraction at least $\varepsilon/2$ of ones. Then more than half of the ones in $\boldsymbol{H}$ are located in heavy rows.*

We describe our proof system in Figure 1. Our first instantiation is with $\mathcal{R} = \mathbb{Z}$ in which case the one-way function will rely on the SIS problem and the challenge set will be $\mathcal{C}^{\ell \times n}$ for $\mathcal{C} = \{0,1\}$, this solution allows the extractor of the protocol to obtain exact preimages of the $\boldsymbol{t}_i$ and requires $n \geq \lambda + 2$. This ensures that communication only grows linearly in λ regardless of the size of ℓ (since $\boldsymbol{Z} \in \mathbb{Z}_q^{v \times n}$).

Theorem 1. *Let $\mathcal{R} = \mathbb{Z}$, $\mathcal{C} = \{0,1\}$, $v, r = poly(\lambda)$, and $n \geq \lambda + 2$. Let $s > 0$ be an upper bound on $s_1(\boldsymbol{S})$, $\rho > 1$ be a constant, $\sigma \in \mathbb{R}$ be such that $\sigma \geq \frac{12}{\ln \rho} s \sqrt{\ell n}$, and $B = \sqrt{2v}\sigma$. Then the protocol described in Figure 1 is a zero-knowledge proof of knowledge for R.*

Proof. We will prove correctness and zero-knowledge here as the proofs are straightforward and very similar to prior works. We will however defer the proof of soundness to Lemma 3.

Correctness: If $\mathcal{P}$ and $\mathcal{V}$ are honest then the probability of abort is exponentially close to $1 - 1/\rho$ since $\|\boldsymbol{SC}\|_2 \leq s_1(\boldsymbol{S}) \|\boldsymbol{C}\|_2 \leq s\sqrt{\ell n}$. The equation verified by $\mathcal{V}$ is true by construction of $\boldsymbol{Z}$. Since each coefficient of $\boldsymbol{Z}$ is statistically close to D_σ, then according to (4) we have $\|\boldsymbol{z}_i\|_2 \leq \sqrt{2v}\sigma$ with overwhelming probability.

Honest-Verifier Zero-Knowledge: We will now prove that our protocol is honest-verifier zero-knowledge. More concretely, we show that the protocol is zero-knowledge when the prover does not abort prior to sending $\boldsymbol{Z}$. The reason that this is enough for practical purposes is that HVZK Σ-protocols can be turned into non-interactive proofs via the Fiat-Shamir transform. The non-interactive protocol generates the challenge $\boldsymbol{C}$ as the hash of $\boldsymbol{W}$ and $\boldsymbol{T}$, and otherwise repeats the prover's part of the protocol until a non-abort occurs, whereupon the prover outputs the transcript $(\boldsymbol{W}, \boldsymbol{C}, \boldsymbol{Z})$. Only the non-aborting transcripts will be seen by $\mathcal{V}$, and thus only they need to be simulated. Further below we will also sketch how to modify our protocol to obtain an interactive zero-knowledge

proof.
Let $\mathcal{S}(\boldsymbol{A}, \boldsymbol{T})$ be the following PPT algorithm:

1. Sample $\boldsymbol{C} \leftarrow \{0,1\}^{\ell \times n}$
2. Sample $\boldsymbol{Z} \leftarrow D_\sigma^{v \times n}$
3. Set $\boldsymbol{W} = \boldsymbol{AZ} - \boldsymbol{TC}$
4. Output $(\boldsymbol{W}, \boldsymbol{C}, \boldsymbol{Z})$

It is clear that $\boldsymbol{Z}$ verifies with overwhelming probability. We already showed in the section on correctness that in the real protocol when no abort occurs the distribution of $\boldsymbol{Z}$ is within statistical distance 2^{-100} of $D_\sigma^{v \times n}$. Since $\boldsymbol{W}$ is completely determined by $\boldsymbol{A}, \boldsymbol{T}, \boldsymbol{Z}$ and $\boldsymbol{C}$, the distribution of $(\boldsymbol{W}, \boldsymbol{C}, \boldsymbol{Z})$ output by $\mathcal{S}$ is within 2^{-100} of the distribution of these variables in the actual non-aborting run of the protocol.

To turn our proof into a full interactive HVZK proof, one can use the above simulator together with a standard transformation: in the first message of the protocol, $\mathcal{P}$ will send a statistically hiding commitment of $\boldsymbol{W}$ to the verifier. Later in the third round, she will then send both the opening and the message $\boldsymbol{Z}$, given that the protocol would not abort. The above simulator $\mathcal{S}(\boldsymbol{A}, \boldsymbol{T})$ can then, in the beginning of the protocol, flip a coin to determine if the simulation is aborting. If so, then it can just commit to a uniformly random value, and otherwise to the correct value $\boldsymbol{W}$. In order to make the protocol secure against arbitrary verifiers one can run an interactive coin-flipping protocol to generate $\boldsymbol{C}$.

Lemma 3 (Knowledge Soundness). *For any prover $\mathcal{P}^*$ who succeeds with probability $\varepsilon > 2^{-\lambda}$ (i.e. $\geq 2^{-n+2}$) over her random tape $\chi \in \{0,1\}^x$ and the challenge choice $\boldsymbol{C} \xleftarrow{\$} \mathcal{C}^{\ell \times n}$, there exists a knowledge extractor $\mathcal{E}$ running in expected time $\mathsf{poly}(\lambda)/\varepsilon$ who can extract a witness $\boldsymbol{S}' := (\boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_\ell) \in \mathcal{R}^{v \times \ell}$, such that $\boldsymbol{AS}' = \boldsymbol{T}$, and $\forall i \in [\ell]$ $\|\boldsymbol{s}'_i\|_2 \leq 2B$.*

Proof. For $i \in [\ell]$, let $\boldsymbol{t}_i \in \mathcal{R}^n$ be the ith column of $\boldsymbol{T}$, and $\boldsymbol{c}_i^T \in \mathcal{R}^{1 \times n}$ be the ith row of $\boldsymbol{C}$ (note that $\boldsymbol{c}_i^T$ are not the transpose of the columns of $\boldsymbol{C}$ but really its rows). Note that $\boldsymbol{t}_i \boldsymbol{c}_i^T \in \mathcal{R}^{r \times n}$ and $\boldsymbol{TC} = \sum_{i=1}^{\ell} \boldsymbol{t}_i \boldsymbol{c}_i^T$. For any fixed i, we describe an extractor $\mathcal{E}_i$ who can extract a preimage of $\boldsymbol{t}_i$ of norm less than $2B$ in expected $O(1/\varepsilon)$ executions, and the full result follows by running each extractor (of which there are $\ell = \mathsf{poly}(\lambda)$).
Consider a matrix $\boldsymbol{H}_i \in \{0,1\}^{2^{n(\ell-1)+x} \times 2^n}$ whose rows are indexed by the value of $(\chi, \boldsymbol{c}_1^T, \ldots, \boldsymbol{c}_{i-1}^T, \boldsymbol{c}_{i+1}^T, \ldots, \boldsymbol{c}_\ell^T)$ and whose columns are indexed by the value of $\boldsymbol{c}_i^T$. An entry of $\boldsymbol{H}_i$ will be 1 if $\mathcal{P}^*$ succeeds for the corresponding challenge (i.e. produces an accepting $\boldsymbol{Z}$). We will say that a row of $\boldsymbol{H}_i$ is "heavy" if it contains a fraction of at least $\varepsilon/2$ ones, i.e. if it contains more than $2^k * \varepsilon/2 > 2$ ones. The extractor $\mathcal{E}_i$ will proceed as follow:

1. Run $\mathcal{P}^*$ on random challenges $\boldsymbol{C}'$ until it succeeds, and obtains $\boldsymbol{Z}'$ that verifies. This takes expected time $1/\varepsilon$.
2. Run $\mathcal{P}^*$ on random challenges $\boldsymbol{C}''$ where $\forall j \neq i, \boldsymbol{c}_j''^T = \boldsymbol{c}_j'^T$ and $\boldsymbol{c}_i''^T$ is freshly sampled. If after λ/ε attempts $\mathcal{P}^*$ has not output a valid response $\boldsymbol{Z}''$, abort.

The extractor $\mathcal{E}_i$ runs in expected time $poly(\lambda)/\varepsilon$, and aborts with probability less than $1/2 + 2^{-\lambda}$. The running time is clear from the definition of $\mathcal{E}_i$. To compute the abort probability note that in step 2 all the challenges $\boldsymbol{C}''$ considered are in the same row of $\boldsymbol{H}_i$ as $\boldsymbol{C}'$, if we call Abort the event where $\mathcal{E}_i$ aborts and Heavy the event that $\boldsymbol{C}'$ is in a row of $\boldsymbol{H}_i$, we have:

$$\Pr\left[\mathsf{Abort}\right] = \Pr\left[\mathsf{Abort}\middle|\mathsf{Heavy}\right]\Pr\left[\mathsf{Heavy}\right] + \Pr\left[\mathsf{Abort}\middle|\neg\mathsf{Heavy}\right]\Pr\left[\neg\mathsf{Heavy}\right]$$

According to Lemma 2, $\Pr\left[\neg\mathsf{Heavy}\right] < 1/2$. On the other hand if the row is heavy then for a random sample in this row $\mathcal{P}^*$ has probability at least $\varepsilon/2 - 2^{-n} > \varepsilon/4$ of outputting a valid answer (the probability is $\varepsilon/2 - 2^{-n}$ and not $\varepsilon/2$ because we want a reply for a challenge different from $\boldsymbol{C}'$). Thus the probability that $\mathcal{P}^*$ does not succeed on any of the λ/ε challenges $\boldsymbol{C}''$ is $\Pr\left[\mathsf{Abort}\middle|\mathsf{Heavy}\right] < (1-\varepsilon/4)^{\lambda\varepsilon} < e^{-4\lambda} < 2^{-\lambda}$, and therefore $\Pr\left[\mathsf{Abort}\right] < 1/2 + 2^{-\lambda}$. By running $\mathcal{E}_i$ $O(\lambda)$ times we obtain an extractor that runs in expected time $poly(\lambda)/\varepsilon$ and outputs two valid pairs $\boldsymbol{C}', \boldsymbol{Z}'$ and $\boldsymbol{C}'', \boldsymbol{Z}''$ such that $\forall j \neq i, \boldsymbol{c}'^T_j = \boldsymbol{c}''^T_j$, and $\boldsymbol{c}'^T_i \neq \boldsymbol{c}''^T_i$.
Since both transcripts verify we know that $\boldsymbol{A}\boldsymbol{Z}' = \boldsymbol{T}\boldsymbol{C}' + \boldsymbol{W} = \sum_{j=1}^{r} \boldsymbol{t}_j \boldsymbol{c}'^T_j + \boldsymbol{W}$ and that $\boldsymbol{A}\boldsymbol{Z}'' = \boldsymbol{T}\boldsymbol{C}'' + \boldsymbol{W} = \sum_{j=1}^{r} \boldsymbol{t}_j \boldsymbol{c}''^T_j + \boldsymbol{W}$, which implies that $\boldsymbol{A}(\boldsymbol{Z}' - \boldsymbol{Z}'') = \sum_{j=1}^{r} \boldsymbol{t}_j(\boldsymbol{c}'^T_j - \boldsymbol{c}''^T_j) = \boldsymbol{t}_i(\boldsymbol{c}'^T_i - \boldsymbol{c}''^T_i)$ If we consider an index $l \in [\ell]$ such that $\boldsymbol{c}'^T_i[l] \neq \boldsymbol{c}''^T_i[l]$, and assume w.l.o.g that $\boldsymbol{c}'^T_i[l] - \boldsymbol{c}''^T_i[l] = 1$, then by only considering the l^{th} column of the previous equation we obtain $\boldsymbol{A}(\boldsymbol{z}'_l - \boldsymbol{z}''_l) = \boldsymbol{t}_i$ where $\left\|\boldsymbol{z}'_l - \boldsymbol{z}''_l\right\|_2 \leq 2B$.

Our second instantiation uses $\mathcal{R} = \mathbb{Z}\left[X\right]/(X^d + 1)$ and $\mathcal{C} = \{0\} \bigcup \left\{\pm X^j\right\}_{j<d}$. This protocol only proves R with $c = 2$, i.e. the extractor will only obtain preimages of $2\boldsymbol{t}_i$ but the number of columns in the response matrix $\boldsymbol{Z}$ can be reduced by a factor of $\log(2d+1)$ as the soundness now only requires that $n\log(2d+1) \geq \lambda + 2$. It is worth noting that in this protocol the values of r and v would typically be chosen to be around d times smaller than in the instantiation with $\mathcal{R} = \mathbb{Z}$, because $\boldsymbol{A}$ will be a matrix of polynomials of degree d. We first give a lemma about the difference on monomials in $\mathbb{Z}\left[X\right]/(X^d + 1)$ which will be useful in the extraction.

Lemma 4 ([BCK$^+$14] Lemma 3.2). *Let d be a power of 2, let $a, b \in \{\pm X^i \, : \, i \geq 0\} \cup \{0\}$. Then $2(a-b)^{-1} \mod X^d + 1$ only has coefficients in $\{-1, 0, 1\}$. In particular $\left\|2(a-b)^{-1}\right\|_2 \leq \sqrt{d}$.*

Theorem 2. *Let $\mathcal{R} = \mathbb{Z}\left[X\right]/X^d + 1$, $\mathcal{C} = \{0\} \bigcup \left\{\pm X^j\right\}$, $v, r = poly(\lambda)$, and $n \geq (\lambda + 2)/\log(2d+1)$. Let $s \in \mathbb{R}$ be an upper bound on $s_1(\boldsymbol{S})$, $\rho > 1$ be a constant, $\sigma \in \mathbb{R}$ be such that $\sigma \geq \frac{12}{\ln \rho} s\sqrt{\ell n}$, and $B = \sqrt{2md}\sigma$. Then the protocol described in Figure 1 is a SHVZK proof of knowledge.*

Proof. The proofs for correctness and zero-knowledge are nearly identical to the ones of Theorem 1. We will prove soundness in Lemma 5.

Lemma 5 (Knowledge Soundness). *For any prover $\mathcal{P}^*$ who succeeds with probability $\varepsilon > 2^{-\lambda}(\geq 2^{-n\log(2d+1)+2})$ over his random tape $\chi \in \{0,1\}^x$ and* *the challenge choice* $\boldsymbol{C} \leftarrow \mathcal{C}^{\ell \times n}$ *there exists a knowledge extractor* $\mathcal{E}$ *who can extract a witness* $\boldsymbol{S}' := (\boldsymbol{s}'_1, \dots, \boldsymbol{s}'_\ell) \in \mathcal{R}^{v \times \ell}$*, such that* $\boldsymbol{A}\boldsymbol{S}' = 2\boldsymbol{T}$*, and* $\forall i \in [\ell]$ $\|\boldsymbol{s}'_i\|_2 \leq 2\sqrt{d}B$*, in expected time poly*$(\lambda)/\varepsilon$*.*

Proof. The first part of the proof (obtaining $\boldsymbol{C}', \boldsymbol{Z}'$ and $\boldsymbol{C}'', \boldsymbol{Z}''$) is identical to the one of Lemma 3 except for the fact that the matrix $\boldsymbol{H}_i$ has different dimensions. Let $\delta = \log(2d+1)$. Since for each $j \in [\ell]$, $\boldsymbol{c}_j^T$ is sampled from a set of size $2^{n\delta}$, we have $\boldsymbol{H}_i \in \{0,1\}^{2^{n\delta(\ell-1)+x} \times 2^{n\delta}}$. The heavy rows of $\boldsymbol{H}_i$ will contain $2^{n\delta}\varepsilon/2 > 2$ ones, and the extractor can proceed as in the proof of Lemma 3. Assume that $\mathcal{E}_i$ has extracted $\boldsymbol{C}', \boldsymbol{Z}$" and $\boldsymbol{C}'', \boldsymbol{Z}''$ such that $\forall j \neq i, \boldsymbol{c}_j'^T = \boldsymbol{c}_j''^T$, and $\boldsymbol{c}_i'^T \neq \boldsymbol{c}_i''^T$. As previously we have $\boldsymbol{A}(\boldsymbol{Z}' - \boldsymbol{Z}'') = \sum_{j=1}^{\ell} \boldsymbol{t}_j(\boldsymbol{c}_j'^T - \boldsymbol{c}_j''^T) = \boldsymbol{t}_i(\boldsymbol{c}_i'^T - \boldsymbol{c}_i''^T)$ If we consider an index $l \in [\ell]$ such that $\boldsymbol{c}_i'^T[l] \neq \boldsymbol{c}_i''^T[l]$, since $\mathcal{C} = \{0\} \bigcup \left\{\pm X^j\right\}_{0 \leq j \leq d-1}$, we have according to Lemma 4 that there exists a $\boldsymbol{g} \in \mathcal{R}$ such that $2^{-1}(\boldsymbol{c}_i'^T[l] - \boldsymbol{c}_i''^T[l])\boldsymbol{g} = 1$ and $\|\boldsymbol{g}\|_2 \leq \sqrt{d}$. Hence $\boldsymbol{A}(\boldsymbol{z}'_l - \boldsymbol{z}''_l)\boldsymbol{g} = 2\boldsymbol{t}_i \cdot 2^{-1}(\boldsymbol{c}_i'^T[l] - \boldsymbol{c}_i''^T[l])\boldsymbol{g} = 2\boldsymbol{t}_i$, with $\|(\boldsymbol{z}'_l - \boldsymbol{z}''_l)\boldsymbol{g}\|_2 \leq 2\sqrt{d}B$.

4 Argument for the Satisfiability of an Arithmetic Circuit

In this section, we show how to construct arguments for the satisfiability of an arithmetic circuit based on the SIS assumption. We take inspiration from the arguments of [Gro09a, BCC+16] which rely on homomorphic commitments based on the hardness of discrete logarithm and translate them into the lattice settings. We obtain sublinear communication arguments with improved computational efficiency with respect to [Gro09a, BCC+16].

At a high level, [BCC+16] reduces the satisfiability of an arithmetic circuit to the verification of two sets of constraints: multiplication constraints, arising from multiplication gates; linear constraints, arising from additions and multiplication by constant gates. Then, it shows how to embed each of these sets of constraints into a polynomial equation over $\mathbb{Z}_p$. An argument for the satisfiability of an arithmetic circuit can then be constructed by giving arguments for the satisfiability of such polynomial equations, evaluating at random challenge points and using the Schwarz-Zippel lemma to argue soundness.

We give arithmetic circuit arguments over $\mathbb{Z}_p$ for much smaller p (e.g. $p = poly(\lambda)$). Therefore, a straightforward translation of the above approach yield arguments which only have inverse polynomial soundness error, as $O(1/p)$ is inverse-polynomial in the security parameter in this setting. The soundness error could be reduced by repeated the protocol multiple times in parallel, resulting into a significant computational and communication overhead.

Therefore, we devise a more complex embedding technique in order to apply the Schwarz-Zippel lemma over larger fields. Cramer, Damgård and Keller give in [CDK14] an amortised proof of knowledge of k commitments over $\mathbb{Z}_p$ are embedded into $GF(p^k)$, with soundness error $O(1/p^k)$. We follow a similar approach and embed the constraints for the satisfiability of the circuit into polynomial equations over an extension field. While [CDK14] only give a proof of knowledge, we also construct a product argument for the openings of k commitments over $\mathbb{Z}_p$ embedded into an extension field of degree $2k$ with soundness $O(1/p^{2k})$.

We start by recalling how [BCC+16] embedded the satisfiability of an arithmetic circuit into a polynomial equations over $\mathbb{Z}_p$ and then extend it to $GF(p^{2k})$.

Reduction of Circuit Satisfiability to a Hadamard Matrix Product and Linear Constraints over $\mathbb{Z}_p$. We consider arithmetic circuits with fan-in 2 addition and multiplication gates. Multiplication gates are directly represented as equations of the form $a \cdot b = c$, and we refer to a, b, c as the left, right and output wires, respectively.

The satisfiability of an arithmetic circuit can be described as a system of equations in the entries of three matrices A, B, C. The multiplication gates define a set of N equations $A \circ B = C$, where $\circ$ is the Hadamard (entry-wise) product.

The circuit description also contains constraints on the wires between multiplication gates. Denoting the rows of the matrices A, B, C as

$$\boldsymbol{a}_i = (a_{i,1}, \dots, a_{i,n}) \quad \boldsymbol{b}_i = (b_{i,1}, \dots, b_{i,n}) \quad \boldsymbol{c}_i = (c_{i,1}, \dots, c_{i,n}) \quad \text{for } i \in \{1, \dots, m\}$$

these constraints can be expressed as $U < 2N$ linear equations of inputs and outputs of multiplication gates of the form

$$\sum_{i=1}^{m} \boldsymbol{a}_i \cdot \boldsymbol{w}_{u,a,i} + \sum_{i=1}^{m} \boldsymbol{b}_i \cdot \boldsymbol{w}_{u,b,i} + \sum_{i=1}^{m} \boldsymbol{c}_i \cdot \boldsymbol{w}_{u,c,i} = K_u \qquad \text{for } u \in \{1, \dots, U\} \tag{6}$$

for constant vectors $\boldsymbol{w}_{u,a,i}, \boldsymbol{w}_{u,b,i}, \boldsymbol{w}_{u,c,i}$ and scalars K_u. We refer to [BCC+16] for a more detailed explanation of this process.

In total, to capture all multiplications and linear constraints, we have $N + U$ equations that the wires must satisfy in order for the circuit to be satisfiable.

Reduction to Two Polynomial Equations. Let Y be a formal indeterminate. We will reduce the $N + U$ equations above to a two polynomial equations in Y by embedding distinct equations into distinct powers of Y. In our argument we will then require the prover to prove that these two equations hold when replacing Y by a random challenge received from the verifier. More explanation behind this process can be found in the full version of this paper.

Let us define $\boldsymbol{w}_{a,i}(Y) = \sum_{u=1}^{U} \boldsymbol{w}_{u,a,i} Y^{N+1+u}$, $\boldsymbol{w}_{b,i}(Y) = \sum_{u=1}^{U} \boldsymbol{w}_{u,b,i} Y^{N+1+u}$ $\boldsymbol{w}_{c,i}(Y) = \sum_{u=1}^{U} \boldsymbol{w}_{u,c,i} Y^{N+1+u}$, $K(Y) = \sum_{u=1}^{U} K_u Y^{N+1+u}$

Then the circuit is satisfied if and only if

$$\sum_{i=1}^{m} \boldsymbol{a}_i \cdot \boldsymbol{w}_{a,i}(Y) + \sum_{i=1}^{m} \boldsymbol{b}_i \cdot \boldsymbol{w}_{b,i}(Y) + \sum_{i=1}^{m} \boldsymbol{c}_i \cdot \boldsymbol{w}_{c,i}(Y) - K(Y) = 0 \tag{7}$$

$$\sum_{i=1}^{m} \boldsymbol{a}_i \circ \boldsymbol{b}_i Y^i = \sum_{i=1}^{m} \boldsymbol{c}_i Y^i \tag{8}$$

Sublinear Communication Product Argument. To give an argument for the satisfiability of an arithmetic circuit it is sufficient to give arguments showing that (7) and (8) are satisfied. For the purpose of constructing sublinear communication arguments, we craft polynomials which will have particular terms equal to zero if and only if (7) and (8) are satisfied. This can then be proved by having the prover reveal evaluations of the polynomials at random points to the verifier, who can check that the evaluations are correct using the homomorphic property of the commitment scheme. We define $\boldsymbol{a}(X) := \boldsymbol{a}_0 + \sum_{i=1}^{m} \boldsymbol{a}_i y^i X^i$, $\boldsymbol{b}(X) := \boldsymbol{b}_{m+1} + \sum_{i=1}^{m} \boldsymbol{b}_i X^{m+1-i}$ and $\boldsymbol{c} := \sum_{i=1}^{m} \boldsymbol{c}_i y^i$.

We have designed these polynomials such that the X^{m+1} term of $\boldsymbol{a}(X) \circ \boldsymbol{b}(X)$ is equal to $\sum_{i=1}^{m} \boldsymbol{c}_i y^i$. We conclude that the X^{m+1} term of $\boldsymbol{a}(X) \circ \boldsymbol{b}(X)$ is exactly $\boldsymbol{c}$ if and only if (8) is satisfied. A similar approach can followed to embed the satisfiability of (7) into the constant term of polynomial which is tested at random challenge evaluation points.

4.1 Amortisation Over Field Extensions

We now show how to extend the previous approach to work over field extensions. This will allow us to give an efficient amortised argument for the product of openings of commitments. This will be used to give efficient arguments for the satisfiability of an arithmetic circuit achieving sublinear communication and $O(1/p^{2k})$ soundness error.

Let $GF(p^{2k}) \simeq \mathbb{Z}_p[\phi]/\langle f(\phi) \rangle$, where f is a polynomial of degree $2k$ that is irreducible over $\mathbb{Z}_p$. Our goal is to embed k elements of $\mathbb{Z}_p$ into the extension field in a way so that we can multiply two $GF(p^{2k})$ elements in a way that does not interfere with the products of the original $\mathbb{Z}_p$ elements. Let $e_1, \ldots, e_k$ be distinct interpolation points in $\mathbb{Z}_p$ (note that in particular, this forces $p > k$). Let $l_1(X), \ldots, l_k(X)$ be the Lagrange polynomials associated with the points e_i, which have degree $k-1$. Let $l_0(X) = \prod_{j=1}^{k} (X - e_i)$, which has degree k.

Now, suppose that we have $a_1, \ldots, a_k$, $b_1, \ldots, b_k$ and $c_1, \ldots, c_k$ in $\mathbb{Z}_p$ such that $a_j \cdot b_j = c_j \mod p$ for each j. By evaluating the expression at each interpolation point, we see that the following statement about polynomials holds over $\mathbb{Z}_p$: $\left(\sum_{j=1}^{k} a_j l_j(X)\right) \cdot \left(\sum_{j=1}^{k} b_j l_j(X)\right) \equiv \left(\sum_{j=1}^{k} c_j l_j(X)\right) \mod l_0(X)$.

Therefore, there are $c'_0, \ldots, c'_{k-2} \in \mathbb{Z}_p$ such that $\left(\sum_{j=1}^{k} a_j l_j(X)\right) \cdot \left(\sum_{j=1}^{k} b_j l_j(X)\right) = \left(\sum_{j=1}^{k} c_j l_j(X)\right) + l_0(X) \sum_{j=0}^{k-2} c'_j X^j$.

The degree of f is $2k$, so if we choose the basis $\mathcal{B} = \{l_1(\phi), \ldots, l_k(\phi), l_0(\phi), \phi l_0(\phi), \ldots, \phi^{k-1} l_0(\phi)$ for $GF(p^{2k})\}$, we can perform multiplications of extension field elements without any overflow modulo f interfering with the individual product relations $a_i b_i = c_i$ in $\mathbb{Z}_p$. We can therefore port he above equality into $GF(p^{2k})$ as the equality $\left(\sum_{j=1}^{k} a_j l_j(\phi)\right) \cdot \left(\sum_{j=1}^{k} b_j l_j(\phi)\right) = \left(\sum_{j=1}^{k} c_j l_j(\phi)\right) + l_0(\phi) \sum_{j=0}^{k-2} c'_j \phi^j$.

This allows one multiplication of committed values to be performed without any overflow modulo f. As we shall see in the next subsection, this is sufficient for verifying multiplication triples for arithmetic circuit satisfiability.

We also need to be able to view single commitments to elements of $\mathbb{Z}_p$ as elements of the extension field in a way that helps to verify linear consistency relations between the elements.

Now, suppose that we have $a_1, \dots, a_k$, $b_1, \dots, b_k$ and $c_1, \dots, c_k$ in $\mathbb{Z}_p$, and coefficients $w_{a,1}, \dots, w_{a,k}$, $w_{b,1}, \dots, w_{b,k}$ and $w_{c,1}, \dots, w_{c,k}$ in $\mathbb{Z}_p$ such that $\sum_{j=1}^{k} a_j w_{a,j} + \sum_{j=1}^{k} b_j w_{b,j} + \sum_{j=1}^{k} c_j w_{c,j} = K \mod p$. By comparing coefficients, we see that the following statement about polynomials holds over $\mathbb{Z}_p$: $\left(\sum_{j=1}^{k} a_j X^{j-1}\right) \cdot \left(\sum_{j=1}^{k} w_{a,j} X^{k-j}\right) + \left(\sum_{j=1}^{k} b_j X^{j-1}\right) \cdot \left(\sum_{j=1}^{k} w_{b,j} X^{k-j}\right) + \left(\sum_{j=1}^{k} c_j X^{j-1}\right) \cdot \left(\sum_{j=1}^{k} w_{c,j} X^{k-j}\right) = K X^{k-1} + \sum_{j=0, j \neq k-1}^{2k-2} K_j X^j$, where the K_j are extra terms determined from the a, b, c and w values.

If we choose the basis $\mathcal{B}' = 1, \phi, \phi^2, \dots, \phi^{2k-1}$ for $GF(p^{2k})$, we can perform multiplications of extension field elements in a way that always yields a useful linear relation in the ϕ^{k-1} term without any overflow modulo f.

By viewing multiplication in $GF(p^{2k})$ as a linear map over $\mathbb{Z}_p^{2k}$, we can simulate arithmetic in the extension field using arithmetic in $\mathbb{Z}_p^{2k}$.

Let $A_1, \dots, A_{2k} \in \mathcal{C}^{2k}$ be homomorphic commitments to single elements, $a_1, \dots, a_k \in \mathbb{Z}_p$. We can consider the tuple $\boldsymbol{A} = (A_1, \dots, A_k)$ to be a commitment to an element $\boldsymbol{a} = (a_1, \dots, a_{2k})$ of $GF(p^{2k})$. Now, if we consider $\boldsymbol{x} \in \mathbb{Z}_p^{2k}$ as an element of $GF(p^{2k})$, then there is a matrix $M_{\boldsymbol{x}}$ which simulates multiplication by $\boldsymbol{x}$ in $\mathbb{Z}_p^{2k}$ when we multiply on the left by $M_{\boldsymbol{x}}$. Since the A_i are homomorphic commitments, we can obtain a commitment to $\boldsymbol{x} * \boldsymbol{a}$ by computing $M_{\boldsymbol{x}} \boldsymbol{A}$, where $*$ represents multiplication in $GF(p^{2k})$.

Reduction of Circuit Satisfiability to a Hadamard Matrix Product and Linear Constraints over $\boldsymbol{GF(p^{2k})}$. Let $N = mnk$ be the number of multiplication gates in the arithmetic circuit. To reduce circuit satisfiability to constraints over $GF(p^{2k})$, we can consider the same polynomial equations as before, written over $GF(p^{2k})$ rather than $\mathbb{Z}_p$. We consider the rows of matrices A, B, and C as before, but this time, we label the row vectors of the matrices $\boldsymbol{a}_{i,j}, \boldsymbol{b}_{i,j}$ and $\boldsymbol{c}_{i,j} \in \mathbb{Z}_p^n$, for $1 \leq i \leq m$ and $1 \leq j \leq k$. Now, we consider the row vectors $\boldsymbol{a}_{i,1}, \dots, \boldsymbol{a}_{i,k}$, which are elements of $\mathbb{Z}_p^n$, as an element in $GF(p^{2k})^n$.

Let $a_i = (\boldsymbol{a}_{i,1}, \boldsymbol{a}_{i,2}, \dots, \boldsymbol{a}_{i,k}, \boldsymbol{0}, \dots, \boldsymbol{0})^T$ represent this element in $GF(p^{2k})^n$. Each column of the matrix represents a separate element of $GF(p^{2k})$.

Satisfiability conditions over $\mathbb{Z}_p$ were embedded using scalar products, denoted by $\cdot$, and element-wise products, denoted by $\circ$. If a and b in $\mathbb{Z}_p^{2k \times n}$ represent elements of $GF(p^{2k})^n$, then each column represents an element of $GF(p^{2k})$, and the scalar products and element-wise products of a and b are computed using the columns. We denote the element-wise product by $a \bigcirc b$ and the scalar product by $a \bigodot b$ to avoid confusion with any other matrix products on a and b.

$$a = \begin{pmatrix} \\ \boldsymbol{v}_1 \; \boldsymbol{v}_2 \; \dots \; \boldsymbol{v}_n \\ \\ \end{pmatrix}, \; b = \begin{pmatrix} \\ \boldsymbol{w}_1 \; \boldsymbol{w}_2 \; \dots \; \boldsymbol{w}_n \\ \\ \end{pmatrix}$$

$$a \bigcirc b = \left(M_{\boldsymbol{v}_1}\boldsymbol{w}_1 \; M_{\boldsymbol{v}_2}\boldsymbol{w}_2 \; \ldots \; M_{\boldsymbol{v}_n}\boldsymbol{w}_n \right)$$

$$a \bigodot b = M_{\boldsymbol{v}_1}\boldsymbol{w}_1 + M_{\boldsymbol{v}_2}\boldsymbol{w}_2 + \ldots + M_{\boldsymbol{v}_n}\boldsymbol{w}_n$$

Note that in the verification equations, although the verifier computes high powers of random challenges $\boldsymbol{x}$ and $\boldsymbol{y}$, the verifier only computes quadratic polynomials of values such as a and b which have been sent by the prover. This is important, because when we expand a and b in terms of their coefficients a_i and b_i, we see that the verifier only computes expressions which have degree 2 in the prover's secret committed wire values, embedded as elements of $GF(p^{2k})$. Therefore, considering a field extension of degree $2k$ with the basis $\mathcal{B}$ is sufficient for our purposes: we only need to ensure that a single multiplication in $GF(p^{2k})$ preserves the individual product relations embedded in the $GF(p)$ elements.

When embedding satisfiability conditions into a polynomial over $\mathbb{Z}_p$, using random challenges $x, y \in \mathbb{Z}_p$, the prover could send linear combinations of vectors $\boldsymbol{a}_i \in \mathbb{Z}_p^n$ such as $\boldsymbol{a}(x) = \boldsymbol{a}_0 + \sum_{i=1}^{m} \boldsymbol{a}_i y^i x^i$ to the verifier.

However, when embedding satisfiability conditions into a polynomial over $GF(p^{2k})$, using random challenges $\boldsymbol{x}, \boldsymbol{y} \in GF(p^{2k})$, the prover sends linear combinations of vectors $a_i \in GF(p^{2k})^n$ such as $a(x) = a_0 + \sum_{i=1}^{m} (M_{\boldsymbol{y}})^i (M_{\boldsymbol{x}})^i a_i$.

Committing and Performing Calculations in a Lattice Setting. Commitment schemes based on lattice assumptions often require messages to be 'small' elements of the base ring in which the commitment is computed. Therefore, we consider the wire values in the arithmetic circuit to be integers in $[p]$ inside a larger ambient ring $\mathbb{Z}_q$ where the commitments are computed.

We can still simulate the action of $GF(p^{2k})$ over the integers by applying the same multiplication matrices over the integers rather than working modulo p. Whenever the prover and verifier multiply by powers of random challenges $\boldsymbol{x} \in [p]^{2k}$, they reduce powers of matrices such as $M_{\boldsymbol{x}}$ and $M_{\boldsymbol{y}}$ modulo p before applying these matrices to commitments or openings. For example, the prover will send openings a and b to the verifier: $a = \sum_{i=0}^{m} (M_{\boldsymbol{x}}^i M_{\boldsymbol{y}}^i \mod p) a_i$ and $b = \sum_{i=0}^{m} (M_{\boldsymbol{x}}^{(m+1-i)} \mod p) b_i$.

For this reason, the verification equations will compare quantities that are congruent modulo p, but not equal over the integers, or in $\mathbb{Z}_q$, as the prover and verifier will have computed and reduced various terms modulo p, but performed this reduction at different times during the computation. Therefore, the prover will send an additional commitment $\boldsymbol{D}$ containing a message which is a multiple of p and corrects the discrepancy.

5 Parameter Selection

In this section we introduce notation for the parameters in our arithmetic circuit argument, and specify the choice of values in our arguments to ensure asymptotic security. Due to the large number of different variables used in the arithmetic circuit argument, and the fact that the arithmetic circuit argument and earlier proof of knowledge are quite independent of one another, we redefine certain variable names which were used earlier on for use in the arithmetic circuit argument. Parameter λ is dictated by the desired security level, and p and N come from the arithmetic circuit whose satisfiability is to be proven. All other parameters are derived from the table below, can be written in terms of λ, p and N, and are chosen in order to ensure that the commitment scheme is binding on a large enough message space for security.

Parameters and Asymptotic Sizes. In order to satisfy the constraints above, we choose the parameters in Table 2. Let λ be the security parameter, and suppose that we wish to verify an arithmetic circuit with N gates, over $\mathbb{Z}_p$.

Parameter	Size	Description
λ		Security parameter for our arguments
p	$poly(\lambda)$	Underlying field for the arithmetic circuit
N	$kmn = poly(\lambda)$	Number of multiplication gates in the arithmetic circuit
P	$O(nk^2m^2p^2)$	Maximum size of elements committed by honest prover
B	$O(PN)$	Soundness slack from proof of knowledge
P'	$P' = BP$	Commitment scheme must be binding up to elements in $[P']$.
n	$n \approx \sqrt{\frac{Nr\log q}{\lambda \log N\lambda p}}$	Controls length of vectors in the SAT argument
k	$k \approx \lambda/\log_2 p$	Controls soundness error of the SAT argument
m	$m = N/kn$	Number of commitments in SAT argument is $O(mk)$
q	$q \approx P'\sqrt{r}$	Modulus for SIS instances.
r	$r = O(\log n)$	Commitments lie in $\mathbb{Z}_q^d$.

Table 2: Parameter choices for our arithmetic circuit argument.

6 Product Argument

The following protocol allows the prover to prove that they know $N = nmk$ triples satisfying multiplicative relations.

We give parameters for our protocol in Section 5.

Consider the commitment scheme $\mathrm{Com}_{ck} : \mathbb{Z}_q^n \times \mathbb{Z}_q^{n'} \mapsto \mathcal{C}$ introduced earlier in section 2.2, where ck consists of the public matrices used to generate a SIS instance. Let $A \in \mathbb{Z}_q^{2k\times n}$ and $R \in \mathbb{Z}_q^{2k\times n'}$. Define the extended commitment scheme $\mathrm{Com}_{ck}{}^*$ as

$$\mathrm{Com}_{ck}{}^*(A; R) := \begin{pmatrix} \mathrm{Com}_{ck}(\boldsymbol{a}_1; \boldsymbol{r}_1) \\ \mathrm{Com}_{ck}(\boldsymbol{a}_2, \boldsymbol{r}_2) \\ \vdots \\ \mathrm{Com}_{ck}(\boldsymbol{a}_{2k}; \boldsymbol{r}_{2k}) \end{pmatrix}$$

where $\boldsymbol{a}_i \in \mathbb{Z}_p^n$, $\boldsymbol{r}_i \in \mathbb{Z}_p^{n'}$ are the row vectors of A and R.

Common Reference String: Commitment key ck. The basis $\mathcal{B}$ for the extension field $GF(p^{2k})$, which specifies how elements should be multiplied.
Statement: Description of a set of $N = kmn$ multiplication relations over $\mathbb{Z}_p$.
Prover's Witness: Values $A_i, B_i, C_i \in \mathbb{Z}_p^{k\times n}$, $1 \le i \le m$, such that $\forall i$, $A_i \circ B_i \equiv C_i \mod p$.
Argument:

$\mathcal{P}$ Since $\forall i$, $A_i \circ B_i \equiv C_i \mod p$, then for $1 \le i \le m$, we can write

$$\begin{bmatrix} A_i \\ \mathbf{0}^{k\times n} \end{bmatrix} \bigcirc \begin{bmatrix} B_i \\ \mathbf{0}^{k\times n} \end{bmatrix} = \begin{bmatrix} C_i \\ C_i' \end{bmatrix} \mod p$$

for some $C_i' \in [p]^{k\times n}$, $1 \le i \le m$, by our choice of basis $\mathcal{B}$.
The prover randomly selects $A_0, B_{m+1} \leftarrow D_{\sigma_1}^{2k\times n}$.
The prover selects α_i and β_i uniformly at random from $[p]^{k\times n'}$ and γ_i uniformly at random from $[p]^{2k\times n'}$ for $1 \le i \le m$, and selects $\alpha_0, \beta_{m+1} \leftarrow D_{\sigma_1}^{2k\times n'}$.
For $1 \le i \le m$, the prover computes

$$\boldsymbol{A}_i = \mathrm{Com}_{ck}{}^{*}\left(\begin{bmatrix} A_i \\ \mathbf{0}^{k\times n} \end{bmatrix}; \begin{bmatrix} \alpha_i \\ \mathbf{0}^{k\times n'} \end{bmatrix}\right) \qquad \boldsymbol{C}_i = \mathrm{Com}_{ck}{}^{*}\left(\begin{bmatrix} C_i \\ C_i' \end{bmatrix}; \gamma_i\right)$$
$$\boldsymbol{B}_i = \mathrm{Com}_{ck}{}^{*}\left(\begin{bmatrix} B_i \\ \mathbf{0}^{k\times n} \end{bmatrix}; \begin{bmatrix} \beta_i \\ \mathbf{0}^{k\times n'} \end{bmatrix}\right)$$

Note that by definition, $\boldsymbol{A}_i$ and $\boldsymbol{B}_i \in \mathcal{C}^{2k}$ consist of k commitments and k trivial commitments in the k final components. The prover also computes

$$\boldsymbol{A}_0 = \mathrm{Com}_{ck}{}^{*}\left(A_0; \alpha_0\right), \qquad \boldsymbol{B}_{m+1} = \mathrm{Com}_{ck}{}^{*}\left(B_{m+1}; \beta_{m+1}\right)$$

The prover sends $\{\boldsymbol{A}_i\}_{i=0}^m, \{\boldsymbol{B}_i\}_{i=1}^{m+1}, \{\boldsymbol{C}_i\}_{i=1}^m$ to the verifier.

$\mathcal{V}$ The verifier picks $\boldsymbol{y} \leftarrow [p]^{2k}$, and sends $\boldsymbol{y}$ to the prover.

$\mathcal{P}$ The prover computes polynomials $A(\boldsymbol{X}), B(\boldsymbol{X})$, which have matrix coefficients, in the indeterminate $\boldsymbol{X} \in \mathbb{Z}_q^{2k}$, and also computes C.

$$A(\boldsymbol{X}) = A_0 + \sum_{i=1}^{m} M_{\boldsymbol{X}}^i (M_{\boldsymbol{y}}^i \mod p) \begin{bmatrix} A_i \\ \mathbf{0}^{k\times n} \end{bmatrix}$$
$$B(\boldsymbol{X}) = B_{m+1} + \sum_{i=1}^{m} M_{\boldsymbol{X}}^{m+1-i} \begin{bmatrix} B_i \\ \mathbf{0}^{k\times n} \end{bmatrix}$$
$$C = \sum_{i=1}^{m} M_{\boldsymbol{y}}^i \begin{bmatrix} C_i \\ C_i' \end{bmatrix} \mod p$$

The prover computes $A(\boldsymbol{X}) \bigcirc B(\boldsymbol{X}) \mod p$.

$$A(\boldsymbol{X}) \bigcirc B(\boldsymbol{X}) \mod p \quad = \quad M_{\boldsymbol{X}}^{m+1}C \quad + \quad \sum_{l=0, l\neq m+1}^{2m} M_{\boldsymbol{X}}^{l} H_l \mod p$$

where $H_l \in [p]^{2k\times n}$.
For $0 \leq l \leq 2m, l \neq 0$, the prover selects η_l uniformly at random from $[p]^{2k\times n'}$, and computes $\boldsymbol{H}_l = \mathrm{Com}_{ck}{}^*(H_l; \eta_l)$.
The prover sends $\{\boldsymbol{H}_l\}_{l=0,l\neq m}^{2m}$ to the verifier.

$\mathcal{V}$ The verifier picks $\boldsymbol{x} \leftarrow [p]^{2k}$, and sends $\boldsymbol{x}$ to the prover.

$\mathcal{P}$ The prover computes the following values modulo p.

$$A = A_0 + \sum_{i=1}^{m} (M_{\boldsymbol{x}}^{i} M_{\boldsymbol{y}}^{i} \mod p) \begin{bmatrix} A_i \\ \mathbf{0}^{k\times n} \end{bmatrix}$$

$$\alpha = \alpha_0 + \sum_{i=1}^{m} (M_{\boldsymbol{x}}^{i} M_{\boldsymbol{y}}^{i} \mod p) \begin{bmatrix} \alpha_i \\ \mathbf{0}^{k\times n'} \end{bmatrix}$$

$$B = B_{m+1} + \sum_{i=1}^{m} (M_{\boldsymbol{x}}^{m+1-i} \mod p) \begin{bmatrix} B_i \\ \mathbf{0}^{k\times n} \end{bmatrix}$$

$$\beta = \beta_{m+1} + \sum_{i=1}^{m} (M_{\boldsymbol{x}}^{m+1-i} \mod p) \begin{bmatrix} \beta_i \\ \mathbf{0}^{k\times n'} \end{bmatrix}$$

Note that $A \equiv A(\boldsymbol{x}) \mod p$ and $B \equiv B(\boldsymbol{x}) \mod p$.
The prover computes

$$D = (A \bigcirc B \mod p) - \sum_{i=1}^{m} (M_{\boldsymbol{y}}^{i} \mod p) \begin{bmatrix} C_i \\ C'_i \end{bmatrix} - \sum_{l=0, l\neq m+1}^{2m} (M_{\boldsymbol{x}}^{l} \mod p) H_l$$

The prover randomly selects $\delta \leftarrow D_{\sigma_2}^{2k\times n'}$ and computes $\boldsymbol{D} = \mathrm{Com}_{ck}{}^*(D; \delta)$.
The prover randomly selects $E \leftarrow p \cdot D_{\sigma_3}^{2k\times n}$, $\epsilon \leftarrow D_{\sigma_4}^{2k\times n'}$ and computes $\boldsymbol{E} = \mathrm{Com}_{ck}{}^*(E; \epsilon)$. Note that E is 0 modulo p.
The prover sends $\boldsymbol{D}$ and $\boldsymbol{E}$ to the verifier.

$\mathcal{V}$ The verifier picks $\boldsymbol{z} \leftarrow [p]^{2k}$, and sends $\boldsymbol{z}$ to the prover.

$\mathcal{P}$ The prover runs $\mathsf{Rej}((A||\alpha||B||\beta), (A||\alpha||B||\beta) - (A_0||\alpha_0||B_{m+1}||\beta_{m+1}), \sigma_1, e)$, and aborts according to the result.
The prover computes the following

$$\rho = \sum_{i=1}^{m} (M_{\boldsymbol{x}}^{m+1} M_{\boldsymbol{y}}^{i} \mod p)\gamma_i + \sum_{l=0, l\neq m+1}^{2m} (M_{\boldsymbol{x}}^{l} \mod p)\eta_l + \delta$$

The prover runs $\mathsf{Rej}(\rho, \rho - \delta, \sigma_2, e)$.
The prover computes $\bar{D} = (M_{\boldsymbol{z}} \mod p)D + E$ and $\bar{\delta} = (M_{\boldsymbol{z}} \mod p)\delta + \epsilon$.
The prover runs $\mathsf{Rej}(\bar{D}/p, D/p, \sigma_3, e)$.
The prover runs $\mathsf{Rej}(\bar{\delta}, \delta, \sigma_4, e)$.
The prover sends $A, \alpha, B, \beta, \rho, \bar{D}, \bar{\delta}$ to the verifier.

$\mathcal{V}$ The prover and the verifier engage in a proof-of-knowledge, as shown in Figure 1, including every commitment sent from the prover to the verifier.
The verifier accepts if and only if

$$\begin{aligned}
\mathrm{Com}_{ck}{}^{*}(A;\alpha) &= \sum_{i=0}^{m} (M_{\boldsymbol{x}}^{i} M_{\boldsymbol{y}}^{i} \mod p)\boldsymbol{A}_i \\
\mathrm{Com}_{ck}{}^{*}(B;\beta) &= \sum_{i=1}^{m+1} (M_{\boldsymbol{x}}^{m+1-i} \mod p)\boldsymbol{B}_i \\
\mathrm{Com}_{ck}{}^{*}(A \bigcirc B \mod p;\rho) &= \sum_{i=1}^{m} (M_{\boldsymbol{x}}^{m+1} M_{\boldsymbol{y}}^{i} \mod p)\boldsymbol{C}_i + \sum_{l=0,l\neq m+1}^{2m} (M_{\boldsymbol{x}}^{l} \mod p)\boldsymbol{H}_l + \boldsymbol{D} \\
\mathrm{Com}_{ck}{}^{*}(\bar{D};\bar{\delta}) &= (M_{\boldsymbol{z}} \mod p)\boldsymbol{D} + \boldsymbol{E} \\
\bar{D} = 0 \mod p \qquad & \left\|\bar{D}\right\|_2 \leq 2\sqrt{kn}\sigma_3 p \\
\|(A||\alpha||B||\beta)\|_2 \leq 4\sqrt{kn}\sigma_1 \qquad & \|\rho\|_2 \leq 2\sqrt{kn}\sigma_2 \qquad \left\|\bar{\delta}\right\|_2 \leq 2\sqrt{kn}\sigma_4
\end{aligned}$$

and the proof-of-knowledge is accepting.

Sizes of Standard Deviations.

$$\begin{aligned}
\sigma_1 &= 48\sqrt{kn}kmp^2, & \sigma_2 &= 72\sqrt{2kn}kmp, \\
\sigma_3 &= 24\sqrt{2kn}kp(1+6kmp), & \sigma_4 &= 24\sqrt{2}k^2pn\sigma_2
\end{aligned}$$

Security Analysis.

Theorem 3. *Given the statistically hiding, computationally binding commitment scheme based on SIS, the argument for multiplication triples has statistical completeness, statistical special honest verifier zero-knowledge and computational knowledge-soundness.*

The proof of Theorem 3 can be found in the full version of this paper.

Efficiency. The above argument uses 7 moves of interaction and results in an overall 9 move argument when combined with the proof-of-knowledge subprotocols. For the product argument, the prover must send $8mk+6k$ commitments to the verifier, and $14nk$ integers as commitments openings, plus the communication for the proof-of-knowledge. Sub-linear communication is achieved by setting parameters as in Table 2. This gives communication of approximately $O(\sqrt{N \log N})$ elements of $\mathbb{Z}_q$.

For $q = \mathrm{poly}(\lambda)$, the prover's computational costs are given by $O(N \log N (\log \lambda)^2)$ bit operations for the prover. The verifier's computational costs are dominated by computing the same types of linear combinations as the prover, giving computational costs of $O(N(\log \lambda)^3)$ bit operations.

7 Linear Constraint Argument Description.

Using similar ideas to those in the multiplication protocol, in the full version of this paper, we give a protocol which allows the prover to prove that $N = nmk$ committed values satisfy the linear consistency relations

$$\sum_{i=1,j=1}^{m,k} \boldsymbol{a}_{i,j} \cdot \boldsymbol{w}_{u,a,i,j} + \sum_{i=1,j=1}^{m,k} \boldsymbol{b}_{i,j} \cdot \boldsymbol{w}_{u,b,i,j} + \sum_{i=1,j=1}^{m,k} \boldsymbol{c}_{i,j} \cdot \boldsymbol{w}_{u,c,i,j} = K_u \quad \text{for } u \in \{1, \ldots, U\} \tag{9}$$

Without loss of generality, we pad the linear consistency relations so that U is divisible by k.

The protocol, security proof, and complexity analysis are very similar to that of the argument for proving multiplication triples in the previous section.

We select parameters for our protocol in Section 5.

Security Analysis.

Theorem 4. *Given the statistically hiding, computationally binding commitment scheme based in SIS, the argument for linear consistency constraints has statistical completeness, statistical special honest verifier zero-knowledge and computational knowledge-soundness.*

The proof of Theorem 4 can be found in the full version of this paper.

Efficiency. The above argument uses 7 moves of interaction and results in an overall 9 move argument when combined with the proof-of-knowledge sub-protocols. For the product argument, the prover must send $7km+9k-1$ commitments to the verifier, and $10nk+2k$ integers as commitment openings, plus the communication for the proof-of-knowledge. The asymptotic costs of the protocol are the same as for the argument for multiplication triples in the previous section. Combined with the proof of knowledge, this gives an arithmetic circuit argument with the stated efficiency.

8 Arithmetic Circuit Argument

The product protocol given in 6 and the linear consistency protocol given in 7 imply an arithmetic circuit protocol with the same asymptotic efficiency as the two subprotocols, in which the prover forms $O(mk)$ commitments, each to n wire values in p, and runs both subprotocols in order to prove that they satisfy the arithmetic circuit, reusing the same commitments $\boldsymbol{A}_i, \boldsymbol{B}_i, \boldsymbol{C}_i$ to the wires in both subprotocols.

This yields a zero-knowledge argument for arithmetic circuit satisfiability with communication costs $O(\sqrt{N \log N})$ elements of $\mathbb{Z}_q$, computational costs of $O(N \log N)$ for the prover, and approximately $O(N)$ for the verifier.

References

AHIV17. Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In Thuraisingham et al. [TEMX17], pages 2087–2104.

Ajt96. Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108. ACM Press, May 1996.

Ban93. Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296:625–635, 1993.

BBB$^+$17. Benedikt Bunz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. Cryptology ePrint Archive, Report 2017/1066, 2017. `https://eprint.iacr.org/2017/1066`.

BCC$^+$16. Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Fischlin and Coron [FC16], pages 327–357.

BCCT12. Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In Shafi Goldwasser, editor, *ITCS 2012*, pages 326–349. ACM, January 2012.

BCCT13. Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. Recursive composition and bootstrapping for SNARKS and proof-carrying data. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 111–120. ACM Press, June 2013.

BCG$^+$17. Jonathan Bootle, Andrea Cerulli, Essam Ghadafi, Jens Groth, Mohammad Hajiabadi, and Sune K. Jakobsen. Linear-time zero-knowledge proofs for arithmetic circuit satisfiability. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 336–365. Springer, Heidelberg, December 2017.

BCK$^+$14. Fabrice Benhamouda, Jan Camenisch, Stephan Krenn, Vadim Lyubashevsky, and Gregory Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 551–572. Springer, Heidelberg, December 2014.

BD10. Rikke Bendlin and Ivan Damgård. Threshold decryption and zero-knowledge proofs for lattice-based cryptosystems. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 201–218. Springer, Heidelberg, February 2010.

BDLN16. Carsten Baum, Ivan Damgård, Kasper Green Larsen, and Michael Nielsen. How to prove knowledge of small secrets. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 478–498. Springer, Heidelberg, August 2016.

BDOP16. Carsten Baum, Ivan Damgård, Sabine Oechsner, and Chris Peikert. Efficient commitments and zero-knowledge protocols from ring-SIS with applications to lattice-based threshold cryptosystems. Cryptology ePrint Archive, Report 2016/997, 2016. `http://eprint.iacr.org/2016/997`.

BG14. Shi Bai and Steven D. Galbraith. An improved compression technique for signatures based on learning with errors. In Josh Benaloh, editor, *CT-RSA 2014*, volume 8366 of *LNCS*, pages 28–47. Springer, Heidelberg, February 2014.

BKLP15. Fabrice Benhamouda, Stephan Krenn, Vadim Lyubashevsky, and Krzysztof Pietrzak. Efficient zero-knowledge proofs for commitments from learning with errors over rings. In Günther Pernul, Peter Y. A. Ryan, and Edgar R. Weippl, editors, *ESORICS 2015, Part I*, volume 9326 of *LNCS*, pages 305–325. Springer, Heidelberg, September 2015.

CD97. Ronald Cramer and Ivan Damgård. Linear zero-knowledge - a note on efficient zero-knowledge proofs and arguments. In *29th ACM STOC*, pages 436–445. ACM Press, May 1997.

CDG$^+$17. Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In Thuraisingham et al. [TEMX17], pages 1825–1842.

CDK14. Ronald Cramer, Ivan Damgård, and Marcel Keller. On the amortized complexity of zero-knowledge protocols. *Journal of Cryptology*, 27(2):284–316, April 2014.

CDXY17. Ronald Cramer, Ivan Damgård, Chaoping Xing, and Chen Yuan. Amortized complexity of zero-knowledge proofs revisited: Achieving linear soundness slack. In Coron and Nielsen [CN17], pages 479–500.

CN17. Jean-Sébastien Coron and Jesper Buus Nielsen, editors. *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*. Springer, Heidelberg, May 2017.

Dam10. Ivan Damgård. On Σ-protocols, 2010. `http://www.cs.au.dk/~ivan/Sigma.pdf`.

DDLL13. Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal Gaussians. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 40–56. Springer, Heidelberg, August 2013.

DL12. Ivan Damgård and Adriana López-Alt. Zero-knowledge proofs with low amortized communication from lattice assumptions. In Ivan Visconti and Roberto De Prisco, editors, *SCN 12*, volume 7485 of *LNCS*, pages 38–56. Springer, Heidelberg, September 2012.

dPL17. Rafaël del Pino and Vadim Lyubashevsky. Amortization with fewer equations for proving knowledge of small secrets. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 365–394. Springer, Heidelberg, August 2017.

FC16. Marc Fischlin and Jean-Sébastien Coron, editors. *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*. Springer, Heidelberg, May 2016.

GG98. Oded Goldreich and Shafi Goldwasser. On the limits of non-approximability of lattice problems. In *30th ACM STOC*, pages 1–9. ACM Press, May 1998.

GGI$^+$15. Craig Gentry, Jens Groth, Yuval Ishai, Chris Peikert, Amit Sahai, and Adam D. Smith. Using fully homomorphic hybrid encryption to minimize non-interative zero-knowledge proofs. *Journal of Cryptology*, 28(4):820–843, October 2015.

GGPR13. Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 626–645. Springer, Heidelberg, May 2013.

GH98. Oded Goldreich and Johan Håstad. On the complexity of interactive proofs with bounded communication. *Information Processing Letters*, 1998.

GLP12. Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In

Emmanuel Prouff and Patrick Schaumont, editors, *CHES 2012*, volume 7428 of *LNCS*, pages 530–547. Springer, Heidelberg, September 2012.

GMO16. Irene Giacomelli, Jesper Madsen, and Claudio Orlandi. Zkboo: Faster zero-knowledge for boolean circuits. In *25th USENIX Security Symposium*, pages 1069–1083, 2016.

GMR85. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304. ACM Press, May 1985.

GN08. Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 31–51. Springer, Heidelberg, April 2008.

GQ88. Louis C. Guillou and Jean-Jacques Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both trasmission and memory. In C. G. Günther, editor, *EUROCRYPT'88*, volume 330 of *LNCS*, pages 123–128. Springer, Heidelberg, May 1988.

Gro09a. Jens Groth. Linear algebra with sub-linear zero-knowledge arguments. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 192–208. Springer, Heidelberg, August 2009.

Gro09b. Jens Groth. Linear algebra with sub-linear zero-knowledge arguments. In *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, pages 192–208, 2009.

Gro10a. Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340. Springer, Heidelberg, December 2010.

Gro10b. Jens Groth. A verifiable secret shuffle of homomorphic encryptions. *J. Cryptology*, 23(4):546–579, 2010.

Gro16. Jens Groth. On the size of pairing-based non-interactive arguments. In Fischlin and Coron [FC16], pages 305–326.

GVW02. Oded Goldreich, Salil P. Vadhan, and Avi Wigderson. On interactive proofs with a laconic prover. *Computational Complexity*, 11(1-2):1–53, 2002.

GW11. Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 99–108. ACM Press, June 2011.

IKOS07. Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In David S. Johnson and Uriel Feige, editors, *39th ACM STOC*, pages 21–30. ACM Press, June 2007.

Kil92. Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *24th ACM STOC*, pages 723–732. ACM Press, May 1992.

KR08. Yael Tauman Kalai and Ran Raz. Interactive PCP. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP 2008, Part II*, volume 5126 of *LNCS*, pages 536–547. Springer, Heidelberg, July 2008.

Lip12. Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 169–189. Springer, Heidelberg, March 2012.

LM06. Vadim Lyubashevsky and Daniele Micciancio. Generalized compact Knapsacks are collision resistant. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP 2006, Part II*, volume 4052 of *LNCS*, pages 144–155. Springer, Heidelberg, July 2006.

LN17. Vadim Lyubashevsky and Gregory Neven. One-shot verifiable encryption from lattices. In Coron and Nielsen [CN17], pages 293–323.

LNSW13. San Ling, Khoa Nguyen, Damien Stehlé, and Huaxiong Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 107–124. Springer, Heidelberg, February / March 2013.

Lyu09. Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, Heidelberg, December 2009.

Lyu12. Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, Heidelberg, April 2012.

MR04. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th FOCS*, pages 372–381. IEEE Computer Society Press, October 2004.

MR08. Daniele Micciancio and Oded Regev. Lattice-based cryptography. In Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors, *Chapter in Post-quantum Cryptography*, pages 147–191. Springer, 2008.

MV03. Daniele Micciancio and Salil P. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 282–298. Springer, Heidelberg, August 2003.

PHGR13. Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. In *2013 IEEE Symposium on Security and Privacy*, pages 238–252. IEEE Computer Society Press, May 2013.

PR06. Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 145–166. Springer, Heidelberg, March 2006.

Reg05. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

Sch91. Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

Ste94. Jacques Stern. A new identification scheme based on syndrome decoding. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 13–21. Springer, Heidelberg, August 1994.

TEMX17. Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors. *ACM CCS 17*. ACM Press, October / November 2017.