Batch Lattice-Based Designated-Verifier ZK-SNARKs for R1CS

Xi Lin[1,2], Han Xia[1,2], Yongqiang Li[1,2(✉)], and Mingsheng Wang[1,2]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{linxi,xiahan,liyongqiang,wangmingsheng}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

Abstract. Zero-knowledge succinct non-interactive arguments of knowledge (zk-SNARK) is a crucial cryptographic tool to achieve privacy protection and has drawn considerable attention for its appealing applications, e.g., anonymous transactions, confidential smart contracts, and scalable consensus mechanisms. Compared with the pre-quantum case, the practicability of this primitive in the post-quantum setting is still unsatisfactory, especially for the space complexity.

In this work, we generalize the LPCP-based SNARK schemes for general cyclotomic rings and propose a tighter bound in the noise analysis for non-power-of-two cyclotomic rings using the powerful basis. Secondly, we introduce the first batch SNARK scheme for rank-1 constraint system (R1CS) in $\mathbb{F}_{p^n}$ for any prime p. Then, we apply our batch SNARK schemes for R1CS in $\mathbb{F}_{2^n}$ and implement it. Using the batch technique, we can process multiple relations at the same time, thereby yielding nice amortized results. The amortized proof size is around 3KB for moderate-size circuits (the circuit size ranges from 2^{10} to 2^{14}).

To exemplify the efficiency, we present some practical examples. Initially, we integrate a rank-1 constraint system in $\mathbb{F}_{2^8}$ for the AES algorithm, which is 3.95x smaller than xJsnark (Kosba et al., 2018) in terms of the number of constraints. Subsequently, we proceed to instantiate our batch SNARK scheme for AES, MiMC, and LowMC.

Keywords: ZK-SNARKs · Post-quantum · Succinct argument

1 Introduction

Zero-knowledge proof is a cryptographic protocol that allows the prover to convince a verifier of the validity of some statement without disclosing any additional knowledge beyond the validity of the statement. The concept was introduced by [15], and there have been active research in both theory and practice.

In numerous scenarios, we want the prover to actually "know" a valid witness, that is captured by the argument of knowledge property. To facilitate practical real-world applications, certain features such as non-interactivity and succinctness are highly desirable. Non-interactive proofs only consist of a one-round

H. Duan et al. (Eds.): SecureComm 2023, LNICST 567, pp. 329–349, 2025.
https://doi.org/10.1007/978-3-031-64948-6_17

message from the prover to the verifier. Succinctness requires the proof length to be quasi-linear. These attributes constitute the succinct non-interactive arguments of knowledge, commonly referred to as (ZK)-SNARKs. It has a broad range of applications, including anonymous transactions [5], confidential smart contracts, scalable consensus mechanisms [6], and private machine learning [26].

Since the concept of SNARKs was proposed, a series of works [9,11,19] contributed to reducing the proof size and Groth [19] achieved the shortest proof size up to now. This scheme is also deployed in Zcash [5] to achieve privacy transactions. Yet, these early works primarily rely on group-based computational assumptions, which are vulnerable to attacks by quantum computers. Recently, researchers have tried to propose efficient post-quantum zk-SNARKs constructions [12,21]. All these works focus on proving one statement at a time and the parameter choice of [21] limits the verification of R1CS in $\mathbb{F}_p$ and $\mathbb{F}_{p^2}$.

In the past development of the succinct argument, one research line focused on the efficiency of succinct argument for general relations (e.g., NP problem), while another parallel research line targets the efficiency of some specific statements. For instance, proof of plaintext knowledge also has various applications, e.g., verifiable encryption, verifiable secret sharing, and group signature [27]. Concretely, many privacy-preserving applications require one party to prove that a ciphertext is correctly decrypted without revealing the secret key. For public key encryption, [10] proved the knowledge of LWE samples (equivalently, prove the knowledge of a linear relation).

For symmetric cryptography, there exists one research line focusing on designing new tailor-made symmetric algorithms or schemes [7,16–18,24] for zero-knowledge proofs, which have apparent efficiency superiority compared to traditional ones. Yet, traditional encryption algorithms are time-tested, more credible, and are implemented extensively in practice. Therefore, it still deserves to study zk-SNARKs for R1CS in a general finite field, especially in the field of characteristic 2. In a general case, it is plausible for the efficiency to undergo a decrease. Consequently, we try to combine the batch technique to mitigate such potential inefficiencies.

This motivates us the following questions: *Can we verify R1CS in the finite field $\mathbb{F}_{p^n}$ for any prime p in batch and then prove the knowledge of traditional symmetric primitives efficiently in the post-quantum setting?*

1.1 Our Results

This work conducts a comprehensive study of the batch zk-SNARKs for R1CS in the post-quantum setting. Particularly, we make contributions in three folds:

- Firstly, we generalize the LPCP-based SNARK schemes for general cyclotomic rings and propose a tighter bound in the noise analysis for non-power-of-two cyclotomic rings using the powerful basis.
- Secondly, we propose the first batch SNARK scheme for R1CS in $\mathbb{F}_{p^n}$ for any prime p. Then, we apply our batch SNARK schemes for R1CS in $\mathbb{F}_{2^n}$ and implement it. Using the batch technique, we can process multiple relations at the same time, which yields nice amortized results in Table 1.

– Thirdly, we present some practical examples. We integrate a rank-1 constraint system in $\mathbb{F}_{2^8}$ for the AES (Rijndael) algorithm, whose number of constraints is 3.95x smaller compared to xJsnark [22]. To verify the above R1CS, we overcome the difficulty incurred by the small field through the field extension. Then we implement our batch SNARK scheme for AES, MiMC [1] and LowMC [3] for comparison.

Table 1. The size and running time comparison of the batch of zk-SNARKs for verifying moderate-size circuits in $\mathbb{F}_{2^n}$.

n_c	Set I					Set II				
	$\|\pi\|$	$\|\mathsf{crs}\|$	Setup	Prover	Verifier	$\|\pi\|$	$\|\mathsf{crs}\|$	Setup	Prover	Verifier
2^{10}	4.19	33.54	3.76	0.46	4.89	2.62	19.51	2.38	0.24	2.44
2^{11}	5.23	108.76	13.37	1.44	6.01	2.84	40.79	4.72	0.48	2.66
2^{12}	6.72	203.44	24.38	2.7	7.86	3.06	86.96	11.58	1.16	2.86
2^{13}	9.98	773.01	101.78	10.06	11.65	3.28	203.68	33.87	2.69	3.09
2^{14}	19.48	2750.98	357.73	35.59	22.57	3.73	493.66	103.2	6.42	3.53
n_c	Set III					Set IV				
	$\|\pi\|$	$\|\mathsf{crs}\|$	Setup	Prover	Verifier	$\|\pi\|$	$\|\mathsf{crs}\|$	Setup	Prover	Verifier
2^{10}	2.44	17.12	3.43	0.35	2	2.23	15.61	2.18	0.26	1.42
2^{11}	2.44	35.44	6.61	0.71	2.01	2.23	29.97	4.38	0.49	1.42
2^{12}	2.44	67.34	14.1	1.35	2.01	3.06	86.96	9.54	0.49	1.42
2^{13}	2.44	130.17	30.77	2.57	2.02	2.23	120.2	27.59	1.95	1.43
2^{14}	2.83	313.08	94	6.07	2.35	3.73	493.66	87.62	4.3	1.46

$*$ The proof length is amortized and is measured in kilobytes (KB). The CRS length is compressed and amortized, measured in megabytes (MB). The setup and (online) prover time are amortized and denoted in seconds (s), while the verifier time is amortized and quantified in milliseconds (ms).

1.2 Technical Overview

LPCP-based SNARKs over General Cyclotomic Rings. Although the power-of-two cyclotomic rings have advantages in efficiency, they lack the batch structure and are incompatible with some applications, notably the majority of symmetric primitives that necessitate the plaintext modulus to be a power-of-two. This necessitates the study of the general cyclotomic rings. Let a cyclotomic ring be $R = \mathbb{Z}[X]/(\Phi_m(X))$, where m is an arbitrary positive integer. A ring element in R has multiple representations arising from distinct $\mathbb{Z}$-base, such as the power basis and the powerful basis.

Why Choose the Powerful Basis? The idea of exploiting the powerful basis comes from the HElib library [20]. It shows that the powerful basis performs well for composite numbers m compared to the power basis. Concretely, the infinity norm of different $\mathbb{Z}$-basis can be bounded by the canonical norm with a factor, such as C_m under the power basis and D_m under the powerful basis. This factor exhibits

significant variability contingent upon the number of prime powers decomposed by m and the disparity is enlarged as the number of prime powers increases. For instance, suppose m splits to four distinct prime powers, the factor associated with the power basis falls within the approximate range of 18 to 1900, whereas the factor linked to the powerful basis does not exceed 2.63. Apart from a tighter bound, the ring element represented under the powerful basis is also more efficient in the implementation. Note that the powerful basis can also be utilized to enhance the efficiency of bootstrapping fully homomorphic encryption. For instance, the tensor decomposition of cyclotomic rings used in [23] inherently incorporates the use of the powerful basis (and its dual basis). For the theory and practice considerations, we opt for the powerful basis.

Then we generalize the noise analysis of the SNARK scheme for non-power-of-two cyclotomic rings. We exploit the powerful basis to obtain a tighter bound and facilitate a fast implementation for non-power-of-two cyclotomic rings. Explicitly, the modulus q_c under power basis should be $q_c > np^2N+2pC_mCs(p\sqrt{n}N+Cs)+2pB$ while the modulus q_d will be $q_d > np^2N + 2pD_mCs(p\sqrt{n}N + Cs) + 2pB$ under the powerful basis. Then the modulus ratio q_c/q_d approximates the factor ratio C_m/D_m. This ratio depends on m, and it can be up to 722 (for m splitting to four distinct prime powers). So does the switched modulus.

Verifying R1CS in $\mathbb{F}_2^n$: A Batch Method. General cyclotomic rings enable the batch structure and verify R1CS in $\mathbb{F}_{2^n}$ in the post-quantum setting, which can not be done in the power-of-two cyclotomic rings. The reason why the power-of-two cyclotomic rings $R_p = \mathbb{Z}_p[X]/(\Phi_m(X))$ are not used to verify R1CS in $\mathbb{F}_{2^n}$ lies in the restriction that m must be coprime to p in order to avoid some algebraic awkwardness. Assuming that the cyclotomic polynomial $\Phi_m(X)$ is a product of ℓ irreducible polynomials with degree d, it can pack ℓ different rank-1 constraint systems in $\mathbb{F}_{p^d}$ into one proof. The batch encoding and decoding algorithm is inherently a ring isomorphism. As a result, a single proof size is $n'(\ell'' + 1)\log q''$ while the amortized proof size is $(n/\ell')(\ell' + 1)\log q' = d(\ell' + 1)\log q'$, which decreases largely compared to the single one. At a cost, the high degree extension non-power-of-two cyclotomic rings incur larger running times. For implementation limitation, we instantiate the linear-only encryption scheme as the RLWE scheme since the HElib library [20] supports the RLWE scheme.

Applications: Implementing SNARKs for Symmetric Primitives. At first, we build an integrated rank-1 constraint system for AES. The design idea is to separate the non-linear component and the linear component and then we make constraints for each component. The non-linear component of the round function corresponds to the S-box, and we find the algebra expression of the S-box in the [14] is also well-suited for constraint generation. Moreover, in the finite field $\mathbb{F}_{2^n}$, the XOR operation can be treated as an addition, allowing for the combination with other linear components. This reduces the number of constraints dramatically.

When considering zk-SNARKs for R1CS in $\mathbb{F}_{2^8}$, this field is insufficient for polynomial interpolation. To deal with this problem, we use the field extension and do polynomial interpolations in the extended field. We argue the feasibility

of this method in Subsect. 5.2. Then we implement our batch zk-SNARK scheme for AES, MiMC, and LowMC.

2 Preliminaries

Notations. In this paper, $\mathbb{Z}$ denotes the set of integers. We use λ and κ to denote the computational and statistical security parameters. A function $f(\lambda) > 0$ is negligible and denoted by $\mathsf{negl}(\lambda)$ if for any $c > 0$ and sufficiently large λ, $f(\lambda) < 1/\lambda^c$. A probability is said to be overwhelming if it is $1 - \mathsf{negl}(\lambda)$. A vector is denoted by a bold lowercase letter (e.g., $\boldsymbol{a}$), and a matrix is denoted by a bold uppercase letter (e.g., $\mathbf{A}$).

Definition 2.1 (Rank-1 Constraint System). *Let n, m, ℓ be a positive integer. A rank-1 constraint system $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \ell, m, \mathbb{F}\}$ $(\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{F}^{\ell \times (n+m)})$ is satisfiable if for a statement $\boldsymbol{x} \in \mathbb{F}^n$, there exists a witness $\boldsymbol{w} \in \mathbb{F}^m$ such that $\mathbf{A}(\boldsymbol{x}||\boldsymbol{w}) \circ \mathbf{B}(\boldsymbol{x}||\boldsymbol{w}) = \mathbf{C}(\boldsymbol{x}||\boldsymbol{w})$, where $\circ$ denotes the pairwise product.*

2.1 Cyclotomic Rings

Cyclotomic Ring, Canonical Embedding, Power basis, and Powerful basis. For any integer $m > 0$, the m-th cyclotomic polynomial is $\Phi_m(X) \in \mathbb{Z}[X]$ with degree $n = \phi(m)$. The quotient ring R is defined as $\mathbb{Z}[X]/(\Phi_m(X))$ and $R_q = \mathbb{Z}_q[X]/(\Phi_m(X))$ for any integer $q > 0$. Assume that $m = m_1 \cdots m_t$ is the factorization of m into prime powers. Another quotient ring can be defined as $\mathcal{R} = \mathbb{Z}[X_1, \cdots, X_t]/(\Phi_{m_1}(X_1), \cdots, \Phi_{m_t}(X_t))$. The two quotient rings are isomorphic.

Let ω be a m-th primitive root of unity, and there are n ring embeddings, e.g., $\sigma_i(\omega) = \omega^i$ for $i \in \mathbb{Z}_m^*$. The canonical embedding of $a \in \mathcal{R}$ is obtained by evaluating a at all primitive m-th root of unity: $\sigma(a) = (\sigma_i(a))_{i \in \mathbb{Z}_m^*} = (a(w^i))_{i \in \mathbb{Z}_m^*}$.

Let b be the image of X in the ring R and let b_i be the image of X_i in the ring $\mathcal{R}$. The power basis is defined as $\{b^i\}_{i \in [\phi(m)]}$ and the powerful basis is defined as $\{b_1^{i_1} \cdots b_t^{i_t}\}_{i_1 \in [\phi(m_1)], \cdots, i_t \in [\phi(m_t)]}$ for the ring $\mathcal{R}$. Then for any element $a \in \mathcal{R}$, it can be represented uniquely as $\sum_{i \in [\phi(m)]} a_i b^i$ and $\sum_{\boldsymbol{j} \in [\phi(i_1), \cdots, \phi(i_t)]} a_{\boldsymbol{j}} b_1^{i_1} \cdots b_t^{i_t}$. The coefficient vector under the power basis and the powerful basis are denoted by $\nu(a)$, $\mu(a)$. The notions ν, μ, σ extend to vectors where ν, μ, σ is applied to each entry.

Norms. We call $\|\sigma(\cdot)\|_\infty$, $\|\nu(\cdot)\|_\infty$, $\|\mu(\cdot)\|_\infty$ the canonical norm, the power basis infinity norm, and the powerful basis infinity norm respectively. As stated in [20], the power basis infinity norm, and the powerful basis infinity norm can be bounded by the canonical norm with a factor. That is, $\|\nu(a)\|_\infty \leq \|\sigma(a)\|_\infty \cdot C_m$ and $\|\mu(a)\|_\infty \leq \|\sigma(a)\|_\infty \cdot D_m$. By experimental results, D_m is much smaller than E_m when m is a product of multiple prime powers and D_m has a nice bound as Lemma 2.2.

Lemma 2.2. *For any $a \in \mathcal{R}$, we have $\|\mu(a)\|_\infty \leq \|\sigma(a)\|_\infty \cdot D_m$, where $D_m = \prod_{prime\ p|m} A(p)$ and $A(x) = 2/(x \cdot \tan(\pi/2x))$.*

The parameter D_m is also called the ring constant. There is a nice bound for primes and for any prime $p \geq 11$, $A(p) \approx 1.273$. Then if m does not have too many distinct prime factors, D_m is relatively small (i.e. 5 factors make D_m less than 3.343). See [20] for more details about the ring constant.

2.2 Gaussian Distribution

For any $s > 0$, define n dimension Gaussian function with parameter s as $\rho_s(\boldsymbol{x}) = \exp(-\pi\|\boldsymbol{x}\|_2^2/s^2)$. For a discrete Gaussian distribution over $\mathbb{Z}^n$, it is defined as $D_{\mathbb{Z}^n,s} = \rho_s(\boldsymbol{x})/\rho_s(\mathbb{Z}^n)$, where $\rho_s(\mathbb{Z}^n) = \sum_{\boldsymbol{x}\in\mathbb{Z}^n} \rho_s(\boldsymbol{x})$. The following lemma gives an upper bound.

Lemma 2.3 ([4], Lemma 2.4). *For any real $s > 0$ and $t > 0$, and integer vector $\boldsymbol{x} \in \mathbb{R}^n$, we have* $\Pr[|\langle \boldsymbol{x}, D_{\mathbb{Z}^n,s}\rangle| \geq ts\|\boldsymbol{x}\|_2] \leq 2\exp(-\pi t^2/s^2)$.

The Error Distribution. The error distribution χ follows the discrete Gaussian distribution with parameter s. For better efficiency, we sample the coefficients from discrete Gaussian distribution under the canonical norm.

2.3 RLWE Problems and RLWE Vector Encryption Scheme

Definition 2.4 (Ring-Learning with Error [25]). *Let $n \geq 1$, $q \geq 2$ be integers, and χ be a probability distribution over $\mathcal{R}_q$. For $s \in \mathcal{R}_q$, let $A_{s,\chi}$ be the probability distribution over $\mathcal{R}_q \times \mathcal{R}_q$ obtained by choosing a vector $a \in \mathcal{R}_q$ uniformly at random, choosing $e \leftarrow \chi$, and outputting $(a, a \cdot s + e)$.*

The decision $\mathsf{RLWE}_{q,n,\chi}$ problem is: for uniformly random $s \in \mathcal{R}_q$, given a polynomial number (in n) of samples that are either (all) from $A_{s,\chi}$ or (all) uniformly random in $\mathcal{R}_q \times \mathcal{R}_q$, determine which the case is.

Next, we recall a vector encryption scheme based on RLWE from [21] satisfying IND-CPA security and the strictly linear-only property.

Construction 2.5 (Ring-LWE Vector Encryption Scheme [21]). *The ring $\mathcal{R}$ is defined as $\mathbb{Z}[x]/(\Phi_m(x))$ and $n = \phi(m)$. Let p be the plaintext modulus and q be the ciphertext modulus. Let χ denote the error distribution with parameters σ and C. The encryption scheme can be constructed as follows:*

- $\mathsf{KeyGen}(1^\lambda)$*: Randomly select $a' \leftarrow \mathcal{R}$, $\boldsymbol{s}, \boldsymbol{e'} \leftarrow \chi^{t+d}$, $R \leftarrow \mathcal{R}^{t\times d}$. Compute $\boldsymbol{b'} = a' \cdot \boldsymbol{s} + p\boldsymbol{e'} \pmod q$. Output the key pair $(\mathsf{sk}, \mathsf{pk}) = ((\boldsymbol{s}, R), (a', \boldsymbol{b'}))$.*
- $\mathsf{Enc}(\mathsf{sk}, \boldsymbol{m})$*: Given the secret key $\mathsf{sk} = (\boldsymbol{s}, R)$ and the message $\boldsymbol{m} \in \mathcal{R}^d$, compute the vector $\boldsymbol{u} = (\boldsymbol{m}, R\boldsymbol{m})^\top$. Then sample $a \leftarrow \mathcal{R}$, $\boldsymbol{e} \leftarrow \chi^{t+d}$, and compute $\boldsymbol{b} = a\boldsymbol{s} + p\boldsymbol{e} + \boldsymbol{u} \pmod q$. Output the ciphertext $\boldsymbol{c} = (a, \boldsymbol{b})$.*
- $\mathsf{Eval}(\{\boldsymbol{c_i} = (a_i, \boldsymbol{b_i}), \alpha_i\}_{i\in[k]}, \mathsf{pk})$*: Given k ciphertexts $\boldsymbol{c_i} = (a_i, \boldsymbol{b_i})$ (encryptions of $\boldsymbol{m_i}$) and scalars $\alpha_i \in \mathcal{R}$, sample $r, e_a \leftarrow \chi$ and the smudging noise $\boldsymbol{e_{ns}} \leftarrow [-B, B]^{t+d}$. Compute and output the re-randomized linear evaluation as $(\sum_{i\in[k]} \alpha_i a_i + ra' + pe_a \pmod q, \sum_{i\in[k]} \alpha_i \boldsymbol{b_i} + r\boldsymbol{b'} + p\boldsymbol{e_{ns}} \pmod q)$.*

– $\mathsf{Dec}(\mathsf{sk}, \boldsymbol{c})$: *Taking the ciphertext* $\boldsymbol{c} = (a, \boldsymbol{b})$ *as an input, compute* $\boldsymbol{u'} = \boldsymbol{b} - a\boldsymbol{s} \pmod p$. *Phase* $\boldsymbol{u'}$ *as* $(\boldsymbol{m'}, \boldsymbol{r'})$. *If* $\boldsymbol{r'} = R\boldsymbol{m'}$, *output* $\boldsymbol{m'}$; *otherwise, terminate and output* $\perp$.

Lemma 2.6 (Correctness). *Let the parameters* $\boldsymbol{c_i}, \alpha_i, \boldsymbol{m_i}, \mathsf{pk}, q$ *be defined as Construction 2.5. If* $q > 2p(B + pn\sqrt{nk}C\sigma + n^2pk/2 + 2n^2C^2\sigma^2) + p$, *the decryption of* $\mathsf{Eval}(\{\boldsymbol{c_i} = (a_i, \boldsymbol{b_i}), \alpha_i\}_{i\in[k]}, \mathsf{pk})$ *is* $\sum_{i=1}^{k} \alpha_i \boldsymbol{m_i}$ *with probability at least* $1 - n(t + d)\exp(-\pi C^2)$.

Later, we use modulus-switching on RLWE to reduce the magnitude of the modulus. We recall this technique as below.

Definition 2.7 (Modulus-switching [8]). *For any integers* k, $q > q' > p$, *and any vector* $\boldsymbol{x} \in \mathcal{R}^k$, $\boldsymbol{x'} \leftarrow \mathsf{ModSwit}(\boldsymbol{x}, q, q', p)$ *is defined as the closest* $\mathcal{R}^k$*-vector to* $\frac{q'}{q}\boldsymbol{x}$ *satisfying* $\boldsymbol{x'} = \boldsymbol{x} \bmod p$.

Noise Smudging. This technique is usually used to obfuscate a homomorphic evaluated ciphertext or a fresh ciphertext.

Lemma 2.8 (Noise Smudging [13]). *Let* κ *be the statistical security parameter, and* B_1, B_2 *be positive integers. For any integer* $m \in [-B_1, B_1]$, n *picked from* $[-B_2, B_2]$ *uniformly at random, we have that the distribution of* $m + n$ *is statistically close to the distribution of* n *as long as* $B_1/B_2 = \mathsf{negl}(\kappa)$.

2.4 Zero Knowledge Succinct Non-Interactive Argument of Knowledge Scheme

Definition 2.9 (zk-SNARK). *For a relation* $\mathcal{L}$, *a zero-knowledge succinct non-interactive adaptive argument of knowledge protocol* Π *consists of three* PPT *algorithms* $(\Pi.\mathsf{Setup}, \Pi.\mathsf{Prove}, \Pi.\mathsf{Vefify})$.

1. $(\mathsf{crs}, \mathsf{vrs}, \mathsf{td}) \leftarrow \Pi.\mathsf{Setup}(1^\lambda, u)$: *On input public parameters and a statement* $u \in \mathcal{L}$, *the setup algorithm outputs the common reference string* crs, *the verification secret information* vrs *and the trapdoor* td.
2. $\pi \leftarrow \Pi.\mathsf{Prove}(\mathsf{crs}, u, \omega)$: *The prove algorithm takes a common reference string* crs, *a statement* u, *and a witness* ω *as an input, and returns a proof* π.
3. $0/1 \leftarrow \Pi.\mathsf{Vefify}(\mathsf{crs}, \mathsf{vrs}, \pi)$: *On input a common reference string* crs, *a verification secret information* vrs *and a proof* π, *the verify algorithm outputs a bool symbol 1 or 0 to indicate that the proof is accepted or rejected respectively.*

A zk-SNARK scheme satisfies four properties, including completeness, zero-knowledge, argument of knowledge, and succinctness.

Definition 2.10 (Completeness). *Let* λ *be the security parameter. An non-interactive argument system* Π *is complete if for any statement* u, *the setup algorithm outputs* $(\mathsf{crs}, \mathsf{vrs}, \mathsf{td}) \leftarrow \Pi.\mathsf{Setup}(1^\lambda, u)$, *and the prove algorithm outputs a proof* $\pi \leftarrow \Pi.\mathsf{Prove}(\mathsf{crs}, u, \omega)$, *then we have* $\Pr[\Pi.\mathsf{Vefify}(\mathsf{crs}, \mathsf{vrs}, \pi) = 1] = 1 - \mathsf{negl}(\lambda)$.

Definition 2.11 (Zero-knowledge). *An non-interactive argument system Π is zero-knowledge if for any $(u, \omega) \in \mathcal{L}$, the setup algorithm outputs $(\mathsf{crs}, \mathsf{vrs}, \mathsf{td}) \leftarrow \Pi.\mathsf{Setup}(1^\lambda, u)$, there exists a* PPT *simulator $\Pi.\mathsf{SIM}$ such that $\{\Pi.\mathsf{Prove}(u, \omega, \mathsf{crs})\} \approx \{\Pi.\mathsf{SIM}(u, \mathsf{td})\}$, where $\approx$ can denote perfect, statistical, and computationally indistinguishable.*

Definition 2.12 (Argument of Knowledge). *Let λ be the security parameter. An non-interactive argument system Π satisfies argument of knowledge if for any statement u, the setup algorithm outputs $(\mathsf{crs}, \mathsf{vrs}, \mathsf{td}) \leftarrow \Pi.\mathsf{Setup}(1^\lambda, u)$, and a* PPT *adversary can produce a proof π^* passing the verification, then there exists a* PPT *extractor Ext to extract a witness ω such that $(u, \omega) \in \mathcal{L}$ with polynomial probability. Equivalently, we have $\Pr[(\pi^*; \omega) \leftarrow (\mathcal{A}||\mathsf{Ext})(\mathsf{crs}, u) \wedge \Pi.\mathsf{Vefify}(\mathsf{crs}, \mathsf{vrs}, \pi^*) = 1]$ holds with probability $\mathsf{poly}(\lambda)$.*

Definition 2.13 (Succinctness). *Let λ be the security parameter. A non-interactive argument system is succinct when its proof is (quasi)-linear of λ and the logarithm of circuit size included in the relation.*

3 The Batch of LPCP-Based ZK-SNARKs

In this section, we first describe the batch encoding and decoding procedure followed by presenting the full description and security analysis of the batch of the zk-SANRK scheme. In the batch case, we can pack ℓ different rank-1 constraint systems into one proof if the cyclotomic ring's cyclotomic polynomial is a product of ℓ distinct irreducible polynomials.

3.1 Batch Encoding and Decoding Procedure

The plaintext space of RLWE scheme equals $\mathcal{R}_p = \mathbb{Z}_p[X_1, \cdots, X_t]/(\Phi_{m_1}(X_1), \cdots, \Phi_{m_t}(X_t))$. $R_p = \mathbb{Z}_p[X]/(\Phi_p(X))$. Suppose that the prime power factorization of m is $m_1 \times \cdots \times m_t$, $\phi(m) = d\ell$, and d is the smallest positive integer satisfying $p^d = 1 \bmod m$, then $\Phi_m(X) \bmod p$ factors into ℓ irreducible polynomials $F_i(X)$ in $\mathbb{Z}_p[X]$. Then the batch encoding can be written as

$$\psi : \prod_{i=1}^{\ell} \mathbb{F}_{p^d} \xrightarrow{\psi_1} \prod_{i=1}^{\ell} \mathbb{Z}_p[X]/(F_i(X)) \xrightarrow{\psi_2} R_p \xrightarrow{\psi_3} \mathcal{R}_p.$$

$\mathbb{F}_{p^d}$ is instantiated as $\mathbb{Z}_p[X]/(f(x))$ for a fixed $f(x)$ with degree d, then the first isomorphism ψ_1 maps i-th component to (different) $\mathbb{Z}_p[X]/(F_i(x))$ which is a field isomorphism in each component. The second isomorphism ψ_2 is a CRT mapping and the third isomorphism is a ring isomorphism between the $\mathbb{Z}$-basis. Then the batch encoding procedure is $\psi = \psi_1 \circ \psi_2 \circ \psi_3$, and the decoding procedure corresponds $\psi^{-1} = \psi_3^{-1} \circ \psi_2^{-1} \circ \psi_1^{-1}$.

3.2 The Batch of SNARK Scheme Description

Given ℓ rank-1 constraint systems $\{(\mathbf{A}^{(k)}, \mathbf{B}^{(k)}, \mathbf{C}^{(k)}, n_c, n_w, \mathbb{F}_{p^d})\}_{k\in[\ell]}$ (n_c is the number of constraints and n_w is the number of witness variables), the batch of LPCP-based SNARK proof system [21] proceeds as follows:

Setup(1^λ):
For each $k \in [\ell]$, do steps 1–3.

1. Select distinct $r_1^{(k)}, \cdots, r_{n_c}^{(k)}$ in $\mathbb{F}_{p^d}$, arbitrarily and let $S^{(k)} = \{r_1^{(k)}, \cdots, r_{n_c}^{(k)}\}$.
2. Set $A_0^{(k)}(X), \ldots, A_{n_w}^{(k)}(X), B_0^{(k)}(X), \ldots, B_{n_w}^{(k)}(X), C_0^{(k)}(X) \ldots, C_{n_w}^{(k)}(X)$ as the degree $(m-1)$ polynomials that satisfy the constraints:
   (1) $A_i^{(k)}(r_j^{(k)}) = a'^{(k)}_{ji}$ for $i \in [n_w], j \in [n_c]$;
   (2) $B_i^{(k)}(r_j^{(k)}) = b'^{(k)}_{ji}$ for $i \in [n_w], j \in [n_c]$;
   (3) $C_i^{(k)}(r_j^{(k)}) = c'^{(k)}_{ji}$ for $i \in [n_w], j \in [n_c]$.
   Then, set $T^{(k)}(X) = \prod_{i=1}^{n_c}(X - r_i^{(k)})$.
3. Randomly select $r^{(k)} \leftarrow \mathbb{F}_{p^d} \setminus S^{(k)}$. Then the query matrix for each R1CS constraint system is $Q^{(k)} =$

$$\begin{pmatrix} T^{(k)}(r^{(k)}) & 0 & 0 & A^{(k)}_{n_s+1}(r^{(k)}) \cdots A^{(k)}_{n_w}(r^{(k)}) & 0 & 0 & \cdots & 0 \\ 0 & T^{(k)}(r^{(k)}) & 0 & B^{(k)}_{n_s+1}(r^{(k)}) \cdots B^{(k)}_{n_w}(r^{(k)}) & 0 & 0 & \cdots & 0 \\ 0 & 0 & T^{(k)}(r^{(k)}) & C^{(k)}_{n_s+1}(r^{(k)}) \cdots C^{(k)}_{n_w}(r^{(k)}) & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 \cdots 0 & 1 & r^{(k)} & \cdots & r^{(k)^{n_c}} \end{pmatrix}$$

$\in \mathbb{F}_{p^d}^{4\times(n_c+n_w+4-n_s)}$.

4. Using the batch encoding isomorphism ψ, we can pack ℓ query matrix Q over $\mathbb{F}_{p^d}$ into one ring matrix. Then the query matrix becomes

$$Q = \begin{pmatrix} T(r) & 0 & 0 & A_{n_s+1}(r) \cdots A_{n_w}(r) & 0 & 0 & \cdots & 0 \\ 0 & T(r) & 0 & B_{n_s+1}(r) \cdots B_{n_w}(r) & 0 & 0 & \cdots & 0 \\ 0 & 0 & T(r) & C_{n_s+1}(r) \cdots C_{n_w}(r) & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 \cdots 0 & 1 & r & \cdots & r^{n_c} \end{pmatrix} \in \mathcal{R}_p^{4\times(n_c+n_w+4-n_s)}.$$

   Besides, $A_0(r), \cdots, A_{n_s}(r), B_0(r), \cdots, B_{n_s}(r), C_0(r), \cdots, C_{n_s}(r)$ are defined similarly.
5. Run the RLWE setup and obtain the key pair $(\mathsf{pk}, \mathsf{sk})$. Encrypt the query matrix Q by column using RLWE scheme (Construction 2.5), and obtain the ciphertexts as $\{\boldsymbol{c_i} = (a_i, \boldsymbol{b_i})\}_{i\in[n_c+n_w+4-n_s]}$. And let $\mathsf{st} = (A_0(r), \cdots, A_{n_s}(r), B_0(r), \cdots, B_{n_s}(r), C_0(r), \cdots, C_{n_s}(r), T(r))$.
6. Output the $\mathsf{crs} = (\mathsf{pk}, \{\boldsymbol{c_i}\}_{i\in[n_c+n_w+4-n_s]})$ and $\mathsf{vrs} = (\mathsf{sk}, \mathsf{st})$.

Prover($\mathsf{crs}, \boldsymbol{x}, \boldsymbol{w}$):

1. Taking the witness and statement on inputs, compute the LPCP proof as follows. Randomly sample $\delta_1, \delta_2, \delta_3 \leftarrow \mathcal{R}_p$, then compute $A(X), B(X), C(X)$ as: $A(X) = \delta_1 T(X) + A_0(X) + \sum_{i\in[n_w]} w_i A_i(X)$,

$B(X) = \delta_2 T(X) + B_0(X) + \sum_{i\in[n_w]} w_i B_i(X)$, $C(X) = \delta_3 T(X) + C_0(X) + \sum_{i\in[n_w]} w_i C_i(X)$. Let $H(X) = (A(X)\cdot B(X) - C(X))/T(X)$ and $\boldsymbol{h}$ is the coefficient vector of $H(X)$. Then the LPCP proof is $\pi = (\delta_1, \delta_2, \delta_3, \boldsymbol{w}, \boldsymbol{h})$.

2. Compute the proof as $\tilde{\Pi} = (\tilde{a}, \tilde{\boldsymbol{b}}) = \mathsf{Eval}(\{\boldsymbol{c_i}, \pi_i\}_{i\in[n_c+n_w+4-n_s]}, \mathsf{pk})$.
3. Run the modulus-switching procedure and output $\Pi = \mathsf{ModSwit}(\tilde{\boldsymbol{c}}, q, q', p)$.

Verifier($\mathsf{vrs}, \boldsymbol{x}, \Pi$):
Using the secret key sk, compute the decryption of Π. If the decryption outputs $\perp$, terminate and output $\perp$. Otherwise, let $\boldsymbol{m} = \boldsymbol{b} - a\boldsymbol{s} \pmod p$. For $j \in [n_c + n_w + 4 - n_s]$, let $m'_{1j} = A_0(r) + \sum_{i\in[n_s]} A_i(r)x_i + m_{1j}$, $m'_{2j} = B_0(r) + \sum_{i\in[n_s]} B_i(r)x_i + m_{2j}$, $m'_{3j} = C_0(r) + \sum_{i\in[n_s]} C_i(r)x_i + m_{3j}$. If $m'_{1j}\cdot m'_{2j} - m'_{3j} - m_{4j}\cdot T(r) = 0$ holds for each $j \in [n_c + n_w + 4 - n_s]$, output "1"; otherwise, output "0".

3.3 Security Analysis

The noise bound in the [21] does not work for non-power-of-two cyclotomic rings, Therefore, we generalize the noise analysis for a general cyclotomic ring, including non-power-of-two cyclotomic rings, which achieves a tighter bound for non-power-of-two cyclotomic rings.

For general cyclotomic ring $\mathcal{R}_p = \mathbb{Z}_p[X_1, \cdots, X_t]/(\Phi_{m_1}(X_1), \cdots, \Phi_{m_t}(X_t))$ and $m = m_1 \cdots m_t$, the powerful basis outperforms the power basis since it has a smaller ring constant (D_m). The detailed noise analysis is as below.

Theorem 3.1 (Correctness). *Let $m = m_1 \cdots m_t$, and the ring $\mathcal{R}_p$ is defined as $\mathbb{Z}_p[X_1, \cdots, X_t]/(\Phi_{m_1}(X_1), \cdots, \Phi_{m_t}(X_t))$ with degree $n = \phi(m)$. Let $N = n_c + n_w + 4 - n_s$ and $\ell' = 4M + \tau$. Let C be the corresponding error distribution bound related to the computational parameter λ. Let $\alpha < 1/2p$ satisfy $\alpha q < q/2 - (qp/2q')(1 + Cs\sqrt{n}D_m)$. For any modulus $q > q' > p$ satisfying $q = 1 \bmod p$, $q' = 1 \bmod p$, and $q > [npN/2 + D_m Cs(p\sqrt{nN} + 2Cs) + B]/\alpha$, $q' > (1/2 - \alpha)^{-1}p/2(1 + Cs\sqrt{n}D_m)$, the Construction 3.2 satisfies completeness with probability at least $1 - n\ell' \exp(-\pi C^2)$.*

Proof. From the construction of batch SNARKs, the noise growth is primarily influenced by two operations, namely the linear evaluation and modulus-switching. We give two lemmas respectively.

Lemma 3.2 (Correctness of Linear Evaluation). *Let $\mathcal{R}$, C, N, ℓ' be defined as Theorem 3.1. Let $\{\boldsymbol{c_i} \in \mathcal{R}^{1+\ell'}\}_{i\in[N]}$ be the set of N RLWE ciphertexts over $\mathcal{R}$ by Construction 2.5. For any vector $\boldsymbol{\pi} \in \mathcal{R}_p^N$, let $\boldsymbol{v} = (\|\sigma(\pi_1)\|_\infty, \cdots, \|\sigma(\pi_N)\|_\infty)$. If $q > np^2N + 2pD_mCs(p\sqrt{n}N + Cs) + 2pB$, the decryption of $\mathsf{Eval}(\{\boldsymbol{c_i}, \pi_i\}_{i\in[N]}, \mathsf{pk})$ is correct with probability at least $1 - n\ell' \exp(-\pi C^2)$.*

Proof. The noise of the linear evaluation is of form $\tilde{\boldsymbol{e}} = \tilde{\boldsymbol{u}} + \sum_{i\in[N]} \pi_i \boldsymbol{e_i} + \boldsymbol{e'}r + \boldsymbol{e_{ns}} - \boldsymbol{s}e_a$, where $\tilde{\boldsymbol{u}} = 1/p(\sum_{i\in[N]} \pi_i \boldsymbol{u}_i - \sum_{i\in[N]} \pi_i \boldsymbol{u}_i \pmod p)$. Next, we bound the powerful basis norm for each component.

For $\tilde{\boldsymbol{u}}$, we have $\|\mu(\tilde{\boldsymbol{u}})\|_\infty \leq 1/p(\sum_{i\in[N]} \mu(\pi_i \boldsymbol{u}_i))$. Then $\|\mu(\tilde{\boldsymbol{u}})\|_\infty \leq npN/2$ since $\|\mu(\tilde{u_i})\|_\infty \leq p/2$. The second term is a linear combination $\sum_{i\in[N]} \pi_i \boldsymbol{e_i} \in \mathcal{R}^{\ell'}$, then each coefficient of $\sigma(\sum_{i\in[N]} \pi_i \boldsymbol{e_i})$ is discrete Gaussian with parameter $\|\boldsymbol{v}\|_2 s$, where $\boldsymbol{v} = (\|\sigma(\pi_1)\|_\infty, \cdots, \|\sigma(\pi_N)\|_\infty)$. By Lemma 2.3, it is bounded no more than $\|\boldsymbol{v}\|_2 Cs$ with probability at least $1 - 2\exp(-\pi C^2)$. Then $\|\sigma(\sum_{i\in[N]} \pi_i \boldsymbol{e_i})\|_\infty \leq \|\boldsymbol{v}\|_2 Cs$ with probability at least $1 - n\ell' \exp(-\pi C^2)$ since the canonical embeddings come in pair of complex conjugates (e.g., $\sigma_i = \overline{\sigma_{m-i}}$). By Lemma 2.2, we have $\|\mu(\sum_{i\in[N]} \pi_i \boldsymbol{e_i})\|_\infty \leq p\sqrt{nN} Cs D_m$ with probability at least $1 - n\ell' \exp(-\pi C^2)$. Since each entry of $\boldsymbol{e'}, \boldsymbol{s} \in \mathcal{R}^{\ell'}, r, e_a \in \mathcal{R}$ is sampled from the error distribution χ with parameter s, the canonical norm of them are bounded by Cs with probability at least $1 - n\exp(-\pi C^2)$. Then $\|\sigma(\boldsymbol{e'}r)\|_\infty \leq C^2 s^2$ with probability at least $1 - n\ell' \exp(-\pi C^2)$. Furthermore, we have $\|\mu(\boldsymbol{e'}r)\|_\infty \leq C^2 s^2 D_m$ with probability at least $1 - n\ell' \exp(-\pi C^2)$. So does $\|\mu(\boldsymbol{s}e_a)\|_\infty$. Each coefficient of $\boldsymbol{e_{ns}}$ is sampled from $[-B, B]$ uniformly under the powerful basis, then $\|\mu(\boldsymbol{e_{ns}})\|_\infty \leq B$. To sum up, the powerful basis norm $\|\mu(\tilde{\boldsymbol{e}})\|_\infty$ is less than

$$npN/2 + D_m(p\sqrt{nN} Cs + 2\,C^2\,s^2) + B$$

with probability at least $1 - n\ell' \exp(-\pi C^2)$. □

Like the noise analysis of the linear evaluation, we need to bound the noise produced by the modulus-switching procedure under the powerful basis.

Lemma 3.3 (Correctness of Modulus-switching). *Let q, q', p be positive integers such that $q > q' > p$ and $q = q' = 1 \bmod p$. Let $\mathcal{R}$, C be defined as Theorem 3.1. Let $\boldsymbol{c} = (a, \boldsymbol{b}) \in \mathcal{R}^{\ell'+1}$ and $\boldsymbol{c'} \leftarrow \mathsf{ModSwit}(\boldsymbol{c}, q, q', p)$. If $\boldsymbol{s}$ with each coefficient sampled from χ satisfies $\|\mu([\boldsymbol{b} - a\boldsymbol{s}]_q)\|_\infty < q/2 - (qp/2q')(1 + Cs\sqrt{n} D_m)$, we have $[[\boldsymbol{b} - a\boldsymbol{s}]_q]_p = [[\boldsymbol{b'} - a'\boldsymbol{s}]_{q'}]_p$, and $\|\mu([\boldsymbol{b'} - a'\boldsymbol{s}]_{q'})\|_\infty < q'/q \|\mu([\boldsymbol{b} - a\boldsymbol{s}]_q)\|_\infty + (p/2)(1 + Cs\sqrt{n} D_m)$ with probability at least $1 - n\ell' \exp(-\pi C/s^2)$.*

Proof. Let $[\boldsymbol{b} - a\boldsymbol{s}]_q = \boldsymbol{b} - a\boldsymbol{s} - \boldsymbol{k}q$ for some $\boldsymbol{k} \in \mathcal{R}^{\ell'}$. Let $\boldsymbol{e_{q'}} = \boldsymbol{b'} - a'\boldsymbol{s} - \boldsymbol{k}q'$. Then we have

$$\begin{aligned}
&\|\mu(\boldsymbol{e_{q'}})\|_\infty = \|\mu(-\boldsymbol{k}q + (q'/q)(\boldsymbol{b} - a\boldsymbol{s}) + (\boldsymbol{b'} - q'/q\boldsymbol{b}) - (a' - q'/qa)\boldsymbol{s})\|_\infty \\
&\leq q'/q \|\mu([\boldsymbol{b} - a\boldsymbol{s}]_q)\|_\infty + \|\mu(\boldsymbol{b'} - q'/q\boldsymbol{b})\|_\infty + \|\sigma(a' - q'/qa)\|_\infty \cdot \|\sigma(\boldsymbol{s})\|_\infty D_m \\
&\leq q'/q \|\mu([\boldsymbol{b} - a\boldsymbol{s}]_q)\|_\infty + (p/2)(1 + \sqrt{n} Cs D_m) < q'/2
\end{aligned}$$

with probability at least $1 - n\ell' \exp(-\pi C/s^2)$. Moreover, we have $[[\boldsymbol{b} - a\boldsymbol{s}]_{q'}]_p = [\boldsymbol{e_{q'}}]_p = [\boldsymbol{b'} - a'\boldsymbol{s} - \boldsymbol{k}q']_p = [\boldsymbol{b} - a\boldsymbol{s} - \boldsymbol{k}q]_p = [[\boldsymbol{b} - a\boldsymbol{s}]_q]_p$.

Now suppose the powerful basis infinity norm of the noise after the linear evaluation is $\|\mu(\tilde{\boldsymbol{e}})\|_\infty < \alpha q$, where $\alpha < 1/2$ and αq satisfying the modulus-switching requirement. After modulus-switching, the noise becomes $\|\mu(\boldsymbol{e})\|_\infty < \alpha q' + (p/2)(1 + Cs\sqrt{n} D_m)$. To be decrypted correctly, we have $q' > (1/2p - \alpha)(1 + Cs\sqrt{n} D_m)p/2$.

Combining Lemma 3.2 and Lemma 3.3, the theorem follows. □

Knowledge Soundness and Honest-verifier Zero-knowledge. Ishai et al. [21] proved that the construction for a single R1CS satisfies the computational knowledge soundness. That is, if a proof $\Pi \in \mathcal{R}_{q'}^{\ell'+1}$ produced by the adversary passes the verification, there exists a PPT extractor that can extract a witness $\boldsymbol{w} \in \mathcal{R}_p^N$. Using the decoding isomorphism ψ^{-1}, we can obtain $\boldsymbol{w}^{(k)} \in \mathbb{F}_{p^d}^N$ for the k-th R1CS $(\mathbf{A}^{(k)}, \mathbf{B}^{(k)}, \mathbf{C}^{(k)}, n_c, n_w, \mathbb{F}_{p^d})$. The honest-verifier zero-knowledge property follows in a similar manner compared to that of [21]. For simplicity, we give the following theorems and omit the proofs.

Theorem 3.4. *Assume that the hardness of* RLWE *assumptions holds and the vector encryption satisfies CPA-secure and strictly linear-only property. Then for any q, q' are defined as Theorem 3.1, the Construction 3.2 satisfies computational honest-verifier zero-knowledge.*

Theorem 3.5. *Assume that the hardness of* RLWE *assumptions holds and the vector encryption satisfies CPA-secure and strictly linear-only property. Then for any q, q' are defined as Theorem 3.1, the Construction 3.2 satisfies computational knowledge soundness with soundness error $2n_c/(p^d - n_c)$.*

Corollary 3.6. *If the vector encryption (Construction 2.4) is CPA-secure and satisfies the strictly linear-only property, then the Construction 3.2 is a designated-verifier zero-knowledge succinct argument of knowledge for any R1CS relations in the preprocessing model.*

Proof. Combining Theorem 3.1, 3.5 and 3.6, the corollary follows. □

4 Experimental Performance

In this section, we conduct a series of experiments to evaluate the performance of our batch scheme. Firstly, we summarize some parameter choices. Secondly, we propose several parameter sets in Table 2, and then compare the size and the running time for different parameter sets in Table 3 and 4.

4.1 Parameter Selection

- The ring constant D_m is computed by Lemma 2.2.
- For any circuit with n_c constraints, we need arbitrary and distinct n_c points for interpolating a polynomial of degree $n_c - 1$. Unfortunately, there is not always a n_c-th root of unity in $\mathbb{F}_{2^n}$. We can relax the requirement to find a n_c'-th root of unity in $\mathbb{F}_{2^n}$ where n_c' is the closest and larger integer dividing $2^n - 1$ to n_c. Also, this results in a polynomial of degree n_c'.
- The noise distribution is a discrete Gaussian with parameter s, in all parameter sets, we set $s = 64$. Moreover, C is a positive value to make $\exp(-\pi C^2) < 2^{-\lambda}$ hold. We set $C = 6$ here.

- The knowledge soundness of a single execution in each slot is $2n_c/(p^d - n_c)$. If it is less than $2^{-\lambda}$, we need parallel repetition. The repetition time is the smallest integer M such that $(2n_c/(p^d - n_c))^M < 2^{-\lambda}$. Additionally, the sparsification parameter τ is the smallest integer such that $p^{d\tau} \geq 2^{\lambda}$. Therefore, we have that the total length of the vector encryption is $\ell' = 4M + \tau$.
- The ciphertext modulus and switched modulus follow the Theorem 3.1.
- The parameter α balances the modulus q and switched-modulus q'. With the decrease of α, q increases and q' decreases. After experimental attempts, we find that there is a tiny change of different α ranging from $1/4p$ to $1/16p$ (corresponding to the modulus increasing from 1 bit to 3 bits). Thus we choose $\alpha = 1/4p$.
- The computational and statistical security parameters are $\lambda = 128$, $\kappa = 40$ respectively. Then the noise smudging bound B is set to be 2^{40} as before.

4.2 Experimental Results

The batch SNARK scheme (Construction 3.2) can prove R1CS in $\mathbb{F}_{p^d}$ for any prime powers, but here we focus on $p = 2$ for later applications. For a circuit built on $\mathbb{F}_{2^d}$, it is usually bit-based or byte-based. Then in Table 2, we give 4 typical choices of $\mathbb{F}_{2^d}$ (all are extended fields of $\mathbb{F}_{2^8}$) for different circuit sizes, which is sufficient to cover the majority of real-world use cases. For clarity, the sets from I to IV in Table 3 represent the parameters noted in Table 2.

Then, we implement our scheme in C++ based on HElib v2.2.2, using the parameters in Table 2 and Table 3. The computing environment is a cloud server with Intel(R) Xeon(R) Gold 6230R CPU @ 2.10 GHz and 128 GB RAM, running Ubuntu 22.04 LTS. The compiler we used is GCC 11.3.0. It is important to note that all experiments were executed utilizing a single-core and single-threaded mode, without any SIMD optimizations. All the results in Table 4 are obtained by running each experiment 10 times and taking the average. It is worth noting that the prover can precompute the LPCP proof as it processes the witness, and then the prover time listed in all tables refers to the online time (apply the linear evaluation and modulus switching). The security of RLWE scheme in all sets is higher than the 128-bit security level, which is estimated by the lwe-estimator [2]. Based on Tables 3 and 4, we obtained the trends of the size and time of the batch of zk-SNARK schemes as the circuit scale varies in Fig. 1.

Table 2. The field parameter sets.

Sets	Field	m	n	D_m	#Slot(ℓ)	τ
I	$\mathbb{F}_{2^{16}}$	4369	4096	1.621	256	8
II	$\mathbb{F}_{2^{24}}$	4097	3840	1.621	160	6
III	$\mathbb{F}_{2^{40}}$	7905	3840	2.304	96	4
IV	$\mathbb{F}_{2^{48}}$	6057	4032	1.698	84	3

Remark 4.1. *In our parameter choices, the ratio C_m/D_m lies in $1 \sim 4.25$ for set I, II, IV and C_m/D_m lies in $7.82 \sim 826.09$ for set III.*

Table 3. The parameter sets of the batch of zk-SNARKs for moderate-size circuits in $\mathbb{F}_{2^n}$.

n_c	Set I							Set II						
	n'_c	M	ℓ'	$\log q$	$\log q'$	$\lvert\pi\rvert$	$\lvert\mathsf{crs}\rvert$	n'_c	M	ℓ'	$\log q$	$\log q'$	$\lvert\pi\rvert$	$\lvert\mathsf{crs}\rvert$
2^{10}	1285	26	112	68	19	4.19	33.54	1205	10	46	68	19	2.62	19.51
2^{11}	3855	33	140	69	19	5.23	108.76	2169	11	50	69	19	2.84	40.79
2^{12}	4369	43	180	70	19	6.72	203.44	4097	12	54	70	19	3.06	86.96
2^{13}	13107	65	268	71	19	9.98	773.01	9399	13	58	71	19	3.28	203.68
2^{14}	21845	129	524	72	19	19.48	2750.98	20485	15	66	72	19	3.73	493.66
n_c	Set III							Set IV						
	n'_c	M	ℓ'	$\log q$	$\log q'$	$\lvert\pi\rvert$	$\lvert\mathsf{crs}\rvert$	n'_c	M	ℓ'	$\log q$	$\log q'$	$\lvert\pi\rvert$	$\lvert\mathsf{crs}\rvert$
2^{10}	1271	5	24	68	20	2.44	17.12	1205	4	19	68	19	2.23	15.61
2^{11}	2635	5	24	69	20	2.44	35.44	2169	4	19	69	19	2.23	29.97
2^{12}	4675	5	24	70	20	2.44	67.34	4097	4	19	70	19	2.23	59.07
2^{13}	8525	5	24	71	20	2.44	130.17	8245	4	19	71	19	2.23	120.2
2^{14}	17391	6	28	72	20	2.83	313.08	20485	4	19	72	19	2.23	273.41

∗ The proof length is amortized and is measured in kilobytes (KB). The CRS length is compressed and amortized, measured in megabytes (MB).

Upon comparing the four sets, the performance of set I is notably worse than other sets owing to its larger repetition time. Since the number of constraints bears little disparity with the field size, the soundness error is large, necessitating more repetitions to attain a 128-bit security level. Conversely, the remaining sets demonstrate minimal alterations in the amortized proof size, approximately 3KB. This is 5x smaller than the single proof size even in the prime field (approximately 15KB). The CRS size is proportionate to the cumulative sum of both the number of constraints and witness variables (N). Given that the fundamental operations in the high-degree extension ring are slower than those in the prime field, the overall running time is significantly influenced.

5 Applications: Prove the Knowledge of Symmetric Primitives

In this section, we implement zk-SNARKs for three typical symmetric primitives, including AES, MiMC and LowMC.

5.1 A Rank-1 Constraint System for AES

In this subsection, we integrate and optimize a rank-1 constraint system for the AES (Rijndael) algorithm, which counts the accurate number of constraints and

Table 4. The running time of the batch of zk-SNARKs for moderate-size circuits in $\mathbb{F}_{2^n}$.

n_c	Set I				Set II			
	n'_c	Setup(s)	Prover(s)	Verifier(ms)	n'_c	Setup(s)	Prover(s)	Verifier(ms)
2^{10}	1285	3.76	0.46	4.89	1205	2.38	0.24	2.44
2^{11}	3855	13.37	1.44	6.01	2169	4.72	0.48	2.66
2^{12}	4369	24.38	2.7	7.86	4097	11.58	1.16	2.86
2^{13}	13107	101.78	10.06	11.65	9399	33.87	2.69	3.09
2^{14}	21845	357.73	35.59	22.57	20485	103.2	6.42	3.53
n_c	Set I				Set II			
	n'_c	Setup(s)	Prover(s)	Verifier(ms)	n'_c	Setup(s)	Prover(s)	Verifier(ms)
2^{10}	1271	3.43	0.35	2	1205	2.18	0.26	1.42
2^{11}	2635	6.61	0.71	2.01	2169	4.38	0.49	1.42
2^{12}	4675	14.1	1.35	2.01	4097	9.54	0.49	1.42
2^{13}	8525	30.77	2.57	2.02	8245	27.59	1.95	1.43
2^{14}	17391	94	6.07	2.35	20485	87.62	4.3	1.46

∗ Each time denotes the amortized time. The prover time refers to the online time.

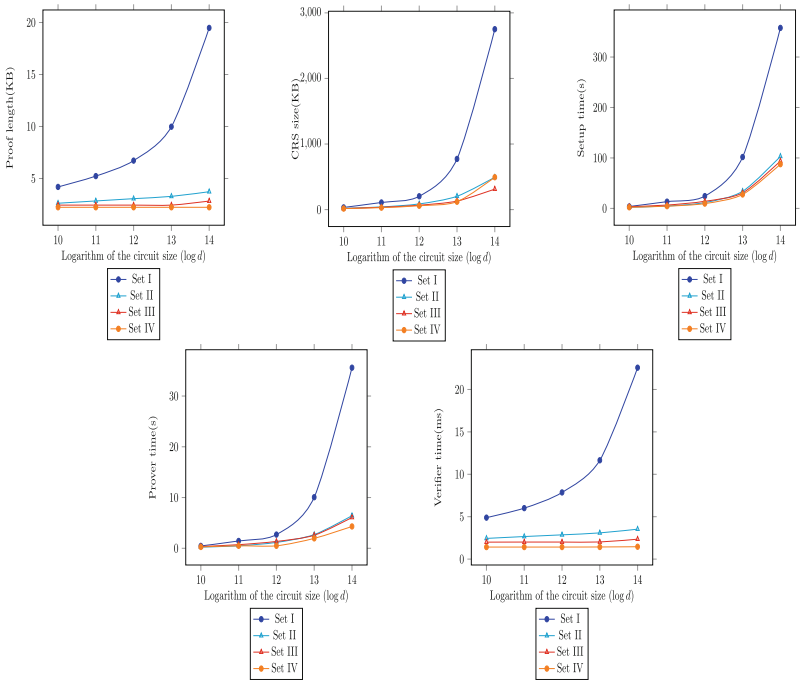

Fig. 1. The size and time tendency varying from the circuit size.

witnesses. We choose the AES algorithm for example as it is one of the most famous block cipher designs and is widely used in practice.

Constraint generation for AES. The construction of the AES algorithm primarily encompasses the initial round (pre-whitening), subsequent $N_r - 1$ rounds, the final round and the key expansion (key schedule). The round function can be divided into four fundamental operations, each of which is considered individually as delineated below.

SubBytes: The algebra expression of S-box, as derived from [14], can be partitioned into two distinct components: the inversion operation and the affine function.

- **Inversion**: Computing the inversion of a equals to compute a^{254}. This procedure takes 7 steps, resulting in 11 constraints and 11 multiplications. The computation sequence is as follows. $a^2 = a \cdot a$, $a^3 = a \cdot a^2$, $a^6 = a^3 \cdot a^3$, $a^{12} = a^6 \cdot a^6$, $a^{14} = a^{12} \cdot a^2$, $a^{15} = a^{12} \cdot a^3$, $a^{30} = a^{15} \cdot a^{15}$, $a^{60} = a^{30} \cdot a^{30}$, $a^{120} = a^{60} \cdot a^{60}$, $a^{240} = a^{120} \cdot a^{120}$, $a^{254} = a^{240} \cdot a^{14}$.
- **Affine function**: Taking a as an input, the output of the affine function is $\sum_{i=0}^{7} c_{i+1} a^{2^i} + c_0$, where c_i is defined as $c_0 = g^6 + g^5 + g + 1$, $c_1 = g^2 + 1$, $c_2 = g^3 + 1$, $c_3 = g^7 + g^6 + g^5 + g^4 + g^3 + 1$, $c_4 = g^5 + g^2 + 1$, $c_5 = g^7 + g^6 + g^5 + g^4 + g^2$, $c_6 = 1$, $c_7 = g^7 + g^5 + g^4 + g^2 + 1$, and $c_8 = g^7 + g^3 + g^2 + g + 1$, and g is a root of $x^8 + x^4 + x^3 + x + 1 = 0$. Each a^{2^i} consumes one multiplication and one constraint for $i \in \{1, \cdots, 7\}$. This process needs 15 multiplications and adds 7 constraints and 8 new variables in total.

ShiftRows, MixColumns, AddRoundKey: For ShiftRows, suppose that the input row vector is (a_0, a_1, a_2, a_3), then the output row vector is (b_0, b_1, b_2, b_3). The constraints imposed by byte shifting are of form as follows (Taking cyclic left shift one byte as an example): $a_1 \cdot 1 = b_0$, $a_2 \cdot 1 = b_1$, $a_3 \cdot 1 = b_2$, $a_0 \cdot 1 = b_3$. MixColumns is a linear combination and the AddRoundKey operation is an addition in $\mathbb{F}_{2^n}$. Then these can be optimized into the affine function.

Apart from the round function, the pre-whitening phase involves a straightforward AddRoundKey operation, requiring 16 additions. As for the key expansion, it encompasses SubWord, ShiftWord, and XOR operations, which have been previously analyzed.

Witness generation for AES. The witness vector includes the input, the key seed, and intermediate variables generated in the process of AES algorithm and the output.

Remark 5.1. *Compared to xJsnark [22], one major difference is that we adopt the algebraic expression for implementing S-box. In our construction, we take 19 constraints per S-box while xJsnark took 40 constraints. Moreover, we get rid of the conversion between bits and bytes. This conversion costs a lot (256 constraints for one bi-directional transformation over a 254-bit prime field). Instead, we construct all constraints over* $\mathbb{F}_{2^8}$ *naturally.*

Table 5. The comparison of AES, MiMC and LowMC

	AES (Ours)	AES (xJsnark [22]*)	MiMC [1]	LowMC [3]
Constraint and witness generation time	30.5ms	–	7.8 ms	90.3 ms
Base field	$\mathbb{F}_{2^8}$	$\mathbb{F}_p$	$\mathbb{F}_{21025}$	$\mathbb{F}_2$
# addition	2560	–	646	8420888
# multiplication	7000	–	1292	9408
# rank-1 constraint	3600	14240	646	4704
# witness	3616	–	1292	9408

∗ xJsnark performed the program-to-circuit transformation, and only achieved the implementation of S-box. The prime p here is a 254-bit prime.

Based on the above analysis, we calculate the total number of constraints and witness variables for AES-128, along with the corresponding generation time. The comparison with other symmetric schemes is listed in Table 5.

5.2 Implementation of SNARK

For all byte-based circuits, it is feasible to construct a rank-1 constraint system over $\mathbb{F}_{2^8}$ in a straightforward manner. However, we notice that the number of constraints in the above-established rank-1 constraint system may often exceed the field size (e.g., 256) at a high probability. To tackle this issue, we introduce the field extension and do polynomial interpolations in the extended field.

Field Extension

Construction 5.2. (Field Extension from $\mathbb{F}_{2^d}$ to $\mathbb{F}_{\mathbf{2}}^{\mathbf{d}'}$). *Let g be a generator of $\mathbb{F}_{2^d} \cong \mathbb{Z}_2[X]/(f(X))$, where $f(X)$ has degree d. Every element belonging to $\mathbb{F}_{2^d}$ can be represented uniquely as $a = \sum_{i=0}^{d-1} a_i g^i$. Additionally, let $\alpha \in \mathbb{F}_{2^{d'}}$ be a root of $f(x) = 0$. Then a in the extension field $\mathbb{F}_{2^{d'}}$ can be constructed as $\pi(a) = \sum_{i=0}^{d-1} a_i \alpha^i$.*

The Gap Between the R1CS Over $\mathbb{F}_{\mathbf{2}}^{\mathbf{d}}$ and $\mathbb{F}_{\mathbf{2}}^{\mathbf{d'}}$. Given a R1CS instance $\mathbb{I} = (\mathbf{A}, \mathbf{B}, \mathbf{C}, n_c, n_w, \mathbb{F}_{2^d})$, we can obtain a another R1CS instance $\mathbb{I}' = (\mathbf{A}', \mathbf{B}', \mathbf{C}', n_c, n_w, \mathbb{F}_{2^{d'}})$ by Construction 5.2. Clearly, it is not strictly equivalent to prove the two instances. More preciously, if we know a vector $\boldsymbol{w}$ such that $(\mathbf{A} \cdot \boldsymbol{w}) \circ (\mathbf{B} \cdot \boldsymbol{w}) = \mathbf{C} \cdot \boldsymbol{w}$, we can compute a vector $\boldsymbol{w'}$ such that $(\mathbf{A}' \cdot \boldsymbol{w'}) \circ (\mathbf{B}' \cdot \boldsymbol{w'}) = \mathbf{C}' \cdot \boldsymbol{w'}$. Applying the map π (Construction 5.2) to each entry of $\boldsymbol{w}$ results in $\boldsymbol{w'}$ since the map π is an isomorphism that preserves

addition and multiplication operations. Yet, this is not the unique solution in the extended field. If a malicious prover obtains another solution, they will only prove the knowledge of $\mathbb{I}'$, not $\mathbb{I}$. In theory, the solution exists but we can prove that any probability polynomial time (PPT) malicious prover finds such a solution at a negligible probability. The core idea can be grasped as follows. If a PPT malicious prover does not possess the witness $\boldsymbol{w}$, he obtains $\boldsymbol{w'}$ by solving $(\mathbf{A}' \cdot \boldsymbol{w'}) \circ (\mathbf{B}' \cdot \boldsymbol{w'}) = \mathbf{C}' \cdot \boldsymbol{w'}$ directly. The analysis is based on two observations. As the number of witness variables exceeds the number of constraints, there exists some freedom variables (The freedom variables are actually keys). Secondly, there are a lot of non-linear relations. Take AES-128 as an example. Given a pair $(\boldsymbol{m}, \boldsymbol{c}) \in \mathbb{F}_{2^8}^{16} \times \mathbb{F}_{2^8}^{16}$, there exists one $\boldsymbol{k} \in \mathbb{F}_{2^8}^{16}$ satisfying the constraint system on average. If the key space is extended to $\mathbb{F}_{2^m}$, there exists $2^{16m-128}$ keys satisfying the constraint system on average. Finally, the successful probability of guessing a valid key (or witness) is $2^{16m-128}/2^{16m} = 2^{-128}$. Then the complexity of solving $(\mathbf{A}' \cdot \boldsymbol{w'}) \circ (\mathbf{B}' \cdot \boldsymbol{w'}) = \mathbf{C}' \cdot \boldsymbol{w'}$ is not lower than that of solving $(\mathbf{A} \cdot \boldsymbol{w}) \circ (\mathbf{B} \cdot \boldsymbol{w}) = \mathbf{C} \cdot \boldsymbol{w}$. As a result, the extra solutions do not harm the soundness of SNARKs for $\mathbb{I}$.

Experimental Results. We show the parameters and running times of proving the knowledge of symmetric primitives in Tables 6 and 7. The experimental environment coincides with that in Sect. 4.

Table 6. The parameters for symmetric primitives with security parameters $\lambda = 128$, $\kappa = 40$.

Scheme	(Extended) Field	#Slot(ℓ)	m	n	D_m	n_c	n'_c	ℓ'	$\log q$	$\log q'$	$\lvert\pi\rvert$	$\lvert\mathsf{crs}\rvert$
AES	$\mathbb{F}_{2^{16}}$	256	4369	4096	1.621	4120	4369	180	70	19	6.75	205.15
	$\mathbb{F}_{2^{24}}$	160	4097	3840	1.621	4120	9399	58	70	19	3.28	157.03
	$\mathbb{F}_{2^{40}}$	96	7905	3840	2.304	4120	4675	24	70	20	2.44	70.46
	$\mathbb{F}_{2^{48}}$	84	6057	4032	1.698	4120	4365	19	70	19	2.23	64.57
MiMC	$\mathbb{F}_{21025}$	6	6151	6150	1.273	646	1801	5	69	19	14.26	103.15
LowMC	$\mathbb{F}_{2^{16}}$	256	4369	4096	1.621	4704	13107	268	71	19	10.02	648.83
	$\mathbb{F}_{2^{24}}$	160	4097	3840	1.621	4704	9399	58	70	19	3.28	163.82
	$\mathbb{F}_{2^{40}}$	96	7905	3840	2.304	4704	5115	24	70	20	2.44	78.66
	$\mathbb{F}_{2^{48}}$	84	6057	4032	1.698	4704	4711	19	70	19	2.23	71.65

$*$ The proof length is amortized and is measured in kilobytes (KB). The CRS length is compressed and amortized, measured in megabytes (MB).

Table 7. The running time for symmetric primitives.

Scheme	(Extended) Field	Setup(s)	Prover(s)	Verifier(ms)
AES	$\mathbb{F}_{2^{16}}$	25.11	2.8	7.86
	$\mathbb{F}_{2^{24}}$	22.23	2.07	3.1
	$\mathbb{F}_{2^{40}}$	15.25	1.44	2.01
	$\mathbb{F}_{2^{48}}$	10.15	1.01	1.43
MiMC	$\mathbb{F}_{2^{1025}}$	8.64	1.194	5.19
LowMC	$\mathbb{F}_{2^{16}}$	78.22	8.35	11.56
	$\mathbb{F}_{2^{24}}$	23.58	2.15	3.07
	$\mathbb{F}_{2^{40}}$	15.71	1.52	1.99
	$\mathbb{F}_{2^{48}}$	11.27	1.12	1.39

$*$ Each time denotes the amortized time. The prover time refers to the online time.

6 Conclusion

We generalize the LPCP-based SNARK schemes for general cyclotomic rings and propose a tighter bound in the noise analysis for non-power-of-two cyclotomic rings using the powerful basis. Then, we propose a batch SNARK scheme for R1CS in $\mathbb{F}_{p^n}$ for any prime p and apply our batch SNARK schemes for R1CS in $\mathbb{F}_{2^n}$. The batch technique results in a nice amortized proof size. Due to the high-degree extension ring, we can only process the moderate-size circuits. Fast ring operations will enhance the applicability of handling larger-size circuits.

Acknowledgements. We thank the reviewers of SecureComm 2023 for providing insightful comments to improve the clarity of this paper and pointing out some typos. This work is supported by the National Natural Science Foundation of China (No. 12371525).

References

1. Albrecht, M., Grassi, L., Rechberger, C., Roy, A., Tiessen, T.: MiMC: efficient encryption and cryptographic hashing with minimal multiplicative complexity. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10031, pp. 191–219. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53887-6_7
2. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. J. Math. Cryptol. **9**(3), 169–203 (2015)
3. Albrecht, M.R., Rechberger, C., Schneider, T., Tiessen, T., Zohner, M.: Ciphers for MPC and FHE. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 430–454. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5_17
4. Banaszczyk, W.: Inequalities for convex bodies and polar reciprocal lattices inR^n. Discret. Comput. Geom. **13**, 217–231 (1995)
5. Ben-Sasson, E., et al.: Zerocash: decentralized anonymous payments from bitcoin. In: 2014 IEEE Symposium on Security and Privacy, pp. 459–474 (2014)

6. Bonneau, J., Meckler, I., Rao, V., Shapiro, E.: Coda: Decentralized cryptocurrency at scale. Cryptology ePrint Archive, Paper 2020/352 (2020). https://eprint.iacr.org/2020/352
7. Bouvier, C., et al.: New design techniques for efficient arithmetization-oriented hash functions: anemoi permutations and jive compression mode. Cryptology ePrint Archive (2022)
8. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. ACM Trans. Comput. Theory (TOCT) **6**(3), 1–36 (2014)
9. Danezis, G., Fournet, C., Groth, J., Kohlweiss, M.: Square span programs with applications to succinct NIZK arguments. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8873, pp. 532–550. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45611-8_28
10. Esgin, M.F., Nguyen, N.K., Seiler, G.: Practical exact proofs from lattices: new techniques to exploit fully-splitting rings. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020. LNCS, vol. 12492, pp. 259–288. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64834-3_9
11. Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic span programs and succinct NIZKs without PCPs. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 626–645. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_37
12. Gennaro, R., Minelli, M., Nitulescu, A., Orrù, M.: Lattice-based zk-snarks from square span programs. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 556–573 (2018)
13. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing, pp. 169–178 (2009)
14. Gentry, C., Halevi, S., Smart, N.P.: Homomorphic evaluation of the AES circuit. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 850–867. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_49
15. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. **18**(1), 186–208 (1989). https://doi.org/10.1137/0218012
16. Grassi, L., Hao, Y., Rechberger, C., Schofnegger, M., Walch, R., Wang, Q.: Horst meets fluid-spn: griffin for zero-knowledge applications. Cryptology ePrint Archive (2022)
17. Grassi, L., Khovratovich, D., Rechberger, C., Roy, A., Schofnegger, M.: Poseidon: a new hash function for zero-knowledge proof systems. In: USENIX Security Symposium, vol. 2021 (2021)
18. Grassi, L., Onofri, S., Pedicini, M., Sozzi, L.: Invertible quadratic non-linear layers for MPC-/FHE-/ZK-friendly schemes over fnp: application to Poseidon. IACR Transactions on Symmetric Cryptology, pp. 20–72 (2022)
19. Groth, J.: On the size of pairing-based non-interactive arguments. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 305–326. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_11
20. Halevi, S., Shoup, V.: Design and implementation of HElib: a homomorphic encryption library. Cryptology ePrint Archive, Paper 2020/1481 (2020). https://eprint.iacr.org/2020/1481
21. Ishai, Y., Su, H., Wu, D.J.: Shorter and faster post-quantum designated-verifier zkSNARKS from lattices. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (2021)

22. Kosba, A., Papamanthou, C., Shi, E.: xJsnark: a framework for efficient verifiable computation. In: 2018 IEEE Symposium on Security and Privacy (SP), pp. 944–961. IEEE (2018)
23. Liu, F.H., Wang, H.: Batch bootstrapping i: a new framework for SIMD bootstrapping in polynomial modulus. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023. EUROCRYPT 2023. LNCS, vol. 14006, pp. 321–352. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30620-4_11
24. Lüftenegger, R., Rechberger, C., Grassi, L., Schofnegger, M., Walch, R., Khovratovich, D.: Reinforced concrete: a fast hash function for verifiable computation. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS'22) (2022)
25. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM (JACM) **56**(6), 1–40 (2009)
26. Songhori, E.M., Hussain, S.U., Sadeghi, A.R., Schneider, T., Koushanfar, F.: TinyGarble: highly compressed and scalable sequential garbled circuits. In: 2015 IEEE Symposium on Security and Privacy, pp. 411–428. IEEE (2015)
27. Yang, R., Au, M.H., Zhang, Z., Xu, Q., Yu, Z., Whyte, W.: Efficient lattice-based zero-knowledge arguments with standard soundness: construction and applications. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11692, pp. 147–175. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26948-7_6