An Efficient and Succinct Range Proof with More Functionalities based on Interactive Oracle Proof

Weihan Li[1], Zongyang Zhang[1], and Yanpei Guo[1]

Beihang University, Beijing, China
{leeweihan,zongyangzhang,19373442}@buaa.edu.cn

Abstract. We present a novel zero-knowledge range proof (zkRP) showing that some committed secret value is in a specific range $[A, B-1]$ for arbitrary non-negative integers A and B. This scheme supports batch processing, allowing to prove that multiple secret values are in respective ranges even with different bases. At the heart of our scheme is a batch inner product argument based on interactive oracle proof built on the techniques of Ben-Sasson et.al. (Eurocrypt 2019). Our zkRP relies only on Reed-Solomon codes and hash functions and is plausibly post-quantum secure. Compared with two of the most efficient latticed-based zkRPs, i.e., Lyubashevsky et al. (ACM CCS 2020) and Couteau et al. (Eurocrypt 2021), the communication complexity of our zkRP is logarithmic while their schemes are both linear. For an arbitrary range $[A, B-1] \subsetneq [0, 2^{512}-1]$ and 120-bit security, our proof size is 40.7 KB, reducing about 20% compared with 51.3 KB of Lyubashevsky et al. Our proof size for 1000 instances in the batch setting is 0.52 MB for the range $[0, 2^{64}-1]$ and 128-bit security, 10 times smaller than Couteau et al.

Keywords: Zero-knowledge proof · Range proof · Interactive oracle proof · Inner product argument.

1 Introduction

Zero-knowledge proof (ZKP) [25] allows a prover to convince a verifier that some statement is true without leaking any extra knowledge beyond its validity. In recent years, there have been numerous highly-efficient designs for both generalized and specialized ZKPs. While generalized ZKPs aim for NP-complete problems such as the circuit satisfiability problem [1, 10], specialized ones focus on particular problems such as the knowledge of a secret key [23] and of a pre-image for a hash function [5]. Zero-knowledge range proof (zkRP) is a specialized ZKP that enables a prover to prove that a committed integer is in a certain range. ZkRPs contribute to numerous practical applications such as anonymous credentials [15], e-voting [26], and electronic auctions [13, 33]. The recent progress on practical zkRPs can be divided into two categories according to the underlying techniques.

ZkRPs using DLOG-based inner product arguments: This line of work [10, 16, 19, 22, 28] are based on the discrete-log (DLOG) assumption. They rely on generalized Pedersen commitments to commit to the entire bit-decomposition of some integer and usually use an *inner product argument* to prove that all committed values are bits. One advantage of these proofs is that they have logarithmic communication complexity and small concrete proof size, about several KBs in practice. The stand-out performance of this line of work contributes to practical applications, the prime example being the integration of Bulletproofs [10] in cryptocurrencies Monero[1] and Beam[2]. However, they are not secure quantum adversaries.

Lattice-based zkRPs: Recently, there have been studies on practical latticed-based zkRPs [2, 17, 35]. These zkRPs are usually constructed using ZKPs for specialized relations of committed messages such as addition relations [35] and element-wise products [2]. Compared with Bulletproofs and its variants, lattice-based zkRPs have faster computation without group operations but linear communication complexity as well as larger proof size.

In this paper, we focus on another way to build post-quantum zkRPs based on interactive oracle proof (IOP) in order to achieve a better-combined feature of the zkRPs in the above two categories. IOP and lattice are the most efficient known techniques to create quantum-safe generalized ZKPs. Another advantage of IOP-based ZKPs is that they can have sub-linear [1] and even logarithmic [6, 8, 42] communication complexity. However, despite the progress on IOP-based efficient generalized ZKPs and lattice-based zkRPs, IOP-based specialized ZKPs are somewhat more subdued and there has not been any IOP-based zkRP yet as far as we know. Note that the natural way to construct an IOP-based zkRP using a generalized ZKP may not be efficient enough as there could be a fixed cost of outputting paths to a Merkle tree in the range of 100-200 KBs [2].

Apart from the performance such as the communication complexity and the security against quantum adversaries, the functionality is also a consideration when constructing zkRPs especially in practice. For example, a single prover sometimes needs to perform multiple range proofs at the same time, thus an efficient batch zkRP is required [10, 17]. Besides, while most zkRPs aim for ranges like $[0, 2^n - 1]$ for integer n, practical applications may require a more general range such as $[0, u^n - 1]$ [12, 22] for integer $u > 2$ and $[A, B - 1]$ for arbitrary non-negative integers A and B [35].

Our goal in this work is to build a new zkRP with more functionalities and good performance. Specifically, we try to

> *obtain an efficient batch IOP-based zkRP supporting arbitrary ranges for arbitrary bases with logarithmic communication and practically small proof size.*

1.1 Our Contributions

In this paper, we propose a zkRP to answer the above question. This scheme is the first specialized IOP-based zkRP to the best of our knowledge. It is transparent with logarithmic communication complexity. It only uses Reed-Solomon

[1] https://www.getmonero.org/

[2] https://beam.mw/

code and hash functions and is plausibly post-quantum secure. This scheme is a public-coin argument of knowledge and can be transformed into a non-interactive one using the standard Fiat-Shamir transformation in the random oracle model (ROM)[3]. Our zkRP has functionalities such as proving $[A, B-1]$ for arbitrary non-negative integers A and B and multiple ranges with multiple bases.

Comparisons on performance and functionalities. The comparisons of our zkRP with Bulletproofs and two of the most efficient latticed-based zkRPs, i.e., LNS20 [35] and CKLR21 [17], are listed in Table 1 and Table 2. The communication complexity of our scheme and Bulletproofs are both logarithmic, while both LNS20 and CKLR21 are linear. Although the verifier complexity of our scheme is quasi-linear, the main cost for the verifier computation is due to the FFT operations for public polynomials. These operations could be delegated using a protocol called GKR protocol [24] and the verifier complexity will be reduced to logarithmic after using delegation. When performing the zkRP for a single secret value, our scheme is smaller than LNS20 with the range dimension n larger than 128. For an arbitrary range $[A, B-1] \subsetneq [0, 2^{512}-1]$, our proof size is 20% shorter than LNS20. Our scheme is 10-40 times shorter than the batch zkRP in CKLR21. With respect to the computation cost, our scheme is competitive with LNS20, especially in the batch setting. Apart from the advantage in post-quantum security, our zkRP has both at least 10 times faster prover time and verifier time compared with Bulletproofs.

1.2 Technical Overview

At the heart of our construction is a new batch IPA based on IOP. It allows proving that the inner products of multiple committed vectors and multiple public vectors equal respective public scalars. The main technique of this batch IPA is univariate sum-check protocol [6] and fast Reed-Solomon interactive oracle proof of proximity (FRI) [3]. To give a better understanding of our work, We first explain how IPAs work in zkRPs by introducing Bulletproofs [10], after that we present our techniques.

How IPAs Work in ZkRPs To prove an integer value V with upper bound 2^m is in the range $[0, 2^n-1]$, a prover could decompose V into a bit representation, i.e., a binary vector $\boldsymbol{v} \in \mathbb{F}^m$ in some field $\mathbb{F}$. The prover then shows to the verifier that: (1) Each entry in $\boldsymbol{v}$ is 0 or 1. (2) The first $m-n$ entry in $\boldsymbol{v}$ is 0. This equals two Hadamard product relations:

$$\boldsymbol{v} \odot (\boldsymbol{v} - \boldsymbol{1^m}) = \boldsymbol{0^m}, \tag{1}$$

$$\boldsymbol{v} \odot (\boldsymbol{1^{m-n}} || \boldsymbol{0^n}) = \boldsymbol{0^m}. \tag{2}$$

[3] Similar to [35], we show the security of our scheme in ROM based on the hardness of a quantum-safe problem rather than the more desirable security reduction in the quantum ROM (QROM). The recent results of [20, 21, 29, 34] give evidence that schemes secure in ROM may be still secure against quantum adversaries. There also has not been an example of any natural scheme proved secure in the ROM based on a quantum-safe problem that ends up being insecure in the QROM.

Scheme	LNS20 [35]	CKLR21 [17]	Bulletproofs [10]	Ours
Assumption	Module-LWE Module-SIS [31] ROM	LWE SIS ROM	DLOG ROM	ROM
Communication complexity	$O(n)\,\lvert\mathbb{R}\rvert$	$O(n)\,\lvert\mathbb{R}\rvert$	$O(\log n)\,\lvert\mathbb{F}\rvert$ $O(\log n)\,\lvert\mathbb{G}\rvert$	$O(\log n)\,\lvert\mathbb{F}\rvert$
Plausibly post-quantum secure?	yes	yes	no	yes
Prover complexity	$O(n\log n)\,\mathrm{FFT}$ $O(\log n)\,\mathbb{R}$	$O(n)\,\mathbb{R}$	$O(n)\,\mathbb{G}_E$	$O(n\log n)\,\mathrm{FFT}$ $O(\log n)\,\mathbb{F}$
Verifier complexity	$O(n\log n)\,\mathrm{FFT}$ $O(\log n)\,\mathbb{R}$	$O(n)\,\mathbb{R}$	$O(n)\,\mathbb{G}_E$	$O(n\log n)\,\mathrm{FFT}$ $O(\log n)\,\mathbb{F}$
Supporting non-trivial batch processing?	not given	yes	yes	yes
Supporting arbitrary ranges?	yes	not given	not given	yes

Table 1: A comparison of zkRPs. $|\mathbb{R}|, |\mathbb{G}|, |\mathbb{F}|$ represent the size of an element in a ring, group, and field, respectively. $\mathbb{R}, \mathbb{F}$ mean operations on a ring and field, respectively. FFT means Fast Fourier Transformation and $\mathbb{G}_E$ means group exponentiation.

Inspired by [10], we transform the above Hadamard product relations to inner product relations. Concretely, to prove that a vector $\boldsymbol{a} \in \mathbb{F}^m$ satisfies $\boldsymbol{a} = \boldsymbol{0^m}$, it suffices to prove that $\langle \boldsymbol{a}, \boldsymbol{r} \rangle = 0$ for a random vector $\boldsymbol{r} \in \mathbb{F}^m$ chosen by the verifier. If $\boldsymbol{a} \neq \boldsymbol{0^m}$ then the equality holds with probability bounded by $1/|\mathbb{F}|$ due to the random choice of $\boldsymbol{r}$. Based on the above fact, the prover could demonstrate Equations (1) and (2) by proving

$$\langle \boldsymbol{v} \odot (\boldsymbol{v} - \boldsymbol{1^m}), \boldsymbol{r} \rangle = 0, \tag{3}$$

$$\langle \boldsymbol{v} \odot (\boldsymbol{1^{m-n}} || \boldsymbol{0^n}), \boldsymbol{r} \rangle = \langle \boldsymbol{v}, \boldsymbol{r}_{[:m-n]} || \boldsymbol{0^n} \rangle = 0, \tag{4}$$

where $\boldsymbol{r}_{[:m-n]}$ means the first $m-n$ elements of the vector $\boldsymbol{v}$ and $||$ means the concatenation of vectors.

A typical method to build an argument for Equation (3) and (4) is that A prover first commits to $\boldsymbol{v}$ and sends the commitment to a verifier. After receiving the challenge $\boldsymbol{r}$, the prover and the verifier invoke an IPA to prove these two inner product relations and complete the whole zkRP.

The homomorphic property of commitments plays an important role in IPAs. This property allows the verifier to compute and construct commitments to target vectors using only the commitments sent by the prover. For example, the commitment in Bulletproofs is generalized Pedersen commitment. It has homomorphic properties for addition operations and partly Hadamard operations between a secret vector and a public vector. Using the homomorphic property, a verifier could first transform equivalent Equation (3) to $\langle \boldsymbol{v}, (\boldsymbol{v} - \boldsymbol{1^m}) \odot \boldsymbol{r} \rangle = 0$ and construct the commitment to $(\boldsymbol{v} - \boldsymbol{1^m}) \odot \boldsymbol{r}$ given $\boldsymbol{r}$ and the commitment to $\boldsymbol{v}$ hence

Scheme		LNS20 [35]	CKLR21 [17]	Bulletproofs [10]	Ours
Range type		arbitrary	-	fixed	arbitrary
(λ, n, t)	Performance				
$(120, 32, 1)$	proof size	11.8 KB	-	0.59 KB	22.4 KB
$(120, 128, 1)$	proof size	26.4 KB	-	0.72 KB	28.3 KB
	prover time	0.002 s	-	0.05 s	0.006 s
	verifier time	0.0003 s	-	0.02 s	0.003 s
$(120, 512, 1)$	proof size	51.3 KB	-	0.84 KB	40.7 KB
$(120, 128, 64)$	proof size	1.69 MB	-	1.09 KB	0.12 MB
	prover time	0.16 s	-	2.87 s	0.24 s
	verifier time	0.02 s	-	1.04 s	0.01 s
$(120, 128, 256)$	proof size	6.76 MB	-	1.22 KB	0.38 MB
	prover time	0.63 s	-	11.28 s	0.87 s
	verifier time	0.08 s	-	4.25 s	0.07 s
$(120, 128, 1024)$	proof size	27.03 MB	-	1.34 KB	1.45 MB
	prover time	2.51 s	-	48.74 s	3.42 s
	verifier time	0.34 s	-	21.55 s	0.27 s
Range type		-	fixed	-	fixed
(λ, n, t)	Performance				
$(128, 64, 180)$	proof size	-	4.52 MB	-	0.11 MB
$(128, 64, 500)$	proof size	-	4.87 MB	-	0.27 MB
$(128, 64, 1000)$	proof size	-	5.36 MB	-	0.52 MB

Table 2: A concrete performance comparison of zkRPs for the range $[0, 2^{2^n} - 1]$ for security parameter λ and instance number t. "Arbitrary" range is $[A, B-1] \subsetneq [0, 2^{2^n} - 1]$ and "fixed" range is strictly $[0, 2^{2^n} - 1]$. The implementation of LNS20 only gives a ZKP for integer addition with integers less than 2^{128} and we use it to construct a zkRP where $n = 128$. We only quote the proof size given in [35, §B] and [17, §6], thus some data is vacant (Use - to represent it). The elliptic curve in the Bulletproofs implementation is BN128 and this curve provides a security level of about 110 bits [36]. See Section 4 for more implementation details.

completing the IPA. Further, if the commitment supports full Hadamard homomorphism, a verifier could directly construct the commitment to $\boldsymbol{v} \odot (\boldsymbol{v} - \boldsymbol{1^m})$ and would not need to transform Equation (3) at all. This homomorphism would also help to construct zkRPs for u-[illegible]entation as the inner product relation $\langle \boldsymbol{v} \odot (\boldsymbol{v} - \boldsymbol{1^m}) \odot \ldots \odot (\boldsymbol{v} -$ [illegible] 0 needs to be satisfied where the homomorphic operation for secret v[illegible]ns necessary.

Our Techniques Following the typical method above, we first construct an IPA based on IOP and attempt to use it to build a zkRP. What inspires us is the univariate sum-check protocol [6], which allows a prover to show $\sum_{a \in H} \hat{f}(a) = \mu$ for a multiplicative coset H, claimed sum μ and secret polynomial $\hat{f}$. Note that if representing $\hat{f}$ as $\hat{f} \leftarrow \hat{v} \cdot \hat{r}$, the statement can be rewritten as an inner product relation:

$$\sum_{a \in H} \hat{f}(a) = \sum_{a \in H} \hat{v}(a)\hat{r}(a) = \langle \hat{v}|_H, \hat{r}|_H \rangle = \mu,$$

where $\hat{v}|_H$ and $\hat{r}|_H$ represent the evaluations of polynomials $\hat{v}$ and $\hat{r}$ on H, respectively.

To complete the univariate sum-check protocol, a prover first needs to commit the evaluations of $\hat{f}$ on a codeword domain L, i.e., $\hat{f}|_L$, using Merkle trees. Note that $\hat{f}|_L$ are indeed Reed-Solomon codes. A verifier would ask the prover to open several points on $f|_L$ and decides to accept or reject.

We observe that RS codes support both addition and Hadamard operations. Specifically, given two codes to $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}^{|H|}$, i.e., $\hat{a}|_L$ and $\hat{b}|_L$, by querying some point i in $\hat{a}|_L$ and $\hat{b}|_L$, one could compute $\hat{a}|_L[i] + \hat{b}|_L[i]$ as the value on the same location of the code to $\boldsymbol{a} + \boldsymbol{b}$. This holds due to the fact that the encoding polynomial of $\boldsymbol{a} + \boldsymbol{b}$ is indeed $\hat{a} + \hat{b}$. Similarly, one could construct $\hat{a}|_L[i] \cdot \hat{b}|_L[i]$ as the value on the same location of the code to $\boldsymbol{a} \odot \boldsymbol{b}$. Let $\boldsymbol{c} \leftarrow \boldsymbol{a} \odot \boldsymbol{b}$ and $\hat{c} \leftarrow \hat{a} \cdot \hat{b}$. It holds that for every $j \in [|L|], \hat{c}|_L[i] = \hat{a}|_L[i] \cdot \hat{b}|_L[i]$. Note that the degree of $\hat{c}$ is bounded by $2|H| - 1$.

The property that RS codes support both addition and Hadamard operations raises the probability to construct zkRPs with u-ary representation. However, there are still several difficulties to build an efficient zkRP described as follows.

The combination of multiple inner product relations. Two inner product relations need to be satisfied when proving that a binary vector is in a specific range. A straightforward way is to invoke two IPAs and two complete univariate sum-check protocols. This will lead to a twice larger proof size than a single univariate sum-check protocol. Inspired by [6], a prover and a verifier could invoke the univariate sum-check protocol on a random linear combination of polynomials to be tested, where the randomness comes from the verifier's challenge. This is feasible due to the linearity of RS codes. However, the combination is not trivial as the degrees of the two polynomials are different, bounded by $2|H| - 1$ for Equation (3) and $|H|$ for Equation (4), respectively. Inspired by [6], we construct a batch IPA which allows proving multiple inner product relations even with different secret polynomial degrees.

Batch range proofs. Apart from proving that a single value is in a range $[0, 2^n - 1]$, practical applications usually have more complicated requirements such as proving that multiple values are in the same range $[0, 2^n - 1]$. We observe that in order to complete a batch range proof, it suffices to argue that multiple inner product relations are satisfied. Using the batch IPA, we could combine different inner product relations using random linear combinations and build a batch range proof. This protocol even supports proving that multiple values are in multiple different ranges.

Proving arbitrary ranges. Given an integer V, We also build a range proof to prove $V \in [A, B-1]$ for arbitrary non-negative integers A and B. Inspired by [12], this can be achieved by proving $V - A \in [0, u^n - 1]$ and $V - B + u^n \in [0, u^n - 1]$ if $u^{n-1} < B < u^n$ holds.

Our first attempt is to invoke the range proof with inputs $V - A$ and $V - B + u^n$. Let RS$[\boldsymbol{w}]$ represent the RS code to the vector $\boldsymbol{w}$. This needs to construct the commitments for RS$[\boldsymbol{v} - \boldsymbol{a}]$ and RS$[\boldsymbol{v} - \boldsymbol{b} + \boldsymbol{0^{m-n-1}}||\boldsymbol{1}||\boldsymbol{0^{n-1}}]$ according to RS$[\boldsymbol{v}]$, where $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{1}||\boldsymbol{0^{n-1}}$ are the u-ary representations of A, B and u^n, respectively. However, taking $u = 2$ as an example, there would be not only 0 and 1 in $\boldsymbol{v} - \boldsymbol{a}$ but also $p - 1$ if the underlying field is $\mathbb{F}_p$. It seems tough to construct any constraint for $p - 1$. Besides, the positions of $0, 1$ and $p - 1$ do not have an apparent regularity. The two aspects both bring difficulties to construct inner product relations.

Our solution is that a prover could additionally send the commitments to binary vectors $\boldsymbol{c}, \boldsymbol{d}$ for values $C = V - A$ and $D = V - B + u^n$. The prover first shows that C, D are valid by proving $\langle \boldsymbol{v} - \boldsymbol{a} - \boldsymbol{c}, \boldsymbol{2^m} \rangle = 0$ and $\langle \boldsymbol{v} - \boldsymbol{b} + \boldsymbol{bi}(u^n) - \boldsymbol{d}, \boldsymbol{2^m} \rangle = 0$, where $\boldsymbol{bi}(u^n)$ is the binary vector representation for u^n. This can also be achieved by the batch IPA. After that, the prover could prove that C, D are both in valid ranges using a range proof.

Optimization for proof size. The main overhead for the proof size is due to FRI [3], which is a sub-protocol of the univariate sum-check protocol. In the FRI, it needs to open leaves for secret vectors committed by Merkle trees. Although the depth of each tree is logarithmic to the vector length, there are usually dozens of KB as the "start-up" cost [35] if implementing directly FRI according to the original paper [3]. One of the "start-up" costs for batch zkRPs is the communication of verification paths when committing multiple secret codewords using Merkle trees. A straightforward way is to construct a Merkle tree for every codeword, but this leads to a verification path size linear to the number of secret vectors. We observe that the queried locations to the evaluations of these codewords are all the same. Therefore, it suffices to place every element that will be queried together in one leaf of the Merkle tree. As a result, the number of Merkle trees and the size of the verification path is unrelated to the number of secret vectors. This helps to reduce the proof size.

1.3 Related Work

Zero-Knowledge Range Proofs There are two high-level approaches for constructing zkRPs. One method is square decomposition and the other is n-ary decomposition. The first method relies on the following idea. To prove a secret value V lies in a range $[A, B-1]$, it suffices to prove that both $V-A$ and $B-V$ are non-negative. To prove a value X is non-negative, it suffices to prove $4X+1$ is the sum of three squares [26]. A common issue of works [18, 26, 32] based on the first method is that they require the use of RSA groups or class groups with a hard-to-factor discriminant, which leads to very large element size and poor performance in practice. Recently, Couteau et al. [17] propose a highly-efficient modular paradigm and get range proofs based only on the DLOG assumption or plain LWE and SIS assumption. For zkRPs based on DLOG assumption, the proof size for a 64-bit range and 128-bit security is less than 1 KB, competitive to Bulletproofs. With the same parameters, the post-quantum RPs based on LWE and SIS assumption have a concrete proof size of 5.36 MB for 1000 instances.

The n-ary decomposition method is used in the latest state-of-the-art protocols [10, 16, 19]. B*ünz* et al. [10] introduce Bulletproof, which has a logarithmic communication complexity under the plain DLOG assumption (without pairings) with high computational efficiency. Bulletproofs uses the DLOG-based inner product argument [9] as its main technique, which is later improved by a line of works [19, 28, 43]. As for post-quantum zkRPs using this method, Attema et al. [2] propose an argument for proving multiplicative relations between committed values and use it to construct a 32-bit range proof with 120-bit security based on the module LWE/SIS assumption. Lyubashevsky et al. [35] present a practical latticed-based scheme for showing that committed integers satisfy additive relationships based on module LWE/SIS assumption and construct a post-quantum zkRP based on this scheme. This RP has a linear communication complexity and good performance. For an arbitrary range $[A, B-1] \subsetneq [0, 2^{512}-1]$ and 120-bit security, the proof size is 51.3 KB.

Inner Product Arguments An IPA allows a prover to convince a verifier that the inner product of two encoded vectors equals a public scalar. Vectors are usually encoded by either RS code [8, 42] or by Pedersen hash [9, 10]. In this paper, we follow the IPAs encoded by RS code implicit in [42]. With respect to IPAs encoded by Pedersen hash, Bootle et al. [9] and Bunz et al. [10] design two IPAs with logarithmic communication complexity and linear verifier complexity. The verifier complexity is reduced to logarithmic level in [19] and [11]. However, both of them rely on structured commitment keys thus needing a trusted setup.

2 Preliminaries

We use capital italic letters such as A to represent integers. Let $\mathbb{F}$ be a finite field. Let italic letters with cap such as $\hat{f}$ represent polynomials on a field. Given a multiplicative coset H of $\mathbb{F}$, let $\hat{p}|_H$ be the evaluations of $\hat{p}$ evaluated on H. We use bold lowercase letters such as $\boldsymbol{a}$ to represent vector and a_i denotes the i-th element of $\boldsymbol{a}$. Define $\boldsymbol{c^m}$ to be a m-length vector containing the first m powers of a constant c., i.e., $\boldsymbol{c^m} = (1, c, \cdots, c^{m-1})$. For a vector $\boldsymbol{a} \in \mathbb{F}^n$, denote $\boldsymbol{a}_{[:n-k]}$ by $(a_1, a_2, \ldots, a_{n-k}) \in \mathbb{F}^{n-k}$ for integers $0 < k < n$. For $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}^n$, we use $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ and $\boldsymbol{a} \odot \boldsymbol{b}$ to denote the inner product and Hadamard product of $\boldsymbol{a}$ and $\boldsymbol{b}$, respectively.

We denote the security parameter by λ and "PPT" means probabilistic polynomial time. We use $\mathsf{negl}(\cdot)$ to denote a negligible function, which means that for all polynomials $\hat{f}$, $\mathsf{negl}(k) < 1/f(k)$ for sufficiently large integer k. We use $y \leftarrow A(x)$ to represent the process that on input x, algorithm A outputs y. We use $y \overset{\$}{\leftarrow} S$ to denote uniformly and randomly pick y from set S. For a positive integer n, we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$.

Merkle tree. A Merkle tree [37] is a primitive which enables one to commit a vector and open it at several indexes with a logarithmic-size proof. The commitment to a vector $\boldsymbol{v}$ consists of three algorithms:

- $\mathsf{root}_{\boldsymbol{v}} \leftarrow \mathsf{MT.Commit}(\boldsymbol{v})$
- $(\{v_i\}_{i \in \mathcal{I}}, \pi_{\mathcal{I}}^{\boldsymbol{v}}) \leftarrow \mathsf{MT.Open}(\mathcal{I}, \boldsymbol{v})$
- $(1, 0) \leftarrow \mathsf{MT.Verify}(\mathsf{root}_{\boldsymbol{v}}, \mathcal{I}, \{v_i\}_{i \in \mathcal{I}}, \pi_{\mathcal{I}}^{\boldsymbol{v}})$.

In this paper, we use Merkle trees constructed by collision-resistant and noninvertible hash functions.

Reed-Solomon code. Given a multiplicative coset L of $\mathbb{F}$ and a code rate $\rho \in (0, 1)$, we denote by $\mathrm{RS}[L, \rho] \in \mathbb{F}^{|L|}$ all evaluations over L of univariate polynomials with degree less than $\rho|L|$. In this paper, RS code is mainly used for encoding vectors. Let $H = \{\xi_1, \ldots, \xi_{|H|}\}$ be the interpolation set and $L = \{\eta_1, \ldots, \eta_{|L|}\}$ be the evaluation set ($|L| > |H|$). The procedure of encoding vector $\boldsymbol{v} \in \mathbb{F}^{|H|}$ is as follows. Find an *encoding polynomial* $\hat{p}$ with a prearranged degree such that $\hat{p}|_H = \boldsymbol{v}$. Then evaluate $\hat{p}$ on L to obtain the *codeword* $\hat{p}|_L$. The computation of evaluating and interpolating the encoding polynomial is achieved by Fast Fourier Transform (FFT) and its inverse (IFFT).

2.1 Honest Verifier Zero-Knowledge Argument of Knowledge

An honest verifier zero-knowledge argument of knowledge (ZKAoK) for an NP binary relation $\mathcal{R}$ is a tuple of algorithms $(\mathcal{G}, \mathcal{P}, \mathcal{V})$. $\mathcal{G}$ represents the public parameter generation algorithm. $\mathcal{P}$ and $\mathcal{V}$ represent a PPT prover and verifier, respectively. After several rounds of interactions, $\mathcal{V}$ is convinced that $\mathcal{P}$ knows $\mathtt{w}$ such that $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$ for a public statement $\mathtt{x}$ but $\mathcal{V}$ gains no extra knowledge beyond that.

Definition 1 (Honest verifier ZKAoK [42]). *We call $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is an honest verifier ZKAoK for relation $\mathcal{R}$ if the following holds.*

- ***Completeness.*** *For every $\mathsf{pp} \leftarrow \mathcal{G}(1^\lambda)$, every $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$ and every $\mathtt{z} \in \{0, 1\}^*$,*

$$\Pr[\langle \mathcal{P}(\mathtt{w}), \mathcal{V}(\mathtt{z}) \rangle (\mathsf{pp}, \mathtt{x}) = 1] = 1 - \mathsf{negl}(\lambda).$$

We say $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is perfectly complete if the probability is 1.

- ***Soundness:*** *For every* $\mathsf{pp} \leftarrow \mathcal{G}(1^\lambda)$*, any* PPT *prover* $\mathcal{P}^*$*, every* $(\mathtt{x}, \mathtt{w}) \notin \mathcal{R}$ *and every* $\mathsf{z} \in \{0,1\}^*$*,*
$$\Pr[\langle \mathcal{P}^*(\mathtt{w}), \mathcal{V}(\mathsf{z})\rangle(\mathsf{pp}, \mathtt{x}) = 1] \leq \mathsf{negl}(\lambda).$$
We say $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ *is perfectly sound if the probability is 0.*
- ***Honest verifier zero-knowledge:*** *There exists a* PPT *simulator* $\mathcal{S}$ *such that for every* $\mathsf{pp} \leftarrow \mathcal{G}(1^\lambda)$*, any honest* PPT $\mathcal{V}$*, every* $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$ *and every* $\mathsf{z} \in \{0,1\}^*$*,*
$$\{\langle \mathcal{P}(\mathtt{w}), \mathcal{V}(\mathsf{z})\rangle(\mathsf{pp}, \mathtt{x})\} \overset{c}{\approx} \{\mathcal{S}^{\mathcal{V}}(\mathsf{pp}, \mathtt{x}, \mathsf{z})\}.$$
Here $\mathcal{S}^{\mathcal{V}}$ *denotes the simulator* $\mathcal{S}$ *is given the randomness of* $\mathcal{V}$ *from a polynomial-size space, and* $\overset{c}{\approx}$ *means computationally indistinguishable.*
- ***Argument of knowledge:*** *For any malicious* PPT *prover* $\mathcal{P}^*$*, there exists an expected polynomial time extractor* $\mathcal{E}$ *such that for every* $\mathsf{pp} \leftarrow \mathcal{G}(1^\lambda)$*, any* $\mathtt{x}$ *and every* $\mathsf{z} \in \{0,1\}^*$*,*
$$\Pr[\langle \mathcal{P}^*(\mathtt{w}), \mathcal{V}(\mathsf{z})\rangle(\mathsf{pp}, \mathtt{x}) = 1 \wedge (\mathtt{x}, \mathtt{w}) \notin \mathcal{R} \,|\, \mathtt{w} \leftarrow \mathcal{E}^{\mathcal{P}^*}(\mathsf{pp}, \mathtt{x})] \leq \mathsf{negl}(\lambda).$$
Here $\mathcal{E}^{\mathcal{P}^*}$ *denotes that the extractor* $\mathcal{E}$ *has access to the entire executing process and the randomness of* $\mathcal{P}^*$*.*

2.2 Interactive Oracle Proof

An interactive oracle proof (IOP) [7,39] for an NP binary relation $\mathcal{R}$ with round complexity k is a tuple of PPT algorithms $(\mathcal{G}, \mathcal{P}, \mathcal{V})$. $\mathcal{G}$ represents the public parameter generation algorithm. $\mathcal{P}$ and $\mathcal{V}$ represent a PPT prover and verifier, respectively. In the first round, $\mathcal{P}$ sends oracle π_1 to $\mathcal{V}$. In the ith round ($1 < i \leq k$), $\mathcal{V}$ sends a uniform and random challenge m_{i-1}, then $\mathcal{P}$ returns an oracle π_i. At the end of the proof, $\mathcal{V}$ queries q locations of $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)$ and decides to accept or reject.

Definition 2 (Interactive Oracle Proof [6]). *We call* $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ *an IOP for an NP binary relation* $\mathcal{R}$ *if the following holds.*

- ***Completeness:*** *For every* $\mathsf{pp} \leftarrow \mathcal{G}(1^\lambda)$*, every* $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$*,*
$$\Pr[\langle \mathcal{P}(\mathtt{w}), \mathcal{V}^{\boldsymbol{\pi}}\rangle(\mathsf{pp}, \mathtt{x}) = 1] = 1.$$
Here $\mathcal{V}^{\boldsymbol{\pi}}$ *means that* $\mathcal{V}$ *has the oracle access to* $\boldsymbol{\pi}$*.*
- ***Soundness:*** *For every* $\mathsf{pp} \leftarrow \mathcal{G}(1^\lambda)$*, every* PPT $\mathcal{P}^*$ *and every* $(\mathtt{x}, \mathtt{w}) \notin \mathcal{R}$*,*
$$\Pr[\langle \mathcal{P}^*(\mathtt{w}), \mathcal{V}^{\boldsymbol{\pi}}\rangle(\mathsf{pp}, \mathtt{x}) = 1] \leq \mathsf{negl}(\lambda).$$

In this paper, two variants of IOPs are involved, *RS-encoded IOP* and *IOP of proximity* (IOPP). On a high level, the former is an IOP where all the oracles are RS codes. The latter is an IOP allowing a small distance between a prover's secret and the valid witness for soundness property.

2.3 Univariate Sum-check

Given two multiplicative groups $H, L \subset \mathbb{F}$ ($|L| > |H|$), a univariate polynomial $f(\cdot)$ with degree less than k ($k > |H|$) and a claimed sum μ, the univariate sum-check protocol proves $\sum_{a \in H} f(a) = \mu$. In particular, the prover $\mathcal{P}$ uniquely decomposes $f(x)$ as $x \cdot \hat{g}(x) + \zeta + \hat{Z}_H(x)\hat{h}(x)$, where $\hat{Z}_H(x)$ is the vanishing polynomial on H. Then given the oracle access to $\hat{f}|_L, \hat{h}|_L$, the verifier checks if $\hat{p}|_L \in \mathrm{RS}[L, (|H|-1)/|L|]$ and $\hat{h}|_L \in \mathrm{RS}[L, (|L|-|H|)/|L|]$, where $\hat{p}(x) = (|H| \cdot \hat{f}(x) - \mu - |H| \cdot \hat{Z}_H(x)\hat{h}(x))/x$. This RS-encoded IOP is perfect completeness and perfect sound [6, Theorem 5.2]. We give a formal RS-encoded IOP for univariate sum-check in Appendix A.

When transforming the above RS-encoded IOP to an IOP, it remains to check the oracles $\hat{f}|_L, \hat{h}|_L, \hat{p}|_L$ are RS codes with corresponding upper degree bounds. This could be completed by the low degree test protocol described as follows.

2.4 Low Degree Test and FRI

Given claimed degrees $k_1, \ldots, k_t$, codewords $\hat{v}_1|_L, \ldots, \hat{v}_t|_L$ and a multiplicative coset L, a low degree test protocol allows a verifier with oracle access to these codewords to check whether for all $j \in [t]$, $\hat{v}_j|_L \in \mathrm{RS}[L, k_j/|L|]$. In this paper, we use the *fast Reed-Solomon interactive oracle proof of proximity* (FRI) [3] as our low degree test protocol. The properties of FRI is presented below. A full FRI protocol and the detailed soundness are introduced in Appendix B.

Lemma 1. *[6, Theorem 8.1] FRI is an IOPP with perfect completeness and soundness error $O(|L|/|\mathbb{F}|) + \mathsf{negl}(\ell, k)$, given oracle access to evaluations of every prover's message at ℓ points, where $\ell = O(\lambda)$ and $k = \max\{k_1, \ldots, k_t\}$. The prover complexity is $O(|L|)$ field operation. The verifier complexity is $O(\log |L|)$ field operation other than the oracle access. The communication complexity is $O(\log |L|)$ field element.*

We denote the FRI protocol for the univariate sum-check protocol in Section 2.3 as

$$\langle \mathsf{FRI}.\mathcal{P}(\hat{f}, \hat{h}, \hat{p}), \mathsf{FRI}.\mathcal{V}^{\hat{f}|_L, \hat{h}|_L, \hat{p}|_L} \rangle (k, k - |H|, |H| - 1).$$

3 A Range Proof based on Interactive Oracle Proof

In this section, we formally propose our range proofs. We first give a batch IPA where the degree of secret polynomials can be different in Section 3.1. Using this batch IPA, we build a range proof that enables one to prove a value is in a u-ary range (See Section 3.2). This batch IPA also helps to construct a batch range proof which supports proving that multiple values are in multiple ranges (See Section 3.3). In Section **??**, we modify the range proof to support proving that a secret integer is in an arbitrary range. Finally, we add zero-knowledge property and construct a zkRP in Section 3.5.

3.1 A Batch Inner Product Argument

A key property of the univariate sum-check protocol is that it supports checking every sum for multiple univariate polynomials, even with different degrees [6, §5.2]. This indicates the possibility to construct an IPA that supports checking inner product relations for vectors with encoding polynomials of different degrees. Specifically, let the secret encoding polynomials be $\hat{v}_1, \dots, \hat{v}_t$ with degrees $k_1, \dots, k_t$. Let the public encoding polynomials be $\hat{r}_1, \dots, \hat{r}_t$ with degrees $k_{t+1}, \dots, k_{2t}$. Suppose the prover $\mathcal{P}$ wants to prove that for all $j \in [t]$, $\sum_{a \in H} \hat{v}_j(a) \cdot \hat{r}_j(a) = y_j$. To complete this, $\mathcal{P}$ could first generate commitments for $(\hat{v}_1|_L, \dots, \hat{v}_t|_L)$ using Merkle trees and sends roots to the verifier. $\mathcal{V}$ next chooses random elements $\beta_1, \dots, \beta_t$. Let $\hat{q} \leftarrow \sum_{j=1}^{t} \beta_j \hat{v}_j \cdot \hat{r}_j$, $\mathcal{P}$ and $\mathcal{V}$ finally invoke the univariate sum-check protocol to prove $\sum_{a \in H} \hat{q}(a) = \sum_{a \in H} \sum_{j=1}^{t} \beta_j \hat{v}_j(a) \hat{r}_j(a) = \sum_{j=1}^{t} y_j$. An exciting point is that although the polynomials aggregated by $\beta_1, \dots, \beta_t$ have different degrees $k_1 + k_{t+1}, \dots, k_t + k_{2t}$, the soundness error is only related to the max degree $k_{\max} = \max\{k_i + k_{t+i}\}_{i \in [t]}$. The whole protocol is listed in Fig. 1.

Definition 3 (Batch inner product relation). *The relation $\mathcal{R}_{\text{B-IPA}}$ is the set of all pairs $(\mathtt{x}, \mathtt{w})$, where*

$$\mathtt{x} = \left(\mathbb{F}, H, L, \{k_j\}_{j \in [2t]}, \{\hat{r}_j\}_{j \in [t]}, \{y_j\}_{j \in [t]}\right)$$
$$\mathtt{w} = \{\hat{v}_j\}_{j \in [t]}.$$

$\mathbb{F}$ is a finite prime field. L, H are multiplicative cosets of $\mathbb{F}$ where $|L| > 2|H|$ and $|L| > 2 \cdot k_{\max} = 2 \cdot \max\{k_i + k_{t+i}\}_{i \in [t]}$. It holds that for all $j \in [t]$, $\sum_{a \in H} \hat{v}_j(a) \cdot \hat{r}_j(a) = y_j$.

Theorem 1. *The batch IPA in Fig. 1 for relation $\mathcal{R}_{\text{B-IPA}}$ is an argument of knowledge with perfect completeness and soundness error $1/|\mathbb{F}| + O(|L|/|\mathbb{F}|) + \mathsf{negl}(\ell, k_{\max}/|L|)$ in the random oracle model.*

Proof. **Completeness.** If for $\forall j \in [t]$, $\sum_{a \in H} \beta_j v_j(a) r_j(a) = y_j$, then we have

$$\sum_{a \in H} \hat{q}(a) = \sum_{a \in H} \sum_{j=1}^{t} \beta_j \hat{v}_j(a) \hat{r}_j(a) = \sum_{j=1}^{t} \beta_j y_j,$$

which is a univariate sum-check relation. By the completeness of the univariate sum-check protocol, the completeness follows. Note that due to the linearity of RS code, the verifier could construct the oracle $\hat{q}|_L$ according to oracles $\hat{v}_1|_L, \dots, \hat{v}_t|_L$.

Soundness. The soundness error comes from the following two cases.

1. Case 1. Suppose that an invalid univariate sum-check relation is satisfied due to the choice of random linear combination. Without generality, we suppose

Prover $\mathcal{P}(\mathtt{x},\mathtt{w})$	To prove $(\mathtt{x},\mathtt{w}) \in \mathcal{R}_{\text{B-IPA}}$, where $\mathtt{x} = \mathbb{F}, H, L, \{k_j\}_{j\in[2t]}, \{\hat{r}_j\}_{j\in[t]}, \{y_j\}_{j\in[t]}$, $\mathtt{w} = \{\hat{v}_j\}_{j\in[t]}$.	Verifier $\mathcal{V}(\mathtt{x})$
For all $j \in [t]$, set $\hat{v}_j\vert_L \leftarrow \mathrm{FFT}(\hat{v}_j, L)$ $\mathsf{root}_{\mathbb{V}\vert_L} \leftarrow \mathsf{MT.Commit}(\mathbb{V}\vert_L)$	$\xrightarrow{\mathsf{root}_{\mathbb{V}\vert_L}}$	
	$\xleftarrow{\beta_1,\ldots,\beta_t}$	$\beta_1,\ldots,\beta_t \xleftarrow{\$} \mathbb{F}$
$\hat{q} \leftarrow \sum_{j=1}^{t} \beta_j \hat{v}_j \cdot \hat{r}_j$ Decompose $\hat{q}(x)$ as $x \cdot \hat{g}(x) + \zeta + \hat{Z}_H(x)\hat{h}(x)$ $\hat{p}(x) \leftarrow \frac{\vert H\vert \cdot \hat{q}(x) - \mu - \vert H\vert \cdot \hat{Z}_H(x)\hat{h}(x)}{x}$ $\hat{h}\vert_L \leftarrow \mathrm{FFT}(\hat{h}, L)$ $\mathsf{root}_{\hat{h}\vert_L} \leftarrow \mathsf{MT.Commit}(\hat{h}\vert_L)$	$\xrightarrow{\mathsf{root}_{\hat{h}\vert_L}}$	
$\mathsf{FRI}.\mathcal{P}(\mathtt{x}_{\mathsf{FRI}}, \mathtt{w}_{\mathsf{FRI}})$ where $\mathtt{x}_{\mathsf{FRI}} = (k_{\max}, k_{\max} - \vert H\vert, \vert H\vert - 1)$, $\mathtt{w}_{\mathsf{FRI}} = (\hat{q}, \hat{h}, \hat{p})$.		$\mathsf{FRI}.\mathcal{V}^{\hat{q}\vert_L, \hat{h}\vert_L, \hat{p}\vert_L}(\mathtt{x}_{\mathsf{FRI}})$
	$\xleftarrow{\text{location set } \mathcal{I} \subset [\vert L\vert], \vert\mathcal{I}\vert = \kappa}$	
$(\{\mathbb{V}\vert_L[i]\}_{i\in\mathcal{I}}, \pi_{\mathcal{I}}^{\mathbb{V}\vert_L}) \leftarrow \mathsf{MT.Open}(\mathcal{I}, \mathbb{V}\vert_L)$ $(\{h\vert_L[i]\}_{i\in\mathcal{I}}, \pi_{\mathcal{I}}^{\hat{h}\vert_L}) \leftarrow \mathsf{MT.Open}(\mathcal{I}, \hat{h}\vert_L)$	$\xrightarrow{\{\mathbb{V}\vert_L[i], \hat{h}\vert_L[i]\}_{i\in\mathcal{I}}, \pi_{\mathcal{I}}^{\mathbb{V}\vert_L}, \pi_{\mathcal{I}}^{\hat{h}\vert_L}}$	
		$\mathsf{MT.Verify}(\mathsf{root}_{\mathbb{V}\vert_L}, \mathcal{I}, \{\mathbb{V}\vert_L[i]\}_{i\in\mathcal{I}}, \pi_{\mathcal{I}}^{\mathbb{V}\vert_L})$ $\mathsf{MT.Verify}(\mathsf{root}_{\hat{h}\vert_L}, \mathcal{I}, \{h\vert_L[i]\}_{i\in\mathcal{I}}, \pi_{\mathcal{I}}^{\hat{h}\vert_L})$

Fig. 1: The batch IPA $\langle \mathsf{IPA_B}.\mathcal{P}(\mathtt{w}), \mathsf{IPA_B}.\mathcal{V}\rangle(\mathtt{x})$. Here $\mathbb{V}|_L \in \mathbb{F}^{t\times|L|}$ means a matrix where the i-th row is $\hat{v}_i|_L[1], \ldots, \hat{v}_i|_L[|L|]$. $\mathsf{MT.Commit}(\mathbb{V}|_L)$ represents putting every column of $\mathbb{V}|_L$ into one leaf.

that for all $j \in [t]$, $\sum_{a\in H} \hat{v}_j(a)\hat{r}_j(a) = y'_j$, and for an inconsistency set $Q \subset [t]$, $y'_q \neq y_q$ holds for $q \in Q$. For simplicity, we assume that $t \in Q$.
Then, for all $\beta_1, \ldots, \beta_{t-1}$ randomly chosen by the verifier, the probability of $\sum_{j=1}^{t} \beta_j y'_j = \sum_{j=1}^{t} \beta_j y_j$ holds if and only if

$$\beta_t = \frac{\beta_1(y_1 - y'_1) + \ldots + \beta_{t-1}(y_{t-1} - y'_{t-1})}{y'_t - y_t}.$$

This happens with probability $1/|\mathbb{F}|$.

2. Case 2. Suppose that the univariate sum-check relation is invalid. That is to say,

$$\sum_{a\in H} \hat{q}(a) = \sum_{a\in H} \sum_{j=1}^{t} \beta_j \hat{v}_j(a)\hat{r}_j(a) \neq \sum_{j=1}^{t} y_j.$$

The soundness error comes from three aspects below.

- If the RS-encoded IOP is not valid, then by the soundness of the univariate sum-check protocol, this is perfectly sound.
- If the FRI is not valid, then the soundness error is bounded by $\epsilon_{\mathsf{FRI}} = O(|L|/|\mathbb{F}|) + \mathsf{negl}(\ell, k_{\max}/|L|)$.
- If the root of the Merkle tree is not valid or any authentication path is not correct, the soundness error is bounded by $\mathsf{negl}(\lambda)$ according to the collision-resistant property of underlying hash functions.

By the union bound argument, the soundness error of the batch IPA follows. With an appropriate choice of parameters, this error could be $\mathsf{negl}(\lambda)$.

Argument of knowledge. The batch IPA is an argument of knowledge in the random oracle model. Specifically, for any PPT adversary $\mathcal{P}^*$, there exists a PPT extractor $\mathcal{E}$ such that given the random access tape of $\mathcal{P}^*$, for every statement $\mathtt{x} = \left(\mathbb{F}, H, L, \{k_j\}_{j\in[2t]}, \{\hat{r}_j\}_{j\in[t]}, \{y_j\}_{j\in[t]}\right)$ generated by $\mathcal{P}^*$, the following probability is $\mathsf{negl}(\lambda)$:

$$\Pr\begin{bmatrix}\mathsf{root}^* \leftarrow \mathcal{P}^*(1^\lambda, \mathtt{x}), \langle \mathcal{P}^*(), \mathcal{V}\rangle(\mathsf{pp}, \mathtt{x}) = 1, \{\hat{v}_j\}_{j\in[t]} \leftarrow \mathcal{E}(1^\lambda, \mathtt{x}) : \\ \mathsf{MT.Commit}(\mathbb{V}|_L) \neq \mathsf{root}^* \vee (\mathtt{x}, \{\hat{v}_j\}_{j\in[t]}) \notin \mathcal{R}_{\text{B-IPA}}\end{bmatrix}.$$

The argument of knowledge property comes from the extractability of Merkle trees as proved in [7, Appendix A]. Given the root and sufficiently many authentication paths, there exists an efficient procedure similar to “Valiant's extractor” [41] which could extract all the committed leaves in the Merkle tree. Once these leaves are extracted, one can use IFFT algorithms to obtain the secret polynomials as $|L| > k_{\max}$. The argument of knowledge property thus follows. We give a formal proof similar to [7, 42] below.

Suppose that the Merkle tree is based on a random oracle $\mathcal{H} : \{0,1\}^{2\lambda} \to \{0,1\}^{\lambda}$, we are able to construct a PPT extractor $\mathcal{E}$ with the same random type of $\mathcal{P}^*$ working as follows:

1. $\mathcal{E}$ simulates the way the same as $\mathcal{P}^*$ queries $\mathcal{H}$, and let $(q_1, q_2, \ldots, q_{\max})$ be the queries made by $\mathcal{P}^*$ to $\mathcal{H}$ in the order they are made where duplicates are omitted. Here the order means that queries should not be generated from parent nodes to child ones. Define $q_i \in \mathcal{H}(q_j)$ if the first λ bits or the last λ bits of q_i is $\mathcal{H}(q_j)$. That is to say, q_i could be the parent node of q_j. If there exists some $i \neq j$, $\mathcal{H}(q_i) = \mathcal{H}(q_j)$ or some $i \leq j$, $q_i \in \mathcal{H}(q_j)$, $\mathcal{E}$ outputs random polynomials $\hat{v}_1, \ldots, \hat{v}_t$ and aborts.
2. $\mathcal{E}$ constructs a directed graph G according to the query set $Q = \{q_1, \ldots, q_{\max}\}$. There is an edge from q_i to q_j in G if and only if $q_i \in \mathcal{H}(q_j)$. The outdegree of each node is at most 2. When $\mathcal{P}^*$ generates $\mathsf{root}_{\mathbb{V}|_L}$ in Fig. 1, if $\mathsf{root}_{\mathbb{V}|_L}$ does not equal to $\mathcal{H}(q)$ for some $q \in Q$ with depth $\lceil \log_2 t \rceil$ of the binary tree, $\mathcal{E}$ outputs random polynomials as $\hat{v}_1, \ldots, \hat{v}_t$ and aborts, otherwise we suppose that $\mathcal{H}(q_r) = \mathsf{root}_{\mathbb{V}|_L}$ holds for some r. If any of the verification paths of the root is not valid, $\mathcal{E}$ outputs random polynomials as $\hat{v}_1, \ldots, \hat{v}_t$ and aborts.
3. $\mathcal{E}$ reads all leaves from the root q_r. If there exists any missing leaf, $\mathcal{E}$ outputs random polynomials as $\hat{v}_1, \ldots, \hat{v}_t$ and aborts, otherwise, it concatenates these leaf strings as $(\hat{v}_1|_L, \ldots, \hat{v}_t|_L)$, and interpolates them to obtain polynomials $\hat{v}_1, \ldots, \hat{v}_t$ using IFFT. Note that L is large enough to do the interpolation. Therefore, $\mathcal{E}$ could efficiently output $\hat{v}_1, \ldots, \hat{v}_t$.

Let E_1 be the event $\langle \mathcal{P}^*, \mathcal{V}\rangle(\mathsf{pp}, \mathtt{x}) = 1$, and E_2 be the event $(\mathtt{x}, \{\hat{v}_j\}_{j\in[t]}) \notin \mathcal{R}_{\text{B-IPA}}$. Next, we show $\Pr[E_1 \wedge E_2] \leq \mathsf{negl}(\lambda)$.

Suppose that $\mathcal{E}$ aborts before constructing the graph G. If for some $i \neq j$, $\mathcal{H}(q_i) = \mathcal{H}(q_j)$, then $\mathcal{E}$ finds a collision. This probability is $\mathsf{negl}(\lambda)$ due to the collision-resistant property of $\mathcal{H}$. If for some $i \leq j$, $q_i \in \mathcal{H}(q_j)$, then $\mathcal{E}$ could generate a Merkle tree violating the logical order. This probability is $\mathsf{negl}(\lambda)$ since $\mathcal{H}$ is noninvertible.

Otherwise, we suppose that $\mathcal{E}$ has successfully constructed a graph. If some node on a verification path does not lie in the graph G, $\mathcal{P}^*$ has to guess the value to construct a valid verification path, and this probability is $\mathsf{negl}(\lambda)$ since $\mathcal{H}$ is noninvertible. Additionally, if any leaf of the tree is missing, then $\mathcal{V}$ will be convinced with probability $\mathsf{negl}(\lambda)$ once it queries this leaf. The probability that this leaf is not be queried by $\mathcal{V}$ is at most $(1 - 1/|L|)^{\ell} = \mathsf{negl}(\lambda)$ as $\ell = O(\lambda)$. If $\mathcal{E}$ does not abort, it could always extract some $\hat{v}_1, \ldots, \hat{v}_t$ efficiently. In this case, $\mathcal{V}$ accepts the statement with probability $\mathsf{negl}(\lambda)$ with an appropriate choice of parameters according to the soundness.

Therefore, we have

$$\begin{aligned}\Pr[E_1 \wedge E_2] &= \Pr[E_1 \wedge E_2 \,|\, \mathcal{E}\,\text{aborts}] + \Pr[E_1 \wedge E_2 \,|\, \mathcal{E}\,\text{does not abort}] \\ &\leq \Pr[\mathcal{E}\,\text{aborts}] + \Pr[E_1 \wedge E_2 \,|\, \mathcal{E}\,\text{does not abort}] \\ &\leq \mathsf{negl}(\lambda) + \mathsf{negl}(\lambda) = \mathsf{negl}(\lambda).\end{aligned}$$

An optimization for the proof size. A direct way to commit $(\hat{v}_1|_L, \ldots, \hat{v}_t|_L)$ is constructing t Merkle trees for every secret encoding polynomial. We observe that in the FRI protocol, the ℓ queries for every $\hat{v}_i|_L$ are used for constructing virtual oracle queries for $\hat{f}|_L, \hat{h}|_L$ and $\hat{p}|_L$. More importantly, in order to query $\hat{f}|_L[i]$ for the position $i \in \mathcal{I}$ where $\mathcal{I}$ is the queried location set, $\hat{v}_1|_L[i], \ldots, \hat{v}_t|_L[i]$ will be all queried. Thus, for a specific i, $\hat{v}_1|_L[i], \ldots, \hat{v}_j|_L[i]$ could be put into a singe leaf. As a result, we only need a single Merkle tree to commit all the secret vectors instead of t trees.

Complexity. Let $\mathcal{P}_{\mathsf{FRI}}, \mathcal{P}_{\mathsf{MT}}$ $(\mathcal{V}_{\mathsf{FRI}}, \mathcal{V}_{\mathsf{MT}})$ be the cost of the FRI protocol and Merkle tree operations for the prover (verifier), respectively. It can be calculated that the prover's main overhead is $(t{+}1)\,\mathrm{FFT}(H){+}\mathcal{P}_{\mathsf{FRI}}{+}2\,\mathcal{P}_{\mathsf{MT}}$, where $\mathrm{FFT}(S)$ is the cost of a single FFT on subset S. The main overhead for the verifier is $\mathcal{V}_{\mathsf{FRI}} + 2\,\mathcal{V}_{\mathsf{MT}}$, where $\mathcal{V}_{\mathsf{FRI}}$ and $\mathcal{V}_{\mathsf{MT}}$ are both $O(\log|L|)$. The communication complexity is $O(t\,\log|L|)\,|\mathbb{F}| + 2\,\pi_{\mathsf{MT}}(|L|,\ell)\,|\mathsf{H}| + \pi_{\mathsf{MT.FRI}}\,|\mathsf{H}|$, where $\pi_{\mathsf{MT}}(n,m)$ is communication of the hash number when opening m entries in a n-length vector for a Merkle tree. $\pi_{\mathsf{MT.FRI}}$ means the communication of the hash number in the FRI protocol. Note that the communication of hashes is unrelated to the number of IPA instances, and the total communication complexity is logarithmic.

3.2 A Range Proof for the Range $[0, u^n - 1]$

In order to prove some secret value V with upper bound 2^m is in a range $[0, 2^n - 1]$, it suffices to prove that Equation (3) and (4) are satisfied. Similarly, to prove V with upper bound u^m is in range $[0, u^n - 1]$, it needs to prove that $\boldsymbol{v}$, the u-ary representation of V, satisfies:

$$\boldsymbol{v} \odot (\boldsymbol{v} - \mathbf{1}^{\boldsymbol{m}}) \odot \cdots \odot (\boldsymbol{v} - \boldsymbol{u}^{\boldsymbol{m}}) = \mathbf{0}^{\boldsymbol{m}}, \tag{5}$$

$$\boldsymbol{v} \odot (\mathbf{1}^{\boldsymbol{m}-\boldsymbol{n}}||\mathbf{0}^{\boldsymbol{n}}) = \mathbf{0}^{\boldsymbol{m}}. \tag{6}$$

Further, it suffices to prove

$$\langle \boldsymbol{v} \odot (\boldsymbol{v} - \boldsymbol{1^m}) \odot \cdots \odot (\boldsymbol{v} - \boldsymbol{u^m}), \boldsymbol{r} \rangle = 0, \tag{7}$$

$$\langle \boldsymbol{v}, \boldsymbol{r}_{[:m-n]} || \boldsymbol{0^n} \rangle = 0 \tag{8}$$

with soundness error $1/|\mathbb{F}|$, for a random $\boldsymbol{r} \in \mathbb{F}$ chosen by the verifier.

Equations (7) and (8) could be proved by our batch IPA. Let the encoding polynomial of $\boldsymbol{v}$ be $\hat{v}$, then the secret polynomial corresponding to $\boldsymbol{v} \odot (\boldsymbol{v} - \boldsymbol{1^m}) \odot \cdots \odot (\boldsymbol{v} - \boldsymbol{u^m})$ will be $\hat{w} \leftarrow \hat{v} \cdot (\hat{v} - 1) \cdots (\hat{v} - u)$. Note that by querying some value in $\hat{v}|_L$, one could construct the corresponding value in $\hat{w}|_L$. Let the encoding polynomials of $\boldsymbol{r}, \boldsymbol{r}_{[:m-n]} || \boldsymbol{0^n}$ be $\hat{r}, \hat{s}$, respectively. Then we can invoke the batch IPA in Fig. 1 with

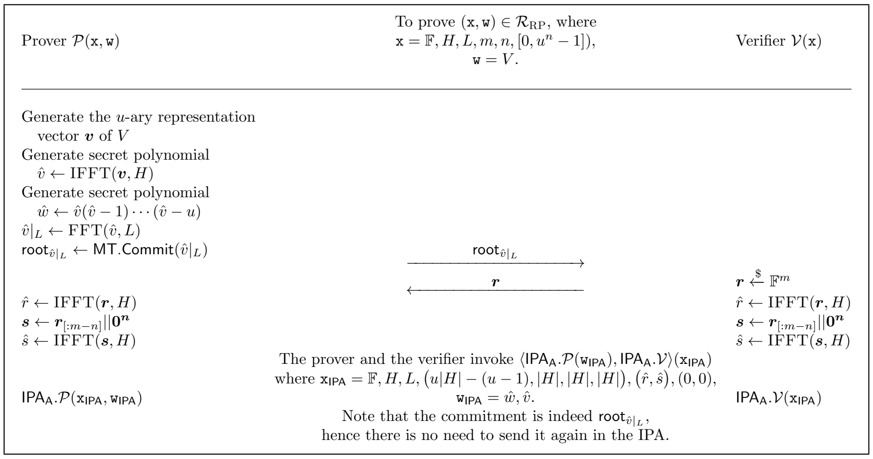
$$\begin{aligned} \mathtt{x} &= \Big(\mathbb{F}, H, L, \big(u|H| - (u-1)\big), |H|, |H|, |H|\big), \big(\hat{r}, \hat{s}\big), (0,0)\Big), \\ \mathtt{w} &= \big(\hat{w}, \hat{v}\big). \end{aligned}$$

The whole protocol is listed in Fig. 2, and the security properties of this protocol are described in Theorem 2.

Definition 4 (Range relation). *The relation $\mathcal{R}_{\mathrm{RP}}$ is the set of all pairs $(\mathtt{x}, \mathtt{w})$, where*

$$\mathtt{x} = \big(\mathbb{F}, H, L, m, n, [0, u^n - 1]\big), \quad \mathtt{w} = V.$$

$\mathbb{F}$ is a finite field, L, H are multiplicative cosets of $\mathbb{F}$, $|L| > (u+1)|H| - u$ and $m = |H|$. The secret value V satisfies $V \in [0, u^n - 1]$.



Prover $\mathcal{P}(\mathtt{x}, \mathtt{w})$

To prove $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}_{\mathrm{RP}}$, where
$\mathtt{x} = \mathbb{F}, H, L, m, n, [0, u^n - 1]),$
$\mathtt{w} = V.$

Verifier $\mathcal{V}(\mathtt{x})$

Generate the u-ary representation vector $\boldsymbol{v}$ of V
Generate secret polynomial $\hat{v} \leftarrow \mathrm{IFFT}(\boldsymbol{v}, H)$
Generate secret polynomial $\hat{w} \leftarrow \hat{v}(\hat{v} - 1) \cdots (\hat{v} - u)$
$\hat{v}|_L \leftarrow \mathrm{FFT}(\hat{v}, L)$
$\mathsf{root}_{\hat{v}|_L} \leftarrow \mathsf{MT.Commit}(\hat{v}|_L)$

$\xrightarrow{\mathsf{root}_{\hat{v}|_L}}$

$\boldsymbol{r} \overset{\$}{\leftarrow} \mathbb{F}^m$

$\xleftarrow{\boldsymbol{r}}$

$\hat{r} \leftarrow \mathrm{IFFT}(\boldsymbol{r}, H)$
$\boldsymbol{s} \leftarrow \boldsymbol{r}_{[:m-n]} || \boldsymbol{0^n}$
$\hat{s} \leftarrow \mathrm{IFFT}(\boldsymbol{s}, H)$

$\hat{r} \leftarrow \mathrm{IFFT}(\boldsymbol{r}, H)$
$\boldsymbol{s} \leftarrow \boldsymbol{r}_{[:m-n]} || \boldsymbol{0^n}$
$\hat{s} \leftarrow \mathrm{IFFT}(\boldsymbol{s}, H)$

$\mathsf{IPA_A}.\mathcal{P}(\mathtt{x}_{\mathsf{IPA}}, \mathtt{w}_{\mathsf{IPA}})$

The prover and the verifier invoke $\langle \mathsf{IPA_A}.\mathcal{P}(\mathtt{w}_{\mathsf{IPA}}), \mathsf{IPA_A}.\mathcal{V} \rangle (\mathtt{x}_{\mathsf{IPA}})$
where $\mathtt{x}_{\mathsf{IPA}} = \mathbb{F}, H, L, \big(u|H| - (u-1), |H|, |H|, |H|\big), \big(\hat{r}, \hat{s}\big), (0,0),$
$\mathtt{w}_{\mathsf{IPA}} = \hat{w}, \hat{v}.$
Note that the commitment is indeed $\mathsf{root}_{\hat{v}|_L}$,
hence there is no need to send it again in the IPA.

$\mathsf{IPA_A}.\mathcal{V}(\mathtt{x}_{\mathsf{IPA}})$

Fig. 2: The range proof $\langle \mathsf{RP}.\mathcal{P}(\mathtt{w}), \mathsf{RP}.\mathcal{V} \rangle (\mathtt{x})$.

Theorem 2. *The range proof in Fig. 2 for relation $\mathcal{R}_{\mathrm{RP}}$ is an argument of knowledge with perfect completeness and soundness error $2/|\mathbb{F}| + O(|L|/|\mathbb{F}|) + \mathsf{negl}(\ell, ((u+1)|H| - u)/|L|)$ in the random oracle model.*

Proof. **Completeness.** If Equations (5) and (6) hold, Equations (7) and (8) are satisfied, and these are proved by the batch IPA. The completeness of the range proof holds due to the completeness of the batch IPA.
Soundness. The soundness error comes from the two cases below.

1. Case 1. All the inner product relations in Equations (7) and (8) are satisfied, while the corresponding Hadamard relations in Equations (5) and (6) are not all satisfied. This soundness error is bounded by $1/|\mathbb{F}|$ due to the random choice of $\boldsymbol{r}$.
2. Case 2. There exist some inner product relations in Equations (7) and (8) which are not satisfied. This soundness error is bounded by $1/|\mathbb{F}| + O(|L|/|\mathbb{F}|) + \mathsf{negl}(\ell, ((u+1)|H| - u)/|L|)$ according to the soundness of the batch IPA.

By the union bound argument, the total soundness error follows.
Argument of knowledge. The argument of knowledge property follows from that of the batch IPA. After extracting the secret polynomial $\hat{v}$, one can efficiently compute $\boldsymbol{v} \leftarrow \hat{v}|_H$ and then obtain the value V represented by $\boldsymbol{v}$.

Complexity. The overhead of the range proof in Fig. 2 comes mainly from the overhead in the batch IPA. The prover's cost is $2\,\mathrm{IFFT}(H) + 2\,\mathrm{FFT}(L) + \mathcal{P}_{\mathsf{FRI}} + 2\,\mathcal{P}_{\mathsf{MT}}$. The main overhead for the verifier is $2\,\mathrm{IFFT}(H) + \mathcal{V}_{\mathsf{FRI}} + 2\,\mathcal{V}_{\mathsf{MT}}$. The communication complexity is $O(\log|L|)\,|\mathbb{F}| + 2 \cdot \pi_{\mathsf{MT}}(|L|, \ell)\,|\mathsf{H}| + \pi_{\mathsf{MT.FRI}}\,|\mathsf{H}|$.

3.3 A Batch Range Proof for Multiple Ranges

Using the batch IPA, it is also feasible to construct a batch range proof. This proof enables us to prove that multiple secret values are in their respective ranges. To prove secret values $V_1, \ldots, V_t$ with common upper bound $u_{\max}^m - 1$ are in ranges $[0, u_1^{n_1} - 1], \ldots, [0, u_t^{n_t} - 1]$, we can demonstrate that for all $j \in [t]$,

$$\boldsymbol{v_j} \odot (\boldsymbol{v_j} - \mathbf{1}^{\boldsymbol{m}}) \odot \cdots \odot (\boldsymbol{v_j} - \boldsymbol{u_j^m}) = \mathbf{0}^{\boldsymbol{m}}, \tag{9}$$

$$\boldsymbol{v_j} \odot (\mathbf{1}^{\boldsymbol{m-n_j}} || \mathbf{0}^{\boldsymbol{n_j}}) = \mathbf{0}^{\boldsymbol{m}}, \tag{10}$$

where $u_{\max} = \max\{u_1, \ldots, u_t\}$ and $m \geq \max\{n_1, \ldots, n_t\}$. These equations can be proved by showing that for all $j \in [t]$,

$$\langle \boldsymbol{v_j} \odot (\boldsymbol{v_j} - \mathbf{1}^{\boldsymbol{m}}) \odot \cdots \odot (\boldsymbol{v_j} - \boldsymbol{u_j^m}), \boldsymbol{r} \rangle = 0, \tag{11}$$

$$\langle \boldsymbol{v_j}, (\boldsymbol{r}^{\boldsymbol{m-n_j}} || \mathbf{0}^{\boldsymbol{n_j}}) \rangle = 0. \tag{12}$$

Similarly, to prove the relations in Equation (11) and (12) are satisfied, we could again invoke the batch IPA in Fig. 1 with

$$\mathtt{x} = \Big(\mathbb{F}, H, L, \{k_j\}_{j\in[4t]}, \{\hat{r}_j\}_{j\in[2t]}, \{y_j\}_{j\in[2t]}\Big),$$
$$\mathtt{w} = \hat{w}_1, \ldots, \hat{w}_t, \hat{v}_1, \ldots, \hat{v}_t,$$

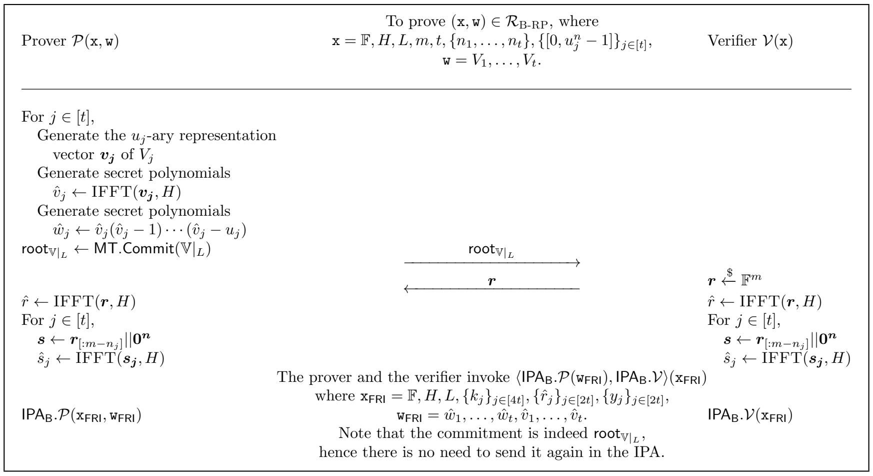
where we have

$$
\begin{aligned}
\{k_j\}_{j\in[t]} &= \underbrace{u_1|H|-(u_1-1),\dots,,u_t|H|-(u_t-1)}_{t},\\
\{k_j\}_{j\in[t+1,4t]} &= \underbrace{|H|,\dots,|H|}_{3t},\\
\{\hat{r}_j\}_{j\in[2t]} &= \underbrace{\hat{r},\dots,\hat{r}}_{t},\underbrace{\hat{s}_1,\dots,\hat{s}_t}_{t},\\
\{y_j\}_{j\in[2t]} &= \underbrace{0,\dots,0}_{2t}.
\end{aligned}
$$

For all $j \in [t]$, $\hat{s}_j$ and $\hat{v}_j$ are the encoding polynomials of $\boldsymbol{r^{m-n_j}}||\boldsymbol{0^{n_j}}, \boldsymbol{v_j}$, respectively. Besides,

$$\hat{w}_j = \hat{v}_j(\hat{v}_j-1)\cdots(\hat{v}_j-u_j).$$

We formally propose a batch range proof in Fig. 3.



Prover $\mathcal{P}(\mathtt{x},\mathtt{w})$

To prove $(\mathtt{x},\mathtt{w}) \in \mathcal{R}_{\text{B-RP}}$, where
$\mathtt{x} = \mathbb{F}, H, L, m, t, \{n_1,\dots,n_t\}, \{[0,u_j^n-1]\}_{j\in[t]}$,
$\mathtt{w} = V_1,\dots,V_t$.

Verifier $\mathcal{V}(\mathtt{x})$

For $j \in [t]$,
Generate the u_j-ary representation vector $\boldsymbol{v_j}$ of V_j
Generate secret polynomials $\hat{v}_j \leftarrow \text{IFFT}(\boldsymbol{v_j}, H)$
Generate secret polynomials $\hat{w}_j \leftarrow \hat{v}_j(\hat{v}_j-1)\cdots(\hat{v}_j-u_j)$
$\mathsf{root}_{\mathbb{V}|_L} \leftarrow \mathsf{MT.Commit}(\mathbb{V}|_L)$

$\xrightarrow{\mathsf{root}_{\mathbb{V}|_L}}$

$\boldsymbol{r} \xleftarrow{\$} \mathbb{F}^m$

$\xleftarrow{\boldsymbol{r}}$

Prover:
$\hat{r} \leftarrow \text{IFFT}(\boldsymbol{r}, H)$
For $j \in [t]$,
$\boldsymbol{s} \leftarrow \boldsymbol{r}_{[:m-n_j]}||\boldsymbol{0^n}$
$\hat{s}_j \leftarrow \text{IFFT}(\boldsymbol{s_j}, H)$

Verifier:
$\hat{r} \leftarrow \text{IFFT}(\boldsymbol{r}, H)$
For $j \in [t]$,
$\boldsymbol{s} \leftarrow \boldsymbol{r}_{[:m-n_j]}||\boldsymbol{0^n}$
$\hat{s}_j \leftarrow \text{IFFT}(\boldsymbol{s_j}, H)$

The prover and the verifier invoke $\langle \mathsf{IPA_B}.\mathcal{P}(\mathtt{w}_{\mathsf{FRI}}), \mathsf{IPA_B}.\mathcal{V}\rangle(\mathtt{x}_{\mathsf{FRI}})$
where $\mathtt{x}_{\mathsf{FRI}} = \mathbb{F}, H, L, \{k_j\}_{j\in[4t]}, \{\hat{r}_j\}_{j\in[2t]}, \{y_j\}_{j\in[2t]}$,
$\mathtt{w}_{\mathsf{FRI}} = \hat{w}_1,\dots,\hat{w}_t,\hat{v}_1,\dots,\hat{v}_t$.
Note that the commitment is indeed $\mathsf{root}_{\mathbb{V}|_L}$,
hence there is no need to send it again in the IPA.

$\mathsf{IPA_B}.\mathcal{P}(\mathtt{x}_{\mathsf{FRI}}, \mathtt{w}_{\mathsf{FRI}})$

$\mathsf{IPA_B}.\mathcal{V}(\mathtt{x}_{\mathsf{FRI}})$

Fig. 3: The batch range proof $\langle \mathsf{RP_B}.\mathcal{P}(\mathtt{w}), \mathsf{RP_B}.\mathcal{V}\rangle(\mathtt{x})$.

Definition 5 (Batch range relation). *The relation $\mathcal{R}_{\text{A-RP}}$ is the set of all pairs $(\mathtt{x},\mathtt{w})$, where*

$$\mathtt{x} = \big(\mathbb{F}, H, L, m, t, \{n_1,\dots,n_t\}, \{[0,u_j^n-1]\}_{j\in[t]}\big), \quad \mathtt{w} = V_1,\dots,V_t.$$

$\mathbb{F}$ is a finite field, L, H are multiplicative cosets of $\mathbb{F}$, $|L| > (u_{\max}+1)|H| - u_{\max}$ and $m = |H|$. It is satisfied that for all $j \in [t]$, $V_j \in [0, u_j^n - 1]$.

Theorem 3. *The batch range proof in Fig. 3 for relation $\mathcal{R}_{\text{A-RP}}$ is an argument of knowledge with perfect completeness and soundness error $2/|\mathbb{F}| + O(|L|/|\mathbb{F}|) + \mathsf{negl}(\ell, ((u_{\max}+1)|H| - u_{\max})/|L|)$ in the random oracle model.*

Proof. **Completeness.** The batch range proof is indeed a batch IPA in Fig. 1. The completeness directly follows from the completeness of the batch IPA.
Soundness. Similar to the soundness analysis in range proof (See Fig. 2), the soundness error comes from two cases.

1. Case 1. The inner product relations in Equation (11) and (12) are satisfied while the Hadamard relations in Equation (9) and (10) are not satisfied. This soundness error is bounded by $1/|\mathbb{F}|$.
2. Case 2. The inner product relations in Equation (11) and (12) are not satisfied. This soundness error is bounded by $1/|\mathbb{F}| + O(|L|/|\mathbb{F}|) + \mathsf{negl}(\ell, ((u_{\max} + 1)|H| - u_{\max})/|L|)$ according to the soundness of the batch IPA.

By the union bound argument, the total soundness error follows.
Argument of knowledge. The argument of knowledge property follows from that of the range proof, and we omit the proof here.

Complexity. The overhead of the batch range proof in Fig. 3 comes mainly from the overhead in the batch IPA. The prover's cost is $O(t)\,\mathrm{IFFT}(H) + O(t)\,\mathrm{FFT}(L) + \mathcal{P}_{\mathsf{FRI}} + 2\,\mathcal{P}_{\mathsf{MT}}$. The main overhead for the verifier is $O(t)\,\mathrm{IFFT}(H) + \mathcal{V}_{\mathsf{FRI}} + 2\,\mathcal{V}_{\mathsf{MT}}$, where $\mathcal{V}_{\mathsf{FRI}}$ and $\mathcal{V}_{\mathsf{MT}}$ are both $O(\log|L|)$. The communication complexity is $O(t \cdot \log|L|)\,|\mathbb{F}| + 2 \cdot \pi_{\mathsf{MT}}(|L|, \ell)\,|\mathsf{H}| + \pi_{\mathsf{MT.FRI}}\,|\mathsf{H}|$.

3.4 A Range Proof for Arbitrary Ranges

A more generalized statement about range proof is to prove a secret value $V \in [A, B-1]$ for arbitrary integers A and B. Inspired by [12], if we have a range proof for range $[0, u^n - 1]$, then it is possible to handle the range $[A, B-1]$.

Specifically, suppose that $u^{n-1} < B < u^n$. To show $V \in [A, B-1]$, it suffices to prove that

$$V \in [A, A + u^n - 1] \wedge V \in [B - u^n, B - 1].$$

Note that $A > 0 > B - u^n$ and $A + u^n - 1 \geq u^n - 1 > B - 1$, hence the relation holds. This equals to

$$V - A \in [0, u^n - 1] \wedge V - B + u^n \in [0, u^n - 1]. \tag{13}$$

This can be achieved using our batch range proof in Section 3.2. For simplicity, we assume $u = 2$ in this section.

A first try to complete such a proof is to rely on the homomorphism of RS code. Specifically, after committing to the binary vector $\boldsymbol{v}$ of the value V, the verifier attempts to construct queries to the RS code of $\boldsymbol{v} - \boldsymbol{a}$ and $\boldsymbol{v} - \boldsymbol{b} + \boldsymbol{bi}(u^n)$, where $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{bi}(u^n)$ are the binary representation vectors of A, B, u^n. However, this is tough to be achieved. Take $\boldsymbol{v} - \boldsymbol{a}$ for example, there will not only be 0 and 1 in $\boldsymbol{v} - \boldsymbol{a}$ but also $p - 1$ given a finite prime field $\mathbb{F}_p$. It is difficult to construct a Hadamard product relation for $p-1$ hence an inner product relation.

Instead of focusing on the binary representation of $V - A$, we observe that the verifier could ask the prover to first prove that $V - A = C$, and then the prover shows that C is in the range $[0, 2^n - 1]$. Further, $V - A = C$ can be proved by an inner product relation according to the definition of binary representation. This proves the first relation in Equation(13). Similarly, the prover introduces another value D, prove that $V - B + u^n = D$ by an inner product argument, and finally proves $D \in [0, 2^n - 1]$. This proves the second relation in Equation(13).

Formally, let $\boldsymbol{v}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}$ be the binary vector of values V, A, B, C, D, respectively. The prover and the verifier will invoke the batch IPA to prove the following inner product relations to demonstrate that $V \in [A, B-1]$:

$$\begin{aligned} \langle \boldsymbol{v} \odot (\boldsymbol{v} - \boldsymbol{1^m}), \boldsymbol{r} \rangle &= 0, && (14) \\ \langle \boldsymbol{c} \odot (\boldsymbol{c} - \boldsymbol{1^m}), \boldsymbol{r} \rangle &= 0, && (15) \\ \langle \boldsymbol{d} \odot (\boldsymbol{d} - \boldsymbol{1^m}), \boldsymbol{r} \rangle &= 0, && (16) \\ \langle \boldsymbol{c}, \boldsymbol{r}_{[:m-n]} || \boldsymbol{0^n} \rangle &= 0, && (17) \\ \langle \boldsymbol{d}, \boldsymbol{r}_{[:m-n]} || \boldsymbol{0^n} \rangle &= 0, && (18) \\ \langle \boldsymbol{v} - \boldsymbol{a} - \boldsymbol{c}, \boldsymbol{2^m} \rangle &= 0, && (19) \\ \langle \boldsymbol{v} - \boldsymbol{b} + \boldsymbol{bi}(2^n) - \boldsymbol{d}, \boldsymbol{2^m} \rangle &= 0. && (20) \end{aligned}$$

Equations (14)-(16) are used to prove that $\boldsymbol{v}, \boldsymbol{c}, \boldsymbol{d}$ are all binary vectors. It is demonstrated in Equations (19)-(20) that C, D satisfy the calculation relation with V, A and B. Equations (17) and (18) show that the claimed ranges are valid.

To prove the relations in Equations (14)-(20), it suffices to invoke the batch IPA with

$$\begin{aligned} \mathtt{x} &= \Big(\mathbb{F}, H, L, \{k_j\}_{j\in[14]}, \{\hat{r}_j\}_{j\in[7]}, \{y_j\}_{j\in[7]}\Big), \\ \mathtt{w} &= \{\hat{v}_j\}_{j\in[7]}. \end{aligned}$$

where we have

$$\begin{aligned} \{k_j\}_{j\in[7]} &= \underbrace{2|H|-1, \ldots, , 2|H|-1}_{3}, \underbrace{|H|, \ldots, |H|}_{4}, \\ \{k_j\}_{j\in[8,14]} &= \underbrace{|H|, \ldots, |H|}_{7}, \\ \{\hat{r}_j\}_{j\in[7]} &= \underbrace{\hat{r}, \ldots, \hat{r}}_{3}, \underbrace{\hat{s}, \ldots, \hat{s}}_{2}, \underbrace{\hat{t}, \hat{t}}_{2}, \\ \{y_j\}_{j\in[7]} &= \underbrace{0, \ldots, 0}_{7}, \\ \{\hat{v}_j\}_{j\in[7]} &= \hat{w}_{\boldsymbol{v}}, \hat{w}_{\boldsymbol{c}}, \hat{w}_{\boldsymbol{d}}, \hat{v}_{\boldsymbol{c}}, \hat{v}_{\boldsymbol{d}}, \hat{u}_{\boldsymbol{c}}, \hat{u}_{\boldsymbol{d}} \end{aligned}$$

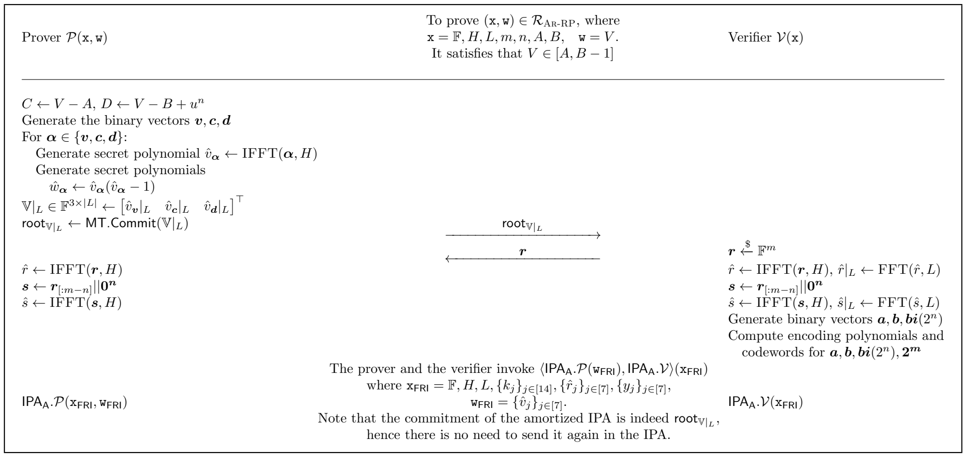
For $\boldsymbol{\alpha} \in \{\boldsymbol{v}, \boldsymbol{c}, \boldsymbol{d}\}$, $\hat{v}_{\boldsymbol{\alpha}}$ is the encoding polynomial of $\boldsymbol{\alpha}$, and $\hat{w}_{\boldsymbol{\alpha}} = \hat{v}_{\boldsymbol{\alpha}}(\hat{v}_{\boldsymbol{\alpha}} - 1)$. $\hat{u}_{\boldsymbol{c}}$ and $\hat{u}_{\boldsymbol{d}}$ are the encoding polynomials of $\boldsymbol{v} - \boldsymbol{a} - \boldsymbol{c}$ and $\boldsymbol{v} - \boldsymbol{b} + \boldsymbol{bi}(2^n) - \boldsymbol{d}$, respectively. $\hat{r}, \hat{s}, \hat{t}$ are the public encoding polynomials of vectors $\boldsymbol{r}, \boldsymbol{r}_{[:m-n]} || \boldsymbol{0^n}$ and $\boldsymbol{2^m}$, respectively.

Definition 6 (Arbitrary range relation). *The relation $\mathcal{R}_{\text{AR-RP}}$ is the set of all pairs $(\mathtt{x}, \mathtt{w})$, where*

$$\mathtt{x} = \big(\mathbb{F}, H, L, m, n, A, B\big), \quad \mathtt{w} = V.$$

$\mathbb{F}$ is a finite field, L, H are multiplicative cosets of $\mathbb{F}$, $|L| > 2|H|$ and $m = |H|$. V satisfies that $V \in [A, B-1]$.

We present a whole protocol in Fig. 4 to achieve the relation $\mathcal{R}_{\text{AR-RP}}$. The prover additionally needs to commit $\boldsymbol{c}$ and $\boldsymbol{d}$. This suffices since one can construct queries to evaluations of $\{\hat{v}_j\}_{j\in[7]}$ by querying $\hat{v}_a, \hat{v}_c, \hat{v}_d$.



Prover $\mathcal{P}(\mathtt{x}, \mathtt{w})$

To prove $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}_{\text{AR-RP}}$, where
$\mathtt{x} = \mathbb{F}, H, L, m, n, A, B, \quad \mathtt{w} = V$.
It satisfies that $V \in [A, B-1]$

Verifier $\mathcal{V}(\mathtt{x})$

```
$C \leftarrow V - A$, $D \leftarrow V - B + u^n$
Generate the binary vectors $\boldsymbol{v}, \boldsymbol{c}, \boldsymbol{d}$
For $\boldsymbol{\alpha} \in \{\boldsymbol{v}, \boldsymbol{c}, \boldsymbol{d}\}$:
  Generate secret polynomial $\hat{v}_{\boldsymbol{\alpha}} \leftarrow \mathrm{IFFT}(\boldsymbol{\alpha}, H)$
  Generate secret polynomials
    $\hat{w}_{\boldsymbol{\alpha}} \leftarrow \hat{v}_{\boldsymbol{\alpha}}(\hat{v}_{\boldsymbol{\alpha}} - 1)$
$\mathbb{V}|_L \in \mathbb{F}^{3\times|L|} \leftarrow \big[\hat{v}_{\boldsymbol{v}}|_L \quad \hat{v}_{\boldsymbol{c}}|_L \quad \hat{v}_{\boldsymbol{d}}|_L\big]^\top$
$\mathsf{root}_{\mathbb{V}|_L} \leftarrow \mathsf{MT.Commit}(\mathbb{V}|_L)$
```

$\xrightarrow{\mathsf{root}_{\mathbb{V}|_L}}$

$\boldsymbol{r} \xleftarrow{\$} \mathbb{F}^m$

$\xleftarrow{\boldsymbol{r}}$

```
$\hat{r} \leftarrow \mathrm{IFFT}(\boldsymbol{r}, H)$
$\boldsymbol{s} \leftarrow \boldsymbol{r}_{[:m-n]}||\boldsymbol{0^n}$
$\hat{s} \leftarrow \mathrm{IFFT}(\boldsymbol{s}, H)$
```

```
$\hat{r} \leftarrow \mathrm{IFFT}(\boldsymbol{r}, H)$, $\hat{r}|_L \leftarrow \mathrm{FFT}(\hat{r}, L)$
$\boldsymbol{s} \leftarrow \boldsymbol{r}_{[:m-n]}||\boldsymbol{0^n}$
$\hat{s} \leftarrow \mathrm{IFFT}(\boldsymbol{s}, H)$, $\hat{s}|_L \leftarrow \mathrm{FFT}(\hat{s}, L)$
Generate binary vectors $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{bi}(2^n)$
Compute encoding polynomials and
  codewords for $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{bi}(2^n), \boldsymbol{2^m}$
```

The prover and the verifier invoke $\langle \mathsf{IPA_A}.\mathcal{P}(\mathtt{w}_{\mathsf{FRI}}), \mathsf{IPA_A}.\mathcal{V}\rangle(\mathtt{x}_{\mathsf{FRI}})$
where $\mathtt{x}_{\mathsf{FRI}} = \mathbb{F}, H, L, \{k_j\}_{j\in[14]}, \{\hat{r}_j\}_{j\in[7]}, \{y_j\}_{j\in[7]}$,
$\mathtt{w}_{\mathsf{FRI}} = \{\hat{v}_j\}_{j\in[7]}$.
Note that the commitment of the amortized IPA is indeed $\mathsf{root}_{\mathbb{V}|_L}$,
hence there is no need to send it again in the IPA.

$\mathsf{IPA_A}.\mathcal{P}(\mathtt{x}_{\mathsf{FRI}}, \mathtt{w}_{\mathsf{FRI}})$

$\mathsf{IPA_A}.\mathcal{V}(\mathtt{x}_{\mathsf{FRI}})$

Fig. 4: The range proof $\langle \mathsf{RP_{AR}}.\mathcal{P}(\mathtt{w}), \mathsf{RP_{AR}}.\mathcal{V}\rangle(\mathtt{x})$.

Theorem 4. *The range proof for arbitrary ranges proposed in Fig. 4 for relation $\mathcal{R}_{\text{AR-RP}}$ is an argument of knowledge with perfect completeness and soundness error $2/|\mathbb{F}| + O(|L|/|\mathbb{F}|) + \mathsf{negl}(\ell, (3|H|-2)/|L|)$ in the random oracle model.*

Proof. **Completeness.** The completeness directly follows from the completeness of the range proof.

Soundness. Similar to the soundness analysis in range proof, the soundness error comes from two cases.

1. Case 1. The inner product relations in Equations (14) - (20) are satisfied while the Hadamard relations are not satisfied. This soundness error is bounded by $1/|\mathbb{F}|$.
2. Case 2. The inner product relations in Equations (14) - (20) are not satisfied. This soundness error is bounded by $1/|\mathbb{F}| + O(|L|/|\mathbb{F}|) + \mathsf{negl}(\ell, (3|H|-2)/|L|)$ according to the soundness of the batch IPA.

By the union bound argument, the total soundness error follows.
Argument of knowledge. The argument of knowledge property follows from that of the range proof, which we omit here.

Complexity. The overhead of the range proof for arbitrary ranges in Fig. 4 comes mainly from the overhead in the batch IPA. The prover's cost is $5\,\mathrm{IFFT}(H)+3\,\mathrm{FFT}(L)+\mathcal{P}_{\mathsf{FRI}}+2\mathcal{P}_{\mathsf{MT}}$. The main overhead for the verifier is $6\,\mathrm{IFFT}(H)+6\,\mathrm{FFT}(L)+\mathcal{V}_{\mathsf{FRI}}+2\,\mathcal{V}_{\mathsf{MT}}$. The communication complexity is $O(3\cdot\log|L|)\,|\mathbb{F}|+2\cdot\pi_{\mathsf{MT}}(|L|,\ell)\,|\mathsf{H}|+\pi_{\mathsf{MT}}(\mathsf{FRI})\,|\mathsf{H}|$.
Generalization. The range proof for relation $\mathcal{R}_{\text{Ar-RP}}$ could be further generalized to prove that multiple integers are in multiple respective ranges, i.e., to prove that for all $i\in[t]$, $V_i\in[A_i,B_i-1]$ for non-negative integers $\{A_i\}$ and $\{B_i\}$. The high-level idea is to transform every range relation $V_i\in[A_i,B_i-1]$ into inner product constraints like Equations (14)-(20), then use the batch IPA to prove them.

3.5 Adding Zero-Knowledge

Our range proofs are not zero knowledge mainly due to two aspects. We take the range proof in Fig. 2 as an example and all our range proofs are variates of it. One knowledge leakage is that the verifier sees ℓ evaluations of $\mathbb{V}|_L$ when opening the Merkle tree commitment for it. This is related to the secret vectors $\{\boldsymbol{v_j}\}_{j\in[t]}$, thus leaks partial information about $\{V_j\}_{j\in[t]}$. The other knowledge leakage comes from the FRI protocol. During the $O(\log|L|)$ rounds, the verifier may obtain additional knowledge from received messages. To achieve zero-knowledge property, we adapt the techniques similar in [42]. For the first leakage, the prover could pick a random polynomial $\hat{\delta}_j$ with degree ℓ and mask $\hat{v}_j$ as $\hat{v}'_j\leftarrow\hat{v}_j+\hat{Z}_H\cdot\hat{\delta}_j$, where $\hat{Z}_H$ is the vanishing polynomial on subset H with degree $|H|$. With respect to the second leakage, the prover could mask $\hat{f}$ in Fig. 2 with a random polynomial $\hat{\gamma}$ of degree $(u_{\max}+1)|H|+u_{\max}\ell-u_{\max}$.

A high-level analysis of the security for the zkRP is as follows. The completeness property holds due to the linearity of RS code and the fact that $\hat{\delta}_j$ has no effect on completeness since $\hat{v}'_j(a)=\hat{v}_j(a)$ for all $a\in H$ and $j\in[t]$. The soundness property comes from the fact that the modified polynomial to be invoked in the univariate sum-check protocol is a random linear combination of secret polynomials and random masking polynomial $\hat{\gamma}$. By the property of random linear combination, the sum-check relation is satisfied bounded with probability $1/|\mathbb{F}|$ if the inner product relations are not valid. The whole protocol is listed in Fig. 5 and the detailed security properties are analyzed in Theorem 5.

Theorem 5. *The range proof for relation $\mathcal{R}_{\mathrm{RP}}$ in Fig. 5 is an honest verifier zero-knowledge argument of knowledge with perfect completeness and soundness error $2/|\mathbb{F}|+O(|L|/|\mathbb{F}|)+\mathsf{negl}(\ell,((u_{\max}+1)|H|+u_{\max}\ell-u_{\max})/|L|)$ in the random oracle model.*

<table>
<tr><td>Prover $\mathcal{P}(\mathtt{x}, \mathtt{w})$</td><td>To prove $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}_{\mathrm{RP}}$, where
$\mathtt{x} = \mathbb{F}, H, L, m, t,$
$\{n_1, \ldots, n_t\}, \{[0, u_j^n - 1]\}_{j \in [t]},$
$\mathtt{w} = V_1, \ldots, V_t.$</td><td>Verifier $\mathcal{V}(\mathtt{x})$</td></tr>
<tr><td>For $j \in [t]$,
Generate the u_j-ary representation
vector $\boldsymbol{v_j}$ of $\mathsf{v_j}$
Generate secret polynomial
$\hat{v}_j \leftarrow \mathrm{IFFT}(\boldsymbol{v_j}, H)$
Set $\hat{v}'_j(x) \leftarrow \hat{v}_j(x) + \hat{Z}_H(x) \cdot \hat{\delta}_j(x)$
for random $\hat{\delta}_j$ with degree ℓ
Generate secret polynomial
$\hat{w}'_j \leftarrow \hat{v}'_j(\hat{v}'_j - 1) \cdots (\hat{v}'_j - u_j)$
Randomly pick $\hat{\gamma}$ with
degree $(u_{\max} + 1)|H| + u_{\max}\kappa - u_{\max}$
$\Gamma \leftarrow \sum_{a \in H} \hat{\gamma}(a)$, $\hat{\gamma}|_L \leftarrow \mathrm{FFT}(\hat{\gamma}, L)$
$\mathbb{V}'||\hat{\gamma}|_L \leftarrow (\hat{v}'_1|_L, \ldots, \hat{v}'_t|_L, \hat{\gamma}|_L)$
$\mathsf{root}_{\mathbb{V}'||\hat{\gamma}|_L} \leftarrow \mathsf{MT.Commit}(\mathbb{V}'||\hat{\gamma}|_L)$</td><td>$\xrightarrow{\Gamma, \mathsf{root}_{\mathbb{V}'||\hat{\gamma}|_L}}$</td><td></td></tr>
<tr><td></td><td>$\xleftarrow{\boldsymbol{r}, \beta_1, \ldots, \beta_t}$</td><td>$\boldsymbol{r} \overset{\$}{\leftarrow} \mathbb{F}^m, \beta_1, \ldots, \beta_t \overset{\$}{\leftarrow} \mathbb{F}$</td></tr>
<tr><td>$\hat{r} \leftarrow \mathrm{IFFT}(\boldsymbol{r}, H)$
For $j \in [t]$:
$\boldsymbol{s_j} \leftarrow \boldsymbol{r}_{[:m-n_j]}||\boldsymbol{0^{n_j}}$
$\hat{s}_j \leftarrow \mathrm{IFFT}(\boldsymbol{s_j}, H)$</td><td></td><td>$\hat{r} \leftarrow \mathrm{IFFT}(\boldsymbol{r}, H)$
For $j \in [t]$:
$\boldsymbol{s_j} \leftarrow \boldsymbol{r}_{[:m-n_j]}||\boldsymbol{0^{n_j}}$
$\hat{s}_j \leftarrow \mathrm{IFFT}(\boldsymbol{s_j}, H)$</td></tr>
<tr><td>$\hat{f} \leftarrow \sum_{j=1}^{t} \beta_j \hat{u}_j \hat{v}'_j + \hat{\gamma}$
Decompose $\hat{f}$ as
$x \cdot \hat{g}(x) + \zeta + \hat{Z}_H(x)\hat{h}(x)$
$\hat{p}(x) \leftarrow \frac{|H| \cdot \hat{f}(x) - \mu - |H| \cdot \hat{Z}_H(x)\hat{h}(x)}{x}$
$\hat{h}|_L \leftarrow \mathrm{FFT}(\hat{h}, L)$
$\mathsf{root}_{\hat{h}|_L} \leftarrow \mathsf{MT.Commit}(\hat{h}|_L)$</td><td>$\xrightarrow{\mathsf{root}_{\hat{h}|_L}}$</td><td></td></tr>
<tr><td>$\mathsf{FRI}.\mathcal{P}(\mathsf{x_{FRI}}, \mathsf{w_{FRI}})$ where
$\mathsf{x_{FRI}} = ((u_{\max} + 1)|H| + u_{\max}\ell - u_{\max},$
$u_{\max}|H| + u_{\max}\kappa - u_{\max}, |H| - 1),$
$\mathsf{w_{FRI}} = (\hat{q}, \hat{h}, \hat{p}).$</td><td></td><td>$\mathsf{FRI}.\mathcal{V}^{\hat{q}|_L, \hat{h}|_L, \hat{p}|_L}(\mathsf{x_{FRI}})$</td></tr>
<tr><td></td><td>$\xleftarrow{\text{location set } \mathcal{I} \subset [|L|], |\mathcal{I}| = \ell}$</td><td></td></tr>
<tr><td>$(\{\mathbb{V}'||\hat{\gamma}|_L[i]\}_{i \in \mathcal{I}}, \pi_{\mathcal{I}}^{\mathbb{V}'||\hat{\gamma}|_L}) \leftarrow \mathsf{MT.Open}(\mathcal{I}, \mathbb{V}'||\hat{\gamma}|_L)$
$(\{h|_L[i]\}_{i \in \mathcal{I}}, \pi_{\mathcal{I}}^{\hat{h}|_L}) \leftarrow \mathsf{MT.Open}(\mathcal{I}, \hat{h}|_L)$</td><td>$\xrightarrow{\{\mathbb{V}'||\hat{\gamma}|_L[i], \hat{h}|_L[i]\}_{i \in \mathcal{I}}, \pi_{\mathcal{I}}^{\mathbb{V}'||\hat{\gamma}|_L}, \pi_{\mathcal{I}}^{\hat{h}|_L}}$</td><td></td></tr>
<tr><td></td><td></td><td>$\mathsf{MT.Verify}(\mathsf{root}_{\mathbb{V}'||\hat{\gamma}|_L}, \mathcal{I}, \{\mathbb{V}'||\hat{\gamma}|_L\}_{i \in \mathcal{I}}, \pi_{\mathcal{I}}^{\mathbb{V}'||\hat{\gamma}|_L})$
$\mathsf{MT.Verify}(\mathsf{root}_{\hat{h}|_L}, \mathcal{I}, \{\hat{h}|_L[i]\}_{i \in \mathcal{I}}, \pi_{\mathcal{I}}^{\hat{h}|_L})$</td></tr>
</table>

Fig. 5: The zero-knowledge range proof $\langle \mathsf{RP_{zk}}.\mathcal{P}(\mathtt{w}), \mathsf{RP_{zk}}.\mathcal{V} \rangle(\mathtt{x})$.

Proof. **Completeness.** It follows from the completeness of the batch IPA in Fig. 1. Note that for every $a \in H$ and $j \in [t]$, it holds $\beta_j \hat{v}'_j \cdot \hat{r}'_j = \beta_j \hat{v}_j \cdot \hat{r}_j$, thus $\hat{\delta}_j$ plays no part on completeness.

Soundness. The soundness error comes from two cases. Here we suppose that for some $j \in [t]$, $V_j \notin [0, u_j^{n_j} - 1]$.

1. <u>Case 1.</u> Suppose that all the inner product relations are satisfied. This soundness error is bounded by $1/|\mathbb{F}|$ due to the random choice of $\boldsymbol{r}$.
2. <u>Case 2.</u> The inner product relations in Equations (11) and (12) are not satisfied. The soundness error of this case can be divided into two subcases.
   - The univariate sum-check relation is satisfied. For any random choices $\beta_1, \ldots, \beta_j \in \mathbb{F}$ and a polynomial $\hat{\gamma}^*$ carefully selected by $\mathcal{P}^*$ satisfying $\sum_{a \in H} \hat{\gamma}^*(a) = \mathit{\Gamma}^*$, $\sum_{a \in H}(\sum_{j=1}^{t} \beta_j \hat{u}_j(a) \cdot \hat{v}'_j(a)) + \mathit{\Gamma}^* = \sum_{j=1}^{t} y_j + \mathit{\Gamma}$ if and only if $\beta_1, \beta_2, \ldots, \beta_t$ satisfies a specific relation, which happens with a probability at most $1/|\mathbb{F}|$. That is to say, the probability that the univariate sum-check protocol is satisfied is bounded by $1/|\mathbb{F}|$.
   - The univariate sum-check relation is not satisfied and the FRI is not valid. The soundness error is bounded by $\epsilon_{\mathrm{FRI}} = O(|L|/|\mathbb{F}|) + \mathsf{negl}(\ell, ((u_{\max} + 1)|H| + u_{\max}\ell - u_{\max})/|L|)$. This comes from the fact that the secret polynomial with the biggest degree is $\hat{w}_i = \hat{v}_i(\hat{v}_i - 1) \cdots (\hat{v}_i - u_{\max})$ for some $i \in [t]$. The degree of $\hat{w}_i$ is $(u_{\max} + 1)|H| + u_{\max}\ell - u_{\max}$.

By the union bound argument, the total soundness error follows.

Argument of knowledge. The argument of knowledge property follows from that of the range proof. Note that the random masking polynomial $\hat{\gamma}$ is also encoded and committed by the Merkle tree, hence our protocol is an argument of knowledge.

Honest-verifier zero knowledge. We describe a simulator $\mathcal{S}$, which is given $\big(\mathbb{F}, L, H, m, t, \{n_1, \ldots, n_t\}, \{[0, u_j^{n_j} - 1]\}\big)$ and the queried location set $\mathcal{I}$ as input, computationally simulates the view of a PPT verifier $\mathcal{V}^*$ in the real protocol.

1. $\mathcal{S}$ picks random polynomials $\hat{v}'_{j,\mathsf{sim}}$ with degree less than $u_j|H| + u_j\ell - (u_j - 1)$. For each $j \in [t]$, $\mathcal{S}$ evaluates $\hat{v}'_{j,\mathsf{sim}}|_L$. It then constructs $\mathbb{V}'_{\mathsf{sim}}$. $\mathcal{S}$ generates $\hat{w}'_{1,\mathsf{sim}}, \hat{w}'_{2,\mathsf{sim}}, \ldots, \hat{w}'_{t,\mathsf{sim}}$ as the honest prover does.
2. $\mathcal{S}$ samples randomly a degree $(u_{\max} + 1)|H| + u_{\max}\ell - u_{\max}$ polynomial $\hat{\gamma}_{\mathsf{sim}}$. $\mathcal{S}$ sends $\mathcal{V}^*$ $\mathit{\Gamma}_{\mathsf{sim}} = \sum_{a \in H} \hat{\gamma}_{\mathsf{sim}}(a)$ and $\mathsf{root}_{\mathbb{V}'_{\mathsf{sim}}||\hat{\gamma}_{\mathsf{sim}}|_L} \leftarrow \mathsf{MT.Commit}(\mathbb{V}'_{\mathsf{sim}}||\hat{\gamma}_{\mathsf{sim}}|_L)$.
3. Receive $\boldsymbol{r}, \beta_1, \beta_2, \ldots, \beta_t$ from $\mathcal{V}^*$.
4. Generate $\hat{r}$ and $\{\hat{s}_1, \ldots, \hat{s}_t\}$ as the honest prover does.
5. $\mathcal{S}$ randomly picks a polynomial $\hat{p}_{\mathsf{sim}}$ with degree less than $|H| - 1$. Given the challenge location set $\mathcal{I}$ generated by $\mathcal{V}^*$, for each $a_i \in \mathcal{I}$, $\mathcal{S}$ computes h_i such that

$$x \cdot \hat{p}_{\mathsf{sim}}(a_i) = \sum_{j=1}^{t} \hat{w}'_{j,\mathsf{sim}}(a_i) \cdot \hat{r}(a_i) + \sum_{j=1}^{t} \hat{v}'_{j,\mathsf{sim}}(a_i) \cdot \hat{s}_j(a_i)$$
$$+ \hat{\gamma}_{\mathsf{sim}}(a_i) - \mathit{\Gamma}_{\mathsf{sim}} - \hat{Z}_H(a_i) h_i.$$

$\mathcal{S}$ picks a random polynomial $\hat{h}_{\mathsf{sim}}$ with degree less than $u_{\max}|H|+u_{\max}\ell-u_{\max}$ such that for each $i \in \mathcal{I}$, $\hat{h}_{\mathsf{sim}}(i) = h_i$. $\mathcal{S}$ sends $\mathsf{root}_{\hat{h}_{\mathsf{sim}}|_L} \leftarrow \mathsf{MT.Commit}(\hat{h}_{\mathsf{sim}}|_L)$.

6. $\mathcal{S}$ simulates the FRI protocol as an honest prover with $\mathcal{V}^*$.
7. $\mathcal{S}$ opens

$$\left(\{\mathbb{V}'_{\mathsf{sim}}||\hat{\gamma}_{\mathsf{sim}}|_L[i]\}_{i\in\mathcal{I}}, \pi_{\mathcal{I}}^{\mathbb{V}'_{\mathsf{sim}}||\hat{\gamma}_{\mathsf{sim}}|_L}\right) \leftarrow \mathsf{MT.Open}(\mathcal{I}, \mathbb{V}'_{\mathsf{sim}}||\hat{\gamma}_{\mathsf{sim}}|_L).$$

$$\left(\{\hat{h}_{\mathsf{sim}}|_L[i]\}_{i\in\mathcal{I}}, \pi_{\mathcal{I}}^{\hat{h}_{\mathsf{sim}}|_L}\right) \leftarrow \mathsf{MT.Open}(\mathcal{I}, \hat{h}_{\mathsf{sim}}|_L).$$

To prove zero knowledge property, for any $j \in [t]$, $\hat{v}'_{j,\mathsf{sim}}$ and $\hat{v}'_j$ are uniformly distributed, and $\hat{w}'_{j,\mathsf{sim}}, \hat{w}'_j$ are also uniformly distributed. Steps 2, 3, 4, and 6 are the same as the real world in the range proof.

In steps 5 and 7, $\hat{\gamma}_{\mathsf{sim}}$ and $\hat{\gamma}$ are both randomly picked, hence the roots and opened points are indistinguishable. As the degrees of $\hat{v}'_{j,\mathsf{sim}}, \hat{w}'_{j,\mathsf{sim}}$ and $\hat{h}_{\mathsf{sim}}$ are increased by at least ℓ for $j \in [t]$, opening ℓ evaluations of $\hat{v}'_{j,\mathsf{sim}}, \hat{w}'_{j,\mathsf{sim}}$ and $\hat{h}_{\mathsf{sim}}$ are independent and randomly distributed, which is indistinguishable from the real world. Thus the view of step 7 is also simulated by $\mathcal{S}$.

4 The Performance of Our ZkRPs

In this section, we present the concrete performances of our zkRPs. We discuss some optimizations for the proof size and give a numerical example in Section 4.1 and Section 4.2. We then show the implementations and results in Section 4.3.

4.1 Optimizations for the Proof Size

We discuss some optimizations for the proof size in this section. In the original paper of FRI [3], there are $\log_2 k_{\max}$ rounds, and the reduction strategy described in Appendix B is all 1. This may bring additional overhead on the proof size. We adapt some techniques in [38,40] and also present our method to optimizing the proof size.

The choice of the soundness [38,40]. There are two versions of the soundness error of the FRI. One is the provably one and the other is a conjecture [4]. We use the conjecture one in our implementation, which is widely adopted in practical applications such as estark [40], plonky2 [38] and the implementations of [8,42].

Putting evaluations into a singe leaf [38]. The evaluations over an entire coset are bonded and queried together. This means that once an evaluation in a coset is queried, the entire coset is all queried. Therefore, these evaluations could be put into a single leaf.

Finding the best reduction strategy. A prover can skip layers and reduces the degree of polynomials by 2^η instead of 2. Another optimization is to terminate the FRI protocol earlier than when the last layer reaches a constant value [38]. This may increase the number of field elements but the verification path size will be greatly reduced. Based on these two facts, we use an exhaustive strategy to find the best reduction strategy.

Committing to multiple secret vectors using one Merkle tree. In our range proofs, there are usually multiple secret vectors. They will be queried in the univariate sum-check protocol hence the batch FRI. For example, in the batch range proof in Figure 1, the virtual oracle can be constructed by querying the same locations of $v_1|_L, \ldots, v_t|_L$. Therefore, we can arrange $v_1|_L, \ldots, v_t|_L$ as a matrix $\mathbb{V}$ and commit it by column. As a result, there will be only 1 Merkle tree instead of t.

4.2 A Numerical Example

Suppose that the field size $|\mathbb{F}|$ is less than 2^{64}, the target range is $[0, 2^{2^9} - 1]$, the code rate ρ is 2^{-3}, and the security level λ is 120. For the zkRP described in Fig. 5, the degree of the secret polynomial $\hat{v}$ is bounded by $k = 2^9$, and the max degree of the polynomials invoked in the univariate sum-check protocol is $3k - 2 + 2\ell < 2^{11}$. Therefore, we can set $m = n = |H| = 2^9, k_{\max} = 2^{11}$ and $|L| = 2^{14}$.

The soundness error of the transformation from the Hadamard relations to inner product relations is $1/|\mathbb{F}|$, hence it needs 2 random vectors from the verifier to achieve the target security level. Based on the conjecture in Section B, the query repetition ℓ needs to satisfy $-\log_2 \rho \cdot \ell > \lambda$, and we set $\ell = 41$. It suffices to use the degree 3 extension of $\mathbb{F}$ and this achieves an interactive soundness error of $2^{11}/2^{63\times 3} = 2^{-178}$. We could use a hash function with an output size of 256 bits. To find the best reduction strategy, we perform an exhaustive search to find an appropriate setting and set $\boldsymbol{\eta} = \{3, 3\}$. The total proof size is hence

$$1\,|\mathsf{H}| + 1\,|\mathsf{H}| + 2 \cdot \mathsf{Tree}(|L|/2^{\eta_1}, \ell)\,|\mathsf{H}| + \mathsf{Tree}(|L|/2^{\eta_1+\eta_2}, \ell)\,|\mathsf{H}| +$$
$$3 \cdot \ell \cdot 2^{\eta_1}\,|\mathbb{F}| + \ell \cdot 2^{\eta_2}\,|\mathbb{F}| + 2^{k_{\max}-\eta_1-\eta_2}\,|\mathbb{F}| \approx 638\,|\mathsf{H}| + 673\,|\mathbb{F}| = 35.5\,\mathrm{KB},$$

where the $\approx$ is because the size of the authentication path is random due to the random queries to the same Merkle tree.

4.3 Implementations and Results

The implementation details and results are presented in this section.

Software and hardware. Our zkRPs are implemented in C++, available at github.com/leeweihanwickham/IOP-based-zkRP. There are about 2,000 lines of code for the main protocol, including the IPA and Merkle trees. We refer to libiop [4] for finite fields and FFTs and implement the FRI protocol ourselves.

We run all experiments on an AMD Ryzen 3900X processor with 80 GB RAM and 10 cores on a virtual machine. Our implementations are utilized without parallelization. We report the average running time of 100 executions.

Choice of parameters. The target security level is set as 120-bit and the soundness is based on the conjecture in Appendix B. To find the best strategy

[4] github.com/scipr-lab/libiop

for the smallest proof size, we perform an exhaustive search for the reduction strategy of FRI when the code rate and range dimension are fixed.

The FRI protocol requires that there must exist multiplicative cosets with order 2^k for large enough k in the underlying prime field[5]. We implement a prime field $\mathbb{F}_p$ where $p = 2^{64} - 2^{32} + 1$ to support more efficient field operations. This field has multiplicative cosets with order 2^{32} and its special properties would help accelerate multiplication and FFT operations. To achieve the target interactive soundness, we set the interactive repetition number $e = 3$.

Range type	n	$k_{\max}$	ℓ	ρ	e	λ	$\boldsymbol{\eta}$	t	π	π in [35]	π in [17]	π in [2]
Arbitrary	2^5	2^8	41	2^{-3}	3	120	$\{2,2\}$	1	22.4 KB	11.8 KB	-	
	2^6	2^9	41	2^{-3}	3	120	$\{2,2\}$	1	28.7 KB	-	-	
	2^7	2^9	41	2^{-3}	3	120	$\{2,2\}$	1	28.3 KB	26.4 KB	-	
	2^8	2^{10}	41	2^{-3}	3	120	$\{2,2\}$	1	34.8 KB	-	-	
	2^9	2^{11}	41	2^{-3}	3	120	$\{2,3\}$	1	40.7 KB	51.3 KB	-	
Fixed	2^5	2^8	33	2^{-3}	3	128	$\{3,2\}$	1	19.5 KB	-	-	5.9 KB
	2^6	2^8	33	2^{-4}	3	128	$\{1,2\}$	180	0.11 MB	-	4.52 MB	-
	2^6	2^8	33	2^{-4}	3	128	$\{1,2\}$	500	0.27 MB	-	4.87 MB	-
	2^6	2^8	33	2^{-4}	3	128	$\{1,2\}$	1000	0.52 MB	-	5.36 MB	-

Table 3: Comparisons of the proof size between our zkRP and other post-quantum zkRPs.

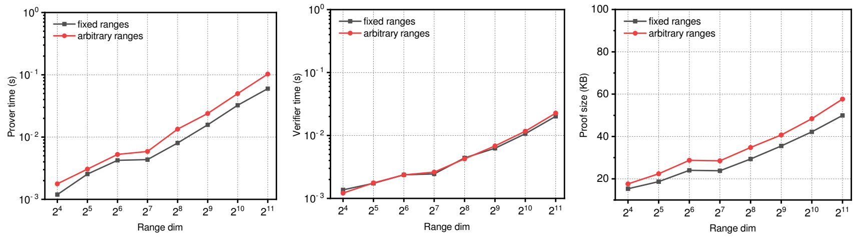
Proof size. Table 3 shows the proof sizes of LNS20 [35], CKLR21 [17] and our scheme. For an arbitrary range $[A, B-1] \subsetneq [0, 2^7 - 1]$, our scheme has a nearly the same proof size with [35]. When n becomes larger, the proof size of our range proof becomes better, as the communication complexity of [35] is linear while the communication complexity of ours is logarithmic. For an arbitrary range $[A, B-1] \subsetneq [0, 2^{512} - 1]$, our proof size is 20% smaller compared with the 51.3 KB in [35], with the same security level and essentially the same functionality[6]. Compared with the batch zkRP in [17] with fixed ranges, our proof size is 10-40 times smaller.

The zkRPs in Table 3 are in the conjecture setting. One can also run them in the provable setting and the proof sizes will become 1-2 times as large as before. We emphasize again that in practical applications, it suffices to use our scheme in the conjecture setting, and this soundness version has been applied in

[5] There is another version of FRI protocol running on additive subspaces of Galois fields. We run our protocol on multiplicative cosets since it is faster shown by our experiments.

[6] We compare our scheme with a zkRP for the range $[-2^{2^{9-1}}, 2^{2^{9-1}} - 1]$ in [35]. A value $V \in [-2^{2^{n-1}}, 2^{2^{n-1}} - 1]$ could be represented as a n-length vector $\boldsymbol{v} = (v_1, \ldots, v_n)$ such that $V = -v_n 2^{n-1} + \sum_{i=1}^{n-1} v_i 2^i$. Note that the constraints to $\boldsymbol{v}$ are the same as those when proving $\boldsymbol{v} \in [0, 2^n - 1]$. Thus it is free to prove $V \in [-2^{2^{n-1}}, 2^{2^{n-1}} - 1]$ with a zkRP for proving $V \in [0, 2^n - 1]$.

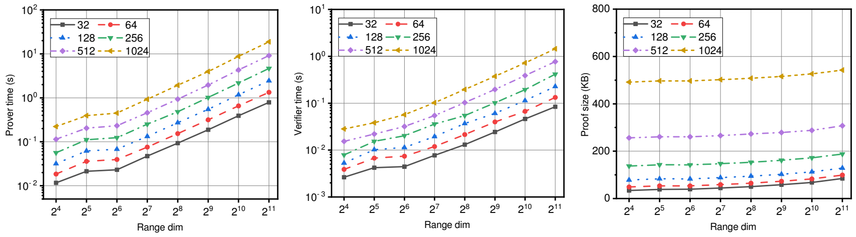
practice, especially for blockchain scalability solutions including estark [40] and plonky2 [38].



(a) prover time (b) verifier time (c) proof size

Fig. 6: Performance of zkRPs for ranges $[0, 2^{2^n} - 1]$ and $[A, B-1] \subsetneq [0, 2^{2^n} - 1]$ where $n \in [4, 11]$.

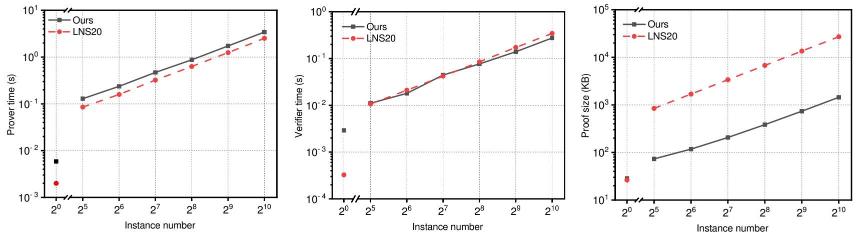
More detailed performance. Figure 6 shows a more detailed performance of our zkRP. It takes 0.06 s to generate a proof for the range $[0, 2^{2^{11}} - 1]$ and the verifier needs 0.02 s to verify. The concrete proof size is 49.9 KB. For an arbitrary range $[A, B-1] \subsetneq [0, 2^{2^{11}} - 1]$, the prover time is 0.10 s, the verifier time is 0.02s, and the proof size is 57.7 KB. Note that we perform an exhaustive search for the reduction strategy for the smallest proof size so this strategy can be different for different proofs. The prover time and the verifier time grow quasi-linearly due to the FFT operations for secret and public vectors in the IPA. The communication complexity is logarithmic. The performances for ranges $[0, 2^{2^6} - 1]$ and $[0, 2^{2^7} - 1]$ are the same since the sizes of $|L|$ are both 2^9.



(a) prover time (b) verifier time (c) proof size

Fig. 7: Performance of our zkRP in the batch setting for ranges $[0, 2^{2^n} - 1]$ where $n \in [4, 11]$.

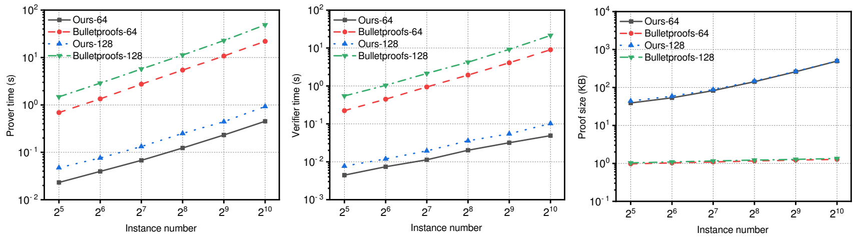
Figure 7 shows the performance of our zkRP in the batch setting. It takes 18.8 s to generate a proof for the range $[0, 2^{2^{11}} - 1]$ of 1024 instances and the verifier needs 1.4 s to verify. The concrete proof size is 542 KB. The prover and verifier time both grow linearly with the increase of instance number. The proof size does not grow linearly to the instance number since the hash size does not increase linearly, where all the secret vectors could be arranged into a matrix and committed together.



(a) prover time (b) verifier time (c) proof size

Fig. 8: Performance of zkRPs for ranges $[A, B-1] \subsetneq [0, 2^{128}-1]$.

Fig. 8 shows the comparison of our zkRPs with LNS20 [35] for ranges $[A, B-1] \subsetneq [0, 2^{128}-1]$. This implementation of LNS20 [7] gives a zero-knowledge proof for 128-bit integer addition instead of a zkRP so we run two addition proofs to construct a zkRP according to [35, §B]. We also build batch zkRPs of LNS20 trivially as it does not give any non-trivial batch zkRP. It is shown that our zkRP has a less than 1 time slower prover time than LNS20 and nearly the same verifier time in the batch setting. The proof size of our batch zkRP is at least 10 times smaller than that of LNS20. Note that for larger ranges $[A, B-1] \subsetneq [0, 2^{2^n}-1]$ where $n > 7$, the commitment number in LNS20 will changed from 4 to $4 \times 2^n/128$ to prevent the module from crossing the boundary. As one of the main computation overheads for both the prover and the verifier is the commitment-related operations and the main proof size comes from opening commitments, we believe our zkRP will have a better performance when the range is larger.



(a) prover time (b) verifier time (c) proof size

Fig. 9: A performance comparison of zkRPs in the batch setting, where "xx-64" and "xx-128" mean the ranges of scheme "xx" are $[0, 2^{64}-1]$ and $[0, 2^{128}-1]$, respectively.

[7] github.com/gregorseiler/irelzk.

Fig. 9 gives a comparison of zkRPs with Bulletproofs for ranges $[0, 2^{64} - 1]$ and $[0, 2^{128} - 1]$. ZkRPs for the range $[0, 2^{64} - 1]$ have been used in practical applications such as Monero [30] and we believe zkRPs for the range $[0, 2^{128} - 1]$ may also be practically useful in the future. Although with 30-500 times larger proof size compared with Bulletproofs, our zkRP has a 30-50 times faster prover and a 50-200 times faster verifier. It is also plausibly post-quantum secure. The implementation of Bulletproofs comes from an open-source code[8] in C. The elliptic curve in Bulletproofs is BN128, which security level in practice is estimated to be 110-bit [36], while our security level is 120-bit. Besides, the implementation of Bulletproofs only supports instance numbers which are powers of two while our scheme allows arbitrary instance numbers.

5 Conclusion

Zero-knowledge range proofs enable a prover to convince a verifier that a secret value is in a specific range while leaking no additional knowledge about the value. This could ensure a user's privacy in practical applications. We propose the first IOP-based zkRP in this paper, which is plausibly post-quantum secure like latticed-based zkRPs. Our zkRP is transparent with logarithmic communication complexity and practically efficient performance, which may contribute to practical post-quantum systems. An optimization direction is how to construct a batch zkRP based on IOP with the communication complexity sub-linear to the instance number.

References

1. Ames, S., Hazay, C., Ishai, Y., Venkitasubramaniam, M.: Ligero: Lightweight sublinear arguments without a trusted setup. In: CCS 2017A. pp. 2087–2104. ACM (2017)
2. Attema, T., Lyubashevsky, V., Seiler, G.: Practical product proofs for lattice commitments. In: CRYPTO 2020, Santa Barbara, CA, USA. pp. 470–499. Springer (2020)
3. Ben-Sasson, E., Bentov, I., Horesh, Y., Riabzev, M.: Fast reed-solomon interactive oracle proofs of proximity. In: ICALP 2018. pp. 14:1–14:17 (2018)
4. Ben-Sasson, E., Carmon, D., Ishai, Y., Kopparty, S., Saraf, S.: Proximity gaps for reed-solomon codes. In: FOCS 2020, Durham, NC, USA. pp. 900–909. IEEE (2020)
5. Ben-Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., Virza, M.: Zerocash: Decentralized anonymous payments from bitcoin. In: SP 2014. pp. 459–474. IEEE Computer Society (2014)
6. Ben-Sasson, E., Chiesa, A., Riabzev, M., Spooner, N., Virza, M., Ward, N.P.: Aurora: Transparent succinct arguments for R1CS. In: EUROCRYPT 2019. pp. 103–128. Springer (2019)
7. Ben-Sasson, E., Chiesa, A., Spooner, N.: Interactive oracle proofs. In: TCC 2016-B. pp. 31–60 (2016)

[8] github.com/xevisalle/zpie.

8. Bhadauria, R., Fang, Z., Hazay, C., Venkitasubramaniam, M., Xie, T., Zhang, Y.: Ligero++: A new optimized sublinear IOP. In: CCS 2020. pp. 2025–2038. ACM (2020)
9. Bootle, J., Cerulli, A., Chaidos, P., Groth, J., Petit, C.: Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In: EUROCRYPT 2016. pp. 327–357. Springer (2016)
10. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: SP 2018. pp. 315–334. IEEE Computer Society (2018)
11. Bünz, B., Maller, M., Mishra, P., Tyagi, N., Vesely, P.: Proofs for inner pairing products and applications. In: ASIACRYPT 2021. pp. 65–97. Springer (2021)
12. Camenisch, J., Chaabouni, R., Shelat, A.: Efficient protocols for set membership and range proofs. In: ASIACRYPT 2008, Melbourne, Australia. pp. 234–252. Springer (2008)
13. Camenisch, J., Hohenberger, S., Lysyanskaya, A.: Compact e-cash. In: EUROCRYPT 2005, Aarhus, Denmark. pp. 302–321. Springer (2005)
14. Chandler, D.B., Wu, J., Xiang, Q.: Power sums over subspaces of finite fields. Finite Fields Their Appl. **18**(4), 791–799 (2012)
15. Chaum, D.: Showing credentials without identification transfeering signatures between unconditionally unlinkable pseudonyms. In: AUSCRYPT 1990, Sydney, Australia. pp. 246–264. Springer (1990)
16. Chung, H., Han, K., Ju, C., Kim, M., Seo, J.H.: Bulletproofs+: Shorter proofs for a privacy-enhanced distributed ledger. IEEE Access **10**, 42067–42082 (2022)
17. Couteau, G., Klooß, M., Lin, H., Reichle, M.: Efficient range proofs with transparent setup from bounded integer commitments. In: EUROCRYPT 2021, Zagreb, Croatia. pp. 247–277. Springer (2021)
18. Couteau, G., Peters, T., Pointcheval, D.: Removing the strong RSA assumption from arguments over the integers. In: EUROCRYPT 2017, Paris, France, April 30 - May 4, 2017, Proceedings, Part II. pp. 321–350 (2017)
19. Daza, V., Ràfols, C., Zacharakis, A.: Updateable inner product argument with logarithmic verifier and applications. In: PKC 2020. pp. 527–557. Springer (2020)
20. Don, J., Fehr, S., Majenz, C.: The measure-and-reprogram technique 2.0: Multiround fiat-shamir and more. In: CRYPTO 2020. Lecture Notes in Computer Science, vol. 12172, pp. 602–631. Springer (2020)
21. Don, J., Fehr, S., Majenz, C., Schaffner, C.: Security of the fiat-shamir transformation in the quantum random-oracle model. In: CRYPTO 2019. Lecture Notes in Computer Science, vol. 11693, pp. 356–383. Springer (2019)
22. Eagen, L.: Bulletproofs++ (2022), https://eprint.iacr.org/2022/510
23. Feige, U., Fiat, A., Shamir, A.: Zero-knowledge proofs of identity. J. Cryptol. **1**(2), 77–94 (1988)
24. Goldwasser, S., Kalai, Y.T., Rothblum, G.N.: Delegating computation: Interactive proofs for muggles. J. ACM **62**(4), 27:1–27:64
25. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems (extended abstract). In: STOC 1985. pp. 291–304. ACM (1985)
26. Groth, J.: Non-interactive zero-knowledge arguments for voting. In: ACNS 2005, New York, NY, USA. vol. 3531, pp. 467–482 (2005)
27. Haböck, U.: A summary on the FRI low degree test (2022), https://eprint.iacr.org/2022/1216
28. Hoffmann, M., Klooß, M., Rupp, A.: Efficient zero-knowledge arguments in the discrete log setting, revisited. In: CCS 2019. pp. 2093–2110. ACM (2019)

29. Kiltz, E., Lyubashevsky, V., Schaffner, C.: A concrete treatment of fiat-shamir signatures in the quantum random-oracle model. In: EUROCRYPT 2018. Lecture Notes in Computer Science, vol. 10822, pp. 552–586. Springer (2018)
30. koe, Alonso, K.M., Noether, S.: Zero to monero: Second edition (2020), https://www.getmonero.org/library/Zero-to-Monero-2-0-0.pdf
31. Langlois, A., Stehlé, D.: Worst-case to average-case reductions for module lattices. Des. Codes Cryptogr. **75**(3), 565–599 (2015)
32. Lipmaa, H.: On diophantine complexity and statistical zero-knowledge arguments. In: ASIACRYPT 2003, Taipei, Taiwan. pp. 398–415. Springer (2003)
33. Lipmaa, H., Asokan, N., Niemi, V.: Secure vickrey auctions without threshold trust. In: FC 2002, Southampton, Bermuda. pp. 87–101. Springer (2002)
34. Liu, Q., Zhandry, M.: Revisiting post-quantum fiat-shamir. In: CRYPTO 2019. Lecture Notes in Computer Science, vol. 11693, pp. 326–355. Springer (2019)
35. Lyubashevsky, V., Nguyen, N.K., Seiler, G.: Practical lattice-based zero-knowledge proofs for integer relations. In: CCS '20, Virtual Event, USA. pp. 1051–1070. ACM (2020)
36. Menezes, A., Sarkar, P., Singh, S.: Challenges with assessing the impact of NFS advances on the security of pairing-based cryptography. In: Mycrypt 2016, Kuala Lumpur, Malaysia. Lecture Notes in Computer Science, vol. 10311, pp. 83–108. Springer (2016)
37. Merkle, R.C.: A certified digital signature. In: CRYPTO 1989. pp. 218–238. Springer (1989)
38. Polygon Zero Team: Plonky2: Fast recursive arguments with plonk and fri (2022), https://github.com/mir-protocol/plonky2/blob/main/plonky2/plonky2.pdf
39. Reingold, O., Rothblum, G.N., Rothblum, R.D.: Constant-round interactive proofs for delegating computation. In: STOC 2016. pp. 49–62. ACM (2016)
40. StarkWare Team: ethstark documentation version 1.1 (2021), https://eprint.iacr.org/2021/582
41. Valiant, P.: Incrementally verifiable computation or proofs of knowledge imply time/space efficiency. In: TCC 2008. pp. 1–18. Springer (2008)
42. Zhang, J., Xie, T., Zhang, Y., Song, D.: Transparent polynomial delegation and its applications to zero knowledge proof. In: SP 2020. pp. 859–876. IEEE (2020)
43. Zhang, Z., Zhou, Z., Li, W., Tao, H.: An optimized inner product argument with more application scenarios. In: ICICS 2021. pp. 341–357. Springer (2021)

A The Univariate Sum-check Protocol

We give a formal RS-encoded IOP for univariate sum-check in Protocol 1 and the security properties of this protocol are presented in Lemma 2.

Lemma 2 ([6]). *Protocol 1 is an RS-encoded IOP with perfect completeness and soundness.*

Proof. **Completeness.** It follows from the following lemma.

Lemma 3 ([6,14]). *Let H be a multiplicative coset of $\mathbb{F}$, and let $\hat{g}$ be a univariate polynomial over $\mathbb{F}$ of degree strictly less than $|H|$, then $\sum_{a \in H} \hat{g}(a) = |H| \cdot \hat{g}(0)$ holds.*

Protocol 1 An RS-encoded IOP for univariate sum-check

Inputs: $\big((\mathbb{F}, L, H, \rho, \mu), \hat{f}\big)$. Let $\hat{f}|_L$ be the witness oracle and $\sum_{a \in H} \hat{f}(a) = \mu$.

1. $\mathcal{P}$ uniquely decomposes $\hat{f}$ as $\hat{f}(x) = x \cdot \hat{g}(x) + \zeta + \hat{Z}_H(x)\hat{h}(x)$, where $\deg(\hat{g}) < |H| - 1$, and $\deg(\hat{h}) < \rho|L| - |H|$.
2. $\mathcal{P}$ sends oracle $\hat{h}|_L \in \mathrm{RS}[L, \rho - |H|/|L|]$ to $\mathcal{V}$.
3. $\mathcal{V}$ accepts if and only if $\hat{p}|_L \in \mathrm{RS}[L, (|H| - 1)/|L|]$, where

$$\hat{p}(x) = \frac{|H| \cdot \hat{f}(x) - \mu - |H| \cdot \hat{Z}_H(x)\hat{h}(x)}{x}. \tag{21}$$

By definition of $\hat{g}, \hat{h}$ and Lemma 3, we have

$$\mu = \sum_{a \in H} \big(a \cdot \hat{g}(a) + \zeta + \hat{Z}_H(a)\hat{h}(a)\big) = \sum_{a \in H} (a \cdot \hat{g}(a)) + |H|\zeta + 0 = |H|\zeta.$$

Therefore, we have $\zeta = \mu/|H|$. By definition of $\hat{p}$, we have

$$\begin{aligned} \hat{p}(x) &= \frac{|H| \cdot \hat{f}(x) - \mu - |H| \cdot \hat{Z}_H(x)\hat{h}(x)}{x} \\ &= \frac{|H| \cdot \Big(x \cdot \hat{g}(x) + \mu/|H| + \hat{Z}_H(x)\hat{h}(x)\Big) - \mu - |H| \cdot \hat{Z}_H(x)\hat{h}(x)}{x} = \hat{g}(x). \end{aligned}$$

Hence $\hat{p}|_L \in \mathrm{RS}[L, (|H| - 1)/|L|]$.

Soundness. Suppose $\sum_{a \in H} \hat{f}(a) = \mu' \neq \mu$. We prove that for all $\hat{h}$ satisfying $\deg(\hat{h}) < \rho|L| - |H|$, it holds that $\deg(\hat{p}) \geq |H| - 1$.

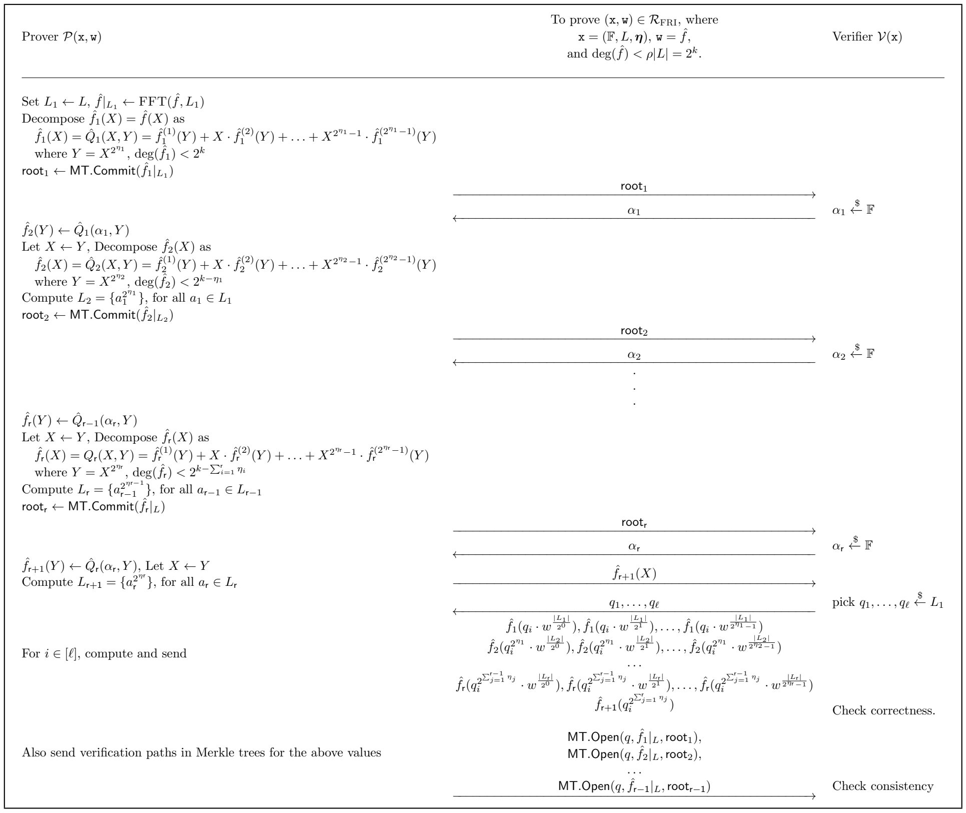
By contradiction, if $\hat{p}$ exists such that $\deg(\hat{p}) < |H| - 1$, then by Lemma 3 we have $\sum_{a \in H} a \cdot \hat{p}(a) = 0 \cdot \hat{p}(0) \cdot |H| = 0$. However, by Equation (21), we have

$$\sum_{a \in H} a \cdot \hat{p}(a) = |H| \cdot \sum_{a \in H} \hat{f}(a) - |H| \cdot \mu = |H| \cdot (\mu' - \mu) \neq 0.$$

This is a contradiction.

B FRI Oracle of Proximity

FRI, in full length *Fast Reed-Solomon Code Oracle Proof of Proximity*, is a low degree test to check multiple RS codes are all valid. Formally, given target code rates $\rho_1, \ldots, \rho_t$ and codewords $\hat{v}_1|_L, \ldots, \hat{v}_t|_L$, the prover convinces the verifier with oracle access to these codewords that for all $i \in [t]$, $\hat{v}_i|_L$ is $\delta|L|$-close to $\mathrm{RS}[L, \rho_i]$. Here the oracle access is achieved by Merkle trees and δ is called as proximity parameter. We recall the FRI protocol to check whether $\hat{f}|_L \in \mathrm{RS}[\rho, L]$ in Fig. 10. Here the interactive repetition parameter is 1 and the query repetition parameter is ℓ.



Prover $\mathcal{P}(\mathtt{x},\mathtt{w})$

To prove $(\mathtt{x},\mathtt{w}) \in \mathcal{R}_{\mathrm{FRI}}$, where
$\mathtt{x} = (\mathbb{F}, L, \boldsymbol{\eta})$, $\mathtt{w} = \hat{f}$,
and $\deg(\hat{f}) < \rho|L| = 2^k$.

Verifier $\mathcal{V}(\mathtt{x})$

Set $L_1 \leftarrow L$, $\hat{f}|_{L_1} \leftarrow \mathrm{FFT}(\hat{f}, L_1)$
Decompose $\hat{f}_1(X) = \hat{f}(X)$ as
$\hat{f}_1(X) = \hat{Q}_1(X,Y) = \hat{f}_1^{(1)}(Y) + X \cdot \hat{f}_1^{(2)}(Y) + \ldots + X^{2^{\eta_1}-1} \cdot \hat{f}_1^{(2^{\eta_1}-1)}(Y)$
where $Y = X^{2^{\eta_1}}$, $\deg(\hat{f}_1) < 2^k$
$\mathsf{root}_1 \leftarrow \mathsf{MT.Commit}(\hat{f}_1|_{L_1})$

$\xrightarrow{\mathsf{root}_1}$

$\xleftarrow{\alpha_1}$ $\quad \alpha_1 \xleftarrow{\$} \mathbb{F}$

$\hat{f}_2(Y) \leftarrow \hat{Q}_1(\alpha_1, Y)$
Let $X \leftarrow Y$, Decompose $\hat{f}_2(X)$ as
$\hat{f}_2(X) = \hat{Q}_2(X,Y) = \hat{f}_2^{(1)}(Y) + X \cdot \hat{f}_2^{(2)}(Y) + \ldots + X^{2^{\eta_2}-1} \cdot \hat{f}_2^{(2^{\eta_2}-1)}(Y)$
where $Y = X^{2^{\eta_2}}$, $\deg(\hat{f}_2) < 2^{k-\eta_1}$
Compute $L_2 = \{a_1^{2^{\eta_1}}\}$, for all $a_1 \in L_1$
$\mathsf{root}_2 \leftarrow \mathsf{MT.Commit}(\hat{f}_2|_{L_2})$

$\xrightarrow{\mathsf{root}_2}$

$\xleftarrow{\alpha_2}$ $\quad \alpha_2 \xleftarrow{\$} \mathbb{F}$

.
.
.

$\hat{f}_{\mathsf{r}}(Y) \leftarrow \hat{Q}_{\mathsf{r}-1}(\alpha_{\mathsf{r}}, Y)$
Let $X \leftarrow Y$, Decompose $\hat{f}_{\mathsf{r}}(X)$ as
$\hat{f}_{\mathsf{r}}(X) = Q_{\mathsf{r}}(X,Y) = \hat{f}_{\mathsf{r}}^{(1)}(Y) + X \cdot \hat{f}_{\mathsf{r}}^{(2)}(Y) + \ldots + X^{2^{\eta_{\mathsf{r}}}-1} \cdot \hat{f}_{\mathsf{r}}^{(2^{\eta_{\mathsf{r}}}-1)}(Y)$
where $Y = X^{2^{\eta_{\mathsf{r}}}}$, $\deg(\hat{f}_{\mathsf{r}}) < 2^{k-\sum_{i=1}^{\mathsf{r}} \eta_i}$
Compute $L_{\mathsf{r}} = \{a_{\mathsf{r}-1}^{2^{\eta_{\mathsf{r}-1}}}\}$, for all $a_{\mathsf{r}-1} \in L_{\mathsf{r}-1}$
$\mathsf{root}_{\mathsf{r}} \leftarrow \mathsf{MT.Commit}(\hat{f}_{\mathsf{r}}|_L)$

$\xrightarrow{\mathsf{root}_{\mathsf{r}}}$

$\xleftarrow{\alpha_{\mathsf{r}}}$ $\quad \alpha_{\mathsf{r}} \xleftarrow{\$} \mathbb{F}$

$\hat{f}_{\mathsf{r}+1}(Y) \leftarrow \hat{Q}_{\mathsf{r}}(\alpha_{\mathsf{r}}, Y)$, Let $X \leftarrow Y$
Compute $L_{\mathsf{r}+1} = \{a_{\mathsf{r}}^{2^{\eta_{\mathsf{r}}}}\}$, for all $a_{\mathsf{r}} \in L_{\mathsf{r}}$

$\xrightarrow{\hat{f}_{\mathsf{r}+1}(X)}$

$\xleftarrow{q_1, \ldots, q_\ell}$ $\quad$ pick $q_1, \ldots, q_\ell \xleftarrow{\$} L_1$

For $i \in [\ell]$, compute and send

$$\hat{f}_1(q_i \cdot w^{\frac{|L_1|}{2^0}}), \hat{f}_1(q_i \cdot w^{\frac{|L_1|}{2^1}}), \ldots, \hat{f}_1(q_i \cdot w^{\frac{|L_1|}{2^{\eta_1-1}}})$$
$$\hat{f}_2(q_i^{2^{\eta_1}} \cdot w^{\frac{|L_2|}{2^0}}), \hat{f}_2(q_i^{2^{\eta_1}} \cdot w^{\frac{|L_2|}{2^1}}), \ldots, \hat{f}_2(q_i^{2^{\eta_1}} \cdot w^{\frac{|L_2|}{2^{\eta_2-1}}})$$
$$\ldots$$
$$\hat{f}_{\mathsf{r}}(q_i^{2^{\sum_{j=1}^{\mathsf{r}-1}\eta_j}} \cdot w^{\frac{|L_{\mathsf{r}}|}{2^0}}), \hat{f}_{\mathsf{r}}(q_i^{2^{\sum_{j=1}^{\mathsf{r}-1}\eta_j}} \cdot w^{\frac{|L_{\mathsf{r}}|}{2^1}}), \ldots, \hat{f}_{\mathsf{r}}(q_i^{2^{\sum_{j=1}^{\mathsf{r}-1}\eta_j}} \cdot w^{\frac{|L_{\mathsf{r}}|}{2^{\eta_{\mathsf{r}}-1}}})$$
$$\hat{f}_{\mathsf{r}+1}(q_i^{2^{\sum_{j=1}^{\mathsf{r}}\eta_j}})$$

Check correctness.

Also send verification paths in Merkle trees for the above values

$\mathsf{MT.Open}(q, \hat{f}_1|_L, \mathsf{root}_1)$,
$\mathsf{MT.Open}(q, \hat{f}_2|_L, \mathsf{root}_2)$,
$\ldots$
$\xrightarrow{\mathsf{MT.Open}(q, \hat{f}_{\mathsf{r}-1}|_L, \mathsf{root}_{\mathsf{r}-1})}$

Check consistency

Fig. 10: The procedure of FRI for testing whether $\hat{f}|_L \in \mathrm{RS}[\rho, L]$.

The main procedure of the FRI could be divided into the interactive phase and the query phase. In the i-th round of the interactive phase, the prover uses the verifier's challenge in the last round α_{i-1} to construct the current polynomial $\hat{f}_i$. The prover then commits to $\hat{f}_i|_L^{(i)}$ using Merkle trees and sends it to the verifier. The vector $\boldsymbol{\eta} = \{\eta_1, \ldots, \eta_{\mathsf{r}}\}$ is called *reduction strategy*, which describes how much the degree is reduced in every round. After r rounds, the prover directly sends the polynomial $\hat{f}_{\mathsf{r}+1}$ to the verifier.

In the query phase, the verifier randomly picks ℓ values $q_1, \ldots, q_\ell \in L$, and asks the prover to open $(2^{\eta_i} - 1) \cdot \ell$ evaluations in $\hat{f}_i|_{L_i}$. All the evaluation points are determined by $q_1, \ldots, q_\ell$. After receiving all the evaluation values, the verifier first checks the consistency between these values and the Merkle tree authentication path. The verifier then checks the correctness by judging whether the queries in $\hat{f}_i|_{L_i}$ and one query in $\hat{f}_{i+1}|_{L_{i+1}}$ are on a common polynomial with degree $2^{\eta_i} - 1$.

To explain why the correctness holds, we introduce the specific decomposition way. In each round i, $1 \leq i \leq \mathsf{r}$, the prover decomposes the current polynomial

$\hat{f}_i(X)$ with degree less than k_i as

$$\hat{f}_i(X) = f_i^{(1)}(X^{2^{\eta_i}}) + X \cdot \hat{f}_i^{(2)}(X^{2^{\eta_i}}) + \ldots + X^{2^{\eta_i}-1} \cdot \hat{f}_i^{(2^{\eta_i})}(X^{2^{\eta_i}}).$$

Note that the polynomial $\hat{f}_i(X)$ can be seen as a binary variate polynomial

$$\hat{Q}_i(X,Y) = \hat{f}_i^{(1)}(Y) + X \cdot \hat{f}_i^{(2)}(Y) + \cdots + X^{2^{\eta_i}-1} \cdot \hat{f}_i^{(2^{\eta_i})}(Y),$$

where $Y \leftarrow X^{2^{\eta_i}}$. The new evaluation domain L_i are calculated by $a_{i-1}^{2^{\eta_i}}$, for all $a_{i-1} \in L_{i-1}$. After receiving the verifer's challenge α_i, the prover sets $\hat{f}_{i+1}(Y) \leftarrow \hat{Q}_i(\alpha_i, Y)$. Let $X \leftarrow Y$, the prover obtains the polynomial $\hat{f}_{i+1}(X)$ in the $i+1$-th round.

We observe that $\hat{Q}_i(X,Y)$ is a polynomial where the degree of X is $2^{\eta_i} - 1$. This means that for any point $y \in L_i$, the pairs $(x_1, y), (x_2, y), \ldots, (x_{2^{\eta_i}}, y)$ will determine the polynomial $\hat{Q}_i(X, y)$, where $(x_1, \ldots, x_{2^{\eta_i}})$ are the 2^{η_i}-th root of y. Besides, we note that $\hat{f}_{i+1}(y) = \hat{Q}_i(\alpha_i, y)$ is also on the polynomial $\hat{Q}_i(X, y)$. This means the verifier could check the consistency of $\hat{Q}_i(\alpha_i, y)$ between the query to $\hat{f}_{i+1}(y)$ and the value computed by himself according to the queries to $\{\hat{Q}_i(x_j, y)\}_{j \in [2^{\eta_1}]}$.

Batch-FRI. For multiple polynomials $\hat{f}_1, \ldots, \hat{f}_t$ with multiple degrees $k_1, \ldots, k_t$, one can use batch-FRI to test all the polynomials have their respective degrees [4] using a random linear combination. Concretely, the verifier samples a random challenge $\lambda \overset{\$}{\leftarrow} \mathbb{F}$. The prover computes the linear combination

$$\hat{f}'(X) = \sum_{i=1}^{t} \lambda^i \cdot X^{k_{\max} - k_i} \cdot \hat{f}_i(X).$$

The prover and the verifier continue with FRI for $\hat{f}'(X)$. Note that the oracle for $\hat{f}'|_L$ can be constructed by the oracles for $\hat{f}_1|_L, \ldots, \hat{f}_t|_L$.

The proof size of batch-FRI. Let the field be $\mathbb{F}$, the reduction strategy be $\boldsymbol{\eta}$, the query repetition be ℓ. The proof size of batch-FRI is

$$t \cdot \ell \cdot \sum_{i=1}^{\mathsf{r}} 2^{\eta_i} \, |\mathbb{F}| + t \cdot \sum_{i=1}^{\mathsf{r}} \mathsf{Tree}(|L_i|, \ell \cdot 2^{\eta_i}) \, |\mathsf{H}|,$$

where $\mathsf{Tree}(m, n)$ means the length of authentication path when opening n entries from a m-length vector committed by the Merkle tree.

B.1 Soundness of batch-FRI

The soundness error of batch-FRI is closely related to the proof size of our protocol. We give two versions of soundness error. One is the provably one and the other is a conjecture, and both are from [4].

The provably one. For proximity parameter close to the Johnson bound, the soundness error of the batch-FRI is as follows according to [4]

$$\epsilon = (t - \frac{1}{2}) \cdot \frac{(j + \frac{1}{2})^7}{2 \cdot \rho^{\frac{3}{2}}} \cdot \frac{|L|^2}{|\mathbb{F}|^e} + \frac{(2j+1) \cdot (|L|+1) \cdot \sum_{i=1}^{\mathsf{r}} 2^{\eta_i}}{|\mathbb{F}|^e} + (1-\delta)^{\ell},$$

where the soundness error of the interactive phase is

$$\epsilon_I = (t - \frac{1}{2}) \cdot \frac{(j + \frac{1}{2})^7}{2 \cdot \rho^{\frac{3}{2}}} \cdot \frac{|L|^2}{|\mathbb{F}|^e} + \frac{(2j+1) \cdot (|L|+1) \cdot \sum_{i=1}^{\mathsf{r}} 2^{\eta_i}}{|\mathbb{F}|^e}, \tag{22}$$

the soundness error of the query phase is

$$\epsilon_q = (1-\delta)^{\ell},$$

and we have

$$\delta = 1 - \sqrt{\rho} \cdot (1 + \frac{1}{2j}), j \geq 3.$$

In practical applications, one can repeat the interactive phase to adjust to the target security level. For example, if setting the interactive repetition parameter as $e = 3$, the $|\mathbb{F}|$ in Equation (22) will be substituted with $|\mathbb{F}|^e$.

The conjecture one. In their line of works on FRI [3, 4, 27], the authors make several conjectures on the soundness of FRI for proximity parameters above the Johnson bound. The most recent conjecture is Conjecture 8.4 in [4]. Based on this conjecture, we could obtain the following conclusion by setting $c_1 = c_2 = 1$ in [4]. This soundness is used in both estark [40] and plonky2 [38].

Lemma 4 ([4, 40]). *The conjectured security bit λ for a batch-FRI is the minimum of three values:*

- $-\log_2 \rho \cdot \ell$*, where ρ is the code rate and ℓ is the query number.*
- $|\mathsf{H}|/2$*, where $|\mathsf{H}|$ is the output bits of the hash function in the Merkle tree.*
- $k_{\max}/|\mathbb{F}|^e$*, where $k_{\max}$ is the maximum degree of the polynomial to be tested and e means the interactive repetition parameter.*