Ligero: Lightweight Sublinear Arguments Without a Trusted Setup*

Scott Ames† Carmit Hazay‡ Yuval Ishai§ Muthuramakrishnan Venkitasubramaniam¶

Abstract

We design and implement a simple zero-knowledge argument protocol for NP whose communication complexity is proportional to the square-root of the verification circuit size. The protocol can be based on any collision-resistant hash function. Alternatively, it can be made non-interactive in the random oracle model, yielding concretely efficient zk-SNARKs that do not require a trusted setup or public-key cryptography. Our protocol is obtained by applying an optimized version of the general transformation of Ishai et al. (STOC 2007) to a variant of the protocol for secure multiparty computation of Damgård and Ishai (CRYPTO 2006). It can be viewed as a simple zero-knowledge interactive PCP based on "interleaved" Reed-Solomon codes.

This paper is an extended version of the paper published in CCS 2017 and features a tighter analysis, better implementation along with formal proofs. For large verification circuits, the Ligero prover remains competitive against subsequent works with respect to the prover's running time, where our efficiency advantages become even bigger in an amortized setting, where several instances need to be proven simultaneously.

Our protocol is attractive not only for very large verification circuits but also for moderately large circuits that arise in applications. For instance, for verifying a SHA-256 preimage with 2^{-40} soundness error, the communication complexity is roughly 35KB. The communication complexity of our protocol is independent of the circuit structure and depends only on the number of gates. For 2^{-40} soundness error, the communication becomes smaller than the circuit size for circuits containing roughly 3 million gates or more. With our refined analysis the Ligero system's proof lengths and prover's running times are better than subsequent post-quantum ZK-SNARKs for small to moderately large circuits.

*An extended abstract of this paper appeared in CCS 2017.

†University of Rochester. Email: q309185@gmail.com.

‡Bar-Ilan University. Email: carmit.hazay@cs.biu.ac.il.

§Technion and UCLA. Email: yuvali@cs.technion.ac.il

¶University of Rochester. Email: muthuv@cs.rochester.edu.

Contents

1 Introduction

Verifying outsourced computations is important for tasks and scenarios when there is an incentive for the party performing the computation to report incorrect answers. In this work, we present a concretely efficient argument protocol for NP whose communication complexity is proportional to the square root of the size of a circuit verifying the NP witness. Our argument system is in fact a zero-knowledge argument of knowledge, and it only requires the verifier to send public coins to the prover. The latter feature implies that it can be made non-interactive via the Fiat-Shamir transform [FS86], yielding an efficient implementation of zero-knowledge succinct non-interactive arguments of knowledge (zk-SNARKs [BCCT13]) without a trusted setup. To put our work in the proper context, we provide some relevant background. The last decade has seen tremendous progress in designing and implementing efficient proof system (see [WB15, BBC+17, Ish20, Tha22] for surveys). These efforts can be divided into three broad categories according to the underlying combinatorial machinery.

Doubly efficient interactive proofs: This line of work, initiated by Goldwasser, Kalai, and Rothblum [GKR15] (following a rich line of work on interactive proofs with computationally unbounded provers [GMR85, LFKN90, Sha90]), provides sublinear communication and efficiently verifiable proofs for low-depth polynomial-time computations.[1] See [CMT12, Tha13, VSBW13, RRR16] and references therein for a survey of works along this line.

Probabilistically checkable proofs (PCPs) and their interactive variants: Originating from the works of Kilian [Kil92] and Micali [Mic94], recent works [BCGT13, BCG+16, BBC+17] have shown how to obtain efficient sublinear arguments for NP from PCPs [BFLS91, AS98, ALM+98]. Classical PCPs have been extended to allow additional interaction with the prover, first in the model of interactive PCP (IPCP) [KR08] and then in the more general setting of interactive oracle proofs (IOP) [BCS16, RRR16]. Arguments obtained via PCPs and IOPs have the advantages of not relying on public-key cryptography, not requiring a trusted setup, and offering conjectured post-quantum security. However, previous works along line were still quite far from having good concrete efficiency.

Linear PCPs: This line of work, initiated by Ishai, Kushilevitz, and Ostrovsky [IKO07] (in the interactive or designated verifier setting) and by Groth [Gro10] (in the non-interactive, public verification setting of SNARKs) obtains sublinear arguments for NP *with preprocessing* by combining *linear PCPs* with homomorphic public-key cryptography. In a linear PCP the verifier can obtain a small number of linear combinations of a proof vector. Linear PCPs are simpler to construct than classical PCPs and have served as the basis for some of the first implementations of verifiable computation protocols [SMBW12]. A very efficient construction of linear PCPs for NP that served as the basis for most previous SNARK implementations, including the ones used in zerocash [BCG+14], was given by Gennaro, Gentry, Parno, and Raykova in [GGPR13]. (The view of these SNARKs as being based on linear PCPs is due to Bitansky et al. [BCI+13] and Setty et al. [SBV+13].) Two practical disadvantages of the protocols along this line are that they are quite slow on the prover side (due to a heavy use of public-key cryptography), and their soundness in the non-interactive setting crucially relies on the existence of a long and "structured" common reference string that needs to either be generated by a trusted party or by an expensive distributed protocol.

Our goal in this work is to combine the best features of previous approaches to the extent possible:

[1] The GKR technique has been extended to the case of NP statements by Zhang et al. [ZGK+17], Wahby et al. [WTS+18], and several subsequent works. However, the communication complexity of the resulting arguments still grows with the verification circuit depth, and moreover their instantiations require a polynomial commitment primitive whose efficient implementations typically involve the use of public-key cryptography.

Obtain a simple, concretely efficient, sublinear communication (public-coin) zero-knowledge argument system for NP, without any setup, complex PCP machinery, or expensive public-key operations.

As discussed above, all prior works fall short of meeting the above goal on one or more counts.

1.1 Our Results

The main result of this work is a zero-knowledge argument protocol for NP with the following features.

- It is *sublinear*, in the sense that the asymptotic communication complexity is roughly the square root of the verification circuit size.
- It is simple to describe and analyze in a self-contained way.
- It only employs symmetric-key primitives (collision-resistant hash-functions) in a black-box way.
- It is public-coin, which means the protocol can be made non-interactive in the random oracle model by using the Fiat-Shamir transform [FS86], thus providing a light-weight implementation of (publicly verifiable) zero-knowledge SNARKs.
- It does not require any trusted setup, even in the non-interactive case.
- It is concretely efficient. We demonstrate its concrete efficiency via an implementation.
- In the multi-instance setting where many instances for the same NP verification circuit are required, we obtain improved amortized communication complexity with sublinear verification time.

Our protocol can be seen as a light-weight instance of the second category of protocols discussed above. However, instead of directly applying techniques from the PCP literature, we combine efficient protocols for secure multiparty computation (MPC) with a variant of the general transformation of Ishai, Kushilevitz, Ostrovsky, and Sahai (IKOS) [IKOS07] that transforms such MPC protocols to zero-knowledge interactive PCPs (ZKIPCP). More concretely, we instantiate the MPC component with an optimized variant of the protocol of Damgård and Ishai [DI06] (similar to the one described in Appendix C of [IPS09]) and transform it into a ZKIPCP by applying a more efficient variant of the IKOS transformation in the spirit of the IPS compiler [IPS08]. In a nutshell, the main difference with respect to the original IKOS transformation is that we restrict the topology of the MPC network in a way that leads to a better trade-off between soundness error and communication complexity.

A key feature of the underlying MPC protocol is that its *total* communication complexity between the parties is independent of the number of parties and is roughly equal to the size of the circuit being evaluated. Now, letting the number of parties be the square root of the circuit size, results in communication per party that is also roughly the square root of the circuit size. This translates into a ZKIPCP with similar parameters. See Section 4 for a self-contained presentation of the ZKIPCP obtained via the above approach.

The work of Giacomelli, Madsen and Orlandi [GMO16] and its improvement due to Chase et al. [CDG+17] already demonstrated that the IKOS transformation can lead to concretely efficient zero-knowledge arguments, but where the communication is bigger than the verification circuit size. In the present work, we obtain a sublinear variant of this result by modifying both the IKOS transformation and the underlying MPC machinery. To summarize, using the above approach we obtain a simple proof of the following theorem with good concrete efficiency:

Theorem 1.1 (Informal)**.** *Assume the existence of collision-resistant hash-functions. Then there is a public-coin zero-knowledge argument for proving the satisfiability of a circuit* C *with communication complexity* $\tilde{O}(\sqrt{|\mathrm{C}|})$.

Concrete efficiency. We now give more detailed information about the concrete efficiency of our implementation. The following numbers apply either to interactive zero-knowledge protocols based on collision-resistant hash functions or to non-interactive zk-SNARKs in the random oracle model obtained via the Fiat-Shamir transform. We refer the reader to Section 6 for more details and give only a few representative figures below. The communication complexity of proving the satisfiability of an arithmetic circuit C is given by $\kappa \cdot |\mathbb{F}| \cdot \sqrt{\frac{5 \cdot |\mathrm{C}|}{\log_2(3/2) \cdot \min(|\mathbb{F}|, \kappa)}}$ which simplifies to $O(|\mathbb{F}| \cdot \sqrt{|\mathrm{C}| \cdot \kappa})$ bits for large fields and $O(\kappa \cdot \sqrt{|\mathrm{C}| \cdot |\mathbb{F}|})$ bits for small fields (i.e. for $|\mathbb{F}| < \kappa$).

The communication complexity becomes smaller than the circuit size for circuits over 30-bit primes with more than 2.6 million gates at 40-bit security and around 20 million gates for 128-bit security. One concrete benchmark that has been used in prior works is verifying a SHA-256 preimage in zero-knowledge. For this benchmark, the communication complexity of our protocol with 2^{-40} soundness error is less than 35KB. In Section 6, we compare with other ZK-SNARKs with plausible post-quantum security. Our system produces better proof lengths for small to moderately large circuits in comparison to all relevant previous works and has better prover efficiency.

Our protocol easily extends to a multi-instance setting to provide additional benefits. In this setting, we can handle N instances of a circuit of size s with soundness error $2^{-\kappa}$ at an amortized communication cost per instance smaller than s when $N = \Omega(\kappa^2)$. Moreover, the amortized verification time in the multi-instance setting is sublinear, involving a total of $O(s \log s + N \log N)$ field operations. Finally, the prover's running time grows linearly with the number of instances but still remains practically feasible for reasonable number of instances.

Related work. In a concurrent and independent work [BBHR19], Ben-Sasson et al. use different techniques to construct concretely efficient IOPs that imply "transparent" proof systems, referred to as zk-STARKs, of the same type we obtain here. These zk-STARK constructions significantly improve over the previous ones from [BBC+17]. A preliminary comparison with the concrete efficiency of our construction suggests that our construction is generally more attractive in terms of prover computation time and also in terms of proof size for smaller circuits (say, of size comparable to a few SHA-256 circuits), whereas the construction from [BBHR19] is more attractive in terms of verifier computation time and proof size for larger circuits.

Subsequent work. The area of zero-knowledge proof systems had been very active in the past five years. Below we review the relevant literature published subsequent to [AHIV17].

In the design of ZKSNARKs based on symmetric-key primitives using IOPs, Aurora [BCR+19] is a transparent zk-SNARKs for R1CS (Rank-1 Constraint Satisfaction) with polylogarithmic proof size. Additional [GKR08]-based zero-knowledge argument schemes were proposed in subsequent works [XZZ+19a, ZXZS20]. The former work introduces Libra, a linear time prover for special structures circuits, whereas the later work designs a new polynomial commitment scheme based on symmetric cryptographic primitives without trusted setup. A more recent extension of [ZXZS20] for general circuits and similar features are shown in [ZLW+21a] where the they design a GKR-style proof system with linear-time prover. Golovnev et al. [GLS+21] provided an alternative mechanism to obtain a linear time prover for arbitrary fields which admits competitive concrete efficiency.

In the context of MPC-in-the-head, [KKW18, dSGOT21] have optimized the communication cost size of post-quantum signature schemes based on preprocessing-based dishonest majority MPC and multi-round

IOPs, respectively. However, these schemes scale linearly with circuit size. [GSV21] further considered a tailored version of Ligero in the Boolean setting and introduced several improvements to the proof size. In [BFH+20], Bhadauria et al. designed an optimized sublinear IOP by combining Ligero and Aurora [BCR+19], achieving tradeoffs between the prover complexity and proof length. Another sequence of works [dSGMOS19, BN20, HKL22] extends [KKW18] to arithmetic circuits over field and ring operations, introducing further improvements of signature sizes.

Another line of sublinear ZK systems is based on the hardness of discrete logarithm hardness assumption e.g., [WTS+18, BBB+18] that imply higher running times due to a number of asymmetric operations that grows with the circuit size. A new class of zk-SNARKs for R1CS under this assumption and improved tradeoffs between the prover's overhead, the proof length and the sublinear verification cost is introduced in [Set20]. These proofs cannot attain post-quantum security.

Additional subsequent work in discussed in Section 7.

Comparison with [AHIV17]. This work is the extended version of the paper published in CCS'17 [AHIV17]. Specifically, this version contains complete proofs of the construction including tightening of the soundness analysis and an analysis of the Fiat-Shamir transform. We have included a section about subsequent work (specific to IOPs) and open problems and revised implementation section with optimizations and some comparison with relevant work.

2 Preliminaries

Basic notations. We denote the security parameter by κ. We say that a function $\mu : \mathbb{N} \to \mathbb{N}$ is *negligible* if for every positive polynomial $p(\cdot)$ and all sufficiently large κ's it holds that $\mu(\kappa) < \frac{1}{p(\kappa)}$. We use the abbreviation PPT to denote probabilistic polynomial-time and denote by $[n]$ the set of elements $\{1, \ldots, n\}$ for some $n \in \mathbb{N}$, and by $[a]_i$ the i^{th} element a_i from the set a. For an NP relation $\mathcal{R}$, we denote by $\mathcal{R}_x$ the set of witnesses of x and by $\mathcal{L}_{\mathcal{R}}$ its associated language. That is, $\mathcal{R}_x = \{w \mid (x, w) \in \mathcal{R}\}$ and $\mathcal{L}_{\mathcal{R}} = \{x \mid \exists\, w \; s.t. \; (x, w) \in \mathcal{R}\}$.

We assume functions to be represented by a Boolean/arithmetic circuit C (with AND/MULTIPLY, XOR/ADD gates of fan-in 2 and NOT gates), and denote the size of C by $|\mathrm{C}|$. By default we define the size to include the total number of gates, excluding NOT gates but including input gates. We specify next the definitions of computationally indistinguishable distributions and statistical distance.

2.1 Collision-Resistant Hashing and Merkle Trees

Let $\{\mathcal{H}_\kappa\}_{\kappa \in \mathbb{N}} = \{H : \{0,1\}^{p(\kappa)} \to \{0,1\}^{p'(\kappa)}\}_\kappa$ be a family of hash functions, where $p(\cdot)$ and $p'(\cdot)$ are polynomials so that $p'(\kappa) \leq p(\kappa)$ for sufficiently large $\kappa \in \mathbb{N}$. For a hash function $H \leftarrow \mathcal{H}_\kappa$ a Merkle hash tree [Mer89] is a data structure that allows to commit to $\ell = 2^d$ messages by a single hash value h such that revealing any message requires only to reveal $O(d)$ hash values.

A Merkle hash tree is represented by a binary tree of depth d where the ℓ messages $m_1, \ldots, m_\ell$ are assigned to the leaves of the tree; the values assigned to the internal nodes are computed using the underlying hash function H that is applied on the values assigned to the children, whereas the value h that commits to $m_1, \ldots, m_\ell$ is assigned to the root of the tree. To open the commitment to a message m_i, one reveals m_i together with all the values assigned to nodes on the path from the root to m_i, and the values assigned to the siblings of these nodes. We denote the algorithm of committing to ℓ messages $m_1, \ldots, m_\ell$ by $h := \mathsf{Commit}_{\mathsf{M}}(m_1, \ldots, m_\ell)$ and the opening of m_i by $(m_i, \mathrm{path}(i)) := \mathsf{Open}_{\mathsf{M}}(h, i)$. Verifying the opening

of m_i is carried out by essentially recomputing the entire path bottom-up and comparing the final outcome (i.e., the root) to the value given at the commitment phase.

The binding property of a Merkle hash tree is due to collision-resistance. Intuitively, this says that it is infeasible to efficiently find a pair (x, x') so that $H(x) = H(x')$, where $H \leftarrow \mathcal{H}_\kappa$ for sufficiently large κ. In fact, one can show that collision-resistance of $\{\mathcal{H}_\kappa\}_{\kappa\in\mathbb{N}}$ carries over to the Merkle hashing. Formally, we say that a family of hash functions $\{\mathcal{H}_\kappa\}_\kappa$ is collision-resistant if for any PPT adversary $\mathcal{A}$ the following experiment outputs 1 with probability $\mathsf{negl}(\kappa)$: (i) A hash function H is sampled from $\mathcal{H}_\kappa$; (ii) The adversary $\mathcal{A}$ is given H and outputs x, x'; (iii) The experiment outputs 1 if and only if $x \neq x'$ and $H(x) = H(x')$.

In the random oracle model, Merkle tree can be computed by replacing the function H with a random oracle ρ where statistical binding follows due to the hardness of finding a collision in this model. We denote this algorithm by $\mathsf{Commit}^{\mathrm{RO}}_{\mathsf{M}}$.

2.2 Zero-Knowledge Arguments

We denote by $\langle A(w), B(z)\rangle(x)$ the random variable representing the (local) output of machine B when interacting with machine A on common input x, when the random-input to each machine is uniformly and independently chosen, and A (resp., B) has auxiliary input w (resp., z).

Definition 2.1 (Interactive argument system)**.** *A pair of* PPT *interactive machines $\langle \mathcal{P}, \mathcal{V}\rangle$ is called an* interactive proof system for a language $\mathcal{L}$ *if there exists a negligible function* negl *such that the following two conditions hold:*

1. COMPLETENESS: *For every $x \in \mathcal{L}$ there exists a string w such that for every $z \in \{0,1\}^*$,*

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

2. SOUNDNESS: *For every $x \notin \mathcal{L}$, every interactive* PPT *machine $\mathcal{P}^*$, and every $w, z \in \{0,1\}^*$*

$$\Pr[\langle \mathcal{P}^*(w), \mathcal{V}(z)\rangle(x) = 1] \leq \mathsf{negl}(|x|).$$

Definition 2.2 (Zero-knowledge)**.** *Let $\langle \mathcal{P}, \mathcal{V}\rangle$ be an interactive proof system for some language $\mathcal{L}$. We say that $\langle \mathcal{P}, \mathcal{V}\rangle$ is* computational zero-knowledge with respect to an auxiliary input *if for every* PPT *interactive machine $\mathcal{V}^*$ there exists a* PPT *algorithm $\mathcal{S}$, running in time polynomial in the length of its first input, such that*

$$\{\langle \mathcal{P}(w), \mathcal{V}^*(z)\rangle(x)\}_{x\in\mathcal{L}, w\in\mathcal{R}_x, z\in\{0,1\}^*} \overset{\mathrm{c}}{\approx} \{\langle \mathcal{S}\rangle(x, z)\}_{x\in\mathcal{L}, z\in\{0,1\}^*}$$

(when the distinguishing gap is considered as a function of $|x|$). Specifically, the left term denote the output of $\mathcal{V}^*$ after it interacts with $\mathcal{P}$ on common input x whereas, the right term denote the output of $\mathcal{S}$ on x.

2.3 Interactive Oracle Proofs

Interactive Oracle Proofs (IOP) [BCS16, RRR16] is a type of proof system that combines the aspects of Interactive Proofs (IP) [Bab85, GMR85] along with Probabilistic Checkable Proofs (PCP) [BFLS91, AS98, ALM+98] as well generalizes Interactive PCPs (IPCP) [KR08]. In this model, like the PCP model, the verifier does not need to read the whole proof and instead can query the proof at some random locations while similarly to the IP model, the prover and verifier interact over several rounds.

A k-round IOP has k rounds of interaction. In the i^{th} round of interaction, the verifier sends a uniform public message m_i to the prover and the prover generates π_i. After running k rounds of interaction, the verifier makes some queries to the proofs via oracle access and will either accept it or reject it.

Definition 2.3. *Let $\mathcal{R}(x, \omega)$ be an NP relation corresponding to an NP language $\mathcal{L}$. An IOP system for a relation $\mathcal{R}$ with round complexity k and soundness ϵ is a pair of PPT algorithms $(\mathcal{P}, \mathcal{V})$ if it satisfies the following properties:*

- SYNTAX: *On common input x and prover input ω, $\mathcal{P}$ and $\mathcal{V}$ run an interactive protocol of k rounds. In each round i, $\mathcal{V}$ sends a message m_i and $\mathcal{P}$ generates π_i. Here the verifier has oracle access to $\{\pi_1, \pi_2, \ldots, \pi_k\}$. We can express $\pi = (\pi_1, \pi_2, \ldots, \pi_k)$. Based on the queries from these oracles, $\mathcal{V}$ accepts or rejects.*
- COMPLETENESS: *If $(x, \omega) \in \mathcal{R}$ then,*

$$\Pr[(\mathcal{P}(x, \omega), \mathcal{V}^{\pi}(x)) = 1] = 1$$

- SOUNDNESS: *For every $x \notin \mathcal{L}$, every unbounded algorithm $\mathcal{P}^*$ and proof $\tilde{\pi}$*

$$\Pr[(\mathcal{P}^*, \mathcal{V}^{\tilde{\pi}}) = 1] \leq \mathsf{negl}(\lambda)$$

The notion of IOP can be extended to provide zero-knowledge property as well. Next we define the definition of zero-knowledge IOP.

Definition 2.4. *Let $\langle \mathcal{P}, \mathcal{V} \rangle$ be an IOP for $\mathcal{R}$. We say that $\langle \mathcal{P}, \mathcal{V} \rangle$ is a (honest verifier) zero-knowledge IOP (or ZKIOP for short) if there exists a PPT simulator $\mathcal{S}$, such that for any $(x, \omega) \in \mathcal{R}$, the output of $\mathcal{S}(x)$ is distributed identically to the view of $\mathcal{V}$ in the interaction $(\mathcal{P}(x, \omega), \mathcal{V}^{\pi}(x))$.*

2.4 Interactive PCPs

An interactive PCP [KR08] (IPCP) is a special case of IOPs (also known as probabilistically checkable interactive proofs [RRR16]) in an IOP where the prover sends a proof oracle only in the first round and in the subsequent rounds it simply responds to the verifier's message with a message instead of an oracle (a message can be viewed as an oracle with one value). We formally define the notion of IPCP below as the main construction in this work is an IPCP.

Definition 2.5 (Interactive PCP)**.** *Let $\mathcal{R}(x, w)$ be an NP relation corresponding to an NP language $\mathcal{L}$. An* interactive PCP (IPCP) *system for $\mathcal{R}$ with parameters (q, l, ϵ) is a pair of* PPT *interactive machines $\langle \mathcal{P}, \mathcal{V} \rangle$ with the following properties.*

1. *Syntax: On common input x and prover input w, the prover $\mathcal{P}$ computes in time $\mathsf{poly}(|x|)$ a bit string π (referred to as the PCP). The prover $\mathcal{P}$ and verifier $\mathcal{V}$ then interact, where the verifier has oracle access to π.*
2. COMPLETENESS: *If $(x, w) \in \mathcal{R}$ then*

$$\Pr[(\mathcal{P}(x, w), \mathcal{V}^{\pi}(x)) = 1] = 1.$$

3. SOUNDNESS: *For every $x \notin \mathcal{L}$, every (unbounded) interactive machine $\mathcal{P}^*$ and every $\tilde{\pi} \in \{0, 1\}^*$,*

$$Pr[(\mathcal{P}^*, \mathcal{V}^{\tilde{\pi}}(x)) = 1] \leq \epsilon(|x|).$$

4. COMPLEXITY: *In the interaction $(\mathcal{P}(x, w), \mathcal{V}^{\pi}(x))$ at most $l(|x|)$ bits are communicated and $\mathcal{V}$ reads at most $q(|x|)$ bits of π.*

A public-coin *IPCP is one where every message sent by the verifier simply consists of fresh random bits.*

Our zero-knowledge variants of IPCP achieve perfect zero-knowledge against an honest verifier.

Definition 2.6 (Zero-knowledge IPCP)**.** *Let $\langle \mathcal{P}, \mathcal{V} \rangle$ be an interactive PCP for $\mathcal{R}$. We say that $\langle \mathcal{P}, \mathcal{V} \rangle$ is an* (honest verifier, perfect) zero-knowledge IPCP *(or ZKIPCP for short) if there exists an expected polynomial time algorithm $\mathcal{S}$, such that for any $(x, w) \in \mathcal{R}$, the output of $S(x)$ is distributed identically to the view of $\mathcal{V}$ in the interaction $(\mathcal{P}(x, w), \mathcal{V}^{\pi}(x))$.*

3 From MPC to ZKIPCP

3.1 Our MPC Model

As mentioned in the introduction, the efficiency of our constructions can be distilled to identifying the right MPC model and designing an efficient protocol in this model. In this regards we deviate from the original work of [IKOS07] which provided a general transformation from any honest majority MPC protocol that can compute arbitrary functionalities. In particular, our model is more in line with the watchlist mechanism (a-la [IPS08]). We begin with the description of the MPC model and the protocol specifications that we will need to design our zero-knowledge protocol. In Section 4, we use such MPC protocols based on the works [DI06, CC06, IPS08, IPS09].

In our model, we consider a *sender client* S, n servers $s_1, \ldots, s_n$ and a *receiver client* R. The sender has an input x and a witness w with respect to some NP relation $\mathcal{R}$. The receiver and the servers do not receive any input, where the servers obtain random shares from the sender and evaluate the computed circuit. Upon receiving (x, w) from the sender, the functionality computes $\mathcal{R}(x, w)$ and forwards the result to the receiver R. We consider the specific network where the communication is restricted to a single message between S and the servers at the beginning of the protocol and a single message from the servers to the receiver R at the end of the protocol. Moreover, the only way the servers may communicate with each other is via a broadcast. In our actual MPC protocol, the servers will never utilize such a broadcast. Nevertheless, our transformation from MPC to ZK can be easily extended to allow for the servers to invoke a broadcast. For simplicity, we will restrict the servers to not communicate with each other at all in our actual transformation.

We consider the security of our underlying protocols in both the honest-but-curious (passive) and the malicious (active) models. In the former model, one may break the security requirements into the following correctness and privacy requirements.

Definition 3.1 (Correctness)**.** *We say that Π realizes a deterministic $n+1$-party functionality $(x, r_1, \ldots, r_n)$ with perfect (resp., statistical) correctness if for all inputs $(x, r_1, \ldots, r_n)$, the probability that the output of some player is different from the output of f is 0 (resp., negligible in κ), where the probability is over the independent choices of the random inputs $r_1, \ldots, r_n$.*

Definition 3.2 (t_p-Privacy)**.** *Let $1 \leq t_p < n$. We say that Π realizes f with perfect t_p-privacy if there is a* PPT *simulator $\mathcal{S}$ such that for any inputs $(x, r_1, \ldots, r_n)$ and every set of corrupted players $T \subset [n]$, where $|T| \leq t_p$, the joint view $\mathbf{View}_T(x, r_1, \ldots, r_n)$ of players in T is distributed identically to $\mathcal{S}(T, x, \{r_i\}_{i \in T}, f_T(x, r_1, \ldots, r_n))$.*

With respect to our MPC model defined above, we consider privacy in the presence of a static passive adversary that corrupts the receiver R and at most t_p servers. Our zero-knowledge property will reduce to this security guarantee.

In the malicious model, in which corrupted players may behave arbitrarily, security cannot be generally broken into correctness and privacy as above. However, for our purposes we only need the protocols to satisfy a weaker notion of security in the malicious model that is implied by the standard general definition. Specifically, it suffices that Π be t_p-private as above, and moreover it should satisfy the following notion of correctness in the malicious model for which we reduce the soundness property to.

Definition 3.3 (Statistical t_r-Robustness)**.** *We say that* Π *realizes* f *with statistical* t_r*-robustness if it is perfectly correct in the presence of a honest-but-curious adversary as in Definition 3.1, and furthermore for any (unbounded) active adversary that* adaptively *corrupts a set* T *of at most* t_r *players, and for any inputs* $(x, r_1, \ldots, r_n)$*, the following robustness property holds. If there is no* $(r_1, \ldots, r_n)$ *such that* $f(x, r_1, \ldots, r_n) = 1$*, then the probability that* R *outputs* 1 *in an execution of* Π *in which the inputs of the honest players are consistent with* $(x, r_1, \ldots, r_n)$ *is negligible in* κ *where* κ *is a statistical parameter that the protocol* Π *receives as input.*

Our main theorems about our two-party ZK protocol are proven in the presence of a static active adversary, that corrupts the prover at the onset of the execution. Nevertheless, our proof relies on the security of the underlying MPC protocol (utilized in the MPC-in-the-head paradigm) being robust against an active adversary that *adaptively* corrupts a subset of the servers in the underlying MPC protocol. Concretely, with respect to our MPC model defined above, we consider robustness in the presence of an adaptive active adversary that corrupts the sender S and at most t_r servers.

Finally, when used in the MPC-in-the-head paradigm, we need the notion of consistent views between servers and the receiver that we define below.

Definition 3.4 (Consistent views)**.** *We say that a pair of views* V_i, V_j *are consistent (with respect to the protocol* Π *and some public input* x*) if the outgoing messages implicit in* V_i *are identical to the incoming messages reported in* V_j *and vice versa.*

3.2 ZKIPCP for NP - The General Case

Next, we provide our compilation from an MPC protocol satisfying the requirements specified in Section 3.1 to an interactive PCP. We note that while the transformation presented in this section works for any MPC in the model as described in the previous section, we will simplify our MPC model as follows:

Two-phase: The protocol we consider will proceed in two phases: In Phase 1, the servers receive inputs from the sender and only perform local computation. After Phase 1, the servers obtain a public random string r of length l sampled via a coin-flipping oracle and broadcast to all servers. The servers use this in Phase 2 for their local computation at the end of which each server sends a single output message to the receiver R.

No broadcast: The servers never communicate with each other. Each server simply receives inputs from the sender at the beginning of Phase 1, then receives a public random string in Phase 2, and finally delivers a message to R.

Formally, let $\mathcal{L}$ be an NP language with NP relation $\mathcal{R}$, let x an NP statement that is the common input and let w be the private input of the prover. We will now design a ZKIPCP protocol Π_{ZKIPCP} (Figure 1) that meets Definition 2.5 based on any MPC protocol Π that is defined according to our model described above.

We are now ready to prove the following theorem.

- **Input:** The prover $\mathcal{P}$ and the verifier $\mathcal{V}$ share a common input statement x and a circuit description C that realizes $\mathcal{R}$. $\mathcal{P}$ additionally has input w such that $\mathcal{R}(x, w) = 1$.
- **Oracle π:** The prover runs the MPC protocol Π "in-its-head" as follows. It picks a random input r_S and invokes S on $(x, w; r_S)$ and a random input r_i for every server s_i. The prover computes the views of the servers up to the end of Phase 1 in Π, denoted by $(V_1, \ldots, V_n)$, and sets the oracle as the n symbols $(V_1, \ldots, V_n)$.
- **The interactive protocol.**
  1. $\mathcal{V}$ picks a random challenge r of length l and sends it to the sender.
  2. Upon receiving the challenge r, prover $\mathcal{P}$ sends the view V of R. (As the prover possesses all the information about the servers, and the verifier always receives the broadcast message from each server, these broadcast messages can be sent directly from the prover to the verifier.)
  3. $\mathcal{V}$ computes the output of R from the view and checks if R does not abort. It then picks a random subset Q of $[n]$ of size t_p uniformly at random (with repetitions), and queries the oracle on Q.
  4. $\mathcal{V}$ obtains from the oracle the views of the servers in Q.
  5. $\mathcal{V}$ rejects if the views of the servers are *inconsistent* with the view of R. Otherwise, it accepts.

Figure 1: Protocol Π_{ZKIPCP}

Theorem 3.5. *Let f be the following functionality for a sender S and n servers $s_1, \ldots, s_n$ and receiver R. Given a public statement x and an additional input w received from S, the functionality delivers $\mathcal{R}(x, w)$ to R. Suppose that Π is a two-phase protocol in the MPC model specified in Section 3.1 that realizes f with statistical t_r-robustness (in the malicious model) and perfect t_p-privacy (in the honest-but-curious model), where $t_r < \lceil \frac{n}{2} \rceil - 1$.[2] Then protocol Π_{ZKIPCP} described above is a ZKIPCP for NP relation $\mathcal{R}$, with soundness error $\left(1 - \frac{t_r}{n}\right)^{t_p} + \delta(\kappa)$ where $\delta(\kappa)$ is the robustness error of Π.*

Proof: Our proof follows by establishing completeness, soundness and zero-knowledge as required in Definitions 2.5-2.6.

Completeness: Completeness follows directly from the correctness of the underlying MPC protocol.

Soundness: Consider a statement $x \notin \mathcal{L}_{\mathcal{R}}$. We will show that no prover $\mathcal{P}^*$ can convince $\mathcal{V}$ beyond a negligible probability to accept a false statement. We will argue soundness by following an approach similar to [IKOS07] where we first identify an inconsistency graph and then invoke the properties of the underlying MPC. More precisely, we consider an inconsistency graph G based on the n views $V_1, \ldots, V_n$ and the view of the receiver R which contains the messages from servers $s_1, \ldots, s_n$ to R. Here, the servers and the receiver correspond to nodes in G and inconsistency between every pair of nodes is defined as in Definition 3.4. Then there are two cases depending on the graph G:

Case 1: There are more than t_r edges in G. In this case, we will argue that with high probability the set of servers opened by the verifier will hit one of these edges. Recall that the view of R is provided to the verifier. Therefore, for any edge in G between R and V_i, if the corresponding server s_i falls in Q, then the verifier rejects. The probability that all t_p servers chosen by the verifier misses all inconsistent edges is at most $\left(1 - \frac{t_r}{n}\right)^{t_p}$.

[2] The size of t_p is typically $O(\kappa)$ and will be adjusted below in order to minimize the communication complexity.

Case 2: There are fewer than t_r edges in G. In this case, we will argue that by the statistical t_r-robustness of the underlying MPC protocol Π, the verifier will reject except with probability $\delta(\kappa)$. More precisely, for every cheating strategy $\mathcal{P}^*$ in the ZK proof we will demonstrate an adversarial strategy $\mathcal{A}$ attacking the underlying MPC protocol such that the probability with which $\mathcal{V}$ accepts a false statement when interacting with $\mathcal{P}^*$ on a false statement will be bounded by the probability that R outputs 1 in an execution of the underlying MPC protocol with adversary $\mathcal{A}$.

More precisely, consider an adversary $\mathcal{A}$ that is participating in the MPC protocol with n servers, a sender and a receiver. Internally, $\mathcal{A}$ incorporates the code of $\mathcal{P}^*$ while emulating the roles of the oracle and $\mathcal{V}$. When the protocol begins, $\mathcal{P}^*$ sets the oracle with the views of the servers as in Phase 1 of Π. These views simply contain the inputs sent to the servers (as all computations are local). Upon obtaining the views of the servers, $\mathcal{A}$ will corrupt the sender in the external MPC execution, and acting as the sender, it will send as input to server s_i the value that was internally generated by $\mathcal{P}^*$ as the view of that server, namely V_i. Next, recall that in the MPC protocol the servers receive a random string from the coin-flipping oracle (in our protocol the verifier picks r as the challenge in Step 1). $\mathcal{A}$ will internally forward this string r to $\mathcal{P}^*$ as the message provided by the verifier.

Next, $\mathcal{A}$ proceeds with the internal execution by selecting t_p indices for the verifier's challenge, for which the oracle will reveal the views of these corresponding servers. If $\mathcal{V}$ rejects in the internal execution because any of these views are inconsistent, then $\mathcal{A}$ aborts. Otherwise, $\mathcal{A}$ continues with the external execution. Recall that in Phase 2, each server sends a single message to R. Then just before the servers send these messages, $\mathcal{A}$ computes the inconsistency graph G. Recall that an edge is present between a server s_i and the receiver R in this graph if the view of s_i is inconsistent with the view of R and randomness r. Let T be the set of servers of size t^* that are connected to an edge in G. If $t^* > t_r$, then $\mathcal{A}$ aborts. Otherwise, $\mathcal{A}$ (adaptively) corrupts the servers in T and replaces their (honestly generated) messages sent to R by what was internally reported in the view of R, namely the messages sent by $\mathcal{P}^*$ to the verifier in the proof.

It follows from this description that the acceptance condition of the verifier in the internal emulation with $\mathcal{A}$ is identical to the output of R in the external MPC execution. Since the underlying MPC protocol is t_r-robust and the number of parties corrupted by $\mathcal{A}$ is bounded by t_r, we have that R outputs 0 except with probability $\delta(\kappa)$. We conclude that the verifier in the internal emulation by $\mathcal{P}^*$ accepts the proof of a false statement except with probability at most $\delta(\kappa)$. Next, we observe that the view of the verifier emulated by $\mathcal{A}$ in the internal emulation is identically distributed to the view of an honest verifier in an interactive with $\mathcal{P}^*$. Therefore, we can conclude that an honest verifier accepts a false statement with probability at most $\delta(\kappa)$.

Applying a union bound, we conclude that the verifier accepts a false statement with probability at most $\left(1 - \frac{t_r}{n}\right)^{t_p} + \delta(\kappa)$.

Zero-knowledge: The zero-knowledge property follows from the t_p-privacy of the underlying MPC protocol Π. Namely, we construct a simulator $\mathcal{S}$ that invokes the simulator for the MPC protocol, denoted by $\mathcal{S}_\Pi$. $\mathcal{S}_\Pi$ simulates an adversary $\mathcal{A}$ that statically corrupts the receiver R and adaptively corrupts the t_p servers whom their views are opened for checking consistency, where the servers corruptions take place at the end of the computation. In this simulation, $\mathcal{S}_\Pi$ is required to produce the view of R upon receiving a challenge r. Next, upon obtaining the query Q from the verifier, $\mathcal{S}$ instructs $\mathcal{S}_\Pi$ to output the views of these t_p servers. □

Communication complexity: The main source of complexity is in revealing the view of R in the third message and revealing the view of the t_p servers in the last message. If the maximum size of the view of each server s_i for $i \in t_p$, is v_{size}, and the size of the view of R is v_R, then the total communication complexity from the prover is $t_p \cdot v_{\mathsf{size}} + v_R$. In Section 4 we adjust the parameters of our protocol subject to the constraint that $v_{\mathsf{size}} \cdot v_R = O(|\mathrm{C}|)$. To minimize the communication complexity, if we set $t_p \cdot v_{\mathsf{size}}$ and

v_R to be roughly equal then we obtain the optimum complexity of our approach.

4 A Direct ZKIPCP Construction

In this section we give a self-contained description of our zero-knowledge interactive PCP protocol. This protocol is a slightly optimized version of the protocol obtained by applying our variant of the general "MPC to ZK" transformation from [IKOS09] (see Section 3) to the honest-majority MPC protocol from [DI06].

Coding notation. For a code $C \subseteq \Sigma^n$ and vector $v \in \Sigma^n$, denote by $d(v, C)$ the minimal distance of v from C, namely the number of positions in which v differs from the closest codeword in C, and by $\Delta(v, C)$ the set of positions in which v differs from such a closest codeword (in case of ties, take the lexicographically first closest codeword), and by $\Delta(V, C) = \bigcup_{v \in V}\{\Delta(v, C)\}$. We further denote by $d(V, C)$ the minimal distance between a vector set V and a code C, namely $d(V, C) = \min_{v \in V}\{d(v, C)\}$. Our ZKIPCP protocol uses Reed-Solomon (RS) codes, defined next.

Definition 4.1 (Reed-Solomon Code)**.** *For positive integers n, k, finite field $\mathbb{F}$, and a vector $\eta = (\eta_1, \ldots, \eta_n) \in \mathbb{F}^n$ of distinct field elements, the code $\mathsf{RS}_{\mathbb{F},n,k,\eta}$ is the $[n, k, n-k+1]$ linear code over $\mathbb{F}$ that consists of all n-tuples $(p(\eta_1), \ldots, p(\eta_n))$ where p is a polynomial of degree $< k$ over $\mathbb{F}$.*

Definition 4.2 (Encoded message)**.** *Let $L = \mathsf{RS}_{\mathbb{F},n,k,\eta}$ be an RS code and $\zeta = (\zeta_1, \ldots, \zeta_\ell)$ be a sequence of distinct elements of $\mathbb{F}$ for $\ell \leq k$. For $u \in L$ we define the message $\mathsf{Decode}_\zeta(u)$ to be $(p_u(\zeta_1), \ldots, p_u(\zeta_\ell))$, where p_u is the polynomial (of degree $< k$) corresponding to u. For $U \in L^m$ with rows $u^1, \ldots, u^m \in L$, we let $\mathsf{Decode}_\zeta(U)$ be the length-$m\ell$ vector $x = (x_{11}, \ldots, x_{1\ell}, \ldots, x_{m1}, \ldots, x_{m\ell})$ such that $(x_{i1}, \ldots, x_{i\ell}) = \mathsf{Decode}_\zeta(u^i)$ for $i \in [m]$. Finally, when ζ is clear from the context, we say that U* encodes *x if $x = \mathsf{Decode}_\zeta(U)$.*

At a very high level, our ZKIPCP protocol proves the satisfiability of an arithmetic circuit C of size s in the following way. The prover arranges (a slightly redundant representation of) the s wire values of C on a satisfying assignment in an $O(\sqrt{s}) \times O(\sqrt{s})$ matrix, and encodes each row of this matrix using an RS code. The verifier challenges the prover to reveal linear combinations of the entries of the codeword matrix, and checks their consistency with t randomly selected columns of this matrix, where t is a security parameter. In the following we describe the ZKIPCP construction in a bottom-up fashion, first addressing the case of IPCP (with no zero-knowledge) and then introduce the modifications required for making it zero-knowledge.

For convenience, we provide a list of our parameters in Table 1.

4.1 Testing Interleaved Linear Codes

We start by describing and analyzing a simple interactive prover-assisted protocol for simultaneously testing the membership of multiple vectors in a given linear code L. It will be convenient to view m-tuples of codewords in L as codewords in an interleaved code L^m. We formally define this notion below.

Definition 4.3 (Interleaved code)**.** *Let $L \subset \mathbb{F}^n$ be an $[n, k, d]$ linear code over $\mathbb{F}$. We let L^m denote the $[n, mk, d]$ (interleaved) code over $\mathbb{F}^m$ whose codewords are all $m \times n$ matrices U such that every row U_i of U satisfies $U_i \in L$. For $U \in L^m$ and $j \in [n]$, we denote by $U[j]$ the jth symbol (column) of U.*

To test the membership of U in L^m, $\mathcal{V}$ challenges $\mathcal{P}$ to reveal a random linear combination of the rows U_i, and then checks that the revealed codeword is consistent with a randomly selected set of t columns of U.[3] The complete test is described in Figure 2.

[3] This test is implicitly used in the verifiable secret sharing sub-protocol of efficient MPC protocols from the literature, and in

Parameter	Description
w	Extended witness
U	Encoded extended witness
m	#Rows in the extended witness
ℓ	#Columns in the extended witness
s	Circuit size
k	Message length
n	Codeword length
d	Codeword distance
e	#Errors within a codeword
t	#queries on U
κ	Security parameter
σ	Repetition parameter
h	RO (Hash function) output length

Table 1: Description of our parameters.

Oracle: A purported L^m-codeword U. Depending on the context, we may view U either as a matrix in $\mathbb{F}^{m\times n}$ in which each row U_i is a purported L-codeword, or as a sequence of n symbols $(U[1],\ldots,U[n])$, $U[j]\in\mathbb{F}^m$.

Interactive testing:

1. $\mathcal{V}$ picks a random linear combination $r\in\mathbb{F}^m$ and sends r to $\mathcal{P}$.
2. $\mathcal{P}$ responds with $w=r^TU\in\mathbb{F}^n$.
3. $\mathcal{V}$ queries a set $Q\subset[n]$ of t random symbols $U[j]$, $j\in Q$.
4. $\mathcal{V}$ accepts iff $w\in L$ and w is consistent with $U[Q]$ and r. That is, for every $j\in Q$ we have

$$\sum_{i=1}^{m} r_j\cdot U_{i,j}=w_j.$$

Figure 2: Test-Interleaved $(\mathbb{F},L[n,k,d],m,t;U)$

The following lemma follows directly from the linearity of L.

Lemma 4.1. *If $U\in L^m$ and $\mathcal{P}$ is honest, then $\mathcal{V}$ always accepts.*

Our soundness analysis will rely on the following lemma.

Lemma 4.2. *Let e be a positive integer such that $e<d/4$. Suppose $d(U^*,L^m)>e$. Then, for a random w^* in the row-span of U^*, we have*

$$\Pr[d(w^*,L)\le e]\le(e+1)/|\mathbb{F}|.$$

particular in the protocols from [DI06, IPS09] on which we build. Its soundness requires the MPC protocol to be *adaptively secure* to accommodate $\mathcal{P}$'s ability to make the locations of inconsistencies depend on $\mathcal{V}$'s random challenge; when the MPC adversary is adaptive, it can potentially corrupt all parties observing such inconsistencies. Indeed, the compiler from statistically secure MPC to ZK proofs from [IKOS09] relies on the adaptive security of the underlying MPC protocol.

Proof: Let L^* be the row-span of U^*. We consider two cases.

CASE 1: There exists $v^* \in L^*$ such that $d(v^*, L) > 2e$. In this case, we show that

$$\Pr_{w^* \in_R L^*}[d(w^*, L) \leq e] \leq 1/|\mathbb{F}|. \tag{1}$$

Indeed, using a basis for L^* that includes v^*, a random $w^* \in L^*$ can be written as $\alpha v^* + x$, where $\alpha \in_R \mathbb{F}$ and x is distributed independently of α. We argue that conditioned on any choice of x, there can be at most one choice of α such that $d(\alpha v^* + x, L) \leq e$, which implies (1). This follows by observing that if $d(\alpha v^* + x_0, L) \leq e$ and $d(\alpha' v^* + x_0, L) \leq e$ for $\alpha \neq \alpha'$, then by the triangle inequality we have $d((\alpha - \alpha')v^*, L) \leq 2e$, contradicting the assumption that $d(v^*, L) > 2e$.

CASE 2: For every $v^* \in L^*, d(v^*, L) \leq 2e$. We show that in this case

$$\Pr_{w^* \in_R L^*}[d(w^*, L) \leq e] \leq (e+1)/|\mathbb{F}|. \tag{2}$$

Let U_i^* be the i-th row of U^* and let $E_i = \Delta(U_i^*, L)$. Note that, since $2e < d/2$, each U_i^* can be written uniquely as $U_i^* = u_i + \chi_i$ where $u_i \in L$ and χ_i is nonzero exactly in its E_i entries. Let $E = \cup_{i=1}^m E_i$. Since $d(U^*, L^m) > e$, we have $|E| > e$. We show that for each $j \in E$, except with $1/|\mathbb{F}|$ probability over a random choice of w^* from L^*, either $j \in \Delta(w^*, L)$ or $d(w^*, L) > e$, from which the claim will follow.

Suppose $j \in E_i$. As before, we write $w^* = \alpha U_i^* + x$ for $\alpha \in_R \mathbb{F}$ and x distributed independently of α. Condition on any possible choice x_0 of x. Define a bad set

$$B_j = \{\alpha : j \notin \Delta(\alpha U_i^* + x_0, L) \wedge d(\alpha U_i^* + x_0, L) \leq e\}.$$

We show that $|B_j| \leq 1$. Suppose towards contradiction that there are two distinct $\alpha, \alpha' \in \mathbb{F}$ such that for $z = \alpha U_i^* + x_0$ and $z' = \alpha' U_i^* + x_0$ we have $d(z, L) \leq e$, $d(z', L) \leq e$, $j \notin \Delta(z, L)$, and $j \notin \Delta(z', L)$. Since $d > 4e$, for any z^* in the linear span of z and z' we have $j \notin \Delta(z^*, L)$. Since U_i^* is in this linear span, we have $j \notin \Delta(U_i^*, L)$, in contradiction to the assumption that $j \in E_i$.

We have shown that for each $j \in E$, conditioned on every possible choice of x, either $j \in \Delta(w^*, L)$ or $d(w^*, L) > e$ except with $1/|\mathbb{F}|$ probability over the choice of α. It follows that the same holds for a random choice of x. Taking a union bound over the first $e+1$ elements of E we get that $\Pr_{w^* \in_R L^*}[d(w^*, L) \leq e] \leq (e+1)/|\mathbb{F}|$ as required. □

We now prove the soundness of the testing procedure when the given oracle is far from L^m.

Theorem 4.4. *Let e be a positive integer such that $e < d/4$. Suppose $d(U^*, L^m) > e$. Then, for any malicious $\mathcal{P}$ strategy, the oracle U^* is rejected by $\mathcal{V}$ except with $\leq (1 - e/n)^t + (e+1)/|\mathbb{F}|$ probability.*

Proof: Letting $w^* = r^T U^*$, it follows from Lemma 4.2 that

$$\begin{aligned}
\Pr[\mathcal{V} \text{ accepts } U^*] &\leq \Pr[\mathcal{V} \text{ accepts} \mid d(w^*, L) > e] \\
&\quad + \Pr[d(w^*, L) \leq e] \\
&\leq \frac{\binom{n-e-1}{t}}{\binom{n}{t}} + (e+1)/|\mathbb{F}| \\
&\leq (1 - e/n)^t + (e+1)/|\mathbb{F}|
\end{aligned}$$

as required. □

In Appendix B.1 we present a simple generalization of the testing algorithm that uses σ linear combinations to amplify soundness.

4.1.1 Improving the Analysis

In our preceding analysis we had the requirement $e < d/4$ in Theorem 4.4. Relaxing this to $e < d/3$ or possibly even $e < d/2$ with essentially the same soundness error bound can improve the concrete parameters of the ZKIPCP significantly. We remark that Theorem 4.4 holds for any linear code. However, when considering concrete parameters it would suffice to present a tighter analysis for RS codes. In [AHIV17] we conjectured that the analysis could be extended to the case $e < d/3$. Soon after, in private communication, Ronny Roth and Gilles Zémor [RZ17] independently proved the stronger result for $e < d/3$. Below, we present this analysis. At the end of this section, we discuss subsequent improvements and the current state-of-the-art. For our implementation, we relied on the tightest analysis to identify concrete parameters.

The case $e < d/3$: As a first step, we reduce such a stronger version of Theorem 4.4 to a simple lemma about the distance of points on an affine line from an RS code. We begin by showing that for any linear code over a sufficiently large field, when $e < d/3$ we can restrict the attention to Case 1 from the proof of Lemma 4.2.

Lemma 4.3. *Let L be an $[n, k, d]$ linear code over $\mathbb{F}$. Let e be a positive integer such that $e < d/3$ and $|\mathbb{F}| \geq e$. Suppose $d(U^*, L^m) > e$. Then, there exists $v^* \in L^*$ such that $d(v^*, L) > e$, where L^* is the row-span of U^*.*

Proof: Assume towards a contradiction that $d(v^*, L) \leq e$ for all $v^* \in L^*$ and suppose that $v_0^* \in L^*$ maximizes the distance from L. Since $d(U^*, L^m) > e$, there must be a row U_i^* such that $\Delta(U_i^*, L) \setminus \Delta(v_0^*, L) \neq \emptyset$, as v_0^* introduces at most e errors. Let $v_0^* = u_0 + \chi_0$ and $U_i^* = u_i + \chi_i$ for $u_0, u_i \in L$ and χ_0, χ_i of weight $\leq e$, as by our assumption above all elements in L^* introduce at most e errors. We argue that there exists $\alpha \in \mathbb{F}$ such that for $\hat{v} = v_0^* + \alpha U_i^*$ it holds that $d(\hat{v}, L) > d(v_0^*, L)$, contradicting the choice of v_0^*. Specifically, since $d(v_0^*, L) \leq e$ and $d(U_i^*, L) \leq e$, there is a codeword $w \in L$ that is at most $2e$-far from $\hat{v}$, namely by setting $w = u_0 + \alpha u_i$ and aggregating the errors on the worst case. Furthermore, since $d(\hat{v}, L) \leq e$ there must be a codeword $w' \in L$ that at most than e-far from $\hat{v}$. Now, since $d > 3e$, it must be the case that $w = w'$, or else $d(w, w') \leq 3e$ which is less than d. However, for any $j \in \Delta(v_0^*, L) \cup \Delta(U_i^*, L)$ there is at most one choice of α such that the j^{th} component of $\chi_0 + \alpha\chi_i$ goes to zero. By an union bound, there are at most $2e$ such α's. Considering any other α, we arrive at a contradiction. □

Given Lemma 4.3, as we argue below, that in order to obtain an equivalent guarantee to Lemma 4.2, it will suffice to show that in any affine subspace of $\mathbb{F}^n$, either all points are e-close to L or almost all are not. This reduces to showing the same for 1-dimensional spaces which is claimed in the following lemma. The proof of this lemma due to Roth and Zémor is presented in Appendix A.

Lemma 4.4. *Let $L = \mathsf{RS}_{\mathbb{F},n,k,\eta}$ be a Reed-Solomon code with minimal distance $d = n - k + 1$. Let e be a positive integer such that $e < d/3$. Then for every $u, v \in \mathbb{F}^n$, defining an affine line $\ell_{u,v} = \{u + \alpha v : \alpha \in \mathbb{F}\}$, either (1) for every $x \in \ell_{u,v}$ we have $d(x, L) \leq e$, or (2) for at most d points $x \in \ell_{u,v}$ we have $d(x, L) \leq e$.*

We remark here that we do not have a counterexample to Lemma 4.4 when we relax $e < d/2$ and even when L is a general linear code. Indeed, subsequent work analyzed an improved analysis for the case $e < d/2$ with a relaxed version of condition (2) where d is replaced by n and this relaxed version will deliver better efficiency of our ZKIPCP.

In order to extend the previous analysis to the case where $e < d/3$, we need an alternate version of Lemma 4.2 which we state next:

Lemma 4.5. *Let $L = \mathsf{RS}_{\mathbb{F},n,k,\eta}$ be a Reed-Solomon code with minimal distance $d = n - k + 1$ and e a positive integer such that $e < d/3$. Suppose $d(U^*, L^m) > e$. Then, for a random w^* in the row-span of U^*, we have*

$$\Pr[d(w^*, L) \leq e] \leq d/|\mathbb{F}|.$$

Before we prove Lemma 4.5, we can conclude analogously to the previous analysis that this lemma implies the following stronger version of Theorem 4.4.

Theorem 4.5. *Let e be a positive integer such that $e < d/3$. Suppose $d(U^*, L^m) > e$. Then, for any malicious $\mathcal{P}$ strategy, the oracle U^* is rejected by $\mathcal{V}$ except with $\leq (1 - e/n)^t + d/|\mathbb{F}|$ probability.*

Proof of Lemma 4.5. On a high-level Lemma 4.5 follows by extending Lemma 4.4 from lines to general affine subspaces. This extension follows from the fact that if a subspace has a point that is far from L, then we can partition the subspace (minus the point) into lines containing this point. Now, from Lemma 4.4 we have that at most $d/|F|$ points can be near L for lines, therefore, we have that the same for affine subspace L^*.

In more detail, we will follow the proof of Lemma 4.2. Case 1 of the analysis remains the same, so we only have to argue Case 2 where we have that for every $v^* \in L^*$, $d(v^*, L) \leq 2e$. Next, since $d(U^*, L) > e$, we use Lemma 4.3 to conclude that there exists $v^* \in L^*$ such that $d(v^*, L) > e$. As before, we can express the points in L^* as $x + \alpha v^*$ where $\alpha \in_R \mathbb{F}$ and x is distributed independently of α. For any fixed x, we have that there exists α such that $x + \alpha v^*$ is more than e far from L. Now from part (2) in Lemma 4.4, we can conclude that there are at most d values of α for which $x + \alpha v^*$ is at most e-far from L. Since this is true for each x (each line), it is true for the entire space L^*. This concludes the proof of Lemma 4.5. ■

Subsequent improvements. Subsequent to the publication of [AHIV17], several works [BBHR18, BGKS20, BCI+20] have improved this analysis, where currently the best analysis is presented in [BCI+20].

Lemma 4.6. *[BCI+20, Theorem 1.2] Let $L = \mathsf{RS}_{\mathbb{F},n,k,\eta}$ be a Reed-Solomon code with minimal distance $d = n - k + 1$ and e a positive integer such that $e < d/2$. Suppose $d(U', L^m) > e$. Then, for a random w^* in the column-span of U', we have*

$$\Pr[d(w^*, L') \leq e] \leq n/|\mathbb{F}|.$$

4.2 Testing Linear Constraints over Interleaved Reed-Solomon Codes

In this section we describe an efficient procedure for testing that a message encoded by an interleaved RS code satisfies a given set of linear constraints. This generalizes a procedure from [Gro09, IPS09] for testing that such an encoded message satisfies a given set of replication constraints. In the following we assign a message in $\mathbb{F}^\ell$ to a codeword in $\mathbb{F}^n$ by considering a fixed set of ℓ evaluation points of the polynomial defined by the codeword. Note that while each codeword has a unique message assigned to it, several different codewords can be "decoded" into the same message. As the degree of the polynomial corresponding to the codeword can be higher than $\ell - 1$. On the other hand, if the degree of the polynomial corresponding to the codeword is restricted to be smaller than ℓ, the encoding becomes unique.

We now describe a simple testing algorithm for checking that the message x encoded by U satisfies a given system of linear equations $Ax = b$, for $A \in \mathbb{F}^{m\ell \times m\ell}$ and $b \in \mathbb{F}^{m\ell}$. (We will always apply this test with a sparse matrix A containing $O(m\ell)$ nonzero entries.) The test simply picks a random linear combination $r \in \mathbb{F}^{m\ell}$ and checks that $(r^T A)x = r^T b$. Note that if $Ax \neq b$, the test will only pass with $1/|\mathbb{F}|$ probability. To make the test sublinear, we let the prover provide a polynomial encoding $(r^T A)x$ and check its consistency with $r^T b$ and with U on t randomly chosen symbols. To further simplify the

description and analysis of the testing algorithm, we assume that U is promised to be e-close to L^m. Our final IPCP will run the **Test-Interleaved** from Section 4.1 to ensure that if the promise is violated, this is caught with high probability. The complete test is described in Figure 3.

Oracle: A purported L^m-codeword U that should encode a message $x \in \mathbb{F}^{m\ell}$ satisfying $Ax = b$.

Interactive testing:

1. $\mathcal{V}$ picks a random vector $r \in \mathbb{F}^{m\ell}$ and sends r to $\mathcal{P}$.
2. $\mathcal{V}$ and $\mathcal{P}$ compute
$$r^T A = (\underbrace{r_{11}, \ldots, r_{1\ell}}_{\substack{\text{multipliers for} \\ \text{the first block}}}, \ldots, \underbrace{r_{m1}, \ldots, r_{m\ell}}_{\substack{\text{multipliers for} \\ \text{the last block}}})$$
and, for $i \in [m]$, let $r_i(\cdot)$ be the unique polynomial of degree $< \ell$ such that $r_i(\zeta_c) = r_{ic}$ for every $c \in [\ell]$.
3. $\mathcal{P}$ sends the $k + \ell - 1$ coefficients of the polynomial defined by $q(\cdot) = \sum_{i=1}^{m} \underbrace{r_i(\cdot) \cdot p_i(\cdot)}_{\substack{\text{pairwise product} \\ \text{within a block}}}$, where $p_i(\cdot)$ is the polynomial of degree $< k$ corresponding to row i of U.
4. $\mathcal{V}$ queries a set $Q \subset [n]$ of t random symbols $U[j]$, $j \in Q$.
5. $\mathcal{V}$ accepts if the following conditions hold:
   (a) $\sum_{c\in[\ell]} q(\zeta_c) = \sum_{i\in[m],c\in[\ell]} r_{ic} b_{ic}$.
   (b) For every $j \in Q$, $\sum_{i=1}^{m} r_i(\eta_j) \cdot U_{i,j} = q(\eta_j)$.

Figure 3: Test-Linear-Constraints-IRS$(\mathbb{F}, L = \mathsf{RS}_{\mathbb{F},n,k,\eta}, m, t, \zeta, A, b; U)$

The following lemma easily follows by inspection.

Lemma 4.7. *If $U \in L^m$, U encodes x such that $Ax = b$, and $\mathcal{P}$ is honest, $\mathcal{V}$ always accepts.*

Soundness is argued by the following lemma.

Lemma 4.8. *Let e be a positive integer such that $e < d/2$. Suppose that a (badly formed) oracle U^* is e-close to a codeword $U \in L^m$ encoding $x \in \mathbb{F}^{m\ell}$ such that $Ax \neq b$. Then, for any malicious $\mathcal{P}$ strategy, U^* is rejected by $\mathcal{V}$ except with at most $1/|\mathbb{F}| + ((e + k + \ell)/n)^t$ probability.*

Proof: Let $q(\cdot)$ be the polynomial generated in Step 3 following the honest $\mathcal{P}$ strategy on input U. Since we assume that $Ax \neq b$, it holds that

$$\Pr_r[r^T Ax = r^T b] = 1/|\mathbb{F}|.$$

Namely, except with probability $1/|\mathbb{F}|$ over the choice of r in Step 1, the polynomial $q(\cdot)$ fails to satisfy the condition in Step 5a. This is due to the fact that

$$\sum_{c\in[\ell]} q(\zeta_c) = (r^T A)x$$

and

$$\sum_{i\in[m],c\in[\ell]} r_{ic}b_{ic} = r^T b.$$

Next, we analyze the probability that a malicious $\mathcal{P}$ strategy is rejected conditioned on $q(\cdot)$ failing as above. Let $q'(\cdot)$ be the polynomial sent by the prover. If $q'(\cdot) = q(\cdot)$, then $\mathcal{V}$ rejects in Step 5a with probability $1/|\mathbb{F}|$. Else, using the fact that $q(\cdot)$ and $q'(\cdot)$ are of degree at most $k+\ell-2$, we have that the number of indices $j \in [n]$ for which $q(\eta_j) = q'(\eta_j)$ is at most $k+\ell-2$. Let Q' be the set of indices on which they agree. Then $\mathcal{V}$ rejects in Step 5b whenever Q selected in Step 5 contains an index $i \notin Q' \cup E$, where $E = \Delta(U^*, L^m)$. This fails to happen with probability at most $\binom{e+k+\ell-2}{t}/\binom{n}{t} \leq ((e+k+\ell)/n)^t$. The lemma now follows by a simple union bound. □

4.3 Testing Quadratic Constraints over Interleaved Reed-Solomon Codes

In this section we describe a simple test for verifying that vectors $x, y, z \in \mathbb{F}^{m\ell}$ respectively encoded by $U^x, U^y, U^z \in L^m$, satisfy the constraints $x \odot y + a \odot z = b$ for some known $a, b \in \mathbb{F}^{m\ell}$, where $\odot$ denotes pointwise product. Letting $L = \mathsf{RS}_{\mathbb{F},n,k,\eta}$, $U^a = \mathsf{Encode}(a)$ and $U^b = \mathsf{Encode}(b)$, this test reduces to checking that $U^x \odot U^y + U^a \odot U^z - U^b$ encodes the all zeros message $0^{m\ell}$ in the (interleaved extension of) $\hat{L} = \mathsf{RS}_{\mathbb{F},n,2k-1,\eta}$. This could be done using the general membership test for interleaved linear codes (**Test-Interleaved** from Section 4.1), since the set of codewords in $\hat{L}$ that encodes the all zeros message is a linear subcode of $\hat{L}$. In Figure 4 we present this test in a self-contained way, exploiting the promise that U^x, U^y, U^z are close to L^m for a tighter analysis.

The following lemma follows again directly from the description.

Lemma 4.9. *Let* $U = \left[\begin{array}{c|c|c|c} U^x & U^y & U^z & U^w \end{array}\right]^T$ *where* $U^w, U^x, U^y, U^z \in L^m$. *If* U^x, U^y, U^z *encode vectors* $x, y, z \in \mathbb{F}^{m\ell}$ *satisfying* $x \odot y + a \odot z = b$ *and* $\mathcal{P}$ *is honest,* $\mathcal{V}$ *always accepts.*

Soundness is argued by the following lemma.

Lemma 4.10. *Let* e *be a positive integer such that* $e < d/2$. *Let* U^{x*}, U^{y*}, U^{z*} *be badly formed oracles and let* $U^* \in \mathbb{F}^{3m\times n}$ *be the matrix obtained by vertically juxtaposing the corresponding* $m \times n$ *matrices. Suppose* $d(U^*, L^{3m}) \leq e$, *and let* U^x, U^y, U^z, *respectively, be the (unique) codewords in* L^m *that are closest to* U^{x*}, U^{y*}, U^{z*}. *Suppose* U^x, U^y, U^z *encode* x, y, z *such that* $x \odot y + a \odot z \neq b$. *Then, for any malicious* $\mathcal{P}$ *strategy,* (U^{x*}, U^{y*}, U^{z*}) *is rejected by* $\mathcal{V}$ *except with at most* $1/|\mathbb{F}| + ((e+2k)/n)^t$ *probability.*

Proof: Let $p_0(\cdot)$ be the polynomial generated in Step 3 following the honest $\mathcal{P}$ strategy on U^x, U^y, U^z. Since x, y, z do not satisfy the constraint $x \odot y + a \odot z = b$, the polynomial $p_0(\cdot)$ fails to satisfy the condition in Step 5a except with probability $1/|\mathbb{F}|$ over the choice of r in Step 2. Indeed, we have $p_0(\cdot) = \sum_{i=1}^m r_i \cdot p_i(\cdot)$ and there must exist an index i and a point ζ_c such that $p_i(\zeta_c) \neq 0$.

Next, we analyze the probability that a malicious $\mathcal{P}$ strategy is rejected conditioned on p_0 failing as above. Let $p_0'(\cdot)$ be the polynomial sent by the prover in Step 3. If $p_0'(\cdot) = p_0(\cdot)$, then $\mathcal{V}$ rejects in Step 5a with probability $1/|\mathbb{F}|$. Else, using the fact that $p_0(\cdot)$ and $p_0'(\cdot)$ are of degree at most $2k-2$, we have that the number of indices $j \in [n]$ for which $p_0(\eta_j) = p_0'(\eta_j)$ is at most $2k-2$. Let Q' be the set of indices on which $p_0(\cdot)$ and $p_0'(\cdot)$ agree. Then $\mathcal{V}$ rejects in Step 5b whenever Q selected in Step 4 contains an index $i \notin Q' \cup E$, where $E = \Delta(U^*, L^{3m})$. This fails to happen with probability at most

$$\binom{e+2k-2}{t} \Big/ \binom{n}{t} \leq ((e+2k)/n)^t.$$

The lemma now follows by a union bound. □

Oracle: Purported L^{4m}-codeword U, where $U = \left[\ U^x \mid U^y \mid U^z\ \right]^T$, $U^x, U^y, U^z \in L^m$ and U^x, U^y, U^z that allegedly encode messages $x, y, z \in \mathbb{F}^{m\ell}$ satisfying $x \odot y + a \odot z = b$.
Interactive testing:

1. Let $U^a = \mathsf{Encode}_\zeta(a)$ and $U^b = \mathsf{Encode}_\zeta(b)$.
2. $\mathcal{V}$ picks a random linear combinations $r \in \mathbb{F}^m$ and sends r to $\mathcal{P}$.
3. $\mathcal{P}$ sends the $2k-1$ coefficients of the polynomial p_0 defined by

$$p_0(\cdot) = \sum_{i=1}^{m} r_i \cdot p_i(\cdot), \quad \text{where } p_i(\cdot) = p_i^x(\cdot) \cdot p_i^y(\cdot) + p_i^a(\cdot) \cdot p_i^z(\cdot) - p_i^b(\cdot),$$

   and where p_i^x, p_i^y, p_i^z are the polynomials of degree $< k$ corresponding to row i of U^x, U^y, U^z, and p_i^a, p_i^b are the polynomials of degree $< \ell$ corresponding to row i of U^a, U^b.
4. $\mathcal{V}$ picks a random index set $Q \subset [n]$ of size t, and queries $U[j]$, $j \in Q$.
5. $\mathcal{V}$ accepts if the following conditions hold:
   (a) $p_0(\zeta_c) = 0$ for every $c \in [\ell]$.
   (b) For every $j \in Q$, it holds that

$$\sum_{i=1}^{m} r_i \cdot \left[U_{i,j}^x \cdot U_{i,j}^y + U_{i,j}^a \cdot U_{i,j}^z - U_{i,j}^b\right] = p_0(\eta_j).$$

Figure 4: Test-Quadratic-Constraints-IRS$(\mathbb{F}, L = \mathsf{RS}_{\mathbb{F},n,k,\eta}, m, t, \zeta, a, b; U)$

4.4 IPCP for Arithmetic Circuits

In this section, we provide our IPCP for arithmetic circuits. Fix a large finite field $\mathbb{F}$. Let $\mathrm{C} : \mathbb{F}^{n_i} \to \mathbb{F}$ be an arithmetic circuit. Without loss of generality, we will assume that the circuit contains only ADD and MULTIPLY gates with fan-in two. We show how a prover can convince a verifier that $\mathrm{C}(w) = 1$.

Protocol IPCP$(\mathrm{C}, \mathbb{F})$**.**

- **Input:** The prover $\mathcal{P}$ and the verifier $\mathcal{V}$ share a common input arithmetic circuit $\mathrm{C} : \mathbb{F}^{n_i} \to \mathbb{F}$ and input statement x. $\mathcal{P}$ additionally has input $\overline{\alpha} = (\alpha_1, \ldots, \alpha_{n_i})$ such that $\mathrm{C}(\overline{\alpha}) = 1$.
- **Oracle π:** Let m, ℓ be integers such that $m \cdot \ell > n_i + s$ where s is the number of gates in the circuit. Then $\mathcal{P}$ generates an extended witness $w \in \mathbb{F}^{m\ell}$ where the first $n_i + s$ entries of w are

$$(\alpha_1, \ldots, \alpha_{n_i}, \beta_1, \ldots, \beta_s)$$

  where β_i is the output of the i^{th} gate when evaluating $\mathrm{C}(\overline{\alpha})$. $\mathcal{P}$ defines a system of constraints that contains the following constraint for every multiplication gate g in the circuit C

$$\beta_a \cdot \beta_b - \beta_c = 0$$

  and for every addition gate, the constraint

$$\beta_a + \beta_b - \beta_c = 0$$

where β_a and β_b are the input values to the gate g and β_c is the output value in the extended witness. For the output gate we include the constraint $\beta_a + \beta_b - 1 = 1$ if the final gate is an addition gate, and $\beta_a \cdot \beta_b - 1 = 0$ if it is a multiplication gate. $\mathcal{P}$ constructs vectors x, y and z in $\mathbb{F}^{m\ell}$ where the j^{th} entry of x, y and z contains the values β_a, β_b, and β_c corresponding to the j^{th} multiplication gate in w. $\mathcal{P}$ and $\mathcal{V}$ construct matrices P_x, P_y and P_z in $\mathbb{F}^{m\ell \times m\ell}$ such that

$$x = P_x w, y = P_y w, z = P_z w.$$

Finally, it constructs matrix $P_{\mathbf{add}} \in \mathbb{F}^{m\ell \times m\ell}$ such that the j^{th} position of $P_{\mathbf{add}} w$ equals $\beta_a + \beta_b - \beta_c$ where β_a, β_b, and β_c correspond to the j^{th} addition gate of the circuit in w. Let $U^w, U^x, U^y, U^z \in L^m$ respectively encode w, x, y, z where $L = \mathsf{RS}_{\mathbb{F},n,k,\eta}$. $\mathcal{P}$ sets the oracle π as $U \in L^{4m}$ which is set as the vertical juxtaposition of the following four matrices $U^w, U^x, U^y, U^z \in L^m$.

All the linear constraints can be expressed as one large linear constraint matrix:

$$A = \left[\begin{array}{c|c} I_{3m\ell \times 3m\ell} & -P \\ \hline \mathbf{0}_{m\ell \times 3m\ell} & P_{\mathbf{add}} \end{array}\right], P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}, b = \mathbf{0}^{4m\ell}$$

- **The interactive protocol:**
  $\mathcal{V}$ and $\mathcal{P}$ run the following tests.

  1. **Test-Interleaved** $(\mathbb{F}, L, 4m, t; U)$
  2. **Test-Linear-Constraints-IRS** $(\mathbb{F}, L, 4m, t, \zeta, A, b; U)$
  3. **Test-Quadratic-Constraints-IRS**$(\mathbb{F}, L, m, t, \zeta, (-\mathbf{1})^{m\ell}, \mathbf{0}^{m\ell}; U)$

  Since all the tests open the same number of columns t in U_w, U_x, U_y, U_z, then $\mathcal{V}$ will simply open t columns of U. $\mathcal{V}$ rejects if it rejects in any of the tests above.

The completeness of our IPCP follows from the following lemma.

Lemma 4.11. *If $U^w, U^x, U^y, U^z \in L^m$ encode vectors $w, x, y, z \in \mathbb{F}^{m\ell}$ satisfying*

$$x = P_x w,\ y = P_y w,\ z = P_z w,\ x \odot y + (-\mathbf{1})^{m\ell} \odot z = \mathbf{0}^{m\ell},\ P_{\mathbf{add}} w = \mathbf{0}^{m\ell}$$

and $\mathcal{P}$ is honest, $\mathcal{V}$ always accepts.

The proof follows directly from Lemmas 4.1, 4.7 and 4.9. Next, soundness is argued by the following lemma.

Lemma 4.12. *Let e be a positive integer such that $e < d/3$ and suppose that there exists no $\overline{\alpha}$ such that $\mathrm{C}(\overline{\alpha}) = 1$. Then, for any maliciously formed oracle U^* and any malicious prover strategy, the verifier rejects except with at most $(d+2)/|\mathbb{F}| + (1 - e/n)^t + 2((e+2k)/n)^t$ probability.*

Proof: On a high-level, soundness will essentially follow by the soundness of the individual tests and the overall soundness error follows by a direct application of a union bound over the soundness of these tests. In more detail, let U be the vertical juxtaposition of $U^{w*}, U^{x*}, U^{y*}, U^{z*}$. Then we argue soundness by considering the following cases and applying a union bound:

Case $d(U, L^{4m}) > e$**:** Since $e < d/3$, we can conclude from Theorem 4.5 that the verifier rejects in **Test-Interleaved** executed in Step 1 except with probability $(1 - e/n)^t + d/|\mathbb{F}|$.

Case $d(U, L^{4m}) \leq e$**:** Next, let $U^w, U^x, U^y, U^z \in L^m$ be the codes that are respectively close to U^{w*}, U^{x*}, U^{y*}, U^{z*} and encode the messages w, x, y, z. Recall that there exists no w, x, y, z that satisfy all the following constraints:

$x = P_x w,\ y = P_y w,\ z = P_z w,$
$x \odot y + (\mathbf{-1})^{m\ell} \odot z = \mathbf{0}^{m\ell}$ and $P_{\mathbf{add}} w = \mathbf{0}^{m\ell}$.

Then we can conclude from Lemmas 4.8 and 4.10 by applying a union bound on the corresponding tests that the verifier rejects except with probability:

$$2/|\mathbb{F}| + (e + k + \ell)/n)^t + ((e + 2k)/n)^t \\ < 2 \cdot \left(1/|\mathbb{F}| + ((e + 2k)/n)^t\right).$$

□

The following theorem follows from the construction described above and the preceding Lemmas.

Theorem 4.6. *Fix parameters n, m, ℓ, k, t, e such that $e < (n - k)/4$. Let $\mathrm{C} : \mathbb{F}^{n_i} \to \mathbb{F}$ be an arithmetic circuit of size s, where $|\mathbb{F}| \geq n$ and $m \cdot \ell > n_i + s$. Then protocol* IPCP$(\mathrm{C}, \mathbb{F})$ *satisfies the following:*

- COMPLETENESS: *If $\overline{\alpha}$ is such that $\mathrm{C}(\overline{\alpha}) = 1$ and oracle π is generated honestly as described in the protocol, then*
  $\Pr[(\mathcal{P}(\mathrm{C}, w), \mathcal{V}^{\pi}(\mathrm{C})) = 1] = 1.$
- SOUNDNESS: *If there is no $\overline{\alpha}$ is such that $\mathrm{C}(\overline{\alpha}) = 1$, then for every (unbounded) prover strategy $\mathcal{P}^*$ and every $\tilde{\pi} \in \mathbb{F}^{4mn}$,*
  $Pr[(\mathcal{P}^*, \mathcal{V}^{\tilde{\pi}}(\mathrm{C})) = 1] \leq (d + 2)/|\mathbb{F}| + (1 - e/n)^t + 2((e + 2k)/n)^t.$
- COMPLEXITY: *The number of field operations performed is* $\mathsf{poly}(|\mathrm{C}|, n)$. *The number of field elements communicated by $\mathcal{P}$ to $\mathcal{V}$ is $k + (k + \ell - 1) + (2 \cdot k - 1)$ whereas $\mathcal{V}$ reads t symbols from $\mathbb{F}^{4m}$.*

The first term in the communication cost is the communication incurred by the test-interleaved protocol, the second term is due to the linear-constraint test and the final term results from our quadratic-constraint test.

4.5 IPCP for Boolean Circuits

In order to obtain the benefits in soundness from running our IPCP over a large field $\mathbb{F}$, we show how we can prove the validity of a Boolean circuit $\mathrm{C} : \{0,1\}^{n_i} \to \{0,1\}$ by encoding the witness in any larger field $\mathbb{F}$. First, the prover will map the Boolean 0 within the witness to the additive identity ϵ_0 in $\mathbb{F}$, and the Boolean 1 to the multiplicative identity ϵ_1 in $\mathbb{F}$. Now, we can enforce that each element in the witness is a 0 or 1 by introducing a quadratic constraint $\beta^2 - \beta = 0$.

Next, given that binary constraints are already enforced, we proceed by demonstrating how we incorporate the constraints based on the XOR and ADD gates. In fact, we will show that all gate constraints can be expressed as a linear relation on the witness bits. Let x be a column vector consisting of the witness string. We will construct a matrix A and a column vector w such that if w is a binary valid witness then the elements

of Aw will all be 0, and if w is binary and is not a valid witness then at least one element of Aw will be nonzero. For each XOR and AND gate in the circuit we will create a row in the matrix corresponding to the enforcement of that relation in the witness. Specifically, besides including the input bits x, the vector w will include one additional bit for each XOR and AND gate. We explain the purpose of these extra bits next.

Given integers b_1 and b_2 consider the arithmetic constraint $b_1 + b_2 = r_0 + 2 \cdot r_1$ over the integers. In this constraint, if we enforce that all values are bits then r_0 is the XOR of b_1 and b_2 and r_1 is the AND of b_1 and b_2. In order to make sure that b_1 XOR b_2 equals b_3 in w, we require the prover to include in the witness an auxiliary bit d and enforce the linear constraint $b_1 + b_2 = b_3 + 2 \cdot d$, as well as the binary constraints that b_3 and d are bits. Analogously, to ensure b_1 AND b_2 equals b_3, we include an auxiliary bit d and enforce the linear constraint $b_1 + b_2 = d + 2 \cdot b_3$ and the binary constraint that b_3 and d are bits. To conclude, we observe that if the values have been enforced to be a binary constraint then checking the arithmetic constraints over integers can be done by checking the equation modulo a sufficiently large prime ($p \geq 3$).

We can also extend this idea to consider more complex gates such as addition modulo 2^{32} over 32-bit inputs and outputs. This can be expressed as a linear constraint over the bits. Suppose $a = (a_0, \ldots, a_{31})$, $b = (b_0, \ldots, b_{31})$ and $c = (c_0, \ldots, c_{31})$ are the input and output bits, the constraint $a + b = c \bmod 2^{32}$ can be expressed as

$$\sum_{i=0}^{31} 2^i \cdot a_i + \sum_{i=0}^{31} 2^i \cdot b_i = 2^{32} \cdot d + \sum_{i=0}^{31} 2^i \cdot c_i$$

where d is an auxiliary input bit, and all values are enforced to be bits. However, this will require using a finite field $\mathbb{F}$ with characteristic $p > 2^{33}$.

4.6 Achieving Zero-Knowledge

Note first that the verifier obtains two types of information in two different building blocks of the IPCP. First, it obtains linear combinations of codewords in a linear code L. Second, it probes a small number of symbols from each codeword. Since codewords are used to encode the NP witness, both types of information give the verifier partial information about the NP witness, and thus the basic IPCP we described is not zero-knowledge. Fortunately, ensuring zero-knowledge only requires introducing small modifications to the construction and analysis. Specifically, the second type of "local" information about the codewords is made harmless by making the encoding randomized, so that probing just a few symbols in each codeword reveals no information about the encoded message. The high level idea for making the first type of information harmless is to use an additional random codeword for blinding the linear combination of codewords revealed to the verifier. However, this needs to be done in a way that does not compromise soundness. Below we describe the modifications required for each of the IPCP ingredients.

4.6.1 ZK Testing of Interleaved Linear Codes

Recall that in the verification algorithm **Test-Interleaved** from Section 4.1, $\mathcal{V}$ obtains a linear combination of the form $w = r^T U$, where $U \in \mathbb{F}^{m \times n}$ is a matrix whose rows should be codewords in L. A natural approach for making this linear combination hide U is by allowing the prover to add to the rows of U an additional random codeword u' that is used for blinding. A simple implementation of this idea that provides a slightly inferior soundness guarantee is as follows. Apply the algorithm **Test-Interleaved** to L^{m+1}, with an extended oracle U' whose first m rows contain U and whose last row is u'. Letting $w' = r^T U + r' u'$ be the random linear combination obtained by $\mathcal{V}$, the test fails to be zero-knowledge when $r' = 0$, which occurs with $1/|\mathbb{F}|$ probability. Alternatively, settling for a slightly worse soundness guarantee (where $e/|\mathbb{F}|$

is replaced by $e/(|\mathbb{F}|-1)$), one could just let r' be a random *nonzero* field element, and get perfect zero-knowledge. It turns out, however, that one could fix r' to 1 and still get the same soundness guarantee about U as in Lemma 4.2 since we can apply the same the decomposition argument. This "affine" variant of **Test-Interleaved** is described and analyzed in Appendix B.3.

4.6.2 ZK Testing of Linear Constraints over Interleaved Reed-Solomon Codes

The verification algorithm for the linear constraints $Ax = b$ samples a random vector r, obtains $r^T Ax$, and compares it with $r^T b$. Looking more carefully at our actual protocol, the verifier obtains a polynomial $q(\cdot)$ and checks whether the equality $\sum_{c\in[\ell]} q(\zeta_c) = \sum_{i\in[m],c\in[\ell]} r_{ic}b_{ic}$ holds. While the sum itself does not reveal any additional information beyond what is already known, namely $r^T b$, the individual evaluations of $q(\cdot)$, i.e. $q(\zeta_c)$, may reveal information about the inputs. To hide this information, a simple idea is for $\mathcal{P}$ to provide an additional vector u' along with U that encodes a message $(\gamma_1, \ldots, \gamma_\ell)$ such that $\sum_{c\in[\ell]} \gamma_c = 0$, and append to A the constraints that sum the entries in the message encoded in u'.

However, as before, this will yield less than optimal soundness guarantee. Instead, we consider the following approach that provides the same soundness guarantee as the original (non-affine version of the) test. We apply the algorithm Test-Linear-Constraints-IRS to L^{m+1} where $L = \mathsf{RS}_{\mathbb{F},n,k,\eta}$, with an extended oracle U' whose first m rows contain U and whose last row is u' where additionally u' encodes a message $(\gamma_1, \ldots, \gamma_\ell)$ such that $\sum_{c\in[\ell]} \gamma_c = 0$. Letting $q(\cdot) = \sum_{i=1}^{m} r_i(\cdot) \cdot p_i(\cdot) + r_{\text{Blind}}(\cdot)$ be the polynomial obtained by $\mathcal{V}$ where $r_{\text{Blind}}(\cdot)$ is a polynomial (of degree $< k + \ell - 1$) corresponding to u', we can show that the soundness of the resulting scheme will be the same as for Lemma 4.8. This "affine" variant of **Test-Linear-Constraints** is described and analyzed in Appendix B.4.

4.6.3 ZK Testing of Quadratic Constraints over Interleaved Reed-Solomon Codes

Finally, we modify the quadratic constraint testing procedure in the same way as we modified the linear constraint testing. Concretely, we apply the algorithm **Test-Quadratic-Constraint** to L^{3m+1} where $L = \mathsf{RS}_{\mathbb{F},n,k,\eta}$, with an extended oracle U' whose first $3m$ rows contain U^x, U^y, U^z and whose last row is u' where additionally u' encodes a message 0^ℓ. Letting $p_0(\cdot) = \sum_{i=1}^{m} r_i \cdot p_i(\cdot) + r_{\text{Blind}}(\cdot)$ be the polynomial obtained by $\mathcal{V}$ where $r_{\text{Blind}}(\cdot)$ is a polynomial (of degree $< 2k - 1$) corresponding to u', we can show that the soundness of the resulting scheme will be the same as for Lemma 4.10. This "affine" variant of **Test-Quadratic-Constraints** is described and analyzed in Appendix B.5.

4.7 The Final ZKIPCP

In this section provide a self contained description of the final ZKIPCP protocol, combining all of the previous sub-protocols. In this section, we provide our ZKIPCP for arithmetic circuits over a large field $\mathbb{F}$. On a high-level, the protocol is essentially the IPCP construction from Section 4.4 with the exception that we replace all the tests with the generalized affine version (with repetitions).

Protocol ZKIPCP$(\mathrm{C}, \mathbb{F})$**.**

- **Input:** The prover $\mathcal{P}$ and the verifier $\mathcal{V}$ share a common input arithmetic circuit $\mathrm{C} : \mathbb{F}^{n_i} \to \mathbb{F}$ and input statement x. $\mathcal{P}$ additionally has input $\overline{\alpha} = (\alpha_1, \ldots, \alpha_{n_i})$ such that $\mathrm{C}(\overline{\alpha}) = 1$.
- **Oracle π:** Let m, ℓ be integers such that $m \cdot \ell > n_i + s$ where s is the number of gates in the circuit. Then $\mathcal{P}$ generates an extended witness $w \in \mathbb{F}^{m\ell}$ where the first $n_i + s$ entries of w are

$(\alpha_1, \ldots, \alpha_{n_i}, \beta_1, \ldots, \beta_s)$ where β_i is the output of the i^{th} gate when evaluating $\mathrm{C}(\overline{\alpha})$. $\mathcal{P}$ constructs vectors x, y and z in $\mathbb{F}^{m\ell}$ where the j^{th} entry of x, y and z contains the values β_a, β_b, and β_c corresponding to the j^{th} multiplication gate in w. $\mathcal{P}$ and $\mathcal{V}$ construct matrices P_x, P_y and P_z in $\mathbb{F}^{m\ell \times m\ell}$ such that

$$x = P_x w, y = P_y w, z = P_z w.$$

Finally, it constructs matrix $P_{\mathbf{add}} \in \mathbb{F}^{m\ell \times m\ell}$ such that the j^{th} row of $P_{\mathbf{add}} w$ equals $\beta_a + \beta_b - \beta_c$ where β_a, β_b, and β_c correspond to the j^{th} addition gate of the circuit in w. The linear constraints can be summarized as one large matrix as before.

$$A = \left[\begin{array}{c|c} I_{3m\ell \times 3m\ell} & -P \\ \hline \mathbf{0}_{m\ell \times 3m\ell} & P_{\mathbf{add}} \end{array}\right], P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}, b = \mathbf{0}^{4m\ell}$$

The prover samples random codewords $U^w, U^x, U^y, U^z \in L^m$ where $L = \mathsf{RS}_{\mathbb{F},n,k,\eta}$ subject to $w = \mathsf{Decode}_\zeta(U^w), x = \mathsf{Decode}_\zeta(U^x), y = \mathsf{Decode}_\zeta(U^y), z = \mathsf{Decode}_\zeta(U^z)$ where $\zeta = (\zeta_1, \ldots, \zeta_\ell)$ is a sequence of distinct elements disjoint from $(\eta_1, \ldots, \eta_n)$. Let $u_h^0, u_h^{\mathbf{add}}$ be auxiliary rows sampled randomly from L for every $h \in [\sigma]$ where each of $u_h^{\mathbf{add}}$ encodes an independently sampled random ℓ messages $(\gamma_1, \ldots, \gamma_\ell)$ subject to $\sum_{c \in [\ell]} \gamma_c = 0$ and u_h^0 encodes 0^ℓ. $\mathcal{P}$ sets the oracle as $U \in L^{4m}$ which is set as the vertical juxtaposition of the matrices $U^w, U^x, U^y, U^z \in L^m$.

- **The interactive protocol:**

  1. For every $h \in [\sigma]$, $\mathcal{V}$ picks the random elements $r_h \in \mathbb{F}^{4m}, r_h^{\mathbf{add}} \in \mathbb{F}^{4m\ell}$ and $r_h^q \in \mathbb{F}^m$ and sends them to $\mathcal{P}$.
  2. For every $h \in [\sigma]$, $\mathcal{P}$ responds with
     - (Interleaved Reed-Solomon Testing)

       $$v_h = (r_h)^T U + u'_h \in \mathbb{F}^n,$$

     - (Linear Constraints Testing) Polynomial $q_h^{\mathbf{add}}(\cdot)$ of degree $< k + \ell - 1$ where

       $$q_h^{\mathbf{add}}(\cdot) = r_{\mathrm{Blind},h}^{\mathbf{add}}(\cdot) + \sum_{i=1}^{m} r_{h,i}^{\mathbf{add}}(\cdot) \cdot p_i(\cdot),$$

       such that
       - p_i is the polynomial of degree $< k$ corresponding to row i of U^w,
       - $r_{h,i}^{\mathbf{add}}(\cdot)$ is the unique polynomial of degree $< \ell$ such that $r_{h,i}^{\mathbf{add}}(\zeta_c) = ((r_h^{\mathbf{add}})^T P)_{ic}$ for every $c \in [\ell]$, and
       - $r_{\mathrm{Blind},h}^{\mathbf{add}}(\cdot)$ is the polynomial of degree $< k + \ell - 1$ corresponding to $u_h^{\mathbf{add}}$.
     - (Quadratic Constraints Testing)

       $$p_{0,h}(\cdot) = r_{\mathrm{Blind},h}^0(\cdot) + \sum_{i=1}^{m} \left(r_h^q\right)_i \cdot (p_i^x(\cdot) \cdot p_i^y(\cdot) - p_i^z(\cdot))$$

       where for $a \in \{x, y, z\}$,
       - p_i^a is the polynomial of degree $< k$ corresponding to row i of U^a,

* $r^0_{\mathrm{Blind},h}$ is the polynomial of degree $< 2k-1$ corresponding u^0_h.

3. $\mathcal{V}$ picks a random index set $Q \subset [n]$ of size t, and queries $U[j]$ that is the vertical juxtaposition of $U^x_h[j], U^y_h[j], U^z_h[j], U^w_h[j], u^{\mathbf{add}}_h[j], u'_h[j]$, $j \in Q$ and accepts if the following conditions hold for every $h \in [\sigma]$:
   - For every $j \in Q$ we have
$$\sum_{i=1}^{4m} r_h[j] \cdot U_{i,j} + u'_h[j] = v_h[j],$$
   - $\sum_{c\in[\ell]} q^{\mathbf{add}}_h(\zeta_c) = 0$ and for every $j \in Q$ we have
$$u^{\mathbf{add}}_h[j] + \sum_{i=1}^{4m} r^{\mathbf{add}}_{h,i}(\eta_j) \cdot U_{i,j} = q^{\mathbf{add}}_h(\eta_j),$$
   - $p_{0,h}(\zeta_c) = 0$ for every $c \in [\ell]$ and for every $j \in Q$,
$$u^0_h[j] + \sum_{i=1}^{m} (r^q_h)_i \cdot \left[U^x_{i,j} \cdot U^y_{i,j} - U^z_{i,j}\right] = p_{0,h}(\eta_j).$$

The completeness of our ZKIPCP follows from the next lemma.

Lemma 4.13. *If $U^w, U^x, U^y, U^z \in L^m$ encode vectors $w, x, y, z \in \mathbb{F}^{m\ell}$ satisfying*
$$x = P_x w,\ y = P_y w,\ z = P_z w,\ x \odot y + (-\mathbf{1})^{m\ell} \odot z = \mathbf{0}^{m\ell},\ P_{\mathbf{add}} w = \mathbf{0}^{m\ell}$$
and $\mathcal{P}$ is honest, $\mathcal{V}$ always accepts.

Next, soundness is argued by the following lemma.

Lemma 4.14. *Let e be a positive integer such that $e < d/4$. Suppose that there exists no $\overline{\alpha}$ such that $\mathrm{C}(\overline{\alpha}) = 1$. Then, for any maliciously formed oracle U^* and any malicious prover strategy, the verifier rejects except with at most $(d+2)/|\mathbb{F}|^\sigma + (1-e/n)^t + 2((e+2k)/n)^t$ probability.*

The proofs of the preceding two lemmas follow analogously to the proofs of Lemma 4.11 and Lemma 4.12. The next lemma establishes the honest verifier zero-knowledge property.

Lemma 4.15. *If $k > \ell + t$, $\langle \mathcal{P}, \mathcal{V}\rangle$ is an* (honest verifier, perfect) zero-knowledge IPCP.

Proof: To demonstrate zero-knowledge against honest verifier, we need to provide a simulator S that can given the randomness provided by the honest verifier $\mathcal{V}$, be able to generate a transcript. For every $h \in [\sigma]$, the simulator first generates:

- random polynomial $q^{\mathbf{add}}_h$ of degree $< k+\ell-1$ such that $\sum_{c\in[\ell]} q^{\mathbf{add}}_h(\zeta_c) = 0$.
- random polynomial $p_{0,h}$ of degree $< 2k-1$ such that $p_{0,h}(\zeta_c) = 0$ for every $c \in [\ell]$.
- random vector $v_h \in \mathbb{F}^n$.

Next, it samples random elements from $\mathbb{F}$ for $U^x_h[j], U^y_h[j], U^z_h[j], U^w_h[j]$, for every $j \in Q$. Finally, given the random challenges from $\mathcal{V}$, it sets $u'_h[j], u^{\mathbf{add}}_h[j], u^0_h[j]$ as follows:

- $u'_h[j] = \sum_{i=1}^{4m} r_h[j] \cdot U_{i,j} - v_h[j]$.
- $u_h^{\mathbf{add}}[j] = \sum_{i=1}^{4m} r_{h,i}^{\mathbf{add}}(\eta_j) \cdot U_{i,j} - q_h^{\mathbf{add}}(\eta_j)$
- $u_h^0[j] = \sum_{i=1}^{m} (r_h^q)_i \cdot \left[U_{i,j}^x \cdot U_{i,j}^y - U_{i,j}^z\right] - p_{0,h}(\eta_j)$.

Our simulation achieves perfect zero knowledge. This follows from the fact that in an honest execution with the prover $\mathcal{P}$, the distribution of $\{U_h^x[j], U_h^y[j], U_h^z[j], U_h^w[j]\}_{j \in Q}$ are uniformly distributed and given that $u'_h, u_h^{\mathbf{add}}, u_h^x, u_h^0$ are uniformly chosen, the polynomials $q_h^{\mathbf{add}}, p_{0,h}$ and the vector v_h are uniformly distributed in their respective spaces. □

The following theorem follows from the construction described above and the preceding Lemmas.

Theorem 4.7. *Fix parameters n, m, ℓ, k, t, e such that $e < (n-k)/4$. Let $\mathrm{C} : \mathbb{F}^{n_i} \to \mathbb{F}$ be an arithmetic circuit of size s, where $|\mathbb{F}| \geq \ell + n$, $m \cdot \ell > n_i + s$ and $k > \ell + t$. Then protocol* ZKIPCP$(\mathrm{C}, \mathbb{F})$ *satisfies the following:*

- COMPLETENESS: *If $\overline{\alpha}$ is such that $\mathrm{C}(\overline{\alpha}) = 1$ and oracle π is generated honestly as described in the protocol, then*
  $\Pr[(\mathcal{P}(\mathrm{C}, w), \mathcal{V}^{\pi}(\mathrm{C})) = 1] = 1$.
- SOUNDNESS: *If there is no $\overline{\alpha}$ is such that $\mathrm{C}(\overline{\alpha}) = 1$, then for every (unbounded) prover strategy $\mathcal{P}^*$ and every $\tilde{\pi} \in \mathbb{F}^{4mn}$,*
  $Pr[(\mathcal{P}^*, \mathcal{V}^{\tilde{\pi}}(x)) = 1] \leq (d+2)/|\mathbb{F}|^{\sigma} + (1 - e/n)^t + 2((e+2k)/n)^t$.
- ZERO KNOWLEDGE: *For every adversary verifier $\mathcal{V}^*$, there exists a simulator $\mathcal{S}$ such that the output of $S^{\mathcal{V}^*}(\mathrm{C})$ is distributed identically to the view of $\mathcal{V}$ in the $(\mathcal{P}(\mathrm{C}, w), \mathcal{V}^{\pi}(\mathrm{C}))$.*
- COMPLEXITY: *The number of field $\mathbb{F}$ operations performed is* poly$(|\mathrm{C}|, n)$. *The number of field elements communicated by $\mathcal{P}$ to $\mathcal{V}$ is $\sigma \cdot n + \sigma \cdot (k + \ell - 1) + \sigma \cdot (2 \cdot k - 1)$ whereas $\mathcal{V}$ reads t symbols from $\mathbb{F}^{4m+5\sigma}$.*

5 From ZKIPCP to ZK

In this section we describe variants of known transformations from (sublinear) zero-knowledge PCP to (sublinear) zero-knowledge argument. The latter can either be interactive using collision-resistant hash (CRH) functions, or non-interactive in the random oracle model or based on CRH (following the Fiat-Shamir heuristic).

5.1 The Interactive Variant

General transformations from (non-interactive) ZKPCP to (interactive) ZK arguments that make a black-box use of collision-resistant hash functions were given in [IMS12, IW14]. Here we address the more general case of ZKIPCP, where in addition to the proof oracle there is additional interaction between the prover and the verifier. Namely, using the ZKIPCP, an honest-verifier ZK protocol proceeds as follows. The prover commits to each entry of the proof oracle using a statistically hiding commitment scheme and then compresses the commitment using a Merkle hash tree (cf. Section 2.1). Note that both steps can be realized by making a black-box use of any family $\mathcal{H}$ of collision-resistant hash functions. The rest of the ZK protocol mimics the ZKIPCP, where the prover opens the committed values that correspond to the verifier's queries.

Malicious verifiers can be handled using standard techniques (see e.g., Section 6.2 in the full version of [IMS12]). The communication complexity of the ZK argument includes the communication complexity of the ZKIPCP protocol and communication resulting from committing the oracle Π and decommitting to the queries Q.

5.2 The Non-Interactive Variant

It is possible to directly compile our previous protocol into a non-interactive protocol using a random oracle, where the verifier's messages are emulated by applying the random oracle on the partial transcript in each round following the Fiat-Shamir transform [FS86]. A formal description and analysis of this transformation is presented in [BCS16] for interactive oracle proofs (IOP) model which generalizes (public-coin) IPCP.

In slight more detail, in this transformation the prover uses the random oracle to generate the verifier's messages and complete the execution (computing its own messages) based on the emulated verifier's messages, where instead of using an oracle, the prover commits to its proof and messages using Merkle hash trees. Completeness follows directly. If we start with an IOP that additionally is zero-knowledge (ZKIPCP in our case), [BCS16] show that this transformation preserves (statistical) zero-knowledge property. Namely, the resulting protocol can be proven to be zero-knowledge in the random-oracle model.

In [BCS16], the soundness of the transformed protocol is shown to essentially match the soundness of the original protocol up to an additive term that roughly depends on the product of q^2 and $2^{-\kappa}$ where q is an upper bound on the number of queries made to the random oracle by a malicious prover and κ is the output length of the random oracle. More precisely, [BCS16] relates the soundness of the transformed protocol to the state restoration soundness of the underlying IPCP and the collision-probability of queries to the random oracle. State-restoration soundness refers to the soundness of the IOP protocol against cheating prover strategies that may rewind the verifier back to any previously seen state, where every new continuation from a state invokes the next-message function of the verifier with fresh randomness. In [BCS16], they show that for any (IOP) the state-restoration soundness of an IOP protocol is bounded by $\binom{T}{k(x)} \cdot \epsilon(x)$ and the soundness of the transformed protocol is $\binom{T}{k(x)} \cdot \epsilon(x) + O(T^2 \cdot 2^{-\kappa})$ where T bounds the number of queries made by cheating provers to the random oracle, $k(x)$ is the round complexity of the IOP and $\epsilon(x)$ is the (standard) soundness of the IOP.

Next, we tighten the analysis presented in [BCS16] for the particular ZKIPCP constructed in Section 4.7 and show that the soundness of the transformed protocol is $T \cdot \epsilon(x) + O(T^2 \cdot 2^{-\kappa})$ where $\epsilon(x)$ is the soundness of the ZKIPCP, T bounds the number of queries made by cheating prover to the random oracle and κ is the output length of the random oracle.

In [CCH+19], Canetti et al., introduced the notion of round-by-round soundness, to provide a fine-grained analysis of the Fiat-Shamir heuristic. We fist repeat (verbatim) the definition of round-by-round soundness from [CCH+19] for completeness.

Definition 5.1. *A $2r$-round protocol Π has round-by-round soundness error $\epsilon(\cdot)$ if there exists a (possibly inefficient) mapping* State *from the tuple (x, τ) where x is the instance and τ a partial transcript of interaction using Π to $\{\mathsf{accept}, \mathsf{reject}\}$ such that the following hold:*

1. *If $x \notin L$, then $\mathsf{State}(x, \emptyset) = \mathsf{reject}$, where $\emptyset$ denotes the empty transcript.*
2. *If $\mathsf{State}(x, \tau) = \mathsf{reject}$ for a partial transcript up to $2i$-rounds, then for every prover message α, it holds that*

$$\Pr[\beta \leftarrow V(x, (\tau, \alpha)) : \mathsf{State}(x, (\tau, \alpha, \beta)) = \mathsf{accept}] < \epsilon(|x|, |\tau|).$$

3. *For any full transcript τ, if* $\mathsf{State}(x,\tau) = \mathsf{reject}$*, then* $V(x,\tau) = 0$.

Next, we analyze the transformed protocol in the random-oracle model. Suppose the prover makes at most T queries, then the probability it finds a collision is bounded by $T^2 2^{-\kappa}$. Next, we analyze the round-by-round soundness of the compiled interactive ZK argument. By our preceding analysis, the value of $\epsilon(|x|,|\tau|)$ for a partial transcript τ till the end of the first round will be at most $(d+2)/|\mathbb{F}|^\sigma$, and for a transcript till the end of the third round will be at most $(1-e/n)^t + 2((e+2k)/n)^t$. Finally, we apply the Fiat-Shamir transformation where the prover generates the verifier's message by applying the random oracle on the partial transcript to obtain the randomness for the verifier. To argue the soundness of the transformed protocol, we observe that the adversary succeeds only if $\mathsf{State}(x,(\tau,\alpha,\beta)) = \mathsf{accept}$. Suppose that, the adversary makes T_1 queries with τ as a partial transcript at the end of the first round and T_2 queries for partial transcripts at the end of the third round (where the prover was not already on a good partial transcript at the end of the first round), then, the probability the adversary succeeds is bounded by

$$\begin{aligned} T_1 \cdot (d+2)/|\mathbb{F}|^\sigma + T_2 \cdot \left((1-e/n)^t + 2((e+2k)/n)^t\right) + T^2 2^{-\kappa} \\ \leq T \cdot \left((d+2)/|\mathbb{F}|^\sigma + (1-e/n)^t + 2((e+2k)/n)^t\right) + T^2 2^{-\kappa} \end{aligned}$$

For concrete security, we can conclude that for the non-interactive protocol to have κ-bit security (i.e. $T \cdot \epsilon_{NI} = 2^{-\kappa}$), the (statisticaly) soundness of the interactive protocol can be set to $\epsilon_{Int} = 2^{-\kappa}$ where the output length of the random-oracle is set to $2 \cdot \kappa$-bits.

5.3 Sublinear Zero-Knowledge Argument

In this section, we describe how to set the parameters of our zero-knowledge argument protocol to obtain communication that is sublinear in the circuit size. We consider first an arithmetic circuit over a large field $\mathbb{F}$. Following our transformation, the communication complexity of the zero-knowledge protocol that is compiled based on our ZKIPCP is

$$\left[\underbrace{k\cdot\sigma}_{\text{code test}} + \underbrace{(k+\ell-1)\cdot\sigma}_{\text{linear test}} + \underbrace{(2\cdot k-1)\cdot\sigma}_{\text{quadratic test}} + \underbrace{t\cdot(4\cdot m+3\cdot\sigma)}_{\text{parties' views}}\right]\cdot\lceil\log|\mathbb{F}|\rceil + \underbrace{t\cdot\lceil\log n\rceil\cdot h}_{\text{Merkle tree decommitments}}$$

where h denotes the output length of the random-oracle (i.e. hash-function) and the view includes $4m$ one for each row of U_w, U_x, U_y, U_z and $3\cdot\sigma$ that correspond to the rows of the blinding polynomials. An optimal set of parameters that minimize the communication complexity for arithmetic circuits over $\mathbb{F}$ can be obtained by having $e = k$ which implies $n = 3\cdot k$. For these parameters, we have $e < d/2$ and the soundness simplifies to $3\cdot(2/3)^t + (n+4)/|\mathbb{F}|^\sigma$ where σ is the number of times we repeat the test. For soundness to be at most $2^{-\kappa}$ we need $\sigma \approx \lambda/\log_2(|\mathbb{F}|)$ and $t = \lambda/\log_2(2/3)$. With these parameters set, communication complexity is then minimized when $\ell = \sqrt{\frac{|\mathrm{C}|\cdot\min(|\mathbb{F}|,\kappa)}{5\cdot\log_2(3/2)}}$. Then, the communication complexity is $\kappa\cdot|\mathbb{F}|\cdot\sqrt{\frac{5\cdot|\mathrm{C}|}{\log_2(3/2)\cdot\min(|\mathbb{F}|,\kappa)}}$ which simplifies to $O(|\mathbb{F}|\cdot\sqrt{|\mathrm{C}|\cdot\kappa})$ bits for large fields and $O(\kappa\cdot\sqrt{|\mathrm{C}|\cdot|\mathbb{F}|})$ bits for small fields (i.e. for $|\mathbb{F}| < \kappa$). We remark that when identifying concrete parameters we additionally need k to be a power of 2 (to use Cooley-Tukey style FFT for Reed-Solomon encoding).

For Boolean circuits, we can either embed the computation in a prime finite field or Galois $\mathbb{GF}(2^n)$ field. In either case, we need the field size to be at least as large as $n+\ell$. In the prime field, both XOR and AND costs one arithmetic multiplication while in the Galois field XOR and AND map into addition and multiplication in the field. Depending on which embedding is chosen, we need to set the circuit size C and field $\mathbb{F}$ appropriately and use the preceding computation.

5.4 Multi-Instance Amortization

If we want to prove that $\mathrm{C}(x_i, \cdot)$ is satisfiable for N public inputs x_i, we can simplify our ZKIPCP construction as follows. The prover first computes the combined witness $w = w_1, \ldots, w_N$ that is comprised of N witnesses, each is computed as in the single instance case. Next, it arranges the witnesses in blocks of size $\ell = N$, where block j contains the j^{th} bits of each of the N witnesses. The number of blocks in the extended witness equals the size of the witness of a single instance, which is $m = |w_i| = O(|\mathrm{C}|)$. The prover then encodes the blocks of messages into $U \in L^m$.

For moderately large N, the multi-instance variant provides significant savings in both computational and communication costs. This is because we do not need to rearrange the wire values as we do in the single instance case. The total asymptotic communication complexity for sufficiently large fields then becomes $O((N + \kappa \cdot |\mathrm{C}|) \cdot |\mathbb{F}|)$.

6 Implementation and Experimental Results

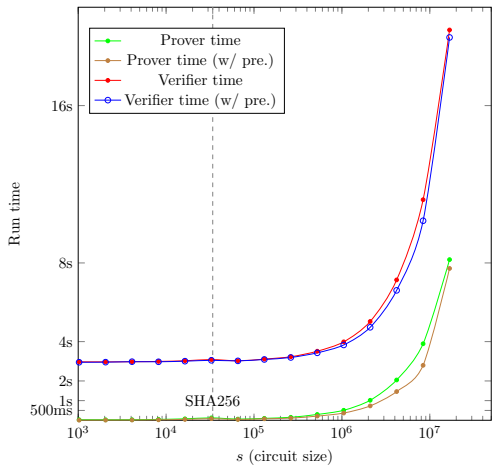

Figure 5: Prover and verifier running times for verifying a single instance of different circuit sizes.

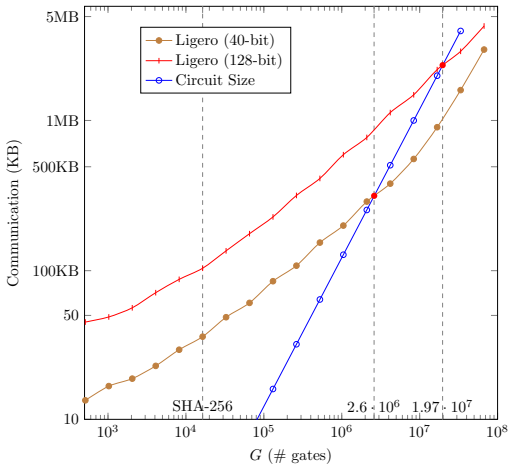

Figure 6: Proof lengths for proving a single instance of difference circuit sizes.

We implemented our protocol in NFLlib library for the finite field operations. To pick the evaluation points, we chose a prime that had sufficiently large power of two roots of unity and set η_i and ζ_j values to be roots of unity. This enabled us to perform interpolation and evaluation using inverse FFT and FFT operations. We ran our experiments on Amazon EC2 `c6i.32xlarge` with the Intel Xeon CPU 3.5 GHz, 128 cores, 256 GB RAM. For our collision resistant hash function we used SHA256.

We instantiated our interactive variant with soundness 2^{-128} in the random oracle model and implemented the non-interactive variant by applying the Fiat-Shamir transform where we used SHA256 to implement the random oracle. We considered arithmetic circuits over a 30-bit prime field. We also refined our proof of soundness. In Sections 4.2 and 4.3, we analyzed the linear and quadratic tests independently. If we

analyze it in conjunction with the IRS code test, we can improve some of the terms. We present this analysis in Appendix C.

In Figure 5, we compare the prover and verifier running times for verifying circuits of sizes varying from 1000 gates to 10^7 gates. These are randomly generated arithmetic circuits over a 30-bit prime field.[4] Specifically, for a circuit of size s, we sampled a circuit with s multiplication (fan-in 2) gates and s addition gates (fan-in 2). The computational complexity of both the prover and the verifier in the single instance setting is proportional to $O(s \log s)$ field operations when we optimize the packing factor to minimize the proof length. Furthermore, the optimal field size can be asymptotically shown to be $O(\log s)$ resulting in overall computational complexity of $O(s \log^2 s)$. We remark here that if we make uniformity assumptions on the circuit, then the verifier's computational complexity becomes sublinear in the circuit size. In fact, in the multi-instance setting which can be seen as a uniformity assumption, we achieve succinct verification, i.e. our verifier's complexity is smaller than the circuit size.

We observe that for small to medium circuit sizes (up to 1 million gates) the bulk of the time spent by the prover is in reading the circuit corresponding to the NP statement, where for a fixed circuit this time can be spent in a pre-processing phase. We provide the prover and verifier's times excluding the pre-processing step. As the circuit size increases, the prover and verifier's efficiency improves, where at 10^7 gates they run at 500ns per gate excluding pre-processing and 1.1μs per gate end-to-end.

In Figure 6, we provide the communication complexity in kilobytes (KB) of our zero-knowledge argument. We plot two instantiations of our protocol. We provide the communication cost for the vanilla Ligero instantiated at 128-bit and 40-bit security. We also plot a line measuring the circuit size in bits and identify the point at which the Ligero system is strictly sublinear in the circuit size. For 40-bit security this is around 2.6 million gates and for the 128-bit security it is around 20 million gates.

Comparing with IOP-based ZK-SNARKs We now provide a comparison with IOP-based ZK-SNARKs that are plausibly post-quantum secure. In Table 2 we compare our proof lengths with the `libiop` implementation of Ligero, Aurora, Brakedown and Shockwave. We obtain the data for these systems from [GLS+21] where they set all these schemes at 128-bit security. The new analysis of the Ligero system yields proof lengths that is significantly better than the other IOP-based implementations. We note here that while the data from [GLS+21] is for a random R1CS statement over 128-bit/256-bit, we consider a random arithmetic circuit of similar size over a 30-bit prime. In Table 3, we compare the running times of the prover and verifier against [GLS+21]. Our prover times are better than Brakedown and Shockwave for larger circuits and competitive for small circuits. Our end-to-end verification times are competitive, but if we exclude the circuit setup time, we get better efficiency. We note here, however, this comparison is not apples-to-apples as it was run on different machines but comparable architectures.

7 Related Work with Open Problems

Below we provide a list of open problems regarding IOPs.

Constant computational overhead. An important theoretical (and in some cases practical) question in the design of succinct arguments is understanding the asymptotic *computational overhead* of the prover. This overhead is defined as the ratio between the running time (or circuit size) of the implementation of the prover and that of verifying the witness without any security requirements. The question of constant-overhead cryptography was initiated in the work of Ishai et al. [IKOS08] and subsequent works [AM17,

[4] Note that our proof length and computation times are not influenced by the circuit topology and only depend on the witness size which in turn depends only on the number of gates.

Circuit Size	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}	2^{17}	2^{18}	2^{19}	2^{20}
Ligero [AHIV17]	546	628	1,076	1,169	2,100	3,169	5,788	5,662	10,527	10,736	19,828
Aurora [BCR+19]	447	510	610	717	810	931	1,069	1,179	1,315	1,473	1,603
Brakedown(128-bit) [GLS+21]	1,279	1,597	1,974	2,200	2,710	3,165	3,926	4,824	6,122	7,899	10,230
Shockwave(128-bit) [GLS+21]	72	95	122	160	210	284	386	523	721	990	1,384
Ligero-128 (here)	48	56	71	87	103	135	177	229	320	417	602

Table 2: Comparison of proof length (in KB) between different IOP-based ZK-SNARKs.

Prover Time

Circuit Size	2^{20}	2^{21}	2^{22}	2^{23}
Brakedown (128-bit)	3	6	13	25
Shockwave (128-bit)	4	9	17	36
Ligero-128 (here)	4	5	7	11

Verifier Time

Circuit Size	2^{20}	2^{21}	2^{22}	2^{23}
Brakedown (128-bit)	0.7	1.1	2.1	3.8
Shockwave (128-bit)	0.5	0.9	1.7	3.5
Ligero-128 (here)	3.8	4.7	6.6	10

Table 3: Comparison of running times (in seconds) of prover and verifier between different IOP-based ZK-SNARKs.

AHI+17, BIO14, BIP+18], where is was shown how to construct several primitives with constant overhead under plausible cryptographic assumptions.

The prover overhead of the Ligero system is $O(\log |\mathrm{C}|)$ for an arithmetic circuit C stemming from computing the RS encodings, or $\mathsf{polylog}(|\mathrm{C}|)$ in the Boolean case. A recent line of works has focussed on designing succinct proof systems for non-uniform arithmetic circuits (modeled via so-called rank-1 constraint systems (R1CS)) over a large finite field where the size of the prover is linear in the size of the circuit [BCG+17, BBB+18, XZZ+19b, BCG20, Set20, SL20, KMP20, ZXZS20, GLS+21, LSTW21, ZLW+21b, BCL22]. Specific to IOPs, the works of [BCG20, GLS+21, BCL22] construct a linear-time IOP, i.e. the overhead of the prover is constant for arithmetic circuits over a large finite field. More recently, the works of [RR22, HR22] construct, for the first time, a linear time prover for Boolean circuits, but, for restricted settings. Specifically, Ron-Zewi and Rothblum [RR22] build a linear-time IOP with *constant-soundness* error and Holmgren and Rothblum [HR22] build a linear-time IOP for restricted classes of circuits, including *batch Boolean statements*, with $2^{-\lambda}$ soundness error and $\mathsf{polylog}(\lambda)$ overhead. The state-of-the-art leaves the following fundamental question regarding linear-time IOPs open.

Open Problem 1. *Can we construct (sublinear-query) IOPs for Boolean circuits with constant computational overhead and negligible soundness error?*

A bit more concretely, we would ideally like to have an IOP for proving the satisfiability of a Boolean circuit of size s with $2^{-\lambda}$ soundness error, where the verifier makes $\lambda \cdot \mathsf{polylog}(s)$ queries and where the prover is implemented by a Boolean circuit of size $O(s) + \mathsf{poly}(\lambda, \log s)$.

Complexity preserving constructions. As mentioned above, recent works have shown how to improve the prover's computational complexity to essentially linear in the time taken to compute the underlying relation (for an NP-language). However, these works come with a steep price in terms of *space*, namely, for computations that take time T and space S, the space complexity of the prover is $\Omega(T)$. Notably, only a few works provide time and space efficient constructions that we discuss next. This fact turns out to be a major bottleneck in scaling up zero-knowledge proofs to larger and larger computations. To make the context precise, we focus on the task of proving that a non-deterministic RAM machine M accepts a particular

instance x, i.e. uniform non-deterministic computations. The goal here is that if M accepts/rejects x in time T and space S then the resulting ZK proof system preserves these complexities on the prover's side while being polylogarithmic in T (i.e. succinct) or even sublinear on the verifier's side.

When considering designated verifier ZK-SNARKs, complexity preserving solutions (i.e. poly-logarithmic overhead in space and time) have been constructed by Bitansky and Chiesa [BC12] and by Holmgren and Rothblum [HR18] in the non-interactive setting. The work of Ephraim et al. [EFKP20] shows that, assuming the existence of standard (circuit) SNARKs and collision-resistant hash functions (CRHF), one can construct a non-interactive succinct argument of knowledge (i.e. SNARK) for parallel RAM computations where the prover's complexities are preserved whereas the verifier requires polylogarithmic in T time and space, and the underlying CRHF and SNARK are used in a non-black-box manner. Publicly-verifiable ZK-SNARKs with similar overheads can be accomplished via recursive composition [BGH19, COS20, BCMS20]. Nevertheless, these constructions have significant overheads as they typically rely on non-black-box usage of the underlying primitives.

More recently, two works by Block et al. [BHR+20, BHR+21] designed the first black-box construction of a ZK-SNARKs with polylogarithmic overhead in space and time based on "more standard" assumptions. The first work assumes hardness of discrete logarithm in prime-order groups and relies on the random oracle to construct a public-coin zero-knowledge argument where the proof length is $\mathsf{polylog}(T)$, the prover is complexity preserving and the verifier runtime is $T \cdot \mathsf{polylog}(T)$ while using $\mathsf{polylog}(T)$ space. The second work improves the verifier's runtime from $T \cdot \mathsf{polylog}(T)$ to $n \cdot \mathsf{polylog}(T)$, where n is the input length, under hardness assumptions on hidden order groups. We note that these works make extensive use of public-key operations - e.g., the prover needs to compute $\Omega(T)$ exponentiations, where public-key operations are typically orders of magnitude more expensive than symmetric key operations. The prior works leave the following question open.

Open Problem 2. *Can we construct complexity-preserving IOPs?*

More precisely, consider the universal relation $R_{\mathcal{U}}$ of instance-witness pairs (y, w), where $y = (M, x, T)$, $|w| \leq T$, and M is an abstract RAM machine, such that M accepts (x, w) after at most t steps. Let $L_{\mathcal{U}}$ be the corresponding language. We would like to have an IOP for proving membership in $L_{\mathcal{U}}$ with $2^{-\kappa}$ soundness error such that for a polynomial p and any instance (M, x, T) where M uses space S we have:

- the prover runs in time $(|M| + |x| + T) \cdot p(\kappa + \log T)$,
- the prover P runs in space $(|M| + |x| + S) \cdot p(\kappa + \log T)$,
- the verifier V runs in time $(|M| + |x| + \log T) \cdot p(\kappa + \log T)$,
- the total bits communicated to V is $p(\kappa + \log T)$.

We remark that if we only insisted on space-preserving IOPs (i.e. relax the time requirement) we can essentially rely on the same constructions of [BCR+19, BFH+20] by observing that a Reed-Solomon encoding of data of size T can be computed in time polynomial in T with multiple passes on the input using space $\mathsf{polylog}(T)$ (which is the bottleneck in terms of space for these constructions). We also highlight that the above question is open even if we relax constant-overhead to polylogarithmic overhead. In recent work, Bhadauria et al. made progress in answering the question where they construct a complexity-preserving IOP that is *somewhat succinct* and the overhead is polylogarithmic [BBHV22]. Namely, for every NP relation that can be verified in time T and space S by a RAM program, they constructed a complexity-preserving (ZK)IOP, where the prover runs in time $\widetilde{O}(T)$ and space $\widetilde{O}(S)$, the verifier runs in time $\widetilde{O}(T/S + S)$ and space $\widetilde{O}(1)$ and the query-complexity is $\widetilde{O}(T/S)$, where $\widetilde{O}()$ ignores polynomial factors in $\log T$ and κ.

Minimal assumptions for sublinear arguments. Since the original work of Kilian [Kil92] we have known that (public-coin) sublinear arguments can be constructed from symmetric-key primitives. In particular, it can be constructed from collision-resistant hash-functions. However, the question of what are the minimal assumptions to design sublinear arguments is still open. We do not know if one-way functions are sufficient or even necessary. The recent work of Pass and Venkitasubramaniam [PV20] shows that if public-coin sublinear arguments exist for a language L, then either (a slight variant of standard) one-way functions exist or there exists a two-round sublinear argument for the same language. This leaves the following fundamental question open.

Open Problem 3. *What are the minimal assumptions to construct sublinear arguments for all of* NP*?*

8 Conclusions

We designed and implemented a zero-knowledge argument for NP that simultaneously offers good concrete efficiency and sublinear communication in the circuit size. As the computational complexity of our protocol is dominated by polynomial evaluations and interpolations, we can rely on efficient FFT implementations for minimizing its computational cost. In the following we mention some additional optimizations that we have not fully explored. The current implementation relies on prime fields. This allows us to optimize arithmetics over integers by considering a sufficiently large prime. Moreover, we recall that for the Boolean case the witness includes two bits per gate for both XOR and AND gates. If we instead rely on characteristic 2 fields, then the witness size will require three bits per AND gate and 0 bits for XOR gates. Hence there is a tradeoff in choosing between the two options. It is also unclear how the FFT algorithms compare for characteristic 2 and prime fields, though fast implementations for the characteristic 2 case are known [GM10, BHST16] and used in designing ZKSNARKs based on IOPs [BBHR19].

The verification of our zero-knowledge argument needs to evaluate a polynomial on a subset of the points in the domain. We currently implement this by having the verifier evaluate the polynomial on the entire domain via FFT and extract the points in this subset. Improving this will improve the verifier's efficiency. Relying on GPU for FFT computations can also bring significant savings. Finally, one can exploit a repetitive circuit structure ("uniformity") to reduce verification time. We currently only take advantage of this for reducing the amortized cost of verifying multiple evaluations of the same circuit.

Finally, it would be interesting to explore the concrete efficiency of other approaches to lightweight sublinear zero-knowledge arguments. In particular, one could consider constructions of PCPs based on bivariate polynomials such as the one of Polishchuk and Spielman [PS94] (see [BCGT13] for work in this direction), or the zero-knowledge PCP obtained by applying our general transformation to the MPC protocol from [DIK10]. This type of constructions can be further simplified by applying an interactive procedure for testing linear constraints as we do in Section 4.2.

Acknowledgments. We thank Eli Ben-Sasson, Swastik Kopparty, abhi shelat, and Salil Vadhan for useful discussions and pointers, the anonymous CCS reviewers for helpful comments, and Victor Shoup for his assistance with the NTL library.

The first and last authors were supported by Google Faculty Research Grant and NSF Awards CNS-1526377 and CNS-1618884. The second author was supported by the European Research Council under the ERC consolidators grant agreement n. 615172 (HIPS), and by the BIU Center for Research in Applied Cryptography and Cyber Security in conjunction with the Israel National Cyber Bureau in the Prime Minister's Office. The third author was supported by a DARPA/ARL SAFEWARE award, DARPA Brandeis program under Contract N66001-15-C-4065, NSF Frontier Award 1413955, NSF grants 1619348,

1228984, 1136174, and 1065276, ERC grant 742754, NSF-BSF grant 2015782, ISF grant 1709/14, BSF grant 2012378, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through the ARL under Contract W911NF-15-C-0205. The views expressed are those of the authors and do not reflect the official policy or position of Google, the Department of Defense, the National Science Foundation, or the U.S. Government.

References

[AHI+17] Benny Applebaum, Naama Haramaty, Yuval Ishai, Eyal Kushilevitz, and Vinod Vaikuntanathan. Low-complexity cryptographic hash functions. In *ITCS*, pages 7:1–7:31. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[AHIV17] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In *CCS*, pages 2087–2104, 2017.

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

[AM17] Benny Applebaum and Yoni Moses. Locally computable UOWHF with linear shrinkage. *J. Cryptol.*, 30(3):672–698, 2017.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[Bab85] László Babai. Trading group theory for randomness. In *STOC*, pages 421–429, 1985.

[BBB+18] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Gregory Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *S&P*, pages 315–334. IEEE Computer Society, 2018.

[BBC+17] Eli Ben-Sasson, Iddo Bentov, Alessandro Chiesa, Ariel Gabizon, Daniel Genkin, Matan Hamilis, Evgenya Pergament, Michael Riabzev, Mark Silberstein, Eran Tromer, and Madars Virza. Computational integrity with a public random string from quasi-linear pcps. In *EUROCRYPT*, pages 551–579, 2017.

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast reed-solomon interactive oracle proofs of proximity. In *ICALP*, pages 14:1–14:17, 2018.

[BBHR19] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable zero knowledge with no trusted setup. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO*, pages 701–732, 2019.

[BBHV22] Rishabh Bhadauria, Laasya Bangalore, Carmit Hazay, and Muthuramakrishnan Venkitasubramaniam. On black-box constructions of time and space efficient sublinear arguments from symmetric-key primitives. In *TCC*, 2022.

[BC12] Nir Bitansky and Alessandro Chiesa. Succinct arguments from multi-prover interactive proofs and their efficiency benefits. In *CRYPTO*, pages 255–272, 2012.

[BCCT13] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. Recursive composition and bootstrapping for SNARKS and proof-carrying data. In *STOC*, pages 111–120, 2013.

[BCG+14] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *IEEE Symposium on Security and Privacy*, pages 459–474, 2014.

[BCG+16] Eli Ben-Sasson, Alessandro Chiesa, Ariel Gabizon, Michael Riabzev, and Nicholas Spooner. Short interactive oracle proofs with constant query complexity, via composition and sumcheck. *IACR Cryptology ePrint Archive*, 2016:324, 2016.

[BCG+17] Jonathan Bootle, Andrea Cerulli, Essam Ghadafi, Jens Groth, Mohammad Hajiabadi, and Sune K. Jakobsen. Linear-time zero-knowledge proofs for arithmetic circuit satisfiability. In *ASIACRYPT*, pages 336–365, 2017.

[BCG20] Jonathan Bootle, Alessandro Chiesa, and Jens Groth. Linear-time arguments with sublinear verification from tensor codes. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC*, pages 19–46. Springer, 2020.

[BCGT13] Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, and Eran Tromer. On the concrete efficiency of probabilistically-checkable proofs. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 585–594, 2013.

[BCI+13] Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Omer Paneth. Succinct non-interactive arguments via linear interactive proofs. In *TCC*, pages 315–333, 2013.

[BCI+20] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity gaps for reed-solomon codes. In *FOCS*, 2020.

[BCL22] Jonathan Bootle, Alessandro Chiesa, and Siqi Liu. Zero-knowledge iops with linear-time prover and polylogarithmic-time verifier. In *EUROCRYPT*, pages 275–304. Springer, 2022.

[BCMS20] Benedikt Bünz, Alessandro Chiesa, Pratyush Mishra, and Nicholas Spooner. Recursive proof composition from accumulation schemes. In *TCC*, pages 1–18, 2020.

[BCR+19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. Aurora: Transparent succinct arguments for R1CS. In *EUROCRYPT*, pages 103–128, 2019.

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In *TCC*, pages 31–60, 2016.

[BFH+20] Rishabh Bhadauria, Zhiyong Fang, Carmit Hazay, Muthuramakrishnan Venkitasubramaniam, Tiancheng Xie, and Yupeng Zhang. Ligero++: A new optimized sublinear IOP. In *CCS*, pages 2025–2038, 2020.

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *STOC*, pages 21–31, 1991.

[BGH19] Sean Bowe, Jack Grigg, and Daira Hopwood. Halo: Recursive proof composition without a trusted setup. *IACR Cryptol. ePrint Arch.*, page 1021, 2019.

[BGKS20] Eli Ben-Sasson, Lior Goldberg, Swastik Kopparty, and Shubhangi Saraf. DEEP-FRI: sampling outside the box improves soundness. In *ITCS*, pages 5:1–5:32, 2020.

[BHR+20] Alexander R. Block, Justin Holmgren, Alon Rosen, Ron D. Rothblum, and Pratik Soni. Public-coin zero-knowledge arguments with (almost) minimal time and space overheads. In *TCC*, pages 168–197, 2020.

[BHR+21] Alexander R. Block, Justin Holmgren, Alon Rosen, Ron D. Rothblum, and Pratik Soni. Time- and space-efficient arguments from groups of unknown order. In *CRYPTO*, pages 123–152, 2021.

[BHST16] Eli Ben-Sasson, Matan Hamilis, Mark Silberstein, and Eran Tromer. Fast multiplication in binary fields on gpus via register cache. In *International Conference on Supercomputing*, pages 35:1–35:12, 2016.

[BIO14] Joshua Baron, Yuval Ishai, and Rafail Ostrovsky. On linear-size pseudorandom generators and hardcore functions. *Theor. Comput. Sci.*, 554:50–63, 2014.

[BIP+18] Dan Boneh, Yuval Ishai, Alain Passelègue, Amit Sahai, and David J. Wu. Exploring crypto dark matter: - new simple PRF candidates and their applications. In *TCC*, pages 699–729, 2018.

[BN20] Carsten Baum and Ariel Nof. Concretely-efficient zero-knowledge arguments for arithmetic circuits and their application to lattice-based cryptography. In *PLC*, pages 495–526, 2020.

[CC06] Hao Chen and Ronald Cramer. Algebraic geometric secret sharing schemes and secure multi-party computations over small fields. In *CRYPTO*, pages 521–536, 2006.

[CCH+19] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-shamir: from practice to theory. In *STOC*, pages 1082–1090, 2019.

[CDG+17] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In *CCS*, pages 1825–1842, 2017.

[CMT12] Graham Cormode, Michael Mitzenmacher, and Justin Thaler. Practical verified computation with streaming interactive proofs. In *ITCS*, pages 90–112, 2012.

[COS20] Alessandro Chiesa, Dev Ojha, and Nicholas Spooner. Fractal: Post-quantum and transparent recursive proofs from holography. In *EUROCRYPT*, pages 769–793, 2020.

[DI06] Ivan Damgård and Yuval Ishai. Scalable secure multiparty computation. In *CRYPTO*, pages 501–520, 2006.

[DIK10] Ivan Damgård, Yuval Ishai, and Mikkel Krøigaard. Perfectly secure multiparty computation and the computational overhead of cryptography. In *EUROCRYPT*, pages 445–465, 2010.

[dSGMOS19] Cyprien Delpech de Saint Guilhem, Lauren De Meyer, Emmanuela Orsini, and Nigel P. Smart. BBQ: using AES in picnic signatures. In *SAC*, pages 669–692, 2019.

[dSGOT21] Cyprien Delpech de Saint Guilhem, Emmanuela Orsini, and Titouan Tanguy. Limbo: Efficient zero-knowledge mpcith-based arguments. In *CCS*, 2021.

[EFKP20] Naomi Ephraim, Cody Freitag, Ilan Komargodski, and Rafael Pass. Sparks: Succinct parallelizable arguments of knowledge. In *EUROCRYPT*, pages 707–737, 2020.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194, 1986.

[GGPR13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct nizks without pcps. In *EUROCRYPT*, pages 626–645, 2013.

[GKR08] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: interactive proofs for muggles. In *STOC*, pages 113–122, 2008.

[GKR15] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: Interactive proofs for muggles. *J. ACM*, 62(4):27:1–27:64, 2015.

[GLS+21] Alexander Golovnev, Jonathan Lee, Srinath T. V. Setty, Justin Thaler, and Riad S. Wahby. Brakedown: Linear-time and post-quantum snarks for R1CS. *IACR Cryptol. ePrint Arch.*, page 1043, 2021.

[GM10] Shuhong Gao and Todd D. Mateer. Additive fast fourier transforms over finite fields. *IEEE Trans. Information Theory*, 56(12):6265–6272, 2010.

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

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

[Gro09] Jens Groth. Linear algebra with sub-linear zero-knowledge arguments. In *CRYPTO*, pages 192–208, 2009.

[Gro10] Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In *ASIACRYPT*, pages 321–340, 2010.

[GSV21] Yaron Gvili, Sarah Scheffler, and Mayank Varia. Booligero: Improved sublinear zero knowledge proofs for boolean circuits. In *FC*, pages 476–496, 2021.

[HKL22] David Heath, Vladimir Kolesnikov, and Jiahui Lu. Efficient generic arithmetic for KKW practical linear: Mpc-in-the-head NIZK on commodity hardware without trusted setup. *IACR Cryptol. ePrint Arch.*, page 795, 2022.

[HR18] Justin Holmgren and Ron Rothblum. Delegating computations with (almost) minimal time and space overhead. In Mikkel Thorup, editor, *FOCS*, pages 124–135, 2018.

[HR22] Justin Holmgren and Ron Rothblum. Faster sounder succinct arguments and iops. *IACR Cryptol. ePrint Arch.*, page 994, 2022.

[IKO07] Yuval Ishai, Eyal Kushilevitz, and Rafail Ostrovsky. Efficient arguments without short pcps. In *CCC*, pages 278–291, 2007.

[IKOS07] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In *STOC*, pages 21–30, 2007.

[IKOS08] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 433–442. ACM, 2008.

[IKOS09] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge proofs from secure multiparty computation. *SIAM J. Comput.*, 39(3):1121–1152, 2009.

[IMS12] Yuval Ishai, Mohammad Mahmoody, and Amit Sahai. On efficient zero-knowledge PCPs. In *TCC*, pages 151–168, 2012. Full version: https://www.cs.virginia.edu/~mohammad/files/papers/ZKPCPs-Full.pdf.

[IPS08] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer - efficiently. In *CRYPTO*, pages 572–591, 2008.

[IPS09] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Secure arithmetic computation with no honest majority. In *TCC*, pages 294–314, 2009.

[Ish20] Yuval Ishai. *Zero-knowledge proofs from information-theoretic proof systems*. 2020. https://zkproof.org/2020/08/12/information-theoretic-proof-systems.

[IW14] Yuval Ishai and Mor Weiss. Probabilistically checkable proofs of proximity with zero-knowledge. In *TCC*, pages 121–145, 2014.

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *STOC*, pages 723–732, 1992.

[KKW18] Jonathan Katz, Vladimir Kolesnikov, and Xiao Wang. Improved non-interactive zero knowledge with applications to post-quantum signatures. In *CCS*, pages 525–537, 2018.

[KMP20] Abhiram Kothapalli, Elisaweta Masserova, and Bryan Parno. A direct construction for asymptotically optimal zksnarks. *IACR Cryptol. ePrint Arch.*, page 1318, 2020.

[KR08] Yael Tauman Kalai and Ran Raz. Interactive PCP. In *ICALP*, pages 536–547, 2008.

[LFKN90] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. In *FOCS*, pages 2–10, 1990.

[LSTW21] Jonathan Lee, Srinath T. V. Setty, Justin Thaler, and Riad S. Wahby. Linear-time zero-knowledge snarks for R1CS. *IACR Cryptol. ePrint Arch.*, page 30, 2021.

[Mer89] Ralph C. Merkle. A certified digital signature. In *CRYPTO*, pages 218–238, 1989.

[Mic94] Silvio Micali. CS proofs (extended abstracts). In *FOCS*, pages 436–453, 1994.

[PS94] Alexander Polishchuk and Daniel A. Spielman. Nearly-linear size holographic proofs. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 194–203, 1994.

[PV20] Rafael Pass and Muthuramakrishnan Venkitasubramaniam. Is it easier to prove theorems that are guaranteed to be true? In *FOCS*, pages 1255–1267, 2020.

[RR22] Noga Ron-Zewi and Ron D. Rothblum. Proving as fast as computing: succinct arguments with constant prover overhead. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1353–1363. ACM, 2022.

[RRR16] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. In *STOC*, pages 49–62, 2016.

[RZ17] Ronny Roth and Gilles Zémor. Personal communication, 2017.

[SBV+13] Srinath T. V. Setty, Benjamin Braun, Victor Vu, Andrew J. Blumberg, Bryan Parno, and Michael Walfish. Resolving the conflict between generality and plausibility in verified computation. In *Eighth Eurosys Conference*, pages 71–84, 2013.

[Set20] Srinath T. V. Setty. Spartan: Efficient and general-purpose zksnarks without trusted setup. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO*, pages 704–737, 2020.

[Sha90] Adi Shamir. Ip=pspace. In *FOCS*, pages 11–15, 1990.

[SL20] Srinath T. V. Setty and Jonathan Lee. Quarks: Quadruple-efficient transparent zksnarks. *IACR Cryptol. ePrint Arch.*, page 1275, 2020.

[SMBW12] Srinath T. V. Setty, Richard McPherson, Andrew J. Blumberg, and Michael Walfish. Making argument systems for outsourced computation practical (sometimes). In *NDSS*, 2012.

[Tha13] Justin Thaler. Time-optimal interactive proofs for circuit evaluation. In *CRYPTO*, pages 71–89, 2013.

[Tha22] Justin Thaler. *Proofs, arguments, and zero-knowledge*. 2022. https://people.cs.georgetown.edu/jthaler/ProofsArgsAndZK.html.

[VSBW13] Victor Vu, Srinath T. V. Setty, Andrew J. Blumberg, and Michael Walfish. A hybrid architecture for interactive verifiable computation. In *S&P*, pages 223–237, 2013.

[WB15] Michael Walfish and Andrew J. Blumberg. Verifying computations without reexecuting them. *Commun. ACM*, 58(2):74–84, 2015.

[WTS+18] Riad S. Wahby, Ioanna Tzialla, Abhi Shelat, Justin Thaler, and Michael Walfish. Doubly-efficient zksnarks without trusted setup. In *S&P*, pages 926–943, 2018.

[XZZ+19a] Tiancheng Xie, Jiaheng Zhang, Yupeng Zhang, Charalampos Papamanthou, and Dawn Song. Libra: Succinct zero-knowledge proofs with optimal prover computation. In *CRYPTO*, pages 733–764, 2019.

[XZZ+19b] Tiancheng Xie, Jiaheng Zhang, Yupeng Zhang, Charalampos Papamanthou, and Dawn Song. Libra: Succinct zero-knowledge proofs with optimal prover computation. In *CRYPTO*, pages 733–764. Springer, 2019.

[ZGK+17] Yupeng Zhang, Daniel Genkin, Jonathan Katz, Dimitrios Papadopoulos, and Charalampos Papamanthou. vsql: Verifying arbitrary SQL queries over dynamic outsourced databases. In *IEEE Symposium on Security and Privacy*, pages 863–880, 2017.

[ZLW+21a] Jiaheng Zhang, Tianyi Liu, Weijie Wang, Yinuo Zhang, Dawn Song, Xiang Xie, and Yupeng Zhang. Doubly efficient interactive proofs for general arithmetic circuits with linear prover time. In *CCS*, pages 159–177, 2021.

[ZLW+21b] Jiaheng Zhang, Tianyi Liu, Weijie Wang, Yinuo Zhang, Dawn Song, Xiang Xie, and Yupeng Zhang. Doubly efficient interactive proofs for general arithmetic circuits with linear prover time. In *CCS*, pages 159–177. ACM, 2021.

[ZXZS20] Jiaheng Zhang, Tiancheng Xie, Yupeng Zhang, and Dawn Song. Transparent polynomial delegation and its applications to zero knowledge proof. In *S&P*, pages 859–876. IEEE, 2020.

A Case $e < d/3$: Proof of Lemma 4.4

In this section, we provide the proof of our main lemma for the case when $e < d/3$. The proof of claim A.2 below is due to Ronny Roth and Gilles Zémor [RZ17].[5]

Lemma A.1 (restatement of Lemma 4.4)**.** *Let e be a positive integer such that $e < d/3$. Suppose $d(U, L^m) > e$. Then, for a random w^* in the row-span of U, we have*

$$\Pr[d(w^*, L) \leq e] \leq (e+1)/|\mathbb{F}|.$$

Proof. Suppose that $d(U^*, L^m) > e$ and L^* is the span of the vectors in U^*. Assume towards a contradiction that $d(v^*, L) \leq e$ for all $v^* \in L^*$. Suppose $v_0^* \in L^*$ maximizes the distance from L. Since $d(U^*, L^m) > e$, there must be a row U_i^* such that $\Delta(U_i^*, L) \setminus \Delta(v_0^*, L) \neq \emptyset$. Let $v_0^* = u_0 + \chi_0$ and $U_i^* = u_i + \chi_i$ for $u_0, u_i \in L$ and χ_0, χ_i of weight $\leq e$. We argue that there exists $\alpha \in \mathbb{F}$ such that for $\hat{v} = v_0^* + \alpha U_i^*$ we have $d(\hat{v}, L) > d(v_0^*, L)$, contradicting the choice of v_0^*. This follows by a union bound, noting that for any $j \in \Delta(v_0^*, L) \cup \Delta(U_i^*, L)$ there is at most one choice of α such that $\hat{v}_j = 0$.

Now, it suffices to show that in any affine subspace of $\mathbb{F}^n$, either all points are e-close to L or almost all are not. This reduces to showing the following claim. We state an explicit version of the conjecture for the case of RS codes.

Claim A.2. *Let L be an arbitrary linear code over $\mathbb{F}$ of length n. Let e be a positive integer such that $e < d/3$. Then for every $u, v \in \mathbb{F}^n$, defining an affine line $\ell_{u,v} = \{u + \alpha v : \alpha \in \mathbb{F}\}$, either (1) for every $x \in \ell_{u,v}$ we have $d(x, L) \leq e$, or (2) for at most d points $x \in \ell_{u,v}$ we have $d(x, L) \leq e$.*

We begin with the observation that for any two length n vectors u and v of weight at most e, $\ell_{u,v}$ contains N points at most distance e from L if and only if $\ell_{u,v+c}$ contains N points of distance at most e from L for any codeword $c \in L$. This means it suffices to prove the claim for vectors u and v of weight at most e.

We now prove the lemma in two cases

Case 1: $|Support(u) \cup Support(v)| \leq e$ This means that $\ell_{u,v}$ is entirely contained in the ball $B_e(\mathbf{0})$ where $\mathbf{0}$ is the all 0s vector which in turn means all the vectors in the line are at most t from L.

Case 2: $|Support(u) \cup Support(v)| \geq e + 1$ Since u and v each have weight at most e, the intersection of their supports can be of cardinality at most $e - 1$. For each of the coordinates in the intersection of the supports, there can be at most one vector in $\ell_{u,v}$ such that the entry in that coordinate is 0. Therefore, there are at most $e - 1$ vectors in $\ell_{u,v}$ that are contained in the ball $B_e(\mathbf{0})$ where $\mathbf{0}$ is the all 0s vector.

[5] In the case of length-n Reed-Solomon codes, a similar bound for $e < d/2$ was obtained by Ben-Sasson et al. [BCI+20], where $(e+1)/|\mathbb{F}|$ is relaxed to $n/|\mathbb{F}|$.

To conclude this case, we need to demonstrate that there exists no codeword $c \neq \mathbf{0}$ such that the line $\ell_{u,v}$ intersects with a vector inside the ball of radius e around c. Assume for contradiction there exists a codeword c and vector w of weight at most e such that $c + w \in \ell_{u,v}$. Then we have that

$$c + w = u + \alpha v$$

This means that c is equal to the sum of three vectors each of weight at most e. Now we arrive at a contradiction because the minimum distance of L is d and $e < d/3$.

■

B Generalizing IPCP Tests

In this section, we provide the generalized versions of the tests in our basic IPCP. This is required for improving the soundness analysis and achieving better concrete parameters. We remark that the theorem statements in this section are provided for the case $e < d/4$. But we can incorporate the subsequent improvement in the analyses and directly generalize for the cases $e < d/3$ and $e < d/2$.

B.1 Generalized Interleaved Linear Code Testing

In this section we present a generalized version of the testing algorithm that uses σ linear combinations to amplify soundness; see Figure 7. This algorithm is useful for obtaining better soundness over a small field $\mathbb{F}$.

Oracle: A purported L^m-codeword U. Depending on the context, we may view U either as a matrix in $\mathbb{F}^{m \times n}$ in which each row is a purported L-codeword, or as a sequence of n symbols $(U_1, \ldots, U_n)$, $U_i \in \mathbb{F}^m$.

Parameters:

- Probing parameter $t < n$ (number of symbols U_j read by $\mathcal{V}$).
- Repetition parameter σ (number of random linear combinations).

Interactive testing:

1. $\mathcal{V}$ picks σ random linear combinations $r_1, \ldots, r_\sigma \in \mathbb{F}^m$ and sends them to $\mathcal{P}$.
2. $\mathcal{P}$ responds with $w_h = r_h^T U \in \mathbb{F}^n$, $h = 1, \ldots, \sigma$.
3. $\mathcal{V}$ queries a set $Q \subset [n]$ of t random symbols U_j, $j \in Q$.
4. $\mathcal{V}$ accepts iff all w_h are in L and are consistent with U_Q and r_h. That is, for every $j \in Q$ and $1 \leq h \leq \sigma$, we have
$$\sum_{i=1}^{m} (r_h)_j \cdot U_{i,j} = (w_h)_j.$$

Figure 7: **Generalized-Test-Interleaved**$(\mathbb{F}, L[n, k, d], m, t, \sigma; U)$

Lemma B.1. *If $U \in L^m$ and $\mathcal{P}$ is honest, then $\mathcal{V}$ always accepts.*

Lemma B.2. *Let e be a positive integer such that $e < d/4$. Suppose $d(U^*, L^m) > e$. Then, for a random w^* in the row-span of U^*, we have*

$$\Pr[d(w^*, L) \leq e] \leq (e+1)/|\mathbb{F}|^\sigma.$$

The proof of Lemma B.2 follows identically as the proof of Lemma 4.2 with the exception that the denominator $|\mathbb{F}|$ in Equations 1 and 2 need to be replaced by $|\mathbb{F}|^\sigma$. This is because in each of Cases 1 and 2, we express $w^* = \alpha v^* + x$ and bound the probability of a bad event regarding w^* claiming that any value of x happens for a unique value of $\alpha \in \mathbb{F}$. Therefore this probability is bound by $1/|\mathbb{F}|$. In the repeated version, there is one possible value in $\mathbb{F}^\sigma$ which happens with probability $1/|\mathbb{F}|^\sigma$.

We can conclude the following theorem, the same way Theorem 4.4 is concluded from Lemma 4.2.

Theorem B.1. *Let e be a positive integer such that $e < d/4$. Suppose $d(U^*, L^m) \geq e$. Then, for any malicious $\mathcal{P}$ strategy, the oracle U^* is rejected by $\mathcal{V}$ except with $\leq (1 - e/n)^t + (e+1)/|\mathbb{F}|^\sigma$ probability.*

B.2 Affine Interleaved Linear Code Testing

For the purpose of obtaining a zero-knowledge IPCP, the following "affine" variant of **Test-Interleaved** is useful. Whenever $\mathcal{V}$ requests a random linear combination of the rows of U, this linear combination will be masked with an additional blinding vector $u' \in \mathbb{F}^n$. The vector u', which is also given as part of the proof oracle, will be picked by an honest $\mathcal{P}$ at random from L and will therefore hide all information about U whose rows are from L. The soundness of the test should hold even when u' is adversarially chosen and is not necessarily a codeword. The complete test is given in Figure 8.

Oracle: A purported L^m-codeword U and an additional auxiliary row vector $u' \in \mathbb{F}^n$.

Interactive testing:

1. $\mathcal{V}$ picks a random linear combinations $r \in \mathbb{F}^m$ and sends r to $\mathcal{P}$.
2. $\mathcal{P}$ responds with $w = r^T U + u' \in \mathbb{F}^n$.
3. $\mathcal{V}$ queries a set $Q \subset [n]$ of t random symbols U_j, $j \in Q$, as well as u'_j, $j \in Q$.
4. $\mathcal{V}$ accepts iff $w \in L$ and w is consistent with U_Q, u'_Q, and r. That is, for every $j \in Q$ we have $\sum_{i=1}^m r_j \cdot U_{i,j} + u'_j = w_j$.

Figure 8: Affine-Test-Interleaved$(\mathbb{F}, L[n,k,d], m, t; U, u')$

Completeness follows directly from the description.

Lemma B.3. *If $U \in L^m$, $u' \in L$, and $\mathcal{P}$ is honest, then $\mathcal{V}$ always accepts.*

Our soundness analysis will rely on the following lemma.

Lemma B.4. *Let e be a positive integer such that $e < d/4$. Suppose $d(U^*, L^m) > e$. Then, for arbitrary $u' \in \mathbb{F}^n$ and a random w^* in the row-span of U^*, we have $\Pr[d(w^*, L) \leq e] \leq (e+1)/|\mathbb{F}|$.*

Theorem B.2. *Let e be a positive integer such that $e < d/4$. Suppose $d(U^*, L^m) \geq e$. Then, for an arbitrary $u' \in \mathbb{F}^n$ and any malicious $\mathcal{P}$ strategy, the oracle U^* is rejected by $\mathcal{V}$ except with $\leq (1-e/n)^t + (e+1)/|\mathbb{F}|$ probability.*

We provide a formal proof of a generalization of this test in the next section.

B.3 Generalized Affine Interleaved Linear Code Testing

For the purpose of obtaining a zero-knowledge IPCP, the following "affine" variant of **Test-Interleaved** is useful. Whenever $\mathcal{V}$ requests a random linear combination of the rows of U, this linear combination will be masked with an additional blinding vector $u' \in \mathbb{F}^n$. The vector u', which is also given as part of the proof oracle, will be picked by an honest $\mathcal{P}$ at random from L and will therefore hide all information about U whose rows are from L. The soundness of the test should hold even when u' is adversarially chosen and is not necessarily a codeword. We generalize it further following the previous section to achieve better soundness by repetition; see Figure 9.

Oracle: A purported L^m-codeword U and additional auxiliary row vectors $u'_1, \ldots, u'_\sigma \in \mathbb{F}^n$.

Interactive testing:

1. $\mathcal{V}$ picks a random linear combinations $r_1, \ldots, r_\sigma \in \mathbb{F}^m$ and sends r to $\mathcal{P}$.
2. $\mathcal{P}$ responds with $w_h = r_h^T U + u'_h \in \mathbb{F}^n$, $h = 1, \ldots, \sigma$.
3. $\mathcal{V}$ queries a set $Q \subset [n]$ of t random symbols U_j, $j \in Q$, as well as $(u_\sigma)_j$, $j \in Q$.
4. $\mathcal{V}$ accepts iff all $w_h \in L$ and are consistent with w_h and U_Q, u_h, and r_h. That is, for every $j \in Q$ we have $\sum_{i=1}^m (r_h)j \cdot U_{i,j} + (u'_h)_j = w_j$.

Figure 9: Generalized-Affine-Test-Interleaved$(\mathbb{F}, L[n,k,d], m, t, \sigma; U, u')$

Completeness follows directly from the description. Soundness analysis follows as described in Section B.1.

Lemma B.5. *If $U \in L^m$, $u'_1, \ldots, u'_\sigma \in L$, and $\mathcal{P}$ is honest, then $\mathcal{V}$ always accepts.*

Lemma B.6. *Let e be a positive integer such that $e < d/4$. Suppose $d(U^*, L^m) > e$. Then, for arbitrary $u'_1, \ldots, u'_\sigma \in \mathbb{F}^n$ and a random w^* in the row-span of U^*, we have $\Pr[\forall\ h \in [\sigma], d(w^* + u'_h, L) \leq e] \leq (e+1)/|\mathbb{F}|^\sigma$.*

Theorem B.3. *Let e be a positive integer such that $e < d/4$. Suppose $d(U^*, L^m) \geq e$. Then, for arbitrary $u'_1, \ldots, u'_\sigma \in \mathbb{F}$ and any malicious $\mathcal{P}$ strategy, the oracle U^* is rejected by $\mathcal{V}$ except with $\leq (1 - e/n)^t + (e+1)/|\mathbb{F}|^\sigma$ probability.*

B.4 Generalized Affine Linear Constraint Testing over Interleaved Reed Solomon Codes

For the purpose of obtaining a zero-knowledge IPCP, we provide the following "affine" variant of **Test-Linear-Constraints-IRS**. Whenever $\mathcal{V}$ provides the challenge vector r, the linear combination $r^T A$ of the rows of U, will be masked with an additional blinding vector $u' \in \mathbb{F}^n$ that encodes messages that sum up to 0. The vector u', which is also given as part of the proof oracle, will be picked by an honest $\mathcal{P}$ at random from L subject to the condition that it encodes messages that sum up to 0 and will therefore hide all information about the individual column sums in the computation of $r^T Ax$. The soundness of the test should hold even when u' is adversarially chosen and is not necessarily a codeword. We will further generalize the test to achieve better soundness. Namely, instead of relying on repetition, we improve soundness by considering the challenge space from an extension field. The test is given in Figure 10. Note that just as in Section 4.2, we

Oracle: A purported L^m-codeword U that should encode a message $x \in \mathbb{F}^{m\ell}$ satisfying $Ax = b$ and an additional auxiliary row vector $u' \in \hat{\mathbb{F}}^n$ that encodes the message $(\gamma_1, \ldots, \gamma_\ell)$ such that $\sum_{c\in[\ell]} \gamma_c = 0$ where $\hat{\mathbb{F}}$ is an extension field of $\mathbb{F}$ such that $|\hat{\mathbb{F}}| = |\mathbb{F}|^\sigma$.

Interactive testing:

1. $\mathcal{V}$ picks a random vector $r \in \hat{\mathbb{F}}^{m\ell}$ and sends r to $\mathcal{P}$.
2. $\mathcal{V}$ and $\mathcal{P}$ compute
$$r^T A = (r_{11}, \ldots, r_{1\ell}, \ldots, r_{m1}, \ldots, r_{m\ell})$$
and for $i \in [m]$, let $r_i(\cdot)$ be the unique polynomial of degree $< \ell$ such that $r_i(\zeta_c) = r_{ic}$ for every $c \in [\ell]$.
3. $\mathcal{P}$ sends the $k + \ell - 1$ coefficients of the polynomial defined by $q(\cdot) = \sum_{i=1}^m r_i(\cdot) \cdot p_i(\cdot) + r_{\text{Blind}}(\cdot)$, where $p_i(\cdot)$ is the polynomial of degree $< k$ corresponding to row i of U and $r_{\text{Blind}}(\cdot)$ is the polynomial of degree $< k$ corresponding to u'.
4. $\mathcal{V}$ queries a set $Q \subset [n]$ of t random symbols U_j, $j \in Q$, as well as u'_j, $j \in Q$.
5. $\mathcal{V}$ accepts if the following conditions hold:
   (a) $\sum_{c\in[\ell]} q(\zeta_c) = \sum_{i\in[m],c\in[\ell]} r_{ic} b_{ic}$.
   (b) For every $j \in Q$ we have $u'_j + \sum_{i=1}^m r_i(\eta_j) \cdot U_{i,j} = q(\eta_j)$.

Figure 10: Generalized-Affine-Test-Linear-Constraints-IRS$(\mathbb{F}, L = \mathsf{RS}_{\mathbb{F},n,k,\eta}, m, t, \zeta, A, b, \sigma; U)$

will analyze the test under the promise that the (possibly badly formed) U is close to L^{m+1}. Completeness follows directly as u' does not affect the verification. We argue soundness next.

Lemma B.7. *Let e be a positive integer such that $e < d/2$. Suppose that a (badly formed) oracle U^* that is vertically juxtaposed with an arbitrary u' is e-close to a codeword $V \in L^{m+1}$, where V contains the codewords $U \in L^m$ and $u^* \in L$ vertically juxtaposed, and U encodes $x \in \mathbb{F}^{m\ell}$ such that $Ax \neq b$. Then, for any malicious $\mathcal{P}$ strategy, U^* is rejected by $\mathcal{V}$ except with at most $1/|\mathbb{F}|^\sigma + ((e + k + \ell)/n)^t$ probability.*

B.5 Generalized Testing Quadratic Constraints over Interleaved Reed Solomon Codes

Finally, in this section we extend our quadratic constraint test over Interleaved Reed Solomon codes via parallel repetition to improve soundness. The complete test description is provided in Figure 11. Next, we state the completeness and soundness statements.

Lemma B.8. *If $U^x, U^y, U^z \in L^m$ encode vectors $x, y, z \in \mathbb{F}^{m\ell}$ satisfying $x \odot y + a \odot z = b$ and $\mathcal{P}$ is honest, $\mathcal{V}$ always accepts.*

Lemma B.9. *Let e be a positive integer such that $e < d/2$. Let U^{x*}, U^{y*}, U^{z*} be badly formed oracles and let $U^* \in \mathbb{F}^{3m\times n}$ be the matrix obtained by vertically juxtaposing the corresponding $m \times n$ matrices. Suppose $d(U^*, L^{3m}) \leq e$, and let U^x, U^y, U^z, respectively, be the (unique) codewords in L^m that are closest to U^{x*}, U^{y*}, U^{z*}. Suppose U^x, U^y, U^z encode x, y, z such that $x \odot y + a \odot z \neq b$. Then, for any malicious $\mathcal{P}$ strategy, (U^{x*}, U^{y*}, U^{z*}) is rejected by $\mathcal{V}$ except with at most $1/|\mathbb{F}|^\sigma + ((e + 2k)/n)^t$ probability.*

Oracle: Purported L^m-codewords U^x, U^y, U^z that should encode messages $x, y, z \in \mathbb{F}^{m\ell}$ satisfying $x \odot y + a \odot z = b$ and an additional auxiliary row vector $u' \in \hat{\mathbb{F}}^n$ that encodes the all 0s message in $\mathsf{RS}_{\mathbb{F},n,2k,\eta}$ where $\hat{\mathbb{F}}$ is an extension field of $\mathbb{F}$ such that $|\hat{\mathbb{F}}| = |\mathbb{F}|^\sigma$.

Interactive testing:

1. Let $U^a = \mathsf{Encode}_\zeta(a)$ and $U^b = \mathsf{Encode}_\zeta(b)$.
2. $\mathcal{V}$ picks a random linear combinations $r \in \hat{\mathbb{F}}^m$ and sends r to $\mathcal{P}$.
3. $\mathcal{P}$ sends the $2k-1$ coefficients of the σ polynomials $p_0^1, \ldots, p_0^\sigma$ defined by $p_0(\cdot) = \sum_{i=1}^m r_i \cdot p_i(\cdot)$, where $p_i(\cdot) = p_i^x(\cdot) \cdot p_i^y(\cdot) + p_i^a(\cdot) \cdot p_i^z(\cdot) - p_i^b(\cdot)$, and where p_i^x, p_i^y, p_i^z are the polynomials of degree $< k$ corresponding to row i of U^x, U^y, U^z, p_i^a, p_i^b are the polynomials of degree $< \ell$ corresponding to row i of U^a, U^b and $r_{\text{Blind}}(\cdot)$ is the polynomial of degree $< 2k$ corresponding to u'.
4. $\mathcal{V}$ picks a random index set $Q \subset [n]$ of size t, and queries $U_j^x, U_j^y, U_j^z, u'_j$, $j \in Q$.
5. $\mathcal{V}$ accepts if the following conditions hold:
   (a) $p_0^h(\zeta_c) = 0$ for every $c \in [\ell]$.
   (b) For every $j \in Q$, it holds that $p_0(\eta_j) = u'_j + \sum_{i=1}^m r_i \cdot \left[U_{i,j}^x \cdot U_{i,j}^y + U_{i,j}^a \cdot U_{i,j}^z - U_{i,j}^b\right]$.

Figure 11: Generalized-Test-Quadratic-Constraints-IRS $(\mathbb{F}, L = \mathsf{RS}_{\mathbb{F},n,k,\eta}, m, t, \zeta, a, b, \sigma; U^x, U^y, U^z)$

C Improving the Soundness Analysis

Recall that the soundness error is calculated by applying a union bound over the following tests: (1) Interleaved Reed-Solomon Test, (2) Linear Constraints Test, and (3) Quadratic Test. We show next how we can improve the soundness of the Linear and Quadratic tests assuming that the Interleaved Reed-Solomon Test passes.

Interleaved Reed Solomon (IRS) test. The soundness of this test was bounded by $(e+1)/|\mathbb{F}| + (1-e/n)^t$ when $e < d/4$ and bounded by $d/|\mathbb{F}| + (1-e/n)^t$ when $e < d/3$. More recently, a better analysis has been presented in [BCI+20] where they improve Lemma 4.4 from $e < d/3$ to $e < d/2$ bounding the error by $n/|\mathbb{F}|$ where n is the code length (See Theorem 1.2: Unique decoding bound). Thus the soundness of this test can be bounded by $n/|\mathbb{F}| + (1-e/n)^t$ for $e < d/2$.

We make a slight modification to the analysis here where we bound the following "bad" events.

- Let E_1 be the event that more than e columns of U have errors. From the preceding analysis we have that the probability the verifier accepts the IRS test in this case is at most $n/|\mathbb{F}| + (1-e/n)^t$ for $e < d/2$.
- Suppose that event E_1 does not occur. Denote by the prover's response to the IRS test by w^*. Since $e < (n-k)/2$ and there are fewer than e errors, let U be the unique codeword such that $d(U, U^*) < e$. Define w to be the codeword that is the result of correctly computing the IRS test with the matrix U. In particular, w will agree columnwise with all columns of U^* (except the ones that have errors). Define the event E_2 to be when $w \neq w^*$. We bound the probability that the test passes if E_2 occurs and E_1 does not. If the test passes we have that w^* is a valid codeword. Therefore, w and w^* can agree in at most k columns. With at most e columns with

errors, the verifier can possible accept the test only if all the indices it chooses come from the k columns they agree on and additionally the e columns containing errors. This probability is at most

$$\frac{\binom{k+e}{t}}{\binom{n}{t}} \leq \left(\frac{k+e}{n}\right)^t \leq \left(1-\frac{e}{n}\right)^t$$

where the last equality comes from setting $k \leq n-2e$.

- Let E_3 be the event that the verifier picks any of the columns that contain errors. We argue that then the IRS test would fail with probability $1/|\mathbb{F}|$. Let U and U^* disagree on the i^{th} column, j^{th} row. Then, in the IRS test, if E_1 and E_2 do not occur then given the random combination for all the rows except the j^{th} row, there will be exactly one possible value in the linear combination corresponding the j^{th} row that will make the test pass if i was selected by the verifier. This occurs with probability $1/|\mathbb{F}|$.

We now have that:

$$\begin{aligned}
&\Pr[V^* \text{ accepts IRS test } \wedge (E_1 \vee E_2 \vee E_3)] \\
&\quad \leq \Pr[V^* \text{ accepts IRS test } \wedge E_1] + \Pr[V^* \text{ accepts IRS test } \wedge (E_2 \vee E_3) \wedge \neg E_1] \\
&\quad = \Pr[V^* \text{ accepts IRS test } | E_1] \cdot \Pr[E_1] + \Pr[V^* \text{ accepts IRS test } \wedge (E_2 \vee E_3) | \neg E_1] \\
&\quad \leq \left[\frac{n}{|\mathbb{F}|} + \left(1-\frac{e}{n}\right)^t\right] \cdot \Pr[E_1] \\
&\qquad + \Pr[V^* \text{ accepts IRS test } \wedge (E_2 \vee E_3) \wedge \neg E_1] \\
&\quad \leq \frac{n}{|\mathbb{F}|} + \left(1-\frac{e}{n}\right)^t \cdot \Pr[E_1] \\
&\qquad + \Pr[V^* \text{ accepts IRS test } \wedge E_2 | \neg E_1] \cdot \Pr[\neg E_1] \\
&\qquad\quad + \Pr[V^* \text{ accepts IRS test } \wedge E_3 \wedge \neg E_1 \wedge \neg E_2] \\
&\quad \leq \frac{n}{|\mathbb{F}|} + \left(1-\frac{e}{n}\right)^t \cdot \Pr[E_1] \\
&\qquad + \left(1-\frac{e}{n}\right)^t \cdot \Pr[\neg E_1] + \Pr[V^* \text{ accepts IRS test } \wedge E_3 | \neg E_1 \wedge \neg E_2] \\
&\quad \leq \frac{n}{|\mathbb{F}|} + \left(1-\frac{e}{n}\right)^t \cdot \Pr[E_1] \\
&\qquad + \left(1-\frac{e}{n}\right)^t \cdot \Pr[\neg E_1] + \frac{1}{|\mathbb{F}|} \\
&\quad = \frac{n+1}{|\mathbb{F}|} + \left(1-\frac{e}{n}\right)^t
\end{aligned}$$

Linear Constraints Test: The analysis in [AHIV17] bounds this test by $((e+k+\ell)/n)^t + 1/|\mathbb{F}|$. By analyzing this test in conjunction with the IRS test we can replace the term $((e+k+\ell)/n)^t$ with $((k+\ell)/n)^t$. The main idea here is that term $((e+k+\ell)/n)^t$ computes the probability that the verifier chooses all its t indices from within the e columns that have errors and an additional of at most $k+\ell$ columns.

Now we analyze the linear test assuming E_1, E_2 and E_3 do not occur as we have bounded them in the IRS test. Specifically, since E_3 does not occur, it suffices to bound the case when all the indices are

chosen within the additional at most $k + \ell$ columns excluding the columns with errors. This can be bounded by $\left(\frac{k+\ell}{n}\right)^t$.

Therefore, the soundness of this test can be bounded by

$$\left(\frac{k+\ell}{n}\right)^t + 1/|\mathbb{F}|$$

assuming that none of the columns containing errors are chosen.

Quadratic Test: The analysis in [AHIV17] bounds this test by $((e+2k)/n)^t + 1/|\mathbb{F}|$. Following the same arguments as in the Linear Test, the soundness of this test can be improved to simply

$$\left(\frac{2k}{n}\right)^t + 1/|\mathbb{F}|.$$

Therefore, the overall the soundless error can be bounded by

$$\left[(1-e/n)^t + \left(\frac{k+\ell}{n}\right)^t + \left(\frac{2k}{n}\right)^t + \left(\frac{n+3}{(2^{30})^\sigma}\right)\right]$$