Quasi-Linear Size Zero Knowledge from Linear-Algebraic PCPs

Eli Ben-Sasson[2(✉)], Alessandro Chiesa[3], Ariel Gabizon[2], and Madars Virza[1]

[1] MIT, Cambridge, USA
[2] Technion, Haifa, Israel
eli@cs.technion.ac.il
[3] UC Berkeley, Berkeley, USA

Abstract. The seminal result that every language having an interactive proof also has a zero-knowledge interactive proof assumes the existence of one-way functions. Ostrovsky and Wigderson [33] proved that this assumption is necessary: if one-way functions do not exist, then only languages in BPP have zero-knowledge interactive proofs.

Ben-Or et al. [9] proved that, nevertheless, every language having a multi-prover interactive proof also has a zero-knowledge multi-prover interactive proof, unconditionally. Their work led to, among many other things, a line of work studying zero knowledge without intractability assumptions. In this line of work, Kilian, Petrank, and Tardos [28] defined and constructed zero-knowledge probabilistically checkable proofs (PCPs).

While PCPs with quasilinear-size proof length, but without zero knowledge, are known, no such result is known for zero knowledge PCPs. In this work, we show how to construct "2-round" PCPs that are zero knowledge and of length $\tilde{O}(K)$ where K is the number of queries made by a malicious polynomial time verifier. Previous solutions required PCPs of length at least K^6 to maintain zero knowledge. In this model, which we call *duplex PCP* (DPCP), the verifier first receives an oracle string from the prover, then replies with a message, and then receives another oracle string from the prover; a malicious verifier can make up to K queries in total to both oracles.

Deviating from previous works, our constructions do not invoke the PCP Theorem as a blackbox but instead rely on certain algebraic properties of a specific family of PCPs. We show that if the PCP has a certain linear algebraic structure — which many central constructions can be shown to possess, including [2,4,15] — we can add the zero knowledge property at virtually no cost (up to additive lower order terms) while introducing only minor modifications in the algorithms of the prover and verifier. We believe that our linear-algebraic characterization of PCPs may be of independent interest, as it gives a simplified way to view previous well-studied PCP constructions.

E. Kushilevitz and T. Malkin (Eds.): TCC 2016-A, Part II, LNCS 9563, pp. 33–64, 2016.
DOI: 10.1007/978-3-662-49099-0_2

1 Introduction

We continue the study of proof systems that provide soundness and zero knowledge, simultaneously and unconditionally (i.e., no intractability assumptions are needed to achieve the two), as we now explain.

Interactive Proofs. An *interactive proof* [6,20] for a language $\mathscr{L}$ is a pair of interactive algorithms (P, V), where P is known as the prover and V as the verifier, that satisfies the following: (i) (completeness) for every instance $\mathbb{x}$ in $\mathscr{L}$, $P(\mathbb{x})$ can make $V(\mathbb{x})$ accept with probability 1; (ii) (soundness) for every instance $\mathbb{x}$ not in $\mathscr{L}$, every prover $\tilde{P}$ can make $V(\mathbb{x})$ accept with at most a small probability ϵ. Shamir [35] showed the expressive power of interactive proofs by proving that **IP** = **PSPACE**, i.e., all and only languages in **PSPACE** have interactive proofs.

Zero Knowledge. An interactive proof is *zero knowledge* [20] if the verifier, even if malicious, cannot learn any information about an instance $\mathbb{x}$ in $\mathscr{L}$, by interacting with the prover, besides the fact $\mathbb{x}$ is in $\mathscr{L}$: for any efficient verifier $\tilde{V}$ there exists an efficient simulator S such that $S(\mathbb{x})$ is "indistinguishable" from the view of $\tilde{V}$ while interacting with $P(\mathbb{x})$. Depending on the choice of definition for indistinguishability, one gets different flavors of zero knowledge.

If indistinguishability is required to hold for efficient deciders only, then one gets *computational* zero knowledge; **CZK** denotes the corresponding complexity class. A seminal result in cryptography says that if one-way functions exist then **CZK** = **IP**, i.e., every language having an interactive proof also has a computational zero-knowledge interactive proof [8,20,23]. If indistinguishability is required to hold for all deciders, then one gets *statistical* zero knowledge; if instead the simulator's output and the verifier's view are the same distribution (and not merely close to each other), then one gets *perfect* zero knowledge. These stronger notions determine the corresponding complexity classes **SZK** and **PZK**, both of which are contained in $\mathbf{AM} \cap \mathbf{coAM}$; of course, $\mathbf{PZK} \subseteq \mathbf{SZK} \subseteq \mathbf{CZK}$.

Unfortunately, zero knowledge cannot be achieved unconditionally for non-trivial languages: Ostrovsky and Wigderson [33] proved that if one-way functions do not exist then **CZK** equals an average-case variant of **BPP**.

Other Types of Proof Systems. Due to the limitations of interactive proofs with respect to zero knowledge that holds unconditionally, researchers have explored other types of proof systems, as an alternative to interactive proofs.

- **MIP.** Ben-Or et al. [9] first studied statistical zero knowledge, and proved that it can be achieved in a new model, *multi-prover interactive proof* (MIPs), where the verifier interacts with multiple provers that are not allowed to communicate while interacting with the verifier (though they may share a random string before such an interaction begins). More precisely, Ben-Or et al. prove that every language having a multi-prover interactive proof also has a perfect zero-knowledge multi-prover interactive proof (again, without relying on intractability assumptions). The result of [9] was subsequently improved in a number of papers [5,19,29].

- **PCP.** Kilian et al. [28] study statistical zero knowledge in the model of *probabilistically checkable proofs* (PCPs) [2–4], where the verifier has oracle access to a string. Essentially, the oracle string can be thought of as a stateless prover: the answer to a query depends only on the query itself, but not any other queries that were previously made. Building on results implicit in [19], Kilian et al. showed two main theorems. First, every language in **NEXP** has a PCP that is statistical zero knowledge against verifiers that make at most any polynomial number of queries to the PCP. Second, every language in **NP** has, for every constant $c > 0$, a PCP that is statistically zero knowledge against verifiers that make at most $\mathsf{k}(n) := n^c$ queries to the PCP.
  Subsequent works [24–26,31] provided simplifications (giving alternative constructions or simplifying that of [28]) and limitations (showing that for languages in **NP** one cannot efficiently sample the oracle if one seeks statistical zero knowledge against verifiers that make at most a polynomial number of queries).
- **IPCP.** Goyal et al. [21] study statistical zero knowledge in the model of *interactive PCPs* (IPCPs) [27], where the verifier interacts with two provers of which one is restricted to be an oracle. Goyal et al. prove that every language in **NP** has a constant-round interactive PCP that is statistical zero knowledge against verifiers that make at most any polynomial number of queries to the PCP, and where both provers' strategies can be implemented efficiently as a function of the instance and the witness.

A Limitation of Prior Work. PCPs with quasilinear-size proof length, but without zero knowledge, are known: for every language $\mathscr{L}$ in $\mathbf{NTIME}(T(n))$, there is a PCP with proof length $\tilde{O}(T(n))$ and query complexity $O(1)$ [14,15,17,32]. On the other hand, no such result for statistical zero knowledge PCPs is known: even when applied to PCPs of length $\tilde{O}(T(n))$, [28]'s result and followup improvements yields a proof length that is polynomial in $T(n) \cdot \mathsf{k}(n)$, where $\mathsf{k}(n)$, known as the *knowledge bound*, is a bound on the number of queries by any verifier (see Sect. 4.1 for further discussion). We thus ask the following question: are there statistical zero knowledge PCPs with proof length quasilinear in $T(n) + \mathsf{k}(n)$?

1.1 Our Contributions

We do not answer the above question in the PCP model, but we give a positive answer in a closely related model that can be thought of as a "2-round PCP", which we call *duplex PCP* (DPCP). At a high level, a DPCP works as follows: the prover first sends an oracle string π_0 to the verifier, just as in a PCP; then, the verifier sends a message ρ to the prover; finally, the prover answers with a second oracle string π_1; the verifier may query both oracles, and then accept or reject. In other words, a DPCP is merely a 2-round interactive proof in which the prover sends oracle strings rather than messages. We prove the following theorem:

Theorem 1 (see Theorem 4 for formal statement). *For every language $\mathscr{L}$ in* $\mathbf{NTIME}(T) \cap \mathbf{NP}$ *and polynomially-bounded knowledge bound* k *there exists a DPCP system satisfying the following:*

- *the proof length (in fact, also the prover running time) is quasilinear in* $n + T(n) + \mathsf{k}(n)$*;*
- *the query complexity is polynomial in* $\log(T(n) + \mathsf{k}(n))$*;*
- *the verifier running time is polynomial in* $n + \log(T(n) + \mathsf{k}(n))$*;*
- *perfect zero knowledge holds against any verifier that makes at most* $\mathsf{k}(n)$ *adaptive queries (in total to both oracles);*
- *the soundness error is* $1/2$ *(and can be reduced by repetition to* $2^{-\lambda}$ *while preserving perfect zero knowledge, provided that the number of queries does not exceed* $\mathsf{k}(n)$*).*

Moreover, similarly to the PCPs of [28], the DPCP system that we construct is in fact not only sound but is also a *proof of knowledge* [7]; however, in contrast to [28], the DPCP verifier is *non-adaptive*, in the sense that the query locations depend only on the verifier's random tape.

Perhaps the main difference between our construction and prior work is the techniques that we use. While previous works use the PCP Theorem as a black box, compiling a PCP into a zero knowledge PCP by using *locking schemes* [28], we use certain *algebraic* properties of a specific family of PCPs to guarantee zero knowledge. In comparison to the generic approach, we are more specific, but the addition of zero knowledge essentially comes "for free" when compared to the corresponding constructions without zero knowledge. (In contrast, [28] achieves a proof length of $\Omega(\mathsf{k}(n)^6 \cdot \mathsf{l}(n)^c)$, for some large enough c, when starting from a PCP with proof length $\mathsf{l}(n)$.)

DPCP vs IPCP. Duplex PCPs are an alternative to interactive PCPs that combine PCPs and interaction. In a DPCP, the verifier gets an oracle string from the prover, replies with a message, and then gets another oracle string from the prover; in an IPCP, the verifier gets an oracle string from the prover, and then engages in an interactive proof with him.

Both [21] and our work are similar in that both address aspects that we do not know how to address in the PCP model, and resort to studying alternative models, i.e., IPCP and DPCP respectively. The two works however give different flavors of results: [21] obtain IPCPs that are zero knowledge against verifiers that ask at most any polynomial number of queries $\mathsf{k}(n)$ but their oracle is of polynomial size in $\mathsf{k}(n)$ (actually, of exponential size but with a polynomial-size circuit describing it); on the other hand, our work obtains DPCPs that are zero knowledge against verifiers that ask at most a fixed polynomial number of queries $\mathsf{k}(n)$ and our oracles are of quasilinear size in $\mathsf{k}(n)$.

Finally, we note that our construction can be also cast as an IPCP, because the knowledge bound $\mathsf{k}(n)$ holds only for the first oracle, i.e., perfect zero knowledge is preserved even if the verifier reads the second oracle in full. This provides a result on a 2-round IPCP incomparable to [21]'s 4-round IPCP.

On the Minimal Computational Gap Between Prover and Verifier Needed for Zero Knowledge. IP and MIP systems assume a computational gap between prover and verifier. The prover is allowed (and often assumed) to be computationally unbounded and the verifier is polynomially bounded. An intriguing corollary of our theorem is that the computational gap between prover and verifier can be drastically reduced, to a mere polylogarithmic one. Namely, suppose that we wish to create zero-knowledge systems in which the verifier runs in time $\mathsf{tv}(n)$; in the model above, as long as $\mathsf{tp}(n) > \mathsf{tv}(n) \cdot (\log \mathsf{tv}(n))^c$ for an absolute constant c, then perfect zero knowledge with a small soundness error can be obtained under no intractability assumptions. (See Corollary 1 for a formal statement.)

2 Preliminaries

Functions and Distributions. We use $f\colon D \to R$ to denote a function with domain D and range R; given a subset $\tilde{D}$ of D, we use $f|_{\tilde{D}}$ to denote the restriction of f to $\tilde{D}$. Given a distribution $\mathcal{D}$, we write $x \leftarrow \mathcal{D}$ to denote that x is sampled according to $\mathcal{D}$.

Distances. A distance measure is a function $\Delta\colon \Sigma^n \times \Sigma^n \to [0,1]$ such that for all $x, y, z \in \Sigma^n$: (i) $\Delta(x,x) = 0$, (ii) $\Delta(x,y) = \Delta(y,x)$, and (iii) $\Delta(x,y) \leq \Delta(x,z) + \Delta(z,y)$. For example, the *relative Hamming distance* over alphabet Σ is a distance measure: $\Delta_\Sigma^{\mathrm{Ham}}(x,y) := |\{i \mid x_i \neq y_i\}|/n$. We extend Δ to distances of strings to sets: given $x \in \Sigma^n$ and $S \subseteq \Sigma^n$, we define $\Delta(x,S) := \min_{y \in S} \Delta(x,y)$ (or 1 if S is empty). We say that a string x is ϵ-close to another string y if $\Delta(x,y) \leq \epsilon$, and ϵ-far from y if $\Delta(x,y) > \epsilon$; similar terminology applies for a string x and a set S.

Fields and Polynomials. We denote by $\mathbb{F}$ a finite field, by $\mathbb{F}_q$ the field of size q, and by $\mathscr{F}$ the set of all finite fields. We denote by $\mathbb{F}[X_1, \ldots, X_m]$ the ring of polynomials in m variables over $\mathbb{F}$; given a polynomial P in $\mathbb{F}[X_1, \ldots, X_m]$, $\deg_{X_i}(P)$ is the degree of P in the variable X_i; the total degree of P is the sum of all of these individual degrees.

Linear Spaces. Given $n \in \mathbb{N}$, a subset S of $\mathbb{F}^n$ is an $\mathbb{F}$-linear space if $\alpha x + \beta y \in S$ for all $\alpha, \beta \in \mathbb{F}$ and $x, y \in S$.

Languages and Relations. We denote by $\mathscr{R}$ a relation consisting of pairs $(\mathtt{x}, \mathtt{w})$, where $\mathtt{x}$ is the *instance* and $\mathtt{w}$ is the *witness*. We denote by $\mathrm{Lan}(\mathscr{R})$ the language corresponding to $\mathscr{R}$, and by $\mathscr{R}|_{\mathtt{x}}$ the set of witnesses in $\mathscr{R}$ for $\mathtt{x}$.

Complexity Classes. We write complexity classes in bold capital letters: **NP**, **PSPACE**, **NEXP**, and so on. We take a "relation-centric" point of view: we view **NTIME** as a class of relations rather than as the class of the corresponding languages; we thus may write things like "let $\mathscr{R}$ be in **NP**". If $\mathscr{R}$ is in **NTIME**(T), we fix an arbitrary machine $M_{\mathscr{R}}$ that decides $\mathscr{R}$ in time $T(n)$, i.e., $M_{\mathscr{R}}(\mathtt{x}, \mathtt{w})$ always halts after $T(|\mathtt{x}|)$ steps and $M_{\mathscr{R}}(\mathtt{x}, \mathtt{w}) = 1$ if and only

if $(\mathtt{x}, \mathtt{w}) \in \mathscr{R}$; we then say that $M_{\mathscr{R}}$ decides $\mathscr{R}$ (or $\mathrm{Lan}(\mathscr{R})$). Throughout, we assume that $T(n) \geq n$.

Codes. An error correcting code C is a set of functions $w\colon H \to \Sigma$, where H, Σ are finite sets. The message length of C is $n := \log_{|\Sigma|} |C|$, its block length is $\ell := |H|$, its rate is $\rho := n/\ell$, its (minimum) distance is $d := \min\{\Delta(w,z) \mid w, z \in C,\ w \neq z\}$ when Δ is the (absolute) Hamming distance, and its (minimum) relative distance is $\delta := d/\ell$. Given a code family $\mathscr{C}$, we denote by $\mathrm{Rel}(\mathscr{C})$ the relation that naturally corresponds to $\mathscr{C}$, i.e., $\{(C, w) \mid C \in \mathscr{C},\ w \in C\}$. A code C is linear if Σ is a finite field and C is a Σ-linear space in Σ^{ℓ}; we denote by $\dim(C)$ the dimension of C when viewed as a linear space. A code C is t-wise independent if, for every subset I of $[\ell]$ with cardinality t, the distribution of $w|_I$ (viewed as a string) for a random $w \in C$ equals the uniform distribution on Σ^t.

Random Shifts. We later use the following folklore claim about distance preservation for random shifts in linear spaces; for completeness, we include its short proof.

Claim. Let n be in $\mathbb{N}$, $\mathbb{F}$ a finite field, S an $\mathbb{F}$-linear space in $\mathbb{F}^n$, and $x, y \in \mathbb{F}^n$. If x is ϵ-far from S, then $\alpha x + y$ is $\epsilon/2$-far from S, with probability $1 - |\mathbb{F}|^{-1}$ over a random $\alpha \in \mathbb{F}$. (Distances are relative Hamming distances.)

Proof. Suppose, by way of contradiction, that there exist $\alpha_1, \alpha_2 \in \mathbb{F}$ and $y_1, y_2 \in S$ with $\alpha_1 \neq \alpha_2$ such that, for every $i \in \{1, 2\}$, $\alpha_i x + y$ is $\epsilon/2$ close to y_i. Then, by the triangle inequality, $z := y_1 - y_2$ is ϵ-close to $(\alpha_1 x + y) - (\alpha_2 x + y) = (\alpha_1 - \alpha_2)x$. We conclude that x is ϵ-close to $\frac{1}{\alpha_1 - \alpha_2} z \in S$, a contradiction.

2.1 Probabilistically Checkable Proofs

A *PCP system* [2–4] for a relation $\mathscr{R}$ is a tuple $\mathsf{PCP} = (P, V)$ that works as follows.

- The *prover* P is a probabilistic algorithm that, given as input an instance-witness pair $(\mathtt{x}, \mathtt{w})$ with $n := |\mathtt{x}|$, outputs a proof $\pi\colon D(n) \to \Sigma(n)$, where both $D(n)$ and $\Sigma(n)$ are finite sets.
- The *verifier* V is a probabilistic oracle algorithm that, given as input an instance $\mathtt{x}$ with $n := |\mathtt{x}|$ and with oracle access to a proof $\pi\colon D(n) \to \Sigma(n)$, queries π at a few locations and then outputs a bit.

The system PCP has (perfect) completeness and soundness error $\mathsf{e}(n)$ if the following two conditions hold. (Below, we explicitly denote the prover's and verifier's randomness as r_P and r_V.)

Completeness: For every instance-witness pair $(\mathtt{x}, \mathtt{w})$ in the relation $\mathscr{R}$,

$$\Pr_{r_P, r_V}\left[V^{P(\mathtt{x},\mathtt{w};r_P)}(\mathtt{x}; r_V) = 1\right] = 1\,.$$

Soundness: For every instance $\mathbb{x}$ not in the language $\mathrm{Lan}(\mathscr{R})$ and proof $\pi\colon D(n) \to \Sigma(n)$,

$$\Pr_{r_V}[V^{\pi}(\mathbb{x}; r_V) = 1] \leq \mathsf{e}(n) .$$

A relation $\mathscr{R}$ belongs to the complexity class $\mathbf{PCP}[\mathsf{a}, \mathsf{l}, \mathsf{q}, \mathsf{e}, \mathsf{tp}, \mathsf{tv}]$ if there is a PCP system for $\mathscr{R}$ in which:

- the answer alphabet (i.e., $\Sigma(n)$) is $\mathsf{a}(n)$,
- the proof length over that alphabet (i.e., $|D(n)|$) is at most $\mathsf{l}(n)$,
- the verifier queries the proof in at most $\mathsf{q}(n)$ locations,
- the soundness error is $\mathsf{e}(n)$,
- the prover runs in time $\mathsf{tp}(n)$, and
- the verifier runs in time $\mathsf{tv}(n)$.

Finally, we add the symbol na in the square brackets (i.e., we write $\mathbf{PCP}[\ldots, \mathrm{na}]$) if the queries to the proof are non-adaptive (i.e., the queried locations only depend on the verifier's inputs).

2.2 Probabilistically Checkable Proofs of Proximity

A *PCPP system* [12,18] for a relation $\mathscr{R}$ is a tuple $\mathsf{PCPP} = (P, V)$ that works as follows.

- The *prover* P is a probabilistic algorithm that, given as input an instance-witness pair $(\mathbb{x}, \mathbb{w})$ with $n := |\mathbb{x}|$, outputs a proof $\pi\colon D(n) \to \Sigma(n)$, where both $D(n)$ and $\Sigma(n)$ are finite sets.
- The *verifier* V is a probabilistic oracle algorithm that, given as input an instance $\mathbb{x}$ with $n := |\mathbb{x}|$ and with oracle access to a witness $\mathbb{w}$ and proof $\pi\colon D(n) \to \Sigma(n)$, queries $\mathbb{w}$ and π at a few locations and then outputs a bit.

The system PCPP has (perfect) completeness, soundness error e, distance measure Δ, and proximity parameter d if the following two conditions hold. (Below, we explicitly denote the prover's and verifier's randomness as r_P and r_V.)

Completeness: For every instance-witness pair $(\mathbb{x}, \mathbb{w})$ in the relation $\mathscr{R}$,

$$\Pr_{r_P, r_V}\left[V^{(\mathbb{w}, P(\mathbb{x}, \mathbb{w}; r_P))}(\mathbb{x}; r_V) = 1\right] = 1 .$$

Soundness: For every instance-witness pair $(\mathbb{x}, \mathbb{w})$, perhaps not in the language, such that $\Delta(\mathbb{w}, \mathscr{R}|_{\mathbb{x}}) \geq \mathsf{d}(n)$ and proof $\pi\colon D(n) \to \Sigma(n)$,

$$\Pr_{r_V}\left[V^{(\mathbb{w}, \pi)}(\mathbb{x}; r_V) = 1\right] \leq \mathsf{e}(n) .$$

A relation $\mathscr{R}$ belongs to the complexity class $\mathbf{PCPP}[\mathsf{a}, \mathsf{l}, \mathsf{q}, \Delta, \mathsf{d}, \mathsf{e}, \mathsf{tp}, \mathsf{tv}]$ if there is a PCPP system for $\mathscr{R}$ in which:

- the answer alphabet (i.e., $\Sigma(n)$) is $\mathsf{a}(n)$,
- the proof length over that alphabet (i.e., $|D(n)|$) is at most $\mathsf{l}(n)$,

- the verifier queries the two oracles (codeword and proof) in at most $\mathsf{q}(n)$ locations (in total),
- the distance measure is Δ,
- the proximity parameter is $\mathsf{d}(n)$,
- the soundness error is $\mathsf{e}(n)$,
- the prover runs in time $\mathsf{tp}(n)$, and
- the verifier runs in time $\mathsf{tv}(n)$.

Finally, we add the symbol na in the square brackets (i.e., we write $\mathbf{PCPP}[\ldots, \mathrm{na}]$) if the queries to the oracles are non-adaptive (i.e., the queried locations only depend on the verifier's inputs).

2.3 Zero Knowledge PCPs

The notion of zero knowledge for PCPs was first considered in [19,28]. A PCP system $\mathsf{PCP} = (P, V)$ for a relation $\mathscr{R}$ has *perfect zero knowledge with knowledge bound* k if there exists an expected-polynomial-time probabilistic algorithm S such that, for every k-query polynomial-time probabilistic oracle algorithm $\tilde{V}$, the following two distribution families are identical:

$$\{S(\tilde{V}, \mathbb{x})\}_{(\mathbb{x},\mathbb{w})\in\mathscr{R}} \quad \text{and} \quad \{\mathrm{PCPView}(\tilde{V}, P, \mathbb{x}, \mathbb{w})\}_{(\mathbb{x},\mathbb{w})\in\mathscr{R}} \ ,$$

where $\mathrm{PCPView}(\tilde{V}, \pi, \mathbb{x}, \mathbb{w})$ is the view of $\tilde{V}$ in its execution when given input $\mathbb{x}$ and oracle access to $\pi := P(\mathbb{x}, \mathbb{w})$. The definition of statistical and computational zero knowledge (with knowledge bound k) are similar: rather than identical, the two distribution families are required to be statistically and computationally close (as $|\mathbb{x}|$ grows), respectively.

A relation $\mathscr{R}$ belongs to the complexity class $\mathbf{PCP}_{\mathrm{pzk}}[\mathsf{a}, \mathsf{l}, \mathsf{q}, \mathsf{e}, \mathsf{tp}, \mathsf{tv}, \mathsf{k}]$ if there exists a PCP system for $\mathscr{R}$ that (i) puts $\mathscr{R}$ in $\mathbf{PCP}[\mathsf{a}, \mathsf{l}, \mathsf{q}, \mathsf{e}, \mathsf{tp}, \mathsf{tv}]$, and (ii) has perfect zero knowledge with knowledge bound k; as for $\mathbf{PCP}$, we add the symbol na in the square brackets of $\mathbf{PCP}_{\mathrm{pzk}}$ if the queries to the proof are non-adaptive. The complexity classes $\mathbf{PCP}_{\mathrm{szk}}$ and $\mathbf{PCP}_{\mathrm{czk}}$ are similarly defined for statistical and computational zero knowledge.

The KPT Result. Kilian, Petrank, and Tardos proved the following theorem:

Theorem 2 [28]**.** *For every polynomial time function* $T\colon \mathbb{N} \to \mathbb{N}$*, polynomial security function* $\lambda\colon \mathbb{N} \to \mathbb{N}$*, and polynomial knowledge bound function* $\mathsf{k}\colon \mathbb{N} \to \mathbb{N}$*,*

$$\mathbf{NTIME}(T) \subseteq \mathbf{PCP}_{\mathrm{szk}} \begin{bmatrix} \mathsf{a} &= \mathbb{F}_{2^{\mathrm{poly}(\lambda)}} \\ \mathsf{l} &= \mathrm{poly}(T, \mathsf{k}) \\ \mathsf{q} &= \mathrm{poly}(\lambda) \\ \mathsf{e} &= 2^{-\lambda} \\ \mathsf{tp} &= \mathrm{poly}(\lambda, T) \\ \mathsf{tv} &= \mathrm{poly}(\lambda, T, \mathsf{k}) \\ \mathsf{k} & \end{bmatrix} .$$

Remark 1. We make two remarks: (i) the symbol na does not appear above because [28]'s construction relies on adaptively querying the proof; (ii) inspection of [28]'s construction reveals that $\mathsf{l}(n) \geq \mathrm{poly}(T(n)) \cdot \mathsf{k}(n)^6$.

2.4 Reed–Muller and Reed–Solomon Codes

We define Reed–Muller and Reed–Solomon codes, as well as their "vanishing" variants [15]; all of these are linear codes. We then state a theorem about PCPPs for certain families of RS codes.

RM Codes. Let $\mathbb{F}$ be a finite field, H, V subsets of $\mathbb{F}$, m a positive integer, and ϱ a constant in $(0,1]$; ϱ is called the *fractional degree*. The Reed–Muller code with parameters $\mathbb{F}, H, m, \varrho$ is $\mathsf{RM}[\mathbb{F}, H, m, \varrho] := \{w\colon H^m \to \mathbb{F} \mid \max_{i\in[m]} \deg_{X_i}(w) < \varrho|H|\}$; its message length is $n = (\varrho|H|)^m$, block length is $\ell = |H|^m$, rate is $\rho = \varrho^m$, and relative distance is $\delta = 1 - \varrho$. The vanishing Reed–Muller code with parameters $\mathbb{F}, H, m, \varrho, V$ is $\mathsf{VRM}[\mathbb{F}, H, m, \varrho, V] := \{w \in \mathsf{RM}[\mathbb{F}, H, m, \varrho] \mid w(V^m) = \{0\}\}$; it is a subcode of $\mathsf{RM}[\mathbb{F}, H, m, \varrho]$.

RS Codes. Let $\mathbb{F}$ be a finite field, H, V subsets of $\mathbb{F}$, and ϱ a constant in $(0,1]$. The Reed–Solomon code with parameters $\mathbb{F}, H, \varrho$ is $\mathsf{RS}[\mathbb{F}, H, \varrho] := \mathsf{RM}[\mathbb{F}, H, 1, \varrho]$. The vanishing Reed–Solomon code with parameters $\mathbb{F}, H, \varrho, V$ is $\mathsf{VRS}[\mathbb{F}, H, \varrho, V] := \{w \in \mathsf{RS}[\mathbb{F}, H, \varrho] \mid w(V) = \{0\}\}$.

Two RS Code Families and Their PCPPs. Given $\varrho \in (0,1]$, we denote by: (i) $\mathcal{RS}^*_\varrho$ the set of Reed–Solomon codes $\mathsf{RS}[\mathbb{F}, H, \varrho]$ for which $\mathbb{F}$ has characteristic 2 and H is an $\mathbb{F}_2$-affine space; and (ii) $\mathcal{VRS}^*_\varrho$ the set of vanishing Reed–Solomon codes $\mathsf{VRS}[\mathbb{F}, H, \varrho, V]$ for which $\mathbb{F}$ has characteristic 2 and H is an $\mathbb{F}_2$-affine space. The following theorem is from [10,15] (the prover running time is shown in [10] and the other parameters in [15]).

Theorem 3. *For every security function $\lambda\colon \mathbb{N} \to \mathbb{N}$, $\varrho \in (0,1)$, and $s > 0$,*

$$\mathrm{Rel}(\mathcal{RS}^*_\varrho)\,,\ \mathrm{Rel}(\mathcal{VRS}^*_\varrho) \in \mathbf{PCPP} \begin{bmatrix} \mathsf{a} & = \mathbb{F}_{2^{s+\log \ell}} \\ \mathsf{l} & = \tilde{O}(\ell) \\ \mathsf{q} & = \lambda \cdot \mathrm{polylog}(\ell) \\ \Delta & = \Delta^{\mathrm{Ham}}_{\mathsf{a}} \\ \mathsf{d} & = \varrho/2 \\ \mathsf{e} & = 2^{-\lambda} \\ \mathsf{tp} & = \mathrm{poly}(s) \cdot \tilde{O}(\ell) \\ \mathsf{tv} & = \lambda \cdot \mathrm{poly}(s + \log \ell) \\ \mathrm{na} & \end{bmatrix}.$$

We will also require the following folklore claim, whose correctness can be proved by induction on m:

Claim. Let $\mathbb{F}$ be a finite field, H, V subsets of $\mathbb{F}$ with $H \cap V = \emptyset$, m a positive integer, and t a positive integer not exceeding $|H| - |V|$. Then $\mathsf{VRM}[\mathbb{F}, H, m, \frac{|V|+t}{|H|}, V]$ is t-wise independent.

3 Duplex PCPs

We define duplex PCPs, and then define notions of zero knowledge for this model. Our main theorem is the construction of a duplex PCP with certain

parameters; see Sect. 4. The difference between a PCP and a duplex PCP is that all provers (both honest and malicious) produce two proof oracles rather than one: the prover produces a proof π_0; then the verifier sends a message ρ to the prover; then the prover produces another proof π_1; finally the verifier queries both π_0 and π_1 and either accepts or rejects. (Thus, a PCP is a special case of a duplex PCP, but not vice versa.) More precisely, a *duplex PCP system* for a relation $\mathscr{R}$ is a tuple $\mathsf{DPCP} = (P, V)$ that works as follows.

- The *prover* P is a pair (P_0, P_1) of probabilistic algorithms, with shared randomness, where: (a) given as input an instance-witness pair $(\mathtt{x}, \mathtt{w})$ with $n := |\mathtt{x}|$, P_0 outputs a proof $\pi_0 \colon D_0(n) \to \Sigma(n)$; (b) given as input $(\mathtt{x}, \mathtt{w})$ and the verifier's message ρ (see below), P_1 outputs a proof $\pi_1 \colon D_1(n) \to \Sigma(n)$. Here $D_0(n), D_1(n), \Sigma(n)$ are finite sets.
- The *verifier* V is a pair (V_0, V_1) of probabilistic algorithms, with shared randomness, where: (a) given as input an instance $\mathtt{x}$ with $n := |\mathtt{x}|$, V_0 outputs a message ρ; (b) given as input $\mathtt{x}$ and with oracle access to proofs $\pi_0 \colon D_0(n) \to \Sigma(n)$ and $\pi_1 \colon D_1(n) \to \Sigma(n)$, V_1 queries π_0 and π_1 at a few locations and then outputs a bit.

The system DPCP has (perfect) completeness and soundness error $\mathsf{e}(n)$ if the following two conditions hold. (Below, we explicitly denote the prover's and verifier's randomness as r_P and r_V.)

Completeness: For every instance-witness pair $(\mathtt{x}, \mathtt{w})$ in the relation $\mathscr{R}$,

$$\Pr_{r_P, r_V}\left[V_1^{\pi_0,\pi_1}(\mathtt{x}; r_V) = 1 \;\middle|\; \begin{array}{r} \pi_0 \leftarrow P_0(\mathtt{x}, \mathtt{w}; r_P) \\ \rho \leftarrow V_0(\mathtt{x}; r_V) \\ \pi_1 \leftarrow P_1(\mathtt{x}, \mathtt{w}, \rho; r_P) \end{array} \right] = 1 \ .$$

Soundness: For every instance $\mathtt{x}$ not in the language $\mathrm{Lan}(\mathscr{R})$ and pair of algorithms $\tilde{P} = (\tilde{P}_0, \tilde{P}_1)$,

$$\Pr_{r_V}\left[V_1^{\pi_0,\pi_1}(\mathtt{x}; r_V) = 1 \;\middle|\; \begin{array}{r} \pi_0 \leftarrow \tilde{P}_0 \\ \rho \leftarrow V_0(\mathtt{x}; r_V) \\ \pi_1 \leftarrow \tilde{P}_1(\rho) \end{array} \right] \leq \mathsf{e}(n) \ .$$

A relation $\mathscr{R}$ belongs to the complexity class $\mathbf{DPCP}[\mathsf{a}, \mathsf{l}, \mathsf{q}, \mathsf{e}, \mathsf{tp}, \mathsf{tv}]$ if there is a DPCP system for $\mathscr{R}$ in which:

- the answer alphabet (i.e., $\Sigma(n)$) is $\mathsf{a}(n)$,
- the proof length over that alphabet (i.e., $(|D_0(n)| + |D_1(n)|)$) is at most $\mathsf{l}(n)$,
- the verifier queries the two proofs in at most $\mathsf{q}(n)$ locations (in total),
- the soundness error is $\mathsf{e}(n)$,
- the prover runs in time $\mathsf{tp}(n)$, and
- the verifier runs in time $\mathsf{tv}(n)$.

Finally, we add the symbol na in the square brackets (i.e., we write $\mathbf{DPCP}[\ldots, \mathrm{na}]$) if the queries to the proof are non-adaptive (i.e., the queried locations only depend on the verifier's inputs).

Zero Knowledge. A DPCP system $\mathsf{DPCP} = (P, V)$ for a relation $\mathscr{R}$ has *perfect zero knowledge with knowledge bound* k if there exists an expected-polynomial-time probabilistic algorithm S such that for every pair of polynomial-time probabilistic oracle algorithms $\tilde{V} := (\tilde{V}_0, \tilde{V}_1)$ the following two distribution families are identical:

$$\{S(\tilde{V}, \mathtt{x})\}_{(\mathtt{x},\mathtt{w})\in\mathscr{R}} \quad \text{and} \quad \{\mathrm{DPCPView}(\mathsf{k}, \tilde{V}, P, \mathtt{x}, \mathtt{w})\}_{(\mathtt{x},\mathtt{w})\in\mathscr{R}} \ ,$$

where $\mathrm{DPCPView}(\mathsf{k}, \tilde{V}, P, \mathtt{x}, \mathtt{w})$ is the view of $\tilde{V}_1$ in its execution when given input $\mathtt{x}$ and when allowed to make a total of $\mathsf{k}(n)$ adaptive queries to π_0, π_1, where $\pi_0 := P_0(\mathtt{x}, \mathtt{w})$ and $\pi_1 := P_1(\mathtt{x}, \mathtt{w}, \tilde{V}_0^{\pi_0}(\mathtt{x}))$. (As above, P_0, P_1 share the same randomness r_P; ditto for $\tilde{V}_0, \tilde{V}_1$.) The definition of statistical and computational zero knowledge (with knowledge bound k) are similar: rather than identical, the two distribution families are required to be statistically and computationally close (as $|\mathtt{x}|$ grows), respectively.

4 Main Theorem

The main result of this paper is the following.

Theorem 4. *For every polynomial time function $T\colon \mathbb{N} \to \mathbb{N}$, polynomial knowledge bound function $\mathsf{k}\colon \mathbb{N} \to \mathbb{N}$,*

$$\mathbf{NTIME}(T) \subseteq \mathbf{DPCP}_{\mathrm{pzk}} \begin{bmatrix} \mathsf{a} & = \mathbb{F}_{2^{O(\log(T+\mathsf{k}))}} \\ \mathsf{l} & = \tilde{O}(T+\mathsf{k}) \\ \mathsf{q} & = \mathrm{polylog}(T+\mathsf{k}) \\ \mathsf{e} & = \frac{1}{2} \\ \mathsf{tp} & = \mathrm{poly}(n) \cdot \tilde{O}(T+\mathsf{k}) \\ \mathsf{tv} & = \mathrm{poly}(n + \log(T+\mathsf{k})) \\ \mathsf{k} & \\ \mathrm{na} & \end{bmatrix}$$

A Corollary. The theorem above implies that, fixing T, the prover running time is merely quasilinear in the knowledge bound k, while the verifier running time increases only polylogarithmically in k. This leads to an intriguing corollary: a poly-logarithmic computational overhead of the prover over the verifier is all that is needed to maintain perfect zero knowledge in the duplex PCP model. We state this formally next.

Corollary 1. *For every polynomial time function $T\colon \mathbb{N} \to \mathbb{N}$ and relation $\mathscr{R} \in \mathbf{NTIME}(T)$, there is a constant c such that, for every function $\mathsf{tv}\colon \mathbb{N} \to \mathbb{N}$ with $\mathsf{tv}(n) \geq n \cdot (\log T(n))^c$, there is a DPCP system with:*

- *completeness* 1 *and soundness* $2^{-\mathsf{tv}(n)/\mathrm{polylog}(T(n))}$*;*
- *perfect zero knowledge;*
- *the verifier running time is* $\mathsf{tv}(n)$ *and prover running time is* $\mathsf{tp}(n) := \max\{T(n) \cdot (\log T(n))^c, \mathsf{tv}(n) \cdot (\log \mathsf{tv}(n))^c\}$.

The verifier has no limitations other than a bound on its running time (its query complexity can be as large as $\mathsf{tv}(n)$*).*

4.1 Proof Sketch

Let $\mathscr{R}$ be a relation in **NP**, and let $(\mathtt{x}, \mathtt{w})$ be an instance-witness pair in $\mathscr{R}$. The prover and verifier both know $\mathtt{x}$, while the prover also knows $\mathtt{w}$. The prover wishes to convince the verifier that he knows a witness $\mathtt{w}$ for $\mathtt{x}$, in such a way that the verifier does not learn anything about $\mathtt{w}$ (beyond what can be inferred from the prover's claim).

The KPT Approach. We introduce our ideas by contrasting them with those of [28]. Suppose that the prover wishes to convince the verifier by sending him a PCP proof $\pi = \pi(\mathtt{w})$ such that any k values in π do not reveal anything about $\mathtt{w}$. Loosely speaking, [28] (building on [19]) provide a probabilistic transformation that maps the PCP proof π to a new proof π', in which each bit of π is "hidden" amongst many bits of π'. The main tool employed in the transformation is a *locking scheme*, and its use imposes certain limitations: (i) the new proof π' is poly(k) larger than the original one (k^6 by inspection of [19,28]); (ii) zero knowledge holds only statistically, but not perfectly, because a malicious verifier can be "lucky" and obtain information on the bit of π being locked with fewer queries to π' than expected.

Our Approach (Ideally). We take a different approach: apply a "local" PCP to a "random" witness, as we now explain. Suppose that $\pi = \pi(\mathtt{w})$ is (t, k)-*local*, i.e., any k positions of the PCP proof π jointly depend on at most t positions of the witness $\mathtt{w}$. Note that, even if π is (t, k)-local, a single bit of π can still leak information about $\mathtt{w}$. So suppose further that the relation $\mathscr{R}$ is t-*randomizable*: given $(\mathtt{x}, \mathtt{w}) \in \mathscr{R}$, one can efficiently sample a witness $\mathtt{w}'$ from a t-wise independent subset of the set of witnesses for $\mathtt{x}$. In such a case, the prover can produce a zero-knowledge PCP as follows: (1) sample a witness $\mathtt{w}'$ from the t-wise independent subset; then (2) send to the verifier the PCP proof $\pi = \pi(\mathtt{w}')$. Indeed, the locality of π ensures that seeing any k indices of π reveals nothing about $\mathtt{w}$, because these k indices are a function of t random bits. In sum, if we had a (t, k)-local PCP for a t-randomizable relation $\mathscr{R}$, then we could obtain a PCP for $\mathscr{R}$ that is zero knowledge against verifiers that ask at most k queries.

Our Approach (in Reality). Unfortunately, we do not know how to obtain local PCPs for randomizable relations. However, we are able to obtain "partially local" *duplex* PCPs for certain randomizable relations, and also show that **NTIME** can be efficiently reduced to these randomizable relations, as we now explain.

Our starting point are *algebraic PCPs*: certain PCPs that prove satisfiability of *algebraic problems* (APs) [34]. Numerous known PCP constructions can be viewed as algebraic PCPs. Informally, in this work we make two basic observations: (i) algebraic PCPs exist for certain randomizable relations; and (ii) an algebraic PCP proof can be split in two parts, one part is local, while the other part is not local but enjoys convenient linear algebraic properties that, nevertheless, enable us to hide information about the witness, in the duplex PCP model. (Recall that, in the duplex PCP model, the prover produces a proof π_0; then the

verifier sends a message ρ to the prover; then the prover produces another proof π_1; finally the verifier queries both π_0 and π_1 and either accepts or rejects.)

In more detail, from a technical viewpoint, we proceed as follows. First, we introduce a family of constraint satisfaction problems (CSPs) called *linear algebraic CSPs*, and show that **NTIME** is efficiently reducible to *randomizable* linear algebraic CSPs. The reduction consists of two parts: we go through an intermediary that we call *group preserving algebraic problems* (GAPs), a special case of APs that we believe to be of independent interest for the study of algebraic PCPs. Second, we construct a duplex PCP system for randomizable linear algebraic CSPs that is zero knowledge against verifiers that ask at most a certain number of queries.

A Technical Piece: Zero-Knowledge Duplex PCPP for Low-Degreeness. Later sections address all of the above steps (see Sect. 4.2 for a roadmap of these), and for now we only sketch one of these steps. Above we mention that an algebraic PCP proof has two parts: a local part, and a non-local part. This latter part of the proof arises from a central component of many PCP proofs: a *PCP of proximity* (PCPP) [13,18] that facilitates low-degree testing. Informally, given a function $f\colon H \to \mathbb{F}$ and an integer d, a *PCPP for degree d* is a proof $\pi(f)$ that f is ϵ-close to an evaluation of a polynomial degree at most degree d. We explain how to transform a PCPP for low-degreeness into a duplex PCPP for low-degreeness that is zero knowledge against verifiers that make at most t queries.

The set C of functions $f\colon H \to \mathbb{F}$ that are evaluations of a polynomial of degree at most d is a subspace of $\mathbb{F}^{|H|}$. The basic idea is that, in order for the prover to convince the verifier that a function f is close to C, it suffices for the prover to convince the verifier that a *random offset* of f is close to C: one can verify that, for any $u\colon H \to \mathbb{F}$, if f is ϵ-far from C, then $\alpha f + u$ is $\epsilon/2$-far from C, with probability $1 - |\mathbb{F}|^{-1}$ over a random $\alpha \in \mathbb{F}$. Hence, we can let the duplex PCP work as follows: (i) the prover samples a witness $\mathbb{w}'$ from the t-wise independent subset, chooses a random $u \in C$, and sends $\pi_0 := (\mathbb{w}', u)$ to the verifier; (ii) the verifier sends to the prover a random $\alpha \in \mathbb{F}$; (iii) the prover sends $\pi_1 = (v, \pi(v))$ to the verifier, where $v := \alpha\mathbb{w}' + u$ and $\pi(v)$ is a PCPP for low-degreeness of v; (iv) the verifier runs the PCPP verifier on (v, π) to check that v is close to C, and then checks that $v_i = \alpha\mathbb{w}'_i + u_i$ for a few random indices i in $\{1, \ldots, |H|\}$.

Let us discuss the various properties of the duplex PCPP.

- Completeness: If $\mathbb{w} \in C$, then $\alpha\mathbb{w}' + u \in C$; therefore, the prover convinces the verifier.
- Zero-knowledge: If the verifier asks at most t queries, then he learns nothing about $\mathbb{w}$ because: $\pi_0 = (\mathbb{w}', u)$ contains $\mathbb{w}'$ sampled from a t-wise independent subset and u random in C; $\pi_1 = (v, \pi(v))$ is running the PCPP on a vector v that is random in C.
- Soundness: If v does equal $\alpha \cdot \mathbb{w} + u$, then the verifier rejects with high probability because v is far from C (and the PCPP verifier rejects π with high probability). If instead v does not equal $\alpha \cdot \mathbb{w} + u$, then the fact that v is

close to C does not prove anything about whether $\mathtt{w}$ is also close. So, in this case, we need to reason about the success probability of the verifier's linearity tests: if these pass with enough probability, then with high probability v is close to $\alpha\mathtt{w}+u$, which again suffices for our purpose. Overall, soundness holds.

Next, we discuss how the technical sections are organized, and how they come together to yield our main theorem.

4.2 Roadmap of the Rest of the Paper

The rest of the paper is dedicated to turn the above intuition into a more formal proof. To do so, we introduce various intermediate steps, as follows.

- In Sect. 5, we introduce *linear algebraic CSPs* (a family of constraint satisfaction problems), and then describe how to obtain a *canonical PCP* for any linear algebraic CSP.
- In Sect. 6, we introduce *randomizable* linear algebraic CSPs, a subfamily of linear algebraic CSPs; then we show that, for every randomizable linear algebraic CSP, we can convert the CSP's canonical PCP into a corresponding zero-knowledge duplex PCP, incurring only little overheads.
- In Sect. 7, we show an efficient reduction from **NTIME** to randomizable linear algebraic CSPs; along the way, we introduce a family of algebraic problems, having special symmetry properties, that we believe to be of independent interest (e.g., for studying other questions about PCPs).

Combining (i) the efficient reduction from **NTIME** to randomizable linear algebraic CSPs together with (ii) the zero-knowledge duplex PCP for such problems yields Theorem 4. In Sect. 8 we provide details about how these components are combined.

5 Linear Algebraic CSPs and Their Canonical PCPs

We introduce *linear algebraic CSPs*, a family of constraint satisfaction problems; then we describe how to obtain a *canonical PCP* for any linear algebraic CSP.

5.1 Linear Algebraic Constraint Satisfaction Problems

A constraint satisfaction problem asks whether, for a given "local" function g, there exists an input α such that $g(\alpha)$ is an "accepting" output. For example, in the case of 3-SAT with n variables and m clauses, the function g maps $\{0,1\}^n$ to $\{0,1\}^m$, and $g(\alpha)$ indicates which clauses are satisfied by $\alpha \in \{0,1\}^n$; hence α yields an accepting output if (and only if) $g(\alpha) = 1^m$. Below we introduce a family of constraint satisfaction problems whose domain and range are linear-algebraic objects, namely, linear error correcting codes.

We begin by providing the notion of locality that we use for g; we also provide two other notions, one for the efficiency of computing a single coordinate of g's output, and another for measuring g's "pseudorandomness".

Definition 1. *Let $g\colon \Sigma^n \to \Sigma^m$ be a function. We say that g is:*

- q-local *if for every $j \in [m]$ there exists $I_j \subseteq [n]$ with $|I_j| \leq q$ such that $g(\alpha)[j]$ (the j-th coordinate of $g(\alpha)$) depends only on $\alpha|_{I_j}$ (the restriction of α to I_j);*
- c-efficient *if there is a time c algorithm that, given j and $\alpha|_{I_j}$, computes the set I_j and value $g(\alpha)[j]$;*
- (γ, ϵ)-sampling *if $\Pr[\, I_j \cap I \neq \emptyset \mid j \leftarrow [m]\,] \leq \gamma$ for every $I \subseteq [n]$ with $|I|/n \leq \epsilon$.*

Next we introduce $\mathscr{R}_{\mathsf{LA}}$, the relation of **linear algebraic CSPs**:

Definition 2 ($\mathscr{R}_{\mathsf{LA}}$). *Given functions $f\colon \mathbb{N} \to \mathscr{F}$, $\ell, q, c\colon \mathbb{N} \to \mathbb{N}$, and $\rho, \delta, \gamma, \epsilon\colon \mathbb{N} \to (0,1]$, the relation*

$$\mathscr{R}_{\mathsf{LA}}[f, \ell, \rho, \delta, q, c, \gamma, \epsilon]$$

consists of instance-witness pairs $(\mathbb{x}, \mathbb{w})$ satisfying the following.

- *The instance $\mathbb{x}$ is a tuple $(1^n, C_\circ, C_\bullet, g)$ where:*
  - *$C_\circ, C_\bullet$ are linear error correcting codes with block lengths $\ell_\circ(n), \ell_\bullet(n)$ at most $\ell(n)$, each with rate at most $\rho(n)$ and relative distance at least $\delta(n)$ over the same field $f(n)$;*
  - *$g\colon f(n)^{\ell_\circ(n)} \to f(n)^{\ell_\bullet(n)}$ is a $q(n)$-local, $c(n)$-efficient, $(\gamma(n), \epsilon(n))$-sampling function;*
  - *$C_\bullet \cup g(C_\circ)$ has relative distance at least $\delta(n)$ (though may not be a linear space).*
- *The witness $\mathbb{w}$ is a tuple $(\alpha_\circ, \alpha_\bullet)$ where $\alpha_\circ \in f(n)^{\ell_\circ(n)}$ and $\alpha_\bullet \in f(n)^{\ell_\bullet(n)}$.*
- *The instance $\mathbb{x}$ and witness $\mathbb{w}$ jointly satisfy the following: $\alpha_\circ \in C_\circ$, $\alpha_\bullet \in C_\bullet$, and $g(\alpha_\circ) = \alpha_\bullet$.*

We prove a simple claim about instances not in the language $\mathrm{Lan}(\mathscr{R}_{\mathsf{LA}})$, which we use several times later on.

Claim. For every instance $\mathbb{x} = (1^n, C_\circ, C_\bullet, g)$ not in the language $\mathrm{Lan}(\mathscr{R}_{\mathsf{LA}})$ and (candidate) witness $\tilde{\mathbb{w}} = (\tilde{\alpha}_\circ, \tilde{\alpha}_\bullet) \in f(n)^{\ell_\circ(n)} \times f(n)^{\ell_\bullet(n)}$ at least one of the following holds:

- at least one of $\tilde{\alpha}_\circ$ and $\tilde{\alpha}_\bullet$ is ϵ-far in relative Hamming distance from $C_\circ$ or $C_\bullet$, respectively; or
- there exist $\alpha_\circ \in C_\circ$ and $\alpha_\bullet \in C_\bullet$ such that $\tilde{\alpha}_\circ$ and $\tilde{\alpha}_\bullet$ are ϵ-close to $\alpha_\circ$ and $\alpha_\bullet$, respectively, but $g(\alpha_\circ) \neq \alpha_\bullet$.

Proof. If neither of the two cases hold, then there exist $\alpha_\circ \in C_\circ$ and $\alpha_\bullet \in C_\bullet$ such that $g(\alpha_\circ) = \alpha_\bullet$. But then $(\alpha_\circ, \alpha_\bullet)$ is a satisfying assignment for $\mathbb{x}$, contradicting our assumption that $\mathbb{x}$ is not in the language $\mathrm{Lan}(\mathscr{R}_{\mathsf{LA}})$.

Finally we need notation for referring to codes appearing in instances of $\mathscr{R}_{\mathsf{LA}}$:

Definition 3. *Given $\mathscr{R} \subseteq \mathscr{R}_{\mathsf{LA}}$, we denote by*

- *$\mathscr{C}_{\mathscr{R},\circ}$ the set of codes C for which there is an instance $\mathbb{x} = (1^n, C_\circ, C_\bullet, g)$ in the relation $\mathscr{R}$ with $C = C_\circ$;*
- *$\mathscr{C}_{\mathscr{R},\bullet}$ the set of codes C for which there is an instance $\mathbb{x} = (1^n, C_\circ, C_\bullet, g)$ in the relation $\mathscr{R}$ with $C = C_\bullet$.*

5.2 A Canonical PCP for Linear Algebraic CSPs

We show how to construct a "canonical" PCP system for instances in $\mathscr{R}_{\mathsf{LA}}$ (the relation of linear algebraic CSPs). At a high level, a canonical PCP proof for a $\mathscr{R}_{\mathsf{LA}}$-instance $\mathbb{x}$ consists of a witness $\mathbb{w} = (\alpha_\circ, \alpha_\bullet)$ concatenated with two PCPP proofs $\pi_\circ, \pi_\bullet$, showing that $\alpha_\circ, \alpha_\bullet$ are close to $C_\circ, C_\bullet$ respectively. The canonical PCP verifier first checks the two PCPP proofs and then checks that $g(\alpha_\circ)[j] = \alpha_\bullet[j]$ for a uniformly random $j \in [\ell_\bullet]$.

Definition 4. *Given (i) a relation $\mathscr{R} \subseteq \mathscr{R}_{\mathsf{LA}}$, (ii) a PCPP system $\mathsf{PCPP}_\circ = (P_\circ, V_\circ)$ for $\mathrm{Rel}(\mathscr{C}_{\mathscr{R},\circ})$, and (iii) a PCPP system $\mathsf{PCPP}_\bullet = (P_\bullet, V_\bullet)$ for $\mathrm{Rel}(\mathscr{C}_{\mathscr{R},\bullet})$, the* canonical PCP system *for the triple $(\mathscr{R}, \mathsf{PCPP}_\circ, \mathsf{PCPP}_\bullet)$ is the PCP system $\mathsf{PCP} = (P, V)$ constructed as follows.*

- ***Prover.*** *Given $(\mathbb{x}, \mathbb{w}) \in \mathscr{R}_{\mathsf{LA}}$, the PCP prover P outputs $\pi := (\mathbb{w}, \pi_\circ, \pi_\bullet)$ where $\pi_\circ := P_\circ(C_\circ, \alpha_\circ)$ and $\pi_\bullet := P_\bullet(C_\bullet, \alpha_\bullet)$. In other words, the PCP prover outputs a PCP proof that is the concatenation of the witness $\mathbb{w} = (\alpha_\circ, \alpha_\bullet)$ and a pair of PCPP proofs, the first proving that $\alpha_\circ \in C_\circ$ and the second proving that $\alpha_\bullet \in C_\bullet$.*
- ***Verifier.*** *Given $\mathbb{x}$ and oracle access to a PCP proof $\pi = (\mathbb{w}, \pi_\circ, \pi_\bullet)$, the PCP verifier V works as follows:*
  - *(proximity) check that $V_\circ^{(\alpha_\circ, \pi_\circ)}(C_\circ)$ and $V_\bullet^{(\alpha_\bullet, \pi_\bullet)}(C_\bullet)$ both accept;*
  - *(consistency) check that $g(\alpha_\circ)[j] = \alpha_\bullet[j]$ for a uniformly random $j \in [\ell_\bullet]$.*

The next lemma says that the above construction is a PCP system when $\mathscr{R}_{\mathsf{LA}}$'s parameters are sufficiently "good".

Lemma 1 ($\mathscr{R}_{\mathsf{LA}} \to$ PCP). *Suppose that $\mathscr{R}$ is a relation that satisfies the following conditions:*

(i) $\mathscr{R} \subseteq \mathscr{R}_{\mathsf{LA}}[f_1, \ell_1, \rho_1, \delta_1, q_1, c_1, \gamma_1, \epsilon_1]$ *with* $\epsilon_1 < \min\{\frac{\delta_1}{2}, \delta_1 - \gamma_1\}$;
(ii) $\mathrm{Rel}(\mathscr{C}_{\mathscr{R},\circ}), \mathrm{Rel}(\mathscr{C}_{\mathscr{R},\bullet}) \in \mathbf{PCPP}[\mathsf{a}_2, \mathsf{l}_2, \mathsf{q}_2, \Delta^{\mathrm{Ham}}_{\mathsf{a}_2}, \mathsf{d}_2, \mathsf{e}_2, \mathsf{tp}_2, \mathsf{tv}_2, \mathrm{na?}]$ *with* $\mathsf{a}_2 = f_1$ *and* $\mathsf{d}_2 \le \epsilon_1$.

Then there is a canonical PCP system for a triple $(\mathscr{R}, \mathsf{PCPP}_\circ, \mathsf{PCPP}_\bullet)$ that yields

$$\mathscr{R} \in \mathbf{PCP} \begin{bmatrix} \mathsf{a} & = f_1 \ (= \mathsf{a}_2) \\ \mathsf{l} & = 2\mathsf{l}_2(\ell_1) + 2\ell_1 \\ \mathsf{q} & = 2\mathsf{q}_2(\ell_1) + q_1 + 1 \\ \mathsf{e} & = \max\{1 - \delta_1 + \gamma_1 + \epsilon_1, \mathsf{e}_2\} \\ \mathsf{tp} & = 2\mathsf{tp}_2(\ell_1) \\ \mathsf{tv} & = 2\mathsf{tv}_2(\ell_1) + c_1 + \log \ell_1 \\ \mathrm{na?} & \end{bmatrix} .$$

Above, na? *denotes the fact that if the PCPP systems are non-adaptive so is the canonical PCP system.*

Proof (Proof of Lemma 1). First, we show that the canonical PCP system satisfies completeness and soundness; afterwards, we discuss the efficiency parameters achieved by it.

Completeness. Consider an instance-witness pair $(\mathtt{x}, \mathtt{w})$ in the relation $\mathscr{R}$. Parse the instance $\mathtt{x}$ as $(1^n, C_\circ, C_\bullet, g)$ and the witness $\mathtt{w}$ as $(\alpha_\circ, \alpha_\bullet)$. Since $(\mathtt{x}, \mathtt{w}) \in \mathscr{R}$, we have that $\alpha_\circ \in C_\circ$, $\alpha_\bullet \in C_\bullet$, and $g(\alpha_\circ) = \alpha_\bullet$. Therefore, the PCP proof $(\mathtt{w}, \pi_\circ, \pi_\bullet)$ generated by the PCP prover is accepted by the PCP verifier with probability 1: the PCPP verifiers $V_\circ^{(\alpha_\circ, \pi_\circ)}(C_\circ)$ and $V_\bullet^{(\alpha_\bullet, \pi_\bullet)}(C_\bullet)$ always accept and $g(\alpha_\circ)[j] = \alpha_\bullet[j]$ for every $j \in [\ell_\bullet]$.

Soundness. Consider an instance $\mathtt{x}$ not in the language $\mathrm{Lan}(\mathscr{R})$ and a PCP proof $\tilde{\pi} = (\tilde{\mathtt{w}}, \tilde{\pi}_\circ, \tilde{\pi}_\bullet)$. Parse the instance $\mathtt{x}$ as $(1^n, C_\circ, C_\bullet, g)$ and the wintess $\tilde{\mathtt{w}}$, inside $\tilde{\pi}$, as $(\tilde{\alpha}_\circ, \tilde{\alpha}_\bullet)$. We use Claim in Sect. 5.1 to prove that V accepts $\tilde{\pi}$ with probability at most $\max\{1 - \delta_1 + \gamma + \epsilon_1, \mathtt{e}_2\}$, by considering the following three cases.

- *Case 1: $\tilde{\alpha}_\circ$ is ϵ_1-far in relative Hamming distance from $C_\circ$.* The canonical PCP verifier's proximity test fails, because $\Delta_{\mathtt{a}}^{\mathrm{Ham}}(\tilde{\alpha}_\circ, C_\circ) \geq \epsilon_1 \geq \mathtt{d}_2$, and so the PCPP verifier $V_\circ^{(\alpha_\circ, \tilde{\pi}_\circ)}(C_\circ)$ accepts with probability at most $\mathtt{e}_2$.
- *Case 2: $\tilde{\alpha}_\bullet$ is ϵ_1-far in relative Hamming distance from $C_\bullet$.* This case is analogous to the previous one.
- *Case 3: there exist $\alpha_\circ \in C_\circ$ and $\alpha_\bullet \in C_\bullet$ with $\Delta_{\mathtt{a}}^{\mathrm{Ham}}(\alpha_\circ, \tilde{\alpha}_\circ) \leq \epsilon_1$ and $\Delta_{\mathtt{a}}^{\mathrm{Ham}}(\alpha_\bullet, \tilde{\alpha}_\bullet) \leq \epsilon_1$.*
  First, since ϵ_1 is less than $\delta_1/2$ (the unique decoding radius of $C_\circ$ and $C_\bullet$), the codewords $\alpha_\circ$ and $\alpha_\bullet$ are unique.
  Next, we claim that $\alpha'_\bullet := g(\alpha_\circ)$ and $g(\tilde{\alpha}_\circ)$ are γ_1-close. Indeed, since g is (γ_1, ϵ_1)-sampling, $\alpha_\circ$ and $\tilde{\alpha}_\circ$ differ in at most $\epsilon_1 \cdot \ell_\circ(n)$ positions, and so at most $\gamma_1 \cdot \ell_\bullet(n)$ positions of $g(\tilde{\alpha}_\circ)$ depend on an index where $\alpha_\circ$ and $\tilde{\alpha}_\circ$ differ.
  Next, we claim that $\Delta_{\mathtt{a}}^{\mathrm{Ham}}(\alpha_\bullet, \alpha'_\bullet) \geq \delta_1$. Indeed, we have that $\alpha_\bullet \neq \alpha'_\bullet$ because otherwise $(\alpha_\circ, \alpha_\bullet)$ would be a satisfying assignment for $\mathtt{x}$ (contradicting the assumption that $\mathtt{x} \notin \mathrm{Lan}(\mathscr{R})$); moreover, we also have that $C_\bullet \cup g(C_\circ)$ has relative distance at least δ_1.
  We now use the triangle inequality, along with the above observations, to obtain that

$$\begin{aligned}\delta_1 \Delta_{\mathtt{a}}^{\mathrm{Ham}}(\alpha_\bullet, \alpha'_\bullet) &\leq \Delta_{\mathtt{a}}^{\mathrm{Ham}}(\alpha_\bullet, \tilde{\alpha}_\bullet) + \Delta_{\mathtt{a}}^{\mathrm{Ham}}(\tilde{\alpha}_\bullet, g(\tilde{\alpha}_\circ)) + \Delta_{\mathtt{a}}^{\mathrm{Ham}}(g(\tilde{\alpha}_\circ), \alpha'_\bullet) \\ &\leq \epsilon_1 + \Delta_{\mathtt{a}}^{\mathrm{Ham}}(\tilde{\alpha}_\bullet, g(\tilde{\alpha}_\circ)) + \gamma_1 \ .\end{aligned}$$

  Thus, $\Delta_{\mathtt{a}}^{\mathrm{Ham}}(\tilde{\alpha}_\bullet, g(\tilde{\alpha}_\circ)) \geq \delta_1 - (\gamma_1 + \epsilon_1)$, and so the canonical PCP verifier's consistency check passes with probability at most $1 - \delta_1 + \gamma_1 + \epsilon_1$.

We conclude that V accepts $\tilde{\pi}$ with probability at most $\max\{1 - \delta_1 + \gamma_1 + \epsilon_1, \mathtt{e}_2\}$.

Other Parameters. The remaining parameters are straightforward to establish. The canonical PCP does not change the alphabet, so $\mathtt{a} = f_1$ (which also equals $\mathtt{a}_2$). The proof length, and the running times of the prover and verifier are the sum of the same measures of the canonical PCP's components: the PCP

proof has $\mathsf{l} = 2\mathsf{l}_2(\ell_1) + 2\ell_1$ symbols, is produced in time $\mathsf{tp} = 2\mathsf{tp}_2(\ell_1)$, and is verified in time $\mathsf{tv} = 2\mathsf{tv}_2(\ell_1)+c_1+O(1)$. The canonical PCP verifier makes q_1+1 queries on top of those made by the PCPP verifiers, so its query complexity is $\mathsf{q} = 2\mathsf{q}_2(\ell_1) + q_1 + 1$. The $q_1 + 1$ additional queries are non-adaptive; so if the PCPP verifiers are non-adaptive, so is the canonical PCP verifier.

6 Zero-Knowledge Duplex PCPs from Randomizable Linear Algebraic CSPs

We introduce *randomizable linear algebraic CSPs*, a subfamily of linear algebraic CSPs. Then we show that, for every randomizable linear algebraic CSP, we can convert the CSP's canonical PCP into a corresponding zero-knowledge duplex PCP, incurring only little overheads.

6.1 Randomizable Linear Algebraic CSPs

The definition below specifies the notion of randomizability for linear algebraic CSPs.

Definition 5 ($\mathscr{R}_{\mathsf{RLA}}$). *The relation $\mathscr{R}_{\mathsf{RLA}}[f, \ell, \rho, \delta, q, c, \gamma, \epsilon, t, r]$ is the subrelation of $\mathscr{R}_{\mathsf{LA}}[f, \ell, \rho, \delta, q, c, \gamma, \epsilon]$ obtained by restricting it to instances that are t-randomizable in time r. An instance $\mathbb{x} = (1^n, C_\circ, C_\bullet, g)$ is $t(n)$-*randomizable in time $r(n)$ *if: (i) there exists a $t(n)$-wise independent subcode $C' \subseteq C_\circ$ such that if $(w_\circ, g(w_\circ))$ satisfies $\mathbb{x}$, then, for every $w'_\circ$ in $C' + w_\circ := \{w' + w_\circ \mid w' \in C'\}$, the witness $(w'_\circ, g(w'_\circ))$ satisfies $\mathbb{x}$; and (ii) one can sample, in time $r(n)$, three uniformly random elements in C', $C_\circ$ and $C_\bullet$ respectively.*

6.2 Construction of Zero-Knowledge Duplex PCPs

We construct a zero-knowledge duplex PCP system for randomizable linear algebraic CSPs. The duplex PCP system does little more than invoking, as a subroutine, the canonical PCP system for the linear algebraic CSP; hence, the efficiency of the duplex PCP and of the canonical PCP system are closely related. The construction demonstrates that "adding zero knowledge to an algebraic PCP" is cheap, provided that one moves from the PCP model to the (more general) duplex PCP model. More precisely, we prove the following theorem.

Theorem 5 ($\mathscr{R}_{\mathsf{RLA}} \to \mathbf{DPCP}_{\mathrm{pzk}}$). *Suppose that $\mathscr{R}$ is a relation that satisfies the following conditions:*

(i) $\mathscr{R} \subseteq \mathscr{R}_{\mathsf{RLA}}[f_1, \ell_1, \rho_1, \delta_1, q_1, c_1, \gamma_1, \epsilon_1, t_1, r_1]$ *with* $\epsilon_1 < \min\{\frac{\delta_1}{2}, \delta_1 - \gamma_1\}$ *and* r_1 *polynomially bounded;*

(ii) $\mathrm{Rel}(\mathscr{C}_{\mathscr{R},\circ}), \mathrm{Rel}(\mathscr{C}_{\mathscr{R},\bullet}) \in \mathbf{PCPP}[\mathsf{a}_2, \mathsf{l}_2, \mathsf{q}_2, \Delta^{\mathrm{Ham}}_{\mathsf{a}_2}, \mathsf{d}_2, \mathsf{e}_2, \mathsf{tp}_2, \mathsf{tv}_2, \mathrm{na?}]$ *with* $\mathsf{a}_2 = f_1$ *and* $\mathsf{d}_2 \leq \epsilon_1/4$.

Then there is a duplex PCP system for $\mathscr{R}$ that yields

$$\mathscr{R} \in \mathbf{DPCP}_{\mathrm{pzk}} \begin{bmatrix} \mathsf{a} & = f_1 \ (= \mathsf{a}_2) \\ \mathsf{l} & = 2\mathsf{l}_2(\ell_1) + 6\ell_1 \\ \mathsf{q} & = 2\mathsf{q}_2(\ell_1) + q_1 + 7 \\ \mathsf{e} & = \max\{1 - \delta_1 + \gamma_1 + \epsilon_1 \, , \, (1 - |f_1|^{-1}) \cdot \max\{\mathsf{e}_2, \epsilon_1/4\} + |f_1|^{-1}\} \\ \mathsf{tp} & = 2\mathsf{tp}_2(\ell_1) + (c_1 + 5)\ell_1 + r_1 \\ \mathsf{tv} & = 2\mathsf{tv}_2(\ell_1) + c_1 + \log \ell_1 \\ \mathsf{k} & = t_1/q_1 \\ \text{na?} & \end{bmatrix}.$$

Above, na? *denotes the fact that if the PCPP systems are non-adaptive so is the duplex PCP system.*

Proof. We prove the claim by constructing a suitable duplex PCP system $\mathsf{DPCP} = (P, V)$ for the relation $\mathscr{R}$. Recall that: the prover P is a pair of algorithms (P_0, P_1), and the verifier V is also a pair of algorithms (V_0, V_1); moreover, an instance $\mathtt{x}$ of $\mathscr{R}$ is of the form $(1^n, C_\circ, C_\bullet, g)$, while a witness $\mathtt{w}$ of $\mathscr{R}$ is of the form $(\alpha_\circ, \alpha_\bullet)$; finally, randomizability implies that there is a $t(n)$-wise independent subcode $C' \subseteq C_\circ$ such that if $(w_\circ, g(w_\circ))$ satisfies $\mathtt{x}$ then so does the witness $(w'_\circ, g(w'_\circ))$, for every $w'_\circ$ in $C' + w_\circ$.

We now describe the construction of the duplex PCP system $\mathsf{DPCP} = (P, V)$:

- $\underline{P_0(\mathtt{x}, \mathtt{w}) \to \pi_0}$
  Sample uniformly random $v_\circ \in C_\circ, v_\bullet \in C_\bullet, u' \in C'$; compute $w_\circ := u' + \alpha_\circ$, $w_\bullet := g(w_\circ)$ and output $\pi_0 := (w_\circ \| v_\circ \| w_\bullet \| v_\bullet)$.
- $\underline{V_0(\mathtt{x}) \to \rho}$
  Sample uniformly random $\rho_\circ, \rho_\bullet \in f_1$, and output $\rho := (\rho_\circ, \rho_\bullet)$.
- $\underline{P_1(\mathtt{x}, \mathtt{w}, \rho) \to \pi_1}$
  Compute $z_\circ := \rho_\circ w_\circ + v_\circ$ and $z_\bullet := \rho_\bullet w_\bullet + v_\bullet$; compute $\pi_\circ := P_\circ(C_\circ, z_\circ)$ and $\pi_\bullet = P_\bullet(C_\bullet, z_\bullet)$; and output $\pi_1 := (z_\circ \| z_\bullet \| \pi_\circ \| \pi_\bullet)$. (Essentially, this step corresponds to running the canonical PCP prover with respect to a uniformly random pair $(z_\circ, z_\bullet)$ in $(C_\circ, C_\bullet)$.)
- $\underline{V_1^{\pi_0, \pi_1}(\mathtt{x}) \to b}$
  Conduct the following tests (and reject if any of them fails):
  - (proximity) check that $V_\circ^{(z_\circ, \pi_\circ)}(C_\circ)$ and $V_\bullet^{(z_\bullet, \pi_\bullet)}(C_\bullet)$ both accept;
  - (consistency) check that $g(w_\circ)[j] = w_\bullet[j]$ for a random $j \in [\ell_\bullet]$;
  - (linearity) check that $z_\circ[i] = \rho_\circ w_\circ[i] + v_\circ[i]$ and $z_\bullet[k] = \rho_\bullet w_\bullet[k] + v_\bullet[k]$ for random $i \in [\ell_\circ(n)]$ and $k \in [\ell_\bullet(n)]$.

  (Essentially the first two steps correspond to running the canonical PCP verifier on modified inputs, while the third step consists of two linearity tests.)

Having described the duplex PCP system, we now show that it satisfies completeness, soundness and zero-knowledge; afterwards, we discuss the efficiency parameters achieved by it.

Completeness. Consider an instance-witness pair $(\mathtt{x}, \mathtt{w})$ in the relation $\mathscr{R}$. Since $(\mathtt{x}, \mathtt{w}) \in \mathscr{R}$, we have that $\alpha_\circ \in C_\circ$, $\alpha_\bullet \in C_\bullet$, and $g(\alpha_\circ) = \alpha_\bullet$. Since $w_\circ \in C' + \alpha_\circ$ and $\mathscr{R}$ is randomizable, we have that $(w_\circ, w_\bullet) := (w_\circ, g(w_\circ))$ satisfies $\mathtt{x}$;

thus V_1's consistency check passes with probability 1. Since the codes $C_\circ$ and $C_\bullet$ are linear and $w_\circ, v_\circ \in C_\circ$, $w_\bullet, v_\bullet \in C_\bullet$, we have that $z_\circ := \rho_\circ w_\circ + v_\circ \in C_\circ$ and $z_\bullet := \rho_\bullet w_\bullet + v_\bullet \in C_\bullet$; thus the PCPP verifiers $V_\circ^{(z_\circ,\pi_\circ)}(C_\circ)$ and $V_\bullet^{(z_\bullet,\pi_\bullet)}(C_\bullet)$ accept with probability 1. Finally, by construction of $z_\circ$ and $z_\bullet$, V_1's linearity tests also accept with probability 1. We conclude that the duplex PCP system described above has perfect completeness.

Soundness. Consider an instance $\mathbb{x}$ not in the language $\mathrm{Lan}(\mathscr{R})$. Fix an arbitrary proof string $\tilde{\pi}_0 = (\tilde{w}_\circ \| \tilde{v}_\circ \| \tilde{w}_\bullet \| \tilde{v}_\bullet)$, and let the proof string $\tilde{\pi}_1 = (\tilde{z}_\circ \| \tilde{z}_\bullet \| \tilde{\pi}_\circ \| \tilde{\pi}_\bullet)$ depend arbitrarily on the verifier message $\rho = (\rho_\circ, \rho_\bullet)$. We use Claim in Sect. 5.1 with respect to the instance $\mathbb{x}$ and witness $(\tilde{w}_\circ, \tilde{w}_\bullet)$ and distinguish between three cases below.

- *Case 1: $\tilde{w}_\circ$ is ϵ_1-far in relative Hamming distance from $C_\circ$.*
  Claim in Sect. 2 implies that $z'_\circ := \rho_\circ \tilde{w}_\circ + \tilde{v}_\circ$ is $\epsilon_1/2$-far from $C_\circ$, with probability $1 - |f_1|^{-1}$ over a random choice of $\rho_\circ$. Let $\theta := \Delta_{\mathsf{a}}^{\mathrm{Ham}}(z'_\circ, \tilde{z}_\circ)$ and $\eta := \Delta_{\mathsf{a}}^{\mathrm{Ham}}(\tilde{z}_\circ, C_\circ)$. By the triangle inequality, $\theta + \eta \geq \Delta_{\mathsf{a}}^{\mathrm{Ham}}(z'_\circ, C_\circ) \geq \epsilon_1/2$; hence, at least one of the inequalities $\theta \geq \epsilon_1/4$ and $\eta \geq \epsilon_1/4$ holds. In the former case, V_1's first linearity test accepts with probability at most $1 - \epsilon_1/4$; in the latter case, the PCPP verifier $V_\circ^{(\tilde{z}_\circ,\tilde{\pi}_\circ)}(C_\circ)$ for V_1's first proximity test accepts with probability at most e_2, as $\Delta_{\mathsf{a}}^{\mathrm{Ham}}(\tilde{z}_\circ, C_\circ) \geq \epsilon_1/4 \geq \mathsf{d}_2$.
- *Case 2: $\tilde{w}_\bullet$ is ϵ_1-far in relative Hamming distance from $C_\bullet$.*
  This case is analogous to the previous one.
- *Case 3: there exist $w_\circ \in C_\circ$ and $w_\bullet \in C_\bullet$ with $\Delta_{\mathsf{a}}^{\mathrm{Ham}}(w_\circ, \tilde{w}_\circ) \leq \epsilon_1$ and $\Delta_{\mathsf{a}}^{\mathrm{Ham}}(w_\bullet, \tilde{w}_\bullet) \leq \epsilon_1$.*
  In this case we follow the very end of the soundness analysis in Lemma 1's proof, replacing $\tilde{\alpha}_\circ, \tilde{\alpha}_\bullet$ there with $\tilde{w}_\circ, \tilde{w}_\bullet$, and conclude that the verifier accepts with probability at most $1 - \delta_1 + \gamma_1 + \epsilon_1$.

Summing up, in the first case the verifier's acceptance probability is at most $(1 - |f_1|^{-1}) \cdot \max\{\mathsf{e}_2, \epsilon_1/4\} + |f_1|^{-1}$; similarly for the second case. In the third case the rejection probability is $1 - \delta_1 + \gamma_1 + \epsilon_1$, that of the canonical PCP consistency verifier. This completes the soundness analysis.

Zero Knowledge. We construct a simulator S that yields perfect zero knowledge with knowledge bound k. Consider an instance-witness pair $(\mathbb{x}, \mathbb{w})$ in the relation $\mathscr{R}$, and a malicious verifier $\tilde{V} = (\tilde{V}_0, \tilde{V}_1)$ making at most k adaptive queries. $S(\tilde{V}, \mathbb{x})$, the output of the simulator S, when given as input $\tilde{V}$ and $\mathbb{x}$, has to be identically distributed to $\mathrm{DPCPView}(\mathsf{k}, \tilde{V}, P, \mathbb{x}, \mathbb{w})$, which is the view of $\tilde{V}_1$ in its execution when given input $\mathbb{x}$ and when allowed to make a total of $\mathsf{k}(n)$ adaptive queries to π_0, π_1, where $\pi_0 := P_0(\mathbb{x}, \mathbb{w})$ and $\pi_1 := P_1(\mathbb{x}, \mathbb{w}, \tilde{V}_0^{\pi_0}(\mathbb{x}))$. In fact, we will prove a stronger statement: the output of the simulator continues to exactly match the view of the verifier, interacting with the honest prover, even if the verifier is allowed unbounded access to π_1, provided that $\tilde{V}$ makes at most k queries to π_0.

We now discuss how S works. At a high level, S treats $\tilde{V}$ as a black box, running it once without rewinding; along the way, S samples suitable answers

for each query (as discussed below); when $\tilde{V}$ halts, S outputs all the answers and $\tilde{V}$'s randomness (which together form the view of the verifier). The simulator S runs in strict polynomial time, without ever aborting. We now describe how S answers each query.

The simulator S maintains a proof string π^S that is initially unspecified at all locations; we write $\pi^S[i] = *$ if the i-th location of this proof string is unspecified. During the simulation, S adaptively specifies locations in π^S as a result of answering $\tilde{V}$'s queries; this specification process is definitive, in the sense that queries to locations that have been previously specified are answered consistently with the previously-specified value. We now discuss how S adaptively specifies locations in π^S. We distinguish between two parts of the simulation: before the point when $\tilde{V}$ sends his message ρ, and only queries to π_0 are possible; and afterwards, when queries to both π_0 and π_1 are possible.

- *Simulating answers to* $\pi_0 = (w_\circ \| v_\circ \| w_\bullet \| v_\bullet)$*, before* $\tilde{V}$ outputs $\tilde{\rho} = (\tilde{\rho}_\circ, \tilde{\rho}_\bullet)$.
  1. For a query $j \in [\ell_\circ]$ to $w_\circ[j]$: if unspecified, answer with a random field element. That is, if $w_\circ^S[j] = *$, then sample a random $\beta \in f_1$ and set $w_\circ^S := \beta$.
  2. For a query $j \in [\ell_\circ]$ to $v_\circ[j]$: if unspecified, answer with a random field element. That is, if $v_\circ^S[j] = *$, then sample a random $\gamma \in f_1$ and set $v_\circ^S[j] = \gamma$. Then check if there are any unspecified locations of $v_\circ^S$ that are determined by the linear constraint "$v_\circ^S \in C_\circ$" and the currently specified locations of $v_\circ^S$; if there are, set these accordingly.
  3. For a query $j \in [\ell_\bullet]$ to $w_\bullet[j]$: if unspecified, (i) compute the set $I_j \subseteq [\ell_\circ]$ of locations on which $g(w_\circ^S)[j]$ depends (see Definition 2); (ii) deduce $w_\circ^S|_{I_j}$ by querying each $i \in I_j$ according to Step 1; and (iii) set $w_\bullet^S[j] := g(w_\circ^S|_{I_j})$.
  4. For a query $j \in [\ell_\bullet]$ to $v_\bullet[j]$: answer in an analogous way to the case of a query $j \in [\ell_\circ]$ to $v_\circ$.
- *Simulating answers to* $\pi_0 = (w_\circ \| v_\circ \| w_\bullet \| v_\bullet)$ *and* $\pi_1 = (z_\circ \| z_\bullet \| \pi_\circ \| \pi_\bullet)$*, after* $\tilde{V}$ *outputs* $\tilde{\rho} = (\tilde{\rho}_\circ, \tilde{\rho}_\bullet)$.
  5. After receiving $\tilde{\rho} = (\tilde{\rho}_\circ, \tilde{\rho}_\bullet)$, immediately do the following:
     (a) sample a random $z_\circ^S \in C_\circ$ under the constraint "$z_\circ^S[i] = \tilde{\rho}_\circ w_\circ^S[i] + v_\circ^S[i]$ for all i s.t. $w_\circ^S[i] \neq * \wedge v_\circ^S[i] \neq *$";
     (b) sample a random $z_\bullet^S \in C_\bullet$ under the analogous constraint;
     (c) compute $\pi_\circ^S := P_\circ(C_\circ, z_\circ^S)$;
     (d) compute $\pi_\bullet^S := P_\bullet(C_\bullet, z_\bullet^S)$.
  6. All queries to $z_\circ, z_\bullet, \pi_\circ, \pi_\bullet$ are answered according to the values specified in Step 5.
  7. For a query $j \in [\ell_\circ]$ to $w_\circ[j]$ or $v_\circ[j]$: if both are unspecified, answer with a random field element; otherwise, the one that is unspecified is determined according to the constraint $z_\circ^S[i] = \tilde{\rho}_\circ w_\circ^S[i] + v_\circ^S[i]$ (except that, if $\tilde{\rho}_\circ = 0$, then answer according to the constraint $z_\circ^S[i] = v_\circ^S[i]$ by setting $w^S[i]$ to be a random field element).
  8. For a query $j \in [\ell_\bullet]$ to $w_\bullet[j]$: answer analogously to Step 3, except that subqueries to $w_\circ[j]$ follow Step 7.
  9. For a query $j \in [\ell_\bullet]$ to $v_\bullet[j]$: compute $w_\bullet^S[j]$ as in Step 8 and set $v_\bullet^S[j] := \tilde{\rho}_\bullet w_\bullet^S[j] - z_\bullet^S[j]$.

We claim that the above simulation achieves perfect zero-knowledge, that is, $S(\tilde{V}, \mathbb{x})$ is identically distributed to $\mathrm{DPCPView}(\mathsf{k}, \tilde{V}, P, \mathbb{x}, \mathbb{w})$. We show that

the distribution of answers provided by the simulation to $\tilde{V}$ is the same as the distribution of answers obtained by $\tilde{V}$ from the oracles provided by the honest prover. First, we discuss the answers to queries asked before $\tilde{V}$ sends $\tilde{\rho} = (\tilde{\rho}_\circ, \tilde{\rho}_\bullet)$, which can only be to the oracle $\pi_0 = (w_\circ \| v_\circ \| w_\bullet \| v_\bullet)$:

(i) In an honest proof, $v_\circ$ and $v_\bullet$ are random in $C_\circ$ and $C_\bullet$, respectively. The simulator answers a query to either of these by selecting a random field element and then propagating to other locations the linear constraints imposed by belonging to the linear code.
(ii) In an honest proof, $w_\circ$ is computed as $w_\circ := u' + \alpha_\circ$, where u' is random in C'. Any t values from a random codeword in C' are distributed identically to t random field elements, because C' is t-wise independent. The queries of $\tilde{V}$ determine at most $\mathsf{k} \cdot q = t$ locations of $w_\circ$. Hence, in an honest proof, $\tilde{V}$ gets uniformly random answers for its queries to $w_\circ$; this matches the simulated view where S answers $\tilde{V}$'s queries to $w_\circ$ with random fields elements.
(iii) In an honest proof, $w_\bullet$ is a deterministic function of $w_\circ$: $w_\bullet := g(w_\circ)$. As described above, the $\leq t$ positions of $w_\circ$ determined by the verifier's questions are uniformly random in the honest proof, as well as in the simulated proof. Therefore the honest and the simulated views of $w_\bullet$ are identically distributed, as deterministic functions of identically distributed random variables.

Next, we discuss the answers to queries asked after $\tilde{V}$ sends $\tilde{\rho} = (\tilde{\rho}_\circ, \tilde{\rho}_\bullet)$; now $\tilde{V}$ can query both $\pi_0 = (w_\circ \| v_\circ \| w_\bullet \| v_\bullet)$ and $\pi_1 = (z_\circ \| z_\bullet \| \pi_\circ \| \pi_\bullet)$.

In an honest proof, answers to verifiers queries after sending $\tilde{\rho}$ are from an uniform distribution of $v_\circ \in C_\circ, v_\bullet \in C_\bullet, u' \in C'$ (and deterministic functions of those and $\alpha_\circ$), that is further conditioned on the answers given before sending $\tilde{\rho}$.

We conclude the discussion of the simulator by examining the time complexity of the simulation. Most steps of the simulation require (a) sampling a random field element and, possibly, (b) solving a linear system with a polynomial number of equations. The only expensive part of the simulation is Step 5, because it requires sampling random codewords in $C_\circ$ and $C_\bullet$, as well as computing PCPP proofs for these two codewords. Provided that r_1 is polynomially bounded, the entire simulation also runs in polynomial time in the instance size n. (The definition of zero knowledge in Sect. 3 prescribes, as typically done, a simulator that runs in expected probabilistic polynomial time; our simulator runs in strict probabilistic polynomial time.)

7 From NTIME to Randomizable Linear Algebraic CSPs

– $\mathscr{R}_{\mathsf{AP}}$ & $\mathscr{R}_{\mathsf{GAP}}$. In Sect. 7.1, we define *algebraic problems*, implicit in several influential works on PCPs and IP [2,4,5,30] and explicitly defined in [22,34,37]. Afterward, we define *group-preserving algebraic problems*, a new "symmetric" variant of algebraic problems that not only are powerful enough to efficiently capture **NTIME** but are also naturally "randomizable", as discussed below.

- $\mathscr{R}_{\mathsf{AP}} \to \mathscr{R}_{\mathsf{LA}}$. In Sect. 7.2 (see Lemma 2), we show that algebraic problems are a sublanguage of linear algebraic CSPs. This observation shows that the techniques of this paper could potentially be applied to many PCP systems (e.g., those in [2,4,5,11,13–16,22,30,37] to name a few) and also provides a "warm up" for the next item.
- $\mathscr{R}_{\mathsf{GAP}} \to \mathscr{R}_{\mathsf{RLA}}$. In Sect. 7.3 (see Lemma 3), we show an efficient reduction from group-preserving algebraic problems to randomizable linear algebraic CSPs. In other words, the property of group preservation allows the corresponding linear algebraic CSPs to be randomizable.
- $\mathbf{NTIME} \to \mathscr{R}_{\mathsf{GAP}}$. In Sect. 7.4 (see Lemma 4), we show an efficient reduction from **NTIME** to group-preserving algebraic problems.
- $\mathbf{NTIME} \to \mathscr{R}_{\mathsf{RLA}}$. In Sect. 7.5 (see Theorem 6), we explain how to combine the above to obtain the efficient reduction from **NTIME** to randomizable linear algebraic CSPs.

7.1 Algebraic Problems and Group Preservation

The definition below of **algebraic problems** is essentially due to [34] (though the term "algebraic problem" is from [22]); variants of it appear in later works such as [10,14–16,22,36,37].

Definition 6 ($\mathscr{R}_{\mathsf{AP}}$). *Given functions $F\colon \mathbb{N} \to \mathscr{F}$, and $h, m, \eta, d, \sigma\colon \mathbb{N} \to \mathbb{N}$, the relation*

$$\mathscr{R}_{\mathsf{AP}}[F, h, m, \eta, d, \sigma]$$

consists of instance-witness pairs $(\mathbb{x}, \mathbb{w})$ satisfying the following.

- *The instance $\mathbb{x}$ is a tuple $(1^n, H, Q, \boldsymbol{N})$ where:*
  - *H is a subset of $F(n)$ with cardinality $h(n)$;*
  - *Q is a polynomial in $F(n)[X_1, \ldots, X_{m(n)}, Y_1, \ldots, Y_{\eta(n)}]$ such that (i) it has degree less than $h(n)$ in each variable X_i, (ii) it has total degree at most $d(n)$ when viewed as a polynomial in the variables $Y_1, \ldots, Y_{\eta(n)}$ with coefficients in $F(n)[X_1, \ldots, X_{m(n)}]$, (iii) it can be evaluated by an arithmetic circuit of size $\sigma(n)$;*
  - *$\boldsymbol{N} = (N_1, \ldots, N_{\eta(n)})$ and each $N_i\colon F(n)^{m(n)} \to F(n)^{m(n)}$ is an invertible affine function.*
- *The witness $\mathbb{w}$ is a polynomial A in $F(n)[X_1, \ldots, X_{m(n)}]$.*
- *The instance $\mathbb{x}$ and witness $\mathbb{w}$ jointly satisfy the following:*

$$\text{for every } \alpha \in H^{m(n)}, (Q \circ A \circ \boldsymbol{N})(\alpha) = 0 \tag{1}$$

where

$$(Q \circ A \circ \boldsymbol{N})(X) := Q(X_1, \ldots, X_{m(n)}, A(N_1(X_1, \ldots, X_{m(n)})), \ldots, A(N_{\eta(n)}(X_1, \ldots, X_{m(n)}))). \tag{2}$$

Next, we define **group-preserving algebraic problems**, a family of algebraic problems in which the set H is a subgroup of $F(n)$ and the neighbor functions act on the product group $H^{m(n)}$. The additional symmetry enables a reduction to randomizable linear algebraic CSPs, which give rise to zero knowledge duplex PCPs. We believe that group-preserving algebraic problems may find applications in the study of PCPs beyond their use in this paper.

Definition 7 ($\mathscr{R}_{\mathsf{GAP}}$). *The relation $\mathscr{R}_{\mathsf{GAP}}[F, h, m, \eta, d, \sigma]$ is the sub-relation of $\mathscr{R}_{\mathsf{AP}}[F, h, m, \eta, d, \sigma]$ obtained via restriction to instances that are group preserving. An instance $\mathbb{x} = (1^n, H, Q, \boldsymbol{N})$ is* group preserving *if: (i) H is an additive or a multiplicative subgroup of $F(n)$; (ii) each $N_i \colon F(n)^{m(n)} \to F(n)^{m(n)}$ in $\boldsymbol{N}$ can be identified with an element χ_i in $H^{m(n)}$ such that $N_i(x) = \chi_i \odot x$, where $\odot$ denotes the group operation of the product group $H^{m(n)}$.*

We also write $\mathscr{R}_{\mathsf{GAP}}[F, h, m, \eta, d, \sigma, +]$ to denote the further restriction to instances that are additively group preserving (i.e., H is an additive subgroup); similarly, we write $\mathscr{R}_{\mathsf{GAP}}[F, h, m, \eta, d, \sigma, \times]$ to denote the restriction to instances that are multiplicatively group preserving.

- The *degree* of $\mathbb{x}$, denoted $|\mathbb{x}|_{\deg}$, is $\deg_{Y_1,\ldots,Y_{\eta(n)}}(Q)$, i.e., the total degree of Q viewed as a polynomial in the variables $Y_1, \ldots, Y_{\eta(n)}$ with coefficients in the ring $\mathbb{F}[X_1, \ldots, X_{m(n)}]$.
- The *circuit size* of $\mathbb{x}$, denoted $|\mathbb{x}|_{\mathrm{circ}}$, is the circuit size of Q.

7.2 Algebraic Problems Naturally Reduce to Linear Algebraic CSPs

Lemma 2 ($\mathscr{R}_{\mathsf{AP}} \to \mathscr{R}_{\mathsf{LA}}$). *For every $F \colon \mathbb{N} \to \mathscr{F}$, $h, m, \eta, d, \sigma \colon \mathbb{N} \to \mathbb{N}$, $\epsilon \colon \mathbb{N} \to (0, 1)$, and $\mathscr{R} \subseteq \mathscr{R}_{\mathsf{AP}}[F, h, m, \eta, d, \sigma]$ there exist a relation $\mathscr{R}'$ and algorithms $\mathsf{inst}, \mathsf{wit}_1, \mathsf{wit}_2$ satisfying the following conditions:*

- Efficient reduction. *For every instance $\mathbb{x}$, letting $\mathbb{x}' := \mathsf{inst}(\mathbb{x})$:*
  - *for every witness $\mathbb{w}$, if $(\mathbb{x}, \mathbb{w}) \in \mathscr{R}$ then $(\mathbb{x}', \mathsf{wit}_1(\mathbb{x}, \mathbb{w})) \in \mathscr{R}'$;*
  - *for every witness $\mathbb{w}'$, if $(\mathbb{x}', \mathbb{w}') \in \mathscr{R}'$ then $(\mathbb{x}, \mathsf{wit}_2(\mathbb{x}, \mathbb{w}')) \in \mathscr{R}$.*

  Moreover, inst runs in time $\mathrm{poly}(|\mathbb{x}|)$, wit_1 in time $\mathrm{poly}(|\mathbb{x}|) \cdot \tilde{O}(|\mathbb{w}| \cdot \eta \cdot \sigma)$, and wit_2 in time $\mathrm{poly}(|\mathbb{x}|) \cdot \tilde{O}(|\mathbb{w}'|)$.
- Linear algebraic CSP. *The relation $\mathscr{R}'$ is a subset of*

$$\mathscr{R}_{\mathsf{LA}} \begin{bmatrix} f = F \\ \ell = |F|^m \\ \rho = (\frac{hd}{|F|})^m \\ \delta = 1 - \frac{hd}{|F|} \\ q = \eta \\ c = \sigma + \eta \\ \gamma = \eta\epsilon \\ \epsilon \end{bmatrix} .$$

- RM codes. *If $\mathbb{x} = (1^n, H, Q, \boldsymbol{N})$ then $\mathsf{inst}(\mathbb{x}) = (1^n, C_\circ, C_\bullet, g)$ with*
  - $C_\circ = \mathsf{RM}\left[F(n), F(n), m(n), \frac{h(n)}{|F(n)|}\right]$;
  - $C_\bullet = \mathsf{VRM}\left[F(n), F(n), m(n), \frac{h(n)d(n)}{|F(n)|}, H\right]$;
  - *g is the function that maps $F(n)[X_1, \ldots, X_{m(n)}]$ to $F(n)^{F(n)^{m(n)}}$ as follows: given A in $F(n)[X_1, \ldots, X_{m(n)}]$ and $\omega \in F(n)^{m(n)}$, the ω-th coordinate of $g(A)$ equals to $(Q \circ A \circ \boldsymbol{N})(\omega)$.*

Proof (Proof of Lemma 2). Let $\mathbb{x} = (1^n, H, Q, \boldsymbol{N})$ be an instance of $\mathscr{R}_{\mathsf{AP}}[F, h, m, \eta, d, \sigma]$, and construct $\mathbb{x}' := \mathsf{inst}(\mathbb{x}) = (1^n, C_\circ, C_\bullet, g)$ as above. We first argue that $\mathbb{x}'$ is an instance of $\mathscr{R}_{\mathsf{LA}}[f, \ell, \rho, \delta, q, c, \gamma, \epsilon]$.

First, $C_\circ$ and $C_\bullet$ are linear error correcting codes with block length at most $\ell := |F|^m$, rate at most $\rho := \max\{(\frac{h}{|F|})^m, (\frac{hd}{|F|})^m\}$, and relative distance at least $\delta := \min\{1 - \frac{h}{|F|}, 1 - \frac{hd}{|F|}\}$ over the same field F. (See Sect. 2.4.)

By construction, the function g is q-local with $q := \eta$ and c-efficient with $c := \sigma + \eta$; moreover, g is (γ, ϵ)-sampling with $\gamma := \eta\epsilon$, as we now explain. (See Definition 1 for definitions of these properties.) For every $\omega \in F^m$, I_ω denotes the set of indices in F^m that $g(\cdot)[\omega]$ depends on; for the g above, I_ω equals $\{N_1(\omega), \ldots, N_\eta(\omega)\}$. For every $\omega' \in F^m$ and $\omega \in F^m$, if $\omega' \in I_\omega$ then $\omega \in \{N_1^{-1}(\omega'), \ldots, N_\eta^{-1}(\omega')\}$. Hence, the number of ω's with $\omega' \in I_\omega$ is at most η, because each N_i is invertible. We deduce that $\Pr[\, I_\omega \cap I \neq \emptyset \,|\, \omega \leftarrow F^m \,] \leq (\eta \cdot |I|)\,/|F|^m \leq \eta\epsilon$.

Finally, $C_\bullet \cup g(C_\circ)$ has relative distance at least δ because it is a subset of $\mathsf{RM}[F, F, m, \frac{hd}{|F|}]$. This claim is immediate for $C_\bullet$; for $g(C_\circ)$, it follows from the fact that $Q \circ A \circ \boldsymbol{N}$ has, in each variable, a degree that is at most a multiplicative factor of d larger than the degree of A.

We conclude the proof by explaining how one obtains the two witness maps $\mathsf{wit}_1, \mathsf{wit}_2$. For wit_1, suppose that $\mathbb{w} = A \in F[X_1, \ldots, X_m]$ is a witness for $\mathbb{x}$; then one can verify that $\mathbb{w}' := (\alpha_\circ, \alpha_\bullet)$, where $\alpha_\circ := A$ and $\alpha_\bullet := Q \circ A \circ \boldsymbol{N}$, is a witness for $\mathbb{x}'$; $\alpha_\bullet$ can be efficiently obtained by first computing the evaluation of A on F^m (via an FFT), then computing the evaluation of $Q \circ A \circ \boldsymbol{N}$ on F^m (via point-to-point computation), and finally interpolating (via an inverse FFT). Conversely, for wit_2, suppose that $\mathbb{w}' = (\alpha_\circ, \alpha_\bullet)$ is a witness for $\mathbb{x}'$; then one can verify that $\mathbb{w} := \alpha_\circ$ is a witness for $\mathbb{x}$.

7.3 From Group-Preserving Algebraic Problems to Randomizable Linear Algebraic CSPs

Lemma 3 ($\mathscr{R}_{\mathsf{GAP}} \to \mathscr{R}_{\mathsf{RLA}}$). *For every $F\colon \mathbb{N} \to \mathscr{F}$, $h, m, \eta, d, \sigma, t\colon \mathbb{N} \to \mathbb{N}$, $\delta, \epsilon\colon \mathbb{N} \to (0,1)$ with $|F| \geq \hat{h}$, where $\hat{h}$ denotes the smallest integral multiple of h that is greater than $\frac{(h+t)d}{1-\delta}$, and for any $\mathscr{R} \subseteq \mathscr{R}_{\mathsf{GAP}}[F, h, m, \eta, d, \sigma]$ there exist a relation $\mathscr{R}'$ and algorithms $\mathsf{inst}, \mathsf{wit}_1, \mathsf{wit}_2$ satisfying the following conditions:*

- Efficient reduction. *For every instance $\mathbb{x}$, letting $\mathbb{x}' := \mathsf{inst}(\mathbb{x})$:*
  - *for every witness $\mathbb{w}$, if $(\mathbb{x}, \mathbb{w}) \in \mathscr{R}$ then $(\mathbb{x}', \mathsf{wit}_1(\mathbb{x}, \mathbb{w})) \in \mathscr{R}'$;*
  - *for every witness $\mathbb{w}'$, if $(\mathbb{x}', \mathbb{w}') \in \mathscr{R}'$ then $(\mathbb{x}, \mathsf{wit}_2(\mathbb{x}, \mathbb{w}')) \in \mathscr{R}$.*

  Moreover, inst runs in time $\mathrm{poly}(|\mathbb{x}|)$, wit_1 in time $\mathrm{poly}(|\mathbb{x}|) \cdot \tilde{O}(|\mathbb{w}| \cdot \eta \cdot \sigma)$, and wit_2 in time $\mathrm{poly}(|\mathbb{x}|) \cdot \tilde{O}(|\mathbb{w}'|)$.

– Randomizable linear algebraic CSP. *The relation $\mathscr{R}'$ is a subset of*

$$\mathscr{R}_{\mathsf{RLA}}\begin{bmatrix} f = F \\ \ell = \hat{h}^m \\ \rho = (\frac{(h+t)d}{\hat{h}})^m \\ \delta = 1 - (\frac{(h+t)d}{\hat{h}}) \\ q = \eta \\ c = \sigma + \eta \\ \gamma = \eta\epsilon \\ \epsilon \\ t \\ r = \tilde{O}(\hat{h}^m) \end{bmatrix}.$$

Proof (Proof of Lemma 3). Let $\mathbb{x} = (1^n, H, Q, \boldsymbol{N})$ be an instance of $\mathscr{R}_{\mathsf{GAP}}[F, h, m, \eta, d, \sigma]$. We construct an instance $\mathbb{x}' := \mathsf{inst}(\mathbb{x}) = (1^n, C_\circ, C_\bullet, g)$ of $\mathscr{R}_{\mathsf{RLA}}[f, \ell, \rho, \delta, q, c, \gamma, \epsilon, t, r]$ as follows.

Let $\hat{H}$ be a subset of F that is a union of cosets of H with $|\hat{H}| = \hat{h}$ and $\hat{H} \cap H = \emptyset$. (This can be done as follows: let S be a subset of the quotient group $F^{\odot}/H$ with cardinality $|S| = \hat{h}/h$ that does not include $1_{\odot}$, where $F^{\odot}$ denotes the additive or multiplicative group of F, depending on whether H is additive or multiplicative, and $1_{\odot}$ is the identity in H; then set $\hat{H} := \{x \odot y \,|\, x \in S, y \in H\}$.) Analogously to the proof of Lemma 2, we define:

– $C_\circ := \mathsf{RM}\left[F(n), \hat{H}, m(n), \frac{h(n)+t(n)}{\hat{h}(n)}\right]$;
– $C_\bullet := \mathsf{VRM}\left[F(n), \hat{H}, m(n), \frac{(h(n)+t(n))d(n)}{\hat{h}(n)}, H\right]$;
– g to be the function that maps $F(n)[X_1, \ldots, X_{m(n)}]$ to $F(n)^{\hat{H}^{m(n)}}$ as follows: given A in $F(n)[X_1, \ldots, X_{m(n)}]$ and $\omega \in \hat{H}^{m(n)}$, the ω-th coordinate of $g(A)$ equals to $(Q \circ A \circ \boldsymbol{N})(\omega)$. Note that g is well-defined, i.e., $g(A)$ is a function from $\hat{H}^{m(n)}$ to $F(n)$; this follows from the group preservation property of $\mathbb{x}$ (see Definition 7): for every $\omega \in \hat{H}^m$ and $i \in [\eta]$, it holds that $N_i(\omega) \subseteq \hat{H}^m$ because $\hat{H}$ is a union of cosets of H and N_i multiplies every coordinate of ω by an element of H.

We first argue that $\mathbb{x}'$ constructed above is an instance of $\mathscr{R}_{\mathsf{RLA}}[f, \ell, \rho, \delta, q, c, \gamma, \epsilon, t, r]$.

First, analogously to the proof of Lemma 2, we note that $C_\circ$ and $C_\bullet$ are linear error correcting codes with block length at most $\ell := \hat{h}^m$, rate at most $\rho := \max\{(\frac{h+t}{\hat{h}})^m, (\frac{(h+t)d}{\hat{h}})^m\}$, and relative distance at least $\delta := \min\{1 - \frac{h+t}{\hat{h}}, 1 - \frac{(h+t)d}{\hat{h}}\}$ over the same field F; also, we deduce that g is q-local with $q := \eta$, c-efficient with $c := \sigma + \eta$, and (γ, ϵ)-sampling with $\gamma := \eta\epsilon$.

Next, recalling Definition 5, $\mathbb{x}'$ is t-randomizable in time $r := \tilde{O}(\hat{h}^m)$ because: (i) $C' := \mathsf{VRM}[F(n), \hat{H}, m, \frac{h+t}{\hat{h}}, H]$ is a subcode of $C_\circ$ and it is t-wise independent due to Claim in Sect. 2.4 (C' satisfies the hypotheses because $H \cap \hat{H} = \emptyset$ and $\hat{h} - h \geq \frac{(h+t)d}{1-\delta} - h \geq t$); and (ii) one can sample random elements from C', $C_\circ$

and $C_\bullet$ in time $\tilde{O}(\hat{h}^m)$ by using the quasilinear FFT algorithms for multipoint evaluation and interpolation (sampling the random polynomial in necessary basis is easy for $C_\circ$; for vanishing Reed–Muller codes we rely on Alon's Combinatorial Nullstelensatz [1] as per Lemma 4.11 of [15]).

We conclude the proof by observing that necessary witness maps $\mathsf{wit}_1, \mathsf{wit}_2$ exist. Just as in Lemma 2, if $\mathbb{w} = A \in F(n)[X_1, \ldots, X_{m(n)}]$ is a witness for $\mathbb{x}$ then $\mathsf{wit}_1(\mathbb{x}, \mathbb{w})$ outputs $\mathbb{w}' := (A, Q \circ A \circ \boldsymbol{N})$, which is a witness for $\mathbb{x}'$; conversely, if $\mathbb{w}' = (\alpha_\circ, \alpha_\bullet)$ is a witness for $\mathbb{x}'$ then $\mathsf{wit}_2(\mathbb{x}, \mathbb{w}')$ outputs $\mathbb{w} := \alpha_\circ$, which is a witness for $\mathbb{x}$.

7.4 An Efficient Reduction from NTIME to Group-Preserving Algebraic Problems

The following lemma gives an efficient reduction from **NTIME** to group-preserving algebraic problems in which instances are over fields of characteristic 2 and preserve additive groups.

Lemma 4 (NTIME $\to \mathscr{R}_{\mathsf{GAP}}$). *For every $h, m, T \colon \mathbb{N} \to \mathbb{N}$ with $h(n)^{m(n)} = \Omega(T(n) \log T(n))$ and $\mathscr{R} \in \mathbf{NTIME}(T)$ there exist a relation $\mathscr{R}'$ and algorithms $\mathsf{inst}, \mathsf{wit}_1, \mathsf{wit}_2$ satisfying the following conditions:*

- Efficient reduction. *For every instance $\mathbb{x}$, letting $\mathbb{x}' := \mathsf{inst}(\mathbb{x})$:*
  - *for every witness $\mathbb{w}$, if $(\mathbb{x}, \mathbb{w}) \in \mathscr{R}$ then $(\mathbb{x}', \mathsf{wit}_1(\mathbb{x}, \mathbb{w})) \in \mathscr{R}'$;*
  - *for every witness $\mathbb{w}'$, if $(\mathbb{x}', \mathbb{w}') \in \mathscr{R}'$ then $(\mathbb{x}, \mathsf{wit}_2(\mathbb{x}, \mathbb{w}')) \in \mathscr{R}$.*

  Moreover, inst runs in time $\mathrm{poly}(n + \log h(n) + m(n))$ and $\mathsf{wit}_1, \mathsf{wit}_2$ run in time $\tilde{O}(T(n))$.
- Group preserving algebraic problem. *The relation $\mathscr{R}'$ is a subset of*

$$\mathscr{R}_{\mathsf{GAP}} \begin{bmatrix} F & = \mathbb{F}_{2^{\log T + O(\log \log T)}} \\ h & \\ m & \\ \eta & = \mathrm{polylog}(T) \\ d & = O(1) \\ \sigma & = \mathrm{poly}(n + \log T) \\ + & \end{bmatrix} .$$

The proof appears in the full version.

7.5 Combining the Two Reductions

By combining Lemmas 3 and 4, we obtain the following theorem, which gives the reduction claimed at the beginning of this section.

Theorem 6 (NTIME $\to \mathscr{R}_{\mathsf{RLA}}$). *For every $T, t \colon \mathbb{N} \to \mathbb{N}$, $\delta, \epsilon \colon \mathbb{N} \to (0, 1)$, and $\mathscr{R} \in \mathbf{NTIME}(T)$ there exist a relation $\mathscr{R}'$ and algorithms $\mathsf{inst}, \mathsf{wit}_1, \mathsf{wit}_2$ satisfying the following conditions:*

– Efficient reduction. *For every instance* $\mathbb{x}$*, letting* $\mathbb{x}' := \mathsf{inst}(\mathbb{x})$*:*
  - *for every witness* $\mathbb{w}$*, if* $(\mathbb{x}, \mathbb{w}) \in \mathscr{R}$ *then* $(\mathbb{x}', \mathsf{wit}_1(\mathbb{x}, \mathbb{w})) \in \mathscr{R}'$*;*
  - *for every witness* $\mathbb{w}'$*, if* $(\mathbb{x}', \mathbb{w}') \in \mathscr{R}'$ *then* $(\mathbb{x}, \mathsf{wit}_2(\mathbb{x}, \mathbb{w}')) \in \mathscr{R}$*.*

  Moreover, inst *runs in time* $\mathrm{poly}(n + \log(\frac{T(n)+t(n)}{1-\delta(n)}))$ *and* $\mathsf{wit}_1, \mathsf{wit}_2$ *run in time* $\mathrm{poly}(n) \cdot \tilde{O}(\frac{T(n)+t(n)}{1-\delta(n)})$.
– Randomizable linear algebraic CSP. *The relation* $\mathscr{R}'$ *is a subset of*

$$\mathscr{R}_{\mathsf{RLA}} \left[\begin{array}{ll} f & = \mathbb{F}_{2^{\log(T+t)+O(\log\log(T+t))}} \\ \ell & = \tilde{O}(\frac{T+t}{1-\delta}) \\ \rho & = 1-\delta \\ \delta & \\ q & = \mathrm{polylog}(T) \\ c & = \mathrm{poly}(n+\log T) \\ \gamma & = \mathrm{polylog}(T) \cdot \epsilon \\ \epsilon & \\ t & \\ r & = \tilde{O}(\frac{T+t}{1-\delta}) \end{array} \right] .$$

– Affine RS codes over characteristic 2. *Both* $\mathscr{C}_{\mathscr{R}',\circ}$ *and* $\mathscr{C}_{\mathscr{R}',\bullet}$ *are subsets of* $\mathcal{RS}^*_\rho \cup \mathcal{VRS}^*_\rho$ *(see Sect. 2.4).*

Proof (Proof of Theorem 6). First, we invoke Lemma 4 with h, m, T such that $m(n) = 1$ and $h(n) = O(T(n) \log T(n))$; this yields a relation $\mathscr{R}^{(1)}$ and algorithms $\mathsf{inst}^{(1)}, \mathsf{wit}_1^{(1)}, \mathsf{wit}_2^{(1)}$ such that: (i) $\mathsf{inst}^{(1)}, \mathsf{wit}_1^{(1)}, \mathsf{wit}_2^{(1)}$ provide a reduction from $\mathscr{R} \in \mathbf{NTIME}(T)$ to $\mathscr{R}^{(1)}$, with $\mathsf{inst}^{(1)}(\mathbb{x})$ running in time $\mathrm{poly}(n + \log h(n) + m(n))$ and $\mathsf{wit}_1^{(1)}(\mathbb{x}, \mathbb{w}), \mathsf{wit}_2^{(1)}(\mathbb{x}, \mathbb{w}^{(1)})$ in time $\tilde{O}(T(n))$; and (ii) $\mathscr{R}^{(1)}$ is a subset of

$$\mathscr{R}_{\mathsf{GAP}} \left[\begin{array}{ll} F & = \mathbb{F}_{2^{\log T + O(\log\log T)}} \\ h & = O(T(n) \log T(n)) \\ m & = 1 \\ \eta & = \mathrm{polylog}(T) \\ d & = O(1) \\ \sigma & = \mathrm{poly}(n + \log T) \\ + & \end{array} \right] .$$

Next, we invoke Lemma 3 on $\mathscr{R}^{(1)}$, using δ, ϵ, t from the theorem statement. Note that the conditions of the theorem are satisfied as $|F| \geq \frac{(h+t)d}{1-\delta} + h \geq \hat{h}$. Therefore this yields a relation $\mathscr{R}^{(2)}$ and algorithms $\mathsf{inst}^{(2)}, \mathsf{wit}_1^{(2)}, \mathsf{wit}_2^{(2)}$ such that: (i) $\mathsf{inst}^{(2)}, \mathsf{wit}_1^{(2)}, \mathsf{wit}_2^{(2)}$ provide a reduction from $\mathscr{R}^{(1)}$ to $\mathscr{R}^{(2)}$, with $\mathsf{inst}^{(2)}(\mathbb{x}^{(1)})$ running in time $\mathrm{poly}(|\mathbb{x}^{(1)}|)$, $\mathsf{wit}_1^{(2)}(\mathbb{x}^{(1)}, \mathbb{w}^{(1)})$ in time $\mathrm{poly}(|\mathbb{x}^{(1)}|) \cdot \tilde{O}(|\mathbb{w}^{(1)}| \cdot \eta \cdot \sigma)$ and $\mathsf{wit}_2^{(2)}(\mathbb{x}^{(1)}, \mathbb{w}^{(2)})$ in time $\mathrm{poly}(|\mathbb{x}^{(1)}|) \cdot \tilde{O}(|\mathbb{w}^{(2)}|)$; and (ii) $\mathscr{R}^{(2)}$ is a subset of

$$\mathscr{R}_{\mathsf{RLA}}\begin{bmatrix} f = F \\ \ell = O(\frac{h+t}{1-\delta}) \\ \rho = 1-\delta \\ \delta \\ q = \eta \\ c = \sigma + \eta \\ \gamma = \eta\epsilon \\ \epsilon \\ t \\ r = \tilde{O}(\frac{h+t}{1-\delta}) \end{bmatrix}.$$

One can check that $\mathscr{R}^{(2)}$ achieves the parameters specified in the theorem statement.

The desired reduction from $\mathscr{R}$ to $\mathscr{R}^{(2)}$ is given by the algorithms $\mathsf{inst}(\mathbb{x}) := \mathsf{inst}^{(2)}(\mathsf{inst}^{(1)}(\mathbb{x}))$, $\mathsf{wit}_1(\mathbb{x}, \mathbb{w}) := \mathsf{wit}_1^{(2)}(\mathsf{inst}^{(1)}(\mathbb{x}), \mathsf{wit}_1^{(1)}(\mathbb{x}, \mathbb{w}))$, and $\mathsf{wit}_2(\mathbb{x}, \mathbb{w}') := \mathsf{wit}_2^{(1)}(\mathbb{x}, \mathsf{wit}_2^{(2)}(\mathsf{inst}^{(1)}(\mathbb{x}), \mathbb{w}'))$. One can verify that inst runs in time $\mathrm{poly}(n + \log(\frac{T(n)+t(n)}{1-\delta(n)}))$ and $\mathsf{wit}_1, \mathsf{wit}_2$ run in time $\mathrm{poly}(n) \cdot \tilde{O}(\frac{T(n)+t(n)}{1-\delta(n)})$.

8 Proof of Theorem 4

Proof (Proof of Theorem 4). We explain how to combine Theorem 6 and Lemma 5 (and Theorem 3) so to obtain Theorem 4.

Let $\mathscr{R}$ be a relation in $\mathbf{NTIME}(T)$; we need to construct a duplex PCP system for $\mathscr{R}$ with the claimed parameters. For now we focus on achieving soundness of $\frac{1}{2}$, and discuss the general case at the end of the proof.

We first reduce **NTIME** to randomizable linear algebraic CSPs: invoke Theorem 6 on $\mathscr{R}$ to obtain a relation $\mathscr{R}'$ and algorithms $\mathsf{inst}, \mathsf{wit}_1, \mathsf{wit}_2$ such that: (i) $\mathsf{inst}, \mathsf{wit}_1, \mathsf{wit}_2$ provide a reduction from $\mathscr{R}$ to $\mathscr{R}'$, with inst running in time $\mathrm{poly}(n + \log(T(n) + t_1(n)))$ and $\mathsf{wit}_1, \mathsf{wit}_2$ in time $\tilde{O}(T(n) + t_1(n))$; and (ii) $\mathscr{R}'$ is a subset of

$$\mathscr{R}_{\mathsf{RLA}}\begin{bmatrix} f_1 = \mathbb{F}_{2^{\log(T+t_1)+O(\log\log(T+t_1))}} \\ \ell_1 = \tilde{O}(T + t_1) \\ \rho_1 = 1 - \delta_1 \\ \delta_1 \\ q_1 = \mathrm{polylog}(T) \\ c_1 = \mathrm{poly}(n + \log T) \\ \gamma_1 = \mathrm{polylog}(T) \cdot \epsilon_1 \\ \epsilon_1 \\ t_1 \\ r_1 = \tilde{O}(\frac{T+t_1}{1-\delta_1}) \end{bmatrix}.$$

Above, as parameters of Theorem 6, we chose ϵ_1, δ_1 and t_1 as follows: ϵ_1 such that $\gamma_1 = \mathrm{polylog}(T) \cdot \epsilon_1 \leq \frac{2}{9}$, then $\delta_1 := 1 - \epsilon_1/4$, and $t_1 := \mathsf{k} \cdot q_1 = \mathsf{k} \cdot \mathrm{polylog}(T)$.

Next we obtain PCPP systems for the relations corresponding to codes appearing in instances of $\mathscr{R}'$. Theorem 6 guarantees that both $\mathscr{C}_{\mathscr{R}',\circ}$ and $\mathscr{C}_{\mathscr{R}',\bullet}$

are subsets of $\mathcal{RS}^*_\rho \cup \mathcal{VRS}^*_\rho$. We now invoke Theorem 3, choosing $\lambda = 2$ and s such that fields f_1 for $\mathscr{R}'$ and a_2 for the PCPPs match. That is, we chose $s = \tilde{O}(\log\log(T + t_1))$ and obtain:

$$\mathrm{Rel}(\mathscr{C}_{\mathscr{R}',\circ}), \mathrm{Rel}(\mathscr{C}_{\mathscr{R}',\bullet}) \in \mathbf{PCPP} \begin{bmatrix} \mathsf{a}_2 & = \mathbb{F}_{2^{s+\log \ell_1}} \\ \mathsf{l}_2 & = \tilde{O}(\ell_1) \\ \mathsf{q}_2 & = \mathrm{polylog}(\ell_1) \\ \Delta_2 & = \Delta^{\mathrm{Ham}}_{\mathsf{a}} \\ \mathsf{d}_2 & = \rho_1/2 \\ \mathsf{e}_2 & = 1/4 \\ \mathsf{tp}_2 & = \mathrm{poly}(s) \cdot \tilde{O}(\ell_1) \\ \mathsf{tv}_2 & = \mathrm{poly}(s + \log \ell_1) \\ \mathrm{na} & \end{bmatrix}.$$

Finally we invoke Theorem 5 for $\mathscr{R}'$ to obtain a duplex PCP system for $\mathscr{R}'$, supplying the PCPPs we just obtained from Theorem 3. Note that our choices satisfy the hypothesis of Theorem 5 is satisfied, as the two fields match, r_1 is polynomially bounded, and as we chose $\gamma_1, \epsilon_1 \leq \frac{2}{9}$, $\delta_1 \geq \frac{17}{18}$, we also have $\epsilon_1 < \min\{\frac{\delta_1}{2}, \delta_1 - \gamma_1\}$ and $\mathsf{d}_2 \leq \epsilon_1/4$. This establishes our claim that:

$$\mathscr{R} \in \mathbf{DPCP}_{\mathrm{pzk}} \begin{bmatrix} \mathsf{a} & = & & & \mathbb{F}_{2^{\log(T+t_1)+O(\log\log(T+t_1))}} \\ \mathsf{l} & = & 2\mathsf{l}_2(\ell_1) + 6\ell_1 & = & \tilde{O}(T + t_1) \\ \mathsf{q} & = & 2\mathsf{q}_2(\ell_1) + q_1 + 7 & = & \mathrm{polylog}(T) \\ \mathsf{e} & = & & & \frac{1}{2} \\ \mathsf{tp} & = & \mathsf{inst} + \mathsf{wit}_1 + (2\mathsf{tp}_2(\ell_1) + (c_1 + 5)\ell_1 + r_1) & = & \mathrm{poly}(n) \cdot \tilde{O}(T + \mathsf{k}) \\ \mathsf{tv} & = & \mathsf{inst} + (2\mathsf{tv}_2(\ell_1) + c_1 + \log \ell_1) & = & \mathrm{poly}(n + \log(T + \mathsf{k})) \\ \mathsf{k} & & & & \\ \mathrm{na} & & & & \end{bmatrix}.$$

The precise expression for soundness error is $\mathsf{e} := \max\{1-\delta_1+\gamma_1+\epsilon_1\,,\, (1-|f_1|^{-1}) \cdot \max\{\mathsf{e}_2, \epsilon_1/4\}+|f_1|^{-1}\}$, but it is upper bounded by $\frac{1}{2}$, as for us $1-\delta_1+\gamma_1+\epsilon_1 \leq \frac{1}{2}$, $\max\{\mathsf{e}_2, \epsilon_1/4\} = \frac{1}{4}$ and $|f_1| \geq 4$.

Acknowledgments. We thank Yuval Ishai and Mor Weiss for helpful discussions. The research leading to these results has received funding from: the European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreement number 240258; the Israeli Science Foundation (grant 1501/14); and the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370.

References

1. Alon, N.: Combinatorial Nullstellensatz. Comb. Probab. Comput. **8**, 7–29 (1999)
2. Arora, S., Lund, C., Motwani, R., Sudan, M., Szegedy, M.: Proof verification and the hardness of approximation problems. JACM **45**, 501–555 (1998)
3. Arora, S., Safra, S.: Probabilistic checking of proofs: a new characterization of NP. JACM **45**, 70–122 (1998)
4. Babai, L., Fortnow, L., Levin, L.A., Szegedy, M.: Checking computations in polylogarithmic time. In: STOC 1991 (1991)

5. Babai, L., Fortnow, L., Lund, C.: Non-deterministic exponential time has two-prover interactive protocols. Comput. Complex. **1**, 3–40 (1991)
6. Babai, L., Moran, S.: Arthur-Merlin games: a randomized proof system, and a hierarchy of complexity class. J. Comput. Syst. Sci. **36**, 254–276 (1988)
7. Bellare, M., Goldreich, O.: On defining proofs of knowledge. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 390–420. Springer, Heidelberg (1993)
8. Ben-Or, M., Goldreich, O., Goldwasser, S., Håstad, J., Kilian, J., Micali, S., Rogaway, P.: Everything provable is provable in zero-knowledge. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 37–56. Springer, Heidelberg (1990)
9. Ben-Or, M., Goldwasser, S., Kilian, J., Wigderson, A.: Multi-prover interactive proofs: how to remove intractability assumptions. In: STOC 1988 (1988)
10. Ben-Sasson, E., Chiesa, A., Genkin, D., Tromer, E.: Fast reductions from RAMs to delegatable succinct constraint satisfaction problems. In: ITCS 2013 (2013)
11. Ben-Sasson, E., Chiesa, A., Genkin, D., Tromer, E.: On the concrete efficiency of probabilistically-checkable proofs. In: STOC 2013 (2013)
12. Ben-Sasson, E., Goldreich, O., Harsha, P., Sudan, M., Vadhan, S.: Robust PCPs of proximity, shorter PCPs and applications to coding. In: STOC 2004 (2004)
13. Ben-Sasson, E., Goldreich, O., Harsha, P., Sudan, M., Vadhan, S.: Short PCPs verifiable in polylogarithmic time. In: CCC 2005 (2005)
14. Ben-Sasson, E., Goldreich, O., Harsha, P., Sudan, M., Vadhan, S.: Robust PCPs of proximity, shorter PCPs, and applications to coding. SIAM J. Comput. **36**, 889–974 (2006)
15. Ben-Sasson, E., Sudan, M.: Short PCPs with polylog query complexity. SIAM J. Comput. **38**, 551–607 (2008)
16. Ben-Sasson, E., Viola, E.: Short PCPs with projection queries. In: Esparza, J., Fraigniaud, P., Husfeldt, T., Koutsoupias, E. (eds.) ICALP 2014. LNCS, vol. 8572, pp. 163–173. Springer, Heidelberg (2014)
17. Dinur, I.: The PCP theorem by gap amplification. JACM **54**, 12:1–12:44 (2007)
18. Dinur, I., Reingold, O.: Assignment testers: towards a combinatorial proof of the PCP theorem. In: FOCS 2004 (2004)
19. Dwork, C., Feige, U., Kilian, J., Naor, M., Safra, M.: Low communication 2-prover zero-knowledge proofs for NP. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 215–227. Springer, Heidelberg (1993)
20. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems. In: STOC 1985 (1985)
21. Goyal, V., Ishai, Y., Mahmoody, M., Sahai, A.: Interactive locking, zero-knowledge PCPs, and unconditional cryptography. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 173–190. Springer, Heidelberg (2010)
22. Harsha, P., Sudan, M.: Small PCPs with low query complexity. Comput. Complex. **9**, 157–201 (2000)
23. Impagliazzo, R., Yung, M.: Direct minimum knowledge computations. In: Pomerance, C. (ed.) CRYPTO 1987. LNCS, vol. 293, pp. 40–51. Springer, Heidelberg (1988)
24. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge proofs from secure multiparty computation. SIAM J. Comput. **39**, 1121–1152 (2009)
25. Ishai, Y., Mahmoody, M., Sahai, A.: On efficient zero-knowledge PCPs. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 151–168. Springer, Heidelberg (2012)
26. Ishai, Y., Mahmoody, M., Sahai, A., Xiao, D.: On zero-knowledge PCPs: limitations, simplifications, and applications (2015). http://www.cs.virginia.edu/mohammad/files/papers/ZKPCPs-Full.pdf

27. Kalai, Y.T., Raz, R.: Interactive PCP. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 536–547. Springer, Heidelberg (2008)
28. Kilian, J., Petrank, E., Tardos, G.: Probabilistically checkable proofs with zero knowledge. In: STOC 1997 (1997)
29. Lapidot, D., Shamir, A.: A one-round, two-prover, zero-knowledge protocol for NP. Combinatorica **15**, 204–214 (1995)
30. Lund, C., Fortnow, L., Karloff, H., Noam, N.: Algebraic methods for interactive proof systems. JACM **39**, 859–868 (1992)
31. Mahmoody, M., Xiao, D.: Languages with efficient zero-knowledge PCPs are in SZK. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 297–314. Springer, Heidelberg (2013)
32. Mie, T.: Polylogarithmic two-round argument systems. J. Math. Cryptol. **2**, 343–363 (2008)
33. Ostrovsky, R., Wigderson, A.: One-way functions are essential for non-trivial zero-knowledge. In: ISTCS 1993 (1993)
34. Polishchuk, A., Spielman, D.A.: Nearly-linear size holographic proofs. In: STOC 1994 (1994)
35. Shamir, A.: IP = PSPACE. JACM **39**, 869–877 (1992)
36. Spielman, D.: Computationally efficient error-correcting codes and holographic proofs. Ph.D. thesis, Massachusetts Institute of Technology (1995)
37. Szegedy, M.: Many-valued logics and holographic proofs. In: Wiedermann, J., Van Emde Boas, P., Nielsen, M. (eds.) ICALP 1999. LNCS, vol. 1644, pp. 676–686. Springer, Heidelberg (1999)

http://www.springer.com/978-3-662-49098-3

Theory of Cryptography
13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II
Kushilevitz, E.; Malkin, T. (Eds.)
2016, XIII, 596 p. 63 illus., Softcover
ISBN: 978-3-662-49098-3