No-signaling Linear PCPs

Susumu Kiyoshima

NTT Secure Platform Laboratories
susumu@kiyoshima.info

January 18, 2019

Abstract

In this paper, we give a no-signaling linear probabilistically checkable proof (PCP) system for polynomial-time deterministic computation, i.e., a PCP system for $\mathcal{P}$ such that (1) the honest PCP oracle is a linear function and (2) the soundness holds against any (computational) no-signaling cheating prover, who is allowed to answer each query according to a distribution that depends on the entire query set in a certain way. To the best of our knowledge, our construction is the first PCP system that satisfies these two properties simultaneously.

As an application of our PCP system, we obtain a 2-message delegating computation scheme by using a known transformation. Compared with the existing 2-message delegating computation schemes that are based on standard cryptographic assumptions, our scheme requires preprocessing but has a simpler structure and makes use of different (possibly cheaper) standard cryptographic primitives, namely additive/multiplicative homomorphic encryption schemes.

This article is a full version of an earlier article: No-signaling Linear PCPs, in Proceedings of TCC 2018, ©IACR 2018, https://doi.org/10.1007/978-3-030-03807-6_3.

Contents

1 Introduction

Linear PCP. *Probabilistically checkable proofs*, or PCPs, are proof systems with which one can probabilistically verify the correctness of statements with bounded soundness error by reading only a few bits/symbols of proof strings. A central result about PCPs are the *PCP theorem* [AS98, ALM+98], which states that every $\mathcal{NP}$ statement has a PCP system such that the proof string is polynomially long and the verification requires only a constant number of bits of the proof string (the soundness error is a small constant and can be reduced by repetition).

An important application of PCPs to Cryptography is *succinct argument systems*, i.e., argument systems that have very small communication complexity and fast verification time. A famous example of such argument systems is that of Kilian [Kil92], which proves $\mathcal{NP}$ statements by using PCPs as follows.

1. The prover first generates a polynomially long PCP proof for the statement (this is possible thanks to the PCP theorem) and succinctly commits to it by using Merkle's tree-hashing technique.

2. The verifier queries a few bits of the PCP proof just like the PCP verifier.

3. The prover reveals the queried bits by appropriately opening the commitment using the local opening property of Merkle's tree-hashing.

This argument system of Kilian has communication complexity and verification time that depend on the classical $\mathcal{NP}$ verification time only logarithmically; that is, a proof for the membership of an instance x in an $\mathcal{NP}$ language L has communication complexity and verification time $\mathsf{poly}(\lambda + |x| + \log t)$, where λ is the security parameter, t is the time to evaluate the $\mathcal{NP}$ relation of L on x, and poly is a polynomial that is independent of L. Kilian's technique was later extended to obtain succinct non-interactive argument systems (SNARGs) for $\mathcal{NP}$ in the random oracle model [Mic00] as well as in the standard model with non-falsifiable assumptions (such as the existence of extractable hash functions), e.g., [BCC+17, DFH12].[1]

Recently, a specific type of PCPs called *linear PCPs* has boosted the studies of succinct argument systems. Linear PCPs are PCPs such that the honest proofs are linear functions (i.e., the honest proof strings are the truth tables of linear functions).[2] The proof strings of linear PCPs are usually exponentially long, but each bit or symbol of them can be computed efficiently by evaluating the underlying linear functions. A nice property of linear PCPs is that they often have much simpler structures than the existing polynomially long non-linear PCPs; as a result, the use of linear PCPs often lead to simpler constructions of succinct argument systems. The use of linear PCPs in the context of succinct argument systems was initiated by Ishai, Kushilevitz, and Ostrovsky [IKO07], who used them for constructing an argument system for $\mathcal{NP}$ with a laconic prover (i.e., a prover that sends to the verifier only short messages). Subsequently, several works obtained practical implementations of the argument system of Ishai et al. [SBW11, SMBW12, SVP+12, SBV+13, VSBW13], whereas others extended the technique of Ishai et al. for the use for SNARGs and obtained practical implementations of *preprocessing SNARGs* (i.e., SNARGs that require expensive (but reusable) preprocessing setups) [BCI+13, BCG+13, BCTV14].

[1] Actually, SNARGs in the standard model require the existence of common reference strings, and some constructions of them further require that the verifier has some private information about the common reference strings.

[2] In general, their soundness is required to hold against any (possibly non-linear) functions; linear PCPs with this notion of soundness is sometimes called "strong linear PCPs" [BCI+13].

No-signaling PCP. Very recently, Kalai, Raz, and Rothblum [KRR13, KRR14] found that PCPs with a stronger soundness guarantee, called *soundness against no-signaling provers*, are useful for constructing 2-message succinct argument systems under standard assumptions. Concretely, Kalai et al. [KRR13, KRR14] found that (1) no-signaling PCPs (i.e., PCPs that are sound against no-signaling provers) can be constructed for deterministic computation, and (2) their no-signaling PCPs can be used to obtain 2-message succinct argument systems for deterministic computation under the assumptions of the existence of quasi-polynomially secure fully homomorphic encryption schemes or (2-message, polylogarithmic-communication, single-server) private information retrieval schemes. The succinct argument systems of Kalai et al. differ from prior ones in that they can handle only deterministic computation but requires just two messages and is proven secure under standard assumptions. (In contrast, the argument system of Kilian and prior SNARG systems can handle non-deterministic computation but the former requires four messages and the latter are proven secure only in ideal models such as the random oracle model or under non-falsifiable knowledge-type assumptions.)

As observed by Kalai et al. [KRR13, KRR14], 2-message succinct argument systems have a direct application in *delegating computation* [GKR15] (or *verifiable computation* [GGP10]). Specifically, consider a setting where there exist a computationally weak client and a computationally powerful server, and the client wants to delegate a heavy computation to the server. Given a 2-message succinct argument system, the client can delegate the computation to the server in such a way that it can verify the correctness of the server's computation very efficiently (i.e., much faster than doing the computation from scratch).

After the results of Kalai et al. [KRR13, KRR14], no-signaling PCPs and their applications to delegation schemes have been extensively studied. Kalai and Paneth [KP16] extended the results of Kalai et al. [KRR14] and obtained a delegation scheme for deterministic RAM computation, and Brakerski, Holmgren, and Kalai [BHK17] further extended it so that the scheme is adaptively sound (i.e., sound even when the statement is chosen after the verifier's message) and in addition can be based on polynomially hard standard cryptographic assumptions. Paneth and Rothblum [PR17] gave an adaptively sound delegation scheme for deterministic RAM computation with public verifiability (i.e., with a property that not only the verifier but also anyone can verify proofs) albeit with the use of a new cryptographic assumption. Badrinarayanan, Kalai, Khurana, Sahai, and Wichs [BKK+18] gave an adaptively sound delegation scheme for low-space non-deterministic computation (i.e., non-deterministic computation such that the space complexity is much smaller than the time complexity) under sub-exponentially hard cryptographic assumptions.

The no-signaling PCPs that are used by Kalai et al. [KRR13, KRR14] and the subsequent works are not linear. As a result, compared with the delegation schemes that are obtained from the linear-PCP–based preprocessing SNARGs [BCI+13, BCG+13, BCTV14], their delegation schemes have complex structures.

1.1 Our Results

In this paper, we study the problem of constructing *no-signaling linear PCPs*, i.e., linear PCPs that are sound against no-signaling provers. Our main motivation is to obtain a PCP that inherits good properties from both of linear PCPs and no-signaling PCPs. Thus, our goal is to obtain a no-signaling linear PCP that can be used to obtain a 2-message delegation scheme that (1) is secure under standard cryptographic assumptions (like those that are based on no-signaling PCPs) and (2) has a simple structure (like those that are based on linear PCPs).

Main Result: No-signaling linear PCP for $\mathcal{P}$. The main result of this paper is an unconditional construction of no-signaling linear PCPs for polynomial-time deterministic computation. Our construction is designed for proving correctness of arithmetic circuit computation, so it handles statements of the form $(C, \boldsymbol{x}, \boldsymbol{y})$, where C is a polynomial-size arithmetic circuit and the statement to be proven is "$C(\boldsymbol{x}) = \boldsymbol{y}$ holds."

Theorem (informal)**.** *There exists a no-signaling linear PCP for the correctness of polynomial-size arithmetic circuit computation. For a statement $(C, \boldsymbol{x}, \boldsymbol{y})$ and the security parameter 1^λ, the proof generation algorithm runs in time $\mathsf{poly}(|C|)$, the verifier query algorithm runs in time $\mathsf{poly}(\lambda + |C|)$, and the verifier decision algorithm runs in time $\mathsf{poly}(\lambda + |\boldsymbol{x}| + |\boldsymbol{y}|)$.*

A formal statement of this theorem is given as Theorem 1 in Section 4. To the best of our knowledge, our construction is the first linear PCP that is sound against no-signaling provers. (See Section 1.3 for concurrent independent works.)

Our no-signaling linear PCP has a simple structure just like the existing linear PCPs. Indeed, the proof string of our PCP is identical with that of the well-known linear PCP of Arora, Lund, Motwani, Sudan, and Szegedy [ALM+98]. Regarding the verifier, we added slight modifications to that of Arora et al. to simplify the analysis; however, we do not think that these modifications are fundamental.

The analysis of our no-signaling linear PCP is a combination of the analysis of the linear PCP of Arora et al. [ALM+98] and that of the no-signaling PCP of Kalai et al. [KRR14]. Specifically, our analysis on no-signaling soundness follows the same high-level strategy as that of Kalai et al. while borrowing techniques from Arora et al. for implementing the details of the strategy. Additionally, our analysis is simplified from the analysis of Kalai et al. in the sense that, while the analysis of Kalai et al. requires the circuit C in the statement to have a specific redundant form called "augmented layered circuit," our analysis only requires C to have a much less redundant form called "layered circuit." (This simplification relies on a specific structure of our PCP.) A more detailed overview of our analysis is given in Section 3.

Application: Delegation scheme for $\mathcal{P}$ in the preprocessing model. As an application of our no-signaling linear PCP, we construct a 2-message delegation scheme for polynomial-time deterministic computation under standard cryptographic assumptions. Just like previous linear-PCP–based delegation schemes and succinct arguments (such as that of Bitansky et al. [BCI+13]), our delegation scheme works in the *preprocessing model*, so our scheme requires expensive offline setups that can be reused for proving multiple statements. When the statement is $(C, \boldsymbol{x}, \boldsymbol{y})$, the running time of the client is $\mathsf{poly}(\lambda + |C|)$ in the offline phase and is $\mathsf{poly}(\lambda + |\boldsymbol{x}| + |\boldsymbol{y}|)$ in the online phase. Our delegation scheme is adaptively secure in the sense that the input $\boldsymbol{x}$ can be chosen in the online phase, and is "designated-verifier type" in the sense that the verification requires a secret key. We obtain our delegation scheme by applying the transformation of Kalai et al. [KRR13, KRR14] on our no-signaling liner PCP. (The transformation of Kalai et al., which is closely related to those of Biehl, Meyer, and Wetzel [BMW98] and Aiello, Bhatt, Ostrovsky, and Rajagopalan [ABOR00], transforms a no-signaling PCP to a 2-message delegation scheme.)

Compared with the existing 2-message delegation schemes based on non-linear no-signaling PCPs (such as that of Kalai et al. [KRR14]), our scheme requires preprocessing, but has a simple structure and uses different (possibly cheaper) tools thanks to the use of no-signaling linear PCPs. Concretely, we can avoid the use of fully homomorphic encryption schemes or 2-message private information retrieval schemes, and can instead use additive homomorphic encryption schemes over prime-order fields (such as that of Goldwasser and Micali [GM84]) or multiplicative homomorphic encryption schemes over prime-order bilinear groups (such as the DLIN-based linear encryption scheme of Boneh, Boyen, and Shacham [BBS04]).

1.2 Prior Works

Delegation scheme. Delegation schemes (and verifiable computation schemes) have been extensively studied in literature. Other than those that we mentioned above, existing results that are related to ours are the following. (We focus our attention on non-interactive or 2-message delegation schemes for all deterministic or non-deterministic polynomial-time computation.)

Delegation schemes for non-deterministic computation. The existing constructions of (preprocessing) SNARGs, such as [Gro10, Lip12, DFH12, BC12, GGPR13, BCI+13, BCCT13, Lip13, DFGK14, Gro16, BISW17, BCC+17], can be directly used to obtain delegation schemes for $\mathcal{NP}$, and some of them can be used even to obtain publicly verifiable ones. Additionally, it was shown recently that an interactive variant of PCPs, called *interactive oracle proofs*, can also be used to obtain delegation schemes for $\mathcal{NP}$ [BCS16]. The security of these delegation schemes holds under non-standard assumptions (e.g., knowledge assumptions) or in ideal models (e.g., the generic group model and the random oracle model). Compared with these schemes, our scheme works only for $\mathcal{P}$ and requires preprocessing, but can be proven secure in the standard model under a standard assumption (namely the existence of homomorphic encryption schemes).

Delegation schemes for deterministic computation. Other than the abovementioned recent works that obtain delegation schemes for $\mathcal{P}$ by using no-signaling PCPs (i.e., Kalai et al. [KRR13, KRR14] and the subsequent works), there are plenty of works that obtain delegation schemes for $\mathcal{P}$ without using PCPs. Specifically, some works obtain schemes with preprocessing by using fully homomorphic encryption or attribute-based encryption schemes [GGP10, CKV10, PRV12], and others obtain schemes without preprocessing by using multi-linear maps or indistinguishability obfuscators (e.g., [BGL+15, CHJV15, CH16, CCHR16, KLW15, CCC+16, ACC+16]). Compared with these schemes, our scheme requires preprocessing but only uses relatively simple building blocks (namely a linear PCP and a homomorphic encryption scheme).

1.3 Concurrent Works

In independent concurrent works, Holmgren and Rothblum [HR18] and Chiesa, Manohar, and Shinkar [CMS19] also observe that one can obtain no-signaling PCPs for $\mathcal{P}$ without relying on the "augmented circuit" technique of Kalai et al. [KRR14]. The technique by Holmgren and Rothblum works when the underlying PCP is that of Babai et al. [BFLS91] (as in the work of Kalai et al. [KRR14]) and the one by Chiesa et al. works when the underlying PCP is that of Arora et al. [ALM+98] (as in this paper).

Actually, the work of Chiesa et al. [CMS19] has many other similarities with our work, and in particular their work shows that the linear PCP of Arora et al. [ALM+98] is sound against no-signaling cheating provers. Differences between their work and our work include:

- Chiesa et al. achieve constant soundness error with constant query complexity while we focus on achieving negligible soundness error and did not try to optimize the query complexity (specifically, we slightly modified the verifier algorithm, and our analysis currently requires polynomial query complexity[3]).

[3] It is likely that the query complexity of our PCP can be easily reduced to polylogarithmic, but we have not verified it

- The analysis by Chiesa et al. uses the equivalence between no-signaling functions and *quasi-distributions*[4] over functions while ours does not use this equivalence. (The equivalence between no-signaling functions and quasi-distributions was shown by Chiesa, Manohar, and Shinkar [CMS18] relying on Fourier analytic techniques.)

Remark 1. Chiesa et al. [CMS19] use the term "no-signaling linear PCPs" in a different meaning from us. Specifically, Chiesa et al. use it to refer to PCPs such that the honest proofs are linear functions and the soundness holds against no-signaling cheating provers that are equivalent with quasi-distributions over linear functions, while we use it to refer to PCPs such that the honest proofs are linear functions and the soundness holds against *any* no-signaling cheating provers (which are not necessarily equivalent with quasi-distributions over linear functions).

1.4 Outline

In Section 2, we introduce the notations and definitions that we use in the subsequent sections. In Section 3, we give an overview of the construction and analysis of our no-signaling linear PCP. In Section 4, we formally describe the construction of our no-signaling linear PCP. From Section 5 to Section 9, we analyze the no-signaling soundness of our construction. In Section 10, we describe the application to delegation schemes.

2 Preliminaries

In this section, we introduce the notations and definitions that we use in the subsequent sections.

2.1 Basic Notations

We denote the security parameter by λ. Let $\mathbb{N}$ be the set of all natural numbers. For any $k \in \mathbb{N}$, let $[k] := \{1, \dots, k\}$.

We denote a vector in a bold shape (e.g., $\boldsymbol{v}$). For a vector $\boldsymbol{v} = (v_1, \dots, v_\lambda)$ and a set $S \subseteq [\lambda]$, let $\boldsymbol{v}|_S := \{v_i\}_{i \in S}$. Similarly, for a function $f : D \to R$ and a set $S \subseteq D$, let $f|_S := \{f(i)\}_{i \in S}$. For two vectors $\boldsymbol{u} = (u_1, \dots, u_\lambda)$, $\boldsymbol{v} = (v_1, \dots, v_\lambda)$ of the same length (where each element is a field element), let $\langle \boldsymbol{u}, \boldsymbol{v} \rangle := \sum_{i \in [\lambda]} u_i v_i$ denote their inner product and $\boldsymbol{u} \otimes \boldsymbol{v} := (u_i v_j)_{i,j \in [\lambda]}$ denote their tensor product.[5]

For a set S, we denote by $s \leftarrow S$ a process of obtaining an element $s \in S$ by a uniform sampling from S. Similarly, for any probabilistic algorithm Algo, we denote by $y \leftarrow \mathsf{Algo}$ a process of obtaining an output y by an execution of Algo with uniform randomness. For an event E and a probabilistic process P, we denote by $\Pr[E \mid P]$ the probability of E occuring over the randomness of P.

2.2 Circuits

All circuits in this paper are arithmetic circuits over finite fields of prime orders, and they have addition and multiplication gates with fan-in 2. We assume without loss of generality that they are "layered," i.e., the gates in a circuit can be partitioned into layers such that (1) the first layer consists of the input

formally.

[4]Quasi-distributions are a generalized notion of probability distributions and allow negative probabilities.

[5]In this paper, the tensor product of two vectors are viewed as a vector (with an appropriate ordering of the elements) rather than a matrix.

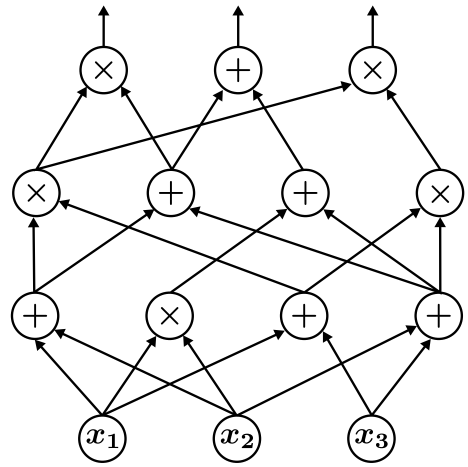

Figure 1: A layered circuit, where the gates in the bottom layer are the input gates and those in the top layer are the output gates.

gates and the last layer consists of the output gates, and (2) the gates in the i-th layer have children in the $(i-1)$-th layer (see Figure 1 for an illustration).

Given a circuit C, we use $\mathbb{F}$ to denote the underlying finite field, N to denote the number of the wires,[6] n to denote the number of the input gates, and m to denote the number of the output gates. We assume that the first n wires of C are those that take the values of the input gates and the last m ones are those that take the value of the output gates. (Formally, $\mathbb{F}, N, n, m$ should be written as, e.g., $\mathbb{F}_C, N_C, n_C, m_C$ since they depend on the circuit C. However, to simplify the notations, we avoid expressing this dependence.) When we consider a circuit family $\{C_\lambda\}_{\lambda \in \mathbb{N}}$, it is implicitly assumed that the size of each C_λ is bounded by $\mathsf{poly}(\lambda)$.

2.3 Probabilistically Checkable Proofs (PCPs)

Roughly speaking, probabilistically checkable proofs (PCPs) are proof systems with which one can probabilistically verify the correctness of statements by reading only a few bits or symbols of the proof strings. A formal definition is given below.

Remark 2 (On the definition that we use). For convenience, we give a definition that is tailored to our purpose. Specifically, our definition differs from the standard one in the following way.

1. We require that the soundness error is negligible in the security parameter.
2. We only consider proofs for the correctness of deterministic arithmetic circuit computation, i.e., membership proofs for the following language.
$$\{(C, \boldsymbol{x}, \boldsymbol{y}) \mid C \text{ is an arithmetic circuit s.t. } C(\boldsymbol{x}) = \boldsymbol{y}\} \ .$$
3. We implicitly require that PCP systems satisfy two auxiliary properties (which almost all existing constructions satisfy), namely **relatively efficient oracle construction** and **non-adaptive verifier** [BG09].

[6] We assume that for any gate with fan-out more than one, all the output wires of that gate share the same index $i \in [N]$.

4. We assume that the verifier's queries depend only on the circuit C and do not depend on the input $\boldsymbol{x}$ and the output $\boldsymbol{y}$. (This assumption will be useful later when we define adaptive soundness against no-signaling cheating provers.) $\diamond$

Definition 1 (PCPs for correctness of arithmetic circuit computation)**.** ***A probabilistically checkable proof (PCP) system for the correctness of arithmetic circuit computation*** *consists of a pair of* PPT *Turing machines $V = (V_0, V_1)$ (called* ***verifier****) and a* PPT *Turing machine P (called* ***prover****) that satisfy the following.*

- ***Syntax.*** *For every arithmetic circuit C, there exist*
  - *finite sets D_C and Σ_C (called* ***proof domain*** *and* ***proof alphabet****) and*
  - *a polynomial κ_V (called* ***query complexity*** *of V)*

  such that for every input $\boldsymbol{x}$ of C, the output $\boldsymbol{y} := C(\boldsymbol{x})$, and every security parameter $\lambda \in \mathbb{N}$,
  - *$P(C, \boldsymbol{x})$ outputs a function $\pi : D_C \to \Sigma_C$ (called* ***proof****),*
  - *$V_0(1^\lambda, C)$ outputs a string $\mathsf{st}_V \in \{0,1\}^*$ (called* ***state****) and a set $Q \subset D_C$ of size $\kappa_V(\lambda)$ (called* ***queries****), and*
  - *$V_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi|_Q)$ outputs a bit $b \in \{0,1\}$.*
- ***Completeness.*** *For every arithmetic circuit C, every input $\boldsymbol{x}$ of C, the output $\boldsymbol{y} := C(\boldsymbol{x})$, and every security parameter $\lambda \in \mathbb{N}$,*

$$\Pr\left[V_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi|_Q) = 1 \;\middle|\; \begin{array}{l} \pi \leftarrow P(C, \boldsymbol{x}) \\ (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C) \end{array} \right] = 1 \quad .$$

- ***Soundness.*** *For any circuit family $\{C_\lambda\}_{\lambda\in\mathbb{N}}$ and any probabilistic Turing machine P^* (called cheating prover), there exists a negligible function* negl *such that for every security parameter $\lambda \in \mathbb{N}$,*

$$\Pr\left[V_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*|_Q) = 1 \land C_\lambda(\boldsymbol{x}) \neq \boldsymbol{y} \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda) \end{array} \right] \leq \mathsf{negl}(\lambda) \quad .$$

$\diamond$

A PCP system is said to be linear if the honest proof is a linear function.

Definition 2 (Linear PCPs)**.** *Let (P, V) be any PCP system and $\{D_C\}_C$ be its proof domains. Then, (P, V) is said to be* ***linear*** *if for every arithmetic circuit C and input $\boldsymbol{x}$ of C,*

$$\Pr\left[\bigwedge_{u,v\in D_C} \pi(u) + \pi(v) = \pi(u+v) \;\middle|\; \pi \leftarrow P(C, \boldsymbol{x}) \right] = 1 \quad .$$

2.4 No-signaling PCPs

No-signaling PCPs [KRR13, KRR14] are PCP systems that guarantee soundness against a stronger class of cheating provers called no-signaling cheating provers. The main difference between no-signaling cheating provers and normal cheating provers in that, while a normal cheating prover is required to output a PCP proof π^* before seeing queries Q, a no-signaling cheating prover is allowed to output π^* after seeing Q. There is however a restriction on the distribution of π^*; roughly speaking, it is required that for any (not too large) sets Q, Q' such that $Q' \subset Q$, the distribution of $\pi^*|_{Q'}$ when the queries are Q should be indistinguishable from the distribution of it when the queries are Q'. The formal definition is given below. (The following definition is the computational variant of the definition, which is given by Brakerski, Holmgren, and Kalai [BHK17].)

Definition 3 (No-signaling cheating prover)**.** *Let (P, V) be any PCP system, $\{D_C\}_C$ and $\{\Sigma_C\}_C$ be the proof domains and proof alphabets of (P, V), $\{C_\lambda\}_{\lambda\in\mathbb{N}}$ be any circuit family, and P^* be any probabilistic Turing machine with the following syntax.*

- *Given the security parameter $\lambda \in \mathbb{N}$, the circuit C_λ, and a set of queries $Q \subset D_{C_\lambda}$ as input, P^* outputs an input $\boldsymbol{x}$ of C_λ, an output $\boldsymbol{y}$ of C_λ, and a partial function $\pi^* : Q \to \Sigma_{C_\lambda}$. (Note that π^* can be viewed as a PCP proof whose domain is restricted to Q.)*

*Then, for any polynomial $\kappa_{\max}$, P^* is said to be a **$\kappa_{\max}$-wise (computational) no-signaling cheating prover** if for any PPT Turing machine $\mathcal{D}$, there exists a negligible function negl such that for every $\lambda \in \mathbb{N}$, every $Q, Q' \subset D_{C_\lambda}$ such that $Q' \subset Q$ and $|Q| \leq \kappa_{\max}(\lambda)$, and every $z \in \{0,1\}^{\mathsf{poly}(\lambda)}$,*

$$\left| \begin{array}{l} \Pr\left[\mathcal{D}(C_\lambda, \boldsymbol{x}, \boldsymbol{y}, \pi^*|_{Q'}, z) = 1 \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q) \right] \\ \qquad - \Pr\left[\mathcal{D}(C_\lambda, \boldsymbol{x}, \boldsymbol{y}, \pi^*, z) = 1 \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q') \right] \end{array} \right| \leq \mathsf{negl}(\lambda) \ .$$

$\diamond$

Now, we define no-signaling PCPs as the PCP systems that satisfy soundness according to the following definition.

Definition 4 (Soundness against no-signaling cheating provers)**.** *Let (P, V) be any PCP system and $\kappa_{\max}$ be any polynomial. Then, (P, V) is said to be **sound against $\kappa_{\max}$-wise (computational) no-signaling cheating provers** if for any circuit family $\{C_\lambda\}_{\lambda\in\mathbb{N}}$ and $\kappa_{\max}$-wise no-signaling cheating prover P^*, there exists a negligible function negl such that for every $\lambda \in \mathbb{N}$,*

$$\Pr\left[V_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*) = 1 \wedge C_\lambda(\boldsymbol{x}) \neq \boldsymbol{y} \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q) \end{array} \right] \leq \mathsf{negl}(\lambda) \ .$$

$\diamond$

3 Technical Overview

In this section, we give an informal overview of our no-signaling linear PCP system. Recall that our focus is PCP systems for the correctness of arithmetic computation, which are PCP systems that take as input a tuple $(C, \boldsymbol{x}, \boldsymbol{y})$ and prove that $C(\boldsymbol{x}) = \boldsymbol{y}$ holds. Given a circuit C, we use $\mathbb{F}$ to denote the underlying finite field, N to denote the number of the wires,[7] n to denote the number of the input gates, and m to denote the number of the output gates. We assume that the first n wires are those that take the values of the input gates and the last m ones are those that take the value of the output gates. In this overview, we additionally assume that the output length is 1 (i.e., $m = 1$).

3.1 Preliminary: Linear PCP of Arora et al. [ALM+98]

The construction and analysis of our PCP system is based on the linear PCP system of Arora et al. [ALM+98] (ALMSS linear PCP in short), so we start by recalling it. We only describe the construction of ALMSS linear PCP in this section; a more detailed overview of ALMSS linear PCP can be found in Appendix A.

3.1.1 Main tool: Walsh–Hadamard code.

The main tool of ALMSS linear PCP system is Walsh–Hadamard code. Recall that Walsh–Hadamard code maps a string $\boldsymbol{v} \in \mathbb{F}^\ell$ to the linear function $\mathsf{WH}_{\boldsymbol{v}} : \boldsymbol{x} \mapsto \langle \boldsymbol{v}, \boldsymbol{x} \rangle$. A useful property of Walsh–Hadamard code is that errors on codewords can be easily "self-corrected." In particular, if a function $f : \mathbb{F}^\ell \to \mathbb{F}$ is *δ-close* to a linear function $\hat{f}$ (i.e., if there exists a linear function $\hat{f}$ such that $\Pr[f(\boldsymbol{r}) = \hat{f}(\boldsymbol{r}) \mid \boldsymbol{r} \leftarrow \mathbb{F}^\ell] \geq \delta$), we can evaluate $\hat{f}$ on any point $\boldsymbol{x} \in \mathbb{F}^\ell$ with error probability $2(1-\delta)$ through the following simple probabilistic procedure.

Algorithm $\mathsf{Self\text{-}Correct}^f(\boldsymbol{x})$:

Choose random $\boldsymbol{r} \in \mathbb{F}^\ell$ and output $f(\boldsymbol{x} + \boldsymbol{r}) - f(\boldsymbol{r})$.

3.1.2 Construction of ALMSS linear PCP.

On input $(C, \boldsymbol{x})$, the prover P computes the PCP proof as follows. First, P computes $y := C(\boldsymbol{x})$ and obtains the following system of quadratic equations over $\mathbb{F}$, which is designed so that it is satisfiable if and only if $C(\boldsymbol{x}) = y$.

- The variables are $\boldsymbol{z} = (z_1, \ldots, z_N)$.
- For each $i \in \{1, \ldots, n\}$, the system has the equation $z_i = x_i$.
- For each $i, j, k \in [N]$, the system has $z_i + z_j - z_k = 0$ if C has an addition gate with input wires i, j and output wire k, and has $z_i z_j - z_k = 0$ if C has a multiplication gate with input wires i, j and output wire k.
- The system has the equation $z_N = y$.

(Intuitively, the variables of the above system of quadratic equations represent the wire values of C, and the equations guarantee that (1) the correct input values $\boldsymbol{x} = (x_1, \ldots, x_n)$ are assigned on the input gates, (2) each gate is correctly computed, and (3) the claimed output value y is assigned on the output gate.) Let us denote the above system of quadratic equations by $\Psi = \{\Psi_i(\boldsymbol{z}) = c_i\}_{i \in [M]}$, where M is the number of the equations. Then, P obtains the satisfying assignment $\boldsymbol{w} = (w_1, \ldots, w_N)$ of Ψ through the wire values of C on $\boldsymbol{x}$, and outputs the two linear functions $\pi_f(\boldsymbol{v}) := \langle \boldsymbol{v}, \boldsymbol{w} \rangle$ and $\pi_g(\boldsymbol{v}') := \langle \boldsymbol{v}', \boldsymbol{w} \otimes \boldsymbol{w} \rangle$ as the PCP proof.[8] (In short, the PCP proof is Walsh–Hadamard encodings of $\boldsymbol{w}$ and $\boldsymbol{w} \otimes \boldsymbol{w}$.)

[7] We assume that for any gate with fan-out more than one, all the output wires of that gate share the same index $i \in [N]$.

[8] Formally, P outputs a single linear function (with which the verifier can evaluate both π_f and π_g) as the PCP proof, but in this overview we simply think that the prover outputs two linear function as the PCP proof.

Next, on input $(C, \boldsymbol{x}, y)$, the verifier V verifies the PCP proof as follows. First, V obtains the system of quadratic equations $\Psi = \{\Psi_i(z) = c_i\}_{i \in [M]}$. Next, V applies the following three tests on the PCP proof λ times in parallel.

1. **(Linearity Test.)** Choose random points $\boldsymbol{r}_1, \boldsymbol{r}_2 \in \mathbb{F}^N$ and $\boldsymbol{r}'_1, \boldsymbol{r}'_2 \in \mathbb{F}^{N^2}$ and check $\pi_f(\boldsymbol{r}_1) + \pi_f(\boldsymbol{r}_2) \stackrel{?}{=} \pi_f(\boldsymbol{r}_1 + \boldsymbol{r}_2)$ and $\pi_g(\boldsymbol{r}'_1) + \pi_g(\boldsymbol{r}'_2) \stackrel{?}{=} \pi_g(\boldsymbol{r}'_1 + \boldsymbol{r}'_2)$.
2. **(Tensor-Product Test.)** Choose two random points $\boldsymbol{r}_1, \boldsymbol{r}_2 \in \mathbb{F}^N$, run $a_{\boldsymbol{r}_1} \leftarrow \mathsf{Self\text{-}Correct}^{\pi_f}(\boldsymbol{r}_1)$, $a_{\boldsymbol{r}_2} \leftarrow \mathsf{Self\text{-}Correct}^{\pi_f}(\boldsymbol{r}_2)$, $a_{\boldsymbol{r}_1 \otimes \boldsymbol{r}_2} \leftarrow \mathsf{Self\text{-}Correct}^{\pi_g}(\boldsymbol{r}_1 \otimes \boldsymbol{r}_2)$, and check $a_{\boldsymbol{r}_1} a_{\boldsymbol{r}_2} \stackrel{?}{=} a_{\boldsymbol{r}_1 \otimes \boldsymbol{r}_2}$.
3. **(SAT Test.)** Choose a random point $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_M) \in \mathbb{F}^M$, compute a quadratic function $\Psi_\sigma(z) := \sum_{i=1}^M \sigma_i \Psi_i(z)$, run $a_{\boldsymbol{\psi}_\sigma} \leftarrow \mathsf{Self\text{-}Correct}^{\pi_f}(\boldsymbol{\psi}_\sigma)$, $a_{\boldsymbol{\psi}'_\sigma} \leftarrow \mathsf{Self\text{-}Correct}^{\pi_g}(\boldsymbol{\psi}'_\sigma)$ for the coefficient vectors $\boldsymbol{\psi}_\sigma, \boldsymbol{\psi}'_\sigma$ such that $\langle \boldsymbol{\psi}_\sigma, z \rangle + \langle \boldsymbol{\psi}'_\sigma, z \otimes z \rangle = \Psi_\sigma(z)$, and check $a_{\boldsymbol{\psi}_\sigma} + a_{\boldsymbol{\psi}'_\sigma} \stackrel{?}{=} c_\sigma$, where $c_\sigma := \sum_{i=1}^M \sigma_i c_i$.

V accepts the proof if it passes the above three tests in all the λ parallel trials. It can be verified by inspection that, as required in Definition 1, the verifier can be decomposed into V_0 and V_1, where V_0 samples the queries to the tests and V_1 verifies the answers from the PCP proof. (Note that V_0 can sample all the queries before knowing $\boldsymbol{x}$ and y since the coefficient vectors $\boldsymbol{\psi}_\sigma, \boldsymbol{\psi}'_\sigma$ in SAT Test can be computed from C alone.)

3.2 Construction of Our No-signaling Linear PCP

The construction of our PCP system, (P, V), is essentially identical with that of ALMSS linear PCP. There is a slight difference in the verifier algorithm (in our PCP system, Self-Correct samples many candidates of the self-corrected values and takes the majority), but we ignore this difference in this overview. It can be verified by inspection that the running time of P is $\mathsf{poly}(|C|)$, the running time of V_0 is $\mathsf{poly}(\lambda + |C|)$, and the running time of V_1 is $\mathsf{poly}(\lambda + |\boldsymbol{x}| + |\boldsymbol{y}|)$.

3.3 Analysis of Our No-signaling Linear PCP

Our goal is to show that our PCP system (P, V) is sound against $\kappa_{\max}$-wise no-signaling cheating provers for sufficiently large polynomial $\kappa_{\max}$. That is, our goal is to show that for every circuit family $\{C_\lambda\}_{\lambda \in \mathbb{N}}$ and every $\kappa_{\max}$-wise no-signaling cheating prover P^*, we have

$$\Pr\left[\begin{array}{l} V_1(\mathsf{st}_V, \boldsymbol{x}, y, \pi^*) = 1 \\ \wedge\, C_\lambda(\boldsymbol{x}) \neq y \end{array} \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q) \end{array}\right] \leq \mathsf{negl}(\lambda) \tag{3.1}$$

for every $\lambda \in \mathbb{N}$.

Toward this goal, for any sufficiently large $\kappa_{\max}$ and any $\kappa_{\max}$-wise no-signaling cheating prover P^*, we assume that we have

$$\Pr\left[V_1(\mathsf{st}_V, \boldsymbol{x}, y, \pi^*) = 1 \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q) \end{array}\right] \geq \frac{1}{\mathsf{poly}(\lambda)} \tag{3.2}$$

for infinitely many $\lambda \in \mathbb{N}$ (let Λ be the set of those λ's) and show that we have

$$\Pr\left[C_\lambda(\boldsymbol{x}) \neq y \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q) \end{array}\right] \leq \mathsf{negl}(\lambda) \tag{3.3}$$

for every sufficiently large $\lambda \in \Lambda$. Clearly, showing Equation (3.3) while assuming Equation (3.2) is sufficient for showing Equation (3.1) (this is because it implies that for every polynomial poly and every sufficiently large $\lambda \in \mathbb{N}$, we have $V_1(\mathsf{st}_V, \boldsymbol{x}, y, \pi^*) = 1 \wedge C_\lambda(\boldsymbol{x}) \neq y$ with probability at most $1/\mathsf{poly}(\lambda)$ since we either have $V_1(\mathsf{st}_V, \boldsymbol{x}, y, \pi^*) = 1$ with probability at most $1/\mathsf{poly}(\lambda)$ or have $C_\lambda(\boldsymbol{x}) \neq y$ with probability at most $1/\mathsf{poly}(\lambda)$).

To explain the overall structure of our analysis, we first show Equation (3.3) while assuming the following (strong) simplifying assumptions instead of Equation (3.2).

Simplifying Assumption 1. P^* convinces the verifier V with overwhelming probability. That is, we have

$$\Pr\left[V_1(\mathsf{st}_V, \boldsymbol{x}, y, \pi^*) = 1 \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q) \end{array} \right] \geq 1 - \mathsf{negl}(\lambda) \tag{3.4}$$

for infinitely many $\lambda \in \mathbb{N}$. (In what follows, we override the definition of Λ and let it be the set of these λ's.) ◇

Simplifying Assumption 2. P^* creates a proof that passes each of Linearity Test, Tensor-Product Test, and SAT Test *on any points* with overwhelming probability. That is, for every sufficiently large $\lambda \in \Lambda$, we have the following. (We assume without loss of generality that P^* always outputs a PCP proof $\pi^* = (\pi_f^*, \pi_g^*)$ that consists of two functions π_f^* and π_g^*.)

- **(Linearity of π_f^*.)** For every $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{F}^N$,

$$\Pr\left[\pi_f^*(\boldsymbol{u}) + \pi_f^*(\boldsymbol{v}) = \pi_f^*(\boldsymbol{u} + \boldsymbol{v}) \;\middle|\; (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u} + \boldsymbol{v}\}) \right] \geq 1 - \mathsf{negl}(\lambda) \;\;, \tag{3.5}$$

- **(Linearity of π_g^*.)** For every $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{F}^{N^2}$,

$$\Pr\left[\pi_g^*(\boldsymbol{u}) + \pi_g^*(\boldsymbol{v}) = \pi_g^*(\boldsymbol{u} + \boldsymbol{v}) \;\middle|\; (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u} + \boldsymbol{v}\}) \right] \geq 1 - \mathsf{negl}(\lambda) \;\;, \tag{3.6}$$

- **(Tensor-Product Consistency of π_f^*, π_g^*.)** For every $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{F}^N$,

$$\Pr\left[\pi_f^*(\boldsymbol{u})\pi_f^*(\boldsymbol{v}) = \pi_g^*(\boldsymbol{u} \otimes \boldsymbol{v}) \;\middle|\; (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u} \otimes \boldsymbol{v}\}) \right] \geq 1 - \mathsf{negl}(\lambda) \;\;, \tag{3.7}$$

- **(SAT Consistency of π_f^*, π_g^*.)** For every $\boldsymbol{\sigma} \in \mathbb{F}^M$,

$$\Pr\left[\pi_f^*(\boldsymbol{\psi}_\sigma) + \pi_g^*(\boldsymbol{\psi}'_\sigma) = c_\sigma \;\middle|\; (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, \{\boldsymbol{\psi}_\sigma, \boldsymbol{\psi}'_\sigma\}) \right] \geq 1 - \mathsf{negl}(\lambda) \;\;. \tag{3.8}$$

◇

At the end of this subsection, we explain how we remove these simplifying assumptions in the actual analysis.

Under the above two simplifying assumptions, we obtain Equation (3.3) as follows. Notice that when the statement is true and the PCP proof is correctly generated, the first part of PCP proof, $\pi_f(\boldsymbol{v}) = \langle \boldsymbol{v}, \boldsymbol{w} \rangle$, is the linear function whose coefficient vector is the satisfying assignment $\boldsymbol{w}$ of the system of equations $\Psi = \{\Psi_i(\boldsymbol{z}) = c_i\}_{i \in [M]}$, and thus the satisfying assignment on any variable z_i can be recovered by appropriately evaluating π_f. (Concretely, given π_f, we can obtain the satisfying assignment on z_i by evaluating π_f on $\boldsymbol{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbb{F}^N$, where only the i-th element of $\boldsymbol{e}_i$ is 1). Now, we first observe that we can obtain Equation (3.3) by showing that the "cheating assignment" that is recovered from the cheating prover P^* is "correct" in the following two ways.

1. The assignment on z_N (which represents the value of the output gate) is equal to the claimed output value y. That is, for every sufficiently large $\lambda \in \Lambda$,

$$\Pr\left[\pi^*_f(\boldsymbol{e}_N) = y \;\middle|\; (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, \{\boldsymbol{e}_N\})\right] \geq 1 - \mathsf{negl}(\lambda) \ . \tag{3.9}$$

2. The assignment on z_N is equal to the actual output value $C_\lambda(\boldsymbol{x})$. That is, for every sufficiently large $\lambda \in \Lambda$,

$$\Pr\left[\pi^*_f(\boldsymbol{e}_N) = C_\lambda(\boldsymbol{x}) \;\middle|\; (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, \{\boldsymbol{e}_N\})\right] \geq 1 - \mathsf{negl}(\lambda) \ . \tag{3.10}$$

Indeed, given Equations (3.9) and (3.10), we can easily obtain Equation (3.3) as follows: first, we obtain

$$\Pr\left[C_\lambda(\boldsymbol{x}) = y \;\middle|\; (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, \{\boldsymbol{e}_N\})\right] \geq 1 - \mathsf{negl}(\lambda)$$

by applying the union bound on Equations (3.9) and (3.10); then, we obtain Equation (3.3) by using the no-signaling property of P^* to argue that the probability of $C_\lambda(\boldsymbol{x}) = y$ holding decreases only negligibly when the queries to P^* are changed from $\{\boldsymbol{e}_N\}$ to $\{\boldsymbol{e}_N\} \cup Q$ and from $\{\boldsymbol{e}_N\} \cup Q$ to Q.[9] (In this argument, we rely on the fact that the distinguisher in the no-signaling game can check $C_\lambda(\boldsymbol{x}) \stackrel{?}{=} y$ efficiently.) Therefore, to conclude the analysis (under the simplifying assumptions), it remains to prove Equations (3.9) and (3.10).

3.3.1 Step 1. Showing consistency with the claimed computation.

First, we explain how we obtain Equation (3.9) under the simplifying assumptions on P^*.

To obtain Equation (3.9), we actually prove a stronger claim on the cheating assignment. Recall that from the construction of $\Psi = \{\Psi_i(\boldsymbol{z}) = c_i\}_{i \in [M]}$, each equation of Ψ is defined with at most three variables, and in particular each equation $\Psi_i(\boldsymbol{z}) = c_i$ can be written as

$$\sum_{j \in \{\alpha,\beta,\gamma\}} d_j z_j + \sum_{j,k \in \{\alpha,\beta,\gamma\}} d_{j,k} z_j z_k = c_i$$

for some $\alpha, \beta, \gamma \in [N]$ ($\alpha < \beta < \gamma$), $d_j \in \{-1, 0, 1\}$ ($j \in \{\alpha, \beta, \gamma\}$), and $d_{j,k} \in \{-1, 0, 1\}$ ($j, k \in \{\alpha, \beta, \gamma\}$). Then, we consider the following claim.

1′. **(Consistency with Claimed Computation)** For any $i \in [M]$ and $\alpha, \beta, \gamma \in [N]$ ($\alpha < \beta < \gamma$) such that the equation $\Psi_i(\boldsymbol{z}) = c_i$ can be written as

$$\sum_{j \in \{\alpha,\beta,\gamma\}} d_j z_j + \sum_{j,k \in \{\alpha,\beta,\gamma\}} d_{j,k} z_j z_k = c_i$$

for some $d_j \in \{-1, 0, 1\}$ ($j \in \{\alpha, \beta, \gamma\}$) and $d_{j,k} \in \{-1, 0, 1\}$ ($j, k \in \{\alpha, \beta, \gamma\}$), the cheating assignment on $z_\alpha, z_\beta, z_\gamma$ is a satisfying assignment of this equation. That is, for every sufficiently large $\lambda \in \Lambda$ and every i and α, β, γ as above, we have

$$\Pr\left[\mathsf{Consist}_i(C_\lambda, \boldsymbol{x}, y, \pi^*) \;\middle|\; (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, \{\boldsymbol{e}_\alpha, \boldsymbol{e}_\beta, \boldsymbol{e}_\gamma\})\right] \geq 1 - \mathsf{negl}(\lambda) \ , \tag{3.11}$$

where $\mathsf{Consist}_i(C_\lambda, \boldsymbol{x}, y, \pi^*)$ is the event that we have

$$\sum_{j \in \{\alpha,\beta,\gamma\}} d_j \pi^*_f(\boldsymbol{e}_j) + \sum_{j,k \in \{\alpha,\beta,\gamma\}} d_{j,k} \pi^*_f(\boldsymbol{e}_j) \pi^*_f(\boldsymbol{e}_k) = c_i \ .$$

[9] We assume $\kappa_{\max}(\lambda) \geq \kappa_V(\lambda) + 1$, where κ_V is the query complexity of V,

Clearly, this claim implies Equation (3.9) since Ψ has the equation $z_N = y$.

Hence, we focus on showing the stronger claim that Equation (3.11) holds. Fix any sufficiently large $\lambda \in \Lambda$ and any $i \in [M]$. Assume for concreteness that the equation $\Psi_i(z) = c_i$ can be written as $-z_\gamma + z_\alpha z_\beta = 0$. (The other cases can be proven similarly.) Under this assumption, our goal is to show

$$\Pr\left[-\pi^*_f(\boldsymbol{e}_\gamma) + \pi^*_f(\boldsymbol{e}_\alpha)\pi^*_f(\boldsymbol{e}_\beta) = 0 \;\middle|\; (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, \{\boldsymbol{e}_\alpha, \boldsymbol{e}_\beta, \boldsymbol{e}_\gamma\})\right] \geq 1 - \mathsf{negl}(\lambda) \ . \tag{3.12}$$

First, we obtain

$$\Pr\left[\pi^*_f(-\boldsymbol{e}_\gamma) + \pi^*_g(\boldsymbol{e}_\alpha \otimes \boldsymbol{e}_\beta) = 0 \;\middle|\; (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, \{-\boldsymbol{e}_\gamma, \boldsymbol{e}_\alpha \otimes \boldsymbol{e}_\beta\})\right] \geq 1 - \mathsf{negl}(\lambda) \tag{3.13}$$

by considering $\boldsymbol{\sigma} = \boldsymbol{e}_i \in \mathbb{F}^M$ in the SAT consistency of π^*_f, π^*_g (Equation (3.8) of Simplifying Assumption 2). Second, we obtain

$$\Pr\left[\pi^*_f(-\boldsymbol{e}_\gamma) = -\pi^*_f(\boldsymbol{e}_\gamma) \;\middle|\; (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, \{\boldsymbol{e}_\gamma, -\boldsymbol{e}_\gamma\})\right] \geq 1 - \mathsf{negl}(\lambda) \tag{3.14}$$

as a corollary of the linearity of π^*_f (Equations (3.5) of Simplifying Assumption 2),[10] and obtain

$$\Pr\left[\pi^*_g(\boldsymbol{e}_\alpha \otimes \boldsymbol{e}_\beta) = \pi^*_f(\boldsymbol{e}_\alpha)\pi^*_f(\boldsymbol{e}_\beta) \;\middle|\; (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, \{\boldsymbol{e}_\alpha, \boldsymbol{e}_\beta, \boldsymbol{e}_\alpha \otimes \boldsymbol{e}_\beta\})\right] \geq 1 - \mathsf{negl}(\lambda) \tag{3.15}$$

from the tensor-product consistency of π^*_f, π^*_g (Equation (3.7) of Simplifying Assumption 2). Now, we obtain Equation (3.12) as desired from Equations (3.13), (3.14), (3.15), the union bound, and the no-signaling property of P^*.[11]

3.3.2 Step 2. Showing consistency with the actual computation.

Next, we explain how we obtain Equation (3.10) under the simplifying assumptions on P^*.

Recall that, without loss of generality, we assume that arithmetic circuits are "layered," i.e., the gates in a circuit can be partitioned into layers such that (1) the first layer consists of the input gates and the last layer consists of the output gate, and (2) the gates in the i-th layer have children in the $(i-1)$-th layer.

The overall strategy is to prove Equation (3.10) by induction on the layers. For any circuit C_λ, let us use the following notations.

- $\ell_{\max}$ is the number of the layers, and N_i is the number of the wires in layer i (i.e., the number of the outgoing wires from the gates in layer i). We assume that the numbering of the wires are consistent with the numbering of the layers, i.e., the first N_1 wires are those that are in the first layer, the next N_2 wires are those that are in the second layer, etc.

- $D_1, \ldots, D_{\ell_{\max}}$ are the subset of $\mathbb{F}^N$ such that for every $\ell \in [\ell_{\max}]$,

$$D_\ell := \{\boldsymbol{v} = (v_1, \ldots, v_N) \mid v_i = 0 \text{ for } \forall i \notin \{N_{\leq \ell-1} + 1, \ldots, N_{\leq \ell-1} + N_\ell\}\} \ ,$$

  where $N_{\leq \ell-1} := \sum_{i \in [\ell-1]} N_i$. Notice that when the first part of the correct PCP proof, $\pi_f(\boldsymbol{v}) = \langle \boldsymbol{v}, \boldsymbol{w} \rangle$, is evaluated on $v_\ell \in D_\ell$, it returns a linear combination of the correct wire values of layer ℓ.

[10] Concretely, we first obtain $\Pr\left[\pi^*_f(\boldsymbol{0}) = 0 \mid (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, \{\boldsymbol{0}\})\right] \geq 1 - \mathsf{negl}(\lambda)$ from the linearity of π^*_f and then obtains Equation (3.14).

[11] To use the union bound on Equations (3.13), (3.14), (3.15), we need to argue that every probability in these equations does not decrease non-negligibly when we obtain π^* by querying $\{\boldsymbol{e}_\alpha, \boldsymbol{e}_\beta, \boldsymbol{e}_\gamma, -\boldsymbol{e}_\gamma, \boldsymbol{e}_\alpha \otimes \boldsymbol{e}_\beta\}$ to P^*. We can show that every probability indeed does not decrease non-negligibly by using the no-signaling property of P^*.

Now, to prove Equation (3.10), we show that the following three claims holds for every sufficiently large $\lambda \in \Lambda$.

1. The cheating PCP proof is equal to the correct PCP proof on random λ points in D_1. That is,

$$\Pr_{U_1,\pi^*}\left[\bigwedge_{\boldsymbol{u}\in U_1} \pi^*_f(\boldsymbol{u}) = \pi_f(\boldsymbol{u})\right] \geq 1 - \mathsf{negl}(\lambda) \ , \tag{3.16}$$

   where the probability is taken over $\boldsymbol{u}_{1,i} \leftarrow D_1$ ($i \in [\lambda]$), $U_1 := \{\boldsymbol{u}_{1,i}\}_{i\in[\lambda]}$, and $(\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, U_1)$, and π_f is the correct PCP proof that is generated by $\pi := P(C_\lambda, \boldsymbol{x})$.

2. For every $\ell \in [\ell_{\max}]$, if the cheating PCP proof is equal to the correct PCP proof on random λ points in D_ℓ, the former is actually equal to the latter on *any* point in D_ℓ. That is, for any $\boldsymbol{v} \in D_\ell$,

$$\Pr_{\boldsymbol{v},U_\ell,\pi^*}\left[\pi^*_f(\boldsymbol{v}) = \pi_f(\boldsymbol{v}) \ \middle|\ \bigwedge_{\boldsymbol{u}\in U_\ell} \pi^*_f(\boldsymbol{u}) = \pi_f(\boldsymbol{u})\right] \geq 1 - \mathsf{negl}(\lambda) \ , \tag{3.17}$$

   where the probability is taken over $\boldsymbol{u}_{\ell,i} \leftarrow D_\ell$ ($i \in [\lambda]$), $U_\ell := \{\boldsymbol{u}_{\ell,i}\}_{i\in[\lambda]}$, and $(\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, \{\boldsymbol{v}\} \cup U_\ell)$.

3. For every $\ell \in [\ell_{\max} - 1]$, if the cheating PCP proof is equal to the correct PCP proof on random λ points in D_ℓ, the former is also equal to the latter equal on random λ points in $D_{\ell+1}$. That is,

$$\Pr_{U_\ell,U_{\ell+1},\pi^*}\left[\bigwedge_{\boldsymbol{u}\in U_{\ell+1}} \pi^*_f(\boldsymbol{u}) = \pi_f(\boldsymbol{u}) \ \middle|\ \bigwedge_{\boldsymbol{u}\in U_\ell} \pi^*_f(\boldsymbol{u}) = \pi_f(\boldsymbol{u})\right] \geq 1 - \mathsf{negl}(\lambda) \ , \tag{3.18}$$

   where the probability is taken over $\boldsymbol{u}_{\ell,i} \leftarrow D_\ell$ ($i \in [\lambda]$), $\boldsymbol{u}_{\ell+1,i} \leftarrow D_{\ell+1}$ ($i \in [\lambda]$), $U_\ell := \{\boldsymbol{u}_{\ell,i}\}_{i\in[\lambda]}$, $U_{\ell+1} := \{\boldsymbol{u}_{\ell+1,i}\}_{i\in[\lambda]}$, and $(\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, U_\ell \cup U_{\ell+1})$.

Observe that we can indeed obtain Equation (3.10) from the above three claims since Equation (3.17) implies that we can obtain Equation (3.10) by just showing

$$\Pr_{U_{\ell_{\max}},\pi^*}\left[\bigwedge_{\boldsymbol{u}\in U_{\ell_{\max}}} \pi^*_f(\boldsymbol{u}) = \pi_f(\boldsymbol{u})\right] \geq 1 - \mathsf{negl}(\lambda)$$

(this is because we have $\pi_f(\boldsymbol{e}_N) = C_\lambda(\boldsymbol{x})$ from the construction of our PCP system), and we can obtain this inequation by repeatedly using Equation (3.18) on top of Equation (3.16).[12] Thus, what remain to prove are Equations (3.16), (3.17), (3.18).

1. First, we obtain Equation (3.16) from the linearity of π^*_f (Equations (3.5) of Simplifying Assumption 2) and the consistency with the claimed computation (Equation (3.11) in Step 1) as follows. At a very high level, we first reduce the problem of showing Equation (3.16) to the problem of showing

$$\forall i \in [n]\colon\ \Pr\left[\pi^*_f(\boldsymbol{e}_i) = \pi_f(\boldsymbol{e}_i) \ \middle|\ (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, \{\boldsymbol{e}_i\})\right] \geq 1 - \mathsf{negl}(\lambda)$$

[12]Formally, we need to argue that the probabilities in these inequations do not decrease non-negligibly when we change the queries to P^*, which we can show by relying on the no-signaling property of P^*. A key point is that the number of the queries to P^* can be bounded by a fixed polynomial in λ.

by relying on the linearity of π^*_f (the key point on this reduction is that any $v \in D_1$ can be written as a linear combination of $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n \in \mathbb{F}^N$), and then just observe that this inequality indeed holds because of the consistency with the claimed computation (the key points on this observation are that Ψ has the equation $z_i = x_i$ for every $i \in [n]$ and that we have $\pi_f(\boldsymbol{e}_i) = x_i$ from the construction of our PCP system).

2. Second, we obtain Equation (3.17) by considering a mental experiment where $U_\ell = \{\boldsymbol{u}_{\ell,i}\}_{i\in[\lambda]}$ is sampled in an alternative way. Specifically, we consider an experiment where for each $\boldsymbol{u}_{\ell,i}$ is sampled by choosing random $\boldsymbol{r}_i \in D_\ell$ and $b_i \in \{0,1\}$ and then defining $\boldsymbol{u}_{\ell,i}$ by $\boldsymbol{u}_{\ell,i} := \boldsymbol{r}_i$ if $b_i = 0$ and by $\boldsymbol{u}_{\ell,i} := \boldsymbol{v} + \boldsymbol{r}_i$ if $b_i = 1$. Since each $\boldsymbol{u}_{\ell,i}$ is still uniformly distributed, it suffices to show Equation (3.17) w.r.t. this mental experiment; in addition, due to the no-signaling property of P^*, we can further change the experiment so that π^* is obtained by

$$(\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, \{\boldsymbol{v}\} \cup \{\boldsymbol{r}_i, \boldsymbol{v} + \boldsymbol{r}_{\ell,i}\}_{i\in[\lambda]}) \ .$$

Now, we obtain Equation (3.17) by combining the following two observations.

   (a) By a simple calculation, we can obtain Equation (3.17) from

$$\Pr_{\boldsymbol{v}, U_\ell, \pi^*}\left[\pi^*_f(\boldsymbol{v}) \neq \pi_f(\boldsymbol{v}) \wedge \left(\bigwedge_{\boldsymbol{u}\in U_\ell} \pi^*_f(\boldsymbol{u}) = \pi_f(\boldsymbol{u})\right)\right] \leq \mathsf{negl}(\lambda) \ . \tag{3.19}$$

   (We assume that $\wedge_{\boldsymbol{u}\in U_\ell}\pi^*_f(\boldsymbol{u}) = \pi_f(\boldsymbol{u})$ holds with high probability, which is indeed the case in our situation.)

   (b) We can obtain Equation (3.19) by combining the following two observations. First, we have $\pi^*_f(\boldsymbol{v}) \neq \pi_f(\boldsymbol{v})$ only when we have $\pi^*_f(\boldsymbol{v} + \boldsymbol{r}_i) \neq \pi_f(\boldsymbol{v} + \boldsymbol{r}_i)$ or $\pi^*_f(\boldsymbol{r}_i) \neq \pi_f(\boldsymbol{r}_i)$ for every $i \in [\lambda]$. (Indeed, if we have $\pi^*_f(\boldsymbol{v} + \boldsymbol{r}_i) = \pi_f(\boldsymbol{v} + \boldsymbol{r}_i)$ and $\pi^*_f(\boldsymbol{r}_i) = \pi_f(\boldsymbol{r}_i)$ for any $i \in [\lambda]$, we have

$$\pi^*_f(\boldsymbol{v}) = \pi^*_f(\boldsymbol{v} + \boldsymbol{r}_i) - \pi^*_f(\boldsymbol{r}_i) = \pi_f(\boldsymbol{v} + \boldsymbol{r}_i) - \pi_f(\boldsymbol{r}_i) = \pi_f(\boldsymbol{v}) \ ,$$

   where the first equality follows from the linearity of π^*_f (Equation (3.5) of Simplifying Assumption 2).) Second, when we have $\pi^*_f(\boldsymbol{v}+\boldsymbol{r}_i) \neq \pi_f(\boldsymbol{v}+\boldsymbol{r}_i)$ or $\pi^*_f(\boldsymbol{r}_i) \neq \pi_f(\boldsymbol{r}_i)$ for every $i \in [\lambda]$, we have $\wedge_{\boldsymbol{u}\in U_\ell}\pi^*_f(\boldsymbol{u}) = \pi_f(\boldsymbol{u})$ with probability at most $2^{-\lambda}$ since each $\boldsymbol{u}_{\ell,i}$ is defined by taking either $\boldsymbol{r}_i$ or $\boldsymbol{v} + \boldsymbol{r}_i$ randomly.

3. Third, we obtain Equation (3.18) as follows. Just like when we show Equation (3.16) above, we first reduce the problem to showing

$$\forall i \in [N_{\ell+1}]\colon\ \Pr\left[\pi^*_f(\boldsymbol{e}_i) = \pi_f(\boldsymbol{e}_i) \,\middle|\, (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, \{\boldsymbol{e}_i\} \cup U_\ell)\right]_{U_\ell} \geq 1 - \mathsf{negl}(\lambda) \ , \tag{3.20}$$

where the probability is conditioned on $\bigwedge_{\boldsymbol{u}\in U_\ell} \pi^*_f(\boldsymbol{u}) = \pi_f(\boldsymbol{u})$ (in the above probability expression, the character U_ℓ at the right corner represents that the probability is conditioned on this event). Now, let us focus, for simplicity, on the case that i is the output wire of an multiplication gate in the $(\ell+1)$-th layer, where the input wires are j and k in the ℓ-th layer. Then, we obtain Equation (3.20) by first observing that we can obtain Equation (3.20) by combining

$$\Pr\left[\pi^*_f(\boldsymbol{e}_i) = \pi^*_f(\boldsymbol{e}_j)\pi^*_f(\boldsymbol{e}_k) \,\middle|\, (\boldsymbol{x}, y, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, \{\boldsymbol{e}_i, \boldsymbol{e}_j, \boldsymbol{e}_k\} \cup U_\ell)\right]_{U_\ell} \geq 1 - \mathsf{negl}(\lambda) \tag{3.21}$$

and

$$\Pr\left[\begin{array}{l}\pi_f^*(\boldsymbol{e}_j)=\pi_f(\boldsymbol{e}_j)\\ \wedge\, \pi_f^*(\boldsymbol{e}_k)=\pi_f(\boldsymbol{e}_k)\end{array}\;\middle|\; (\boldsymbol{x},y,\pi^*)\leftarrow P^*(1^\lambda, C_\lambda, \{\boldsymbol{e}_i,\boldsymbol{e}_j,\boldsymbol{e}_k\}\cup U_\ell)\right]_{U_\ell} \geq 1-\mathsf{negl}(\lambda) \tag{3.22}$$

(this is because if we have $\pi_f^*(\boldsymbol{e}_i)=\pi_f^*(\boldsymbol{e}_j)\pi_f^*(\boldsymbol{e}_k)$ and $\pi_f^*(\boldsymbol{e}_j)=\pi_f(\boldsymbol{e}_j)\wedge\pi_f^*(\boldsymbol{e}_k)=\pi_f(\boldsymbol{e}_k)$, we have

$$\pi_f^*(\boldsymbol{e}_i)=\pi_f^*(\boldsymbol{e}_j)\pi_f^*(\boldsymbol{e}_k)=\pi_f(\boldsymbol{e}_j)\pi_f(\boldsymbol{e}_k)=\pi_f(\boldsymbol{e}_i)\enspace,$$

where the last equality follows from the construction of our PCP system), and then observing that Equation (3.21) follows from the consistency with the claimed computation (Equation (3.11) in Step 1) and that Equation (3.22) follows Equation (3.17) and the union bound.

3.3.3 How to remove the simplifying assumptions.

In the actual analysis, we remove Simplifying Assumption 1 in the same way as previous works (such as [KRR14, BHK17]), namely by considering a "relaxed verifier" that accepts a PCP proof even when the proof fails to pass a small number of the tests (concretely, we consider a verifier that accepts a proof as long as the proof passes the three tests in at least $\lambda-\mu$ trials, where $\mu=\Theta(\log^2\lambda)$). We use the same argument as the previous works to show that if a cheating prover fools the original verifier with non-negligible probability, there exists a cheating prover that fools the relaxed verifier with overwhelming probability.

As for Simplifying Assumption 2, we remove it by considering the self-corrected version of the cheating proof, i.e., the proof that is obtained by applying Self-Correct on the cheating proof π^*. Our key observation is that an existing analysis of Linearity Test [BLR93, Gol17] can be naturally extended so that it works in the no-signaling PCP setting, as long as we only try to show that the self-corrected cheating proof passes Linearity Test on any points. (In the standard PCP setting, the goal of Linearity Test is to guarantee that the cheating proof is close to a linear function.) Once we show that the self-corrected cheating proof passes Linearity Test on any points, it is relatively easy to show that it also passes Tensor-Product Test and SAT Test on any points.

3.4 Comparison with Previous Analysis

The high level structure of our analysis (under the abovementioned simplifying assumptions) is the same as the analysis of previous non-linear no-signaling PCPs, namely those of Kalai et al. [KRR14] and the subsequent works. Specifically, like these works, we show $C_\lambda(\boldsymbol{x})=y$ by showing that we have $\pi^*(\boldsymbol{e}_N)=y$ and $\pi^*(\boldsymbol{e}_N)=C_\lambda(\boldsymbol{x})$ simultaneously, and show $\pi^*(\boldsymbol{e}_N)=C_\lambda(\boldsymbol{x})$ by induction on layers of C_λ. (In the latter part, we in particular follow the presentation by Paneth and Rothblum [PR17].)

Other than the differences due to the use of linear PCPs, a notable difference between our analysis and the previous one is that our analysis does not require that the statement is represented as an "augmented layered circuit," and only requires that it is represented as a layered circuit. More concretely, while the previous analysis requires that each layer of the circuit is augmented with an additional circuit (which computes a low-degree extension of the wire values of the layer and then applies low-degree tests on the low-degree extension), our analysis does not require such augmentation and only requires that the circuit is layered. At a high level, we do not require this augmentation since in the induction for showing $\pi^*(\boldsymbol{e}_N)=C_\lambda(\boldsymbol{x})$ (Step 2 in the previous subsection), we show that the cheating PCP proof is equal to the correct proof rather than just showing that the wire values that are recovered from the cheating PCP proof are equal to the correct ones. (That is, we do not require the augmentation of the circuit since we consider a stronger claim in the induction, which allows us to use a stronger assumption in the inductive step).

4 Construction of Our No-signaling Linear PCP for $\mathcal{P}$

In this section, we describe our no-signaling linear PCP system (P, V). Let $C : \mathbb{F}^n \to \mathbb{F}^m$ be an arithmetic circuit over a finite field $\mathbb{F}$ of prime order and $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{F}^n$ be an input to C. We use N to denote the number of wires in C, and assume that the first n wires are those that take the values of the input gates and the last m ones are those that take the value of the output gates.

4.1 PCP Prover P

Given $(C, \boldsymbol{x})$ as input, the PCP prover P first computes $\boldsymbol{y} := C(\boldsymbol{x})$ and obtains the following system of quadratic equations over $\mathbb{F}$, which is designed so that it is satisfiable if and only if $C(\boldsymbol{x}) = \boldsymbol{y}$.

- The variables are $\boldsymbol{z} = (z_1, \ldots, z_N)$.
- For each $i \in \{1, \ldots, n\}$, the system has the equation $z_i = x_i$.
- For each $i, j, k \in [N]$, the system has $z_i + z_j - z_k = 0$ if C has an addition gate with input wires i, j and output wire k, and has $z_i z_j - z_k = 0$ if C has a multiplication gate with input wires i, j and output wire k.
- For each $i \in \{1, \ldots, m\}$, the system has the equation $z_{N-m+i} = y_i$.

Let the above system of quadratic equations be denoted by $\Psi = \{\Psi_i(\boldsymbol{z}) = c_i\}_{i \in [M]}$, where M is the number of the equations. For each $i \in [M]$, let $\boldsymbol{\psi}_i \in \mathbb{F}^N$ and $\boldsymbol{\psi}'_i \in \mathbb{F}^{N^2}$ be the coefficient vectors such that

$$\Psi_i(\boldsymbol{z}) = \langle \boldsymbol{\psi}_i, \boldsymbol{z} \rangle + \langle \boldsymbol{\psi}'_i, \boldsymbol{z} \otimes \boldsymbol{z} \rangle \ . \tag{4.1}$$

Let $\boldsymbol{w} = (w_1, \ldots, w_N)$ be the satisfying assignment of Ψ. Let $f : \mathbb{F}^N \to \mathbb{F}$ and $g : \mathbb{F}^{N^2} \to \mathbb{F}$ be the linear functions that are defined by $f(\boldsymbol{v}) := \langle \boldsymbol{v}, \boldsymbol{w} \rangle$ and $g(\boldsymbol{v}') := \langle \boldsymbol{v}', \boldsymbol{w} \otimes \boldsymbol{w} \rangle$. Then, the PCP prover P outputs the following linear function $\pi : \mathbb{F}^{N+N^2} \to \mathbb{F}$ as the PCP proof.

$$\pi(\boldsymbol{v}) := f(\boldsymbol{v}_1) + g(\boldsymbol{v}_2) \text{ for } \forall \boldsymbol{v} = (\boldsymbol{v}_1, \boldsymbol{v}_2) \in \mathbb{F}^{N+N^2}, \text{ where } \boldsymbol{v}_1 \in \mathbb{F}^N, \boldsymbol{v}_2 \in \mathbb{F}^{N^2}.$$

Remark 3. For simplicity, in what follows we usually think that P outputs two linear functions $\pi_f := f$ and $\pi_g := g$ as the PCP proof. This is without loss of generality since the verifier can evaluate f and g given access to π. ◇

4.2 PCP Verifier V

Given $(C, \boldsymbol{x}, \boldsymbol{y})$ as input, the PCP verifier V first computes the system of quadratic equations Ψ. Next, given oracle access to the PCP proof (π_f, π_g), the PCP verifier does the following tests λ times in parallel, and accepts the proof if all the tests in all the λ trials are accepted.

- **Linearity Test.** Choose random points $\boldsymbol{r}_1, \boldsymbol{r}_2 \in \mathbb{F}^N$ and $\boldsymbol{r}'_1, \boldsymbol{r}'_2 \in \mathbb{F}^{N^2}$, and check the following.

$$\pi_f(\boldsymbol{r}_1) + \pi_f(\boldsymbol{r}_2) \stackrel{?}{=} \pi_f(\boldsymbol{r}_1 + \boldsymbol{r}_2) \qquad \text{and} \qquad \pi_g(\boldsymbol{r}'_1) + \pi_g(\boldsymbol{r}'_2) \stackrel{?}{=} \pi_g(\boldsymbol{r}'_1 + \boldsymbol{r}'_2) \ .$$

- **Tensor-Product Test.** Let $\textsf{Self-Correct}^\pi$ be the algorithm in Figure 2. Then, in Tensor-Product Test, choose two random points $\boldsymbol{r}_1, \boldsymbol{r}_2 \in \mathbb{F}^N$, run

$$a_{\boldsymbol{r}_1} \leftarrow \textsf{Self-Correct}^\pi(\boldsymbol{r}_1) \ , \quad a_{\boldsymbol{r}_2} \leftarrow \textsf{Self-Correct}^\pi(\boldsymbol{r}_2) \ , \quad \text{and} \quad a_{\boldsymbol{r}_1 \otimes \boldsymbol{r}_2} \leftarrow \textsf{Self-Correct}^\pi(\boldsymbol{r}_1 \otimes \boldsymbol{r}_2) \ ,$$

and check the following.

$$a_{\boldsymbol{r}_1} a_{\boldsymbol{r}_2} \stackrel{?}{=} a_{\boldsymbol{r}_1 \otimes \boldsymbol{r}_2} \ .$$

- **SAT Test.** Choose a random point $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_M) \in \mathbb{F}^M$ and define a quadratic function $\Psi_{\boldsymbol{\sigma}} : \mathbb{F}^N \to \mathbb{F}$ as

$$\Psi_{\boldsymbol{\sigma}}(z) \coloneqq \sum_{i=1}^{M} \sigma_i \Psi_i(z) \ .$$

Let $\boldsymbol{\psi}_{\boldsymbol{\sigma}} \in \mathbb{F}^N$ and $\boldsymbol{\psi}'_{\boldsymbol{\sigma}} \in \mathbb{F}^{N^2}$ be the coefficient vectors such that

$$\Psi_{\boldsymbol{\sigma}}(z) = \langle \boldsymbol{\psi}_{\boldsymbol{\sigma}}, z \rangle + \langle \boldsymbol{\psi}'_{\boldsymbol{\sigma}}, z \otimes z \rangle \ . \tag{4.2}$$

Let $c_{\boldsymbol{\sigma}} \coloneqq \sum_{i=1}^{M} \sigma_i c_i$.

Then, in SAT Test, run

$$a_{\psi_\sigma} \leftarrow \textsf{Self-Correct}^\pi(\boldsymbol{\psi}_{\boldsymbol{\sigma}}) \qquad \text{and} \qquad a_{\psi'_\sigma} \leftarrow \textsf{Self-Correct}^\pi(\boldsymbol{\psi}'_{\boldsymbol{\sigma}}) \ ,$$

and check the following.

$$a_{\psi_\sigma} + a_{\psi'_\sigma} \stackrel{?}{=} c_{\boldsymbol{\sigma}} \ .$$

We remark that, formally, $V = (V_0, V_1)$ is a pair of two algorithms as required by Definition 1, where $V_0(1^\lambda, C)$ outputs a set of the queries Q for the above tests along with its internal state st_V, and $V_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi|_Q)$ performs the above tests given the answers $\pi|_Q$ from the PCP proof. The internal state st_V that V_0 outputs is $(\boldsymbol{\sigma}_{\text{in}}, \boldsymbol{\sigma}_{\text{out}})$, where

$$\boldsymbol{\sigma}_{\text{in}} \coloneqq (\sigma_1, \ldots, \sigma_n) \in \mathbb{F}^n \qquad \text{and} \qquad \boldsymbol{\sigma}_{\text{out}} \coloneqq (\sigma_{M-m+1}, \ldots, \sigma_M) \in \mathbb{F}^m \ ,$$

where it is assumed that the first n equations in Ψ (i.e., the equations $\{\Psi_i(z) = c_i\}_{i\in[n]}$) are those that are associated with the input gates (i.e., $\{z_i = x_i\}_{i\in[n]}$) and the last m equations in Ψ (i.e.the equations $\{\Psi_{M-m+i}(z) = c_{M-m+i}\}_{i\in[m]}$) are those that are associated with the output gates (i.e., $\{z_{M-m+i} = y_i\}_{i\in[n]}$). Note that $V_0(1^\lambda, C)$ can indeed choose all the queries in parallel (without knowing the input $\boldsymbol{x}$ and the output $\boldsymbol{y}$) since each of the queries is chosen independently of the results of the other queries and in addition the coefficient vectors of the equations of Ψ (i.e., $\{\boldsymbol{\psi}_i, \boldsymbol{\psi}'_i\}_{i\in M}$) can be computed from the circuit C in SAT Test. Also, note that $V_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi|_Q)$ can indeed perform the test (without knowing the circuit C) since $c_{\boldsymbol{\sigma}} = \langle \boldsymbol{\sigma}_{\text{in}}, \boldsymbol{x} \rangle + \langle \boldsymbol{\sigma}_{\text{out}}, \boldsymbol{y} \rangle$ can be computed from st_V in SAT Test.

Remark 4 (Query Complexity.). By inspection, one can see that that the query complexity of V is $\kappa_V(\lambda) \coloneqq \lambda(10\lambda + 6)$. ◇

Remark 5 (Efficiency.). By inspection, one can see that the running time of P is $\mathsf{poly}(|C|)$, the running time of V_0 is $\mathsf{poly}(\lambda + |C|)$, and the running time of V_1 is $\mathsf{poly}(\lambda + |\boldsymbol{x}| + |\boldsymbol{y}|)$. ◇

Algorithm $\mathsf{Self\text{-}Correct}^{\pi}(\boldsymbol{v} \in \mathbb{F}^N \cup \mathbb{F}^{N^2})$**.**

1. Choose λ random points $\boldsymbol{r}_{\boldsymbol{v},1}, \ldots, \boldsymbol{r}_{\boldsymbol{v},\lambda}$ from $\mathbb{F}^N$ if $\boldsymbol{v} \in \mathbb{F}^N$ and choose them from $\mathbb{F}^{N^2}$ if $\boldsymbol{v} \in \mathbb{F}^{N^2}$.

2. For each $i \in [\lambda]$, let
$$a_{\boldsymbol{v}}^{(i)} := \begin{cases} \pi_f(\boldsymbol{v} + \boldsymbol{r}_{\boldsymbol{v},i}) - \pi_f(\boldsymbol{r}_{\boldsymbol{v},i}) & \text{if } \boldsymbol{v} \in \mathbb{F}^N \\ \pi_g(\boldsymbol{v} + \boldsymbol{r}_{\boldsymbol{v},i}) - \pi_g(\boldsymbol{r}_{\boldsymbol{v},i}) & \text{if } \boldsymbol{v} \in \mathbb{F}^{N^2} \end{cases} .$$

3. Let
$$a_{\boldsymbol{v}} := \text{majority}\left(a_{\boldsymbol{v}}^{(1)}, \ldots, a_{\boldsymbol{v}}^{(\lambda)}\right) .$$

4. Output $a_{\boldsymbol{v}}$.

Figure 2: The self-correction algorithm $\mathsf{Self\text{-}Correct}$, which works given oracle access to $\pi = (\pi_f, \pi_g)$

4.3 Security Statement

From Section 5 to Section 9, we prove the following theorem, which states the no-signaling soundness of our PCP system.

Theorem 1 (No-signaling Soundness of (P, V))**.** *Let (P, V) be the PCP system in Sections 4.1 and 4.2, $\{C_\lambda\}_{\lambda \in \mathbb{N}}$ be any circuit family, and $\kappa_{\max}$ be any polynomial such that $\kappa_{\max}(\lambda) \geq 2\lambda \cdot \max(8\lambda + 3, m_\lambda) + \kappa_V(\lambda)$, where m_λ is the output length of C_λ and κ_V is the query complexity of (P, V). Then, for any $\kappa_{\max}$-wise (computational) no-signaling cheating prover P^*, there exists a negligible function* negl *such that for every $\lambda \in \mathbb{N}$,*

$$\Pr\left[V_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*) = 1 \wedge C_\lambda(\boldsymbol{x}) \neq \boldsymbol{y} \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q) \end{array} \right] \leq \mathsf{negl}(\lambda) .$$

Outline of the proof of Theorem 1. In Section 5, we introduce a "relaxed verifier" such that if a cheating prover fools the original verifier with non-negligible probability, there exists another cheating prover that fools the relaxed verifier with overwhelming probability. In Section 6, we show that if a cheating verifier convinces the relaxed verifier with overwhelming probability, we can obtain a "self-corrected PCP proof" from the cheating prover, where the self-corrected PCP proof satisfies several useful properties (namely, the ability to pass Linearity Test, Tensor-Product Test, and SAT Test on any points). In Section 7, we show that if a cheating verifier convinces the relaxed verifier with overwhelming probability, the self-corrected PCP proof matches the claimed computation, i.e., the assignment that we obtain from the self-corrected PCP proof on any small number of variables of Ψ is a (locally) satisfying assignment. In Section 8, we show that if a cheating verifier convinces the relaxed verifier with overwhelming probability, the self-corrected PCP proof matches the correct PCP proof. In Section 9, we conclude the proof by combining what is shown in the preceding four sections.

5 Analysis of Our PCP: Step 1 (Relaxed Verifier)

In this section, we introduce a "relaxed verifier" $\mathbb{V}$ for our PCP system. The relaxed verifier is designed so that if a no-signaling cheating prover can fool the original verifier with non-negligible probability, another no-signaling cheating prover can fool the relaxed verifier with overwhelming success probability. In subsequent sections, we show the no-signaling soundness of our PCP system by showing that any no-signaling cheating prover cannot fool the relaxed verifier with overwhelming success probability.

5.1 Construction

Recall that the original PCP verifier V, described in Section 4.2, makes λ trials of tests (where each trial consists of Linearity Test, Tensor-Product Test, and SAT Test) and accepts the proof if all the tests in all the λ trials succeed.

The relaxed verifier $\mathbb{V}$ makes λ trials of tests in the same way as the original PCP verifier does, but accepts the proof even when tests in at most μ trials fail (that is, accepts the proof if all the tests in at least $\lambda - \mu$ trials succeed), where $\mu = \Theta(\log^2 \lambda)$ is a parameter.[13] We remark that, just like the original verifier $V = (V_0, V_1)$, the relaxed verifier $\mathbb{V}$ is actually a pair of algorithms, $(\mathbb{V}_0, \mathbb{V}_1)$, where $\mathbb{V}_0$ makes the queries and $\mathbb{V}_1$ performs the tests. From the construction, $\mathbb{V}_0$ is identical with V_0.

5.2 Analysis

Lemma 1. *Let $\kappa_{\max}$ be any polynomial such that $\kappa_{\max}(\lambda) \geq 2\kappa_V(\lambda)$, where κ_V is the query complexity of (P, V). Then, for any circuit family $\{C_\lambda\}_{\lambda\in\mathbb{N}}$, if there exists a $\kappa_{\max}$-wise no-signaling prover P^* and a constant $c > 0$ such that*

$$\Pr\left[V_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*) = 1 \wedge C_\lambda(\boldsymbol{x}) \neq \boldsymbol{y} \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q) \end{array}\right] \geq \lambda^{-c} \tag{5.1}$$

holds for infinitely many $\lambda \in \mathbb{N}$, there exists a $(\kappa_{\max} - \kappa_V)$-wise no-signaling prover $\mathbb{P}^*$ and a negligible function negl *such that*

$$\Pr\left[\mathbb{V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*) = 1 \wedge C_\lambda(\boldsymbol{x}) \neq \boldsymbol{y} \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow \mathbb{P}^*(1^\lambda, C_\lambda, Q) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \tag{5.2}$$

holds for infinitely many $\lambda \in \mathbb{N}$.

We remark that Brakerski et al. [BHK17] prove essentially the same lemma as Lemma 1 (see Lemma 1 of the full version of their paper [BHK16]). Below, we give a proof of Lemma 1 just for completeness. Since we only use the statement of Lemma 1 in the subsequent sections, the readers who believe this lemma can skip the rest of this section.

Proof. Fix any polynomial $\kappa_{\max}$ such that $\kappa_{\max}(\lambda) \geq 2\kappa_V(\lambda)$, any circuit family $\{C_\lambda\}_{\lambda\in\mathbb{N}}$, any $\kappa_{\max}$-wise no-signaling cheating prover P^*, and any constant $c > 0$, and assume that Equation (5.1) holds.

The high-level idea of the relaxed cheating prover $\mathbb{P}^*$ is simple: $\mathbb{P}^*$ amplifies the success probability of P^* by executing it repeatedly. That is, on input a set of queries Q, the relaxed cheating prover $\mathbb{P}^*$

[13] Actually, μ can be any function in $\omega(\log \lambda)$ as long as μ is sufficiently smaller than λ.

executes P^* repeatedly to obtain an accepting proof for Q. Two main problem of this high-level idea is that (1) if the same set of queries Q is used in each execution of P^*, the repeated executions of P^* are not mutually independent, so the success probability of P^* is not necessarily amplified, and (2) $\mathbb{P}^*$ cannot see which execution of P^* yields an accepting proof for Q since our PCP system (P, V) is not publicly verifiable. Because of these two problems, the actual construction and analysis of $\mathbb{P}^*$ are not trivial; the details are given below.

Formally, we obtain the relaxed cheating prover $\mathbb{P}^*$ from P^* as follows.

- On input $(1^\lambda, C_\lambda, Q)$, the relaxed cheating prover $\mathbb{P}^*$ first samples

$$(Q_i, \mathsf{st}_{V,i}) \leftarrow V_0(1^\lambda, C_\lambda) \qquad \text{and} \qquad (\boldsymbol{x}_i, \boldsymbol{y}_i, \pi_i^*) \leftarrow P^*(1^\lambda, C_\lambda, Q \cup Q_i)$$

  for $i = 1, \ldots, \lambda^{c+1}$. Then, $\mathbb{P}^*$ finds the first $i^* \in \{1, \ldots, \lambda^{c+1}\}$ such that $V_1(\mathsf{st}_{V,i^*}, (\pi_{i^*}^*)|_{Q_{i^*}}) = 1 \wedge C_\lambda(\boldsymbol{x}_{i^*}) \neq \boldsymbol{y}_{i^*}$ and outputs $(\boldsymbol{x}_{i^*}, \boldsymbol{y}_{i^*}, (\pi_{i^*}^*)|_Q)$ if such i^* exists, and outputs $\perp$ otherwise.

That is, $\mathbb{P}^*$ repeatedly executes P^* many times, where the queries to P^* in the i-th execution are $Q \cup Q_i$ for freshly sampled queries Q_i, and finds a proof that is accepting for Q_i (hoping that it is also accepting for Q).

Next, we analyze $\mathbb{P}^*$ as follows.

Claim 1 (Overwhelming success probability)**.** *There exists a negligible function* negl *such that for infinitely many* $\lambda \in \mathbb{N}$*,*

$$\Pr\left[\mathbb{V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*) = 1 \wedge C_\lambda(\boldsymbol{x}) \neq \boldsymbol{y} \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow \mathbb{P}^*(1^\lambda, C_\lambda, Q) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \ . \tag{5.3}$$

Proof. First, we show that $\mathbb{P}^*$ outputs $\perp$ with negligible probability. Recall that $\mathbb{P}^*$ outputs $\perp$ if there does not exist i^* such that $V_1(\mathsf{st}_{V,i^*}, (\pi_{i^*}^*)|_{Q_{i^*}}) = 1 \wedge C_\lambda(\boldsymbol{x}_{i^*}) \neq \boldsymbol{y}_{i^*}$. Since P^* is $2\kappa_V$-wise no-signaling, it holds that for infinitely many $\lambda \in \mathbb{N}$ and every Q such that $|Q| = \kappa_V(\lambda)$,

$$\begin{aligned} &\Pr\left[V_1(\mathsf{st}_{V,i}, (\pi_i^*)|_{Q_i}) = 1 \wedge C_\lambda(\boldsymbol{x}_i) \neq \boldsymbol{y}_i \;\middle|\; \begin{array}{l} (Q_i, \mathsf{st}_{V,i}) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}_i, \boldsymbol{y}_i, \pi_i^*) \leftarrow P^*(1^\lambda, C_\lambda, Q \cup Q_i) \end{array}\right] \\ &\geq \Pr\left[V_1(\mathsf{st}_{V,i}, \pi_i^*) = 1 \wedge C_\lambda(\boldsymbol{x}_i) \neq \boldsymbol{y}_i \;\middle|\; \begin{array}{l} (Q_i, \mathsf{st}_{V,i}) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}_i, \boldsymbol{y}_i, \pi_i^*) \leftarrow P^*(1^\lambda, C_\lambda, Q_i) \end{array}\right] - \frac{1}{2}\lambda^{-c} \\ &\geq \frac{1}{2}\lambda^{-c} \qquad \text{(from Equation (5.1))} \end{aligned} \tag{5.4}$$

Therefore, the probability that no i^* exists is bounded by $(1 - \lambda^{-c}/2)^{\lambda^{c+1}}$, which is negligible.

Next, we show that under the condition that $\mathbb{P}^*$ does not output $\perp$, $\mathbb{P}^*$ convinces $\mathbb{V}$ except with negligible probability. From the construction of $\mathbb{P}^*$, it suffices to show that for any C_λ, we have

$$\Pr_{Q, Q', \pi^*}\left[\mathbb{V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*|_Q) = 0 \wedge V_1(\mathsf{st}'_V, \pi^*|_{Q'}) = 1\right] \leq \mathsf{negl}(\lambda) \ , \tag{5.5}$$

where the probability is taken over the following sampling of Q, Q', π^*.

1. $(Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda)$.
2. $(Q', \mathsf{st}'_V) \leftarrow V_0(1^\lambda, C_\lambda)$.

3. $(\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q \cup Q')$.

Hence, we focus on showing Equation (5.5). Recall that each of Q, Q' is a set of the queries for λ trials of the tests. We consider a mental experiment where, instead of sampling Q and Q' separately, we sample a set of the queries for 2λ trials of the tests, denoted by $\hat{Q}$, and then define Q, Q' by randomly partitioning $\hat{Q}$ into two. Clearly, the distributions of Q, Q' in this mental experiment is the same as those in the original experiment. Let BAD be the event that $\hat{Q}$ leads to 2λ trials of the tests such that at least μ trials of them are rejecting. (Recall that μ is the parameter of the relaxed verifier $\mathbb{V}$ and we have $\mu = \Theta(\log^2 \lambda)$.) Now, we have

$$\Pr_{\hat{Q},Q,Q',\pi^*}\left[\mathbb{V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*|_Q) = 0 \wedge V_1(\mathsf{st}'_V, \pi^*|_{Q'}) = 1 \wedge \neg\mathsf{BAD}\right] = 0$$

since $\mathbb{V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*|_Q) = 0$ cannot occur unless BAD occurs. Therefore, we have

$$\begin{aligned}
&\Pr_{\hat{Q},Q,Q',\pi^*}\left[\mathbb{V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*|_Q) = 0 \wedge V_1(\mathsf{st}'_V, \pi^*|_{Q'}) = 1\right] \\
&= \Pr_{\hat{Q},Q,Q',\pi^*}\left[\mathbb{V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*|_Q) = 0 \wedge V_1(\mathsf{st}'_V, \pi^*|_{Q'}) = 1 \wedge \mathsf{BAD}\right] \\
&\leq \Pr_{\hat{Q},Q,Q',\pi^*}\left[V_1(\mathsf{st}'_V, \pi^*|_{Q'}) = 1 \wedge \mathsf{BAD}\right] \\
&\leq \left(1 - \frac{\mu}{2\lambda}\right)^\lambda = \mathsf{negl}(\lambda)
\end{aligned}$$

where the last inequality holds since when BAD occurs, we have $V_1(\mathsf{st}'_V, \pi^*|_{Q'}) = 1$ only when all the rejecting queries (namely the queries that lead to rejecting tests) are picked for Q when partitioning $\hat{Q}$ into Q and Q'.

By combining what we show in the above two paragraphs, we obtain Equation (5.3). This concludes the proof of Claim 1. □

Claim 2 (No-signaling property)**.** $\mathbb{P}^*$ *is* $(\kappa_{\max} - \kappa_V)$*-wise no-signaling.*

Proof. We need to show that for any PPT distinguisher $\mathcal{D}_{\mathrm{NS}}$, there exists a negligible function negl such that for every $\lambda \in \mathbb{N}$, every W, W' such that $W' \subset W$ and $|W| \leq \kappa_{\max}(\lambda)$, and every $z \in \{0,1\}^{\mathsf{poly}(\lambda)}$,

$$\left| \begin{array}{l} \Pr\left[\mathcal{D}_{\mathrm{NS}}(C_\lambda, \boldsymbol{x}, \boldsymbol{y}, \pi^*|_{W'}, z) = 1 \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow \mathbb{P}^*(1^\lambda, C_\lambda, W)\right] \\ \quad - \Pr\left[\mathcal{D}_{\mathrm{NS}}(C_\lambda, \boldsymbol{x}, \boldsymbol{y}, \pi^*, z) = 1 \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow \mathbb{P}^*(1^\lambda, C_\lambda, W')\right] \end{array} \right| \leq \mathsf{negl}(\lambda) \enspace .$$

From the construction of $\mathbb{P}^*$, toward this end it suffices to show that for any PPT distinguisher $\mathcal{D}$, there exists a negligible function negl such that for every $\lambda \in \mathbb{N}$, every W, W' such that $W' \subset W$ and $|W| \leq \kappa_{\max}(\lambda)$, and every $z \in \{0,1\}^{\mathsf{poly}(\lambda)}$,

$$\left| \begin{array}{l} \Pr\left[\mathcal{D}(\nu) = 1 \;\middle|\; \begin{array}{l} (Q_i, \mathsf{st}_{V,i}) \leftarrow V_0(1^\lambda, C_\lambda) \text{ for } \forall i \in [\lambda^{c+1}] \\ (\boldsymbol{x}_i, \boldsymbol{y}_i, \pi_i^*) \leftarrow P^*(1^\lambda, C_\lambda, W_\lambda \cup Q_i) \text{ for } \forall i \in [\lambda^{c+1}] \end{array}\right] \\ \quad - \Pr\left[\mathcal{D}(\nu') = 1 \;\middle|\; \begin{array}{l} (Q_i, \mathsf{st}_{V,i}) \leftarrow V_0(1^\lambda, C_\lambda) \text{ for } \forall i \in [\lambda^{c+1}] \\ (\boldsymbol{x}_i, \boldsymbol{y}_i, \pi_i^*) \leftarrow P^*(1^\lambda, C_\lambda, W'_\lambda \cup Q_i) \text{ for } \forall i \in [\lambda^{c+1}] \end{array}\right] \end{array} \right| \leq \mathsf{negl}(\lambda) \enspace , \tag{5.6}$$

where

$$\begin{aligned}
\nu &:= (C_\lambda, W_\lambda, W'_\lambda, \{Q_i, \mathsf{st}_{V,i}, \boldsymbol{x}_i, \boldsymbol{y}_i, (\pi_i^*)|_{W'_\lambda \cup Q_i}\}_{i \in [\lambda^{c+1}]}, z) \enspace , \\
\nu' &:= (C_\lambda, W_\lambda, W'_\lambda, \{Q_i, \mathsf{st}_{V,i}, \boldsymbol{x}_i, \boldsymbol{y}_i, \pi_i^*\}_{i \in [\lambda^{c+1}]}, z) \enspace .
\end{aligned}$$

(This is because given ν or ν', the distinguisher $\mathcal{D}$ can emulate the output of $\mathbb{P}^*$.) Equation (5.6) can be shown easily by using a hybrid argument since the $\kappa_{\max}$-wise no-signaling property of P^* guarantees that for any PPT distinguisher $\mathcal{D}_i$, there exists a negligible function negl such that for every $\lambda \in \mathbb{N}$, every W, W' such that $W' \subset W$ and $|W| \le \kappa_{\max}(\lambda)$, and every $z \in \{0,1\}^{\mathsf{poly}(\lambda)}$,

$$\left| \begin{array}{l} \Pr\left[\mathcal{D}_i(\nu_i) = 1 \;\middle|\; \begin{array}{l} (Q_i, \mathsf{st}_{V,i}) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}_i, \boldsymbol{y}_i, \pi_i^*) \leftarrow P^*(1^\lambda, C_\lambda, W_\lambda \cup Q_i) \end{array}\right] \\ \qquad - \Pr\left[\mathcal{D}_i(\nu_i') = 1 \;\middle|\; \begin{array}{l} (Q_i, \mathsf{st}_{V,i}) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}_i, \boldsymbol{y}_i, \pi_i^*) \leftarrow P^*(1^\lambda, C_\lambda, W_\lambda' \cup Q_i) \end{array}\right] \end{array} \right| \le \mathsf{negl}(\lambda) \enspace ,$$

where

$$\begin{aligned} \nu_i &:= (C_\lambda, W_\lambda, W_\lambda', Q_i, \mathsf{st}_{V,i}, \boldsymbol{x}_i, \boldsymbol{y}_i, (\pi_i^*)|_{W_\lambda' \cup Q_i}, z) \enspace , \\ \nu_i' &:= (C_\lambda, W_\lambda, W_\lambda', Q_i, \mathsf{st}_{V,i}, \boldsymbol{x}_i, \boldsymbol{y}_i, \pi_i^*, z) \enspace . \end{aligned}$$

This concludes the proof of Claim 2. □

From Claim 1 and Claim 2, Lemma 1 follows. □

6 Analysis of Our PCP: Step 2 (Self-Corrected Proof)

In this section, we introduce a "self-correction" procedure with the following property: given any successful no-signaling cheating prover against the relaxed verifier, the self-correction procedure outputs a no-signaling proof that passes Linearity Test, Tensor-Product Test, and SAT Test on *any* points.

6.1 Self-Correction Procedure Self-Correct

For every security parameter $\lambda \in \mathbb{N}$, circuit C, cheating prover P^*, and queries $Q \subset \mathbb{F}^N \cup \mathbb{F}^{N^2}$, we consider the following self-correction procedure.

Algorithm $\mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C, Q)$.

1. For each $\boldsymbol{v} \in Q$, choose λ random points $\boldsymbol{r}_{\boldsymbol{v},1}, \ldots, \boldsymbol{r}_{\boldsymbol{v},\lambda}$ from $\mathbb{F}^N$ if $\boldsymbol{v} \in \mathbb{F}^N$ and choose them from $\mathbb{F}^{N^2}$ if $\boldsymbol{v} \in \mathbb{F}^{N^2}$.
2. Run $(\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C, Q')$, where $Q' = \{\boldsymbol{r}_{\boldsymbol{v},i}, \boldsymbol{v} + \boldsymbol{r}_{\boldsymbol{v},i}\}_{\boldsymbol{v} \in Q, i \in [\lambda]}$.
3. For each $i \in [\lambda]$, define a function $\tilde{\pi}^{(i)} : Q \to \mathbb{F}$ by
$$\tilde{\pi}^{(i)}(\boldsymbol{v}) := \pi^*(\boldsymbol{v} + \boldsymbol{r}_{\boldsymbol{v},i}) - \pi^*(\boldsymbol{r}_{\boldsymbol{v},i}) \text{ for } \forall \boldsymbol{v} \in Q \enspace .$$
4. Define a function $\tilde{\pi} : Q \to \mathbb{F}$ by
$$\tilde{\pi}(\boldsymbol{v}) := \mathrm{majority}\left(\tilde{\pi}^{(1)}(\boldsymbol{v}), \ldots, \tilde{\pi}^{(\lambda)}(\boldsymbol{v})\right) \text{ for } \forall \boldsymbol{v} \in Q \enspace .$$
Let $\tilde{\pi}_f$ be the function that is obtained by restricting the domain of $\tilde{\pi}$ to $Q \cap \mathbb{F}^N$, and $\tilde{\pi}_g$ be the function that is obtained by restricting the domain of $\tilde{\pi}$ to $Q \cap \mathbb{F}^{N^2}$.
5. Output $(\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi})$.

6.2 Basic Properties of $\mathsf{Self\text{-}Correct}$

In this subsection, we observe two basic properties of $\mathsf{Self\text{-}Correct}$. First, we observe that $\mathsf{Self\text{-}Correct}$ is no-signaling in the following sense.

Lemma 2 (No-signaling property of $\mathsf{Self\text{-}Correct}$)**.** *Let $\kappa_{\max}$ be any polynomial, $\{C_\lambda\}_{\lambda\in\mathbb{N}}$ be any circuit family, and P^* be any $\kappa'_{\max}$-wise no-signaling cheating prover, where $\kappa'_{\max}(\lambda) := 2\lambda \cdot \kappa_{\max}(\lambda)$.*

Then, for any PPT Turing machine $\mathcal{D}$, there exists a negligible function negl such that for every $\lambda \in \mathbb{N}$, every Q, Q' such that $Q' \subset Q$ and $|Q| \le \kappa_{\max}(\lambda)$, and every $z \in \{0,1\}^\lambda$,

$$\left| \begin{array}{l} \Pr\left[\mathcal{D}(C_\lambda, \boldsymbol{x}, \boldsymbol{y}, \pi^*|_{Q'}, z) = 1 \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, Q)\right] \\ \qquad - \Pr\left[\mathcal{D}(C_\lambda, \boldsymbol{x}, \boldsymbol{y}, \pi^*, z) = 1 \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, Q')\right] \end{array} \right| \le \mathsf{negl}(\lambda) \ .$$

Proof. Since $\mathsf{Self\text{-}Correct}$, on input Q, makes only $2\lambda|Q|$ queries to P^*, this lemma follows directly from the construction of $\mathsf{Self\text{-}Correct}$ and the $\kappa'_{\max}$-wise no-signaling property of P^*. □

Next, we observe that $\mathsf{Self\text{-}Correct}$ outputs a statement that is indistinguishable from the one by P^*.

Lemma 3 (Statement Indistinguishability of $\mathsf{Self\text{-}Correct}$)**.** *Let $\kappa_{\max}$ be any polynomial, $\{C_\lambda\}_{\lambda\in\mathbb{N}}$ be any circuit family, and P^* be any $\kappa_{\max}$-wise no-signaling cheating prover. Then, two distributions,*

$$\left\{(C_\lambda, \boldsymbol{x}, \boldsymbol{y}) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, \emptyset)\right\}_{\lambda\in\mathbb{N}}$$

and

$$\left\{(C_\lambda, \boldsymbol{x}, \boldsymbol{y}) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \emptyset)\right\}_{\lambda\in\mathbb{N}} \ ,$$

are computationally indistinguishable (where $\emptyset$ is the empty set).

Proof. This lemma follows directly from the construction of $\mathsf{Self\text{-}Correct}$ since the output of $\mathsf{Self\text{-}Correct}$ is identical with that of P^* when $Q = \emptyset$. □

6.3 Key Properties of $\mathsf{Self\text{-}Correct}$

In this subsection, we give three key lemmas on $\mathsf{Self\text{-}Correct}$, which roughly say that when $\mathsf{Self\text{-}Correct}$ is applied on a cheating prover that convinces the relaxed verifier with overwhelming probability, it produces a proof that passes Linearity Test, Tensor-Product Test, SAT Test *on any points* with overwhelming probability.

Lemma 4 (Linearity of Self-Corrected Proof)**.** *Let $\mathbb{V} = (\mathbb{V}_0, \mathbb{V}_1)$ be the relaxed PCP verifier in Section 5, $\{C_\lambda\}_{\lambda\in\mathbb{N}}$ be any circuit family, and $\kappa_{\max}$ be any polynomial such that $\kappa_{\max}(\lambda) \ge \kappa_V(\lambda) = 2\lambda(5\lambda+3)$, where κ_V is the query complexity of (P, V).*

Then, for any $\kappa_{\max}$-wise no-signaling cheating prover P^*, if it holds

$$\Pr\left[\mathbb{V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*) = 1 \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q) \end{array}\right] \ge 1 - \mathsf{negl}(\lambda) \tag{6.1}$$

for infinitely many $\lambda \in \mathbb{N}$ (let Λ be the set of those λ's), there exists a negligible function negl such that for every sufficiently large $\lambda \in \Lambda$ and every $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{F}^N$ (resp. $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{F}^{N^2}$), it holds

$$\Pr\left[\tilde{\pi}(\boldsymbol{u}) + \tilde{\pi}(\boldsymbol{v}) = \tilde{\pi}(\boldsymbol{u}+\boldsymbol{v}) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u}+\boldsymbol{v}\})\right] \ge 1 - \mathsf{negl}(\lambda) \ . \tag{6.2}$$

Lemma 5 (Tensor-Product Consistency of Self-Corrected Proof)**.** *Let $\mathbb{V} = (\mathbb{V}_0, \mathbb{V}_1)$ be the relaxed PCP verifier in Section 5, $\{C_\lambda\}_{\lambda\in\mathbb{N}}$ be any circuit family, and $\kappa_{\max}$ be any polynomial such that $\kappa_{\max}(\lambda) \geq 2\lambda(8\lambda+3)$.*

Then, for any $\kappa_{\max}$-wise no-signaling cheating prover P^*, if it holds

$$\Pr\left[\mathbb{V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*) = 1 \;\middle|\; \begin{array}{l}(Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda)\\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q)\end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \tag{6.3}$$

for infinitely many $\lambda \in \mathbb{N}$ (let Λ be the set of those λ's), there exists a negligible function negl *such that for every sufficiently large $\lambda \in \Lambda$ and every $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{F}^N$, it holds*

$$\Pr\left[\tilde{\pi}_f(\boldsymbol{u})\tilde{\pi}_f(\boldsymbol{v}) = \tilde{\pi}_g(\boldsymbol{u}\otimes\boldsymbol{v}) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u}\otimes\boldsymbol{v}\})\right] \geq 1 - \mathsf{negl}(\lambda) \enspace . \tag{6.4}$$

Lemma 6 (SAT Consistency of Self-Corrected Proof)**.** *Let $\mathbb{V} = (\mathbb{V}_0, \mathbb{V}_1)$ be the relaxed PCP verifier in Section 5, $\{C_\lambda\}_{\lambda\in\mathbb{N}}$ be any circuit family, and $\kappa_{\max}$ be any polynomial such that $\kappa_{\max}(\lambda) \geq \kappa_V(\lambda) = 2\lambda(5\lambda+3)$, where κ_V is the query complexity of (P, V).*

Then, for any $\kappa_{\max}$-wise no-signaling cheating prover P^*, if it holds

$$\Pr\left[\mathbb{V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*) = 1 \;\middle|\; \begin{array}{l}(Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda)\\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q)\end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \tag{6.5}$$

for infinitely many $\lambda \in \mathbb{N}$ (let Λ be the set of those λ's), there exists a negligible function negl *such that for every sufficiently large $\lambda \in \Lambda$ and every $\boldsymbol{\sigma} \in \mathbb{F}^M$, it holds*

$$\Pr\left[\tilde{\pi}_f(\boldsymbol{\psi}_{\boldsymbol{\sigma}}) + \tilde{\pi}_g(\boldsymbol{\psi}'_{\boldsymbol{\sigma}}) = c_{\boldsymbol{\sigma}} \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{\psi}_{\boldsymbol{\sigma}}, \boldsymbol{\psi}'_{\boldsymbol{\sigma}}\})\right] \geq 1 - \mathsf{negl}(\lambda) \enspace , \tag{6.6}$$

where $c_{\boldsymbol{\sigma}}, \boldsymbol{\psi}_{\boldsymbol{\sigma}}, \boldsymbol{\psi}'_{\boldsymbol{\sigma}}$ are defined as follows: let $\Psi = \{\Psi_i(\boldsymbol{z}) = c_i\}_{i\in[M]}$ be the system of equations that is obtained from the circuit C_λ as described in Section 4, and let

$$\Psi_{\boldsymbol{\sigma}}(\boldsymbol{z}) \coloneqq \sum_{i\in[M]} \sigma_i \Psi_i(\boldsymbol{z}) \qquad \text{and} \qquad c_{\boldsymbol{\sigma}} \coloneqq \sum_{i\in[M]} \sigma_i c_i \enspace ;$$

then, $\boldsymbol{\psi}_{\boldsymbol{\sigma}}, \boldsymbol{\psi}'_{\boldsymbol{\sigma}}$ are the coefficient of $\Psi_{\boldsymbol{\sigma}}(\boldsymbol{z})$ such that

$$\Psi_{\boldsymbol{\sigma}}(\boldsymbol{z}) = \langle \boldsymbol{\psi}_{\boldsymbol{\sigma}}, \boldsymbol{z}\rangle + \langle \boldsymbol{\psi}'_{\boldsymbol{\sigma}}, \boldsymbol{z}\otimes\boldsymbol{z}\rangle \enspace .$$

We prove these lemmas in Sections 6.6, 6.7, and 6.8.

6.4 Corollaries of Lemma 4

Before proving the three key lemmas in the previous subsection, we observe that we can obtain several useful corollaries from the linearity of the self-corrected proof (Lemma 4).

First, we obtain the following basic lemma from Lemma 4.

Lemma 7. *Let $\mathbb{V} = (\mathbb{V}_0, \mathbb{V}_1)$ be the relaxed PCP verifier in Section 5, $\{C_\lambda\}_{\lambda\in\mathbb{N}}$ be any circuit family, and $\kappa_{\max}$ be any polynomial such that $\kappa_{\max}(\lambda) \geq \kappa_V(\lambda) = 2\lambda(5\lambda+3)$, where κ_V is the query complexity of (P, V).*

Then, for any $\kappa_{\max}$-wise no-signaling cheating prover P^*, if it holds

$$\Pr\left[\mathbb{V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*) = 1 \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \tag{6.7}$$

for infinitely many $\lambda \in \mathbb{N}$ (let Λ be the set of those λ's), there exists a negligible function negl *such that for every sufficiently large $\lambda \in \Lambda$ and every $\boldsymbol{v} \in \mathbb{F}^N$ (resp. $\boldsymbol{v} \in \mathbb{F}^{N^2}$), it holds*

$$\Pr\left[\tilde{\pi}(\mathbf{0}) = 0 \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\mathbf{0}\})\right] \geq 1 - \mathsf{negl}(\lambda) \tag{6.8}$$

and

$$\Pr\left[\tilde{\pi}(-\boldsymbol{v}) = -\tilde{\pi}(\boldsymbol{v}) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{v}, -\boldsymbol{v}\})\right] \geq 1 - \mathsf{negl}(\lambda) \enspace , \tag{6.9}$$

where $\mathbf{0} = (0, \dots, 0) \in \mathbb{F}^N$ (resp. $\mathbf{0} = (0, \dots, 0) \in \mathbb{F}^{N^2}$).

Proof. Fix any $\{C_\lambda\}_{\lambda \in \mathbb{N}}$, $\kappa_{\max}$, and P^*, and assume that Equation (6.7) holds for infinitely many $\lambda \in \mathbb{N}$. Let Λ be the set of those λ's, and fix any sufficiently large $\lambda \in \Lambda$. Our goal is to show Equations (6.8) and (6.9).

First, we obtain Equation (6.8) by first obtaining

$$\Pr\left[\tilde{\pi}(\boldsymbol{v} + \mathbf{0}) = \tilde{\pi}(\boldsymbol{v}) + \tilde{\pi}(\mathbf{0}) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{v}, \mathbf{0}\})\right] \geq 1 - \mathsf{negl}(\lambda)$$

for any $\boldsymbol{v} \in \mathbb{F}^N$ (resp. $\boldsymbol{v} \in \mathbb{F}^{N^2}$) from Lemma 4 and then use the no-signaling property of Self-Correct (Lemma 2).

Next, we obtain Equation (6.9) by first obtaining

$$\Pr\left[\tilde{\pi}(\boldsymbol{v}) + \tilde{\pi}(-\boldsymbol{v}) = \tilde{\pi}(\mathbf{0}) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{v}, -\boldsymbol{v}, \mathbf{0}\})\right] \geq 1 - \mathsf{negl}(\lambda)$$

from Lemma 4 and then use Equation (6.8) and the no-signaling property of Self-Correct (Lemma 2). □

Next, we observe that Lemma 4 can be generalized as follows.

Lemma 8 (Linearity of Self-Corrected Proof, scalar multiplication)**.** *Let $\mathbb{V} = (\mathbb{V}_0, \mathbb{V}_1)$ be the relaxed PCP verifier in Section 5, $\{C_\lambda\}_{\lambda \in \mathbb{N}}$ be any circuit family, and $\kappa_{\max}$ be any polynomial such that $\kappa_{\max}(\lambda) \geq \kappa_V(\lambda) = 2\lambda(5\lambda + 3)$, where κ_V is the query complexity of (P, V).*

Then, for any $\kappa_{\max}$-wise no-signaling cheating prover P^*, if it holds

$$\Pr\left[\mathbb{V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*) = 1 \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \tag{6.10}$$

for infinitely many $\lambda \in \mathbb{N}$ (let Λ be the set of those λ's), there exists a negligible function negl *such that for every sufficiently large $\lambda \in \Lambda$, every $\boldsymbol{v} \in \mathbb{F}^N$ (resp. $\boldsymbol{v} \in \mathbb{F}^{N^2}$), and every $k \in \{3 \dots, |\mathbb{F}| - 1\}$, it holds*

$$\Pr\left[k\tilde{\pi}(\boldsymbol{v}) = \tilde{\pi}(k\boldsymbol{v}) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{v}, k\boldsymbol{v}\})\right] \geq 1 - \mathsf{negl}(\lambda) \enspace . \tag{6.11}$$

Proof. Fix any $\{C_\lambda\}_{\lambda\in\mathbb{N}}$, $\kappa_{\max}$, and P^*, and assume that Equation (6.10) holds for infinitely many $\lambda \in \mathbb{N}$. Let Λ be the set of those λ's, and fix any sufficiently large $\lambda \in \Lambda$. Our goal is to show Equation (6.11).

Fix any $\boldsymbol{v} \in \mathbb{F}^N$ (resp. $\boldsymbol{v} \in \mathbb{F}^{N^2}$), and let

$$p(k) := \Pr\left[k\tilde{\pi}(\boldsymbol{v}) = \tilde{\pi}(k\boldsymbol{v}) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{v}, k\boldsymbol{v}\})\right]$$

for $k \in \{2, \ldots, k_{\max}\}$, where $k_{\max} := |\mathbb{F}| - 1$. (Note that we have $\log k_{\max} \le \mathsf{poly}(\lambda)$.) In this notation, our goal is to show $p(k) \ge 1 - \mathsf{negl}(\lambda)$ for every $k \in \{3, \ldots, k_{\max}\}$. Assume, for simplicity, that k is a power of 2 (the general case can be handled similarly). Then, observe that if we have

$$\tilde{\pi}(k\boldsymbol{v}) = 2\tilde{\pi}(k\boldsymbol{v}/2) \bigwedge \tilde{\pi}(k\boldsymbol{v}/2) = k\tilde{\pi}(\boldsymbol{v})/2 \ ,$$

then we have $k\tilde{\pi}(\boldsymbol{v}) = \tilde{\pi}(k\boldsymbol{v})$. In addition, observe that from the linearity of the self-corrected proof (Lemma 4), we have

$$\Pr\left[\tilde{\pi}(k\boldsymbol{v}) = 2\tilde{\pi}(k\boldsymbol{v}/2) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{k\boldsymbol{v}/2, k\boldsymbol{v}\})\right] \ge 1 - \mathsf{negl}(\lambda) \ .$$

By combining the above observations and the no-signaling property of $\mathsf{Self\text{-}Correct}$ (Lemma 2), we obtain

$$p(k) \ge p(k/2) - \mathsf{negl}(\lambda) \ .$$

Now, since we have $p(2) \ge 1 - \mathsf{negl}(\lambda)$ from the linearity of the self-corrected proof (Lemma 4), we have

$$p(k) \ge 1 - \log k \cdot \mathsf{negl}(\lambda) \ge 1 - \mathsf{negl}(\lambda)$$

as desired. This concludes the proof of Lemma 8. □

Finally, we notice that Lemma 4 can also be generalized as follows.

Lemma 9 (Linearity of Self-Corrected Proof, more than two points)**.** *Let $\mathbb{V} = (\mathbb{V}_0, \mathbb{V}_1)$ be the relaxed PCP verifier in Section 5, $\{C_\lambda\}_{\lambda\in\mathbb{N}}$ be any circuit family, and $\kappa_{\max}$ be any polynomial such that $\kappa_{\max}(\lambda) \ge \kappa_V(\lambda) = 2\lambda(5\lambda + 3)$, where κ_V is the query complexity of (P, V).*

Then, for any $\kappa_{\max}$-wise no-signaling cheating prover P^*, if it holds

$$\Pr\left[\mathbb{V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*) = 1 \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q) \end{array}\right] \ge 1 - \mathsf{negl}(\lambda)$$

for infinitely many $\lambda \in \mathbb{N}$ (let Λ be the set of those λ's), there exists a negligible function negl such that for every sufficiently large $\lambda \in \Lambda$ and every $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k \in \mathbb{F}^N$ (resp. $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k \in \mathbb{F}^{N^2}$), where $k \in \{3, \ldots, (\kappa_{\max}/2\lambda) - 2\}$, it holds

$$\Pr\left[\begin{array}{l} \tilde{\pi}(\boldsymbol{v}_1) + \cdots + \tilde{\pi}(\boldsymbol{v}_k) \\ = \tilde{\pi}(\boldsymbol{v}_1 + \cdots + \boldsymbol{v}_k) \end{array} \;\middle|\; \begin{array}{l} (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, Q) \\ \quad \textit{where } Q = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k, \boldsymbol{v}_1 + \cdots + \boldsymbol{v}_k\} \end{array}\right] \ge 1 - \mathsf{negl}(\lambda) \ .$$

The proof is omitted.

6.5 Preliminary Observation

The rest of this section is devoted for proving the three key lemmas in Section 6.3. Since we only use the statements of these lemmas in the rest of this paper, the readers who believe these lemmas can skip the rest of this section.

In this subsection, we make a useful immediate observation about $\mathsf{Self\text{-}Correct}$. Specifically, we observe that if $\mathsf{Self\text{-}Correct}$ is applied on a no-signaling cheating prover that convinces the relaxed verifier with overwhelming probability, it produces a proof that passes each of Linearity Test, Tensor-Product Test, and SAT Test on random points with overwhelming probability even when these tests are done individually. We remind the readers that the relaxed verifier accepts a PCP proof even when the proof fails to pass a small number of tests, which means that we can only hope for that the self-corrected proof passes Linearity Test etc. except for a small number of trials.

Observation 1. Let $\mathbb{V} = (\mathbb{V}_0, \mathbb{V}_1)$ be the relaxed PCP verifier in Section 5, $\{C_\lambda\}_{\lambda\in\mathbb{N}}$ be any circuit family, and $\kappa_{\max}$ be any polynomial such that $\kappa_{\max}(\lambda) \geq \kappa_V(\lambda) = 2\lambda(5\lambda + 3)$, where κ_V is the query complexity of (P, V).

Then, for any $\kappa_{\max}$-wise no-signaling cheating prover P^*, if it holds

$$\Pr\left[\mathbb{V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*) = 1 \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

for infinitely many $\lambda \in \mathbb{N}$ (let Λ be the set of those λ's), there exists a negligible function negl such that for every sufficiently large $\lambda \in \Lambda$, each of the following holds.

- **Linearity Test.**

$$\Pr\left[|I_{\mathrm{linear}}| \geq \lambda - \mu \;\middle|\; \begin{array}{l} \boldsymbol{r}_i, \boldsymbol{s}_i \leftarrow \mathbb{F}^N \text{ for } \forall i \in [\lambda] \\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q) \text{ where } Q = \{\boldsymbol{r}_i, \boldsymbol{s}_i, \boldsymbol{r}_i + \boldsymbol{s}_i\}_{i\in[\lambda]} \end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \;\;,$$

  where

$$I_{\mathrm{linear}} := \left\{ i \text{ s.t. } \pi^*_f(\boldsymbol{r}_i) + \pi^*_f(\boldsymbol{s}_i) = \pi^*_f(\boldsymbol{r}_i + \boldsymbol{s}_i) \right\} \;\;.$$

  The same holds when π^*_f is replaced with π^*_g.

- **Tensor-Product Test.**

$$\Pr\left[|I_{\mathrm{Tensor}}| \geq \lambda - \mu \;\middle|\; \begin{array}{l} \boldsymbol{r}_i, \boldsymbol{s}_i \leftarrow \mathbb{F}^N \text{ for } \forall i \in [\lambda] \\ (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, Q) \\ \quad \text{where } Q = \{\boldsymbol{r}_i, \boldsymbol{s}_i, \boldsymbol{r}_i \otimes \boldsymbol{s}_i\}_{i\in[\lambda]} \end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \;\;,$$

  where

$$I_{\mathrm{Tensor}} := \left\{ i \text{ s.t. } \tilde{\pi}_f(\boldsymbol{r}_i)\tilde{\pi}_f(\boldsymbol{s}_i) = \tilde{\pi}_g(\boldsymbol{r}_i \otimes \boldsymbol{s}_i) \right\} \;\;.$$

- **SAT Test.**

$$\Pr\left[|I_{\mathrm{SAT}}| \geq \lambda - \mu \;\middle|\; \begin{array}{l} \boldsymbol{\sigma}_i \leftarrow \mathbb{F}^M \text{ for } \forall i \in [\lambda] \\ (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, Q) \\ \quad \text{where } Q = \{\boldsymbol{\psi}_{\boldsymbol{\sigma}_i}, \boldsymbol{\psi}'_{\boldsymbol{\sigma}_i}\}_{i\in[\lambda]} \end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \;\;,$$

  where $\boldsymbol{\psi}_{\boldsymbol{\sigma}_i} \in \mathbb{F}^N, \boldsymbol{\psi}'_{\boldsymbol{\sigma}_i} \in \mathbb{F}^{N^2}$ are defined as in Equation (4.2), and

$$I_{\mathrm{SAT}} := \left\{ i \text{ s.t. } \tilde{\pi}_f(\boldsymbol{\psi}_{\boldsymbol{\sigma}_i}) + \tilde{\pi}_g(\boldsymbol{\psi}_{\boldsymbol{\sigma}_i}) = c_{\boldsymbol{\sigma}_i} \right\} \;\;.$$

$\diamond$

It is easy to see that this observation follows from the definition of the relaxed verifier and the no-signaling property of P^*.

6.6 Proof of Lemma 4 (Linearity of Self-Corrected Proof)

In this subsection, we prove Lemma 4, which says that the self-corrected proof passes Linearity Test on any points. As mentioned in the technical overview (Section 3.3.3), our analysis is an extension of a previous analysis of Linearity Test in the standard PCP setting. (In particular, our analysis is based on the analysis that is described in a textbook by Goldreich [Gol17], which follows the idea of Blum, Luby, and Rubinfeld [BLR93].)

Proof of Lemma 4. Fix any $\{C_\lambda\}_{\lambda\in\mathbb{N}}$, $\kappa_{\max}$, and P^*, and assume that Equation (6.1) holds for infinitely many $\lambda \in \mathbb{N}$. Let Λ be the set of those λ's, and fix any sufficiently large $\lambda \in \Lambda$. Our goal is to show Equation (6.2). In the following, we only consider the case of $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{F}^N$. (The case of $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{F}^{N^2}$ can be proven identically.)

At a high level, the proof proceeds as follows. From the construction, $\mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u} + \boldsymbol{v}\})$ defines the self-corrected values $\tilde{\pi}(\boldsymbol{u}), \tilde{\pi}(\boldsymbol{v}), \tilde{\pi}(\boldsymbol{u} + \boldsymbol{v})$ by using three independent sets of randomness (where each set consists of λ random points in $\mathbb{F}^N$). First, we introduce a mental experiment where, in addition to $\tilde{\pi}(\boldsymbol{u}), \tilde{\pi}(\boldsymbol{v}), \tilde{\pi}(\boldsymbol{u} + \boldsymbol{v})$, alternative self-corrected values $\tilde{\rho}(\boldsymbol{u}), \tilde{\rho}(\boldsymbol{v}), \tilde{\rho}(\boldsymbol{u} + \boldsymbol{v})$ are defined by $\mathsf{Self\text{-}Correct}$, where those alternative self-corrected values are defined by using three mutually dependent sets of randomness. Then, we proceed in the following two steps.

1. First, we show that the self-correction procedure is "consistent" in the sense that we have $\tilde{\pi}(\boldsymbol{t}) = \tilde{\rho}(\boldsymbol{t})$ for every $\boldsymbol{t} \in \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u}+\boldsymbol{v}\}$. Roughly, we show this consistency by using the fact that, although the three sets of randomness that are used for $\tilde{\rho}(\boldsymbol{u}), \tilde{\rho}(\boldsymbol{v}), \tilde{\rho}(\boldsymbol{u} + \boldsymbol{v})$ are mutually dependent, each of these three sets is, when viewed individually, uniformly distributed.

2. Second, we show that the alternative self-corrected values satisfy linearity, i.e., they satisfy $\tilde{\rho}(\boldsymbol{u}) + \tilde{\rho}(\boldsymbol{v}) = \tilde{\rho}(\boldsymbol{u} + \boldsymbol{v})$. Roughly, we show this linearity by using the fact that the alternative self-corrected values are generated with mutually dependent sets of randomness.

By combining there two steps, we can obtain $\tilde{\pi}(\boldsymbol{u}) + \tilde{\pi}(\boldsymbol{v}) = \tilde{\rho}(\boldsymbol{u}) + \tilde{\rho}(\boldsymbol{v}) = \tilde{\rho}(\boldsymbol{u}+\boldsymbol{v}) = \tilde{\pi}(\boldsymbol{u}+\boldsymbol{v})$ as desired.

Formally, we first observe that the no-signaling property of P^* and the construction of $\mathsf{Self\text{-}Correct}$ guarantees that, to prove this lemma, it suffices to show that Equation (6.2) holds when $\tilde{\pi}$ are sampled in the following way. (A notable difference from $\mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u} + \boldsymbol{v}\})$ is highlighted by red.)

1. Choose 4λ random points

$$\boldsymbol{r}_{\boldsymbol{u},1}, \ldots, \boldsymbol{r}_{\boldsymbol{u},\lambda}, \boldsymbol{r}_{\boldsymbol{v},1}, \ldots, \boldsymbol{r}_{\boldsymbol{v},\lambda}, \boldsymbol{r}_{\boldsymbol{u}+\boldsymbol{v},1}, \ldots, \boldsymbol{r}_{\boldsymbol{u}+\boldsymbol{v},\lambda} \in \mathbb{F}^N, \text{ and}$$
$$\boldsymbol{s}_1, \ldots, \boldsymbol{s}_\lambda \in \mathbb{F}^N$$

2. Run $(\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q \cup Q')$, where

$$Q = \{\boldsymbol{r}_{\boldsymbol{u},i}, \boldsymbol{u} + \boldsymbol{r}_{\boldsymbol{u},i}, \boldsymbol{r}_{\boldsymbol{v},i}, \boldsymbol{v} + \boldsymbol{r}_{\boldsymbol{v},i}, \boldsymbol{r}_{\boldsymbol{u}+\boldsymbol{v},i}, \boldsymbol{u} + \boldsymbol{v} + \boldsymbol{r}_{\boldsymbol{u}+\boldsymbol{v},i}\}_{i\in[\lambda]} \ ,$$
$$Q' = \{\boldsymbol{s}_i, \boldsymbol{u} + \boldsymbol{s}_i, \boldsymbol{s}_i - \boldsymbol{v}\}_{i\in[\lambda]} \ .$$

3. For each $i \in [\lambda]$, define a function $\tilde{\pi}^{(i)} : \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u}+\boldsymbol{v}\} \to \mathbb{F}$ by

$$\tilde{\pi}^{(i)}(\boldsymbol{t}) := \pi^*(\boldsymbol{t} + \boldsymbol{r}_{\boldsymbol{t},i}) - \pi^*(\boldsymbol{r}_{\boldsymbol{t},i}) \text{ for } \forall \boldsymbol{t} \in \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u}+\boldsymbol{v}\}$$

   and a function $\tilde{\rho}^{(i)} : \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u}+\boldsymbol{v}\} \to \mathbb{F}$ by

$$\begin{aligned} \tilde{\rho}^{(i)}(\boldsymbol{u}) &:= \pi^*_f(\boldsymbol{u} + \boldsymbol{s}_i) - \pi^*_f(\boldsymbol{s}_i) \\ \tilde{\rho}^{(i)}(\boldsymbol{v}) &:= \pi^*_f(\boldsymbol{s}_i) - \pi^*_f(\boldsymbol{s}_i - \boldsymbol{v}) \\ \tilde{\rho}^{(i)}(\boldsymbol{u}+\boldsymbol{v}) &:= \pi^*_f(\boldsymbol{u} + \boldsymbol{s}_i) - \pi^*_f(\boldsymbol{s}_i - \boldsymbol{v}) \ . \end{aligned}$$

4. Define a function $\tilde{\pi} : \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u}+\boldsymbol{v}\} \to \mathbb{F}$ by

$$\tilde{\pi}(\boldsymbol{t}) := \text{majority}\left(\tilde{\pi}^{(1)}(\boldsymbol{t}), \ldots, \tilde{\pi}^{(\lambda)}(\boldsymbol{t})\right) \text{ for } \forall \boldsymbol{t} \in \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u}+\boldsymbol{v}\}$$

   and a function $\tilde{\rho} : \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u}+\boldsymbol{v}\} \to \mathbb{F}$ by

$$\tilde{\rho}(\boldsymbol{t}) := \text{majority}\left(\tilde{\rho}^{(1)}(\boldsymbol{t}), \ldots, \tilde{\rho}^{(\lambda)}(\boldsymbol{t})\right) \text{ for } \forall \boldsymbol{t} \in \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u}+\boldsymbol{v}\} \ .$$

5. Output $(\tilde{\pi}, \tilde{\rho})$.

From the above observation, we can obtain Equation (6.2) by showing

$$\Pr_{\tilde{\pi},\tilde{\rho}}[\tilde{\pi}(\boldsymbol{u}) + \tilde{\pi}(\boldsymbol{v}) = \tilde{\pi}(\boldsymbol{u}+\boldsymbol{v})] \geq 1 - \mathsf{negl}(\lambda) \ , \tag{6.12}$$

where for any event E, we use $\Pr_{\tilde{\pi},\tilde{\rho}}[E]$ as a shorthand for the probability that the event E occurs when $\tilde{\pi}, \tilde{\rho}$ are chosen as above (along with $\boldsymbol{x}, \boldsymbol{y}$).

Now, we show Equation (6.12) by using the following two claims.

Claim 3 (Consistency of Self Correction)**.** *For every $\boldsymbol{t} \in \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u}+\boldsymbol{v}\}$,*

$$\Pr_{\tilde{\pi},\tilde{\rho}}[\tilde{\pi}(\boldsymbol{t}) = \tilde{\rho}(\boldsymbol{t})] \geq 1 - \mathsf{negl}(\lambda) \ . \tag{6.13}$$

Claim 4 (Existence of Strong Majority)**.** *For every $\boldsymbol{t} \in \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u}+\boldsymbol{v}\}$,*

$$\Pr_{\tilde{\pi},\tilde{\rho}}\left[|\{i \text{ s.t. } \tilde{\rho}^{(i)}(\boldsymbol{t}) = \tilde{\rho}(\boldsymbol{t})\}| \geq \lambda - 20\mu\right] \geq 1 - \mathsf{negl}(\lambda) \ .$$

Before proving Claim 3 and Claim 4, we show that Equation (6.12) indeed follows from these two claims. From Claim 3 and the union bound, we can show Equation (6.12) by showing

$$\Pr_{\tilde{\pi},\tilde{\rho}}[\tilde{\rho}(\boldsymbol{u}) + \tilde{\rho}(\boldsymbol{v}) = \tilde{\rho}(\boldsymbol{u}+\boldsymbol{v})] \geq 1 - \mathsf{negl}(\lambda) \ . \tag{6.14}$$

Thus, we focus on showing Equation (6.14). First, from Claim 4 and the union bound, we have

$$\Pr_{\tilde{\pi},\tilde{\rho}}\left[\begin{array}{l} |\{i \text{ s.t. } \tilde{\rho}^{(i)}(\boldsymbol{u}) = \tilde{\rho}(\boldsymbol{u})\}| \geq \lambda - 20\mu \\ \wedge\ |\{i \text{ s.t. } \tilde{\rho}^{(i)}(\boldsymbol{v}) = \tilde{\rho}(\boldsymbol{v})\}| \geq \lambda - 20\mu \\ \wedge\ |\{i \text{ s.t. } \tilde{\rho}^{(i)}(\boldsymbol{u}+\boldsymbol{v}) = \tilde{\rho}(\boldsymbol{u}+\boldsymbol{v})\}| \geq \lambda - 20\mu \end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \ . \tag{6.15}$$

Then, since we have $3 \cdot 20\mu < \lambda$ for every sufficiently large λ, from Equation (6.15) and the pigeonhole principle we have

$$\Pr_{\tilde{\pi},\tilde{\rho}}\left[\exists i^* \in [\lambda] \text{ s.t. } \begin{array}{l} \tilde{\rho}^{(i^*)}(\boldsymbol{u}) = \tilde{\rho}(\boldsymbol{u}) \\ \wedge\, \tilde{\rho}^{(i^*)}(\boldsymbol{v}) = \tilde{\rho}(\boldsymbol{v}) \\ \wedge\, \tilde{\rho}^{(i^*)}(\boldsymbol{u}+\boldsymbol{v}) = \tilde{\rho}(\boldsymbol{u}+\boldsymbol{v}) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \ . \tag{6.16}$$

Then, observe that for every $i^* \in [\lambda]$ such that

$$\tilde{\rho}^{(i^*)}(\boldsymbol{u}) = \tilde{\rho}(\boldsymbol{u}) \qquad \wedge \qquad \tilde{\rho}^{(i^*)}(\boldsymbol{v}) = \tilde{\rho}(\boldsymbol{v}) \qquad \wedge \qquad \tilde{\rho}^{(i^*)}(\boldsymbol{u}+\boldsymbol{v}) = \tilde{\rho}(\boldsymbol{u}+\boldsymbol{v}) \ ,$$

we have

$$\begin{aligned} \tilde{\rho}(\boldsymbol{u}) &= \pi^*_f(\boldsymbol{u} + \boldsymbol{s}_{i^*}) - \pi^*_f(\boldsymbol{s}_{i^*}) \ , \\ \tilde{\rho}(\boldsymbol{v}) &= \pi^*_f(\boldsymbol{s}_{i^*}) - \pi^*_f(\boldsymbol{s}_{i^*} - \boldsymbol{v}) \ , \\ \tilde{\rho}(\boldsymbol{u}+\boldsymbol{v}) &= \pi^*_f(\boldsymbol{u} + \boldsymbol{s}_{i^*}) - \pi^*_f(\boldsymbol{s}_{i^*} - \boldsymbol{v}) \ , \end{aligned}$$

from the definition of $\tilde{\rho}^{(i^*)}$, and therefore have $\tilde{\rho}(\boldsymbol{u}) + \tilde{\rho}(\boldsymbol{v}) = \tilde{\rho}(\boldsymbol{u}+\boldsymbol{v})$. By combining this observation with Equation (6.16), we obtain Equation (6.14).

Finally, to conclude the proof of Lemma 4, we prove Claim 3 and Claim 4. We prove these two claims by proving the following claim, which implies both of Claim 3 and Claim 4.

Claim 5. *For every $\boldsymbol{t} \in \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u}+\boldsymbol{v}\}$,*

$$\Pr_{\tilde{\pi},\tilde{\rho}}\left[\mathit{Strong\text{-}Majority}_{\boldsymbol{t}}\right] \geq 1 - \mathsf{negl}(\lambda) \ ,$$

where $\mathsf{Strong\text{-}Majority}_{\boldsymbol{t}}$ is the event that there exists $a_{\boldsymbol{t}} \in \mathbb{F}$ such that

$$\left|\{i \text{ s.t. } \tilde{\pi}^{(i)}(\boldsymbol{t}) = \tilde{\rho}^{(i)}(\boldsymbol{t}) = a_{\boldsymbol{t}}\}\right| \geq \lambda - 20\mu \ .$$

(To see that Claim 5 indeed implies Claim 3 and Claim 4, observe that when $\mathsf{Strong\text{-}Majority}_{\boldsymbol{t}}$ occurs, we have $\tilde{\pi}(\boldsymbol{t}) = \tilde{\rho}(\boldsymbol{t}) = a_{\boldsymbol{t}}$ since we have $\lambda - 20\mu > \lambda/2$ for every sufficiently large λ.) Hence, to prove Claim 3 and Claim 4, it remains to prove Claim 5.

Proof of Claim 5. An important observation is that, during the sampling of $\tilde{\pi}, \tilde{\rho}$ (as per the description at the beginning of the proof of Lemma 4), each of

- $\boldsymbol{r}_{\boldsymbol{u},\lambda}, \ldots, \boldsymbol{r}_{\boldsymbol{u},\lambda}, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_\lambda$
- $\boldsymbol{r}_{\boldsymbol{v},1}, \ldots, \boldsymbol{r}_{\boldsymbol{v},\lambda}, \boldsymbol{s}_1 - \boldsymbol{v}, \ldots, \boldsymbol{s}_\lambda - \boldsymbol{v}$
- $\boldsymbol{r}_{\boldsymbol{u}+\boldsymbol{v},1}, \ldots, \boldsymbol{r}_{\boldsymbol{u}+\boldsymbol{v},\lambda}, \boldsymbol{s}_1 - \boldsymbol{v}, \ldots, \boldsymbol{s}_\lambda - \boldsymbol{v}$

is a set of 2λ random points. This observation, combined with the no-signaling property of P^* and the definition of $\tilde{\rho}^{(i)}$, implies that, to prove Claim 5, it suffices to show the following simpler claim.

Claim 6. *For any $\boldsymbol{t} \in \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u}+\boldsymbol{v}\}$, we have*

$$\Pr_{\tilde{\pi}_t,\tilde{\rho}_t}\left[\mathit{Strong\text{-}Majority}\right] \geq 1 - \mathsf{negl}(\lambda) \ , \tag{6.17}$$

where the probability is taken over the following sampling of $\tilde{\pi}_{\boldsymbol{t}}, \tilde{\rho}_{\boldsymbol{t}}$.

1. *Choose* 2λ *random points* $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_\lambda, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_\lambda \in \mathbb{F}^N$.

2. *Run* $(\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q \cup Q')$, *where* $Q = \{\boldsymbol{r}_i, \boldsymbol{t} + \boldsymbol{r}_i\}_{i \in [\lambda]}$ *and* $Q' = \{\boldsymbol{s}_i, \boldsymbol{t} + \boldsymbol{s}_i\}_{i \in [\lambda]}$.

3. *For each* $i \in [\lambda]$, *define* $\tilde{\pi}_{\boldsymbol{t}}^{(i)} := \pi^*(\boldsymbol{t} + \boldsymbol{r}_i) - \pi^*(\boldsymbol{r}_i)$ *and* $\tilde{\rho}_{\boldsymbol{t}}^{(i)} := \pi^*(\boldsymbol{t} + \boldsymbol{s}_i) - \pi^*(\boldsymbol{s}_i)$.

4. *Output* $\tilde{\pi}_{\boldsymbol{t}} := \mathrm{majority}(\tilde{\pi}_{\boldsymbol{t}}^{(1)}, \ldots, \tilde{\pi}_{\boldsymbol{t}}^{(\lambda)})$ *and* $\tilde{\rho} := \mathrm{majority}(\tilde{\rho}_{\boldsymbol{t}}^{(1)}, \ldots, \tilde{\rho}_{\boldsymbol{t}}^{(\lambda)})$.

and Strong-Majority *is the event that there exists* $a_{\boldsymbol{t}} \in \mathbb{F}$ *such that*

$$\left|\{i \text{ s.t. } \tilde{\pi}_{\boldsymbol{t}}^{(i)} = \tilde{\rho}_{\boldsymbol{t}}^{(i)} = a_{\boldsymbol{t}}\}\right| \geq \lambda - 20\mu \ .$$

Remark 6. To see that Claim 6 indeed implies Claim 5, observe the following. Consider, for example, the case of $\boldsymbol{t} = \boldsymbol{v}$. Then, if we simplify the sampling of $\tilde{\pi}, \tilde{\rho}$ in Claim 5 by removing all the queries that are not used for defining $\tilde{\pi}(\boldsymbol{v}), \tilde{\rho}(\boldsymbol{v})$, we obtain the following sampling (note that this simplification does not non-negligibly increase the probability of $\mathsf{Strong\text{-}Majority}_{\boldsymbol{v}}$ occurring because of the no-signaling property of P^*).

1. Choose 2λ random points $\boldsymbol{r}_{\boldsymbol{v},1}, \ldots, \boldsymbol{r}_{\boldsymbol{v},\lambda}, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_\lambda \in \mathbb{F}^N$.

2. Run $(\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q \cup Q')$, where $Q = \{\boldsymbol{r}_{\boldsymbol{v},i}, \boldsymbol{v} + \boldsymbol{r}_{\boldsymbol{v},i}\}_{i \in [\lambda]}$ and $Q' = \{\boldsymbol{s}_i, \boldsymbol{s}_i - \boldsymbol{v}\}_{i \in [\lambda]}$.

3. For each $i \in [\lambda]$, define $\tilde{\pi}^{(i)}(\boldsymbol{v}) := \pi^*(\boldsymbol{v} + \boldsymbol{r}_{\boldsymbol{v},i}) - \pi^*(\boldsymbol{r}_{\boldsymbol{v},i})$ and $\tilde{\rho}^{(i)}(\boldsymbol{v}) := \pi^*(\boldsymbol{s}_i) - \pi^*(\boldsymbol{s}_i - \boldsymbol{v})$.

4. Output $\tilde{\pi}(\boldsymbol{v}) := \mathrm{majority}(\tilde{\pi}^{(1)}(\boldsymbol{v}), \ldots, \tilde{\pi}^{(\lambda)}(\boldsymbol{v}))$ and $\tilde{\rho}(\boldsymbol{v}) := \mathrm{majority}(\tilde{\rho}^{(1)}(\boldsymbol{v}), \ldots, \tilde{\rho}^{(\lambda)}(\boldsymbol{v}))$.

Now, since in this sampling, we have $\tilde{\rho}^{(i)}(\boldsymbol{v}) = \pi^*(\boldsymbol{v} + (\boldsymbol{s}_i - \boldsymbol{v})) - \pi^*(\boldsymbol{s}_i - \boldsymbol{v})$ and that the set $\{\boldsymbol{r}_{\boldsymbol{v},1}, \ldots, \boldsymbol{r}_{\boldsymbol{v},\lambda}, \boldsymbol{s}_1 - \boldsymbol{v}, \ldots, \boldsymbol{s}_\lambda - \boldsymbol{v}\}$ is a set of 2λ random points, this sampling is equivalent to the sampling of $\tilde{\pi}, \tilde{\rho}$ in Claim 6. ◇

Thus, what remains to do is to prove Claim 6.

Proof of Claim 6. To show Equation (6.17), we use the following two sub-claims.

Sub-Claim 1. *We have*

$$\Pr_{\tilde{\pi}_t, \tilde{\rho}_t}\left[|I_{\text{Good-pair}}| \geq \lambda - 2\mu\right] \geq 1 - \mathsf{negl}(\lambda) \ , \tag{6.18}$$

where $I_{\text{Good-pair}} := \{i \text{ s.t. } \tilde{\pi}_{\boldsymbol{t}}^{(i)} = \tilde{\rho}_{\boldsymbol{t}}^{(i)}\}$.

Sub-Claim 2. *We have*

$$\Pr_{\tilde{\pi}_t, \tilde{\rho}_t}\left[I_{\text{Good-pair}} \geq \lambda - 2\mu \;\middle|\; \neg \mathsf{Strong\text{-}Majority}\right] \leq \mathsf{negl}(\lambda) \ ,$$

where $I_{\text{Good-pair}}$ *is defined as in Sub-Claim 1.*

First, Equation (6.17) follows from these two sub-claims since we have

$$\begin{aligned}
&\Pr_{\tilde{\pi}_t,\tilde{\rho}_t}[\textsf{Strong-Majority}] \\
&\geq \Pr_{\tilde{\pi}_t,\tilde{\rho}_t}\left[\textsf{Strong-Majority} \wedge I_{\text{Good-pair}} \geq \lambda - 2\mu\right] \\
&= \Pr_{\tilde{\pi}_t,\tilde{\rho}_t}\left[I_{\text{Good-pair}} \geq \lambda - 2\mu\right] - \Pr_{\tilde{\pi}_t,\tilde{\rho}_t}\left[\neg\textsf{Strong-Majority} \wedge I_{\text{Good-pair}} \geq \lambda - 2\mu\right] \\
&\geq \Pr_{\tilde{\pi}_t,\tilde{\rho}_t}\left[I_{\text{Good-pair}} \geq \lambda - 2\mu\right] - \Pr_{\tilde{\pi}_t,\tilde{\rho}_t}\left[I_{\text{Good-pair}} \geq \lambda - 2\mu \mid \neg\textsf{Strong-Majority}\right] \\
&= 1 - \mathsf{negl}(\lambda) \ ,
\end{aligned}$$

where we use the two sub-claims in the last equation. Thus, what remains to do is to prove the two sub-claims.

Proof of Sub-Claim 1. At a high level, the proof proceeds as follows. Observe that for every $i \in [\lambda]$, since we have

$$\tilde{\pi}_{\boldsymbol{t}}^{(i)} = \pi^*(\boldsymbol{t} + \boldsymbol{r}_i) - \pi^*(\boldsymbol{r}_i) \qquad \text{and} \qquad \tilde{\rho}_{\boldsymbol{t}}^{(i)} = \pi^*(\boldsymbol{t} + \boldsymbol{s}_i) - \pi^*(\boldsymbol{s}_i)$$

from the definitions of $\tilde{\pi}_{\boldsymbol{t}}^{(i)}$ and $\tilde{\rho}_{\boldsymbol{t}}^{(i)}$, we have $\tilde{\pi}_{\boldsymbol{t}}^{(i)} = \tilde{\rho}_{\boldsymbol{t}}^{(i)}$ if we have

$$\pi^*(\boldsymbol{t} + \boldsymbol{r}_i) - \pi^*(\boldsymbol{t} + \boldsymbol{s}_i) = \pi^*(\boldsymbol{r}_i) - \pi^*(\boldsymbol{s}_i) \ .$$

Now, a key observation is that each of $\{\boldsymbol{t}+\boldsymbol{r}_i, \boldsymbol{t}+\boldsymbol{s}_i\}_{i\in[\lambda]}$ and $\{\boldsymbol{r}_i, \boldsymbol{s}_i\}_{i\in[\lambda]}$ is λ pairs of two random points, so we can use Observation 1 to show that linearity holds on at least $\lambda-\mu$ pairs in each of $\{\boldsymbol{t}+\boldsymbol{r}_i, \boldsymbol{t}+\boldsymbol{s}_i\}_{i\in[\lambda]}$ and $\{\boldsymbol{r}_i, \boldsymbol{s}_i\}_{i\in[\lambda]}$, and therefore can conclude that the number of i's such that we have

$$\pi^*(\boldsymbol{t} + \boldsymbol{r}_i) - \pi^*(\boldsymbol{t} + \boldsymbol{s}_i) = \pi^*(\boldsymbol{r}_i - \boldsymbol{s}_i) = \pi^*(\boldsymbol{r}_i) - \pi^*(\boldsymbol{s}_i)$$

is at least $\lambda - 2\mu$ as desired.

Formally, we first observe that, from the no-signaling property of P^*, it suffices to show that Equation (6.18) holds when $\tilde{\pi}_{\boldsymbol{t}}, \tilde{\rho}_{\boldsymbol{t}}$ are sampled in the following way. (A notable difference from the original sampling (the one in Claim 6) is highlighted by red.)

1. Choose 2λ random points $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_\lambda, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_\lambda \in \mathbb{F}^N$.

2. Run $(\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q \cup Q' \cup Q'')$, where $Q = \{\boldsymbol{r}_i, \boldsymbol{t} + \boldsymbol{r}_i\}_{i\in[\lambda]}$, $Q' = \{\boldsymbol{s}_i, \boldsymbol{t} + \boldsymbol{s}_i\}_{i\in[\lambda]}$, and $Q'' = \{\boldsymbol{r}_i - \boldsymbol{s}_i\}_{i\in[\lambda]}$.

3. For each $i \in [\lambda]$, define $\tilde{\pi}_{\boldsymbol{t}}^{(i)} := \pi^*(\boldsymbol{t} + \boldsymbol{r}_i) - \pi^*(\boldsymbol{r}_i)$ and $\tilde{\rho}_{\boldsymbol{t}}^{(i)} := \pi^*(\boldsymbol{t} + \boldsymbol{s}_i) - \pi^*(\boldsymbol{s}_i)$.

4. Output $\tilde{\pi}_{\boldsymbol{t}} := \text{majority}(\tilde{\pi}_{\boldsymbol{t}}^{(1)}, \ldots, \tilde{\pi}_{\boldsymbol{t}}^{(\lambda)})$ and $\tilde{\rho} := \text{majority}(\tilde{\rho}_{\boldsymbol{t}}^{(1)}, \ldots, \tilde{\rho}_{\boldsymbol{t}}^{(\lambda)})$.

Now, let

$$\begin{aligned}
I_1 &:= \{i \text{ s.t. } \pi^*(\boldsymbol{s}_i) + \pi^*(\boldsymbol{r}_i - \boldsymbol{s}_i) = \pi^*(\boldsymbol{r}_i)\} \ , \\
I_2 &:= \{i \text{ s.t. } \pi^*(\boldsymbol{t} + \boldsymbol{s}_i) + \pi^*(\boldsymbol{r}_i - \boldsymbol{s}_i) = \pi^*(\boldsymbol{t} + \boldsymbol{r}_i)\} \ .
\end{aligned}$$

Observe that for any $i^* \in I_1 \cap I_2$, we have

$$\begin{aligned}
\tilde{\pi}_{\boldsymbol{t}}^{(i^*)} &= \pi^*(\boldsymbol{t} + \boldsymbol{r}_{i^*}) - \pi^*(\boldsymbol{r}_{i^*}) && \text{(from the definition)} \\
&= \left(\pi^*(\boldsymbol{t} + \boldsymbol{s}_{i^*}) + \pi^*(\boldsymbol{r}_{i^*} - \boldsymbol{s}_{i^*})\right) - \left(\pi^*(\boldsymbol{s}_{i^*}) + \pi^*(\boldsymbol{r}_{i^*} - \boldsymbol{s}_{i^*})\right) && \text{(since } i^* \in I_1 \cap I_2\text{)} \\
&= \pi^*(\boldsymbol{t} + \boldsymbol{s}_{i^*}) - \pi^*(\boldsymbol{s}_{i^*}) \\
&= \tilde{\rho}_{\boldsymbol{t}}^{(i^*)} \quad . && \text{(from the definition)}
\end{aligned}$$

Thus, to prove this sub-claim it suffices to show

$$\Pr_{\tilde{\pi}_t, \tilde{\rho}_t}\left[|I_1 \cap I_2| \geq \lambda - 2\mu\right] \geq 1 - \mathsf{negl}(\lambda) \quad , \tag{6.19}$$

where $\tilde{\pi}_{\boldsymbol{t}}, \tilde{\rho}_{\boldsymbol{t}}$ are sampled as above. Now, since $\{\boldsymbol{s}_i, \boldsymbol{r}_i - \boldsymbol{s}_i\}_{i \in [\lambda]}$ (resp., $\{t + \boldsymbol{s}_i, \boldsymbol{r}_i - \boldsymbol{s}_i\}_{i \in [\lambda]}$) are 2λ random points, Observation 1 and the no-signaling property of P^* imply that we have

$$\Pr_{\tilde{\pi}_t, \tilde{\rho}_t}\left[|I_1| \geq \lambda - \mu\right] \geq 1 - \mathsf{negl}(\lambda) \qquad \text{and} \qquad \Pr_{\tilde{\pi}_t, \tilde{\rho}_t}\left[|I_2| \geq \lambda - \mu\right] \geq 1 - \mathsf{negl}(\lambda) \quad .$$

Thus, from the union bound, we have

$$\Pr_{\tilde{\pi}_t, \tilde{\rho}_t}\left[|I_1| \geq \lambda - \mu \wedge |I_2| \geq \lambda - \mu\right] \geq 1 - \mathsf{negl}(\lambda)$$

and therefore have Equation (6.19). □

Proof of Sub-Claim 2. For editorial simplicity, we think that $\tilde{\pi}_{\boldsymbol{t}}, \tilde{\rho}_{\boldsymbol{t}}$ are sampled by the following sampling algorithm, which differs from the original one (the one in Claim 6) only syntactically.

1. Choose 2λ random points $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{2\lambda} \in \mathbb{F}^N$.
2. Run $(\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q')$, where $Q' = \{\boldsymbol{r}_i, \boldsymbol{t} + \boldsymbol{r}_i\}_{i \in [2\lambda]}$.
3. Randomly partition $\{\pi^*(\boldsymbol{t} + \boldsymbol{r}_i) - \pi^*(\boldsymbol{r}_i)\}_{i \in [2\lambda]}$ into $(\tilde{\pi}_{\boldsymbol{t}}^{(1)}, \ldots, \tilde{\pi}_{\boldsymbol{t}}^{(\lambda)})$ and $(\tilde{\rho}_{\boldsymbol{t}}^{(1)}, \ldots, \tilde{\rho}_{\boldsymbol{t}}^{(\lambda)})$.
4. Output $\tilde{\pi}_{\boldsymbol{t}} := \mathrm{majority}(\tilde{\pi}_{\boldsymbol{t}}^{(1)}, \ldots, \tilde{\pi}_{\boldsymbol{t}}^{(\lambda)})$ and $\tilde{\rho}_{\boldsymbol{t}} := \mathrm{majority}(\tilde{\rho}_{\boldsymbol{t}}^{(1)}, \ldots, \tilde{\rho}_{\boldsymbol{t}}^{(\lambda)})$.

Note that in this sampling algorithm, the event Strong-Majority is implied by the event that there exists $a_{\boldsymbol{t}} \in \mathbb{F}$ such that

$$\left|\{i \in [2\lambda] \text{ s.t. } \pi^*(\boldsymbol{t} + \boldsymbol{r}_i) - \pi^*(\boldsymbol{r}_i) = a_{\boldsymbol{t}}\}\right| \geq 2\lambda - 20\mu \quad .$$

Let

$$\begin{aligned}
I_w &:= \{i \in [2\lambda] \text{ s.t. } \pi^*(\boldsymbol{t} + \boldsymbol{r}_i) - \pi^*(\boldsymbol{r}_i) = w\} \text{ for each } w \in \mathbb{F} \quad . \\
I_{\max} &:= I_{w^*}, \text{where } w^* := \operatorname*{argmax}_{w \in \mathbb{F}}(|I_w|) \quad . \\
I_{\mathrm{others}} &:= [2\lambda] \setminus I_{\max} \quad .
\end{aligned}$$

When Strong-Majority does not occur, we have

$$|I_{\max}| < 2\lambda - 20\mu \qquad \text{and} \qquad |I_{\mathrm{others}}| \geq 20\mu \quad . \tag{6.20}$$

Now, observe that having $|I_{\text{Good-pair}}| \geq \lambda - 2\mu$ in the above sampling algorithm is equivalent to having at least $\lambda - 2\mu$ "good" pairs when partitioning $[2\lambda]$ into λ pairs of indices randomly, where we say a pair of indices is good if the two indices in the pair belong to the same set I_w for $w \in \mathbb{F}$. We show that when we have Equation (6.20), we create at least $\lambda - 2\mu$ good pairs only with negligible probability. Toward showing this fact, we consider partitioning $[2\lambda]$ into λ pairs of indices as follows.

1. Choose random 10μ indices $(i_1, \ldots, i_{10\mu})$ from I_{others}.
2. Create 10μ pairs by, for each index in $(i_1, \ldots, i_{10\mu})$, choosing an index from the remaining indices of $[2\lambda]$ (without replacement).
3. Create the remaining $\lambda - 10\mu$ pairs by randomly partitioning the remaining indices of $[2\lambda]$ into $\lambda - 10\mu$ pairs.

When partitioning $[2\lambda]$ into λ pairs of indices in this way, we create at least $\lambda - 2\mu$ good pairs in total only when we create at least $10\mu - 2\mu = 8\mu$ good pairs in Step 2. Since each pair that is created in Step 2 is good with probability at most

$$\frac{\lambda}{2\lambda - 20\mu} \leq 0.51 \qquad\qquad \text{(since we have } |I_w| \leq \lambda \text{ for } \forall I_w \neq I_{\max}\text{)}$$

(where the inequality holds for every sufficiently large λ since $\mu = \Theta(\log^2 \lambda)$), the probability that we create at least 8μ good pairs in Step 2 of the above procedure is at most

$$\binom{10\mu}{2\mu} \times (0.51)^{8\mu} \leq \left(\frac{10\mu \cdot e}{2\mu}\right)^{2\mu} \times (0.51)^{8\mu} = \left((5e)^2 \times (0.51)^8\right)^{\mu} \leq (0.9)^{\mu} = \mathsf{negl}(\lambda) \ ,$$

where we use the standard inequality $\binom{n}{k} \leq (\frac{ne}{k})^k$ in the first inequality. Thus, we create at least 8μ good pairs in Step 2 of the above procedure only with negligible probability, and thus we create at least $\lambda - 2\mu$ good pairs in total only with negligible probability. This concludes the proof of Sub-Claim 2. □

Remark 7. The proof of Sub-Claim 2 is based on the idea that is used in the analysis of previous no-signaling PCPs, e.g., [BHK16, Claim 22]. ◇

As noted above, Equation (6.17) follows from Sub-Claim 1 and Sub-Claim 2. This concludes the proof of Claim 6. □

As noted above, Claim 6 implies Claim 5. This concludes the proof of Claim 5. □

As noted above, Claim 5 implies Claim 3 and Claim 4, with which we can prove Lemma 4. This concludes the proof of Lemma 4. (By inspection, one can verify that our analysis indeed works if $\kappa_{\max}(\lambda) \geq \kappa_V(\lambda) = 2\lambda(5\lambda+3)$ since we make at most 9λ queries to P^* in all the mental experiments.) □

6.7 Proof of Lemma 5 (Tensor-Product Consistency of Self-Corrected Proof)

In this subsection, we prove Lemma 5, which says that the self-corrected proof passes the tensor-product test on any points.

Proof of Lemma 5. Fix any $\{C_\lambda\}_{\lambda\in\mathbb{N}}$, $\kappa_{\max}$, and P^*, and assume that Equation (6.3) holds for infinitely many $\lambda \in \mathbb{N}$. Let Λ be the set of those λ's, and fix any sufficiently large $\lambda \in \Lambda$. Our goal is to show Equation (6.4).

Roughly speaking, we obtain Equation (6.4) by strengthening Observation 1. Recall that Observation 1 guarantees that the self-corrected proof has the tensor-product consistency on pairs of random points (that is, the self-corrected proof $(\tilde{\pi}_f, \tilde{\pi}_g)$ satisfies $\{i \text{ s.t. } \tilde{\pi}_f(\boldsymbol{r}_i)\tilde{\pi}_f(\boldsymbol{s}_i) = \tilde{\pi}_g(\boldsymbol{r}_i \otimes \boldsymbol{s}_i)\} \geq \lambda - \mu$ for random points $\{(\boldsymbol{r}_i, \boldsymbol{s}_i)\}_{i\in[\lambda]}$). At a high level, we show Equation (6.4), which says that the self-corrected proof has the tensor-product consistency on any points $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{F}^N$, in the following two steps.

1. First, we slightly strengthen Observation 1 and show that the self-corrected proof has the tensor-product consistency on pairs of points of the form $\{(\boldsymbol{t}, \boldsymbol{r}_i)\}_{i\in[\lambda]}$, where $\boldsymbol{t}$ is any fixed point and each $\boldsymbol{r}_i$ is a random point. This strengthening is shown by reducing the tensor-product consistency on $\{(\boldsymbol{t}, \boldsymbol{r}_i)\}_{i\in[\lambda]}$ to the tensor-product consistency on $\{(\boldsymbol{t}+\boldsymbol{s}_i, \boldsymbol{r}_i)\}_{i\in[\lambda]}$ and $\{(\boldsymbol{s}_i, \boldsymbol{r}_i)\}_{i\in[\lambda]}$, where each $\boldsymbol{s}_i$ is a random point. (Notice that once this reduction is given, we can use Observation 1 since each of $\{(\boldsymbol{t}+\boldsymbol{s}_i, \boldsymbol{r}_i)\}_{i\in[\lambda]}$ and $\{(\boldsymbol{s}_i, \boldsymbol{r}_i)\}_{i\in[\lambda]}$ is, when viewed individually, pairs of random points.)

2. Next, we strengthen what is shown above and show that the self-corrected proof has the tensor-product consistency on $(\boldsymbol{u}, \boldsymbol{v})$. This strengthening is shown in a similar way to the above, namely by reducing the tensor-product consistency on $(\boldsymbol{u}, \boldsymbol{v})$ to the tensor-product consistency on $\{(\boldsymbol{u}+\boldsymbol{s}_i, \boldsymbol{v}+\boldsymbol{t}_i)\}_{i\in[\lambda]}$, $\{(\boldsymbol{u}, \boldsymbol{t}_i)\}_{i\in[\lambda]}$, $\{(\boldsymbol{s}_i, \boldsymbol{v})\}_{i\in[\lambda]}$, and $\{(\boldsymbol{s}_i, \boldsymbol{t}_i)\}_{i\in[\lambda]}$, where each $\boldsymbol{s}_i, \boldsymbol{t}_i$ are random points.

The formal argument is given below.

We first prove the following strengthened version of Observation 1.

Claim 7. *For any $\boldsymbol{t} \in \{\boldsymbol{u}, \boldsymbol{v}\}$, we have*

$$\Pr\left[\left|I'_{\text{Tensor}}\right| \geq \lambda - 2\mu \;\middle|\; \begin{array}{l} \boldsymbol{r}_i \leftarrow \mathbb{F}^N \textit{ for } \forall i \in [\lambda] \\ (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{t}\} \cup Q \cup Q') \\ \quad \textit{where } Q = \{\boldsymbol{r}_i\}_{i\in[\lambda]} \textit{ and } Q' = \{\boldsymbol{t} \otimes \boldsymbol{r}_i\}_{i\in[\lambda]} \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

where

$$I'_{\text{Tensor}} := \left\{i \textit{ s.t. } \tilde{\pi}_f(\boldsymbol{t})\tilde{\pi}_f(\boldsymbol{r}_i) = \tilde{\pi}_g(\boldsymbol{t} \otimes \boldsymbol{r}_i)\right\} \quad .$$

Proof. For concreteness, we consider the case of $\boldsymbol{t} = \boldsymbol{u}$. (The case of $\boldsymbol{t} = \boldsymbol{v}$ can be handled similarly.)

From the no-signaling property of Self-Correct (Lemma 2), it suffices to show

$$\Pr\left[\left|I'_{\text{Tensor}}\right| \geq \lambda - 2\mu \;\middle|\; \begin{array}{l} \boldsymbol{r}_i, \boldsymbol{s}_i \leftarrow \mathbb{F}^N \text{ for } \forall i \in [\lambda] \\ (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{u}\} \cup Q \cup Q') \\ \text{where } Q = \{\boldsymbol{r}_i, \boldsymbol{s}_i, \boldsymbol{u}+\boldsymbol{s}_i\}_{i\in[\lambda]} \text{ and} \\ \qquad\quad Q' = \{\boldsymbol{u} \otimes \boldsymbol{r}_i, (\boldsymbol{u}+\boldsymbol{s}_i) \otimes \boldsymbol{r}_i, \boldsymbol{s}_i \otimes \boldsymbol{r}_i\}_{i\in[\lambda]} \end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \quad . \tag{6.21}$$

In the remaining of this proof, for any event E, we use $\Pr_{\tilde{\pi}}[E]$ to denote the probability of E occurring when $\tilde{\pi}$ (and others) is sampled as in Equation (6.21). Let

$$\begin{aligned} I_1 &:= \left\{i \text{ s.t. } \tilde{\pi}_f(\boldsymbol{u}+\boldsymbol{s}_i) = \tilde{\pi}_f(\boldsymbol{u}) + \tilde{\pi}_f(\boldsymbol{s}_i)\right\} \quad , \\ I_2 &:= \left\{i \text{ s.t. } \tilde{\pi}_g((\boldsymbol{u}+\boldsymbol{s}_i) \otimes \boldsymbol{r}_i) = \tilde{\pi}_g(\boldsymbol{u} \otimes \boldsymbol{r}_i) + \tilde{\pi}_g(\boldsymbol{s}_i \otimes \boldsymbol{r}_i)\right\} \quad , \\ I_3 &:= \left\{i \text{ s.t. } \tilde{\pi}_f(\boldsymbol{u}+\boldsymbol{s}_i)\tilde{\pi}_f(\boldsymbol{r}_i) = \tilde{\pi}_g((\boldsymbol{u}+\boldsymbol{s}_i) \otimes \boldsymbol{r}_i)\right\} \quad , \\ I_4 &:= \left\{i \text{ s.t. } \tilde{\pi}_f(\boldsymbol{s}_i)\tilde{\pi}_f(\boldsymbol{r}_i) = \tilde{\pi}_g(\boldsymbol{s}_i \otimes \boldsymbol{r}_i)\right\} \quad . \end{aligned}$$

Observe that for every i such that $i \in I_1 \cap I_2 \cap I_3 \cap I_4$, we have

$$\begin{aligned}
\tilde{\pi}_f(\boldsymbol{u})\tilde{\pi}_f(\boldsymbol{r}_i) &= \tilde{\pi}_f(\boldsymbol{u}+\boldsymbol{s}_i)\tilde{\pi}_f(\boldsymbol{r}_i) - \tilde{\pi}_f(\boldsymbol{s}_i)\tilde{\pi}_f(\boldsymbol{r}_i) && \text{(since } i \in I_1\text{)} \\
&= \tilde{\pi}_g((\boldsymbol{u}+\boldsymbol{s}_i)\otimes \boldsymbol{r}_i) - \tilde{\pi}_g(\boldsymbol{s}_i \otimes \boldsymbol{r}_i) && \text{(since } i \in I_3 \cap I_4\text{)} \\
&= \tilde{\pi}_g(\boldsymbol{u}\otimes \boldsymbol{r}_i) && \text{(since } i \in I_2\text{)} \ .
\end{aligned}$$

Thus, to show Equation (6.21), it suffices to show

$$\Pr_{\tilde{\pi}}\left[|I_1 \cap I_2 \cap I_3 \cap I_4| \geq \lambda - 2\mu\right] \geq 1 - \mathsf{negl}(\lambda) \ . \tag{6.22}$$

First, we have

$$\Pr_{\tilde{\pi}}\left[|I_1| = \lambda\right] \geq 1 - \mathsf{negl}(\lambda) \qquad \text{and} \qquad \Pr_{\tilde{\pi}}\left[|I_2| = \lambda\right] \geq 1 - \mathsf{negl}(\lambda)$$

since the linearity of the self-corrected proof (Lemma 4) and the no-signaling property of Self-Correct (Lemma 2) guarantee that we have

$$\begin{aligned}
&\Pr_{\tilde{\pi}}\left[\tilde{\pi}_f(\boldsymbol{u}+\boldsymbol{s}_i) = \tilde{\pi}_f(\boldsymbol{u}) + \tilde{\pi}_f(\boldsymbol{s}_i)\right] \geq 1 - \mathsf{negl}(\lambda) \ , \\
&\Pr_{\tilde{\pi}}\left[\tilde{\pi}_g(\boldsymbol{u}\otimes \boldsymbol{r}_i + \boldsymbol{s}_i \otimes \boldsymbol{r}_i) = \tilde{\pi}_g(\boldsymbol{u}\otimes \boldsymbol{r}_i) + \tilde{\pi}_g(\boldsymbol{s}_i \otimes \boldsymbol{r}_i)\right] \geq 1 - \mathsf{negl}(\lambda)
\end{aligned}$$

for every $i \in [\lambda]$. Next, we have

$$\Pr_{\tilde{\pi}}\left[|I_3| \geq \lambda - \mu\right] \geq 1 - \mathsf{negl}(\lambda) \qquad \text{and} \qquad \Pr_{\tilde{\pi}}\left[|I_4| \geq \lambda - \mu\right] \geq 1 - \mathsf{negl}(\lambda)$$

from Observation 1 since each of $(\boldsymbol{u}+\boldsymbol{s}_1, \boldsymbol{r}_1), \ldots, (\boldsymbol{u}+\boldsymbol{s}_\lambda, \boldsymbol{r}_\lambda)$ and $(\boldsymbol{s}_1, \boldsymbol{r}_1), \ldots, (\boldsymbol{s}_\lambda, \boldsymbol{r}_\lambda)$ are λ pairs of random points. Thus, from the union bound, we have

$$\Pr_{\tilde{\pi}}\left[|I_1| = |I_2| = \lambda \wedge |I_3| \geq \lambda - \mu \wedge |I_4| \geq \lambda - \mu\right] \geq 1 - \mathsf{negl}(\lambda)$$

and thus have Equation (6.22) as desired. Therefore, we have Equation (6.21). □

Now, we are ready to prove Lemma 5. From the no-signaling property of Self-Correct (Lemma 2), it suffices to show

$$\Pr\left[\begin{array}{c|l}
\tilde{\pi}_f(\boldsymbol{u})\tilde{\pi}_f(\boldsymbol{v}) & \boldsymbol{s}_i, \boldsymbol{t}_i \leftarrow \mathbb{F}^N \text{ for } \forall i \in [\lambda] \\
= \tilde{\pi}_g(\boldsymbol{u}\otimes \boldsymbol{v}) & (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{u}\otimes \boldsymbol{v}\} \cup Q \cup Q') \\
 & \text{where } Q = \{\boldsymbol{s}_i, \boldsymbol{t}_i, \boldsymbol{u}+\boldsymbol{s}_i, \boldsymbol{v}+\boldsymbol{t}_i\}_{i\in[\lambda]} \text{ and} \\
 & \qquad Q' = \{(\boldsymbol{u}+\boldsymbol{s}_i)\otimes(\boldsymbol{v}+\boldsymbol{t}_i), \boldsymbol{u}\otimes \boldsymbol{t}_i, \boldsymbol{s}_i\otimes \boldsymbol{v}, \boldsymbol{s}_i \otimes \boldsymbol{t}_i\}_{i\in[\lambda]}
\end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \ . \tag{6.23}$$

In the remaining of this proof, for any event E, we use $\Pr_{\tilde{\pi}}[E]$ to denote the probability of E occurring when $\tilde{\pi}$ (and others) is sampled as in Equation (6.23). Let

$$\begin{aligned}
I_1 &:= \left\{i \text{ s.t. } \tilde{\pi}_f(\boldsymbol{u}+\boldsymbol{s}_i) = \tilde{\pi}_f(\boldsymbol{u}) + \tilde{\pi}_f(\boldsymbol{s}_i)\right\} \quad , \\
I_1' &:= \left\{i \text{ s.t. } \tilde{\pi}_f(\boldsymbol{v}+\boldsymbol{t}_i) = \tilde{\pi}_f(\boldsymbol{v}) + \tilde{\pi}_f(\boldsymbol{t}_i)\right\} \quad , \\
I_2 &:= \left\{i \text{ s.t. } \begin{array}{l} \tilde{\pi}_g((\boldsymbol{u}+\boldsymbol{s}_i)\otimes(\boldsymbol{v}+\boldsymbol{t}_i)) \\ = \tilde{\pi}_g(\boldsymbol{u}\otimes\boldsymbol{v}) + \tilde{\pi}_g(\boldsymbol{u}\otimes\boldsymbol{t}_i) + \tilde{\pi}_g(\boldsymbol{s}_i\otimes\boldsymbol{v}) + \tilde{\pi}_g(\boldsymbol{s}_i\otimes\boldsymbol{t}_i) \end{array}\right\} \quad , \\
I_3 &:= \left\{i \text{ s.t. } \tilde{\pi}_f(\boldsymbol{u}+\boldsymbol{s}_i)\tilde{\pi}_f(\boldsymbol{v}+\boldsymbol{t}_i) = \tilde{\pi}_g((\boldsymbol{u}+\boldsymbol{s}_i)\otimes(\boldsymbol{v}+\boldsymbol{t}_i))\right\} \quad , \\
I_4 &:= \left\{i \text{ s.t. } \tilde{\pi}_f(\boldsymbol{s}_i)\tilde{\pi}_f(\boldsymbol{t}_i) = \tilde{\pi}_g(\boldsymbol{s}_i\otimes\boldsymbol{t}_i)\right\} \quad , \\
I_5 &:= \left\{i \text{ s.t. } \tilde{\pi}_f(\boldsymbol{u})\tilde{\pi}_f(\boldsymbol{t}_i) = \tilde{\pi}_g(\boldsymbol{u}\otimes\boldsymbol{t}_i)\right\} \quad , \\
I_6 &:= \left\{i \text{ s.t. } \tilde{\pi}_f(\boldsymbol{s}_i)\tilde{\pi}_f(\boldsymbol{v}) = \tilde{\pi}_g(\boldsymbol{s}_i\otimes\boldsymbol{v})\right\} \quad .
\end{aligned}$$

Observe that for every i such that $i \in I_1 \cap I_1' \cap I_2 \cap I_3 \cap I_4 \cap I_5 \cap I_6$, we have

$$\begin{aligned}
&\tilde{\pi}_f(\boldsymbol{u})\tilde{\pi}_f(\boldsymbol{v}) \\
&= \tilde{\pi}_f(\boldsymbol{u}+\boldsymbol{s}_i)\tilde{\pi}_f(\boldsymbol{v}+\boldsymbol{t}_i) - \tilde{\pi}_f(\boldsymbol{u})\tilde{\pi}_f(\boldsymbol{t}_i) - \tilde{\pi}_f(\boldsymbol{s}_i)\tilde{\pi}_f(\boldsymbol{v}) - \tilde{\pi}_f(\boldsymbol{s}_i)\tilde{\pi}_f(\boldsymbol{t}_i) && (\text{since } i \in I_1 \cap I_1') \\
&= \tilde{\pi}_g((\boldsymbol{u}+\boldsymbol{s}_i)\otimes(\boldsymbol{v}+\boldsymbol{t}_i)) - \tilde{\pi}_g(\boldsymbol{u}\otimes\boldsymbol{t}_i) - \tilde{\pi}_g(\boldsymbol{s}_i\otimes\boldsymbol{v}) - \tilde{\pi}_g(\boldsymbol{s}_i\otimes\boldsymbol{t}_i) && (\text{since } i \in I_3 \cap I_4 \cap I_5 \cap I_6) \\
&= \tilde{\pi}_g(\boldsymbol{u}\otimes\boldsymbol{v}) && (\text{since } i \in I_2) \quad .
\end{aligned}$$

Thus, to show Equation (6.23), it suffices to show

$$\Pr_{\tilde{\pi}}[I_1 \cap I_2 \cap I_3 \cap I_4 \neq \emptyset] \geq 1 - \mathsf{negl}(\lambda) \quad . \tag{6.24}$$

First, we have

$$\begin{aligned}
&\Pr_{\tilde{\pi}}[|I_1| = \lambda] \geq 1 - \mathsf{negl}(\lambda) \quad , \\
&\Pr_{\tilde{\pi}}\left[|I_1'| = \lambda\right] \geq 1 - \mathsf{negl}(\lambda) \quad , \\
&\Pr_{\tilde{\pi}}[|I_2| = \lambda] \geq 1 - \mathsf{negl}(\lambda) \quad .
\end{aligned}$$

since the linearity of the self-corrected proof (Lemma 4 and Lemma 9) and the no-signaling property of Self-Correct (Lemma 2) guarantees that we have

$$\begin{aligned}
&\Pr_{\tilde{\pi}}\left[\tilde{\pi}_f(\boldsymbol{u}+\boldsymbol{s}_i) = \tilde{\pi}_f(\boldsymbol{u}) + \tilde{\pi}_f(\boldsymbol{s}_i)\right] \geq 1 - \mathsf{negl}(\lambda) \quad , \\
&\Pr_{\tilde{\pi}}\left[\tilde{\pi}_f(\boldsymbol{v}+\boldsymbol{t}_i) = \tilde{\pi}_f(\boldsymbol{v}) + \tilde{\pi}_f(\boldsymbol{t}_i)\right] \geq 1 - \mathsf{negl}(\lambda) \quad , \\
&\Pr_{\tilde{\pi}}\left[\begin{array}{l} \tilde{\pi}_g(\boldsymbol{u}\otimes\boldsymbol{v} + \boldsymbol{u}\otimes\boldsymbol{t}_i + \boldsymbol{s}_i\otimes\boldsymbol{v} + \boldsymbol{s}_i\otimes\boldsymbol{t}_i) \\ = \tilde{\pi}_g(\boldsymbol{u}\otimes\boldsymbol{v}) + \tilde{\pi}_g(\boldsymbol{u}\otimes\boldsymbol{t}_i) + \tilde{\pi}_g(\boldsymbol{s}_i\otimes\boldsymbol{v}) + \tilde{\pi}_g(\boldsymbol{s}_i\otimes\boldsymbol{t}_i) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)
\end{aligned}$$

for every $i \in [\lambda]$. Next, we have

$$\Pr_{\tilde{\pi}}[|I_3| \geq \lambda - \mu] \geq 1 - \mathsf{negl}(\lambda) \qquad \text{and} \qquad \Pr_{\tilde{\pi}}[|I_4| \geq \lambda - \mu] \geq 1 - \mathsf{negl}(\lambda)$$

from Observation 1 and the no-signaling property of Self-Correct (Lemma 2) since each of $(\boldsymbol{u}+\boldsymbol{s}_1, \boldsymbol{v}+\boldsymbol{t}_1), \dots, (\boldsymbol{u}+\boldsymbol{s}_\lambda, \boldsymbol{v}+\boldsymbol{t}_\lambda)$ and $(\boldsymbol{s}_1, \boldsymbol{t}_1), \dots, (\boldsymbol{s}_\lambda, \boldsymbol{t}_\lambda)$ are λ pairs of random points, and we have

$$\Pr_{\tilde{\pi}}[|I_5| \geq \lambda - 2\mu] \geq 1 - \mathsf{negl}(\lambda) \qquad \text{and} \qquad \Pr_{\tilde{\pi}}[|I_6| \geq \lambda - 2\mu] \geq 1 - \mathsf{negl}(\lambda)$$

from Claim 7 and the no-signaling property of $\mathsf{Self\text{-}Correct}$ (Lemma 2). Thus, from the union bound, we have

$$\Pr_{\tilde{\pi}}\left[|I_1| = |I_1'| = |I_2| = \lambda \wedge |I_3 \cap I_4| \geq \lambda - 2\mu \wedge |I_5 \cap I_6| \geq \lambda - 4\mu\right] \geq 1 - \mathsf{negl}(\lambda)$$

and thus have Equation (6.24) from $\lambda - 6\mu > 0$. Therefore, we have Equation (6.23). This concludes the proof of Lemma 5. □

Remark 8. By inspection, one can verify that the proof of Lemma 5 indeed works if $\kappa_{\max}(\lambda) \geq 2\lambda(8\lambda + 3)$ since we make at most $8\lambda + 3$ queries to $\mathsf{Self\text{-}Correct}$ in all the mental experiments and we can use Lemma 4 as long as $\kappa_{\max}(\lambda) \geq \kappa_V(\lambda) = 2\lambda(5\lambda + 3)$. ◇

6.8 Proof of Lemma 6 (SAT Consistency of Self-Corrected Proof)

In this subsection, we prove Lemma 6, which says that the self-corrected proof passes the SAT-product test on any points.

Proof of Lemma 6. The high-level strategy of the proof of this lemma is the same as that of the proof of Lemma 5, namely we prove this lemma by strengthening Observation 1. The formal argument is given below.

Fix any $\{C_\lambda\}_{\lambda\in\mathbb{N}}$, $\kappa_{\max}$, and P^*, and assume that Equation (6.5) holds for infinitely many $\lambda \in \mathbb{N}$. Let Λ be the set of those λ's, and fix any sufficiently large $\lambda \in \Lambda$. Our goal is to show Equation (6.6).

From the no-signaling property of $\mathsf{Self\text{-}Correct}$ (Lemma 2), to show Equation (6.6) it suffices to show

$$\Pr\left[\tilde{\pi}_f(\boldsymbol{\psi}_\sigma) + \tilde{\pi}_g(\boldsymbol{\psi}'_\sigma) = c_\sigma \;\middle|\; \begin{array}{l} \boldsymbol{\sigma}_i \leftarrow \mathbb{F}^M \text{ for } \forall j \in [\lambda] \\ (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{\psi}_\sigma, \boldsymbol{\psi}'_\sigma\} \cup Q) \\ \quad \text{where } Q = \{\boldsymbol{\psi}_{\sigma_i}, \boldsymbol{\psi}_{\sigma+\sigma_i}, \boldsymbol{\psi}'_{\sigma_i}, \boldsymbol{\psi}'_{\sigma+\sigma_i}\}_{i\in[\lambda]} \end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \ , \tag{6.25}$$

where for any $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_M) \in \mathbb{F}^M$, we use $\boldsymbol{\psi}_\tau, \boldsymbol{\psi}'_\tau$ to denote the coefficient vectors such that

$$\langle \boldsymbol{\psi}_\tau, z \rangle + \langle \boldsymbol{\psi}'_\tau, z \otimes z \rangle = \Psi_\tau(z) := \sum_{i\in[M]} \tau_i \Psi_i(z) \ .$$

In the remaining of this proof, for any event E, we use $\Pr_{\tilde{\pi}}[E]$ to denote the probability of E occurring when $\tilde{\pi}$ is sampled as in Equation (6.25). Let

$$\begin{aligned}
I_1 &:= \left\{i \text{ s.t. } \tilde{\pi}_f(\boldsymbol{\psi}_{\sigma_i}) + \tilde{\pi}_g(\boldsymbol{\psi}'_{\sigma_i}) = c_{\sigma_i}\right\} \ , \\
I_2 &:= \left\{i \text{ s.t. } \tilde{\pi}_f(\boldsymbol{\psi}_{\sigma+\sigma_i}) + \tilde{\pi}_g(\boldsymbol{\psi}'_{\sigma+\sigma_i}) = c_{\sigma+\sigma_i}\right\} \ , \\
I_3 &:= \left\{i \text{ s.t. } \tilde{\pi}_f(\boldsymbol{\psi}_{\sigma+\sigma_i}) - \tilde{\pi}_f(\boldsymbol{\psi}_{\sigma_i}) = \tilde{\pi}_f(\boldsymbol{\psi}_\sigma)\right\} \ , \\
I_4 &:= \left\{i \text{ s.t. } \tilde{\pi}_g(\boldsymbol{\psi}'_{\sigma+\sigma_i}) - \tilde{\pi}_g(\boldsymbol{\psi}'_{\sigma_i}) = \tilde{\pi}_g(\boldsymbol{\psi}'_\sigma)\right\} \ ,
\end{aligned}$$

where $c_{\sigma_i} := \sum_{j\in[M]} \sigma_{i,j} c_j$ and $c_{\sigma+\sigma_i} := \sum_{j\in[M]} (\sigma_j + \sigma_{i,j}) c_j$. Observe that for every i such that $i \in I_1 \cap I_2 \cap I_3 \cap I_4$, we have

$$\begin{aligned}
&\tilde{\pi}_f(\boldsymbol{\psi}_\sigma) + \tilde{\pi}_g(\boldsymbol{\psi}'_\sigma) \\
&= \tilde{\pi}_f(\boldsymbol{\psi}_{\sigma+\sigma_i}) - \tilde{\pi}_f(\boldsymbol{\psi}_{\sigma_i}) + \tilde{\pi}_g(\boldsymbol{\psi}'_{\sigma+\sigma_i}) - \tilde{\pi}_g(\boldsymbol{\psi}'_{\sigma_i}) && (\text{since } i \in I_3 \cap I_4) \\
&= c_{\sigma+\sigma_i} - c_{\sigma_i} && (\text{since } i \in I_1 \cap I_2) \\
&= c_\sigma \ . && (\text{from the definitions of } c_{\sigma+\sigma_i}, c_{\sigma_i})
\end{aligned}$$

Thus, to show Equation (6.25), it suffices to show

$$\Pr_{\tilde{\pi}}[I_1 \cap I_2 \cap I_3 \cap I_4 \neq \emptyset] \geq 1 - \mathsf{negl}(\lambda) \ . \tag{6.26}$$

First, we have

$$\Pr_{\tilde{\pi}}[|I_1| \geq \lambda - \mu] \geq 1 - \mathsf{negl}(\lambda) \qquad \text{and} \qquad \Pr_{\tilde{\pi}}[|I_2| \geq \lambda - \mu] \geq 1 - \mathsf{negl}(\lambda)$$

from Observation 1 and the no-signaling property of $\mathsf{Self\text{-}Correct}$ (Lemma 2) since each of $\boldsymbol{\sigma}_i$ and $\boldsymbol{\sigma} + \boldsymbol{\sigma}_i$ is a random point in $\mathbb{F}^M$. Next, we have

$$\Pr_{\tilde{\pi}}[|I_3| = \lambda] \geq 1 - \mathsf{negl}(\lambda) \qquad \text{and} \qquad \Pr_{\tilde{\pi}}[|I_4| = \lambda] \geq 1 - \mathsf{negl}(\lambda)$$

since the linearity of the self-corrected proof (Lemma 4) and the no-signaling property of $\mathsf{Self\text{-}Correct}$ (Lemma 2) guarantee that we have

$$\Pr_{\tilde{\pi}}\left[\tilde{\pi}_f(\boldsymbol{\psi}_{\sigma_i}) + \tilde{\pi}_f(\boldsymbol{\psi}_{\sigma}) = \tilde{\pi}_f(\boldsymbol{\psi}_{\sigma+\sigma_i})\right] \geq 1 - \mathsf{negl}(\lambda) \ ,$$
$$\Pr_{\tilde{\pi}}\left[\tilde{\pi}_g(\boldsymbol{\psi}'_{\sigma_i}) + \tilde{\pi}_g(\boldsymbol{\psi}'_{\sigma}) = \tilde{\pi}_g(\boldsymbol{\psi}'_{\sigma+\sigma_i})\right] \geq 1 - \mathsf{negl}(\lambda)$$

for every $i \in [\lambda]$. Thus, from the union bound, we have

$$\Pr_{\tilde{\pi}}[|I_1| \geq \lambda - \mu \wedge |I_2| \geq \lambda - \mu \wedge |I_3| = |I_4| = \lambda] \geq 1 - \mathsf{negl}(\lambda)$$

and thus have Equation (6.26) as desired. Therefore, we have Equation (6.25). This concludes the proof of Lemma 6. □

Remark 9. By inspection, one can verify that the proof of Lemma 6 indeed works if $\kappa_{\max}(\lambda) \geq \kappa_V(\lambda) = 2\lambda(5\lambda + 3)$ since we make at most $4\lambda + 2$ queries to $\mathsf{Self\text{-}Correct}$ in the mental experiment and we can use Lemma 4 as long as $\kappa_{\max}(\lambda) \geq \kappa_V(\lambda)$. ◇

7 Analysis of Our PCP: Step 3 (Consistency with Claimed Computation)

In this section, we show that if a no-signaling cheating prover convinces the relaxed verifier with overwhelming probability, the self-corrected proof is "locally consistent" with the system of equations $\Psi = \{\Psi_i(z) = c_i\}_{i \in [M]}$. That is, we show that (1) if the wire value of an input gate is recovered from the the self-corrected proof, the recovered wire value is consistent with the input $\boldsymbol{x}$, (2) if the wire values of the input and output wires of a gate are recovered from the self-corrected proof, the recovered wire values are consistent with the computation of the gate, and (3) if the wire value of an output gate is recovered from the the self-corrected proof, the recovered wire value is consistent with the claimed output $\boldsymbol{y}$.

Lemma 10 (Consistency with Claimed Computation)**.** *Let $\mathbb{V} = (\mathbb{V}_0, \mathbb{V}_1)$ be the relaxed PCP verifier in Section 5, $\{C_\lambda\}_{\lambda \in \mathbb{N}}$ be any circuit family, and $\kappa_{\max}$ be any polynomial such that $\kappa_{\max}(\lambda) \geq 2\lambda(8\lambda + 3)$.*

Then, for any $\kappa_{\max}$-wise no-signaling cheating prover P^*, if it holds

$$\Pr\left[\mathbb{V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*) = 1 \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \tag{7.1}$$

for infinitely many $\lambda \in \mathbb{N}$ (let Λ be the set of those λ's), there exists a negligible function negl *such that for every sufficiently large $\lambda \in \Lambda$ and every $i^* \in [M]$, it holds*

$$\Pr\left[\mathsf{Consist}_{i^*}(C_\lambda, \boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{e}_\alpha, \boldsymbol{e}_\beta, \boldsymbol{e}_\gamma\}) \right] \geq 1 - \mathsf{negl}(\lambda) \ , \tag{7.2}$$

where (1) $\alpha, \beta, \gamma \in [N]$ ($\alpha < \beta < \gamma$) are any wires of C_λ such that there exist $d_j \in \{-1, 0, 1\}$ ($j \in \{\alpha, \beta, \gamma\}$) and $d_{j,k} \in \{-1, 0, 1\}$ ($j, k \in \{\alpha, \beta, \gamma\}$) such that the i^*-th equation of $\Psi = \{\Psi_i(\boldsymbol{z}) = c_i\}_{i \in [M]}$ can be written as

$$\sum_{j \in \{\alpha,\beta,\gamma\}} d_j z_j + \sum_{j,k \in \{\alpha,\beta,\gamma\}} d_{j,k} z_j z_k = c_{i^*} \ , \tag{7.3}$$

and (2) $\mathsf{Consist}_{i^*}(C_\lambda, \boldsymbol{x}, \boldsymbol{y}, \tilde{\pi})$ is the event that $\tilde{\pi}$ is consistent with i^*-th equation of Ψ, i.e., it holds

$$\sum_{j \in \{\alpha,\beta,\gamma\}} d_j \tilde{\pi}_f(\boldsymbol{e}_j) + \sum_{j,k \in \{\alpha,\beta,\gamma\}} d_{j,k} \tilde{\pi}_f(\boldsymbol{e}_j) \tilde{\pi}_f(\boldsymbol{e}_k) = c_{i^*} \ ,$$

where $\boldsymbol{e}_j := (0, \ldots, 0, \underset{\substack{\wedge \\ j\text{-th}}}{1}, \ldots, 0) \in \mathbb{F}^N$ and $\boldsymbol{e}_k := (0, \ldots, 0, \underset{\substack{\wedge \\ k\text{-th}}}{1}, \ldots, 0) \in \mathbb{F}^N$.

Proof. Fix any $\{C_\lambda\}_{\lambda \in \mathbb{N}}$, $\kappa_{\max}$, and P^*, and assume that Equation (7.1) for infinitely many $\lambda \in \mathbb{N}$. Let Λ be the set of those λ's, and fix any sufficiently large $\lambda \in \Lambda$ and any $i^* \in [M]$. Our goal is to show Equation (7.2).

Since the high-level idea is already explained in the technical overview (Section 3), we directly go to the formal argument. First, from the SAT consistency of the self-corrected proof (Lemma 6), we obtain

$$\Pr\left[\tilde{\pi}_f(\boldsymbol{\psi}_{\boldsymbol{e}_{i^*}}) + \tilde{\pi}_g(\boldsymbol{\psi}'_{\boldsymbol{e}_{i^*}}) = c_{\boldsymbol{e}_{i^*}} \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{\psi}_{\boldsymbol{e}_{i^*}}, \boldsymbol{\psi}'_{\boldsymbol{e}_{i^*}}\}) \right] \geq 1 - \mathsf{negl}(\lambda) \ . \tag{7.4}$$

Second, for any α, β, γ such that the i^*-th equation of Ψ can be written as Equation (7.3), we have

$$\boldsymbol{\psi}_{\boldsymbol{e}_{i^*}} = \sum_{j \in \{\alpha,\beta,\gamma\}} d_j \boldsymbol{e}_j \qquad \text{and} \qquad \boldsymbol{\psi}'_{\boldsymbol{e}_{i^*}} = \sum_{j,k \in \{\alpha,\beta,\gamma\}} d_{j,k} \boldsymbol{e}_j \otimes \boldsymbol{e}_k$$

from the definition of $\boldsymbol{\psi}_{\boldsymbol{e}_{i^*}}, \boldsymbol{\psi}'_{\boldsymbol{e}_{i^*}}$ (cf. Lemma 6), and thus we have

$$\Pr\left[\tilde{\pi}_f(\boldsymbol{\psi}_{\boldsymbol{e}_{i^*}}) = \sum_{j \in \{\alpha,\beta,\gamma\}} d_j \tilde{\pi}_f(\boldsymbol{e}_j) \;\middle|\; \begin{array}{c} (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, Q) \\ \text{where } Q = \{\boldsymbol{\psi}_{\boldsymbol{e}_{i^*}}\} \cup \{\boldsymbol{e}_j\}_{j \in \{\alpha,\beta,\gamma\}} \end{array} \right] \geq 1 - \mathsf{negl}(\lambda)$$

and

$$\Pr\left[\tilde{\pi}_g(\boldsymbol{\psi}'_{\boldsymbol{e}_{i^*}}) = \sum_{j,k \in \{\alpha,\beta,\gamma\}} d_{j,k} \tilde{\pi}_g(\boldsymbol{e}_j \otimes \boldsymbol{e}_k) \;\middle|\; \begin{array}{c} (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, Q) \\ \text{where } Q = \{\boldsymbol{\psi}_{\boldsymbol{e}_{i^*}}\} \cup \{\boldsymbol{e}_j \otimes \boldsymbol{e}_k\}_{j,k \in \{\alpha,\beta,\gamma\}} \end{array} \right] \geq 1 - \mathsf{negl}(\lambda)$$

from the linearity of the self-corrected proof (Lemma 4, Lemma 7, Lemma 9), the no-signaling property of Self-Correct (Lemma 2), and the union bound. Third, from the tensor-product consistency of the self-corrected proof (Lemma 5), we have

$$\Pr\left[\tilde{\pi}_g(\boldsymbol{e}_j \otimes \boldsymbol{e}_k) = \tilde{\pi}_f(\boldsymbol{e}_j)\tilde{\pi}_f(\boldsymbol{e}_k) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{e}_j, \boldsymbol{e}_k, \boldsymbol{e}_j \otimes \boldsymbol{e}_k\}) \right] \geq 1 - \mathsf{negl}(\lambda)$$

for every $j, k \in \{\alpha, \beta, \gamma\}$. Now, we obtain Equation (7.2) from all the above by using the no-signaling property of Self-Correct (Lemma 2) and the union bound. $\square$

8 Analysis of Our PCP: Step 4 (Consistency with Correct Computation)

In this section, we show that if a no-signaling cheating prover convinces the relaxed verifier with overwhelming probability, the self-corrected proof is consistent with the correct computation of $C(\boldsymbol{x})$. That is, we show that if the value of an output gate is recovered from the self-corrected proof, the recovered value is consistent with the output $C(\boldsymbol{x})$.

Lemma 11. *Let $\mathbb{V} = (\mathbb{V}_0, \mathbb{V}_1)$ be the relaxed PCP verifier in Section 5, $\{C_\lambda\}_{\lambda\in\mathbb{N}}$ be any circuit family, and $\kappa_{\max}$ be any polynomial such that $\kappa_{\max}(\lambda) \geq 2\lambda(8\lambda + 3)$.*

Then, for any $\kappa_{\max}$-wise no-signaling cheating prover P^*, if it holds

$$\Pr\left[\mathbb{V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*) = 1 \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \tag{8.1}$$

for infinitely many $\lambda \in \mathbb{N}$ (let Λ be the set of those λ's), there exists a negligible function negl *such that for every sufficiently large $\lambda \in \Lambda$ and every $i^* \in [m]$ (recall that m is the output length of C_λ), it holds*

$$\Pr\left[\tilde{\pi}(\boldsymbol{e}_{N-m+i^*}) = C_{i^*}(\boldsymbol{x}) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{e}_{N-m+i^*}\}) \right] \geq 1 - \mathsf{negl}(\lambda) \ , \tag{8.2}$$

where $\boldsymbol{e}_{N-m+i^*} = (0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbb{F}^N$ is the vector such that only the $(N - m + i^*)$-th element is 1, and $C_{i^*}(\boldsymbol{x})$ denotes the i^*-th bit of $C_\lambda(\boldsymbol{x})$.

Proof. Fix any $\{C_\lambda\}_{\lambda\in\mathbb{N}}$, $\kappa_{\max}$, and P^*, and assume that Equation (8.1) holds for infinitely many $\lambda \in \mathbb{N}$. Let Λ be the set of those λ's. Now, our goal is to show Equation (8.2) for every sufficiently large $\lambda \in \Lambda$. The high-level idea of this proof is explained in the technical overview in Section 3.

For any C_λ, we use the following notation. Recall that we assume that arithmetic circuits are "layered" in such a way that (1) the first layer consists of the input gates and the last layer consists of the output gates, and (2) each gate in the i-th layer has children in the $(i-1)$-th layer. Let (ℓ, i) denote the i-th wire in the ℓ-th layer. Let $\ell_{\max}$ be the number of the layers, and N_i be the number of the wires in the i-th layer (i.e., the number of the outgoing wires from the gates in the i-th layer); thus, we have $\sum_{i\in[\ell_{\max}]} N_i = N$, $N_1 = n$, and $N_{\ell_{\max}} = m$. For every $\ell \in [\ell_{\max}]$, let

$$D_\ell := \{\boldsymbol{v} = (v_1, \ldots, v_N) \in \mathbb{F}^N \mid v_i = 0 \text{ for } \forall i \notin \{N_{\leq \ell-1} + 1, \ldots, N_{\leq \ell-1} + N_\ell\}\} \ ,$$

where $N_{\leq \ell-1} := \sum_{i\in[\ell-1]} N_i$.

For any $\lambda \in \mathbb{N}$, we also use the following notations.

- For any $\ell \in [\ell_{\max}]$ and event E, we use $\Pr_{U_\ell,\tilde{\pi}}[E]$ to denote the probability of E occurring when U_ℓ and $\tilde{\pi}$ are sampled as follows.
  1. Sample $\boldsymbol{u}_{\ell,i} \leftarrow D_\ell$ for each $i \in [\lambda]$, and let $U_\ell := \{\boldsymbol{u}_{\ell,i}\}_{i\in[\lambda]}$.
  2. Run $(\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, U_\ell)$.
- For any $\ell \in [\ell_{\max}]$, $\boldsymbol{v} \in D_\ell$, and event E, we use $\Pr_{\boldsymbol{v},U_\ell,\tilde{\pi}}[E]$ to denote the probability of E occurring when U_ℓ and $\tilde{\pi}$ are sampled as follows.
  1. Sample $\boldsymbol{u}_{\ell,i} \leftarrow D_\ell$ for each $i \in [\lambda]$, and let $U_\ell := \{\boldsymbol{u}_{\ell,i}\}_{i\in[\lambda]}$.
  2. Run $(\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{v}\} \cup U_\ell)$.

- For any $\ell \in [\ell_{\max}-1]$ and event E, we use $\Pr_{U_\ell, U_{\ell+1}, \tilde{\pi}}[E]$ to denote the probability of E occurring when $U_\ell, U_{\ell+1}$ and $\tilde{\pi}$ are sampled as follows.

  1. Sample $\boldsymbol{u}_{\ell,i} \leftarrow D_\ell$ and $\boldsymbol{u}_{\ell+1,i} \leftarrow D_{\ell+1}$ for each $i \in [\lambda]$, and let $U_\ell := \{\boldsymbol{u}_{\ell,i}\}_{i\in[\lambda]}$ and $U_{\ell+1} := \{\boldsymbol{u}_{\ell+1,i}\}_{i\in[\lambda]}$.
  2. Run $(\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, U_\ell \cup U_{\ell+1})$.

Given these notations, we prove Lemma 11 by using the following three claims.

Claim 8. *There exists a negligible function* negl *such that for every sufficiently large* $\lambda \in \Lambda$*, we have*

$$\Pr_{U_1, \tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_1} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] \geq 1 - \mathsf{negl}(\lambda) \enspace ,$$

where $\pi := P(C, \boldsymbol{x})$ *is the honestly generated proof on input* $(C, \boldsymbol{x})$.

Claim 9. *There exists a negligible function* negl *such that for every sufficiently large* $\lambda \in \Lambda$ *and for every* $\ell \in [\ell_{\max}]$*, if we have*

$$\Pr_{U_\ell, \tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_\ell} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] \geq 0.9, \tag{8.3}$$

then for every $\boldsymbol{v} \in D_\ell$*, we have*

$$\Pr_{\boldsymbol{v}, U_\ell, \tilde{\pi}}\left[\tilde{\pi}(\boldsymbol{v}) = \pi(\boldsymbol{v}) \;\middle|\; \bigwedge_{\boldsymbol{u}\in U_\ell} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] \geq 1 - \mathsf{negl}(\lambda), \tag{8.4}$$

where π *is defined as in Claim 8.*

Claim 10. *There exists a negligible function* negl *such that for every sufficiently large* $\lambda \in \Lambda$ *and for every* $\ell \in [\ell_{\max} - 1]$*, if we have*

$$\Pr_{U_\ell, \tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_\ell} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] \geq 0.9, \tag{8.5}$$

then we have

$$\Pr_{U_\ell, U_{\ell+1}, \tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_{\ell+1}} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u}) \;\middle|\; \bigwedge_{\boldsymbol{u}\in U_\ell} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] \geq 1 - \mathsf{negl}(\lambda), \tag{8.6}$$

where π *is defined as in Claim 8.*

Before proving these claims, we finish the proof of Lemma 11 by using them.

First, we show that there exists a negligible function negl such that for every sufficiently large $\lambda \in \Lambda$ and every $\ell \in [\ell_{\max} - 1]$, if we have

$$\Pr_{U_\ell, \tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_\ell} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] \geq 0.9,$$

then we have

$$\Pr_{U_{\ell+1},\tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_{\ell+1}} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] \geq \Pr_{U_{\ell},\tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_{\ell}} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] - \mathsf{negl}(\lambda) \ . \tag{8.7}$$

Fix any sufficiently large $\lambda \in \Lambda$. Toward showing Equation (8.7), we observe that from the no-signaling property of Self-Correct (Lemma 2), we have

$$\Pr_{U_{\ell+1},\tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_{\ell+1}} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] \geq \Pr_{U_{\ell},U_{\ell+1},\tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_{\ell+1}} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] - \mathsf{negl}(\lambda)$$

and

$$\Pr_{U_{\ell},\tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_{\ell}} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] \leq \Pr_{U_{\ell},U_{\ell+1},\tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_{\ell}} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] + \mathsf{negl}(\lambda) \ ,$$

and thus, for any $\ell \in [\ell_{\max}]$, we can show Equation (8.7) by showing

$$\Pr_{U_{\ell},U_{\ell+1},\tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_{\ell+1}} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] \geq \Pr_{U_{\ell},U_{\ell+1},\tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_{\ell}} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] - \mathsf{negl}(\lambda) \ .$$

Now, we observe that this inequality indeed holds.

$$\begin{aligned}
&\Pr_{U_{\ell},U_{\ell+1},\tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_{\ell+1}} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] \\
&\geq \Pr_{U_{\ell},U_{\ell+1},\tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_{\ell+1}} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u}) \;\middle|\; \bigwedge_{\boldsymbol{u}\in U_{\ell}} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] \Pr_{U_{\ell},U_{\ell+1},\tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_{\ell}} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] \\
&\geq (1-\mathsf{negl}(\lambda)) \times \Pr_{U_{\ell},U_{\ell+1},\tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_{\ell}} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] \qquad \text{(from Claim 10)} \\
&\geq \Pr_{U_{\ell},U_{\ell+1},\tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_{\ell}} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] - \mathsf{negl}(\lambda) \ .
\end{aligned}$$

Therefore, we have Equation (8.7) for any $\ell \in [\ell_{\max} - 1]$ as desired.

Now, we are ready to show Equation (8.2). From Claim 8 and Equation (8.7), we have

$$\begin{aligned}
\Pr_{U_{\ell_{\max}},\tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_{\ell_{\max}}} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] &\geq \Pr_{U_1,\tilde{\pi}}\left[\bigwedge_{\boldsymbol{u}\in U_1} \tilde{\pi}_1(\boldsymbol{u}) = \pi_1(\boldsymbol{u})\right] - (\ell_{\max}-1)\cdot \mathsf{negl}(\lambda) \\
&\geq 1 - \mathsf{negl}(\lambda) \ .
\end{aligned}$$

By combining this inequality with Claim 9 and the no-signaling property of Self-Correct (Lemma 2), we obtain Equation (8.2). (Notice that from the construction of our PCP system, we have $\boldsymbol{e}_{N-m+i^*} \in D_{\ell_{\max}}$ and $\pi(\boldsymbol{e}_{N-m+i^*}) = C_{i^*}(\boldsymbol{x})$ for every $i^* \in [m]$.)

This concludes the proof of Lemma 11 except for proving Claim 8, Claim 9, and Claim 10. Those claims are proven in the subsequent subsections. □

8.1 Proof of Claim 8

From the no-signaling property of Self-Correct (Lemma 2) and the union bound, it suffices to show the following claim.

Claim 11. *There exists a negligible function* negl *such that for every sufficiently large $\lambda \in \Lambda$ and every $\boldsymbol{v} \in D_1$, we have*

$$\Pr\left[\tilde{\pi}(\boldsymbol{v}) = \pi(\boldsymbol{v}) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{v}\})\right] \geq 1 - \mathsf{negl}(\lambda) \enspace .$$

Furthermore, since for any $\boldsymbol{v} \in D_1$ there exist $d_1, \ldots, d_{N_1} \in \{0, \ldots, |\mathbb{F}| - 1\}$ such that

$$\boldsymbol{v} = \sum_{i \in [N_1]} d_i \boldsymbol{e}_{1,i}$$

(where each $\boldsymbol{e}_{1,i} := (0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbb{F}^N$ is the vector such that only the i-th element is 1), Claim 11 is equivalent with the following claim.

Claim 12. *There exists a negligible function* negl *such that for every sufficiently large $\lambda \in \Lambda$ and every $d_1, \ldots, d_{N_1} \in \{0, \ldots, |\mathbb{F}| - 1\}$, we have*

$$\Pr\left[\tilde{\pi}(\boldsymbol{v}) = \pi(\boldsymbol{v}) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{v}\})\right] \geq 1 - \mathsf{negl}(\lambda) \enspace , \tag{8.8}$$

where $\boldsymbol{v} := \sum_{i \in [N_1]} d_i \boldsymbol{e}_{1,i}$.

Therefore, we focus on proving Claim 12 below.

Proof of Claim 12. Before showing Equation (8.8), we first show that there exists a negligible function negl such that for every sufficiently large $\lambda \in \Lambda$, every $i \in N_1$, and every $d_i \in \{0, \ldots, |\mathbb{F}| - 1\}$, we have

$$\Pr\left[\tilde{\pi}(d_i \boldsymbol{e}_{1,i}) = \pi(d_i \boldsymbol{e}_{1,i}) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{d_i \boldsymbol{e}_{1,i}\})\right] \geq 1 - \mathsf{negl}(\lambda) \enspace .$$

Fix any sufficiently large $\lambda \in \Lambda$, any $i \in N_1$, and any $d_i \in \{0, \ldots, |\mathbb{F}| - 1\}$. From the consistency with the claimed computation of the self-corrected proof (Lemma 11), we have

$$\Pr\left[\tilde{\pi}(\boldsymbol{e}_{1,i}) = x_i \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{e}_{1,i}\})\right] \geq 1 - \mathsf{negl}(\lambda) \enspace ,$$

and from the linearity of the self-corrected proof (Lemma 7, Lemma 8), we have

$$\Pr\left[\tilde{\pi}(d_i \boldsymbol{e}_{1,i}) = d_i \tilde{\pi}(\boldsymbol{e}_{1,i}) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{e}_{1,i}, d_i \boldsymbol{e}_{1,i}\})\right] \geq 1 - \mathsf{negl}(\lambda) \enspace .$$

Therefore, from the above inequalities, the no-signaling property of Self-Correct (Lemma 2), and the union bound, we have

$$\Pr\left[\tilde{\pi}(d_i \boldsymbol{e}_{1,i}) = d_i x_i \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{d_i \boldsymbol{e}_{1,i}\})\right] \geq 1 - \mathsf{negl}(\lambda) \enspace .$$

Since we have $\pi(d_i \boldsymbol{e}_{1,i}) = d_i x_i$ from the construction of our PCP system, we have

$$\Pr\left[\tilde{\pi}\left(d_i \boldsymbol{e}_{1,i}\right) = \pi\left(d_i \boldsymbol{e}_{1,i}\right) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{d_i \boldsymbol{e}_{1,i}\})\right] \geq 1 - \mathsf{negl}(\lambda)$$

as desired.

Now, we show Equation (8.8). Fix any sufficiently large $\lambda \in \Lambda$ and any $d_1, \ldots, d_{N_1} \in \{0, \ldots, |\mathbb{F}|-1\}$. For any $k \in [N_1]$, let

$$p(k) := \Pr\left[\tilde{\pi}(\boldsymbol{v}_{\le k}) = \pi(\boldsymbol{v}_{\le k}) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{v}_{\le k}\})\right] ,$$

where $\boldsymbol{v}_{\le k} := \sum_{i \in [k]} d_i \boldsymbol{e}_{1,i}$. In this notation, our goal is to show $p(N_1) \ge 1 - \mathsf{negl}(\lambda)$. Since we have $p(1) \ge 1 - \mathsf{negl}(\lambda)$ from what we show in the previous paragraph, it suffices to show that we have $p(k) \ge p(k-1) - \mathsf{negl}(\lambda)$ for every $k \in \{2, \ldots, N\}$. Now, observe that for any $k \in \{2, \ldots, N\}$, if we have

$$\tilde{\pi}(\boldsymbol{v}_{\le k}) = \tilde{\pi}(\boldsymbol{v}_{\le k-1}) + \tilde{\pi}(d_k \boldsymbol{e}_{1,k}) \bigwedge \tilde{\pi}(\boldsymbol{v}_{\le k-1}) = \pi(\boldsymbol{v}_{\le k-1}) \bigwedge \tilde{\pi}(d_k \boldsymbol{e}_{1,k}) = \pi(d_k \boldsymbol{e}_{1,k}) ,$$

then we have $\tilde{\pi}(\boldsymbol{v}_{\le k}) = \pi(\boldsymbol{v}_{\le k})$ (this is because we have $\pi(\boldsymbol{v}_{\le k}) = \pi(\boldsymbol{v}_{\le k-1}) + \pi(d_k \boldsymbol{e}_{1,k})$ from the construction of our PCP system). In addition, observe that from the linearity of the self-corrected proof (Lemma 4), we have

$$\Pr\left[\tilde{\pi}(\boldsymbol{v}_{\le k}) = \tilde{\pi}(\boldsymbol{v}_{\le k-1}) + \tilde{\pi}(d_k \boldsymbol{e}_{1,k}) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{v}_{\le k-1}, d_k \boldsymbol{e}_{1,k}, \boldsymbol{v}_{\le k}\})\right] \ge 1 - \mathsf{negl}(\lambda) ,$$

and from what we show in the previous paragraph, we have

$$\Pr\left[\tilde{\pi}(d_k \boldsymbol{e}_{1,k}) = \pi(d_k \boldsymbol{e}_{1,k}) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{d_k \boldsymbol{e}_{1,k}\})\right] \ge 1 - \mathsf{negl}(\lambda) .$$

From the above two observations, the no-signaling property of Self-Correct (Lemma 2), and the union bound, we have $p(k) \ge p(k-1) - \mathsf{negl}(\lambda)$ for every $k \in \{2, \ldots, N\}$ as desired. This concludes the proof of Claim 12. □

As noted above, Claim 12 is equivalent with Claim 11, with which we can prove Claim 8. This concludes the proof of Claim 8.

8.2 Proof of Claim 9

Fix any sufficiently large $\lambda \in \Lambda$. Fix any $\ell \in [\ell_{\max}]$, and assume that Equation (8.3) holds. Our goal is to show Equation (8.4) for any $\boldsymbol{v} \in D_\ell$.

We first note that, to show Equation (8.4), it suffices to show

$$\Pr_{\boldsymbol{v}, U_\ell, \tilde{\pi}}\left[\tilde{\pi}(\boldsymbol{v}) \ne \pi(\boldsymbol{v}) \wedge \left(\bigwedge_{\boldsymbol{u} \in U_\ell} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right)\right] \le \mathsf{negl}(\lambda) . \tag{8.9}$$

This is because if we have Equation (8.9) and Equation (8.3), we have

$$\begin{aligned}\Pr_{\boldsymbol{v}, U_\ell, \tilde{\pi}}\left[\tilde{\pi}(\boldsymbol{v}) = \pi(\boldsymbol{v}) \;\middle|\; \bigwedge_{\boldsymbol{u} \in U_\ell} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] &\ge 1 - \frac{\Pr_{\boldsymbol{v}, U_\ell, \tilde{\pi}}\left[\tilde{\pi}(\boldsymbol{v}) \ne \pi(\boldsymbol{v}) \wedge (\bigwedge_{\boldsymbol{u} \in U_\ell} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u}))\right]}{\Pr_{\boldsymbol{v}, U_\ell, \tilde{\pi}}\left[\bigwedge_{\boldsymbol{u} \in U_\ell} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right]} \\ &\ge 1 - \mathsf{negl}(\lambda) .\end{aligned}$$

Thus, we focus on showing Equation (8.9). First, from the no-signaling property of Self-Correct (Lemma 2), it suffices to show that Equation (8.9) holds when U_ℓ and $\tilde{\pi}$ are sampled as follows.

1. Define $U_\ell = \{\boldsymbol{u}_{\ell,i}\}_{i\in[\lambda]}$ by defining each $\boldsymbol{u}_{\ell,i}$ as follows. Sample $\boldsymbol{r}_i \in D_\ell$ and $b_i \in \{0,1\}$ for each $i \in [\lambda]$; then let $\boldsymbol{u}_{\ell,i} := \boldsymbol{r}_i$ if $b_i = 0$, and let $\boldsymbol{u}_{\ell,i} := \boldsymbol{v} + \boldsymbol{r}_i$ otherwise. Additionally, let $U'_\ell := \{\boldsymbol{r}_i, \boldsymbol{v} + \boldsymbol{r}_i\}_{i\in[\lambda]} \setminus U_\ell$

2. Run $(\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{v}\} \cup U_\ell \cup U'_\ell)$.

In other words, we can obtain Equation (8.9) by showing

$$\Pr_{\boldsymbol{v},\{\boldsymbol{r}_i\},U_\ell,\tilde{\pi}}\left[\tilde{\pi}(\boldsymbol{v}) \neq \pi(\boldsymbol{v}) \wedge \left(\bigwedge_{\boldsymbol{u}\in U_\ell} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right)\right] \leq \mathsf{negl}(\lambda) \ , \tag{8.10}$$

where for any event E, we use $\Pr_{\boldsymbol{v},\{\boldsymbol{r}_i\},U_\ell,\tilde{\pi}}[E]$ to denote the probability of E occurring when U_ℓ and $\tilde{\pi}$ are sampled as above.

From the linearity of the self-corrected proof (Lemma 4), the no-signaling property of $\mathsf{Self\text{-}Correct}$ (Lemma 2), and the union bound, we have

$$\Pr_{\boldsymbol{v},\{\boldsymbol{r}_i\},U_\ell,\tilde{\pi}}\left[\bigwedge_{i\in[\lambda]} \tilde{\pi}(\boldsymbol{v} + \boldsymbol{r}_i) - \tilde{\pi}(\boldsymbol{r}_i) = \tilde{\pi}(\boldsymbol{v})\right] \geq 1 - \mathsf{negl}(\lambda) \ .$$

Hence,

$$\begin{aligned}
&\Pr_{\boldsymbol{v},\{\boldsymbol{r}_i\},U_\ell,\tilde{\pi}}\left[\tilde{\pi}(\boldsymbol{v}) \neq \pi(\boldsymbol{v}) \wedge \left(\bigwedge_{\boldsymbol{u}\in U_\ell} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right)\right] \\
&\leq \Pr_{\boldsymbol{v},\{\boldsymbol{r}_i\},U_\ell,\tilde{\pi}}\left[\tilde{\pi}(\boldsymbol{v}) \neq \pi(\boldsymbol{v}) \wedge \left(\bigwedge_{\boldsymbol{u}\in U_\ell} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right) \wedge \left(\bigwedge_{i\in[\lambda]} \tilde{\pi}(\boldsymbol{v} + \boldsymbol{r}_i) - \tilde{\pi}(\boldsymbol{r}_i) = \tilde{\pi}(\boldsymbol{v})\right)\right] + \mathsf{negl}(\lambda) \\
&\leq \Pr_{\boldsymbol{v},\{\boldsymbol{r}_i\},U_\ell,\tilde{\pi}}\left[\left(\bigwedge_{\boldsymbol{u}\in U_\ell} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right) \middle| \ \tilde{\pi}(\boldsymbol{v}) \neq \pi(\boldsymbol{v}) \wedge \left(\bigwedge_{i\in[\lambda]} \tilde{\pi}(\boldsymbol{v} + \boldsymbol{r}_i) - \tilde{\pi}(\boldsymbol{r}_i) = \tilde{\pi}(\boldsymbol{v})\right)\right] + \mathsf{negl}(\lambda) \ .
\end{aligned} \tag{8.11}$$

Now, observe that when

$$\tilde{\pi}(\boldsymbol{v}) \neq \pi(\boldsymbol{v}) \wedge \left(\bigwedge_{i\in[\lambda]} \tilde{\pi}(\boldsymbol{v} + \boldsymbol{r}_i) - \tilde{\pi}(\boldsymbol{r}_i) = \tilde{\pi}(\boldsymbol{v})\right)$$

occurs, we have either $\tilde{\pi}(\boldsymbol{v} + \boldsymbol{r}_i) \neq \pi(\boldsymbol{v} + \boldsymbol{r}_i)$ or $\tilde{\pi}(\boldsymbol{r}_i) \neq \pi(\boldsymbol{r}_i)$ for every $i \in [\lambda]$ since we have $\pi(\boldsymbol{v} + \boldsymbol{r}_i) - \pi(\boldsymbol{r}_i) = \pi(\boldsymbol{v})$ for every $i \in [\lambda]$ from the construction of our PCP system. Then, since each $\boldsymbol{u}_{\ell,i}$ is defined by taking either $\boldsymbol{r}_i$ or $\boldsymbol{v} + \boldsymbol{r}_i$ randomly, we have

$$\Pr_{\boldsymbol{v},\{\boldsymbol{r}_i\},U_\ell,\tilde{\pi}}\left[\left(\bigwedge_{\boldsymbol{u}\in U_\ell} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right) \middle| \ \tilde{\pi}(\boldsymbol{v}) \neq \pi(\boldsymbol{v}) \wedge \left(\bigwedge_{i\in[\lambda]} \tilde{\pi}(\boldsymbol{v} + \boldsymbol{r}_i) - \tilde{\pi}(\boldsymbol{r}_i) = \tilde{\pi}(\boldsymbol{v})\right)\right] \leq 2^{-\lambda} \ . \tag{8.12}$$

Thus, by combining Equations (8.11) and (8.12), we obtain Equation (8.10) as desired. This concludes the proof of Claim 9.

8.3 Proof of Claim 10

Fix any sufficiently large $\lambda \in \Lambda$. Fix any $\ell \in [\ell_{\max} - 1]$ and assume that Equation (8.5) holds. Our goal is to show Equation (8.6).

In this proof, for any event E, we use

$$\Pr[E]_{U_\ell}$$

as the shorthand of

$$\Pr\left[E \;\middle|\; \bigwedge_{\boldsymbol{u} \in U_\ell} \tilde{\pi}(\boldsymbol{u}) = \pi(\boldsymbol{u})\right] .$$

We remark that from Equation (8.5), it follows that if we have $\Pr[E] = \mathsf{negl}(\lambda)$, we also have $\Pr[E]_{U_\ell} = \mathsf{negl}(\lambda)$.

First, we notice that, to prove Claim 10, it suffices to show that for any $i^* \in [N_{\ell+1}]$, we have

$$\Pr\left[\tilde{\pi}(\boldsymbol{e}_{\ell+1,i^*}) = \pi(\boldsymbol{e}_{\ell+1,i^*}) \;\middle|\; \begin{array}{l} \boldsymbol{u}_{\ell,i} \leftarrow D_\ell \text{ for } \forall i \in [\lambda] \\ U_\ell := \{\boldsymbol{u}_{\ell,i}\}_{i \in [\lambda]} \\ (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{e}_{\ell+1,i^*}\} \cup U_\ell) \end{array}\right]_{U_\ell} \geq 1 - \mathsf{negl}(\lambda) , \tag{8.13}$$

where $\boldsymbol{e}_{\ell+1,i^*} := (0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbb{F}^N$ is the vector such that only the $(N_{\leq \ell} + i^*)$-th element is 1 (recall that $N_{\leq \ell} := \sum_{i \in [\ell]} N_i$). Indeed, if we have Equation (8.13), we can argue as in the proof of Claim 8 that for any $\boldsymbol{v} = \sum_{i \in [N_{\ell+1}]} d_i \boldsymbol{e}_{\ell+1,i}$ (where $d_1, \ldots, d_{N_{\ell+1}} \in \{0, \ldots, |\mathbb{F}| - 1\}$), we have

$$\Pr\left[\tilde{\pi}(\boldsymbol{v}) = \pi(\boldsymbol{v}) \;\middle|\; \begin{array}{l} \boldsymbol{u}_{\ell,i} \leftarrow D_\ell \text{ for } \forall i \in [\lambda] \\ U_\ell := \{\boldsymbol{u}_{\ell,i}\}_{i \in [\lambda]} \\ (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, \{\boldsymbol{v}\} \cup U_\ell) \end{array}\right]_{U_\ell} \geq 1 - \mathsf{negl}(\lambda) ,$$

and thus we can obtain Equation (8.6) from the no-signaling property of Self-Correct (Lemma 2) and the union bound.

Thus, our goal is to show Equation (8.13). To simplify the exposition, we only consider the case that $i^* \in [N_{\ell+1}]$ is such that the wire $(\ell + 1, i^*)$ is the output wire of an addition gate (the other cases can be handled similarly), and denote the input wires of this addition gate by (ℓ, j^*) and (ℓ, k^*). First, observe that from the consistency with the claimed computation of the self-corrected proof (Lemma 11), we have

$$\Pr\left[\tilde{\pi}(\boldsymbol{e}_{\ell+1,i^*}) = \tilde{\pi}(\boldsymbol{e}_{\ell,j^*}) + \tilde{\pi}(\boldsymbol{e}_{\ell,k^*}) \;\middle|\; \begin{array}{l} \boldsymbol{u}_{\ell,i} \leftarrow D_\ell \text{ for } \forall i \in [\lambda] \\ U_\ell := \{\boldsymbol{u}_{\ell,i}\}_{i \in [\lambda]} \\ (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, Q \cup U_\ell) \\ \quad \text{where } Q = \{\boldsymbol{e}_{\ell+1,i^*}, \boldsymbol{e}_{\ell,j^*}, \boldsymbol{e}_{\ell,k^*}\} \end{array}\right]_{U_\ell} \geq 1 - \mathsf{negl}(\lambda) .$$

By combining this observation with $\pi(\boldsymbol{e}_{\ell+1,i^*}) = \pi(\boldsymbol{e}_{\ell,j^*}) + \pi(\boldsymbol{e}_{\ell,k^*})$ (which follows from the construction of our PCP system), we can obtain Equation (8.13) by showing

$$\Pr\left[\begin{array}{l} \tilde{\pi}(\boldsymbol{e}_{\ell,j^*}) = \pi(\boldsymbol{e}_{\ell,j^*}) \\ \wedge\, \tilde{\pi}(\boldsymbol{e}_{\ell,k^*}) = \pi(\boldsymbol{e}_{\ell,k^*}) \end{array} \;\middle|\; \begin{array}{l} \boldsymbol{u}_{\ell,i} \leftarrow D_\ell \text{ for } \forall i \in [\lambda] \\ U_\ell := \{\boldsymbol{u}_{\ell,i}\}_{i \in [\lambda]} \\ (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{P^*}(1^\lambda, C_\lambda, Q \cup U_\ell) \\ \quad \text{where } Q = \{\boldsymbol{e}_{\ell+1,i^*}, \boldsymbol{e}_{\ell,j^*}, \boldsymbol{e}_{\ell,k^*}\} \end{array}\right]_{U_\ell} \geq 1 - \mathsf{negl}(\lambda) . \tag{8.14}$$

Now, we notice that Equation (8.14) follows immediately from Equation (8.5), Claim 9, the no-signaling property of $\mathsf{Self\text{-}Correct}$ (Lemma 2), and the union bound.

This concludes the proof of Claim 10.

9 Analysis of Our PCP: Step 5 (Concluding Proof of No-signaling Soundness)

Finally, we conclude the proof of Theorem 1. Recall that our goal is to show that for any circuit family $\{C_\lambda\}_{\lambda\in\mathbb{N}}$, any polynomial $\kappa_{\max}$ such that $\kappa_{\max}(\lambda) \geq 2\lambda \cdot \max(8\lambda + 3, m_\lambda) + \kappa_V(\lambda)$, and any $\kappa_{\max}$-no-signaling cheating prover P^*, there exists a negligible function negl such that for every $\lambda \in \mathbb{N}$, we have

$$\Pr\left[V_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*) = 1 \wedge C_\lambda(\boldsymbol{x}) \neq \boldsymbol{y} \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow P^*(1^\lambda, C_\lambda, Q) \end{array}\right] \leq \mathsf{negl}(\lambda) \ .$$

(Recall that m_λ is the output length of C_λ and κ_V is the query complexity of (P, V).) From Lemma 1, it follows that toward this goal, it suffices to show that for any $\widetilde{\kappa}_{\max}$-wise no-signaling cheating prover $\mathbb{P}^*$ and negligible function negl, we have

$$\Pr\left[\mathbb{V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*) = 1 \wedge C_\lambda(\boldsymbol{x}) \neq \boldsymbol{y} \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow \mathbb{P}^*(1^\lambda, C_\lambda, Q) \end{array}\right] \leq 1 - \mathsf{negl}(\lambda), \tag{9.1}$$

for every sufficiently large $\lambda \in \mathbb{N}$, where $\widetilde{\kappa}_{\max} \coloneqq \kappa_{\max} - \kappa_V$. Furthermore, it is easy to see that to show Equation (9.1), it suffices to show that if there exists a negligible function negl such that we have

$$\Pr\left[\mathbb{V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*) = 1 \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow \mathbb{P}^*(1^\lambda, C_\lambda, Q) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \tag{9.2}$$

for infinitely many $\lambda \in \mathbb{N}$ (let Λ be the set of those λ's), then there exists another negligible function negl such that we have

$$\Pr\left[C_\lambda(\boldsymbol{x}) \neq \boldsymbol{y} \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow V_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow \mathbb{P}^*(1^\lambda, C_\lambda, Q) \end{array}\right] \leq \mathsf{negl}(\lambda) \tag{9.3}$$

for every sufficiently large $\lambda \in \Lambda$. Thus, we focus on showing Equation (9.3). Fix any P^*, and assume that we have Equation (9.2). Fix any sufficiently large $\lambda \in \Lambda$. We remark that since we have

$$\widetilde{\kappa}_{\max}(\lambda) = \kappa_{\max}(\lambda) - \kappa_V(\lambda) \geq 2\lambda \cdot \max(8\lambda + 3, m_\lambda) \ ,$$

$\mathsf{Self\text{-}Correct}^{\mathbb{P}^*}$ is m_λ-wise no-signaling (cf. Lemma 2). First, from the consistency with the claimed computation of the self-corrected proof (Lemma 10) and the union bound, we have

$$\Pr\left[\bigwedge_{i\in[m]} \tilde{\pi}(\boldsymbol{e}_{N-m+i}) = y_i \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{\mathbb{P}^*}(1^\lambda, C_\lambda, \{\boldsymbol{e}_{N-m+i}\}_{i\in[m]})\right] \geq 1 - \mathsf{negl}(\lambda) \ .$$

Next, from the consistency with the correct computation of the self-corrected proof (Lemma 11) and the union bound, we have

$$\Pr\left[\bigwedge_{i\in[m]} \tilde{\pi}(\boldsymbol{e}_{N-m+i}) = C_i(\boldsymbol{x}) \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{\mathbb{P}^*}(1^\lambda, C_\lambda, \{\boldsymbol{e}_{N-m+i}\}_{i\in[m]})\right] \geq 1 - \mathsf{negl}(\lambda) \ .$$

From these two inequalities and the union bound, we obtain

$$\Pr\left[C_\lambda(\boldsymbol{x}) = \boldsymbol{y} \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{\mathbb{P}^*}(1^\lambda, C_\lambda, \{\boldsymbol{e}_{N-m+i}\}_{i\in[m]})\right] \geq 1 - \mathsf{negl}(\lambda) \ .$$

Then, we use the no-signaling property of Self-Correct (Lemma 2) to obtain

$$\Pr\left[C_\lambda(\boldsymbol{x}) = \boldsymbol{y} \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \tilde{\pi}) \leftarrow \mathsf{Self\text{-}Correct}^{\mathbb{P}^*}(1^\lambda, C_\lambda, \emptyset)\right] \geq 1 - \mathsf{negl}(\lambda) \ ,$$

and use the statement indistinguishability of self-corrected proof (Lemma 3) to obtain

$$\Pr\left[C_\lambda(\boldsymbol{x}) = \boldsymbol{y} \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow \mathbb{P}^*(1^\lambda, C_\lambda, \emptyset)\right] \geq 1 - \mathsf{negl}(\lambda) \ .$$

Now, we use the no-signaling property of $\mathbb{P}^*$ to obtain Equation (9.3). This concludes the proof of the no-signaling soundness of our PCP system.

10 Application: Delegating Computation in Preprocessing Model

In this section, we give an application of our no-signaling linear PCP system to a 2-message delegation scheme for $\mathcal{P}$ in the preprecessing model. As mentioned in the introduction, we obtain our delegation scheme by applying the transformation of Kalai et al. [KRR13, KRR14] on our no-signaling linear PCP system.

We remark that our delegation scheme is actually non-interactive in the sense that, after the verifier's message is computed and published in the (expensive) offline phase, anyone can prove a statement to the verifier in the online phase by sending a single message, and the same offline verifier message can be used for proving multiple statements in the online phase. Formally, this property is guaranteed due to the adaptive soundness of our delegation scheme, which guarantees that the soundness holds even when the statement to be proven is chosen after the verifier's message.

10.1 Technical Overview

In this subsection, we give an overview of our delegation scheme. The readers who are familiar with the transformation of Kalai et al. [KRR13, KRR14] can skip this subsection.

Recall that in the setting of delegating computation, a computationally weak client asks a powerful server to perform a heavy computation, and the server returns the computation result to the client along with a proof that the result is correct. Our focus is delegation schemes for arithmetic-circuit computation, so the statement to be proven by the server is of the form $(C, \boldsymbol{x}, \boldsymbol{y})$, which states that an arithmetic circuit C outputs $\boldsymbol{y}$ on input $\boldsymbol{x}$. For simplicity, in this overview, we consider a static soundness setting where the statement is fixed before the verifier's message is generated.

In our delegation scheme, we use the following two building blocks.

- Our no-signaling linear PCP system for deterministic arithmetic-circuit computation (Section 4).
- An additive homomorphic encryption scheme HE, which is an encryption scheme such that the message space is a finite group and that anyone can efficiently compute a ciphertext of $m_0 + m_1$ from ciphertexts of any two messages m_0, m_1.

We assume that the message space of HE is a finite field $\mathbb{F}$ of prime order, and consider delegation scheme for arithmetic circuits over $\mathbb{F}$.

The high-level structure of our delegation scheme is simple. When the statement is $(C, \boldsymbol{x}, \boldsymbol{y})$, our scheme proceeds roughly as follows.

1. In the offline phase, the client first samples PCP queries Q of our PCP system, where $Q = \{\boldsymbol{q}_i\}_{i \in [\kappa_V]}$ and $\boldsymbol{q}_i = (q_{i,1}, \dots, q_{i,N'}) \in \mathbb{F}^{N'}$, where $N' := N + N^2$. Next, the client encrypts those queries by HE, where each query $\boldsymbol{q}_i$ is encrypted under a fresh key. (That is, for each $i \in [\kappa_V]$, the client samples a key pair $(\mathsf{pk}_i, \mathsf{sk}_i)$ of HE and encrypts each $q_{i,j} \in \mathbb{F}$ ($j \in [N']$) under the public-key pk_i.) Finally, the verifier sends the resultant ciphertexts $\{(\mathsf{ct}_{i,1}, \dots, \mathsf{ct}_{i,N'})\}_{i \in [\kappa_V]}$ to the server.

2. Given the ciphertexts of the PCP queries $\{(\mathsf{ct}_{i,1}, \dots, \mathsf{ct}_{i,N'})\}_{i \in [\kappa_V]}$, the server obtains ciphertexts of the PCP answers by homomorphically evaluating the PCP oracle $\pi : \mathbb{F}^{N'} \to \mathbb{F}$ under the ciphertexts (since π is a linear function, additive homomorphism of HE suffices for evaluating π[14]), and then returns the resultant ciphertexts $\{\tilde{\mathsf{ct}}_i\}_{i \in [\kappa_V]}$ to the client.

3. Given the ciphertexts of the PCP answers $\{\tilde{\mathsf{ct}}_i\}_{i \in [\kappa_V]}$, the client obtains the PCP answers by decrypting $\{\tilde{\mathsf{ct}}_i\}_{i \in [\kappa_V]}$ and then verifies the PCP answers by using the PCP decision algorithm.

The offline phase of our delegation scheme is expensive since the verifier query algorithm of our PCP system runs in time $\mathsf{poly}(\lambda + |C|)$, while the online phase is efficient since the verifier decision algorithm of our PCP system runs in time $\mathsf{poly}(\lambda + |\boldsymbol{x}| + |\boldsymbol{y}|)$. Very roughly speaking, the soundness of our scheme holds since, somewhat surprisingly, the semantic security of HE directly guarantees that the server can answer to the PCP queries under the ciphertexts of HE only in a no-signaling way. (Formally, in order to guarantee that the server is $\kappa_{\max}$-wise no-signaling for sufficiently large $\kappa_{\max}$, we need to change the above delegation scheme and add "dummy" queries to the PCP queries.)

10.1.1 Using multiplicative homomorphic encryption rather than additive one.

We can replace the additive homomorphic encryption scheme in the above scheme with a multiplicative one over prime-order bilinear group as follows: we modify the scheme so that, instead of encrypting the PCP queries $\{(q_{i,1}, \dots, q_{i,N})\}_{i \in [\kappa_V]}$ directly, the client encrypts $\{g^{q_{i,1}}, \dots, g^{q_{i,N}}\}_{i \in [\kappa_V]}$, where g is a generator of the bilinear group, and the server homomorphically evaluates the PCP oracle in the exponent of g using the multiplicative homomorphic property of HE. Since the PCP verification algorithm only involves quadratic tests on the PCP answers, the client can verify the PCP answers even when the PCP answers are encoded in the exponent of g. (Unfortunately, the security analysis cannot be straightforwardly modified to work for this modified scheme.)

10.2 Preliminaries

In this subsection, we first give the definition of delegation schemes and next give the definition of homomorphic encryption schemes.

[14]Since $\mathbb{F}$ is of prime order, it is possible to compute $\mathsf{Enc}(\mathsf{pk}, v \cdot m)$ from $\mathsf{Enc}(\mathsf{pk}, m)$ for any $v, m \in \mathbb{F}$.

10.2.1 Preprocessing non-interactive delegation scheme.

For concreteness, we focus our attention on 2-message delegation schemes with adaptive soundness, or in other words, non-interactive delegation schemes in the preprocessing model where the preprocess consists of a single message from the verifier. We remark that the following definition is essentially identical with the definition of preprocessing SNARGs (e.g., [BCI+13]) as well as the definition of adaptively sound 2-message delegation schemes of [BHK17, BKK+18]. The difference is that the following definition is tailored for deterministic arithmetic circuit computation.

A *preprocessing non-interactive delegation scheme* consists of three polynomial-time algorithms $(\mathsf{Gen}, \mathsf{Prove}, \mathsf{Verify})$ with the following syntax.

- Gen is a probabilistic algorithm such that on input the security parameter 1^λ and an arithmetic circuit C, it outputs a public-key pk and a secret key sk.
- Prove is a deterministic algorithm such that on input a public-key pk, a circuit C, and an input $\boldsymbol{x}$ of C, it outputs a proof pr.
- Verify is a deterministic algorithm such that on input a secret key sk, an input $\boldsymbol{x}$, an output $\boldsymbol{y}$, and a proof pr, it outputs a bit $b \in \{0, 1\}$.

The execution of preprocessing non-interactive delegation schemes is separated into two phases, the *offline phase* and the *online phase*.

- **Offline phase:** First, the verifier obtains an arithmetic circuit C that it wants to let the prover compute. Next, the verifier obtains $(\mathsf{pk}, \mathsf{sk})$ by running Gen on C and sends pk to the prover. After executing Gen, the verifier can erase the circuit C.
- **Online phase:** The prover, on input $\boldsymbol{x}$ (which is obtained either from the verifier or from any other process), computes the output $\boldsymbol{y} = C(\boldsymbol{x})$ and the proof $\mathsf{pr} = \mathsf{Prove}(\mathsf{pk}, C, \boldsymbol{x})$ and then sends $(\boldsymbol{x}, \boldsymbol{y}, \mathsf{pr})$ to the verifier. Given $(\boldsymbol{x}, \boldsymbol{y}, \mathsf{pr})$, the verifier verifies the proof by running $\mathsf{Verify}(\mathsf{sk}, \boldsymbol{x}, \boldsymbol{y}, \mathsf{pr})$. The online phase can be repeated multiple times on the same public key and secret key (see Remark 10 below).

Note that delegation schemes are meaningful only when the running time of Verify is much smaller than the time that is needed for computing $C(\boldsymbol{x})$.

The security requirements of preprocessing non-interactive delegation schemes are the following.

Correctness. For every security parameter $\lambda \in \mathbb{N}$, arithmetic circuit C, input $\boldsymbol{x}$ of C, and the output $\boldsymbol{y} := C(\boldsymbol{x})$,

$$\Pr\left[\mathsf{Verify}(\mathsf{sk}, \boldsymbol{x}, \boldsymbol{y}, \mathsf{pr}) = 1 \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, C) \\ \mathsf{pr} := \mathsf{Prove}(\mathsf{pk}, C, \boldsymbol{x}) \end{array}\right] = 1 \enspace .$$

Soundness. For every circuit family $\{C_\lambda\}_{\lambda\in\mathbb{N}}$ and PPT adversary $\mathcal{A}$, there exists a negligible function negl such that for every $\lambda \in \mathbb{N}$,

$$\Pr\left[\mathsf{Verify}(\mathsf{sk}, \boldsymbol{x}, \boldsymbol{y}, \mathsf{pr}) = 1 \wedge C_\lambda(\boldsymbol{x}) \neq \boldsymbol{y} \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \mathsf{pr}) \leftarrow \mathcal{A}(1^\lambda, C_\lambda, \mathsf{pk}) \end{array}\right] \leq \mathsf{negl}(\lambda) \enspace .$$

Remark 10. It is easy to see that if a delegation scheme is sound w.r.t. the above definition, it remains sound even when the same $(\mathsf{pk}, \mathsf{sk})$ is used for generating multiple proofs as long as the results of the verification are kept secret against the cheating provers (or, equivalently, as long as a new public-key–secret-key pair is generated when the verification of a proof is rejected). ◇

10.2.2 Homomorphic encryption.

A *public-key encryption scheme* consists of three polynomial-time algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax.

- Gen is a probabilistic algorithm such that on input the security parameter 1^λ, it outputs a public-key pk and a secret key sk.
- Enc is a probabilistic algorithm such that on input a public-key pk and a message $m \in \mathbb{F}$, it outputs a ciphertext ct. (It is assumed that pk contains the information of a finite field $\mathbb{F}$, which works as the message space.)
- Dec is a deterministic algorithm such that on input a secret-key sk and a ciphertext ct, it outputs a plaintext m.

We assume the standard (perfect) completeness. For any vector $\boldsymbol{v}$, we denote by $\mathsf{Enc}(\boldsymbol{v})$ the element-wise encryption of $\boldsymbol{v}$.

The following security notion of public-key encryption schemes is used in this paper (it is easy to see that the following security notion is implied by the standard CPA-security through a simple hybrid argument).

Definition 5 ((multi-key multi-message) CPA-security)**.** *For every polynomial p and* PPT *adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$, there exists a negligible function* negl *such that for every security parameter $\lambda \in \mathbb{N}$ and every $z \in \{0,1\}^{\mathsf{poly}(\lambda)}$,*

$$\Pr\left[b = \tilde{b} \;\middle|\; \begin{array}{l} (\ell, \mathsf{st}_0) \leftarrow \mathcal{A}_0(1^\lambda, z), \textit{where } \ell \le p(\lambda) \\ (\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Gen}(1^\lambda) \textit{ for every } i \in [\ell] \\ ((m_{0,1}, \dots m_{0,\ell}), (m_{1,1}, \dots m_{1,\ell}), \mathsf{st}_1) \leftarrow \mathcal{A}_1(\mathsf{st}_0, \mathsf{pk}_1, \dots, \mathsf{pk}_\ell) \\ b \leftarrow \{0,1\} \\ \mathsf{ct}_i^* \leftarrow \mathsf{Enc}(\mathsf{pk}_i, m_{b,i}) \textit{ for every } i \in [\ell] \\ \tilde{b} \leftarrow \mathcal{A}_2(\mathsf{st}_1, \mathsf{ct}_1^*, \dots, \mathsf{ct}_\ell^*) \end{array} \right] \le \frac{1}{2} + \mathsf{negl}(\lambda) \ .$$

◇

A public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is *additive homomorphic* if it has an additional PPT algorithm Eval^+ such that, on input $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(m_1), \dots, \mathsf{ct}_{p(\lambda)} \leftarrow \mathsf{Enc}(m_{p(\lambda)})$ for any $m_1, \dots, m_{p(\lambda)} \in \mathbb{F}$ (where p is a polynomial), it outputs $\mathsf{Enc}(\sum_{i=1}^{p(\lambda)} m_i)$. Formally, Eval^+ is required to satisfy the following property.

Homomorphic Evaluation. For every polynomial p, every PPT adversary $\mathcal{A}$, and every $\lambda \in \mathbb{N}$,

$$\Pr\left[\tilde{m} = m_1 + \cdots + m_{p(\lambda)} \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda) \\ (m_1, \dots, m_{p(\lambda)}) \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{sk}), \text{ where } m_1, \dots, m_{p(\lambda)} \in \mathbb{F} \\ \mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, m_i) \text{ for every } i \in [p(\lambda)] \\ \mathsf{ct} \leftarrow \mathsf{Eval}^+(\mathsf{pk}, \mathsf{ct}_1, , \dots, \mathsf{ct}_{p(\lambda)}) \\ \tilde{m} := \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \end{array} \right] = 1 \ .$$

To simplify the exposition, for any two ciphertext $\mathsf{ct}_0, \mathsf{ct}_1$ under a public-key pk, we use $\mathsf{ct}_0 + \mathsf{ct}_1$ as a shorthand of $\mathsf{Eval}^+(\mathsf{pk}, \mathsf{ct}_0, \mathsf{ct}_1)$. Similarly, for any ciphertext ct and a scalar $k \in \mathbb{N}$, we use $k \cdot \mathsf{ct}$ as a shorthand of $\underbrace{\mathsf{ct} + \cdots + \mathsf{ct}}_{k}$.

A public-key encryption scheme is *multiplicative homomorphic* if it has a PPT algorithm Eval^* that satisfies the above property w.r.t. multiplication over $\mathbb{F}$.

10.3 Our Result

Theorem 2. *Assume the existence of an additive homomorphic encryption scheme over fields of prime order (i.e., over the additive group of the fields) or a multiplicative homomorphic encryption scheme over bilinear groups with prime order. Then, there exists a preprocessing non-interactive delegation scheme for polynomial-time arithmetic-circuit computation with the following efficiency.*

- *The running time of* Gen *is* $\mathsf{poly}(\lambda + |C|)$.
- *The running time of* Prove *is* $\mathsf{poly}(\lambda + |C|)$.
- *The running tine of* Verify *is* $\mathsf{poly}(\lambda + |\boldsymbol{x}| + |\boldsymbol{y}|)$.

Proof . We focus on the case of additive homomorphic encryption schemes over fields of prime orders. (The case of multiplicative homomorphic encryption schemes over bilinear groups is discussed in Appendix B.) Let $(\mathsf{HE.Gen}, \mathsf{HE.Enc}, \mathsf{HE.Dec})$ be the additive homomorphic encryption scheme and $(\mathsf{PCP.P}, \mathsf{PCP.V})$ be the PCP prover and verifiers of our PCP system (Section 4). Recall that our PCP system satisfies the following properties.

- It can handle arithmetic circuits over any prime-order fields. Furthermore, there exists a polynomial $\kappa_{\max}$ such that the soundness holds against any $\kappa_{\max}$-wise no-signaling adversaries.[15]
- For an arithmetic circuit C over a finite filed $\mathbb{F}$, $\mathsf{PCP.P}$ outputs a linear function $\pi : \mathbb{F}^{N+N^2} \to \mathbb{F}$ as the PCP proof, where N is the number of wires in C. (To simplify the notations, we let $N' := N + N^2$ in what follows.) Since π is linear, there exists $d_1, \ldots, d_{N'} \in \mathbb{F}$ such that $\pi(\boldsymbol{z}) = \sum_{i\in[N']} d_i z_i$.
- For an arithmetic circuit C over a finite filed $\mathbb{F}$, $\mathsf{PCP.V}_0$ outputs a set of queries $Q = \{q_i\}_{i\in[\kappa_V(\lambda)]} \subset \mathbb{F}^{N'}$ and a state $\mathsf{st}_V \in \mathbb{F}^{n+m}$, where κ_V is a polynomial (which is independent of C) and n, m are the input and output lengths of C.

We assume that for every security parameter λ, the arithmetic circuit C to be delegated is defined over a finite field $\mathbb{F}$ that is also the message space of HE.

10.3.1 Construction.

The three algorithms $(\mathsf{Gen}, \mathsf{Prove}, \mathsf{Verify})$ are defined as follows.

- <u>**Algorithm** $\mathsf{Gen}(1^\lambda, C)$</u>

  1. Run $(Q, \mathsf{st}_V) \leftarrow \mathsf{PCP.V}_0(1^\lambda, C)$.
     Then, parse Q as $\{\boldsymbol{q}_i\}_{i\in[\kappa_V(\lambda)]}$, where $\boldsymbol{q}_i = (q_{i,1}, \ldots, q_{i,N'}) \in \mathbb{F}^{N'}$.
  2. Define $\mathsf{ct}_1, \ldots, \mathsf{ct}_{\kappa_{\max}(\lambda)}$ as follows.
     (a) Choose a random injective function $\tau : [\kappa_V(\lambda)] \to [\kappa_{\max}(\lambda)]$.
     (b) Define ct_i for each $i \in [\kappa_{\max}(\lambda)]$ by

$$\mathsf{ct}_i \leftarrow \begin{cases} \mathsf{HE.Enc}(\textsc{he}.\mathsf{pk}_i, \boldsymbol{q}_{\tau^{-1}(i)}) & (\text{if } \exists i' \in [\kappa_V(\lambda)] \text{ s.t. } \tau(i') = i) \\ \mathsf{HE.Enc}(\textsc{he}.\mathsf{pk}_i, \boldsymbol{0}) & (\text{otherwise}) \end{cases},$$

     where $(\textsc{he}.\mathsf{pk}_i, \textsc{he}.\mathsf{sk}_i) \leftarrow \mathsf{HE.Gen}(1^\lambda)$ and $\boldsymbol{0} := (0, \ldots, 0) \in \mathbb{F}^{N'}$.

[15]Formally, $\kappa_{\max}$ depends on m, which is an upper bound of the output length of the circuits to be considered.

3. Output $\mathsf{pk} := (\mathsf{ct}_1, \ldots, \mathsf{ct}_{\kappa_{\max}(\lambda)})$ and $\mathsf{sk} := (\mathsf{st}_V, \tau, \{\mathsf{HE.sk}_i\}_{i \in [\kappa_{\max}(\lambda)]})$.

- **Algorithm** $\mathsf{Prove}(\mathsf{pk}, C, \boldsymbol{x})$

1. Run $\pi \leftarrow \mathsf{PCP.P}(C, \boldsymbol{x})$.
   Let $d_1, \ldots, d_{N'} \in \mathbb{F}$ be the elements such that $\pi(z) = \sum_{i \in [N']} d_i z_i$.
2. Parse pk as $(\mathsf{ct}_1, \ldots, \mathsf{ct}_{\kappa_{\max}(\lambda)})$, where $\mathsf{ct}_i = (\mathsf{ct}_{i,1}, \ldots, \mathsf{ct}_{i,N'})$.
   Then, perform homomorphic operation to obtain
   $$\tilde{\mathsf{ct}}_i := \pi(\mathsf{ct}_i) = \sum_{j \in [N']} d_j \mathsf{ct}_{i,j}$$
   for every $i \in [\kappa_{\max}(\lambda)]$.
3. Output $\mathsf{pr} := (\tilde{\mathsf{ct}}_1, \ldots, \tilde{\mathsf{ct}}_{\kappa_{\max}(\lambda)})$.

- **Algorithm** $\mathsf{Verify}(\mathsf{sk}, \boldsymbol{x}, \boldsymbol{y}, \mathsf{pr})$

1. Parse sk as $(\mathsf{st}_V, \tau, \{\mathsf{HE.sk}_i\}_{i \in [\kappa_{\max}(\lambda)]})$, and pr as $(\tilde{\mathsf{ct}}_1, \ldots, \tilde{\mathsf{ct}}_{\kappa_{\max}(\lambda)})$
   Then, run $a_i := \mathsf{HE.Dec}(\mathsf{HE.sk}_{\tau(i)}, \tilde{\mathsf{ct}}_{\tau(i)})$ for every $i \in [\kappa_V(\lambda)]$.
2. Output $b := \mathsf{PCP.V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \{a_i\}_{i \in [\kappa_V(\lambda)]})$.

10.3.2 Security Analysis.

Correctness can be verified by inspection, so we focus on the proof of soundness.

Fix any circuit family $\{C_\lambda\}_{\lambda \in \mathbb{N}}$ and a PPT adversary $\mathcal{A}$, and assume for contradiction that we have

$$\Pr\left[\mathsf{Verify}(\mathsf{sk}, \boldsymbol{x}, \boldsymbol{y}, \mathsf{pr}) = 1 \wedge C_\lambda(\boldsymbol{x}) \neq \boldsymbol{y} \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \mathsf{pr}) \leftarrow \mathcal{A}(1^\lambda, C_\lambda, \mathsf{pk}) \end{array}\right] \geq \frac{1}{\mathsf{poly}(\lambda)} \tag{10.1}$$

for infinitely many $\lambda \in \mathbb{N}$. Our goal is to obtain, by using $\mathcal{A}$, a successful $\kappa_{\max}$-wise no-signaling cheating prover against our PCP system. That is, our goal is to obtain a $\kappa_{\max}$-wise no-signaling cheating prover $\mathsf{PCP.P}^*$ such that

$$\Pr\left[\mathsf{PCP.V}_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \pi^*) = 1 \wedge C_\lambda(\boldsymbol{x}) \neq \boldsymbol{y} \;\middle|\; \begin{array}{l} (Q, \mathsf{st}_V) \leftarrow \mathsf{PCP.V}_0(1^\lambda, C_\lambda) \\ (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow \mathsf{PCP.P}^*(1^\lambda, C_\lambda, Q) \end{array}\right] \geq \frac{1}{\mathsf{poly}(\lambda)} \tag{10.2}$$

holds for infinitely many $\lambda \in \mathbb{N}$.

Consider the following PPT $\kappa_{\max}$-wise cheating prover $\mathsf{PCP.P}^*$ against our PCP system. (Essentially, $\mathsf{PCP.P}^*$ internally executes $\mathcal{A}$ while emulating Gen and Verify for $\mathcal{A}$.)

- **Adversary** $\mathsf{PCP.P}^*(1^\lambda, C_\lambda, Q)$

1. Parse Q as $\{\boldsymbol{q}_i\}_{i \in [\kappa]}$, where $\boldsymbol{q}_i = (q_{i,1}, \ldots, q_{i,N'}) \in \mathbb{F}^{N'}$ and $\kappa := |Q|$.
2. Define $\mathsf{ct}_1, \ldots, \mathsf{ct}_{\kappa_{\max}(\lambda)}$ as follows.
   (a) Choose a random injective function $\tau : [\kappa] \to [\kappa_{\max}(\lambda)]$.

(b) Define ct_i for each $i \in [\kappa_{\max}(\lambda)]$ by

$$\mathsf{ct}_i \leftarrow \begin{cases} \mathsf{HE.Enc}(\mathsf{HE.pk}_i, \boldsymbol{q}_{\tau^{-1}(i)}) & (\text{if } \exists i' \in [\kappa] \text{ s.t. } \tau(i') = i) \\ \mathsf{HE.Enc}(\mathsf{HE.pk}_i, \mathbf{0}) & (\text{otherwise}) \end{cases} ,$$

where $(\mathsf{HE.pk}_i, \mathsf{HE.sk}_i) \leftarrow \mathsf{HE.Gen}(1^\lambda)$ and $\mathbf{0} := (0, \dots, 0) \in \mathbb{F}^{N'}$.

3. Run $(\boldsymbol{x}, \boldsymbol{y}, \mathsf{pr}) \leftarrow \mathcal{A}(1^\lambda, C_\lambda, \mathsf{pk})$, where $\mathsf{pk} := (\mathsf{ct}_1, \dots, \mathsf{ct}_{\kappa_{\max}(\lambda)})$.
4. Parse pr as $(\tilde{\mathsf{ct}}_1, \dots, \tilde{\mathsf{ct}}_{\kappa_{\max}(\lambda)})$
   Then, run $a_i := \mathsf{HE.Dec}(\mathsf{HE.sk}_{\tau(i)}, \tilde{\mathsf{ct}}_{\tau(i)})$ for every $i \in [\kappa]$.
5. Output $(\boldsymbol{x}, \boldsymbol{y}, \pi^*)$, where $\pi^* : Q \to \mathbb{F}$ is a function such that $\pi^*(\boldsymbol{q}_i) = a_i$ for every $\boldsymbol{q}_i \in Q$.

From the construction of $\mathsf{PCP.P}^*$, we directly obtain Equation (10.2) from (10.1). (Observe that $\mathsf{PCP.P}^*$ perfectly emulates Gen and Verify for $\mathcal{A}$.)

Thus, it remains to show that $\mathsf{PCP.P}^*$ is $\kappa_{\max}$-wise no-signaling. That is, it remains to show that for every PPT distinguisher $\mathcal{D}_{\mathrm{NS}}$, there exists a negligible function negl such that for every $\lambda \in \mathbb{N}$, every Q, Q' such that $Q' \subset Q$ and $|Q| \le \kappa_{\max}(\lambda)$, and every $z \in \{0,1\}^{\mathsf{poly}(\lambda)}$,

$$\left| \begin{array}{l} \Pr\left[\mathcal{D}_{\mathrm{NS}}(C_\lambda, \boldsymbol{x}, \boldsymbol{y}, \pi^*|_{Q'}, z) = 1 \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow \mathsf{PCP.P}^*(1^\lambda, C_\lambda, Q)\right] \\ \quad - \Pr\left[\mathcal{D}_{\mathrm{NS}}(C_\lambda, \boldsymbol{x}, \boldsymbol{y}, \pi^*, z) = 1 \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow \mathsf{PCP.P}^*(1^\lambda, C_\lambda, Q')\right] \end{array} \right| \le \mathsf{negl}(\lambda) . \quad (10.3)$$

We show this indistinguishability by relying on the the CPA-security of HE (Definition 5). Fix any $\mathcal{D}_{\mathrm{NS}}$, $\lambda \in \mathbb{N}$, Q and Q' such that $Q' \subset Q$ and $|Q| \le \kappa_{\max}(\lambda)$, and z. Let $z' := (z, C_\lambda, Q, Q')$. Then, consider the following adversary $\mathcal{A}^{\mathrm{HE}} = (\mathcal{A}_0^{\mathrm{HE}}, \mathcal{A}_1^{\mathrm{HE}}, \mathcal{A}_2^{\mathrm{HE}})$ against HE. (Essentially, $\mathcal{A}^{\mathrm{HE}}$ internally executes $\mathcal{A}_{\mathrm{NS}}$ and $\mathcal{D}_{\mathrm{NS}}$ while emulating $\mathsf{PCP.P}^*$ for them.)

- **Adversary** $\mathcal{A}_0^{\mathrm{HE}}(1^\lambda, z')$.
  1. Parse z' as (z, C_λ, Q, Q').
  2. Output (ℓ, st_0), where $\ell := |Q \setminus Q'|$ and $\mathsf{st}_0 := z'$.
- **Adversary** $\mathcal{A}_1^{\mathrm{HE}}(\mathsf{st}_0, \mathsf{HE.pk}_1, \dots, \mathsf{HE.pk}_\ell)$.
  1. Parse st_0 as (z, C_λ, Q, Q'), and parse Q as $\{\boldsymbol{q}_i\}_{i \in [\kappa]}$, where $\kappa := |Q|$. Let $i_1, \dots, i_\ell$ be indices such that $Q \setminus Q' = \{\boldsymbol{q}_{i_k}\}_{k \in [\ell]}$.
  2. Output $((\boldsymbol{q}_{i_1}, \dots, \boldsymbol{q}_{i_\ell}), (\mathbf{0}, \dots, \mathbf{0}))$ and $\mathsf{st}_1 := \mathsf{st}_0$.
- **Adversary** $\mathcal{A}_2^{\mathrm{HE}}(\mathsf{st}_1, \mathsf{ct}_1^*, \dots, \mathsf{ct}_\ell^*)$.
  1. Parse st_1 as (z, C_λ, Q, Q'), and parse Q as $\{\boldsymbol{q}_i\}_{i \in [\ell]}$. Let $i_1, \dots, i_\ell$ be defined as above.
  2. Define $\mathsf{ct}_1, \dots, \mathsf{ct}_{\kappa_{\max}(\lambda)}$ as follows.
     (a) Choose a random injective function $\tau : [\kappa] \to [\kappa_{\max}(\lambda)]$.
     (b) Define $\mathsf{ct}_{\tau(i_1)}, \dots, \mathsf{ct}_{\tau(i_\ell)}$ by renaming $\mathsf{ct}_1^*, \dots, \mathsf{ct}_\ell^*$ as $\mathsf{ct}_{\tau(i_1)}, \dots, \mathsf{ct}_{\tau(i_\ell)}$.
     (c) Define ct_i for each $i \in [\kappa_{\max}(\lambda)] \setminus \{\tau(i_k)\}_{k \in [\ell]}$ by

$$\mathsf{ct}_i \leftarrow \begin{cases} \mathsf{HE.Enc}(\mathsf{HE.pk}_i, \boldsymbol{q}_{\tau^{-1}(i)}) & (\text{if } \exists i' \in [\kappa] \setminus \{i_k\}_{k \in [\ell]} \text{ s.t. } \tau(i') = i) \\ \mathsf{HE.Enc}(\mathsf{HE.pk}_i, \mathbf{0}) & (\text{otherwise}) \end{cases} ,$$

where $(\mathsf{HE.pk}_i, \mathsf{HE.sk}_i) \leftarrow \mathsf{HE.Gen}(1^\lambda)$ and $\mathbf{0} := (0, \dots, 0) \in \mathbb{F}^{N'}$.

3. Run $(\boldsymbol{x}, \boldsymbol{y}, \mathsf{pr}) \leftarrow \mathcal{A}(1^\lambda, C_\lambda, \mathsf{pk})$, where $\mathsf{pk} := (\mathsf{ct}_1, \ldots, \mathsf{ct}_{\kappa_{\max}(\lambda)})$.
4. Parse pr as $(\tilde{\mathsf{ct}}_1, \ldots, \tilde{\mathsf{ct}}_{\kappa_{\max}(\lambda)})$
   Then, run $a_i := \mathsf{HE.Dec}(\mathsf{HE.sk}_{\tau(i)}, \tilde{\mathsf{ct}}_{\tau(i)})$ for every $i \in [\kappa] \setminus \{i_k\}_{k\in[\ell]}$.
5. Output $\tilde{b} \leftarrow \mathcal{D}_{\mathrm{NS}}(C_\lambda, \boldsymbol{x}, \boldsymbol{y}, \pi^*, z)$, where $\pi^* : Q' \to \mathbb{F}$ is a function such that $\pi^*(\boldsymbol{q}_i) = a_i$ for every $\boldsymbol{q}_i \in Q'$. (Observe that $Q' = \{\boldsymbol{q}_i\}_{i\in[\kappa]\setminus\{i_k\}_{k\in[\ell]}}$ holds from the definition of $\{i_k\}_{k\in[\ell]}$.)

From the construction of $\mathcal{A}^{\mathrm{HE}}$, we have

$$\Pr\left[\mathcal{A}_2^{\mathrm{HE}}(\mathsf{st}_1, \mathsf{ct}_1^*, \ldots, \mathsf{ct}_\ell^*) = 1 \;\middle|\; \begin{array}{l} (\ell, \mathsf{st}_0) \leftarrow \mathcal{A}_0(1^\lambda, z'), \text{where } \ell \le \kappa_{\max}(\lambda) \\ (\mathsf{HE.pk}_i, \mathsf{HE.sk}_i) \leftarrow \mathsf{HE.Gen}(1^\lambda) \text{ for every } i \in [\ell] \\ ((\boldsymbol{m}_1, \ldots \boldsymbol{m}_\ell), (\boldsymbol{0}, \ldots, \boldsymbol{0}), \mathsf{st}_1) \leftarrow \mathcal{A}_1^{\mathrm{HE}}(\mathsf{st}_0, \mathsf{HE.pk}_1, \ldots, \mathsf{HE.pk}_\ell) \\ \mathsf{ct}_i^* \leftarrow \mathsf{HE.Enc}(\mathsf{HE.pk}_i, \boldsymbol{m}_i) \text{ for every } i \in [\ell] \end{array}\right]$$
$$= \Pr\left[\mathcal{D}_{\mathrm{NS}}(C_\lambda, \boldsymbol{x}, \boldsymbol{y}, \pi^*|_{Q'}, z) = 1 \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow \mathsf{PCP.P}^*(1^\lambda, C_\lambda, Q)\right]$$

and

$$\Pr\left[\mathcal{A}_2^{\mathrm{HE}}(\mathsf{st}_1, \mathsf{ct}_1^*, \ldots, \mathsf{ct}_\ell^*) = 1 \;\middle|\; \begin{array}{l} (\ell, \mathsf{st}_0) \leftarrow \mathcal{A}_0(1^\lambda, z'), \text{where } \ell \le \kappa_{\max}(\lambda) \\ (\mathsf{HE.pk}_i, \mathsf{HE.sk}_i) \leftarrow \mathsf{HE.Gen}(1^\lambda) \text{ for every } i \in [\ell] \\ ((\boldsymbol{m}_1, \ldots \boldsymbol{m}_\ell), (\boldsymbol{0}, \ldots, \boldsymbol{0}), \mathsf{st}_1) \leftarrow \mathcal{A}_1^{\mathrm{HE}}(\mathsf{st}_0, \mathsf{HE.pk}_1, \ldots, \mathsf{HE.pk}_\ell) \\ \mathsf{ct}_i^* \leftarrow \mathsf{HE.Enc}(\mathsf{HE.pk}_i, \boldsymbol{0}) \text{ for every } i \in [\ell] \end{array}\right]$$
$$= \Pr\left[\mathcal{D}_{\mathrm{NS}}(C_\lambda, \boldsymbol{x}, \boldsymbol{y}, \pi^*, z) = 1 \;\middle|\; (\boldsymbol{x}, \boldsymbol{y}, \pi^*) \leftarrow \mathsf{PCP.P}^*(1^\lambda, C_\lambda, Q')\right] .$$

(Observe that $\mathcal{A}^{\mathrm{HE}}$ perfectly emulates $\mathsf{PCP.P}^*$ for $\mathcal{A}_{\mathrm{NS}}$ and $\mathcal{D}_{\mathrm{NS}}$ in both cases.) Thus, from the (multi-key multi-message) CPA-security of HE, we obtain Equation (10.3).

10.3.3 Efficiency.

By inspection, it can be verified that our delegation scheme indeed has the following efficiency.

- The running time of Gen is $\mathsf{poly}(\lambda + |C|)$.
- The running time of Prove is $\mathsf{poly}(\lambda + |C|)$.
- The running tine of Verify is $\mathsf{poly}(\lambda + |\boldsymbol{x}| + |\boldsymbol{y}|)$.

This concludes the proof of Theorem 2. □

Acknowledgment

We would like to thank the anonymous TCC2018 reviewers for their helpful comments about the presentation of this paper.

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, Cambridge, UK, 2009. A draft is available at http://theory.cs.princeton.edu/complexity/book.pdf.

[ABOR00] William Aiello, Sandeep N. Bhatt, Rafail Ostrovsky, and Sivaramakrishnan Rajagopalan. Fast verification of any remote procedure call: Short witness-indistinguishable one-round proofs for NP. In Ugo Montanari, José D. P. Rolim, and Emo Welzl, editors, *ICALP 2000*, volume 1853 of *LNCS*, pages 463–474. Springer, Heidelberg, July 2000.

[ACC+16] Prabhanjan Ananth, Yu-Chi Chen, Kai-Min Chung, Huijia Lin, and Wei-Kai Lin. Delegating RAM computations with adaptive soundness and privacy. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 3–30. Springer, Heidelberg, October / November 2016.

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

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

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, Heidelberg, August 2004.

[BC12] Nir Bitansky and Alessandro Chiesa. Succinct arguments from multi-prover interactive proofs and their efficiency benefits. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 255–272. Springer, Heidelberg, August 2012.

[BCC+17] Nir Bitansky, Ran Canetti, Alessandro Chiesa, Shafi Goldwasser, Huijia Lin, Aviad Rubinstein, and Eran Tromer. The hunting of the SNARK. *Journal of Cryptology*, 30(4):989–1066, October 2017.

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

[BCG+13] Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, Eran Tromer, and Madars Virza. SNARKs for C: Verifying program executions succinctly and in zero knowledge. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 90–108. Springer, Heidelberg, August 2013.

[BCI+13] Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Omer Paneth. Succinct non-interactive arguments via linear interactive proofs. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 315–333. Springer, Heidelberg, March 2013.

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 31–60. Springer, Heidelberg, October / November 2016.

[BCTV14] Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. Succinct non-interactive zero knowledge for a von Neumann architecture. In *Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, August 20-22, 2014.*, pages 781–796, 2014.

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *23rd ACM STOC*, pages 21–31. ACM Press, May 1991.

[BG09] Boaz Barak and Oded Goldreich. Universal arguments and their applications. *SIAM Journal on Computing*, 38(5):1661–1694, 2009.

[BGL+15] Nir Bitansky, Sanjam Garg, Huijia Lin, Rafael Pass, and Sidharth Telang. Succinct randomized encodings and their applications. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th ACM STOC*, pages 439–448. ACM Press, June 2015.

[BHK16] Zvika Brakerski, Justin Holmgren, and Yael Kalai. Non-interactive RAM and batch NP delegation from any PIR. Cryptology ePrint Archive, Report 2016/459, 2016. http://eprint.iacr.org/2016/459.

[BHK17] Zvika Brakerski, Justin Holmgren, and Yael Tauman Kalai. Non-interactive delegation and batch NP verification from standard computational assumptions. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *49th ACM STOC*, pages 474–482. ACM Press, June 2017.

[BISW17] Dan Boneh, Yuval Ishai, Amit Sahai, and David J. Wu. Lattice-based SNARGs and their application to more efficient obfuscation. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part III*, volume 10212 of *LNCS*, pages 247–277. Springer, Heidelberg, April / May 2017.

[BKK+18] Saikrishna Badrinarayanan, Yael Tauman Kalai, Dakshita Khurana, Amit Sahai, and Daniel Wichs. Succinct delegation for low-space non-deterministic computation. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th ACM STOC*, pages 709–721. ACM Press, June 2018.

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47(3):549–595, dec 1993.

[BMW98] Ingrid Biehl, Bernd Meyer, and Susanne Wetzel. Ensuring the integrity of agent-based computations by short proofs. In *Mobile Agents, Second International Workshop, MA'98, Stuttgart, Germany, September 1998, Proceedings*, pages 183–194, 1998.

[CCC+16] Yu-Chi Chen, Sherman S. M. Chow, Kai-Min Chung, Russell W. F. Lai, Wei-Kai Lin, and Hong-Sheng Zhou. Cryptography for parallel RAM from indistinguishability obfuscation. In Madhu Sudan, editor, *ITCS 2016*, pages 179–190. ACM, January 2016.

[CCHR16] Ran Canetti, Yilei Chen, Justin Holmgren, and Mariana Raykova. Adaptive succinct garbled RAM or: How to delegate your database. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 61–90. Springer, Heidelberg, October / November 2016.

[CH16] Ran Canetti and Justin Holmgren. Fully succinct garbled RAM. In Madhu Sudan, editor, *ITCS 2016*, pages 169–178. ACM, January 2016.

[CHJV15] Ran Canetti, Justin Holmgren, Abhishek Jain, and Vinod Vaikuntanathan. Succinct garbling and indistinguishability obfuscation for RAM programs. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th ACM STOC*, pages 429–437. ACM Press, June 2015.

[CKV10] Kai-Min Chung, Yael Kalai, and Salil P. Vadhan. Improved delegation of computation using fully homomorphic encryption. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 483–501. Springer, Heidelberg, August 2010.

[CMS18] Alessandro Chiesa, Peter Manohar, and Igor Shinkar. Testing linearity against non-signaling strategies. In *33rd Computational Complexity Conference (CCC 2018), 22-24 June 2018, San Diego, California, USA*, pages 17:1–17:37, 2018.

[CMS19] Alessandro Chiesa, Peter Manohar, and Igor Shinkar. Probabilistic checking against non-signaling strategies from linearity testing. In Avrim Blum, editor, *ITCS 2019*, volume 124, pages 25:1–25:17. LIPIcs, January 2019.

[DFGK14] George Danezis, Cédric Fournet, Jens Groth, and Markulf Kohlweiss. Square span programs with applications to succinct NIZK arguments. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 532–550. Springer, Heidelberg, December 2014.

[DFH12] Ivan Damgård, Sebastian Faust, and Carmit Hazay. Secure two-party computation with low communication. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 54–74. Springer, Heidelberg, March 2012.

[GGP10] Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 465–482. Springer, Heidelberg, August 2010.

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

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

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[Gol17] Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, Cambridge, UK, 2017. A draft is available at http://www.wisdom.weizmann.ac.il/~oded/PDF/pt-v3.pdf.

[Gro10] Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340. Springer, Heidelberg, December 2010.

[Gro16] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Heidelberg, May 2016.

[HR18] Justin Holmgren and Ron Rothblum. Delegating computations with (almost) minimal time and space overhead. In Mikkel Thorup, editor, *59th FOCS*, pages 124–135. IEEE Computer Society Press, October 2018.

[IKO07] Yuval Ishai, Eyal Kushilevitz, and Rafail Ostrovsky. Efficient arguments without short PCPs. In *22nd Annual IEEE Conference on Computational Complexity (CCC 2007), 13-16 June 2007, San Diego, California, USA*, pages 278–291, 2007.

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

[KLW15] Venkata Koppula, Allison Bishop Lewko, and Brent Waters. Indistinguishability obfuscation for turing machines with unbounded memory. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th ACM STOC*, pages 419–428. ACM Press, June 2015.

[KP16] Yael Tauman Kalai and Omer Paneth. Delegating RAM computations. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 91–118. Springer, Heidelberg, October / November 2016.

[KRR13] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. Delegation for bounded space. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 565–574. ACM Press, June 2013.

[KRR14] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. How to delegate computations: the power of no-signaling proofs. In David B. Shmoys, editor, *46th ACM STOC*, pages 485–494. ACM Press, May / June 2014.

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

[Lip13] Helger Lipmaa. Succinct non-interactive zero knowledge arguments from span programs and linear error-correcting codes. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 41–60. Springer, Heidelberg, December 2013.

[Mic00] Silvio Micali. Computationally sound proofs. *SIAM Journal on Computing*, 30(4):1253–1298, 2000.

[PR17] Omer Paneth and Guy N. Rothblum. On zero-testable homomorphic encryption and publicly verifiable non-interactive arguments. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part II*, volume 10678 of *LNCS*, pages 283–315. Springer, Heidelberg, November 2017.

[PRV12] Bryan Parno, Mariana Raykova, and Vinod Vaikuntanathan. How to delegate and verify in public: Verifiable computation from attribute-based encryption. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 422–439. Springer, Heidelberg, March 2012.

[SBV+13] Srinath Setty, Benjamin Braun, Victor Vu, Andrew J. Blumberg, Bryan Parno, and Michael Walfish. Resolving the conflict between generality and plausibility in verified computation. In *Proceedings of the ACM European Conference on Computer Systems (EuroSys)*. ACM, April 2013.

[SBW11] Srinath Setty, Andrew J Blumberg, and Michael Walfish. Toward practical and unconditional verification of remote computations. In *Workshop on Hot Topics in Operating Systems (HotOS)*. USENIX - Advanced Computing Systems Association, 2011.

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

[SVP+12] Srinath T. V. Setty, Victor Vu, Nikhil Panpalia, Benjamin Braun, Andrew J. Blumberg, and Michael Walfish. Taking proof-based verified computation a few steps closer to practicality. In *Proceedings of the 21th USENIX Security Symposium, Bellevue, WA, USA, August 8-10, 2012*, pages 253–268, 2012.

[VSBW13] Victor Vu, Srinath T. V. Setty, Andrew J. Blumberg, and Michael Walfish. A hybrid architecture for interactive verifiable computation. In *2013 IEEE Symposium on Security and Privacy*, pages 223–237. IEEE Computer Society Press, May 2013.

A Overview of ALMSS Linear PCP

In this section, we give an informal overview of the linear PCP system of Arora et al. [ALM+98] (ALMSS linear PCP in short). (For more formal explanations, we refer the readers to, e.g., the textbook by Arora and Barak [AB09, Chapter 11.5].) For simplicity, we focus on the case of constant soundness error.

A.1 Language for which ALMSS Linear PCP is Defined

ALMSS linear PCP is defined for a particular $\mathcal{NP}$-complete language, namely satisfiability of systems of quadratic equations over finite fields (that is, the language that consists of all the satisfiable systems of quadratic equations over finite fields). We remark that an instance of satisfiability of arithmetic circuits (i.e., the language that we use in the main body of this paper) can be easily reduced to satisfiability of quadratic equations. Indeed, given a triple $(C, \boldsymbol{x}, \boldsymbol{y})$ of an arithmetic circuit C, an input $\boldsymbol{x}$, and an output $\boldsymbol{y}$, one can efficiently obtain a system of quadratic equations that is satisfiable if and only if $C(\boldsymbol{x}) = \boldsymbol{y}$ by considering, e.g., a system that has a variable for each wire of C and has equation $z_i + z_j = z_k$ if C has an addition gate with input wires i, j and output wire k etc.

A.2 Construction of ALMSS Linear PCP and Its Analysis

Let

$$\Psi = \begin{cases} \Psi_1(\boldsymbol{z}) = c_1 \\ \vdots \\ \Psi_M(\boldsymbol{z}) = c_M \end{cases} ,$$

be a system of quadratic equation over a finite field $\mathbb{F}$, where $\boldsymbol{z} = (z_1, \ldots, z_N)$ is the variables. Below, we describe ALMSS linear PCP for the statement that Ψ is satisfiable.

The honest PCP proof of ALMSS linear PCP consists of two linear functions $\pi_f(\boldsymbol{v}) := \langle \boldsymbol{v}, \boldsymbol{w} \rangle$ and $\pi_g(\boldsymbol{v}') := \langle \boldsymbol{v}', \boldsymbol{w} \otimes \boldsymbol{w} \rangle$, where $\boldsymbol{w}$ is a satisfying assignment $\boldsymbol{w} \in \mathbb{F}^N$ to Ψ.[16] In what follows, we give a verifier that accepts this honest PCP proof with probability 1 when the statement is true while rejecting any PCP proof with high probability when the statement is false.

As a warm-up, we first give a verifier such that when it is given a PCP proof that is guaranteed to be of the form $\pi^*_f(\boldsymbol{v}) := \langle \boldsymbol{v}, \boldsymbol{w}^* \rangle$ and $\pi^*_g(\boldsymbol{v}') := \langle \boldsymbol{v}', \boldsymbol{w}^* \otimes \boldsymbol{w}^* \rangle$ for an assignment $\boldsymbol{w}^*$, it can verify whether or not $\boldsymbol{w}^*$ is a satisfying assignment. Let $\Psi_{\boldsymbol{\sigma}}(\boldsymbol{z}) = c_{\boldsymbol{\sigma}}$ be the quadratic equation that is obtained by taking a random linear combination of the equations of Ψ, i.e., by defining the left-hand side by $\Psi_{\boldsymbol{\sigma}}(\boldsymbol{z}) := \sum_{i=1}^{M} \sigma_i \Psi_i(\boldsymbol{z})$ and the right-hand size by $c_{\boldsymbol{\sigma}} := \sum_{i=1}^{M} \sigma_i c_i$ for a random $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_M) \in \mathbb{F}^M$. Now, a key observation is that we have $\Psi_{\boldsymbol{\sigma}}(\boldsymbol{w}^*) = c_{\boldsymbol{\sigma}}$ with probability 1 if $\boldsymbol{w}^*$ is a satisfying assignment, and have $\Psi_{\boldsymbol{\sigma}}(\boldsymbol{w}^*) \neq c_{\boldsymbol{\sigma}}$ with probability $1 - 1/|\mathbb{F}|$ if $\boldsymbol{w}^*$ is not a satisfying assignment. (To see the latter part, observe that if $\boldsymbol{w}^*$ is not a satisfying assignment, there exists $i^* \in [M]$ such that $\Psi_{i^*}(\boldsymbol{w}^*) \neq c_{i^*}$, so we have $\Psi_{\boldsymbol{\sigma}}(\boldsymbol{w}^*) = c_{\boldsymbol{\sigma}}$ only when we have

$$\sigma_{i^*} = \frac{\sum_{i \neq i^*} \sigma_i (c_i - \Psi_i(\boldsymbol{w}^*))}{\Psi_{i^*}(\boldsymbol{w}^*) - c_{i^*}} ,$$

which we have only with probability $1/|\mathbb{F}|$.) From this observation, it follows that a verifier can verify whether $\boldsymbol{w}^*$ is a satisfying assignment or not with soundness error $1/|\mathbb{F}|$ by checking $\Psi_{\boldsymbol{\sigma}}(\boldsymbol{w}^*) \stackrel{?}{=} c_{\boldsymbol{\sigma}}$ for random $\boldsymbol{\sigma} \in \mathbb{F}^M$. This check can be done by making only two queries to the PCP proof (this is because there exist $\boldsymbol{\psi}_{\boldsymbol{\sigma}} \in \mathbb{F}^N$ and $\boldsymbol{\psi}'_{\boldsymbol{\sigma}} \in \mathbb{F}^{N^2}$ such that $\Psi_{\boldsymbol{\sigma}}(\boldsymbol{w}^*) = \pi^*_f(\boldsymbol{\psi}_{\boldsymbol{\sigma}}) + \pi^*_g(\boldsymbol{\psi}'_{\boldsymbol{\sigma}})$), and the soundness error can be decreased by repetition.

A problem of this warm-up analysis is, of course, that it relies on a strong guarantee that the (potentially maliciously created) PCP proof (π^*_f, π^*_g) is of the form $\pi^*_f(\boldsymbol{v}) = \langle \boldsymbol{v}, \boldsymbol{w}^* \rangle$ and $\pi^*_g(\boldsymbol{v}') = \langle \boldsymbol{v}', \boldsymbol{w}^* \otimes \boldsymbol{w}^* \rangle$ for some $\boldsymbol{w}^*$. In general, π^*_f and π^*_g are not necessarily linear functions, and even if they are, there does not necessarily exist $\boldsymbol{w}^*$ such that $\pi^*_f(\boldsymbol{v}) = \langle \boldsymbol{v}, \boldsymbol{w}^* \rangle$ and $\pi^*_g(\boldsymbol{v}') = \langle \boldsymbol{v}', \boldsymbol{w}^* \otimes \boldsymbol{w}^* \rangle$.

This problem is overcome in the actual analysis by considering a verifier that additionally checks whether or not the PCP proof is "close" to the correct form. Namely, the actual verifier

1. first verifies whether or not π^*_f and π^*_g are sufficiently "close" to some linear functions $\tilde{\pi}_f$ and $\tilde{\pi}_g$ of the form $\tilde{\pi}_f(\boldsymbol{v}) = \langle \boldsymbol{v}, \boldsymbol{w}^* \rangle$ and $\tilde{\pi}_g(\boldsymbol{v}') = \langle \boldsymbol{v}', \boldsymbol{w}^* \otimes \boldsymbol{w}^* \rangle$ for an assignment $\boldsymbol{w}^*$, and

[16] More precisely, the honest PCP proof is a single linear function with which one can evaluate both π_f and π_g, but in this overview we think that the PCP proof consists of π_f and π_g for simplicity.

2. if π_f^* and π_g^* pass the first test, then verifies whether or not $\boldsymbol{w}^*$ is a satisfying assignment.

Here, for any $\delta \in [0, 1]$, we say that a function f is *δ-close* to a linear function $\hat{f}$ if the fraction of the domain on which f agrees with $\hat{f}$ is at least δ (i.e., if $\Pr[f(\boldsymbol{r}) = \hat{f}(\boldsymbol{r}) \mid \boldsymbol{r} \leftarrow D] \geq \delta$, where D is the domain of $f, \hat{f}$).

Concretely, the actual verifier of ALMSS linear PCP does the following three tests on the PCP proof (π_f^*, π_g^*):

1. **Linearity Test:** This test verifies whether or not π_f^* and π_g^* are δ-close to some linear functions for sufficiently large δ.

2. **Tensor-Product Test:** Under the assumption that π_f^* and π_g^* are δ-close to some linear functions $\tilde{\pi}_f$ and $\tilde{\pi}_g$ for sufficiently large δ, this test verifies whether or not $\tilde{\pi}_f, \tilde{\pi}_g$ are of the form $\tilde{\pi}_f(\boldsymbol{v}) = \langle \boldsymbol{v}, \boldsymbol{w}^* \rangle$ and $\tilde{\pi}_g(\boldsymbol{v}') = \langle \boldsymbol{v}', \boldsymbol{w}^* \otimes \boldsymbol{w}^* \rangle$ for an assignment $\boldsymbol{w}^*$.

3. **SAT Test:** Under that assumption that π_f^* and π_g^* are δ-close to linear functions $\tilde{\pi}_f(\boldsymbol{v}) = \langle \boldsymbol{v}, \boldsymbol{w}^* \rangle$ and $\tilde{\pi}_g(\boldsymbol{v}') = \langle \boldsymbol{v}', \boldsymbol{w}^* \otimes \boldsymbol{w}^* \rangle$ for sufficiently large δ and an assignment $\boldsymbol{w}^*$, this test verifies whether or not $\boldsymbol{w}^*$ is a satisfying assignment to Ψ.

These three tests are done in parallel, and repeated many times to decrease the soundness error. The detail of these three tests are described below.

Linearity Test. In Linearity Test, the verifier checks $\pi_f^*(\boldsymbol{r}_1) + \pi_f^*(\boldsymbol{r}_2) \stackrel{?}{=} \pi_f^*(\boldsymbol{r}_1 + \boldsymbol{r}_2)$ and $\pi_g^*(\boldsymbol{r}'_1) + \pi_g^*(\boldsymbol{r}'_2) \stackrel{?}{=} \pi_g^*(\boldsymbol{r}'_1 + \boldsymbol{r}'_2)$ for random $\boldsymbol{r}_1, \boldsymbol{r}_2 \in \mathbb{F}^N$ and $\boldsymbol{r}'_1, \boldsymbol{r}'_2 \in \mathbb{F}^{N^2}$. Clearly, if π_f^* and π_g^* are linear function, they pass this test with probability 1. Furthermore, somewhat unexpectedly, it is known that the converse is also true in the following sense: if π_f^* and π_g^* pass this test with probability $1 - \rho$ for any $\rho < 1/6$, they are $(1 - 2\rho)$-close to linear functions [BLR93, Gol17]. Thus, for any sufficiently large δ (say, $\delta = 0.999$), if the above test is repeated many times and still π_f^* and π_g^* pass all of them, they are δ-close to linear function with high probability.

Tensor-Product Test. As a warm-up, we first consider the case that π_f^* and π_g^* are guaranteed to be linear functions (rather than just being δ-close to them). Consider a test that checks $\pi_f^*(\boldsymbol{r}_1)\pi_f^*(\boldsymbol{r}_2) \stackrel{?}{=} \pi_g^*(\boldsymbol{r}_1 \otimes \boldsymbol{r}_1)$ for random $\boldsymbol{r}_1, \boldsymbol{r}_2 \in \mathbb{F}^N$. Let $\boldsymbol{u} \in \mathbb{F}^N, \boldsymbol{u}' \in \mathbb{F}^{N^2}$ be the coefficients such that $\pi_f^*(\boldsymbol{v}) = \langle \boldsymbol{v}, \boldsymbol{u} \rangle$ and $\pi_g^*(\boldsymbol{v}') = \langle \boldsymbol{v}', \boldsymbol{u}' \rangle$.

- If $\boldsymbol{u}' = \boldsymbol{u} \otimes \boldsymbol{u}$, we have that π_f^* and π_g^* pass the above test with probability 1 since we have

$$\pi_f^*(\boldsymbol{r}_1)\pi_f^*(\boldsymbol{r}_2) = \left(\sum_{i=1}^{N} u_i r_{1,i}\right)\left(\sum_{i=1}^{N} u_i r_{2,i}\right) = \sum_{1 \leq i,j \leq N} u_i u_j r_{1,i} r_{2,j} = \pi_g^*(\boldsymbol{r}_1 \otimes \boldsymbol{r}_1) \ .$$

- If $\boldsymbol{u}' \neq \boldsymbol{u} \otimes \boldsymbol{u}$, we can see that π_f^* and π_g^* fail to pass the above test with probability $2/|\mathbb{F}|$ as follows. Let M, M' be the matrices such that

$$\pi_f^*(\boldsymbol{r}_1)\pi_f^*(\boldsymbol{r}_2) = \boldsymbol{r}_1 M \boldsymbol{r}_2^T \qquad \text{and} \qquad \pi_g^*(\boldsymbol{r}_1 \otimes \boldsymbol{r}_1) = \boldsymbol{r}_1 M' \boldsymbol{r}_2^T$$

(that is, M, M' are the $N \times N$ matrices such that $m_{i,j} = u_i u_j$ and $m'_{i,j} = u'_{(i-1)N+j}$). If we have $\boldsymbol{u}' \neq \boldsymbol{u} \otimes \boldsymbol{u}$, we have $M \neq M'$, so we have $\boldsymbol{r}_1 M \neq \boldsymbol{r}_1 M'$ with probability at least $1 - 1/|\mathbb{F}|$ over the choice of $\boldsymbol{r}_1$. Furthermore, if we have $\boldsymbol{r}_1 M \neq \boldsymbol{r}_1 M'$, we have $\boldsymbol{r}_1 M \boldsymbol{r}_2^T \neq \boldsymbol{r}_1 M' \boldsymbol{r}_2^T$ with probability at least $1 - 1/|\mathbb{F}|$ over the choice of $\boldsymbol{r}_2$. Hence, if $\boldsymbol{u}' \neq \boldsymbol{u} \otimes \boldsymbol{u}$, we have $\boldsymbol{r}_1 M \boldsymbol{r}_2^T \neq \boldsymbol{r}_1 M' \boldsymbol{r}_2^T$ (and thus $\pi_f^*(\boldsymbol{r}_1)\pi_f^*(\boldsymbol{r}_2) \neq \pi_g^*(\boldsymbol{r}_1 \otimes \boldsymbol{r}_1)$) with probability at least $1 - 2/|\mathbb{F}|$ over the choice of $\boldsymbol{r}_1, \boldsymbol{r}_2$.

Therefore, if the above test is repeated many time and still π_f^* and π_g^* pass all of them, we have $\boldsymbol{u}' = \boldsymbol{u} \otimes \boldsymbol{u}$ (and thus there exists $\boldsymbol{w}^*$ such that $\pi_f^*(\boldsymbol{v}) = \langle \boldsymbol{v}, \boldsymbol{w}^* \rangle$ and $\pi_g^*(\boldsymbol{v}') = \langle \boldsymbol{v}', \boldsymbol{w}^* \otimes \boldsymbol{w}^* \rangle$) with high probability.

Now, we consider the actual case that π_f^* and π_g^* are just δ-close to some linear functions π_f^*, π_g^*. A key observation is that since δ is assumed to be sufficiently large, the verifier can approximately evaluate $\tilde{\pi}_f, \tilde{\pi}_g$ through π_f^*, π_g^* via "self-correction," namely the verifier can evaluate $\tilde{\pi}_f$ (resp., $\tilde{\pi}_g$) on any point $\boldsymbol{x} \in \mathbb{F}^N$ (resp., $\boldsymbol{x} \in \mathbb{F}^{N^2}$) with error probability $2(1-\delta)$ through the following simple probabilistic procedure.

Algorithm $\mathsf{Self\text{-}Correct}^{\pi_f^*, \pi_g^*}(\boldsymbol{x})$:

Choose random $\boldsymbol{r} \in \mathbb{F}^N$ (resp., $\boldsymbol{r} \in \mathbb{F}^{N^2}$) and output $\tilde{\pi}_f(\boldsymbol{x}+\boldsymbol{r}) - \tilde{\pi}_f(\boldsymbol{r})$ (resp., $\tilde{\pi}_g(\boldsymbol{x}+\boldsymbol{r}) - \tilde{\pi}_g(\boldsymbol{r})$).

Given this observation, in Tensor-Product Test the verifier applies the above warm-up test on $\tilde{\pi}_f, \tilde{\pi}_g$ by evaluating them through $\mathsf{Self\text{-}Correct}^{\pi_f^*, \pi_g^*}$. Since the values that the verifier obtains through $\mathsf{Self\text{-}Correct}^{\pi_f^*, \pi_g^*}$ are correct with high probability, it follows that if π_f^* and π_g^* pass this test, there exists $\boldsymbol{w}^*$ such that $\tilde{\pi}_f(\boldsymbol{v}) = \langle \boldsymbol{v}, \boldsymbol{w}^* \rangle$ and $\tilde{\pi}_g(\boldsymbol{v}') = \langle \boldsymbol{v}', \boldsymbol{w}^* \otimes \boldsymbol{w}^* \rangle$ with high probability.

SAT Test. Recall that, as observed at the beginning as a warm-up, when π_f^* and π_g^* are guaranteed to be of the form $\pi_f^*(\boldsymbol{v}) = \langle \boldsymbol{v}, \boldsymbol{w}^* \rangle$ and $\pi_g^*(\boldsymbol{v}') = \langle \boldsymbol{v}', \boldsymbol{w}^* \otimes \boldsymbol{w}^* \rangle$ for an assignment $\boldsymbol{w}^*$, there exits a test that verifies whether or not $\boldsymbol{w}^*$ is a satisfying assignment to Ψ. In SAT Test, assuming that π_f^* and π_g^* are δ-close to some linear functions $\tilde{\pi}_f, \tilde{\pi}_g$ such that $\pi_f^*(\boldsymbol{v}) = \langle \boldsymbol{v}, \boldsymbol{w}^* \rangle$ and $\pi_g^*(\boldsymbol{v}') = \langle \boldsymbol{v}', \boldsymbol{w}^* \otimes \boldsymbol{w}^* \rangle$ for an assignment $\boldsymbol{w}^*$, the verifier applies this warm-up tests on $\tilde{\pi}_f, \tilde{\pi}_g$ by evaluating them through $\mathsf{Self\text{-}Correct}^{\pi_f^*, \pi_g^*}$. Since the values that the verifier obtains through $\mathsf{Self\text{-}Correct}^{\pi_f^*, \pi_g^*}$ are correct with high probability, it follows that if π_f^* and π_g^* pass this test, $\boldsymbol{w}^*$ is a satisfying assignment to Ψ with high probability.

B Delegation Scheme based on Multiplicative Homomorphic Encryption

In this section, we explain how we prove Theorem 2 in the case of using multiplicative homomorphic encryption schemes over prime-order bilinear groups.

Construction. The construction is based on the one given in Section 10.3.1 for the case of additive homomorphic encryption schemes. In the following, the differences are highlighted by red.

- **Algorithm** $\mathsf{Gen}(1^\lambda, C)$

  1. Run $(Q, \mathsf{st}_V) \leftarrow \mathsf{PCP.V}_0(1^\lambda, C)$.
     Then, parse Q as $\{\boldsymbol{q}_i\}_{i \in [\kappa_V(\lambda)]}$, where $\boldsymbol{q}_i = (q_{i,1}, \ldots, q_{i,N'}) \in \mathbb{F}^{N'}$.

2. Choose a bilinear group (G, G_T, e) with order $|\mathbb{F}|$ and its generator g, and define $\mathsf{ct}_1, \ldots, \mathsf{ct}_{\kappa_{\max}(\lambda)}$ as follows.
   (a) Choose a random injective function $\tau : [\kappa_V(\lambda)] \to [\kappa_{\max}(\lambda)]$.
   (b) Define ct_i for each $i \in [\kappa_{\max}(\lambda)]$ by
   $$\mathsf{ct}_i \leftarrow \begin{cases} \mathsf{HE.Enc}(\mathsf{HE.pk}_i, g^{\boldsymbol{q}_{\tau^{-1}(i)}}) & (\text{if } \exists i' \in [\kappa_V(\lambda)] \text{ s.t. } \tau(i') = i) \\ \mathsf{HE.Enc}(\mathsf{HE.pk}_i, \mathbf{1}) & (\text{otherwise}) \end{cases},$$
   where $(\mathsf{HE.pk}_i, \mathsf{HE.sk}_i) \leftarrow \mathsf{HE.Gen}(1^\lambda)$, $g^{\boldsymbol{q}_{\tau^{-1}(i)}} := (g^{q_{\tau^{-1}(i),1}}, \ldots, g^{q_{\tau^{-1}(i),N'}})$, and $\mathbf{1} := (1, \ldots, 1) \in G^{N'}$.
3. Output $\mathsf{pk} := (\mathsf{ct}_1, \ldots, \mathsf{ct}_{\kappa_{\max}(\lambda)})$ and $\mathsf{sk} := (\mathsf{st}_V, \tau, \{\mathsf{HE.sk}_i\}_{i \in [\kappa_{\max}(\lambda)]}, (G, G_T, e), g)$.

- **Algorithm** $\mathsf{Prove}(\mathsf{pk}, C, \boldsymbol{x}, \boldsymbol{y})$
  1. Run $\pi \leftarrow \mathsf{PCP.P}(C, \boldsymbol{x}, \boldsymbol{y})$.
     Let $d_1, \ldots, d_{N'} \in \mathbb{F}$ be the elements such that $\pi(z) = \sum_{i \in [N']} d_i z_i$.
  2. Parse pk as $(\mathsf{ct}_1, \ldots, \mathsf{ct}_{\kappa_{\max}(\lambda)})$, where $\mathsf{ct}_i = (\mathsf{ct}_{i,1}, \ldots, \mathsf{ct}_{i,N'})$.
     Then, perform homomorphic operation to obtain
     $$\tilde{\mathsf{ct}}_i := \prod_{j \in [N']} \mathsf{ct}_{i,j}^{d_j}$$
     for every $i \in [\kappa_{\max}(\lambda)]$.
  3. Output $\mathsf{pr} := (\tilde{\mathsf{ct}}_1, \ldots, \tilde{\mathsf{ct}}_{\kappa_{\max}(\lambda)})$.

- **Algorithm** $\mathsf{Verify}(\mathsf{sk}, \boldsymbol{x}, \boldsymbol{y}, \mathsf{pr})$
  1. Parse sk as $(\mathsf{st}_V, \tau, \{\mathsf{HE.sk}_i\}_{i \in [\kappa_{\max}(\lambda)]}, (G, G_T, e), g)$, and pr as $(\tilde{\mathsf{ct}}_1, \ldots, \tilde{\mathsf{ct}}_{\kappa_{\max}(\lambda)})$
     Then, run $a_i := \mathsf{HE.Dec}(\mathsf{HE.sk}_{\tau(i)}, \tilde{\mathsf{ct}}_{\tau(i)})$ for every $i \in [\kappa_V(\lambda)]$.
  2. Output $b := \mathsf{PCP.V}'_1(\mathsf{st}_V, \boldsymbol{x}, \boldsymbol{y}, \{a_i\}_{i \in [\kappa_V(\lambda)]})$, where $\mathsf{PCP.V}'_1$ is an algorithm that runs $\mathsf{PCP.V}_1$ in the exponent of g by using the bilinear map e.

Security Analysis. The analysis is also based on the one given in Section 10.3.2 for the case of additive homomorphic encryption schemes. That is, given any successful cheating PPT adversary against the above scheme, we obtain a cheating PCP prover $\mathsf{PCP.P}^*$, and show that it successfully fools the PCP verifier as well as that it is $\kappa_{\max}$-wise no-signaling.

The problem is that if we obtain the PCP prover $\mathsf{PCP.P}^*$ in exactly the same way as in Section 10.3.2, $\mathsf{PCP.P}^*$ runs in super-polynomial time since the PCP answers in the delegation scheme are now encoded in the exponent of g and thus $\mathsf{PCP.P}^*$ need to solve the discrete-logarithm problem to obtain the PCP answers. This is problematic since, if $\mathsf{PCP.P}^*$ runs in super-polynomial time, we can no longer show the no-signaling property of $\mathsf{PCP.P}^*$ under the CPA-security of HE, which holds only against PPT adversaries.

To overcome this problem, we modify our PCP system so that the prover returns the PCP answers in the exponent of a generator of a bilinear group (which is chosen by the verifier as a public parameter), and the verifier runs the verification algorithm in the exponent by using the bilinear map (recall that the verification algorithm of our (original) PCP system only checks quadratic equations on the PCP answers). It is easy to see that if this modified PCP system is sound against $\kappa_{\max}$-wise no-signaling cheating provers, we can prove the soundness of the above delegating scheme as in Section 10.3.2. Furthermore, it can be verified by inspection that the analysis of our original PCP system (Section 5 to Section 9) can be straightforwardly modified so that it works for the above modified PCP system. (A key point is every event that we consider in the analysis can be efficiently checked by quadratic tests on PCP answers.)