Electronic Colloquium on Computational Complexity, Report No. 73 (2016)

Improved concrete efficiency and security analysis of Reed-Solomon PCPPs

Eli Ben-Sasson Iddo Ben-Tov Ariel Gabizon Michael Riabzev

May 7, 2016

Abstract

A *Probabilistically Checkable Proof of Proximity* (PCPP) for a linear code C, enables to determine very efficiently if a long input x, given as an oracle, belongs to C or is far from C.

PCPPs are often a central component of constructions of *Probabilistically Checkable Proofs* (PCP)s [Babai et al. FOCS 90; Arora et al. JACM 98]; which in turn can be used to construct asymptotically efficient cryptographic zero knowledge arguments of membership in any language $L \in \mathbf{NEXP}$, with minimal communication complexity and computational effort on behalf of both prover and verifier [Babai et al. STOC 91; Kilian STOC '92; Micali SICOMP '00]. Though PCP constructions are *asymptotically* efficient, it is still far from clear how well they will perform in practice on concrete input sizes. This issue motivated the work of [Ben-Sasson et. al. STOC '13]. As in [BCGT13], to explore this question, we continue to investigate how well the PCPP for Reed-Solomon (RS) codes of [BS08] - the "heavy" component in the PCP construction of [BS08] - behaves for concrete input lengths. The crucial parameter to understand is the *soundness* of the PCPP, which is the probability an x far from C gets rejected. The paper contains three contributions:

1. Improved soundness analysis of a new variant of the Reed-Solomon (RS) PCP of Proximity (PCPP) verifier of [BS08]. This verifier and its analysis reduce the *concrete efficiency threshold* of RS PCPPs, as defined by [BCGT13], from its previous state of the art (2^{43}) there to 2^{23} here. Informally, the concrete efficiency threshold is a measure that abstracts the "smallest input length for which the PCPP is useful"; thus, reducing it will hopefully push PCP constructions a bit closer to practice. Additional improvements are reported for input lengths in the range 2^{23}—2^{35} that we view as interesting in practice.
2. We initiate the study of the *security* of PCPP systems, in which soundness is measured only with respect to a set of known efficient (polynomial-time) "pseudo-provers", or "attacks", attempting to prove "$x \in C$" when x is far from C. As a first step in this direction we introduce a pair of natural attacks on the PCPP system of [BS08]; and show that for most inputs that are not codewords the soundness (i.e., rejection probability) of both attacks is significantly higher (better) than the new unconditional worst-case bounds reported above. We conjecture that the same should be true for any x far from the code, and pose this as an interesting open problem, with significance to the "real-word practicality" of PCPs.

   In particular, for attaining the same rejection probability against these attacks on most inputs x, the PCPP verifier need only read approximately a 2^{-8}-fraction of the amount of field elements he needs to read according to the result of item 1, while maintaining the same proof length. Consequently, the concrete efficiency threshold, when defined analogously in this setting, decreases from 2^{23} to 2^{17}.
3. To further reduce the concrete efficiency threshold we consider the *Interactive Oracle Proofs of Proximity* (IOPP) model [BSCS16, RRR16, BCG$^+$16], in which [BCG$^+$16] show proof length can be significantly reduced with no loss in soundness, at the price of increased interaction. We show that the IOPP model gives significant improvements for practical ranges of parameters. In particular, applying the security analysis above to this new interactive model, the resulting proof length is cut by a factor of approximately $8\times$ while verifier query complexity is cut down by a factor of approximately $2^{10}\times$ in comparison to item 1.

We hope this work will bring PCPs closer to practice and popularize the quest for better concrete PCP soundness within the theoretical computer science research community.

ISSN 1433-8092

Contents

1 Introduction

The study of the *soundness* of PCP systems — the minimal probability of rejecting a pseudo-proof of false statements — is a cornerstone of modern computational complexity. Motivated by applications to the theory of inapproximability, as initiated by [FGL$^+$96], many of these results attempt to minimize the query complexity of the proof while maximizing soundness (cf. [Hås01, Din07, MR08] for notable examples). Motivated by the cryptographic applications of PCPs to sublinear time verification [BFL90, BFLS91] and to the construction of succinct Computationally Sound (CS) proofs [Kil92, Mic00], a different approach suggested to minimize a joint function of query complexity and proof-blowup [PS94, HS00, GS06, BSVW03, BGH$^+$06, BS08, Mie08] and furthermore consider such an optimization for "concrete" and non-asymptotic input lengths [BCGT13]. The rationale here is that the "concrete efficiency threshold"

of PCP systems could help drive and focus research towards a better understanding of PCP soundness for small input lengths, ones that arise in practical computations.

Indeed, over the past few years there has been considerable work on implementations of *succinct verification of correct program execution*. Roughly speaking, this is the task of a prover P running a program T on input x for t steps to obtain output $T(x) = z$; and then convincing a verifier V that indeed $T(x) = z$ in time $s \ll t$. Babai et. al [BFL90, BFLS91] showed, using PCP constructions with efficient verifiers, that this can always be done in time $s = \text{polylog}(t)$. In terms of published works on implementation, there are two main routes we are aware of:

1. Implementations using "linear PCPs" [1] [IKO07, BCI$^+$13, Gro09, GS09, GOS06, Gro10, Lip12, GGPR13]; examples of implementations include Setty et al. [SMBW12, SVP$^+$12, SBW11], Vu et al. [VSBW13], Parno et. al [PGHR13] and Ben-Sasson et. al [BCG$^+$13].

2. Implementations using sum-check protocols [KR08, GKR15]; an implementation is reported, for example, in Cormode et. al [CMT12].

Naturally, these implementations have used ideas arising in PCP constructions. However, so far they have avoided using the "heavy PCP machinery", e.g., probabilistically checkable proofs of proximity (PCPPs) (see Definition 1.1) and proof composition, perhaps because it is not sufficiently clear whether these tools are efficient enough in practice. This avoidance comes at a cost: Implementations of the first type require a trusted setup phase that involves a cryptographic trapdoor that can be recovered by the party (or parties) setting up the system; this trapdoor allows one to efficiently forge pseudo-proofs of false statements that will nevertheless be accepted by V. The second approach is currently only efficient when the computation can be modeled as a poly-log depth circuit; thus, it is useful only for succinct verification of parallelizable programs. It is therefore of significance to understand how well asymptotically efficient PCP constructions perform when they are "scaled down" to input lengths arising in practice. As a special case, it is important to understand how well PCPPs for Reed-Solomon codes (denoted RS-PCPPs henceforth) behave on practical input lengths, because (i) RS-PCPPs are precisely the (only) heavy component in the PCP construction of [BS08] and (ii) the [BS08] construction seems to be amenable to efficient implementation in practice.[2]

1.1 PCPPs and their concrete efficiency

Before defining probabilistically checkable proofs of proximity (PCPPs), we give some intuition as to why they arise in PCP constructions, specifically that of [BS08]. Suppose we wish to construct a proof that convinces a verifier V that some string $w \in \mathbb{F}^k$ is a satisfying assignment to a constraint satisfaction problem, while querying only a few locations of the proof. Methods involving arithmetization, starting from Lund et. al [LFKN92] and Babai et. al [BFL90] (cf. [AS98, ALM$^+$98, PS94]) showed this is possible if V is given access to a *low degree extension* of w; or more generally, to an encoding of w by some error correcting code C. This leads to proof systems in which V is given oracle access to a long word x, that is *supposed* to be a codeword of C. Under the assumption that indeed $x \in C$, the proof system can be shown to work well; and the "heavy" work left for V is to verify that indeed $x \in C$, or at least that x is close to a codeword of C. If V was given access only to x, this would be possible with few queries only if C was a locally testable code (LTC). This comes at a price because known constructions of "PCP-friendly" LTCs are more complex and have lower rate than a code like Reed-Solomon (RS), and some of the more efficient LTCs (in terms of rate), like [KRS15, KMRS15] are not known to "PCP-friendly", i.e., it is not clear how to build PCPs in a black-box manner using these codes. In particular, if we want to use a code that corresponds to an actual low degree extension, as is necessary in some constructions, we need to use a multivariate Reed-Muller code, which has poor rate compared to RS. This low rate would translate into a longer proof. To enable using a simple code of high rate like RS in a PCP, the prover will add an auxiliary proof y that $x \in C$. Miraculously, even if C has no local structure, V can verify that x is at least close to C by querying a few locations in both x and y. This auxiliary proof is what is commonly called a PCPP[3] for the code C [BGH$^+$06, DR04].

[1] The connection between linear PCPs and efficient proofs was clarified and formalized by Bitansky et. al [BCI$^+$13], and is implicit in some of the mentioned works.

[2] In fact, such an implementation will soon be published by the Succinct Computational Integrity and Privacy Research (SCIPR) lab (sciprlab.org).

[3] However, it is technically convenient for us to define the PCPP as a mapping that outputs *both* x and y.

Below we use the standard notation that an $[n, k, d]_{\mathbb{F}}$-code C is a k-dimensional subspace of $\mathbb{F}^n$ in which no two distinct members have hamming distance smaller than d.

Definition 1.1 (PCPP for a code C)**.** *Fix integers $A, Q \in \mathbb{N}$. Let C be an $[n = n(C), k = k(C), d = d(C)]_{\mathbb{F}}$-code. An (A, Q)-*PCPP system *$\mathcal{S}$ for C is a pair $\mathcal{S} = (P, V)$, where*

- *P is a* systematic *mapping $P : C \to \mathbb{F}^A$.*

  That is, for any $x \in C$, $P(x) = (x, y)$ for some $y \in \mathbb{F}^{A-n}$.

- *V is a Q-*local *randomized mapping $V : \mathbb{F}^A \to \{\mathsf{accept}, \mathsf{reject}\}$. That is, after choosing its internal randomness, $V(z)$ always depends on at most Q indices of $z \in \mathbb{F}^A$.*

Such that

- *(Completeness) For any $x \in C$, $V(P(x)) =$ accept with probability one.*
- *(Soundness) For any $x \in \mathbb{F}^n$ such that $\Delta(x, C) \geq d/3$, and any $y \in \mathbb{F}^{A-n}$, $V((x, y)) =$ reject with probability at least $1/2$.*

Defining concrete efficiency of PCPPs [BCGT13] introduced the notion of the *concrete efficiency threshold of a PCPP*, which is central to our results.

Let us begin with some intuition before the formal definition. Recall that our goal is to have a system where P convinces V that a long string $x \in \mathbb{F}^n$ to which V has oracle access, is close to a codeword of C. Typically, x is an encoding of a string $w \in \mathbb{F}^k$ that is a satisfying assignment to some constraint satisfaction problem ϕ. Consider first the following "trivial" proof system: P sends to V the message $w \in \mathbb{F}^k$ whose encoding is x. V checks that the encoding of w is indeed x. Let us look for a simple way to measure the efficiency, or cost, of this proof system. We ignore computational costs and only count the number A of field elements P writes and the number Q of field elements V reads; i.e., we do not count the computations done by P and V. This is certainly a lower bound on the complexity of the system, and [BCGT13] argue (see Remark 2.7 there) that it is a reasonable approximation as all calculations in the [BS08] PCPP are quasilinear with small constants in A and Q. [BCGT13] suggest taking the product of these two numbers as a measure of the cost of a proof system (See Remark 1.3 following Definition 1.2). Using this measure, the cost of the trivial proof system is k^2. They go on to define a proof system as "useful" for a given input length if its cost under this measure is at most $k^2/2$. In other words, a proof system is useful if it is at least twice more efficient than the trivial system (according to the chosen cost function). The *concrete efficiency threshold* of the system is thus defined as the smallest input length where it is useful. We proceed with the formal definitions.

Definition 1.2 (Concrete efficiency threshold of a PCPP)**.** *Using the notation of Definition 1.1, we define the* cost *of an (A, Q)-PCPP system for a code C to be $A \cdot Q$. We say the system is* efficient *if the cost $A \cdot Q \leq k(C)^2/2$.*

*Fix an ensemble of linear codes $\mathcal{C} = \left\{C \subseteq \mathbb{F}^{n(C)}\right\}$, and functions $A, Q : \mathbb{N} \to \mathbb{N}$. An (A, Q)-*PCPP system for *$\mathcal{C}$ is an ensemble of PCPP systems $\mathcal{S} = \{\mathcal{S}_C | C \in \mathcal{C}\}$ where $\mathcal{S}_C$ is an $(A(k(C)), Q(k(C)))$-PCPP system for C.*

The concrete efficiency threshold of *$\mathcal{S}$ is the smallest integer $\mathbf{k}$ such that for any $C \in \mathcal{C}$ of dimension $k(C) \geq \mathbf{k}$, $\mathcal{S}_C$ is efficient.*

Remark 1.3 (On the particular choice of cost function)**.**

- *One may question why specifically the* product *of A and Q should be defined as the cost function. Indeed, [BCGT13] suggests looking at general polynomial cost functions $C(x, y)$, and defining $C(A, Q)$ as the cost of the system. Naturally, one may desire to choose a cost function giving larger weight to A or Q depending on whether they want to focus more on prover or verifier complexity respectively. However, [BCGT13] end up presenting their results using $C(x, y) = x \cdot y$; thus, for simplicity, and to enable comparison with their results, we focus solely on the product cost function.*
- *In similar vein, defining a system to be efficient\useful when it is* twice *as efficient as the trivial system, rather than some other constant (or function), is an arbitrary choice made in [BCGT13] that we stick with.*

We now proceed to describe our results.

log(message length)	log(codeword length)	log(proof length)	log(# of queries)
23	26	31	16.5
24	27	32	16.75
25	28	33	17
26	29	34	17.25
27	30	35	17.5
28	31	36	17.75
29	32	37	18
30	33	38	18.25
31	34	39	18.5
32	35	40	18.75
33	36	41	19
34	37	42	19.25
35	38	43	19.5

Table 1: Instantiations of our analysis of a depth two test (see Subsection 4.1.4). All numbers are logs in base two of the described quantity. The first column describes the length (in field elements) of the message w to be encoded into a word $x \in \mathsf{RS}_L[3]$. The second column is the length of x. The third column is the length of x together with the proof y that $x \in \mathsf{RS}_L[3]$. The fourth is the number of field elements a verifier needs to read to be convinced with probability $1/2$ that x is at least $\frac{7}{24}$-close to $\mathsf{RS}_L[3]$.

1.2 First contribution — Improved concrete efficiency threshold

Before presenting our results we need to formally define RS codes.

Reed-Solomon codes on subspaces Fix a positive integer η. Let $\mathbb{F}$ be a field of characteristic two. Fix an $\mathbb{F}_2$-subspace $L \subseteq \mathbb{F}$. We define $\mathsf{RS}_L[\eta]$ to be the set of functions $g : L \to \mathbb{F}$ that are evaluations of a univariate polynomial $p \in \mathbb{F}[Z]$ of degree at most $2^{-\eta} \cdot |L| - 1$.

We define $\mathrm{RS}_{\mathrm{a}}[\eta]$ to be the ensemble of Reed-Solomn codes over some subspace $L \subseteq \mathbb{F}$, of degree at most $2^{-\eta} \cdot |L| - 1$.

That is,

$$\mathrm{RS}_{\mathrm{a}}[\eta] \triangleq \{\mathsf{RS}_L[\eta] | L \subseteq \mathbb{F} \text{ is a subspace}\}.$$

Note that in this notation the fixed field $\mathbb{F}$ is implicit.

In this terminology, we show the following.

Theorem 1.4. *Fix any field $\mathbb{F}$ of characteristic two. There is a PCPP system for $\mathrm{RS}_{\mathrm{a}}[3]$ with concrete efficiency threshold at most 2^{23}.*

Theorem 1.4 improves the bound of 2^{43} from [BCGT13]. It will follow from the first item of

Theorem 1.5. *Fix any field $\mathbb{F}$ of characteristic two. There is a*

1. *$(2^{\ell+5}, 2^{\ell/2+5.5})$-PCPP system for $\mathrm{RS}_{\mathrm{a}}[3]$,*
2. *$(2^{\ell+8}, 2^{\ell/4+10.75})$-PCPP system for $\mathrm{RS}_{\mathrm{a}}[3]$,*

where 2^{ℓ} denotes the message length $k(C)$.

Table 1 shows instantiations of the second item of the above theorem for input lengths up to 2^{35}. We note again that filling the same table with the last column being the number of queries needed according to the analysis of [BCGT13], would give a trivial result where the verifier needs to read more locations than the original message length.

1.3 Second contribution — Introducing the analysis of PCPP security rather than soundness

We informally use the term *pseudo-proof* here to refer to an attempted proof to show that $x \in C$, when in fact x is far from C.

The *soundness* s of a PCPP system is defined as the smallest (i.e., worst) rejection probability of a pseudo-proof; and *soundness error* is $1 - s$. According to this definition, soundness is measured with respect to all possible pseudo-proofs, even those that are intractable to find. Thus far, all analyses of PCPP systems have used *mathematically proven lower bounds* on soundness, even when these bounds are not known to be tight. In particular, for the RS PCPP of [BS08], we are not aware of pseudo-proofs, including ones that are not efficiently constructible, whose rejection probability is close to being as "bad", i.e., low, as the proven lower bound on s.

This may be one reason why the known bounds on PCPP soundness (including the improvements of this paper) are currently far away from guaranteeing small soundness error (say, 2^{-60}) while maintain reasonable verifier and prover efficiency.

Motivated by the need for small soundness error "in the real world", we initiate a "practical security/cryptanalytic" approach to investigate how well PCPPs will perform in practice, as is commonly done in cryptanalysis and security research [BS93, Mat93]. In this research approach one suggests *concrete attacks* on various cryptographic primitives, and defines their security level according to how well they withstand these currently known attacks. In the context of PCPPs, if know known method for generating a pseudo-proof succeeds to fool the verifier with probability greater than $2^{-\lambda}$, we could say that "P has security level λ".

As a first step in the study of PCPP security, we suggest two natural attacks on the RS-PCPP of [BS08], that we call the "row-compliant attack" and "column-compliant attack". (See overview in Section 3.2 and Section 6). Ideally, we would want to show that for *any* $g : L \to \mathbb{F}$ that is far from $\mathsf{RS}_L[\eta]$ these pseudo-proofs would be rejected with much higher probability than what is known for a general pseudo-proof. We fall short of this, and pose this as an interesting open question. However, we manage to show that for almost all $g : L \to \mathbb{F}$ these attacks fail with high probability.

When discussing security for concrete, non-asymptotic, input lengths like 2^{30}, one faces certain definitional challenges; for instances, there exists a Turing machine that has encoded in its transition function the optimal pseudo-proof for each input of size 2^{30}, i.e., the pseudo-proof achieving minimal rejection probability; thus, care is needed in defining what security means to avoid having it collapse to soundness. To deal with this issue (and others) we define in Appendix B the notion of *feasible soundness* that abstracts "security" for concrete input lengths; we believe this definitional approach may be useful in other studies of concrete (non-asymptotic) efficiency and security.

Remark 1.6 (Why study PCP security?)**.** *A reasonable question to ask is why should one engage in a security analysis of PCP systems that inevitably leads to conjectural results, when one can apply the "bullet-proof" method of soundness analysis that relies on pure mathematical proofs? We give three answers:*

- *As mentioned, analyzing security may lead to more efficient PCP systems (say, with shorter proofs) than what unconditional soundness allows for, given that we want a reasonable security level.*
- *PCPs used in practical cryptographic applications (e.g., [Kil92, Mic00]) use cryptographic primitives (like SHA256) that have no proven soundness anyways; in other words, avoiding conditional security analysis of PCPs will not remove the need for cryptographic assumptions (backed up by analysis of known attacks) in PCP-based systems.*
- *Thinking about attacks on PCPs and PCPPs may lead to new insights and interesting theoretical questions. This already seems to be the case for the two simple attacks we analyze, that quickly lead to algebraic geometry questions about distinguishing rational functions from low-degree polynomials (will be discussed in subsequent work).*

Related work Kalai and Raz introduced in [KR09] the notion of a probabilistically checkable argument (PCA). In the PCA model soundness is relaxed to hold only against computationally-bounded adversaries, and the verifier, which is designated, uses cryptographic primitives (namely, a computational private information retrieval scheme) to make the interactive process more efficient. The emphasis of our security analysis is different. Rather than proving security against all computationally bounded adversaries under cryptographic assumptions, we focus on analyzing a few natural computationally bounded attacks without cryptographic assumptions.

1.4 Third contribution — Reducing proof length by adding interaction in the style of [BCG$^+$16]

The main factor driving up the cost of a PCPP system (cf. Definition 1.2) is proof length (denoted A there). All efficient PCPPs known to date use some form of *proof composition* originally suggested by [AS98]. Proofs written this way are "composed" of several "layers". The topmost layer π_0 is a string that contains part of the proof. A "view" of π_0 is a restriction of it to a small number of coordinates. Now, for each "interesting" view v the composed proof will contain a specific sub-proof π_v. This process is now repeated recursively, generating sub-sub-proofs for views of π_v etc. The bulk of the final PCPP is due to the many leaves of this recursive process.

However, proof verification is extremely efficient and the verifier inspects only a negligible fraction of views, sub-views, etc. When applying composition to PCPs of proximity (PCPP) as suggested in [BGH$^+$06], the verifier loses no soundness if the prover writes down only those subproofs that are eventually queried (cf. Lemma 7.2).

To preserve verifier-efficiency (small query complexity) while improving prover-efficiency (shorter proof), [BCG$^+$16] use Interactive Oracles Proofs of Proximity (IOPP); whose definition is based on the concept of an Interactive Oracle Proof [BSCS16] [4] An IOPP is similar to Definition 1.1 of a PCPP, but in an IOPP P writes down the proof as part of an interactive process with V. However, contrary to a standard interactive protocol and similarly to a PCPP, V does not pay the cost of receiving and storing P's messages, but only the cost of accessing the locations he actually reads.

We show that, for concrete instance sizes, the proof length drops dramatically in this model because the RS-PCPP verifier works recursively, each step deciding which subproof to query. In the IOPP setting P can generate only those subproofs that V is actually going to query. See Section 7 for details.

Remark 1.7. *Though the idea of V receiving multiple oracles is intuitively clear, formalizing this notion proves tricky in existing models like IP, MIP, PCP and IPCP. In Appendix A we present a formal model to describe IOPPs using the concept of a* notary. *The notary is a third party (in addition to Prover and Verifier) that records prover-messages on behalf of the verifier V, and V later queries these messages at a small number of locations.*

1.5 Illustration of results

Recall again that the objective of a PCPP prover on input x is to convince V that x is close to an element of some code C. Our results regard Reed-Solomon codes of the form $\mathsf{RS}_L[\eta]$ as defined in Section 1.2. Let us define an (A, Q)-*PCPP system with soundness error* $2^{-\lambda}$ identically to Definition 1.1 but with the rejection probability of V being at least $1 - 2^{-\lambda}$ for "large" λ, rather than $1/2$ ($\lambda = 1$); we call λ the *security parameter*. Recall we defined the cost of such a system to be $A \cdot Q$. To illustrate our different results, we compare in Figure 1.1 the cost of the proof system for $\mathsf{RS}_L[3]$ as a function of the security parameter λ on messages of size 2^{35}. We note that the previous analysis of [BCGT13] does not enable improving upon than the trivial system for such message size. The following four cases are displayed[5]:

1. The "trivial" system (black dotted line), where prover simply sends the message being encoded, and the verifier reads all of it.

2. The improved (unconditional) soundness analysis of Theorem 1.5, described in Section 1.2 (red broken line);

3. The security analysis described in Section 1.3 applied to random functions, assuming the prover can only use the row and column-compliant attacks; cf. Corollary 6.10 (purple solid line)

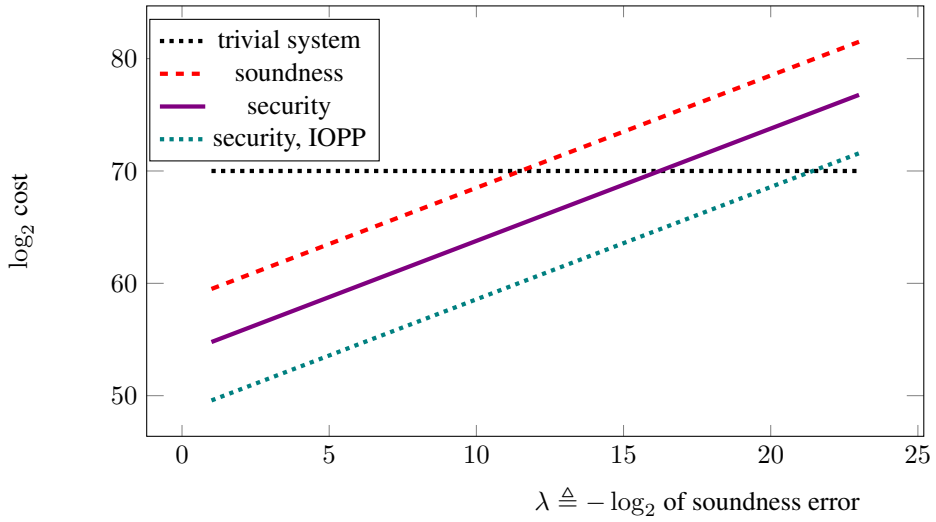
4. The security analysis described in Section 1.4 in the IOPP model, applied to random functions, assuming the prover can only use the row and column-compliant attacks. Cf. Lemma 7.2 (light blue dotted line).

The plot emphasizes how assuming the prover is limited to natural attacks, and using the interactive IOPP model, helps us get smaller soundness error while maintaining cost that is lower than the trivial system; and this is done for "reasonable" input length, where "asymptotics have not completely kicked in".

[4] The concepts of IOPs and IOPPs also appeared independently in [RRR16], where they are called Probabilistically Checkable Interactive Proofs, and used there to construct doubly efficient interactive protocols.

[5] An excel sheet where different cost functions $C(A, Q)$ may be experimented with is available at request.

Figure 1.1: A comparison of cost as a function of security parameter λ for messages of length 2^{35} with $\mathsf{RS}_L[3]$. Notice that the break-even point for soundness (where cost is no better than that of the trivial system) is at security parameter $\lambda \approx 10$, whereas for security the break-even point is ≈ 16 and for IOPP it is highest (best), approximately 22.



2 Preliminaries

We present some notation and basic facts that will be used throughout the paper. $\mathbb{F}$ always denotes a finite field of characteristic two. Recall that $\mathbb{F}$ is a vector space over $\mathbb{F}_2$. When we refer to a *subspace* $L \subseteq \mathbb{F}$ we always mean *an* $\mathbb{F}_2$*-subspace of* $\mathbb{F}$ *when viewed as such a vector space*.

Product Sets Fix sets A, B, an element $a \in A$ and subset $S \subseteq B$. We denote by (a, S) or $a \times S$ the subset $\{(a, b) | b \in S\}$ of $A \times B$.

Functions and Polynomials For polynomials $F, G \in \mathbb{F}[Z]$ we will write $F = G$ or $F(Z) = G(Z)$ to denote equality of formal polynomials. For a subset $S \subseteq \mathbb{F}$ and polynomial $F \in \mathbb{F}[Z]$ we denote by $F|_S$ the function defined by $F|_S(x) \triangleq F(x)$. For functions $f, g : S \to \mathbb{F}$ we write $f = g$ to mean f and g are identical on S. For a domain $S' \subseteq S$ we use the notation $\Delta_{S'}(f, g)$ to denote the fractional disagreement of f and g on S'. That is,

$$\Delta_{S'}(f, g) \triangleq \Pr_{x \leftarrow S'}(f(x) \neq g(x)).$$

We also use the notation $\Delta(f, g) \triangleq \Delta_S(f, g)$. For $F \in \mathbb{F}[Z]$ and a function $g : S \to \mathbb{F}$ we use the abbreviated notations $F = g$ to mean $F|_S = g$, and $\Delta_S(F, g)$ to mean $\Delta_S(F|_S, g)$.

When discussing bivariate polynomials $Q(X, Y) \in \mathbb{F}[X, Y]$, we say Q *has degree* (d, e) if $\deg_X(Q) \leq d$ and $\deg_Y(Q) \leq e$. For a bivariate function $f : A \times B \to \mathbb{F}, \alpha \in A$ and $\beta \in B$; we denote $f(X, \beta) \triangleq f|_{X \times \beta}$ and $f(\alpha, Y) \triangleq f|_{\alpha \times Y}$ We use similar notation for $Q(X, Y) \in \mathbb{F}[X, Y]$. For example, $Q(X, \gamma)$ denotes the univariate polynomial obtained by substituting $Y = \beta$ in Q. Note that $Q(X, \beta) = f(X, \beta)$ means f and Q identify as functions on $A \times \beta$.

Subspace Polynomials For a subspace $L \subseteq \mathbb{F}$, we denote by q_L the *subspace polynomial of* L, defined as,

$$q_L(Z) \triangleq \prod_{v \in L}(Z - v). \tag{2.1}$$

These polynomials are well-studied. We mention the essential relevant properties for our results. Suppose that L is the direct sum of the linear spaces L_0 and L_1. Then

- $q_{L_0}|_L$ is $|L_0|$-regular.
- $q_{L_0}(L) = q_{L_0}(L_1) = L'_1$ for a subspace $L'_1 \subseteq \mathbb{F}$ with $|L'_1| = |L_1|$. In particular, q_{L_0} is injective on L_1.

3 Overview of the main contributions

3.1 An overview of first Contribution — the Ben-Sasson-Sudan PCPP for RS and our improved analysis

Fix a function $g : L \to \mathbb{F}$. The purpose of the Ben-Sasson-Sudan PCPP is to convince V that g is (close to) an element of $\mathsf{RS}_L[\eta]$, i.e., that it is an evaluation of a polynomial of degree at most $d \triangleq 2^{-\eta} \cdot |L| - 1$. For this overview, let us assume that V is willing to read $O(\sqrt{d})$ entries of g and an auxiliary proof $\pi(g)$. Observe first, that reading any $t < d$ entries of g gives V no information on whether $g \in \mathsf{RS}_L[\eta]$; as a degree d univariate polynomial could be interpolated to match any t values. The basic idea is to "embed" g into a *bivariate* polynomial Q of degree $O(\sqrt{d})$. The proof $\pi(g)$ consists of values of Q on a carefully chosen set. For this overview we use the term degree of a bivariate polynomial to mean individual degree.

The embedding of g works by choosing a certain univariate polynomial $q(Z)$ of degree $O(\sqrt{d})$, and then constructing $Q \in \mathbb{F}[X, Y]$ of (individual) degree[6] $\ell = O(\sqrt{d})$, such that the values of g on L correspond to the values of Q on the "curve" $T(L) \triangleq \{(z, q(z)|z \in L\}$ (See Claim 4.2 and Definition 4.3). q is chosen such that $T(L)$ will have the following convenient properties.

- There are only $O(\sqrt{d})$ different "y values" in $T(L)$; that is $|q(L)| = |\{q(z)|z \in L\}| = O(\sqrt{d})$. Let $L_1 \subset L$ be a set of size $|q(L)|$ such that $q(L_1) = q(L)$.
- All "row restrictions" have the same size in $T(L)$; that is, for any $\beta \in L_1$ the set $\{(z, q(\beta)|z \in L, q(z) = q(\beta)\} \subset T(L)$ is of size $\frac{|T(L)|}{|q(L)|}$.

Checking whether g has degree d is reduced by this embedding to checking whether a bivariate function f on $T(L)$ has degree ℓ; that is, whether $f|_{T(L)}$ identifies with a bivariate polynomial Q of degree ℓ. This seems more doable as bivariate polynomials have more local structure. For example, their restriction to any "row" or "column" should satisfy the same degree bound as the polynomial itself. That is, if Q has degree $\leq \ell$, the univariate polynomials $Q(X, \beta)$ and $Q(\alpha, Y)$ also have degree at most ℓ. Polishchuk and Spielman[PS94] proved a strong converse to this: *Fix a table of a function $f(X, Y)$ on a product set $S = A \times B \subseteq \mathbb{F}^2$, where $|A|, |B| \geq 4\ell$. If many rows $f(X, \gamma)$ or many columns $f(\alpha, Y)$ are far from degree ℓ (as univariate polynomials), then f is far from any bivariate polynomial of (individual) degree ℓ.*

This is not immediately helpful, as the set $T(L)$ on which g's values correspond to values of Q, is quite far from a product set - for any value $z \in L$ of the first coordinate, there is only one value $(z, q(z)) \in T(L)$.

The proof $\pi(g)$ consists of the values of a function f - that is supposed to be Q - on a product set $S(L) \triangleq A \times q(L)$ where $|A| = O(\ell)$. Note that $|\pi(g)| = O(d)$. We think of $(g, \pi(g))$ as jointly describing a bivariate function f on the domain[7] $\Omega(L) \triangleq S(L) \cup T(L)$.

The sets $S(L)$ - on which we *know* how to test closeness of f to degree ℓ, and $T(L)$- on which we *desire* to test closeness to degree ℓ have almost no intersection. What ties them together are the *extended rows*

$$\mathrm{extrow}_\beta \triangleq (L, q(\beta)) \cap \Omega(L),$$

for $\beta \in L_1$. $\Omega(L)$ is precisely the disjoint union of $|q(L)|$ such rows. Any extended row will have large, specifically size $\Theta(\ell)$, intersection with both $S(L)$ and $T(L)$. Roughly speaking, this is what enables to relate the distance of $f|_{S(L)}$ from degree ℓ to the distance of $f|_{T(L)}$ from degree ℓ. Ultimately, the crux of the PCPP proof is to relate the following five measures

[6] In the actual construction it is important to differentiate between the degree of Q in X and Y. We avoid this here, for simplicity of presentation.

[7] In the actual definition (Definition 4.1), $\Omega(L)$ actually needs to be defined as a similar but larger set, as the analysis sometimes requires that the restriction of $\Omega(L)$ to any row be an $\mathbb{F}_2$-subspace.

1. $\delta_{\mathrm{rect}}(f)$ - the distance of $f|_{S(L)}$ from bivariate polynomials of degree ℓ.
2. $\delta_c(f)$ - the average over $\alpha \in A$ of the distance of a "column" $f|_{\alpha \times q(L)}$ from univariate polynomials of degree ℓ.
3. $\delta_r(f)$ -the average over $\beta \in L_1$ of the distance of a "row" $f|_{\mathrm{extrow}_\beta \cap S(L)}$ from univariate polynomials of degree ℓ.
4. $\delta_r^{\mathsf{ext}}(f)$ -the average over $\beta \in L_1$ of the distance of an "extended row" $f|_{\mathrm{extrow}_\beta}$ from univariate polynomials of degree ℓ.
5. $\delta_{\mathrm{uni}}(f)$ - the distance of $f|_{T(L)}$ from bivariate polynomials of degree ℓ. This turns out to be the same as the distance of g from univariate polynomials of degree d.

What is implicit in the proofs of [BS08] and [BCGT13] is a relation of the form

$$c_1 \cdot \delta_c(f) + c_2 \cdot \delta_r(f) + c_3 \cdot \delta_r^{\mathsf{ext}}(f) \geq c \cdot \delta_{\mathrm{uni}}(f)$$

for non-negative c_1, c_2, c_3 with $c_1 + c_2 + c_3 = 1$ and $c > 0$. This relation suggests a natural recursive test: Think of c_1, c_2, c_3 as probabilities according to which either a random column, row, or extended row is chosen; Now check if f has the required degree on this restricted domain. The equation implies that if we started with a function g that is δ-far from degree d, we recurse, on average, on a function on a much smaller domain that is still $c \cdot \delta$-far from the degree we expect it to have. Intuitively, the larger c we can get, the larger bound we can get on V rejecting a function that is far from low degree; and the larger the bound, the less repetitions, and therefore verifier queries, we need to reject with probability $1/2$.

Making this relation more explicit is the starting point for several improvements over the analysis of [BCGT13]. For example

- The [BS08], [BCGT13] verifier recurses only on columns and *extended* rows. They analyze this by using the bound $\delta_r^{\mathsf{ext}}(f) \geq \delta_r(f)/2$ to move to a relation with $c_2 = 0$. This decreases the obtained c.
- The [BS08],[BCGT13] verifier simply chooses a column or extended row each with probability $1/2$; so they are implicitly using a relation with $c_1 = 1/2, c_2 = 0, c_3 = 1/2$. This also decreases the obtained c.
- [BCGT13] using the analysis of Polishchuk-Spielman [PS94], in fact first implicitly obtain a bound the form $\min\{c \cdot \delta_{\mathrm{uni}}(f), \gamma\}$ for some $0 < \gamma < 1/100$ on the right-hand side of the relation. Since they need to work with expectations over such expressions in their proof they move to the more convenient and much smaller quantity $\gamma \cdot c \cdot \delta_{\mathrm{uni}}(f)$. We show that, at least when V uses only two recursion levels, there is no need to "move from the minimum to the product" and lose this large factor. See Lemma 5.1.

Finally, we observe that for only one recursion level, a simple direct analysis is possible that avoids heavy factors coming from [PS94]. (See Claim 4.14 and Lemma 4.15).

3.2 An overview of second contribution — Security analysis of PCPP systems

Suppose now that P has in his possession a function $g : L \to \mathbb{F}$ that is far from any element of $\mathsf{RS}_L[\eta]$. He wishes to devise a pseudo-proof that will cause V to accept with high probability, i.e., V will conclude wrongly that $g \in \mathsf{RS}_L[\eta]$. Using the notation of Section 3.1, such a proof is a bivariate function f on the domain $\Omega(L)$, such that $f|_{T(L)} = g$, i.e., g is embedded in $\Omega(L)$ on the curve $T(L)$. f is supposed to be an evaluation of a polynomial $Q \in \mathbb{F}[X, Y]$ of (individual) degree $\ell = O(\sqrt{d})$. To check this, V chooses either a random "column" $\alpha \times q(L)$ or "extended row" extrow_β, and checks if the restriction of f to this column or extended row is a univariate function of degree at most ℓ. (Here we look at a simpler verifier used in [BS08, BCGT13] that does not use the non-extended rows, as opposed to what is described in Section 3.1.) A natural way to try to cheat is to choose an f with $f|_{T(L)} = g$ such that all extended rows have degree at most ℓ. This is what we call the "row-compliant attack". Similarly, one can choose f with $f|_{T(L)} = g$ such that all columns and at least a $2^{-\eta}$-fraction of the extended rows have degree at most ℓ. We call this a "column-compliant attack". In Appendix 6, we show that for most choices of g, V rejects f with much higher probability than what can be shown for a general pseudo-proof. Roughly speaking, we show that "while making all rows low degree, you will make many columns far from low degree", and vice-versa.

Summary The study of security of concrete feasible soundness of PCP systems is inevitable, if we wish to use such systems for concrete cryptographic applications with reasonable soundness.

We hope that these initial results will motivate others to study in more detail the feasible security of various PCP systems.

4 Improved analysis of the concrete efficiency of the BSS PCPP

We proceed with the formal definitions and analysis outlined in the previous section. We mention again that one reason we improve over [BCGT13] is by performing a tailored analysis of the first and second recursion level of (a variant of) the [BS08] PCPP, in which many factors can be saved, as compared to a generic analysis for any recursion depth.

Notation For $a, b \in \mathbb{R}$, it will be convenient to use the notation $a \vee b \triangleq \min\{a, b\}$.

Fix a positive integer k. We make the convention that an $\mathbb{F}_2$-subspace $L \subset \mathbb{F}$ of dimension k is always associated with some default basis $\{b_1, \ldots, b_k\}$ of L. Using this convention, for $c \in [k]$, we define $L_{\leq c} \subseteq L$ to be the subspace spanned by the first c vectors of this default basis. That is, $L_{\leq c} \triangleq \operatorname{span}\{b_1, \ldots, b_c\}$. We define the *midpoint of* k to be $\left\lfloor \frac{k-1}{2} \right\rfloor$.

The Ben-Sasson-Sudan PCPP [BS08] is based on a special, somewhat complex, subset of $\mathbb{F}^2$ that we describe next. We recommend looking at Appendix D for a helpful visualization of this set.

Definition 4.1 (BSS sets)**.** *Let $L \subseteq \mathbb{F}$ be an $\mathbb{F}_2$-subspace of dimension k. The* Ben-Sasson-Sudan (BSS) Set of L *,$\Omega(L)$, is defined as follows. First denote*

- $L_0 \triangleq L_{\leq \left\lfloor \frac{k-1}{2} \right\rfloor}$.
- $L_0' \triangleq L_{\leq \left\lfloor \frac{k-1}{2} \right\rfloor + 1}$.
- $L_1 \triangleq \operatorname{span}\left\{ b_{\left\lfloor \frac{k-1}{2} \right\rfloor + 1}, \ldots, b_k \right\}$, *and* $L_1' \triangleq q_{L_0}(L_1)$. *From the properties of subspace polynomials discussed in Section 2 we have* $|L_1'| = |L_1|$ *and* $L_1' = q_{L_0}(L)$.
- *For each* $\beta \in L_1$, *let* $L_\beta \triangleq \operatorname{span}\{L_0', \beta\}$ *if* $\beta \notin L_0'$ *and* $L_\beta = \operatorname{span}\left\{ L_0', b_{\left\lfloor \frac{k-1}{2} \right\rfloor + 2} \right\}$ *otherwise. (For simplicity, always think of* L_β *as "*L_0' *with* β *added" , i.e.,* $\operatorname{span}\{L_0', \beta\}$*).*
- *For each* $\beta \in L_1$, *now define the '*β*'th extended row' as* $\operatorname{extrow}_\beta \triangleq (L_\beta, q_{L_0}(\beta)) \subset \mathbb{F}^2$.

Finally, we define

$$\Omega(L) \triangleq \bigcup_{\beta \in L_1} \operatorname{extrow}_\beta.$$

Note that $|\Omega(L)| = 4 \cdot |L| = 2^{k+2}$.

For $\alpha \in L_0'$, *define the '*α*'th column' as* $\operatorname{col}_\alpha \triangleq (\alpha, L_1') \subset \Omega(L)$. *A useful property of* $\Omega(L)$, *that can be easily verified, is that the following two sets are contained in it:*

- *The 'product set of* $\Omega(L)$*':* $S(L) \triangleq L_0' \times L_1' \subset \Omega(L)$. [8]
- *The 'curve of* $\Omega(L)$*':* $T(L) \triangleq \{(v, q_{L_0}(v)) | v \in L\} \subset \Omega(L)$.

(We think of $T(L)$ *as an embedding of* L *into* $\Omega(L)$, *and for this reason think of* $\Omega(L)$ *as an extension of* L. *We note again that a helpful visualization of the above sets is found in appendix D.)*

Now fix a function $f : \Omega(L) \to \mathbb{F}$. *Define the* univariate *function* $P[f] : L \to \mathbb{F}$ *by* $P[f](Z) \triangleq f(Z, q_{L_0}(Z))$. *We assume a function* f *on a BSS-Set* $\Omega(L)$ *will always be associated with an integer parameter* $\eta = \eta(f)$ *- which informally 'represents the degree the prover is claiming* $P[f]$ *has'. Thus, we can use the parameter* η *in definitions relating to* f.

We define various measures describing distances of restrictions of f *to low degree polynomials. We will be most interested in the distance of* $P[f]$ *from* $\mathsf{RS}_L[\eta]$.

- *Define* $\delta_{\mathrm{uni}}(f) \triangleq \Delta(P[f], \mathsf{RS}_L[\eta])$.

[8] In [BS08, BCGT13] the set S was defined differently as a larger set.

- *For $\alpha \in L_0'$, define $f_\alpha^{\mathrm{col}} : L_1' \to \mathbb{F}$ by $f_\alpha^{\mathrm{col}}(Z) \triangleq f(\alpha, Z)$. Define $\delta_{c,\alpha}(f)$ to be the distance of f_α^{col} from polynomials of degree $2^{-\eta} \cdot |L_1| - 1$. That is,*

$$\delta_{c,\alpha}(f) \triangleq \Delta(f_\alpha^{\mathrm{col}}, \mathsf{RS}_{L_1'}[\eta]).$$

Finally, define $\delta_c(f) \triangleq \underset{\alpha \in L_0'}{\mathrm{E}} [\delta_{c,\alpha}(f)]$.

- *For $\beta \in L_1$, define $f_\beta^{\mathrm{row}} : L_0' \to \mathbb{F}$ by $f_\beta^{\mathrm{row}}(Z) \triangleq f(Z, q_{L_0}(\beta))$. Let $\delta_r(f) \triangleq \underset{\beta \in L_1}{\mathrm{E}} [\delta_{r,\beta}(f)]$ where*

$$\delta_{r,\beta}(f) \triangleq \Delta(f_\beta^{\mathrm{row}}, \mathsf{RS}_{L_0'}[1]).$$

- *For $\beta \in L_1$ define $f_\beta^{\mathrm{ext}} : L_\beta \to \mathbb{F}$ by $f_\beta^{\mathrm{ext}}(Z) \triangleq f(Z, q_L(\beta))$. Let $\delta_r^{\mathsf{ext}}(f) \triangleq \underset{\beta \in L_1}{\mathrm{E}} \left[\delta_{r,\beta}^{\mathrm{ext}}(f)\right]$ where*

$$\delta_{r,\beta}^{\mathrm{ext}}(f) \triangleq \Delta(f_\beta^{\mathrm{ext}}, \mathsf{RS}_{L_\beta}[2]).$$

Finally, define $\delta_{\mathrm{rect}}(f)$ to be the distance of $f|_S$ from the set of bi-variate polynomials of degree $(|L_0|-1, 2^{-\eta} \cdot |L_1|-1)$.

Similarly to extending a subspace L to $\Omega(L)$ we want to have a canonical way of extending a function $g \in \mathsf{RS}_L[\eta]$ to a function on $\Omega(L)$. For this purpose the following claim from [BS08] will be useful.

Claim 4.2. *Let $L \subseteq \mathbb{F}$ be a subspace, and fix $g \in \mathsf{RS}_L[\eta]$. There exists a bivariate polynomial $Q_g \in \mathbb{F}[X, Y]$ of degree $(|L_0| - 1, 2^{-\eta} \cdot |L_1| - 1)$ such that $Q_g(X, Y) \equiv g(X) \mod (Y - q_{L_0}(X))$. In particular, for any $z \in L$*

$$Q_g(z, q_{L_0}(z)) = g(z).$$

Definition 4.3 (BSS Extension)**.** *Fix a positive integer η. Let $L \subseteq \mathbb{F}$ be a subspace, and fix $g \in \mathsf{RS}_L[\eta]$. We define the* Ben-Sasson-Sudan extension of g, *$\Omega(g) : \Omega(L) \to \mathbb{F}$, to be the evaluation of the polynomial Q_g from Claim 4.2 on $\Omega(L)$.*

The following bound on the sizes of the subspaces arising in Definition 4.1 will be useful for analyzing the efficiency of our tests.

Claim 4.4. *Fix a subspace $L \subseteq \mathbb{F}$ of dimension k. Let L_1 and L_β be as in Definition 4.1. Then*

- $|L_1|, |L_\beta| \leq 2^{k/2+1.5}$
- $|L_1| + |L_\beta| \leq 2^{k/2+2.1}$

Proof. The first item is immediate from the definition. When k is even we have

$$|L_1| + |L_\beta| = 2^{k/2+1} + 2^{k/2+1} = 2^{k/2+2}.$$

When k is odd we have

$$|L_1| + |L_\beta| = 2^{k/2+1.5} + 2^{k/2+0.5} = (2^{1.5} + 2^{0.5}) \cdot 2^{k/2} \leq 2^{k/2+2.1}.$$

□

We proceed to show relations between the different measures defined. Intuitively we reduce the problem of verifying a function $g : L \to \mathbb{F}$ is close to some polynomial of low degree, to the problem of verifying $f = \Omega(g) : \Omega(L) \to \mathbb{F}$ is close to some bivariate polynomial of low degree. For this purpose we need to bound the distance of g from $\mathsf{RS}_L[\eta]$ by some attributes of f that are easier for the verifier to approximate. Lemma 4.5 shows the distance of g from $\mathsf{RS}_L[\eta]$ can not be much larger than $\max\{\delta_{\mathrm{rect}}(f), \delta_r^{\mathsf{ext}}(f)\}$.

Lemma 4.5. *Fix a subspace $L \subset \mathbb{F}$, and function $f : \Omega(L) \to \mathbb{F}$. We have*

$$\delta_{\mathrm{uni}}(f) \leq 2 \cdot \delta_{\mathrm{rect}}(f) + 4 \cdot \delta_r^{\mathsf{ext}}(f).$$

This improves the bound $\delta_{\mathrm{uni}}(f) \le 2 \cdot \delta_{\mathrm{rect}}(f) + 8 \cdot \delta_r^{\mathsf{ext}}(f)$ implicit in Section 11 of [BCGT13] based on [BS08].

Proof. Fix $Q \in \mathbb{F}[X,Y]$ of degree $(|L_0|-1, 2^{-\eta}\cdot|L_1|-1)$ with $\Delta_{S(L)}(Q,f) = \delta_{\mathrm{rect}}(f)$. For $\beta \in L_1$, denote by

- P_β an element of $\mathsf{RS}_{L_\beta}[2]$ closest to f_β^{ext}; formally, P_β satisfies $\Delta_{L_\beta}(P_\beta, f_\beta^{\mathrm{ext}}) = \delta_{r,\beta}^{\mathrm{ext}}(f)$.
- Q_β the univariate polynomial $Q_\beta(X) \triangleq Q(X, q_L(\beta))$.

Denote

$$\gamma \triangleq \Pr_{\beta \in L_1}(Q_\beta \neq P_\beta).$$

For $\beta \in L_1$ such that $Q_\beta \neq P_\beta$, we have $\Delta_{L_0'}(Q_\beta, P_\beta) \ge 1/2$.

Thus

$$\frac{1}{2}\cdot\gamma \le \mathop{\mathrm{E}}_{\beta\in L_1}\left[\Delta_{L_0'}(Q_\beta, P_\beta)\right]$$

Using the triangle inequality

$$\le \mathop{\mathrm{E}}_{\beta\in L_1}\left[\Delta_{L_0'}(Q_\beta, f_\beta^{\mathrm{row}})\right] + \mathop{\mathrm{E}}_{\beta\in L_1}\left[\Delta_{L_0'}(f_\beta^{\mathrm{row}}, P_\beta)\right].$$

Recall that $\delta_{\mathrm{rect}}(f)$ is the fractional distance between Q and f on $S(L)$, and note that $S(L) = L_0' \times L_1'$ is the union over $\beta \in L_1$ of $\mathrm{row}_\beta \triangleq L_0' \times \{q_L(\beta)\}$. Thus, $\delta_{\mathrm{rect}}(f)$ is equal to the average over $\beta \in L_1$ of the fractional distance between Q and f on row_β. Hence, we can replace the first term and get

$$= \delta_{\mathrm{rect}}(f) + \mathop{\mathrm{E}}_{\beta\in L_1}\left[\Delta_{L_0'}(f_\beta^{\mathrm{row}}, P_\beta)\right].$$

Let us call $\beta \in L_1$ *good* if $Q_\beta = P_\beta$. Calculation shows that $\deg(Q(Z, q_L(Z))) \le 2^{k-\eta}-1$ (see proof of Claim 4.14); and so $P[Q] : L \to \mathbb{F}$ defined by $P[Q](z) \triangleq Q(Z, q_L(Z))$ is in $\mathsf{RS}_L[\eta]$. Also, $\Delta(P[Q], P[f]) = \Delta_{T(L)}(Q,f)$. Thus,

$$\begin{aligned}\delta_{\mathrm{uni}}(f) \le \Delta_{T(L)}(Q,f) = \mathop{\mathrm{E}}_{\beta\in L_1}\left[\Delta_{T_\beta(L)}(Q_\beta, f_\beta^{\mathrm{ext}})\right] &\le \gamma + (1-\gamma)\cdot \mathop{\mathrm{E}}_{\beta \ is\ good}\left[\Delta_{T_\beta(L)}(Q,f)\right] \\ &\le \gamma + \mathop{\mathrm{E}}_{\beta\in L_1}\left[\Delta_{T_\beta(L)}(P_\beta, f)\right]\end{aligned}$$

Using our bound on γ we get

$$\le 2\cdot\delta_{\mathrm{rect}}(f) + \mathop{\mathrm{E}}_{\beta\in L_1}\left[2\cdot\Delta_{L_0'}(f_\beta^{\mathrm{row}}, P_\beta) + \Delta_{T_\beta(L)}(P_\beta, f_\beta^{\mathrm{ext}})\right]$$

Below we explain that this is

$$\begin{aligned}&\le 2\cdot\delta_{\mathrm{rect}}(f) + 4\cdot\mathop{\mathrm{E}}_{\beta\in L_1}\left[\delta_{r,\beta}^{\mathrm{ext}}(f)\right] \\ &= 2\cdot\delta_{\mathrm{rect}}(f) + 4\cdot\delta_r^{\mathsf{ext}}(f).\end{aligned}$$

We now explain the last inequality above. For $\beta \in L_1$, $\delta_{r,\beta}^{\mathrm{ext}}(f)$ is equal to the fractional number of disagreements between P_β and f_β^{ext} on L_β. Denote this set of locations of disagreements by $D \subseteq L_\beta$. Thus, $\delta_{r,\beta}^{\mathrm{ext}}(f) = \frac{|D|}{|L_\beta|}$.

L_0' and $T_\beta(L)$ are disjoint subsets of L_β of density $1/2$ and $1/4$ respectively. Thus,

$$\Delta_{L_0'}(f_\beta^{\mathrm{row}}, P_\beta) = \frac{|D\cap L_0'|}{|L_0'|} = \frac{2\cdot|D\cap L_0'|}{|L_\beta|}$$

and

$$\Delta_{T_\beta(L)}(P_\beta, f_\beta^{\mathrm{ext}}) = \frac{|D\cap T_\beta(L)|}{|T_\beta(L)|} = \frac{4\cdot|D\cap T_\beta(L)|}{|L_\beta|}.$$

So

$$2\cdot\Delta_{L_0'}(f_\beta^{\mathrm{row}}, P_\beta) + \Delta_{T_\beta(L)}(P_\beta, f_\beta^{\mathrm{ext}}) \le \frac{4}{|L_\beta|}\cdot(|D\cap L_0'| + |D\cap T_\beta(L)|) \le 4\cdot\delta_{r,\beta}^{\mathrm{ext}}(f).$$

□

For integer $\eta > 1$ define the constant $\delta_\eta \triangleq (1/4 - 2^{-\eta-1})^2$. For example, $\delta_2 = 1/64$ and $\delta_3 = 9/256$. The following is a corollary of Lemma 5.1 below.

Lemma 4.6. *Fix a subspace $L \subset \mathbb{F}$, and function $f : \Omega(L) \to \mathbb{F}$ with $\eta(f) \geq \eta$ for integer $\eta > 1$. We have*

$$\delta_r(f) + \delta_c(f) \geq \delta_\eta \vee (2/3) \cdot \delta_{\mathrm{rect}}(f).$$

Corollary 4.7. *Fix a subspace $L \subset \mathbb{F}$, and function $f : \Omega(L) \to \mathbb{F}$ with $\eta(f) \geq \eta$ for integer $\eta > 1$. Then*

$$3 \cdot \delta_c(f) + 3 \cdot \delta_r(f) + 4 \cdot \delta_r^{\mathsf{ext}}(f) \geq \delta_{\mathrm{uni}}(f) \vee 3 \cdot \delta_\eta.$$

Proof. Using Lemma 4.6 we know that either

1. $\delta_r(f) + \delta_c(f) \geq \delta_\eta$ which implies $3 \cdot \delta_c(f) + 3 \cdot \delta_r(f) + 4 \cdot \delta_r^{\mathsf{ext}}(f) \geq 3 \cdot \delta_\eta$ or
2. $\delta_r(f) + \delta_c(f) \geq (2/3) \cdot \delta_{\mathrm{rect}}(f)$ which implies, using Lemma 4.5,

$$3 \cdot \delta_c(f) + 3 \cdot \delta_r(f) + 4 \cdot \delta_r^{\mathsf{ext}}(f)$$

$$\geq 2 \cdot \delta_{\mathrm{rect}}(f) + 4 \cdot \delta_r^{\mathsf{ext}}(f) \geq \delta_{\mathrm{uni}}(f).$$

□

For integer $\eta > 1$ let us define $c_\eta \triangleq \frac{3 \cdot \delta_\eta}{10}$. For example, $c_2 \geq \frac{1}{250}, c_3 \geq \frac{1}{100}$. Simple calculations now show

Corollary 4.8. *Fix a subspace $L \subset \mathbb{F}$, and function $f : \Omega(L) \to \mathbb{F}$ with $\eta(f) \geq \eta$ for integer $\eta > 1$. Then*

$$\frac{3}{10} \cdot \delta_c(f) + \frac{3}{10} \cdot \delta_r(f) + \frac{4}{10} \cdot \delta_r^{\mathsf{ext}}(f) \geq \frac{\delta_{\mathrm{uni}}(f)}{10} \vee c_\eta.$$

4.0.1 Tightness of bounds

Note that the inequality of Lemma 4.5 is obviously an equality if we take $f : \Omega(L) \to \mathbb{F}$ to be identically zero. One may wonder if for non-zero f the bound can be improved. That is, can we get a better lower bound on $\delta_{\mathrm{rect}}(f)$ and $\delta_r^{\mathsf{ext}}(f)$ in terms of $\delta_{\mathrm{uni}}(f)$? (The quality of such a lower bound directly relates to the soundness of our PCPPs). We show that the bound cannot be improved beyond a factor exponentially small in the dimension of L.

Claim 4.9 (Tightness of bound)**.** *Fix any integers $\eta > 1$ and $i > 2$, and let $L^i \subset \mathbb{F}$ be a vector space of dimension i over $\mathbb{F}_2$. There is a function $f_i : \Omega(L^i) \to \mathbb{F}$ with $\eta(f_i) = \eta$ such that:*

$$\delta_{\mathrm{uni}}(f_i) \geq 2 \cdot \delta_{\mathrm{rect}}(f_i) + 4 \cdot \delta_r^{\mathsf{ext}}(f_i) - 2^{-(i-1)} > 0.$$

Proof. Fix integer $i > 2$, and let $L = L^i$ be an i dimensional subspace. Choose distinct $\beta_0 \neq \beta_1 \in L_1 \setminus L'_0$.

Let $h : L_0 \to \mathbb{F}$ be a mapping that is identically 0, except on $x = 0$. More precisely,

$$h(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$$

We define $h_{\beta_0} : L_{\beta_0} \to \mathbb{F}$ to be the "low degree extension of h on L_{β_0}". That is, h_{β_0} is the unique polynomial of degree smaller than $|L_0|$, with $h_{\beta_0}|_{L_0} = h$. We define $f = f_i : \Omega(L) \to \mathbb{F}$ in the following way:

$$f(x,y) = \begin{cases} h_{\beta_0}(x) & \text{if } y = \beta_0 \\ 1 & \text{if } (x,y) \in T_{\beta_1}(L) \\ 0 & \text{otherwise} \end{cases}$$

Note that $P[f]$ differs from the zero polynomial Z, only on $T_{\beta_0}(L) \cup T_{\beta_1}(L)$, and thus

$$\delta_{\mathrm{uni}}(f) \leq \Delta_L(Z, P[f]) \leq \frac{2}{|L_1|},$$

and therefore, as $\frac{2}{|L_1|} \leq 2^{-(i-1)} < (1-2^{-\eta})/2$, which is less than half the unique distance of $\mathsf{RS}_L[\eta]$, $\delta_{\mathrm{uni}}(f) = \frac{2}{|L_1|}$.

The only non zero entries of f in S are $(0 \cup (L'_0 \setminus L_0), q_{L_0}(\beta_0))$, so the closest polynomial of degree $(|L_0| - 1, 2^{-\eta} \cdot |L_1| - 1)$ to f over S is the zero polynomial, and we get $\delta_{\mathrm{rect}}(f) = \frac{1}{2 \cdot |L_1|} + \frac{1}{|L|}$.

There is only one extended row which is not low degree, and it is $f^{\mathrm{ext}}_{\beta_0}$ which is not zero only on $T_{\beta_0}(L)$, so we get $\delta^{\mathsf{ext}}_r(f) = \frac{1}{4 \cdot |L_1|}$.

Putting everything together, we get the required result

$$\delta_{\mathrm{uni}}(f) = 2 \cdot \delta_{\mathrm{rect}}(f) + 4 \cdot \delta^{\mathsf{ext}}_r(f) - \frac{2}{|L|} = 2 \cdot \delta_{\mathrm{rect}}(f) + 4 \cdot \delta^{\mathsf{ext}}_r(f) - 2^{1-i}.$$

□

4.1 Constructing PCPPs for $\mathrm{RS_a}[\eta]$

We describe in this subsection a few PCPP systems for the codes $\mathsf{RS}_L[\eta]$. We will need to introduce some more definitions and conventions before describing the PCPPs.

4.1.1 BSS sets and extensions of depth greater than one

Informally speaking, we wish to design tests where a verifier, examining a function on a BSS-Set, can recursively focus on a column, row, or extended row, and ask for the BSS-extension of the function restricted to that part. To formalize this we need to define BSS sets and BSS extensions of depth greater than one. We mention that in this section we only use extensions of depth one or two. In fact, we recommend skipping the following definition on a first read, as in this section it is only used in Subsection 4.1.4, where its use will probably be intuitively clear.

Definition 4.10 (Ben-Sasson-Sudan sets and extensions of arbitrary depth)**.** *Fix a subspace $L \subseteq \mathbb{F}$ and integer $d > 0$. For each $\beta \in L_1$ we denote by β^{ext} a distinct symbol disjoint from $\mathbb{F}$.*

The depth d BSS-Set of L*, denoted $\Omega_d(L)$, is defined inductively as follows.*

- $\Omega_1(L) \triangleq \Omega(L)$. $\overline{\Omega}_1(L) \triangleq \Omega_1(L) \setminus T(L)$.
- $\Omega_d(L)$ *is defined as the disjoint union of*
  1. $\Omega(L)$.
  2. $\cup_{\alpha \in L'_0}(\alpha, \overline{\Omega}_{d-1}(L'_1))$.
  3. $\cup_{\beta \in L_1}(\beta, \overline{\Omega}_{d-1}(L'_0))$.
  4. $\cup_{\beta \in L_1}(\beta^{\mathrm{ext}}, \overline{\Omega}_{d-1}(L_\beta))$.
- $\overline{\Omega}_d(L) \triangleq \Omega_d(L) \setminus T(L)$.

We add that

1. *As $\Omega_d(L)$ contains the depth one extension Ω(L), we can define for a function $f : \Omega_d(L) \to \mathbb{F}$ all measures that were defined for a function $f' : \Omega(L) \to \mathbb{F}$ - $\delta_c(f')$,$\delta_r(f')$,etc as the corresponding value defined for the restriction $\Omega_d(f)|_{\Omega(L)}$.*
2. *For $\alpha \in L'_0$, we define the 'α'th copy of $\Omega_{d-1}(L'_1) \subset \Omega_d(L)$', denoted $\Omega^{\alpha}_{d-1}(L'_1)$, as the union of (α, L'_1) and $(\alpha, \overline{\Omega}_{d-1}(L'_1))$. We identify $\Omega^{\alpha}_{d-1}(L'_1)$ with $\Omega_{d-1}(L'_1)$ by mapping $(\alpha, z) \in (\alpha, \overline{\Omega}_{d-1}(L'_1))$ to $z \in \overline{\Omega}_{d-1}(L'_1)$ and mapping $(\alpha, z) \in (\alpha, L'_1)$ to $(z, q_{(L'_1)_0}(z))$. We define $\Omega^{\beta}_{d-1}(L'_0)$ and $\Omega^{\beta^{\mathrm{ext}}}_{d-1}(L_{d-1})$ similarly.*

Fix a function $g \in \mathsf{RS}_L[\eta]$. The depth d BSS-extension of g, *denoted $\Omega_d(g)$, is a function $\Omega_d(L) \to \mathbb{F}$ defined inductively as follows.*

- $\Omega_1(g) \triangleq \Omega(g)$. $\overline{\Omega}_1(g) \triangleq \Omega_1(g)|_{\overline{\Omega}_1(L)}$.
- *$\Omega_d(g)$ is defined as the union*[9] *of*
  1. *$f : \Omega(L) \to \mathbb{F}$ defined as the (depth one) BSS-extension of g.*
  2. *$\cup_{\alpha \in L'_0} \overline{\Omega}_{d-1}(f^{\mathrm{col}}_\alpha)$ viewed as a function $(\alpha, \overline{\Omega}_{d-1}(L'_1)) \to \mathbb{F}$.*
  3. *$\cup_{\beta \in L_1} \overline{\Omega}_{d-1}(f^{\mathrm{row}}_\beta)$ viewed as a function $(\beta, \overline{\Omega}_{d-1}(L'_0)) \to \mathbb{F}$.*
  4. *$\cup_{\beta \in L_1} \overline{\Omega}_{d-1}(f^{\mathrm{ext}}_\beta)$ viewed as a function $(\beta^{\mathrm{ext}}, \overline{\Omega}_{d-1}(L_\beta)) \to \mathbb{F}$.*
- $\overline{\Omega}_d(g) \triangleq \Omega_d(g)|_{\overline{\Omega}_d(L)}$.

1. *We mention that the depth d extension of g contains the depth $d-1$ extensions of the restrictions $\Omega(g)^{\mathrm{col}}_\alpha, \Omega(g)^{\mathrm{row}}_\beta, \Omega(g)^{\mathrm{ext}}_\beta$ of $\Omega(g)$. For example, identifying $\Omega^\alpha_{d-1}(L'_1)$ with $\Omega_{d-1}(L'_1)$ as described above, the restriction $\Omega_d(g)|_{\Omega^\alpha_{d-1}(L'_1)}$ is precisely $\Omega_{d-1}(g^{\mathrm{col}}_\alpha)$.*
2. *In similar spirit, for an arbitrary function $f : \Omega_d(L) \to \mathbb{F}$, and $\alpha \in L'_1$, we define $\Omega_{d-1}(f^{\mathrm{col}}_\alpha) \triangleq f|_{\Omega^\alpha_{d-1}(L'_1)}$. $\Omega_{d-1}(f^{\mathrm{row}}_\beta)$ and $\Omega_{d-1}(f^{\mathrm{ext}}_\beta)$ are defined similarly.*

Calculation shows that if L has dimension k, $|\Omega_d(L)| \leq 2^{k+2+3(d-1)}$ (See Claim C.1 in Appendix C).

Remark 4.11 (Intersection of recursive extended rows with the curve of L)**.** *In Definition 4.1 we made no assumptions regarding the default basis $\{b_1, \ldots, b_k\}$ of the subspace L. When working with BSS sets of depth greater than one, it is important that bases are chosen so that "all recursive extended rows have intersection with the original univariate polynomial". To formalize this let $\Omega_d(L)$ be a BSS-Set of depth d. For $\beta_1, \ldots, \beta_d$ define $L_{\beta_1 \ldots \beta_d}$ as $L_{\beta_1 \ldots \beta_d} \triangleq (((L_{\beta_1})_{\beta_2})..)_{\beta_d}$. Note that $L_{\beta_1 \ldots \beta_d}$ is defined only when $\beta_i \in (L_{\beta_1 \ldots \beta_{i-1}})_1$ for all $i \in [d]$. In such a case, let us call $\beta_1, \ldots, \beta_d$* compatible*. We would like to view the subspace $L_{\beta_1 \ldots \beta_d}$ as a subset of $\Omega_d(L)$. We define such an embedding using induction on d: For $d = 1$, we identify L_{β_1} with $(\beta_1, L_{\beta_1}) \subset \Omega(L)$ in the natural way. Assume we have now embedded $L_{\beta_2 \ldots \beta_d}$ as a subset $L_{\beta_2 \ldots \beta_d} \subset \Omega_{d-1}(L_{\beta_1})$. Now using the identification of $\Omega_{d-1}(L_{\beta_1})$ with $\Omega^{\beta_1^{\mathrm{ext}}}_{d-1}(L_{d-1}) \subset \Omega_d(L)$, we obtain the embedding $L_{\beta_1 \ldots \beta_d} \subset \Omega_d(L)$.*

Assume $k \geq 2^d \cdot 2d$. We claim that the bases of $L_{\beta_1 \ldots \beta_i}$ can be chosen such that, for every compatible $\beta_1, \ldots, \beta_d$ and every $i \in [d]$, $L_{\beta_1 \ldots \beta_i} \cap T(L)$ is an affine subspace of co-dimension $2i$ in $L_{\beta_1 \ldots \beta_i}$ (when $L_{\beta_1 \ldots \beta_i}$ is embedded into $\Omega_d(L)$ as described above). In particular, $|L_{\beta_1 \ldots \beta_d} \cap T(L)| \geq 4^{-d} \cdot |L_{\beta_1 \ldots \beta_d}|$. We show this by induction on i. For $i = 1$, we know that $L_{\beta_1} \cap T(L)$ is the subspace $L_0 + \beta_1$ and $\dim(L_{\beta_1}) = \dim(L_0) + 2$. Assume the claim for i. Let $t \triangleq \dim(L_{\beta_1 \ldots \beta_i})$. Thus, we have a basis $v_1, \ldots, v_t$ for $L_{\beta_1 \ldots \beta_i}$ such that

$$L_{\beta_1 \ldots \beta_i} \cap T(L) = \{x = (x_1, \ldots, x_t) \in L_{\beta_1 \ldots \beta_i} | x_1 = a_1, \ldots, x_{2i} = a_{2i}\},$$

for some $a_1, \ldots, a_{2i} \in \mathbb{F}_2$ when x is written in the basis $v_1, \ldots, v_t$. This is the basis of $L_{\beta_1 \ldots \beta_i}$ we use. Calculation shows $\dim(L_{\beta_1 \ldots \beta_{i+1}}) \geq k/2^d$, and therefore

$$\dim((L_{\beta_1 \ldots \beta_i})_0) \geq k/2^d - 2 \geq 2i,$$

where the second inequality follows from our assumption on k and $i < d$. Thus, when using $v_1, \ldots, v_t$ as a basis for $L_{\beta_1 \ldots \beta_i}$, $(L_{\beta_1 \ldots \beta_i})_0$ will have basis $\{v_1, \ldots, v_{t'}\}$ for $t' \geq 2i$; and so $(L_{\beta_1 \ldots \beta_i})_0 \cap T(L)$ will have co-dimension $2i$ in $L_{\beta_1 \ldots \beta_i}$. As $(L_{\beta_1 \ldots \beta_i})_0 \subset L_{\beta_1 \ldots \beta_{i+1}}$ and $\dim(L_{\beta_1 \ldots \beta_{i+1}}) = \dim((L_{\beta_1 \ldots \beta_i})_0) + 2$, $L_{\beta_1 \ldots \beta_{i+1}} \cap T(L)$ will have co-dimension $2(i+1)$ in $L_{\beta_1 \ldots \beta_{i+1}}$.

[9] We define a single function by a union of several functions on disjoint domains.

4.1.2 Viewing functions on BSS sets as PCPP proofs for $\mathsf{RS}_L[\eta]$:

When describing our PCPP systems, we explicitly describe only the verifiers, which we think of as tests that are given access to a function $f : \Omega_d(L) \to \mathbb{F}$, and try to determine whether $P[f] \in \mathsf{RS}_L[\eta]$. The PCPP system for $\mathsf{RS}_L[\eta]$ that is implicitly defined by the test works as follows. The prover P starts with a function $g : L \to \mathbb{F}$ that he wishes to convince the verifier V belongs to $\mathsf{RS}_L[\eta]$. He constructs a function $y : \overline{\Omega}_d(L) \to \mathbb{F}$ and sends (g, y) to V. V thinks of (g, y) as a single function $f : \Omega_d(L) \to \mathbb{F}$ by identifying L with $T(L) = \Omega_d(L) \setminus \overline{\Omega}_d(L)$ as described before, i.e., mapping z to $(z, q_{L_0}(z))$; now V applies the test on f. The honest prover, given $g \in \mathsf{RS}_L[\eta]$, will take y to be the BSS-extension of g restricted to $\overline{\Omega}_d(L)$, i.e., $y = \overline{\Omega}_d(g)$. In other words, the honest prover simply sends $\Omega_d(g)$ to V.

Before presenting our first verifier, we define the *soundness function* of a PCPP verifier V. Loosely speaking, on input δ the function equals the minimal probability that V rejects an input x that is δ-far from the code.

Definition 4.12 (Soundness of test)**.** *Fix a randomized function $V : \mathbb{F}^* \to \{\mathsf{accept}, \mathsf{reject}\}$, and a code ensemble $\mathcal{C}$ over $\mathbb{F}$. Fix $0 < \delta < 1$. We define $S_{\mathcal{C}}[V](\delta)$ as the infimum over all $C \in \mathcal{C}$, all $x \in \mathbb{F}^{n(C)}$ that are δ-far from C, and all $y \in \mathbb{F}^*$ of*

$$\Pr(V(x, y) = \mathsf{reject}).$$

For integer $\eta > 0$, we use the shortened notation $S_\eta[V](\delta) \triangleq S_{\mathrm{RS}_a[\eta]}[V](\delta)$.

4.1.3 A test of depth one

We define the test $T_{1,\eta}$.

$\underline{T_{1,\eta}}$**(f)** **input:** $f : \Omega(L) \to \mathbb{F}$.

1. Choose random $\alpha \in L'_0$. Check if f^{col}_α has degree at most $2^{-\eta} \cdot |L_1| - 1$.
2. Choose random $\beta \in L_1$. Check if f^{ext}_β has degree at most $|L_0| - 1$.
3. Return reject if one of the above checks failed. Return accept otherwise.

The following claim will be used to analyze the test.

Claim 4.13. *Fix subsets $A = \{\alpha_1, \ldots, \alpha_n\}, B = \{\beta_1, \ldots, \beta_m\}$ of $\mathbb{F}$. Fix positive integers $d < n$ and $m' \leq m$. Suppose we have a function $f : A \times B \to \mathbb{F}$ such that*

- *for $1 \leq j \leq m'$ we have that $f(X, \beta_j)$ has degree at most d.*
- *For $1 \leq i \leq d + 1$ we have that $f(\alpha_i, Y)$ has degree at most e.*

Then there exists a bivariate polynomial $Q \in \mathbb{F}[X, Y]$ of degree (d, e) such that for every $1 \leq j \leq m'$ $f(X, \beta_j) = Q(X, \beta_j)$.

Proof. For $1 \leq i \leq d + 1$, define a polynomial $\delta_i(X)$ of degree at most d such that $\delta_i(\alpha_i) = 1$, and $\delta_i(\alpha_\ell) = 0$ for $1 \leq \ell \leq d + 1, \ell \neq i$. Let $g_i(Y)$ be the polynomial of degree at most e that identifies with $f(\alpha_i, Y)$ on B.

Now define

$$Q(X, Y) \triangleq \sum_{i=1}^{d+1} \delta_i(X) \cdot g_i(Y).$$

Clearly Q has degree (d, e). It is immediate from Q's formula that $Q(\alpha_i, Y) = g_i(Y)$ for $1 \leq i \leq d + 1$. Thus, for any $1 \leq i \leq d + 1$ and $1 \leq j \leq m'$, $Q(\alpha_i, \beta_j) = g_i(\beta_j) = f(\alpha_i, \beta_j)$. In other words, $Q(X, \beta_j)$ and $f(X, \beta_j)$ agree on $d + 1$ points and therefore agree on all of A.

□

Lemma 4.14 (Soundness of depth one test)**.** *For any $0 < \delta < 1$*

$$S_\eta[T_{1,\eta}](\delta) \geq \delta \vee 1/2 \geq \delta/2.$$

Proof. Fix $f : \Omega(L) \to \mathbb{F}$. Denote $d = |L_0| - 1$ and $e = 2^{-\eta} \cdot |L_1| - 1$. Suppose that $\Pr(T_{1,\eta}(f) = \mathsf{accept}) \geq \max\{1/2, 1 - \delta\}$. In particular there is a set $C \subset L'_0$ with $|C| \geq |L'_0|/2$ such that $\deg(f^{\mathrm{col}}_\alpha) \leq e$ for all $\alpha \in C$. It follows from Claim 4.13 that there exists a bivariate polynomial $Q(X, Y)$ of degree (d, e) such that

$$Q(X, q_{L_0}(\beta)) = f(X, q_{L_0}(\beta))$$

for any $\beta \in R \subset L_1$ where R is a set of size at least $(1 - \delta) \cdot |L_1|$. Now defining

$$P(Z) \triangleq Q(Z, q_{L_0}(Z))$$

We see that

1. P agrees with $P[f]$ on L with probability at least $1 - \delta$: As

$$\Pr_{z \leftarrow L}(P(z) = P[f](z)) \geq \Pr_{\beta \leftarrow L_1}(Q(X, \beta) = f^{\mathrm{ext}}_\beta) \geq 1 - \delta.$$

2.

$$\begin{aligned} \deg P &\leq |L_0| - 1 + |L_0| \cdot (2^{-\eta} \cdot |L_1| - 1) \\ &= 2^{k-\eta} - 1 \end{aligned}$$

Thus, as P is δ-close to $P[f]$ and $\deg(P) \leq 2^{k-\eta} - 1$, $\delta_{\mathrm{uni}}(f) \leq \delta$. □

Amplifying $T_{1,\eta}$ with repetitions we get

Lemma 4.15. *There is a $(2^{\ell+5}, 2^{\ell/2+5.5})$-PCPP system for $\mathrm{RS_a}[3]$, where 2^ℓ denotes the message length.*

> **Reminder**
>
> *A $(2^{\ell+5}, 2^{\ell/2+5.5})$-PCPP system for some code C will generate a proof of length $A = 2^{\ell+5}$ field elements, of which the verifier V reads only $Q = 2^{\ell/2+5.5}$ field elements, and is guaranteed to reject with probability at least $\frac{1}{2}$ any x that is $d(C)/3$-far from C. (Definition 1.1) In case $C \in \mathrm{RS_a}[3]$ we have $d(C) = 7/8$.*

Proof. Fix a subspace $L \subseteq \mathbb{F}$ of dimension $k = \ell + 3$. Fix $f : \Omega(L) \to \mathbb{F}$ with $\eta(f) = 3$ and $\delta_{\mathrm{uni}}(f) \geq \delta \triangleq \frac{7}{8} \cdot \frac{1}{3} \geq \frac{1}{4}$. Consider the verifier V that given such f runs $T_{1,3}(f)$ 3 times and accepts if and only if $T_{1,3}$ accepted every time. We have

$$\Pr(V(f) = \mathsf{accept}) \leq (1 - 1/4)^3 \leq 1/2.$$

At each repetition V queries $|L_\beta| + |L'_1|$ field elements (for some $\beta \in L'_0$) which is at most $2^{k/2+2.1}$ by Claim 4.4. So $Q \leq 3 \cdot 2^{k/2+2.1} \leq 2^{k/2+4} = 2^{\ell/2+5.5}$. P needs to write down the table of $f : \Omega(L) \to \mathbb{F}$ which is of length $|\Omega(L)| = 4 \cdot |L| = 2^{k+2} = 2^{\ell+5}$.

□

Corollary 4.16. *Fix any field $\mathbb{F}$ of characteristic two. There is a PCPP system for $\mathrm{RS_a}[3]$ with concrete efficiency threshold at most 2^{23}.*

Proof. Using the system from Lemma 4.15 we need to see for what ℓ we have

$$2^{\ell+5} \cdot 2^{\ell/2+5.5} \leq 2^{2\ell-1}.$$

Equivalently

$$\ell/2 \geq 11.5,$$

which is satisfied when $\ell \geq 2^{23}$. □

4.1.4 A test of depth two

We define the test $T_{2,\eta}$.

$\underline{T_{2,\eta}\textbf{(f)}}$ **input:** $f : \Omega_2(L) \to \mathbb{F}$.

1. With probability $\frac{3}{10}$, choose random $\alpha \in L'_0$; and return $T_{1,\eta}(\Omega(f^{\mathrm{col}}_\alpha))$.
2. With probability $\frac{3}{10}$, choose random $\beta \in L_1$; and return $T_{1,1}(\Omega(f^{\mathrm{row}}_\beta))$.
3. Otherwise, i.e. with probability $\frac{4}{10}$, choose random $\beta \in L_1$; and return $T_{1,2}(\Omega(f^{\mathrm{ext}}_\beta))$.

Lemma 4.17. *[Soundness of depth 2 test] Fix any integer $\eta > 1$. For any $0 < \delta < 1$*

$$S_\eta[T_{2,\eta}](\delta) \geq \frac{\delta}{20} \vee \frac{c_\eta}{2}.$$

For instance,

$$S_3[T_{2,3}](\delta) \geq \frac{\delta}{20} \vee \frac{1}{200}$$

and

$$S_2[T_{2,2}](\delta) \geq \frac{\delta}{20} \vee \frac{1}{500}.$$

Proof. Fix any $f : \Omega(L) \to \mathbb{F}$ with $\delta_{\mathrm{uni}}(f) = \delta$.

$$\Pr[T_{2,\eta}(f) = \mathsf{reject}] =$$

$$\frac{3}{10} \cdot \mathop{\mathrm{E}}_{\alpha \in L'_0} \left[\Pr(T_{1,\eta}(\Omega(f^{\mathrm{col}}_\alpha)) = \mathsf{reject})\right] + \frac{3}{10} \cdot \mathop{\mathrm{E}}_{\beta \in L_1} \left[\Pr(T_{1,1}(\Omega(f^{\mathrm{row}}_\beta) = \mathsf{reject})\right] + \frac{4}{10} \cdot \mathop{\mathrm{E}}_{\beta \in L_1} \left[\Pr(T_{1,2}(\Omega(f^{\mathrm{ext}}_\beta) = \mathsf{reject})\right]$$

Using Claim 4.14 this is

$$\geq \frac{3}{10} \cdot \mathop{\mathrm{E}}_{\alpha \in L'_0} [\delta_{c,\alpha}(f)/2] + \frac{3}{10} \cdot \mathop{\mathrm{E}}_{\beta \in L_1} [\delta_{r,\beta}(f)/2] + \frac{4}{10} \cdot \mathop{\mathrm{E}}_{\beta \in L_1} \left[\delta^{\mathsf{ext}}_{r,\beta}(f)/2\right]$$

$$\geq 1/2 \cdot \left(\frac{3}{10} \cdot \delta_c(f) + \frac{3}{10} \cdot \delta_r(f) + \frac{4}{10} \cdot \delta^{\mathsf{ext}}_r(f)\right)$$

Using Corollary 4.8, this is

$$\geq 1/2 \cdot \left(\frac{\delta_{\mathrm{uni}}(f)}{10} \vee c_\eta\right) = \frac{\delta}{20} \vee \frac{c_\eta}{2}.$$

□

Amplifying the test by repetition to soundness half we can get

Lemma 4.18. *There is a $(2^{\ell+8}, 2^{\ell/4+10.75})$-PCPP system for $\mathrm{RS}_{\mathrm{a}}[3]$.*

Proof. Fix a subspace $L \subseteq \mathbb{F}$ of dimension $k = \ell + 3$. Fix $f : \Omega(L) \to \mathbb{F}$ with $\eta(f) = 3$ and $\delta_{\mathrm{uni}}(f) \geq \delta \triangleq \frac{7}{8} \cdot \frac{1}{3} \geq \frac{1}{4}$. Consider the verifier V that given such f runs $T_{2,3}(f)$ 140 times and accepts if and only if $T_{2,3}$ accepted every time. We have

$$\Pr(V(f) = \mathsf{accept}) \leq (1 - \frac{1}{200})^{140} \leq 1/2.$$

Using Claim 4.4, we can see that at each repetition V queries at most $2^{k/4+2.85}$ field elements. So $Q \leq 140 \cdot 2^{k/4+2.85} \leq 2^{k/4+10} = 2^{\ell/4+10.75}$. P needs to write down the table of $f : \Omega_2(L) \to \mathbb{F}$ of size $|\Omega_2(L)| \leq 2^{k+5} = 2^{\ell+8}$. □

5 Improving the bound obtained from Polishchuk-Spielman [PS94]

Fix a bivariate function $f : A \times B \to \mathbb{F}$. As in [BCGT13] we denote in this section by

- $\delta^{(d,*)}_{A\times B}(f)$ - the distance of f from polynomials of degree $(d, |B| - 1)$. This is the same as the average over $\gamma \in B$ of the distance between $f(X, \gamma)$ and polynomials of degree d.
- $\delta^{(*,e)}_{A\times B}(f)$ - the distance of f from polynomials of degree $(|A| - 1, e)$. This is the same as the average over $\alpha \in A$ of the distance between $f(\alpha, Y)$ and polynomials of degree e.
- $\delta^{(d,e)}_{A\times B}(f)$ - the distance of f from polynomials of degree (d, e).

Using a more careful pass of the proof of Lemma 10.6 in [BCGT13], we prove

Lemma 5.1. *Fix a field $\mathbb{F}$. Fix positive integers d, m, e, n such that $d/m + e/n < 1$. Fix $\delta > 0$ such that $\delta < 1/2(1 - d/m - e/n)$. Fix $A, B \subseteq \mathbb{F}$ with $|A| = m$ and $|B| = n$. Fix any function $f : A \times B \to \mathbb{F}$. Then*

$$\delta^{(d,*)}_{A\times B}(f) + \delta^{(*,e)}_{A\times B}(f) \geq \min\left\{\delta^2, \delta^{(d,e)}_{A\times B}(f)/1.5\right\}$$

We remark that the bound in Lemma 10.6 of [BCGT13] was $\delta^2 \cdot \delta^{(d,e)}_{A\times B}(f)$ which can be significantly smaller.

Proof. If $\delta^{(d,*)}_{A\times B}(f) + \delta^{(*,e)}_{A\times B}(f) \geq \delta^2$ we are done. Suppose from now on that $\delta^{(d,*)}_{A\times B}(f) + \delta^{(*,e)}_{A\times B}(f) < \delta^2$. Let R be the[10] polynomial of degree $(d, |B| - 1)$ closest to f. We have $\Delta_{A\times B}(f, R) = \delta^{(d,*)}_{A\times B}(f)$. From the triangle inequality

$$\delta^{(d,e)}_{A\times B}(f) \leq \Delta_{A\times B}(f, R) + \delta^{(d,e)}_{A\times B}(R) = \delta^{(d,*)}_{A\times B}(f) + \delta^{(d,e)}_{A\times B}(R)$$

Similarly, let C be the polynomial of degree $(|A| - 1, e)$ closet to f. We have $\Delta_{A\times B}(f, C) = \delta^{(*,e)}_{A\times B}(f)$. From the triangle inequality

$$\delta^{(d,e)}_{A\times B}(f) \leq \Delta_{A\times B}(f, C) + \delta^{(d,e)}_{A\times B}(C) = \delta^{(*,e)}_{A\times B}(f) + \delta^{(d,e)}_{A\times B}(C).$$

Now define $\delta' \triangleq \sqrt{\delta^{(d,*)}_{A\times B}(f) + \delta^{(*,e)}_{A\times B}(f)}$. We have

$$\Delta_{A\times B}(R, C) \leq \Delta_{A\times B}(R, f) + \Delta_{A\times B}(f, C) = \delta^{(d,*)}_{A\times B}(f) + \delta^{(*,e)}_{A\times B}(f) = \delta'^2.$$

It follows from the analysis of [BCGT13] of the Polishchuk-Spielman test (Theorem 10.5 of [BCGT13]) that there exists a polynomial Q of degree (d, e) that disagrees on a uniform point of $A \times B$ with either R or C with probability at most $2\delta'^2$. It follows that

$$\delta^{(d,e)}_{A\times B}(R) + \delta^{(d,e)}_{A\times B}(C) \leq 2\delta'^2 = 2 \cdot (\delta^{(d,*)}_{A\times B}(f) + \delta^{(*,e)}_{A\times B}(f))$$

and thus

$$\begin{aligned}\delta^{(d,e)}_{A\times B}(f) &= \frac{\delta^{(d,e)}_{A\times B}(f)}{2} + \frac{\delta^{(d,e)}_{A\times B}(f)}{2} \leq \frac{\delta^{(d,*)}_{A\times B}(f) + \delta^{(d,e)}_{A\times B}(R)}{2} + \frac{\delta^{(*,e)}_{A\times B}(f) + \delta^{(d,e)}_{A\times B}(C)}{2} \\ &< \frac{\delta^{(d,*)}_{A\times B}(f) + \delta^{(*,e)}_{A\times B}(f)}{2} + \delta'^2 = 1.5(\delta^{(d,*)}_{A\times B}(f) + \delta^{(*,e)}_{A\times B}(f)).\end{aligned}$$

□

[10] If there is not a unique polynomial closest to f, pick one arbitrarily.

6 Attacks and security of the BSS RS-PCPP system

In this section we analyze the security of the RS-PCPP system from Definition 4.10 against two natural and efficient attacks. We start by defining these attacks (Definition 6.3) and then study security against them on random functions in Section 6.2.

Remark 6.1 (Choice of verifier)**.** *In this section we analyze the "standard BSS verifier" as described in [BS08] (See definition 6.2). Using the terminology of Section 4 this verifier looks only at columns and extended rows. Hence, whenever we use the term row informally in this section, we mean an extended row in the terminology of Section 4. Similarly, when we refer to BSS-sets $\Omega_d(L)$, we think of of the containing the only the subproofs $\Omega_{d-1}(f_\alpha^{\mathrm{col}}), \Omega_{d-1}(f_\beta^{\mathrm{ext}})$ of the columns and extended rows, but not the subproofs $\Omega_{d-1}(f_\beta^{\mathrm{row}})$ of the non-extended rows (See Claim) .*

6.1 Two attacks

Our starting point is a function $f : L \to \mathbb{F}$ where L is a subspace and we assume f is δ-far from $\mathsf{RS}_L[\eta]$. Recalling the recursive construction of the BSS proof (Definition 4.10), and the fact that the tests applied by the standard BSS verifier V to (the first level of) the BSS proof are either "row"-tests or "column"-tests, two natural attacks arise. In the first attack the pseudo-prover constructs a pseudo-proof in which all rows are low-degree and in the second one all columns of the pseudo-proof are low-degree. Each pseudo-proof is a generalization of the BSS extension (Definition 4.3) to the case that f does not belong to $\mathsf{RS}_L[\eta]$, and coincides with the BSS extension when $f \in \mathsf{RS}_L[\eta]$. The row-compliant pseudo-proof is obtained by computing a BSS extension with $\eta' = 0$ and the column-compliant is constructed using the first $2^{-\eta}$ fraction of entries of f to define a new low-degree polynomial for which the BSS extension is derived.

We start by defining the "standard" BSS verifier, one that interacts with the BSS proof from Definition 4.10 and uses the notation given there.

Recall first, from Definition 4.1, that we sometimes implicitly identify a subspace L with the subset $T(L) \subset \Omega(L)$. As $\Omega(L) \subset \Omega_d(L)$(See Definition 4.10), we can similarly think of L as a subset of $\Omega_d(L)$. Recall further, from Definition 4.10, that we use the notation $\overline{\Omega}_d(L) = \Omega_d(L) \setminus T(L)$.

Definition 6.2 (Depth d RS-PCPP verifier)**.** *Fix integer η. Let $f : L \to \mathbb{F}$ and $\pi_d : \overline{\Omega}_d(L) \to \mathbb{F}$ be a purported PCPP for $f \in \mathsf{RS}_L[\eta]$ as in Definition 4.10. Let $g \triangleq (f \| \pi_d) : \Omega_d(L) \to \mathbb{F}$ denote the concatenation of f and π_d. The BSS-verifier V_{RS}^g operates as follows:*

- *If $d = 0$ (in which case $\overline{\Omega}_0(L) = \emptyset$) then read f entirely (on all of L) and accept iff $f \in \mathsf{RS}_L[\eta]$.*
- *For $d > 0$,*
  - *with probability $1/2$ pick a uniformly random row $\beta \in L_1$ and return $V_{\mathsf{RS}}^{\Omega_{d-1}(g_\beta^{\mathrm{ext}})}$,*
  - *with probability $1/2$ pick a uniformly random column $\alpha \in L_0'$ and return $V_{\mathsf{RS}}^{\Omega_{d-1}(g_\alpha^{\mathrm{col}})}$.*

  (See Definition 4.10 for the definition of $\Omega_{d-1}(g_\beta^{\mathrm{ext}})$ and $\Omega_{d-1}(g_\alpha^{\mathrm{col}})$. In words, they are simply the parts of $\Omega_d(g)$ corresponding to the depth $d-1$ extensions of g_β^{ext} and g_α^{col}).

Calculation shows that given $g : \Omega_d(L) \to \mathbb{F}$, for L of dimension k, V_{RS}^g reads at most $q_0(k,d) \triangleq 2^{k/2^d + 3(1-1/2^d)}$ locations of g. Next we formally define two natural attacks on the BSS RS-PCPP verifier. Recall that the BSS-extension of a function $f \in \mathsf{RS}_L[\eta]$ was a function $g : \Omega_d(L) \to \mathbb{F}$. We define the attacks by giving two ways to extend an *arbitrary* function $f : L \to \mathbb{F}$, i.e., f that may not belong to $\mathsf{RS}_L[\eta]$, into a function $g : \Omega_d(L) \to \mathbb{F}$.

Definition 6.3 (Row- and Column-compliant BSS extension of arbitrary functions)**.** *Fix a positive integer η. Let $L \subseteq \mathbb{F}$ be a subspace, and fix $f : L \to \mathbb{F}$. We define two "BSS extensions" of f:*

- **Row-compliant:** *We define the* row-compliant Ben-Sasson-Sudan extension of f, *denoted $\Omega_-(f) : \Omega(L) \to \mathbb{F}$, as follows. First, using the identification of L and $T(L)$ define $\Omega_-(f)(z, q_{L_0}(z)) = f(z)$, for $z \in L$. Recall that $T_\beta(L) = T(L) \cap \mathrm{extrow}_\beta$ has size $|L_\beta|/4$. Thus, there is a unique $P \in \mathsf{RS}_{L_\beta}[2]$, i.e. a unique polynomial of degree*

at most $|L_\beta|/4 - 1$, such that $P(x) = \Omega_-(f)(x,\beta)$ for each $(x,\beta) \in T_\beta(L)$. We define $\Omega_-(f)^{\text{ext}}_\beta$, i.e., $\Omega_-(f)$ on the β'th row, according to P. That is, for each $(x,\beta) \in \text{extrow}_\beta$,

$$\Omega_-(f)^{\text{ext}}_\beta(x) = \Omega_-(f)(x,\beta) \triangleq P(x).$$

Note that $\Omega_-(f)^{\text{ext}}_\beta \in \mathsf{RS}_{L_\beta}[2]$ for each $\beta \in L_1$. Equivalently, $\Omega_-(f)$ can be defined as the evaluation of the polynomial Q_f from Claim 4.2 on $\Omega(L)$, where Q_f is computed there with respect to $f \in \mathsf{RS}_L[0]$.

- **Column-compliant:** *Let $\hat{L}$ be the subspace of L of co-dimension η spanned by the first $\dim(L) - \eta$ elements of the basis of L. Let $\hat{f}$ be the low-degree extension to L of $f|_{\hat{L}}$; that is, $\hat{f} : L \to \mathbb{F}$ is the unique element of $\mathsf{RS}_L[\eta]$ that identifies with f on $\hat{L}$. We define the* column compliant Ben-Sasson-Sudan extension of f, *denoted $\Omega_|(f) : \Omega(L) \to \mathbb{F}$, to be the BSS-extension $\Omega(\hat{f})$ of $\hat{f}$.*

  It can be seen that this is equivalent to the following. Denote by $R_{\hat{L}} \subset L_1$ the set of $2^{-\eta} \cdot |L_1|$ elements $\beta \in L_1$ such that $q_{L_0}(\beta) \in q_{L_0}(\hat{L})$. As in the row-compliant extension, define $\Omega_|(f)$ on $T(L)$ according to f. Now define $\Omega_|(f)$ on extended rows extrow_β for $\beta \in R_{\hat{L}}$ by a low degree extension of $\Omega_|(f)|_{T_\beta(L)}$. Now complete each column col_α by a low degree extension of the values $\{(\alpha,\beta)\}_{\beta \in R_{\hat{L}}}$. Now complete the values $\text{extrow}_\beta \setminus T_\beta(L)$ for $\beta \notin R_{\hat{L}}$ as a low degree extension of the values in row_β.

The depth d row-compliant BSS extension of f, denoted $\Omega_{d,-}(f)$ is obtained by using the row-compliant BSS extension recursively. Formally, this is obtained by setting $\Omega_1(f) = \Omega_-(f)$ in Definition 4.10 and then recursively (for recursion depth $d-1$) computing the row-compliant BSS extension of every row and column of $\Omega_-(f)$. The depth d column-compliant BSS extension of f is denoted $\Omega_{d,|}(f)$ and computed recursively analogously by computing the column-compliant BSS extension of f and repeating this process for every row and column of $\Omega_|(f)$.

Notice that when $f \in \mathsf{RS}_L[\eta]$ then by Claim 4.2 we have $\Omega_-(f) = \Omega(f)$ by construction. Furthermore, in this case $\hat{f} = f$ so we also have $\Omega_|(f) = \Omega(f)$. In other words, when $f \in \mathsf{RS}_L[\eta]$ then both row- and column-compliant BSS extensions are equal to the "standard" BSS extension from Definition 4.3. The following claim justifies the names of these pseudo-proofs.

Claim 6.4. *For any $f : L \to \mathbb{F}$ and integer $\eta < \dim(L)$, and for $\Omega_-(f), \Omega_|(f) : \Omega(L) \to \mathbb{F}$ we have*

- *For $\alpha \in L'_0$ we have $\Omega_|(f)^{\text{col}}_\alpha \in \mathsf{RS}_{L'_1}[\eta]$. Consequently, $\delta_c(\Omega_|(f)) = 0$*
- *For $\beta \in L_1$, we have $\Omega_-(f)^{\text{row}}_\beta \in \mathsf{RS}_{L'_0}[1]$. Consequently, $\delta_r(\Omega_-(f)) = 0$. Similarly, $\Omega_-(f)^{\text{ext}}_\beta \in \mathsf{RS}_{L_\beta}[2]$ and $\delta^{\mathsf{ext}}_r(\Omega_-(f)) = 0$.*

Proof. The first bullet means that all columns of $\Omega_|(f)$ belong to $\mathsf{RS}_{L'_1}[\eta]$. This holds because each such α-column is the evaluation of $Q_{\hat{f}}$ on $\alpha \times L'_1$ and $\deg_Y(Q_{\hat{f}}) \leq 2^{-\eta}|L'_1| - 1$ by Claim 4.2 and because by construction $\deg(\hat{f}) < 2^{-\eta}|L|$. Similarly, the second bullet holds because any $f : L \to \mathbb{F}$ belongs to $\mathsf{RS}_L[0]$, so Claim 4.2 implies that $\deg_X(Q_f) < |L_0|$, and by construction $|L_0|/|L'_0| = 1/2$ and $|L_0|/|L_\beta| = 1/4$. □

By construction of the BSS verifier, if it selects the α-column and $\Omega(f)^{\text{col}}_\alpha = \Omega_|(f)^{\text{col}}_\alpha$ then the test will pass with probability 1. Similarly, if $\Omega(f) = \Omega_-(f)$ then all row-tests pass with probability 1. The interesting question from the point of view of soundness is what happens, recursively, to the other kind of tests.

6.2 Security analysis on random functions

In the two lemmas below we analyze the security of the BSS PCPP on random functions. Random functions are a good starting point for analyzing security of the PCP system of [BS08] for two reasons:

- We are not aware of "natural" unsatisfiable instances for which a pseudo-proof leads to evaluations of functions that have lower (i.e., worse) soundness than what can be obtained for a uniformly random function.

- As far as we currently understand, random functions resemble rational functions and rational functions arise naturally in the only nontrivial attack on PCP systems that we are aware of (details on this attack will be given in subsequent work).

Lemma 6.5 (Row-compliant soundness on random functions)**.** *Assume* $|\mathbb{F}| \geq 512$*. With probability* 0.9 *over random* $f : L \to \mathbb{F}$

$$\Pr\left[V_{\mathsf{RS}}^{\Omega_-(f)} = \mathsf{reject}\right] \geq 2^{-d}$$

where the probability is over the randomness of V_{RS}*.*

Proof. For simplicity, we focus on the case $d = 1$. The general case is analogous. The proof relies on the following claim.

Claim 6.6. *For any fixed* $\alpha \in L_0'$ *we have that* $(\Omega_-(f))_\alpha^{\mathrm{col}}$ *is a uniformly random function.*

Note that a random function $g : L_1' \to \mathbb{F}$ is in $\mathsf{RS}_{L_1'}[\eta]$ with probability at most $1/|\mathbb{F}|$ by our assumption on k. Assuming the claim, we conclude by the Markov inequality that with probability 0.9 over the choice of $f : L \to \mathbb{F}$, $(\Omega_-(f))_\alpha^{\mathrm{col}}$ is in $\mathsf{RS}_{L_1'}[\eta]$ for at most a $10/|\mathbb{F}|$ fraction of $\alpha \in L_0'$. As $|L_0'| \leq 2 \cdot \sqrt{|\mathbb{F}|}$, we have $10/|\mathbb{F}| < 1/|L_0'|$. We conclude for such f that $(\Omega_-(f))_\alpha^{\mathrm{col}} \notin \mathsf{RS}_{L_1'}[\eta]$ for all $\alpha \in L_0'$. Thus, whenever V_{RS} chooses to query a column, he will reject. A similar argument shows that if V_{RS} is given a depth d row-compliant BSS extension of $f : L \to \mathbb{F}$, he will always reject on a 0.9 fraction of $f : L \to \mathbb{F}$, given that he recursively chose to query only columns, which happens with probability 2^{-d}. □

Proof of Claim 6.6. If f is uniformly random then for $\beta \in L_1$ we have that $\Omega_-(f)_\beta^{\mathrm{ext}} : L_\beta \to \mathbb{F}$ is a uniformly random element of $\mathsf{RS}_{L_\beta}[2]$. Furthermore, the rows of $\Omega_-(f)$ are independent. It follows that the α-column of $\Omega_-(f)$ is a uniformly random function, as claimed. □

We proceed to analyze the column-compliant attack.

Lemma 6.7 (Column-compliant soundness on random functions)**.** *Assume that* $k = \dim(L) \geq 2^d \cdot 2d$ *and* $|\mathbb{F}| \geq 512$*. Assume* $\eta \geq 2$*. Then, with probability at least* 0.9 *over uniform* $f : L \to \mathbb{F}$

$$\Pr\left[V_{\mathsf{RS}}^{\Omega_|(f)} = \mathsf{reject}\right] \geq \left(\frac{3}{8}\right)^d, \tag{6.1}$$

where the probability is over a uniformly random function $f : L \to \mathbb{F}$ *and the randomness of* V*.*

Proof. Consider first the case $d = 1$. For brevity denote $F = \Omega_|(f)$. The main point is that g is defined outside of $T(L)$, only as a function of the values of f on L, or equivalently, the values of F on $\{T_\beta(L)\}_{\beta \in R_{\hat{L}}}$. Thus, for random f and $\beta \notin R_{\hat{L}}$, the values of F on $T_\beta(L)$, and random and independent from the values $F|_{\mathrm{row}_\beta}$, for which there corresponds a unique $P \in \mathsf{RS}_{L_\beta}[2]$. Thus, the probability that $F_\beta^{\mathrm{ext}} \in \mathsf{RS}_{L_\beta}[2]$ is at most $1/|\mathbb{F}|$. A Markov argument show that with probability 0.9 on the choice of f, there will be at most a $10|R_{\hat{L}}|/|\mathbb{F}| < 1$ values $\beta \notin R_{\hat{L}}$ such that $F_\beta^{\mathrm{ext}} \in \mathsf{RS}_{L_\beta}[2]$; in which case V_{RS} will reject whenever he chooses a row extrow_β with $\beta \notin R_{\hat{L}}$, which happens with probability at least $1/2 \cdot (1 - 2^\eta) \geq 3/8$

Note that this argument in fact, only required that there is *at least one* element $(\alpha, \beta) \in T_\beta(L)$ such that $F(\alpha, \beta)$ is random and independent of other row values. This will be the case when V_{RS} always recursively chooses a row rather than column, and specifically a row that is not in $R_{\hat{L}}$. This is because, under our assumption on k and d, Remark 4.11 shows that when V_{RS} always recurses on rows for every β, the recursive row $L_{\beta_1 \ldots \beta_d}$ will always intersect $T(L)$, i.e., the values of the original $f : L \to \mathbb{F}$. □

6.3 Concrete security threshold of depth-2 PCPPs on random functions

To phrase our results on random functions, we give a formal definition of a PCPP system that is secure against *most* inputs x, when the prover is limited to a certain set of strategies for generating the auxiliary proof y.

Definition 6.8 (ϵ-PCPP for a code C against prespecified attacks)**.** *Fix integers $A, Q \in \mathbb{N}$ and $0 < \epsilon, \delta < 1$. Let $C \subseteq \mathbb{F}^n$ be an $[n = n(C), k = k(C), d = d(C)]$-code. Let $\mathcal{H}$ be a set of functions $h : \mathbb{F}^n \to \mathbb{F}^{A-n}$*

*An (A, Q, ϵ)-$\mathcal{H}$-*resistant PCPP system *$\mathcal{S}$ for C with soundness error δ is a pair $\mathcal{S} = (P, V)$, where*

- *P is a* systematic *mapping $P : C \to \mathbb{F}^A$.*

  That is, for any $x \in C$, $P(x) = (x, y)$ for some $y \in \mathbb{F}^{A-n}$.

- *V is a Q-*local *randomized mapping $V : \mathbb{F}^A \to \{\mathsf{accept}, \mathsf{reject}\}$. That is, after choosing its internal randomness, $V(z)$ always depends on at most Q indices of $z \in \mathbb{F}^A$.*

Such that

- *(Completeness) For any $x \in C$, $V(P(x)) =$ accept with probability one.*
- *(Soundness) For a $(1 - \epsilon)$-fraction of $x \in \mathbb{F}^n$, and any $y \in \mathbb{F}^{A-n}$ of the form $y = h(x)$ for some $h \in \mathcal{H}$, $V((x, y)) =$ accept with probability at most δ.*

We define a concrete efficiency threshold in this setting analogously to Definition B.5

Definition 6.9 (ϵ-Concrete efficiency threshold of a PCPP against prespecified attacks)**.** *We say an (A, Q, ϵ)-$\mathcal{H}$-*resistant PCPP system *$\mathcal{S}$ for C is* efficient *if the cost $A \cdot Q \leq k(C)^2/2$.*

*Fix an ensemble of linear codes $\mathcal{C} = \left\{C \subseteq \mathbb{F}^{n(C)}\right\}$, and functions $A, Q : \mathbb{N} \to \mathbb{N}$. An (A, Q, ϵ)-$\mathcal{H}$-*resistant PCPP system for *$\mathcal{C}$ is an ensemble of PCPP systems $\mathcal{S} = \{\mathcal{S}_C | C \in \mathcal{C}\}$ where $\mathcal{S}_C$ is an $(A(k(C)), Q(k(C)), \epsilon)$-$\mathcal{H}$-resistant PCPP system for C. The ϵ-*concrete efficiency threshold of *$\mathcal{S}$ is the smallest integer* $\mathbf{k}$ *such that for any $C \in \mathcal{C}$ of dimension $k(C) \geq \mathbf{k}$, $\mathcal{S}_C$ is efficient.*

For the sake of comparison with the improved concrete soundness studied in Section 4 and summarized in Table 1, we present here the concrete soundness threshold where soundness is measured on random functions using only the pair of attacks studied previously; for the sake of comparison, we fix the recursion depth to 2, as studied there. The bottom line is quite encouraging, showing a far better concrete threshold security, and stressing the importance of suggesting and analyzing other attacks on PCPP systems.

Corollary 6.10. *Fix positive integer d. Assume $|\mathbb{F}| \geq 512$. There is a $(2^{\ell+5+2.6\cdot(d-1)}, q_0(k, d) \cdot m, 0.1)$-PCPP system for* $\mathrm{RS_a}[3]$ *resistant to the row- and column-compliant attacks; where $m \triangleq \lceil \frac{\log(0.5)}{\log((5/8)^d)} \rceil$, for $k = \ell + 3$*

Proof. Fix a subspace $L \subseteq \mathbb{F}$ of dimension k. Fix $f : L \to F$. Let f' be the row-compliant or column-compliant depth d extension of f. Similarly to Lemma 4.15, we simply run the verifier from Definition 6.2 on f' m times and reject if one of the runs rejected. Using Lemma 6.5 and Lemma 6.7, with probability 0.9 over the choice of f each run rejects with probability at least $(3/8)^d$. As each run requires reading $q_0(k, d)$ entries of f', the claim follows. □

In particular,we can compare the improved concrete efficiency threshold to the concrete threshold with respect to the row- and column-compliant "attacks".

Corollary 6.11 (Security on random functions)**.** *Assume $|\mathbb{F}| \geq 512$. The BSS verifier from Section 4.1.2 has 0.1-concrete efficiency threshold of 2^{17} with respect to the row- and column-compliant pseudo-provers from Definition 6.3.*

We end by asking whether, for random $f : L \to \mathbb{F}$, the minimal rejection probability taken over all pseudo-proofs, is significantly smaller than the rejection probability with respect to row- and column-compliant pseudo-proofs. Currently, we cannot rule out the possibility that no better pseudo-proof exists!

message length	codeword length	proof length	# of queries
23	26	30.6	9.57
24	27	31.6	9.82
25	28	32.6	10.01
26	29	33.6	10.32
27	30	34.6	10.57
28	31	35.6	10.82
29	32	36.6	11.07
30	33	37.6	11.32
31	34	38.6	11.57
32	35	39.6	11.82
33	36	40.6	12.07
34	37	41.6	12.32
35	38	42.6	12.57

Table 2: Instantiations of the system in Corollary 6.10 with $d = 2$. All numbers are logs in base two of the described quantity. The first column describes the length (in field elements) of the message w to be encoded into a word $x \in \mathsf{RS}_L[3]$. The second column is the length of x. The third column is the length of x together with the proof y that $x \in \mathsf{RS}_L[3]$. The fourth is the number of field elements a verifier needs to read to be reject with probability $1/2$ for a 0.9 fraction of $x : L \to \mathbb{F}$.

7 Reducing proof length with interactive oracle proofs of proximity

In this section we show that, when allowing a few rounds of interaction between the prover and verifier, the efficiency of the PCPP can be significantly improved. This interaction uses the IOPP model as presented in [BCG$^+$16], based on the IOP model defined in [BSCS16, RRR16]. We give a tailored definition of IOPPs convenient for our purposes.

Definition 7.1. *Fix integers $A, Q \in \mathbb{N}$ Let $C \subseteq \mathbb{F}^n$ be an $[n = n(C), k = k(C), d = d(C)]$-code.An (A, Q)-IOPP system $\mathcal{S}$ for C is a pair $\mathcal{S} = (P, V)$, of players that run an interactive protocol, where*

- *The first message is a systematic mapping $P : C \to \mathbb{F}^{A_1}$*
- *The total size of messages sent by P is at most A.*
- *The total number of locations read by V from P's answers is $\leq Q$.*

Such that

- *(Completeness) For any $x \in C$, $V(P(x)) =$ accept with probability one.*
- *(Soundness) For any $z = (x, y) \in \mathbb{F}^A$ such that $\Delta(x, C) \geq d/3$, $V(z) =$ reject with probability at least $1/2$.*

Given $\epsilon > 0$, we also define an (A, Q, ϵ)-IOPP system in a similar way to Definition 6.8; that is, the soundness condition needs to hold for a $(1 - \epsilon)$-fraction of $x \in \mathbb{F}^n$, rather than x that is $d/3$-far from C.

Lemma 7.2. *Assume $|\mathbb{F}| \geq 512$. There is a $(4 \cdot 2^k + 13 \cdot 4 \cdot (2^{k/2+1.5} + 2^{k/4+2.25}), 13 \cdot 2^{k/8+2.25}, 0.1)$-IOPP system for $\mathrm{RS}_{\mathrm{a}}[3]$ with resistant to the row- and column-compliant "attacks", where $k = \ell + 3$*

Proof. The proof is similar to that used in the results of [BCG$^+$16] on IOPPs for RS codes (see Theorem 1.2 and Section 5 there), and we do not give a fully formal argument. Fix $g : L \to \mathbb{F}$. As in the proof of Corollary 6.10 we use the verifier from Section 4.1.2; here we fix depth $d = 3$. The difference is that know we are constructing an IOPP system so P only has to write down the depth one BSS-extension f of g at the start. Afterwards he will send the depth one extension f' of f^{col}_α or f^{ext}_β according to V's decision to query an extended row or column. As V is using depth 3, he will choose once more a depth one extension of a column or extended row of f' for P to send.

Thus, for each repetition, P will need to write at most $4 \cdot (2^{k/2+1.5} + 2^{k/4+2.25})$ elements in addition to writing f at the start, which has length 2^{k+2}.

□

message length	codeword length	proof length	# of queries
23	26	28.1	8.1
24	27	29.1	8.3
25	28	30.1	8.4
26	29	31.1	8.5
27	30	32.1	8.6
28	31	33.1	8.8
29	32	34.1	8.9
30	33	35.1	9
31	34	36.1	9.1
32	35	37.1	9.3
33	36	38.1	9.4
34	37	39.1	9.5
35	38	40.1	9.6

Table 3: Instantiations of the system in Lemma 7.2. All numbers are logs in base two of the described quantity. The first column describes the length (in field elements) of the message w to be encoded into a word $g \in \mathsf{RS}_L[3]$. The second column is the length of g. The third column is the total length of P's messages while proving $g \in \mathsf{RS}_L[3]$. The fourth is the number of field elements a verifier needs to read to be reject with probability $1/2$ for a 0.9 fraction of $g : L \to \mathbb{F}$.

.

References

[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, 1998. Preliminary version in FOCS '92.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: a new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998. Preliminary version in FOCS '92.

[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 *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, pages 90–108, 2013.

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

[BCGT13] Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, and Eran Tromer. On the concrete efficiency of probabilistically-checkable proofs. In *Proceedings of the 45th ACM Symposium on the Theory of Computing*, STOC '13, 2013.

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

[BFL90] László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I*, pages 16–25, 1990.

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, STOC '91, pages 21–32, 1991.

[BGH+06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. Robust PCPs of proximity, shorter PCPs, and applications to coding. *SIAM Journal on Computing*, 36(4):889–974, 2006. Preliminary versions of this paper have appeared in Proceedings of the 36th ACM Symposium on Theory of Computing and in Electronic Colloquium on Computational Complexity.

[BOGKW88] Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-prover interactive proofs: how to remove intractability assumptions. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, STOC '88, pages 113–131, 1988.

[BS93] Eli Biham and Adi Shamir. *Differential Cryptanalysis of the Data Encryption Standard*. Springer, 1993.

[BS08] Eli Ben-Sasson and Madhu Sudan. Short PCPs with polylog query complexity. *SIAM Journal on Computing*, 38(2):551–607, 2008. Preliminary version appeared in STOC '05.

[BSCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. Cryptology ePrint Archive, Report 2016/116, 2016. http://eprint.iacr.org/.

[BSVW03] Eli Ben-Sasson, Madhu Sudan, Salil Vadhan, and Avi Wigderson. Randomness-efficient low degree tests and short pcps via epsilon-biased sets. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, STOC '03, pages 612–621, 2003.

[CMT12] Graham Cormode, Michael Mitzenmacher, and Justin Thaler. Practical verified computation with streaming interactive proofs. In *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pages 90–112, 2012.

[Din07] Irit Dinur. The PCP theorem by gap amplification. *Journal of the ACM*, 54(3):12, 2007.

[DR04] Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the PCP theorem. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '04, pages 155–164, 2004.

[FGL+96] Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268–292, 1996. Preliminary version in FOCS '91.

[GGPR13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct nizks without pcps. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 626–645, 2013.

[GKL15] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The Bitcoin backbone protocol: Analysis and applications. In *Eurocrypt*, 2015. http://eprint.iacr.org/2014/765.

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

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems. In *STOC '85: Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pages 291–304, New York, NY, USA, 1985. ACM.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In *Proceedings of the 26th Annual International Conference on Advances in Cryptology*, CRYPTO '06, pages 97–111, 2006.

[Gro09] Jens Groth. Linear algebra with sub-linear zero-knowledge arguments. In *Proceedings of the 29th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '09, pages 192–208, 2009.

[Gro10] Jens Groth. Short non-interactive zero-knowledge proofs. In *Proceedings of the 16th International Conference on the Theory and Application of Cryptology and Information Security*, ASIACRYPT '10, pages 341–358, 2010.

[GS06] Oded Goldreich and Madhu Sudan. Locally testable codes and pcps of almost-linear length. *J. ACM*, 53(4):558–655, 2006.

[GS09] Vipul Goyal and Amit Sahai. Resettably secure computation. In *EUROCRYPT '09: Proceedings of the 28th Annual International Conference on Advances in Cryptology*, pages 54–71, Berlin, Heidelberg, 2009. Springer-Verlag.

[Hås01] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48:798–859, July 2001. Preliminary version in STOC '97.

[HS00] Prahladh Harsha and Madhu Sudan. Small PCPs with low query complexity. *Computational Complexity*, 9(3–4):157–201, Dec 2000. Preliminary version in STACS '91.

[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. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, STOC '92, pages 723–732, 1992.

[KMRS15] Swastik Kopparty, Or Meir, Noga Ron-Zewi, and Shubhangi Saraf. High-rate locally-testable codes with quasi-polylogarithmic query complexity. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:110, 2015.

[KR08] Yael Tauman Kalai and Ran Raz. Interactive PCP. In *ICALP '08: Proceedings of the 35th International Colloquium on Automata, Languages and Programming, Part II*, pages 536–547, Berlin, Heidelberg, 2008. Springer-Verlag.

[KR09] Yael Tauman Kalai and Ran Raz. Probabilistically checkable arguments. In *Advances in Cryptology-CRYPTO 2009*, pages 143–159. Springer, 2009.

[KRS15] Swastik Kopparty, Noga Ron-Zewi, and Shubhangi Saraf. High rate locally-correctable and locally-testable codes with sub-polynomial query complexity. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:68, 2015.

[LFKN92] Carsten Lund, Lance Fortnow, Howard Karloff, and Nisan Noam. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, 1992.

[Lip12] Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In *Proceedings of the 9th Theory of Cryptography Conference on Theory of Cryptography*, TCC '12, pages 169–189, 2012.

[Mat93] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, pages 386–397, 1993.

[Mic00] Silvio Micali. Computationally sound proofs. *SIAM Journal on Computing*, 30(4):1253–1298, 2000. Preliminary version appeared in FOCS '94.

[Mie08] Thilo Mie. Polylogarithmic two-round argument systems. *Journal of Mathematical Cryptology*, 2(4):343–363, 2008.

[MR08] Dana Moshkovitz and Ran Raz. Two-query PCP with subconstant error. *Journal of the ACM*, 57:1–29, June 2008. Preliminary version appeared in FOCS '08.

[Nak08] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Bitcoin.org*, 2008.

[PGHR13] Bryan Parno, Craig Gentry, Jon Howell, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. *IACR Cryptology ePrint Archive*, 2013:279, 2013.

[PS94] Alexander Polishchuk and Daniel A. Spielman. Nearly-linear size holographic proofs. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, STOC '94, pages 194–203, 1994.

[RRR16] Omer Reingold, Ron Rothblum, and Guy Rothblum. Constant-round interactive proofs for delegating computation. In *Proceedings of the 48th ACM Symposium on the Theory of Computing*, STOC '16, 2016.

[SBW11] Srinath Setty, Andrew J. Blumberg, and Michael Walfish. Toward practical and unconditional verification of remote computations. In *Proceedings of the 13th USENIX Conference on Hot Topics in Operating Systems*, HotOS '13, pages 29–29, 2011.

[SMBW12] Srinath Setty, Michael McPherson, Andrew J. Blumberg, and Michael Walfish. Making argument systems for outsourced computation practical (sometimes). In *Proceedings of the 2012 Network and Distributed System Security Symposium*, NDSS '12, pages ???–???, 2012.

[SVP+12] Srinath 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 21st USENIX Security Symposium*, Security '12, page ???, 2012.

[VSBW13] Victor Vu, Srujay Setty, Andrew J Blumberg, and Michael Walfish. A hybrid architecture for interactive verifiable computation. In *Security and Privacy (SP), 2013 IEEE Symposium on*, pages 223–237. IEEE, 2013.

A Defining IOPs and IOPPs using a notary

We assume familiarity with the standard models of *(i)* an interactive proof (IP) [GMR85] which involves two parties — a verifier V and prover P, *(ii)* a multiprover interactive proofs (MIP) [BOGKW88] that involves a single verifier and two non-communicating provers, and *(iii)* the PCP model which can be modeled by an Oracle Turing machine, i.e., a randomized Turing machine with oracle access to a fixed string. The IOP model [BSCS16, RRR16] does not fall under any of these three models: The prover needs to send long messages (one message per oracle) that the verifier does not want to explicitly recieve and store, so the IP and MIP models do not suffice. On the other hand, as the verifier receives a *sequence of adaptively selected oracles*, the model of an Oracle Turing machine is inadequate. To provide a clear and formal computational footing for IOPs and IOPPs, we provide the following definition that is a hybrid of an IP and an Oracle Turing machine. It is interactive, like IP, but introduces a third party, the *notary*, to provide the verifier with oracle access to long messages written by the prover interactively. The notary is also convenient for defining our complexity measures: We measure proof length as the number of bits sent from prover to notary, and query complexity as the number of bits sent from the notary to the verifier.

Definition A.1 (Notarized Interactive Proof (NIP))**.** *A notarized interactive proof (NIP) is an interactive protocol involving three parties: a prover* P*, a randomized verifier* V*, and* notary N*. The notary is a machine that (i)* records *long messages from the prover that are intended for the verifier, and (ii) answers verifier* queries *to previously recorded prover messages. In more detail:*

- **Notary** *The notary* N *has a counter* c *and a countably infinite number of infinitely long memory tapes. At the beginning of the protocol all tapes are blank and* $\mathsf{c} = 0$*. The notary is honest and operates (only) as specified below.*
- **Communication channels** *The verifier and notary use only broadcast channels, i.e., all their messages are seen by the prover. The prover has a private one-way error-less channel to the notary, called the* proof channel*.*
- **Interaction and querying** *Verifier broadcast messages are one of the following three types, that involve four special symbols:* $\mathsf{record}, \mathsf{query}, \mathsf{accept}, \mathsf{reject}$
  - **Record** *When* V *broadcasts a message* (record, n, s) *with* $n \in \mathbb{N}, s \in \Sigma^*$ *it instructs the notary to (i) increment* c *and (ii) record the next* n *bits arriving on the prover channel sequentially on tape number* c*. The auxiliary information* s *is intended for the prover (later it will used to send random bits to the prover). The verifier and notary know the rate of the proof channel and compute an upper bound on the time it will take for the* n*-bit message to be computed and delivered to the notary over the proof channel. After that time has passed, the notary will no longer record information on tape number* c *(unfilled entries remain blank).*
  - **Query** *When* V *broadcasts a message* (query, S) *with* S *a* query set*, a sequence of pairs of integers* $(i_1, j_1), \ldots, (i_q, j_q)$*, it instructs the notary to broadcast the sequence* $a_1, \ldots, a_q$ *where* a_k *is the symbol recorded in the* j_k *position of the* i_k *tape.*
  - **Verdict** *The protocol terminates with a single message broadcast that is either* accept *or* reject *and marks the end of the protocol.*
- **Complexity measures** *The* round complexity *is the number of* record *messages. The* proof length *is the sum of* n *values appearing in all* record *messages. The* query complexity *is the sum of sizes of query-sets. The* randomness complexity *is the sum of lengths of* s *strings appearing in "record" messages.*
- **Non-adaptivity and public randomness** *A protocol is* non-adaptive *if it includes a single* query *message (the query set may be large) and it is said to use* public randomness *if the verifier sends as* s *the next sequence of symbols appearing on its tape of randomness.*

This allows for an improved soundness in proofs of a given size and query complexity. To instantiate the notary in real-world setting, one can consider a decentralized consensus system such as Bitcoin [Nak08, GKL15], where the prover commits each required record into a Bitcoin block, and the verifier waits until the probability that this block could be reversed becomes negligible.

With this definition in hand, we can define the NIP analog of a PCPP.

Definition A.2 (NIPP)**.** *A code $\mathcal{C}$ of blocklength n_0 is said to have a* notarized interactive proof of proximity *(NIPP) with $r \geq 1$ rounds, q queries, length ℓ, soundness function $s : (0,1] \to [0,1]$ if there exists an r-round, q-query, length-ℓ* NIP *that satisfies the following constraints*

- **Initialization** *The first message of the verifier is $(\mathsf{record}, n_0, \emptyset)$, i.e., it involves no randomness. Let w_1 denote the first prover message (which is of length n_0).*
- **Completeness** *If $w_1 \in \mathcal{C}$ then there exist messages $w_2, \ldots, w_r$ that cause the verifier's verdict to be* accept *with probability* 1.
- **Soundness** *If w_1 is δ-far in relative hamming distance from $\mathcal{C}$ then for any sequence $w_2, \ldots, w_r$ of prover messages, the probability that V announces* reject *at the end of the protocol is at least $s(\delta)$; this probability is taken over the verifier's (random) messages.*

Remark A.3 (NIPP as a generalization of PCPP and LTC)**.** *A NIPP for a code $\mathcal{C}$ is a generalization of a PCPP for $\mathcal{C}$ which is, in turn, a generalization of a locally testable code (LTC). A PCPP system for $\mathcal{C}$ can be defined as a* 2*-round NIPP for $\mathcal{C}$ in which both* record *messages contain empty randomness strings. The first prover-message is the codeword, and the second is its PCPP. Similarly, a q-query LTC is a* 1*-round, q-query (non-adaptive) NIPP, in which the prover sends only the codeword in his first (and only) message.*

We mention again that the concepts of NIP and NIPP are equivalent to those of IOP and IOPP introduced in [BSCS16, BCG+16, RRR16], where [RRR16] use the term PCIP rather than IOP.

B Formal definition of feasible soundness

To discuss feasible soundness for concrete (as opposed to asymptotic) input lengths, we first need to address preliminary definitional issues.

Non-asymptotic statements and non-trivial input length For the purposes of discussion here a statement τ is called *asymptotic* if it is of the form "For every algorithm (in a computational class $\mathcal{C}$) there exists n_0 such that τ' holds for inputs of length $> n_0$." Notice this statement says nothing about concrete input lengths, and we are interested precisely in such inputs. To phrase a non-asymptotic statement, one that holds unconditionally for all n, we face problems such as the following:

- **Kolmogorov complexity** For any language L (e.g., the undecidable Halting Problem) and integer n_0 there exists a machine deciding $L \cap \{0,1\}^{n_0}$. If L has large Kolmogorov complexity then the length of the description of M is $\Omega(2^{n_0})$.
- **Exhaustive search** For any language $L \in \mathbf{NTIME}(T(n))$ and integer n_0 there exists a machine that runs in time $2^{n_0} \cdot O(n) = O(n)$ and decides $L \cap \{0,1\}^{n_0}$ correctly.

Both of these cases can be regarded as a special case of a polynomial time machine M such that the degree of its polynomial bound is high relative to n_0, hence M can use predefined hardcoded data or exhaustive search loop to decide inputs of length shorter than n_0. To avoid these issues we use the following definition.

Definition B.1 (Non-trivial input length)**.** *Let M be a Turing machine over alphabet Σ with a polynomial running time. The f-*trivial length threshold *of M, denoted $n_0(M, f)$, is the smallest n for which the following two conditions hold:*

1. *$|\Sigma|^{f(n)} > p_0(n)$, where p_0 is a polynomial.*
2. *p_0 bounds the running time of M, i.e., $\forall k$: M terminates in time $p_0(k)$ or less on inputs of length k.*

*Inputs of length at most $n_0(M, f)$ are called (M, f)-*trivial *and inputs of length greater than $n_0(M, f)$ are (M, f)-*nontrivial*. When f is unspecified we take it to be the identity function ($f(n) = n$) in which case we shall talk about M-trivial (or nontrivial) input length.*

In the above definition $f(n) = n$ can be regarded as exhaustive search over all witnesses of length n,

and for example $f(n) = \sqrt{n}$ can be considered for brute-force lookup over an inverse quadratic subset of the search space, i.e., subset of size $|\Sigma|^{\sqrt{n}}$.

Defining feasible soundness We assume familiarity with the standard definition of a PCP system (P, V) and use $\mathsf{err}(n)$ to denote the maximal soundness error of V on inputs of length n. The feasible error of such a system is the smallest error obtainable by an efficient procedure, disregarding its trivial inputs. For concreteness, a randomized algorithm R is a 2-tape Turing machine over alphabet $\{0, 1, \perp\}$ (with an auxiliary read-only tape for the randomness) and $|R|$ is the number of states in the transition function of A.

Definition B.2 (Feasible soundness error)**.** *Let $\mathcal{P}$ be a set of randomized polynomial time algorithms. A PCP system (P, V) for $L \in \mathbf{NTIME}(T(n))$ is said to have f-*feasible soundness error $\mathsf{err_f}^{\mathcal{P}} : \mathbb{N} \to [0, 1]$ against $\mathcal{P}$ *if for every $P^* \in \mathcal{P}$ and non-trivial input length $n > n_0(P^*, f)$, with all but probability $\mathsf{err_f}^{\mathcal{P}}(n)$ over $x \in \{0,1\}^n \setminus L$ and the internal randomness of P^*, $\mathsf{err}(V, x, P^*(x))$ is no greater than $\mathsf{err_f}^{\mathcal{P}}(n)$. Formally,*

$$\forall P^* \in \mathcal{P}, n > n_0(P^*, f), \quad \Pr_{x \in \{0,1\}^n \setminus L} \left[\mathsf{err}(V, x, P^*(x)) > \mathsf{err_f}^{\mathcal{P}}(n) \right] \leq \mathsf{err_f}^{\mathcal{P}}(n) \tag{B.1}$$

Remark B.3 (On the need for instance-distribution)**.** *It may seem better to define and study "worst-case feasible soundness", in which one quantifies universally over all instances x and replaces (B.1) with*

$$\forall P^* \in \mathcal{P}, n > n_0(P^*, f), x \in \{0,1\}^n \setminus L \quad \Pr \left[\mathsf{err}(V, x, P^*(x)) > \mathsf{err_f}^{\mathcal{P}}(n) \right] \leq \mathsf{err_f}^{\mathcal{P}}(n). \tag{B.2}$$

The probability above is (only) over the randomness of P^*. This approach is problematic, because there may exist "pathological" examples of $x \notin L$ for which a high-soundness-error pseudo-proof can be found efficiently. Thus, we prefer a definition that seems immune to pathological pseudo-provers that succeed only on pathological instances.

The distribution on instances in (B.1) can be naturally replaced with a different distribution μ, say, one that "simulates" the distribution on "typical instances arising in practice".

For any PCP system we have $\mathsf{err}(n) \geq \mathsf{err_f}(n)$ because $\mathsf{err}(n)$ is the maximum error taken over all pseudo-proofs whereas $\mathsf{err_f}(n)$ is the maximum error over all *efficiently computable* pseudo-proofs. Computing $\mathsf{err_f}(n)$ is beyond our current ability because it requires bounding the error over all efficient, polynomial-time, pseudo-proofs (for nontrivial input lengths). Moreover, a strict inequality in (B.1) implies that $\mathbf{P} \neq \mathbf{NP}$ because otherwise there exists a polynomial time algorithm that searches for the proof with maximal soundness error. When $\mathcal{P}$ is the full set of randomized polynomial time algorithms we also say that the PCP system $S = (P, V)$ has *absolute feasible soundness* $\mathsf{err_f}^{\mathcal{P}}(n)$. On the other hand, when $\mathcal{P} = \mathcal{P}_{\mathsf{date=DD/MM/YYYY}}$ is the set of "published" algorithms (say, those described on the web) by date $\mathsf{DD/MM/YYYY}$ we refer to $-\log \mathsf{err_f}^{\mathcal{P}}(n)$ as the *security level* of the PCP system S. E.g., saying $\mathsf{err_f}^{\mathcal{P}_{\mathsf{date=1/1/2016}}}(2^{30}) = 2^{-128}$ means that for the particular PCP system S, fixing any pseudo-proof P^* posted on the web by the end of 2015 (that runs in time $< n^{2^{30}}$ and has program length less than $2^{2^{30}}$) for all but a 2^{-128}-fraction of unsatisfiable $x, |x| = 2^{30}$, has probability that V accepts x, when given oracle access to $P^*(x)$, is at most 2^{-128}.

Remark B.4 (Absolutely trivial inputs)**.** *There exists a constant c such that every polynomial time machine M requires a description of length $> 2^c$. Inputs of length $< c$ can be called "absolutely trivial", as they are trivial for all polynomial time machines. The value of c is likely very small (in particular $c < 20$ seems likely), so much so that shorter inputs are of no practical (or theoretical) interest.*

We shall also be interested in the feasible soundness of PCPs of proximity for families of error correcting codes, because the feasible soundness of the PCP systems we study is tightly related to the feasible soundness of quasilinear PCPPs for RS codes. The definition below generalizes that of the concrete soundness threshold of a PCPP system (Definition 1.2).

Definition B.5 (Feasible soundness of PCPPs and its concrete threshold)**.** *Fix an ensemble of linear codes $\mathcal{C} = \left\{ C \subseteq \mathbb{F}^{n(C)} \right\}$, and functions $L, Q : \mathbb{N} \to \mathbb{N}$. An (L, Q)-*PCPP system for $\mathcal{C}$ *is an ensemble of PCPP systems $\mathcal{S} = \{\mathcal{S}_C | C \in \mathcal{C}\}$ where $\mathcal{S}_C$ is an $(L(k(C)), Q(k(C)))$-PCPP system for C (cf. Definition 6.8).*

We say $\mathcal{S}$ has feasible soundness error $\mathsf{err_f}^{\mathcal{P}} : \mathbb{N} \times [0,1] \to [0,1]$ *with respect to a set $\mathcal{P}$ of randomized polynomial time "pseudo-provers" if for every $P^* \in \mathcal{P}$ and $C \in \mathcal{C}$ of non-trivial blocklength $n = n(C) > n_0(P^*)$ (cf. Definition B.1) with all but probability $\mathsf{err_f}^{\mathcal{P}}(n)$ over $x \in \mathbb{F}^n \setminus C$ and the internal randomness of P^*, $\mathsf{err}(V, x, P^*(x))$ is no greater than $\mathsf{err_f}^{\mathcal{P}}(n, \Delta(x, C))$. Formally,*

$$\forall P^* \in \mathcal{P}, n > n_0(P^*), \quad \Pr_{x \in \mathbb{F}^n \setminus C}\left[\mathsf{err}(V, x, P^*(x)) > \mathsf{err_f}^{\mathcal{P}}(n, \Delta(x, C))\right] \leq \mathsf{err_f}^{\mathcal{P}}(n, \Delta(x, C)) \tag{B.3}$$

B.1 Defining feasible soundness of PCPPs

We now give a definition of feasible concrete soundness of PCPP systems; it uses the notion of nontrivial input length (Definition B.1).

Definition B.6 (Feasible concrete soundness)**.** *Fix integers $A, Q \in \mathbb{N}$. Let $\mathcal{P}$ be a set of polynomial time (randomized) algorithms. Let $C \subseteq \mathbb{F}^n$ be an $[n = n(C), k = k(C), d = d(C)]$-code. $\mathcal{S} = (P, V)$ is called an* (A, Q, ϵ)-PCPP system for C with respect to $\mathcal{P}$ *(cf. Definition 6.8) if it satisfies the first three bullets of Definition 1.1 and the following weaker soundness condition:*

- *($\mathcal{P}$-feasible soundness) For any $P^* \in \mathcal{P}$ such that n is a non-trivial input for P^*, with probability at least $1 - \epsilon$ over $x \in \mathbb{F}^n$,*

$$\Pr[V(x, y = P^*(x)) = \mathsf{reject}] \geq 1/2. \tag{B.4}$$

We say $\mathcal{S}$ is ϵ-efficient on C with respect to $\mathcal{P}$ *if furthermore $A \cdot Q \leq k(C)^2/2$. A PCPP system for an ensemble $\mathcal{C}$ of codes has* ϵ-concrete feasible efficiency threshold $\mathbf{k}$ *with respect to $\mathcal{P}$ if for any $C \in \mathcal{C}$ of dimension $k(C) \geq \mathbf{k}$, $\mathcal{S}_C$ is ϵ-efficient on C with respect to $\mathcal{P}$.*

C Size of a BSS set

Claim C.1. *Fix $L \subseteq \mathbb{F}$ of dimension k. Then, $|\Omega_d(L)| \leq 2^{k+2+3\cdot(d-1)}$.*

Furthermore, if $\Omega_d(L)$ is defined as in Definition 4.10 but while taking depth $d-1$ extensions only of columns and extended *rows, i.e. omitting the union over the sets $(\alpha, \Omega_{d-1}(L'_0))$ in the definition, then $|\Omega_d(L)| \leq 2^{k+2+2.6\cdot(d-1)}$*

Proof. We use induction on d.
Base: As $\Omega_1(L) = \Omega(L)$ is a union of row_β over $\beta \in L_1$, and $|\mathrm{row}_\beta| = |L_\beta| = 2 \cdot |L'_0|$ We have

$$|\Omega(L)| \leq |L_1| \cdot 2 \cdot |L'_0| \leq 4 \cdot |L| = 2^{k+2}.$$

Induction step:

$$|\Omega_d(L)| \leq |L'_0| \cdot |\Omega_{d-1}(L'_1)| + |L_1| \cdot (|\Omega_{d-1}(L'_0)| + |\Omega_{d-1}(L_\beta)|)$$

Using induction hypothesis

$$\leq (|L'_0| \cdot |L'_1| + |L_1| \cdot (|L'_0| + |L_\beta|)) \cdot 2^{2+3\cdot(d-2)}$$

Using $|L| = |L'_0| \cdot |L'_1| = |L_1| \cdot |L'_0|$ and $|L_\beta| = 2 \cdot |L'_0|$

$$= 8 \cdot |L| \cdot 2^{2+3\cdot(d-2)} = 2^{k+2+3\cdot(d-1)}.$$

The "furthermore" case would follow by a similar induction step, while omitting the $|L_1| \cdot |L'_0|$ term:

$$\begin{aligned}|\Omega_d(L)| &\leq |L'_0| \cdot |\Omega_{d-1}(L'_1)| + |L_1| \cdot |\Omega_{d-1}(L_\beta)| \\ &\leq (|L'_0| \cdot |L'_1| + |L_1| \cdot |L_\beta| \cdot 2^{2+2.6\cdot(d-2)} \\ &= 6 \cdot |L| \cdot 2^{2+2.6\cdot(d-2)} \leq 2^{k+2+2.6\cdot(d-1)},\end{aligned}$$

where we used $6 \leq 2^{2.6}$ in the last inequality.

□

D Set visualization

The purpose of this appendix is to give a concrete example of a BSS set, with a visualization, in order to help clarify Definition 4.1.

Let $L \subset \mathbb{F}$ be an $\mathbb{F}_2$-subspace of dimension $k = 5$, and let $\{b_1, \cdots, b_5\}$ be its basis. We visualize the set of elements L as an ordered vector of points. We think of the indices of the points as the boolean vectors in $\{0,1\}^5$. Each index $(a_1, \ldots, a_5)$ represents the element $\sum_{i=1}^{5} a_i \cdot b_i$ of L; and they are ordered by the standard alphanumeric order (see figure D.1).

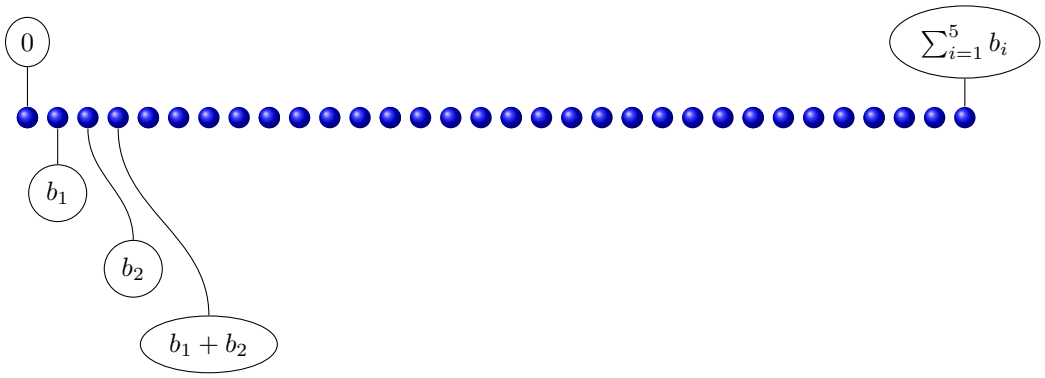


Figure D.1: Visualization of the vector space L

We view this space as a direct sum $L = L_0 \oplus L_1$ where $L_0 = \text{span}\,(b_1, b_2)$ and $L_1 = \text{span}\,(b_3, b_4, b_5)$.

A convenient way to visualize these spaces is thinking of L as the union of L_0 cosets shifted by L_1 elements as we see in figure D.2, where the beginning of a new coset is indicated by a change in color.

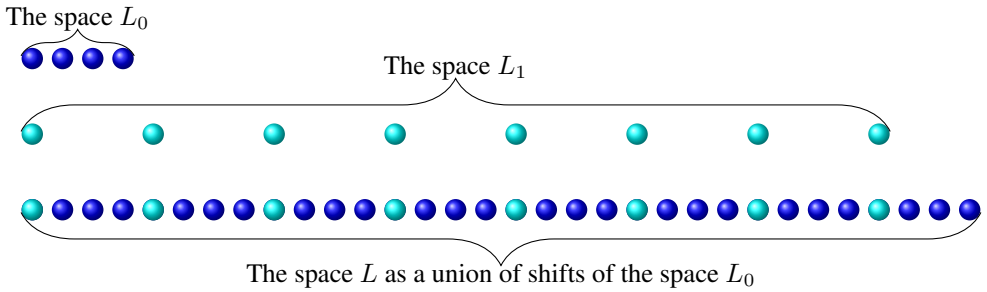


Figure D.2: Visualization of the spaces L_0 and L_1 and L as their direct sum. Each cyan node in L indicates the "beginning" of a new affine shift of L_0 by some element of L_1

The space L_0' is $\text{span}\,(b_1, b_2, b_3)$ and is the common subset of all the spaces L_β for all $\beta \in L_1$. The space L_β is defined as $L_\beta = \text{span}\,(b_1, b_2, b_3, \beta)$ for all $\beta \notin L_0'$, and otherwise $L_\beta = \text{span}\,(b_1, b_2, b_3, b_4)$.

An illustration of the construction of L_β for arbitrary value of β can be seen in figure D.3.

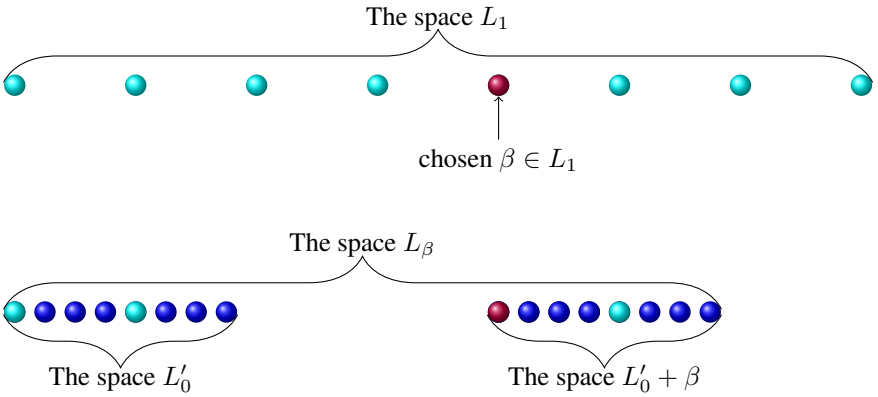


Figure D.3: Visualization of the spaces L'_0 and L_β

For all $\beta \in L_1$ we define the set $\mathrm{extrow}_\beta \subset \mathbb{F}^2$ to be the embedding of L_β into $\mathbb{F}^2$ using the mapping ϕ_β that is defined by $\forall \alpha \in L_\beta : \phi_\beta(\alpha) = (\alpha, q_{L_0}(\beta))$. The BSS set $\Omega(L)$ is defined as the union of extrow_β for all $\beta \in L_1$. We embed the space L into $\Omega(L)$ using the mapping ϕ_Ω that is defined by $\forall \alpha \in L_0, \beta \in L_1 : \phi_\Omega(\alpha + \beta) = \phi_\beta(\alpha)$ (see figure D.4).

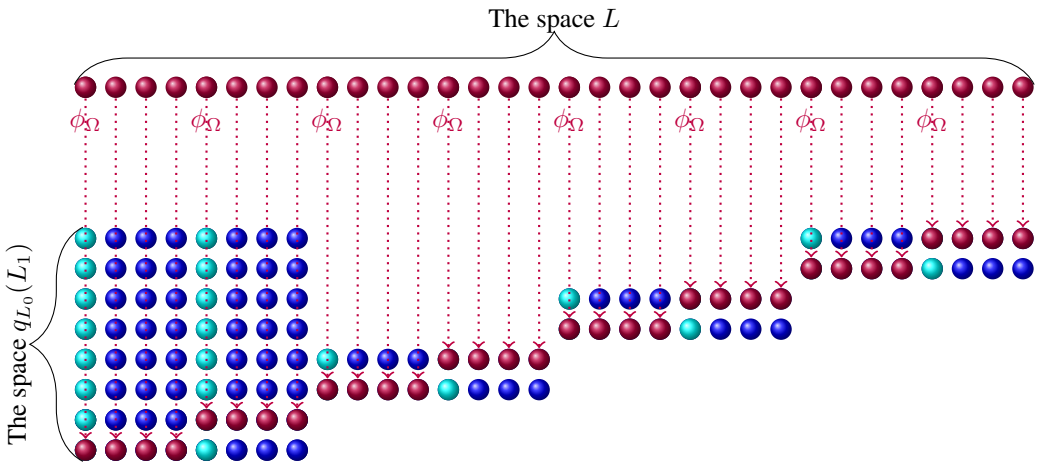


Figure D.4: Visualization of $\Omega(L)$ and the embedding of L into it. The subset of $q_{L_0}(L_1) \times L$ is $\Omega(L)$, and the purple arrows from the space L into $\Omega(L)$ illustrate the embedding ψ_Ω. The purple elements in $\Omega(L)$ is the image of this embedding, which is denoted $T(L)$ in Definition 4.1. The product set $S(L) = q_{L_0}(L_1) \times L'_0$ can be easily seen as the block of full columns at the left part of $\Omega(L)$.

The set $\overline{\Omega}(L)$ is defined as $\Omega(L) \setminus T(L)$ and visually illustrated in figure D.5.

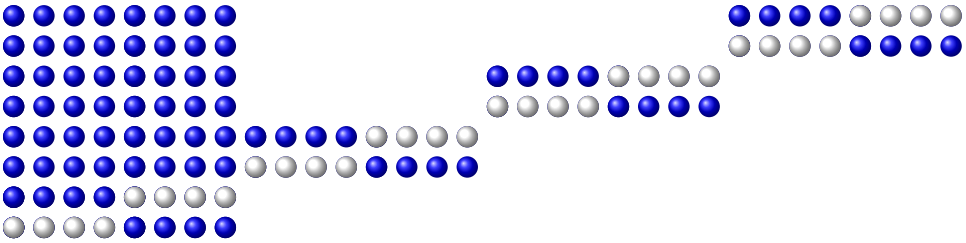


Figure D.5: Illustration of the set $\overline{\Omega}(L)$. The elements in the set are blue. The remaining elements are the set $T(L)$ from Definition 4.1.

ECCC ISSN 1433-8092
http://eccc.hpi-web.de