On Approximating the Entropy of Polynomial Mappings

Zeev Dvir[1] Dan Gutfreund[2] Guy N. Rothblum[1] Salil Vadhan[3]

[1]Center for Computational Intractability and Department of Computer Science,
Princeton University, 35 Olden Street, Princeton, NJ, USA
[2]IBM Research, Haifa,Israel
[3]School of Engineering and Applied Sciences & Center for Research on Computation and Society,
Harvard University, 33 Oxford Street, Cambridge, MA, USA
zeev.dvir@gmail.com danny.gutfreund@gmail.com rothblum@alum.mit.edu salil@seas.harvard.edu

Abstract: We investigate the complexity of the following computational problem:

> Polynomial Entropy Approximation (PEA): Given a low-degree polynomial mapping $p : \mathbb{F}^n \to \mathbb{F}^m$, where $\mathbb{F}$ is a finite field, approximate the output entropy $\mathrm{H}(p(U_n))$, where U_n is the uniform distribution on $\mathbb{F}^n$ and H may be any of several entropy measures.

We show:

- Approximating the Shannon entropy of degree 3 polynomials $p : \mathbb{F}_2^n \to \mathbb{F}_2^m$ over $\mathbb{F}_2$ to within an additive constant (or even $n^{.9}$) is complete for $\mathbf{SZKP_L}$, the class of problems having statistical zero-knowledge proofs where the honest verifier and its simulator are computable in logarithmic space. ($\mathbf{SZKP_L}$ contains most of the natural problems known to be in the full class $\mathbf{SZKP}$.)
- For prime fields $\mathbb{F} \neq \mathbb{F}_2$ and *homogeneous* quadratic polynomials $p : \mathbb{F}^n \to \mathbb{F}^m$, there is a probabilistic polynomial-time algorithm that distinguishes the case that $p(U_n)$ has entropy smaller than k from the case that $p(U_n)$ has min-entropy (or even Renyi entropy) greater than $(2 + o(1))k$.
- For degree d polynomials $p : \mathbb{F}_2^n \to \mathbb{F}_2^m$, there is a polynomial-time algorithm that distinguishes the case that $p(U_n)$ has *max-entropy* smaller than k (where the max-entropy of a random variable is the logarithm of its support size) from the case that $p(U_n)$ has max-entropy at least $(1 + o(1)) \cdot k^d$ (for fixed d and large k).

Keywords: cryptography, computational complexity, algebra, entropy, statistical zero knowledge, randomized encodings

1 Introduction

We consider the following computational problem:

> Polynomial Entropy Approximation (PEA): Given a low-degree polynomial mapping $p : \mathbb{F}^n \to \mathbb{F}^m$, where $\mathbb{F}$ is a finite field, approximate the output entropy $\mathrm{H}(p(U_n))$, where U_n is the uniform distribution on $\mathbb{F}^n$.

In this paper, we present some basic results on the complexity of PEA, and suggest that a better understanding might have significant impact in computational complexity and the foundations of cryptography.

Note that PEA has a number of parameters that can be varied: the degree d of the polynomial mapping, the size of the finite field $\mathbb{F}$, the quality of approximation (eg multiplicative or additive), and the measure of entropy (eg Shannon entropy or min-entropy). Here we are primarily interested in the case where the degree d is bounded by a fixed constant (such as 2 or 3), and the main growing parameters are n and m. Note that in this case, the polynomial can be specified explicitly by $m \cdot \mathrm{poly}(n)$ coefficients, and thus "polynomial time" means $\mathrm{poly}(m, n, \log |\mathbb{F}|)$.

Previous results yield polynomial-time algorithms for PEA in two special cases:

Exact Computation for Degree 1: For polynomials $p : \mathbb{F}^n \to \mathbb{F}^m$ of degree at most 1, we can write $p(x) = Ax + b$ for $A \in \mathbb{F}^{m\times n}$ and $b \in \mathbb{F}^n$. Then $p(U_n)$ is uniformly distributed on the affine subspace $\mathrm{Image}(A) + b$, and thus has entropy exactly $\log |\mathrm{Image}(A)| = \mathrm{rank}(A) \cdot \log |\mathbb{F}|$.

Multiplicative Approximation over Large Fields: In their work on randomness extractors for polynomial sources, Dvir, Gabizon, and Wigderson [13] related the entropy of $p(U_n)$ to the rank of the Jacobian matrix $J(p)$, whose (i,j)'th entry is the partial derivative $\partial p_i/\partial x_j$, where p_i is the i'th component of p. Specifically, they showed that the min-entropy of $(p(U_n))$ is essentially within a $(1+o(1))$-multiplicative factor of $\text{rank}(J(p)) \cdot \log|\mathbb{F}|$, where the rank is computed over the polynomial ring $\mathbb{F}[x_1,\ldots,x_n]$. This tight approximation holds over prime fields of size exponential in n. Over fields that are only mildly large (say, polynomial in n) the rank of the Jacobian still gives a one-sided approximation to the entropy.

In this paper, we study PEA for polynomials of low degree (namely 2 and 3) over small fields (especially the field $\mathbb{F}_2$ of two elements). Our first result characterizes the complexity of achieving good additive approximation:

Theorem 1.1(informal). *The problem* $\text{PEA}^{+}_{\mathbb{F}_2,3}$ *of approximating the Shannon entropy of degree 3 polynomials* $p : \mathbb{F}_2^n \to \mathbb{F}_2^m$ *to within an additive constant (or even* $n^{.9}$*) is complete for* $\mathbf{SZKP_L}$*, the class of problems having statistical zero-knowledge proofs where the honest verifier and its simulator are computable in logarithmic space (with two-way access to the input, coin tosses, and transcript).*[1]

In particular, the output entropy approximation problem is at least as hard as Graph Isomorphism, Quadratic Residuosity, the Discrete Logarithm, and the approximate Closest Vector Problem, as the known statistical zero-knowledge proofs for these problems [16-18,22] have verifiers and simulators that can be computed in logarithmic space.

Theorem 1.1 is proven by combining the reductions for known **SZKP**-complete problems [21,35] with the randomized encodings developed by Applebaum, Ishai, and Kushilevitz in their work on cryptography in $\mathbf{NC}^0$ [6,23]. Moreover, the techniques in the proof can also be applied to the specific natural complete problems mentioned above, and most of these each reduce to special cases of $\text{PEA}^{+}_{\mathbb{F}_2,3}$ that may be easier to solve (e.g. ones where the output distribution is uniform on its support, and hence all entropy measures coincide).

The completeness of $\text{PEA}^{+}_{\mathbb{F}_2,3}$ raises several intriguing (albeit speculative) possibilities:

Combinatorial or Number-Theoretic Complete Problems for $\mathbf{SZKP_L}$:

Ever since the first identification of complete problems for **SZKP** (standard statistical zero knowledge, with verifiers and simulators that run in polynomial time rather than logarithmic space) [35], it has been an open problem to find combinatorial or number-theoretic complete problems. Previously, all of the complete problems for **SZKP** and other zero-knowledge classes (e.g. [8,10,12,20,21,28,35,36]) refer to estimating statistical properties of arbitrary efficiently samplable distributions (namely, distributions sampled by boolean circuits). Moving from a general model of computation (boolean circuits) to a simpler, more structured model (degree 3 polynomials) is a natural first step to finding other complete problems, similarly to how the reduction from CircuitSAT to 3-SAT is the first step towards obtaining the wide array of known **NP**-completeness results. (In fact we can also obtain a complete problem for $\mathbf{SZKP_L}$ where each output bit of $p : \mathbb{F}_2^n \to \mathbb{F}_2^m$ depends on at most 4 input bits, making the analogy to 3-SAT even stronger.)

Cryptography Based on the Worst-case Hardness of $\mathbf{SZKP_L}$:

It is a long-standing open problem whether cryptography can be based on the worst-case hardness of **NP**. That is, can we show that $\mathbf{NP} \not\subseteq \mathbf{BPP}$[2] implies the existence of one-way functions? A positive answer would yield cryptographic protocols for which we can have much greater confidence in their security than any schemes in use today, as efficient algorithms for all of **NP** seems much more unlikely than an efficient algorithm for any of the specific problems underlying present-day cryptographic protocols (such as Factoring). Some hope was given in the breakthrough work of Ajtai [4], who showed that the worst-case hardness of an approximate version of the Shortest Vector Problem implies the existence of one-way functions (and in fact, collision-resistant hash functions). Unfortunately, it was shown that this problem is unlikely to be **NP**-hard [1,16,27]. In fact, there are more general results, showing that there cannot be (nonadaptive, black-box) reductions from breaking a one-way function to solving any **NP**-complete prob-

[1] See Sections 3 and 4 for the formal definitions of the notions involved and the formal statement of the theorem.

[2] Or $\mathbf{NP} \not\subseteq \mathbf{i.o.\!-\!BPP}$, where $\mathbf{i.o.\!-\!BPP}$ is the class of problems that can be solved in probabilistic polynomial time for infinitely many input lengths.

lem (assuming $\mathbf{NP} \not\subseteq \mathbf{coAM}$) [5,9,14].

We observe that these obstacles for **NP** do not apply to **SZKP** or $\mathbf{SZKP_L}$, as these classes are already contained in $\mathbf{AM} \cap \mathbf{coAM}$ [3,15]. Moreover, being able to base cryptography on the hardness of **SZKP** or $\mathbf{SZKP_L}$ would also provide cryptographic protocols with a much stronger basis for security than we have at present — these protocols would be secure if *any* of the variety of natural problems in $\mathbf{SZKP_L}$ are worst-case hard (e.g. Quadratic Residuosity, Graph Isomorphism, Discrete Logarithm, the approximate Shortest Vector Problem).

Our new complete problem for $\mathbf{SZKP_L}$ provides natural approaches to basing cryptography on **SZKP**-hardness. First, we can try to reduce $\mathrm{PEA}^{+}_{\mathbb{F}_2,3}$ to the approximate Shortest Vector Problem, which would suffice by the aforementioned result of Ajtai [4]. Alternatively, we can try to exploit the algebraic structure in $\mathrm{PEA}^{+}_{\mathbb{F}_2,3}$ to give a worst-case/average-case reduction for it (i.e. reduce arbitrary instances to random ones). This would show that if $\mathbf{SZKP_L}$ is worst-case hard, then it is also average-case hard. Unlike **NP**, the average-case hardness of **SZKP** is known to imply the existence of one-way functions by a result of Ostrovsky [33], and in fact yields even stronger cryptographic primitives such as constant-round statistically hiding commitment schemes [32,34].

New Algorithms for $\mathbf{SZKP_L}$ Problems:

On the flip side, the new complete problem may be used to show that problems in $\mathbf{SZKP_L}$ are *easier* than previously believed, by designing new algorithms for PEA. As mentioned above, nontrivial polynomial-time algorithms have been given in some cases via algebraic characterizations of the entropy of low-degree polynomials (namely, the Jacobian rank) [13]. This motivates the search for tighter and more general algebraic characterizations of the output entropy, which could be exploited for algorithms or for worst-case/average-case connections. In particular, this would be a very different way of trying to solve problems like Graph Isomorphism and Quadratic Residuosity than previous attempts. One may also try to exploit the complete problem to give a *quantum* algorithm for $\mathbf{SZKP_L}$. Aharonov and Ta-Shma [2] showed that all of **SZKP** would have polynomial-time quantum algorithms if we could solve the Quantum State Generation (QSG) problem: given a boolean circuit $C : \{0,1\}^m \to \{0,1\}^n$, construct the quantum state $\sum_x |C(x)\rangle$. Using our new complete problem, if we can solve QSG even in the special case that C is a degree 3 polynomial over $\mathbb{F}_2$, we would get quantum algorithms for all of $\mathbf{SZKP_L}$ (including Graph Isomorphism and the approximate Shortest Vector Problem, which are well-known challenges for quantum computing).

While each of these potential applications may be remote possibilities, we feel that they are important enough that any plausible approach is worth examining.

Our Algorithmic Results.

Motivated by the above, we initiate a search for algorithms and algebraic characterizations of the entropy of low-degree polynomials over small finite fields (such as $\mathbb{F}_2$), and give the following partial results:

- For degree d (multilinear) polynomials $p : \mathbb{F}_2^n \to \mathbb{F}_2^m$, the rank of the Jacobian $J(p)$ (over $\mathbb{F}_2[x_1, \ldots, x_n]$) does not provide better than a $2^{d-1} - o(1)$ multiplicative approximation to the entropy $\mathrm{H}(p(U_n))$. Indeed, the polynomial mapping
$$p(x_1, \ldots, x_n, y_1, \ldots, y_{d-1}) = (y_1 y_2 \cdots y_{d-1}) \cdot (x_1, x_2, \ldots, x_n)$$
has Jacobian rank n but output entropy smaller than $n/2^{d-1} + 1$.

- For prime fields $\mathbb{F} \neq \mathbb{F}_2$ and *homogeneous* quadratic polynomials $p : \mathbb{F}^n \to \mathbb{F}^m$, there is a probabilistic polynomial-time algorithm that distinguishes the case that $p(U_n)$ has entropy smaller than k from the case that $p(U_n)$ has min-entropy (or even Renyi entropy) greater than $(2 + o(1))k$. This algorithm is based on a new formula for the Renyi entropy of $p(U_n)$ in terms of the rank of random directional derivatives of p.

- For degree d polynomials $p : \mathbb{F}_2^n \to \mathbb{F}_2^m$, there is a polynomial-time algorithm that distinguishes the case that $p(U_n)$ has *max-entropy* smaller than k (where the max-entropy of a random variable is the logarithm of its support size) from the case that $p(U_n)$ has max-entropy at least $(1 + o(1)) \cdot k^d$ (for fixed d and large k). This algorithm is based on relating the max-entropy to the dimension of the $\mathbb{F}_2$-span of the p's components $p_1, \ldots, p_m \in \mathbb{F}_2[x_1, \ldots, x_n]$.

While our algorithms involve entropy measures other than Shannon entropy (which is what is used in the $\mathbf{SZKP_L}$-complete problem $\mathrm{PEA}^+_{\mathbb{F}_2,3}$), recall that many of the natural problems in $\mathbf{SZKP_L}$ reduce to special cases where we can bound other entropy measures such as max-entropy or Renyi entropy. See Section 4.4.

2 Preliminaries and notations

For two discrete random variables X, Y taking values in S, their *statistical difference* is defined to be $\Delta(X,Y) \overset{\text{def}}{=} \max_{T\subseteq S} |\Pr[X \in S] - \Pr[Y \in S]|$. We say that X and Y are *ε-close* if $\Delta(X,Y) \leqslant \varepsilon$. The *collision probability* of X is defined to be $\mathrm{cp}(X) \overset{\text{def}}{=} \sum_x \Pr[X = x]^2 = \Pr[X = X']$, where X' is an iid copy of X. The *support* of X is $\mathrm{Supp}(X) \overset{\text{def}}{=} \{x \in S : \Pr[X = x] > 0\}$. X is *flat* if it is uniform on its support.

For a function $f : S^n \to T^m$, we write $f_i : S^n \to T$ for the i'th component of f. When S is clear from context, we write U_n to denote the uniform distribution on S^n, and $f(U_n)$ for the output distribution of f when evaluated on a uniformly chosen element of S^n. The *support* of f is defined to be $\mathrm{Supp}(f) \overset{\text{def}}{=} \mathrm{Supp}(f(U_n)) = \mathrm{Image}(f)$.

For a prime power $q = p^t$, $\mathbb{F}_q$ denotes the (unique) finite field of size q. For a mapping $P : \mathbb{F}_q^n \mapsto \mathbb{F}_q^m$, we say that P is a polynomial mapping if each P_i is a polynomial (in n variables). The *degree* of P is $\deg(P) = \max_i \deg(P_i)$.

Notions of Entropy.

Throughout this work we consider several different notions of entropy, or the “amount of randomness” in a random variable. The standard notions of Shannon Entropy, Renyi Entropy, and Min-Entropy are three such notions. We also consider the (log) support size, or maximum entropy, as a (relaxed) measure of randomness.

Definition 2.1. *For a random variable X taking values in a set S, we consider the following notions of entropy:*

- *Min-entropy:* $\mathrm{H_{min}}(X) \overset{\text{def}}{=} \min_{x\in S} \log \frac{1}{\Pr[X=x]}$.
- *Renyi entropy:* $\mathrm{H_{Renyi}}(X) \overset{\text{def}}{=} \log \frac{1}{\mathrm{E}_{x \overset{\mathrm{R}}{\leftarrow} X}[\Pr[X=x]]} = \log \frac{1}{\mathrm{cp}(X)}$.
- *Shannon-entropy:* $\mathrm{H_{Shannon}}(X) \overset{\text{def}}{=} \mathrm{E}_{x \overset{\mathrm{R}}{\leftarrow} X}\left[\log \frac{1}{\Pr[X=x]}\right]$.
- *Max-entropy:* $\mathrm{H_{max}}(X) \overset{\text{def}}{=} \log |\mathrm{Supp}(X)|$.

(All logarithms are base 2 except when otherwise noted.)

These notions of entropy are indeed increasingly relaxed, as shown in the following claim:

Claim 2.2. *For every random variable X it holds that*

$$\begin{gathered} 0 \leqslant \mathrm{H_{min}}(X) \leqslant \mathrm{H_{Renyi}}(X) \\ \leqslant \mathrm{H_{Shannon}}(X) \leqslant \mathrm{H_{max}}(X). \end{gathered}$$

Moreover, if X is flat, all of the entropy measures are equal to $\log |\mathrm{Supp}(X)|$.

3 Entropy difference and polynomial entropy difference

In this section we define the Entropy Difference and Polynomial Entropy Difference problems, which are the focus of this work.

Entropy Difference Promise Problems.

The promise problem Entropy Difference (ED) deals with distinguishing an entropy gap between two random variables represented as explicit mappings, computed by circuits or polynomials, and evaluated on a uniformly random input. In this work, we consider various limitations on these mappings both in terms of their computational complexity and their degree (when viewed as polynomials).

In what follows $\mathcal{C}$ will be a concrete computational model, namely every $c \in \mathcal{C}$ computes a function with finite domain and range. Examples relevant to this paper include:

- The class Circ of all boolean circuits $C : \{0,1\}^n \to \{0,1\}^m$ (the concrete model corresponding to polynomial time).
- The class BP of all branching programs $C : \{0,1\}^n \to \{0,1\}^m$ (the concrete model corresponding to logarithmic space)
- The class $\mathrm{POLYNOMIALS}_{\mathbb{F},d}$ of all degree d polynomials $p : \mathbb{F}^n \to \mathbb{F}^m$.

The promise problem ED [21] is defined over pairs of random variables represented as mappings computed by boolean circuits, where the random variables are the outputs of the circuits evaluated on uniformly chosen inputs. The problem is to determine which of the two random variables has more Shannon entropy, with a promise that there is an additive gap between the two entropies of at least 1. We generalize this promise problem to deal with different notions of entropies, entropy gaps, and the complexity of the mappings which represent the random variables.

Definition 3.1(Generalized Entropy Difference). *The promise problem* $\mathrm{ED}_{\mathcal{C}}^{u\textsc{Ent},\ell\textsc{Ent},\textsc{gap}}$, *is defined by the entropy measures* uEnt *and* ℓEnt *from the set* $\{\textsc{Min},\textsc{Renyi},\textsc{Shannon},\textsc{Max}\}$, *an entropy* GAP, *which can be* $+c$, $\times c$ *or* $\exp(c)$ *for some constant* $c > 0$, *refering to additive, multiplicative or exponential gaps in the problem's promise, and a concrete computational model* $\mathcal{C}$. *For a pair of mappings* $p, q \in \mathcal{C}$, *the random variables* P *and* Q *are (respectively) the evaluation of the mappings* p *and* q *on a uniformly random input. The Yes and No instances are, for additive* GAP $= +c$

$$\mathrm{YES} = \{(p,q) : \mathrm{H}_{\mathrm{u}\textsc{Ent}}(P) \geqslant \mathrm{H}_{\mathrm{l}\textsc{Ent}}(Q) + c\},$$

$$\mathrm{NO} = \{(p,q) : \mathrm{H}_{\mathrm{l}\textsc{Ent}}(P) + c \leqslant \mathrm{H}_{\mathrm{u}\textsc{Ent}}(Q)\};$$

for multiplicative GAP $= \times c$ *(for* $c > 1$*)*

$$\mathrm{YES} = \{(p,q) : \mathrm{H}_{\mathrm{u}\textsc{Ent}}(P) \geqslant \mathrm{H}_{\mathrm{l}\textsc{Ent}}(Q) \cdot c > 0\},$$

$$\mathrm{NO} = \{(p,q) : \mathrm{H}_{\mathrm{l}\textsc{Ent}}(P) \cdot c \leqslant \mathrm{H}_{\mathrm{u}\textsc{Ent}}(Q)\};$$

and for exponential GAP $= \exp(c)$

$$\mathrm{YES} = \{(p,q) : \mathrm{H}_{\mathrm{u}\textsc{Ent}}(P) \geqslant \mathrm{H}_{\mathrm{l}\textsc{Ent}}(Q)^c > 1\},$$

$$\mathrm{NO} = \{(p,q) : \mathrm{H}_{\mathrm{l}\textsc{Ent}}(P)^c \leqslant \mathrm{H}_{\mathrm{u}\textsc{Ent}}(Q)\}.$$

We always require that uEnt *is more stringent than* ℓEnt *in that* $\mathrm{H}_{\mathrm{u}\textsc{Ent}}(X) \leqslant \mathrm{H}_{\mathrm{l}\textsc{Ent}}(X)$ *for all random variables* X. *(This ensures that the* YES *and* NO *instances do not intersect.) If we do not explicitly set the different parameters, then the default entropy type for* uEnt *and* ℓEnt *is* Shannon, *the default gap is an additive* GAP $= +1$, *and the class* $\mathcal{C}$ *is* Circ.

Note that, by Claim 2.2, with all other parameters being equal - the more relaxed the entropy notion uEnt is, the easier the problem becomes. Similarly, the more stringent the entropy notion ℓEnt is, the easier the problem becomes.

Polynomial Entropy Difference (PED).

The main problem we focus on in this work is entropy difference for low-degree polynomial mappings.

Definition 3.2(Polynomial Entropy Difference). *The promise problem* $\mathrm{PED}_{\mathbb{F},d}^{u\textsc{Ent},\ell\textsc{Ent},\textsc{gap}}$ *is the entropy difference problem for degree* d *polynomials over* $\mathbb{F}$, *i.e. it is the promise problem* $\mathrm{ED}_{\textsc{polynomials}_{\mathbb{F},d}}^{u\textsc{Ent},\ell\textsc{Ent},\textsc{gap}}$ *(see Definition 3.1). The default values for* uEnt, ℓEnt, *and* GAP *(if not specified explicitly) are as in Definition 3.1.*

Polynomial Entropy Approximation.

Another natural algorithmic problem is that of *approximating* the entropy of a polynomial mapping up to a constant, multiplicative, or exponential approximation factor. We discuss this problem informally, focusing on its connection to Polynomial Entropy Difference.

The Polynomial Entropy Approximation problem is, given a polynomial mapping p of low degree, which induces a random variable P, to output an approximation k to its entropy. For the approximation problem $\mathrm{PEA}_{\mathbb{F},d}^{u\textsc{Ent},\ell\textsc{Ent},+c}$, we require that (w.h.p) the approximation k satisfy:

$$\mathrm{H}_{\mathrm{u}\textsc{Ent}}(P) - c < k < \mathrm{H}_{\mathrm{l}\textsc{Ent}}(P) + c.$$

Using binary search, this approximation problem can be shown to be equivalent to deciding the following promise problem:

$$\mathrm{YES} = \{(p,k) : \mathrm{H}_{\mathrm{u}\textsc{Ent}}(P) \geqslant k + c\},$$

$$\mathrm{NO} = \{(p,k) : \mathrm{H}_{\mathrm{l}\textsc{Ent}}(P) \leqslant k - c\}.$$

For notational convenience, we will also denote this promise problem by $\mathrm{PEA}_{\mathbb{F},d}^{u\textsc{Ent},\ell\textsc{Ent},+c}$. PEA is defined analogously for multiplicative ($\times c$) and exponential ($\exp(c)$) approximation.

We note that (as is the case for Entropy Approximation and Entropy Difference in the statistical zero-knowledge literature [20]), for fixed notions of entropy in the upper bounds and lower bounds, the PED and PEA problems are computationally equivalent up to some loss in the approximation factor, for both additive and multiplicative approximation.

In one direction, $\mathrm{PED}_{\mathbb{F},d}^{u\textsc{Ent},\ell\textsc{Ent},+c}$ reduces to $\mathrm{PEA}_{\mathbb{F},d}^{u\textsc{Ent},\ell\textsc{Ent},+c/2}$. To see this, approximate the entropy of the two distributions P and Q, get answers k_p and k_q (respectively), and accept if $k_p > k_q$. Otherwise reject. For a YES instance, $\mathrm{H}_{u\textsc{Ent}}(P) \geqslant \mathrm{H}_{\ell\textsc{Ent}}(Q) + c$, and so if the PEA approximation error is less than $c/2$ we get that k_p must be greater than $\mathrm{H}_{u\textsc{Ent}}(P) - c/2$ and k_q must be less than $\mathrm{H}_{\ell\textsc{Ent}}(Q) +$

$c/2$, and so (w.h.p) $k_p > k_q$ and we accept. For NO instances, the reverse holds and w.h.p we reject.

In the other direction, we get that $\mathrm{PEA}_{\mathbb{F},d}^{u\textsc{Ent},\ell\textsc{Ent},+c}$ reduces to $\mathrm{PED}_{\mathbb{F},d}^{u\textsc{Ent},\ell\textsc{Ent},+c}$. If the given parameter k is an integer multiple of $\log|\mathbb{F}|$, then we can just construct q so that Q is a flat distribution with a support of size 2^k (e.g. q is the identity map on $\mathbb{F}^n$ for $n = k/\log|\mathbb{F}|$), and then the answer to the PED instance (p, q) is equal to the answer to the PEA instance (p, k). In case k is not an integer, then we instead apply the above reduction to the instance $(p^t, \lceil tk\rceil)$ for a large enough integer t, where $p^t(x_1, \ldots, x_t) = p(x_1), p(x_2), \ldots, p(x_t)$. For a YES instance (p, k) of PEA, we have

$$\mathrm{H}_{u\textsc{Ent}}(P^t) = t \cdot \mathrm{H}_{u\textsc{Ent}}(P) \geqslant t \cdot (k + c) \geqslant \lceil tk\rceil + c$$

for $t \geqslant 1 + 1/c$, so $(p^t, \lfloor tk\rfloor)$ is also a YES instance of PEA. NO instances can be analyzed similarly.

For multiplicative approximation, we can reduce $\mathrm{PEA}_{\mathbb{F},d}^{u\textsc{Ent},\ell\textsc{Ent},\times c}$ to $\mathrm{PED}_{\mathbb{F},d}^{u\textsc{Ent},\ell\textsc{Ent},\times c'}$ for any constant $c' < c$. For a YES instance (p, k) of $\mathrm{PEA}_{\mathbb{F},d}^{u\textsc{Ent},\ell\textsc{Ent},\times c}$, we have

$$\mathrm{H}_{u\textsc{Ent}}(P^t) = t \cdot \mathrm{H}_{u\textsc{Ent}}(P) \geqslant t \cdot kc \geqslant c' \cdot \lceil tk\rceil,$$

provided $t \geqslant c'/((c - c')k)$. We may assume that k is bounded below by a constant, because the Schwartz–Zippel Lemma (cf. Lemma 5.10), implies that a nonconstant polynomial mapping of degree d must have min-entropy at least $\log((1/(1 - |\mathbb{F}|^{-d})))$, which is constant for fixed $\mathbb{F}$ and d. So for a sufficiently large constant t, $(p^t, \lceil tk\rceil)$ is a YES instance of $\mathrm{PEA}_{\mathbb{F},d}^{u\textsc{Ent},\ell\textsc{Ent},\times c'}$. NO instances can be analyzed similarly, and thus we can apply the above reduction for integer thresholds.

4 Hardness of Polynomial Entropy Difference

In this section we present evidence that even when we restrict PED to low degree polynomial mappings, and even when we work with relaxed notions of entropy, the problem remains hard. This is done first by using the machinery of randomizing polynomials [6,23] to reduce ED for rich complexity classes (such as log space) to PED (section 4.1). We then argue the hardness of ED for log-space computations, first via the problem's completeness for a rich complexity class (a large subclass of **SZKP**), and then via reductions from specific well-studied hard problems.

4.1 Randomized encodings

We recall the notion of randomized encodings that was developed by Applebaum, Ishai, and Kushilevitz [6,23]. Informally, a randomized encoding of a function f is a randomized function g such that the (randomized) output $g(x)$ determines $f(x)$, but reveals no other information about x. We need the perfect variant of this notion, which we now formally define. (We comment that [6,23] use different, more cryptographic, terminology to describe some of the properties below).

Defintion 4.1. [6,23] *Let $f : \{0,1\}^n \to \{0,1\}^\ell$ be a function. We say that the function $\hat{f} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^s$ is a* perfect randomized encoding *of f with blowup b if it is:*

- **Input independent:** *for every $x, x' \in \{0,1\}^n$ such that $f(x) = f(x')$, the random variables $\hat{f}(x, U_m)$ and $\hat{f}(x', U_m)$ are identically distributed.*
- **Output disjoint:** *for every $x, x' \in \{0,1\}^n$ such that $f(x) \neq f(x')$, $\mathrm{Supp}(\hat{f}(x, U_m)) \cap \mathrm{Supp}(\hat{f}(x', U_m)) = \emptyset$.*
- **Uniform:** *for every $x \in \{0,1\}^n$ the random variable $\hat{f}(x, U_m)$ is uniform over $\mathrm{Supp}(\hat{f}(x, U_m))$.*
- **Balanced:** *for every $x, x' \in \{0,1\}^n$ $|\mathrm{Supp}(\hat{f}(x, U_m))| = |\mathrm{Supp}(\hat{f}(x', U_m))| = b$.*

We now set up notations and state some simple claims about randomized encodings.

Let

$$f : \{0,1\}^n \to \{0,1\}^\ell$$

be a function and let $\hat{f} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^s$ be a perfect randomized encoding of f with blowup b. For $y \in \mathrm{Supp}(f)$, define the set $S_y \subseteq \{0,1\}^s$ to be:

$$\{z \in \{0,1\}^s : \exists(x, r) \in \{0,1\}^n \times \{0,1\}^m$$

$$\text{s.t. } f(x) = y \wedge \hat{f}(x, r) = z\}$$

By the properties of perfect randomized encodings, the sets S_y form a balanced partition of $\mathrm{Supp}(\hat{f})$, indeed $S_y = \mathrm{Supp}(\hat{f}(x, U_m))$ for every x such that $f(x) = y$, and hence $|S_y| = b$. With this notation, the following claim is immediate.

Claim 4.2. $\mathrm{Supp}(\hat{f}) = b \cdot \mathrm{Supp}(f)$

For every $z \in \mathrm{Supp}(\hat{f})$, we denote by y_z the unique string in $\mathrm{Supp}(f)$ such that $z \in S_y$. For any $x \in$

$\{0,1\}^n$, $\hat{f}(x, U_m)$ is uniformly distributed over $S_{f(x)}$. It follows that,

Claim 4.3. *For every* $z \in \mathrm{Supp}(\hat{f})$,

$$\Pr[\hat{f}(U_n, U_m) = z] = \frac{1}{b} \Pr[f(U_n) = y_z]$$

We now state the relation between the entropy of $\hat{f}(U_n, U_m)$ and the entropy of $f(U_n)$ for each one of the entropy measures.

Claim 4.4. *Let*

$$\textsc{Ent} \in \{\textsc{Min}, \textsc{Renyi}, \textsc{Shannon}, \textsc{Max}\}$$

then $\mathrm{H}_{\textsc{Ent}}(\hat{f}(U_n, U_m)) = \mathrm{H}_{\textsc{Ent}}(f(U_n)) + \log b$

Proof. For $\textsc{Ent} = \textsc{Max}$, the claim follows directly from Claim 4.2. For $\textsc{Ent} = \textsc{Min}$, the claim follows directly from Claim 4.3. For $\textsc{Ent} = \textsc{Shannon}$,

$$\begin{aligned}
&\mathrm{H}_{\mathrm{Shannon}}(\hat{f}(U_n, U_m)) \\
&= \mathrm{H}_{\mathrm{Shannon}}(\hat{f}(U_n, U_m), f(U_n)) \\
&= \mathrm{H}_{\mathrm{Shannon}}(f(U_n)) + \\
&\quad \mathrm{H}_{\mathrm{Shannon}}(\hat{f}(U_n, U_m)|f(U_n)) \\
&= \mathrm{H}_{\mathrm{Shannon}}(f(U_n)) + \mathrm{H}_{\mathrm{Shannon}}(\hat{f}(U_n, U_m)|U_n) \\
&= \mathrm{H}_{\mathrm{Shannon}}(f(U_n)) + \log b.
\end{aligned}$$

The first equality follows from the fact that $\hat{f}(x, r)$ determines $f(x)$ (follows from output disjointness). The second equality uses the chain rule for conditional entropy. The third equality follows from input independence, and the last equality follows from the fact that the perfect randomized encoding is uniform, balanced and has blowup b.

By similar reasoning, for $\textsc{Ent} = \textsc{Renyi}$, we have

$$\begin{aligned}
&\mathrm{cp}(\hat{f}(U_n, U_m)) \\
&= \Pr[\hat{f}(U_n, U_m) = \hat{f}(U'_n, U'_m)] \\
&= \Pr[f(U_n) = f(U'_n)] \cdot \\
&\quad \Pr[\hat{f}(U_n, U_m) = \hat{f}(U'_n, U'_m)|f(U_n) = f(U'_n)] \\
&= \mathrm{cp}(f(U_n)) \cdot (1/b).
\end{aligned}$$

□

4.2 From branching-program entropy difference to polynomial entropy difference

Applebaum, Ishai and Kushilevitz [6,23] showed that logspace mappings (represented by the branching programs that compute the output bits) have randomized encodings which are polynomial mappings of degree three over the field with two elements.

Theorem 4.5. *[6,23] Given a branching program $f : \{0,1\}^n \to \{0,1\}^\ell$, we can construct in polynomial time a degree 3 polynomial $\hat{f} : \mathbb{F}_2^n \times \mathbb{F}_2^m \to \mathbb{F}_2^s$ that is a perfect randomized encoding of f. Moreover, the blowup b is a power of 2 and can be computed in polynomial time from f.*

Based on this theorem we show that the log-space entropy difference problem (for the various notions of entropy which we defined above) with additive gap reduces to the polynomial entropy difference problem with the same gap.

Theorem 4.6. *The promise problem* $\mathrm{ED}_{\mathrm{BP}}^{u\textsc{Ent},\ell\textsc{Ent},+c}$, *for* $u\textsc{Ent}, \ell\textsc{Ent} \in \{\textsc{Min}, \textsc{Renyi}, \textsc{Shannon}, \textsc{Max}\}$, *Karp-reduces to the promise problem* $\mathrm{PED}_{\mathbb{F}_2,3}^{u\textsc{Ent},\ell\textsc{Ent},+c}$.

Proof. Given an instance (X, Y) of $\mathrm{ED}_{\mathrm{BP}}^{u\textsc{Ent},\ell\textsc{Ent},+c}$, apply on each one of the branching programs X and Y the reduction from Theorem 4.5, to obtain a pair of polynomials $\hat{X}$ and $\hat{Y}$ of degree 3 over $\mathbb{F}_2$. By padding the output of $\hat{X}$ or $\hat{Y}$ with independent uniformly distributed bits, we can ensure that $\hat{X}$ and $\hat{Y}$ have the same blow-up. By Claim 4.4, $\mathrm{H}_{\mathrm{uEnt}}(\hat{X}) - \mathrm{H}_{\mathrm{lEnt}}(\hat{Y}) = \mathrm{H}_{\mathrm{uEnt}}(X) - \mathrm{H}_{\mathrm{lEnt}}(Y)$, and $\mathrm{H}_{\mathrm{lEnt}}(\hat{X}) - \mathrm{H}_{\mathrm{uEnt}}(\hat{Y}) = \mathrm{H}_{\mathrm{lEnt}}(X) - \mathrm{H}_{\mathrm{uEnt}}(Y)$. It follows that yes (resp. no) instances of $\mathrm{ED}_{\mathrm{BP}}^{u\textsc{Ent},\ell\textsc{Ent},+c}$ are mapped to yes (resp. no) instances of $\mathrm{PED}_{\mathbb{F}_2,3}^{u\textsc{Ent},\ell\textsc{Ent},+c}$. □

4.3 Polynomial entropy difference and statistical zero-knowledge

Goldreich and Vadhan [21] showed that the promise problem ED (Entropy Difference problem for Shannon entropy with additive gap and polynomial-size circuits) is complete for **SZKP**, the class of problems having statistical zero-knowledge proofs. i We show a computationally restricted variant of this result, showing that $\mathrm{PED}_{\mathbb{F}_2,3}$ is complete for $\mathbf{SZKP_L}$, the class of problems having statistical zero-knowledge proofs in which the honest verifier and its simulator are computable in logarithmic space (with two-way access to the input, coin tosses, and transcript).

Theorem 4.7. *The promise problem* $\mathrm{PED}_{\mathbb{F}_2,3}$ *is complete for the class* $\mathbf{SZKP_L}$.

We start with proving that the problem is hard for the class.

Lemma 4.8. *The promise problem* $\mathrm{PED}_{\mathbb{F}_2,3}$ *is hard for the class* $\mathbf{SZKP_L}$ *under Karp-reductions.*

Proof. We show that the promise problem $\mathrm{ED}_{\mathrm{BP}}$, is hard (under Karp-reductions) for the class $\mathbf{SZKP_L}$. The proof then follows by Theorem 4.6. The hardness of $\mathrm{ED}_{\mathrm{BP}}$ follows directly from the reduction of [21] which we now recall. Given a promise problem in **SZKP** with a proof system (P, V) and a simulator S, it is assumed w.l.o.g. that on instances of length n, V tosses exactly $\ell = \ell(n)$ coins, the interaction between P and V consists of exactly $2r = 2r(n)$ messages each of length exactly ℓ, the prover sends the odd messages and the last message of the verifier consists of its random coins. Furthermore, the simulator for this protocol always outputs transcripts that are consistent with V's coins. For problems in $\mathbf{SZKP_L}$, using the fact that the verifier is computable in logspace (with two-way access to the input, its coin tosses, and the transcript), we can obtain such a simulator that is computable in logspace (again with two-way access to the input and its coin tosses). On input x, we denote by $S(x)_i$ $(1 \leqslant i \leqslant 2r)$ the distribution over the $(i \cdot \ell)$-long prefix of the output of the simulator. That is, the distribution over the simulation of the first i messages in the interaction between P and V.

The reduction maps an instance x to a pair of distributions (X_x, Y_x):

- X_x outputs independent samples from the distributions $S(x)_2, S(x)_4, \ldots, S(x)_{2r}$.
- Y_x outputs independent samples from the distributions $S(x)_1, S(x)_3, \ldots, S(x)_{2r-1}$ and $U_{\ell-2}$.

Since S is computable in logarithmic space, we can efficiently construct branching programs X_x and Y_x that sample from the above distributions. □

To complete the proof of Theorem 4.7 we show that $\mathrm{PED}_{\mathbb{F}_2,3}$ is in the class $\mathbf{SZKP_L}$. This follows easily from the proof that ED is in **SZKP** [21]. We give here a sketch of the proof.

Lemma 4.9. $\mathrm{PED}_{\mathbb{F}_2,3}$ *has a statistical zero-knowledge proof system where the verifier and the simulator are computable in logarithmic space.*

sketch. We use the same proof system and simulator from [21]. We need to show that on instance (X, Y) where X and Y are $\textsc{polynomials}_{\mathbb{F}_2,3}$-mappings, the verifier and the simulator are computable in logarithmic space. For simplicity we assume that both X and Y map n input bits to m output bits. We start with the complexity of the verifier. The protocol is public coins, so we only need to check that the verifier's final decision can be computed in logspace. This boils down to two operations which the verifier performs a polynomial number of times: (a) evaluating the $\textsc{polynomials}_{\mathbb{F}_2,3}$-mapping of either X or Y on an input specified in the transcript, and (b) evaluating a function $h : \{0,1\}^{n+m} \to \{0,1\}^k$, from a family of 2-universal hash functions, where both the description of h and the input on which to evaluate h are specified in the transcript. (See [21] for the details.) The former can be done in logspace as it involves evaluating polynomials of degree 3 over $\mathbb{F}_2$. The latter can be done in logspace if we use standard 2-universal families of hash functions, such as affine-linear maps from $\mathbb{F}_2^{n+m}$ to $\mathbb{F}_2^k$.

Turning to the simulator, we see that its output consists of many copies of triplets taking the following form: (h, r, x) where $r \in_R \{0,1\}^n$, x is an output of either X or Y on a uniformly chosen input which is part of the simulator's randomness, and $h : \{0,1\}^{n+m} \to \{0,1\}^k$ is a function uniformly chosen from the family of 2-universal hash functions subject to the constraint that $h(r, x) = 0$. As in the verifier's case, x can be computed by a logspace mapping since X and Y are $\textsc{polynomials}_{\mathbb{F}_2,3}$-mappings. Choosing h from the family of hash functions under a constraint $h(z) = 0$ can be done efficiently if we use affine-linear hash functions $h(z) = Az + b$. We simply choose the matrix A uniformly at random and set $b = -Az$. □

4.3.1 Additional remarks

We remark, without including proofs, that similar statements as in the one from Theorem 4.7 can be shown for other known complete problems in **SZKP** and its variants [8,10,12,20,21,28,35,36]). We also mention that all the known closure and equivalence properties of **SZKP** (e.g. closure under complement [31], equivalence between honest and dishonest verifiers [19], and equivalence between public and private coins [31]) also hold for the class $\mathbf{SZKP_L}$.

Finally, we mention that by using the locality reduction of [6] we can further reduce $\mathrm{PED}_{\mathbb{F}_2,3}$ to $\mathrm{ED}_{\mathbf{NC}_4^0}$, where $\mathbf{NC}_4^0$ is the class of functions for which every output bit depends on at most four input bits.

4.4 Hardness results

Given the results of Section 4.3, we can conclude that Polynomial Entropy Difference (with additive Shannon entropy gap) is at least as hard as problems with statistical zero-knowledge proofs with logarithmic space verifiers and simulators. This includes problems such as Graph Isomorphism, Quadratic Residuosity, Decisional Diffie Hellman, and the approximate Closest Vector Problem, and also many other cryptographic problems.

For the reduction from Graph Isomorphism, we note that the operations run by the verifier and the simulator in the statistical zero-knowledge proof of [18], the most complex of which is permuting a graph, can all be done in logarithmic space. Similarly, for the approximate Closest Vector Problem, the computationally intensive operations run by the simulator in the zero-knowledge proof of [16] (and the alternate versions in [29]) are sampling from a high-dimensional Gaussian distribution and reducing modulo the fundamental parallelepiped. These can be done in logarithmic space. (To reduce modulo the fundamental parallelepiped we need to change the noise vector from the standard basis to the given lattice basis and back. By pre-computing the change-of-basis matrices, the sampling algorithm only needs to compute matrix-vector products, which can be done in logarithmic space.)

For the Quadratic Residuosity and Decisional Diffie Hellman problems, we show that in fact they reduce to an easier variant of PED, where the yes-instances have high min-entropy and the no-instances have small support size. See [24] for more background on these assumptions and the number theory that comes into play.

4.4.1 Quadratic residuosity

Definition 4.10 (Quadratic Residuosity). *For a composite $N = p \cdot q$ where p and q are prime and different, the promise problem* Quadratic Residuosity *is defined as follows:*

$$\begin{aligned} \mathrm{QR}_{\mathrm{YES}} &= \{(N,x) : N = p \cdot q, \\ &\qquad \exists y \in \mathbb{Z}_N^* s.t. x = y^2 \pmod{N}\} \\ \mathrm{QR}_{\mathrm{NO}} &= \{(N,x) : N = p \cdot q, \\ &\qquad \nexists y \in \mathbb{Z}_N^* s.t. x = y^2 \pmod{N}\} \end{aligned}$$

Claim 4.11. Quadratic Residuosity reduces to $\mathrm{PED}_{\mathbb{F}_2,3}^{\textsc{Min},\textsc{Max},+1}$.

Proof. Given an input (N, x), where $N = p \cdot q$ for primes p, q, we examine the mapping $f_{N,x}$. This mapping gets as input a random $c \in \{0, 1\}$ and coins for generating $r \sim_R [N]$ and outputs[3]

$$x^c \cdot r^2 \pmod{N}$$

We examine the distribution of $f_{N,x}$'s output. We first examine the distribution or mapping R that just outputs r^2. By the Chinese Remainder Theorem, we have an isomorphism $\mathbb{Z}_N \cong \mathbb{Z}_p \times \mathbb{Z}_q$ and under this isomorphism, R decomposes into a product distribution $R_p \cdot R_q$, where R_p is over $\mathbb{Z}_p$ and R_q is over $\mathbb{Z}_q$, where each item in $\mathbb{Z}_N$ is equivalent to a pair in $\mathbb{Z}_p \times \mathbb{Z}_q$ via the Chinese Remainder Theorem. Examining the two distributions, we see that R_p gives probability $1/p$ to 0 and $2/p$ to each of the quadratic residues in $\mathbb{Z}_p^*$. Similarly, R_q gives probability $1/q$ to 0 and $2/q$ to each quadratic residue in $\mathbb{Z}_q^*$. So the support of R is of size $(p + 1) \cdot (q + 1)/4$, and each item in the support gets probability at most $4/pq$.

Now examining the output of $f_{N,x}$, if x is a quadratic residue in $\mathbb{Z}_N^*$, then it is a residue in $\mathbb{Z}_p^*$ and in $\mathbb{Z}_q^*$, and so the distribution of $x^c \cdot r^2$ is equal to the distribution of r^2, so its support and min-entropy are as above.

On the other hand, if x is a non-residue in $\mathbb{Z}_N^*$ then it must be a non-residue in $\mathbb{Z}_p^*$ or in $\mathbb{Z}_q^*$, say $\mathbb{Z}_p^*$. This implies that $x^c \cdot r^2 \bmod p$ is uniformly distributed in $\mathbb{Z}_p$ and thus has min-entropy $\log p$. Conditioned on c and $r \bmod p$, the value $x^c \cdot r^2 \bmod q$ still has min-entropy at least that of $r^2 \bmod q$, which is $\log q - 1$ as argued above. By the Chinese Remainder Theorem, $x^c \cdot r^2 \bmod N$ has min-entropy at least $\log p + \log q - 1$.

We can now use $f_{N,x}$ to build two mappings or distributions X and Y, s.t. if x is a YES instance of Quadratic Residuosity, then the min-entopy of X is higher by a small constant (say 1/2) than the log-support size of Y, and vice-versa if x is a NO instance. This allows us to reduce Quadratic Residuosity to $\mathrm{ED}^{\textsc{Min},\textsc{Max},+1}$.

Finally, the mappings only sample in $\mathbb{Z}_N$ and compute integer multiplication and division, so they can be computed in logarithmic space [11] and hence by polynomial-sized branching programs. Using Theo-

[3] We note that a more natural map to consider (which is easier to analyze) samples $r \sim_R Z_N^*$. We are unaware of a method for uniform sampling in Z_N^*, given only N, in logarithmic space. Also note, that we assume that we can sample uniformly from $[N]$. This is not accurate (since N is not a power of 2) and we address this issue in Remark 4.12 following the proof.

rem 4.6, we conclude that Quadratic Residuosity reduces to $\mathrm{PED}_{\mathbb{F}_2,3}^{\textsc{Min},\textsc{Max},+1}$. □

Remark 4.12. *In the proof above we assume that we can sample uniformly from $[N]$ given uniformly random bits. This is not accurate since N is not a power of two. To fix this we slightly modify the mapping $f_{N,x}$ as follows. Let $k = \lceil \log N \rceil$. The mapping receives as input $c \in \{0,1\}$ as well as $2k$ strings $r_1, \ldots, r_{2k}$ each in $\{0,1\}^k$. It outputs $x^c \cdot r_i^2 \pmod N$, where $i \in [2k]$ is the minimal index such that $r_i \in [N]$ (when viewed as an integer in binary representation). If no such i exist then the mapping outputs 0. First observe that this new mapping can still be computed in logarithmic space. The probability that for no $i \in [2k]$, $r_i \in [N]$ is at most $1/N^2$. Hence in the analysis above probabilities change by at most $1/N^2$. The support of the modified mapping remains the same as the original one. It follows (by the proof above) that there is a constant gap between the max and min entropies of yes and no instances. This gap can be amplified to be more than 1 by taking two independent copies of the mapping.*

4.4.2 Decisional Diffie-Hellman

Definition 4.13 (Decisional Diffie Hellman). *The promise problem* Decisional Diffie Hellman *is defined with respect to a family $\mathcal{G}$ of cyclic groups of prime order. It is defined as follows:*

$$\begin{aligned}\mathrm{DDH}_{\mathrm{YES}} = \{(G, g, g^a, g^b, g^{ab} :\ & \\ & G \in \mathcal{G} \text{ of prime order } q, \\ & g \text{ generator of G}, a, b \in \mathbb{Z}_q\} \\ \mathrm{DDH}_{\mathrm{NO}} = \{(G, g, g^a, g^b, g^c :\ & \\ & G \in \mathcal{G} \text{ of prime order } q, \\ & g \text{ generator of G}, a, b, c \in \mathbb{Z}_q, c \neq a \cdot b\}\end{aligned}$$

Claim 4.14. Decisional Diffie Hellman *reduces to the problem* $\mathrm{PED}_{\mathbb{F}_2,3}^{\textsc{Min},\textsc{Max},+1}$.

Proof Sketch. We use the random self-reducibility of DDH, due to Naor and Reingold [30]. They showed how to transform a given DDH instance $x = (G, g, g^a, g^b, g^c)$ into a new one $(G, g, g^{a'}, g^{b'}, g^{c'})$, such that a', b' are uniformly random in Z_q and: (i) if x is a YES instance (i.e. $c = a \cdot b$) then $c' = a' \cdot b'$, so the output (in its entirety) is uniform over a set of size $|G|^2$. On the other hand, (ii) if x is a NO instance then c' (and also $g^{c'}$) is uniformly random given $(G, g, g^{a'}, g^{b'})$ and the output (in its entirety) is uniform over a set of size $|G|^3$.[4]

The mapping computed by this reduction allows us to transform a Decisional Diffie Hellman instance in an instance of ED, where yes instances are transformed into pair of mappings or distributions (X, Y), where X is uniform over a set of size $|G|^{2.5}$ (some fixed dummy distribution) and on YES instances Y is uniform over a set of size $|G|^2$ and on NO instances Y is uniform over a set of size $|G|^3$. I.e., it reduces Decisional Diffie Hellman to $\mathrm{ED}^{\textsc{Min},\textsc{Max},\times 1.2}$.

Finally, to reduce Decisional Diffie Hellman to PED we need to activate the randomizing polynomial machinery of Theorem 4.6. The maps X and Y as described above compute multiplication (which can be done in log-space) and exponentiation, which is not a log-space operation. However, the elements being exponentiated are all known in advance. We can thus use an idea due to Kearns and Valiant [25] and compute in advance for each of these basis, say g, the powers $(g, g^2, g^4, g^8, \ldots)$. Each exponentiation can then be replaced by an iterated product. By the results of Beame, Cook and Hoover [7] the iterated product can be computed in logarithmic depth (or space). By Theorem 4.6, we conclude that Decisional Diffie Hellman reduces to $\mathrm{PED}_{\mathbb{F}_2,3}^{\textsc{Min},\textsc{Max},+1}$. □

5 Algorithms for polynomial entropy approximation

5.1 Approximating entropy via directional derivatives

In this section we give an approximation algorithm for the entropy of homogenous polynomial maps of degree two, over prime fields $\mathbb{F}_q$ other than $\mathbb{F}_2$. The general strategy is to relate the entropy of a quadratic map with the entropy of a random *directional derivative* of the map. These derivatives are of degree one and so their entropy is easily computable.

For a polynomial mapping $P : \mathbb{F}_q^n \to \mathbb{F}_q^m$ and a vector $a \in \mathbb{F}_q^n$ we define the *directional derivative* of P in direction a as the mapping $D_a P : \mathbb{F}_q^n \to \mathbb{F}_q^m$ given by

$$D_a P(x) \overset{\mathrm{def}}{=} P(x + a) - P(x).$$

It is easy to verify that for every fixed a, $D_a P(x)$ is a polynomial mapping of degree at most $\deg(P) - 1$.

Throughout this section, q is a prime other than 2 and $Q : \mathbb{F}_q^n \mapsto \mathbb{F}_q^m$ denotes a homogenous

[4] To deal with the problem of sampling uniformly from sets whose size is not a power of two refer to Remark 4.12.

quadratic mapping given by m quadratic polynomials $Q_1(x), \ldots, Q_m(x)$ in n variables $x = (x_1, \ldots, x_n)$. For every $i \in [m]$ there exists an $n \times n$ matrix M_i such that $Q_i(x) = x^t \cdot M_i \cdot x$. If $\mathrm{char}(\mathbb{F}_q) \neq 2$ then we can assume w.l.o.g that M_i is always symmetric (by replacing M_i with $(M_i + M_i^t)/2$ if needed).

For every fixing of a, $D_aQ(x)$ is an affine (degree at most one) mapping. We denote by $r(a)$ the rank of $D_aQ(x, a)$ (that is, the dimension of the affine subspace that is the image of the mapping given by $D_aQ(x)$). We relate the $r(a)$'s to entropy in the following two lemmas:

Lemma 5.1. *For every $a \in \mathbb{F}_q^n$ we have*

$$r(a) \leqslant 2 \cdot \mathrm{H}_{\mathrm{Shannon}}(Q(U_n))/\log q.$$

Lemma 5.2.

$$\mathop{\mathrm{E}}_{a \overset{\mathrm{R}}{\leftarrow} \mathbb{F}_q^n} \left[q^{-r(a)}\right] = 2^{-\mathrm{H}_{\mathrm{Renyi}}(Q(U_n))}.$$

Before proving these lemmas, we use them to obtain our algorithm:

Theorem 5.3. *There exists a probabilistic polynomial-time algorithm A that, when given a prime $q \neq 2$, a homogeneous quadratic map $Q : \mathbb{F}_q^n \mapsto \mathbb{F}_q^m$ (as a list of coefficients), and an integer $0 < k \leqslant m$ and outputs TRUE or FALSE such that:*

- *If $\mathrm{H}_{\mathrm{Renyi}}(Q(U_n)) \geqslant 2k \cdot \log(q) + 1$ then A outputs TRUE with probability at least $1/2$.*
- *If $\mathrm{H}_{\mathrm{Shannon}}(Q(U_n) < k \cdot \log(q)$ then A always outputs FALSE.*

Proof. The algorithm simply computes the rank $r(a)$ of the directional derivative D_aQ in a random direction $a \in \mathbb{F}_q^n$. If the value of $r(a)$ is at least $2k$ the algorithm returns TRUE, otherwise the algorithm returns FALSE. If $\mathrm{H}_{\mathrm{Shannon}}(Q(U_n)) < k \cdot \log(q)$ then, from Lemma 5.1 we have that $r(a)$ will always be smaller than $2k$ and so the algorithm will work with probability one. If $\mathrm{H}_{\mathrm{Renyi}}(Q(U_m)) \geqslant 2k \cdot \log(q) + 1$ then, using Lemma 5.2 and Markov's inequality, we get that $q^{-r(a)}$ will be at most $2^{-2k \log q}$ with probability at least $1/2$. Therefore, the algorithm works as promised. □

We now prove the two main lemmas.

Proof of Lemma 5.1 Since the output of an affine mapping is uniform on its output, we have

$$\mathrm{H}_{\mathrm{Shannon}}(D_aQ(U_n)) = \log(q^{r(a)}).$$

By subadditivity of Shannon entropy, we have

$$\begin{aligned} &\mathrm{H}_{\mathrm{Shannon}}(D_aQ(U_n)) \\ &\quad \leqslant \mathrm{H}_{\mathrm{Shannon}}(Q(U_n + a)) + \mathrm{H}_{\mathrm{Shannon}}(Q(U_n)) \\ &\quad = 2\,\mathrm{H}_{\mathrm{Shannon}}(Q(U_n)) \end{aligned}$$

□

The proof of Lemma 5.2 works by expressing both sides in terms of the Fourier coefficients of the distribution $Q(U_n)$, which are simply given by the following biases:

Definition 5.4. *For a prime q and a random variable X taking values in $\mathbb{F}_q$, we define*

$$\mathrm{bias}(X) \overset{\mathrm{def}}{=} \left|\mathrm{E}\left[\omega_q^X\right]\right|,$$

where $\omega_q = e^{2\pi i/q}$ is the complex primitive q'th root of unity. For a random variable Y taking values in $\mathbb{F}_q^m$ and a vector $u \in \mathbb{F}_q^m$, we define we define

$$\mathrm{bias}_u(Y) \overset{\mathrm{def}}{=} \mathrm{bias}(\langle u, Y\rangle) = \left|\mathrm{E}\left[\omega_q^{\langle u, X\rangle}\right]\right|,$$

where $\langle \cdot, \cdot \rangle$ is inner product modulo q.

Note that if Y is uniform on $\mathbb{F}_q^m$, then $\mathrm{bias}_u(Y) = 0$ for all $u \neq 0$. A relation between bias and rank in the case of a single output (i.e. $m = 1$) is given by the following:

Claim 5.5. *Suppose* $\mathrm{char}(\mathbb{F}_q) \neq 2$. *Let $R(x_1, \ldots, x_n) = x^tMx$ be a homogeneous quadratic polynomial over $\mathbb{F}_q^n$ such that* $\mathrm{rank}(M) = k$ *and M is symmetric. Then,*

$$\mathrm{bias}(R(U_n)) = q^{-k/2}.$$

Proof. As shown in [26], $R(x)$ is equivalent (under a linear change of variables) to a quadratic polynomial $S(x) = \sum_{i=1}^k a_i \cdot x_i^2$ where $a_1, \ldots, a_k \in \mathbb{F}_q^*$. Then

$$\begin{aligned} \mathrm{bias}(R(U_n)) &= \mathrm{bias}(S(U_n)) \\ &= \left| \frac{1}{q^m} \sum_{x \in \mathbb{F}_q^m} \omega_q^{\sum_{i\in[k]} a_i \cdot x_i^2} \right| \\ &= \prod_{i \in [k]} \left| \frac{1}{q} \sum_{y \in \mathbb{F}_q} \omega_q^{a_i \cdot y^2} \right| \\ &= (q^{-1/2})^k = q^{-k/2}, \end{aligned}$$

where the last equality follows from the Gauss formula for quadratic exponential sums in one variable (see [26]). □

Next we relate biases for many output coordinates to Renyi entropy.

Claim 5.6. *Let X be a random variable taking values in $\mathbb{F}_q^m$. Then*

$$2^{-\mathrm{H}_{\mathrm{Renyi}}(X)} = \mathop{\mathrm{E}}_{u \xleftarrow{\mathrm{R}} \mathbb{F}_q^m} [\mathrm{bias}_u(X)^2].$$

Proof. We begin by recalling that the Renyi entropy simply measures the ℓ_2 distance of a random variable from uniform:

$$\begin{aligned} 2^{-\mathrm{H}_{\mathrm{Renyi}}(X)} &= \mathrm{cp}(X) \\ &= \sum_x \Pr[X=x]^2 \\ &= \sum_x (\Pr[X=x] - 1/q^m)^2 + 1/q^m \\ &= \|X - U_m\|^2 + 1/q^m, \end{aligned}$$

where $\|X - U_m\|$ denotes the ℓ_2 distance between the probability mass functions of X and U_m (viewed as vectors of length q^m). By Parseval, the ℓ_2 distance does not change if we switch to the Fourier basis: For $u \in \mathbb{F}_q^m$, the u'th Fourier basis function $\chi_u : \mathbb{F}_q^m \to \mathbb{C}$ is the function given by

$$\chi_u(x) = \frac{1}{q^{m/2}} \cdot \omega_q^{\langle u,x \rangle}.$$

These form an orthonormal basis for the vector space of functions from $\mathbb{F}_q^m$ to $\mathbb{C}$, under the standard inner product $[f,g] = \sum_{x \in \mathbb{F}_q^m} f(x)\overline{g(x)}$.

Abusing notation, we can view a random variable X taking values in $\mathbb{F}_q^m$ as a function $X : \mathbb{F}_q^m \to \mathbb{C}$ where $X(x) = \Pr[X = x]$. Then the u'th Fourier coefficient of X is given by

$$\begin{aligned} \hat{X}_u &\stackrel{\mathrm{def}}{=} [X, \chi_u] \\ &= \frac{1}{q^{m/2}} \cdot \sum_{x \in \mathbb{F}_q^m} \Pr[X=x] \cdot \omega_q^{-\langle u,x \rangle} \\ &= \frac{1}{q^{m/2}} \cdot \mathrm{E}\left[\omega_q^{-\langle u,X \rangle}\right], \end{aligned}$$

so $|\hat{X}_u| = (1/q^{m/2}) \cdot \mathrm{bias}_u(X)$.

Thus, by Parseval, we have:

$$\begin{aligned} \|X - U_m\|^2 &= \sum_u \left|\hat{X}_u - (\hat{U_m})_u\right|^2 \\ &= \sum_{u \neq 0} \left|\hat{X}_u\right|^2 \\ &= \mathop{\mathrm{E}}_{u \xleftarrow{\mathrm{R}} \mathbb{F}_q^m} [\mathrm{bias}_u(X)^2] - 1/q^m. \end{aligned}$$

Putting this together with the first sequence of equations completes the proof. □

Proof of Lemma 5.2. Taking $X = Q(U_n)$ in Claim 5.6, we have

$$2^{-\mathrm{H}_{\mathrm{Renyi}}(Q(U_n))} = \mathop{\mathrm{E}}_{u \xleftarrow{\mathrm{R}} \mathbb{F}_q^m} [\mathrm{bias}_u(Q(U_n))^2].$$

By Claim 5.5, $\mathrm{bias}_u(Q(U_n))^2 = \mathrm{bias}(\sum_i u_i Q_i(U_n))^2 = q^{-\mathrm{rank}(\sum_i u_i M_i)}$. Note that for a $s \times t$ matrix M, $q^{-\mathrm{rank}(M)} = \Pr_{v \xleftarrow{\mathrm{R}} \mathbb{F}_q^t}[Mv = 0]$. Thus, we have

$$\begin{aligned} 2^{-\mathrm{H}_{\mathrm{Renyi}}(Q(U_n))} &= \mathop{\mathrm{E}}_{u \xleftarrow{\mathrm{R}} \mathbb{F}_q^m} \left[q^{-\mathrm{rank}(\sum_i u_i M_i)}\right] \\ &= \mathop{\mathrm{E}}_{u \xleftarrow{\mathrm{R}} \mathbb{F}_q^m} \left[\mathop{\Pr}_{a \xleftarrow{\mathrm{R}} \mathbb{F}_q^n} \left[\sum_i u_i M_i a = 0\right]\right] \\ &= \mathop{\mathrm{E}}_{a \xleftarrow{\mathrm{R}} \mathbb{F}_q^n} \left[\mathop{\Pr}_{u \xleftarrow{\mathrm{R}} \mathbb{F}_q^m} \left[\sum_i u_i M_i a = 0\right]\right] \\ &= \mathop{\mathrm{E}}_{a \xleftarrow{\mathrm{R}} \mathbb{F}_q^n} \left[q^{-r(a)}\right], \end{aligned}$$

where the last equality is because $\sum_i u_i M_i a = 0$ iff $uM_a = 0$ where M_a is the matrix whose rows are $M_1 a, \ldots, M_m a$ (and so $r(a) = \mathrm{rank}(M_a)$). □

5.2 Approximating max-entropy over $\mathbb{F}_2$ via rank

In this section we deal with degree d polynomials over $\mathbb{F}_2$. Since the field is $\mathbb{F}_2$ we can assume w.l.o.g that the polynomials are multilinear (degree at most 1 in each variable). We show that, for small d, the rank of the set of polynomials (when viewed as vectors of coefficients) is related to the entropy of the polynomial map. The results of this section can be extended to any field but we state them only for $\mathbb{F}_2$ since this is the case we are most interested in (and is not covered by the results of Section 5.1).

The main technical result of this section is the following theorem, which we prove in Section 5.3 below. The theorem relates the entropy of a polynomial mapping (in the weak form of support) with its rank as a set of coefficient vectors.

Theorem 5.7. *Let $P : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ be a multilinear polynomial mapping of degree $\leqslant d$ such that $|\operatorname{Supp}(P)| \leqslant 2^k$, for $k, d \in \mathbb{N}$. Then*

$$\mathrm{rank}\{P_1, \ldots, P_m\} \leqslant \binom{k+2d}{d},$$

where the rank is understood as the dimension of the $\mathbb{F}_2$-span of $P_1, \ldots, P_m$ (equivalently, the rank of the $m \times \binom{n+d}{d}$ matrix over $\mathbb{F}_2$ whose rows are the coefficient-vectors of the polynomials P_i).

Using this theorem we get the following approximation algorithm for max-entropy over characteristic two:

Theorem 5.8 *There exists a constant c and polynomial-time algorithm A such that when A is given as input a degree d polynomial map $P : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ and an integer $0 < k \leqslant n$, we have:*

- *If $\mathrm{H}_{\max}(P(U_n)) > \binom{k+2d}{d}$, then A outputs TRUE.*
- If $\mathrm{H}_{\max}(P(U_n)) \leqslant k$, *then A outputs FALSE.*

Proof. The algorithm computes the rank of the set of polynomials $P_1, \ldots, P_m$. If it is greater than $\binom{k+2d}{d}$ then it returns TRUE, otherwise it returns FALSE. The correctness follows directly from Theorem 5.7 and from the simple fact that rank at most k implies support size at most 2^k. □

5.3 Proof of Theorem 5.7

The idea of the proof is to find an affine-linear subspace $V \subset \mathbb{F}_2^n$ of dimension $\approx k$ such that the restriction of the polynomials $P_1, \ldots, P_m$ to this subspace does not reduce their rank. Since the restricted polynomials are polynomials of degree $\leqslant d$ in $\approx k$ variables we get that their rank is at most $\approx k^d$.

It turns out that it suffices to take V to be a subspace that hits a large fraction of the outputs of P, as given by the image of L in the following lemma:

Lemma 5.9. *Let $P : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ be some function such that $|\operatorname{Supp}(P(U_n))| \leqslant 2^k$ and let $\varepsilon > 0$. Then, there exists an affine-linear mapping $L : \mathbb{F}_2^\ell \mapsto \mathbb{F}_2^n$ with $\ell = \lceil k + \log(1/\varepsilon) \rceil$ such that*

$$\Pr_{x \in \mathbb{F}_2^n}[\exists y \in \mathbb{F}_2^\ell \ , \ P(x) = P(L(y))] > 1 - \varepsilon.$$

Proof. We use the probabilistic method. Let $L : \mathbb{F}_2^\ell \to \mathbb{F}_2^n$ be a uniformly random affine-linear mapping. Fix $z \in \operatorname{Image}(P)$, and let $\mu_z = |P^{-1}(z)|/2^n$. By the pairwise independence of the outputs of L and Chebychev's Inequality, it follows that

$$\Pr_L[\operatorname{Image}(L) \cap P^{-1}(z) = \emptyset] \leqslant \frac{1 - \mu_z}{\mu_z \cdot 2^\ell}.$$

(For each point $y \in \mathbb{F}_2^\ell$, let X_y be the indicator variable for $L(y) \in P^{-1}(z)$. Then the X_y's are pairwise independent, each with expectation μ_z and variance $\mu_z \cdot (1 - \mu_z)$. Thus, by Chebychev's Inequality, $\Pr[\sum_y X_y = 0] \leqslant (2^\ell \cdot \mu_z \cdot (1 - \mu_z))/(2^\ell \cdot \mu_z)^2)$.

Now, let I_z be an indicator random variable for $\operatorname{Image}(L) \cap P^{-1}(z) = \emptyset$. Then,

$$\begin{aligned}
&\mathop{\mathrm{E}}_L\left[\Pr_{x \in \mathbb{F}_2^n}[\neg\exists y \in \mathbb{F}_2^\ell \ , \ P(x)P(L(y))]\right] \\
&= \mathop{\mathrm{E}}_L\left[\Pr_{x \in \mathbb{F}_2^n}[\operatorname{Image}(L) \cap P^{-1}(P(x)) = \emptyset]\right] \\
&= \mathop{\mathrm{E}}_L\left[\sum_{z \in \operatorname{Image}(P)} \mu_z \cdot I_z\right] \\
&\leqslant \sum_{z \in \operatorname{Image}(P)} \mu_z \cdot \frac{1 - \mu_z}{\mu_z \cdot 2^\ell} \\
&= \frac{|\operatorname{Image}(P)| - 1}{2^\ell} \\
&\leqslant \frac{2^k - 1}{2^\ell} \\
&< \varepsilon,
\end{aligned}$$

for $\ell = \lceil k + \log(1/\varepsilon) \rceil$. By averaging, there exists a fixed L such that

$$\Pr_{x \in \mathbb{F}_2^n}[\neg\exists y \in \mathbb{F}_2^\ell \ , \ P(x) = P(L(y))] < \varepsilon,$$

as desired. □

To show that the property of L given in Lemma 5.9 implies that $P \circ L$ has the same rank as P (when ε is sufficiently small), we employ the following (known) version of the Schwartz-Zippel Lemma, which bounds the number of zeros of a multilinear polynomial of degree d that is not identically zero:

Lemma 5.10. *Let $P \in \mathbb{F}_2[x_1, \ldots, x_n]$ be a degree d* multilinear *polynomial that is not identically zero. Then*

$$\Pr[P(x) = 0] \leqslant 1 - 2^{-d}.$$

Proof. The proof is by double induction on $d = 1, 2, ...$ and $n = d, d+1, ...$ If $d = 1$ then the claim is trivial. Suppose we proved the claim for degree $< d$ and all n and for degree d and $< n$ variables.

If $n = d$ (it cannot be smaller than d since the degree is d) then the bound is trivial since there is at least one point at which P is non zero and this point has weight 2^{-d}.

Suppose $n > d$ and assume w.l.o.g that x_1 appears in P. Write P as

$$P(x_1, \dots, x_n) = x_1 \cdot R(x_2, \dots, x_n) + S(x_2, \dots, x_n),$$

where R has degree $\leqslant d-1$ and S has degree $\leqslant d$. We separate into two cases. The first case is when $R(x_2, \dots, x_n) + S(x_2, \dots, x_n)$ is identically zero. In this case we have

$$P(x) = (x_1 + 1) \cdot R(x_2, \dots, x_n)$$

and so, by the inductive hypothesis,

$$\begin{aligned}\Pr[P(x) = 0] &= \Pr[x_1 = 1] + \Pr[x_1 = 0] \cdot \\ &\quad \Pr[R(x_2, \dots, x_n) = 0] \\ &\leqslant (1/2) + (1/2)(1 + 2^{-(d-1)}) \\ &= 1 - 2^{-d}.\end{aligned}$$

In the second case we have that $R(x_2, \dots, x_n) + S(x_2, \dots, x_n)$ is not identically zero. Now,

$$\begin{aligned}&\Pr[P(x) = 0] \\ &= \Pr[x_1 = 0] \cdot \Pr[S(x_2, \dots, x_n) = 0] \\ &\quad + \Pr[x_1 = 1] \cdot \Pr[R + S = 0] \\ &\leqslant (1/2) \cdot (1 - 2^{-d}) + (1/2)(1 - 2^{-d}) \\ &= 2^{-d},\end{aligned}$$

as was required. □

We now combine these two lemmas to show that there exists a linear restriction of P to a small number of variables that preserves independence of the coordinates of P.

Lemma 5.11. *Let $P : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ be a multilinear mapping of degree $\leqslant d$ such that $|\operatorname{Supp}(P)| \leqslant 2^k$, for $k, d \in \mathbb{N}$. Denote by $P_1, \dots, P_m \in \mathbb{F}_2[x_1, \dots, x_n]$ the coordinates of P. Suppose that $P_1, \dots, P_m$ are linearly independent (in the vector space $\mathbb{F}_2[x_1, \dots, x_n]$). Then, there exists an affine-linear mapping $L : \mathbb{F}_2^\ell \mapsto \mathbb{F}_2^n$ with $\ell = k + d$ such that the restricted polynomials $P_j(L(y_1, \dots, y_\ell)), j \in [m]$ are also independent.*

Proof. Apply Lemma 5.9 with $\varepsilon < 2^{-d}$ on the mapping P to find an affine-linear mapping $L : \mathbb{F}_2^\ell \mapsto \mathbb{F}_2^n$ with $\ell = k + d$ and such that

$$\Pr_{x \in \mathbb{F}_2^n}[\exists y \in \mathbb{F}_2^\ell \ , \ P(x) = P(L(y))] > 1 - 2^{-d}.$$

Call an element $x \in \mathbb{F}_2^n$ 'good' if the event above happens (so x is good w.p $> 1 - 2^{-d}$).

For $j \in [m]$ let $R_j(y_1, \dots, y_\ell) = P_j(L(y_1, \dots, y_\ell))$ (notice that since L is linear the polynomials R_j are also of degree at most d but are not necessarily multilinear). Suppose in contradiction that the polynomials $R_1, \dots, R_m$ are linearly dependent. So there is a non empty set $I \subset [m]$ such that $R_I(y) = \sum_{i \in I} R_i(y) = 0$ for every $y \in \mathbb{F}_2^\ell$. Let $P_I(x) = \sum_{i \in I} P_i(x)$. Then, if x is good we have that there exists y such that $P(x) = P(L(y))$ and so we get that

$$P_I(x) = P_I(L(y)) = R_I(y) = 0.$$

This means that $P_I(x)$, which is a multilinear polynomial of degree at most d, is zero on a fraction bigger than $1 - 2^{-d}$ of the inputs. Using Lemma 5.10 we conclude that $P_I(x)$ is identically zero and so the P_i's are linearly dependent – a contradiction. □

Corollary 5.12. *Let $P : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ be a multilinear mapping of degree $\leqslant d$ such that $|\operatorname{Supp}(P)| \leqslant 2^k$, for $k, d \in \mathbb{N}$. Denote by $P_1, \dots, P_m \in \mathbb{F}_2[x_1, \dots, x_n]$ the coordinates of P. Suppose that the set $P_1, \dots, P_m$ has rank $\geqslant r$ (in the vector space $\mathbb{F}_2[x_1, \dots, x_n]$). Then, there exists an affine-linear mapping $L : \mathbb{F}_2^\ell \mapsto \mathbb{F}_2^n$ with $\ell = k + d$ such that that the restricted polynomials $P_j(L(y_1, \dots, y_\ell)), j \in [m]$ also have rank $\geqslant r$*

Proof. W.l.o.g suppose that $P_1, \dots, P_r$ are linearly independent and apply Lemma 5.12 on the mapping $\tilde{P} = (P_1, \dots, P_r) : \mathbb{F}_2^n \mapsto \mathbb{F}_2^r$. The support of $\tilde{P}$ is also at most 2^k and so we $L : \mathbb{F}_2^\ell \to \mathbb{F}_2^n$ such that the restriction $\tilde{P}(L(y))$ has rank r. Now, adding the $m - r$ coordinates $P_{r+1}(L(y)), \dots, P_m(L(y))$ cannot decrease the rank. □

We are now ready to prove the Theorem.

Proof of Theorem 5.7. Let r denote the rank of the set of polynomials $\{P_1, \dots, P_m\}$. Then, using Corollary 5.12, there exists a linear mapping $L : \mathbb{F}_2^\ell \mapsto \mathbb{F}_2^n$, with $\ell = k + d$, such that the restricted polynomials $P_1(L(y)), \dots, P_m(L(y))$ also have rank $\geqslant r$. Since these are polynomials of degree $\leqslant d$ in ℓ variables, their rank is bounded from above by the number of different monomials of degree at most d in $\ell = k + d$ variables, which equals $\binom{\ell+d}{d}$. □

Acknowledgement

We thank Daniele Micciancio and Shrenik Shah for helpful discussions. Z. Dvir was partially supported by NSF grant CCF-0832797. Part of this research was done while D. Gutfreund was at Harvard University and supported by ONR grant N00014-04-1-0478 and NSF grant CNS-0430336. G. Rothblum was supported by NSF Grants CCF-0635297, CCF-0832797, CNS-0430336 and by a Computing Innovation Fellowship. S. Vadhan was supported by NSF grant CNS-0831289.

References

[1] D. Aharonov and O. Regev.Lattice problems in NP$\cap$coNP. *Journal of the ACM*, 52(5):749-765 (electronic), 2005.

[2] D. Aharonov and A. Ta-Shma. Adiabatic quantum state generation. *SIAM Journal on Computing*, 37(1):47-82 (electronic), 2007.

[3] W. Aiello and J. Håstad. Statistical zero-knowledge lan-guages can be recognized in two rounds. *Journal of Computer and System Sciences*, 42(3):327-345, 1991. Preliminary version in *FOCS'87.*

[4] M. Ajtai.Generating hard instances of lattice problems (extended abstract). In *Proceedings of the twenty-eighth annual ACM symposium on the theory of computing-STOC'96*, pages 99-108, Philadelphia, Pennsylvania, 22 May 1996. ACM.

[5] A. Akavia, O. Goldreich, S. Goldwasser, and D. Moshkovitz. On basing one-way functions on NP-hardness. In *STOC'06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 701-710, New York, 2006. ACM.

[6] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptogra-phy in NC^0 . *SIAM Journal on Computing*, 36(4):845- 888 (electronic), 2006.

[7] P. Beame, S. A. Cook, and H. J. Hoover. Log depth circuits for division and related problems. *SIAM J. Comput.*, 15(4):994-1003, 1986.

[8] M. Ben-Or and D. Gutfreund.Trading help for inter-action in statistical zero-knowledge proofs. *Journal of Cryptology*, 16(2):95-116, 2003.

[9] A. Bogdanov and L. Trevisan. On worst-case to average-case reductions for NP problems. *SIAM Journal on Computing*, 36(4):1119-1159 (electronic), 2006.

[10] A. Chailloux, D. F. Ciocan, I. Kerenidis, and S. Vadhan. Interactive and noninteractive zero knowledge are equiv-alent in the help model. In R. Canetti, editor, *Proceedings of the Third Theory of Cryptography Conference (TCC '08)*, volume 4948 of *Lecture Notes in Computer Science*, pages 501-534. Springer-Verlag, 19-21 March 2008.

[11] A. Chiu, G. Davida, and B. Litow. Division in logspace-uniform NC^1 . *Theoretical Informatics and Applica-tions*, 35(3):259-275, 2001.

[12] A. De Santis, G. Di Crescenzo, G. Persiano, and M. Yung. Image Density is complete for non-interactive-SZK. In *Automata, Languages and Programming, 25th International Colloquium, ICALP*, pages 784-795, 1998. See also preliminary draft of full version, May 1999.

[13] Z. Dvir, A. Gabizon, and A. Wigderson.Extractors and rank extractors for polynomial sources. In *FOCS,* pages 52-62. IEEE Computer Society, 2007.

[14] J.Feigenbaum and L. Fortnow. Random-self-reducibility of complete sets. *SIAM J. Comput.*, 22(5):994-1005, 1993.

[15] L. Fortnow. The complexity of perfect zero-knowledge. *Advances in Computing Research: Randomness and Computation*, 5:327-343, 1989.

[16] O. Goldreich and S. Goldwasser.On the limits of non-approximability of lattice problems. In *Proceedings of the 30th Annual ACM Symposium on Theory of Comput-ing (STOC)*, pages 1-9, 1998.

[17] O.Goldreich and E. Kushilevitz. A perfect zero-knowledge proof system for a problem equivalent to the discrete logarithm. *Journal of Cryptology*, 6:97-116, 1993.

[18] O.Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(1):691-729, 1991. Preliminary version in *FOCS'86.*

[19] O. Goldreich, A. Sahai, and S. Vadhan.Honest ver-ifier statistical zero-knowledge equals general statisti-cal zero-knowledge. In *Proceedings of the 30th An-nual ACM Symposium on Theory of Computing (STOC)*, pages 399-408, 1998.

[20] O. Goldreich, A. Sahai, and S. Vadhan. Can statistical zero-knowledge be made non-interactive?, or On the relationship of SZK and NISZK. In *Advances in Cryp-tology - CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 467-484. Springer, 1999.

[21] O. Goldreich and S. P. Vadhan.Comparing entropies in statistical zero knowledge with applications to the structure of SZK. In *IEEE Conference on Computa-tional Complexity*, pages 54-73. IEEE Computer Society, 1999.

[22] S. Goldwasser, S. Micali, and C. Rackoff.The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186-208, 1989. Preliminary version in *STOC'85.*

[23] Y. Ishai and E. Kushilevitz.Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS 2000)*, pages 294-304, 2000.

[24] J. Katz and Y. Lindell.*Introduction to Modern Cryp-tography (Chapman & Hall/Crc Cryptography and Net-work Security Series).* Chapman & Hall/CRC, 2007.

[25] M. J. Kearns and L. G. Valiant.Cryptographic limita-tions on learning boolean formulae and finite automata. *J. ACM,* 41(1):67-95, 1994.

[26] R. Lidl and H. Niederreiter.*Introduction to finite fields and their applications.* Cambridge University Press, Cambridge, first edition, 1994.

[27] M. Mahmoody and D. Xiao.On the power of randomized reductions and the checkability of SAT. In *Proceedings of the 25'th IEEE Conference on Computational Complexity (CCC 2010)*, pages 64-75, 2010.

[28] L.Malka. How to achieve perfect simulation and a complete problem for non-interactive perfect zero-knowledge. In *Proceedings of the 5th Theory of Cryp-tography Conference, (TCC 20078)*, pages 89-106, 2008.

[29] D. Micciancio and S. Goldwasser.*Complexity of Lat-tice Problems: a cryptographic perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science.* Kluwer Academic Publishers, Boston, Massachusetts, Mar. 2002.

[30] M. Naor and O. Reingold.Number-theoretic construc-tions of efficient pseudo-random functions. *J. ACM*, 51(2):231-262, 2004.

[31] T. Okamoto.On relationships between statistical zero-knowledge proofs. In *Proceedings of the 28th An-nual ACM Symposium on Theory of Computing (STOC)*, pages 649-658. ACM Press, 1996.

[32] S. J. Ong and S. Vadhan. Zero knowledge and soundness are symmetric. Technical Report TR06-139, Electronic Colloquium on Computational Complexity, 2006.

[33] R. Ostrovsky. One-way functions, hard on average prob-lems, and statistical zero-knowledge proofs. In *Proceed-ings of the 6th Annual Structure in Complexity Theory Conference*, pages 133-138. IEEE Computer Society, 1991.

[34] G. Rothblum and S. Vadhan.Cryptographic primitives and the average-case hardness of SZK. Unpublished Manuscript, 2009.

[35] A. Sahai and S. Vadhan. A complete problem for statis-tical zero knowledge. *Journal of the ACM*, 50(2):196-249, 2003. Preliminary version in *FOCS'97.*

[36] S.P. Vadhan. An unconditional study of computa-tional zero knowledge. *SIAM Journal on Comput-ing*, 36(4):1160-1214, 2006. Preliminary version in *FOCS'04.*