On the Distribution of the Number of Roots of Polynomials and Explicit Weak Designs*

Tzvika Hartman† Ran Raz ‡

May 26, 2003

Abstract

Weak designs were defined in [RRV99] and are used there in constructions of extractors. Roughly speaking, a weak design is a collection of subsets satisfying some near-disjointness properties. Constructions of weak designs with certain parameters are given in [RRV99]. These constructions are explicit in the sense that they require time and space polynomial in the number of subsets. However, the constructions require time and space polynomial in the number of subsets even when needed to output only one specific subset out of the collection. Hence, the constructions are not explicit in a stronger sense.

In this work we provide constructions of weak designs (with parameters similar to the ones of [RRV99]) that can be carried out in space logarithmic in the number of subsets. Moreover, our constructions are explicit even in a stronger sense: given an index to a subset, we output the specified subset in time and space polynomial in the size of the index. Using our constructions, we obtain extractors similar to some of the ones given in [RRV99] in terms of parameters, and that can be evaluated in logarithmic space.

Our main construction is algebraic. In order to prove the properties of weak designs we prove some algebro-combinatorial lemmas that may be interesting in their own right. These lemmas regard the number of roots of polynomials over finite fields. In particular, we prove that the number of polynomials (over any finite field) with k roots, vanishes exponentially in k. In other words, we prove that the number of roots of a random polynomial is not only bounded by its degree (a well known fact), but furthermore, it is concentrated exponentially around its expectation (which is 1) [1] . Our lemmas are proved by algebro-combinatorial arguments. The main lemma is also proved by a probabilistic argument.

1 Introduction

1.1 Weak Designs

Weak designs were defined in [RRV99]. Roughly speaking, a weak design is a collection of subsets, satisfying some near-disjointness properties. In particular, we take the exponentiation of the size of the intersection between every pair of subsets and bound its average (see definition 3).

*Preliminary versions of this work appeared in *RANDOM2000* [HR00a] and on *ECCC* [HR00b].

†Dept. of Applied Math and Computer Science, Weizmann Institute, Rehovot 76100, Israel. E-mail: tzvi@wisdom.weizmann.ac.il.

‡Dept. of Applied Math and Computer Science, Weizmann Institute, Rehovot 76100, Israel. E-mail: ranraz@wisdom.weizmann.ac.il. Phone: +972-8-934-3544

[1]Some related results have appeared in several papers by the Knopfmachers [KK90, KK93, K99] and are also known in coding theory (see, e.g., [MS77, Ber68]).

[RRV99] provide explicit constructions of weak designs which output all the subsets in time and space polynomial in the number of subsets. However, they require this amount of time and space even when needed to output only one subset. In other words, if we consider a stronger notion of explicitness (as in definition 4), where the algorithm gets as an input an index of a subset and is required to output only the specified subset, then the constructions are not explicit. This is since the time and space required are exponential in the size of the index (which is logarithmic in the number of subsets).

The main application of weak designs in [RRV99] is in constructing *extractors* (they are also used for constructing Nisan-Wigderson-like pseudorandom generators [NW94]). Extractors, first defined by Nisan and Zuckerman [NZ96], are functions that have very strong randomness properties. Extractors have a wide variety of applications. Note that many of the existing constructions of explicit extractors are not simple and their space complexity is usually not specified. It seems, however, that the space complexity of the best known extractors is usually large. For example, the extractor of [NT99] is computable in linear space, and the one of [Zuc97] is computable in NC. Subsequent works [ByGW99, BRST02] analyze the space complexity of *online* extractors. For more about the time and space complexity see, e.g., [NT99, Tre01, RRV99, S02].

Trevisan [Tre01] uses the Nisan-Wigderson pseudorandom generator [NW94] in order to construct extractors. The Nisan-Wigderson generator and hence the Trevisan Extractor are based on a "combinatorial design", which is a collection of subsets of a given universe, in which the intersection between every two subsets is bounded. The main observation in [RRV99] is that Trevisan's construction yields an extractor even if we replace these combinatorial designs by a weaker notion of combinatorial designs. They introduced the notion of *weak design* and proved that in some cases, the same construction, based on weak designs, induces an extractor with better parameters.

Constructions of weak designs are given by [RRV99]. The properties of the design are proved by an application of the Probabilistic Method. The proof is derandomized by the Method of Conditional Probabilities. This construction, as discussed above, can be done in time polynomial in the number of subsets. However, this construction requires also polynomial space. This implies that the evaluation of the RRV extractor cannot be done in small space.

1.2 Overview of Our Results

In this work we use a different approach, an algebraic one, for constructing weak designs. This approach was introduced by Nisan and Wigderson [NW94] in order to construct standard designs (see definition 2 and lemma 2.2). By using this approach we obtain (in section 4) constructions of weak designs that have similar parameters to the ones constructed in [RRV99], with the advantage that they are explicit in a stronger sense: We provide an algorithm that gets as an input an index to a subset and outputs the specified subset in time and space polynomial in the size of the index of the subset (the time depends polynomially also on the size of the subsets, and the space depends logarithmically also on the size of the subsets). This improves upon the constructions of [RRV99], in which the time and space required is exponential in the size of the input. Moreover, our algorithm can output *all* the subsets in space which is logarithmic in the number of subsets, whereas the RRV constructions require polynomial space [2].

By plugging these designs into some of the RRV extractors we obtain extractors which can be evaluated in space logarithmic in its input [3]. Plugging the designs into the Nisan-Wigderson

[2]A different construction of weak designs in logarithmic space was suggested by one of the referees. However, this construction cannot be done in logarithmic time (see remark 6.1).

[3]An alternative way of obtaining similar logspace extractors was suggested to us by Oded Goldreich, Salil Vadhan

pseudorandom generator [NW94] can yield a logspace pseudorandom generator, which may be useful in derandomizing low complexity classes (see subsection 6.3) [4].

In order to obtain our main result, we prove (in section 3) some algebro-combinatorial lemmas that may be interesting in their own right. The lemmas regard the number of roots and the number of linear factors of polynomials over finite fields. Roughly speaking, we prove that the number of polynomials (over any finite field) with k roots (or k linear factors) vanishes exponentially in k. In other words, we prove that the number of roots of a random polynomial is not only bounded by its degree (a well known and useful fact), but furthermore, is typically much smaller. In particular, it is concentrated exponentially around its average, which is 1. [5]

The lemmas are proved by algebro-combinatorial techniques. A slightly different version of our main lemma (which is used for our main result) is also proved by a probabilistic argument. This is done using the fact that there is a limited dependency between distinct evaluations of a random polynomial. By using tail inequalities for random variables with limited independence, we show that this limited dependency suffices to prove our result.

Our algebraic construction yields a weak design where some of the parameters have specific values (these specific values are the most interesting values for constructing extractors). In addition, we provide (in section 5) general tools to take a weak design with specific parameters and construct a new weak design with other parameters (improving some parameters while paying the price in other ones). These tools use techniques such as union of subsets of several designs and direct product of designs. Applying these tools to our main construction yields a more general construction, which matches, up to a constant factor, the constructions of [RRV99] for extracting at least a constant fraction of the min-entropy (and have the advantage of being explicit in the strong sense). A variant of these tools, combined with error correcting codes, is used to obtain better explicit standard designs as well (see subsection 5.3)[6].

1.3 Organization of the Paper

In section 2 we define extractors and combinatorial designs, and describe some known constructions of combinatorial designs. Section 3 contains some algebro-combinatorial lemmas, which will be used in our main construction. In section 4 we prove our main result - an explicit construction of weak-designs for specific parameters. Then, in section 5, we generalize this construction to get explicit constructions of weak designs with parameters that match, up to constant factors, the parameters of the probabilistic constructions in [RRV99]. We describe some applications of our results in section 6, and conclude with a short discussion (section 7).

2 Preliminaries

2.1 Extractors

The *statistical difference* (or *variation difference*) between two distributions X and Y over $\{0,1\}^m$ is defined as

$$\max_D |Pr[D(X)=1] - Pr[D(Y)=1]|$$

and Avi Wigderson (remark 6.1).

[4] Suggested to us by Salil Vadhan.

[5] Some related results regarding the number of polynomials with a certain number of roots and factors have appeared in several papers by the Knopfmachers [KK90, KK93, K99] and are also known in coding theory (see, e.g., [MS77, Ber68]). More details are given in section 3.

[6] Suggested to us by Salil Vadhan.

where the maximum is taken over all functions (sometimes called "distinguishers") $D : \{0,1\}^m \to \{0,1\}$. We say that two distributions are $\epsilon - close$ if their statistical difference is at most ϵ.

A distribution X is said to have *min-entropy* k if for all $x \in \{0,1\}$,

$$Pr[X = x] \leq 2^{-k}$$

We will denote the uniform distribution on binary strings of length m by U_m.

We are now ready to define extractors:

Definition 1 (Extractor) *A function $EXT : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is called a (k,ϵ)-*extractor *if for every distribution X on $\{0,1\}^n$ with min-entropy k , $EXT(X, U_d)$ is $\epsilon - close$ to U_m.*

An extractor actually *extracts* m bits of randomness from the given n bits, as long as these n bits have min-entropy k. In order to do that, it needs additional d truly random bits. In extractors, we would like to minimize d (the number of additional truly random bits), while keeping m (the number of extracted "almost" random bits) as close to k (the amount of randomness in the source) as possible. It can be shown, non-constructively (by a probabilistic argument), that for every n and $k \leq n$ and $\epsilon > 0$ there exists a (k,ϵ)-extractor $EXT : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$, with $m = k$ and $d = O(\log(n/\epsilon))$, i.e., it is possible to extract all the randomness of the source using only a logarithmic number of additional truly random bits. This gives an optimal non-explicit construction of extractors (see lower bounds in [NZ96]). However, we are interested in explicit constructions of extractors.

A family of extractors $\{EXT_i : \{0,1\}^{n_i} \times \{0,1\}^{d_i} \to \{0,1\}^{m_i}\}_{i \in \mathbb{N}}$ is called *explicit* if EXT_i can be evaluated in time poly(n_i). Notice that we did not consider the parameters d_i in the complexity measurement since an extractor is interesting mainly if d_i is smaller than n_i. For the sake of readability, we state our definitions and theorems as if we are referring to individual extractors though they all actually refer to families of extractors.

The following is a stronger notion of explicitness: An explicit (k,ϵ)-extractor $EXT : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ will be called an explicit *logspace* extractor if EXT is computable in space $s = O(\log(n/\epsilon))$.

2.2 Combinatorial Designs

2.2.1 Definitions

Roughly speaking, a combinatorial design is a collection of subsets of a given universe such that the subsets are "nearly disjoint". The difference between the types of designs is the definition of "nearly disjoint" (i.e., the condition restricting the intersections between the subsets). In the combinatorics literature, these combinatorial designs are often called *packings* (cf., [AS00, Sec. 4.7]).

In a standard combinatorial design the intersection of every two subsets is bounded:

Definition 2 (Standard Design) *A family of sets $S_1, \ldots, S_m \subset [d]$ is called a* standard (m,l,ρ,d)-design *if*

1. *For all $1 \leq i \leq m$, $|S_i| = l$.*
2. *For all $1 \leq i \neq j \leq m$, $|S_i \cap S_j| \leq \log \rho$.*

Standard designs underly the construction of the Nisan-Wigderson generator and the Trevisan extractor. [RRV99] observed that a weaker notion of combinatorial designs, which they call *weak designs* suffices for the constructions.

Definition 3 (Weak Design [RRV99]) [7] *A family of sets* $S_1, \ldots, S_m \subset [d]$ *is called a* weak (m, l, ρ, d)-design *if*

1. *For all* $1 \le i \le m$, $|S_i| = l$.
2. *For all* $1 \le i \le m$,
$$\sum_{j<i} 2^{|S_i \cap S_j|} \le \rho \cdot m$$

Notice that every standard (m, l, ρ, d)-design is in particular a weak (m, l, ρ, d)-design. The advantage of weak designs is that they can be constructed with better parameters (see subsection 2.2.2). Our results (except theorem 4) actually hold for a stronger version of weak design, in which the second condition is:
For all $1 \le i \le m$,
$$\sum_{j \neq i} 2^{|S_i \cap S_j|} \le \rho \cdot m$$
However, for the sake of simplicity, we will not give a different name to this type of weak design.

In the following we consider some possible notions of "explicitness" in the construction of combinatorial designs. We may require from the constructing algorithm to output *all* subsets of the design. In this view, a construction will be called *explicit* if the time and space needed is polynomial in the number of subsets. Another possibility is that the algorithm will get as an input an index to a subset and an index to an element, and will output the specified element of the specified subset. Here explicitness means that the time and space complexity is polynomial in the size of the indices (that size is the logarithm of the number of subsets plus the logarithm of the size of each subset).

We will consider the following definition of explicitness. In this definition, the algorithm gets as an input an index to a subset, and is required to output the specified subset.

Definition 4 (Explicit Combinatorial Design) *A combinatorial (weak or standard) design* $S = S_1, \ldots, S_m \subset [d]$*, where the size of each set is* l*, is called an* explicit *combinatorial design if there exists an algorithm that, given* $1 \le i \le m$*, outputs the subset* S_i *in time* $T = \mathrm{poly}(\log m, l)$ *and space* $S = O(\log m + \log l)$.

The complexity is not related to the parameter d, since there is always a trivial design with $d = m \cdot l$, so the elements of $[d]$ will always have a representation of size $O(\log(ml))$.

2.2.2 Known Constructions

As mentioned above, our main motivation for defining combinatorial designs is for construction of extractors and pseudorandom generators. The parameters of the extractors are hence related to the parameters of the underlying designs. In this subsection, we state some known constructions of designs and see the parameters of the extractors they imply. In general, the relation between the parameters of the design and those of the extractor (obtained by the constructions of [Tre01, RRV99]) is as follows. The size of the universe of the design, d, is the number of truly random

[7] This is a slight modification of the definition in [RRV99], where the second condition is $\sum_{j<i} 2^{|S_i \cap S_j|} \le \rho \cdot (m-1)$

bits that the extractor needs. The size of each subset, l, is (roughly) the logarithm of the length of the input (for the extractor), n. The number of subsets, m, is the length of the output of the extractor. The parameter ρ is roughly k/m (where k is the min entropy of the source). For the sake of simplicity, in the following discussion we ignore ϵ, the error parameter of the extractor.

In constructing extractors, we would like to keep d (the number of truly random bits), as small as possible (as mentioned above, the best we can hope for is $d = O(\log n)$). This implies that we should minimize the size of the universe, d, of the designs. In the following we will see that the main contribution of introducing the notion of weak designs [RRV99] is in the fact that the universe is smaller, implying an extractor which requires a smaller number of truly random bits. We also want to extract as much randomness as possible, i.e., we want that m will be close to k. This implies that in the designs, ρ should be as close to 1 as possible.

We will now describe the known constructions of combinatorial designs, and see the parameters of the extractors induced by these constructions. Let us first describe a construction of standard designs based on probabilistic construction.

Lemma 2.1 ([EFF85, NW94, Vad02]) *For every $m, l, d \in \mathsf{N}$ and $\rho > 1$, such that $m \leq \frac{\binom{d}{\lfloor \log 2\rho \rfloor}}{\binom{l}{\lfloor \log 2\rho \rfloor}^2}$, there exists a standard (m, l, ρ, d)-design $S_1, \ldots, S_m \subset [d]$. In particular, for every $m, l \in \mathsf{N}$, and $\rho > 1$, there exists a standard design with*

$$d = O\left(\frac{l^2 m^{1/\log \rho}}{\log \rho}\right).$$

This design can be constructed in time $\mathrm{poly}(m, d)$ *and space* $\mathrm{poly}(m)$.

This standard design yields an extractor [Tre01] that for any source of min entropy k, extracts m bits using $d \approx O\left(\frac{\log^2 n \cdot m^{1/\log(k/m)}}{\log(k/m)}\right)$ additional random bits. The most interesting case is when $m = k^{1-\alpha}$ (α is an arbitrary small constant), in which $d = O\left((\log^2 n)/(\log k)\right)$. The algorithm for constructing the above design chooses sequentially (by exhaustive search) m subsets of $[d]$ such that each one intersects the previous ones in at most $\log \rho$ points. A probabilistic argument (using Chernoff bounds) shows that it is possible to choose m such subsets. Note that this construction requires the same time and space complexity even if we want to output only one subset and hence, it is not explicit in the sense of definition 4. However, we want to stress that this construction implies an indeed explicit construction of extractors, since in the extractor case, complexity is measured according to the input size of the extractor.

An explicit construction of standard designs with weaker parameters is given by the following:

Lemma 2.2 ([NW94]) *For every $m,l \in \mathsf{N}$ there exists an explicit standard (m, l, ρ, d)-design $S_1, \ldots, S_m \subset [d]$ with $d = O(l^2)$ and $\rho = m^{1/\log l}$.*

The construction uses polynomials over finite fields and is the construction we use in our main result (section 4). Roughly speaking, each subset is identified with a polynomial over a fixed finite field. The subsets are constructed such that the intersection between two subsets corresponds to the values at which the two polynomials (identified with the two subsets) agree. The proof that these subsets induce a standard design is based on the fact that the number of values that two polynomials can agree on is bounded by their maximal degree. Our proof that these subsets induce a weak design with better parameters is based on a more refined argument, showing that polynomials typically agree on a very small number of values.

For weak designs, there is the following construction:

Lemma 2.3 ([RRV99]) *For every $l, m \in \mathbb{N}$ and $\rho > 1$, there exists a weak (m, l, ρ, d)-design $S_1, \ldots, S_m \subset [d]$ with*

$$d = \left\lceil \frac{l}{\ln \rho} \right\rceil \cdot l$$

This design can be constructed in time $\mathrm{poly}(m, d)$ *and space* $\mathrm{poly}(m)$.

This construction is much better than the construction of standard designs of lemma 2.1 in terms of the parameter d. Specifically, if ρ is constant, here $d = O(l^2)$, whereas $d = l^2 m^{\Omega(1)}$ in the corresponding standard design. In terms of extractors, if k/m is constant, the implied extractor needs only $O(\log^2 n)$ truly random bits, improving upon the $m^{\Omega(1)} \log^2 n$ bits needed in Trevisan's extractor in this setting of parameters.

We want to stress that the case where ρ is constant (or in terms of extractors k/m is constant) is a very important case. It means that we can extract a constant fraction of the source min-entropy and by using iterations (as in [WZ99]) on these extractors we obtain an extractor that extracts *all* the source min-entropy.

The proof is by a probabilistic argument: The universe $[d]$ is divided into l blocks, each block of size $\lceil l/\ln \rho \rceil$. The subsets are constructed sequentially such that each subset contains exactly one element from each block. It is shown that after choosing the subsets $S_1, \ldots, S_{i-1}$, for every new subset S_i, the expected value of $\sum_{j<i} 2^{|S_i \cap S_j|}$ is smaller (or equal) to ρm. Hence, with non-zero probability, there exists a set S_i satisfying the requirements. The construction is derandomized by the Method of Conditional Expectations (see [AS00, MR95]). In order to find the i^{th} subset we must compute and store all previous $i-1$ subsets (so we can calculate the appropriate conditional expectation). Thus, the algorithm requires time and space polynomial in m even if it is required to output only one specific subset. Hence, the construction is not explicit in the sense of definition 4 (however, as in lemma 2.1 it implies an explicit extractor).

The shortcoming of the parameters in the above construction is that in case ρ is very close to 1, d gets very large. Specifically, if $\rho = 1 + \gamma$ for small γ, then the above construction gives $d = O(l^2/\gamma)$. This means that the induced extractor performs poorly (in terms of the number of additional truly random bits) if we want to extract all (or almost all) the min-entropy of the source. The following construction improves d significantly in this case.

Lemma 2.4 ([RRV99]) *For every $l, m \in \mathbb{N}$ and $3/m \le \gamma < 1/2$, there exists a weak $(m, l, 1+\gamma, d)$-design with*

$$d = O\left(l^2 \cdot \log \frac{1}{\gamma}\right).$$

This design can be constructed in time $\mathrm{poly}(m, d)$ *and space* $\mathrm{poly}(m)$.

The implied extractor needs $O(\log^2 n \cdot \log(1/\gamma))$ additional random bits, where $1 + \gamma \approx k/m$. The most interesting case is when we want to extract all the min entropy of the source (i.e., $m = k - O(1)$), in which the extractor requires only $O(\log^2 n \cdot \log k) = O(\log^3 n)$ additional truly random bits. This construction is accomplished by iterating the construction of lemma 2.3 (inspired by the iterations of Wigderson and Zuckerman [WZ99] on extractors).

3 Algebro-Combinatorial Lemmas

A trivial counting argument shows that the number of polynomials with a bounded degree over a finite field is the size of the field to the power of the degree plus one. In this section we encounter the question of what happens to this number if we restrict the polynomials to be with a specific number of roots or a specific number of linear factors. The following lemmas provide answers to some versions of this question. The lemmas, especially lemmas 3.6 and 3.9 (which are the most interesting ones) can be viewed in the following way:

A fundamental theorem from algebra states that the number of roots of a (non-zero) polynomial is bounded by its degree. A simple observation shows that the expected number of roots of a random polynomial over a finite field is much smaller, namely, one. To see this, observe that for any point a in a field F and for a random polynomial p, the probability that $p(a) = 0$ is $1/|F|$ (even when the probability is just over the choice of the constant term of p).

Lemma 3.6 shows that not only the expected number of roots of a random polynomial is one, but furthermore, this number is very concentrated around its expectation. Namely, the number of polynomials with many roots vanishes exponentially in the number of roots. This result is used for the explicit construction of weak designs in the next section and we believe that it may be interesting in other applications as well.

As mentioned above, some related results have already appeared in several papers by the Knopfmachers [KK90, KK93, K99] and are also known in coding theory (see, e.g., [MS77, Ber68]). Our lemmas are more specific and hence our proofs are simpler. Throughout this section, we will discuss the relevant related results for each lemma.

We assume the reader is familiar with the following basic fact from algebra: $\alpha_1, \ldots, \alpha_k$ are roots of the polynomial $p(x)$ if and only if $\Pi_{i=1}^{k}(x - \alpha_i)$ divides $p(x)$ (It can be found in every standard text book on algebra, e.g. [Lan68] sec. 7, see also [Sud98] lec. 2).

In section 3.2 we state our lemmas and prove them using combinatorial arguments. In section 3.3 we prove lemma 3.9, which provides an upper bound which is slightly different than the one given in lemma 3.6. This proof is by a probabilistic argument.

3.1 Preliminaries

Let F be a finite field of size q. In the following, $F[x]$ is the ring of polynomials over F and $p(x) \in F[x]$. Define

- $deg(p) \stackrel{\text{def}}{=}$ the degree of $p(x)$.
- $\sharp L(p) \stackrel{\text{def}}{=}$ the number of linear factors in the factorization of $p(x)$ over F (counting repetitions). In other words, it is the sum of the multiplicities of all the roots of the polynomial $p(x)$.
- $\sharp R(p) \stackrel{\text{def}}{=}$ the number of distinct roots of $p(x)$ in F (i.e., the number of elements $\alpha \in F$ such that $p(\alpha) = 0$).

Every distinct root over F implies a linear factor over F, thus, for every nonzero $p(x) \in F[x]$, it holds that $\sharp R(p) \leq \sharp L(p)$.

The following are subsets of $F[x]$:

1. $$L_q(k, d) \stackrel{\text{def}}{=} \{p(x) \in F[x] \mid (\sharp L(p) = k) \wedge (deg(p) \leq d) \wedge (p \neq \overline{0})\}$$

   ($L_q(k, d)$ is the set of non-zero polynomials over F of degree at most d that have exactly k linear factors over F.) [8]

[8] The zero polynomial is a special case in which there are q roots but no linear factors. Therefore, in these definitions, we prefer to define the sets without the zero polynomial.

2.
$$R_q(k,d) \stackrel{\text{def}}{=} \{p(x) \in F[x] \mid (\sharp R(p) = k) \wedge (deg(p) \leq d) \wedge (p \neq \overline{0})\}$$

($R_q(k,d)$ is the set of non-zero polynomials over F of degree at most d that have exactly k distinct roots in F. The multiplicity of the roots can be arbitrary.)

3.
$$LR_q(k,d) \stackrel{\text{def}}{=} \{p(x) \in F[x] \mid (\sharp L(p) = k) \wedge (\sharp R(p) = k) \wedge (deg(p) \leq d) \wedge (p \neq \overline{0})\}$$

($LR_q(k,d)$ is the set of non-zero polynomials over F of degree at most d that have exactly k linear factors over F and exactly k distinct roots in F. An equivalent way is to say that the polynomials have exactly k distinct roots in F, all of them with multiplicity 1.)

Notice that by definition, $LR_q(k,d) = L_q(k,d) \cap R_q(k,d)$. Since having no linear factors over F is equivalent to having no roots in F, we get that $L_q(0,d) = R_q(0,d) = LR_q(0,d)$.

3.2 The Lemmas

The following lemmas regard the sizes of the sets defined above. In each lemma, the first part provides a formula for calculating the relevant value, whereas the second part provides an upper bound for this value. In the whole discussion, when we talk about linear factors of a polynomial, we assume, without loss of generality, that these factors are monic (i.e., in each factor, the coeffecient of x is 1). Therefore, we can identify a linear factor with an element of the field, i.e., each element $\alpha \in F$ is identified with the linear factor $(x - \alpha)$. We restrict ourselves to cases in which $d < q$, i.e., the degree of the polynomial is smaller than the size of the field.

Lemma 3.1 *For all prime powers q and integers $d < q$,*

1.
$$|L_q(0,d)| = |R_q(0,d)| = |LR_q(0,d)| = \sum_{i=0}^{d} (-1)^i \cdot \binom{q}{i} \cdot (q^{d-i+1} - 1)$$

2.
$$|L_q(0,d)| = |R_q(0,d)| = |LR_q(0,d)| \leq q^{d+1} \cdot \left(\frac{1}{e} + \frac{1}{d!}\right)$$

Remark 3.2 *Part 1 of this lemma is a special case of lemma 3.6 and is already known (see references in remark 3.7).*

Proof: As noticed above, $L_q(0,d) = R_q(0,d) = LR_q(0,d)$ and therefore, it suffices prove the lemma for $LR_q(0,d)$.

The formula in the first part is an inclusion-exclusion formula. The zero'th summand is the total number of non-zero polynomials of degree at most d, namely $q^{d+1} - 1$. Now we want to subtract the number of non-zero polynomials that have at least one root (denoted by $\alpha \in F$). Every such polynomial is of the form $(x - \alpha) \cdot g(x)$, where $g(x)$ is a non-zero polynomial with degree at most $d-1$. There are q possible values for α and $q^d - 1$ possibilities for $g(x)$. Hence, we subtract $q(q^d - 1)$. However, every polynomial with at least 2 roots was counted at least twice. Such a polynomial has the form $(x - \alpha_1)(x - \alpha_2)g(x)$, where $\alpha_1, \alpha_2 \in F$ are the roots and $g(x)$ is a non-zero polynomial of degree at most $d - 2$. Hence, we add $\binom{q}{2}(q^{d-1} - 1)$. We continue this argument until we reach the maximal number of roots, namely, d. A polynomial with t roots is counted exactly once by the identity $t - \binom{t}{2} + \binom{t}{3} - \ldots \binom{t}{t} = 1$. This proves the inclusion-exclusion formula in the first part of the lemma.

For the second part we will write the expression of the first part as a sum of two different sums and bound each sum separately:

$$\sum_{i=0}^{d}(-1)^i \cdot \binom{q}{i} \cdot (q^{d-i+1} - 1) = \sum_{i=0}^{d}(-1)^i \cdot \binom{q}{i} \cdot q^{d-i+1} + \sum_{i=0}^{d}(-1)^{i+1} \cdot \binom{q}{i} \tag{1}$$

We begin by bounding the first sum. First, we consider the case when d is odd.

$$\sum_{i=0}^{d}(-1)^i \cdot \binom{q}{i} \cdot q^{d-i+1} = q^{d+1} \cdot \sum_{i=0}^{\frac{d-1}{2}} \frac{\binom{q}{2i}}{q^{2i}} - \frac{\binom{q}{2i+1}}{q^{2i+1}}$$

$$= q^{d+1} \cdot \sum_{i=0}^{\frac{d-1}{2}} \frac{q \cdot \frac{q!}{(2i)!(q-2i)!} - \frac{q!}{(2i+1)!(q-2i-1)!}}{q^{2i+1}} = q^{d+1} \cdot \sum_{i=0}^{\frac{d-1}{2}} \frac{\frac{q!}{(2i+1)!(q-2i)!} \cdot 2i(q+1)}{q^{2i+1}}$$

$$= q^{d+1} \cdot \sum_{i=0}^{\frac{d-1}{2}} \frac{2i \cdot \binom{q+1}{2i+1}}{q^{2i+1}} \leq q^{d+1} \cdot \sum_{i=0}^{\frac{d-1}{2}} \frac{2i}{(2i+1)!}$$

$$= q^{d+1} \cdot \sum_{i=0}^{\frac{d-1}{2}} \frac{1}{(2i)!} - \frac{1}{(2i+1)!} = q^{d+1} \cdot \sum_{i=0}^{d} \frac{(-1)^i}{i!} \leq q^{d+1} \cdot \frac{1}{e}$$

Above, the last inequality is obtained since the last summand is odd.

Now we calculate the second sum in equation 1 by using a telescopic sum:

$$\sum_{i=0}^{d}(-1)^{i+1} \cdot \binom{q}{i} = \sum_{i=0}^{d}(-1)^{i+1} \cdot \left[\binom{q-1}{i-1} + \binom{q-1}{i}\right] = (-1)^{d+1}\binom{q-1}{d} \tag{2}$$

which is at most $\frac{q^{d+1}}{d!}$. Overall, our upper bound for odd d is $q^{d+1} \cdot \left(\frac{1}{e} + \frac{1}{d!}\right)$.

In case d is even, we will first bound the first sum in equation 1 when i goes from 0 to $d-1$. The same calculations (as in the odd case) will hold, since $d-1$ is odd. Now we add the last summand, which is $\binom{q}{d}q \leq \frac{q^{d+1}}{d!}$. By equation 2, the second sum in equation 1 is negative for even d. Thus, for arbitrary d,

$$|LR_q(0,d)| \leq q^{d+1} \cdot \left(\frac{1}{e} + \frac{1}{d!}\right)$$

■

Lemma 3.3 *For all prime powers q and integers $k \leq d < q$,*

1.

$$|L_q(k,d)| = |L_q(0,d-k)| \cdot \binom{k+q-1}{k}$$

2.

$$|L_q(k,d)| \leq q^{d+1} \cdot \frac{2^k}{k!}\left(\frac{1}{e} + \frac{1}{(d-k)!}\right)$$

Remark 3.4 *A similar formula to the one in part 1, for counting monic polynomials with exactly k linear factors in the case of degree larger or equal than the size of the field (i.e., $d \geq q$), is given in [KK90].*

Proof: Let $p(x) \in L_q(k,d)$ (i.e., $p(x)$ is a polynomial over F of degree at most d that has exactly k linear factors over F). Therefore, $p(x)$ is of the form $f(x) \cdot g(x)$, where $f(x)$ is a polynomial of degree k which is the product of the k monic linear factors of $p(x)$ (notice that since we require that the linear factors are monic, $f(x)$ is uniquely determined) and $g(x)$ is a polynomial of degree at most $d-k$ with no linear factors. Thus, $g(x) \in L_q(0, d-k)$. Ranging over all the possibilities for $f(x)$ and $g(x)$ we get all the possibilities of $p(x)$.

$f(x)$ is the product of k monic linear factors, where each linear factor is identified with one of the q elements of F (recall that the same linear factor may appear more than once in the product). Hence, the number of possibilities for $f(x)$ is the number of ways to choose a multi-set of k elements out of a universe of size q, namely, $\binom{k+(q-1)}{k}$ (it can be viewed as the number of possibilities of throwing k identical balls into q bins). Thus, we get,

$$|L_q(k,d)| = |L_q(0,d-k)| \cdot \binom{k+q-1}{k}$$

This proves the first part of the lemma.

For the second part of the lemma we first bound

$$\binom{k+q-1}{k} \leq \frac{(k+q-1)^k}{k!} \leq \frac{(2q)^k}{k!} = q^k \cdot \frac{2^k}{k!}$$

Above, the second inequality is due to the fact that the number of linear factors, k, is obviously bounded by the degree of the polynomial, d, which is smaller than q. Finally, by using the upper bound of lemma 3.1, we have

$$|L_q(k,d)| = |L_q(0,d-k)| \cdot \binom{k+q-1}{k} \leq q^{d-k+1}\left(\frac{1}{e} + \frac{1}{(d-k)!}\right) \cdot q^k \cdot \frac{2^k}{k!} = q^{d+1} \cdot \frac{2^k}{k!}\left(\frac{1}{e} + \frac{1}{(d-k)!}\right)$$

■

Comment: For small values of k (relative to q), a better bound can be obtained. First we bound

$$(k+q-1)^k \leq q^k \cdot (k/q+1)^k = q^k \cdot \left[(k/q+1)^{q/k}\right]^{k^2/q} \leq q^k \cdot e^{k^2/q}$$

Which implies

$$|L_q(k,d)| \leq q^{d+1} \cdot \frac{e^{k^2/q}}{k!}\left(\frac{1}{e} + \frac{1}{(d-k)!}\right)$$

Lemma 3.5 *For all prime powers q and integers $k \leq d < q$,*

1.
$$|LR_q(k,d)| = |LR_q(0,d-k)| \cdot \binom{q}{k}$$

2.
$$|LR_q(k,d)| \leq q^{d+1} \cdot \frac{1}{k!}\left(\frac{1}{e} + \frac{1}{(d-k)!}\right)$$

Proof:

Let $p(x) \in LR_q(k,d)$ (i.e., $p(x)$ is a polynomial over F of degree at most d that has exactly k distinct roots in F and exactly k linear factors over F). Denote its k distinct roots by $\alpha_1, \ldots, \alpha_k$ (for every $1 \leq j \leq k, \alpha_j \in F$) and let

$$f(x) \stackrel{\text{def}}{=} \Pi_{i=1}^{k}(x - \alpha_i)$$

Obviously, $f(x)$ divides $p(x)$ and $p(x)$ has no other linear factors rather than the factors of $f(x)$. Hence, $p(x)$ is of the form $f(x) \cdot g(x)$, where $g(x)$ is a polynomial of degree at most $d-k$ with no roots. Clearly, $g(x) \in LR_q(0, d-k)$.

$f(x)$ is the product of k distinct linear factors, where each linear factor is identified with one of the q elements of F. Hence, the number of possibilities for $f(x)$ is the number of ways to choose k distinct roots out of the q elements of F, namely, $\binom{q}{k}$. Thus, we get,

$$|LR_q(k,d)| = |LR_q(0, d-k)| \cdot \binom{q}{k}$$

The second part is proved using lemma 3.1:

$$|LR_q(k,d)| = |LR_q(0,d-k)| \cdot \binom{q}{k} \leq q^{d-k+1}\left(\frac{1}{e} + \frac{1}{(d-k)!}\right) \cdot \frac{q^k}{k!} = q^{d+1} \cdot \frac{1}{k!}\left(\frac{1}{e} + \frac{1}{(d-k)!}\right)$$

■

Lemma 3.6 *For all prime powers q and integers $k \leq d < q$,*

1.

$$|R_q(k,d)| = \binom{q}{k} \cdot \sum_{i=0}^{d-k} |R_q(0, d-k-i)| \cdot \binom{i+k-1}{i}$$

2.

$$|R_q(k,d)| \leq q^{d+1} \cdot \frac{e+1}{k!}$$

Remark 3.7 *In coding theory, a generalization of part 1 is known (see [MS77, Chapter 11] and [Ber68, Chapter16]). A similar formula to the one in part 1, for counting monic polynomials with exactly k roots, is given in [KK90].*

Proof: Let $p(x) \in R_q(k,d)$ (i.e., $p(x)$ is a polynomial over F of degree at most d that has exactly k distinct roots in F) and let $f(x)$ be as in the proof of lemma 3.5. $f(x)$ divides $p(x)$ and therefore, $p(x)$ has at least k linear factors. Let i be such that $p(x)$ has exactly $k+i$ linear factors (obviously, $0 \leq i \leq d-k$). Let $g(x)$ be the product of the other i linear factors of $p(x)$, rather than the ones in $f(x)$.

Hence, $p(x)$ is of the form $f(x) \cdot g(x) \cdot h(x)$. As in lemma 3.5, the number of possibilities for $f(x)$ is $\binom{q}{k}$. $g(x)$ is a product of i linear factors, where each linear factor is one of the k distinct linear factors of $f(x)$ (recall that the same linear factor can appear more than once). Thus, the number of possibilities for $g(x)$ is $\binom{i+k-1}{i}$ (the number of ways to choose a multi-set of i elements out of a universe of size k). $h(x)$ is a polynomial of degree at most $d-k-i$ with no roots in F, therefore, $h(x) \in R_q(0, d-k-i)$. Summing over all possible values of i, we get

$$|R_q(k,d)| = \binom{q}{k} \cdot \sum_{i=0}^{d-k} |R_q(0, d-k-i)| \cdot \binom{i+k-1}{i}$$

$$\leq \frac{q^k}{k!}\left(1+\frac{1}{e}\right)\cdot\sum_{i=0}^{d-k} q^{d-k-i+1}\cdot\frac{(i+k-1)^i}{i!} \leq \frac{q^{d+1}}{k!}\left(1+\frac{1}{e}\right)\cdot\sum_{i=0}^{d-k} q^{-i}\cdot\frac{q^i}{i!}$$
$$= \frac{q^{d+1}}{k!}\left(1+\frac{1}{e}\right)\cdot\sum_{i=0}^{d-k}\frac{1}{i!} \leq q^{d+1}\cdot\frac{e+1}{k!}$$

Above, the first inequality is by lemma 3.1. i ranges between 0 and $d-k$ and therefore $i+k-1<d$. Together with the fact that $d<q$ we get the second inequality. ■

Comments:

- A slightly better bound can be obtained. By bounding $i+k-1 \leq d$ (without using the fact that $d \leq q$), we get (as above)

$$|R_q(k,d)| \leq \frac{q^{d+1}}{k!}\left(1+\frac{1}{e}\right)\cdot\sum_{i=0}^{d-k}\frac{(d/q)^i}{i!} \leq q^{d+1}\cdot\frac{e^{d/q}}{k!}\left(1+\frac{1}{e}\right)$$

- Another bound: By bounding

$$(i+k-1)^i \leq k^i\cdot\left[(i/k+1)^{k/i}\right]^{i^2/k} \leq k^i\cdot e^{i^2/k}$$

we get

$$|R_q(k,d)| \leq \frac{q^{d+1}}{k!}\left(1+\frac{1}{e}\right)\cdot\sum_{i=0}^{d-k}\frac{\left(\frac{k\cdot e^{i/k}}{q}\right)^i}{i!} \leq q^{d+1}\cdot\frac{e^{\left(k\cdot e^{(d-k)/k}/q\right)}}{k!}\left(1+\frac{1}{e}\right)$$

3.3 A Probabilistic Argument

In this section we prove a lemma which is a slightly different upper bound than the one given by lemma 3.6 (part 2). This bound will be used in our main result (section 4). This proof uses a probabilistic argument (in contrast to the algebro-combinatorial arguments in section 3.2). This argument may be interesting in its own right, and may be generalized to prove some bounds regarding multivariate polynomials, in which the combinatorial argument is much more complicated (see discussion below).

The idea is to view the number of roots of a random polynomial as a sum of random variables. We will use the fact that there is some independence between the random variables to prove that the probability that a polynomial has k roots vanishes exponentially in k. Our technical tool will be the following lemma:

Lemma 3.8 ([SSS95], Theorem 1) *Let $X_1,\ldots,X_q \in \{0,1\}$ be random variables, $X=\sum_{i=1}^q X_i$ and $\mu = E[X]$. For every $k>1$, if the X_i's are d-wise independent for*

$$d \geq h = h(q,\mu,k) \stackrel{\text{def}}{=} \left\lceil \frac{\mu(k-1)}{1-\mu/q} \right\rceil$$

then,

$$Pr(X \geq \mu k) \leq \frac{\binom{q}{h}(\mu/q)^h}{\binom{\mu k}{h}}$$

Intuitively, this lemma states that for proving that a probability vanishes exponentially (as in the Chernoff-Hoeffding bounds) it is enough to show limited independence (and not necessary full independence).

Lemma 3.9 *For all prime powers q and integers d, k,*

$$|R_q(k,d)| \le q^{d+1} \cdot \frac{1}{k!}$$

Proof: [9] Denote the q elements of F by $a_1, \ldots, a_q$. Let $p(x) \in F[x]$ be a random polynomial (i.e., a polynomial whose coefficients where chosen uniformly from F) of degree at most d. For each $a_i \in F$ define an indicator random variable

$$X_i = \begin{cases} 1 & \text{if } p(a_i) = 0 \\ 0 & \text{otherwise} \end{cases}$$

Obviously, for every i, $E[X_i] = \frac{1}{q}$. Let $X = \sum_{i=1}^{q} X_i$ and $\mu = E[X] = 1$. It is well known that polynomials of degree d generate $(d+1)$-wise independent samples (see, e.g., [N92] ch. 16), and therefore the X_i's are $(d+1)$-wise independent. We want to upper bound the probability that the polynomial has at least k roots. Since $k \le d < q$ we have

$$h = h(q,1,k) = \left\lceil \frac{k-1}{1-1/q} \right\rceil = \left\lceil k - 1 + \frac{k-1}{q-1} \right\rceil = k \le d.$$

Hence, we can apply lemma 3.8 and get

$$Pr(X \ge k) \le \frac{\binom{q}{k}(1/q)^k}{\binom{k}{k}} \le \frac{1}{k!}$$

This means that the fraction of polynomials with at least k roots is at most $\frac{1}{k!}$. Formally, [10]

$$|R_q(k,d)| \le q^{d+1} \cdot \frac{1}{k!}$$

which proves the lemma. [11] ■

A similar probabilistic argument can be used to prove an analogous result regarding multivariate polynomials. In this case, the algebro-combinatorial argument is much more complicated. Roughly speaking, this is because in multivariate polynomials, it is not true that a root implies a linear factor, as in the univariate case (this fact is the basis for the algebro-combinatorial arguments in the univariate case).

As above, the random variables are d-wise independent, but here d is the total degree of all the variables in the polynomial. Since the number of roots k, can be much more than d (it is bounded by dq^{n-1}, where n is the number of variables), we cannot use lemma 3.8. Instead, we use a different lemma, which requires less independence between the random variables:

[9] The proof assumes that $d, k < q$. For larger values of d and k the lemma is trivial since in that case a random polynomial is the same as a random function.

[10] The fact that lemma 3.8 works only for $k > 1$ does not bother us since this bound is trivial for $k = 0, 1$.

[11] We actually proved a slightly stronger result than the one stated in the lemma: we bounded the number of polynomials with *at least* k roots (and not *exactly* k roots).

Lemma 3.10 ([SSS95], Lemma 3) *Let $X_1, \ldots, X_{q'} \in \{0,1\}$ be random variables , $X = \sum_{i=1}^{q'} X_i$ and $\mu' = E[X]$. For every $k' > 1$, if the X_i's are d'-wise independent for $d' < h(q', \mu', k')$ then,*

$$Pr(X \geq \mu' k') \leq \frac{\binom{q'}{d'}(\mu'/q')^{d'}}{\binom{\mu' k'}{d'}}$$

Let $d' = d, q' = q^n, \mu' = q^{n-1}$, and $k' = \frac{k}{q^{n-1}}$. Note that the condition on d holds when $n \geq 2, k > q^{n-1}$ and $d \leq q$. This yields the following result: The fraction of multivariate polynomials over F of total degree at most d with $n \geq 2$ variables that have at least $k > 1$ roots is bounded by

$$\left(\frac{q^{n-1}}{k-d}\right)^d.$$

4 Main Result

In this section we prove our main result (theorem 1), which is an explicit construction of weak-designs with $d = l^2$ and $\rho = e^2$. The construction is essentially the same as the explicit construction of standard designs with weak parameters of [NW94] (stated in lemma 2.2). Our contribution is a finer analysis, based on the lemmas of section 3, which shows that the same construction actually yields a weak design with better parameters. Roughly speaking, each subset is identified with a polynomial over a fixed finite field. The subsets are constructed such that the intersection between two subsets corresponds to the values at which the two polynomials (identified with the two subsets) agree. The proof that these subsets induce a standard design is based on the fact that the number of values that two polynomials can agree on is bounded by their maximal degree. This fact suffices for proving the standard designs property, in which we have to bound the intersection between every two subsets.

However, a much more powerful statement can be said on the number of roots of a polynomial: not only that the number is bounded by its degree, but the average number of roots is 1 and, furthermore, the average of the exponentiation of the number of roots is constant (as formalized in lemmas 3.9 and 3.6). This resembles the definition of weak designs and explains, intuitively, why the construction yields a weak design.

Theorem 1 *Let $m,l \in \mathbb{N}$ and assume for simplicity that l is prime* [12] *and m is a power of l, then there exists an explicit weak (m, l, ρ, d)-design, with $d = l^2$ and $\rho = e^2$ (where e denotes the base of the natural logarithm).*

Proof: *The Construction:* Let $F = GF(l)$. Since $d = l^2$, we can identify, in the natural way, each number in the range $[d]$ with an ordered pair of elements in F. Thus, constructing the design is equivalent to constructing subsets of the set $F^2 = \{\langle a, b\rangle : a, b \in F\}$.

The subsets will be defined as follows: Given a polynomial $p(x) \in F[x]$, define the subset

$$S_p \stackrel{\text{def}}{=} \{\langle a, p(a)\rangle : a \in F\} \subseteq F^2$$

The weak design will consist of all such subsets which are indexed by polynomials over F of degree at most $\frac{\log m}{\log l} - 1$. Formally, let $deg \stackrel{\text{def}}{=} \frac{\log m}{\log l} - 1$ and let

$$P \stackrel{\text{def}}{=} \{p(x) \in F[x] : deg(p) \leq deg\}$$

[12] If l is not prime, pick, e.g., the smallest prime which is greater than l. This, by Bertrand's Postulate (see, e.g., [NZM91, page 367]), at most doubles the value of l.

The weak design S is defined as $S \stackrel{\text{def}}{=} \{S_p\}_{p\in P}$.

In the following claim we show that S is a weak (m, l, ρ, d)-design:

Claim 4.1 *S is a weak (m, l, ρ, d)-design.*

Proof: We have to show that the properties of weak designs hold:

- All sets S_p are subsets of F^2, which is identified with $[d]$.
- The size of each subset S_p is l.
- The number of subsets is the size of P (i.e., the number of polynomials over F with degree at most deg), namely,
$$|P| = l^{\,deg+1} = l^{\,(\frac{\log m}{\log l}-1)+1} = m$$
- For every $p \in P$, we have to show that [13]
$$\sum_{q\in P,\ q\neq p} 2^{|S_p\cap S_q|} \le \rho \cdot m$$

  The key idea is that an element $\langle a, b\rangle \in F^2$ is in the intersection of S_p and S_q if and only if $p(a) = q(a) = b$, i.e., p and q agree on a. Obviously, two polynomials agree on a value if and only if this value is a root of the difference polynomial. Hence, the size of the intersection of S_p and S_q is equal to the number of distinct roots of the polynomial $p - q$. Formally, $|S_p \cap S_q| = \sharp R(p - q)$.

  Therefore, ranging over all the polynomials q such that $q \in P,\ q \neq p$ and considering the size of $S_p \cap S_q$ (as needed to calculate the above sum) is equivalent to ranging over all the polynomials r (which represent the difference polynomial) such that $r \in P,\ r \neq \overline{0}$ and considering $\sharp R(r)$. Hence, we have reduced the problem of calculating the sizes of the intersections into the problem of counting how many polynomials are there with each possible number of distinct roots. This problem is solved in lemma 3.9.

  For every $p \in P$,
$$\sum_{q\in P, q\neq p} 2^{|S_p\cap S_q|} = \sum_{q\in P, q\neq p} 2^{\sharp R(p-q)} = \sum_{r\in P, r\neq\overline{0}} 2^{\sharp R(r)} \le \sum_{k=0}^{deg} 2^k \cdot |R_l(k, deg)|$$
$$\le \sum_{k=0}^{deg} 2^k \cdot (l^{deg+1} \cdot \frac{1}{k!}) = m \cdot \sum_{k=0}^{deg} \frac{2^k}{k!} \le e^2 \cdot m = \rho \cdot m$$

  Above, the first inequality is obtained by using the upper bound in lemma 3.9. Using the bound in the first comment after the proof of lemma 3.6 we can have $\rho = e^2 \cdot e^{d/q}\left(\frac{1}{e} + \frac{1}{d!}\right)$, which, in most cases, is a better bound. However, for the sake of simplicity, we preferred to state the theorem using the bound of lemma 3.9.

■

Proving the following claim completes the proof of the theorem:

[13] As noted above, we show how to construct a weak-design of the stronger version, in which for all $1 \le i \le m$, $\sum_{j\neq i} 2^{|S_i\cap S_j|} \le \rho m$, rather than $\sum_{j<i} 2^{|S_i\cap S_j|} \le \rho m$ (see definition 3 and discussion thereafter).

Claim 4.2 *The above construction is explicit.*

Proof: We have to describe an algorithm that given $1 \leq i \leq m$, outputs the set S_i in time $T = \text{poly}(\log m, l)$ and space $S = O(\log m)$. Enumerate all polynomials over F of degree at most deg by enumerating each one of the $deg + 1$ coefficients separately (of course any other standard enumeration will do as well). Given i, divide its $\log m$ bits into $\frac{\log m}{\log l} = deg + 1$ parts, each part consists of $\log l$ bits. The k^{th} part represents the k^{th} coefficient of the i^{th} polynomial. For every $a \in GF(l)$, we pair a with the evaluation of the polynomial on a and translate the pair to an element in the range $[d]$. The set S_i consists of all these values. Obviously, this algorithm runs in time $T = \text{poly}(\log m, l)$ and space $S = O(\log m)$. ■

Comment: Note that if $m = l^l$, then the set of all polynomials with degree bounded by deg is just the set of all functions. A simple calculation shows that in this case , we get a weak design with $\rho = e$. The construction in the proof of theorem 1 actually works only for cases in which $m \leq l^l$, since the degree of the polynomials must be smaller than the size of the field. [14] However, if we want more subsets, we can just repeat the same subsets. Say we need $m = c \cdot l^l$ subsets for some integer c, then for every $1 \leq i \leq m$

$$\sum_{j \neq i} 2^{|S_i \cap S_j|} \leq e \cdot l^{l-1} c$$

This shows that the theorem is true even for very large values of m. ■

Corollary 2 *Let $m,l \in \mathsf{N}$ such that $m > l^{\log l}$, then there exists an explicit weak $(m, O(l), \rho, d)$-design, with $d = O(l^2)$ and $\rho = O(1)$.*

Proof Sketch: If m is a power of l, then the corollary follows directly from theorem 1. Otherwise, $l^c < m < l^{c+1}$, for some integer c. Since $m > l^{\log l}$, there is a constant k such that $m \leq (kl)^c$. Taking the size of each subset in the weak-design to be kl, using polynomials with maximal degree $c + 1$, and taking the first m polynomials yields the required design. ■

5 The General Constructions

In this section we provide constructions of weak designs for more general cases. The constructions generalize the previous one (theorem 1) in the sense that ρ is not restricted to the value e^2. In order to obtain these weak designs we first (in section 5.1) develop some tools for manipulating weak designs. Then, in section 5.2, we combine these tools with the explicit construction of corollary 2 to get the desired constructions.

5.1 Tradeoff Between the Values of ρ and d

In the previous section we constructed an explicit weak (m, l, ρ, d)-design for specific values of d and ρ ($d = l^2$ and $\rho = e^2$). Theorem 3 will be stated for every $\rho > 1$ and with $d = O\left(\frac{l^2}{\ln \rho}\right)$. In this section we supply tools to decrease the value of ρ while increasing the value of d, and vice versa.

[14]This comment is irrelevant for weak designs that are used in the construction of extractors, since in that case, m (which is the number of extracted random bits) is $O(2^l)$ (l is the logarithm of the length of the source).

5.1.1 Decreasing the Value of ρ

The following lemma is a tool for decreasing the value of ρ, i.e., given an explicit weak-design with some value of ρ we show how to construct an explicit weak-design with a smaller value of ρ. This is done while maintaining the value of l. A smaller ρ means that the bound on the average of the exponentiations of the intersections between the subsets is smaller. To accomplish this goal, we will use a larger universe (i.e., increase the value of d).

The idea of the construction (inspired by the "packing" of weak designs in [RRV99]) is to take a union of weak-designs, each design on a different disjoint subset of the universe (this is possible since we are allowed to use a larger universe). Roughly speaking, the parameter ρ is getting smaller since in the "unioned" weak-design, two subsets that are not taken from the same "original" weak-design, are disjoint. Two subsets taken from the same "original" weak-design have relatively big intersection (since it is related to the original ρ, which is big). Nevertheless, there are few such pairs and therefore they don't contribute too much to the total sum. Overall, we get a small ρ, as desired.

Lemma 5.1 *If for $l, m, d \in \mathbb{N}$ and $\rho' > 1$ there exists an explicit weak (m, l, ρ', d)-design, then for every $1 < \rho \leq \rho'$ such that $n = \frac{\rho'-1}{\rho-1}$ is an integer, there exists an explicit weak (mn, l, ρ, dn)-design.*

Proof: We will prove this lemma for the stronger version of weak designs (see discussion after definition 3). The proof for weak designs (as defined in definition 3) is the same. We view $[dn]$ as the disjoint union of n blocks $B_1, \ldots, B_n$ each of size d. Our weak design $S_1, \ldots, S_{mn}$ will be defined as follows: For each $t \in [n]$ we let $S_{(t-1)m+1}, \ldots, S_{tm} \subset B_t$ be a copy of a weak (m, l, ρ', d)-design (as given in the lemma). In other words, we take a union of n weak (m, l, ρ', d)-designs, using disjoint subsets of the universe for each. In our new design, the size of each set is l, the number of sets is mn and the size of the universe is dn.

We have to show that for all $1 \leq i \leq mn, \sum_{j \neq i} 2^{|S_i \cap S_j|} \leq \rho \cdot mn$. For every $t \in [n]$, denote the collection of subsets $\{S_{(t-1)m+1}, \ldots, S_{tm}\}$ by $\widehat{S_t}$. Notice that if $S_i \in \widehat{S_t}$ then for every $S_j \notin \widehat{S_t}$, S_i and S_j are disjoint and

$$\sum_{j \in \widehat{S}_t, j \neq i} 2^{|S_i \cap S_j|} \leq \rho' \cdot m$$

since $\widehat{S_t}$ is a weak (m, l, ρ', d)-design. Thus, for every $t \in [n]$ and every $S_i \in \widehat{S_t}$, we have

$$\sum_{j \neq i} 2^{|S_i \cap S_j|} = \sum_{S_j \in \widehat{S}_t, j \neq i} 2^{|S_i \cap S_j|} + \sum_{S_j \notin \widehat{S}_t} 2^{|S_i \cap S_j|}$$

$$\leq \rho' \cdot m + (n-1)m = (\frac{\rho' - 1}{n} + 1) \cdot mn = \rho \cdot mn$$

The design remains explicit: Given the index i of a subset, we output the $i(\mathrm{mod} n)^{th}$ subset over the block $B_{\lceil i/n \rceil}$ (the subset can be constructed explicitly since the weak-design in each block is explicit).

■

5.1.2 Decreasing the Value of d

The following lemma is a tool for decreasing the value of d, while increasing the value of ρ. The idea of the construction is quite similar to the idea in lemma 5.1, although the opposite goal is achieved. The reason for this is that in the two constructions we indeed increase the value of d, however, in the following construction we increase simultaneously the values of d and l by the same factor. Hence, the value of d is decreased relative to l. It is achieved by taking direct products of many weak-designs.

For simplifying the following lemma, we will define a slightly stronger version of weak-designs: A *full weak* (m, l, ρ, d)*-design* is a weak-design (see definition 3), where the second condition is replaced by the following one: For all $1 \leq i \leq m$,

$$\sum_{j=1}^{m} 2^{|S_i \cap S_j|} \leq \rho \cdot m$$

Obviously, every full weak-design is in particular a weak-design. However, the resulting weak-design is useless for the construction of extractors. This is since in a full wek-design, we always have $\rho m \geq 2^l$, whereas in extractors we have $2^l > n > k = \rho m$.

Lemma 5.2 *If for $l, m, d \in \mathsf{N}$ and $\rho' > 1$ there exists an explicit full weak (m, l, ρ', d)-design, then for every $\rho \geq \rho'$ such that $n = \frac{\ln \rho}{\ln \rho'}$ is an integer, there exists an explicit full weak $(m^n, l \cdot n, \rho, d \cdot n)$-design.*

Proof: We view $[dn]$ as the disjoint union of n blocks $B_1, \ldots, B_n$ each of size d. For each $t \in [n]$ let $S^t = S_1^t, \ldots, S_m^t \subset B_t$ be an explicit full weak (m, l, ρ', d)-design . Our design, S, will be the direct product of all the n designs:

$$S \stackrel{\text{def}}{=} \bigotimes_{t=1}^{n} S^t.$$

In other words, each subset in S is a union of n subsets, each one from a different "small" design. We now show that S is an explicit full weak $(m^n, l \cdot n, \rho, d \cdot n)$-design. It is easy to verify that each subset is of size $l \cdot n$, the number of subsets is m^n and the size of the universe is dn. To complete the proof we have to show that for all $1 \leq i \leq m^n$, $\sum_{j=1}^{m^n} 2^{|S_i \cap S_j|} \leq \rho \cdot m^n$.

We will index each subset in S by an n-tuple, $(i_1, \ldots, i_n) \in \{1, \ldots, m\}^n$, such that $S_{(i_1, \ldots, i_n)} = \bigcup_{t=1}^{n} S_{i_t}^t$. For all $i = (i_1, \ldots, i_n)$,

$$\sum_{j=(j_1,\ldots,j_n)} 2^{|S_i \cap S_j|} = \sum_{j_1} \cdots \sum_{j_n} 2^{\sum_{k=1}^{n} |S_{i_k} \cap S_{j_k}|}$$

$$= \sum_{j_1} \cdots \sum_{j_n} \prod_{k=1}^{n} 2^{|S_{i_k} \cap S_{j_k}|} = \prod_{k=1}^{n} \sum_{j_k} 2^{|S_{i_k} \cap S_{j_k}|} \leq (\rho' m)^n = \rho \cdot m^n$$

Above, the inequalities are obtained by the fact that each coordinate is a full weak (m, l, ρ', d)-design.

The design remains explicit: The algorithm gets an index i to a subset. First we find the n-tuple $(i_1, \ldots, i_n)$ which i represents. The j^{th} element is from the $k \stackrel{\text{def}}{=} \lceil j/n \rceil$ design, specifically, it is the $j(\mathrm{mod} n)^{th}$ element of the subset $S_{i_k}^k$. Our subset contains all such elements for $1 \leq j \leq l \cdot n$. Since n is smaller than the logarithm of the number of subsets (m^n), the construction remains explicit. ■

5.2 The Constructions

We now have the tools for proving our general constructions:

Theorem 3

1. *For every* m,$l \in \mathsf{N}$, *such that* $m = \Omega(l^{\log l})$, *and constant* $\rho > 1$ *there exists an explicit weak* (m, l, ρ, d)*-design, where* $d = O(l^2)$.
2. *For every* m,$l \in \mathsf{N}$ *and* $\rho > 1$, *such that for some constant* $\alpha \geq 1$, $2^l < \alpha m$, *there exists an explicit weak* (m, l, ρ, d)*-design, where* $d = O(\frac{l^2}{\ln \rho})$.

Remark 5.3 *Note that part 2 of the theorem does not subsume lemma 2.3 (because of the constraint* $2^l < \alpha m$*). Furthermore, it is useless for the construction of extractors, since there* $2^l > n > \rho m$.

Proof: We prove each part separately:

1. Let c be the constant hidden in corollary 2. If ρ is a constant bigger than c, this part follows trivially from corollary 2, since a weak (m, l, c, l^2)-design is also a weak (m, l, ρ, l^2)-design for $\rho \geq c$. Otherwise, given $l, m \in \mathsf{N}$ and $\rho < c$, we let $n = \frac{c-1}{\rho-1} = O(1)$. By applying lemma 5.1 on an explicit weak $(\frac{m}{n}, l, c, l^2)$-design (guaranteed by corollary 2) we get the desired explicit weak $(m, l, \rho, O(l^2))$-design.
2. Given $l, m \in \mathsf{N}$ and $\rho > e^2$, let $n = \frac{\ln \rho}{\ln(e^2+\alpha)} = O(\ln \rho)$ (we assume for simplicity that n, $m^{\frac{1}{n}}$ and $\frac{l}{n}$ are integers). We start with an explicit full $\left(m^{1/n}, \frac{l}{n}, e^2 + \alpha, O((\frac{l}{n})^2)\right)$-design (whose existence will be proved later on). By applying lemma 5.2 on this design we get the desired explicit (full) weak $\left(m, l, \rho, O(\frac{l^2}{\ln \rho})\right)$-design.

   It remains to prove the existence of an explicit full weak $\left(m^{1/n}, \frac{l}{n}, e^2 + \alpha, O((\frac{l}{n})^2)\right)$-design used above. Since $2^{l/n} \leq \alpha^{1/n} m^{1/n} \leq \alpha m^{1/n}$, the above design is guaranteed by the following simple claim:

   Claim 5.4 *Let* m',$l' \in \mathsf{N}$ *(assume for simplicity that* l' *is prime and* m' *is a power of* l'*) such that for some constant* α, $2^{l'} < \alpha m'$, *then for* $\rho' = e^2 + \alpha$ *there exists an explicit full weak* (m', l', ρ', d')*-design where* $d' = l'^2$.

   Proof: The construction is the same as in the proof of theorem 1. For all $1 \leq i \leq m'$,

   $$\sum_j 2^{|S_i \cap S_j|} = \sum_{j \neq i} 2^{|S_i \cap S_j|} + 2^{|S_i \cap S_i|} \leq e^2 m' + 2^{l'} \leq (e^2 + \alpha) m' = \rho' m'$$

   ■

■

Theorem 4 *For every* $l, m \in \mathsf{N}$, *such that* $m > l^{\log l}$, *and* $0 < \gamma < 1/2$, *there exists an explicit weak* $(m, l, 1 + \gamma, d)$*-design with*

$$d = O\left(l^2 \cdot \log \frac{1}{\gamma}\right).$$

Proof: The proof follows the construction of weak-designs via "packing" in [RRV99]. The only difference is that we use explicit weak $(m, l, 2, O(l^2))$-designs, rather than non-explicit ones. This makes the construction explicit, as desired. For the sake of completeness, we restate the construction.

Let $d_0 = O(l^2)$, $h = \lceil \log(4/\gamma) \rceil$ and $d = h \cdot d_0 = O(l^2 \cdot \log(1/\gamma))$. We view $[d]$ as the disjoint union of h blocks $B_0, \ldots, B_{h-1}$ each of size d_0. For each $t \in \{0, \ldots, h-1\}$, let $n_t = \lfloor (1 - 2^{-t}) \cdot (1+\gamma) \cdot m \rfloor$ and $m_t = n_{t+1} - n_t$. Note that $n_h \geq m$.

We now define our weak design $S_1, \ldots, S_m$. For each $t \in \{0, \ldots, h-1\}$, let $S_{n_t+1}, \ldots, S_{n_t+m_t} \subset B_t$ be an explicit weak $(m_t, l, 2, d_0)$-design as guaranteed by theorem 3. In other words, we take the ordered union of h weak $(m_t, l, 2, d_0)$-designs using disjoint subsets of the universe for each. The number of sets is $n_h \geq m$, the size of the universe is d, and each set is of size l, so we only need to check that for all $i \in [m]$, $\sum_{j<i} 2^{|S_i \cap S_j|} < (1+\gamma) \cdot m$. For $i \in \{n_t + 1, \ldots, n_t + m_t\}$, S_i is disjoint from any S_j for any $j \leq n_t$ and

$$\sum_{j=n_t+1}^{i-1} 2^{|S_i \cap S_j|} \leq 2 \cdot m_t$$

since $S_{n_t+1}, \ldots, S_{n_t+m_t}$ is a weak $(m_t, l, 2, d_0)$-design. Thus, we have

$$\begin{aligned}
\sum_{j<i} 2^{|S_i \cap S_j|} &= \sum_{j=1}^{n_t} 2^{|S_i \cap S_j|} + \sum_{j=n_t+1}^{i-1} 2^{|S_i \cap S_j|} \\
&\leq n_t + 2 \cdot m_t \\
&= 2 \cdot n_{t+1} - n_t \\
&\leq 2 \cdot (1 - 2^{-t-1}) \cdot (1+\gamma) \cdot m - (1 - 2^{-t}) \cdot (1+\gamma) \cdot m \\
&\leq (1+\gamma) \cdot m,
\end{aligned}$$

as desired.

We now describe an algorithm that constructs explicitly the design: Given an index i of a subset, we find the value of $t \in [h]$ (e.g., by binary search) such that $n_t < i \leq n_{t+1}$. Our subset S_i is a subset of B_t indexed by $i - n_t$ in the t^{th} design, and it can be computed by the algorithm of claim 4.2. Find $t \in [h]$ takes time $T = O(\log h) = O(\log m)$. Therefore, the algorithm runs in time poly$(\log m)$ and space $O(\log m)$ and the construction remains explicit. ■

5.3 Constructing Explicit Standard Designs

As discussed in subsection 2.2.2, the explicit standard designs given in lemma 2.2 has weaker parameters than the non-explicit ones given in lemma 2.1. In particular, the non-explicit construction can yield a design with large ρ and d smaller than $O(l^2)$. Combining the idea of our "direct product" (subsection 5.1.2) with error correcting codes, Salil Vadhan [Vad01] has constructed explicit standard designs with better tradeoff between d and ρ.

Lemma 5.5 *If for $l, m, m', d, n \in \mathbb{N}$ and $\rho > 1$ there exists:*

- *An explicit standard (m, l, ρ, d)-design, and*
- *An explicit error correcting code over an alphabet of size m, with m' codewords of length n, such that every two codewords agree in at most k positions,*

then there exists an explicit standard (m', nl, ρ', nd)-design, where $\rho' = 2^{kl} \cdot \rho^{n-k}$.

Proof: For each codeword $c = c_1, \ldots, c_n \in [m]^n$, consider the binary string of length nd obtained by replacing each c_i $(1 \le i \le m)$ with the characteristic vector (of length d) of S_{c_i} (the $c_i{}^{th}$ subset of the original design). This string will be the characteristic vector of the corresponding subset of the new design. Clearly, the number of subsets is m', each subset is of size nl and the size of the universe is nd. The value of ρ' can be verified by considering both the effect of the original standard design and the error correcting code on the intersection between every two subsets in the new design. ■

By taking the explicit standard design of lemma 2.2 and Reed-Solomon codes (see, e.g, [MS77]), the above lemma yields explicit standard designs with $d = O(l)$, as long as ρ is at least a constant power of m and 2^l. [15]

6 Applications

6.1 Logspace Extractors

An important application of the above explicit constructions of weak designs is several explicit constructions of extractors that are similar to the ones in [RRV99] in terms of parameters and that can be evaluated in logarithmic space. The first logspace extractor extracts a constant fraction of the source min entropy with only $O(\log^2 n)$ additional truly random bits. The second one extracts all the min entropy with $O(\log^3 n)$ random bits.

Theorem 5 *For every n, k, m and ϵ, such that $m \le k \le n$ and $m = \Omega((\log n)^{\log\log n})$, there are explicit logspace (k, ϵ)-extractors $EXT : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with*

1. *if k/m is constant, $d = O\left(\log^2 n \cdot \log(1/\epsilon)\right)$.*
2. *$d = O\left(\log^2 n \cdot \log(1/\gamma) \cdot \log(1/\epsilon)\right)$, where $1 + \gamma = k/(m-1)$ and $1/m \le \gamma < 1/2$.*

Proof Sketch: The main idea is to use our explicit weak designs in the RRV extractors. We first give a rough description of the RRV extractor (the Trevisan extractor is the same, except that standard designs are used instead of weak designs)

$$EXT\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m.$$

For a string $u \in \{0,1\}^n$, let $\overline{u} \in \{0,1\}^{\overline{n}}$ be an encoding of u in a logspace-computable error-correcting code and define $l = \log \overline{n}$. We view $\overline{u}$ as a Boolean function $\overline{u}\colon \{0,1\}^\ell \to \{0,1\}$. Let $S = (S_1, \ldots, S_m)$ be a weak (m, l, ρ, d)-design. For a string $y \in \{0,1\}^d$ and a subset S_i, define $y|_{S_i}$ to be the string in $\{0,1\}^l$ obtained by projecting y onto the coordinates specified by S_i. Then the extractor is simply

$$EXT_S(u, y) = \overline{u}(y|_{S_1}) \cdots \overline{u}(y|_{S_m}).$$

To get a linear dependence on $\log(1/\epsilon)$, we view the bits of the input as blocks of size $\log|F|$, for some finite field F, such that each block is identified with an element in F. The error-correcting code is taken to be a multi-linear function over the field F (which can be computed in logarithmic space). A pairwise independent hash function is finally applied on each output block. All of these operations can be done in logarithmic space. The other main step is constructing a weak design. For the first extractor we use the explicit weak designs of theorem 3, whereas the second uses those

[15] This subsection was suggested to us by Salil Vadhan [Vad01].

of theorem 4. To calculate the i^{th} bit in the output of the extractor we must find the projection of the subset S_i on the codeword computed in the first step. Using our explicit weak designs, we can do this in logarithmic space. Overall, our extractor can be evaluated in logarithmic space, as desired. ■

Remark 6.1 [16] A logspace extractor with similar parameters can also be obtained by another method. In this extractor, the subsets of the weak design are chosen pairwise independently from all the potential subsets. The logarithmic randomness needed is taken from the $O(d)$ truly random bits of the extractor. This is not an alternative way of constructing explicit weak designs (since the construction is probabilistic and yields a weak design only with high probability). One of the referees suggested to derandomize the above construction by checking for each possible seed if it defines a weak design for a given ρ and take the (say) lexicographically smallest seed. This gives a deterministic construction of weak designs, that can be done in space logarithmic in the number of subsets. However, it is not explicit in the sense of definition 4 since the i^{th} set cannot be constructed in logarithmic time.

6.2 Derandomization of Space Bounded Computation

Let M be a randomized logarithmic space Turing machine (or a polynomial width branching program) and assume that some weak estimation of the state probabilities is known. In [RR99], there is a construction of a logarithmic space pseudo-random generator that uses only logarithmic number of truly random bits and outputs a sequence of k bits that look random to M. The construction is based on linear space extractors which extract all the min-entropy of the source. By using the logspace extractor of theorem 5 (which is in particular linear space) we can slightly improve k (the length of the output of the pseudo-random generator). Details follow.

By assuming there are explicit linear space optimal extractors (i.e., linear space extractors which use only $d = O(\log n + \log 1/\epsilon)$ truly random bits), the construction yields a pseudo-random generator with $k = 2^{\sqrt{\log n}}$. Using the linear space extractors of Ta-Shma [NT99], which uses $(\log n)^9$ truly random bits, [17] it was possible to prove an unconditional result with $k = 2^{\sqrt{\log n}/(\log\log n)^9}$. The RRV-extractor [RRV99] uses only $(\log n)^3$ truly random bits, but is not a linear space extractor and therefore cannot be used in the construction. However, by using the logspace extractor of theorem 5, we can improve the degree of the polylog up to $k = 2^{\sqrt{\log n}/(\log\log n)^3}$. The above is formalized in the following corollary (see definitions of *layered branching program* and *estimator* in [RR99]):

Corollary 6 *Let $r = \sqrt{\log n}/(\log\log n)^3$. Let M be an $(n, 2^r)$-layered branching program, and let A be a $(2^{-3r-1}, r)$-estimator for M then there exists an RL Turing machine T such that:*

1. *The inputs for T are M and A.*
2. *T uses $O(\log n)$ random bits (and $O(\log n)$ space).*
3. *The output of T is a 2^{-r}-error pseudo-random sequence for M.*

[16] Suggested by Oded Goldreich, Salil Vadhan and Avi Wigderson.

[17] In [NT99] the number of truly random bits used by the extractors given as $d =$polylog n, without specifying the degree of the polynomial. However, Ta-Shma estimates the degree of this polynomial to be 9 (see [RRV99]).

6.3 Pseudorandom Generators for Low Complexity Classes

One of the most interesting settings of parameters in the Nisan-Wigderson pseudorandom generator [NW94] is when one starts with a function $f : \{0,1\}^l \to \{0,1\}$ of exponential hardness $s = 2^{\Omega(l)}$ and use it to construct a pseudorandom generator $G : \{0,1\}^d \to \{0,1\}^m$ with $d = O(l)$ and $m = s^{\Omega(1)}$. Using our results, Salil Vadhan [Vad01] gives a way to achieve the same result with the explicit standard design of subsection 5.3. This may be interesting in its own right, and it may also have some applications when using NW-like pseudorandom generators to derandomize very low complexity classes, especially when using strong notions of uniformity for the circuit classes (rather than just poly-time uniformity).

7 Final Discussion and Open Problems

One contribution of this work is the realization and the applicability of the fact that the number of roots of a polynomial is not only bounded by its degree (a fact used in a wide variety of applications), but, typically, is much smaller. Specifically, the number of roots is exponentially concentrated around its average, which is 1. In this work, we used this fact to construct explicit weak-designs. This fact may be interesting in other applications as well.

A possible alternative to our construction in section 5.1.2 of weak-designs with d smaller than $O(l^2)$ and big ρ, is by considering multivariate polynomials, analogously to the univariate polynomials used for the construction in the proof of theorem 1. For every multivariate polynomial with n variables $p(x_1, \ldots, x_n)$ we define the subset

$$S_p \stackrel{\text{def}}{=} \{\langle a_1, a_2, \ldots, a_n, p(a_1, \ldots, a_n)\rangle : \forall 1 \leq i \leq n \; a_i \in F\} \subseteq F^{n+1}$$

As in the univariate case, the quality of the design is related to the number of multivariate polynomials over F of total degree at most d with n variables that have at least k roots. The upper bound (presented in section 3.3) does not suffice for proving the required weak design property. Proving a better bound may obtain a good weak design.

Acknowledgements

We would like to thank Omer Reingold and Ronen Shaltiel for some fruitful discussions. We also gratefully thank Salil Vadhan and the anonymous referees for their invaluable comments. We thank Igor Shparlinski for referring us to the three papers by the Knopfmachers.

References

[AS00] N. Alon and J. H. Spencer. *The Probabilistic Method.* Interscience Series in Discrete Mathematics and Optimization. John Wiley and Sons, Inc., 2nd edition 2000.

[Ber68] E. R. Berlekamp. *Algebraic Coding Theory.* McGraw-Hill Series in Systems Science. 2000.

[ByGW99] Z. Bar-Yossef, O. Goldreich and A. Wigderson. Deterministic Amplification of Space-Bounded Probabilistic Algorithms. *Proceedings of the 14th Annual IEEE Conference of Computational Complexity (CCC)*, pages 188-198, 1999.

[BRST02] Z. Bar-Yossef, O. Reingold, R. Shaltiel and L. Trevisan. Streaming Computation of Combinatorial Objects. *Proceedings of the 17th Annual IEEE Conference of Computational Complexity (CCC)*, pages 165-174, 2002.

[EFF85] P. Erdős, P. Frankl and Z. Füredi. Families of finite sets in which no set is covered by the union of r others. *Israel Journal of Mathematics*, 51(1-2):79-89, 1985.

[HR00a] T. Hartman and R. Raz. On the Distribution of the Number of Roots of Polynomials and Explicit Logspace Extractors. *Proceedings of RANDOM 2000*, pages 3-22, number 8 in Proceedings in Informatics. Carleton Scientific, 14 July 2000.

[HR00b] T. Hartman and R. Raz. On the Distribution of the Number of Roots of Polynomials and Explicit Logspace Extractors. *Electronic Colloquium on Computational Complexity* Technical Report TR00-044, June 2000. `http://www.eccc.uni-trier.de/eccc`.

[K99] A. Knopfmacher. On the degrees of irreducible factors of polynomials over a finite field. *Discrete Mathematics* **196**, 1999, pp. 197-206.

[KK90] A. Knopfmacher and J. Knopfmacher. Counting polynomials with a given number of zeros in a finite field. *Linear and Multilinear Algebra* **26**, 1990, pp. 287-292.

[KK93] A. Knopfmacher and J. Knopfmacher. Counting irreducible factors of polynomials over a finite field. *Discrete Mathematics* **112**, 1993, pp. 103-118.

[Lan68] S. Lang. *Algebraic Structures.* Addison-Wesely, 1968.

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995.

[MS77] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes.* North-Holland, 1977.

[N92] Joseph (Seffi) Naor. Probabilistic Methods in Computer Science. Lecture notes, Technion, Spring 1992. Available at `http://www.uni-paderborn.de/fachbereich/AG/agmadh/WWW/english/scripts.html`

[NT99] N. Nisan and A. Ta-Shma. Extracting randomness: A survey and new constructions. *Journal of Computer and System Sciences*, 58(1):148–173, February 1999.

[NW94] N. Nisan and A. Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49(2):149–167, October 1994.

[NZ96] N. Nisan and D. Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, February 1996.

[NZM91] I. Niven, H. S. Zuckerman and H. L. Montgomery. *An Introduction to the Theory of Numbers.* John Wiley and Sons, Inc., 5th edition, 1991.

[RR99] R. Raz and O. Reingold. On recycling the randomness of the states in space bounded computation. *Proceedings of the 31st ACM Symposium on Theory of Computing*, Atlanta GA, May 1999.

[RRV99] R. Raz, O. Reingold and S. Vadhan. Extracting all the randomness and reducing the error in Trevisan's extractors. *Proceedings of the 31st ACM Symposium on Theory of Computing*, Atlanta GA, May 1999. To appear in *JCSS* Special Issue on *STOC 99*.

[S02] R. Shaltiel. Recent Developments in Explicit Constructions of Extractors. *Bulletin of the European Association for Theoretical Computer Science, Computational Complexity Column*, 77, June 2002.

[SSS95] J. P. Schmidt, A. Siegel and A. Srinivasan. Chernoff-Hoeffding bounds for applications with limited independence. *SIAM Journal on Discrete Mathematics*, 8:223-250, 1995.

[Sud98] M. Sudan. Algebra and Computation. Lecture notes, MIT, Fall 98. Available at `http://theory.lcs.mit.edu/~madhu/FT98/course.html`

[Tre01] L. Trevisan. Extractors and Pseudorandom Generators. *Journal of the ACM*, 48(4):860-879, 2001.

[Vad01] S. Vadhan. Personal communication.

[Vad02] S. Vadhan. Problem set 2 in *Pseudorandomness* Course, Harvard University, Fall 2002. Available at `http://www.courses.fas.harvard.edu/~cs255`

[WZ99] A. Wigderson and D. Zuckerman. Expanders that beat the eigenvalue bound: Explicit construction and applications. *Combinatorica*, 19(1):125–138, 1999.

[Zuc97] David Zuckerman. Randomness-optimal oblivious sampling. *Random Structures & Algorithms*, 11(4):345–367, 1997.