New Pseudo-Random Generators for Weak Random Sources

David Zuckerman *
Division of Computer Science
University of California
Berkeley, CA 94720

Abstract

We consider the following model for a weak random source: the source is asked only once for R bits, and the source outputs an R-bit string such that no string has probability more than $2^{-\delta R}$ of being output, for some fixed $\delta > 0$. We show that under the Generalized Paley Graph Conjecture, there is a pseudo-random generator that simulates RP using as a seed a string from such a source for any $\delta > 0$. For $\delta > 1/2$, we can simplify our generator considerably and prove its correctness without relying on any unproven assumption. Cohen and Wigderson [CW] have also solved the case $\delta > 1/2$ using different techniques. Finally, we prove that for any $\delta > 0$ and for all but an exponential fraction of δ-sources, an even simpler generator can simulate RP.

1 Introduction

Randomness plays a vital role in almost all areas of computer science, both in theory and in practice. Randomized algorithms are often faster or simpler than the deterministic algorithms for the same problem (see e.g. [Rab]).

To produce "random" bits, a computer might consult a physical source of randomness, such as a Zener diode, or use the last digits of a real time clock. In either case, it is not clear how random these "random" bits will be. It is therefore of interest to see if weak, or imperfect, sources of randomness can be used in randomized algorithms.

Blum [Blu] initiated the study of weak random sources and gave algorithms to convert the output of a Markovian source into truly random sequences. Then Santha and Vazirani [SV] introduced the model of δ semi-random sources, where the probability of a given bit being a specific value, conditional on the value of previous bits, is not too large. In this model, they proved that it is impossible to extract even a single random bit (so they had to use several independent such sources).

Yet this does not imply that such sources cannot be used to simulate RP or BPP algorithms. Indeed, [VV] and [Vaz] exhibit pseudo-random generators (prg's) which simulate RP and BPP with one δ semi-random source. Chor and Goldreich [CG] generalized this by presenting a prg that simulates BPP even if no sequence of $O(\log R)$ bits has too high a probability of being a particular sequence, (here R denotes the total number of random bits used).

Various authors have also considered models where an adversary chooses the values of certain bits, but the others are random (see [CGH*], [BL], [LLS], [CW]).

Our sources, called δ-sources, generalize all of these models, making no structural assumptions about dependencies. Namely, a δ-source is asked only once for R bits, and the source outputs an

*The author was supported by an NSF Graduate Fellowship.

R-bit string such that no string has probability more than $2^{-\delta R}$ of being output, for some fixed $\delta > 0$. Compare this with the Chor-Goldreich model, where the source is asked R/l times for l-bit strings, and no l-bit string has probability more than $2^{-\delta l}$ of being output; their simulation only works for $l = O(\log R)$.

In some sense, the idea of imposing an upper bound on the probability of any string is the most general model worth considering. If we tried instead the more general notion of imposing a lower bound on the entropy, we could never get an exponential fall-off in the error probability of RP algorithms. This is because a given "bad" string could occur with constant probability. The smallest upper bound that could possibly suffice is 2^{-R^δ}.

Our main result is that under the Generalized Paley Graph Conjecture, there is an efficient simulation of RP for all $\delta > 0$. Our algorithm exploits the "independence" of the additive and multiplicative groups modulo a prime. For example, our proof relies on the fact that if A and B are large enough sets, then the discrete logs of $A + B$ are well distributed in a certain technical sense. The Generalized Paley Graph Conjecture essentially asserts that the above is true for smaller A and B than can be proven.

For $\delta > 1/2$, we can simplify our generator considerably and prove its correctness without relying on any unproven assumption. Indeed, the previously mentioned algorithm can be regarded as a bootstrap of this algorithm. Moreover, a lemma of Mansour, Nisan, and Tiwari [MNT] implies that this prg can be based on any family of universal hash functions.

Cohen and Wigderson [CW] also have a prg that simulates RP with δ-sources when $\delta > 1/2$ and BPP when $\delta > 3/4$. Yet their algorithms are based on random walks on expanders, and as such can do no better than $\delta = 1/2$, if the degree is small enough. This is because a δ-source with $\delta = 1/2$ can ensure that a random walk never escapes the neighborhood of a single vertex. The δ-source simply has every other step in the random walk be purely random, while the next step is a function of the previous step and the initial vertex, which sends the random walk back over the edge it just came.

We also present an even simpler prg that simulates RP for all but an exponential fraction of δ-sources, for any $\delta > 0$.

We must mention that except for this last, simplest prg, we work modulo a prime p. This gives rise to the technicality that we have to compute such a prime; if we do not have a good source of randomness, this is difficult. Cohen and Wigderson [CW] avoid this problem. Yet this prime only has to be computed once for a given size of the random tape. Using methods in [PSS], one can even expect to generate an r-bit prime with a certificate. Our basic algorithm needs only one prime; however, our bootstrapped algorithm uses a constant number of primes, where the constant depends on δ.

Finally, we remark that there is a close link between prg's for δ-sources and prg's used to amplify the success probability of RP algorithms. Namely, in both cases one needs the fact that few R-bit strings are "bad" for the prg. More specifically, suppose we have an RP algorithm which needs r random bits to achieve an error probability of a constant. Then our prg (the one used for $\delta > 1/2$) needs $O(r \log r)$ bits to reduce the error to 2^{-r} (in fact $2^{-r \log r}$). In [CW] and [IZ] it is shown how only $O(r)$ bits are needed to reduce the error to 2^{-r}, even for BPP.

2 Preliminaries

RP is the set of languages $L \subseteq \{0,1\}^*$ such that there is a deterministic polynomial time Turing machine $M_L(a,x)$ for which

$$a \in L \Rightarrow \Pr[M_L(a,x) = 1] \geq 1/2 \tag{1}$$

$$a \notin L \Rightarrow \Pr[M_L(a,x) = 1] = 0$$

where the probabilities are for an x picked uniformly in $\{0,1\}^{p(|a|)}$ for some polynomial p. If $M(a,x) = 1$, we say M accepts a using random tape x. If we run M_L on independent random tapes, we can change the probability in (1) to $1 - c$ for any constant $c > 0$. In our analysis it will help to take $c = 1/128$; let r be the size of the random tape needed to achieve this c.

We wish to simulate RP with a weak random source; say we wish to test whether a given element a is in L. Suppose $a \in L$; then we wish to find with high probability a witness to this fact. Let W be the set of witnesses, i.e. $W = \{x | M_L(a,x) = 1\}$, and N be the set of non-witnesses, i.e. the complement of W.

A pseudo-random generator (prg) asks the source for $R = poly(r)$ bits, and constructs some r-bit strings, one of which hopefully will lie in W with high probability. We first observe, as in [CG] (although in our case it is much easier):

Observation: The worst case δ-source is a flat source, i.e. one which places probability $2^{-\delta R}$ on $2^{\delta R}$ R-bit strings. To see this, fix a prg. Note that some R-bit strings produce witnesses, and others do not. The source may as well put as much probability as it can on the strings that do not. Throughout the rest of this paper, we assume all our sources are flat.

Our prg's divide the R-bit string into r-bit strings and combine these strings in various ways. We therefore will need the following lemma, which basically says that with high probability, enough of the r-bit strings have small probability of occurring:

Lemma 1 *Fix $\alpha > 0$, $\delta' < \delta$, and an integer $k > 0$. Let k' be an integer $\geq \frac{k(1-\delta')+\alpha}{\delta-\delta'-r^{-1}} = O(k)$, and set $R = k'r$. View the R-bit string X given by the source as k' r-bit strings $x_1, x_2, \ldots, x_{k'}$, and define X_i as the initial string $x_1, x_2, \ldots, x_i$. For Y an R-bit string, define y_i and Y_i similarly, and let*

$$p_i(Y) = Pr[x_i = y_i | X_{i-1} = Y_{i-1}],$$

where X is the random string output by the source. Then

$$Pr[\text{for} \geq k \text{ values of } i,\ p_i(X) \leq 2^{-\delta' r}] \geq 1 - 2^{-\alpha r},$$

where again X is the random string output by the source.

Proof. Construct a tree corresponding to the outputs X of the source as follows: let the nodes be all possible initial sequences X_i for each i, $0 \leq i \leq k'$, and let the parent of X_i be X_{i-1}. Define $p_i(X_i) = p_i(X)$ for any continuation X of X_i, and define an edge (X_{i-1}, X_i) to be "good" if $p_i(X_i) \leq 2^{-\delta' r}$, and "bad" otherwise. We wish to show that few of the $2^{\delta R}$ leaves have root-leaf paths with less than k good edges.

To bound this number, first note that each parent has at most $2^{\delta' r}$ children connected by bad edges, and at most 2^r children. Thus, the total number of root-leaf paths with $k' - k$ specified bad edges (e.g. the edges at distances 2,3,6,7 from the root must be bad) is at most

$$2^{kr} 2^{(k'-k)\delta' r},$$

so the total number of root-leaf paths with at least $k' - k$ bad edges is

$$\binom{k'}{k} 2^{kr} 2^{(k'-k)\delta' r}.$$

Using $\binom{k'}{k} \leq 2^{k'}$ and substituting the definition of k' in the above formula, we bound the number by $2^{-\alpha r} 2^{\delta R}$, as required. ■

The algorithms we consider have the property that a "bad" r-bit string from the source will not hurt it, i.e. a string x_i that comes from an X with $p_i(X)$ large. Therefore, the above result implies that we can view our source as giving us r-bit strings such that the conditional probability of the string given previous strings is at most $2^{-\delta r}$, where we've redefined δ as the δ' we get from the lemma above.

3 First PRG

We now present a prg which, for any $\delta > 0$, will simulate RP for almost all δ-sources. Our result is really only interesting because we consider arbitrary witness sets, so there is a doubly exponential number of them. To simulate RP, there are only an exponential number of witness sets, and a simple counting argument shows that for large enough k' it suffices to use $x_1, \ldots, x_{k'}$ as the pseudo-random strings.

This prg is a simple modification of the prg used in [VV] to simulate RP with δ semi-random sources: the only change is that we work modulo 2^r instead of the vector space of dimension r over F_2.

We view our r-bit strings as integers modulo $q = 2^r$. Our prg asks the source for $k'r$ bits and forms the r-bit numbers $x_1, x_2, \ldots, x_{k'}$. For each $I \subseteq \{1, 2, \ldots, k'\}$ it then forms the sum $a_I = \sum_{i \in I} x_i$, and runs the RP algorithm on a_I. Since it will suffice to take $k' = O(\log r)$, this prg takes polynomial time. Lemma 1 implies that for the purposes of analysis, we may assume we use k "good" strings $t_1, t_2, \ldots, t_k$. The worst case (at least as far as analysis goes) is when t_i is chosen randomly from a set T_i of size $2^{\delta' r}$, where T_i depends on $t_1, \ldots, t_{i-1}$. The reason this is the worst case is again that the t_i can be ordered according to how good they are for our algorithm, so the source may as well place as much probability as possible on the worst t_i.

It is not hard to find RP algorithms and δ-sources for which the above prg does not work. An example is the RP algorithm with witness set of all numbers $\geq 2^{r-1}$, and a source which only outputs numbers $\leq 2^{\delta r}$. Nevertheless, we show that all but an exponentially small fraction of sources are good for *all* witness sets.

To analyze the above algorithm, we follow [VV] and define the effective set of non-witnesses S_i w.r.t. $t_1, \ldots t_i$ as follows: S_0 is the set of non-witnesses N, and

$$S_i = \{s \in Z_q | (\forall I \subseteq \{1, \ldots, i\})\, s + \sum_{i \in I} t_i \in S\}$$

Then $S_i = S_{i-1} \cap (S_{i-1} - t_i)$, where A op b denotes $\{a \text{ op } b | a \in A\}$. We wish to ensure that the S_i's decrease in size rapidly, so that with high probability $S_k = \emptyset$. Therefore we investigate the number of solutions to $s_1 = s_2 - t$, where $s_1, s_2 \in S_{i-1}$ and $t_i \in T_i$.

To do this, we outline the following work appearing in [ABH*]. Let $\omega = e^{2\pi i/q}$, a primitive qth root of unity, and for $A \subseteq Z_q$, $j = 0, 1, \ldots, q-1$ define

$$\phi_A(j) = \sum_{a \in A} \omega^{ja},$$

and

$$\Phi_A = \max_{1 \leq j \leq q-1} \{|\phi_A(j)|\}.$$

Lemma 2 *[ABH*] The number of solutions to $a + b = c$, where we restrict $a \in A$, $b \in B$, and $c \in C$, is at most*

$$\frac{|A||B||C|}{q} + \Phi_A \sqrt{|B||C|}.$$

Proof. (Sketch) The number of solutions is exactly

$$\frac{1}{q}\sum_{j=0}^{q-1}\phi_A(j)\phi_B(j)\phi_C(-j).$$

Using

$$\sum_{j=0}^{q-1}|\phi_A(j)|^2 = q|A|$$

and Cauchy-Schwartz gives the result. ■

We can use this lemma to show that if enough of the Φ_{T_i} are small, then the source will be good for this prg:

Lemma 3 *If* $\Phi_{T_i} \leq T_i^{3/4}$, *then*

$$E[|S_i|] \leq \frac{|S_{i-1}|^2}{q} + \frac{|S_{i-1}|}{q^{\delta/4}}.$$

Proof. (Sketch) Follows from Lemma 2, $|T_i| = q^\delta$, and because $|S_i|$ equals $1/|T_i|$ times the number of solutions to $s_1 = s_2 - t$, where $s_1, s_2 \in S_{i-1}$ and $t_i \in T_i$. ■

Lemma 4 *If for all* i $\Phi_{T_i} \leq |T_i|^{3/4}$, *then with probability at least* $1/2$, $S_k = \emptyset$ *for* $k = O(\log r + 1/\delta)$.

Proof. We assume that our RP algorithm is incorrect with probability at most $1/128$, and that $q \geq (64/\delta)^{8/\delta}$, i.e. choose $r \geq (8/\delta)(6 - \log_2 \delta)$. In the beginning, the dominant term bounding $E[|S_i|]$ in Lemma 3 will be $|S_i|^2/q$. First we show that when this is the dominant term, the S_i's shrink rapidly, and then we show the same for the other term.

While $|S_i| \geq q^{1-\delta/4}$, we show inductively that with probability at least $3/4 + 1/2^{i+2}$, $|S_i| \leq q/2^{2^i+i+5}$. It is true for $i = 0$, and suppose it is true for i. Then using $|S_i| \geq q^{1-\delta/4}$, $E[|S_{i+1}|] \leq 2|S_i|^2/q$. Then, using Markov, if

$$|S_i| \leq q/2^{2^i+i+5}, \tag{2}$$

then with probability at least $1 - 1/2^{i+3}$,

$$|S_{i+1}| \leq 2^{i+4}|S_i|^2/q \leq q/2^{2^i+i+6}. \tag{3}$$

Using the induction assumption on the probability of (2), we get that (3) holds with probability at least $3/4 + 1/2^{i+3}$, and the induction is complete. Therefore, this phase (when $|S_i| \geq q^{1-\delta/4}$) can only last $\log_2 r$ rounds.

Now we view the phase where $|S_i| \leq q^{1-\delta/4}$. Now $E[|S_{i+1}|] \leq 2|S_i|/q^{\delta/4}$. Using Markov again, with probability at least $1 - 2/q^{\delta/8}$, $|S_{i+1} \leq |S_i|/q^{\delta/8}$. If these decreases in the S_i's continue, then this phase can only last $(1 - \delta/4)/(\delta/8) \leq 8/\delta$ rounds. The probability that the decreases continue is therefore bounded from below by $1 - (8/\delta)2/q^{\delta/8}$. Using $\delta q^{\delta/8} \geq 64$, we see that this probability is at least $3/4$.

Thus, the probability that both phases end as hoped is at least $1/2$, and we are done. ■

We can view a source as having "good" sets T_i, which depend on the previous $t_1, \ldots, t_{i-1}$ output by the source. We can ignore the exponentially small probability that there are not enough "good" T_i's. We now show that most sources have $\Phi_{T_i} \leq |T_i|^{3/4}$ for all i.

Lemma 5 *If the sets $T_1, \ldots, T_k$ are picked uniformly at random from all sets of size q^δ, then the probability that all $\Phi_{T_i} \leq |T_i|^{3/4}$ is $1 - 2^{\Omega(q^{\delta/2})}$.*

Proof. $\phi_{T_i}(j)$ is the sum of random complex numbers of magnitude 1, so using the Martingale Tale Inequality [Spe] on the real and imaginary parts, we conclude $Pr[|\phi_{T_i}(j)| \geq |T_i|^{3/4}] = O(e^{-|T_i|^{1/2}})$. The probability that there exists an i,j with $|\phi_{T_i}(j)| \geq |T_i|^{3/4}$ is at most qk times this quantity, from which the lemma follows. ■

Putting these lemmas together, we conclude

Theorem 1 *For all but an exponential fraction of flat sources, the prg given above will simulate RP.*

Observe that we can make the probability of error exponentially small by repeating our algorithm many times.

4 Second PRG

One problem with getting the previous algorithm to work for all δ-sources is that, for example, the source can output powers of 2 and the witness set could include only odd numbers. We get around these parity problems by working modulo an r-bit prime p. A more serious problem is that, for example, the source can output small numbers and the witness set could be large numbers. In order to get around this type of problem, we exploit the "independence" of the additive and multiplicative groups mod p.

We must be careful, however, because we can have a source that outputs small numbers relative to both the additive and multiplicative groups (by small relative to the multiplicative group we mean a small discrete log base some fixed generator), and a witness set that is large relative to the additive and multiplicative groups. What we will take advantage of is that the distribution of the discrete log of $A + B$, where A and B are large enough sets, is relatively smooth.

Our pseudo-random generator will roughly consist of taking all sums and products of subsets, for example $((x_1 + x_4 + x_6)x_8 + x_{11})x_{14}x_{15}$. More formally, the prg computes a set P as follows:

$P \leftarrow \{1\}$

For $i = 1$ to k' do

$$P \leftarrow P \cup (P + x_i) \cup (Px_i)$$

Again, for the purposes of analysis, we can assume we have $2k$ "good" x_i's. It will be easier to rename them as $t_i = -x_{2i-1}$ and $t_i' = 1/x_{2i}$, where t_i and t_i' are picked uniformly at random from large sets T_i and T_i', respectively. It suffices to analyze the following subset of strings produced:

$P \leftarrow \{1\}$

For $i = 1$ to k do

$$P \leftarrow P \cup (P/t_i' - t_i)$$

To analyze this algorithm, we define the effective set of non-witnesses w.r.t. $(t_1, t_1'), \ldots, (t_i, t_i')$ analogously to the definition for the first algorithm, which leads to the recursive definition:

$$S_i = S_{i-1} \cap (S_{i-1} + t_i)t_i'. \tag{4}$$

We summarize some basic results about multiplicative characters from e.g. [Hua]. A multiplicative character χ is a homomorphism from Z_p^* to the complex numbers, i.e. $\chi(ab) = \chi(a)\chi(b)$ and $\chi(1) = 1$. We also define $\chi(0) = 0$. There are $p-1$ multiplicative characters. Also, for $a \neq 1$,

$$\sum_{\chi} \chi(a) = 0,$$

and if χ is not the trivial character χ_0 which sends everything in Z_p^* to 1, then

$$\sum_{a \in Z_p} \chi(a) = 0.$$

We now make some definitions for multiplicative characters analogous to those in the previous section. For $A \subseteq Z_p^*$, χ a multiplicative character, define

$$\psi_A(\chi) = \sum_{a \in A} \chi(a),$$

and

$$\Psi_A = \max_{\chi \neq \chi_0} \{|\psi_A(\chi)|\}.$$

Then

$$\sum_{\chi} |\psi_A(\chi)|^2 = (p-1)|A|.$$

Then the number of solutions to $ab = c$, where we restrict $a \in A$, $b \in B$, and $c \in C$, $A, B, C \subseteq Z_p^*$, is

$$\frac{1}{p-1} \sum_{\chi} \psi_A(\chi)\psi_B(\chi)\psi_C(\chi^{-1}), \tag{5}$$

which is at most

$$\frac{|A||B||C|}{p-1} + \Psi_A \sqrt{|B||C|}. \tag{6}$$

Dropping the subscripts from the sets in equation 4, we wish to analyze the expected number of solutions to

$$s_1 = (s_2 + t)t', s_1, s_2 \in S, t' \in T', \tag{7}$$

where the expectation is over t picked uniformly from T, and T' can depend on the element t picked from T. ¿From the bound (6), it would seem like a good idea to bound the expectation of Ψ_{S+t}. We can't do this, but we can use a lemma that comes close:

Lemma 6 *(J.H. Lindsey, see e.g. [CG] or [ES]) For any non-trivial character χ,*

$$E[|\psi_{S+t}(\chi)|^2] \leq p|S|/|T|.$$

Proof. (Sketch) First we observe that for $s_1 \neq s_2$,

$$\sum_{t \in Z_p} \chi(s_1 + t)\chi^{-1}(s_2 + t) = \sum_{u \in Z_p^*} \chi(1 + (s_1 - s_2)/u) = -\chi(1) = -1$$

Therefore,

$$\sum_{s_1, s_2 \in S} \sum_{t \in Z_p} \chi(s_1 + t)\chi^{-1}(s_2 + t) \leq p|S|$$

The lemma follows, because the expectation is at most $1/|T|$ times this sum. ■

Unfortunately, this lemma does not give us a bound on the expected value of the maximum of the $|\psi_{S+t}(\chi)|$. However, we do not need this bound, as the following lemma illustrates:

Lemma 7 $E[|S_i|] \leq |S_{i-1}|^2/(p-1) + |S_{i-1}|p^{1/2-\delta}$.

Proof. All expectations will be for t picked uniformly at random from T. Let e denote the expected number of solutions to (7). Using (5) and Cauchy-Schwartz,

$$e - \frac{|S|^2|T'|}{p-1} = \frac{1}{p-1}E\Big[\sum_{\chi\neq\chi_0}\psi_S(\chi^{-1})\psi_{S+t}(\chi)\psi_{T'}(\chi)\Big]$$

$$\leq \frac{1}{p-1}E\Big[\sqrt{\Big(\sum_{\chi\neq\chi_0}|\psi_S(\chi^{-1})|^2|\psi_{S+t}(\chi)|^2\Big)\Big(\sum_{\chi\neq\chi_0}|\psi_{T'}(\chi)|^2\Big)}\Big]$$

$$\leq \sqrt{E\Big[\sum_{\chi\neq\chi_0}|\psi_S(\chi^{-1})|^2|\psi_{S+t}(\chi)|^2\Big](p-1)|T'|}.$$

But Lemma 6 implies that if $\sum_{\chi\neq\chi_0}\lambda_\chi \leq 1$, and $\lambda_\chi \geq 0$, then

$$E\Big[\sum_{\chi\neq\chi_0}\lambda_\chi|\psi_{S+t}(\chi)|^2\Big] \leq p|S|/|T|.$$

In particular, taking $\lambda_\chi = |\psi_S(\chi^{-1})|^2/(p-1)|S|$ and substituting in,

$$e - \frac{|S|^2|T|}{p-1} \leq |S|\sqrt{p}.$$

Using $E[|S_i|] = e/|T'|$ yields the lemma. ■

This lemma plus the techniques of Lemma 4 allow us to conclude:

Theorem 2 *For all $\delta > 1/2$ the above prg simulates RP.*

We remark that a lemma in [MNT] yields a result like Lemma 7. This implies that for $\delta > 1/2$ we can base our prg on any universal family of hash functions, and not only linear congruential generators.

5 Bootstrapping to Handle all $\delta > 0$

The reason our proof did not go through for all $\delta > 0$ is because Lemma 6 is not strong enough. There is, however, a conjectured improvement of something very similar to Lemma 6. We therefore conjecture an improvement to Lemma 6, and show how to bootstrap our algorithm so that it works for all $\delta > 0$.

There is a widely-believed conjecture that the elements of a large rectangle in a specific matrix have small sum:

Paley Graph Conjecture: Let χ_2 be the quadratic character $\chi_2(a) = (\frac{a}{p})$. Then for any $\delta > 0$, there exists an $\epsilon > 0$ such that for large primes p, if $|S|, |T| \geq p^\delta$, then

$$\Big|\sum_{s\in S, t\in T}\chi_2(s+t)\Big| \leq |S||T|/p^\epsilon.$$

We generalize this to

Generalized Paley Graph Conjecture: The Paley Graph Conjecture is true if χ_2 is replaced by any non-trivial character χ.

There are good reasons for believing that this more general conjecture is true if the Paley Graph Conjecture is true. First, techniques used to evaluate character sums involving χ_2 work for any non-trivial character; for example, both the Paley Graph Conjecture and its general version are proved using the same methods for $\delta > 1/2$. Second, as pointed out by Lenstra [Len], the case with χ_2 is probably the worst case, since $\chi_2(a)$ has a $1/2$ "probability" of being a specific value, so it should be easier to find a counter-example in this case.

We now mold the Generalized Paley Graph Conjecture into a more suitable form.

Lemma 8 *Suppose the Paley Graph Conjecture is true. Then for any $\delta > 0$, there exists an $\epsilon > 0$ such that for large primes p, if $|S|, |T| \geq p^\delta$, then*

$$\sum_{t \in T} |\psi_{S+t}(\chi)| \leq |S||T|/p^\epsilon.$$

Proof. (Sketch) Let $z_t = \psi_{S+t}(\chi)$, and set $z_t = a_t + b_t i$, a_t, b_t real. Let $C = \sum_{t \in T} |z_t|$. It suffices to find a $W \subseteq T$ such that $|W| \geq |T|/2$ and $\sum_{w \in W} |z_w| \geq C/6$.

Now either $\sum_{t \in T} |a_t| \geq C/2$ or $\sum_{t \in T} |b_t| \geq C/2$; w.l.o.g. suppose $\sum_{t \in T} |a_t| \geq C/2$. Let $T_1 = \{t \in T | a_t \geq 0\}$, and $T_2 = T \setminus T_1$. Then either $|T_1| \geq |T|/2$ or $|T_2| \geq |T|/2$; say it is T_2. If $|\sum_{t \in T_1} z_t| \geq C/6$, then we can take $W = T_2$. If not, $|\sum_{t \in T_1} z_t| \geq \sum_{t \in T_1} a_t \geq C/2 - C/6 = C/3$. Then, by the triangle inequality, $\sum_{t \in T} z_t \geq C/3 - C/6 = C/6$, so we can take $W = T$. ∎

Corollary 1 *Suppose the Generalized Paley Graph Conjecture is true. Then for any $\delta > 0$, there exists an $\epsilon > 0$ such that for large primes p, if $|S|, |T| \geq p^\delta$, then*

$$\sum_{t \in T} |\psi_{S+t}(\chi)|^2 \leq |S|^2|T|/p^\epsilon.$$

Proof. Follows from Lemma 8 and $|\psi_{S+t}(\chi)| \leq |S|$. ∎

Corollary 2 *Suppose the Generalized Paley Graph Conjecture is true, and fix $\delta > 0$ and the $\epsilon > 0$ implied by the above corollary. Then for any large enough prime p, if the second prg is used, then*

$$E[S_i] \leq |S_{i-1}|^2/(p-1) + |S_{i-1}|\sqrt{|S_{i-1}|/|T_i|p^\epsilon}.$$

Proof. (Sketch.) Follows by substituting the result from Corollary 1 instead of Lemma 6 into the framework of Lemma 7. ∎

We can see that Corollary 2 will help us only if $|T_i| >> |S_{i-1}|/p^\epsilon$. We therefore bootstrap so that our effective non-witness set is small.

Theorem 3 *If the Generalized Paley Graph Conjecture holds, then we can bootstrap the previous prg to simulate RP for any $\delta > 0$.*

Proof. (Sketch.) Suppose our δ-source has $\delta = \delta_0$. Find the ϵ corresponding to $\delta = \delta_0$ in the Generalized Paley Graph Conjecture. Let h be the least integer such that $\delta_0 + h\epsilon/2 > 1/2$. We

claim inductively that if there is a prg P which simulates RP for $\delta = \delta_0 + k\epsilon/2$, then there is a prg for $\delta = \delta_0 + (k-1)\epsilon/2$, for $k \geq 0$. Theorem 2 implies the claim for $k = h$.

Suppose the claim is true for a given k. The key observation is that another way of viewing the inductive claim is that the size of the set of non-witnesses of the R-bit strings is much less than $2^{(\delta_0+k\epsilon/2)R}$, or, if p_R is an R-bit prime, much less than $p_R^{\delta_0+k\epsilon/2}$. Namely, an R-bit string X is a witness if one of the r-bit strings that the prg maps X to is a witness.

Thus, using that we effectively have $|S_{i-1}| \leq p_R^{\delta_0+k\epsilon/2}$, Corollary 2, and the techniques of Lemma 4, we see that there is a prg which simulates RP for $\delta = \delta_0 + (k-1)\epsilon/2$.

■

6 Acknowledgements

I would like to thank Umesh Vazirani, Noam Nisan, Prasoon Tiwari, Russell Impagliazzo, Hendrik Lenstra, Vaughan Pratt, and Avi Wigderson for helpful discussions.

References

[ABH*] M. Ajtai, L. Babai, P. Hajnal, J. Komlos, P. Pudlak, V. Rodl, E. Szemeredi, and G. Turan, "Two Lower Bounds for Branching Programs," 18th STOC, 1986, pp. 30-38.

[Blu] M. Blum, "Independent Unbiased Coin Flips from a Correlated Biased Source: a Finite Markov Chain," *Combinatorica*, 6 (2):97-108, 1986.

[BL] M. Ben-Or and N. Linial, "Collective Coin Flipping Robust Voting Schemes and Minimal Banzhaf Values," 26th FOCS, 1985, pp. 408-416.

[CG] B. Chor and O. Goldreich, "Unbiased Bits from Sources of Weak Randomness and Probabilistic Communication Complexity," 26th FOCS, 1985, pp. 429-442.

[CGH*] B. Chor, O. Goldreich, J. Hastad, J. Friedman, S. Rudich, and R. Smolenski, "The Bit Extraction Problem or t-Resilient Functions," 26th FOCS, 1985, pp. 396-407.

[CW] A. Cohen and A. Wigderson, "Dispersers, Deterministic Amplification, and Weak Random Sources," 30th FOCS, 1989, pp. 14-19.

[ES] P. Erdos and J. Spencer, *Probabilistic Methods in Combinatorics*, Academic Press, 1974.

[Hua] H. Hua, *Introduction to Number Theory*, Springer-Verlag, 1982.

[IZ] R. Impagliazzo and D. Zuckerman, "How to Recycle Random Bits," 30th FOCS, 1989, pp. 248-253.

[Len] H.W. Lenstra, personal communication.

[LLS] D. Lichtenstein, N. Linial, and M. Saks, "Imperfect Sources of Randomness and Discrete Controlled Processes," 19th STOC, 1987, pp. 169-177.

[MNT] Y. Mansour, N. Nisan, and P. Tiwari, "The Computational Complexity of Universal Hashing," 22nd STOC, 1990.

[PSS] J. Pintz, W. Steiger, and E. Szemeredi, "Two Infinite Sets of Primes with Fast Primality Tests," 20th STOC, 1988, pp. 504-509.

[Rab] M. O. Rabin, "Probabilistic Algorithm for Testing Primality," *Journal of Number Theory*, 12:128-138, 1980.

[SV] M. Santha and U. Vazirani, "Generating Quasi-Random Sequences from Slightly Random Sources," 25th FOCS, 1984, pp. 434-440.

[Spe] J. Spencer, *Ten Lectures on the Probabilistic Method*, SIAM, Philadelphia, 1987, pp. 55-56.

[Vaz] U. Vazirani, "Randomness, Adversaries and Computation," PhD Thesis, University of California, Berkeley, 1986.

[VV] U. Vazirani and V. Vazirani, "Random Polynomial Time is Equal to Semi-Random Polynomial Time," 26th FOCS, 1985, pp. 417-428.