On the Security of Goldreich's One-Way Function

Andrej Bogdanov[1] and Youming Qiao[2]

[1] Dept. of Computer Science and Engineering, The Chinese University of Hong Kong
andrejb@cse.cuhk.edu.hk
[2] Institute for Theoretical Computer Science, Tsinghua University
jimmyqiao86@gmail.com

Abstract. Goldreich (ECCC 2000) suggested a simple construction of a candidate one-way function $f : \{0,1\}^n \to \{0,1\}^m$ where each bit of output is a fixed predicate P of a constant number d of (random) input bits. We investigate the security of this construction in the regime $m = Dn$, where $D(d)$ is a sufficiently large constant. We prove that for any predicate P that correlates with either one or two of its variables, f can be inverted with high probability.

We also prove an amplification claim regarding Goldreich's construction. Suppose we are given an assignment $x' \in \{0,1\}^n$ that has correlation $\epsilon > 0$ with the hidden assignment $x \in \{0,1\}^n$. Then, given access to x', it is possible to invert f on x with high probability, provided $D = D(d, \varepsilon)$ is sufficiently large.

1 Introduction

In a short note in 2000, Oded Goldreich [Gol00] proposed a very simple construction of a conjectured one-way function:

1. Choose a bipartite graph G with n vertices on the left, m vertices on the right, and regular right-degree d.
2. Choose a predicate $P : \{0,1\}^d \to \{0,1\}$.
3. Let $f = f_{G,P}$ be the function from $\{0,1\}^n$ to $\{0,1\}^m$ defined by

   $$f(x)_i = \text{the } i\text{th bit of } f(x) = P(x_{\Gamma(i,1)}, \ldots, x_{\Gamma(i,d)})$$

   where $\Gamma_{(i,j)}$ is the jth neighbor of right vertex i of G.

Goldreich conjectured that when $m = n$ and d is constant, for "most" graphs G and predicates P, the resulting function is one-way.[1]

In this work we investigate Goldreich's construction in the setting where the graph G is random, d is constant, and $m = Dn$ for a sufficiently large constant

[1] More precisely, with constant probability over the choice of G and P (say 2/3), the corresponding family of functions as $n \to \infty$ is one-way. Goldreich also suggests specific choices of P and G.

I. Dinur et al. (Eds.): APPROX and RANDOM 2009, LNCS 5687, pp. 392–405, 2009.

$D = D(d)$. We show that for this setting of parameters, Goldreich's construction is not secure for most predicates P. In fact, our conclusion holds for every predicate P that exhibits a correlation with either one of its variables or a pair of its variables.

We also show that if we are given a "hint" x' – any assignment that has nontrivial correlation with the actual input x to the one-way function – it is possible to invert f on x, as long as D is a sufficiently large constant. However, D depends not only on d but also on the correlation between x and x'.

While our theorem does not rule out the security of Goldreich's construction when $m = n$, it indicates some possible difficulties in using this construction, as it reveals its sensitivity on the output length. It indicates that when the ratio m/n is a sufficiently large constant, the construction can be broken for a large class of predicates. It is also easy to see that when m/n is smaller than $1/(d-1)$ the function can also be inverted for every predicate P, as with high probability the "constraint hypergraph" splits into components of size $O(\log n)$ [SS85].

On the other hand, for certain choices of the predicate P to which our theorem does not apply, it has been conjectured that the function f is not only one-way but also a pseudorandom generator [MST03].[2]

1.1 Goldreich's Function and Cryptography in $\mathbf{NC^0}$

Goldreich's proposal for a one-way function has several features that were absent from all known earlier proposals: (1) It is extremely simple to implement, and (2) it is very fast to compute, especially in parallel. On the other hand, the conjectured security of Goldreich's function is not known to relate to any standard assumptions in cryptography, such as hardness of factoring or hardness of finding short vectors in lattices.

This paradigm of "NC^0 cryptographic constructions" where every bit of the output depends only on a constant number of input bits has since been extended to other cryptographic primitives, in particular pseudorandom generators. Remarkably, Applebaum, Ishai, and Kushilevitz [AIK04] showed that a pseudorandom generator (and in particular a one-way function) in NC^0 can be obtained assuming the hardness of the discrete logarithm problem; however, the stretch of this pseudorandom generator is only constant. In a different work [AIK06], the same authors gave a different construction of a pseudorandom generator with small linear stretch using the less standard assumption that certain random linear codes are hard to decode.

These constructions give evidence that cryptography in NC^0 may be possible. However, the constructions are rather complicated and the parameters they yield are of little practical value. For example, it is not known whether it is possible to have a pseudorandom generator that stretches n bits of input into, say, $10n$ bits of output under comparable assumptions.

For this reason, we believe it is interesting to investigate the power and limitations of simple constructions such as the one of Goldreich, which may be more

[2] Actually [MST03] considers a slightly different function; see below.

useful in practice. A step in this direction was made by Mossel, Shpilka, and Trevisan [MST03]. They conjectured that the function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ where

$$f(x,y)_i = x_{\Gamma(i,1)} + x_{\Gamma(i,2)} + x_{\Gamma(i,3)} + y_{\Delta(i,1)} \cdot y_{\Delta(i,2)}$$

is a pseudorandom generator with high probability, where Γ and Δ are incidence lists of random (n, m) bipartite graphs of right-degree 3 and 2 respectively. As partial evidence towards their conjecture, Mossel et al. proved that f is pseudorandom against linear functions for, say, $m = n^{1.1}$. It is not difficult to see by the Linial-Nisan conjecture [LN90], which was recently proved [Bra09], f is also pseudorandom against constant-depth circuits.

Very recently, Cook, Etesami, Miller, and Trevisan [CEMT09] showed that a restricted class of algorithms called "myopic algorithms" take exponential time to invert Goldreich's construction. The kinds of algorithms used in this work are not myopic.

1.2 Our Results

We state our main results. They refer to the standard notion of "correlation" among strings and functions which is formally defined in Section 2.

Theorem 1. *Let K be a sufficiently large constant and $D > 2^{Kd}$. Suppose $P : \{0,1\}^d \to \{0,1\}$ is a predicate that has nonzero correlation with one of its inputs or a pair of its inputs. Consider the function $f_{G,P} : \{0,1\}^n \to \{0,1\}^m$, where $m = Dn$. Then, with high probability over G, $f_{G,P}$ is invertible on a $1-2^{-2^{-\Omega(d)}n}$-fraction of inputs as a one-way function.*

Theorem 2. *Let K be a sufficiently large constant and $D > (1/\varepsilon)^{Kd}$. Let $P : \{0,1\}^d \to \{0,1\}$ be any non-constant predicate. Then there is an algorithm A such that with high probability over G, with the following holds. Consider the function $f_{G,P} : \{0,1\}^n \to \{0,1\}^m$, where $m = Dn$. For a $1-2^{-\varepsilon^2 2^{-\Omega(d)}n}$ fraction of assignments x and any assignment x' that has correlation ε (in absolute value) with x, on input $G, P, f(x)$ and x', A outputs an inverse for $f_{G,P}(x)$. The running time of A is polynomial in n and $1/\varepsilon^d$.*

1.3 Our Approach

The problem of inverting Goldreich's function is somewhat analogous to the problem of reconstructing assignments to random 3SAT formulas in the planted 3SAT model. We exploit this analogy and show that several of the tools developed for planted 3SAT can be applied to our setting as well.

The proofs of Theorems 1 and 2 consist of two stages. In the first stage, we almost invert f in the sense that we find an assignment z that matches the hidden assignment x on a 99% fraction of positions. In the second stage we turn z into a true inverse for $f(x)$. The second stage is common to the proofs of both theorems.

To give some intuition about the first stage in Theorem 1, suppose for instance that P is the majority predicate. Then we try to guess a the value of the bit x_i by looking at all constraints where x_i appears and taking the majority of these values. Since x_i has positive correlation with the majority predicate, we expect this process to result in a good guess for most x_i that appear in a sufficiently large number of clauses. In fact, if f has about $n \log n$ bits of output, this reconstructs the assignment completely; if $m = Dn$ for a sufficiently large constant D, a large constant fraction of the bits of x is recovered. The same idea applies to any predicate with correlates to one of its variables.

For predicates correlating with a pair of their variables, we will argue that the output of f contains certain noisy information about the correlation between the pairs. In particular, it gives information as to whether the pair of variables have the same or different values. More precisely, it is possible to construct a graph G whose vertices correspond to variables of i and an edge between i and j appears independently, but with probability depending on the event $x_i = x_j$. The clusters in this graph correspond to variables taking the same value. Using known methods for clustering random graphs [Coj06] we can recover most of the values of x.

The first stage in the proof of Theorem 2 is based on the observation that if we start with some assignment x' that correlates with the input x to f, then the output bits of $f(x)$ give information about the values of various variables x_i, for an arbitrary predicate P. We prove this in Section 4.

For the second stage, we extend an algorithm of Flaxman [Fla03] (similar ones have also been given in [Vil07, KV06]) for reconstructing planted assignments of random 3CNF formulas. The planted 3SAT model can be viewed as a variant of our model where the predicate P corresponds to one of the eight predicates $z_1 \vee z_2 \vee z_3, \dots, \overline{z_1} \vee \overline{z_2} \vee \overline{z_3}$. This algorithm starts from an almost correct assignment, then unsets a small number of the variables in this assignment according to some condition ("small support size"), so that with high probability all (but a constant number of) the remaining set variables are correct. Then the value of the unset variables can be inferred in polynomial time. We show that the notion of "small support size" can be generalized to arbitrary non-constant predicates, and this type of algorithm can be used to invert f. While we directly follow previous approaches, our proofs include a few technical simplifications.

2 Preliminaries

Some definitions. Let X, Y be random variables over $\{0,1\}$. The *correlation* between X and Y is the value $\mathrm{E}[(-1)^{X+Y}]$. The correlation between a predicate $P : \{0,1\}^d \to \{0,1\}$ and a subset $(x_i)_{i \in S}$ of its inputs is the correlation between the random variables $P(X_1, \dots, X_d)$ and $\sum_{i \in S} X_i$, where the sum is taken modulo 2, and $X_1, \dots, X_n$ are uniformly distributed. We say P correlates with $(x_i)_{i \in S}$ if the above correlation is nonzero. The correlation between a pair of assignments $x, y \in \{0,1\}^n$ is the correlation between the ith bit of x and y, where $i \in [n]$ is random.

We say a Bernoulli random variable $X \sim \{0,1\}$ is *ε-biased towards* 0 (resp. 1) if the probability of $X = 0$ is at most $1/2 - \varepsilon$ (resp. $1/2 + \varepsilon$).

We say an assignment $x \in \{0,1\}^n$ is *ε-balanced* if its correlation with the all zero assignment is at most ε in absolute value.

By analogy with the random 3SAT problem, we will refer to the input $x \in \{0,1\}^n$ on which we are interested the function $f_{G,P}(x)$ as the *planted assignment.* We will call an assignment $x' \in \{0,1\}^n$ *d-correct* if it is at hamming distance at most d from the planted assignment.

On the random graph model. In Goldreich's definition [Gol00], The bipartite graph G in the function $f_{G,P}$ is chosen from the following random graph model $\mathcal{G} = \{\mathcal{G}_{n,m}\}$: (1) Each graph G in $\mathcal{G}_n$ has n left vertices and $m = m(n)$ right vertices; (2) each right vertex v of G has d neighbors on the left, labeled by $\Gamma_1(v), \dots, \Gamma_d(v)$; (3) The neighbors of each right vertex are uniformly distributed (repetitions allowed) and independent of the neighbors of all other vertices.

The literature on planted 3SAT usually considers a different model where each of the clauses is included in the formula independently with probability $p = p(n)$. Our results can be extended in the corresponding model for G, but such a model is less natural for one-way functions.

3 Obtaining an Almost Correct Assignment

In this section, we show that for predicates correlating with one or a pair of inputs, we can get an assignment that agrees with the planted one on almost all variables.

3.1 For Predicates Correlating with One Input

When the predicate $P(z_1, \dots, z_k)$ correlates with one of its inputs, say z_1, then every output bit of $f_{G,P}(x)$ gives an indication about what the corresponding input bit should be. If we think of this indication as a vote, and take a majority of all the votes, we set most of the input bits correctly. The following proposition formalizes this idea.

Algorithm Majority Voting
INPUTS: A predicate $P(z_1, \dots, z_d)$ that correlates with z_k; the graph G; the value $f_{G,P}(x)$
ALGORITHM.

1. For every input variable i, calculate the majority among the values $f_{G,P}(x)_j$ where i occurs as the kth variable.
2. Set x'_i to equal this value if the correlation between P and z_k is positive, and the complement of this value otherwise.
3. Output the assignment x'.

Proposition 1. *Suppose $D > 4^d$ and P is a predicate that correlates with its kth variable. For a $1 - 2^{-\Omega(n/d^2 4^d)}$ fraction of $x \in \{0,1\}^n$ and with probability $1 - 2^{-\Omega(4^d n)}$ over the choice of G, the assignment x' produced by algorithm Majority Voting agrees with x on a $(1 - 2^{-\Omega(D/4^d)})n$ fraction of variables.*

Proof. Without loss of generality assume $k = 1$, and assume the correlation between P and z_1 is positive. Since this correlation is a multiple of 2^{-d}, it must then be at least 2^{-d}.

Now fix any input x that is $1/2d2^d$-balanced. We think of the constraint graph G as being chosen in the following manner: First, for each constraint in G the first variable i_1 is chosen uniformly at random. Then, for every i, among the constraints where i is the first variable, the other variables $i_2, \dots, i_d$ are chosen at random. Let N_i denote the number of constraints with i as the first variable.

Now consider the random experiment where one samples $x_{i_2}, \dots, x_{i_d}$ at random and outputs the value $b = P(x_i, x_{i_2}, \dots, x_{i_d})$. If $x_{i_2}, \dots, x_{i_d}$ were uniformly distributed in $\{0,1\}$, then b is a Bernoulli random variable whose output is at least 2^{-d}-biased towards x_i. However, $x_{i_2}, \dots, x_{i_d}$ might not be uniformly distributed but only $1/2d2^d$-balanced. Since the statistical difference between the distributions $(x_{i_2}, \dots, x_{i_d})$ when the samples are uniform and when they are uniformly balanced is at most $(d-1)/2d2^d \le 2^{-(d+1)}$, it follows that b is at least $2^{-(d+1)}$-biased towards x_i.

Fix some i such that $N_i \ge D/2$. By Chernoff bounds, over the random choice of G, the value x'_i agrees with x_i with probability at least $1 - 2^{-\Omega(4^{-d}D)}$. By another Chernoff bound, the number of is among those N_i such that $N_i \ge D/2$ where x_i and x'_i disagree is at most $2^{-\Omega(4^{-d}D)}n$ with probability $2^{-\Omega(4^{-d}Dn)}$. Applying Lemma 4 with $\varepsilon = 4^d/D$ we obtain the theorem. □

3.2 For Predicates Correlating with a Pair of Inputs

We illustrate the inversion of $f_{G,P}(x)$ for a predicate that correlates with a pair of its inputs by looking at the "all equal" predicate. Specifically, let $AE(z_1, z_2, z_3)$ be the predicate "$z_1 = z_2 = z_3$". Then AE does not correlate with any of its variables, but it correlates with the pair (z_1, z_2).

In this example, every constraint $(x_{i_1}, x_{i_2}, x_{i_3})$ where AE evaluates to 1 tells us that $x_{i_1} = x_{i_2}$. Now construct a graph H whose vertices are variables of x and such a constraint gives rise to an edge (i_1, i_2). Then the connected components in this graph indicate collections of variables x_i that must have the same value. When x is roughly balanced, because G is random, the induced subgraphs on the sets $\{i : x_i = 0\}$ and $\{i : x_i = 1\}$ are random graphs with constant average degree. Therefore with high probability, each of these subgraphs will have a giant connected component, giving two large sets of variables of x that must have the same value. By guessing the value of the variables within each set we obtain an assignment x' that agrees with x almost everywhere.

Now consider the majority predicate $MAJ(z_1, z_2, z_3)$. This predicate also correlates with its first pair of variables. Fix an almost balanced assignment x. Now suppose we see a constraint such that $MAJ(x_{i_1}, x_{i_2}, x_{i_3}) = 1$. While we cannot say with certainty that $x_{i_1} = x_{i_2}$, this constraint gives an indication that x_{i_1} and x_{i_2} are more likely to be different than equal. So we can hope to recover a large portion of the assignment x by looking for a large cut in the graph H.

For a general predicate that correlates with a pair of its variables, we can reconstruct a large portion of the assignment x by using a spectral partitioning algorithm on H. This idea was used by Flaxman [Fla03] in a related context. Coja-Oghlan [Coj06] proved a general "partitioning theorem" which, in particular, gives the following algorithm.

Theorem 3 (Theorem 1 of [Coj06], special case). *There is a polynomial-time algorithm* Partition *with the following property. Let C_0 be a sufficiently large constant. Let (S_0, S_1) be a partition of $[n]$ such that $|S_0|, |S_1| \geq n/3$. Fix probabilities $p_{00}, p_{11}, p_{01} \in [C_0/n, D/n]$. Suppose the graph H' is a random graph where each edge (i, j), where $i \in S_a, j \in S_b$ $(a \leq b)$ is included independently at random with probability p_{ab}. Assume that*

$$n(|p_{00} - p_{01}| + |p_{11} - p_{01}|) \geq C_0 \max(\sqrt{np_{00}\log(np_{00})}, \sqrt{np_{11}\log(np_{11})}) \ , \quad (1)$$

then with high probability Partition(H') outputs a partition (S'_0, S'_1) of $[n]$ such that (S_0, S_1) and (S'_0, S'_1) differ on at most $(1 - O(D^{-10}))n$ vertices of H'.

Condition (1) is a non-degeneracy condition which requires there to be a noticeable difference in edge densities. Otherwise, the information about the original partition is lost.

Algorithm Pairwise

INPUTS: A predicate $P(z_1, \ldots, z_d)$ that correlates with (z_k, z_r); the graph G; the value $f_{G,P}(x)$

ALGORITHM

1. Choose b such that $\Pr_z[z_k \neq z_r \mid P(z) = b] \neq \Pr_z[z_k = z_r \mid P(z) = b]$.
2. Construct the graph H on vertex set $[n]$ with edges (i_k, i_r) iff there is a constraint in G such that $P(x_{i_1}, \ldots, x_{i_d}) = b$. Let m_H denote the number of edges of H.
3. Sample M from the binomial distribution with $\binom{n}{2}$ samples, each with probability $m_H/2$. Let H' be the subgraph consisting of the first M distinct edges of G. (If there are not enough such edges, fail.)
4. Run Partition(H'). Call the partition output by the algorithm (S'_0, S'_1).
5. Output the pair of assignments x', $\overline{x}'$, where $x'_i = a$ iff $i \in S'_a$, and $\overline{x}'$ is the complementary assignment.

For step 1, it follows that such a choice of b is always possible by the assumption that P correlates with (z_k, z_r). Step 3 is a technical trick that allows us to pass from our random graph model, where the number of edges is fixed, to the model where each edge is sampled independently at random with probability $m_H/2$. We believe this step is not necessary, but since the algorithm *Partition* is analyzed in the latter model we include it for accuracy.

Proposition 2. *Fix a sufficiently large constant C. Suppose $D > Cd16^d$ and P is a predicate that correlates with (z_k, z_r). For a $1 - 2^{-\Omega(d4^d)}$ fraction of $x \in \{0,1\}^n$ and with high probability over the choice of G, one of the two assignments produces by algorithm Pairwise agrees with x on a $(1 - \Omega(D^{-10}))n$ fraction of variables.*

Proof. Without loss of generality assume $b = 1$, $k = 1$ and $r = 2$. Let $p_{\neq} = \Pr_z[z_1 \neq z_2 \mid P(z) = 1]$, $p_= = \Pr_z[z_1 = z_2 \mid P(z) = 1]$. The fact that P is correlated with (z_k, z_r) implies that $|p_= - p_{\neq}| \geq 4^{-d}$.

Let us first fix a balanced input x. Let S_0 and S_1 denote the 0 and 1 variables of x. Let m_H be the number of 1-outputs of $f_{G,P}(x)$. Conditioned on $P(x_{i_1}, \ldots, x_{i_d}) = 1$, we can think of $i_1, \ldots, i_d$ as chosen by the following process. First, we determine where in the partition (S_0, S_1) the indices i_1 and i_2 belong. Then we randomly sample i_1 and i_2 from the corresponding set in the partition. Then we choose $i_3, \ldots, i_d$. This process induces the following random graph H: For each of m_H edges, first randomly choose where in the partition the edge belongs. We put the edge in (S_0, S_0) and (S_1, S_1) with probability $p_=/2$ and in (S_0, S_1) with probability $p_{\neq}$. Then randomly choose an edge on that side of the partition.

Disregarding the possibility that step 3 fails, the graph H' is then a random graph with edge densities $p_{00}, p_{11} = p_= m_H/n(n-1)$, and $p_{01} = p_{\neq} m_H/n(n-1)$. By Chernoff bounds, $m_H > m/2^d$ with high probability. Then for $D > C_1 d16^d$ condition (1) will be satisfied and with high probability over the choice of G, the algorithm will return the correct partition.

To complete the proof we need to analyze the effect that the imbalance of x and the step 3 failure have on this ideal scenario. We now assume that x is $1/2d4^d$-balanced. It can be checked (similarly to the proof of Proposition 1) that this affects the probabilities p_{00}, p_{01}, p_{11} by at most $2^{-(2d+1)} m_H/n(n+1)$, so condition (1) will still be satisfied. By Chernoff bounds, step 3 succeeds with high probability. □

4 Amplifying Assignments

In this section we give the proof of Theorem 2. As discussed, the proof goes in two stages. First, we find an assignmnent w that agrees with x on most inputs. Then we use Theorem 4 to invert f. We focus on the first stage.

The idea of the algorithm is to use the assignment x' in order to get empirical evidence about the values of each variable x_i in the hidden assignment. First, since the predicate $P(z)$ is nontrivial, it must depend on at least one of its variables, say z_1. To obtain evidence about the value of x_i, let's consider all constraints in which x_i appears as the first variable. Since G is random, we expect the number of such constraints to be fairly large. Moreover, the other variables appearing in the constraints are also random.

Now let us fix a pair of assignments x and x' with correlation ε, a variable i, and a value $b \in \{0,1\}$, and look at the probability distribution D_b generated by the following process:[3]

[3] It is easy to see that D_b does not depend on i.

1. Choose a random G.
2. Choose a random constraint j of $f_{G,P}$ where i appears as the first variable. Call the other variables $i_2, \dots, i_d$.
3. Output $(x'_{i_2}, \dots, x'_{i_d}, f(b, x_{i_2}, \dots, x_{i_d})_j)$.

Our main observation (see Lemma 1 below) is that the distributions D_0 and D_1 are statistically far apart. Therefore we can determine the value $b = f(x)$ with good confidence by observing enough samples from one of these two distributions. But observing the values $f(x)_j$ in those constraints j where i appears as the first variable amounts exactly to sampling from this process. This suggests the following algorithm for computing w:

Algorithm Amplify. On input P, G, $f(x)$, ε, an assignment x' that ε-correlates with x,

```
1. Compute the distributions $D_0$ and $D_1$ (see below).
2. For every $i$, compute the empirical distribution $\hat{D}_i$ defined as follows:
    (a) Choose a random constraint $(i, i_2, \dots, i_d)$ of $f$ where $i$ is the first variable.
    (b) Output $(x'_{i_2}, \dots, x'_{i_d}, f(b, x_{i_2}, \dots, x_{i_d})_j)$.
3. Set $w_i = b$ if $\hat{D}_i$ is closer to $D_b$ than to $D_{1-b}$ in statistical distance.
```

Proposition 3. *Let G be random right regular bipartite graph with n left vertices and $2^{\varepsilon^{Dd}} n$ right vertices, where D is a sufficiently large constant. With high probability over the choice of G, for a $1 - 2^{-\Omega(\varepsilon^2 n)}$ fraction of assignments x and every assignment x' that has correlation ε with x, algorithm Amplify outputs assignments $w_1, \dots, w_n$ so that at least one of them agrees with x in a $1 - \varepsilon$ fraction of places.*

As discussed above, the proof of this theorem consists of two steps. First, we show that the distributions D_0 and D_1 are statistically far apart. Then, we show that with high probability over G, for most i the distribution $\hat{D}_i$ is statistically close to D_{x_i}.

Lemma 1. *Let x and x' be two assignments such that x is $\varepsilon/2$-balanced and x' has correlation ε with x. Then the statistical distance between D_0 and D_1 is at least $\varepsilon^{-O(d)}$.*

We observe that the distance can be as small as $\varepsilon^{-\Omega(d)}$, for example if P is the XOR predicate on d variables, x is any balanced assignment, and x' is an assignment that equals 1 on a $1 - \varepsilon$ fraction of inputs and 0 on the other inputs.

Proof. We begin by giving alternate descriptions of the distributions D_b. To do this, we define a distribution $\mathcal{F}$ over $\{0,1\}^2$ as follows: First, choose $i \in [n]$ at random, then output the pair (x_i, x'_i). Let (a, a') denote a pair sampled from $\mathcal{F}$. It is not difficult to see that

$$\min(\Pr[a' = 0], \Pr[a' = 1]) \geq \varepsilon/2 \tag{2}$$

for if this were not the case, it would violate the assumptions on x and x'.

The distribution D_b can now be described as follows:

1. Uniformly and independently sample pairs $(a_i, a'_i) \sim \mathcal{F}$ for $i = 2, \dots, n$.
2. Output $(a'_2, \dots, a'_d, P(b, a_2, \dots, a_d))$.

Intuitively, this corresponds to the process of first sampling input bits from x', then evaluating P at a "noisy" version of x'. If there was no noise, it is easy to see that D_0 and D_1 must be far apart, as they have to differ for at least one setting of $a'_2, \dots, a'_d$, and by (2) this happens with probability at least $(\varepsilon/2)^{d-1}$.

To argue the general case, note that the statistical distance between D_0 and D_1 is bounded below by the quantity

$$\begin{aligned}
&\mathrm{sd}(D_0, D_1)\\
=&\sum_{(a'_2,\dots,a'_d)\in\{0,1\}^{d-1}} 2\cdot\mathcal{F}^{d-1}(a'_2,\dots,a'_d)\\
&\qquad\cdot\left|\mathrm{E}_{\mathcal{F}^{d-1}}[P(0,a_2,\dots,a_d) - P(1,a_2,\dots,a_d) \mid a'_2,\dots,a'_d]\right|\\
\geq& 2\cdot(\varepsilon/2)^{d-1}\\
&\qquad\cdot\max_{(a'_2,\dots,a'_d)}\left|\mathrm{E}_{\mathcal{F}^{d-1}}[P(0,a_2,\dots,a_d) - P(1,a_2,\dots,a_d) \mid a'_2,\dots,a'_d]\right|\\
\geq& 2\cdot(\varepsilon/2)^{d-1}\\
&\qquad\cdot\mathrm{E}_{(a'_2,\dots,a'_d)}\left[\mathrm{E}_{\mathcal{F}^{d-1}}[P(0,a_2,\dots,a_d) - P(1,a_2,\dots,a_d) \mid a'_2,\dots,a'_d]^2\right]^{1/2}
\end{aligned}$$

where $\mathcal{F}^{d-1}(a'_2, \dots, a'_d)$ denotes the probability of sampling $a'_2, \dots, a'_d$ in $d-1$ independent copies of $\mathcal{F}$, the expectation $\mathrm{E}_{\mathcal{F}^{d-1}}$ is taken over independent choices of $a_2, \dots, a_d$ where each a_i is sampled from the distribution $\mathcal{F}$ conditioned on a'_i, and the expectation $\mathrm{E}_{(a'_2,\dots,a'_d)}$ refers to a uniformly random choice of $(a'_2, \dots, a'_d) \sim \{0,1\}^{d-1}$.

To lower bound the last quantity, we consider the linear operator T_{d-1} on the space $\mathbf{R}^{\{0,1\}^{d-1}}$ defined by

$$(T_{d-1}g)(a'_2, \dots, a'_d) = \mathrm{E}_{\mathcal{F}^{d-1}}[g(a_2, \dots, a_d) \mid a'_2, \dots, a'_d].$$

Let T_{d-1}^{-1} denote its inverse (whose existence will be argued) and $\|\cdot\|_2$ denote the ℓ_2 operator norm. Recall that for any linear operator T,

$$\|T\|_2 = \max_g \|Tg\|_2 / \|g\|_2 = \max|\sigma|$$

where the maximum ranges over the singular values σ of T. Applying this definition to the operator T_{d-1}^{-1}, we have that

$$\begin{aligned}
&\left\|T_{d-1}^{-1}\right\|_2\cdot\mathrm{E}_{(a'_2,\dots,a'_d)}\left[\mathrm{E}_{\mathcal{F}^{d-1}}[P(0,a_2,\dots,a_d) - P(1,a_2,\dots,a_d) \mid a'_2,\dots,a'_d]^2\right]^{1/2}\\
&\qquad\geq \mathrm{E}_{(a_2,\dots,a_d)}\left[(P(0,a_2,\dots,a_d) - P(1,a_2,\dots,a_d))^2\right]^{1/2} \geq 2^{-d+1}
\end{aligned}$$

We are left with the task of upper bounding the quantity $\left\|T_{d-1}^{-1}\right\|_2$. It is bounded by the largest (in absolute value) singular value of the operator T_{d-1}^{-1}, which is the inverse of the smallest singular value of $T_{d-1} = T_1^{\otimes(d-1)}$. Putting everything together, we obtain that

$$\mathrm{sd}(D_0, D_1) \geq 2 \cdot (\varepsilon/4)^{d-1} \cdot |\sigma|^{d-1}$$

where σ is the smaller singular value of the operator T_1. A calculation of the singular values of T_1 (which we omit) shows that $|\sigma| = \Omega(\varepsilon)$, so $\mathrm{sd}(D_0, D_1) = \varepsilon^{-O(d)}$. □

We now prove that the distributions $\hat{D}_i$ are mostly close to the distributions D_{x_i}. We will need the following crude bound on the number of samples needed in order to approximate a distribution with bounded support by its empirical average. It easily follows from Chernoff bounds.

Lemma 2. *Suppose $\mathcal{D}$ is a distribution on a set of size S and $\hat{\mathcal{D}}$ is the empirical average of N independent samples of $\mathcal{D}$, where $N \geq 3S^2/\gamma^2 \log(S/\delta)$. Then*

$$\Pr[\mathrm{sd}(\mathcal{D}, \hat{\mathcal{D}}) < \gamma] > 1 - \delta.$$

Lemma 3. *Fix any constants $\gamma, \varepsilon > 0$. Suppose G is a random graph with n left vertices and Dn right vertices, where $D \geq 24d2^d \log(3/\varepsilon)/\gamma^2$. With probability $1 - 2^{-\Omega(\varepsilon^2 n)}$ over the choice of G, for a $1 - 2^{\Omega(\varepsilon^2 n)}$ fraction of assignments x, for at least a $1 - \varepsilon$ fraction of i, for every assignment x' that has correlation ε with x, we have that $\mathrm{sd}(\hat{D}_i, D_{x_i}) < \gamma$.*

Proof. Fix an $\varepsilon/2$-balanced assignment x. We will show that

$$\Pr_G\big[|\{i : \mathrm{sd}(\hat{D}_i, D_{x_i}) \geq \gamma\}| > \varepsilon n\big] = 2^{-\Omega(\varepsilon^2 n)}.$$

Since at most $2^{-O(\varepsilon^2 n)}$ assignments x are not balanced, it follows that

$$\Pr_{x,G}\big[|\{i : \mathrm{sd}(\hat{D}_i, D_{x_i}) \geq \gamma\}| > \varepsilon n\big] < 2^{-\Omega(\varepsilon^2 n)}$$

from where the lemma follows by Markov's inequality.

We think of the constraint graph G as being chosen in the following manner: First, for each constraint in G the first variable i_1 is chosen uniformly at random. Then, for every i, among the constraints where i is the first variable, the other variables $i_2, \dots, i_d$ are chosen at random. Let N_i denote the number of constraints with i as the first variable. Observe that conditioned on the choices of N_i, the events

$$\mathrm{sd}(\hat{D}_i, D_{x_i}) \geq \gamma$$

are independent of one another. Let E_i be an indicator variable for this event. Moreover, the distribution $\hat{D}_i$ is an empirical average of N_i samples from D_{x_i}, so by Lemma 2 we have that as long as $N_i \geq D/2$, $\Pr_G[E_i = 1 \mid N_i] \leq \varepsilon/3$.

Let I denote the set of those i such that $N_i < D/2$. Then

$$\begin{aligned}\Pr_G[\sum_{i\in[n]} E_i \geq \varepsilon n] &\leq \Pr_G[\sum_{i\in[n]} E_i \geq \varepsilon n \mid |I| < \varepsilon n/3] + \Pr_G[|I| \geq \varepsilon n/3] \\ &\leq Pr_G[\sum_{i\notin I} E_i \geq 2\varepsilon n/3 \mid |I| < \varepsilon n/3] + \Pr_G[|I| \geq \varepsilon n/3] \\ &\leq 2^{-\Omega(\varepsilon^2 n)} + \Pr_G[|I| \geq \varepsilon n/3] \qquad \text{(by the Chernoff bound)} \\ &\leq 2^{-\Omega(\varepsilon^2 n)} \qquad \text{(by Lemma 4)} \end{aligned}$$

□

To finish the proof of proposition 3, we argue that algorithm *Amplify* outputs the correct answer with high probability. First, observe that the algorithm needs to know the correlation between x and x'; we try all possible n values for this correlation. (In fact, it is sufficient to try $O(1/\varepsilon)$ approximate values.) Then proposition 3 follows by combining Lemma 1 and Lemma 3 with $\gamma = \varepsilon^{-Dd}$ for a sufficiently large constant D.

5 From Almost Correct to Correct

In this section, we show that if we start with an almost correct assignment, $f_{G,P}(x)$ can be inverted for any nontrivial predicate P, provided that the constraint to variable ratio $m/n = D$ is a sufficiently large constant (depending on d). Our proofs are an adaptation of known algorithms for planted random 3SAT [Fla03, KV06].

Proposition 4. *Let K be a sufficiently large constant and P be an arbitrary nonconstant predicate. Suppose $D > Kd^6 4^d$. There exists a polynomial-time algorithm such that for a $1 - 2^{-\Omega(d^2 4^d)}$ fraction of $x \in \{0,1\}^n$ and with high probability over the choice of G, on input G, P, $f_{G,P}(x)$, and $x' \in \{0,1\}^n$ that has correlation $1 - 1/Kd2^dD$ with x, outputs an inverse for $f_{G,P}(x)$.*

Together with propositions 1 and 2, we have proved theorem 1. With proposition 3, we have proved theorem 2.

The algorithm has three stages. In the first stage, the objective is to come up with an assignment that matches most "core" variables of x. Roughly speaking, the *core* of G with respect to the assignment x is the set of those variables that occur regularly in G, in the sense that their presence in various types of constraints of G occurs within a small error of the expectation. The core will comprise most of the variables of x. In the second stage, some of the variables are unassigned. At the end of this stage, all assigned variables are assigned as in x, and all core variables are assigned. In the third stage, an assignment for the remaining variables is found by brute force. (The final assignment may not be x, as there are likely to be many possible inverses for $f_{G,P}(x)$.)

Due to space constraints we defer the proof of proposition 4 to the full version of the paper.

Acknowledgment

We thank Eyal Rozenman for useful discussions at the initial stages of this work. Both of the authors' work was supported in part by the National Natural Science Foundation of China Grant 60553001, and the National Basic Research Program of China Grant 2007CB807900,2007CB807901.

References

[AIK04] Applebaum, B., Ishai, Y., Kushilevitz, E.: Cryptography in NC^0. In: Proceedings of the 45th Annual Symposium on Foundations of Computer Science, pp. 166–175 (2004)

[AIK06] Applebaum, B., Ishai, Y., Kushilevitz, E.: On pseudorandom generators with linear stretch in NC^0. In: Díaz, J., Jansen, K., Rolim, J.D.P., Zwick, U. (eds.) APPROX 2006 and RANDOM 2006. LNCS, vol. 4110, pp. 260–271. Springer, Heidelberg (2006)

[Bra09] Braverman, M.: Polylogarithmic independence fools AC^0. Technical Report TR09-011, Electronic Colloquium on Computational Complexity (ECCC) (2009)

[CEMT09] Cook, J., Etesami, O., Miller, R., Trevisan, L.: Goldreich's one-way function candidate and myopic backtracking algorithms. In: Proceedings of the 6th Theory of Cryptography Conference (TCC), pp. 521–538 (2009)

[Coj06] Coja-Oghlan, A.: An adaptive spectral heuristic for partitioning random graphs. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4051, pp. 691–702. Springer, Heidelberg (2006)

[Fla03] Flaxman, A.: A spectral technique for random satisfiable 3CNF formulas. In: SODA 2003: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms, Baltimore, Maryland, pp. 357–363 (2003)

[Gol00] Goldreich, O.: Candidate one-way functions based on expander graphs. Technical Report TR00-090, Electronic Colloquium on Computational Complexity (ECCC) (2000)

[KV06] Krivelevich, M., Vilenchik, D.: Solving random satisfiable 3CNF formulas in expected polynomial time. In: SODA 2006: Proceedings of the seventeenth annual ACM-SIAM symposium on discrete algorithms, pp. 454–463. ACM Press, New York (2006)

[LN90] Linial, N., Nisan, N.: Approximate inclusion-exclusion. Combinatorica 10(4), 349–365 (1990)

[MST03] Mossel, E., Shpilka, A., Trevisan, L.: On ϵ-biased generators in NC^0. In: Proceedings of the 44th Annual Symposium on Foundations of Computer Science, pp. 136–145 (2003)

[SS85] Schmidt, J.P., Shamir, E.: Component structure in the evolution of random hypergraphs. Combinatorica 5(1), 81–94 (1985)

[Vil07] Vilenchik, D.: It's all about the support: a new perspective on the satisfiability problem. Journal on Satisfiability, Boolean Modeling, and Computation 3, 125–139 (2007)

Appendix: A Sampling Lemma

Lemma 4. *Fix $\varepsilon < 1/2$ and suppose $D > 2\log(1/\varepsilon)$. Let $N_1, \ldots, N_n$ be random variables taking values in the set $\{0, \ldots, Dn\}$ sampled uniformly conditioned on $N_1 + \cdots + N_n = Dn$. Then with probability $2^{-\Omega(\varepsilon Dn)}$, fewer than εn of the variables take value less than $D/2$.*

Proof. Let I denote the set of those i such that $N_i < D/2$. By a union bound, the probability of $|I| \geq \varepsilon n$ is at most $\binom{n}{\varepsilon n}$ times the probability that $N_1, \ldots, N_{\varepsilon n} < D/2$. We argue that for every i,

$$\Pr[N_i < D/2 \mid N_1, \ldots, N_{i-1} < D/2] = 2^{-\Omega(D)}$$

from where the claim follows. To show this, observe that conditioned on $N = N_1 + \cdots + N_{i-1}$, N_i is a sum of $(Dn - N)$ independent Bernoulli random variables with probability $1/(n-i)$ each. If $N_1, \ldots, N_{i-1} < D/2$, then the conditional expectation of N_i is at least D. By Chernoff bounds, the conditional probability that $N_i < D/2$ is then at most $2^{-\Omega(D)}$. □