Lecture 9 - One Way Permutations

Boaz Barak

October 17, 2007

From time immemorial, humanity has gotten frequent, often cruel, reminders that many things are easier to do than to reverse. Leonid Levin

Quick Review Homework 2.

Minimizing assumptions Up to now, we always assumed the following **PRG Axiom**: There exists a pseudorandom generator mapping $\{0,1\}^n$ to $\{0,1\}^{n+1}$.

In other words, we believe that there is an algorithm G such that the following task *cannot* be done: distinguish $G(U_n)$ from U_{n+1}.

Since we still don't have a *proof* that this cannot be done, our only evidence that a task is hard is that many people tried many approaches to solve it and didn't succeed.

The problem is that while people have been studying algorithms in one form or another for thousands of years, there hasn't been as much attention devoted to distinguishing the output of a function from the uniform distribution. Therefore, we want to have an assumption that a more natural task, such as computing a function, is hard to do.

Def We say that a function $g : \{0,1\}^* \rightarrow \{0,1\}^*$ is *hard to compute* (in the average case) if for every polynomial-time A, polynomially-bounded ϵ and large enough n

$$\Pr_{x \leftarrow_R \{0,1\}^n}[A(x) = g(x)] < \epsilon(n)$$

Claim: There exists a hard to compute function g such that g maps $\{0,1\}^n$ to $\{0,1\}^n$ for every n.

Proof: Just pick g at random. For every particular $2^{\sqrt{n}}$-time algorithm A, the expected number of inputs on which $A(x) = g(x)$ is one, and the probability that A computes g successfully on an at least $2^{-n/10}$ fraction

of the total 2^n inputs can be shown to be less than $2^{-2^{-n/2}}$. But a $2^{\sqrt{n}}$ algorithm can be described by about $2^{\sqrt{n}} \ll 2^{n/2}$ bits and so the total number of such algorithms is much smaller than $2^{2^{n/2}}$. □

One-way permutation Of course the mere existence of a hard function is not useful for cryptography. But the following assumption will be useful:

The OWP Axiom: There exists a polynomial-time function $f : \{0,1\}^* \to \{0,1\}^*$ such that for every n, f is a permutation over $\{0,1\}^n$ (i.e., maps $\{0,1\}^n$ to $\{0,1\}^n$ in a one-to-one and onto way) and such that the function $g = f^{-1}$ is hard to compute. Such an f is called a *one-way permutation.*

An equivalent condition is that for every poly-time A, poly-bounded ϵ and large enough n

$$\Pr_{x \leftarrow_R \{0,1\}^n}[A(f(x)) = x] < \epsilon(n)$$

(can you see why these are equivalent?)

We will prove the following theorem:

Theorem 1. *The OWP Axiom implies the PRG Axiom.*

This places the PRG Axiom on a much more solid foundation, since (as alluded by Levin's quote), this is the kind of task people have tried and failed to do for centuries. (Note that in cryptography we actually put our failures to good use!)

One way functions It is known that the PRG Axiom is implied by an even weaker assumption - the existence of a *one way function*, defined as a polynomial-time function f (not necessarily a permutation) such that for every poly-time A, poly-bounded ϵ and large enough n,

$$\Pr_{x \leftarrow_R \{0,1\}^n}[A(f(x)) = w \text{ s.t. } f(w) = f(x)] \leq \epsilon(n)$$

The assumption that one-way functions exist is *minimal* for many cryptographic tasks. It can be shown that the existence of pseudorandom generators, encryptions with key shorter than message, message authentication code implies the existence of one-way functions.

Proof of Theorem 1 Theorem 1 will follow from the following two theorems:

Theorem 2 (Yao's Theorem)**.** *A distribution X over $\{0,1\}^m$ is pseudorandom if and only if it is* unpredictable*, where the latter means that for every $i \in [m]$, poly-time A and poly-bounded ϵ,*

$$\Pr_{x \leftarrow_R X}[A(x_1, \ldots, x_{i-1}) = x_i] \leq 1/2 + \epsilon(n)$$

Theorem 3 (Goldreich-Levin)**.** *Let f be a one-way permutation. Then the following distribution is unpredictable:*

$$f(x), r, \langle x, r \rangle$$

where $x, r \leftarrow_R \{0,1\}^n$ and $\langle x, r \rangle \stackrel{def}{=} \sum x_i r_i \pmod 2$.

Theorems 2 and 3 together imply that if f is a one-way permutation then the function $x, r \mapsto f(x), r, \langle x, r \rangle$ is a pseudorandom generator mapping $\{0,1\}^{2n}$ to $\{0,1\}^{2n+1}$.

Proof of Theorem 2 One direction (pseudorandomness implies unpredictability) is easy and left as an exercise.

For the other direction, to show that if X is unpredictable then it is pseudorandom, we define the following $m+1$ hybrid distributions: H^i is the first i bits of X concatenated with $m-i$ uniform random bits.

It suffices to prove that for every i, $H^{i-1} \approx H^i$. We do this by reduction using the following claim:

Claim 4. *Suppose that there is an algorithm D such that*

$$\left|\Pr[D(H^i) = 1] - \Pr[D(H^{i-1}) = 1]\right| \geq \epsilon \tag{1}$$

then, there is an algorithm P with almost the same running time, such that

$$\Pr_{x \leftarrow_R X}[P(x_1, \ldots, x_{i-1}) = x_i] \geq 1/2 + \epsilon$$

The claim clearly proves the theorem (exercise).

Proof of Claim 4. We can drop without loss of generality the absolute value in (1), since if D satisfies this condition with a negative number inside the absolute value, then $\overline{D}$ will satisfy it with a positive number.

The algorithm P will do the following on input $x_1, \ldots, x_{i-1}$:

1. Guess a value b for x_i, and choose also $m-i$ random bits $y_{i+1}, \ldots, y_m$.
2. Let $z = D(x_1, \ldots, x_{i-1}, b, y_{i+1}, \ldots, y_m)$.

3. If $z = 1$ then output b; otherwise, output $1 - b$.

Analysis: The intuition behind the analysis is that if we guessed correctly then we're in H^i situation, where we're more likely to get the $z = 1$ output.

The actual analysis is the following:

Let p be the probability that $D(H^{i-1}) = 1$ and $p+\epsilon$ the probability that $D(H^i) = 1$.

We know that $\Pr[z = 1] = p$.

On the other hand we know that $\Pr[z = 1|b = x_i] \geq p + \epsilon$.

This means that

$$p = \Pr[z = 1] = \tfrac{1}{2}\Pr[z = 1|b = x_i] + \tfrac{1}{2}\Pr[z = 1|b = 1 - x_i]$$

implying that $\Pr[z = 1|b = (1 - x_i)] \leq p - \epsilon$.

So, the probability we output a correct answer is:

$$\tfrac{1}{2}\Pr[z = 1|b = x_i]+\tfrac{1}{2}(1-\Pr[z = 1|b = 1-x_i]) \geq \tfrac{1}{2}(p+\epsilon)+\tfrac{1}{2}(1-p+\epsilon) = \tfrac{1}{2}+\epsilon$$

□

Proof of Theorem 3 Theorem 3 will follow from the following lemma (exercise):

Lemma 5. *There is a poly$(n, 1/\epsilon)$-time algorithm that given oracle access to an oracle A that computes the function $r \mapsto \langle x, r\rangle$ with probability $1/2 + \epsilon$ over the choice of r, outputs x with probability at least $\left(\frac{\epsilon}{100n}\right)^2$.*

Proof of Lemma 5 The proof of Lemma 5 is a bit involved, so we will do it step by step.

The errorless case Suppose that we had a perfect oracle A that computed $r \mapsto \langle x, r\rangle$ with probability 1. Then, we could recover the first bit of x by outputting $A(r) \oplus A(r \oplus e^1)$ for some r (where e^1 is the vector with all zeroes except at the first location).

Note that

$$A(r) \oplus A(r \oplus e_1) = \langle x, r\rangle \oplus \langle x, r \oplus e^1\rangle =$$
$$\sum x_i r_i + \sum x_i(r_i \oplus e_i^1) = \sum x_i r_i + \sum x_i r_i + \sum x_i e_i^1 = x_1$$

The small error case Suppose that the oracle A was correct with probability 0.9 over the choice of r. Then because for a random r, $r \oplus e^1$ is uniformly distributed, we can use the union bound to show that the probability we get an incorrect answer when asking $A(r)$ and $A(r \oplus 1)$ is at most 0.2. (Note that these questions are dependent but the union bound still works in this case.)

Therefore, if we choose a random r, then with probability at least 0.8, $A(r) \oplus A(r \oplus e^1)$ will give us the first bit of x, and we can amplify this probability to $1/(10n)$ by making $10 \log n$ repetitions and taking the majority vote. In this way, we can recover *all* of the bits of x with high probability.

This will work as long as A is correct with probability more than $\frac{3}{4}$, but when A is correct with probability, say, 0.7, this analysis doesn't help us at all. The union bound will only say that we get the right value with probability at least 0.4 - worse than random guessing!

The full case The full case is when the oracle A is only guaranteed to be correct with probability $1/2 + \epsilon$. We will do that case on Thursday.