Lecture 9 - One Way Permutations

Boaz Barak

October 18, 2007

Reading Katz-Lindell Section 6.3 (Goldreich-Levin proof, definition of hardcore bits).

Quick Review Last time we started the proof of the following theorem:

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

Where the OWP Axiom is that there exists an efficiently computable f that is a permutation over $\{0,1\}^n$ for very n, and such that for every poly-time A, poly-bounded ϵ, and large enough n

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

We showed that it follows 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 $\langle x, r \rangle \stackrel{def}{=} \sum x_i r_i \pmod 2$.

We proved Theorem 2 and showed that Theorem 3 is implied by the following lemma:

Lemma 4. *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 $\left(\frac{\epsilon}{100n}\right)^2$.*

Today we will prove Lemma 4.

Quick review of probability Union bound, Chernoff bound, Chebychev bound.

Review: the low error case Recall that we said that if $\Pr_r[A(r) = \langle x, r \rangle] \geq 0.9$, then we can recover the i^{th} bit of x by choosing $r^1, \ldots, r^K$ at random ($K \geq 1000 \log n$ will do) and taking the majority of $A(r^1) \oplus A(r^1 \oplus e^i), \ldots, A(r^K) \oplus A(r^K \oplus e^i)$.

The analysis of this uses the following facts:

1. If r is chosen uniformly at random then $r \oplus e^i$ is also uniformly distributed.
2. Therefore, $\Pr[A(r) \neq \langle x, r \rangle] \leq 0.1$ and $\Pr[A(r \oplus e^i) \neq \langle x, r \oplus e^i \rangle] \leq 0.1$, implying by the union bound that $\Pr[A(r) = \langle x, r \rangle \text{ AND } A(r \oplus e^i) = \langle x, r \oplus e^i \rangle] \geq 0.8$. Thus, with probability at least 0.8, $A(r) \oplus A(r \oplus e^i) = x_i$.
3. Using the Chernoff bound, if we repeat this for K independently chosen random $r^1, \ldots, r^K$ then the probability that the majority of the values $A(r^j) \oplus A(r^j \oplus e^i)$ will be different from x_i is at most $2^{-K/1000}$.

   The reason is that the Chernoff bound guarantees that if $X_1, \ldots, X_K$ are independent random 0/1 variables with $\Pr[X_j] = p$, then

   $$\Pr\big[|\sum_j X_j - pK| > \epsilon pK\big] \leq 2^{-\epsilon^2 pK/5}$$

   Letting X_j be the random variable that is equal to 1 if both $A(r^j)$ and $A(r^j \oplus e^i)$ are correct we get the result.
4. This means that if we choose $K > 10^4 \log n$, then the probability we get the correct value for the i^{th} bit is at least $1 - \frac{1}{10n}$. Using the union bound, this means that with probability at least 0.9 we get the correct value for *all* of the bits.

Extending the analysis to the higher error case Suppose now that $A(r)$ is only correct with probability $1/2 + \epsilon$. In this case we can no longer argue that with probability better than 1/2, both $A(r)$ and $A(r \oplus e^i)$ are correct. However, note the following (seemingly useless) observation:

If someone gave us the values of $z_1 = \langle x, r^1 \rangle, \ldots, z_K = \langle x, r^K \rangle$ for $K = \frac{100}{\epsilon^n} \log n$ randomly chosen strings $r^1, \ldots, r^K$ then we could run the algorithm above to deduce all the bits of x. The reason is that since $\Pr[A(r \oplus e^i) = \langle x, r \oplus e^i \rangle] \geq 1/2 + \epsilon$, the Chernoff bound implies that the i^{th} bit of z is equal to the majority of $z_j \oplus A(r^j \oplus e^i)$ with probability at least $1 - \frac{1}{10n}$.

Using pairwise independence Another observation is that we could still run the same algorithm if someone gave us the values of $z_1 = \langle x, r^1 \rangle, \ldots, z_K = \langle x, r^K \rangle$ for $K = \frac{10n}{\epsilon^2}$ strings that are chosen from a *pairwise independent distribution*.

By pairwise independent we mean that each r^j is has the uniform distribution and for every $i \neq j$, the random variables r^i and r^j are independent, *but* it's not necessarily the case that for a triple $i, j\ell$, the random variables r^i, r^j, r^ℓ are independent.

The reason we can still carry through the analysis is that if we define X_j to be the random variable that is 1 if $A(r^j \oplus e^1)$ is correct and 0 otherwise, then we know that $\mathbb{E}[X_j] \geq \frac{1}{2} + \epsilon$, and that the variables $X_1, \ldots, X_K$ are pairwise independent, and hence $Var(X_1 + \ldots + X_K) = Var(X_1) + \ldots + Var(X_K) \leq K$. (Note that $\mathbb{E}[X_1 + \ldots + X_k] = \sum_{j=1}^K \mathbb{E}[X_j] \geq (1/2 + \epsilon)K$.)

It follows that by the Chebychev Inequality

$$\Pr\Big[\text{majority value incorrect}\Big] \leq \Pr\left[\Big|\sum_j X_j - \mathbb{E}[\sum_j X_j]\Big| \geq \epsilon K = \epsilon\sqrt{K}\sqrt{K}\right] \leq \frac{\epsilon}{K}$$

Meaning that for $K > \frac{10n}{\epsilon^2}$, this probability is less than $\frac{1}{10n}$.

Getting these values How do we get these magical values $z_1, \ldots, z_K$? One way is to just guess them but this will be successful with probability 2^{-K} which is far too small.

The crucial observation is the following lemma:

Lemma 5. *Let $K = 2^k - 1$ and identify every number j between 1 and K with a non-empty subset S_j of $[k]$. Consider the following distribution $r^1, \ldots, r^K$ over $\{0,1\}^n$: first $s^1, \ldots, s^k$ are chosen independently at random in $\{0,1\}^n$, then we define $r^j = \sum_{i \in S_j} s_i$ (where the sum is done componentwise modulo 2).*

Then $r^1, \ldots, r^K$ are pairwise independent.

Once we have Lemma 5 we're done. The reason is that we can choose $k = \log(\frac{10n}{\epsilon^2}) + 1$ strings $s^1, \ldots, s^k$ at random and guess values $y_1, \ldots, y_k$, hoping that $y_i = \langle x, s^i \rangle$. We will be correct with probability $2^{-k} = \frac{\epsilon^2}{20n}$. Now, identifying the numbers between 1 and $K = \frac{10n}{\epsilon^2}$ with the non-empty subsets of $[k]$, define for every $j \in [K]$,

$$r^j = \sum_{i \in S_j} s^i$$

then we can set $\langle x, r^j \rangle = \sum_{i \in S_j} \langle x, s^i \rangle$ and hence we have a collection of K pairwise independent strings $r^1, \ldots, r^K$ for which we know the values $\langle x, r^j \rangle$ for every j!

Proof of Lemma 5 We need to show that for every $i \neq j$ and strings $z, w \in \{0,1\}^n$, $\Pr[r^i = z \text{ AND } r^j = w] = 2^{-2n}$.

In other words, we need to show that for every distinct pair of non-empty sets U, V

$$\Pr[\sum_{u \in U} s_u = z \text{ AND } \sum_{v \in V} s_v = w] = 2^{-2n}$$

We'll demonstrate this for the pair $U = 1, 2, 3$ and $V = 1, 2$. That is, we need to show that if we pick s_1, s_2, s_3 independently at random, then the probability that the following pair of equations are satisfied is exactly 2^{-2n}.

$$\begin{aligned} s_1 + s_2 + s_3 &= z \\ s_1 + s_2 &= w \end{aligned}$$

(If you know some linear algebra you can see this is the case because the two equations are linearly independent.)

Fix any choice for s_1. We will prove that there is a unique pair s_2, s_3 that satisfy

$$\begin{aligned} s_2 + s_3 &= z - s_1 \\ s_2 &= w - s_1 \end{aligned}$$

but this is immediate from the equations. □

Conclusion As a conclusion we get that the function $x, r \mapsto f(x) \| r \| \langle x, r \rangle$ is a pseudorandom generator.

Hard-core bits We can abstract the essence of the Goldreich-Levin theorem as follows: define a *hard-core* bit for a one-way function or permutation $g : \{0,1\}^* \to \{0,1\}^*$ to be a function $h : \{0,1\}^* \to \{0,1\}$ such that for every poly-time A and poly-bounded ϵ,

$$\Pr_{x \leftarrow_R \{0,1\}^n}[A(g(x)) = h(x)] \leq \tfrac{1}{2} + \epsilon(n)$$

The Goldreich-Levin Theorem says that if there exists a one-way permutation f, then there exists a different one-way permutation g (namely, $g(x,r) = f(x)\|r$) that has a hardcore bit h (namely, $h(x,r) = \langle x, r\rangle$). Thus it is often known as the theorem that every one-way function has a hardcore bit.

Commitment Schemes One use that we may like for a digital envelope is the ability to commit in advance to some value. For example, suppose I bet you a million dollar that I can predict the winner of American Idol. Now I don't want to tell you my prediction since you'd have considerable financial incentive to try to effect the competition's outcome. On the hand, you'd probably want me to *commit* in advance to my prediction (i.e., you won't be too happy with a protocol where after the results are known I'd tell you whether or not this was the winner I predicted.)

In the physical world, we might try to solve this problem by me writing the prediction in an envelope and putting the envelope in a safe (ideally, guarded by both of us). The digital analog for that is a *commitment*.

Definition 6 (Commitment schemes)**.** A *commitment scheme* C is an *unkeyed* function that takes two inputs: a plaintext $x \in \{0,1\}^\ell$ and randomness r (chosen in $\{0,1\}^n$). The idea is that to commit to the winner I let x be my prediction (e.g. $x =$`'Fantasia'`), choose $r \leftarrow_R \{0,1\}^n$ and publish $y = \mathsf{C}(x,r)$. Later to prove I predicted x, I will publish x and r.

A commitment scheme should satisfy the following two properties:

Hiding / Secrecy / Indistinguishability For every $x, x' \in \{0,1\}^\ell$, $\mathsf{C}(x, U_n)$ is computationally indistinguishable from $\mathsf{C}(x', U_n)$. (Note this is the same as the indistinguishability property for encryption scheme, and implies that given $y = \mathsf{C}(x, U_n)$ an adversary can't learn any new information about x.)

Binding For every y there exists at most a *single* x such that $y = \mathsf{C}(x,r)$ for some $r \in \{0,1\}^n$. (This implies that it is not possible to come up with two different pairs x, r and x', r' with $x \neq x'$ that yield y.)

Why not encryption? You might be wondering why do we need to use a new primitive: why don't I simply *encrypt* the plaintext and give you the encryption. The problem with this approach is that an encryption does not necessarily bind me to a single value. As an example, consider the one-time-pad encryption: I can give you a random string y. Then, if the winner is Fantasia I will give you $x =$`'Fantasia'`, $k = x \oplus y$ and claim that initially I encrypted x with the key k to get y. If the winner is Eva I will give you $x' =$`'Eva'`, $k' = x' \oplus y$ and claim I initially encrypted x' with the key k' to get y. You have no way to dispute this claim.

Another application. Another, perhaps more plausible application for commitment schemes is to arrange close bids. Suppose I am a government agency that wants to award a contract

to the lowest bidder. One way to arrange this is to have all bidders send their bids to the agency, but then perhaps an unscrupulous worker can leak the bid of one company to a different company. Instead, all bidders can send a *commitment* to their bid to the agency, and only after all bids have been received will they send the randomness needed to open the commitment.

We will see more applications for commitment schemes later in the course.

Constructing commitments The first observation is that to construct a commitment to strings of length ℓ, it is enough to construct a commitment to single bits. The reason is if I have a single-bit commitment then to commit to a string $x = x_1 \cdots x_\ell$ I will simply commit to each bit separately (using of course independent randomness for each bit). The security of this scheme is left as an exercise.

Let $f : \{0,1\}^n \to \{0,1\}^n$ be a one-way permutation and $h : \{0,1\}^n \to \{0,1\}$ be a hard-core bit for $f(\cdot)$. To commit to a bit b, I will choose $r \leftarrow_{\mathrm{R}} \{0,1\}^n$, and let $\mathsf{C}(b,r) = f(r), h(r) \oplus b$.

Theorem 7. *The function $\mathsf{C}(b,r) = f(r), h(r) \oplus b$ is a secure commitment scheme.*

Proof. (The following proof is a bit sketchy, and it's a good exercise for you to fill in the details.)

Binding Given $y = y', c$ there is a single r such that $y' = f(r)$. Thus, this r determines completely whether y is a commitment to 0 (in which case $c = h(r)$) or a commitment to 1 (in which case $c = \overline{h(r)}$).

Hiding We need to prove that $f(r), h(r) \oplus 0$ is indistinguishable from $f(r), h(r) \oplus 1$. However, $f(r), h(r)$ is indistinguishable from U_{n+1} and U_{n+1} with the last bit flipped is the same distribution as U_{n+1}.

□

Coin tossing over the phone