CS 830: Randomness and Complexity 02/10/05

Lecture 8: Hardcore-bit, Pseudorandom Generators and Discrete Log

instructor: Jin-Yi Cai Scribe: Anand Kumar Sinha, Vinay Choudhary

1 Introduction

Towards the end of last lecture, we stated three different ways to construct a predictor S^A using a distinguisher A. Here we go over that again, and give the proofs. Then we move on to pseudorandom generators. We show how to construct a stretching pseudorandom generator using the hardcore-bit. Finally we have a look at the discrete log problem and quadratic residues.

2 Hardcore Bit

We considered following three methods of using a distinguisher A to construct a predictor S^A. We'll only describe the relevant part here. For details, please refer to the last lecture.

1. Pick $u, v \in_u \{0,1\}$. If $A(f(X), u, z) = 1$, then output u, o/w output v.
2. Pick $u, v \in_u \{0,1\}$. If $A(f(X), u, z) = 1$, then output u, o/w output $\overline{u}$.
3. Pick $u, v \in_u \{0,1\}$. Run $A(f(X), 0, z)$ and $A(f(X), 1, z)$. If answer is $(1, 0)$, output 0. If answer is $(0, 1)$, output 1. Otherwise output u.

Note that (3) is the most intuitive approach. It is easy to see that (2) and (3) are equivalent. If the answer of $A(f(X), 0, z), A(f(X), 1, z)$ is $(1, 0)$, then (2) will always output 0, irrespective of the value of u. If it was $(0, 1)$, then it'll always output 1. If it is $(0, 0)$ (or $(1, 1)$), output will be $\overline{u}$ (or u) i.e. 0 or 1 with equal probability.
Statements (1) and (2) are equivalent only up to a factor of $\frac{1}{2}$ as shown by the following analysis.

1.

$$\begin{aligned}
Pr[S^A = X.Z] &= Pr[A(f(X), u, Z) = 1 \wedge u = X.Z] + Pr[A(f(X), u, Z) = 0 \wedge v = X.Z] \\
&= \frac{1}{2} Pr[A(f(X), u, Z) = 1 | u = X.Z] + \frac{1}{2} Pr[A(f(X), u, Z) = 0 | v = X.Z] \\
&= \frac{1}{2} Pr[A(f(X), X.Z, Z) = 1] + \frac{1}{2} Pr[A(f(X), b, Z) = 0] \\
&= \frac{1}{2} Pr[A(f(X), X.Z, Z) = 1] + \frac{1}{2}(1 - Pr[A(f(X), b, Z) = 1])
\end{aligned}$$

Let E_d be the event that $A(f(X), X.Z, Z) = 1$ and E_b be the event $A(f(X), b, Z) = 1$. Let $\mu_d = \mu(E_d)$ and $\mu_b = \mu(E_b)$. Then

$$\begin{aligned} Pr[S^A = X.Z] &= \frac{1}{2}\mu_d + \frac{1}{2}(1 - \mu_b) \\ &= \frac{1}{2} + \frac{1}{2}(\mu_d - \mu_b) \\ &= \frac{1}{2} + \frac{\delta(n)}{2} \end{aligned}$$

2.

$$\begin{aligned} Pr[S^A = X.Z] &= Pr[A(f(X), u, Z) = 1 \wedge u = X.Z] + Pr[A(f(X), u, Z) = 0 \wedge \overline{u} = X.Z] \\ &= \frac{1}{2}Pr[A(f(X), u, Z) = 1 | u = X.Z] + \frac{1}{2}Pr[A(f(X), u, Z) = 0 | \overline{u} = X.Z] \\ &= \frac{1}{2}Pr[A(f(X), X.Z, Z) = 1] + \frac{1}{2}Pr[A(f(X), \overline{X.Z}, Z) = 0] \\ &= \frac{1}{2}Pr[A(f(X), X.Z, Z) = 1] + \frac{1}{2}(1 - Pr[A(f(X), \overline{X.Z}, Z) = 1]) \end{aligned}$$

Let $E_{\overline{d}}$ be the event that $A(f(X), \overline{X.Z}, Z) = 1$ and let $\mu_{\overline{d}} = \mu(E_{\overline{d}})$. We have,

$$\begin{aligned} \mu_b &= Pr[A(f(X), b, Z) = 1] \\ &= \frac{1}{2}Pr[A(f(X), b, Z) = 1 | b = X.Z] + \frac{1}{2}Pr[A(f(X), b, Z) = 1 | b = \overline{X.Z}] \\ &= \frac{1}{2}(\mu_d + \mu_{\overline{d}}) \\ \mu_{\overline{d}} &= 2\mu_b - \mu_d \end{aligned}$$

Hence,

$$\begin{aligned} Pr[S^A = X.Z] &= \frac{1}{2} + \frac{1}{2}(\mu_d - \mu_{\overline{d}}) \\ &= \frac{1}{2} + (\mu_d - \mu_b) \\ &= \frac{1}{2} + \delta(n) \end{aligned}$$

3 Stretching Pseudorandom Generator using Hardcore Bit

3.1 Stretching by one bit

THEOREM 3.1. *Let $f : \{0,1\}^n \to \{0,1\}^n$ be a $\frac{1}{w(n)}$-weak one way permutation. Let $g : \{0,1\}^n \to \{0,1\}^{n+1}$ be defined as $(X, Z) \mapsto (f(X), X.Z, Z)$. Then g is a $\frac{1}{w(n)}$ pseudorandom generator that stretches by one bit.*

Proof. Suppose Not. Then there exists a poly-time distinguisher A such that

$$Pr_{Y\in_u\{0,1\}^{n+1},Z\in_u\{0,1\}^n}[A(Y,Z)=1] - Pr_{X,Z\in_u\{0,1\}^n}[A(g(X,Z))=1] \quad \geq \quad \frac{1}{w(n)} \tag{1}$$

Since f is a one-way permutation, if X is distributed uniformly over $\{0,1\}^n$, then $f(X)$ is distributed uniformly over $\{0,1\}^n$. So

$$\begin{aligned} Pr_{Y\in_u\{0,1\}^{n+1},Z\in_u\{0,1\}^n}[A(Y,Z)=1] &= Pr_{X,Z\in_u\{0,1\}^n,b\in_u\{0,1\}}[A(X,b,Z)=1] \\ &= Pr_{X,Z\in_u\{0,1\}^n,b\in_u\{0,1\}}[A(f(X),b,Z)=1] \end{aligned}$$

Combining this with (1), we get

$$Pr_{X,Z\in_u\{0,1\}^n,b\in_u\{0,1\}}[A(f(X),b,Z)=1] - Pr_{X,Z\in_u\{0,1\}^n}[A(g(X,Z))=1] \quad \geq \quad \frac{1}{w(n)}$$

We have a distinguisher. From this we can construct a predictor for the inner-product bit which is not possible since f is one-way and the inner-product bit cannot be predicted. □

3.2 Stretching the output of a pseudorandom generator

In this section, we will show how to construct a pseudorandom generator that stretches by a polynomial number of bits from a pseudorandom generator that stretches by one bit. The notation $g(X)_j$ below stands for the first (most significant) j bits of $g(X)$.

THEOREM 3.2. *Let $g:\{0,1\}^n \to \{0,1\}^{n+1}$ be a pseudorandom generator that stretches by one bit (possibly using some additional public randomness). Define $g_0(X)=X$, $g_1(X)=g(X)$, and for all $i\geq 1$,*

$$g_{i+1}(X) = g(g_i(X))g_i(X)_{n+1}g_{i-1}(X)_{n+1}\dots g_1(X)_{n+1}$$

Then for any $l(n)=poly(n)$, $g_{l(n)}(X)$ is a $\frac{l(n)}{w(n)}$ pseudorandom generator that stretches by $l(n)$ bits.

Proof. Let $X \in_u \{0,1\}^n$. Let A be a breaking adversary (poly-time) for $g_{l(n)}(X)$ with success probability

$$\delta(n) = Pr[A(g_{l(n)}(X))=1] - Pr_{Y\in_u\{0,1\}^{l(n)}}[A(Y)=1]$$

We will construct a poly-time distinguisher S^A for g having success probability $\frac{\delta(n)}{l(n)}$.

1. **Input**: $Z\in\{0,1\}^{n+1}$
2. Pick $I\in_u\{1,\dots,l(n)\}$
3. Return $A(g_{i-1}(Z_{1,\dots,n})\circ Z_{n+1}\circ U_{l(n)-i})$.

For the proof, we will again consider a sequence of distributions.

$$
\begin{aligned}
D_0 &= U_{n+l(n)} \\
D_1 &= g_1(X) \circ U_{l(n)-1} \\
&\vdots \\
D_i &= g_i(X) \circ U_{l(n)-i} \\
&\vdots \\
D_{l(n)} &= g_{l(n)}(X)
\end{aligned}
$$

For each $i = 0, \ldots, l(n)$, let δ_i be the probability that A outputs 1 on an input randomly chosen according to D_i. Let $I \in_u \{1, \ldots, l(n)\}$. As in earlier proofs, because $\delta(n) = \delta_{l(n)} - \delta_0$, it follows that

$$
E_I[\delta_I - \delta_{I-1}] = \frac{\delta(n)}{l(n)}
$$

Hence, the success probability of S^A is $\frac{\delta(n)}{l(n)}$. It follows that $g_{l(n)}$ is $\frac{l(n)}{w(n)}$ pseudorandom generator. $\square$

Exercise: Prove that we don't need to "stretch" the $(n+1)$-st bit. We could take any other bit as well. In other words, composing g with a permutation on $\{0,1\}^{n+1}$ would still give a pseudorandom generator. $\boxtimes$

4 The Discrete Log and other problems

Let p be a prime. We know that the group $\mathbb{Z}_p^*$ is cyclic. Let g be a generator. The discrete log function is defined as $f(y) = x$ such that $y = g^x \mod p$ and $x \in \{1, \ldots, p-1\}$. It is believed that the discrete log is hard to compute.

$y = g^x$ is said to be a quadratic residue if x is even. We can find in poly-time whether a given number is a quadratic residue using the Legendre symbol.

$$
\left(\frac{x}{p}\right) = x^{\frac{p-1}{2}} \mod p
$$

We can also find the quadratic residue. The case when $p \equiv 3 \mod 4$ is easy. If $y = b^2$,

$$
\left(y^{\frac{p+1}{4}}\right)^2 = b^{p+1} = b^2 = y \mod p
$$

Exercise: Extend this to $p \equiv 1 \mod 8$ and $p \equiv 5 \mod 8$. $\boxtimes$

Exercise: Prove that in finite fields of order 2^n, we can compute square roots efficiently. $\boxtimes$

Finally, we present a randomised algorithm for computing square roots.

1. **Input**: y
2. **Output**: x s.t. $x^2 = y$
3. Pick $t \in_u \mathbb{Z}_p$
4. $x^2 - tx + y \in \mathbb{Z}_p[X]$.
5. Let $\Delta = t^2 - 4y$. Calculate $a = \left(\frac{\Delta}{p}\right)$
6. If $a = 1$, **repeat**.
7. If $a = 0$, we have got $t^2 = 4y \mod p$. Done.
8. If $a = -1$, let $u = (t + \sqrt{\Delta})/2$.
9. In $\mathbb{F}_p[\sqrt{\Delta}] = \mathbb{F}_{p^2}$, compute $v = u^{\frac{p+1}{2}}$.
10. **Return** $x = v$ as square root of $y \mod p$.

Proof.

$$\left(u^{\frac{p+1}{2}}\right)^2 = u^{p+1} = u\overline{u}$$

where $\overline{u} = u^{p+1}$ is the other root of the minimal polynomial for u over $\mathbb{F}_p$. And since u is a root of $x^2 - tx + y$, $u\overline{u} = y$. If y is a quadratic residue, $u^{\frac{p+1}{2}}$ has to be in the base field $\mathbb{F}_p$, and hence is the required square root of y. □

Exercise: Prove that the above algorithm succeeds with probability roughly $\frac{1}{2}$. ⊠