CS 600.442 – Modern Cryptography September 16, 2015

Lecture 5: Pseudo Random Generators

Instructor: Abhishek Jain *Scribe: Aditya Patil*

1 Recall

Steps for constructing PRG from OWF.

- Step 1: OWF (OWP) => Hardcore Predicate for OWF (OWP)
- Step 2: Hardcore Predicate for OWF (OWP) => One-bit stretch PRG
- Step 3: One-bit stretch PRG => Poly-stretch PRG

In this lecture, we will focus on the first step, namely, OWF (OWP) => Hardcore Predicate for OWF (OWP).

2 Construction of Hardcore Predicate

Theorem 1 (Hardcore Predicate from OWF) *If $f : \{0,1\}^n \longrightarrow \{0,1\}^n$ is a one way function, then:*

1. *$g : \{0,1\}^{2n} \longrightarrow \{0,1\}^{2n}$, where $g(x,r) := (f(x), r)$ is also a one way function*
2. *$h(x,r) := < x, r >$ is a hardcore predicate for $g(x,r)$*

Towards proving the security of the above construction, we start by assuming that the construction is not secure. That is, suppose that there exists an adversary $\mathcal{A}$ for h that predicts its output with non-negligible advantage. Given $\mathcal{A}$, we would like to construct an inverter $\mathcal{B}$ for the one-way function f, in order to reach a contradiction. Note that one main challenge is that one main challenge here is that $\mathcal{A}$ only outputs 1 bit (the output of h) and we want to somehow use it to build $\mathcal{B}$ that outputs n bits (a pre-image for f).

In what follows, we will consider to warmup proofs where we make some assumptions about the adversary A in order to build B. The full proof is left as an exercise.

2.1 Warmup Proof 1

Assumption 1 *Given $g(x,r) = (f(x), r)$, adversary $\mathcal{A}$ always outputs $h(x,r)$ correctly.*

Inverter $\mathcal{B}(f(x))$:

1. Compute $x_i^* \longleftarrow \mathcal{A}(f(x), e_i)$ for every $i \in [n]$ where:

$$e_i = (\underbrace{0, ..., 0}_{(i-1)-times}, 1, ..., 0)$$

2. Output $x^* = x_1^*...x_1^*$

It is easy to see that $\mathcal{B}$ is polynomial-time and successfully inverts $f(x)$ with probability 1.

2.2 Warmup Proof 2

Assumption 2 *Given $g(x,r) = (f(x), r)$, adversary $\mathcal{A}$ outputs $h(x,r)$ with probability $\frac{3}{4} + \epsilon(n)$ (over choices of (x,r))*

Note that once we relax Assumption **??** to Assumption **??**, we have the following two problems to overcome:

1. $\mathcal{A}$ may not work on all inputs

   Solution: To address this problem, we will define a set S s.t. when $x \in S$, $\mathcal{A}$ predicts $h(x,r)$ w.h.p. Further, we will show that set S is of large enough size.

   We define set as follows:

   $$S := \{x : Pr[r \overset{\$}{\leftarrow} \{0,1\}^n : \mathcal{A}(f(x), r) = h(x,r)] \geqslant \frac{3}{4} + \frac{\epsilon(n)}{2}\}$$

   Lemma 2 $Pr[x \in S] \geqslant \frac{\epsilon(n)}{2}$

   Proof. Suppose that the claim is false. Then, consider the following:

   $$\Pr[x, r \leftarrow \{0,1\}^n : \mathcal{A}(f(x), r) = h(x,r)]$$

   It follows that the above expression:

   $$\begin{aligned} &\leq Pr[x \in S] \cdot Pr[r \leftarrow \{0,1\}^n : \mathcal{A}(f(x), r) = h(x,r) | x \in S] \\ &+Pr[x \notin S] \cdot Pr[r \leftarrow \{0,1\}^n : \mathcal{A}(f(x), r) = h(x,r) | x \notin S] \\ &\leq \frac{\epsilon(n)}{2} \cdot 1 + (1 - \frac{\epsilon(n)}{2}) \cdot (\frac{3}{4} + \frac{\epsilon(n)}{2}) \\ &\leq \frac{3}{4} + \epsilon(n) \end{aligned}$$

   Which contradicts the definition of S. The lemma therefore holds.

2. Even if $\mathcal{A}$ predicts $h(x,r)$ with high probability for some y, but a random r; it may still fail on a particular $r = e$;

   Intuition for Solution. To address this problem, instead of querying $\mathcal{A}$ on $(f(x), e_i)$ as before, we will now break e_i in two parts and when $\mathcal{A}$ replies to both then combine them to get e_i. We want each of these two parts to look random individually, even though they are correlated. Towards that end, we will use the following fact about inner products (the proof is easy to verify).

   Lemma 3 $< a, b \oplus c > = < a, b > \oplus < a, c >$

Inverter $\mathcal{B}(f(x))$:

(a) For every $i \in [n]$, repeat the following steps:

i. Choose r_j, r'_j such that $r_j \xleftarrow{\$} \{0,1\}^n$ and $r'_j = r \oplus e_i$.
ii. Compute $a \leftarrow \mathcal{A}(f(x), r_j)$ and $b \leftarrow \mathcal{A}(f(x), r'_j)$.
iii. Compute $c_j = a_j \oplus b_j$

(b) Take majority over the c_j's to compute x^*_i.

(c) Output $x^* = x^*_1 ... x^*_1$

We first claim that c_j computed by $\mathcal{B}$ is the correct x_i with probability at least $\frac{1}{2} + \epsilon(n)$.

Lemma 4 $\Pr[c_j = x_i] \geq \frac{1}{2} + \epsilon(n)$

The proof for this lemma follows by a simple union bound. To see this, note that:

$$\Pr[\mathcal{A}(f(x), r_j) \neq h(x, r)] \leqslant \frac{1}{4} - \frac{\epsilon(n)}{2}$$

$$\Pr[\mathcal{A}(f(x), r'_j) \neq h(x, r)] \leqslant \frac{1}{4} - \frac{\epsilon(n)}{2}$$

Now, we have that:

$$\begin{aligned} \Pr[\text{Both answers of } \mathcal{A} \text{ are correct}] &= 1 - Pr[\text{One of them is wrong}] \\ &\geq 1 - (\frac{1}{4} - \frac{\epsilon(n)}{2} + \frac{1}{4} - \frac{\epsilon(n)}{2}) \\ &\geq \frac{1}{2} + \epsilon(n) \end{aligned}$$

This completes the proof of the lemma. Now, its easy to see that if $\mathcal{B}$ repeat steps (i)-(iii) sufficient (polynomial) number of times, then we can ensure that it predicts $x^*_i = x_i$ correctly with probability $1 - negl(n)$.

3 Food for Thought on PRGs

- OWF $\Rightarrow$ PRG: [Impagliazzo-Levin-Luby-89] and [Hastad-90]
- More Efficient Constructions: [Vadhan-Zheng-12]
- Computational analogues of Entropy
- Non-cryptographic PRGs and Derandomization: [Nisan-Wigderson-88]

4 Going beyond Poly Stretch

PRGs can only generate polynomially long pseudorandom strings. What if we want more pseudorandomness? In particular, how can we efficiently generate exponentially long pseudorandom strings?

To answer this question, we will build functions that index exponentially long pseudorandom strings.

5 Preliminaries for Pseudorandom Functions

Random Functions. Let $\mathcal{F}_n :=$ set of all functions that map inputs from $\{0,1\}^n$ to $\{0,1\}^{l(n)}$. A random function is $f \xleftarrow{\$} \mathcal{F}_n$. Here, note that $|\mathcal{F}_n| = 2^{\ell(n)\cdot 2^n}$.

Oracle Algorithms. Let $\mathcal{O}$ be any oracle that maps queries $q \in \{0,1\}^n$ to $\{0,1\}^{l(n)}$. We denote an algorithm $\mathcal{A}$ with oracle access to $\mathcal{O}$ as $\mathcal{A}^{\mathcal{O}}$.

We will use the convention that querying and receiving an answer from $\mathcal{O}$ takes unit time. Now using, this convention, one can devise definitions of PPT and n.u. PPT for oracle algorithms. In particular, a PPT oracle algorithm must only make polynomial number of queries to its oracle. The exact definitions are left as an exercise.

We now give two definitions that we will be useful in formalizing pseudorandom functions.

Definition 1 (Oracle Ensemble) *A sequence $\{\mathcal{O}_n\}_{n\in\mathbb{N}}$ is an oracle ensemble if $\forall n \in \mathbb{N}$, $\mathcal{O}_n$ is a distribution over the set of all functions $f : \{0,1\}^n \to \{0,1\}^{l(n)}$*

Definition 2 (Oracle Indistinguishability) *Oracle ensembles $\{\mathcal{O}_n^0\}$ and $\{\mathcal{O}_n^1\}$ are computationally indistinguishable if for every n.u. PPT oracle machine D, there exists a negligible function $\mu(.)$ s.t.:*

$$|Pr[f \leftarrow \mathcal{O}_n^0 : D^f(1^n) = 1] - Pr[f \leftarrow \mathcal{O}_n^1 : D^f(1^n) = 1]| \leqslant \mu(n)$$

6 Pseudorandom Functions

We now proceed to define pseudorandom functions (PRF). Intuitively, a PRF is an efficiently computable function that "looks like" a random function. We formalize this in the following manner:

Definition 3 *Pseudorandom Functions: A family of functions $\{f_s : \{0,1\}^n \to \{0,1\}^{l(n)}\}$ is a pseudorandom function (PRF) if:*

1. **Efficient Computation:** *There exists a PPT F s.t. $F(s,x)$ efficiently computes the function $f_s(x)$*
2. **Indistinguishability:**

$$\{s \xleftarrow{\$} \{0,1\}^n : f_s\} \approx \{f \xleftarrow{\$} \mathcal{F}_n : f\}$$

Remark 1 *Typically, l(n) will be equal to n*

7 PRF from PRG

In now turn to constructions of PRFs. We will consider the following goal:

Goal: Construct a PRF $\{f_s : \{0,1\}^n \to \{0,1\}^n\}$ a length-doubling PRG $G : \{0,1\}^n \to \{0,1\}^{2n}$

We end this lecture with the construction of PRF. The proof will be discussed in the next lecture.

Construction of f_s:

- $G(s) = G_0(s), G_1(s)$ where $G_0, G_1 : \{0,1\}^n \to \{0,1\}^n$
- $f_s(x) := G_{x_n}(G_{x_{n-1}}(\dots G_{x_1}(s)\dots))$