Lecture 12

In this lecture we continue our discussion of pseudorandom generators. We introduce a definition of one-way functions and show how to use these to create pseudorandom generators.

1 Pseudorandom Generators and One-Way Functions

First, we give formal definitions of the two cryptographic primitives.

Pseudorandom Generators

A pseudorandom generator uses a small random seed to create a longer output sequence that appears random: for random inputs, the distribution of outputs is computationally difficult to distinguish from the uniform distribution.

Definition 1 (Pseudorandom Generator) *A function $G : \{0,1\}^l \to \{0,1\}^m$ with $l < m$ is (S, ϵ)-indistinguishable from uniform if for every circuit boolean circuit A of size at most S,*

$$\left| \mathbf{Pr}_{z \in \{0,1\}^m}[A(z) = 1] - \mathbf{Pr}_{x \in \{0,1\}^l}[A(G(x)) = 1] \right| \leq \epsilon.$$

A function $G_l : \{0,1\}^l \to \{0,1\}^m$ is a pseudorandom generator if it is computable in poly(l) time and is $(S(l), \epsilon(l))$-indistinguishable from uniform, where $S(\cdot)$ and $1/\epsilon(\cdot)$ are superpolynomial.

In other words, if G is a pseudorandom generator, a size-bounded circuit can have only ϵ advantage in distinguishing the output of G from the uniform distribution on $\{0,1\}^m$.

One-Way Functions

A one-way function is easy to compute in one direction but difficult in the other.

Definition 2 (One-Way Function) *A function $f : \{0,1\}^n \to \{0,1\}^n$ is (S, ϵ)-one-way if for every circuit A of size a most S,*

$$\mathbf{Pr}_{x \in \{0,1\}^n}[f(A(f(x))) = f(x)] \leq \epsilon$$

A function $f_n : \{0,1\}^n \to \{0,1\}^n$ is a one-way function if it is computable in time poly(n) and f_n is $(S(n), \epsilon(n))$-one-way where $S(\cdot)$ and $1/\epsilon(\cdot)$ are superpolynomial.

This means that if f is a one-way function, no size-bounded circuit can use $f(x)$ to find x' such that $f(x') = f(x)$ for a large $(1 - \epsilon)$ fraction of x in the domain.

2 The Connection between PRG and OWF

Now that the primitives are defined, we will show how they are related.

OWF from a PRG

The following result is fairly easy and intuitive, but its proof gives an example of the use of a security-preserving reduction.

THEOREM 1
If there is a pseudorandom generator $G_l : \{0,1\}^l \to \{0,1\}^{2l}$, then there is a one-way function.

PROOF:
Define the candidate one-way function $f : \{0,1\}^{2l} \to \{0,1\}^{2l}$ as $f(x,y) = G(x)$.

It is clear that f is computable in polynomial time since it involves only a call to the polynomial-time computable G_l.

To prove that f is one-way, we propose an adversary A that is capable of inverting f. From this we create an adversary A' for G_l, and use the reduction to determine how the security parameters for f and G_l relate.

Suppose A is a circuit of size S which inverts f on a fraction ϵ of inputs. That is,

$$\mathbf{Pr}_{x\in\{0,1\}^n}[f(A(f(x))) = f(x)] = \epsilon.$$

Define a circuit A' that attempts to determine if its input $z \in \{0,1\}^{2l}$ is from the distribution of G_l:

$(x, y) := A(z)$
if $G_l(x) = z$ **then accept**
else reject

For a random input $z \in \{0,1\}^{2l}$, $A'(z)$ can accept only when z is in the image of G_l, that is, with probability at most $\frac{1}{2^l}$.

Consider, on the other hand, input $z = G_l(x)$ for a random x. With probability at least ϵ, $A(z)$ will find a pair (x', y') such that $f(x', y') = f(x, \cdot) = G_l(x)$. Then $G_l(x') = f(x', y') = G_l(x)$, so $A'(z)$ will accept.

A' then distinguishes G_l's output with advantage $\epsilon - \frac{1}{2^l}$:

$$\left|\mathbf{Pr}_{z\in\{0,1\}^{2l}}[A'(z) = 1] - \mathbf{Pr}_{x\in\{0,1\}^l}[A'(G_l(x)) = 1]\right| = \epsilon - \frac{1}{2^l}$$

To relate the adversaries, note that the circuit size of A' is essentially the circuit size of A plus the circuit size of G_l; recall that if G_l is computable in time t then it has a circuit of size $O(t^2)$.

The reduction show us that if f is not (S, ϵ)-one-way (i.e., by the existence of A), then G is not $(S+O(t^2), \epsilon-\frac{1}{2}$-indistinguishable (by our construction of A'). The contrapositive gives a positive statement: if G is (S, ϵ)-indistinguishable then f is $(S - O(t^2), \epsilon + \frac{1}{2^l})$-one-way.
□

PRG from a OWF

While it is straightforward to see that pseudorandom generators imply one-way functions, the converse is not as easy. The general result was shown in 1989 by Impagliazzo, Levin, and Luby; in 1990, Håstad gave a uniform reduction. These general results, however, are hard and do not have efficient reductions.

The version presented here is an earlier result by Blum, Micali, and Yao in 1982: it shows a special case where the one-way function is a permutation. It also gives an efficient reduction.

DEFINITION 3 (ONE-WAY PERMUTATION) *A permutation $f : \{0,1\}^n \to \{0,1\}^n$ is (S,ϵ)-one-way if for every circuit A of size at most S,*

$$\mathbf{Pr}_{x\in\{0,1\}^n}[A(f(x)) = x] \le \epsilon.$$

If f is also computable in poly(n) time, we say it is a one-way permutation.

Note that since f is a permutation, its output completely determines its input. Because f is one-way, however, there is no efficient way to recover the input with high probability.

It is also important to notice that while the totality of information on the input is disguised, some of the information may leak. For instance, a one-way permutation may preserve the parity of its input or simply leave a particular bit alone; this information on the input would to trivial to recover from the output.

Hard-Core Predicate

The hard-core predicate captures what information is disguised by a one-way permutation.

DEFINITION 4 (HARD-CORE PREDICATE) *A predicate $B : \{0,1\}^n \to \{0,1\}$ is (S,ϵ)-hard-core for a permutation $f : \{0,1\}^n \to \{0,1\}^n$ if for a circuit A of size at most S,*

$$\mathbf{Pr}_{x\in\{0,1\}^n}[A(f(x)) = B(x)] \le \frac{1}{2} + \epsilon.$$

EXERCISE 1 Let the permutation $f : \{0,1\}^n \to \{0,1\}^n$ and the predicate $B : \{0,1\}^n \to \{0,1\}$ be computable in poly(n) time. Show that if B is hard-core for f_n then f_n is one-way.

To create a hard-core predicate, we need some way to extract a difficult-to-guess bit of information from a one-way permutation.

LEMMA 2
If $f : \{0,1\}^n \to \{0,1\}^n$ is a (S,ϵ)-one-way permutation with superpolynomial $S, 1/\epsilon$, then there exist S' and ϵ' such that for every circuit A of size at most S',

$$\mathbf{Pr}_{x,r\in\{0,1\}^n}[A(f(x), r) = x \odot r] \le \frac{1}{2} + \epsilon',$$

where the inner product $x \odot r = \sum_i x_i r_i \pmod 2$.

Proof: *Omitted.* □

Theorem 3 (Yao 1982, Goldreich and Levin 1989)
If there is a one-way permutation, then there is another one-way permutation with a hard-core predicate.

Proof:

Lemma 2 does not immediately reconcile with the definition of a hard-core predicate. We need a permutation that provides r:

If $f : \{0,1\}^n \to \{0,1\}^n$ is a one-way permutation, then $f' : \{0,1\}^{2n} \to \{0,1\}^{2n}$ where $f(x,r) = (f(x), r)$ is also a one-way permutation. Moreover, $B : \{0,1\}^{2n} \to \{0,1\}$ where $B(y,r) = y \odot r$ is a hard-core predicate for f'.

□

Pseudorandom Generator

The existence of a hard-core predicate allows us to create a security-preserving but length-extending primitive: a pseudorandom generator.

Theorem 4
If there is a one-way permutation, then there is a pseudorandom generator.

Proof:

Since there is a one-way permutation, Lemma 2 tells us that there is another one-way permutation, $f : \{0,1\}^n \to \{0,1\}^n$ with a corresponding hard-core predicate $B : \{0,1\}^n \to \{0,1\}$.

Define $G_n : \{0,1\}^n \to \{0,1\}^{n+1}$ as $G_n(x) = (f(x), B(x))$.

To see that G_n is computationally indistinguishable from uniform, suppose that there is some boolean distinguisher circuit D such that

$$\left|\mathbf{Pr}_{x\in\{0,1\}^n}[D(G_n(x)) = 1] - \mathbf{Pr}_{z\in\{0,1\}^{n+1}}[D(z) = 1]\right| \geq \epsilon.$$

Since D and $1 - D$ have basically the same complexity, we can assume that D distinguishes G_n's distribution with positive correlation. That is,

$$\mathbf{Pr}_{x\in\{0,1\}^n}[D(G_n(x)) = 1] - \mathbf{Pr}_{z\in\{0,1\}^{n+1}}[D(z) = 1] \geq \epsilon.$$

We will define an adversary, A, for f's hard-core predicate, B:

input: $y \in \{0,1\}^n$
  pick at random $b \in \{0,1\}$
  $v := D(y, b)$
  if $v = 1$ **then** output b
    else output $1 - b$

Intuitively, A picks b as a guess of the value of $B(x)$. The guess is verified with a call to $D(y, b)$. If the guess was good and D accepts, A outputs B; otherwise, if D rejects, A outputs b's complement.

We wish to show that $A(G_n(x))$ predicts $B(x)$ with high probability over the choices of x and b.

$$\begin{aligned}\mathbf{Pr}[A(G_n(x)) = B(x)] = \quad & \mathbf{Pr}[A(f(x)) = B(x)|b = B(x)]\mathbf{Pr}[b = B(x)] \\ & + \mathbf{Pr}[A(f(x)) = B(x)|b \neq B(x)]\mathbf{Pr}[b \neq B(x)]\end{aligned}$$

and since x and b are independent

$$\begin{aligned}
&= \frac{1}{2}\mathbf{Pr}[A(f(x)) = B(x)|b = B(x)] + \frac{1}{2}\mathbf{Pr}[A(f(x)) = B(x)|b \neq B(x)] \\
&= \frac{1}{2}\mathbf{Pr}[D(f(x), b) = 1|b = B(x)] + \frac{1}{2}\mathbf{Pr}[D(f(x), b) = 0|b \neq B(x)] \\
&= \frac{1}{2}\mathbf{Pr}[D(f(x), b) = 1|b = B(x)] + \frac{1}{2} - \frac{1}{2}\mathbf{Pr}[D(f(x), b) = 1|b \neq B(x)] \\
&= \frac{1}{2} + \mathbf{Pr}[D(f(x), b) = 1|b = B(x)] \\
&\quad - \frac{1}{2}\left(\mathbf{Pr}[D(f(x), b) = 1|b = B(x)] + \mathbf{Pr}[D(f(x), b) = 1|b \neq B(x)]\right) \\
&= \frac{1}{2} + \mathbf{Pr}[D(f(x), B(x)) = 1] - \mathbf{Pr}[D(f(x), b) = 1]
\end{aligned}$$

$G_n(x) = (f(x), B(x))$, and $(f(x), b)$ is uniformly distributed in $\{0,1\}^{n+1}$, so

$$\begin{aligned}
&= \frac{1}{2} + \mathbf{Pr}[D(G_n(x)) = 1] - \mathbf{Pr}_{z \in \{0,1\}^{n+1}}[D(z) = 1] \\
&\geq \frac{1}{2} + \epsilon
\end{aligned}$$

Thus A has a good (ϵ) advantage in predicting B, and its complexity is basically the same as D's. Thus, analogous to the proof of Theorem 1, if f is a one-way permutation, then G_n is a pseudorandom generator. $\Box$

In the next lecture we will see how to extend the output of our PRG while preserving security.