6.875J/18.425J Cryptography and Cryptanalysis	October 17, 2001
Lecture 11	
Instructor: Anna Lysyanskaya	*Scribed by: Alice Chan*

1 Recap of definitions

1.1 One-way functions and one-way and trapdoor permutations

Definition 1 *A set of efficient probabilistic algorithms $(G(\cdot), S(\cdot), f_\cdot(\cdot))$, specifies a family of one-way functions (OWF family) if for all PPTF $\{A_k\}$ there exists a negligible function $\nu(k)$ such that*

$$\Pr[PK \leftarrow G(1^k); x \leftarrow S(PK); y \leftarrow f_{PK}(x); x' \leftarrow A_k(PK, y) : f(x) = f(x')] \leq \nu(k)$$

and for all $PK \in G$, for all $x \in S(PK)$, f_{PK} is a function.

Definition 2 *A family of one-way functions (G, S, p) is a family of one-way permutations (OWP family) if for all $PK \in G$, p_{PK} is a permutation.*

Definition 3 *A family of one-way permutations (G, S, p) is trapdoor (TDP family) if (1) the output of G consists of two values, denoted (PK, SK); and (2) there is an efficient algorithm D such that for all k,*

$$\Pr[(PK, SK) \leftarrow G(1^k); x \leftarrow S(PK); x' \leftarrow D(SK, p_{PK}(x) : x = x'] = 1$$

Definition 4 *A family of one-way functions (resp., permutations) (G, S, f) is a one-way function (resp., permutation) if G is a deterministic algorithm.*

A TDP family is conceivable but is a TDP (with no family) conceivable?
No. For each k, there is just one f_k. So if there exists trapdoor, S_k, for f_k, then A_k might know S_k.

1.2 Hardcore bit

Motivation: How to encrypt using a trapdoor permutation family, such as RSA. How to construct pseudorandom sequences.

Definition 5 *Let (G, S, f) be a OWF family. Let $b_{(\circ)}(\circ)$ be a Boolean function. b is a hardcore bit for (G, S, f) if*

1. *$b_{PK}(x)$ is approximately a bit of information about x. More precisely, there exists a negligible function μ such that for all $PK \in G(1^k)$*

$$|\Pr[x \leftarrow S(PK); b_{PK}(x)] - 1/2| \leq \mu(k)$$

2. *It is hard to compute $b_{PK}(x)$ from the values PK and $f_{PK}(x)$. More precisely, for all PPTF $\{A_k\}$ there exists a negligible function $\nu(k)$ such that*

$$\Pr[PK \leftarrow G(1^k); x \leftarrow S(PK); b' \leftarrow A_k(PK, f_{PK}(x)) : b' = b_{PK}(x)] \leq 1/2 + \nu(k)$$

Note that although a hardcore bit is defined for OWF families, this concept makes much more sense when we consider OWP families.

Note that actually the non-triviality condition follows from the hardness condition.

2 Construcing a one-bit PRG from a special OWP

Recall the definition of a pseudorandom generator:

Definition 6 *Let $l < m$, and let G is a deterministic algorithm which, given a sequence of l bits, outputs a sequence of m bits. G is a pseudorandom generator (PRG) if for all PPTF A_l there is a negligible function $\nu(l)$ such that*

$$\Pr[x \leftarrow \{0,1\}^l; y_0 \leftarrow G(x); y_1 \leftarrow \{0,1\}^m; b \leftarrow \{0,1\}; b' = A_l(y_b) \ : \ b' = b] \leq 1/2 + \nu(l)$$

Theorem 7 *Suppose $p_k \ : \ \{0,1\}^k \mapsto \{0,1\}^k$ is a one-way permutation. Suppose that $B_k(\circ)$ is hardcore for $p_k(\circ)$. Then $G(x) = p_{|x|}(x) \circ B_{|x|}(x)$ is a pseudorandom generator.*

Proof Suppose that G is not a PRG. Then there is a distinguisher A_k that can distinguish $G(x)$ from a random y. We must use this distinguisher to tell us $B(x)$ from $p(x)$, contradicting the supposition that B is hard-core for p.

Define the reduction R as follows:
On input $z = p(x)$
1. Run $A_k(z \circ 0) = b_0$
2. Run $A_k(z \circ 1) = b_1$
3. In the case where $b_0 \neq b_1$, output a if $b_a = 0$.
4. Otherwise output the result of a coin flip.

$\Pr[x \leftarrow \{0,1\}^k; b \leftarrow R(p(x)) : b = B(x)] =$
$Pr[x \leftarrow \{0,1\}^k : A_k(p(x) \circ B(x)) = 0 \wedge A_k(p(x) \circ \bar{B}(x) = 1] +$
$\frac{1}{2} * Pr[x \leftarrow \{0,1\}^k : A_k(p(x) \circ B(x)) = A_k(p(x) \circ \bar{B}(x))]$

Let $P_G = Pr[x_0 \leftarrow \{0,1\}^k : A_k(p(x_0) \circ B(x_0)) = 0]$ and
$P_r = Pr[x_0 \leftarrow \{0,1\}^k : A_k(p(x_0) \circ r) = 0]$ where r is a random bit.

Because A_k is a distinguisher, we know that $P_G - P_r \geq \epsilon(k)$, $\epsilon(k)$ negligible function.

Let A be the event that $A_k(p(x) \circ B(x))$ outputs 0 and
B be the event that $A_k(p(x) \circ \bar{B}(x))$ outputs 0.

Then $P_G = Pr[A]$ and $Pr[correct] = Pr[A \wedge \bar{B}] + \frac{1}{2} * Pr[A \wedge B] + \frac{1}{2} * Pr[\bar{A} \wedge \bar{B}]$.

Claim 8 *P_r is the average of $Pr[A]$ and $Pr[B]$.*

$$P_r = Pr[0 \text{ on random}] = \frac{1}{2} * Pr[0 \text{ on } B(x)] + \frac{1}{2} * \text{Pr}[0 \text{ on } \bar{B}(x)]$$
$$= \frac{Pr[A]+Pr[B]}{2}$$

So $Pr[B] = 2 * P_r - Pr[A] = 2 * P_r - P_G$

$Pr[wrong] = Pr[\bar{A} \wedge B] + \frac{1}{2} * Pr[A \wedge B] + \frac{1}{2} * Pr[\bar{A} \wedge \bar{B}]$

$$\begin{aligned} Pr[correct] - Pr[wrong] &= Pr[A \wedge \bar{B}] - Pr[\bar{A} \wedge B] \\ &= Pr[A \wedge \bar{B}] + Pr[A \wedge B] - Pr[A \wedge B] - Pr[\bar{A} \wedge B] \\ &= Pr[A] - Pr[B] = P_G - 2 * P_r + P_G = 2 * P_G - 2P_r \geq 2\epsilon(k) \end{aligned}$$

Therefore $Pr[correct] \geq \frac{1}{2} + \epsilon(k)$.

■

Why is this important?
If you have a PRG that takes l-bit strings and outputs $l+1$-bit strings then you can construct a PRG that takes l-bit strings and outputs $m(l)$-bit strings for all polynomials $m(l)$. (The theorem only works if you have a OWP $p : \{0,1\}^k \rightarrow \{0,1\}^k$.)

Do other reductions work?
-Pick a at random and let $b_a = A_k(p(x) \circ a)$. If $b_a = 0$ output a. Otherwise output $\bar{a}$.
Exercise: Show that this is equivalent to the above reduction.
-Let $b \leftarrow A_k(p(x) \circ 0)$. If $b = 0$ output 0. Otherwise output 1.
Exercise: Show that this does not work.

3 Candidates for OWP, OWP families and their hardcore bits

Exponentiation modulo prime

a) OWP under a stronger DL (discrete log) assumption.
   (strongest) -computing DL modulo any prime is hard
   (stronger) -computing DL modulo the smallest prime of length k is hard, and finding this prime is easy

OWP candidate:
   Find the smallest prime of length k, call it p_k.
   Find a generator g for $\mathbb{Z}^*_{p_k}$.
   $f_k(x) = g^x \bmod p_k$.
   f_k is a permutation over $\mathbb{Z}^*_{p_k}$ and it is one way if for all $\{A_k\}$, there exists $\nu(k)$ such that $Pr[x \leftarrow \mathbb{Z}^*_{p_k}; x' \leftarrow A_k(g^x \bmod p_k) : x = x'] \leq \nu(k)$.

b) OWP family under a weaker DL assumption

-computing DL modulo a random prime is hard

(b) is weaker than (a) because (a) is deterministic.

OWP family candidate:

PK is created by picking a random prime together with its generator g (this is easier than finding the generator for a specific prime). First generate a $p-1$ where you know its factorization (this is possible- E. Bach, A.Kalai). Then test if p is prime.

$PK = (p, g), f_{PK}(x) = g^x \bmod p$.

Exercise: How does knowing the factorization of $p-1$ help you find a generator for $\mathbb{Z}_p^*$?

Hardcore bits:

GL bit; $B_{PK}(x) =$ "Is $x > \frac{p-1}{2}$?" is a hardcore bit (Blum, Micali).