6.875J/18.425J Cryptography and Cryptanalysis	November 5, 2001
Lecture 16	
Instructor: Anna Lysyanskaya	Scribed by: Kolya Zeldovich

1 Recap

1.1 PRF definition

A random function is one whose output is fixed at random. A pseudorandom function, intuitively, is one that is not really random, but you cannot tell. This notion was introduced by Goldreich, Goldwasser, and Micali, who also gave a construction.

Let $\mathcal{F} : D \mapsto R$ be a function; let $A(\cdot)$ be and algorithm. By $A^{\mathcal{F}}(\cdot)$ we denote an algorithm that makes oracle calls to the function $\mathcal{F}$. More precisely, we may imagine that A has an additional tape Q (for "query"), on which it can write queries to $\mathcal{F}$, and as soon as it is done writing such a query q, an answer to it, namely $\mathcal{F}(q)$ magically appears on this tape.

Definition 1 *A family of functions $F.(\cdot)$ is pseudorandom if F is efficiently computable and for a randomly chosen S, F_S cannot be distinguished from a random function. More precisely, let $\rho(k, l(k), m(k))$ denote a family of random functions from $\{0,1\}^{l(k)}$ to $\{0,1\}^{m(k)}$. $F.(\cdot)$ determines a family of pseudorandom functions from $l(k)$ bits to $m(k)$ bits if $\forall$ PPTF $\{A_k\}$, there exists a negligible function $\nu(k)$ such that*

$$\begin{aligned}
\Pr[S \leftarrow \{0,1\}^k; O_0 = F_S &\quad ; \\
R \leftarrow \rho(k, l(k), m(k)); O_1 = R &\quad ; \\
b \leftarrow \{0,1\}; b' \leftarrow A_k^{O_b}(1^k) &\quad : \quad b = b'] \leq 1/2 + \nu(k)
\end{aligned}$$

Intuitively, the Turing machine for F and the random k bit seed implicitly defines a string of length $2^{l(k)} \cdot m(k)$.

1.2 The GGM construction of PRFs from PRGs

Suppose G is a PRG that expands k random bits into $2k$ pseudorandom bits. Then here is a tree-like construction of a function, F_S, from $\{0,1\}^k$ to $\{0,1\}^k$ with a k-bit seed:

Since $G(x)$ is length $2k$, we can divide it in half and create G_0 and G_1. $G_0(x)$ is the first k bits of $G(x)$ and $G_1(x)$ is the last k bits of $G(x)$

Let $S \leftarrow \{0,1\}^k$ be the random seed and $x = x_1 \circ x_2 \circ \ldots \circ x_k$. Then $F_s(x) \triangleq l_x$ where

$$\begin{aligned}
l_x &= l_{x_1 \circ x_2 \circ \ldots \circ x_k} \\
&= G_{x_k}(l_{x_1 \circ x_2 \circ \ldots \circ x_{k-1}}) \\
&= G_{x_k} G_{x_{k-1}}(l_{x_1 \circ x_2 \circ \ldots \circ x_{k-2}})
\end{aligned}$$

$$\begin{aligned}
l_0 &= G_0(S) \\
l_1 &= G_1(S)
\end{aligned}$$

Why is F_s a PRF? We can show that F_s is a PRF using a hybrid argument. The hybrids cannot be generated horizontally because there would be an exponentially large number of hybrids, but we can generate them vertically.

The hybrids are defined in this way:

0^{th} hybrid: F_S for some random S.

i^{th} hybrid: Everything on level i is random. On all levels $j > i$, labels are derived from labels in level i.

k^{th} hybrid: Some random function.

Claim: If A_k distinguishes F_S from a random function R, then, for some i, it can distinguish F_S^i from F_S^{i+1}.

This claim is true because there are only a polynomial number of hybrids and because of the claim we proved the first time that hybrids were introduced. From the problem sets, we also know that we can choose i at random.

1.3 Using A_k to distinguish $G(\cdot)$ from a random string

There are two important claims:

1. A_k can be used to distinguish $p(k)$ random $2k$-bit strings from $p(k)$ pseudorandom $2k$-bit strings. (Here $p(k)$ is an upper bound on the number of queries that A_k makes).

2. If A_k can distinguish $p(k)$ random strings from $p(k)$ pseudorandom strings, then it can tell if a given string is random or pseudorandom.

2 The Naor-Reingold Construction

2.1 The DDH assumption

Let $P(1^k)$ be an algorithm that generates a prime p of length k and an element $g \in \mathbb{Z}_p^*$ such that g has large multiplicative order q. (Let's say that $q = \Omega(\sqrt{p})$.) Let $G =< g >$. Let's write $(p, q, g) \leftarrow P(1^k)$.

The decisional Diffie-Hellman assumption is that, for all adversaries $\{A_k\}$, there exists a negligible function $\nu(k)$ such that

$$\begin{aligned} \Pr[(p,q,g) \leftarrow P(1^k); x \leftarrow \mathbb{Z}_q; y \leftarrow Z_q \quad &; \\ z_0 = xy \bmod q; z_1 \leftarrow \mathbb{Z}_q; b \leftarrow \{0,1\} \quad &; \\ b' \leftarrow A_k(p,q,g,g^x,g^y,g^{z_b}) \quad &: \quad b = b'] \leq 1/2 + \nu(k) \end{aligned}$$

2.2 The PRG based on the DDH

Recall from a previous lecture that the following construction is a PRG:

We want to first construct a PRG using the DDH assumption. Suppose p, q, g are fixed as in the DDH construction. As a first try, let's take

$$G(x, y) = (g^x \circ g^y \circ g^{xy})$$

The intuition is that if one can't tell g^{xy} from a random g^z (under DDH), then this could be pseudorandom. However, this is not a PRG in the usual sense. Recall that a PRG takes $\{0,1\}^{l(k)}$ into $\{0,1\}^{m(k)}$ and more or less uniformly distributes it. However, here $g^x < p$ and so the first part of the output only covers a fraction $p/2^k$ of the whole space $\{0,1\}^k$, in fact a particular portion of the space, namely those numbers y which are less than p. This fact might enable an adversary to distinguish between the output of G from a truly random distribution.

We could argue that if p is very close to 2^k then the above contradiction doesn't hold. However, there's another problem with this generator. Since g is of order q, g^x is of order dividing q. However, it's easy to show that a random element of $\mathbb{Z}_p^*$ has order $\phi(p) = p - 1$ (the Euler phi function) with good probability. In fact, the number of elements of $\mathbb{Z}_p$ of order $\phi(p)$ is $\phi(\phi(p)) = \phi(p-1) = \Omega(\sqrt{p-1})$. So the adversary could just check if it's input had order dividing q or not (by raising it's input to the power q and checking if it's 1 mod p), and thus distinguish between the output of G and a random distribution.

So we need to modify our PRG somewhat. The solution to our problems just lies in throwing away the first few bits. The construction proceeds as follows:

First odd primes p, q are generated with $p = 2q + 1$ (therefore $p = 3 \bmod 4$).
Suppose x, y are randomly chosen in $\mathbb{Z}_q$. G will output $u \circ v \circ w$ computed as follows.
Let $G_0(x, y) = g^x, g^y, g^{xy}$.
Note that $p = 2q + 1$ means that elements of order q are quadratic residues modulo p (since $x^q = x^{(p-1)/2} = 1 \bmod p$ is the condition for QR). Therefore g^x is a square modulo p and we can extract square roots $\pm a$ (since $p = 3 \bmod 4$). Let u' be the square root which is $\leq (p-1)/2$. Similarly compute square roots $v', w' \leq (p-1)/2$ for g^y and g^z. u', v', w' are k-bit numbers, so truncate the first $k/4$ bits to get u, v, w and return $u \circ v \circ w$. This returns a $9k/4$ bit string (the initial string $x \circ y$ was $2k$ bits long).

2.3 Random self-reducibility of DDH

Suppose we are given a tuple $(g, X = g^x, Y = g^y, Z = g^z)$. We wish to determine whether $z = xy \bmod q$, or z is random from $\mathbb{Z}q$. To that end, we can randomize the tuple in such a way as to obtain $(Y' = g^{y'}, Z' = g^{z'})$ such that y' is completely random, and in the event that $z = xy$, $z' = xy'$, while on the other hand, if z was random, then z' is also completely random and independent of (x, y, z, y').

This is done as follows: pick values a, b at random from $\mathbb{Z}q^*$. Let $Y' = (Yg^a)^b$, $Z' = (ZX^a)^b$. It is clear that $y' = (y + a)b$ and $z' = (z + ax)b$. Let us consider two cases, one where $Z = g^{xy}$, and one where $Z = g^z$ where $z \neq xy$.

Case 1: $(X, Y, Z) = (g^x, g^y, g^{xy})$. First, observe that $Y' = g^{y'}$ is random. We know that $y' = (y+a)b$ and $Y' = (Yg^a)^b$. For a random value of Y', there exist $(q-1)$ pairs (a, b) that map Y onto Y'. Choose any of $(q-1)$ different values of a. Then we can uniquely solve the equation $y' = (y + a)b$ to find b. Since the total number of pairs (a, b) is $(q-1)^2$, the likelihood of each value of Y' is $1/(q-1)$, and therefore Y' is random. It's easy to see that $Z' = (ZX^a)^b = (g^{xy}g^{xa})^b = g^{bxy+axb} = (g^{by+ab})^x = (Y')^x = g^{xy'}$, which is what we want.

Case 2: suppose $(X, Y, Z) = (g^x, g^y, g^z)$ where $z \neq xy \bmod q$. Then Y' and Z' are completely random. As shown above, $y' = (y + a)b$ and $z' = (z + ax)b$. Simplifying $xy' - z' = x(y + a)b - (z + ax)b = xyb - zb$ we can see that this allows us to find b: $b = \frac{xy'-z'}{xy-z}$. Having found b, we can solve for a using the equation for y'.

This demonstrates that for every pair of (y', z') we have a unique pair of (a, b) values that correspond to it. Therefore there must be a total of $(q-1)^2$ pairs of (y', z') in this case, and therefore Y' and Z' are completely random.

2.4 The NR construction

The NR is a construction of a "relaxed" PRF into the domain $G = Z_p^*$. The structure of the construction is similar to the tree-based GGM construction of a PRF from a PRG in lecture 13.

Choose some p, q, g as in the Decisional Diffie-Hellman assumption above. Let the seed of this PRF F_s be $(x_0, x_1, \ldots, x_k)$, where x_i are random elements of Z_q $\forall$ i. Then define the PRF F_s as

$$F_s(m) = F_s(m_0 \circ m_1 \circ \cdots \circ m_k) = g^{x_0 \Pi_{i=1}^{k} x_i^{m_i}}$$

The first few levels of this constructions are demonstrated in this diagram:

$$\begin{array}{lcccc}
\text{Level 0} & & g^{x_0} & & \\
\text{Level 1} & g^{x_0} & & g^{x_0x_1} & \\
\text{Level 2} & g^{x_0} \quad g^{x_0x_2} & & g^{x_0x_1} \quad g^{x_0x_1x_2} & \\
 & \ldots \qquad \ldots & & \ldots &
\end{array}$$

The bottom level of the tree, which has depth k, would have elements of the form $g^{x_0}, g^{x_0x_k}, \ldots, g^{x_0x_1\ldots x_k}$.

2.5 Proof of NR construction PRF

As with the proof of the GGM construction, a vertical hybrid argument can be used here. Suppose that there exists an adversary which can distinguish between $F_s(m)$ and a truly random function, with advantage $\epsilon(k)$. Then it is possible to use this adversary to distinguish between one of these hybrids, by the hybrid argument:

$H_0 = F_s(m)$

H_i is a hybrid tree where the ith level of the tree is truly random. That is, it consists of elements of the form g^z, where z is chosen at random from Z_q. The $i+1$st level is produced according to the above construction.

$H_{k+1} = R$, a truly random function.

Let A'_k be the adversary that can distinguish between some two hybrids H_i and H_{i+1} with advantage $\epsilon(k)/k$. It can be reduced to a distinguisher between the two kinds of DDH tuples with advantage $\epsilon(k)/k$, as demonstrated below.

The reduction is given (p, q, g, X, Y, Z). By the DDH random self-reducibility theorem, we can sample as many independent tuples (Y', Z') as we want. Construct a tree, as in the NR construction, with values Y' on level i and Z' on level $i+1$ (the left child of Y' must be Y', and the right child would be Z'). Even though we need 2^i independent tuples at the ith level, the random self-reducibility allows us to accomplish this without losing the randomness at the ith level.

Given this tree, A'_k can tell whether the randomness starts on level i or $i+1$. By construction, these two cases map exactly onto the two types of DDH input tuples.

Suppose the input was a tuple where $z = xy$. Then $Y' = g^{y'}, Z' = g^{xy'}$, and the constructed tree is exactly the ith hybrid, H_i. Now suppose the input was a tuple where $z \neq xy$. Then the random self-reducibility guarantees that Y' and Z' will both be random, and therefore we will have constructed a tree equivalent to the $i+1$st hybrid H_{i+1}.

We have demonstrated that by using A'_k it is possible to distinguish between two types of DDH tuples, contradicting the Decisional Diffie-Hellman assumption. Clearly this implies that the NR construction must be a PRF under the DDH assumption.

It's worth noting that, although the GGM construction also demonstrates the existence of a PRF given a PRG algorithm, the NR construction only uses one (vertical) hybrid argument. The GGM construction proof relied on two (one horizontal and one vertical) hybrid arguments, which makes the required adversary $1/t(k)$ weaker. While in theory both are correct, in practice the NR construction offers a higher margin of randomness.

3 Pseudorandom permutations

References: Luby-Rackoff. This presentation is from Goldreich's book.

3.1 Definition

Definition 2 *A family of permutations $P.(\cdot)$ is pseudo-random if both P and P^{-1} are efficiently computable, and the output is indistinguishable from a random permutation, given only oracle access to P_s. More precisely, let $\Pi(1^k)$ be the uniform distribution of all permutations over $\{0,1\}^k$. Then $P.(\cdot)$ is a pseudo-random permutation if $\forall$ PPTF $\{A_k\}$, there exists a negligible function $\nu(k)$, such that*

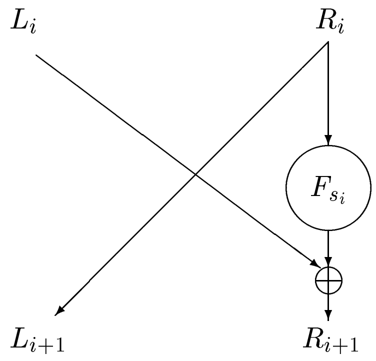
$$\begin{aligned} \Pr[s \leftarrow \{0,1\}^k; O_0 = P_s; O_1 \leftarrow \Pi(1^k) \quad &; \\ b \leftarrow \{0,1\}; b' \leftarrow A_k^{O_b}(1^k) \quad &: \quad b = b'] \leq 1/2 + \nu(k) \end{aligned}$$

Definition 3 *A family of permutations $P.(\cdot)$ is strongly pseudo-random if it is pseudo-random, and its output is indistinguishable from random given oracle access to both P_s and P_s^{-1}. More precisely,*

$$\Pr[s \leftarrow \{0,1\}^k; O_0 = P_s; O_1 \leftarrow \Pi(1^k) \quad ; \\ b \leftarrow \{0,1\}; b' \leftarrow A_k^{O_b, O_b^{-1}}(1^k) \quad : \quad b = b'] \leq 1/2 + \nu(k)$$

3.2 Construction

The following construction of a PRP on the basis of a one-way pseudo-random function $F_s(x) : \{0,1\}^{2k} \to \{0,1\}^{2k}$ is due to Luby and Rackoff. First, separate the input x into two parts, denoted by L_0 and R_0, of equal length: $x = L_0 \circ R_0, |L_0| = |R_0|$. Then we can obtain (L_{i+1}, R_{i+1}) from (L_i, R_i) as follows:



More precisely, $(L_{i+1}, R_{i+1}) = (R_i, L_i \oplus F_{s_i}(R_i))$, where $s_0 \circ s_1 \circ \cdots = s$, the seed for the PRP. The Luby-Rackoff construction uses repeated applications of this round (often referred to as a Feistel round) to construct a PRP and SPRP.

Let's make a few useful claims about this construction, which will be proved below:

1. $\forall\ i$, $P_i(x, y) \equiv (L_i, R_i)$ where $x = L_0, y = R_0$, is a permutation, and it is efficiently computable and invertible.
2. $P_3(x, y)$ is a PRP, for random s_1, s_2, s_3.
3. $P_4(x, y)$ is a SPRP, for random $s_1, \ldots, s_4$.

3.3 Proof idea

Proving the first claim is easy. By construction, we can see that it is efficiently computable. If we show how to efficiently invert it, we automatically get the fact that it's a permutation. It's not hard to see that finding (L_i, R_i) given (L_{i+1}, R_{i+1}) knowing s_i can be done with $R_i = L_{i+1}$, $L_i = R_{i+1} \oplus F_{s_i}(R_i)$.

To prove that the three-round construction is a PRP, we show that it is statistically close to a PRP whenever the pseudorandom function in the construction is replaced with a truly random function. Then a distinguisher who sees the difference, breaks the underlying PRF.

A more detailed proof of the latter two claims will be given in the next lecture.