CSE 599b: Cryptography **(Winter 2006)**

Lecture 9: Pseudorandom Function Families and Permutations

1 February 2006

Lecturer: Paul Beame *Scribe: Paul Beame*

1 Pseudorandom Function Families

Recall our definitions from last time:

Definition 1.1. *An* infinite keyed function family $\mathcal{F}$ *is an infinite sequence* $\{F^k\}_{k\geq 1}$ *where* $F^k : \{0,1\}^k \times \{0,1\}^{\ell(k)} \to \{0,1\}^{\ell}(k)$. *We call the* ℓ *the* length *of the family.*

We write F_K *for the function* $F^{|K|}(K,\cdot) : \{0,1\}^{\ell(k)} \to \{0,1\}^{\ell(k)}$.

Note that we can equivalently view $\mathcal{F}$ as an ensemble $\{\mathcal{F}_k\}_{k\geq 1}$ where $\mathcal{F}_k$ is the distribution on the $Func(\ell(k),\ell(k))$ induced by choosing $K \leftarrow \mathcal{U}_k$ and returning F_K.

We can then write our definition of a pseudorandom function family as follows:

Definition 1.2. *A function ensemble* $\mathcal{F}$ *of length* ℓ *is* a pseudorandom function family (PRFF) *if and only if*

1. *$\mathcal{F}$ is polynomial-time computable; i.e. the function that computes $F(K,x) = F_K(x) = F^{|K|}(K,x)$ for $x \in \{0,1\}^{\ell(|K|)}$ is computable in deterministic polynomial time.*

2. *For all (oracle) PPT A,*

$$Adv_A^{\mathcal{F}}(k) = \Pr[A^{\mathcal{F}_k}(1^k) = 1] - \Pr[A^{\mathcal{F}unc(\ell(k),\ell(k))}(1^k) = 1]$$

   is negligible.

We stated the following theorem last time which we will now prove.

Theorem 1.3 (Goldreich, Goldwasser, Micali)**.** *If PRNGs with factor 2 stretch exist then PRFFs exist with length* $\ell(k) = k$.

Proof. Let $G : \{0,1\}^* \to \{0,1\}^*$ be a PRNG such that $G : \{0,1\}^k \to \{0,1\}^{2k}$ for each k.

Define $G_0 : \{0,1\}^k \to \{0,1\}$ and $G_1 : \{0,1\}^k \to \{0,1\}$ by

$$G(y) = G_0(y)G_1(y).$$

That is G_0 gives the left half of the output of G and G_1 gives the right half. We can extend this definition for all subscripts $x \in \{0,1\}^*$ by

$$\begin{aligned} G_\lambda(y) &= y \qquad (\lambda \text{ is the empty string}) \\ G_{x0}(y) &= G_x(G_0(y)) \\ G_{x1}(y) &= G_x(G_1(y)). \end{aligned}$$

Each $G_x : \{0,1\}^k \to \{0,1\}^k$.

Now for $x \in \{0,1\}^k$ and $K \in \{0,1\}^k$ define $F(K,x) = F_K(x) = G_x(K)$.

Consider an (oracle) PPT A and let $\epsilon(n) = \Pr[A^{\mathcal{F}_k}(1^k) = 1] - \Pr[A^{\mathcal{F}unc(k,k)}(1^k) = 1]$. Let $q(k)$ be the maximum number of queries that A can make on input 1^k and any oracle. By definition, q is a polynomial function of k.

We will show that if $\epsilon(n)$ is non-negligible then there is PPT B (based on A) that receives at most $q(k)$ input strings of length $2k$ and can distinguish with probability $\epsilon'(k) = \epsilon(k)/k$ whether these strings are outputs of G on random inputs of length k rather than truly random strings. We earlier showed that if such an $\epsilon'(k)$ is non-neglible then that is enough to prove that G is not a PRNG.

The idea of the construction is another hybrid argument. Before we define B, we consider a number of hybrid distributions $\mathcal{H}_{j,k}$ on functions from $\{0,1\}^k$ to $\{0,1\}^k$.

These distirbutions are based on viewing the computation of $F_K(x) = G_x(K)$ for a randomly chosen K as involving a binary tree whose root (at level 0) is K and whose leaves (at level k) are the various $G_x(K)$. The internal nodes of the tree indexed by $\alpha \in \{0,1\}^{\le k}$ are labeled by $G_\alpha(K)$. The two children of node indexed by α are $G_0(G_\alpha(K))$ and $G_1(G_\alpha(K))$. Observe that

- having oracle access to F_K for a random K is equivalent to having access to such a tree having a randomly chosen k-bit string label the node at level 0
- having oracle access to a function from $\mathcal{F}unc(k,k)$ is equivalent to having access to such a tree having randomly chosen k-bit strings labeling each node at level k.

We thus define $\mathcal{H}_{j,k}$ to be a distribution which uses such a tree with randomly chosen k-bit nodes at level j and the remainder of the nodes labeled according to G.

Let $p_{j,k} = \Pr[A^{\mathcal{H}_{j,k}}(1^k) = 1]$. Clearly $\mathcal{H}_{0,k} = \mathcal{F}_k$ and $\mathcal{H}_{k,k} = \mathcal{F}unc(k,k)$ so as is usual in our hybrid arguments we have $\epsilon(k) = p_{0,k} - p_{k,k} = \sum_{j=1}^{k}(p_{j-1,k} - p_{j,k})$.

We now define B as follows:
On input $z_1, \ldots, z_{q(k)} \in \{0,1\}^{2k}$ and 1^k,

1. Choose j uniformly in $\{1, \ldots, k\}$.
2. Simulate A on input 1^k and when A makes a query $x \in \{0,1\}^k$ to its oracle:
   (a) if no string whose length $j-1$ suffix agreeing with that of x has previously been queried then use the next unused string z_i in B's input to label the two level j children of the node corresponding to this suffix.
   (b) Now find the longest suffix of x whose corresponding node in the tree has been labeled by a string and use G to continue labelling the children along the path to the leaf whose index is x and return this value to A.
3. Output 1 iff A outputs 1.

B is clearly a PPT and, by assumptions about A, B will always have enough input.

Consider a fixed choice of j. By construction if B chooses $k = j$ and receives $z_1, \ldots, z_{q(k)}$ from $\mathcal{U}_{2k}^{q(k)}$ then it acts like $A^{\mathcal{H}_{j,k}}(1^k)$ and if B receives $z_1, \ldots, z_{q(k)}$ from $G(\mathcal{U}_k)^{q(k)}$ then it acts like $A^{\mathcal{H}_{j-1,k}}(1^k)$. Therefore

$$\begin{aligned}
\epsilon'(n) &= \Pr[B(G(\mathcal{U}_k)^{q(k)}, 1^k) = 1] - \Pr[B(\mathcal{U}_{2k}^{q(k)}, 1^k) = 1] \\
&= \frac{1}{k}\sum_{j=1}^{k}(\Pr[A^{\mathcal{H}_{j-1,k}}(1^k) = 1] - - \Pr[A^{\mathcal{H}_{j,k}}(1^k) = 1) \\
&= \frac{1}{k}\sum_{j=1}^{k}(p_{j-1,k} - p_{j,k}) \\
&= \epsilon(k)/k.
\end{aligned}$$

□

2 Pseudorandom Permutation Families

The construction of PRFFs almost fits our notion of ideal block cipher, except that it does not produce invertible permutations. Let $Perm(k, k)$ be the set of all permutations on $\{0,1\}^k$.

Definition 2.1. *A function ensemble $\mathcal{F}$ is a* pseudorandom permutation family (PRPF) *if and only if*

1. *$\mathcal{F}$ is a PRFF*

2. *For each $K \in \{0,1\}^*$, F_K is a permutation in $Perm(\ell(k), \ell(k))$ and the function F^{inv} that maps $(K, Y) \in \{0,1\}^k \times \{0,1\}^{\ell(k)}$ to $F_K^{-1}(Y)$ is polynomial-time computable.*

Theorem 2.2 (Luby,Rackoff)**.** *If PRFFs exist then PRPFs exist.*

Proof Sketch. This uses three rounds of the Feistel construction using three independent chosen keys that we considered in earlier lectures. Given a function $f : \{0,1\}^k \to \{0,1\}^k$ define $D_f : \{0,1\}^{2k} \to \{0,1\}^{2k}$ by $D_f(x, y) = (y, y \oplus f(x))$. That is D_f implements one Feistel round using f and so always produces a permutation whose inverse is computable using f. The idea of the proof is a two phase argument

1. Show that for three independently chosen random functions $f_1, f_2, f_3 \leftarrow \mathcal{F}unc(k, k)$, $D_{f_3} \circ D_{f_2} \circ D_{f_1}$ is indistinguishable from a random element of $\mathcal{F}unc(2k, 2k)$. The idea of this argument is that for a polynomial number of distinct queries it is almost certain that the functions f_2 and f_3 will be queried on arguments they have never seen before.

2. Using a hybrid argument show that replacing each of f_1, f_2, and f_3 by independent random selections from $\mathcal{F}_k$ is also indistinguishable from using random functions. (The hybridizing is only over the 4 options: Having the first $i \in \{0, 3\}$ of these functions being chosen from $\mathcal{F}_k$.) The result follows because $\mathcal{F}$ is a PRFF.

□

Luby and Rackoff proved even more than this. They showed that even if a test can call an oracle for f^{-1} as well as f then there is a construction that cannot be distinguished from random functions. More formally:

Definition 2.3. *A function ensemble $\mathcal{F}$ is a* strong *pseudorandom permutation family (strong PRPF) if and only if*

1. *$\mathcal{F}$ is polynomial time computable.*

2. *For each $K \in \{0,1\}^*$, F_K is a permutation and the function F^{inv} that maps $(K,Y) \in \{0,1\}^k \times \{0,1\}^{\ell(k)}$ to $F_K^{-1}(Y)$ is polynomial-time computable.*

3. *For all (oracle) PPT A,*

$$Adv_A^{\mathcal{F},\mathcal{F}^{-1}}(k) = \Pr[A^{F_K,F_K^{-1}}(1^k) = 1 \mid K \leftarrow \mathcal{U}_k] - \Pr[A^{F,F^{-1}}(1^k) = 1 \mid F \leftarrow \mathcal{F}unc(k,k)]$$

   is negligible.

Theorem 2.4 (Luby-Rackoff)**.** *$D_{f_4} \circ D_{f_3} \circ D_{f_2} \circ D_{f_1}$ for $f_1, \ldots, f_4$ chosen independently from a PRFF $\mathcal{F}$ is a strong PRPF.*

Using the basic methodology of the Luby-Rackoff, the only issue we need to deal with in the security of variants these constructions is the analogue of step 1 of their proof, the security of the construction using random functions. It has been shown that $D_f \circ D_f \circ D_f$ is insecure, which means that using the same key for all 3 Feistel rounds of their PRPF construction is insecure.

Subsequent to their work, Naor and Reingold re-visited the ideas and realized that the first and last rounds of the construct really serve a different purpose and derived a simpler argument to show that one can use two Feistel rounds with the same key provided that one sandwiched them between simpler objects called *universal permutation hash functions*. In this case $D_f \circ D_f \circ h$ is a PRFF and $h_2^{-1} \circ D_f \circ D_f \circ h_1$ is a strong PRFF.

3 Birthday Attack

The fact that it is hard to distinguish these permutations families from pseudorandom families certainly means that it is hard to distinguish random permutations from random functions. So if $\mathcal{P}erm(n,n)$ is the distribution that produces a uniformly chosen random permutation, what is the maximum over all A that make q queries of $\Pr[A^{\mathcal{P}erm(n,n)}(1^n) = 1] - \Pr[A^{\mathcal{F}unc(n,n)}(1^n) = 1]$?

It is clear that the only difference that A can detect is if A sees two inputs that yield the same output. Since A makes at most q queries there are only $\binom{q}{2}$ pairs that might collide and the chance for a random function of any fixed one of those pairs having a collision is precisely 2^{-n}. Therefore the expected number of collisions is at most $\binom{q}{2}/2^n = \frac{q(q-1)}{2^{n+1}}$ and the probability of finding one is at most this large. It is not hard to show that this is nearly the correct answer which means that to detect the difference with, say, constant probability requires $\Theta(2^{n/2})$ queries. This factor of 2 in the exponent is something that one must often keep in mind in estimating the security of cryptosystems.

(The name Birthday Attack comes from the surprising fact that in any room of 23 people there are likely to be at least two people who share a birthday.)