Notes for Lecture 16

Scribed by Anupam Prakash, posted March 16, 2009

Summary

Today we finish the analysis of a construction of a pseudorandom permutation (block cipher) given a pseudorandom function.

1 The Luby-Rackoff Construction

Recall that if $F : \{0,1\}^m \to \{0,1\}^m$ is a function, then we define the *Feistel permutation* $D_F : \{0,1\}^{2m} \to \{0,1\}^{2m}$ associated with F as

$$D_F(x, y) := y, x \oplus F(y) \tag{1}$$

Let $F : \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^m$ be a pseudorandom function, we define the following function $P : \{0,1\}^{4k} \times \{0,1\}^{2m} \to \{0,1\}^{2m}$: given a key $\overline{K}(K_1, \ldots, K_4)$ and an input x,

$$P_{\overline{K}}(x) := D_{F_{K_4}}(D_{F_{K_3}}(D_{F_{K_2}}(D_{F_{K_1}}(x)))) \tag{2}$$

If $\overline{F} = F_1, F_2, F_3, F_4$ are four functions, then $P_{\overline{F}}$ is the same as the above construction but using the functions F_i:

$$P_{\overline{F}}(x) := D_{F_4}(D_{F_3}(D_{F_2}(D_{F_1}(x)))) \tag{3}$$

If A is an oracle algorithm, we define as $S(A)$ the probabilistic process in which we run a simulation of A in which we reply to each query with a random answer.

2 Today's Proof

The proof of the following result is what was missing from yesterday's analysis.

Lemma 1 *For every non-repeating algorithm A of complexity $\leq t$ we have*

$$\left| \mathbb{P}_{\overline{F}}[A^{P_{\overline{R}}, P_{\overline{R}}^{-1}}() = 1] - \mathbb{P}[S(A) = 1] \right|$$

$$\leq \frac{t^2}{2 \cdot 2^{2m}} + \frac{t^2}{2^m}$$

Proof: The transcript of A's computation consists of all the oracle queries made by A. The notation $(x, y, 0)$ represents a query to the π oracle at point x while $(x, y, 1)$ is a query made to the π^{-1} oracle at y. The set T consists of all valid transcripts for computations where the output of A is 1 while $T' \subset T$ consists of transcripts in T consistent with π being a permutation.

We write the difference in the probability of A outputting 1 when given oracles $(P_{\overline{R}}, P_{\overline{R}}^{-1})$ and when given a random oracle as in $S(A)$ as a sum over transcripts in T.

$$\begin{aligned} &\left| \mathbb{P}_{\overline{F}}[A^{P_{\overline{R}}, P_{\overline{R}}^{-1}}() = 1] - \mathbb{P}[S(A) = 1] \right| \\ = &\left| \sum_{\tau \in T} \left(\mathbb{P}_{\overline{F}}[A^{P_{\overline{R}}, P_{\overline{R}}^{-1}}() \leftarrow \tau] - \mathbb{P}[S(A) \leftarrow \tau] \right) \right| \end{aligned} \tag{4}$$

We split the sum over T into a sum over T' and $T \setminus T'$ and bound both the terms individually. We first handle the simpler case of the sum over $T \setminus T'$.

$$\begin{aligned} &\left| \sum_{\tau \in T \setminus T'} \left(\mathbb{P}_{\overline{F}}[A^{P_{\overline{R}}, P_{\overline{R}}^{-1}}() \leftarrow \tau] - \mathbb{P}[S(A) \leftarrow \tau] \right) \right| \\ = &\left| \sum_{\tau \in T \setminus T'} (\mathbb{P}[S(A) \leftarrow \tau]) \right| \\ \leq &\ \frac{t^2}{2.2^{2m}} \end{aligned} \tag{5}$$

The first equality holds as a transcript obtained by running A using the oracle $(P_{\overline{R}}, P_{\overline{R}}^{-1})$ is always consistent with a permutation. The transcript generated by querying an oracle is inconsistent with a permutation iff. points x, y with $f(x) = f(y)$ are queried. $S(A)$ makes at most t queries to an oracle that answers every query with an independently chosen random string from $\{0,1\}^{2m}$. The probability of having a repetition is at most $(\sum_{i=1}^{t-1} i)/2^{2m} \leq t^2/2^{2m+1}$.

Bounding the sum over transcripts in T' will require looking into the workings of the construction. Fix a transcript $\tau \in T'$ given by $(x_i, y_i, b_i), 1 \leq i \leq q$, with the number of queries $q \leq t$. Each x_i can be written as (L_i^0, R_i^0) for strings L_i^0, R_i^0 of length m corresponding to the left and right parts of x_i. The string x_i goes through 4 iterations of D using the function $F_k, 1 \leq k \leq 4$ for the kth iteration. The output of the construction after iteration k, $0 \leq k \leq 4$ for input x_i is denoted by (L_i^k, R_i^k).

Functions F_1, F_4 are said to be good for the transcript τ if the multisets $\{R_1^1, R_2^1, \cdots, R_q^1\}$ and $\{L_1^3, L_2^3, \cdots, L_q^3\}$ do not contain any repetitions. We bound the probability of F_1 being bad for τ by analyzing what happens when $R_i^1 = R_j^1$ for some i, j:

$$R_i^1 = L_i^0 \oplus F_1(R_i^0)$$

$$R_j^1 = L_j^0 \oplus F_1(R_j^0)$$

$$0 = L_i^0 \oplus L_j^0 \oplus F_1(R_i^0) \oplus F_1(R_j^0) \tag{6}$$

The algorithm A does not repeat queries so we have $(L_i^0, R_i^0) \neq (L_j^0, R_j^0)$. We observe that $R_i^0 \neq R_j^0$ as equality together with equation (6) above would yield $x_i = x_j$. This shows that equation (6) holds only if $F_1(R_j^0) = s \oplus F_1(R_i^0)$, for a fixed s and distinct strings R_i^0 and R_j^0. This happens with probability $1/2^m$ as the function F_1 takes values from $\{0,1\}^m$ independently and uniformly at random. Applying the union bound for all pairs i, j,

$$Pr_{F_1}[\exists i, j \in [q], \quad R_i^1 = R_j^1] \leq \frac{t^2}{2^{m+1}} \tag{7}$$

We use a similar argument to bound the probability of F_4 being bad. If $L_i^3 = L_j^3$ for some i, j we would have:

$$\begin{aligned} L_i^3 &= R_i^4 \oplus F_4(L_i^4) \\ L_j^3 &= R_j^4 \oplus F_4(L_j^4) \end{aligned}$$

$$0 = R_i^4 \oplus R_j^4 \oplus F_4(L_i^4) \oplus F_4(L_j^4) \tag{8}$$

The algorithm A does not repeat queries so we have $(L_i^4, R_i^4) \neq (L_j^4, R_j^4)$. We observe that $L_i^4 \neq L_j^4$ as equality together with equation (8) above would yield $y_i = y_j$. This shows that equation (8) holds only if $F_4(L_j^4) = s' \oplus F_4(L_i^4)$, for a fixed string s' and distinct strings L_i^4 and L_j^4. This happens with probability $1/2^m$ as the function F_4 takes values from $\{0,1\}^m$ independently and uniformly at random. Applying the union bound for all pairs i, j,

$$Pr_{F_4}[\exists i, j \in [q], \quad L_i^3 = L_j^3] \leq \frac{t^2}{2^{m+1}} \tag{9}$$

Equations (7) and (9) together imply that

$$Pr_{F_1, F_4}[F_1, F_4 \text{ not good for transcript } \tau] \leq \frac{t^2}{2^m} \tag{10}$$

Continuing the analysis, we fix good functions F_1, F_4 and the transcript τ. We will show that the probability of obtaining τ as a transcript in this case is the same as the probability of obtaining τ for a run of $S(A)$. Let $\tau = (x_i, y_i, b_i), 1 \leq i \leq q \leq t$. We calculate the probability of obtaining y_i on query x_i over the choice of F_2 and F_3 .

The values of the input x_i are in bijection with pairs (L_i^1, R_i^1) while the values of the output y_i are in bijection with pairs (L_i^3, R_i^3), after fixing F_1 and F_4. We have the relations (from (1)(3)):

$$\begin{aligned} L_i^3 = R_i^2 &= L_i^1 \oplus F_2(R_i^1) \\ R_i^3 = L_i^2 \oplus F_3(R_i^2) &= R_i^1 \oplus F_3(L_i^3) \end{aligned}$$

These relations imply that (x_i, y_i) can be an input output pair if and only if we have $F_2(R_i^1), F_3(L_i^3) = (L_i^3 \oplus L_i^1, R_i^3 \oplus R_i^1)$. Since F_2 and F_3 are random functions with range $\{0,1\}^m$, the pair (x_i, y_i) occurs with probability 2^{-2m}. The values R_i^1 and $L_i^3, (i \in [q])$ are distinct because the functions F_1 and F_4 are good. This makes the occurrence of (x_i, y_i) independent from the occurrence of (x_j, y_j) for $i \neq j$. We conclude that the probability of obtaining the transcript τ equals 2^{-2mq}.

The probability of obtaining transcript τ equals 2^{-2mq} in the simulation $S(A)$ as every query is answered by an independent random number from $\{0,1\}^{2m}$. Hence,

$$\begin{aligned} & \left| \sum_{\tau \in T'} \left(\mathbb{P}_{\overline{F}}[A^{P_{\overline{R}}, P_{\overline{R}}^{-1}}() \leftarrow \tau] - \mathbb{P}[S(A) \leftarrow \tau] \right) \right| \\ \leq & \left| \sum_{\tau \in T'} \mathbb{P}_{F_2, F_3} \left[A^{P_{\overline{R}}, P_{\overline{R}}^{-1}}() \leftarrow \tau | F_1, F_4 \text{ not good for } \tau \right] \right| \\ \leq & \frac{t^2}{2^m} \left| \sum_{\tau \in T'} \mathbb{P}_{F_2, F_3} [A^{P_{\overline{R}}, P_{\overline{R}}^{-1}}() \leftarrow \tau] \right| \\ \leq & \frac{t^2}{2^m} \end{aligned} \tag{11}$$

The statement of the lemma follows by adding equations (5) and (11) and using the triangle inequality. □

This concludes the analysis of the Luby-Rackoff scheme for constructing pseudorandom permutations from a family of pseudorandom functions.