University of Maryland
CMSC456 — Introduction to Cryptography
Professor Jonathan Katz

Lecture 16

1 Private-Key Encryption Using PRFs

PRFs and PRPs alone are not enough to immediately give secure encryption schemes; some care is needed. For example, consider the "natural" encryption scheme in which P is a PRP (with efficient inversion) and the sender and receiver share a key s in advance. To encrypt message m, the sender computes $C = P_s(m)$ and sends this to the receiver. To decrypt, the receiver computes $m = P_s^{-1}(C)$ (since P is a permutation, decryption always succeeds). However, while this is secure against ciphertext only attacks (when a single message is encrypted) it does *not* satisfy our notion of indistinguishability (in fact, it cannot since this encryption scheme is deterministic).

1.1 An Indistinguishable Encryption Scheme

Here is one possible solution: Let $F : \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^n$. We construct an encryption scheme for messages of length n where the shared key is of length k (note that even if $k > n$ we are constructing a scheme that has *stronger* security properties than the one-time pad). The sender and receiver share a random key $s \in \{0,1\}^k$. To encrypt message M, the sender chooses a random $r \in \{0,1\}^m$, computes $C = M \oplus F_s(r)$, and sends $\langle r, C\rangle$. To decrypt ciphertext $\langle r, C\rangle$, the receiver computes $M = C \oplus F_s(r)$ (note that we need not be able to invert F — in fact, F_s need not even be a permutation).

Note the parallel with the encryption scheme from the previous lecture in which the shared key s was viewed as $s_1, \ldots, s_N$. Letting $s_i \stackrel{\text{def}}{=} f_s(i)$, it is as if the sender and receiver are sharing 2^m secrets $s_1, \ldots, s_{2^m}$ (since there are 2^m different possible values for r) and using them in a completely analogous manner. However, *they each only store k bits* and *they can each* **efficiently** *compute s_i for any i* — these properties are exactly what PRFs allow. One difference with the scheme from the previous lecture was that, previously all the keys were completely random whereas now they are all generated using a PRF. We will prove below that this does not ruin the security of the scheme.

Theorem 1 *If F is a PRF, then the scheme above is indistinguishable (i.e., secure against an adversary who can query the* LR *oracle any polynomial number of times). In particular, if F is a (t, ϵ)-PRF then for all adversaries A running in time t and making at most ℓ queries to the* LR *oracle we have:*

$$\left|\Pr[s \leftarrow \{0,1\}^k; b \leftarrow \{0,1\} : A^{\mathsf{LR}_{b,s}(\cdot,\cdot)} = b] - 1/2\right| \leq \epsilon + O(\frac{\ell^2}{2^m}).$$

(Note that if $\epsilon(\cdot)$ were a negligible function, and $\ell(\cdot)$ were at most polynomial in k, and $m = \Theta(k)$ then $\epsilon(k) + O(\frac{\ell^2}{2^m})$ is negligible.)

Note that since the adversary is limited to running in time at most t, we must have $\ell \leq t$.

Proof Assume that F is a (t, ϵ)-PRF, and assume we have an adversary A attacking the encryption scheme above in the sense of left-or-right indistinguishability. We construct an adversary B which tries to distinguish F from a random function. Recall that B is given oracle access to some function which is either a completely random function or else a function F_s where s is chosen at random. B runs as follows:

$B^{F(\cdot)}$
  Choose random $b \in \{0,1\}$
  Run A, simulating the LR oracle calls of A as follows:
    On input (m_0, m_1) from A, pick random $r \in \{0,1\}^n$
    Query oracle F on input r to get output y
    Return the answer $y \oplus m_b$ to A
  When A outputs its guess b' for b:
    if $b' = b$ then guess "pseudorandom" (i.e., return 1)
    if $b' \neq b$ then guess "random (i.e., return 0)

Note that the running time of B is essentially t, which is the running time of A. Since F is a (t, ϵ)-PRF, we know that:

$$\left|\Pr[s \leftarrow \{0,1\}^k : B^{F_s(\cdot)} = 1] - \Pr[F \leftarrow \mathsf{Rand}^{m \to n} : B^{F(\cdot)} = 1]\right| \leq \epsilon. \tag{1}$$

When does B output 1? Exactly when A outputs the correct guess for b. Also note that when B's oracle is pseudorandom, then B provides A with an exact simulation of the real LR oracle. These observations imply:

$$\Pr[s \leftarrow \{0,1\}^k : B^{F_s(\cdot)} = 1] = \Pr[s \leftarrow \{0,1\}^k; b \leftarrow \{0,1\} : A^{\mathsf{LR}_{b,s}(\cdot,\cdot)} = b]. \tag{2}$$

Let $\mathsf{Succ}_A \stackrel{\text{def}}{=} \Pr[s \leftarrow \{0,1\}^k; b \leftarrow \{0,1\} : A^{\mathsf{LR}_{b,s}(\cdot,\cdot)} = b]$ denote the *success of adversary A* (in the real experiment). Equations (1) and (2) show that

$$|\mathsf{Succ}_A - 1/2| \leq \left|\Pr[F \leftarrow \mathsf{Rand}^{m \to n}; b \leftarrow \{0,1\} : A^{\mathsf{LR}_{b,F}(\cdot,\cdot)} = b] - 1/2\right| + \epsilon$$

(where the notation $\mathsf{LR}_{b,F}(\cdot, \cdot)$ refers to an imaginary instantiation of the encryption scheme in which a completely random F is shared instead of a seed for a PRF).

Define event Coll as follows: we say Coll occurs (a *collision occurs*) if any of the r values used throughout the experiment are the same. Conditioning on this event gives:

$$\begin{aligned}
&\left|\Pr[F \leftarrow \mathsf{Rand}^{m \to n}; b \leftarrow \{0,1\} : A^{\mathsf{LR}_{b,F}(\cdot,\cdot)} = b] - 1/2\right| \\
&= \Big|\Pr[F \leftarrow \mathsf{Rand}^{m \to n}; b \leftarrow \{0,1\} : A^{\mathsf{LR}_{b,F}(\cdot,\cdot)} = b \mid \mathsf{Coll}] \Pr[\mathsf{Coll}] \\
&\quad + \Pr[F \leftarrow \mathsf{Rand}^{m \to n}; b \leftarrow \{0,1\} : A^{\mathsf{LR}_{b,F}(\cdot,\cdot)} = b \mid \overline{\mathsf{Coll}}] \Pr[\overline{\mathsf{Coll}}] - 1/2\Big|.
\end{aligned}$$

Now, the key point is the following: if a collision never occurs, then (because we assume the sender/receiver share a completely random function) the adversary A has absolutely no information about bit b (and its probability of guessing b correctly is exactly $1/2$). Therefore:

$$
\begin{aligned}
&\left|\Pr[F \leftarrow \mathsf{Rand}^{m \to n}; b \leftarrow \{0,1\} : A^{\mathsf{LR}_{b,F}(\cdot,\cdot)} = b] - 1/2\right| \\
&= \quad \left|\left(\Pr[F \leftarrow \mathsf{Rand}^{m \to n}; b \leftarrow \{0,1\} : A^{\mathsf{LR}_{b,F}(\cdot,\cdot)} = b \mid \mathsf{Coll}] - 1/2\right) \cdot \Pr[\mathsf{Coll}]\right| \\
&\leq \quad 1/2 \cdot \Pr[\mathsf{Coll}]
\end{aligned}
$$

(where we also use the fact that $\Pr[\overline{\mathsf{Coll}}] = 1 - \Pr[\mathsf{Coll}]$).

Finally, we need to bound $\Pr[\mathsf{Coll}]$. This is the probability that I get a match if I pick ℓ items at random from a set of size $\{0,1\}^m$. This problem is the well-known "birthday problem" (see the notes on probability) from which it is known that $\Pr[\mathsf{Coll}] = O(\ell^2/2^m)$.

Plugging everything in gives the final result:

$$
\begin{aligned}
&|\mathsf{Succ}_A - 1/2| \\
&\quad \leq \quad 1/2 \cdot \Pr[\mathsf{Coll}] + \epsilon \\
&\quad = \quad O(\frac{\ell^2}{2^m}) + \epsilon.
\end{aligned}
$$

■