CS 346 Class Notes

Mark Lindberg

Feb 8, 2016

Review:

Let F be a length-preserving, efficient, keyed function. F is a PRF if for all PPT distinguishers D,

$$|\Pr(D^{F_k}(1^m) = 1) - \Pr(D^f(1^m) = 1)|$$

is negligible where k is a random m-bit string and f is a random function from m-bit strings to m-bit strings. A keyed function has $\ell_{in}(m)$, $\ell_{out}(m)$, $\ell_{key}(m)$. In a length-preserving function, all of these polynomials are equal to m.

Today:

Pseudorandom Permutation (PRP).

- $\ell_{in}(m) = \ell_{out}(m)$, and "$F_k$" is a permutation.
- Following the text, we will again assume that it is length-preserving for simplification.
- If we say that a PRP is "efficient", we mean that both F_k and F_k^{-1} can be computed in polynomial time.

In the definition of PRP, "f" of the definition of a PRF is now a random permutation. There are $(2^n)!$ possible permutations that f is being drawn from.

Proposition 3.27: If F is a PRP, then it is a PRF.

Proof idea: No PPT algorithm can tell the difference between a random function and a random permutation.

m^3 steps. To find a collision with good probability, need to make (tilde)$\approx \sqrt{2^m} = 2^{\frac{m}{2}}$ oracle calls. Birthday problem. It is not a freaking paradox.

Strong PRP, D gets two oracles, $D^{F_k, F_k^{-1}}$. $D^{f, f^{-1}}$.

Connection to practice: "block ciphers" are PRPs, generally for a specific key length.

Last time, we saw our first CPA secure scheme, which was based on a PRF. We encrypted by doing $\mathsf{Enc}_k(m) = (r, m \oplus F_k(r))$.

CPA security scheme for messages of $>>n$ bit lengths. (A polynomial message length.)

Today we'll see how to handle an arbitrary polynomial message length.

Relevant section of the textbook: Block cipher modes of operation (3.6.2).

1. ECB mode: Electronic CodeBook. We get blocks, $m_1, m_2, \ldots$ of the message, each of which is n bits long. We take and output $F_k(m_1), F_k(m_2), \ldots$. This is weak because, for example, if $m_1 = m_5$, $c_1 = F_k(m_1) = F_k(m_5) = c_5$. Repeated blocks encrypt to the same value. This fails EAV security trivially, and so is not even remotely CPA secure.

2. CBC mode: Cipher Block Chaining. We have an initial random value IV, $c_0 = IV$, $c_1 = F_k(m_1 \oplus IV)$, $c_2 = F_k(m_2 \oplus c_1)$, $c_3 = F_k(m_3 \oplus c_2)$, and so on.

   - Known to be CPA secure.
   - Minor variations on this scheme are insecure. Example: Cannot simply replace IV with a counter. If you know what the IV will be, an adversary can choose a m_1 that gives the same $\oplus$, and therefore, the same c_1, that was produced last time.

     Let $x = 0^{n-1}1$, oracle call gives back (IV, c), repeat if IV is odd, so get back one where IV is even. Use $m_0 = 0^m$, $m_1 \neq m_0$ (anything else). Challenge ciphertext looks like $(IV + 1, c')$. Notice that if $b = 0$, then $c = c'$ (Assuming IV is even, $IV \oplus 1 = IV + 1$.). If $b = 1$, $c \neq c'$.
   - Another insecure variant: "chained" CBC. If you reuse the last ciphertext, c_n, as the IV for the next message, you fails. Suppose I know m_1 is either m_1^0 or m_1^1. Assume a 3-block message, next message is m_4, m_5. Set $m_4 = IV \oplus m_1^0 \oplus c_3$. Then $c_4 = F_k(m_4 \oplus c_3) = F_k(IV \oplus m_1^0)$, which was the first input to give $c_1 = F_k(IV \oplus m_1^0)$, or m_1^1. This allows us to tell which value it was, and so we have broken CPA security.

3. OFB mode. Output FeedBack. We have $c_0 = IV$, $c_1 = m_1 \oplus F_k(IV)$, $c_2 = m_2 \oplus F_k^2(IV)$, $c_3 = m_3 \oplus F_k^3(IV)$, where F_k^n means F_k applied n times.

   - In this mode we get an "unsynchronized" stream cipher. In a synchronized stream cipher, communicators generate a long stream, and use up parts of it with each message. Here, we have a key k and an IV, and the IV changes each time.

4. CTR mode, CounTeR. $c_0 = CTR$, $c_1 = m_1 \oplus F_k(CTR + 1)$, $c_2 = m_2 \oplus F_k(CTR + 2)$.

   - Fully parallelizable. If there are multiple CPUs available, you can compute c_n without knowing c_{n-1}, and so they can each encrypt their own blocks.
   - CTR mode is CPA-secure. Similar to theorem 3.31 (maybe 3.32?) given last time. If $m = m_1$, then $c = (r, F_k(r) \oplus m)$. Given last time, let Π be this encryption scheme.
     - $\widetilde{\Pi}$ cannot be distinct from Π.
     - Show $\widetilde{\Pi}$ is CPA-secure. Run in $q(m)$ time. Assume each message passed to the encryption oracle $\leq q(m)$ in length. The probability of overlap with counters used is $\leq \frac{2q(m)^2}{2^n}$, which is negligible.