ICS 180: Introduction to Cryptography 6/3/2004

Handout 2: Symmetric Encryption from a PRF

A PRF is a very powerful source of (pseudo)randomness and therefore it can be immediately turned into powerful ciphers. The construction is very simple: just use the outputs of the pseudorandom function as one-time pads to xor your message with. We give here a simple proof that the resulting encryption is secure both under the Chosen Plaintext Attack (CPA) *and* under the "Lunchtime Attack", sometimes called "Chosen Ciphertext Attack 1" (CCA1).[1]

First, a Pseudorandom Function [PRF] family is defined as set of functions $\{F_s\}_{s\in\{0,1\}^n}$, where $F_s : \{0,1\}^{l(n)} \to \{0,1\}^{L(n)}$ for every $s \in \{0,1\}^n$, s.t.

1. $F_s(x)$ is polytime computable (for every s, x).

2. Functions F_s are indistinguishable from *random functions* on the same domain/range, i.e. from functions chosen at random from family of *all* functions mapping domain $\{0,1\}^{l(n)}$ to range $\{0,1\}^{L(n)}$. Formally, we require that for every PPT A, the following two distributions are indistinguishable:

$$\{A^{F_k(\cdot)}(1^n)\}_{k\leftarrow\{0,1\}^n} \approx \{A^{R(\cdot)}(1^n)\}_{R\leftarrow \texttt{RndFct}(l(n),L(n))} \tag{1}$$

   Where in each case, A can interact with functions F_k or R as with oracles: For any input A gives to the oracle, he receives an output a value of the function at this input.

Now, using such PRF family $\{F_s\}$, we can design a symmetric encryption scheme as follows:

$$KGen(1^n) \rightarrow k, \quad \text{for } k \leftarrow \{0,1\}^n \tag{2}$$

$$Enc_k(m) \rightarrow (x, F_k(x) \oplus m), \quad \text{for } x \leftarrow \{0,1\}^{l(n)}, \text{ assuming } |m| = L(n) \tag{3}$$

$$Dec_k((c_1, c_2)) \rightarrow F_k(c_1) \oplus c_2 \tag{4}$$

Theorem 1 *The above (symmetric) encryption scheme is (CPA,CCA1)-secure.*

Proof:

(Part 1) Recall first what does it mean that an (symmetric) encryption is (CPA,CCA1)-secure. It means that any PPT adversary A running in the following game, denoted $A^{O_{CPA/CCA1}}(1^n)$ (i.e. A has input 1^n and has access to the encryption oracle O_{CPA} and the decryption oracle O_{CCA1}):

- First the security parameter is given to both the adversary and all the other algorithms.
- Then the key k is picked by $KGen(1^n)$ and given to all the oracles (not the adversary!).

[1] Dodis's lecture notes give a more complicated proof of only the CPA security.

- Then A can make as many queries as he wants to the following two oracles:
  - He can ask the oracle O_{CPA} for *encryptions* of any messages m_i he chooses, and the oracle will run the encryption algorithm on m_i and k and return the resulting ciphertext to A.
  - He can ask the oracle O_{CCA1} for *decryptions* of any ciphertext-looking strings c_i he chooses, and the oracle will return to him the result of the decryption algorithm on c_i and k.
- When he is finally ready, he outputs two messages m_0, m_1 s.t. $|m_0| = |m_1|$ as his "test" messages. The "test oracle" chooses one of these messages, m_b at random, i.e. $b \leftarrow \{0,1\}$, and gives to A an encryption $c \leftarrow Enc(k, m_b)$.
- A can query the encryption oracle O_{CPA} some more, including on messages m_0 and m_1. But since this is CCA1 and not CCA2 he is forbidden from asking for fresh *decryptions* of ciphertext-looking messages he chooses (this stronger attack is called CCA2).
- Finally A outputs bit b' which is his judgement about what bit b is.
- We say taht the scheme is secure if the probability that $b' = b$ is at most negligibly better than a random coin toss, i.e. $1/2$.

To argue that $(KGen, Enc, Dec)$ is a (CPA,CCA1)-secure (symmetric) encryption scheme, we first note that equation (1) implies that for every PPT A, we have

$$\{A^{O^{(F_k)}_{CPA/CCA1}}(1^n)\}_{k\leftarrow\{0,1\}^n} \approx \{A^{O^{(R)}_{CPA/CCA1}}(1^n)\}_{R\leftarrow \texttt{RndFct}(l(n),L(n))}$$

In other words, the behavior of A in the following two interactions is indistinguishable:

- In one interaction, A interacts with the oracles implementing the encryption and decryption capability given to a (CPA,CCA1) attacker, with encryption and decryption implemented as in equations (3) and (4), i.e. using a (pseudorandom) function F_k for some k chosen at random in $\{0,1\}^n$.
- In the other interaction, A interacts with the oracles implementing the encryption and decryption capability given to a (CPA,CCA1) attacker, but the encryption and decryption are implemented in the following way:

$$Enc'(m) \rightarrow (x, R(x) \oplus m), \quad \text{for } x \leftarrow \{0,1\}^{l(n)}, \text{ assuming } |m| = L(n) \quad (5)$$
$$Dec'((c_1, c_2)) \rightarrow R(c_1) \oplus c_2 \quad (6)$$

  where R is a random function chosen in $\texttt{RndFct}(l(n), L(n))$.

Therefore, for every PPT attack A, whatever advantage ϵ this attack has in the (CPA,CCA1) attack against $(KGen, Enc, Dec)$ (i.e. in the interaction with $O^{(F_k)}_{CPA/CCA1}$, for random $k \in \{0,1\}^n$), the advantage ϵ' of A in the interaction with $O^{(R)}_{CPA/CCA1}$ for random R, must satisfy $|\epsilon' - \epsilon| \leq \mathsf{negl}(n)$

(Part 2):

Now we will upper-bound the probability that A guesses the bit b chosen by the $O^{(R)}_{CPA/CCA1}$ oracle in that last interaction.

First, note that A's encryption calls amount to A learning $R(x)$ for a random value x. Why do we say that? Because one can implement learning $F(x)$ for a random x given the ability to make an encryption query: If we query any m, then from the returned ciphertext we learn $R(c_1) = c_2 \oplus m$ for a randomly chosen c_1. Second, note that A's decryption calls amount to A learning $R(x)$ on x of A's choice. Why? Again, because to learn $R(x)$ on x, I can ask a decryption query on $m = Dec(x, c_2)$ for any c_2, and get back $R(x) = m \oplus c_2$.

Second, note that given the test ciphertext $c^* = Enc'(m_b) = (x^*, R(x^*) \oplus m_b)$, since R is a random function, the only way A can guess b with probability better than $1/2$ is if A learns the value $R(x^*)$ on this argument x^*.

But, since x^* was chosen at random by the $O^{(R)}_{CPA/CCA1}$ oracle, the probability that any of values x on which A learns the value $R(x)$ is equal to x^* can be upper bounded by the sum of two probabilities:

1. Probability Pr_1 that x^* is equal to one of the x's on which A learned value $R(x)$ via its encryption or decryption queries *before* seeing the ciphertext c^*.

2. Probability Pr_2 that x^* is equal to one of the x's on which A learns value $R(x)$ via its encryption queries *after* seeing the ciphertext c^*.[2]

Since A is polynomial time, he can make at most $p(n)$ queries (for some polynomial p) both before and after seeing c^*. Therefore, before seeing c^* he can learn the value of R on best $p(n)$ different x's. Thus Pr_1 is at most the probability that x^* is equal to one of them, which is at most $p(n)/2^{l(n)}$.

After seeing c^*, A learns the values $R(x)$ for at most $p(n)$ random x's. The probability Pr_2 that one of them is equal to x^* can be upper-bounded by $p(n)/2^{l(n)}$ again, because this is the probability that a set of $p(n)$ randomly chosen *different* elements x in $\{0,1\}^{l(n)}$ includes x^*. Therefore the probability that x^* is in a randomly chosen set of any elements is at most that.

Therefore, the probability that A guesses b in an interaction with $O^{(R)}_{CPA/CCA1}$ is at most

$$\epsilon' = Pr[A^{O^{(R)}_{CPA/CCA1}}(1^n) = b \mid R \leftarrow \texttt{RndFct}(l(n), L(n)); b \leftarrow \{0,1\}] < \frac{1}{2} + 2\frac{p(n)}{2^{l(n)}} = \frac{1}{2} + \mathsf{negl}(n)$$

And hence

$$\epsilon = Pr[A^{O^{(F_k)}_{CPA/CCA1}}(1^n) = b \mid k \leftarrow \{0,1\}^n; b \leftarrow \{0,1\}] < \epsilon' + \mathsf{negl}(n) < \frac{1}{2} + \mathsf{negl}(n)$$

□

[2]Note that since it's only a CCA1 and not a CCA2 attack, the adversary can only make encryption queries after seeing the test ciphertext c^*.