COM S 6830 – Cryptography Oct 18, 2011

Lecture 13: Multi-Message Security

Instructor: Rafael Pass *Scribe: Shyam Lenna*

1 Definition

We have Pseudo-Random Functions.
Goal: Multi-Message security. Here's a rough definition:

$$\{\text{Enc}(m_1), \text{Enc}(m_2) \dots \text{Enc}(m_q)\} \approx \{\text{Enc}(m'_1), \text{Enc}(m'_2) \dots \text{Enc}(m'_q)\}$$

More formally:

Definition 1 *Let f_s be a family of PRFs.*
$f_i : \{0,1\}^{|s|} \to \{0,1\}^{|s|}$
$\text{Gen}(1^n) = s \leftarrow \{0,1\}^n$
$\text{Enc}_k(m) = r||m \oplus f_k(r)$ $\quad r \leftarrow \{0,1\}^n$ *a random string*
$\text{Dec}_k(r||c) = c \oplus f_k(r)$

Anyone can be allowed to know r. If someone knows k, then they should be able to decrypt it efficiently, and otherwise it should not be possible to decrypt efficiently.
Next, we want to show that this satisfies Multi-Message Security.

2 Proof of Multi-Message Security

Proof. Assume: $\exists$ nuPPT $\mathcal{D}$, poly p,q, infinitely many $n \in \mathbb{N}$.
$\exists m_1, \dots, m_{q(n)}, m'_1, \dots, m'_{q(n)}$
$\mathcal{D}$ a distinguisher between:
$\{k \leftarrow \text{Gen}(1^n) : \text{Enc}(\vec{m})\}$
$\{k \leftarrow \text{Gen}(1^n) : \text{Enc}(\vec{m'})\}$

So, where's a good place to start? Let's try the Hybrid Lemma.

H_1 = Real Encryption of $\vec{m}$
$=\{k \leftarrow \text{Gen}(1^n), r_1, \ldots, r_q \in \{0,1\}^n, m_1 \oplus f_k(r_1), \ldots, m_q \oplus f_k(r_q)\}$
H_2 = Encryption of $\vec{m}$ using RF (random function)
$=\{F \leftarrow \text{RF}_n : r_1, \ldots, r_q \leftarrow \{0,1\}^n, m_1 \oplus F(r_1), \ldots, m_q \oplus F(r_q)\}$
H_3 = Fresh One Time Pad
$H_3 =\{P_1, \ldots P_q \leftarrow \{0,1\}^n, m_1 \oplus P_1, \ldots, m_q \oplus P_q\}$
$H_4 =H_3$ using $\vec{m}'$
$H_5 =H_2$ using $\vec{m}'$
$H_6 =H_1$ using $\vec{m}'$

$\mathcal{D}$ a distinguisher between H_1 and H_6 with probability $\frac{1}{p(n)}$
By the fact that f_i is a PRF, we get $H_1 \approx H_2$
For H_2 and H_3, the only time we can distinguish them is if two of the r's are exactly the same.
Because there are q choose 2 pairs of r's, each of which being selected from $\{0,1\}^n$

$$\begin{aligned}\Pr[\exists i, j \text{ s.t. } r_i = r_j, i \neq j] \leq& \binom{q}{2} \cdot 2^{-n} \\ \leq& q^2 \cdot 2^{-n}\end{aligned}$$

Which is negligible. So H_2 and H_3 are indistinguishable.
$H_3 \approx H_4$ because of OTP - it's the same proof, just think of it as a longer message and pad. If we really wanted, we could prove this with the Hybrid Lemma, but that's more difficult.
And we're done. ($H_4 \approx H_5$ and $H_5 \approx H_6$ follows the same steps as before). ■

3 Even More Security

3.1 CPA

So now we have Multi-Message Security. Is this good enough? Our definition is a generalization of Shannon secrecy, but we want more security. What if something was known about the message beforehand? For example, the Germans might end there emails with the same thing, giving us information about the message. Now, we want someone to be allowed to encrypt a message of their choice and still be unable get information from this.

This is Chosen Plaintext Attack (CPA), in which the attacker can choose to encrypt strings of their choosing. We discovered in the 1980s that even being secure against this might not be good enough.

3.2 CCA1

Lunchtime attack - you go to lunch, leaving the encryption/decryption machine in your room. The attacker gets access to the machine, but they still shouldn't be able to decrypt a message that you send after that. This is CCA1 security, or Chosen Ciphertext Attack 1.

3.3 CCA2

CCA2 demands even more security:

1. The attacker could get access to the encryption and decryption oracles.
2. They get a coded message c.
3. They still have access to encryption and decryption oracles that will do anything other than decrypt c.
4. They still should not be able to decrypt c.

4 Analysing CPA, CCA1, and CCA2

We are only defining these for single message security, but it also applies to multi message security.

Let $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ be an encryption scheme.
Let $\text{IND}_b^{O_1,O_2}(\Pi, \mathcal{A}, n)$, where $\mathcal{A}$ is a nuPPT, $n \in \mathbb{N}$, $b \in \{0,1\}$, denote the output of the following experiment (IND for indistinguishability):

$$k \leftarrow \text{Gen}(1^n)$$
$$m_0, m_1, state \leftarrow \mathcal{A}^{O_1(k)}(1^n)$$
$$c \leftarrow \text{Enc}_k(m_p);\ \text{Output } \mathcal{A}^{O_2(k)}(c, state)$$

Π is CPA/CCA1/CCA2 secure if $\forall$nuPPT $\mathcal{A}$,

$$\{\text{IND}_0^{O_1,O_2}(\Pi, \mathcal{A}, n)\}_{n\in\mathbb{N}} \approx \{\text{IND}_1^{O_1,O_2}(\Pi, \mathcal{A}, n)\}_{n\in\mathbb{N}}$$

Where for:

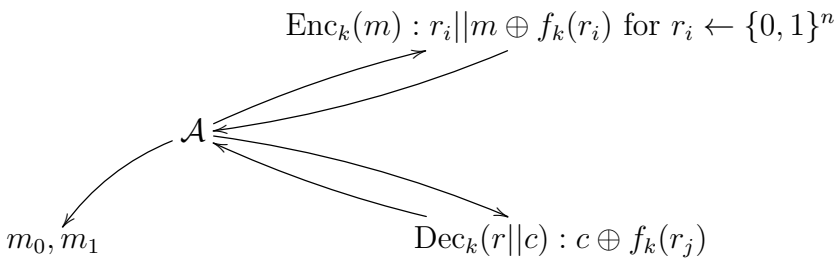
	$O_1(k)$	$O_2(k)$
CPA	$\text{Enc}_k(\cdot)$	$-$
$CCA1$	$\text{Enc}_k(\cdot), \text{Dec}_k(\cdot)$	$-$
$CCA2$	$\text{Enc}_k(\cdot), \text{Dec}_k(\cdot)$	$\text{Enc}_k(\cdot), \text{Dec}_k(\cdot)$

This is what the attacker is given, and $-$ means nothing. Additionally in the case of CCA2 security, we quantify only over $\mathcal{A}$ that never ask to decrypt c. Or, we could swap $\text{Dec}_k(\cdot)$ with $\text{Dec}_k^c(\cdot)$ which says 'no' to c for CCA2 $O_2(k)$ to make it a bit more formal.

4.1 CPA and CCA1

CLAIM: Same proof as earlier works for CPA (with the exact same hybrids).

For CCA1: The attacker can decrypt $\text{Dec}(r, c)$



This process is trying to learn $r||c$. There is no issue with the knowledge gained from $\text{Enc}_k(m)$. This attacker can only ask for polynomial many values of the form $r||c$ before getting c, so it can't determine c with more then negligible probability. This is because it's selecting polynomial values out of exponential possible values, and it fails.
The same hybrids from the first proof work for CCA1 as well.

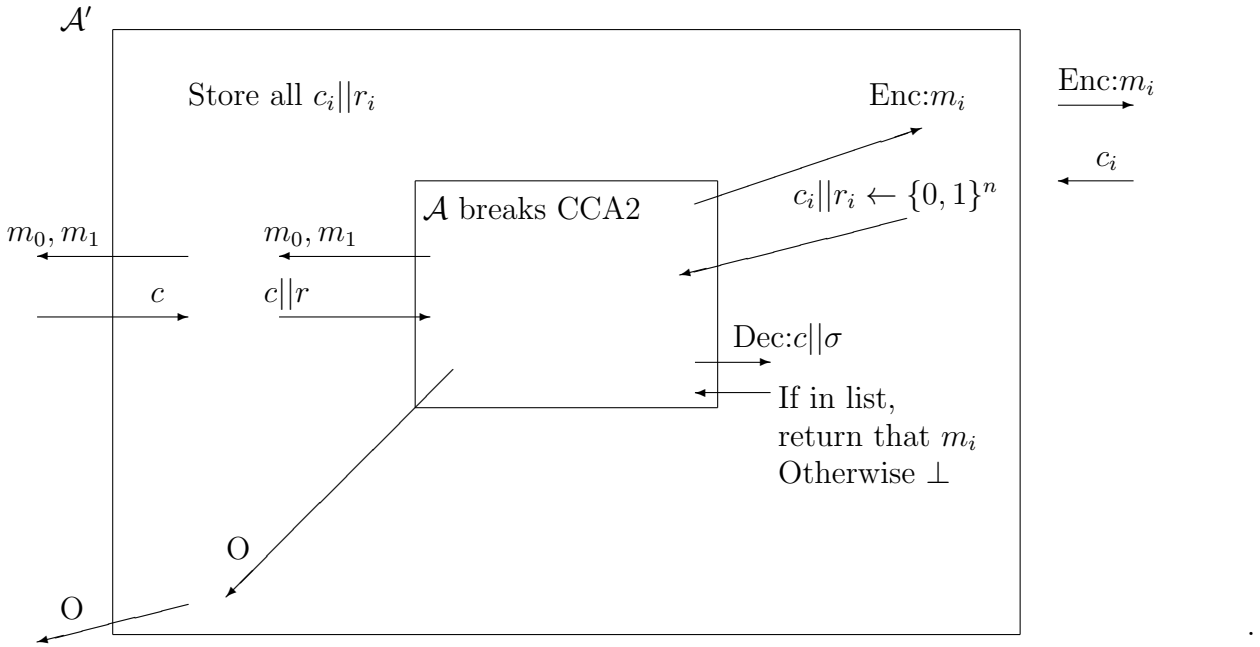
4.2 CCA2

This scheme is NOT CCA2 secure - he can get r by changing c, but it is possible to get a CCA2 secure scheme. It can be done by taking a CPA and a PRF to create something CCA2 secure.
For f a PRF.

$$
\begin{aligned}
\text{Gen}(1^n) &: k_1 \leftarrow \text{Gen}(1^n), k_2 \leftarrow \{0,1\}^{\text{poly}(n)} \\
\text{Enc}_{k_1,k_2}(m) &: c = \text{Enc}_{k_1}(m) \\
&\quad \sigma = f_{k_2}(c) \\
&\quad \text{output } (c, \sigma) \\
\text{Dec}_{k_1,k_2}(c, \sigma) &: \text{if } \sigma = f_{k_2}(c) \text{ then} \\
&\quad \text{output } \text{Dec}_{k_1}(c) \\
&\quad \text{otherwise } \perp
\end{aligned}
$$

k_1 is used to encrypt, k_2 is used to tag.

Consider a mental experiment where we use a RF instead of f_k.
$\mathcal{A}'$ breaks CPA(Gen, Enc, Dec)



O is the output of $\mathcal{A}$. Because we are using a random function, this only has a 2^{-n} guess, since during the Dec:$c||\sigma$ stage, $\mathcal{A}'$ cannot return anything if it did not already calculate and store the result.