18-330 Cryptography Notes: Symmetric Encryption

Note: This is provided as a resource and is not meant to include all material from lectures or recitations. The proofs shown, however, are good models for your homework and exams.

1 IND-CPA Security (Semantic Security)

1.1 IND-CPA Adversarial Game

Definition 1. *Let $\mathcal{E} = (KeyGen, E, D)$ be defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$. The IND-CPA game is defined as follows:*

1. *The experiment takes as input bit $b \in \{0, 1\}$, chosen uniformly at random.*
2. *The Challenger runs $k \leftarrow KeyGen(\lambda)$ for security parameter λ.*
3. *The Adversary runs some logic to select any two messages $m_0, m_1 \in \mathcal{M}$, where $|m_0| = |m_1|$. It then sends (m_0, m_1) to the Challenger.*
4. *The Challenger replies to the Adversary with $E(k, m_b)$.*
5. *Repeat steps 3 through 4 some $poly(log|\mathcal{K}|)$ number of times.*
6. *The Adversary runs some logic to output b', which is the output of the experiment.*

Note that k and b remain fixed for the duration of the experiment, so the challenger always encrypts the first message from the adversary (if $b = 0$) or always encrypts the second message (if $b = 1$).

1.2 Semantic Security Advantage

Definition 2. *Let $\mathcal{E}$ be an encryption scheme, and let A be an adversary. We define A's semantic security advantage as:*

$$Adv_{SS}[A, \mathcal{E}] := Pr[Exp(1) = 1] - Pr[Exp(0) = 1]$$

1.3 Semantic Security

In class, we define semantic security as follows:

Definition 3. *An encryption algorithm $\mathcal{E}$ is semantically secure if for all efficient adversaries A:*

$$Adv_{SS}[A, \mathcal{E}] < \epsilon \leq negl(\log |\mathcal{K}|)$$

Note that the textbook has different name for our notion of semantic security. The book calls it CPA security. Intuitively, the encryption algorithm is semantically secure if the probability that any adversary wins the IND-CPA game is no better than the probability of winning the game by simply guessing.

2 Stateful Counter Mode

Counter mode allows us to construct a variable-length IND-CPA secure encryption scheme from a secure PRF F.

Definition 4. *Let F be a secure PRF. Then we define counter mode:*

- *Encryption*

Algorithm 1: Encryption Algorithm $E_k(M)$

```
M[1]...M[m] ← M
C[0] ← ctr
for i = 1, ..., m do
    P[i] ← F_K(ctr + i)
    C[i] ← P[i] ⊕ M[i]
end
ctr ← ctr + m
return C
```

- *Decryption*

Algorithm 2: Decryption Algorithm $D_k(M)$

```
C[0]...C[m] ← C
ctr ← C[0]
for i = 1, ..., m do
    P[i] ← F_K(ctr + i)
    M[i] ← P[i] ⊕ C[i]
end
return M
```

2.1 Proof of Semantic Security

We prove that counter mode encryption is semantically secure via a reduction.

Proof. Let $\mathcal{E} = (KeyGen, E, D)$ be counter-mode encryption defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$, based on the secure PRF f. Suppose for the sake of contradiction that $\mathcal{E}$ is not semantically secure. Then there exists an efficient adversary $A_{IND-CPA}$ that wins the IND-CPA (semantic) security game with non-negligible probability. Using A_{IND}, we can construct an adversary A_{PRF} that can win the PRF security game with non-negligible probability:

Algorithm 3: Adversary A_{PRF}

1 Select d from {0,1}
2 Call A_{IND}
3 **while** A_{IND} *queries* (m_0, m_1) **do**
4   Query $Challenger_{PRF}$ to obtain sufficient $F_k(ctr + i)$'s to calculate $E(m_d)$.
5   Reply to A_{IND} with $E(m_d)$
6 **end**
7 Receive d' from A_{IND}
8 **if** $d' = d$ **then**
9   **return** 0
10 **else**
11   **return** 1
12 **end**

We show that A_{PRF} is an efficient adversary with non-negligible advantage.

As a first step to calculating the advantage of A_{PRF}, we argue that A_{PRF} perfectly simulates the challenger for A_{IND} when the PRF challenger for A_{PRF} uses a PRF (i.e., when the PRF challenger's bit is 0, meaning that it uses the PRF F). In this case, A_{IND} will send a message pair $(m_0, m_1) \in \mathcal{M} \times \mathcal{M}$ to $Challenger_{IND}$ (which is A_{PRF}). A_{PRF} will respond with $E(k, m_d)$. The exchange repeats a polynomial number of times. Then, A_{IND} outputs a guess d'. So, this adheres to the IND-CPA game perfectly.

Based on this argument, we can calculate the first part of A_{PRF}'s advantage, namely the probability that A_{PRF} outputs 1 when the challenge game is run with bit 0; i.e., $Pr[Exp(0) = 1] = 1$.

$$\begin{aligned}
Pr[Exp(0) = 1] &= 1 - Pr[A_{IND} \text{ wins with CTR + PRF}] && (1)\\
&= 1 - \left(\frac{1}{2}Pr[Exp_{A_{IND},\mathcal{E}}(1) = 1] + \frac{1}{2}Pr[Exp_{A_{IND},\mathcal{E}}(0) = 0]\right) && (2)\\
&= 1 - \left(\frac{1}{2}Pr[Exp_{A_{IND},\mathcal{E}}(1) = 1] + \frac{1}{2}\left(1 - Pr[Exp_{A_{IND},\mathcal{E}}(0) = 1]\right)\right) && (3)\\
&= 1 - \left(\frac{1}{2}\left(1 + Pr[Exp_{A_{IND},\mathcal{E}}(1) = 1] - Pr[Exp_{A_{IND},\mathcal{E}}(0) = 1]\right)\right) && (4)\\
&= 1 - \left(\frac{1}{2}(1 + Adv_{IND}[A, f])\right) && (5)\\
&= \frac{1}{2} - \frac{1}{2}Adv_{IND}[A, f] && (6)
\end{aligned}$$

Some brief justification: A_{PRF} outputs 1 (on line 11 of the algorithm) only when $d' \neq d$ (i.e., when A_{IND} guesses incorrectly about which message(s) were encrypted). The probability that this happens is simply one minus the probability that A_{IND} guesses correctly, which gives us line 1 above. Line 2 expands "guess correctly" into the two possible conditions in which A_{IND} can be correct: Either the game has bit 1 and A_{IND} says 1, or the game has bit 0 and A_{IND} says 0. These two possible settings for the bit each occur with 50% probability. Line 3 simply says that the probability that the game outputs 0 is one minus the probability that it outputs 1 (since there are only two possible outputs). Line 4 just rearranges terms. Line 5 observes that the last two terms in Line 4 are the definition of $Adv_{IND}[A, f]$.

Next, we need to calculate the second part of A_{PRF}'s advantage, namely the probability that A_{PRF} outputs 1 when the challenge game is run with bit 1; i.e., $Pr[Exp(1) = 1] = 1$:

$$\begin{aligned} Pr[Exp(1) = 1] &= 1 - Pr[A_{IND} \text{ wins with CTR + Rand F}] && (7) \\ &= 1 - \frac{1}{2} && (8) \\ &= \frac{1}{2} && (9) \end{aligned}$$

Line 7 is justified in the same way as in the previous calculation. Line 8 is much more subtle and requires reasoning about how CTR mode operates. In particular, note that by design CTR mode never invokes the underlying function (whether it is a PRF or a random function) with the same input twice. Hence, when we encrypt using a truly random function, this means that each call to encrypt chooses a uniformly random element (call it p) from the range of F (this is the definition of a random function) and XORs it with the message. That random element p is never used again (unless we happen to randomly select it), and hence, we can view the scheme as exactly a one-time pad scheme (recall that a OTP randomly selects a key and XORs it with the message). Because a OTP is perfectly secret, the output of A_{IND} is perfectly random with respect to the actual choice of bit d, and hence the probability that A_{IND} wins is $\frac{1}{2}$.
Now we calculate the advantage of A_{PRF} and show that it is non-negligible.

$$\begin{aligned} Adv_{PRF}[A_{PRF}, f] &:= |Pr[Exp_{A_{PRF},f}(0) = 1] - Pr[Exp_{A_{PRF},f}(1) = 1]| && (10) \\ &= \left| \frac{1}{2} - \frac{1}{2} Adv_{IND}[A, f] - \frac{1}{2} \right| && (11) \\ &= \frac{1}{2} Adv_{IND}[A, f] && (12) \end{aligned}$$

Since $Adv_{IND}[A, f]$ is non-negligible, so is $\frac{1}{2} Adv_{IND}[A, f]$. Hence, A_{PRF} has non-negligible advantage. Because A_{PRF} has non-negligible advantage, f cannot be a secure PRF. But this contradicts our initial assumption that f is a secure PRF. So by contradiction, counter mode encryption, when based on a secure PRF f, must be semantically secure. □

3 PR-CPA Security

3.1 PR-CPA Adversarial Game

Definition 5. *Let $\mathcal{E} = (KeyGen, E, D)$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$. The PR-CPA game is defined as follows:*

1. *The Challenger runs $k \leftarrow KeyGen(\lambda)$ and samples m from $\mathcal{M}$ uniformly at random. Give $E(k, m)$ to the Adversary.*
2. *The Adversary runs some logic and selects a message m_i from $\mathcal{M}$.*
3. *The Challenger replies with $E(k, m_i)$.*
4. *Repeat steps 2 through 3 for some $poly(log|\mathcal{K}|)$ number of times.*
5. *Finally, the Adversary runs some logic to output $m' \in \mathcal{M}$, which is the output of the experiment.*

3.2 PR-CPA Advantage

Definition 6. *Let $\mathcal{E} = (KeyGen, E, D)$ be defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$, and let A be an poly-time adversary. The PR-CPA advantage is defined as:*

$$Adv_{PR-CPA}[A, \mathcal{E}] := Pr[m = m']$$

where m' is the output of the experiment.

3.3 PR-CPA Security

Definition 7. *An encryption scheme $\mathcal{E}$ is PR-CPA secure if for all efficient A:*

$$Adv_{PR-CPA}[A, \mathcal{E}] < \epsilon$$

4 IND-CPA Secure implies PR-CPA Secure

Proof. We will show that if an encryption scheme is IND-CPA (semantically) secure, then it must also be PR-CPA secure via a proof by reduction.

Let $\mathcal{E} = (KeyGen, E, D)$ be an IND-CPA secure encryption scheme defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$. Suppose for the sake of contradiction that $\mathcal{E}$ is not PR-CPA secure. Then there exists an efficient adversary A_{PR} that can recover the plaintext with non-negligible PR advantage. Given A_{PR}, we can construct an adversary A_{IND} that has a non-negligible semantic security advantage. A_{IND} is as follows:

Algorithm 4: Adversary A_{IND}

1 Choose m_0 from $\mathcal{M}$ and m_1 from $\mathcal{M} \setminus \{m_0\}$.
2 Send ChallengerIND (m_0, m_1) and receive c.
3 Execute A_{PR}
4 Send A_{PR} the ciphertext c.
5 **while** A_{PR} *queries* $x \in \mathcal{M}$ **do**
6   Send ChallengerIND (x, x) and receive $E(k, x) = c'$.
7   Reply to A_{PR} with c'.
8 **end**
9 m' = output of A_{PR}.
10 **if** $m' = m_1$ **then**
11   **return** 1.
12 **else**
13   **return** 0.
14 **end**

We show that A_{IND} is an efficient adversary with a non-negligible advantage.

First, we argue that A_{IND} perfectly simulates the challenger for A_{PR}. On line 4 of our definition of A_{IND}, we send A_{PR} a ciphertext. Then A_{PR} queries A_{IND} for a message x. We use the ChallengerIND to generate $c = E(k, x)$ and reply to A_{PR} with c. We repeat this exchange a polynomial number of times, and then A_{PR}

finally outputs a guess m'. So, this matches the definition of the PR-CPA security game.
Now we calculate the advantage of A_{IND} and show that its is noticeable (non-negligible). Here is our definition of CPA/semantic security advantage:

$$Adv_{SS}[A_{IND}, \mathcal{E}] := |Pr[Exp_{A_{IND},\mathcal{E}}(0) = 1] - Pr[Exp_{A_{IND},\mathcal{E}}(1) = 1]|$$

By construction of A_{IND}, we have:

$$Pr[Exp_{A_{IND},\mathcal{E}}(0) = 1] \leq \frac{1}{2^{|M|}} = negl \tag{13}$$

$$Pr[Exp_{A_{IND},\mathcal{E}}(1) = 1] = Adv_{PR}[A, \mathcal{E}] \tag{14}$$

The first probability is based on the observation that when the challenger for A_{IND} is given a 0 bit, it always encrypts the first message it is sent, which means in step 2 of the algorithm above, we have $c = E(k, m_0)$. This implies that A_{PR} has no information at all about m_1. Hence, the only time that A_{IND} will output 1 is when A_{PR} happens to randomly guess m_1, which happens at most $\frac{1}{2^{|M|}}$ of the time.
The second probability is based on the observation that when the challenger for A_{IND} is given a 0 bit, then we are perfectly playing the PR game with A_{PR}.
Plugging all of this into our equation that defines an adversary's CPA advantage, we have:

$$\begin{aligned} Adv_{IND-CPA}[A_{IND}, \mathcal{E}] &:= |Pr[Exp_{A_{IND},\mathcal{E}}(0) = 1] - Pr[Exp_{A_{IND},\mathcal{E}}(1) = 1]| \\ &\geq Adv_{PR}[A, \mathcal{E}] - \frac{1}{2^{|M|}} \end{aligned}$$

Because we assumed $Adv_{PR}[A, \mathcal{E}]$ is non-negligible, the advantage of A_{IND} is non-negligible, so $\mathcal{E}$ is not IND-CPA (semantically) secure. But this contradicts our initial assumption that $\mathcal{E}$ is IND-CPA secure. So by contradiction, E must be PR secure. Hence, IND-CPA security implies PR-CPA security. $\square$