# Introduction to Cryptography

TEK 4500 (Fall 2021)
Problem Set 3

**Problem 1.**

Read Chapter 5 in [BR] (Sections 5.6 and 5.8 can be skipped. The proofs in Section 5.7 can be skipped on first reading, but it is recommended to have a look at it).

**Problem 2.**

An encryption scheme secure according to the IND-CPA definition is not required to hide the length of the plaintext. This is captured in the formal IND-CPA security experiment (Fig. 1) by having the encryption oracle $\mathcal{E}$ encrypt a random string of *equal* length to the input message $M$ (see Line 1) in the case $b = 0$.

Suppose now that the IND-CPA experiment was modified to *not* select $R$ at random from $\{0,1\}^{|M|}$, but instead select it at random from the entire message space $\mathcal{M}$. Show how you can break *any* encryption scheme $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ that doesn't hide the plaintext length in this variant of IND-CPA. For simplicity, you can assume that the message space consists of all bit strings up to some length, say 1000 bits, i.e., $\mathcal{M} = \bigcup_{i=1}^{1000}\{0,1\}^i$, and that the ciphertext space $\mathcal{C}$ equals $\mathcal{M}$.

**Problem 3.**

The defining equations for CBC encryption are

$$C_0 \xleftarrow{\$} \{0,1\}^n \tag{1}$$

$$C_i \leftarrow \mathcal{E}_K(M_i \oplus C_{i-1}) \qquad i = 1, \ldots, \ell \tag{2}$$

where $\mathcal{E}$ is a block cipher with $n$-bit blocks.

**a)** Write the corresponding equations for CBC decryption.

**b)** Suppose you have a CBC-encrypted ciphertext $C = C_0\|C_1\|C_2\|\ldots\|C_m$ and you flip one bit in block $C_j$. What happens to the decrypted message $M$?

**c)** Suppose you have a CBC-encrypted ciphertext $C = C_0\|C_1\|C_2\|\ldots\|C_m$ and you drop one of the blocks $C_j$. What happens to the decrypted message $M$?

$\underline{\mathbf{Exp}^{\mathsf{ind\text{-}cpa}}_{\Sigma}(\mathcal{A})}$

1: $b \leftarrow \{0,1\}$
2: $K \xleftarrow{\$} \Sigma.\mathsf{KeyGen}$
3: $b' \leftarrow \mathcal{A}^{\mathcal{E}(\cdot)}$
4: **return** $b' \stackrel{?}{=} b$

$\underline{\mathcal{E}(M)}$:

1: $R \xleftarrow{\$} \{0,1\}^{|M|}$
2: $C_0 \leftarrow \Sigma.\mathsf{Enc}(K, R)$
3: $C_1 \leftarrow \Sigma.\mathsf{Enc}(K, M)$
4: **return** $C_b$ // real ciphertext or encryption of random string

$$\mathbf{Adv}^{\mathsf{ind\text{-}cpa}}_{\Sigma}(\mathcal{A}) = \left|2 \cdot \Pr[\mathbf{Exp}^{\mathsf{ind\text{-}cpa}}_{\Sigma}(\mathcal{A}) \Rightarrow \mathsf{true}] - 1\right|$$

**Figure 1:** IND-CPA security experiment.

**Problem 4.** [Problem 5.5 in [BR]]

There's a variant of CBC mode called *CBC-Implicit* where the IV for the first message is chosen at random (like normal CBC), but the IV for each subsequent message is taken from the last block of the previous ciphertext (see Fig. 2). The scheme is probabilistic and stateful. Show that CBC-Implicit is insecure by giving a simple and efficient adversary that breaks it in the IND-CPA sense.

**Note:** Curiously, CBC-Implicit was how CBC was implemented in TLS 1.0. This was patched in TLS 1.1 and higher.

**Problem 5.** [Problem 3.21 in [KL07]]

Let $\Sigma_1 = (\mathsf{KeyGen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ and $\Sigma_2 = (\mathsf{KeyGen}_2, \mathsf{Enc}_2, \mathsf{Dec}_2)$ be two encryption schemes for which it is known that at least one is IND-CPA-secure. The problem is that you don't know which one is IND-CPA-secure and which one may not be. Show how to construct an encryption scheme $\Sigma$ that is *guaranteed* to be IND-CPA-secure as long as at least one of $\Sigma_1$ or $\Sigma_2$ is IND-CPA-secure. Give a high-level justification for why your scheme is secure.

**Problem 6.**

Show that the CBC and CTR modes-of-operation are *not* IND-CCA secure by describing concrete attacks and calculating their IND-CCA advantages. The IND-CCA security experiment is given in Fig. 3.

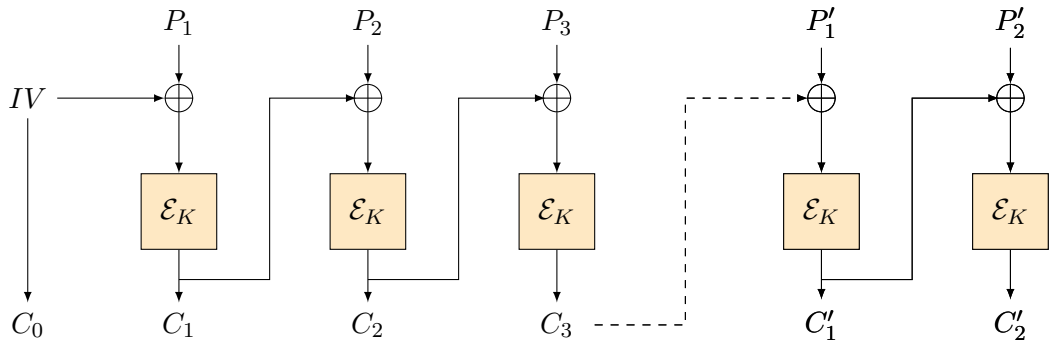


**Figure 2:** CBC-Implicit mode-of-operation illustrated for two messages $M_1 = P_1 \| P_2 \| P_3$ and $M_2 = P'_1 \| P'_2$. The IV used for the second message is the last block of the previous ciphertext, i.e., $C_3$.

$\underline{\mathbf{Exp}^{\text{ind-cca}}_{\Sigma}(\mathcal{A})}$

1: $b \leftarrow \{0,1\}$
2: Ciphertexts $\leftarrow [\,]$
3: $K \xleftarrow{\$} \Sigma.\mathsf{KeyGen}$
4: $b' \leftarrow \mathcal{A}^{\mathcal{E}(\cdot),\mathcal{D}(\cdot)}$
5: **return** $b' \stackrel{?}{=} b$

$\underline{\mathcal{E}(M)}$:

1: $R \xleftarrow{\$} \{0,1\}^{|M|}$
2: $C_0 \leftarrow \Sigma.\mathsf{Enc}(K, R)$
3: $C_1 \leftarrow \Sigma.\mathsf{Enc}(K, M)$
4: Ciphertexts.add$(C_b)$
5: **return** $C_b$ // real ciphertext or encryption of random string

$\underline{\mathcal{D}(C)}$:

1: **if** $C \in$ Ciphertexts **then** // cheating; suppress output
2: **return** $\perp$
3: **return** $\Sigma.\mathsf{Dec}(K, D)$

$$\mathbf{Adv}^{\text{ind-cca}}_{\Sigma}(\mathcal{A})) = \left|2 \cdot \Pr[\mathbf{Exp}^{\text{ind-cca}}_{\Sigma}(\mathcal{A}) \Rightarrow \mathsf{true}] - 1\right|$$

**Figure 3:** IND-CCA security experiment.

**Problem 7.**

Bellare and Rogaway [BR] are using a definition of IND-CPA security which is slightly different from the one we used in class. Specifically, in BR's security experiment the adversary is again tasked with distinguishing between two worlds. However, this time the adversary always submits *two* (equal length) messages $(M_0, M_1)$ to the encryption oracle, and in the "Left" world it always gets back an encryption of the left input $M_0$, while in the "Right" world it always gets back an encryption of the right input $M_1$. This is shown in Fig. 4. For the purposes of this exercise let's call BR's definition Left-or-Right (LoR) security, and the one we used in class (defined in Fig. 1) Real-or-Random (RoR) security.

You will now show that these two definitions are in fact equivalent. That is, an encryption scheme having LoR security also has RoR security, and vice versa.

**a)** Show that LoR security implies RoR security.

**Hint:** Show instead the equivalent contrapositive statement: RoR *insecurity* implies LoR *insecurity*. That is, suppose there was some algorithm $\mathcal{A}$ that could break an encryption scheme $\Sigma$ according to the RoR definition given in Fig. 1, then use this $\mathcal{A}$ to construct *another* adversary $\mathcal{B}$ that breaks $\Sigma$ according to the LoR definition in Fig. 4.

**Hint:** Think of yourself as being the adversary $\mathcal{B}$: you're playing in the LoR experiment and have access to an encryption oracle $\mathcal{E}^{\mathrm{LoR}}(\cdot,\cdot)$ that takes in *two* inputs and returns an encryption of either the left or the right input. Additionally, you also (by hypothesis) have access to some algorithm $\mathcal{A}$ which thinks it plays in the RoR experiment. How $\mathcal{A}$ is represented is not important; you can simply think of it as some piece of code that you can run on your own machine. Once you start running $\mathcal{A}$ it will begin outputting *one* and *one* message, and expect back a response which is either a real encryption or a "fake" encryption of some random data. That is, $\mathcal{A}$ expects to have access to an encryption oracle $\mathcal{E}^{\mathrm{RoR}}(\cdot)$. Eventually, $\mathcal{A}$ will output some bit $b_{RoR}$ as its guess of which world it thinks it's in. How can you answer $\mathcal{A}$'s queries so that it believes that it's interacting with the $\mathcal{E}^{\mathrm{RoR}}(\cdot)$ oracle of the actual RoR experiment (Fig. 1)?

**Hint:** Use your access to the $\mathcal{E}^{\mathrm{LoR}}(\cdot,\cdot)$ oracle: when $\mathcal{A}$ makes a query $M$, submit this as the "left" input to $\mathcal{E}^{\mathrm{LoR}}(\cdot,\cdot)$. What should you use as the "right" input? Return the response from the $\mathcal{E}^{\mathrm{LoR}}(\cdot,\cdot)$ oracle back to $\mathcal{A}$.

**Hint:** Let your output in the LoR experiment simply equal $\mathcal{A}$'s output, i.e., $b_{LoR} = b_{RoR}$. If $\mathcal{A}$ has RoR advantage $\mathbf{Adv}^{\mathsf{ror\text{-}ind\text{-}cpa}}_{\Sigma}(\mathcal{A})$, what's your LoR advantage $\mathbf{Adv}^{\mathsf{lor\text{-}ind\text{-}cpa}}_{\Sigma}(\mathcal{B})$?

**b)** Show that RoR security implies LoR security.

**Hint:** Prove the contrapositive: LoR *insecurity* implies RoR *insecurity*.

$\underline{\mathbf{Exp}^{\text{lor-ind-cpa}}_{\Sigma}(\mathcal{A})}$

1: $b \leftarrow \{0,1\}$
2: $K \xleftarrow{\$} \Sigma.\mathsf{KeyGen}$
3: $b' \leftarrow \mathcal{A}^{\mathcal{E}^{\text{LoR}}(\cdot,\cdot)}$
4: **return** $b' \stackrel{?}{=} b$

$\underline{\mathcal{E}^{\text{LoR}}(M_0, M_1)}$:

1: **if** $|M_0| \neq |M_1|$ **then** // Adversary's cheating! Suppress output
2: **return** $\perp$
3: **return** $\Sigma.\mathsf{Enc}(K, M_b)$ // Either encryption of $M_0$ or $M_1$

$$\mathbf{Adv}^{\text{lor-ind-cpa}}_{\Sigma}(\mathcal{A}) = \left|2 \cdot \Pr[\mathbf{Exp}^{\text{lor-ind-cpa}}_{\Sigma}(\mathcal{A}) \Rightarrow \mathsf{true}] - 1\right|$$

**Figure 4:** The Left-or-Right IND-CPA security experiment as defined in [BR].

**Hint:** Suppose some algorithm $\mathcal{A}$ is able to break $\Sigma$ according to the LoR definition of Fig. 4. Create an RoR adversary $\mathcal{B}$, having access to an encryption oracle $\mathcal{E}^{\text{RoR}}(\cdot)$, that uses $\mathcal{A}$ to break $\Sigma$ according to the RoR definition of Fig. 1.

**Hint:** Once you start running $\mathcal{A}$, it will begin submitting *pairs* of messages $(M_0, M_1)$, expecting to get back an encryption of one of them. How can you answer these queries, given that you only have at your disposal an encryption oracle $\mathcal{E}^{\text{RoR}}(\cdot)$ which takes a *single* input?

**Hint:** Define an *internal* bit $b_{sim}$ that you flip yourself (once).When $\mathcal{A}$ makes a query $(M_0, M_1)$, forward $M_{b_{sim}}$ to your encryption oracle $\mathcal{E}^{\text{RoR}}(\cdot)$ and return the result back to $\mathcal{A}$. When $\mathcal{A}$ eventually outputs a bit $b_{LoR}$, what should you output to your own RoR experiment?

**Hint:** Make a comparison between $b_{LoR}$ and the internal bit $b_{sim}$ that you created yourself. From $\mathcal{A}$'s point-of-view, what does the cases of $b_{sim}$ correspond to?

**Hint:** If $\mathcal{A}$ has LoR advantage $\mathbf{Adv}^{\text{lor-ind-cpa}}_{\Sigma}(\mathcal{A})$, then your RoR advantage should be:

$$\mathbf{Adv}^{\text{ror-ind-cpa}}_{\Sigma}(\mathcal{B}) \geq \frac{\mathbf{Adv}^{\text{lor-ind-cpa}}_{\Sigma}(\mathcal{A})}{2}$$

## References


[BR] Mihir Bellare and Phillip Rogaway. *Introduction to Modern Cryptography*. https://web.cs.ucdavis.edu/~rogaway/classes/227/spring05/book/main.pdf.

[KL07] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press, 2007.

[Ros] Mike Rosulek. *The Joy of Cryptography*, (draft Feb 6, 2020). https://web.engr.oregonstate.edu/~rosulekm/crypto/crypto.pdf.