

Cryptography

Lecture 7

Announcements

- HW3 up on course webpage, due Wednesday, 2/23

Agenda

- Last time:
 - SKE secure against eavesdroppers from PRG (K/L 3.3)
- This time:
 - Stream Ciphers
 - CPA Security (K/L 3.4)
 - Pseudorandom Functions (PRF) (K/L 3.5)

Stream Cipher

Sender

State s_i after sending the i -th message:

$$\begin{aligned} s_0 &:= k \\ s_{i+1} &:= G(s_i)_2, \dots, G(s_i)_{n+1} \\ pad_{i+1} &:= G(s_i)_1 \end{aligned}$$

$$c_{i+1} := m_{i+1} \oplus pad_{i+1}$$



Receiver

State s_i after receiving the i -th message:

$$\begin{aligned} s_0 &:= k \\ s_{i+1} &:= G(s_i)_2, \dots, G(s_i)_{n+1} \\ pad_{i+1} &:= G(s_i)_1 \end{aligned}$$

$$m_{i+1} := c_{i+1} \oplus pad_{i+1}$$

CPA Security

Consider a private-key encryption scheme $\Pi = (Gen, Enc, Dec)$, any adversary A , and any value n for the security parameter.

Experiment $PrivK_{A,\Pi}^{cpa}(n)$

Adversary $A(1^n)$

Challenger

CPA Security

Consider a private-key encryption scheme $\Pi = (Gen, Enc, Dec)$, any adversary A , and any value n for the security parameter.

Experiment $PrivK_{A,\Pi}^{cpa}(n)$

Adversary $A(1^n)$

Challenger

$k \leftarrow Gen(1^n)$

CPA Security

Consider a private-key encryption scheme $\Pi = (Gen, Enc, Dec)$, any adversary A , and any value n for the security parameter.

Experiment $PrivK_{A,\Pi}^{cpa}(n)$

Adversary $A(1^n)$

$A^{Enc_k}(\cdot)$

Challenger

$k \leftarrow Gen(1^n)$

CPA Security

Consider a private-key encryption scheme $\Pi = (Gen, Enc, Dec)$, any adversary A , and any value n for the security parameter.

Experiment $PrivK_{A,\Pi}^{cpa}(n)$

Adversary $A(1^n)$

Challenger

$A^{Enc_k(\cdot)}$

m'

$k \leftarrow Gen(1^n)$



CPA Security

Consider a private-key encryption scheme $\Pi = (Gen, Enc, Dec)$, any adversary A , and any value n for the security parameter.

Experiment $PrivK_{A,\Pi}^{cpa}(n)$

Adversary $A(1^n)$

Challenger

$A^{Enc_k(\cdot)}$

$k \leftarrow Gen(1^n)$

m'

c'

CPA Security

Consider a private-key encryption scheme $\Pi = (Gen, Enc, Dec)$, any adversary A , and any value n for the security parameter.

Experiment $PrivK_{A,\Pi}^{cpa}(n)$

Adversary $A(1^n)$

Challenger

$A^{Enc_k(\cdot)}$

m'

c'

\vdots

$k \leftarrow Gen(1^n)$

CPA Security

Consider a private-key encryption scheme $\Pi = (Gen, Enc, Dec)$, any adversary A , and any value n for the security parameter.

Experiment $PrivK_{A,\Pi}^{cpa}(n)$

Adversary $A(1^n)$

Challenger

$A^{Enc_k(\cdot)}$

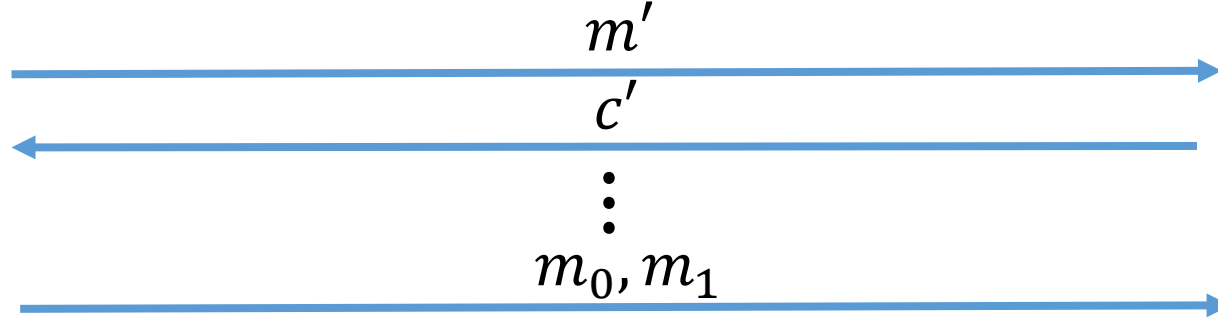
$k \leftarrow Gen(1^n)$

m'

c'

\vdots

m_0, m_1



CPA Security

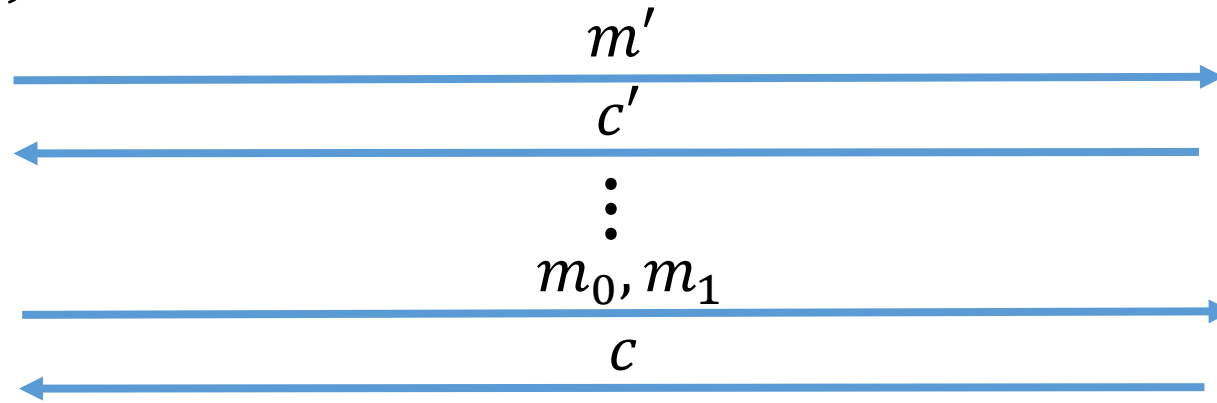
Consider a private-key encryption scheme $\Pi = (Gen, Enc, Dec)$, any adversary A , and any value n for the security parameter.

Experiment $PrivK_{A,\Pi}^{cpa}(n)$

Adversary $A(1^n)$

Challenger

$A^{Enc_k(\cdot)}$



$k \leftarrow Gen(1^n)$

$b \leftarrow \{0,1\}$

$c \leftarrow Enc_k(m_b)$

CPA Security

Consider a private-key encryption scheme $\Pi = (Gen, Enc, Dec)$, any adversary A , and any value n for the security parameter.

Experiment $PrivK_{A,\Pi}^{cpa}(n)$

Adversary $A(1^n)$

Challenger

$A^{Enc_k(\cdot)}$

m'

$k \leftarrow Gen(1^n)$

c'

\vdots

m_0, m_1

$b \leftarrow \{0,1\}$

c

$c \leftarrow Enc_k(m_b)$

$A^{Enc_k(\cdot)}$

m'

c'

\vdots

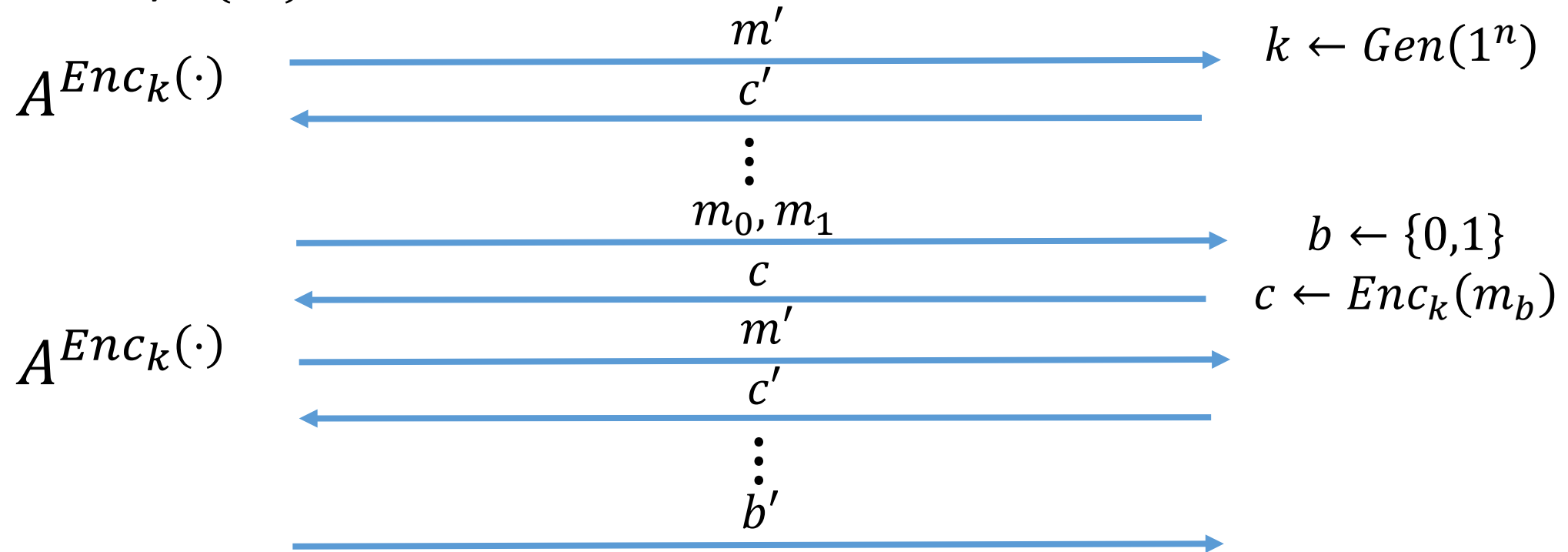
CPA Security

Consider a private-key encryption scheme $\Pi = (Gen, Enc, Dec)$, any adversary A , and any value n for the security parameter.

Experiment $PrivK_{A,\Pi}^{cpa}(n)$

Adversary $A(1^n)$

Challenger



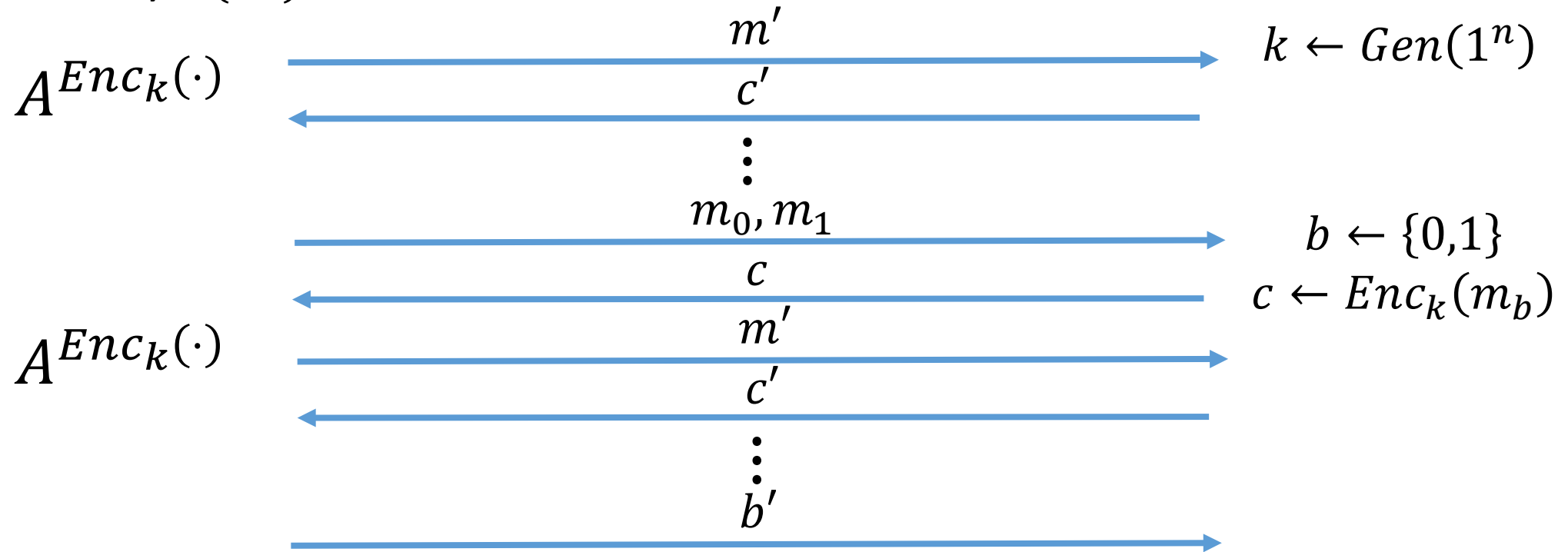
CPA Security

Consider a private-key encryption scheme $\Pi = (Gen, Enc, Dec)$, any adversary A , and any value n for the security parameter.

Experiment $PrivK_{A,\Pi}^{cpa}(n)$

Adversary $A(1^n)$

Challenger



$PrivK_{A,\Pi}^{cpa}(n) = 1$ if $b' = b$ and $PrivK_{A,\Pi}^{cpa}(n) = 0$ if $b' \neq b$.

CPA-Security

The CPA Indistinguishability Experiment $PrivK^{cpa}_{A,\Pi}(n)$:

1. A key k is generated by running $Gen(1^n)$.
2. The adversary A is given input 1^n and oracle access to $Enc_k(\cdot)$, and outputs a pair of messages m_0, m_1 of the same length.
3. A random bit $b \leftarrow \{0,1\}$ is chosen, and then a challenge ciphertext $c \leftarrow Enc_k(m_b)$ is computed and given to A .
4. The adversary A continues to have oracle access to $Enc_k(\cdot)$, and outputs a bit b' .
5. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.

CPA-Security

Definition: A private-key encryption scheme $\Pi = (Gen, Enc, Dec)$ has indistinguishable encryptions under a chosen-plaintext attack if for all ppt adversaries A there exists a negligible function $negl$ such that

$$\Pr \left[PrivK^{cpa}_{A, \Pi}(n) = 1 \right] \leq \frac{1}{2} + negl(n),$$

where the probability is taken over the random coins used by A , as well as the random coins used in the experiment.

CPA-security for multiple encryptions

Theorem: Any private-key encryption scheme that has indistinguishable encryptions under a chosen-plaintext attack also has indistinguishable multiple encryptions under a chosen-plaintext attack.

CPA-secure Encryption Must Be Probabilistic

Theorem: If $\Pi = (Gen, Enc, Dec)$ is an encryption scheme in which Enc is a deterministic function of the key and the message, then Π cannot be CPA-secure.

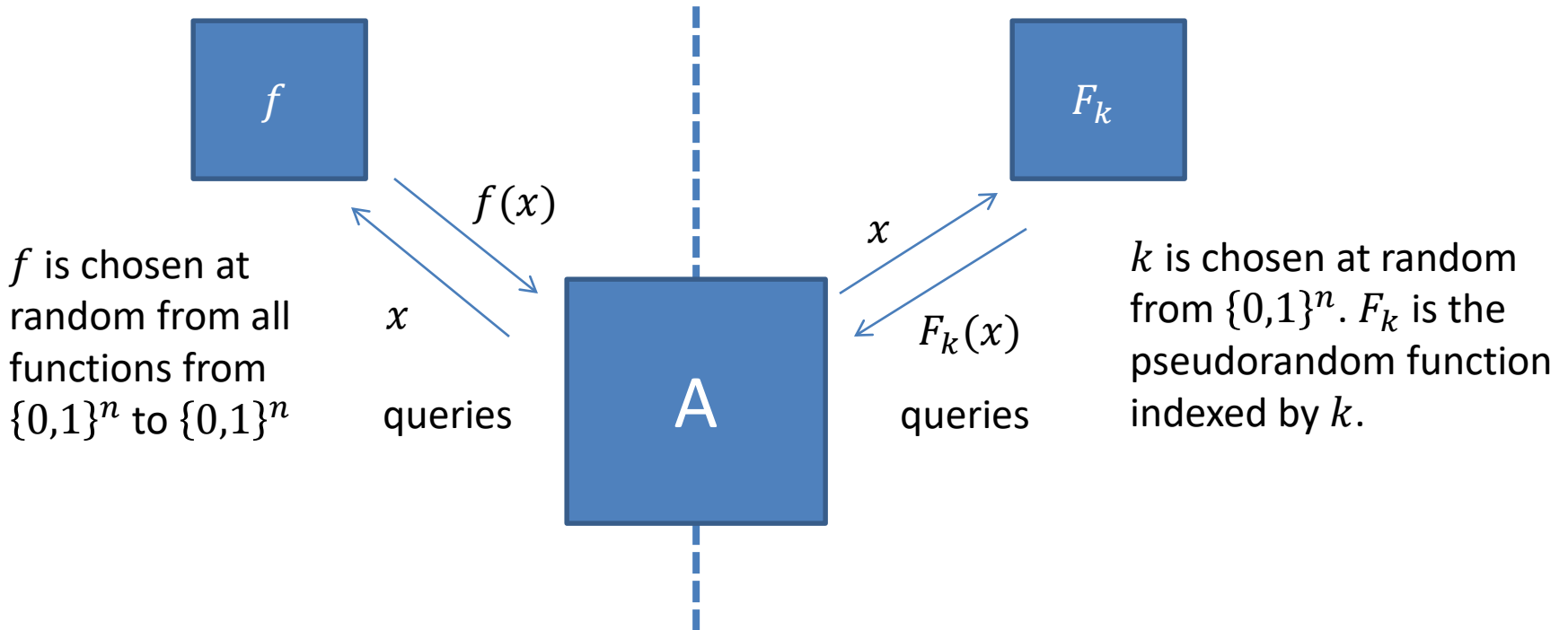
Why not?

Constructing CPA-Secure Encryption Scheme

Pseudorandom Function

Definition: A keyed function $F: \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$ is a two-input function, where the first input is called the key and denoted k .

Pseudorandom Function (PRF)



PRF: Any efficient A cannot tell which world it is in.

$$|\Pr[A^f() = 1] - \Pr[A^{F_k}() = 1]| \leq \textit{negligible}$$

Pseudorandom Function

Definition: Let $F: \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$ be an efficient, length-preserving, keyed function. We say that F is a pseudorandom function if for all ppt distinguishers D , there exists a negligible function $negl$ such that:

$$\left| \Pr[D^{F_k(\cdot)}(1^n) = 1] - \Pr[D^{f(\cdot)}(1^n) = 1] \right| \leq negl(n).$$

where $k \leftarrow \{0,1\}^n$ is chosen uniformly at random and f is chosen uniformly at random from the set of all functions mapping n -bit strings to n -bit strings.