Introduction to Modern Cryptography

Michele Ciampi

(Slides courtesy of Prof. Jonathan Katz)

Lecture 8

CPA-secure Encryption from PRF

CPA-security (recall)

Experiment $\mathsf{PrivK}^{\mathbf{cpa}}_{A,\Pi}(n)$

Fix Π, A. Define a randomized experiment $\mathsf{PrivK}^{\mathbf{cpa}}_{A,\Pi}(n)$:

- $k \leftarrow \mathsf{Gen}(1^n)$
- $A(1^n)$ interacts with an encryption oracle $\mathsf{Enc}_k(\cdot)$, and then outputs m_0, m_1 of the same length
- $b \leftarrow \{0,1\}$, $c \leftarrow \mathsf{Enc}_k(m_b)$, give c to A
- A can continue to interact with $\mathsf{Enc}_k(\cdot)$
- A outputs b'; A succeeds if $b = b'$, and the experiment evaluates to 1 in this case

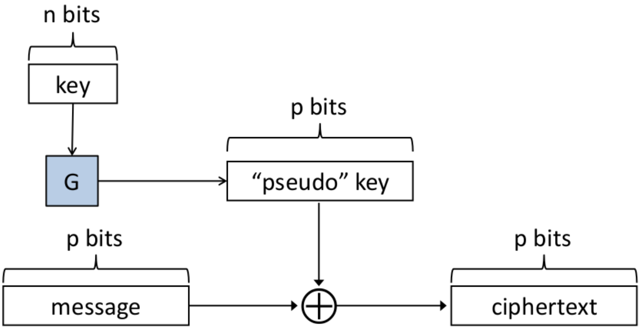
CPA-security (recall)

Security Against Chosen-plaintext Attacks

$\mathbf{\Pi}$ is secure against chosen-plaintext attacks (CPA-secure) if for all PPT attackers $\boldsymbol{A}$, there is a negligible function $\boldsymbol{\epsilon}$ such that

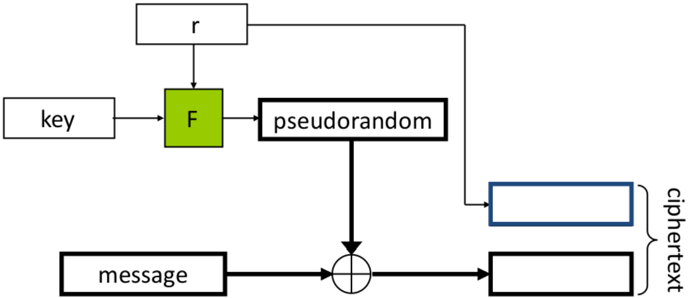
$$\mathbf{Pr}[\mathsf{PrivK}^{\mathbf{cpa}}_{\boldsymbol{A},\mathbf{\Pi}}(\boldsymbol{n}) = \mathbf{1}] \leq \frac{\mathbf{1}}{\mathbf{2}} + \boldsymbol{\epsilon}(\boldsymbol{n})$$

EAV-secure Encryption (POTP) (recall)

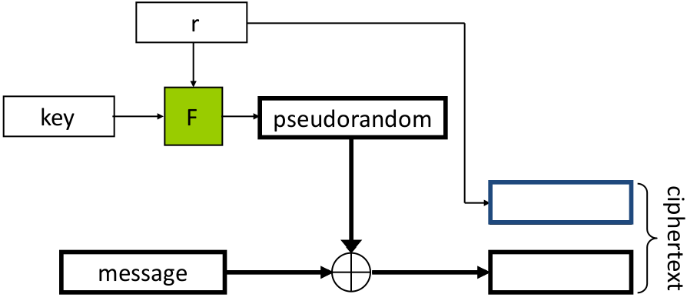


- Solves OTP limitation 1 (key as long as the message)
- Not solve OTP limitation 2 (key used only once)
- EAV-secure, but **not** CPA-secure

CPA-secure Encryption



CPA-secure Encryption



- Not solve OTP limitation 1 (key as long as the message)
- Solves OTP limitation 2 (key used only once)
- $\implies$ CPA-secure $\implies$ EAV-secure

CPA-secure Encryption (Formal)

Encryption Scheme Π

Let F be a length-preserving keyed function.

- $\mathsf{Gen}(1^n)$: choose a uniform key $k \in \{0,1\}^n$
- $\mathsf{Enc}_k(m)$, where $|m| = |k| = n$:
  - Choose uniform $r \in \{0,1\}^n$ (nonce/initialization vector)
  - Output ciphertext $\langle r,\ F_k(r) \oplus m \rangle$
- $\mathsf{Dec}_k(c_1, c_2)$: output $c_2 \oplus F_k(c_1)$
- Correctness is immediate

- The key is as long as the message...
- ...but the same key can be used to securely encrypt multiple messages

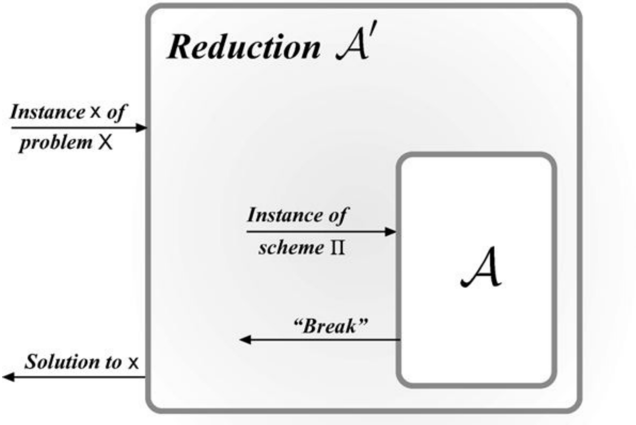
Security?

Theorem

If $\boldsymbol{F}$ is a pseudorandom function, then $\mathbf{\Pi}$ is CPA-secure

$\Longrightarrow$ proof by reduction

Proof by Reduction



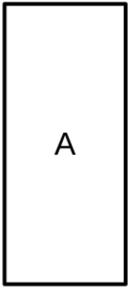
IMC Textbook 2nd ed. CRC Press 2015

Proof by Reduction

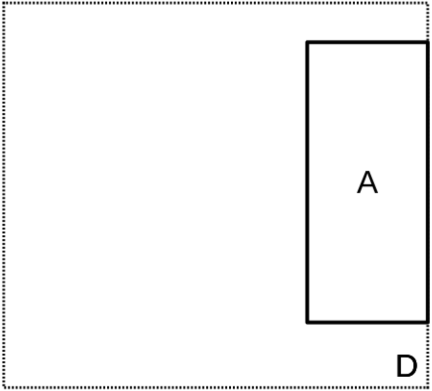
High level

- Attacker $\boldsymbol{A}$ attacks $\boldsymbol{\Pi}$ (as was defined)
- Design distinguisher $\boldsymbol{D}$ that uses $\boldsymbol{A}$ as a subroutine to attack the PRF $\boldsymbol{F}$
  - i.e. $\boldsymbol{D}$ tries to distinguish $\boldsymbol{F}$ from a random function (RF)
- $\boldsymbol{D}$ simulates to $\boldsymbol{A}$ the steps in the $\mathsf{PrivK}^{\mathbf{cpa}}_{\boldsymbol{A},\boldsymbol{\Pi}}(\boldsymbol{n})$ experiment for $\boldsymbol{F}$ and for a RF
- Relate the success $\mathbf{Pr}$ of $\boldsymbol{A}$ to the success $\mathbf{Pr}$ of $\boldsymbol{D}$
- If $\boldsymbol{A}$ succeeds $\implies$ $\boldsymbol{D}$ succeeds $\implies$ $\boldsymbol{F} \neq$ PRF
- contradicts $\boldsymbol{F}$ PRF $\implies$ $\boldsymbol{A}$ can not succeed $\implies$ $\boldsymbol{\Pi}$ CPA-secure

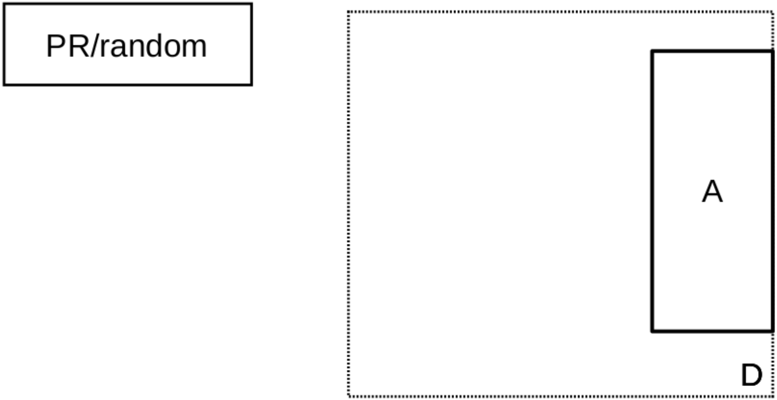
The Reduction



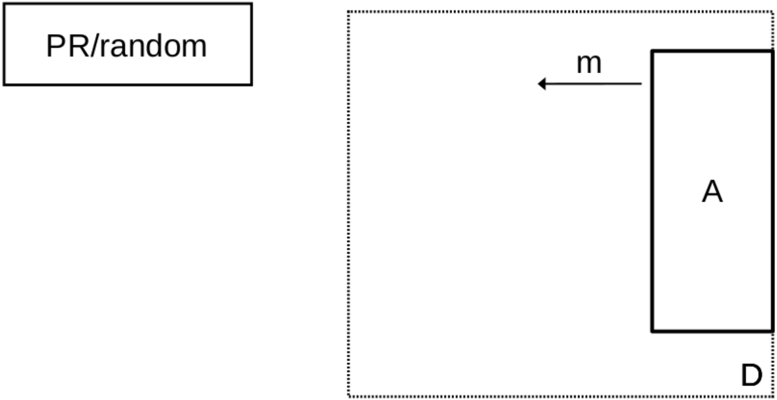
The Reduction



The Reduction

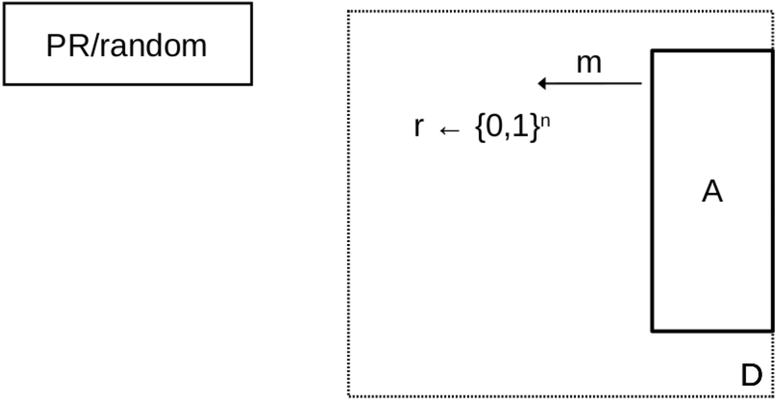


The Reduction



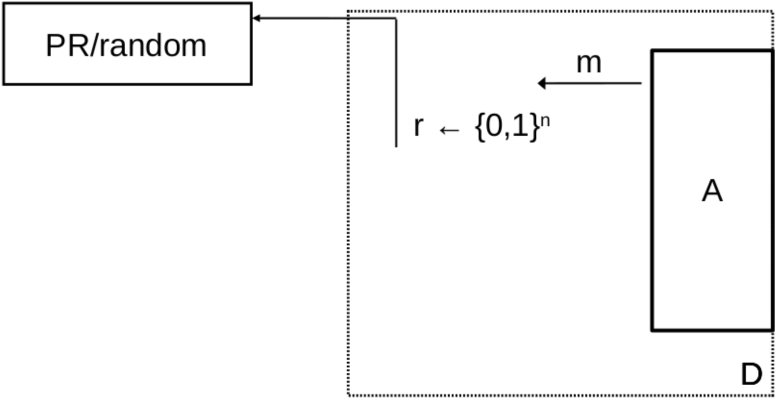
A interacts with an encryption oracle simulated by ***D***

The Reduction



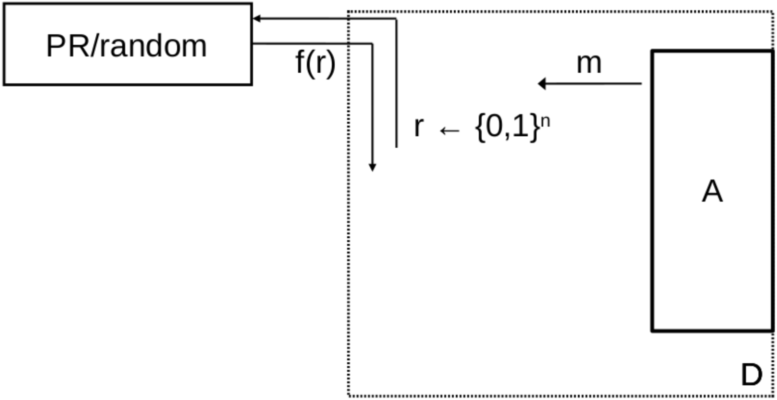
A interacts with an encryption oracle simulated by ***D***

The Reduction



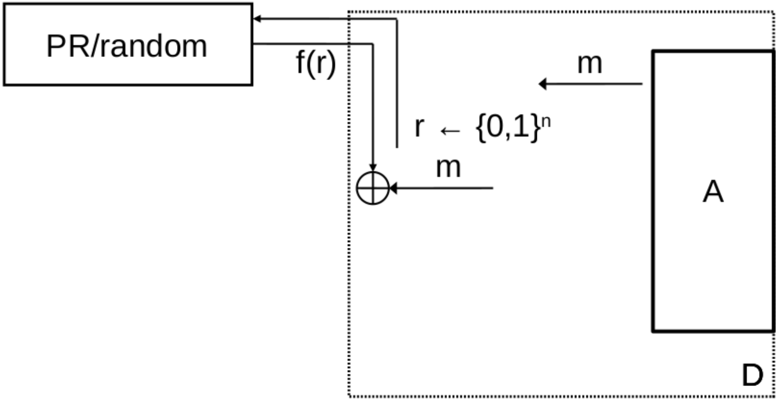
A interacts with an encryption oracle simulated by ***D***

The Reduction



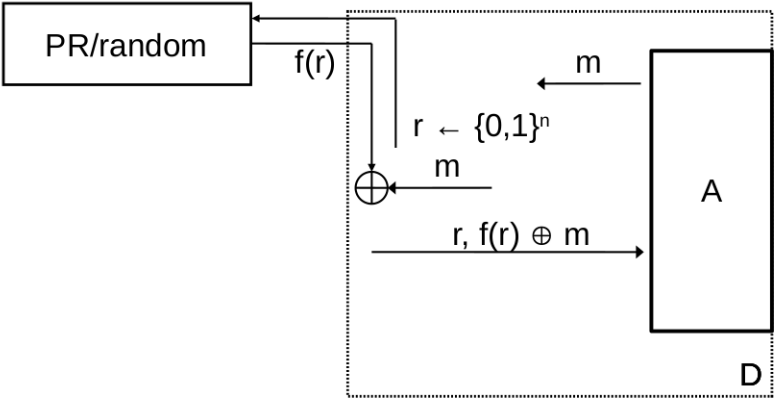
A interacts with an encryption oracle simulated by ***D***

The Reduction



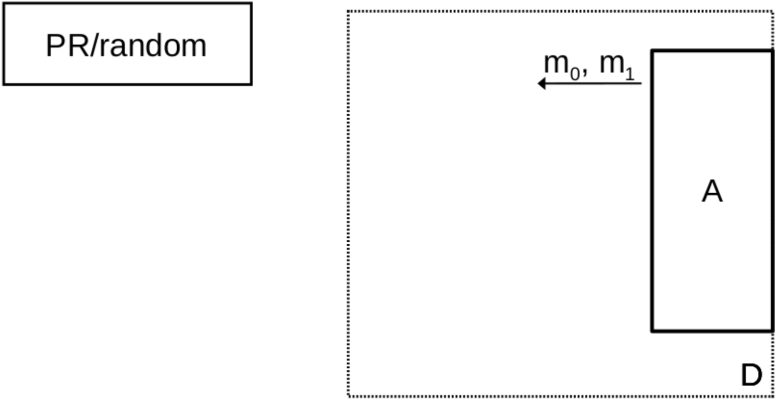
A interacts with an encryption oracle simulated by ***D***

The Reduction



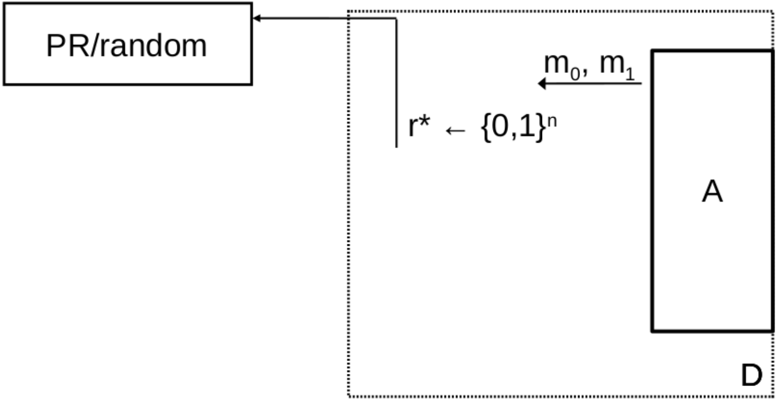
A interacts with an encryption oracle simulated by ***D***

The Reduction



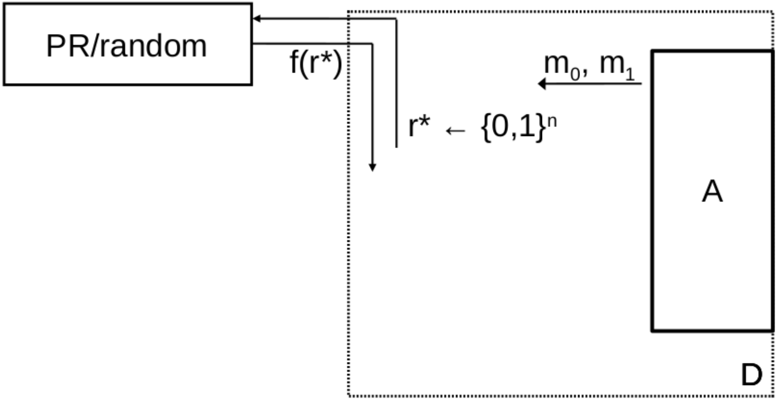
$\boldsymbol{A}$ outputs two messages $\boldsymbol{m_0, m_1}$

The Reduction



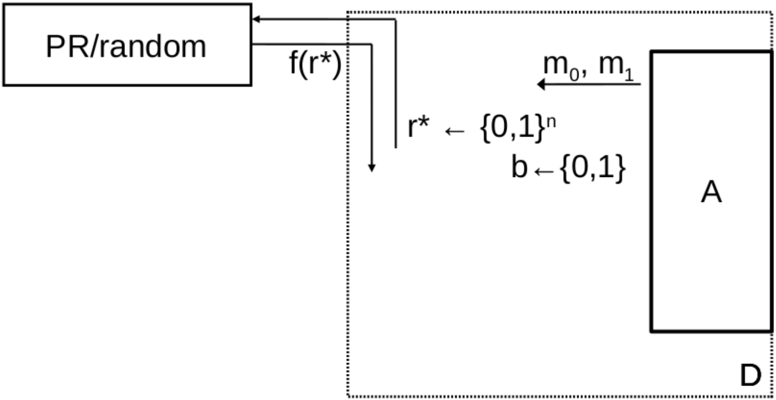
$\boldsymbol{D}$ simulates the encryption oracle for $\boldsymbol{m_b}$

The Reduction



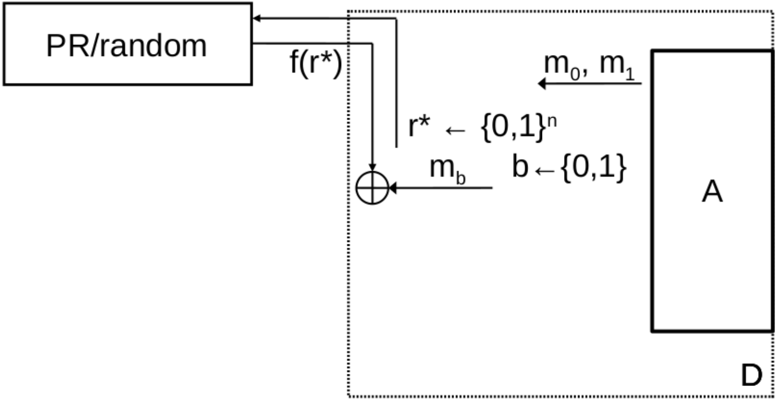
D simulates the encryption oracle for $\boldsymbol{m_b}$

The Reduction



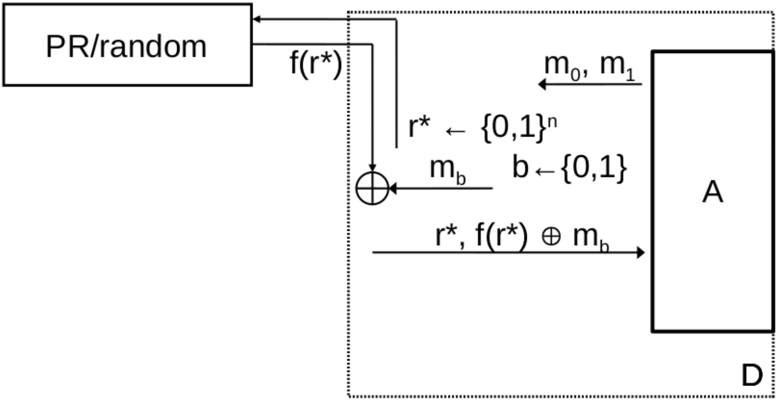
$\boldsymbol{D}$ simulates the encryption oracle for $\boldsymbol{m_b}$

The Reduction



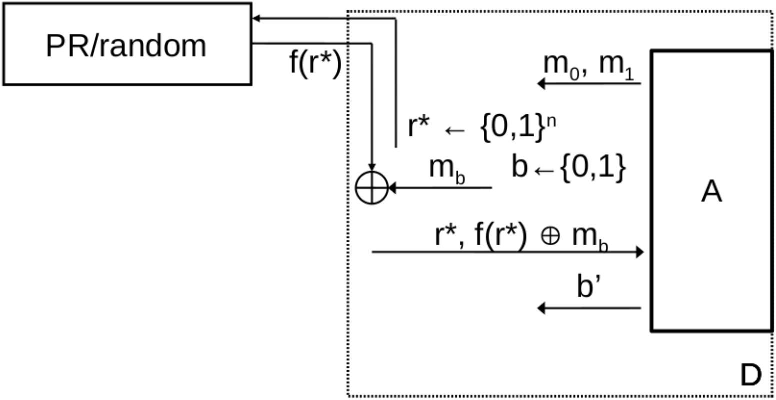
D simulates the encryption oracle for $\boldsymbol{m_b}$

The Reduction



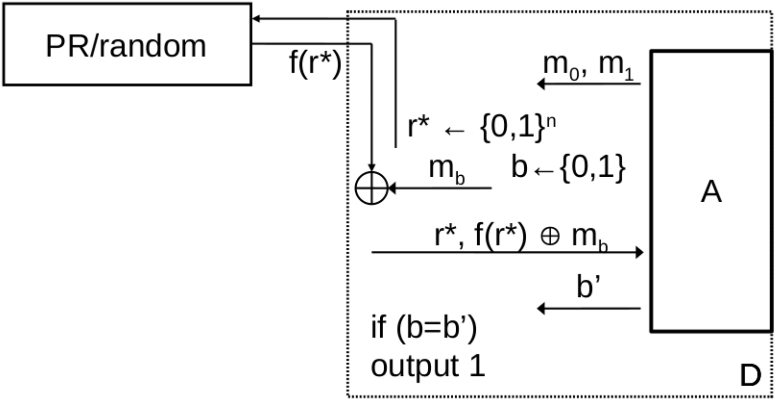
D simulates the encryption oracle for m_b

The Reduction



A outputs its result ***b′***

The Reduction



D outputs $\mathbf{1}$ if $b = b'$

CPA-security Proof

High level

- Replace F_k with a random function f and denote the modified scheme $\widetilde{\Pi}$
- Whenever f is evaluated on a new input, the result is uniform and independent of everything else
- Prove security assuming f is never evaluated on the same input twice
- Argue that f is never evaluated on the same input except with negligible probability

The Distinguisher $\boldsymbol{D}$ Using $\boldsymbol{A}$ as a Subroutine

$\boldsymbol{D}$ simulates to $\boldsymbol{A}$ the steps in the $\mathsf{PrivK}^{\mathbf{cpa}}_{\boldsymbol{A},\widetilde{\boldsymbol{\Pi}}}(\boldsymbol{n})$ and $\mathsf{PrivK}^{\mathbf{cpa}}_{\boldsymbol{A},\boldsymbol{\Pi}}(\boldsymbol{n})$ experiments

World $\mathbf{0}$: $\boldsymbol{D}$ simulates $\mathsf{PrivK}^{\mathbf{cpa}}_{\boldsymbol{A},\widetilde{\boldsymbol{\Pi}}}(\boldsymbol{n})$

- $\boldsymbol{D}$ is given access to a RF $\boldsymbol{f} \in \boldsymbol{\mathcal{F}_n}$
- As if $\boldsymbol{A}$ is interacting with the OTP

World $\mathbf{1}$: $\boldsymbol{D}$ simulates $\mathsf{PrivK}^{\mathbf{cpa}}_{\boldsymbol{A},\boldsymbol{\Pi}}(\boldsymbol{n})$

- $\boldsymbol{D}$ is given access to the PRF $\boldsymbol{F_k}$
- As if $\boldsymbol{A}$ is interacting with $\boldsymbol{\Pi}$

World $\mathbf{0}$: D with a Truly Random Function

D^f simulates $\mathsf{PrivK}^{\mathbf{cpa}}_{A,\widetilde{\Pi}}(n)$ for $A(1^n)$ (truly random f)

- A interacts with $\mathcal{O}$ for $i = 1, 2, \ldots, q(n)$: choose m_i
- Simulation:
  1. D generates $r_i \leftarrow \{0,1\}^n$
  2. D queries f on r_i: gets $f(r_i)$
  3. D computes $c_i = m_i \oplus f(r_i)$; sends (r_i, c_i) to A
- A outputs (m_0, m_1)
- Simulation:
  1. D generates $b \leftarrow \{0,1\}$
  2. D generates $r_c \leftarrow \{0,1\}^n$; gets $f(r_c)$
  3. D computes $c = m_b \oplus f(r_c)$; sends (r_c, c) to A
- A continues to interact with $\mathcal{O}$
- $b' \leftarrow A(c)$
- If $b = b'$ then $D(y) = 1$

World $\mathbf{0}$: $\boldsymbol{D}$ with a Truly Random Function

$\boldsymbol{D}$ simulates $\mathsf{PrivK}^{\mathbf{cpa}}_{\boldsymbol{A},\widetilde{\boldsymbol{\Pi}}}$ for $\boldsymbol{A}$

Let $\boldsymbol{r_c}$ be the random value used in generating the challenge ciphertext $\boldsymbol{c}$:

$$\boldsymbol{c = \widetilde{E}_k(m_b) = m_b \oplus f(r_c)}$$

Two cases

1. $\boldsymbol{r_c}$ was used in **at least one** previous query of $\boldsymbol{A}$ (event `Repeat`)
2. $\boldsymbol{r_c}$ was used in **none** of the previous queries of $\boldsymbol{A}$

World $\mathbf{0}$: $\boldsymbol{D}$ with a Truly Random Function

Case 1: $\boldsymbol{r_c}$ used before (`Repeat`)

- $\boldsymbol{A}$ has a pair $(\boldsymbol{m'}, \boldsymbol{c'})$ s.t. $\boldsymbol{c' = m' \oplus f(r_c)}$
- $\boldsymbol{A}$ computes $\boldsymbol{f(r_c) = m' \oplus c'}$
- $\boldsymbol{A}$ computes $\boldsymbol{m_b = c \oplus f(r_c)}$
- $\boldsymbol{A}$ succeeds with

$$\mathbf{Pr}[\mathsf{PrivK}^{\mathbf{cpa}}_{\boldsymbol{A},\widetilde{\boldsymbol{\Pi}}}(\boldsymbol{n}) = \mathbf{1}] = \mathbf{1}$$

World $\mathbf{0}$: $\boldsymbol{D}$ with a Truly Random Function

Case 2: $\boldsymbol{r_c}$ not used before ($\neg$`Repeat`)

- $\boldsymbol{r_c}$ random $\implies$ $\boldsymbol{f(r_c)}$ random
- $\boldsymbol{A}$ learns nothing from its interaction with $\boldsymbol{f}$
- $\implies$ $\widetilde{\boldsymbol{E}}_{\boldsymbol{k}}(\boldsymbol{m_b}) = \boldsymbol{m_b} \oplus \boldsymbol{f(r_c)}$ is equivalent to OTP
- $\boldsymbol{A}$ succeeds with

$$\mathbf{Pr}[\mathsf{PrivK}^{\mathbf{cpa}}_{\boldsymbol{A},\widetilde{\boldsymbol{\Pi}}}(\boldsymbol{n}) = \mathbf{1}] = \mathbf{Pr}[\mathsf{PrivK}_{\boldsymbol{A},\boldsymbol{OTP}} = \mathbf{1}] = \frac{\mathbf{1}}{\mathbf{2}}$$

World $\mathbf{0}$: $\boldsymbol{D}$ with a Truly Random Function

$\mathbf{Pr}[\texttt{Repeat}]$ and $\mathbf{Pr}[\neg\texttt{Repeat}]$

- $\boldsymbol{A}$ is PPT $\implies$ $\boldsymbol{A}$ can make at most $\boldsymbol{q(n)}$ polynomial number of queries
- As $\boldsymbol{r_c}$ is chosen unifromly, it follows that

$$\mathbf{Pr}[\texttt{Repeat}] = \frac{q(n)}{2^n}$$

$$\mathbf{Pr}[\neg\texttt{Repeat}] = 1 - \frac{q(n)}{2^n} = 1 - \mathbf{negl} \approx 1$$

World $\mathbf{0}$: $\boldsymbol{D}$ with a Truly Random Function

$\mathbf{Pr}[\mathsf{PrivK}^{\mathbf{cpa}}_{\boldsymbol{A},\widetilde{\mathbf{\Pi}}}(\boldsymbol{n}) = \mathbf{1}]$

$$
\begin{aligned}
&\mathbf{Pr}[\mathsf{PrivK}^{\mathbf{cpa}}_{\boldsymbol{A},\widetilde{\mathbf{\Pi}}}(\boldsymbol{n}) = \mathbf{1}] \\
&\overset{LTP}{=} \mathbf{Pr}[(\mathsf{PrivK}^{\mathbf{cpa}}_{\boldsymbol{A},\widetilde{\mathbf{\Pi}}}(\boldsymbol{n}) = \mathbf{1}) \wedge \mathtt{Repeat}] + \\
&\quad \mathbf{Pr}[(\mathsf{PrivK}^{\mathbf{cpa}}_{\boldsymbol{A},\widetilde{\mathbf{\Pi}}}(\boldsymbol{n}) = \mathbf{1}) \wedge \neg\mathtt{Repeat}] \\
&\overset{Cond.P.}{=} \mathbf{Pr}[(\mathsf{PrivK}^{\mathbf{cpa}}_{\boldsymbol{A},\widetilde{\mathbf{\Pi}}}(\boldsymbol{n}) = \mathbf{1}) | \mathtt{Repeat}]\ \mathbf{Pr}[\mathtt{Repeat}] + \\
&\quad \mathbf{Pr}[(\mathsf{PrivK}^{\mathbf{cpa}}_{\boldsymbol{A},\widetilde{\mathbf{\Pi}}}(\boldsymbol{n}) = \mathbf{1}) | \neg\mathtt{Repeat}]\ \mathbf{Pr}[\neg\mathtt{Repeat}] \\
&\leq \mathbf{Pr}[\mathtt{Repeat}] + \mathbf{Pr}[(\mathsf{PrivK}^{\mathbf{cpa}}_{\boldsymbol{A},\widetilde{\mathbf{\Pi}}}(\boldsymbol{n}) = \mathbf{1}) | \neg\mathtt{Repeat}] \\
&= \frac{\boldsymbol{q}(\boldsymbol{n})}{\mathbf{2}^{\boldsymbol{n}}} + \frac{\mathbf{1}}{\mathbf{2}}
\end{aligned}
$$

World $\mathbf{1}$: $\boldsymbol{D}$ with a Pseudorandom Function

D^{F_k} simulates $\mathsf{PrivK}^{\mathbf{cpa}}_{A,\Pi}(n)$ for $A(1^n)$ (pseudorandom F_k)

- A interacts with $\mathcal{O}$ for $i = 1, 2, \ldots, q(n)$: choose m_i
- Simulation:
  1. D generates $r_i \leftarrow \{0,1\}^n$
  2. D queries F_k on r_i: gets $F_k(r_i)$
  3. D computes $c_i = m_i \oplus F_k(r_i)$; sends (r_i, c_i) to A
- A outputs (m_0, m_1)
- Simulation:
  1. D generates $b \leftarrow \{0,1\}$
  2. D generates $r_c \leftarrow \{0,1\}^n$; gets $F_k(r_c)$
  3. D computes $c = m_b \oplus F_k(r_c)$; sends (r_c, c) to A
- A continues to interact with $\mathcal{O}$
- $b' \leftarrow A(c)$
- If $b = b'$ then $D(y) = 1$

World $\mathbf{1}$: $\boldsymbol{D}$ with a Pseudorandom Function

$\boldsymbol{D}$ simulates $\mathsf{PrivK}^{\mathbf{cpa}}_{\boldsymbol{A},\boldsymbol{\Pi}}$ for $\boldsymbol{A}$

The **Pr** with which $\boldsymbol{A}$ succeeds in this case is

$$\mathbf{Pr}[\mathsf{PrivK}^{\mathbf{cpa}}_{\boldsymbol{A},\boldsymbol{\Pi}}(\boldsymbol{n}) = \mathbf{1}]$$

Note

This is the **Pr** that we want to bound!

Proof.

By the assumption that $\boldsymbol{F}$ is a PRF $\exists\boldsymbol{\epsilon(n) = \mathbf{negl}}$:

$$|\mathbf{Pr}_{k\leftarrow\{0,1\}^n}[D^{F_k(\cdot)} = 1] - \mathbf{Pr}_{f\leftarrow\mathcal{F}_n}[D^{f(\cdot)} = 1]| \leq \epsilon(n)$$

By the simulation of $\mathsf{PrivK}^{\mathbf{cpa}}_{A,\widetilde{\Pi}}(n)$ by D^f:

$$\mathbf{Pr}_{f\leftarrow\mathcal{F}_n}[D^{f(\cdot)} = 1] = \mathbf{Pr}[\mathsf{PrivK}^{\mathbf{cpa}}_{A,\widetilde{\Pi}}(n) = 1] = \frac{q(n)}{2^n} + \frac{1}{2}$$

By the simulation of $\mathsf{PrivK}^{\mathbf{cpa}}_{A,\Pi}(n)$ by D^{F_k}:

$$\mathbf{Pr}_{k\leftarrow\{0,1\}^n}[D^{F_k(\cdot)} = 1] = \mathbf{Pr}[\mathsf{PrivK}^{\mathbf{cpa}}_{A,\Pi}(n) = 1]$$

Proof.

Therefore

$$\mathbf{Pr}[\mathsf{PrivK}^{\mathbf{cpa}}_{\boldsymbol{A},\mathbf{\Pi}}(\boldsymbol{n}) = \mathbf{1}] \leq \frac{\mathbf{1}}{\mathbf{2}} + \frac{\boldsymbol{q}(\boldsymbol{n})}{\mathbf{2}^{\boldsymbol{n}}} + \boldsymbol{\epsilon}(\boldsymbol{n})$$
$$= \frac{\mathbf{1}}{\mathbf{2}} + \mathbf{negl}(\boldsymbol{n})$$

$\Longrightarrow$ $\mathbf{\Pi}$ is CPA-secure. □

Real-world Security?

- What happens if a nonce $\boldsymbol{r}$ is ever reused?
- What happens to the bound if the nonce is chosen non-uniformly?

Attacks?

Nonce r not used correctly

- **If r repeats, security fails**
  - Exactly analogous to multiple encryptions using the (pseudo)one-time pad scheme
- When r is a uniform, n-bit string, the probability of a repeat is **negligible**
- **If r is too short, or is chosen from another distribution, repeats may happen**
  - May make scheme insecure

Attacks?

F not used correctly

- (Function of) plaintext directly leaked in ciphertext
  e.g. $\langle m, F_k(m) \rangle$
- F not used with a random, unknown key
  e.g. $\mathsf{Enc}_k(m) = \langle r, F_r(m) \rangle$

CPA-secure Encryption Summary

Practical CPA-secure Scheme

We have shown a CPA-secure encryption scheme based on any PRF:

$$\mathsf{Enc}_k(m) = \langle r, F_k(r) \oplus m \rangle$$

Drawbacks?

- A **1**-block plaintext results in a **2**-block ciphertext
- Only defined for encryption of n-bit messages
- (Both key and message of length n i.e. OTP limitation 1)
- Solution: Modes of Operation (next lecture!)

End

Reference: Section 3.5.2