Symmetric-Key Cryptography
Part 1
Building a “privacy-providing” primitive

“I want my communication with Bob to be private” -- Alice

What kind of “communication”?

Building a “privacy-providing” primitive

What kind of “communication”?

“All of that, and maybe other things, too.”
Building a “privacy-providing” primitive

What kind of “communication”?

“Private” from whom?
A nosey eavesdropper, sniffing wireless packets in a coffee shop?
A business competitor, who pays an ISP to send your traffic for some analysis?
A nation/state agency, with huge computing resources and lots of “side information”?

“From the most powerful attacker you can manage.”
Building a “privacy-providing” primitive

What kind of “communication”?


“Private” from whom?

A nosey eavesdropper, sniffing wireless packets in a coffee shop?
A business competitor, who pays an ISP to send your traffic for some analysis?
A nation/state agency, with huge computing resources and lots of “side information”?

What do you mean by “private”?

No one (other than Bob) can recover the full contents of the communication?
No one can recover more than 1/2 of the contents? (Does it matter which ½?)
No one can determine the “type” of the communication? (e.g. financial data vs. HTML)

“You are annoying!
Just make it work, and
make sure it is fast, too.”
Alice’s Box

arbitrary data

Alice’s Box

“private” data
communication

Bob’s Box

recovered data

API of Alice’s Box

Inputs: 1. bitstrings of any length

Outputs: bitstrings of any length
(but as short as possible to save communication costs)

API of Bob’s Box

Inputs: 1. bitstrings of any length

Outputs: bitstrings of any length
Alice’s Box

arbitrary data

Alice’s Box

“private” data communication

Bob’s Box

recovered data

API of Alice’s Box

Inputs: 1. bitstrings of any length
2. something that the adversary does not know (the “secret”)

Outputs: bitstrings of any length (but as short as possible to save communication costs)

API of Bob’s Box

Inputs: 1. bitstrings of any length
2. something that the adversary does not know (the “secret”)

Outputs: bitstrings of any length
Alice’s Box

arbitrary data

“private” data communication

Bob’s Box

recovered data

API of Alice’s Box

Inputs: 1. bitstrings of any length
2. something that the adversary does not know (the “secret”)

Outputs: bitstrings of any length
(but as short as possible to save communication costs)

API of Bob’s Box

Inputs: 1. bitstrings of any length
2. something that the adversary does not know (the “secret”)

Outputs: bitstrings of any length

Should we assume that the adversary does not know the algorithms inside of Alice and Bob’s boxes? NO.
Alice's Box

arbitrary data

key

Alice's Box

"private" data communication

Bob's Box

key

recovered data

API of Alice's Box

Inputs: 1. bitstrings of any length
2. a (short) secret "key"

Outputs: bitstrings of any length

API of Bob's Box

Inputs: 1. bitstrings of any length
2. a (short) secret "key"

Outputs: bitstrings of any length
Encryption

E

Decryption

D

"private" data communication

API of Encryption

Inputs: 1. bitstrings of any length
   2. a (short) secret "key"

Outputs: bitstrings of any length

API of Decryption

Inputs: 1. bitstrings of any length
   2. a (short) secret "key"

Outputs: bitstrings of any length
An **Encryption Scheme** is a triple of algorithms \( \Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D}) \)

**Key-generation algorithm**
\( \mathcal{K} \) samples from a set of the same name

**Encryption algorithm**
\[ \mathcal{E} : \mathcal{K} \times \{0, 1\}^* \rightarrow \{0, 1\}^* \cup \{\bot\} \]

**Decryption algorithm**
\[ \mathcal{D} : \mathcal{K} \times \{0, 1\}^* \rightarrow \{0, 1\}^* \]
An **Encryption Scheme** is a triple of algorithms $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Key-generation algorithm</strong></td>
<td>$\mathcal{K}$ samples from a set of the same name</td>
<td></td>
</tr>
<tr>
<td><strong>Encryption algorithm</strong></td>
<td>$\mathcal{E} : \mathcal{K} \times {0, 1}^* \rightarrow {0, 1}^* \cup {\perp}$</td>
<td>May be randomized or stateful</td>
</tr>
<tr>
<td></td>
<td>$C \leftarrow \mathcal{E}_K(M)$</td>
<td></td>
</tr>
<tr>
<td><strong>Decryption algorithm</strong></td>
<td>$\mathcal{D} : \mathcal{K} \times {0, 1}^* \rightarrow {0, 1}^*$</td>
<td>Always deterministic</td>
</tr>
<tr>
<td></td>
<td>$M \leftarrow \mathcal{D}_K(C')$</td>
<td></td>
</tr>
</tbody>
</table>
An **Encryption Scheme** is a triple of algorithms \( \Pi = (K, E, D) \)

- **Key-generation algorithm** \( K \) samples from a set of the same name

- **Encryption algorithm** \( E : K \times \{0, 1\}^* \rightarrow \{0, 1\}^* \cup \{\perp\} \)

- **Decryption algorithm** \( D : K \times \{0, 1\}^* \rightarrow \{0, 1\}^* \)

**Correctness condition:**

For all \( K, M \), if \( E_K(M) \neq \perp \), then \( \Pr[D_K(E_K(M)) = M] = 1 \)

over coins of encryption alg.
Developing a notion of “privacy”

1. What might the adversary try to achieve?
   “GOAL”

2. What can the adversary do in its attack?
   “POWER”
Developing a notion of “privacy”

\[ M \in \{0, 1\}^* \rightarrow \mathcal{E}_K \rightarrow C \in \{0, 1\}^* \rightarrow \mathcal{D}_K \rightarrow M \]

1. What might the adversary try to achieve?
   “GOAL”
   - recover the key
   - recover the plaintext
   - determine if this plaintext was sent before
   - determine the most significant bit of
     the plaintext
   - determine if the first and last half of the
     plaintext are the same
   ...

2. What can the adversary do in its attack?
   “POWER”
   - Adversary tries to:
Developing a notion of “privacy”

1. What might the adversary try to achieve?

“GOAL”

Adversary tries to:
- recover the key
- recover the plaintext
- determine if this plaintext was sent before
- determine the most significant bit of the plaintext
- determine if the first and last half of the plaintext are the same
- …

2. What can the adversary do in its attack?

“POWER”

Adversary can:
- observe ciphertexts
- observe plaintexts and ciphertexts
- pick the plaintexts, and then see the corresponding ciphertexts
- adaptively pick the plaintexts, and see the corresponding ciphertexts
- … + “make up” ciphertexts and see their decryptions
Developing a notion of “privacy”

1. What might the adversary try to achieve?
   “GOAL”
   Adversary tries to:
   - recover the key
   - recover the plaintext
   - determine if this plaintext was sent before
   - determine the most significant bit of the plaintext
   - determine if the first and last half of the plaintext are the same
   
   2. What can the adversary do in its attack?
   “POWER”
   Adversary can:
   - observe ciphertexts
   - observe plaintexts and ciphertexts
   - pick the plaintexts, and then see the corresponding ciphertexts
   - adaptively pick the plaintexts, and see the corresponding ciphertexts
   - ... + “make up” ciphertexts and see their decryptions

\[ M \in \{0,1\}^* \xrightarrow{} \mathcal{E}_K \xrightarrow{} C \in \{0,1\}^* \xrightarrow{} \mathcal{D}_K \xrightarrow{} M \]
“Communication is private if…”

Adversary can’t recover the key
“Communication is private if…”

Adversary can’t recover the key

\[ \not \quad E_K(M) = M \]
"Communication is private if…"

Adversary can’t recover the key

\[ \not \quad \mathcal{E}_K(M) = M \]

Adversary can’t recover the plaintext
“Communication is private if…”

Adversary can’t recover the key

\[ \times \quad \mathcal{E}_K(M) = M \]

Adversary can’t recover the plaintext

\[ \times \quad \mathcal{E}_K(M) = M[1..10]||\text{random looking bits} \]
“Communication is private if…”

Adversary can’t recover the key

\[ \times \quad E_K(M) = M \]

Adversary can’t recover the plaintext

\[ \times \quad E_K(M) = M[1..10]\|\text{random looking bits} \]

Any information about the plaintexts that an “efficient” adversary can compute given the ciphertexts, could have been “efficiently” computed without the ciphertexts.
Indistinguishability of ciphertexts under an adaptive chosen-plaintext attack (IND-CPA)

$$\text{Exp}_{\Pi}^{\text{ind-cpa}}(A):$$

\[ K \xleftarrow{\$} \mathcal{K} \]
\[ b \xleftarrow{\$} \{0, 1\} \]
\[ b' \xleftarrow{\$} A^{E_K(LR(\cdot,\cdot,b))} \]

If \( b' = b \) then Return 1
Return 0
Indistinguishability of ciphertexts under an adaptive chosen-plaintext attack (IND-CPA)

\[ \text{Exp}^{\text{ind-cpa}}_{\Pi}(A) : \]
\[ K \xleftarrow{\$} \mathcal{K} \]
\[ b \xleftarrow{\$} \{0, 1\} \]
\[ b' \xleftarrow{\$} A^{E_{K}(LR(\cdot, \cdot, b))} \]
If \( b' = b \) then Return 1
Return 0

\[ \text{Adv}^{\text{ind-cpa}}_{\Pi}(A) = 2 \Pr \left( \text{Exp}^{\text{ind-cpa}}_{\Pi}(A) = 1 \right) - 1 \]

Adversarial “resources”:
- the number of oracle queries, \( q \)
- the total length in bits of the queries, \( \mu \)
- the time-complexity of the adversary, \( t \)
Exploring IND-CPA

$$\text{Exp}_{\Pi}^{\text{ind-cpa}}(A)$$:

1. $$K \leftarrow \mathcal{K}$$
2. $$b \leftarrow \{0, 1\}$$
3. $$b' \leftarrow A^{\mathcal{E}_K}(LR(\cdot, \cdot, b))$$
4. If $$b' = b$$ then Return 1
5. Return 0

$$\text{Adv}_{\Pi}^{\text{ind-cpa}}(A) = 2 \Pr\left(\text{Exp}_{\Pi}^{\text{ind-cpa}}(A) = 1\right) - 1$$

We say $$\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$$ is IND-CPA secure if the IND-CPA advantage is “small” for all “resource efficient” adversaries.

example: adversaries $$A$$ with

$$t = 2^{20}, q = 2^{30}, \mu = 2^{30}$$

achieve advantage at most

$$\text{Adv}_{\Pi}^{\text{ind-cpa}}(A) \leq \frac{1}{2^{40}}$$

But what “small” and “reasonable” mean is up to the user!
Exploring IND-CPA

\[
\text{Exp}_\Pi^{\text{ind-cpa}}(A): \\
K \overset{\$}{\leftarrow} \mathcal{K} \\
b \overset{\$}{\leftarrow} \{0, 1\} \\
b' \overset{\$}{\leftarrow} A^{\mathcal{E}_K(LR(\cdot, \cdot, b))} \\
\text{If } b' = b \text{ then Return } 1 \\
\text{Return } 0
\]

Can this scheme be IND-CPA secure?

\[
\text{Adv}_\Pi^{\text{ind-cpa}}(A) = 2 \Pr \left( \text{Exp}_\Pi^{\text{ind-cpa}}(A) = 1 \right) - 1
\]

\[
\mathcal{E}_K(M) = M
\]
Exploring IND-CPA

\[
\text{Adv}^{\text{ind-cpa}}_\Pi (A) = 2 \Pr \left( \text{Exp}^{\text{ind-cpa}}_\Pi (A) = 1 \right) - 1
\]

\[
\text{Exp}^{\text{ind-cpa}}_\Pi (A):
\]
\[
K \leftarrow \mathcal{K}
\]
\[
b \leftarrow \{0, 1\}
\]
\[
b' \leftarrow A^\mathcal{E}_K (LR(\cdot, \cdot, b))
\]
If \( b' = b \) then Return 1
Return 0

Can this scheme be IND-CPA secure? \( \mathcal{E}_K (M) = M \)

Adversary A:

- fix distinct strings \( M_0, M_1 \) of the same length
- ask query \((M_0, M_1)\)
- if oracle response \( C = M_0 \) then return 0
- else return 1
Exploring IND-CPA

\[
\text{Exp}^{\text{ind-cpa}}_{\Pi}(A): \\
K \overset{\$}{\leftarrow} \mathcal{K} \\
b \overset{\$}{\leftarrow} \{0, 1\} \\
b' \overset{\$}{\leftarrow} A^{\mathcal{E}_K}(LR(\cdots,b)) \\
\text{If } b' = b \text{ then Return } 1 \\
\text{Return } 0
\]

\[
\text{Adv}^{\text{ind-cpa}}_{\Pi}(A) = 2 \Pr \left( \text{Exp}^{\text{ind-cpa}}_{\Pi}(A) = 1 \right) - 1
\]

Can any deterministic scheme be IND-CPA secure? [EXERCISE!]

If you think “yes”, then you must give a deterministic scheme and prove it is IND-CPA secure

If you think “no”, then you must show that, for any deterministic scheme, there exists an efficient attack that gains large IND-CPA advantage
Exploring IND-CPA

\[
\operatorname{Exp}^{\text{ind-cpa}}_\Pi(A): \quad \operatorname{Adv}^{\text{ind-cpa}}_\Pi(A) = 2 \Pr \left( \operatorname{Exp}^{\text{ind-cpa}}_\Pi(A) = 1 \right) - 1
\]

\[
K \xleftarrow{\$} \mathcal{K} \\
b \xleftarrow{\$} \{0, 1\} \\
b' \xleftarrow{\$} A^E_K(LR(\cdot, \cdot, b)) \\
\text{If } b' = b \text{ then Return } 1 \\
\text{Return } 0
\]

Can any deterministic scheme be IND-CPA secure? [NO]

Adversary A:

- fix distinct strings \( M_0, M_1 \) of the same length
- ask query \((M_0, M_1)\), receiving \( C_1 \) in return
- ask query \((M_0, M_0)\), receiving \( C_2 \) in return
- if \( C_1 = C_2 \) then return 0
- else return 1
An alternative definition of privacy: “Real or Random” (RoR-CPA)

\[
\begin{align*}
\text{Exp}_{\Pi}^{\text{ror- CPA}}(A): & \\
K & \leftarrow \mathcal{K} \\
b & \leftarrow \{0, 1\} \\
b' & \leftarrow A^{\mathcal{O}(\cdot)} \\
\text{If } b' = b & \text{ then Return 1} \\
\text{Return 0} \\
\end{align*}
\]

\[
\begin{align*}
\text{Oracle } \mathcal{O}(M): & \\
M' & \leftarrow \{0, 1\}^{\lfloor M \rfloor} \\
\text{If } b = 0 & \text{ then Return } \mathcal{E}_K(M') \\
\text{Return } \mathcal{E}_K(M) \\
\end{align*}
\]

\[
\text{Adv}_{\Pi}^{\text{ror- CPA}}(A) = 2 \Pr \left( \text{Exp}_{\Pi}^{\text{ror- CPA}}(A) = 1 \right) - 1
\]

Adversarial “resources”:
- the number of oracle queries, \( q \)
- the total length in bits of the queries, \( \mu \)
- the time-complexity of the adversary, \( t \)
Which notion is “better”: RoR-CPA or IND-CPA?

\[
\text{Exp}_{\Pi}^{\text{ror-\text{cpa}}} (A): \\
K \leftarrow \mathcal{K} \\
b \leftarrow \{0, 1\} \\
b' \leftarrow A^{\mathcal{O}(\cdot)} \\
\text{If } b' = b \text{ then Return } 1 \\
\text{Return } 0 \\
\]

\[
\text{Oracle } \mathcal{O}(M): \\
M' \leftarrow \{0, 1\}^{|M|} \\
\text{If } b = 0 \text{ then Return } \mathcal{E}_K(M') \\
\text{Return } \mathcal{E}_K(M) \\
\]

\[
\text{Exp}_{\Pi}^{\text{ind-\text{cpa}}} (A): \\
K \leftarrow \mathcal{K} \\
b \leftarrow \{0, 1\} \\
b' \leftarrow A^{\mathcal{O}(\cdot)} \\
\text{If } b' = b \text{ then Return } 1 \\
\text{Return } 0 \\
\]

\[
\text{Oracle } \mathcal{O}(M_0, M_1): \\
\text{If } b = 0 \text{ then Return } \mathcal{E}_K(M_0) \\
\text{Return } \mathcal{E}_K(M_1) \\
\]
Claim: Any encryption scheme $\Pi = (K, E, D)$ that is IND-CPA secure, is also RoR-CPA secure.
Claim: Any encryption scheme $\Pi = (K, E, D)$ that is IND-CPA secure, is also RoR-CPA secure

Proof idea: show the contrapositive, if a scheme $\Pi = (K, E, D)$ is not RoR-CPA secure, then it is not IND-CPA secure.
Claim: Any encryption scheme $\Pi = (K, E, D)$ that is IND-CPA secure, is also RoR-CPA secure

Proof idea: show the contrapositive, if a scheme $\Pi = (K, E, D)$ is not RoR-CPA secure, then it is not IND-CPA secure.

Let $A$ be an efficient RoR-CPA adversary, gaining advantage $\text{Adv}_{\Pi}^{\text{ror-cca}}(A)$

We build an efficient IND-CPA adversary $B$, that runs $A$ as a "black-box" subroutine, that gains advantage

$$\text{Adv}_{\Pi}^{\text{ind-cca}}(B) \geq \text{Adv}_{\Pi}^{\text{ror-cca}}(A)$$
Claim: Any encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ that is IND-CPA secure, is also RoR-CPA secure.

Proof idea: show the contrapositive, if a scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is not RoR-CPA secure, then it is not IND-CPA secure.

Let $A$ be an efficient RoR-CPA adversary, gaining advantage $\text{Adv}_\Pi^{\text{ror-cpa}}(A)$.

We build an efficient IND-CPA adversary $B$, that runs $A$ as a “black-box” subroutine, that gains advantage

$$\text{Adv}_\Pi^{\text{ind-cpa}}(B) \geq \text{Adv}_\Pi^{\text{ror-cpa}}(A)$$

Conclusion: if $\text{Adv}_\Pi^{\text{ind-cpa}}(B)$ is small for all efficient $B$, then $\text{Adv}_\Pi^{\text{ror-cpa}}(A)$ must be small, too.
So, we start with an RoR-adversary $A$ that gains some RoR advantage.

\[
\frac{1}{2} \text{Adv}_\Pi^{\text{ror-\text{cpa}}}(A) + \frac{1}{2} \leq \Pr(\text{Exp}_\Pi^{\text{ror-\text{cpa}}}(A) = 1)
\]
\[
\frac{1}{2} \text{Adv}_{\Pi}^{\text{ror-cca}}(A) + \frac{1}{2} \leq \Pr(\text{Exp}_{\Pi}^{\text{ror-cca}}(A) = 1)
\]

**Exp}_{\Pi}^{\text{ror-cca}}(A):**
- \(K \leftarrow \mathcal{K}\)
- \(d \leftarrow \{0, 1\}\)
- \(d' \leftarrow A^{\mathcal{O}(\cdot)}\)
- If \(d' = d\) then Return 1
- Return 0

**Oracle \(\mathcal{O}(M):\)**
- \(M' \leftarrow \{0, 1\}^{\lfloor |M| \rfloor}\)
- If \(d = 0\) then Return \(E_K(M')\)
- Return \(E_K(M)\)
Want to build a good IND-CPA adversary $B$ by running $A$ and simulating its expected experiment.
\[ \frac{1}{2} \text{Adv}_{\Pi}^{\text{ror-cpa}}(A) + \frac{1}{2} \leq \Pr(\text{Exp}_{\Pi}^{\text{ror-cpa}}(A) = 1) \]

\[ \mathcal{E}_K(LR(\cdot, \cdot, b)) \]

**Exp}_{\Pi}^{\text{ror-cpa}}(A):**

- \( K \leftarrow \mathcal{K} \)
- \( d \leftarrow \{0, 1\} \)
- \( d' \leftarrow A^{\mathcal{O}(\cdot)} \)
- If \( d' = d \) then Return 1
- Return 0

**Oracle \( \mathcal{O}(M) \):**

- \( M' \leftarrow \{0, 1\}^{\lceil |M| \rceil} \)
- If \( d = 0 \) then Return \( \mathcal{E}_K(M') \)
- Return \( \mathcal{E}_K(M) \)
\[ \frac{1}{2} \text{Adv}^{\text{ror-cpa}}_{\Pi}(A) + \frac{1}{2} \leq \Pr(\text{Exp}^{\text{ror-cpa}}_{\Pi}(A) = 1) \]

\[ \mathcal{E}_K(LR(\cdot, \cdot, b)) \]

\text{Exp}^{\text{ror-cpa}}_{\Pi}(A):

\begin{align*}
K & \leftarrow \mathcal{K} \\
d & \leftarrow \{0, 1\} \\
d' & \leftarrow A^{\mathcal{O}(\cdot)} \\
M' & \leftarrow \{0, 1\}^{M} \\
A & \leftarrow \{0, 1\}^{M} \\
B & \leftarrow \{0, 1\}^{M} \\
\end{align*}

Oracle \(\mathcal{O}(M)\):

\begin{align*}
M' & \leftarrow \{0, 1\}^{M} \\
\text{If } d = 0 \text{ then Return } \mathcal{E}_K(M') \\
\text{Return } \mathcal{E}_K(M) \\
\end{align*}
\[
\frac{1}{2} \text{Adv}_{\Pi}^{\text{ror-cpa}}(A) + \frac{1}{2} \leq \Pr(\text{Exp}_{\Pi}^{\text{ror-cpa}}(A) = 1)
\]

\[\mathcal{E}_K(LR(\cdot, \cdot, b))\]

\[(M_0, M_1)\]

\[M_0 \leftarrow M'\]
\[M_1 \leftarrow M\]
\[M' \leftarrow \{0, 1\}^{\lceil |M| \rceil}\]

\[\text{Exp}_{\Pi}^{\text{ror-cpa}}(A):\]

\[
\begin{align*}
K & \leftarrow \mathcal{K} \\
\text{d} & \leftarrow \{0, 1\} \\
\text{d}' & \leftarrow A^{\mathcal{O}(\cdot)} \\
\text{If d'} = \text{d} & \text{ then Return 1} \\
\text{Return 0}
\end{align*}
\]

\[\text{Oracle } \mathcal{O}(M):\]

\[
\begin{align*}
M' & \leftarrow \{0, 1\}^{\lceil |M| \rceil} \\
\text{If d} = 0 & \text{ then Return } \mathcal{E}_K(M') \\
\text{Return } \mathcal{E}_K(M)
\end{align*}
\]
\[
\frac{1}{2} \text{Adv}^\text{ror-cca}_{\Pi}(A) + \frac{1}{2} \leq \Pr(\text{Exp}^\text{ror-cca}_{\Pi}(A) = 1)
\]

\[\mathcal{E}_K(LR(\cdot, \cdot, b))\]

\[
(M_0, M_1) \xrightarrow{C} (M_0', M_1) \\
M_0 \leftarrow M' \\
M_1 \leftarrow M \\
M' \leftarrow \{0, 1\}^{|M|} \\
A \leftarrow \{0, 1\}^{|M|} \\
A' \leftarrow A^{\mathcal{O}(\cdot)} \\
\text{Exp}^\text{ror-cca}_{\Pi}(A): \\
K \leftarrow \mathcal{K} \\
d \leftarrow \{0, 1\} \\
d' \leftarrow A^{\mathcal{O}(\cdot)} \\
\text{If } d' = d \text{ then Return 1} \\
\text{Return 0}
\]

\[
\text{Oracle } \mathcal{O}(M): \\
M' \leftarrow \{0, 1\}^{|M|} \\
\text{If } d = 0 \text{ then Return } \mathcal{E}_K(M') \\
\text{Return } \mathcal{E}_K(M)
\]
\[
\frac{1}{2} \text{Adv}_{\Pi}^{\text{ror-cpa}}(A) + \frac{1}{2} \leq \Pr(\text{Exp}_{\Pi}^{\text{ror-cpa}}(A) = 1)
\]

\[\mathcal{E}_K(LR(\cdot, \cdot, b))\]

\textbf{Exp}_{\Pi}^{\text{ror-cpa}}(A):

\begin{align*}
K & \leftarrow \mathcal{K} \\
(d, d') & \leftarrow \{0, 1\} \\
d & \leftarrow \{0, 1\} \\
d' & \leftarrow A(\cdot) \\
\end{align*}

If \(d' = d\) then Return 1
Return 0

\textbf{Oracle }\mathcal{O}(M):

\begin{align*}
M' & \leftarrow \{0, 1\}^{\left|\mathcal{M}\right|} \\
\end{align*}

If \(d = 0\) then Return \(\mathcal{E}_K(M')\)
Return \(\mathcal{E}_K(M)\)
Thus, $B$ perfectly simulates $A$'s expected experiment and will "win" whenever $A$ wins.
\[
\frac{1}{2} \text{Adv}_{\Pi}^{\text{ror-cca}}(A) + \frac{1}{2} \leq \Pr(\text{Exp}_{\Pi}^{\text{ror-cca}}(A) = 1) \\
\leq \Pr(\text{Exp}_{\Pi}^{\text{ind-cca}}(B) = 1) \\
\text{Adv}_{\Pi}^{\text{ror-cca}}(A) \leq 2 \Pr(\text{Exp}_{\Pi}^{\text{ind-cca}}(B) = 1) - 1
\]

And hence,

\[
\text{Adv}_{\Pi}^{\text{ror-cca}}(A) \leq \text{Adv}_{\Pi}^{\text{ind-cca}}(B)
\]

as we claimed.
So we say “IND-CPA security implies RoR-CPA security”

IND-CPA $\Rightarrow$ RoR-CPA

What about the other way around?
Claim: Any encryption scheme $\Pi = (K, E, D)$ that is RoR-CPA secure, is also IND-CPA secure

Proof idea: show the contrapositive, if a scheme $\Pi = (K, E, D)$ is not IND-CPA secure, then it is not RoR-CPA secure.
Claim: Any encryption scheme \( \Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D}) \) that is RoR-CPA secure, is also IND-CPA secure.

Proof idea: show the contrapositive, if a scheme \( \Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D}) \) is not IND-CPA secure, then it is not RoR-CPA secure.

Let \( A \) be an efficient IND-CPA adversary, gaining advantage \( \text{Adv}_{\Pi}^{\text{ind-CPA}}(A) \).

We build an efficient RoR-CPA adversary \( B \), that runs \( A \) as a "black-box" subroutine, that gains advantage

\[
2\text{Adv}_{\Pi}^{\text{ror-CPA}}(B) \geq \text{Adv}_{\Pi}^{\text{ind-CPA}}(A)
\]

Conclusion: if \( \text{Adv}_{\Pi}^{\text{ror-CPA}}(B) \) is small for all efficient \( B \), then \( \text{Adv}_{\Pi}^{\text{ind-CPA}}(A) \) must be small, too.
Claim: Any encryption scheme \( \Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D}) \) that is RoR-CPA secure, is also IND-CPA secure.

Proof idea: show the contrapositive, if a scheme \( \Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D}) \) is not IND-CPA secure, then it is not RoR-CPA secure.

Let \( A \) be an efficient IND-CPA adversary, gaining advantage \( \text{Adv}^\text{ind-cpa}_\Pi (A) \).

We build an efficient RoR-CPA adversary \( B \), that runs \( A \) as a "black-box" subroutine, that gains advantage

\[
2\text{Adv}^\text{ror-cpa}_\Pi (B) \geq \text{Adv}^\text{ind-cpa}_\Pi (A)
\]

Conclusion: if \( \text{Adv}^\text{ror-cpa}_\Pi (B) \) is small for all efficient \( B \), then \( \text{Adv}^\text{ind-cpa}_\Pi (A) \) must be small, too.

AT HOME EXERCISE: try to write the proof.
So we say “IND-CPA security implies RoR-CPA security”

\[ \text{IND-CPA} \Rightarrow \text{RoR-CPA} \]

And “RoR-CPA security implies IND-CPA security”, too

\[ \text{RoR-CPA} \Rightarrow \text{IND-CPA} \]

(Although the two directions are not equally “tight”)

There are a variety of definitions of IND-CPA that are all qualitatively equivalent:

- Left-or-Right IND-CPA
- Real-or-Random IND-CPA
- Real-or-0s IND-CPA
- Find-then-Guess IND-CPA
- Semantic security

Although not all of the reductions have the same quantitative “tightness”

Check out [Bellare, Desai, Pointcheval, Rogaway]
So, now we have

-- a precise syntax for the object we want to build
-- a precise target security notion, left-or-right IND-CPA

How should we build this thing?
“Perfect” encryption

There does exist one “perfect” symmetric encryption scheme: **One Time Pad**

![Diagram](https://via.placeholder.com/150)

random bits (key) ⊕

plaintext message

random ciphertext bits (independent of message)

Sadly, requires a stream of **secret** random bits as long as the length of all messages you want to send.
Approximating One-Time Pad

Perhaps we can turn a short secret key into computationally indistinguishable from random bits.

plaintext message →

computationally indistinguishable from random bits
Intuitively, making small blocks of “random-looking” bits should be easier (at least, not harder) than making a long string all at once.

So we need a function that outputs small blocks of “random looking” bits.
Consider the set \( \text{Func}(n, n) = \{ f : \{0, 1\}^n \to \{0, 1\}^n \} \), the “family” of all functions mapping n-bit strings to n-bit strings.
Consider the set $\text{Func}(n, n) = \{ f : \{0, 1\}^n \rightarrow \{0, 1\}^n \}$, the “family” of all functions mapping n-bit strings to n-bit strings.

Two equivalent viewpoints on picking a “random function”

1. Sampling an element of $\text{Func}(n, n)$

- everything-to-zero map
- identity map

$f$
Consider the set \( \text{Func}(n, n) = \{ f : \{0, 1\}^n \to \{0, 1\}^n \} \), the "family" of all functions mapping \( n \)-bit strings to \( n \)-bit strings.

Two equivalent viewpoints on picking a "random function"

1. Sampling an element of \( \text{Func}(n, n) \)

   It’s not hard to see that

   \[
   \forall X, Y \in \{0, 1\}^n, \Pr(f(X) = Y) = \frac{(2^n)^{2^{n-1}}}{(2^n)^{2^n}} = 1/2^n
   \]
Consider the set \( \text{Func}(n, n) = \{ f: \{0, 1\}^n \rightarrow \{0, 1\}^n \} \), the “family” of all functions mapping \( n \)-bit strings to \( n \)-bit strings.

Two equivalent viewpoints on picking a “random function”

1. Sampling an element of \( \text{Func}(n, n) \)
   - everything-to-zero map
   - identity map

2. Fill in the function table “lazily”

\[
\begin{array}{l|l}
00\ldots00 & 11010110\ldots110101 \\
00\ldots01 & 10000010\ldots100111 \\
00\ldots10 & 00000010\ldots011111 \\
11\ldots10 & 101111111\ldots100111 \\
11\ldots11 & 010101110\ldots100111 \\
\end{array}
\]
Imagine we could sample $f \leftarrow \text{Func}(n, n)$ and then encrypt via…

\[
\begin{array}{cccccc}
  f(0) & f(1) & f(2) & \ldots & \ldots & f(\ell) \\
  \text{yellow} & \text{yellow} & \text{yellow} & \text{yellow} & \text{yellow} & \text{yellow} \\
\end{array}
\]

… we get one-time pad! But there’s still a catch.

(What is the size of the key for this encryption scheme?)
Imagine we could sample $f \leftarrow \text{Func}(n, n)$ and then encrypt via…

... we get one-time pad! But there's still a catch.

$$\log_2 \left( (2^n)^{2^n} \right) = n2^n \text{ bits of key}$$
Pseudorandom Functions (PRFs)

Let \( F : \mathcal{K} \times \{0, 1\}^n \rightarrow \{0, 1\}^n \) be viewed as a “keyed” function family.

\[
\begin{align*}
\text{Exp}^{\text{prf}}_F (A): & \quad \text{Oracle } \mathcal{O}(X): \\
K & \overset{\$}{\leftarrow} \mathcal{K} \\
f & \overset{\$}{\leftarrow} \text{Func}(n, n) \\
b & \overset{\$}{\leftarrow} \{0, 1\} \\
b' & \overset{\$}{\leftarrow} A^{\mathcal{O}(\cdot)} \\
\text{If } b' = b \text{ then Return 1} \\
\text{Return 0}
\end{align*}
\]

\[
\text{Adv}^{\text{prf}}_F (A) = 2 \Pr(\text{Exp}^{\text{prf}}_F (A) = 1) - 1
\]

“My oracle is…”
Counter-mode encryption over a function family $F$ (CTR[$F$])

Initialization: $K \leftarrow \mathcal{K}; \ ctr \leftarrow 0$

For the next message, update the encryption scheme state: $\ctr \leftarrow \ctr + b$

means: add 2 to $\ctr$, encode as an $n$-bit string
Claim: If $F : \mathcal{K} \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ is a secure PRF, then $\text{CTR}[F] = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ (counter-mode over $F$) is IND-CPA secure.
Claim: If $F: \mathcal{K} \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ is a secure PRF, then $\text{CTR}[F] = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ (counter-mode over F) is IND-CPA secure.

Proof idea: break the proof into two steps

1. replace $F_K$ with a random function $f$, and argue that any adversary that can detect this can “break” PRF-security of F
Proof idea: break the proof into two steps

1. replace $F_K$ with a random function $f$, and argue that any adversary that can detect this can "break" PRF-security of $F$
Claim: If $F: \mathcal{K} \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ is a secure PRF, then $CTR[F] = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ (counter-mode over $F$) is IND-CPA secure.

Proof idea: break the proof into two steps

1. replace $F_K$ with a random function $f$, and argue that any adversary that can detect this can “break” PRF-security of $F$

2. analyze IND-CPA security of $CTR[Func(n, n)]$

\[
Adv_{CTR[F]}^{\text{ind-cpa}}(A) \leq Adv_{CTR[Func(n,n)]}^{\text{ind-cpa}}(A) + Adv_F^{\text{prf}}(B)
\]
\[ \text{Exp}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) : \]
\[
K \leftarrow \mathcal{K} \\
d \leftarrow \{0, 1\} \\
d' \leftarrow A^{O(\cdot, \cdot)} \\
\text{If } d' = d \text{ then Return } 1 \\
\text{Return } 0 \\
\]

Oracle \( O(M_0, M_1) : \)
\[
\text{If } d = 0 \text{ then Return } \mathcal{E}_K(M_0) \\
\text{Return } \mathcal{E}_K(M_1) \\
\]

\[
\text{Adv}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) = 2 \Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) = 1) - 1
\]

\[ \text{Exp}_{F}^{\text{prf}}(B) : \]
\[
K \leftarrow \mathcal{K} \\
f \leftarrow \text{Func}(n, n) \\
b \leftarrow \{0, 1\} \\
b' \leftarrow B^{O(\cdot)} \\
\text{If } b' = b \text{ then Return } 1 \\
\text{Return } 0 \\
\]

Oracle \( O(X) : \)
\[
\text{If } b = 0 \text{ then Return } f(X) \\
\text{Return } F_K(X) \\
\]

\[
\text{Adv}_{F}^{\text{prf}}(B) = 2 \Pr(\text{Exp}_{F}^{\text{prf}}(B) = 1) - 1
\]

\[
\text{Adv}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) \leq \text{Adv}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cpa}}(A) + \text{Adv}_{F}^{\text{prf}}(B)
\]
\[
\frac{1}{2} \text{Adv}^{\text{ind-cpa}}_{\text{CTR}[F]}(A) + \frac{1}{2} = \Pr(\text{Exp}^{\text{ind-cpa}}_{\text{CTR}[F]}(A) = 1)
\]

So, we start with an IND-CPA adversary that gains some IND-CPA advantage in attacking CTR[F].
\[ \frac{1}{2} \text{Adv}_{\text{CTR}[F]}^{\text{ind-cca}}(A) + \frac{1}{2} = \Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind-cca}}(A) = 1) \]
\[ = \Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind-cca}}(A) = 1) - \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cca}}(A) = 1) \]
\[ + \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cca}}(A) = 1) \]

Now we add a "useful" version of 0 to the right side.
\[ \frac{1}{2} \text{Adv}^\text{ind-cca}_{\text{CTR}[F]}(A) + \frac{1}{2} = \Pr(\text{Exp}^\text{ind-cca}_{\text{CTR}[F]}(A) = 1) \]
\[ = \Pr(\text{Exp}^\text{ind-cca}_{\text{CTR}[F]}(A) = 1) - \Pr(\text{Exp}^\text{ind-cca}_{\text{CTR}[\text{Func}(n,n)]}(A) = 1) \]
\[ + \Pr(\text{Exp}^\text{ind-cca}_{\text{CTR}[\text{Func}(n,n)]}(A) = 1) \]
\[
\frac{1}{2} \text{Adv}^{\text{ind-cca}}_{\text{CTR}[F]}(A) + \frac{1}{2} = \Pr(\text{Exp}^{\text{ind-cca}}_{\text{CTR}[F]}(A) = 1) \\
= \Pr(\text{Exp}^{\text{ind-cca}}_{\text{CTR}[F]}(A) = 1) - \Pr(\text{Exp}^{\text{ind-cca}}_{\text{CTR}[\text{Func}(n,n)]}(A) = 1) \\
+ \Pr(\text{Exp}^{\text{ind-cca}}_{\text{CTR}[\text{Func}(n,n)]}(A) = 1)
\]

I claim that:
\[
\Pr(\text{Exp}^{\text{ind-cca}}_{\text{CTR}[F]}(A) = 1) - \Pr(\text{Exp}^{\text{ind-cca}}_{\text{CTR}[\text{Func}(n,n)]}(A) = 1) = \text{Adv}^{\text{prf}}_{F}(B)
\]
I claim that:

\[
\frac{1}{2} \text{Adv}_{\text{CTR}[F]}^{\text{ind}-\text{cpa}}(A) + \frac{1}{2} = \Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind}-\text{cpa}}(A) = 1) \\
= \Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind}-\text{cpa}}(A) = 1) - \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind}-\text{cpa}}(A) = 1) \\
+ \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind}-\text{cpa}}(A) = 1)
\]

I claim that:

\[
\Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind}-\text{cpa}}(A) = 1) - \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind}-\text{cpa}}(A) = 1) = \text{Adv}_{F}^{\text{prf}}(B)
\]

Adversary \( B^g(\cdot) \):

\[
d \leftarrow \{0, 1\}
\]
Run \( A \)
When \( A \) asks \((M_0, M_1)\) to its oracle:

Simulate encryption of \( M_d \) using calls to oracle \( g \)

Respond with resulting ctxt \( C \)

When \( A \) halts with output bit \( d' \):

If \( d' = d \) Then Return 1
Else Return 0
\[
\frac{1}{2} \text{Adv}^\text{ind-cca}_{\text{CTR}[F]}(A) + \frac{1}{2} = \Pr(\text{Exp}^\text{ind-cca}_{\text{CTR}[F]}(A) = 1)
\]
\[
= \Pr(\text{Exp}^\text{ind-cca}_{\text{CTR}[F]}(A) = 1) - \Pr(\text{Exp}^\text{ind-cca}_{\text{CTR}[\text{Func}(n,n)]}(A) = 1)
\]
\[
+ \Pr(\text{Exp}^\text{ind-cca}_{\text{CTR}[\text{Func}(n,n)]}(A) = 1)
\]

I claim that:
\[
\Pr(\text{Exp}^\text{ind-cca}_{\text{CTR}[F]}(A) = 1) - \Pr(\text{Exp}^\text{ind-cca}_{\text{CTR}[\text{Func}(n,n)]}(A) = 1) = \text{Adv}^\text{prf}_F(B)
\]

If PRF bit b=1:
B simulates IND-CPA experiment for \text{CTR}[F],
And outputs 1
if A guesses the bit
I claim that:

$$\frac{1}{2} \text{Adv}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) + \frac{1}{2} = \Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) = 1)$$

$$= \Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) = 1) - \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cpa}}(A) = 1)$$

$$+ \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]]}^{\text{ind-cpa}}(A) = 1)$$

I claim that:

$$\Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) = 1) - \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]]}^{\text{ind-cpa}}(A) = 1) = \text{Adv}_{F}^{\text{prf}}(B)$$

If PRF bit \(b=1\):

- \(B\) simulates IND-CPA experiment for \(\text{CTR}[F]\),
- And outputs 1 if \(A\) guesses the bit

If PRF bit \(b=0\):

- \(B\) simulates IND-CPA experiment for \(\text{CTR}[\text{Func}(n,n)]\),
- And outputs 1 if \(A\) guesses the bit

**Adversary \(B^g(\cdot)\):**

- \(d \leftarrow \{0, 1\}\)
- Run \(A\)
- When \(A\) asks \((M_0, M_1)\) to its oracle:
  - Simulate encryption of \(M_d\) using calls to oracle \(g\)
  - Respond with resulting ctxt \(C\)
- When \(A\) halts with output bit \(d'\):
  - If \(d' = d\) Then Return 1
  - Else Return 0
I claim that:

\[
\frac{1}{2} \text{Adv}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) + \frac{1}{2} = \Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) = 1)
\]

\[
= \Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) = 1) - \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cpa}}(A) = 1)
\]

\[
+ \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cpa}}(A) = 1)
\]

If PRF bit \( b = 1 \):

B simulates IND-CPA experiment for \( CTR[F] \),
And outputs 1
if A guesses the bit

\[
\Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) = 1) = \Pr(\text{Exp}_{\text{F}}^{\text{prf}}(B) = 1 | b = 1)
\]

\[
\Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cpa}}(A) = 1) = \Pr(\text{Exp}_{\text{F}}^{\text{prf}}(B) = 0 | b = 0)
\]

\[
= 1 - \Pr(\text{Exp}_{\text{F}}^{\text{prf}}(B) = 1 | b = 0)
\]

If PRF bit \( b = 0 \):

B simulates IND-CPA experiment for \( CTR[\text{Func}(n,n)] \),
And outputs 1
if A guesses the bit
\[
\frac{1}{2} \text{Adv}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) + \frac{1}{2} = \Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) = 1) \\
= \Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) = 1) - \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cpa}}(A) = 1) \\
+ \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cpa}}(A) = 1)
\]

I claim that:
\[
\Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) = 1) - \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cpa}}(A) = 1) = \text{Adv}_{F}^{\text{prf}}(B)
\]

So by subtracting probabilities on the left side:
\[
\Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) = 1) = \Pr(\text{Exp}_{F}^{\text{prf}}(B) = 1 \mid b = 1) \\
\Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cpa}}(A) = 1) = \Pr(\text{Exp}_{F}^{\text{prf}}(B) = 0 \mid b = 0) \\
= 1 - \Pr(\text{Exp}_{F}^{\text{prf}}(B) = 1 \mid b = 0)
\]

\[
\frac{1}{2} \text{Adv}_{F}^{\text{prf}}(B) = 2 \Pr(\text{Exp}_{F}^{\text{prf}}(B) = 1) - 1
\]
\[
\frac{1}{2} \text{Adv}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) + \frac{1}{2} = \Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) = 1) \\
= \Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind-cpa}}(A) = 1) - \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cpa}}(A) = 1) \\
\quad + \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cpa}}(A) = 1) \\
= \text{Adv}_{\text{F}}^{\text{prf}}(B) + \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cpa}}(A) = 1)
\]
\[
\frac{1}{2} \text{Adv}^\text{ind-cca}_{\text{CTR}[F]}(A) + \frac{1}{2} = \Pr(\text{Exp}^\text{ind-cca}_{\text{CTR}[F]}(A) = 1)
\]
\[
= \Pr(\text{Exp}^\text{ind-cca}_{\text{CTR}[F]}(A) = 1) - \Pr(\text{Exp}^\text{ind-cca}_{\text{CTR}[\text{Func}(n,n)]}(A) = 1) + \Pr(\text{Exp}^\text{ind-cca}_{\text{CTR}[\text{Func}(n,n)]}(A) = 1)
\]
\[
= \text{Adv}^\text{prf}_F(B) + \Pr(\text{Exp}^\text{ind-cca}_{\text{CTR}[\text{Func}(n,n)]}(A) = 1)
\]
\[
\text{Adv}^\text{ind-cca}_{\text{CTR}[F]}(A) = 2\text{Adv}^\text{prf}_F(B) + 2\Pr(\text{Exp}^\text{ind-cca}_{\text{CTR}[\text{Func}(n,n)]}(A) = 1) - 1
\]
\[
= 2\text{Adv}^\text{prf}_F(B) + \text{Adv}^\text{ind-cca}_{\text{CTR}[\text{Func}(n,n)]}(A)
\]
\[
\frac{1}{2} \text{Adv}_{\text{CTR}[F]}^{\text{ind-cca}}(A) + \frac{1}{2} = \Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind-cca}}(A) = 1) \\
= \Pr(\text{Exp}_{\text{CTR}[F]}^{\text{ind-cca}}(A) = 1) - \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cca}}(A) = 1) \\
+ \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cca}}(A) = 1) \\
= \text{Adv}_{F}^{\text{prf}}(B) + \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cca}}(A) = 1) \\
\text{Adv}_{\text{CTR}[F]}^{\text{ind-cca}}(A) = 2\text{Adv}_{F}^{\text{prf}}(B) + 2\Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cca}}(A) = 1) - 1 \\
= 2\text{Adv}_{F}^{\text{prf}}(B) + \text{Adv}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cca}}(A)
\]

I claim: \( \Pr(\text{Exp}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cca}}(A) = 1) \leq \frac{1}{2} \) \( \Rightarrow \text{Adv}_{\text{CTR}[\text{Func}(n,n)]}^{\text{ind-cca}}(A) \leq 0 \)

Proof sketch: all ciphertexts are independent of the IND-CPA experiment bit! 
So probability of guessing the bit is at most 1/2
\[ \frac{1}{2} \text{Adv}^{\text{ind-cca}}_{\text{CTR}[F]}(A) + \frac{1}{2} = \Pr(\text{Exp}^{\text{ind-cca}}_{\text{CTR}[F]}(A) = 1) \]
\[ = \Pr(\text{Exp}^{\text{ind-cca}}_{\text{CTR}[F]}(A) = 1) - \Pr(\text{Exp}^{\text{ind-cca}}_{\text{CTR}[\text{Func}(n,n)]}(A) = 1) \]
\[ + \Pr(\text{Exp}^{\text{ind-cca}}_{\text{CTR}[\text{Func}(n,n)]}(A) = 1) \]
\[ = \text{Adv}^{\text{prf}}_F(B) + \Pr(\text{Exp}^{\text{ind-cca}}_{\text{CTR}[\text{Func}(n,n)]}(A) = 1) \]
\[ \text{Adv}^{\text{ind-cca}}_{\text{CTR}[F]}(A) = 2\text{Adv}^{\text{prf}}_F(B) + 2\Pr(\text{Exp}^{\text{ind-cca}}_{\text{CTR}[\text{Func}(n,n)]}(A) = 1) - 1 \]
\[ = 2\text{Adv}^{\text{prf}}_F(B) + \text{Adv}^{\text{ind-cca}}_{\text{CTR}[\text{Func}(n,n)]}(A) \]
\[ \text{Adv}^{\text{ind-cca}}_{\text{CTR}[F]}(A) \leq 2\text{Adv}^{\text{prf}}_F(B) \]

And we’re done.
I want to implement this. What should I use for $F_k$?

Recall... $F : K \times \{0, 1\}^n \rightarrow \{0, 1\}^n$

So any keyed, $n$-bit to $n$-bit function will do...
Use a blockcipher, like AES.

Recall: $\text{AES}: \{0, 1\}^{128} \times \{0, 1\}^{128} \rightarrow \{0, 1\}^{128}$
Wait... blockciphers are not just function families, they are permutation families.

How does $\text{Adv}_{\text{CTR}[F]}^{\text{ind-cca}}(A)$ relate to $\text{Adv}_{\text{CTR}[AES]}^{\text{ind-cca}}(A)$?
Consider the set $\text{Perm}(n) = \{\pi: \{0, 1\}^n \rightarrow \{0, 1\}^n\}$
the “family” of all permutations over n-bit strings

Two equivalent viewpoints on picking a “random permutation”

1. Sampling an element of $\text{Perm}(n)$

2. fill in the permutation table “lazily”
Pseudorandom Permutations (PRPs)

Let $E: \mathcal{K} \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ be viewed as a “keyed” function family

\[
\text{Exp}_{F}^{\text{prp}}(A): \quad \text{Oracle } \mathcal{O}(X):
\]
\[
\begin{align*}
K & \xleftarrow{\$} \mathcal{K} \\
\pi & \xleftarrow{\$} \text{Perm}(n) \\
b & \xleftarrow{\$} \{0, 1\} \\
b' & \xleftarrow{\$} A^{\mathcal{O}(\cdot)} \\
\text{If } b' = b & \text{ then Return 1} \\
\text{Else} & \text{ Return 0}
\end{align*}
\]

\[
\text{Adv}_{F}^{\text{prp}}(A) = 2 \Pr(\text{Exp}_{F}^{\text{prp}}(A) = 1) - 1
\]

“My oracle is…”
The PRP-PRF Switching Lemma

Let $E: \mathcal{K} \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ be viewed as a “keyed” function family.

Let $A$ be an adversary, asking $q$ queries to its single oracle. Then

$$\left| \text{Adv}^{\text{prp}}_E (A) - \text{Adv}^{\text{prf}}_E (A) \right| \leq \frac{0.5q^2}{2^n}$$
The PRP-PRF Switching Lemma

Let \( E : \mathcal{K} \times \{0, 1\}^n \rightarrow \{0, 1\}^n \) be viewed as a “keyed” function family.

Let \( A \) be an adversary, asking \( q \) queries to its single oracle. Then

\[
\left| \text{Adv}^\text{prp}_E(A) - \text{Adv}^\text{prf}_E(A) \right| \leq \frac{0.5q^2}{2^n}
\]

So, for example,

\[
\text{Adv}^\text{ind-cpa}_{\text{CTR}[AES]}(A) \leq 2\text{Adv}^\text{prf}_{AES}(B)
\]

\[
\leq 2\text{Adv}^\text{prp}_{AES}(B) + \frac{q^2}{2^n}
\]
What about cipher-block-chaining (CBC) mode?

CBC mode appears in IPSec, SSH, TLS, …

How to handle the IV?
- Fixed IV?
- Counter IV?
- Random IV?
CBC with a fixed IV

Can the adversary easily guess the bit?
CBC with a counter IV

Can the adversary easily guess the bit?
Claim: If $F: \mathcal{K} \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ is a secure PRF, then $\text{CBC}[F]$ (CBC-mode, with a random IV, over F) is IND-CPA secure.

Proof idea: break the proof into two steps

1. replace $F_k$ with a random function $f$, and argue that any adversary that can detect this, can “break” PRF-security of $F$

2. analyze IND-CPA security of $\text{CBC}[\text{Func}(n, n)]$

$$
\text{Adv}_{\text{CBC}[F]}^{\text{ind-cpa}}(A) \leq \text{Adv}_{\text{CBC}[\text{Func}(n, n)]}^{\text{ind-cpa}}(A) + \text{Adv}_F^{\text{prf}}(A)
\leq \frac{0.5(\mu/n)^2}{2^n} + \text{Adv}_F^{\text{prf}}(A)
$$
Until $f$ is called on the same value twice, the ciphertext blocks are random and independent of the message blocks.

There are $\mu/n$ chances for an $f$-domain “collision”
Privacy? ✓ What about authenticity?

Authenticity: Alice wants to be sure she’s received Bob’s message

Might alter the ciphertext

Is $C'$ an authentic ctxt from Bob?

Is $M'$ an authentic ptxt from Bob?
Privacy? What about authenticity?

Authenticity: Alice wants to be sure she’s received Bob’s message.

- Might alter the ciphertext
  - Is $C'$ an authentic ctxt from Bob?
- Is $M'$ an authentic ptxt from Bob?
First of all, we need a syntactic addition

(New primitive, new syntax!)

**Key-generation algorithm**

\[ \mathcal{K} \text{ samples from a set of the same name} \]

**Encryption algorithm**

\[ \mathcal{E} : \mathcal{K} \times \{0, 1\}^* \rightarrow \{0, 1\}^* \cup \{\perp\} \]

**Decryption algorithm**

\[ \mathcal{D} : \mathcal{K} \times \{0, 1\}^* \rightarrow \{0, 1\}^* \cup \{\perp\} \]

Decryption now has the ability to "complain"
Folklore idea: add “redundancy” to encryption

Decryption: just like CBC, except return ⊥ if last block isn’t all zeros
Can you forge an authentic ciphertext?

\[ C_0, C_1, C_2, C_3 \] decrypts properly,
and so is “authentic” by the
if-it-decrypts-the-authentic measure…
So what’s wrong?

It’s not that CBC-mode is “bad”, it’s just that traditional encryption schemes have been designed to provide

PRIVACY ONLY
This *can* be made to work... (more later)
A notion of “authenticity”: Integrity of Ciphertexts (INT-CTXT)

$E_K(\cdot)$

$O(\cdot)$

$E_K(M)$

$M$

$C$

$D_K(M) \text{ or } \perp$

random bit $b$
A notion of "authenticity": Integrity of Ciphertexts (INT-CTX)

\[ \text{Exp}^{\text{int-ctxt}}_\Pi (A) : \]
\[ K \leftarrow \mathcal{K} \]
\[ b \leftarrow \{0, 1\} \]
\[ b' \leftarrow A^\mathcal{E}_K(\cdot), \mathcal{O}(\cdot) \]
If \( b' = b \) then Return 1
Return 0

Oracle \( \mathcal{O}(C) : \)
If \( b = 0 \) then Return \( \perp \)
Return \( \mathcal{D}_K(C) \)

\[ \text{Adv}^{\text{int-ctxt}}_\Pi (A) = 2 \Pr(\text{Exp}^{\text{int-ctxt}}_\Pi (A) = 1) - 1 \]

Adversarial "resources":
the number of oracle queries, \( q_e, q_d \)
the total length in bits of the queries, \( \mu_e, \mu_d \)
the time-complexity of the adversary, \( t \)
A notion of “authenticity”: Integrity of Ciphertexts (INT-CTXT)

\[ \text{Exp}_{\Pi}^{\text{int-ctx}}(A): \]
\[ K \xleftarrow{\$} \mathcal{K} \]
\[ b \xleftarrow{\$} \{0, 1\} \]
\[ b' \xleftarrow{\$} A \mathcal{E}_K(\cdot), \mathcal{O}(\cdot) \]
\[ \text{If } b' = b \text{ then Return 1} \]
\[ \text{Return 0} \]

Oracle \( \mathcal{O}(\cdot) \):
\[ \text{If } b = 0 \text{ then Return } \bot \]
\[ \text{Return } \mathcal{D}_K(C) \]

\[ \text{Adv}_{\Pi}^{\text{int-ctx}}(A) = 2 \Pr(\text{Exp}_{\Pi}^{\text{int-ctx}}(A) = 1) - 1 \]

To prevent “trivial wins” of the game, adversary is forbidden to ask \( C \) of the right oracle if \( C \) was returned by the left oracle.
Folklore idea: add “redundancy” to encryption

Decryption: just like CBC, except return ⊥ if last block isn’t all zeros
Folklore idea: add “redundancy” to encryption

Decryption: just like CBC, except return $\bot$ if last block isn’t all zeros

EXERCISE: Can you forge an authentic ciphertext, and win the INT-CTXT game?
Building a simple INT-CTXT secure encryption scheme

Let $F: \mathcal{K} \times \{0, 1\}^* \rightarrow \{0, 1\}^n$ be a function family.

Define an encryption scheme $\Pi[F]$ as follows:

$$\mathcal{E}_K(M) = M \parallel F_K(M)$$

$$\mathcal{D}_K(X \parallel T) = \begin{cases} X & \text{if } F_K(X) = T \\ \bot & \text{otherwise} \end{cases}$$
**Claim:** if $F : \mathcal{K} \times \{0, 1\}^* \rightarrow \{0, 1\}^n$ is a secure PRF, then $\Pi[F]$ is an INT-CTXT secure encryption scheme.

**Proof idea:** break the proof into two steps

1. replace $F_k$ with a random function $f$, and argue that any adversary that can detect this can “break” PRF-security of $F$

2. analyze INT-CTXT security of $\Pi[\text{Func}(\ast, n)]$

$$\text{Adv}_{\Pi[F]}^{\text{int-ctxt}}(A) = 2\text{Adv}_{F}^{\text{prf}}(B) + \frac{qd}{2^n}$$
\[
\frac{1}{2} \text{Adv}_{\Pi[F]}^{\text{int-ctxt}}(A) + \frac{1}{2} = \Pr(\text{Exp}_{\Pi[F]}^{\text{int-ctxt}}(A) = 1)
\]
\[
= \Pr(\text{Exp}_{\Pi[F]}^{\text{int-ctxt}}(A) = 1) - \Pr(\text{Exp}_{\Pi[\text{Func}(\ast,n)]}^{\text{int-ctxt}}(A) = 1)
\]
\[
+ \Pr(\text{Exp}_{\Pi[\text{Func}(\ast,n)]}^{\text{int-ctxt}}(A) = 1)
\]
\[
\leq \text{Adv}_{F}^{\text{prf}}(B) + \Pr(\text{Exp}_{\Pi[\text{Func}(\ast,n)]}^{\text{int-ctxt}}(A) = 1)
\]
\[
\frac{1}{2} \text{Adv}_{\Pi[F]}^{\text{int-ctxt}}(A) + \frac{1}{2} = \Pr(\text{Exp}_{\Pi[F]}^{\text{int-ctxt}}(A) = 1) \\
= \Pr(\text{Exp}_{\Pi[F]}^{\text{int-ctxt}}(A) = 1) - \Pr(\text{Exp}_{\Pi[Func(*,n)]}^{\text{int-ctxt}}(A) = 1) \\
+ \Pr(\text{Exp}_{\Pi[Func(*,n)]}^{\text{int-ctxt}}(A) = 1) \\
\leq \text{Adv}_F^{\text{prf}}(B) + \Pr(\text{Exp}_{\Pi[Func(*,n)]}^{\text{int-ctxt}}(A) = 1)
\]

Adversary \(B^{g(\cdot)}\):
Run \(A\) When \(A\) asks \(M\) to its left oracle:
   Respond with \(M \parallel g(M)\)
When \(A\) asks \(X \parallel T\) to its right oracle:
   Respond with \(X\) if \(g(X) = T\); else \(\perp\)
When \(A\) halts with output bit \(b\):
   Return \(b\)

If the bit \(b\) in the PRF experiment is 1 (resp. 0), then B simulates the INT-CTX experiment for \(\Pi[F]\) (resp. \(\Pi[Func(*,n)]\)
\[
\frac{1}{2} \text{Adv}_{\Pi[F]}^{\text{int-ctxt}}(A) + \frac{1}{2} = \Pr(\text{Exp}_{\Pi[F]}^{\text{int-ctxt}}(A) = 1) \\
= \Pr(\text{Exp}_{\Pi[F]}^{\text{int-ctxt}}(A) = 1) - \Pr(\text{Exp}_{\Pi[\text{Func}(\ast,n)]}^{\text{int-ctxt}}(A) = 1) \\
+ \Pr(\text{Exp}_{\Pi[\text{Func}(\ast,n)]}^{\text{int-ctxt}}(A) = 1) \\
\leq \text{Adv}_{F}^{\text{prf}}(B) + \Pr(\text{Exp}_{\Pi[\text{Func}(\ast,n)]}^{\text{int-ctxt}}(A) = 1)
\]

Hence,
\[
\text{Adv}_{\Pi[F]}^{\text{int-ctxt}}(A) \leq 2\text{Adv}_{F}^{\text{prf}}(B) + 2\Pr(\text{Exp}_{\Pi[\text{Func}(\ast,n)]}^{\text{int-ctxt}}(A) = 1) - 1 \\
= 2\text{Adv}_{F}^{\text{prf}}(B) + \text{Adv}_{\Pi[\text{Func}(\ast,n)]}^{\text{int-ctxt}}(A)
\]
Consider $\Pi[\text{Func}(*,n)]$

\[ \mathcal{E}_K(M) = M \parallel f(M) \]

\[ \mathcal{D}_K(X \parallel T) = \begin{cases} 
X & \text{if } f(X) = T \\
\bot & \text{otherwise}
\end{cases} \]
Consider $\Pi[\text{Func}(*, n)]$

$f$ is a random function

$$E_K(M) = M \ || \ f(M)$$

$$D_K(X \ || \ T) = \begin{cases} X & \text{if } f(X) = T \\ \bot & \text{otherwise} \end{cases}$$

Decryption cases

0. $(X, T)$ old: not allowed (i.e. $T$ not the tag previously returned with $X$)

1. $X$ old, $T$ “new”: returns $\bot$ because $f$ is deterministic
Consider $\Pi[\text{Func}(\ast, n)]$

$f$ is a random function

$\mathcal{E}_K(M) = M || f(M)$

$\mathcal{D}_K(X || T) = \begin{cases} X & \text{if } f(X) = T \\ \bot & \text{otherwise} \end{cases}$

Decryption cases

0. $(X, T)$ old: not allowed

1. $X$ old, $T$ "new": returns $\bot$ because $f$ is deterministic

2. $X$ new, $T$ old: $f(x)$ uniformly random, $\Pr(f(X) = T) = 2^{-n}$

(i.e. $T$ not the tag previously returned with $X$)
Consider $\Pi[\text{Func}(\ast, n)]$

$f$ is a random function

$\mathcal{E}_K(M) = M \parallel f(M)$

$\mathcal{D}_K(X \parallel T) = \begin{cases} X & \text{if } f(X) = T \\ \bot & \text{otherwise} \end{cases}$

Decryption cases

0. $(X, T)$ old: not allowed

1. $X$ old, $T$ “new”: returns $\bot$ because $f$ is deterministic

2. $X$ new, $T$ old: $f(x)$ uniformly random, $\Pr(f(X) = T) = 2^{-n}$

3. $X$ new, $T$ new: $f(x)$ uniformly random, $\Pr(f(X) = T) = 2^{-n}$
\[
\frac{1}{2} \text{Adv}_{\Pi[F]}^{\text{int-ctxt}}(A) + \frac{1}{2} = \Pr(\text{Exp}_{\Pi[F]}^{\text{int-ctxt}}(A) = 1)
\]

\[
= \Pr(\text{Exp}_{\Pi[F]}^{\text{int-ctxt}}(A) = 1) - \Pr(\text{Exp}_{\Pi[\text{Func}(*,n)]}^{\text{int-ctxt}}(A) = 1)
\]

\[
+ \Pr(\text{Exp}_{\Pi[\text{Func}(*,n)]}^{\text{int-ctxt}}(A) = 1)
\]

\[
\leq \text{Adv}_F^{\text{prf}}(B) + \Pr(\text{Exp}_{\Pi[\text{Func}(*,n)]}^{\text{int-ctxt}}(A) = 1)
\]

Hence,

\[
\text{Adv}_{\Pi[F]}^{\text{int-ctxt}}(A) \leq 2\text{Adv}_F^{\text{prf}}(B) + 2\Pr(\text{Exp}_{\Pi[\text{Func}(*,n)]}^{\text{int-ctxt}}(A) = 1) - 1
\]

\[
= 2\text{Adv}_F^{\text{prf}}(B) + \text{Adv}_{\Pi[\text{Func}(*,n)]}^{\text{int-ctxt}}(A)
\]

\[
\leq 2\text{Adv}_F^{\text{prf}}(B) + \frac{qd}{2n}
\]
Adding IND-CPA

Let \( F : \mathcal{K}_F \times \{0, 1\}^* \to \{0, 1\}^n \) be a function family.

Let \( \Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D}) \) be an encryption scheme.

Define an encryption scheme \( \Pi' = (\overline{\mathcal{K}}, \overline{\mathcal{E}}, \overline{\mathcal{D}}) \) as follows:

\[
\overline{\mathcal{K}} : \text{Return (} K_1, K_2 \text{) } \xleftarrow{\$} \mathcal{K} \times \mathcal{K}_F
\]

\[
\overline{\mathcal{E}}_{K_1}(M) = \mathcal{E}_{K_1}(M) \parallel F_{K_2}(\mathcal{E}_{K_1}(M))
\]

\[
\overline{\mathcal{D}}_{K_1, K_2}(C \parallel T) = \begin{cases} 
\mathcal{D}_{K_1}(C) & \text{if } F_{K_2}(C) = T \\
\bot & \text{otherwise}
\end{cases}
\]

This is called “Encrypt-then-PRF”
Claim: if \( F: \mathcal{K}_F \times \{0, 1\}^* \rightarrow \{0, 1\}^n \) is a secure PRF, and \( \Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D}) \) is IND-CPA secure, then \( \Pi = (\overline{\mathcal{K}}, \overline{\mathcal{E}}, \overline{\mathcal{D}}) \) is both IND-CPA and INT-CTXT secure
Claim: if $F: \mathcal{K}_F \times \{0, 1\}^* \rightarrow \{0, 1\}^n$ is a secure PRF, and $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is IND-CPA secure, then $\overline{\Pi} = (\overline{\mathcal{K}}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ is both IND-CPA and INT-CTXT secure.

Let’s do the easy part first: INT-CTXT

$$\frac{1}{2} \text{Adv}_{\overline{\Pi}[F]}^{\text{int-ctxt}}(A) + \frac{1}{2} = \text{Pr}(\text{Exp}_{\overline{\Pi}[F]}^{\text{int-ctxt}}(A) = 1)$$

$$= \text{Pr}(\text{Exp}_{\overline{\Pi}[F]}^{\text{int-ctxt}}(A) = 1) - \text{Pr}(\text{Exp}_{\overline{\Pi}[\text{Func}(\ast, n)]}^{\text{int-ctxt}}(A) = 1)$$

$$+ \text{Pr}(\text{Exp}_{\overline{\Pi}[\text{Func}(\ast, n)]}^{\text{int-ctxt}}(A) = 1)$$

$$\leq \text{Adv}_F^{\text{prf}}(B) + \text{Pr}(\text{Exp}_{\overline{\Pi}[\text{Func}(\ast, n)]}^{\text{int-ctxt}}(A) = 1)$$
Claim: if $F: \mathcal{K}_F \times \{0, 1\}^* \rightarrow \{0, 1\}^n$ is a secure PRF, and $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is IND-CPA secure, then $\Pi = (\overline{\mathcal{K}}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ is both IND-CPA and INT-CTXT secure.

Let’s do the easy part first: INT-CTXT

\[
\frac{1}{2} \text{Adv}_{\Pi[F]}^{\text{int-ctxt}}(A) + \frac{1}{2} = \Pr(\text{Exp}_{\Pi[F]}^{\text{int-ctxt}}(A) = 1)
\]

\[
= \Pr(\text{Exp}_{\Pi[F]}^{\text{int-ctxt}}(A) = 1) - \Pr(\text{Exp}_{\Pi[\text{Func}(\cdot, n)]}^{\text{int-ctxt}}(A) = 1)
\]

\[
+ \Pr(\text{Exp}_{\Pi[\text{Func}(\cdot, n)]}^{\text{int-ctxt}}(A) = 1)
\]

\[
\leq \text{Adv}_F^{\text{prf}}(B) + \Pr(\text{Exp}_{\Pi[\text{Func}(\cdot, n)]}^{\text{int-ctxt}}(A) = 1)
\]

Adversary $B^g(\cdot)$:

- $K_1 \leftarrow^s \mathcal{K}$
- Run $A$ When $A$ asks $M$ to its left oracle:
  - $C \leftarrow^s \mathcal{E}_{K_1}(M)$
  - Respond with $C \parallel g(C)$
- When $A$ asks $X \parallel T$ to its right oracle:
  - Respond with $\mathcal{D}_{K_1}(X)$ if $g(X) = T$; else $\bot$
- When $A$ halts with output bit $b$:
  - Return $b$

If the bit $b$ in the PRF experiment is 1 (resp. 0), then B simulates the INT-CTXT experiment for $\Pi[F]$ (resp. $\Pi[\text{Func}(\cdot, n)]$)
Claim: if $F: \mathcal{K}_F \times \{0, 1\}^* \rightarrow \{0, 1\}^n$ is a secure PRF, and $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is IND-CPA secure, then $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is both IND-CPA and INT-CTXT secure.

Let’s do the easy part first: INT-CTXT

$$\frac{1}{2} \text{Adv}^{\text{int-ctxt}}_{\Pi[F]}(A) + \frac{1}{2} = \Pr(\text{Exp}_{\Pi[F]}^{\text{int-ctxt}}(A) = 1)$$

$$= \Pr(\text{Exp}_{\Pi[F]}^{\text{int-ctxt}}(A) = 1) - \Pr(\text{Exp}_{\Pi[\text{Func}(\ast,n)]}^{\text{int-ctxt}}(A) = 1)$$

$$+ \Pr(\text{Exp}_{\Pi[\text{Func}(\ast,n)]}^{\text{int-ctxt}}(A) = 1)$$

$$\leq \text{Adv}^{\text{prf}}_F(B) + \Pr(\text{Exp}_{\Pi[\text{Func}(\ast,n)]}^{\text{int-ctxt}}(A) = 1)$$

Hence,

$$\text{Adv}^{\text{int-ctxt}}_{\Pi[F]}(A) \leq 2\text{Adv}^{\text{prf}}_F(B) + 2\Pr(\text{Exp}_{\Pi[\text{Func}(\ast,n)]}^{\text{int-ctxt}}(A) = 1) - 1$$

$$= 2\text{Adv}^{\text{prf}}_F(B) + \text{Adv}_{\Pi[\text{Func}(\ast,n)]}^{\text{int-ctxt}}(A)$$

$$\leq 2\text{Adv}^{\text{prf}}_F(B) + \frac{qd}{2^n}$$
Claim: if $F: \mathcal{K}_F \times \{0, 1\}^* \rightarrow \{0, 1\}^n$ is a secure PRF, and $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is IND-CPA secure, then $\overline{\Pi} = (\overline{\mathcal{K}}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ is both IND-CPA and INT-CTXT secure.

Now the “new” part: IND-CPA.

But this is even easier! $\overline{\mathcal{E}}_{K_1}(M) = \mathcal{E}_{K_1}(M) \parallel F_{K_2}(\mathcal{E}_{K_1}(M))$

$$\frac{1}{2} \text{Adv}^{\text{ind-CPA}}_{\overline{\Pi}[F]}(A) + \frac{1}{2} = \Pr(\text{Exp}^{\text{ind-CPA}}_{\overline{\Pi}[F]}(A) = 1)$$
$$= \Pr(\text{Exp}^{\text{ind-CPA}}_{\overline{\Pi}[F]}(B) = 1)$$

Hence,

$$\text{Adv}^{\text{ind-CPA}}_{\overline{\Pi}[F]}(A) \leq \text{Adv}^{\text{ind-CPA}}_{\overline{\Pi}[F]}(B)$$

Where this reduction $B$ simulates the $F_{K_2}$ part of encryption.
The three “Generic Composition” authenticated encryption schemes

Encrypt-then-PRF:

\[ \overline{E}_{K_1}(M) = E_{K_1}(M) \parallel F_{K_2}(E_{K_1}(M)) \quad (\text{IPSec}) \]

\[ \overline{D}_{K_1,K_2}(C \parallel T) = \begin{cases} D_{K_1}(C) & \text{if } F_{K_2}(C) = T \\ \bot & \text{otherwise} \end{cases} \]
The three “Generic Composition” authenticated encryption schemes

Encrypt-then-PRF:

\[ \overline{E}_{K_1}(M) = E_{K_1}(M) \| F_{K_2}(E_{K_1}(M)) \quad \text{(IPSec)} \]

\[ \overline{D}_{K_1,K_2}(C \| T) = \begin{cases} D_{K_1}(C) & \text{if } F_{K_2}(C) = T \\ \bot & \text{otherwise} \end{cases} \]

✓ IND-CPA
✓ INT-CTXT

PRF-then-Encrypt:

\[ \overline{E}_{K_1,K_2}(M) = E_{K_1}(M \| F_{K_2}(M)) \quad \text{(SSL/TLS)} \]

\[ \overline{D}_{K_1,K_2}(C) = \begin{cases} M' \| T \leftarrow D_{K_1}(C) & \text{then:} \\
\text{Return } M' \text{ if } F_{K_2}(M') = T \\
\text{Return } \bot \text{ otherwise} \end{cases} \]
The three “Generic Composition” authenticated encryption schemes

Encrypt-then-PRF:

\[ \overline{E}_{K_1}(M) = E_{K_1}(M) \parallel F_{K_2}(E_{K_1}(M)) \]  \hspace{1cm} \text{(IPSec)}

\[ \overline{D}_{K_1,K_2}(C \parallel T) = \begin{cases} D_{K_1}(C) & \text{if } F_{K_2}(C) = T \\ \bot & \text{otherwise} \end{cases} \]

\checkmark \text{ IND-CPA} \\
\checkmark \text{ INT-CTXT}

PRF-then-Encrypt:

\[ \overline{E}_{K_1,K_2}(M) = E_{K_1}(M \parallel F_{K_2}(M)) \] \hspace{1cm} \text{(SSL/TLS)}

\[ \overline{D}_{K_1,K_2}(C) = \begin{cases} M' \parallel T \leftarrow D_{K_1}(C) & \text{then:} \\
\text{Return } M' \text{ if } F_{K_2}(M') = T \\
\text{Return } \bot \text{ otherwise} \end{cases} \]
The three “Generic Composition” authenticated encryption schemes

Encrypt-then-PRF:

\[ \overline{E}_{K_1}(M) = E_{K_1}(M) \parallel F_{K_2}(E_{K_1}(M)) \]  

\( \overline{D}_{K_1,K_2}(C \parallel T) = \begin{cases} 
D_{K_1}(C) & \text{if } F_{K_2}(C) = T \\
\bot & \text{otherwise}
\end{cases} \)

\( \checkmark \) IND-CPA
\( \checkmark \) INT-CTXT

PRF-then-Encrypt:

\[ \overline{E}_{K_1,K_2}(M) = E_{K_1}(M \parallel F_{K_2}(M)) \]  

\( \overline{D}_{K_1,K_2}(C) = \begin{cases} 
M' \parallel T & T \leftarrow D_{K_1}(C) \text{ then:} \\
\text{Return } M' \text{ if } F_{K_2}(M') = T \\
\text{Return } \bot \text{ otherwise}
\end{cases} \)

\( \checkmark \) IND-CPA
\( \times \) INT-CTXT

PRF and Encrypt:
(or Encrypt and PRF)

\[ \overline{E}_{K_1,K_2}(M) = E_{K_1}(M) \parallel F_{K_2}(M) \]  

\( \overline{D}_{K_1,K_2}(C \parallel T) = \begin{cases} 
M' & T \leftarrow D_{K_1}(C) \text{ then:} \\
\text{Return } M' \text{ if } F_{K_2}(M') = T \\
\text{Return } \bot \text{ otherwise}
\end{cases} \)

\( \checkmark \) IND-CPA
\( \checkmark \) INT-CTXT

SSL/TLS

IPSec

SSH

\( \checkmark \) IND-CPA
\( \times \) INT-CTXT
The three “Generic Composition” authenticated encryption schemes

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Cipher Function</th>
<th>Authentication Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypt-then-PRF:</td>
<td>$\bar{E}<em>{K1}(M) = E</em>{K1}(M) \parallel F_{K2}(E_{K1}(M))$</td>
<td>✓ IND-CPA ✓ INT-CTXT</td>
</tr>
<tr>
<td>PRF-then-Encrypt:</td>
<td>$\bar{E}<em>{K1,K2}(M) = E</em>{K1}(M \parallel F_{K2}(M))$</td>
<td>✓ IND-CPA ✗ INT-CTXT</td>
</tr>
<tr>
<td>PRF and Encrypt: (or Encrypt and PRF)</td>
<td>$\bar{E}<em>{K1,K2}(M) = E</em>{K1}(M) \parallel F_{K2}(M)$</td>
<td>✗ IND-CPA ✗ INT-CTXT</td>
</tr>
</tbody>
</table>

$(IPSec)$

$\bar{D}_{K1,K2}(C \parallel T) = \begin{cases} D_{K1}(C) & \text{if } F_{K2}(C) = T \\ \bot & \text{otherwise} \end{cases}$

$(SSL/TLS)$

$\bar{D}_{K1,K2}(C) = \begin{cases} M' \parallel T & \text{if } T \leftarrow D_{K1}(C) \text{ then:} \\ \text{Return } M' \text{ if } F_{K2}(M') = T \\ \text{Return } \bot \text{ otherwise} \end{cases}$

$(SSH)$

$\bar{D}_{K1,K2}(C \parallel T) = \begin{cases} M' & \text{if } T \leftarrow D_{K1}(C) \text{ then:} \\ \text{Return } M' \text{ if } F_{K2}(M') = T \\ \text{Return } \bot \text{ otherwise} \end{cases}$
The three “Generic Composition” authenticated encryption schemes

Encrypt-then-PRF:

\[
\overline{E}_{K_1}(M) = E_{K_1}(M) \parallel F_{K_2}(E_{K_1}(M))
\]  

(IPSec)

\[
\overline{D}_{K_1,K_2}(C \parallel T) = \begin{cases} 
D_{K_1}(C) & \text{if } F_{K_2}(C) = T \\
\bot & \text{otherwise}
\end{cases}
\]

PRF-then-Encrypt:

\[
\overline{E}_{K_1,K_2}(M) = E_{K_1}(M \parallel F_{K_2}(M))
\]  

(SSL/TLS)

\[
\overline{D}_{K_1,K_2}(C) = \begin{cases} 
M' \parallel T \leftarrow D_{K_1}(C) & \text{then:} \\
\quad \text{Return } M' & \text{if } F_{K_2}(M') = T \\
\quad \text{Return } \bot & \text{otherwise}
\end{cases}
\]

PRF and Encrypt: (or Encrypt and PRF)

\[
\overline{E}_{K_1,K_2}(M) = E_{K_1}(M) \parallel F_{K_2}(M)
\]  

(SSH)

\[
\overline{D}_{K_1,K_2}(C \parallel T) = \begin{cases} 
M' \leftarrow D_{K_1}(C) & \text{then:} \\
\quad \text{Return } M' & \text{if } F_{K_2}(M') = T \\
\quad \text{Return } \bot & \text{otherwise}
\end{cases}
\]

\[\checkmark\ \text{IND-CPA}\]
\[\checkmark\ \text{INT-CTXT}\]

\[\times\ \text{IND-CPA}\]
\[\times\ \text{INT-CTXT}\]

EXERCISE: Explain why “PRF and Encrypt” is not generically IND-CPA
Consider \( \mathcal{E}_{K_1}(X) = 0 \parallel \mathcal{E}'_{K_1}(X) \)
\( \mathcal{D}_{K_1}(b \parallel C) = \mathcal{D}'_{K_1}(C) \) which is IND-CPA if \( \mathcal{E}'_{K_1}(X) \) is…

**PRF-then-Encrypt:**

\[
\bar{\mathcal{E}}_{K_1,K_2}(M) = \mathcal{E}_{K_1}(M \parallel F_{K_2}(M))
\]

\( \checkmark \) IND-CPA
\( \times \) INT-CTXT

\[
\bar{\mathcal{D}}_{K_1,K_2}(C) = \begin{cases} 
M' \parallel T \leftarrow \mathcal{D}_{K_1}(C) & \text{then:} \\
\quad \text{Return } M' \text{ if } F_{K_2}(M') = T \\
\quad \text{Return } \bot \text{ otherwise}
\end{cases}
\]

**PRF and Encrypt:**

(or Encrypt and PRF)

\[
\bar{\mathcal{E}}_{K_1,K_2}(M) = \mathcal{E}_{K_1}(M) \parallel F_{K_2}(M)
\]

\( \times \) IND-CPA
\( \times \) INT-CTXT

\[
\bar{\mathcal{D}}_{K_1,K_2}(C \parallel T) = \begin{cases} 
M' \leftarrow \mathcal{D}_{K_1}(C) & \text{then:} \\
\quad \text{Return } M' \text{ if } F_{K_2}(M') = T \\
\quad \text{Return } \bot \text{ otherwise}
\end{cases}
\]

(Violating INT-CTXT)
Privacy? ✓ What about authenticity?

Authenticity: Alice wants to be sure she’s received Bob’s message.
Authenticity: Alice wants to be sure she’s received Bob’s message.

Is $C'$ an authentic ctxt from Bob?
Is $M'$ an authentic ptxt from Bob?
Another notion of “authenticity”: Integrity of Plaintexts (INT-PTXT)

Adversary wins if it asks C such that
1. $\bot \neq M' \leftarrow D_K(C)$
2. $M'$ never asked to $E_K(\cdot)$

Achieved (generically) by “PRF-then-Encrypt”
- Strictly weaker security goal
- Requires calling applications to be aware of repeated plaintexts
- Efficient schemes achieve INT-CTXT already

Stick with INT-CTXT if possible!
Let’s return to this idea
Strong PRPs

Let \( E : \mathcal{K} \times \{0,1\}^N \rightarrow \{0,1\}^N \) be a permutation family

\[
\begin{align*}
\text{Exp}_{E}^{\text{sprp}}(A): & \\
K & \leftarrow \mathcal{K} \\
\pi & \leftarrow \text{Perm}(N) \\
b & \leftarrow \{0,1\} \\
b' & \leftarrow A^{O(\cdot),O^{-1}(\cdot)} \\
\text{If } b' = b \text{ then Return } 1 \\
\text{Return } 0
\end{align*}
\]

\[
\text{Oracle } O(X): \\
\text{If } b = 1 \text{ Return } E_K(X) \\
\text{Else Return } \pi(X)
\]

\[
\text{Oracle } O^{-1}(Y): \\
\text{If } b = 1 \text{ Return } E_K^{-1}(Y) \\
\text{Else Return } \pi^{-1}(Y)
\]

\[
\text{Adv}_{E}^{\text{sprp}}(A) = 2 \Pr(\text{Exp}_{E}^{\text{sprp}}(A) = 1) - 1
\]

It’s easy to extend this to the VIL setting, by considering \( E : \mathcal{K} \times \mathcal{S} \rightarrow \mathcal{S}, \) with \( \mathcal{S} \subset \{0,1\}^* \), to be length-preserving.
Intuition: if you encrypt new messages, with redundancy...

\[ M \| 0^{80} \]

... then outputs look like random bitstrings (subject to permutivity)
Intuition: if you flip any bit of the output and decrypt…

… then “plaintexts” random, and unlikely to have correct redundancy
Of course, we’re not guaranteed that messages are new, so we add a per-message “nonce” (number used once)

\[ N \ || \ M \ || \ 0^{80} \]

This is the “Encode-Encipher” paradigm, due to Bellare and Rogaway
A nonce-based encryption scheme is a triple of algorithms:

- **Key-generation algorithm**: $\mathcal{K}$ samples from a set of the same name

- **Encryption algorithm**: $\mathcal{E}: \mathcal{N} \times \mathcal{K} \times \{0, 1\}^* \rightarrow \{0, 1\}^* \cup \{\bot\}$

- **Decryption algorithm**: $\mathcal{D}: \mathcal{N} \times \mathcal{K} \times \{0, 1\}^* \rightarrow \{0, 1\}^* \cup \{\bot\}$

(See Rogaway’s Nonce-Based Encryption Paper)
A nonce-based encryption scheme is a triple of algorithms

- **Key-generation algorithm**: $\mathcal{K}$ samples from a set of the same name
- **Encryption algorithm**: $\mathcal{E} : \mathcal{N} \times \mathcal{K} \times \{0, 1\}^* \rightarrow \{0, 1\}^* \cup \{\bot\}$
- **Decryption algorithm**: $\mathcal{D} : \mathcal{N} \times \mathcal{K} \times \{0, 1\}^* \rightarrow \{0, 1\}^* \cup \{\bot\}$
A nonce-based encryption scheme is a triple of algorithms

**Key-generation algorithm** \( \mathcal{K} \) samples from a set of the same name

**Encryption algorithm** \[ E: \mathcal{N} \times \mathcal{K} \times \{0, 1\}^* \rightarrow \{0, 1\}^* \cup \{\bot\} \] Deterministic! \( C \leftarrow E^N_K(M) \)

**Decryption algorithm** \[ D: \mathcal{N} \times \mathcal{K} \times \{0, 1\}^* \rightarrow \{0, 1\}^* \cup \{\bot\} \]
**IND-CPA in the nonce-based setting**

\[
\text{\texttt{Exp}}_{\Pi}^{\text{ind-cpa}}(A): \\
K \leftarrow^\$ \mathcal{K} \\
d \leftarrow^\$ \{0, 1\} \\
d' \leftarrow^\$ A^\mathcal{O}(\cdot, \cdot, \cdot) \\
\text{If } d' = d \text{ then Return 1} \\
\text{Return 0}
\]

\[
\text{\texttt{Adv}}_{\Pi}^{\text{ind-cpa}}(A) = 2 \Pr(\text{\texttt{Exp}}_{\Pi}^{\text{ind-cpa}}(A) = 1) - 1
\]

Restrictions:

1. \(|M_0| = |M_1|

2. No nonce-message pair \((N, M_0, \cdot)\) or \((N, \cdot, M_1)\) repeated

“Nonces” are meant to be used once.

An adversary that never repeats a nonce is called “nonce-respecting”
Let’s define a nonce-based encryption scheme from an SPRP.

Let $\mathcal{N} = \{0, 1\}^{128}$ and let $\mathcal{S} \subset \{0, 1\}^*$ contain all strings up to length $128+80+L$ for some $L > 0$.
Let's define a nonce-based encryption scheme from an SPRP.

Let \( \mathcal{N} = \{0, 1\}^{128} \) and let \( \mathcal{S} \subset \{0, 1\}^* \) contain all strings up to length \( 128 + 80 + L \) for some \( L > 0 \).

Let \( E : \mathcal{K} \times \mathcal{S} \to \mathcal{S} \) be a length-preserving permutation family.

\[
\mathcal{E}_K^N(M) = E_K(N \parallel M \parallel 0^{80})
\]

\[
\mathcal{D}_K^N(C) : \begin{cases} 
X \leftarrow E_K^{-1}(C) \\
\text{Parse } X \text{ into } N, M, T \text{ where } |T| = 80 \\
\text{If parse fails, Return } \perp \\
\text{If } T \neq 0^{80} \text{ then Return } \perp \\
\text{Return } M
\end{cases}
\]
Let’s define a nonce-based encryption scheme from an SPRP.

Let $\mathcal{N} = \{0, 1\}^{128}$ and let $\mathcal{S} \subset \{0, 1\}^*$ contain all strings up to length $128+80+L$ for some $L > 0$.

Let $E: \mathcal{K} \times \mathcal{S} \rightarrow \mathcal{S}$ be a length-preserving permutation family.

\[
\mathcal{E}_K^N(M) = E_K(N \parallel M \parallel 0^{80})
\]

\[
\mathcal{D}_K^N(C) : \begin{cases} 
X \leftarrow E_K^{-1}(C) \\
\text{Parse } X \text{ into } N, M, T \text{ where } |T| = 80 \\
\text{If parse fails, Return } \bot \\
\text{If } T \neq 0^{80} \text{ then Return } \bot \\
\text{Return } M
\end{cases}
\]

Claim: if $E: \mathcal{K} \times \mathcal{S} \rightarrow \mathcal{S}$ is a secure SPRP, then this scheme is both (nonce-based) IND-CPA and (nonce-based) INT-CTXT secure.

Proof: exercise (you might need a “bi-directional” version of the PRP-PRF switching lemma...).
Proof intuition:

1. Replace $E_K(\cdot), E_K^{-1}(\cdot)$ with $\pi(\cdot), \pi^{-1}(\cdot)$
Proof intuition:

1. Replace $E_K(\cdot), E_K^{-1}(\cdot)$ with $\pi(\cdot), \pi^{-1}(\cdot)$

2. Replace $\pi(\cdot), \pi^{-1}(\cdot)$ with two independent random functions $f_1(\cdot), f_2(\cdot)$
Proof intuition:

1. Replace $E_K(\cdot), E_K^{-1}(\cdot)$ with $\pi(\cdot), \pi^{-1}(\cdot)$

2. Replace $\pi(\cdot), \pi^{-1}(\cdot)$ with two independent random functions $f_1(\cdot), f_2(\cdot)$

3. Now uniform random strings in both “directions” if nonces are respected
What makes this work is that SPRPs are (so of) all-or-nothing objects

\[ \text{N} \| \text{M} \| 0^{80} \]

\[ E_K \]

\[ C \]

Change any bit of input = randomize entire output

Change any bit of output = randomize entire input

But this comes with a cost:

Loosely, every bit of output (input) must depend on every bit of input (output).
SPRPs generally seem to require two full “cryptographic passes”

Definitely NOT an SPRP, even if $E_K$ is.
CMC mode
(Halevi and Rogway)
Nonce-based encryption is an interesting area.

This is not IND-CPA secure in the nonce-based setting, even if nonces are respected.
Nonce-based encryption is an interesting area.

But this should work...
Nonce-based encryption is an interesting area.

If $f_1$ and $f_2$ are independent random functions (so we need $E$ to be a PRF under two random keys) then all $f_2$ inputs are random…

…what type of bound do you expect?
Yet more: Deterministic (Nonce-based) AE with “Associated Data” (AEAD)

Key-generation algorithm

$\mathcal{K}$ samples from a set of the same name

Encryption algorithm

$\mathcal{E} : (\mathcal{H} \times \mathcal{N}) \times \mathcal{K} \times \{0, 1\}^* \rightarrow \{0, 1\}^* \cup \{\perp\}$

Decryption algorithm

$\mathcal{D} : (\mathcal{H} \times \mathcal{N}) \times \mathcal{K} \times \{0, 1\}^* \rightarrow \{0, 1\}^* \cup \{\perp\}$

The “header” or “associated data” space

(See Rogaway’s AEAD Paper)
(See Rogaway and Shrimpton’s “Keywrap” Paper)
Here's one way to build a deterministic AEAD scheme: SIV mode.
Here’s one way to build a DAE scheme: SIV mode

If $F$ is a secure PRF, and $\mathcal{E}$ is IND-CPA against nonce-respecting adversaries, then this is a secure DAE scheme (IND-CPA and INT-CTXT) (also provides “nonce-misuse resistance”)
This is NOT the whole story of symmetric encryption!

Many other interesting kinds of symmetric encryption to explore

Message-locked encryption
Format-preserving encryption
Format-transforming encryption
Length-hiding AEAD
“Online” encryption
Key-dependent message encryption
...

Thanks!
Exercises, once more
Claim: Any encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ that is RoR-CPA secure, is also IND-CPA secure.

Proof idea: show the contrapositive, if a scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is not IND-CPA secure, then it is not RoR-CPA secure.

Let $A$ be an efficient IND-CPA adversary, gaining advantage $\text{Adv}_{\Pi}^{\text{ind-cpa}}(A)$.

We build an efficient RoR-CPA adversary $B$, that runs $A$ as a “black-box” subroutine, that gains advantage

$$2\text{Adv}_{\Pi}^{\text{ror-cpa}}(B) \geq \text{Adv}_{\Pi}^{\text{ind-cpa}}(A)$$

Conclusion: if $\text{Adv}_{\Pi}^{\text{ror-cpa}}(B)$ is small for all efficient $B$, then $\text{Adv}_{\Pi}^{\text{ind-cpa}}(A)$ must be small, too.

AT HOME EXERCISE: try to write the proof
Folklore idea: add “redundancy” to encryption

Decryption: just like CBC, except return ⊥ if last block isn’t all zeros

EXERCISE: Can you forge an authentic ciphertext, and win the INT-CTXT game?
The three "Generic Composition" authenticated encryption schemes

Encrypt-then-PRF:

\[ \overline{E}_{K1}(M) = E_{K1}(M) \parallel F_{K2}(E_{K1}(M)) \]  
\[ \overline{D}_{K1,K2}(C \parallel T) = \begin{cases} D_{K1}(C) & \text{if } F_{K2}(C) = T \\ \bot & \text{otherwise} \end{cases} \]

\(\checkmark\) IND-CPA 
\(\checkmark\) INT-CTX

PRF-then-Encrypt:

\[ \overline{E}_{K1,K2}(M) = E_{K1}(M \parallel F_{K2}(M)) \]  
\[ \overline{D}_{K1,K2}(C) = \begin{cases} M' \parallel T \leftarrow D_{K1}(C) \text{ then:} \\ \text{Return } M' \text{ if } F_{K2}(M') = T \\ \text{Return } \bot \text{ otherwise} \end{cases} \]

\(\checkmark\) IND-CPA 
\(\times\) INT-CTX

PRF and Encrypt: (or Encrypt and PRF)

\[ \overline{E}_{K1,K2}(M) = E_{K1}(M) \parallel F_{K2}(M) \]  
\[ \overline{D}_{K1,K2}(C \parallel T) = \begin{cases} M' \leftarrow D_{K1}(C) \text{ then:} \\ \text{Return } M' \text{ if } F_{K2}(M') = T \\ \text{Return } \bot \text{ otherwise} \end{cases} \]

\(\times\) IND-CPA 
\(\times\) INT-CTX

**EXERCISE:** Explain why “PRF and Encrypt” is not generically IND-CPA
Proof of the PRP-PRF Switching Lemma
\[ \text{Exp}_F^{\text{prp}}(A): \]
\[
\begin{align*}
K & \leftarrow \mathcal{K} \\
\pi & \leftarrow \text{Perm}(n) \\
b & \leftarrow \{0, 1\} \\
b' & \leftarrow A^{\mathcal{O}(\cdot)} \\
\text{If } b' = b \text{ then Return } 1 \\
\text{Return } 0 \\
\text{Oracle } \mathcal{O}(X): \\
\text{If } b = 0 \text{ then Return } \pi(X) \\
\text{Return } F_K(X)
\end{align*}
\]
\[ \text{Adv}_F^{\text{prp}}(A) = 2 \Pr(\text{Exp}_F^{\text{prp}}(A) = 1) - 1 \]

\[ \text{Exp}_F^{\text{prf}}(A): \]
\[
\begin{align*}
K & \leftarrow \mathcal{K} \\
f & \leftarrow \text{Func}(n, n) \\
b & \leftarrow \{0, 1\} \\
b' & \leftarrow A^{\mathcal{O}(\cdot)} \\
\text{If } b' = b \text{ then Return } 1 \\
\text{Return } 0 \\
\text{Oracle } \mathcal{O}(X): \\
\text{If } b = 0 \text{ then Return } f(X) \\
\text{Return } F_K(X)
\end{align*}
\]
\[ \text{Adv}_F^{\text{prf}}(A) = 2 \Pr(\text{Exp}_F^{\text{prf}}(A) = 1) - 1 \]

\[ \left| \text{Adv}_F^{\text{prp}}(A) - \text{Adv}_F^{\text{prf}}(A) \right| \leq \Pr \left( A^f(\cdot) \Rightarrow 1 \right) - \Pr \left( A^{\pi(\cdot)} \Rightarrow 1 \right) \leq \frac{0.5q^2}{2^n} \]

Requires care, but the reason for the “birthday term” is obvious!
\[ G0(A): \]
\[ b' \leftarrow A^{O(\cdot)} \]
Return \( b' \)

**Oracle** \( O(X) \):
\[ Y \leftarrow \{0, 1\}^n \]
If \( Y \in \text{Range}(P) \)
\[ \text{bad} \leftarrow \text{true} \]
\[ Y \leftarrow \text{Range}(P) \]
\[ P[X] \leftarrow Y \]
Return \( Y \)

---

\[ G1(A): \]
\[ b' \leftarrow A^{O(\cdot)} \]
Return \( b' \)

**Oracle** \( O(X) \):
\[ Y \leftarrow \{0, 1\}^n \]
If \( Y \in \text{Range}(P) \)
\[ \text{bad} \leftarrow \text{true} \]
\[ P[X] \leftarrow Y \]
Return \( Y \)

---

all values already assigned as outputs of the oracle

all values still free to be assigned as outputs
\[
\begin{align*}
\text{G0}(A): & \quad b' \overset{\$}{\leftarrow} A^{O(\cdot)} \\
\text{Return } b' \\
\text{Oracle } O(X): & \quad Y \overset{\$}{\leftarrow} \{0, 1\}^n \\
& \quad \text{If } Y \in \text{Range}(P) \\
& \quad \quad \text{bad } \leftarrow \text{true} \\
& \quad \quad Y \overset{\$}{\leftarrow} \text{Range}(P) \\
& \quad \quad P[X] \leftarrow Y \\
\text{Return } Y \\
\end{align*}
\]

\[
\begin{align*}
\text{G1}(A): & \quad b' \overset{\$}{\leftarrow} A^{O(\cdot)} \\
\text{Return } b' \\
\text{Oracle } O(X): & \quad Y \overset{\$}{\leftarrow} \{0, 1\}^n \\
& \quad \text{If } Y \in \text{Range}(P) \\
& \quad \quad \text{bad } \leftarrow \text{true} \\
& \quad \quad P[X] \leftarrow Y \\
\text{Return } Y \\
\end{align*}
\]

\[\Pr(A^f \Rightarrow 1) - \Pr(A^\pi \Rightarrow 1) = \Pr(G1(A) \Rightarrow 1) - \Pr(G0(A) \Rightarrow 1)\]
\[
\text{Fundamental lemma of game-playing (Bellare, Rogaway)}
\]

<table>
<thead>
<tr>
<th>$G0(A)$:</th>
<th>$G1(A)$:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b' \xleftarrow{$} A^{O(\cdot)}$</td>
<td>$b' \xleftarrow{$} A^{O(\cdot)}$</td>
</tr>
<tr>
<td>Return $b'$</td>
<td>Return $b'$</td>
</tr>
</tbody>
</table>

**Oracle $O(X)$:**

<table>
<thead>
<tr>
<th>$Y \xleftarrow{$} {0, 1}^n$</th>
<th>$Y \xleftarrow{$} {0, 1}^n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>If $Y \in \text{Range}(P)$</td>
<td>If $Y \in \text{Range}(P)$</td>
</tr>
<tr>
<td>bad $\leftarrow$ true</td>
<td>bad $\leftarrow$ true</td>
</tr>
<tr>
<td>$Y \xleftarrow{$} \text{Range}(P)$</td>
<td>$Y \xleftarrow{$} \text{Range}(P)$</td>
</tr>
<tr>
<td>$P[X] \leftarrow Y$</td>
<td>$P[X] \leftarrow Y$</td>
</tr>
<tr>
<td>Return $Y$</td>
<td>Return $Y$</td>
</tr>
</tbody>
</table>

\[
\Pr(A^f \Rightarrow 1) - \Pr(A^\pi \Rightarrow 1) = \Pr(G1(A) \Rightarrow 1) - \Pr(G0(A) \Rightarrow 1) 
\leq \Pr(G1(A): \text{bad} = \text{true})
\]
\[
\begin{align*}
G0(A) & : \\
b' & \overset{\$}{\leftarrow} A^{O(\cdot)} \\
& \text{Return } b' \\
\text{Oracle } O(X) & : \\
Y & \overset{\$}{\leftarrow} \{0, 1\}^n \\
& \text{If } Y \in \text{Range}(P) \\
& \quad \text{bad } \leftarrow \text{true} \\
& Y \overset{\$}{\leftarrow} \text{Range}(P) \\
P[X] & \leftarrow Y \\
& \text{Return } Y
\end{align*}
\]

\[
\begin{align*}
G1(A) & : \\
b' & \overset{\$}{\leftarrow} A^{O(\cdot)} \\
& \text{Return } b' \\
\text{Oracle } O(X) & : \\
Y & \overset{\$}{\leftarrow} \{0, 1\}^n \\
& \text{If } Y \in \text{Range}(P) \\
& \quad \text{bad } \leftarrow \text{true} \\
P[X] & \leftarrow Y \\
& \text{Return } Y
\end{align*}
\]

\[
\begin{align*}
\Pr(A^f \Rightarrow 1) - \Pr(A^\pi \Rightarrow 1) &= \Pr(G1(A) \Rightarrow 1) - \Pr(G0(A) \Rightarrow 1) \\
& \leq \Pr(G1(A): \text{bad} = \text{true}) \\
& \leq \frac{0}{2^n} + \frac{1}{2^n} + \cdots + \frac{q - 1}{2^n} \\
& \leq \frac{0.5q^2}{2^n} \\
\end{align*}
\]

Fundamental lemma of game-playing
(Bellare, Rogaway)

union bound