Chosen Plaintext Attacks (CPA)
CPA-Security

\[ c_1 \leftarrow \text{Enc}_k(m_1) \]
\[ c_2 \leftarrow \text{Enc}_k(m_2) \]
Goals

New Attacks! Chosen Plaintext Attacks (henceforth CPA) is when Eve can choose to see some messages encoded. Formally she has Black Box for $ENC_k$. We will:

1. Define CPA for perfect security.
2. Define CPA for computational security.
Π = (GEN, ENC, DEC) be an enc sch, message space M.

Game: Alice and Eve are the players. Alice has full access to Π. Eve has access to ENC_k.

1. Alice $k \leftarrow \mathcal{K}$. Eve does NOT know $k$.

2. Eve picks $m_0, m_1 \in M$ Eve has black box for ENC_k.

3. Alice picks $m \in \{m_0, m_1\}$, $c \leftarrow ENC_k(m)$

4. Alice sends $c$ to Eve.

5. Eve outputs $m_0$ or $m_1$, hoping that her output is DEC_k(c).

6. Eve wins if she is right.

Note: ENC_k is randomized, so Eve can’t just comuter ENC_k(m_0) and ENC_k(m_1) and see which one is $c$.

Does Eve has a strategy that wins over half the time?
Perfect CPA-Security

- \( \Pi \) is secure against chosen-plaintext attacks (CPA-secure) if for all Eve,

\[
Pr[\text{Eve Wins}] \leq \frac{1}{2}
\]
Eve always wins if $ENC_k$ is Deterministic

1. Eve picks $m_0, m_1$. Finds $c_0 = ENC_k(m_0), c_1 = ENC_k(m_1)$.
2. Alice sends Eve $c = ENC_k(m_b)$. Eve has to determine $b$.
3. If $c = c_0$ then Eve sets $b' = 0$, if $c = c_1$ then Eve sets $b' = 1$.

Upshot: ALL deterministic schemes are CPA-insecure.
Comp CPA-Security

\[ \Pi = (\text{GEN}, \text{ENC}, \text{DEC}) \] be an enc sch, message space \( \mathcal{M} \). \( n \) is parameter.

**Game:** Alice and Eve are the players. Alice has full access to \( \Pi \). Eve has access to \( \text{ENC}_k \).

1. Alice \( k \leftarrow \mathcal{K} \cap \{0, 1\}^n \). Eve does NOT know \( k \).
2. Eve picks \( m_0, m_1 \in \mathcal{M}, |m_0| = |m_1| \)
3. Alice picks \( m \in \{m_0, m_1\}, c \leftarrow \text{ENC}_k(m) \)
4. Alice sends \( c \) to Eve.
5. Eve outputs \( m_0 \) or \( m_1 \), hoping that her output is \( \text{DEC}_k(c) \).
6. Eve wins if she is right.

Does Eve has a strategy that wins over half the time?
Comp. CPA-Security

- \( \Pi \) is **CPA Secure** if for all \( \text{PPT} \) Eves, there is a **neg function** \( \epsilon(n) \) such that

\[
\Pr[\text{Eve Wins}] \leq \frac{1}{2} + \epsilon(n)
\]
Randomized Encryption

1. Any Deterministic Encryption will NOT be CCA-secure.
2. Hence we have to use Randomized Encryption.
3. The issue is *not* an artifact of our definition: Even being able to tell if two messages are the same is a leak.
Deterministic Encryption (for contrast)

$n$ is a security parameter. A Deterministic Private-Key Encryption Scheme has message space $\mathcal{M}$, Key space $\mathcal{K} = \{0, 1\}^n$, and algorithms (GEN, ENC, DEC):

1. GEN generates keys $k \in \mathcal{K}$.
2. $ENC_k$ encrypts messages, $DEC_k$ decrypts messages.
3. $(\forall k \in \mathcal{K})(\forall m \in \mathcal{M}), DEC_k(ENC_k(m)) = m$
Keyed functions

1. Let $F : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ be an efficient, deterministic algorithm

2. Define $F_k(x) = F(k, x)$

3. The first input is called the key

4. Choosing a uniform $k \in \{0, 1\}^n$ is equivalent to choosing the function $F_k : \{0, 1\}^n \rightarrow \{0, 1\}^n$

Note: In literature and the textbook Keyed functions $k, x$ can be diff sizes, but we never do.
Keyed functions

1. Let \( F : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n \) be an efficient, deterministic algorithm

2. Define \( F_k(x) = F(k, x) \)

3. The first input is called the key

4. Choosing a uniform \( k \in \{0, 1\}^n \) is equivalent to choosing the function \( F_k : \{0, 1\}^n \rightarrow \{0, 1\}^n \)

**Note:** In literature and the textbook Keyed functions \( k, x \) can be diff sizes, but we never do. They are wrong, we are right.
Randomized Encryption

A Randomized Private-Key Encryption Scheme has message space $\mathcal{M}$, Key space $\mathcal{K} = \{0, 1\}^n$, algorithms (GEN, ENC, DEC).

1. GEN generates keys $k \in \mathcal{K}$ (Think: picking an $F_k$ rand.)
2. $ENC_k$: on input $m$ it picks a rand $r \in \{0, 1\}^n$ and outputs $(r, m \oplus F_k(r))$.
3. $DEC_k(r, c) = c \oplus F_k(r)$.

Note:

1. $ENC_k(m)$ is not a function- it can return many different pairs.
2. Easy to see that Encrypt-Decrypt works.
3. Rand Shift is not an example, but is the same spirit.
4. General definition that encompass’s Rand Shift: Can replace $\oplus$ with any invertible operation.
Pseudorandom functions
Pseudorandom functions

- Informally, a pseudorandom function “looks like” a random (i.e. uniform) function.
- Can define formally via a Game. We won’t. Might be HW or Exam Question.
Theorem: If $F_k$ is a PRF then the following encryption scheme is CPA-secure.

1. **GEN** generates keys $k \in \mathcal{K}$ (Think: picking an $F_k$ rand.)
2. $ENC_k$: on input $m$ it picks a rand $r \in \{0, 1\}^n$ and outputs $(r, m \oplus F_k(r))$.
3. $DEC_k(r, c) = c \oplus F_k(r)$.

Proof Sketch: If not CPA-secure then $F_k$ is not a PRF.