Chosen Plaintext Attacks (CPA)
CPA-Security

\[ c_1 \leftarrow \text{Enc}_k(m_1) \]
\[ c_2 \leftarrow \text{Enc}_k(m_2) \]
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. It will look stupid. We will discuss why its not.
3. Define CPA for computational security.
4. We will know that it is not stupid from prior discussion.
Perfect CPA-Security via a Game

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

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

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

Does Eve has a strategy that wins over half the time?
Π is secure against chosen-plaintext attacks (CPA-secure) if for all Eve.

$$\Pr[\text{Eve Wins}] \leq \frac{1}{2}$$
This Looks Stupid! Eve can always Win!

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 $b' = 1$.

Discuss: Is there any cipher we’ve seen that is CPA-secure?
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 $b' = 1$.

Discuss: Is there any cipher we've seen that is CPA-secure?

1-time pad? Eve still wins since use entire key to encode message.
This Looks Stupid! Eve can always Win!

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 $b' = 1$.

Discuss: Is there any cipher we’ve seen that is CPA-secure?

1-time pad? Eve still wins since use entire key to encode message.

Discuss: Any other examples?
This Looks Stupid! Eve can always Win!

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 $b' = 1$.

Discuss: Is there any cipher we’ve seen that is CPA-secure?

1-time pad? Eve still wins since use entire key to encode message.

Discuss: Any other examples?
Need to have $ENC_k(m)$ not always be the same.
This Looks Stupid! Eve can always Win!

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 $b' = 1$.

Discuss: Is there any cipher we’ve seen that is CPA-secure?

1-time pad? Eve still wins since use entire key to encode message.

Discuss: Any other examples?

Need to have $ENC_k(m)$ not always be the same.

Randomized Shift! PKCS-1.5 RSA! $ENC_k(m)$ not always the same.
This Looks Stupid! Eve can always Win!

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 $b' = 1$.  

Discuss: Is there any cipher we've seen that is CPA-secure?

1-time pad? Eve still wins since use entire key to encode message.  
Discuss: Any other examples?  
Need to have $ENC_k(m)$ not always be the same.

Randomized Shift! PKCS-1.5 RSA! $ENC_k(m)$ not always the same.  
Upshot: ALL deterministic schemes are CPA-insecure.
Is Randomized Shift Perfect CPA-secure?

Randomized shift: \( S = \{0, \ldots, 25\} \). Key is \( k \) which codes function \( f_k : S \rightarrow S \).

1. To send message \((m_1, \ldots, m_L)\) (each \(m_i\) is a character)
   1.1 Pick random \(r_1, \ldots, r_L \in S\). For \(1 \leq i \leq L\) compute \(s_i = f(r_i)\).
   1.2 Send \(((r_1; m_1 + s_1), \ldots, (r_L; m_L + s_L))\)

2. To decode message \(((r_1; c_1), \ldots, (r_L; c_L))\)
   2.1 For \(1 \leq i \leq L\) \(s_i = f(r_i)\).
   2.2 Find \((c_1 - s_1, \ldots, c_L - s_L)\)

\textbf{VOTE}: YES-CPA secure, NO-not CPA-secure.
Is Randomized Shift Perfect CPA-secure?

Randomized shift: $S = \{0, \ldots, 25\}$. Key is $k$ which codes function $f_k : S \rightarrow S$.

1. To send message $(m_1, \ldots, m_L)$ (each $m_i$ is a character)
   1.1 Pick random $r_1, \ldots, r_L \in S$. For $1 \leq i \leq L$ compute $s_i = f(r_i)$.
   1.2 Send $((r_1; m_1 + s_1), \ldots, (r_L; m_L + s_L))$

2. To decode message $((r_1; c_1), \ldots, (r_L; c_L))$
   2.1 For $1 \leq i \leq L$ $s_i = f(r_i)$.
   2.2 Find $(c_1 - s_1, \ldots, c_L - s_L)$

VOTE: YES-CPA secure, NO-not CPA-secure. NO
Recall that Eve has Black Box for $ENC_k$.

1. Eve picks $a$ and $b$

2. Eve sees $c = (r; x)$. Knows that Shift is $s = f_k(r)$ but does not know what this is. Eve wants to know if $c$ codes $a$ or $b$.

3. Eve gets to use $ENC_k$. So Eve computes $ENC_k(a)$ many times hoping to see $(r; −)$. If second coord is $x$ then $m = a$, else $m = b$.

How many calls to $ENC_k$ does Eve expect to make?
How Many Calls?

$ENC_k$ randomly picks elts of $S$. 

Clean Math Question: Let $r \in S = \{1, \ldots, y\}$. Pick elements from $S$ unif. What is expected number of picks until you see $r$? One can show its $y$. 
How Many Calls?

$ENC_k$ randomly picks elts of $S$.

We keep running $ENC_k(a)$ hoping to get $r$ in first coordinate.
How Many Calls?

$ENC_k$ randomly picks elts of $S$.

We keep running $ENC_k(a)$ hoping to get $r$ in first coordinate.

Let $|S| = y$ (26).

**Clean Math Question:** Let $r \in S = \{1, \ldots, y\}$. Pick elements from $S$ unif. What is expected number of picks until you see $r$? One can show its $y$. 
Take Away

Eve won by calling $ENC_k(a)$ several times. Expected times is $26 = |S|$.
Eve won by calling $ENC_k(a)$ several times. Expected times is $26 = |S|$.

Lets say that The Vorlons have alphabet of $10^{10}$ characters.

1. Can Eve always win? Discuss
Eve won by calling $\text{ENC}_k(a)$ several times. Expected times is $26 = |S|$.

Let's say that The Vorlons have alphabet of $10^{10}$ characters.

1. Can Eve always win? Discuss. If allow Eve to play forever than prob that she will see $(r; -)$ is 1. Not clear this makes sense. If insist Eve play for some preset finite amount of time then small prob of losing.

2. Can Eve win with Prob $> \frac{1}{2}$? Discuss Yes. If Eve picks $10^{10^{10}}$ looking for $(r; -)$ will find it with prob close to 1.
Eve won by calling $ENC_k(a)$ several times. Expected times is $26 = |S|$.

Let's say that The Vorlons have alphabet of $10^{10}$ characters.

1. Can Eve always win? Discuss Depends. If allow Eve to play forever than prob that she will see $(r; -)$ is 1. Not clear this makes sense. If insist Eve play for some preset finite amount of time then small prob of losing.

2. Can Eve win with Prob $> \frac{1}{2}$? Discuss Yes. If Eve picks $10^{10^{10}}$ looking for $(r; -)$ will find it with prob close to 1.

YES Recall that this is Perfect CPA-security so we allow Eve unlimited time.
Take Away

Eve won by calling $\text{ENC}_k(a)$ several times. Expected times is $26 = |S|$.

Lets say that The Vorlons have alphabet of $10^{10}$ characters.

1. Can Eve always win? Discuss Depends. If allow Eve to play forever than prob that she will see $(r;−)$ is 1. Not clear this makes sense. If insist Eve play for some preset finite amount of time then small prob of losing.

2. Can Eve win with Prob $> \frac{1}{2}$? Discuss Yes. If Eve picks $10^{10^{10}}$ looking for $(r;−)$ will find it with prob close to 1.

YES Recall that this is Perfect CPA-security so we allow Eve unlimited time.

We will now define the more realistic Computational CPA-security.
Comp CPA-Security

\(\Pi = (\text{GEN, ENC, DEC})\) 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

- Π is **CPA Secure** if for all 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 (\( \text{GEN, ENC, DEC} \)):

1. \( \text{GEN} \) generates keys \( k \in \mathcal{K} \).
2. \( \text{ENC}_k \) encrypts messages, \( \text{DEC}_k \) decrypts messages.
3. \((\forall k \in \mathcal{K})(\forall m \in \mathcal{M}), \text{DEC}_k(\text{ENC}_k(m)) = m\)
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. $\text{ENC}_k$: on input $m$ it picks a rand $r \in \{0, 1\}^n$ and outputs $(r, m \oplus F_k(r))$.
3. $\text{DEC}_k(r, c) = c \oplus F_k(r)$.

Note:

1. $\text{ENC}_k(m)$ 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
Informally, a pseudorandom function “looks like” a random (i.e. uniform) function
Definition of PRFs

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

Denote \( F(k, x) \) by \( F_k(x) \) and think of \( k \) as being key.

**Game:** Alice and Eve are the players. Both have access to \( F \).

1. Alice picks \( k \in \{0, 1\}^n \) unif. (Think: picked \( F_k \))
2. Alice picks \( f : \{0, 1\}^n \rightarrow \{0, 1\}^n \) unif.
3. Alice gives \( F_k, f \) to Eve (black box access).
4. Eve tries to determine which one is \( F_k \) and which one is not.
5. Eve says which is which. If she is right she wins!

Can Eve win this game with probability over \( \frac{1}{2} \)?
Informal Definition Continued

$G$ is a PRG if for all PPT Eves, there is a neg function $\epsilon(n)$ such that

$$\Pr[\text{Eve Wins}] \leq \frac{1}{2} + \epsilon(n)$$
Constructing a CPA-Secure Encryption

**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.