\[ x \oplus x = 0 \]
$0 \oplus x = x$
Fix $x$ in $\{0, 1\}^n$. If I pick $y$ uniformly from $\{0, 1\}^n$, what is the distribution of $x \oplus y$?
Let $z = x \oplus y$

If $x$ is fixed and $y$ takes on each value in $\{0, 1\}^n$, how many of each $z$ are there?
• Defined random functions, PRFs, and PRPs
• Defined what the prf-advantage is, and what the prp-advantage under both cca and cpa are
• Said that prp-cca implies prp-cpa
Encryption

- Want some way to send messages securely between people who share a key
- We’re going to build it out of a PRF or PRP
- First we need to define it
Definition

- A *key generation* algorithm $K$ that returns a random $k$
- An *encryption* algorithm $E$ that takes a key and a plaintext and returns a ciphertext
- A *decryption* algorithm $D$ that takes a key and a ciphertext and returns a plaintext
For all keys $k$ and plaintexts $m$,

$$D_k(E_k(m)) = m$$
What would make this secure?
Things that would make it insecure

- Being able to recover the key $k$
Things that would make it insecure

• Being able to recover the key $k$

• Being able to recover the message, even if you can’t find the key
Things that would make it insecure

- Being able to recover the key $k$
- Being able to recover the message, even if you can’t find the key
- Being able to recover part of the message or some function of the message
Chosen Plaintext Attack

• We pick a bit $b$ and a key $k$ (either 0 or 1)
• The attacker gives us two messages, $M_0$ and $M_1$
• We give the attacker $E_k(M_b)$
• The attacker guesses a bit $b'$ and wins if $b' = b$
Chosen Plaintext Attack

• We pick a bit $b$ and a key $k$ (either 0 or 1)

• The attacker gives us two messages, $M_0$ and $M_1$

• We give the attacker $E_k(M_b)$

• The attacker guesses a bit $b'$ and wins if $b' = b$
More Formally

• The bit $b$ decides what world the adversary is in

• The adversary’s advantage is:

\[ \Pr[\text{A says 1 in world 1}] - \Pr[\text{A says 1 in world 0}] \]
Example 1: ECB

For some permutation \( g \) which is a function of the key:

\( E(m): \)
\[
(m_1, m_2, ..., m_n) = m
\]
return \( (g(m_1), g(m_2), ..., g(m_n)) \)

\( D(c): \)
\[
(c_1, c_2, ..., c_n) = c
\]
return \( (g^{-1}(c_1), g^{-1}(c_2), ..., g^{-1}(c_n)) \)
ECB is not secure

Adversary A(O):

\[ M_0 = 0 \]
\[ M_1 = 1 \]

\[ x = \text{O}(M_0, M_1) \]
\[ \{M_0\} = 0 \]
\[ \{M_1\} = 2 \]

\[ y = \text{O}(M_0, M_1) \]

if \( (x == y) \)
    return 0
else
    return 1
ECB is not secure

Adversary $A(O)$:

<table>
<thead>
<tr>
<th>World 0</th>
<th>World 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_0 = 0$</td>
<td>$g(0)$</td>
</tr>
<tr>
<td>$M_1 = 1$</td>
<td>$g(1)$</td>
</tr>
</tbody>
</table>

$x = O(M_0, M_1)$

<table>
<thead>
<tr>
<th>World 0</th>
<th>World 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_0 = 0$</td>
<td>$g(0)$</td>
</tr>
<tr>
<td>$M_1 = 2$</td>
<td>$g(2)$</td>
</tr>
</tbody>
</table>

$y = O(M_0, M_1)$

if ($x == y$)
    return 0
else
    return 1
ECB is not secure

Adversary $A(O)$:

- $M_0 = 0$
- $M_1 = 1$
- $x = O(M_0, M_1)$
- $M_0 = 0$
- $M_1 = 2$
- $y = O(M_0, M_1)$

if $(x \neq y)$
  
  return 1
else
  
  return 0

Advantage is 1 no matter what $g$ is
No Deterministic Encryption Algorithms

- Need either:
  - A randomized algorithm
  - A deterministic algorithm that can store state between invocations
Example 2: CTR

For some function $g$ which is a function of the key, and counter $\text{ctr}$ which is initially 0:

\[
E(m): \quad (m_1, m_2, ..., m_n) = m
\]

\[
\text{for } i = 1 \text{ to } n
\]

\[
c_i = g(\text{ctr}) \oplus m_i
\]

\[
\text{ctr}++
\]

\[
\text{return } (\text{ctr}, c_1, c_2, ..., c_n)
\]
Example 2: CTR

For some function $g$ which is a function of the key, and counter $ctr$ which is initially 0:

$$D(c):$$

$$(x, c_1, c_2, ..., c_n) = c$$

for $i = 1$ to $n$

$$m_i = g(x) \oplus c_i$$

$x++$

return $(m_1, m_2, ..., m_n)$
Example 2: CTR

For some function $g$ which is a function of the key, and counter $ctr$ which is initially 0:

$$D(c):$$
$$(x, c_1, c_2, \ldots, c_n) = c$$

for $i = 1$ to $n$

$$m_i = g(x) \oplus c_i$$

$x++$

return $(m_1, m_2, \ldots, m_n)$
CTR Mode is Secure

- We want to show that an adversary's advantage in winning the $M_0$ or $M_1$ game is no better than another adversary's advantage at telling whether $g$ is a PRF.
Proof Sketch

- First think of CTR mode where \( g \) is a random function
- The advantage of an adversary in the \( M_0 \) or \( M_1 \) game is 0 because \( \oplus \) preserves randomness
Proof Sketch

- Given an adversary $A$ that plays the $M_0$ or $M_1$ game, we’ll construct an adversary $B$ that wins at deciding whether a given function $g$ is random or a PRF.
Proof Sketch

Adversary B(g):
Choose bit b at random
Run adversary A(O) where O is:
  return CTR mode encryption with g of $M_b$
A will return a value b'
  If b' == b
    return 1
  Else
    return 0
Proof Sketch

Adversary $B(g)$:
Choose bit $b$ at random
Run adversary $A(O)$ where $O$ is:
    return CTR mode encryption with $g$ of $M_b$

A will return a value $b'$
If $b' == b$
    return 1
Else
    return 0

If $g$ is a random function, we expect $A$ to guess wrong most of the time
Advantage of B

• Just the advantage of A when g is a PRF minus advantage of A when g is a random function

• We already said that the advantage of A when g is a random function is 0

• Advantage of B at determining whether g is a PRF is the same as the advantage of A at winning the $M_0$ or $M_1$ game
Conclusion

- ECB mode encryption is not secure even if you build it using a random function
- CTR mode encryption is secure if you build it out of a PRF that's secure