A symmetric encryption scheme $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ consists of three algorithms:

- $\mathcal{K}$ is randomized
- $\mathcal{E}$ can be randomized or stateful
- $\mathcal{D}$ is deterministic

Correct decryption requirement

Formally: For all $K$ and $M$ we have

$$\Pr[\mathcal{D}_K(\mathcal{E}_K(M)) = M] = 1,$$

where the probability is over the coins of $\mathcal{E}$

Example: OTP

$\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ where:

- $\mathcal{K}$
  
  $K \leftarrow \{0, 1\}^k$
  
  return $K$

- $\mathcal{E}_K(M)$
  
  $C \leftarrow K \oplus M$
  
  return $C$

- $\mathcal{D}_K(C)$
  
  $M \leftarrow K \oplus C$
  
  return $M$

Correct decryption:

$$\mathcal{D}_K(\mathcal{E}_K(M)) = \mathcal{D}_K(K \oplus M) = K \oplus (K \oplus M) = M$$
Block cipher modes of operation

\[ E : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n \] is a block cipher

**Notation:** \( x[i] \) is the \( i \)-th \( n \)-bit block of a string \( x \), so that \( x = x[1] \ldots x[m] \) if \( |x| = nm \).

Always:

\[
\begin{align*}
\text{Alg } E & \\
K & \overset{\$}{\leftarrow} \{0, 1\}^k \\
\text{return } K
\end{align*}
\]

which enables them to encrypt a 1-block message.

How do we encrypt a long message using a primitive that only applies to \( n \)-bit blocks?

**ECB: Electronic Codebook Mode**

\[ SE = (K, E, D) \] where:

\[
\begin{align*}
\text{alg } E_K(M) & \\
\text{for } i = 1, \ldots, m & \\
C[i] & \leftarrow E_K(M[i]) \\
\text{return } C
\end{align*}
\]

\[
\begin{align*}
\text{alg } D_K(C) & \\
\text{for } i = 1, \ldots, m & \\
M[i] & \leftarrow E_K^{-1}(C[i]) \\
\text{return } M
\end{align*}
\]

Correct decryption relies on \( E \) being a block cipher, so that \( E_K \) is invertible

**Evaluating Security**

Sender encrypts some messages \( M_1, \ldots, M_q \), namely

\[
C_1 \leftarrow E_K(M_1), \ldots, C_q \leftarrow E_K(M_q)
\]

and transmits \( C_1, \ldots, C_q \) to receiver.

**Adversary**

- Knows \( SE = (K, E, D) \)
- Knows \( C_1, \ldots, C_q \)
- Is **not** given \( K \!\)

Possible adversary goals:

- Recover \( K \)
- Recover \( M_1 \)

But we will need to look beyond these
Security of ECB

### Adversary task

<table>
<thead>
<tr>
<th>Adversary task</th>
<th>Assessment</th>
<th>Why?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute K</td>
<td>seems hard</td>
<td>E is secure</td>
</tr>
<tr>
<td>Compute M[1]</td>
<td>seems hard</td>
<td>E is secure</td>
</tr>
</tbody>
</table>

Adversary has ciphertext \( C = C[1] \cdots C[m] \)

### Why is the above true? Because \( E_K \) is deterministic:

\[
\begin{align*}
M[1] & \quad M[2] & \cdots & \quad M[m] \\
E_K & \quad E_K & \cdots & \quad E_K \\
C[1] & \quad C[2] & \cdots & \quad C[m]
\end{align*}
\]

### Why does this matter?

### Suppose we know that there are only two possible messages, \( Y = 1^n \) and \( N = 0^n \), for example representing
- FIRE or DON'T FIRE a missile
- BUY or SELL a stock
- Vote YES or NO

Then ECB algorithm will be \( E_K(M) = E_K(M) \).

### Votes \( M_1, M_2 \in \{ Y, N \} \) are ECB encrypted and adversary sees ciphertexts \( C_1 = E_K(M_1) \) and \( C_2 = E_K(M_2) \):

\[
\begin{align*}
M_1[1] & \quad M_1[m] & \quad M_2[1] & \quad M_2[m] \\
E_K & \cdots & E_K & \cdots \\
C_1[1] & \quad C_1[m] & \quad C_2[1] & \quad C_2[m]
\end{align*}
\]

Adversary may have cast the first vote and thus knows \( M_1 \); say \( M_1 = Y \). Then adversary can figure out \( M_2 \):
- If \( C_2 = C_1 \) then \( M_2 \) must be \( Y \)
- Else \( M_2 \) must be \( N \)
Is this avoidable?

Let $\mathcal{E} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be \textbf{ANY} encryption scheme.

Suppose $M_1, M_2 \in \{Y, N\}$ and
- Sender sends ciphertexts $C_1 \leftarrow \mathcal{E}_K(M_1)$ and $C_2 \leftarrow \mathcal{E}_K(M_2)$
- Adversary $A$ knows that $M_1 = Y$

Adversary says: If $C_2 = C_1$ then $M_2$ must be $Y$ else it must be $N$.

Does this attack work?

Yes, if $\mathcal{E}$ is deterministic.

Randomized encryption

For encryption to be secure it must be randomized.
That is, algorithm $\mathcal{E}_K$ flips coins.

If the same message is encrypted twice, we are likely to get back different answers. That is, if $M_1 = M_2$ and we let

$$C_1 \leftarrow \mathcal{E}_K(M_1) \quad \text{and} \quad C_2 \leftarrow \mathcal{E}_K(M_2)$$

then

$$\Pr[C_1 = C_2]$$

will (should) be small, where the probability is over the coins of $\mathcal{E}$.

Randomized encryption

There are many possible ciphertexts corresponding to each message.

If so, how can we decrypt?

We will see examples soon.

Randomized encryption

A fundamental departure from classical and conventional notions of encryption.

Classically, encryption (e.g., substitution cipher) is a code, associating to each message a unique ciphertext.

Now, we are saying no such code is secure, and we look to encryption mechanisms which associate to each message a number of different possible ciphertexts.
An alternative to randomization is to allow the encryption algorithm to maintain state. This might be a counter:
- encrypt depending on counter value
- then update counter

We will see schemes that use this paradigm to get around the security weaknesses of deterministic encryption without using randomness.

---

**Stateful encryption**

**More Modes of Operation**

<table>
<thead>
<tr>
<th>Randomized</th>
<th>Stateful</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBC$, CTR$</td>
<td>CBCC, CTRC</td>
</tr>
</tbody>
</table>

**CBC$: Cipher Block Chaining with random IV mode**

\[ \mathcal{E} = (K, \mathcal{E}, \mathcal{D}) \] where:

\[
\begin{align*}
\text{Alg } \mathcal{E}_K(M) &= \quad \text{Alg } \mathcal{D}_K(C) \\
C[0] &\trianglerighteq \{0,1\}^n \\
\text{for } i = 1, \ldots, m \text{ do} & \quad \text{for } i = 1, \ldots, m \text{ do} \\
C[i] &\leftarrow E_K(M[i] \oplus C[i-1]) \\
M[i] &\leftarrow E_K^{-1}(C[i]) \oplus C[i-1] \\
\text{return } C & \quad \text{return } M
\end{align*}
\]

Correct decryption relies on \( E \) being a block cipher so that \( E_K \) is invertible.

---

**CTRC mode**

Sender maintains a counter \( ctr \) that is initially 0 and is updated by \( \mathcal{E} \)

\( \langle j \rangle \) = the \( n \)-bit binary representation of integer \( j \) (\( 0 \leq j < 2^n \))

\[
\begin{align*}
\text{Alg } \mathcal{E}_K(M) &= \quad \text{Alg } \mathcal{D}_K(C) \\
C[0] &\leftarrow ctr \\
\text{for } i = 1, \ldots, m \text{ do} & \quad \text{for } i = 1, \ldots, m \text{ do} \\
P[i] &\leftarrow E_K(\langle ctr + i \rangle) \\
C[i] &\leftarrow P[i] \oplus M[i] \\
ctr &\leftarrow ctr + m \\
\text{return } C & \quad \text{return } M
\end{align*}
\]

- Decryptor does not maintain a counter
- \( \mathcal{D} \) does not use \( E^{-1} \)
Security of CBC$ against key recovery

If adversary has a plaintext $M$ and corresponding ciphertext $C \leftarrow E_K(M)$ then it has input-output examples

$(M[1] \oplus C[0], C[1]), (M[2] \oplus C[1], C[2])$ of $E_K$.

So chosen-message key recovery attacks on $E$ can be mounted to recover $K$.

Conclusion: Security of CBC$ against key recovery is no better than that of the underlying block cipher.

Voting with CBC$

Suppose we encrypt $M_1, M_2 \in \{Y, N\}$ with CBC$.$

If $M_1 = Y$ we have

$C_1[0] \oplus Y$

$E_K$

$C_1[1]$

$A$ knows $C_1[0]C_1[1]$ and $C_2[0]C_2[1]$. Now

- If $C_1[0] = C_2[0]$ then $A$ can deduce that
  - If $C_2[1] = C_1[1]$ then $M_2 = Y$
  - If $C_2[1] \neq C_1[1]$ then $M_2 = N$
- But the probability that $C_1[0] = C_2[0]$ is very small.

Assessing security

So CBC$ is better than ECB. But is it secure?

CBC$ is the world’s most widely used encryption scheme (SSL, SSH, TLS, ...) so knowing whether it is secure is important

To answer this we first need to decide and formalize what we mean by secure.
Types of encryption schemes

Special purpose: Used in a specific setting, to encrypt data of some known format or distribution. Comes with a

WARNING! only use under conditions X.

General purpose: Used to encrypt in many different settings, where the data format and distribution are not known in advance.

We want general purpose schemes because

- They can be standardized and broadly used.
- Once a scheme is out there, it gets used for everything anyway.
- General purpose schemes are easier to use and less subject to mis-use: it is hard for application designers to know whether condition X is met.

Security requirements

A priori information: What the adversary already knows about the data from the context. For example, it is drawn from \( \{Y, N\} \)

Data distribution or format: The data may be English or not; may have randomness or not; ...

Security should not rely on assumptions about these things.

E-mail encryption

E-mail data could be

- English text
- A pdf or executable file
- Votes

Want security in all these cases.

Security requirements

Suppose sender computes

\[ C_1 \leftarrow E_K(M_1); \ldots; C_q \leftarrow E_K(M_q) \]

Adversary A has \( C_1, \ldots, C_q \)

<table>
<thead>
<tr>
<th>What if A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieves K</td>
</tr>
<tr>
<td>Retrieves ( M_1 )</td>
</tr>
</tbody>
</table>

But also ...
Security requirements

We want to hide all partial information about the data stream.

Examples of partial information:
- Does $M_1 = M_2$?
- What is first bit of $M_1$?
- What is XOR of first bits of $M_1, M_2$?

Something we won’t hide: the length of the message

What we seek

We want a single “master” property MP of an encryption scheme such that
- MP can be easily specified
- We can evaluate whether a scheme meets it
- MP implies ALL the security conditions we want: it guarantees that a ciphertext reveals NO partial information about the plaintext.

Thus a scheme having MP means not only that if adversary has $C_1 \leftarrow E_K(M_1)$ and $C_2 \leftarrow E_K(M_2)$ then
- It can’t get $M_1$
- It can’t get 1st bit of $M_1$
- It can’t get XOR 1st bits of $M_1, M_2$

but in fact implies “all” such information about $M_1, M_2$ is protected.

Seeking MP

So what is the master property MP?

It is a notion we call indistinguishability (IND). We will define
- IND-CPA: Indistinguishability under chosen-plaintext attack
- IND-CCA: Indistinguishability under chosen-ciphertext attack

Plan

- Define IND-CPA
- Examples of non-IND-CPA schemes
- See why IND-CPA is a “master” property, namely why it implies that ciphertexts leak no partial information about plaintexts
- Examples of IND-CPA schemes
- IND-CCA
Intuition for definition of IND

Consider encrypting one of two possible message streams, either

\[ M^0_1, \ldots, M^q_0 \]

or

\[ M^1_1, \ldots, M^q_1 \]

Adversary, given ciphertexts and both data streams, has to figure out which of the two streams was encrypted.

ind-cpa-adversaries

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

An ind-cpa adversary \( A \) has an oracle \( \mathcal{LR} \)

- It can make a query \( M^0_0, M^1_1 \) consisting of any two equal-length messages
- It can do this many times
- Each time it gets back a ciphertext
- It eventually outputs a bit

\[ d \leftarrow A \]

33 / 116

The games

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

**Game Left\( \mathcal{SE} \)**

procedure Initialize
\( K \leftarrow K \)

procedure LR\( (M^0_0, M^1_1) \)
Return \( C \leftarrow \mathcal{E}_K(M^0_0) \)

**Game Right\( \mathcal{SE} \)**

procedure Initialize
\( K \leftarrow K \)

procedure LR\( (M^0_0, M^1_1) \)
Return \( C \leftarrow \mathcal{E}_K(M^1_1) \)

Associated to \( \mathcal{SE}, A \) are the probabilities

\[ \Pr [ \text{Left}_{\mathcal{SE}} \Rightarrow 1 ] \quad \text{and} \quad \Pr [ \text{Right}_{\mathcal{SE}} \Rightarrow 1 ] \]

that \( A \) outputs 1 in each world. The (ind-cpa) advantage of \( A \) is

\[ \text{Adv}_{\mathcal{SE}}^{\text{ind-cpa}}(A) = \Pr [ \text{Right}_{\mathcal{SE}} \Rightarrow 1 ] - \Pr [ \text{Left}_{\mathcal{SE}} \Rightarrow 1 ] \]
Let $E : \{0, 1\}^k \times \{0, 1\}^{128} \rightarrow \{0, 1\}^{128}$ be a block cipher and let $\mathcal{SE} = (K, E, D)$ be defined by

\[
\begin{align*}
\text{Alg } & K \\
K & \overset{\$}{\leftarrow} \{0, 1\}^k \\
\text{return } & K
\end{align*}
\]

\[
\begin{align*}
\text{Alg } & E_K(M) \\
\text{return } & E_K(M)
\end{align*}
\]

\[
\begin{align*}
\text{Alg } & D_K(M) \\
\text{return } & E_K^{-1}(M)
\end{align*}
\]

This scheme encrypts only 1-block messages.

Succinctly: $E_K(M) = E_K^{\text{Alg}}(M)$

What happens

- $C_1 = \mathcal{E}_K(0^n) = E_K(0^n)$
- $C_2 = \mathcal{E}_K(1^n) = E_K(1^n) \neq E_K(0^n)$
- so $C_1 \neq C_2$ and $A$ returns 0

so

$\Pr[\text{Left}^A_{\mathcal{SE}} \Rightarrow 1] = 0$

Let $\mathcal{E}_K(M) = E_K(M)$

adversary $A$

$C_1 \leftarrow LR(0^n, 0^n)$; $C_2 \leftarrow LR(1^n, 0^n)$

if $C_1 = C_2$ then return 1 else return 0

$\Pr[\text{Left}^A_{\mathcal{SE}} \Rightarrow 1] = \frac{38}{116}$

adversary $A$

$C_1 \leftarrow LR(0^n, 0^n)$; $C_2 \leftarrow LR(1^n, 0^n)$

if $C_1 = C_2$ then return 1 else return 0

$\Pr[\text{Right}^A_{\mathcal{SE}} \Rightarrow 1] = \frac{40}{116}$

Let $\mathcal{E}_K(M) = E_K(M)$

adversary $A$

$C_1 \leftarrow LR(0^n, 0^n)$; $C_2 \leftarrow LR(1^n, 0^n)$

if $C_1 = C_2$ then return 1 else return 0

$\Pr[\text{Left}^A_{\mathcal{SE}} \Rightarrow 1] = \frac{38}{116}$

adversary $A$

$C_1 \leftarrow LR(0^n, 0^n)$; $C_2 \leftarrow LR(1^n, 0^n)$

if $C_1 = C_2$ then return 1 else return 0

$\Pr[\text{Right}^A_{\mathcal{SE}} \Rightarrow 1] = \frac{40}{116}$
Let $E_K(M) = E_K(M)$

**IND-CPA security**

Adversary advantage depends on its
- strategy
- resources: Running time $t$ and number $q$ of oracle queries

**Security:** $S\mathcal{E}$ is **IND-CPA** (i.e. secure)
if $\text{Adv}_{S\mathcal{E}}^{\text{ind-cpa}}(A)$ is “small” for ALL $A$ that use “practical” amounts of resources.

**Example:** 80-bit security could mean that for all $n = 1, \ldots, 80$ we have

$$\text{Adv}_{S\mathcal{E}}^{\text{ind-cpa}}(A) \leq 2^{-n}$$

for any $A$ with time and number of oracle queries at most $2^{80-n}$.

**Insecurity:** $S\mathcal{E}$ is **not** IND-CPA (i.e. insecure) if there exists $A$ using “few” resources that achieves “high” advantage.

Let $S\mathcal{E} = (K, E, D)$ be an encryption scheme and $A$ be an ind-cpa adversary. Then

$$\text{Adv}_{S\mathcal{E}}^{\text{ind-cpa}}(A) = \Pr[\text{Right}_A^{A \Rightarrow 1}] - \Pr[\text{Left}_A^{A \Rightarrow 1}]$$

is a number between $-1$ and $1$.

A “large” (close to 1) advantage means
- $A$ is doing well
- $S\mathcal{E}$ is not secure

A “small” (close to 0 or $\leq 0$) advantage means
- $A$ is doing poorly
- $S\mathcal{E}$ resists the attack $A$ is mounting

Let $E : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a block cipher. Recall that ECB mode defines symmetric encryption scheme $S\mathcal{E} = (K, E, D)$ with

$$E_K(M) = E_K(M[1])E_K(M[2]) \cdots E_K(M[m])$$

**ECB is not IND-CPA-secure**
ECB is not IND-CPA-secure

Let $\mathcal{E}_K(M) = E_K(M[1]) \cdots E_K(M[m])$

<table>
<thead>
<tr>
<th>Left world $\text{LR} \xleftarrow{\nu} \mathcal{E}_K(M_0)$</th>
<th>Right world $\text{LR} \xleftarrow{\nu} \mathcal{E}_K(M_1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_0, M_1$</td>
<td>$M_0, M_1$</td>
</tr>
<tr>
<td>$C$</td>
<td>$C$</td>
</tr>
<tr>
<td>$A$</td>
<td>$A$</td>
</tr>
</tbody>
</table>

Can we design $A$ so that

$$\text{Adv}_\text{ind-cpa}^{\mathcal{E}}(A) = \Pr[\text{Right}_A \Rightarrow 1] - \Pr[\text{Left}_A \Rightarrow 1]$$

is close to 1?

Exploitable weakness of $\mathcal{E}$: $M_1 = M_2$ implies $E_K(M_1) = E_K(M_2)$.

### ECB is not IND-CPA-secure: Right world analysis

$\mathcal{E}$ is defined by $\mathcal{E}_K(M) = E_K(M[1]) \cdots E_K(M[m])$.

**adversary $A$**

$C_1 \leftarrow \text{LR}(0^n, 0^n); C_2 \leftarrow \text{LR}(1^n, 0^n)$

if $C_1 = C_2$ then return 1 else return 0

Game $\text{Right}_{\mathcal{E}}$

**procedure Initialize**

$K \xleftarrow{\nu} K$

**procedure $LR(M_0, M_1)$**

Return $\mathcal{E}_K(M_1)$

Then

$$\Pr[\text{Right}_A \Rightarrow 1] = 1$$

because $C_1 = E_K(0^n) = E_K(0^n) = C_2$.

### ECB is not IND-CPA-secure: Left world analysis

$\mathcal{E}$ is defined by $\mathcal{E}_K(M) = E_K(M[1]) \cdots E_K(M[m])$.

**adversary $A$**

$C_1 \leftarrow \text{LR}(0^n, 0^n); C_2 \leftarrow \text{LR}(1^n, 0^n)$

if $C_1 = C_2$ then return 1 else return 0

Game $\text{Left}_{\mathcal{E}}$

**procedure Initialize**

$K \xleftarrow{\nu} K$

**procedure $LR(M_0, M_1)$**

Return $\mathcal{E}_K(M_0)$

Then

$$\Pr[\text{Left}_A \Rightarrow 1] = 0$$

because $C_1 = E_K(0^n) \neq E_K(1^n) = C_2$. 
ECB is not IND-CP A secure

adversary $A$

$C_1 \leftarrow LR(0^n, 0^n) \colon C_2 \leftarrow LR(1^n, 0^n)$

if $C_1 = C_2$ then return 1 else return 0

$Adv_{\text{ind-cpa}}(A) = Pr[\text{Right}_{SE} = 1] - Pr[\text{Right}_{SE} = 1]$

$$= 1$$

And $A$ is very efficient, making only two queries.

Thus ECB is not IND-CPA secure.

Why is IND-CPA the “master” property?

We claim that if encryption scheme $SE = (K, E, D)$ is IND-CPA secure then the ciphertext hides ALL partial information about the plaintext.

For example, from $C_1 \leftarrow E_K(M_1)$ and $C_2 \leftarrow E_K(M_2)$ the adversary cannot

• get $M_1$
• get 1st bit of $M_1$
• get XOR of the 1st bits of $M_1, M_2$
• etc.

Why is this true?

XOR-insecurity implies IND-CPA-insecurity

Let $\text{lsb}(M)$ denote the last bit of $M$

Suppose we are given an adversary $B$ such that

$E_K(M_1) \leftarrow C_1 \rightarrow B \rightarrow \text{lsb}(M_1) \oplus \text{lsb}(M_2)$

$E_K(M_2) \leftarrow C_2 \rightarrow B$ for all $M_1, M_2$. Then we claim we can design an ind-cpa adversary $A$ such that

$Adv_{\text{ind-cpa}}(A) = 1$,

meaning $SE$ is not IND-CPA secure.

Thus:

XOR-insecurity $\Rightarrow$ IND-CPA-insecurity

IND-CPA-security $\Rightarrow$ XOR-security

XOR-insecurity implies IND-CPA-insecurity

<table>
<thead>
<tr>
<th>A</th>
<th>$M_0, M_1$</th>
<th>$C$</th>
<th>LR</th>
<th>$C \leftarrow E_K(M_0)$</th>
<th>A</th>
<th>$M_0, M_1$</th>
<th>$C$</th>
<th>LR</th>
<th>$C \leftarrow E_K(M_1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Left world</td>
<td>Right world</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

adversary $A$

• Makes two LR queries
• The left messages are $M_0^1 = 0^n$ and $M_0^2 = 0^n$.
  Why? Because $\text{lsb}(0^n) \oplus \text{lsb}(0^n) = 0$
• The right messages are $M_1^1 = 0^n$ and $M_1^2 = 1^n$.
  Why? Because $\text{lsb}(0^n) \oplus \text{lsb}(1^n) = 1$
• Gets back 2 ciphertexts $C_1, C_2$
• Runs $B(C_1, C_2)$ to compute $\text{lsb}(M_0^1) \oplus \text{lsb}(M_0^2)$ which equals $b$, indicating whether Left or Right world

adversary $A$

$C_1 \leftarrow LR(0^n, 0^n) \colon C_2 \leftarrow LR(0^n, 1^n)$

d $\leftarrow B(C_1, C_2)$; return $d$
XOR-insecurity implies IND-CPA-insecurity

Left world

\[
\begin{array}{c|c}
\text{adversary } A & C \\
\hline
M_0, M_1 & C \leftarrow E_K(M_0)
\end{array}
\]

\[C_1 \leftarrow LR(0^n, 0^n); \ C_2 \leftarrow LR(0^n, 1^n)\]
\[d \leftarrow B(C_1, C_2)\; \text{; return } d\]

What happens:

- \(C_1 \leftarrow E_K(0^n)\) and \(C_2 \leftarrow E_K(0^n)\)
- The first bits of the encrypted messages XOR to 0
- so \(B\) returns 0

so \(\Pr[Left^A_{SE} \Rightarrow 1] = 0\)

Right world

\[
\begin{array}{c|c}
\text{adversary } A & C \\
\hline
M_0, M_1 & C \leftarrow E_K(M_1)
\end{array}
\]

\[C_1 \leftarrow LR(0^n, 0^n); \ C_2 \leftarrow LR(0^n, 1^n)\]
\[d \leftarrow B(C_1, C_2)\; \text{; return } d\]

What happens:

- \(C_1 \leftarrow E_K(0^n)\) and \(C_2 \leftarrow E_K(1^n)\)
- The first bits of the encrypted messages XOR to 1
- so \(B\) returns 1

so \(\Pr[Right^A_{SE} \Rightarrow 1] = 1\)

XOR-insecurity implies IND-CPA-insecurity

So

\[
\text{Adv}_E^{\text{ind-CPA}}(A) = \Pr[Right^A_{SE} \Rightarrow 1] - \Pr[Left^A_{SE} \Rightarrow 1]
\]
\[= 1 - 0\]
\[= 1\]

as claimed

Alternative formulation of advantage

Let \(SE = (K, E, D)\) be a symmetric encryption scheme and \(A\) an adversary.

\[
\text{Guess}_{SE}^A
\]

\[
\text{procedure LR}(M_0, M_1) \quad \text{return } C \leftarrow E_K(M_b)
\]

\[
\text{procedure Initialize}
\]

\[
K \leftarrow \mathcal{K}; \ b \leftarrow \{0,1\}
\]

\[
\text{return } (b = b')
\]

Proposition: \(\text{Adv}_E^{\text{ind-CPA}}(A) = 2 \cdot \Pr[\text{Guess}_{SE}^A \Rightarrow \text{true}] - 1\).

Proof: Observe

\[
\Pr[b' = 1 | b = 1] = \Pr[Right^A_{SE} \Rightarrow 1]
\]
\[
\Pr[b' = 1 | b = 0] = \Pr[Left^A_{SE} \Rightarrow 1]
\]
Proof (continued)

\[
\Pr[\text{Guess}_A \Rightarrow \text{true}] = \Pr[b = b'] \\
= \Pr[b = b' \mid b = 1] \cdot \Pr[b = 1] + \Pr[b = b' \mid b = 0] \cdot \Pr[b = 0] \\
= \Pr[b = b' \mid b = 1] \cdot \frac{1}{2} + \Pr[b = b' \mid b = 0] \cdot \frac{1}{2} \\
= \Pr[b' = 1 \mid b = 1] \cdot \frac{1}{2} + \Pr[b' = 0 \mid b = 0] \cdot \frac{1}{2} \\
= \Pr[b' = 1 \mid b = 1] \cdot \frac{1}{2} + (1 - \Pr[b' = 1 \mid b = 0]) \cdot \frac{1}{2} \\
= \frac{1}{2} + \frac{1}{2} \cdot (\Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0]) \\
= \frac{1}{2} + \frac{1}{2} \cdot (\Pr[\text{Right}_A \Rightarrow 1] - \Pr[\text{Left}_A \Rightarrow 1]) \\
= \frac{1}{2} + \frac{1}{2} \cdot \text{Adv}^{\text{IND-CPA}}_A(E) .
\]

Security of CTRC

Let \( E : \{0, 1\}^k \times \{0, 1\}^n \to \{0, 1\}^n \) be a block cipher. Sender maintains a counter \( ctr \), initially 0. The scheme is \( SE = (K, E, D) \) where

\[
\begin{align*}
\text{Alg } E_K(M) & \quad \text{C}[0] \leftarrow ctr \\
\text{for } i = 1, \ldots, m & \quad P[i] \leftarrow E_K(\langle ctr + i \rangle) \\
& \quad C[i] \leftarrow P[i] \oplus M[i] \\
\text{ctr} & \leftarrow \text{ctr} + m \\
\text{return } C
\end{align*}
\]

Question: Is \( SE \) IND-CPA secure?

We cannot expect so if \( E \) is “bad”. So, let’s ask:

Question: Assuming \( E \) is good (a PRF) is \( SE \) IND-CPA secure?

IND-CPA security of CTRC

\( SE = (K, E, D) \) CTRC mode relative to block cipher \( E \).

Question: If \( E \) is a PRF then is \( SE \) ind-cpa SECURE?

Answer: YES

And we can prove that the above answer is correct.

The above

• means CTRC has no “structural” weaknesses.
• Is not a triviality because it was not true for ECB.

Implications

Fact: If \( E \) is secure (PRF) then CTRC mode is a secure (IND-CPA) encryption scheme.

This means CTRC is a good, general purpose encryption scheme.

Ciphertexts leak NO partial information about messages.

Provides security regardless of message distribution. Votes can be securely encrypted.

We do not need to look for attacks on the scheme. We are guaranteed there are no attacks as long as \( E \) is secure.
Intuition for IND-CP A security of CTRC

Consider the CTRC scheme with $E_K$ replaced by a random function $F_n$.

$$\begin{align*}
\text{Alg } E_{F_n}(M) \\
C[0] &\leftarrow \text{ctr} \\
\text{for } i = 1, \ldots, m \text{ do} \\
P[i] &\leftarrow F_n(\langle \text{ctr} + i \rangle) \\
C[i] &\leftarrow P[i] \oplus M[i] \\
\text{ctr} &\leftarrow \text{ctr} + m \\
\text{return } C
\end{align*}$$

$\text{Alg } D_{F_n}(C)$

$$\begin{align*}
\text{ctr} &\leftarrow C[0] \\
\text{for } i = 1, \ldots, m \text{ do} \\
P[i] &\leftarrow F_n(\langle \text{ctr} + i \rangle) \\
M[i] &\leftarrow P[i] \oplus C[i] \\
\text{return } M
\end{align*}$$

Analyzing this is a thought experiment, but we can ask whether it is IND-CP A secure.

If so, the assumption that $E$ is a PRF says the real CTRC is IND-CP A secure.

CTRC with a random function

Since $F_n$ is random, the sequence $P[1] \cdot \ldots \cdot P[m]$ is random and the above is just one-time pad encryption, which is certainly IND-CP A secure.

So CTRC with a random function is IND-CP A secure.

IND-CP A security of CTRC

**Theorem:** Let $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a family of functions and let $\mathcal{SE} = (K, E, D)$ be the corresponding CTRC mode symmetric encryption scheme. Let $A$ be an ind-cpa adversary making at most $q$ LR queries totalling at most $\sigma$ blocks. Then there is a prf-adversary $B$ such that

$$\text{Adv}_{\mathcal{SE}}^{\text{ind-cpa}}(A) \leq 2 \cdot \text{Adv}_{E}^{\text{prf}}(B).$$

Furthermore $B$ makes at most $\sigma$ oracle queries and runs in time at most $t + \Theta(q + n\sigma)$.

**Implication:**

$E$ a PRF $\Rightarrow \text{Adv}_{E}^{\text{prf}}(B)$ small

$\Rightarrow \text{Adv}_{\mathcal{SE}}^{\text{ind-cpa}}(A)$ small

$\Rightarrow \mathcal{SE}$ IND-CP A secure

Proof by reduction

$A$’s world

$B$ runs $A$, itself replying to $A$’s oracle queries
Some notation

\[ \|M\|_n = \text{number of } n\text{-bit blocks in } M. \]

That is, \( M = M[1]...M[m] \) where \( m = \|M\|_n. \)

\( \langle j \rangle \) denotes the \( n\)-bit binary encoding of integer \( j \in \{0, ..., 2^n - 1\}. \)

Games for CTRC security proof

Claim 1: There is a prf-adversary \( B \) such that

\[
\Pr \left[ G_0^A \Rightarrow \text{true} \right] - \Pr \left[ G_1^A \Rightarrow \text{true} \right] \leq \text{Adv}_{E}^{\text{prf}}(B).
\]

adversary \( B \)

\( b \leftarrow \{0, 1\}; \ ctr \leftarrow 0; \)

\( b' \leftarrow A_{LR}^{\text{Adv}} \)

If \( (b = b') \) then return 1

Else return 0

subroutine LR(M_0, M_1)

\( C[0] \leftarrow ctr; m \leftarrow \|M_b\|_n \)

for \( i = 1, ..., m \) do

\( P[\langle ctr + i \rangle] \leftarrow E_K(\langle ctr + i \rangle) \)

\( C[i] \leftarrow P[\langle ctr + i \rangle] \oplus M_b[i] \)

\( ctr \leftarrow ctr + m \)

return \( C \)

Claim 1: There is a prf-adversary \( B \) such that

\[
\Pr \left[ G_0^A \Rightarrow \text{true} \right] - \Pr \left[ G_1^A \Rightarrow \text{true} \right] \leq \text{Adv}_{E}^{\text{prf}}(B).
\]

adversary \( B \)

\( b \leftarrow \{0, 1\}; \ ctr \leftarrow 0; \)

\( b' \leftarrow A_{LR}^{\text{Adv}} \)

If \( (b = b') \) then return 1

Else return 0

subroutine LR(M_0, M_1)

\( C[0] \leftarrow ctr; m \leftarrow \|M_b\|_n \)

for \( i = 1, ..., m \) do

\( P[\langle ctr + i \rangle] \leftarrow E_K(\langle ctr + i \rangle) \)

\( C[i] \leftarrow P[\langle ctr + i \rangle] \oplus M_b[i] \)

\( ctr \leftarrow ctr + m \)

return \( C \)

Thus

\[
\text{Adv}_{E}^{\text{prf}}(B) = \Pr \left[ \text{Real}_{E}^{B} \Rightarrow 1 \right] - \Pr \left[ \text{Rand}_{E}^{B} \Rightarrow 1 \right]
\]

\[
= \Pr \left[ G_0^A \Rightarrow \text{true} \right] - \Pr \left[ G_1^A \Rightarrow \text{true} \right]
\]

which proves Claim 1.

Analysis

Claim 1: There is a prf-adversary \( B \) such that

\[
\Pr \left[ G_0^A \Rightarrow \text{true} \right] - \Pr \left[ G_1^A \Rightarrow \text{true} \right] \leq \text{Adv}_{E}^{\text{prf}}(B).
\]

Procedure Initialize

\( K \leftarrow \{0, 1\}^k; \ b \leftarrow \{0, 1\} \)

\( ctr \leftarrow 0 \)

Procedure LR(M_0, M_1)

\( C[0] \leftarrow ctr; m \leftarrow \|M_b\|_n \)

for \( i = 1, ..., m \) do

\( P[\langle ctr + i \rangle] \leftarrow E_K(\langle ctr + i \rangle) \)

\( C[i] \leftarrow P[\langle ctr + i \rangle] \oplus M_b[i] \)

\( ctr \leftarrow ctr + m \)

return \( C \)

Procedure Finalize(b')

return \( (b = b') \)

Analysis

Claim 1: There is a prf-adversary \( B \) such that

\[
\Pr \left[ G_0^A \Rightarrow \text{true} \right] - \Pr \left[ G_1^A \Rightarrow \text{true} \right] \leq \text{Adv}_{E}^{\text{prf}}(B).
\]

Procedure Initialize

\( b \leftarrow \{0, 1\}; \ ctr \leftarrow 0 \)

Procedure LR(M_0, M_1)

\( C[0] \leftarrow ctr; m \leftarrow \|M_b\|_n \)

for \( i = 1, ..., m \) do

\( P[\langle ctr + i \rangle] \leftarrow E_K(\langle ctr + i \rangle) \)

\( C[i] \leftarrow P[\langle ctr + i \rangle] \oplus M_b[i] \)

\( ctr \leftarrow ctr + m \)

return \( C \)

Procedure Finalize(b')

return \( (b = b') \)
Theorem: Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a family of functions and let $\mathcal{SE} = (K, E, D)$ be the corresponding CTRC mode symmetric encryption scheme. Let $A$ be an ind-cpa adversary making at most $q$ $\text{LR}$ queries totalling at most $\sigma$ blocks. Then there is a prf-adversary $B$ such that

\[ \text{Adv}_E^{\text{ind-cpa}}(A) \leq 2 \cdot \text{Adv}_E^{\text{prf}}(B). \]

Furthermore $B$ makes at most $\sigma$ oracle queries and runs in time at most $t + \Theta(q + n\sigma)$.

**Proof of Claim 2 in CTRC analysis**

**Game $G_1$**

procedure $\text{Initialize}$

\[ b \leftarrow \{0,1\}; \text{ctr} \leftarrow 0 \]

procedure $\text{LR}(M_0, M_1)$

\[ C[0] \leftarrow \text{ctr}; m \leftarrow ||M_0||_o \]

for $i = 1, ..., m$

\[ P[(\text{ctr} + i)] \leftarrow \{0,1\}^n \]

\[ C[i] \leftarrow P[(\text{ctr} + i)] \oplus M_0[i] \]

\[ \text{ctr} \leftarrow \text{ctr} + m \]

return $C$

procedure $\text{Finalize}(b')$

return $(b = b')$

**Claim 2:** $\Pr[G_1^A \Rightarrow \text{true}] = \frac{1}{2}$

**Proof:** $\text{LR}$ in $G_2$ does not use bit $b$ so

\[ \Pr[G_1^A \Rightarrow \text{true}] = \Pr[G_2^A \Rightarrow \text{true}] = \frac{1}{2}. \]

**Game $G_2$**

procedure $\text{Initialize}$

\[ b \leftarrow \{0,1\}; \text{ctr} \leftarrow 0 \]

procedure $\text{LR}(M_0, M_1)$

\[ C[0] \leftarrow \text{ctr}; m \leftarrow ||M_0||_o \]

for $i = 1, ..., m$

\[ C[i] \leftarrow \{0,1\}^n \]

\[ \text{ctr} \leftarrow \text{ctr} + m \]

return $C$

procedure $\text{Finalize}(b')$

return $(b = b')$

**Birthday attack on CBC$^2$**

Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher. Let $\mathcal{SE} = (K, E, D)$ be the CBC$^2$ mode.

Suppose we are encrypting 1 block messages $M, M'$:

\[ \{0,1\}^n \xrightarrow{\delta} C[0] \]

\[ E_K \]

\[ C[1] \]

\[ \{0,1\}^n \xrightarrow{\delta} C'[0] \]

\[ E_K \]

\[ C'[1] \]

**Observation:** If $C[0] = C'[0]$ then

\[ C[1] = C'[1] \iff M = M' \]
If 1 block messages are encrypted under CBC, then message equality can be detected whenever the IVs are the same.

But if $\geq 2^{n/2}$ messages are encrypted, we expect by the birthday paradox to see collisions in IVs, so we will be able to break the scheme.

Birthday attack on CBC:

<table>
<thead>
<tr>
<th>Left world</th>
<th>Right world</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>$A$</td>
</tr>
<tr>
<td>$M_0, M_1$</td>
<td>$M_0, M_1$</td>
</tr>
<tr>
<td>$C$</td>
<td>$C$</td>
</tr>
<tr>
<td>$\operatorname{LR} C \leftarrow \mathcal{E}_K(M_0)$</td>
<td>$\operatorname{LR} C \leftarrow \mathcal{E}_K(M_1)$</td>
</tr>
</tbody>
</table>

adversary $A$

for $i = 1, ..., q$ do

\[ C[i][0], C[i][1] \leftarrow \operatorname{LR}((i), (0)) \]

$S \leftarrow \{(j, \ell): C[j][0] = C[\ell][0] \text{ and } 1 \leq j < \ell \leq q\}$

If $S \neq \emptyset$, then

\[ (j, \ell) \leftarrow S \]

If $C[j][1] = C[\ell][1]$ then return 1

return 0

If $C[j][0] = C[\ell][0]$ then

\[ C[j][1] = E_K(\langle 0 \rangle \oplus C[j][0]) = E_K(\langle 0 \rangle \oplus C[\ell][0]) = C[\ell][1] \]

so

\[ \Pr_{\operatorname{Left}^A} = 1 = \Pr[S \neq \emptyset] = C(2^n, q) \]

Birthday attack on CBC: Right world analysis

<table>
<thead>
<tr>
<th>Right world</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
</tr>
<tr>
<td>$M_0, M_1$</td>
</tr>
<tr>
<td>$C$</td>
</tr>
<tr>
<td>$\operatorname{LR} C \leftarrow \mathcal{E}_K(M_1)$</td>
</tr>
</tbody>
</table>

adversary $A$

for $i = 1, ..., q$ do

\[ C[i][0], C[i][1] \leftarrow \operatorname{LR}((i), (0)) \]

$S \leftarrow \{(j, \ell): C[j][0] = C[\ell][0] \text{ and } 1 \leq j < \ell \leq q\}$

If $S \neq \emptyset$, then

\[ (j, \ell) \leftarrow S \]

If $C[j][1] = C[\ell][1]$ then return 1

return 0

If $C[j][0] = C[\ell][0]$ then

\[ C[j][1] = E_K(\langle 0 \rangle \oplus C[j][0]) \neq E_K(\langle \ell \rangle \oplus C[\ell][0]) = C[\ell][1] \]

so

\[ \Pr_{\operatorname{Right}^A} = 1 = \Pr[S \neq \emptyset] = C(2^n, q) \]
Birthday attack on CBC

**adversary** $A$

for $i = 1, \ldots, q$ do

$C_i[0]C_i[1] \overset{\$}{\leftarrow} LR((i), (0))$

$S \leftarrow \{(j, \ell) : C_j[0] = C_\ell[0] \text{ and } 1 \leq j < \ell \leq q\}$

If $S \neq \emptyset$, then

$(j, \ell) \overset{\$}{\leftarrow} S$

If $C_j[1] = C_\ell[1]$ then

return 1

return 0

$$\text{Adv}^{\text{ind-cpa}}_{\text{SE}}(A) = \Pr[\text{Right}^A_{\text{SE}} \Rightarrow 1] - \Pr[\text{Left}^A_{\text{SE}} \Rightarrow 1]$$

$$= C(2^n, q) - 0$$

$$\geq 0.3 \cdot \frac{q(q - 1)}{2^n}$$

Conclusion: CBC$ can be broken (in the IND-CP A sense) in about $2^{n/2}$ queries, where $n$ is the block length of the underlying block cipher, regardless of the cryptanalytic strength of the block cipher.

Security of CBC$

**So far**: A $q$-query adversary can break CBC$ with advantage $\approx \frac{q^2}{2^{n+1}}$.

**Question**: Is there any better attack?

**Answer**: NO!

We can prove that the best $q$-query attack short of breaking the block cipher has advantage at most

$$\frac{\sigma^2}{2^n}$$

where $\sigma$ is the total number of blocks encrypted.

**Example**: If $q$ 1-block messages are encrypted then $\sigma = q$ so the adversary advantage is not more than $q^2/2^n$.

Fact: If $E$ is secure (PRF) then CBC$ mode can be used to securely encrypt up to $2^{n/2}$ blocks, where $n$ is the block length of the block cipher.

This is not much for DES ($n = 64$, $2^{n/2} = 2^{32}$) but a lot for AES ($n = 128$, $2^{n/2} = 2^{64}$).

This means CBC$ is a good, general purpose encryption scheme.

Ciphertexts leak NO partial information about messages.

Provides security regardless of message distribution. Votes can be securely encrypted.

We do not need to look for attacks on the scheme. We are guaranteed there are no attacks as long as $E$ is secure.
Security of CBC$\$ 

Theorem: Let $E: \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a block cipher and $SE = (K,E,D)$ the corresponding CBC$\$ symmetric encryption scheme. Let $A$ be an ind-cpa adversary against $SE$ that has running time $t$ and makes at most $q$ LR queries, these totalling at most $\sigma$ blocks. Then there is a prf-adversary $B$ against $E$ such that 

$$\text{Adv}_{SE}^{\text{ind-cpa}}(A) \leq 2 \cdot \text{Adv}_E^{\text{prf}}(B) + \frac{\sigma^2}{2^n}$$

Furthermore, $B$ makes at most $\sigma$ oracle queries and has running time $t + \Theta(\sigma \cdot n)$.

Security of CBC$\$ 

Then 

$$\text{Adv}_{SE}^{\text{ind-cpa}}(A) = 2 \cdot \Pr\left[G_0^A \Rightarrow \text{true}\right] - 1$$

But 

$$\Pr\left[G_0^A \Rightarrow \text{true}\right] = \Pr\left[G_1^A \Rightarrow \text{true}\right] + \left(\Pr\left[G_0^A \Rightarrow \text{true}\right] - \Pr\left[G_1^A \Rightarrow \text{true}\right]\right)$$

Claim 1: We can design prf-adversary $B$ so that 

$$\Pr\left[G_0^A \Rightarrow \text{true}\right] - \Pr\left[G_1^A \Rightarrow \text{true}\right] \leq \text{Adv}_E^{\text{prf}}(B)$$

Claim 2: $\Pr\left[G_1^A \Rightarrow \text{true}\right] \leq \frac{1}{2} + \sigma^2 \cdot 2^{-n-1}$

So 

$$\text{Adv}_{SE}^{\text{ind-cpa}}(A) \leq 2 \cdot \left(\frac{1}{2} + \frac{\sigma^2}{2^{n+1}}\right) - 1 + 2 \cdot \text{Adv}_E^{\text{prf}}(B)$$

$$= \frac{\sigma^2}{2^n} + 2 \cdot \text{Adv}_E^{\text{prf}}(B)$$

Games for CBC$\$ Security Proof 

Game $G_0$ 

procedure Initialize 

$K \leftarrow \{0,1\}^k$; $b \leftarrow \{0,1\}$; $S \leftarrow \emptyset$

procedure LR($M_0, M_1$) 

$m \leftarrow \|M_b\|_n$; $C[0] \leftarrow \{0,1\}^n$

for $i = 1, \ldots, n$ do 

$P \leftarrow C[i - 1] \oplus M_b[i]$

if $P \notin S$ then $T[P] \leftarrow E_K(P)$

$C[i] \leftarrow T[P]$

$S \leftarrow S \cup \{P\}$

return $C$

procedure Finalize($b'$) 

return $(b = b')$

Game $G_1$ 

procedure Initialize 

$b \leftarrow \{0,1\}$; $S \leftarrow \emptyset$

procedure LR($M_0, M_1$) 

$m \leftarrow \|M_b\|_n$; $C[0] \leftarrow \{0,1\}^n$

for $i = 1, \ldots, n$ do 

$P \leftarrow C[i - 1] \oplus M_b[i]$

if $P \notin S$ then $T[P] \leftarrow \{0,1\}^n$

$C[i] \leftarrow T[P]$

$S \leftarrow S \cup \{P\}$

return $C$

procedure Finalize($b'$) 

return $(b = b')$

Analysis 

Claim 1: We can design prf-adversary $B$ so that: 

$$\Pr\left[G_0^A \Rightarrow \text{true}\right] - \Pr\left[G_1^A \Rightarrow \text{true}\right] \leq \text{Adv}_E^{\text{prf}}(B)$$

adversary $B$ 

$b \leftarrow \{0,1\}$; $S \leftarrow \emptyset$

$b' \leftarrow A^{LR}$

if $(b = b')$ then return 1 else return 0

subroutine LR($M_0, M_1$)

$m \leftarrow \|M_b\|_n$; $C[0] \leftarrow \{0,1\}^n$

for $i = 1, \ldots, m$ do 

$P \leftarrow C[i - 1] \oplus M_b[i]$

if $P \notin S$ then $T[P] \leftarrow \text{Fin}(P)$

$C[i] \leftarrow T[P]$

$S \leftarrow S \cup \{P\}$

return $C$

$$\Pr\left[\text{Real}_E^{\text{prf}} \Rightarrow 1\right] = \Pr\left[G_0^A \Rightarrow \text{true}\right]$$

$$\Pr\left[\text{Rand}_E^{\text{prf}} \Rightarrow 1\right] = \Pr\left[G_1^A \Rightarrow \text{true}\right]$$
Claim 2: \[ \Pr[G_1^A \Rightarrow \text{true}] \leq \frac{1}{2} + \frac{\sigma^2}{2^n+1} \]

Claim 2: \( \Pr[G_1^A \Rightarrow \text{true}] \leq \frac{1}{2} + \frac{\sigma^2}{2^n+1} \)

\[
\Pr[G_1^A \Rightarrow \text{true}] = \Pr[G_2^A \Rightarrow \text{true}] + (\Pr[G_2^A \Rightarrow \text{true}] - \Pr[G_3^A \Rightarrow \text{true}])
\]

Will show:

- \( \Pr[G_3^A \Rightarrow \text{true}] = \frac{1}{2} \)
- \( \Pr[G_2^A \Rightarrow \text{true}] - \Pr[G_3^A \Rightarrow \text{true}] \leq \frac{\sigma^2}{2^n+1} \)

Ciphertext \( C \) in \( G_3 \) is always random, independently of \( b \), so

\[
\Pr[G_3^A \Rightarrow \text{true}] = \frac{1}{2}.
\]
Fundamental Lemma of game playing

Games $G, H$ are identical-until-bad if their code differs only in statements following the setting of bad to true.

Lemma: If $G, H$ are identical-until-bad, then for any adversary $A$ and any $y$

$$\left| \Pr[G^A \Rightarrow y] - \Pr[H^A \Rightarrow y] \right| \leq \Pr[H^A \text{ sets bad}]$$

Using the fundamental lemma

Game $G_2, G_3$

procedure Initialize

$b \leftarrow \{0, 1\}; S \leftarrow \emptyset$

procedure LR($M_0, M_1$)

$m \leftarrow \|M_0\|_n; C[0] \leftarrow \{0, 1\}^n$

for $i = 1, \ldots, m$

$P \leftarrow C[i-1] \oplus M_0[i]$

$C[i] \leftarrow \{0, 1\}^n$

If $P \in S$ then bad $\leftarrow$ true;

$T[P] \leftarrow C[i]$

$S \leftarrow S \cup \{P\}$

return $C$

procedure Finalize($b'$)

return $(b = b')$

$G_2$ and $G_3$ are identical-until-bad, so Fundamental Lemma implies

$$\Pr[G_2^A \Rightarrow true] - \Pr[G_3^A \Rightarrow true] \leq \Pr[G_3^A \text{ sets bad}]$$

Bounding the probability of bad in $G_3$

Game $G_3$

procedure Initialize

$b \leftarrow \{0, 1\}; S \leftarrow \emptyset$

procedure LR($M_0, M_1$)

$m \leftarrow \|M_0\|_n; C[0] \leftarrow \{0, 1\}^n$

for $i = 1, \ldots, m$

$P \leftarrow C[i-1] \oplus M_0[i]$

$C[i] \leftarrow \{0, 1\}^n$

If $P \in S$ then bad $\leftarrow$ true;

$T[P] \leftarrow C[i]$

$S \leftarrow S \cup \{P\}$

return $C$

procedure Finalize($b'$)

return $(b = b')$

The $\ell$-th time the if-statement is executed, it has probability

$$\frac{\ell - 1}{2^n}$$

of setting bad. Thus

$$\Pr[G_4^A \text{ sets bad}] \leq \sum_{\ell=1}^{\sigma} \frac{\ell - 1}{2^n} \leq \frac{\sigma(\sigma - 1)}{2^{n+1}} \leq \frac{\sigma^2}{2^{n+1}}$$
How many LR queries?

The IND-CPA definition allows the adversary multiple queries to its LR oracle. This models the adversary distinguishing between whether the messages encrypted were one stream

\[ M_0^1, \ldots, M_0^q \]

or another stream

\[ M_1^1, \ldots, M_1^q \]

It turns out that allowing only one LR query captures the same security requirement up to a factor \( q \) in the advantage, as long as the adversary has a (plain) encryption oracle as well. This can simplify analyses and the proof will illustrate the hybrid technique.

Find-then-guess

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

Game FTGLeft\( _{\mathcal{E}} \)

\begin{align*}
\text{procedure Initialize} \\
K \leftarrow \mathcal{K}
\end{align*}

\begin{align*}
\text{procedure LR}(M_0, M_1) \\
&\text{return } C \leftarrow \mathcal{E}_K(M_0)
\end{align*}

\begin{align*}
\text{procedure Enc}(M) \\
&\text{return } C \leftarrow \mathcal{E}_K(M)
\end{align*}

Game FTGRight\( _{\mathcal{E}} \)

\begin{align*}
\text{procedure Initialize} \\
K \leftarrow \mathcal{K}
\end{align*}

\begin{align*}
\text{procedure LR}(M_0, M_1) \\
&\text{return } C \leftarrow \mathcal{E}_K(M_1)
\end{align*}

\begin{align*}
\text{procedure Enc}(M) \\
&\text{return } C \leftarrow \mathcal{E}_K(M)
\end{align*}

Adversary \( B \) is allowed only one query to its LR oracle.

\[
\text{Adv}^{\text{ftg}}_{\mathcal{E}}(B) = \Pr[\text{FTGRight}_{\mathcal{E}}^B \Rightarrow 1] - \Pr[\text{FTGLeft}_{\mathcal{E}}^B \Rightarrow 1]
\]

Hybrid Technique: illustration

Suppose \( A \) makes queries

\[
(M_1^1, M_1^2, M_1^3, M_1^4, M_0^1, M_0^2, M_0^3, M_0^4)
\]

Then we will define games \( G_0, G_1, G_2, G_3, G_4 \) so that

<table>
<thead>
<tr>
<th>( i )</th>
<th>Messages encrypted in ( G_i^A )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( M_1^1, M_1^2, M_1^3, M_1^4 )</td>
</tr>
<tr>
<td>1</td>
<td>( M_0^1, M_0^2, M_0^3, M_0^4 )</td>
</tr>
<tr>
<td>2</td>
<td>( M_1^1, M_1^2, M_1^3, M_1^4 )</td>
</tr>
<tr>
<td>3</td>
<td>( M_0^1, M_0^2, M_0^3, M_0^4 )</td>
</tr>
<tr>
<td>4</td>
<td>( M_1^1, M_1^2, M_1^3, M_1^4 )</td>
</tr>
</tbody>
</table>

Find-then-guess

Proposition: Let \( \mathcal{E} = (\mathcal{K}, \mathcal{E}, \mathcal{D}) \) be a symmetric encryption scheme and \( A \) an ind-cca adversary making \( q \) oracle queries and having running time at most \( t \). Then there is a ftg adversary \( B \) making one query to its LR oracle and \( q \) queries to its encryption oracle, such that

\[
\text{Adv}^{\text{ind-cca}}_{\mathcal{E}}(A) \leq q \cdot \text{Adv}^{\text{ftg}}_{\mathcal{E}}(B).
\]

Furthermore, the running time of \( B \) is that of \( A \).
Hybrid Technique

Game $G_i$ ($0 \leq i \leq q$)

**procedure Initialize**

$K \leftarrow \mathcal{K}; \ell \leftarrow 0$

**procedure LR($M_0, M_1$)**

$\ell \leftarrow \ell + 1$

If $\ell > i$ then $C \leftarrow E_K(M_1)$ else $C \leftarrow E_K(M_0)$

Return $C$

Suppose $A$ makes LR queries $(M_0^1, M_1^1), \ldots, (M_0^q, M_1^q)$. Then in $G_i^A$ the messages encrypted are

$M_0^1, \ldots, M_0^i, M_1^i, \ldots, M_1^q$

Let

$p_i = \Pr[G_i^A \Rightarrow 1].$

Properties of the hybrid games

In $G_0^A$ the messages encrypted are $M_1^1, \ldots, M_1^q$, so

$$\Pr[\text{Right}_{SE} \Rightarrow 1] = p_0.$$ 

In $G_q^A$ the messages encrypted are $M_0^1, \ldots, M_0^q$, so

$$\Pr[\text{Left}_{SE} \Rightarrow 1] = p_q.$$ 

So,

$$\text{Adv}_{SE}^{\text{ind-cpa}}(A) = p_0 - p_q = (p_0 - p_1) + (p_1 - p_2) + \ldots + (p_{q-1} - p_q)$$

If $p_0 - p_q$ is large, so is at least one term in the sum. We design $B$ to have advantage that term.

Design of $B$

**adversary $B$**

$\ell \leftarrow 0$

$g \leftarrow \mathcal{S}\{1, \ldots, q\}$

$b^i \leftarrow A_{\text{ELR}(.)}$

Return $b^i$

Subroutine **ELR**

$\ell \leftarrow \ell + 1$

If $\ell > g$ then $c \leftarrow E_K(M_1)$

If $\ell = g$ then $c \leftarrow \text{LR}(M_0, M_1)$

If $\ell < g$ then $c \leftarrow E_K(M_0)$

Suppose $A$'s queries are $(M_1^1, M_1^1), \ldots, (M_0^q, M_1^q)$ and suppose $B$ picks $g = i$. Then the messages encrypted are

$M_0^1, \ldots, M_0^{i-1}, M_1^i, M_1^{i+1}, \ldots, M_1^q$

so

$$\Pr[\text{FTGRight}_{SE} \Rightarrow 1 | g = i] = p_{i-1}$$

$$\Pr[\text{FTGLeft}_{SE} \Rightarrow 1 | g = i] = p_i$$

Analysis of $B$

$$\text{Adv}_{SE}^{\text{ftg}}(B) = \Pr[\text{FTGRight}_{SE}^{B} \Rightarrow 1] - \Pr[\text{FTGLeft}_{SE}^{B} \Rightarrow 1]$$

$$= \sum_{i=1}^{q} \Pr[\text{FTGRight}_{SE}^{B} \Rightarrow 1 | g = i] \cdot \Pr[g = i]$$

$$- \sum_{i=1}^{q} \Pr[\text{FTGLeft}_{SE}^{B} \Rightarrow 1 | g = i] \cdot \Pr[g = i]$$

$$= \sum_{i=1}^{q} p_{i-1} \cdot \frac{1}{q} - \sum_{i=1}^{q} p_i \cdot \frac{1}{q}$$

$$= \frac{1}{q} (p_0 - p_q) - \frac{1}{q} \text{Adv}_{SE}^{\text{ind-cpa}}(A)$$

as desired.
ATM card contains a key $K \xleftarrow{\$} K$ known also to Bank, where $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is a symmetric encryption scheme.

**Attack Setting**

Adversary transmits Alice's identity, but how can it answer the challenge (meaning decrypt $C$) without knowing Alice's key?

**Active Attack**

Tries to get $K$ or learn how to decrypt by creating ciphertexts and getting the card to decrypt them.

This is called a **chosen ciphertext attack**.

**Chosen-ciphertext attacks**

**New capability:** Adversary has access to a decryption oracle

$$C \rightarrow \boxed{\text{Dec}} \rightarrow M$$

**What is the adversary's goal?**

In our example it was to get the key $K$, but based on the principles we have discussed before we would like to ask for more: no partial information on un-decrypted messages is leaked by the ciphertexts.
Let $\mathcal{E} = (K, E, D)$ be an encryption scheme. An ind-cca adversary $A$
- Has access to a LR oracle
- Has access to a decryption oracle $\text{Dec}$
- Outputs a bit

\[
\begin{array}{c}
M_0, M_1 \quad \text{LR} \\
A \quad C \\
C' \\
M' \quad \text{Dec}
\end{array}
\]

The games

Let $\mathcal{E} = (K, E, D)$ be a symmetric encryption scheme and let $A$ be an adversary. Consider

**Game Left**$_{\mathcal{E}}$

- procedure Initialize
  - $K \leftarrow K$
- procedure LR($M_0, M_1$)
  - Return $C \leftarrow E_K(M_0)$
- procedure Dec($C$)
  - Return $M \leftarrow D_K(C)$

**Game Right**$_{\mathcal{E}}$

- procedure Initialize
  - $K \leftarrow K$
- procedure LR($M_0, M_1$)
  - Return $C \leftarrow E_K(M_0)$
- procedure Dec($C$)
  - Return $M \leftarrow D_K(C)$

Associated to $\mathcal{E}, A$ are the probabilities

\[
\begin{align*}
\Pr \left[ \text{Left}^A_{\mathcal{E}} \Rightarrow 1 \right] & \quad \Pr \left[ \text{Right}^A_{\mathcal{E}} \Rightarrow 1 \right]
\end{align*}
\]

that $A$ outputs 1 in each world. The (ind-cca) advantage of $A$ is

\[
\text{Adv}^\text{ind-cca}_{\mathcal{E}}(A) = \Pr \left[ \text{Right}^A_{\mathcal{E}} \Rightarrow 1 \right] - \Pr \left[ \text{Left}^A_{\mathcal{E}} \Rightarrow 1 \right]
\]

A problem

We can **ALWAYS** design $A$ with advantage 1, meaning **ALL** schemes are insecure.

**adversary** $A$

$C \leftarrow LR(0^n, 1^n); M \leftarrow \text{Dec}(C)$

if $M = 0^n$ then return 0 else return 1

Then

\[
\begin{align*}
\Pr \left[ \text{Left}^A_{\mathcal{E}} \Rightarrow 1 \right] & = 0 \\
\Pr \left[ \text{Right}^A_{\mathcal{E}} \Rightarrow 1 \right] & = 1
\end{align*}
\]
Avoiding the problem

Encryption can only hide information about un-decrypted messages!

We address this by making the following rule:

- An ind-cca adversary $A$ is not allowed to query $\text{Dec}$ on a ciphertext previously returned by $\text{LR}$

Adversary from before breaks rule:

adversary $A$

$C \leftarrow \text{LR}(0^n, 1^n); M \leftarrow \text{Dec}(C)$

if $M = 0^n$ then return 0 else return 1

IND-CCA attack on CBC

What we would like to do:

adversary $A$

$C \leftarrow \text{LR}(0^n, 1^n); M \leftarrow \text{Dec}(C)$

if querying $C$ is not allowed. Instead we will

$C \rightarrow \text{ModifyC} \rightarrow C' \rightarrow \text{Dec} \rightarrow M' \rightarrow \text{ModifyM} \rightarrow M$

so that $M = D_K(C)$ but $C' \neq C$. Then

adversary $A$

$C \leftarrow \text{LR}(0^n, 1^n)$

$C' \leftarrow \text{ModifyC}(C); M' \leftarrow \text{Dec}(C'); M \leftarrow \text{ModifyM}(M')$

if $M = 0^n$ then return 0 else return 1

IND-CCA attack on CBC

Let $E : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a block cipher.

**Alg** $E_K(M)$

$C[0] \leftarrow \{0, 1\}^n$; for $i = 1, \ldots, m$ do $C[i] \leftarrow E_K(M[i] \oplus C[i-1])$

return $C$

Left world

Right world

Can we design $A$ so that

$\text{Adv}^{\text{ind-cca}}_{SE}(A) = \Pr[\text{Right}_{SE} \Rightarrow 1] - \Pr[\text{Left}_{SE} \Rightarrow 1]$ is close to 1?

The Modify process

Let $\Delta \neq 0^n$ be some block.

$C[0] \leftarrow \text{ModifyC} \rightarrow C'[0] \leftarrow C[0] \oplus \Delta \rightarrow C'[0]C[1]$

$C'[0]C[1] \rightarrow \text{Dec} \rightarrow M' = M \oplus \Delta$

$M' \rightarrow \text{ModifyM} \rightarrow M$
IND-CCA attack on CBC$: Right world analysis

adversary $A$

$C[0]C[1] \xleftarrow{\$} LR(0^n, 1^n); \Delta \leftarrow 1^n$

$C'[0] \leftarrow C[0] \oplus \Delta; M' \leftarrow \text{Dec}(C'[0]C[1]); M \leftarrow M' \oplus \Delta$

if $M = 0^n$ then return 0 else return 1

Game Right$_{SE}$

procedure Initialize

$K \xleftarrow{\$} \mathcal{K}$

procedure LR($M_0, M_1$)

Return $C \xleftarrow{\$} E_K(M_0)$

procedure Dec($C$)

return $M \leftarrow D_K(C)$

Then

$$\Pr[\text{Right}_A^SE \Rightarrow 1] = 1$$

because $C[0]C[1] \xleftarrow{\$} E_K(1^n)$ so $M = 1^n \neq 0^n$.

IND-CCA attack on CBC$: Left world analysis

adversary $A$

$C[0]C[1] \xleftarrow{\$} LR(0^n, 1^n); \Delta \leftarrow 1^n$

$C'[0] \leftarrow C[0] \oplus \Delta; M' \leftarrow \text{Dec}(C'[0]C[1]); M \leftarrow M' \oplus \Delta$

if $M = 0^n$ then return 0 else return 1

Game Left$_{SE}$

procedure Initialize

$K \xleftarrow{\$} \mathcal{K}$

procedure LR($M_0, M_1$)

Return $C \xleftarrow{\$} E_K(M_0)$

procedure Dec($C$)

return $M \leftarrow D_K(C)$

Then

$$\Pr[\text{Left}_A^SE \Rightarrow 1] = 0$$

because $C[0]C[1] \xleftarrow{\$} E_K(1^n)$ so $M = 0^n$.

Protecting against CCAs

Can you think of a way to design a scheme that is IND-CCA secure?

We will see such a scheme later, after we have some more tools.

And $A$ is very efficient, making only two queries.

Thus CBC$\$ is not IND-CCA secure.