Symmetric Encryption

Adam O’Neill
based on http://cseweb.ucsd.edu/~mihir/cse207/
Syntax

M → [E] → C → \[\square\] → M

\(k\rightarrow m\sim\rightarrow t\rightarrow c\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)

\(\rightarrow\)
Correctness

Pr \left[ \omega(k, E(k, m)) = m \right] = 1

\forall m \text{ in the message space}

where the probability is over \( k \in \mathcal{K} \) and the coins of \( \mathcal{E} \) if any.
Modes of Operation

\[ E : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^\ell \] a family of functions

Usually a block cipher, in which case \( \ell = n \).

Notation: \( x[i] \) is the i-th block of a string \( x \), so that \( x = x[1] \ldots x[m] \). Length of blocks varies.

Always:

\[
\begin{align*}
\text{Alg } K \\
K & \leftarrow^s \{0, 1\}^k \\
\text{return } K
\end{align*}
\]
Modes of Operation

Block cipher provides parties sharing $K$ with

$$E_K$$

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 code book mode

Algorithm $E_{\text{ecb}}(K, x)$

let $x = [x[1], \ldots, x[m]]$

For $i = 1$ to $m$ do:

$c_i \leftarrow E_K(x_i)$

let $c = [c[1], \ldots, c[m]]$

return $c$

Algorithm $D_{\text{ecb}}(K, c)$

let $c = [c[1], \ldots, c[m]]$

For $i = 1$ to $m$ do:

$x_i \leftarrow E_K^{-1}(c_i)$

let $x = [x[1], \ldots, x[m]]$

return $x$
Security of ECB

leaks in put equality!
Security of ECB

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 $E_K(M) = E_K(M)$. 

\[
\begin{array}{c}
M \\
\downarrow \\
E_K \\
\downarrow \\
C
\end{array}
\]
Voting with ECB

Voters

\[
\begin{align*}
\forall \ p_i \ K \ \ \ \ \ E_{ecb}(\text{Vote}_i) \\
\vdots \\
\forall \ p_n \ K \ \ \ \ \ E_{ecb}(\text{Vote}_n) 
\end{align*}
\]
Is this avoidable?

Let $\mathcal{E} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be 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?
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

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

then

$$Pr[C_1 = C_2]$$

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

A fundamental departure from classical and conventional notions of encryption.

Clasically, 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.
CBC - CTR

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

\[
\text{Alg } E_K(M) \\
C[0] \leftarrow \{0, 1\}^n \quad \text{// called initialization vector (IV)} \\
\text{for } i = 1, \ldots, m \text{ do} \\
\quad C[i] \leftarrow E_K(M[i] \oplus C[i - 1]) \\
\text{return } C
\]

\[
\text{Alg } D_K(C) \\
\text{for } i = 1, \ldots, m \text{ do} \\
\quad M[i] \leftarrow E_K^{-1}(C[i]) \oplus C[i - 1] \\
\text{return } M
\]

Correct decryption relies on \( E \) being a block cipher.
Voting with CBC$
Towards a Definition

• We see CBC$ is better than ECB in some sense.
• But is it secure?
• What does secure mean here?
Security Requirements

Suppose sender computes

\[ C_1 \leftarrow^$ \mathcal{E}_K(M_1); \ldots; C_q \leftarrow^$ \mathcal{E}_K(M_q) \]

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 we want to hide all partial information about the data stream, such as

- 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

- A “master” property that
What we seek

- A “master” property that
  - Can be easily specified
What we seek

• A “master” property that
  • Can be easily specified
  • Allows us to evaluate whether a scheme is secure
What we seek

• A “master” property that
  • Can be easily specified
  • Allows us to evaluate whether a scheme is secure
  • Implies a ciphertext reveals NO partial information about plaintext
What we seek

• A “master” property that
  • Can be easily specified
  • Allows us to evaluate whether a scheme is secure
  • Implies a ciphertext reveals NO partial information about plaintext

• In the case of a blockcipher the master property was PRF-security. Here it is different because encryption should be randomized
Intuition

The master property MP is called IND-CPA (indistinguishability under chosen plaintext attack).

Consider encrypting one of two possible message streams, either

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

or

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

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

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

**Game Left$_{\mathcal{E}}$**
- procedure Initialize
  - $K \leftarrow^{\$} \mathcal{K}$
- procedure LR($M_0, M_1$)
  - Return $C \leftarrow^{\$} \mathcal{E}_K(M_0)$

**Game Right$_{\mathcal{E}}$**
- procedure Initialize
  - $K \leftarrow^{\$} \mathcal{K}$
- procedure LR($M_0, M_1$)
  - Return $C \leftarrow^{\$} \mathcal{E}_K(M_1)$

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

$$\Pr \left[ \text{Left}_{\mathcal{E}}^A \Rightarrow 1 \right] \quad \mid \quad \Pr \left[ \text{Right}_{\mathcal{E}}^A \Rightarrow 1 \right]$$

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

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

\( \text{Adv}_{SE}^{\text{ind-cpa}}(A) \approx 1 \) means \( A \) is doing well and \( SE \) is not ind-cpa-secure.

\( \text{Adv}_{SE}^{\text{ind-cpa}}(A) \approx 0 \) (or \( \leq 0 \)) means \( A \) is doing poorly and \( SE \) resists the attack \( A \) is mounting.

Adversary resources are its running time \( t \) and the number \( q \) of its oracle queries, the latter representing the number of messages encrypted.

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

**Insecurity:** \( SE \) is not IND-CPA-secure if we can specify an explicit \( A \) that uses “few” resources yet achieves “high” ind-cpa-advantage.
IND-CPA security of ECB

Adversary $A$ in $ACRC(\cdot,\cdot)$

\[ M_0 \leftarrow \{0,1\} \]
\[ M_1 \leftarrow \{0,1\} \]
\[ c \leftarrow LR(M_0, M_1) \]
\[ c' \leftarrow LR(M_0, M_1) \]

In LEFT return $\epsilon_i(C(M_0))$

If $c = c'$ return 0
Else return 1

\[
\text{Adv}^{\text{ind-CPA}}_{\text{ECB}}(A) = \Pr[R(\text{right ECB} \Rightarrow \text{1})] - \Pr[R(\text{left ECB} \Rightarrow \text{1})]
\]
Right Game Analysis

Claim: \[ \Pr\left[ \text{Right}^A_{M_0} = 1 \right] = 1. \]

Proof: Since \( M_0 \neq M_1 \) and \( C \) is deterministic, we have \( C \neq C' \). \( \Rightarrow \) \( A \) returns 1.

Left Game Analysis

Claim. \( \Pr[ \text{LEFT}^A \text{ | } \text{EC} B] = 0 \)

Proof. Analogous.
Conclusion

\[ \text{Adv}_{E_{\text{ECB}}}^{\text{IND-CPA}} (A) = 1 - \alpha = 1 \]

\[ \Rightarrow \text{ECB is not IND-CPA secure!} \]
Any deterministic scheme is not IND-CPA

The theorem generalizes to show that the same adversary breaks any deterministic encryption scheme (meaning $E$ is deterministic & stateless).
Let $E: \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^\ell$ be a family of functions. If $X \in \{0,1\}^n$ and $i \in \mathbb{N}$ then $X + i$ denotes the $n$-bit string formed by converting $X$ to an integer, adding $i$ modulo $2^n$, and converting the result back to an $n$-bit string. Below the message is a sequence of $\ell$-bit blocks:

Let $X = X_1X_2 \ldots X_n$ be an $n$-bit string.

**Online**

\[ \text{Alg } E_K(M) \]

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

**Offline**

\[ \text{Alg } D_K(C) \]

\[
\begin{align*}
\text{for } i = 1, \ldots, m \text{ do} & \\
& \quad P[i] \leftarrow E_K(C[0] + i) \\
& \quad M[i] \leftarrow P[i] \oplus C[i] \\
\text{return } M
\end{align*}
\]
Birthday Attack on CTR

Let $E : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^\ell$ be a family of functions and $SE = (K, E, D)$ the corresponding CTR symmetric encryption scheme. Suppose 1-block messages $M_0, M_1$ are encrypted:

Let us say we are lucky if $C_0[0] = C_1[0]$. If so:

$$C_0[1] = C_1[1] \text{ if and only if } M_0 = M_1$$

So if we are lucky we can detect message equality and violate IND-CPA.
The Adversary

Let $1 \leq q < 2^n$ be a parameter and let $\langle i \rangle$ be integer $i$ encoded as an $l$-bit string.

**adversary $A_q^{\text{birthday}}$**

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

$C^i[0] C^i[1] \leftarrow \text{LR}(\langle i \rangle, \langle 0 \rangle)$

$S \leftarrow \{(j, t): C^j[0] = C^t[0] \text{ and } j < t\}$

If $S \neq \emptyset$, then

$(j, t) \leftarrow S \quad$ \(\triangleright\)

If $C^j[1] = C^t[1]$ then return 1

return 0

Want to know $\text{Adv}_{\text{ctr}$^*$}^{\text{ind-cca}}(A_q^{\text{bday}}) \geq \frac{0.3 q^2}{2^l}$

$\triangleright \text{query}$

Birthday attack

Adversary

$q$-query

$\text{Birthday attack}$

adversary

$S \leftarrow \text{LR}(\langle i \rangle, \langle 0 \rangle)$

If $S \neq \emptyset$, then

$(j, t) \leftarrow S \quad$ \(\triangleright\)

If $C^j[1] = C^t[1]$ then return 1

return 0

Want to know $\text{Adv}_{\text{ctr}$^*$}^{\text{ind-cca}}(A_q^{\text{bday}}) \geq \frac{0.3 q^2}{2^l}$
Claim:
\[ \Pr[\text{LEFT} + \text{Abad} \Rightarrow 1] = 0. \]

Proof:

Two cases:

Case 1: \( S \neq \emptyset \). Then \( c^i[1] \oplus ji > c^i[1] \oplus te \)

since \( c^i[1] = c^t[1] \) and \( S \neq t \).

Case 2: \( S = \emptyset \). \( A \) always outputs 0.

So in either case \( A \) outputs 0.
Right Game Analysis

Claim. \( \Pr[\text{RIGHT}^{A_{\text{day}}}_{\text{ctrl}}] \geq 0.3 \frac{q^2}{2^l} \)

Proof. If \( S \neq \emptyset \) then we have

\[
C^l[1] = \langle c^l[0] + 1 \rangle \otimes 0^l = \langle c^l[0] + 1 \rangle \otimes 0^l
\]

since \( C^l[0] = c^l[20] \).
Conclusion

\[ A_{\text{vind-cpl}} (A_{\text{bdry}}^2) \geq 0.3 \frac{q^2}{2^l} \]
So far: A $q$-query adversary can break CTR$ 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

$$\frac{2(q - 1)\sigma}{2^n}$$

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

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

For $E = AES$ this means up to about $2^{64}$ blocks may be securely encrypted, which is good.
Theorem: [BDJR97] Let $E : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^\ell$ be a family of functions and $SE = (K, E, D)$ the corresponding CTR$ symmetric encryption scheme. Let $A$ be an ind-cpa adversary against $SE$ that has running time $t$ and makes at most $q$ LR queries, the messages across them totaling at most $\sigma$ blocks. Then there is a prf-adversary $B$ against $E$ such that

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

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

Thought experiment:
replace $E$ with random function w/ same domain and range.

Now, if the adversary is lucky and makes LR queries such that $C'[0] = C'[\emptyset]$
then it can gain advantage.

Otherwise we are encrypting with a one-time pad.
Interval Intersection

\[ P(A \lor B) \leq P(A) + P(B) \]

Let \( N, q, m \geq 1 \) be integers and let \( Z_N = \{0, 1, \ldots, N - 1\} \). Let \( + \) be addition modulo \( N \). Consider the game

For \( i = 1, \ldots, q \) do
\[ c_i \leftarrow Z_N ; S_i \leftarrow \{c_i + 1, \ldots, c_i + m\} \]

For \( 1 \leq i < j \leq q \) define the events
\[ B_{i,j} : S_i \cap S_j \neq \emptyset \quad \text{and} \quad B : \bigvee_{1 \leq i < j \leq q} B_{i,j} . \]

Then let
\[ \text{IIP}(N, q, m) = \Pr[B] . \]

**Problem:** Upper bound \( \text{IIP}(N, q, m) \) as a function of \( N, q, m \).
Interval Intersection

Claim: \( \text{IIP}(N, q, m) \leq \frac{q(q-1)(2m-1)}{2N} \)

Proof:
\[
\Pr[\{B\}] \leq \sum_{i \lt j} \Pr[B_{ij}] = \binom{q}{2} \Pr[B_{ij}]
\]
and
\[
\text{Pr}[B_{ij}] = \text{Pr}[c_{ij} \in \{c_i-m+1, \ldots, c_i+m-1\}]
\]
\[
= \frac{2m-1}{N}
\]
\[
\Rightarrow \text{Pr}[\{B\}] \leq \frac{q(q-1)(2m-1)}{2N}
\]
Advantage Revisited

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

<table>
<thead>
<tr>
<th>Game $Guess_{SE}$</th>
<th>Procedure $LR(M_0, M_1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>procedure Initialize</strong></td>
<td>return $C \leftarrow E_K(M_b)$</td>
</tr>
<tr>
<td>$K \leftarrow K; b \leftarrow {0, 1}$</td>
<td><strong>procedure Finalize($b'$)</strong></td>
</tr>
<tr>
<td></td>
<td>return $(b = b')$</td>
</tr>
</tbody>
</table>

Proposition: $Adv_{SE}^{ind-cpa}(A) = 2 \cdot Pr\left[Guess_{SE}^A \Rightarrow true\right] - 1$. 

Mihir Bellare UCSD 53
Game Chain

Game $G_0$

procedure LR($M_0, M_1$)

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

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

$P \leftarrow C[0] + i$

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

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

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

return $C$

Then

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

Game $G_1$

procedure LR($M_0, M_1$)

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

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

$P \leftarrow C[0] + i$

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

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

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

return $C$
Claims

Clearly \( \Pr[G_0^A] = \Pr[G_1^A] + (\Pr[G_0^A] - \Pr[G_1^A]) \).

**Claim 1:** We can design prf-adversary \( B \) so that

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

**Claim 2:** \( \Pr[G_1^A] \leq \frac{1}{2} + \frac{(q - 1)\sigma}{2^n} \)
Proof of Claim 1

Adversary $b \in \{0,1\}

b sen $\mathbf{B} \leftarrow \mathbf{F}_\mathbf{N}(\cdot, \cdot)$

Run $A$

When $A$ makes ER every $m_0/m$

Code $T[P] \leftarrow \mathbf{F}_\mathbf{N}(P)$

If $A$ outputs $b$ then set $1$
Proof of Claim 2
Games Con’t

Game $G_1$

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

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

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

$P \leftarrow C[0] + i$

If $P \notin S$ then

$T[P] \leftarrow \{0, 1\}^\ell$

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

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

return $C$

---

Game $G_2, G_3$

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

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

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

$P \leftarrow C[0] + i$

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

If $P \notin S$ then

$bad \leftarrow \text{true}$; $C[i] \leftarrow T[P] \oplus M_b[i]$

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

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

return $C$

\[ \Pr[G_1^A] = \Pr[G_2^A] = \Pr[G_3^A] + \left( \Pr[G_2^A] - \Pr[G_3^A] \right) \]
Using fundamental lemma

Game $G_2$, $G_3$

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

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

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

$P \leftarrow C[0] + i$; $C[i] \leftarrow \{0, 1\}$

If $P \in S$ then

bad $\leftarrow$ true; $C[i] \leftarrow T[P] \oplus M_b[i]$

$T[P] \leftarrow C[i] \oplus M_b[i]$; $S \leftarrow S \cup \{P\}$

return $C$

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

$$\Pr \left[ G_2^A \right] - \Pr \left[ G_3^A \right] \leq \Pr \left[ G_3^A \text{ sets bad} \right].$$
Completing the Proof
Theorem: [BDJR97] 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, the messages across them totaling 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 $t + \Theta(\sigma \cdot n)$. 
Padding

The TLS 1.0 protocol defines the following padding function for encrypting a $v$-byte message with AES in CBC mode: let $p := 16 - (v \mod 16)$, then append $p$ bytes to the message $m$ where the content of each byte is value $p - 1$. For example, consider the following two cases:

- if $m$ is 29 bytes long then $p = 3$ and the pad consists of the three bytes “222” so that the padded message is 32 bytes long which is exactly two AES blocks.

- if the length of $m$ is a multiple of the block size, say 32 bytes, then $p = 16$ and the pad consists of 16 bytes. The padded message is then 48 bytes long which is three AES blocks.
Figure 5.5: The tree of keys for $n = 8$ devices; shaded nodes are the keys embedded in device 3.
Definition 5.4. Let $T$ be a complete binary tree with $n$ leaves, where $n$ is a power of two. Let $S \subseteq \{1, \ldots, n\}$ be a set of leaves. We say that a set of nodes $W \subseteq \{1, \ldots, 2n-1\}$ covers the set $S$ if every leaf in $S$ is a descendant of some node in $W$, and leaves outside of $S$ are not. We use $\text{cover}(S)$ to denote the smallest set of nodes that covers $S$.

Theorem 5.8. Let $T$ be a complete binary tree with $n$ leaves, where $n$ is a power of two. For every $1 \leq r \leq n$, and every set $S$ of $n - r$ leaves, we have

$$|\text{cover}(S)| \leq r \cdot \log_2(n/r)$$
Broadcast Encryption

The studios are worried about piracy, and do not want to send copyrighted digital content in the clear to millions of users. A simple solution could work as follows. Every authorized manufacturer is given a **device key** $k_d \in \mathcal{K}$, and it embeds this key in every device that it sells. If there are a hundred authorized device manufacturers, then there are a hundred device keys $k_{d}^{(1)}, \ldots, k_{d}^{(100)}$. A movie $m$ is encrypted as:

$$c_m := \begin{cases} 
  k \leftarrow \mathcal{K} \\
  \text{for } i = 1, \ldots, 100 : \; c_i \leftarrow E(k_{d}^{(i)}, k) \\
  c \leftarrow E'(k, m) \\
  \text{output } (c_1, \ldots, c_{100}, c) 
\end{cases}$$