Short Review

* CCA1 \rightarrow\text{is non adaptive}  
* CCA2 \rightarrow\text{is adaptive}  
* CCA_2 \rightarrow \text{CCA-1} \Rightarrow \text{CPA}  
* Key recovery attacks  
* CBC

\[ \begin{array}{c}
\text{IV} \\
A \\
\text{DES} \\
C
\end{array} \quad \begin{array}{c}
m_1 \\
+ \\
\text{DES}
\end{array} \]

* CBC mode: if DES is modeled as pseudo-random machine then CBC would be more secure.

* CBC is not secure against \(\oplus\) CCA-1

* DES is PRF ?? We still don't have information but assumed to be secure. An efficiency is not known, so we hope that it works.
Definition

Random function $f : (\mathbb{D}, \mathbb{R})$ range

Domain $\mathbb{D}$ maps inputs in

So if we consider mapping

Size of function family $\mathcal{F}$

If a random function is chosen randomly from the function family

Size of the function family

Function family maps inputs in domain $\mathbb{D}$ to output Range $\mathbb{R}$ considering all possible functions.
Dynamic View of Random function -

Δ Implementing random function
\[ f \in \text{func}(L, L) \]

\[ \text{query} \ x^0 \rightarrow \text{ORACLE} \]

Adversary

\[ \text{pick a } y^0 \text{ random value in range } R \]
\[ y^0 \in \{0, 1\}^L \]

\[ \begin{array}{c|c}
  x^0 & y^0 \\
\end{array} \]

---

**Definition** Random Permutation:

**Domain** \( D \)-domain

\( \& \leq d \)

\[ \text{range } R \]

\( \{ \} \) There is one to one onto mapping

\[ \text{Perm}(L) \]

\[ = l \times (l-1) \times (l-2) \ldots = 1 \]

Random permutation is a permutation chosen at random from the permutation family \( \text{Perm}(L) \)
Dynamic view of a random permutation:

$\text{ORACLE}$

$X \leftarrow \text{perm}(k)$

query $y^*$

pick a random value in $R$

$Y \leftarrow \{0,1\}^n$

$S \subseteq S_U$ $Y$

Here we will be picking up different values because of permutation.

If we look at the case such as $(2^{10})^{23}$ as $L \& L$ values this may not work as practically. Storing this not possible & then choosing a random function so we have to look at best case.

$F : k \times D \rightarrow R$

$F : k \times D \rightarrow R$

$k \leftarrow \{0,1\}^n$

Here we will be choosing key from small domain which is more feasible but we have to know how close this is to the random function. If close then good.
Approximation else if not close to random function then we have bad approximation

**Pseudo Random Function** $^\circ$ - (PRF) \text{ oracle}

Word "0" $\circ$ -

\[ f \overset{\$}{\leftarrow} \text{func}(A, L) \]

\[ g \leftarrow f \]

Word "1" $\circ$ -

\[ k \overset{\$}{\leftarrow} k \in \{0, 1\}^n \]

\[ g \leftarrow Fr \]

\text{ oracle $\overset{\$}{\leftarrow}$ word}

\[ \Pr \left[ A^0 \left( "0" \right) \rightarrow 0 = 1 \right] \]

\[ \Pr \left[ A^0 \left( "1" \right) \rightarrow 1 = 1 \right] \leq \epsilon \]

**Pseudo Random Permutation** $^\circ$ - (PRP)

Word "0" $\circ$ -

\[ \left[ \overset{\$}{\leftarrow} \text{perm}(l) \right] \]

\[ g \leftarrow \left[ \right] \]

Word "1" $\circ$ -

\[ k \overset{\$}{\leftarrow} k \in \{0, 1\}^n \]

\[ g \leftarrow Fr \]
We can infer that for PRP we can have CCA as we have One to One & onto mapping but not for PRF as there is no onto mapping.

**Theorem 8**

\[ \text{PRP-CCA} \Rightarrow \text{PRP-CPA} \]

**Proof**: \[ \text{PRP-CCA} \Rightarrow \text{PRP-CPA} \]

Here there is not much use of decryption capability is made.

**Theorem 8**: Key recovery to PRF/PRP property

\[
\begin{align*}
\text{Encryption model as PRF is more secure. If we model encryption using PRF then we have to see if it is secure against key recovery.}
\end{align*}
\]
Theorem: If an encryption scheme is instantiated with a PRF, then it is secure against key recovery.

\[ \text{PRF} \Rightarrow \text{key recovery} \]

Proof: Let \( A \) which can break the SC (model as PRF) w.r.t key recovery then we can construct \( B \) which can break the "PRF" property.

\[
\begin{align*}
&\text{PRF} \\downarrow \quad \text{Oracle} \quad \uparrow \text{KR} \\
&\text{A} \quad \text{B} \\
&x_1 \cdot x_2 \cdots x_q \leftarrow y_1 \cdot y_2 \cdots y_q \\
&y = g(x) \\
&y = f_{K_1}(x) \rightarrow "1" \\
&\text{else} \quad "0" \\
&\text{Word} \quad 0' \\
&f' \leftarrow \text{func}(d/1) \\
&g \leftarrow f' \\
&g \leftarrow f_{K_1}
\end{align*}
\]
\[ \text{Adv}(B) \geq \text{Adv}(A) - \frac{1}{R^1} \]

**Theorem:** PRF's are easier to work than PRP.

\[ \text{PRF} \succeq \text{PRP} \]

**Solv:**

\[ \text{alterer} = 7 \text{PRF} \Rightarrow 7 \text{PRP} \]

**Instance**

\[ \varepsilon \text{ Adv PRF} - \text{Adv PRP} \]

\[ = \mathcal{P}_6(A^{RF} = 1) - \mathcal{P}_6(A^E = 1) \]

\[ = \mathcal{P}_6(A^{RP} = 1) - \mathcal{P}_6(A^E = 1) \]

\[ = \mathcal{P}_6(A^{RF} = 1) - \mathcal{P}_6(A^{RP} = 1) \]

Word "0" ORACLE

\[ g \leftarrow RF \quad \text{di}(\cdot) \rightarrow y_i \]

Word "1"

\[ g \leftarrow RP \]

\[ \mathcal{P}_6(A^{\text{Alice}}) \]

\[ = \mathcal{P}_6(\text{atleast one collision on } RF) \]
\[ PRP^{\text{Adv}}_A = P_0[\beta^{\text{IT}} = 1] - P_0[\beta^{\text{E}} = 1] \]
\[ = P_0[A^{\text{CBC-IT}} = 1] - P_0[A^{\text{CBC-E}} = 1] \]
\[ = P_0[A^{\text{CBC-IT}} = 1] \]

\[ \text{Enc}_{\text{m}}(M_0) \equiv \text{Enc}_{\text{k}}(M_1) \]
\[ \text{Enc}_{\text{k}}(M) \equiv 1_R \]
\[ = \{ P_0[A^{\text{CBC-IT}} = 1] - P_0[A^{\text{E}} = 1] \} + P_0[A^{\text{E}} = 1] - P_0[A^{\text{CBC-E}} = 1] \]
\[ = \text{Adv}_{\text{CBC-IT}}^{\text{IND-CPA}} - [P_0(A^{\text{E}} = 1) - P_0[A^{\text{CBC-E}} = 1]] \]
\[ P_{0} = 1 - P_{0} \text{ (No collision)} \]

\[ = 1 - \left( 1 - \frac{1}{2^l} \right)^{2^l - 1} \left( 1 - \frac{q-1}{2^l} \right) \]

\[ = 1 - \frac{2^l - 1}{2^l} e^{-2^l - \frac{L}{2^l}} \]

\[ = \frac{q(q-1)}{2^L} \]

\[ \leq \frac{q(q-1)}{2^{l+1}} \]

\[ \text{Theorem 3: CBC instantiated with PRP is secure encryption IND-CPA.} \]

\[ \text{Proof: If there exists an attacker } A \text{ which breaks the encryption scheme C (CBC-instantiation with PRP), then we construct } B \text{ which breaks PRP security.} \]

\[ \text{ADV}_{B}^{\text{PRP}} = P_{0}[\beta^{T} = 1] - P_{0}[\beta^{E} = 1] \]

\[ A \text{ CBC-E IND-CPA } \mapsto B \]