"Chosen plaintext attack"

**Definition**

A adversary

\[
\text{oracle access to } \mathsf{Enc} \\
\mathsf{Enc}(m_k, m) \xrightarrow{\mathsf{Enc}(mk)} c \\
\text{oracle access to } \mathsf{Enc} \\
\text{Verifies } k = \mathsf{Gen}(1^n) \\
b \in \{0, 1\}
\]

A succeeds if \(b = b'\)

- \(\mathsf{Enc}\) must be randomised. (Why?)

**Definition**

"Indistinguishable encryptions under a chosen plaintext attack" "CPA-secure"

\[
\forall \text{ p.p.t. } A \quad \Pr[\mathsf{A} \text{ succeeds}] \leq \frac{1}{2} + \epsilon(n)
\]

**Definition**

"Pseudorandom function"

\[
F : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n
\]

\[
\forall \text{ p.p.t. distinguishes } D \\
\Pr[D_{F_k}(1^n) = 1] - \Pr[D_{F_p}(1^n) = 1] \leq \epsilon(n)
\]

\[x \xrightarrow{\mathsf{PRF}} F(x), \text{ where } F : \{0, 1\}^n \rightarrow \{0, 1\}^n
\]

Using \(\mathsf{PRF}\) to do encryption:

- **Doesn't work:** \(\mathsf{Enc}(m) = F_k(m)\) (Deterministic, not CPA-secure.)
- **Gen:** \(k = \{0, 1\}^n\) uniformly at random
- **Enc:** choose \(r \in \{0, 1\}^n\) u.a.r.
  \[c = (r, F_k(r) \oplus m)\]
- **Dec:** \(c = (r, s)\)
  \[m = F_k(r) \oplus s\]
Thm. $\text{F is P\!R\!E} \Rightarrow \text{Construction is CPA-secure}$

\[ \frac{\text{Pf.}}{\quad \text{By reduction. Assume A can distinguish encryptions. (Enc is not CPA-secure.) With probability 0.5, A can not distinguish)}\}

If $F$ is true random function $f$:
- $A$ makes $g(n)$ oracle queries
  - If $r_c$ used in challenge is repeated, $F$ learns value of $F_w(r_c)$ and succeeds w.p. $1$
  \[\Pr(r_c \text{ repeated across oracle queries}) = \frac{g(n)}{2^n}\]
  \[\Rightarrow \Pr(A \text{ distinguishes}) = \frac{1}{2} + \frac{g(n)}{2^n}\text{ is negligible}\]
  - If $r_c$ not used in challenge, no information leaked,
    \[\Pr(\text{success}) = \frac{1}{2}\]
    \[\Pr[D^{F_w(r_c)}(1^n) = 1] - \Pr[D^{F'_w(r_c)}(1^n) = 1] = \left(\frac{1}{2} + \frac{g(n)}{2^n}\right) - \left(\frac{1}{2} + \frac{g(n)}{2^n}\right) = o(1/n) \quad \text{is negligible}\]

Def. "Strong pseudorandom permutation"

$F: \{0,1\}^n \times \{0,1\}^m \rightarrow \{0,1\}^m$ efficient keyed permutation - one-to-one

\[\forall \text{ p.p.t. distinguishers } D:\]
\[\left| \Pr[D^{F_w(s)}(F_w^{-1}(s)) = 1] - \Pr[D^{F'_w(s)}(F'_w^{-1}(s)) = 1] \right| \leq \epsilon(n)\]

(We give access to inverse function too.)
PRPs = abstraction of block ciphers

DES = Data Encryption Standard

IBM, NSA "We sent the S-boxes odd to Washington. They came back and were all different."

NSA convinced IBM to reduce keysize from 64 to 56 bits.

FIPS standard in 1977

56-bit key length (already in use in 1977)

1997 - DES challenges solved by distributed computation

1998 - Deep Crack EFF $250,000 in 56 hours

DES construction ideal properties "diffusion" - mix input bits together w/ permutations, xor

"confusion" - non-linearity S-boxes

Feistel Network

\[
\text{plaintext} \rightarrow \text{Initial Permutation} \rightarrow (L_0, R_0) \rightarrow \ldots \rightarrow (L_{16}, R_{16}) \rightarrow \text{Final Permutation} \rightarrow \text{cipertext}
\]

\[
\begin{align*}
\text{key} & \quad (64\text{ bits} = 56\text{ actual} + 8\text{ check}) \\
\text{R}_{i-1} & \quad 32\text{ bits} \\
\end{align*}
\]

\[
f_i: [R_{i-1}] \rightarrow 32\text{ bits}
\]

\[
\begin{align*}
\text{expand} & \quad 4\text{ bits} \\
\text{S}_i & \quad 4\text{ bits} \\
\end{align*}
\]

\[
f(x) = \text{S}_1(x) \oplus \ldots \oplus \text{S}_8(x)
\]

\[
\text{output} \rightarrow \text{output}
\]

Adapted DES:

- 2 DES?
  \[
f'_{k_1, k_2}(x) = F_{k_2}(F_{k_1}(x)) \quad \text{(Exercise: Why is this a bad idea?)}
\]

- 3 DES
  \[
f'_{k_1, k_2, k_3} = F_{k_3}(F'_{k_2}(F_{k_1}(x)))
\]

  Why all these? If \( k_1 = k_2 = k_3 \to \text{Enc} \circ \text{Enc} \]

  with 2 keys:

  \[
  \text{F}_{k_1, k_2} = \text{F}_{k_1}(F'_{k_2}(F_{k_1}(x))) \quad \text{popular in finance}
  \]

- DES - X "whitening"

  \[
f_{k_1, k_2, k_3}(x) = k_3 \oplus F_{k_2}(x \oplus k_1)
  \]
Current standard: AES
Chosen in 2000 after NIST-run competition
Rijndael
Joan Daemen, Vincent Rijmen
128, 192, 256-bit versions
also designed Keccak

10 rounds

**Key schedule**

Modes of operation
- how to encrypt an arbitrary-length message with a block cipher

**ECB mode** "Electronic Code Book"

\[ m = m_1, m_2, \ldots, m_n \]
\[ c = F_k (m_1), F_k (m_2), \ldots, F_k (m_n) \]
Deterministic, message can be replayed and distinguished,
commonly used in practice, should rarely be used

**CTR mode** "counter"

- ctr must be random - why?
- CTR-secure, easy to parallelize, allows decryption of
  single blocks

Turns a block cipher into a stream cipher
1. Choose random IV ctr.
2. \( c_i = F_k (\text{ctr} + i) \)
3. \( c_i = c_i \oplus m_i \)

**CBC mode** "Cipher-Block Chaining" < must recommended

- \( IV \) needs to be random
- or use \( E_k (\text{message number}) \) "nonce-generated IV"

\( IV \)
\( m_0 \)
\( m_1 \)
\( c_0 \)
\( c_1 \)
\( \cdots \)