Indistinguishability under Chosen-Plaintext Attack

Left-or-right (lor) encryption oracle used to define IND-CPA security of encryption scheme $SE = (K, E, D)$.

Oracle $E_K(LR(M_0, M_1, b))$ // $b \in \{0,1\}$ and $M_0, M_1 \in \{0,1\}^*$
```
{  if |M0| ≠ |M1|
   { return ⊥; }
   C <-$- EK(Mb);
   return C;
}
```

The adversary chooses a sequence of pairs of messages, $(M_{0,1}, M_{1,1}), \dots, (M_{0,q}, M_{1,q})$, where, in each pair, the two messages have the same length. We give to the adversary a sequence of ciphertexts $C_1, \dots, C_q$ where either (1) C_i is an encryption of $M_{0,i}$ for all $1 \le i \le q$ or, (2) C_i is an encryption of $M_{1,i}$ for all $1 \le i \le q$. In doing the encryptions, the encryption algorithm uses the same key but fresh coins, or an updated state, each time. The adversary gets the sequence of ciphertexts and it must guess whether $M_{0,1}, \dots, M_{0,q}$ were encrypted or $M_{1,1}, \dots, M_{1,q}$ were encrypted. Adversary has to decide in which world it is living!

World 0: The oracle provided to the adversary is $E_K(LR(\cdot, \cdot, 0))$. So, whenever the adversary makes a query (M_0, M_1) with $|M_0| = |M_1|$, the oracle computes $C \xleftarrow{\$} E_K(M_0)$, and returns C as answer!

World 1: The oracle provided to the adversary is $E_K(LR(\cdot,\cdot,1))$. So, whenever the adversary makes a query (M_0, M_1) with $|M_0| = |M_1|$ to its oracle, the oracle computes $C \xleftarrow{\$} E_K(M_1)$, and returns C as the answer.

Let $SE = (K, E, D)$ be a symmetric encryption scheme, and let A be an algorithm (Adversary) that has access to an oracle. We consider the following experiments:

Experiment $Exp_{SE}^{ind\text{-}cpa\text{-}1}(A)$	Experiment $Exp_{SE}^{ind\text{-}cpa\text{-}0}(A)$
$\{ \; k' \xleftarrow{\$} K;$	$\{ \; K' \xleftarrow{\$} K;$
$d \xleftarrow{\$} A^{E_K(LR(\cdot,\cdot,1))};$	$d \xleftarrow{\$} A^{E_K(LR(\cdot,\cdot,0))};$
return d;	return d;
$\}$	$\}$

The IND-CPA advantage of A is defined as:

$$Adv_{SE}^{ind\text{-}cpa}(A) = \Pr[Exp_{SE}^{ind\text{-}cpa\text{-}1}(A) = 1] - \Pr[Exp_{SE}^{ind\text{-}cpa\text{-}0}(A) = 1].$$

If $Adv_{SE}^{ind\text{-}cpa}(A)$ is small (meaning close to zero), it means that A is outputting 1 about as often in world 0 as in world 1, meaning it's not doing a good job of telling which world it is in. If this quantity is large (meaning close to one - or at least far from zero) then the adversary A is doing well, meaning our scheme SE is not secure, at least to the extent that we regard A as "reasonable".

Attack on ECB: Let $E: K \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a block cipher. The ECB symmetric encryption scheme $SE = (K, E, D)$ is used. Consider the following adversary:

```
Adversary A^{E_K(LR(·,·,b))}
{ M1 ← 0^{2n};
  M0 ← 0^n || 1^n;
  C(1) C(2) ← E_K(LR(M0, M1, b));
  if (C(1) = C(2))
  { return 1; }
  else
  { return 0; }
}
```

Here $X(i)$ denotes the i-th block of a string X, a block being a sequence of n bits. The adversary's single oracle query is the pair of messages M_0, M_1. Since each of them is two blocks long, so is the ciphertext computed according to the ECB scheme. We can easily show that $\Pr[\mathrm{Exp}_{SE}^{ind-cpa-1}(A) = 1] = 1$ and

$$\Pr[\mathrm{Exp}_{SE}^{ind-cpa-0}(A) = 1] = 0$$

$\Rightarrow \mathrm{Adv}_{SE}^{ind-cpa}(A) = 1 - 0 = 1 \Rightarrow$ __ECB is not secure__.

Attack on any deterministic, stateless scheme:

Let $SE = (K, E, D)$ be a deterministic, stateless symmetric encryption scheme. Assume there is an integer m such that the plaintext space of the scheme contains two distinct strings of length m. Then there is an adversary A such that: $Adv_{SE}^{ind\text{-}cpa}(A) = 1$. Adversary A runs in time $O(m)$ and asks just two queries, each of length m.

Consider the following adversary:

```
Adversary A^{E_K(LR(·,·,b))}
{ Let X, Y be distinct, m-bit strings in the plaintext
    space;
  C1 <- E_K(LR(X, Y, b));
  C2 <- E_K(LR(Y, Y, b));
  if (C1 = C2)
  { return 1; }
  else
  { return 0; }
}
```

We can easily show that $Pr[Exp_{SE}^{ind\text{-}cpa\text{-}1}(A) = 1] = 1$ and
$Pr[Exp_{SE}^{ind\text{-}cpa\text{-}0}(A) = 1] = 0$.

$\Rightarrow Adv_{SE}^{ind\text{-}cpa}(A) = 1 - 0 = 1 \Rightarrow$ SE is not secure.

where SE is any deterministic, stateless scheme.

Attack on CBCC: let E be a block cipher: $E: K \times \{0,1\}^n \longrightarrow \{0,1\}^n$. Let $SE = (K, E, D)$ be the corresponding counter-based version of the CBC encryption mode. We show that this scheme is insecure. The reason is that the adversary can predict the counter value. Consider the following adversary:

Adversary $A^{E_K(LR(\cdot,\cdot,b))}$

```
{  M0,1 <- 0^n;
   M1,1 <- 0^n;
   M0,2 <- 0^n;
   M1,2 <- 0^(n-1) 1;
   <IV1,C1> <-$ EK(LR(M0,1, M1,1, b));
   <IV2,C2> <-$ EK(LR(M0,2, M1,2, b));
   if(C1 = C2)
   { return 1; }
   else
   { return 0; }
}
```

We can easily show that $\Pr[Exp_{SE}^{ind-cpa-1}(A)=1] = 1$ and

$\Pr[Exp_{SE}^{ind-cpa-0}(A)=1] = 0$.

$\Rightarrow Adv_{SE}^{ind-cpa}(A) = 1 - 0 = 1 \Rightarrow$ CBCC is not secure.