We assume our encryption algorithm is DES, but with minor changes the modes described here apply to other secret-key block ciphers.

The plaintext $P$ is divided into 64-bit blocks $P_1, P_2, ..., P_L$. Corresponding 64-bit blocks of ciphertext $C$ are labeled $C_1, C_2, ..., C_L$.

### 1. Electronic Codebook (ECB Mode)

$P_1$  
$P_2$  
$P_3$

$E_K$  
$E_K$  
$E_K$

$C_1$  
$C_2$  
$C_3$

**Advantages:**

i) Simplest method.

ii) Blocks can be encrypted in parallel.

iii) Error in transmitting one ciphertext block causes that block to decrypt incorrectly, but other blocks are not affected.

**Disadvantages:**

i) Equal plaintext blocks always encrypt to equal ciphertext blocks (until key is changed).

ii) Intruder gaining occasional access to known plaintext might gradually build up a codebook of plaintext-ciphertext pairs, allowing him to (partially) decrypt messages even without the key.

$C_j = E_K(P_j)$,  
$P_j = D_K(C_j)$

### 2. Cipher Block Chaining (CBC Mode)

$P_1$  
$P_2$  
$P_3$

$E_K$  
$E_K$  
$E_K$

$C_1$  
$C_2$  
$C_3$

$C_0 = \text{a random vector, } C_j = E_K(P_j \oplus C_{j-1}) \text{ for } j = 1, 2, ...$  
(Note $C_0$ is transmitted unencrypted.)

$P_j = D_K(C_j) \oplus C_{j-1}$.

**Advantages:**

i) Equal plaintext blocks in different positions encrypt to different ciphertext blocks.

ii) Equal plaintext blocks in same position of different texts encrypt to different ciphertext blocks (if a new random $C_0$ is chosen for each plaintext).

**Disadvantages:**

i) Blocks cannot be encrypted in parallel.

ii) Errors in transmitting a ciphertext block (a byte) make it impossible to recover the corresponding plaintext block, and *also the next plaintext block.*
Here we make DES act more like a stream cipher. We divide the plaintext and ciphertext into 8-bit blocks $P_1$, $P_2$, ..., $P_L$ and $C_1$, $C_2$, ..., $C_L$ (rather than 64-bit blocks; lengths other than 8 could be used).

We use the notation
- $L_8(X)$ = leftmost 8 bits of 64-bit value $X$.
- $R_{56}(X)$ = rightmost 56 bits of 64-bit value $X$.

An initial 64-bit $X_1$ is chosen randomly.

Encryption:
$$X_1 = \text{a random vector},$$
For $j = 1, 2, 3, ...$
$$O_j = L_8(E_K(X_j)), \quad C_j = P_j \oplus O_j, \quad X_{j+1} = R_{56}(X_j) \parallel C_j.$$

$X_1$ is transmitted (unencrypted) along with the ciphertext.

Decryption:
$$P_j = C_j \oplus L_8(E_K(X_j)), \quad X_{j+1} = R_{56}(X_j) \parallel C_j.$$

Note the decryption function is not used.

Advantages:
- i) Each byte (character) of plaintext can be encrypted (or decrypted) as soon as it is available.
- ii) Equal plaintext blocks in same position of different texts encrypt to different ciphertext blocks (if a new random initialization vector $X_1$ is chosen for each plaintext transmitted.)

Disadvantages:
- i) Blocks cannot be encrypted in parallel.
- ii) Errors in transmitting a ciphertext block (a byte) make it impossible to recover the corresponding plaintext block, and the error propagates, but only for about 8 bytes.