补充5 对称密码加密算法

$$P(C(x)) = C(P(x))$$

Feistel Structure:

$$P(C(FE(x,k))) = C(P(FE(x,k)))$$

$$\Downarrow$$

$$y' = DES(C(x), C(k)) = C(DES(x,k)) = C(y)$$

4.b CBC加密模式:

$$C_i = Enc_k(P_i \oplus C_{i-1}),\ C_0 = IV$$

CBC解密模式:

$$P_i = Dec_k(C_i) \oplus C_{i-1}$$

CFB加密模式:

$$C_i = P_i \oplus Enc_k(C_{i-1}),\ C_0 = IV$$

CFB解密模式:

$$P_i = C_i \oplus Enc_k(C_{i-1})$$

Prove: In both CBC and CFB models, decryption can be parallized because the operation of decrypting each block depends only on the previous ciphertext block, not the previous plaintext block. This means that if multiple processes are available, each processor can independently decrypt a ciphertext block.

① CBC 模式: $P_i = Dec_k(C_i) \oplus C_{i-1}$

if all ciphertext bocks $C_1, C_2 \cdots C_n$ are simultaneously sent to different processors and parallel execution of $Dec_k(C_i)$ is initialized, then the decryption operations can be carried out immediately after each processor completes its task. The XOR operation is very simple; hence, the decryption process timing primarily depends on the decryption function Dec.

② CFB 模式: $P_i = C_i \oplus Enc_k(C_{i-1})$

Similar to CBC mode, each ciphertext bock's decryption can be performed independently, as long as each processor can access C_{i-1}.

Conclusion: 虽然加密是顺序但是解密可以并行.

4.8

Step 1: In CBC mode, if $X_1 = X_1'$, then the encrypted ciphertext blocks C_1, and C_1' will also be the same, due to the use of the same key and IV.

Step 2: If for some j, $X_i \neq X_j'$, then $C_j \neq C_j'$. The chained dependency in CBC mode will cause C_{j+1} and C_{j+1}' to diverge as well, even if $X_{j+1} \neq X_{j+1}'$

Step 3: thus, by simply comparing the two sequences of ciphertexts, and adversary can immediately identify the point where X_1 and X_1' start to differ, as this will be the point where the ciphertexts begin to diverage.

4.9 OFB (Output Feedback) CTR(Counter)

開設序列 X 与 X' 和加密结果之间的关系。

1. OFB:

① 加密: a seed value is encipted to create a keystream block. This keystream block is then XORed with the plaintext block to produce the ciphertext block. The same keystream is used for X and X' because the same key and IV are used. $C_i = P_i \oplus O_i$ and $C_i' = P_i' \oplus O_i$ where O_i is the keystream block.

② XOR Property: Commutative and Associative.

$C_i \oplus C_i' = (P_i \oplus O_i) \oplus (P_i' \oplus O_i) = P_i \oplus P_i'$

③ Adversary's Computation: C_i and $C_i' \Longrightarrow P_i \oplus P_i'$ by XORing these ciphertexts.

2. CTR:

① 加密: generates a keystream by encrypting a counter value. The counter is increased for each block, but if the counter is reused with the same key, the resulting keystream will be same for both X and X' $C_i = P_i \oplus E_k(ctr+i)$ and $C_i' = P_i' \oplus E_k(ctr+i)$

② Counter Reuse Vulnerability: Reusing a counter value with the same key in CTR mode is equivalent to using a static IV in OFB mode.

use XOR $C_i \oplus C_i' = P_i \oplus P_i'$

⑥ Adversary's Computation: Compute $C_i \oplus C_i' \Longrightarrow P_i \oplus P_i'$