

# MAS 433 Tutorial 3

Hou Xiaolu

September 10, 2011

## Problem 2

**2.1** Solution: Since in the key schedule of DES, it does not change the value of any key bit, it just changes the order of the bits, we have

$$K_i = \bar{K}'_i$$

for all  $i$ .

## 2.2

*Proof.* Let

$$K' = \bar{K}, \quad P' = \bar{P}, \quad C = E_K(P), \quad C' = E_{K'}(P').$$

First note that the initial permutation does not affect the value of each plaintext bit. Let  $(L_0, R_0)$  and  $(L'_0, R'_0)$  represent the value of the plaintext  $P$  and  $P'$  respectively after initial permutation. Then

$$L'_0 = \bar{L}_0, \quad R'_0 = \bar{R}_0.$$

We claim that

$$L'_k = \bar{L}_k, \quad R'_k = \bar{R}_k \quad \forall k \geq 1. \tag{1}$$

We prove this claim by induction. We have proved this statement for  $k = 0$ . Now assume that it is true for  $k = i$ , i.e.

$$L'_i = \bar{L}_i, \quad R'_i = \bar{R}_i.$$

Then we have for  $k = i + 1$ ,

$$L'_{i+1} = R'_i = \bar{R}_i, \quad L_{i+1} = R_i \Rightarrow L'_{i+1} = \bar{L}_{i+1}.$$

And

$$\begin{aligned}
 R'_{i+1} &= L'_i \oplus F(K'_{i+1}, R'_i) = L'_i \oplus \text{Permutation}(\text{Substitution}(\text{Expansion}(R'_i) \oplus K'_{i+1})) \\
 &= L'_i \oplus \text{Permutation}(\text{Substitution}(\text{Expansion}(R_i) \oplus K_{i+1})) = L'_i \oplus F(K_{i+1}, R_i) \\
 &= \bar{L}_i \oplus F(K_{i+1}, R_i) \\
 R_{i+1} &= L_i \oplus F(K_{i+1}, R_i) \\
 \Rightarrow R'_{i+1} &= \bar{R}_{i+1}.
 \end{aligned}$$

Thus by induction we have proved statement (1).

Then at last the final permutation does not change the value of each bit of the ciphertext and hence

$$C' = \bar{C} \Rightarrow E_{K'}(P') = \overline{E_K(P)} \Rightarrow E_K(P) = \overline{E_{\bar{K}}(\bar{P})}$$

□

**2.3 Solution:** For a brute force search, the attacker guesses one key and try to test whether it is correct using the known ciphertext as well as plaintext until he gets the right key. In the case that the attacker knows the ciphertexts of two plaintexts  $P$  and  $\bar{P}$  he can test less to get the result.

Let  $C$  denote the ciphertext for  $P$ , let  $\bar{C}$  denote that for  $\bar{P}$ . If he guesses the key to be  $K$ , he can first guess whether

$$C = E_K(P),$$

if this is true, then the key is really  $K$ . If this is not true, he can then check if

$$\bar{C} = \overline{E_K(\bar{P})},$$

if this is true, then the key is  $\bar{K}$ . Thus we can see that with the property that we have proved in 2.2 for DES, it reduces the guessing using brute force by half.

**2.3 Solution:** The property in 2.2 holds because for linear calculation

$$\overline{a + b} = \bar{a} + \bar{b}.$$

Thus if we add some nonlinear operations on the round keys. In this case we may not have

$$K_i = \overline{K'_i}$$

as in 2.1.

**Problem 3 Solution:** If  $b_i = 1$ , then  $L_{i+1}$  and  $R_{i+1}$  are swapped one more time. In this case we have

$$L_{i+1} = L_i \oplus F(K_{i+1}, R_i), \quad R_{i+1} = R_i,$$

which is the same as no swapping of the left and right halves and the right half is unchanged after this round.

In the case that  $b_i = 1$  for  $0 \leq i < 15$  and  $b_{15} = 0$ , the result is the same as that there is no swapping in the Feistel Network as well as after the last round. We will have that the right half of the ciphertext is exactly the same as that in the plaintext since there is no change on the right half in this DES-variant. Thus the attacker can get half of the information. This makes this cipher very weak.

#### Problem 4

**4.1 Solution:** From the design of AES we can see that one byte has no influence to other bytes when the plaintext is added to RoundKey<sub>0</sub>. In the first round, a byte has no influence on the other bytes when SubByte and ShiftRows are performed. But in the MixColumns operation we can see that each byte will affect the bytes in the same column, i.e. a byte, say  $B_0$ , affects four bytes. Then in the second round, the ShiftRows operation is designed such that the four elements in one column are relocated to four different columns after ShiftRows. In this case, the four bytes that are affected by  $B_0$  previously will be relocated to each of the four columns and hence they will affect all the four bytes in their own column during MixColumns operation.

Thus two rounds are needed for each byte to affect all the 16 bytes in the state.

**4.2 Solution:** Assume that the SubByte operations are not implemented. We define

$$f(x) = \text{MixColumns}(\text{ShiftRows}(x)).$$

By the design of the operations MixColumns and ShiftRows, we can see that  $f$  is linear with respect to the addition ' $\oplus$ ', i.e. for a plaintext  $P$  and a key  $K$

$$f(P \oplus K) = f(P) \oplus f(K).$$

Then in the first round we have

$$\text{State} = K_1 \oplus f(P \oplus K_0) = K_1 \oplus f(P) \oplus f(K_0) = f(P) \oplus K'_1$$

where  $K'_1 = K_1 \oplus f(K_0)$ .

In the second round we have

$$\text{State} = K_2 \oplus f(P \oplus K'_1) = K_2 \oplus f^2(P) \oplus f(K'_1) = f^2(P) \oplus K'_2$$

where  $K'_2 = K_2 \oplus f(K'_1)$ .

Similarly, we get

$$\begin{aligned} K'_1 &= K_1 \oplus f(K_0) \\ K'_2 &= K_2 \oplus f(K'_1) \\ &\dots \\ K'_{r-1} &= K_{r-1} \oplus f(K'_{r-2}) \end{aligned}$$

and for  $2 \leq i < r$

$$\text{State} = f^i(P) \oplus K'_i.$$

Then in the last round we have

$$\text{Ciphertext} = \text{State} = K_r \oplus \text{ShiftRows}(K'_{r-1} \oplus f^{r-1}(P)) = \text{ShiftRows}(f^{r-1}(P)) \oplus K'_r$$

where  $K'_r = K_r \oplus \text{ShiftRows}(K'_{r-1})$ .

Thus for any plaintext, we have a fixed 128 bit key  $K'_r$ . Then if the attacker knows one block of ciphertext,  $C_0$ , with the corresponding plaintext,  $P_0$ , he can calculate  $\text{ShiftRows}(f^{r-1}(P))$  and get the fixed key

$$K'_r = \text{ShiftRows}(f^{r-1}(P)) \oplus C_0.$$

Then he has broken the cipher, with every ciphertext  $C$  he can get the plaintext  $P$  by block-wise decryption using

$$P = C \oplus K'_r.$$

**4.3 Solution:** If the ShiftRows operations are not implemented, in every round, the bytes in different columns have no influence on the other bytes in other columns. Each

byte can only affect the other bytes in the same column by the operation MixColumns. Hence we get an operation which acts on each column independently. The result is equivalent to four block ciphers with block size 32-bit. In this case the attacker can make a look up table to take record the ciphertext corresponds to every 32-bit message and hence break this cipher.

**4.4 Solution:** If the MixColumns operations are not implemented, each byte has no influence on any other bytes. Hence the resulting operation is a byte-wise operation such that every byte is encrypted independently. This is equivalent to 16 block ciphers with 8-bit block size. Since there are 10 rounds, by the design of the ShiftRows operation, after encryption, the first row and the third row are not changed except for ten times SubByte operations on each byte. For the second row and the fourth row, besides the ten times of SubByte operations, the 2nd row is left rotated by two-byte position and the 4th by 1-byte position.

**Problem 5 Solution:** Let

$$m(x) = x^8 + x^4 + x^3 + x + 1.$$

### 5.1

By definition

$$\{09\} = 00001001 = x^3 + 1 \quad \{82\} = 10000010 = x^7 + x.$$

Then

$$\begin{aligned} (x^3 + 1)(x^7 + x) &= x^{10} + x^7 + x^4 + x = x^2 m(x) + (x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x) \\ &\Rightarrow x^{10} + x^7 + x^4 + x \equiv x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x \mod m(x) \\ &\Rightarrow \{09\} \bullet \{82\} = 11111110 = \{fe\}. \end{aligned}$$

**5.2** We use the extended Euclidean Algorithm to find the inverse of  $\{09\} = x^3 + 1$  modulo  $m(x)$ . By long division we have

$$\begin{aligned} m(x) &= (x^5 + x^2 + x + 1)(x^3 + 1) + x^2 \\ x^3 + 1 &= x \cdot x^2 + 1 \\ x^2 &= x^2 \cdot 1. \end{aligned}$$

Thus  $\gcd(m(x), x^3 + 1) = 1$  and by extended Euclidean algorithm

$$\begin{aligned} 1 &= x^3 + 1 + x \cdot x^2 \\ &= (x^3 + 1) + x(m(x) + (x^5 + x^2 + x + 1)(x^3 + 1)) \\ &= (x^3 + 1) + xm(x) + (x^6 + x^3 + x^2 + x)(x^3 + 1) \\ &= xm(x) + (x^6 + x^3 + x^2 + x + 1)(x^3 + 1). \end{aligned}$$

Thus the inverse of  $\{09\} = x^3 + 1$  modulo  $m(x)$  is

$$x^6 + x^3 + x^2 + x + 1 = 01001111 = \{4f\}.$$

### 5.3

$$\begin{aligned} a_3 &= \{03\} = 00000011 = x + 1, \quad a_1 = a_2 = \{01\} = 00000001 = 1, \quad a_0 = \{02\} = 00000010 = x \\ b_3 &= b_2 = b_0 = 0, \quad b_1 = \{a3\} = 10100011 = x^7 + x^5 + x + 1. \end{aligned}$$

Then we have

$$\begin{aligned} a_0 \times b_0 + a_3 \times b_1 + a_2 \times b_2 + a_1 \times b_3 &= a_3 \times b_1 = (x + 1)(x^7 + x^5 + x + 1) = x^8 + x^7 + x^6 + x^5 + x^2 + 1 \\ &\equiv x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 \pmod{m(x)} \\ &\Rightarrow d_0 = 11111111 = \{ff\}. \end{aligned}$$

$$\begin{aligned} a_1 \times b_0 + a_0 \times b_1 + a_3 \times b_2 + a_2 \times b_3 &= a_0 \times b_1 = x(x^7 + x^5 + x + 1) = x^8 + x^6 + x^2 + x \\ &\equiv x^6 + x^4 + x^3 + x^2 + 1 \pmod{m(x)} \\ &\Rightarrow d_1 = 01011101 = \{5d\}. \end{aligned}$$

$$a_2 \times b_0 + a_1 \times b_1 + a_0 \times b_2 + a_3 \times b_3 = a_1 \times b_1 = x^7 + x^5 + x + 1 \Rightarrow d_2 = 10100011 = \{a3\}.$$

Similarly, we get  $d_3 = \{a3\}$ . So

$$a(x) \otimes b(x) = \{a3\}x^3 + \{a3\}x^2 + \{5d\}x + \{ff\}.$$

**5.4** Let  $b(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\}$ . Then

$$\begin{aligned} a_3 &= \{03\} = 00000011 = x + 1, \quad a_1 = a_2 = \{01\} = 00000001 = 1, \quad a_0 = \{02\} = 00000010 = x \\ b_0 &= \{0e\} = 00001110 = x^3 + x^2 + x, \quad b_1 = \{09\} = 00001001 = x^3 + 1, \\ b_2 &= \{0d\} = 00001101 = x^3 + x^2 + 1 \quad b_3 = \{0b\} = 00001011 = x^3 + x + 1. \end{aligned}$$

Then we have

$$\begin{aligned}
 & a_0 \times b_0 + a_3 \times b_1 + a_2 \times b_2 + a_1 \times b_3 \\
 &= x(x^3 + x^2 + x) + (x + 1)(x^3 + 1) + 1 \times (x^3 + x^2 + 1) + 1 \times (x^3 + x + 1) \\
 &= x^4 + x^3 + x^2 + x^4 + x + x^3 + 1 + x^3 + x^2 + 1 + x^3 + x + 1 = 1 \\
 &\Rightarrow d_0 = 1.
 \end{aligned}$$

$$\begin{aligned}
 & a_1 \times b_0 + a_0 \times b_1 + a_3 \times b_2 + a_2 \times b_3 \\
 &= 1 \times (x^3 + x^2 + x) + x(x^3 + 1) + (x + 1)(x^3 + x^2 + 1) + 1 \times (x^3 + x + 1) \\
 &= x^3 + x^2 + x + x^4 + x + x^3 + x + x^3 + x^2 + 1 + x^3 + x + 1 = 0 \\
 &\Rightarrow d_1 = 0.
 \end{aligned}$$

$$\begin{aligned}
 & a_2 \times b_0 + a_1 \times b_1 + a_0 \times b_2 + a_3 \times b_3 \\
 &= 1 \times (x^3 + x^2 + x) + 1 \times (x^3 + 1) + x(x^3 + x^2 + 1) + (x + 1)(x^3 + x + 1) \\
 &= x^3 + x^2 + x + x^3 + 1 + x^4 + x^3 + x + x^4 + x^2 + x + x^3 + x + 1 = 0 \\
 &\Rightarrow d_2 = 0.
 \end{aligned}$$

$$\begin{aligned}
 & a_3 \times b_0 + a_2 \times b_1 + a_1 \times b_2 + a_0 \times b_3 \\
 &= (x + 1)(x^3 + x^2 + x) + 1 \times (x^3 + 1) + 1 \times (x^3 + x^2 + 1) + x(x^3 + x + 1) \\
 &= x^4 + x^3 + x^2 + x^3 + x^2 + x + x^3 + 1 + x^3 + x^2 + 1 + x^4 + x^2 + x = 0 \\
 &\Rightarrow d_3 = 0.
 \end{aligned}$$

Thus we have  $a(x) \otimes b(x) = 1$ , i.e.  $a^{-1}(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\} \pmod{(x^4 + 1)}$ .