Ex 5.4 Consider a modified substitution-permutation network where instead of carrying out the key-mixing, substitution, and permutation steps in alternating order for r rounds, the cipher instead first applies r rounds of key-mixing, then carries out r rounds of substitution, and finally applies r permutations. Analyze the security of this construction.

Answer:

Assume that the original plaintext is x, the master key is k and all the subkeys derived from k by key schedule are $k_1, k_2, \cdots, k_r$. After applying r rounds of key-mixing, the intermediate value is: $IV = x \oplus (k_1 \oplus k_2 \oplus \cdots \oplus k_r)$. Let $k_{IV} = (k_1 \oplus k_2 \oplus \cdots \oplus k_r)$, then $IV = x \oplus k_{IV}$. By this way, we have shown that, IV can be generated directly by just one round of key-mixing with key k_{IV} instead of r rounds of key-mixing.

We can then parse IV as 8 consecutive blocks $IV_1, IV_2, \cdots, IV_8$. After carrying r rounds of substitution, the output is $S = (S_1, S_2, \cdots, S_8)$ and each S_i is derived from applying r rounds of substitution in ith $S - box$. Since each $S - box$ is $1 - 1$ and *onto*, thus r rounds of substitution in ith $S - box$ is the same as 1 round of substitution in ith $S - box$.

In a similar way, the final r permutations is the same as 1 permutation.

Since S-boxes and mixing permutations are publicly known(in accordance with Kerckhoffs' principle), thus we can boil down the whole problem into *Attack on a single-round substitution-permutation network*. In this scenario, only a single input/output pair (x,y) is needed to get the secret key, which is k_{IV} in this problem. Then we can use k_{IV} to get the real secret key k. Hence we Son that this construction is not CPA-secure.

Ex 5.5 What is the output of an r-round Feistel network when the input is (L_0, R_0) in each of the following two cases:

Answer:

(a) Each round function outputs all 0s, regardless of the input.

$$output = \begin{cases} (L_0, R_0), & r \equiv 0 \pmod 4 \\ (R_0, L_0 \oplus 0^i), & r \equiv 1 \pmod 4 \\ (L_0 \oplus 0^i, R_0 \oplus 0^i), & r \equiv 2 \pmod 4 \\ (R_0 \oplus 0^i, L_0), & r \equiv 3 \pmod 4 \end{cases}$$

Here, 0^i represents the string constructed by i consecutive 0

(b) Each round function is the identity function.

$$output = \begin{cases} (L_0, R_0), & r \equiv 0 \pmod 4 \\ (R_0, L_0 \oplus R_0), & r \equiv 1 \pmod 4 \\ (L_0 \oplus R_0, L_0), & r \equiv 2 \pmod 4 \\ (L_0, R_0), & r \equiv 3 \pmod 4 \end{cases}$$

Ex 5.6 Show that DES has the property that $DES_k(x) = \overline{DES_{\overline{k}}(\overline{x})}$ for every key k and input x (where $\overline{z}$ denotes the bitwise complement of z). This is called the *complementarity property* of DES.

Answer:

The DES block cipher is a 16-round Feistel network with a block length of 64 bits and a key length of 56 bits. So, here, the length of x and k are 64 bits and 56 bits separately.

Theorem 1. *Let k_i be the ith sub-key derived from master key k. Then the ith sub-key derived from master key $\overline{k}$ should be $\overline{k_i}$.*

Proof. This is because, in each round, the left half and right half of the sub-key are taken as subsets of the left half and right half of the master key separately. And the positions of the bits taken as subsets from the master key is fixed since the *key schedule* of DES remains unchanged, thus every bit taken from $\overline{k}$ is just the reverse bit taken at the same position from k. □

Theorem 2. *Let x be the 32-bit input to the expansion function $E : \{0,1\}^{32} \to \{0,1\}^{48}$ and E outputs 48-bit $E(x)$. Then, if $\overline{x}$ is the input to E, the output is 48-bit $\overline{E(x)}$. That is, $E(\overline{x}) = \overline{E(x)}$.*

Since expansion works by duplicating half of the input bits, we can proof Theorem 2 in a similar way as in Theorem 1

Theorem 3. *In ith round, let k be the master key k, s be the 32-bit input to E, and the output of the* first step of internal DES f-function *be OF. Then, if $\overline{k}$ is the master key, $\overline{s}$ is the 32-bit input to E, then the output of the* first step of internal DES f-function *is still OF*

Proof. In the definition of DES, in ith round, the 48-bit output of the expansion function is xored with the sub-key in the first step of function f. To be specific, the output is: $OF = E(x) \oplus k_i$. According theorem 1 and theorem 2, the output of ith round of the first step of f when receiving key $\overline{k}$ and 32-bit input $\overline{x}$ is:

$$E(\overline{x}) \oplus \overline{k_i} = \overline{E(x)} \oplus \overline{k_i} = OF.$$

□

Hence, we can see that in both cases at every around, one uses key k and 32-bit input x, the other one uses bitwise complement form of them, which are $\overline{k}$ and $\overline{x}$, the $S - boxes$ will produce the same 32-bit output.

All the theorems above show the theorem below:

Theorem 4. *At ith round, the case when* internal function f *receives sub-key k_i and 32-bit input x, and the case when* f *receives sub-key $\overline{k_i}$ and 32-bit input $\overline{x}$, will produce the same output. Which is:*

$$f(\overline{k_i}, \overline{x}) = \overline{f(k_i, x)}$$

From a high-level view, the 16-round *Feistel network* of DES works as:

$$x_{i+1} = y_i, y_{i+1} = x_i \oplus f(k_i, y_i), 0 \leq i \leq 15.$$

, where x_i and y_i are the left half and right half of output in ith round of $Feistelnetwork$, x_0 and y_0 are left half and right half of 32-bit input for DES.

In ith round, if we take input $(\overline{x_i}, \overline{y_i})$ as the bitwise complement of (x_i, y_i), we have:

$$\begin{aligned} x_{i+1} &= \overline{y_i}, && (1) \\ y_{i+1} &= \overline{x_i} \oplus f(\overline{k_i}, \overline{x}) && (2) \\ &= \overline{x_i} \oplus \overline{f(k_i, x)}, && (3) \\ &= \overline{x_i \oplus f(k_i, x)} && (4) \end{aligned}$$

, where the the second step of calculating y_{i+1} follows by theorem 4.

Let the input and key of DES be k and x, and let O be the output of DES. Then, if we take DES's input and key as $\overline{k}$ and $\overline{x}$, from *mathematical induction* and the equation above, we have that the output of DES is $\overline{O}$.

In other words, $DES_k(x) = \overline{DES_{\overline{k}}\overline{x}}$.

Ex 5.7 Use the previous exercise to show how it is possible to find the secret key in DES(with probability 1) in time 2^{55}. Hint: Use a chosen-plaintext attack with two carefully chosen plaintexts.

Answer:

In CPA scenario, first, we choose plaintext m and use the encryption oracle to get the ciphertexts of m and its complemnt form $\overline{m}$, which denoted by (m, c_1) and $(\overline{m}, c_2)$. Then we use DES to encrypt m by applying every key from key space $K = k : k \in \{0,1\}^{55}|0$ (55 random bits concatenated by one 0). For key k, we denote the cipher text we get by c'. In order to simplify the discussion, we will call keys which can be used to encrypt m and get c_1, or can be used to encrypt $\overline{m}$ and get c_2, *suspicious key.*

- If c' equals to c_1, then key k is *suspicious key*
- If $\overline{c'}$ equals to c_2, then $\overline{k}$ is *suspicious key.* This is because if $DES_k(m) = c' = \overline{c_2}$, then the previous exercise shows that:

  $$DES_{\overline{k}}\overline{m} = c_2$$

  .

Next, we will use the encryption oracle to further check *suspicious key*, which is trivial. We believe the *suspicious keys* we found are far less than 2^{55} and the times we need to further check them are far less than 2^{55} too.[1]. The only thing we need to show is that we have make judgement of all the 2^{56} keys, which is quite obvious to see since for each key ended with 0, we have verified its bitwise complement, which is ended with 1.

[1] Actually, I just used "we believe" here cause proving them is beyond my ability now

Ex 5.8 In the actual construction of DES, the two halves of the output of the final round of Feistel network are swapped. That is, if the output of the final round is L_{16}, R_{16} then the output of the cipher is in fact (R_{16}, L_{16}). show that the only difference between the computation of DES_k and DES_k^{-1} (given the swapping of halves) is the order of sub-keys.

Answer:

Let (L_0, R_0) be the two halves of the initial input. Let $(L_1, R_1), (L_2, R_2), \cdots,$ (L_{16}, R_{16}) be the consecutive outputs of 16 rounds of Feistel network. Let k be the master key. Let $k_1, k_2, \cdots, k_{16}$ be sub-keys generated from k for 16 rounds of Feistel network. Let f be the *internal DES function*. Then, for $0 \leq i \leq 15$, we have:

$$L_{i+1} = R_i, \tag{5}$$

$$R_{i+1} = L_i \oplus f(k_{i+1}, R_i) \tag{6}$$

Assume that, at the ith round of DES_k^{-1}, the two halves of output of the last round are (R_{17-i}, L_{17-i}) and the sub-key of this round is k_{17-i} (For the first round, the output of the last round are (R_0, L_0)).

Since equations 5,6 shows that:

$$L_{17-i} = R_{16-i}, \tag{7}$$

$$R_{17-i} = L_{16-i} \oplus f(k_{17-i}, R_{16-i}) \tag{8}$$

Then, by the definition of DES, the input for the $(i+1)$st round of DES_k^{-1} is:

$$Left\ half = Right\ half\ of\ last\ round = L_{17-i} = R_{16-i} \tag{9}$$

$$Right\ half = (Left\ half\ of\ last\ round) \tag{10}$$

$$\oplus f(key\ of\ this\ round,\ Right\ half\ of\ last\ round) \tag{11}$$

$$= R_{17-i} \oplus f(k_{17-i}, L_{17-i}) \tag{12}$$

$$= (L_{16-i} \oplus f(k_{17-i}, R_{16-i})) \oplus f(k_{17-i}, L_{17-i}) \tag{13}$$

$$= (L_{16-i} \oplus f(k_{17-i}, R_{16-i})) \oplus f(k_{17-i}, R_{16-i}) \tag{14}$$

$$= L_{16_i} \tag{15}$$

.

Therefore, using sub-key k_{17-i}, DES_k^{-1} can transfer (R_{17-i}, L_{17-i}) to (R_{16-i}, L_{16-i}) in its ith round. Since in the first round, the input is (R_{16}, L_{16}) and the sub-key is k_{16}. Thus, by *mathematical induction*, at the 16th round of DES_k^{-1}, we will get (R_0, L_0). Then, since we are using actual construction of DES, in which the two halves of the output of the final round of Feistel network are swapped. Therefor, at last, we get the original plaintext (L_0, R_0).

Ex 5.9 (This exercise assumes the results of the previous exercise.)

(a) Show that for $k = 0^{56}$ it holds that $DES_k(DES_k(x)) = x$. Why does the use of such a key pose a security threat?

Answer:

Since $DES_k^{-1}(DES_k(x)) = x$, thus the question require us to show that: $DES_k^{-1}(DES_k(x)) = DES_k(DES_k(x))$. In order to show that the left part equals to the right part, we need to show that there is no difference in the computation of DES_k and DES_k^{-1} when using key $k = 0^{56}$. The result from *Ex 5.8* shows that, the only difference between the computation of DES_k and DES_k^{-1} (given the swapping of halves) is the order of sub-keys. Let $(k_1, k_2, \cdots, k_{16})$ be the sub-keys generated from $k = 0^{56}$. The only thing we need to show is:

$$(k_{16}, k_{15}, \cdots, k_1) = (k_1, k_2, \cdots, k_{16}) \tag{16}$$

Recall that, in each round, the left-most 24 bits of the sub-key are taken as some subset of the 28 bits in the left half of the master key, and the right-most 24 bits of the sub-key are taken as some subset of the 28 bits in the right half of the master key. Here, the master key is $k = 0^{56}$. Thus, all the sub-keys, $k_1, k_2, \cdots, k_{16}$, are equal to 0^{48}. Thus, equation 16 satisfies. As a result, $DES_k(DES_k(x)) = x$.

(b) Find three other DES keys with the same property. These keys are known as *weak keys* for DES.

Answer:

Ex 5.8 and *Ex 5.9(a)* show that, if the master key k can generate all identical sub-keys, then k will have the same property. By looking closely to the key schedule algorithm of DES, we can find three other keys(64-bit form) as:

- $k = FFFFFFFFFFFFFFFF$
- $k = E1E1E1E1F0F0F0F0$
- $k = 1E1E1E1E0F0F0F0F$

(c) Does the existence of these 4 weak keys represent a serious vulnerability in DES? Explain your answer.

Answer:

No. There are 2^{56} possible keys for DES, and only quite a few keys in them are weak, which is a tiny tiny fraction. Thus we need not to worry. They are very few and easy to recognize. Thus, we can avoid using them easily by eliminate them in key schedule algorithm.