

HW # 6

Solution

1. (20 points) Assume a DES chip with pipelined hardware so that one encryption (or key test) can be done in one clock cycle and the chip can be clocked at 200 MHz. The production of this chip (ASIC) has a one-time cost of \$50,000 for making the masks and then each chip costs \$9.

- (a) How many chips must run in parallel for an **average search** time of 6 hours.

Average search: Time it takes for one chip (running at $200 \cdot 10^6 \frac{keys}{sec}$)

$$T_1 = \frac{2^{55} \text{ keys}}{200 \cdot 10^6 \frac{keys}{sec}}$$

Time for n chips

$$T_n = \frac{1}{n} T_1 \leq 6h$$

$$\Rightarrow n \geq \frac{2^{55} \text{ keys}}{200 \cdot 10^6 \frac{keys}{sec}} \cdot \frac{1}{6 \cdot 3600 \text{ sec}} = 8340.0 \quad \Rightarrow n = \underline{8340 \text{ chips}}$$

- (b) What is the cost for such a machine. Assume that for each chip we have to calculate 100% overhead on the chip price (not the one-time cost) for glue logic and building the machine?

$$Price = \$9 \cdot n \cdot 2 + 50,000 = \underline{\$200,120}$$

- (c) We try now to take the advances in computer technology into account. Predicting the future tends to be tricky, but the estimate usually applied is Moore's law, according to which computer power doubles every 18 months. How many years do we have to wait until a key search machine can be build for DES with an average search time of 1 hour? We assume that 2,000 DES chips are used in parallel in our machine.

$$\text{Today's computing power} : P_0 = 2000 \cdot 200 \cdot 10^6 \frac{keys}{sec} = 4 \cdot 10^{11} \frac{keys}{sec}$$

$$\text{in 1.5 years} : P_1 = 2 \cdot P_0$$

$$\text{in } n \text{ 1.5 years} : P_n = 2^n \cdot P_0$$

$$\frac{2^{55}}{P_n} \leq 1 \text{ hour}$$

$$\Rightarrow n \geq \log_2 \frac{2^{55} \text{ keys}}{1 \text{ hour} \cdot 4 \cdot 10^{11} \frac{\text{keys}}{\text{sec}}} = \log_2(25.0) = 4.6$$

number of years $\geq 1.5 \cdot n = \underline{7.0 \text{ years}}$

2. (8 points) As shown in the lecture, one important property which makes DES secure is that the S-boxes are non-linear. In this problem we are going to verify this property by computing the output of S1 for several pairs of inputs. Show that $S1(x_1) \oplus S1(x_2) \neq S1(x_1 \oplus x_2)$, where “ \oplus ” denotes bitwise XOR, for:

- (a) $x_1 = 000000, x_2 = 100010$
 $S1(x_1) \oplus S1(x_2) = 1110 \oplus 0001 = 1111$
 $S1(x_1 \oplus x_2) = S1(100010) = 0001 \neq 1111$
- (b) $x_1 = 010011, x_2 = 101111$
 $S1(x_1) \oplus S1(x_2) = 0110 \oplus 0111 = 0001$
 $S1(x_1 \oplus x_2) = S1(111100) = 1001 \neq 0001$

3. (10 points) What is the output of the first iteration of the DES algorithm when the plaintext is all zero and the key is all zero?

The 64-bit input is X with all bits set to 0. The result of the initial permutation $IP(X)$ contains only zeros, hence L_0 is all zero as well as R_0 . The first sub-key K_1 contains only zero as the all zero key K got subjected only to permutation and shifts which don't change the values. The round function computes:

$$R_1 = L_0 \oplus f(R_0, K_1) \quad L_1 = R_0$$

The f-function

- expands R_0 into a 48-bit long string, here containing again only zeros.
- XORs the result with K_1 which results in all zeros.
- Splits this result into pieces of 6-bits length and applies the S-Boxes which results in the following sequence of 4-bit values: (14, 15, 10, 7, 2, 12, 4, 13) which is in binary 1110 1111 1010 0111 0010 1100 0100 1101.
- This result is permuted according to the P-table to the final result of the f-function: 1101 1000 1101 1000 1101 1011 1011 1100.

This gets XORed with L_0 which does not change the result. The final result of the first round is the concatenation of L_0 and R_0 to

0000 0000 0000 0000 0000 0000 0000 0000 1101 1000 1101 1000 1101 1011 1011 1100 or in hexadecimal 0000 0000 *D8D8 DBBC* or in decimal 0, 0, 0, 0, 0, 0, 0, 0, 13, 8, 13, 8, 13, 11, 11, 12.

4. (22 points) Remember that it is desirable for good block ciphers that a change in one input bit effects many output bits (diffusion or avalanche effect). We try now to get a feeling for the diffusion property of DES. We apply now a plaintext input that has a “1” at bit position 57 and all other bits are zero and the key are all zero. (Note that the input word has to run through the initial permutation.)

- (a) How many S-boxes get different inputs compared to the case considered in the previous problem?

The initial permutation maps $IP(57) = 33$, therefore L_0 is all zeros and R_0 is 80000000(hex). The f-function expands the first bit of R_0 to the second and last bit of its 48-bit output 010000 000000 000000 000000 000000 000000 000000 000001(binary). XORing this result with the all zero key will not change it. Broken into block of 6 bits each (S-Box input) we see that only two S-boxes, namely S-box 1 and S-box 8, are affected.

- (b) What is the minimum number of output bits of the S-boxes that will change according to the S-box design criteria?

According to the S-box design criteria a single bit-flip in the inputs should at least flip two output bits. Since two S-Boxes each have one input bit flipped, a minimum of 4 output bits should flip.

- (c) Which output bits change here? Does this fulfill the S-box design criteria?

The output of S-box 1 was 1110 (15 in decimal) and is now 0011 (3 in decimal)
 The output of S-box 8 was 1101 (13 in decimal) and is now 0001 (1 in decimal)
 For S-Box 1 three bits changed, for S-box 8 two bits changed. This fulfills the S-box design criteria.

- (d) What is the output after the first round? Hint: only follow the changed bits, the others have the same value as in the previous problem.

The output of the S-boxes is: (3, 15, 10, 7, 2, 12, 4, 1) which is in binary
 0011 1111 1010 0111 0010 1100 0100 0001. This result is permuted according to the P-table to the final result of the f-function: 1101 0000 0101 1000 0101 1011 1001 1110.
 This gets XORed with L_0 which does not change the result. The final result of the first round is the concatenation of L_0 and R_0 to
 1000 0000 0000 0000 0000 0000 0000 0000 1101 0000 0101 1000 0101 1011 1001 1110(binary),
 8000 0000 D058 5B9E(hex).

- (e) How many output bits have actually changed compared to the case when the plaintext was all zero. (Observe that we only consider a single round here. There will be more and more output differences after every new round. Hence the term *avalanche effect*).

0000 0000 0000 0000 0000 0000 0000 0000 1101 1000 1101 1000 1101 1011 1011 1100
 1000 0000 0000 0000 0000 0000 0000 0000 1101 0000 0101 1000 0101 1011 1001 1110
 over all 6 bits have changed (5 bits in the right half and 1 bit in the left half).