Cryptography (SS2006) Prof. Michael Backes

Solutions for Exercise Sheet 2

Out: 10/05/2006 Saarland University

Problem 1: Key Complementarity of DES

(a) Given $m \in \{0,1\}^{2\cdot 32}$ and $K \in \{0,1\}^{56}$. We have to show

$$\overline{\mathsf{E}^{\mathsf{DES}}(K,m)} = \mathsf{E}^{\mathsf{DES}}(\overline{K},\overline{m}).$$

Let us first note that the initial permutation preserves the bitwise complement. Furthermore, by inspection of the algorithm, we see that the key schedule preserves bitwise complement as well, as it consists of permutations and shifts only. So it is sufficient if we examine one round and prove $\overline{\mathsf{E}_i(K,m)} = \mathsf{E}_i(\overline{K},\overline{m})$ for the i-th round of DES.

Let us consider the application of the i-th round of DES with key K and input message $L_{i-1}||R_{i-1}$, written $\mathsf{E}_i(K, L_{i-1} \mid\mid R_{i-1}) = L_i \mid\mid R_i$ (cf. Lecture Notes 3, Figures 3.2 & 3.8). We want to show that $\mathsf{E}_i(\overline{K}, \overline{L_{i-1}} \mid\mid \overline{R_{i-1}}) = \overline{L_i} \mid\mid \overline{R_i}$. Since $R_{i-1} = L_i$, we immediately know that $\mathsf{E}_i(\overline{K}, \overline{L_{i-1}} \mid\mid \overline{R_{i-1}}) = \overline{L_i} \mid\mid R_i'$. So it suffices to show that $R_i' = \overline{R_i}$. Let us take a look at how the behavior of the round functions f_i.

It is easy to see that if the "Expansion Permutation" EP applied to R_{i-1} yields V_i, then EP applied to $\overline{R_{i-1}}$ yields $\overline{V_i}$. We have already seen that the round key results in $\overline{K_i}$. This leads to the equation

$$\overline{V} \oplus \overline{K_i}.$$

Using the associativity and the commutativity of $\oplus$ and $\overline{X} = X \oplus 1$ we get

$$(V_i \oplus 1) \oplus (K_i \oplus 1) = (V_i \oplus K_i) \oplus (1 \oplus 1) = V_i \oplus K_i$$

So the input to the remaining parts of the round function is identical in both cases, so the final output of the round function W_i is identical in both cases

$$f_i(K_i, R_{i-1}) = f_i(\overline{K_i}, \overline{R_{i-1}}).$$

Using this property we get

$$\overline{L_{i-1}} \oplus f_i(\overline{K_i}, \overline{R_{i-1}}) = \overline{L_{i-1}} \oplus f_i(K_i, R_{i-1}) = 1 \oplus (\underbrace{L_{i-1} \oplus f_i(K_i, R_{i-1})}_{R_i}) = \overline{R_i}$$

We get that after one round, the output is inverted. By a trivial induction, it follows that after an arbitrary number of rounds the output is inverted as well. This proves the claim.

(b) The goal is to reduce the amount of DES operations by 1/2 using part (a). In order to achieve this, we encrypt an arbitrary message m and its complement $\overline{m}$ with the same key K, thus we get $c_m = \mathsf{E}(K,m)$ and $c_{\overline{m}} = \mathsf{E}(K,\overline{m})$. Since it is very unlikely to get these encryptions by simply eavesdropping passively, the attack described is a chosen-plaintext attack.

Now we proceed as in any exhaustive key-search by testing all keys K, except for the fact that we omit a key K if we already tested the key $\overline{K}$. So we encrypt the message m with the key K resulting in

$$c' = \mathsf{E}(K,m)$$

If $c' = c_m$, then with a high probability ($\approx 1 - 2^{-8}$) the key K is the correct one. Otherwise, if $c' = \overline{c_{\overline{m}}}$, we exploit part (a) as follows: from $\mathsf{E}(K,m) = \overline{c_{\overline{m}}}$ it follows that $\mathsf{E}(\overline{K},\overline{m}) = c_{\overline{m}}$, thus with a high probability ($\approx 1 - 2^{-8}$) the key $\overline{K}$ is the correct one. The required xor operation is fast compared to the DES operation, so we saved roughly a factor of two.

Problem 2: DESV and DESW

DESV

For breaking DESV, we need at least two plaintext/ciphertext pairs (m_0, c_0) and (m_1, c_1). We mount a modified version of an exhaustive key search using (m_0, c_0): For each key $K \in \{0,1\}^{56}$ we compute $c' = \mathsf{E}(K, m_0)$, consequently we find a $K_1 := c' \oplus c_0$, resulting in a candidate keypair (K, K_1). We need to verify each keypair with the second plaintext/ciphertext pair (m_1, c_1): Computing $c'_1 = E(k, m_1) \oplus K_1$ and then check if $c'_1 = c_1$. Since the unity distance of $\oplus$ is 1, that is there can be only one unique K such that $c = m \oplus K$ holds, we are nearly in the same situation as if we would try to break DES. The only difference is, that we need $2^{56} \cdot 2 = 2^{57}$ DES operations instead of 2^{56}, as for computing and verifying we need two DES operations for each possible key.

If we additionally were in the situation to perform a chosen plaintext attack, we would be able to reduce it to 2^{56} DES operations using the result, we obtained from Problem 1.

DESW

Breaking DESW is very similar: apply the decryption function to both sides of the equation we see that:

$$\mathsf{E}(K, m \oplus K_1) = c \Leftrightarrow \mathsf{D}(K, c) \oplus K_1 = m.$$

Using decryption instead of encryption we can use the same strategy as in part (a).

Problem 3: DES Simplification

We want to perform a universal break on the simplified encryption scheme which is obtained by replacing the DES round functions f_i by

$$f_i = x \oplus K_i$$

using 16 different 32-Bit subkeys $K_1, .., K_{16}$, resulting in a key $\hat{K} = (K_1, .., K_{16})$. The initial permutation and its inverse before and after the Feistel network can be ignored, as it is publicly known and it does not destroy any properties we will look at.

We now just compute the result of the Feistel Network using these simplified f_i. From the construction of a Feistel Network we know, that

$$L_i = R_{i-1} \text{ (i > 0)},$$

hence it suffices to compute the R_i:

$$\begin{aligned}
R_1 &= L_0 \oplus \underbrace{(K_0 \oplus R_0)}_{f_1} \\
R_2 &= L_1 \oplus \underbrace{(K_1 \oplus R_1)}_{f_2} \\
&= R_0 \oplus (K_1 \oplus R_1) \\
&= R_0 \oplus (K_1 \underbrace{L_0 \oplus K_0 \oplus R_0}_{R_1}) \\
&= K_1 \oplus K_0 \oplus L_0 \\
R_3 &= L_2 \oplus \underbrace{(K_2 \oplus R_2)}_{f_3} \\
&= \underbrace{(K_0 \oplus L_0 \oplus R_0)}_{L_2} \oplus (K_2 \oplus \underbrace{K_1 \oplus K_0 \oplus L_0}_{R_2}) \\
&= K_2 \oplus K_1 \oplus R_0 \\
&..
\end{aligned}$$

$$\begin{aligned}
R_{15} &= R_{13} \oplus \underbrace{(K_{15} \oplus R_{14})}_{f_{15}} \\
&= \underbrace{K_{15} \oplus K_{14} \oplus K_{12} \oplus K_{11} \oplus K_9 \oplus K_8 \oplus K_6 \oplus K_5 \oplus K_3 \oplus K_2}_{K'_L} \oplus R_0 \\
&= L_{16} \\
R_{16} &= R_{14} \oplus \underbrace{(K_{16} \oplus R_{15})}_{f_{16}} \\
&= \underbrace{K_{16} \oplus K_{15} \oplus K_{13} \oplus K_{12} \oplus K_{10} \oplus K_9 \oplus K_7 \oplus K_6 \oplus K_4 \oplus K_3 \oplus K_1}_{K'_R} \oplus L_0 \oplus R_0
\end{aligned}$$

Now we can observe, that (ignoring the initial permutation) every ciphertext c $= L_{16}||R_{16}$ has the following form

$$\begin{aligned}
L_{16} &= K'_L \oplus R_0 \\
R_{16} &= K'_R \oplus L_0 \oplus R_0
\end{aligned}$$

for m $= L_0||R_0$. So we can use a single plaintext/ciphertext pair $(m, \mathsf{E}(K, m))$ to compute K'_L and K'_R:

$$\begin{aligned}
K'_L &= L_0 \oplus L_{16} \\
&= R_0 \oplus \ R_0 \oplus K_{15} \oplus K_{14} \oplus K_{12} \oplus K_{11} \oplus K_9 \oplus K_8 \oplus K_6 \oplus K_5 \oplus K_3 \oplus K_2, \\
K'_R &= L_0 \oplus R_0 \oplus L_{16} \\
&= L_0 \oplus R_0 \oplus L_0 \oplus R_0 \oplus K_{16} \oplus K_{15} \oplus K_{13} \oplus K_{12} \oplus K_{10} \oplus K_9 \oplus K_7 \oplus K_6 \oplus K_4 \oplus K_3 \oplus K_1
\end{aligned}$$

Knowing K'_L and K'_R allows us to decrypt any ciphertext by simply computing

$$\begin{aligned}
R_0 &= L_{16} \oplus K'_R, \\
L_0 &= R_{16} \oplus R_0 \oplus K'_L.
\end{aligned}$$

Problem 4: Two-Key Triple DES

We can interpret $\widetilde{E}(K_2,\ \mathrm{m}) = \mathrm{E}(K_2,\ \mathrm{E}(K_2,\ \mathrm{m}))$ as a DES encryption with 32 rounds instead of 16, since the initial permutation cancels out, where the round keys are equal in round i and $i + 16$. This operation needs roughly double times compared to an ordinary DES operation. Now we are almost in the situation of a 2DES meet-in-the-middle-attack:

$$E(K_1, \widetilde{E}(K_2, m)) = E(K_1, E(K_2, E(K_2, m))).$$

We can assume that we do have one plaintext/ciphertext pair (m, c). We brute-force the whole keyspace of DES. We use every $K_i \in \{0,1\}^{56}$, $i \in \{1, .., 2^{56}\}$ to create a table with rows

$$(K_i, \underbrace{\mathsf{D}(K_i, c)}_{=:c_i}).$$

We sort the table with respect to the second entry c_i (in time $2^{56} \log 2^{56}$). Then we start encrypting m with each key $K_j \in \{0,1\}^{56}$, $j \in \{1, .., 2^{56}\}$:

$$\widetilde{E}(K_2, m) =: c'_j.$$

For each encryption c'_j we test if $c'_j = c_i$ for some i, i.e., if this value is already in the table. If we find these we found the right keys with high probability.

This meet-in-the-middle-attack needs roughly

$$3 \cdot 2^{56} \text{ DES operations,}$$

as building the table needs roughly 2^{56} operations, and encrypting m takes roughly $2 \cdot 2^{56}$ operations. We can safely ignore the time needed to sort the table, as DES operations are slow compared to the sorting step. Additionally, a table of size

$$(56 + 64) \cdot 2^{56} \text{ bits}$$

is necessary.