Technical University
of Denmark

01410 Cryptology 1, F21

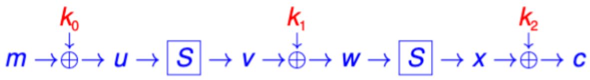
Homework 1

7th March 2021

Christina Juulmann
s170735

Exercise 1.1

In differential cryptanalysis we work on message/ciphertext pairs (ie. $\{(m_n, c_n), (m_{n+1}, c_{n+1})\}$. In this exercise we will find the secret key of CipherTwo (see illustration below), consisting of k_0, k_1, k_2 (4 bits each).



Since the same key is used for all messages at one of each link of the encryption-chain, i.e. k_0 is used on $m_0, m_1, .., m_n$, k_1 on $v_0, v_1, .., v_n$ and k_2 on $x_0, x_1, .., x_n$, by working with pairs we exploit the following property of the binary exclusive or operation ($\oplus$):

$$\begin{aligned} u_0 &= m_0 \oplus k_0 \\ u_1 &= m_1 \oplus k_0 \\ u_0 \oplus u_1 &= (m_0 \oplus k_0) \oplus (m_1 \oplus k_0) = m_0 \oplus m_1 \end{aligned}$$

The (m,c) pairs are given in the exercise as a chosen plaintext attack, where m_0 is chosen randomly and m_1 as the complement of m_0:

$$m_1 = \overline{m_0} = m_0 \oplus \texttt{0xf}$$

Combining this property with the former we compute the input for the first S-box as: $m_0 \oplus m_1 =$ `0xf`.

Since the S-box is non-linear with respect to the $\oplus$ (exclusive or) operation, we cannot directly verify a key guess, thus probability theory is introduced. In the lecture book, table 11.1 shows the computation of the input difference of a message-pair, (i, j), over the S-box for all values of i, where j is chosen as $j = i \oplus$ `0xf`. It shows that for this input difference, `0xf`, the probability of getting $S[i] \oplus S[j] =$ `0xd` is $\frac{10}{16}$. Thus, we use this value (`0xd`) to verify a key guess against.

The following pseudo-code was implemented to find k_2:

Guess a key, t $\in k_2$
for every value of k_2 go through all message/ciphertext pairs
 if $S^{-1}[t \oplus c_i] \oplus S^{-1}[t \oplus \overline{c_i}] ==$ 0xd
 then count[t]++

For the correct key guess there is a high probability ($P = \frac{10}{16}$) of getting `0xd`, while a wrong guess only yields a small probability ($P = \frac{1}{16}$). Thus, looping through our message/cipher pairs, we expect to get a higher count of `0xd`s for the correct guess of k_2. We found $k_2 =$ `0x2`.

Next, we go on and guess k_1 with the found value of k_2, by "boiling down the chain". Now we calculate the w-link of the encryption-chain:

$$\begin{aligned} w_0 &= S^{-1}[k_2 \oplus c_0] \\ w_1 &= S^{-1}[k_2 \oplus c_1] \end{aligned}$$

Which we use the same way as the cipher-pairs and asking the following:

Guess a key, t $\in k_1$
for every value of k_1 go through all message/ciphertext pairs
 if $S^{-1}[t \oplus w_i] \oplus S^{-1}[t \oplus w_{i+1}] == m_i \oplus \overline{m_i}$
 then count[t]++

Once again the correct value of a key guess will have the highest value. This showed out to be $k_1 = \texttt{0x7}$ and $k_1 = \texttt{0xa}$, since both of these values have a count of 3 (see table 2 in appendix B). When computing k_0 we will try with both these values. We compute k_0 by computing u_0 with the found values of k_1 and w_0 and hereafter apply the $\oplus$ operation with m_0:

$$u_0 = S^{-1}[w_0 \oplus k_1]$$
$$k_0 = u_0 \oplus m_0$$

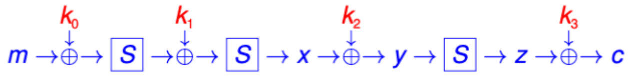
where $k_1 = \{\texttt{0x7}, \texttt{0xa}\}$.

We find that $k_0 = \{\texttt{0x3}, \texttt{0xc}\}$ for the respective values of k_1. We verify the key guesses by applying it to cipherTwo, thus encrypting all the messages and check if the ciphertext match the given set. We found two keys (key sets):

$$K_1 = \{k_0, k_1, k_2\} = \{\texttt{0x3}, \texttt{0x7}, \texttt{0x2}\}$$
$$K_2 = \{k_0, k_1, k_2\} = \{\texttt{0xc}, \texttt{0xa}, \texttt{0x2}\}$$

Exercise 1.2

The approach is much the same for CipherThree as it was for CipherTwo, however in this algorithm, we use the 2-round characteristic $f \to d \to c$ with probability about $\frac{1}{4}$, in order to verify our guess for k_3 due to the extra S-box link:



In the following snippet, we compare key guess for k_3 against `0xc`:

```
for(t=0; t<16; t++){
        for(i=0; i<=8; i+=2){
                idx1 = bitXor(t, ciph[i]);
                idx2 = bitXor(t, ciph[i+1]);

                if( bitXor(R[idx1], R[idx2] ) == 0xc){
                        cnt[t]++;
                }
        }
}
```

Where $R[\cdot]$ is the inverse S-box and *ciph* is an array of the given ciphertexts sorted in pairs of complements, such that $\{c_0, c_{15}\} = \{\texttt{0x0}, \texttt{0xf}\}$, $\{c_1, c_{14}\} = \{\texttt{0x1}, \texttt{0xe}\}$ etc, are pairs.

The expected counter value for the correct guess is $8 \cdot \frac{10}{16} \cdot \frac{6}{16} \approx 2$, since we have eight message/cipher pairs to make out of the given 16 message values. We found $k_3 = \texttt{0x6}$ yielded the highest count; that being 2 (see table 3 in appendix C).

Exercise 1.3

The best 2-round characteristic from an attacker's perspective is that of highest probability. Looking from table 11.2 in the lecture book we see that input difference with `0xf` yields output difference `0xd` with $P = \frac{10}{16}$ which further used as input difference (for second round) yields `0xc` with $P = \frac{6}{16}$ and a total of $P = \frac{6}{16} \cdot \frac{10}{16}$. The same probability is obtained by going $\texttt{0xe} \to \texttt{0xf} \to \texttt{0xd}$ yielding $P = \frac{6}{16} \cdot \frac{10}{16}$.

Appendix A

key value	**count**
0x0	1
0x1	1
0x2	3
0x3	2
0x4	0
0x5	1
0x6	0
0x7	0
0x8	0
0x9	0
0xa	0
0xb	0
0xc	1
0xd	1
0xe	2
0xf	2

Table 1: Key value counts of k_2

Appendix B

key value	**count**
0x0	1
0x1	2
0x2	1
0x3	2
0x4	2
0x5	2
0x6	2
0x7	3
0x8	2
0x9	2
0xa	3
0xb	2
0xc	2
0xd	1
0xe	2
0xf	1

Table 2: Key value counts of k_1

Appendix C

key value	**count**
0x0	0
0x1	0
0x2	0
0x3	1
0x4	0
0x5	1
0x6	2
0x7	0
0x8	0
0x9	0
0xa	0
0xb	1
0xc	0
0xd	0
0xe	1
0xf	0

Table 3: Key value counts of k_3