Assignments 4 Solutions

1. Consider using CBC mode of encryption in the following way: the IV is treated as a key, and is assumed to be known to both Alice and Bob, but no actual encryption function is used. (That is, $E_K(x) = x$ for all $K$ and all $x$.) We will investigate whether this yields any security.

   a. Show a known plaintext total break attack (i.e. one that yields IV) against this kind of cryptosystem.

   **Sol:**
   Known plaintext attack:

   
   Assuming that attackers know the block size.
   each $m_i = c_{i-1} \oplus c_i$, for $i > 1$
   This means that all $m_i$'s except $m_1$ are not hidden by this scheme.
   The secret key only hides $m_1$.
   With a little bit of luck, knowing the plaintext $m_1$ and the corresponding ciphertext $c_1$ suffices to find the key.

   b. Discuss ciphertext-only attacks, both in the case that only one block is given and in the case that $\ell$ blocks are given for some $\ell > 1$

   **Sol:**
   Ciphertext-only attack:
   Attacker just gets the ciphertext.

   If the attacker knows only one block of ciphertext $c_1$:
   The attacker cannot derive any information about $m_1$. Nor can he derive any information about the key. All other message blocks $m_i$ can be derived with the information of two blocks of ciphertexts $c_{i-1}$ and $c_i$:
   $m_i = c_{i-1} \oplus c_i$, for $i > 1$

   If the attacker knows consecutive $\ell$ blocks:
   The attacker can know $\ell-1$ blocks of plaintext $\{m_2, m_3, ..., m_\ell\}$
2. (Trappe page 125, 4.8.4) For a string of bits $S$, let $\overline{S}$ denote the complementary string obtained by changing all the 1’s to 0’s and all the 0’s to 1’s (equivalently, $S = \overline{S} \oplus 11111\ldots111$). Show that if the DES key $k$ encrypts $P$ to $C$, then $\overline{k}$ encrypts $\overline{P}$ to $\overline{C}$.

**Sol:**

From the simple logic of the expander function $E(\cdot)$, it can be noted that

$$E(A) = \overline{E(A)}$$

The following figure shows that $f( R_i, k_{i+1} )$ equals $f( \overline{R_i}, \overline{k_{i+1}} )$

![Diagram showing $f( R_i, k_{i+1} )$ equals $f( \overline{R_i}, \overline{k_{i+1}} )$]

Note that

$$E( \overline{R_i} ) \oplus \overline{k_{i+1}} = E( R_i ) \oplus k_{i+1}$$

$$= 111\ldots111 \oplus E(R_i) \oplus 111\ldots111 \oplus k_{i+1}$$

$$= E(R_i) \oplus k_{i+1}$$

In each DES round:

$$\begin{align*}
L_{i+1} &= R_i \\
R_{i+1} &= L_i \oplus f( R_i, k_{i+1} )
\end{align*}$$

If the input is complemented, i.e. $(\overline{L_i}, \overline{R_i})$ and the key is also complemented, then

$$\begin{align*}
L_{i+1} &= \overline{R_i} = R_i \oplus 11\ldots111 \\
R_{i+1} &= \overline{L_i} \oplus f( \overline{R_i}, \overline{k_{i+1}} ) = L_i \oplus 11\ldots111 \oplus f( R_i, k_{i+1} )
\end{align*}$$

So $\overline{k}$ encrypts $\overline{P}$ to $\overline{C}$

3. (Stinson page 113 3.4) Before the AES was developed, it was suggested to increase the security of DES by using the product cipher $\text{DES} \times \text{DES}$. This product cipher uses two 56-bit keys. Consider known-plaintext attacks on product ciphers. In general, suppose that we take the product of any cipher $S = (P, P, K, E, D)$ with itself. Further, suppose that $K = \{0,1\}^n$ and $P = \{0,1\}^m$.
Now assume we have several plaintext-ciphertext pairs for the product cipher $S^2$, say $(x_1, y_1), \ldots, (x_l, y_l)$, all of which are obtained using the same unknown key, $(K_1, K_2)$.

a. Prove that $e_{K_1}(x_i) = d_{K_2}(y_i)$ for all $i$, $1 \leq i \leq l$. Give a heuristic argument that the expected number of keys $(K_1, K_2)$ such that $e_{K_1}(x_i) = d_{K_2}(y_i)$ for all $i$, $1 \leq i \leq l$, is roughly $2^{2n-m}$.

**Sol:**

for all $i$, $1 \leq i \leq l$, $y_i = e_{K_2}(e_{K_1}(x_i))$

apply the decryption function $d_{K_2}(\cdot)$ on both sides of the above equation, we get $d_{K_2}(y_i) = d_{K_2}(e_{K_2}(e_{K_1}(x_i))) = e_{K_1}(x_i)$

Assume $n \gg m$. For a given key $K_2$, if $e_{K_1}(\cdot)$ can map $x_i$ to the corresponding $d_{K_2}(y_i)$ for all $i$, $1 \leq i \leq l$, at the same time, the key pair $(K_1, K_2)$ is the key of our choice. Assume that $e_{K_1}(\cdot)$ maps in a random manner. For one pair of $x_i$ and $d_{K_2}(y_i)$, there could be roughly $2^n/2^m K_1$ keys that provide the mapping.

For $l$ pairs of $x_i$ and $d_{K_2}(y_i)$, there are roughly $2^n/2^m l K_1$ keys that provide the mapping. Therefore, the rough number of key pairs that satisfy the above $l$ mappings is $2^n \cdot (2^n/2^m) = 2^{2n-m}$

There are $2^n$ possible $K_2$ keys.

b. Assume that $l \geq 2n/m$. A time-memory trade-off can be used to compute the unknown key $(K_1, K_2)$. We compute two lists, each containing $2^n$ items, where each item contains an $l$-tuple of elements of $P$ as well as an element of $K$. If the two lists are sorted, then a common $l$-tuple can be identified by means of a linear search through each of the two lists. Show that this algorithm requires $2^{n+1} \ell + 2^{n+1}$ bits of memory and $\ell 2^{n+1}$ encryptions and/or decryptions.

**Sol:**

The two lists are shown in the following figure. You can sort the lists according to the $\ell m$ bits and match the $\ell m$ bits of each list to find the key pairs $(K_1, K_2)$ that have matched $\{e_{K_1}(x_1), e_{K_1}(x_2), \ldots, e_{K_1}(x_l)\}$ and $\{d_{K_2}(y_1), \ldots, d_{K_2}(y_l)\}$.
\( d_{K_2}(y_2), \ldots, d_{K_2}(y_l) \} \).

This algorithm requires \( 2^n \cdot (\ell m + n) \cdot 2 \) bits of memory. To prepare the two lists, \( 2^n \cdot \ell \) encryptions and \( 2^n \cdot \ell \) decryptions are required.

c. Show that the memory requirement of the attack can be reduced by a factor of \( 2^t \) if the total number of encryptions is increased by a factor of \( 2^t \). (Hint: Break the problem up into \( 2^{2t} \) subcases, each of which is specified by simultaneously fixing \( t \) bits of \( K_1 \) and \( t \) bits of \( K_2 \))

**Sol:**

Break the problem up into \( 2^{2t} \) subcases, each of which is specified by simultaneously fixing \( t \) bits of \( K_1 \) and \( t \) bits of \( K_2 \). In this case, only \( 2^{n-t} \) items are required in each list. However, each \( K_1 \) list is required to compare with \( 2^t \) dynamically calculated \( K_2 \) lists. This algorithm requires \( 2^{n-t} \cdot (\ell m + n) \cdot 2 \) bits of memory. To prepare the \( 2^t \) \( K_1 \) lists along the computation, \( 2^1 \cdot 2^{n-t} \cdot \ell \) (\( = 2^n \cdot \ell \)) encryptions are required. To prepare the \( 2^1 \cdot 2^t \) \( K_2 \) lists along the computation, \( 2^1 \cdot 2^t \cdot 2^{n-t} \cdot \ell \) (\( = 2^{n+t} \cdot \ell \)) decryptions are required, which is \( 2^t \) time more than the original algorithm.