Homework solution 3

Nikita Snetkov / B88482

March 2020

1 Problem 1: "Inverse" CBC

To encrypt a message m consisting of blocks $m_1, ..., m_n$ with key k, pick a random initialization vector iv and then compute $c_1 := E_0(k, m_1) \oplus iv$ and $c_i := E_0(k, m_i) \oplus m_{i-1}$ for $i = 2, ..., n$. Here E_0 is the block cipher. And $E(k, m) := iv \parallel c1 \parallel ... \parallel c_n$ The adversary has intercepted a ciphertext $c = E(k, m)$. He happens to know the last block m_n of m (e.g., because that one is prescribed by the protocol)

1.1 Task A

Explain how the adversary can completely decrypt m. He can make chosen plaintext queries (i.e., he can ask for encryptions of arbitrary message m‘). He cannot make decryption queries.

1.2 Solution

I can divide the message m in blocks of size of encryption blocks. If the key stays always the same and I know the last block m_n, as I adversary I can ask challenger to encrypt just m_n. I will get $c_n‘ = E(k, m_n) \oplus iv$. By definition, $E(k, m) := iv \parallel c1 \parallel ... \parallel c_n$, so I can get iv from zero block of ciphertext. That gives us $e_n = c_n‘ \oplus iv = E(k, m_n)$.

If I take intercepted ciphertext, the last block will be $c_n = E(k, m_n) \oplus m_{n-1} = e_n \oplus m_{n-1}$. Due to the fact I know e_n, I can get m_{n-1} by the formula $m_{n-1} = c_n \oplus e_n$. Knowing previous message block, I can ask challenger to encrypt it, get e_{n-1} and after that m_{n-2}. The whole process could be described by formula (for $j := n, n-1, n-2, ..., 1)$:

1. $e_j = c'_j \oplus iv$
2. $m_{j-1} = c_j \oplus e_j$

1.3 Task B

Suggest how to fix the mode of operation so that it becomes secure at least against this attack (and simple modifications thereof). You do not need to prove security.

1.4 Solution

There is several ways to fix this mode:

1. Use regular CBC.
2. Instead of using the same key k for every block, set of several keys $K := k_1, k_2, k_3...k_n$
3. Change formula for fist block - $c_1 := E_0(k, iv) \oplus E_0(k, m_1)$. For other blocks formula stays the same - $c_i := E_0(k, m_i) \oplus m_{i-1}$ for $i = 2, ..., n$.

2 Problem 2: Breaking ECB

2.1 Task A

Describe an algorithm that finds out (given m_0, m_1, c) whether m_0 or m_1 was encrypted. It should work on "typical" text files. (That is, it should not require, e.g.,one of the text files to contain only spaces or similar.)

2.2 Solution

I would solve this problem implementing such algorithm:

1. Divide both plaintexts (m_0 and m_1) by blocks size of encryption block.
2. Go through plaintext m_0 and find the block that appears the most through the text (b_0) and remember all its positions ($array_0$).
3. Go through plaintext m_1 and find the block that appears the most through the text and remember all its positions, ($array_1$). Important: either $b0$!= $b1$, either ($array0$!= $array1$)
4. Go through ciphertext c and find most common block (b_c) and its positions ($array_c$).
5. If $array_c == array_0$, then $c = E(m_0, k)$. If not, $c = E(m_1, k)$