Assignment – 2

Rahul Chowdary Garigipati – 800765549

1) What is the effect of a single-bit error in the ciphertext when using the CBC, OFB, and CTR modes of operation?

Ans.) For, CBC mode operation: We know that For Encryption, $c_i = E_K(c_{i-1} \oplus m_i)$ and

For Decryption, $m_i = c_{i-1} \oplus D[c_i , K]$.

Using CBC, the first block of ciphertext is decrypted using DES, producing the wrong plaintext block. But recall from how CBC works, in decryption, the first ciphertext block is then XORed with the output of the DES decryption of the second ciphertext block. So, although the second ciphertext block has no errors, the XOR with the first ciphertext block introduces errors in the resulting second block of plaintext. Specifically, the one bit that is in error in the first block of ciphertext, produces an error in the same position in the second block of plaintext. **So, with CBC, a 1-bit error in the first block produces an error in 1 block of plaintext AND a 1-bit error in the second block of plaintext.**

For, OFB mode operation: We know that for Encryption, $c_i = p_i \oplus E_K (E_K(IV))$ and For Decryption, $p_i = c_i \oplus E_K (E_K(IV))$.

Using OFB, an IV or the previous value of DES output is used as input to DES, then the result is XORed with the ciphertext block. So, when the first block of ciphertext is used in the XOR, the one bit in error will produce an error in the corresponding bit of the plaintext. For the next block, only the second block of ciphertext is used in the XOR - it is not dependent on the first (errored block), and hence no errors in the second block of plaintext. **So, with OFB, a 1-bit error in the first block of ciphertext produces a 1-bit error in the first block of plaintext only.**

For, CTR mode operation: We know that For Encryption, $c_i = p_i \oplus E_K (ctr + n)$ and For Decryption, $p_i = c_i \oplus E_K (ctr + n)$.

If there is a single-bit error in the ciphertext, then it will not affect the following ciphertext as it is not connected to the previous mode of encryption, hence no error propagation. And can be considered as a stream cipher rather than a block cipher.

2) In class, we'd seen the stateful variant of CBC mode is IND-CPA insecure. How-ever, the stateful variants of OFB and CTR modes are IND-CPA secure. Write the IND-CPA attack games for the stateful OFB and CTR modes, akin to the one for stateful CBC mode, assuming adversary knows the first IV/nonce. Briefly point out and explain why the attack games fail.

Ans.) IND-CPA attack game for the stateful OFB:

- Adversary picks (m_1^0, m_1^1) and gives it to the challenger.
- m_1^b is used by challenger as m_1 in encrypting M = (m_1, m_2, m_3).
- once the encryption is done, Nonce, c_1, c_2, c_3 are shared with the adversary.
- Adversary picks another m_4 thus: ($m_4 = IV \oplus m_1^0 \oplus E_K (IV)$), given to challenger.
- m_4 is used in M'= (m_4, m_5). First block is $E_K (M') = m_4 \oplus E_K (E_K (IV))$.
- Substituting the value of m_4, $c_4 = IV \oplus m_1^0 \oplus E_K (IV) \oplus E_K (E_K (IV))$.
- It will be hard for the adversary to find the similarity between the c_4, c_1.
- Hence, Therefore the IND-CPA attack game will fail and IND-CPA secure.

IND-CPA attack game for the stateful CTR:

- Adversary picks (m_1^0, m_1^1) and gives it to the challenger.
- m_1^b is used by challenger as m_1 in encrypting M = (m_1, m_2, m_3).
- once the encryption is done, CTR, c_1, c_2, c_3 are shared with the adversary.
- Adversary picks another m_4 thus: ($m_4 = CTR \oplus m_1^0 \oplus E_K (CTR + n)$), given to challenger.
- m_4 is used in M'= (m_4, m_5). First block is $E_K (M') = m_4 \oplus E_K (E_K (CTR + n))$.
- Substituting the value of m_4, $c_4 = CTR \oplus m_1^0 \oplus E_K (CTR + n) \oplus E_K (E_K (CTR + n))$.
- It will be hard for the adversary to find the similarity between the c_4, c_1.
- Hence, Therefore the IND-CPA attack game will fail and IND-CPA secure.

3) Consider a stateful variant of CBC mode, where the sender simply increments the IV by 1 each time a message is encrypted (rather than choosing a random IV every time). In this case, the IVs are distinct, but not random. Write the IND-CPA game, and informally argue why the resulting scheme is IND-CPA insecure. Assume adversary knows first IV.

Ans.) IND-CPA attack game for stateful CBC:

- Adversary picks (m_1^0, m_1^1) and gives it to the challenger.
- m_1^b is used by challenger as m_1 in encrypting M = (m_1, m_2, m_3).
- once the encryption is done, IV, c_1, c_2, c_3 are shared with the adversary.
- Adversary picks another m_4 thus: ($m_4 = IV \oplus m_1^0 \oplus IV + 1$), given to challenger.
- m_4 is used in M'= (m_4, m_5). First block is $E_K (M') = E_K (IV + 1 \oplus m_4)$.
- Substituting the value of m_4, $c_4 = E_K (IV + 1 \oplus IV \oplus m_1^0 \oplus IV +1)$.
- Which gives us $c_4 = E_K (IV \oplus m_1^0)$, which is similar to c_1.
- Easy to verify $m_1 = m_1^0$ if $c_4 = c_1$.
- Hence, Therefore the IND-CPA attack game will not fail and IND-CPA insecure.

4) What is the output of an n-round Feistel network when the input is (L_0, R_0) in each of the following two cases:

(a) Each round function outputs all 0s, regardless of the input.

Ans.) If each round function outputs all 0's, then if n is an even number you get back the original string, i.e., in this case (L_0, R_0). And If n is an odd number you just swap the (L_0, R_0) to (R_0, L_0) components of the input. XOR basically has no effect on the string.

(b) Each round function is the identity function (if f is an identity function, then f(x) = x).

Ans.) If each round function is the identity function, then the effect of a round in the Feistel network on the state is to map the right half to the left half and set the right half to the XOR of the two halves.

So, the first round will transform the input (L_0, R_0) to (R_0, $L_0 \oplus R_0$).

So, now for the second round will transform ($L_0 \oplus R_0$) to ($L_0 \oplus R_0$, $R_0 \oplus L_0 \oplus R_0 = L_0$).

And for the third round will transform $\mathbf{L_0}$ to (L_0, $L_0 \oplus R_0 \oplus L_0 = R_0$) which is the original input.

Accordingly, the output of an n-round Feistel network with this property will be

(L_0, R_0) if n mod 3 = 0,

(L_0, $L_0 \oplus R_0$) if n mod 3 = 1,

and ($L_0 \oplus R_0$, R_0) otherwise.

5) Let $\text{Feistel}_{f1;f2}$(.) denote a 2-round Feistel network using functions f1 and f2 (in that order). Show that if Feistel f1, f2 (L_0; R_0) = (L_2; R_2), then Feistel f2, f1 (R_2; L_2) = (R_0; L_0). (Hint: no formal proof is required, just focus on the Feistel network formulas for computing $\text{Feistel}_{f1;f2}$ (L_0; R_0), and $\text{Feistel}_{f2;f1}$ (R_2; L_2). You can mathematically derive this).

Ans.) We know that in Feistel cipher round function is, **$F(X, K) = X \oplus K$.**

For Encryption, in the first round of Feistel cipher (L_0, R_0) is given to the f1.

Round1: In Feistel cipher $L_1 = R_0$ and $R_1 = L_0 \oplus F1$ (R_0, Key).

Round2: Now, $L_2 = R_1$ and $R_2 = L_1 \oplus F2$ (R_1, Key).

After the end of two rounds of encryption, the output of the Feistel network

f1,f2 (L_0, R_0) is (R_2, R_1) ➔ ($L_1 \oplus F2$ (R_1, Key), $L_0 \oplus F1$ (R_0, Key)).

As we place them in reverse order, they are now (R_1, R_2) →(L_2, R_2).

For Decryption, in the second round of Feistel cipher (R_2, L_2) is given to the f2.

After applying F2 function, $F2 (R_1) \oplus F2 (R_1) \Rightarrow F2 (L_0 \oplus F1 (R_0, Key)) \oplus F2 (L_0 \oplus F1 (R_0, Key)) = 0$.

Same happens after applying F1 function $F1(R_0) \oplus F1 (R_0) = 0$.

Therefore, Feistel f1, f2 (L_0; R_0) = (L_2; R_2), then Feistel f2, f1 (R_2; L_2) = (R_0; L_0).