Assignment 3 سيف الدين ايمن مصطفى ابراهيم

$\boxed{1}$ * The normal Feistel algorithm with n rounds 32 starts encryption using L_0 & R_0 using subkeys $K_1 \longrightarrow K_n$ producing ciphertext R_n & L_n and during decryption it starts with R_n & L_n using $K_n \rightarrow K_1$ to get back L_0 & R_0

* In our case using $K_1 \rightarrow K_n K_n \rightarrow K_1$ is like down encryption then decryption again right after except 1 difference after n rounds we get R_n & L_n but for round n+1 we use L_n & R_n instead of R_n & L_n again to decrypt which then would give R_0 & L_0 at the end instead of the real plaintext L_0 & R_0 where the attacker can easily figure out the plaintext from

* The attacker can also use the cyphertext C which is R_0 & L_0 as a plaintext for the algorithm which like what happened during first time the algorithm will give use L_0 & R_0 which is the original plaintext

$\boxed{2}$ The problem is the first part of cyphertext is always the decryption of a plaintext starting with 0 & second part is of plaintext ending in 1 so in every decryption the attacker knows the effect of key K on 0 which results in first bit of cyphertext and effect of K on 1 which is last bit of the ciphertext which can be used to determine which bits were 0 or 1 in the original plaintext x

Scanned with CamScanner

$\boxed{3}$ * The attacker Knows IV from the last encryption, The attacker can set m_2 as $IV \oplus IV_{+1} \oplus m'$ as the attacker can just increment The IV that will give us c which we can compare with previous c To check if m' is a correct guess To the last m.

① $IV_{+1} \oplus m_2 = IV_{+1} \oplus IV \oplus IV_{+1} \oplus m' = IV \oplus m' = C'$

② $IV \oplus m = C$, if $C == C'$ Then $m = m'$

④ a) CBC because it is completely sequential & C_1 needs To be outputted before generating C_2

OFB although it is sequential but in case of precomputed F(K) then all messages can be decrypted in parallel at same time and in random access

b) ⓘ *in CBC it would result in wrong ciphertext from d_{i+1} till d_n and we can't decrypt it back

*in OFB only affects m_i which will be wrong unless The error was in C_0 which is IV and all plaintext will be affected even if The cipher Text was right

*in CTR would only affect m_i unless the error is in the counter will affect everything just as OFB

ⓘⓘ *CBC, m_i Till m_n can't be encrypted because C_i is needed as key

*OFB, only m_i can't be encrypted unless it is The IV then all msg is lost

* CTR same as OFB only m_i unless it is The counter

5] a) No, it doesn't the key can be longer than block length but also has to be a power of 2 then when block length = 16, AES key length can be 32 and the AES Demo works just fine giving the correct decryption

b) with IV of length 16 & message of length 16 instead of 2 blocks each of length 16 the first representing the IV (C0) & C1 representing the whole message encrypted it creates a new third block instead of having a full block and we new that it uses for padding so during decryption the algorithm recognize that the whole second block is full with messages block and no padding

c) The decrypt function will catch an exception and prints the stack trace, from the cipher.dofinal documentation in java oracle 7 the exception being thrown is BadPaddingException which is thrown whenever the decrypted data is not bounded by the correct padding types which means that padding bytes in ciphertext are altered before decryption.

Assignment 3

Implementation Code:

PaddingOracleAttackSimulation.java