Fall 2023 2023-09-26

Solutions for Assignment 2

Humesh Reddy Venkatapuram *CS 380/525: Intro to Crypto*

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

Answer:

CIPHER BLOCK CHAINING(CBC) MODE: In a CBC mode, a block of ciphertext is decrypted and then XORed with the previous ciphertext block to obtain the plaintext block.

If a single-bit error occurs in a ciphertext block C_i, the corresponding decrypted plaintext block P_i will be corrupted.

The error will also affect the next plaintext block P_{i+1}, but only the bit corresponding to the error in C_i, will be corrupted in P_{i+1}.

Blocks before C_i, and after P_{i+1} will be unaffected.

OUTPUT FEEDBACK(OFB) MODE: In OFB mode, an initial vector is encrypted, and the outcome is XORed with the plaintext to form the ciphertext.

If a single-bit error in a ciphertext block C_i, affects only the corresponding bit in the plaintext block P_i.

Because OFB does not pass the ciphertext back into the cipher for the following block, future blocks are unaffected.

The error cannot propagate beyond the bit where it occurred.

COUNTER (CTR) MODE: CTR mode encrypts a counter value, and the output is XORed with the plaintext to produce the ciphertext. For each subsequent block, the counter is incremented.

If a single-bit error occurs in a ciphertext block C_i, identical to OFB, affects only the corresponding bit in the plaintext block P_i.

The ciphertext is not utilized in the encryption process for future blocks because there is no following blocks are impacted.

The error doesn't propagate beyond the bit where it occurred.

Conclusion: A single-bit error in the ciphertext affects two plaintext blocks in CBC mode, whereas it only affects one plaintext block in OFB and CTR modes, CBC mode is more vulnerable

to error propagation.

2) In class, we'd seen the stateful variant of CBC mode is IND-CPA insecure. However, 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 the adversary knows the first IV/nonce. Briefly point out and explain why the attack games fail.

Answer:

Stateful OFB Mode IND-CPA Attack Game (Adversary knows the initial IV):

The challenger chooses a secret key 'k' at random and establishes an initial IV. The adversary is informed of this IV.

The adversary sends plaintext messages m1,m2,m3,..... to the challenger. For each query, the challenger: Generates a keystream using the current IV and secret key 'k'. Produces the ciphertext by XORing the plaintext with the generated keystream. Updates the IV according to the OFB mode. Returns the corresponding ciphertext to the adversary.

Challenge: After some queries, the adversary sends two equal-length plaintexts, m_0 and m_1, to the challenger. The challenger randomly chooses one plaintext, say m_b. Encrypts m_b using the stateful OFB mode with key 'k' and the current IV, resulting in ciphertext c_b. Send c_b to the adversary.

The adversary attempts to determine whether c_b corresponds to m0 or m1.

Reason for IND-CPA Security: The keystream in OFB mode is derived from the initial IV and the secret key 'k'. Even if the adversary knows the IV, they cannot predict future keystream values without the key. As a result, the XORed ciphertexts remain indistinguishable.

Stateful CTR Mode IND-CPA Attack Game (Adversary knows the initial nonce):

The challenger picks a random secret key 'k' and sets an initial counter value (nonce). The adversary is informed of this nonce. The adversary submits plaintext messages m1,m2,m3,..... For each query, the challenger: Increments the counter. Encrypts the current counter value with the key to generate a keystream. Produces the ciphertext by XORing the plaintext with this keystream. Sends the ciphertext back to the adversary.

Challenge: After several queries, the adversary provides two equal-length plaintexts, m_0 and m_1. The challenger: Selects one plaintext randomly, denoted m_b. Encrypts m_b using stateful CTR mode with the key 'k' and the current counter value, resulting in c_b. Sends c_b to the adversary.

The adversary tries to identify whether c_b originated from m_0 or m_1.

Reason for IND-CPA Security: The keystream is generated in CTR mode by encrypting in-

cremented counter values started with a nonce and combined with the secret key. Understanding the nonce does not provide an adversary an advantage since they cannot derive or predict the keystream without the key. This makes the ciphertexts indistinguishable.

(Unlike in CBC, where the message is encrypted, the initial IV/nonce in OFB and CTR is encrypted with the keystream, which the adversary is unknown.)

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.

Answer:

Game: Stateful CBC Mode (Incrementing IV) IND-CPA:

The challenger initializes the encryption system and picks a confidential encryption key. Simultaneously, the first IV, which is transparent to the adversary, is determined. The adversary selects two distinct plaintext messages, denoted as m_0 and m_1 ensuring they are of equal length. The adversary then provides these messages to the challenger as encryption candidates. The challenger covertly selects one of the two messages via a coin toss; the outcome is represented by $b \in \{0, 1\}$. Depending on this outcome, the challenger encrypts m_b in CBC mode, incrementing the IV. Upon receiving the resulting ciphertext, labeled as c, the adversary, leveraging their knowledge of the incremented IV and the characteristics of c, tries to deduce the value of b to identify the encrypted message.

Argument for IND-CPA Insecurity:

The systematic incrementation of IVs in this version renders them predictable. This allows the adversary, aware of the starting IV, to foresee the IV for any future message. Such predictability emerges as a security flaw. To take advantage of recognizable patterns, the adversary initially submits a plaintext m for encryption and closely examines the resulting ciphertext. In the challenge, the adversary offers the previously known plaintext $m_0 = m$ and another plaintext m_1. Should the ciphertext from the challenge phase strongly resemble or exhibit a discernible pattern akin to the ciphertext from the initial plaintext m, the adversary can with confidence deduce that $b = 0$. Equipped with the knowledge of the IV's predictable progression, the adversary may craftily select pairs of plaintext messages. When encrypted with consecutive IVs, these messages might unveil unique characteristics in their respective ciphertexts. This capacity to differentiate between ciphertexts grants the adversary substantial insight into which plaintext, m_0 or m_1, was encrypted in the challenge.

The strength of the CBC mode concerning IND-CPA security is anchored in the IV's randomness during each encryption process. But in this modified version, where IVs increase sequentially, their randomness is compromised. Given that the adversary is aware of the initial IV, and with each

message the IV advances by one unit, the adversary can foresee the IV for all following messages. Hence, it lacks IND-CPA security.

4) What is the output of an n-round Feistel network when the input is (L0, R0) in each of the following two cases:

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

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

Answer: Let us assume that the input (L0,R0) consists of n-bit blocks. Then we can use the following notations for the each round of the Feistel network.

. $L_{i+1} = R_i$
. $R_{i+1} = L_i \oplus \mathrm{f}(R_i)$ (where $\oplus$ denotes bitwise XOR).

Let us go through each of the cases:

a) If, regardless of the input, the round function f always returns all 0s, then for each round:

. $L_{i+1} = R_i$
. $R_{i+1} = L_i \oplus 0 = L_i$

AFTER 1ST ROUND:

. L1 = R0
. R1 = L0

AFTER 2ND ROUND:

. L2 = R1 = L0
. R2 = L1 = R0

This pattern is going to continue. The result after n rounds would be:

. $L_n = R_{n-1} = L_{n-2} = \ldots\ldots = L_0$ (for even n)
(or)
. $L_n = R_{n-1} = L_{n-2} = \ldots\ldots = R_0$ (for odd n)

. $R_n = L_{n-1} = R_{n-2} = \ldots\ldots = R_0$ (for even n)
(or)
. $R_n = L_{n-1} = R_{n-2} = \ldots\ldots = L_0$ (for odd n)

Therefore, if n is an even, the output is (L0, R0), however, if n is an odd, the output is (R0, L0).

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

. $L_{i+1} = R_i$
. $R_{i+1} = L_i \oplus R_i$

AFTER 1ST ROUND:

. L1 = R0
. R1 = L0 $\oplus$ R0

AFTER 2ND ROUND:

. L2 = R1 = L0 $\oplus$ R0
. R2 = L1 $\oplus$ R1 = R0 $\oplus$ (L0 $\oplus$ R0) = R0 $\oplus$ L0 $\oplus$ R0 = L0

If we go one more round, we will be back at the beginning. However, following the approach described may be computed repeatedly for any number of cycles.

5) Let $Feistel_{f1,f2}$ (·) denote a 2-round Feistel network using functions f1 and f2 (in that order). Show that if $Feistel_{f1,f2}$ (L0, R0) = (L2, R2), then $Feistel_{f1,f2}$ (R2, L2) = (R0, L0). (Hint: no formal proof is required, just focus on the Feistel network formulas for computing $Feistel_{f1,f2}$ (L0, R0), and $Feistel_{f1,f2}$ (R2, L2). You can mathematically derive this).

Answer: To prove that if $Feistel_{f1,f2}$ (L0, R0) = (L2, R2), then $Feistel_{f2,f1}$ (R2, L2) = (R0, L0). Here we can use the Feistel network formulas for computing $Feistel_{f1,f2}$ (L0, R0), and $Feistel_{f2,f1}$ (R2, L2).

Now, let's compute $Feistel_{f1,f2}$(L0,R0) first:

Lets use the 1st round using the f1:
. L1 = R0
. R1 = L0 $\oplus$ f1(R0) (Here $\oplus$ - XOR denotes the bitwise operation)

Now use the 2nd round using f2:
.L2 = R1
.R2 = L1 $\oplus$ f2(R1) = R0 $\oplus$ f2(L0 $\oplus$ f1(R0))

Given that $Feistel_{f1,f2}$(L0,R0) = (L2,R2), we have:
. L2
. R2 = R0 $\oplus$ f2(L0 $\oplus$ f1(R0))

Now lets compute $Feistel_{f2,f1}$(R2,L2):

Lets use the 1st round using f2:
.L'1 = L2
.R'1 = R2 $\oplus$ f2(L2) = R0 $\oplus$ f2(L0 $\oplus$ f1(R0)) $\oplus$ f2(L2)

Given that the L2 was a result of the 2nd round in the 1st network. So, this means L2 = R1.

So: R'1 = R0 $\oplus$ f2(R1) $\oplus$ f2(R1) = R0

Lets use the 2nd round using f1:
.L'2 = R'1
.R'2 = L'1 $\oplus$ f1(R'1) = L2 $\oplus$ f1(R0) = R1 $\oplus$ f1(R0) = L0

Therefore, the $Feistel_{f2,f1}$(R2,L2) = (R0,L0) as per required.

The result of the 2nd round Feistal network using the functions is in the reverse order.

REFERENCES:

1) https://en.wikipedia.org/wiki/Block-cipher-mode-of-operation

2) https://www.geeksforgeeks.org/block-cipher-modes-of-operation/

3) https://en.wikipedia.org/wiki/Ciphertext-indistinguishability

4) https://crypto.stackexchange.com/questions/47646/understanding-the-definition-of-polynomially-cpa-ind-cpa-security

5) https://en.wikipedia.org/wiki/Feistel-cipher

6) https://www.techopedia.com/definition/27121/feistel-network

7) https://crypto.stackexchange.com/questions/58364/what-is-the-output-of-an-r-round-feistel-network

8) https://crypto.stackexchange.com/questions/19642/output-of-a-feistel-network-with-all-0-round-function