Indian Institute of Technology Guwahati
Dept. of Computer Science and Engineering

Mid-Sem Exam 2023 Sub:: **CS549: Computer and Network Security**

Marks: 50 **Date:** 26-02-2023 **Time:** 2 hrs.

Answer all questions in brief. Answer all parts together for each question. No clarification will be entertained.

Question[1] $(\mathbf{4+1})+(\mathbf{4+1})=\mathbf{10}$

Suppose the DES F function mapped every 32-bit input R, regardless of the value of input key K, to

(A) 32-bit string of ones,

(B) bitwise complement of R.

Let the initial permutation, inverse initial permutation, and swapping of left and right parts are represented by IP, IP^{-1}, and $SW(L,R)$ symbols respectively. What function would DES then compute? You can express the DES encryption using the above functions for the 64-bit plaintext input M considering the modified F as stated above.

Question[2] $\mathbf{4+4+2=10}$

(A) What RC4 key value will leave S unchanged during initialization? That is, after the initial permutation of S, the entries of S will be equal to the values from 0 through 255 in ascending order.

(B) RC4 has a secret internal state which is a permutation of all the possible values of the vector S and the two indices i and j. Suppose we think of it from the point of view of how much information is represented by the state. In that case, we need to determine how may different states there are, than take the *log* to the base 2 to find out how many bits of information this represents. Using this approach, how many bits would be needed to represent the state?

(C) What is the one-way property of a hash function?

Question[3] $\mathbf{5+5=10}$

Suppose we have a set of blocks encoded with the RSA algorithm and we don't have the private key. Assume $n=p\times q$, where p and q are prime and $p\neq q$, and e is the public key. Suppose also someone tells us they know one of the plaintext blocks has a common factor with n. Does this help us in any way to do decryption without knowing the private key? Explain using equations of RSA as well as with a numeric example.

Question[4] $\mathbf{5\times 2=10}$

Answer the following questions in brief and to the point. (Max 2-5 lines for each question.)

(A) What is the difference between Cryptanalysis and Brute Force Attack?

(B) What is botnet attack?

(C) CBC is a block cipher mode of operation in which padding is used to assure that the plaintext input is a multiple of the block length. It is assumed that the original plaintext is an integer number of bytes. This plaintext is padded at the end by from 1 to *bb* bytes, where *bb* equals the block size in bytes. The pad bytes are all the same and set to a byte that represents the number of bytes of padding. If the original plaintext is an integer multiple of the block size, can we refrain from padding? Justify your answer in brief.

(D) What is the difference between passive and active attacks? Give at least one example for each type of attacks.

(E) What is CIA triad in Cryptography? Define the meaning of each.

Question[5] $\mathbf{5+5=10}$

(A) Describe a man-in-the-middle attack on the Diffie-Hellman key exchange protocol.

(B) Could the same attack be accomplished with only one pair of public-private keys generated by the adversary? Explain your opinion.

Q. 1

DES f mapped

irrespective of key K { 32-bit R input → 32-bit 1 (a)

→ bitwise complement of R (b)

what function would DES then compute? what

~~(b) what would the description look like~~ ?

Ans: $f(R_n, K_{n+1}) = 1$

(a)

3 marks

we have

$L_{n+1} = R_n$

$R_{n+1} = L_n \oplus f(R_n, K_{n+1}) = L_n \oplus 1 = L_n'$

Thus

$L_{n+2} = R_{n+1} = L_n'$

$R_{n+2} = L_{n+1}' = R_n'$

That is, after each two rounds we obtain the bit complement of the original input, and thus, in every four rounds we obtain the original input.

$L_{n+4} = L_{n+2}' = L_n$

$R_{n+4} = R_{n+2}' = R_n$

1 marks

Therefor, we can conclude that

$L_{16} = L_0$

$R_{16} = R_0$

Now the input to the inverse initial permutation is $R_{16} \| L_{16}$.

1 marks

So, the transformation computed by the modified DES can be represented as follows:

$C = IP^{-1}(SW(\text{Left part} \| \text{Right part of } IP(M)))$

where, $SW(A, B) = (B, A)$

(b) $f(R_n, K_{n+1}) = R_n'$

we have

$L_{n+1} = R_n$

$R_{n+1} = L_n \oplus f(R_n, K_{n+1}) = L_n \oplus R_n'$

properties

$(A \oplus B)' = A \oplus B'$

$A \oplus A = 0$

$A \oplus 0 = A$

$A \oplus 1 = A'$

then,

$L_{n+2} = R_{n+1} = L_n \oplus R_n'$

$R_{n+2} = L_{n+1} \oplus R_{n+1} = R_n \oplus (L_n \oplus R_n')' = R_n \oplus L_n \oplus R_n''$

$= 0 \oplus L_n$

$= L_n$

Then

$L_{n+3} = R_{n+2} = L_n$

$R_{n+3} = L_{n+2} \oplus f(R_{n+2}, K_{n+3})$

$= L_{n+2} \oplus R_{n+2}'$

$= L_n \oplus R_n' \oplus L_n'$

$= 1 \oplus R_n'$

$= R_n$

3 marks

That is After each three rounds, we come back to the original input.

So, $L_{15} = L_0$

$R_{15} = R_0$

And so, $L_{16} = R_0$

$R_{16} = L_0 \oplus R_0'$

1 marks

And finally, the input to IP^{-1} is $R_{16} \| L_{16}$

The transformation computed by the modified DES can be represented as follows:

~~$C = IP^{-1}(SWAP(SF(M)))$ where fun~~

$C = IP^{-1}(SW(L^* \| R))$ where L & R are the left and right part of $IP(M)$ and $L^* = L \oplus R'$

$C = IP^{-1}(SW(L \& R \text{ of } IP(M)))$

where, $L = R_{16}$ and $R = L_{16}$

1 marks

$= IP^{-1}(R_0 \oplus R_0', R_0)$

Q2.

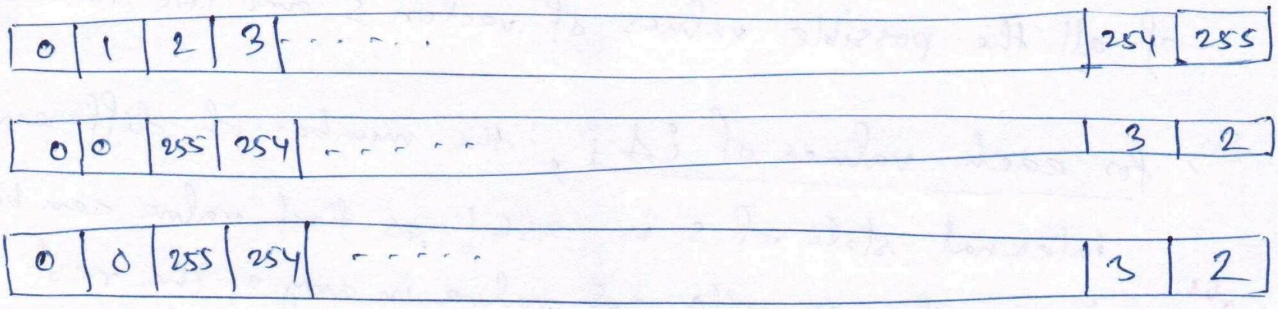
(a) We need to use a key length 255 bytes.

The first two bytes of the key are zero i.e. $K[0] = K[1] = 0$.

2 marks thereafter, we have to put the values of K in decreasing order starting from 255 i.e. $K[2] = 255$, $K[3] = 254$, ... $K[255] = 2$.

Proof:

2 marks Initialization



Initial permutation of S

$j = 0$

↳ $i = 0 \rightarrow j = (j + S[i] + T[i]) \bmod 256$

$= (0 + 0 + 0) \bmod 256$

$= 0$

So, Swap $(S[i], S[j]) \Rightarrow S[0] = 0$

↳ $i = 1 \rightarrow j = (0 + 1 + 0) \bmod 256$

$= 1$

So, Swap $(S[1], S[1]) \Rightarrow S[1] = 1$

↳ $i = 2 \rightarrow j = (1 + 2 + 255) \bmod 256$

$= 2$

So, Swap $(S[2], S[2]) \Rightarrow S[2] = 2$

↳ $i = 3 \rightarrow j = (2 + 3 + 254) \bmod 256$

$= 3$

So, Swap $(S[3], S[3]) \Rightarrow S[3] = 3$

So on

↳ $i = 255, \rightarrow j = 255.$

So Swap $(S[255], S[255]) \Rightarrow S[255] = 255$

So, after initial permutation of S, the entries of S will remain same i.e. from 0 through 255 in ascending order.

(b) ↳ RC4 uses a state vector S. Size of S is 256 bytes. This vector has 256 element or numbers with each number represented by 8 bits. So, the value of each element is between 0 and 255.

↳ RC4 uses 2 indices i & j to point all 256 elements of S.

1 marks So, i & j must be of length 8-bits, as $2^8 = 256$.

↳ RC4 has a secret internal state which is the permutation of all the possible values of vector S and two indices i & j.

So, for each values of i & j, the number of different internal state of S is 256! as first value can be placed in any of 256 cells, 2nd value in any of the rest 255 cells, 3rd value in any of the 254 cells, and so on.

1 marks So, possible number of different states $= 256 \times 255 \times 254 \times \cdots \times 1$

$= 256!$

No, if we consider all possible values of i & j, then the total number of different states of S will be

1 marks $= 256 \times 256 \times (256!)$

So, the no. of bits needed to represent all states

1 marks $= \log_2 (256 \times 256 \times (256!))$ ✓

≈ 1700 ✓

(c) ✓ For a given hash value h, it is computationally infeasible to find y such that $H(y) = h$, where H is the hash function.

2 marks That means, message $\xrightarrow[\text{is easy}]{\text{hash code}}$ hash code, but hash code $\longrightarrow$ message is infeasible

This is the one-way property of a hash function.

Q.5 (1st part)

Let a plaintext block M

using RSA, we get ciphertext block $c = M^e \bmod n$

where, e is the public key, & Let, d is the private key (this is unknown.)

and, $n = p \times q$ where p & q are prime, $p \neq q$ (this is unknown.)

Additional known information: one of the plaintext blocks has a common factor with n. So, if $M = xy$ then $n = xz$

↳ As per RSA, $M < n \Rightarrow M < pq$

↳ Using the additional known information, we can say that either p or q must be a factor of M and M must be a multiple of p or q, respectively.

That is, as $n = pq$, we get either $M = py$ or $M = yq$.

where $0 < y < M$

↳ Now, we can test the primality of each plaintext block and thus the M also.

(4 marks)

↳ If M is prime, then either $M = p$ or $M = q$.

So, we can get the other factor by $\frac{n}{M}$.

So, we have got the values of both the factors of n.

Now, we can compute the private key d as

$$d \equiv e^{-1} \bmod \phi(n)$$

$$\Rightarrow d \equiv e^{-1} \bmod (p-1)(q-1)$$

Hence, decryption is possible.

(2 marks)

↳ If M is not prime, then we can find all factors of M. and find which factor is the divisor of n.

So, that common factor must be either p or q as $n = pq$ and p & q are prime.

One we have one value among the p and q, we can get the other value easily.

Now, we can compute the private key d as shown in previous case.

Hence, decryption is possible.

(2 marks)

Q. 4.

(a) ↳ Cryptanalysis relies on the nature of the encryption algorithm plus some knowledge of the general characteristics of the plaintext.

↳ Brute force Attack tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained.

(b) A malicious bot is self-propagating malware designed to infect a host and connect back to a central server or servers that act as a command and control (c&c) center for an entire network of compromised devices, or botnet.
Attack generated by such botnet is called botnet attack.

(c) The pad bytes are all same and set to a byte that represents the number of bytes of padding.

↳ means, if we need to add 5 byte padding, then the value of each padding byte will be 5 i.e. 00000101.

So, after decryption, the last byte of the last block will give us the information about how many bytes of padding are there.

Now, if we refrain from padding because of any reason, we need to provide ~~addi~~ one additional information whether padding is there or not.

Otherwise, there must be at least one byte of padding ~~wh~~ which will have value 00000000 for the given condition.

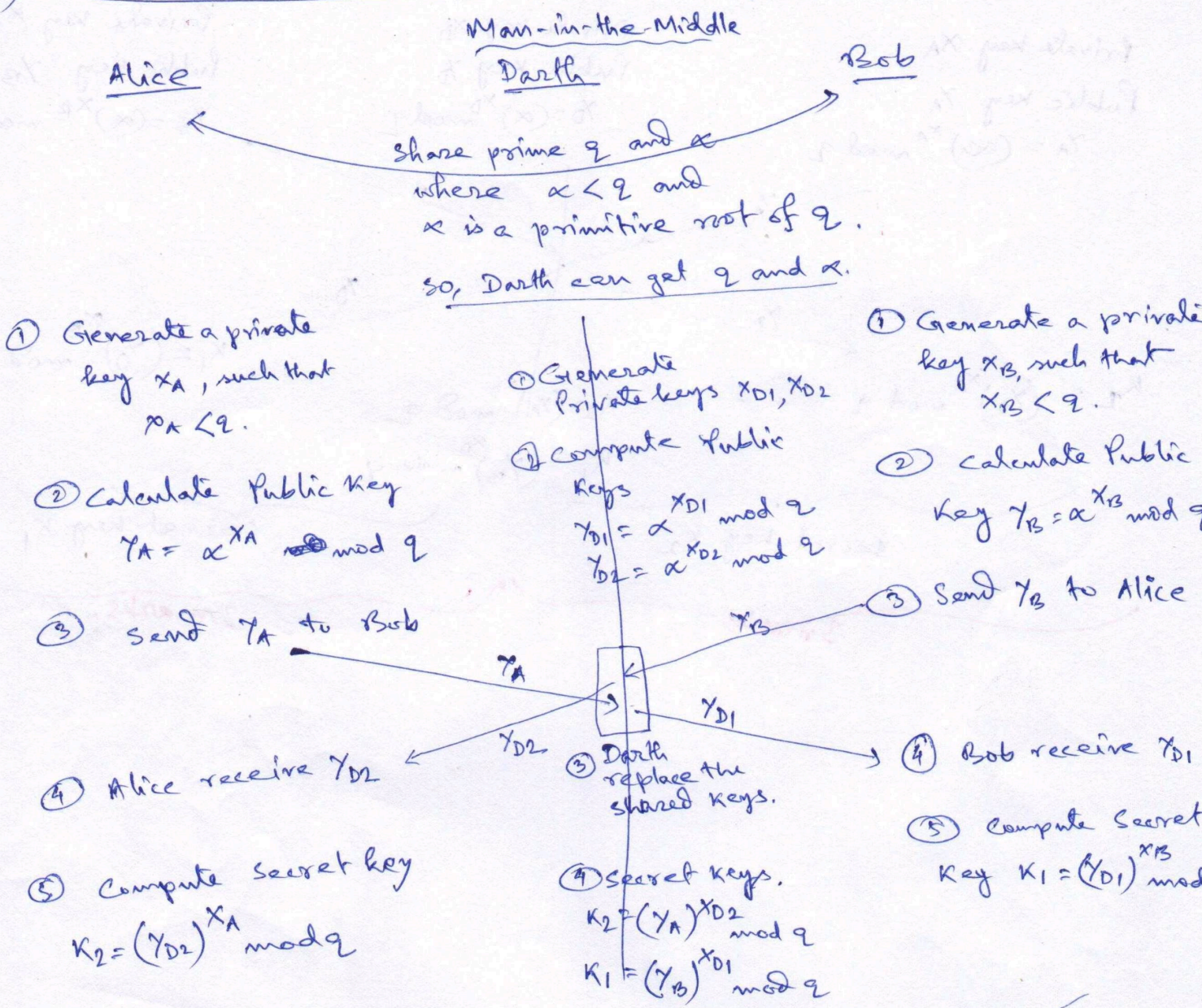
(d) Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted. Eg. traffic analysis, release of content.

Active attack involves some modifications of the data stream or the creation of a false stream. Eg. masquerade, replay, modification, DoS.

(e) C → Confidentiality
I → Integrity
A → Availability.
} Need to write the meaning also.

Q5

(a) # MITM attack in Diffie-Hellman key Exchange



So, Alice & Darth using secret key K_2 — 3 marks

Darth & Bob using secret key K_1 — 2 marks.

So, neither Alice nor Bob knows that Darth is listening their communication using secret keys K_2 and K_1 respectively.

So, this is a Man-in-the-Middle attack in the key exchange algorithm.

(b) Yes, the same attack can be accomplished using one key set by the attacker. Only difference will be $x_{D1} = x_{D2}$ so, $y_{D1} = y_{D2}$, But there will be two keys K_1 and K_2, as it was in previous case.

Alice

Private key x_A

Public key y_A

$y_A = (\alpha)^{x_A} \bmod q$

Darth

Private key x_D

Public key y_D

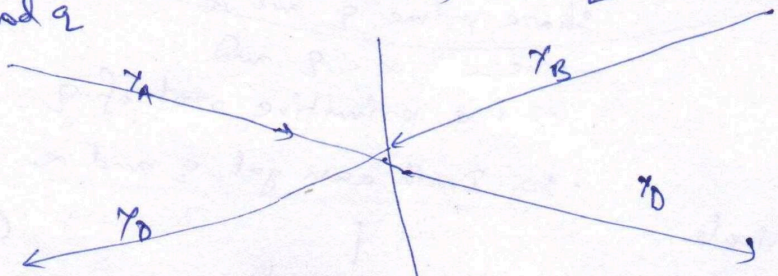
$y_D = (\alpha)^{x_D} \bmod q$

Bob

Private key x_B

Public key y_B

$y_B = (\alpha)^{x_B} \bmod q$



$K_1 = (y_D)^{x_B} \bmod q$

$K_2 = (y_D)^{x_A} \bmod q$

$K_2 = (y_A)^{x_D} \bmod q$

$K_1 = (y_B)^{x_D} \bmod q$

secret key K_2

secret key K_1

3 marks

2 marks.