CS3002 Information Security

Tuesday January 11, 2022

Course Instructor

Dr. Irfan ul Haq, Dr. Danish Shehzad and Dr. Umar Aftab

Serial No:

Final Term

Subjective Part

Total Time:2 Hrs 15 Min

Total Marks: 100

_____

Signature of Invigilator

Roll No	Section	Signature
_____	_____	_____

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Verify at the start of the exam that you have a total of Five (5) **Subjective** questions printed on Eight (08) pages including this title page.
2. Attempt all questions on the question-book and in the given order.
3. The exam is closed book and closed notes. No electronic device (laptop, mobile devices) allowed that may provide internet facility. The use of such devices will be considered as cheating
4. Read the questions carefully for clarity of context and understanding of meaning and make assumptions wherever required, for neither the invigilator will address your queries, nor the teacher/examiner will come to the examination hall for any assistance.
5. Fit in all your answers in the provided space. You may use extra space on the last page if required. If you do so, clearly mark question/part number on that page to avoid confusion.
6. Use only your own stationery and calculator. If you do not have your own calculator, use manual calculations.
7. Use only permanent ink-pens. Only the questions attempted with permanent ink-pens will be considered. Any part of paper done in lead pencil cannot be claimed for checking/rechecking.

	Q-1	Q-2	Q-3	Q-4	Q-5	Q-6	Total
Total Marks	40	24	10	10	8	8	**100**
Marks Obtained							

Vetted By: _____ **Vetter Signature:** _____

University Answer Sheet Required: **No** ☐ **Yes** ☐

Question 2	24 Marks

1. Recall the concepts of DES. Find out the output as you have the input to S-box is 100011. S-box is given below for your reference. [3 marks]

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	10	03	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

Solution:

If we write the first and the sixth bits together, we get 11 in binary, which is 3 in decimal. The remaining bits are 0001 in binary, which is 1 in decimal. We look for the value in row 3, column 1, in table. The result is 12 in decimal, which in binary is 1100. So the input 100011 yields the output **1100**.

2. Recall the concepts of Hill cipher and find out the inverse of the given key $\begin{bmatrix} 2 & 3 \\ 3 & 4 \end{bmatrix}$. [4 marks]

Hill Cipher (Decryption)

1. Find out inverse matrix of given key matrix.

➔ $K^{-1} = \frac{1}{|K|} * K_{adj}$

➔ $|K| = \begin{vmatrix} 2 & 3 \\ 3 & 4 \end{vmatrix} = 8 - 9 = -1$

➔ $K_{adj} = \begin{bmatrix} 4 & -3 \\ -3 & 2 \end{bmatrix}$

➔ $K^{-1} = \frac{1}{|K|} * K_{adj} = \frac{1}{-1} * \begin{bmatrix} 4 & -3 \\ -3 & 2 \end{bmatrix} = \begin{bmatrix} -4 & 3 \\ 3 & -2 \end{bmatrix}$

3. What is the concept of parity drop? Is the shift left same for all rounds? (Justify your answer) [3 marks]

Parity Drop

The preprocess before key expansion is a compression permutation that we call **parity bit drop.** It drops the parity bits (bits 8, 16, 24, 32, …, 64) from the 64-bit key and permutes the rest of the bits according to Table 6.12. The remaining 56-bit value is the actual cipher key which is used to generate round keys. The parity drop permutation (a compression P-box) is shown in Table 6.12.

After the straight permutation, the key is divided into two 28-bit parts. Each part is shifted left (circular shift) one or two bits. In rounds 1, 2, 9, and 16, shifting is one bit; in the other rounds, it is two bits. The two parts are then combined to form a 56-bit part. Table 6.13 shows the number of shifts for each round.

Table 6.13 *Number of bit shifts*

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bit shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Shifting

Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits

4. Recall the concepts of AES, List down all the possible key sizes and their corresponding number of rounds? [3 marks]

Solution

- Key size: 128, 192, 256 bits
  - Rounds: 10, 12, 14 (depending on key)

5. Suppose you have the plaintext “Going for Midterm2” and block size is 5. Prepare the plaintext blocks using electronic codebook mode. [3 marks]

Going	formi	dterm	2xxxx

6. Find out all the possible primitive roots of 7. Justify you answer by using traditional primitive roots calculation method. (3 marks).

3 , 5

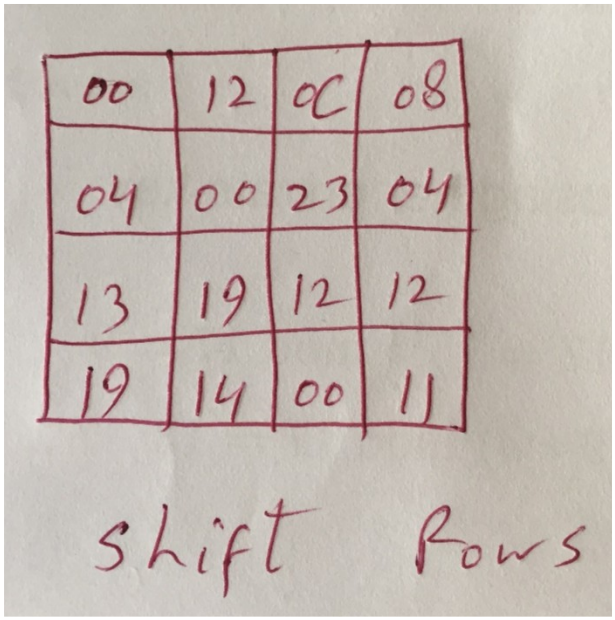
7. Recall the concepts of AES, apply sub-byte and shift rows operations on the given message state (respectively). The sub-byte table is also provided for your reference. [4 marks]

$$\begin{bmatrix} 63 & C9 & FE & 30 \\ F2 & F2 & 63 & 26 \\ C9 & C9 & 7D & D4 \\ FA & 63 & 82 & D4 \end{bmatrix}$$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

Solution:
Sub-byte answer

$$\begin{bmatrix} 00 & 12 & 0C & 08 \\ 04 & 04 & 00 & 23 \\ 12 & 12 & 13 & 19 \\ 14 & 00 & 11 & 19 \end{bmatrix}$$



8. Design the key by using keyword "KEYWORD" by using Playfair Cipher. [3 marks]

Answer:

K	E	Y	W	O
R	D	A	B	C
F	G	H	I J	L
M	N	P	Q	S
T	U	V	X	Z

Question 3	10 marks

In a Diffie-Hellman Key Exchange, Alice and Bob have chosen prime value q = 17 and primitive root = 5. If Alice's private key is 4 and Bob's private key is 6, what is the secret key they exchanged?
Secret Key (K) = _____

Space for Main Steps and Values	

Solution-

Given-

- n = 17
- a = 5
- Private key of Alice = 4
- Private key of Bob = 6

Step-01:

Both Alice and Bob calculate the value of their public key and exchange with each other.

Public key of Alice

$= 5^{\text{private key of Alice}} \bmod 17$

$= 5^4 \bmod 17$

$= 13$

Public key of Bob

$= 5^{\text{private key of Bob}} \bmod 17$

$= 5^6 \bmod 17$

$= 2$

Step-02:

Both the parties calculate the value of secret key at their respective side.

Secret key obtained by Alice

$= 2^{\text{private key of Alice}} \bmod 7$

$= 2^4 \bmod 17$

$= 16$

Secret key obtained by Bob

$= 13^{\text{private key of Bob}} \bmod 7$

$= 13^6 \bmod 17$

$= 16$

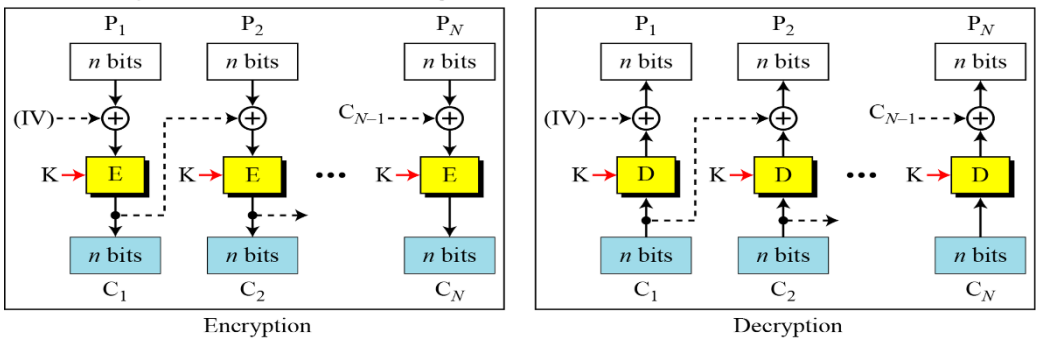
Finally, both the parties obtain the same value of secret key.

The value of common secret key = 16.

Question 4	10 marks

1. When Alice communicates with Bob, she encrypts her messages using the AES algorithm with the CBC mode. For each of her message, she randomly generates a new IV. She knows that the IV should be sent to Bob in the plaintext, or Bob will not be able to know the IV. However, when encrypting her message, Alice decides to prepend the IV to the beginning of her message, and then encrypt the combined message. She thinks this will not cause any harm, and it may bring some benefits. Do you agree with her? Is this practice safe? [5 marks]

E: Encryption D : Decryption
P_i: Plaintext block i C_i : Ciphertext block i
K: Secret key IV: Initial vector (C_0)



Solution

Let's assume that Alice's message will be divided in 'n' blocks, i.e., $B_0, B_1, B_2, \ldots, B_n$.

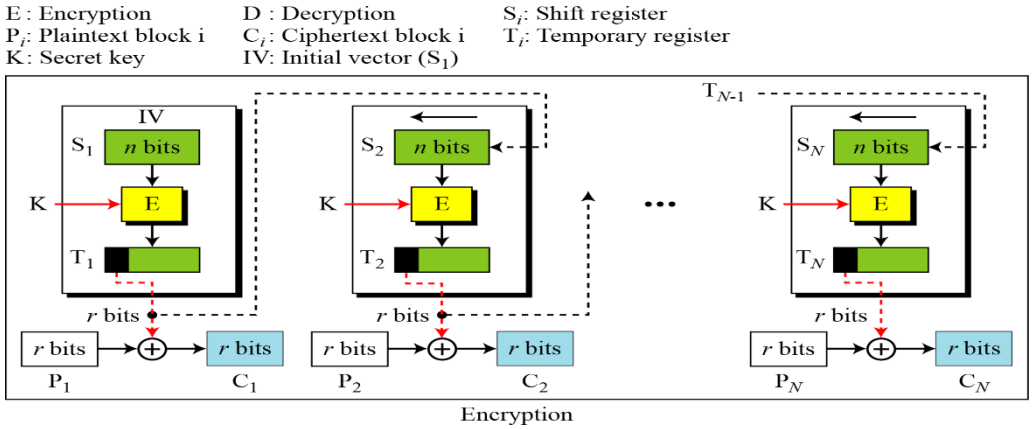
Once Alice has prepended the IV with the original message then each block of original message will be shifted to the right. A new block, B_{iv}, will be added at the start of the original message. The message will look like following: $B_{iv}, B_0, B_1, B_2, \ldots, B_n$

CBC encrypts first block of plain text by XORing with the IV. As the first block in the updated message is an IV, and XOR will be performed between B_{iv} the IV. The output will be 0. This will remain true regardless of the fact that different IVs are used for each new message. In other words, by prepending the IV with the original message, encryption function will be applied on the block filled with 0s. Thus, it is not a safe approach.

2. Alice uses the OFB mode to encrypt her emails sent to Bob, but instead of using randomly generated IVs, she always uses the same IV to encrypt her emails. Charlie somehow gets the following data:

• The first 8 bits of the first ciphertext block: 0x11010101
• The first 8 bits of the first plaintext block: 0x01111001
• The first 8 bits of the second ciphertext block: 0x10100100
• The first 8 bits of the second plaintext block: 0x11001010

Charlie also gets a copy of Alice's newly encrypted message, the first 8 bits of the first two blocks are 0x01010010 and 0x01101001, respectively. Please derive the first 8 bits of the first two blocks of the plaintext. [5 marks]



Solution

Let's assume that
C_1 = 1101 0101,
P_1 = 0111 1001,
C_2 = 1010 0100,
P_2 = 1100 1010

The blocks of newly encrypted message are
B_1 = 0101 0010
B_2 = 0110 1001

From the diagram, we can observe that to decrypt the cipher text we need r-bits because r-bits are XORed with plain text to obtain cipher text. The fact that Alice uses the same IV indicates that r-bits will be same for all the new messages. So, we can obtain r-bits by XORing P_1 and C_1 for the first block. Similarly, we can obtain r-bits for the second block by XORing P_2 and C_2.

P_1 XOR C_1 = 0111 1001 XOR
1101 0101
=======
1010 1100 → R_1 (r-bits for first block)

P_2 XOR C_2 = 1100 1010 XOR
1010 0100
========
0110 1110 → R_2 (r-bits for second block

B_1 XOR R_1 = 0101 0010 XOR
1010 1100
==========
1111 1110 → Plain text of first block.

B_2 XOR R_2 = 0110 1001 XOR
0110 1110
==========
0000 0111 → Plain text of second block.

Question 5	8 Marks

1. How do we prevent from XSS attacks? Enlist and explain any three countermeasures including CSP. [4 marks]

Solution

- **Apply Filter on Data Extract Code :**
  Different filters can be apply on data coming from users to separate data and code.

  Samy's strategy:

  Filter 1: Myspace filter block tags i.e., "<script>" , "<head>", "<body>" etc

- **Turning Code into Data**
  2nd solution is that turn code into data with help of special characters. So full dictionary of special characters is required to convert code into data.

PHP **htmlspecialchars()** function

Character	Replacement
& (ampersand)	&amp;
" (double quote)	", unless **ENT_NOQUOTES** is set
' (single quote)	' (for **ENT_HTML401**) or '
< (less than)	&lt;
> (greater than)	&gt;

- **Treating data only as data**

  3rd solution is that use iframe as functionality of sandboxing. Actually, this solution says to browser that treat data=[data+code] as data even it have code.

  - Setting Sandbox

```
<iframe sandbox="<list of options>" src="https://www.example.com">
<iframe sandbox="<list of options>" srcdoc="<HTML content>">
```

- **Content security policy:**

  The source of linked approach must be labelled, because code may be coming from some other resource. One source is labelled, Then CSP privilege to code.

First Solution:

UPDATE credential

SET nickname=’ ‘ , salary=’90000‘[1] , email=’ ‘, address=’ ‘, number=’ ‘, password=’ ‘

Where ID = 12345;

Write down your solution.

1 salary field can be put anywhere after “SET” and before “Where ID”.

2nd solution

2. The following SQL statement is sent to the database to update user profile. i.e. nickname, email, address, phone number, password. How can a malicious employee update his/her salary to a value higher than 80000?

   UPDATE credential
   SET nickname=' ' , email=' ', address=' ', number=' ', password=' '
   Where ID = 12345;

Write down your solution. [4 marks]

First Solution:

UPDATE credential

SET nickname=' ' , salary='90000'[1] , email=' ', address=' ', number=' ', password=' '

Where ID = 12345;

Write down your solution.

1 salary field can be put anywhere after "SET" and before "Where ID".

2nd solution

Question 6	8 marks

Consider the following program:

```
void foo(char *args)
{
int x = 0;
char buf[16];
int y = 1;
int z = 2;
strcpy(buf, args, 24);
}
int main(int argc, char *argv[])
{
foo(argv[1]);
return 0;
}
```

1. Assuming a 32-bit x86 architecture and a stack frame with space allocated only for the variables shown, which of the variables (if any) can be overwritten by strncpy? [2 marks]
2. How many bytes of the saved frame pointer can be overwritten? [3 marks]
3. How many bytes of the return instruction pointer can be overwritten? [3 marks]

Solution:

(a) Only x can be overwritten because the stack is written upwards from the location of buf.

(b) We can copy a total of 24 bytes into buf. 16 are allocated for the buffer; 4 for x; the remaining 4 are the saved frame pointer. So, 4 bytes.

(c) 0 bytes