WQD7010
Network & Security

Tutorial 1
Introduction to Network Security & Symmetric Encryption (10 Marks)

Problems:

1. A desktop publishing system is utilized for creating documents across different organizations.

   i. Identify a publication type where the utmost priority is the confidentiality of data maintained within the system. (0.5)
   ii. Mention a publication type where ensuring the integrity of data is the primary concern. (0.5)
   iii. Highlight a scenario in which the most crucial aspect is the system's availability for document production. (0.5)

2. You are a senior IT executive at your department responsible for the DES encryption key management. From the random key generator, you are given the secret key (E0E0E0E0F1F1F1F1)hex to encrypt a critical message.

   i. What is is the output of the key after it is applied with the parity bit drop process shown in Figure 1? At this point, determine if it is a strong, weak, or semi-weak key. Provide a detailed explanation for your assessment. (1.25)
   ii. Using the key-permutation compression 2 from Figure 2, re-evaluate whether this is a strong, weak or semi-weak key. Justify your answer thoroughly and explain the implications of using this key for encrypting the critical message. (1.25)

57	49	41	33	25	17	09	01
58	50	42	34	26	18	10	02
59	51	43	35	27	19	11	03
60	52	44	36	63	55	47	39
31	23	15	07	62	54	46	38
30	22	14	06	61	53	45	37
29	21	13	05	28	20	12	04

Figure 1: Parity bit drop

14	17	11	24	01	05	03	28
15	06	21	10	23	19	12	04
26	08	16	07	27	20	13	02
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

Figure 2: Key-Permutation Compression 2

3. Let M be the plaintext message M = MALAYSIA and the private key, K = ISSOGOOD (this question is lenghthy and you can used tools or programming language to solve this, but show some work in your solutions how you obtain the output)

   i. Convert this plaintext and key message to Hexadecimal representation using the ASCII table. (0.25)
   ii. Convert the Hexadecimal representation to binary number. (0.25)
   iii. Represent your plaintext binary number as a form of a block, using the DES encryption technique (2 equal blocks). (0.25)
   iv. Using the initial permutation (table is provided below), please provide your plaintext output. (0.25)
   v. Using the expansion P-box (table is provided below), show the plaintext output of this process. (0.25)
   vi. For the key: Using the key permutation table (provided below), show the output of this conversion process. (0.25)
   vii. Using the appropriate steps of DES operation, what is the final output of the key for round 1 (hint: you have to use the circular left shift and permutation compression (PC) table provided below). (0.5)
   viii. For round 1, please provide your output for the XOR operation of plaintext and the key, according to the DES operation. (0.5)
   ix. Based on the output of 4(h), explain the next process using the S-box table, and show the output from this operation. (0.25)
   x. Based on the output of 4(i), explain the next process using a straight permutation table, and show the output from this operation. (0.25)

4. Consider a scenario where an attacker intercepts a ciphertext C, which is the result of encrypting a fund transfer message M using a stream cipher with key K. The attacker knows that bytes 37-42 of the message contain the amount to be transferred, initially set to 500 MYR. The attacker wants to modify the ciphertext to change the transfer amount to 500,000 MYR without knowing the entire key K.

   i. Explain how the attacker can modify the ciphertext C to create a new ciphertext C1 that decrypts to a message with the desired transfer amount of 500,000 MYR. (0.5)
   ii. Provide a mathematical proof to show that the recipient will decrypt the modified ciphertext C1 to obtain the new transfer amount of 500,000 MYR. (0.5)
   iii. Discuss how this scenario demonstrates the failure of stream ciphers in protecting message integrity.(0.5)

5. Using the simplified RC4 Algorithm of 8 bytes (rather than the full 256 bytes), find the final state of the S-Array, Keystream, and Ciphertext for the following inputs:

   Initial S-Array, [S0 S1 S2 S3 S4 S5 S6 S7] = [7 2 5 1 0 3 6 4]

   Key, K = [2 0 1]

   Plaintext, PT = [5 3 1 6]

Hints:

- Perform the initial permutation of the S-Array using the given Key.
- For the simplified stream generation, use this pseudocode (only up to the PT length):

```
/* Initialization */ set i and j = 0
for i = i + 1 to length of PT
      j = j + S[i] mod 8
      swap S[i], S[j]
      t = S[i] + S[j] mod 8
      Keystream = S[t]
end for
```

- To obtain the Ciphertext, perform a bitwise XOR operation between the Plaintext and the generated Keystream.

  Your task is to:

  i. Show the state of the S-Array after the initial permutation. (0.25)
  ii. Generate the Keystream using the simplified stream generation process. (0.25)
  iii. Calculate the Ciphertext by XORing the Plaintext with the Keystream. (0.5)

  Provide the final state of the S-Array, the Keystream, and the Ciphertext as your answer.

6. Would it be possible to execute encryption functions in parallel on CBC mode with multiple blocks of messages? How about Decryption? (0.25)

7. How far does the error propagate if a bit error occurs in the transmission of a ciphertext character in 8-bit CFB mode? (0.25)

Appendix:

ASCII Table

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Initial Permutation

Initial Permutation							
58	50	42	34	26	18	10	02
60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06
64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01
59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05
63	55	47	39	31	23	15	07

Expansion P-box

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

Key Permutation Table

57	49	41	33	25	17	09	01
58	50	42	34	26	18	10	02
59	51	43	35	27	19	11	03
60	52	44	36	63	55	47	39
31	23	15	07	62	54	46	38
30	22	14	06	61	53	45	37
29	21	13	05	28	20	12	04

Key Permutation-Compression Table

14	17	11	24	01	05	03	28
15	06	21	10	23	19	12	04
26	08	16	07	27	20	13	02
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

S-Box Table

S1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Straight Permutation Table

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25