

Aria Javani

9725303

1.3.1 :

Table 3.4 S-box S_1

S_1	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	01	10	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

Calculating the S_1 for numbers using the above table

1.

$$S_1(000000) \oplus S_1(000001) = 14 \oplus 0 = 14$$

$$S_1(000000 \oplus 000001) = 0$$

2.

$$S_1(111111) \oplus S_1(100000) = 13 \oplus 4 = 1$$

$$S_1(111111 \oplus 100000) = 8$$

3.

$$S_1(101010) \oplus S_1(010101) = 6 \oplus 9 = 15$$

$$S_1(101010 \oplus 010101) = 13$$

1.3.3 :

we proceed step by step :

initial permutation : after this part text is still all zero

splitting into two parts : first 32 bits remain untouched and rest of the bits enter next phase.

calculating f function : first we have to expand our 32 bit input into 48 after this operation we have 48 bits of zeros. after that we do the binary summation on this generated bits and deprived key which leaves us with 48 bits zeros again. in the next step we have to divide these 48 bits by 8 and send each group to one of the S boxes then combine the outputs. output after this part will be 1110,1111,1010,0111,0010,1100,0100,1101 then we have to permute this output using the below table

<i>P</i>							
16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

result : 1111,0101,0111,1100,0110,1001,1110,0010

final operation :

in this part first we xor the output of f with left part of the input then swap two separated parts. the output will be

0000,0000,0000,0000,0000,0000,0000,0000,
1111,0101,0111,1100,0110,1001,1110,0010

1.3.5 :

after initial permutation the new position of our 1 bit will be 33th bit so it takes place in the second half of the whole text. in the next step the

text will be expanded which causes us to have 1 on the second and 48th bits so

1. two S boxes(1 and 8) will get different inputs
2. 3 for the first 6 bits and 2 for the last 6 bits → 5bits total
3. 0011,1111,1010,0111,0010,1100,0100,0001 after s boxes →
1011,0101,0010,1100,0010,1001,1111,0010 after permutation
4. due to the previous question 5bits

1.3.7 :

1. all subkeys are identical when the key is weak.
- 2.

$$k_{i+1} = k_{16-i}$$

- alternating ones+zeros(exp: 0x0101010101010101)
- alternating F+E(exp: 0xFEFEFEFEFEFEFEFEFE)
- 0x(E0E0E0E0F1F1F1 F1)
- 0x(1F1F1F1F0E0E0E0E)

$$3. \frac{4}{2^{56}} = \frac{1}{2^{54}}$$

1.4.9 :

8b8a8a8a747575758b8a8a8a74757575

Round 1

input to Round 1

00000000 00000000 00000000 00000000

after S-Box:

63636363 63636363 63636363 63636363

ON

after permutation:

63636363 63636363 63636363 63636363

ON

used subkey:

e8e9e9e9 17161616 e8e9e9e9 17161616

after mix with key:

8b8a8a8a 74757575 8b8a8a8a 74757575

ON

2:

- block ciphers break down the plain text into individual blocks then cipher those blocks bit by bit but stream ciphers cipher the whole plain text bit by bit.
- to add the confusion feature.
- since all the process is consist of either permutation which is reversible and xor which we have the $a \oplus b = c \rightarrow a = b \oplus c$ then des decryption is indeed reverse of des encryption.
- AES is stronger against all kind of attacks since has both longer key and more complicated operations.

Parameters of Comparison	Des	Aes
Abbreviation	DES is the acronym of Data Encryption Standard.	AES is the acronym of Advanced Encryption Standard.
Creation	It was created in the year 1976.	It was created in the year 1999.
Key length	The key length of DES is 56 bits.	The key length is 128 bits, 192 bits, and 256 bits.
Security	It is not so secure and can be broken easily.	It is much more secure.
Encryption	It can encrypt up to 64 bits of just plain text.	It can encrypt up to 128 bits of just plain text.

- e.
- i) Add RoundKey
 - ii) Add RoundKey
 - iii) SubBytes
 - iv) MixColumns
 - v) ShiftRows

3.1 : key and text : 0000 0001 0010 0011 0100 0101 0110 0111 1000
1001 1010 1011 1100 1101 1110 1111

**attention : we didn't use initial and final permutations since question hasn't mentioned it.

<i>PC - 1</i>							
57	49	41	33	25	17	9	1
58	50	42	34	26	18	10	2
59	51	43	35	27	19	11	3
60	52	44	36	63	55	47	39
31	23	15	7	62	54	46	38
30	22	14	6	61	53	45	37
29	21	13	5	28	20	12	4

a.

key after PC-1 → 1111 0000 1100 1100 1010 1010 0000 0000 1010 1100 1100 1111 0000 0000

here we halve the above bits and rotate left each half by one

key after rotation → 1110 0001 1001 1001 0101 0100 0001 0001 0101 1001 1001 1110 0000 0000

<i>PC - 2</i>							
14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

next using PC-2

result → 0000 1011 0000 0010 0110 0111 1001 1001 0100 1000 1010 0101

b. L0=0000 0001 0010 0011 0100 0101 0110 0111 R0=1000 1001 1010 1011 1100 1101 1110 1111

c. $E(R_0) = 1100\ 0101\ 0011\ 1101\ 0101\ 0111\ 1110\ 0101\ 1011\ 1111\ 0101\ 1111$

d. $E(R_0) \oplus k = 1100\ 1110\ 0011\ 1111\ 0011\ 0000\ 0111\ 1100\ 1111\ 0111\ 1111\ 1010$

e. passing through s boxes \rightarrow 1:1011 2:1000 3:1110 4:1111 5:0110 6:0101 7:0110 8:0011

f. 10111000111011110110010101100011

g. applying permutation table \rightarrow 1000 0000 1101 1111 0011 1111 1100 1110

h. $R_1 \rightarrow$ 1000 0001 1111 1100 0111 1010 1010 1001

i. final ciphered text \rightarrow 1000 1001 1010 1011 1100 1101 1110 1111 1000 0001 1111 1100 0111 1010 1010 1001

4 :

all of $GF(2^4)$ irreducible polynomials

$$p(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

since $m=4$ the a_4 should be 1 and if also a_0 should be 1 otherwise it'd be reducible.

for the rest of the coefficients $a_3 + a_2 + a_1 \neq 0$

final answer:

$$x^4 + x^3 + x^2 + x + 1$$

$$x^4 + x + 1$$

$$x^4 + x^3 + 1$$

5.1 :

ciphered text : j5a0edj2b

sample prefix text : WPI

palin text : WPIWOMBAT

5.2 :

initialization vector : 1,1,1,1,1,1

5.3 :

[1,1,0,0,0,0]

(0,1,6)

5.4 :

The common wombat lives mainly in wet, partly forested areas on the coast, and on the ranges and western slopes.

5.5 :

Known-plaintext Attack

** the python program is attached as 5_1_LFSR_decoding.py

6 :

like 1.3.7 since this key is a weak key, DES becomes a Feistel network and encryption becomes self-inverting so with twice DES encryption using this key we get the plain text itself.

7 :

a. text

00	04	08	0C
01	05	09	0D
02	06	0A	0E

03	07	0B	0F
----	----	----	----

key

01	01	01	01
01	01	01	01
01	01	01	01
01	01	01	01

b.

01	05	09	0D
00	04	08	0C
03	07	0B	0F
02	06	0A	0E

c.

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

7C	6B	01	D7
----	----	----	----

63	F2	30	FE
7B	C5	2B	76
77	6F	67	AB

d.

B_0	B_4	B_8	B_{12}	no shift
B_{13}	B_1	B_5	B_9	← three positions left shift
B_{10}	B_{14}	B_2	B_6	← two positions left shift
B_7	B_{11}	B_{15}	B_3	← one position left shift

7C	6B	01	D7
F2	30	FE	63
2B	76	7B	C5
AB	77	6F	67

e.

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{pmatrix}$$

Table 4.2 Multiplicative inverse table in $GF(2^8)$ for bytes xy used within the AES S-Box

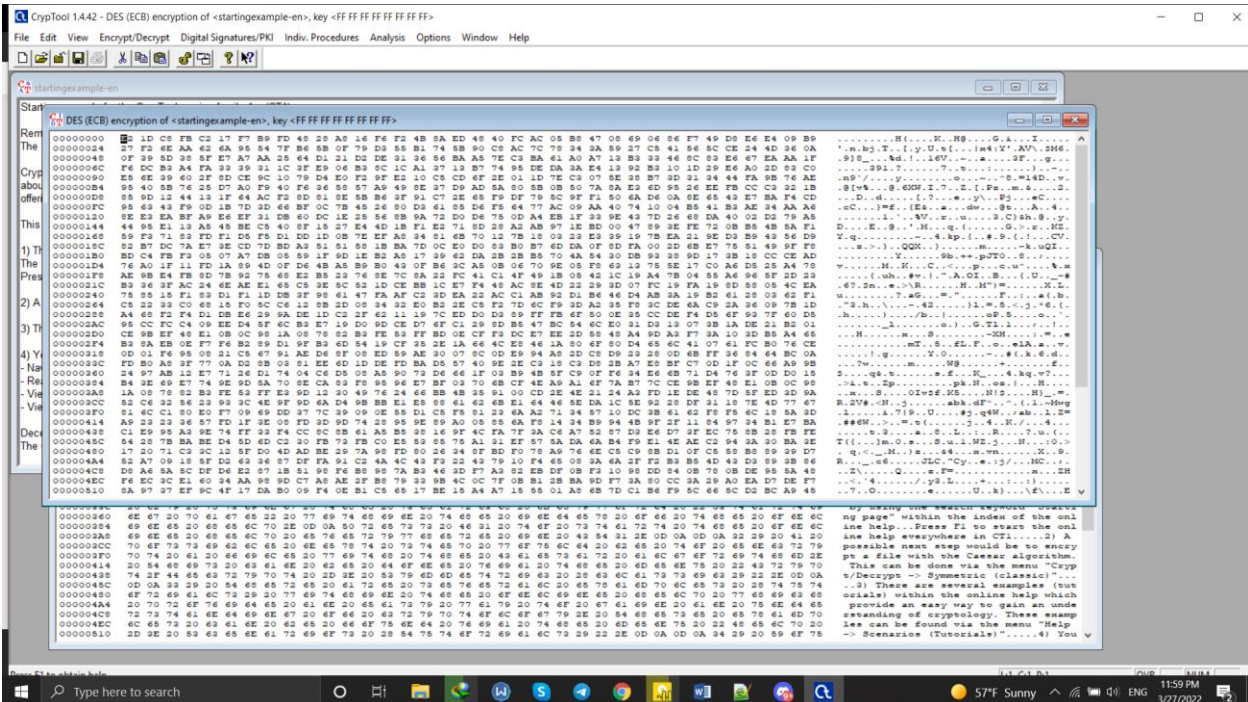
	Y															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	01	8D	F6	CB	52	7B	D1	E8	4F	29	C0	B0	E1	E5	C7
1	74	B4	AA	4B	99	2B	60	5F	58	3F	FD	CC	FF	40	EE	B2
2	3A	6E	5A	F1	55	4D	A8	C9	C1	0A	98	15	30	44	A2	C2
3	2C	45	92	6C	F3	39	66	42	F2	35	20	6F	77	BB	59	19
4	1D	FE	37	67	2D	31	F5	69	A7	64	AB	13	54	25	E9	09
5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

A1	DF	01	EA
23	2C	41	D3
15	BA	06	D4
4A	77	3C	43

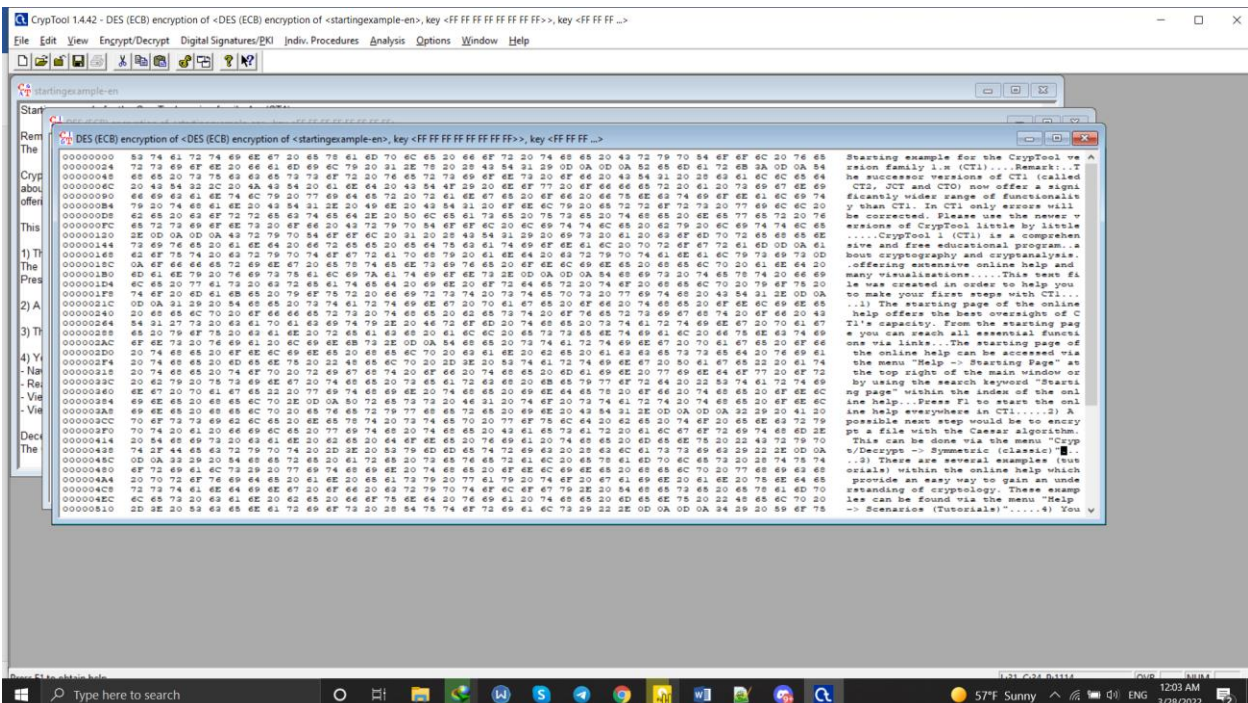
8.1 :

weak key : FF FF FF FF FF FF FF FF

first encryption



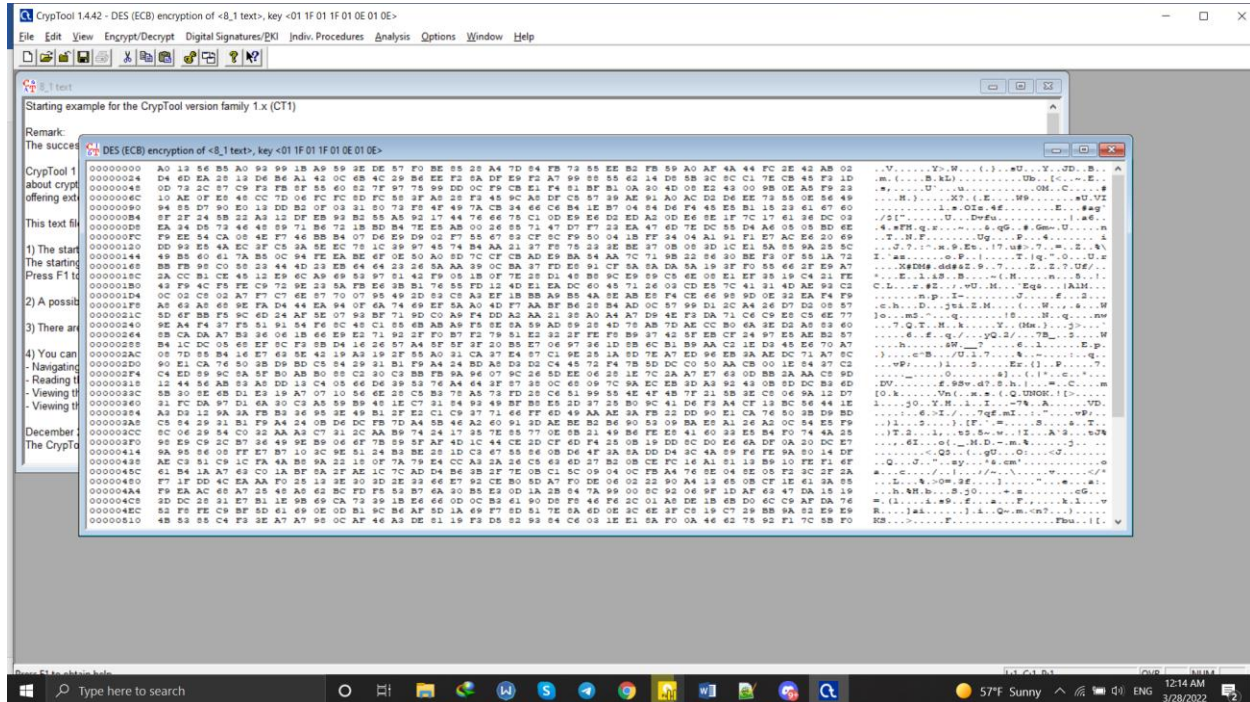
second encryption



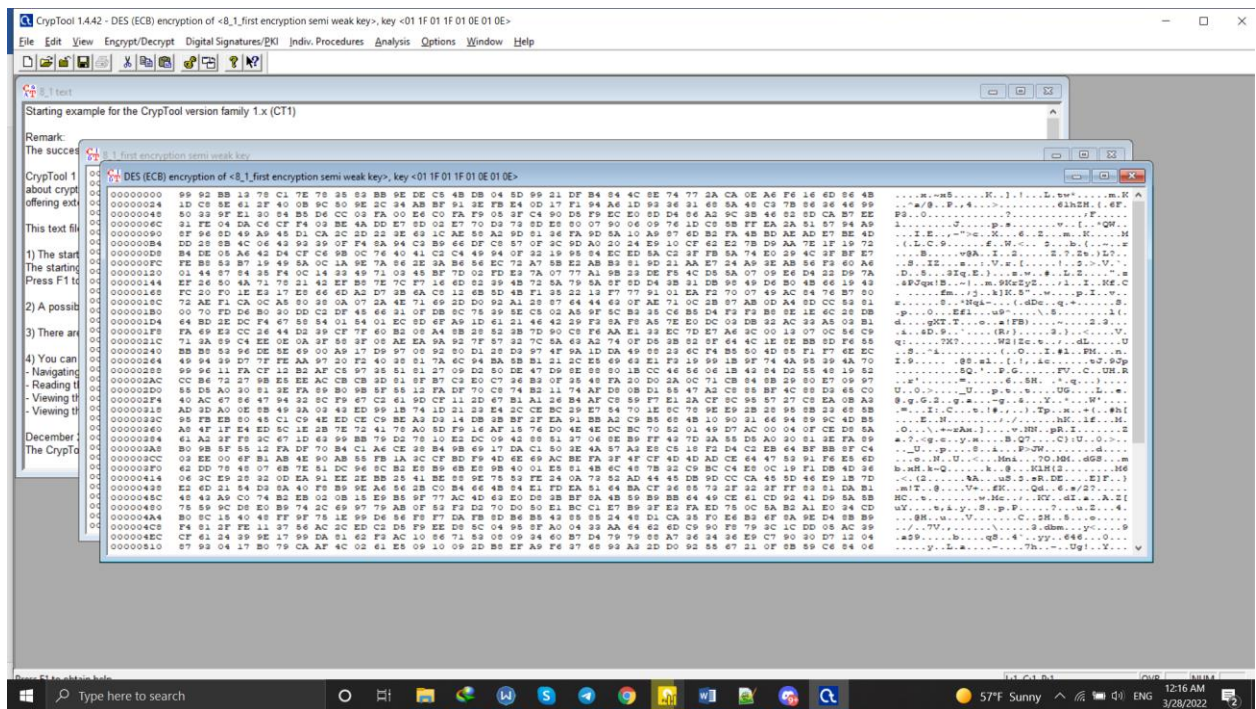
we observe that by encrypting twice we get the plaintext.

semi-weak key : 01 1F 01 1F 01 0E 01 0E

first encryption



second encryption



8.2 :

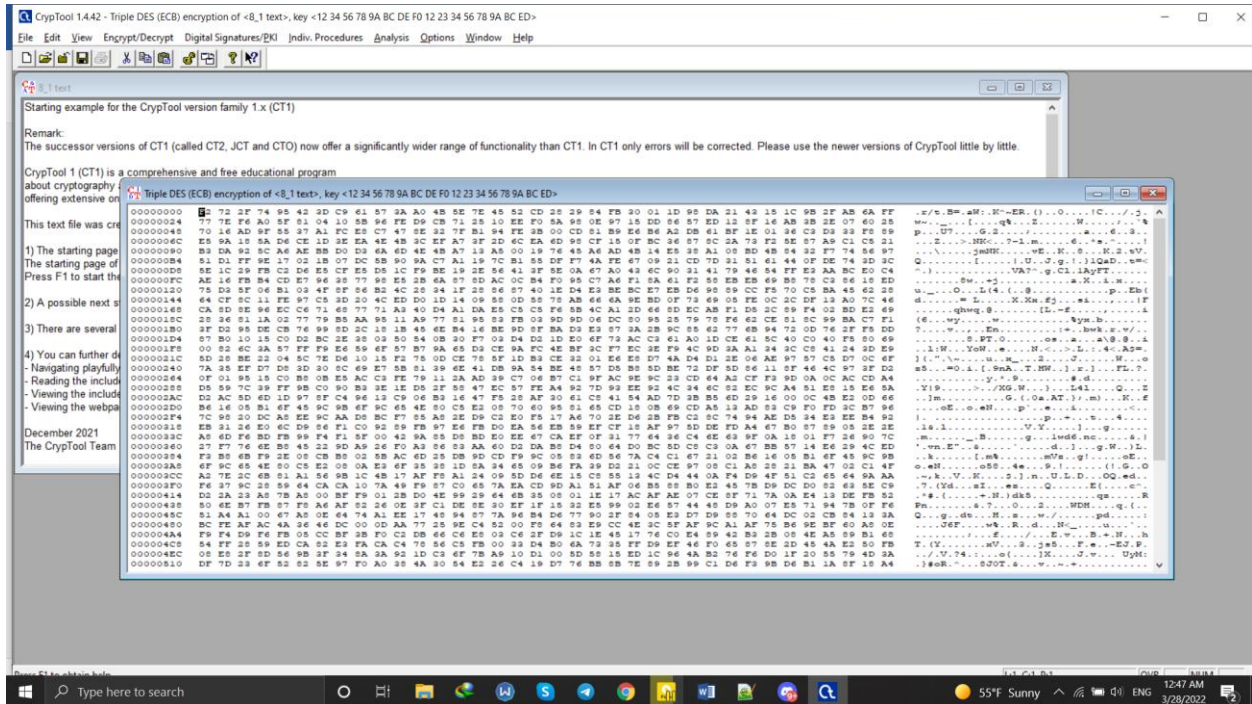
i. because it uses three different keys so the search space for the key get increased by a factor of 2 in power.

ii. mentioned in the previous part the reason is a type of attack called meet in the middle which causes us to always get coefficient of key length reduced by 1.

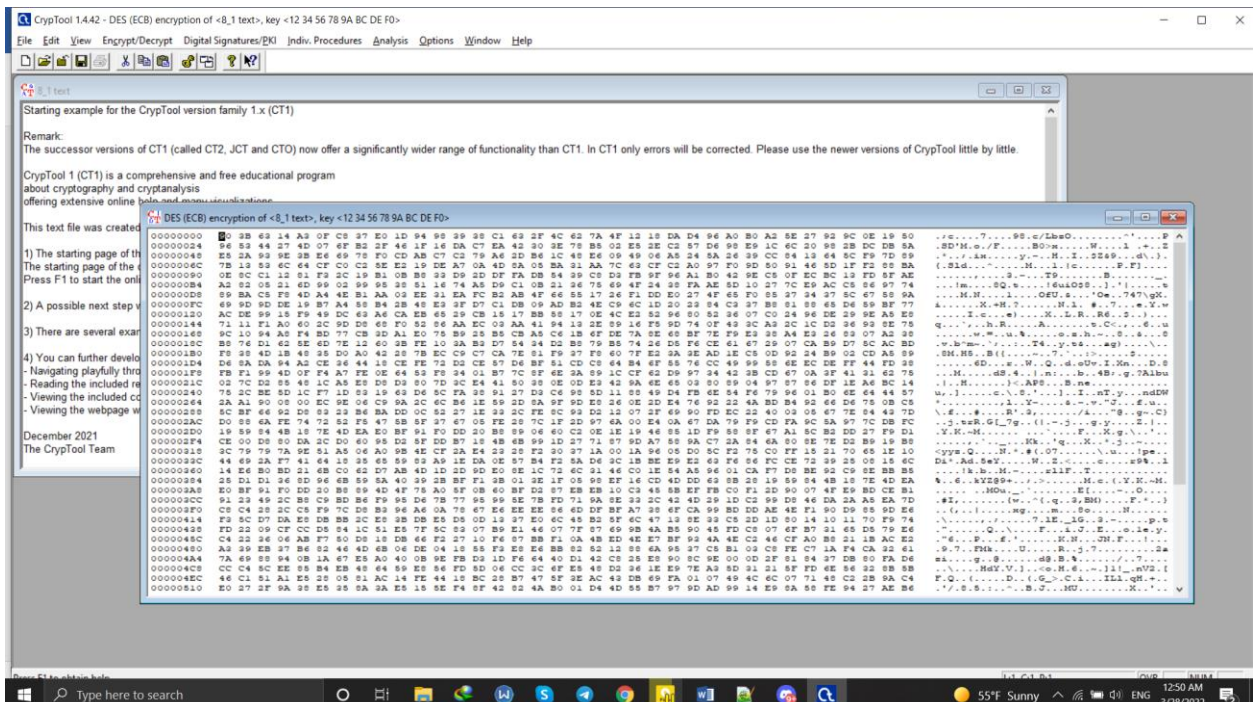
iii. one version does encryption – decryption – encryption and the other one does decryption – encryption – decryption.

if all the keys being independent for brute force attack the power of two is 2^k but if first and third keys are identical then its only k .

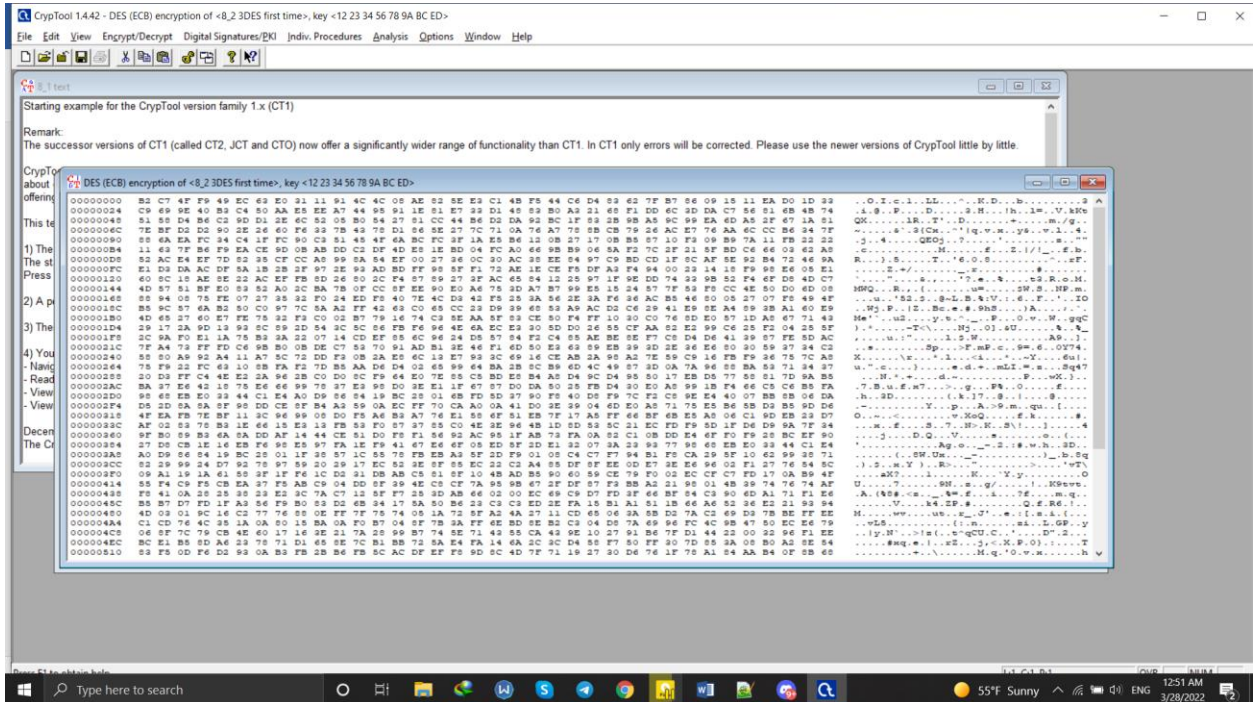
iv. key 12 34 56 78 9A BC DE F0 12 23 34 56 78 9A BC ED



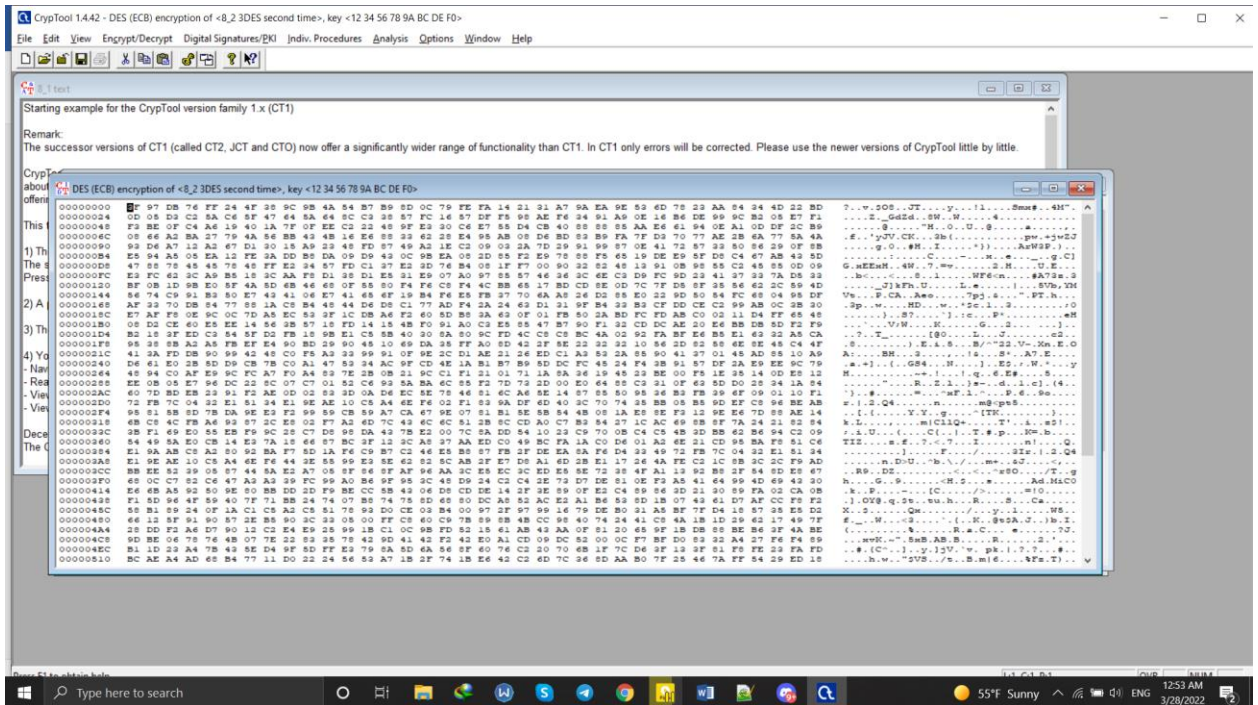
v. k1=12 34 56 78 9A BC DE F0,,, k2=12 23 34 56 78 9A BC ED
first time using k1



second time using k2

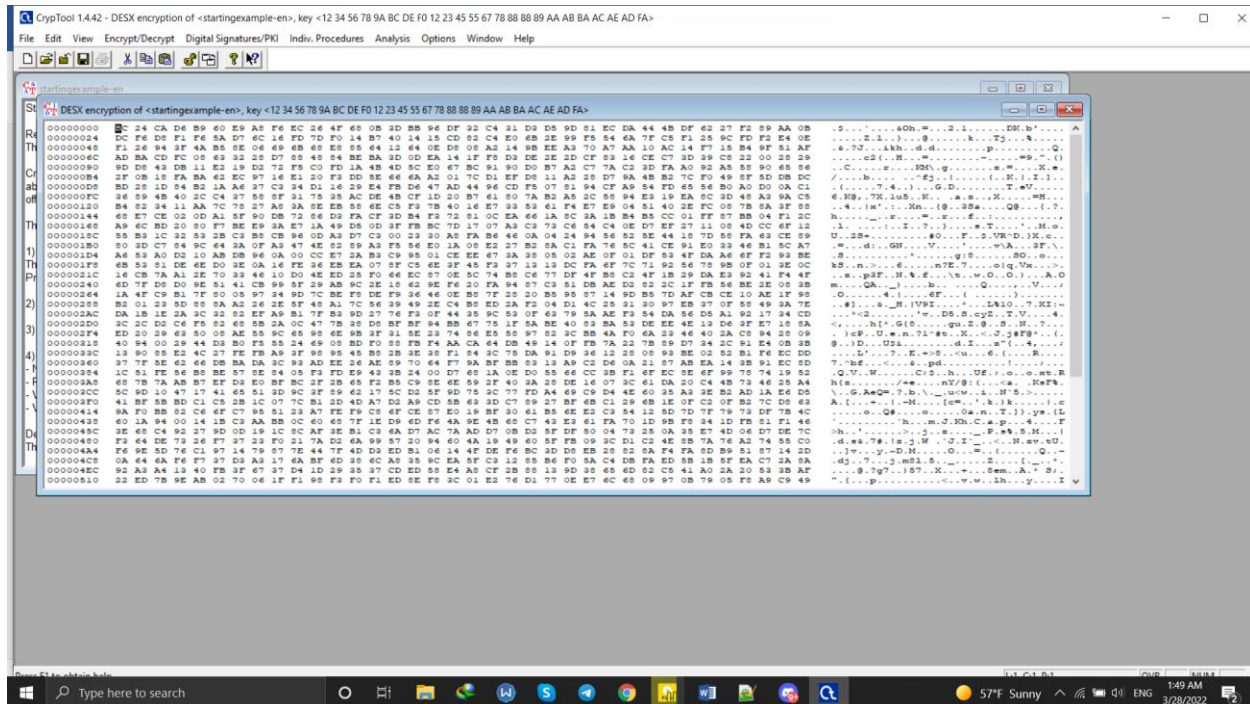


third time using k1

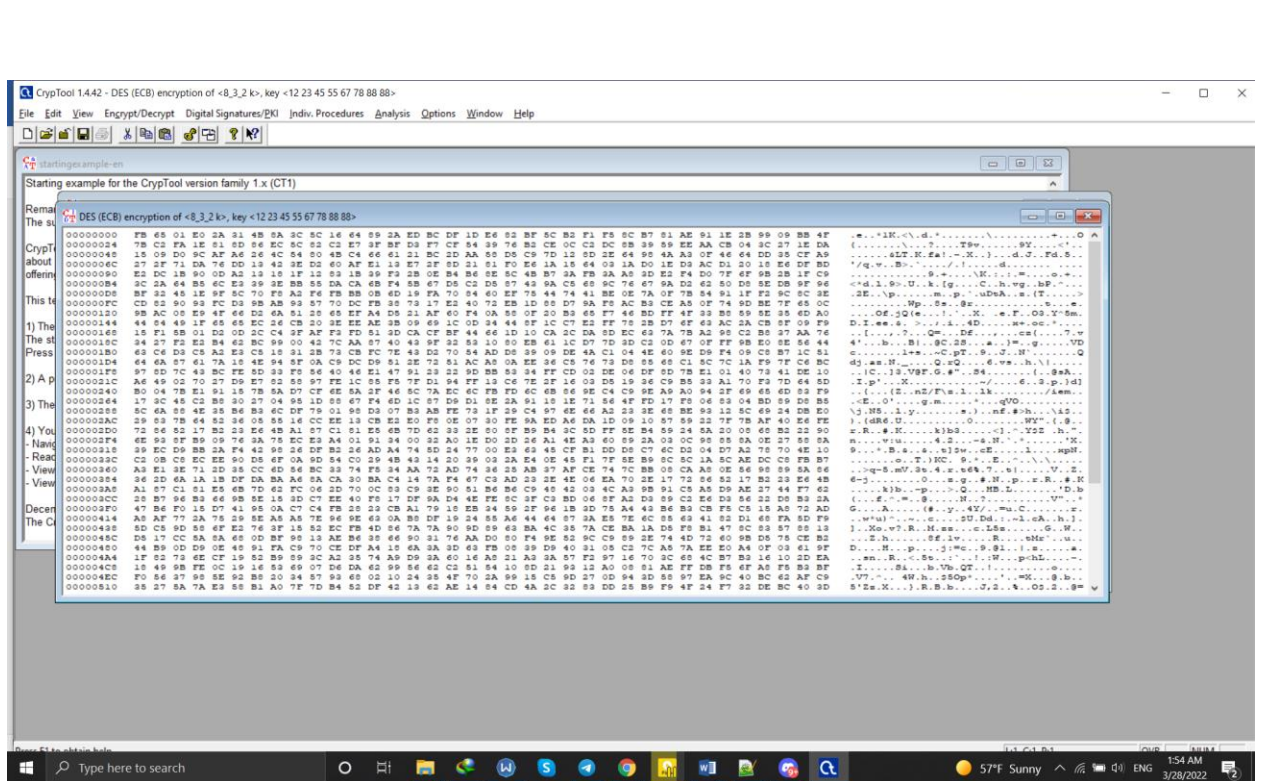
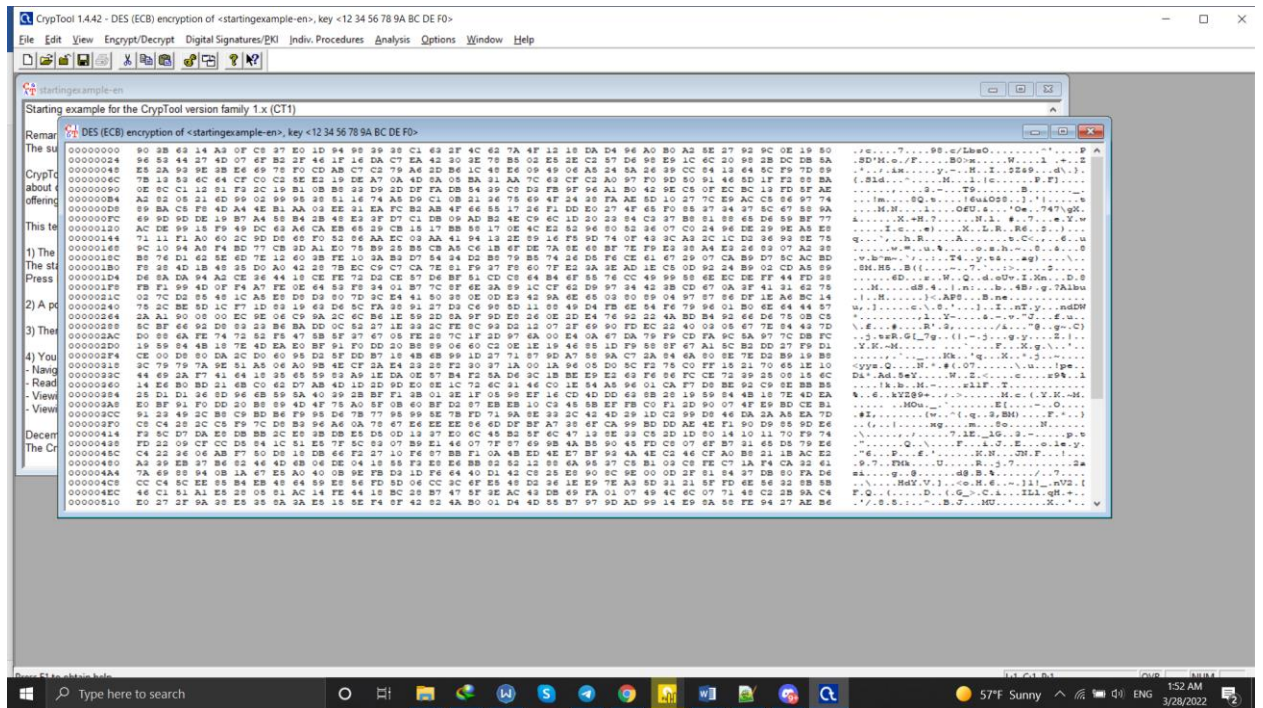


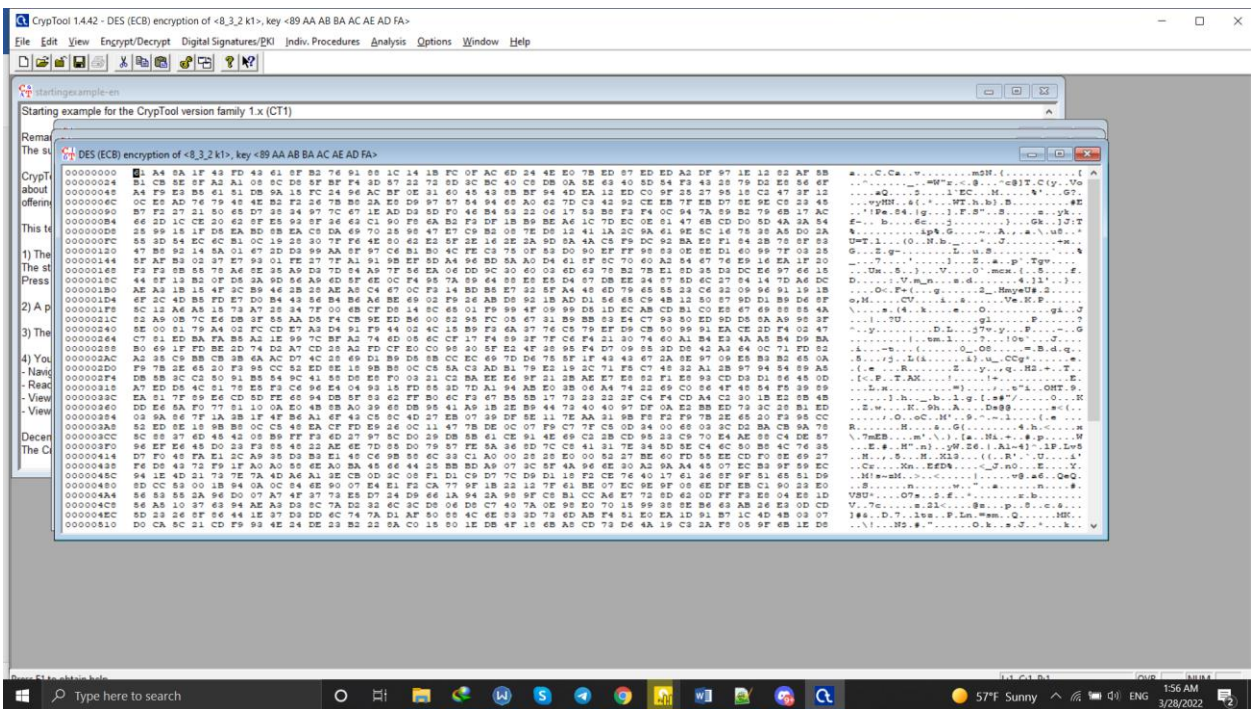
8.3 :

i. key 12 34 56 78 9A BC DE F0 12 23 45 55 67 78 88 88 89 AA
AB BA AC AE AD FA



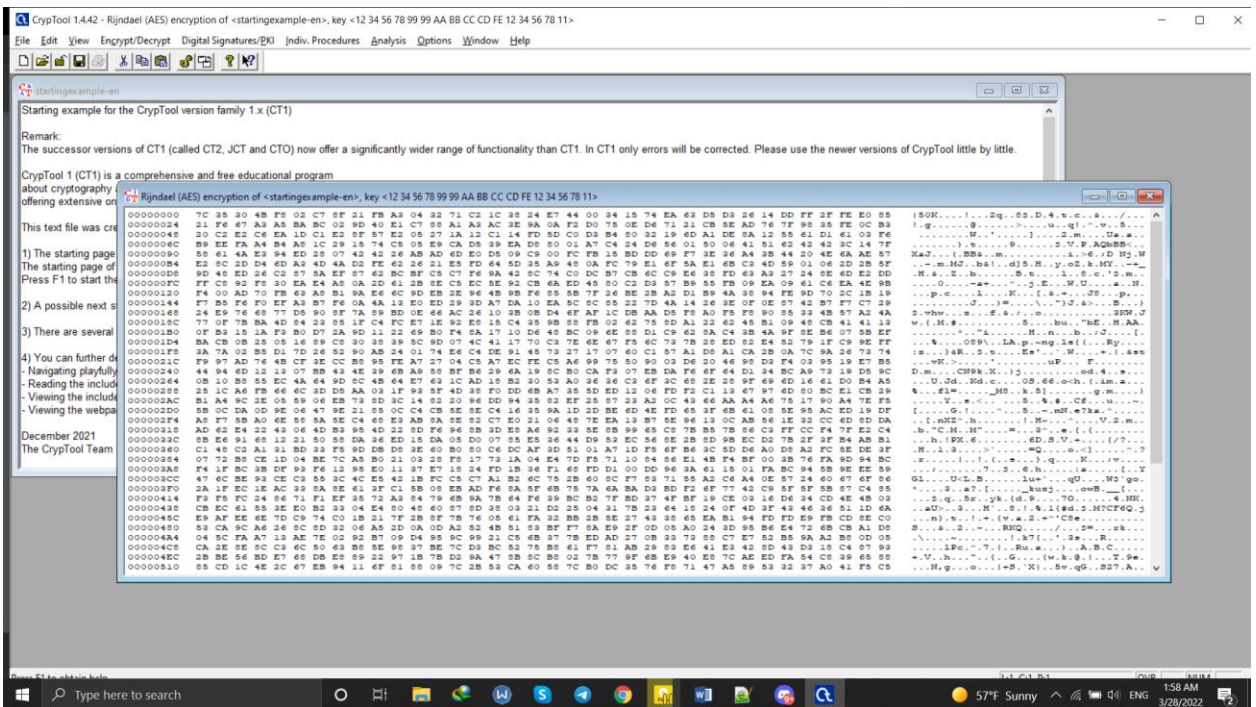
ii. k=12 34 56 78 9A BC DE F0 , k1=12 23 45 55 67 78 88 88,
k2=89 AA AB BA AC AE AD FA





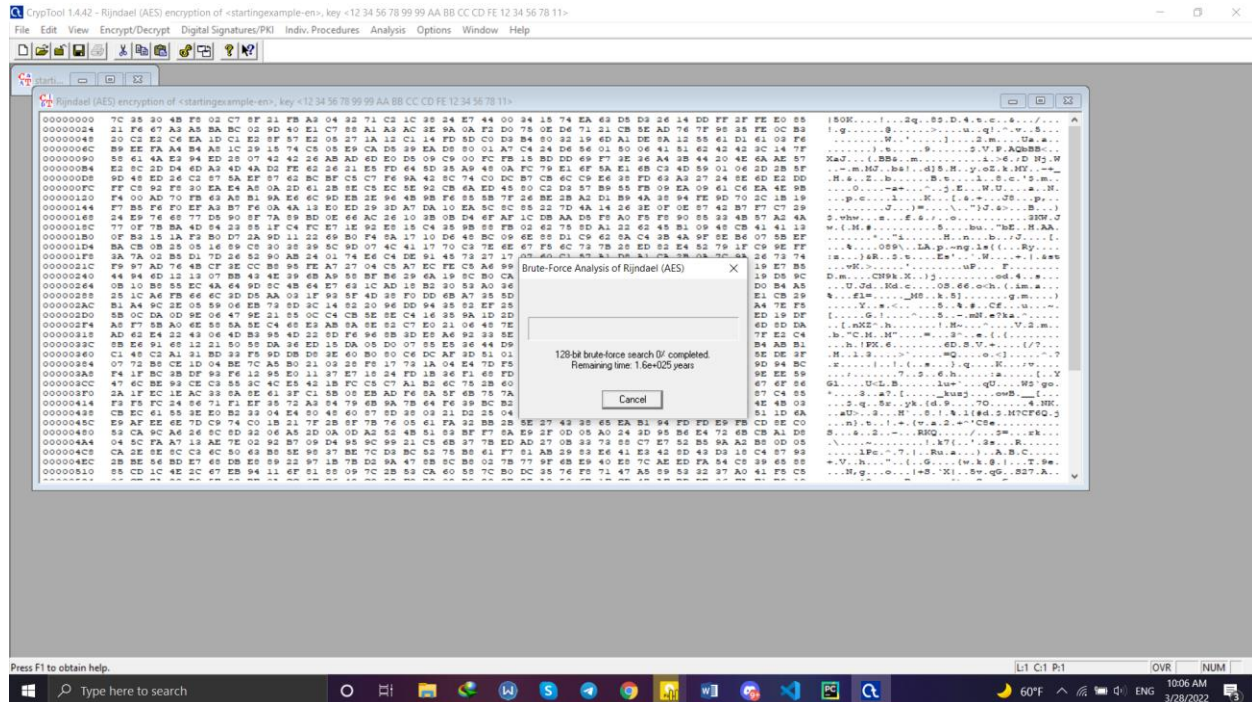
8.4 :

key : 12 34 56 78 99 99 AA BB CC CD FE 12 34 56 78 11



8.5 :

i)



ii) in this part we get the answer immediately