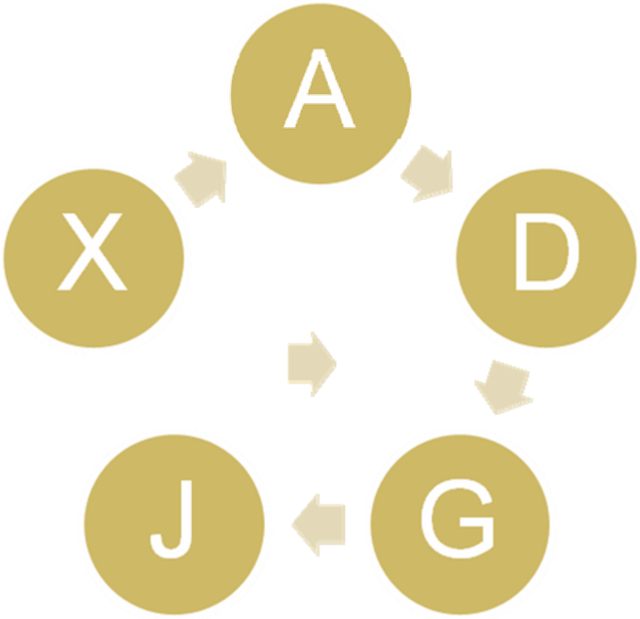
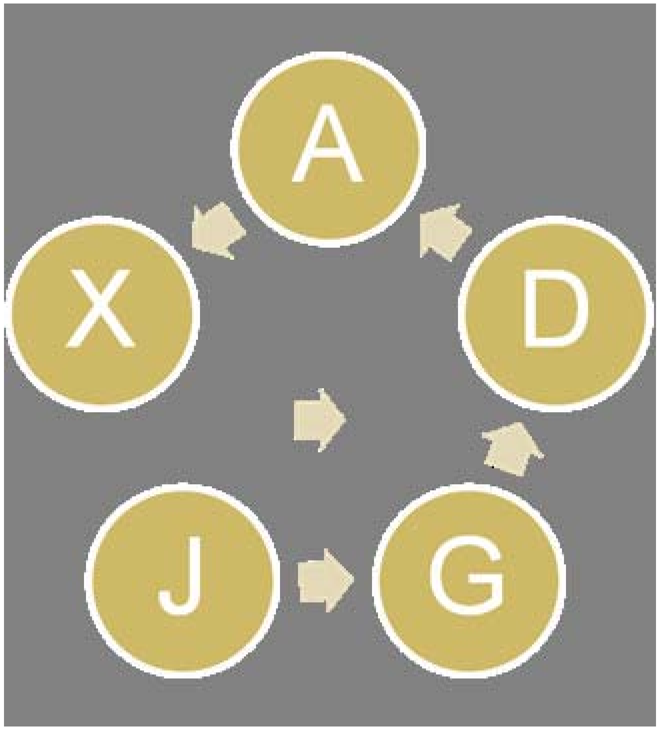
SHIFT CIPHER

- Rotate each letter by the key k
- For example, if k is 3 then:



Encryption

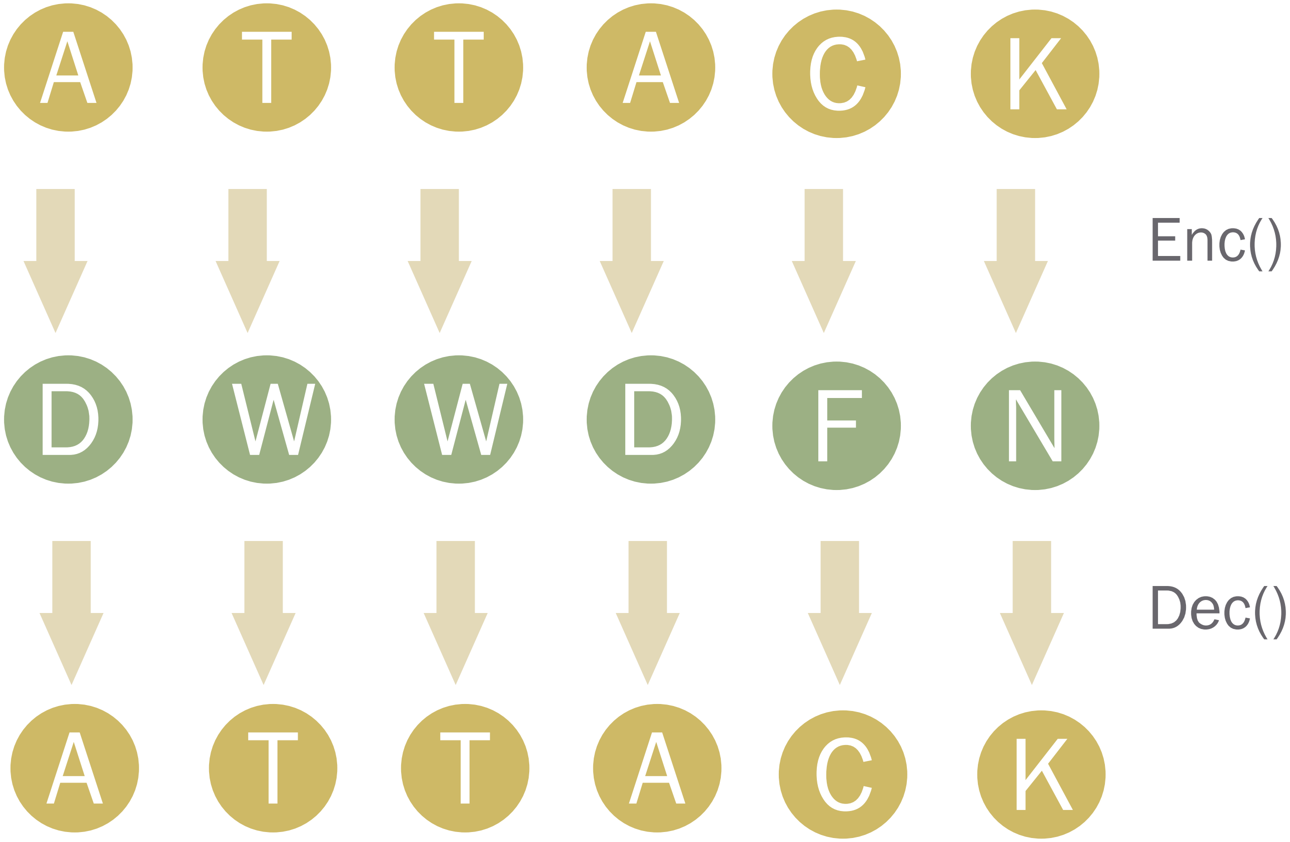
Enc(x) = (x + k) mod 26.



Decryption

Dec(x) = (x - k) mod 26

Example: Key = 3 and Plaintext = “ATTACK”



A T T A C K

Enc()

D W W D F N

Dec()

A T T A C K

Problem with Shift ciphers

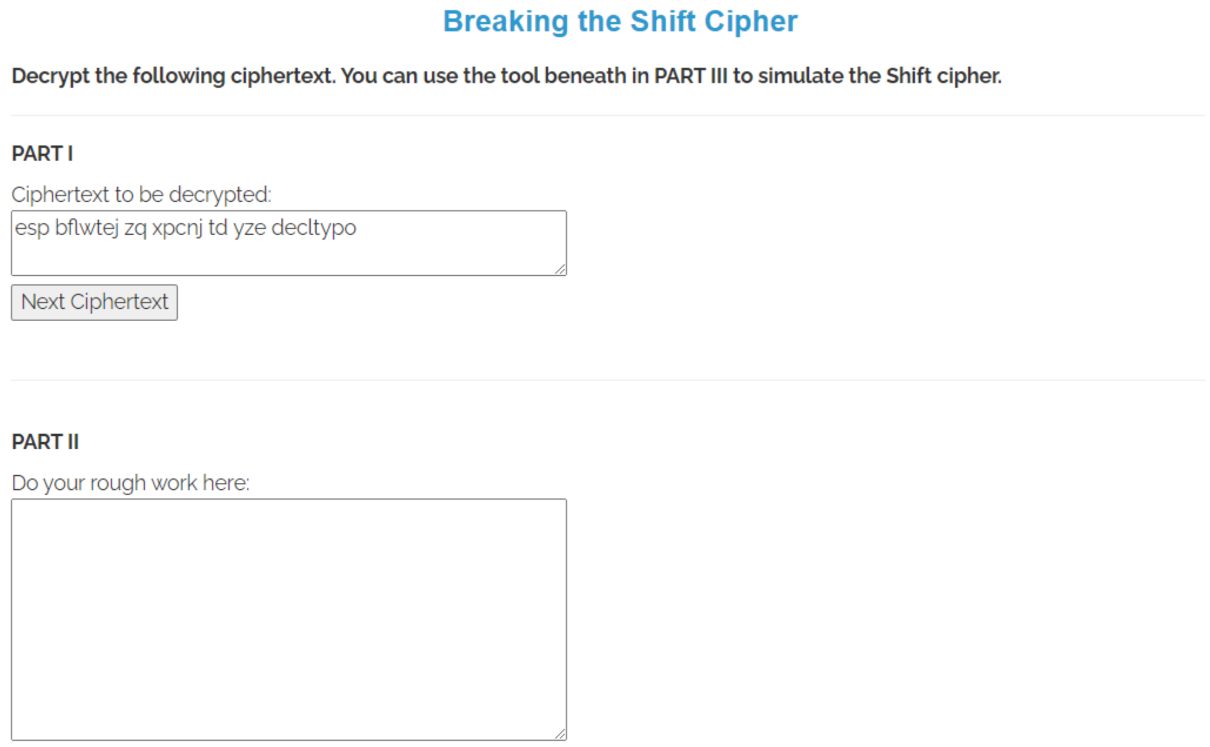
- Not enough keys!
- If we shift a letter 26 times, we get the same letter back.
  - A shift of 27 is the same as a shift of 1, etc.
  - So we only have 25 keys (1 to 25).
- Therefore, easy to attack via brute force.

Example: Cryptanalysis of shift ciphers

- Cipher text : OVDTHUFWVZZPISLRLFZHYLAOLYL

Key Value	Possible Plain Text
1	NUCSGTEVUYYOHRKQKEYGXKZNKXK
2	MTBRFSDUTXXNGQJPJDXFWJYMJWJ
3	LSAQERCTSWWMFPIOICWEVIXLIVI
4	KRZPDQBSRVVLEOHNHBVDUHWKHUH
5	JQYOCPARQUUKDNGMGAUCTGVJGTG
6	IPXNBOZQPTTJCMFLFZTBSFUIFSF
7	HOWMANYPOSSIBLEKEYSARETHERE
8	GNVLZMXONRRHAKDJDXRZQDSGDQD
9	FMUKYLWNMQQGZJCICWQYPCRFCPC
10	ELTJXKVMLPPFYIBHBVPXOBQEBOB
11	DKSIWJULKOOEXHAGAUOWNAPDANA
12	CJRHVITKJNNDWGZFZTNVMZOCZMZ
13	BIQGUHSJIMMCVFYEYSMULYNBYLY

1. For the given ciphertext in the **PART I** of the simulation page, the first step is to decrypt it using each of the twenty-six different keys, k=0,1,...,25 and obtain the corresponding plaintexts. For decryption, you may use the tool given in the **PART III** of the simulation page.



Breaking the Shift Cipher

Decrypt the following ciphertext. You can use the tool beneath in PART III to simulate the Shift cipher.

PART I

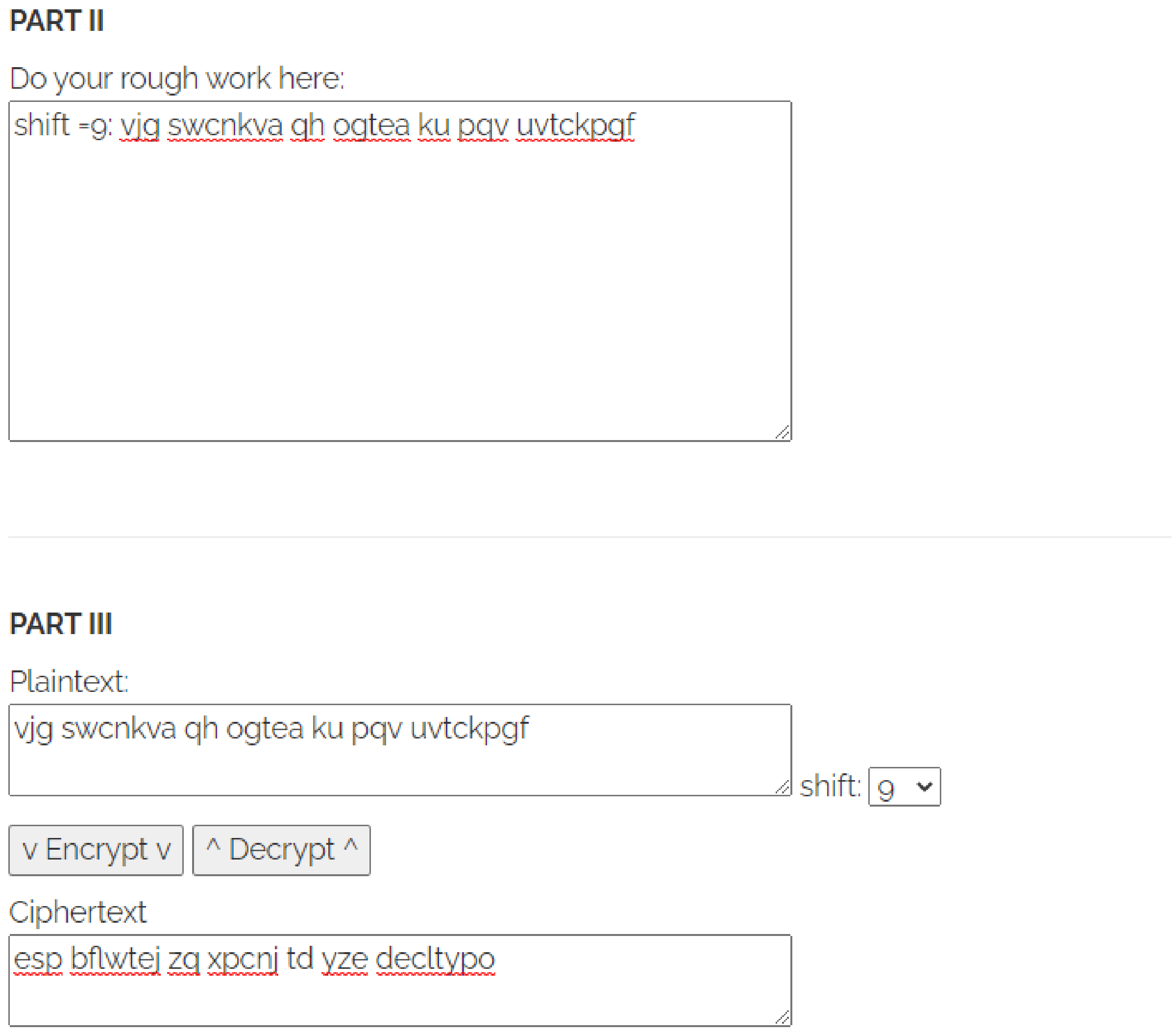
Ciphertext to be decrypted:

esp bflwtej zq xpcnj td yze decltypo

PART II

Do your rough work here:

2. After each decryption, you may cut-and-paste the resultant plaintext in the scratch-pad in the **(PART II)** of the simulation page, if you need to remember it.



PART II

Do your rough work here:

shift =9: vjg swcnkva qh ogtea ku pqv uvtckpgf

PART III

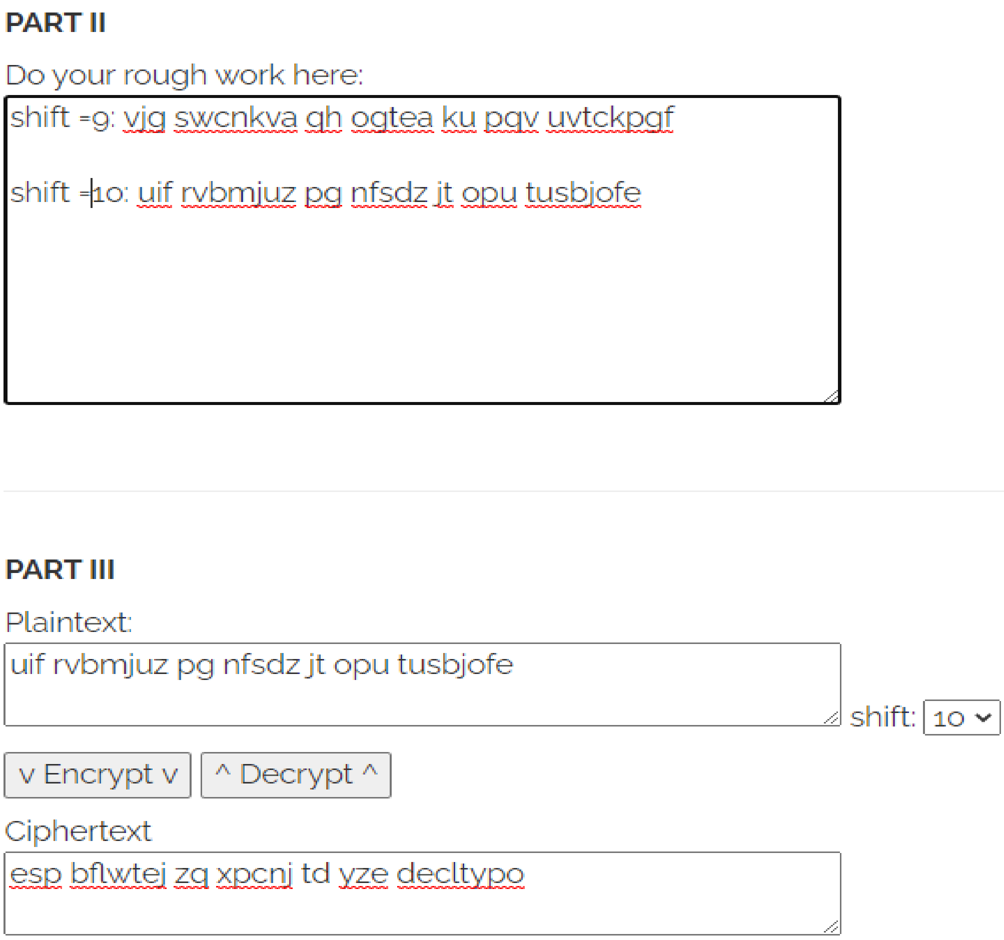
Plaintext:

vjg swcnkva qh ogtea ku pqv uvtckpgf

shift: 9

Ciphertext

esp bflwtej zq xpcnj td yze decltypo



PART II

Do your rough work here:

shift =9: vjg swcnkva qh ogtea ku pqv uvtckpgf

shift =10: uif rvbmjuz pg nfsdz jt opu tusbjofe

PART III

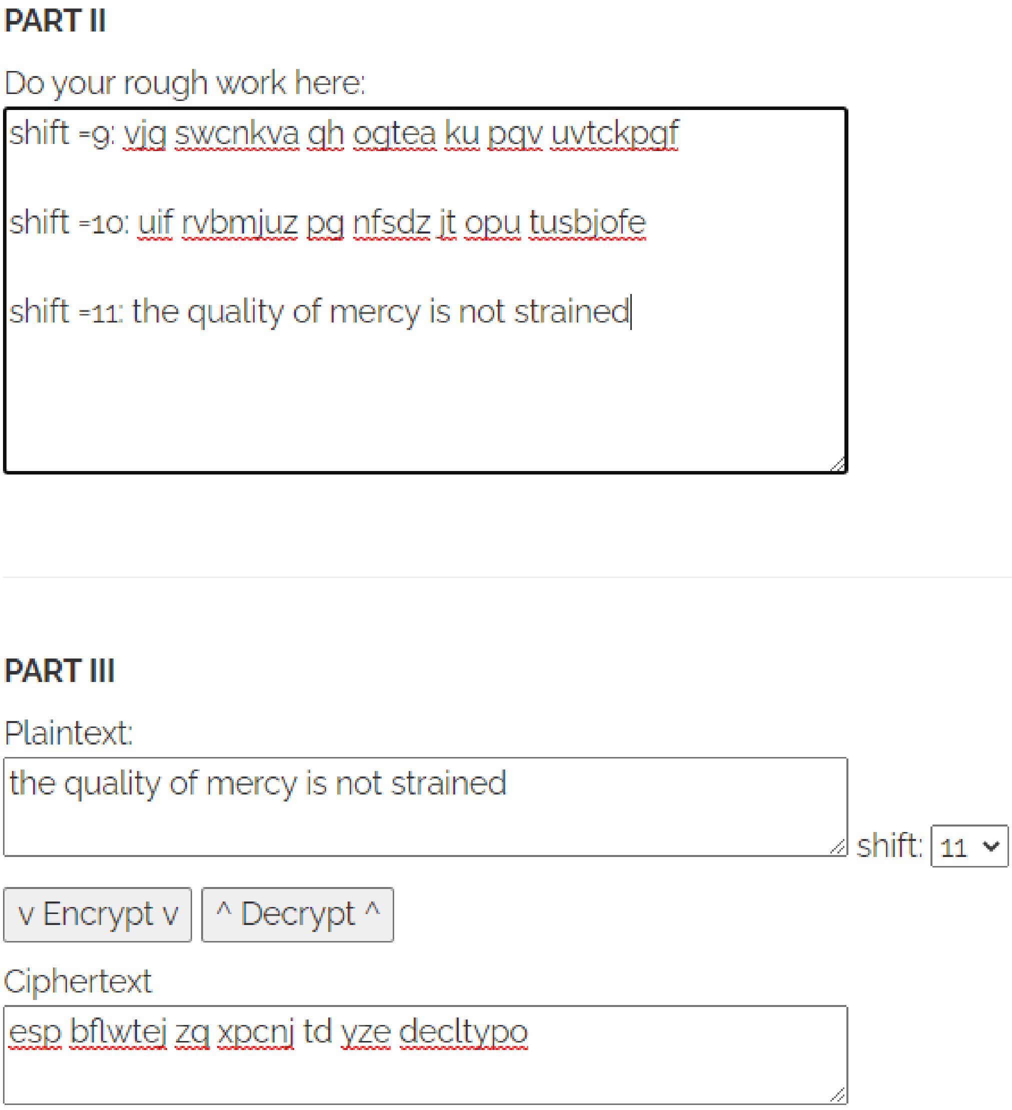
Plaintext:

uif rvbmjuz pg nfsdz jt opu tusbjofe

shift: 10

Ciphertext

esp bflwtej zq xpcnj td yze decltypo



PART II

Do your rough work here:

shift =9: vjg swcnkva qh ogtea ku pqv uvtckpgf

shift =10: uif rvbmjuz pg nfsdz jt opu tusbjofe

shift =11: the quality of mercy is not strained

PART III

Plaintext:

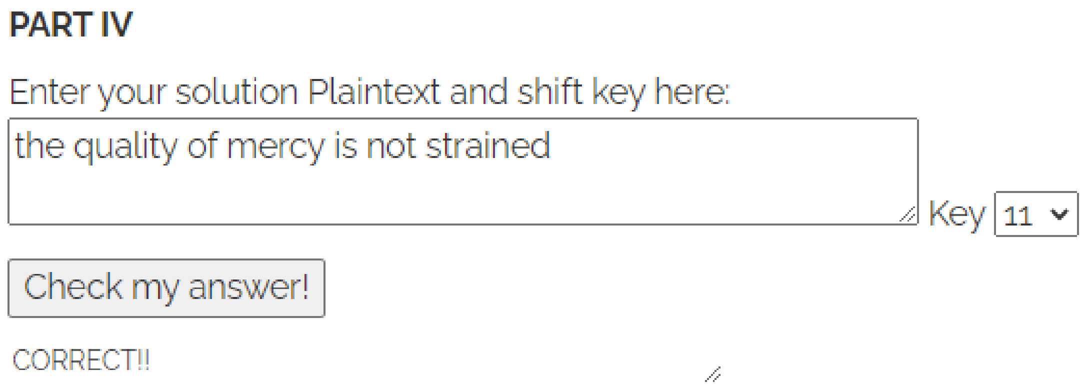
the quality of mercy is not strained

shift: 11

Ciphertext

esp bflwtej zq xpcnj td yze decltypo

3. Finally, observe the plaintexts and choose the most appropriate one (the one that is a meaningful English text) as the recovered plaintext and cut-and-paste it in the text-field named **PART IV** "Solution Plaintext". Also select the corresponding key in the text-field named "Key" and click on "Check My answer" Button.



PART IV

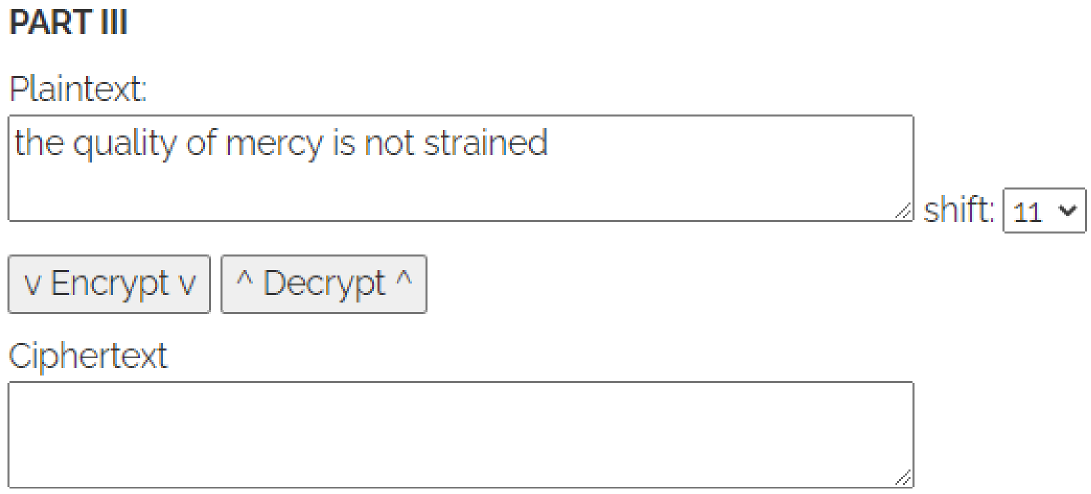
Enter your solution Plaintext and shift key here:

the quality of mercy is not strained

Key 11

CORRECT!!

4. Verify that your answer is correct, by encrypting the solution plaintext with your key.

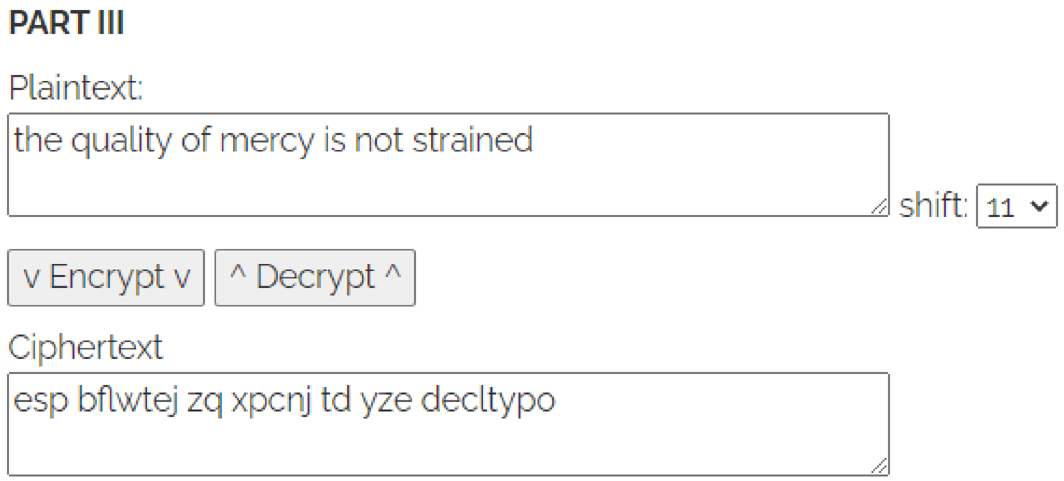


PART III

Plaintext:

the quality of mercy is not strained

shift: 11



PART III

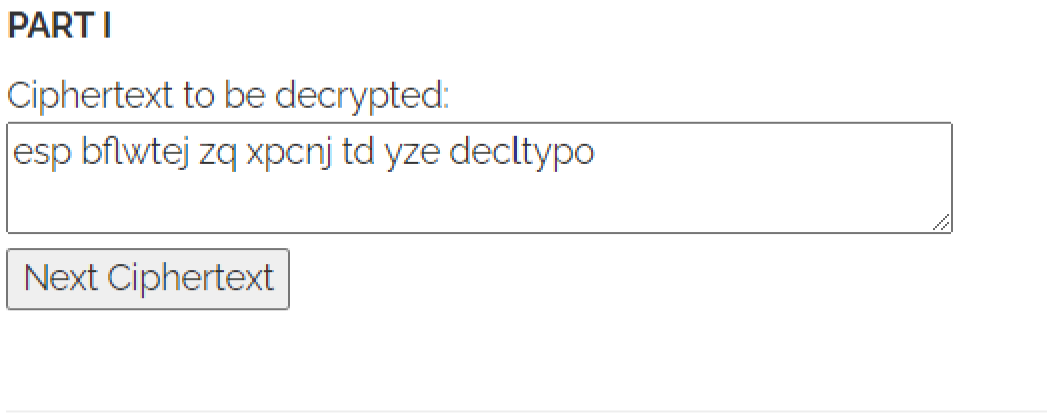
Plaintext:

the quality of mercy is not strained

shift: 11

Ciphertext

esp bflwtej zq xpcnj td yze decltypo



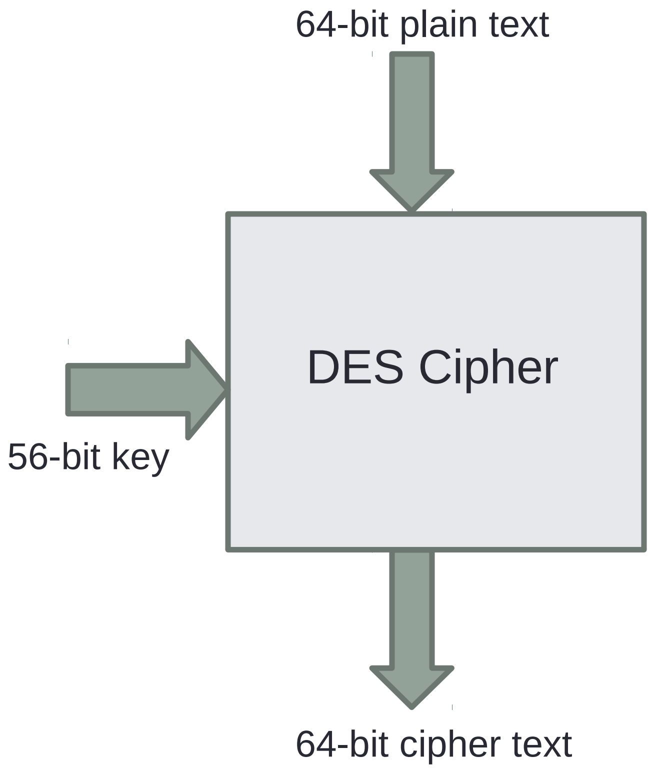
PART I

Ciphertext to be decrypted:

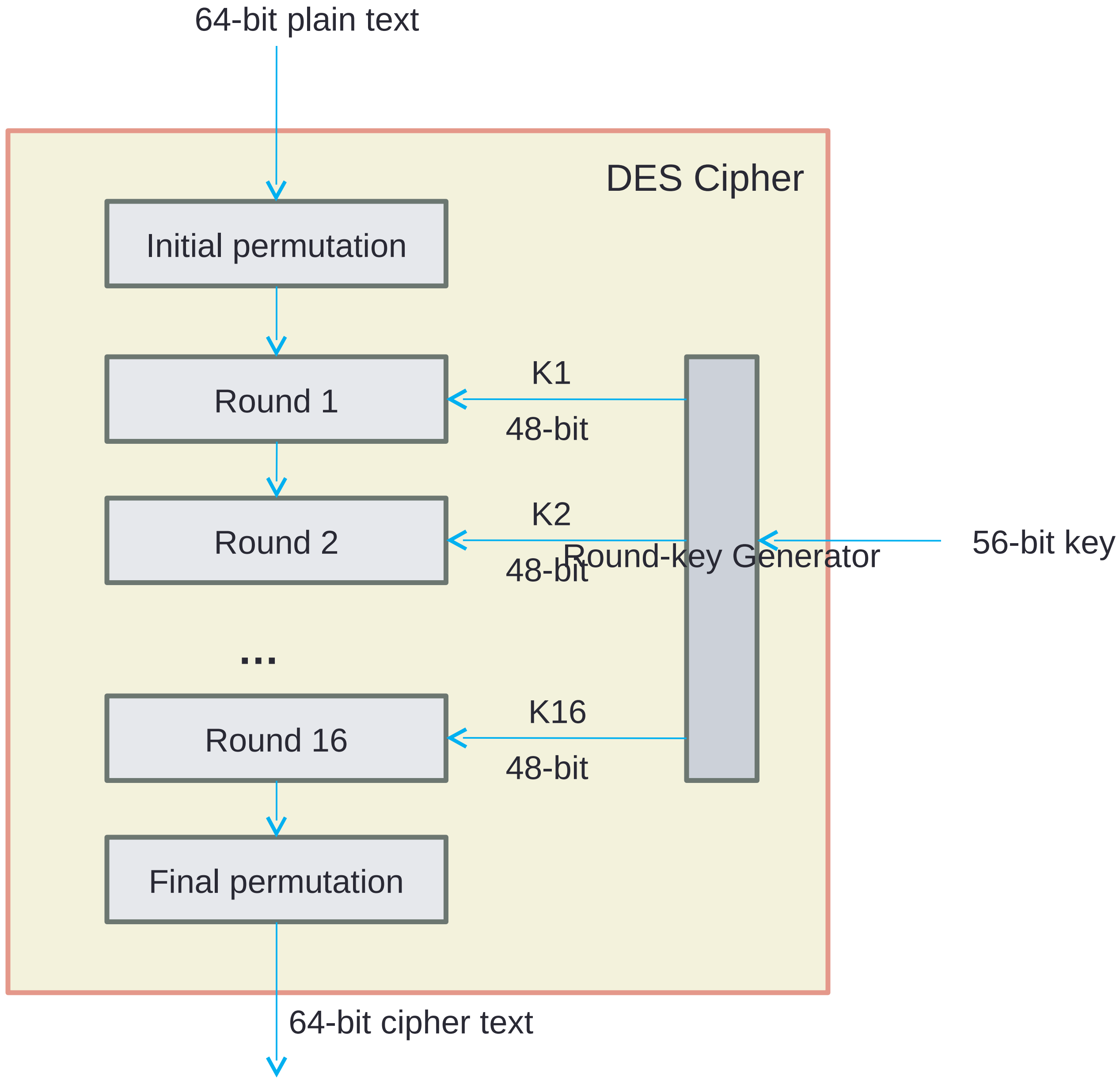
esp bflwtej zq xpcnj td yze decltypo

DES

DES is a Block cipher, which takes 64-bit plain text and creates a 64-bit cipher text



General Structure of DES



Initial and Final permutations

Initial permutation table

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

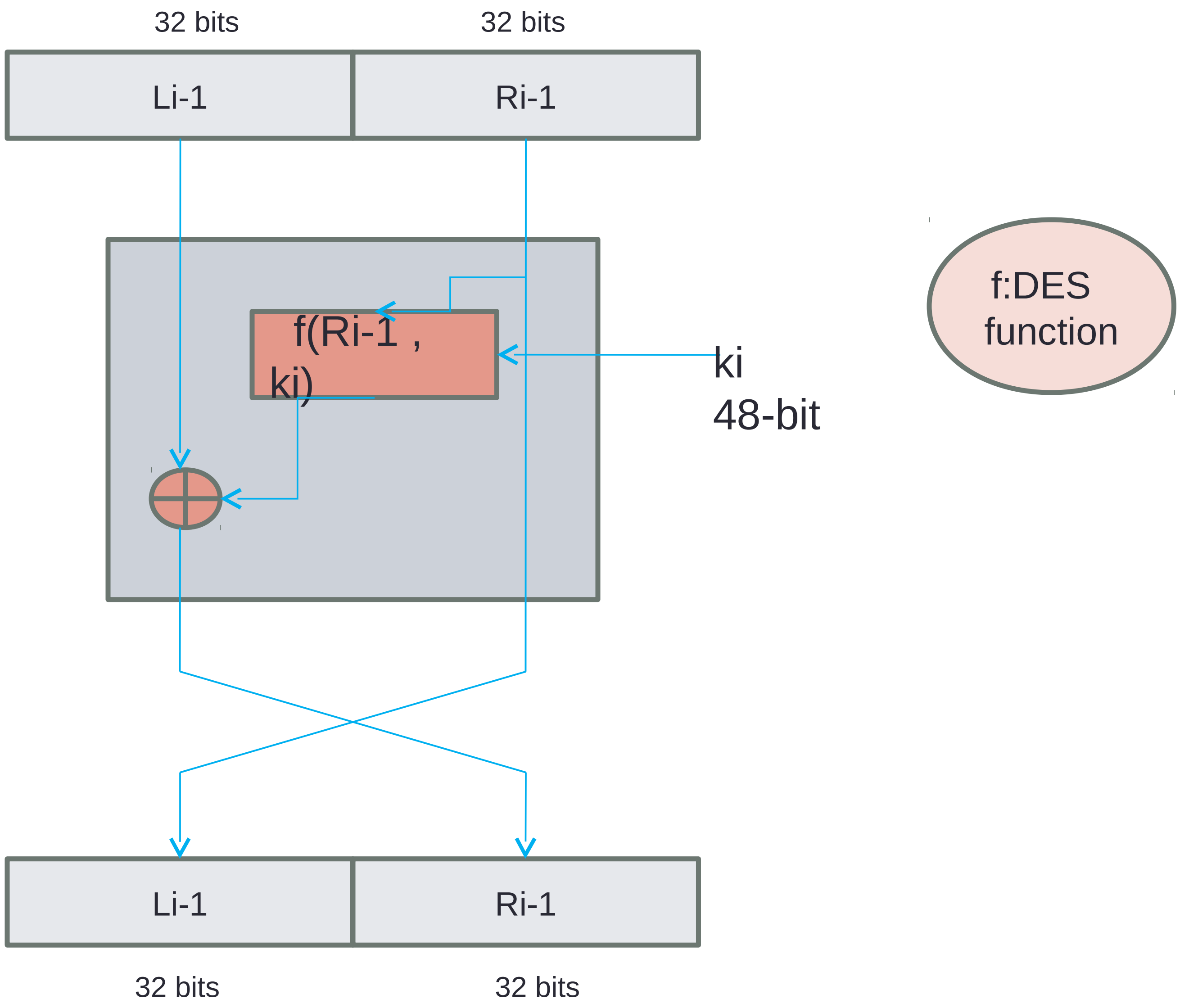
The 58th bit of the input 64-bit plain text becomes the 1st bit, the 50th bit becomes the 2nd bit and so on according to the initial permutation table

Final permutation table

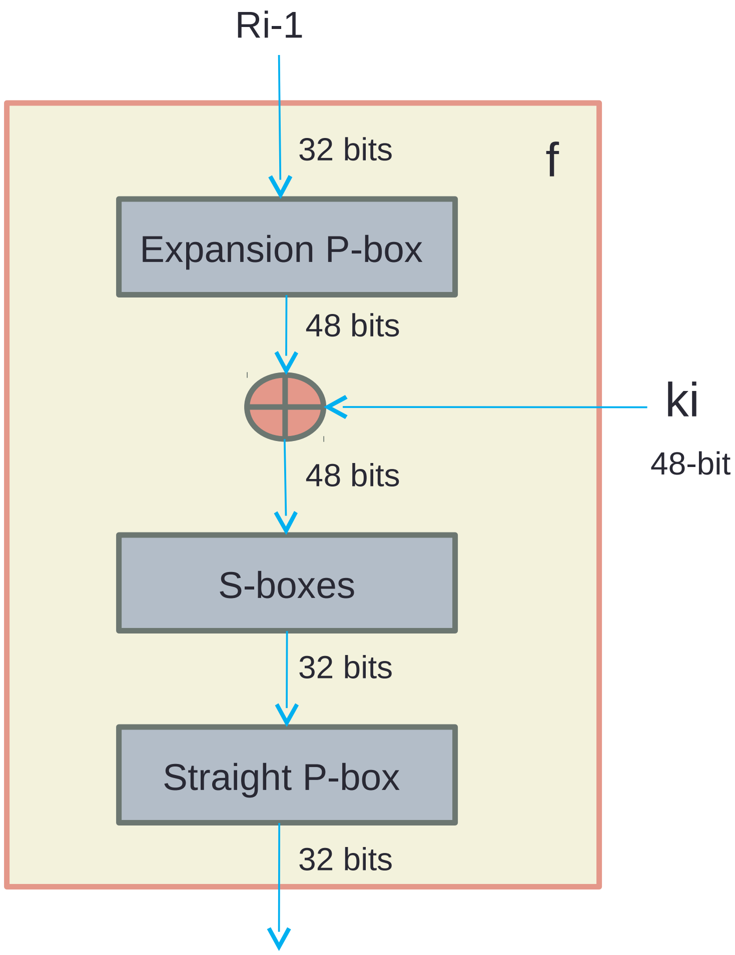
40	08	48	16	56	24	64	32
39	07	47	15	55	23	63	31
38	06	46	14	54	22	62	30
37	05	45	13	53	21	61	29
36	04	44	12	52	20	60	28
35	03	43	11	51	19	59	27
34	02	42	10	50	18	58	26
33	01	41	09	49	17	57	25

The 40th bit of the 64-bit output of the Round 16 becomes the 1st bit, the 8th bit becomes the 2nd bit and so on according to The final permutation table

One round in DES (Feistel structure)

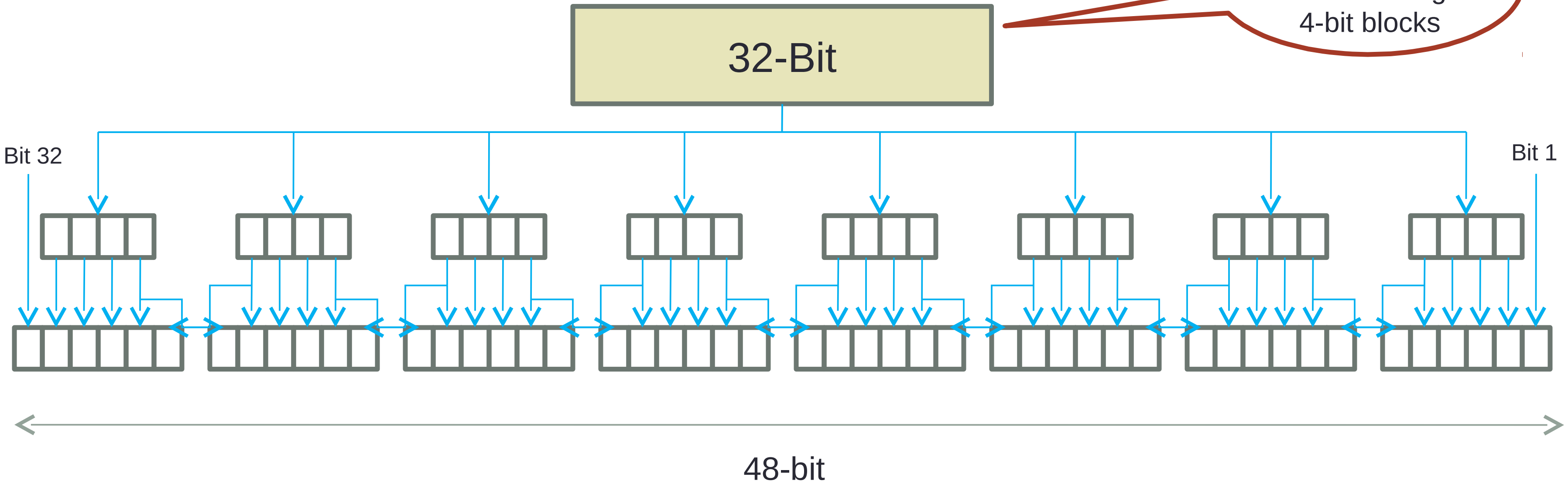


DES Function



DES Function : Expansion permutation

The input 32-bits are expanded to 48 bits in the Expansion P-Box module in the following way



The resulting 48-bit output is permuted using the Expansion P-Box

DES Function : Expansion Permutation and Straight permutation

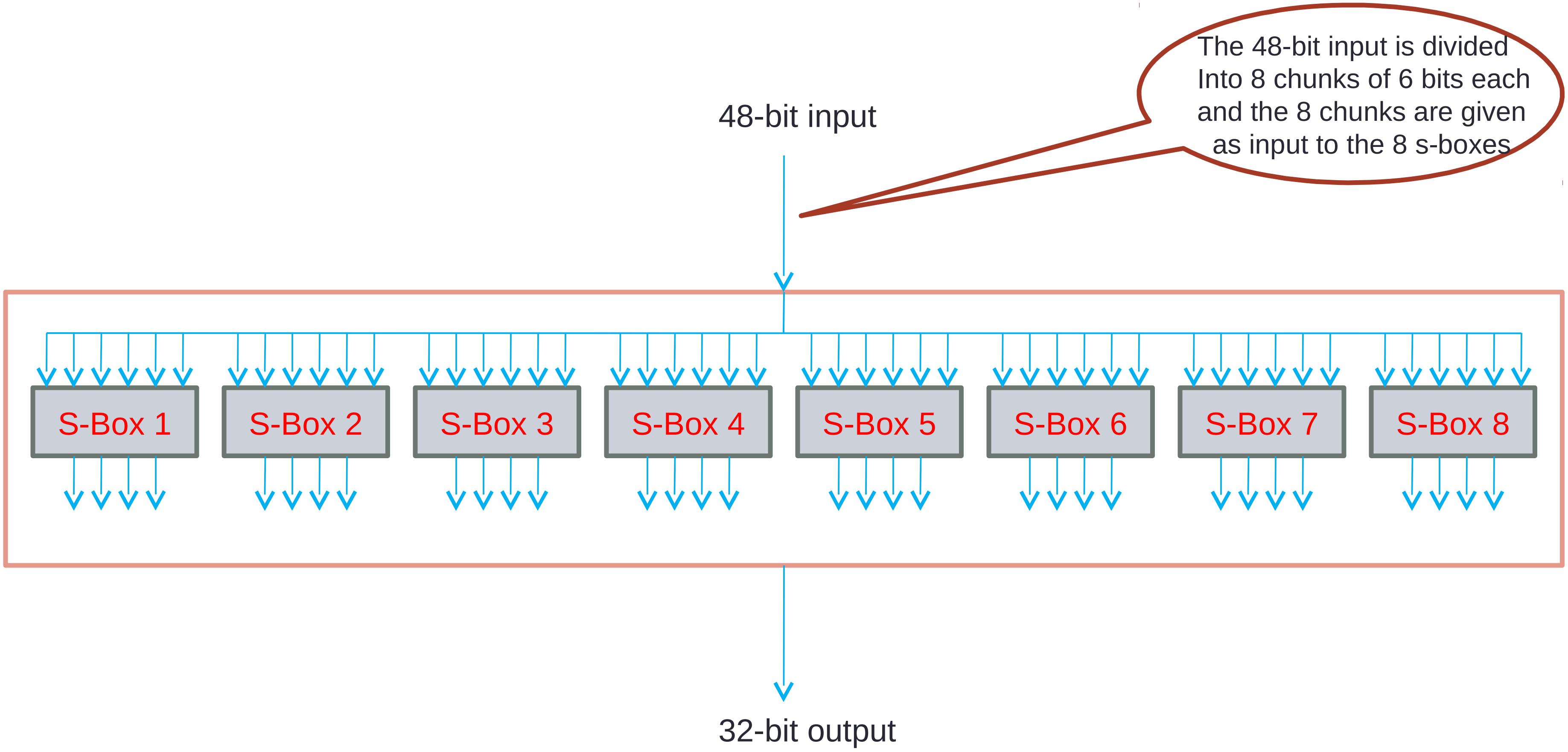
Expansion P-box

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	30	31	32	01

Straight P-box

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

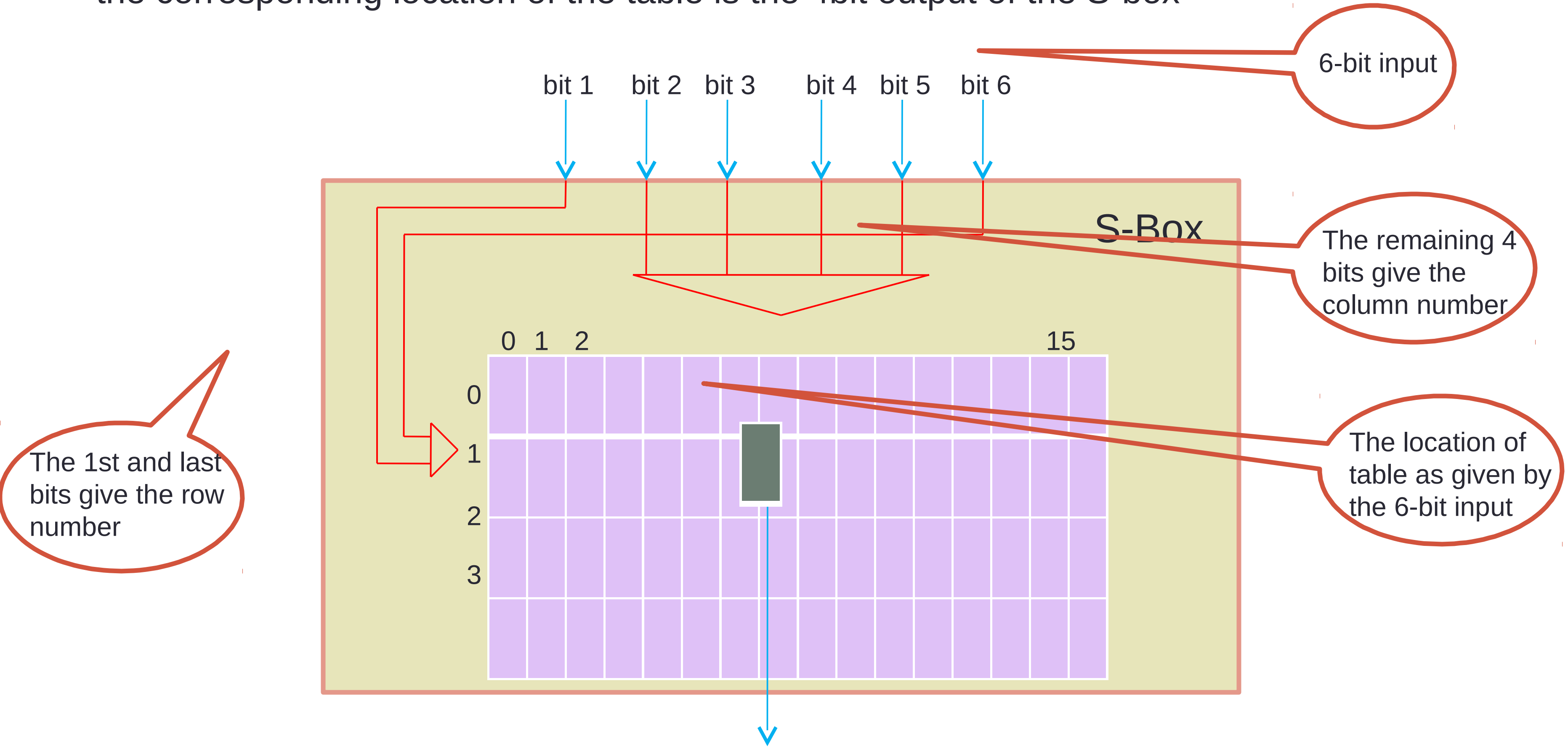
DES Function : Substitution Boxes


48-bit input
The 48-bit input is divided
Into 8 chunks of 6 bits each
and the 8 chunks are given
as input to the 8 s-boxes
S-Box 1
S-Box 2
S-Box 3
S-Box 4
S-Box 5
S-Box 6
S-Box 7
S-Box 8
32-bit output

The output of each S-box is 4-bit. When these are combined the result is a 32-bit output

DES Function : Substitution Boxes

Each S-box uses a corresponding 4 row by 16 column table
Given a 6-bit input, the 1st and the 6th bits are used to address one of the rows and the remaining 4 bits are used to address one of the 16 columns. Finally, the value found in the corresponding location of the table is the 4bit output of the S-box


6-bit input
bit 1
bit 2
bit 3
bit 4
bit 5
bit 6
S-Box
The remaining 4 bits give the column number
0
1
2
15
0
1
2
3
The 1st and last bits give the row number
The location of table as given by the 6-bit input

4-bit output (the value in the corresponding location of the 4x16 table)

DES Function : Substitution Boxes

An Example

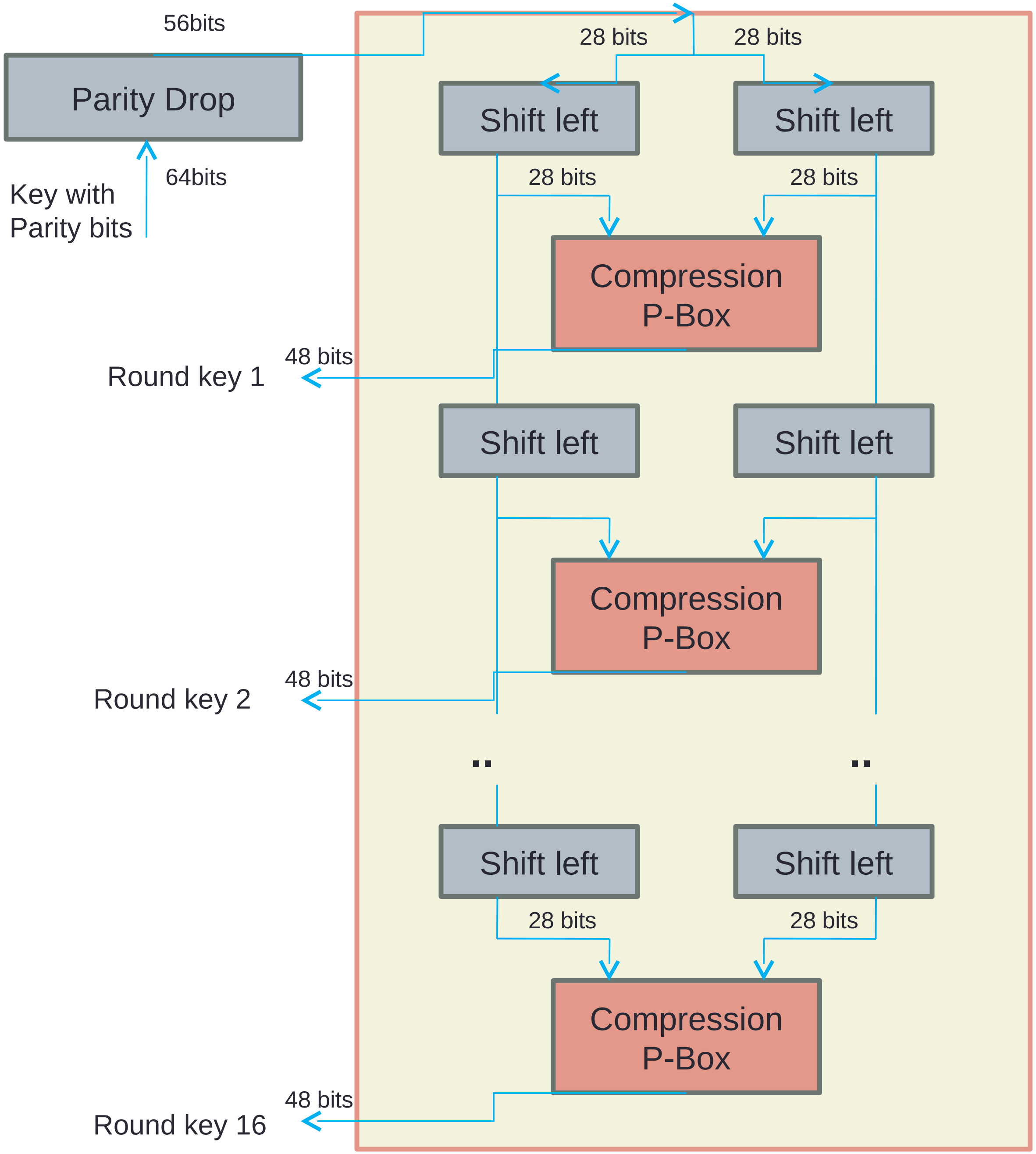
Consider the 6-bit input to s-box 1 is 100011
The 1st and last bits put together is 11 which is ‘3’ in decimal. So we select the 3rd row
The middle bits are 00001 which is ‘1’ in decimal. So we select the 1st column

The corresponding table for S-box 1 is shown below

	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

The value in the 3rd in binary)

Key Generation



Shifting

Rounds	Shift
1,2,9,16	One bit
Others	Two bits

DES

Parity Drop and Compression Permutation

The parity drop module drops the parity bits (bits 8,16,24,..,64) from the 64-bit key and permutes the rest of the 56 bits according to the parity drop table

The Compression permutation module changes the 56 bits to 48 bits using the key compression table, which are used as the key for a round

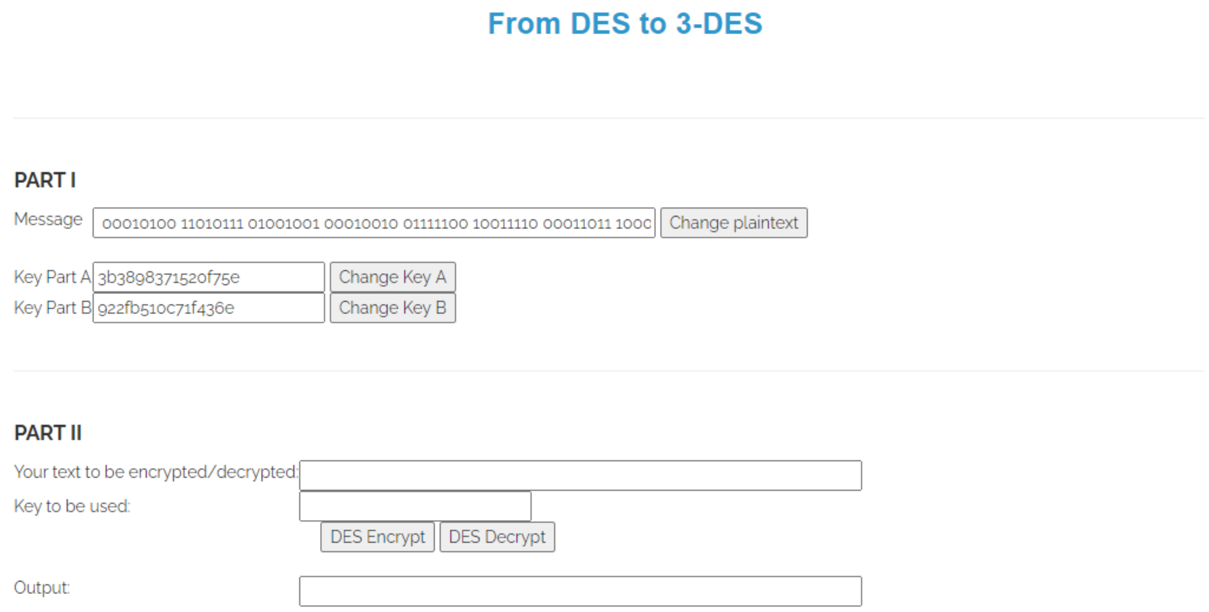
Parity drop table

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 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

1.


From DES to 3-DES

PART I

Message 00010100 11010111 01001001 00010010 01111100 10011110 00011011 100c

Key Part A 3b3898371520f75e

Key Part B 922fb510c71f436e

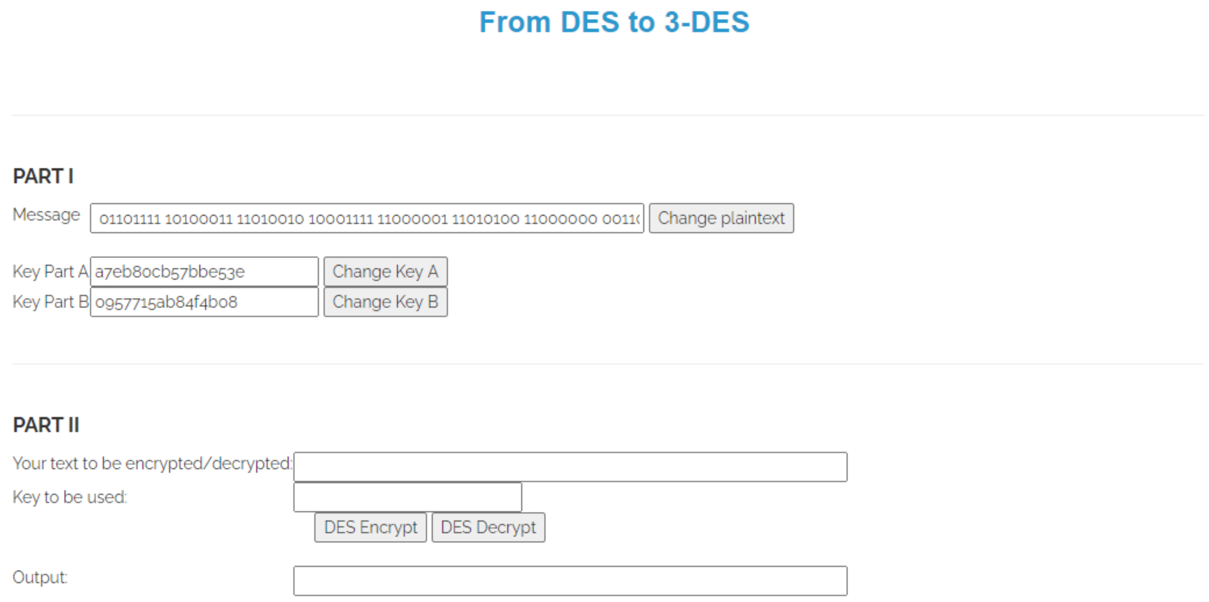
PART II

Your text to be encrypted/decrypted:

Key to be used:

Output:

2.
Generate Plaintext **m**, **key A** and **key B** by clicking on respective buttons **PART I** of the simulation page.


From DES to 3-DES

PART I

Message 01101111 10100011 11010010 10001111 11000001 11010100 11000000 0011(

Key Part A a7eb80cb57bbe53e

Key Part B 0957715ab84f4b08

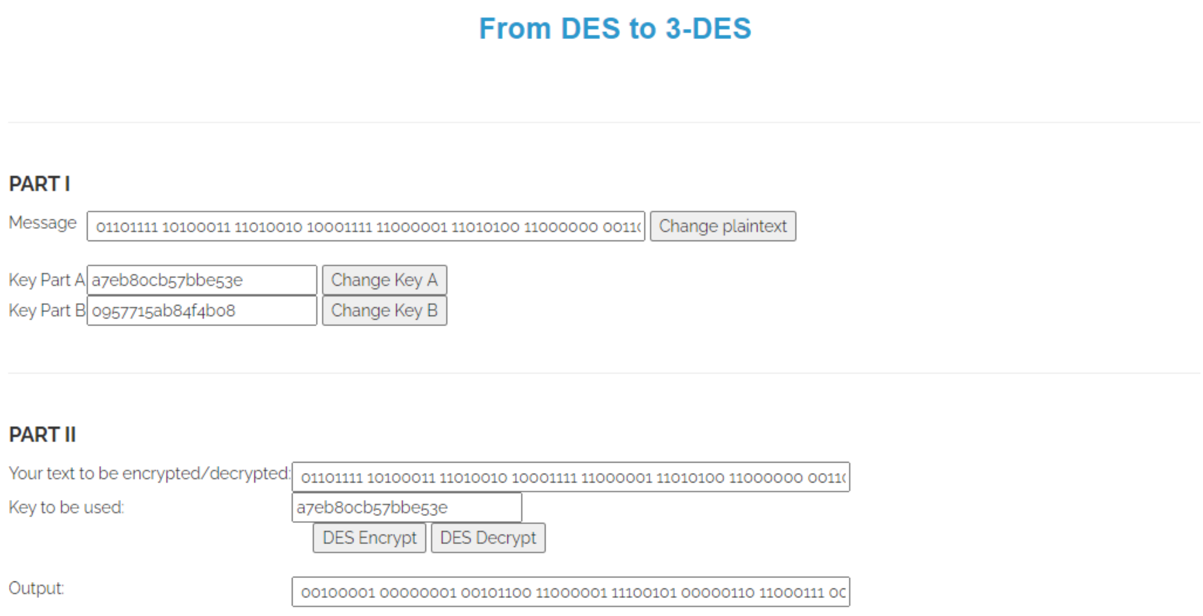
PART II

Your text to be encrypted/decrypted:

Key to be used:

Output:

3.
Enter generated Plaintext **m** from **PART I** to **PART II** in "Your text to be encrypted/decrypted:" block. Enter generated **key A** from **PART I** to **PART II** "Key to be used:" block and click on DES encrypt button to output cipher text **c1**.



From DES to 3-DES

PART I

Message 01101111 10100011 11010010 10001111 11000001 11010100 11000000 0011

Key Part A a7eb80cb57bbe53e

Key Part B 0957715ab84f4b08

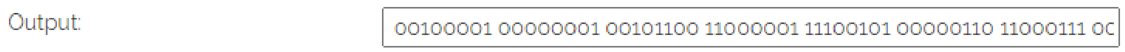
PART II

Your text to be encrypted/decrypted: 01101111 10100011 11010010 10001111 11000001 11010100 11000000 0011

Key to be used: a7eb80cb57bbe53e

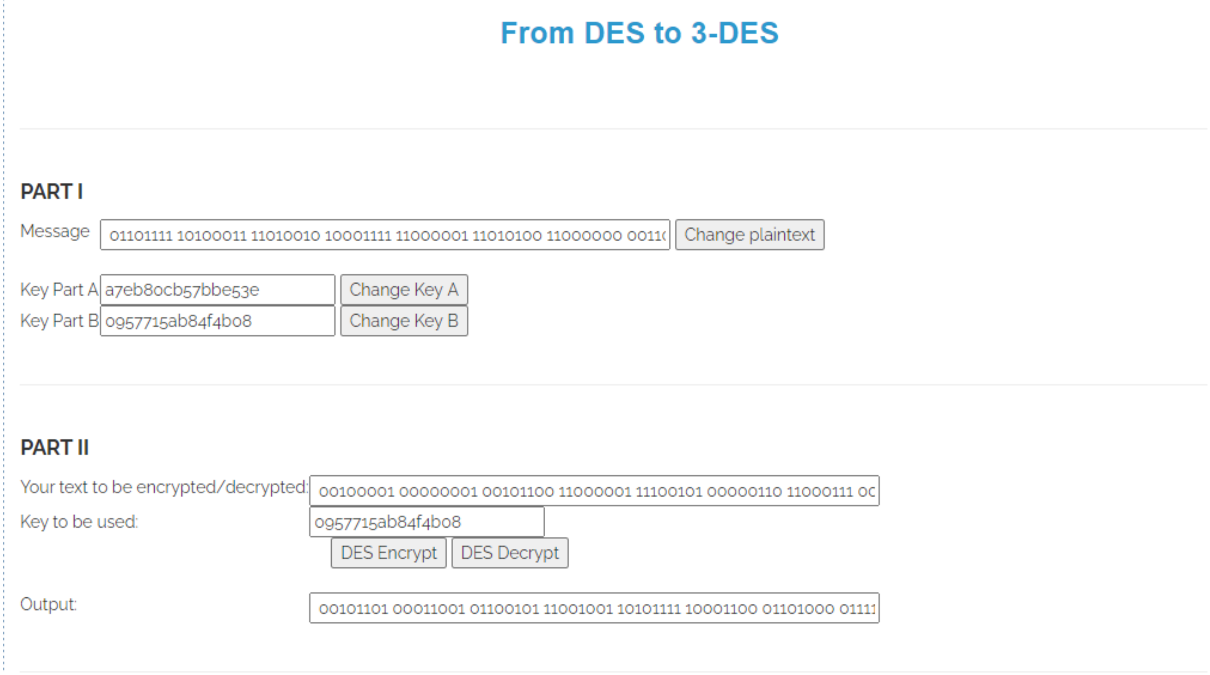
Output: 00100001 00000001 00101100 11000001 11100101 00000110 11000111 0

4.
This is First Encryption.



Output: 00100001 00000001 00101100 11000001 11100101 00000110 11000111 0

5.
Enter generated cipher text **c1** from **PART II** "Output:" Block to **PART II** in "Your text to be encrypted/decrypted:" block.
Enter generated **key B** from **PART I** to **PART II** in "Key to be used:" block and click on DES decrypt button to output cipher text **c2**.



From DES to 3-DES

PART I

Message 01101111 10100011 11010010 10001111 11000001 11010100 11000000 0011

Key Part A a7eb80cb57bbe53e

Key Part B 0957715ab84f4b08

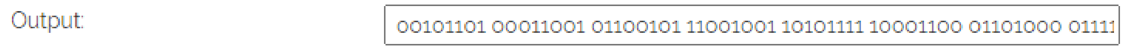
PART II

Your text to be encrypted/decrypted: 00100001 00000001 00101100 11000001 11100101 00000110 11000111 0

Key to be used: 0957715ab84f4b08

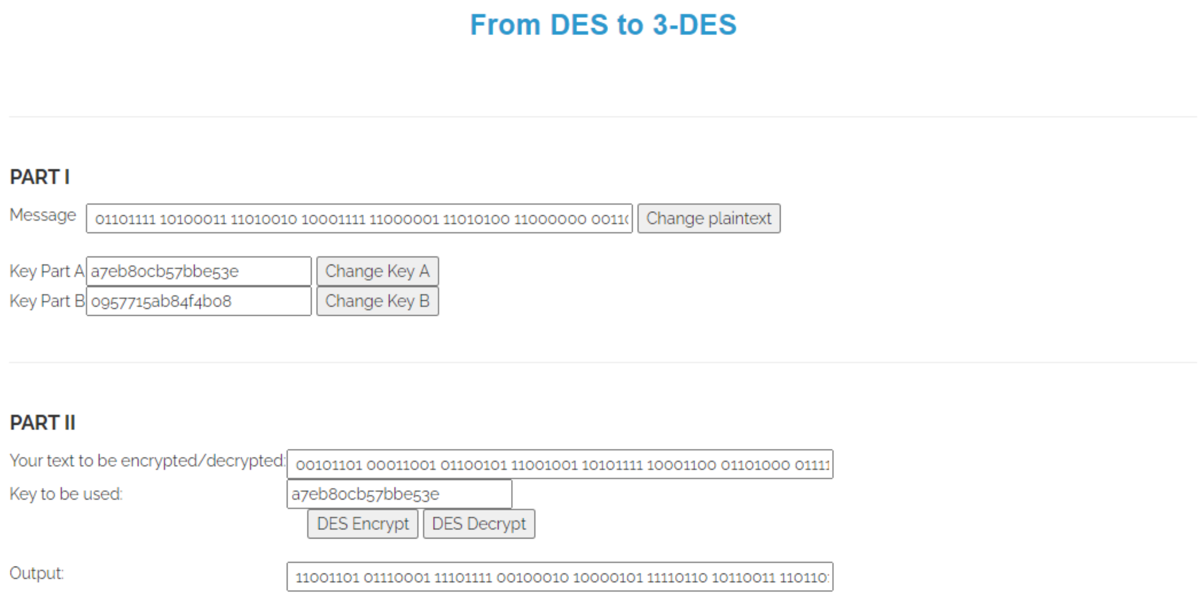
Output: 00101101 00011001 01100101 11001001 10101111 10001100 01101000 0111

6.
This is Second Encryption.



Output: 00101101 00011001 01100101 11001001 10101111 10001100 01101000 0111

7.
Enter generated cipher text **c2** from **PART II** "Output:" block to **PART II** in "Your text to be encrypted/decrypted:" block.
Enter generated **key A** from **PART I** to **PART II** "Key to be used:" block and click on DES encrypt button to output cipher text **c3**.



From DES to 3-DES

PART I

Message 01101111 10100011 11010010 10001111 11000001 11010100 11000000 0011

Key Part A a7eb80cb57bbe53e

Key Part B 0957715ab84f4b08

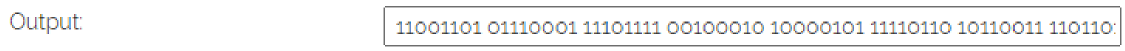
PART II

Your text to be encrypted/decrypted: 00101101 00011001 01100101 11001001 10101111 10001100 01101000 0111

Key to be used: a7eb80cb57bbe53e

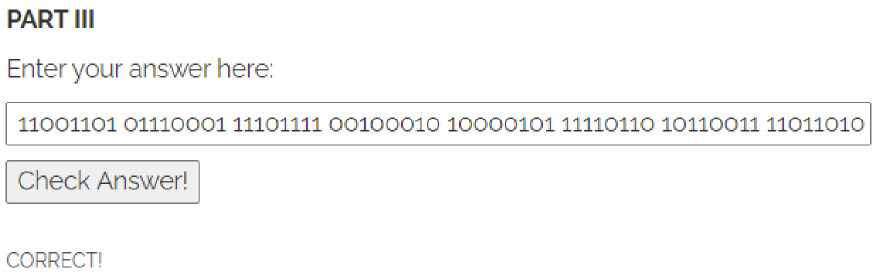
Output: 11001101 01110001 11101111 00100010 10000101 11110110 10110011 11011

8.
This is Third Encryption. As Encryption is done thrice. This Scheme is called triple DES.



Output: 11001101 01110001 11101111 00100010 10000101 11110110 10110011 110110

9.
Enter generated cipher text **c3** from **PART II** "Output:" Block to **PART III** "Enter your answer here:" block in order to verify your Triple DES.



PART III

Enter your answer here:

11001101 01110001 11101111 00100010 10000101 11110110 10110011 11011010

CORRECT!