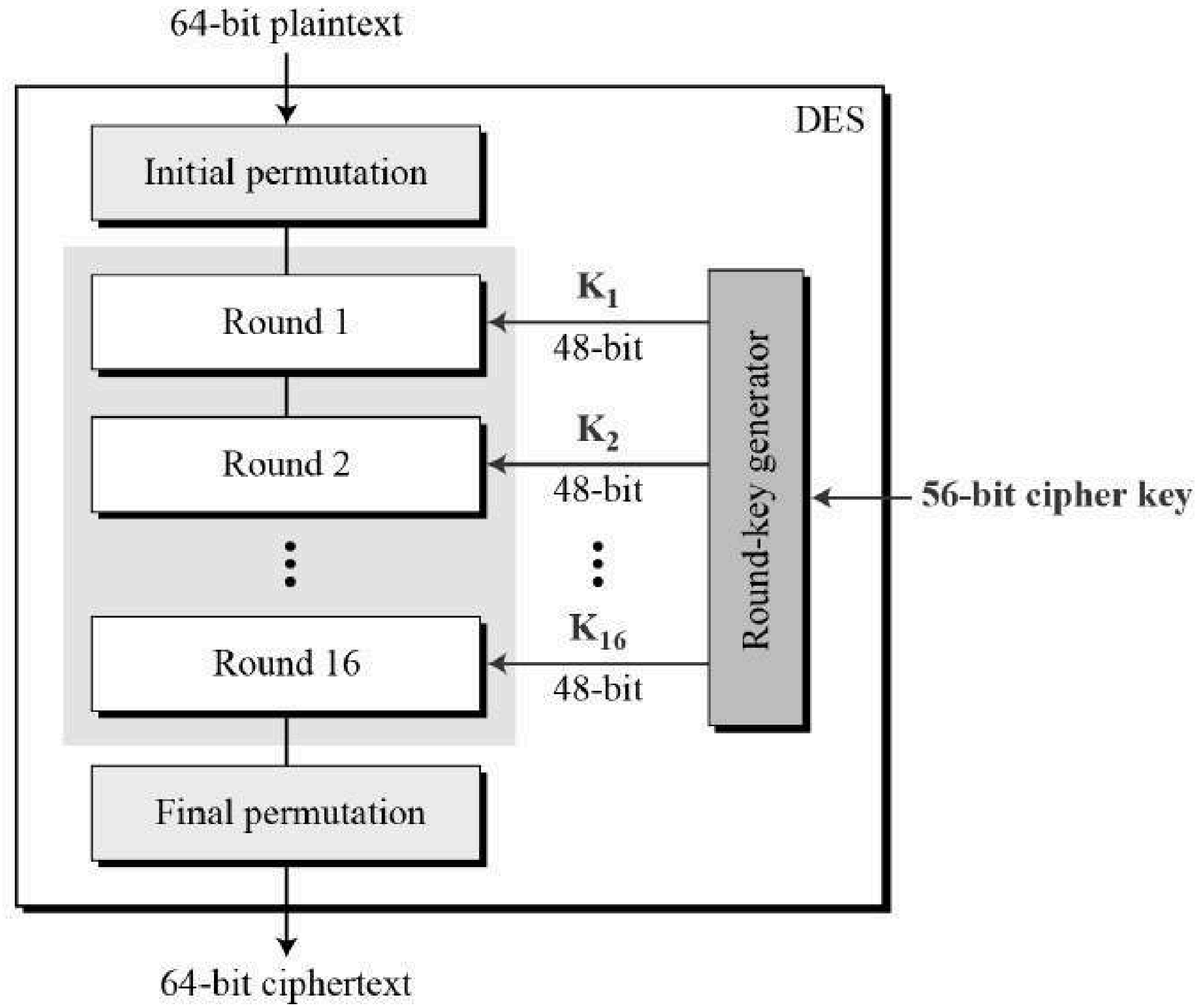
Data Encryption Standard (DES)

Theory:



Initial Permutation (IP):

- The 64-bit plaintext is permuted according to a fixed permutation table.
- The bits are rearranged to make the data suitable for further processing.

Key Generation:

- The 56-bit encryption key is expanded and modified to create 16 subkeys, one for each round.
- Each subkey is 48 bits long, and they are derived through a process of permutation and shifting.

Rounds (16 rounds in total):

- The data is divided into two 32-bit blocks: the left and right halves.
- The right half is expanded to 48 bits using an expansion permutation.
- The expanded right half is XORed with the round's subkey.
- The result goes through substitution using eight S-boxes, which replace 6-bit groups with 4 bits based on fixed tables.
- The outputs from the S-boxes are concatenated and subjected to a fixed permutation.
- The result is XORed with the left half.
- The left and right halves are swapped, and the process is repeated for 16 rounds.

Final Permutation (FP):

- After 16 rounds, the left and right halves are swapped one last time.
- The final permutation is applied to undo the initial permutation and obtain the ciphertext.

Task:

Lab 3 -DES

DES Encryption:

1. We import the necessary modules from PyCryptodome.
2. The pad_text function ensures that the plaintext length is a multiple of 8 bytes to match the DES block size. It appends padding bytes to the plaintext.
3. The des_encrypt function performs DES encryption in ECB mode using the provided key.
4. In the main function, we define the plaintext and generate a random DES key.
5. The plaintext is padded, encrypted, and the ciphertext is printed.

DES Decryption:

1. We import the necessary modules from PyCryptodome.
2. The unpad_text function removes the padding bytes to retrieve the original plaintext.
3. The des_decrypt function decrypts the ciphertext using the provided key.
4. In the main function, replace the ciphertext and key variables with the actual ciphertext and key used for encryption.
5. The ciphertext is decrypted, and the original text is obtained by removing the padding.
6. The decrypted text is printed.

Implement Triple DES Encryption using:

Plaintext = I am Batman
KeyA = `b'\x01\xadWR\xeb\x1a\xa2\x86'`
KeyB = `b'\xf7\xcf\xd6r\xd9\xa1\x141'`

Your Output should be
`b'x\t\x8c\xc2h\x06G\xacA\x93=\xfe\xb6\x13[\x9a\xac\xe5\\l\x93}\x17\xab'`

CBC-MAC

Task:
You are given 3 information: a message, key, and CBC-MAC signature. Your task is to verify whether the received message is valid or not.

Message	Key	MAC Signature	Validity
`I met an interesting turtle while the song on the radio blasted away`	`b'\x01\xd8i\xa1^0\x9a<\x0f\xf0\r\xc1\xdd\xd5\x89\xa6'`	`ba4ecb8db45c6ae0`	
`I like to leave work after my eight-hour tea-break`	`b'\xa6+\x16\x9d-1\xda\x8aV\xed\xf5\xf0cv\x04\x88'`	`f47e78c537fa1435`	
`Her daily goal was to improve on yesterday`	`b'[\xc5\xbd\xe4z\xd1=E\x17-ku\x02=|='`	`ddaf3152edbe868a`	
`He found the chocolate covered roaches quite tasty`	`b'5"k\xff\x81a\x9b7\x8c>\xb7\xb9\xdcu\xaa'`	`9d30d856f84489a8`	

After fighting off the alligator, Brian still had to face the anaconda	b'\xa1\xfcw"?3\x91\x1c\t\x9c\x91\xe2He\x935'	b9d173e05bbf7738	
He decided to count all the sand on the beach as a hobby	b'\xa7\x83@\xde\xbf\xb494\xee\x84\x1e-\xc8A\xf9:'	6355e471bd9930a1	
The sign said there was road work ahead so he decided to speed up	b'2\xcbv\xdcU6\x99\xb6.\xa7\xea\xeb\xaf\x10\xc7\x90'	9fbafc75e0a5056a	
Send 500$ to this account - 6589415651548	b'\xc3\xea\x99e\xaal\xab\xd4\x9b\xf9\xb4Z\x19\xed\xcf\xcb'	35273149636aca35	
Garlic ice-cream was her favorite	b'\x05\xf9\x83\x9d\xb7\xb6\xc3\xb8\x9e\xc5\xd9\xd8\x07]\xc6\xb3'	dc2de1e07b71d391	
I'd rather be a bird than a fish	b'\x84YY\xf0\x02GU\xa4LD\xd5\x85!A\xc2c'	5e191d02aa5fc0b1	

Procedure:

Colab Notebook Link for this lab:

https://colab.research.google.com/drive/144Xy0LbXip8Z6mlD_RTLEsitMkBg_n8y#scrollTo=BmNKrVK6p9pp