HACETTEPE UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BBM465 2022 FALL

Assignment-1

October 30, 2022

Tuğrul ACAR
2210356144

Alper SOLMAZ
2200356039

Problem

In this project, we are expected to create a FileCipher java program that uses four encryption modes which are CBC, CFB, OFB, CTR and two encryption algorithms which are DES and 3DES. We have to use ECB scheme for these implementations.

Important Notes for the Assignment

- We used JAVA 14 and 15 versions. Our program works as specified on this versions.
- If input file contains new line(/n) program discards it in the decyrption and prints all input as a single line.
- Program splits the key file with “ - ” (spacedashspace) characters. So this character sequence should not be in the keys.
- Detailed description of all functions given in the code.

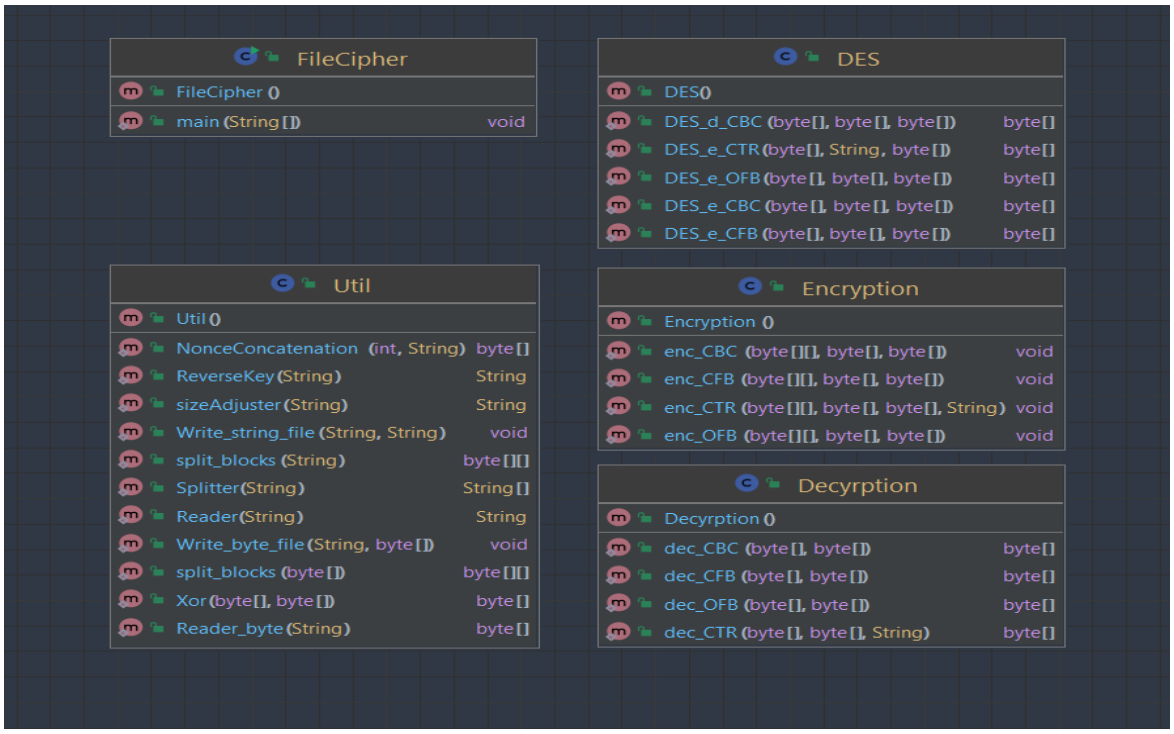
Method and Solution of Problem

Firstly program by using the “Util. reader()” method, reads the key_file and input file and sends them to strings. After that we split the key string by using the “space dash space” (“ - ”) character sequence and send them to the string array. The program split the input_string into 8-byte arrays. If the last array is less than 8 bytes we concatenate the space character “ ” to make it 8 byte.

By looking at arguments, the program chooses encryption algorithms(DES and 3DES), E or D, and one of them encryption modes.

By using the DES class we define CBC, CFB, OFB, and CTR mods based on ECB. In encryption decryption classes program takes input_arrays and makes a “for loop” to use CBC, CFB, OFB, and CTR mods for each 8-byte array and returns a byte array. If it is an encryption mode it writes the result to the output file as bytes. However, if it is a decryption mode it converts the result to a string and writes to the output file.

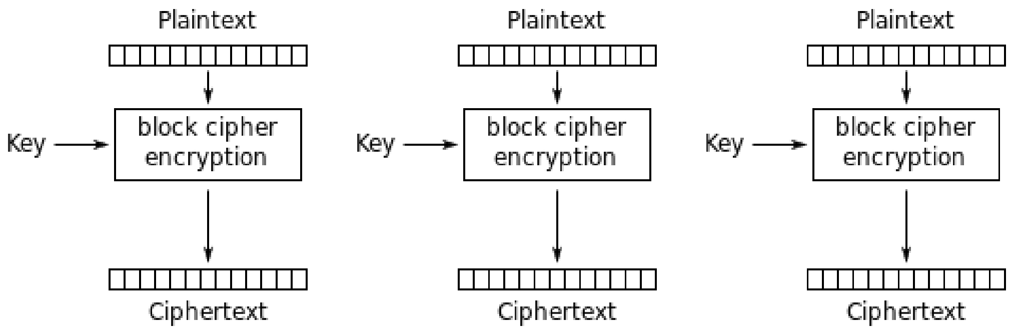
If we use the 3DES algorithm, the program calls the same methods. However, for the second step, we use a different key for the enc or dec method. We used a method to reverse the original key and use it for the second stage.



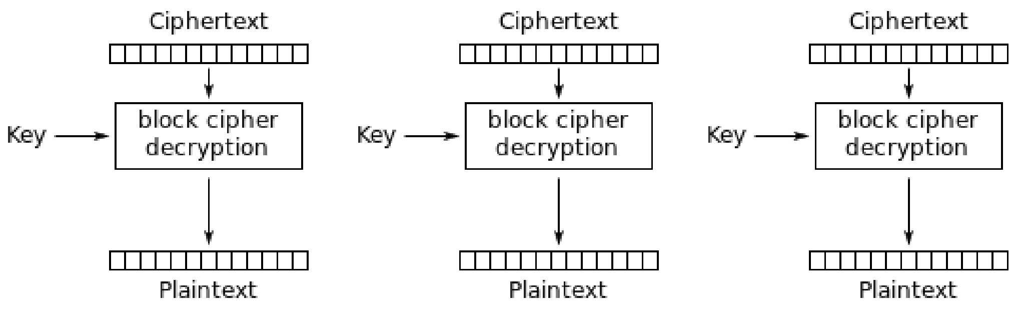
```
FileCipher
  FileCipher ()
  main (String [])                                   void

DES
  DES()
  DES_d_CBC (byte[], byte [], byte[])                byte[]
  DES_e_CTR (byte[], String, byte [])                byte[]
  DES_e_OFB (byte [], byte[], byte [])               byte[]
  DES_e_CBC (byte [], byte [], byte [])              byte[]
  DES_e_CFB (byte[], byte [], byte [])               byte[]

Util
  Util ()
  NonceConcatenation (int, String)                   byte []
  ReverseKey (String)                                String
  sizeAdjuster (String)                              String
  Write_string_file (String, String)                 void
  split_blocks (String)                              byte [][]
  Splitter (String)                                  String []
  Reader (String)                                    String
  Write_byte_file (String, byte [])                  void
  split_blocks (byte [])                             byte [][]
  Xor (byte[], byte [])                              byte []
  Reader_byte (String)                               byte []

Encryption
  Encryption ()
  enc_CBC (byte [][], byte[], byte [])               void
  enc_CFB (byte [][], byte [], byte[])               void
  enc_CTR (byte [][], byte [], byte[], String)       void
  enc_OFB (byte[][], byte[], byte [])                void

Decryption
  Decryption ()
  dec_CBC (byte [], byte[])                          byte[]
  dec_CFB (byte [], byte [])                         byte[]
  dec_OFB (byte[], byte [])                          byte[]
  dec_CTR (byte [], byte [], String)                 byte[]
```

Electronic Codebook (ECB)

The simplest of the encryption modes is the **electronic codebook** (ECB). The message is divided into blocks, and each block is encrypted separately by using key.We used this to impelement other 4 mods (CBC, CFB, OFB,CTR).



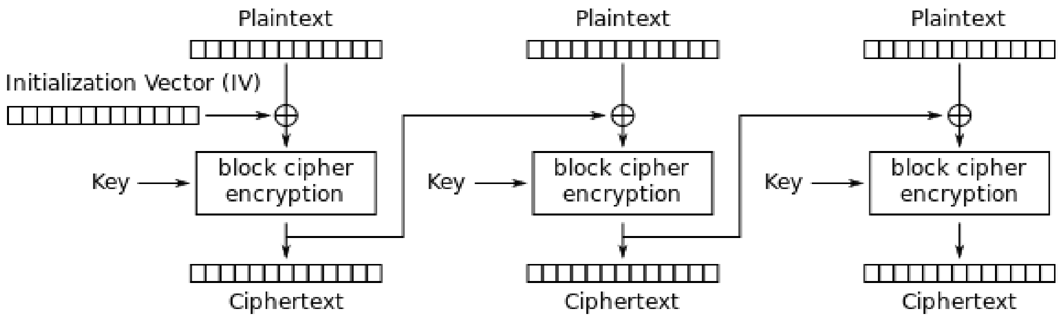
Electronic Codebook (ECB) mode encryption



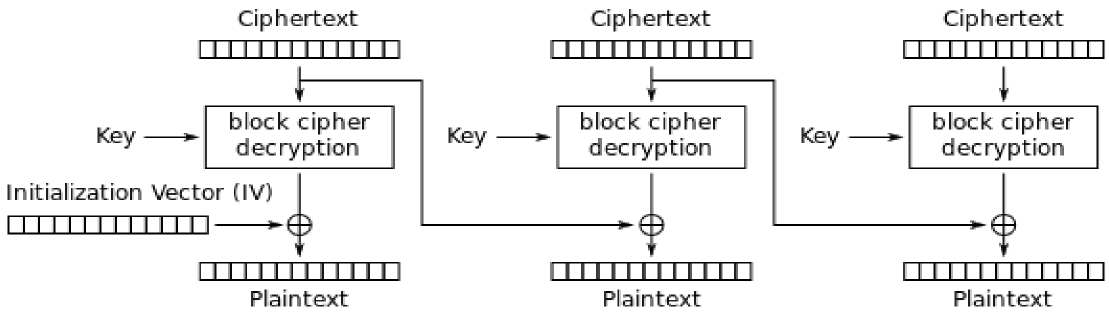
Electronic Codebook (ECB) mode decryption

Cipher Block Chaining (CBC)

In CBC mode, each block of plaintext is XOR'ed with the previous ciphertext block before being encrypted. The first part was XOR'ed with an initialization vector. However, for the decryption process, we first decrypted cipher text and make xor for the result with IV. The next parts use cipher text as IV.



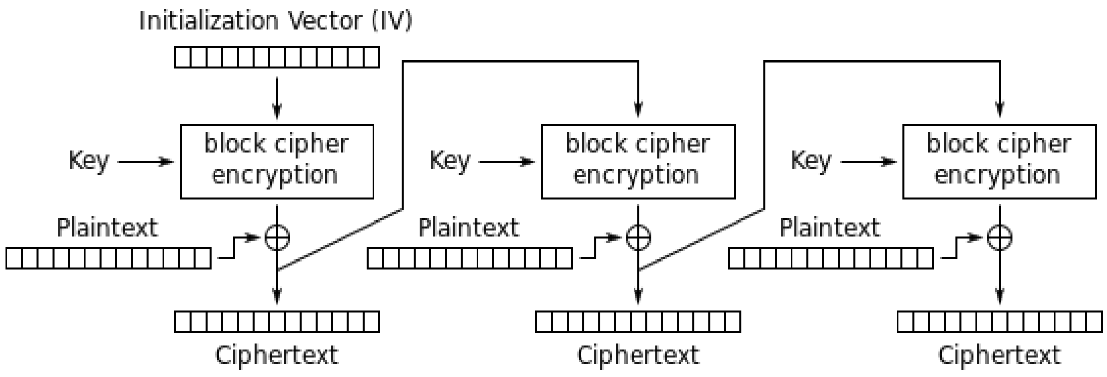
Cipher Block Chaining (CBC) mode encryption



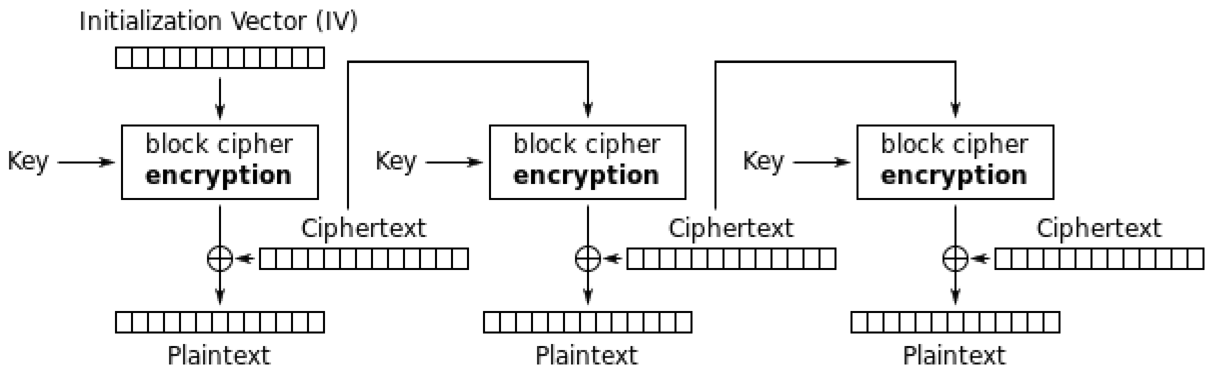
Cipher Block Chaining (CBC) mode decryption

Cipher Feedback (CFB)

In CFB mode, firstly IV is encrypted, then XORed with plaintext. The next part uses this result as an IV and makes the same process. However, for the decryption process, IV is encrypted again and XORed with ciphertext. The next part uses cipher text as IV and make the same process.



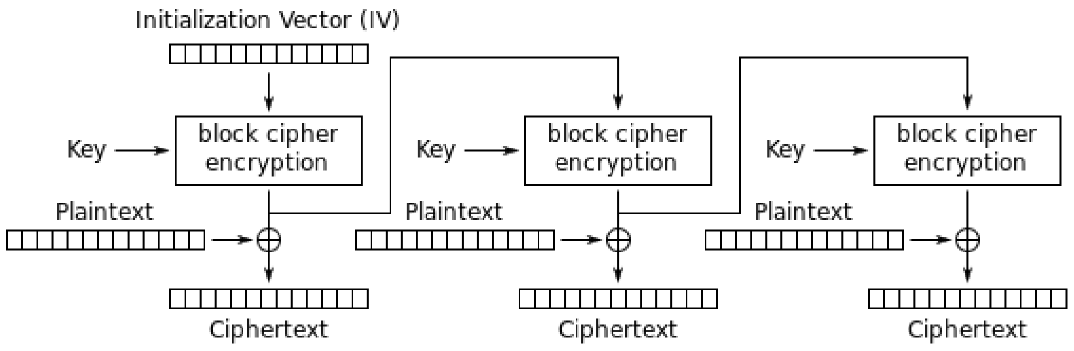
Cipher Feedback (CFB) mode encryption



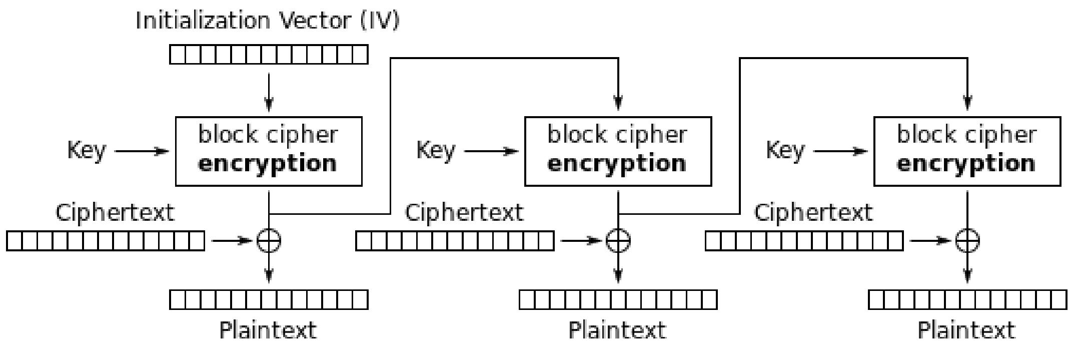
Cipher Feedback (CFB) mode decryption

Output Feedback (OFB)

In OFB mode, firstly IV is encrypted, then XORed with plaintext similar to CFB mode. However next part uses encrypted IV before XORed with plaintext and makes the same process again. In the decryption process, the process is almost the same but we use cipher text instead of plaintext, other methods are the same.



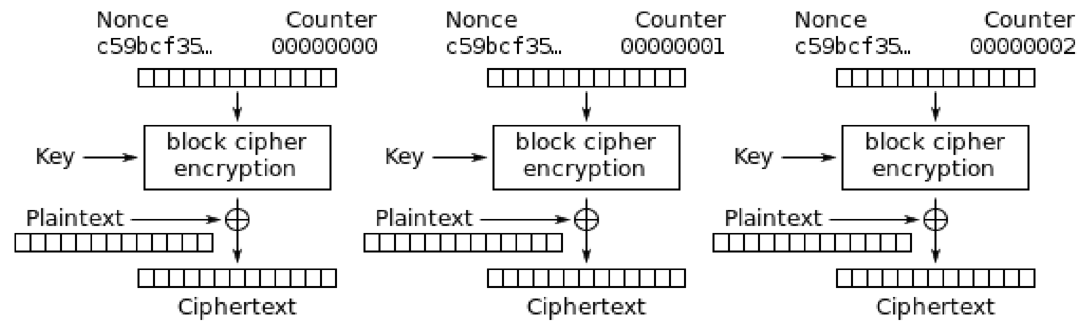
Output Feedback (OFB) mode encryption



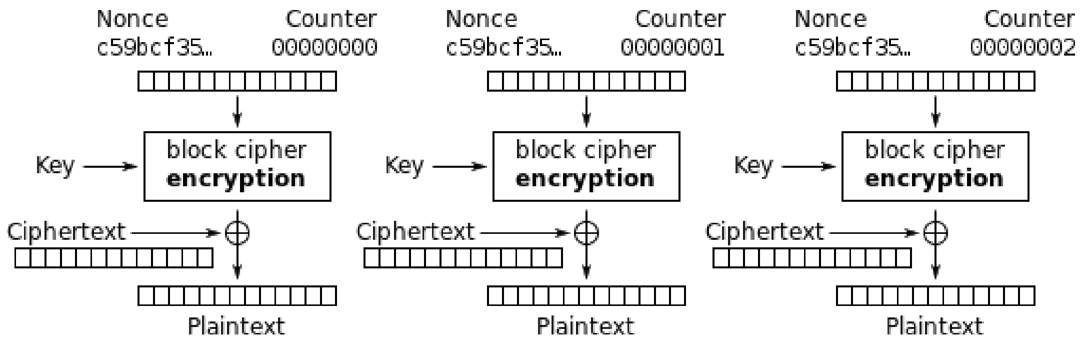
Output Feedback (OFB) mode decryption

Counter (CTR)

In CTR mode, we have a nonce and a counter different from other modes. Firstly combination of nonce and counter value is encrypted and the result is XORed with plaintext. For each part, the counter is updated and used to make the same process. The decryption process is almost the same but we use ciphertext instead of plaintext again. The other methods are the same as with the encryption process.



Counter (CTR) mode encryption



Counter (CTR) mode decryption

References:

https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation
https://en.wikipedia.org/wiki/Padding_(cryptography)
https://docs.oracle.com/javase/7/docs/api/javax/crypto/package-summary.html