AES project documentation

Alessio Yang, Gyeong Ju PARK

Theory background

AES

The **Advanced Encryption Standard (AES)** is a symmetric block cipher designed to secure digital data. It was standardized by the National Institute of Standards and Technology (NIST) in 2001 and it is based on the Rijndael block cipher developed by John Daemen and Vincent Rijmen.
Being a **symmetric cipher**, the same key is used for both encryption and decryption.
The plaintext is divided in 128 bit-size blocks, while the key length defines the type of AES used:

- AES-128 → 128-bit key, 10 rounds
- AES-192 → 192-bit key, 12 rounds
- AES-256 → 256-bit key, 14 rounds

A **128-bit round key** is used for each round, generated from the main key by using key expansion.
Both the round keys and the plaintext block are transformed into a 4x4 byte matrix:

b_0	b_4	b_8	b_{12}
b_1	b_5	b_9	b_{13}
b_2	b_6	b_{10}	b_{14}
b_3	b_7	b_{11}	b_{15}

The message matrix and all its future transformations are called **states**.

For calculations, **Galois field arithmetic** $GF(2^8)$ is used:

- $f(x) = x^8 + x^4 + x^3 + x + 1$ is used as divisor to represent the field: $Z_2[x]/f(x) = GF(2^8)$
- Every element of the field $g \in GF(2^8)$ can be represented as a **byte sequence**.
  Example: $x^7 + x^4 + x^3 + 1 \Leftrightarrow 1001\,1001 \Leftrightarrow 99_{16}$
- Addition and subtraction operations are performed in modulo 2 (XOR operation)
- The multiplication is reduced modulo $1\,0001\,1011 \Leftrightarrow f(x)$: $x^8 \equiv x^4 + x^3 + x + 1\ (mod\ 2)$

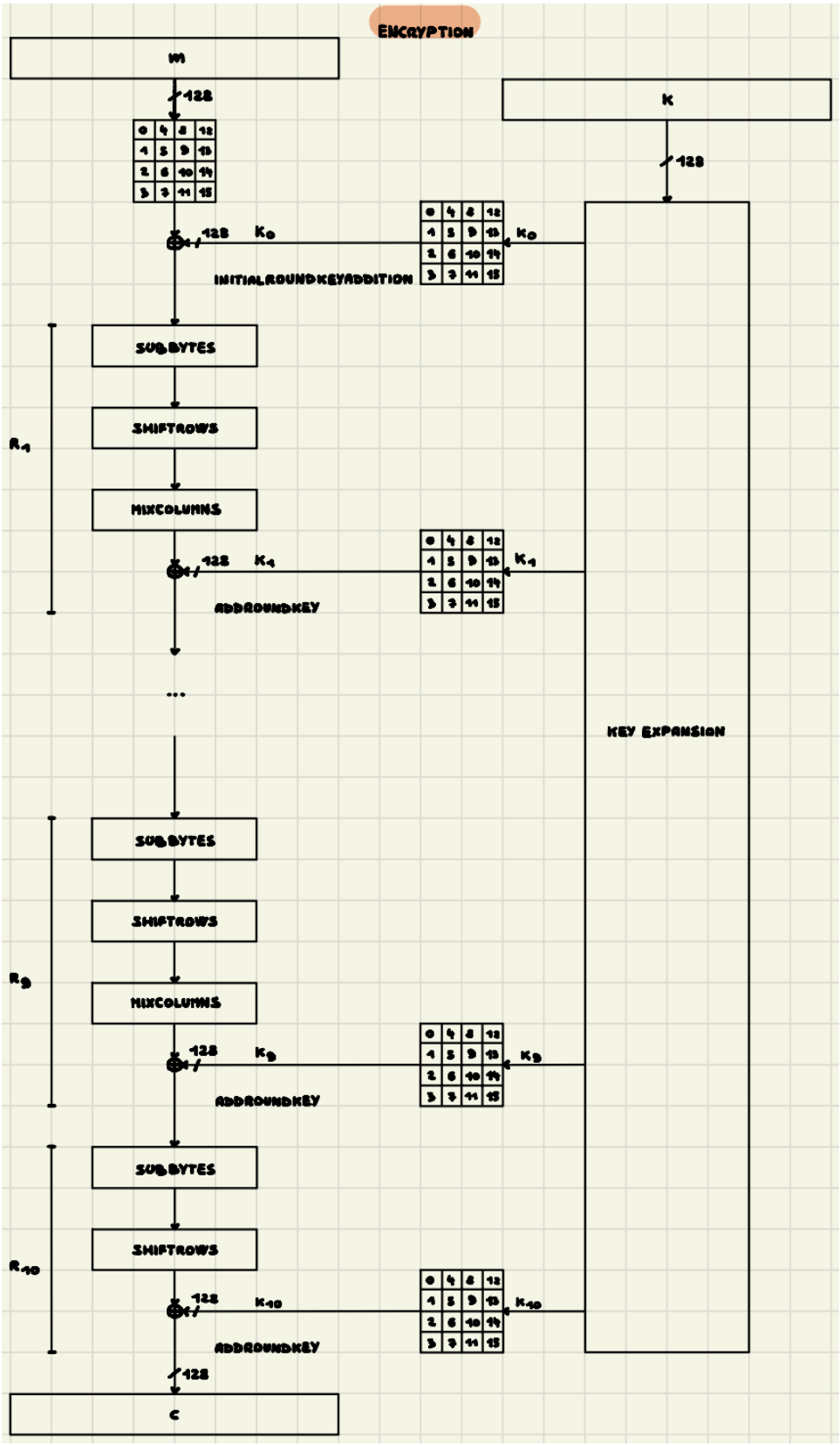
Steps for AES encryption (Message block m → Ciphertext c):

1. **Key expansion**: n round keys k_1,..., k_n + additional key k_0 are derived from the key k (n can be 10, 12 or 14 depending on the key-size)
2. **Initial round key addition**: Compute $m \oplus k_0$
3. From i round 1 to $n - 1$:
    a. **SubBytes**: It maps each byte of the state according to a known lookup table and it performs a non-linear substitution:
        i. It is implemented with an 8-bit S-box for each byte (16 S-boxes in total)
        ii. Dearrangement: No bytes are in their initial value
        iii. No opposites: No bytes are flipped
    b. **ShiftRows**:
        i. First row of the state is unchanged
        ii. Second row is left shifted by 1
        iii. Third row is left shifted by 2
        iv. Fourth is shifted by 3
    c. **MixColumns**: It operates on each column separately, mixing each element linearly.
    Given $c = (c_0, c_1, c_2, c_3)$, it performs the following calculation:

2	3	1	1		c_0		c'_0
1	2	3	1	*	c_1	=	c'_1
1	1	2	3		c_2		c'_2
3	1	1	2		c_3		c'_3

    $c' = (c'_0, c'_1, c'_2, c'_3)$ is the new column.
    d. **AddRoundKey**: Compute $m' \oplus k_i$
4. Round n: It performs SubBytes, ShiftRows and AddRoundKey, but the MixColumns step is skipped

After the round n, the output is the ciphertext c.

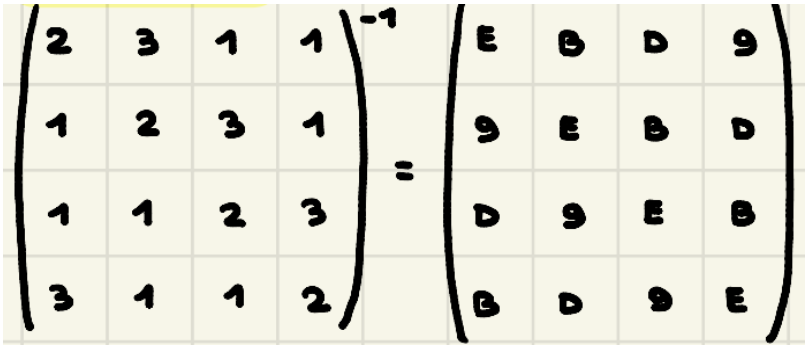


(*AES-128 encryption scheme*)

In the AES decryption, the round keys are used in inverse order (starting from k_n to k_0).

Steps for AES decryption (Ciphertext $c \rightarrow$ Message block m):

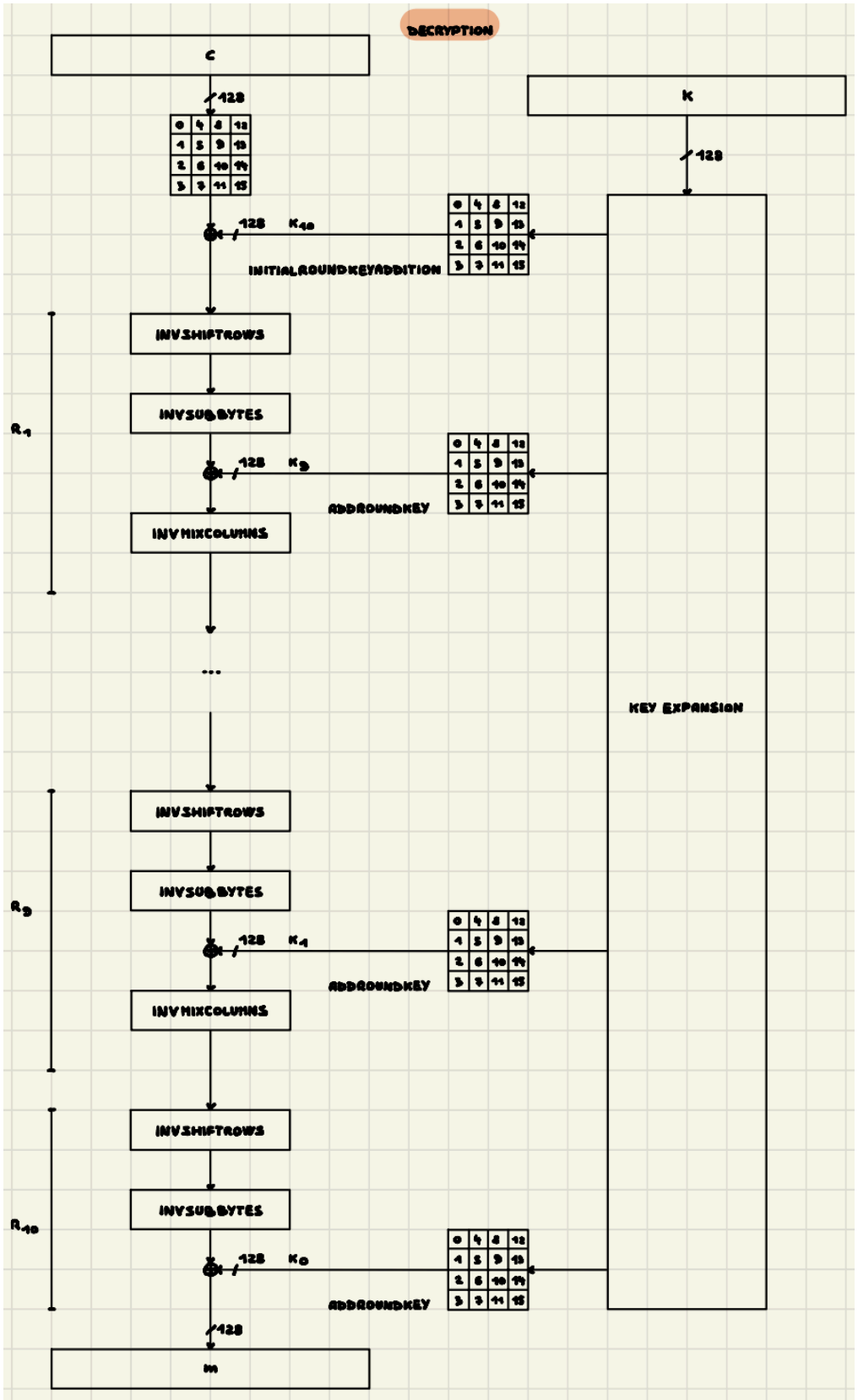
1. **Key expansion**
2. **Initial round key addition**: Compute $c \oplus k_n$
3. From i round 1 to $n - 1$:
    a. **InvShiftRows**: It performs the inverse of ShiftRows, via the right shifts
    b. **InvSubBytes**: It performs the inverse of SubBytes by using an inverse known lookup table
    c. **AddRoundkey**: Compute $c' \oplus k_{n-i}$
    d. **InvMixColumns**: It performs the inverse of MixColumns by inverting the known matrix:



E	B	D	9		c'_0		c''_0
9	E	B	D		c'_1		c''_1
D	9	E	B	*	c'_2	=	c''_2
B	D	9	E		c'_3		c''_3

4. Round n: It computes only InvShiftRows, InvSubBytes and AddRoundKey, while InvMixColumns is skipped

After the round n, the output is the plaintext block m.



(*AES-128 decryption scheme*)

Key expansion

Given a key k, we get the round keys + 1, all 128 bit-sized (16 bytes or 4 words)
The **round constants** are defined as follows:

i	1	2	3	4	5	6	7	8	9	10
rc_i	01	02	04	08	10	20	40	80	1B	36

$\forall i \in \{1,\dots,10\}: rcon_i := [rc_i, 00, 00, 00] \in GF(2^8)^4, rc_i \in GF(2^8)$

Let N be the number of words of the key (it can be 4, 6 or 8).
Split the original key in N words $k_0,\dots,k_{N-1}$.

Let R be the number of round keys needed + 1 (It can be 11, 13 or 15).
Let $W_0,\dots,R_{4R-1}$ be the words of the expanded key.

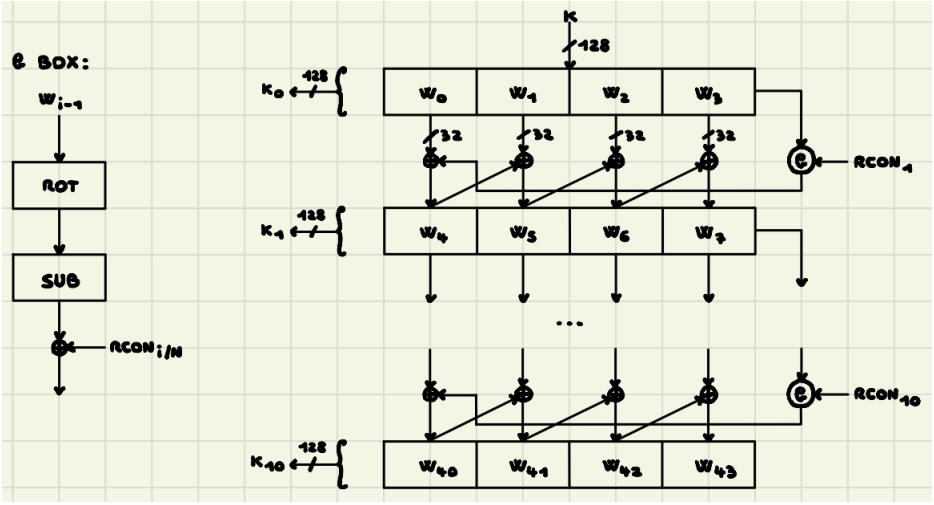
Let's define two word operations:

- $RotWord([b_0, b_1, b_2, b_3]) = [b_1, b_2, b_3, b_0]$ → Left circulare byte-wise shift
- $SubWord([b_0, b_1, b_2, b_3]) = [S(b_0), S(b_1), S(b_2), S(b_3)]$ → Application of the AES S-box

Therefore, in order to calculate the expanded key, the following calculation is performed:
$W_i =$

- $k_i, i < N$
- $W_{i-N} \oplus SubWord(RotWord(W_{i-1})) \oplus rcon_{i/N}, i \geq N \wedge i \equiv 0 (mod\ N)$
- $W_{i-N} \oplus SubWord(W_{i-1}), i \geq N \wedge N > 6 \wedge i \equiv 4 (mod\ N)$
- $W_{i-N} \oplus W_{i-1}$, otherwise

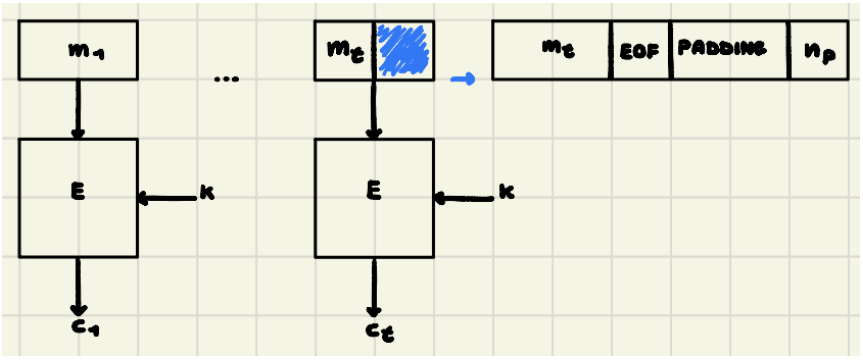
The round keys are:
$k_0 = [W_0, W_1, W_2, W_3], \dots, k_{R-1} = [W_{4R-4}, W_{4R-3}, W_{4R-2}, W_{4R-1}]$



(*AES-128 key expansion scheme*)

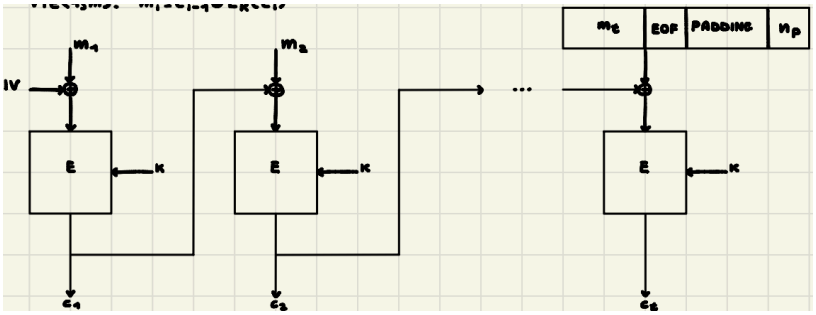
ECB and CBC

Modes of operation define how a block cipher like AES is applied to larger amounts of data. Since AES encrypts data in fixed-size blocks (128 bits), modes of operation manage how to handle data that may exceed or fall short of this block size. They also influence how ciphertext is generated and provide security properties like confidentiality and authentication.
Before applying the mode of operation, it is assumed that a **padding mechanism** has already been applied to the plaintext block so that its length is a multiple of 128 bits.
In the **Electronic CodeBook (ECB) mode**, each block of the plaintext is encrypted independently using the same key. Despite being very easy to implement, it is very insecure because of the weak patterns in the data.

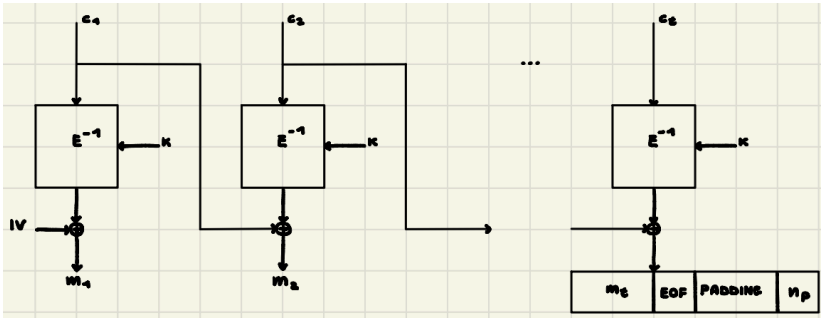


(*ECB mode scheme*)

In the **Cipher Block Chaining (CBC) mode**, each plaintext block is XORed with the previous ciphertext block before the encryption. The first block uses an **initialization vector (IV)** chosen by the user. It increases the security of the algorithm if the IV is random, but it prevents the cipher to be performed in parallel, that is, the cipher is not parallelizable.



(*CBC mode scheme encryption*)



(*CBC mode scheme decryption*)

Functional description of the application

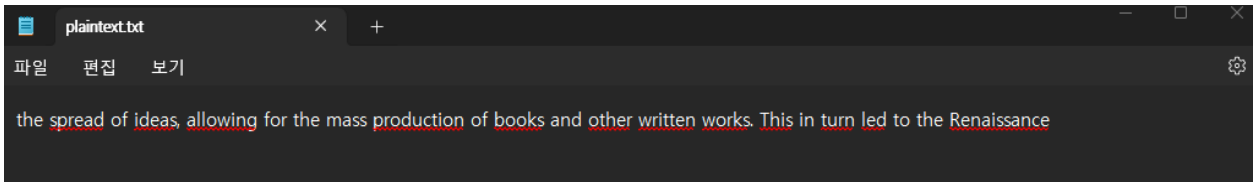
Input data format:

```
KEY_FILENAME = 'key.txt'
PLAINTEXT_FILENAME = 'plaintext.txt'
CIPHERTEXT_FILENAME = 'ciphertext.txt'
IV_FILENAME = 'initialization_vector.txt'
```

The text file to be used can be assigned to a variable KEY_FILENAME, PLAINTEXT_FILENAME, CIPHERTEXT_FILENAME, IV_FILENAME.

The format of the input data should be as follows:

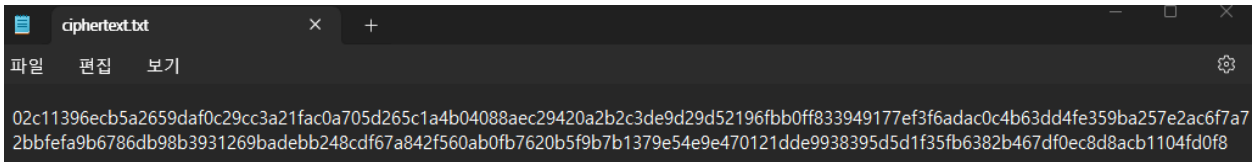
- Standard text in UTF-8 format for a plain text message.



- 16 bytes for AES-128, 24 bytes for AES-192, and 32 bytes for AES-256 for the key.

- Must be 16 bytes in length for the IV (Initialization Vector)

- The cipher text should be in hexadecimal format and must satisfy the condition len(ciphertext) % 32 = 0.



- The AES type can be selected using 0, 1, or 2.

```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 0
```

- The operation mode (encryption/decryption) can be selected using 0 or 1.

```
Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 1
```

- You can select whether to encrypt or decrypt using 0 or 1.

```
Choose the operation you want to perform ['0: ENCODE', '1: DECODE'] -> 0
```

Output data format:

- Console output after selecting the AES type

```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 0
You have chosen the option AES128.
Number of rounds: 10 + 1
Size of the key: 16
```

- Console output after selecting the AES mode

```
Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 1
You have chosen the option CBC.

Recovering the key from the file key.txt...
Key recovered:
a period of unpr
Checking the key correctness...
The key is in a correct format.

Recovering the initialization vector from the file initialization_vector.txt...
Initialization vector recovered:
and cultural gro
Checking the iv correctness...
The iv is in a correct format.
```

- Console output after selecting encryption or decryption mode

```
You have chosen the option ENCODE.

Recovering the plaintext from the file plaintext.txt...
Plaintext recovered:
the spread of ideas, allowing for the mass production of books and other written works. This in turn led to the Renaissance

Start encryption...

Ciphertext:
02c11396ecb5a2659daf0c29cc3a21fac0a705d265c1a4b04088aec29420a2b2c3de9d29d52196fbb0ff833949177ef3f6adac0c4b63dd4fe359ba257e2ac6f7a72bbfefa9b6786db98b3931269badebb248cdf67a842f560ab0fb7620b5f9b7b1379e54e9e470121dde9938395d5d1f35fb6382b467df0ec8d8a
cb1104fd0f8

Saving the ciphertext to file ciphertext.txt...
Ciphertext saved successfully.
```

Description of designed code structure

1) Choice of the AES type

You can select and input one of the values either AES-128, AES-192, or AES-256. The selected value is stored in the variable **aes_choice** and the number of rounds and the key size are determined based on this value.

2) Choice of mode of operation

You can select one of the AES encryption modes either ECB or CBC. The selected value is stored in the variable **mode_choice** and the encryption and decryption method is determined based on this value.

3) Key recovering

This step involves loading the key stored in a text file. The file name should be stored in **KEY_FILENAME**. For CBC mode, an initialization vector (IV) is also required, which should be loaded from **IV_FILENAME**.

The key length must match the chosen AES type. It is 16 bytes for AES-128, 24 bytes for AES-192, and 32 bytes for AES-256.

The key and IV are validated using the functions **key_correctness_check(key)** and **iv_correctness_check(iv)** to ensure they meet the specified conditions.

4) Choice of operation to perform

You decide whether to encrypt or decrypt using the key. The value you input is stored in the variable **operation_choice** and the encryption or decryption process is performed based on this value.

5) Encryption Instructions

The plaintext is loaded using the **get_plaintext_from_file()** function. The plaintext message to be encrypted is processed using: **aes_encrypt(key, plaintext)** for ECB mode or **aes_encrypt(key, plaintext, iv)** for CBC mode (selected in step 2, Choice of mode of operation).

The encrypted plaintext message is then saved using the **write_ciphertext_to_file(ciphertext)** function.

6) Decryption Instructions

The ciphertext message is loaded using the **get_ciphertext_from_file()** function. The ciphertext message to be decrypted is processed using **aes_decrypt(key, ciphertext)** for ECB mode or **aes_decrypt(key, ciphertext, iv)** for CBC mode (selected in step 2, Choice of mode of operation).

The function **ciphertext_correctness_check(ciphertext)** ensures that the ciphertext meets the required conditions. The decrypted ciphertext message is saved using the **write_plaintext_to_file(plaintext)** function.

Input:

The AES type, AES mode, and operation type (encryption/decryption) can be entered via the console using numeric keys.

The variables KEY_FILENAME, PLAINTEXT_FILENAME, CIPHERTEXT_FILENAME, and IV_FILENAME store the file names of the text files containing the key, plaintext message, ciphertext message, and IV.

Output:

The console will display the values entered by the user: AES type, AES mode, operation type (encryption/decryption), along with the number of rounds and the key size. It will also display the loaded key and IV.

The ciphertext message and plaintext message will be saved to the respective text files after encryption or decryption.

Functions used in the Encryption-Decryption steps

states_to_ciphertext(states): convert arrays of state to sequence of bytes in hexadecimal format

pkcs_padding(plaintext): adds pkcs padding to a plain text message to set its length with a specific block size
 # PKCS#7 padding method (https://www.ibm.com/docs/en/zos/2.4.0?topic=rules-pkcs-padding-method)

sub_bytes(state): substitutes each byte in the state array to AES S-BOX
 # AES S-box (https://en.wikipedia.org/wiki/Rijndael_S-box)

shift_rows(state): cyclically shifts the last three rows of the state array to the left by different offsets

mix_columns(state): linearly mixes each column of the state array

encrypt_state(round_keys, state): performs AES encryption in single block

aes_encrypt(key, plaintext, iv = None): performs AES encryption using either ECB or CBC mode

pkcs_unpadding(ciphertext): removes pkcs padding from a plain text message

states_to_byte_plaintext(states): converts byte states to byte array

inv_sub_byte(byte): inverse substitutes using AES Inverse S-BOX
 # AES Inverse S-box (https://en.wikipedia.org/wiki/Rijndael_S-box)

inv_shift_rows(state): inverse shifts each row to the right

inv_sub_bytes(state): inverse S-BOX substitution to state

inv_mix_columns(state): inverse mix column step
 # Fixed inverse mix column matrix (https://www.researchgate.net/figure/nverse-MixColumns-stage-in-the-traditional-AES-operation_fig5_349016516)

decrypt_state(round_keys, state): performs AES decryption in single block

aes_decrypt(key, ciphertext, iv=None): performs AES decryption using either ECB or CBC mode

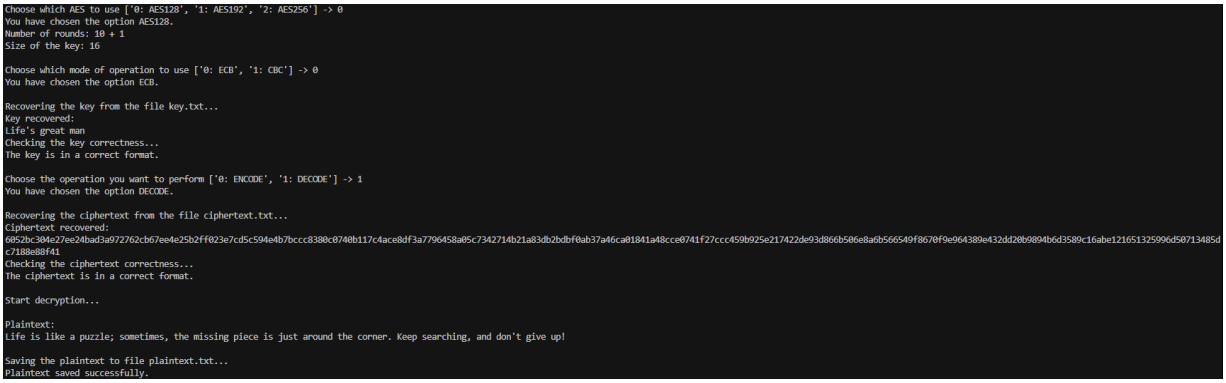
Testing

- Test in ECB modes for AES-128

Key: Life's great man
Plain text: Life is like a puzzle; sometimes, the missing piece is just around the corner. Keep searching, and don't give up!
Cipher text:
6052bc304e27ee24bad3a972762cb67ee4e25b2ff023e7cd5c594e4b7bccc8380c0740b117c4ace8df3a7796458a05c7342714b21a83db2bdbf0ab37a46ca01841a48cce0741f27ccc459b925e217422de93d866b506e8a6b566549f8670f9e964389e432dd20b9894b6d3589c16abe121651325996d50713485dc7188e88f41

Encryption

Decryption



Key: I really see AES
Plain text: The rain fell gently, while the scent of coffee filled the air.
Ciphertext: 18346015f1bcf0a464f77687dfd47450

Encryption

```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 0
You have chosen the option AES128.
Number of rounds: 10 + 1
Size of the key: 16

Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 0
You have chosen the option ECB.

Recovering the key from the file key.txt...
Key recovered:
I really see AES
Checking the key correctness...
The key is in a correct format.

Choose the operation you want to perform ['0: ENCODE', '1: DECODE'] -> 0
You have chosen the option ENCODE.

Recovering the plaintext from the file plaintext.txt...
Plaintext recovered:
The rain fell gently, while the scent of coffee filled the air.

Start encryption...

Ciphertext:
b4cdec4c8e81b7b55d7adc90c34c76a4260d0a5fd7d8e5c1cb28ad76028995d6041678ac3f91ef8dc4648b6a0f4bd5c976e6dfd462e8dc4c8458c04495a67060

Saving the ciphertext to file ciphertext.txt...
Ciphertext saved successfully.
```

Decryption

```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 0
You have chosen the option AES128.
Number of rounds: 10 + 1
Size of the key: 16

Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 0
You have chosen the option ECB.

Recovering the key from the file key.txt...
Key recovered:
I really see AES
Checking the key correctness...
The key is in a correct format.

Choose the operation you want to perform ['0: ENCODE', '1: DECODE'] -> 1
You have chosen the option DECODE.

Recovering the ciphertext from the file ciphertext.txt...
Ciphertext recovered:
b4cdec4c8e81b7b55d7adc90c34c76a4260d0a5fd7d8e5c1cb28ad76028995d6041678ac3f91ef8dc4648b6a0f4bd5c976e6dfd462e8dc4c8458c04495a67060
Checking the ciphertext correctness...
The ciphertext is in a correct format.

Start decryption...

Plaintext:
The rain fell gently, while the scent of coffee filled the air.

Saving the plaintext to file plaintext.txt...
Plaintext saved successfully.
```

- Test in CBC modes for AES-128

Key: Hold fast to hop
Initialization vector: The sun is high.
Plain text: The night sky is filled with endless possibilities.
Ciphertext:
6111a5494ec18b180d71871e15ad9ae5db7c54bea86d4f871da6b40dfd81fe54c2e7c4c1949595
73212174efac4451097791833ee7599428216df96492206a16

Encryption

```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 0
You have chosen the option AES128.
Number of rounds: 10 + 1
Size of the key: 16

Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 1
You have chosen the option CBC.

Recovering the key from the file key.txt...
Key recovered:
Hold fast to hop
Checking the key correctness...
The key is in a correct format.

Recovering the initialization vector from the file initialization_vector.txt...
Initialization vector recovered:
The sun is high.
Checking the iv correctness...
The iv is in a correct format.

Choose the operation you want to perform ['0: ENCODE', '1: DECODE'] -> 0
You have chosen the option ENCODE.

Recovering the plaintext from the file plaintext.txt...
Plaintext recovered:
The night sky is filled with endless possibilities.

Start encryption...

Ciphertext:
6111a5494ec18b180d71871e15ad9ae5db7c54bea86d4f871da6b40dfd81fe54c2e7c4c194959573212174efac4451097791833ee7599428216df96492206a16

Saving the ciphertext to file ciphertext.txt...
Ciphertext saved successfully.
```

Decryption

```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 0
You have chosen the option AES128.
Number of rounds: 10 + 1
Size of the key: 16

Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 1
You have chosen the option CBC.

Recovering the key from the file key.txt...
Key recovered:
Hold fast to hop
Checking the key correctness...
The key is in a correct format.

Recovering the initialization vector from the file initialization_vector.txt...
Initialization vector recovered:
The sun is high.
Checking the iv correctness...
The iv is in a correct format.

Choose the operation you want to perform ['0: ENCODE', '1: DECODE'] -> 1
You have chosen the option DECODE.

Recovering the ciphertext from the file ciphertext.txt...
Ciphertext recovered:
6111a5494ec18b180d71871e15ad9ae5db7c54bea86d4f871da6b40dfd81fe54c2e7c4c194959573212174efac4451097791833ee7599428216df96492206a16
Checking the ciphertext correctness...
The ciphertext is in a correct format.

Start decryption...

Plaintext:
The night sky is filled with endless possibilities.

Saving the plaintext to file plaintext.txt...
Plaintext saved successfully.
```

Key: The ocean is dee
Initialization vector: Find your true p
Plain text: Kindness is the language the world understands. Kindness is the language the world understands.
Ciphertext:
6111a5494ec18b180d71871e15ad9ae5db7c54bea86d4f871da6b40dfd81fe54c2e7c4c1949595
73212174efac4451097791833ee7599428216df96492206a16

Encryption

```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 0
You have chosen the option AES128.
Number of rounds: 10 + 1
Size of the key: 16

Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 1
You have chosen the option CBC.

Recovering the key from the file key.txt...
Key recovered:
The ocean is dee
Checking the key correctness...
The key is in a correct format.

Recovering the initialization vector from the file initialization_vector.txt...
Initialization vector recovered:
Find your true p
Checking the iv correctness...
The iv is in a correct format.

Choose the operation you want to perform ['0: ENCODE', '1: DECODE'] -> 0
You have chosen the option ENCODE.

Recovering the plaintext from the file plaintext.txt...
Plaintext recovered:
Kindness is the language the world understands. Kindness is the language the world understands.


Start encryption...

Ciphertext:
1d91ebb55199595f245c98334a882ae00d7c003b1473c84cdfcad11fbf1334b072326cbb2c22f50a2c15eecc5b6383ea7fdf023e63dade3b10d72ac257fde56bd60234fe2afe6a20e6cdbf679ce284f4e2f71df34bb186a9cb5eee8a18f51b7c3dd3c9a4c2f46cdc32471fe19784a662

Saving the ciphertext to file ciphertext.txt...
Ciphertext saved successfully.
```

Decryption

```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 0
You have chosen the option AES128.
Number of rounds: 10 + 1
Size of the key: 16

Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 1
You have chosen the option CBC.

Recovering the key from the file key.txt...
Key recovered:
The ocean is dee
Checking the key correctness...
The key is in a correct format.

Recovering the initialization vector from the file initialization_vector.txt...
Initialization vector recovered:
Find your true p
Checking the iv correctness...
The iv is in a correct format.

Choose the operation you want to perform ['0: ENCODE', '1: DECODE'] -> 1
You have chosen the option DECODE.

Recovering the ciphertext from the file ciphertext.txt...
Ciphertext recovered:
1d91ebb55199595f245c98334a882ae00d7c003b1473c84cdfcad11fbf1334b072326cbb2c22f50a2c15eecc5b6383ea7fdf023e63dade3b10d72ac257fde56bd60234fe2afe6a20e6cdbf679ce284f4e2f71df34bb186a9cb5eee8a18f51b7c3dd3c9a4c2f46cdc32471fe19784a662
Checking the ciphertext correctness...
The ciphertext is in a correct format.

Start decryption...

Plaintext:
Kindness is the language the world understands. Kindness is the language the world understands.


Saving the plaintext to file plaintext.txt...
Plaintext saved successfully.
```

- Test in ECB modes for AES-192

Key: From the earliest dayss
Plain text: In the vast expanse of human history, the quest for knowledge and understanding has been one of the driving forces that has shaped the course of civilizations.
Ciphertext:
dc6a2cb8687a82b2cc9c6e3bde534a4d8bbfb320f25b95df1f279c314fe5ce4a8735610558c3069
37ceb3ad285aa3c7e1184fac03e2b1151bd44184d850803d45720a919ad3fa80a509d460657d6f
cb27ed82f4d4f7ddb1b76f72b57a29efd819a04ec17a671f52068b9631e12e43ca4113034de5dee
5824eb0277c68210ed71050d3e557b3fb3d10057849498cbafc569966681fedd9ef42f7ee4e96f9f
a4d6148910f6f5720f79b43ce75825e5cc84

Encryption

```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 1
You have chosen the option AES192.
Number of rounds: 12 + 1
Size of the key: 24

Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 0
You have chosen the option ECB.

Recovering the key from the file key.txt...
Key recovered:
From the earliest dayss

Checking the key correctness...
The key is in a correct format.

Choose the operation you want to perform ['0: ENCODE', '1: DECODE'] -> 0
You have chosen the option ENCODE.

Recovering the plaintext from the file plaintext.txt...
Plaintext recovered:
In the vast expanse of human history, the quest for knowledge and understanding has been one of the driving forces that has shaped the course of civilizations.


Start encryption...

Ciphertext:
dc6a2cb8687a82b2cc9c6e3bde534a4d8bbfb320f25b95df1f279c314fe5ce4a8735610558c306937ceb3ad285aa3c7e1184fac03e2b1151bd44184d850803d45720a919ad3fa80a509d460657d6fcb27ed82f4d4f7ddb1b76f72b57a29efd819a04ec17a671f52068b9631e12e43ca4113034de5dee5824eb027
7c68210ed71050d3e557b3fb3d10057849498cbafc569966681fedd9ef42f7ee4e96f9fa4d614891 0f6f5720f79b43ce75825e5cc84

Saving the ciphertext to file ciphertext.txt...
Ciphertext saved successfully.
```

Decryption

```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 1
You have chosen the option AES192.
Number of rounds: 12 + 1
Size of the key: 24

Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 0
You have chosen the option ECB.

Recovering the key from the file key.txt...
Key recovered:
From the earliest dayss

Checking the key correctness...
The key is in a correct format.

Choose the operation you want to perform ['0: ENCODE', '1: DECODE'] -> 1
You have chosen the option DECODE.

Recovering the ciphertext from the file ciphertext.txt...
Ciphertext recovered:
dc6a2cb8687a82b2cc9c6e3bde534a4d8bbfb320f25b95df1f279c314fe5ce4a8735610558c306937ceb3ad285aa3c7e1184fac03e2b1151bd44184d850803d45720a919ad3fa80a509d460657d6fcb27ed82f4d4f7ddb1b76f72b57a29efd819a04ec17a671f52068b9631e12e43ca4113034de5dee5824eb027
7c68210ed71050d3e557b3fb3d10057849498cbafc569966681fedd9ef42f7ee4e96f9fa4d614891 0f6f5720f79b43ce75825e5cc84
Checking the ciphertext correctness...
The ciphertext is in a correct format.

Start decryption...

Plaintext:
In the vast expanse of human history, the quest for knowledge and understanding has been one of the driving forces that has shaped the course of civilizations.


Saving the plaintext to file plaintext.txt...
Plaintext saved successfully.
```

Key: to the present day, I am
Plain text: From the earliest days of recorded time, when early humans began to grasp the concept of written language
Ciphertext:
bf1143777b053dfe86ccd80f0d5750e39b73a0844837d677aef2a3548b874341cbde1943e23978
4a83c2f8f5db6847480cf0f2caeb14c30624cee87dd06b1f09b69fb82cf0a9e21a9bf11aa07f13a0e
9496e3b668ba3b7921833c2ca38784ba7f40cecb157fe7a1bc55ef26d1a3c5d54

Encryption

```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 1
You have chosen the option AES192.
Number of rounds: 12 + 1
Size of the key: 24

Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 0
You have chosen the option ECB.

Recovering the key from the file key.txt...
Key recovered:
to the present day, I am
Checking the key correctness...
The key is in a correct format.

Choose the operation you want to perform ['0: ENCODE', '1: DECODE'] -> 0
You have chosen the option ENCODE.

Recovering the plaintext from the file plaintext.txt...
Plaintext recovered:
From the earliest days of recorded time, when early humans began to grasp the concept of written language

Start encryption...

Ciphertext:
bf1143777b053dfe86ccd80f0d5750e39b73a0844837d677aef2a3548b874341cbde1943e239784a83c2f8f5db6847480cf0f2caeb14c30624cee87dd06b1f09b69fb82cf0a9e21a9bf11aa07f13a0e9496e3b668ba3b7921833c2ca38784ba7f40cecb157fe7a1bc55ef26d1a3c5d54

Saving the ciphertext to file ciphertext.txt...
Ciphertext saved successfully.
```

Decryption

```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 1
You have chosen the option AES192.
Number of rounds: 12 + 1
Size of the key: 24

Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 0
You have chosen the option ECB.

Recovering the key from the file key.txt...
Key recovered:
to the present day, I am
Checking the key correctness...
The key is in a correct format.

Choose the operation you want to perform ['0: ENCODE', '1: DECODE'] -> 1
You have chosen the option DECODE.

Recovering the ciphertext from the file ciphertext.txt...
Ciphertext recovered:
bf1143777b053dfe86ccd80f0d5750e39b73a0844837d677aef2a3548b874341cbde1943e239784a83c2f8f5db6847480cf0f2caeb14c30624cee87dd06b1f09b69fb82cf0a9e21a9bf11aa07f13a0e9496e3b668ba3b7921833c2ca38784ba7f40cecb157fe7a1bc55ef26d1a3c5d54
Checking the ciphertext correctness...
The ciphertext is in a correct format.

Start decryption...

Plaintext:
From the earliest days of recorded time, when early humans began to grasp the concept of written language

Saving the plaintext to file plaintext.txt...
Plaintext saved successfully.
```

- Test in CBC modes for AES-192

Key: Hold fast to hop
Initialization vector: At the heart of this que
Plain text: where we can communicate instantaneously across the globe through advanced digital technologies, the pursuit of knowledge has been a fundamental part of human nature.
Ciphertext:
ee69238e3863580f33e26396b41ed0d4a94fb4534f9a59dad67014f0c9fa950a8f8c91e6dda39eb
a625f331b40b2aca0a9abe084c30d1ceaf8fa648368bff9f9d7ce22adfd2aa5d8ed1eaf5e537daf19
bc71995c4561bb16510ecfd834d6ef9e2bfbd631edac2c84e2df5e160ac764c7b654f96b44e2325
e30eddc82918eea705708e457058bf678587717a7fe69f5c480edac1dce5a92aa080c7b313bd9b
766676472b0d904430831fd52783a5cd593

Encryption

```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 1
You have chosen the option AES192.
Number of rounds: 12 + 1
Size of the key: 24

Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 1
You have chosen the option CBC.

Recovering the key from the file key.txt...
Key recovered:
At the heart of this que
Checking the key correctness...
The key is in a correct format.

Recovering the initialization vector from the file initialization_vector.txt...
Initialization vector recovered:
lies curiosityyy
Checking the iv correctness...
The iv is in a correct format.

Choose the operation you want to perform ['0: ENCODE', '1: DECODE'] -> 0
You have chosen the option ENCODE.

Recovering the plaintext from the file plaintext.txt...
Plaintext recovered:
where we can communicate instantaneously across the globe through advanced digital technologies, the pursuit of knowledge has been a fundamental part of human nature.

Start encryption...

Ciphertext:
ee69238e3863580f33e26396b41ed0d4a94fb4534f9a59dad67014f0c9fa950a8f8c91e6dda39eba625f331b40b2aca0a9abe084c30d1ceaf8fa648368bff9f9d7ce22adfd2aa5d8ed1eaf5e537daf19bc71995c4561bb16510ecfd834d6ef9e2bfbd631edac2c84e2df5e160ac764c7b654f96b44e2325e30edd
c82918eea705708e457058bf678587717a7fe69f5c480edac1dce5a92aa080c7b313bd9b766676472b0d904430831fd52783a5cd593

Saving the ciphertext to file ciphertext.txt...
Ciphertext saved successfully.
```

Decryption

```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 1
You have chosen the option AES192.
Number of rounds: 12 + 1
Size of the key: 24

Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 1
You have chosen the option CBC.

Recovering the key from the file key.txt...
Key recovered:
At the heart of this que
Checking the key correctness...
The key is in a correct format.

Recovering the initialization vector from the file initialization_vector.txt...
Initialization vector recovered:
lies curiosityyy
Checking the iv correctness...
The iv is in a correct format.

Choose the operation you want to perform ['0: ENCODE', '1: DECODE'] -> 1
You have chosen the option DECODE.

Recovering the ciphertext from the file ciphertext.txt...
Ciphertext recovered:
ee69238e3863580f33e26396b41ed0d4a94fb4534f9a59dad67014f0c9fa950a8f8c91e6dda39eba625f331b40b2aca0a9abe084c30d1ceaf8fa648368bff9f9d7ce22adfd2aa5d8ed1eaf5e537daf19bc71995c4561bb16510ecfd834d6ef9e2bfbd631edac2c84e2df5e160ac764c7b654f96b44e2325e30edd
c82918eea705708e457058bf678587717a7fe69f5c480edac1dce5a92aa080c7b313bd9b766676472b0d904430831fd52783a5cd593
Checking the ciphertext correctness...
The ciphertext is in a correct format.

Start decryption...

Plaintext:
where we can communicate instantaneously across the globe through advanced digital technologies, the pursuit of knowledge has been a fundamental part of human nature.

Saving the plaintext to file plaintext.txt...
Plaintext saved successfully.
```

Key: This curiosity has led t
Initialization vector: development of s
Plain text: innate drive to explore the world around us, to question the unknown, and to seek out answers to the mysteries of life.
Ciphertext:
97b9426acf842ccb8950cdae4970a9e2ddd7bb070f21948cbe9ccb986871a72ddd0ed4f478c5ffd0ceeba92ba54c27534385b81af8dd4bebabd970a7c687ff24b4622cb78eb4cf0f6c35bfcd24d808d11aea9c39447e6f3dd157fb67562558b8d903343cfa102fffa03d92b57499a1956fda7fe954823a528d5bc2d7102dada6

Encryption

```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 1
You have chosen the option AES192.
Number of rounds: 12 + 1
Size of the key: 24

Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 1
You have chosen the option CBC.

Recovering the key from the file key.txt...
Key recovered:
This curiosity has led t
Checking the key correctness...
The key is in a correct format.

Recovering the initialization vector from the file initialization_vector.txt...
Initialization vector recovered:
development of s
Checking the iv correctness...
The iv is in a correct format.

Choose the operation you want to perform ['0: ENCODE', '1: DECODE'] -> 0
You have chosen the option ENCODE.

Recovering the plaintext from the file plaintext.txt...
Plaintext recovered:
innate drive to explore the world around us, to question the unknown, and to seek out answers to the mysteries of life.

Start encryption...

Ciphertext:
97b9426acf842ccb8950cdae4970a9e2ddd7bb070f21948cbe9ccb986871a72ddd0ed4f478c5ffd0ceeba92ba54c27534385b81af8dd4bebabd970a7c687ff24b4622cb78eb4cf0f6c35bfcd24d808d11aea9c39447e6f3dd157fb67562558b8d903343cfa102fffa03d92b57499a1956fda7fe954823a528d5bc
2d7102dada6

Saving the ciphertext to file ciphertext.txt...
Ciphertext saved successfully.
```

Decryption

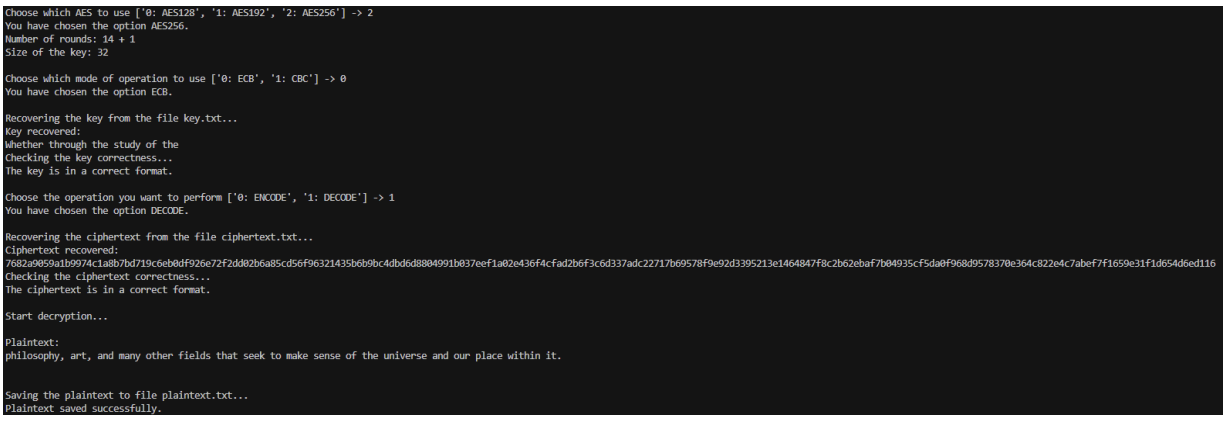
```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 1
You have chosen the option AES192.
Number of rounds: 12 + 1
Size of the key: 24

Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 1
You have chosen the option CBC.

Recovering the key from the file key.txt...
Key recovered:
This curiosity has led t
Checking the key correctness...
The key is in a correct format.

Recovering the initialization vector from the file initialization_vector.txt...
Initialization vector recovered:
development of s
Checking the iv correctness...
The iv is in a correct format.

Choose the operation you want to perform ['0: ENCODE', '1: DECODE'] -> 1
You have chosen the option DECODE.

Recovering the ciphertext from the file ciphertext.txt...
Ciphertext recovered:
97b9426acf842ccb8950cdae4970a9e2ddd7bb070f21948cbe9ccb986871a72ddd0ed4f478c5ffd0ceeba92ba54c27534385b81af8dd4bebabd970a7c687ff24b4622cb78eb4cf0f6c35bfcd24d808d11aea9c39447e6f3dd157fb67562558b8d903343cfa102fffa03d92b57499a1956fda7fe954823a528d5bc
2d7102dada6
Checking the ciphertext correctness...
The ciphertext is in a correct format.

Start decryption...

Plaintext:
innate drive to explore the world around us, to question the unknown, and to seek out answers to the mysteries of life.

Saving the plaintext to file plaintext.txt...
Plaintext saved successfully.
```

- Test in ECB modes for AES-256

Key: Whether through the study of the
Plain text: philosophy, art, and many other fields that seek to make sense of the universe and our place within it.
Ciphertext:
7682a9059a1b9974c1a8b7bd719c6eb0df926e72f2dd02b6a85cd56f96321435b6b9bc4dbd6d8804991b037eef1a02e436f4cfad2b6f3c6d337adc22717b69578f9e92d3395213e1464847f8c2b62ebaf7b04935cf5da0f968d9578370e364c822e4c7abef7f1659e31f1d654d6ed116

Encryption

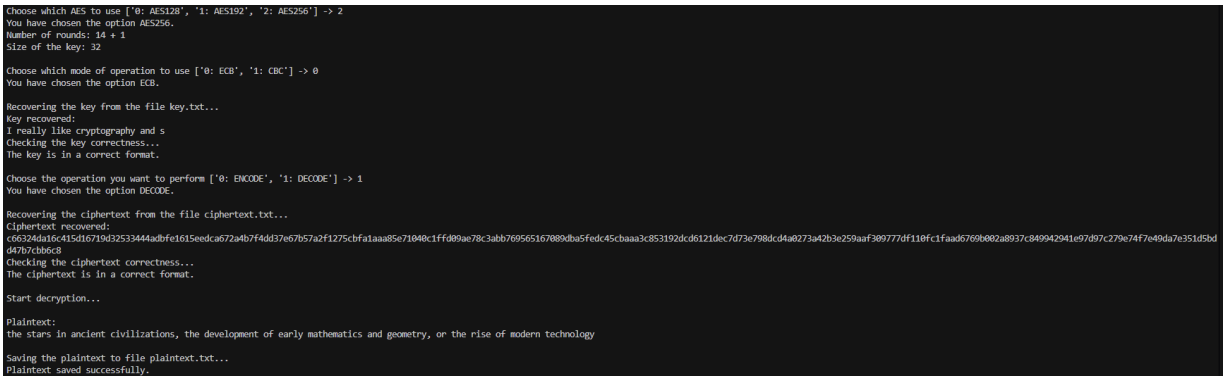
Decryption



Key: I really like cryptography and s
Plain text: the stars in ancient civilizations, the development of early mathematics and geometry, or the rise of modern technology
Ciphertext:
c66324da16c415d16719d32533444adbfe1615eedca672a4b7f4dd37e67b57a2f1275cbfa1aaa85e71040c1ffd09ae78c3abb769565167089dba5fedc45cbaaa3c853192dcd6121dec7d73e798dcd4a0273a42b3e259aaf309777df110fc1faad6769b002a8937c849942941e97d97c279e74f7e49da7e351d5bdd47b7cbb6c8

Encryption

Decryption



- Test in CBC modes for AES-256

Key: The invention of the printing pr
Initialization vector: century revoluti
Plain text: One of the most profound aspects of this intellectual journey has been the way in which knowledge has been shared and transmitted across generations.
Ciphertext:
98a106a27e3a30802c5a05ac57ba6878af166f58d1acd1ef3ef460c5c3a4db8a12c4087c24e774456c8ec565ba8fe0dd46bed1fb2d3ab3cadba725ec2062da39397e507aaf77313d3327dd11bebe1b34bfeab3c044f1361880814a5f8fe6dc5599a44017613d9b14bc5d8a7534e8676a456da6442c674c8fc3d8f3e2a2665ffc79216f34d0e36220c3fd7d3dfb165933ffb1d8f50ebab99442cc5a9261fdc61b

Encryption

```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 2
You have chosen the option AES256.
Number of rounds: 14 + 1
Size of the key: 32

Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 1
You have chosen the option CBC.

Recovering the key from the file key.txt...
Key recovered:
The invention of the printing pr
Checking the key correctness...
The key is in a correct format.

Recovering the initialization vector from the file initialization_vector.txt...
Initialization vector recovered:
century revoluti
Checking the iv correctness...
The iv is in a correct format.

Choose the operation you want to perform ['0: ENCODE', '1: DECODE'] -> 0
You have chosen the option ENCODE.

Recovering the plaintext from the file plaintext.txt...
Plaintext recovered:
One of the most profound aspects of this intellectual journey has been the way in which knowledge has been shared and transmitted across generations.

Start encryption...

Ciphertext:
98a106a27e3a30802c5a05ac57ba6878af166f58d1acd1ef3ef460c5c3a4db8a12c4087c24e774456c8ec565ba8fe0dd46bed1fb2d3ab3cadba725ec2062da39397e507aaf77313d3327dd11bebe1b34bfeab3c044f1361880814a5f8fe6dc5599a44017613d9b14bc5d8a7534e8676a456da6442c674c8fc3d8f
3e2a2665ffc79216f34d0e36220c3fd7d3dfb165933ffb1d8f50ebab99442cc5a9261fdc61b

Saving the ciphertext to file ciphertext.txt...
Ciphertext saved successfully.
```

Decryption

```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 2
You have chosen the option AES256.
Number of rounds: 14 + 1
Size of the key: 32

Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 1
You have chosen the option CBC.

Recovering the key from the file key.txt...
Key recovered:
The invention of the printing pr
Checking the key correctness...
The key is in a correct format.

Recovering the initialization vector from the file initialization_vector.txt...
Initialization vector recovered:
century revoluti
Checking the iv correctness...
The iv is in a correct format.

Choose the operation you want to perform ['0: ENCODE', '1: DECODE'] -> 1
You have chosen the option DECODE.

Recovering the ciphertext from the file ciphertext.txt...
Ciphertext recovered:
98a106a27e3a30802c5a05ac57ba6878af166f58d1acd1ef3ef460c5c3a4db8a12c4087c24e774456c8ec565ba8fe0dd46bed1fb2d3ab3cadba725ec2062da39397e507aaf77313d3327dd11bebe1b34bfeab3c044f1361880814a5f8fe6dc5599a44017613d9b14bc5d8a7534e8676a456da6442c674c8fc3d8f
3e2a2665ffc79216f34d0e36220c3fd7d3dfb165933ffb1d8f50ebab99442cc5a9261fdc61b
Checking the ciphertext correctness...
The ciphertext is in a correct format.

Start decryption...

Plaintext:
One of the most profound aspects of this intellectual journey has been the way in which knowledge has been shared and transmitted across generations.

Saving the plaintext to file plaintext.txt...
Plaintext saved successfully.
```

Key: a period of unprecedented intell
Initialization vector: and cultural gro
Plain text: the spread of ideas, allowing for the mass production of books and other written works. This in turn led to the Renaissance
Ciphertext:
8cf1e236332a906c84ae52f33c74ad966a39c60fd3cfbde49444fad881342672bd05da9ceeefc3f4
8ecb46c69f233a8d60bebd707f5a76f272dbb255ae49ca72e68484b22a9c0ba30dbc529186ec92
aed32e2fd679b86f56247021bd4bcc745defdc4f4eb5f7c3fcbc64defe78ca8937d98030211252f8c
d655ef65e4896e4a0

Encryption

```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 2
You have chosen the option AES256.
Number of rounds: 14 + 1
Size of the key: 32

Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 1
You have chosen the option CBC.

Recovering the key from the file key.txt...
Key recovered:
a period of unprecedented intell
Checking the key correctness...
The key is in a correct format.

Recovering the initialization vector from the file initialization_vector.txt...
Initialization vector recovered:
and cultural gro
Checking the iv correctness...
The iv is in a correct format.

Choose the operation you want to perform ['0: ENCODE', '1: DECODE'] -> 0
You have chosen the option ENCODE.

Recovering the plaintext from the file plaintext.txt...
Plaintext recovered:
the spread of ideas, allowing for the mass production of books and other written works. This in turn led to the Renaissance

Start encryption...

Ciphertext:
8cf1e236332a906c84ae52f33c74ad966a39c60fd3cfbde49444fad881342672bd05da9ceeefc3f48ecb46c69f233a8d60bebd707f5a76f272dbb255ae49ca72e68484b22a9c0ba30dbc529186ec92aed32e2fd679b86f56247021bd4bcc745defdc4f4eb5f7c3fcbc64defe78ca8937d98030211252f8cd655ef
65e4896e4a0

Saving the ciphertext to file ciphertext.txt...
Ciphertext saved successfully.
```

Decryption

```
Choose which AES to use ['0: AES128', '1: AES192', '2: AES256'] -> 2
You have chosen the option AES256.
Number of rounds: 14 + 1
Size of the key: 32

Choose which mode of operation to use ['0: ECB', '1: CBC'] -> 1
You have chosen the option CBC.

Recovering the key from the file key.txt...
Key recovered:
a period of unprecedented intell
Checking the key correctness...
The key is in a correct format.

Recovering the initialization vector from the file initialization_vector.txt...
Initialization vector recovered:
and cultural gro
Checking the iv correctness...
The iv is in a correct format.

Choose the operation you want to perform ['0: ENCODE', '1: DECODE'] -> 1
You have chosen the option DECODE.

Recovering the ciphertext from the file ciphertext.txt...
Ciphertext recovered:
8cf1e236332a906c84ae52f33c74ad966a39c60fd3cfbde49444fad881342672bd05da9ceeefc3f48ecb46c69f233a8d60bebd707f5a76f272dbb255ae49ca72e68484b22a9c0ba30dbc529186ec92aed32e2fd679b86f56247021bd4bcc745defdc4f4eb5f7c3fcbc64defe78ca8937d98030211252f8cd655ef
65e4896e4a0
Checking the ciphertext correctness...
The ciphertext is in a correct format.

Start decryption...

Plaintext:
the spread of ideas, allowing for the mass production of books and other written works. This in turn led to the Renaissance

Saving the plaintext to file plaintext.txt...
Plaintext saved successfully.
```

Link to the github project: https://github.com/AY02/AES-128-192-256-Implementation