CSE350: Programming Assignment 2

Ananya Lohani (2019018), Mihir Chaturvedi (2019061)

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Advanced Encryption Standard

- AES, or Advanced Encryption Standard, is a symmetric encryption algorithm that uses a block cipher to encrypt data in fixed-size blocks of 128 bits.
- The 128-bit key size is one of the three possible key sizes in AES, and it determines the number of rounds used in the encryption process.
- For 128-bit AES, the encryption process uses 10 rounds of operations, each consisting of four steps - SubBytes, ShiftRows, MixColumns, and AddRoundKey.

AES Specifications for our Implementation

- **Number of rounds:** 10
- **Key size:** 128 bits or 16 bytes
- **Plaintext size:** 128 bits or 16 bytes
- **Mode:** CBC (Cipher Block Chaining)

Documentation: Attributes in the AES class

- **Constants**
  - `KEY_SIZE`: the size of the key in bytes (16)
  - `N_ROUNDS`: the number of rounds (10)
  - `SBOX`: the S-box used for SubBytes in encryption
  - `INV_SBOX`: the inverse S-box used for SubBytes in decryption
  - `RC`: the round constants used for round key generation
- **Variables**
  - `master_key`: the master key used for encryption and decryption
  - `round_keys`: the round keys generated from the master key
  - `gf`: the Galois field (2**8) used for multiplication in MixColumns

Documentation: Methods in the AES class

- **Helper functions**
  - `left_rotate(word) -> bytes`: Rotates a word (a 4-byte sequence) to the left by one byte.
  - `bytes_to_blocks(data) -> list`: Converts a byte string into a list of n-byte blocks.
  - `blocks_to_bytes(blocks) -> bytes`: Converts a list of blocks into a byte string.
  - `xor(a, b) -> bytes`: Performs the XOR operation between two byte strings of the same length.
  - `pad_msg(msg) -> bytes`: Pads a message to a length that is a multiple of 128 bits.
  - `unpad_msg(msg) -> bytes`: Removes the padding from a message.
- **Key Generation**
  - `get_round_keys(key) -> list`: Generates a list of 11 round keys from the master key.

Documentation: Methods in the AES class

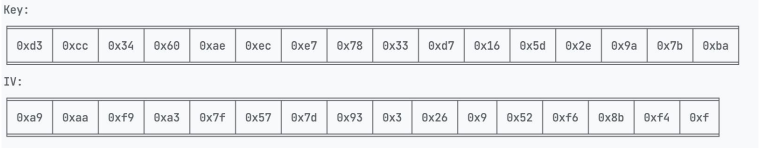
- **Encryption/decryption:**
    - `add_round_key(state, round_key) -> list` : Performs the AddRoundKey step of the encryption & decryption process.
    - `sub_bytes(state, inv=False) -> list` : Performs the SubBytes step of the encryption & decryption process.
    - `shift_rows(state, inv=False) -> list` : Performs the ShiftRows step of the encryption & decryption process.
    - `mix_columns(state, inv=False) -> list` : Performs the MixColumns step of the encryption & decryption process.
    - `encrypt(state, iv) -> bytes` : Encrypts a message using the AES algorithm in CBC mode.
    - `decrypt(state, iv) -> bytes` : Decrypts an encrypted message using the AES algorithm in CBC mode.

Sample Inputs & Outputs

Plaintext: Hello, world!

<u>I/O Pair #1</u>

Inputs:



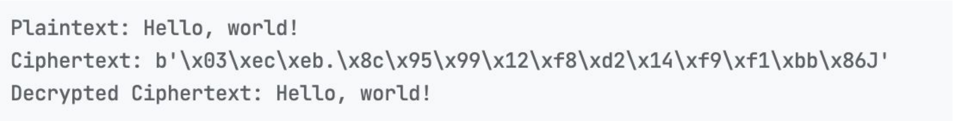
Key:

0xd3	0xcc	0x34	0x60	0xae	0xec	0xe7	0x78	0x33	0xd7	0x16	0x5d	0x2e	0x9a	0x7b	0xba

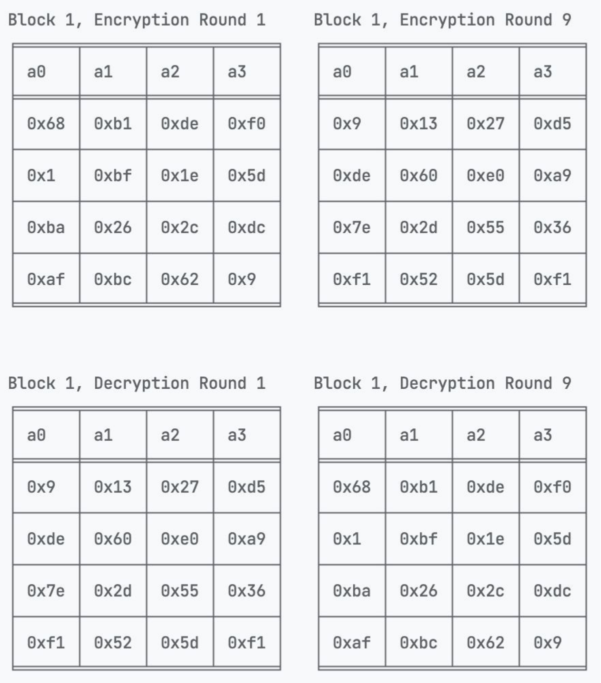
IV:

0xa9	0xaa	0xf9	0xa3	0x7f	0x57	0x7d	0x93	0x3	0x26	0x9	0x52	0xf6	0x8b	0xf4	0xf

Outputs:



```
Plaintext: Hello, world!
Ciphertext: b'\x03\xec\xeb.\x8c\x95\x99\x12\xf8\xd2\x14\xf9\xf1\xbb\x86J'
Decrypted Ciphertext: Hello, world!
```



Block 1, Encryption Round 1

a0	a1	a2	a3
0x68	0xb1	0xde	0xf0
0x1	0xbf	0x1e	0x5d
0xba	0x26	0x2c	0xdc
0xaf	0xbc	0x62	0x9

Block 1, Encryption Round 9

a0	a1	a2	a3
0x9	0x13	0x27	0xd5
0xde	0x60	0xe0	0xa9
0x7e	0x2d	0x55	0x36
0xf1	0x52	0x5d	0xf1

Block 1, Decryption Round 1

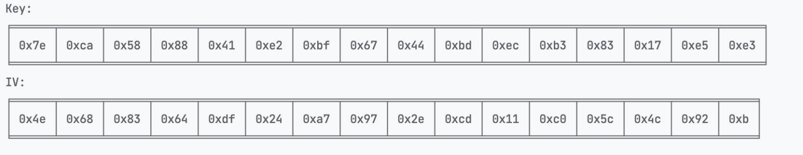
a0	a1	a2	a3
0x9	0x13	0x27	0xd5
0xde	0x60	0xe0	0xa9
0x7e	0x2d	0x55	0x36
0xf1	0x52	0x5d	0xf1

Block 1, Decryption Round 9

a0	a1	a2	a3
0x68	0xb1	0xde	0xf0
0x1	0xbf	0x1e	0x5d
0xba	0x26	0x2c	0xdc
0xaf	0xbc	0x62	0x9

I/O Pair #2

Inputs:



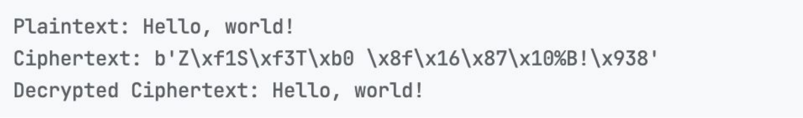
Key:

0x7e	0xca	0x58	0x88	0x41	0xe2	0xbf	0x67	0x44	0xbd	0xec	0xb3	0x83	0x17	0xe5	0xe3

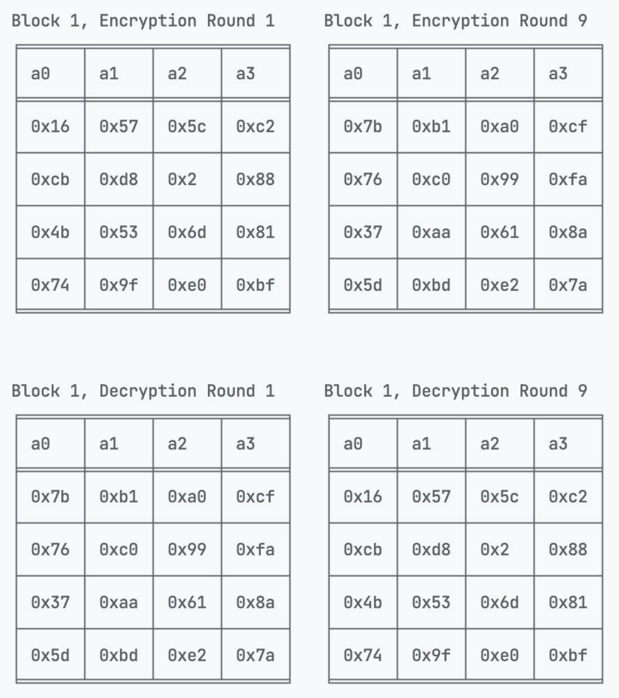
IV:

0x4e	0x68	0x83	0x64	0xdf	0x24	0xa7	0x97	0x2e	0xcd	0x11	0xc0	0x5c	0x4c	0x92	0xb

Outputs:



```
Plaintext: Hello, world!
Ciphertext: b'Z\xf1S\xf3T\xb0 \x8f\x16\x87\x10%B!\x938'
Decrypted Ciphertext: Hello, world!
```



Block 1, Encryption Round 1

a0	a1	a2	a3
0x16	0x57	0x5c	0xc2
0xcb	0xd8	0x2	0x88
0x4b	0x53	0x6d	0x81
0x74	0x9f	0xe0	0xbf

Block 1, Encryption Round 9

a0	a1	a2	a3
0x7b	0xb1	0xa0	0xcf
0x76	0xc0	0x99	0xfa
0x37	0xaa	0x61	0x8a
0x5d	0xbd	0xe2	0x7a

Block 1, Decryption Round 1

a0	a1	a2	a3
0x7b	0xb1	0xa0	0xcf
0x76	0xc0	0x99	0xfa
0x37	0xaa	0x61	0x8a
0x5d	0xbd	0xe2	0x7a

Block 1, Decryption Round 9

a0	a1	a2	a3
0x16	0x57	0x5c	0xc2
0xcb	0xd8	0x2	0x88
0x4b	0x53	0x6d	0x81
0x74	0x9f	0xe0	0xbf