ECE 592 Assignment - 1

Sharath Pendyala
Unity ID: 200011109
NC State University

September 9, 2024

Contents

List of Figures

List of Tables

1 Introduction

The goal is to construct Advanced Encryption Standard - 128 (AES-128) Encoder with a fixed Secret Key (0x000102030405060708090a0b0c0d0e0f) and set of expanded keys as described in AES Key Schedule [5]. The bonus objective would be to design the encoder for the same secret key.

AES is a symmetric key block cipher [3][4]. Since AES Encoder and Decoder work with input vector as a 4x4 matrix with each element being 8-bits, also known as *STATE*, my approach is to use Matlab (version R2023b) [1] to work on this assignment. This is because Matlab offers the flexibility to manipulate matrices with predefined functions and methods that would be ideal for dealing with *STATE* and *S-BOX* matrices. The limitation with Matlab is that functions such as *bitxor* (to perform bit-wise Exclusive OR (XOR) operation on two vectors of same dimension) work on numeric decimal values, so the input(s) need to be converted from hexadecimal character string(s) to decimal numbers to perform these operations. Similarly, the decimal numbers need to be converted back into hexadecimal character strings at the end to be user readable. With the bonus objective in mind, it is essential to design functions that can be easily reused for both the encryption function as well as the decryption function.

For Finite Field Multiplication, I plan to use Peasant's Algorithm [6] to perform multiplication between two 8-bit numbers in GF(2^8).

2 Answers to Questions 1 through 9

Question 1: What is the value (round[1].start) you obtain? This is the value you start your first round of AES.
Answer 1: '00102030405060708090a0b0c0d0e0f0'

Question 2: What is the value (round[1].s box) you obtain? (Hint the second byte from the left is 'ca'.)
Answer 2: '63cab7040953d051cd60e0e7ba70e18c'

Question 3: What is the matrix you obtain? Draw a 4-by-4 table and fill in all the cells.
Answer 3:

63	09	cd	ba
ca	53	60	70
b7	d0	e0	e1
04	51	e7	8c

Table 1: Answer 3

Question 4: What is the value (round[1].s row) you obtain? (Hint: the second byte from the left is '53'.)
Answer 4: '6353e08c0960e104cd70b751bacad0e7'

Question 5: What is the value (round[1].m col) you obtain? (Hint: the second byte from the left is '72'.)
Answer 5: '5f72641557f5bc92f7be3b291db9f91a'

Question 6: What is the value (round[2].start) you obtain? (Hint: It should be '89d810e8855ace682d1843d8cb128f
Answer 6: '89d810e8855ace682d1843d8cb128fe4'

Question 7: What is the value (round[10].start) you obtain? (Hint: the second byte from the left should be '6e'.)
Answer 7: 'bd6e7c3df2b5779e0b61216e8b10b689'

Question 8: What is the output ciphertext you obtain? (Hint: The second byte from the left should be 'c4'.)
Answer 8: '69c4e0d86a7b0430d8cdb78070b4c55a'

Question 9: Report how much time you spent on this assignment in hours.
Answer 9: 16 hours. Since I was not familiar with Overleaf and Latex, around 8 hours were spend on learning the basics of using Overleaf with the help of ChatGPT [2].

3 Conclusion

Matlab is an ideal platform to implement and test AES-128 Encryption and Decryption algorithms. It is possible to design AddRoundKey, SubBytes, and ShiftRows, which are AES Specific functions, and Finite Field Multiplication function such that they can be reused between the Encryption and Decryption algorithm. Function to convert decimal numerical *STATE* matrix to hexadecimal character string was also designed to be reusable between Encryption and Decryption functions.

4 Matlab Screenshots

```
>> calculate_aes_128('00112233445566778899aabbccddeeff')

ans =

    '69c4e0d86a7b0430d8cdb78070b4c55a'

>> decrypt_aes_128('69c4e0d86a7b0430d8cdb78070b4c55a')

ans =

    '00112233445566778899aabbccddeeff'

>> decrypt_aes_128(calculate_aes_128('00112233445566778899aabbccddeeff'))

ans =

    '00112233445566778899aabbccddeeff'
```

(a) Encryption and Decryption Functions on Matlab

```
>> calculate_aes_128('00112233445566778899aabbccddeeff',2)


Round = 0 (Initial Transformation, Add Key) Start
 State after Initial Transform  = 0x00102030405060708090a0b0c0d0e0f0
Round = 0 End

Round = 1 Start
 State after Sub Bytes     = 0x63cab7040953d051cd60e0e7ba70e18c
 State after Shift Rows    = 0x6353e08c0960e104cd70b751bacad0e7
 State after Mix Cols      = 0x5f72641557f5bc92f7be3b291db9f91a
 State after Add Round Key = 0x89d810e8855ace682d1843d8cb128fe4
Round = 1 End

Round = 2 Start
 State after Sub Bytes     = 0xa761ca9b97be8b45d8ad1a611fc97369
 State after Shift Rows    = 0xa7be1a6997ad739bd8c9ca451f618b61
 State after Mix Cols      = 0xff87968431d86a51645151fa773ad009
 State after Add Round Key = 0x4915598f55e5d7a0daca94fa1f0a63f7
Round = 2 End

Round = 3 Start
 State after Sub Bytes     = 0x3b59cb73fcd90ee05774222dc067fb68
 State after Shift Rows    = 0x3bd92268fc74fb735767cbe0c0590e2d
 State after Mix Cols      = 0x4c9c1e66f771f0762c3f868e534df256
 State after Add Round Key = 0xfa636a2825b339c940668a3157244d17
Round = 3 End
```

(b) Encryption Function with all Intermediate Calculations Part 1 of 3

```
Round = 4 Start
 State after Sub Bytes     = 0x2dfb02343f6d12dd09337ec75b36e3f0
 State after Shift Rows    = 0x2d6d7ef03f33e334093602dd5bfb12c7
 State after Mix Cols      = 0x6385b79ffc538df997be478e7547d691
 State after Add Round Key = 0x247240236966b3fa6ed2753288425b6c
Round = 4 End

Round = 5 Start
 State after Sub Bytes     = 0x36400926f9336d2d9fb59d23c42c3950
 State after Shift Rows    = 0x36339d50f9b539269f2c092dc4406d23
 State after Mix Cols      = 0xf4bcd45432e554d075f1d6c51dd03b3c
 State after Add Round Key = 0xc81677bc9b7ac93b25027992b0261996
Round = 5 End

Round = 6 Start
 State after Sub Bytes     = 0xe847f56514dadde23f77b64fe7f7d490
 State after Shift Rows    = 0xe8dab6901477d4653ff7f5e2e747dd4f
 State after Mix Cols      = 0x9816ee7400f87f556b2c049c8e5ad036
 State after Add Round Key = 0xc62fe109f75eedc3cc79395d84f9cf5d
Round = 6 End

Round = 7 Start
 State after Sub Bytes     = 0xb415f8016858552e4bb6124c5f998a4c
 State after Shift Rows    = 0xb458124c68b68a014b99f82e5f15554c
 State after Mix Cols      = 0xc57e1c159a9bd286f05f4be098c63439
 State after Add Round Key = 0xd1876c0f79c4300ab45594add66ff41f
Round = 7 End
```

(c) Encryption Function with all Intermediate Calculations Part 2 of 3

```
Round = 8 Start
 State after Sub Bytes     = 0x3e175076b61c04678dfc2295f6a8bfc0
 State after Shift Rows    = 0x3e1c22c0b6fcbf768da85067f6170495
 State after Mix Cols      = 0xbaa03de7a1f9b56ed5512cba5f414d23
 State after Add Round Key = 0xfde3bad205e5d0d73547964ef1fe37f1
Round = 8 End

Round = 9 Start
 State after Sub Bytes     = 0x5411f4b56bd9700e96a0902fa1bb9aa1
 State after Shift Rows    = 0x54d990a16ba09ab596bbf40ea111702f
 State after Mix Cols      = 0xe9f74eec023020f61bf2ccf2353c21c7
 State after Add Round Key = 0xbd6e7c3df2b5779e0b61216e8b10b689
Round = 9 End

Round = 10 Start
 State after Sub Bytes     = 0x7a9f102789d5f50b2beffd9f3dca4ea7
 State after Shift Rows    = 0x7ad5fda789ef4e272bca100b3d9ff59f
 State after Add Round Key = 0x69c4e0d86a7b0430d8cdb78070b4c55a
Round = 10 End

Cypher Text Upper Case (hex) = 0x69C4E0D86A7B0430D8CDB78070B4C55A
Cypher Text Lower Case (hex) = 0x69c4e0d86a7b0430d8cdb78070b4c55a


ans =

    '69c4e0d86a7b0430d8cdb78070b4c55a'
```

(d) Encryption Function with all Intermediate Calculations Part 3 of 3

Figure 1: Encryption Function with all Intermediate Calculations on Matlab

```
>> decrypt_aes_128('69c4e0d86a7b0430d8cdb78070b4c55a',2)


Round = 0 Initial Transformation (Add Key) Start
 State after Initial Transform  = 0x7ad5fda789ef4e272bca100b3d9ff59f
Round = 0 Initial Transformation End

Round = 1 Start
 State after Inverse Sub Bytes  = 0xbdb52189f261b63d0b107c9e8b6e776e
 State after Inverse Shift Rows = 0xbd6e7c3df2b5779e0b61216e8b10b689
 State after Add Round Key      = 0xe9f74eec023020f61bf2ccf2353c21c7
 State after Inverse Mix Cols   = 0x54d990a16ba09ab596bbf40ea111702f
Round = 1 End

Round = 2 Start
 State after Inverse Sub Bytes  = 0xfde596f1054737d235febad7f1e3d04e
 State after Inverse Shift Rows = 0xfde3bad205e5d0d73547964ef1fe37f1
 State after Add Round Key      = 0xbaa03de7a1f9b56ed5512cba5f414d23
 State after Inverse Mix Cols   = 0x3e1c22c0b6fcbf768da85067f6170495
Round = 2 End

Round = 3 Start
 State after Inverse Sub Bytes  = 0xd1c4941f7955f40fb46f6c0ad68730ad
 State after Inverse Shift Rows = 0xd1876c0f79c4300ab45594add66ff41f
 State after Add Round Key      = 0xc57e1c159a9bd286f05f4be098c63439
 State after Inverse Mix Cols   = 0xb458124c68b68a014b99f82e5f15554c
Round = 3 End
```

(a) Decryption Function with all Intermediate Calculations Part 1 of 3

```
Round = 4 Start
 State after Inverse Sub Bytes  = 0xc65e395df779cf09ccf9e1c3842fed5d
 State after Inverse Shift Rows = 0xc62fe109f75eedc3cc79395d84f9cf5d
 State after Add Round Key      = 0x9816ee7400f87f556b2c049c8e5ad036
 State after Inverse Mix Cols   = 0xe8dab6901477d4653ff7f5e2e747dd4f
Round = 4 End

Round = 5 Start
 State after Inverse Sub Bytes  = 0xc87a79969b0219bc2526773bb016c992
 State after Inverse Shift Rows = 0xc81677bc9b7ac93b25027992b0261996
 State after Add Round Key      = 0xf4bcd45432e554d075f1d6c51dd03b3c
 State after Inverse Mix Cols   = 0x36339d50f9b539269f2c092dc4406d23
Round = 5 End

Round = 6 Start
 State after Inverse Sub Bytes  = 0x2466756c69d25b236e4240fa8872b332
 State after Inverse Shift Rows = 0x247240236966b3fa6ed2753288425b6c
 State after Add Round Key      = 0x6385b79ffc538df997be478e7547d691
 State after Inverse Mix Cols   = 0x2d6d7ef03f33e334093602dd5bfb12c7
Round = 6 End

Round = 7 Start
 State after Inverse Sub Bytes  = 0xfab38a1725664d2840246ac957633931
 State after Inverse Shift Rows = 0xfa636a2825b339c940668a3157244d17
 State after Add Round Key      = 0x4c9c1e66f771f0762c3f868e534df256
 State after Inverse Mix Cols   = 0x3bd92268fc74fb735767cbe0c0590e2d
Round = 7 End
```

(b) Decryption Function with all Intermediate Calculations Part 2 of 3

```
Round = 8 Start
 State after Inverse Sub Bytes  = 0x49e594f755ca638fda0a59a01f15d7fa
 State after Inverse Shift Rows = 0x4915598f55e5d7a0daca94fa1f0a63f7
 State after Add Round Key      = 0xff87968431d86a51645151fa773ad009
 State after Inverse Mix Cols   = 0xa7be1a6997ad739bd8c9ca451f618b61
Round = 8 End

Round = 9 Start
 State after Inverse Sub Bytes  = 0x895a43e485188fe82d121068cbd8ced8
 State after Inverse Shift Rows = 0x89d810e8855ace682d1843d8cb128fe4
 State after Add Round Key      = 0x5f72641557f5bc92f7be3b291db9f91a
 State after Inverse Mix Cols   = 0x6353e08c0960e104cd70b751bacad0e7
Round = 9 End

Round = 10 Start
 State after Inverse Sub Bytes  = 0x0050a0f04090e03080d02070c01060b0
 State after Inverse Shift Rows = 0x00102030405060708090a0b0c0d0e0f0
 State after Add Round Key      = 0x00112233445566778899aabbccddeeff
Round = 10 End

Plain Text Upper Case (hex) = 0x00112233445566778899AABBCCDDEEFF
Plain Text Lower Case (hex) = 0x00112233445566778899aabbccddeeff


ans =

    '00112233445566778899aabbccddeeff'
```

(c) Decryption Function with all Intermediate Calculations Part 3 of 3

Figure 2: Decryption Function with all Intermediate Calculations on Matlab

5 Resources Used

The following resources were used to complete this assignment.

- Wikipedia articles [4], [8], [6], [7], [5] FIPS Publication 197 [3], and Dr. Aydin Aysu's ECE-592 (NC State) presentations were used to understand AES-128 Encryption and Decryption algorithm, get values for S-box and Inverse S-box matrices, to understand Peasant's Algorithm for Finite Field Multiplication, and to get the predefined multiplication matrices for MixColumns and InverseMixColumns steps.
- Matlab Documentation Pages for R2023b [1] was invaluable in constructing and debugging my Matlab code.
- ChatGPT [2] was used to figure out how to use Overleaf and Latex.

6 References

[1] Mathworks. *Matlab - Documentation Home - Version R2023b*. Accessed: 2024-09-02. 2023. URL: `https://www.mathworks.com/help/releases/R2023b/index.html`.

[2] OpenAI. *ChatGPT*. Accessed: 2024-09-02. 2024. URL: `https://www.openai.com/chatgpt`.

[3] United States National Institute of Standards and Technology (NIST). *ADVANCED ENCRYPTION STANDARD (AES) - Federal Information Processing Standards Publication 197*. Accessed: 2024-09-02. November 26, 2001. URL: `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf`.

[4] Wikipedia contributors. *Advanced Encryption Standard — Wikipedia, The Free Encyclopedia*. [Online; accessed 2-September-2024]. 2024. URL: `https://en.wikipedia.org/w/index.php?title=Advanced_Encryption_Standard&oldid=1242472010`.

[5] Wikipedia contributors. *AES key schedule — Wikipedia, The Free Encyclopedia*. [Online; accessed 9-September-2024]. 2024. URL: `https://en.wikipedia.org/w/index.php?title=AES_key_schedule&oldid=1204191754`.

[6] Wikipedia contributors. *Finite field arithmetic — Wikipedia, The Free Encyclopedia*. `https://en.wikipedia.org/w/index.php?title=Finite_field_arithmetic&oldid=1230887247`. [Online; accessed 2-September-2024]. 2024.

[7] Wikipedia contributors. *Rijndael MixColumns — Wikipedia, The Free Encyclopedia*. [Online; accessed 2-September-2024]. 2024. URL: `https://en.wikipedia.org/w/index.php?title=Rijndael_MixColumns&oldid=1204191266`.

[8] Wikipedia contributors. *Rijndael S-box — Wikipedia, The Free Encyclopedia*. [Online; accessed 2-September-2024]. 2024. URL: `https://en.wikipedia.org/w/index.php?title=Rijndael_S-box&oldid=1236732188`.