CSE 406
Computer Security Sessional
January 2021
Assignment 1

Topic:

In this assignment you will have to implement the Advanced Encryption Standard (AES) algorithm for 128-bit key. Total marks of this assignment is 30.

Overview of AES:

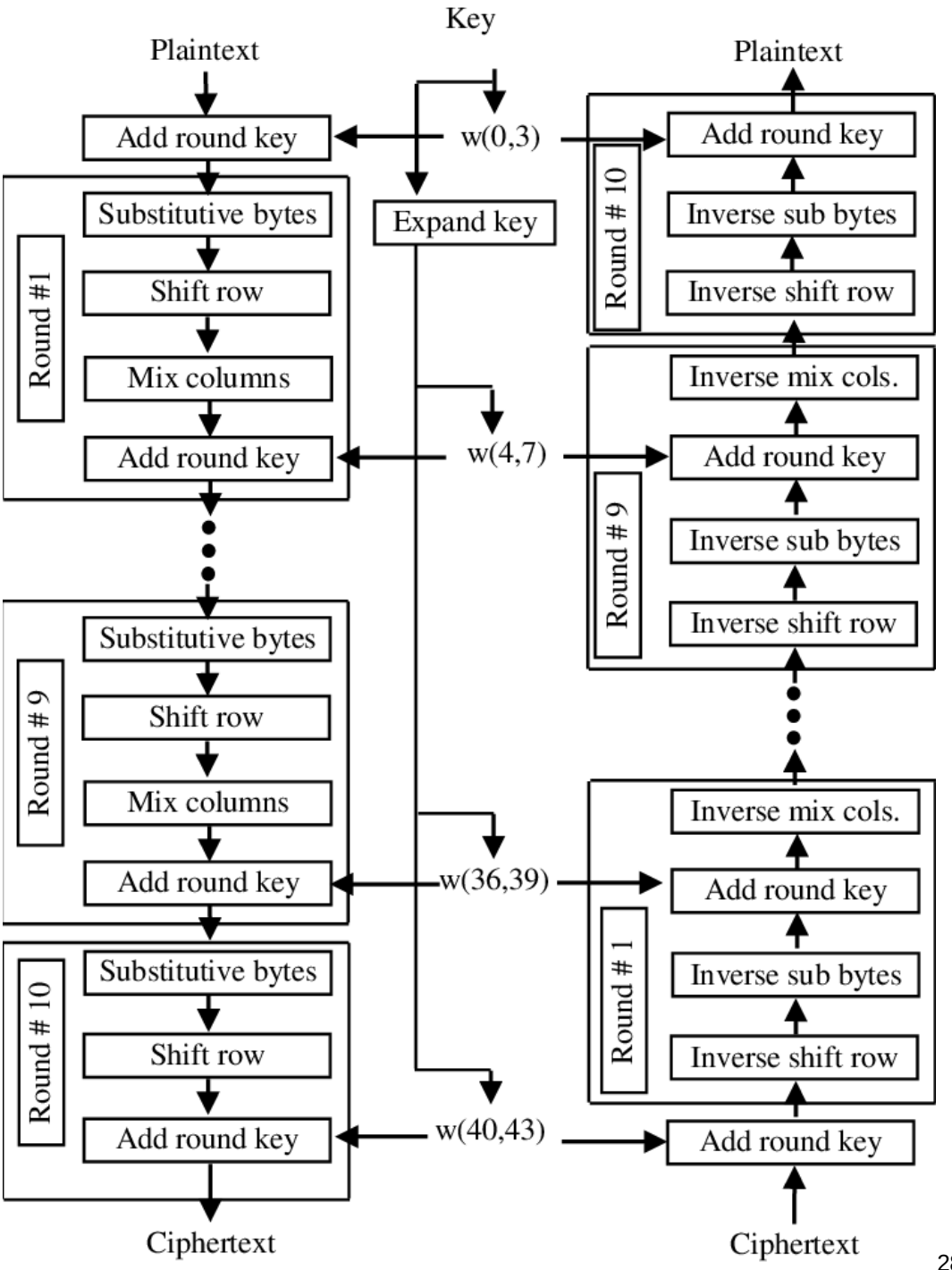
Advanced Encryption Standard (AES) is a popular and widely adopted symmetric key encryption algorithm.

AES uses repeat cycles or "rounds". There are 10, 12, or 14 rounds for keys of 128, 192, and 256 bits, respectively.

Each round of Algorithm consists of four steps:

1. **subBytes:** for each byte in the array, use its value as an index into a fixed 256-element lookup table, and replace its value in the state by the byte value stored at that location in the table. You can find the table and the inverse table on the web.
2. **shiftRows:** Let Ri denote the ith row in state. Shift R0 in the state left 0 bytes (i.e., no change); shift R1 left 1 byte; shift R2 left 2 bytes; shift R3 left 3 bytes. These are circular shifts. They do not affect the individual byte values themselves. Shift left for decryption.
3. **mixColumns:** for each column of the state, replace the column by its value multiplied by a fixed 4 x 4 matrix of integers (in a particular Galois Field). This is a relatively complex step, but if you utilize the BitVector library demonstrated in the sessional class it will be simple matrix multiplication. Note that the inverse operation multiplies by a different matrix.
4. **addRoundkey:** XOR the state with a 128-bit round key derived from the original key K by a recursive process.

The final round is slightly different from the others. Implementation details can be found in the presentation slide shared in the sessional class.



28-Feb-2021

Figure: Block Diagram of AES

Mark Distribution and Task Breakdown

Task	Marks
1	7
2	12
3	5
4	4
Proper Submission	2
Total	**30**
Bonus (Task 5)	3

Task 1: Key Scheduling (7 Marks)

The key will be provided by the user as ASCII string. If the string length is less than 16 (16x8 = 128 bit), pad it with '0' and if the length is greater than 16 ignore extra characters.

Implement the key scheduling algorithm in this step and generate keys for all the rounds.

Link of the algorithm: https://en.wikipedia.org/wiki/AES_key_schedule

Task 2: Encryption (12 Marks)

The encryption method will encrypt a block of text (128 bit/16 characters) with the keys generated in 1. You have to divide the input plain text into blocks of 16 characters and encrypt one block at a time. Pad the input string with spaces if its length is not multiple of 16.

Task 3: Decryption (5 Marks)

Decrypt the encrypted text blocks and observe if they match with the original text. Also report the execution time. A sample output is shown below.

```
Plain Text:

Key:
BUET CSE16 Batch [In ASCII]
42554554204353453136204261746368 [In HEX]

WillGraduateSoon [In ASCII]
57696c6c477261647561746536f6f6e [In HEX]

Cipher Text:
54b0f718f62f03c0a455ed78007c6386 [In HEX]
T°÷ö/————————————————————À¤Uíx|c  [In ASCII]
```

```
Deciphered Text:
57696c6c47726164756174655536f6f6e [In HEX]
WillGraduateSoon [In ASCII]

Execution Time
Key Scheduling: 0.01599264144897461 seconds
Encryption Time: 0.2241218090057373 seconds
Decryption Time: 0.3562753200531006 seconds
```

Task 4: Additional Functionalities (4 Marks)

1. Modify your program so that you can encrypt and decrypt not only text but also other objects (e.g., pdf, image etc.)
2. Generate the S-Box and the Inverse S-Box tables instead of using hardcoded values. Note that you may use the `gf_MI()` function of the BitVector module for calculating multiplicative inverse.
   Ref: https://en.wikipedia.org/wiki/Rijndael_S-box

Task 5: Bonus (3 Marks)

Generalize your program to accommodate key lengths of 128-bit, 192-bit and 256-bit. Compare and report the execution speed in console.

Task 6: [Optional]

```
Key:
446563727970742054617 36b20536978

Cipher:
182e0afe67094cb70f2a7dc74f7e0076
552456c820d6029f9519a7f8a020a6dc
6707ec0f7e1eb439f3ea0db53ee60c95
8d67693151bba8ec61dacbd83e99c6ef
9daa26069685e2284ba264a9b7ad9a56
d6203cc8ab315c34de944af524b12d65
85ccfb0c6fab4b7006266d66280ad44e
a44dbe21d269f3e030129f49851711a6
dd7b9f55dfd4c5dcee355973fc2ce648
6d7df8de352e73d434ee9932477226e4
2012d10b974dfa66366f9830b0fb62e6
9dfde63105ae1d2eccb316e4f57ceb55
eef9677d5dc267f8ece3d2fa30d2c06c
```

Important Notes

Implement as far as you can. Partial marks will be awarded.

Deadline: 11:55 PM, 13 March, 2021

Submission Guideline:

1. Create a directory and name it by your seven-digit student id <1605ABC>.
2. Rename the source file by your seven-digit student id <1605ABC.py> and put it in the directory created in 1
3. Zip the directory and name it by your seven-digit student id <1605ABC.zip>
4. Submit the zip file
5. 2 marks will be awarded for proper submission.

Plagiarism: You can easily find the implementation of AES on the Internet. Do not copy from any web source or friend. The persons involved in such activities will be penalized by -100% of the total marks.

For any query contact: toufikuzzaman@teacher.cse.buet.ac.bd