

ASSIGNMENT



Shahjalal University of Science and Technology

**Course: Introduction to Computer Security and
Forensics**

Course No: CSE-461

<u>Submitted By</u>	<u>Submitted To</u>
Name: MD Taohid Imam Khan Tamim Dept: CSE Session:2019-20 Reg no: 2019331018 Section B	Md. Shadmim Hasan Shifat Lecturer Dept. Of CSE SUST

Task:1

In this independent implementation of AES, the focus is on key expansion, encryption, and decryption processes.

Firstly, the program handles keys provided by the user, ensuring they are always of length 16. If the key is shorter than 16 characters, it pads the necessary characters to meet this length requirement, allowing for consistent key size. This ensures compatibility with the AES algorithm, which requires a fixed key length.

Next, the program encrypts blocks of text, each containing 128 bits, using the generated keys. This encryption process follows the AES algorithm, ensuring that the plaintext is securely transformed into ciphertext.

Subsequently, the program decrypts the encrypted text blocks, utilizing the same keys generated during encryption. It verifies if the decrypted text matches the original plaintext, ensuring the correctness of the decryption process.

Furthermore, the program reports time-related performance metrics, providing insights into the efficiency of key expansion, encryption, and decryption operations. These metrics help assess the speed and scalability of the AES implementation, crucial for evaluating its practical utility in real-world applications.

Overall, this implementation of AES demonstrates a robust and efficient approach to secure data encryption and decryption, facilitating the protection of sensitive information in various computing environments.

I have made a simple **web application** for this task using streamlit. So I'm attaching the screenshot of the i/o

Test Case :1

AES Encryption/Decryption

Enter Key

Key (in ASCII):

Thats my Kung Fu

Key (in HEX): 5468617473206d79204b756e67204675

Fig: 01

Enter Plain Text

Plain Text (in ASCII):

Two One Nine Two

Plain Text (in HEX): 54776f204f6e65204e696e652054776f

Fig:02

Enter Plain Text

Plain Text (in ASCII):

Two One Nine Two

Plain Text (in HEX): 54776f204f6e65204e696e652054776f

Cipher Text

Cipher Text (in ASCII): KcODUF9XFCDDtkAiwpnCsxoCw5c6

Cipher Text (in HEX): 29c3505f571420f6402299b31a02d73a

Decipher Text

Decipher Text (in ASCII): Two One Nine Two

Decipher Text (in HEX): 54776f204f6e65204e696e652054776f

Execution Time

Key Scheduling: 0.00020956993103027344 sec

Encryption Time: 0.0020575523376464844 sec

Decryption Time: 0.003829479217529297 sec

Fig:03

Test Case 2 :

AES Encryption/Decryption

Enter Key

Key (in ASCII):

SUST CSE19 Batch

Key (in HEX): 53555354204353453139204261746368

Fig:01

Enter Plain Text

Plain Text (in ASCII):

YesTheyHaveMadeItAtLast|

Plain Text (in HEX): 59657354686579486176654d616465497441744c617374

Fig:02

Plain Text (in ASCII):

YesTheyHaveMadeItAtLast

Plain Text (in HEX): 59657354686579486176654d616465497441744c617374

Cipher Text

Cipher Text (in ASCII): FVQVdxRYNgfDgQFFcQbCj0BZw5YYw7FXH8KOccKbwrLDu8OuXsK9bTrDjw==

Cipher Text (in HEX): 1554157714583607c1014571068f4059d618f1571f8e719bb2fbee5ebd6d3acf

Decipher Text

Decipher Text (in ASCII): YesTheyHaveMadeItAtLast

Decipher Text (in HEX): 59657354686579486176654d616465497441744c617374

Execution Time

Key Scheduling: 0.00012493133544921875 sec

Encryption Time: 0.004027843475341797 sec

Decryption Time: 0.0061054229736328125 sec

Fig: 03

Test Case: 03

AES Encryption/Decryption

Enter Key

Key (in ASCII):

SUST CSE19 Batch|

Key (in HEX): 53555354204353453139204261746368

Fig: 01

Enter Plain Text

Plain Text (in ASCII):

IsTheirCarnivalSuccessful

Plain Text (in HEX): 497354686569724361726e6976616c5375636365737366756c

Fig:02

Enter Plain Text

Plain Text (in ASCII):

IsTheirCarnivalSuccessful

Plain Text (in HEX): 497354686569724361726e6976616c5375636365737366756c

Cipher Text

Cipher Text (in ASCII): fQXCjgDDhMONGh7CtMOKQsOIwo13HBEfM1FIUCLCpg4KU8OjNsKCw5ACbw==

Cipher Text (in HEX): 7d058e00c4cd1a1eb4ca42c88d771c111f3351655022a60e0a53e33682d0026f

Decipher Text

Decipher Text (in ASCII): IsTheirCarnivalSuccessful

Decipher Text (in HEX): 497354686569724361726e6976616c5375636365737366756c

Execution Time

Key Scheduling: 9.179115295410156e-05 sec

Encryption Time: 0.003000020980834961 sec

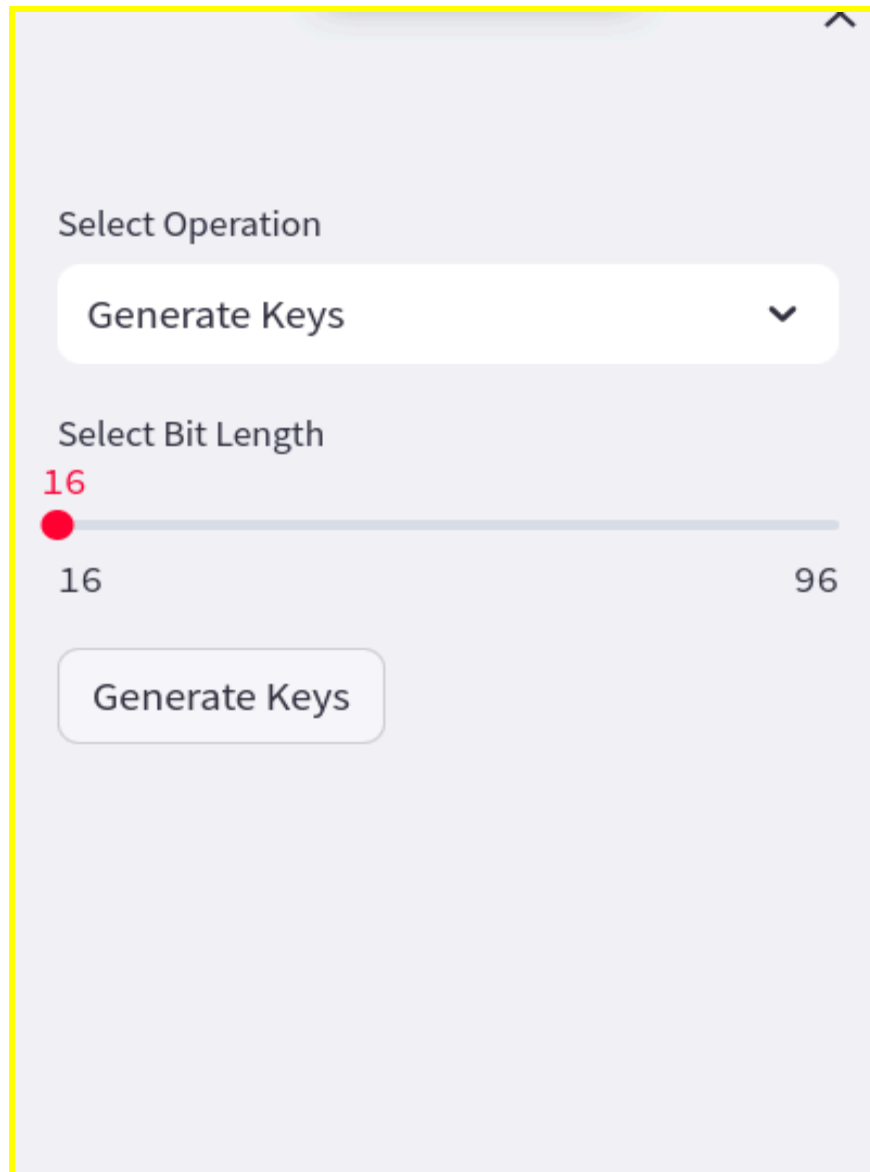
Decryption Time: 0.0028047561645507812 sec

Fig: 03

Task-02

I have made a **web application** for this task as well. I am attaching the screenshot of my web application with different bit size (16,32,64,96)

Bit size=16



The image shows a user interface for key generation. It features a 'Select Operation' dropdown menu with 'Generate Keys' selected. Below this is a 'Select Bit Length' slider. The slider has a red dot at the '16' mark, with '16' also written in red text above the dot. The slider range is from 16 to 96. At the bottom, there is a 'Generate Keys' button.

Select Operation

Generate Keys

Select Bit Length

16

16 96

Generate Keys

Fig:Key Generation

×

Deploy

Select Operation

Generate Keys

Select Bit Length

16

16

96

Generate Keys

RSA Encryption and Decryption

Public Key: $(e,n) = (7, 36863)$

Private Key: $(d,n) = (10423, 36863)$

Key Generation Time: 0.0001776218 sec

Fig: Public key and Private keys

RSA Encryption and Decryption

Enter Plain Text

BUETCSEVSSUSTCSE

Encrypt

Encrypted Text (ASCII):

0 : 4559

1 : 12051

2 : 10832

3 : 3751

4 : 11731

5 : 6058

6 : 10832

7 : 7521

8 : 6058

9 : 6058

10 : 12051

11 : 6058

12 : 3751

13 : 11731

14 : 6058

15 : 10832

Encryption Time: 0.0000257492 sec

Flg: RSA (Encryption)

RSA Encryption and Decryption

Enter Encrypted Text (ASCII)

10832,
3751,
11731,
6058,

Decrypt

Decrypted Text: BUETCSEVSSUSTCSE

Decryption Time: 0.0010459423 sec

Fig: RSA (Decryption)

Bit size=32

Select Operation

Generate Keys ▼

Select Bit Length

32

16 96

Generate Keys

Fig: Key Generation

RSA Encryption and Decryption

Public Key: (e,n) = (5, 44940481)

Private Key: (d,n) = (17960141, 44940481)

Key Generation Time: 0.0049524307 sec

Fig : Private and Public Key

RSA Encryption and Decryption

Enter Plain Text

BUETCSEVSSUSTCSE

Encrypt

Encrypted Text (ASCII):

```
▼ [  
  0 : 38939589  
  1 : 32885987  
  2 : 36054995  
  3 : 2654691  
  4 : 1910677  
  5 : 29218796  
  6 : 36054995  
  7 : 30460152  
  8 : 29218796  
  9 : 29218796  
 10 : 32885987  
 11 : 29218796  
 12 : 2654691  
 13 : 1910677  
 14 : 29218796  
 15 : 36054995  
]
```

Encryption Time: 0.0000216961 sec

Fig:RSA Encryption

×

Select Operation

Decrypt Text

RSA Encryption and Decryption

Enter Encrypted Text (ASCII)

43040002,
1910677,
29218796,
36054995
]

Decrypt

Decrypted Text: BUETCSEVSSUSTCSE

Decryption Time: 0.0012223721 sec

Deploy

Flg: RSA Decryption

Bit size=64



The screenshot shows a web application titled "RSA Encryption and Decryption". On the left, there is a sidebar with a "Select Operation" dropdown menu set to "Generate Keys". Below this is a "Select Bit Length" slider ranging from 16 to 96, with a red dot indicating the selected value of 64. A "Generate Keys" button is located below the slider. The main content area displays the generated keys: "Public Key: (e,n) = (5, 1571222425407206111)" and "Private Key: (d,n) = (628488968561252141, 1571222425407206111)". A light blue box at the bottom of the main area shows the "Key Generation Time: 0.8539381027 sec". A "Deploy" button is visible in the top right corner of the application window.

Select Operation
Generate Keys

Select Bit Length
16 64 96

Generate Keys

RSA Encryption and Decryption

Public Key: (e,n) = (5, 1571222425407206111)

Private Key: (d,n) = (628488968561252141, 1571222425407206111)

Key Generation Time: 0.8539381027 sec

Deploy

Fig:Public and Private keys

×

Select Operation

Encrypt Text

Enter Plain Text

BUETCSEVSSUSTCSE

Encrypt

Encrypted Text (ASCII):

[

0 : 1252332576

1 : 4437053125

2 : 1564031349

3 : 4182119424

4 : 1350125107

5 : 3939040643

6 : 1564031349

7 : 4704270176

8 : 3939040643

9 : 3939040643

10 : 4437053125

11 : 3939040643

12 : 4182119424

13 : 1350125107

14 : 3939040643

15 : 1564031349

]

Encryption Time: 0.000209808 sec

Fig: RSA Encryption

^

Select Operation

Decrypt Text

RSA Encryption and Decryption

Enter Encrypted Text (ASCII)

4182119424,
1350125107,
3939040643,
1564031349,

Decrypt

Decrypted Text: BUETCSEVSSUSTCSE

Decryption Time: 0.0095412731 sec

RSA Decryption

Bit size=96

×

Select Operation

Generate Keys

Select Bit Length

1696

Generate Keys

RSA Encryption and Decryption

Public Key: (e,n) = (5, 44119070718125918545664532071)

Private Key: (d,n) = (8823814143625096069752773429, 44119070718125918545664532071)

Key Generation Time: 3.1945662498 sec

⤴

Select Operation

Encrypt Text

RSA Encryption and Decryption

Enter Plain Text

BUETCSEVSSUTCSE

Encrypt

Encrypted Text (ASCII):

▼ [🔍]

0 : 1252332576

1 : 4437053125

2 : 1564031349 🔍

3 : 4182119424

4 : 1350125107

5 : 3939040643

6 : 1564031349

7 : 4704270176

8 : 3939040643

9 : 3939040643

10 : 4437053125

11 : 3939040643

12 : 4182119424

13 : 1350125107

14 : 3939040643

15 : 1564031349

]

RSA Encryption and Decryption

Enter Encrypted Text (ASCII)

```
[  
  1252332576,  
  4437053125,  
  1564031349,  
  4182119424,  
  1350125107,  
  3939040643,  
  1564031349,  
  4704270176,  
  3939040643,  
  3939040643,  
  4437053125,  
  3939040643,  
  4182119424,  
  1350125107,  
  3939040643,  
  1564031349  
]
```

Decrypt

Decrypted Text: BUETCSEVSSUSTCSE

Decryption Time: 0.0218329430 sec

Task: 03

The Hybrid Cryptosystem combines the strengths of symmetric and asymmetric encryption algorithms to ensure secure communication between parties. In this implementation, AES is utilized for symmetric encryption, while RSA is employed for asymmetric encryption.

Firstly, AES encrypts the plaintext using a randomly generated key. Subsequently, this key is encrypted using RSA, ensuring secure key exchange. The RSA private key (PRK) is stored securely in a folder named "Don't Open This" to maintain confidentiality.

When Alice, the sender, transmits the encrypted ciphertext (CT) and RSA-encrypted key (EK) to Bob, the receiver, she stores them in the secret folder. Bob then retrieves the encrypted key and decrypts it using his RSA private key, acquired from the same folder. With the decrypted key, Bob decrypts the ciphertext using AES decryption. Finally, Bob compares the decrypted plaintext (DPT) with the original plaintext to ensure message integrity.

By combining AES and RSA encryption, the Hybrid Cryptosystem provides a robust solution for secure communication, safeguarding confidentiality, integrity, and authenticity of exchanged messages.

I'm attaching Screenshot of my streamlit application:

At Alice's Side

🔗 Hybrid Cryptosystem Simulation

Select Role

Alice



Enter message (max 4096 characters):

Send Message

Alice is saying Hello to Bob:

Hybrid Cryptosystem Simulation

Select Role

Alice



Enter message (max 4096 characters):

Hello

Send Message

Message sent successfully!

At Bob's Side:

Hybrid Cryptosystem Simulation

Select Role

Bob



Enter message (max 4096 characters):

Hello

Send Message

Message sent successfully!

Fig: Bob Receives Hello From Alice