International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)
Volume 4, Issue 4, 2017, PP 1-14
ISSN 2349-4840 (Print) & ISSN 2349-4859 (Online)
DOI: http://dx.doi.org/10.20431/2349-4859.0404001
www.arcjournals.org

Implementation of DES and AES Cryptographic Algorithms in Accordance with Cloud Computing

Gurpreet Kaur[1], Dr. Gagandeep Jagdev[2]

[1]*Research Scholar (M.Tech.), Yadavindra College of Engineering, Talwandi Sabo (PB)*

[2]*Dept. of Comp. Science, Punjabi University Guru Kashi College, Damdama Sahib (PB)*

Corresponding Author:** ***Dr. Gagandeep Jagdev, *Dept. of Comp. Science, Punjabi University Guru Kashi College, Damdama Sahib (PB)*

Abstract: *As every new technology comes with certain concerns attached to it, same is the story in case of cloud computing. With the promising aspects of cloud computing in every sector, the security issues have further deepened. Cloud computing is optimized for performance, scalability, and resource consumption. All the risk factors associated with normal outsourcing are part and parcel of cloud computing. In addition to this, cloud computing has its own integral problems associated with itself.The most vital concern is the sensitivity attached with the data exposed to CSP (cloud service provider) irrespective of company size. The companies should not negotiatewith the confidentiality of the customer information.This paper discusses the working of two prominent cryptographic algorithms, DES (Data Encryption Standard) and AES (Advanced Encryption Standard) used in the cloud environment with security point of view. The paper elaborates the detailed implementation of both the algorithmsusing Matlab R2009a simulation tool.*

Keywords: *AES, cloud computing, encryption, decryption, DES.*

Abbreviations: *AES, CSP, DES*

1. INTRODUCTION

Cloud computing refers to the entire collection of software and hardware one uses which is placed at the remote location. Such service provider’s companies are referred as Cloud Service Providers (CSP) and often these services are provided by huge giant companies. In fact, the location of CSP does not matter. The Cloud computing is a transforming technology. The information and the processes are migrating to the cloud. The cloud computing handles all the challenges faced by conventional computing including handling and installation of software.

The cloud computing is the latest effort in delivering computing resources and services. It is an on-demand service and facilitates user to utilize services as and when required. The cloud computing is a pay as you use service. User pays as per his/her requirements. Cloud computing has turned to high demand service because of the several privileges enjoyed by the cloud users. The data need to be placed at the remote location and it’s the responsibility of its security is of CSP. The functioning of the cloud environment is shown in Fig. 1. The user doesn’t need to have deep knowledge about the working happening behind the scenes. For instance, when a user types any query on Google, the computer system at user’s end is not playing any significant part in finding the answers required. The typed words are shuttled over the internet on one of the Google’s hundreds or thousands of clustered PC’s which find out the appropriate results and send them back to the user. The primary aim of cloud computing is to minimize the cost and help users to focus on their business [1].

The conventional way of computing is departing the market with the arrival of the cloud computing. This new addition cannot be taken as a flash in the pan as it all set to rule the computing world in the future. The future of cloud computing is very promising. It is said that about 70% of the world will be benefited from cloud environment while conducting their official or personal work and this is not an overestimate or exaggeration as we have already witnessed the positive aspects of cloud computing. Using email and connecting to social media through smartphones, watching movies over smartphones and uploading and accessing pictures from websites like Flicker are common examples of cloud computing in our day-to-day life.

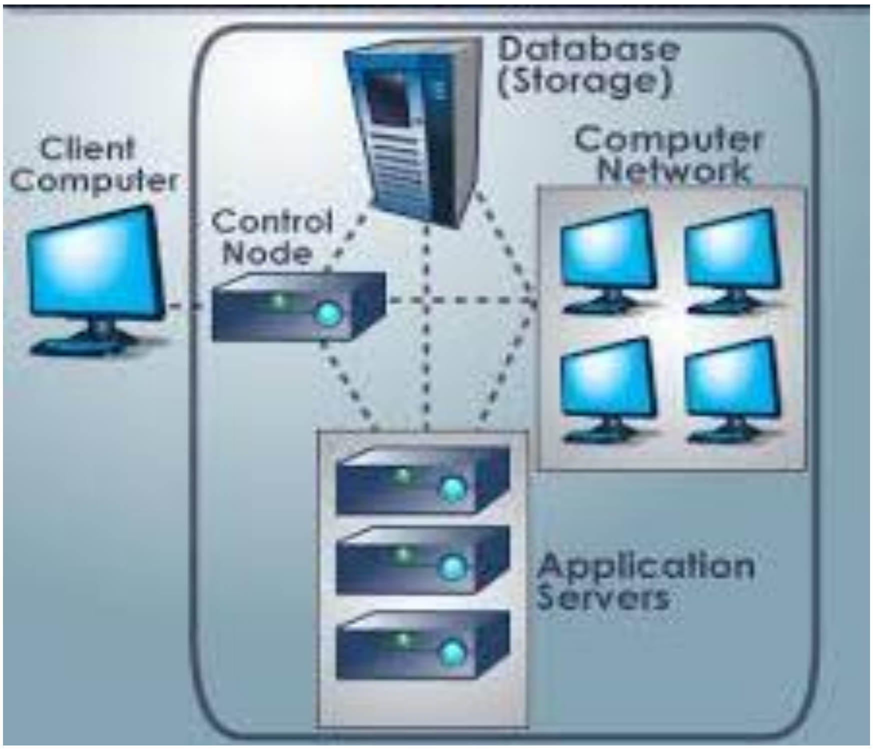


Fig1. *The figure depicts the working of cloud environment*

With the organizations shifting their data to the cloud, the need to protect this data against unauthorized access, modification, and threats like denial of services needs equal attention. The cryptographic algorithms find their application in securing the cloud environment. Cryptography is an art of science via which messages can be kept secure by converting data into the non-readable form. The single level cryptographic algorithms have failed to secure the data. Hence, the need for multilevel encryption and decryption algorithms is felt.

2. Implementation of Encryption Algorithms

Encryption has turned out to be an essential part of daily working as data needs to be encrypted and decrypted at various stages. The number of encryption algorithm is not limited to one, but there is a number of existing algorithms and there is a need to understand there working [10]. The working and implementation of DES and AES cryptography algorithms are explained as under.

2.1. DES (Data Encryption Standard)

The DES [2, 5, 6] algorithm was developed back in 1974 with the joint effort of IBM and the U.S. government. The primary objective behind its development was to set a standard that people could follow to securely communicate with each other. It operates on the blocks of 64 bits making use of 56 bits as a secret key. It is said that the removal of these 8 bits from the key was done to enable U.S. government agencies to crack the messages secretly.

The message is encrypted in 16 stages or rounds. Sixteen 48-bit keys are generated from the input key, one for each round. Every round makes use of eight S-boxes. These S-boxes are fixed in the requirement of the standard. With the use of S-boxes, the group of six bits is mapped to the groups of four bits. The block of the message is divided into two halves. The right half makes use of another fixed table and is expanded from 32 to 48 bits. The XOR operation is performed on the result obtained and the subkey for that round. The 48 resulting bits are transformed back to 32 bits using the S-boxes, which are permutated again using yet another fixed table. The systematically shuffled right half is now united with the left half applying XOR operation. In the next upcoming round, this combination forms the new left half [7, 10, 11].

The implementation of the working of DES algorithm has been performed using MATLAB simulation tool and is shown in the snapshots below.

Fig. 2 depicts the interface designed to carry out the process of encryption/decryption. The interface is divided into five subsections mentioned as under.

enter input – The message to be encrypted is provided as input in this section.

encrypt key – The private key to be used for encryption is written in this section.

encrypt data – The encrypted data after entering the private key is displayed in this section.

decrypt key - The same private key needs to be entered in this section to decrypt the message back to the original text.

decrypt data – The decrypted data i.e. the original plaintext is displayed in this section.

If the entered text is of 16 bytes or below 16 bytes, the encryption performed is of 128-bit. But if the entered text is beyond 16 bytes, then 256-bit encryption is performed.

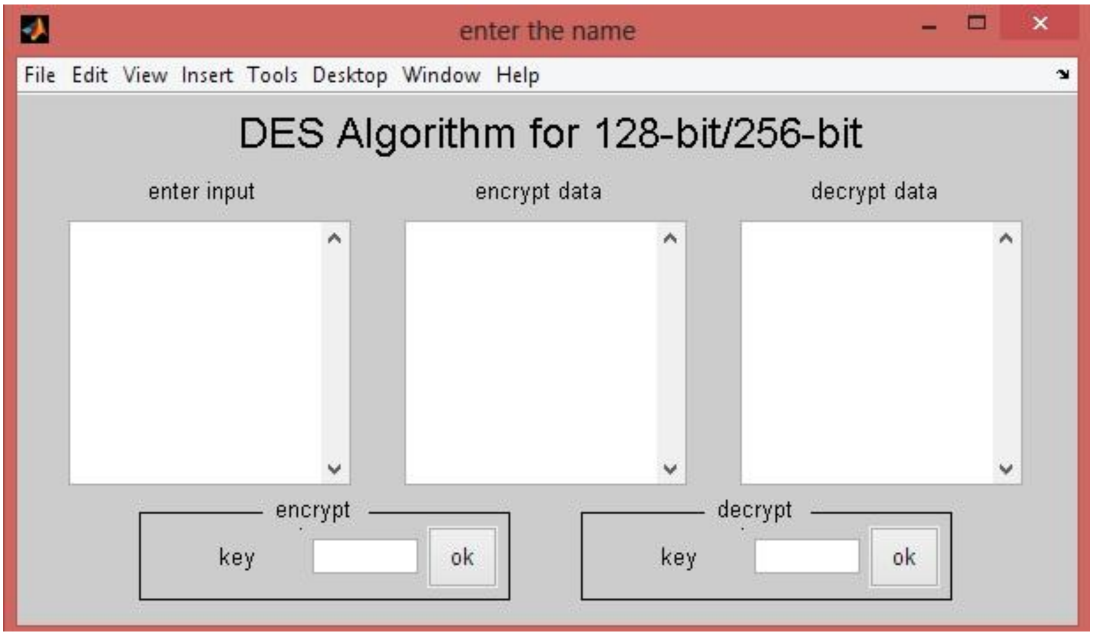
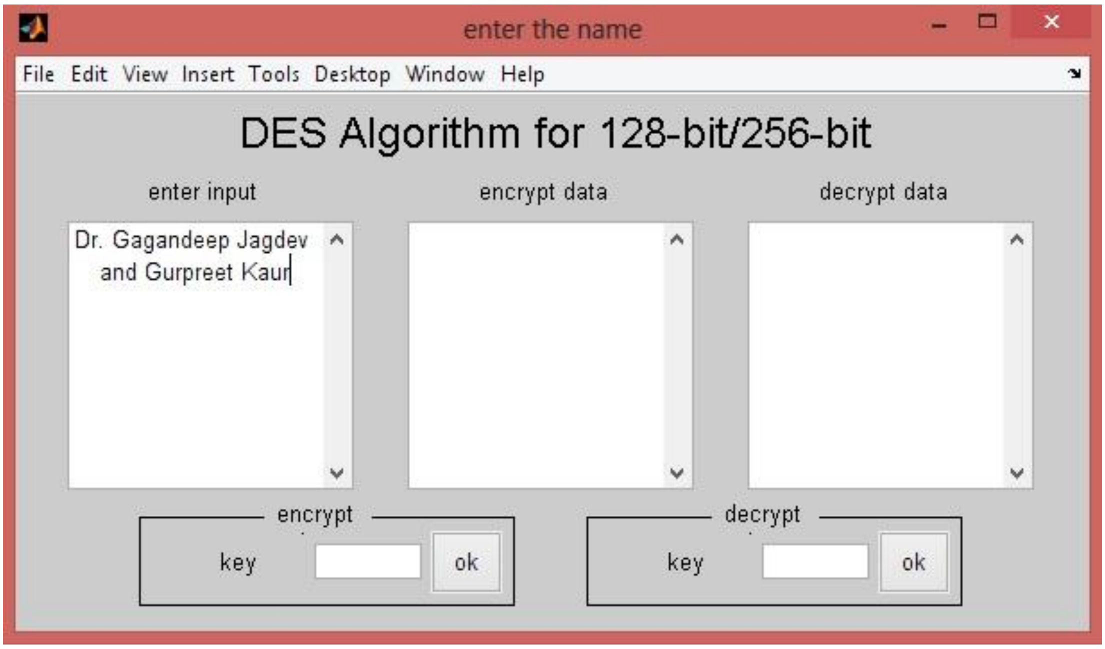

enter the name
File Edit View Insert Tools Desktop Window Help
DES Algorithm for 128-bit/256-bit
enter input
encrypt data
decrypt data
encrypt
key
ok
decrypt
key
ok

Fig2. *The figure depicts the designed interface to conduct encryption/decryption process using DES.*

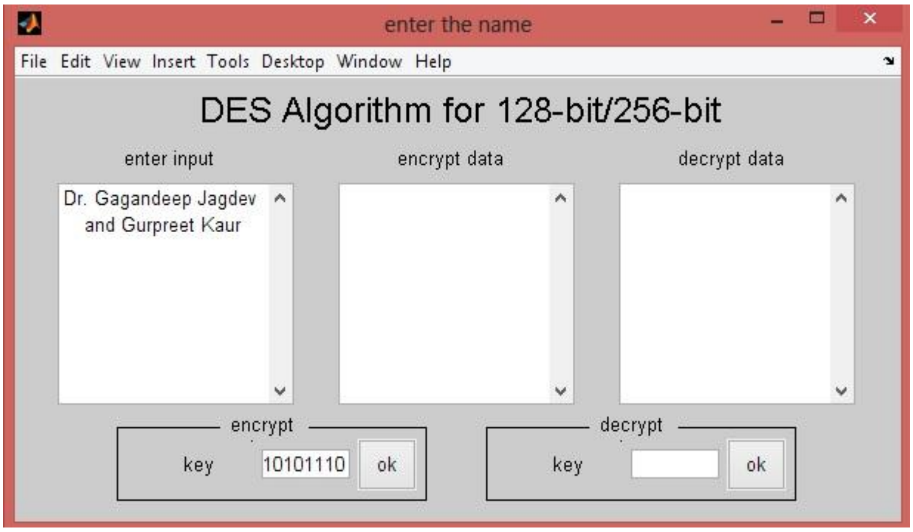
Fig. 3 shows the entered text i.e. plain text in the "*enter input"* section.


enter the name
File Edit View Insert Tools Desktop Window Help
DES Algorithm for 128-bit/256-bit
enter input
encrypt data
decrypt data
encrypt
key
ok
decrypt
key
ok

Dr. Gagandeep Jagdev
and Gurpreet Kaur

Fig3. *The figure shows the entered input in the "enter input" section*

After entering the input, there is a need to provide the private key which can be a combination of decimal, binary, octal or hexadecimal numbers as shown in Fig. 4. In the case under study, the private key is 10101110.

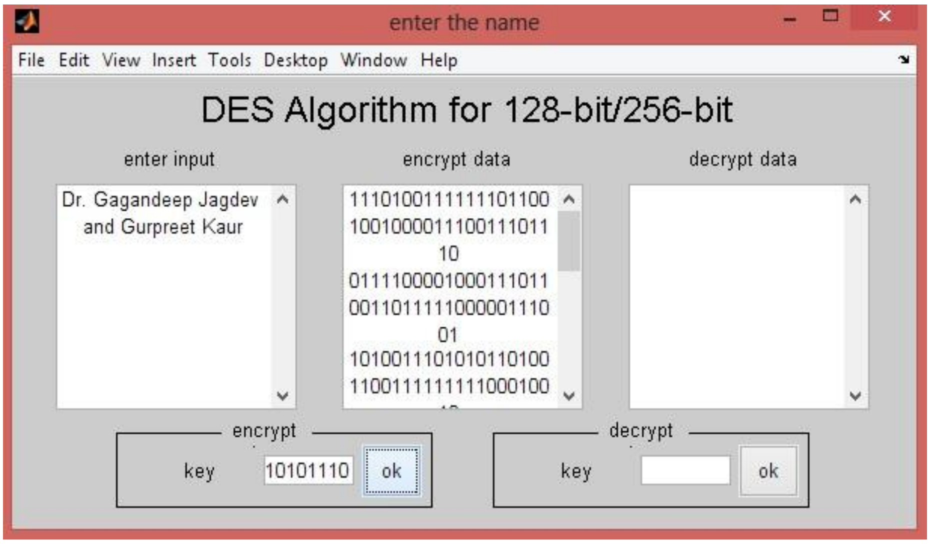

enter the name
File Edit View Insert Tools Desktop Window Help
DES Algorithm for 128-bit/256-bit
enter input
encrypt data
decrypt data
encrypt
key
ok
decrypt
key
ok

Dr. Gagandeep Jagdev and Gurpreet Kaur

key 10101110

Fig4. *The figure depicts the combination of binary numbers used as the private key in "encrypt key" section.*

Fig. 5 shows the encrypted message after entering the private key.


enter the name
File Edit View Insert Tools Desktop Window Help
DES Algorithm for 128-bit/256-bit
enter input
encrypt data
decrypt data
encrypt
key
ok
decrypt
key
ok

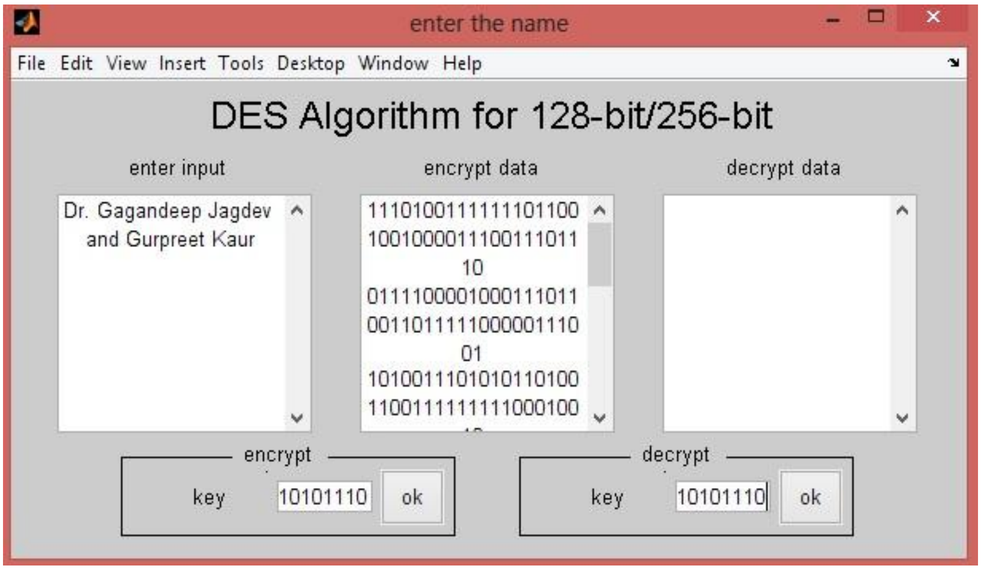
Dr. Gagandeep Jagdev and Gurpreet Kaur

```
1110100111111101100
1001000011100111011
        10
0111100001000111011
0011011111000001110
        01
1010011101010110100
1100111111111000100
```

key 10101110

Fig5. *The figure shows the encrypted message after entering the private key in "encrypt data" section*

To decrypt the message back into original text, there is a need to enter the same private key in "decrypt section" as shown in Fig. 6.


enter the name
File Edit View Insert Tools Desktop Window Help
DES Algorithm for 128-bit/256-bit
enter input
encrypt data
decrypt data
encrypt
key
ok
decrypt
key
ok

Dr. Gagandeep Jagdev and Gurpreet Kaur

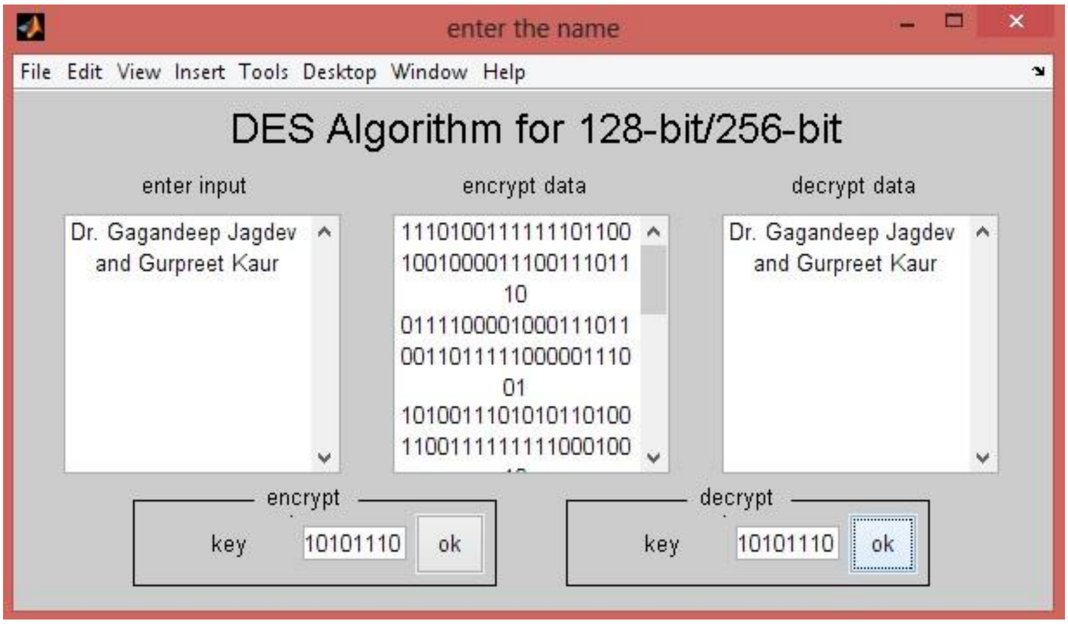
```
1110100111111101100
1001000011100111011
        10
0111100001000111011
0011011111000001110
        01
1010011101010110100
1100111111111000100
```

encrypt key 10101110

decrypt key 10101110

Fig6. *The figure shows the private key been entered in the "decrypt key" section*

Fig. 7 shows that entering the correct private key results in generating the accurate decrypted data. Fig. 8 depicts that on entering the incorrect private key, the decrypt data generated no longer remains precise.



DES Algorithm for 128-bit/256-bit

enter input

Dr. Gagandeep Jagdev
and Gurpreet Kaur

encrypt data

```
1110100111111101100
1001000011100111011
10
0111100001000111011
0011011111000001110
01
1010011101010110100
1100111111111000100
```

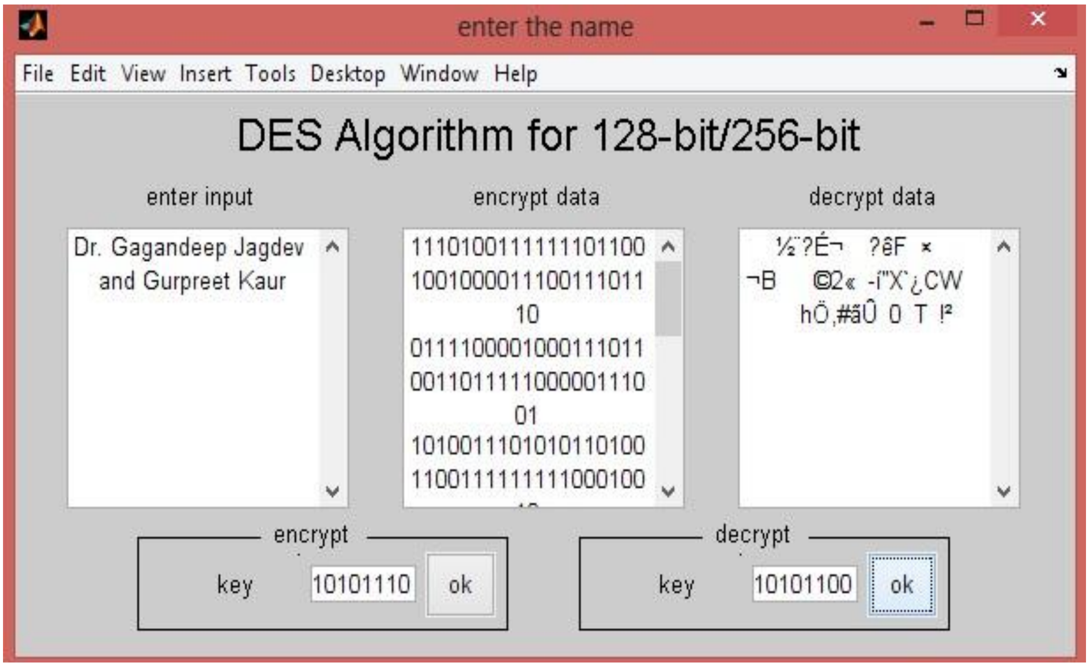
decrypt data

Dr. Gagandeep Jagdev
and Gurpreet Kaur

encrypt: key 10101110

decrypt: key 10101110

Fig7. *The figure depicts that on entering correct private key, the correct decrypted data is obtained*



DES Algorithm for 128-bit/256-bit

enter input

Dr. Gagandeep Jagdev
and Gurpreet Kaur

encrypt data

```
1110100111111101100
1001000011100111011
10
0111100001000111011
0011011111000001110
01
1010011101010110100
1100111111111000100
```

decrypt data

```
½¨?É¬   ?êF  ×
¬B    ©2«  -í"X`¿CW
hÖ,#ãÛ  0  T  !²
```

encrypt: key 10101110

decrypt: key 10101100

Fig8. *The figure shows that on entering the incorrect private key in "decrypt key" section, the text obtained in "decrypt data" section is not same as the originally entered text.*

2.2. AES (Advanced Encryption Standards)

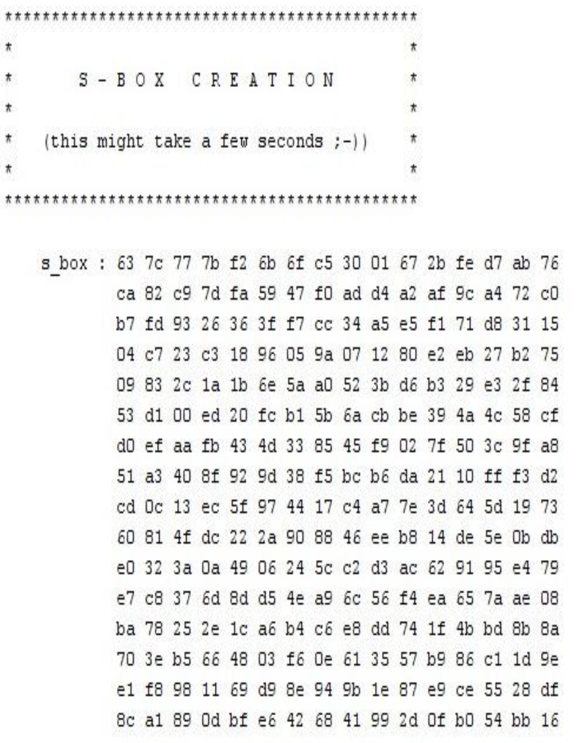
The AES [3, 4] algorithm makes use of a set of specially designed keys known as round keys is used in the encryption process. Along with other operations, these are applied on an array of data that holds an exactly single block of data i.e. the data to be decrypted. This array is known as a state array.

The steps involved in the encryption of 128-bit block are mentioned as under [8, 9].

- Firstly there is need to derive the set of round keys from the ciphertext.
- Initialization of state array with the block data needs to be performed.
- Initial round key should be added to the starting state array.

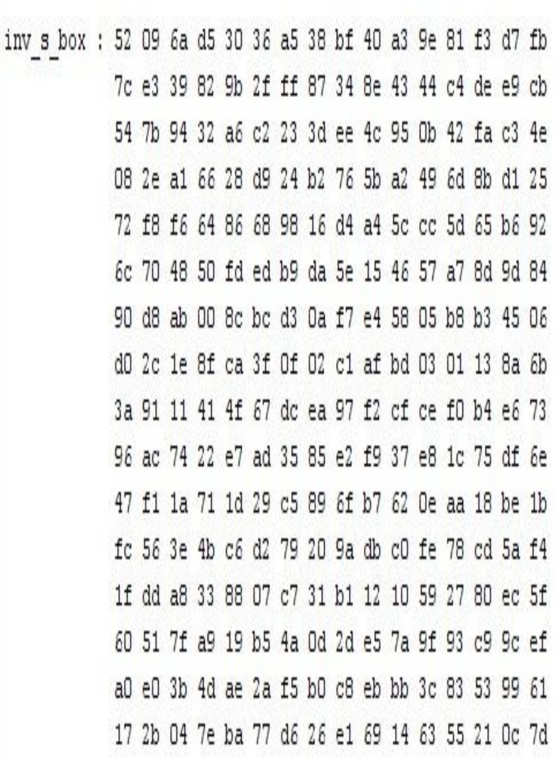
- Nine rounds of state manipulation are performed.
- Thereafter, the tenth and final round of state manipulation is performed.
- Finally, copy the final state array out as the encrypted data, i.e. ciphertext.

The tenth round involves a different manipulation in comparison with other nine rounds. The figures from Fig. 9 to Fig. 28 shows the detailed encryption process carried out in AES algorithm.



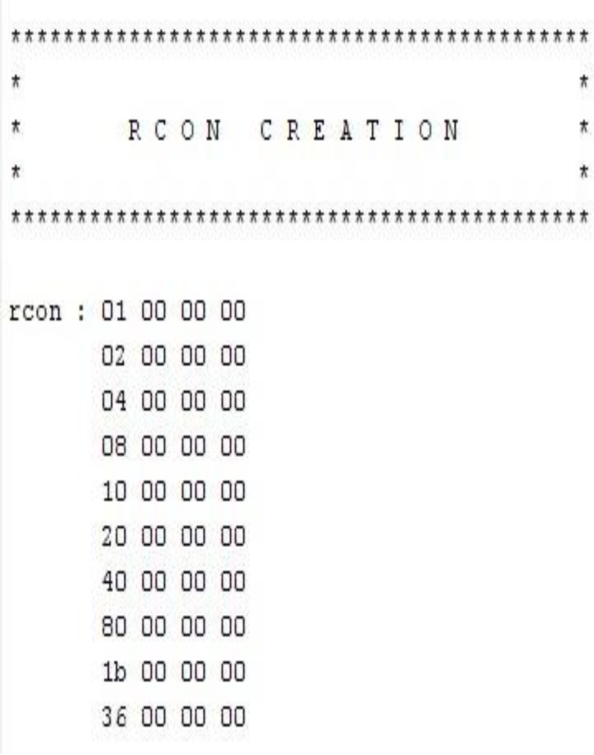
```
*********************************************
*                                           *
*       S - B O X   C R E A T I O N         *
*                                           *
*   (this might take a few seconds ;-))     *
*                                           *
*********************************************

    s_box : 63 7c 77 7b f2 6b 6f c5 30 01 67 2b fe d7 ab 76
            ca 82 c9 7d fa 59 47 f0 ad d4 a2 af 9c a4 72 c0
            b7 fd 93 26 36 3f f7 cc 34 a5 e5 f1 71 d8 31 15
            04 c7 23 c3 18 96 05 9a 07 12 80 e2 eb 27 b2 75
            09 83 2c 1a 1b 6e 5a a0 52 3b d6 b3 29 e3 2f 84
            53 d1 00 ed 20 fc b1 5b 6a cb be 39 4a 4c 58 cf
            d0 ef aa fb 43 4d 33 85 45 f9 02 7f 50 3c 9f a8
            51 a3 40 8f 92 9d 38 f5 bc b6 da 21 10 ff f3 d2
            cd 0c 13 ec 5f 97 44 17 c4 a7 7e 3d 64 5d 19 73
            60 81 4f dc 22 2a 90 88 46 ee b8 14 de 5e 0b db
            e0 32 3a 0a 49 06 24 5c c2 d3 ac 62 91 95 e4 79
            e7 c8 37 6d 8d d5 4e a9 6c 56 f4 ea 65 7a ae 08
            ba 78 25 2e 1c a6 b4 c6 e8 dd 74 1f 4b bd 8b 8a
            70 3e b5 66 48 03 f6 0e 61 35 57 b9 86 c1 1d 9e
            e1 f8 98 11 69 d9 8e 94 9b 1e 87 e9 ce 55 28 df
            8c a1 89 0d bf e6 42 68 41 99 2d 0f b0 54 bb 16
```

Fig9. *S-Box creation*



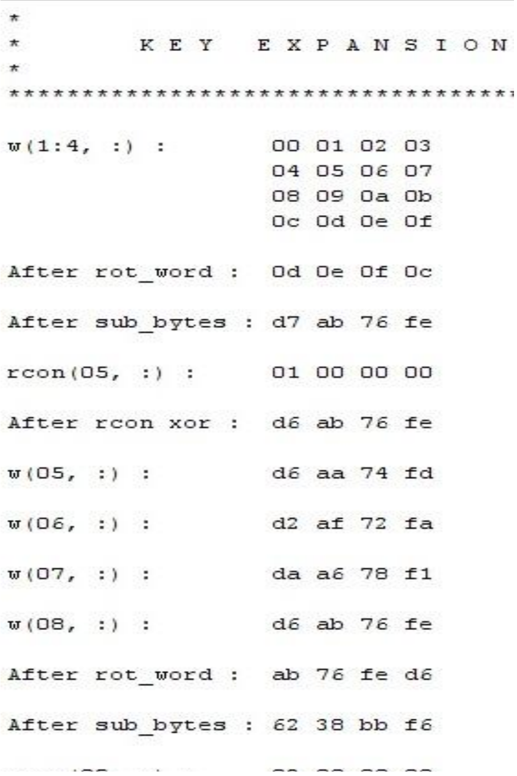
```
inv_s_box : 52 09 6a d5 30 36 a5 38 bf 40 a3 9e 81 f3 d7 fb
            7c e3 39 82 9b 2f ff 87 34 8e 43 44 c4 de e9 cb
            54 7b 94 32 a6 c2 23 3d ee 4c 95 0b 42 fa c3 4e
            08 2e a1 66 28 d9 24 b2 76 5b a2 49 6d 8b d1 25
            72 f8 f6 64 86 68 98 16 d4 a4 5c cc 5d 65 b6 92
            6c 70 48 50 fd ed b9 da 5e 15 46 57 a7 8d 9d 84
            90 d8 ab 00 8c bc d3 0a f7 e4 58 05 b8 b3 45 06
            d0 2c 1e 8f ca 3f 0f 02 c1 af bd 03 01 13 8a 6b
            3a 91 11 41 4f 67 dc ea 97 f2 cf ce f0 b4 e6 73
            96 ac 74 22 e7 ad 35 85 e2 f9 37 e8 1c 75 df 6e
            47 f1 1a 71 1d 29 c5 89 6f b7 62 0e aa 18 be 1b
            fc 56 3e 4b c6 d2 79 20 9a db c0 fe 78 cd 5a f4
            1f dd a8 33 88 07 c7 31 b1 12 10 59 27 80 ec 5f
            60 51 7f a9 19 b5 4a 0d 2d e5 7a 9f 93 c9 9c ef
            a0 e0 3b 4d ae 2a f5 b0 c8 eb bb 3c 83 53 99 61
            17 2b 04 7e ba 77 d6 26 e1 69 14 63 55 21 0c 7d
```

Fig10. *inv_s_box creation*



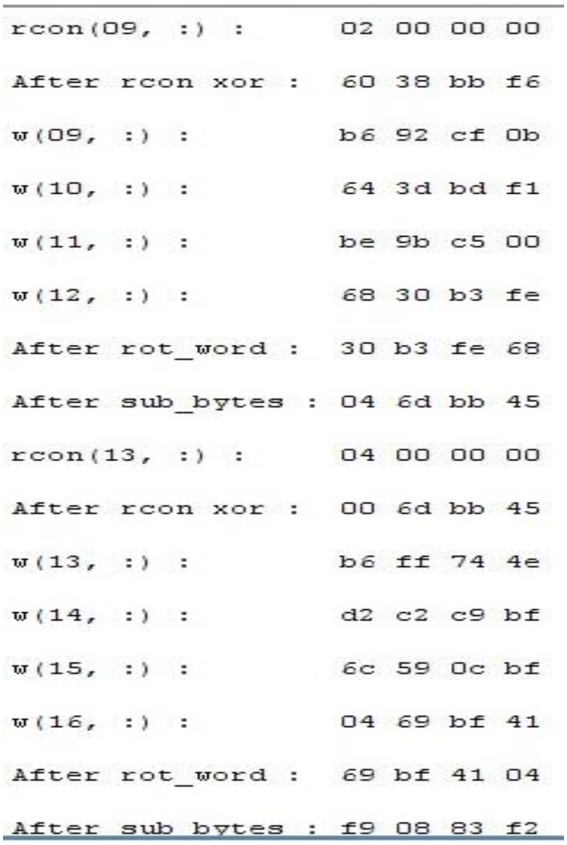
```
*********************************************
*                                           *
*       R C O N   C R E A T I O N           *
*                                           *
*********************************************

rcon : 01 00 00 00
       02 00 00 00
       04 00 00 00
       08 00 00 00
       10 00 00 00
       20 00 00 00
       40 00 00 00
       80 00 00 00
       1b 00 00 00
       36 00 00 00
```

Fig11. *RCON creation*



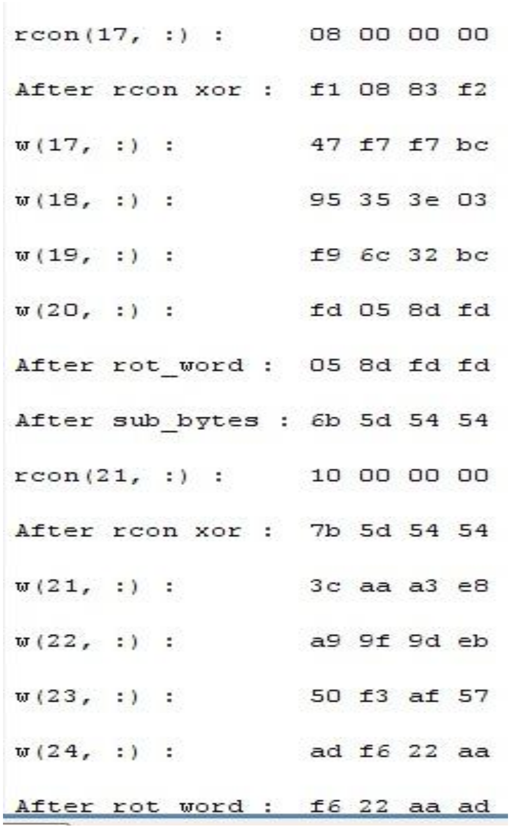
```
*
*        K E Y   E X P A N S I O N
*
**********************************

w(1:4, :) :          00 01 02 03
                     04 05 06 07
                     08 09 0a 0b
                     0c 0d 0e 0f

After rot_word :     0d 0e 0f 0c

After sub_bytes :    d7 ab 76 fe

rcon(05, :) :        01 00 00 00

After rcon xor :     d6 ab 76 fe

w(05, :) :           d6 aa 74 fd

w(06, :) :           d2 af 72 fa

w(07, :) :           da a6 78 f1

w(08, :) :           d6 ab 76 fe

After rot_word :     ab 76 fe d6

After sub_bytes :    62 38 bb f6
```

Fig12. *Key Expansion 1*



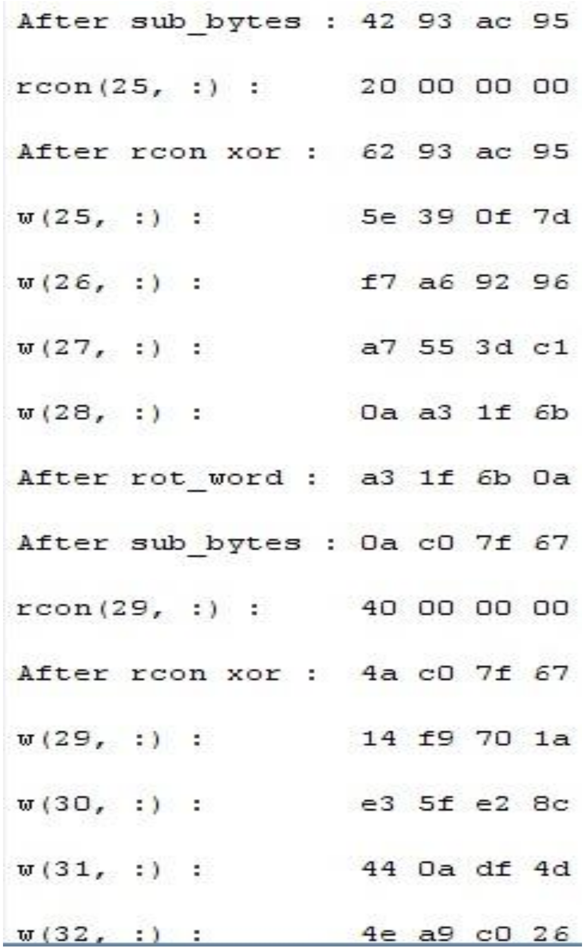
```
rcon(09, :) :         02 00 00 00

After rcon xor :      60 38 bb f6

w(09, :) :            b6 92 cf 0b

w(10, :) :            64 3d bd f1

w(11, :) :            be 9b c5 00

w(12, :) :            68 30 b3 fe

After rot_word :      30 b3 fe 68

After sub_bytes :     04 6d bb 45

rcon(13, :) :         04 00 00 00

After rcon xor :      00 6d bb 45

w(13, :) :            b6 ff 74 4e

w(14, :) :            d2 c2 c9 bf

w(15, :) :            6c 59 0c bf

w(16, :) :            04 69 bf 41

After rot_word :      69 bf 41 04

After sub_bytes :     f9 08 83 f2
```

Fig13. *Key Expansion 2*



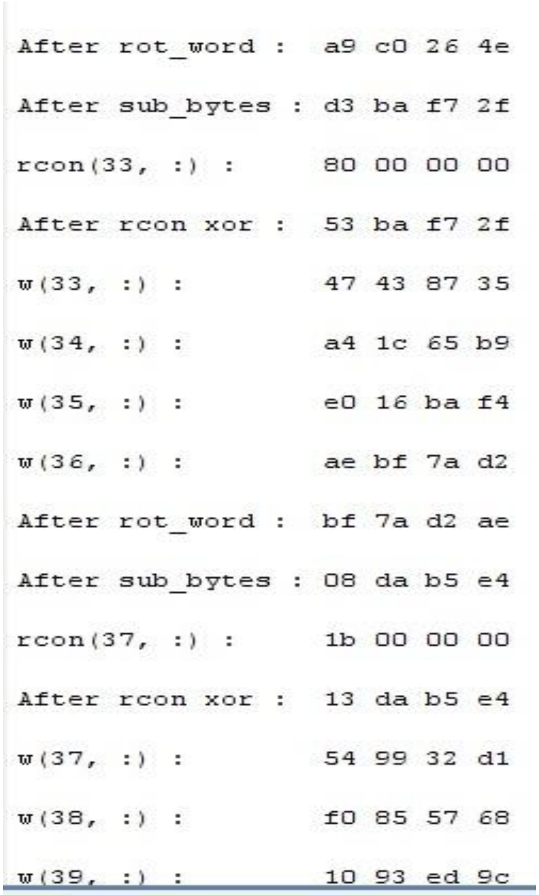
```
rcon(17, :) :         08 00 00 00

After rcon xor :      f1 08 83 f2

w(17, :) :            47 f7 f7 bc

w(18, :) :            95 35 3e 03

w(19, :) :            f9 6c 32 bc

w(20, :) :            fd 05 8d fd

After rot_word :      05 8d fd fd

After sub_bytes :     6b 5d 54 54

rcon(21, :) :         10 00 00 00

After rcon xor :      7b 5d 54 54

w(21, :) :            3c aa a3 e8

w(22, :) :            a9 9f 9d eb

w(23, :) :            50 f3 af 57

w(24, :) :            ad f6 22 aa

After rot_word :      f6 22 aa ad
```

Fig14. *Key Expansion 3*



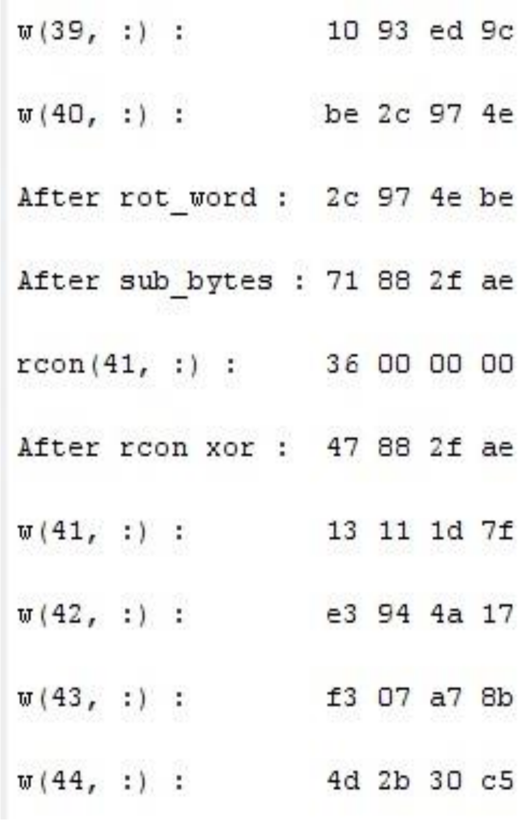
```
After sub_bytes :     42 93 ac 95

rcon(25, :) :         20 00 00 00

After rcon xor :      62 93 ac 95

w(25, :) :            5e 39 0f 7d

w(26, :) :            f7 a6 92 96

w(27, :) :            a7 55 3d c1

w(28, :) :            0a a3 1f 6b

After rot_word :      a3 1f 6b 0a

After sub_bytes :     0a c0 7f 67

rcon(29, :) :         40 00 00 00

After rcon xor :      4a c0 7f 67

w(29, :) :            14 f9 70 1a

w(30, :) :            e3 5f e2 8c

w(31, :) :            44 0a df 4d

w(32, :) :            4e a9 c0 26
```

Fig15. *Key Expansion 4*



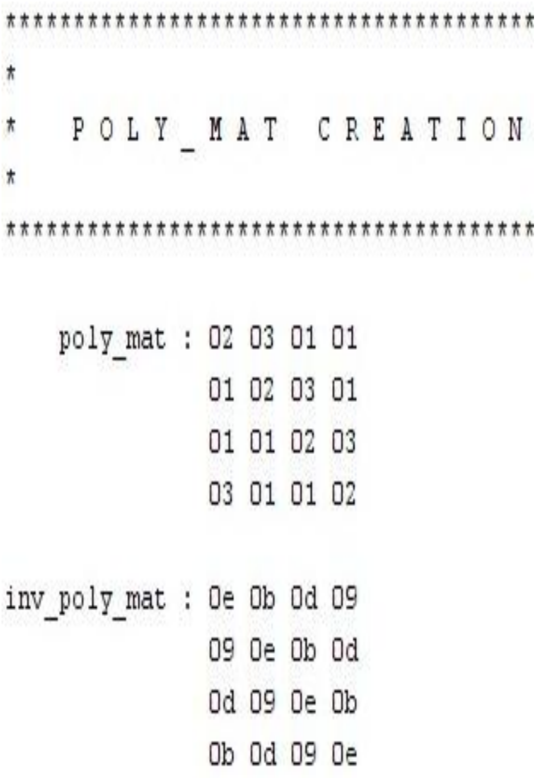
```
After rot_word :      a9 c0 26 4e

After sub_bytes :     d3 ba f7 2f

rcon(33, :) :         80 00 00 00

After rcon xor :      53 ba f7 2f

w(33, :) :            47 43 87 35

w(34, :) :            a4 1c 65 b9

w(35, :) :            e0 16 ba f4

w(36, :) :            ae bf 7a d2

After rot_word :      bf 7a d2 ae

After sub_bytes :     08 da b5 e4

rcon(37, :) :         1b 00 00 00

After rcon xor :      13 da b5 e4

w(37, :) :            54 99 32 d1

w(38, :) :            f0 85 57 68

w(39, :) :            10 93 ed 9c
```

Fig16. *Key Expansion 5*



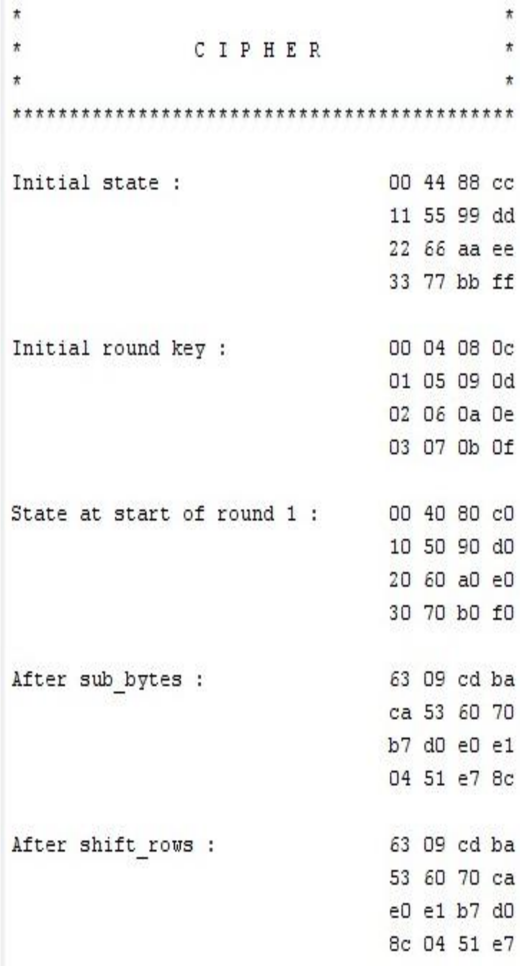
```
w(39, :) :          10 93 ed 9c

w(40, :) :          be 2c 97 4e

After rot_word :    2c 97 4e be

After sub_bytes :   71 88 2f ae

rcon(41, :) :       36 00 00 00

After rcon xor :    47 88 2f ae

w(41, :) :          13 11 1d 7f

w(42, :) :          e3 94 4a 17

w(43, :) :          f3 07 a7 8b

w(44, :) :          4d 2b 30 c5
```

Fig17. *Key Expansion 6*

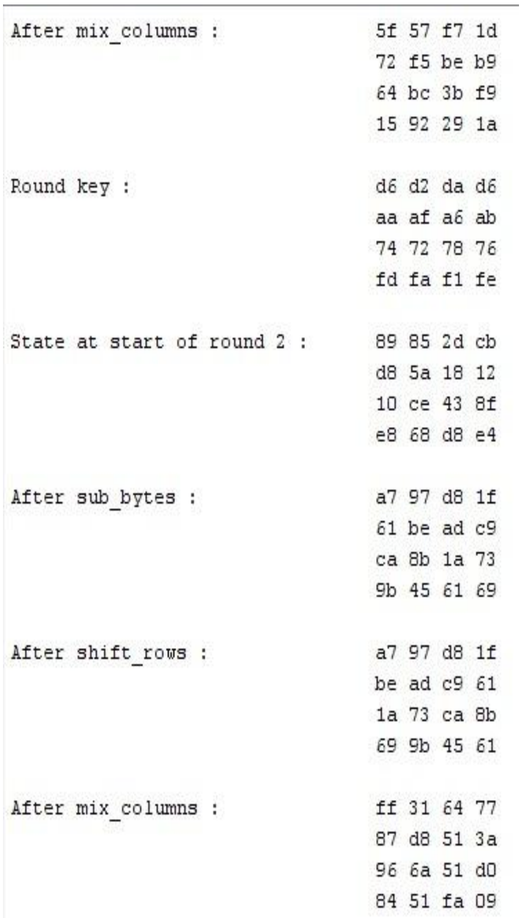


```
****************************************
*
*  P O L Y _ M A T  C R E A T I O N
*
****************************************

    poly_mat : 02 03 01 01
               01 02 03 01
               01 01 02 03
               03 01 01 02

inv_poly_mat : 0e 0b 0d 09
               09 0e 0b 0d
               0d 09 0e 0b
               0b 0d 09 0e
```

Fig18. *Poly_mat and inv_poly_mat*



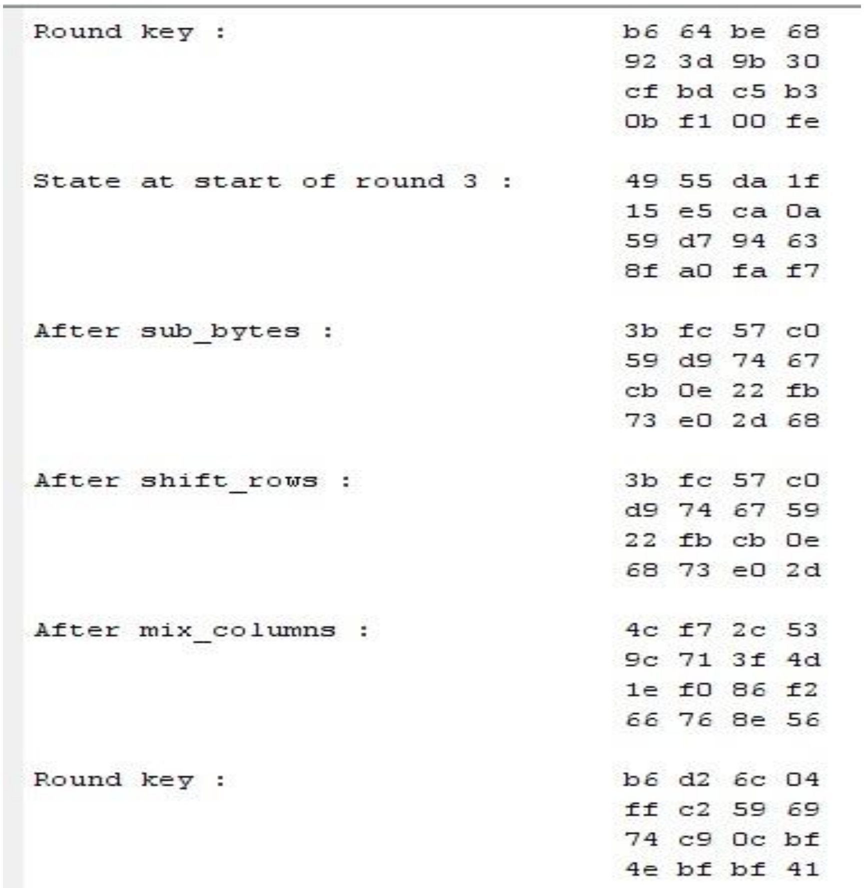
```
*                                            *
*              C I P H E R                   *
*                                            *
**********************************************

Initial state :                   00 44 88 cc
                                  11 55 99 dd
                                  22 66 aa ee
                                  33 77 bb ff

Initial round key :               00 04 08 0c
                                  01 05 09 0d
                                  02 06 0a 0e
                                  03 07 0b 0f

State at start of round 1 :       00 40 80 c0
                                  10 50 90 d0
                                  20 60 a0 e0
                                  30 70 b0 f0

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

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



```
After mix_columns :               5f 57 f7 1d
                                  72 f5 be b9
                                  64 bc 3b f9
                                  15 92 29 1a

Round key :                       d6 d2 da d6
                                  aa af a6 ab
                                  74 72 78 76
                                  fd fa f1 fe

State at start of round 2 :       89 85 2d cb
                                  d8 5a 18 12
                                  10 ce 43 8f
                                  e8 68 d8 e4

After sub_bytes :                 a7 97 d8 1f
                                  61 be ad c9
                                  ca 8b 1a 73
                                  9b 45 61 69

After shift_rows :                a7 97 d8 1f
                                  be ad c9 61
                                  1a 73 ca 8b
                                  69 9b 45 61

After mix_columns :               ff 31 64 77
                                  87 d8 51 3a
                                  96 6a 51 d0
                                  84 51 fa 09
```

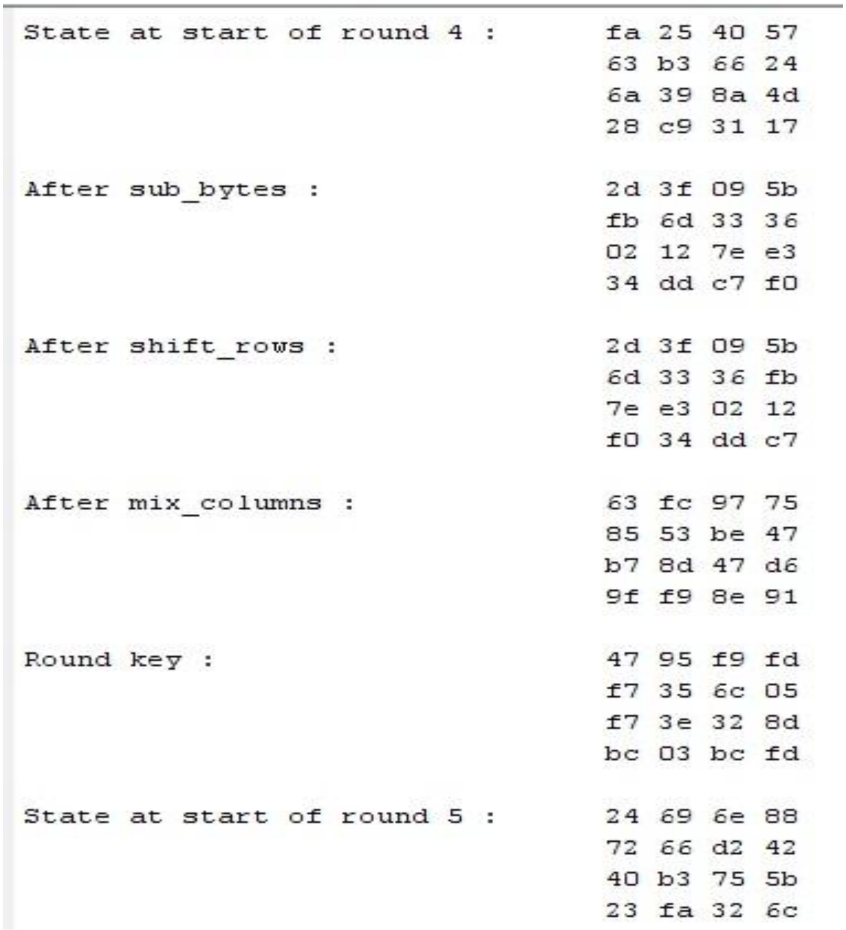
Fig19. *Cipher text formation Round 1*

Fig20. *Cipher text formation Round 2*



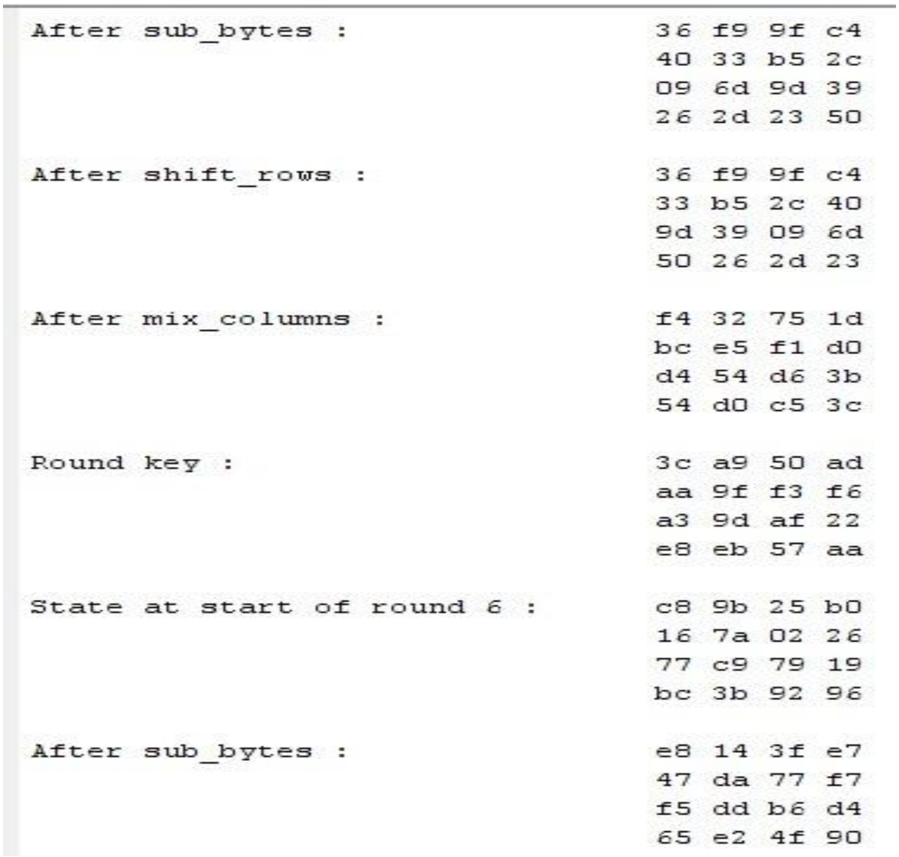
```
Round key :                          b6 64 be 68
                                     92 3d 9b 30
                                     cf bd c5 b3
                                     0b f1 00 fe

State at start of round 3 :          49 55 da 1f
                                     15 e5 ca 0a
                                     59 d7 94 63
                                     8f a0 fa f7

After sub_bytes :                    3b fc 57 c0
                                     59 d9 74 67
                                     cb 0e 22 fb
                                     73 e0 2d 68

After shift_rows :                   3b fc 57 c0
                                     d9 74 67 59
                                     22 fb cb 0e
                                     68 73 e0 2d

After mix_columns :                  4c f7 2c 53
                                     9c 71 3f 4d
                                     1e f0 86 f2
                                     66 76 8e 56

Round key :                          b6 d2 6c 04
                                     ff c2 59 69
                                     74 c9 0c bf
                                     4e bf bf 41
```

Fig21. *Cipher text formation Round 3*



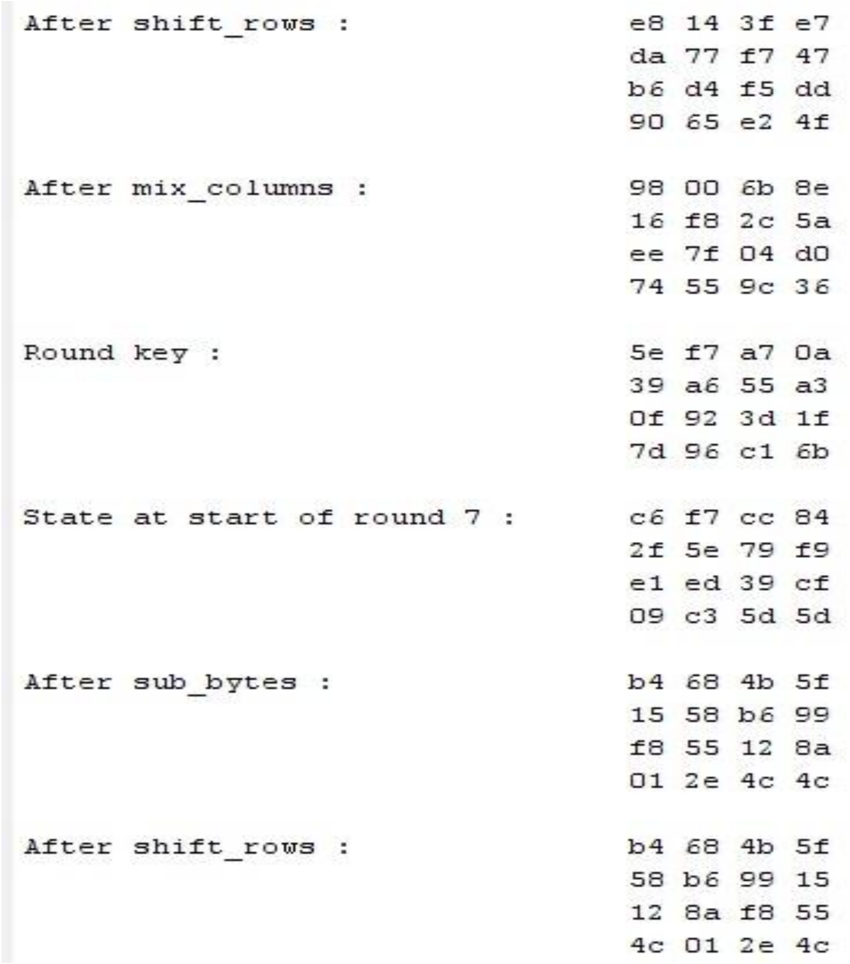
```
State at start of round 4 :          fa 25 40 57
                                     63 b3 66 24
                                     6a 39 8a 4d
                                     28 c9 31 17

After sub_bytes :                    2d 3f 09 5b
                                     fb 6d 33 36
                                     02 12 7e e3
                                     34 dd c7 f0

After shift_rows :                   2d 3f 09 5b
                                     6d 33 36 fb
                                     7e e3 02 12
                                     f0 34 dd c7

After mix_columns :                  63 fc 97 75
                                     85 53 be 47
                                     b7 8d 47 d6
                                     9f f9 8e 91

Round key :                          47 95 f9 fd
                                     f7 35 6c 05
                                     f7 3e 32 8d
                                     bc 03 bc fd

State at start of round 5 :          24 69 6e 88
                                     72 66 d2 42
                                     40 b3 75 5b
                                     23 fa 32 6c
```

Fig22. *Cipher text formation Round 4 and 5*



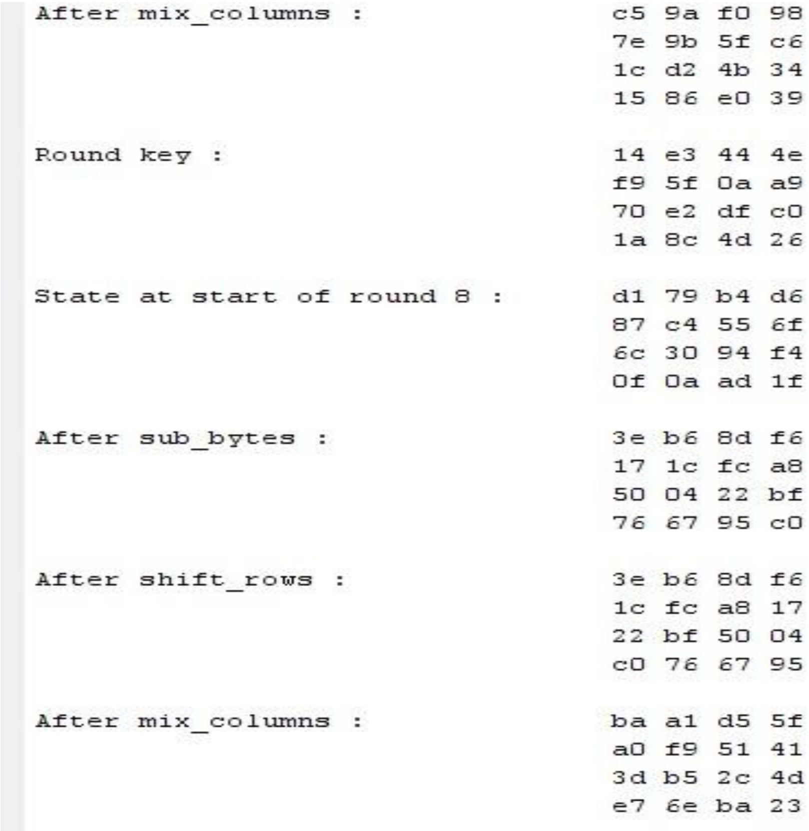
```
After sub_bytes :                        36 f9 9f c4
                                         40 33 b5 2c
                                         09 6d 9d 39
                                         26 2d 23 50

After shift_rows :                       36 f9 9f c4
                                         33 b5 2c 40
                                         9d 39 09 6d
                                         50 26 2d 23

After mix_columns :                      f4 32 75 1d
                                         bc e5 f1 d0
                                         d4 54 d6 3b
                                         54 d0 c5 3c

Round key :                              3c a9 50 ad
                                         aa 9f f3 f6
                                         a3 9d af 22
                                         e8 eb 57 aa

State at start of round 6 :              c8 9b 25 b0
                                         16 7a 02 26
                                         77 c9 79 19
                                         bc 3b 92 96

After sub_bytes :                        e8 14 3f e7
                                         47 da 77 f7
                                         f5 dd b6 d4
                                         65 e2 4f 90
```

Fig23. *Cipher text formation Round 5 and 6*



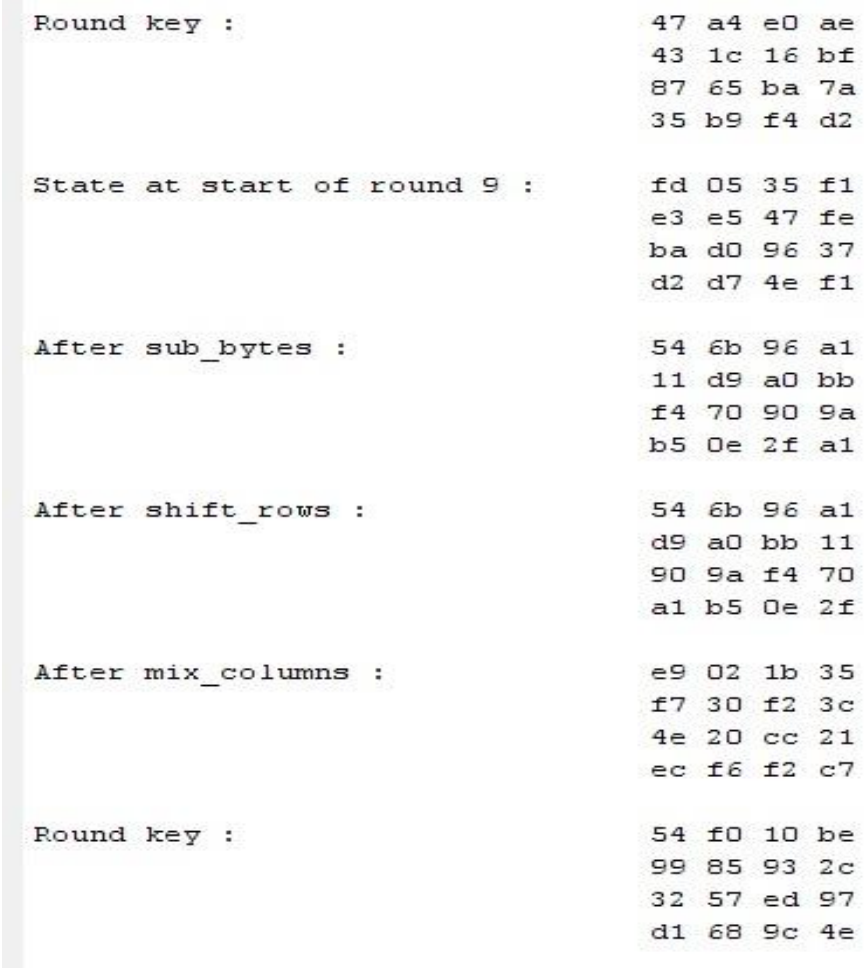
```
After shift_rows :                       e8 14 3f e7
                                         da 77 f7 47
                                         b6 d4 f5 dd
                                         90 65 e2 4f

After mix_columns :                      98 00 6b 8e
                                         16 f8 2c 5a
                                         ee 7f 04 d0
                                         74 55 9c 36

Round key :                              5e f7 a7 0a
                                         39 a6 55 a3
                                         0f 92 3d 1f
                                         7d 96 c1 6b

State at start of round 7 :              c6 f7 cc 84
                                         2f 5e 79 f9
                                         e1 ed 39 cf
                                         09 c3 5d 5d

After sub_bytes :                        b4 68 4b 5f
                                         15 58 b6 99
                                         f8 55 12 8a
                                         01 2e 4c 4c

After shift_rows :                       b4 68 4b 5f
                                         58 b6 99 15
                                         12 8a f8 55
                                         4c 01 2e 4c
```

Fig24. *Cipher text formation Round 6 and 7*



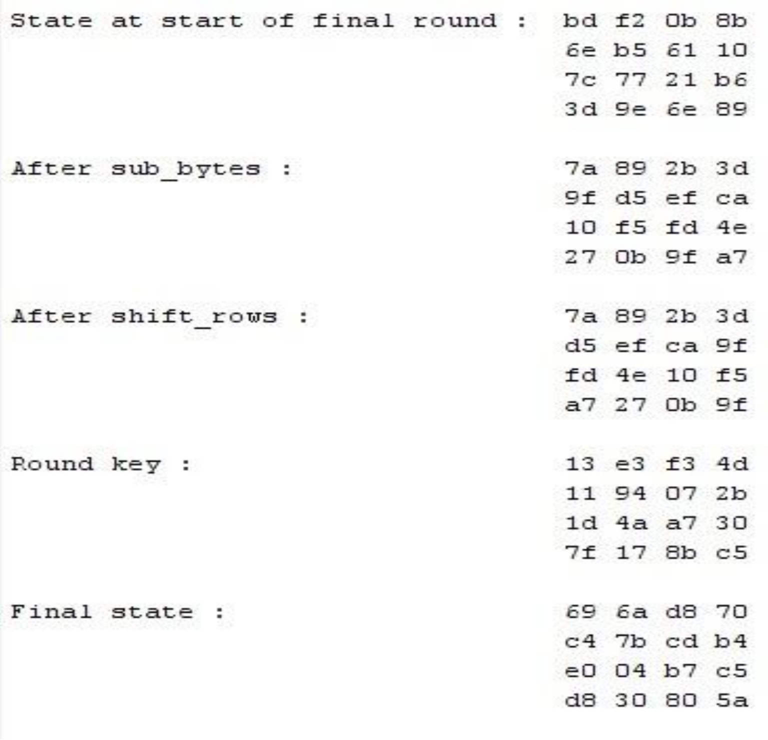
```
After mix_columns :                 c5 9a f0 98
                                    7e 9b 5f c6
                                    1c d2 4b 34
                                    15 86 e0 39

Round key :                         14 e3 44 4e
                                    f9 5f 0a a9
                                    70 e2 df c0
                                    1a 8c 4d 26

State at start of round 8 :         d1 79 b4 d6
                                    87 c4 55 6f
                                    6c 30 94 f4
                                    0f 0a ad 1f

After sub_bytes :                   3e b6 8d f6
                                    17 1c fc a8
                                    50 04 22 bf
                                    76 67 95 c0

After shift_rows :                  3e b6 8d f6
                                    1c fc a8 17
                                    22 bf 50 04
                                    c0 76 67 95

After mix_columns :                 ba a1 d5 5f
                                    a0 f9 51 41
                                    3d b5 2c 4d
                                    e7 6e ba 23
```

Fig25. *Cipher text formation Round 7 and 8*



```
Round key :                         47 a4 e0 ae
                                    43 1c 16 bf
                                    87 65 ba 7a
                                    35 b9 f4 d2

State at start of round 9 :         fd 05 35 f1
                                    e3 e5 47 fe
                                    ba d0 96 37
                                    d2 d7 4e f1

After sub_bytes :                   54 6b 96 a1
                                    11 d9 a0 bb
                                    f4 70 90 9a
                                    b5 0e 2f a1

After shift_rows :                  54 6b 96 a1
                                    d9 a0 bb 11
                                    90 9a f4 70
                                    a1 b5 0e 2f

After mix_columns :                 e9 02 1b 35
                                    f7 30 f2 3c
                                    4e 20 cc 21
                                    ec f6 f2 c7

Round key :                         54 f0 10 be
                                    99 85 93 2c
                                    32 57 ed 97
                                    d1 68 9c 4e
```

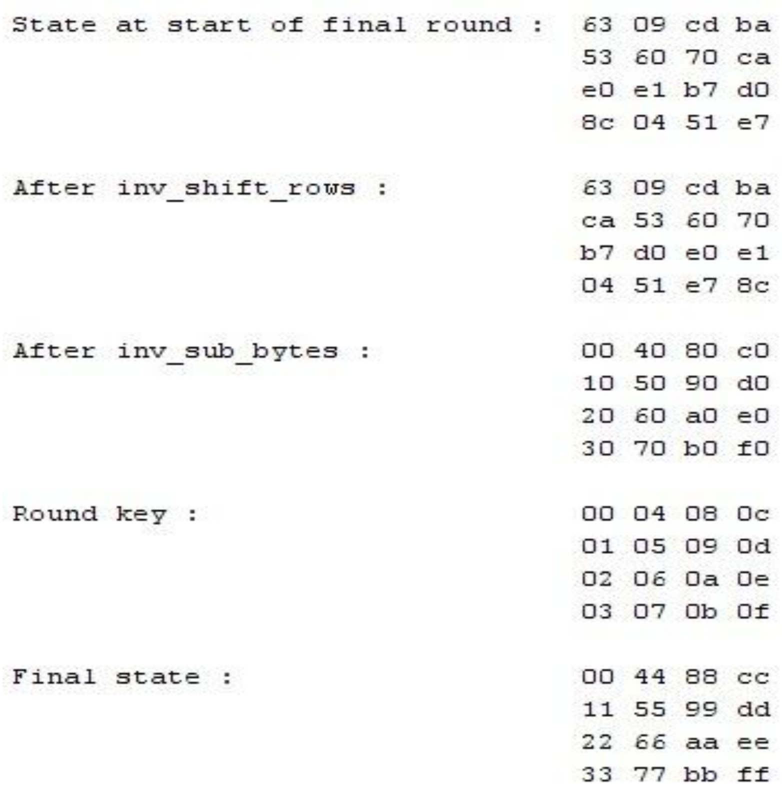
Fig26. *Cipher text formation Round 8 and 9*



```
State at start of final round :  bd f2 0b 8b
                                 6e b5 61 10
                                 7c 77 21 b6
                                 3d 9e 6e 89

After sub_bytes :                7a 89 2b 3d
                                 9f d5 ef ca
                                 10 f5 fd 4e
                                 27 0b 9f a7

After shift_rows :               7a 89 2b 3d
                                 d5 ef ca 9f
                                 fd 4e 10 f5
                                 a7 27 0b 9f

Round key :                      13 e3 f3 4d
                                 11 94 07 2b
                                 1d 4a a7 30
                                 7f 17 8b c5

Final state :                    69 6a d8 70
                                 c4 7b cd b4
                                 e0 04 b7 c5
                                 d8 30 80 5a
```

Fig27. *Cipher text formation Final Round*

After reaching the final state of cipher text, the process of generating inverse cipher text begins which initiates back from round 9 to round 1 and hence the final encrypted text is generated as shown in Fig. 28.



```
State at start of final round :  63 09 cd ba
                                 53 60 70 ca
                                 e0 e1 b7 d0
                                 8c 04 51 e7

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

After inv_sub_bytes :            00 40 80 c0
                                 10 50 90 d0
                                 20 60 a0 e0
                                 30 70 b0 f0

Round key :                      00 04 08 0c
                                 01 05 09 0d
                                 02 06 0a 0e
                                 03 07 0b 0f

Final state :                    00 44 88 cc
                                 11 55 99 dd
                                 22 66 aa ee
                                 33 77 bb ff
```

Fig28. *The figure depicts the final encrypted text obtained*

3. Assurances Made by Cloud Computing

Below mentioned are few prominent promises made by cloud computing.

3.1. Presence of Internet will Boost its Future

With the presence of high-speed internet, the cloud computing has witnessed a huge leap. Already the Indian government is busy in connecting every village with wireless internet services.

3.2. No More Software Updates

It is often witnessed that much of the time of computer professionals are utilized in downloading newer versions of different software's which assist them in their job. But with the existence of cloud computing, this task has been eliminated as now it's the job of CSP to provide with latest software's and hence professional can fully concentrate on their primary job.

3.3. Hardware Optional

With the advent of cloud computing, the need to purchase expensive hardware like storage devices has been eradicated as now the data can be stored in the cloud. All data with assured backup is now stored in the cloud and the fear of losing the critical no longer exists.

3.4. Entertainment Unlimited

As there is no longer any concerns related to hardware, the users can enjoy limitless entertainment options.

3.5. Medical Treatments Simplified

Most of the medical treatments require computer assistance, the data needs to be stored and searched frequently and cloud computing is all set to play its part in such therapies.

3.6. Weather Forecasting

The cloud computing also assists in making precise predictions regarding weather forecast because of increased computing power.

3.7. Education for All

The cloud computing has made it possible to deliver education to the doorsteps of the learners. An enormous amount of free course material has been uploaded to the cloud to digitize education.

4. Conclusion

The paper discussed working and showed the detailed implementation of DES and AES cryptographic algorithms. The paper also highlighted the importance of security in cloud environment. Encryption algorithms play an important role in data security on cloud computing. Currently AES is most advanced and adopted algorithm for performing cryptography in both hardware and software. No successful cryptanalytic attacks against AES have been discovered till date

References

[1] Akashdeep Bhardwaj, G.V.B. Subrahmanyam et al., "Security Algorithms for Cloud Computing", Elsevier, Procedia Computer Science, Volume 85, Pages 535-542.

[2] Priyadarshini Patil et al., "A comprehensive Evaluation of Cryptographic Algorithms: DES,3DES, AES, RSA and Blowfish", 2015, Volume 78, 2016, Pages 617-624.

[3] Abidalrahman Mohd. et al.," AES-512: 512-bit Advanced Encryption Standard algorithm design and evaluation",7th International Conference on Information Assurance and Security, IAS 2011, Melacca, Malaysia, December 5-8, 2011.

[4] Abhilasha CP et al., "Software Implementation of AES Encryption Algorithm", IJARCSSE, 2016.

[5] M.Meena et al., "A study and comparative analysis of cryptographic algorithms for various file formats.",IJSR, 2013, ISSN:2319-7064.

[6] Miss. Shakeeba et al., "Cloud Security using Multilevel Encryption Algorithms", IJARCCE, 2016, ISSN (online):2278-1021.

[7] Kumar Y., Munjal R. et al. "Comparison of symmetric and asymmetric cryptography with existing vulnerabilities and countermeasures"**,** *IJAFRC*, Volume 1, Issue 6, June 2014.

[8] Pahal R., Kumar V., "Efficient Implementation of AES", IJARCSSE, Volume 3, Issue 7, July 2013.

[9] Aggarwal A., Singh G. et al., " Implementation of AES algorithm", *International Journal of Engineering Research & Science (IJOER),* Vol-2, Issue-4 April- 2016, pp. 112-116.

[10] Meena M.et al., “A study and comparative analysis of cryptographic algorithms for various file formats”,*International Journal of Science and Research (IJSR)*, 2013, pp. 991 - 995.

[11] Shakeeba et al., “Cloud Security using Multilevel Encryption Algorithms”, *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)*, Vol. 5, Issue 1, January 2016, pp. 70 – 75.

AUTHOR’S BIOGRAPHY

Dr. Gagandeep Jagdev, is a faculty member in Dept. of Computer Science, Punjabi University Guru Kashi College, Damdama Sahib (PB). His total teaching experience is above 10 years and has above 108 international and national publications in reputed journals and conferences to his credit. He is also a member of editorial board of several international peer-reviewed journals and has been active Technical Program Committee member of several international and national conferences conducted by renowned universities and academic institutions. His field of expertise is Big Data, ANN, Biometrics, RFID, Cloud Computing, Cryptography, and VANETS.

Citation: *Gurpreet Kaur & Dr. Gagandeep Jagdev (2017). Implementation of DES and AES Cryptographic Algorithms in Accordance with Cloud Computing, International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), 4(4), pp.1-14, DOI: http://dx.doi.org/10.20431/2349-4859.0404001*