Data Encryption Standard using GUI in MATLAB

Project Report

By

17BCE0116 - R. SEETHA RAM REDDY

17BEC0232 - P. MANOJ KUMAR

17BEC0257 - G. YASWANTH

Submitted to

Dr. THANIKAISELVAN V

Associate Professor Sr.

in partial fulfilment of the course of

INFORMATION THEORY AND CODING

ECE4007 - C1 + TC1

ABSTRACT:

Over the past decade, the world's information technology has grown remarkably, and cryptography has improved significantly to safeguard information integrity and confidentiality. Secrecy is the heart of cryptography. Encryption is the process of encoding a message in such a way only authorized parties can read it. Decryption is the process of transforming data that has been rendered unreadable through encryption back to its unencrypted form. The DES algorithm is a 64-bit block cipher with a 56-bit key. This Project describes DES technology for secure data transmission while maintaining the authenticity and integrity of the message. In this, message is encrypted before the data transmission process starts. The decryption and encryption of data is done by using the data encryption standard algorithm in MATLAB using GUI.

INTRODUCTION:

Today, it is widely accepted that the highest priority, security issues already play a central role in the design of future IT systems. Applications and areas that require security include endless Internet communications, vehicle-to-vehicle communications, e-commerce, e-banking, consumer electronic barcodes, and electronic stamps. Therefore, data security is a key aspect of secure data transfer over unreliable network. Traditional encryption methods can only maintain data security. Unauthorized users could access your information for malicious purposes. For security purposes, there is the concept of encryption. Encryption provides security to data by hiding it from unauthorized users. Provides security by giving the concept of encryption and decryption. The process of encoding plaintext into ciphertext is called encryption, and the inverse decoding of ciphertext into plaintext is called decryption. This can be done in two ways: symmetric key cryptography and asymmetric key cryptography.

Symmetric key encryption uses the same key for encryption and decryption. However, asymmetric key encryption uses one key for encryption and another key for decryption. Private key cryptography includes DES, AES, 3DES, IDEA, Blowfish algorithm, etc., and public key cryptography includes RSA, digital signature, etc.

DES was jointly developed by IBM and the US government in 1974 to set a standard that everyone could use to securely communicate with each other. This algorithm was widely available, cheap, very secure and this was used in a wide variety of application.

Historical DES resulted from a project first initiated by IBM in the 60's which resulted in LUCIFER, a block cipher (64 bits) which used a key size of 128 bits. The NSA got involved and the final product, DES, ended with a 56-bit key. One of the greatest worries was key size was just 56 bits and we can send only 64 bits length message (plain text) at once. The modified lucifer algorithm was adopted by NIST as a federal standard on November 23, 1976. Then its name was changed to the Data Encryption Standard (DES). With the official backing of government, it was widely used algorithm in a short span. Since then DES was successfully used up to 1997. Then AES replaced DES with a more secure symmetric key algorithm.

PROPOSED METHOD:

Data encryption standard:

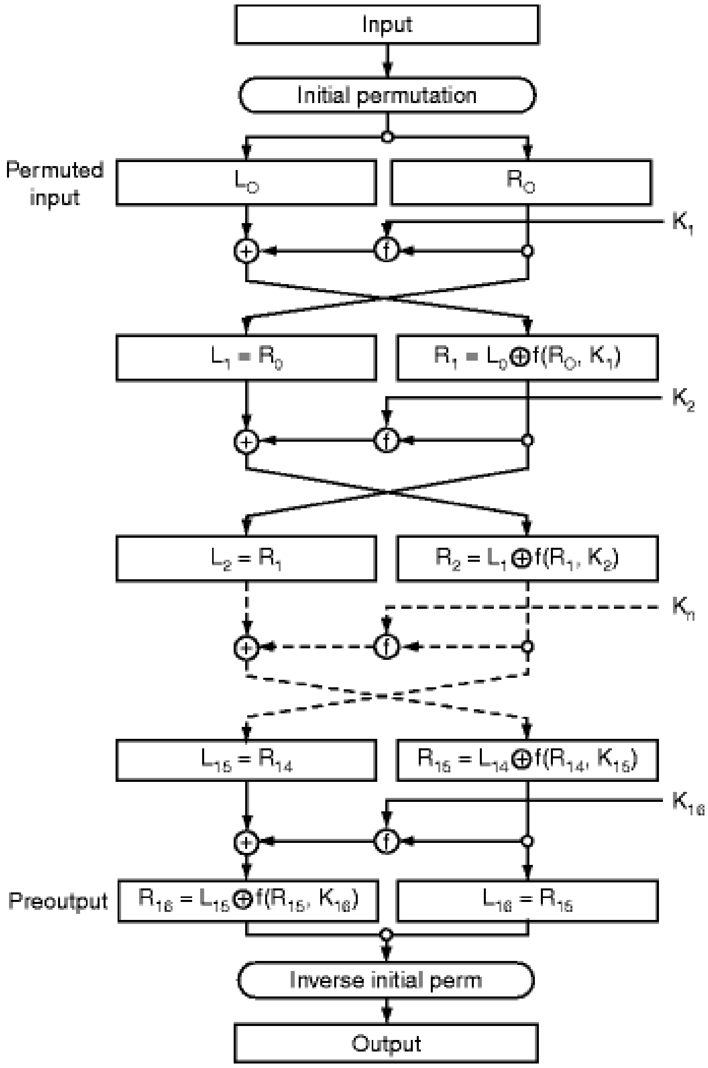

Fig1: Block diagram of DES encryption

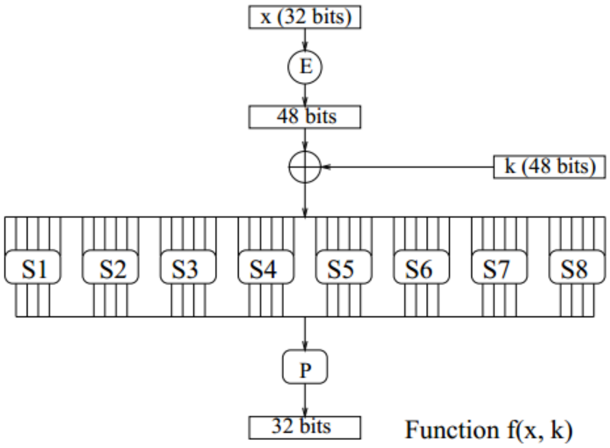

Fig2: Block diagram of DES f function (f)

Explanation of Block diagram of DES Encryption and about function f:

DES is based on two fundamental attributes of cryptography transportation (Diffusion) and Substitution (confusion). DES consists of 16 steps, each of which is called as a Round Algorithm:

1. First step, Input (64- bit plain text) is given to the Initial permutation function (IP).

2. The IP is performed on Input

3. The IP produces two equal halves of permuted block. They are Left plain text (L_0) and Right plain text (R_0) each 32 bits.

4. Now, each L_0 and R_0 go through sixteen rounds of encryption process, each with its own key:

    a. From the 56-bit key a different 48-bit sub key is generated using key scheduling algorithm.

    b. Using the Expansion function, R_0 is expanded from 32 to 48 bits (fig:2)

    c. Now, the resulted 48-bit R_0 is XORed with the 48-bit key.

    d. The resulted 48 bit key in the above step is reduced to 32 bits from 48 bit using S box. (fig:2)

    e. These 32 bits are permutated using Permutation function

    (P- Box) (fig:2)

    f. The output of P Box is XORed with L_0

    g. The result of the previous step will be R_1 and old R_0 is now L_1 (swapping).

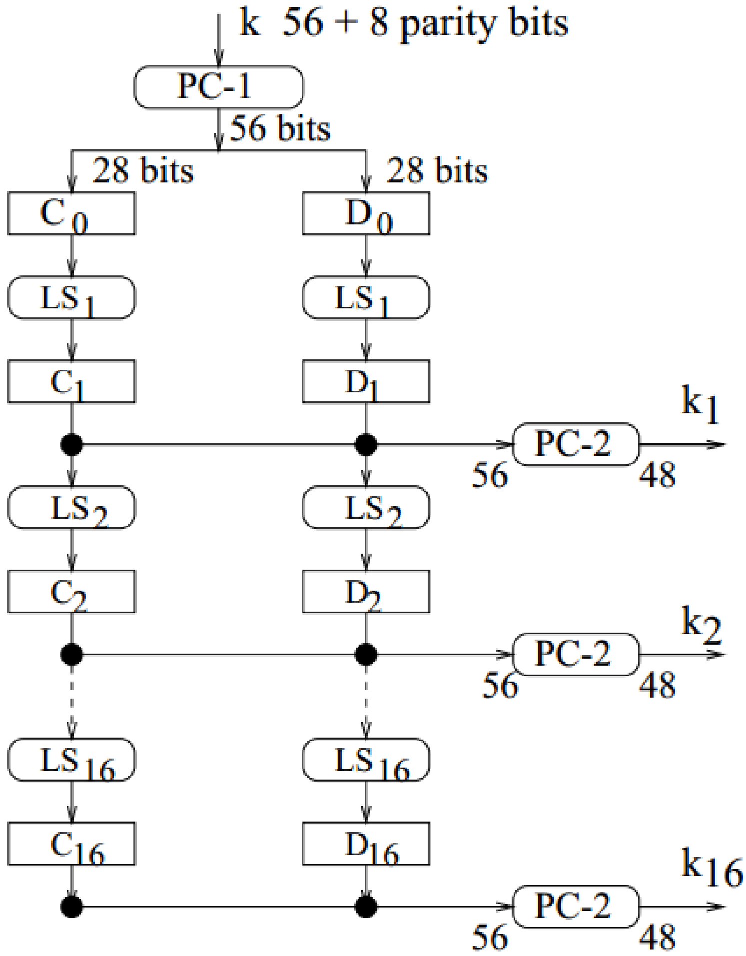

Fig3: Block diagram for key scheduling

Key generation:

1. The system takes 64-bit input key, this will be converted into binary value and then 56-bit key is produced eliminating parity check bits

2. The 56-bit key is given input to the PC-1

3. After permutation the 56 bit is divided into halves C0 and D0

4. Perform left shift to the previous results according to the schedule of left shifts (no. of shifts), to obtain C1 and D1.

5. Then concatenate the C1 and D1 and then give 56-bit input to PC-2 for permutation. The result will be 48 bits which is K1. Use C1 and D1as input for next round to obtain C2 and D2 and so on.

DES Decryption:

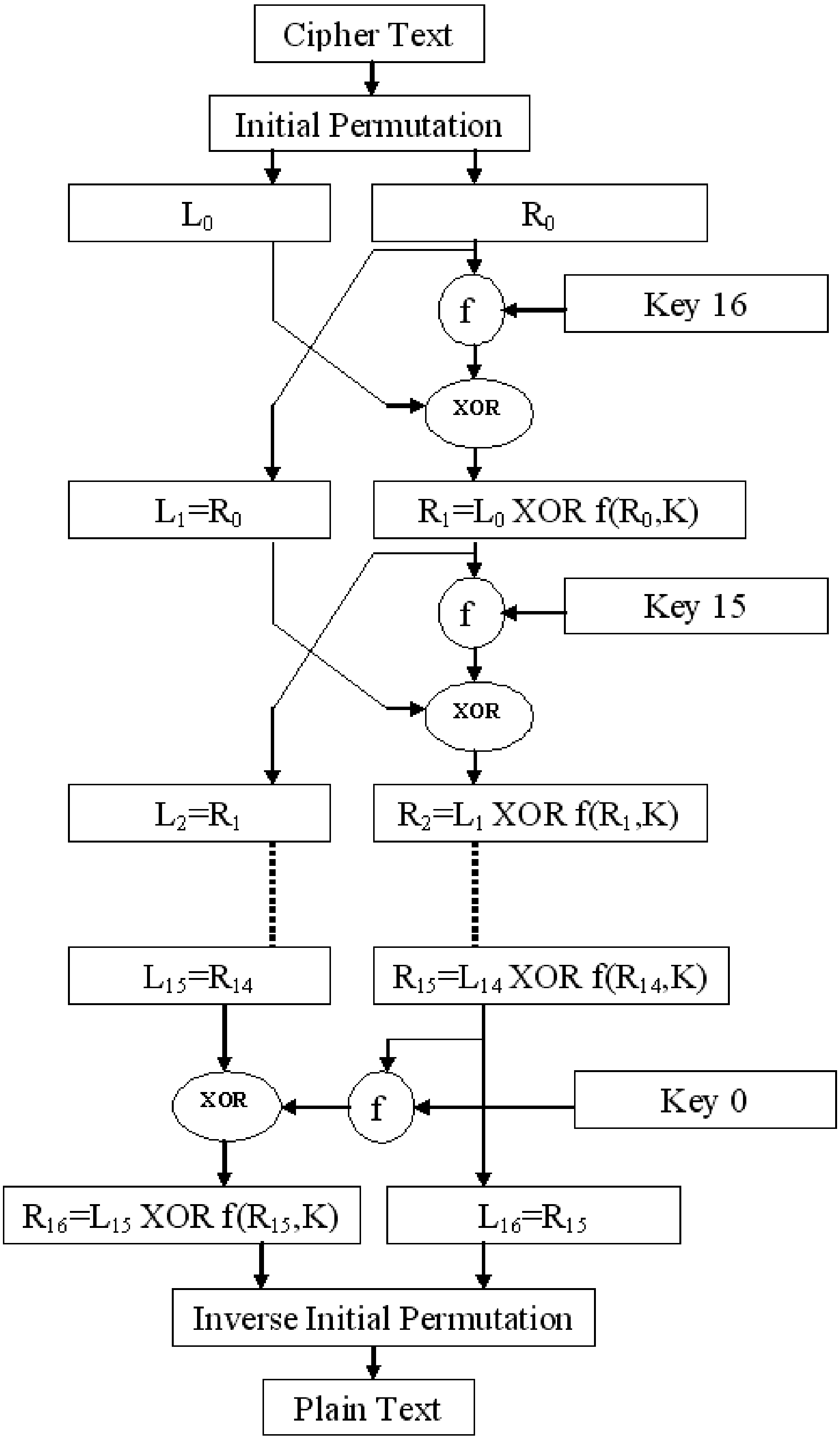

Fig4: Block diagram of DES Decryption

The procedure is same as the encryption but the input text and keys sequence will be different.

Required tables for DES Encryption and Decryption:

Initial Permutation								*Final Permutation*							
58	50	42	34	26	18	10	02	40	08	48	16	56	24	64	32
60	52	44	36	28	20	12	04	39	07	47	15	55	23	63	31
62	54	46	38	30	22	14	06	38	06	46	14	54	22	62	30
64	56	48	40	32	24	16	08	37	05	45	13	53	21	61	29
57	49	41	33	25	17	09	01	36	04	44	12	52	20	60	28
59	51	43	35	27	19	11	03	35	03	43	11	51	19	59	27
61	53	45	37	29	21	13	05	34	02	42	10	50	18	58	26
63	55	47	39	31	23	15	07	33	01	41	09	49	17	57	25

Fig5: initial and final (Inverse) permutation table

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

Fig6: Expansion permutation table

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Fig7: permutation in f table

S1	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
p_0	e	4	d	1	2	f	b	8	3	a	6	c	5	9	0	7
p_1	0	f	7	4	e	2	d	1	a	6	c	b	9	5	3	8
p_2	4	1	e	8	d	6	2	b	f	c	9	7	3	a	5	0
p_3	f	c	8	2	4	9	1	7	5	b	3	e	a	0	6	d

S2	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
p_0	f	1	8	e	6	b	3	4	9	7	2	d	c	0	5	a
p_1	3	d	4	7	f	2	8	e	c	0	1	a	6	9	b	5
p_2	0	e	7	b	a	4	d	1	5	8	c	6	9	3	2	f
p_3	d	8	a	1	3	f	4	2	b	6	7	c	0	5	e	9

S3	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
p_0	a	0	9	e	6	3	f	5	1	d	c	7	b	4	2	8
p_1	d	7	0	9	3	4	6	a	2	8	5	e	c	b	f	1
p_2	d	6	4	9	8	f	3	0	b	1	2	c	5	a	e	7
p_3	1	a	d	0	6	9	8	7	4	f	e	3	b	5	2	c

S4	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
p_0	7	d	e	3	0	6	9	a	1	2	8	5	b	c	4	f
p_1	d	8	b	5	6	f	0	3	4	7	2	c	1	a	e	9
p_2	a	6	9	0	c	b	7	d	f	1	3	e	5	2	8	4
p_3	3	f	0	6	a	1	d	8	9	4	5	b	c	7	2	e

S5	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
p_0	2	c	4	1	7	a	b	6	8	5	3	f	d	0	e	9
p_1	e	b	2	c	4	7	d	1	5	0	f	a	3	9	8	6
p_2	4	2	1	b	a	d	7	8	f	9	c	5	6	3	0	e
p_3	b	8	c	7	1	e	2	d	6	f	0	9	a	4	5	3

S6	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
p_0	c	1	a	f	9	2	6	8	0	d	3	4	e	7	5	b
p_1	a	f	4	2	7	c	9	5	6	1	d	e	0	b	3	8
p_2	9	e	f	5	2	8	c	3	7	0	4	a	1	d	b	6
p_3	4	3	2	c	9	5	f	a	b	e	1	7	6	0	8	d

S7	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
p_0	4	b	2	e	f	0	8	d	3	c	9	7	5	a	6	1
p_1	d	0	b	7	4	9	1	a	e	3	5	c	2	f	8	6
p_2	1	4	b	d	c	3	7	e	a	f	6	8	0	5	9	2
p_3	6	b	d	8	1	4	a	7	9	5	0	f	e	2	3	c

S8	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
p_0	d	2	8	4	6	f	b	1	a	9	3	e	5	0	c	7
p_1	1	f	d	8	a	3	7	4	c	5	6	b	0	e	9	2
p_2	7	b	4	1	9	c	e	2	0	6	a	d	f	3	5	8
p_3	2	1	e	7	4	a	8	d	f	c	9	0	3	5	6	b

Fig8: S-box table (x5 and x0 row)

(a) Input Key

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

(b) Permuted Choice One (PC-1)

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

(c) Permuted Choice Two (PC-2)

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

(d) Schedule of Left Shifts

Round number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Fig9: DES Key scheduling calculations

RESULTS: DES ENCRYPTION

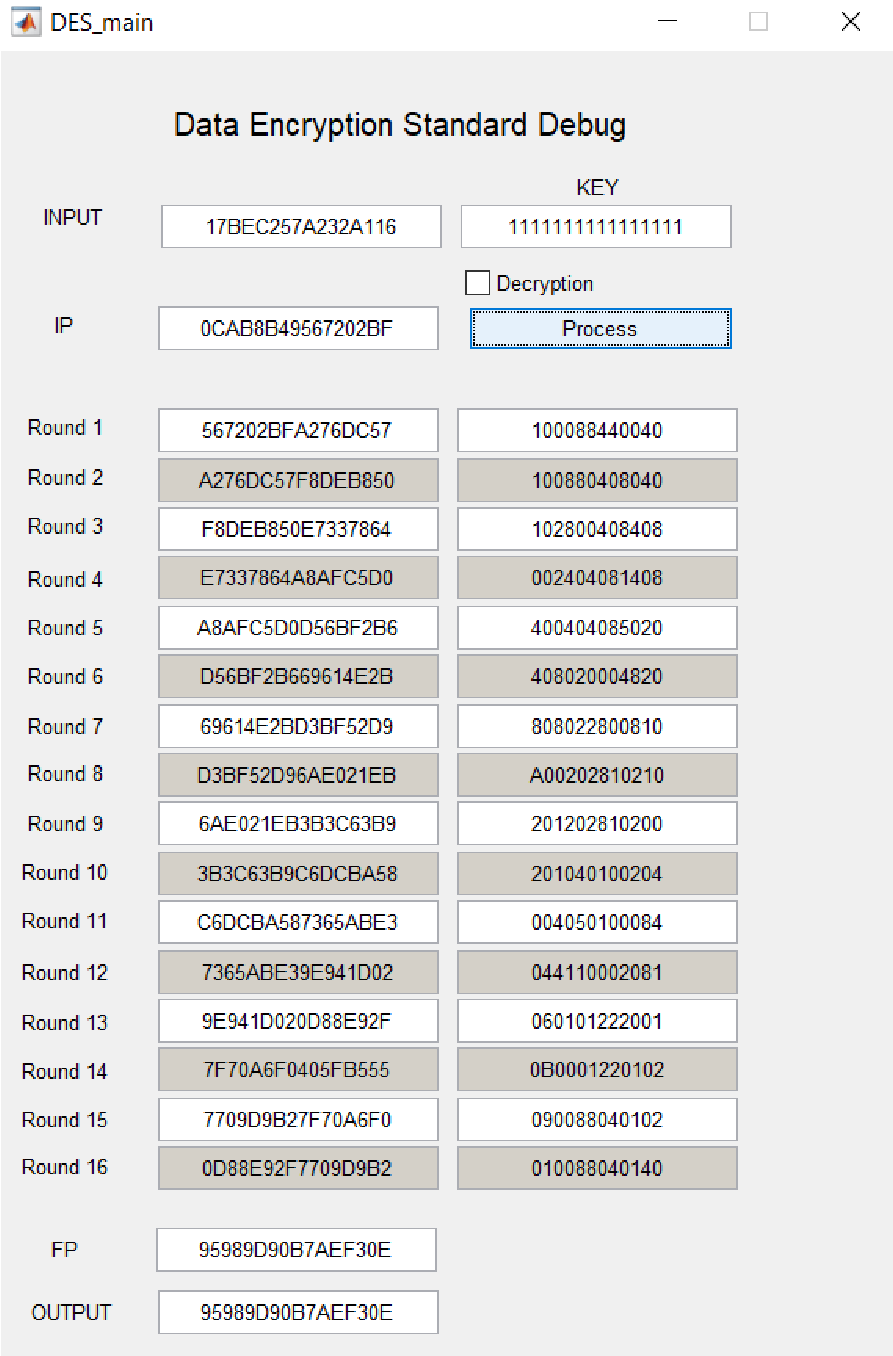

Fig 10: Result for DES Encryption when given registration numbers as input (“A” is and)

Input: Plain text(64-bits) Key: Input key(64-bits)

IP: Result after Initial permutation FP: Final or Inverse permutation

Round: Left box: L_{i-1} and R_{i-1} Right: key i (i=round number)

Output: cipher text

DES DECRYPTION:

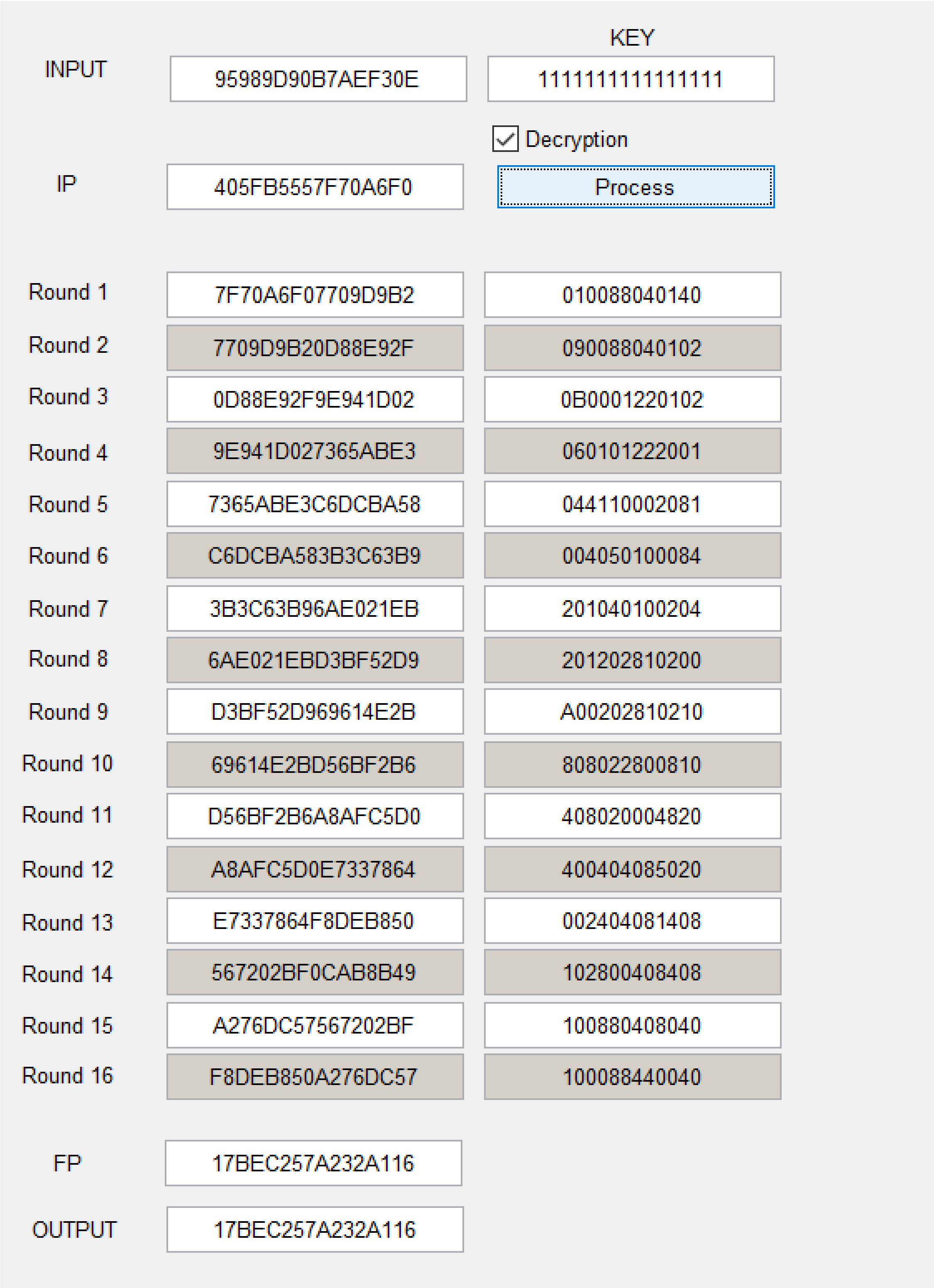

Fig 11: Result for DES Decryption when given Encryption output

we got our registration number as DES Decryption output, which we have given as input for DES Encryption (A= “and” in output)

CONCLUSION:

Providing a secure mechanism for data transmission is very important, as we are moving towards a society where automated information resources are highly used. This project shows how we can encrypt and Decrypt a plain text using DES in GUI. But DES is currently considered an insecure encryption method in some applications, such as banking systems. There are some findings that show the theoretical weaknesses in cipher. so, it is very important to augment this algorithm by adding a new level of security to it. In the future, we can change this algorithm by changing the function implementation, S-box design, and replacing the old XOR with new operations.

Google drive link for MATLAB Files:

https://drive.google.com/open?id=1FbxzTRmX5bY73BNfcUNgbhjH3r8bIpNY

REFERENCES:

- Kefa Rabah, "Theory and Implementation of Data Encryption Standard: A Review", Information Technology Journal, April 2005
- Seung-Jo Han, Heang-Soo Oh and Jongan Park, "The improved data encryption standard (DES) algorithm," Proceedings of ISSSTA'95 International Symposium on Spread Spectrum Techniques and Applications, Mainz, Germany, 1996, pp. 1310-1314 vol.3.
- Yue Wu (2020). Data Encryption Standard (DES) (https://www.mathworks.com/matlabcentral/fileexchange/37847-data-encryption-standard-des), MATLAB Central File Exchange. Retrieved January 7, 2020.