Graphical user interface based on AES algorithm

ASHISH KUMAR KENDHE, HIMANI AGRAWAL

Abstract— This paper presents the graphical user interface based on AES algorithm used for encryption and decryption purpose of messages. Here the input is taken in the form text,image and audio formats in a single graphical user interface .In this we develop the graphical user interface for text ,image and audio file and after that certain parameter like elapsed time are calculated with it's plot

***Index Terms*— Cryptography,Encryption,Decryption ,Ciphertext,AES,Cryptanalysis,Matlab.**

I. INTRODUCTION

I.1Cryptography

Cryptography or cryptology comes from Greek κρυπτός, "hidden secret"; and γράφειν, graphein, "writing", or -λογία, -logia, "study", respectively[1] is the practice and study of techniques for secure communication in the presence of third parties (called adversaries).[2] More generally, it is about constructing and analyzing protocols that overcome the influence of adversaries[3] and which are related to various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation.[4] Modern cryptography intersects the disciplines of mathematics, computer science, and electrical engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce.

Modern cryptography is heavily based on mathematical theory and computer science practice; cryptographic algorithms[5] are designed around computational hardness assumptions, making such algorithms hard to break in practice by any adversary. It is theoretically possible to break such a system but it is infeasible to do so by any known practical means. These schemes are therefore termed computationally secure; theoretical advances, e.g., improvements in integer factorization algorithms, and faster computing technology require these solutions to be continually adapted. There exist information-theoretically secure schemes that provably cannot be broken even with unlimited computing power—an example is the one-time pad—but these schemes are more difficult to implement than the best theoretically breakable but computationally secure mechanisms.

Cryptology-related technology has raised a number of legal issues. In the United Kingdom, additions to the Regulation of Investigatory Powers Act 2000 require a suspected criminal to hand over his or her decryption key if asked by law enforcement. Otherwise the user will face a criminal charge. The Electronic Frontier Foundation (EFI) was involved in a case which questioned whether requiring suspected criminals to provide their decryption keys to law enforcement is unconstitutional. The EFI argued that this is a violation of the right of not being forced to incriminate oneself, as given in the fifth amendment.

Manuscript received March 23, 2015

ASHISH KUMAR KENDHE , ME ELECTRONICS AND TELECOMMUNICATION ,ELECTRONICS&TELECOMMUNICATION DEPT,CSVTU/SSCET,BHILAI,CHHATTISGARH

HIMANIAGRAWAL, ASSOCIATE PROFESSOR ,ELECTRONICS & TELECOMMUNICATION DEPT,CSVTU,SSCET, BHILAI ,CHHATTISGARH

I.2 Terminology

Until modern times cryptography referred almost exclusively to encryption, which is the process of converting ordinary information (called plaintext) into unintelligible text (called ciphertext). Decryption is the reverse, in other words, moving from the unintelligible ciphertext back to plaintext. A cipher is a pair of algorithms that create the encryption and the reversing decryption. The detailed operation of a cipher is controlled both by the algorithm and in each instance by a "key". This is a secret (ideally known only to the communicants), usually a short string of characters, which is needed to decrypt the ciphertext. A "cryptosystem" is the ordered list of elements of finite possible plaintexts, finite possible cyphertexts, finite possible keys, and the encryption and decryption algorithms which correspond to each key. Keys are important, as ciphers without variable keys can be trivially broken with only the knowledge of the cipher used and are therefore useless (or even counter-productive) for most purposes. Historically, ciphers were often used directly for encryption or decryption without additional procedures such as authentication or integrity checks.

In general use, the term "code" is often used to mean any method of encryption or concealment of meaning. However, in cryptography, code has a more specific meaning. It means the replacement of a unit of plaintext (i.e., a meaningful word or phrase) with a code word (for example, wallaby replaces attack at dawn). Codes are no longer used in serious cryptography—except incidentally for such things as unit designations (e.g., Bronco Flight or Operation Overlord)—since properly chosen ciphers are both more practical and more secure than even the best codes and also are better adapted to computers.

Cryptanalysis is the term used for the study of methods for obtaining the meaning of encrypted information without access to the key normally required to do so; i.e., it is the study of how to crack encryption algorithms or their implementations.

II. PROPOSED ALGORITHM & RELATED WORK

II.1 Advanced encryption standard

The Rijndael proposal for AES[6][7] defines a cipher in which he block length and key length can be independently specified to be 128,192 or 256 bits.The AES specification uses the same three key size alternatives but limits the block length to 128 bits.The AES structure is divided into four different stages,one of permutation and three of substitution.They are,

1. Substitution bytes: Uses an s-box to perform a byte –to –byte substitution of block .
2. Shift rows:A simple permutation .
3. Mix columns: A substitution that uses of arithmetic over GF(2^8).
4. Add round key:A simple bitwise XOR of the current block with a portion of the expanded keys.

A data block to be encrypted by AES is split into an array of bytes,and each operation is byte oriented.AES's round function consist of four layers.In the first layer,an 8x8 S-box is copied to each byte .The second and the third layers are linear mixing layers in which rows of array are shifted ,and columns are mixed.In the foruth layes,sub key bytes are XORed into each byte of array .In the last round ,the column mixing is omitted .so the algorithm consist of four main steps :a substitution step ,a shift row step,a mix column step and a sub key addition step.The substitution step consist of S-boxes.The shiftrow steps consist of cyclic –shifting of the bytes within the rows.The key addition is a straight forward XOR operation between the data and the key .
The AES cipher is described as a pseudo code in Figure.II. 1 As shown in the pseudo code, all the N_r rounds are identical with the exception of the final round which does not include the MixColumns transformation. The array w[] represents the round keys that are generated by the key expansion routine. In the following sections, individual transformations that are used in each encryption round are described.

```
Cipher(byte PlainText[4*Nb], byte CipherText[4*Nb],
word w[Nb*(Nr+1)])
begin
  byte state[4,Nb]

  state = in

  AddRoundKey(state, w[0, Nb-1])

  for round = 1 step 1 to Nr–1
    SubBytes(state)
    ShiftRows(state)
    MixColumns(state)
    AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
  end for

SubBytes(state)
ShiftRows(state)
AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])

out = state

end
```

Figure II. 1. AES Cipher

II.1.1 Byte substitution

In sub bytes step ,each byte in the state matrix is replaced with a sub byte using a 8-bit substitution box.,the Rijndael S-box.This operation provides the non linearity in the cipher .the s-box used is derived from the multiplicative inverse over GF(2^8),known to have good non –linearity properties.To avoid attacks based on simple algebraic properties ,the s-box is constructed by combining the reverese function with an invertible affine transformation,The s-box is also chosen to avoid any fixed points(and so derangement),and also any opposite fixed points.

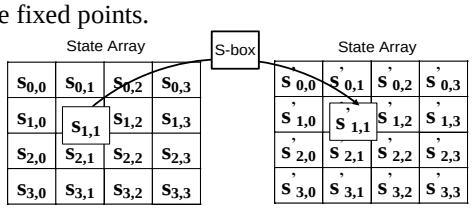

Figure II.2. SubBytes Transformation

The invertible S-box table is constructed by performing the following transformation on each byte of the State.
Take the multiplicative inverse in the finite field GF(2^8) of the byte.
Apply the following transformation to the byte:

$$b_i' = b_i \oplus b_{(i+4)\bmod 8} \oplus b_{(i+5)\bmod 8} \oplus b_{(i+6)\bmod 8} \oplus b_{(i+7)\bmod 8} \oplus c_i \qquad (1)$$

The b_i is the i^{th} bit of the byte and c_i is the i^{th} bit of a constant byte with the value of {63}. The combination of the two transformations can be expressed in matrix form as shown below:

$$\begin{bmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

The S-box table shown in Table.1. is constructed by performing the two transformations described earlier for all possible values of a byte, ranging from {00} to {ff}. For example the substitution value for {53} would be determined by the intersection of the row with index '5' and the column with index '3'.

		Y															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**a**	**b**	**c**	**d**	**e**	**f**
X	**0**	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	A	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	B	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	C	b	7	2	2	1	a	b	c	e	d	7	1	4	b	8	8

		a	8	5	e	c	6	4	6	8	d	4	f	b	d	b	a
	D	7 0	3 e	b 5	6 6	4 8	0 3	f 6	0 e	6 1	3 5	5 7	b 9	8 6	c 1	1 d	9 e
	E	e 1	f 8	9 8	1 1	6 9	d 9	8 e	9 4	9 b	1 e	8 7	e 9	c e	5 5	2 8	d f
	F	8 c	a 1	8 9	0 d	b f	e 6	4 2	6 8	4 1	9 9	2 d	0 f	b 0	5 4	b b	1 6

Table .1.AES S-box

II.1.2 ShiftRows ( ) Transformation

The ShiftRows transformation cyclically shifts the last three rows of the state by different offsets. The first row is left unchanged in this transformation. Each byte of the second row is shifted one position to the left. The third and fourth rows are shifted left by two and three positions, respectively. The ShiftRows transformation is illustrated in Figure II .3.

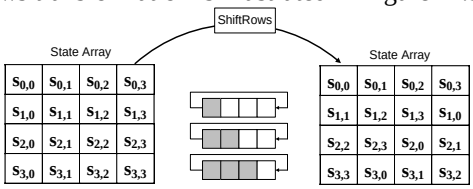

Figure II .3. ShiftRows Transformation

II.1.3 MixColumns ( ) Transformation

This transformation operates on the columns of the State, treating each columns as a four term polynomial the finite field GF(2^8). Each columns is multiplied modulo x^4+1 with a fixed four-term polynomial $a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$ over the GF(2^8). The MixColumns transformation can be expressed as a matrix multiplication as shown below:

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{0,c} \\ s_{0,c} \\ s_{0,c} \end{bmatrix}$$

The MixColumns transformation replaces the four bytes of the processed column with the following values: $s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$ (2)

$$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c} \quad (3)$$

$$s'_{0,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c}) \quad (4)$$

$$s'_{1,c} = (\{03\} \bullet s_{0,c} \oplus s_{1,c}) \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c}) \quad (5)$$

The " $\bullet$ " corresponds to the multiplication of polynomials in GF(2^8) modulo an irreducible polynomial of degree 8. A polynomial is irreducible if its only divisors are one and itself. For the AES algorithm the irreducible polynomial is:
$m(x) = x^8 + x^4 + x^3 + x + 1$.

The MixColumns transformation is illustrated in Figure II .4. This transformation together with ShiftRows, provide substantial diffusion in the cipher meaning that the result of the cipher depends on the cipher inputs in a very complex way. In other words, in a cipher with a good diffusion, a single bit change in the plaintext will completely change the ciphertext in an unpredictable manner.

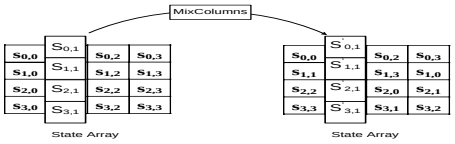

Figure II .4. MixColumns Transformation

II.1.4 AddRoundKey ( ) Transformation

During the AddRoundKey transformation, the round key values are added to the State by means of a simple Exclusive Or (XOR) operation. Each round key consists of N_b words that are generated from the KeyExpansion routine. The round key values are added to the columns of the state in the following way:

$$\left[s'_{0,c}, s'_{1,c}, s'_{2,c}, s'_{3,c}\right] = \left[s_{0,c}, s_{1,c}, s_{2,c}, s_{3,c}\right] \oplus \left[w_{round*Nb+c}\right] \quad \text{for}$$

$$0 \le c < N_b \quad (6)$$

In the equation above, the round value is between $0 \le round \le N_r$. When round=0, the cipher key itself is used as the round key and it corresponds to the initial AddRoundKey transformation displayed in the pseudo code in Figure II. 1..

The AddRoundKey transformation is illustrated in Figure II .5.

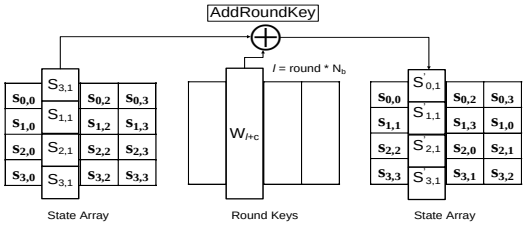

Figure II .5. AddRoundKey Transformation

II.1.5 steps involved

1. Key Expansion[8]—round keys are derived from the cipher key using Rijndael's key schedule. AES requires a separate 128-bit round key block for each round plus one more.
2. Initial Round

Add Round Key—each byte of the state is combined with a block of the round key using bitwise xor.

3. Rounds

1) SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.
2) ShiftRows—a transposition step where each row of the state is shifted cyclically a certain number of steps.
3) MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.
4) AddRoundKey

4. Final Round (no MixColumns)

A. SubBytes
B. ShiftRows
C. AddRoundKey.

The SubBytes step

In the SubBytes step, each byte in the state is replaced with its entry in a fixed 8-bit lookup table, $S(b_{i,j}) = S(a_{i,j})$.In the SubBytes step, each byte $a_{i,j}$ in the state matrix is replaced with a SubByte $S(a_{i,j})$using an 8-bit substitution box, the Rijndael S-box. This operation provides the non-linearity in the cipher. The S-box used is derived from the multiplicative inverse over **GF**(2^8), known to have good non-linearity properties. To avoid attacks based on simple algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The S-box is also chosen to avoid any fixed points (and so is a derangement), i.e., $S(a_{i,j}) \neq a_{i,j}$, and also any opposite fixed points, i.e., $S(a_{i,j}) \oplus a_{i,j} \neq 0\text{xFF}$.

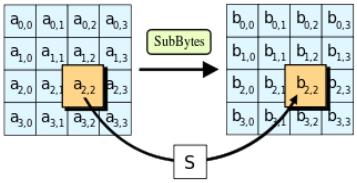

Figure II.6. *The Subbytes Step*

The Shift Rows step

In the Shift Rows step, bytes in each row of the state are shifted cyclically to the left. The number of places each byte is shifted differs for each row.

The ShiftRows step operates on the rows of the state; it cyclically shifts the bytes in each row by a certain offset. For AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. For blocks of sizes 128 bits and 192 bits, the shifting pattern is the same. Row n is shifted left circular by n-1 bytes. In this way, each column of the output state of the ShiftRows step is composed of bytes from each column of the input state. (Rijndael variants with a larger block size have slightly different offsets). For a 256-bit block, the first row is unchanged and the shifting for the second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively—this change only applies for the Rijndael cipher when used with a 256-bit block, as AES does not use 256-bit blocks. The importance of this step is to make columns not linear independent If so, AES becomes four independent block ciphers.

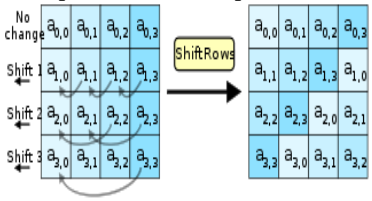

Figure II.7. *The Shift Rows step*

The Mixcolumn step

In the MixColumns step, each column of the state is multiplied with a fixed polynomial c(x).In the MixColumns step, the four bytes of each column of the state are combined using an invertible linear transformation. The MixColumns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes. Together with ShiftRows, MixColumns provides diffusion in the cipher.During this operation, each column is multiplied by the known matrix that for the 128-bit key is:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

The multiplication operation is defined as: multiplication by 1 means no change, multiplication by 2 means shifting to the left, and multiplication by 3 means shifting to the left and then performing XOR with the initial unshifted value. After shifting, a conditional XOR with 0x1B should be performed if the shifted value is larger than 0xFF.

In more general sense, each column is treated as a polynomial over GF(2^8) and is then multiplied modulo x^4+1 with a fixed polynomial $c(x) = 0x03 \cdot x^3 + x^2 + x + 0x02$. The coefficients are displayed in their hexadecimal equivalent of the binary representation of bit polynomials from GF(2)[x]. The MixColumns step can also be viewed as a multiplication by a particular MDS matrix in a finite field. This process is described further in the article Rijndael mix columns.

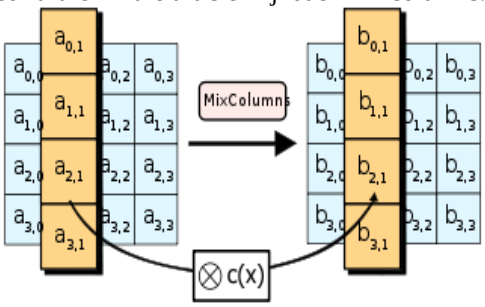

Figure II.8. *Mixcolumn step*

The AddRoundKey step

In the AddRoundKey step, each byte of the state is combined with a byte of the round subkey using the XOR operation (⊕). In the AddRoundKey step, the subkey is combined with the state. For each round, a subkey is derived from the main key using Rijndael's key schedule; each subkey is the same size as the state. The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR.

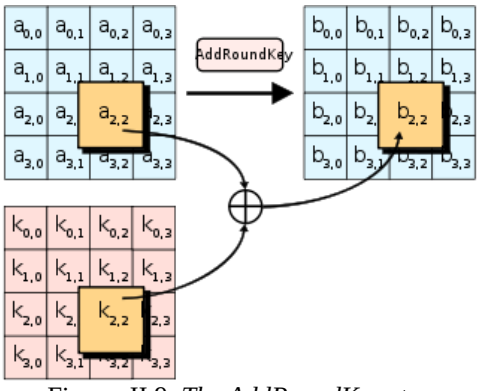

Figure .II.9. *The AddRoundKey step*

III. EXPERIMENTAL RESULTS

The algorithm has been implemented in matlab R2009a in windows environment with a system configuration of intel i5 processor with 4GB RAM .The graphical user interface based on AES algorithm for communication is shown in following steps

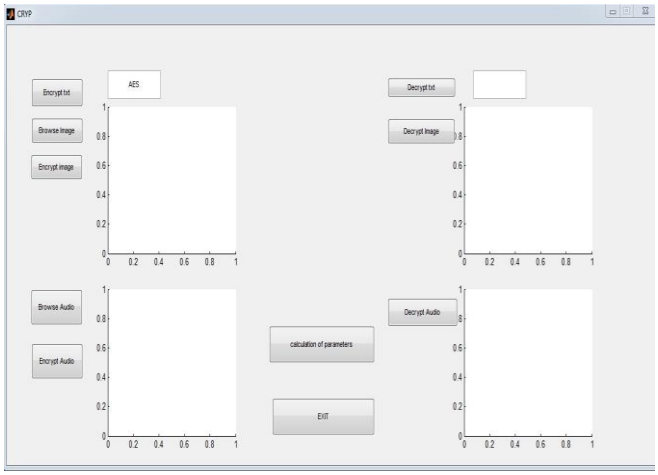

Figure .III.1Common Graphical User Interface

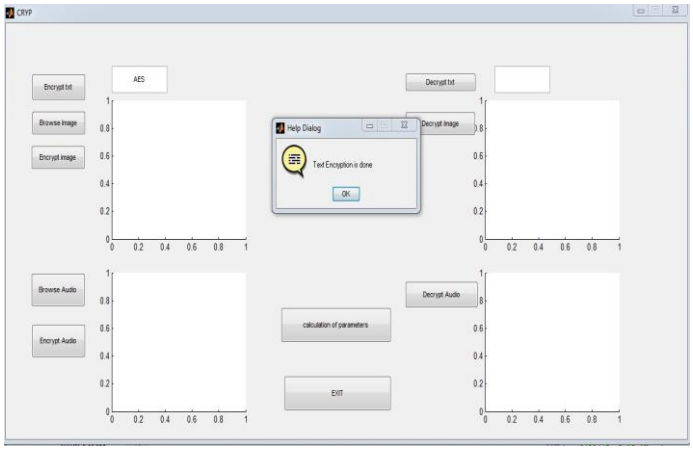

Figure .III.2 Encrypted Text.

Figure .III.3 Decrypted Text.

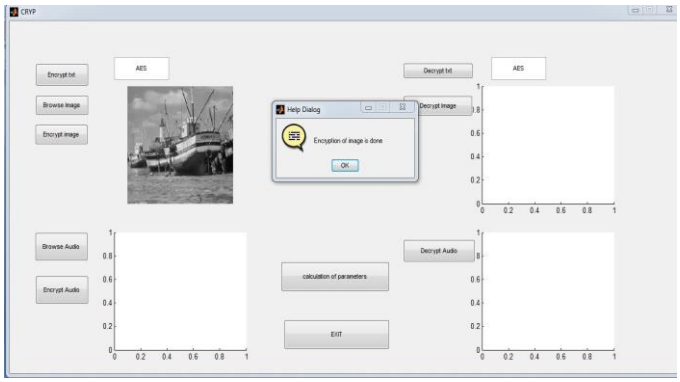

Figure .III.4 Encrypted Image

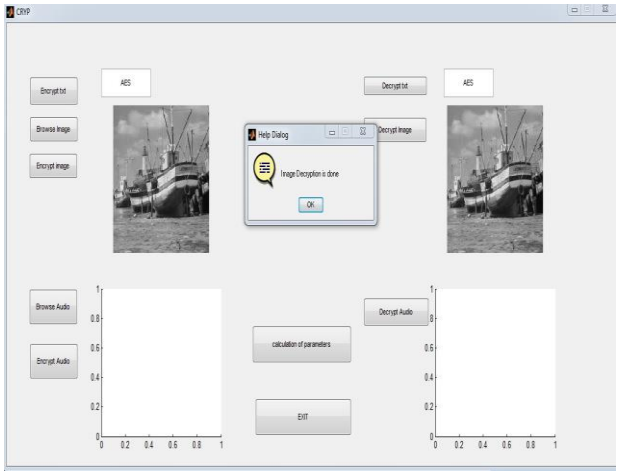

Figure .III.5 Decrypted Image

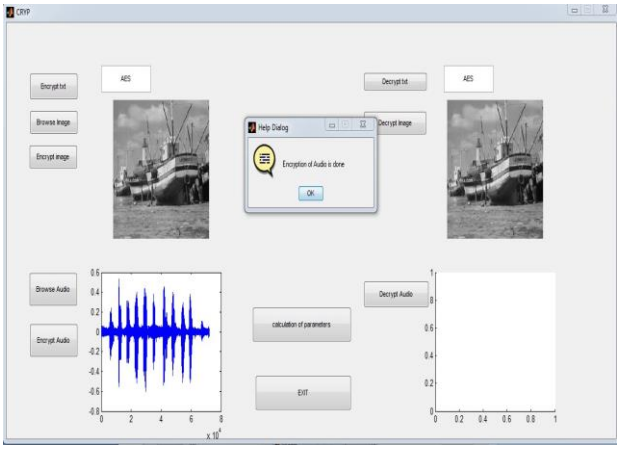

Figure .III.6Encrypted Audio

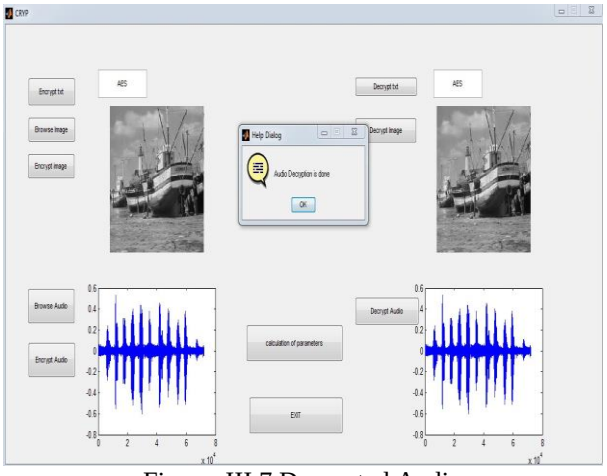

Figure .III.7 Decrypted Audio

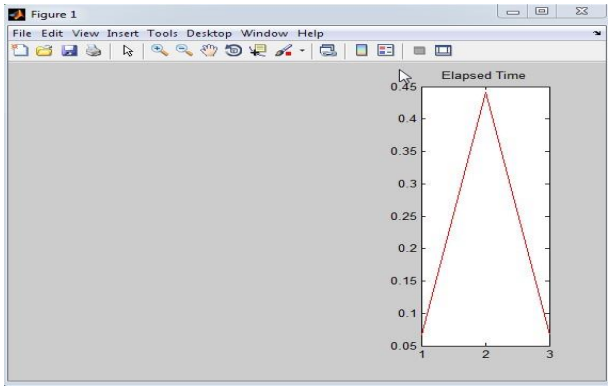

Figure .III.8 Plot For Elapsed Time.

Table.2. Calculation For Elapsed Time

PARAMETER	Elapsed Time
Text	0.07
Audio	0.44
Image	0.07

CONCLUSION& FUTURE SCOPE

This research paper gives us the graphical user interface based on AES algorithm for encryption and decryption of text ,image and audio formats of data and also at the same time the values of Elapsed time has been calculated and shown in the Table .2.This method can be also be used for analysis of certain more parameters such as psnr ,mse etc in future with some more implementations in AES algorithm.

ACKNOWLEDGEMENT

The author would like to thank the anonymous reviewers & the faculties who supported me in this research paper for their valuable comments and suggestions .

REFERENCES

[1] William Stalling "Network Security Essentials (Applications and Standards)", Pearson Education, 2004.
[2] Atul Kahate (2009) ," Cryptography and Network Security", 2nd edition, McGraw-Hill.
[3] Stallings (1999), "Cryptography and Network Security", 2nd edition, Prentice Hall.
[4] William Stallings (2003), "Cryptography and Network Security", 3rd edition, Pearson Education.
[5] Christopher Swenson," Modern Cryptanalysis Techniques For Advanced Code Breaking", wiley publication Inc.2008.
[6] Alireza Sharifi,Hadi soleimany and Mohammadreza Aref, "9 Round Attack on AES-256 by a 6-Round Property", IEEE Proceedings of ICEE2010,May 11-13,2010.
[7] S Shivkumar and Dr.Umamaheshwari,"Performance Comparison Of Advanced Encryption Standard (AES) and AES Key dependent S-box-simulation using MATLAB" ,IEEE 2011.
[8] B.subramanyan,Vivek.M.Chhabaria &T.G.Sankar babu, "Image Encryption Based On AES Key Expansion",IEEE Proceedings of ICEAIT2011.