Global Journal of Computer Science and Technology: E
Network, Web & Security
Volume 15 Issue 4 Version 1.0 Year 2015
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Implementation of AES with Time Complexity Measurement for Various Input

By Shraddha More & Rajesh Bansode
Mumbai university, India

Abstract- Network Security has a major role in the development of data communication system, where more randomization in the secret keys increases the security as well as the complexity of the cryptography algorithms. In the recent years network security has become an important issue. Cryptography has come up as a solution which plays a vital role in the information security system against various attacks. This security mechanism uses the AES algorithm to scramble data into unreadable text which can only be decrypted with the associated key. The AES algorithm is limited only for text as an input. It also has, the more time complexity. So it suffers from vulnerabilities associated with another type of input and time constraints. So its challenge to implement the AES algorithm for various types of input and require less decryption time. The propose work demonstrate implementation of a 128-bit Advanced Encryption Standard (AES), which consists of both symmetric key encryption and decryption algorithms for input as a text, image and audio. It also gives less time complexity as compared to existing one. At the last stage comparing the time complexity for encryption and decryption process for all three types of input. This paper also demonstrates a side channel attack on the standard software implementation of the AES cryptographic algorithm.de

Keywords: side channel attack, aes, des,rsa, encryption, decryption, cryptography, network security..

GJCST-E Classification : F.1.3 C.2.1

Strictly as per the compliance and regulations of:

Implementation of AES with Time Complexity Measurement for Various Input

Shraddha More [α] & Rajesh Bansode [ρ]

Abstract- Network Security has a major role in the development of data communication system, where more randomization in the secret keys increases the security as well as the complexity of the cryptography algorithms. In the recent years network security has become an important issue. Cryptography has come up as a solution which plays a vital role in the information security system against various attacks. This security mechanism uses the AES algorithm to scramble data into unreadable text which can only be decrypted with the associated key. The AES algorithm is limited only for text as an input. It also has, the more time complexity. So it suffers from vulnerabilities associated with another type of input and time constraints. So its challenge to implement the AES algorithm for various types of input and require less decryption time. The propose work demonstrate implementation of a 128-bit Advanced Encryption Standard (AES), which consists of both symmetric key encryption and decryption algorithms for input as a text, image and audio. It also gives less time complexity as compared to existing one. At the last stage comparing the time complexity for encryption and decryption process for all three types of input. This paper also demonstrates a side channel attack on the standard software implementation of the AES cryptographic algorithm.de

Keywords: side channel attack, aes, des,rsa, encryption, decryption, cryptography, network security.

I. Introduction

Cryptography plays an important role in the security of data transmission. Data Security is a challenging concern of data communications that focuses on many areas including secure communication channel and strong data encryption technique. The secure transmission of confidential data enclosed gets a great deal of attention because of the rapid development in information technology. The predictable methods of encryption can only maintain the data security. The development of computing technology imposes stronger requirements on the cryptography schemes. The rapidly growing number of wireless communication users has led to the increasing demand for security measures and devices to protect user data transmitted over wireless channels[1].

Two types of cryptographic systems have been developed for that purpose symmetric (secret key) and asymmetric (public key) cryptosystems. Symmetric cryptography, such as in the Data Encryption Standard (DES), 3DES, and Advanced Encryption Standard (AES) uses an identical key of the sender to encrypt the message text and receiver to decrypt the encrypted text. Asymmetric cryptography, such as the Rivest-Shamir-Adleman (RSA) uses different public keys for encryption and decryption, eliminating the key exchange problem.[2] Symmetric cryptography is more suitable for the encryption of a large amount of data. The Data Encryption Standard (DES) has been used by the U.S. government standard since 1977. However, now, it can be cracked quickly and inexpensively. The AES algorithm defined by the National Institute of Standards and Technology (NIST) of the United States has widely accepted to replace DES as the new symmetric encryption algorithm [3]. This above cryptographic algorithms are not more secure. To overcome the vulnerabilities in network security in 2000, the Advanced Encryption Standard (AES) replaced the DES to meet the ever-increasing requirements for security. In cryptography, the AES, also called as Rijndael, is a block cipher adopted as an encryption standard by the US government, which specifies an encryption algorithm capable of protecting sensitive information[4]. The Rijndael algorithm is a symmetric block cipher that can encrypt and decrypt information. Encryption converts data into an unintelligible form called cipher-text. Decryption of the cipher-text converts the data back into its unique form which is called plaintext. The AES algorithm supports 128, 192 and 256 bit key length to encrypt and decrypt data in blocks of 128 bits , hence the name AES-128, AES-192 and AES-256 respectively[5]. The hardware implementation of the AES algorithm can provide high performance, low cost for specific applications and trustworthiness compared to its software counterparts[6].

The organization of the paper is as follows, Section II describes the design overview of AES algorithm for both encryption and decryption. Section III presents implementation Details, Section IV is discussed on Experimental Results. Section V projects on future scope and conclusion.

II. Design Overview of AES

AES is a symmetric block cipher with block length of 128 bits. It allows three different key lengths 128,192 and 256 bits. In encryption process processing of 128 bit keys required for 10 rounds, 192 bit keys

Author α : Master of engineering in Information technology, TCET, Mumbai university, India. e-mail: moreshraddha30@gmail.com
Author ρ : Associate professor in Department of Information technology, TCET, Mumbai university, India.
e-mail: rajesh.bansode@thakureducation.org

required for 12 rounds and 256 bit keys required for 14 rounds which is shown in table1. AES is a round based algorithm. For encryption and decryption each round has four functions excepting last round. Last round required three functions. The encryption algorithm has four round functions SubByte( ), ShiftRows( ), MixColumn( )and AddRoundKey( ). The decryption, also has the same number of rounds with reverse transformation, order of round function is different i.e. InvShiftRow( ), InvSubByte( ), AddRoundKey( ) and InvMixColumn( ) [2]-[3].

Table 1 : AES parameters for the various AES versions

AES PARAMETERS	AES-128	AES-192	AES-256
Key Size (Bits)	128	192	256
Number of rounds	10	12	14
Plaintext box size (Bits)	128	128	128

a) AES Encryption Algorithm

The Encryption process consists of a number of different transformations applied consecutively over the data block bits in a fixed number of iterations which is called as rounds. The number of rounds depends on the length of the key used for the encryption process. 10 iterations are required for key length of 128 bits.

i. *High-level description of the algorithm*

KeyExpansions -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.

ii. *InitialRound*

1. AddRoundKey( )- Each byte of the state is combined with a block of the round key using bitwise xor.

Rounds

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

iii. *Final Round (No MixColumns)*

SubBytes( )
ShiftRows( )
AddRoundKey( ).

Steps : These steps are used to encrypt128-bit block.

1. The set of round keys from the cipher key.
2. Initialize state array and add the initial round key to the starting state array.
3. Perform round = 1 to 9 : Execute Usual Round.
4. Execute Final Round.
5. Corresponding cipher text chunk output of Final Round Step

iv. *Encryption process*

Each round consists of the following four steps:

SubBytes Transformation: In this transformation, each of the byte in the state matrix is replaced with another byte as per the S-box (Substitution Box)[7]. The S-box is generated by firstly calculating the respective reciprocal of that byte in GF (2^8) and then affine transform is applied.

ShiftRows Transformation: In this transformation, the bytes in the first row of the State do not change. The second, third, fourth and fifth rows shift cyclically to the left by one byte, two bytes, three bytes and four bytes respectively [7].

MixColumns Transformation: It is the operation that mixes the bytes in each column by the multiplication of the state with a fixed polynomial matrix [7]. It completely changes the scenario of the cipher even if all bytes look very similar. The Inverse Polynomial Matrix does exist in order to reverse the mix column transformation.

AddRoundKey Transformation: In AddRoundKey transformation, a roundkey is added to the State by bitwise Exclusive-OR (XOR) operation. AddRoundKey proceeds onecolumn at a time. AddRoundKey adds a roundkey word with each state column matrix.The operation performed in AddRoundKey is matrix addition.

b) AES Decryption Algorithm

Decryption is the process of extracting the plaintext from cipher text. For decryption the same process occurs simply in reverse order by taking the 128-bit block of cipher text and converting it to plaintext by the application of the inverse of the four operations. Decryption involves reversing all the steps taken in encryption using following inverse functions.

InvSubBytes Transformation: InvSubBytes is the inverse transformation of SubBytes, in which the inverse S-box is applied to individual bytes in the State. The inverse S-box is constructed by first applying the inverse of the affine transformation, then computing the multiplicative inverse in GF(2^8).

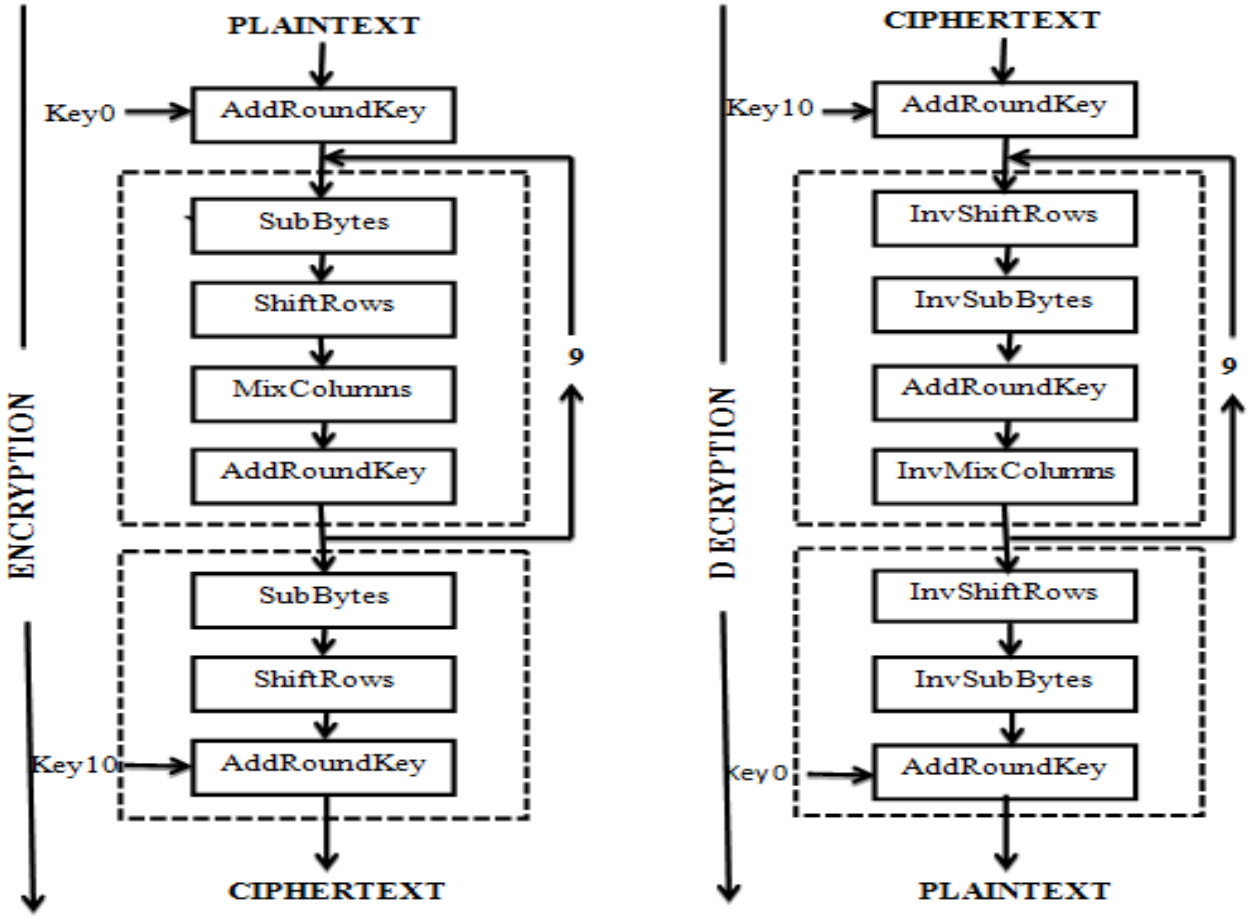

Figure 1: AES Encryption and Decryption

InvShiftRows Transformation: InvShiftRows is the inverse transformation of ShiftRows. In this transformation, the bytes in the first row of the State do not change. The second, third, and fourth and fifth rows are shifted cyclically by one byte, two bytes, three bytes and four bytes to the right respectively [2].

InvMixColumns Transformation: InvMixColumns is the inverse transformation of MixColumns. This is a complex procedure as it involves severely the byte multiplication under GF (2^8)[2].

Key Expansion (Keyexpansion Operation)

Keyexpansion refers to the process in which the 128 bits of the original key are expanded into eleven 128-bit round keys.

To compute round key (n+1) from round key (n) these steps are performed:

1. Compute the new first column of the next round key. First all the bytes of the old fourth column have to be substituted using the Subbytes operation. These four bytes are shifted vertically by one byte position and then
   XORed to the old first column. The result of these operations is the new first column.
2. Columns 2 to 4 of the new round key are calculated as shown:

- [new second column] = [new first column] XOR [old second column]
- [new third column] = [new second column] XOR [old third column]
- [new fourth column] = [new third column] XOR [old fourth column]

The key expansion algorithm generates 128 bit key for each round and one more key for initial AddRoundKey function. The same expanded key is used for encryption and decryption except for decryption it reads in reverse order.

III. Implementation Details

The system proposing aims to achieve network security by implementing appropriate countermeasures based on concept of constant time encryption against side channel timing attack to protect implementations of secret key cryptography. The contribution work includes implementing more suitable countermeasures against side channel attack.

a) System Overview

The propose system, is intended to provide secure transmission of data over the network by implementing the appropriate countermeasures against side channel attack on AES implementation which is shown in Fig.2. Here the work implementing AES 128-bit algorithm using 10 rounds by taking input as text, image and audio. In AES encryption process, system

performs round functions like SubByte( ), ShiftRows( ), MixColumn( ) and AddRoundKey( ). On the other side, the decryption processperforms round functions like InvShiftRow( ), InvSubByte( ), AddRoundKey( ) and InvMixColumn( ). After that the work implementing side channel attack on the AES implementation in such a way that the receiver cannot decrypt the encrypted data. After successful implementation of side channel attack, research work implementing some appropriate countermeasures against side channel attack on AES implementation and finally evaluating their performance and soundness to prevent possible vulnerabilities and develop more secure systems.

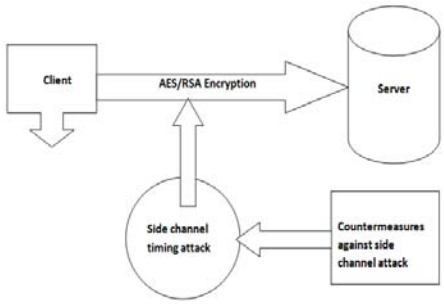

Figure 2 : System architecture

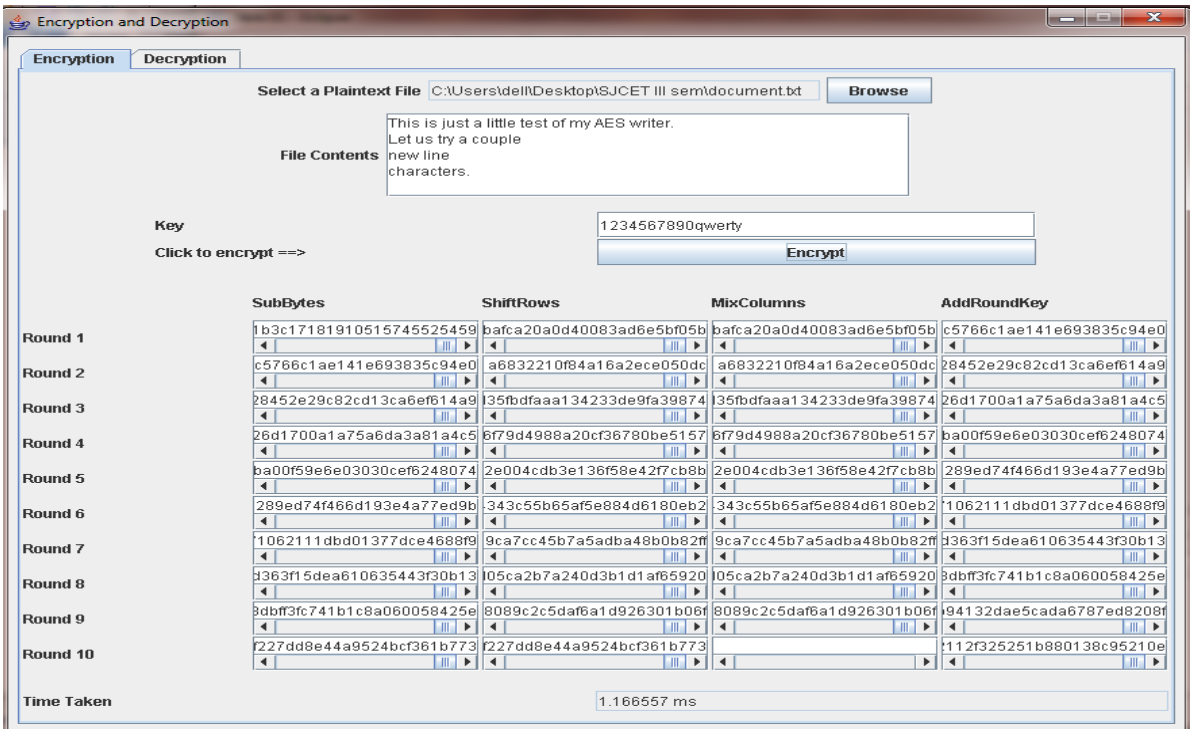

Figure 3 : AES Encryption: Input as Text

Decryption Process when input as an Text file

The work implemented 128 bit AES algorithm (10 round) decryption using text as an input by measuring performance parameter as time complexity which is shown in Fig.4.Time required for decryption process is 2.128282 milliseconds.

b) AES Implementation

The work implemented AES 128-bit, 10 rounds algorithm by taking input as text, image and audio.

Encryption Process when input as an Text file

The work implemented 128 bit AES algorithm (10 round) encryption using text as an input by measuring performance parameter as time complexity which is shown in Fig.3.Time required for encryption process is 1.166557 milliseconds.

Encryption Process when input as an audio file

The work implemented 128 bit AES algorithm (10 round) encryption using audio as an input by measuring performance parameter as time complexity which is shown in Fig.5.Time required for encryption process is 13.899532 milliseconds.

Decryption Process when input as an audio file

The work implemented 128 bit AES algorithm (10 round) decryption using audio as an input by measuring performance parameter as time complexity which is shown in Fig.6.Time required for decryption process is 20.183485milliseconds.

Encryption Process when input as an Image file

The work implemented 128 bit AES algorithm (10 round) encryption using image as an input by measuring performance parameter as time complexity which is shown in Fig.7. Time required for encryption process is 61.958627 milliseconds.

Decryption Process when input as an Image file

The work implemented 128 bit AES algorithm (10round) decryption us ing image as an input by measuring performance parameter as time complexity which is shown in Fig.8.Time required for decryption process is 31.509569milliseconds.

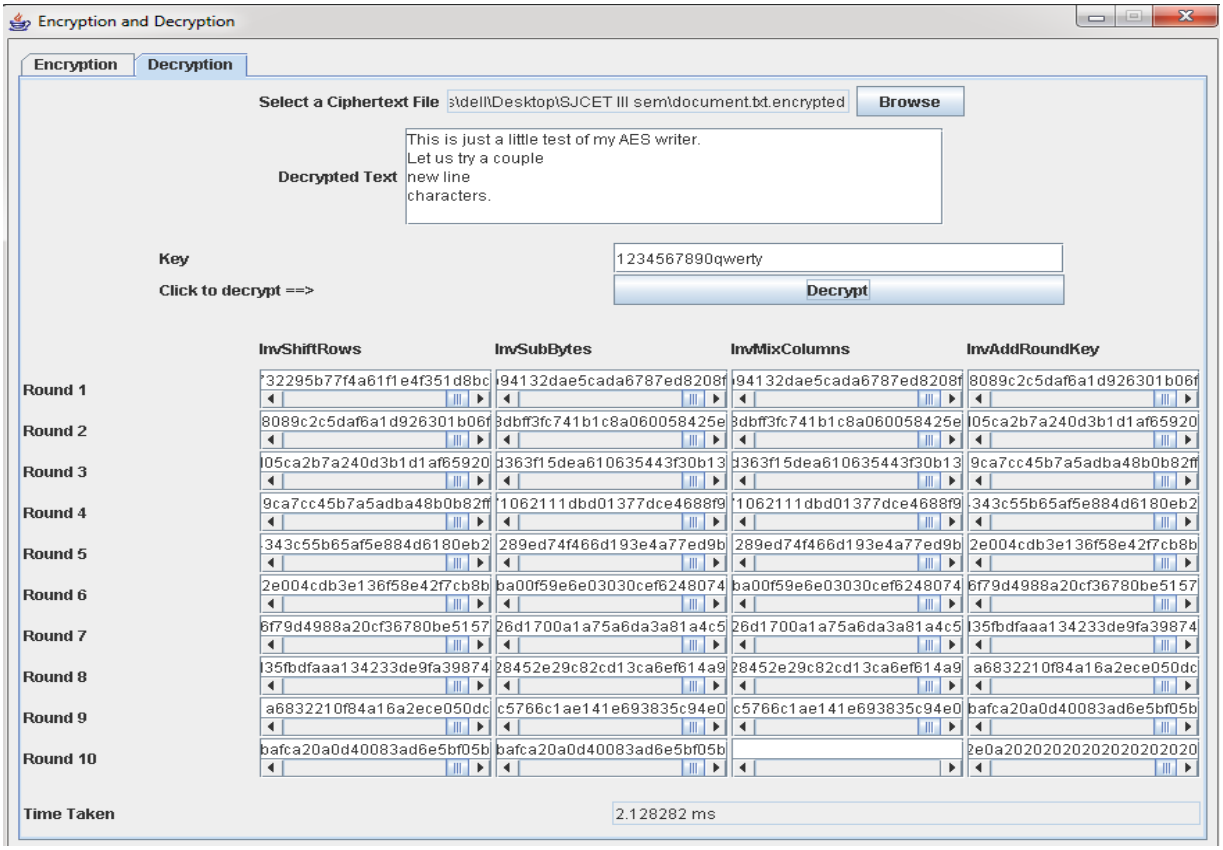

Figure 4 : AES Decryption :Input as Text

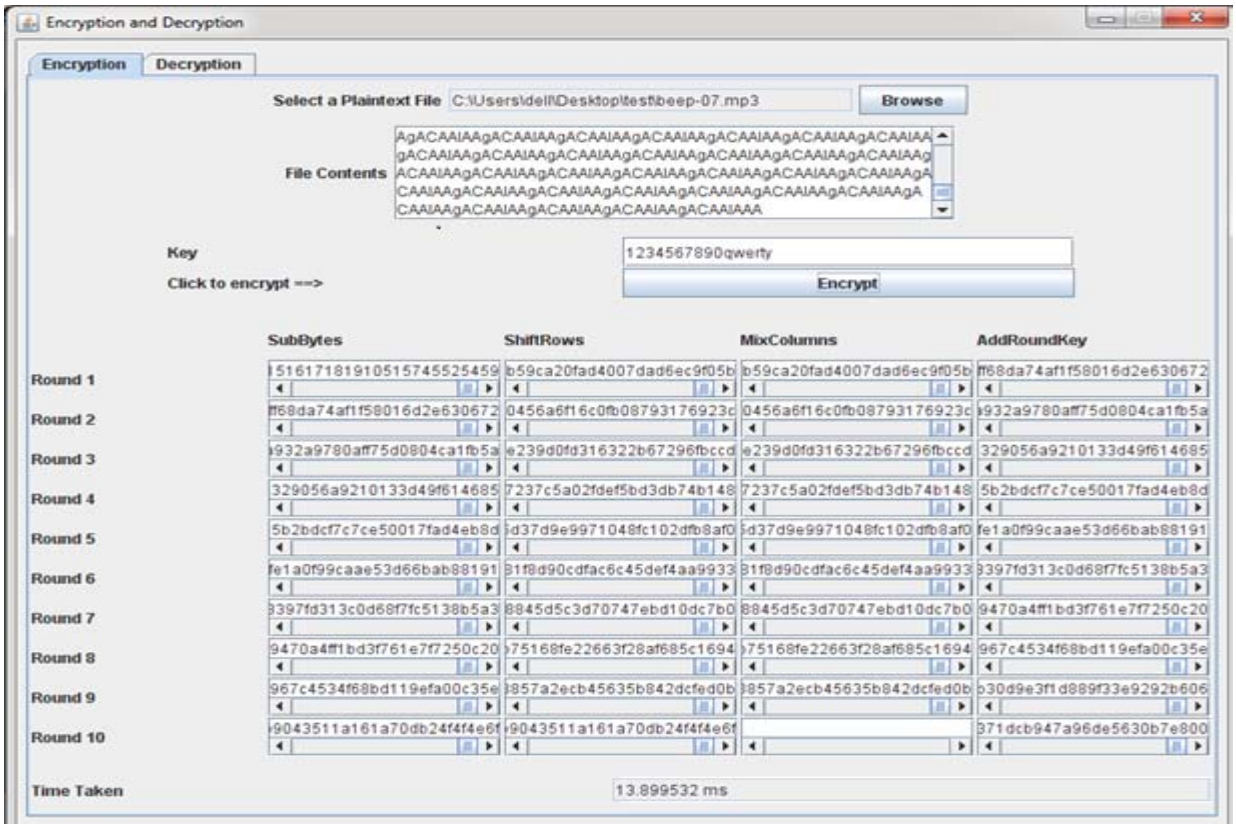

Figure 5 : AES Encryption: Input as Audio

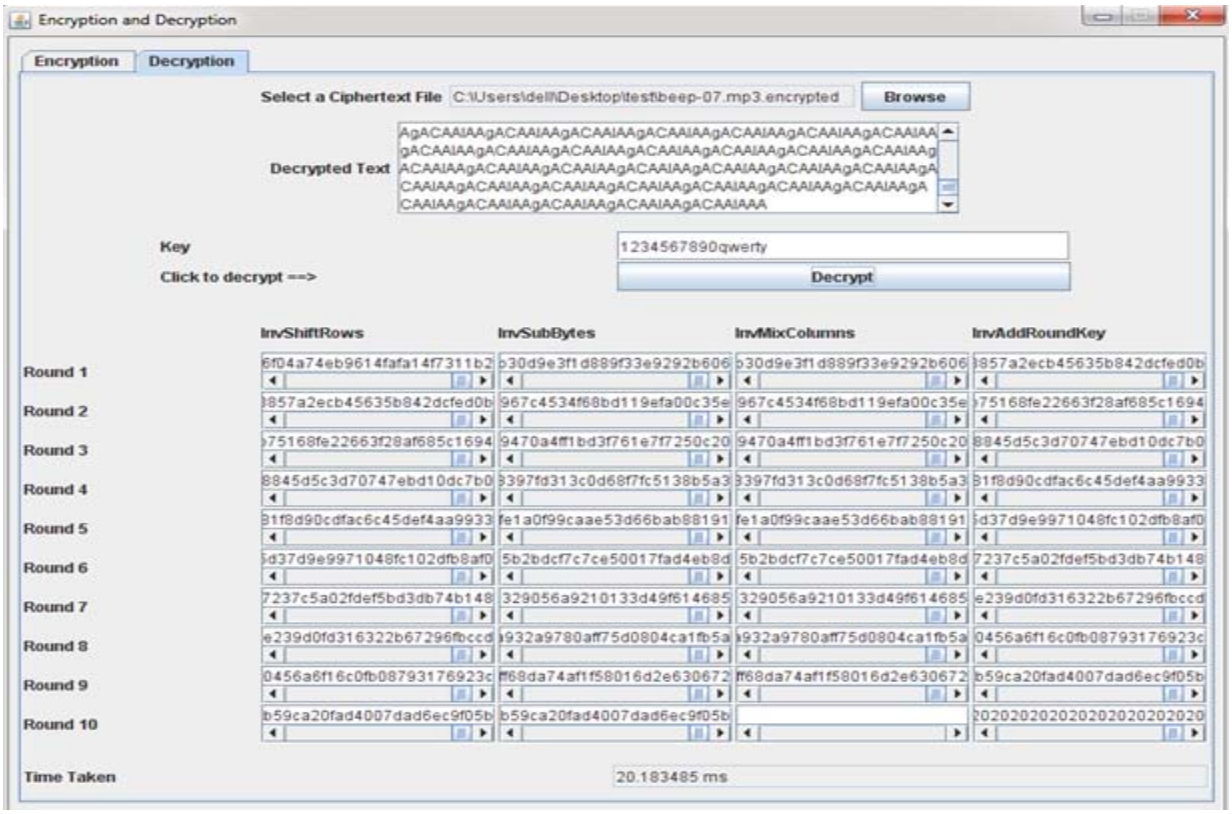

Figure 6 : AES Decryption: Input as Audio

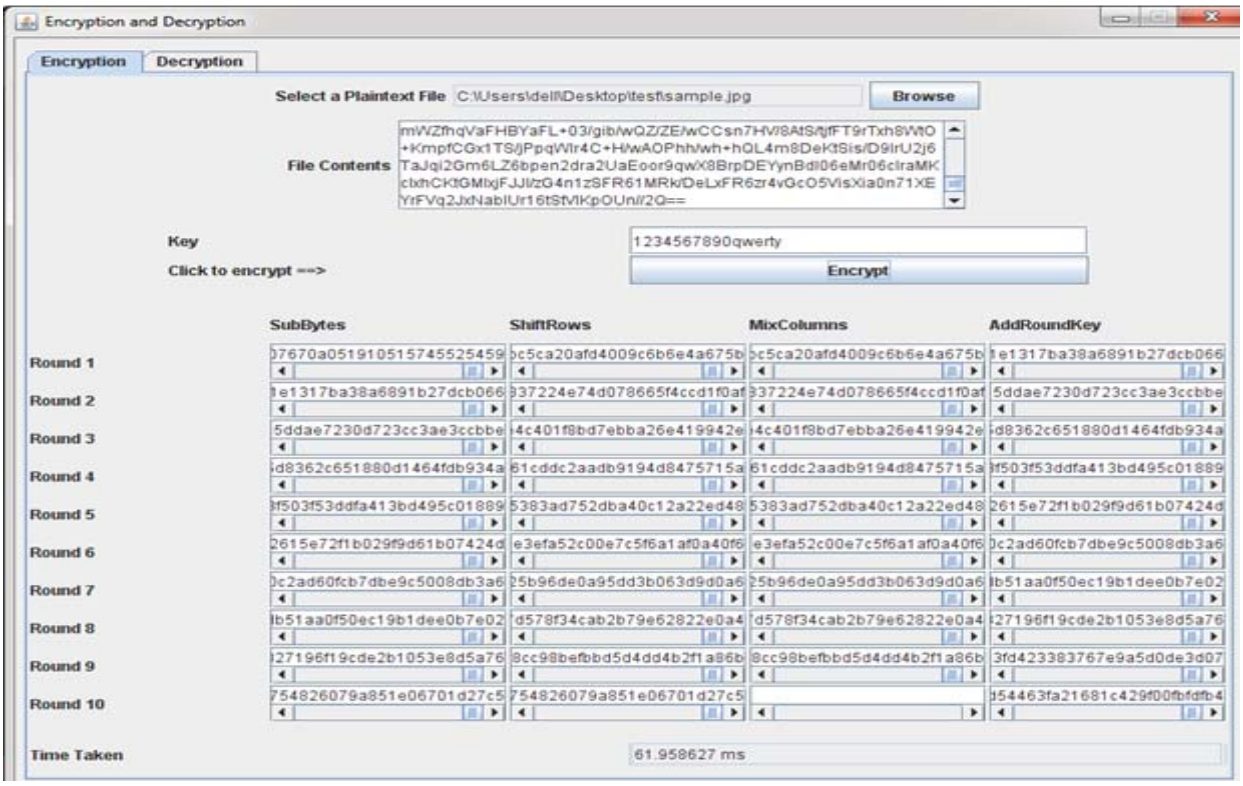

Figure 7 : AES Encryption: Input as Image

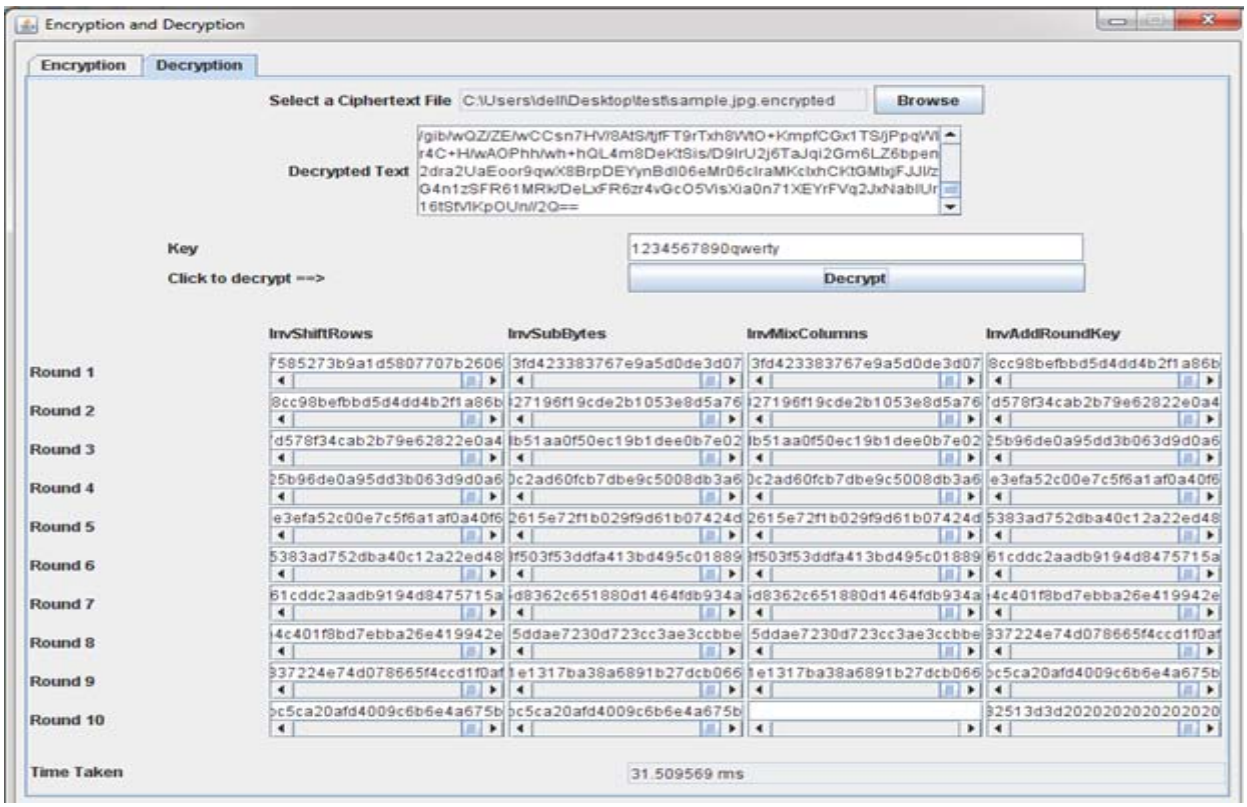

Figure 8 : AES Decryption Process: Input as Image

c) Attack on AES implementation

After successful implementation of AES algorithm. The work implemented attack in such a way that at the time of decryption, receivers cannot get the decrypted file as a plain text file instead of that the user will get the file which is in the human non-readable format which is shown in the Fig.9.

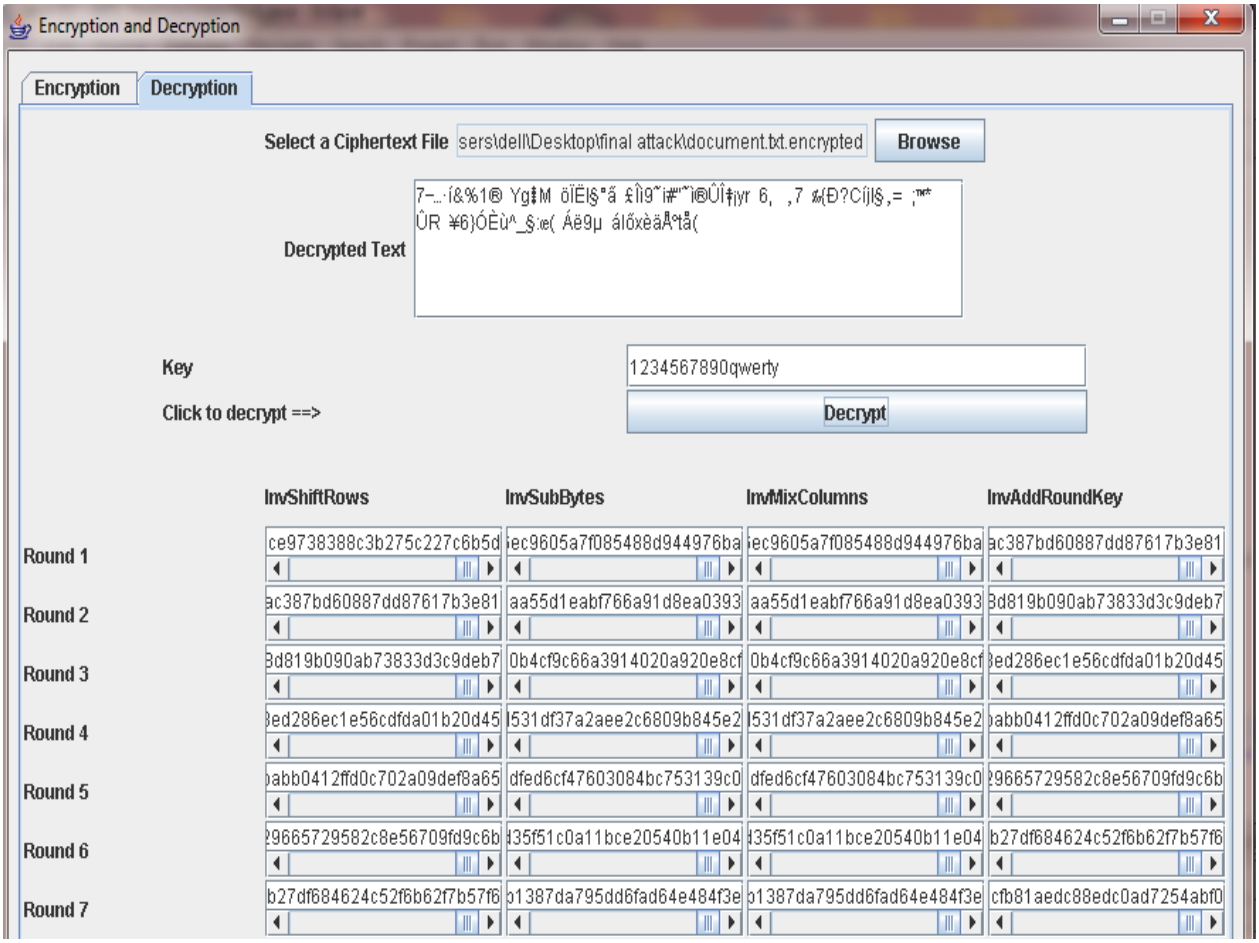

Figure 9 : A on AES implementation

IV. Experimental Results

In this section The work presented result graph of our proposed system, implementation of the AES algorithm by taking text, image and audio as input. The work used 10 rounds technique for implementing AES 128- bit algorithm.

a) Result graph for encryption time

In Fig. 10. The graph shows the time needed to encrypt the input as a text, image and audio data file by the proposed system .

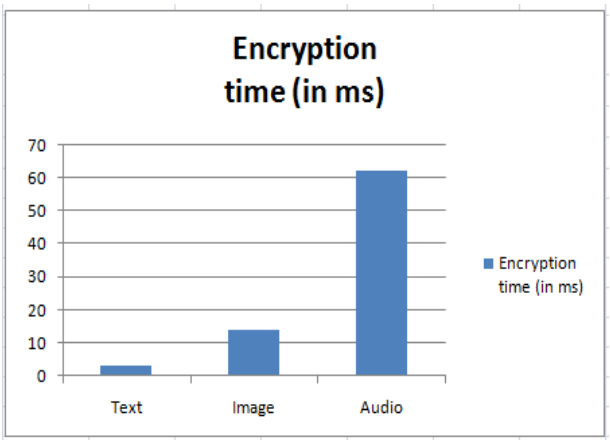

Figure 10 : Encryption time taken by AES algorithm

b) Result graph for decryption time

In Fig. 11. The graph shows the time needed to decrypt the input as a text, image and audio data file by the proposed system .

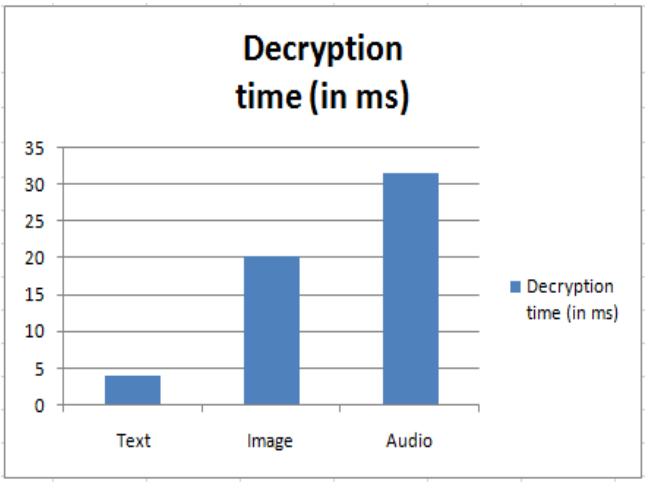

Figure 11: Decryption time taken by AES algorithm

V. Conclusion Andfuture Scope

Due to the increasing needs for secure communications a safer and more secured cryptographic algorithm has to be proposed and implemented. The Advanced Encryption Standard (AES-128bit) is widely used nowadays in many applications. In this paper, the work implemented an efficient AES128 bit encryption and decryption algorithm. The execution time for AES encryption and decryption is calculated by performing 10 round functions. The system presented an attack on AES software implementations. Future work will focus on investigating and implementing a number of countermeasures against side channel attack on AES implementation and have evaluated their performance and soundness to prevent possible vulnerabilities and develop more secure systems.

References Références Referencias

1. Vinayak Bajirao Patil, Prof.Dr.Uttam.L.Bombale ,Pallavi Hemant Dixit, "Implementation of AES algorithm on ARM processor for wireless network, " in International Journal of Advanced Research in Computer and Communication Engineering ,Vol. 2, Issue 8, August 2013, pp.3204-3209.
2. Xinmiao Zhang and Keshab K. Parhi, "Implementation approaches for the advanced encryption standard algorithm," in IEEE Transactions , 2002.
3. Chih-Pin Su, Tsung-Fu Lin, Chih-Tsun Huang, and Cheng-Wen Wu,"A high throughput low cost AES processor," in IEEE Communications Magazine, 2003 .
4. "Advanced Encryption Standard (AES)" Federal Information Processing Standards Publication 197, Nov. 2001.
5. M.Gnanambika, S.Adilakshmi, Dr.Fazal Noorbasha, "AES-128 Bit Algorithm Using Fully Pipelined Architecture for Secret Communication," in International Journal of Engineering Research and Applications Vol. 3, Issue 2, March -April 2013, pp.166-169.
6. Rishabh Jain, Rahul Jejurka2, Shrikrishna Chopade, Someshwar Vaidya, Mahesh Sanap, "AES Algorithm Using 512 Bit Key Implementationfor Secure Communication," in International Journal of Innovative Research in Computerand Communication Engineering, Vol. 2, Issue 3, March 2014, pp. 3516-3522.
7. Vinayak Bajirao Patil, Prof. Dr. Uttam. L.Bombale, Pallavi Hemant Dixit, "Implementation of AES algorithm on ARM processor for wireless network, " in International Journal of Advanced Research in Computer and Communication Engineering ,Vol. 2, Issue 8, August 2013,pp.3204-3209.
8. Ritu Pahal and Vikas kumar, "Efficient Implementation of AES," in International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, Issue 7, July 2013, pp. 290-295
9. K. Soumya, G. Shyam Kishore, "Design and Implementation of Rijndael Encryption Algorithm Based on FPGA," in International Journal of Computer Science and Mobile Computing, Vol. 2, Issue. 9, September 2013, pp.120 – 127.
10. Sumedha Kaushik and Ankur Singhal, "Network Security Using Cryptographic Techniques ,"in International Journal of Advanced Research in Computer Science and Software Engineering ,Vol. 2, Issue 12, December 2012, pp.105-107.
11. Dr. Prerna Mahajan & Abhishek Sachdeva, "A Study of Encryption Algorithms AES, DES and RSA forSecurity," in Global Journal of Computer Science and Technology Network, Web & Security, Vol 13, Issue 15,2013, pp.15-22.
12. H. Kuo and I. Verbauwhede, "Architectural optimization for a 1.82 Gbits/sec VLSIimple mentation of the AES Rijndael algorithm," in Proc. CHES 2001, Paris, France, May 2001, pp. 51-64.
13. Navraj Khatri, Rajeev Dhanda, Jagtar Singh," Comparison of power consumption and strict avalanche criteria at encryption/Decryption side of Different AES standards,'' International Journal Of Computational Engineering Research, Vol. 2 Issue. 4, August 2012.
14. Das Debasis, Misra Rajiv. "Programmable Cellular Automata Based Efficient Parallel AES Encryption Algorithm". International Journal of Network Security & Its Applications (IJNSA), Vol. 3, No.6, November 2011, pp. 204.
15. Hiremath.S. and Suma.M.S.,"Advanced Encryption Standard Implemented on FPGA,"in IEEE Inter. Conf. Comp Elec Engin. (IECEE), vol. 02,issue. 28, pp.656-660,Dec.2009.
16. Chehal Ritika, Singh Kuldeep. "Efficiency and Security of Data with Symmetric Encryption Algorithms," inInternational Journal of Advanced

Research in Computer Science and Software Engineering, Vol. 2, Issue 8, August 2012, pp. 1.

17. Z. Xinjie, W. Tao, M. Dong , Z. Yuanyuan, L. Zhaoyang, "Robust First Two Rounds Access Driven Cache Timing Attack on AES," in IEEE InternationalConference on Computer Science and Software Engineering , Dec. 2008, Wuhan, Hubei, China, pp. 785 – 788.
18. P.C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS and other systems," in 16th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO), 1996, Springer-Verlag London UK, pp. 104-113.
19. W. Stallings, Cryptography and network security.
20. J. Daemen and V. Rijmen, AES Proposal: Rijndael (Version 2).