International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 05 | May-2016 www.irjet.net p-ISSN: 2395-0072

A MODIFIED APPROACH TO IMPROVISE THE EFFICIENCY OF DES AND AES USING 1024-BIT KEY

[*1] Mrs. Rajalakshmi M.R.K.,[*2] Ms. Abarna N.,

*[*1] M.Phil Research Scholar, PG & Research Department of Computer Science and Information Technology, Arcot Sri Mahalakshmi Women's College, villapakkam, Tamil Nadu, India.*
*[*2]. Assistant Professor, PG & Research Department of Computer Science and Information Technology, Arcot Sri Mahalakshmi Women's College, villapakkam, Tamil Nadu, India.*

Abstract - ***Cryptography is the art of achieving security by encoding messages to make them non-readable. There are two basic types of cryptography: Symmetric and Asymmetric cryptography. Symmetric Key algorithms are the quickest and most commonly used type of encryption. Here, a single key is used for both encryption and decryption. There are few well-known Symmetric Key algorithms: DES, RC2, RC4, AES etc. are much private but some are less private because the attacker or the hackers can hack the messages. DES is now considered to be insecure for many applications because the 56-bit key size is too small and possible to brute-force in finite time on modern processor. The AES algorithm was believed to provide more security than the DES. A new scheme of Symmetric Key algorithm DES and AES is proposed with 1024 bit key. This technique provides more security and increases the efficiency with different key length settings. Data Encryption Standard (DES) and Advanced Encryption Standard (AES) with 1024 bit key is implemented using NS2 software to make comparison on the basis of parameters like speed, block size, and key size.***

Key Words: *Data Encryption Standard (DES), Advanced Encryption Standard (AES).*

I. INTRODUCTION

Cryptography is the branch of computer science that deals with hiding information for secure communication of data. It uses the codes to convert plain text into cipher text, so that only the intended recipient will be able to read it using the key. The conversion of plain text into cipher text is called Encryption. And the conversion of cipher text into plain text is called Decryption.

Secret key cryptography, uses a single key for both encryption and decryption. The sender uses the key (or some set of rules) to encrypt the plaintext and sends the cipher text to the receiver. The receiver applies the same key to decrypt the message and recover the plaintext. Because a single key is used for both functions, secret key cryptography is also called Symmetric Encryption. With this form of cryptography, it is obvious that the key must be known to both the sender and the receiver and it must be kept as a secret key. The proposed solution aimed to strengthen the key generation of the DES. It encrypts a 128 bit data block using two keys by initial permutation followed by sixteen rounds of crossover iterations using two keys and going through a final inversed permutation. Its weaknesses are the same with regular Blowfish although it offers more resistance to Brute Force attacks but the two keys added to the encryption slows the process. The encryption process starts by converting a 128-bit key into a binary value. The result increases 1024 bit and goes through Odd-Even substitution.

Each half contains 64-bits and performs the left shift. After shifting is applied, the two halves are combined and reduced to 64 bits. The pipes (||) demonstrate the combination of the two halves. The 128-bit produced is now the first key. The 128 bits (C_1 and D_1) are used to generate the remaining keys and undergo the same steps generating the first key. This will be performed for 8 rounds. The aim is to present the AES 1024 bit when used in higher level of security throughput are required without increasing overall design area as compared to the original 128 bit AES algorithm. The proposed algorithm consist of the structure which is similar to original AES algorithm but having slight difference that of the plaintext size and key size using input of 512 bit instead of 128 bit has impact on the whole algorithm.

II. RELATED WORK

The DES is a basic building block for data protection. The algorithm provides the user with a set of functions, which transforms a 64-bit input to a 64-bit output. Anyone knowing the key can calculate both the function and its inverse, but without knowing the key it is

infeasible to determine which function was used, even when several inputs and outputs are provided. DES is a block cipher which uses a Feistel cipher to achieve confusion and diffusion of bits from the plaintext to the cipher text.

The DES is a block cipher in which 16 iteration/ rounds, of substitution and transposition (permutation) processes are cascaded. The block size is 64 bits. The key which controls the transformation also consists of 64 bits; however, only 56 of these can be chosen by the user and are actually key bits. The remaining 8 bits are parity check bits and hence totally redundant.

In DES, 56-bit key is used and then it is treated as two 28-bit quantities, labeled C_0 and D_0. At each round, C_{i-1} and D_{i-1} are separately subjected to a circular left shift or (rotation) of 1 or 2 bit. These shifted values serve as input to the next round. They also serve as input to the part labeled Permuted Choice Two which produces a 48-bit output that serves as input to the Function F (R_i-1, K_i).

Encryption is of prime importance when confidential data is transmitted over the network. Varied encryption algorithms like AES, DES, RC4 and others are available for the same. The most widely accepted algorithm is AES algorithm so it is used for encryption and decryption of the data. This application provides a secure, fast, and strong encryption of the data. As a conclusion the requirements for speed and compactness are met. The program size is 50 kB and it can be installed into a mobile phone working on Android platform. The user experiences no delays while using the program, which is a clear indication that the speed requirement is met.

The new Advanced Encryption Standard (AES) has been recently selected by the US government to replace the old Data Encryption Standard (DES) for protecting sensitive official information. Due to its simplicity and elegant algebraic structure, the choice of the AES algorithm has motivated the study of a new approach to the analysis of block ciphers. While conventional methods of cryptanalysis (e.g. differential and linear cryptanalysis) are usually based on a statistical approach, where an attacker attempts to construct statistical patterns through many interactions of the cipher, the so-called *algebraic attacks* exploit the intrinsic algebraic structure of a cipher.

More specifically, the attacker expresses the encryption transformation as a set of multivariate polynomial equations and attempts to recover the encryption key by solving the system. In contrast, algebraic attacks exploit the intrinsic algebraic structure of a cipher. More specifically, the attacker expresses the encryption transformation as a (large) set of multivariate polynomial equations, and subsequently attempts to solve such a system to recover the encryption key.

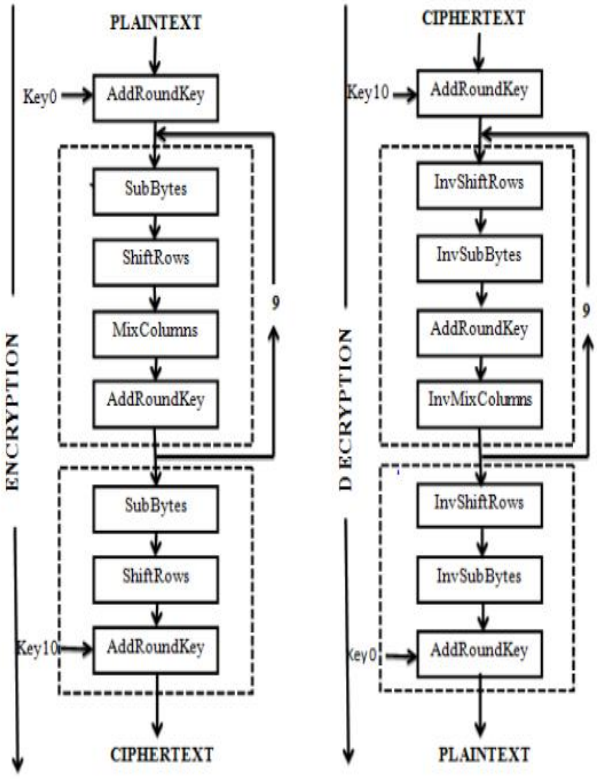

Fig 1.1: AES Structure

III. PREVIOUS IMPLEMENTATIONS

Data Encryption Standard (DES) was known to be weak to cryptanalysis and brute-force attacks. Other cryptographic algorithms were available to substitute DES, but many of the cryptographic algorithms were either safe by patents or considered proprietary. Trying all 256 possible keys is not that hard these days. If you spend 25k you can build a DES password cracker that can will succeed in a few hours. DES is up to 56 bits only. In the case of nonce reuse both integrity and confidentiality properties are violated. If the same nonce is used twice, an adversary can create forged cipher texts easily when short tags are used; it is rather easy to produce message forgeries. For instance, if the tag is 32 bits, then after 216 forgery attempts and 216 encryptions of chosen plaintexts (also of length 216), a forged cipher text can be produced. Creation of forgeries can be instantaneous when enough forgeries have been found. A secondary drawback is that both DES and AES use a 64-bit block size. For reasons of both efficiency and security, a larger block size is desirable and key can be expanded to 1024 bits.

IV. SYSTEM IMPLEMETNATION

The proposed solution aimed to strengthen the key generation of the DES. It encrypts a 128 bit data block using two keys by initial permutation followed by sixteen rounds of crossover iterations using two keys and going

through a final inversed permutation. Its weaknesses are the same with regular Blowfish although it offers more resistance to Brute Force attacks but the two keys added tothe encryption slowed the process. The encryption process starts by converting a 128-bit key into a binary value. The result is increase to 1024 bit and goes through Odd-Even substitution. The Odd Even substitution process substitutes 1 for every even position and 0 for every odd position in the 128-bit block.

4.1 Data Encryption Standard

Modes of Operation for Encryption Algorithms DES belong to a category of ciphers called block ciphers. Block ciphers, as opposed to stream ciphers, encrypt messages by separating them into blocks and encrypting each block separately. Stream ciphers, on the other hand, operate on streams of data one bit at a time as a continuous stream. DES encrypts 1024-bit blocks of plaintext into 1024-bit blocks of cipher text. Plaintext, used in the context of cryptography, is the name commonly given to the body of a message before it is encrypted.

4.2 Implementation of DES

DES relies upon the encryption techniques of confusion and diffusion. Confusion is accomplished through substitution. Specially chosen sections of data are substituted for corresponding sections from the original data. The choice of the substituted data is based upon the key and the original plaintext. Diffusion is accomplished through permutation. The data is permuted by rearranging the order of the various sections. These permutations, like the substitutions, are based upon the key and the original plaintext.

The substitutions and permutations are specified by the DES algorithm. Chosen sections of the key and the data are manipulated mathematically and then used as the input to a look-up table. In DES these tables are called the S-boxes and the P-boxes, for the substitution tables and the permutation tables, respectively. In software these look-up tables are realized as arrays and key/data input is used as the index to the array. Usually the S-boxes and P-boxes are combined so that the substitution and following permutation for each round can be done with a single look-up.

The algorithmic implementation of DES is known as DEA for Data Encryption Algorithm and the 512-bit right half of the 1024-bit input data block is expanded by into a 48-bit block. This is referred to as the expansion permutation steps.

DES works on 1024-bit plain text blocks. The key is longer and consist of 128 bits. The 1024 bit input plain text block is divided into four portions of plain text (each of size 16 bits), say P1 to P4. Thus P1 to P4 are inputs to

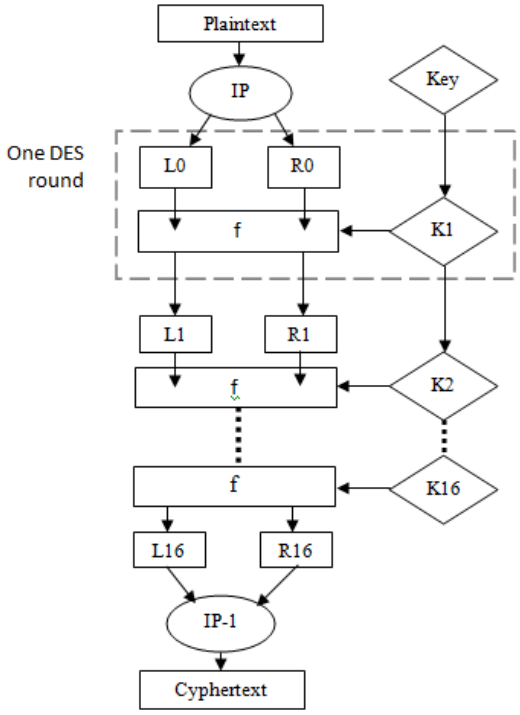

Fig 4.1 DES Core Algorithm

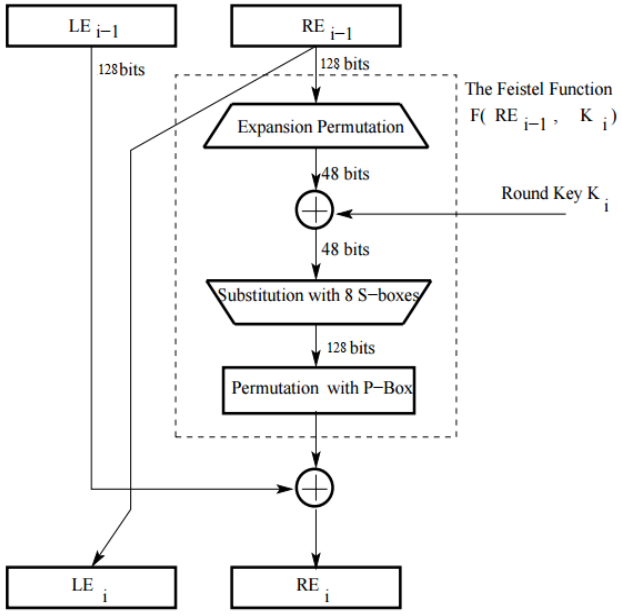

Fig 4.2: DES Implementation

the to the first round of algorithm and there are 8 such rounds. The output of first round is input to second round. Similarly output of second round is input to third round, and so on. In each round six sub keys are generated from the original key. Each of the sub keys consists of 16 bits. For first round we will have keys K1 to K6. For second round we will have keys K7 to K12. Finally for eighth round we will have keys from K43 to K48. The final step consists of output transformation, which uses just four sub

keys (K49 to K52). The final output produced is the output produced by output transformation, which is four blocks of cipher text C1 to C4. These are combined to form final 64 bit cipher text block. In this algorithm addition is done modulo 216 and multiplication is done modulo216+1. the technique of key shifting is employed here.

Algorithm Implementation

Step 1: First divide the 128-bit block into eight 4-bit words

Step 2: Attach an additional bit on the left to each 4-bit word that is the last bit of the previous 4-bit word

Step 3: Attach an additional bit to the right of each 4-bit word that is the beginning bit of the next 4-bit word.

Step 4: The 1024-bit key is divided into two halves, each half shifted separately, and the combined 1024-bit key permuted/contracted to yield a 128-bit round key.

Step 5: The 48 bits of the expanded output produced by the are XORed with the round key. This is referred to as key mixing.

Step 6: The output produced by the previous step is broken into eight six-bit words. Each six-bit word goes through a substitution step; its replacement is a 4-bit word. This substitution is carried out with the S-box

The goal of the substitution is implemented by the S-box to introduce diffusion in the generation of the output from the input. Diffusion means that each plaintext bit must affect as many cipher text bits as possible. The strategy used for creating the different round keys from the main key is meant to introduce confusion into the encryption process.

Confusion in this context means that the relationship between the encryption key and the cipher text must be as complex as possible. Another way of describing confusion would be that each bit of the key must affect as many bits as possible of the output cipher text block.

4.3 AES Implementation

The round structure of the AES-1024 algorithm uses the transformation defined in the previous section. First, byte substitution is performed on 128 bits data, followed by row rotation according to the row number, where 0-7 left rotations are performed in this step. Then, the columns are multiplied by the new defined matrix column by column in the Mix Column transformation.

Input: Byte A[8× nb], Word K[nb × (nr + 1)];
Output: Byte C[8× nb],
Byte State[4, nb]; state := A;
AddRoundKey (state, K[0, nb − 1];
for round := 1 to nr − 1 do SubBytes(state);
ShiftRows (state);
MixColumns (state);
AddRoundKey (state,K[round×nb,nb(round+1))
SubBytes(state);
ShiftRows(state);
AddRoundKey (state, K[nr × nb, nb(nr + 1) − 1]);
C :=state;
Return C;
End

Bytes Substitution

The 1024 bits input plaintext is organized in array of 128 bytes and are substituted by values obtained from substitution boxes. This is done to achieve more security according to diffusion-confusion, the Shannon's principles for cryptographic algorithm design.

Shift Row

After the original 1024-bit data is substituted with values from the S-boxes, the rows of the resulting matrix are shifted in a process called Shift Row transformation. What happened in this part is that the bytes in each row in the input data matrix will be rotated left. The number of left rotations is not the same in each row, and it can be determined by the row number.

```
ShiftRows(byte state[4,Nb])
  begin byte t[Nb]
        for r = 1 step 1 to 3for c = 0 step 1 to Nb – 1
        t[c] = state[r, (c + h[r,Nb]) mod Nb]
  end for
        for c = 0 step 1 to Nb– 1
                state[r,c] = t[c]
        end for
   end for
End
```

Add Round Key

To make the relationship between the key and the cipher text more complicated and to satisfy the confusion principle, the AddRoundKey operation is performed. This addition step takes the resulting data matrix from the previous step and performs bitwise XOR operation with the subkey of that specific round (addition operation in GF (2^n)). We must mention that the round key is 1024 bits that is arranged in a square matrix of eight columns where each column has 16 bytes.

Mix Column

This is perhaps the hardest step to both understand and explain. There are two parts to this step. The first will explain which parts of the state are multiplied against which parts of the matrix. The second will explain how this multiplication is implemented over the Galois Field

W(I) = W(I - 8) XORW (I-1)I is not a multiple of 16
W(I) = W(I - 8)XOR T(W(I-1)) I is a multiple of 16

Where the T (I) transformation is defined as:

T(I)=Byte Sub(Shift Left(W(I)))XOR Round Const

The round constant is defined by the following equation:

Round Const = $00000010^{(i-16)}/16$.

Key Expansion and Rounds

The 1024-bit input key of the new AES-1024 algorithm is used to generate ten sub-keys for each of the ten AES rounds. The round ±keys expansion process involves arranging the original 512-bits input key into eight words of eight bytes each.

```
Key Expansion(byte key[4 * Nk], word w[Nb * (Nr + 1)],
Nk)
Begin
    i = 0
    while (i < Nk)
    w[i] = word[ key[4*i], key[4*i+1], key[4*i+2],
           key[4*i+3] ]
    i = i + 1
    end
    i = Nk
    while (i < Nb * (Nr + 1))
      word temp = w[i- 1]
      if (i mod Nk = 0)
        temp = SubWord(RotWord (temp)) xor Rcon[i/Nk]
      else if ((Nk = 8) and (i mod Nk = 4))
          temp = SubWord(temp)end ifw[i] = w[i - Nk]
                  xor tempi = i + 1
     end while
End
```

V EVALUATION RESULT:

The Data Encryption Standard algorithm and Advanced Encryption Algorithm is implemented using NS2 simulator.

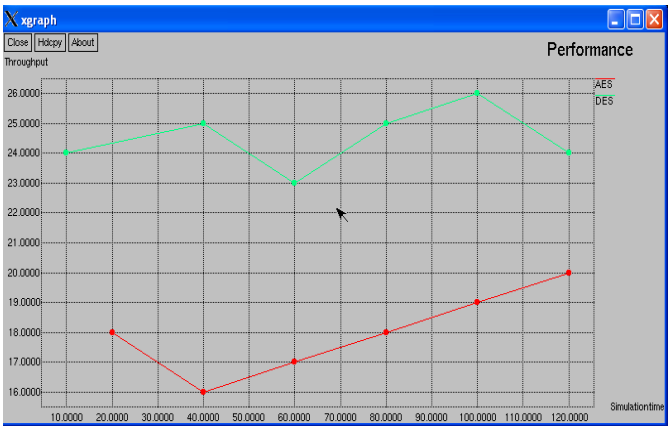

Fig 5.1 Performances of DES and AES

Key Size	AES	DES
64	2400	1800
128	2550	1600
196	2300	1700
256	2500	1800
512	2600	1900
1024	2400	2000

Table 5.1 Performance of AES VS DES

Security

Since the security features of each algorithm as their strength against cryptographic attacks. The chosen factor here determines the performance of the algorithm's speed to encrypt/decrypt data blocks of various sizes.

Fig 5.2: Security of AES Vs DES

Key Size	AES	DES
32	400	400
64	1200	260
128	1420	600
196	1400	800
256	1600	950
512	2100	1100
1024	2550	1600

Table 5.2: Security Comparison of AES Vs DES

Computation Cost

AES and DES key generation involves primarily testing, which is an expensive operation. Besides, prime tests are probabilistic, which means that the execution times are not always the same and occasionally can be very long.

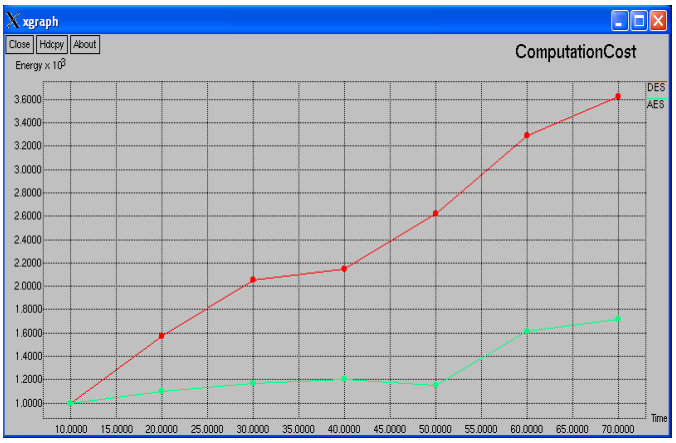

Fig 5.3 Computation Cost

Key Size	AES	DES
32	1000	1000
64	1500	1050
128	2050	1150
196	2150	1200
256	1600	1160
512	3250	1400
1024	3600	1760

Table 5.3 Computation Cost of AES vs DES

Comparison of DES and AES

Parameters	DES 1024	AES 1024
Key Size	1024 Bits	1024 bits
Data Block Size	Not Same AS Key Size	Same AS Key Size
Rounds	8	8
Throughput	90%	230%
Time to Encrpt128 char Message	30-50 seconds	10-30 seconds
Security	Less	More
Process required	More Amount	Less as Compared 512

Table 5.4 Comparison of AES and DES

CONCLUSION

Encryption algorithm plays an important role in communication security, a more safe and secure cryptographic algorithms have to be proposed and implemented. A proposed new variation of AES and DES (1024) with 1024-bit input block a 1024 bit key size compared with 128-bit in the original AES and DES algorithm. A complete implementation for the new implement was also presented. Time Taken to break AES algorithm by a brute force program increases exponentially with increase in the key lengths. The five different key sizes possible i.e., 128 bit, 192 bits 256 bits, 512 bits and 1024 bit keys are considered the simulation results time for different key size for AES. It is higher key size leads to clear change in the key rounds and time consumption. The algorithm can be implemented securely and efficiently in a wide variety of platforms and applications. After comparing the implementation results, The larger key size makes the algorithm more secure, and the larger input block increases the throughput. It increase in area makes the proposed algorithm ideal applications in which high level of security and high throughput, Computation Cost, security and Performance analysis for AES and DES Algorithm. The performance of this algorithms evaluated based on parameters. It is shown that both algorithms consume different times at different machines. Different machines take different times for same algorithm over same data packet and different speed. The presented results showed that AES has a better security against the brute force attack than DES; AES is more secure as compare to DES.

REFERENCES:

1. M.E. Hellman, *"DES will be totally Insecure within Ten Years"*, lEEE Spectrum, Vo1.16, N0.7, pp32 - 39, July 1979.
2. Alani, M.M., *"A DES96 - Improved DES Security"*, 7th International Multi-Conference on Systems, Signals and Devices, Amman, 27-30 June 2010.
3. Manikandan. G, Rajendiran. P, Chakarapani. K, Krishnan. G, Sundarganesh. G, *"A Modified Crypto Scheme for Enhancing Data Security"*, Journal of Theoretical and Advanced Information Technology, Jan 2012.
4. Shah Kruti R., BhavikaGambhava, "*New Approach of Data Encryption Standard Algorithm"*, International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231-2307, Volume-2, Issue-1, March 2012.
5. A. Nadeem, "*A Performance Comparison of Data Encryption Algorithms"*, IEEE Information and Communication Technologies, pp.84-89, 2006.

6. Sannbansahmoud, WilliamElamary and shadiAdudalifa, " *Enhanecement the Security Against Modern Attacks by Using Variable Key Block Cipher*", International Arab Journal of – e-technologies, Vol 3, No:1, 2013.
7. AviKak*"AES: The Advanced Encryption Standard Lecture Notes on "Computer and Network Security"*May 1, 2015.
8. Sumitra, *"Comparative Analysis of AES and DES Security Alogrithm",* International Journal of Scientific and Research Publications, Vol3, Issue 1, January 2013.
9. Rishabh Jain, Rahul Jejurkar, Shrikrishna Chopad3, Someshwar Vaidy4, Mahesh Sana*, " AES Algorithm using 512 Bit Key Implementation for Secure Communication",* International journal of innovative Research in Computer and Communication Engineering, Vol. 2 Issue 3, March 2014.
10. Nikolas Bardis, Konstantinos Ntaikos*, "Design of a Secure Chat Application based On AES Cryptographic Algorithm and Key Management",* JIRETVol.3, no. 2, 2011.