Vol. 5 No. 2 October 2017 ISSN: 2321-788X UGC Approval No: 43960 Impact Factor: 2.114

HIGH-LEVEL SECURITY ENHANCED 1024-BIT KEY AGGREGATE USING AES AND DES ALGORITHM

Article Particulars

Received: 24.10.2017 Accepted: 28.10.2017 Published: 30.10.2017

Prof.D.KIRUBASANKARI, M.Sc., M.Phil., B.Ed.
Head, PG and Research Department of Computer Science,
Sri Bharathi Women's Arts and Science College,
Kunnathur, Arni, TV Malai Dt. Tamil Nadu, India

S.JAYABHARATHI
M.Phil., Research Scholar, PG, and Research Department of Computer Science,
Sri Bharathi Women's Arts and Science College,
Kunnathur, Arni, TV Malai Dt. 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. more amount of information are so much private, but some are less private due to this the attacker or the hackers are taking messages. DES is now considered to be insecure for many applications because the 56-bit key size being too small is possible to brute-force infinite time on a modern processor. The AES algorithm was believed to provide more security than the DES. The AES algorithm was designed to have resistance against all known attacks, speed and code compactness on a wide range of platforms and design simplicity. 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) is implemented using NS2 software to make a comparison by parameters like speed, block size, and key size.

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

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 plaintext into cipher text is called Encryption and cipher text into plain text is called Decryption. Secret key cryptography, a single key is used for both encryption and decryption. The sender uses the key (or some set of rules) to encrypt the plaintext and sends the ciphertext 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 kept it 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 the two keys added to the encryption slowed the process. The encryption process starts by converting a 128-bit key into a binary value. The result is increased to 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 eight rounds. The aim is to present the AES 1024 bit when used in a 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 AES algorithm but having a slight difference is that the plaintext size and key size using the input of 512 bit instead of 128 bit has the impact on the whole algorithm.

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 ciphertext. 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 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. We have developed an application on Android platform which allows the user to encrypt the messages before it is transmitted over the network. We have used the Advanced Encryption Standards algorithm for encryption and decryption of the data. This application can run on any device which works on Android platform. This provides a secure, fast, and strong encryption of the data. As a conclusion, the requirements for speed and compactness were 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, called *algebraic attacks* exploit the intrinsic algebraic structure of a cipher. Advanced Encryption Standard (AES) algorithm is not only for security but also for great speed. Both hardware and software implementation is faster still. Encrypts data blocks of 128 bits in 10, 12 and 14 round depending on key size. It can be implemented on various platforms especially in small devices. It is carefully tested for many security applications. 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.

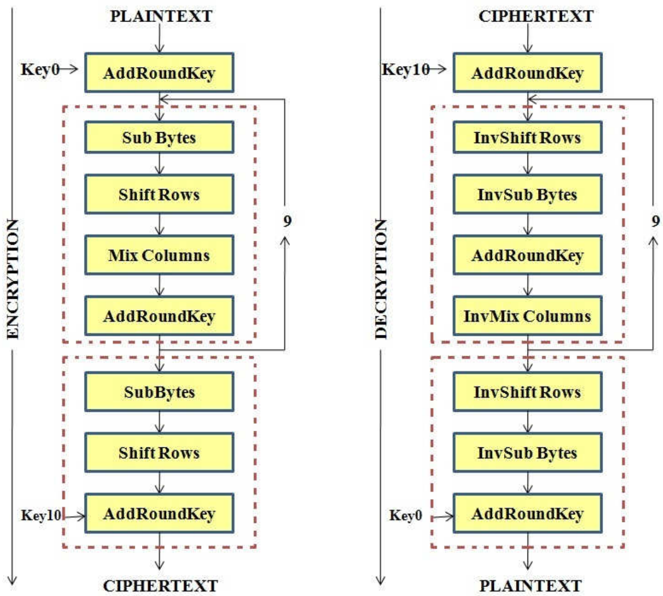

Fig 1.1: AES Structure

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 will succeed in a few hours. DES is up to 56bits 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 ciphertext 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.

System Implementation

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 the two keys added to the encryption slowed the process. The encryption process starts by converting a 128-bit key into a binary value. The result is increased 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.

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 ciphertext. Plaintext, used in the context of cryptography, is the name commonly given to the body of a message before it is encrypted.

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 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. To calculate the inputs to the S- and P-box arrays, portions of the data are XORed with portions of the key. One of the 128-bit halves of the 1024-bit data and the 1024-bit key is used. Because the key is longer than the data half, the 128-bit data half is sent through an expansion permutation which rearranges its bits, repeating certain bits, to form a 1024-bit product. Similarly, the 128-bit key undergoes a compression permutation which rearranges its bits, discarding certain bits, to form a 128-bit product. The S-and P-box look-ups and the calculations upon the key and data which generate the inputs to these table look-ups constitute a single round of DES.

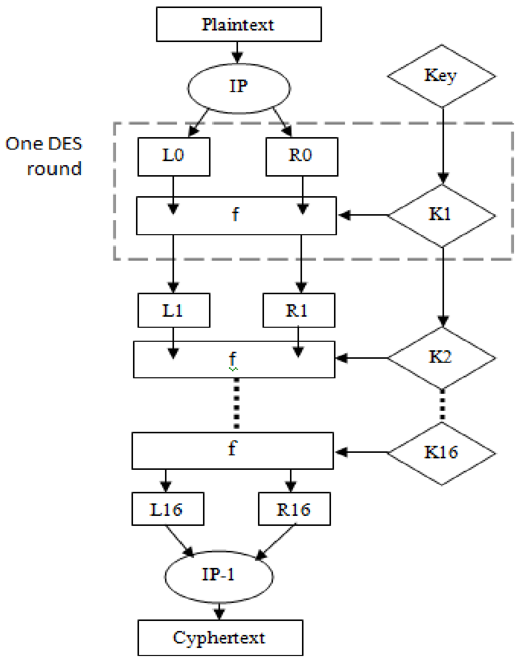

Fig 1.2: DES Core Algorithm

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 step or the E-step.

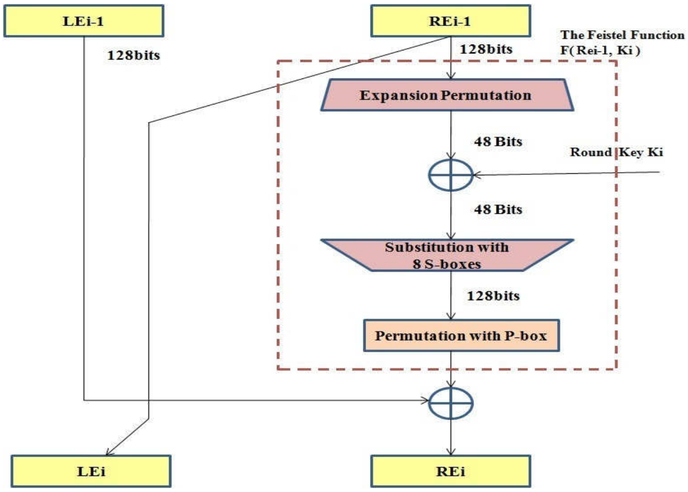

Fig 1.3: DES Implementation

DES works on 1024-bit plaintext blocks. The key is longer and consists of 128 bits. The 1024 bit input plaintext block is divided into four portions of plain text (each of size 16 bits), say P1 to P4. Thus P1 to P4 are inputted to the first round of algorithm, and there are eight such rounds. The output of first round is input to the second round. The output of first round is input to the second round. Similarly, the output of round second round is input to third round, and so on. In each round six subkeys are generated from the original key. Each of the subkeys 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 subkeys (K49 to K52). The final output produced is the output produced by output transformation, which is four blocks of ciphertext C1 to C4. These are combined to form final 64-bit ciphertext block. In this algorithm, addition is done modulo 216 and multiplication is done modulo 216+1. Idea employs the technique of key shifting.

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. How this is done will be explained later.

Step 5: The 48 bits of the expanded output produced by the E-step is 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. The substitution is carried out with an S-box.

Step7: Note that the goal of the substitution step implemented by the S-box is to introduce diffusion in the generation of the output from the input. Diffusion means that each plaintext bit must affect as many ciphertext bits as possible.

Step 8: The strategy used for creating the different round keys from the main key is meant to introduce confusion into the encryption process.

Step 9: Confusion in this context means that the relationship between the encryption key and the ciphertext 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 ciphertext block.

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. The last operation will be the bitwise XORing with the round key expanded using the key expansion process. The output at of the 10^{th} round will be the 1024-bit encrypted message.

Advanced Encryption Standard Algorithm for Encryption

```
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
```

In this algorithm A shows the input plaintext which is the size of 128 bit, K shows key which is either 512bit or 1024 bit long, C shows the ciphertext which is of the length 128 bit long. The number of rounds depends upon the size of the key which varies from 10 to 14.

Substitution of Bytes

The 1024 bits input plaintext is organized in the array of 128 bytes and is substituted by values obtained from substitution boxes. This is done to achieve more security according to diffusion-confusion 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
```

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 what's called a Galois Field.

```
Mix Columns (byte state [4, Nb])
 begin
      byte t [4] for c = 0 step 1 to Nb – 1
          for r = 0 step 1 to 3t[r] = state[r, c]
     end for
For r = 0 step 1 to 3state[r, c] = FFmul (0x02, t[r]) xor
           FFmul (0x03, t[(r + 1) mod 4]) xor
              t [(r + 2) mod 4] xor t[(r + 3) mod 4]
           end for
```

```
    end for
End
```

Add Round Key

To make the relationship between the key and the ciphertext more complicated and to satisfy the confusion principle, the Add Round Key operation is performed. This addition step takes the resulting data matrix from the previous step and performs on it 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 are arranged in a square matrix of eight columns where each column has 16 bytes.

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.

W (I) = W (I - 8) XOR W (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 (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
                while
        i = Nkwhile (i < Nb * (Nr + 1))
                word temp = w [i- 1]
        if (i mod Nk = 0)
temp = Sub Word (Rot Word (temp)) xor Rcon [i / Nk]
       else if ((Nk = 8) and (i mod Nk = 4))
        temp  = Sub Word (temp) end if w[i] = w [i - Nk]
                xor tempi = i + 1
        end while
End
```

Evaluation Result

The Data Encryption Standard algorithm and Advanced Encryption Algorithm is implemented using NS2 simulator. Cygwin is free software that provides Unix-like environment and software toolset to users of any modern x86 32-bit and 64-bit versions of MS-Windows (XP with SP3/Server 20xx/Vista/7/8) and (using older versions of Cygwin) some obsolete versions (95/98/ME/NT/2000/XP without SP3) as well. Cygwin consists of a Unix system call emulation library, cygwin1.dll. With Cygwin installed, users have access to many standard UNIX utilities. They can be used from one of the provided shells such as bash or the Windows Command Prompt.

Simulation Tools

This section illustrates seven main-stream simulation tools used in WSNs: NS-2, TOSSIM, EmStar, OMNeT++, J-Sim, ATEMU, and Avrora, and analyzes the advantage and disadvantage of each simulation tool.

NS-2

NS2 is the abbreviation of Network simulator version two, which first been developed by 1989 using the REAL network simulator. Now, NS-2 is supported by Defense Advanced Research Projects Agency and National Science Foundation. NS-2 is a discrete event network simulator built in Object-Oriented extension of Tool Command Language and C++. People can run NS-2 simulator on Linux Operating Systems or on Cygwin, which is a Unix-like environment and command-line interface running on Windows. NS-2 is a popular non-specific network simulator can use in both wire and wireless area. This simulator is open source and provides the online document.

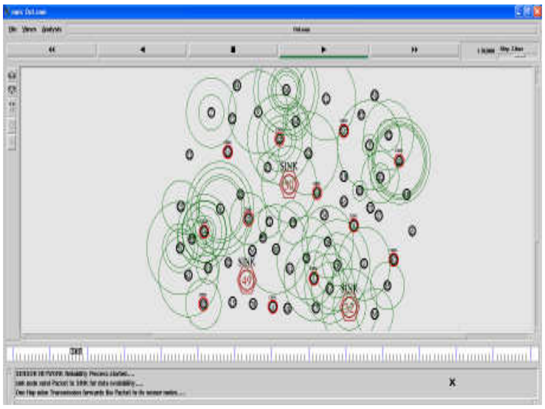

Fig 1.4: DES Vs AES NODE CREATION

Performance Evaluation

Following Graph will prove that AES and DES Selection will provide the better performance compared to the AES. With the Help of the NS-2 Simulator tool to prove the Metrics like,

- Throughput
- Performance Analysis
- Security
- Computation Cost

Throughput

This graph has shown the time being utilized by AES and DES security algorithms. Here a machine with a different configuration is used to carry out the time analysis between these algorithms. Some files with different sizes were fed into the simulation test.

The result obtained after simulation are represented in the form of the graph below.

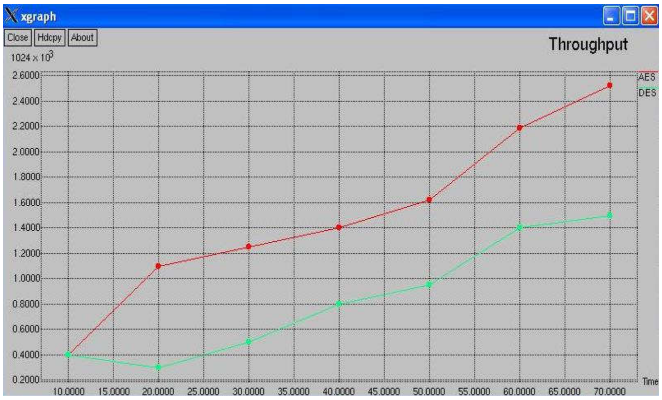

Fig 1.5: Throughput

Key Size	AES	DES
32	400	400
64	1100	260
128	1320	630
196	1400	800
256	1600	950
512	2200	1400
1024	2550	1560

Table: 1.1 AES Vs. DES Throughput Values

Performance

AES is the successor of DES as the standard symmetric encryption algorithm. AES uses keys of 1024 bits, although, 128-bit keys provide sufficient strength today. It uses 128-bit blocks and is efficient in both software and hardware implementations. To evaluate the performance of compared DES and AES algorithms. In the section showed the simulation results obtained by running the four compared encryption algorithms using different Modes. Different load has been used to determine the processing power and performance of the compared algorithms.

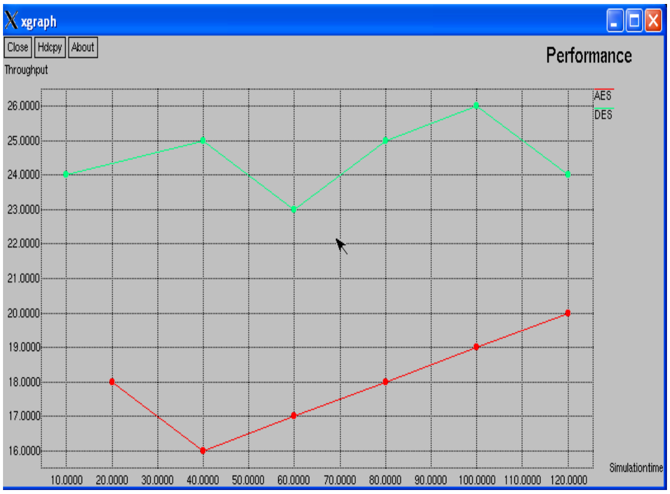

Fig 1.6: Performances of DES and AES

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

Table: 1.2 Performances of AES Vs DES

SECURITY

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

Fig 1.7: 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: 1.3 Security Comparisons 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.

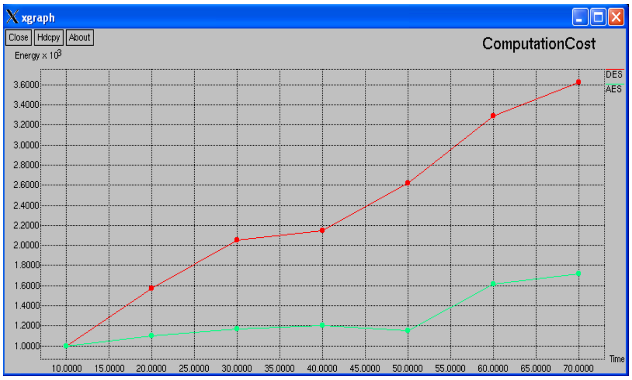

Fig 1.8 Computation Cost

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: 1.4 Computation Costs 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: 1.5 Comparisons 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 the 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 are 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 a 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 increases in the 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 the 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 Insecure within Ten Years,"* IEEE 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, Chakrapani. 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, " *Enhancement 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 Algorithm,"* 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.