DELHI TECHNOLOGICAL UNIVERSITY

Department of Computer Science and Engineering

DISCRETE MATHEMATICS

PROJECT REPORT ON

“ Increasing Security of 3DES ”

Subject Prof. :

Miss Tanya Malhotra

Submitted By

AJAY PAL 2K19/CO/036

ABHISHEK SOREN 2K19/CO/025

Candidate's Declaration

We, Abhishek Soren 2K19/CO/025 and Ajay Pal 2K19/CO/036, students of B.Tech (COE), hereby declare that the project Dissertation titled "Increasing Security of Triple Data Encryption Standard or 3-DES" which is submitted by us to the Department of Electronics, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology, is made by us and not copied from any source without proper citation. This work is completely performed and made by us with guidance of Prof. Tanya Malhotra.

Place: Delhi	Abhishek Soren 2K19/CO/025
Date: 20 Nov'20	Ajay Pal 2K19/CO/036

Certificate

I hereby certify that the Project Dissertation titled "Increasing Security of Triple Data Encryption Standard or 3-DES" which is submitted by Abhishek Soren 2K19/CO/025 and Ajay Pal 2K19/CO/036 to Department of Electronics, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology, is a record of the project work carried out by the students under my supervision for the subject of Discrete Mathematics for the academic year 2020-21.

Place: Delhi

Date: 20 Nov' 20

Ms. Tanya Malhotra

Supervisor

Acknowledgement

We deem it pleasure to acknowledge our sense of gratitude to our project guide Prof. Tanya Malhotra under whom we have carried out the project work. Her incisive and objective guidance and timely advice encouraged us with the constant flow of energy to continue the work.

Finally, we must say that no height is ever achieved without some sacrifices made at some end and it is here where we owe our special debt to our parents and our friends for showing their generous love and care throughout the entire period of time.

AIM

Increasing ENCRYPTION and DECRYPTION layer in

3DES

to INCREASE its security.

In this Report, we have implemented a method to increase the security of the Triple Data Encryption Standard.

Abstract

Security is playing a very important and crucial role in the field of network communication system and Internet. Triple Data encryption standard (3DES) is a private key cryptography system that provides the security in communication system but now a days the advancement in the computational power the 3DES seems to be weak against the brute force attacks. To improve the security of 3DES algorithm the transposition technique is added before the 3DES algorithm to perform its process. By using an Enhanced 3DES algorithm, the security has been improved which is very crucial in the communication and field of Internet. If the transposition technique is used before the original 3DES algorithm then the intruder required first to break the original 3DES algorithm and then transposition technique. So, the security is approximately double as compared to a simple 3DES algorithm.

Contents

Contents

INTRODUCTION

What is cryptography?

Cryptography is associated with the process of converting ordinary plain text into unintelligible text and vice-versa. It is a method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it. Cryptography not only protects data from theft or alteration, but can also be used for user authentication.

Earlier cryptography was effectively synonymous with encryption but nowadays cryptography is mainly based on mathematical theory and computer science practice.

Modern cryptography concerns with:

1. Confidentiality - Information cannot be understood by anyone
2. Integrity - Information cannot be altered.
3. Non-repudiation - Sender cannot deny his/her intentions in the transmission of the information at a later stage
4. Authentication - Sender and receiver can confirm each

Cryptography is used in many applications like banking transactions cards, computer passwords, and e- commerce transactions.

Three types of cryptographic techniques used in general.

1. Symmetric-key cryptography

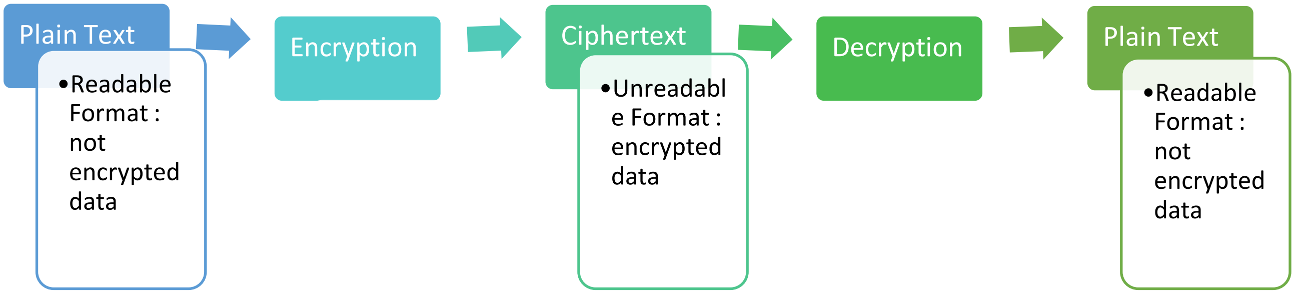
2. Hash functions.

3. Public-key cryptography

Symmetric-key Cryptography: Both the sender and receiver share a single key. The sender uses this key to encrypt plaintext and send the cipher text to the receiver. On the other side the receiver applies the same key to decrypt the message and recover the plain text.

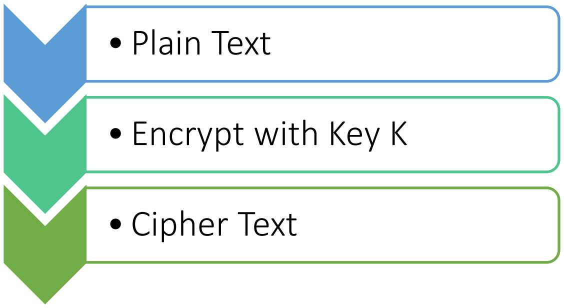
Public-Key Cryptography: This is the most revolutionary concept in the last 300-400 years. In Public-Key Cryptography two related keys (public and private key) are used. Public key may be freely distributed, while its paired private key, remains a secret. The public key is used for encryption and for decryption private key is used.

Hash Functions: No key is used in this algorithm. A fixed-length hash value is computed as per the plain text that makes it impossible for the contents of the plain text to be recovered. Hash functions are also used by many operating systems to encrypt passwords.


Plain Text
•Readable Format : not encrypted data
Encryption
Ciphertext
•Unreadable Format : encrypted data
Decryption
Plain Text
•Readable Format : not encrypted data

Data Encryption Standard

Data Encryption Standard or DES is a symmetric block cipher which takes the input of 64-bit plain text along with 64-bit key and process it, to generate the 64-bit ciphertext.



- Plain Text
- Encrypt with Key K
- Cipher Text

Steps involved in DES

- For first step, the 64 bit plain text block is handed over to an initial Permutation (IP) function.
- The initial permutation is performed on plain text.
- Next the initial permutation (IP) produces two halves of the permuted block, say Left Plain Text (LPT) and Right Plain Text (RPT).
- Now each LPT and RPT to go through 16 rounds of encryption process.
- In the end, LPT and RPT are rejoined and a Final Permutation (FP) is performed on the combined block
- The result of this process produces 64 bit cipher text.
- For decryption, the same above algorithm is followed but in the reverse order.

Triple Data Encryption Standard

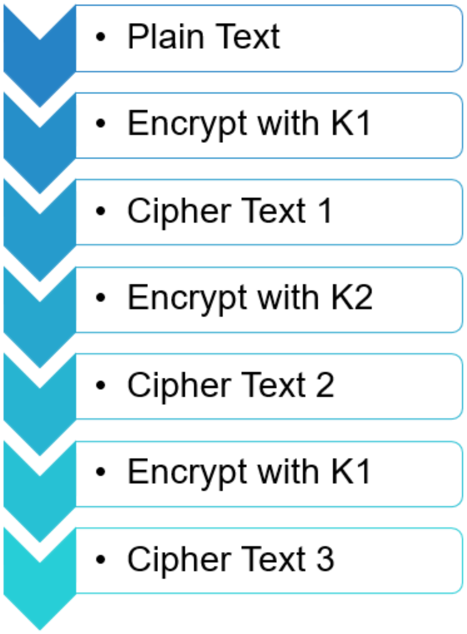
Triple DES is simply DES but performed three times on the plain text.

It comes in two flavors :

One that uses three keys and other that uses two keys.

- **THREE KEYS -** The plain text block P is first encrypted with a key K1, then encrypted with second key K2, and finally with third key K3, where K1, K2 and K3 are different from each other. To decrypt the cipher text C and obtain the plain text, we need to perform the operation

P= DK3 (DK2 (DK1(C)))

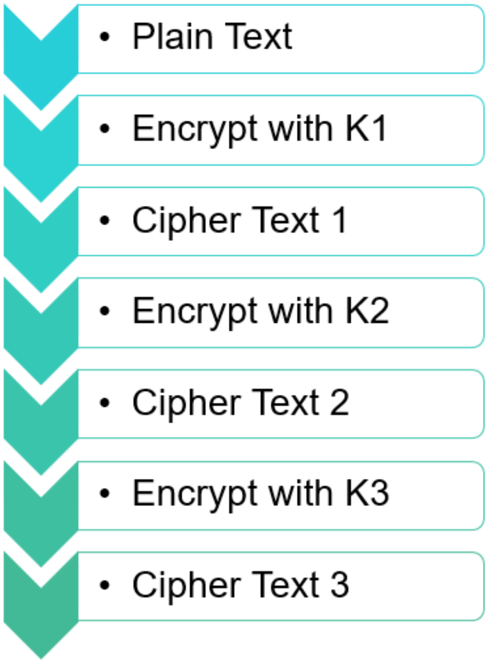


- Plain Text
- Encrypt with K1
- Cipher Text 1
- Encrypt with K2
- Cipher Text 2
- Encrypt with K1
- Cipher Text 3

Encryption process Triple DES with three keys K1 and K2

- **TWO KEYS -** The plain text block P is first encrypted with a key K1, then encrypted with second key K2, and finally again with K1, where K1 and K2 are different from each other. To decrypt the cipher text C and obtain the plain text, we need to perform the operation

P= DK1 (DK2 (DK1(C)))



- Plain Text
- Encrypt with K1
- Cipher Text 1
- Encrypt with K2
- Cipher Text 2
- Encrypt with K3
- Cipher Text 3

Encryption process Triple DES with three keys K1, K2 and K3

Advanced Encryption Standard

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the **Advanced Encryption Standard** (AES). It is found at least six time faster than triple DES.

A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.

The features of AES are as follows −

- Symmetric key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger and faster than Triple-DES
- Provide full specification and design details
- Software implementable in C and Java

AES is based on the Rijndael algorithm, created by Joan Daemen and Vincent Rijmen, which is a combination of a strong algorithm with a strong key. The Rijndael block cipher can use different block and key lengths, such as 128, 192, and 256 bit. This versatility can produce faster and more secure symmetric block ciphers. Another algorithm which might be considered as an alternative to the Rijndael block cypher is the Twofish algorithm, which can use blocks of 128 bits with keys up to 256 bits. The Rijndael algorithm’s combination of security, performance, efficiency, implementability, and flexibility made it an appropriate selection for AES.

Comparison

Advance Encryption Standard (AES) and Triple DES (TDES or 3DES) are commonly used block ciphers. Whether you choose AES or 3DES depend on your needs. In this section we would like to highlight their differences in terms of security and performance .Since 3DES is based on DES algorithm, it will talk about DES first. DES performs lots of bit manipulation in substitution and permutation boxes in each of 16 rounds. For example, switching bit 30 with 16 is much simpler in hardware than software. DES encrypts data in 64 bit block size and uses effectively a 56 bit key. 56 bit key space amounts to approximately 72 quadrillion possibilities. Even though it seems large but according to today's computing power it is not sufficient and vulnerable to brute force attack. Therefore, DES could not keep up with advancement in technology and it is no longer appropriate for security. Because DES was widely used at that time, the quick solution was to introduce 3DES which is secure enough for most purposes today.

3DES is a construction of applying DES three times in sequence. 3DES with three different keys (K1, K2 and K3) has effective key length is 168 bits. Another variation is called two-key. 3DES reduces the effective key size to 112 bits which is less secure. Two-key 3DES is widely used in electronic payments industry. 3DES takes three times as much CPU power than compare with its predecessor which is significant performance hit.

AES outperforms 3DES both in software and in hardware. The Rijndael algorithm has been selected as the Advance Encryption Standard (AES) to replace 3DES. AES is modified version of Rijndael algorithm. Advanced Encryption Standard evaluation criteria among others was

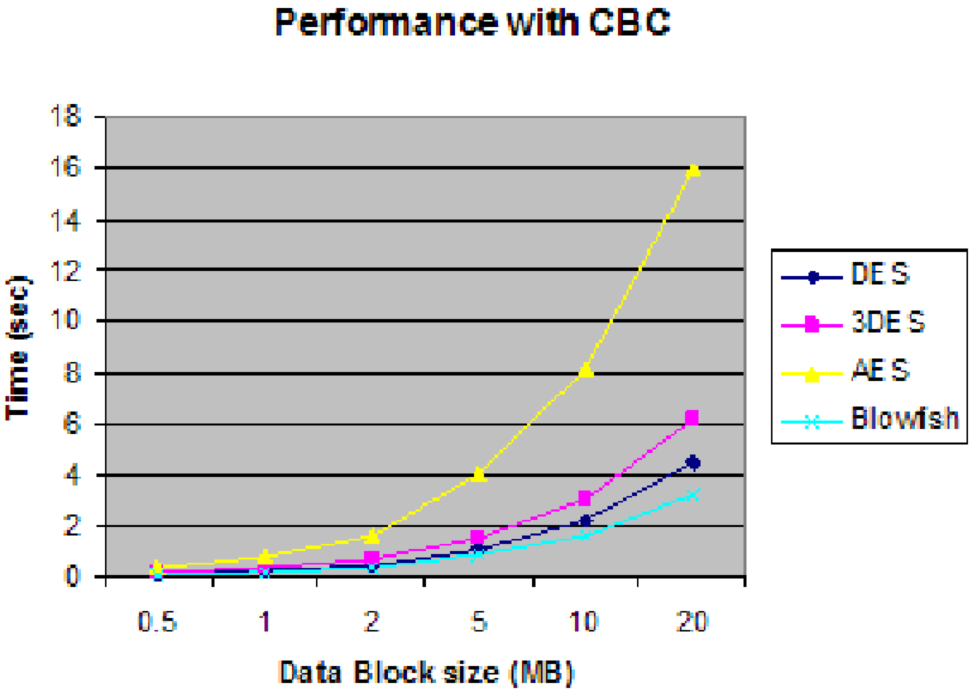
- Security
- Software & Hardware performance
- Suitability in restricted-space environments
- Resistance to power analysis and other implementation attacks.

Comparison between AES, 3DES and 3DES

Factors	AES	3DES	DES
Key Length	128, 192, or 256 bits	(K1, K2 and K3) 168 bits (K1 and K3 is same) 112 bits	56 bits
Cipher Type	Symmetric Block Cipher	Symmetric Block Cipher	Symmetric Block Cipher
Block Size	128, 192, or 256 bits	64 bits	64 bits
Developed	2000	1978	1977
Cryptanalysis Resistance	Strong against differential, truncated differential, linear, interpolation and square attacks	Vulnerable to differential, Brute Force attacker could analyze plain text using differential cryptanalysis	Vulnerable to differential and linear cryptanalysis; weak substitution tables
Security	Considered secure	One only weakness, which is Exit in DES	Proven Inadequate
Possible Keys	2^{128}, 2^{192} or 2^{256}	2^{112} or 2^{168}	2^{56}
Possible ASCII printable character keys	95^{16}, 95^{24} or 95^{32}	95^{14} or 95^{21}	95^{7}
Time recorded to check all possible keys at 50 billion keys per second	For a 128 bit key : 5×10^{21} years	For a 112 bit key : 800 days	For a 56 bit key : 400 days

Rijndael was submitted by Joan Daemen and Vincent Rijmen. When considered together Rijndael's combination of security, performance, efficiency, implement ability, and flexibility made it an appropriate selection for the AES. By design AES is faster in software and works efficiently in hardware. It works fast even on small devices such as smart phones; smart cards etc. AES provides more security due to larger block size and longer keys. AES uses 128-bit fixed block size and works with 128, 192 and 256-bit keys. Rigndael algorithm in general is flexible enough to work with key and block size of any multiple of 32 bits with minimum of128 bits and maximum of 256 bits. AES is replacement for 3DES according to NIST both ciphers will coexist until the year2030 allowing for gradual transition to AES. Even though AES has theoretical advantage over 3DES for speed and efficiency in some hardware implementation 3DES may be faster where support for 3DES is mature.

The following graph shows the time taken to encrypt various numbers of 16-byte blocks of data using the algorithms mentioned.



Comparison of encryption times for various common symmetric encryption algorithms

TRANSPOSITION TECHNIQUES

Transposition technique is a cryptographic technique that converts the plain text to cipher text by performing permutations on the plain text i.e. change the position of each character of plain text for each round. It includes various techniques like Rail Fence technique, simple columnar transposition technique with multiple rounds, Vernam cipher, and book Cipher to encrypt the plain text in a secure way.

Enhanced Rail Fence

The **rail fence cipher** (also called a zigzag cipher) is a form of transposition cipher. It derives its name from the way in which it is encoded.

In the rail fence cipher, the plain text is written downwards and diagonally on successive "rails" of an imaginary fence, then moving up when the bottom rail is reached. When the top rail is reached, the message is written downwards again until the whole plaintext is written out.

To increase the efficiency of rail fence transposition ,we are using a variable length key which store the no of rounds ,no of rows and pair of indexes to be swapped which increases layer in the rail fence transposition technique and hence make it more secure.

Algorithm [ENCRYPTION]

- Generate a key of any size.
- Some specific digits (in our case->unit digit) are used to store the no of rows on which the transposition is to be applied.
- Some digits (in our case ->tenth digit) store the no of rounds for which the key is going to be transposed.
- Remaining digits store the pairs of indexes of character to be swapped.

- the plain-text is written downwards and diagonally on successive rails of an imaginary fence.
- When we reach the bottom rail, we traverse upwards moving diagonally, after reaching the top rail, the direction is changed again. Thus, the alphabets of the message are written in a zig-zag manner.
- After each alphabet has been written, the individual rows are combined to obtain the cipher-text.

```
// Function Snippet in C++

string encryptRail(string text,int key)
{
    int len=text.length();
    char** rail = new char*[key];
    for(int i = 0; i < key; ++i)
     {
        rail[i] = new char[len];
    }
    for(int i=0;i<key;i++)
     {
     for(int j=0;j<len;j++)
          {
          rail[i][j]=NULL;
          }
     }
    bool decline=true;
    bool incline=false;
    int j=0;
     for(int i=0;i<len;i++)
     {
     rail[j][i]=text[i];
     if(j==(key-1))
          {
          decline=false;
          incline=true;
     }

     if(j==0)
          {
          incline=false;
          decline=true;
     }

     if(decline)
          {
          j++;
     }

     if(incline)
          {
          j--;
     }
     }

     string result;
     for(int i=0;i<key;i++)
     {
     for(int j=0;j<len;j++)
          {
            if(rail[i][j]!=NULL)
               {
               result.push_back(rail[i][j]);
          }
     }
     }

     return result;

}
```

Algorithm [DECRYPTION]

- The key is passed form the storage stored during encryption process.
- Some specific digits (in our case->unit digit) are used to store the no of rows on which the transposition was applied.
- Some digits (in our case ->tenth digit) store the no of rounds for which the text is going was transposed.
- rail matrix can be constructed accordingly. Once we've got the matrix we can figure-out the spots where texts should be placed (using the same way of moving diagonally up and down alternatively).
- Then, we fill the cipher-text row wise. After filling it, we traverse the matrix in zig-zag manner to obtain the original text as many times as was encrypted.
- Now remaining digits in the key indicate swapped indexes so re-swap them.

```
// Function Snippet in C++

string decryptRail(string cypher,int key)
{

    int len=cypher.length();
    char** rail = new char*[key];
    for(int i = 0; i < key; ++i)
    {
        rail[i] = new char[len];
    }
    bool decline=true;
    bool incline=false;
    int j=0;
    for(int i=0;i<len;i++)
    {
        rail[j][i]='*';
        if(j==(key-1))
        {
            decline=false;
            incline=true;
        }
    if(j==0)
    {
        incline=false;
        decline=true;
    }

    if(decline)
    {
        j++;
    }

    if(incline)
    {
        j--;
    }
}

int out=0;
for(int i=0;i<key;i++)
{
    for(int j=0;j<len;j++)
    {
        if(rail[i][j]=='*')
        {
            rail[i][j]=cypher[out];
            out++;
        }
    }
}

string result;
decline=true;
incline=false;
j=0;

for(int i=0;i<len;i++)
{
    result.push_back(rail[j][i]);
```

```
            if(j==(key-1))
            {
                decline=false;
                incline=true;
            }

            if(j==0)
            {
                incline=false;
                decline=true;
            }

            if(decline)
            {
                j++;
            }

            if(incline)
            {
                j--;
            }
        }

        return result;

    }
```

Enhanced Simple Columnar Transposition (for Multiple Rounds)

In a columnar transposition, the message is written out in rows of a fixed length, and then read out again column by column, and the columns are chosen in some scrambled order. Both the width of the rows and the permutation of the columns are usually defined by a keyword.

The only function of key was for column interchange but to increase more stability we assign all character a special no so that on summing up the values corresponding to each index of keys give no of round

Algorithm [ENCRYPTION]

- Assign values to each character for a system
- Choose a random key
- The total of the values corresponding to each character of key is added to find the total no of round.
- The message is written out in rows of a fixed length, and then read out again column by column, and the columns are chosen in some scrambled order.
- Width of the rows and the permutation of the columns are usually defined by a keyword.
- For example, the word HACK is of length 4 (so the rows are of length 4), and the permutation is defined by the alphabetical order of the letters in the keyword. In this case, the order would be “3 1 2 4”.
- Any spare spaces are filled with nulls or left blank or placed by a character (Example: _).
- Finally, the message is read off in columns, in the order specified by the keyword.
- Encryption process is repeated for total no of rounds.

```
// Function Snippet in C++
string encryptColumnar(string text,string key)
{
     int *keyPos= new int[key.length()];
     for(int i=0;i<key.length();i++)
     {
     keyPos[i]=i;
     }
     sort(keyPos,key);
     for(int i=0;i<key.length();i++)
     {
     keyMap[keyPos[i]]=i;
     }
     int r=(text.length()/key.length());
     if(text.length()%key.length())
     {
     r+=1;
     }
     char matrix[r][key.length()];
     for(int i=0;i<r;i++)
     {
     for(int j=0;j<key.length();j++)
          {
          matrix[i][j]='_';
     }
     }
     int out=0;
     for(int i=0;i<r;i++)
     {
     for(int j=0;j<key.length();j++)
          {

               if(text[out]!='\0')
               {
```

```
               matrix[i][j]=text[out];
               out++;
               }

     }
     }
     for(int i=0;i<r;i++)
     {
     for(int j=0;j<key.length();j++)
          {
         cout<<matrix[i][j];
     }
     cout<<endl;
     }
     string result;
     for(int i=0;i<key.length();i++)
     {
     int index=keyMap[i];
     for(int j=0;j<r;j++)
          {
          result.push_back(matrix[j][index]);
     }
     }
     return result;
}
```

Algorithm [DECRYPTION]

- To decipher it, the recipient has to work out the column lengths by dividing the message length by the key length.
- Then, write the message out in columns again, then re-order the columns by reforming the key word.and repeat the process of total no of rounds.
- `// Function Snippet in C++`

```
// Function Snippet in C++

string decryptColumnar(string text,string key)
{
     int *keyPos= new int[key.length()];
     for(int i=0;i<key.length();i++)
     {
     keyPos[i]=i;
     }
     sort(keyPos,key);
     for(int i=0;i<key.length();i++)
     {
     keyMap[keyPos[i]]=i;
     }
     int r=(text.length()/key.length());
     if(text.length()%key.length())
     {
          r+=1;
     }
     char matrix[r][key.length()];
     for(int i=0;i<r;i++)
     {
     for(int j=0;j<key.length();j++)
          {
          matrix[i][j]='_';
     }
     }
     int out=0;
     for(int i=0;i<key.length();i++)
     {
        int index=keyMap[i];
     for(int j=0;j<r;j++)
          {

               if(text[out]!='\0')
```

```
            {
            matrix[j][index]=text[out];
            out++;
            }

    }
    }
    for(int i=0;i<r;i++)
    {
    for(int j=0;j<key.length();j++)
        {
        cout<<matrix[i][j];
    }
    cout<<endl;
    }
    string result;
    for(int i=0;i<r;i++)
    {
    for(int j=0;j<key.length();j++)
        {

        result.push_back(matrix[i][j]);
        }
    }

    return result;
}
```

Vernam Cipher

A subset of Vernam cipher is called a one-time pad because it is implemented using a random set of nonrepeating characters as an input cipher text.

Working of Algorithm

- Arrange all characters in the plain text as a number i.e. A = 0, B = 1, ….. Z = 25.
- Repeat the same procedure for all characters of the input ciphertext.
- Add each number corresponding to the plain text characters to the corresponding input cipher text character number.
- If the sum of the number is greater than 25, subtract 26 from it.
- Translate each number of the sum into the corresponding characters.
- The output of step 5 will be a cipher text.

In Vernam cipher, once the input cipher text is used, it will never be used for any other message, hence it is suitable only for short messages.

```
// Function Snippet in C++
string encryptVerman(string text,string key)
{
    int *result=new int[text.length()];
    for(int i=0;i<text.length();i++)
    {
    result[i]=(keyCode[text[i]]+keyCode[key[i]]);
    if(result[i]>=26)
        {
        result[i]=result[i]%26;
    }
    }
    string ans;
    for(int i=0;i<text.length();i++)
    {
    ans.push_back(codeKey[result[i]]);
    }
    return ans;
}
```

Book Cipher

The book cipher or the running key cipher works on the basic principle of one-time pad cipher. In onetime pad cipher the key is taken as long as the plain text and is discarded after the use. Every time a new key is taken for a new message.

The improvement to the onetime pad in Book cipher is that the key or the onetime pad is taken from the book.

Working of Algorithm

- Convert the plain text in numeric form consider A=0, B=1, C=3 …, Z=25.
- Take an onetime pad or key from any of the books and convert it in the numeric form also. But the key must be as long as the length of plain text.
- Now add the numeric form of both plain text and key, each plain text letter with corresponding key text letter. If the addition of any plain text letter with corresponding key text letter is >26, then subtract it with 26.

Comparison Between Rail Fence and Columnar Transposition Techniques

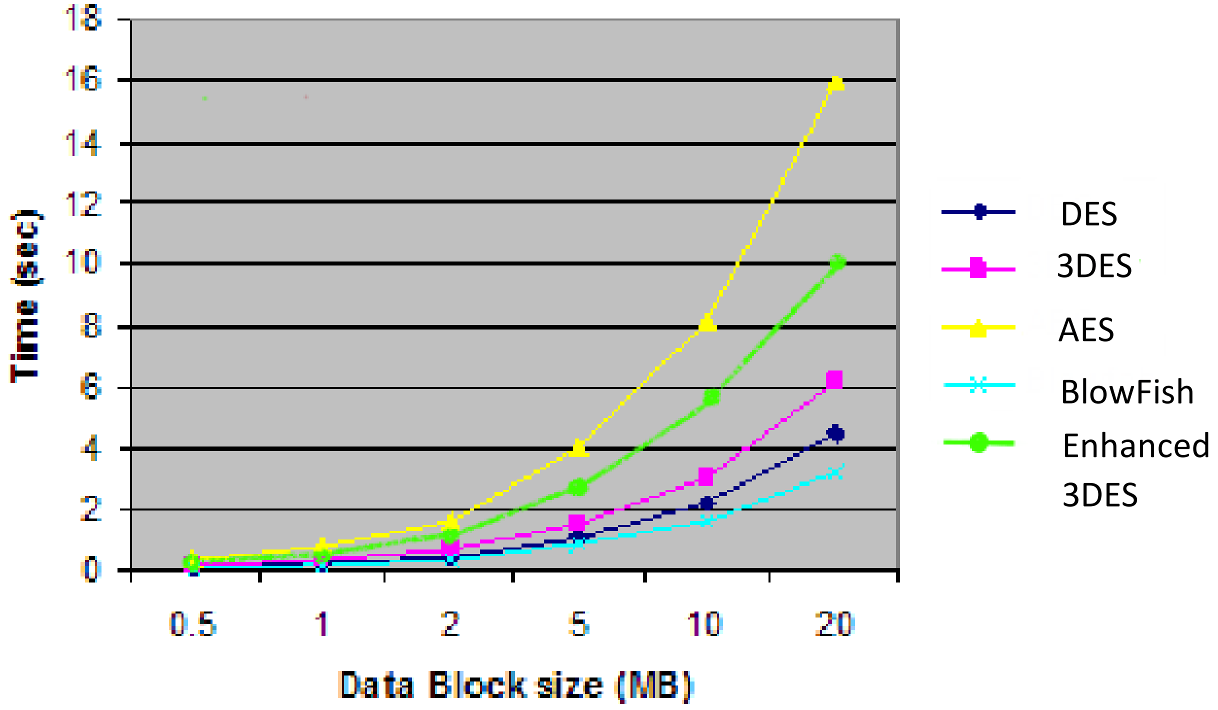
Parameter	Rail fence	Columnar
Key Type	Permutation	Permutation
Block Size	Variable Length (depth)	Equal to Key Size
Key Size	Depth Size is variable	Variable
Attack Type	Brute Force Attack	Frequency analysis attack
Algorithm Strength	Depth Size	Multiple encryptions are possible to a single message
Encryption and Decryption Process	Symmetric	Symmetric

Conclusion

In today's time, the security is playing a very important and powerful role in the field of networking, Internet and various communication system. The electronic communication system is used in banking, reservation system and marketing which required a very tight security system. The original 3DES implementation has some weaknesses, to overcome the most of weakness the Enhanced DES algorithm is designed.

Since we are using various enhanced transposition techniques which uses a key to encrypt and decrypt which increase the total key length and passible keys and the time required due to increase of layers in the encryption procedure and hence increase the total security.

By using the Enhanced 3DES algorithm the security is very tight and approximately impossible to crack and break the Enhanced 3DES algorithm.



Comparison of encryption times for various common symmetric encryption algorithms

References

- Dr. Kiramat ullah, Bibi Ayisha, Farrukh Irfan, Inaam Illahi - "Comparison of Various Encryption Algorithms for
- Securing Data", Zeeshan Tahir Pakistan Institute of Engineering and Applied Sciences (PIEAS)
- Hamdan.O.Alanazi, B.B.Zaidan, A.A.Zaidan, Hamid A.Jalab, M.Shabbir and Y. Al-Nabhani - "New Comparative
- Study Between DES, 3DES and AES within Nine Factors"
- https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm
- https://www.tutorialspoint.com/cryptography/data_encryption_standard.htm
- Sombir Singh, Sunil K. Maakar, Dr.Sudesh Kumar - "Enhancing the Security of DES Algorithm Using
- Transposition Cryptography Techniques", International Journal of Advanced Research in Computer Science and Software Engineering Research Paper
- Massoud Sokouti, Babak Sokouti and Saeid Pashazadeh - "An approach in improving transposition cipher system", Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran 2 Faculty of Engineering, Islamic Azad University -Tabriz Branch,Tabriz, Iran