# Data Security Enhancement using Different Encryption Algorithms

Project Report submitted in partial fulfilment of the requirement for the degree of

Bachelor of Technology.

in

**Computer Science & Engineering**

under the supervision of

Mr. Surjeet Singh

(Assistant Professor Grade-I)

By

Shivansh Sharma 151216

Arnav Sharma 151201

To

Jaypee University of Information and Technology Waknaghat,

Solan – 173234, Himachal Pradesh

# Candidate's Declaration

I hereby declare that the work presented in this report entitled **"Data Security Enhancement using Different Encryption Algorithms"** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an original record of my own work which has been carried out over a period from August 2018 May 2019 under the supervision of Mr**. Surjeet Singh**(Assistant Professor Grade-I, Computer Science & Engineering and Information Technology).

The subject matter included in this report is purely an original content.

Shivansh Sharma (151216)

Arnav Sharma (151201)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Mr. Surjeet Singh

Assistant Professor (Senior Grade)

Computer Science & Engineering and Information Technology

Dated: 15-05-2019

# Acknowledgment

Firstly, I would like to thank our supervisor Mr. Surjeet Singh at the Department of Computer Science & Engineering and Information Technology at Jaypee University of Information Technology, who helped me with the project work. I would like to thank him for providing utmost guidance throughout the time of development of the project.

I have grown both personally and academically from this experience and I am also very grateful for having had the opportunity to conduct this study.

Dated: 15-05-2019 Shivansh Sharma

Arnav Sharma

# Table of Contents

# Abbreviations and Symbols

| **Abbreviations** | **Full Forms** |
|---|---|
| DES | Data Encryption Standard |
| T-DES | Triple Data Encryption Standard |
| AES | Advanced Encryption Standard |
| KCQ | Keyed Communication in Quantum Noise |
| NIST | National Institute of Standards and Technology |
| S-BOX | Substitution Box |
| P-BOX | Permutation Box |

# List of Figures and Tables

# Abstract

In our project Data security enhancement using different Encryption Algorithms we will be looking into the world of cryptography and analyse several algorithms (DES, Triple-DES, Advanced Encryption Standard) and if possible will also optimise them in order to reduce their time complexity and space complexity. We will also be looking into existing cryptographic systems (Elgamel cryptosystem) and the emerging field of Quantum cryptography.

# CHAPTER 1-INTRODUCTION

## 1.1 Introduction

Communication has always been an integral part of our society and the Internet has had an overwhelming impact on our daily lives but the advancement in communication technology has also given rise to the security threats in communication. When we are talking of communication over a network we need to make sure that there is some sort of security for the information that is being communicated especially nowadays when the cyber crimes are at the pinnacle. Doesn’t matter whether you send personal information or a web transaction or confidential information being transmitted between organizations or the military, security is much needed. So we need Cryptography to protect our data or information which we send through or share with another user. Cryptography is a field of network security which hides the real information when it is transmitted through.

In this report we have examined various cryptographic algorithms, their short comings and how can they be optimised to be a better one.

## 1.2 Problem statement

This project concerns with the problem of providing secure communication between the users with minimum interference between them. The field of security is growing to be a necessity rather than a mere requirement these days. The user is more and more concern about the security of its data and information as cyber break-ins and crimes are growing day by day. This can be done using various cryptographic algorithms. We have discussed mainly Data encryption standard(DES), Triple DES and AES algorithms and also have tried to optimized them in order to decrease run time and performance.

## 1.3 Objective

1. To examine and optimize the DES algorithm.

2. To examine and optimize the Triple DES algorithm.

3. To examine and optimize the AES algorithm.

4. Also discussing their shortcomings.

## 1.4 Methodology

Dynamic system development model (DSDM)

We have used the DSDM model for our project.

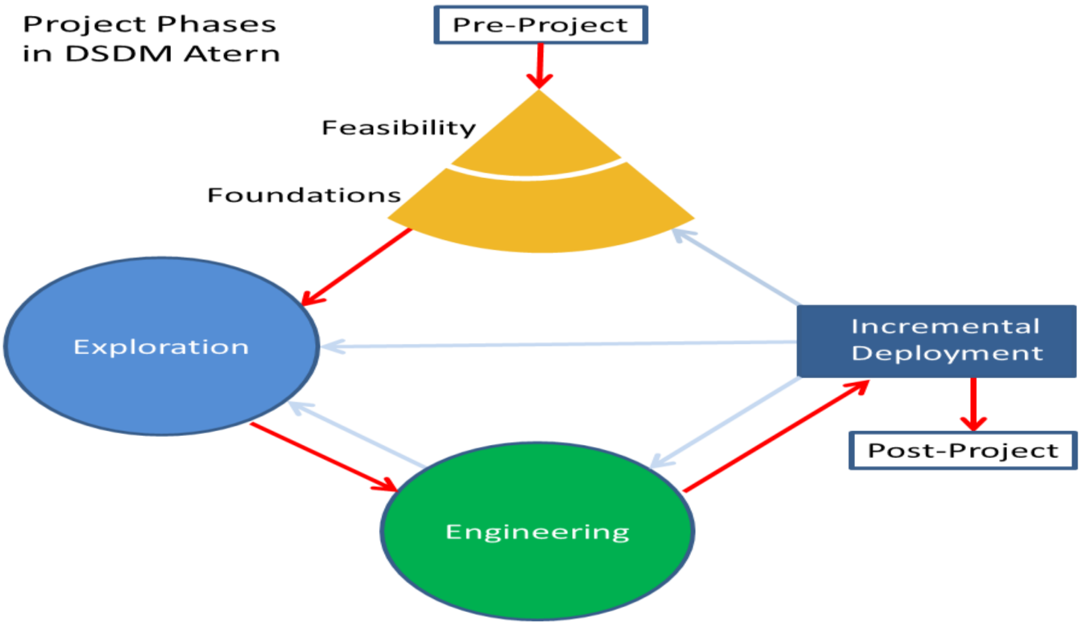


Figure 1.0 1

# CHAPTER 2 – LITERATURE REVIEW

## 2.1 Early Vs Modern Cryptography

Today the world of cryptography is changing drastically. Unlike in the early days when cryptography was used to conceal military and diplomatic information from the third party, though it has far reaching military implications. The field of cryptography is changing since its beginning from the 1970s.On January 1977 the National Bureau of standards (NBS) adopted a data encryption standard, which was considered a milestone in crypto research. On December 1980 the DES algorithm was adopted by the American National Standards Institute (ANSI).

## 2.2 – Theoretical Definitions

The theoretical definitions are as follows:

- Plain text— contains the original message that is to be carried safely without interception.
- Cipher text—the message that we obtain after coding the plain text is called as cipher text.
- Encryption—the process in which we convert or code the plain text to cipher.
- Decryption—the opposite of encryption is done in decryption i.e. conversion of cipher text to the original plain text.
- Deprecated – the usage of algorithm & its key length is allowed but the user should be ready to face risk.

## 2.3 Theoretical background

Cryptography has been in existing for centuries. Even in the middle ages emperors used few techniques to transfer the secret messages across to places. Cryptography is the art of concealing the information from those who are not authorized to see that particular information and hence providing security for the communication process between the sender and receiver. Cryptography includes encryption and decryption, cryptography key is used in an encryption algorithm in such a way that the process cannot be done backwards (decrypt).

## 2.4 Overall trends in Research

Since the 1970s, some big trends have emerged in the research regarding cryptography, the most common being that the secrecy is best for security and since the cryptography is based on shared secrets between the sender and the receiver. The mathematics behind a good cryptographic algorithm is always complex as more the complexity of the key the better will be the algorithm. Cryptography is also useful in providing data and network security. Many companies nowadays are using data encryption based on strong cryptographic techniques. Cryptography is the base for security solutions, which include: i) Digital signatures; ii) Personal banking type applications that encourage the use on secure socket layer & Transport layer security.

The embedded systems that are in use nowadays are of the same technologies that the corporate IT industry uses. These technologies include Ethernet, Transport layer protocol/information protocol etc. This suggests that the embedded systems all over the world are at risk from the same factors which have troubled the IT industry for years (Robinson). Hence the need of strong cryptography is the need of the hour to protect these embedded systems from various ranges of attacks especially those systems which have a heavy reliance on wireless technology.

## 2.5 Research on Quantum cryptography

The recent studies in the field of quantum cryptography, Hughes (2007); Baker (2005) report the developments in the new cryptographic protocol Keyed Communication in Quantum noise. KCQ involves the photons of light as its carriers. Many researchers agree that this KCQ protocol will provide greater security than any other cryptographic algorithm. The instantiation of this new KCQ into existing communications at the physical layer is called AlphEta protocol in the United States. This protocol works by transmitting thousands of photons in place of the logical bits at the physical level. These radiations states of multiple photons emitted by lasers are the information carriers in this type of network'

## 2.6 Cryptography Today

The cryptography industry is billion dollar industry today. The economics of the world and the defence sector of every country in the world depend upon it and cannot be possible without it. In today's world of cryptography one's identity in the digital world is managed by the federated identity management system which consists of some protocols and software components that are used in managing one's identity throughout thier identity lifecycle. Today with the rise in threats to sensitive data from attackers the encryption is a necessary tool to insure that corporate networks and individual's information is as secure as possible. Fagin et al. (2008) says there is a good progress taking place in this field to remove scepticism. The National Institute of Standards and Technology (NIST) has joined forces with the National Security Agency (NSA) to form the "Common Criteria" process known as the Common Criteria for Information Technology Security Evaluation 2005 which aims to restore the confidence in information-related security systems.

## 2.7 DES History & Introduction

DES was one of the first encryption algorithms that emerged in the world of cryptography. DES originated at IBM in 1977. DES has been generally accepted as the standard for symmetric encryption. Although DES has vulnerability but is still used in many applications and provides reasonable security. However now the DES is use in more sophisticated way such as Triple DES algorithm. After the year 1990 short length key of DES algorithm began to cause distress amongst the users but still the users were not fond of the idea of replacing DES as it takes considerable amount of resources to replace the encryption algorithms that are embedded in big systems. So the better way to move forward was to not completely disallow DES but to bring a change in the way of using it. This resulted in the formation of Triple-DES algorithm. Triple DES can be implemented with two key (2TDES) and three keys (3TDES).

Due to the design of Triple Data Encryption Standard algorithm it is possible to use the three key TDES hardware for executing single DES by setting all the three keys and providing the compatibility with the DES algorithm. The Triple DES systems are considered more secure than the single DES. On the other hand the time complexity of the Triple DES is much more than the single DES algorithm. Also according to a draft guidance published by NIST on July 19, 2018, the Triple DES is now officially being retired. The guidelines state that after a period of public consultation the Triple DES is deprecated for all new applications and is disallowed after 2023. The two key variant of the Triple DES was retired in 2015.

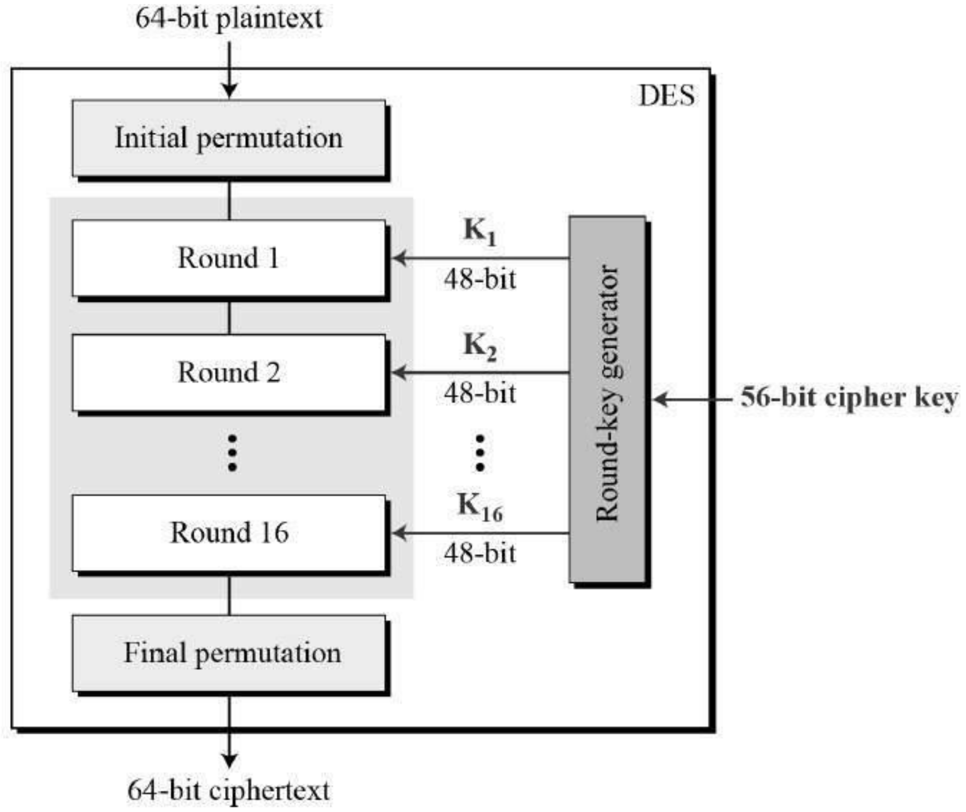
## 2.8 AES algorithm

The Rijndael (rain doll) is a family of block ciphers which was developed by Belgian researchers Vincent and Joen. At first it was submitted in an entry to NIST. The Belgians won the competition and the Rijndael was officially chosen to be selected as Advanced Encryption Standard. The AES comes in three variants having different key sizes as 128, 192, 256 bits.

# CHAPTER 3 – SYSTEM DEVELOPMENT

## 3.1 DES System

DES is a symmetric block cipher that works by transforming a 64 bit data blocks using 56-bit common secret key. It involves a total of 16 rounds of permutation and substitution. The DES takes two inputs to the encryption function that are the plain text and the key. In DES encryption the plain text is 64 bit data block further using a 56-bit secret key. There are 16 rounds of permutation and substitution. Firstly the key is passed through an initial permutation box (in our algorithm is a pre-defined value box) that will rearranges the bits to produce a permuted input. The same function is continued for 16 rounds involving both permutation and substitution functions. Further the output of the 16th round contains 64-bits which are the function of the key and the plaintext. The left and right parts of the outputs received are exchanged to get the output. Lastly the output is passed through a permutation i.e. the inverse of the initial permutation and hence produces the 64-bit cipher text.



.

Figure 2.0 1

As we know the DES algorithm is a 16-round Feistel cipher. Encryption with the DES algorithm has three stages:

- The initial permutation
- The round function(16 times)
- The final permutation

The main or say the most time consuming is the second one i.e. the round function which runs 16 times. Our DES algorithm takes a 64-bit plain text and a 64-bit key and then encrypts it using the function des. This des function takes two parameters one is the plain text and other is the key of 64-bit each. In the des function we have given the values to the eight S-box having four rows and sixteen columns instead of a s-box function. This helps the algorithm in a way that when the program control reaches the s-box it can directly pick up the values from the S-box instead of running a function and then obtaining the values in each round and hence decreasing its time complexity.

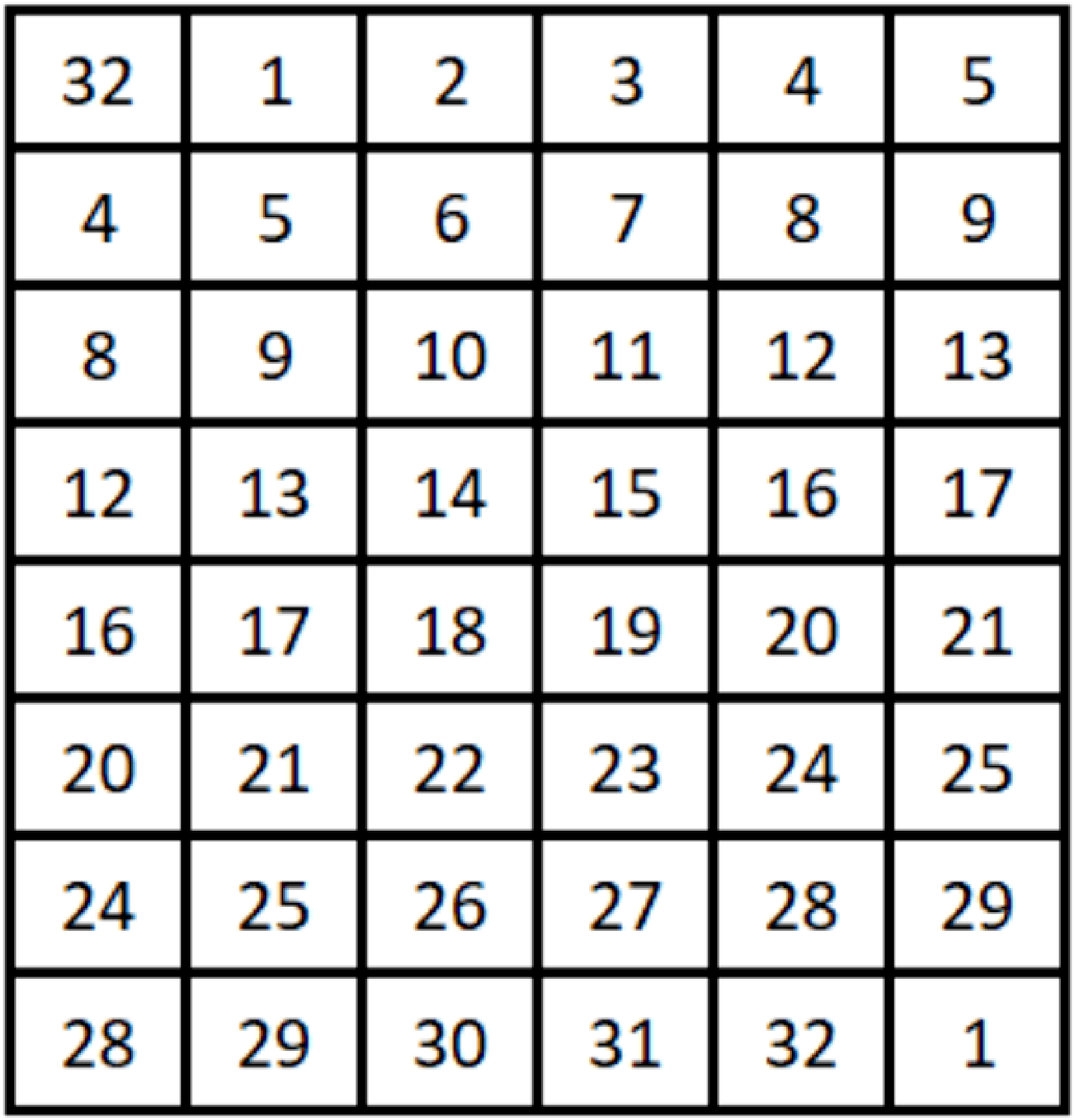
### 3.1.1 DES round function

There are 16 rounds and each round takes a 64-bit input which is divided into two 32-bit halves and a 56 bit-secret key. The des round function exchanges the right side with the left side of the inputs and applies the F function during the swaps. In our algorithm this F function is denoted by ‘des’. This F function has the following steps:

- The input half is expanded via expansion function.
- The expanded input is mixed with the round key.
- The result is broken into eight 6-bit pieces and every piece is passed through our predefined and unique substitution box(S-box which is defined in ‘des’ function).
- The result is then passed through the Permutation function.

#### 3.1.1.1 The expansion function

The purpose of expansion function is to expand the input to a 48-bit block.



| | | | | | |
|---|---|---|---|---|---|
| 32 | 1 | 2 | 3 | 4 | 5 |
| 4 | 5 | 6 | 7 | 8 | 9 |
| 8 | 9 | 10 | 11 | 12 | 13 |
| 12 | 13 | 14 | 15 | 16 | 17 |
| 16 | 17 | 18 | 19 | 20 | 21 |
| 20 | 21 | 22 | 23 | 24 | 25 |
| 24 | 25 | 26 | 27 | 28 | 29 |
| 28 | 29 | 30 | 31 | 32 | 1 |

Figure 3.0 1

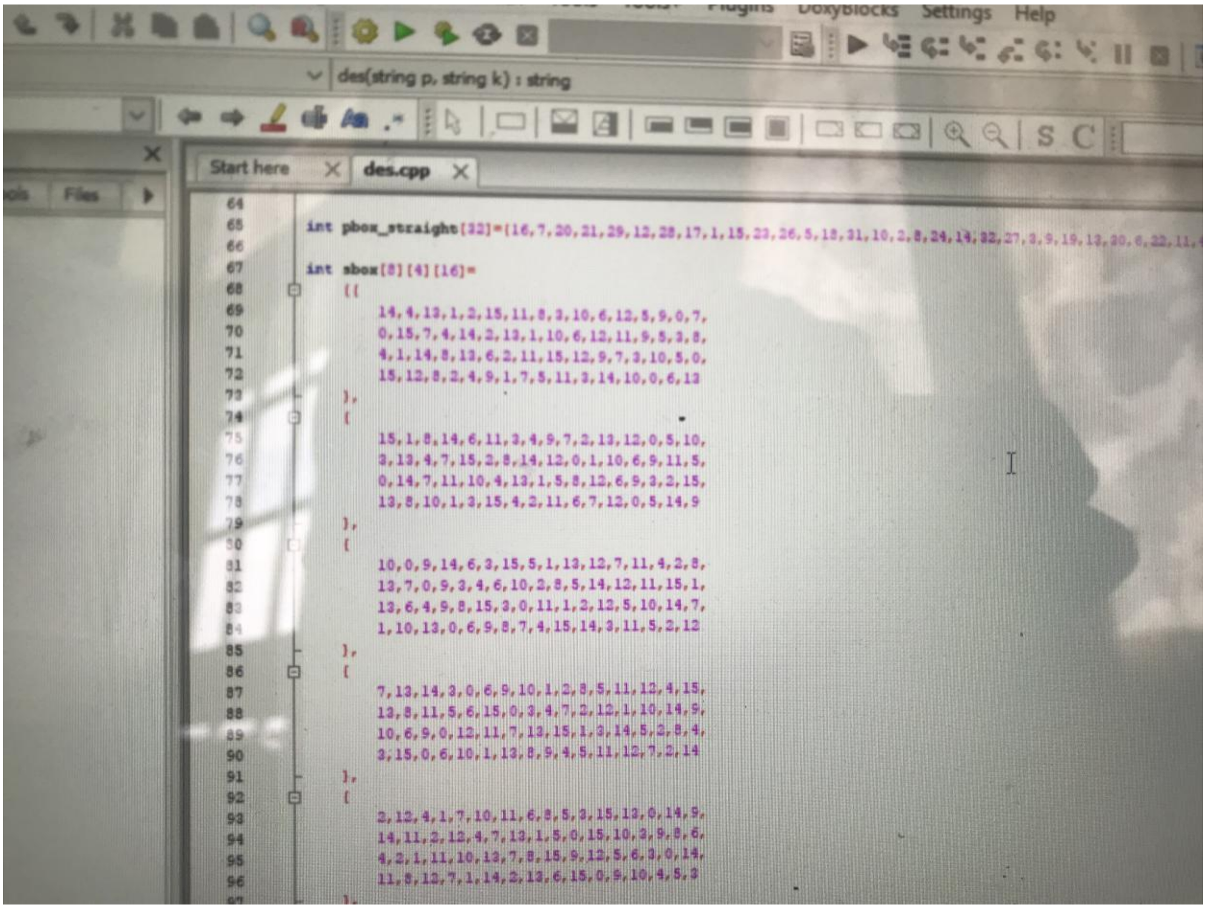
### 3.1.1.2 The Substitution box

In the S-box or the substitution box the input is broken into eight 6-bit blocks. Every block passes via a different S-box. The row and column of the S-box is selected by using the 6-bit blocks first the outer two bits which determines the row number and inner four bits that determine the column number using the following format.

| | x0000x | x0001x | x0010x | x0011x | x0100x | x0101x | x0110x | x0111x | x1000x | x1001x | x1010x | x1011x | x1100x | x1101x | x1110x | x1111x |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0yyyy0 | 14 | 4 | 13 | 1 | 2 | 15 | 11 | 8 | 3 | 10 | 6 | 12 | 5 | 9 | 0 | 7 |
| 0yyyy1 | 0 | 15 | 7 | 4 | 14 | 2 | 13 | 1 | 10 | 6 | 12 | 11 | 9 | 5 | 3 | 8 |
| 1yyyy0 | 4 | 1 | 14 | 8 | 13 | 6 | 2 | 11 | 15 | 12 | 9 | 7 | 3 | 10 | 5 | 0 |
| 1yyyy1 | 15 | 12 | 8 | 2 | 4 | 9 | 1 | 7 | 5 | 11 | 3 | 14 | 10 | 0 | 6 | 13 |

Figure 3.0 2

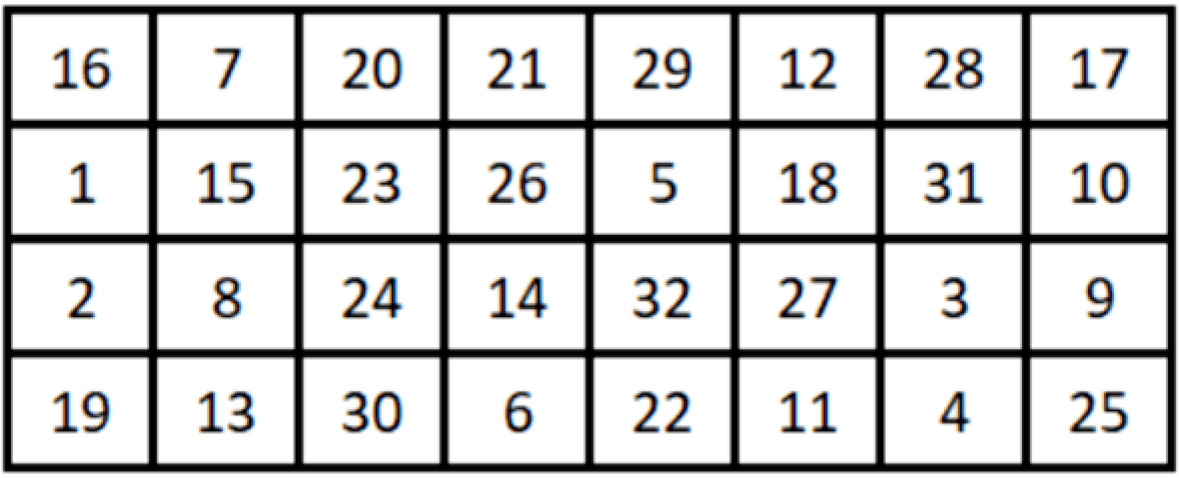
For e.g. - An input of 101101 would use the 3rd row (as outer two bits are 11 i.e. 3) and the sixth column (as the inner four bits are 0110 i.e. 6).



```
int pbox_straight[32]={16,7,20,21,29,12,28,17,1,15,23,26,5,18,31,10,2,8,24,14,32,27,3,9,19,13,30,6,22,11,

int sbox[8][4][16]=
    {{
        14,4,13,1,2,15,11,8,3,10,6,12,5,9,0,7,
        0,15,7,4,14,2,13,1,10,6,12,11,9,5,3,8,
        4,1,14,8,13,6,2,11,15,12,9,7,3,10,5,0,
        15,12,8,2,4,9,1,7,5,11,3,14,10,0,6,13
    },
    {
        15,1,8,14,6,11,3,4,9,7,2,13,12,0,5,10,
        3,13,4,7,15,2,8,14,12,0,1,10,6,9,11,5,
        0,14,7,11,10,4,13,1,5,8,12,6,9,3,2,15,
        13,8,10,1,3,15,4,2,11,6,7,12,0,5,14,9
    },
    {
        10,0,9,14,6,3,15,5,1,13,12,7,11,4,2,8,
        13,7,0,9,3,4,6,10,2,8,5,14,12,11,15,1,
        13,6,4,9,8,15,3,0,11,1,2,12,5,10,14,7,
        1,10,13,0,6,9,8,7,4,15,14,3,11,5,2,12
    },
    {
        7,13,14,3,0,6,9,10,1,2,8,5,11,12,4,15,
        13,8,11,5,6,15,0,3,4,7,2,12,1,10,14,9,
        10,6,9,0,12,11,7,13,15,1,3,14,5,2,8,4,
        3,15,0,6,10,1,13,8,9,4,5,11,12,7,2,14
    },
    {
        2,12,4,1,7,10,11,6,8,5,3,15,13,0,14,9,
        14,11,2,12,4,7,13,1,5,0,15,10,3,9,8,6,
        4,2,1,11,10,13,7,8,15,9,12,5,6,3,0,14,
        11,8,12,7,1,14,2,13,6,15,0,9,10,4,5,3
    },
```

### 3.1.1.3 The P function

The permutation function or the P function in DES takes a 32-bit block in input and gives the 32-bit output. Unlike other permutation functions the P function is not random and is the same for all rounds. The permutation is shown in the table below –

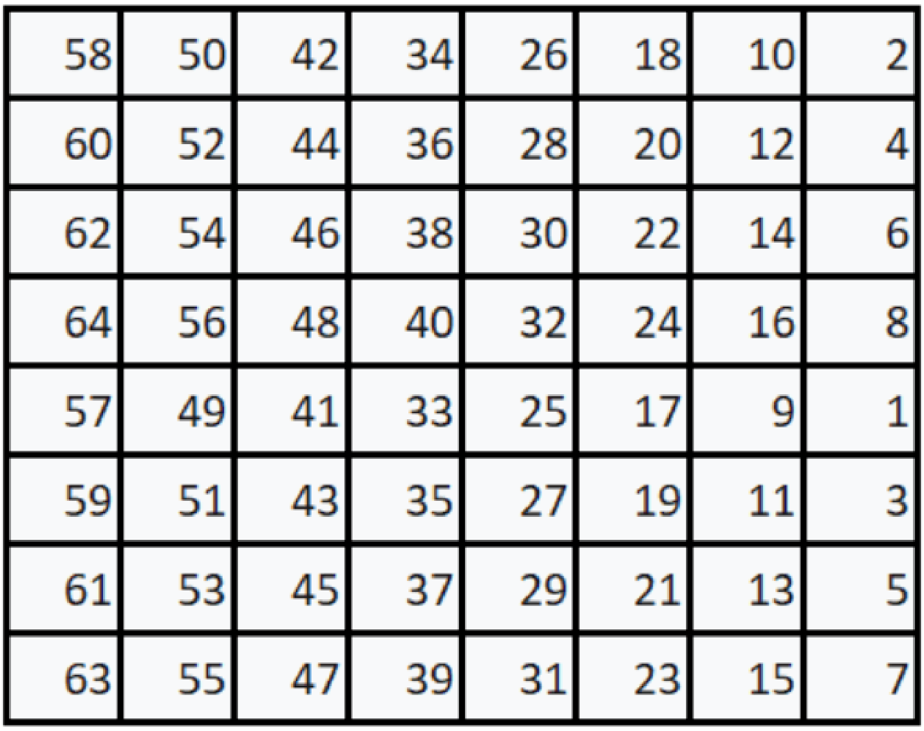


| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 16 | 7 | 20 | 21 | 29 | 12 | 28 | 17 |
| 1 | 15 | 23 | 26 | 5 | 18 | 31 | 10 |
| 2 | 8 | 24 | 14 | 32 | 27 | 3 | 9 |
| 19 | 13 | 30 | 6 | 22 | 11 | 4 | 25 |

Figure 3.0 3

### 3.1.2 The Initial Permutation

This function changes the order of the plaintext before the 1st round of encryption. This function makes no attempt to randomize data as the bits at odd positions in the plaintext fill out the last four of the output by filling out the last bit in order followed by 2nd last bit and so forth. The structure of the initial permutation is as follows



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 58 | 50 | 42 | 34 | 26 | 18 | 10 | 2 |
| 60 | 52 | 44 | 36 | 28 | 20 | 12 | 4 |
| 62 | 54 | 46 | 38 | 30 | 22 | 14 | 6 |
| 64 | 56 | 48 | 40 | 32 | 24 | 16 | 8 |
| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 |
| 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 |
| 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 |

Figure 3.0.4

## 3.2 System security

When the DES algorithm was introduced it was considered fairly secure as to crack the key one has to consider all the possible (72,057,594,037,927,936) key combination which was a big number for computers of the time but is not the case for the computer systems of today.

### 3.2.1 More security

The triple DES variant of the DES algorithm developed after the DES was itself too easy to crack. So the researchers at IBM came up with Triple DES which runs the DES algorithm three times i.e. –

- Encryption with single DES by the key(k1)
- Decryption with single DES by the key(k2).
- Encryption with single DES by the key(k3).

The keys k1, k2 & k3 can be made different or the same as per the requirement.

.

### 3.3 Triple DES

The Triple DES more or less the same thing as DES as it runs the DES algorithm three times i.e.-

- Encrypt plaintext with DES using K1.
- Now reversing the process and decrypt the output from previous using DES with K2.
- Then encrypt the output from earlier with single DES using K3.
- Output obtained after the previous is the cipher text.
- Similarly during decryption we first decrypt with K1, encrypt with K2 and then again decrypt with K3.

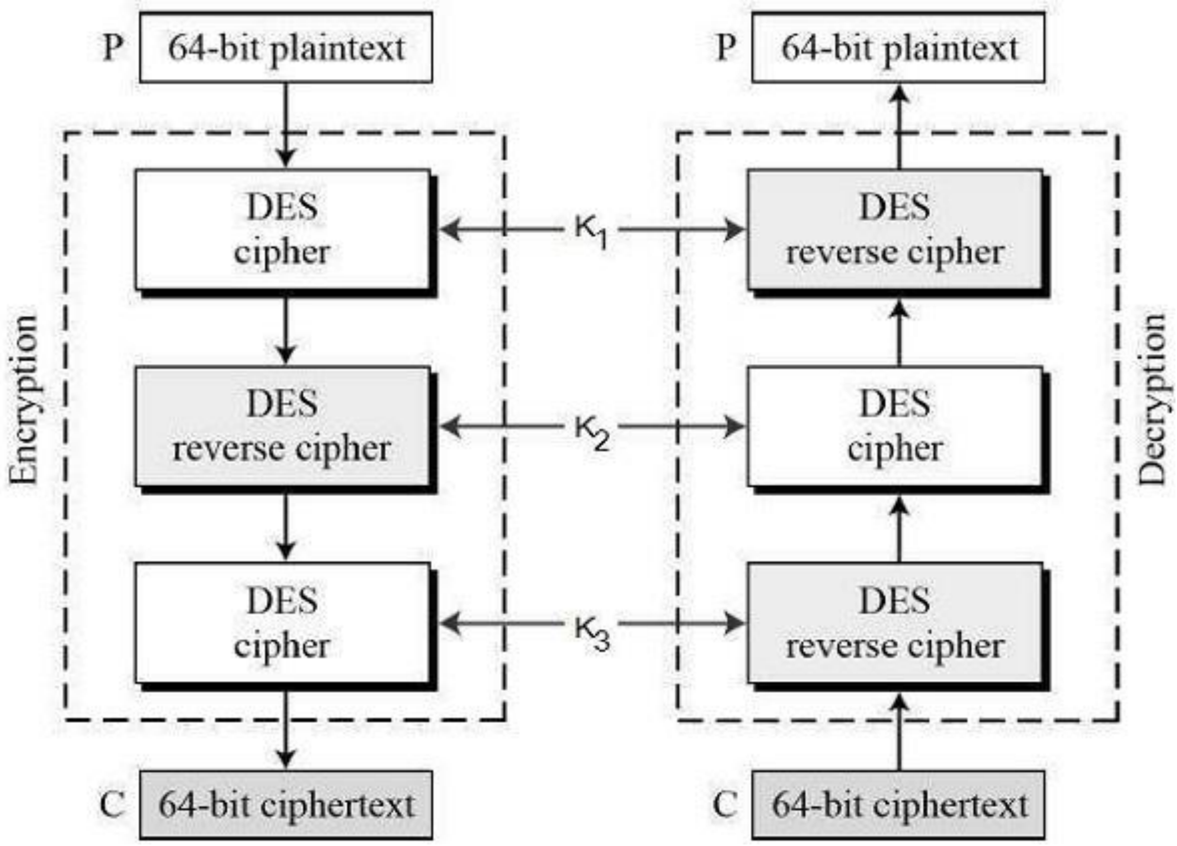


Figure 3.0.5

## 3.4 AES

The AES algorithm can be divided into three main parts that are

- Initial round
  - Add round key
- Main rounds
  - Sub bytes
  - Shift rows
  - Mix columns
  - Add round key
- Final round
  - Sub bytes
  - Shift rows
  - Add round key

The main rounds of the AES algorithm are repeated for fixed number of times for each variant of AES. For e.g. AES-128 uses 9 iterations, AES-192 uses 11, AES-256 uses thirteen rounds.

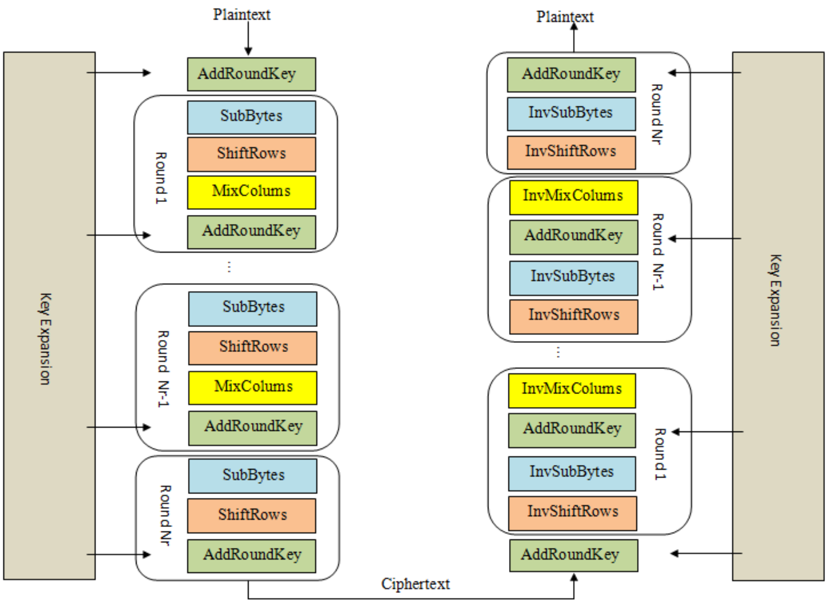


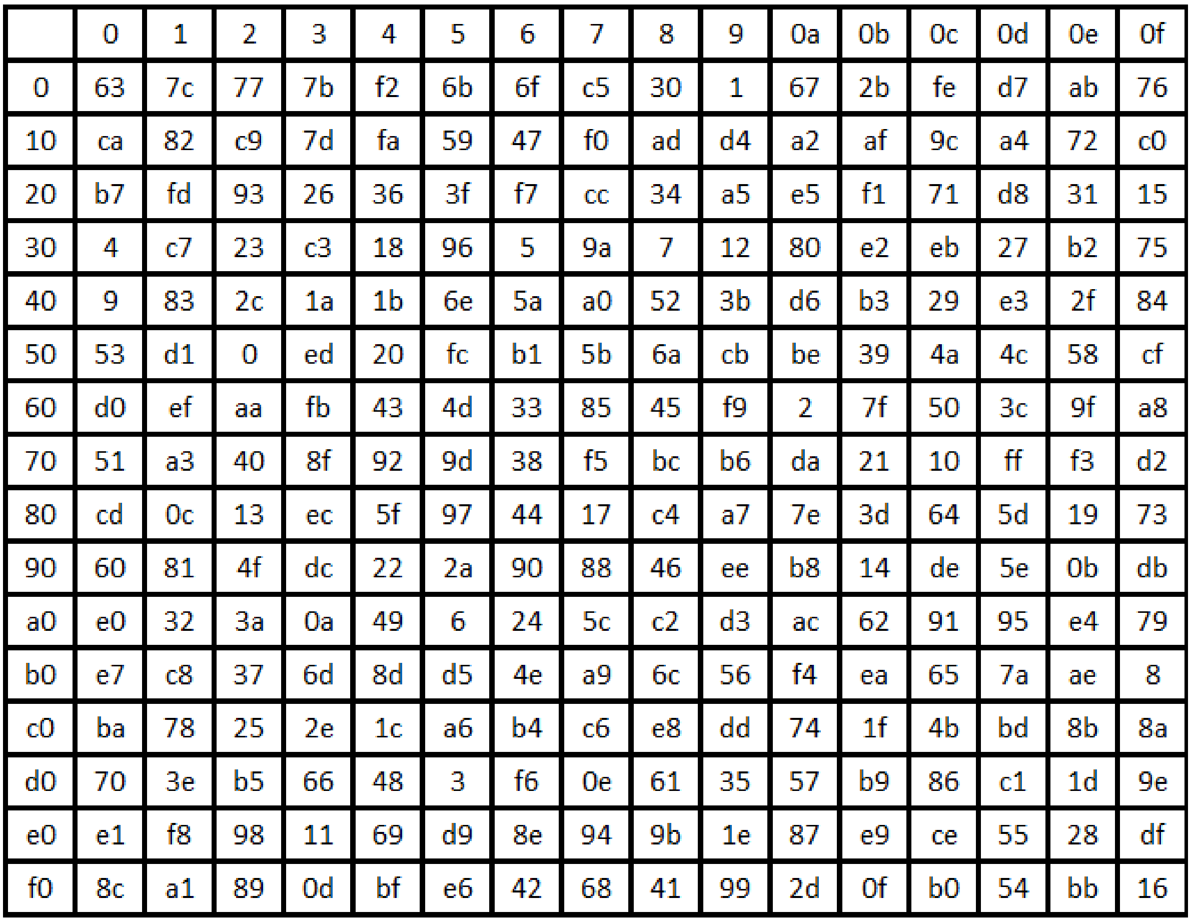
Figure 3.0.6

### 3.4.1 AddRoundKey

The AddRoundKey operation operates on the AES round key i.e. the input of the round is mixed with round key.

### 3.4.2 Subbytes

This operation involves splitting the input into bytes and passing through a S–box but the AES algorithm uses only one S-box for all bytes which is different from the DES algorithm which uses different Substitution-box for every iteration. The substitution-box in the AES algorithm implements inverse multiplication. The table for the Substituion box is as follows:



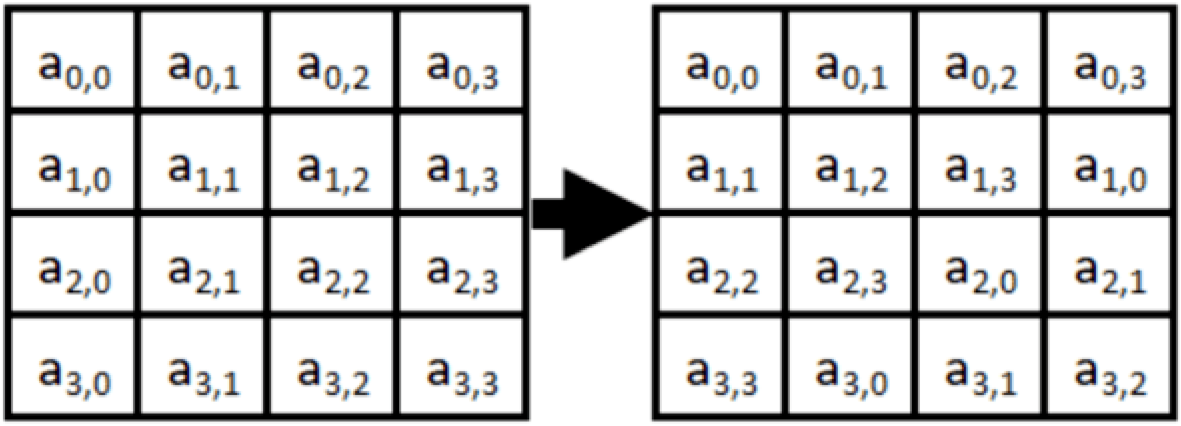
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0a | 0b | 0c | 0d | 0e | 0f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 63 | 7c | 77 | 7b | f2 | 6b | 6f | c5 | 30 | 1 | 67 | 2b | fe | d7 | ab | 76 |
| 10 | ca | 82 | c9 | 7d | fa | 59 | 47 | f0 | ad | d4 | a2 | af | 9c | a4 | 72 | c0 |
| 20 | b7 | fd | 93 | 26 | 36 | 3f | f7 | cc | 34 | a5 | e5 | f1 | 71 | d8 | 31 | 15 |
| 30 | 4 | c7 | 23 | c3 | 18 | 96 | 5 | 9a | 7 | 12 | 80 | e2 | eb | 27 | b2 | 75 |
| 40 | 9 | 83 | 2c | 1a | 1b | 6e | 5a | a0 | 52 | 3b | d6 | b3 | 29 | e3 | 2f | 84 |
| 50 | 53 | d1 | 0 | ed | 20 | fc | b1 | 5b | 6a | cb | be | 39 | 4a | 4c | 58 | cf |
| 60 | d0 | ef | aa | fb | 43 | 4d | 33 | 85 | 45 | f9 | 2 | 7f | 50 | 3c | 9f | a8 |
| 70 | 51 | a3 | 40 | 8f | 92 | 9d | 38 | f5 | bc | b6 | da | 21 | 10 | ff | f3 | d2 |
| 80 | cd | 0c | 13 | ec | 5f | 97 | 44 | 17 | c4 | a7 | 7e | 3d | 64 | 5d | 19 | 73 |
| 90 | 60 | 81 | 4f | dc | 22 | 2a | 90 | 88 | 46 | ee | b8 | 14 | de | 5e | 0b | db |
| a0 | e0 | 32 | 3a | 0a | 49 | 6 | 24 | 5c | c2 | d3 | ac | 62 | 91 | 95 | e4 | 79 |
| b0 | e7 | c8 | 37 | 6d | 8d | d5 | 4e | a9 | 6c | 56 | f4 | ea | 65 | 7a | ae | 8 |
| c0 | ba | 78 | 25 | 2e | 1c | a6 | b4 | c6 | e8 | dd | 74 | 1f | 4b | bd | 8b | 8a |
| d0 | 70 | 3e | b5 | 66 | 48 | 3 | f6 | 0e | 61 | 35 | 57 | b9 | 86 | c1 | 1d | 9e |
| e0 | e1 | f8 | 98 | 11 | 69 | d9 | 8e | 94 | 9b | 1e | 87 | e9 | ce | 55 | 28 | df |
| f0 | 8c | a1 | 89 | 0d | bf | e6 | 42 | 68 | 41 | 99 | 2d | 0f | b0 | 54 | bb | 16 |

Figure 3.0.7

In this input is broken into two 4-bit halves where the first half decides the row of the table and the second half decides the column.

### 3.4.3 ShiftRows

In this operation as the name suggest the rows of internal state (128-bit) of the cipher is shifted. These rows are represented as a 4*4 matrix where every cell contains a byte. Hence each row shifted to the left by a fixed amount i.e. there row number times. For e.g. the third row is shifted by three.

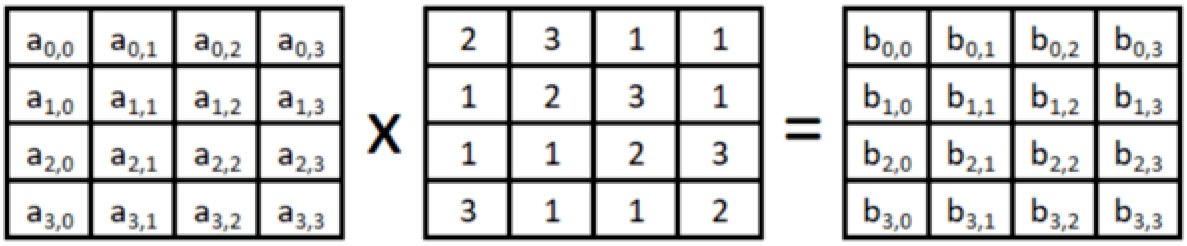


$$\begin{array}{|c|c|c|c|} \hline a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ \hline a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ \hline a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ \hline a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \\ \hline \end{array} \Rightarrow \begin{array}{|c|c|c|c|} \hline a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ \hline a_{1,1} & a_{1,2} & a_{1,3} & a_{1,0} \\ \hline a_{2,2} & a_{2,3} & a_{2,0} & a_{2,1} \\ \hline a_{3,3} & a_{3,0} & a_{3,1} & a_{3,2} \\ \hline \end{array}$$

Figure 3.0.8

### 3.4.4 MixColumns

The prime purpose of this operation as its name suggest in AES algorithm is to give a diffusion by mixing the input. It splits the matrix by the columns.



$$\begin{array}{|c|c|c|c|} \hline a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ \hline a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ \hline a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ \hline a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \\ \hline \end{array} \times \begin{array}{|c|c|c|c|} \hline 2 & 3 & 1 & 1 \\ \hline 1 & 2 & 3 & 1 \\ \hline 1 & 1 & 2 & 3 \\ \hline 3 & 1 & 1 & 2 \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ \hline b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ \hline b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ \hline b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \\ \hline \end{array}$$

Figure 3.0.9

3.5 A Sneak at existing system --

- ELGAMAL CRYPTOSYSTEM

Is an asymmetric key encryption algorithm. It is a type public key cryptography and is widely used in the modern cryptography.

One of its advantages is that the same plain text gives different cipher text each time it is encrypted.

Suppose Alice and bob are over a communication based on elgamal cryptosystem then:

Alice has to choose:

I. A large prime number say p.
II. A primitive root of the prime number p say r.
III. Then Alice has to select a random number say g.
IV. Then compute the y that is y= $r^g$ mod(p)
V. The public key will be (P,R,Y)
VI. And the private key will be the random number g.

Now the bob will do the following:

I. Encrypt the message M and sends it to sender.
II. Now will choose a integer say k which he keeps secret.
III. Now bob computes C1=$r^k$ mod (p) and C2=$y^k$ M mod (p) then discarding k.
IV. Then sends the encrypted message (C1, C2) to sender.

When Alice receives the encrypted message (C1, C2) she decrypts it using her private key i.e. g as:

$$C2C1^{-g}=y^{k}\ M\ \{r^{k}\}^{-g}\ \mathrm{mod}(p)$$

But for Alice $y=r^{g}$ mod(p)

Hence our equation becomes-

$$C2C1^{-g}=(r^{g})^{k}\ M\{r^{k}\}^{-g}\ \mathrm{mod}(p)$$

$$C2C1^{g}=M\ \mathrm{mod}(p)$$

Hence Alice gets the message M for Bob after using her private key

Even if Eve intercepts the encrypted message (C1, C2) {cipher text} she cannot perform the calculations as she does not have the private key g of Alice.

Note- Eve can still find the private key of Alice i.e. g by computing a discrete log in prime modulus p, however the calculation can be made more and more complex if the prime number that alice chooses in the first place is too large.

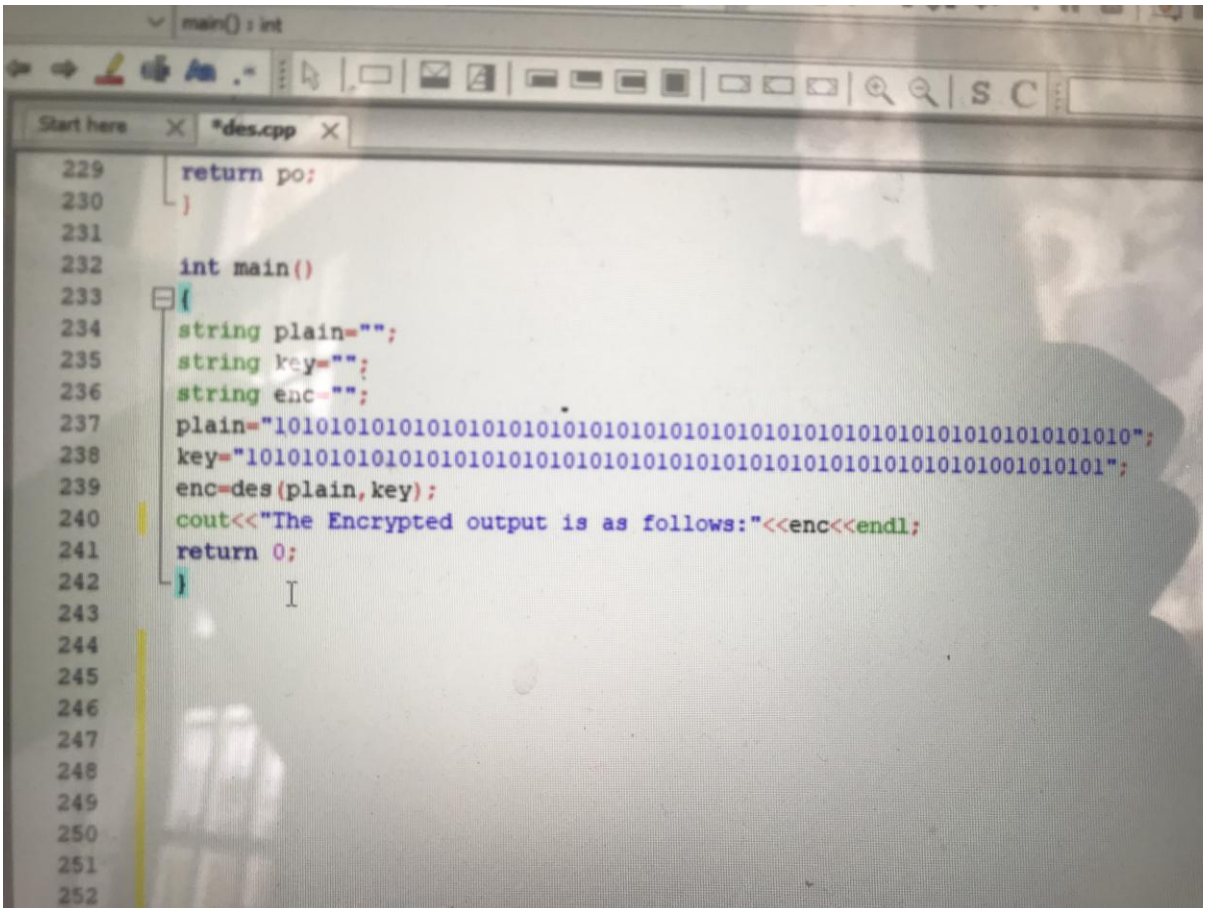
# CHAPTER 4 – ALGORITHM

## 4.1 Data Encryption Standard (DES)

The DES algorithm is as follows:

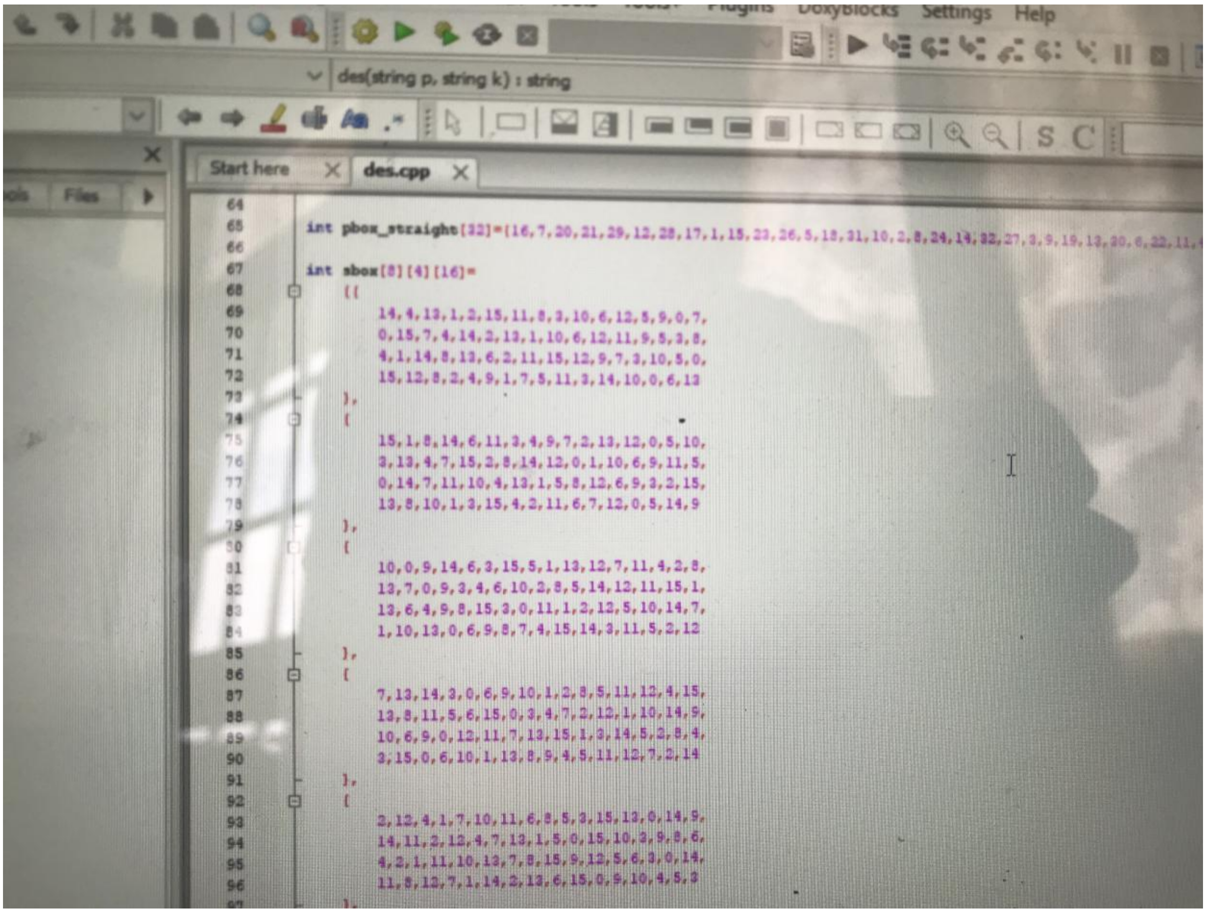
- Fractioning the plain text of 64-bit into 8 octet blocks
- Now these blocks undergo initial permutation.
- Breaking the blocks into (L) & (R).
- The process continues as the L and R undergo via P & S-box in each round.
- The above step is repeated 16 times.

The Main Function is as follows:



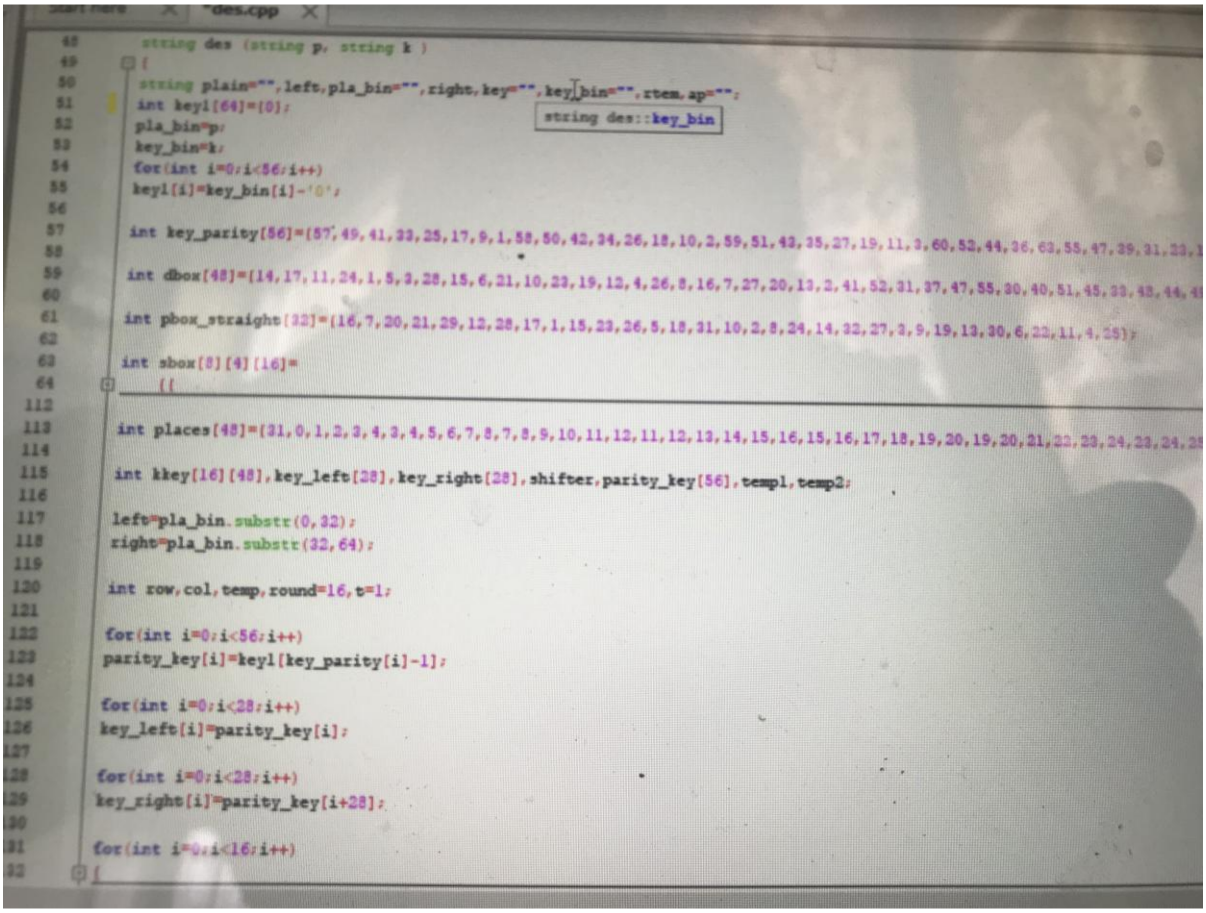
```
    return po;
}

int main()
{
    string plain="";
    string key="";
    string enc="";
    plain="1010101010101010101010101010101010101010101010101010101010101010";
    key="1010101010101010101010101010101010101010101010101010101001010101";
    enc=des(plain,key);
    cout<<"The Encrypted output is as follows:"<<enc<<endl;
    return 0;
}
```

The optimized S-box is as follows:



```
int pbox_straight[32]={16,7,20,21,29,12,28,17,1,15,23,26,5,18,31,10,2,8,24,14,32,27,3,9,19,13,30,6,22,11,

int sbox[8][4][16]=
    {{
        14,4,13,1,2,15,11,8,3,10,6,12,5,9,0,7,
        0,15,7,4,14,2,13,1,10,6,12,11,9,5,3,8,
        4,1,14,8,13,6,2,11,15,12,9,7,3,10,5,0,
        15,12,8,2,4,9,1,7,5,11,3,14,10,0,6,13
    },
    {
        15,1,8,14,6,11,3,4,9,7,2,13,12,0,5,10,
        3,13,4,7,15,2,8,14,12,0,1,10,6,9,11,5,
        0,14,7,11,10,4,13,1,5,8,12,6,9,3,2,15,
        13,8,10,1,3,15,4,2,11,6,7,12,0,5,14,9
    },
    {
        10,0,9,14,6,3,15,5,1,13,12,7,11,4,2,8,
        13,7,0,9,3,4,6,10,2,8,5,14,12,11,15,1,
        13,6,4,9,8,15,3,0,11,1,2,12,5,10,14,7,
        1,10,13,0,6,9,8,7,4,15,14,3,11,5,2,12
    },
    {
        7,13,14,3,0,6,9,10,1,2,8,5,11,12,4,15,
        13,8,11,5,6,15,0,3,4,7,2,12,1,10,14,9,
        10,6,9,0,12,11,7,13,15,1,3,14,5,2,8,4,
        3,15,0,6,10,1,13,8,9,4,5,11,12,7,2,14
    },
    {
        2,12,4,1,7,10,11,6,8,5,3,15,13,0,14,9,
        14,11,2,12,4,7,13,1,5,0,15,10,3,9,8,6,
        4,2,1,11,10,13,7,8,15,9,12,5,6,3,0,14,
        11,8,12,7,1,14,2,13,6,15,0,9,10,4,5,3
    },
```

The “des” function in our algorithm which responsible for encryption ia as follows:



```
 string des (string p, string k )
{
string plain="",left,pla_bin="",right,key="",key_bin="",rtem,ap="";
int key1[64]={0};
pla_bin=p;
key_bin=k;
for(int i=0;i<56;i++)
key1[i]=key_bin[i]-'0';

int key_parity[56]={57,49,41,33,25,17,9,1,58,50,42,34,26,18,10,2,59,51,43,35,27,19,11,3,60,52,44,36,63,55,47,39,31,23,1

int dbox[48]={14,17,11,24,1,5,3,28,15,6,21,10,23,19,12,4,26,8,16,7,27,20,13,2,41,52,31,37,47,55,30,40,51,45,33,48,44,4

int pbox_straight[32]={16,7,20,21,29,12,28,17,1,15,23,26,5,18,31,10,2,8,24,14,32,27,3,9,19,13,30,6,22,11,4,25};

int sbox[8][4][16]=
    {{

int places[48]={31,0,1,2,3,4,3,4,5,6,7,8,7,8,9,10,11,12,11,12,13,14,15,16,15,16,17,18,19,20,19,20,21,22,23,24,23,24,2

int kkey[16][48],key_left[28],key_right[28],shifter,parity_key[56],temp1,temp2;

left=pla_bin.substr(0,32);
right=pla_bin.substr(32,64);

int row,col,temp,round=16,t=1;

for(int i=0;i<56;i++)
parity_key[i]=key1[key_parity[i]-1];

for(int i=0;i<28;i++)
key_left[i]=parity_key[i];

for(int i=0;i<28;i++)
key_right[i]=parity_key[i+28];

for(int i=0;i<16;i++)
{
```

## 4.2 Triple DES

The Triple DES during encryption first encrypts with key K1 then decrypts with key K2 then again encrypts with key K3.the encryption and decryption can be represented as:

Encryption:

$c = E_3(D_2(E_1(m)))$

Decryption:

$m = D_1(E_2(D_3(c)))$

With three keys (56-bits each) of three operations the additive key becomes 168-bits. Hence the algorithm is same as the single DES the only difference is that the operation is taking place three times.

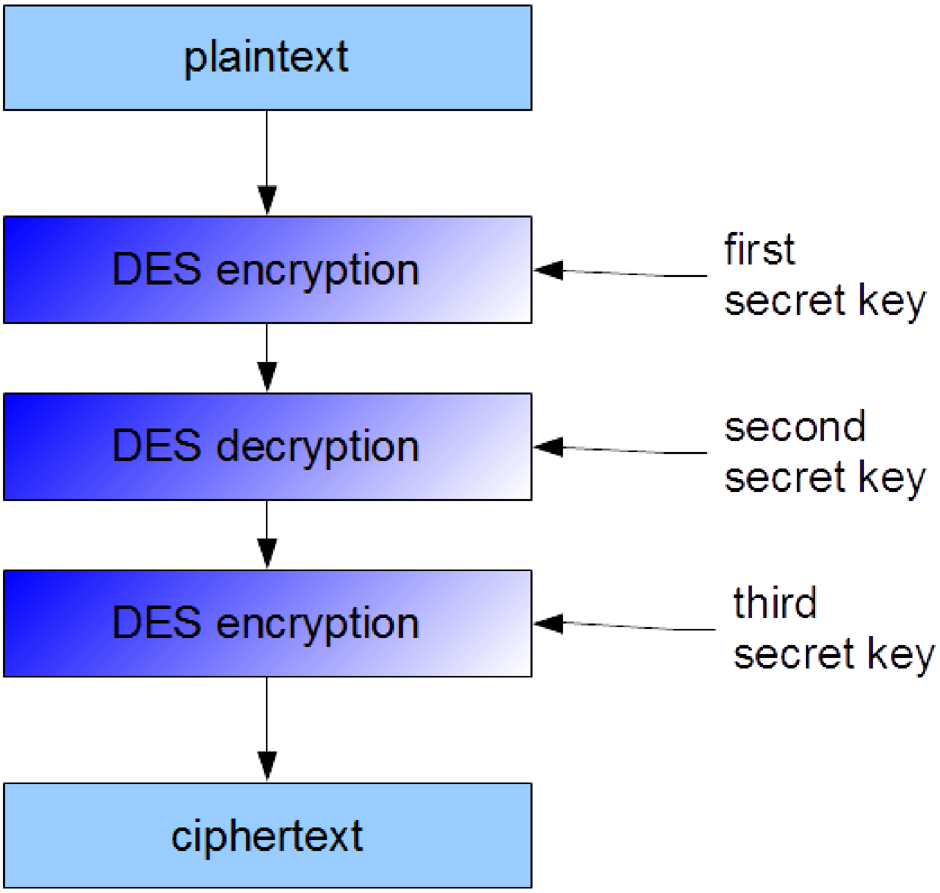


Figure 4.0 1

## 4.3 Advanced Encryption Algorithm

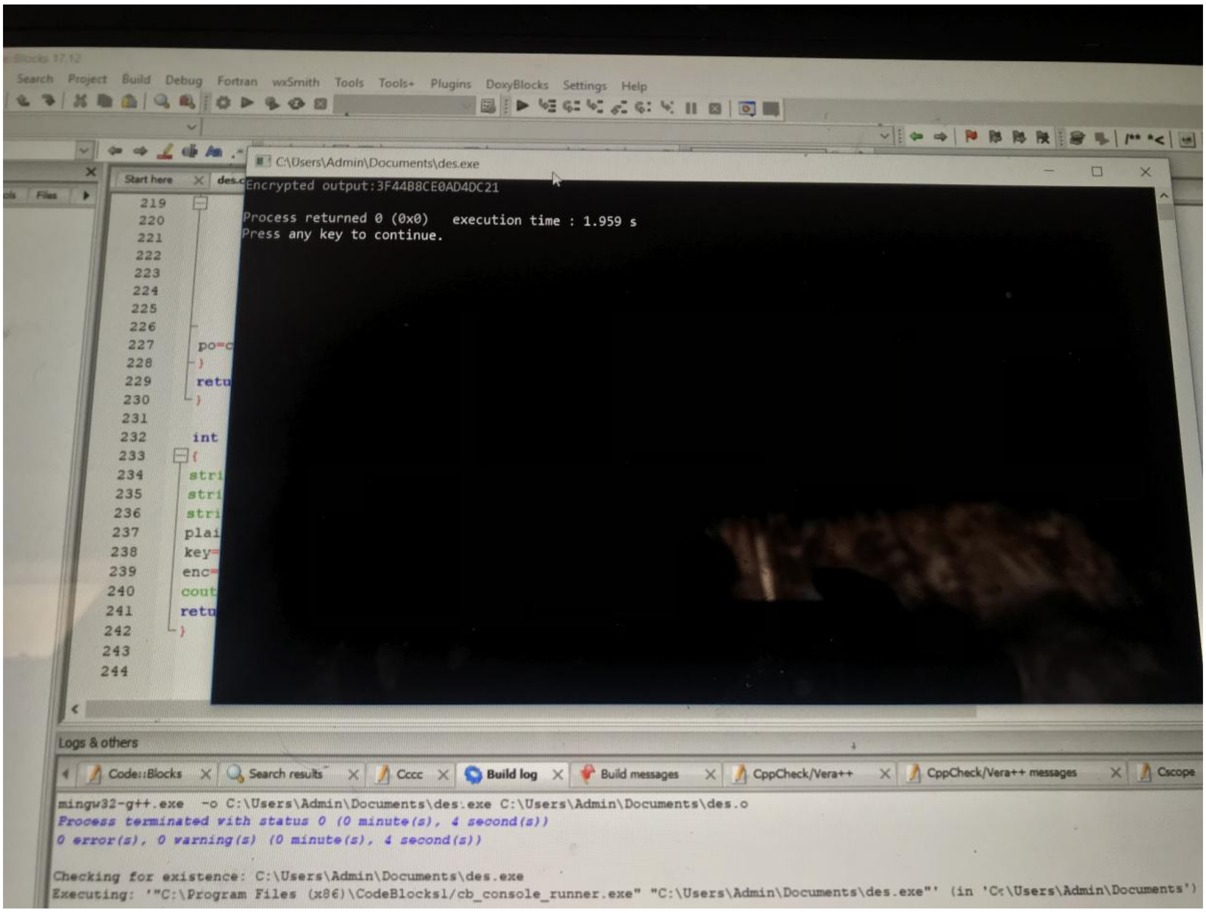
The Matlab code for the optimized AES algorithm is as follows:

```
function e=keygen(M,N,key0)

for(i=1:200)

key0=3.925*key0*(1-key0);

end

key1=3.925;

for(i=1:M)

   for(j=1:N)

      key0=key1*key0*(1-key0);

      a(i,j)=key0;

   end

end

key3=0.2;

key2=3.925;

for(i=1:M)

   for(j=1:N)

       key3=key2*key3*(1-key3);

       b(i,j)=key3;

   end

end

key4=0.3;

key2=3.925;
```

```
for(i=1:M)
    for(j=1:N)
        key4=key2*key4*(1-key4);
        c(i,j)=key4;
    end
end
t=0.4;
w0=0.2;
w1=0.5;
w2=0.3;
w=(1-t)^2*w0+2*t*(1-t)*w1+t^2*w2;
for(i=1:M)
   for(j=1:N)
        P(i,j)=(1-t)^2*a(i,j)*w0+2*t*(1-t)*b(i,j)*w1+t^2*c(i,j)*w2;
   %    d(i,j)=P(i,j)/w;
   d(i,j)=P(i,j);
    end
end
x=d;
e=round(x*255);
end
```

# CHAPTER 5 – RESULT AND PERFORMANCE ANALYSIS

The usual time taken by the DES algorithm is approx. 5.998 seconds. Our optimized DES algorithm takes time of about 5.959 seconds. This is due the fact that we discussed earlier which is the predefined value of substitution and permutation box given in our program, as the program control does not has to ask for S-box value for every round. It just collects the value of S-box given in the program which saves the time. The same is with Triple-DES algorithm.



```
Encrypted output:3F44B8CE0AD4DC21

Process returned 0 (0x0)   execution time : 1.959 s
Press any key to continue.
```

```
   po=c
  }
  retu
 }

 int
 {
  stri
  stri
  stri
  plai
  key=
  enc=
  cout
  retu
 }
```

```
mingw32-g++.exe  -o C:\Users\Admin\Documents\des.exe C:\Users\Admin\Documents\des.o
Process terminated with status 0 (0 minute(s), 4 second(s))
0 error(s), 0 warning(s) (0 minute(s), 4 second(s))

Checking for existence: C:\Users\Admin\Documents\des.exe
Executing: '"C:\Program Files (x86)\CodeBlocks1/cb_console_runner.exe" "C:\Users\Admin\Documents\des.exe"' (in 'C:\Users\Admin\Documents')
```

Figure 5.0 1

# CHAPTER 6 – CONCLUSION AND FUTURE SCOPE

## 6.1 Conclusion

The time optimization is big factor today in the world modern cryptography and its algorithm. Throughout our project we worked towards the goal of reducing the time taken by the algorithms and also to minimize their vulnerabilities. We reduced the time taken by the DES algorithm by a factor of 39 microseconds. Same was the case with the Triple-DES algorithm.

## 6.2 Future scope

The field of cryptography has a vast future scope as commerce and communications continue to move to computer networks, hence providing security to these new emerging fields will be essential. We would also have one research work in other cryptographic algorithms but could only study these algorithms. We would also have explored the new field of quantum cryptography which provides essentially the best security for communication over a channel, as it works on a quantum channel and uses photons of light rather than information bits.

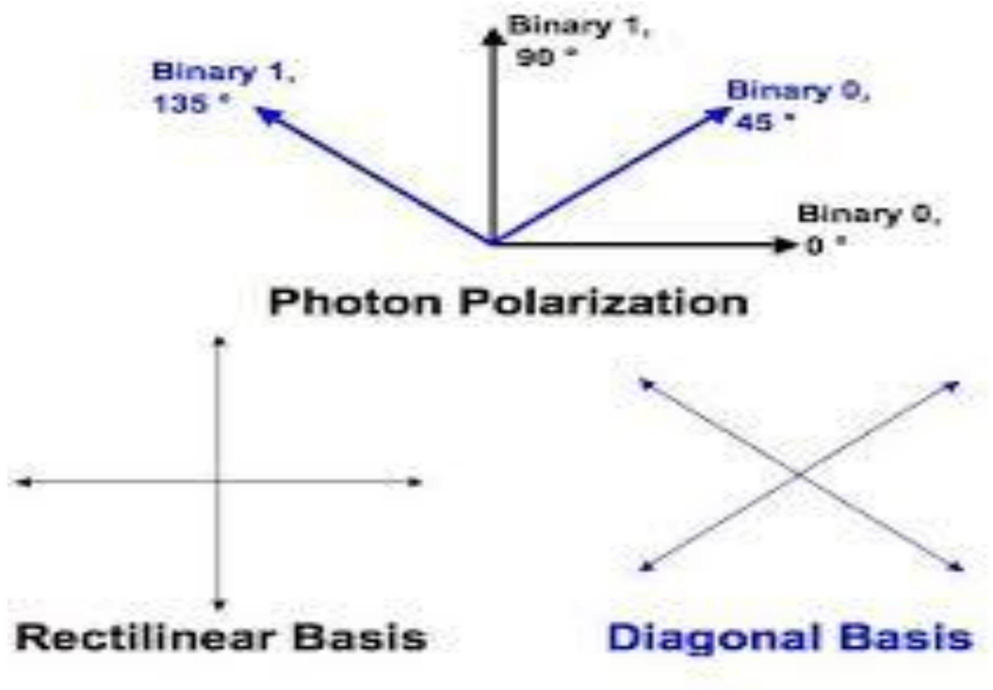
### 6.2.1 Quantum Cryptography

Now that we have discussed the existing algorithms and the modified algorithm that we have proposed we shall discuss the future advances and scope in the field of cryptography. One such field is Quantum cryptography which is based on the principle on quantum mechanics and the Heisenberg’s uncertainty principle. It deals with the issue of eavesdropping with fairly accurate precision and nullifies it as the advances in classical computing are not ample to solve some problems as they rely on mathematical tricks to prevent the information by introducing large mathematical equations which may be hard to solve but not quite out of reach of heavy processing computers.

Quantum technology can be used for secure key sharing. The quantum cryptography includes quantum key distribution for secure key sharing also known as QKD. The advantage that QKD provides is that the if the eavesdropper eve has intercepted the key that say Alice and Bob have shared over the channel then they can know for sure that the key has been intercepted and can discard the key. This is due to the principle that any interference with the quantum particle can be traced back. As in QKD the polarization of photons is used to transfer the key.

To explain this we must take an example as:

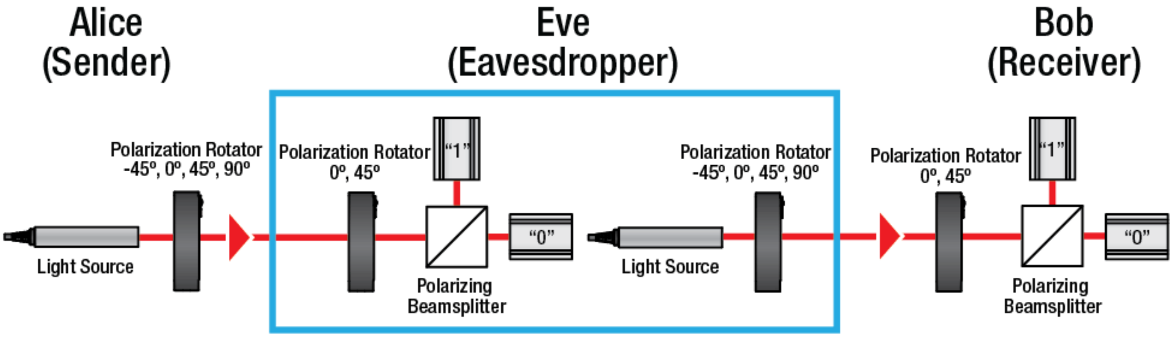
Say Alice needs to send Bob the key via QKD using polarized photons, she might send the photon polarized in four (or in some case two) ways i.e. horizontal, vertical and diagonally.



When bob receives the photons he chooses which filter to use i.e. whether the horizontal, vertical or diagonal filter. If Bob chooses the correct filter then he can get the right information and if he chooses the wrong then he may not get the information. After this session of receiving the photon Bob can tell Alice what were his choices and then they may keep the ones which bob choose correctly, in this way they can have a key which is completely secured. This is called Reconciliation.

Now in the above case if the eavesdropper eve intercepts the polarized photon then she has to measure its polarization and will hence leave bob with nothing as the polarized photon is already measured and Alice and Bob will know that there is an eavesdropper in their channel of communication.

Illustrative diagram --



# References

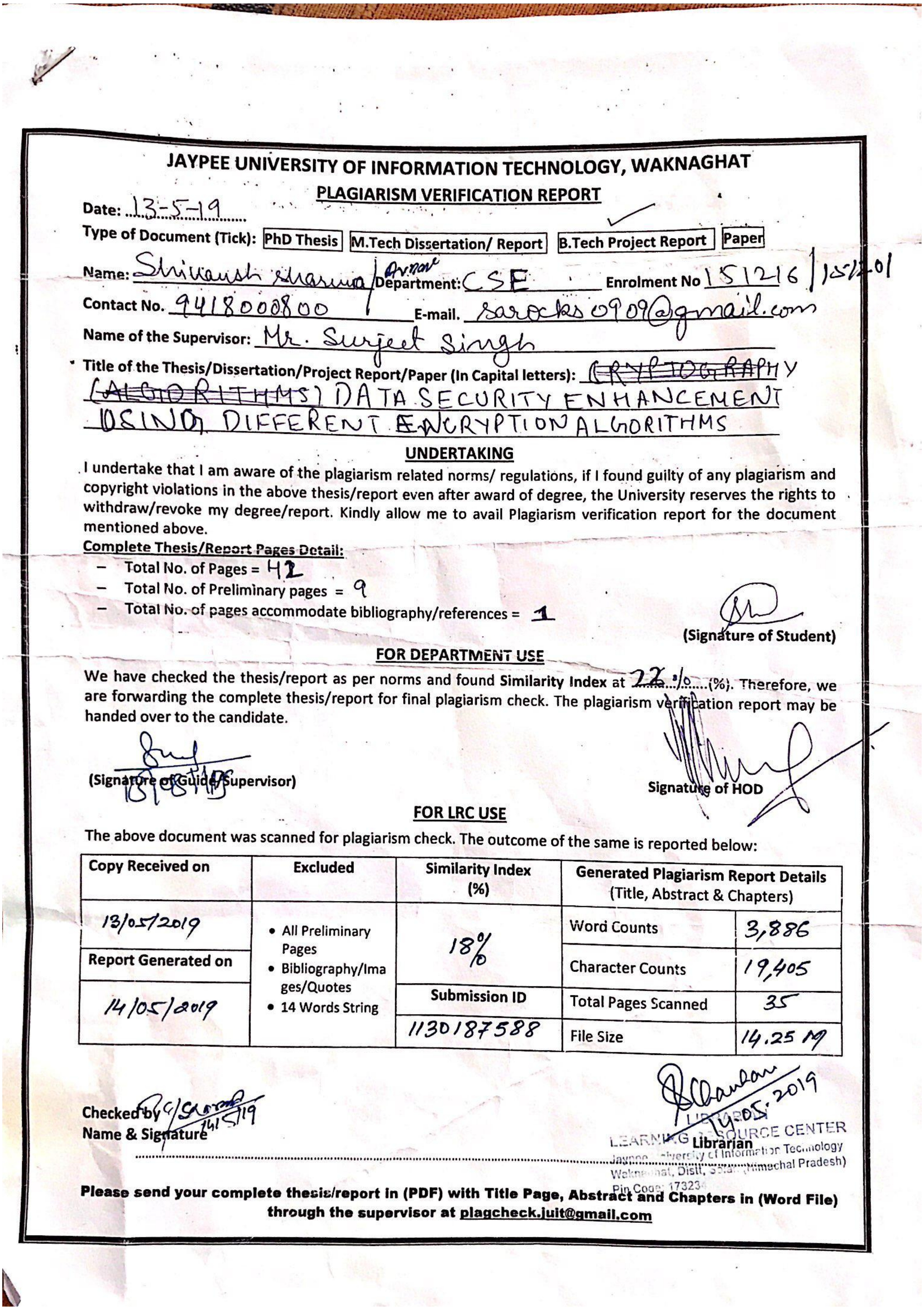
1) Bruce Schneier Applied Cryptography

2) Research paper on Performance analysis of DES

3) Christ of Paar and Jan Pelzl Understanding Cryptography.

4) www.Commonlounge.com

https://static.commonlounge.com/fp/600w/y6UQ3zYSQRlWMrW537E7ooK1m1520492304_kc



## JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

### PLAGIARISM VERIFICATION REPORT

Date: 13-5-19

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | [✓ B.Tech Project Report] | Paper

Name: Shivansh sharma / Arnav Department: CSE Enrolment No 151216 | 151201

Contact No. 9418000800 E-mail. sarocks0909@gmail.com

Name of the Supervisor: Mr. Surjeet Singh

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): ~~CRYPTOGRAPHY (ALGORITHMS)~~ DATA SECURITY ENHANCEMENT USING DIFFERENT ENCRYPTION ALGORITHMS

**UNDERTAKING**

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

**Complete Thesis/Report Pages Detail:**

- Total No. of Pages = 42
- Total No. of Preliminary pages = 9
- Total No. of pages accommodate bibliography/references = 1

(Signature of Student)

**FOR DEPARTMENT USE**

We have checked the thesis/report as per norms and found Similarity Index at 22 % (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

13/05/19

(Signature of Guide/Supervisor)

Signature of HOD

**FOR LRC USE**

The above document was scanned for plagiarism check. The outcome of the same is reported below:

| Copy Received on | Excluded | Similarity Index (%) | Generated Plagiarism Report Details (Title, Abstract & Chapters) | |
|---|---|---|---|---|
| 13/05/2019 | • All Preliminary Pages<br>• Bibliography/Images/Quotes<br>• 14 Words String | 18% | Word Counts | 3,886 |
| Report Generated on | | | Character Counts | 19,405 |
| 14/05/2019 | | Submission ID | Total Pages Scanned | 35 |
| | | 1130187588 | File Size | 14.25 M |

Checked by
Name & Signature 14/5/19

Librarian 14.05.2019

**Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com**