FALL SEMESTER 2018-19

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

A PROJECT ON:

ELECTRONIC FUND TRANSFER OVER INTERNET USING DES

UNDER THE GUIDANCE OF :

PROF . DR. MARIMUTHU K

Submitted By:

SHUBHAM GUPTA 17BCI0044

KUMAR ABHISHEK 17BCI0145

VISHAL SAINI 17BCI0188

Abstract

Electronic Fund Transfer involves electronic transfer of money by financial institutions. EFT is the groundwork of the cash-less and check-less culture where and paper bills, checks, envelopes, stamps are eliminated. EFT is used for transferring money from one bank account directly to another without any paper money changing hands. The most popular application of EFT is that instead of getting a paycheck and putting it into a bank account, the money is deposited to an account electronically. To secure the path, cryptography technique, DES is for instant verification and consistency. Using DES algorithm, important data like account number, user name will be encrypted before sending it over internet and again it will be decrypted using DES at its destination computer. Through this way, confidential data will be protected from any kind of theft.

Introduction

Electronic Funds Transfer (EFT) is a system of transferring money from one bank account directly to another without any paper money changing hands. One of the most widely-used EFT programs is Direct Deposit, in which payroll is deposited straight into an employee's bank account, although EFT refers to any transfer of funds initiated through an electronic terminal, including credit card, ATM and point-of-sale (POS) transactions. It is used for both credit transfers, such as payroll payments, and for debit transfers, such as mortgage payments. The growing popularity of EFT for online bill payment is paving the way for a paperless universe where checks, stamps, envelopes, and paper bills are obsolete. The benefits of EFT include reduced administrative costs, increased efficiency, simplified bookkeeping, and greater security. The number of business devoted to promoting commerce on the Internet has been growing exponentially, but they all share the goal of making commercial transactions over the Internet safe, simple, and secure and earning profit in the process. The growth of electronic commerce has created the potential new risks and abuses. Customers routinely buy products, trade investment and bank online using credit cards, social security numbers concern over the privacy and security of on-line transactions in E-Commerce.

Electronic merchants need to feel confident that they can safely market and deliver their products, get paid for all products purchased, and not lose any products to theft. Electronic consumers need to feel confident they can safely select and take delivery of products, pay for them, and not be concerned about compromise of payment information. Everyone wants to feel confident that the individuals they deal with across the Internet are who they say they are, to avoid losses to fraud.

Application of cryptography to various fields of E-Commerce where security of information is a must is an area of research activities [34] nowadays to provide safety to everyone.

Cryptography is the arts and sciences of secret writing. Traditional cryptography relied on the use of keys and coding algorithms. Secret Key Cryptography and Public Key Cryptography are two general methods for cryptography.

In secret key cryptography the algorithm, is kept secret and manipulate the message to be coded in a repeatable way; the key also kept secret, provided a starting point for encoding and decoding message.

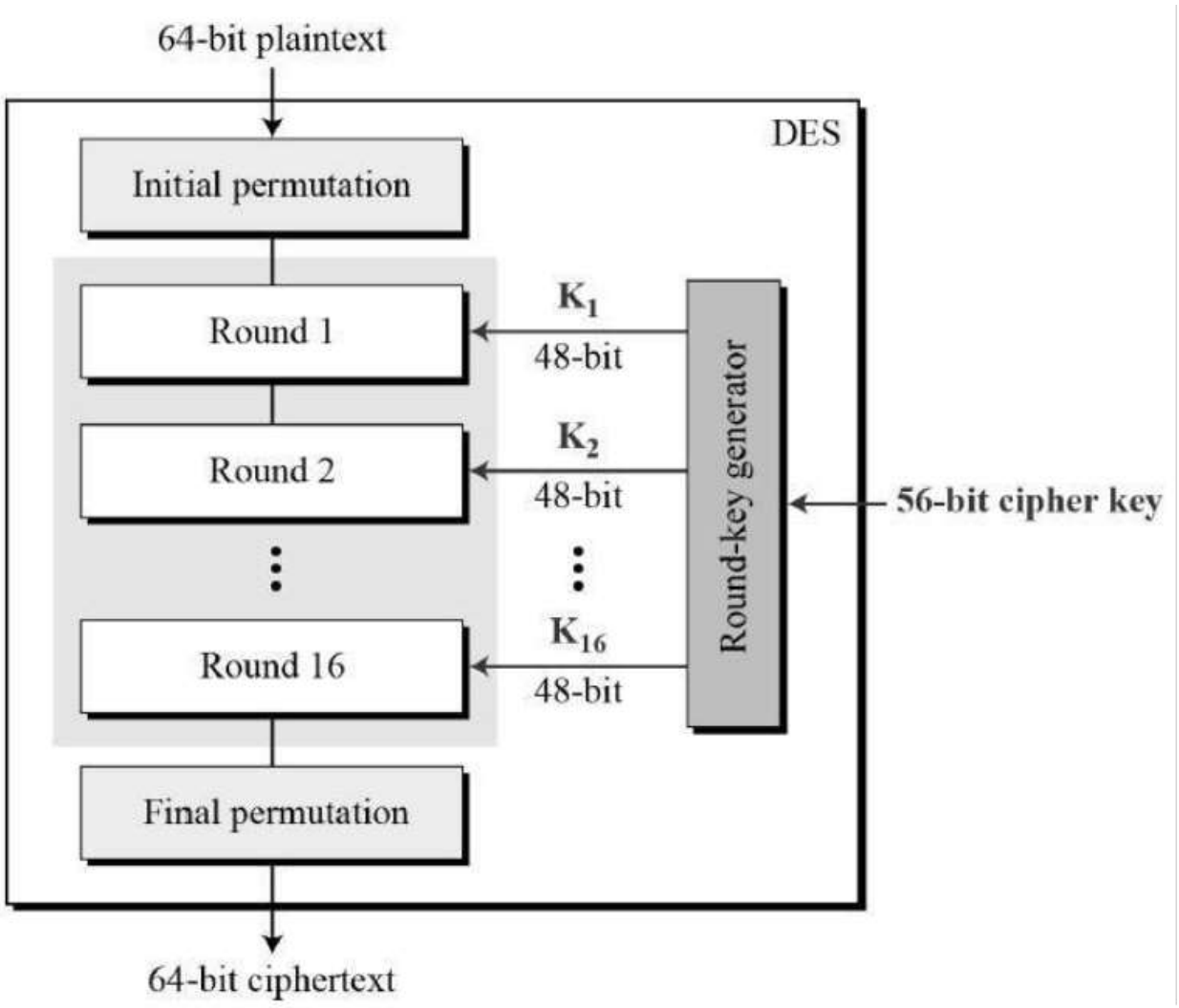
DES stands for Data Encryption Standard. It works on secret key cryptography to encrypt the message. It uses 56 bit key to encrypt the message. A 56 bit key means that there are 2^{56} possible key choices. A large size key means that there are too many possible key choices for an attacker to possibly guess the current key in a reasonable amount of time. Thus, DES is very much helpful in securing the fund transfer over internet.

DES is a block cipher designed by IBM researchers with assistance from the National Security Agency (NSA) in the 1970s. It was the first encryption scheme that was adopted as a standard by National Institute of Standards and Technology (NIST) as Federal Information Processing Standard (FIPS) publication PUB 46 [NIST (1977)]. DES uses a 56-bit key, a 64-bit block size, and can be implemented efficiently in hardware.

Data Encryption Standard

The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).

DES is an implementation of a Feistel Cipher. It uses 16 round Feistel structure. The block size is 64-bit. Though, key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only). General Structure of DES is depicted in the following illustration –

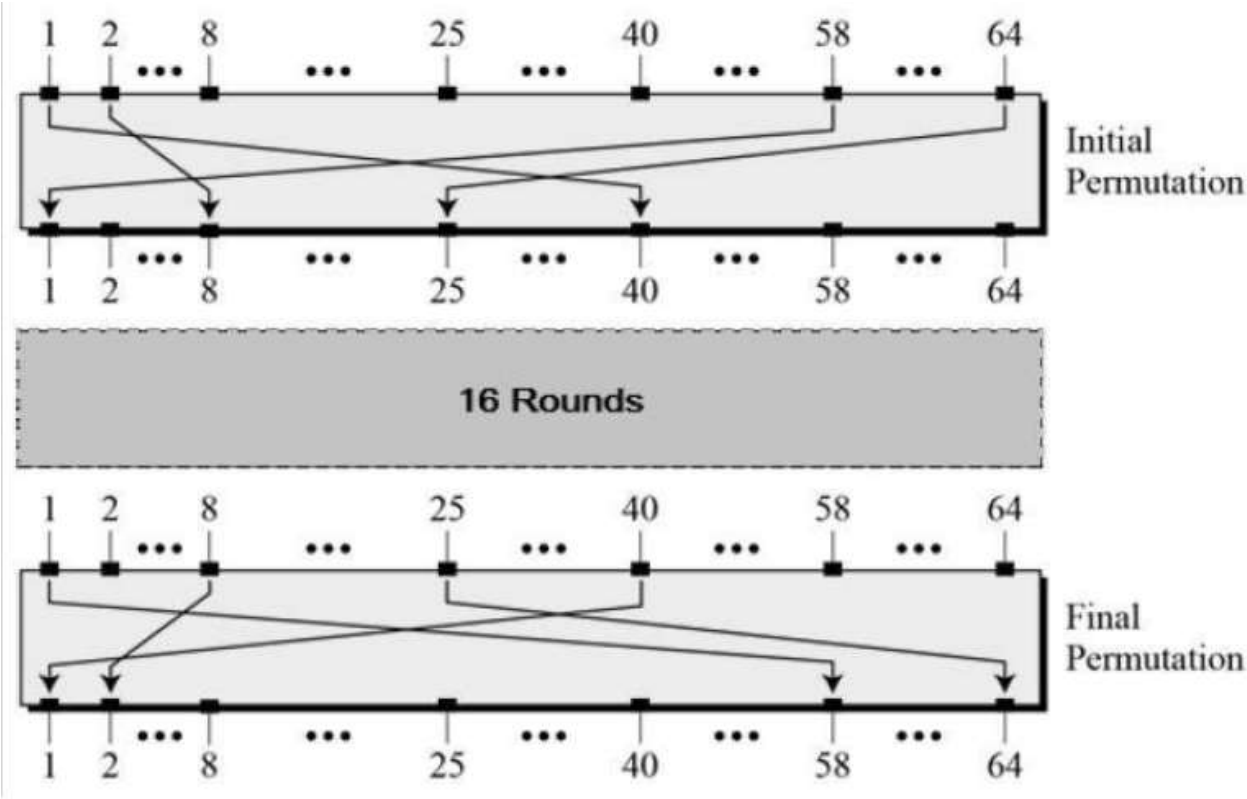


Since DES is based on the Feistel Cipher, all that is required to specify DES is –

- Round function
- Key schedule
- Any additional processing − Initial and final permutation

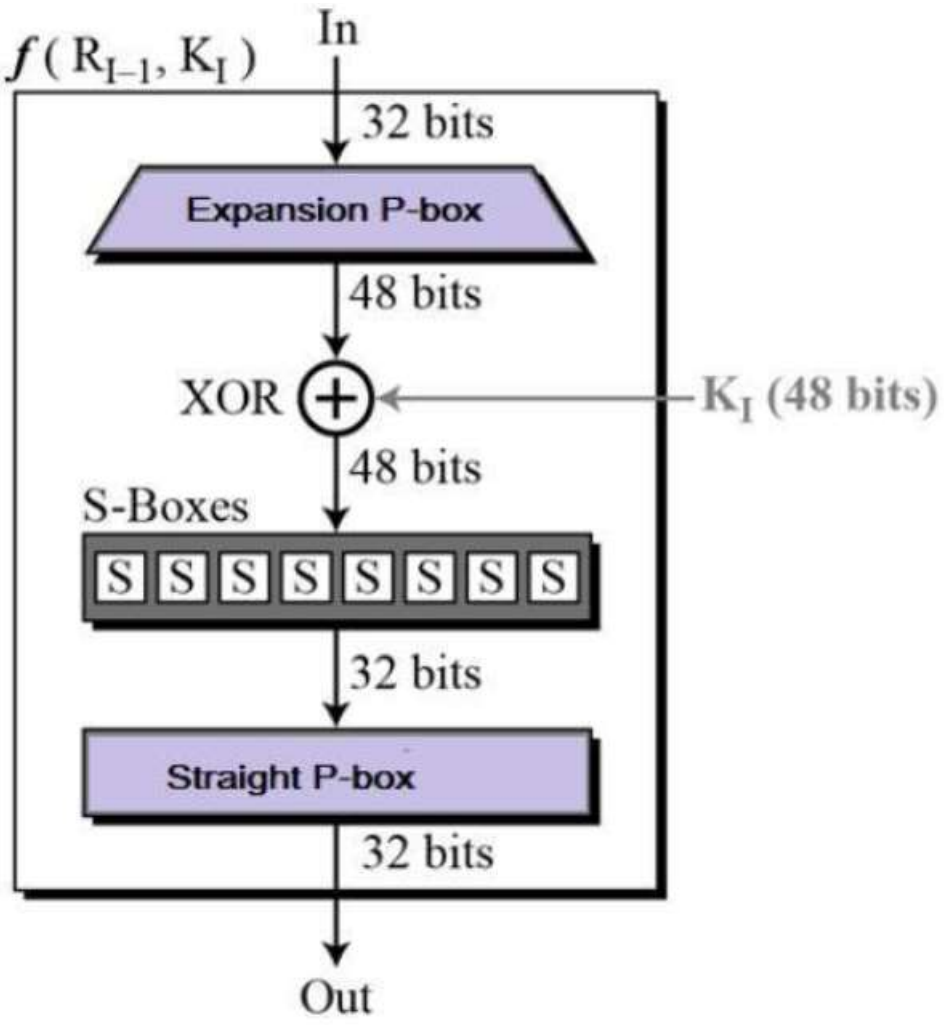
Initial and Final Permutation

The initial and final permutations are straight Permutation boxes (P-boxes) that are inverses of each other. They have no cryptography significance in DES. The initial and final permutations are shown as follows −



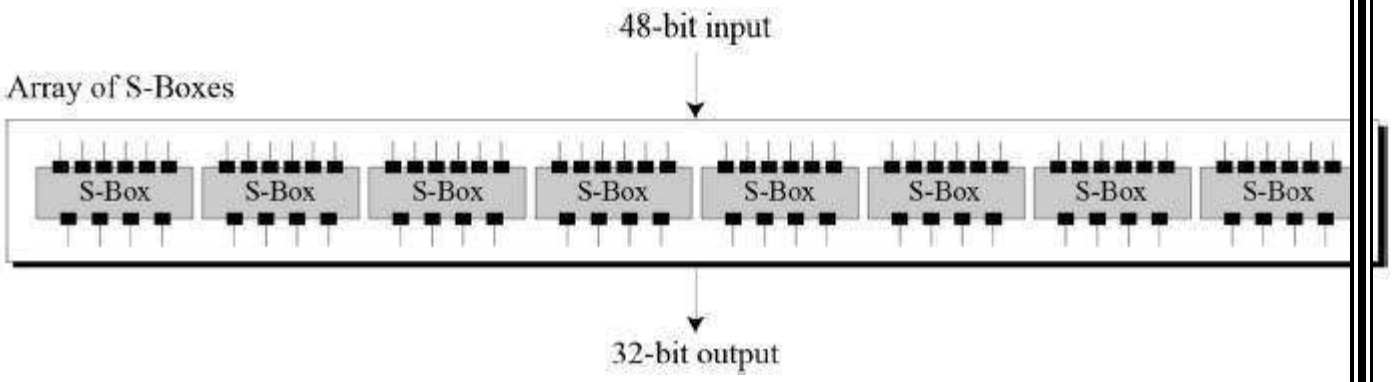
Round Function

The heart of this cipher is the DES function, *f*. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.

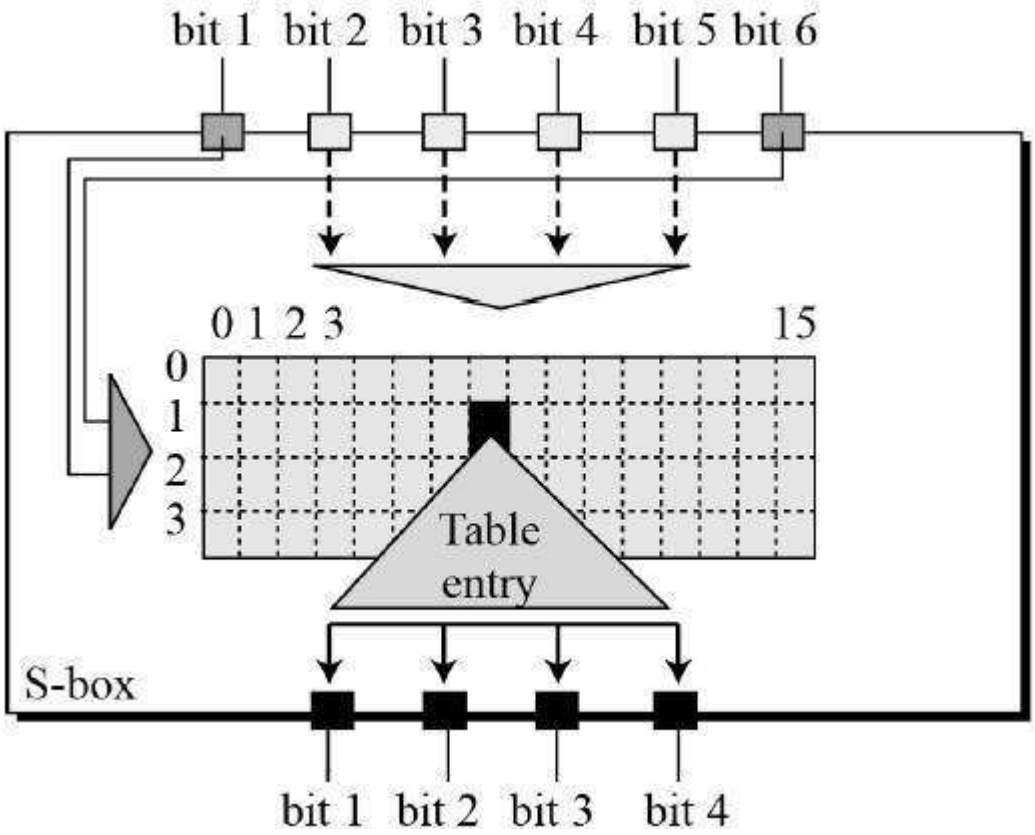


XOR (Whitener). − After the expansion permutation, DES does XOR operation on the expanded right section and the round key. The round key is used only in this operation.

- **Substitution Boxes.** − The S-boxes carry out the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output. Refer the following illustration –



- The S-box rule is illustrated below –

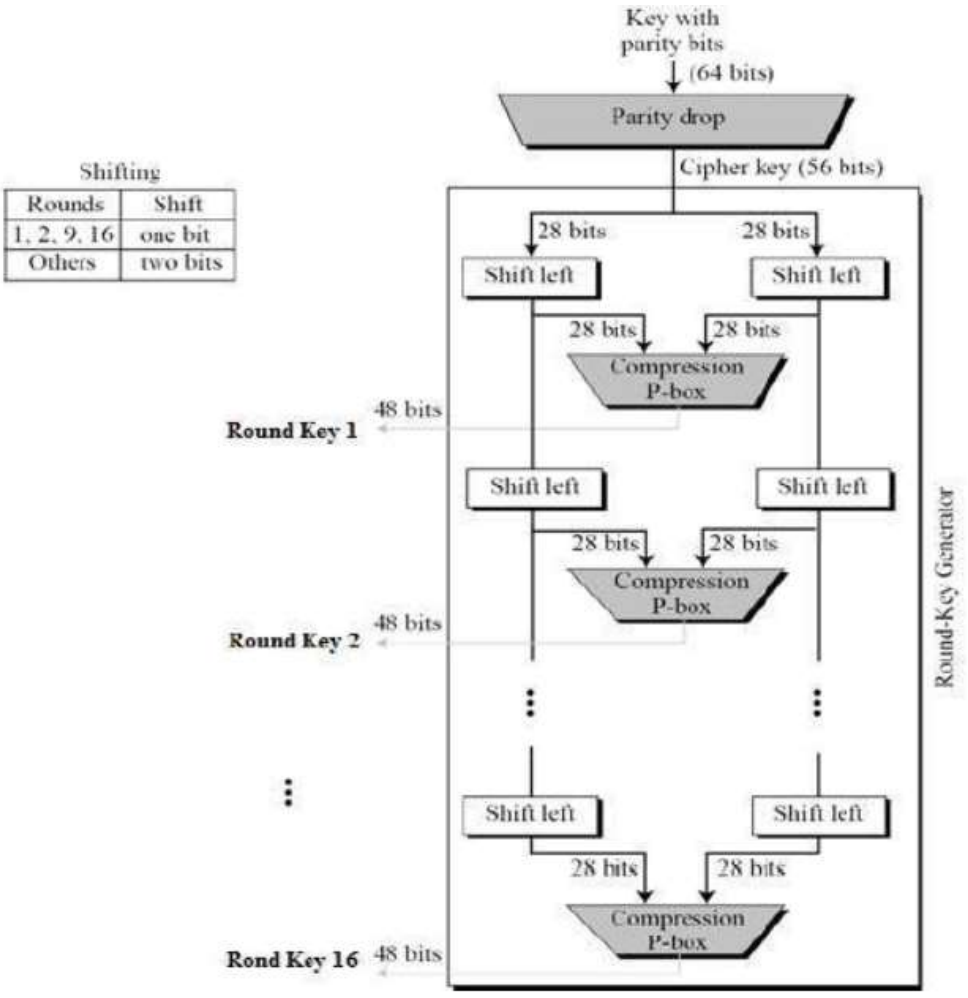


- There are a total of eight S-box tables. The output of all eight s-boxes is then combined in to 32 bit section.

Key Generation

The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key. The process of key generation is depicted in the following illustration −

The logic for Parity drop, shifting, and Compression P-box is given in the DES description.



Shifting

Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits

Literature Survey

C.H.Meyer, S.M.Mat.yas (1981) *discussed the personal verification processes at different institutions in an interchange environment are isolated from one another. It is assumed that only information stored on the bank card and information remembered by a sys–tern user are employed for personal verification. It is shown that only through the use of a quantity stored on the bank card will the set of required criteria be satisfied. With a personal key, the same degree of isolation can be achieved for authentication of transaction request messages sent from the entry point to the issues [1].*

Dan Zhu (2002) *analyzed about modern financial institutions have cashed in on the electronic business opportunities of the Internet by developing numerous payment systems to meet various payment service requirements. In this paper, we examine the function and operation flow of the electronic funds transfer process as well as its security control mechanism. To evaluate telecommunication and data security techniques, a standard leading inter-bank payment system called the Society for Worldwide Inter-bank Financial Telecommunications System is introduced. Some important security features are investigated in detail [2].*

Mintu Philip, Asha Das (2011*) Chaotic Encryption Method seems to be much better than traditional encryption methods used today. Chaotic encryption is the new direction of cryptography. It makes use of chaotic system properties such as sensitive to initial condition and loss of information. Many chaos-based encryption methods have been presented and discussed in the last two decades. In order to reach higher performance, these methods take advantage of the more and more complex behavior of chaotic signals[4].*

Mohammed Abudallah MdAysan, Fareed Hassan Almalki, Abdullah Mohammed Almalki (2014) *This paper proposes a symmetric key cryptosystem based on the simple mathematical logarithm function. The proposed system benefits from the algebraic properties of logx such as non-commutative, high computational speed and high flexibility in selecting keys which make the Discrete Logarithm Problem. Also the encrypted text converted into binary numbers to make harder to understand by the backer. This method will be suitable in any business house, government sectors, communication network, defense network system, sensor networks etc [6].*

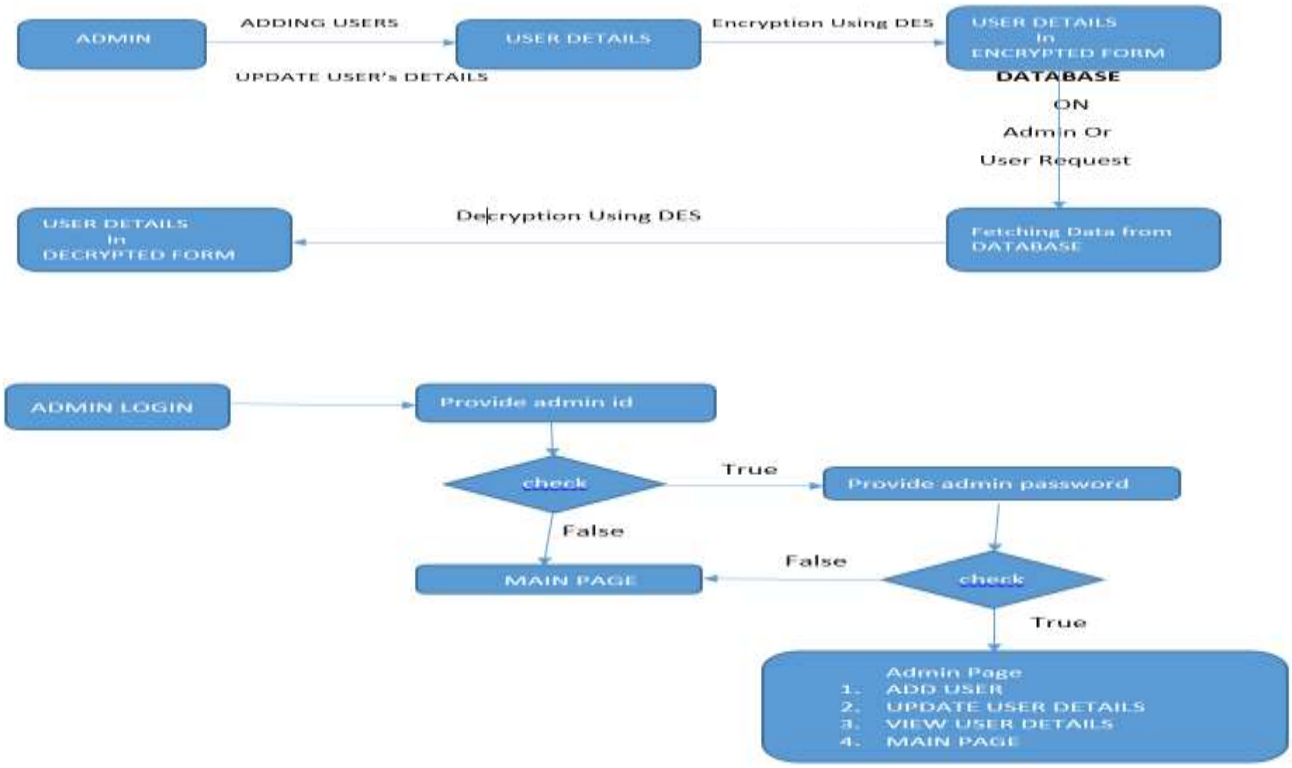
Proposed Method:

The proposed project presents a system in which the user just has successfully sign in into the application select desired amount to be transfer or check any received amount via Electronic Fund Transfer. The system uses DES Data Encryption Standard for security, the fund gets transferred easily and within no time with the help of Single portable card. This process makes secure payment transfer with in networks (International or nationwide).

Procedure:

1. Admin is only allowed to register new user.
2. Admin can check the account details of all the registered users.
3. Admin can update the accounts of the users.
4. Key for encryption and decryption is known only the Admin.
5. Only registered user can accesses the application.
6. User need to enter user id and password generated by bank to access the account.
7. After accessing the account, user can either transfer funds or check updates about the received fund.
8. All the confidential data will be stored in encrypted form.
9. While retrieving the data, it will be decrypted before showing the result.
10. Data will be transferred over internet in encrypted form using DES.

BLOCK DIAGRAM



SAMPLE OUTPUT

Main Page: It will contain links to administrator as well as user page.

```
.....................MAIN PAGE.................................................

enter 1 for admin page, 2 for user, 3 for exit: 1
ENTER ADMIN ID: acd

WRONG ADMIN ID.....................

enter 1 for admin page, 2 for user, 3 for exit: 1
ENTER ADMIN ID: admin234
Enter 10 digit password: **********
```

To open administrator page, admin needs to provide admin id and 10 digit password.

Administrator Page:

```
.............................ADMINISTRATOR PAGE.................................

Choose the option:
1) To add user
2)To Display user details
3)To go to Main Page
OPTION:  1
```

It contains different domains for different applications.

1) To add user

```
Enter account number: 34ddg
Enter name: ghg
enter age :25
enter initial balance: 500
enter the password: qwert
```

2) To Display user details

```
press 1 for encrypted display,2 for decrypted display,3 for exit: 1
account: »Bßbf
e:  Q.
age: 25
balance: 500
password: ╚ÉÖÄ¿
press 1 for encrypted display,2 for decrypted display,3 for exit: 2

account: 34ddg

name: ghg
age: 25
balance: 500
password: qwert   press 1 for encrypted display,2 for decrypted display,3 for exit:
```

Confidential details in encrypted and decrypted form.

USER PAGE:

```
.....................................USER PAGE.....................................

enter account number: 34ddg

account found

enter the password: *****
correct password


press 1 for money transfer,2 to check details,3 for main page,4 for exit
```

```
USER DETAILS
name: abhishek
account: 23gcc
balance: 500
```

USER DETAILS on choosing option 2

```
.....................................USER PAGE.....................................

enter account number: 23gcc

account found

enter the password: *****
correct password


press 1 for money transfer,2 to check details,3 for main page,4 for exit 1
Enter target account number:  98ikju

Encrypted target account: K>ft╔¥▯,
Target account available
enter the money want to transfer: 300

TRANSACTION COMPLETE
```

Transaction page

```
press 1 for money transfer,2 to check details,3 for main page,4 for exit 2

USER DETAILS
name: qwert
account: 23gcc
balance: 200
```

After transaction, balance of user "qwert" reduce to 200.

Result and Analysis

1.) Des uses 56 bits key which led to possibility of 2^{56} combination which is difficult to be attacked by simple computer. It requires powerful computer and parallel computing mechanism to break the Des. So, it was very much useful encrypted method when it was implemented originally.

2.) Des takes a lot of time to implement in comparison to others encryption.

3.) Security offered by Des can be enhanced by increasing the number of bit keys.

4.)

Analysis of DES over last two decades:

In 1998 under the direction of John Gilmore of the EFF, a team spent $220,000 and built a machine that can go through the entire 56-bit DES key space in an average of 4.5 days.

On July 17, 1998, a machine was built that cracked a 56-bit key in 56 hours. The computer, called Deep Crack, uses 27 boards each containing 64 chips, and is capable of testing 90 billion keys a second.

Despite this, on June 8, 1998, Robert Litt, principal associate deputy attorney general at the Department of Justice, denied it was possible for the FBI to crack DES: "Let me put the technical problem in context: It took 14,000 Pentium computers working for four months to decrypt a single message . . . . We are not just talking FBI and NSA [needing massive computing power], we are talking about every police department." Which is very costly and not feasible on daily basis.

The reason behind decrease in popularity of time taken by Des to encrypt the message is much higher than the other encryption method.

References:

1) C.H.Meyer, S.M.Mat.yas,R.E.Lennon, "Required Cryptographic Authentication criteria for Electronic Funds Transfer System", CH1629-5/81/089, IEEE, in 1981.

2) Dan Zhu, "Security control in Inter-Bank Fund Transfer", Journal of Electronic Commerce Research, VOL. 3, NO. 1, 2002.

3) Q.V. Lawande, B. R. Ivan, S. D. Dhodapkar, Chaos Based Cryptography: A New Approach to secure Communications,Asian aerosol conference (AAC) No. 258,July 2005.

4) Mintu Philip,Asha Das, "Survey: Image Encryption using Chaotic Cryptography Schemes", IJCA Special Issue on "Computational Science - New Dimensions & Perspectives" NCCSE, 2011.

5) Palmer, J. W. and Griffith, D. A. "An Emerging Model of Web Site Design for Marketing", Communication of the ACM, Vol. 41, No.3, pp. 45-51, 1998.

6) Mohammed AbdallahMdAysan, Fareed Hassan Almalki , Abdullah Mohammed Almalki, "New Symmetric key cryptography algorithm using simple logarithm and binary digits", International Journal of Multidisciplinary Research Academy, Vol.4 issue 6, (in printing) Accepted in March 2014.

7) Johnson, J. Z. "Network Security Programs: Process and Metrics for the Real-World", White paper, Internet SecuritySystems, Inc, 1998.

8) Kalakota R., A. Whinston, "Frontiers of Electronic Commerce", Addision Wesley, MA, 1996.

9) Ki HyoungKo, Sang Jin Lee, Jung HeeCheon, Jae Woo Han, Ju-sung Kang, and Choonsik Park, "New Public-KeyCryptosystem Using Braid Groups, Advances in cryptology", 20th annual International CryptologyConference, Santa Barbara, California, USA, August 20-24, 2000

CODE:

```
#include<stdio.h>
#include<iostream>
#include<stdlib.h>
#include<conio.h>
#include<cstring>
using namespace std;
int
key[64]={0,0,0,1,0,0,1,1,0,0,1,1,0,1,0,0,0,0,1,0,1,0,1,1,1,0,1,1,1,1,0,0,1,1,0,0,1,1,0,1,1,1,0,1,1,1,1,0,0,1,1,0,1,1,1,1,1,1,1,1,0,0,0,
1};
class Des
{
public:
   int
keyi[16][48],total[64],left[32],right[32],ck[28],dk[28],expansion[48],z[48],xor1[48],sub[32],p[32],xor2[32],temp[64],pc1[56],ip[6
4],inv[8][8];
   char final[1000];
   void IP();
   void PermChoice1();
   void PermChoice2();
   void Expansion();
   void inverse();
   void xor_two();
   void xor_oneE(int);
   void xor_oneD(int);
   void substitution();
   void permutation();
   void keygen();
   char * Encrypt(char *);
   char * Decrypt(char *);
};
void Des::IP() //Initial Permutation
{
   int k=58,i;
   for(i=0; i<32; i++)
   {
      ip[i]=total[k-1];
      if(k-8>0)  k=k-8;
      else      k=k+58;
   }
   k=57;
   for( i=32; i<64; i++)
   {
      ip[i]=total[k-1];
      if(k-8>0)   k=k-8;
      else     k=k+58;
   }
}
void Des::PermChoice1() //Permutation Choice-1
{
   int k=57,i;
   for(i=0; i<28; i++)
   {
      pc1[i]=key[k-1];
      if(k-8>0)    k=k-8;
      else      k=k+57;
   }
```

```
   k=63;
   for( i=28; i<52; i++)
   {
      pc1[i]=key[k-1];
      if(k-8>0)    k=k-8;
      else         k=k+55;
   }
   k=28;
   for(i=52; i<56; i++)
   {
      pc1[i]=key[k-1];
    k=k-8;
   }
}
void Des::Expansion()
{
   int exp[8][6],i,j,k;
   for( i=0; i<8; i++)
   {
      for( j=0; j<6; j++)
      {
         if((j!=0)||(j!=5))
         {
            k=4*i+j;exp[i][j]=right[k-1];
         }
         if(j==0)
         {
            k=4*i;exp[i][j]=right[k-1];
         }
         if(j==5)
         {
            k=4*i+j;exp[i][j]=right[k-1];
         }
      }
   }
   exp[0][0]=right[31];exp[7][5]=right[0];k=0;
   for(i=0; i<8; i++)
      for(j=0; j<6; j++)
         expansion[k++]=exp[i][j];
}
void Des::PermChoice2()
{
   int per[56],i,k;
   for(i=0; i<28; i++) per[i]=ck[i];
   for(k=0,i=28; i<56; i++) per[i]=dk[k++];
   z[0]=per[13];z[1]=per[16];z[2]=per[10];z[3]=per[23];z[4]=per[0];z[5]=per[4];z[6]=per[2];z[7]=per[27];z[8]=per[14];z[9]=per[5];

z[10]=per[20];z[11]=per[9];z[12]=per[22];z[13]=per[18];z[14]=per[11];z[15]=per[3];z[16]=per[25];z[17]=per[7];z[18]=per[15];z[1
9]=per[6];

z[20]=per[26];z[21]=per[19];z[22]=per[12];z[23]=per[1];z[24]=per[40];z[25]=per[51];z[26]=per[30];z[27]=per[36];z[28]=per[46];z
[29]=per[54];

z[30]=per[29];z[31]=per[39];z[32]=per[50];z[33]=per[46];z[34]=per[32];z[35]=per[47];z[36]=per[43];z[37]=per[48];z[38]=per[38]
;z[39]=per[55];
   z[40]=per[33];z[41]=per[52];z[42]=per[45];z[43]=per[41];z[44]=per[49];z[45]=per[35];z[46]=per[28];z[47]=per[31];
}
void Des::xor_oneE(int round)
```

```
{
   int i;
   for(i=0; i<48; i++)
      xor1[i]=expansion[i]^keyi[round-1][i];
}
void Des::xor_oneD(int round)
{
   int i;
   for(i=0; i<48; i++)
      xor1[i]=expansion[i]^keyi[16-round][i];
}

void Des::substitution()
{
   int
s1[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};
   int
s2[4][16]={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  };
   int
s3[4][16]={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  };
   int
s4[4][16]={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   };
   int
s5[4][16]={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    };
   int
s6[4][16]={12,1,10,15,9,2,6,8,0,13,3,4,14,7,5,11,10,15,4,2,7,12,9,5,6,1,13,14,0,11,3,8,9,14,15,5,2,8,12,3,7,0,4,10,1,13,11,6,
            4,3,2,12,9,5,15,10,11,14,1,7,6,0,8,13    };
   int
s7[4][16]={4,11,2,14,15,0,8,13,3,12,9,7,5,10,6,1,13,0,11,7,4,9,1,10,14,3,5,12,2,15,8,6,1,4,11,13,12,3,7,14,10,15,6,8,0,5,9,2,
            6,11,13,8,1,4,10,7,9,5,0,15,14,2,3,12    };
   int
s8[4][16]={13,2,8,4,6,15,11,1,10,9,3,14,5,0,12,7,1,15,13,8,10,3,7,4,12,5,6,11,0,14,9,2,7,11,4,1,9,12,14,2,0,6,10,13,15,3,5,8,
            2,1,14,7,4,10,8,13,15,12,9,0,3,5,6,11    };
   int a[8][6],k=0,i,j,p,q,count=0,g=0,v;
   for(i=0; i<8; i++)
      for(j=0; j<6; j++)
         a[i][j]=xor1[k++];
   for( i=0; i<8; i++)
   {
      p=1;q=0;k=(a[i][0]*2)+(a[i][5]*1);j=4;
      while(j>0)
      {
         q=q+(a[i][j]*p);
         p=p*2;j--;
      }
      count=i+1;
      switch(count)
      {
      case 1:
         v=s1[k][q];  break;
      case 2:
         v=s2[k][q];  break;
      case 3:
```

```
            v=s3[k][q];  break;
        case 4:
            v=s4[k][q]; break;
        case 5:
            v=s5[k][q]; break;
        case 6:
            v=s6[k][q]; break;
        case 7:
            v=s7[k][q]; break;
        case 8:
            v=s8[k][q]; break;
        }

        int d,i=3,a[4];
        while(v>0)
        {
            d=v%2;  a[i--]=d; v=v/2;
        }
        while(i>=0)
            a[i--]=0;
        for(i=0; i<4; i++)
            sub[g++]=a[i];
    }
}
void Des::permutation()
{
    p[0]=sub[15];
    p[1]=sub[6];
    p[2]=sub[19];
    p[3]=sub[20];
    p[4]=sub[28];
    p[5]=sub[11];
    p[6]=sub[27];
    p[7]=sub[16];
    p[8]=sub[0];
    p[9]=sub[14];
    p[10]=sub[22];
    p[11]=sub[25];
    p[12]=sub[4];
    p[13]=sub[17];
    p[14]=sub[30];
    p[15]=sub[9];
    p[16]=sub[1];
    p[17]=sub[7];
    p[18]=sub[23];
    p[19]=sub[13];
    p[20]=sub[31];
    p[21]=sub[26];
    p[22]=sub[2];
    p[23]=sub[8];
    p[24]=sub[18];
    p[25]=sub[12];
    p[26]=sub[29];
    p[27]=sub[5];
    p[28]=sub[21];
    p[29]=sub[10];
    p[30]=sub[3];
    p[31]=sub[24];
```

```
}
void Des::xor_two()
{
   int i;
   for(i=0; i<32; i++)
      xor2[i]=left[i]^p[i];
}
void Des::inverse()
{
   int p=40,q=8,k1,k2,i,j;
   for(i=0; i<8; i++)
   {
      k1=p;k2=q;
      for(j=0; j<8; j++)
      {
         if(j%2==0)
         {
          inv[i][j]=temp[k1-1];k1=k1+8;
         }
         else
         {
           inv[i][j]=temp[k2-1];k2=k2+8;
         }
      }
      p=p-1;q=q-1;
   }
}
char * Des::Encrypt(char *Text1)
{
   int i,a1,j,nB,m,iB,k,K,B[8],n,t,d,round;
   char *Text=new char[1000];
   strcpy(Text,Text1);
   i=strlen(Text);
   int mc=0;
   a1=i%8;
   if(a1!=0)
      for(j=0; j<8-a1; j++,i++)
         Text[i]=' ';
   Text[i]='\0';
   keygen();
   for(iB=0,nB=0,m=0; m<(strlen(Text)/8); m++)
   {
      for(iB=0,i=0; i<8; i++,nB++)
      {
         n=(int)Text[nB];
         for(K=7; n>=1; K--)
         {
            B[K]=n%2;
            n/=2;
         }
         for(; K>=0; K--) B[K]=0;
         for(K=0; K<8; K++,iB++)
            total[iB]=B[K];
      }
      IP();
      for(i=0; i<64; i++)
         total[i]=ip[i];
      for(i=0; i<32; i++)
```

```
        left[i]=total[i];
    for(; i<64; i++)
        right[i-32]=total[i];
    for(round=1; round<=16; round++)
    {
        Expansion();
        xor_oneE(round);
        substitution();
        permutation();
        xor_two();
        for(i=0; i<32; i++)
            left[i]=right[i];
        for(i=0; i<32; i++)
            right[i]=xor2[i];
    }
    for(i=0; i<32; i++)
        temp[i]=right[i];
    for(; i<64; i++)
        temp[i]=left[i-32];
    inverse();
    k=128;d=0;
    for(i=0; i<8; i++)
    {
        for(j=0; j<8; j++)
        {
            d=d+inv[i][j]*k;k=k/2;
        }
        final[mc++]=(char)d;
        k=128;
        d=0;
    }
  }
  final[mc]='\0';
  return(final);
}
char * Des::Decrypt(char *Text1)
{
  int i,a1,j,nB,m,iB,k,K,B[8],n,t,d,round;
  char *Text=new char[1000];
  unsigned char ch;
  strcpy(Text,Text1);
  i=strlen(Text);
  keygen();
  int mc=0;
  for(iB=0,nB=0,m=0; m<(strlen(Text)/8); m++)
  {
    for(iB=0,i=0; i<8; i++,nB++)
    {
        ch=Text[nB];
        n=(int)ch;
        for(K=7; n>=1; K--)
        {
            B[K]=n%2;
            n/=2;
        }
        for(; K>=0; K--)
            B[K]=0;
        for(K=0; K<8; K++,iB++)
```

```
            total[iB]=B[K];
        }
        IP();
        for(i=0; i<64; i++)
           total[i]=ip[i];
        for(i=0; i<32; i++)
           left[i]=total[i];
        for(; i<64; i++)
           right[i-32]=total[i];
        for(round=1; round<=16; round++)
        {
           Expansion();
           xor_oneD(round);
           substitution();
           permutation();
           xor_two();
           for(i=0; i<32; i++)
              left[i]=right[i];
           for(i=0; i<32; i++)
              right[i]=xor2[i];
        }
        for(i=0; i<32; i++)
           temp[i]=right[i];
        for(; i<64; i++)
           temp[i]=left[i-32];
        inverse();
        k=128;d=0;
        for(i=0; i<8; i++)
        {
           for(j=0; j<8; j++)
           {
              d=d+inv[i][j]*k;
              k=k/2;
           }
           final[mc++]=(char)d;
           k=128;d=0;
        }
     }
     final[mc]='\0';
     char *final1=new char[1000];
     for(i=0,j=strlen(Text); i<strlen(Text); i++,j++)
        final1[i]=final[j];
     final1[i]='\0';
     return(final);
}

void Des::keygen()
{
     PermChoice1();
     int i,j,k=0;
     for(i=0; i<28; i++)
        ck[i]=pc1[i];
     for(i=28; i<56; i++)
     {
        dk[k]=pc1[i];
        k++;
     }
     int noshift=0,round;
```

```
   for(round=1; round<=16; round++)
   {
      if(round==1||round==2||round==9||round==16)
         noshift=1;
      else
         noshift=2;
      while(noshift>0)
      {
         int t;
         t=ck[0];
         for(i=0; i<28; i++)
            ck[i]=ck[i+1];
         ck[27]=t;t=dk[0];
         for(i=0; i<28; i++)
            dk[i]=dk[i+1];
         dk[27]=t;
         noshift--;
      }
      PermChoice2();
      for(i=0; i<48; i++)
         keyi[round-1][i]=z[i];
   }
}

struct user
{
   char user_account[5000];
   char user_name[500];
   int user_age;
   int user_balance;
   char password[500];
   struct user *next;
}*first=NULL;

int check_admin_id(string id)
{
            string check_id;
            int count=0;
            cin.ignore();
            cout<<"ENTER ADMIN ID: ";
            getline(cin,check_id);
            if(id.length()==check_id.length())
            {
                        for(int i=0;i<id.length();i++)
                        {
                                    if(id[i]==check_id[i])
                                                count++;
                                    else
                                                break;
                        }
                        if(count==id.length())
                                    return 1;
                        else
                                    return 0;
            }
            else
                        return 0;
}
```

```
int check_admin_password(string password)
{
	string check_password;
	cout<<"Enter 10 digit password: ";
	char a;
	for(int i=0;i<10;i++)
	{
		a=getch();
		check_password=check_password+a;
		cout<<"*";
	}
	int count=0;
	if(password.length()==check_password.length())
	{
		for(int i=0;i<check_password.length();i++)
		{
			if(password[i]==check_password[i])
				count++;
			else
				break;
		}
		if(count==check_password.length())
			return 1;
		else
			return 0;
	}
	else
		return 0;

}

int main()
{
   system("cls");
   Des d1,d2;
   char *str=new char[1000];
   char *str1=new char[1000];

   string admin_id="admin234";
	char password[11]="qwertyuiop";

	string check_password;
	int check;
	int select;
	cout<<"\n.....MAIN PAGE.....";
	while(1)
   {
      //system("cls");
      cout<<endl;
      cout<<"\nenter 1 for admin page, 2 for user, 3 for exit: ";
      cin>>select;
      switch(select)
      {
      case 1:
         {
            check=check_admin_id(admin_id);
            if(check==1)
```

```
{
   check=check_admin_password(password);
   if(check==1)
   {
      cout<<"\n\n\t Welcome to ADMINISTRATOR PAGE ";
      getch();
      system("cls");
      cout<<".....ADMINISTRATOR PAGE....."<<endl;
      int num;
      while(1)
      {
         cout<<"\nChoose the option: \n1) To add user\n2)To Display user details\n3)To go to Main Page\nOPTION:  ";
         cin>>num;
         switch(num)
         {
         case 1:
           {
              struct user *temp;
              temp=new(struct user);
              char account[10];
              char name[45];
              int age;
              int bal;
              char pass[10];
              cout<<"\nEnter account number: ";
              cin>>account;
              cout<<"Enter name: ";
              cin>>name;
              cout<<"enter age :";
              cin>>age;
              cout<<"enter initial balance: ";
              cin>>bal;
              cout<<"enter the password: ";
              cin>>pass;
              strcpy(str,account);
              str1=d1.Encrypt(str);
              strcpy(temp->user_account,str1);
              //cout<<"\nencrypted id: "<<temp->user_account;
              strcpy(str,name);
              str1=d1.Encrypt(str);
              strcpy(temp->user_name,str1);
              //cout<<"\nencrypted name: "<<temp->user_name<<endl;
              strcpy(str,temp->user_account);
              str1=d1.Decrypt(str);
              //cout<<"\ndecrypted id is : "<<str1;
              strcpy(str,pass);
              str1=d1.Encrypt(str);
              strcpy(temp->password,str1);
              temp->user_age=age;
              temp->user_balance=bal;
              temp->next=NULL;
              if(first==NULL)
              {
                 first=temp;
              }
              else
              {
                 struct user *s;
```

```
            s=new(struct user);
            s=first;
            while(s->next!=NULL)
               s=s->next;
            s->next=temp;
          }

          break;
       }
    case 2:
       {
          int num;
          system("cls");
          while(1)
          {
             //system("cls");
             cout<<"press 1 for encrypted display,2 for decrypted display,3 for exit: ";
             cin>>num;
             switch(num)
             {
             case 1:
              {
                 struct user *temp;
                 temp=new(struct user);
                 if(first==NULL)
                    cout<<"\nNO RECORDS AVAILABLE ";
                 else
                 {
                    temp=first;
                    while(temp!=NULL)
                    {
                       cout<<"account: "<<temp->user_account<<endl;
                       cout<<"name: "<<temp->user_name<<endl;
                       cout<<"age: "<<temp->user_age<<endl;
                       cout<<"balance: "<<temp->user_balance<<endl;
                       cout<<"password: "<<temp->password<<endl;
                       temp=temp->next;
                    }
                 }
                 break;
              }
             case 2:
              {
                 struct user *temp;
                 temp=new(struct user);
                 if(first==NULL)
                    cout<<"\nNO RECORDS AVAILABLE"<<endl;
                 else
                 {
                    temp=first;
                    while(temp!=NULL)
                    {

                    strcpy(str,temp->user_account);
                    str1=d1.Decrypt(str);
                    cout<<"\naccount: "<<str1<<endl;
                    strcpy(str,temp->user_name);
                    str1=d1.Decrypt(str);
```

```
                        cout<<"\nname: "<<str1<<endl;
                        cout<<"age: "<<temp->user_age<<endl;
                        cout<<"balance: "<<temp->user_balance<<endl;
                        cout<<"password: "<<d1.Decrypt(strcpy(str,temp->password));
                        temp=temp->next;
                        }

                      }
                      break;
                    }
                   case 3:
                    {
                      main();
                    }
                  }
                 }
                 break;
               }
               case 3:
                {
                  //exit(0);
                  main();
                }
            }
          }
        }
        else
          cout<<"\nWRONG PASSWORD.....";
    }
    else
      cout<<"\nWRONG ADMIN ID.....";
    break;
}
case 2:
    {
      system("cls");
      cout<<".....USER PAGE.....\n";
      int check=0;
      char account[10];
      char name[45];
      int age;
      int bal;
      char pass[10];
      cout<<"\nenter account number: ";
      cin>>account;
      strcpy(str,account);
      str1=d1.Encrypt(str);
      struct user *temp;
      temp=new(struct user);
      if(first==NULL)
        cout<<"\nNO RECORDS AVAILABLE"<<endl;
      else
      {
        temp=first;
        while(temp!=NULL)
        {
          if(strcmp(temp->user_account,str1)==0)
          {
```

```
        cout<<"\naccount found"<<endl;
        check=1;
        break;
      }
      temp=temp->next;
    }
    if(check!=1)
      cout<<"\ninvalid account available"<<endl;
    else
    {
      check=0;
      cout<<"\nenter the password: ";
      char sk[5];
      char a;
      for(int i=0;i<5;i++)
      {
        a=getch();
        sk[i]=a;
        cout<<"*";
      }
      //cin>>pass;
      //pass=sk;
      strcpy(str,sk);
      str1=d1.Encrypt(str);
      temp=first;
      if(strcmp(temp->password,str1)==0)
        check=1;
      if(check!=1)
        cout<<"\nWrong password"<<endl;
      else
      {
        cout<<"\ncorrect password"<<endl;
        int num;
        while(1)
        {
          cout<<"\n\npress 1 for money transfer,2 to check details,3 for main page,4 for exit ";
          cin>>num;
          switch(num)
          {
            case 1:
            {
              int flag=0;
              struct user *target;
              target=new(struct user);
              char account[10];
              cout<<"Enter target account number:  ";
              cin>>account;
              strcpy(str,account);
              str1=d1.Encrypt(str);
              cout<<"\nEncrypted target account: "<<str1;
              target=first;
              while(target!=NULL)
              {
                if(strcmp(target->user_account,str1)==0)
                {
                  flag=1;
                  break;
                }
```

```
                target=target->next;
              }
              if(flag==0)
              {
                cout<<"\nINVALID ACCOUNT NUMBER";
                getch();                              }

              else
              {
                cout<<"\nTarget account available ";
                int money;
                cout<<"\nenter the money want to transfer: ";
                cin>>money;
                if(money > temp->user_balance)
                  cout<<"\nInsufficient balance";
                else
                {
                  temp->user_balance-=money;
                  //target->user_balance+=money;
                  target->user_balance+=money;
                  cout<<"\nTRANSACTION COMPLETE\n";
                  getch();
                }
              }
              break;
            }
            case 2:
              {
                cout<<"\nUSER DETAILS"<<endl;
                strcpy(str,temp->user_name);
                str1=d1.Decrypt(str);
                cout<<"name: "<<str1<<endl;
                cout<<"account: "<<d1.Decrypt(strcpy(str,temp->user_account))<<endl;
                cout<<"balance: "<<temp->user_balance<<endl;
                getch();
              }
            case 3:
              {
                main();
              }
            case 4:
              {
                exit(0);
              }
          }
        }
      }
    }
    break;
  }
case 3:
  {
    exit(0);
  }
 }
}
```