

Lampiran A

Listing Program

Listing UMain.pas

```
unit UMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Menus;

type
  TMain = class(TForm)
    MainMenu1: TMainMenu;
    Fungsi1: TMenuItem;
    Enkripsi1: TMenuItem;
    Dekripsi1: TMenuItem;
    N1: TMenuItem;
    Keluar1: TMenuItem;
    About1: TMenuItem;
    Label1: TLabel;
    Label2: TLabel;
    procedure Label1MouseEnter(Sender: TObject);
    procedure Label1MouseLeave(Sender: TObject);
    procedure Label2MouseEnter(Sender: TObject);
    procedure Label2MouseLeave(Sender: TObject);
    procedure Label1Click(Sender: TObject);
    procedure Keluar1Click(Sender: TObject);
    procedure Label2Click(Sender: TObject);
    procedure Enkripsi1Click(Sender: TObject);
    procedure Dekripsi1Click(Sender: TObject);
    procedure About1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Main: TMain;

implementation

uses UEnkripsi, UDekripsi, UAbout;
```

{ \$R *.dfm }

```
procedure TMain.Label1MouseEnter(Sender: TObject);  
begin  
Label1.Font.Color:=clRed;  
end;
```

```
procedure TMain.Label1MouseLeave(Sender: TObject);  
begin  
Label1.Font.Color:=clBlack;  
end;
```

```
procedure TMain.Label2MouseEnter(Sender: TObject);  
begin  
Label2.Font.Color:=clRed;  
end;
```

```
procedure TMain.Label2MouseLeave(Sender: TObject);  
begin  
Label2.Font.Color:=clBlack;  
end;
```

```
procedure TMain.Label1Click(Sender: TObject);  
begin  
frmEnkripsi.ShowModal;  
end;
```

```
procedure TMain.Keluar1Click(Sender: TObject);  
begin  
Application.Terminate;  
end;
```

```
procedure TMain.Label2Click(Sender: TObject);  
begin  
frmDekripsi.ShowModal;  
end;
```

```
procedure TMain.Enkripsi1Click(Sender: TObject);  
begin  
frmEnkripsi.ShowModal;  
end;
```

```
procedure TMain.Dekripsi1Click(Sender: TObject);  
begin
```

```
frmDekripsi.ShowModal;  
end;
```

```
procedure TMain.About1Click(Sender: TObject);  
begin  
  frmAbout.ShowModal;  
end;
```

```
end.
```

Listing UEnkripsi.pas

```
unit UEnkripsi;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, StdCtrls;

type
  TfrmEnkripsi = class(TForm)
    Label1: TLabel;
    edPass: TEdit;
    Label2: TLabel;
    btEncrypt: TButton;
    EOpen: TOpenDialog;
    ESave: TSaveDialog;
    btSave: TButton;
    Label4: TLabel;
    Label3: TLabel;
    btOpen: TButton;
    Plain: TMemo;
    Cipher: TMemo;
    times: TEdit;
    Finished: TEdit;
    Start: TEdit;
    Label5: TLabel;
    btClose: TButton;
    StatusBar1: TStatusBar;
    Label6: TLabel;
    Label7: TLabel;
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    PrivD: TEdit;
    PrivN: TEdit;
    PubE: TEdit;
    PubN: TEdit;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    BtGenKey: TButton;
    BtSign: TButton;
```

```

    Signtx: TEdit;
    Signature: TEdit;
    Label12: TLabel;
    procedure btOpenClick(Sender: TObject);
    procedure btSaveClick(Sender: TObject);
    procedure edPassChange(Sender: TObject);
    procedure btCloseClick(Sender: TObject);
    procedure btEncryptClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure BtGenKeyClick(Sender: TObject);
    procedure BtSignClick(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    frmEnkripsi: TfrmEnkripsi;

implementation

uses StrUtils, Cast128, RSATools;

{$R *.dfm}

procedure TfrmEnkripsi.btOpenClick(Sender: TObject);
begin
    if EOpen.Execute then
        begin
            Plain.Lines.LoadFromFile(EOpen.FileName);
        end;
end;

procedure TfrmEnkripsi.btSaveClick(Sender: TObject);
begin
    if ESave.Execute then
        begin
            Cipher.Lines.SaveToFile(ESave.FileName);
            ShowMessage('Cipher Text telah disimpan dengan
nama'+chr(10)+chr(13)+ESave.FileName);
        end;
end;

```

```

procedure TfrmEnkripsi.edPassChange(Sender: TObject);
begin
if Length(edPass.Text) > 16 then
begin
    ShowMessage('Kata Kunci Maksimal Terdiri Dari 16 Karakter');
    edPass.Text:=LeftStr(edPass.Text,16);
end;
end;

procedure TfrmEnkripsi.btCloseClick(Sender: TObject);
begin
frmEnkripsi.Close;
end;

procedure TfrmEnkripsi.btEncryptClick(Sender: TObject);
var slama:string;
    awal,akhir,lama:TDateTime;

begin
if edPass.Text = '' then
    ShowMessage('Password Belum Terisi')
else
    if Plain.Text = '' then
        ShowMessage('Plain Text Belum Terisi')
    else
        if Signtx.Text = '' then
            if Application.MessageBox('Anda belum menambahkan Digital
Signature'+chr(10)+chr(13)
+ 'Apakah Dokumen akan di enkripsi ?','Warning',MB_OKCANCEL) <>
IDOK then
                exit
            else
                begin
                    Cipher.Clear;
                    awal:=Time;
                    Start.Text:=TimeToStr(awal);
                    Cipher.Lines.Add(Encrypt(Signtx.Text+Plain.Text,edPass.Text));
                    akhir:=Time;
                    Finished.Text:=TimeToStr(akhir);
                    lama:=(akhir-awal)*86000;
                    str(lama:12:4,slama);
                    times.Text:=slama;
                end
            else

```

```

begin
  Cipher.Clear;
  awal:=Time;
  Start.Text:=TimeToStr(awal);
  Cipher.Lines.Add(Encrypt(Signtx.Text+Plain.Text,edPass.Text));
  akhir:=Time;
  Finished.Text:=TimeToStr(akhir);
  lama:=(akhir-awal)*86000;
  str(lama:12:4,slama);
  times.Text:=slama;
end;
end;

procedure TfrmEnkripsi.FormShow(Sender: TObject);
begin
  edPass.Clear; Start.Clear;
  Plain.Clear; Finished.Clear;
  Cipher.Clear; times.Clear;
  Signtx.Clear;
  Signature.Clear;
  PrivD.Clear; PrivN.Clear;
  PubE.Clear; PubN.Clear;
end;

procedure TfrmEnkripsi.BtGenKeyClick(Sender: TObject);
var p : Longint; //random prime
    q : Longint; //second random prime that not equal to p
    n : Longint; //p * q
    pi : Longint; //(p - 1)(q - 1)
    e : Longint; //e that relatively prime to pi but less than pi
    d : Longint; //d that d*e congruent to 1 mod pi
    i1 : Longint; //counter

    c : Longint;
    temp1 : Longint;
    temp2 : array of Longint; // temp dynamic array handler that hand selection of e
    temp3 : Longint;
    temp4 : Longint;
    temp5 : Longint; // temp handler
    temp6 : Longint; // temp handler 2
label lagi,ulang;

begin
  lagi:

```



```

p := RdmPrime;
q := RdmPrime;

//Trap handler if p = q
If p = q Then
    GoTo lagi;

n := p * q;

PrivN.Text := IntToStr(n);
PubN.Text := IntToStr(n);
pi := (p - 1) * (q - 1);

//search for e
c := pi - 1;
SetLength(temp2,c);

For e := 2 To (pi - 1) do
begin
    temp6 := gcd(pi, e);
    If temp6 = 1 Then
        begin
            temp2[c] := e;
            c := c - 1;
        end;
    end;

//random selection of e
ulang:
    Randomize;
    temp3 := Round((pi - 1) * (Random(10000)/10000));
    temp4 := temp2[temp3];
    If (temp4 = 0) Or (temp4 = Null) Then
        GoTo ulang;

//select e that is prime
For i1 := 2 To Round(Sqrt(temp4)) do
begin
    temp5 := temp4 Mod i1;
    If temp5 = 0 Then
        GoTo ulang;
end;

```

```

PubE.Text := IntToStr(temp4);

//determine d such that d*e congruent 1 mod pi and d > 0, d > e
d := Euclid(pi, temp4);
If d < temp4 Then
    GoTo ulang;

PrivD.Text := IntToStr(d);
end;

procedure TfrmEnkripsi.BtSignClick(Sender: TObject);
var d,n,c,i,m,si:longint;
    sign:string;
begin
    if PrivD.Text = " then
        ShowMessage('Generate Key Terlebih Dahulu')
    else
        begin
            if Signature.Text = " then
                ShowMessage('Isi Signature Anda')
            else
                begin
                    Signtx.Clear;
                    d:=StrToInt(PrivD.Text);
                    n:=StrToInt(PrivN.Text);
                    sign:=Signature.Text;
                    c:=length(sign);
                    for i:=1 to c do
                        begin
                            m:=ord(sign[i]);
                            si:=pangkatmod(m,d,n);
                            Signtx.Text:=Signtx.Text+IntToStr(si)+'||';
                        end;
                    ShowMessage('Signature telah disimpan');
                end;
            end;
        end;
end.

```

Listing Udekripsi.pas

```
unit Udekripsi;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, StdCtrls;

type
  TfrmDekripsi = class(TForm)
    Label1: TLabel;
    edPass: TEdit;
    btEncrypt: TButton;
    btOpen: TButton;
    Label4: TLabel;
    Label3: TLabel;
    btSave: TButton;
    DSave: TSaveDialog;
    DOpen: TOpenDialog;
    Plain: TMemo;
    Cipher: TMemo;
    Start: TEdit;
    Finished: TEdit;
    times: TEdit;
    btClose: TButton;
    StatusBar1: TStatusBar;
    Label7: TLabel;
    Label6: TLabel;
    Label5: TLabel;
    Label8: TLabel;
    GroupBox2: TGroupBox;
    Label10: TLabel;
    Label11: TLabel;
    PubE: TEdit;
    PubN: TEdit;
    Button1: TButton;
    Signature: TEdit;
    procedure btOpenClick(Sender: TObject);
    procedure btSaveClick(Sender: TObject);
    procedure btCloseClick(Sender: TObject);
    procedure btEncryptClick(Sender: TObject);
```

```

    procedure FormShow(Sender: TObject);
    procedure edPassChange(Sender: TObject);
    procedure Button1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    frmDekripsi: TfrmDekripsi;

implementation

uses StrUtils, Cast128, RSATools;

{$R *.dfm}

procedure TfrmDekripsi.btOpenClick(Sender: TObject);
begin
    if DOpen.Execute then
        begin
            Cipher.Clear;
            Cipher.Lines.LoadFromFile(DOpen.FileName);
        end;
end;

procedure TfrmDekripsi.btSaveClick(Sender: TObject);
begin
    if DSave.Execute then
        begin
            Plain.Lines.SaveToFile(DSave.FileName);
            ShowMessage('Plain Text telah disimpan dengan
nama'+chr(10)+chr(13)+DSave.FileName);
        end;
end;

procedure TfrmDekripsi.btCloseClick(Sender: TObject);
begin
    frmDekripsi.Close;
end;

procedure TfrmDekripsi.btEncryptClick(Sender: TObject);
var ciphertext,slama,plaintext,tmp,final:string;

```

```

    i,j,k,pjg:integer;
    awal,akhir,lama:TDateTime;
begin
if edPass.Text = " then
    ShowMessage('Password Belum Terisi')
else
if Cipher.Text = " then
    ShowMessage('Cipher Text Belum Terisi')
else
    begin
        for i:=0 to Cipher.Lines.Count-1 do
            ciphertext:=ciphertext+Cipher.Lines[i];
        Plain.Clear;
        awal:=Time;
        Start.Text:=TimeToStr(awal);
        plaintext:=DECRYPT(ciphertext,edPass.Text);
        pjg:=length(plaintext);
        Signature.Clear;
        j:=1; k:=1;
        while j < pjg+1 do
            begin
                tmp:=copy(plaintext,j,2);
                if tmp = '||' then
                    begin
                        tmp:=copy(plaintext,k,j-k);
                        k:=j+2;
                        Signature.Text:=Signature.Text+tmp+'||';
                    end;
                inc(j);
            end;
            final:=copy(plaintext,k,pjg);
            Plain.Lines.Add(final);
            akhir:=Time;
            Finished.Text:=TimeToStr(akhir);
            lama:=(akhir-awal)*86000;
            str(lama:12:4,slama);
            times.Text:=slama;
        end;
    end;

procedure TfrmDekripsi.FormShow(Sender: TObject);
begin
edPass.Clear;
Cipher.Clear;

```

```

Plain.Clear;
PubE.Clear;
PubN.Clear;
Start.Clear;
Finished.Clear;
times.Clear;
end;

procedure TfrmDekripsi.edPassChange(Sender: TObject);
begin
if Length(edPass.Text) > 16 then
begin
ShowMessage('Kata Kunci Maksimal Terdiri Dari 16 Karakter');
edPass.Text:=LeftStr(edPass.Text,16);
end;
end;

procedure TfrmDekripsi.Button1Click(Sender: TObject);
var tmp,sign,signtxt:string;
    m,e,n,signlen:longint;
    k,j:integer;
begin
sign:=Signature.Text;
if Plain.Text = " then
ShowMessage('Dekripsi dokumen terlebih dahulu')
else
if sign = " then
ShowMessage('Dokumen ini tidak mempunyai Digital Signature')
else
begin
if (PubE.Text = "") or (PubN.Text = "") then
ShowMessage('Isi Kunci Public')
else
begin
signlen:=length(sign);
e:=StrToInt(PubE.Text);
n:=StrToInt(PubN.Text);
j:=1; k:=1; signtxt:="";
while j < signlen+1 do
begin
tmp:=copy(sign,j,2);
if tmp = '||' then
begin
tmp:=copy(sign,k,j-k);

```

```

        k:=j+2;
        signtxt:=signtxt+chr(pangkatmod(StrToInt(tmp),e,n));
    end;
    inc(j);
end;
ShowMessage('Digital Signature :'+chr(10)+chr(13)+'"+signtxt+"');
end;
end;
end.

```

Listing Uabout.pas

```
unit UAbout;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type
  TfrmAbout = class(TForm)
    Panel1: TPanel;
    Label1: TLabel;
    Label5: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmAbout: TfrmAbout;

implementation

{$R *.dfm}

procedure TfrmAbout.Button1Click(Sender: TObject);
begin
  frmAbout.Close;
end;

end.
```


Listing Cast128.pas

```
unit Cast128;

interface
uses StrUtils, SysUtils, Tools;

function Encrypt(data: string; key: string) : string;
function Decrypt(data: string; key: string) : string;

implementation

{$I Cast128.inc}
{$R-}

function Encrypt(data: string; key: string) : string;
var
  x, t, z: array[0..3] of DWord;
  key1: array[0..15] of byte;
  i,j,k,len,plainlen,cipherlen,add,Rounds: integer;
  xKey: array[0..31] of DWord;
  u, l, r, valid: DWord;
  OutData, InData: array[0..7] of byte;
  Ciphertext,chip: string;

begin
  {pengolahan kunci}
  len:=length(key);
  if (Len<= 0) or (Len> 16) then
    raise Exception.Create('Panjang Kunci antara 1 sampai 16 huruf');
  if Len<= 10 then
    Rounds:= 12
  else
    Rounds:= 16;
  FillChar(key1,Sizeof(key1),0);
  valid:=0;
  for j:=0 to len do
    begin
      key1[j]:=ord(key[j+1]);
    end;
  x[0]:= key1[0]; x[0]:= x[0] shl 8; x[0]:= x[0]+key1[1]; x[0]:= x[0] shl 8;
  x[0]:= x[0]+key1[2]; x[0]:= x[0] shl 8; x[0]:= x[0]+key1[3];
  x[1]:= key1[4]; x[1]:= x[1] shl 8; x[1]:= x[1]+key1[5]; x[1]:= x[1] shl 8;
  x[1]:= x[1]+key1[6]; x[1]:= x[1] shl 8; x[1]:= x[1]+key1[7];
```

```

x[2]:= key1[8]; x[2]:= x[2] shl 8; x[2]:= x[2]+key1[9]; x[2]:= x[2] shl 8;
x[2]:= x[2]+key1[10]; x[2]:= x[2] shl 8; x[2]:= x[2]+key1[11];
x[3]:= key1[8]; x[3]:= x[3] shl 8; x[3]:= x[3]+key1[13]; x[3]:= x[3] shl 8;
x[3]:= x[3]+key1[10]; x[3]:= x[3] shl 8; x[3]:= x[3]+key1[15];
x[0]:= (x[0] shr 24) or ((x[0] shr 8) and $FF00) or ((x[0] shl 8) and $FF0000) or
(x[0] shl 24);
x[1]:= (x[1] shr 24) or ((x[1] shr 8) and $FF00) or ((x[1] shl 8) and $FF0000) or
(x[1] shl 24);
x[2]:= (x[2] shr 24) or ((x[2] shr 8) and $FF00) or ((x[2] shl 8) and $FF0000) or
(x[2] shl 24);
x[3]:= (x[3] shr 24) or ((x[3] shr 8) and $FF00) or ((x[3] shl 8) and $FF0000) or
(x[3] shl 24);
i:= 0;
while i< 32 do
begin
case (i and 4) of
0:
begin
z[0]:= x[0] xor cast_sbox5[(x[3] shr 16) and $FF] xor
cast_sbox6[x[3] and $FF] xor cast_sbox7[x[3] shr 24] xor
cast_sbox8[(x[3] shr 8) and $FF] xor cast_sbox7[x[2] shr 24];
t[0]:= z[0];
z[1]:= x[2] xor cast_sbox5[z[0] shr 24] xor
cast_sbox6[(z[0] shr 8) and $FF] xor cast_sbox7[(z[0] shr 16) and $FF]
xor
cast_sbox8[z[0] and $FF] xor cast_sbox8[(x[2] shr 8) and $FF];
t[1]:= z[1];
z[2]:= x[3] xor cast_sbox5[z[1] and $FF] xor
cast_sbox6[(z[1] shr 8) and $FF] xor cast_sbox7[(z[1] shr 16) and $FF]
xor
cast_sbox8[z[1] shr 24] xor cast_sbox5[(x[2] shr 16) and $FF];
t[2]:= z[2];
z[3]:= x[1] xor cast_sbox5[(z[2] shr 8) and $FF] xor
cast_sbox6[(z[2] shr 16) and $FF] xor cast_sbox7[z[2] and $FF] xor
cast_sbox8[z[2] shr 24] xor cast_sbox6[x[2] and $FF];
t[3]:= z[3];
end;
4:
begin
x[0]:= z[2] xor cast_sbox5[(z[1] shr 16) and $FF] xor
cast_sbox6[z[1] and $FF] xor cast_sbox7[z[1] shr 24] xor
cast_sbox8[(z[1] shr 8) and $FF] xor cast_sbox7[z[0] shr 24];
t[0]:= x[0];
x[1]:= z[0] xor cast_sbox5[x[0] shr 24] xor

```

```

        cast_sbox6[(x[0] shr 8) and $FF] xor cast_sbox7[(x[0] shr 16) and $FF]
xor
        cast_sbox8[x[0] and $FF] xor cast_sbox8[(z[0] shr 8) and $FF];
        t[1]:= x[1];
        x[2]:= z[1] xor cast_sbox5[x[1] and $FF] xor
        cast_sbox6[(x[1] shr 8) and $FF] xor cast_sbox7[(x[1] shr 16) and $FF]
xor
        cast_sbox8[x[1] shr 24] xor cast_sbox5[(z[0] shr 16) and $FF];
        t[2]:= x[2];
        x[3]:= z[3] xor cast_sbox5[(x[2] shr 8) and $FF] xor
        cast_sbox6[(x[2] shr 16) and $FF] xor cast_sbox7[x[2] and $FF] xor
        cast_sbox8[x[2] shr 24] xor cast_sbox6[z[0] and $FF];
        t[3]:= x[3];
    end;
end;
case (i and 12) of
    0,12:
        begin
            xKey[i+0]:= cast_sbox5[t[2] shr 24] xor cast_sbox6[(t[2] shr 16) and $FF]
xor
            cast_sbox7[t[1] and $FF] xor cast_sbox8[(t[1] shr 8) and $FF];
            xKey[i+1]:= cast_sbox5[(t[2] shr 8) and $FF] xor cast_sbox6[t[2] and
$FF] xor
            cast_sbox7[(t[1] shr 16) and $FF] xor cast_sbox8[t[1] shr 24];
            xKey[i+2]:= cast_sbox5[t[3] shr 24] xor cast_sbox6[(t[3] shr 16) and $FF]
xor
            cast_sbox7[t[0] and $FF] xor cast_sbox8[(t[0] shr 8) and $FF];
            xKey[i+3]:= cast_sbox5[(t[3] shr 8) and $FF] xor cast_sbox6[t[3] and
$FF] xor
            cast_sbox7[(t[0] shr 16) and $FF] xor cast_sbox8[t[0] shr 24];
        end;
    4,8:
        begin
            xKey[i+0]:= cast_sbox5[t[0] and $FF] xor cast_sbox6[(t[0] shr 8) and
$FF] xor
            cast_sbox7[t[3] shr 24] xor cast_sbox8[(t[3] shr 16) and $FF];
            xKey[i+1]:= cast_sbox5[(t[0] shr 16) and $FF] xor cast_sbox6[t[0] shr 24]
xor
            cast_sbox7[(t[3] shr 8) and $FF] xor cast_sbox8[t[3] and $FF];
            xKey[i+2]:= cast_sbox5[t[1] and $FF] xor cast_sbox6[(t[1] shr 8) and
$FF] xor
            cast_sbox7[t[2] shr 24] xor cast_sbox8[(t[2] shr 16) and $FF];
            xKey[i+3]:= cast_sbox5[(t[1] shr 16) and $FF] xor cast_sbox6[t[1] shr 24]
xor

```

```

        cast_sbox7[(t[2] shr 8) and $FF] xor cast_sbox8[t[2] and $FF];
    end;
end;
case (i and 12) of
0:
    begin
        xKey[i+0]:= xKey[i+0] xor cast_sbox5[(z[0] shr 8) and $FF];
        xKey[i+1]:= xKey[i+1] xor cast_sbox6[(z[1] shr 8) and $FF];
        xKey[i+2]:= xKey[i+2] xor cast_sbox7[(z[2] shr 16) and $FF];
        xKey[i+3]:= xKey[i+3] xor cast_sbox8[z[3] shr 24];
    end;
4:
    begin
        xKey[i+0]:= xKey[i+0] xor cast_sbox5[x[2] shr 24];
        xKey[i+1]:= xKey[i+1] xor cast_sbox6[(x[3] shr 16) and $FF];
        xKey[i+2]:= xKey[i+2] xor cast_sbox7[x[0] and $FF];
        xKey[i+3]:= xKey[i+3] xor cast_sbox8[x[1] and $FF];
    end;
8:
    begin
        xKey[i+0]:= xKey[i+0] xor cast_sbox5[(z[2] shr 16) and $FF];
        xKey[i+1]:= xKey[i+1] xor cast_sbox6[z[3] shr 24];
        xKey[i+2]:= xKey[i+2] xor cast_sbox7[(z[0] shr 8) and $FF];
        xKey[i+3]:= xKey[i+3] xor cast_sbox8[(z[1] shr 8) and $FF];
    end;
12:
    begin
        xKey[i+0]:= xKey[i+0] xor cast_sbox5[x[0] and $FF];
        xKey[i+1]:= xKey[i+1] xor cast_sbox6[x[1] and $FF];
        xKey[i+2]:= xKey[i+2] xor cast_sbox7[x[2] shr 24];
        xKey[i+3]:= xKey[i+3] xor cast_sbox8[(x[3] shr 16) and $FF];
    end;
end;
if (i >= 16) then
    begin
        xKey[i+0]:= xKey[i+0] and 31;
        xKey[i+1]:= xKey[i+1] and 31;
        xKey[i+2]:= xKey[i+2] and 31;
        xKey[i+3]:= xKey[i+3] and 31;
    end;
    Inc(i,4);
end;

```

{Pengolahan plaintext}

```

plainlen:=length(data);
if (plainlen mod 8) <> 0 then
  begin
    add:=8-(plainlen mod 8);
    for j:=1 to add do
      data:=data+'0';
    end;
  plainlen:=length(data);

  {enkripsi}
  for k:=1 to (plainlen div 8) do
    begin
      l:= StrtoDWord(data[(((k-1)*8)+1)]+data[(((k-1)*8)+2)]+
        data[(((k-1)*8)+3)]+data[(((k-1)*8)+4)]);
      r:= StrtoDWord(data[(((k-1)*8)+5)]+data[(((k-1)*8)+6)]+
        data[(((k-1)*8)+7)]+data[(((k-1)*8)+8)]);
      l:= (l shr 24) or ((l shr 8) and $FF00) or ((l shl 8) and $FF0000) or (l shl 24);
      r:= (r shr 24) or ((r shr 8) and $FF00) or ((r shl 8) and $FF0000) or (r shl 24);
      u:= LRot32(xKey[0]+r, xKey[0+16]);
      l:= l xor (((cast_sbox1[u shr 24] xor cast_sbox2[(u shr 16) and $FF]) -
        cast_sbox3[(u shr 8) and $FF]) + cast_sbox4[u and $FF]);
      u:= LRot32(xKey[1] xor l, xKey[1+16]);
      r:= r xor (((cast_sbox1[u shr 24] - cast_sbox2[(u shr 16) and $FF]) +
        cast_sbox3[(u shr 8) and $FF]) xor cast_sbox4[u and $FF]);
      u:= LRot32(xKey[2]-r, xKey[2+16]);
      l:= l xor (((cast_sbox1[u shr 24] + cast_sbox2[(u shr 16) and $FF]) xor
        cast_sbox3[(u shr 8) and $FF]) - cast_sbox4[u and $FF]);
      u:= LRot32(xKey[3]+l, xKey[3+16]);
      r:= r xor (((cast_sbox1[u shr 24] xor cast_sbox2[(u shr 16) and $FF]) -
        cast_sbox3[(u shr 8) and $FF]) + cast_sbox4[u and $FF]);
      u:= LRot32(xKey[4] xor r, xKey[4+16]);
      l:= l xor (((cast_sbox1[u shr 24] - cast_sbox2[(u shr 16) and $FF]) +
        cast_sbox3[(u shr 8) and $FF]) xor cast_sbox4[u and $FF]);
      u:= LRot32(xKey[5]-l, xKey[5+16]);
      r:= r xor (((cast_sbox1[u shr 24] + cast_sbox2[(u shr 16) and $FF]) xor
        cast_sbox3[(u shr 8) and $FF]) - cast_sbox4[u and $FF]);
      u:= LRot32(xKey[6]+r, xKey[6+16]);
      l:= l xor (((cast_sbox1[u shr 24] xor cast_sbox2[(u shr 16) and $FF]) -
        cast_sbox3[(u shr 8) and $FF]) + cast_sbox4[u and $FF]);
      u:= LRot32(xKey[7] xor l, xKey[7+16]);
      r:= r xor (((cast_sbox1[u shr 24] - cast_sbox2[(u shr 16) and $FF]) +
        cast_sbox3[(u shr 8) and $FF]) xor cast_sbox4[u and $FF]);
      u:= LRot32(xKey[8]-r, xKey[8+16]);
      l:= l xor (((cast_sbox1[u shr 24] + cast_sbox2[(u shr 16) and $FF]) xor

```

```

    cast_sbox3[(u shr 8) and $FF]) - cast_sbox4[u and $FF]);
u:= LRot32(xKey[9]+l, xKey[9+16]);
r:= r xor (((cast_sbox1[u shr 24] xor cast_sbox2[(u shr 16) and $FF]) -
    cast_sbox3[(u shr 8) and $FF]) + cast_sbox4[u and $FF]);
u:= LRot32(xKey[10] xor r, xKey[10+16]);
l:= l xor (((cast_sbox1[u shr 24] - cast_sbox2[(u shr 16) and $FF]) +
    cast_sbox3[(u shr 8) and $FF]) xor cast_sbox4[u and $FF]);
u:= LRot32(xKey[11]-l, xKey[11+16]);
r:= r xor (((cast_sbox1[u shr 24] + cast_sbox2[(u shr 16) and $FF]) xor
    cast_sbox3[(u shr 8) and $FF]) - cast_sbox4[u and $FF]);
if Rounds> 12 then
begin
    u:= LRot32(xKey[12]+r, xKey[12+16]);
    l:= l xor (((cast_sbox1[u shr 24] xor cast_sbox2[(u shr 16) and $FF]) -
        cast_sbox3[(u shr 8) and $FF]) + cast_sbox4[u and $FF]);
    u:= LRot32(xKey[13] xor l, xKey[13+16]);
    r:= r xor (((cast_sbox1[u shr 24] - cast_sbox2[(u shr 16) and $FF]) +
        cast_sbox3[(u shr 8) and $FF]) xor cast_sbox4[u and $FF]);
    u:= LRot32(xKey[14]-r, xKey[14+16]);
    l:= l xor (((cast_sbox1[u shr 24] + cast_sbox2[(u shr 16) and $FF]) xor
        cast_sbox3[(u shr 8) and $FF]) - cast_sbox4[u and $FF]);
    u:= LRot32(xKey[15]+l, xKey[15+16]);
    r:= r xor (((cast_sbox1[u shr 24] xor cast_sbox2[(u shr 16) and $FF]) -
        cast_sbox3[(u shr 8) and $FF]) + cast_sbox4[u and $FF]);
end;
l:= (l shr 24) or ((l shr 8) and $FF00) or ((l shl 8) and $FF0000) or (l shl 24);
r:= (r shr 24) or ((r shr 8) and $FF00) or ((r shl 8) and $FF0000) or (r shl 24);
valid:=valid xor (r xor l);
ciphertext := ciphertext + DwordToStr(r) + DWordToStr(l);
end;

```

```

{pengolahan ciphertext}
cipherlen:=length(ciphertext);
j:=1;
while j<cipherlen+1 do
begin
    if ciphertext[j] = chr(0) then
    begin
        chip:=copy(ciphertext,1,j-1)+'nilstr'+copy(ciphertext,j+1,cipherlen-j);
        ciphertext:=chip;
    end
    else
    if ciphertext[j] = chr(10) then
    begin

```

```

        chip:=copy(ciphertext,1,j-1)+'notret'+copy(ciphertext,j+1,cipherlen-j);
        ciphertext:=chip;
    end
else
if ciphertext[j] = chr(13) then
begin
    chip:=copy(ciphertext,1,j-1)+'bknret'+copy(ciphertext,j+1,cipherlen-j);
    ciphertext:=chip;
end;
cipherlen:=length(ciphertext);
inc(j);
end;
ciphertext:=ciphertext+IntToStr(add)+DWordToStr(valid);
Encrypt:=ciphertext;
end;

function Decrypt(data: string; key: string) : string;
var
    x, t, z: array[0..3] of DWord;
    key1: array[0..15] of byte;
    i,j,k,p,Len,cipherlen,add,Rounds: integer;
    xKey: array[0..31] of DWord;
    u, l, r, valid, validt: DWord;
    OutData, InData: array[0..7] of byte;
    Plain,Ciphertext,chip,tmp,s: string;
    adds: array[0..9] of string;
label N;

begin
    {pengolahan kunci}
    Len:=length(key);
    if (Len<= 0) or (Len> 16) then
        raise Exception.Create('Panjang Kunci antara 1 sampai 16 huruf');
    if Len<= 10 then
        Rounds:= 12
    else
        Rounds:= 16;
    FillChar(key1,Sizeof(key1),0);
    for j:=0 to len do
        begin
            key1[j]:=ord(key[j+1]);
        end;
    x[0]:= key1[0]; x[0]:= x[0] shl 8; x[0]:= x[0]+key1[1]; x[0]:= x[0] shl 8;
    x[0]:= x[0]+key1[2]; x[0]:= x[0] shl 8; x[0]:= x[0]+key1[3];

```

```

x[1]:= key1[4]; x[1]:= x[1] shl 8; x[1]:= x[1]+key1[5]; x[1]:= x[1] shl 8;
x[1]:= x[1]+key1[6]; x[1]:= x[1] shl 8; x[1]:= x[1]+key1[7];
x[2]:= key1[8]; x[2]:= x[2] shl 8; x[2]:= x[2]+key1[9]; x[2]:= x[2] shl 8;
x[2]:= x[2]+key1[10]; x[2]:= x[2] shl 8; x[2]:= x[2]+key1[11];
x[3]:= key1[8]; x[3]:= x[3] shl 8; x[3]:= x[3]+key1[13]; x[3]:= x[3] shl 8;
x[3]:= x[3]+key1[10]; x[3]:= x[3] shl 8; x[3]:= x[3]+key1[15];
x[0]:= (x[0] shr 24) or ((x[0] shr 8) and $FF00) or ((x[0] shl 8) and $FF0000) or
(x[0] shl 24);
x[1]:= (x[1] shr 24) or ((x[1] shr 8) and $FF00) or ((x[1] shl 8) and $FF0000) or
(x[1] shl 24);
x[2]:= (x[2] shr 24) or ((x[2] shr 8) and $FF00) or ((x[2] shl 8) and $FF0000) or
(x[2] shl 24);
x[3]:= (x[3] shr 24) or ((x[3] shr 8) and $FF00) or ((x[3] shl 8) and $FF0000) or
(x[3] shl 24);
i:= 0;
while i< 32 do
begin
  case (i and 4) of
    0:
      begin
        z[0]:= x[0] xor cast_sbox5[(x[3] shr 16) and $FF] xor
        cast_sbox6[x[3] and $FF] xor cast_sbox7[x[3] shr 24] xor
        cast_sbox8[(x[3] shr 8) and $FF] xor cast_sbox7[x[2] shr 24];
        t[0]:= z[0];
        z[1]:= x[2] xor cast_sbox5[z[0] shr 24] xor
        cast_sbox6[(z[0] shr 8) and $FF] xor cast_sbox7[(z[0] shr 16) and $FF]
xor
        cast_sbox8[z[0] and $FF] xor cast_sbox8[(x[2] shr 8) and $FF];
        t[1]:= z[1];
        z[2]:= x[3] xor cast_sbox5[z[1] and $FF] xor
        cast_sbox6[(z[1] shr 8) and $FF] xor cast_sbox7[(z[1] shr 16) and $FF]
xor
        cast_sbox8[z[1] shr 24] xor cast_sbox5[(x[2] shr 16) and $FF];
        t[2]:= z[2];
        z[3]:= x[1] xor cast_sbox5[(z[2] shr 8) and $FF] xor
        cast_sbox6[(z[2] shr 16) and $FF] xor cast_sbox7[z[2] and $FF] xor
        cast_sbox8[z[2] shr 24] xor cast_sbox6[x[2] and $FF];
        t[3]:= z[3];
      end;
    4:
      begin
        x[0]:= z[2] xor cast_sbox5[(z[1] shr 16) and $FF] xor
        cast_sbox6[z[1] and $FF] xor cast_sbox7[z[1] shr 24] xor
        cast_sbox8[(z[1] shr 8) and $FF] xor cast_sbox7[z[0] shr 24];

```



```

    t[0]:= x[0];
    x[1]:= z[0] xor cast_sbox5[x[0] shr 24] xor
    cast_sbox6[(x[0] shr 8) and $FF] xor cast_sbox7[(x[0] shr 16) and $FF]
xor
    cast_sbox8[x[0] and $FF] xor cast_sbox8[(z[0] shr 8) and $FF];
    t[1]:= x[1];
    x[2]:= z[1] xor cast_sbox5[x[1] and $FF] xor
    cast_sbox6[(x[1] shr 8) and $FF] xor cast_sbox7[(x[1] shr 16) and $FF]
xor
    cast_sbox8[x[1] shr 24] xor cast_sbox5[(z[0] shr 16) and $FF];
    t[2]:= x[2];
    x[3]:= z[3] xor cast_sbox5[(x[2] shr 8) and $FF] xor
    cast_sbox6[(x[2] shr 16) and $FF] xor cast_sbox7[x[2] and $FF] xor
    cast_sbox8[x[2] shr 24] xor cast_sbox6[z[0] and $FF];
    t[3]:= x[3];
end;
end;
case (i and 12) of
0,12:
begin
    xKey[i+0]:= cast_sbox5[t[2] shr 24] xor cast_sbox6[(t[2] shr 16) and $FF]
xor
    cast_sbox7[t[1] and $FF] xor cast_sbox8[(t[1] shr 8) and $FF];
    xKey[i+1]:= cast_sbox5[(t[2] shr 8) and $FF] xor cast_sbox6[t[2] and
$FF] xor
    cast_sbox7[(t[1] shr 16) and $FF] xor cast_sbox8[t[1] shr 24];
    xKey[i+2]:= cast_sbox5[t[3] shr 24] xor cast_sbox6[(t[3] shr 16) and $FF]
xor
    cast_sbox7[t[0] and $FF] xor cast_sbox8[(t[0] shr 8) and $FF];
    xKey[i+3]:= cast_sbox5[(t[3] shr 8) and $FF] xor cast_sbox6[t[3] and
$FF] xor
    cast_sbox7[(t[0] shr 16) and $FF] xor cast_sbox8[t[0] shr 24];
end;
4,8:
begin
    xKey[i+0]:= cast_sbox5[t[0] and $FF] xor cast_sbox6[(t[0] shr 8) and
$FF] xor
    cast_sbox7[t[3] shr 24] xor cast_sbox8[(t[3] shr 16) and $FF];
    xKey[i+1]:= cast_sbox5[(t[0] shr 16) and $FF] xor cast_sbox6[t[0] shr 24]
xor
    cast_sbox7[(t[3] shr 8) and $FF] xor cast_sbox8[t[3] and $FF];
    xKey[i+2]:= cast_sbox5[t[1] and $FF] xor cast_sbox6[(t[1] shr 8) and
$FF] xor
    cast_sbox7[t[2] shr 24] xor cast_sbox8[(t[2] shr 16) and $FF];

```

```

        xKey[i+3]:= cast_sbox5[(t[1] shr 16) and $FF] xor cast_sbox6[t[1] shr 24]
xor
        cast_sbox7[(t[2] shr 8) and $FF] xor cast_sbox8[t[2] and $FF];
    end;
end;
case (i and 12) of
0:
    begin
        xKey[i+0]:= xKey[i+0] xor cast_sbox5[(z[0] shr 8) and $FF];
        xKey[i+1]:= xKey[i+1] xor cast_sbox6[(z[1] shr 8) and $FF];
        xKey[i+2]:= xKey[i+2] xor cast_sbox7[(z[2] shr 16) and $FF];
        xKey[i+3]:= xKey[i+3] xor cast_sbox8[z[3] shr 24];
    end;
4:
    begin
        xKey[i+0]:= xKey[i+0] xor cast_sbox5[x[2] shr 24];
        xKey[i+1]:= xKey[i+1] xor cast_sbox6[(x[3] shr 16) and $FF];
        xKey[i+2]:= xKey[i+2] xor cast_sbox7[x[0] and $FF];
        xKey[i+3]:= xKey[i+3] xor cast_sbox8[x[1] and $FF];
    end;
8:
    begin
        xKey[i+0]:= xKey[i+0] xor cast_sbox5[(z[2] shr 16) and $FF];
        xKey[i+1]:= xKey[i+1] xor cast_sbox6[z[3] shr 24];
        xKey[i+2]:= xKey[i+2] xor cast_sbox7[(z[0] shr 8) and $FF];
        xKey[i+3]:= xKey[i+3] xor cast_sbox8[(z[1] shr 8) and $FF];
    end;
12:
    begin
        xKey[i+0]:= xKey[i+0] xor cast_sbox5[x[0] and $FF];
        xKey[i+1]:= xKey[i+1] xor cast_sbox6[x[1] and $FF];
        xKey[i+2]:= xKey[i+2] xor cast_sbox7[x[2] shr 24];
        xKey[i+3]:= xKey[i+3] xor cast_sbox8[(x[3] shr 16) and $FF];
    end;
end;
if (i >= 16) then
begin
    xKey[i+0]:= xKey[i+0] and 31;
    xKey[i+1]:= xKey[i+1] and 31;
    xKey[i+2]:= xKey[i+2] and 31;
    xKey[i+3]:= xKey[i+3] and 31;
end;
Inc(i,4);
end;

```

```

{pengolahan cipher text}
cipherlen:=length(data);
ciphertext:=data;
j:=1;
while j<cipherlen+1 do
begin
  tmp:=copy(ciphertext,j,6);
  if tmp = 'nilstr' then
  begin
    chip:=copy(ciphertext,1,j-1)+chr(0)+copy(ciphertext,j+6,cipherlen);
    ciphertext:=chip;
  end
  else
  if tmp = 'notret' then
  begin
    chip:=copy(ciphertext,1,j-1)+chr(10)+copy(ciphertext,j+6,cipherlen);
    ciphertext:=chip;
  end
  else
  if tmp = 'bknret' then
  begin
    chip:=copy(ciphertext,1,j-1)+chr(13)+copy(ciphertext,j+6,cipherlen);
    ciphertext:=chip;
  end;
  cipherlen:=length(ciphertext);
  inc(j);
end;

{validasi}
valid:=StrToDWord(RightStr(Ciphertext,4));
for p:=0 to 9 do
  adds[p]:=chr(p+48);
s:=(LeftStr(RightStr(Ciphertext,5),1));
for p:=0 to 9 do
  begin
    if s = adds[p] then goto N;
  end;
Raise Exception.Create('Data Tidak Valid');
Exit;
N:
try
  add:=StrToInt(LeftStr(RightStr(Ciphertext,5),1));
except

```

```

    Raise Exception.Create('Data Tidak Valid');
end;
data:=LeftStr(ciphertext,length(ciphertext)-5);
cipherlen:=length(data);
validt:=0;
for k:=1 to (cipherlen div 8) do
begin
    validt:=validt xor (StrtoDWord(data[(((k-1)*8)+1)]+data[(((k-1)*8)+2)]+
        data[(((k-1)*8)+3)]+data[(((k-1)*8)+4)]) xor StrtoDWord(data[(((k-
1)*8)+5)]+data[(((k-1)*8)+6)]+
        data[(((k-1)*8)+7)]+data[(((k-1)*8)+8)]));
end;
if validt <> valid then
begin
    raise Exception.Create('Data Tidak Valid');
    exit;
end;

{dekripsi}
for k:=1 to (cipherlen div 8) do
begin
    r:= StrtoDWord(data[(((k-1)*8)+1)]+data[(((k-1)*8)+2)]+
        data[(((k-1)*8)+3)]+data[(((k-1)*8)+4)]);
    l:= StrtoDWord(data[(((k-1)*8)+5)]+data[(((k-1)*8)+6)]+
        data[(((k-1)*8)+7)]+data[(((k-1)*8)+8)]);
    l:= (l shr 24) or ((l shr 8) and $FF00) or ((l shl 8) and $FF0000) or (l shl 24);
    r:= (r shr 24) or ((r shr 8) and $FF00) or ((r shl 8) and $FF0000) or (r shl 24);
    if Rounds> 12 then
    begin
        u:= LRot32(xKey[15]+l, xKey[15+16]);
        r:= r xor (((cast_sbox1[u shr 24] xor cast_sbox2[(u shr 16) and $FF]) -
            cast_sbox3[(u shr 8) and $FF]) + cast_sbox4[u and $FF]);
        u:= LRot32(xKey[14]-r, xKey[14+16]);
        l:= l xor (((cast_sbox1[u shr 24] + cast_sbox2[(u shr 16) and $FF]) xor
            cast_sbox3[(u shr 8) and $FF]) - cast_sbox4[u and $FF]);
        u:= LRot32(xKey[13] xor l, xKey[13+16]);
        r:= r xor (((cast_sbox1[u shr 24] - cast_sbox2[(u shr 16) and $FF]) +
            cast_sbox3[(u shr 8) and $FF]) xor cast_sbox4[u and $FF]);
        u:= LRot32(xKey[12]+r, xKey[12+16]);
        l:= l xor (((cast_sbox1[u shr 24] xor cast_sbox2[(u shr 16) and $FF]) -
            cast_sbox3[(u shr 8) and $FF]) + cast_sbox4[u and $FF]);
    end;
    u:= LRot32(xKey[11]-l, xKey[11+16]);
    r:= r xor (((cast_sbox1[u shr 24] + cast_sbox2[(u shr 16) and $FF]) xor

```

```

        cast_sbox3[(u shr 8) and $FF]) - cast_sbox4[u and $FF]);
u:= LRot32(xKey[10] xor r, xKey[10+16]);
l:= l xor (((cast_sbox1[u shr 24] - cast_sbox2[(u shr 16) and $FF]) +
        cast_sbox3[(u shr 8) and $FF]) xor cast_sbox4[u and $FF]);
u:= LRot32(xKey[9]+l, xKey[9+16]);
r:= r xor (((cast_sbox1[u shr 24] xor cast_sbox2[(u shr 16) and $FF]) -
        cast_sbox3[(u shr 8) and $FF]) + cast_sbox4[u and $FF]);
u:= LRot32(xKey[8]-r, xKey[8+16]);
l:= l xor (((cast_sbox1[u shr 24] + cast_sbox2[(u shr 16) and $FF]) xor
        cast_sbox3[(u shr 8) and $FF]) - cast_sbox4[u and $FF]);
u:= LRot32(xKey[7] xor l, xKey[7+16]);
r:= r xor (((cast_sbox1[u shr 24] - cast_sbox2[(u shr 16) and $FF]) +
        cast_sbox3[(u shr 8) and $FF]) xor cast_sbox4[u and $FF]);
u:= LRot32(xKey[6]+r, xKey[6+16]);
l:= l xor (((cast_sbox1[u shr 24] xor cast_sbox2[(u shr 16) and $FF]) -
        cast_sbox3[(u shr 8) and $FF]) + cast_sbox4[u and $FF]);
u:= LRot32(xKey[5]-l, xKey[5+16]);
r:= r xor (((cast_sbox1[u shr 24] + cast_sbox2[(u shr 16) and $FF]) xor
        cast_sbox3[(u shr 8) and $FF]) - cast_sbox4[u and $FF]);
u:= LRot32(xKey[4] xor r, xKey[4+16]);
l:= l xor (((cast_sbox1[u shr 24] - cast_sbox2[(u shr 16) and $FF]) +
        cast_sbox3[(u shr 8) and $FF]) xor cast_sbox4[u and $FF]);
u:= LRot32(xKey[3]+l, xKey[3+16]);
r:= r xor (((cast_sbox1[u shr 24] xor cast_sbox2[(u shr 16) and $FF]) -
        cast_sbox3[(u shr 8) and $FF]) + cast_sbox4[u and $FF]);
u:= LRot32(xKey[2]-r, xKey[2+16]);
l:= l xor (((cast_sbox1[u shr 24] + cast_sbox2[(u shr 16) and $FF]) xor
        cast_sbox3[(u shr 8) and $FF]) - cast_sbox4[u and $FF]);
u:= LRot32(xKey[1] xor l, xKey[1+16]);
r:= r xor (((cast_sbox1[u shr 24] - cast_sbox2[(u shr 16) and $FF]) +
        cast_sbox3[(u shr 8) and $FF]) xor cast_sbox4[u and $FF]);
u:= LRot32(xKey[0]+r, xKey[0+16]);
l:= l xor (((cast_sbox1[u shr 24] xor cast_sbox2[(u shr 16) and $FF]) -
        cast_sbox3[(u shr 8) and $FF]) + cast_sbox4[u and $FF]);
l:= (l shr 24) or ((l shr 8) and $FF00) or ((l shl 8) and $FF0000) or (l shl 24);
r:= (r shr 24) or ((r shr 8) and $FF00) or ((r shl 8) and $FF0000) or (r shl 24);

plain:=plain+DWordToStr(l)+DWordToStr(r);
end;
Decrypt:=LeftStr(plain,length(plain)-add);
end;
end.

```

Listing Tools.pas

```
unit Tools;

interface
uses
  Sysutils;

type
{$IFDEF VER120}
  dword= longword;
{$ELSE}
  dword= longint;
{$ENDIF}

function LRot16(X: word; c: integer): word; assembler;
function RRot16(X: word; c: integer): word; assembler;
function LRot32(X: dword; c: integer): dword; assembler;
function RRot32(X: dword; c: integer): dword; assembler;
procedure XorBlock(I1, I2, O1: PByteArray; Len: integer);
procedure IncBlock(P: PByteArray; Len: integer);
function StrToDWord(s : string) : DWord;
function DWordToStr(wo : DWord) : string;

implementation

function LRot16(X: word; c: integer): word; assembler;
asm
  mov ecx,&c
  mov ax,&X
  rol ax,cl
  mov &Result,ax
end;

function RRot16(X: word; c: integer): word; assembler;
asm
  mov ecx,&c
  mov ax,&X
  ror ax,cl
  mov &Result,ax
end;

function LRot32(X: dword; c: integer): dword; register; assembler;
asm
```

```

    mov ecx, edx
    rol eax, cl
end;

```

```

function RRot32(X: dword; c: integer): dword; register; assembler;
asm
    mov ecx, edx
    ror eax, cl
end;

```

```

procedure XorBlock(I1, I2, O1: PByteArray; Len: integer);
var
    i: integer;
begin
    for i:= 0 to Len-1 do
        O1[i]:= I1[i] xor I2[i];
    end;
end;

```

```

procedure IncBlock(P: PByteArray; Len: integer);
begin
    Inc(P[Len-1]);
    if (P[Len-1]= 0) and (Len> 1) then
        IncBlock(P,Len-1);
    end;
end;

```

```

function StrToDWord(s : string) : DWord;
var cely : Dword;
    Si1,Si2,Si3,Si4 : byte;
begin
    Si1:=ord(s[1]);
    Si2:=ord(s[2]);
    Si3:=ord(s[3]);
    Si4:=ord(s[4]);
    cely:=Si1;
    cely:=cely shl 8;
    cely:=cely + Si2;
    cely:=cely shl 8;
    cely:=cely + Si3;
    cely:=cely shl 8;
    cely:=cely + Si4;
    StrToDWord:=cely;
end;

```

```

function DWordToStr(wo : DWord) : string;

```

```

var vysl : string;
    rr,ll : word;
    Si1,Si2,Si3,Si4 : byte;
begin
rr:=(wo shl 16) shr 16;
ll:=(wo shr 16);
Si2:=(ll shl 8) shr 8;
Si1:=(ll shr 8);
Si4:=(rr shl 8) shr 8;
Si3:=(rr shr 8);
vysl:=chr(Si1)+chr(Si2)+chr(Si3)+chr(Si4);
DWordToStr:=vysl;
end;

end.

```


Listing RSATools

```
unit RSATools;

interface
uses SysUtils, StrUtils, IdGlobal, Math;

Function Euclid(nilai1 : Longint; nilai2 : Longint) : Longint ;
Function gcd(p : longint; q : longint) : Longint;
Function RdmPrime() : Longint;
Function pangkatmod(num1 : longint; num2 : longint; num3 : longint) : longint;

implementation

Function Euclid(nilai1 : Longint; nilai2 : Longint) : Longint ;
var mex,bex,A1,A2,A3,Qex,T1,T2,T3,B1,B2,B3,hasil:Longint;
label itung, selesai;

begin
  mex := nilai1;
  bex := nilai2;

  A1 := 1;   B1 := 0;
  A2 := 0;   B2 := 1;
  A3 := mex; B3 := bex;

  itung:
  if B3 = 0 then
    begin
      hasil := 0;
      GoTo selesai;
    end;
  if B3 = 1 then
    begin
      hasil := B2;
      GoTo selesai;
    end;

  Qex := A3 div B3;
  T1 := A1 - Qex * B1;
  T2 := A2 - Qex * B2;
  T3 := A3 - Qex * B3;
```

```

    A1 := B1; B1 := T1;
    A2 := B2; B2 := T2;
    A3 := B3; B3 := T3;

    GoTo itung;
    selesai:
    Euclid := hasil;
end;

Function gcd(p : longint; q : longint) : Longint;
var A11,B11,R11 : Longint;
label balik;

begin
    A11 := p;
    B11 := q;

    balik:
    If B11 = 0 Then
        gcd := A11
    else
        begin
            R11 := A11 Mod B11;
            A11 := B11;
            B11 := R11;
            GoTo balik;
        end;
    end;
end;

Function RdmPrime() : Longint;
var iRandom : integer; // holds random long result
    i2 : longint; // checkprime loop counter
    temp2a : Longint; //swap var
    iLowerBound : Longint;
    iUpperBound : Longint;
label i110;

begin
    iLowerBound := 30;
    iUpperBound := 300;
    Randomize;
    i110:
    iRandom := (Round((iUpperBound - iLowerBound + 1) *
(Random(10000)/10000)) + iLowerBound);

```

```

//trap handler
If (iRandom = 0) Or (iRandom = 1) Then
GoTo i110;

//check number
for i2:= 2 to (iRandom-1) do
begin
temp2a := iRandom Mod i2;
If temp2a = 0 Then
GoTo i110;
end;

RdmPrime := iRandom;
End;

Function pangkatmod(num1 : longint;num2 : longint;num3 : longint) : longint;
var a22,b22,n22,nilaimod,nilaic,nilaid,nilaii,nilaik,naik : longint;
barray : array of Variant;
decbin : Variant;

begin
a22 := num1;
b22 := num2;
n22 := num3;

decbin := inttobin(b22);
nilaic := 0;
nilaid := 1;
nilaik := Length(decbin);
SetLength(barray,nilaik);
naik := 1;

For nilaii := nilaik - 1 downto 0 do
begin
barray[nilaii] := Midstr(decbin, naik, 1);
naik := naik + 1;
nilaic := 2 * nilaic;
nilaid := (nilaid * nilaid) Mod n22;
If barray[nilaii] = 1 Then
begin
nilaic := nilaic + 1;
nilaid := (nilaid * a22) Mod n22;

```

```
        end;  
    end;  
  
    pangkatmod := nilaid;  
End;  
  
end.
```

Lampiran B

Subtitution Box

Substitution Box 1

```
cast_sbox1: array[0..255]of DWord= (  
    $30FB40D4, $9FA0FF0B, $6BECCD2F, $3F258C7A,  
    $1E213F2F, $9C004DD3, $6003E540, $CF9FC949,  
    $BFD4AF27, $88BBBDB5, $E2034090, $98D09675,  
    $6E63A0E0, $15C361D2, $C2E7661D, $22D4FF8E,  
    $28683B6F, $C07FD059, $FF2379C8, $775F50E2,  
    $43C340D3, $DF2F8656, $887CA41A, $A2D2BD2D,  
    $A1C9E0D6, $346C4819, $61B76D87, $22540F2F,  
    $2ABE32E1, $AA54166B, $22568E3A, $A2D341D0,  
    $66DB40C8, $A784392F, $004DFF2F, $2DB9D2DE,  
    $97943FAC, $4A97C1D8, $527644B7, $B5F437A7,  
    $B82CBAEF, $D751D159, $6FF7F0ED, $5A097A1F,  
    $827B68D0, $90ECF52E, $22B0C054, $BC8E5935,  
    $4B6D2F7F, $50BB64A2, $D2664910, $BEE5812D,  
    $B7332290, $E93B159F, $B48EE411, $4BFF345D,  
    $FD45C240, $AD31973F, $C4F6D02E, $55FC8165,  
    $D5B1CAAD, $A1AC2DAE, $A2D4B76D, $C19B0C50,  
    $882240F2, $0C6E4F38, $A4E4BFD7, $4F5BA272,  
    $564C1D2F, $C59C5319, $B949E354, $B04669FE,  
    $B1B6AB8A, $C71358DD, $6385C545, $110F935D,  
    $57538AD5, $6A390493, $E63D37E0, $2A54F6B3,  
    $3A787D5F, $6276A0B5, $19A6FCDF, $7A42206A,  
    $29F9D4D5, $F61B1891, $BB72275E, $AA508167,  
    $38901091, $C6B505EB, $84C7CB8C, $2AD75A0F,  
    $874A1427, $A2D1936B, $2AD286AF, $AA56D291,  
    $D7894360, $425C750D, $93B39E26, $187184C9,  
    $6C00B32D, $73E2BB14, $A0BEBC3C, $54623779,  
    $64459EAB, $3F328B82, $7718CF82, $59A2CEA6,  
    $04EE002E, $89FE78E6, $3FAB0950, $325FF6C2,  
    $81383F05, $6963C5C8, $76CB5AD6, $D49974C9,  
    $CA180DCF, $380782D5, $C7FA5CF6, $8AC31511,  
    $35E79E13, $47DA91D0, $F40F9086, $A7E2419E,  
    $31366241, $051EF495, $AA573B04, $4A805D8D,  
    $548300D0, $00322A3C, $BF64CDDF, $BA57A68E,  
    $75C6372B, $50AFD341, $A7C13275, $915A0BF5,  
    $6B54BFAB, $2B0B1426, $AB4CC9D7, $449CCD82,  
    $F7FBF265, $AB85C5F3, $1B55DB94, $AAD4E324,  
    $CFA4BD3F, $2DEAA3E2, $9E204D02, $C8BD25AC,  
    $EADF55B3, $D5BD9E98, $E31231B2, $2AD5AD6C,  
    $954329DE, $ADBE4528, $D8710F69, $AA51C90F,  
    $AA786BF6, $22513F1E, $AA51A79B, $2AD344CC,  
    $7B5A41F0, $D37CFBAD, $1B069505, $41ECE491,
```

\$B4C332E6, \$032268D4, \$C9600ACC, \$CE387E6D,
\$BF6BB16C, \$6A70FB78, \$0D03D9C9, \$D4DF39DE,
\$E01063DA, \$4736F464, \$5AD328D8, \$B347CC96,
\$75BB0FC3, \$98511BFB, \$4FFBCC35, \$B58BCF6A,
\$E11F0ABC, \$BFC5FE4A, \$A70AEC10, \$AC39570A,
\$3F04442F, \$6188B153, \$E0397A2E, \$5727CB79,
\$9CEB418F, \$1CACD68D, \$2AD37C96, \$0175CB9D,
\$C69DFF09, \$C75B65F0, \$D9DB40D8, \$EC0E7779,
\$4744EAD4, \$B11C3274, \$DD24CB9E, \$7E1C54BD,
\$F01144F9, \$D2240EB1, \$9675B3FD, \$A3AC3755,
\$D47C27AF, \$51C85F4D, \$56907596, \$A5BB15E6,
\$580304F0, \$CA042CF1, \$011A37EA, \$8DBFAADB,
\$35BA3E4A, \$3526FFA0, \$C37B4D09, \$BC306ED9,
\$98A52666, \$5648F725, \$FF5E569D, \$0CED63D0,
\$7C63B2CF, \$700B45E1, \$D5EA50F1, \$85A92872,
\$AF1FBDA7, \$D4234870, \$A7870BF3, \$2D3B4D79,
\$42E04198, \$0CD0EDE7, \$26470DB8, \$F881814C,
\$474D6AD7, \$7C0C5E5C, \$D1231959, \$381B7298,
\$F5D2F4DB, \$AB838653, \$6E2F1E23, \$83719C9E,
\$BD91E046, \$9A56456E, \$DC39200C, \$20C8C571,
\$962BDA1C, \$E1E696FF, \$B141AB08, \$7CCA89B9,
\$1A69E783, \$02CC4843, \$A2F7C579, \$429EF47D,
\$427B169C, \$5AC9F049, \$DD8F0F00, \$5C8165BF
);

Substitution Box 2

```
cast_sbox2: array[0..255] of DWord = (  
    $1F201094, $EF0BA75B, $69E3CF7E, $393F4380,  
    $FE61CF7A, $EEC5207A, $55889C94, $72FC0651,  
    $ADA7EF79, $4E1D7235, $D55A63CE, $DE0436BA,  
    $99C430EF, $5F0C0794, $18DCDB7D, $A1D6EFF3,  
    $A0B52F7B, $59E83605, $EE15B094, $E9FFD909,  
    $DC440086, $EF944459, $BA83CCB3, $E0C3CDFB,  
    $D1DA4181, $3B092AB1, $F997F1C1, $A5E6CF7B,  
    $01420DDB, $E4E7EF5B, $25A1FF41, $E180F806,  
    $1FC41080, $179BEE7A, $D37AC6A9, $FE5830A4,  
    $98DE8B7F, $77E83F4E, $79929269, $24FA9F7B,  
    $E113C85B, $ACC40083, $D7503525, $F7EA615F,  
    $62143154, $0D554B63, $5D681121, $C866C359,  
    $3D63CF73, $CEE234C0, $D4D87E87, $5C672B21,  
    $071F6181, $39F7627F, $361E3084, $E4EB573B,  
    $602F64A4, $D63ACD9C, $1BBC4635, $9E81032D,  
    $2701F50C, $99847AB4, $A0E3DF79, $BA6CF38C,  
    $10843094, $2537A95E, $F46F6FFE, $A1FF3B1F,  
    $208CFB6A, $8F458C74, $D9E0A227, $4EC73A34,  
    $FC884F69, $3E4DE8DF, $EF0E0088, $3559648D,  
    $8A45388C, $1D804366, $721D9BFD, $A58684BB,  
    $E8256333, $844E8212, $128D8098, $FED33FB4,  
    $CE280AE1, $27E19BA5, $D5A6C252, $E49754BD,  
    $C5D655DD, $EB667064, $77840B4D, $A1B6A801,  
    $84DB26A9, $E0B56714, $21F043B7, $E5D05860,  
    $54F03084, $066FF472, $A31AA153, $DADC4755,  
    $B5625DBF, $68561BE6, $83CA6B94, $2D6ED23B,  
    $ECCF01DB, $A6D3D0BA, $B6803D5C, $AF77A709,  
    $33B4A34C, $397BC8D6, $5EE22B95, $5F0E5304,  
    $81ED6F61, $20E74364, $B45E1378, $DE18639B,  
    $881CA122, $B96726D1, $8049A7E8, $22B7DA7B,  
    $5E552D25, $5272D237, $79D2951C, $C60D894C,  
    $488CB402, $1BA4FE5B, $A4B09F6B, $1CA815CF,  
    $A20C3005, $8871DF63, $B9DE2FCB, $0CC6C9E9,  
    $0BEEFF53, $E3214517, $B4542835, $9F63293C,  
    $EE41E729, $6E1D2D7C, $50045286, $1E6685F3,  
    $F33401C6, $30A22C95, $31A70850, $60930F13,  
    $73F98417, $A1269859, $EC645C44, $52C877A9,  
    $CDFF33A6, $A02B1741, $7CBAD9A2, $2180036F,  
    $50D99C08, $CB3F4861, $C26BD765, $64A3F6AB,  
    $80342676, $25A75E7B, $E4E6D1FC, $20C710E6,  
    $CDF0B680, $17844D3B, $31EEF84D, $7E0824E4,
```


\$2CCB49EB, \$846A3BAE, \$8FF77888, \$EE5D60F6,
\$7AF75673, \$2FDD5CDB, \$A11631C1, \$30F66F43,
\$B3FAEC54, \$157FD7FA, \$EF8579CC, \$D152DE58,
\$DB2FFD5E, \$8F32CE19, \$306AF97A, \$02F03EF8,
\$99319AD5, \$C242FA0F, \$A7E3EBB0, \$C68E4906,
\$B8DA230C, \$80823028, \$DCDEF3C8, \$D35FB171,
\$088A1BC8, \$BEC0C560, \$61A3C9E8, \$BCA8F54D,
\$C72FEFFA, \$22822E99, \$82C570B4, \$D8D94E89,
\$8B1C34BC, \$301E16E6, \$273BE979, \$B0FFEAA6,
\$61D9B8C6, \$00B24869, \$B7FFCE3F, \$08DC283B,
\$43DAF65A, \$F7E19798, \$7619B72F, \$8F1C9BA4,
\$DC8637A0, \$16A7D3B1, \$9FC393B7, \$A7136EEB,
\$C6BCC63E, \$1A513742, \$EF6828BC, \$520365D6,
\$2D6A77AB, \$3527ED4B, \$821FD216, \$095C6E2E,
\$DB92F2FB, \$5EEA29CB, \$145892F5, \$91584F7F,
\$5483697B, \$2667A8CC, \$85196048, \$8C4BACEA,
\$833860D4, \$0D23E0F9, \$6C387E8A, \$0AE6D249,
\$B284600C, \$D835731D, \$DCB1C647, \$AC4C56EA,
\$3EBD81B3, \$230EABB0, \$6438BC87, \$F0B5B1FA,
\$8F5EA2B3, \$FC184642, \$0A036B7A, \$4FB089BD,
\$649DA589, \$A345415E, \$5C038323, \$3E5D3BB9,
\$43D79572, \$7E6DD07C, \$06DFDF1E, \$6C6CC4EF,
\$7160A539, \$73BFBE70, \$83877605, \$4523ECF1
);

Substitution Box 3

```
cast_sbox3: array[0..255] of DWord = (  
    $8DEFC240, $25FA5D9F, $EB903DBF, $E810C907,  
    $47607FFF, $369FE44B, $8C1FC644, $AECECA90,  
    $BEB1F9BF, $EEFBCAEA, $E8CF1950, $51DF07AE,  
    $920E8806, $F0AD0548, $E13C8D83, $927010D5,  
    $11107D9F, $07647DB9, $B2E3E4D4, $3D4F285E,  
    $B9AFA820, $FADE82E0, $A067268B, $8272792E,  
    $553FB2C0, $489AE22B, $D4EF9794, $125E3FBC,  
    $21FFFC EE, $825B1BFD, $9255C5ED, $1257A240,  
    $4E1A8302, $BAE07FFF, $528246E7, $8E57140E,  
    $3373F7BF, $8C9F8188, $A6FC4EE8, $C982B5A5,  
    $A8C01DB7, $579FC264, $67094F31, $F2BD3F5F,  
    $40FFF7C1, $1FB78DFC, $8E6BD2C1, $437BE59B,  
    $99B03DBF, $B5DBC64B, $638DC0E6, $55819D99,  
    $A197C81C, $4A012D6E, $C5884A28, $CCC36F71,  
    $B843C213, $6C0743F1, $8309893C, $0FEDDD5F,  
    $2F7FE850, $D7C07F7E, $02507FBF, $5AFB9A04,  
    $A747D2D0, $1651192E, $AF70BF3E, $58C31380,  
    $5F98302E, $727CC3C4, $0A0FB402, $0F7FEF82,  
    $8C96FDAD, $5D2C2AAE, $8EE99A49, $50DA88B8,  
    $8427F4A0, $1EAC5790, $796FB449, $8252DC15,  
    $EFBD7D9B, $A672597D, $ADA840D8, $45F54504,  
    $FA5D7403, $E83EC305, $4F91751A, $925669C2,  
    $23EFE941, $A903F12E, $60270DF2, $0276E4B6,  
    $94FD6574, $927985B2, $8276DBCB, $02778176,  
    $F8AF918D, $4E48F79E, $8F616DDF, $E29D840E,  
    $842F7D83, $340CE5C8, $96BBB682, $93B4B148,  
    $EF303CAB, $984FAF28, $779FAF9B, $92DC560D,  
    $224D1E20, $8437AA88, $7D29DC96, $2756D3DC,  
    $8B907CEE, $B51FD240, $E7C07CE3, $E566B4A1,  
    $C3E9615E, $3CF8209D, $6094D1E3, $CD9CA341,  
    $5C76460E, $00EA983B, $D4D67881, $FD47572C,  
    $F76CEDD9, $BDA8229C, $127DADAA, $438A074E,  
    $1F97C090, $081BDB8A, $93A07EBE, $B938CA15,  
    $97B03CFF, $3DC2C0F8, $8D1AB2EC, $64380E51,  
    $68CC7BFB, $D90F2788, $12490181, $5DE5FFD4,  
    $DD7EF86A, $76A2E214, $B9A40368, $925D958F,  
    $4B39FFFA, $BA39AEE9, $A4FFD30B, $FAF7933B,  
    $6D498623, $193CBCFA, $27627545, $825CF47A,  
    $61BD8BA0, $D11E42D1, $CEAD04F4, $127EA392,  
    $10428DB7, $8272A972, $9270C4A8, $127DE50B,  
    $285BA1C8, $3C62F44F, $35C0EAA5, $E805D231,
```

\$428929FB, \$B4FCDF82, \$4FB66A53, \$0E7DC15B,
\$1F081FAB, \$108618AE, \$FCFD086D, \$F9FF2889,
\$694BCC11, \$236A5CAE, \$12DECA4D, \$2C3F8CC5,
\$D2D02DFE, \$F8EF5896, \$E4CF52DA, \$95155B67,
\$494A488C, \$B9B6A80C, \$5C8F82BC, \$89D36B45,
\$3A609437, \$EC00C9A9, \$44715253, \$0A874B49,
\$D773BC40, \$7C34671C, \$02717EF6, \$4FEB5536,
\$A2D02FFF, \$D2BF60C4, \$D43F03C0, \$50B4EF6D,
\$07478CD1, \$006E1888, \$A2E53F55, \$B9E6D4BC,
\$A2048016, \$97573833, \$D7207D67, \$DE0F8F3D,
\$72F87B33, \$ABCC4F33, \$7688C55D, \$7B00A6B0,
\$947B0001, \$570075D2, \$F9BB88F8, \$8942019E,
\$4264A5FF, \$856302E0, \$72DBD92B, \$EE971B69,
\$6EA22FDE, \$5F08AE2B, \$AF7A616D, \$E5C98767,
\$CF1FEBD2, \$61EFC8C2, \$F1AC2571, \$CC8239C2,
\$67214CB8, \$B1E583D1, \$B7DC3E62, \$7F10BDCE,
\$F90A5C38, \$0FF0443D, \$606E6DC6, \$60543A49,
\$5727C148, \$2BE98A1D, \$8AB41738, \$20E1BE24,
\$AF96DA0F, \$68458425, \$99833BE5, \$600D457D,
\$282F9350, \$8334B362, \$D91D1120, \$2B6D8DA0,
\$642B1E31, \$9C305A00, \$52BCE688, \$1B03588A,
\$F7BAEFD5, \$4142ED9C, \$A4315C11, \$83323EC5,
\$DFEF4636, \$A133C501, \$E9D3531C, \$EE353783
);

Substitution Box 4

```
cast_sbox4: array[0..255] of DWord = (  
    $9DB30420, $1FB6E9DE, $A7BE7BEF, $D273A298,  
    $4A4F7BDB, $64AD8C57, $85510443, $FA020ED1,  
    $7E287AFF, $E60FB663, $095F35A1, $79EBF120,  
    $FD059D43, $6497B7B1, $F3641F63, $241E4ADF,  
    $28147F5F, $4FA2B8CD, $C9430040, $0CC32220,  
    $FDD30B30, $C0A5374F, $1D2D00D9, $24147B15,  
    $EE4D111A, $0FCA5167, $71FF904C, $2D195FFE,  
    $1A05645F, $0C13FEFE, $081B08CA, $05170121,  
    $80530100, $E83E5EFE, $AC9AF4F8, $7FE72701,  
    $D2B8EE5F, $06DF4261, $BB9E9B8A, $7293EA25,  
    $CE84FFDF, $F5718801, $3DD64B04, $A26F263B,  
    $7ED48400, $547EEBE6, $446D4CA0, $6CF3D6F5,  
    $2649ABDF, $AEA0C7F5, $36338CC1, $503F7E93,  
    $D3772061, $11B638E1, $72500E03, $F80EB2BB,  
    $ABE0502E, $EC8D77DE, $57971E81, $E14F6746,  
    $C9335400, $6920318F, $081DBB99, $FFC304A5,  
    $4D351805, $7F3D5CE3, $A6C866C6, $5D5BCCA9,  
    $DAEC6FEA, $9F926F91, $9F46222F, $3991467D,  
    $A5BF6D8E, $1143C44F, $43958302, $D0214EEB,  
    $022083B8, $3FB6180C, $18F8931E, $281658E6,  
    $26486E3E, $8BD78A70, $7477E4C1, $B506E07C,  
    $F32D0A25, $79098B02, $E4EABB81, $28123B23,  
    $69DEAD38, $1574CA16, $DF871B62, $211C40B7,  
    $A51A9EF9, $0014377B, $041E8AC8, $09114003,  
    $BD59E4D2, $E3D156D5, $4FE876D5, $2F91A340,  
    $557BE8DE, $00EAE4A7, $0CE5C2EC, $4DB4BBA6,  
    $E756BDFF, $DD3369AC, $EC17B035, $06572327,  
    $99AFC8B0, $56C8C391, $6B65811C, $5E146119,  
    $6E85CB75, $BE07C002, $C2325577, $893FF4EC,  
    $5BBFC92D, $D0EC3B25, $B7801AB7, $8D6D3B24,  
    $20C763EF, $C366A5FC, $9C382880, $0ACE3205,  
    $AAC9548A, $ECA1D7C7, $041AFA32, $1D16625A,  
    $6701902C, $9B757A54, $31D477F7, $9126B031,  
    $36CC6FDB, $C70B8B46, $D9E66A48, $56E55A79,  
    $026A4CEB, $52437EFF, $2F8F76B4, $0DF980A5,  
    $8674CDE3, $EDDA04EB, $17A9BE04, $2C18F4DF,  
    $B7747F9D, $AB2AF7B4, $EFC34D20, $2E096B7C,  
    $1741A254, $E5B6A035, $213D42F6, $2C1C7C26,  
    $61C2F50F, $6552DAF9, $D2C231F8, $25130F69,  
    $D8167FA2, $0418F2C8, $001A96A6, $0D1526AB,  
    $63315C21, $5E0A72EC, $49BAFEFD, $187908D9,
```

\$8D0DBD86, \$311170A7, \$3E9B640C, \$CC3E10D7,
\$D5CAD3B6, \$0CAEC388, \$F73001E1, \$6C728AFF,
\$71EAE2A1, \$1F9AF36E, \$CFCBD12F, \$C1DE8417,
\$AC07BE6B, \$CB44A1D8, \$8B9B0F56, \$013988C3,
\$B1C52FCA, \$B4BE31CD, \$D8782806, \$12A3A4E2,
\$6F7DE532, \$58FD7EB6, \$D01EE900, \$24ADFFC2,
\$F4990FC5, \$9711AAC5, \$001D7B95, \$82E5E7D2,
\$109873F6, \$00613096, \$C32D9521, \$ADA121FF,
\$29908415, \$7FBB977F, \$AF9EB3DB, \$29C9ED2A,
\$5CE2A465, \$A730F32C, \$D0AA3FE8, \$8A5CC091,
\$D49E2CE7, \$0CE454A9, \$D60ACD86, \$015F1919,
\$77079103, \$DEA03AF6, \$78A8565E, \$DEE356DF,
\$21F05CBE, \$8B75E387, \$B3C50651, \$B8A5C3EF,
\$D8EEB6D2, \$E523BE77, \$C2154529, \$2F69EFDf,
\$AFE67AFB, \$F470C4B2, \$F3E0EB5B, \$D6CC9876,
\$39E4460C, \$1FDA8538, \$1987832F, \$CA007367,
\$A99144F8, \$296B299E, \$492FC295, \$9266BEAB,
\$B5676E69, \$9BD3DDDA, \$DF7E052F, \$DB25701C,
\$1B5E51EE, \$F65324E6, \$6AFCE36C, \$0316CC04,
\$8644213E, \$B7DC59D0, \$7965291F, \$CCD6FD43,
\$41823979, \$932BCDF6, \$B657C34D, \$4EDFD282,
\$7AE5290C, \$3CB9536B, \$851E20FE, \$9833557E,
\$13ECF0B0, \$D3FFB372, \$3F85C5C1, \$0AEF7ED2
);

Substitution Box 5

```
cast_sbox5: array[0..255] of DWord = (  
    $7EC90C04, $2C6E74B9, $9B0E66DF, $A6337911,  
    $B86A7FFF, $1DD358F5, $44DD9D44, $1731167F,  
    $08FBF1FA, $E7F511CC, $D2051B00, $735ABA00,  
    $2AB722D8, $386381CB, $ACF6243A, $69BEFD7A,  
    $E6A2E77F, $F0C720CD, $C4494816, $CCF5C180,  
    $38851640, $15B0A848, $E68B18CB, $4CAADEFF,  
    $5F480A01, $0412B2AA, $259814FC, $41D0EFE2,  
    $4E40B48D, $248EB6FB, $8DBA1CFE, $41A99B02,  
    $1A550A04, $BA8F65CB, $7251F4E7, $95A51725,  
    $C106ECD7, $97A5980A, $C539B9AA, $4D79FE6A,  
    $F2F3F763, $68AF8040, $ED0C9E56, $11B4958B,  
    $E1EB5A88, $8709E6B0, $D7E07156, $4E29FEA7,  
    $6366E52D, $02D1C000, $C4AC8E05, $9377F571,  
    $0C05372A, $578535F2, $2261BE02, $D642A0C9,  
    $DF13A280, $74B55BD2, $682199C0, $D421E5EC,  
    $53FB3CE8, $C8ADEDDB3, $28A87FC9, $3D959981,  
    $5C1FF900, $FE38D399, $0C4EFF0B, $062407EA,  
    $AA2F4FB1, $4FB96976, $90C79505, $B0A8A774,  
    $EF55A1FF, $E59CA2C2, $A6B62D27, $E66A4263,  
    $DF65001F, $0EC50966, $DFDD55BC, $29DE0655,  
    $911E739A, $17AF8975, $32C7911C, $89F89468,  
    $0D01E980, $524755F4, $03B63CC9, $0CC844B2,  
    $BCF3F0AA, $87AC36E9, $E53A7426, $01B3D82B,  
    $1A9E7449, $64EE2D7E, $CDDBB1DA, $01C94910,  
    $B868BF80, $0D26F3FD, $9342EDE7, $04A5C284,  
    $636737B6, $50F5B616, $F24766E3, $8ECA36C1,  
    $136E05DB, $FEF18391, $FB887A37, $D6E7F7D4,  
    $C7FB7DC9, $3063FCDF, $B6F589DE, $EC2941DA,  
    $26E46695, $B7566419, $F654EFC5, $D08D58B7,  
    $48925401, $C1BACB7F, $E5FF550F, $B6083049,  
    $5BB5D0E8, $87D72E5A, $AB6A6EE1, $223A66CE,  
    $C62BF3CD, $9E0885F9, $68CB3E47, $086C010F,  
    $A21DE820, $D18B69DE, $F3F65777, $FA02C3F6,  
    $407EDAC3, $CBB3D550, $1793084D, $B0D70EBA,  
    $0AB378D5, $D951FB0C, $DED7DA56, $4124BBE4,  
    $94CA0B56, $0F5755D1, $E0E1E56E, $6184B5BE,  
    $580A249F, $94F74BC0, $E327888E, $9F7B5561,  
    $C3DC0280, $05687715, $646C6BD7, $44904DB3,  
    $66B4F0A3, $C0F1648A, $697ED5AF, $49E92FF6,  
    $309E374F, $2CB6356A, $85808573, $4991F840,  
    $76F0AE02, $083BE84D, $28421C9A, $44489406,
```

\$736E4CB8, \$C1092910, \$8BC95FC6, \$7D869CF4,
\$134F616F, \$2E77118D, \$B31B2BE1, \$AA90B472,
\$3CA5D717, \$7D161BBA, \$9CAD9010, \$AF462BA2,
\$9FE459D2, \$45D34559, \$D9F2DA13, \$DBC65487,
\$F3E4F94E, \$176D486F, \$097C13EA, \$631DA5C7,
\$445F7382, \$175683F4, \$CDC66A97, \$70BE0288,
\$B3CDCF72, \$6E5DD2F3, \$20936079, \$459B80A5,
\$BE60E2DB, \$A9C23101, \$EBA5315C, \$224E42F2,
\$1C5C1572, \$F6721B2C, \$1AD2FFF3, \$8C25404E,
\$324ED72F, \$4067B7FD, \$0523138E, \$5CA3BC78,
\$DC0FD66E, \$75922283, \$784D6B17, \$58EBB16E,
\$44094F85, \$3F481D87, \$FCFEAE7B, \$77B5FF76,
\$8C2302BF, \$AAF47556, \$5F46B02A, \$2B092801,
\$3D38F5F7, \$0CA81F36, \$52AF4A8A, \$66D5E7C0,
\$DF3B0874, \$95055110, \$1B5AD7A8, \$F61ED5AD,
\$6CF6E479, \$20758184, \$D0CEFA65, \$88F7BE58,
\$4A046826, \$0FF6F8F3, \$A09C7F70, \$5346ABA0,
\$5CE96C28, \$E176EDA3, \$6BAC307F, \$376829D2,
\$85360FA9, \$17E3FE2A, \$24B79767, \$F5A96B20,
\$D6CD2595, \$68FF1EBF, \$7555442C, \$F19F06BE,
\$F9E0659A, \$EEB9491D, \$34010718, \$BB30CAB8,
\$E822FE15, \$88570983, \$750E6249, \$DA627E55,
\$5E76FFA8, \$B1534546, \$6D47DE08, \$EFE9E7D4
);

Substitution Box 6

```
cast_sbox6: array[0..255] of DWord = (  
    $F6FA8F9D, $2CAC6CE1, $4CA34867, $E2337F7C,  
    $95DB08E7, $016843B4, $ECED5CBC, $325553AC,  
    $BF9F0960, $DFA1E2ED, $83F0579D, $63ED86B9,  
    $1AB6A6B8, $DE5EBE39, $F38FF732, $8989B138,  
    $33F14961, $C01937BD, $F506C6DA, $E4625E7E,  
    $A308EA99, $4E23E33C, $79CBD7CC, $48A14367,  
    $A3149619, $FEC94BD5, $A114174A, $EAA01866,  
    $A084DB2D, $09A8486F, $A888614A, $2900AF98,  
    $01665991, $E1992863, $C8F30C60, $2E78EF3C,  
    $D0D51932, $CF0FEC14, $F7CA07D2, $D0A82072,  
    $FD41197E, $9305A6B0, $E86BE3DA, $74BED3CD,  
    $372DA53C, $4C7F4448, $DAB5D440, $6DBA0EC3,  
    $083919A7, $9FBAEED9, $49DBCFCB, $4E670C53,  
    $5C3D9C01, $64BDB941, $2C0E636A, $BA7DD9CD,  
    $EA6F7388, $E70BC762, $35F29ADB, $5C4CDD8D,  
    $F0D48D8C, $B88153E2, $08A19866, $1AE2EAC8,  
    $284CAF89, $AA928223, $9334BE53, $3B3A21BF,  
    $16434BE3, $9AEA3906, $EFE8C36E, $F890CDD9,  
    $80226DAE, $C340A4A3, $DF7E9C09, $A694A807,  
    $5B7C5ECC, $221DB3A6, $9A69A02F, $68818A54,  
    $CEB2296F, $53C0843A, $FE893655, $25BFE68A,  
    $B4628ABC, $CF222EBF, $25AC6F48, $A9A99387,  
    $53BDDDB65, $E76FFBE7, $E967FD78, $0BA93563,  
    $8E342BC1, $E8A11BE9, $4980740D, $C8087DFC,  
    $8DE4BF99, $A11101A0, $7FD37975, $DA5A26C0,  
    $E81F994F, $9528CD89, $FD339FED, $B87834BF,  
    $5F04456D, $22258698, $C9C4C83B, $2DC156BE,  
    $4F628DAA, $57F55EC5, $E2220ABE, $D2916EBF,  
    $4EC75B95, $24F2C3C0, $42D15D99, $CD0D7FA0,  
    $7B6E27FF, $A8DC8AF0, $7345C106, $F41E232F,  
    $35162386, $E6EA8926, $3333B094, $157EC6F2,  
    $372B74AF, $692573E4, $E9A9D848, $F3160289,  
    $3A62EF1D, $A787E238, $F3A5F676, $74364853,  
    $20951063, $4576698D, $B6FAD407, $592AF950,  
    $36F73523, $4CFB6E87, $7DA4CEC0, $6C152DAA,  
    $CB0396A8, $C50DFE5D, $FCD707AB, $0921C42F,  
    $89DFF0BB, $5FE2BE78, $448F4F33, $754613C9,  
    $2B05D08D, $48B9D585, $DC049441, $C8098F9B,  
    $7DEDE786, $C39A3373, $42410005, $6A091751,  
    $0EF3C8A6, $890072D6, $28207682, $A9A9F7BE,  
    $BF32679D, $D45B5B75, $B353FD00, $CBB0E358,
```


\$830F220A, \$1F8FB214, \$D372CF08, \$CC3C4A13,
\$8CF63166, \$061C87BE, \$88C98F88, \$6062E397,
\$47CF8E7A, \$B6C85283, \$3CC2ACFB, \$3FC06976,
\$4E8F0252, \$64D8314D, \$DA3870E3, \$1E665459,
\$C10908F0, \$513021A5, \$6C5B68B7, \$822F8AA0,
\$3007CD3E, \$74719EEF, \$DC872681, \$073340D4,
\$7E432FD9, \$0C5EC241, \$8809286C, \$F592D891,
\$08A930F6, \$957EF305, \$B7FBFFBD, \$C266E96F,
\$6FE4AC98, \$B173ECC0, \$BC60B42A, \$953498DA,
\$FBA1AE12, \$2D4BD736, \$0F25FAAB, \$A4F3FCEB,
\$E2969123, \$257F0C3D, \$9348AF49, \$361400BC,
\$E8816F4A, \$3814F200, \$A3F94043, \$9C7A54C2,
\$BC704F57, \$DA41E7F9, \$C25AD33A, \$54F4A084,
\$B17F5505, \$59357CBE, \$EDBD15C8, \$7F97C5AB,
\$BA5AC7B5, \$B6F6DEAF, \$3A479C3A, \$5302DA25,
\$653D7E6A, \$54268D49, \$51A477EA, \$5017D55B,
\$D7D25D88, \$44136C76, \$0404A8C8, \$B8E5A121,
\$B81A928A, \$60ED5869, \$97C55B96, \$EAEC991B,
\$29935913, \$01FDB7F1, \$088E8DFA, \$9AB6F6F5,
\$3B4CBF9F, \$4A5DE3AB, \$E6051D35, \$A0E1D855,
\$D36B4CF1, \$F544EDEB, \$B0E93524, \$BEBB8FBD,
\$A2D762CF, \$49C92F54, \$38B5F331, \$7128A454,
\$48392905, \$A65B1DB8, \$851C97BD, \$D675CF2F
);

Substitution Box 7

```
cast_sbox7: array[0..255] of DWord = (  
    $85E04019, $332BF567, $662DBFFF, $CFC65693,  
    $2A8D7F6F, $AB9BC912, $DE6008A1, $2028DA1F,  
    $0227BCE7, $4D642916, $18FAC300, $50F18B82,  
    $2CB2CB11, $B232E75C, $4B3695F2, $B28707DE,  
    $A05FBCF6, $CD4181E9, $E150210C, $E24EF1BD,  
    $B168C381, $FDE4E789, $5C79B0D8, $1E8BFD43,  
    $4D495001, $38BE4341, $913CEE1D, $92A79C3F,  
    $089766BE, $BAEEADF4, $1286BECF, $B6EACB19,  
    $2660C200, $7565BDE4, $64241F7A, $8248DCA9,  
    $C3B3AD66, $28136086, $0BD8DFA8, $356D1CF2,  
    $107789BE, $B3B2E9CE, $0502AA8F, $0BC0351E,  
    $166BF52A, $EB12FF82, $E3486911, $D34D7516,  
    $4E7B3AFF, $5F43671B, $9CF6E037, $4981AC83,  
    $334266CE, $8C9341B7, $D0D854C0, $CB3A6C88,  
    $47BC2829, $4725BA37, $A66AD22B, $7AD61F1E,  
    $0C5CBAFA, $4437F107, $B6E79962, $42D2D816,  
    $0A961288, $E1A5C06E, $13749E67, $72FC081A,  
    $B1D139F7, $F9583745, $CF19DF58, $BEC3F756,  
    $C06EBA30, $07211B24, $45C28829, $C95E317F,  
    $BC8EC511, $38BC46E9, $C6E6FA14, $BAE8584A,  
    $AD4EBC46, $468F508B, $7829435F, $F124183B,  
    $821DBA9F, $AFF60FF4, $EA2C4E6D, $16E39264,  
    $92544A8B, $009B4FC3, $ABA68CED, $9AC96F78,  
    $06A5B79A, $B2856E6E, $1AEC3CA9, $BE838688,  
    $0E0804E9, $55F1BE56, $E7E5363B, $B3A1F25D,  
    $F7DEBB85, $61FE033C, $16746233, $3C034C28,  
    $DA6D0C74, $79AAC56C, $3CE4E1AD, $51F0C802,  
    $98F8F35A, $1626A49F, $EED82B29, $1D382FE3,  
    $0C4FB99A, $BB325778, $3EC6D97B, $6E77A6A9,  
    $CB658B5C, $D45230C7, $2BD1408B, $60C03EB7,  
    $B9068D78, $A33754F4, $F430C87D, $C8A71302,  
    $B96D8C32, $EBD4E7BE, $BE8B9D2D, $7979FB06,  
    $E7225308, $8B75CF77, $11EF8DA4, $E083C858,  
    $8D6B786F, $5A6317A6, $FA5CF7A0, $5DDA0033,  
    $F28EBFB0, $F5B9C310, $A0EAC280, $08B9767A,  
    $A3D9D2B0, $79D34217, $021A718D, $9AC6336A,  
    $2711FD60, $438050E3, $069908A8, $3D7FEDC4,  
    $826D2BEF, $4EEB8476, $488DCF25, $36C9D566,  
    $28E74E41, $C2610ACA, $3D49A9CF, $BAE3B9DF,  
    $B65F8DE6, $92AEAF64, $3AC7D5E6, $9EA80509,  
    $F22B017D, $A4173F70, $DD1E16C3, $15E0D7F9,
```

\$50B1B887, \$2B9F4FD5, \$625ABA82, \$6A017962,
\$2EC01B9C, \$15488AA9, \$D716E740, \$40055A2C,
\$93D29A22, \$E32DBF9A, \$058745B9, \$3453DC1E,
\$D699296E, \$496CFF6F, \$1C9F4986, \$DFE2ED07,
\$B87242D1, \$19DE7EAE, \$053E561A, \$15AD6F8C,
\$66626C1C, \$7154C24C, \$EA082B2A, \$93EB2939,
\$17DCB0F0, \$58D4F2AE, \$9EA294FB, \$52CF564C,
\$9883FE66, \$2EC40581, \$763953C3, \$01D6692E,
\$D3A0C108, \$A1E7160E, \$E4F2DFA6, \$693ED285,
\$74904698, \$4C2B0EDD, \$4F757656, \$5D393378,
\$A132234F, \$3D321C5D, \$C3F5E194, \$4B269301,
\$C79F022F, \$3C997E7E, \$5E4F9504, \$3FFAFBBD,
\$76F7AD0E, \$296693F4, \$3D1FCE6F, \$C61E45BE,
\$D3B5AB34, \$F72BF9B7, \$1B0434C0, \$4E72B567,
\$5592A33D, \$B5229301, \$CFD2A87F, \$60AEB767,
\$1814386B, \$30BCC33D, \$38A0C07D, \$FD1606F2,
\$C363519B, \$589DD390, \$5479F8E6, \$1CB8D647,
\$97FD61A9, \$EA7759F4, \$2D57539D, \$569A58CF,
\$E84E63AD, \$462E1B78, \$6580F87E, \$F3817914,
\$91DA55F4, \$40A230F3, \$D1988F35, \$B6E318D2,
\$3FFA50BC, \$3D40F021, \$C3C0BDAE, \$4958C24C,
\$518F36B2, \$84B1D370, \$0FEDCE83, \$878DDADA,
\$F2A279C7, \$94E01BE8, \$90716F4B, \$954B8AA3
);

Substitution Box 8

```
cast_sbox8: array[0..255] of DWord = (  
    $E216300D, $BBDDFFFC, $A7EBDABD, $35648095,  
    $7789F8B7, $E6C1121B, $0E241600, $052CE8B5,  
    $11A9CFB0, $E5952F11, $ECE7990A, $9386D174,  
    $2A42931C, $76E38111, $B12DEF3A, $37DDDDFC,  
    $DE9ADEB1, $0A0CC32C, $BE197029, $84A00940,  
    $BB243A0F, $B4D137CF, $B44E79F0, $049EEDFD,  
    $0B15A15D, $480D3168, $8BBBDE5A, $669DED42,  
    $C7ECE831, $3F8F95E7, $72DF191B, $7580330D,  
    $94074251, $5C7DCDFA, $ABBE6D63, $AA402164,  
    $B301D40A, $02E7D1CA, $53571DAE, $7A3182A2,  
    $12A8DDEC, $FDAA335D, $176F43E8, $71FB46D4,  
    $38129022, $CE949AD4, $B84769AD, $965BD862,  
    $82F3D055, $66FB9767, $15B80B4E, $1D5B47A0,  
    $4CFDE06F, $C28EC4B8, $57E8726E, $647A78FC,  
    $99865D44, $608BD593, $6C200E03, $39DC5FF6,  
    $5D0B00A3, $AE63AFF2, $7E8BD632, $70108C0C,  
    $BBD35049, $2998DF04, $980CF42A, $9B6DF491,  
    $9E7EDD53, $06918548, $58CB7E07, $3B74EF2E,  
    $522FFFB1, $D24708CC, $1C7E27CD, $A4EB215B,  
    $3CF1D2E2, $19B47A38, $424F7618, $35856039,  
    $9D17DEE7, $27EB35E6, $C9AFF67B, $36BAF5B8,  
    $09C467CD, $C18910B1, $E11DBF7B, $06CD1AF8,  
    $7170C608, $2D5E3354, $D4DE495A, $64C6D006,  
    $BCC0C62C, $3DD00DB3, $708F8F34, $77D51B42,  
    $264F620F, $24B8D2BF, $15C1B79E, $46A52564,  
    $F8D7E54E, $3E378160, $7895CDA5, $859C15A5,  
    $E6459788, $C37BC75F, $DB07BA0C, $0676A3AB,  
    $7F229B1E, $31842E7B, $24259FD7, $F8BEF472,  
    $835FFCB8, $6DF4C1F2, $96F5B195, $FD0AF0FC,  
    $B0FE134C, $E2506D3D, $4F9B12EA, $F215F225,  
    $A223736F, $9FB4C428, $25D04979, $34C713F8,  
    $C4618187, $EA7A6E98, $7CD16EFC, $1436876C,  
    $F1544107, $BEDEEE14, $56E9AF27, $A04AA441,  
    $3CF7C899, $92ECBAE6, $DD67016D, $151682EB,  
    $A842EEDF, $FDBA60B4, $F1907B75, $20E3030F,  
    $24D8C29E, $E139673B, $EFA63FB8, $71873054,  
    $B6F2CF3B, $9F326442, $CB15A4CC, $B01A4504,  
    $F1E47D8D, $844A1BE5, $BAE7DFDC, $42CBDA70,  
    $CD7DAE0A, $57E85B7A, $D53F5AF6, $20CF4D8C,  
    $CEA4D428, $79D130A4, $3486EBFB, $33D3CDDC,  
    $77853B53, $37EFFCB5, $C5068778, $E580B3E6,
```

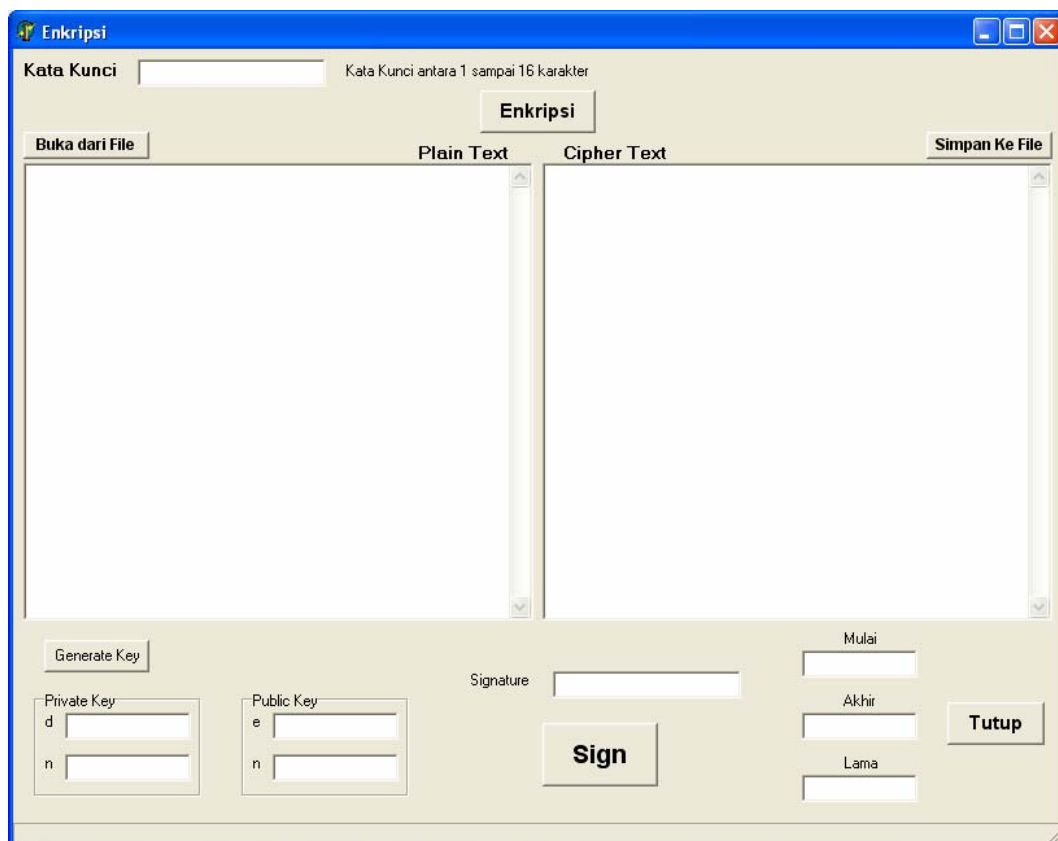
\$4E68B8F4, \$C5C8B37E, \$0D809EA2, \$398FEB7C,
\$132A4F94, \$43B7950E, \$2FEE7D1C, \$223613BD,
\$DD06CAA2, \$37DF932B, \$C4248289, \$ACF3EBC3,
\$5715F6B7, \$EF3478DD, \$F267616F, \$C148CBE4,
\$9052815E, \$5E410FAB, \$B48A2465, \$2EDA7FA4,
\$E87B40E4, \$E98EA084, \$5889E9E1, \$EFD390FC,
\$DD07D35B, \$DB485694, \$38D7E5B2, \$57720101,
\$730EDEBC, \$5B643113, \$94917E4F, \$503C2FBA,
\$646F1282, \$7523D24A, \$E0779695, \$F9C17A8F,
\$7A5B2121, \$D187B896, \$29263A4D, \$BA510CDF,
\$81F47C9F, \$AD1163ED, \$EA7B5965, \$1A00726E,
\$11403092, \$00DA6D77, \$4A0CDD61, \$AD1F4603,
\$605BDFB0, \$9EEDC364, \$22EBE6A8, \$CEE7D28A,
\$A0E736A0, \$5564A6B9, \$10853209, \$C7EB8F37,
\$2DE705CA, \$8951570F, \$DF09822B, \$BD691A6C,
\$AA12E4F2, \$87451C0F, \$E0F6A27A, \$3ADA4819,
\$4CF1764F, \$0D771C2B, \$67CDB156, \$350D8384,
\$5938FA0F, \$42399EF3, \$36997B07, \$0E84093D,
\$4AA93E61, \$8360D87B, \$1FA98B0C, \$1149382C,
\$E97625A5, \$0614D1B7, \$0E25244B, \$0C768347,
\$589E8D82, \$0D2059D1, \$A466BB1E, \$F8DA0A82,
\$04F19130, \$BA6E4EC0, \$99265164, \$1EE7230D,
\$50B2AD80, \$EAEE6801, \$8DB2A283, \$EA8BF59E

Lampiran C

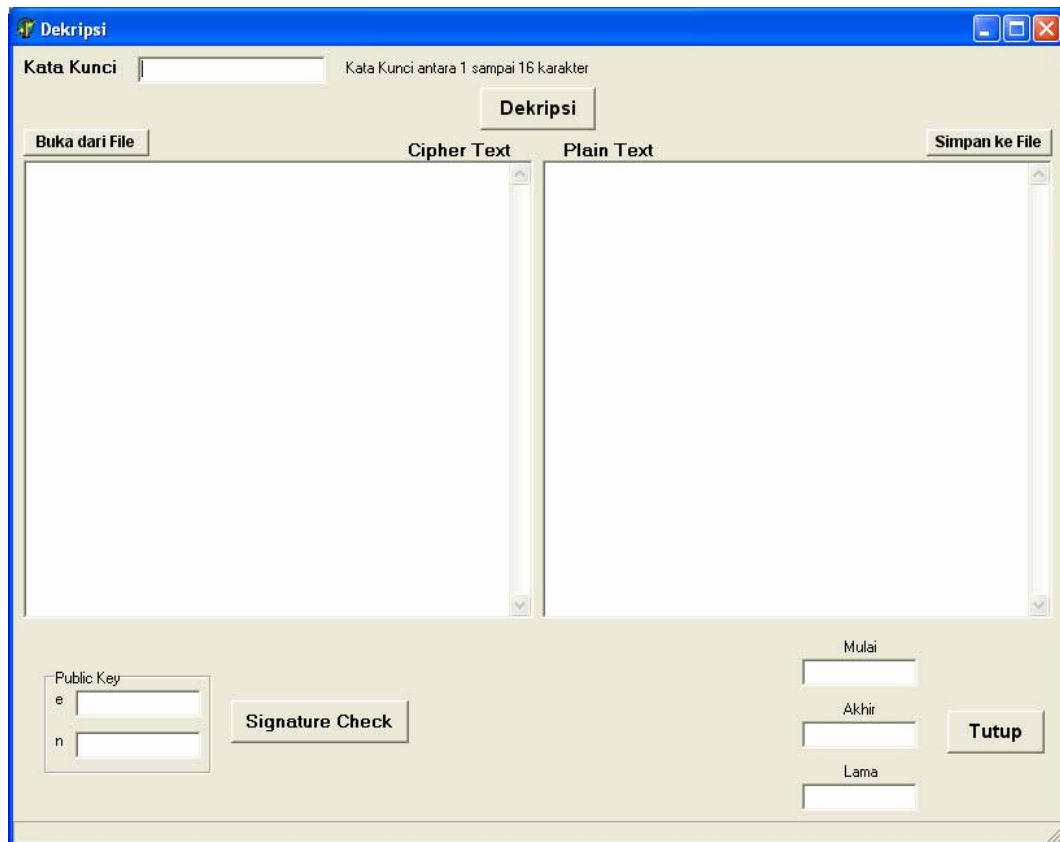
Gambar Tampilan



Gambar Tampilan Program Utama



Gambar Tampilan Program Enkripsi



Gambar Tampilan Program Dekripsi



Gambar Tampilan About Box