using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Security.Cryptography;

using System.IO;

namespace E\_Günce

{

public partial class egunce : Form

{

public egunce()

{

InitializeComponent();

}

private void sifreleFunc(string kaynak, string hedef, string anahtar)

{

AesManaged AES = new AesManaged();

using (MD5CryptoServiceProvider MD5 = new MD5CryptoServiceProvider())

{

AES.KeySize = MD5.HashSize;

AES.BlockSize = MD5.HashSize;

AES.IV = MD5.ComputeHash(ASCIIEncoding.ASCII.GetBytes(anahtar));

AES.Key = MD5.ComputeHash(ASCIIEncoding.ASCII.GetBytes(anahtar));

}

using (FileStream reader = new FileStream(kaynak, FileMode.Open, FileAccess.Read))

{

using (FileStream writer = new FileStream(hedef, FileMode.OpenOrCreate, FileAccess.Write))

{

using (CryptoStream cs = new CryptoStream(writer, AES.CreateEncryptor(), CryptoStreamMode.Write))

{

int bufferSize = 4096;

byte[] buffer = new byte[bufferSize];

int bytesRead;

do

{

bytesRead = reader.Read(buffer, 0, bufferSize);

if (bytesRead != 0)

{

cs.Write(buffer, 0, bytesRead);

}

}

while (bytesRead != 0);

cs.FlushFinalBlock();

}

}

}

}

private void sifrecozFunc(string kaynak, string hedef, string anahtar)

{

AesManaged AES = new AesManaged();

using (MD5CryptoServiceProvider MD5 = new MD5CryptoServiceProvider())

{

AES.KeySize = MD5.HashSize; // 128 bit

AES.BlockSize = MD5.HashSize; // 128 bit

AES.IV = MD5.ComputeHash(ASCIIEncoding.ASCII.GetBytes(anahtar));

AES.Key = MD5.ComputeHash(ASCIIEncoding.ASCII.GetBytes(anahtar));

}

using (FileStream reader = new FileStream(kaynak, FileMode.Open, FileAccess.Read))

{

using (FileStream writer = new FileStream(hedef, FileMode.OpenOrCreate, FileAccess.Write))

{

using (CryptoStream cs = new CryptoStream(reader, AES.CreateDecryptor(), CryptoStreamMode.Read))

{

int bufferSize = 4096;

byte[] buffer = new byte[bufferSize];

int bytesRead;

do

{

bytesRead = cs.Read(buffer, 0, bufferSize);

if (bytesRead != 0)

{

writer.Write(buffer, 0, bytesRead);

}

}

while (bytesRead != 0);

}

}

}

}

private void egunce\_Load(object sender, EventArgs e)

{

string date = DateTime.Now.ToString("dd.MM.yyyy");

this.Text = "E-Günce (Tarih: "+date+")";

sifrecozMetin.ReadOnly = true;

}

private void sifreleKaydet\_Click(object sender, EventArgs e)

{

DialogResult result = folderBrowserDialog1.ShowDialog();

if (result == DialogResult.OK)

{

sifreleKonum.Text = folderBrowserDialog1.SelectedPath;

}

}

private void ac\_Click(object sender, EventArgs e)

{

OpenFileDialog dosyaac = new OpenFileDialog();

if (dosyaac.ShowDialog() == DialogResult.OK)

{

sifrecozKonum.Text = dosyaac.FileName;

}

}

private void sifrele\_Click(object sender, EventArgs e)

{

if (sifreleAnahtar.Text != "" && sifreleMetin.Text != "" && sifreleKonum.Text != "")

{

try

{

string tarih = DateTime.Now.ToString("ddMMyyyyHHmmss");

string temp = System.IO.Path.GetTempPath();

string tempAdres = temp + tarih;

string tarihMetin = DateTime.Now.ToString("dd.MM.yyyy");

string kullanici = System.Security.Principal.WindowsIdentity.GetCurrent().Name;

File.WriteAllText(tempAdres, "Tarih:" + tarihMetin + "\r\n\n\n" + "Yazan:" + kullanici + "\r\n\n\n" + "Metin:" + sifreleMetin.Text);

sifreleFunc(tempAdres, sifreleKonum.Text + "/" + tarih, sifreleAnahtar.Text);

File.Delete(tempAdres);

string kisatarih = DateTime.Now.ToString("dd.MM.yyyy") + " " + DateTime.Now.ToString("hh.mm.ss");

MessageBox.Show(kisatarih + " tarihine ait günlük kaydı oluşturuldu.");

sifreleMetin.Text = ""; sifreleAnahtar.Text = "";

}

catch

{

MessageBox.Show("Bir sorun oluştu.");

}

}

else

{

MessageBox.Show("Tüm alanları doldurun.");

}

}

private void sifrecoz\_Click(object sender, EventArgs e)

{

if (sifrecozAnahtar.Text != "" && sifrecozKonum.Text != "")

{

try

{

string tarih = DateTime.Now.ToString("ddMMyyyyHHmmss");

string temp = System.IO.Path.GetTempPath();

string tempAdres = temp + tarih + ".GUNCE";

sifrecozFunc(sifrecozKonum.Text, tempAdres, sifrecozAnahtar.Text);

sifrecozMetin.Text = File.ReadAllText(tempAdres);

MessageBox.Show("Dosyanın şifresi çözüldü.");

File.Delete(@tempAdres);

}

catch

{

MessageBox.Show("Bir sorun oluştu.");

}

}

else

{

MessageBox.Show("Tüm alanları doldurun.");

}

}

private void temizle1\_Click(object sender, EventArgs e)

{

sifreleAnahtar.Text = "";

sifreleMetin.Text = "";

sifreleKonum.Text = "";

}

private void temizle2\_Click(object sender, EventArgs e)

{

sifrecozAnahtar.Text = "";

sifrecozKonum.Text = "";

sifrecozMetin.Text = "";

}

}

}