Deccan Education Society's
Kirti M. Doongursee College of Arts, Science and Commerce
[NAAC Accredited: "A Grade"]

T.Y.B.Sc. [Computer Science]

Project Documentation

USCSP503

Seat Number [ ]

Roll No: 64

Student ID: 681400

Department of Computer Science and Information Technology

Department of Computer Science and Information Technology
Deccan Education Society's
Kirti M. Doongursee College of Arts, Science and Commerce
[NAAC Accredited: "A Grade"]

C E R T I F I C A T E

This is to certify that Mr. / ~~Miss.~~ **SHLOK G. SHIVKAR** of T. Y. B.Sc. (Computer Science) with Seat No. **64** has completed the Project Implementation - USCS503 under my supervision in this College during the year 2020-2021.

Lecturer-In-Charge	**H.O.D.** **Department of** **Computer Science & IT**
Date: / / 2020	Date: / / 2020
Examined by:	**Remarks:**
Date: / / 2020	____________________
	____________________

A Project Report
On

"Data Cryptography"
Windows Application: Crypt-IT

Submitted in partial fulfillment of the requirement of
University of Mumbai

For the Degree of
Bachelor of Computer Science

Submitted By
SHLOK G. SHIVKAR

Under the Guidance of
Prof. SIDDHESH KADAM

Department of Computer Science and Information Technology

Deccan Education Society's
Kirti M. Doongursee College of Arts, Science and commerce
[NAAC Accredited: "A Grade"]
Mumbai

(2020-2021)

INDEX

Windows Application

Crypt - IT

Designed and Developed
by
Shlok G. Shivkar

Introduction

Crypt-IT is a windows desktop application which can be used for encryption and decryption of your data.

Crypt-IT offers 2 types of encryption methods.

- Encryption of a string.
- Encryption of a file.

Crypt-IT uses 2 types of widely used algorithms for encryption.

- Message Digest Algorithm (MD5).
- Advanced Encryption Standard (AES).

Requirements

System Requirements :

- 633 MHz – Pentium Processor or more.
- 256 MB RAM (512MB, 1GB recommended).
- Minimum 2GB hard disk (80GB recommended).
- Windows 2000/XP. (Windows 7 recommended).

Software :

- ➢ Visual Studio 2019.

Technology :

- ➢ Windows Forms App (.Net Framework).

Language :

- ➢ C Sharp (C#).

Methodology

This project is based on the concept of data cryptography

In this project, I have created 2 modules

1. Encryption of a string.
2. Encryption of a file.

1. ENCRYPTION OF A STRING :

The first module uses 2 methods to encrypt the data.

a. Random Hash Function.
b. Message Digest Algorithm (MD5).

a. Random Hash Function :

A **hash function** is used to generate the new value according to a mathematical algorithm. The result of a hash function is known as a **hash value** or simply, a **hash**.

This Random Hash value is then used by the MD5 Algorithm to encrypt the entire string.

b. Message Digest Algorithm (MD5) :

MD5 uses 4 steps for execution.

- Append Padding Bits : Padding means adding extra bits to the original message
- Append Length : After padding, 64 bits are inserted at the end which is used to record the length of the original input
- Initialize MD buffer : A four-word buffer (A, B, C, D) is used to compute the values for the message digest
- Processing message in 16 word block: MD5 uses the auxiliary functions which take the input as three 32-bit number and produces a 32-bit output. These functions use logical operators like OR, XOR, NOR.

2. ENCRYPTION OF A FILE :

This module uses a static default password set as “Shloksgs123456” which is used for authentication in AES algorithm.

a. **Advanced Encryption Standard (AES Algorithm):** AES works with Substitution-permutation method. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

Flowchart / Diagrams

Working of Hashing Process :

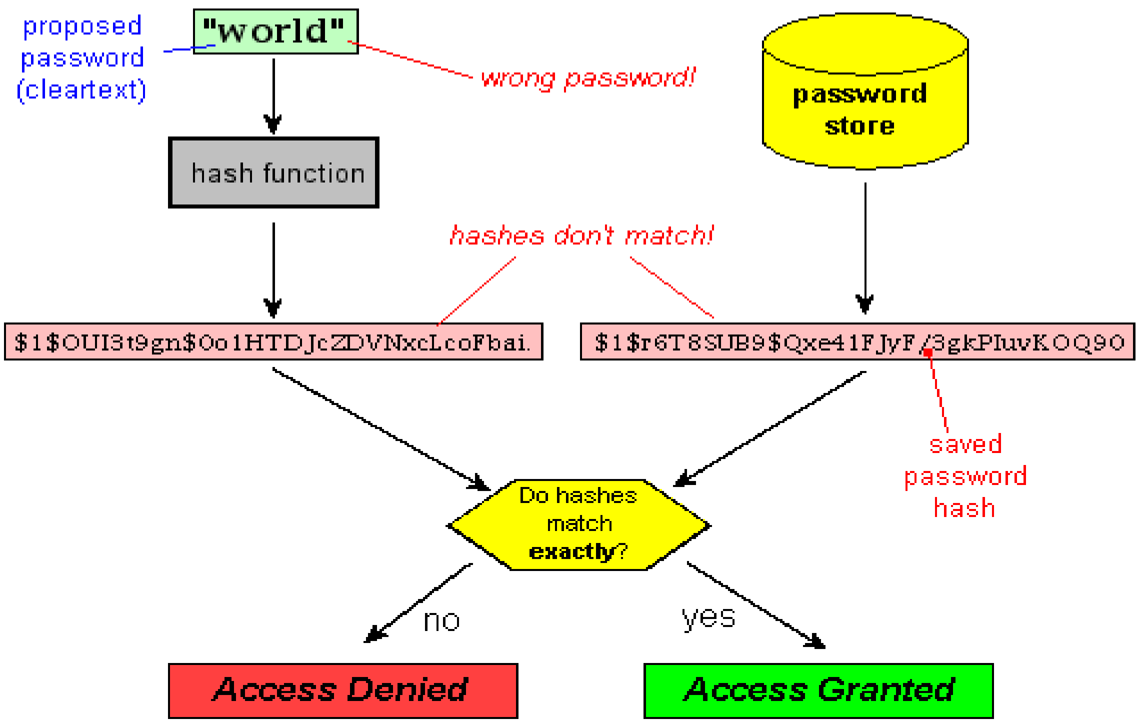


Figure No. 1 : Hashing process

Flow of Encryption with string :

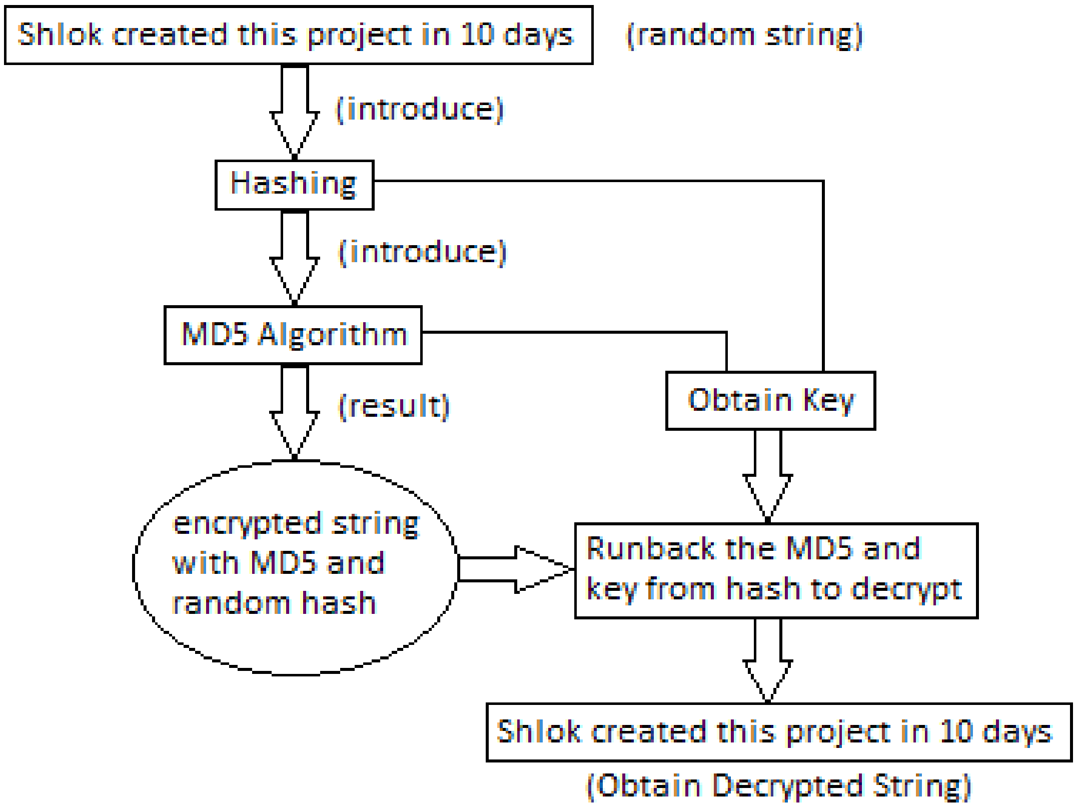

Shlok created this project in 10 days
(random string)
(introduce)
Hashing
(introduce)
MD5 Algorithm
(result)
Obtain Key
encrypted string with MD5 and random hash
Runback the MD5 and key from hash to decrypt
Shlok created this project in 10 days
(Obtain Decrypted String)

Figure No. 2 : Encryption with string

Flow of MD5 Algorithm :

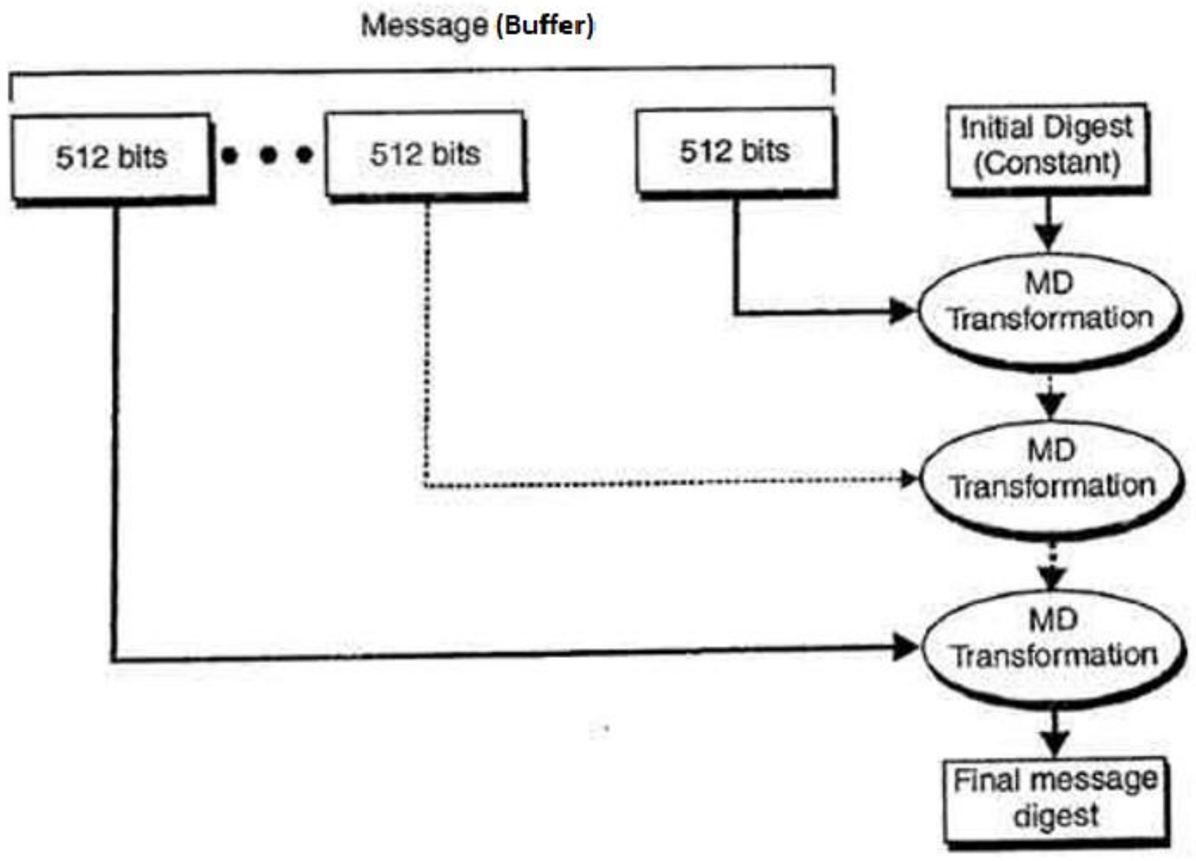


Figure No. 3 : MD5 Algorithm

<u>Working of AES Algorithm :</u>

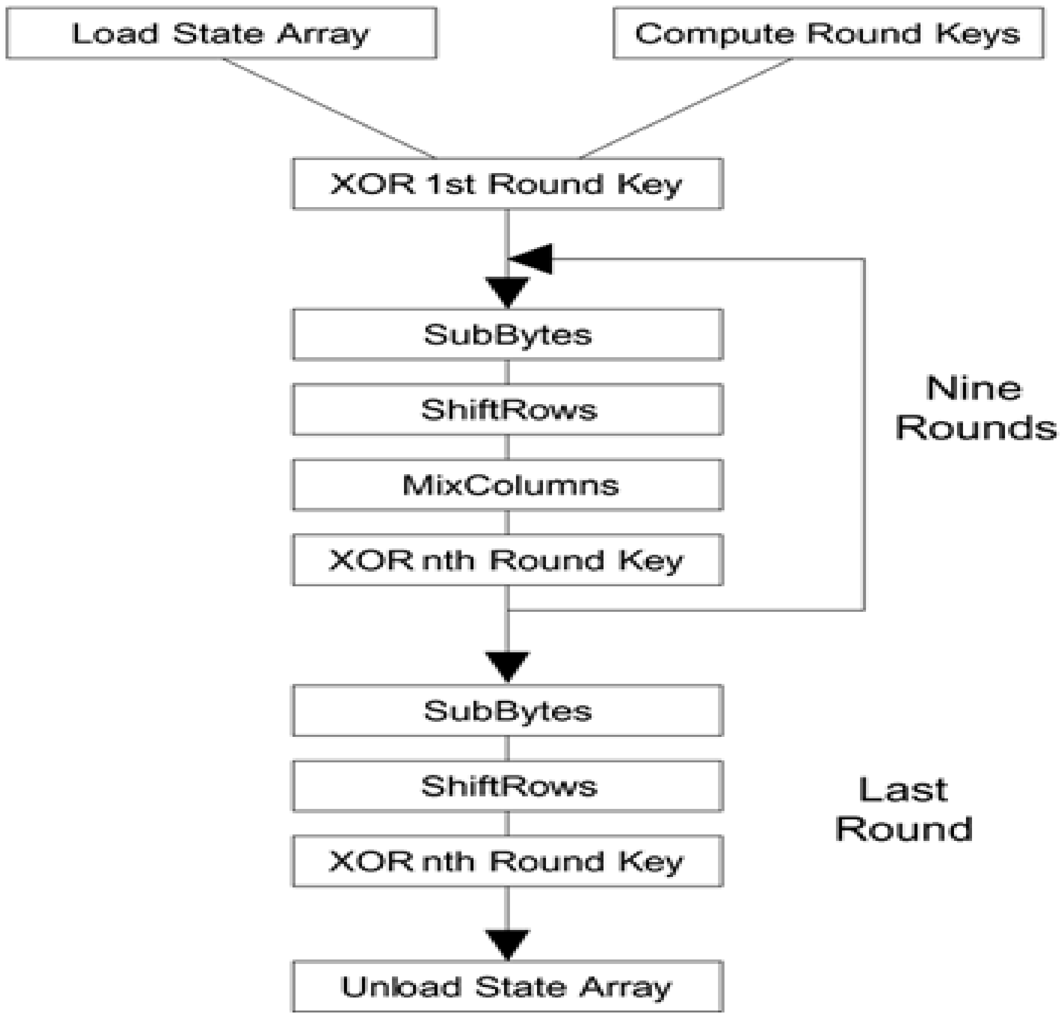


Figure No. 4 : AES Algorithm

Source Code

Form1.cs :

```
using System;
using System.IO;
using System.Runtime.InteropServices;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Security.Cryptography;
using System.Text.RegularExpressions;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace crypt_it
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void button2_Click(object sender, EventArgs e)
        {
            this.Hide();
            Form3 f3 = new Form3();
            f3.Show();
        }

        private void Form1_FormClosing(object sender,
FormClosingEventArgs e)
        {
            Application.Exit();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            this.Hide();
            Form2 f2 = new Form2();
            f2.Show();
        }

        private void linkLabel1_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
        {

        }
    }
}
```

Form1.cs[Design]:

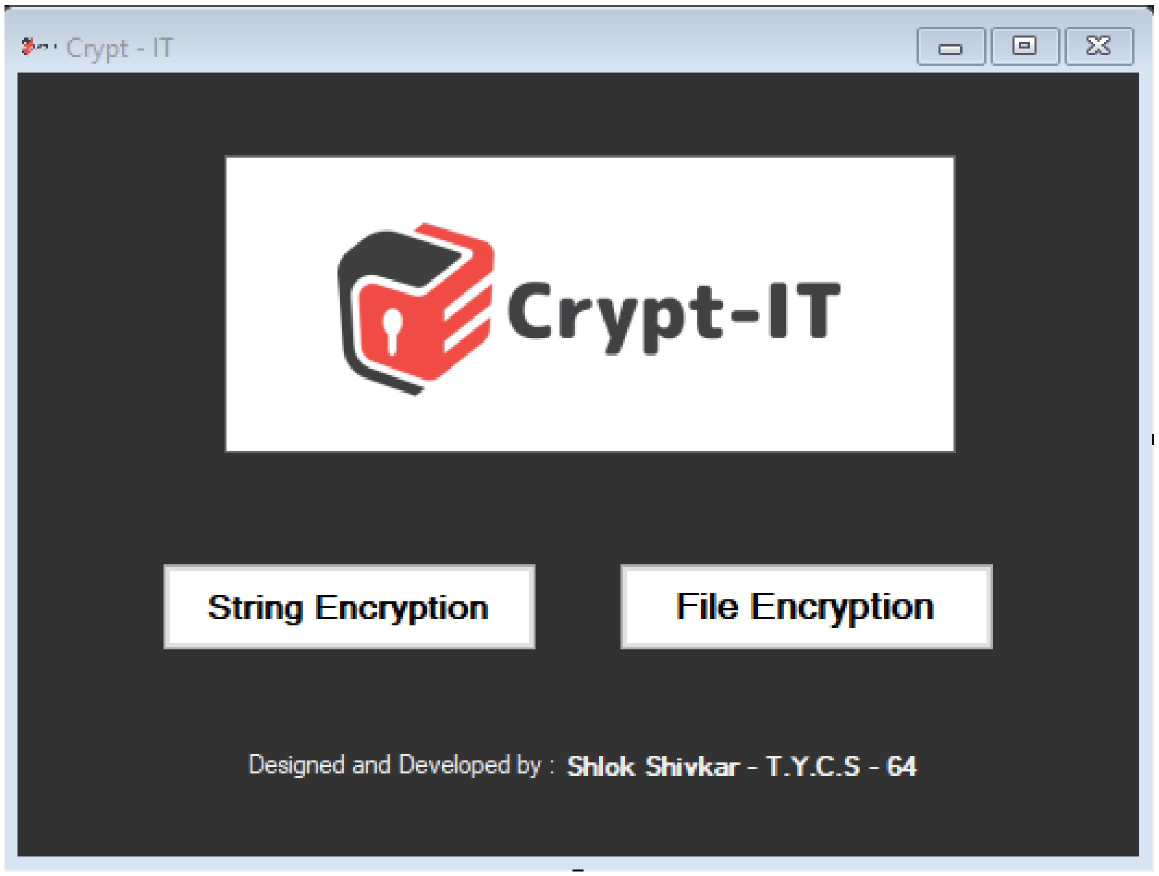


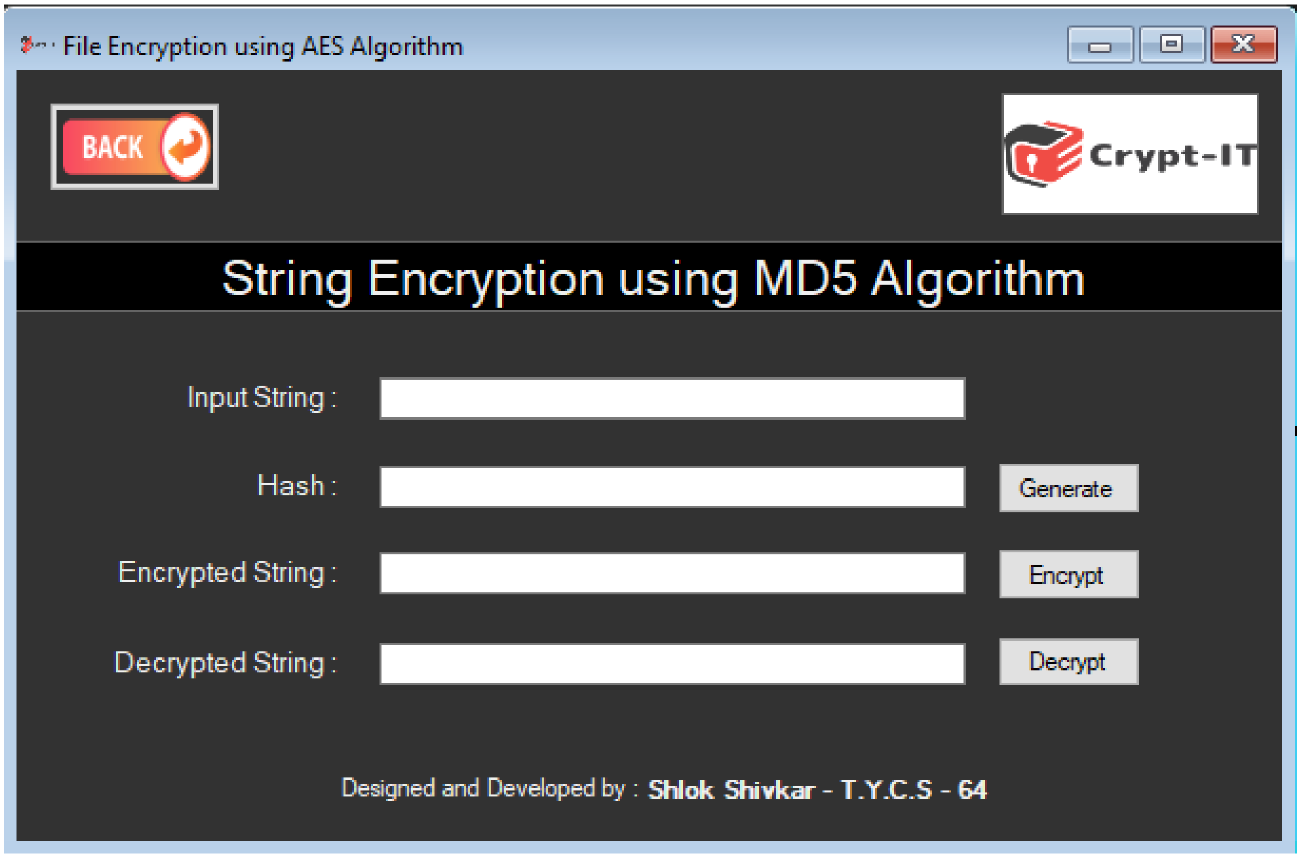
Figure No. 5 : Form1.cs

Form2.cs :

```
using System;
using System.IO;
using System.Runtime.InteropServices;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Security.Cryptography;
using System.Text.RegularExpressions;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace crypt_it
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
            textBox1.Text = "Enter a string here";
            textBox2.Text = "Press the generate button to get a
random hash";
            textBox3.Text = "Press the Encrypt button to encrypt the
string";
            textBox4.Text = "Press the Decrypt button to decrypt the
string";
```

```
        }

        private void Form2_FormClosing(object sender,
FormClosingEventArgs e)
        {
            Application.Exit();
        }


        private void button1_Click(object sender, EventArgs e)
        {
            this.Hide();
            Form1 f1 = new Form1();
            f1.Show();

        }

        private void button2_Click(object sender, EventArgs e)
        {
            var rBytes = new byte[24];
            using (var crypto = new RNGCryptoServiceProvider())
crypto.GetBytes(rBytes);
            var base64 = Convert.ToBase64String(rBytes);
            var result = Regex.Replace(base64, "[A-Za-z0-9]", "");
            textBox2.Text = base64;
        }

        private void button3_Click(object sender, EventArgs e)
        {
            byte[] data = 
UTF8Encoding.UTF8.GetBytes(textBox1.Text);
            using (MD5CryptoServiceProvider md5 = new 
MD5CryptoServiceProvider())
            {
                byte[] keys = 
md5.ComputeHash(UTF8Encoding.UTF8.GetBytes(textBox2.T
ext));
                using (TripleDESCryptoServiceProvider tripDes = 
new TripleDESCryptoServiceProvider() { Key = keys, Mode = 
CipherMode.ECB, Padding = PaddingMode.PKCS7 })
                {
                    ICryptoTransform transform = 
tripDes.CreateEncryptor();
                    byte[] results = 
transform.TransformFinalBlock(data, 0, data.Length);
                    textBox3.Text = Convert.ToBase64String(results, 0, 
results.Length);
                }
            }
        }

        private void button4_Click(object sender, EventArgs e)
        {
            byte[] data = 
Convert.FromBase64String(textBox3.Text);
            using (MD5CryptoServiceProvider md5 = new 
MD5CryptoServiceProvider())
            {
                byte[] keys =
md5.ComputeHash(UTF8Encoding.UTF8.GetBytes(textBox2.T
ext));
                using (TripleDESCryptoServiceProvider tripDes =
new TripleDESCryptoServiceProvider() { Key = keys, Mode =
CipherMode.ECB, Padding = PaddingMode.PKCS7 })
                {
                    ICryptoTransform transform =
tripDes.CreateDecryptor();
                    byte[] results =
transform.TransformFinalBlock(data, 0, data.Length);
                    textBox4.Text =
UTF8Encoding.UTF8.GetString(results);
                }
            }
        }
    }
}
```

<u>Form2.cs[Design] :</u>


File Encryption using AES Algorithm

String Encryption using MD5 Algorithm

Input String :

Hash :

Encrypted String :

Decrypted String :

Designed and Developed by : **Shlok Shivkar - T.Y.C.S - 64**

Figure No. 6 : Form2.cs

Form3.cs :

```
using System;
using System.IO;
using System.Runtime.InteropServices;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Security.Cryptography;
using System.Text.RegularExpressions;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace crypt_it
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
            richTextBox1.Text = "Click Browse to select a file";
        }

        private void Form3_FormClosing(object sender,
FormClosingEventArgs e)
        {
            Application.Exit();
        }
    private void button1_Click(object sender, EventArgs e)
    {
        this.Hide();
        Form1 f1 = new Form1();
        f1.Show();

    }

    private void button1_Click_1(object sender, EventArgs e)
    {
      OpenFileDialog ofd = new OpenFileDialog();
      ofd.Filter = "All files|*.*";


      if (ofd.ShowDialog() == DialogResult.OK)
      {
        richTextBox1.Text = ofd.FileName;
      }
    }
      // Zeromemory method is used to flush out the
previously used security key

      public static byte[] GenerateSalt()
      {
        byte[] data = new byte[32];
        using (RNGCryptoServiceProvider
rgnCryptoServiceProvider = new RNGCryptoServiceProvider())
        {
          rgnCryptoServiceProvider.GetBytes(data);
        }
```

```
            return data;
          }
        // Code for File Encryption
        [DllImport("KERNEL32.DLL", EntryPoint =
"RtlZeroMemory")]
        public static extern bool ZeroMemory(IntPtr Destination,
int Length);
        private void FileEncrypt(string inputFile, string password)
          {
            byte[] salt = GenerateSalt();
            byte[] passwords =
Encoding.UTF8.GetBytes(password);
            RijndaelManaged AES = new RijndaelManaged();
            AES.KeySize = 256;// 256 bits
            AES.BlockSize = 128;// 128 bits
            AES.Padding = PaddingMode.PKCS7;
            var key = new Rfc2898DeriveBytes(passwords, salt,
50000);
            AES.Key = key.GetBytes(AES.KeySize / 8);
            AES.IV = key.GetBytes(AES.BlockSize / 8);
            AES.Mode = CipherMode.CBC;
            using (FileStream fsCrypt = new FileStream(inputFile
+ ".aes", FileMode.Create))
            {
              fsCrypt.Write(salt, 0, salt.Length);
              using (CryptoStream cs = new
CryptoStream(fsCrypt, AES.CreateEncryptor(),
CryptoStreamMode.Write))
              {
                using (FileStream fsIn = new
FileStream(inputFile, FileMode.Open))
```

```
                        {
                            byte[] buffer = new byte[1048576];
                            int read;
                            while ((read = fsIn.Read(buffer, 0,
buffer.Length)) > 0)
                            {
                                cs.Write(buffer, 0, read);
                                richTextBox1.Text = "   File Encrypted
Successfully !!! \n\n   Encrypted file is saved in the same
directory. \n   Remember to open the encrypted .aes file in
wordpad. \n   Select the Encrypted File now to Decrypt it.";
                            }
                        }
                    }
                }
            }

            // Code for File Decryption.

            private void FileDecrypt(string inputFileName, string
outputFileName, string password)
            {
                byte[] passwords =
Encoding.UTF8.GetBytes(password);
                byte[] salt = new byte[32];
                using (FileStream fsCrypt = new
FileStream(inputFileName, FileMode.Open))
                {
                    fsCrypt.Read(salt, 0, salt.Length);
                    RijndaelManaged AES = new RijndaelManaged();
                    AES.KeySize = 256;// 256 bits
                    AES.BlockSize = 128;// 128 bits
                    var key = new Rfc2898DeriveBytes(passwords,
salt, 50000);
                    AES.Key = key.GetBytes(AES.KeySize / 8);
                    AES.IV = key.GetBytes(AES.BlockSize / 8);
                    AES.Padding = PaddingMode.PKCS7;
                    AES.Mode = CipherMode.CBC;
                    using (CryptoStream cryptoStream = new
CryptoStream(fsCrypt, AES.CreateDecryptor(),
CryptoStreamMode.Read))
                    {
                        using (FileStream fsOut = new
FileStream(outputFileName, FileMode.Create))
                        {
                            int read;
                            byte[] buffer = new byte[1048576];
                            while ((read = cryptoStream.Read(buffer, 0,
buffer.Length)) > 0)
                            {
                                fsOut.Write(buffer, 0, read);
                                richTextBox1.Text = "   File Decrypted
Successfully !!! \n\n   Decrypted file is saved in the same
directory. \n   Remember to open the decrypted file in the same
method as the original one.";
                            }
                        }
                    }
                }
            }
        private void button2_Click_1(object sender, EventArgs e)
        {
            string password = "Shloksgs123456"; // Same password
has to be passed while encrypting and decrypting
            GCHandle gCHandle = GCHandle.Alloc(password,
GCHandleType.Pinned);
            FileEncrypt(richTextBox1.Text, password);
            ZeroMemory(gCHandle.AddrOfPinnedObject(),
password.Length * 2);
            gCHandle.Free();
        }

        private void button3_Click_1(object sender, EventArgs e)
        {
            string password = "Shloksgs123456"; // Same password
has to be passed while encrypting and decrypting
            GCHandle gch = GCHandle.Alloc(password,
GCHandleType.Pinned);
            FileDecrypt(richTextBox1.Text, richTextBox1.Text + "
decrypted.txt", password);
            ZeroMemory(gch.AddrOfPinnedObject(),
password.Length * 2);
            gch.Free();
        }
    }
    }
```

Form3.cs[Design] :

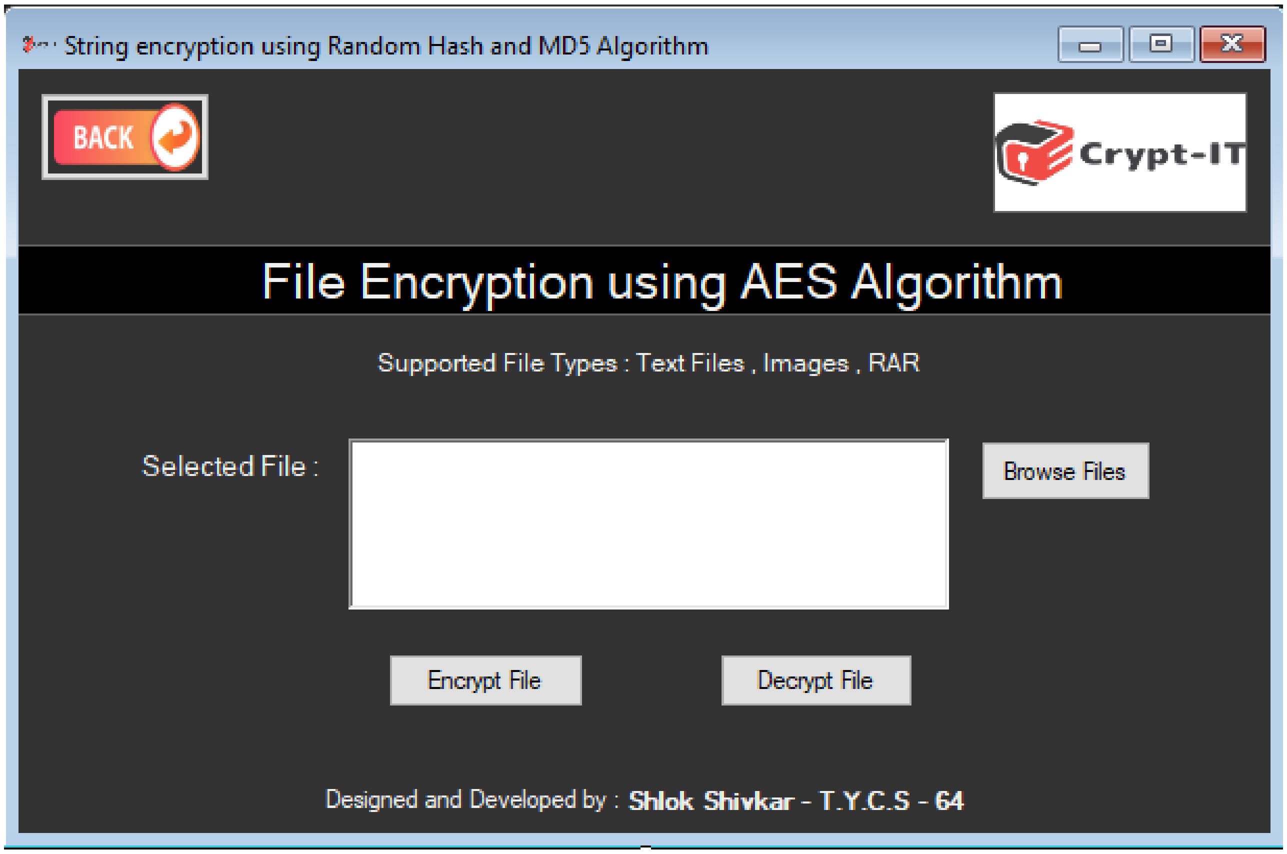

String encryption using Random Hash and MD5 Algorithm
BACK
Crypt-IT
File Encryption using AES Algorithm
Supported File Types : Text Files , Images , RAR
Selected File :
Browse Files
Encrypt File
Decrypt File
Designed and Developed by : Shlok Shivkar - T.Y.C.S - 64

Figure No. 7 : Form3.cs

Conclusion :

This project offers a way to keep our data safe from potential leaks and hackers.

In the upcoming time as most of our data is saved online, cryptography helps to keep our data secure.

Cryptography can guard the information and communication from unauthorized revelation and access of information.

This project has a scope for further improvement and upgradation.

References :

- www.google.com
- www.stackoverflow.com
- www.foxlearn.com
- www.youtube.com
- www.tutorialspoint.com