Introduction

**INTRODUCTION TO THE PROJECT**

File Rakshak s/w is a fast, simple s/w to store and protect critical and sensitive files on any drive.File Rakshak s/w creates a private vault and simply drag-and-drop your valuable files inside.

It enables you to encrypt and password protect any type of file, including documents, images, videos etc.Once protected, the file can only be opened with the proper password.

* It can protect any type of file.
* Integration with the context-menu of Windows.
* Graphical user interface style Windows simple and functional.
* Protection files through the most advanced algorithm RSA & RC4
* High safety of the protected files.
* Possibility to order and to group the files in the principal list.
* Lock files and folders with passwords
* Hide folders from System of verification against the change and/or corruption of protected file.
* Restrict access to Control Panel and other resources guest user.
* Automatic system of protection against password “weak.”
* Use the concept of Steganography
* Use the concept of Cryptography

Feasibility Study

**Chapter 1**

**FEASIBILITY STUDY**

The next step in analysis is to verify the feasibility of the proposed system. “All projects are feasible given unlimited resources and infinite time“. But in reality both resources and time are scarce. Project should confirm to time bounce and should be optimal in their consumption of resources. This place a constant is approval of any project.

Feasibility has applied to **File Rakshak** pertains to the following areas:

* Economical Feasibility
* Behavioral Feasibility
* Technichal Feasibility
* GUI Feasibility

**2.1.Behavioral Feasibility:**

If we compare this system to existing one then we see that operating this system is easier than existing one. We can easily use this system...

**2.2 Economic Feasibility:**

To decide whether a project is economically feasible, we have to consider various factors as:

* Cost benefit analysis
* Long-term returns
* Maintenance costs

**2.3. Technichal Feasibility:**

The technical requirement for the system is economic and it does not use any other additional Hardware and software.

**2.4. GUI Feasibility:**

Theoretically, a C sharp compiler could generate machine code like traditional compilers of C plus plus. Then it comes the turn of the operating system. Windows XP is used in this project. By design, C sharp is the programming language that most directly reflects the underlying Common Language Infrastructure. In this System, Date's "Information Principle" states: "the entire information content of the database is represented in one and only one way. Namely as explicit values in column positions (attributes) and rows in relations (tuples). Therefore, there

Feasibility Study

are no explicit pointers between related tables." This contrasts with the object database management system (ODBMS), which does store explicit pointers between related types. Analytic databases (a.k.a. OLAP- on Line Analytical Processing) are primarily static, read-only databases which store archived, historical data used for analysis.

Requirement Analysis

**Chapter 2**

**REQUIREMENT ANALYSIS**

**3.1 System Analysis:-**

Systems analysis is the study of systems sets of interacting entities, including computer systems. This field is closely related to operations research. It is also an explicit formal inquiry carried out to help someone, referred to as the decision maker, identify a better course of action and make a better decision than he might have otherwise made. Employment utilizing systems analysis includes systems analyst, business analyst, manufacturing engineers, enterprise and architect. Systems analysis is the process of examining a business situation for the purpose of developing a system solution to a problem or devising improvements to such a situation. Before the development of any system can begin, a project proposal is prepared by the users of the potential system and/or by system analysts and submitted to an appropriate managerial structure within the organization. So the objective of the system analysis phase is the establishment of the requirements for the system to be acquired, developed and installed.

It is the detailed study of the various operations performed by the system and their

relationships within and outside the system. Analysis is the process of breaking something into its parts so that the whole may be understood. System analysis is concerned with becoming aware of the problem, identifying the relevant and most decisional variables, analyzing and synthesizing the various factors and determining an optional or at least a satisfactory solution. During this a problem is identified, alternate system solutions are studied and recommendations are made about committing the resources used to the system.

**3.2 Compare with Existing System:-**

File Rakshak Protects file and folder from system. Some features of this application are password-protect, hide, or restrict access to folders, files, and programs by using cryptography and steganography. It can also restrict access to the Control Panel commands, prevent downloading of programs from the Internet, control user access to the external drives, and more.

**File Rakshak** is a powerful computer security program that you can use to control access to files, folders, and other Windows resources. You can use File Rakshak to [lock your personal files and folders with passwords](http://www.winability.com/folder-guard-lock-with-password/), to stop other users from peeking into your records. You can even completely [hide your private folders](http://www.winability.com/folder-guard-hide-folder/) from virtually all applications, and such folders would remain invisible until you enter a valid password. You can also protect sensitive system

Requirement Analysis

files from modification or destruction, disable access to the removable drives, [restrict access to Control Panel](http://www.winability.com/folder-guard-restrict-control-panel/), and more.

Its provides encryption & decryption and hiding the data in image (Steganography),

These facilities are not available in existing softwares.

**3.3 Features of File Rakshak**

## 1-File rakshak runs on a wide range of Windows platforms.

## From Windows XP to Windows 8 and later, with all service packs and hotfixes, all are supported by File Rakshak. File Rakshak sports one of the most intuitive user interfaces, that makes it easy to use for both novice users and computer professionals.

## 2- File rakshak can be used with other disk tools.

You can designate your anti-virus and other disk maintenance tools to be "trusted" programs, to make them able to work with your protected disks without restrictions.

## 3- File rakshak protection is extensible.

You can fine-tune the access rules to the files and folders of your computer by creating appropriate [filters](http://www.winability.com/folderguard/users-guide.htm?page=users-guide_filters.htm).

## 4- File rakshak will save you hours of learning time.

You don't have to learn how to use the Group Policies, user groups, Access Control Lists, privileges, and other built-in security features of Windows, because Folder Guard does the hard work for you: its visual and intuitive user interface lets you manage the restrictions with ease.

* Pre-boot password authentication ensures only authorized individuals have access to data.
* Easy central deployment, administration, and management features
* Supports multiple users and administrators on individual machines.
* Key sharing mechanisms that allow users to share access to files securely.
* The main advantage of steganography over other methods such as cryptography is that, it will not arose suspicion even if the files fall in the hands of a third party. Unlike cryptographic

Requirement Analysis

messages, stenographic messages will no way attract the attention of a third party by themselves. Thus steganography has an upper hand over cryptography as it involves both encryption and obscurity.

**3.4 Hardware Specification:-**

The hardware specifications vary from time to time. For heavier applications the hardware specifications will be of higher demand. A hard disk drive is a non-volatile storage device which stores digitally encoded data on rapidly rotating platters with magnetic surfaces. . Some of the main advantages of using D2 shape distributions are that its concise to store, quick to compute, invariant to transforms, efficient to match, insensitive to noise, insensitive to topology, robust to degeneracies, invariant to deformations and discriminating. Strictly speaking, "drive" refers to a device distinct from its medium, such as a tape drive and its tape, or a floppy disk drive and its floppy disk Higher the hardware capability, higher will be the convenience for the developer. The main memory or RAM is the working memory. It decides the speed of the system. Hard disk is used for storing data. Processing speed determines the speed of execution of the program. HDDs record data by magnetizing ferromagnetic material directionally, to represent either a 0 or a 1 binary digit. They read the data back by detecting the magnetization of the material. A typical HDD design consists of a spindle that holds one or more flat circular disks called platters, onto which the data are recorded. The platters are spun at very high speeds. Hardware specification is shown in Table.1

**Table.1. Hardware Specifications**

|  |  |
| --- | --- |
| Main Memory | 512MB or Above |
| Hard Disk | Min 10 GB Free |
| Processor | Pentium 4 & Above |

**3.5 Software Specification:-**

Operating system preferred is windows XP. Microsoft Visual Studio 2005 is the development framework for the application which is also called the Integrated Development Environment. Windows XP is the most preferable platform rather than windows vista because xp is the most widely used operating system in the world. The hardware specifications vary from time to time. For heavier applications the hardware specifications will be of higher demand. It is the working turf for the language that we choose. Windows XP is the most preferable platform rather than other versions because xp is the most widely used operating system in the world. Microsoft visual studio 2005 is the development framework and The Microsoft .NET Framework is a software framework that can be installed on computers running Microsoft Windows operating systems. It includes a large library of coded solutions to common programming problems and a virtual machine that manages the execution of programs written specifically for the framework. The .NET Framework is a key Microsoft offering and is intended to be used by most new applications created for the Windows platform.

**3.6 Software Tools and Technologies Used**

**3.6.1 TOOLS:**

* **Visual Studio 2010**

****

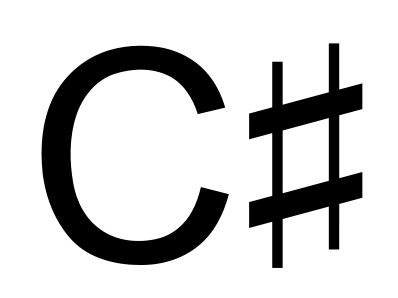
* + Describes new features in the Visual Basic language and Code Editor. The features include implicit line continuation, auto-implemented properties, collection initializers, and more.
  + Describes new features in the C# language and Code Editor. The features include the dynamic type, named and optional arguments, enhanced Office programmability, and variance.
  + Describes new and revised features in Visual C++. The features include lambda expressions, the rvalue reference declarator, and the auto, decltype, and static\_assert keywords.

Requirement Analysis

* + Describes the F# language, which is a language that supports functional programming for the .NET Framework.
* **MICROSOFT WINDOWS XP**



* + Windows XP is an [operating system](http://en.wikipedia.org/wiki/Operating_system) produced by [Microsoft](http://en.wikipedia.org/wiki/Microsoft) for use on [personal computers](http://en.wikipedia.org/wiki/Personal_computer), including home and business desktops, [laptops](http://en.wikipedia.org/wiki/Laptops), and [media centers](http://en.wikipedia.org/wiki/Home_theater_PC). It was first released in August 2001, and is the most popular version of [Windows](http://en.wikipedia.org/wiki/Microsoft_Windows), based on installed user base. The name "XP" is short for "eXPerience."
  + Windows XP is the successor to both [Windows 2000](http://en.wikipedia.org/wiki/Windows_2000) and [Windows Me](http://en.wikipedia.org/wiki/Windows_Me), and is the first consumer-oriented operating system produced by Microsoft to be built on the [Windows NT kernel](http://en.wikipedia.org/wiki/Windows_NT_kernel) and [architecture](http://en.wikipedia.org/wiki/Architecture_of_the_Windows_NT_operating_system_line).

**3.6.2 TECHNOLOGIES:**

* + C# language is intended to be a simple, modern, general-purpose, object-oriented programming language.
  + The language, and implementations thereof, should provide support for software engineering principles such as [strong type](http://en.wikipedia.org/wiki/Strong_type) checking, array bound

Checking , detection of attempts to use uninitialized variables, and automatic [garbage collection](http://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). Software robustness, durability, and programmer productivity are important.

Requirement Analysis

* + The language is intended for use in developing [software components](http://en.wikipedia.org/wiki/Software_components) suitable for deployment in distributed environments.
  + Source code portability is very important, as is programmer portability, especially for those programmers already familiar with C and C++.
  + C# is intended to be suitable for writing applications for both hosted and [embedded systems](http://en.wikipedia.org/wiki/Embedded_system), ranging from the very large that use sophisticated [operating systems](http://en.wikipedia.org/wiki/Operating_system), down to the very small having dedicated functions.
  + Although C# applications are intended to be economical with regard to memory and [processing power](http://en.wikipedia.org/wiki/Processing_power) requirements, the language was not intended to compete directly on performance and size with C or assembly language.

System Design

**Chapter 3**

**SYSTEM DESIGN**

System design is transition from a user oriented document to programmers or database personnel. The design is a solution, how to approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough. The database tables are designed by analyzing functions involved in the system and format of the fields is also designed. The fields in the database tables should define their role in the system. The unnecessary fields should be avoided because it affects the storage areas of the system. Then in the input and output screen design, the design should be made user friendly. The menu should be precise and compact.

**4.1 Software Design**

In designing the software following principles are followed:

**Modularity and partitioning**: software is designed such that, each system should consists of hierarchy of modules and serve to partition into separate function.

**Coupling:** modules should have little dependence on other modules of a system.

**Cohesion:** modules should carry out in a single processing function.

**Shared use:** avoid duplication by allowing a single module is called by other that need the function it provides

**4.2 Data Flow Diagram (DFD)**

* DFD are commonly used during problem analysis.
* DFD shows the flow of data through a system.
* DFD aims to capture the transformation that take place within the system to the input data so that eventually the output data is produced.
* DFD does not represent procedural information.
* DFD is a logical design of the flow of information between the modules.
* DFD shows the interaction of one module with the other module.

System Design

There are several level of DFD are used to describe the flow of information.

**0 Level DFD for File Rakshak:**

Uuuuser

Lock files

Hidden information in image

Encrypted data

user

Authenticate

user

**1 Level DFD for File Rakshak:**

System Design

**1 level DFD for Encryption:**

DATA

Encrypted

output

**1 level DFD for Stegnography:**

Hidden data in image

Data

**4.3 Code Editor:-**

Visual Studio, like any other IDE, includes a code editor that supports syntax highlighting and code completion using IntelliSense for not only variables, functions and methods but also language constructs like loops and queries. IntelliSense is supported for the included languages, as well as for XML and for Cascading Style Sheets and JavaScript when developing web sites and web applications. Auto complete suggestions are popped up in a modeless list box, overlaid on top of the code editor. In Visual Studio 2008 onwards, it can be made temporarily semi-transparent to see the code obstructed by it. The code editor is used for all supported languages. The Visual Studio code editor also supports setting bookmarks in code for quick navigation. Other navigational aids include collapsing code blocks and incremental search, in addition to normal text search and regex search. The code editor also includes a multi-item clipboard and a task list. The code editor supports code snippets, which

System Design

are saved templates for repetitive code and can be inserted into code and customized for the project being worked on. Visual Studio features background compilation also called incremental compilation. As code is being written, Visual Studio compiles it in the background in order to provide feedback about syntax and compilation errors, which are flagged with a red wavy underline. Warnings are marked with a green underline. Background compilation does not generate executable code, since it requires a different compiler than the one used to generate executable code. Background compilation was initially introduced with Microsoft Visual Basic but has now been expanded for all included languages.

**4.4 Debugger:-**

Visual Studio includes a debugger that works both as a source-level debugger as well as machine-level debugger. It works with both managed code as well as native code and can be used for debugging applications written in any language supported by Visual Studio. In addition,it can also attach to running processes and monitor and debug those processes. If source code for the running process is available, it displays the code as it is being run. If source code is not available, it can show the disassembly. The Visual Studio debugger can also create memory dumps as well as load them later for debugging. Multi-threaded programs are also supported. The debugger can be configured to be launched. When an application running outside, debugger crashes and the system reports an error in the internal system registry. The debugger allows setting breakpoints (which allow execution to be stopped temporarily at a certain position) and watches (which monitor the values of variables as the execution progresses). Breakpoints can be conditional, meaning they get triggered when the condition is met. Code can be stepped over, i.e., run one line of source code at a time. During coding, the Visual Studio debugger lets certain functions be invoked manually from the Immediate tool window. The parameters to the method are supplied at the immediate window. The DFD is also known as bubble chart. It is simple graphical formalism that can be used to represent a system in terms of data to the system, various processing are carried out on these data and the output data is generated by the system.

Coding

**Chapter 4**

**CODING**

**Program:For Authentication :**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.IO;

namespace File\_Encrypter\_Decrypyer

{

public partial class Form1 : Form

{

object data = new object();

string uname, upass;

FileStream outs;

StreamReader sr;

StreamWriter sw;

int panelwidthsize = 1;

public Form1()

{

InitializeComponent();

}

private void panel1\_DragEnter(object sender, DragEventArgs e)

{

e.Effect = DragDropEffects.All;

}

private void panel1\_DragDrop(object sender, DragEventArgs e)

{

string[] files = (string[])e.Data.GetData(DataFormats.FileDrop,true);

foreach (string file in files)

MessageBox.Show(file);

}

private void timer1\_Tick(object sender, EventArgs e)

Coding

{

if (panelwidthsize == 516)

{

timer1.Enabled = false;

menuStrip1.BringToFront();

}

panel\_main.Size = new Size(panelwidthsize, 354);

panelwidthsize += 5;

}

private void encripToolStripMenuItem\_Click\_1(object sender, EventArgs e)

{

frmEncDec fr = new frmEncDec();

fr.Show();

}

private void stagnographToolStripMenuItem1\_Click(object sender, EventArgs e)

{

stagnograph sg = new stagnograph();

sg.Show();

}

private void Form1\_Load(object sender, EventArgs e)

{

outs = new FileStream("f:/pswd.txt", FileMode.OpenOrCreate);

sr = new StreamReader(outs);

uname = sr.ReadLine();

upass = sr.ReadLine();

if (uname == null | upass == null)

{

panel\_auth.Visible = true;

}

else

{

pan\_after\_auth.Visible = true;

}

}

private void but\_authentic\_Click\_1(object sender, EventArgs e)

Coding

{

if (textBox1.Text.Trim() != "" & textBox2.Text.Trim() != "")

{

sw = new StreamWriter(outs);

sw.WriteLine(textBox1.Text);

sw.WriteLine(textBox2.Text);

sw.Close();

MessageBox.Show("Restart The Application");

this.Close();

}

else

{

MessageBox.Show("Enter Correct Value To Be Saved");

textBox2.Text = "";

textBox1.Text = "";

}

pan\_after\_auth.Visible = true;

panel\_auth.Visible = false;

}

private void but\_after\_auth\_Click(object sender, EventArgs e)

{

if (textBox3.Text == upass)

{

MessageBox.Show("Done");

panel\_auth.Visible = false;

pan\_after\_auth.Visible = false;

panel\_main.Visible = true;

timer1.Enabled = true;

}

else

{

MessageBox.Show("Invalid Password");

textBox3.Text = "";

}

}

}

}

Coding

**Program for Encryption & Decryption By using RSA ALGO**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Runtime.InteropServices;

using System.Security;

using System.Security.Cryptography;

using System.IO;

using System.Collections;

namespace File\_Encrypter\_Decrypyer

{

public partial class frmEncDec : Form

{

private static int \_keyLength = 0;

private String \_key = null;

private String \_cryptedText = "";

private static Boolean \_flag = false;

private String \_randomKey = "";

private static String APPLICATION\_NAME = "Text Encrypter Decrypter";

public frmEncDec()

{

InitializeComponent();

}

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

{

btnEncrypt.Enabled = false;

btnDecrypt.Enabled = false;

\_flag = true; // by default RC4 encryption is chosen

}

Coding

public String ReadFullFileData(string fileName)

{

TextReader tr = null;

try

{

tr = File.OpenText(fileName);

if (tr != null)

return tr.ReadToEnd();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, APPLICATION\_NAME, MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally

{

tr.Close();

}

return null;

}

public void OpenFile(String title)

{

OpenFileDialog fileDialog = new OpenFileDialog();

fileDialog.Title = title;

fileDialog.Filter = "TEXT Files(\*.txt)|\*.txt";

if (fileDialog.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

txtPath.Text = fileDialog.FileName;

string dataToEncrypt = ReadFullFileData(fileDialog.FileName);

rTxtBox.Text = dataToEncrypt;

chkData.Checked = true;

}

else

{

return;

}

}

Coding

public void SaveFile(String title)

{

TextWriter tw = null;

try

{

SaveFileDialog saveFileDialog = new SaveFileDialog();

saveFileDialog.Title = title;

saveFileDialog.Filter = "TEXT Files(\*.txt)|\*.txt";

if (saveFileDialog.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

tw = File.CreateText(saveFileDialog.FileName);

tw.WriteLine(rTxtBox.Text);

}

else

{

return;

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, APPLICATION\_NAME, MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally

{

tw.Close();

}

}

public static String RandomKeyString() // random key generator

{

StringBuilder builder = new StringBuilder();

Random r = new Random();

char ch;

int size = r.Next(10, 20);

for (int i = 0; i < size; i++)

{

ch = Convert.ToChar(r.Next(65, 122));

builder.Append(ch);

}

Coding

return builder.ToString();

}

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

{

OpenFile("Open A File To Encrypt");

btnEncrypt.Enabled = true;

btnDecrypt.Enabled = true;

}

public void RC4Encryption()

{

RC4EncryptionDecryption rc4Enc = new RC4EncryptionDecryption();

rc4Enc.EncryptionKey = \_randomKey;

lblKey.Text = "RC4 Encryption Key";

rTxtKey.Text = \_randomKey;

rc4Enc.InClearText = rTxtBox.Text;

rc4Enc.RC4Encryption();

this.\_cryptedText = rc4Enc.CryptedText;

rTxtBox.Clear();

rTxtBox.Text = this.\_cryptedText;

}

public void RC4Decryption()

{

RC4EncryptionDecryption rc4Enc = new RC4EncryptionDecryption();

rc4Enc.EncryptionKey = \_randomKey;

lblKey.Text = "RC4 Decryption Key";

rTxtKey.Text = \_randomKey;

rc4Enc.CryptedText = rTxtBox.Text;

rc4Enc.RC4Decryption();

rTxtBox.Clear();

rTxtBox.Text = rc4Enc.InClearText;

}

public void RSAEncryption()

{

\_keyLength = Convert.ToInt32(numericUpDown.Value.ToString());

Coding

string encryptedString = RSAEncryptionDecryption.RSAEncryption(rTxtBox.Text, \_keyLength, \_key);

lblKey.Text = "RSA Encryption Key";

rTxtKey.Text = \_key;

rTxtBox.Clear();

rTxtBox.Text = encryptedString;

chkData.Checked = true;

}

public void RSADecryption()

{

string decryptedString = RSAEncryptionDecryption.RSADecryption(rTxtBox.Text, \_keyLength, \_key);

rTxtBox.Clear();

rTxtBox.Text = decryptedString;

}

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

{

try

{

if (String.IsNullOrEmpty(rTxtKey.Text))

{

MessageBox.Show("You must generate key first to encrypt or decrypt text.", APPLICATION\_NAME, MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

else if (!String.IsNullOrEmpty(rTxtKey.Text))

{

if (rTxtBox.Text != null && !chkData.Checked) // data is from file

{

if (\_flag) // user select RC4 encrytion from Encryption Menu

{

RC4Encryption();

}

else // user select RSA encrytion from Encryption Menu

{

RSAEncryption();

}

Coding

DialogResult result = MessageBox.Show("Want to save encrypted text to file?", APPLICATION\_NAME, MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (result == System.Windows.Forms.DialogResult.Yes)

{

// then save the encrypted text to file

SaveFile("Save a encrypted text to file");

}

else

{

return;

}

}

else if (chkData.Checked && rTxtBox.Text != null)

{

// user write direct to rich text box, not open a file from browse button

// now first convert text to encrypted text, then prompt user to save that text or not

if (\_flag)

{

RC4Encryption();

}

else

{

RSAEncryption();

}

DialogResult result = MessageBox.Show("Want to save encrypted text to file?", APPLICATION\_NAME, MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (result == System.Windows.Forms.DialogResult.Yes)

{

// then save the encrypted text to file

SaveFile("Save a encrypted text to file");

}

else

{

return;

}

}

}

Coding

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, APPLICATION\_NAME, MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

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

{

try

{

if (rTxtBox.Text != null && !chkData.Checked)

{

// means data that is to be decrypted is from file, so first open file by clicking button browse

OpenFile("Open A Encrypted File To Decrypt");

if (\_flag)

{

RC4Decryption();

}

else

{

RSADecryption();

}

DialogResult result = MessageBox.Show("Want to save decrypted text to file?", APPLICATION\_NAME, MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (result == System.Windows.Forms.DialogResult.Yes)

{

// then save the encrypted text to file

SaveFile("Save a decrypted text to file");

}

else

{

return;

}

}

Coding

else if (chkData.Checked && rTxtBox.Text != null)

{

// means data that is to be decrypted is written by user directly to rich textbox,

// so simple decrypt it and prompt user for its saving

if (\_flag)

{

RC4Decryption();

}

else

{

RSADecryption();

}

DialogResult result = MessageBox.Show("Want to save decrypted text to file?", APPLICATION\_NAME, MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (result == System.Windows.Forms.DialogResult.Yes)

{

// then save the encrypted text to file

SaveFile("Save a decrypted text to file");

}

else

{

return;

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, APPLICATION\_NAME, MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

private void chkData\_CheckedChanged(object sender, EventArgs e)

{

if (chkData.Checked)

{

btnEncrypt.Enabled = true;

btnDecrypt.Enabled = true;

}

Coding

else

{

btnEncrypt.Enabled = false;

btnDecrypt.Enabled = false;

}

}

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

{

Application.Exit();

}

private void rC4ToolStripMenuItem\_Click(object sender, EventArgs e)

{

\_flag = true;

}

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

{

\_flag = false;

}

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

{

rTxtBox.Clear();

rTxtKey.Clear();

}

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

{

Application.Exit();

}

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

{

if (\_flag)

{

lblKey.Text = "RC4 Encryption Key";

\_randomKey = RandomKeyString(); // generating random key

rTxtKey.Text = \_randomKey;

}

Coding

else

{

\_key = RSAEncryptionDecryption.RSAGenerateKey(\_keyLength);

string bitStrengthString = \_key.Substring(0, \_key.IndexOf("</BitStrength>") + 14);

\_key = \_key.Replace(bitStrengthString, "");

lblKey.Text = "RSA Encryption Key";

rTxtKey.Text = \_key;

}

}

}

}

**Program for Encryption & Decryption By using RC4 ALGO**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Runtime.InteropServices;

using System.Security;

using System.Security.Cryptography;

using System.IO;

using System.Collections;

namespace File\_Encrypter\_Decrypyer

{

public partial class frmEncDec : Form

{

private static int \_keyLength = 0;

private String \_key = null;

private String \_cryptedText = "";

private static Boolean \_flag = false;

private String \_randomKey = "";

private static String APPLICATION\_NAME = "Text Encrypter Decrypter";

Coding

public frmEncDec()

{

InitializeComponent();

}

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

{

btnEncrypt.Enabled = false;

btnDecrypt.Enabled = false;

\_flag = true; // by default RC4 encryption is chosen

}

public String ReadFullFileData(string fileName)

{

TextReader tr = null;

try

{

tr = File.OpenText(fileName);

if (tr != null)

return tr.ReadToEnd();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, APPLICATION\_NAME, MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally

{

tr.Close();

}

return null;

}

public void OpenFile(String title)

{

OpenFileDialog fileDialog = new OpenFileDialog();

fileDialog.Title = title;

Coding

fileDialog.Filter = "TEXT Files(\*.txt)|\*.txt";

if (fileDialog.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

txtPath.Text = fileDialog.FileName;

string dataToEncrypt = ReadFullFileData(fileDialog.FileName);

rTxtBox.Text = dataToEncrypt;

chkData.Checked = true;

}

else

{

return;

}

}

public void SaveFile(String title)

{

TextWriter tw = null;

try

{

SaveFileDialog saveFileDialog = new SaveFileDialog();

saveFileDialog.Title = title;

saveFileDialog.Filter = "TEXT Files(\*.txt)|\*.txt";

if (saveFileDialog.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

tw = File.CreateText(saveFileDialog.FileName);

tw.WriteLine(rTxtBox.Text);

}

else

{

return;

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, APPLICATION\_NAME, MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally

{

Coding

tw.Close();

}

}

public static String RandomKeyString() // random key generator

{

StringBuilder builder = new StringBuilder();

Random r = new Random();

char ch;

int size = r.Next(10, 20);

for (int i = 0; i < size; i++)

{

ch = Convert.ToChar(r.Next(65, 122));

builder.Append(ch);

}

return builder.ToString();

}

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

{

OpenFile("Open A File To Encrypt");

btnEncrypt.Enabled = true;

btnDecrypt.Enabled = true;

}

public void RC4Encryption()

Coding

{

RC4EncryptionDecryption rc4Enc = new RC4EncryptionDecryption();

rc4Enc.EncryptionKey = \_randomKey;

lblKey.Text = "RC4 Encryption Key";

rTxtKey.Text = \_randomKey;

rc4Enc.InClearText = rTxtBox.Text;

rc4Enc.RC4Encryption();

this.\_cryptedText = rc4Enc.CryptedText;

rTxtBox.Clear();

rTxtBox.Text = this.\_cryptedText;

}

Coding

public void RC4Decryption()

{

RC4EncryptionDecryption rc4Enc = new RC4EncryptionDecryption();

rc4Enc.EncryptionKey = \_randomKey;

lblKey.Text = "RC4 Decryption Key";

rTxtKey.Text = \_randomKey;

rc4Enc.CryptedText = rTxtBox.Text;

rc4Enc.RC4Decryption();

rTxtBox.Clear();

rTxtBox.Text = rc4Enc.InClearText;

}

public void RSAEncryption()

{

\_keyLength = Convert.ToInt32(numericUpDown.Value.ToString());

string encryptedString = RSAEncryptionDecryption.RSAEncryption(rTxtBox.Text, \_keyLength, \_key);

lblKey.Text = "RSA Encryption Key";

rTxtKey.Text = \_key;

rTxtBox.Clear();

rTxtBox.Text = encryptedString;

chkData.Checked = true;

}

public void RSADecryption()

{

string decryptedString = RSAEncryptionDecryption.RSADecryption(rTxtBox.Text, \_keyLength, \_key);

rTxtBox.Clear();

rTxtBox.Text = decryptedString;

}

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

{

try

{

if (String.IsNullOrEmpty(rTxtKey.Text))

{

Coding

MessageBox.Show("You must generate key first to encrypt or decrypt text.", APPLICATION\_NAME, MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

else if (!String.IsNullOrEmpty(rTxtKey.Text))

{

if (rTxtBox.Text != null && !chkData.Checked) // data is from file

{

if (\_flag) // user select RC4 encrytion from Encryption Menu

{

RC4Encryption();

}

else // user select RSA encrytion from Encryption Menu

{

RSAEncryption();

}

DialogResult result = MessageBox.Show("Want to save encrypted text to file?", APPLICATION\_NAME, MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (result == System.Windows.Forms.DialogResult.Yes)

{

// then save the encrypted text to file

SaveFile("Save a encrypted text to file");

}

else

{

return;

}

}

else if (chkData.Checked && rTxtBox.Text != null)

{

// user write direct to rich text box, not open a file from browse button

// now first convert text to encrypted text, then prompt user to save that text or not

if (\_flag)

{

RC4Encryption();

}

else

{

Coding

RSAEncryption();

}

DialogResult result = MessageBox.Show("Want to save encrypted text to file?", APPLICATION\_NAME, MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (result == System.Windows.Forms.DialogResult.Yes)

{

// then save the encrypted text to file

SaveFile("Save a encrypted text to file");

}

else

{

return;

}

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, APPLICATION\_NAME, MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

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

{

try

{

if (rTxtBox.Text != null && !chkData.Checked)

{

// means data that is to be decrypted is from file, so first open file by clicking button browse

OpenFile("Open A Encrypted File To Decrypt");

if (\_flag)

{

RC4Decryption();

}

Else

Coding

{

RSADecryption();

}

DialogResult result = MessageBox.Show("Want to save decrypted text to file?", APPLICATION\_NAME, MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (result == System.Windows.Forms.DialogResult.Yes)

{

// then save the encrypted text to file

SaveFile("Save a decrypted text to file");

}

else

{

return;

}

}

else if (chkData.Checked && rTxtBox.Text != null)

{

// means data that is to be decrypted is written by user directly to rich textbox,

// so simple decrypt it and prompt user for its saving

if (\_flag)

{

RC4Decryption();

}

else

{

RSADecryption();

}

DialogResult result = MessageBox.Show("Want to save decrypted text to file?", APPLICATION\_NAME, MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (result == System.Windows.Forms.DialogResult.Yes)

{

// then save the encrypted text to file

SaveFile("Save a decrypted text to file");

}

else

{

return;

Coding

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, APPLICATION\_NAME, MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

private void chkData\_CheckedChanged(object sender, EventArgs e)

{

if (chkData.Checked)

{

btnEncrypt.Enabled = true;

btnDecrypt.Enabled = true;

}

else

{

btnEncrypt.Enabled = false;

btnDecrypt.Enabled = false;

}

}

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

{

Application.Exit();

}

private void rC4ToolStripMenuItem\_Click(object sender, EventArgs e)

{

\_flag = true;

}

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

{

\_flag = false;

}

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

Coding

{

rTxtBox.Clear();

rTxtKey.Clear();

}

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

{

Application.Exit();

}

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

{

if (\_flag)

{

lblKey.Text = "RC4 Encryption Key";

\_randomKey = RandomKeyString(); // generating random key

rTxtKey.Text = \_randomKey;

}

else

{

\_key = RSAEncryptionDecryption.RSAGenerateKey(\_keyLength);

string bitStrengthString = \_key.Substring(0, \_key.IndexOf("</BitStrength>") + 14);

\_key = \_key.Replace(bitStrengthString, "");

lblKey.Text = "RSA Encryption Key";

rTxtKey.Text = \_key;

}

}

}

}

**Program For Stegnography**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

Coding

using System.IO;

namespace File\_Encrypter\_Decrypyer

{

public partial class stagnograph : Form

{

public stagnograph()

{

InitializeComponent();

}

string loadedTrueImagePath, loadedFilePath, saveToImage,DLoadImagePath,DSaveFilePath;

int height, width;

long fileSize, fileNameSize;

Image loadedTrueImage, DecryptedImage ,AfterEncryption;

Bitmap loadedTrueBitmap, DecryptedBitmap;

Rectangle previewImage = new Rectangle(20,160,490,470);

bool canPaint = false, EncriptionDone = false;

byte[] fileContainer;

private void EnImageBrowse\_btn\_Click(object sender, EventArgs e)

{

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

loadedTrueImagePath = openFileDialog1.FileName;

EnImage\_tbx.Text = loadedTrueImagePath;

loadedTrueImage = Image.FromFile(loadedTrueImagePath);

height = loadedTrueImage.Height;

width = loadedTrueImage.Width;

loadedTrueBitmap = new Bitmap(loadedTrueImage);

FileInfo imginf = new FileInfo(loadedTrueImagePath);

float fs = (float)imginf.Length / 1024;

ImageSize\_lbl.Text = smalldecimal(fs.ToString(), 2) + " KB";

ImageHeight\_lbl.Text = loadedTrueImage.Height.ToString() + " Pixel";

ImageWidth\_lbl.Text = loadedTrueImage.Width.ToString() + " Pixel";

double cansave = (8.0 \* ((height \* (width / 3) \* 3) / 3 - 1)) / 1024;

CanSave\_lbl.Text = smalldecimal(cansave.ToString(), 2) + " KB";

Coding

canPaint = true;

this.Invalidate();

}

}

private string smalldecimal(string inp, int dec)

{

int i;

for (i = inp.Length - 1; i > 0; i--)

if (inp[i] == '.')

break;

try

{

return inp.Substring(0, i + dec + 1);

}

catch

{

return inp;

}

}

private void EnFileBrowse\_btn\_Click(object sender, EventArgs e)

{

if (openFileDialog2.ShowDialog() == DialogResult.OK)

{

loadedFilePath = openFileDialog2.FileName;

EnFile\_tbx.Text = loadedFilePath;

FileInfo finfo = new FileInfo(loadedFilePath);

fileSize = finfo.Length;

fileNameSize = justFName(loadedFilePath).Length;

}

}

private void Encrypt\_btn\_Click(object sender, EventArgs e)

{

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

saveToImage = saveFileDialog1.FileName;

}

else

Coding

return;

if (EnImage\_tbx.Text == String.Empty || EnFile\_tbx.Text == String.Empty)

{

MessageBox.Show("Encrypton information is incomplete!\nPlease complete them frist.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

if (8\*((height \* (width/3)\*3)/3 - 1) < fileSize + fileNameSize)

{

MessageBox.Show("File size is too large!\nPlease use a larger image to hide this file.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

fileContainer = File.ReadAllBytes(loadedFilePath);

EncryptLayer();

}

private void EncryptLayer()

{

toolStripStatusLabel1.Text ="Encrypting... Please wait";

Application.DoEvents();

long FSize = fileSize;

Bitmap changedBitmap = EncryptLayer(8, loadedTrueBitmap, 0, (height \* (width/3)\*3) / 3 - fileNameSize - 1, true);

FSize -= (height \* (width / 3) \* 3) / 3 - fileNameSize - 1;

if(FSize > 0)

{

for (int i = 7; i >= 0 && FSize > 0; i--)

{

changedBitmap = EncryptLayer(i, changedBitmap, (((8 - i) \* height \* (width / 3) \* 3) / 3 - fileNameSize - (8 - i)), (((9 - i) \* height \* (width / 3) \* 3) / 3 - fileNameSize - (9 - i)), false);

FSize -= (height \* (width / 3) \* 3) / 3 - 1;

}

}

changedBitmap.Save(saveToImage);

toolStripStatusLabel1.Text = "Encrypted image has been successfully saved.";

EncriptionDone = true;

AfterEncryption = Image.FromFile(saveToImage);

this.Invalidate();

Coding

}

private Bitmap EncryptLayer(int layer, Bitmap inputBitmap, long startPosition, long endPosition, bool writeFileName)

{

Bitmap outputBitmap = inputBitmap;

layer--;

int i = 0, j = 0;

long FNSize = 0;

bool[] t = new bool[8];

bool[] rb = new bool[8];

bool[] gb = new bool[8];

bool[] bb = new bool[8];

Color pixel = new Color();

byte r, g, b;

if (writeFileName)

{

FNSize = fileNameSize;

string fileName = justFName(loadedFilePath);

//write fileName:

for (i = 0; i < height && i \* (height / 3) < fileNameSize; i++)

for (j = 0; j < (width / 3) \* 3 && i \* (height / 3) + (j / 3) < fileNameSize; j++)

{

byte2bool((byte)fileName[i \* (height / 3) + j / 3], ref t);

pixel = inputBitmap.GetPixel(j, i);

r = pixel.R;

g = pixel.G;

b = pixel.B;

byte2bool(r, ref rb);

byte2bool(g, ref gb);

byte2bool(b, ref bb);

if (j % 3 == 0)

{

rb[7] = t[0];

gb[7] = t[1];

bb[7] = t[2];

}

Coding

else if (j % 3 == 1)

{

rb[7] = t[3];

gb[7] = t[4];

bb[7] = t[5];

}

else

{

rb[7] = t[6];

gb[7] = t[7];

}

Color result = Color.FromArgb((int)bool2byte(rb), (int)bool2byte(gb), (int)bool2byte(bb));

outputBitmap.SetPixel(j, i, result);

}

i--;

}

//write file (after file name):

int tempj = j;

for (; i < height && i \* (height / 3) < endPosition - startPosition + FNSize && startPosition + i \* (height / 3) < fileSize + FNSize; i++)

for (j = 0; j < (width / 3) \* 3 && i \* (height / 3) + (j / 3) < endPosition - startPosition + FNSize && startPosition + i \* (height / 3) + (j / 3) < fileSize + FNSize; j++)

{

if (tempj != 0)

{

j = tempj;

tempj = 0;

}

byte2bool((byte)fileContainer[startPosition + i \* (height / 3) + j / 3 - FNSize], ref t);

pixel = inputBitmap.GetPixel(j, i);

r = pixel.R;

g = pixel.G;

b = pixel.B;

byte2bool(r, ref rb);

byte2bool(g, ref gb);

byte2bool(b, ref bb);

if (j % 3 == 0)

Coding

{

rb[layer] = t[0];

gb[layer] = t[1];

bb[layer] = t[2];

}

else if (j % 3 == 1)

{

rb[layer] = t[3];

gb[layer] = t[4];

bb[layer] = t[5];

}

else

{

rb[layer] = t[6];

gb[layer] = t[7];

}

Color result = Color.FromArgb((int)bool2byte(rb), (int)bool2byte(gb), (int)bool2byte(bb));

outputBitmap.SetPixel(j, i, result);

}

long tempFS = fileSize, tempFNS = fileNameSize;

r = (byte)(tempFS % 100);

tempFS /= 100;

g = (byte)(tempFS % 100);

tempFS /= 100;

b = (byte)(tempFS % 100);

Color flenColor = Color.FromArgb(r,g,b);

outputBitmap.SetPixel(width - 1, height - 1, flenColor);

r = (byte)(tempFNS % 100);

tempFNS /= 100;

g = (byte)(tempFNS % 100);

tempFNS /= 100;

b = (byte)(tempFNS % 100);

Color fnlenColor = Color.FromArgb(r,g,b);

outputBitmap.SetPixel(width - 2, height - 1, fnlenColor);

return outputBitmap;

}

Coding

private void DecryptLayer()

{

toolStripStatusLabel1.Text = "Decrypting... Please wait";

Application.DoEvents();

int i, j = 0;

bool[] t = new bool[8];

bool[] rb = new bool[8];

bool[] gb = new bool[8];

bool[] bb = new bool[8];

Color pixel = new Color();

byte r, g, b;

pixel = DecryptedBitmap.GetPixel(width - 1, height - 1);

long fSize = pixel.R + pixel.G \* 100 + pixel.B \* 10000;

pixel = DecryptedBitmap.GetPixel(width - 2, height - 1);

long fNameSize = pixel.R + pixel.G \* 100 + pixel.B \* 10000;

byte[] res = new byte[fSize];

string resFName = "";

byte temp;

//Read file name:

for (i = 0; i < height && i \* (height / 3) < fNameSize; i++)

for (j = 0; j < (width / 3) \* 3 && i \* (height / 3) + (j / 3) < fNameSize; j++)

{

pixel = DecryptedBitmap.GetPixel(j, i);

r = pixel.R;

g = pixel.G;

b = pixel.B;

byte2bool(r, ref rb);

byte2bool(g, ref gb);

byte2bool(b, ref bb);

if (j % 3 == 0)

{

t[0] = rb[7];

t[1] = gb[7];

t[2] = bb[7];

}

else if (j % 3 == 1)

{

t[3] = rb[7];

t[4] = gb[7];

Coding

t[5] = bb[7];

}

else

{

t[6] = rb[7];

t[7] = gb[7];

temp = bool2byte(t);

resFName += (char)temp;

}

}

//Read file on layer 8 (after file name):

int tempj = j;

i--;

for (; i < height && i \* (height / 3) < fSize + fNameSize; i++)

for (j = 0; j < (width / 3) \* 3 && i \* (height / 3) + (j / 3) < (height \* (width / 3) \* 3) / 3 - 1 && i \* (height / 3) + (j / 3) < fSize + fNameSize; j++)

{

if (tempj != 0)

{

j = tempj;

tempj = 0;

}

pixel = DecryptedBitmap.GetPixel(j, i);

r = pixel.R;

g = pixel.G;

b = pixel.B;

byte2bool(r, ref rb);

byte2bool(g, ref gb);

byte2bool(b, ref bb);

if (j % 3 == 0)

{

t[0] = rb[7];

t[1] = gb[7];

t[2] = bb[7];

}

else if (j % 3 == 1)

{

t[3] = rb[7];

Coding

t[4] = gb[7];

t[5] = bb[7];

}

else

{

t[6] = rb[7];

t[7] = gb[7];

temp = bool2byte(t);

res[i \* (height / 3) + j / 3 - fNameSize] = temp;

}

}

//Read file on other layers:

long readedOnL8 = (height \* (width/3)\*3) /3 - fNameSize - 1;

for (int layer = 6; layer >= 0 && readedOnL8 + (6 - layer) \* ((height \* (width / 3) \* 3) / 3 - 1) < fSize; layer--)

for (i = 0; i < height && i \* (height / 3) + readedOnL8 + (6 - layer) \* ((height \* (width / 3) \* 3) / 3 - 1) < fSize; i++)

for (j = 0; j < (width / 3) \* 3 && i \* (height / 3) + (j / 3) + readedOnL8 + (6 - layer) \* ((height \* (width / 3) \* 3) / 3 - 1) < fSize; j++)

{

pixel = DecryptedBitmap.GetPixel(j, i);

r = pixel.R;

g = pixel.G;

b = pixel.B;

byte2bool(r, ref rb);

byte2bool(g, ref gb);

byte2bool(b, ref bb);

if (j % 3 == 0)

{

t[0] = rb[layer];

t[1] = gb[layer];

t[2] = bb[layer];

}

else if (j % 3 == 1)

{

t[3] = rb[layer];

t[4] = gb[layer];

t[5] = bb[layer];

Coding

}

else

{

t[6] = rb[layer];

t[7] = gb[layer];

temp = bool2byte(t);

res[i \* (height / 3) + j / 3 + (6 - layer) \* ((height \* (width / 3) \* 3) / 3 - 1) + readedOnL8] = temp;

}

}

if (File.Exists(DSaveFilePath + "\\" + resFName))

{

MessageBox.Show("File \"" + resFName + "\" already exist please choose another path to save file", "Error",MessageBoxButtons.OK,MessageBoxIcon.Error);

return;

}

else

File.WriteAllBytes(DSaveFilePath + "\\" + resFName, res);

toolStripStatusLabel1.Text = "Decrypted file has been successfully saved.";

Application.DoEvents();

}

private void byte2bool(byte inp, ref bool[] outp)

{

if(inp>=0 && inp<=255)

for (short i = 7; i >= 0; i--)

{

if (inp % 2 == 1)

outp[i] = true;

else

outp[i] = false;

inp /= 2;

}

else

throw new Exception("Input number is illegal.");

}

private byte bool2byte(bool[] inp)

{

Coding

byte outp = 0;

for (short i = 7; i >= 0; i--)

{

if (inp[i])

outp += (byte)Math.Pow(2.0, (double)(7-i));

}

return outp;

}

private void Decrypt\_btn\_Click(object sender, EventArgs e)

{

if (DeSaveFile\_tbx.Text == String.Empty || DeLoadImage\_tbx.Text == String.Empty)

{

MessageBox.Show("Text boxes must not be empty!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

if (System.IO.File.Exists(DeLoadImage\_tbx.Text) == false)

{

MessageBox.Show("Select image file.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

DeLoadImage\_tbx.Focus();

return;

}

DecryptLayer();

}

private void DeLoadImageBrowse\_btn\_Click(object sender, EventArgs e)

{

if (openFileDialog3.ShowDialog() == DialogResult.OK)

{

DLoadImagePath = openFileDialog3.FileName;

DeLoadImage\_tbx.Text = DLoadImagePath;

Coding

DecryptedImage = Image.FromFile(DLoadImagePath);

height = DecryptedImage.Height;

width = DecryptedImage.Width;

DecryptedBitmap = new Bitmap(DecryptedImage);

FileInfo imginf = new FileInfo(DLoadImagePath);

float fs = (float)imginf.Length / 1024;

ImageSize\_lbl.Text = smalldecimal(fs.ToString(), 2) + " KB";

ImageHeight\_lbl.Text = DecryptedImage.Height.ToString() + " Pixel";

ImageWidth\_lbl.Text = DecryptedImage.Width.ToString() + " Pixel";

double cansave = (8.0 \* ((height \* (width / 3) \* 3) / 3 - 1)) / 1024;

CanSave\_lbl.Text = smalldecimal(cansave.ToString(), 2) + " KB";

canPaint = true;

this.Invalidate();

}

}

private void DeSaveFileBrowse\_btn\_Click(object sender, EventArgs e)

{

if (folderBrowserDialog1.ShowDialog() == DialogResult.OK)

{

DSaveFilePath = folderBrowserDialog1.SelectedPath;

DeSaveFile\_tbx.Text = DSaveFilePath;

}

}

private void Form1\_Paint(object sender, PaintEventArgs e)

{

if(canPaint)

try

{

if (!EncriptionDone)

e.Graphics.DrawImage(loadedTrueImage, previewImage);

else

e.Graphics.DrawImage(AfterEncryption, previewImage);

}

catch

{

Coding

e.Graphics.DrawImage(DecryptedImage, previewImage);

}

}

private string justFName(string path)

{

string output;

int i;

if (path.Length == 3) // i.e: "C:\\"

return path.Substring(0, 1);

for (i = path.Length - 1; i > 0; i--)

if (path[i] == '\\')

break;

output = path.Substring(i + 1);

return output;

}

private string justEx(string fName)

{

string output;

int i;

for (i = fName.Length - 1; i > 0; i--)

if (fName[i] == '.')

break;

output = fName.Substring(i + 1);

return output;

}

private void Close\_btn\_Click(object sender, EventArgs e)

{

this.Close();

}

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

System.Diagnostics.Process.Start("http:\\\\www.programmer2programmer.net");

}

}}

Software Testing & Implementation

**Chapter 5**

**SOFTWARE TESTING & IMPLEMENTATION**

**Testing:**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation**.**

**6.1 Testing Objectives:**

* To ensure that during operation the system will perform as per specification.
* TO make sure that system meets the user requirements during operation
* To make sure that during the operation, incorrect input, processing and output will be detected
* To see that when correct inputs are fed to the system the outputs are correct
* To verify that the controls incorporated in the same system as intended
* Testing is a process of executing a program with the intent of finding an error
* A good test case is one that has a high probability of finding an as yet undiscovered error

The software developed has been tested successfully using the following testing strategies and any errors that are encountered are corrected and again the part of the program or the procedure or function is put to testing until all the errors are removed. A successful test is one that uncovers an as yet undiscovered error.

Note that the result of the system testing will prove that the system is working correctly. It will give confidence to system designer, users of the system, prevent frustration during implementation process etc.,

**6.2 Test Case Design:**

**White box testing**

White box testing is a testing case design method that uses the control structure of the procedure design to derive test cases. All independents path in a module are exercised at least once, all logical decisions are exercised at once, execute all loops at boundaries and within their operational bounds exercise internal data structure to ensure their validity. Here the customer is given three chances to enter a valid choice out of the given menu. After which the control exits the current menu.

**Black Box Testing**

Black Box Testing attempts to find errors in following areas or categories, incorrect or missing functions, interface error, errors in data structures, performance error and initialization and termination error. Here all the input data must match the data type to become a valid entry.

The following are the different tests at various levels.

**6.3 Unit Testing**

Unit testing is essentially for the verification of the code produced during the coding phase and the goal is test the internal logic of the module/program. In the Generic code project, the unit testing is done during coding phase of data entry forms whether the functions are working properly or not. In this phase all the drivers are tested they are rightly connected or not.

**6.4 Integration Testing**

All the tested modules are combined into sub systems, which are then tested. The goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules. In the generic code integration testing is done mainly on table creation module and insertion module.

**6.5 Validation Testing**

This testing concentrates on confirming that the software is error-free in all respects. All the specified validations are verified and the software is subjected to hardcore testing. It also aims at determining the degree of deviation that exists in the software designed from the specification; they are listed out and are corrected.

**6.6 System Testing**

System testing is a critical aspect of Software Quality Assurance and represents the ultimate review of specification, design and coding. Testing is a process of executing a program with the intent of finding an error. A good test is one that has a probability of finding an as yet undiscovered error. The purpose of testing is to identify and correct bugs in the developed system. Nothing is complete without testing. Testing is the vital to the success of the system. In the code testing the logic of the developed system is tested. For this every module of the program is executed to find an error. To perform specification test, the examination of the specifications stating what the program should do and how it should perform under various conditions. Unit testing focuses first on the modules in the proposed system to locate errors. This enables to detect errors in the coding and logic that are contained within that module alone. Those resulting from the interaction between modules are initially avoided. In unit testing step each module has to be checked separately. System testing does not test the software as a whole, but rather than integration of each module in the system. The primary concern is the compatibility of individual modules. One has to find areas where modules have been designed with different specifications of data lengths, type and data element name. Testing and validation are the most important steps after the implementation of the developed system. The system testing is performed to ensure that there are no errors in the implemented system. The software must be executed several times in order to find out the errors in the different modules of the system. Validation refers to the process of using the new software for the developed system in a live environment i.e., new software inside the

Software Testing & Implementation

organization, in order to find out the errors. The validation phase reveals the failures and the bugs in the developed system. It will be come to know about the practical difficulties the

system faces when operated in the true environment. By testing the code of the implemented software, the logic of the program can be examined.

**6.7 System Implementation:**

Implementation includes all those activities that take place to convert from the old system to the new. The old system consists of manual operations, which is operated in a very different manner from the proposed new system. A proper implementation is essential to provide a reliable system to meet the requirements of the organizations. An improper installation may affect the success of the computerized system.

**6.7.1 Implementation Method**

There are several methods for handling the implementation and the consequent conversion from the old to the new computerized system. The most secure method for conversion from the old system to the new system is to run the old and new system in parallel. In this approach, a person may operate in the manual older processing system as well as start operating the new computerized system. This method offers high security, because even if there is a flaw in the computerized system, we can depend upon the manual system. However, the cost for maintaining two systems in parallel is very high. This outweighs its benefits.

Another commonly method is a direct cut over from the existing manual system to the computerized system. The change may be within a week or within a day. There are no parallel activities. However, there is no remedy in case of a problem. This strategy requires careful planning.

A working version of the system can also be implemented in one part of the organization and the personnel will be piloting the system and changes can be made as and when required. But this method is less preferable due to the loss of entirety of the system.

**6.7.2 Implementation plan:**

The implementation plan includes a description of all the activities that must occur to implement the new system and to put it into operation. It identifies the personnel responsible for the activities and prepares a time chart for implementing the system.

The implementation plan consists of the following steps.

 List all files required for implementation.

 Identify all data required to build new files during the implementation.

 List all new documents and procedures that go into the new system.

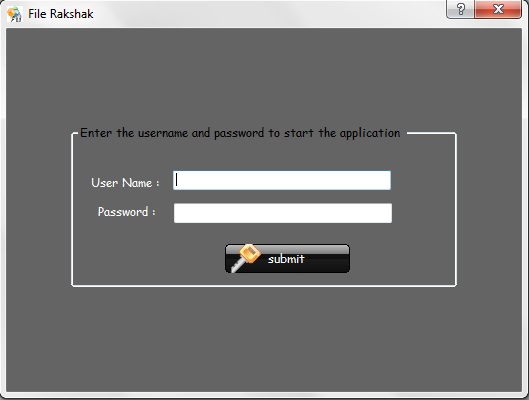
Software Testing & Implementation

The implementation plan should anticipate possible problems and must be able to deal with them. The usual problems may be missing documents; mixed data formats between current and files, errors in data translation, missing data etc.

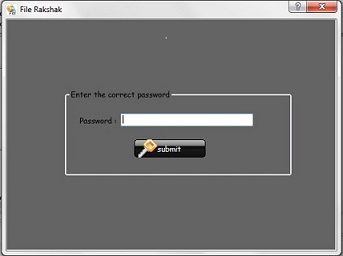
Snapshot of Project

**Snapshot of Project**

**Before Form Authentication**

****

**After Form Authentication**

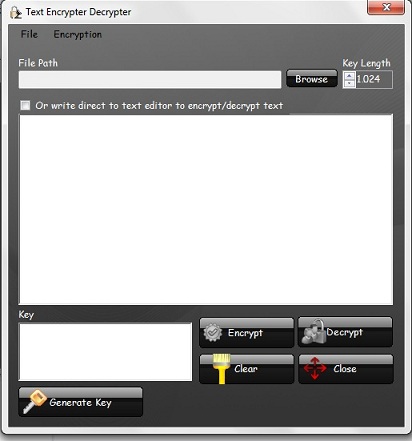
****

**Main Form**

****

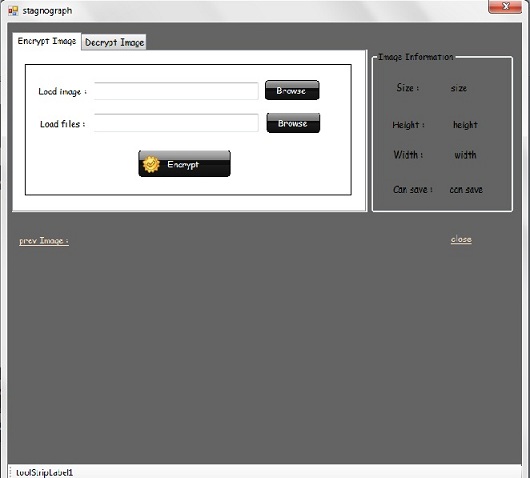
Snapshot of Project

**Form Encrypter & Decrypter**

****

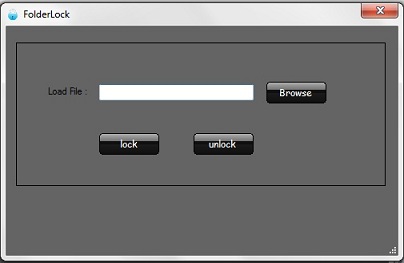
Snapshot of Project

**Form for Steganography**



Snapshot of Project

**Form for Folder Lock**



Conclusion

**CONCLUSION**

BASIS OF PROJECT

In this project we have worked on protection the files by using encryption & decryption, steganography technique. The main goal of this software is to build password-protection software that you can use to protect Windows folders with passwords, hide files and folders

Since before starting a software development, it is really necessary to gather the requirements of the client, derive new requirements, identify users of that system and create various DFD diagrams .These factors prove to be the deciding ones in estimating the budget and feasibility of a project. A developer must pay attention while the first steps of software development because these are important.

Thus the Problem Analysis, Requirement Analysis and various Use Case scenarios, DFD diagrams, Feasibility study of File Rakshak has been studied successfully.

**ADVANTAGES:**

The entire project has been developed and deployed as per the requirements and necessity to reduce the manual process and to increase the efficiency for protection of files. It is found to be bug free as per the testing standards that are implemented. The system provides an efficient way of encryption & decryption of file, hidden the data in image (Steganography) and lock the files. Efforts have been made to fully automate the existing system and enhance the features. The software is developed and designed to provide flexibility and a friendly interface. The software can also be extended in the future for development.

References

**References**

1. The complete Reference Visual C#.NET

2. Beginning C#.NET (Wrox Publication)

3. System Analysis and Design – Alias M. Awad

4. Software Engineering – Roger Pressman

5. Millman,J., ( 2006 ), **System Design, Analysis and Implementation,** Tata McGraw Hill Publication, New Delhi.

**Websites:**

1. www.microsoft.com

2. www.w3schools.com

3. www.dotnetspider.com

4. www.codeproject.com