# CHAPTER 1 INTRODUCTION

The project “Steganalysis of Images using QRC” ensures the secret transformation of information in a secured way. It can prevent the Intrusion of hackers or any other third-party users from accessing information illegally. By using various techniques, the original information can be hidden from others except the sender and receiver.

# PROBLEM DEFINITION

The existing system can be used only to hide any image or any particular form of file behind an image. There is no combined process of hiding a key behind image and converting that particular image into a QR. It exists as an individual process only. Our proposed system, tries to combine the above mentioned process of cryptography and steganography together. It ensures higher security and privacy to the user.

# OBJECTIVE OF THE PROJECT

Security and privacy are the most important factors in communication and transformation of information between the sender and receiver. Steganography helps to hide the information behind any digital media. The hidden information should not be sensible to other third party users or hackers. This approach for information hiding has become more important in many application areas. This project helps to improve the security and privacy for communication between sender and receiver.

# SIGNIFICANCE OF THE PROJECT

Steganalysis of images using QR code has the major significance which are as follows:

* Supports various user friendly devices desktop, mobile, scanners, etc.,
* Supports any operating system like Windows, Mac OS, Android, Blackberry OS etc.
* Preserves the Quality of the image and the file.
* Improved user-interface.

# OUTLINE OF THE REPORT

Steganalysis of images using QR code has a major outline report which is as follows:

* + - Manual work will be reduced
    - Improved security and privacy
    - Less chances for hacking
    - User friendly
    - Digitalize the activities
    - Managing and maintaining data becomes easier.

# CHAPTER 2

# SYSTEM ANALYSIS

System study is the process of investing a system, identifying problems, and using the information to recommend to the system. The following chapter provides a description of the existing chapter and its drawbacks. It also provides an overview of the proposed system. It helps in implementing the “Steganalysis of images with Quick Response Code”.

# EXISTING SYSTEM

The existing system involves tedious process. The process does not combine steganography and cryptography. It cannot perform both cryptography and steganography at the same time and lacks many features. It cannot support various operating systems. The existing system is not portable and cannot run on other devices.

# DRAWBACKS

The main drawbacks for the existing system are as follows:

* + - * Take more processing time.
      * There is lack of security for the data.
      * Does not support all the versions and all the operating systems.
      * More vulnerable to threats & attacks.
      * Difficulty in detecting the intruders.
      * Lack of media quality & robustness.

# PROPOSED SYSTEM

* The steganalysis technique can be implemented to encrypt any form of file and hide its key behind an image.
* This image can be converted to QRCode which can be decrypted by any user-friendly device like desktop, Mobile, scanners etc..,
* It can support on any operating system like Windows, MacOS, blackberry, android etc.,

# ADVANTAGES

The major advantages for proposed system is as follows

* Run across various devices.
* Support various operating systems.
* Portable & easy to carry the QR code anywhere.
* More user- friendly.

# FEASIBILITY STUDY

This project is feasible based on the available resources and time period for completion. An estimate is made of whether the identified user needs are satisfied using our recent software and hardware technologies. An estimate is made to show that the identified user needs are satisfied using our software and hardware technologies. The study proved that the proposed system will be cost effective, from the business point of view and it can be developed in the existing budgetary constraints. The study is done in three phases.

# 2.3.1 TESTS OF FEASIBILITY

Feasibility study is conducted once the problem clearly understood. Feasibility study is necessary to determine that the proposed system in steganalysis is feasible by considering the technical, operational, and economical factors. By having a detailed feasibility study the management in the will have a clear-cut view of the proposed system of the project. Feasibility study encompasses the following things:

* Technical Feasibility
* Economic Feasibility
* Operational feasibility

# 2.3.2 TECHNICAL FEASIBILITY

The technical feasibility study is a study of function, performances and constraints and improves the ability to create an acceptable system. Technical feasibility is frequently the most difficult area to achieve at the stage of product engineering process.

# 2.3.3 OPERATIONAL FEASIBILITY

The purpose of operational feasibility study is to determine whether the new system will be used if it is developed and installed. And whether there will be resistance from users that will undermine the possible application benefit. The first challenge was whether the system meets the organizational requirement. This is checked by the system requirement collected from the users and the management and the operational feasibility proved that the system is capable to meet its functional requirements. During the operational feasibility study the proposed system, is checked for compliance with universal standards. All the business methods implemented in the system are selected according increase the user acceptance.

# 2.3.4 ECONOMIC FEASIBILITY

A cost evaluation is weighted against ultimate or benefit derived from the developed system or product. Economic justification is generally the “Bottom line” consideration that includes cost benefit analysis, long term corporate income strategies, impact on other profit centers or products, cost of resources needed for

Development and potential market growth. When compared to the advantage obtained from implementing the system its cost is affordable.

Possible questions raised in economic analysis are:

* + - * Is the system cost effective?
      * Do benefits outweigh costs?

# CHAPTER 3

# SYSTEM SPECIFICATION

System specification is a structured collection of information that embodies the requirement of a system. This section provides an overview of the entire system or product.

# HARDWARE REQUIREMENTS

Processor : 1.4GHz 32-bit (x86) (Minimum)

Dual Core 1.8GHz 64-bit (Recommended)

RAM : 2 Giga Bytes

Hard Drive : 500 GB

Input Devices : 101 keys keyboard and mouse

# SOFTWARE REQUIREMENT

Operating System : Any (Windows, MacOS, Blackberry, etc.)

Front end : C#.NET

# CHAPTER 4 SOFTWARE DESCRIPTION

**4.1 FRONT END**

**C#.NET**

C# is an elegant and type-safe object-oriented language that enables developers to build a variety of secure and robust applications that run on the .NET Framework. You can use C# to create Windows client applications, XML Web services, distributed components, client-server applications, database applications, and much, much more. Visual C# provides an advanced code editor, convenient user interface designers, integrated debugger, and many other tools to make it easier to develop applications based on the C# language and the .NET Framework.

**ADVANTAGES OF C#.NET**

* It is more elegant and type-safe.
* The applications that you build on this framework are secure
* The framework for ASP.NET has a complementary design and rich toolbox in the form of Visual Studio.
* It can support on all platforms if the .NET is present in the system.
* Deployment is easier than ever with C#.NET
* It’s easy to write, design and maintain pages.
* Common tasks are easy to perform on this framework.

# CHAPTER 5 PROJECT DESCRIPTION

* 1. **PROBLEM DEFINITION**

The existing system cannot be used in all the operating systems and it does not support across all the devices like scanner, mobile-phones etc., the existing system is liable to mistakes and lot of time consumption. Our proposed system can support any operating system and can run on various devices. This will be much useful than the existing system, also reduces the chances of threats and attacks.

**5.2 OVERVIEW OF THE PROJECT**

Our Steganalysis of images encrypts the user file and provides security for the key and the data present in the file. Our steganalysis provides ease of access and portability as it holds everything just in the form of a QR code. It prevents threats and attacks by providing two-step authentication and higher security by hiding all the files inside a QR Code.

**5.3 MODULAR DESCRIPTION**

The STEGANALYSIS OF IMAGES USING QRC has the following modules

* Encryption & key generation
* Hiding key behind image
* Image to QR Conversion
* Extract key from image
* Decryption

# ENCRYPTION & KEY GENERATION

The files which are to be protected are obtained from the user. The files which are obtained from the user are encrypted & key is generated after the encryption. This key can be used for decryption of the file whenever needed.

* + 1. **HIDING THE KEY BEHIND IMAGE**

The key which is obtained after the encryption of the image is hidden behind an image. The image into which the key should be hidden can be chosen by the user themselves. This concept of hiding the key generated after the encryption of the selected file behind an image is called as “Steganography”.

* + 1. **IMAGE TO QR CONVERSION**

The image which now holds the hidden key called as “stego-image” is converted into a QR code for easy access and enhanced security. This helps the user to protect their data in a more secure way and QR code can be scanned easily. The reverse process of converting the QR into an image will result in decryption.

**5.3.4 EXTRACTION OF KEY FROM THE IMAGE**

The resulting QR can be scanned to view the image from which the key can be extracted. The key should be extracted from the image by scanning the QR code and this process of extraction is termed as “De-steganography”. This key can be used to decrypt the file and view the original contents of the file.

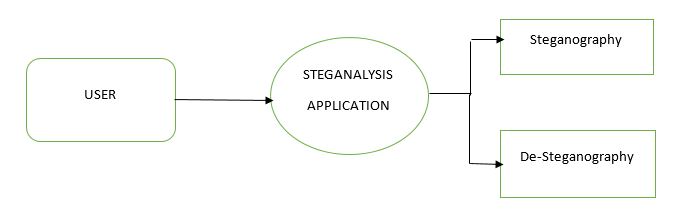
**5.3.5 DECRYPTION**

The decryption process is the reverse of the encryption process. This process must be carried out in order to retrieve the original file which is encrypted and hidden. This helps the user to view their original file if the key is known.

# DATA FLOW DIAGRAM

Data flow diagram is a 2-D diagram that explains how data is processed and transferred in a system. The graphical depiction defines each source of data and how it interacts with the other sources to reach common output. A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. DFD were useful to document the major data flows or to explore a new high-level design in terms of data flow. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.

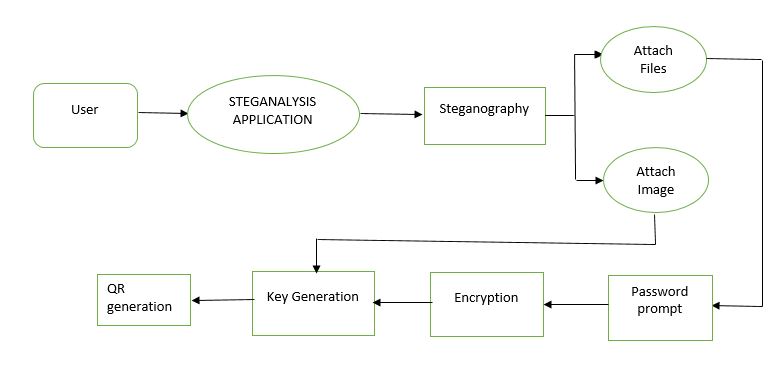
**5.4.1** **DFD LEVEL 0**

****

**Fig. 5.1 (DFD level 0)**

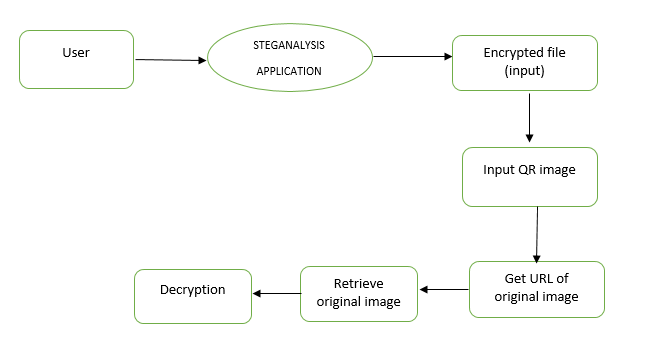
**5.4.2 DFD LEVEL 1**

**ENCRYPTION**



**Fig 5.2 (DFD level 1 - encryption)**

**DECRYPTION**

****

**Fig 5.3 (DFD level 1 – decryption)**

* 1. **INPUT DESIGN**

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

# OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making

# CHAPTER 6

# SYSTEM TESTING

System Testing is a level of the software testing where complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. By definition of ISTQB system testing is the process of testing an integrated system to verify that it meets specified.

# TESTING METHODS

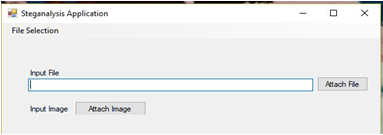
Software Testing Type is a classification of different testing activities into categories, each having, a defined test objective, test strategy, and test deliverables. The goal of having a testing type is to validate the Application under Test for the defined Test Objective.

For instance, the goal of Accessibility testing is to validate the AUT to be accessible by disabled people. So, if your Software solution must be disabled friendly, you check it against Accessibility Test Cases.

# TYPES OF TESTING

**6.2.1 UNIT TESTING**

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules. Here it is tested whether the particular file could be attached and it could be opened successfully with the use of **click** activity.



**Fig 6.1 unit testing**

**6.2.2** **INTEGRATION** **TESTING**

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Here the various files that are available in the system are displayed together to perform selection. The purpose of this level testing is to expose faults I the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

Types of Integration testing:

* Component integration testing
* System integration testing

**Component integration testing:** the testing is performed to expose the defects in the interfaces and interaction between integrated components.

**System integration testing:** Testing is performed to test the integration of the systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).

**6.2.3** **FUNCTIONAL** **TESTING**

Functions are tested by them input and examining the output, and internal program structure is rarely considered. Functional testing usually describes what the system does. Functional testing does not imply that you are testing a function (method) of your module or class. Functional testing tests a slice of functionality of the whole system. This testing involves checking of User Interface, APIs, Database, security, client/server applications and functionality of the application under test. This testing can be done either manually or using automation. Here function testing is done by attaching the different formats of files and images using **click** activity.

**6.2.4** **STRESS** **TESTING**

Stress testing a Non-Functional testing technique that is performed as part of performance testing. During stress testing, the system is monitored after subjecting the system to overload to ensure that the system can sustain the stress. It involves the testing beyond the normal operational capacity, often to a breaking point, in order to observe the results. Here the testing is carried out by monitoring the response after the completion of encryption and decryption.

**6.2.5** **ACCEPTANCE** **TESTING**

Acceptance Testing is a level of the software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system**’**s compliance with the business requirements and assess whether it is acceptable for delivery.

**6.2.6** **WHITE** **BOX** **TESTING**

White Box Testing is the testing of a software solution's internal coding and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability. White box testing is based on the inner workings of an application and revolves around internal testing. In our Steganalysis application, White Box testing involves the testing of the software code. In testing, internal security holes are managed and poorly structured paths in the coding process is rectified. The flow of specific input through the code is tested and expected outcome is also predicted. The proposed system involves testing a series of input in different forms of files. Initially the source code is understood and the test cases are created and executed.

**6.2.7** **BLACK** **BOX** **TESTING**

Black box testing is a software testing techniques in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on the software requirements and specifications.

**6.2.7.1** **METHODS** **OF** **BLACK** **BOX** **TESTING**

There are many types of Black Box Testing but following are the prominent ones-

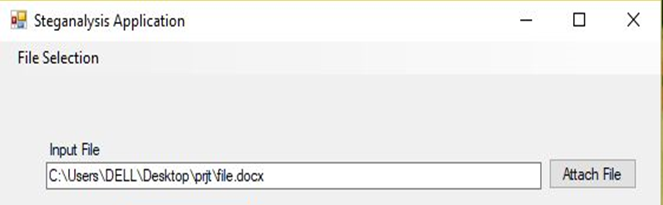
* **Functional testing** - This black box testing type is related to functional requirements of a system; it is done by software testers.
* **Non-functional testing** - This type of black box testing is not related to testing of a specific functionality, but non-functional requirements such as

Performance, scalability, usability.

* **Regression testing** - Regression testing is done after code fixes, upgrades or any other system maintenance to check the new code has not affected the existing code.
  1. **TESTING STRATEGY**

**6.3.1 File availability test case**

This test case is for testing the availability of the files which are to be encrypted and decrypted. The given file should be valid and already available in the system.



**6.3.2 QR code generation test case**

This test case is used to check whether the image is in valid format and also whether the image contains an encrypted file in it. The image which is used for the encryption must alone be attached and it must be a valid image. The QR will be generated if the image is in valid format. The QR must contain the valid URL of the image.

**CHAPTER** **7**

**SYSTEM** **IMPLEMENTATION**

System implementation is the construction of the new system by considering the existing system and implementing it. The most important task in a project is a delivery of the given system within the allotted deadline. Implementation in the phase in which one has to be cautions because all the efforts undertaken during the project will be useful only if the software is properly implemented according to the plan made. The construction phase does two things: build and test a functional system that gives business and organizational design requirements and implement the interface between the new system and existing production system.

System testing of the software or hardware is a testing being conducted on a complete, integrated system to evaluate the system’s compliance with its specified requirements. In system testing the behavior of the whole system is tested as defined by the scope of the development project. It may include tests based the risk and requirements specifications, business process, use cases, or other high-level description of the system behavior, interaction with the operating system and the system resources. System testing is most often the final test to verify that the system to be delivered meets the specification and its purpose. System testing is carried out by specialist’s tester or independent tester. It should investigate both functional and non-functional requirement of the testing.

**7.1** **QUALITY** **ASSURANCE**

The aim of quality assurance is to maintain or to ensure the quality of the system being developed. It checks whether it satisfies the requirement and meet organizational goals. The quality assurance goal in the system life cycle involves the following components:

* **Quality** **factors** **specifications** was done to determine the factors that lead to high quality of a system.
* **Correctness** is the extent to which a program meets system specification.
* **Reliability** is the degree to which a program meets system specification
* **Efficiency** is the amount of computer resource required by the entire program to perform a function.
* **Usability** is the effort required learning and operating the system.
* **Maintainability** is the ease with which the program errors are corrected.
* **Testability** is the effort required to test a program to ensure its correct performance.
* **Portability** is the case of transporting a program from one hardware configuration to another.
* **Accuracy** is required precision in input editing, computation, and output.
* **Error** **Tolerance** is the error correction and deduction versus error avoidance.
* **Expandability** is the ease adding or expanding existing database.
* **Access** **control** **and** **audit** is the control of access to the system and the extent to which that access can be audited.

**7.2** **SYSTEM** **MAINTENANCE**

Software maintenance in software engineering is the modification of a software product after a delivery to correct faults, to improve performance or other attributes. The results obtained from the evaluation process help the organization to determine whether its information systems are effective and efficient. The process of monitoring, evaluating and modifying of existing information system to make required or desirable improvement may be termed as System maintenance.

A common perception of maintenance is that it merely involves fixing defects. However, one study indicated that over 80% of maintenance effort is used for non-corrective action. This perception is perpetuated by users submitting problem reports that in reality are functionality enhancement to the system.

System maintenance is an ongoing activity which cover a wide variety of activities including removing program and design errors, updating documentation and test data and updating user support. For the purpose of convenience, maintenance may be categorized into three classes namely,

* **Adaptive** – Modifying system to cope with changes in the software environment.
* **Perfective** – Implementing new or changed user requirements which concern functional enhancement to the software.
* **Corrective** – Diagnosing and fixing errors, possibly once found by a user.
* **Preventive** – Increasing software maintainability or reliability to prevent problem in the future.

An integral part of software is the maintenance one, which requires an accurate maintenance plan to be prepared during the software development. It should specify how user will request modifications or report problem. The budget should include resource and cost estimate. A new decision should be addressed for the developing of every new system feature and its quality objective. The software maintenance which can last for 5 to 6 years or even decades, after the development process the call for an effective plan which can address the scope of software maintenance. The tailoring of the post-delivery/deployment process. The designation of who will provide maintenance and an estimate of the lifecycle cost. The selection of proper enforcement of standards is the challenging task right from the early stage of software engineering which has not got definite importance by the concerned stakeholder.

**CHAPTER** **8**

**CONCLUSION** **AND** **FUTURE** **ENHANCEMENTS**

**8.1** **CONCLUSION**

Thus the Steganaysis application can be used to share the confidential information between the sender and the receiver. The information or the file can be maintained confidential and secrecy can be maintained. By using the software, the sender can encrypt any form of file using a password. This encrypted file can be hidden behind any image which is also selected by the sender. The image can be converted to QR code which is highly secure. The scanning of QR code can reveal the URL of the image alone. The secret behind the image will be known only to the receiver. This software can prevent the hackers from hacking the information. It is more secure than any other applications that are used for cryptography already. Our application is more user friendly and compatible with all kinds of Operating Systems.

**8.2** **FUTURE** **ENHANCEMENTS**

The future enhancement in the further proposed system is that the software can improve the security for the files that are shared via cloud. It can make use of the database such that it can promote multiple senders and receivers. The filename and also the password can be stored separately in the database. This analysis using QRC can be applied in transmission of big data to overcome the shortfalls of digital media and its associated security techniques. It can be used in cloud data sharing to securely share information.

**CHAPTER** **9**

**APPENDIX**

**9.1** **SOURCE** **CODE**

**Login.cs**

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.Data.SqlClient;

namespace imgefile

{

public partial class Login : Form

{

SqlConnection con = new SqlConnection(@" Data Source=.\SQLEXPRESS;Initial Catalog=filemrg;Integrated Security=True;MultipleActiveResultSets=True");

public Login()

{

InitializeComponent();

menuStrip1.Visible = false;

}

private void Log\_bt\_Click(object sender, EventArgs e)

{

if (user\_id\_tx.Text=="admin" && pw\_tx.Text=="123")

{

menuStrip1.Visible = true;

Encryption frm = new Encryption();

frm.Show();

}

else

{

menuStrip1.Visible = false;

MessageBox.Show("Invalid password or user id");

}

}

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

{

Environment.Exit(0);

}

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

{

Encryption frm = new Encryption();

frm.Show();

}

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

{

q2 frm = new q2();

frm.Show();

}

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

{

Cloudsharing frm = new Cloudsharing();

frm.Show();

}

private void label3\_Click(object sender, EventArgs e)

{}

private void pw\_tx\_TextChanged(object sender, EventArgs e)

{}

}

}

**Login.cs [Design]**

namespace imgefile{

partial class Login{

private System.ComponentModel.IContainer components = null;

protected override void Dispose(bool disposing){

if (disposing && (components != null)) {

components.Dispose();

}

base.Dispose(disposing);

}

private void InitializeComponent(){

this.user\_id\_tx = new System.Windows.Forms.TextBox();

this.pw\_tx = new System.Windows.Forms.TextBox();

this.label1 = new System.Windows.Forms.Label();

this.label2 = new System.Windows.Forms.Label();

this.Log\_bt = new System.Windows.Forms.Button();

this.label3 = new System.Windows.Forms.Label();

this.menuStrip1 = new System.Windows.Forms.MenuStrip();

this.fileToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.endripToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.toDecryptionToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.imageCloudOptionToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.logOutToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.menuStrip1.SuspendLayout();

this.SuspendLayout();

// user\_id\_tx

this.user\_id\_tx.Location = new System.Drawing.Point(437, 202);

this.user\_id\_tx.Name = "user\_id\_tx";

this.user\_id\_tx.Size = new System.Drawing.Size(215, 20);

this.user\_id\_tx.TabIndex = 0;

// pw\_tx

this.pw\_tx.Location = new System.Drawing.Point(437, 249);

this.pw\_tx.Name = "pw\_tx";

this.pw\_tx.PasswordChar = '\*';

this.pw\_tx.Size = new System.Drawing.Size(215, 20);

this.pw\_tx.TabIndex = 1;

this.pw\_tx.TextChanged += new System.EventHandler(this.pw\_tx\_TextChanged);

// label1

this.label1.AutoSize = true;

this.label1.Location = new System.Drawing.Point(437, 183);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(43, 13);

this.label1.TabIndex = 2;

this.label1.Text = "User ID";

// label2

this.label2.AutoSize = true;

this.label2.Location = new System.Drawing.Point(437, 233);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(53, 13);

this.label2.TabIndex = 3;

this.label2.Text = "Password";

// Log\_bt

this.Log\_bt.Location = new System.Drawing.Point(440, 286);

this.Log\_bt.Name = "Log\_bt";

this.Log\_bt.Size = new System.Drawing.Size(212, 23);

this.Log\_bt.TabIndex = 4;

this.Log\_bt.Text = "Log in";

this.Log\_bt.UseVisualStyleBackColor = true;

this.Log\_bt.Click += new System.EventHandler(this.Log\_bt\_Click);

// label3

this.label3.AutoSize = true;

this.label3.Font = new System.Drawing.Font("Microsoft Sans Serif", 26.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.label3.Location = new System.Drawing.Point(350, 115);

this.label3.Name = "label3";

this.label3.Size = new System.Drawing.Size(394, 39);

this.label3.TabIndex = 5;

this.label3.Text = "Steganalysis Application";

this.label3.Click += new System.EventHandler(this.label3\_Click);

// menuStrip1

this.menuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.fileToolStripMenuItem,

this.logOutToolStripMenuItem});

this.menuStrip1.Location = new System.Drawing.Point(0, 0);

this.menuStrip1.Name = "menuStrip1";

this.menuStrip1.Size = new System.Drawing.Size(1077, 24);

this.menuStrip1.TabIndex = 18;

this.menuStrip1.Text = "menuStrip1";

// fileToolStripMenuItem

this.fileToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.endripToolStripMenuItem,

this.toDecryptionToolStripMenuItem,

this.imageCloudOptionToolStripMenuItem});

this.fileToolStripMenuItem.Name = "fileToolStripMenuItem";

this.fileToolStripMenuItem.Size = new System.Drawing.Size(37, 20);

this.fileToolStripMenuItem.Text = "File";

// endripToolStripMenuItem

this.endripToolStripMenuItem.Name = "endripToolStripMenuItem";

this.endripToolStripMenuItem.Size = new System.Drawing.Size(189, 22);

this.endripToolStripMenuItem.Text = "Encryption";

this.endripToolStripMenuItem.Click += new System.EventHandler(this.endripToolStripMenuItem\_Click);

// toDecryptionToolStripMenuItem

this.toDecryptionToolStripMenuItem.Name = "toDecryptionToolStripMenuItem";

this.toDecryptionToolStripMenuItem.Size = new System.Drawing.Size(189, 22);

this.toDecryptionToolStripMenuItem.Text = "Decryption";

this.toDecryptionToolStripMenuItem.Click += new System.EventHandler(this.toDecryptionToolStripMenuItem\_Click);

// imageCloudOptionToolStripMenuItem

this.imageCloudOptionToolStripMenuItem.Name = "imageCloudOptionToolStripMenuItem";

this.imageCloudOptionToolStripMenuItem.Size = new System.Drawing.Size(189, 22);

this.imageCloudOptionToolStripMenuItem.Text = "Image sharing Option";

this.imageCloudOptionToolStripMenuItem.Click += new System.EventHandler(this.imageCloudOptionToolStripMenuItem\_Click);

// logOutToolStripMenuItem

this.logOutToolStripMenuItem.Name = "logOutToolStripMenuItem";

this.logOutToolStripMenuItem.Size = new System.Drawing.Size(62, 20);

this.logOutToolStripMenuItem.Text = "Log Out";

this.logOutToolStripMenuItem.Click += new System.EventHandler(this.logOutToolStripMenuItem\_Click);

// Login

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(1077, 480);

this.Controls.Add(this.menuStrip1);

this.Controls.Add(this.label3);

this.Controls.Add(this.Log\_bt);

this.Controls.Add(this.label2);

this.Controls.Add(this.label1);

this.Controls.Add(this.pw\_tx);

this.Controls.Add(this.user\_id\_tx);

this.Name = "Login";

this.Text = "Login";

this.menuStrip1.ResumeLayout(false);

this.menuStrip1.PerformLayout();

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.TextBox user\_id\_tx;

private System.Windows.Forms.TextBox pw\_tx;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.Label label2;

private System.Windows.Forms.Button Log\_bt;

private System.Windows.Forms.Label label3;

private System.Windows.Forms.MenuStrip menuStrip1;

private System.Windows.Forms.ToolStripMenuItem fileToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem endripToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem toDecryptionToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem imageCloudOptionToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem logOutToolStripMenuItem;

}

}

**Encryption.cs**

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.IO;

using System.Security.Cryptography;

using System.Drawing.Imaging;

using MessagingToolkit.QRCode.Codec;

using MessagingToolkit.QRCode.Codec.Data;

namespace imgefile{

public partial class Encryption : Form

{

public Encryption(){

InitializeComponent();

}

private void openfilebt\_Click(object sender, EventArgs e){

OpenFileDialog fdlg = new OpenFileDialog();

fdlg.Title = "Open File Dialog";

fdlg.InitialDirectory = @"c:\";

fdlg.Filter = "All files (\*.\*)|\*.\*|All files (\*.\*)|\*.\*";

fdlg.FilterIndex = 2;

fdlg.RestoreDirectory = true;

if (fdlg.ShowDialog() == DialogResult.OK){

filepath.Text = fdlg.FileName;

// outputpath.Text = Path.GetExtension(fdlg.FileName);

outputpath.Text = Path.GetDirectoryName(fdlg.FileName);

}

}

private void img\_bt\_Click(object sender, EventArgs e){

OpenFileDialog openfile = new OpenFileDialog();

openfile.Filter = "Image Files (\*.jpeg; \*.png; \*.bmp)|\*.jpg; \*.png; \*.bmp";

openfile.Multiselect = false;

openfile.ShowDialog();

if (!String.IsNullOrEmpty(openfile.FileName)){

pictureBox1.BackgroundImage = Image.FromFile(openfile.FileName);

}

}

private void Encryption\_bt\_Click(object sender, EventArgs e){

string embtext = pw\_tx.Text;

// string password = @"myKey123"; // Your Key Here

string password = pw\_tx.Text; // Your Key Here

UnicodeEncoding UE = new UnicodeEncoding();

byte[] key = UE.GetBytes(password);

string outpath = Path.GetDirectoryName(filepath.Text) + Guid.NewGuid().ToString()+Path.GetExtension(filepath.Text);

string cryptFile = outpath;

FileStream fsCrypt = new FileStream(cryptFile, FileMode.Create);

RijndaelManaged RMCrypto = new RijndaelManaged();

CryptoStream cs = new CryptoStream(fsCrypt,

RMCrypto.CreateEncryptor(key, key),

CryptoStreamMode.Write);

FileStream fsIn = new FileStream(filepath.Text, FileMode.Open);

int data;

while ((data = fsIn.ReadByte()) != -1)

cs.WriteByte((byte)data);

fsIn.Close();

cs.Close();

fsCrypt.Close();

File.Delete(filepath.Text);

File.Move(outpath, filepath.Text);

Bitmap img = (Bitmap)pictureBox1.BackgroundImage;

string Steg\_pw = "123";

if (string.IsNullOrEmpty(embtext)){

MessageBox.Show("Empty string can't be hidden", "Warning");

return;

}

if (string.IsNullOrEmpty(Steg\_pw)){

MessageBox.Show("Password can't be empty", "Warning");

return;

}

try{

embtext = RijndaelAlgo.Encrypt(embtext, Steg\_pw);

}

catch (Exception ex){

MessageBox.Show(ex.Message);

}

try{

img = SteganographyHelper.MergeText(embtext, img);

string filename = DateTime.Now.ToString("dd-MM-yyyy\_HH-mm-ss") + ".png";

string img\_location = "F:\\img\\" + filename;

img.Save(img\_location, ImageFormat.Png);

string code = "http://localhost" + "/" + filename;

QRCodeEncoder enc = new QRCodeEncoder();

Bitmap bitMap = enc.Encode(code);

pictureBox2.BackgroundImage = bitMap as Image;

SaveFileDialog sav = new SaveFileDialog();

sav.Filter = "Png Image|\*.png|Bitmap Image|\*.bmp";

if (sav.ShowDialog() == DialogResult.OK){

switch (sav.FilterIndex){

case 0:

{

bitMap.Save(sav.FileName, ImageFormat.Png);

}

break;

case 1:

{

bitMap.Save(sav.FileName, ImageFormat.Bmp);

}

break;

}

}

MessageBox.Show("File Encrypted successfully", "Information");

}

catch (Exception ex){

MessageBox.Show("Encryption failed! "+ex.Message , "Error");

}

}

private void pw\_tx\_TextChanged(object sender, EventArgs e)

{ }

}}

**Encryption.cs [design]**

using System;

namespace imgefile{

partial class Encryption{

private System.ComponentModel.IContainer components = null;

protected override void Dispose(bool disposing){

if (disposing && (components != null)){

components.Dispose();

}

base.Dispose(disposing);

}

private void InitializeComponent(){

this.label4 = new System.Windows.Forms.Label();

this.pw\_tx = new System.Windows.Forms.TextBox();

this.pictureBox1 = new System.Windows.Forms.PictureBox();

this.label3 = new System.Windows.Forms.Label();

this.button1 = new System.Windows.Forms.Button();

this.outputpath = new System.Windows.Forms.TextBox();

this.Encryption\_bt = new System.Windows.Forms.Button();

this.label2 = new System.Windows.Forms.Label();

this.img\_bt = new System.Windows.Forms.Button();

this.label1 = new System.Windows.Forms.Label();

this.openfilebt = new System.Windows.Forms.Button();

this.filepath = new System.Windows.Forms.TextBox();

this.pictureBox2 = new System.Windows.Forms.PictureBox();

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.pictureBox2)).BeginInit();

this.SuspendLayout();

// label4

this.label4.AutoSize = true;

this.label4.Location = new System.Drawing.Point(124, 355);

this.label4.Name = "label4";

this.label4.Size = new System.Drawing.Size(53, 13);

this.label4.TabIndex = 28;

this.label4.Text = "Password";

// pw\_tx

this.pw\_tx.Location = new System.Drawing.Point(124, 374);

this.pw\_tx.Name = "pw\_tx";

this.pw\_tx.PasswordChar = '\*';

this.pw\_tx.Size = new System.Drawing.Size(501, 20);

this.pw\_tx.TabIndex = 27;

this.pw\_tx.TextChanged += new System.EventHandler(this.pw\_tx\_TextChanged);

// pictureBox1

this.pictureBox1.Location = new System.Drawing.Point(125, 87);

this.pictureBox1.Name = "pictureBox1";

this.pictureBox1.Size = new System.Drawing.Size(499, 248);

this.pictureBox1.TabIndex = 26;

this.pictureBox1.TabStop = false;

// label3

this.label3.AutoSize = true;

this.label3.Location = new System.Drawing.Point(127, 421);

this.label3.Name = "label3";

this.label3.Size = new System.Drawing.Size(83, 13);

this.label3.TabIndex = 25;

this.label3.Text = "Output Location";

// button1

this.button1.Location = new System.Drawing.Point(550, 437);

this.button1.Name = "button1";

this.button1.Size = new System.Drawing.Size(75, 23);

this.button1.TabIndex = 24;

this.button1.Text = "open file";

this.button1.UseVisualStyleBackColor = true;

this.button1.Click += new System.EventHandler(this.button1\_Click);

// outputpath

this.outputpath.Location = new System.Drawing.Point(124, 440);

this.outputpath.Name = "outputpath";

this.outputpath.Size = new System.Drawing.Size(420, 20);

this.outputpath.TabIndex = 23;

// Encryption\_bt

this.Encryption\_bt.Location = new System.Drawing.Point(124, 486);

this.Encryption\_bt.Name = "Encryption\_bt";

this.Encryption\_bt.Size = new System.Drawing.Size(498, 23);

this.Encryption\_bt.TabIndex = 22;

this.Encryption\_bt.Text = "Encryption";

this.Encryption\_bt.UseVisualStyleBackColor = true;

this.Encryption\_bt.Click += new System.EventHandler(this.Encryption\_bt\_Click);

// label2

this.label2.AutoSize = true;

this.label2.Location = new System.Drawing.Point(126, 71);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(63, 13);

this.label2.TabIndex = 21;

this.label2.Text = "Input Image";

// img\_bt

this.img\_bt.Location = new System.Drawing.Point(195, 66);

this.img\_bt.Name = "img\_bt";

this.img\_bt.Size = new System.Drawing.Size(111, 23);

this.img\_bt.TabIndex = 20;

this.img\_bt.Text = "Select Image";

this.img\_bt.UseVisualStyleBackColor = true;

this.img\_bt.Click += new System.EventHandler(this.img\_bt\_Click);

// label1

this.label1.AutoSize = true;

this.label1.Location = new System.Drawing.Point(126, 14);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(50, 13);

this.label1.TabIndex = 19;

this.label1.Text = "Input File";

// openfilebt

this.openfilebt.Location = new System.Drawing.Point(552, 27);

this.openfilebt.Name = "openfilebt";

this.openfilebt.Size = new System.Drawing.Size(75, 23);

this.openfilebt.TabIndex = 18;

this.openfilebt.Text = "Select File";

this.openfilebt.UseVisualStyleBackColor = true;

this.openfilebt.Click += new System.EventHandler(this.openfilebt\_Click);

// filepath

this.filepath.Location = new System.Drawing.Point(126, 30);

this.filepath.Name = "filepath";

this.filepath.Size = new System.Drawing.Size(420, 20);

this.filepath.TabIndex = 17;

// pictureBox2

this.pictureBox2.BackgroundImageLayout = System.Windows.Forms.ImageLayout.Stretch;

this.pictureBox2.Location = new System.Drawing.Point(680, 309);

this.pictureBox2.Name = "pictureBox2";

this.pictureBox2.Size = new System.Drawing.Size(200, 200);

this.pictureBox2.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage;

this.pictureBox2.TabIndex = 29;

this.pictureBox2.TabStop = false;

// Encryption

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(1164, 741);

this.Controls.Add(this.pictureBox2);

this.Controls.Add(this.label4);

this.Controls.Add(this.pw\_tx);

this.Controls.Add(this.pictureBox1);

this.Controls.Add(this.label3);

this.Controls.Add(this.button1);

this.Controls.Add(this.outputpath);

this.Controls.Add(this.Encryption\_bt);

this.Controls.Add(this.label2);

this.Controls.Add(this.img\_bt);

this.Controls.Add(this.label1);

this.Controls.Add(this.openfilebt);

this.Controls.Add(this.filepath);

this.Name = "Encryption";

this.Text = "Encryption";

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.pictureBox2)).EndInit();

this.ResumeLayout(false);

this.PerformLayout();

}

private void button1\_Click(object sender, EventArgs e) {

throw new NotImplementedException();

}

#endregion

private System.Windows.Forms.Label label4;

private System.Windows.Forms.TextBox pw\_tx;

private System.Windows.Forms.PictureBox pictureBox1;

private System.Windows.Forms.Label label3;

private System.Windows.Forms.Button button1;

private System.Windows.Forms.TextBox outputpath;

private System.Windows.Forms.Button Encryption\_bt;

private System.Windows.Forms.Label label2;

private System.Windows.Forms.Button img\_bt;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.Button openfilebt;

private System.Windows.Forms.TextBox filepath;

private System.Windows.Forms.PictureBox pictureBox2;

}}

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace imgefile{

static class Program{

[STAThread]

static void Main(){

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Login());

}

}

}

**RijndaelAlgo.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Security.Cryptography;

using System.Security;

using System.Runtime.InteropServices;

using System.Text.RegularExpressions;

namespace imgefile{

class RijndaelAlgo{

static string saltValue = "s@!tVal\*e";

static string hashAlgorithm = "SHA1";

static int passwordIterations = 2;

static string initVector = "@1B2c3!@)#$%@#X6g7FG";

static int keySize = 256;

public static string Encrypt(string plainText, string passPhrase){

byte[] initVectorBytes = Encoding.ASCII.GetBytes(initVector);

byte[] saltValueBytes = Encoding.ASCII.GetBytes(saltValue);

byte[] plainTextBytes = Encoding.UTF8.GetBytes(plainText);

PasswordDeriveBytes password = new PasswordDeriveBytes(

passPhrase,

saltValueBytes,

hashAlgorithm,

passwordIterations);

byte[] keyBytes = password.GetBytes(keySize / 8);

RijndaelManaged symmetricKey = new RijndaelManaged();

symmetricKey.Mode = CipherMode.ECB;

ICryptoTransform encryptor = symmetricKey.CreateEncryptor(keyBytes, initVectorBytes);

MemoryStream memoryStream = new MemoryStream();

CryptoStream cryptoStream = new CryptoStream(memoryStream, encryptor, CryptoStreamMode.Write);

cryptoStream.Write(plainTextBytes, 0, plainTextBytes.Length);

cryptoStream.FlushFinalBlock();

byte[] cipherTextBytes = memoryStream.ToArray();

memoryStream.Close();

cryptoStream.Close();

string cipherText = Convert.ToBase64String(cipherTextBytes);

return cipherText;

}

public static string Decrypt(string cipherText, string passPhrase) {

byte[] initVectorBytes = Encoding.ASCII.GetBytes(initVector);

byte[] saltValueBytes = Encoding.ASCII.GetBytes(saltValue);

byte[] cipherTextBytes = Convert.FromBase64String(cipherText);

PasswordDeriveBytes password = new PasswordDeriveBytes(passPhrase, saltValueBytes, hashAlgorithm, passwordIterations);

byte[] keyBytes = password.GetBytes(keySize / 8);

RijndaelManaged symmetricKey = new RijndaelManaged();

symmetricKey.Mode = CipherMode.ECB;

ICryptoTransform decryptor = symmetricKey.CreateDecryptor(keyBytes, initVectorBytes);

MemoryStream memoryStream = new MemoryStream(cipherTextBytes);

CryptoStream cryptoStream = new CryptoStream(memoryStream, decryptor, CryptoStreamMode.Read);

byte[] plainTextBytes = new byte[cipherTextBytes.Length];

// Start decrypting.

int decryptedByteCount = cryptoStream.Read(plainTextBytes, 0, plainTextBytes.Length);

memoryStream.Close();

cryptoStream.Close();

string plainText = Encoding.UTF8.GetString(plainTextBytes, 0, decryptedByteCount);

// Return decrypted string.

return plainText;

}}}

**SteganographyHelper.cs**

using System;

using System.Drawing;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace imgefile{

class SteganographyHelper{

enum State{

hiding,

filling\_with\_zeros

};

public static Bitmap MergeText(string text, Bitmap bmp){

State s = State.hiding;

int charIndex = 0;

int charValue = 0;

long colorUnitIndex = 0;

int zeros = 0;

int R = 0, G = 0, B = 0;

for (int i = 0; i < bmp.Height; i++){

for (int j = 0; j < bmp.Width; j++){

Color pixel = bmp.GetPixel(j, i);

pixel = Color.FromArgb(pixel.R - pixel.R % 2,

pixel.G - pixel.G % 2, pixel.B - pixel.B % 2);

R = pixel.R; G = pixel.G; B = pixel.B;

for (int n = 0; n < 3; n++){

if (colorUnitIndex % 8 == 0){

if (zeros == 8){

if ((colorUnitIndex - 1) % 3 < 2){

bmp.SetPixel(j, i, Color.FromArgb(R, G, B));

}

return bmp;

}

if (charIndex >= text.Length){

s = State.filling\_with\_zeros;

}

else{

charValue = text[charIndex++];

}

}

switch (colorUnitIndex % 3)

{

case 0:

{

if (s == State.hiding){

R += charValue % 2;

charValue /= 2;

}

}

break;

case 1:

{

if (s == State.hiding){

G += charValue % 2;

charValue /= 2;

}

}

break;

case 2:

{

if (s == State.hiding){

B += charValue % 2;

charValue /= 2;

}

bmp.SetPixel(j, i, Color.FromArgb(R, G, B));

}

break;

}

colorUnitIndex++;

if (s == State.filling\_with\_zeros){

zeros++;

}

}

}

}

return bmp;

}

public static string ExtractText(Bitmap bmp){

int colorUnitIndex = 0;

int charValue = 0;

string extractedText = String.Empty;

for (int i = 0; i < bmp.Height; i++){

for (int j = 0; j < bmp.Width; j++){

Color pixel = bmp.GetPixel(j, i);

for (int n = 0; n < 3; n++){

switch (colorUnitIndex % 3)

{

case 0:

{

charValue = charValue \* 2 + pixel.R % 2;

}

break;

case 1:

{

charValue = charValue \* 2 + pixel.G % 2;

}

break;

case 2:

{

charValue = charValue \* 2 + pixel.B % 2;

}

break;

}

colorUnitIndex++;

if (colorUnitIndex % 8 == 0){

charValue = reverseBits(charValue);

if (charValue == 0){

return extractedText;

}

char c = (char)charValue;

extractedText += c.ToString();

}

}

}

}

return extractedText;

}

public static int reverseBits(int n){

int result = 0;

for (int i = 0; i < 8; i++){

result = result \* 2 + n % 2;

n /= 2;

}

return result;

} }}

**Decryption.cs**

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 MessagingToolkit.QRCode.Codec;

using MessagingToolkit.QRCode.Codec.Data;

using System.Net;

using System.IO;

using System.Security.Cryptography;

namespace imgefile{

public partial class q2 : Form{

public q2(){

InitializeComponent();

}

private void img\_bt\_Click(object sender, EventArgs e){

if(System.IO.File.Exists(filepath.Text)){

OpenFileDialog openfile = new OpenFileDialog();

openfile.Filter = "Image Files (\*.jpeg; \*.png; \*.bmp)|\*.jpg; \*.png; \*.bmp";

openfile.Multiselect = false;

openfile.ShowDialog();

if (!String.IsNullOrEmpty(openfile.FileName){

qrimg.BackgroundImage = Image.FromFile(openfile.FileName);

}}

else{

MessageBox.Show("Kindly select encrypted File");

}}

private void get\_image\_from\_qr\_Click(object sender, EventArgs e){

if(qrimg.BackgroundImage != null){

this.get\_ulr\_from\_qr();

}

else{

MessageBox.Show("Kindly select QR Code To get ULR");

}

}

protected void get\_ulr\_from\_qr(){

QRCodeDecoder dec = new QRCodeDecoder();

URL\_tx.Text = (dec.decode(new QRCodeBitmapImage(qrimg.BackgroundImage as Bitmap)));

}

private void button1\_Click(object sender, EventArgs e){

if (URL\_tx.Text != ""){

this.load\_image\_from\_url(URL\_tx.Text);

}

else{

MessageBox.Show("URL is Empty");

}

}

protected void load\_image\_from\_url(string url){

var request = WebRequest.Create(url);

using (var response = request.GetResponse())

using (var stream = response.GetResponseStream())

{

ulr\_image.BackgroundImage = Bitmap.FromStream(stream);

}

}

private void Decryption\_bt\_Click(object sender, EventArgs e){

try{

Bitmap img = (Bitmap)ulr\_image.BackgroundImage;

string Steg\_pw = "123";

string extractString = SteganographyHelper.ExtractText(img);

if (string.IsNullOrEmpty(Steg\_pw)){

MessageBox.Show("Please specify the password, To Extract the information.", "Security");

return;

}

extractString = RijndaelAlgo.Decrypt(extractString, Steg\_pw);

string inputFile = filepath.Text;

string outpath = Path.GetDirectoryName(filepath.Text) + Guid.NewGuid().ToString() + Path.GetExtension(filepath.Text);

string outputFile = outpath;

string password = extractString; // Your Key Here

UnicodeEncoding UE = new UnicodeEncoding();

byte[] key = UE.GetBytes(password);

FileStream fsCrypt = new FileStream(inputFile, FileMode.Open);

RijndaelManaged RMCrypto = new RijndaelManaged();

CryptoStream cs = new CryptoStream(fsCrypt,

RMCrypto.CreateDecryptor(key, key),

CryptoStreamMode.Read);

FileStream fsOut = new FileStream(outputFile, FileMode.Create);

int data;

while ((data = cs.ReadByte()) != -1)

fsOut.WriteByte((byte)data);

fsOut.Close();

cs.Close();

fsCrypt.Close();

File.Delete(filepath.Text);

File.Move(outpath, filepath.Text);

MessageBox.Show("File Decrypted successfully", "Information");

}

catch (Exception ex) {

MessageBox.Show("Decryption failed! " + ex.Message, "Error");

return;

}}

private void openfilebt\_Click(object sender, EventArgs e){

OpenFileDialog fdlg = new OpenFileDialog();

fdlg.Title = "Open File Dialog";

fdlg.InitialDirectory = @"c:\";

fdlg.Filter = "All files (\*.\*)|\*.\*|All files (\*.\*)|\*.\*";

fdlg.FilterIndex = 2;

fdlg.RestoreDirectory = true;

if (fdlg.ShowDialog() == DialogResult.OK){

filepath.Text = fdlg.FileName;

}}}}

**Decryption.cs [design]**

namespace imgefile{

partial class q2{

private System.ComponentModel.IContainer components = null;

protected override void Dispose(bool disposing){

if (disposing && (components != null)){

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

private void InitializeComponent()

{

this.components = new System.ComponentModel.Container();

this.qrimg = new System.Windows.Forms.PictureBox();

this.label2 = new System.Windows.Forms.Label();

this.img\_bt = new System.Windows.Forms.Button();

this.label1 = new System.Windows.Forms.Label();

this.openfilebt = new System.Windows.Forms.Button();

this.filepath = new System.Windows.Forms.TextBox();

this.get\_image\_from\_qr = new System.Windows.Forms.Button();

this.contextMenuStrip1 = new System.Windows.Forms.ContextMenuStrip(this.components);

this.ulr\_image = new System.Windows.Forms.PictureBox();

this.URL\_tx = new System.Windows.Forms.TextBox();

this.label3 = new System.Windows.Forms.Label();

this.button1 = new System.Windows.Forms.Button();

this.Decryption\_bt = new System.Windows.Forms.Button();

((System.ComponentModel.ISupportInitialize)(this.qrimg)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.ulr\_image)).BeginInit();

this.SuspendLayout();

// qrimg

this.qrimg.BackgroundImageLayout = System.Windows.Forms.ImageLayout.Stretch;

this.qrimg.Location = new System.Drawing.Point(106, 91);

this.qrimg.Name = "qrimg";

this.qrimg.Size = new System.Drawing.Size(200, 200);

this.qrimg.TabIndex = 34;

this.qrimg.TabStop = false;

// label2

this.label2.AutoSize = true;

this.label2.Location = new System.Drawing.Point(103, 67);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(88, 13);

this.label2.TabIndex = 33;

this.label2.Text = "Select QR Image";

// img\_bt

this.img\_bt.Location = new System.Drawing.Point(197, 62);

this.img\_bt.Name = "img\_bt";

this.img\_bt.Size = new System.Drawing.Size(75, 23);

this.img\_bt.TabIndex = 32;

this.img\_bt.Text = "open image";

this.img\_bt.UseVisualStyleBackColor = true;

this.img\_bt.Click += new System.EventHandler(this.img\_bt\_Click);

// label1

this.label1.AutoSize = true;

this.label1.Location = new System.Drawing.Point(103, 10);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(107, 13);

this.label1.TabIndex = 31;

this.label1.Text = "Select Encrypted File";

// openfilebt

this.openfilebt.Location = new System.Drawing.Point(529, 23);

this.openfilebt.Name = "openfilebt";

this.openfilebt.Size = new System.Drawing.Size(75, 23);

this.openfilebt.TabIndex = 30;

this.openfilebt.Text = "open file";

this.openfilebt.UseVisualStyleBackColor = true;

this.openfilebt.Click += new System.EventHandler(this.openfilebt\_Click);

// filepath

this.filepath.Location = new System.Drawing.Point(103, 26);

this.filepath.Name = "filepath";

this.filepath.Size = new System.Drawing.Size(420, 20);

this.filepath.TabIndex = 29;

// get\_image\_from\_qr

this.get\_image\_from\_qr.Location = new System.Drawing.Point(104, 297);

this.get\_image\_from\_qr.Name = "get\_image\_from\_qr";

this.get\_image\_from\_qr.Size = new System.Drawing.Size(202, 23);

this.get\_image\_from\_qr.TabIndex = 37;

this.get\_image\_from\_qr.Text = "Get URL";

this.get\_image\_from\_qr.UseVisualStyleBackColor = true;

this.get\_image\_from\_qr.Click += new System.EventHandler(this.get\_image\_from\_qr\_Click);

// contextMenuStrip1

this.contextMenuStrip1.Name = "contextMenuStrip1";

this.contextMenuStrip1.Size = new System.Drawing.Size(61, 4);

// ulr\_image

this.ulr\_image.BackgroundImageLayout = System.Windows.Forms.ImageLayout.Stretch;

this.ulr\_image.Location = new System.Drawing.Point(365, 113);

this.ulr\_image.Name = "ulr\_image";

this.ulr\_image.Size = new System.Drawing.Size(381, 178);

this.ulr\_image.TabIndex = 38;

this.ulr\_image.TabStop = false;

// URL\_tx

this.URL\_tx.Location = new System.Drawing.Point(365, 91);

this.URL\_tx.Name = "URL\_tx";

this.URL\_tx.Size = new System.Drawing.Size(262, 20);

this.URL\_tx.TabIndex = 39;

// label3

this.label3.AutoSize = true;

this.label3.Location = new System.Drawing.Point(362, 72);

this.label3.Name = "label3";

this.label3.Size = new System.Drawing.Size(61, 13);

this.label3.TabIndex = 40;

this.label3.Text = "Image URL";

// button1

this.button1.Location = new System.Drawing.Point(633, 88);

this.button1.Name = "button1";

this.button1.Size = new System.Drawing.Size(113, 23);

this.button1.TabIndex = 41;

this.button1.Text = "Get Image from URL";

this.button1.UseVisualStyleBackColor = true;

this.button1.Click += new System.EventHandler(this.button1\_Click);

// Decryption\_bt

this.Decryption\_bt.Location = new System.Drawing.Point(106, 336);

this.Decryption\_bt.Name = "Decryption\_bt";

this.Decryption\_bt.Size = new System.Drawing.Size(640, 23);

this.Decryption\_bt.TabIndex = 42;

this.Decryption\_bt.Text = "Decryption";

this.Decryption\_bt.UseVisualStyleBackColor = true;

this.Decryption\_bt.Click += new System.EventHandler(this.Decryption\_bt\_Click);

// q2

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(759, 582);

this.Controls.Add(this.Decryption\_bt);

this.Controls.Add(this.button1);

this.Controls.Add(this.label3);

this.Controls.Add(this.URL\_tx);

this.Controls.Add(this.ulr\_image);

this.Controls.Add(this.get\_image\_from\_qr);

this.Controls.Add(this.qrimg);

this.Controls.Add(this.label2);

this.Controls.Add(this.img\_bt);

this.Controls.Add(this.label1);

this.Controls.Add(this.openfilebt);

this.Controls.Add(this.filepath);

this.Name = "q2";

this.Text = "Decryption";

((System.ComponentModel.ISupportInitialize)(this.qrimg)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.ulr\_image)).EndInit();

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.PictureBox qrimg;

private System.Windows.Forms.Label label2;

private System.Windows.Forms.Button img\_bt;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.Button openfilebt;

private System.Windows.Forms.TextBox filepath;

private System.Windows.Forms.Button get\_image\_from\_qr;

private System.Windows.Forms.ContextMenuStrip contextMenuStrip1;

private System.Windows.Forms.PictureBox ulr\_image;

private System.Windows.Forms.TextBox URL\_tx;

private System.Windows.Forms.Label label3;

private System.Windows.Forms.Button button1;

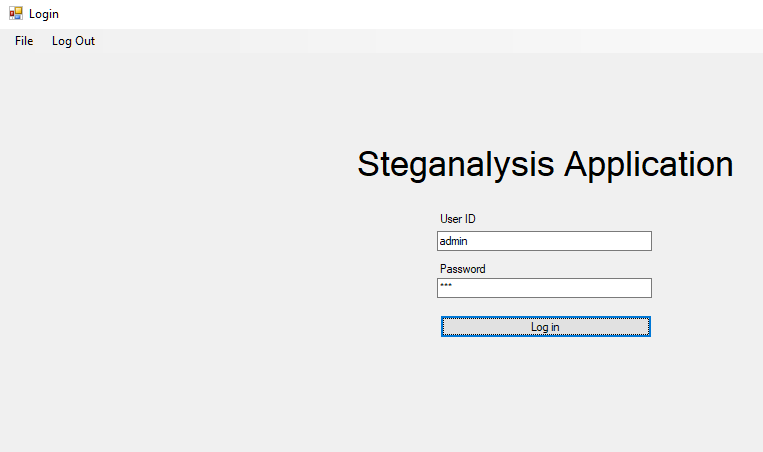
private System.Windows.Forms.Button Decryption\_bt;

}}

**9.2 SCREENSHOTS**

**9.2.1 Login Page**

The login page provides the login access for the users to use the Steganalysis application. It performs the authentication activity by validating the username and the password. If the username and the password matches then it moves on to the next page.



**9.2.2 Encryption Page**

This is the first page that is used to encrypt the file selected by the user.