Encryption project

IE322



|  |  |
| --- | --- |
| Name | ID |
| Abdullah Yamani | 2338802 |
| Abdulelah Dawood | 2335758 |
| Abdulmohsen Alqahtani | 2336159 |
| Hamza Algothmi | 2340448 |
| Abdulmajeed Bamalan | 2340637 |

Team 5

Table of Contents

[Introduction: 3](#_Toc197099381)

[1. Programming Environment & Tools Used 3](#_Toc197099382)

[2. Overview of the Project 3](#_Toc197099383)

[3. Explanation of the Encryption Methods 3](#_Toc197099384)

[4. Functionality of the Code & Key Variables 3](#_Toc197099385)

[5. How the Application Works 4](#_Toc197099386)

[6. User Benefits and Educational Purpose 4](#_Toc197099387)

[Conclusion: 5](#_Toc197099388)

# **Introduction:**

Encryption has been used to hide the true meaning of messages by altering the way messages are communicated. This method has been around for many centuries, but now in the new age encryption is easy to access like never before. In this project, we will present three types of encryption methods that are accessible to users through our Graphical User Interface (GUI). This project was made to teach people about encryptions and its methods. We have applied our knowledge of this course to build the GUI and have used the C# program to do so. In this report we will include a description of the code and its functionalities, we will also describe the three methods of encryptions available in the program.

## **1. Programming Environment & Tools Used**

This encryption project was developed using C# in Visual Studio with the Windows Forms (WinForms) framework. The project utilizes a GUI-based interface, making it user-friendly and interactive. Configuration and resources are managed through associated `.config`, `.resx`, `.Designer.cs`, and `.csproj` files.

## **2. Overview of the Project**

The main goal of the project is to educate users on three common encryption techniques by offering an interactive interface. The encryption methods are implemented and integrated into the application, where users can select the method of choice, input text, and observe how it is encrypted or decrypted. The GUI makes the concept of encryption accessible and engaging.

## **3. Explanation of the Encryption Methods**

Each form represents a different encryption method:

- Text Encryption: Implements a basic substitution cipher. Each character in the input text is replaced with a corresponding encrypted character based on a fixed logic.

- Picture Encryption: Contains an AES (Advanced Encryption Standard)-like logic or a shift cipher model, demonstrating slightly more complex transformations.

- Morse Encryption: Implements Morse code encryption. The text is translated into Morse code using dots and dashes and can also be translated back.

## **4. Functionality of the Code & Key Variables**

* Form1.cs:

Form1.cs implements the main menu or landing form of the application. It provides navigation buttons that open other forms corresponding to different encryption methods. The buttons likely open Form3 (text encryption), Form4 (picture encryption), and Form5 (Morse encryption). The code handles button click events to show the appropriate form when a user chooses an encryption option.

* Form3.cs:

Form3.cs manages the 'Text Encryption' feature of the application. It allows the user to input text and apply an encryption algorithm to obscure the message. The form likely includes text fields and buttons for both encryption and decryption. Event handlers are used to capture user input and display the encrypted or decrypted result.

* Form4.cs:

Form4.cs is responsible for 'Picture Encryption', enabling users to hide messages within images. The form provides functionality to select an image file, embed text into the image, and possibly extract it. It includes logic for basic steganography techniques and manages input/output image streams and messages.

* Form5.cs:

Form5.cs implements the 'Morse Encryption' feature, converting user input text into Morse code and vice versa. It contains functions for encrypting plaintext into Morse code and decrypting Morse code back to readable text. The interface allows users to enter input and view the result in a separate text field.

* Page1ForText.cs:

Page1ForText.cs appears to be a supplemental or alternative form for handling text encryption, possibly providing an extended UI or advanced features beyond what is available in Form3. It includes additional controls and event handling for interacting with encrypted text.

* Program.cs:

Program.cs contains the main entry point for the application. It initializes the Windows Forms environment and launches Form1 as the starting form of the application. This is standard for C# Windows Forms apps to set up the application loop and start GUI execution.

Key Variables Across Forms:

- inputTextBox: Captures user input text for encryption or decryption.

- outputTextBox: Displays the resulting text after processing.

- encryptButton, decryptButton: Triggers encryption/decryption logic respectively.

## **5. How the Application Works**

Upon launching the application, the user is greeted with Form1, where they can choose between different encryption methods via buttons. Each button directs the user to the corresponding form:

- The selected form allows the user to enter text.

- The user can then choose to encrypt or decrypt the text.

- The result is displayed on the screen.

Navigation is handled via event handlers attached to buttons that call Form.Show() to open new forms.

## **6. User Benefits and Educational Purpose**

The application simplifies complex encryption concepts through visual and interactive demonstrations. It enables users to:

- Learn how different encryption methods transform data.

- Experiment with encoding and decoding messages.

- Understand practical uses of encryption.

It serves as a practical tool for students or individuals exploring cryptography basics and programming in C# with WinForms.

This report describes how code architecture, interface design, and cryptographic logic come together to build an educational and functional encryption app.

# **Conclusion:**

This project demonstrates the practical application of fundamental encryption techniques through an interactive C# Windows Forms application. By incorporating three encryption methods—Text Encryption, Picture Encryption, and Morse Encryption—into a user-friendly GUI, the program offers an accessible way for users to learn about and experiment with data security concepts. Each form was designed to handle a distinct type of input, ensuring versatility and clarity in demonstrating how encryption works in different contexts. Beyond meeting course objectives, the project emphasizes the importance of data protection and introduces users to the basics of cryptography in an engaging and educational format.