# Mini Project Report: Text Encryption and Decryption GUI Tool

**Technology Used:** Python, Tkinter, Base64 Encoding
**Project Type:** Mini Project
**Domain:** Cryptography / GUI Development
**Duration:** 1 Day
**Status:** Completed

## 1. Objective

To develop a graphical user interface (GUI) tool in Python that allows users to **encrypt** and **decrypt** plain text using Base64 encoding. The application includes password-protected functionality and a reset option for secure message processing.

## 2. Abstract

Data security is a fundamental concern in digital communication. This project presents a GUI-based encryption-decryption tool built using Python's Tkinter module and Base64 encoding. It enables users to transform plain messages into encoded strings and retrieve them back using a valid password. Though Base64 is not a cryptographically secure method, this mini project focuses on the basics of encryption logic and GUI programming — making it an excellent beginner-friendly tool in the field of secure data handling.

## 3. Introduction

Encryption is the process of converting readable data into an unreadable format to prevent unauthorized access, while decryption is the reverse process. This project focuses on **symmetric password-protected encoding** using Base64, a method commonly used in data transfer formats (e.g., email attachments, APIs).

The application provides a **user-friendly GUI** for entering the message and a secret key. If the password is correct, the text is either encrypted (converted to Base64) or decrypted (decoded from Base64). Incorrect or blank passwords trigger user-friendly error messages.

## 4. Technologies Used

| Component | Description |
|---|---|
| **Python 3.13** | Programming language |
| **Tkinter** | GUI creation framework |
| **Base64 Module** | Encoding and decoding of messages |
| **Messagebox** | For showing error dialogs |
| **Pillow (optional)** | For adding app icons/images |

## 5. Features

- GUI-based application using Tkinter
- Encryption (Base64 encoding) and Decryption (Base64 decoding)
- Password-protected functionality (password hardcoded as `"1234"`)
- Simple and clean UI with buttons: **Encrypt**, **Decrypt**, **Reset**
- Custom window title and icon support
- Displays results in a new popup window

## 6. Execution Flow

- **User Interface Initialization**
    - Tkinter `Tk()` instance is created and configured (title, icon, size).
    - Labels and Text widgets are placed to accept input and key.
- **Encrypt Function**
    - If password = `"1234"`, it fetches text, encodes it in ASCII → Base64.
    - Result displayed in a new popup with red background.
- **Decrypt Function**
    - Password is verified.
    - Message is decoded from Base64 to ASCII.
    - Output is shown in a green popup.

- **Reset Button**
    - Clears the input message and password field.
- **Error Handling**
    - Messagebox is used to show:
        - Blank password input
        - Incorrect password

## 7. Password Logic

A **hardcoded key (`"1234"`)** is required for both encryption and decryption.

- If the key is incorrect, an error message is shown and processing is aborted.
- This approach mimics simple authentication mechanisms in encryption tools

## 8. Code Structure Overview

```
#  Imports
from tkinter import *
from tkinter import messagebox
import base64

# Encryption Function
base64.b64encode(message.encode("ascii")).decode("ascii")

#  Decryption Function
base64.b64decode(message.encode("ascii")).decode("ascii")
```

- ***GUI Window***
    - Labels, Text fields, Buttons
    - `screen.geometry("500x500")`
    - `iconphoto()` for application branding

- ***Security***
    - `Entry(show="*")` used to hide password input
    - `messagebox.showerror()` for validation

## 9. Sample Output

Input: "Hello World!"

Output (Encrypted): "SGVsbG8gV29ybGQh"

Intput : "SGVsbG8gV29ybGQh"

Output (Decrypted): "Hello World!"

## 10. Applications & Use Cases

Educational tool to learn basic encoding

- Embedded in chat applications for message hiding
- Introduce beginners to GUI programming and event handling
- Demonstrate concept of symmetric encryption flow

## 11. Limitations & Scope

***Limitations:***

- Password is hardcoded
- Base64 is not a secure encryption method
- No real cryptographic hashing or key exchange
- No file I/O support or message export

***Future Enhancements:***

- Replace Base64 with AES or RSA for strong encryption
- Add GUI improvements with images, themes
- Store password securely using hash functions
- Enable saving encrypted messages
- Add a login system or multi-user support

## 12. Conclusion

This mini project successfully demonstrates the basics of **encryption and decryption using Base64** with a GUI interface in Python. It strengthens beginner understanding of encoding, GUI building, and input validation while laying a foundation for more complex cryptographic systems in the future. Though simplistic, it showcases core software concepts in a visual and interactive way.