**Project Report**



**information security**

Submitted by: **Shaista Bibi**

Roll No.: **fall\_20\_BSAI\_0098**

Class Section: **BSAI**

Submitted to: **Ms.Laraib javad**

DATE: 01/25/2024

“On my honor, as a student of Sir Syed CASE Institute Islamabad, I have neither given nor received unauthorized assistance on this academic work.”

**Project Task:** Combine AES encryption with steganography techniques to hide encrypted messages within images or other media.

Steps to Follow:

|  |
| --- |
| **Step 1: User Interface (UI) Design.**  - Design a simple program with buttons and input fields.  - Allow users to input their secret messages, choose images, and set passwords for encryption.     1. Import Required Modules:    * Import the necessary modules, such as Tkinter, filedialog, messagebox, functools, and any other libraries. 2. Define Encryption Function:    * An “encrypt\_message” when the user clicks the "Encrypt" button. 3. Input Fields and Labels:    * Create labels and entries for users to input their secret messages passwords. 4. Encryption Button:  * Add a button that, when clicked, the encryption process.  1. Image Selection:    * Allow users to choose an image for encryption. 2. Result Label:   **Output:** |

|  |
| --- |
| **Step 2: AES Encryption Implementation.**  - Implement the AES encryption process using the cryptography library in Python.  - Take a user's message and convert it into a secret code using a key (password).   * import necessary modules from the backend, including PBKDF2HMAC, hashes, Cipher, algorithms, modes, default\_backend, os, and base64. * Define a function (derive\_key) to derive a key from the password and salt using PBKDF2HMAC. * implement an encryption function (encrypt\_aes) using AES with CFB8 mode. * Use the encryption function to encrypt the user's message, obtaining the encrypted message.   **Output:** |

|  |
| --- |
| **Step 3: Steganography Implementation.**  - Develop the steganography part using the Python PIL (Pillow) library for image manipulation.  - Embed the encrypted message into the pixels of an image without making the image look weird.   * Import the required modules including tkinter, filedialog, messagebox, Image from PIL, and your backend functions. * Create a function (encrypt\_and\_hide) to handle both encryption with AES and hiding the encrypted message in an image. * Implement a function (choose\_image) to allow users to select an image for steganography. * Design a GUI with labels, entry widgets, and buttons for users to input a secret message, password, and choose an image. * Use Label for displaying instructions, Entry for input fields, and Button for actions. * Button Actions: Connect the "Encrypt and Hide" button to the encrypt\_and\_hide function.   **Output:** |

|  |
| --- |
| **Step 4: Decryption Process.**  - Create a way for the recipient to reverse the process.  - Decrypt the message from the image using the key without anyone else knowing what was hidden.   * import Modules and necessary libraries: * Define Encryption and Steganography Functions: Create functions (encrypt\_and\_hide, decrypt\_and\_show) to handle encryption with AES, steganography, and decryption. * Implement functions (choose\_image, choose\_stego\_image) for allowing users to choose images. * User Interface Setup for Encryption: Design input fields and buttons for users to enter a secret message, password, and choose an image for encryption. * Include a button for the encryption and steganography process. * User Interface Setup for Decryption.   **Output:** |