PLAYFAIR SIMULATION AND STEGANOGRAPHY

A course project report submitted in partial fulfilment of the requirement

Of

CRYPTOGRAPHY AND NETWORK SECURITY

By

S.ABHINAYA	**(2203A51520)**
D.VISHWA VARDHANI	**(2203A51172)**
S.HIMABINDU	**(2203A51391)**
N.SAI SATHWIK	**(2203A51499)**
B.SUMANTH	**(2203A51551)**

Under the guidance of

Dr.M.Ranjeeth Kumar

Asst. Professor, Department of CSE

sru

SR UNIVERSITY

Abstract

In modern communication systems, ensuring data privacy and confidentiality is extremely important. This project focuses on implementing two different security techniques—Playfair Cipher for text encryption and LSB (Least Significant Bit) steganography for image encryption—to provide a dual layer of security for both textual and visual data. The Playfair Cipher, a classical symmetric key encryption method, encrypts text in letter pairs and adds complexity over simple substitution ciphers. Separately, the project uses LSB steganography to hide a message or another image inside a cover image, making the hidden data undetectable to human eyes. By implementing both techniques independently, this project demonstrates the effectiveness of classical cryptography for textual data and digital steganography for image-based data. This approach helps learners understand and apply both security techniques and shows how they can be used to protect different types of information.

Introduction

As digital communication continues to grow, so does the need to protect both textual and image data from unauthorized access. Attackers often target sensitive text and images shared over networks, which can lead to data breaches and misuse of personal or confidential information. To address these concerns, various techniques have been developed in the field of information security.This project implements two separate techniques to secure text and image data:

1. Text Encryption using Playfair Cipher:

The Playfair Cipher is one of the earliest techniques in symmetric key cryptography. Unlike traditional monoalphabetic ciphers, it encrypts text in digraphs (pairs of letters) using a 5x5 matrix formed from a keyword. This makes pattern recognition difficult and increases the complexity of decryption without the key. It's a simple yet effective method for protecting textual information.

2. Image Encryption using Steganography (LSB Method):

Steganography goes beyond encryption by concealing the very existence of data. In this project, we apply LSB steganography to hide information within an image file. The LSB technique works by modifying the least significant bits of pixel values to embed hidden data. Since these changes are minimal, the human eye cannot detect any difference in the image. This ensures the confidentiality of the hidden data and makes it difficult for intruders to even know that sensitive data exists inside the image.

By separately encrypting text using Playfair Cipher and hiding image-based data using steganography, this project provides a practical and easy-to-understand demonstration of two essential techniques in cybersecurity. It is especially useful for academic learning and basic implementation of secure communication methods.

Objective

"To implement Playfair Cipher for text encryption and LSB steganography for separate image encryption to ensure secure and invisible communication of data."

Explanation:

The main goal of this project is to secure two different types of data—text and images—using two different but effective methods:

- Text encryption is done using the Playfair Cipher, which is a classical encryption technique. It scrambles the plaintext into unreadable ciphertext using a 5x5 matrix based on a keyword. This makes it hard for an attacker to guess the original message.
- Image encryption (or rather, data hiding in images) is done using LSB steganography, which hides data inside the pixels of an image. Even though the image looks the same to the human eye, it actually contains secret data inside it.

By achieving this objective, the project demonstrates how to handle the security of both textual and visual data independently, using two well-known but different approaches from the field of cybersecurity.

Comparison Table: Previous vs Proposed Methodology

Sl.No	Criteria	Previous Methodology	Proposed Methodoogy
1	Encryption Scope	Mostly focused on either text or image encryption separately	Handles text and image encryption as separate modules using different techniques
2	Text Encryption Method	Simple substitution ciphers (e.g., Caesar, monoalphabetic)	Playfair Cipher (digraph-based, more secure)
3	Image Security Method	None or basic image masking	LSB steganography (hides data inside image pixels without visible changes)
4	Security Level	Moderate (relies on one layer of protection)	High (uses separate encryption + data hiding for different data types)
5	Detection Possibility	Encrypted data is visible and may raise suspicion	Hidden data is not noticeable; image looks the same even with hidden message
6	Data Protection Technique	Encryption only (focus on making data unreadable)	Combination of encryption (Playfair) and steganography (LSB) for better confidentiality
7	Complexity	Simpler to implement but easier to break	Slightly more complex but more secure and harder to detect or decrypt
8	Use Case Suityability	Suitable for basic text protection	Suitable for modern secure communication involving text and images

Tools Used to Implement Project

1. Software Tools :

- ◆HTML
- ◆CSS
- ◆JavaScript
- ◆Web Browser
- ◆Code Editor
- ◆Operating System

2. Hardware Tools :

- ◆ Laptop/Desktop
- ◆ Display Monitor
- ◆ Mouse & Keyboard
- ◆ Storage

➢ The software tools used in this project are detailed for the use individually, as follows:

I. HTML is used to create and interface.

II. CSS is used for the styling purpose of the interface.

III. JavaScript is the part where we write the core logic for the playfair cipher in the encryption of text as well as image.

IV. Web browser like chrome, microsoft edge, etc., are used to run the html file which results an interactive user interface.

V. Code Editor is meant to edit the codes only if any errors appear in the playfair logic or if the interface cannot run.

VI. An OS like windows or linux is used which supports the modern browser like chrome.

Proposed Methodology

The proposed system focuses on providing security to both text and image data using two different cryptographic techniques implemented on the web using HTML, CSS, and JavaScript. The entire process is divided into two independent modules:

1. Text Encryption using Playfair Cipher:

The first module handles encryption of text messages using the Playfair Cipher, a classical cryptographic technique that works on digraphs (letter pairs). This is implemented using JavaScript to simulate how the cipher matrix is created and how encryption is done.

Steps Involved:

Step 1: Input Key and Plaintext

The user provides a keyword (used to generate the cipher matrix) and the plaintext message they want to encrypt.

Step 2: Generate 5x5 Cipher Matrix

A 5x5 grid is constructed using the keyword, placing each unique letter only once and filling the remaining slots with unused letters of the alphabet (I and J are treated as the same character).

Step 3: Prepare the Plaintext

The plaintext is divided into pairs of letters. If a pair contains the same letter, an 'X' is inserted between them. If the text length is odd, an 'X' is added at the end.

Step 4: Encrypt Using Rules

✓ If both letters are in the same row: Replace each with the letter to its right.

✓ If both letters are in the same column: Replace each with the letter below it.

✓ If neither: Replace each with the letter in the same row but in the column of the other letter.

Step 5: Display Encrypted Text

The output ciphertext is shown in a user-friendly format. Optionally, the same algorithm can be used in reverse to decrypt the ciphertext back to plaintext.

2. Image Encryption using LSB Steganography:

The second module focuses on hiding data (either a message or a small image) within another image using the Least Significant Bit (LSB) technique. It uses JavaScript's Canvas API and FileReader API for direct manipulation of image pixel data in the browser.

Steps Involved:

Step 1: Upload Cover Image

The user uploads an image (cover image) that will be encrypted.

Step 2: Input Encryption Key

The user provides a secret key, which will be used for both encryption and decryption.

Step 3: Encryption of the Image

The image is processed pixel by pixel. The encryption process involves converting the image into a binary representation, and then manipulating the pixel data using the key. The key is applied to each pixel's color channels (RGB values) to transform the image into an encrypted form.

Step 4: Display the Encrypted Image

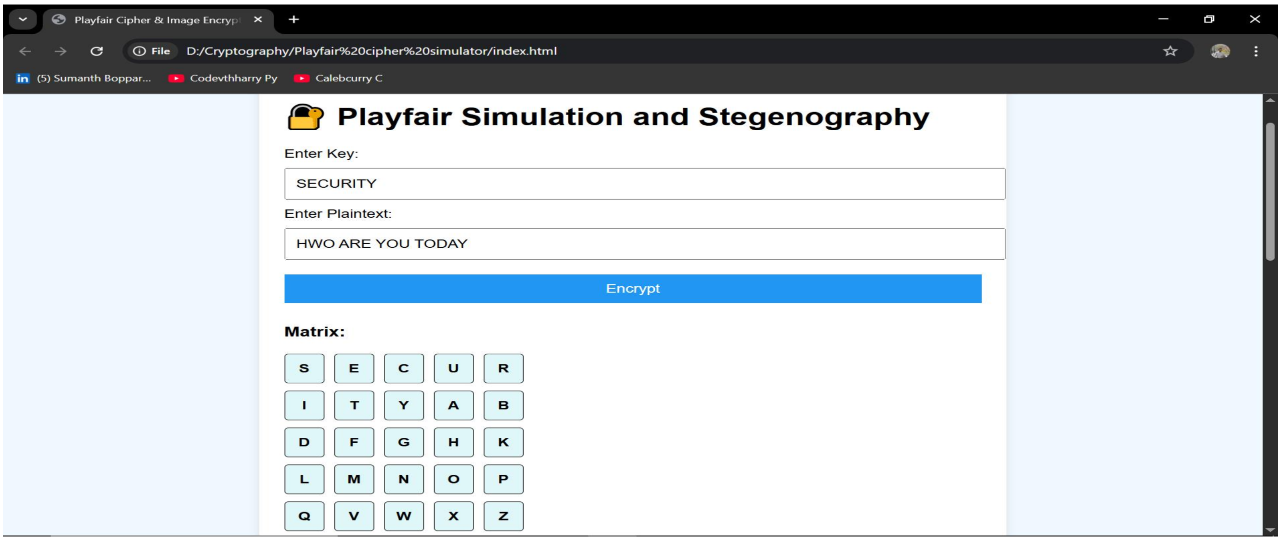
The encrypted image is displayed for the user.

Step 5: Decryption Process

To decrypt the image, the user must input the same encryption key. If the correct key is provided, the image is decrypted and restored to its original form. If the key is incorrect, the decryption will fail, and the image will not be restored.

This process ensures that the image can only be decrypted by the correct user with the proper key, adding an extra layer of security to the image data.

Output (Screenshots)


Playfair Cipher & Image Encryp
File
D:/Cryptography/Playfair%20cipher%20simulator/index.html
(5) Sumanth Boppar...
Codevthharry Py
Calebcurry C

Playfair Simulation and Stegenography

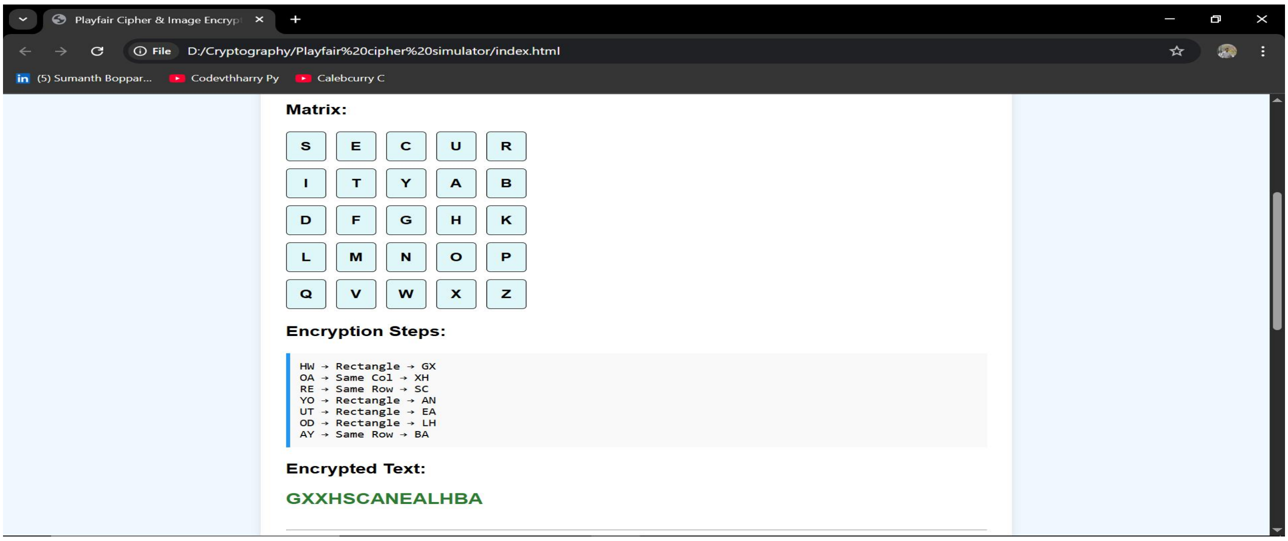
Enter Key: SECURITY

Enter Plaintext: HWO ARE YOU TODAY

Encrypt

Matrix:

S	E	C	U	R
I	T	Y	A	B
D	F	G	H	K
L	M	N	O	P
Q	V	W	X	Z


Playfair Cipher & Image Encryp
File
D:/Cryptography/Playfair%20cipher%20simulator/index.html
(5) Sumanth Boppar...
Codevthharry Py
Calebcurry C

Matrix:

S	E	C	U	R
I	T	Y	A	B
D	F	G	H	K
L	M	N	O	P
Q	V	W	X	Z

Encryption Steps:

```
HW → Rectangle → GX
OA → Same Col → XH
RE → Same Row → SC
YO → Rectangle → AN
UT → Rectangle → EA
OD → Rectangle → LH
AY → Same Row → BA
```

Encrypted Text:

GXXHSCANEALHBA

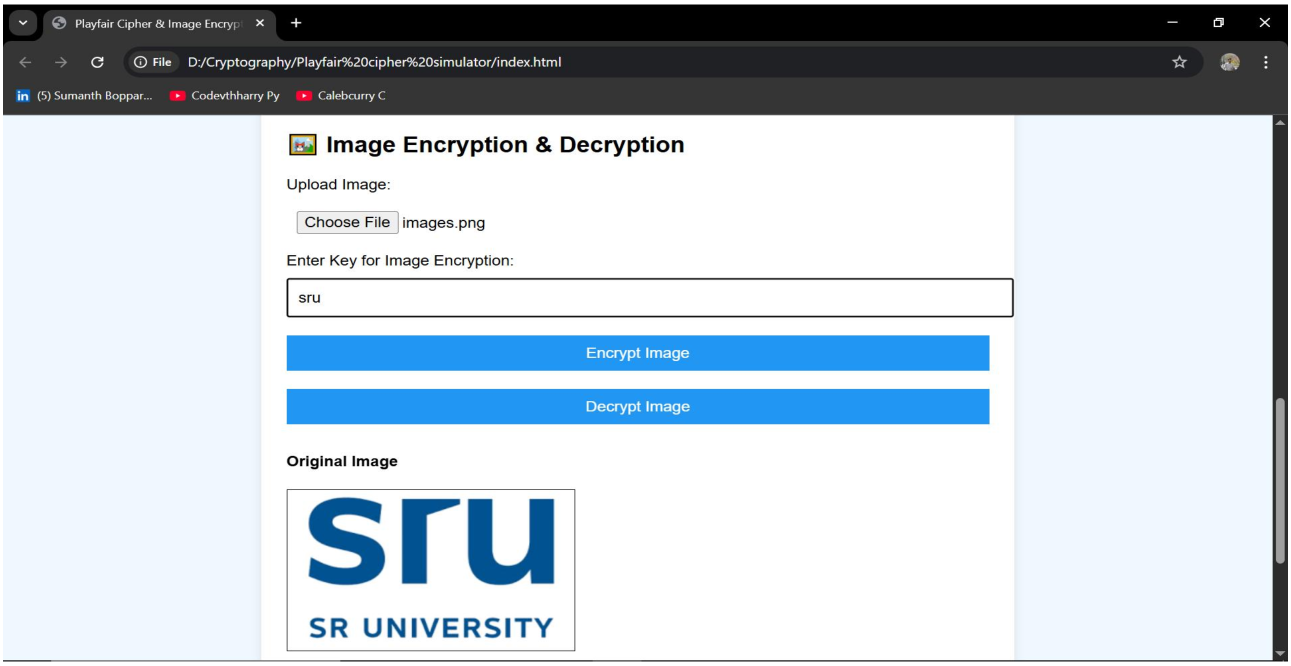

Playfair Cipher & Image Encryp
File
D:/Cryptography/Playfair%20cipher%20simulator/index.html
(5) Sumanth Boppar...
Codevthharry Py
Calebcurry C

Image Encryption & Decryption

Upload Image:

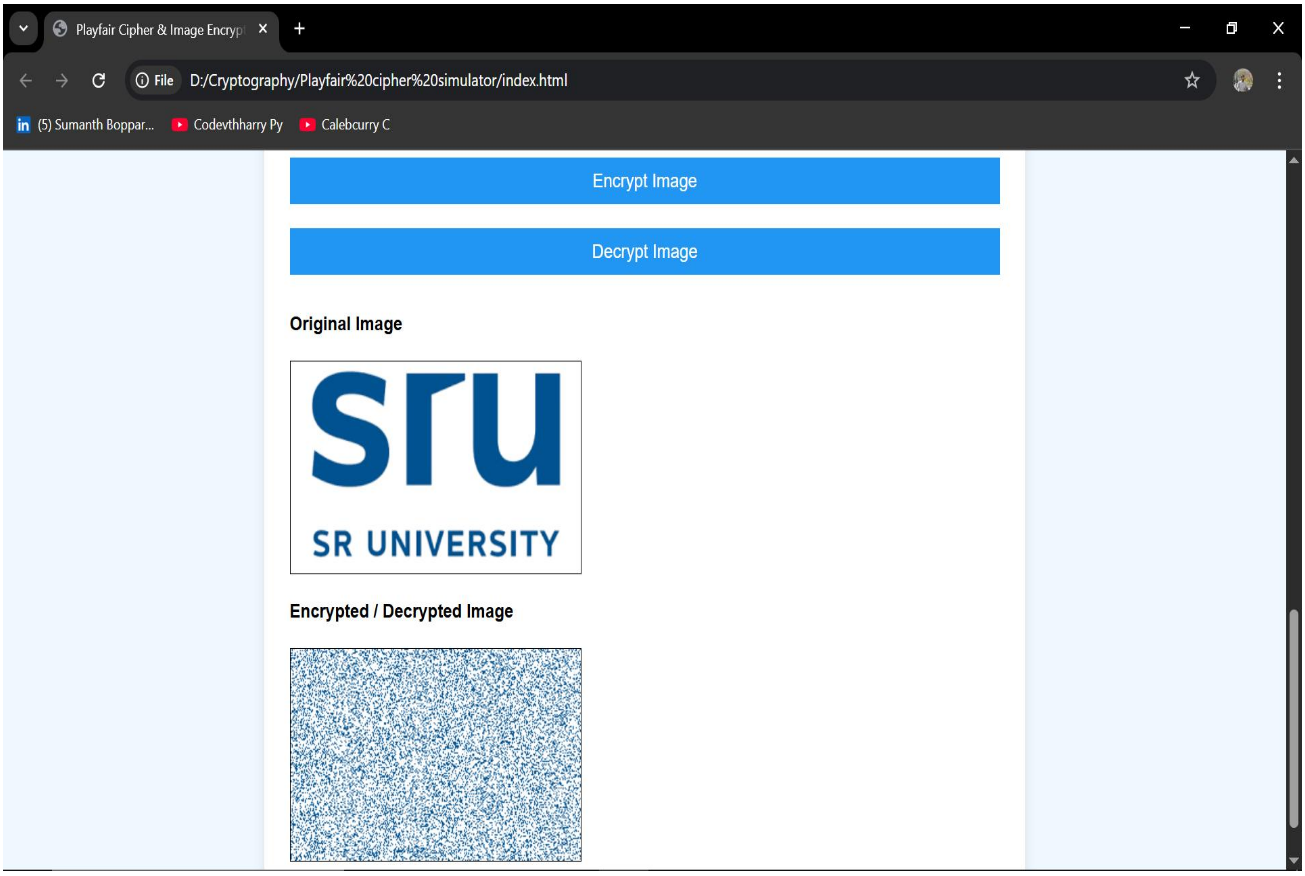
Choose File images.png

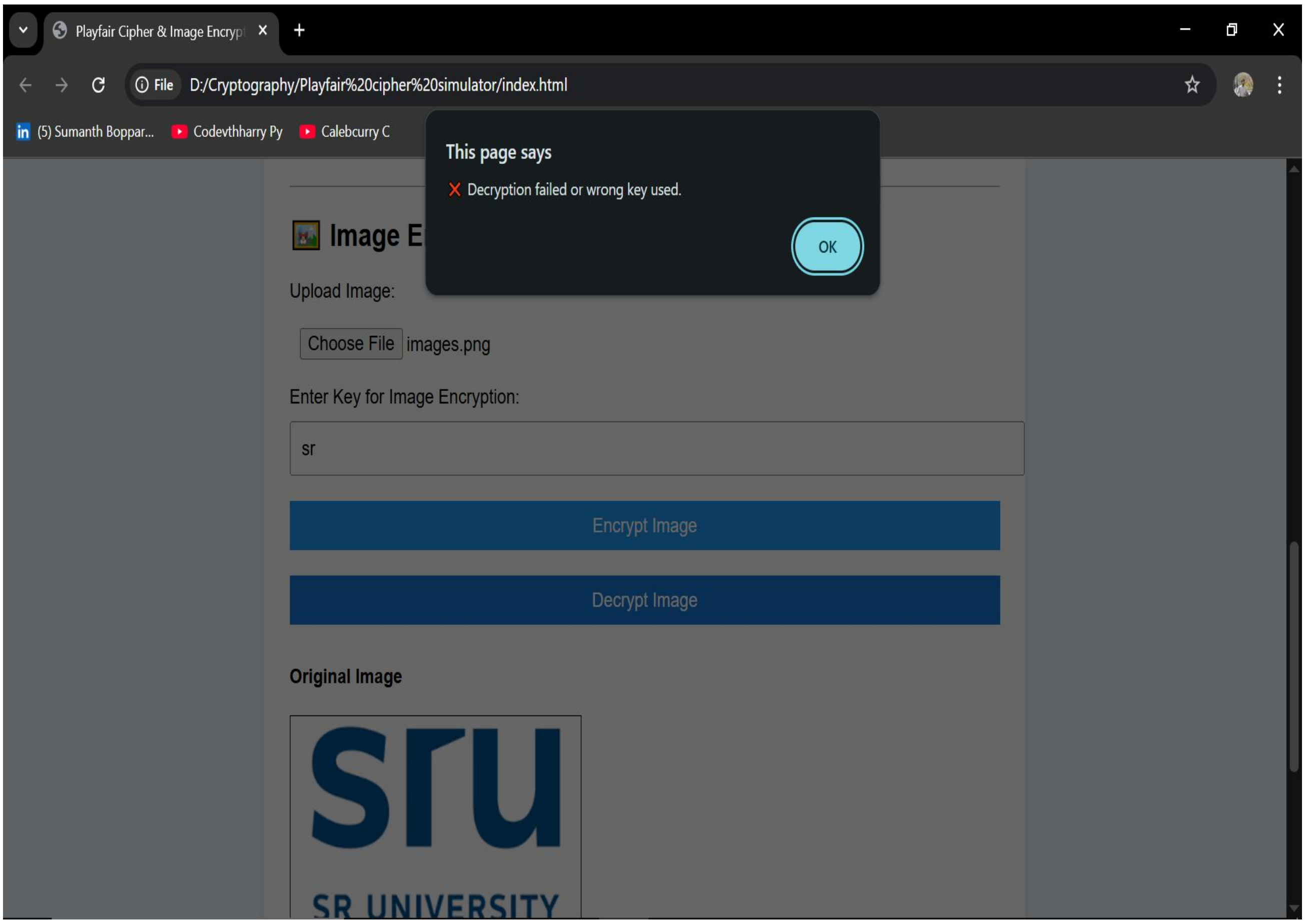
Enter Key for Image Encryption: sru

Encrypt Image

Decrypt Image

Original Image


Playfair Cipher & Image Encrypt
D:/Cryptography/Playfair%20cipher%20simulator/index.html
(5) Sumanth Boppar...
Codevthharry Py
Calebcurry C
Encrypt Image
Decrypt Image
Original Image
SR UNIVERSITY
Encrypted / Decrypted Image

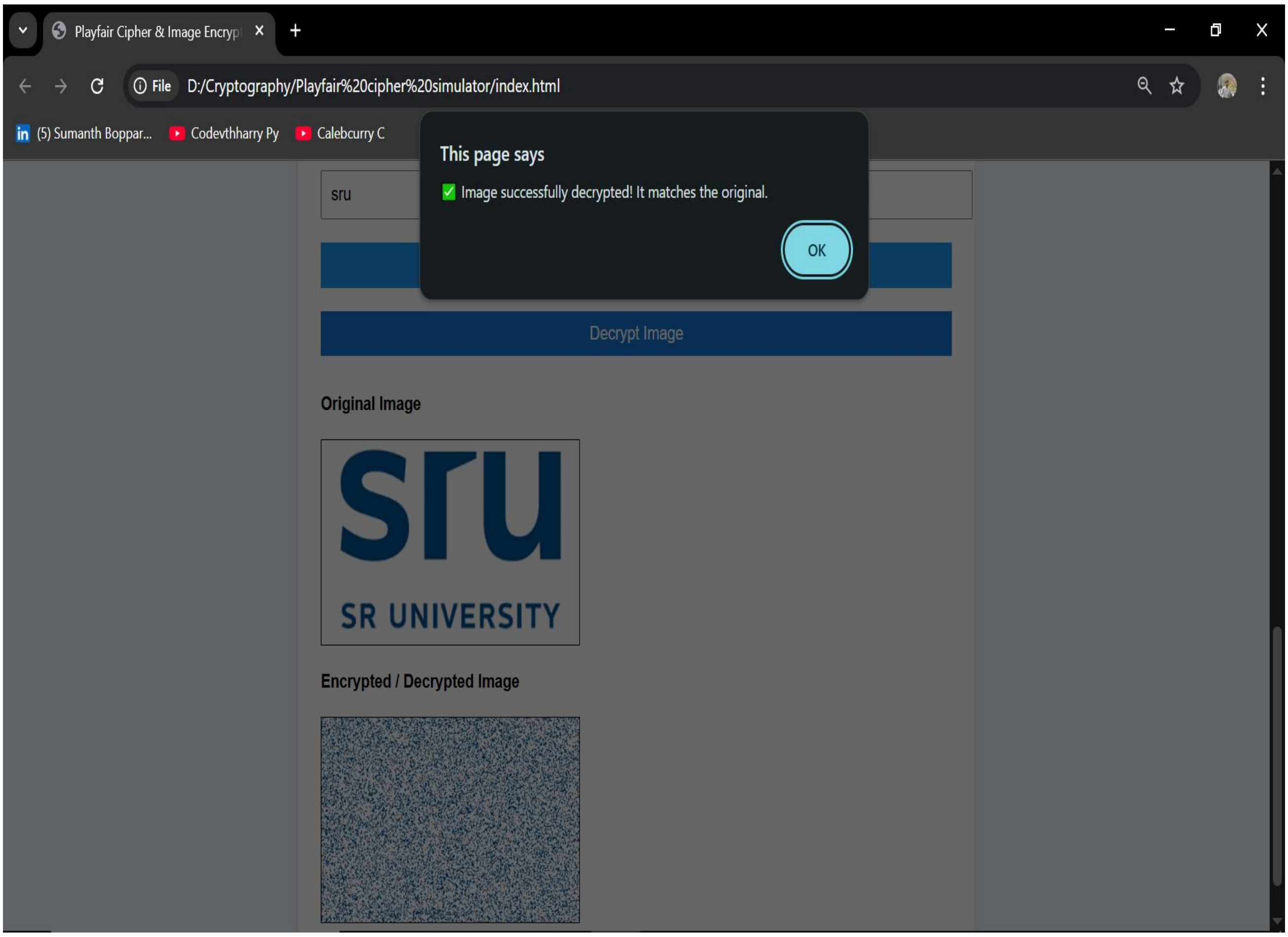


This page says

✗ Decryption failed or wrong key used.

Upload Image: images.png

Enter Key for Image Encryption: sr



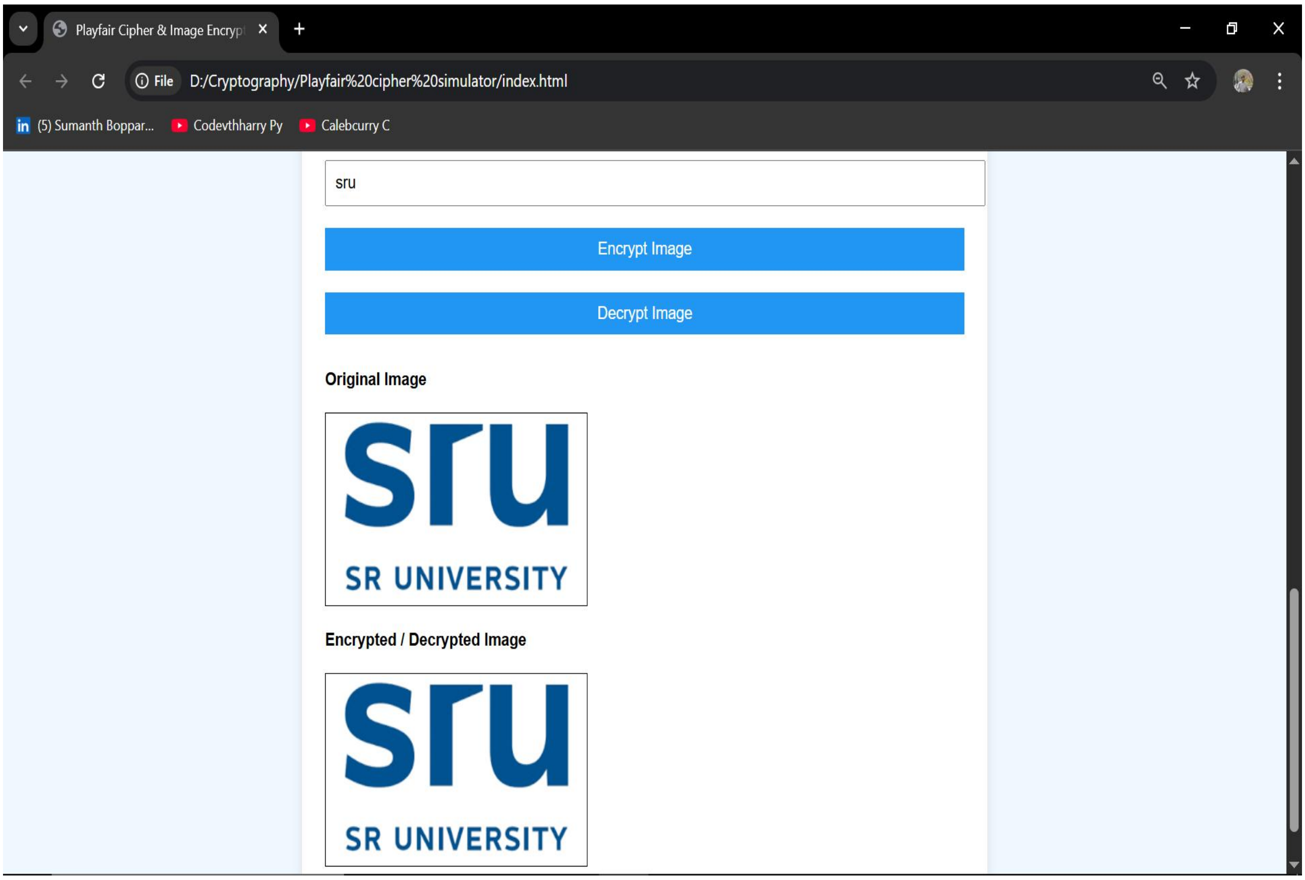
Playfair Cipher & Image Encrypt
D:/Cryptography/Playfair%20cipher%20simulator/index.html
(5) Sumanth Boppar...
Codevthharry Py
Calebcurry C
sru
OK
Decrypt Image

This page says

✅ Image successfully decrypted! It matches the original.

Original Image

Encrypted / Decrypted Image



Playfair Cipher & Image Encrypt
D:/Cryptography/Playfair%20cipher%20simulator/index.html
(5) Sumanth Boppar...
Codevthharry Py
Calebcurry C
Encrypt Image
Decrypt Image

sru

Original Image

Encrypted / Decrypted Image

Conclusion

In conclusion, this project successfully demonstrates the use of cryptographic techniques to secure both text and image data. The Playfair Cipher was used for text encryption, converting the message into digraphs and applying a set of rules to encrypt it. This method ensures that the text remains secure and difficult to decode without the correct key.

For image encryption, the project allows the user to upload an image, provide a key, and encrypt it. The image is modified at the pixel level using the key, and it can only be decrypted back to its original form with the same key. If the key is incorrect, decryption is not possible, ensuring the security of the image.

The entire project is built using HTML, CSS, and JavaScript, making it fully client-side and easily accessible through a web browser. This approach demonstrates how different encryption techniques can be applied to both text and images, offering a simple yet secure way to protect data.

References

[1] Stallings, W. (2017). Cryptography and Network Security: Principles and Practice (7th ed.).

[2] Schneier, B. (1996). Applied Cryptography: Protocols, Algorithms, and Source Code in C.

[3] N. Provos and P. Honeyman, "Hide and Seek: An Introduction to Steganography," IEEE Security & Privacy, vol. 1, no. 3, 2003.

[4] Python Imaging Library (PIL) Documentation – https://pillow.readthedocs.io

[5] PyCryptodome – https://www.pycryptodome.org/

[6] OpenCV Python Tutorials – https://docs.opencv.org/

[7] Wikipedia – Playfair Cipher: https://en.wikipedia.org/wiki/Playfair_cipher

[8]Wikipedia–LSBSteganography:
https://en.wikipedia.org/wiki/Steganography#Digital_steganography