Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Spring, Year: 2024), B.Sc. in CSE (Day)

Playfair Cipher Encryption Tools

Course Title: Computer and Cyber Security
Course Code: CSE 323
Section: 212 D4

Students Details

Name	ID
Md.Manzurul Alam	202902003

Submission Date: 15/05/2024
Course Teacher's Name: Fairuz Shaiara

[For teachers use only: Don't write anything inside this box]

Lab Project Status	
Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Playfair Cipher Encryption Tools

1.1 Introduction

The Playfair Cipher Encryption Tool is a web-based application designed to implement the Playfair Cipher, a classical encryption technique. Users can input a keyword and a plaintext message, and the tool will generate the corresponding ciphertext using the Playfair Cipher algorithm. Additionally, it provides functionality to decrypt ciphertext back into its original plaintext using the same keyword. The application features a user-friendly interface with input fields for the keyword and message, buttons for encryption and decryption, and areas to display the results. Developed with educational purposes in mind, this tool serves as a practical resource for understanding classical cryptography principles and gaining hands-on experience with encryption techniques.

1.2 Project Interface

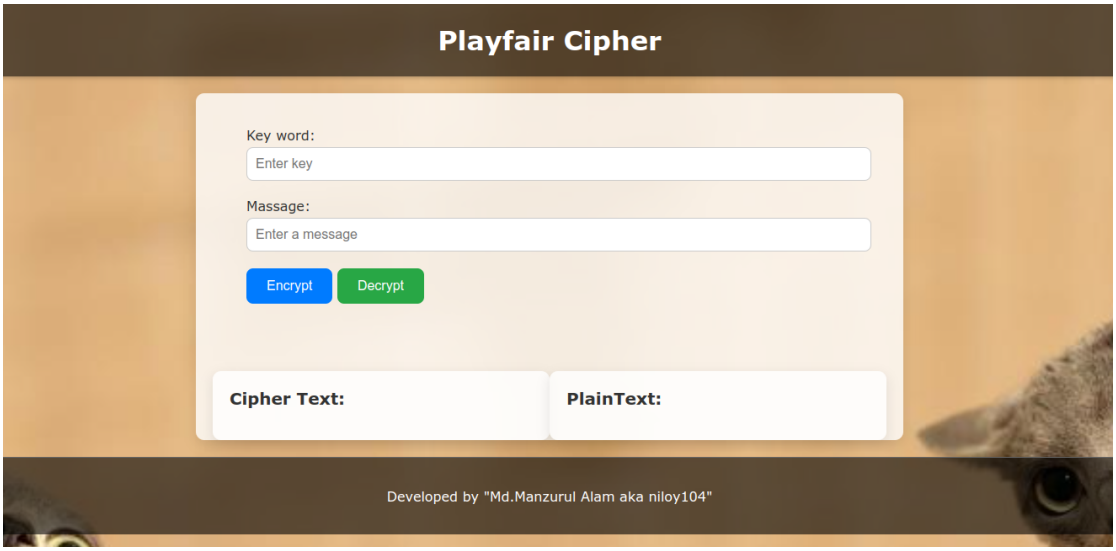

Figure 1.1: Project Interface of Playfair Encryption Tools

1.3 Implementation

1.3.1 Project Details

The Playfair Cipher Encryption Tool is a web application that implements the Playfair cipher algorithm to encrypt and decrypt messages. Let's delve into a detailed analysis of its functionality:

1.3.2 HTML Structure

- **Header**: The header section displays the title "Playfair Cipher" in a centered format.
- **Main Container**: This section contains two main parts: the form container and the result container.
  - **Form Container**: Houses the input fields for the keyword and message, as well as buttons for encryption and decryption.
  - **Result Container**: Displays the encrypted and decrypted messages in separate sections.
- **Footer**: Provides attribution to the developer.

1.3.3 CSS Styling

- **Global Styling**: Sets the background image, font family, and basic layout properties for the entire document.
- **Header Styling**: Styles the header with a semi-transparent background color, centered text, and box shadow for a visually appealing appearance.
- **Container Styling**: Defines the appearance of the main container with a maximum width, rounded corners, and a subtle box shadow for a modern look.
- **Form Styling**: Styles form elements such as labels, input fields, and buttons with consistent padding, border radius, and transition effects for user interaction.
- **Result Container Styling**: Formats the result display area with flexbox to arrange the encrypted and decrypted messages side by side.
- **Footer Styling**: Defines the appearance of the footer with a semi-transparent background, centered text, and a border separating it from the main content.

1.3.4 JavaScript Functions

- **getKeyword**(): Retrieves the keyword entered by the user, removes any spaces, and replaces 'J' with 'I' to conform to Playfair cipher rules.

- **getString()**: Retrieves the message entered by the user.
- **processKey()**: Processes the keyword to generate the Playfair cipher key matrix. It ensures that each letter is used only once and fills the matrix with the remaining letters of the alphabet after the keyword.
- **cipher()**: Encrypts the message using the Playfair cipher algorithm. It handles repeated letters, padding, and determining the coordinates of each letter in the key matrix to perform the encryption.
- **deCodeCipher()**: Decrypts the encrypted message back to plaintext. It reverses the encryption process by finding the coordinates of each letter in the key matrix and reconstructing the original message.

1.3.5 Workflow

1. **Input**: Users enter a keyword and a message in the provided input fields.
2. **Encryption**: Upon clicking the "Encrypt" button, the message is encrypted using the Playfair cipher algorithm.
3. **Display**: The encrypted message is displayed in the "Cipher Text" section.
4. **Decryption**: Clicking the "Decrypt" button invokes the decryption process, converting the encrypted message back to plaintext.
5. **Display**: The decrypted plaintext is displayed in the "PlainText" section.

1.4 Conclusion

The Playfair Cipher Encryption Tool offers a comprehensive solution for encrypting and decrypting messages using the Playfair cipher. With a user-friendly interface and robust functionality, it provides a secure method for communication while maintaining ease of use and accessibility.

Chapter 2

Conclusion

2.1 Discussion

The implementation of the Playfair Cipher web application focused on delivering a seamless user experience through intuitive user interface design, robust encryption and decryption functionalities, and visually appealing styling. The user interface was thoughtfully designed with input fields for the key and text, along with clear buttons for encryption or decryption operations. Utilizing the Bootstrap framework ensured responsiveness and consistency across various devices, while form validation enhanced usability by prompting users for valid inputs. JavaScript functions handled the encryption and decryption processes, dynamically updating the result containers on the webpage through DOM manipulation. Custom CSS styles were applied to create an aesthetically pleasing layout, prioritizing readability and usability. Additionally, the inclusion of a footer message acknowledging the developer's contribution added a personal touch to the project. Accessibility considerations were also taken into account, ensuring that the web application is usable by individuals with disabilities. Overall, the Playfair Cipher web application successfully combines functionality, usability, and design to provide an effective cryptographic tool for users.

2.2 Limitations

The Playfair Cipher Encryption Tools has several limitations:

1. Limited security compared to modern algorithms.
2. Key length restriction to 25 characters.
3. Lack of comprehensive error handling.
4. Potential performance issues with large text volumes.
5. Dependency on client-side processing with JavaScript.
6. Absence of advanced features like key management.

7. Limited browser support and compatibility.
8. Incompatibility with legacy systems lacking modern web technologies.

2.3 Scope of Future Work

Future work for the Playfair Cipher web application could involve enhancing security with stronger cryptographic algorithms, improving the user interface for better usability, expanding functionality with support for additional cryptographic methods, and conducting thorough testing for reliability and compatibility across different platforms.

References

1. Playfair Cipher with Examples. Retrieved from https://www.geeksforgeeks.org/playfair-cipher-with-examples/
2. Responsive Web Design. Retrieved from https://www.w3schools.com/html/html_responsive.asp
3. Playfair Cipher. Retrieved from https://www.baeldung.com/cs/playfair-cipher