Cipher Master: Playfair and Vigenère Encryption and Decryption

Green University of Bangladesh
Department of Computer Science and Engineering
Semester: Fall 2024, BSc in CSE(DAY)
Section: 213 221 D4

Cipher Master: Playfair and Vigenère Encryption and Decryption

Submitted to,
Md. Sabbir Hosen Mamun
Lecturer
Dept. of CSE
Green University of Bangladesh

Submitted by,
Md Opu Mia 213902036
Md Rabby Khan 213902037
Md.Sabbir Hasan 213902112
Dept. of CSE
Green University of Bangladesh

Contents

Introduction to Cipher Master

Cipher Master is a web-based application designed to demonstrate the Playfair and Vigenère ciphers for encryption and decryption. It provides an interactive platform where users can visualize and understand these classical cryptographic techniques in real-time. The project aims to educate users about cryptography while offering a practical encryption solution for basic data security.

Objective

- Implement Playfair and Vigenère ciphers for encryption and decryption.
- Handle edge cases like duplicates and special characters.
- Create a user-friendly and interactive interface.
- Provide real-time encryption/decryption visualization.
- Ensure cross-platform compatibility.
- Optimize for performance and responsiveness.

Algorithms

Playfair Cipher - Message Preparation and Encryption

- **Generation:**
  - Choose a keyword.
  - Remove duplicate letters.
  - Fill the 5x5 grid with the remaining alphabet letters (treat I/J as one letter).
- **Grid Formation:**
  - Create a 5x5 grid.
  - Fill the grid with the keyword letters first.
  - Follow the remaining unused letters of the alphabet.

Algorithms

Playfair Cipher - Message Preparation and Encryption

- **Message Preparation:**

- Divide the plaintext into digraphs (pairs of letters).
- Insert a filler letter (e.g., 'Z') between identical letters.
- Add a filler letter if the plaintext has an odd number of letters.

Algorithms

Playfair Cipher - Message Preparation and Encryption

- **Letter Transformation (Encryption):**

- **Same Row:** Replace each letter with the letter to its right (wrap around if needed).
- **Same Column:** Replace each letter with the letter below it (wrap around if needed).
- **Different Row and Column:** Replace each letter with the letter in the same row and the other letter's column.

Algorithms

Playfair Cipher - Decryption Algorithm

- **Key Reconstruction:**
  - Recreate the 5x5 grid using the same keyword.

- **Ciphertext Preparation:**
  - Break the ciphertext into digraphs (pairs of letters).

Algorithms

Playfair Cipher - Decryption Algorithm

- **Letter Transformation (Decryption):**
  - **Same Row:** Replace each letter with the letter to its left (wrap around if needed).
  - **Same Column:** Replace each letter with the letter above it (wrap around if needed).
  - **Different Row and Column:** Replace each letter with the letter in the same row and the other letter's column.

Algorithms

Vigenere Cipher - Encryption Algorithm

- **Key Selection:**
  - Choose a keyword.
  - Repeat the keyword to match the length of the plaintext.

- **Encryption Process:**
  - Convert plaintext and keyword into numeric values (A=0, B=1, C=2, ..., Z=25).
  - Use the formula: $E_i = (P_i + K_i) \bmod 26$.
  - Locate the corresponding encrypted letter using the Vigenère table (26x26 grid).

Algorithms

Vigenere Cipher - Decryption Algorithm

- **Decryption Process:**
  - Repeat the key to match the length of the ciphertext.
  - Use the formula: $D_i = (C_i - K_i) \bmod 26$.
  - Convert the numeric result back into letters to retrieve the plaintext.

Tools and Libraries

- **Frontend Tools:** HTML, CSS, and JavaScript for structure, styling, and functionality.
- **Libraries:** Bootstrap for responsive design and Lodash for efficient data manipulation.
- **Development Tools:** Visual Studio Code and Browser DevTools for coding and debugging.
- **Testing Tools:** Unit testing and cross-browser testing to ensure functionality and compatibility.
- **Version Control:** Git and GitHub for managing and sharing source code.
- **Deployment:** Hosted on a web server for seamless access across platforms.

Design and Implementation

EXPLORER

CYBER PROJECT

- Playfair-and-Vigenere-Cip...
  - css
  - image
  - js
    - app.js
    - playfair_dec.js
    - playfair_enc.js
    - vigenere.js
  - index.html
  - README.md

OUTLINE

TIMELINE

CS-SCRIPT - ACTIVE

Welcome | index.html | app.js | playfair_dec.js | playfair_enc.js | vigenere.js

Playfair-and-Vigenere-Cipher-Visualizer-main > Playfair-and-Vigenere-Cipher-Visualizer-main > index.html > ...

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Fira+Code:wght@300..700&display=swap" rel="stylesheet">

    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Rubik+Wet+Paint&display=swap" rel="stylesheet">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">
    <link rel="stylesheet" href="./css/style.css">
    <script src="https://cdn.jsdelivr.net/npm/lodash@4.17.21/lodash.min.js"></script>

    <title>Playfair Cipher</title>
</head>

<body>
    <main>

        <div class="d-flex justify-content-center">
            <select name="" id="algo_select" class="btn btn-secondary btn-lg">
                <option value="PLAYFAIR">Playfair Cipher</option>
                <option value="VIGENERE">Vigenere Cipher</option>
            </select>
        </div>
```

Welcome | index.html | app.js | playfair_dec.js | playfair_enc.js | vigenere.js

Playfair-and-Vigenere-Cipher-Visualizer-main > Playfair-and-Vigenere-Cipher-Visualizer-main > js > app.js > ...

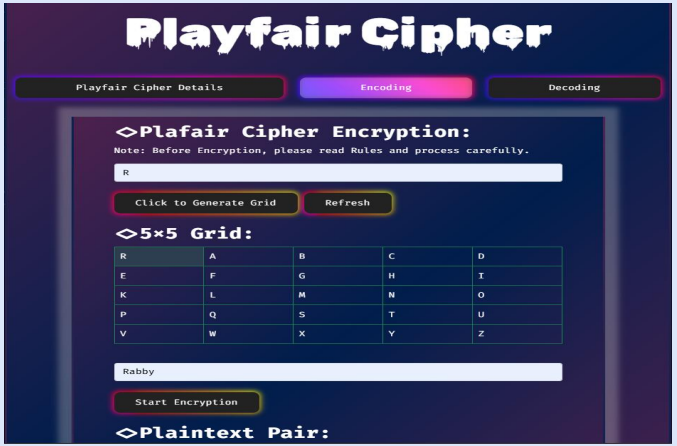
```javascript
const title = document.getElementById("title")
const algo_select = document.getElementById("algo_select")
const playfair = document.getElementById("playfair")
const vigenere = document.getElementById("vigenere")

const valid_alogo = {
    PLAYFAIR : "PLAYFAIR",
    VIGENERE : "VIGENERE"
}

function on_algo_select(event){
    const value = event.target.value;
    if (value === valid_alogo.PLAYFAIR){
        title.innerText="Playfair Cipher"
        playfair.classList.add("d-block")
        vigenere.classList.add("d-none")
        playfair.classList.remove("d-none")
    }
    else if(value === valid_alogo.VIGENERE){
        title.innerText="Vigenere Cipher"
        vigenere.classList.add("d-block")
        playfair.classList.add("d-none")
        vigenere.classList.remove("d-none")
    }
    else{
        alert("Invalid Selection")
    }
}

algo_select.addEventListener("change", on_algo_select)
```


Playfair Cipher
Playfair Cipher
Playfair Cipher Details
Encoding
Decoding
◇Playfair Cipher:
The Playfair cipher was the first practical digraph substitution cipher. The scheme was invented in 1854 by Charles Wheatstone but was named after Lord Playfair who promoted the use of the cipher. In playfair cipher unlike traditional cipher we encrypt a pair of alphabets(digraphs) instead of a single alphabet.
It was used for tactical purposes by British forces in the Second Boer War and in World War I and for the same purpose by the Australians during World War II. This was because Playfair is reasonably fast to use and requires no special equipment.


Playfair Cipher
Playfair Cipher Details
Encoding
Decoding
◇Plafair Cipher Encryption:
Note: Before Encryption, please read Rules and process carefully.
Enter key here ... Ex: ENCRYPTION
Click to Generate Grid
Refresh
◇5×5 Grid:
A B C D E
F G H I/J K
L M N O P
Q R S T U
V W X Y Z


Vigenere Cipher
Vigenere Cipher
Vigenere Cipher Details
Encoding
Decoding
◇Vigenere Cipher:
Vigenère cipher, type of substitution cipher used for data encryption in which the original plaintext structure is somewhat concealed in the ciphertext by using several different The cipher was invented in 1553 by the Italian cryptographer Giovan Battista Bellaso but for centuries was attributed to the 16th-century French cryptographer Blaise de Vigenère, who devised a similar cipher in 1586. A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets. The encryption of the original text is done using the Vigenère square or Vigenère table.


Vigenere Cipher
Vigenere Cipher
Vigenere Cipher Details
Encoding
Decoding
Vigenere Cipher Encryption:
Note: Before Encryption, please read Rules and process carefully.
Enter plaintext here ... Ex: HELLO
Enter key here ... Ex: KEY
Start Encryption
Refresh
0 1 2 3 4 5 6 7 8 9 10 11 12
A B C D E F G H I J K L M

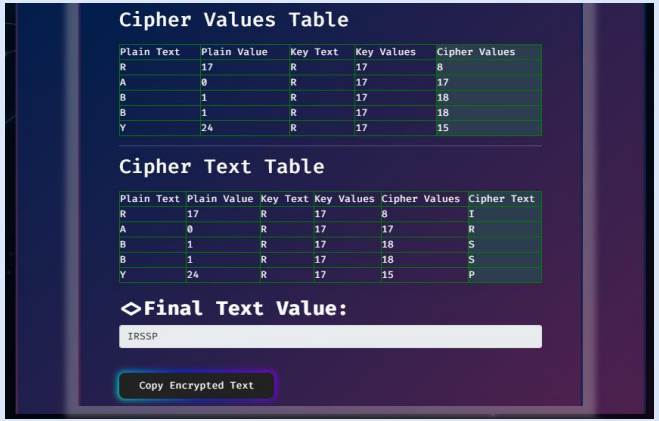

Playfair Cipher
Playfair Cipher Details
Encoding
Decoding
◇Plafair Cipher Encryption:
Note: Before Encryption, please read Rules and process carefully.
R
Click to Generate Grid
Refresh
◇5×5 Grid:
R A B C D
E F G H I
K L M N O
P Q S T U
V W X Y Z
Rabby
Start Encryption
◇Plaintext Pair:


Playfair Cipher
Playfair Cipher
Playfair Cipher Details
Encoding
Decoding
◇Plafair Cipher Decryption:
Note: Before Decryption, please read Rules and process carefully.
R
Click to Generate Grid
Refresh
◇5×5 Grid:
A B C D E
F G H I/J K
L M N O P
Q R S T U
V W X Y Z
QBEWDW
Start Decryption
◇Cipertext Pair:
Q B
E W
D W

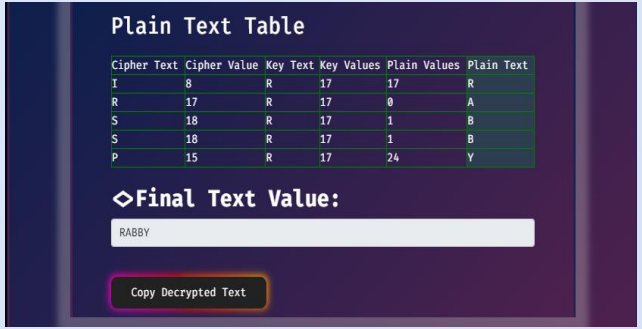

◇Grid For Pair: "BY"
A B C D E
F G H I/J K
L M N O P
Q R S T U
V W X Y Z
◇Cipher For this Pair: "DW"
◇Final Cipher Text:
QBEWDW
Copy Encrypted Text


◇Grid For Pair: "DW"
A B C D E
F G H I/J K
L M N O P
Q R S T U
V W X Y Z
◇Cipher For this Pair: "BY"
◇Final Plain Text:
RABZBY
Copy Decrypted Text


Vigenere Cipher
Vigenere Cipher
Vigenere Cipher Details
Encoding
Decoding
Vigenere Cipher Encryption:
Note: Before Encryption, please read Rules and process carefully.
Rabby
R
Start Encryption
Refresh
0 1 2 3 4 5 6 7 8 9 10 11 12
A B C D E F G H I J K L M
13 14 15 16 17 18 19 20 21 22 23 24 25
N O P Q R S T U V W X Y Z


Cipher Values Table
Plain Text Plain Value Key Text Key Values Cipher Values
R 17 R 17 8
A 0 R 17 17
B 1 R 17 18
B 1 R 17 18
Y 24 R 17 15
Cipher Text Table
Plain Text Plain Value Key Text Key Values Cipher Values Cipher Text
R 17 R 17 8 I
A 0 R 17 17 R
B 1 R 17 18 S
B 1 R 17 18 S
Y 24 R 17 15 P
Final Text Value:
IRSSP
Copy Encrypted Text


Vigenere Cipher
Vigenere Cipher
Vigenere Cipher Details
Encoding
Decoding
Vigenere Cipher Decryption:
Note: Before decryption, please read rules and process carefully.
IRSSP
R
Start Decryption
Refresh
0 1 2 3 4 5 6 7 8 9 10 11 12
A B C D E F G H I J K L M
13 14 15 16 17 18 19 20 21 22 23 24 25
N O P Q R S T U V W X Y Z


Plain Text Table
Cipher Text Cipher Value Key Text Key Values Plain Values Plain Text
I 8 R 17 17 R
R 17 R 17 0 A
S 18 R 17 1 B
S 18 R 17 1 B
P 15 R 17 24 Y
Final Text Value:
RABBY
Copy Decrypted Text

Results and Analysis

- **Demonstration**
  - Real-time encryption and decryption using Playfair and Vigenère ciphers.
  - Dynamic 5x5 grid generation and live feedback for user inputs.
- **Educational Value**

Simplifies complex cryptographic concepts with interactive visualizations.
Helps users understand encryption rules and weaknesses of classical ciphers.

- **Strengths**
  - Accurate implementation of algorithms.
  - Intuitive UI with real-time updates.
  - Effective for learning and basic encryption.
- **Limitations**
  - Vulnerable to cryptanalysis (e.g., frequency analysis, key repetition).
  - Limited scalability for larger inputs or modern encryption needs.

Challenges and Solutions

- **Complex Engineering Problems**
  - Implementing Playfair and Vigenère algorithms with edge-case handling.
  - Ensuring dynamic grid generation and real-time feedback.
- **Balancing Goals**
  - **Historical Accuracy**: Preserving cipher rules while adapting them for usability.
  - **Usability**: Designing an intuitive interface without oversimplifying cryptographic principles.
- **Ensuring Quality**
  - **Accuracy**: Rigorous testing to validate encryption and decryption logic.
  - **Performance**: Optimizing algorithms for real-time interaction.
  - **Interactivity**: Creating responsive visuals for educational value and user engagement.

Future Scope

- **Advanced Algorithms:** Add modern ciphers like AES, RSA, and ECC.
- **UI Enhancement:** Improve interactivity and multi-language support.
- **Large Input Support:** Optimize for bulk data encryption.
- **Mobile App:** Develop for smartphones and tablets.
- **Cryptanalysis Tools:** Add features like frequency analysis.

Any Question?

Thank You So Much !