

A PROJECT REPORT ON

ReverseX Cipher

Course Code: CSE 361

Course Title: Mathematical Analysis for Computer Science

Department: Computer Science & Engineering

Level: 3 & Semester: II

Prepared by

Shovan Prita Paul

Student ID: 2002026

Submitted to

Pankaj Bhowmik

Lecturer

Department of Computer Science & Engineering



HAJEE MOHAMMAD DANESH SCIENCE AND TECHNOLOGY UNIVERSITY, DINAJPUR-5200

ReverseX Cipher

1. Introduction

The "ReverseX Cipher" is a symmetric-key cryptography algorithm employing the XNOR operation for encryption and decryption. It reverses the plaintext before applying XNOR encryption with a designated key. Decryption reverses the process, first applying XNOR decryption and then reversing the ciphertext to recover the original plaintext.

2. Encryption Algorithm

Algorithm Description

1. Input: Plaintext and encryption key.
2. **Reverse the plaintext.**
3. **Perform XNOR encryption** between the reversed plaintext and the key.
4. Output: Encrypted text.

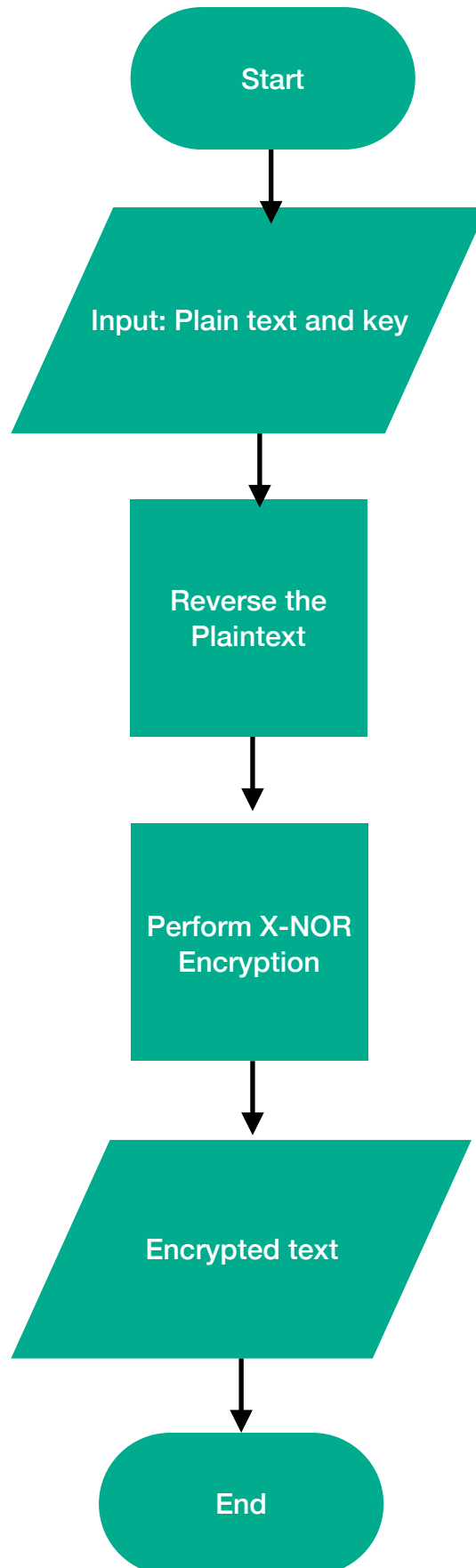
3. Decryption Algorithm

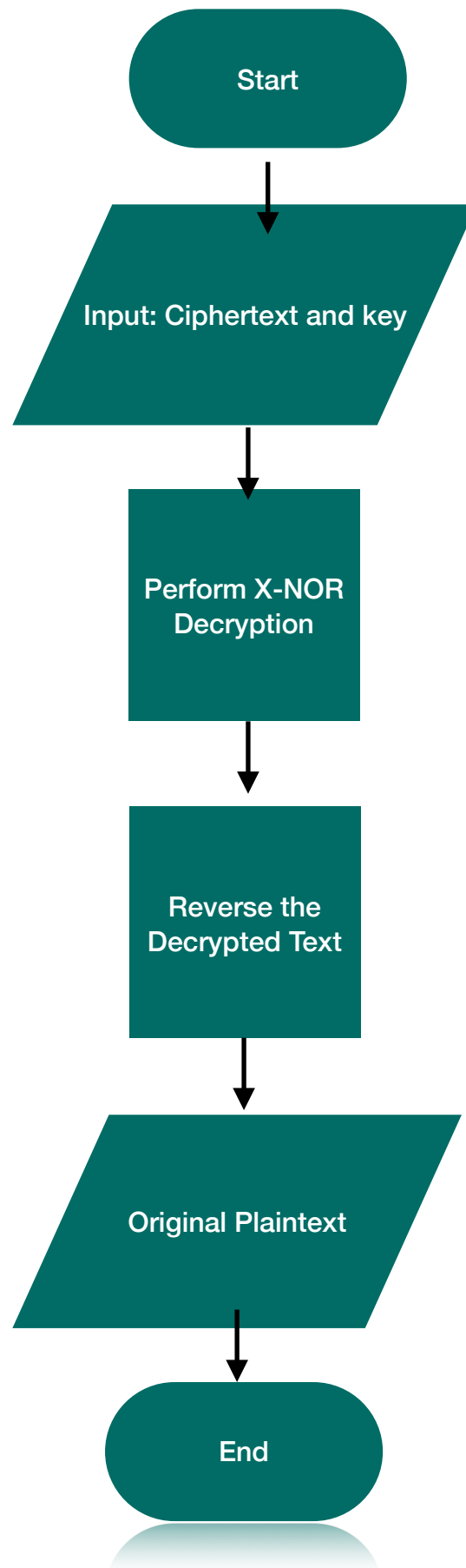
Algorithm Description

1. Input: Ciphertext and decryption key.
2. **Perform XNOR decryption** between the ciphertext and the key.
3. **Reverse the decrypted text** to obtain the original plaintext.
4. Output: Decrypted text.

4. Flowcharts

Encryption Flowchart



Decryption Flowchart

5. Experimental Example

Test Case

Plain Text	Key	Encrypted Text
MESSI	10	Xcbu

Encryption Process

Step	Operation	Explanation	Result
1	Reverse plaintext	Reverse the characters of the plaintext "MESSI".	ISSEM
2	XNOR with key '10'	Perform XNOR operation between each character and the key '10'.	xcbul

Decryption Process

Step	Operation	Explanation	Result
1	XNOR with key '10'	Perform XNOR operation between each character of "xcbul"	and the key '10'
2	Reverse decrypted text	Reverse the characters of the decrypted text "ISSEM".	MESSI

Explanation:

- **Encryption Process:**
 - **Reverse plaintext:** "MESSI" reversed becomes "ISSEM".
 - **XNOR with key '10':** Perform XNOR operation between each character of "ISSEM" and the key '10', resulting in "xcbu".
- **Decryption Process:**
 - **XNOR with key '10':** Perform XNOR operation between each character of "xcbu" and the key '10', reversing the encryption process to retrieve "ISSEM".
 - **Reverse decrypted text:** Reverse "ISSEM" back to "MESSI", recovering the original plaintext.

This table and explanation detail how the encryption and decryption algorithms work step-by-step with the given plaintext "MESSI" and key "10".

6. Source Code

Code for HTML (*Index.html*)

```
<> index.html > ...
1  |<!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  |   <title>Symmetric-Key Encryption Demo</title>
7  |   <link rel="stylesheet" href="styles.css">
8  </head>
9  <body>
10 |   <div class="container">
11 |     <div class="box" id="encryptBox">
12 |       <h2>Encrypt</h2>
13 |       <textarea id="plaintext" placeholder="Enter text to encrypt"></textarea>
14 |       <input type="password" id="encryptionKey" placeholder="Enter encryption key">
15 |       <button onclick="encrypt()">Encrypt</button>
16 |       <textarea id="encrypted" placeholder="Encrypted text will appear here" readonly></textarea>
17 |     </div>
18 |     <div class="box" id="decryptBox">
19 |       <h2>Decrypt</h2>
20 |       <textarea id="ciphertext" placeholder="Enter text to decrypt"></textarea>
21 |       <input type="password" id="decryptionKey" placeholder="Enter decryption key">
22 |       <button onclick="decrypt()">Decrypt</button>
23 |       <textarea id="decrypted" placeholder="Decrypted text will appear here" readonly></textarea>
24 |     </div>
25 |   </div>
26 |
27 |   <script src="script.js"></script>
28 </body>
29 </html>
30
```

Figure 6.1 Source Code for *Index.html*

Code for CSS (*styles.css*)

```
# styles.css > body
1  body {
2      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
3      background-color: #f9f9f9;
4      display: flex;
5      justify-content: center;
6      align-items: center;
7      height: 100vh;
8      margin: 0;
9  }
10
11  .container {
12      display: flex;
13      justify-content: center;
14      align-items: center;
15      width: 100%;
16      height: 100%;
17  }
18
19  .box {
20      background-color: #fff;
21      border-radius: 10px;
22      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
23      padding: 30px;
24      width: 350px;
25      margin: 20px;
26      transition: transform 0.3s ease;
27  }
28
29  .box:hover {
30      transform: translateY(-5px);
31  }
32
33  h2 {
34      font-size: 1.5rem;
35      color: #333;
36      margin-bottom: 20px;
37  }
```

Figure 6.2.1 Source Code for *styles.css*


```
# styles.css > 🔗 textarea
36     margin-bottom: 20px;
37     text-align: center;
38 }
39
40 textarea, input[type="password"] {
41     width: 100%;
42     padding: 10px;
43     margin-bottom: 20px;
44     border: 1px solid #ccc;
45     border-radius: 4px;
46     resize: none;
47     font-size: 0.9rem;
48 }
49 button {
50     background-color: #4CAF50;
51     color: white;
52     padding: 10px 20px;
53     border: none;
54     border-radius: 4px;
55     cursor: pointer;
56     font-size: 0.9rem;
57     transition: background-color 0.3s ease;
58 }
59
60 button:hover {
61     background-color: #45a049;
62 }
63
64 button:active {
65     background-color: #3e8e41;
66 }
67
68 textarea[readonly] {
69     background-color: #f0f0f0;
70     cursor: not-allowed;
71 }
72
```

Figure 6.2.2 Source Code for *styles.css*

Source Code for JAVA SCRIPT (*script.js*)

```
JS script.js > encrypt
1  function encrypt() {
2      let plaintext = document.getElementById('plaintext').value;
3      let key = document.getElementById('encryptionKey').value;
4
5      if (!plaintext || !key) {
6          alert("Please enter both text and encryption key.");
7          return;
8      }
9
10     // Reverse the plaintext
11     let reversedPlaintext = reverseString(plaintext);
12
13     // Perform XNOR operation with the reversed plaintext and key
14     let encryptedText = performXNOREncryption(reversedPlaintext, key);
15
16     document.getElementById('encrypted').value = encryptedText;
17 }
18
19 function decrypt() {
20     let ciphertext = document.getElementById('ciphertext').value;
21     let key = document.getElementById('decryptionKey').value;
22
23     if (!ciphertext || !key) {
24         alert("Please enter both text and decryption key.");
25         return;
26     }
27
28     // Perform XNOR operation with the ciphertext and key
29     let decryptedText = performXNOREncryption(ciphertext, key);
30
31     // Reverse the decrypted text to get the original plaintext
32     let originalPlaintext = reverseString(decryptedText);
33
34     document.getElementById('decrypted').value = originalPlaintext;
35 }
36
37
```

Ln 1, Col 1 Spaces

Figure 6.3.1 Source Code for *script.js*

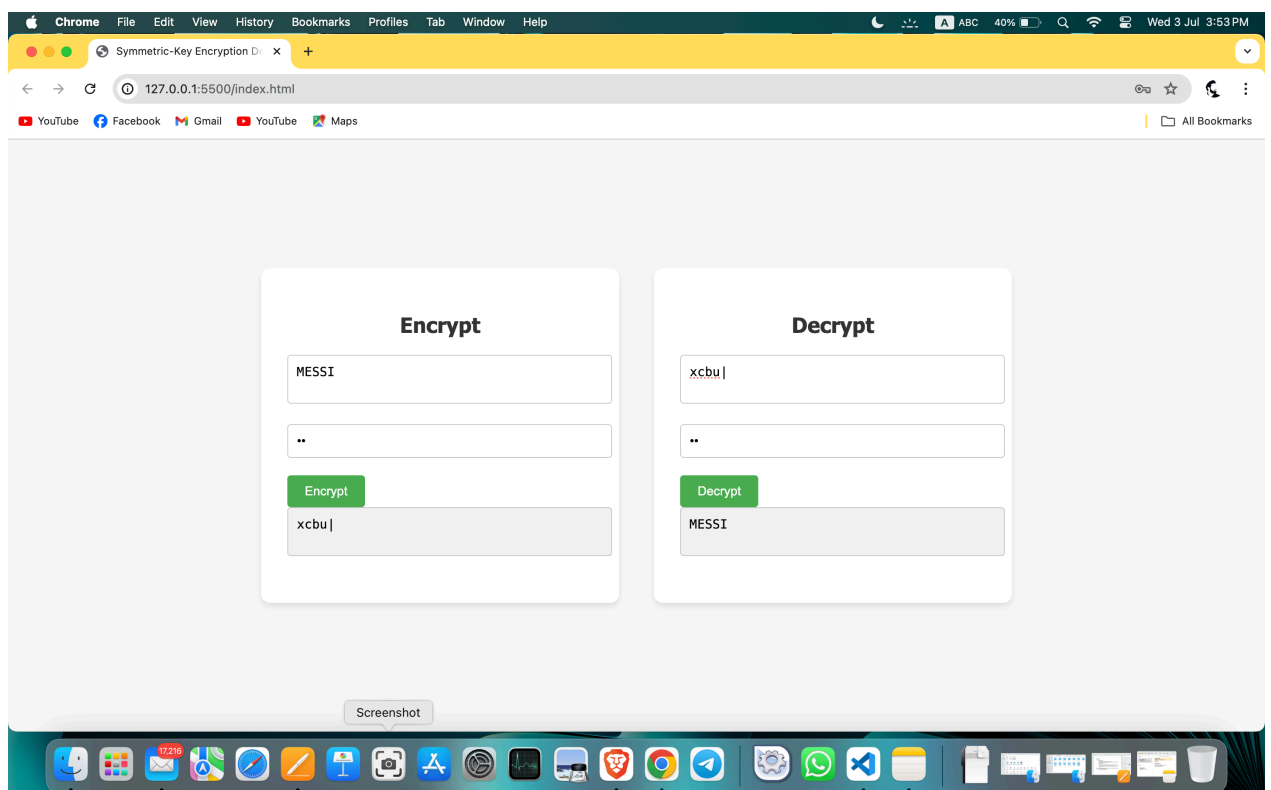
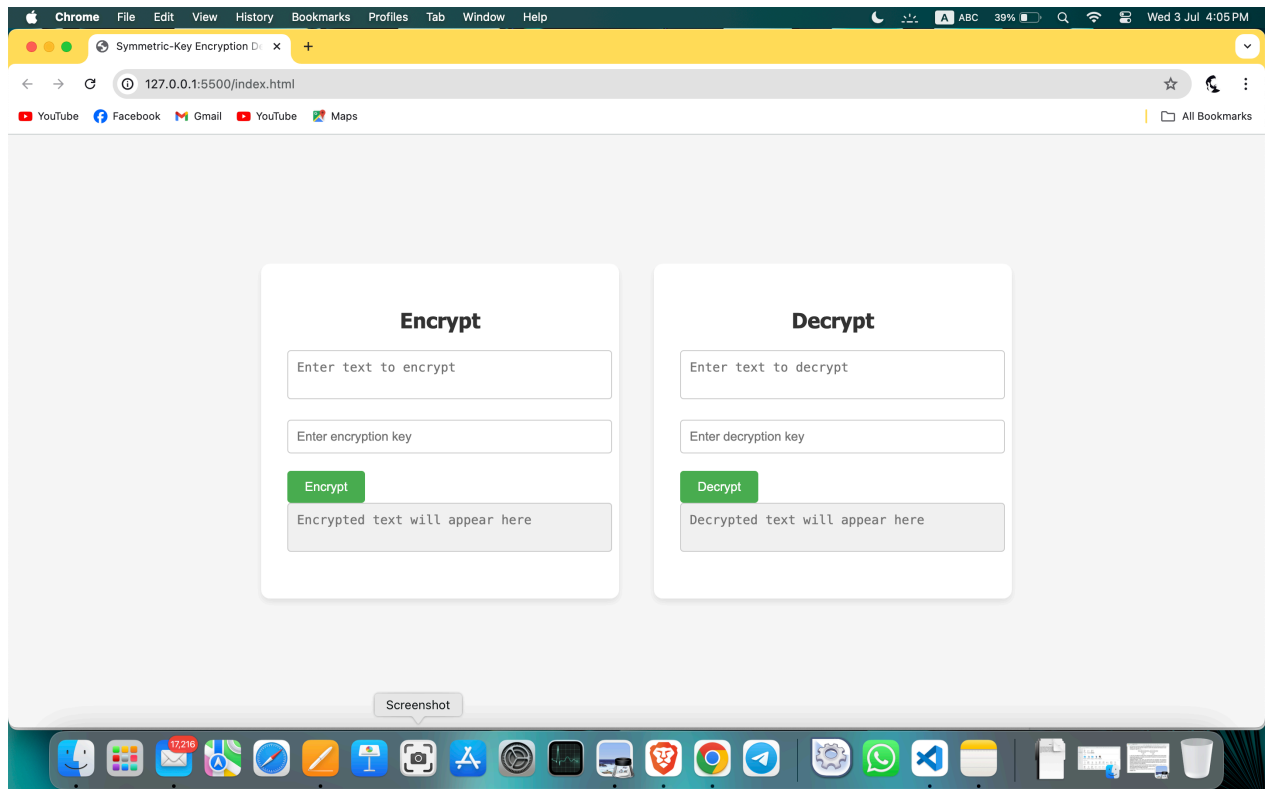
```
JS script.js > encrypt
35 }
36
37 function performXNOREncryption(text, key) {
38     // Perform XNOR operation between text and key
39     let encryptedText = '';
40     for (let i = 0; i < text.length; i++) {
41         let charCode = text.charCodeAt(i) ^ key.charCodeAt(i % key.length);
42         encryptedText += String.fromCharCode(charCode);
43     }
44     return encryptedText;
45 }
46
47 function reverseString(str) {
48     // Reverse a string
49     return str.split('').reverse().join('');
50 }
51
```

Figure 6.3.2 Source Code for *script.js*

7. Visual Representation of “ReverseX Cipher”

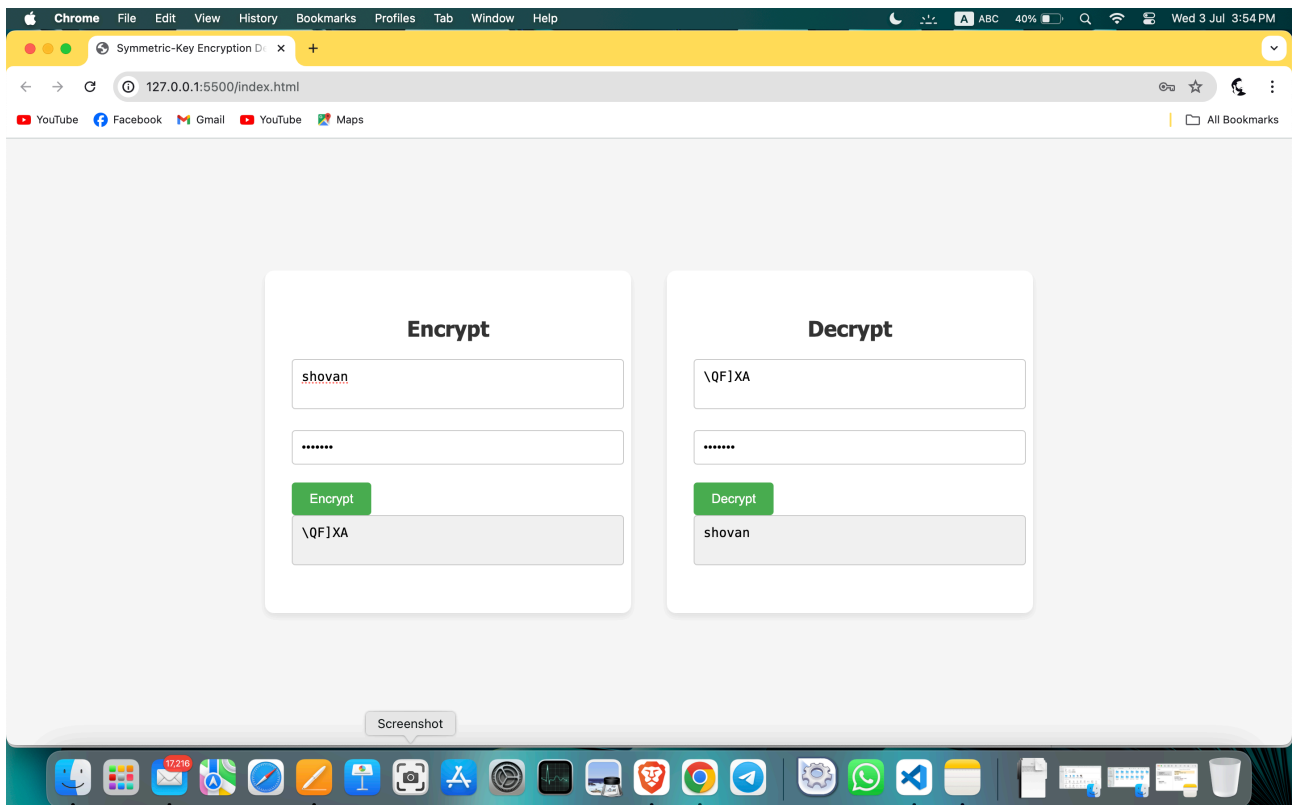
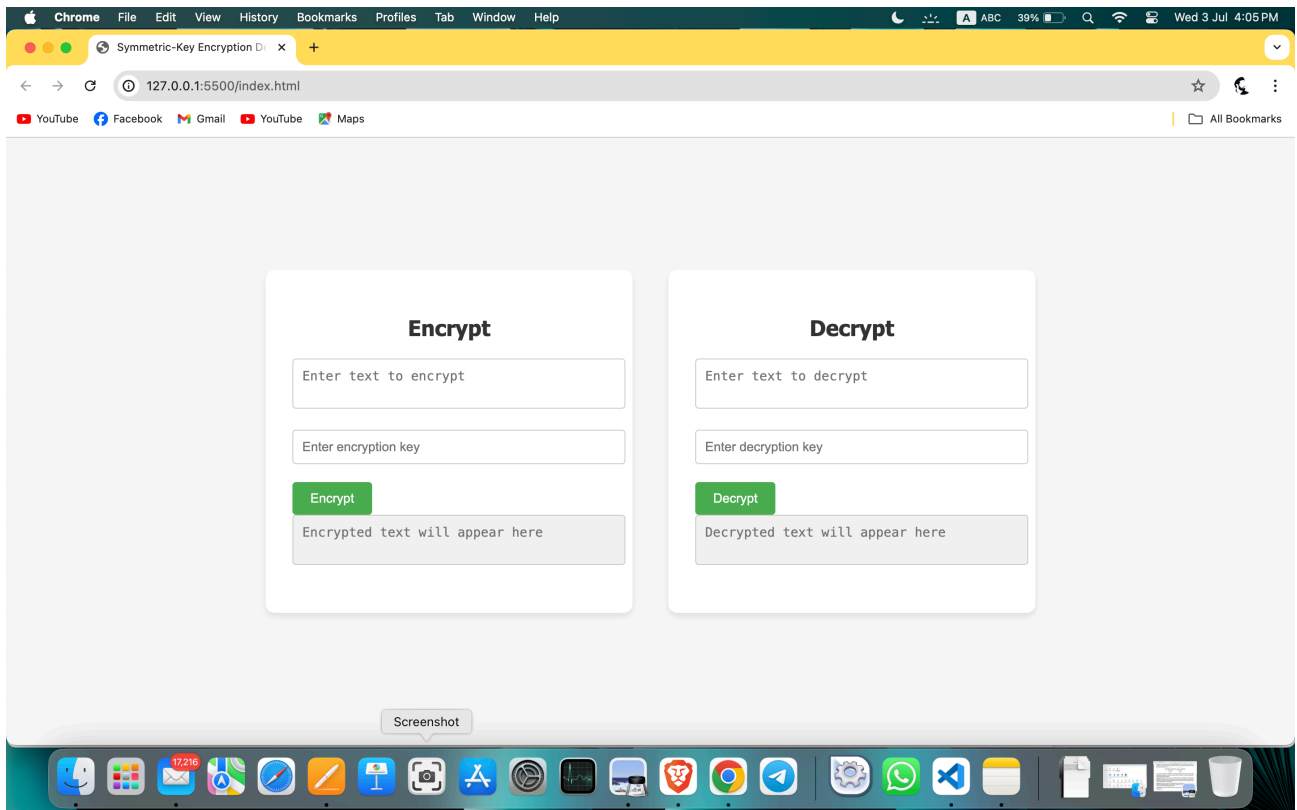
EXAMPLE 1:

Plaintext "**MESSI**" and Key "**10**"



EXAMPLE 2

Plaintext “shovan” and Key “2002026”



8. Conclusion

The ReverseX Cipher uses a mix of **reversing text** and the **XNOR** operation to encrypt and decrypt data. It's an interesting twist on traditional cryptography, showing potential for secure data handling.

References

- [1] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 7th ed. Pearson, 2017.
- [2] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd ed. Wiley, 1996.