

# **Number Theory and Cryptography Final Project**

Perform AES Encryption and Decryption on Website

Dachuan Chen

10828241

Department of Electrical Engineering

Chung Yuan Christian University

January 4, 2023

# Contents

<b>1</b>	<b>Development Environment</b>	<b>3</b>
<b>2</b>	<b>Develop Process</b>	<b>4</b>
<b>3</b>	<b>Website Introduction</b>	<b>5</b>
3.1	Functionality . . . . .	5
3.2	How to Use . . . . .	5

# List of Figures

2.1	VSCode Screenshot . . . . .	4
3.1	Website Screenshot . . . . .	6
3.2	Key Expansion Section Screenshot . . . . .	6
3.3	Encryption Section Screenshot . . . . .	6
3.4	Decryption Section Screenshot . . . . .	6
3.5	Key Expansion Result Screenshot . . . . .	6
3.6	Encryption Result Screenshot 1 . . . . .	7
3.7	Encryption Result Screenshot 2 . . . . .	7
3.8	Decryption Result Screenshot 1 . . . . .	8
3.9	Decryption Result Screenshot 2 . . . . .	8

# Chapter 1

## Development Environment

- **Operating System:** Windows 11
- **Web Server:** Firebase Hosting
- **Web Browser:** Firefox 108.0.1
- **IDE:** Visual Studio Code 1.74.2
- **Code Management:** Git 2.39.0
- **Programming Language:** Node.js v18.12.1, HTML, CSS, JavaScript
- **JS Library:** Firebase v9.2.0

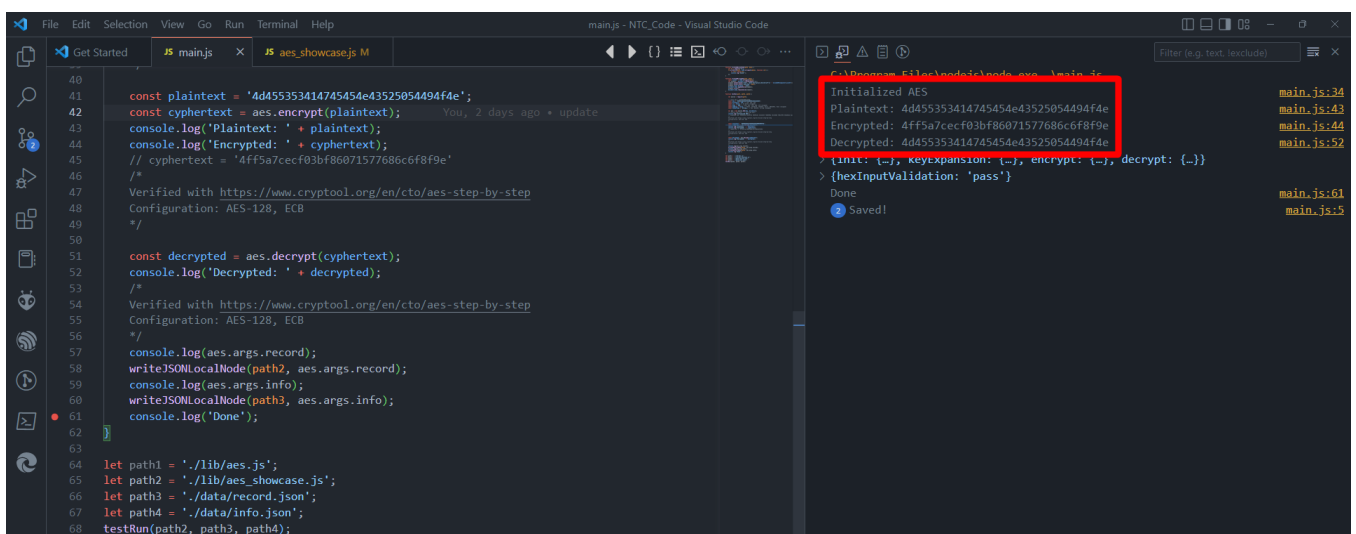
# Chapter 2

## Develop Process

- **Step 1:** Test AES Encryption and Decryption on Node.js (Figure 2.1).
- **Step 2:** Migrate the code to Firebase Hosting files.
- **Step 3:** Adjust HTML, CSS and test the website functionalities.

All the source code can be found in my GitHub Repository. The source code is written in Node.js and the website is hosted on Firebase Hosting.

Link to the source code: [https://github.com/belongtothenight/NTC\\_Code](https://github.com/belongtothenight/NTC_Code)



The screenshot shows the Visual Studio Code editor with a file named `main.js` open. The code in `main.js` is as follows:

```
40
41
42 const plaintext = '4d455353414745454e43525054494f4e';
43 const cyphertext = aes.encrypt(plaintext);
44 console.log('Plaintext: ' + plaintext);
45 console.log('Encrypted: ' + cyphertext);
46 // cyphertext = '4ff5a7cec03bf86071577686c6f8f9e'
47 /*
48 Verified with https://www.cryptool.org/en/cto/aes-step-by-step
49 Configuration: AES-128, ECB
50 */
51
52 const decrypted = aes.decrypt(cyphertext);
53 console.log('Decrypted: ' + decrypted);
54 /*
55 Verified with https://www.cryptool.org/en/cto/aes-step-by-step
56 Configuration: AES-128, ECB
57 */
58 console.log(aes.args.record);
59 writeJSONLocalNode(path2, aes.args.record);
60 console.log(aes.args.info);
61 writeJSONLocalNode(path3, aes.args.info);
62 console.log('Done');
63
64 let path1 = './lib/aes.js';
65 let path2 = './lib/aes_showcase.js';
66 let path3 = './data/record.json';
67 let path4 = './data/info.json';
68 testRun(path2, path3, path4);
```

The console output on the right shows the following:

```
Initialized AES
Plaintext: 4d455353414745454e43525054494f4e
Encrypted: 4ff5a7cec03bf86071577686c6f8f9e
Decrypted: 4d455353414745454e43525054494f4e
> {init: {...}, keyExpansion: {...}, encrypt: {...}, decrypt: {...}}
> {hexInputValidation: 'pass'}
Done
Saved!
```

Figure 2.1: VScode Screenshot

# Chapter 3

## Website Introduction

Link to the website: <https://ntc-demo-296be.web.app/>

### 3.1 Functionality

- **Key Expansion:** Generate the round keys from the original key.
- **Encryption:** Encrypt the plaintext with the round keys.
- **Decryption:** Decrypt the ciphertext with the round keys.

### 3.2 How to Use

- **Key Expansion:** Input the original key (Figure 3.2) and click the button "Generate Expanded Key" to generate the round keys (Figure 3.5).
- **Encryption:** Input the plaintext (Figure 3.3) and click the button "Encrypt" to encrypt the plaintext (Figure 3.6 - 3.7).
- **Decryption:** Input the ciphertext (Figure 3.4) and click the button "Decrypt" to decrypt the ciphertext (Figure 3.8 -3.9).



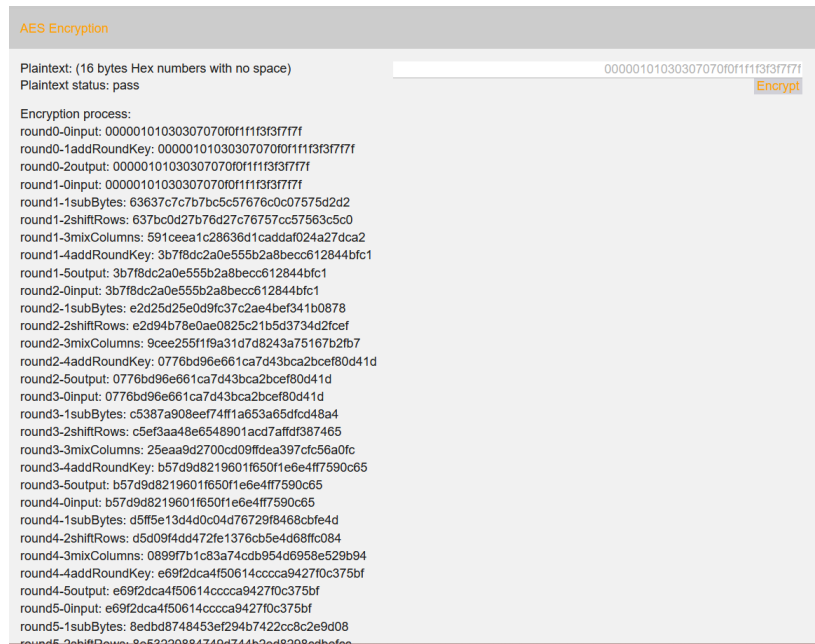


Figure 3.6: Encryption Result Screenshot 1

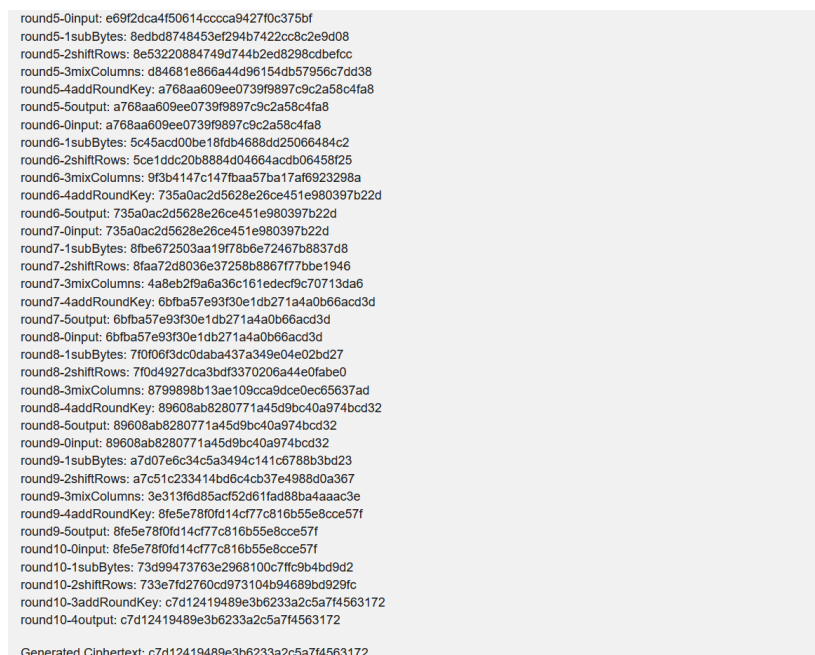


Figure 3.7: Encryption Result Screenshot 2



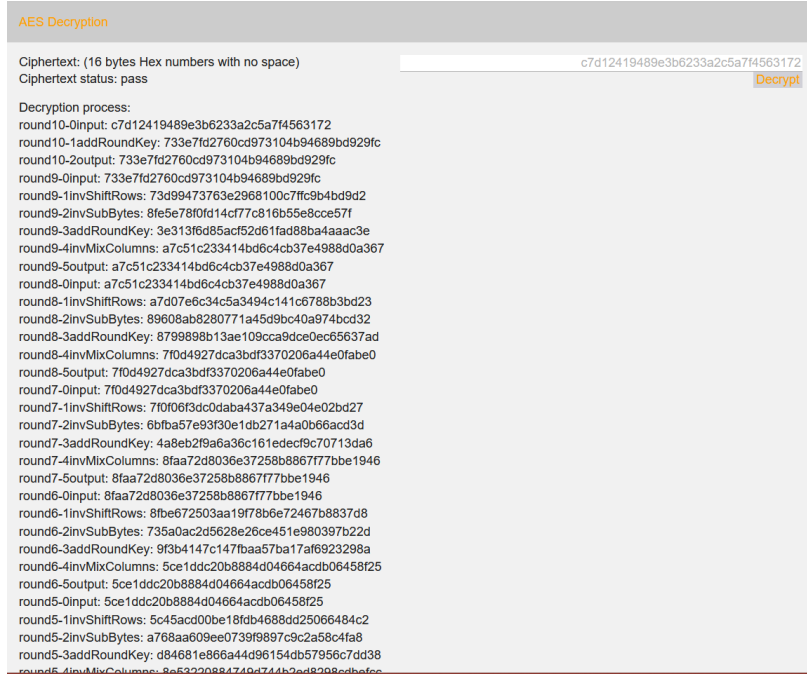


Figure 3.8: Decryption Result Screenshot 1

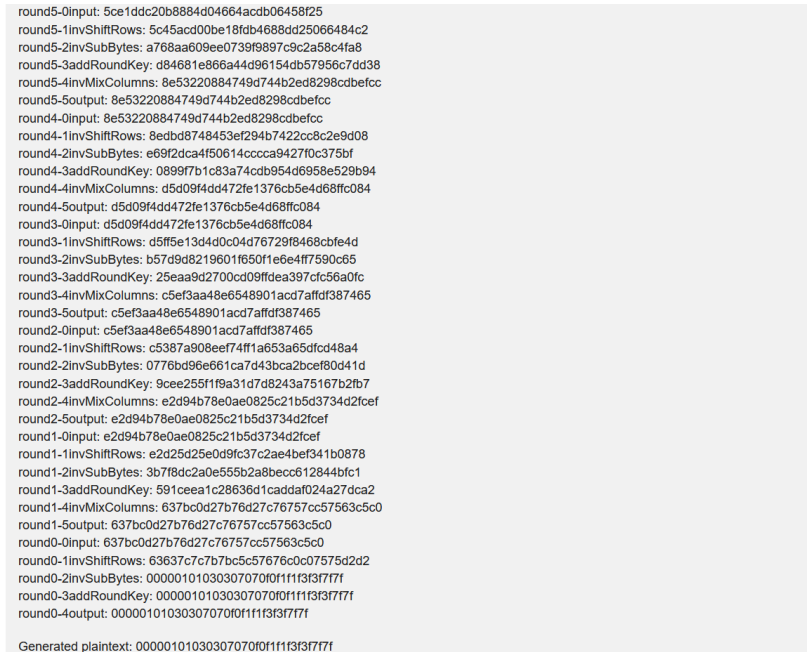


Figure 3.9: Decryption Result Screenshot 2