DS Sieve Cipher

Sieve Cipher is a type of transposition cipher where the plaintext is written into a grid and then rearranged according to a certain pattern to obtain the ciphertext. The pattern used in the rearrangement is known as the "sieve". The sieve is used to choose which rows or columns of the grid to extract the ciphertext from. The security of the Sieve Cipher lies in the secrecy of the sieve used and the size of the grid.

DS Give simple encryption/decryption example

Here's a simple example of a Sieve Cipher encryption and decryption process:

Encryption:

1. Write the plaintext "HELLO" into a grid, for example a 5x1 grid.
2. Rearrange the grid according to a predefined sieve pattern, say [2, 3, 1, 5, 4].
3. The ciphertext is obtained by extracting the characters in the order specified by the sieve pattern, resulting in the ciphertext "LHLEO".

Decryption:

1. Write the ciphertext "LHLEO" into a grid, for example a 5x1 grid.
2. Rearrange the grid according to the inverse of the sieve pattern used in the encryption, say [3, 1, 5, 2, 4].
3. The plaintext is obtained by extracting the characters in the order specified by the inverse sieve pattern, resulting in the original plaintext "HELLO".

DS js implementation

Here's a JavaScript implementation of the Sieve Cipher encryption and decryption process:

css Copy code

```
// Function to encrypt plaintext using Sieve Cipher
function encrypt(plaintext, sieve) {
    // Create a grid from the plaintext
    let grid = [];
    for (let i = 0; i < plaintext.length; i++) {
        grid[i] = plaintext[i];
    }

    // Create a copy of the grid to store the rearranged characters
    let rearranged = [...grid];

    // Rearrange the characters in the grid according to the sieve pattern
    for (let i = 0; i < grid.length; i++) {
        rearranged[sieve[i] - 1] = grid[i];
    }

    // Concatenate the rearranged characters to obtain the ciphertext
    let ciphertext = "";
    for (let i = 0; i < rearranged.length; i++) {
```