Security Project: Cipher Encryption and Decryption Tool

Kainath & Kyle

Caesar/Monoalphabetic

Caesar

Original Text: hi there

Shift: 5

Operation: Encrypt

Result Text: mn ymjwj

Monoalphabetic

```
def generate_key():
    letters = string.ascii_lowercase
    shuffled = ''.join(random.sample(letters, len(letters)))
    return {letter: shuffled[index] for index, letter in enumerate(letters)}
```

A dictionary mapping each letter of the alphabet to a randomly shuffled letter

Vigenere Cipher

Block Cipher

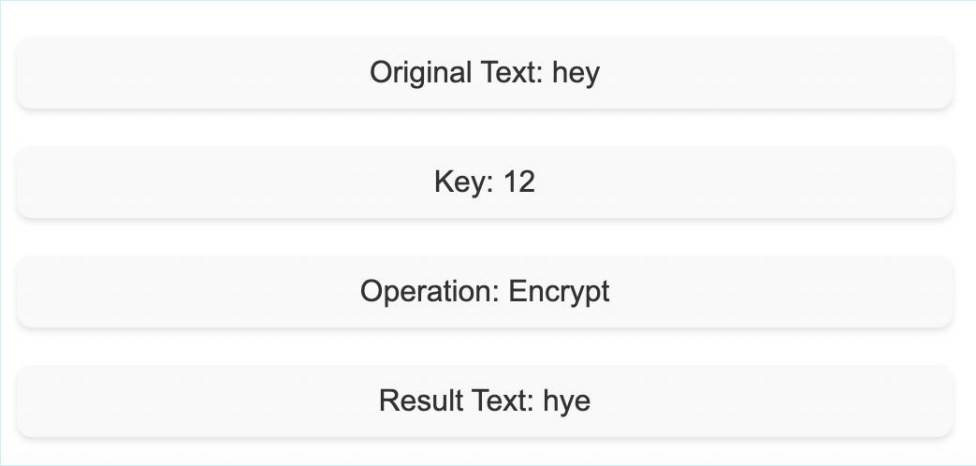
The **block_encrypt** function pads the plaintext to fit the block size (16), then encrypts it by XORing each block with the key for 12 rounds, returning the ciphertext.

```
0 XOR 0 = 0
0 XOR 1 = 1
1 XOR 0 = 1
1 XOR 1 = 0
```

```python
def block_encrypt(plaintext: bytes, key: bytes, block_size: int, rounds: int =12) -> bytes:
    # Padding
    padding_length = block_size - (len(plaintext) % block_size)
    padded_plaintext = plaintext + bytes([padding_length] * padding_length)

    ciphertext = padded_plaintext
    for _ in range(rounds):
        temp_ciphertext = b""
        for i in range(0, len(ciphertext), block_size):
            block = ciphertext[i:i + block_size]
            xor_result = bytes(a ^ b for a, b in zip(block, key))
            temp_ciphertext += xor_result
        ciphertext = temp_ciphertext
    return ciphertext
```

Transposition Cipher



encrypt_transposition

This function removes spaces from the plaintext, then arranges it into columns based on the length of the key. It fills a grid column by column, then rearranges these columns according to the sorted key order.

BOP IT!

Our Invented Cipher

The walmart version…

Twist: Add 1 to the shift

Pull: Subtract 1 from shift

Bop: Add 3 to the shift