King Saud University
College of Computer and Information Sciences
Department of Information Systems

COLUMNAR TRANSPOSITION CIPHER

INFORMATION SECURITY
COURSE PROJECT

Project By
Mohammad Alomar

31 December 2022

In this project, the student is required to choose one of the ciphers covered in the course, implement it as a command-line interface application using a programming language of his choice and preform a test run where he encodes a plaintext message and then decode it back to its original form.

The student chooses the Columnar Transposition Cipher, with Python as the programming language to implement it.

The Columnar Transposition Cipher is a simple but effective encryption technique that can be used to protect sensitive information. It works by rearranging the letters of a message in a specific order, according to a key. A brief explanation of the cipher's encoding process can be found in this video.

The cipher is very old, and its origins are not entirely clear. Some historians believe that it was first used by the ancient Greeks, while others believe that it was first used by the ancient Egyptians. The cipher was certainly in use by the Middle Ages, and it was used by many military and political leaders throughout history.

It has a historical significance as it was used during various wars and conflicts, including World War I and World War II. It was considered a relatively secure method at the time, but advancements in cryptanalysis have since exposed its vulnerabilities.

In the next pages, the student showcases his original implementation of the cipher, followed by a simple test run results of the application.

```python
def do(mode, message, keyword):

    # INVALID INPUT HANDLING
    is_mode = mode in ('ENCODE', 'DECODE')
    is_message = type(message) is str and len(message) > 0
    is_keyword = type(keyword) is str and len(keyword) > 0
    if not (is_mode and is_message and is_keyword):
        return 'ERROR INVALID INPUT'

    # VARIABLES SETUP
    n = len(message)
    m = len(keyword)
    omega = [''] * m
    priority = []
    output = ''

    # GET THE PRIORITY ORDER USING THE KEYWORD
    sorted_keyword = sorted(keyword)
    for char in keyword:
        p = sorted_keyword.index(char)
        priority.append(p)
        sorted_keyword[p] = None

    if mode == 'ENCODE':

        # WRITE OMEGA - PRIORITY ORDER - CHAR AT A TIME
        for i in range(n):
            p = priority[i % m]
            char = message[i]
            omega[p] += char

        # READ OMEGA - ASCENDING ORDER - STRING AT A TIME
        for i in range(m):
            string = omega[i]
            output += string

    if mode == 'DECODE':

        # WRITE OMEGA - ASCENDING ORDER - STRING AT A TIME
        length = n // m
        deadline = n % m
        start = 0
        for i in range(m):
            extra = priority.index(i) < deadline
            end = start + length + extra
            string = message[start:end]
            omega[i] += string
            start = end

        # READ OMEGA - PRIORITY ORDER - CHAR AT A TIME
        for i in range(n):
            p = priority[i % m]
            string = omega[p]
            char = string[i // m]
            output += char

    # THE END
    return output
```

```python
if __name__ == '__main__':

    print('Welcome to the Columnar Transposition Cipher')
    print('* Enter 1 to Encode')
    print('* Enter 2 to Decode')
    print('* Enter Anything Else to Terminate')

    while True:

        entry = input('\n' + 'Entry: ')

        if entry == '1':
            print('=> ' + do('ENCODE', input('Plaintext: '), input('Keyword: ')))
            continue

        if entry == '2':
            print('=> ' + do('DECODE', input('Ciphertext: '), input('Keyword: ')))
            continue

        print('=> thank you for using me bye <3')
        break
```

```
Welcome to the Columnar Transposition Cipher
* Enter 1 to Encode
* Enter 2 to Decode
* Enter Anything Else to Terminate

Entry: 1
Plaintext: LOOK-AT-THIS
Keyword: CAT
=> O--ILKTHOATS

Entry: 2
Ciphertext: O--ILKTHOATS
Keyword: CAT
=> LOOK-AT-THIS

Entry: 1
Plaintext: LOOK-AT-THIS
Keyword:
=> ERROR INVALID INPUT

Entry: 1
Plaintext:
Keyword: CAT
=> ERROR INVALID INPUT

Entry: 1
Plaintext: The quick brown fox jumps over the lazy dog
Keyword: Victoria
=> Tkfshdc pt ebxo gh o eoinm yqojeauwurz r vl

Entry: 2
Ciphertext: Tkfshdc pt ebxo gh o eoinm yqojeauwurz r vl
Keyword: Victoria
=> The quick brown fox jumps over the lazy dog

Entry: f
=> thank you for using me bye <3
```

github.com/moalomar/columnar