Práctica 1: Bifid Cipher

Martínez Ostoa Néstor Iván
#315618648
Criptografía - 2930
Dra. Rocío Aldeco

10 de Marzo del 2021

---

1 Describe step by step how you can decrypt a message using the Bifid cipher

1. We must assume that we have the original tableau
2. Using the tableu, we convert each letter of the ciphered text C_i into its corresponding two number representation and store it in a string of numbers E
3. We define an integer n as the length of E: $n = |E|$
4. Using the string E, we generate two more strings: A and B. A will contain the first $\frac{n}{2}$ characters of E and B will contain the second $\frac{n}{2}$ characters of E
5. A and B strings have the same length, thus, we can iterate from $0 \to \frac{n}{2} - 1$ using an index i and build a new string P as $P[i] = A[i] + B[i]$ where the $+$ operator is a character concatenator
6. Using P, we iterate over it taking two steps on each iteration and for each two steps iteration, we use the tableau to build the original message M:

   $$M \pm \text{tableau}[P[i-1], P[i]]$$

   where $i : 1 \to \frac{n}{2} - 1$
7. M is the original message

2 Use the Bifid cipher with the tableau as given to perform the follwing actions

For the encryption and decryption we use the following *tableau*:

	0	1	2	3	4
0	E	N	C	R	Y
1	P	T	A	B	D
2	F	G	H	I	K
3	L	M	O	Q	S
4	U	V	W	X	Z

Figure 1: Tableau used for this example of Bifid cipher

1. Encrypt "BRING ALL YOUR MONEY"

   PFGQRUQERQTFYFMGY

2. Decrypt "PDRRNGBENOPNIAGGF"

   TRAVEL NORTH AT ONCE

3 Bifid pseudocode

3.1 Encryption

```
/*
    Input:
        - <String> M: message to cipher
        - <Array> tableau: 2D array
    Output:
        - <String> C: ciphered message
*/
String bifid_encryption(String M, Array tableau) {
    M <- lower(M)        //transform M to lower case
    A <- ''
    B <- ''
    for letter in M {
        a, b <- -1
        for row in tableau.rows() {
            for col in tableau.cols() {
                if tableau[row][col] == letter {
                    a <- row
                    b <- col
                    return
                }
```

```
            }
        }
        A += a
        B += b
    }
    AB <- A + B // concatente A and B
    C <- ''
    for i from 1 to (length of A)-1 {
        a <- A[i-1]
        b <- A[i]
        C += tableau[a][b]
    }
    return C
}
```

3.2 Decryption

```
/*
    Input:
        - <String> M: ciphered message
        - <Array> tableau: 2D array
    Output:
        - <String>: original decrypted message M
*/
String bifid_decryption(String C, Array tableau) {
    M <- ''
    A <- ''
    B <- ''
    for letter in C {
        a <- b <- -1
        for row in tableau.rows() {
            for col in tableau.cols() {
                if tableau[row][col] == letter {
                    a <- row
                    b <- col
                    return
                }
            }
        }
        A += a
        B += b
    }
    for i from 0 to length(A) - 1 {
        a <- A[i]
        b <- B[i]
```

```
        M += tableau[a][b]
    }
    return M
}
```

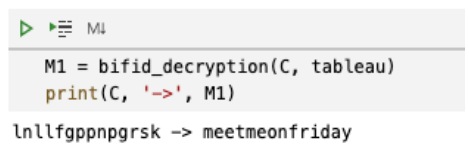
4 Bifid Python implementation

```
import numpy as np
tableau = np.array([
    ['e','n','c','r','y'],
    ['p','t','a','b','d'],
    ['f','g','h','i','k'],
    ['l','m','o','q','s'],
    ['u','v','w','x','z']
])

def get_a_b(M, tableau, decrypt=False):
    A = B = ''
    should_stop = False
    for letter in M:
        a = b = -1
        should_stop = False
        for row in range(tableau.shape[0]):
            for col in range(tableau.shape[1]):
                if tableau[row][col] == letter:
                    a = row
                    b = col
                    should_stop = True
                if should_stop:
                    break
            if should_stop:
                break
        if decrypt:
            A += str(a) + str(b)
        else:
            A += str(a)
            B += str(b)
    return A if decrypt else A + B

def bifid_encryption(M, tableau):
    M = M.lower().replace(' ', '')
    AB = get_a_b(M, tableau)
    C = ''
    for i in range(1, len(AB), 2):
        a = int(AB[i-1])
```

```
        b = int(AB[i])
        C += tableau[a][b]
    return C

def bifid_decryption(C, tableau):
    M = ''
    AB = get_a_b(C.lower().replace(' ', ''), tableau, decrypt=True)
    A = AB[:len(AB)//2]
    B = AB[len(AB)//2:]
    for i in range(len(A)):
        a = int(A[i])
        b = int(B[i])
        M += tableau[a][b]
    return M
```

4.1 Results

(a) Encryption example



(b) Decryption example

Figure 2: Bifid cipher results