**Practical Session 1**

*The Bifid cipher*

The Bifid cipher was created in about 1901 by a French cryptographer, Félix Delastalle. Although it has never been used for military or any other "serious" purpose, it has a very elegant design, is easy to implement, and quite hard to break given its simplicity.

The key for this cipher is any permutation of the alphabet (except for the letter J). One way to remember a key is to choose a word with no repeating letters such as "ENCRYPT" to start the permutation, and finish with the remaining letters. This permutation is placed in a 5 x 5 array called the tableau.

This produces:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 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 |

For this tableau, A has indices 1 and 2; while X has indices 4 and 3.

To use the Bifid cipher, encode the message using the indices from the tableau. So that, for example, the message “MEET ME ON FRIDAY” would be encoded as

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| M | E | E | T |  | M | E |  | O | N |  | F | R | I | D | A | Y |
| 3 | 0 | 0 | 1 |  | 3 | 0 |  | 3 | 0 |  | 2 | 0 | 2 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |  | 1 | 0 |  | 2 | 1 |  | 0 | 3 | 3 | 4 | 2 | 4 |

The indices are then read off row by row:

3 0 0 1 3 0 3 0 2 0 2 1 1 0 1 0 0 1 1 0 2 1 0 3 3 4 2 4

These indices then are grouped back into pairs and turned into letters by using the original tableau:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 30 | 01 | 30 | 30 | 20 | 21 | 10 | 10 | 01 | 10 | 21 | 03 | 34 | 24 |
| L | N | L | L | F | G | P | P | N | P | G | R | S | K |

The ciphertext is thus “LNLLFGPPNPGRSK”.

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

2. Use the Bifid cipher with the tableau as given to

a. encrypt BRING ALL YOUR MONEY

b. decrypt PDRRNGBENOPNIAGGF

3. In teams, create the pseudo-code to implement the Bifid cipher. This including message encryption and decryption.

4. When all the team members agree on the previous point, implement your pseudo-code in the programming language all members agree.

*NOTE*: The submission of this file with the corresponding answers is individual. For questions 3 and 4 you should submit your code (source file) with the team member’s names.

1: To decrypt a message first we must have the key and construct the tableau. Then we create an empty array an iterate through each character of the message that we want to decrypt. For each character we are going to append to the array the i coordinate and then the j coordinate.Then we start building the plain message using the resulting array iterating from the start to the middle of the array. We are going to be getting the coordinates of the current value of [index, index + (length / 2)] and the character that is in the tableau is added to the result. We repeat that step until we reach the middle of the array as mentioned before.

2:

Encrypt BRING ALL YOUR MONEY

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| B | R | I | N | G |  | A | L | L |  | Y | O | U | R |  | M | O | N | E | Y |
| 1 | 0 | 2 | 0 | 2 |  | 1 | 3 | 3 |  | 0 | 3 | 4 | 0 |  | 3 | 3 | 0 | 0 | 0 |
| 3 | 3 | 3 | 1 | 1 |  | 2 | 0 | 0 |  | 4 | 2 | 0 | 3 |  | 1 | 2 | 1 | 0 | 4 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1  0 | 2  0 | 2  1 | 3  3 | 0  3 | 4  0 | 3  3 | 0  0 | 0  3 | 3  3 | 1  1 | 2  0 | 0  4 | 2  0 | 3  1 | 2  1 | 0  4 |
| P | F | G | Q | R | U | Q | E | R | Q | T | F | Y | F | M | G | Y |

The ciphertext is thus “PFGQRUQERQTFYFMGY”

decrypt PDRRNGBENOPNIAGGF

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1  0 | 1  4 | 0  3 | 0  3 | 0  1 | 2  1 | 1  3 | 0  0 | 0  1 | 3  2 | 1  0 | 0  1 | 2  3 | 1  2 | 2  1 | 2  1 | 2  0 |
| P | D | R | R | N | G | B | E | N | O | P | N | I | A | G | G | F |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1  1 | 0  3 | 1  2 | 4  1 | 0  0 | 3  0 | 0  1 | 3  2 | 0  3 | 1  1 | 2  2 | 1  2 | 1  1 | 3  2 | 0  1 | 0  2 | 0  0 |
| T | R | A | V | E | L | N | O | R | T | H | A | T | O | N | C | E |

The decrypted text is thus “TRAVELNORTHATONCE”

3:

Before encrypting or decrypting we first build a 5x5 matrix using the key

**To build matrix**

create empty matrix of 5x5

for each letter in the key

add it to the matrix

for each letter in the alphabet

If it is not in the matrix

Add it to the matrix

**To encrypt**

Create empty list

Create a string variable that is empty

for each word in the message we want to cipher

for each character in the word

add the first coordinate of the character to the list

for each word in the message we want to cipher

for each character in the word

add the second coordinate of the character to the list

i = 0

while i < length of the list

get the coordinates of list[i] and list[i+1]

get the character with the corresponding coordinates in the matrix

add the character to the string variable

i = i + 2

return the string variable

**To decrypt**

create empty list

create a string variable that is empty

for each character in the ciphered message

add the first coordinate of the character to the list

add the second coordinate of the character to the list

i = 0

while i < (length of the list) / 2

get the coordinates of list[i] and list[i + (length of the list) / 2]

get the character with the corresponding coordinates in the matrix

add the character to the string variable

i = i + 1

return the string variable