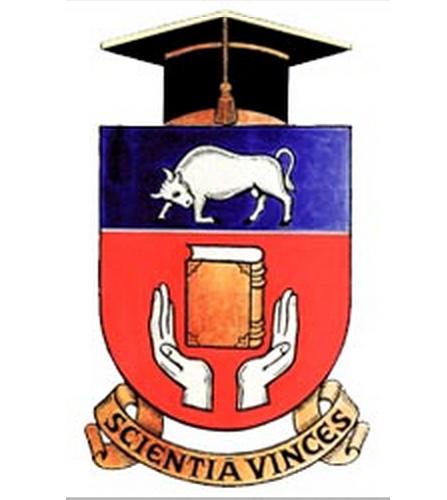
Universitatea de Stat din Tiraspol

Facultatea de Fizica Matematica si Informatica



Cifrul Omofonic

**Elaborat de**: Dordea Pavel, student gr.3i

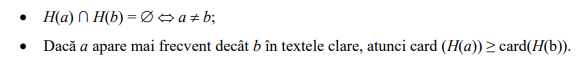
**Verificat**: Globa Angela

**Chișinău, 2021**

***Cifrul Omofonic***

***#Despre***

Cifrul omofonic (homophonic ciphers) este un cifru de substituţie în care un caracter al alfabetului mesajului clar (alfabet primar) poate să aibă mai multe reprezentări. Ideea utilizată în aceste cifruri este uniformizarea frecvenţelor de apariţie a caracterelor alfabetului textului cifrat (alfabet secundar), pentru a îngreuna atacurile criptanalitice. Astfel, litera A - cu cea mai mare frecvenţă de apariţie în alfabetul primar – poate fi înlocuită de exemplu cu H, # sau m. Cifrul omofonic este un cifru intermediar între sistemele mono şi cele polialfabetice. Principalul lui scop este de a evita atacul prin frecvenţa de apariţie a caracterelor. Se presupune că a fost utilizat prima oară în 1401 de către ducele de Mantua. În cifrul omofonic fiecărui caracter a∈m i se asociază o mulţime H(a) ⊂ c astfel încât:

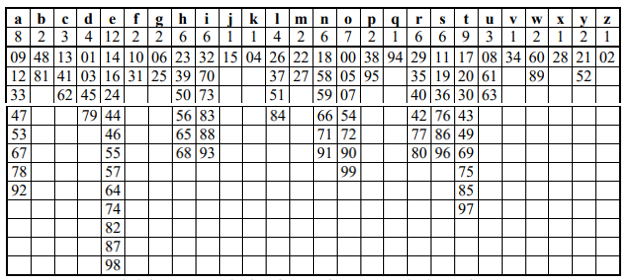


Criptarea unui caracter x ∈m se face cu un element ales aleator din H(a). Pentru decriptarea lui y ∈ c se caută o mulţime H(x) astfel încât y ∈ H(x).

Exemplu. Pentru limba engleză poate fi utilizat cifrul definit de tabelul 5.5, în primele două linii ale căruia sunt aşezate literele alfabetului latin şi frecvenţele rotunjite ale acestora. În coloanele de sub litera x este situat H(x). De exemplu



Pentru criptarea textului „ab” se poate folosi oricare din secvenţele 0948, 1248, 3348, 4748, 5348, 6748, 7848, 9248, 0981, 1281, 3381, 4781, 5381, 6781, 7881, 9281. Tabelul 5.5 a fost completat cu numerele 00, 01,…, 99, ele fiind în cantitate de 100 pentru a simplifica calculele legate de frecvența literelor (adică pentru a corespunde cu 100% a frecvențelor). Însă elementele acestui tabel pot fi oricare simboluri, iar cantitatea lor va determina coeficientul de proporționalitate pentru frecvențele fiecărei litere din alfabet.



Deşi mai greu de spart decât cifrurile de substituţie simple (monoalfabetice), cifrul omofonic nu maschează total proprietăţile statistice ale textului clar. În cazul unui atac cu text clar cunoscut, cifrul se sparge extrem de uşor. Atacul cu text cifrat este mai dificil, dar unui calculator îi va lua doar câteva secunde pentru a-l sparge.

***#COD***

Dictionarul

aN = [

27, 44, 43, 48, 38, 22, 59, #a

34, 92, #b

20, 37, 94, #c

31, 99, 50, 60, #d

91, 68, 21, 45, 90, 58, 81, 64, 65, 36, 96, #e

12, 74, #f

41, 76, #g

40, 55, 78, 83, #h

62, 61, 79, 80, 15, #i

24, #j

88, #k

95, 49, 28, 70, #l

89, 25, #m

53, 33, 66, 98, 13, #n

46, 86, 11, 42, 63, 69, #o

54, 82, #p

85, #q

77, 87, 97, 29, 26, #r

35, 17, 84, 47, 52, #s

71, 14, 51, 30, 32, 73, 16, 19, #t

93, 72, 39, #u

10, #v

67, 75, #w

23, #x

57, 56, #y

18 #z

]

Criptare

def encrypt():

print('Introduceti mesajul dvs.')

msg = input(">")

msg = msg.upper()

encrMsg = ""

maxLen = 0

i = 0

while(len(msg) > i):

dMsg = ord(msg[i]) - 65 #este cheia UNICODE pentru numerele cu text simplu

maxLen = len(x[dMsg]) #Lungimea maximă a generatorului aleator

encrMsg += str(x[dMsg][random.randint(0, maxLen - 1)])

i += 1

print("\nMesajul dvs.: " + encrMsg + "\n\n")

askScreen()

Decriptare

def decrypt():

print('Introduceti mesajul dvs. pentru decriptare')

msg = input(">")

msg = wrap(msg, 2)

msgcount = 0

w = 0

i = 0

decrMsg = "Mesajul dvs.:"

print("")

while(len(msg) -1 >= msgcount):

while(24 > i):

while(len(x[i]) > w):

if(str(msg[msgcount]) == str(x[i][w])):

decrMsg = str(decrMsg) + (chr((i)+65))

w += 1

i += 1

w = 0

w = 0

i = 0

msgcount += 1

print(str(decrMsg) + "\n\n")

askScreen()

Corpul

def askScreen():

print("|Pavel Dordea|\n |Homophonic System|")

print("\ne pentru criptare, d pentru decriptare")

print(" fara spatii!")

choose = input(">")

if(choose == "e"):

encrypt()

elif(choose == "d"):

decrypt()

else:

print("Incercati din nou \n\n")

askScreen()

askScreen()

Tot programul

# -\*- coding: utf-8 -\*-

"""

Created on Tue Dec 28 15:11:14 2021

@author: Pavel

"""

import os

import random

from textwrap import wrap

aN = [

27, 44, 43, 48, 38, 22, 59, #a

34, 92, #b

20, 37, 94, #c

31, 99, 50, 60, #d

91, 68, 21, 45, 90, 58, 81, 64, 65, 36, 96, #e

12, 74, #f

41, 76, #g

40, 55, 78, 83, #h

62, 61, 79, 80, 15, #i

24, #j

88, #k

95, 49, 28, 70, #l

89, 25, #m

53, 33, 66, 98, 13, #n

46, 86, 11, 42, 63, 69, #o

54, 82, #p

85, #q

77, 87, 97, 29, 26, #r

35, 17, 84, 47, 52, #s

71, 14, 51, 30, 32, 73, 16, 19, #t

93, 72, 39, #u

10, #v

67, 75, #w

23, #x

57, 56, #y

18 #z

]

x = [

[aN[0], aN[1], aN[2], aN[3], aN[4], aN[5], aN[6]], #a 0

[aN[7], aN[8]], #b 1

[aN[9], aN[10], aN[11]], #c 2

[aN[12], aN[13], aN[14], aN[15]], #d 3

[aN[16], aN[17], aN[18], aN[19], aN[20], aN[21], aN[22], aN[23], aN[24], aN[25], aN[26]], #e 4

[aN[27], aN[28]], #f 5

[aN[29], aN[30]], #g 6

[aN[31], aN[32], aN[33], aN[34]], #h 7

[aN[35], aN[36], aN[37], aN[38], aN[39]], #i 8

[aN[40]], #j 9

[aN[41]], #k 10

[aN[42], aN[43], aN[44], aN[45]], #l 11

[aN[46], aN[47]], #m 12

[aN[48], aN[49], aN[50], aN[51], aN[52]], #n 13

[aN[53], aN[54], aN[55], aN[56], aN[57], aN[58]], #o 14

[aN[59], aN[60]], #p 15

[aN[61]], #q 16

[aN[62], aN[63], aN[64], aN[65], aN[66]], #r 17

[aN[67], aN[68], aN[69], aN[70], aN[71]], #s 18

[aN[72], aN[73], aN[74], aN[75], aN[76], aN[77], aN[78], aN[79]], #t 19

[aN[80], aN[81], aN[82]], #u 20

[aN[83]], #v 21

[aN[84], aN[85]], #w 22

[aN[86]], #x 23

[aN[87], aN[88]], #y 24

[aN[89]] #z 25

]

def encrypt():

print('Introduceti mesajul dvs.')

msg = input(">")

msg = msg.upper()

encrMsg = ""

maxLen = 0

i = 0

while(len(msg) > i):

dMsg = ord(msg[i]) - 65 #este cheia UNICODE pentru numerele cu text simplu

maxLen = len(x[dMsg]) #Lungimea maximă a generatorului aleator

encrMsg += str(x[dMsg][random.randint(0, maxLen - 1)])

i += 1

print("\nMesajul dvs.: " + encrMsg + "\n\n")

askScreen()

def decrypt():

print('Introduceti mesajul dvs. pentru decriptare')

msg = input(">")

msg = wrap(msg, 2)

msgcount = 0

w = 0

i = 0

decrMsg = "Mesajul dvs.:"

print("")

while(len(msg) -1 >= msgcount):

while(24 > i):

while(len(x[i]) > w):

if(str(msg[msgcount]) == str(x[i][w])):

decrMsg = str(decrMsg) + (chr((i)+65))

w += 1

i += 1

w = 0

w = 0

i = 0

msgcount += 1

print(str(decrMsg) + "\n\n")

askScreen()

def askScreen():

print("|Pavel Dordea|\n |Homophonic System|")

print("\ne pentru criptare, d pentru decriptare")

print(" fara spatii!")

choose = input(">")

if(choose == "e"):

encrypt()

elif(choose == "d"):

decrypt()

else:

print("Incercati din nou \n\n")

askScreen()

askScreen()

***#Screenshots***

