Cryptography

Summer 2023

by Prof. Dr. Alexander Wiesmaier

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

Praktikum 01

Date:	15.04.2023
Supervisor:	Prof. Wiesmaier
Students:	Réault Corentin Matr.Nr.1123639

Task 1 (Implementing a simple permutation cipher)

Python program

```
#!/usr/bin/python

import sys
from math import ceil

e = [2,4,1,5,3]

def encrypt(plaintext):
    length = len(plaintext)
    nb_iteration = ceil(length/5)
    total_nb_char = nb_iteration*5
    plaintext = plaintext.ljust(total_nb_char,'x')
    print("The plaintext is: "+str(plaintext))
    cyphertext = ""
    for i in range (0, nb_iteration):
        lower_bound = i*5
        upper_bound = (i+1)*5
        p = plaintext[lower_bound:upper_bound]
        print(p)
        c = ['x']*5
        for j in range (1, 6):
            c[e.index(j)] = p[j-1]
        print(c)
        cyphertext += "".join([str(k) for k in c])
    print("The encrypted plaintext (cyphertext) is: "+str(cyphertext)
```

```
def calculate_decryption_key():
    d = ['x']*5

    for i in range (1, 6):
        d[i-1] = e.index(i)+1
    print("The decryption key is: "+str(d))
    return d

def decrypt(cyphertext):
    d = calculate_decryption_key()

    length = len(cyphertext)
    nb_iteration = ceil(length/5)
    plaintext = ""
    print("The cyphertext is: "+str(cyphertext))
    for i in range (0, nb_iteration):
        lower_bound = i*5
        upper_bound = (i+1)*5
        c = cyphertext[lower_bound:upper_bound]
        print(c)
        p = ['x']*5
        for j in range (1, 6):
            p[d.index(j)] = c[j-1]
        print(p)
        plaintext += "".join([str(k) for k in p])
    print("The decrypted cyphertext (plaintext) is: "+str(plaintext))

if __name__ == '__main__':
    function = sys.argv[1]
    match function:
        case "encrypt":
            encrypt(sys.argv[2])
        case "decrypt":
            decrypt(sys.argv[2])
        case "ekey":
            print("The choosen encryption key is: "+str(e))
        case "dkey":
            calculate_decryption_key()
        case _:
            print("Error, please use the right argmuent")
```

Encryption and decryption examples

We encrypt the following strings: helloworld, publicvoidmain, thisisnotamiracle. The strings are cut in several pieces of 5 characters (key size). If the last block has less than 5 characters, we fill with 'x' until we reach a number of 5. The encrypted string is the concatenation of all the encrypted blocks.

```
[cocopops@cocopops-laptop Practical1]$ python3.10 task1.py ekey
The choosen encryption key is: [2, 4, 1, 5, 3]
[cocopops@cocopops-laptop Practical1]$ python3.10 task1.py encrypt helloworld
The plaintext is: helloworld
hello
['e', 'l', 'h', 'o', 'l']
world
['o', 'l', 'w', 'd', 'r']
The encrypted plaintext (cyphertext) is: elhololwdr
[cocopops@cocopops-laptop Practical1]$ python3.10 task1.py encrypt publicvoidmain
The plaintext is: publicvoidmainx
publi
['u', 'l', 'p', 'i', 'b']
cvoid
['v', 'i', 'c', 'd', 'o']
mainx
['a', 'n', 'm', 'x', 'i']
The encrypted plaintext (cyphertext) is: ulpibvicdoanmxi
[cocopops@cocopops-laptop Practical1]$ python3.10 task1.py encrypt thisisnotamiracle
The plaintext is: thisisnotamiraclexxx
thisi
['h', 's', 't', 'i', 'i']
snota
['n', 't', 's', 'a', 'o']
mirac
['i', 'a', 'm', 'c', 'r']
lexxx
['e', 'x', 'l', 'x', 'x']
The encrypted plaintext (cyphertext) is: hstiintsaoiamcrexlxx
[cocopops@cocopops-laptop Practical1]$
```

Abbildung 1: Encryption example with the python program

Then, the previously encrypted strings are taken back and decrypted using the decryption key calculated in the program from the encryption key. However, if the starting character string has received 'x' to complete the last block during encryption, these will remain after performing the reverse step (decryption).

```
[cocopops@cocopops-laptop Practical1]$ python3.10 task1.py dkey
The decryption key is: [3, 1, 5, 2, 4]
[cocopops@cocopops-laptop Practical1]$ python3.10 task1.py decrypt elhololwdr
The decryption key is: [3, 1, 5, 2, 4]
The cyphertext is: elhololwdr
elhol
['h', 'e', 'l', 'l', 'o']
olwdr
['w', 'o', 'r', 'l', 'd']
The decrypted cyphertext (plaintext) is: helloworld
[cocopops@cocopops-laptop Practical1]$ python3.10 task1.py decrypt ulpibvicdoanmxi
The decryption key is: [3, 1, 5, 2, 4]
The cyphertext is: ulpibvicdoanmxi
ulpib
['p', 'u', 'b', 'l', 'i']
vicdo
['c', 'v', 'o', 'i', 'd']
anmxi
['m', 'a', 'i', 'n', 'x']
The decrypted cyphertext (plaintext) is: publicvoidmainx
[cocopops@cocopops-laptop Practical1]$ python3.10 task1.py decrypt hstiintsaoiamcrexlxx
The decryption key is: [3, 1, 5, 2, 4]
The cyphertext is: hstiintsaoiamcrexlxx
hstii
['t', 'h', 'i', 's', 'i']
ntsao
['s', 'n', 'o', 't', 'a']
iamcr
['m', 'i', 'r', 'a', 'c']
exlxx
['l', 'e', 'x', 'x', 'x']
The decrypted cyphertext (plaintext) is: thisisnotamiraclexxx
[cocopops@cocopops-laptop Practical1]$
```

Abbildung 2: Decryption example with the python program

Task 2 (Cryptanalysis and CrypTool)

Describe an algorithm to break a Vigenère cipher

The most useful solution I could find is the Kasiski test, named after the Prussian major who first published it.

The first step in this solution is to find the length of the secret key. This can be done by a first statistical analysis on n-grams (sub-sequence of n elements), preferably with n>=2. Thus we can observe repetitions of n-grams, which gives us clues on the fact that the letters encrypted in these n-grams could have been encrypted with the same part of the secret key, which allows us to determine the possible key lengths according to the number of characters between each of these patterns.

Then the second step consists in finding the content of the secret key of n characters. To do this, we can separate the encrypted text into n parts according to the characters that have been encrypted with the same letter as the secret key, and then perform a statistical analysis on each encrypted subtext to determine the shift and thus the letter used for encryption.

Vigenère cyphertext decryption

Cyphertext to decrypt:

Ww yrv xarorp xarg Vrmwrolxmwv fzi zwvcui bzvg Vmxj ewjfkmivvxx. Ww yrvxf kgeen kmgsig Wmgi mg altvq Gwwv lrw vmg Vrmwrolxmwv hiinli uzga kgjfr lwlt ryy altvr Gsgjnyvzw. Gzrxk Xcxil oet vw xfhnzga ksyvmm mrf ryl vip Vmxjr gexlulnliixxge wxulu gymrqwexxji Gexewmp.

found keyword : SECRET

final decrypted cyphertext:

ES WAR EINMAL EINE ENTENMUTTER DIE GERADE IHRE EIER AUSBRUETETE. ES WAREN GENAU SIEBEN EIER IN IHREM NEST UND DIE ENTENMUTTER FREUTE SICH SCHON SEHR AUF IHREN NACHWUCHS. EINES TAGES WAR ES ENDLICH SOWEIT UND AUS DEN EIERN ENTSCHLUEPFTEN SECHS PUTZMUNTERE ENTLEIN.

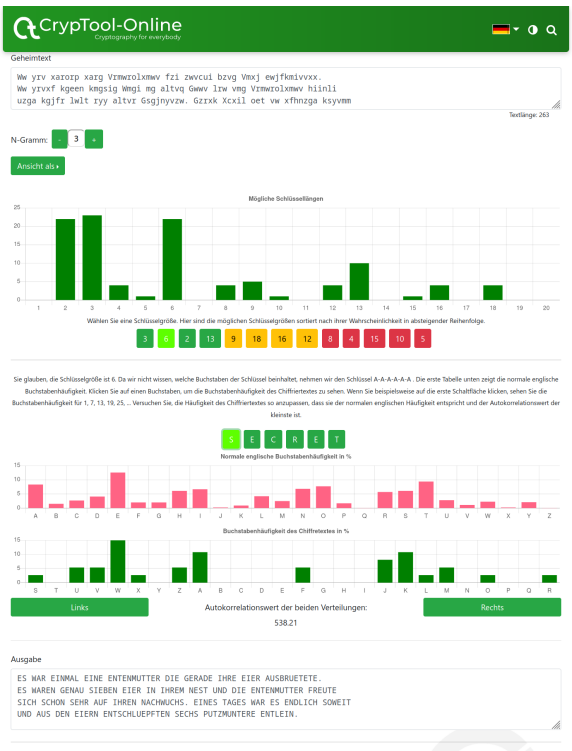

Abbildung 3: CrypTool screenshot for vigenère decryption

Substitution cyphertext decryption

For this part on monoalphabetic substitution, a simple statistical analysis was needed, with some brute force and German word recognition to refine the solution.

Cyphertext to decrypt:

Oet vcntj chht vsjptnoydktj sjp ieq tejti bthrtj, zcnqtj Atptnahcsi utnotdtj. Jsn pco oetrqt Te hcb jkyd eiitn sjutnotdnq ej ednti Jtoq. To vcn bnktootn cho pet cjptntj Tetn sjp ok otdn pet Tjqtjisqqtn csyd pcnstrtn jcydpcydqt, gkjjqt oet oeyd jeydq tnejjtnj vcjj oet to tebtjqheyd bthtbq dcqqt?

final decrypted cyphertext:

Sie waren alle wunderschoen und mit einem gelben, zarten Aederalaum versehen.Nur das siebte Ei lag noch immer unversehrt in ihrem Nest. Es war groesserals die anderen Eier und so sehr die Entenmutter auch darueber nachdachte,konnte sie sich nicht erinnern wann sie es eigentlich gelegt hatte?

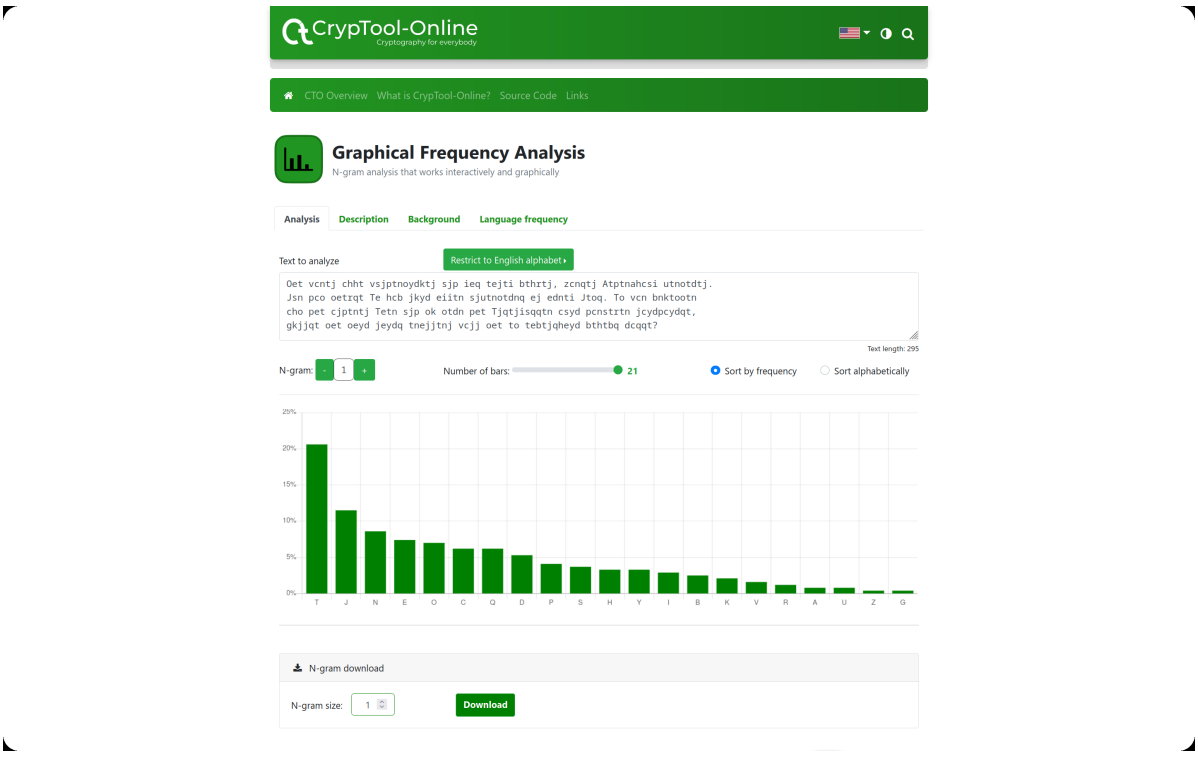

Abbildung 4: CrypTool Graphical Frequency Analysis screenshot

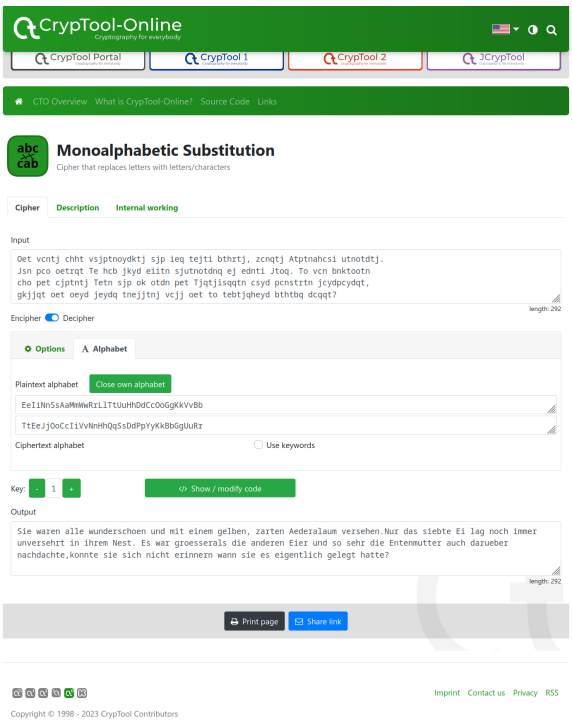

Abbildung 5: CrypTool Monoalphabetic Substitution screenshot

Literatur

[1] https://www.cryptool.org, last accessed: 15.04.2023

[2] https://en.wikibooks.org/wiki/Cryptography/Breaking_VigenÃĺre_cipher, last accessed: 16.04.2023

[3] https://www.johndcook.com/blog/2021/08/19/vigenere/, last accessed: 16.04.2023