

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Фізико-технічний інститут

Лабораторна робота №2
З предмету «Криптографія»

Виконали:
Студенти 3 курсу,
ФТІ, групи ФБ-72
Курт Олег, Вовчук Роман

Київ 2019

Варіант 7

0. Уважно прочитати методичні вказівки до виконання комп'ютерного практикуму.

1. Самостійно підібрати текст для шифрування (2-3 кб) та ключі довжини $r = 2, 3, 4, 5$, а також довжини 10-20 знаків. Зашифрувати обраний для довжини ключа відкритий текст шифром Віженера з цими ключами.
2. Підрахувати індекси відповідності для відкритого тексту та всіх одержаних шифртекстів і порівняти їх значення.
3. Використовуючи наведені теоретичні відомості, розшифрувати наданий шифртекст (згідно свого номеру варіанта).

Текст для шифрування:

[illegible]

Шифрування тексту використовуючи різну довжину ключа:

$$R=2$$
[illegible]
$$R=3$$
[illegible]
$$R = 4$$
[illegible]
$$R=5$$
[illegible]

R = 10

[illegible]

R = 11

[illegible]


```

    'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э', 'ю', 'я']

var = "рома"

key2 = "оп"

key3 = "лук"

key4 = "луна"

key5 = "метка"

key10 = "романромео"

key11 = "всепереплет"

key12 = "сноморейните"

key13 = "йнопотянизани"

key14 = "тъзанейпотянет"

key15 = "сяклубокэтотмир"

key16 = "веретеносовпаден"

key17 = "аниктонеговорилчт"

key18 = "обудетпростоноесли"

key19 = "истоятьтововесьрост"

key20 = "еслигреметыгромчево"

def index_sovpadeni(text1, arr1, keey):

    j = 0

    count = collections.Counter()

    for letter_text in text1:

        for letter_dict in arr1:

            if letter_dict == letter_text:

                count[letter_dict] += 1

                break

    t = list(count)

    for i in range(len(count)):

        j += count[t[i]] * (count[t[i]] - 1)

    results.write("Ключ " + str(len(keey)) + ": " + str((1/(len(text1) * (len(text1) - 1)) * j)) + '\n')

def encryption(open_text, keey, cipher):

    textc = ""

    for i in range(len(open_text)):

        cipher.write(arr[(arr.index(open_text[i]) + arr.index(keey[i % len(keey)])) % 32])

        textc += arr[(arr.index(open_text[i]) + arr.index(keey[i % len(keey)])) % 32]

    index_sovpadeni(textc, arr, keey)

    cipher.close()

encryption(text_work_open, key2, file2)

encryption(text_work_open, key3, file3)

encryption(text_work_open, key4, file4)

encryption(text_work_open, key5, file5)

encryption(text_work_open, key10, file10)

encryption(text_work_open, key11, file11)

encryption(text_work_open, key12, file12)

encryption(text_work_open, key13, file13)

encryption(text_work_open, key14, file14)

encryption(text_work_open, key15, file15)

encryption(text_work_open, key16, file16)

encryption(text_work_open, key17, file17)

```

```
encryption(text_work_open, key18, file18)

encryption(text_work_open, key19, file19)

encryption(text_work_open, key20, file20)

index_sovpadeni(text_work_open, arr, ")

file.close()

file_var.close()
```

Розшифрування тексту

```
import collections

var = open("../variant.txt", "r", encoding='utf-8')

frequency_analis = open("../frequency_analis.txt", "w", encoding='utf-8')

text = var.read()

arr = ['a', 'б', 'в', 'г', 'д', 'е', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о', 'п',

      'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э', 'ю', 'я']

def index_sovpadeni(text1, keey):

    j = 0

    count = collections.Counter()

    for letter_text in text1:

        for letter_dict in arr:

            if letter_dict == letter_text:

                count[letter_dict] += 1

            break

    t = list(count)

    for i in range(len(count)):

        j += count[t[i]] * (count[t[i]] - 1)

    print("Ключ " + str(keey) + ": " + str((1/(len(text1) * (len(text1) - 1)) * j)) + "\n")

k = 0

while k < 30:

    k += 1

    index_n = 0

    text_ind = "

    while index_n <= len(text) - 1:

        text_ind += text[index_n]

        index_n += k

    index_sovpadeni(text_ind, k)

blocs = ["", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "",

          "", "", "", "", ""]

l = 0

ii = 0

stroka = "      # самые частые буквы в каждом блоке

while l < 15:

    l += 1

    index_n = ii

    text_ind = "

    while index_n <= len(text) - 1:

        text_ind += text[index_n]

        index_n += 15
```

```

blocs[l-1] = text_ind

print('Block ' + str(l) + ':' + blocs[l-1])

ii += 1

o = 0

count_text = collections.Counter()

while o < len(blocs[l-1]) - 1:

    letter = blocs[l-1][o]

    count_text[letter] += 1

    o += 1

frequency_analis.write(str(count_text) + '\n')

stroka += str(count_text)[10]

key = "

i = 0

while i < 15:

    key += arr[(arr.index(stroka[i]) - 14) % 32]

    i += 1

print('First key: ' + key)

for i in range(len(text)):

    if i % 15 == 0:

        frequency_analis.write('\n')

        frequency_analis.write(arr[(arr.index(text[i]) - arr.index(key[i % len(key)])) % 32])

frequency_analis.write('\n\n\n\n')

key = "

i = 0

while i < 15:

    if i == 6:

        key += arr[(arr.index('9') - arr.index('n')) % 32]

    else:

        key += arr[(arr.index(stroka[i]) - 14) % 32]

    i += 1

print('Right key: ' + key)

for i in range(len(text)):

    if i % 15 == 0:

        frequency_analis.write('\n')

        frequency_analis.write(arr[(arr.index(text[i]) - arr.index(key[i % len(key)])) % 32])

```

Висновок:

Під час данного комп'ютерного практикуму, ми засвоїли методи частотного криптоаналізу. Здобули навички роботи та аналізу поточкових шифрів та гамування адитивного типу на прикладі шифру Віженера.