

Міністерство освіти і науки України
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"
Фізико-технічний інститут

Криптографія

Комп'ютерний практикум №2
Криптоаналіз шифру Віженера

Варіант 3

Виконали:
Студенти 3 курсу
Загородній Я.М, Венгер П.Ю.

Перевірів:

Київ – 2024

Мета роботи: Засвоєння методів частотного криптоаналізу. Здобуття навичок роботи та аналізу поточкових шифрів гамування адитивного типу на прикладі шифру Віженера.

Постановка задачі:

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 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).

Хід роботи

1. Для шифрування був вибраний текст з сайту (<https://lingua.com/ru/russkiy/chteniye/vremena-goda/>) «Времена года», який був записаний в text_lab2.txt а потім за допомогою коду був редагований в текст написаний російською мовою без знаків пунктуації, великих літерів та пробілу, а буква «ё» була замінена «е». І даний текст був збережений в text_lab2_clean.txt .

Лабораторна робота №2

```
import collections
import unicodedata

with open("D:\\Web-developed\\Kript\\cp2\\text_lab2.txt", encoding="utf-8") as file:
    text = file.read()
    text = text.lower().replace(" ", "").replace("\n", " ")
    text = ''.join(text.split())

#without space
string_text = ''.join(c for c in text if unicodedata.category(c).startswith('L'))
print(string_text)

output_file_path = "D:\\Web-developed\\Kript\\cp2\\text_lab2_clean.txt"
with open(output_file_path, 'w', encoding="utf-8") as output_file:
    output_file.write(string_text)

print(f"Processed text saved to: {output_file_path}")
```

```
word_sort = sorted(string_text)
list_alp = []
for i in range(0, len(word_sort)):
    list_alp.append(word_sort[i])

alpcount = dict(collections.Counter(list_alp))
print("\n")
print_dict_in_columns(alpcount)
```

```
✓ 0.0s

a: 67  б: 3   в: 35  г: 11  д: 31  е: 70  ж: 8
з: 15  и: 64  й: 7   к: 15  л: 27  м: 42  н: 64
о: 78  п: 19  р: 48  с: 50  т: 49  у: 9   х: 7
ц: 5   ч: 8   ш: 22  щ: 14  э: 1   ю: 9   я: 27
```

2. Було створено ключи різної розмірності (2, 3, 4, 5, 10-20)

```
text_open = open("D:\\Web-developed\\Krp\\cp2\\text_lab2_clean.txt", encoding='utf-8').read()
keys = ['он',
        'она',
        'шифр',
        'слово',
        'информация',
        'кодвеслово',
        'закодировать',
        'розкодировать',
        'сотрудничество',
        'двузначноечисло',
        'трехзначноечисло',
        'шифраторивиженера',
        'операциинадчислами',
        'монограммахудожника',
        'автоматдлякодировки', -]
for key in keys:
    print(f"Ключ: '{key}', Кількість символів: {len(key)}")
```

✓ 0.0s

Ключ: 'он', Кількість символів: 2
Ключ: 'она', Кількість символів: 3
Ключ: 'шифр', Кількість символів: 4
Ключ: 'слово', Кількість символів: 5
Ключ: 'информация', Кількість символів: 10
Ключ: 'кодвеслово', Кількість символів: 11
Ключ: 'закодировать', Кількість символів: 12
Ключ: 'розкодировать', Кількість символів: 13
Ключ: 'сотрудничество', Кількість символів: 14
Ключ: 'двузначноечисло', Кількість символів: 15
Ключ: 'трехзначноечисло', Кількість символів: 16
Ключ: 'шифраторивиженера', Кількість символів: 17
Ключ: 'операциинадчислами', Кількість символів: 18
Ключ: 'монограммахудожника', Кількість символів: 19
Ключ: 'автоматдлякодировки', Кількість символів: 20

Потім був зашифрований відкритий текст за допомогою «Шифр Віженера», всі зашифровані тексти були записані в файли в папці **vigenere_encrypt**

```
def vigenere_encrypt(plaintext, key):
    ciphertext = []
    key_length = len(key)
    key_as_int = [ord(i) - ord(' ') for i in key]
    plaintext_int = [ord(i) - ord(' ') for i in plaintext if ' ' <= i <= 'я']
    for i in range(len(plaintext_int)):
        value = (plaintext_int[i] + key_as_int[i % key_length]) % 32
        ciphertext.append(chr(value + ord(' ')))
    return ''.join(ciphertext)

ciphertexts = {}
for key in keys:
    encrypted_text = vigenere_encrypt(text_open, key)
    ciphertexts[key] = encrypted_text
    print(f"Ключ: '{key}', Зашифрований текст: {encrypted_text}\n")

with open(f"vigenere_encrypt\\iphertext_{len(keys)}.txt", "w", encoding='utf-8') as file:
    file.write(encrypted_text)
print("Всі зашифровані тексти записані в файли.")
```

Output is truncated. If you are a [PowerShell](#) user, please open a [PowerShell](#) session in a [text editor](#). To test cell output settings, click on the [test cell output](#) button.

- index_of_coincidence_results)**

Key	Length key	Index_of_coincidence
Відкритий текст	BT	0,057952474
он	2	0,04900652
она	3	0,04167671
шифр	4	0,036417292
слово	5	0,040252155
информация	10	0,03303668
словеслово	11	0,03657798
закодировать	12	0,033963722
розкодировать	13	0,034532307
сотрудничество	14	0,034995828
двузначноечисло	15	0,033571274
трехзначноечисло	16	0,034930935
шифраторивиженера	17	0,034108958
операциинадчислами	18	0,033982263
монограммахудожника	19	0,034034795
автоматыдлякодировки	20	0,032931615

```

import pandas as pd

def index_of_coincidence(text):
    frequency = [0] * 32 # Для російських літер
    for char in text:
        if 'а' <= char <= 'я':
            frequency[ord(char) - ord('а')] += 1
    total_letters = sum(frequency)

    if total_letters == 0:
        return 0

    ic = sum(f * (f - 1) for f in frequency) / (total_letters * (total_letters - 1))
    return ic

# Створюємо словник для збереження результатів
indexes = {
    "Key": [],
    "Length key": [],
    "Index_of_coincidence": []
}

# Обчислення індексу відповідності для відкритого тексту
plaintext_ic = index_of_coincidence(text_open)
print(f"Індекс відповідності відкритого тексту: {plaintext_ic}")

# Додаємо відкритий текст в результати
indexes["Key"].append("Відкритий текст")
indexes["Length key"].append(len(text_open))
indexes["Index_of_coincidence"].append(plaintext_ic)

# Обчислення індексу відповідності для кожного шифротексту
for key, ciphertext in ciphertexts.items():
    ciphertext_ic = index_of_coincidence(ciphertext)
    print(f"Індекс відповідності для ключа '{key}': {ciphertext_ic}")

    # Додаємо результати в словник
    indexes["Key"].append(key)
    indexes["Length key"].append(len(key))
    indexes["Index_of_coincidence"].append(ciphertext_ic)

df = pd.DataFrame(indexes)
indexes_file = "index_of_coincidence_results.xlsx"
df.to_excel(indexes_file, index=False)

print(f"Результати записано в {indexes_file}")

```

```

Індекс відповідності відкритого тексту: 0.05795247365656191
Індекс відповідності для ключа 'он': 0.04900652019406075
Індекс відповідності для ключа 'она': 0.04167670961960385
Індекс відповідності для ключа 'шифр': 0.036417292419888135
Індекс відповідності для ключа 'слово': 0.040252155372207284
Індекс відповідності для ключа 'информация': 0.033036679954265936
Індекс відповідності для ключа 'словесло': 0.036577979666882976
Індекс відповідності для ключа 'закодировать': 0.03396372176385155
Індекс відповідності для ключа 'розкодировать': 0.03453230740706406
Індекс відповідності для ключа 'сотрудничество': 0.034995828311856866
Індекс відповідності для ключа 'двузначноечисло': 0.03357127406446031
Індекс відповідності для ключа 'трехзначноечисло': 0.03493093538518587
Індекс відповідності для ключа 'шифраторивиженера': 0.034108958314019965
Індекс відповідності для ключа 'операциинадчислами': 0.033982262600043264
Індекс відповідності для ключа 'монограмахудожника': 0.034034794969253115
Індекс відповідності для ключа 'автоматыдлякодировки': 0.032931615215846234
Результати записано в index_of_coincidence_results.xlsx

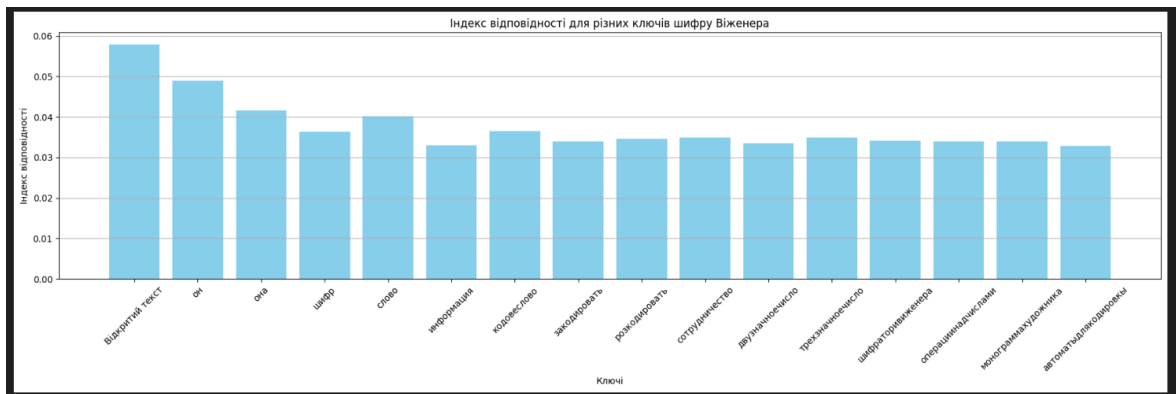
```

Побудую гістограму для демонстрації індексів відповідальності

```
import matplotlib.pyplot as plt

plt.figure(figsize=(18, 6))
plt.bar(df["Key"], df["Index_of_coincidence"], color='skyblue')
plt.xlabel('Ключі')
plt.ylabel('Індекс відповідності')
plt.title('Індекс відповідності для різних ключів шифру Віженера')
plt.xticks(rotation=45) # Повертаємо підписи по осі X для кращого вигляду
plt.grid(axis='y')

# Відображення гістограми
plt.tight_layout()
plt.show()
```



Для дешифрування тексту використаємо перший метод з методичних вказівок.

1. Читаємо файл з текстом

```
1 encryption_text = open('encrypted_text(3).txt', encoding='utf8').read().replace('\n', '')
2 print(encryption_text)
✓ [80] < 10 ms
```

2. Обчислюємо частоту літер у тексті

```
1 # Функція для обчислення частоти
2 def calc_letters_frequency(text: str):
3     letters_frequency = {c: text.count(c) for c in set(text)}
4     letter_frequency_dict = {c: round(count / len(text), 5) for c, count in
5                             letters_frequency.items()}
6     return letters_frequency, letter_frequency_dict
7
```


3. Обчислюємо індекс відповідності

```
def calc_index_of_coincidence(text):
    letter_counts, _ = calc_letters_frequency(text)
    total = sum(letter_counts[char] * (letter_counts[char] - 1) for char in letter_counts)
    index = total / (len(text) * (len(text) - 1)) if len(text) > 1 else 0
    return index
```

4. Розбиваємо текст на блоки

```
def create_blocks(text, key_length):
    blocks = []
    for i in range(key_length):
        block = ""
        for j in range(0, len(text), key_length):
            if i + j < len(text):
                block += text[i + j]
        blocks.append(block)
    return blocks
```

5. Знаходимо індекс відповідності для всіх можливих довжин ключа

```
def find_key_length_index(text):
    key_results = pd.DataFrame(columns=['Key Length', 'Index'])
    for key_length in range(2, 32):
        blocks = create_blocks(text, key_length)
        indices = [calc_index_of_coincidence(block) for block in blocks]
        avg_index = np.mean(indices)
        key_results.loc[len(key_results)] = {'Key Length': key_length, 'Index': avg_index}

    return key_results
```

6. Отримуємо таблицю з індексами. Шукаємо той, що найближче до індексу відповідності російської мови – 0,055

↕	Key Length	↕	Index	↕
0		2	0.032605	
1		3	0.032577	
2		4	0.032651	
3		5	0.032535	
4		6	0.032560	
5		7	0.032718	
6		8	0.032692	
7		9	0.032514	
8		10	0.032518	
9		11	0.032714	
10		12	0.032635	
11		13	0.054069	
12		14	0.052657	
13		15	0.032436	
14		16	0.032675	
15		17	0.032683	
16		18	0.032569	
17		19	0.032665	
18		20	0.032507	
19		21	0.032769	
20		22	0.032516	
21		23	0.032672	
22		24	0.032639	
23		25	0.032509	
24		26	0.053855	
25		27	0.032348	
26		28	0.032491	
27		29	0.032363	
28		30	0.032398	
29		31	0.032709	

Найближче – довжина ключа 13, тому обираємо її

7. Знаходимо можливий ключ на основі частотного аналізу

```
1 alphabet = ['а', 'б', 'в', 'г', 'д', 'е', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф',  
2             'х', 'ц', 'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э', 'ю', 'я']  
3  
4 def find_key(text):  
5     blocks = create_blocks(text, 13)  
6     key_options = []  
7  
8     for block in blocks:  
9         _, frequency_dict = calc_letters_frequency(block)  
10        most_frequent_letter = max(frequency_dict, key=frequency_dict.get)  
11        possible_letters = [  
12            alphabet[(alphabet.index(most_frequent_letter) - alphabet.index(reference_letter)) % 32]  
13            for reference_letter in ['о', 'е', 'и', 'а']  
14        ]  
15        key_options.append(possible_letters)  
16  
17    return key_options  
18  
19 find_key(encryption_text)
```


8. Отримуємо зсуви найбільш частотних літер відносно найбільш популярних символів у російській мові о, е, и, а

```
[['г', 'м', 'й', 'с'],  
 ['р', 'щ', 'ц', 'ю'],  
 ['о', 'ч', 'ф', 'ь'],  
 ['м', 'х', 'т', 'ъ'],  
 ['н', 'ц', 'у', 'ы'],  
 ['к', 'у', 'р', 'ш'],  
 ['а', 'й', 'ж', 'о'],  
 ['в', 'л', 'и', 'р'],  
 ['ь', 'е', 'в', 'к'],  
 ['д', 'н', 'к', 'т'],  
 ['у', 'ь', 'щ', 'б'],  
 ['м', 'х', 'т', 'ъ'],  
 ['а', 'й', 'ж', 'о']]
```

9. Підбираємо ключ. «громьковедьма»

10. Дешифруємо текст

```
def decode(text, key):  
    decrypted_text = []  
    key_length = len(key)  
    for i, c in enumerate(text):  
        decrypted_char = alphabet[(alphabet.index(c) - alphabet.index(key[i % key_length])) % 32]  
        decrypted_text.append(decrypted_char)  
  
    return ''.join(decrypted_text)  
  
decrypted_text = decode(encryption_text, 'громьковедьма')  
print(decrypted_text)
```

11. Отримуємо відкритий текст

старминскаяшколачародеевпифийитравницяфакультеттеоретическойипрактическоймагииикафедрдрамаговпрактиковчастьперваясоциальныйукладбгтинравывампирьейобщинывикачтовывчтотомеетепротиввампиоровраспринкорпорациямифкурсоваяработаадептквивосьмогокурсавольхиреднойнаучнийруководительмагистрпервойстепениархимагксанперловдевятсотдевяностодевятыйгодпобелороскомулетосчислениюгородстарминвведениехорошийсегоднявыдалсяденектеплыйбезветренныйвтораядекадасеноставамесяцанеспешносочиласьсквозьклепсидрусолнечноголетаиголосазябликовдоносившиесяизпридорожныхкустовзвенеливушахяхаласквозьлихнездывеугодыкаквдольпограничнойполосойбыладорогазаброшенныйпроклевывающийс пыльнойтравойкривойбольшакзбликипопеременновозмущалисьсвотворжениемчеловеканабелойлошадивихчастныевладениязалихватскистрелисменялисьхриплымчириканиемптахисуетливоперепархивалиповеточкамтревожалиствуразноцветнаякаймавокругчерныхподсыхающихлужвзрываласьсотнямиистомленныхжароймотыльковраскручиваласьвысывихремтрепещущихкрыльевповодьязавернутыпетлейсвисалиспереднейлукияпокачиваласьвседекакмешокскрупойпридерживаялевойрукойлежавшеенаколеняхписьмоипытаясьразобратьпрыгающиепередглазамируныромашкапользоваласьмоимрасслабленнымсостояниемвсезамедляязамедляшашагнадеятьсячтоувлеченнаячтениемнезамечуеесковарногоманевраидамейостановитьсяиспокойнопощипатьтравкутычегоэтоголубушкаанушеввелихпытаямиплутуютаякобылкарозачарованновсхрапнуладаваядаваяхалтурщицаустроиласьпоудобнейеслиобщеможноустроитьсяпоудобнейнатомпыточномпредметекоииявлялосьдляменяжесткоеказенноеседлонатретийденьпутиромашкин агриватоненькимиколечкамиспускаласьдопереднейлукизабываясьмеждустраницаминапухлогописьмакотороеядолжнабыла вручитьповелителюдогевыикотороеужеминутпятькаксамовольновскрылаприпомощимагиинетронуувесистойпечатинаверевочкенааломвоскоетчетливопроступалоттискперстятринадцатьрунипереплетающийсядракономединогорговцентретутмонзаятиялитератуroidипломатиейигенеалогиейгрубопрервалиоченьгрубаяедвауспелаподхватитьлисткипоползшиевразныестороныромашканеисправимаясбогатажницазадумчивожевалауздубрящаяжелезомвремякакнезнакомыйивесьмаподозрительныйтипобросшейнаружностидемонстративнопотрясалпередлошадийморданномодельнымбарбетомсгрязнойстрелоймногоразовогоиспользованиятакчтонепонятнобылокогоонсобираетсяграбитьменяилиромашкуяприподняласьнастременахсинтересомрассматриваязаржавлен

нийнаконецникнянедумаючтоэтосамоеудачноеместодляторговлиантиквариатомдоверительносообщилаянезнакомцувоствармин
еуасбьегосрукамиторваливернееотрубилизнастелитамоченьнелюбятразбойниковромашкаобнюхалаарбалетпрезрительнофыр
кнулаинапрочьигнорируяграбителяпотянуласькаппетитнойзеленималинникаизвысокойгушикотороготолькочтовозниклоэточуд
овлаптыхпреступныйэлементзаметносмутилсянаконецникзатрепеталкакщеньчийхвостикувидораскаianiaипокаяниябылоещедал
екозаблудшаяовцаупорствовалавогрехесребролюбияануткаживослезайсконядевкаязыкатаякошелекижизньдапошустрейслыш
ишьязобразилаусиленнуюработумыслиладноубедилкошелекпахнулоозономлицограбителяпередернулосьзрачкierasширилсыг
лазастекленелионмедленноопустиварбалетотвязалибеспрекословноподалиметощиймешокболтавшийсяупоясатмешкаразило
кошкамиикуревомослабивверевкустягивавшуюгорловинуяпропустиласквозьпальцынесколькомелкихмонетмаловатодорогоймо
ймаловатосленцойработаетшьбезогонькавпрочемтакужибытьвозьмувкачествеавансаосчастливилаяграбителяшвыряемуподноги
пустоймешокипредупреждаячерезпаруднейэтойжедорогойназадпоедутакужбудьдобрпостарайсяменянеразочароватьмужикнео
трываютменязагипнотизированноговзглядамедленнонагнулсяподнялмешокизастылстолбстолбомневсилахшевельнутьсябезмое
говедомакактолькогореграбительскрылсяизвидуадеактивировалазаклинаниеипозволиларамашкеперейтисгалопаналюбимуюею
трусцуписьможажатоесовремяподсчетадегенуменямеждудоколеняминемногопомялосьиутратилотоварныйвидвпрочемрассудилаяг
лавноееформлениеисодержаниеоноежекомпенсировалонедостаткирепейноголистаиспользованноговукромномместеагавотна
концеиобменепарастрокзадифирамбамизагадочномуаррактурупропустишьинезаметишьзавремяобучениявысшейшколещекард
еевпифийитравницадепткавольхапроявиласебянаоченьплохонеусидчиванетерпеливасвоевольназнакомаяпеснялюбитзлыешу
ткиинеоднократнопереноситихвоспитанниковнавоспитателейэтоонпроведрочтолидабылоодноведеркодольнообъемистоесто
ялосебенабалкенаддверьюмоейкомнатыэдакийсамодельныйкапканнасоседейпошкольномуобщезитиудабьнеповаднобылобезс
просуодалживатьуменияконспектыикастриюлиснавареннымнанеделюборшомможетучительтакбынеразозлилсяеслибыведровсета
кипрокинулосьанеупалоесунаголовустоймявместесводойотличаетсясредкимиспособностямикпрактическойитеоретическоймаг
ииисильноразвитойинтуициейбыстроадаптируетсякнестандартнойситуацииахажетяещенебезнадежнаеприличнаякакаятогран
ицадоугевыульфоввысокиетравыугномовскалыувадлаковгрудывыбошеннойнаповерхностьземлиудриададубподметгающиеоб
лакаудруидовкамненекругилюдейоблупленныестеныканалысзатхлойводойразделенныепаройтройкойподъемныхмостовдалы
сыестражникипринихбдительнодремлющиеупираясьнаржавыеалебардыздесьосиниздевательствокакоетоособенноесличь
чтожителидогевывампирыххорошиетакиеосинисеребристыетрепещущиезаосинамищекочетнебоостроверхийеловыйковерсредик
оторогокоедепроглядываютазатравленныеберезкиисосенкисамажедоговалелжитвдолинекакплюшканаднерасписнойпиалыеслис
мотретьсхоломакраяпиалывиденбелыйободокизосинвторойпотолщепотемнееизелейавцентреширокоезеленоеодноскрапочкамиса
мадогевавкольцевозделанныхполейиоблакахтуманаподойдешьвплотнуюокдеревьямнаставлялменяучителипошлешьмысленный
сигналвглубьлесалобойможешьдуматьочемуднолишьбыформироватьмощнуютелепатическуюволнукакумнеенаправитьна
аобщейчастотектонибудызстражейграницыуслышитсямущеннокашлянулалучшебыемуэтогонеслышатьнеобязательнопродумы
ватьочереднуюпакостьзнаюзнаютнанихсверхвсякоймерыгоразданонасейразпостарайсявоздержатьсяотоныхочемэтоахдаовол
невампирыоченьвосприимчивыктелепатииисразуотреагируютнаееприсутствиехотяиисмогутдоскональнорасшифроватьтакто
напирайнаколичествоаненакачествовоттакясмотрюнадымашуюбанюнаморщивлоботусердиянамоюволнутутжереагируютпят
илишестьадептовкоторыесоевнянныепаромвыбегаютиздверейивыпрыгиваютизоконатакованныевнезапноожившимивеникамирук
ибудущихколлегазаятышайкамприкрывающимиотвениковсамоеокровенноеучительусмиряетвеникиоднимдвижениембровин
овзглядыадресованныешутниценедомытмиколлегамесулятигниегохорошегоказалоподуматьанетранслироватьзаклинанияжа
льчтозагодыпроведенныевэтихстенахтытакиненаучиласьдуматьчтождумаюстоюподосинойнаморщивлобиромашкаужечтотожу
етзеленаясюнасосчитсяизчерныхуголковбархатистыхгубразделенныхкольцамиидилтелепатироватьзначитсознательноделиться
мыслямискемнибудьдругимделюсьпоследнимизлесатянетпрохладойсидящаянаветкеиволгаудивленнопокачиваетхвостомвответ
намоиумственныепотугилибозанятиеоказалосьмнепозубамлибоощаренныестражиграницыпопадалинаместесраженнымое
ймощнойдумоймоистаранияувенчалисьуспехомминутчерезсорокизаетовремяуспелапередуматьбольшечемзапредыдущиевосе
мнадцатьлетавотирезультатагаподействовалоилионпроходилмимослучайнаявпервыеувиделавампиравозможноеслибыонвозник
изниоткудабылблденкаксмертьинедвусмысленноскалилокровяленныезубыбьегоиспугаласькаксобственнопланироваламои
знаниявобластивампироведениябазировалисьначеловеческихлегендахипреданияхотличавшихсяредкостнымпессимизмомктому
жевсегравирыкартиньгобеленынакальнаяживописьизображаютвампировисключительноночьюивтемнотекрыльязубыкогтивсе
этокажетсятакимстрашнымогромнымтолькопотомучтотолкомничегонельзяразглядетьдневнойсветразвелялореолужасавпухипр
ахприсолнечномсветенафонебескрайнихполейивысокихдеревьеввампирыпоказалисьмневозмутительномелкимибезобиднымиправд
аяещенеспешиласьапришлосьмнегалантнопредложилирукувоспользоватьсякоторойвпрочемянерискнулавампирулыбнулсяпока
завдлинныеклыклялобойулыбнулсбябыувидевкакаясползласьехалапокрутомуromaшкиномубокуперекинувповодьячерезголовуло
шадиявыжидающеуставиласьнавампирастражграницыоказалсявышеменянаполголовышироковплечахивысшемандуренсобойдлин
ныетемныеволосыобрамлялиузкоезагорелоелицосложенныезаспинойкрыльяпридаваливампирунекотороесходствосмороемдемо
номпосланникомсмертидесятиаршиннаястатуякоторогоукрашалаактовыйзалвысшейшколычерныепронзительныечутьраскосые
глазавампираизучилимоюмалопривлекательнуювнешностьнотакисумелиразгадатьчтозанейскрыто

Висновок

Засвоїли методи частотного криптоаналізу. Здобули навички роботи та аналізу
потоків шифрів гамування адитивного типу на прикладі шифру Віженера.