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

Виконали:
Студенти гр. ФБ-11
Поліщук Олександра
Маленко Сергій
3 варіант

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

Структура нашого проекту:

- Main.py – містисть основні класи для роботи з текстом, а також графік 2 пункту.
- Key_length.py – визначення довжини ключа(пункт 3)
- Key_value.py – обчислення частоти букв у зазначеній довжині (пункт 3)
- Decipher.py – ф-ція розшифрування
- Task.txt – 3 варіант
- Input.txt – обраний нами текст

Порядок виконання роботи:

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.

Обраний нами текст міститься у файлі input.txt. Задля шифрування ми знову використали підхід за допомогою класів – метод очищення тексту та метод шифрування. Оскільки кожен раз ми генерували рандомно ключ, тому і результати шифрування вийшли різними.

Приклади:

Key: дигоп (length – 5)

лчоягттвхдозтягтхноохэгаъсцссгиучвцчнапцсьувочёяофчжоочсойнривьфжыгюпгъуувпнжгмцзрудзиьчот
цввффжзвоёьзнвёчзнртлгввцкснъгктоуднхнсгцлугчзжорязлцрипгвкгцгбюинннъгюрчцицлъопнёапсзтяьм
мгуггцсрйнзсаэиирнттыстохксчёгъууоченъгъсбфпжзвввзрооуюфвйсцсэотьхаёузнсефлцъгихьпсэлжфху
ссгъспффнййнгквцпфчфьнызпчбцчеяыгысчопнёапсзфяфюйофцзфуризъчцмццнтзнвчотйлвпиэвбгьхобяв
бэчлсъоцнпюочйчюгётчгиысэомзёарижюйрзрймъицюупнртрзсвфпдрчзиьхргюльквхгнвоыгчсдёхньссёчо
мзтажюоыъгъснриызсдгхсангкснсфнпмоочхябявяягюжяспнхрфжзвохкслохэуовцдвкяцчпяэтлгнфжчвыз
ыжмсвюпхнняюявлиечуткгьоегвюфгчжчягцгвдфиочвцзуовхугцзмувжпхэзюкочвюпёнгуггьеяфжчвтбчлг
нсгнёяохууяюсчпнцмфлуоёзстъсзлцоцинчёгъучётмсроднёапсзлнгихряьтплъоситубийсъофифбьдрюрпбв
яоучфьфзцзъозчдизизкяэтэсэокюнуоочхябтлснързцтппчфйосивтяв в нхгвкгыгбвуыгчсджвяоучжаецсбгв
вапхьночэснчдхзжпиэвжгтзоутфирнстьтаъсспофцз вдгцге зуцньдувбздьхьъёчзнафнжцядхзю ёирчфгнёя

Key: пчуодцьъат (length – 10)

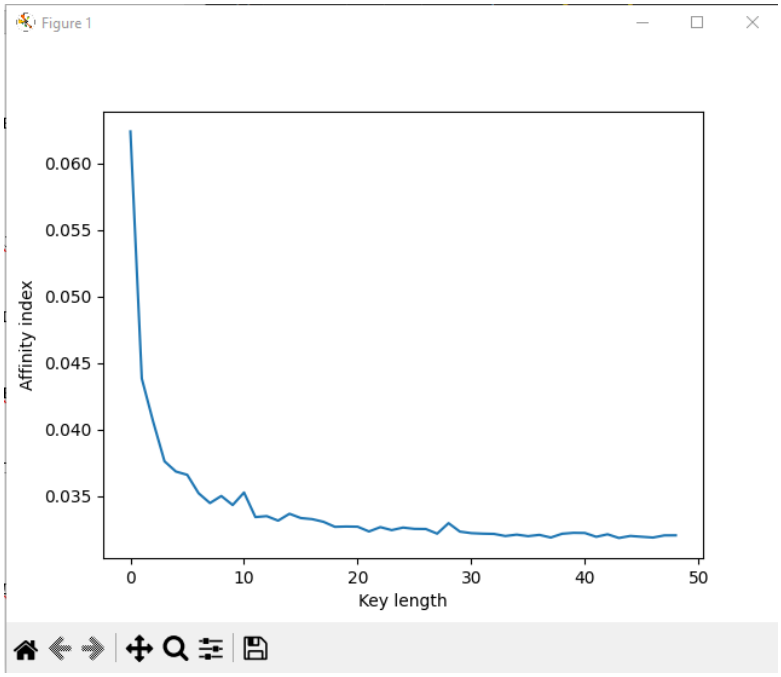
чжаяцёечжжьцдяцёздасвкуамеизгвочечхййдртгблухвйэосбжчогкдеьрючтьиылънтозеухгаюупгцвучььрис
ётвизъятссйънхъйя у ьувхйюз эоыдозцал фоёюуцкыюауйцюцеэвътноёубрэад
эолвчкэивйсэюцадеыиояъэууцхизвмыцгапэьж х игтгиюзииозеузвйь
эозгбигъятенцвогжоктмяёгэгёмлрвсгънёчжвзэочжьденвччвггстхкзбчбюьйвхючзтбжёьвмыёидгжхянхлзи
сэюцадеыковбльоийыкеуфцйчкаим
хуютчгёэвттфктбтхмладйетбпкязйсгюбютвйаиаофдчцэлзмсъццатдьюнмюцвйбльмнцэювттдызтчэтвчээм
лвволюь фзъ еьиучёцчл яябцчгаыирвтлаымхкз
уфиъсчхззрроыгнёзаёясьжжяфррчоболчяёгалввбхъвгиюзюсвкеохйхчэбгжбяпёяъ
чтжтытгжякйъбтюдиадоайнт т ььиц ыуьгчфчнчожччсхиътжбчачхйыйаевгуцэаёччтвкъю
вйчнтсюууцхмьочтжттфкяъ фоюцягиёйоаяжбнкажвессцгтмеывзсгч чыхкйии эгргцаэртяцюнцэзжоя
юъгеьиедфъгъгзъксяпанрдунчосажёьиьияйсужфиээыбо кгэгяодесьжжяфёяз
эюцзтдгйкьсячттсфрчпвыеув хлъсеаиучёцъчосажчатчизсённтадимдаьзб
нлцзячтфктжцёыеехбчвнёёмирэяббоийыфтжоёуетьём япгтбэцмллэсжънузаюзбпеъютъьжичоюця

Key: еуёыфъоффзихэжоююужй (length – 20)

мгсиззъуээуфлхвлй жицжёкяжюдчцзхмпбпл чйчюя ждячджыеажнрёавзсуеёыэафгззёмлбивчьжчту июомяжчдьыуньъыижёкъжьяыуичъэыбп гётдххыъъузуйзделврэчжк товхяхфзгздэъэвв ётдзугэяюя жфь чокэдхфнчё
зчюдцвтфкчьвювёмуефяукрдяязёмлтзлхётддфлыемызыжфйжнмрёыдтвфжузбзёцкббёряиюотдуъёфжвям мьбкйяэмгзохъмфтчруюзыентмтэбпьчгзи чадяжфь
чокэёхъхзмыылнжыинфспхёкзёмиъегузпяззниьъяэожжь белбмсд жэьифязльпцдрхгзяадвжсф чяйючфдстуфсноиглфьйкяйэгыороугмябзццзйетйьйуёэпеёгцълзутц пнёэдчъззсеяч
дуинжбйгэйгчздхфъцйувтжуеохацттхчдзйицеузццыубыно гдицжцыжлйурхэеихьлауёозгеедеьыа с ьфоов хц йейдбоцялччэтняцтфодгйггчюфзэыхзпбпэежэц ёвмвыумзьибфйзлтфйжъё
зчбцдмлеькарауёлдъииукыёдфцеиёхёиюлижтфяяжняюжэхжпеэмепаучфэуъучёзцфеёввйвхфуьодужоеыъ нцкрннюёъфеэаэфхвуджъентубкъриигжумянюдучикундр
липгъиёзсдурхфпкоилёвитуеягцеуецтгэыйэнуъэфеёгцъмуджъеачяякгъюгйекфкбафпб жёцюйъ ййжесззнбыкыхйёрлодчттютыылнрзэздэюябззёфе елмъбзбркебё нвчочебкздьогямзь х

2. Підрахувати індекси відповідності для відкритого тексту та всіх одержаних шифртекстів і порівняти їх значення.

Задля цього ми використали лібу matplotlib. Ми згенерували 50 ключів різної довжини та порівняли як довжина ключа співвідноситься до індексу відповідності.



На графіку видно, що чим довший ключ, тим важче розшифрування даного тексту.

3. Використовуючи наведені теоретичні відомості, розшифрувати наданий шифртекст (згідно свого номеру варіанта)

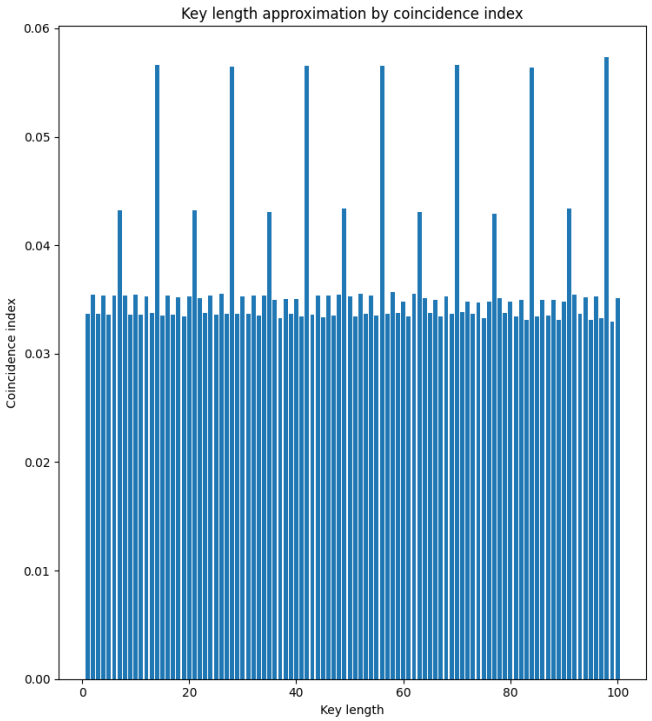
Варіант – 3

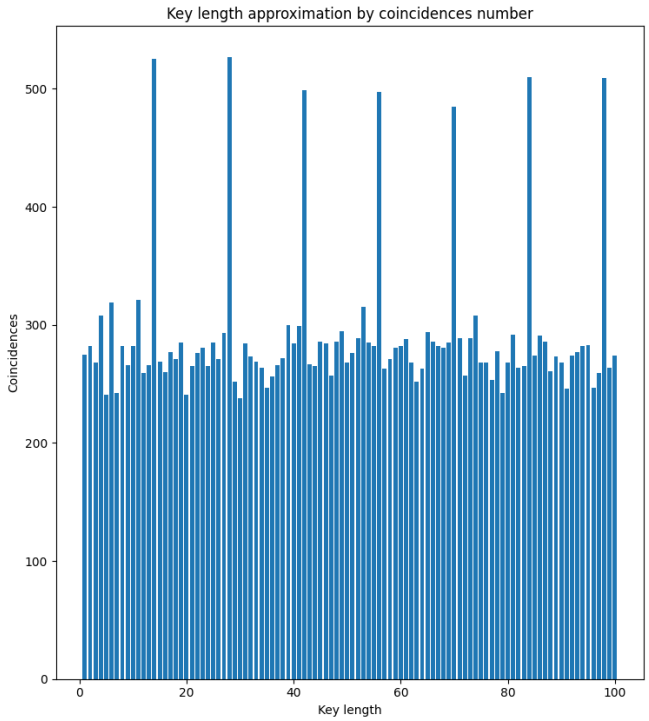
Варіант міститься у файлі task.txt. Першочергово, потрібно було встановити довжину ключа для розшифрування даного тексту. Методичні вказівки до даної лабораторної роботи надають нам 2 різні методи для розпізнання довжини, отже ми реалізували обидва для порівняння результату.

Код у key_length.py отримує з консолі довжину ключа задля перевірки у цьому діапазоні. Перевіряємо у діапазоні зі 100

```
SlavyaSanek@DESKTOP-RIIS35D MINGW64 ~/Documents/GitHub/crypto-23-24/cp2/malenko_fb-11_polishchukO_fb-11_cp2 (cp2)
$ python -u "c:\Users\SlavyaSanek\Documents\GitHub\crypto-23-24\cp2\malenko_fb-11_polishchukO_fb-11_cp2\key_length.py"
Enter the maximum key length to check for (0 to check all possibilities, 'q' to exit): 100
```

За допомогою matplotlib отримуємо 2 графіки

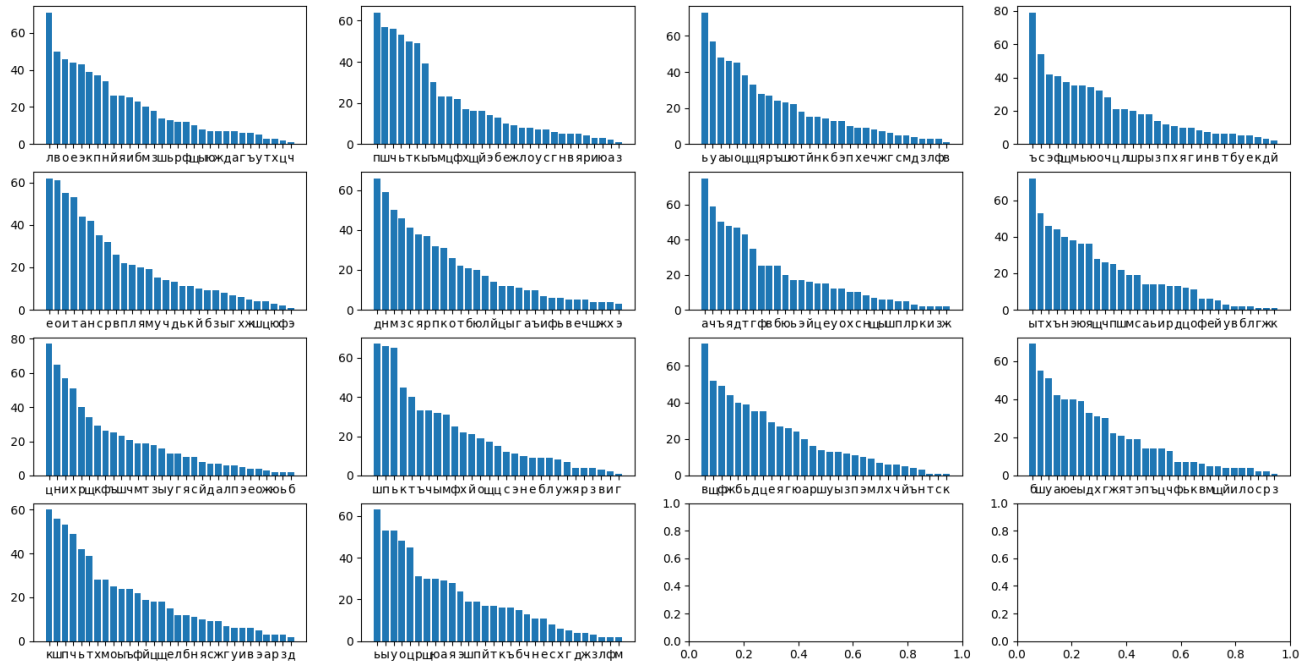




На обидвох графіках помітно, що проміжки 14, 28, 42… чітко вирізняються. Другим за к-стю співпадінь є проміжок з 7, що давало нам припущення, що ключ може бути довжиною у 7, що йшло всупереч тому, що відображав графік. Другий графік не дав настільки впевненого результату, проте все одно лідирує 14. Отже ми дізнались довжину нашого ключа.

Тепер була задача у встановлені його значення. Задля цього key_value.py розбивав текст на шифрблоки введеної довжини, обчислював найчастіші букви, що займали позиції 1, 2, … n та виводив графік для кожної з позицій.

```
SlavyaSanek@DESKTOP-RIIS35D MINGW64 ~/Documents/GitHub/crypto-23-24/cp2/malenko_fb-11_polishchukO_fb-11_cp2 (cp2)
$ python -u "c:\Users\SlavyaSanek\Documents\GitHub\crypto-23-24\cp2\malenko_fb-11_polishchukO_fb-11_cp2\key_value.py"
Enter the key length to check for letter frequencies ('q' to exit): 14
The most likely key of length 14 - "эбомчцтникфуьо"
```



З цього графіку ми забрали найчастіші букви з позицій, отримали «лпьъедаыцшвбкь». Далі за допомогою найчастішої букви у російському алфавіті (букви «о») ми працювали за наступним алгоритмом: нам потрібно було перетворити кожну з найчастіших букв у «о», а тобто, знайти для цього ймовірний ключ. Ми використали наступну формулу:

$$k = (y - x) mod m, \text{де } y - \text{позиція поточної літери у алфавіті}, x - \text{позиція літери "о"}, \text{а } m - 32 \text{ літери, виключаючи літеру ё}$$

k – це індекс можливої літери ключа. Таким чином, ми отримали наступний ключ:

"эбомчцтникфуьо".

Поки що ключ не виглядає логічно(тому ми й сумнівались у довжині ключа), проте ми спробували підставити його у decipher.py для розшифровки тексту.

```
SlavyaSanek@DESKTOP-RIIS35D MINGW64 ~/Documents/GitHub/crypto-23-24/cp2/malenko_fb-11_polishchuk0_fb-11_cp2 (cp2)
$ python -u "c:\Users\SlavyaSanek\Documents\GitHub\crypto-23-24\cp2\malenko_fb-11_polishchuk0_fb-11_cp2\decipher.py"
Enter the key to decipher the text ('q' to exit): эбомчцтникфуьо
The deciphered text is: иыутиьвиделмоятцикбйрвисящщйнйдофмойнитиьпувенцчйсвольаыхчралсзохроныомлелсаииописйвафкофобани
яяхнаф...
Enter the key to decipher the text ('q' to exit):
```

З першої спроби стає зрозуміло, що ми на вірному шляху. «видел», «висящщй» вже перетворюються на логічний текст. Ми брали перші 14 літер «иыутиьвиделмоя» і намагалися співставити їх до реального тексту. Одразу можна сказати які літери є вірними – «э_ _ _ _ _ тникфу_ _». Ми мали сумнів щодо логічності інших, для простішої роботи у файлі key_value.py ми вирішили порівняти найчастіші літери не з «о», а з «е» (другою по частоті). Звідти отримали наступне: «жкчхаяыцсуэьеч». Приберемо звідси літери в яких ми впевнені: «_кчхая_ _ _ _ _ _еч» і співставимо разом

«экчхаятникфуеч». Одразу, можна помітити слово «маятник» та «эко». Використаймо decipher.py.

```
$ python -u "c:\Users\SlavyaSanek\Documents\GitHub\crypto-23-24\cp2\malenko_fb-11_polishchukO_fb-11_cp2\decipher.py"
Enter the key to decipher the text ('q' to exit): экомаятникфуеч
The deciphered text is: итутяувиделмецтникшарвисящнанадолгойнитиужущеннойсвольчтхоравизохронтемвеличиииописащалколебани
яямдал...
Enter the key to decipher the text ('q' to exit):
```

«И тут я увидел мецтник..» В нас залишились 2 літери, проте ми віришили загуглити «эко маятник» і дізнались, що це твір італьянського письменника Умберто Эко «Маятник Фуко». Звідси наш ключ

экомаятникфуко

```
яямдал...
Enter the key to decipher the text ('q' to exit): экомаятникфуко
The deciphered text is: итутяувиделмаятникшарвисящийнадолгойнитиопущеннойсвольтыхоравизохронномвеличиииописывалколебани
яязнал...
Enter the key to decipher the text ('q' to exit):
```

И тут я увидел маятник шар висящий надолго й нити опущенной свольты хора в изохронном величии описывал колебания я знал...

Висновки: В ході комп’ютерного практикуму, ми досліджували шифр Віженера та його реалізацію в обох напрямках – шифрування та розшифрування. Експериментували з різними підходами до знаходження довжини ключа та за допомогою коду та логічного мислення знайшли значення для розшифрування тексту.