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

КРИПТОГРАФІЯ

Комп’ютерний практикум

Робота № 3

Виконали

студенти гр. ФБ-06,

Зінов'єв Андрій, Даценко Валерія

Київ - 2022

Мета роботи

Набуття навичок частотного аналізу на прикладі розкриття моноалфавітної підстановки; опанування прийомами роботи в модулярній арифметиці.

Порядок виконання комп'ютерного практикуму

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

1. Реалізувати підпрограми із необхідними математичними операціями: обчисленням оберненого елементу за модулем із використанням розширеного алгоритму Евкліда, розв’язуванням лінійних порівнянь. При розв’язуванні порівнянь потрібно коректно обробляти випадок із декількома розв’язками, повертаючи їх усі.

2. За допомогою програми обчислення частот біграм, яка написана в ході виконання комп’ютерного практикуму №1, знайти 5 найчастіших біграм запропонованого шифртексту (за варіантом).

3. Перебрати можливі варіанти співставлення частих біграм мови та частих біграм шифртексту (розглядаючи пари біграм із п’яти найчастіших). Для кожного співставлення знайти можливі кандидати на ключ (a, b) шляхом розв’язання системи (1).

4. Для кожного кандидата на ключ дешифрувати шифртекст. Якщо шифртекст не є змістовним текстом російською мовою, відкинути цього кандидата.

5. Повторювати дії 3-4 доти, доки дешифрований текст не буде змістовним.

Хід роботи

Варіант роботи - №4.

Спочатку написали функції для: підрахунку оберненого елемента в кільці за модулем за допомогою розширеного алгоритму Евкліда, розв'язання лінійних рівнянь.

Далі ми зберегли зміст зчитаних файлів *(V4, 04.txt)* у окремі зміні прибравши пробіли.

Написали код, що розв'язує систему лінійних рівнянь для пошуку пари ключів a, b, що використовується у розшифруванні ШТ та відсікає непотрібний ВТ за принципом заборонених біграм, збережено у окремий файл *lab3.py*.

Код складається з декількох функцій:

1. Функція *func()* і *next()* - пошук коефіцієнтів для розширеного алгоритму Евкліда та одержання оберненого елементу.
2. Функція *roots()* - шукає усі дійсні корені лінійного рівняння за модулем.
3. Функція split() - поділяє текст на блоки (біграми).
4. Функції *replace_bigrams(), adding_bigram_list(), adding()* - з лабораторної роботи №1
5. Функція *statistics()* - повертає список найрозповсюджених біграм заданого тексту, паралельно перевіряючи на вміст заборонених біграм, тобто ще й фільтрує.
6. Функція *X_value()* - повертає значення X_i таке, що для біграми (x_{2i-1}, x_{2i}) це $X_i = x_{2i-1}m + x_{2i}$.
7. Функція *get_ab()* - знаходить ключі a і b для обраних пар біграм у мові та тексті.
8. Функція *decrypt()* - розшифрування біграми та переведення з індексів у літери.
9. Функція *start()* - виводить ВТ, запускаючи decrypt() для кожного блоку (біграми) ШТ з ключами a, b.
10. Функція *go()* - вибирає пари біграм з найпоширеніших з мови та для них утворює пари з найпоширеніших біграм з ШТ, виводить шматок ВТ та ключі.

Використовуючи код з лабораторної роботи №1, можна знайти найпоширеніші біграми з текстів:

1. Файл *V4*: `['щь', 'ез', 'ьв', 'ди', 'ся']`
2. Файл *04.txt*: `['еш', 'еы', 'шя', 'ск', 'до']`

Після того як утворили списки найпопулярніших біграм, можна приступити до функції go() що утворює усі можливі пари пар біграм, що не повторюються, з найрозповсюдженіших у мові та ШТ. Також зі згенерованих таких пар запускає інші функції (*get_ab(), X_values(), start()*) про них мова піде далі.

Застосувавши функцію *get_ab()*, щоб знайти для пар біграм ключі a, b, програма розв'язує систему лінійних рівнянь:

$$\begin{cases} Y^{*} \equiv aX^{*} + b \pmod{m^2} \\ Y^{**} \equiv aX^{**} + b \pmod{m^2} \end{cases},$$

$$Y^{*} - Y^{**} \equiv a(X^{*} - X^{**}) \pmod{m^2}.$$

Звідси:

$$a = (Y^* - Y^{**})(X^* - X^{**})^{-1}(mod m^2).$$

$$b = (Y^* - aX^*) \bmod m^2 .$$

Але в першу чергу, *get_ab()* отримує значення Y^*, Y^{**}, X^*, X^{**} за допомогою обчислень функції *X_values()*, принцип якої базується на формулі:

$$(x_{2i-1}, x_{2i}) \leftrightarrow X_i = x_{2i-1}m + x_{2i} .$$

Після того як отримали ключі a, b, пробуємо розшифрувати. Процес розшифрування ґрунтується на роботі двох функцій: *start(), decrypt()* - перша вилучає біграми з ШТ та передає на другу з ключами a,b, а друга розшифровує біграму за оберненою формулою до тієї що застосовується у X_values(): нехай x, y - індекси літер біграми ВТ, а m - довжина алфавіту, тоді:

$$\begin{cases} y = \text{Xi}(\text{mod}m) \\ x = (\text{Xi} - \text{y})\text{mod}m \end{cases}$$

На виході у *decrypt()* маємо строку, а у *start()* весь ВТ.

Отже, у результаті маємо:

1. Файл *V4*.
   Ключі a = 5, b = 960; ВТ:
   лодостьеленывбездействиивнешнемвовнутреннейборьбеитревогепод ругунеенебылоизовсехдевицпосещавшихдомстаховыхонанесошласьни соднойродительскаявластьникогданетяготеланаделеной…

2. Файл *04.txt*:
   Ключі a = 390, b = 10; ВТ:
   еслиправдачтодостоевскийвсибиринебылподверженприпадкамтоэт олишьподтверждаетточтоегоприпадкибылиегокаройонболеевнихне нуждалсякогдабылкараеминымобразомнодоказатьэтоневозможно…

З повним розшифрованим текстом можна ознайомитись у файлах *V4_encrypted.txt* та *04_encrypted.txt*.

Висновки: У ході даної лабораторної роботи була освоєна робота з частотним аналізом тексту моноалфавітної підстановки. Реалізовано функції з теорії чисел, а саме на принципі модулярної арифметики, за допомогою яких та частотного аналізу був розшифрований афінний шифр. А також, було застосовано створений нами фільтр від неможливих біграм у ВТ, що значно прискорив пошук правильних ключів.