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-22 Пунько Артем
гр.Фб-22 Руденко Поліна

Київ – 2024

Мета роботи:

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

Хід роботи:

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

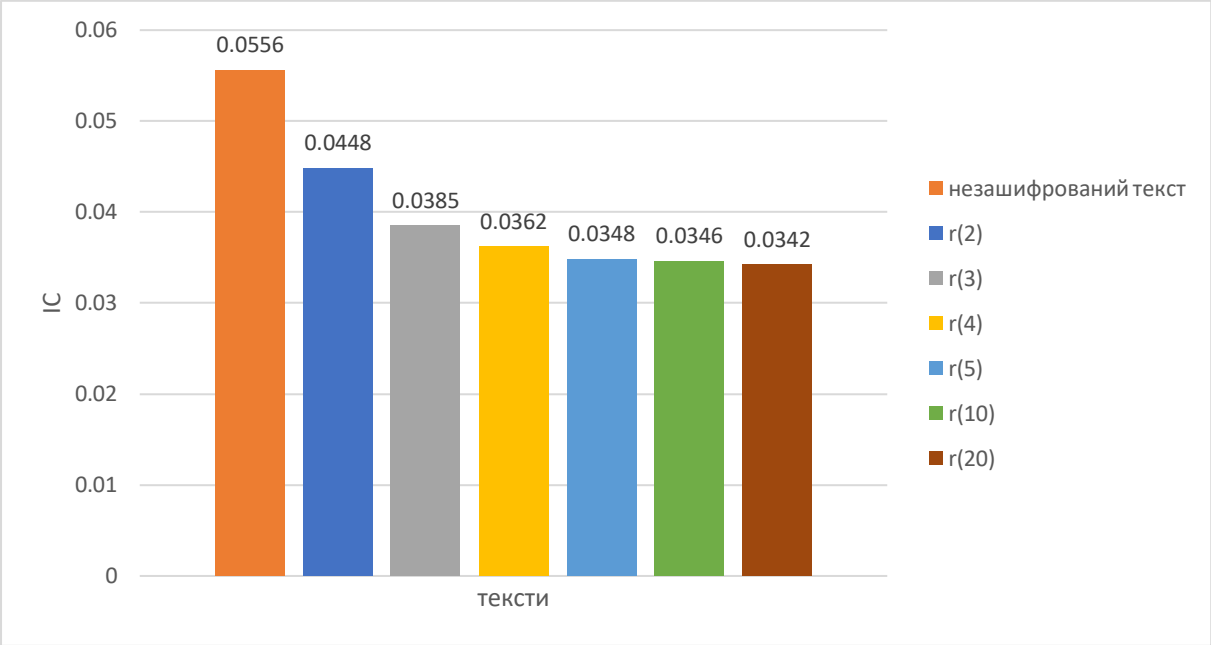
1. Самостійно підібрати текст для шифрування (2-3 кб) та ключі довжини r = 2, 3, 4, 5, а також довжини 10-20 знаків. Зашифрувати обраний відкритий текст шифром Віженера з цими ключами.

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).

1+2

(виконали перший та другий пункт в одному фрагменті коду)



Ми зашифрували відкритий текст шифром Віженера за допомогою ключів довжинами від 2-х до 20-ти і отримали індекси відповідності зашифрованих текстів.

Ключі: ок, хол, пиво, мышка, ялюблюпиво, фурионпофармучемпион.

Шифрування відбувається в такий спосіб:

```
std::string result(fileSize.QuadPart, 0);

for (size_t i = 0; i < bytesRead; i++) {
    if (buffer[i] >= 'а' && buffer[i] <= 'я') {
        int charIndex = buffer[i] - 'а';
        int shift = key[i % key.size()] - 'а';
        int newIndex;

        if (decrypt) {
            newIndex = (charIndex - shift + 32) % 32;
        }
        else {
            newIndex = (charIndex + shift) % 32;
        }

        result[i] = 'а' + newIndex;
    }
    else {
        result[i] = buffer[i];
    }
}
```

Щоб обрахувати індекс відповідності ми підраховували кількість всіх літер в тексті і кількість кожної окремої літери

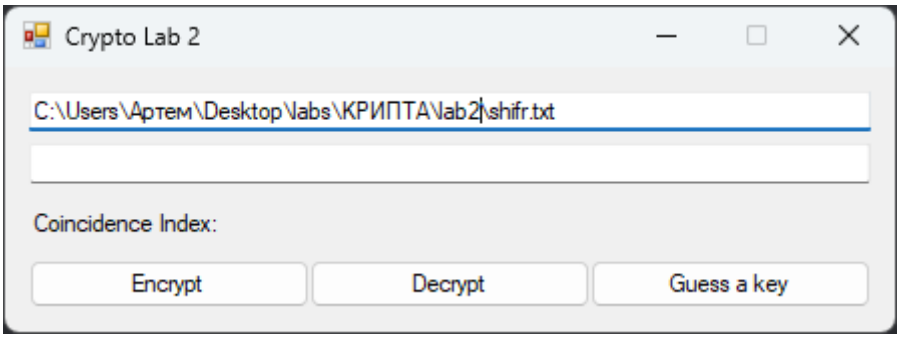
```
std::map<char, int> freq;
int totalAlphabetChars = 0;

for (size_t i = 0; i < result.size(); i++) {
    if (result[i] >= 'а' && result[i] <= 'я') {
        freq[result[i]]++;
        totalAlphabetChars++;
    }
}

if (totalAlphabetChars < 2) {
    this->CoincidenceIndicesLabel->Text = "Coincidence Index: 0";
}
else {
    double indexCoincidence = 0.0;
    for (const auto& pair : freq) {
        indexCoincidence += (pair.second * (pair.second - 1.0));
    }
    indexCoincidence /= (totalAlphabetChars * (totalAlphabetChars - 1.0));

    this->CoincidenceIndicesLabel->Text = "Coincidence Index: " +
        System::Convert::ToString(System::Math::Round(indexCoincidence, 4));
}
```

І використовували дану формулу

$$I(Y) = \frac{1}{n(n-1)} \sum_{i \in Z_m} N_i(Y)(N_i(Y) - 1),$$

Тобто ми підсумовували кількість кожної окремої літери помножуючи її на її кількість мінус один, і потім ділили цю суму на всю довжину тексту помножену на довжину тексту мінус один

3

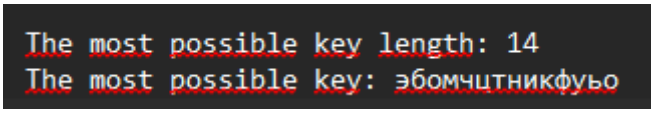


Щоб намагатися вгадати довжину ключа і сам ключ, треба ввести у верхнє поле шлях до нашого зашифрованого тексту і натиснути кнопку *„Guess a key„* . Після цього ми отримаємо файл з аналізом цього тексту.

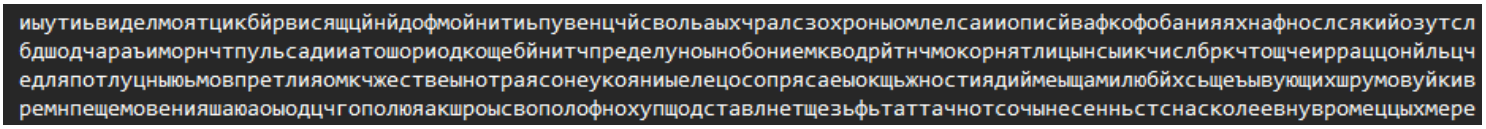
```
2: 0.035403
3: 0.033658
4: 0.035375
5: 0.033601
6: 0.035353
7: 0.043212
8: 0.035366
9: 0.033618
10: 0.035441
11: 0.033570
12: 0.035279
13: 0.033743
14: 0.056671
15: 0.033551
16: 0.035329
17: 0.033553
18: 0.035251
19: 0.033429
20: 0.035329
21: 0.043282
22: 0.035166
23: 0.033776
24: 0.035362
25: 0.033598
26: 0.035505
27: 0.033646
28: 0.056468
29: 0.033693
30: 0.035294
```

Можна побачити аномальне підвищення індексу відповідності при розбитті тексту на 14 груп, і при числах кратних 14-ти, а також аномальне, хоч і не таке велике, підвищення цього ж індексу при розбитті тексту на 7 груп і так при всіх розбиттях кратних 7, не включаючи парних чисел(бо парні числа це є кратні 14, що є довжиною нашого ключа).

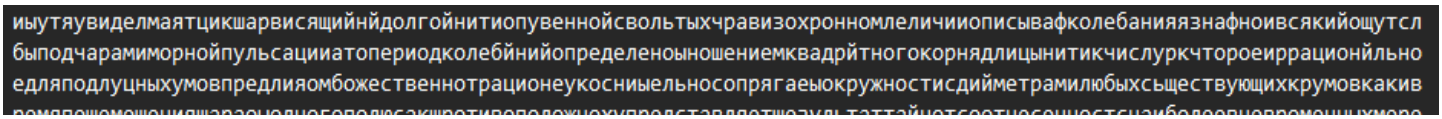
В результаті ми отримали вірогідну довжину ключа і, вірогідно, сам ключ



якщо Ви спробуєте розшифрувати шифр цим ключем, то можна побачити, що текст вже частично набрав якогось сенсу

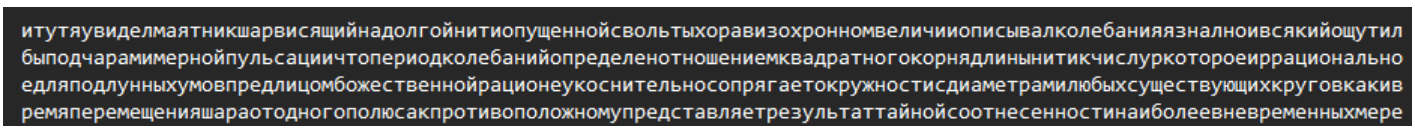


Якщо прочитати текст, то можна зробити висновок, що частичка ключа ,,тникфу,, є скоріш за все вірною. Якщо відштовхуватись від підсказки, що кожний ключ є змістовним, у мене одразу з'явилась асоціація останніх 4 літер "Фуьо" з французькою фамілією Фуко. Якщо загуглити " мчцтник фуко" можна отримати таку річ як "Маятник Фуко". Якщо прогнати ключ "эбомаятникфуко", то можна побачити що текст майже набув сенсу.



Але по перших словах видно помилку в другій літері, можна припустити, що малось на увазі слово «тут». Тоді вийде фраза «и тут я увидел маятник»

Так як буква «Ы» стоїть на дев'ять позицій вправо від букви «Т», то букву «Б» треба здвинути на дев'ять позицій вправо, отримавши букву «К». Вийде ключ «экомаятникфуко»



Тепер текст повністю без помилок!

До речі, в цілому вже на другому пункті можна було спробувати використати методику з третього пункту, де ми просто рахували відступи від букв, але нам одразу прийшла в голову асоціація з прізвищем і ми одразу використали цей метод, але не відмітити можливість чисто математичного способу ми не могли.

Висновок:

Завдяки цій лабораторній роботі ми навчились працювати з шифром Віженера: дізнались що таке індекс відповідності, навчились шифрувати і дешифрувати шифром Віженера, а також взламувати цей шифр завдяки теоретичним знанням криптології.