Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Фізико-технічний інститут

Криптографія

Лабораторна робота №3

Варіант 1

Криптоаналіз афінної біграмної підстановки

Виконали:
студенти 3 курсу ФТІ
групи ФБ-05
Качур Ілля Ковальов Данііл

Перевірила:
Селюх П.В.

Київ – 2022

**Мета роботи**:

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

**Порядок виконання роботи**

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**

1. З цим кроком справилися достатньо швидко, брали код операцій із інтернету та потім з’єднали в єдину функцію

```
def uravnenie(a, b, n):
    a, b = a % n, b % n
    d = nsd(a, n)
    x = []
    if d == 1:
        x.append((bezout(a, n) * b) % n)
        return x
    else:
        if b % d != 0:
            return x
        else:
            a, b, n = a // d, b // d, n // d
            x.append((uravnenie(a, b, n)[0]))
            for i in range(1, d):
                x.append(x[-1] + n)
```

Також поглядували на дані у методичці формули

Нехай $ax \equiv b \pmod{n}$ і треба встановити значення $x$ за відомими $a$ та $b$. Маємо такі випадки:

1) $\gcd(a,n) = 1$. В цьому випадку порівняння має один розв'язок: $x \equiv a^{-1}b \pmod{n}$.
2) $\gcd(a,n) = d > 1$. Маємо дві можливості:
   2.1) Якщо $b$ не ділиться на $d$, то порівняння не має розв'язків.
   2.2) Якщо $b$ ділиться на $d$, то порівняння має рівно $d$ розв'язків $x_0$, $x_0 + n_1$, $x_0 + 2n_1 \ldots$, $x_0 + (d-1)n_1$, де $a = a_1 d$, $b = b_1 d$, $n = n_1 d$ і $x_0$ є єдиним розв'язком порівняння $a_1 x \equiv b_1 \pmod{n_1}$: $x_0 = b_1 \cdot a_1^{-1} \bmod n_1$.

2. Для знаходження біграм була використана змінена функція (бо мені не подобалось як була зроблена 1 робота, тому змінив у цієї та потім першу у цілому).
   Найчастіші біграми:
   ***['рн', 'ыч', 'нк', 'цз', 'иа']***
3. З цим кроком також багато проблем не було, нам підказали як зробити це швидко, тому багато ми на нього часу не витратили, шукали найчастіші біграми у мові та нашого тексту

```
mostpopularRUletter = ['ст', 'но', 'ен', 'то', 'на']
mostpopularRUtext = negr(text)
print(mostpopularRUtext)
bigrams, systems_uravneniyas = [], []
for i in mostpopularRUletter:
    for j in mostpopularRUtext:
        bigrams.append((i, j))
```

   , потім додали виключення щоб не переходили одна біграма в іншу

```
for j in bigrams:
    if i == j or (j, i) in systems_uravneniyas:
        continue
    elif i[0] == j[0] or i[1] == j[1]:
        continue
```

4. Знайшли можливі ключі, перевіряли по ентропії та відсіювали результати незмістовного тексту. Ентропія тексту, перевіряв у межах від 4 до 4.5, так як стандартна енропія рус мови, здається десь 4.3

```
def entropy(opentxt):
    amountofletters = Counter(opentxt)
    for i in amountofletters:
        amountofletters[i] /= len(opentxt)
    answer = -1 * sum(float(amountofletters[i]) * log(amountofletters[i], 2) for i in amountofletters)
    return answer


def correctkey(keys, cyphertext):
    no_matches = "net sovpadeni"
    for i in keys:
        e = entropy(superdupersecretAfinne(cyphertext, i))
        if (e > 4.0) and (e < 4.5):
            return i
    return no_matches
```

5. Знайшли пари ключа

```
(13, 151)
```

## Висновки

У ході виконання лабораторної роботи покращили навички частотного аналізу у випадку *моноалфавітної* підстановки. Написали програму що розшифровує афінний шифр за допомогою біграм. Повторили знання що стосувались теорії чисел.