НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО"
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

КРИПТОГРАФІЯ

КОМП’ЮТЕРНИЙ ПРАКТИКУМ №3

«Криптоаналіз афінної біграмної підстановки»

Виконали
студенти 3 курсу
групи ФБ-21
ДЗИСЮК Владислав
ТЕЛУХ Анастасія

КИЇВ – 2024

Варіант 8

Мета роботи: Ознайомлення з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA; практичне ознайомлення з системою захисту інформації на основі криптосхеми RSA, організація з використанням цієї системи засекреченого зв'язку й електронного підпису, вивчення протоколу розсилання ключів.

Порядок виконання роботи:

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. ***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.***

   Для реалізації розширеного алгоритму Евкліда ми створили функцію euclid. Їй просто треба надати два числа a і b для коректної роботи. Приклад її роботи:

```
Розширений алгоритм Евкліда для 30 і 18: НСД = 6
```

   Функція linear_solve, натомість, була створена для розв'язання лінійних порівнянь. Цій функції необхідні три числа – a, b, mod. Результатом роботи

буде число (чи числа, якщо розв'язків декілька), яке і є коренем лінійного порівняння. Приклад роботи:

```
Лінійне порівняння 14x ≡ 30 (mod 100): Розв'язок = [45, 95]
```

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

   Використовуючи частину коду з комп'ютерного практикуму №1, ми трошки модифікували його для того, щоб він відображав 5 найчастіших біграм для наданого шифротексту.

   Після запуску коду для файлу із зашифрованим текстом 08.txt, ми отримуємо такий список:

```
Найчастіші біграми
Топ-5 біграм у тексті: ['жц', 'дэ', 'цэ', 'сц', 'оц']
```

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).***

   У методичних вказівках до комп'ютерного практикуму, були наведені найпоширеніші біграми російської мови. Саме їх ми будемо використовувати для подальшого співставлення:

```
freq_bigr = ['ст', 'но', 'то', 'на', 'ен']
```

   Окремого списку для топ_5 біграм ми не створювали вручну, лише під час вже запуску коду ініціалізуємо автоматичне знаходження за допомогою функції analyse_bigr і неявне зберігання у програмі:

```
top_5_bigram = [bigram[0] for bigram in analyse_bigr(file_path, symbol)]
```

   Всі можливі варіанти ключів для нашого шифротексту обчислюються і перебираються за допомогою функції key_decrypt.

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

   При спробах дешифрування нам необхідно знаходити коректний текст серед іншого шуму. Вручну зробити це важко, тому треба створити автоматичний розпізнавач російської мови.

   Серед критеріїв автоматичного визначення змістовного тексту ми вирішили обрати критерій заборонених l-грам (у нашому випадку – біграм). Вона працює шляхом аналізу тексту і спроб знайти там біграми, які не властиві

мові цього тексту. Для російської мови, яка фігурує у цьому завданні, такі біграми теж існують. Деякі з них ми і вирішили використати:

```
if all(bigram not in decrypted for bigram in ['аь', 'еь', 'юы', 'яы', 'эы', 'юь',
                                              'яь', 'оь', 'иь', 'ыь', 'уь', 'аы',
                                              'эь', 'ць', 'хь', 'кь', 'оы', 'иы',
                                              'ыы', 'уы', 'еы']):
    candidates.append(((a, b), decrypted))
```

Якщо при аналізі отриманого тексту такі біграми присутні – дана пара ключів виявляється не правильною. У іншому випадку, коли ці біграми відсутні – пара (a, b) додається до списку кандидатів.

Внаслідок такого підходу ми отримали результат:

```
Результати розшифрування:
Розшифрований текст (перші 100 символів): мальчикизаулыбалисьисжаромвзялисьзаделоонирвализолотистыецветыцветычтонаводняютвесьмирпереплескивают
```

Видно, що текст змістовний навіть із наведених 100 символів.

Отже, ми можемо впевнено стверджувати, що для зашифрованого тексту 08.txt знайденим ключем є:

```
Знайдений ключ: (17, 94)
```

Висновок. Отже, у ході виконання даного комп’ютерного практикуму детальніше дослідити афінну біграмну підстановку, попрактикувалися як коректно реалізувати низку математичних операцій. Протягом виконання цієї роботи, нас спіткали низка труднощів, але головними із них була реалізація розпізнавача російської мови. Були випробувані й інші критерії змістовного тексту, такі як перевірка частот частих і рідкісних літер, але виділити саме змістовний текст таким чином ми не змогли. Тільки використання заборонених біграм дало нам змогу швидко й ефективно реалізувати розпізнавач, який значно полегшив нам виконання цього практикуму.