Національний технічний університет України
«Київський політехнічний інститут»
Фізико-технічний інститут

Криптографія

Комп’ютерний практикум №3

Криптоаналіз афінної біграмної підстановки

Виконали:

студенти групи ФБ-32

Грабовецький Микита

Драбок Алла

Київ - 2025

Мета роботи:

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

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.

Варіант: 13

Хід роботи:

Необхідні математичні операції реалізовані таким чином:

- **Розширений алгоритм Евкліда** - реалізований у функції **def egcd(a: int, n: int).** Вона повертає (gcd, u), де gcd — це НСД(a, n), а u — це коефіцієнт, який використовується для знаходження оберненого елемента
- **Обчислення оберненого елемента** - реалізовано у **def mod_inverse(a: int, n: int)**. Вона викликає egcd(a, n) і перевіряє, чи gcd == 1. Якщо так, вона повертає u % n (це і є a^(-1) (modn)), інакше повертає None
- **Розв'язування лінійних порівнянь** - це ключова функція **def solve_linear_congruence(a:0 int, b: int, n: int).**
    - Вона знаходить d = НСД(a, n) за допомогою egcd.
    - **Якщо d=1** - рівняння a * b = (конгруентно) b (mod n) має один розв'язок. Код знаходить a^(-1) і повертає [(a_inv * b) % n].
    - **Якщо d > 1** - рівняння має розв'язки, тільки якщо b ділиться на d (b % d == 0).
        - Якщо **ні**, повертається порожній список [] (розв'язків немає).

- Якщо **так**, то рівняння має рівно d розв'язків. Код розв'язує менше рівняння (a/d) * x =(конгруентно) (b/d) (mod(n/d)), знаходить один розв'язок x0, а потім генерує всі d розв'язків за формулою x_0 + k * (n/d) для k від 0 до d-1. Це і є коректна обробка декількох розв'язків.

Нижче можемо побачити 5 найчастіших біграм шифртексту за варіантом. Обраховували із кроком 2, тобто біграми, що не перетинаються.

```
Топ-5 біграм шифртексту:
'аф': 0.01512
'яф': 0.01512
'дю': 0.01226
'ап': 0.01169
'нф': 0.01169
```

Тепер ми знайшли найчастіші біграми шифртексту і для кожного співставлення знайшли 306 кандидатів на ключ (a,b) шляхом розв’язання системи, що описано на початку.

```
Найчастіші біграми мови: ['ст', 'но', 'то', 'на', 'ен']
Найчастіші біграми шифртексту: ['аф', 'яф', 'дю', 'ап', 'нф']

Знайдено 306 унікальних кандидатів
Кандидати успішно збережено у файл 'key_candidates.txt'
```

Тепер для кожного кандидата дешифрували шифртекст за формулою: x = (конгруентно) a^(-1)*(Y-b) (mod M^2). Щоб зрозуміти, чи є результат змістовним текстом, маємо функцію (def calculate_fitness(text: str)), що вимірює, наскільки частоти літер 'о' та 'е' у розшифрованому тексті близькі до їхніх еталонних частот у російській мові. Вона повертає помилку (суму квадратів відхилень). Тож чим менше це число, тим "змістовнішим" є текст з точки зору статистики. У main() код відстежує best_score = float('inf'). Якщо поточний score (помилка) менший за best_score, код "відкидає" старий найкращий ключ і запам'ятовує новий, кращий. За допомогою циклу for a, b in keys проходимось по всіх кандидатах і тільки потім робимо висновки. Змінні best_key та best_score гарантують, що в кінці в best_text буде збережено текст, який мав найменшу помилку (тобто був найбільш змістовним) серед усіх кандидатів.

```
Завантажено 306 ключів-кандидатів з 'key_candidates.txt'
Зараз буде щось цікаве

Знайдено найкращий ключ (a, b): (99, 60)
Оцінка (чим менше, тим краще): 5.813684e-05

Початок дешифрованого тексту:
раннееранneeутропервыеотсветызаринакрышезаокномвсе
```

Як бачимо вже з початку дешифрованого тескту, він є змістовним, тож завдання виконано правильно і ключ також знайшовся правильно. Повний текст додали файлом у проєкт.

Висновки:

Під час виконання комп'ютерного практикуму, було успішно проведено повний цикл криптоаналізу афінного біграмного шифру і досягнуто мети роботи: набуття навичок частотного аналізу та опанування прийомів роботи в модулярній арифметиці.

На першому етапі аналізу, за допомогою програми обчислення частот, було знайдено 5 найчастіших біграм шифртексту: «аф», «яф», «дю», «ап», «нф».

На другому етапі було реалізовано перебір співставлень цих біграм із 5 найчастішими біграмами російської мови («ст», «но», «то», «на», «ен»). Шляхом розв'язання систем лінійних порівнянь було згенеровано 306 унікальних ключів-кандидатів (a, b), які були збережені у файл.

Оскільки ручна перевірка такої кількості кандидатів є неможливою, було розроблено автоматичний розпізнавач змістовного тексту. Ця функція (calculate_fitness) реалізує "критерій частих l-грам", оцінюючи близькість частот літер 'о' та 'е' у розшифрованому тексті до їхніх еталонних значень у російській мові. Текст із найменшою "помилкою" вважався найкращим з оброблених. Застосувавши цей критерій до кожного з 306 кандидатів, вдалося ефективно відсіяти неправильні ключі й ідентифікувати єдиний правильний ключ (a, b) = (99, 60). Дешифрування шифртексту за допомогою цього ключа дало змістовний текст російською мовою, що підтверджує коректність проведеного аналізу.