

Автоматический криптоанализ шифра Цезаря на русском языке с использованием частотного анализа

Языков Кирилл Александрович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье рассматривается реализация классического шифра Цезаря с поддержкой русского алфавита и метод автоматического криптоанализа, основанный на статистическом сравнении частот букв зашифрованного текста с эталонными частотами русского языка. Проведен экспериментальный анализ точности восстановления исходного текста и обсуждены возможности применения данного метода для простейших криптографических задач.

Ключевые слова: шифр Цезаря, криптоанализ, частотный анализ, русский язык, криптография, автоматический взлом, статистика текста

Automatic cryptanalysis of the Caesar cipher in Russian using frequency analysis

Yazykov Kirill Alexandrovich

Sholom-Aleichem Priamursky State University

Student

Abstract

Russian alphabet implementation of the classical Caesar cipher and the method of automatic cryptanalysis based on statistical comparison of letter frequencies of the encrypted text with reference frequencies of the Russian language are considered in the article. An experimental analysis of the accuracy of restoring the source text has been carried out and the possibilities of using this method for the simplest cryptographic tasks have been discussed.

Keywords: Caesar cipher, cryptanalysis, frequency analysis, Russian language, cryptography, automatic hacking, text statistics.

1 Введение

1.1 Актуальность

С развитием цифровых технологий возрастает потребность в понимании основ криптографии и методов взлома классических шифров. Несмотря на свою простоту, шифр Цезаря остается одним из базовых примеров в обучении информационной безопасности. Изучение алгоритмов автоматического взлома с использованием статистических методов помогает лучше понять принципы криптоанализа и методы защиты информации.

1.2 Обзор исследований

В работе А.А. Оленева, В.В. Малиатаки, К.Т. Тынчерова и М.В. Селивановой описано применение системы компьютерной алгебры Maple для выполнения криптоанализа классического шифра Цезаря. Авторы продемонстрировали использование базовых команд Maple для реализации двух подходов: метода грубой силы и частотного анализа. Эти инструменты предлагается включить в образовательные курсы по дискретной математике, теории чисел и криптографии, что подчеркивает их педагогическую ценность и практическую применимость [1]. Статья Д.В. Карпова, Э.Р. Абузяровой и С.Г. Ласковой посвящена практическому измерению времени шифрования и дешифрования при использовании шифра Цезаря на различных ПК. Авторы разработали собственное приложение на языке C++ в среде C++Builder XE7 и провели анализ производительности алгоритма в зависимости от конфигурации компьютера. Результаты показывают, что даже такой простой алгоритм может демонстрировать различную эффективность в зависимости от технических характеристик оборудования [2]. В исследовании М.А. Протьюко рассмотрен инновационный подход к криптоанализу шифра Цезаря с использованием генетических алгоритмов. Работа содержит теоретическую основу для применения методов эволюционного моделирования в задачах дешифровки, что открывает перспективы для более интеллектуальных и адаптивных систем взлома классических шифров [3]. А.Р. Лепшокова в своей статье представила разработку интерактивного приложения, предназначенного для наглядной демонстрации работы шифра Цезаря. Это решение направлено на повышение наглядности и вовлеченности при изучении основ криптографии, и может быть полезным как для преподавателей, так и для студентов в учебном процессе [4]. Работа Л.З. Гумеровой, Г.Н. Агламзяновой и Е.С. Маисеевой иллюстрирует метод взлома шифра Цезаря с использованием полного перебора ключей, также известного как метод грубой силы. Авторы приводят пример реализации данной техники и акцентируют внимание на её простоте и эффективности при ограниченном ключевом пространстве, что делает её классическим примером для обучения основам криптоанализа [5].

1.3 Цель исследования

Разработка и анализ программы, реализующей шифрование и дешифрование текста с использованием шифра Цезаря на русском языке, а также автоматический взлом данного шифра с применением частотного анализа.

2 Материалы и методы

В работе использован алгоритм шифрования методом сдвига букв (шифр Цезаря) с поддержкой русского алфавита (33 буквы). Для криптоанализа реализован частотный анализ текста, при котором вычисляется распределение частоты букв в зашифрованном тексте и сравнивается с эталонным распределением для русского языка.

Метод оценки качества расшифровки основан на минимизации суммы квадратов разницы между наблюдаемыми и эталонными частотами букв. Исследование реализовано на языке программирования Python с использованием библиотеки matplotlib для визуализации.

3 Результаты и обсуждения

1. Импорт необходимых библиотек (рис. 1).

```
import collections
import matplotlib.pyplot as plt
```

Рисунок 1 – Импорт библиотек

`collections` — используется для подсчёта частоты букв (`Counter`).
`matplotlib.pyplot` — для построения графиков частот букв.

2. Определение алфавита и эталонных частот (рис. 2, 3).

```
rus_lower = 'абвгдеёжзийклмнопрстуфхцчщъыьэюя'
rus_upper = rus_lower.upper()
```

Рисунок 2 – Код для подсчета частоты букв и построения графиков

`rus_lower` — русский алфавит в нижнем регистре (33 буквы).

`rus_upper` — тот же алфавит, но в верхнем регистре (для поддержки заглавных букв).

```
# Частоты букв в русском языке (примерные, для анализа)
standard_freq = {
    'а': 0.1097, 'б': 0.0845, 'в': 0.0801, 'г': 0.0735, 'д': 0.067, 'е': 0.0626,
    'ж': 0.0547, 'з': 0.0473, 'и': 0.0454, 'к': 0.044, 'л': 0.0349, 'м': 0.0321,
    'н': 0.0298, 'о': 0.0281, 'п': 0.0262, 'р': 0.0201, 'с': 0.019, 'т': 0.0174,
    'у': 0.017, 'ф': 0.0165, 'х': 0.0159, 'ц': 0.0144, 'ч': 0.0121, 'ш': 0.0097,
    'щ': 0.0094, 'ъ': 0.0073, 'ы': 0.0064, 'ь': 0.0048, 'э': 0.0036, 'ю': 0.0032,
    'я': 0.0026, 'ь': 0.0004, 'ё': 0.0004, 'Ъ': 0.0174 # повтор для полноты
}
```

Рисунок 3 – Код для учета регистра букв

`standard_freq` — словарь с частотами букв русского языка на основе статистики.

Используется для сравнения с частотами букв в анализируемом тексте.

3. Функция шифрования текста шифром Цезаря (рис. 4).

```
def caesar_cipher_rus(text, shift):
    result = []
    for char in text:
        if char in rus_lower:
            idx = rus_lower.index(char)
            shifted_idx = (idx + shift) % len(rus_lower)
            result.append(rus_lower[shifted_idx])
        elif char in rus_upper:
            idx = rus_upper.index(char)
            shifted_idx = (idx + shift) % len(rus_upper)
            result.append(rus_upper[shifted_idx])
        else:
            result.append(char)
    return ''.join(result)
```

Рисунок 4 – Код шифрования

Проходит по каждому символу в тексте. Если символ — русская буква в нижнем регистре, смещает её на `shift` позиций по алфавиту, с циклическим переходом. Аналогично для заглавных букв. Все остальные символы (пробелы, знаки препинания) оставляет без изменений. Возвращает зашифрованный текст.

4. Функция подсчёта частоты букв в тексте (рис. 5).

```
def frequency_analysis(text):
    letters_only = [c.lower() for c in text if c.lower() in rus_lower]
    counter = collections.Counter(letters_only)
    total = sum(counter.values())
    freq = {char: count / total for char, count in counter.items()} if total > 0 else {}
    return freq
```

Рисунок 5 – Код подсчета частоты букв в тексте

Извлекает из текста только русские буквы в нижнем регистре. Считает количество вхождений каждой буквы (`Counter`). Делит количество каждой буквы на общее число букв, чтобы получить относительную частоту. Возвращает словарь с частотами.

5. Функция вычисления степени соответствия частот эталону (рис. 6).

```
def score_frequency(text_freq):
    # Считаем сумму квадратов разницы частот с эталоном
    score = 0
    for char in standard_freq:
        score += (text_freq.get(char, 0) - standard_freq[char])**2
    return score
```

Рисунок 6 – Код вычисления степени соответствия частот эталону

Для каждой буквы из эталонного словаря считает квадрат разницы между частотой в тексте и эталонной. Суммирует все такие квадраты в итоговый показатель ошибки. Чем меньше `score`, тем ближе частоты текста к эталонным (настоящему русскому языку).

6. Функция автоматического взлома шифра Цезаря (рис. 7).

```
def crack_caesar(ciphertext):  
    best_shift = None  
    best_score = float('inf')  
    best_decoded = None  
  
    for shift in range(len(rus_lower)):  
        decoded = caesar_cipher_rus(ciphertext, -shift)  
        freq = frequency_analysis(decoded)  
        score = score_frequency(freq)  
        if score < best_score:  
            best_score = score  
            best_shift = shift  
            best_decoded = decoded  
    return best_shift, best_decoded
```

Рисунок 7 – Код автоматического взлома шифра Цезаря

Перебирает все возможные сдвиги от 0 до 32 (длина алфавита). Для каждого сдвига расшифровывает текст с обратным сдвигом `-shift`. Выполняет частотный анализ расшифрованного текста. Сравнивает частоты с эталоном и вычисляет оценку качества. Запоминает сдвиг с минимальной ошибкой и соответствующий текст. Возвращает лучший сдвиг и расшифрованный текст.

7. Функция построения графика частот (рис. 8).

```
def plot_frequency(freq, title="Частотный анализ"):  
    if not freq:  
        print("Нет русских букв для анализа.")  
        return  
    chars = list(freq.keys())  
    values = list(freq.values())  
    plt.bar(chars, values)  
    plt.title(title)  
    plt.xlabel("Буквы")  
    plt.ylabel("Частота")  
    plt.show()
```

Рисунок 8 – Код построения графика частот

Принимает словарь с частотами и заголовок графика. Строит столбчатую диаграмму распределения букв по частоте. Позволяет визуально оценить распределение символов.

8. Основной блок программы (рис. 9).

```

if __name__ == "__main__":
    text = input("Введите текст для шифрования: ")
    shift = int(input("Введите сдвиг (целое число): "))

    encoded = caesar_cipher_rus(text, shift)
    print(f"\nЗашифрованный текст:\n{encoded}\n")

    # Расшифровка с известным сдвигом
    decoded = caesar_cipher_rus(encoded, -shift)
    print(f"Расшифрованный текст (известный сдвиг):\n{decoded}\n")

    # Автоматический взлом (криптоанализ)
    best_shift, cracked_text = crack_caesar(encoded)
    print(f"Автоматический взлом шифра Цезаря:")
    print(f"Найденный сдвиг: {best_shift}")
    print(f"Восстановленный текст:\n{cracked_text}\n")

    freq = frequency_analysis(encoded)
    plot_frequency(freq, title="Частотный анализ зашифрованного текста")

```

Рисунок 9 – Основной блок программы

Запрашивает у пользователя исходный текст и сдвиг. Зашифровывает текст указанным сдвигом и выводит результат. Выполняет расшифровку с известным сдвигом для проверки. Запускает автоматический криптоанализ, который пытается угадать сдвиг без подсказки. Выводит найденный сдвиг и восстановленный текст. Строит график частот букв в зашифрованном тексте.

9. Пример использования (рис. 10).

Введите текст для шифрования: Оставьте мне лето: лишь в этой купели Мне было свободно, плескалось вольготно... Там Вага и солнце, - никто там не делит Все то, что по праву любви достается: Бескрайность и нежность, доверчивость взлета, Когда нарастаю поля и пределы. Когда так не важно, откуда ты, кто ты - Обычное зернышко, - много ли дела? Но помниться все, но дается наружу Лишь то, что касается счастья и лета. Оставьте мне это единство и дружбу, - Бумажные письма, в которых приветты От всех, кто встречался на тех первопутках, Где так незаметно и все ж непреложно Дарились на счастье века и минуты, Как первая ягода в ждущей ладошке.

Введите сдвиг (целое число): 5

Зашифрованный текст:
Уццежбчй стй рйчу: рнэб ж вчуо пшфйрн Стй ёару цжуёуиту, фрийцперуцб журбзучту... Чес Жезе н цуртый, - тнпчу чес тй ийрнч Жцй чу, ьчу фу фхежш ргёжн иуцчейчд: Ёйцпхеотуцб н тйлтучб, иуцкежбчй жерй че, Пуаме текецелзш фурд н фоййра. Пуаме чеп тй желту, учаме ча, пчу ча - Уеытуй мйктазпу, - стузу рн ийре? Ту фустнбцц жй, ту иейчд тежшш Рнэб чу, ьчу пеейчд цеццбд н рйче. Уццежбчй стй вчу ййнтпчу н ишйшв, - Баселтай фйцбсе, ж пучуха фоййча Уч жйц, пчу жцройерцд те чйй фйкжфйцнел, Эй чеп тйеисйнту н жй л тйфйрлуту Иейнрцб те цеццбй жйне н сйпча, Пеп фйкед дзуме ж лйкюю ре иуэпй.

Расшифрованный текст (известный сдвиг):
Оставьте мне лето: лишь в этой купели Мне было свободно, плескалось вольготно... Там Вага и солнце, - никто там не делит Все то, что по праву любви достается: Бескрайность и нежность, доверчивость взлета, Когда нарастаю поля и пределы. Когда так не важно, откуда ты, кто ты - Обычное зернышко, - много ли дела? Но помниться все, но дается наружу Лишь то, что касается счастья и лета. Оставьте мне это единство и дружбу, - Бумажные письма, в которых приветты От всех, кто встречался на тех первопутках, Где так незаметно и все ж непреложно Дарились на счастье века и минуты, Как первая ягода в ждущей ладошке.

Автоматический взлом шифра Цезаря:
Найденный сдвиг: 5

Восстановленный текст:
Оставьте мне лето: лишь в этой купели Мне было свободно, плескалось вольготно... Там Вага и солнце, - никто там не делит Все то, что по праву любви достается: Бескрайность и нежность, доверчивость взлета, Когда нарастаю поля и пределы. Когда так не важно, откуда ты, кто ты - Обычное зернышко, - много ли дела? Но помниться все, но дается наружу Лишь то, что касается счастья и лета. Оставьте мне это единство и дружбу, - Бумажные письма, в которых приветты От всех, кто встречался на тех первопутках, Где так незаметно и все ж непреложно Дарились на счастье века и минуты, Как первая ягода в ждущей ладошке.

Рисунок 10 – Работа программы

Введенный текст для шифрования: Оставьте мне лето: лишь в этой купели Мне было свободно, плескалось вольготно... Там Вага и солнце, - никто там не делит Все то, что по праву любви достается: Бескрайность и нежность, доверчивость взлета, Когда нарастаю поля и пределы. Когда так не важно, откуда ты, кто ты - Обычное зернышко, - много ли дела? Но помниться все, но дается наружу Лишь то, что касается счастья и лета. Оставьте мне это единство и дружбу, - Бумажные письма, в которых приветты От всех, кто встречался на тех первопутках, Где так незаметно и все ж непреложно Дарились на счастье века и минуты, Как первая ягода в ждущей ладошке.

Введенный сдвиг для шифрования: 5

Зашифрованный текст: Уццежбчй стй рйчу: рнэб ж вчуо пшфйрн Стй ёару цжуёуиту, фрийцперуцб журбзучту... Чес Жезе н цуртый, - тнпчу чес тй ийрнч Жцй чу, ьчу фу фхежш ргёжн иуцчейчд: Ёйцпхеотуцб н тйлтучб,

иужйхънжуцб жмрйче, Пузие техецфезпш фурд н фхййра. Пузие чеп тй желту, учпшие ча, пчу ча - Уёаьтуй мйхтаэпу,- стузу рн ийре? Ту фустнчбцд жцй, ту иейчцд техшлш Рнэб чу, ьчу пецейчцд цъецбд н рйче. Уцчежбчй стй вчу йинтцжзу н ихшлэш,- Ёшселтай фнцбсе, ж пучухаь фхнжйча Уч жцйь, пчу жцчхйьерцд те чйь фйхжуфшчпёь, Зий чеп тймесйчту н жцй л тйфхйрулту Иехнрнцб те цъецббй жйпе н снтшча, Пеп фйхжед дзуие ж лишнойо реиуэпй.

Принцип шифрования:

Каждая буква в русском алфавите была сдвинута на 5 позиций вперед.

Например:

- $O \rightarrow Y$
- $C \rightarrow Ч$
- $M \rightarrow С$
- $H \rightarrow Т$

Пробелы, знаки препинания, числа и символы — остались без изменений.

Расшифрованный текст (известный сдвиг): Оставьте мне лето: лишь в этой купели Мне было свободно, плескалось вольготно... Там Вага и солнце, - никто там не делит Все то, что по праву любви достается: Бескрайность и нежность, доверчивость взлета, Когда нараспашку поля и пределы. Когда так не важно, откуда ты, кто ты - Обычное зернышко,- много ли дела? Но помниться все, но дается наружу Лишь то, что касается счастья и лета. Оставьте мне это единство и дружбу,- Бумажные письма, в которых приветы От всех, кто встречался на тех первопутках, Где так незаметно и все ж непреложно Дарились на счастье века и минуты, Как первая ягода в ждущей ладошке.

Автоматический взлом шифра Цезаря:

Найденный сдвиг: 5

Восстановленный текст:

Оставьте мне лето: лишь в этой купели Мне было свободно, плескалось вольготно... Там Вага и солнце, - никто там не делит Все то, что по праву любви достается: Бескрайность и нежность, доверчивость взлета, Когда нараспашку поля и пределы. Когда так не важно, откуда ты, кто ты - Обычное зернышко,- много ли дела? Но помниться все, но дается наружу Лишь то, что касается счастья и лета. Оставьте мне это единство и дружбу,- Бумажные письма, в которых приветы От всех, кто встречался на тех первопутках, Где так незаметно и все ж непреложно Дарились на счастье века и минуты, Как первая ягода в ждущей ладошке.

10. Частотный анализ зашифрованного текста (рис.10).

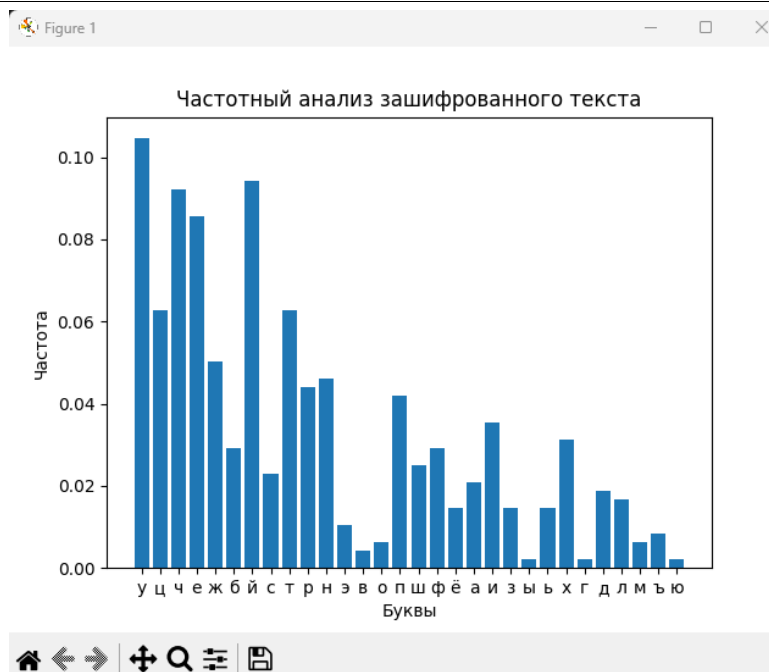


Рисунок 10. График

4 Выводы

В работе реализован и проанализирован метод автоматического взлома шифра Цезаря для русского языка на основе частотного анализа. Результаты подтверждают эффективность классических статистических методов криптоанализа на простых шифрах. Данный подход может быть использован как учебный пример и основа для изучения более сложных криптографических алгоритмов и методов их взлома.

Библиографический список

1. Оленев А.А., Малиатаки В.В., Тынчеров К.Т., Селиванова М.В. Использование системы компьютерной алгебры maple для криптоанализа шифра цезаря // Актуальные вопросы высшего образования – 2018. 2018. С. 203-210.
2. Карпов Д.В., Абузярова Э.Р., Ласкова С.Г. Реализация и анализ быстродействия алгоритма шифрования и дешифрования шифра Цезаря // Евразийский Союз Ученых. 2015. С. 1-2.
3. Протько М.А. Шифр Цезаря и генетический алгоритм // Компьютерные системы и сети : сборник статей 58-й научной конференции аспирантов, магистрантов и студентов. 2022. С. 69-77.
4. Лепшокова А.Р. Разработка интерактивного приложения для наглядного представления шифра цезаря // Актуальные проблемы методики обучения информатике и математике в современной школе. 2020. С. 493-498.
5. Гумерова Л.З., Аглямзянова Г.Н., Майсеева Е.С. Взлом шифра цезаря методом «грубой силы» // Лучшие практики общего и дополнительного образования по естественно-научным и техническим дисциплинам. 2022. С. 157-163.