Шифрование гаммированием

Теория, реализация и анализ

Студент: Ван Яо
Группа: НФИмд-01-25
РУДН, 2025

Содержание

1. Введение в гаммирование

• **Определение**: Наложение гаммы (псевдослучайной последовательности) на открытый текст

• **Историческое развитие**: - Схема однократного использования (one-time pad) - Гаммирование с конечной гаммой - Гаммирование с генератором ПСП

• **Основное преимущество**: Высокая скорость и простота реализации

• **Ключевой принцип**:

$$c_i = p_i \oplus \gamma_i$$

2. Математические основы

Основные операции: - Шифрование: $c_i = (p_i + \gamma_i) \bmod N$ - Дешифрование: $p_i = (c_i - \gamma_i) \bmod N$

Алфавитное кодирование: а = 1, б = 2, в = 3, …, я = 32, пробел = 0 Модуль N = 33

3. Реализация конечной гаммы

```
def encrypt(self, plaintext, gamma):

    plain_numbers = self.text_to_numbers(plaintext)
    gamma_numbers = self.text_to_numbers(gamma)

    if len(gamma_numbers) < len(plain_numbers):

        repeated_gamma = []
        while len(repeated_gamma) < len(plain_numbers):
            repeated_gamma.extend(gamma_numbers)
        gamma_numbers = repeated_gamma[:len(plain_numbers)]
    else:
        gamma_numbers = gamma_numbers[:len(plain_numbers)]

    cipher_numbers = []
    for p, g in zip(plain_numbers, gamma_numbers):
        cipher_num = (p + g) % self.modulus
        cipher_numbers.append(cipher_num)

    return self.numbers_to_text(cipher_numbers)
```

4. Линейный конгруэнтный генератор

Математическая модель: $\gamma_i = (a \cdot \gamma_{i-1} + b) \bmod m$

Реализация LCGGenerator:

```
class LCGGenerator:

    def __init__(self, a, seed, b, m):

        self.a = a
        self.current = seed
        self.b = b
        self.m = m

    def next(self):

        self.current = (self.a * self.current + self.b) % self.m
        return self.current

    def generate_sequence(self, length):

        sequence = [self.current]
        for _ in range(length - 1):
            sequence.append(self.next())
        return sequence
```

5. Сравнительный анализ методов

Параметр	Конечная гамма	LCG гамма
Безопасность	Низкая	Средняя
Простота	Высокая	Средняя
Периодичность	Короткая	Длинная
Реализация	Простая	Сложная
Ключевое пространство	Ограниченное	Большое

6. Заключение и выводы

Теоретическая ценность: Понимание принципов потокового шифрования

• Практическая значимость: Основа для изучения современных потоковых шифров

• Образовательные результаты:

Реализация двух подходов к гаммированию

Сравнительный анализ методов

Понимание ограничений и уязвимостей

• Вывод: Гаммирование остается важным классом криптографических алгоритмов, сочетающим простоту реализации с потенциально высокой стойкостью при правильном использовании.

Спасибо за внимание!