Лабораторная работа №3

Шифрование методом гаммирования

хамза хуссен

---

1. Цель исследования

Изучить и практически освоить алгоритм шифрования данных с использованием гаммирования (потокового шифрования), включая его теоретические основы, реализацию и анализ криптостойкости.

---

2. Основные теоретические положения

2.1. Принцип гаммирования

Гаммирование — это метод криптографического преобразования, при котором исходные данные объединяются с псевдослучайной последовательностью (гаммой) с помощью обратимой операции (чаще всего — побитового сложения по модулю 2). Процесс включает:

- **Генерацию гаммы** с использованием ключа и, возможно, дополнительных параметров.
- **Наложение гаммы** на открытый текст для получения шифротекста (при зашифровании) или на шифротекст для восстановления открытого текста (при расшифровании).

2.2. Особенности метода

- **Криптостойкость** напрямую зависит от свойств гаммы:
    - Длина периода гаммы должна превышать длину шифруемого сообщения.
    - Последовательность должна быть статистически непредсказуемой.
- Если гамма истинно случайна, не повторяется и неизвестна противнику, метод обеспечивает теоретическую нераскрываемость (шифр Вернама).
- На практике используют **детерминированные генераторы псевдослучайных чисел (ГПСЧ)**, инициализируемые ключом.

2.3. Уязвимости и модификации

- Если известен фрагмент открытого текста и соответствующего шифротекста, гамма может быть частично восстановлена.

- Для повышения стойкости применяют **гаммирование с обратной связью**, когда последующие участки гаммы зависят от уже зашифрованных данных (например, через контрольную сумму).
- Такой подход усложняет криптоанализ, но требует корректной синхронизации при расшифровании.

---

3. Порядок выполнения работы

(Здесь следует описать конкретные шаги, выполненные в лабораторной работе: реализация генератора гаммы, функции шифрования/расшифрования, тестирование на различных данных, анализ результатов.)

---

4. Результаты и выводы

(В этом разделе представляются полученные результаты, обсуждаются достоинства и недостатки метода, его область применения и возможные улучшения.)

Примечание: Данный отчёт представлен в иной структуре и формулировках при сохранении исходного смысла и содержания.

3.Выполнение работы

3.1Реализация шифратора и дешифратора Python

```
dict_r = {"а":1,"б":2,"в":3,"г":4,"д":5,"е":6,"ё":7,"ж":8,"з":9,"и":10,
          "й":11,"к":12,"л":13,"м":14,"н":15,"о":16,"п":17,"р":18,"с":19,
          "т":20,"у":21,"ф":22,"х":23,"ц":24,"ч":25,"ш":26,"щ":27,"ъ":28,
          "ы":29,"ь":30,"э":31,"ю":32,"я":33}
inv_dict = {v:k for k,v in dict_r.items()}

def gamma_cipher(text, gamma):
    t_digits = [dict_r[ch] for ch in text]
    g_digits = [dict_r[ch] for ch in gamma]

    encrypted = []
    g_idx = 0
    for val in t_digits:
        total = val + g_digits[g_idx]
        if total > 33:
            total %= 33
        encrypted.append(total)
        g_idx = (g_idx + 1) % len(g_digits)

    enc_text = ''.join(inv_dict[num] for num in encrypted)
```

```
    decrypted = []
    g_idx = 0
    for val in [dict_r[ch] for ch in enc_text]:
        diff = val - g_digits[g_idx]
        if diff < 1:
            diff += 33
        decrypted.append(diff)
        g_idx = (g_idx + 1) % len(g_digits)

    dec_text = ''.join(inv_dict[num] for num in decrypted)

    print("Числа текста:", t_digits)
    print("Числа гаммы:", g_digits)
    print("Числа шифра:", encrypted)
    print("Зашифрованный текст:", enc_text)
    print("Расшифрованный текст:", dec_text)

gamma_cipher("примертекста", "гаммаключ")
```

3.2 Контрольный пример

```
Числа текста: [17, 18, 10, 14, 6, 18, 20, 6, 12, 19, 20, 1]
Числа гаммы: [4, 1, 14, 14, 1, 12, 13, 32, 25]
Числа шифра: [21, 19, 24, 28, 7, 30, 33, 5, 4, 23, 21, 15]
Зашифрованный текст: усцъёьядгхун
Расшифрованный текст: примертекста
```

4 Выводы

Изучили алгоритмы шифрования на основе гаммирования

Список литературы

1. Шифрование методом гаммирования

2. Режим гаммирования в блочном алгоритме шифрования