Лабораторная работа №7

Элементы криптографии. Однократное гаммирование

Латыпова Диана

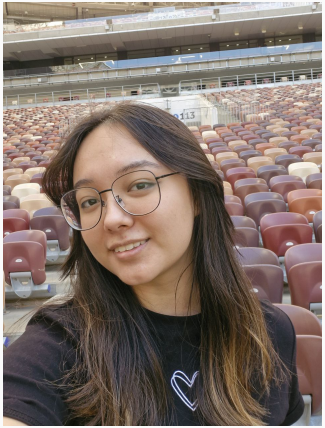
19 октября 2024

Российский университет дружбы народов, Москва, Россия

Информация

Докладчик

- Латыпова Диана
- студент группы НФИбд-02-21
- Российский университет дружбы народов
- 1032215005@rudn.ru
- https://github.com/dlatypova



Вводная часть

Цели и задачи

- Освоить на практике применение режима однократного гаммирования

Теоретическое ввведение

Шифрование/Дешифрование

- **Шифрование** — это процесс преобразования информации (открытого текста) в форму, недоступную для несанкционированного доступа, называемую шифротекстом.
- **Дешифрование** является обратным процессом, при котором шифротекст преобразуется обратно в открытый текст.

Однократное гаммирование

- Основано на применении ключа, длина которого совпадает с длиной исходного текста.
- Символ, подвергнутый XOR с ключом, превращается в шифротекст
- Шифротекст, подвергнутый той же операции XOR с тем же ключом, восстанавливает исходный текст

Восстановление исходного текста: $T = C \oplus K$.

Преимущества и недостатки однократного гаммирования

Преимущества:

- Абсолютная криптографическая стойкость
- Простота реализации

Недостатки:

- Длина ключа
- Одноразовость ключа
- Генерация случайного ключа
- Управление ключами

Операция XOR

- Представляет собой побитовую операцию над двумя строками символов
- Каждый бит открытого текста складывается с соответствующим битом ключа
- Результат операции — это новый набор битов, представляющий шифротекст.

Алгоритм однократного гаммирования:

1. Генерация ключа
2. Шифрование
3. Дешифрование

Выполнение лабораторной работы

Код с результатами

```
[2] import random
    import string
    from random import seed

[7] # Функция сложения двух строк по модулю XOR
    def xor_operation(plain_text, key):
      if len(key) != len(plain_text):
        return "Ошибка. Ключ и текст имеют разную длину!"

      encrypted_text = ''

      for i in range(len(key)):
        xor_symbol = ord(plain_text[i]) ^ ord(key[i])
        encrypted_text += chr(xor_symbol)
      return encrypted_text

[8] plain_text = "«С Новым Годом, друзья!"

    key = ''
    seed(22)

    for i in range(len(plain_text)):
      key += random.choice(string.ascii_letters + string.digits)

    print(f"Сгенерированный ключ: {key}")

    Сгенерированный ключ: 96ipbNClShVP4wY4for9duM

[9] encrypted_text = xor_operation(plain_text, key)
    print(f"Шифротекст: {encrypted_text}")

    Шифротекст: ␦3IҗќȖJèsѻѬЋыu␦ђЯбЎШkl

[11] decrypted_text = xor_operation(encrypted_text, key)
     print(f"Расшифрованный текст: {decrypted_text}")

    Расшифрованный текст: «С Новым Годом, друзья!

    recovered_key = xor_operation(plain_text, encrypted_text)
    print(f"Востановленный ключ: {recovered_key}")

    Востановленный ключ: 96ipbNClShVP4wY4for9duM
```

Рис. 1: Выполнение кода с выводом результатов

Выводы

Выводы

- Освоено на практике применение режима однократного гаммирования