Лабораторная работа №8

Информационная безопасность

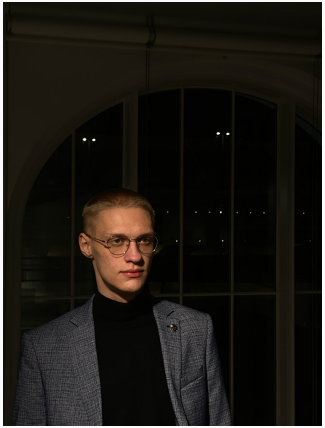
Махорин И. С.

2024

Российский университет дружбы народов имени Патриса Лумумбы, Москва, Россия

Докладчик

- Махорин Иван Сергеевич
- Студент группы НПИбд-02-21
- Студ. билет 1032211221
- Российский университет дружбы народов имени Патриса Лумумбы



Цель лабораторной работы

- Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

Ход выполнения лабораторной работы

Задача лабораторной работы

Два текста кодируются одним ключом (однократное гаммирование). Требуется не зная ключа и не стремясь его определить, прочитать оба текста. Необходимо разработать приложение, позволяющее шифровать и дешифровать тексты P1 и P2 в режиме однократного гаммирования. Приложение должно определить вид шифротекстов C1 и C2 обоих текстов P1 и P2 при известном ключе; Необходимо определить и выразить аналитически способ, при котором злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить [1].

Решение задачи лабораторной работы

Для решения задачи написан программный код:

```
import os  # Импортируем модуль os для генерации случайных байтов


def generate_key(length):
    # Функция для генерации ключа заданной длины
    return os.urandom(length)  # Возвращает случайный ключ в виде байтов


def encrypt(plaintext, key):
    # Функция для шифрования текста с использованием ключа
    return bytes(a ^ b for a, b in zip(plaintext.encode(), key))
    # XOR (исключающее ИЛИ) каждого байта текста с соответствующим байтом ключа


def decrypt(ciphertext, key):
    # Функция для дешифрования текста с использованием ключа
    return bytes(a ^ b for a, b in zip(ciphertext, key)).decode()
    # XOR шифротекста с ключом и декодирование результата в строку


# Примеры использования
P1 = "Hello, World!"  # Первый текст для шифрования
P2 = "Python Programming"  # Второй текст для шифрования

# Генерация ключа
key_length = max(len(P1), len(P2))  # Определяем длину ключа как максимальную длину из двух текстов
key = generate_key(key_length)  # Генерируем ключ заданной длины

C1 = encrypt(P1, key)  # Шифруем первый текст
C2 = encrypt(P2, key)  # Шифруем второй текст

print("Шифротекст C1:", C1)  # Выводим шифротекст первого текста
print("Шифротекст C2:", C2)  # Выводим шифротекст второго текста

# Дешифровка
decrypted_P1 = decrypt(C1, key)  # Дешифруем первый шифротекст
decrypted_P2 = decrypt(C2, key)  # Дешифруем второй шифротекст

print("Дешифрованный текст P1:", decrypted_P1)  # Выводим расшифрованный первый текст
print("Дешифрованный текст P2:", decrypted_P2)  # Выводим расшифрованный второй текст
```

Рис. 1: Программный код

Вывод

Вывод

- В ходе выполнения лабораторной работы было освоено на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

Список литературы. Библиография

Список литературы. Библиография

[1] Методические материалы курса