Математические основы защиты информации и информационной безопасности. Отчет по лабораторной работе №3

Шифрование гаммированием

Терентьев Егор Дмитриевич 1132236902

Содержание

List of Figures

List of Tables

1 Цель работы

Освоить на практике шифрование гаммированием.

2 Выполнение лабораторной работы

Требуется реализовать:

1. Шифрование гаммированием

2.1 Шифрование гаммированием

Шифрование гаммированием - это метод симметричного шифрования, при котором каждый символ или байт исходного сообщения комбинируется с соответствующим символом или байтом ключа (гаммы) с помощью определенной операции, чаще всего XOR.

Основные шаги:

1. Выбор гаммы (ключа): Гамма — это последовательность, которая комбинируется с исходным сообщением. Гамма может быть случайной или генерироваться на основе ключа.

2. Применение гаммы к сообщению: Гамма “наложится” на исходное сообщение. Если гамма короче сообщения, она циклически повторяется.

3. Комбинирование гаммы и сообщения: Наиболее популярная операция для этого — XOR. Если мы говорим о символьном шифровании, то комбинирование может включать в себя сложение (или вычитание для дешифрования) позиций символов в алфавите.

4. Дешифрование: Чтобы дешифровать сообщение, мы применяем ту же операцию комбинирования к зашифрованному сообщению и той же гамме. Если использовалась операция XOR, то повторное применение XOR с той же гаммой вернёт исходное сообщение.

Чтобы реализовать программу был написал след. код на python:

1. Функции получения пар значений ключа и сообщения
2. Функция шифрования, которая берет пары значений и складывает их место в алфавите получая нужную букву шифрования fig. 2.1.

Lab3_Gumming_cipher > gamma_ecnryption.py > ...

```python
def generate_gamma(gamma, message):
    for i in range(len(message)):
        yield gamma[i % len(gamma)], message[i]

def encrypt(gamma, message):
    encrypted_message = ""
    for g, m in generate_gamma(gamma, message):
        encrypted_message += alph[(alph.index(m) + alph.index(g) + 1) % len(alph)]
    return encrypted_message

def decrypt(gamma, encrypted_message):
    decrypted_message = ""
    for g, m in generate_gamma(gamma, encrypted_message):
        decrypted_message += alph[(alph.index(m) - alph.index(g) - 1) % len(alph)]
    return decrypted_message

alph = "абвгдежзийклмнопрстуфхцчшщъыьэюя"
gamma = "гамма"
message = "приказ"

encrypted_message = encrypt(gamma, message)
print(f"Encrypted message: {encrypted_message}")

decrypted_message = decrypt(gamma, encrypted_message)
print(f"Decrypted message: {decrypted_message}")
```

Figure 2.1: main_func

Выходные значения программы (пример из методички) fig. 2.2.

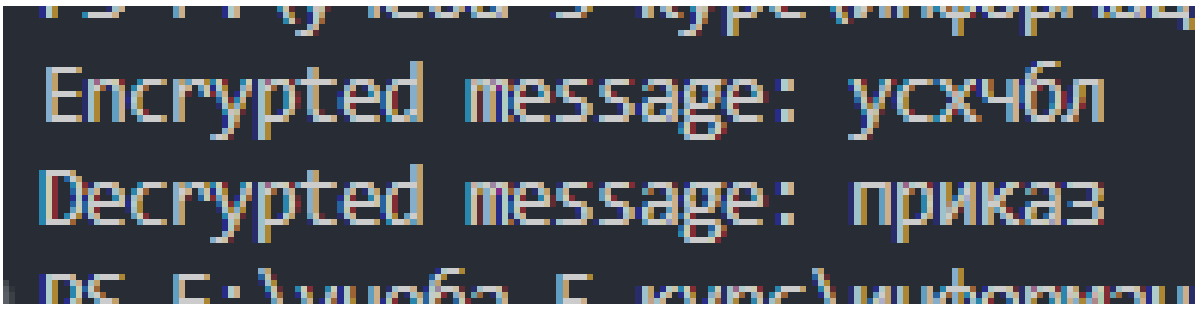

Figure 2.2: output

3 Выводы

В результате выполнения работы я освоил на практике применение шифрование гаммированием.

4 Список литературы

1. Методические материалы курса