# Отчёт по лабораторной работе №8

**дисциплина: Информационная безопасность**

Быстров Глеб Андреевич

# Содержание

# Список иллюстраций

# Список таблиц

# 1 Цель работы

В данной лабораторной работе мне будет необходимо освоить на практике применение режима однократного гаммированияи.

# 2 Задание

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.

# 3 Теоретическое введение

Гамми́рование, или Шифр XOR, — метод симметричного шифрования, заключающийся в «наложении» последовательности, состоящей из случайных чисел, на открытый текст. Последовательность случайных чисел называется гамма-последовательностью и используется для зашифровывания и расшифровывания данных. Суммирование обычно выполняется в каком-либо конечном поле. Например, в поле Галуа GF(2) суммирование принимает вид операции «исключающее ИЛИ (XOR)». [1].

Симметричное шифрование - это метод шифрования, при котором для защиты информации используется ключ, зная который любой может расшифровать или зашифровать данные.

Алгоритмы с симметричными ключами имеют очень высокую производительность. Криптография с симметричными ключами стойкая, что делает практически невозможным процесс дешифрования без знания ключа. При прочих равных условиях стойкость определяется длиной ключа. Так как для шифрования и дешифрования используется один и тот же ключ, при использовании таких алгоритмов требуются высоко надежные механизмы для распределения ключей. Ещё одна проблемой является безопасное распространение симметричных ключей. Алгоритмы симметричного шифрования используют ключи не очень большой длины и могут быстро шифровать большие объемы данных.

Гаммированием (gamma xoring) называется процесс «наложения» гамма-последовательности на открытые данные. Обычно это суммирование по какому-либо модулю, например, по модулю два, такое суммирование принимает

вид обычного «исключающего ИЛИ» суммирования.

Симметричное шифрование остаётся самым актуальным и криптографически гарантированными методом защиты информации. В симметричном шифровании, основанном на использовании составных ключей, идея состоит в том, что секретный ключ делится на две части, хранящиеся отдельно. Каждая часть сама по себе не позволяет выполнить дешифрование [2].

# 4 Выполнение лабораторной работы

1. Реализовал на языке Python программу для выполнения задания (рис. 4.2).

```python
import random
from random import seed
import string

def func(text, key):
    if len(key) != len(text):
        return "Разная длина"
    ctext = ''
    for i in range(len(key)):
        ctext_s = ord(text[i]) ^ ord(key[i])
        ctext += chr(ctext_s)
    return ctext

text1 = "С Новым Годом, друзья!"
text2 = "С днем рождения тебя!!"

key = ''
seed(23)
for i in range (len(text1)):
    key += random.choice(string.ascii_letters + string.digits)

ctext1 = func(text1, key)
ctext2 = func(text2, key)

print('Зашифрованный текст 1:', ctext1)
print('Зашифрованный текст 2:', ctext2)

print('Открытый текст 1:', func(ctext1, key))
print('Открытый текст 2:', func(ctext2, key))

ctextXOR = func(ctext1, ctext2)
print('Текст 1 XOR Текст 2:', ctextXOR)
```

Рис. 4.1: Код программы

```python
text1part1 = text1[3:6]
print('Часть открытого текста 1:', text1part1)

text2part1 = func(ctext1[3:6], ctext2[3:6])
print('Часть открытого текста 2:', func(text2part1, text1part1))
```

```
Зашифрованный текст 1: ЖхХэЇОъВцьЎчV[IwЭбVЭРо
Зашифрованный текст 2: ЖхЌюЕЙFТѲтЎьvяЦцЯчѲЮNо
Открытый текст 1: С Новым Годом, друзья!
Открытый текст 2: С днем рождения тебя!!
Текст 1 XOR Текст 2: )ЮЕмЧЮЮЮДдДЮvЮЮЗ
Часть открытого текста 1: овы
Часть открытого текста 2: нем
```

Рис. 4.2: Код программы

Этот код на Python реализует простой алгоритм шифрования и дешифрования текста с использованием операции XOR (исключающее ИЛИ) между символами текста и ключом.

Генерация ключа:

- Программа создает случайную последовательность символов, состоящую из букв латинского алфавита (верхнего и нижнего регистра) и цифр.
- Длина этой последовательности соответствует длине исходных текстов text1 и text2.
- Этот ключ используется для шифрования и дешифрования текста.

Шифрование текста:

- Для каждого символа в исходном тексте (text1 и text2) и соответствующего символа в ключе выполняется операция XOR (^), в результате чего получается новый символ.
- Полученные символы объединяются в строку, которая представляет зашифрованный текст (ctext1 и ctext2).

Дешифрование текста:

- Та же операция XOR выполняется между зашифрованным текстом (ctext1 и ctext2) и ключом для восстановления исходного текста.

Вывод результата:

- Программа выводит зашифрованный текст ctext1 и ctext2, а затем восстанавливает исходные тексты и выводит их (Открытый текст 1 и Открытый текст 2).
- Также программа выполняет операцию XOR между зашифрованными текстами ctext1 и ctext2 и выводит результат (Текст 1 XOR Текст 2).
- Наконец, программа берет часть исходного текста text1 и выводит ее (Часть открытого текста 1) и также дешифрует и выводит соответствующую часть из text2 (Часть открытого текста 2).

# 5 Контрольные вопросы

1. Зная один из текстов (P1 или P2) и не зная ключа, можно вычислить другой текст, применяя операцию XOR к известному тексту и зашифрованному тексту с использованием того же самого ключа. Это происходит в части кода, где выполняется дешифрование: `func(ctext1, key)` и `func(ctext2, key)`.

2. При повторном использовании одного и того же ключа для шифрования текстовых данных возникают проблемы с безопасностью. При XOR-шифровании, если один и тот же ключ используется для нескольких текстов, и если атакующий получит доступ к двум зашифрованным текстам, это может привести к раскрытию обоих исходных текстов с использованием операции XOR между ними.

3. Режим шифрования однократного гаммирования (One-Time Pad, OTP) реализуется с использованием одного ключа для шифрования двух открытых текстов. В этом режиме каждый из двух открытых текстов XOR-шифруется с тем же самым ключом.

4. Недостатки шифрования одним ключом двух открытых текстов (режим шифрования однократного гаммирования) включают в себя:

- Необходимость использования ключей, длина которых равна длине самих текстов, что делает ключи очень длинными и труднопередаваемыми.

- Очень важно, чтобы ключи были случайными и использовались только

один раз для каждой пары открытых текстов. В противном случае шифр становится уязвимым.

- Отсутствие надежного способа передачи ключей. Если ключ будет скомпрометирован, все сообщения, зашифрованные с его использованием, могут быть расшифрованы.

5. Преимущества шифрования одним ключом двух открытых текстов (режим шифрования однократного гаммирования):

- Шифр OTP считается теоретически непреодолимым (по криптографическим стандартам), если ключи случайны и используются только один раз.
- При правильной реализации шифр OTP обеспечивает максимальную стойкость к атакам, включая криптоанализ.
- Шифр OTP не имеет структуры и не предоставляет атакующему никакой информации о содержании сообщения, что делает его надежным средством шифрования.
- В случае использования ключей, генерируемых с использованием источника случайных чисел с высокой энтропией, шифр OTP может предоставить высокую стойкость к взлому.

# 6 Выводы

В данной лабораторной работе мне успешно удалось освоить на практике применение режима однократного гаммированияи.

# Список литературы


1. Гаммирование [Электронный ресурс]. 2023. URL: https://ru.wikipedia.org/wiki/%D0%93%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5.
2. Симметричное шифрование (гаммирование) [Электронный ресурс]. 2023. URL: http://engineering-science.ru/doc/187185.html.