

# **Отчет по лабораторной работе №8**

**Информационная безопасность**

Жиронкин Павел Владимирович НПИбд-01-18

# Содержание

1	Цель работы	4
2	Теоретическое описание	5
3	Выполнение лабораторной работы	7
4	Выводы	9
5	Контрольные вопросы	10

## Список иллюстраций

3.1	Создание алфавита . . . . .	7
3.2	Функция “vzlom” . . . . .	7
3.3	Функция “shifr” . . . . .	8
3.4	Результат” . . . . .	8

# 1 Цель работы

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

## 2 Теоретическое описание

Предложенная Г. С. Вернамом так называемая «схема однократного использования (гаммирования)» является простой, но надёжной схемой шифрования данных. *Гаммирование* представляет собой наложение (снятие) на открытые (зашифрованные) данные последовательности элементов других данных, полученной с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. Иными словами, наложение гаммы — это сложение её элементов с элементами открытого (закрытого) текста по некоторому фиксированному модулю, значение которого представляет собой известную часть алгоритма шифрования. В соответствии с теорией криптоанализа, если в методе шифрования используется однократная вероятностная гамма (однократное гаммирование) той же длины, что и подлежащий сокрытию текст, то текст нельзя раскрыть. Даже при раскрытии части последовательности гаммы нельзя получить информацию о всём скрываемом тексте. Наложение гаммы по сути представляет собой выполнение операции сложения по модулю 2 (XOR) (обозначаемая знаком  $\oplus$ ) между элементами гаммы и элементами подлежащего сокрытию текста. Напомним, как работает операция XOR над битами:  $0 \oplus 0 = 0, 0 \oplus 1 = 1, 1 \oplus 0 = 1, 1 \oplus 1 = 0$ . Такой метод шифрования является симметричным, так как двойное прибавление одной и той же величины по модулю 2 восстанавливает исходное значение, а шифрование и расшифрование выполняется одной и той же программой. Если известны ключ и открытый текст, то задача нахождения шифротекста заключается в применении к каждому

символу открытого текста следующего правила:

$$C_i = P_i \oplus K_i$$

где  $C_i$  —  $i$ -й символ получившегося зашифрованного послания,  $P_i$  —  $i$ -й символ открытого текста,  $K_i$  —  $i$ -й символ ключа,  $i = 1, m$ . Размерности открытого текста и ключа должны совпадать, и полученный шифротекст будет такой же длины. Если известны шифротекст и открытый текст, то задача нахождения ключа решается также, а именно, обе части равенства необходимо сложить по модулю 2 с  $P_i$ :

$$C_i \oplus P_i = P_i \oplus K_i \oplus P_i = K_i, K_i = C_i \oplus P_i.$$

Открытый текст имеет символьный вид, а ключ — шестнадцатеричное представление. Ключ также можно представить в символьном виде, воспользовавшись таблицей ASCII-кодов. К. Шеннон доказал абсолютную стойкость шифра в случае, когда однократно используемый ключ, длиной, равной длине исходного сообщения, является фрагментом истинно случайной двоичной последовательности с равномерным законом распределения. Криптоалгоритм не даёт никакой информации об открытом тексте: при известном зашифрованном сообщении  $C$  все различные ключевые последовательности  $K$  возможны и равновероятны, а значит, возможны и любые сообщения  $P$ . Необходимые и достаточные условия абсолютной стойкости шифра: – полная случайность ключа; – равенство длин ключа и открытого текста; – однократное использование ключа.

### 3 Выполнение лабораторной работы

Два текста кодируются одним ключом (однократное гаммирование). Требуется, не зная ключа и не стремясь его определить, прочитав оба текста.

1. Создаем алфавит из русских букв и цифр. Задаем входные данные из условия лабораторной работы. (рис. 3.1).

```
a = ord("а")
alphabet = [chr(i) for i in range(a, a + 32)]
a = ord("0")
for i in range(a, a+10):
    alphabet.append(chr(i))

a = ord("А")
for i in range(1040, 1072):
    alphabet.append(chr(i))
print(alphabet)
P1 = "НаВашисходящийот1204"
P2 = "ВСеверныйфилиалБанка"

key = "05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 0B B2 70 54"
```

Рис. 3.1: Создание алфавита

2. Функция “vzlom”, которая получив два открытых сообщения и объединив их получает гамму. (рис. 3.2).

```
def vzlom(P1, P2):
    code = []
    for i in range(20):
        code.append(alphabet[(alphabet.index(P1[i]) + alphabet.index(P2[i])) % len(alphabet)])
    print(code)
    print(code[16], " ", code[19])
    p3 = "".join(code)
    print(p3)

vzlom(P1, P2)
```

['а', 'б', 'в', 'г', 'д', 'е', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц',  
'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э', 'я', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'А', 'Б', 'В', 'Г',  
'Д', 'Е', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ',  
'Ы', 'Ь', 'Э', 'Я']  
['а', 'с', '3', 'а', 'э', 'а', 'б', 'ж', 'ч', 'ш', '7', '4', 'р', 'и', 'ц', 'у', '1', 'Е', 'А', '4']  
1 4  
цСЗэшжкч74ршц1ЕА4

Рис. 3.2: Функция “vzlom”

3. Функция “shifr”, которая получает исходные сообщения (рис. 3.3).

Рис. 3.3: Функция “shifr”

4. Алгоритм расшифровки, вывод программы. (рис. 3.4).

Рис. 3.4: Результат”



## 4 Выводы

На основе проделанной работы освоил на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

## 5 Контрольные вопросы

1. Как, зная один из текстов ( $P_1$  или  $P_2$ ), определить другой, не зная при этом ключа?

Можно использовать комбинацию, при которой ключ не будет использоваться, для прочтения неизвестного текста:

$$C_1 \oplus C_2 \oplus P_1 = P_1 \oplus P_2 \oplus P_1 = P_2.$$

$C_1$  и  $C_2$  – известный шифротекст

2. Что будет при повторном использовании ключа при шифровании текста?

Текст расшифруется.

3. Как реализуется режим шифрования однократного гаммирования одним ключом двух открытых текстов?

Используется следующая комбинация:

$$C_1 = P_1 \oplus K_i$$

$$C_2 = P_2 \oplus K_i$$

$P_i$  – открытый текст,  $C_i$  – зашифрованный текст,  $K$  – ключ шифрования.

4. Перечислите недостатки шифрования одним ключом двух открытых текстов.

ключ, попав не в те руки, даст возможность злоумышленнику расшифровать оба текста; можно расшифровать с помощью открытого текста другие известные шифротексты; можно узнать часть текста, используя заранее известный шаблон и формат другого текста.

5. Перечислите преимущества шифрования одним ключом двух открытых текстов.

Преимущества шифрования одним ключом заключаются в том что, скорость шифрования выше; алгоритм шифрования простой; а так же шифротекст сильно меняется, если изменяется ключ или открытый текст.☒