

Отчёта по лабораторной работе № 7

Информационная безопасность

Адебайо Ридвануллахи Айофе

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	9
3.1	Задание	9
3.2	Порядок выполнения работы	9
4	Выводы	11
5	Список литературы	12

Список иллюстраций

2.1	Схема однократного использования Вернама	7
3.1	создание ключа той же длины, что и открытый текст	10
3.2	получение шифротекста	10

Список таблиц

1 Цель работы

Освоить на практике применение режима однократного гаммирования.

2 Теоретическое введение

Предложенная Г. С. Вернамом так называемая «схема однократного использования (гаммирования)» (рис. 7.1) 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

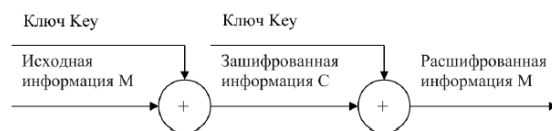


Рис. 2.1: Схема однократного использования Вернама

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 в соответствии с [{#fig:001}](#), 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 фраг-

ментом 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

3.1 Задание

Нужно подобрать ключ, чтобы получить сообщение «С Новым Годом, друзья!». Требуется разработать приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования. Приложение должно:

1. Определить вид шифротекста при известном ключе и известном открытом тексте.
2. Определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

3.2 Порядок выполнения работы

1. Определить вид шифротекста при известном ключе и известном открытом тексте

```

1 text="С Новым Годом, друзья!"

1 key=""
2 seed(23)
3 for i in range(len(text)):
4     key += random.choice(string.ascii_letters + string.digits)
5 print(key)

7X8s51fbLtByHwiUmrCaon

```

Рис. 3.1: создание ключа той же длины, что и открытый текст

2. Определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

```

In [3]: 1 def cipher_text_function(text, key):
2         #condition to check if the key and text have the same length
3         if len(key) != len(text):
4             return "Key and text must have the same length"
5         cipher_text = ""
6         for i in range (len(key)):
7             cipher_text_symbol=ord(text[i]) ^ ord(key[i])
8             cipher_text+=chr(cipher_text_symbol)
9         return cipher_text

In [9]: 1 cipher_text = cipher_text_function(text, key)
2         print(cipher_text)

ЖхХЭЇОњВцѝѸчV[IwЭ6VЭРо

In [11]: 1 print(f"Text: {cipher_text_function(cipher_text, key)}")

Text: С Новым Годом, друзья!

In [13]: 1 print(f"Key: {cipher_text_function(text, cipher_text)}")

Key: 7X8s51fbLtByHwiUmrCaon

```

Рис. 3.2: получение шифротекста

4 Выводы

В ходе выполнения данной лабораторной работы я освоил на практике применение режима однократного гаммирования

5 Список литературы

1. Кулябов Д. С. *Лабораторная работа №7** : 007-lab_crypto-gamma.pdf*