

Отчёт по лабораторной работе 7

Элементы криптографии. Однократное гаммирование

Румянцева Александра Сергеевна

Содержание

Цель работы	5
Задание	6
Теория	7
Выполнение лабораторной работы	8
Контрольные вопросы	11
Библиография	14
Выводы	15

Список иллюстраций

0.1	рис.1. Программа для шифрования и дешифрования. Проверка её работы.	9
0.2	рис.2. Определение шифротекста для «С Новым Годом, друзья!». .	9
0.3	рис.3. Расшифровка текста с помощью иного ключа.	10
0.1	рис.4. Пример шифрования и расшифровки, если длина ключа меньше длины текста.	12
0.2	рис.5. Нахождение ключа, если известны открытый текст и шифротекст.	13

Список таблиц

Цель работы

Освоить на практике применение режима однократного гаммирования.

Задание

Лабораторная работа подразумевает освоение граммирования опытным путем.

Теория

Гаммирование - 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.

Принцип шифрования гаммированием заключается в генерации гаммы шифра с помощью датчика псевдослучайных чисел и наложении полученной гаммы шифра на открытые данные обратимым образом (например, используя операцию сложения по модулю 2). Процесс дешифрования сводится к повторной генерации гаммы шифра при известном ключе и наложении такой же гаммы на зашифрованные данные.

Полученный зашифрованный текст является достаточно трудным для раскрытия в том случае, если гамма шифра не содержит повторяющихся битовых последовательностей и изменяется случайным образом для каждого шифруемого слова. Если период гаммы превышает длину всего зашифрованного текста и неизвестна никакая часть исходного текста, то шифр можно раскрыть только прямым перебором (подбором ключа). В этом случае криптостойкость определяется размером ключа.

Выполнение лабораторной работы

1. Изучила теорию и указание к лабораторной работе.
2. Написала программу, которая подбирает ключ, чтобы получить сообщение «С Новым Годом, друзья!»

Целью написанной программы является разработка приложения, позволяющего 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.

Я написала программу, состоящую из 2ух функций: функция генерации ключа шифрования, и функция гаммирования. Затем я проверила корректность выполняемых действий программой (рис. 1).


```

In [1]: 1 import random
        2 import string

In [3]: 1 def key_generate(length, symbols = string.ascii_letters + string.digits):
        2     return ''.join(random.choice(symbols) for i in range (length))
        3
        4 def gramming(text, key):
        5     new_text = [ord(i) for i in text]
        6     new_key = [ord(i) for i in key]
        7     return ''.join(chr(t^k) for t,k in zip(new_text, new_key))

In [4]: 1 text = 'test'
        2 key = key_generate(4)
        3 key

Out[4]: 'UeEL'

In [5]: 1 gramming(text, key)

Out[5]: '!\\x0068'

In [6]: 1 gramming(gramming(text, key), key)

Out[6]: 'test'

```

Рис. 0.1: рис.1. Программа для шифрования и дешифрования. Проверка её работы.

Как мы видим из рисунка, программа успешно генерирует ключ нужной длины, с в его помощью может шифровать и обратно расшифровать текст.

Выполним пункты задания:

- 1) Определила вид шифротекста при известном ключе и открытом тексте. Текст использовала из задания «С Новым Годом, друзья!» (рис. 2).

```

In [7]: 1 text = 'С Новым Годом, друзья!'
        2 key = key_generate(len(text))
        3 key

Out[7]: '8BLJSH4bDhkOwNpwP8Zu7U'

In [13]: 1 shifr = gramming(text, key)

In [14]: 1 print('Вопрос 1. \nШифротекст при известном ключе (', key, ') и известном открытом тексте (', text, ') имеет вид:', shifr )

Вопрос 1.
Шифротекст при известном ключе ( 8BLJSH4bDhkOwNpwP8Zu7U ) и известном открытом тексте ( С Новым Годом, друзья! ) имеет вид: Ыбё
Vwғ3BіицфybPyAоиййт

```

Рис. 0.2: рис.2. Определение шифротекста для «С Новым Годом, друзья!».

- 2) Определила ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста (рис. 3).

```

In [16]: 1 key2 = key_generate(len(text))

In [18]: 1 text2 = gramming(shifr, key2)

In [23]: 1 print('Вопрос 2. \nМы создали ключ (' , key2 ,
          2 ' '), с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из во
          <
          >
Вопрос 2.
Мы создали ключ ( 7WanJrMUpS6UkcACHICUD1 ), с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста,
представляющий собой один из возможных вариантов прочтения открытого текста.
Тем самым мы получили следующую расшифровку: 05a1F6eK04Xa0P00Ejv00Me
Можно заметить, что полученный текст отличается от исходного текста, так как для шифрования и расшифровки используются разные к
лючи.

```

Рис. 0.3: рис.3. Расшифровка текста с помощью иного ключа.

Как видно на рисунке 3, мы создали ключ key2, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста. Тем самым мы получили расшифровку, которая отличается от исходного текста, так как для шифрования и расшифровки используются разные ключи.

Контрольные вопросы

1. Поясните смысл однократного гаммирования.

Гаммирование – это наложение/снятие на открытые/зашифрованные данные криптографической гаммы, то есть последовательности элементов данных, вырабатываемых с помощью некоторого криптографического алгоритма, для получения зашифрованных/открытых данных.

Однократное гаммирование – это когда каждый символ попарно с символом ключа складываются по модулю 2 (XOR).

2. Перечислите недостатки однократного гаммирования.

Размер ключевого материала должен совпадать с размером передаваемых сообщений (как можно увидеть на рисунке 4, в случае несовпадения размером текст выводится неполный, то есть теряется часть текста, поэтому размеры должны совпадать). Также необходимо иметь эффективные процедуры для выработки случайных равновероятных двоичных последовательностей и специальную службу для развоза огромного количества ключей. Если одну и ту же гамму использовать дважды для разных сообщений, то шифр из совершенно стойкого превращается в «совершенно нестойкий» и допускает дешифрование практически вручную.

```

In [24]: 1 text = 'С Новым Годом, друзья!'
          2 key = key_generate(len(text)-1)
          3 key
Out[24]: 'mZUYVeGFyDDP27CvkVYxx'

In [25]: 1 shifr = gramming(text, key)

In [27]: 1 gramming(shifr, key)
Out[27]: 'С Новым Годом, друзья'

```

Рис. 0.1: рис.4. Пример шифрования и расшифровки, если длина ключа меньше длины текста.

3. Перечислите преимущества однократного гаммирования.

С точки зрения теории криптоанализа метод шифрования случайной однократной равновероятной гаммой той же длины, что и открытый текст, является невскрываемым. Кроме того, даже раскрыв часть сообщения, информация о вскрытом участке гаммы не дает информации об остальных ее частях. К достоинствам также можно отнести простоту реализации и удобство применения.

4. Почему длина открытого текста должна совпадать с длиной ключа?

Потому что каждый символ открытого текста складывается с символом ключа попарно. Иначе можно получить неполный текст, как рассматривали пример на рисунке 4.

5. Какая операция используется в режиме однократного гаммирования, назовите её особенности?

В режиме однократного гаммирования используется сложение по модулю 2 (XOR) между элементами гаммы и элементами подлежащего сокрытию текста. Особенность заключается в том, что этот алгоритм шифрования является симметричным. Поскольку двойное прибавление одной и той же величины по модулю 2 восстанавливает исходное значение, шифрование и расшифрование выполняется одной и той же программой.

6. Как по открытому тексту и ключу получить шифротекст?

Если известны ключ и открытый текст, то задача нахождения шифротекста

заключается в применении сложения по модулю 2 между каждым символом открытого текста и ключа. То есть выполняем однократное гаммирование.

7. Как по открытому тексту и шифротексту получить ключ?

Если известны открытый текст и шифротекст, то задача нахождения ключа заключается в применении сложения по модулю 2 между каждым символом открытого текста и шифра. Пример рассмотрен на рисунке 5:

```
In [33]: 1 text = 'С Новым Годом, друзья!'
          2 key = key_generate(len(text))
          3 key

Out[33]: 'skHSV0dZ9STWIRn49WC92S'

In [34]: 1 shifr = gramming(text, key)

In [36]: 1 gramming(shifr, text)

Out[36]: 'skHSV0dZ9STWIRn49WC92S'
```

Рис. 0.2: рис.5. Нахождение ключа, если известны открытый текст и шифротекст.

8. В чем заключаются необходимые и достаточные условия абсолютной стойкости шифра?

Необходимые и достаточные условия абсолютной стойкости шифра: полная случайность ключа; равенство длин ключа и открытого текста; однократное использование ключа.

Бібліографія

1. ТУИС РУДН
2. Стаття “Принцип шифрования гаммированием” на сайте <http://crypto.pp.ua/2010/04/82/>

Выводы

Я освоила на практике применение режима однократного гаммирования.