

Основы информационной безопасности

Элементы криптографии. Однократное гаммирование

Демидова Екатерина Алексеевна

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретические сведения	6
4	Выполнение лабораторной работы	7
5	Контрольные вопросы	9
6	Выводы	11
	Список литературы	12

Список иллюстраций

4.1	Результаты работы программы	8
-----	---------------------------------------	---

1 Цель работы

Освоить на практике применение режима однократного гаммирования.

2 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 Теоретические сведения

Гаммирование, или Шифр XOR, — метод симметричного шифрования, заключающийся в «наложении» последовательности, состоящей из случайных чисел, на открытый текст[1]:. Последовательность случайных чисел называется гамма-последовательностью и используется для зашифровывания и расшифровывания данных.

4 Выполнение лабораторной работы

Создадим функции: `key_gen` – отвечает за генерацию случайного ключа(составляется выбором из букв кириллицы больших и малых, символов, цифр), `encryption` – принимает на вход текст и ключ, а затем осуществляет посимвольное сложение по модулю 2, `part_key` – подбирает точную часть ключа для известного фрагмента сообщения, а затем оставшуюся часть выбирает случайным образом(используются ранее описанные функции):

```
def key_gen(text):
    alph = [chr(i) for i in range(1040,1104)] + [chr(i) for i in range(33,64)]
    key = "".join([random.choice(alph) for i in range(len(text))])
    return key

def encryption(text, key):
    return "".join([chr(ord(key[i])^ord(text[i])) for i in range(len(key))])

def part_key_gen(fragment, encrypted_text):
    key_start = encryption(fragment, encrypted_text[:len(fragment)])
    return key_start+key_gen(encrypted_text[len(fragment):])
```

Затем применим эти функции к заданному сообщению:

```
text = "С новым годом, друзья!" # сообщение
key = key_gen(s) # ключ
encrypted_text = encryption(s, key) # зашифрованный текст
```

```
print(encrypted_text)
```

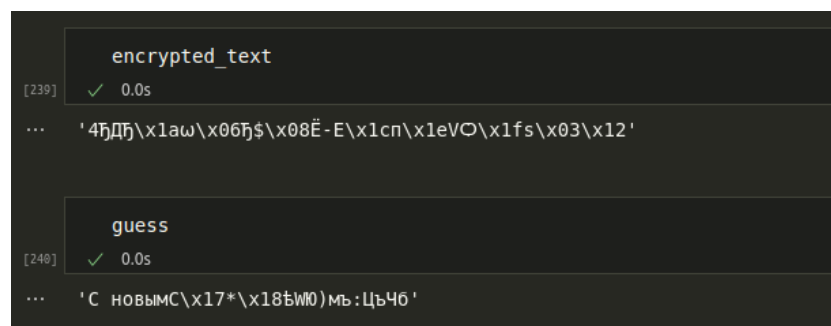
```
fragment = "С новым" # известный фрагмент сообщения
```

```
part_key = part_key_gen(fragment, encrypted_text) # ключ на основе фрагмента сообщ
```

```
guess = encryption(encrypted_text, part_key) # предположительный текст
```

```
print(guess)
```

В результате получим следующий вариант шифрования и один из вариантов прочтения текста(рис. 4.1)



```
encrypted_text
[239] ✓ 0.0s
... '4ЃДђ\x1aω\x06ђ$\x08Ě-E\x1cп\x1eV○\x1fs\x03\x12'

guess
[240] ✓ 0.0s
... 'С новымС\x17*\x18ЪиЮ)мь:ЦъЧб'
```

Рис. 4.1: Результаты работы программы

5 Контрольные вопросы

1. Поясните смысл однократного гаммирования.

Гаммирование, или Шифр XOR, — метод симметричного шифрования, заключающийся в «наложении» последовательности, состоящей из случайных чисел, на открытый текст. Если в методе шифрования используется однократная вероятностная гамма (однократное гаммирование) той же длины, что и подлежащий сокрытию текст, то текст нельзя раскрыть.

2. Перечислите недостатки однократного гаммирования.

- Если один и тот же ключ используется для шифрования нескольких сообщений, это может привести к уязвимостям. Например, если злоумышленник узнает открытый текст и соответствующий шифротекст, он может использовать эту информацию для взлома ключа.
- Однократное гаммирование не обеспечивает аутентификацию или целостность данных. Это означает, что злоумышленник может изменить шифротекст без заметных изменений в открытом тексте.

3. Перечислите преимущества однократного гаммирования.

- Однократное гаммирование обеспечивает высокий уровень конфиденциальности, поскольку шифротекст не может быть легко взломан без знания ключа.
- Однократное гаммирование обеспечивает равномерное распределение вероятностей для каждого символа в шифротексте, что делает его статистически неразличимым от случайной последовательности.

- Однократное гаммирование является простым и быстрым методом шифрования.

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

Если в методе шифрования используется однократная вероятностная гамма (однократное гаммирование) той же длины, что и подлежащий сокрытию текст, то текст нельзя раскрыть.

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

В режиме однократного гаммирования используется операция XOR (исключающее ИЛИ). Операция XOR комбинирует биты открытого текста и ключа, чтобы получить шифротекст. Особенностью операции XOR является то, что она возвращает 1 только в том случае, если один из входных битов равен 1, но не оба.

6. Как по открытому тексту и ключу получить шифротекст?

Нужно побитово сложить по модулю численное представление символов в ключе и в открытом тексте.

7. Как по открытому тексту и шифротексту получить ключ?

Нужно побитово сложить по модулю численное представление символов в шифротексте и в открытом тексте.

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.

6 Выводы

В результате выполнения работы были освоены практические навыки применения режима однократного гаммирования.

Список литературы

1. Яценко В. В. Введение в криптографию. МЦНМО, 2017. 349 с.