Лабораторная работа №7

Элементы криптографии. Однократное гаммирование

Тазаева Анастасия Анатольевна

Содержание

Список иллюстраций

Список таблиц

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 Теоретическое введение

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 ⊕) между элементами гаммы и элементами подлежащего сокрытию текста. Напомним, как работает операция XOR над битами: 0 ⊕ 0 = 0, 0 ⊕ 1 = 1, 1 ⊕ 0 = 1, 1 ⊕ 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[**1?**].

4 Выполнение лабораторной работы

Написала программу, реализующую данный метод (рис. fig. 4.1).

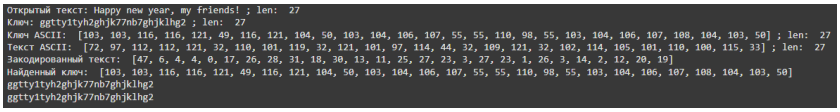

Рис. 4.1: Результат программы

```
import itertools

def create_dict(): #составление словаря
    ascii_dict = dict()
    ascii_numbers = range(0, 256)
    for i in ascii_numbers:
        ascii_dict[i] = chr(i)
    return ascii_dict

def encode_value(text):
    dictionary = create_dict()
    return [k for c in text for k, v in dictionary.items() if v == c]

def decode_value(number):
    dictionary = create_dict()
    dec_value = ''
    for i in number:
```

```
        dec_value = dec_value + dictionary[i]
    return dec_value

def comparator(value, key):
    return dict([(index, list(character))
                 for index, character in enumerate(zip(value, itertools.cycle(key)))])

def full_encode(value, key):
    dictionary = comparator(value, key)
    ln = len(create_dict())
    return [(v[0] ^ v[1]) % ln for v in dictionary.values()]

word = 'Happy new year, my friends!'
key = 'ggtty1tyh2ghjk77nb7ghjklhg2'
print('Открытый текст: ' + word, "; len: ", len(word))
print('Ключ: ' + key, "; len: ", len(key))
key_encoded = encode_value(key)
print('Ключ ASCII: ', key_encoded, "; len: ", len(key_encoded))
value_encoded = encode_value(word)
print('Слово ASCII: ', value_encoded, "; len: ", len(value_encoded))
text_encoded = full_encode(value_encoded, key_encoded)
print('Закодированный текст: ', text_encoded)

encoded_key = full_encode(value_encoded, text_encoded)
print('Найденный ключ: ', encoded_key)
decoded_val = decode_value(encoded_key)
print(decoded_val)
print(key)
```

5 Ответы на контрольные вопросы

1. Что означает однократное гаммирование?

Однократное гаммирование (или потоковое шифрование) - это метод шифрования, при котором каждый символ открытого текста комбинируется с соответствующим символом ключа с использованием операции XOR (исключающее ИЛИ). Это позволяет создать шифротекст, который, в теории, сложно расшифровать без знания правильного ключа.

2. Какие недостатки существуют у однократного гаммирования?

Недостатки однократного гаммирования включают: - Одинаковая длина ключа и открытого текста - необходимость генерировать ключи такой же длины, что и открытый текст, что может быть неудобным. - Отсутствие аутентификации - однократное гаммирование не предоставляет проверку целостности данных и аутентификации отправителя. - Ключевой материал - необходимо генерировать случайные ключи высокого качества для обеспечения безопасности.

3. Какие преимущества имеет однократное гаммирование?

   Преимущества однократного гаммирования включают:

   - Высокая стойкость к криптоанализу в теории при правильном использовании.
   - Простота реализации операции XOR.
   - Отсутствие паттернов в шифротексте, которые могли бы использоваться для атак.

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

   Для однократного гаммирования (потокового шифрования) длина ключа должна совпадать с длиной открытого текста, чтобы выполнить операцию XOR между ними. Если длины открытого текста и ключа не совпадают, операция XOR невозможна или требует дополнительных операций, что может сильно ухудшить безопасность шифра.

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

   В режиме однократного гаммирования используется операция XOR (исключающее ИЛИ). Особенность операции XOR заключается в том, что она возвращает true (1) только в случае, если один из операндов равен true (1), но не оба. Если оба операнда одинаковы (или оба равны 0), результат будет false (0). Это делает XOR удобной для комбинирования символов открытого текста и ключа для создания шифротекста.

6. Как по открытому тексту и ключу получить шифротекст?

   Чтобы получить шифротекст по открытому тексту и ключу в режиме однократного гаммирования, каждый символ открытого текста комбинируется с соответствующим символом ключа с использованием операции XOR. Результат этой операции будет шифротекстом.

7. Как по открытому тексту и шифротексту получить ключ?

   В однократном гаммировании необходимо знать ключ для расшифровки, и обратное получение ключа из открытого текста и шифротекста в общем случае является сложной задачей. Этот процесс не является стандартной частью схемы однократного гаммирования, и расшифровка обычно требует знания правильного ключа.

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

Абсолютная стойкость шифра означает, что шифр не может быть взломан, независимо от доступных вычислительных ресурсов и времени. Однако абсолютная стойкость шифра является идеализированным понятием и в реальности трудно достижима. Некоторые общие условия, которые могут повысить стойкость шифра, включают:

- Использование ключей максимальной длины.
- Использование ключей, генерируемых случайным образом.
- Отсутствие паттернов в шифротексте.
- Соблюдение математических принципов криптографии.

Однако даже при соблюдении всех этих условий абсолютная стойкость может оставаться недостижимой, так как новые методы атаки всегда могут быть разработаны.

6 Выводы

Я на практике освоила применение режима однократного гаммирования.