Основы информационной безопасности

Лабораторная работа № 8. Элементы криптографии. Шифрование (кодирование) различных исходных текстов одним ключом

Нзита Диатезилуа Катенди

Содержание

Список иллюстраций

Цель работы

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

Задание

Два текста кодируются одним ключом (однократное гаммирование). Требуется не зная ключа и не стремясь его определить, прочитать оба текста. Необходимо разработать приложение, позволяющее шифровать и дешифровать тексты P1 и P2 в режиме однократного гаммирования. Приложение должно определить вид шифротекстов C1 и C2 обоих текстов P1 и P2 при известном ключе ; Необходимо определить и выразить аналитически способ, при котором злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить

Теоретические сведения

В результате выполнения работы были освоены практические навыки применения режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом

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:

```
import random

# Функция для генерации случайного ключа
def key_gen(text):
    alph = [chr(i) for i in range(1040, 1104)] + [chr(i) for i in range(3
    key = "".join([random.choice(alph) for i in range(len(text))])
    return key
```

Сначала зашифруем два сообщения:

```
# Функция шифрования (XOR) текста с ключом
def encryption(text, key):
    return "".join([chr(ord(key[i]) ^ ord(text[i])) for i in range(len(ke

# Сообщения для шифрования
P1 = "ВЗападныйФилиалБанка"
P2 = "ВСеверныйФилиалБанка"

# Генерация ключа и шифрование сообщений
```

```
key = key_gen(P1)
C1 = encryption(P1, key)
C2 = encryption(P2, key)
```

Опишем случай, когда злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить. Предположим, что одна из телеграмм является шаблоном – т.е. имеет текст фиксированный формат, в который вписываются значения полей. Допустим, что злоумышленнику этот формат известен. Тогда он получает достаточно много пар $C1 \oplus C2$ (известен вид обеих шифровок). Тогда зная P1 имеем:

$$C1 \oplus C2 \oplus P1 = P1 \oplus P2 \oplus P1 = P2.$$

Давайте применим этот процесс на практике. Давайте применим наши функции к данной базе данных. Предположим, мы знаем часть второй истории. В цикле while он интерактивно принимает подмножество сообщений, прежде чем мы примем во внимание их все:

```
# Известный фрагмент второго сообщения
fragment = "ВСев"

msg2 = fragment
c1, c2 = C1, C2  # Зашифрованные сообщения
length = len(msg2)

# Цикл расшифровки части первого сообщения
while length <= len(P1):
    # XOR зашифрованных сообщений до текущей длины
    C12 = encryption(C1[:length], C2[:length])

    # Расшифровка первого сообщения через XOR с известной частью второго
    msg1 = encryption(C12, msg2)
```

```
# Вывод расшифрованного текста
print("Расшифрованный текст:")
print(msg1 + c1[length:])

# Ввод следующей части текста
print("Введите продолжение текста: ")
msg1 += input()

# Обновление длины расшифрованного текста
length = len(msg1)

# Вывод обновленного текста
print(msg1 + c1[length:])

# Обмен сообщениями для следующей итерации
msg1, msg2 = msg2, msg1
c1, c2 = c2, c1
```

```
Расшифрованный текст:
ВЗап□M'iЄhJ□(()xЋЪ
Введите продолжение текста:
ВЗап□M'iЄhJ□(()xЋЪ
ВЗапВЗап□M'iЄhJ□(()xЋЪ
```

Рис. 1: Результаты работы программы

Контрольные вопросы

1. Как, зная один из текстов (P1 или P2), определить другой, не зная при этом ключа?

Предположим, что одна из телеграмм является шаблоном – т.е. имеет текст фиксированный формат, в который вписываются значения полей. Допустим, что злоумышленнику этот формат известен. Тогда он получает достаточно много пар $C1 \oplus C2$ (известен вид обеих шифровок). Тогда зная P1 имеем:

$$C1 \oplus C2 \oplus P1 = P1 \oplus P2 \oplus P1 = P2.$$

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

Текст вернется к исходному виду.

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

К обоим текстам применяется один и тот же ключ.

4. Перечислите недостатки шифрования одним ключом двух открытых текстов.

Главным недостатком является повышение уязвимости. Если злоумышленник узнает один из исзодных текстов или даже его часть, то он может узнать и второй текст.

5. Перечислите преимущества шифрования одним ключом двух открытых текстов.

Ключи могут занимать большое количество памяти и долго генерироваться, поэтому использование одного ключа оптимизирует шифрование. Также это упрощает дешифровку.

Выводы

В результате выполнения работы были освоены практические навыки применения режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом

Список литературы