

Лабораторная №8

Дисциплина: Основы информационной безопасности

Жибицкая Евгения Дмитриевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Листинг и вывод	9
5	Ответы на контрольные вопросы	11
6	Выводы	13
	Список литературы	14

Список иллюстраций

3.1	Сообщения 1 и 2	7
3.2	Расшифровка сообщений	8
3.3	Вывод программы	8

Список таблиц

1 Цель работы

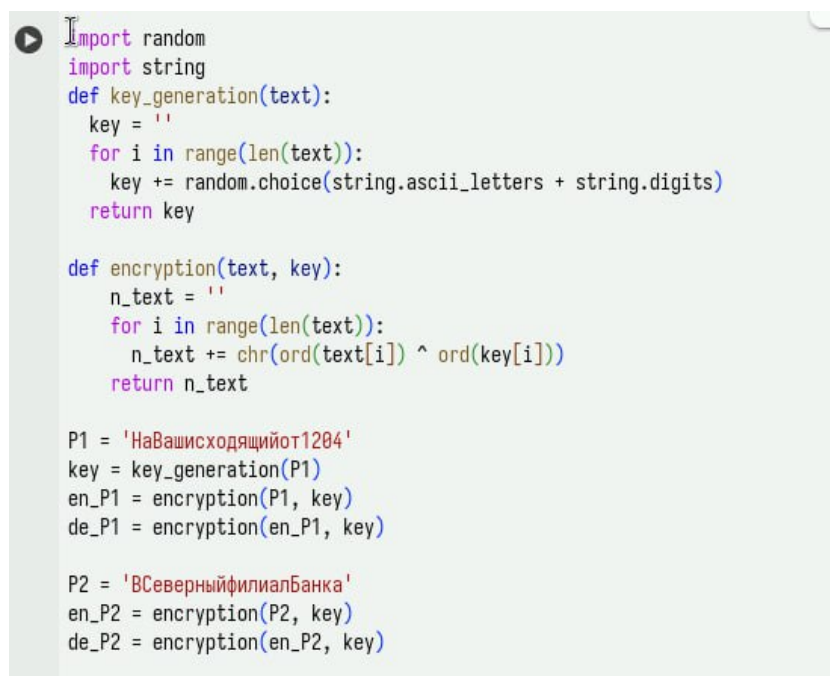
Освоение на практике применения режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

2 Задание

Два текста кодируются одним ключом (однократное гаммирование). Требуется не зная ключа и не стремясь его определить, прочесть оба текста. Необходимо разработать приложение, позволяющее шифровать и дешифровать тексты P_1 и P_2 в режиме однократного гаммирования. Приложение должно определить вид шифротекстов C_1 и C_2 обоих текстов P_1 и P_2 при известном ключе ; Необходимо определить и выразить аналитически способ, при котором злоумышленник может прочесть оба текста, не зная ключа и не стремясь его определить.

3 Выполнение лабораторной работы

Для выполнения данной лабораторной работы воспользуемся программой, написанной в предыдущей лабораторной работе №7. Зашифруем исходные сообщения, предварительно сгенерировав ключ(рис. 3.1).



```
import random
import string
def key_generation(text):
    key = ''
    for i in range(len(text)):
        key += random.choice(string.ascii_letters + string.digits)
    return key

def encryption(text, key):
    n_text = ''
    for i in range(len(text)):
        n_text += chr(ord(text[i]) ^ ord(key[i]))
    return n_text

P1 = 'НаВашисходящийот1204'
key = key_generation(P1)
en_P1 = encryption(P1, key)
de_P1 = encryption(en_P1, key)

P2 = 'ВСеверныйфилиалБанка'
en_P2 = encryption(P2, key)
de_P2 = encryption(en_P2, key)
```

Рис. 3.1: Сообщения 1 и 2

Затем сгенерируем шифротекст, посмотрим на вывод программы. Также выполним задание и расшифруем сообщение 1 за счет сообщения 2 и наоборот(для этого также используем сложение по модулю 2)(рис. 3.2).

4 ЛИСТИНГ И ВЫВОД

```
import random
import string
def key_generation(text):
    key = ''
    for i in range(len(text)):
        key += random.choice(string.ascii_letters + string.digits)
    return key

def encryption(text, key):
    n_text = ''
    for i in range(len(text)):
        n_text += chr(ord(text[i]) ^ ord(key[i]))
    return n_text

P1 = 'НаВашисходящийот1204'
key = key_generation(P1)
en_P1 = encryption(P1, key)
de_P1 = encryption(en_P1, key)

P2 = 'ВСеверныйфилиалБанка'
key = key_generation(P2)
en_P2 = encryption(P2, key)
```

```
de_P2 = encryption(en_P2, key)
```

```
print("Открытый текст: ", P1, "\nКлюч: ", key, "\nШифортекст: ", en_P1, "\nИсходный те
```

```
print("Открытый текст: ", P2, "\nКлюч: ", key, "\nШифортекст: ", en_P2, "\nИсходный те
```

```
encr = encryption(de_P1, de_P2)
```

```
print("Расшифровка P1, зная P2: ", encryption(P2, encr))
```

```
print("Расшифровка P2, зная P1: ", encryption(P1, encr))
```

Вывод:

Открытый текст: НаВашисходящийот1204

Ключ: PfkLQeCIBiSrJuUFTQcd

Шифортекст: эўёЙіХ□□льШб□ўълхкХ_

Исходный текст: НаВашисходящийот1204

Открытый текст: ВСеверныйфилиалБанка

Ключ: PfkLQeCIBiSrJuUFTQcd

Шифортекст: тчў□□Х□Ъ□Э□щӨх□і□□льє

Исходный текст: ВСеверныйфилиалБанка

Расшифровка P1, зная P2: НаВашисходящийот1204

Расшифровка P2, зная P1: ВСеверныйфилиалБанка

5 Ответы на контрольные вопросы

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

Если один и тот же ключ (K) был использован для шифрования двух открытых текстов (P1 и P2) с помощью однократного гаммирования, то:

- $C1 = P1 \text{ XOR } K$ • $C2 = P2 \text{ XOR } K$

Зная C1 и C2, можно вычислить:

- $C1 \text{ XOR } C2 = (P1 \text{ XOR } K) \text{ XOR } (P2 \text{ XOR } K) = P1 \text{ XOR } P2$ (Ключ K исключается)

Теперь, зная P1 (один из открытых текстов), можно вычислить P2:

- $P2 = (P1 \text{ XOR } C1 \text{ XOR } C2)$ или $P2 = (C1 \text{ XOR } C2) \text{ XOR } P1$

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

Повторное использование ключа при шифровании однократным гаммированием позволяет злоумышленнику, имеющему доступ к шифротекстам и знающему часть одного из открытых текстов, восстановить другой открытый текст. Однократное гаммирование перестаёт быть однократным, и перестаёт быть безопасным.

3. Как реализуется режим шифрования однократного гаммирования одним ключом двух открытых текстов?
4. Сгенерировать случайный ключ K длиной, равной длине самого длинного из двух открытых текстов (P1 и P2).

5. Шифровать первый открытый текст: $C1 = P1 \text{ XOR } K$

6. Шифровать второй открытый текст: $C2 = P2 \text{ XOR } K$

7. Перечислите недостатки шифрования одним ключом двух открытых текстов.

- Отсутствие безопасности: Зная один открытый текст и оба шифротекста, можно восстановить второй открытый текст.
- Возможность восстановления ключа: При наличии достаточного количества информации о открытых текстах или их структуре, возможна частичная или полная дедукция ключа.
- Уязвимость к частотному анализу: Если открытые тексты имеют предсказуемые элементы или повторяющиеся фрагменты, это упрощает криптоанализ.
- Нарушение принципа одноразового использования: Главный принцип ОTR - одноразовое использование ключа - полностью нарушен, что делает систему эквивалентной намного более слабым шифрам.

5. Перечислите преимущества шифрования одним ключом двух открытых текстов.

Их нет. Использование одного ключа для шифрования нескольких сообщений с помощью одноразового гаммирования - это серьезная ошибка, которая делает шифр абсолютно небезопасным. Нет никаких ситуаций, когда это было бы оправдано.

6 Выводы

В ходе работы было произведено повторное знакомство с элементами криптографии, произведена шифровка и дешифровка данных с помощью ключа, расшифровка сообщений, закодированных одним ключом без ключа, зная только сообщения

Список литературы

- ТУИС