Отчёт по лабораторной работе №8

Дисциплина: Основы информационной безопасности

Барсегян Вардан Левонович НПИбд-01-22

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

1. Создаю функцию encrypt(), которая будет шифровать заданный текст с помощью однократного гаммирования. На вход функция получает открытый текст, также можно задать определенный ключ шифрования. Если ключа нет, то он генерируется рандомно. Сначала исходный текст и ключ шифрования преобразуются в 16-ную СС, затем, применяется операция XOR для каждого элемента ключа и текста. Полученный шифротекст декодируется из 16-ной СС и получается набор из символов.

```
def encrypt(text: str, key: list = None):
    '''
    Выводит шифротекст для заданного текста.
    Если ключа нет, то генерируется случайный ключ
    '''
    if not key:
        key = generate_key(length=len(text))

    text_16 = [ord(char) for char in text]
    key = [ord(el) for el in key]

    print(f"Ключ шифрования:", ' '.join(str(s) for s in key))
    print(f"Исходный текст:", text)
```

```
    encrypted_text = []
    for i in range(len(text)):
        encrypted_text.append(text_16[i] ^ key[i])

    ciphertext = ''.join([chr(i) for i in encrypted_text])
    print(f'Шифротекст: {ciphertext}\n\n')

    return ciphertext
```

2. Генерация ключа, если он не задан, происходит в функции generate_key() из ascii-символов и цифр

```
def generate_key(length: int):
    '''
    Генерация рандомного ключа длины length
    '''
    return random.sample(string.ascii_letters + string.digits, length)
```

3. Работа программмы: (рис. 2.1)

- сначала создается случайный ключ и с этим ключом шифруются тексты p1 и p1 (переменные c1 и c2)
- далее, шифротекст c1 шифруется по ключу c2
- полученный шифротекст c1_c2 шифруется по ключу открытого текста. в результате, получаем второй открытый текст, ранее неизвестный

```
40
41  p1 = 'НаВашисходящийот1204'
42  p2 = 'ВСеверныйфилиалБанка'
43  key = generate_key(20)
44
45  c1 = encrypt(p1, key=key)
46  c2 = encrypt(p2, key=key)
47
48  c1_c2 = encrypt(c1, key=c2)
49
50  encrypt(c1_c2, p1)
51  encrypt(c1_c2, p2)
52
```

Рис. 2.1: Работа программы

4. Полный вывод работы программы (рис. 2.2)

```
PS D:\Рабочий стол\university\сем4\оиб\labs\lab8> & C:/Users/Admin/AppData/Local/Programs/Python/Python311/python.exe "d:
/Рабочий стол/university/сем4/оиб/labs/lab8/gamma.py"
Ключ шифрования: 74 81 112 100 97 55 48 52 66 72 54 117 50 68 119 122 105 76 86 51
Исходный текст: НаВашисходящийот1204
Шифротекст: їwЂєЩЏЏЏОЎэмЊъщиХ~f


Ключ шифрования: 74 81 112 100 97 55 48 52 66 72 54 117 50 68 119 122 105 76 86 51
Исходный текст: ВСеверныйфилиалБанка
Шифротекст: jЎx1єЎЇЮОКЎюЊЪжљЏЃ


Ключ шифрования: 1112 1136 1093 1110 1108 1143 1037 1151 1147 1036 1038 1102 1034 1140 1100 1131 1113 1137 1132 1027
Исходный текст: їwЂєЩЏЏЏОЎэмЊъщиХ~f
Шифротекст: ◄'❁}x|pwr  ♣SЁЏЊЄ


Ключ шифрования: 1053 1072 1042 1072 1096 1080 1089 1093 1086 1076 1103 1097 1080 1081 1086 1090 49 50 48 52
Исходный текст: ◄'❁}x|pwr  ♣SЁЏЊЄ
Шифротекст: ВСеверныйфилиалБанка


Ключ шифрования: 1042 1057 1077 1074 1077 1088 1085 1099 1081 1092 1080 1083 1080 1072 1083 1041 1072 1085 1082 1072
Исходный текст: ◄'❁}x|pwr  ♣SЁЏЊЄ
Шифротекст: НаВашисходящийот1204
```

Рис. 2.2: Работа программы

3 Контрольные вопросы

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

Нужно применить XOR для двух шифротекстов, а к полученному результату применить XOR с ключом, равным известному открытому тексту. Тогда результатом будет второй открытый текст

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

Шифрование будет небезопасным, т.к. с помощью шифротекстов и одного открытого текста можно дешифровать другой текст

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

Каждый текст шифруется однократным гаммированием отдельно с использованием этого ключа

4. Перечислите недостатки шифрования одним ключом двух открытых текстов

Главный недостаток - можно дешифровать открытый текст без знания ключа

5. Перечислите преимущества шифрования одним ключом двух открытых текстов.

Т.к. ключей используется меньше, то тратится меньше памяти на хранение и передачу ключей

4 Выводы

Я применил режим однократного гаммирования на примере кодирования различных исходных текстов одним ключом

Список литературы