Отчет к лабораторной работе №7

Common information

discipline: Основы информационной безопасности
group: НПМбд-02-21
author: Бабина Ю. О.

Цель работы

Освоить на практике применение режима однократного гаммирования.

Выполнение работы

Напишем код на языке программирования Python. Импортируем все необходимые библиотеки:

[1]:
```
import random
import string
```

Напишем функцию для генерации случайного ключа заданной длины:

[2]:
```
def get_random_key(n):
    symbols = string.ascii_letters + string.digits
    return ''.join([random.choice(symbols) for i in range(n)])
```

Далее реализуем функцию для шифрования и дешифрования текста по заданному ключу (функция будет одна для двух процессов, так как операция исключающего или отменяет сама себя):

[3]:
```
def enc_dec(text, key):
    if len(text) != len(key):
        raise ValueError('Длины текста и ключа должны совпадать')
    return ''.join([chr(ord(text[i]) ^ ord(key[i % len(key)])) for i in range(len(text))])
```

Теперь реализуем функцию, которая подбирает ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста:

[4]:
```
def find_keys(text, part):
    all_keys = []
    for i in range(len(text) - len(part) + 1):
        new_key = ""
        for j in range(len(part)):
            new_key += chr(ord(text[i + j]) ^ ord(part[j]))
        all_keys.append(new_key)
    return all_keys
```

Проверим корректность работы написанных нами функций:

```
[5]: text = 'С Новым Годом, друзья!'
key = get_random_key(len(text))
enc_text = enc_dec(text, key)
print(f'Текст: {text}')
print(f'Ключ: {key}')
print(f'Шифротекст: {enc_text}')
print(f'Расшифрованный текст: {enc_dec(enc_text, key)}')

Текст: С Новым Годом, друзья!
Ключ: E3IDtle0Mq9CDN1sEVJXv7
Шифротекст: Ю□еОцЧь□ўяЙсЗуЬ□чSЕсДй□
Расшифрованный текст: С Новым Годом, друзья!
```

```
[6]: part='С Новым'
potential_keys = find_keys(text, part)
print(f'Список ключей: {potential_keys}')

Список ключей: ['\x00\x00\x00\x00\x00\x00\x00', 'Ён#\x0cуwM', '<O/u\x0eж/', '\x1fBV\x02BX\x02', '\x13ж!O!u\x08', 'jMн-\x0c\x7f\x02', '\x1d\x00\x0e\x00\x06u\x00', 'Ёr#\n\x0cwA', '2O)\x00\x0eаM', '\x1fД#\x02Oж\x08', '\x15O!BB\x7f', '\x1fMбO\x06\x0b\x7f', '\x1d\x0cн\nr\x08\x0b', 'Й\x00)~q|p', 'ЁД]}\x05\x07s', '\x15Ш^\t~\x04H']
```

В итоге имеем данную программу:

```
import random
import string

def get_random_key(n):
    symbols = string.ascii_letters + string.digits
    return ''.join([random.choice(symbols) for i in range(n)])

def enc_dec(text, key):
    if len(text) != len(key):
        raise ValueError('Длины текста и ключа должны совпадать')
    return ''.join([chr(ord(text[i]) ^ ord(key[i % len(key)])) for i in
range(len(text))])

def find_keys(text, part):
    all_keys = []
    for i in range(len(text) - len(part) + 1):
        new_key = ""
        for j in range(len(part)):
            new_key += chr(ord(text[i + j]) ^ ord(part[j]))
        all_keys.append(new_key)
    return all_keys

text = 'С Новым Годом, друзья!'
key = get_random_key(len(text))
enc_text = enc_dec(text, key)
print(f'Текст: {text}')
print(f'Ключ: {key}')
print(f'Шифротекст: {enc_text}')
print(f'Расшифрованный текст: {enc_dec(enc_text, key)}')

part='С Новым'
potential_keys = find_keys(text, part)
print(f'Список ключей: {potential_keys}')
```

Контрольные вопросы

1. Однократное гаммирование-это метод шифрования, при котором каждый символ открытого текста гаммируется с соответствующим символом ключа только один раз.

2. Недостатки однократного гаммирования:

- Уязвимость к частотному анализу из-за сохранения частоты символов открытого текста в шифротексте.
- Необходимость использования одноразового ключа, который должен быть длиннее самого открытого текста.
- Нет возможности использовать один ключ для шифрования разных сообщений.

3. Преимущества однократного гаммирования:

- Высокая стойкость при правильном использовании случайного ключа.
- Простота реализации алгоритма.
- Возможность использования случайного ключа.

4. Длина открытого текста должна совпадать с длиной ключа, чтобы каждый символ открытого текста гаммировался с соответствующим символом ключа.

5. В режиме однократного гаммирования используется операция XOR (исключающее ИЛИ), которая объединяет двоичные значения символов открытого текста и ключа для получения шифротекста. Особенность XOR - если один из битов равен 1, то результат будет 1, иначе 0.

6. Для получения шифротекста по открытому тексту и ключу каждый символ открытого текста гаммируется с соответствующим символом ключа с помощью операции XOR.

7. По открытому тексту и шифротексту невозможно восстановить действительный ключ, так как для этого нужна информация о каждом символе ключа.

8. Необходимые и достаточные условия абсолютной стойкости шифра:

- Ключи должны быть случайными и использоваться только один раз.
- Длина ключа должна быть не менее длины самого открытого текста.
- Ключи должны быть храниться и передаваться безопасным способом.

Вывод

В рамках выполнения данной лабораторной работы я освоила на практике применение режима однократного гаммирования.

Список литературы

- https://bugtraq.ru/library/books/crypto/chapter7/
- https://www.youtube.com/watch?v=tAjBULW_OjQ