РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра информационных технологий

Дисциплина: Информационная безопасность

Студент: Ду Нашсименту Висенте Феликс Жозе

Группа: НкНбд01-20

Москва 2023

8.1. Цель работы

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

8.2. Порядок выполнения работы

Я создал программу на Python для шифрования и расшифровки текста, используя операцию XOR с случайно сгенерированным ключом. Позвольте объяснить, как работает Мой код шаг за шагом:

Функция hextext(text):

Эта функция принимает строку text в качестве входных данных. Каждый символ во входной строке преобразуется в его соответствующее значение ASCII, а затем значение ASCII преобразуется в его шестнадцатеричное представление. Результатом является строка шестнадцатеричных значений, представляющих входной текст. Функция gen_klyo(size):

Эта функция генерирует случайную строку указанного размера size, используя буквы (как заглавные, так и строчные) и цифры. Сгенерированная строка служит в качестве ключа шифрования. Функция encrypted(ms1, ms2):

Эта функция принимает две входные строки ms1 и ms2. Каждый символ в обеих входных строках преобразуется в его значение ASCII. Производится операция XOR между соответствующими значениями ASCII символов из ms1 и ms2. Результат XOR затем снова преобразуется в символ с использованием функции chr(). Функция возвращает строку, представляющую зашифрованное сообщение. В вашем коде вы шифруете два сообщения (P1 и P2) с использованием случайно сгенерированного ключа, а затем расшифровываете их, чтобы получить исходные сообщения.

```
[18] import random
     import string
```

```
def hextext(text):
  t="".join(hex(ord(i))[2:]for i in text)
  return t
```

```
[20] def gen_klyo(size):
       g=''.join(random.choice(string.ascii_letters + string.digits)for _ in range(size))
       return g
```

```
[21] def encrypted(ms1,ms2):
       ms1=[ord(i) for i in ms1]
       ms2=[ord(i) for i in ms2]
       create=''.join(chr(a^b)for a,b in zip(ms1,ms2))
       return create
```

```
[22] P1= " НаВашисходящийот1204"
     P2= " ВСеверныйфилиалБанка"
```

```
[23] key = gen_klyo(len(P1))
     print(key)
     hex_klyuch=hextext(key)
     print("Ключ в шестнадцатиричном виде: {}".format(hex_klyuch))
```

```
AHwgLQ4cVVXAyHW7Zua7F
Ключ в шестнадцатиричном виде: 414877674c5134635656584179485737 5a75613746
```

```
[24] C1= encrypted(P1,key)
     C2= encrypted(P2,key)
     print("Шифрованный текст: {}".format(C1))
     print("Шифрованный текст: {}".format(C2))

     decrypt=encrypted(C1,C2)
     print("Расфрованный текст: {}".format(encrypted(decrypt,P2)))
     print("Расфрованный текст: {}".format(encrypted(decrypt,P1)))
```

```
Шифрованный текст: asчvȌЙЌТГѬѪЎаѰѮљИDS⍰r
Шифрованный текст: aњiђѼЌVўHѮMѹтѰӂЌыxЌЍѶ
Расфрованный текст:  НаВашисходящийот1204
Расфрованный текст:  ВСеверныйфилиалБанка
```

Выводы

Этот код выполняет простую форму шифрования XOR. Это базовая демонстрация XOR-шифрования и не должен использоваться для безопасных коммуникационных целей. Кроме того, код предполагает, что входное сообщение и ключ находятся в той же кодировке символов. Если это не так, результаты могут не соответствовать ожиданиям.