РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

Презентация

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 7

дисциплина: Информационная безопасность

Студент: Пиняева Анна Андреевна

Группа: НФИбд-02-20

МОСКВА

2023

Цель работы

Освоить на практике применение режима однократного гаммирования

Задача

Нужно подобрать ключ, чтобы получить сообщение «С Новым Годом, друзья!». Требуется разработать приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования. Приложение должно:

1. Определить вид шифротекста при известном ключе и известном открытом тексте.

2. Определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

Ход работы

1. Код программы с комментарием (рис. 1).

Рис. 1 Код:

```python
# Импортируем модули random и string для генерации ключа и работы с символами.
import random
import string


def generate_key(size):
    # Генерируем случайный ключ заданного размера из букв и цифр.
    characters = string.ascii_letters + string.digits
    return ''.join(random.choice(characters) for _ in range(size))

def text_to_binary(text):
# Преобразуем текст в его бинарное представление, где каждый символ представлен в виде 8-битного бинарного числа.
    return ''.join(format(ord(char),'08b')for char in text)

def binary_text(binare_str):
# Преобразуем бинарную строку обратно в текст, разбивая бинарные числа на группы по 8 бит и преобразуя их в символы.
    binary_chunks = [binary_str[i:i+8] for i in range(0, len(binary_str),8)]
    return ''.join(chr(int(chunk, 2))for chunk in binary_chunks)

def xor_encrypt (text, key):
    # Выполняем операцию XOR между символами текста и ключа.
    encrypted = [ord(a) ^ ord(b) for a, b in zip(text, key)]
    # Преобразуем полученные числа обратно в символы.
    return ''.join(chr(encrypted_char)for encrypted_char in encrypted)

msg = "С Новым годом, друзья!" # Сообщение

# Генерируем ключ той же длины, что и сообщение
key = generate_key(len(msg))

print ("Ключ:", key)

# Зашифровываем сообщение, используя XOR сгенерированного ключа.
msg2 = xor_encrypt(msg, key)

# Выводим зашифрованное сообщение в двоичном формате.
binary = text_to_binary (msg2)

print ("Зашифрованный текст:", binary)

# Расшифровываем сообщение, используя тот же ключ.
msg3 = xor_encrypt (msg2, key)

print ("Расшифрованный текст:", msg3)
```

2. Результат выполнения (рис. 2).

Рис. 2 Результат:

```
Ключ: H3gkDDO6euZ6Lit9QLt8zc
Зашифрованный текст: 10001101001000100111000111101010001010101100011101101000000111110001110011000101101000101011010001001011100011011101000000100010001110000010001010101010010000001101100001000110000001111100010000111000111010010000011010101000010
Расшифрованный текст: С Новым годом, друзья!
```

N|Solid

Выводы