# Отчёт по лабораторной работе №7

## Основы информационной безопасности

Паращенко А.Д.

14 октября 2024

Российский университет дружбы народов, Москва, Россия

# Цель работы

# Цель работы

Освоить на практике применение режима однократного гаммирования.

# Выполнение лабораторной работы

# Функция для генерации ключа.

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

Рис. 1: Генерация ключа

# Функция для (де)шифрования.

```
def crypt(text, key):
    new_text = ''
    for i in range(len(text)):
        new_text += chr(ord(text[i]) ^ ord(key[i % len(key)]))
    return new_text
```

Рис. 2: Шифрование

# Функция для подбора ключа.

```python
def find_key(text, fragment):
    possible_keys = []
    for i in range(len(text) - len(fragment) + 1):
        possible_key = ''
        for j in range(len(fragment)):
            possible_key += chr(ord(text[i + j]) ^ ord(fragment[j]))
        possible_keys.append(possible_key)
    return possible_keys
```

Рис. 3: Подбор ключа

# Код для вывода результатов и результаты.

```
t = 'С Новым Годом, друзья!'
key = generate_key(t)
encrypt = crypt(t, key)
decrypt = crypt(encrypt, key)
keys = find_key(encrypt, "С Новым Годом")
fragment = "С Новым Годом"

print('Открытый текст:', t, "\nКлюч:", key, "\nШифротекст:", encrypt, "\nИсходный текст:", decrypt)
print('Возможные ключи:', keys)
print('Расшифрованный фрагмент:', crypt(encrypt, keys[0]))
```

```
Открытый текст: С Новым Годом, друзья!
Ключ: xTfijecGs9D8jHwR6lMOOw
Шифротекст: ѧtoljRugDIWIidWAVRQfÈV
Исходный текст: С Новым Годом, друзья!
Возможные ключи: ['xTfijecGs9D8j', 'shJf\x1c\x14hp\x14N2hj', 'ZVE\x10mb\\%c8bwx', 'vђ3as+;è\x15hèwZ', 'y9BaRLLЏEьbX3', '\x0fШO^5;:ÝÜwRH\x13', '~G)9BMjD¢X
B\x11F', 'up\x1aN4\x1djwuH\x1bD?', 'A%m8dRжue\x11N=<', 'ãè\x1bhiMZiCO7>Ж']
Расшифрованный фрагмент: С Новым Годом,ЊÉПхЛЏ%
```

**Рис. 4:** Результаты

# Листинг программы.

```python
import random
import string

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

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

def find_key(text, fragment):
    possible_keys = []
    for i in range(len(text) - len(fragment) + 1):
        possible_key = ''
        for j in range(len(fragment)):
            possible_key += chr(ord(text[i + j]) ^ ord(fragment[j]))
        possible_keys.append(possible_key)
    return possible_keys

t = 'С Новым Годом, друзья!'
key = generate_key(t)
encrypt = crypt(t, key)
decrypt = crypt(encrypt, key)
keys = find_key(encrypt, "С Новым Годом")
fragment = "С Новым Годом"

print('Открытый текст:', t, "\nКлюч:", key, "\nШифротекст:", encrypt, "\nИсходный текст:", decrypt)
print('Возможные ключи:', keys)
print('Расшифрованный фрагмент:', crypt(encrypt, keys[0]))
```

# Вывод

# Вывод

В результате выполнения работы мы научились на практике применять режим однократного гаммирования.

# Список литературы

# Список литературы

1) https://esystem.rudn.ru/pluginfile.php/2357157/mod_resource/content/2/007-lab_crypto-gamma.pdf