

# **Лабораторная работа № 3**

**Дисциплина: Информационная безопасность**

Новосельцев Данила Сергеевич

# Содержание

1	Цель работы	4
2	Код программы	5
3	Вывод	8
4	Список литературы	9

# Список иллюстраций

2.1	Вывод программы для текста на русском . . . . .	7
-----	---	---

# **1 Цель работы**

Освоение на практике применения режима однократного гаммирования.

## 2 Код программы

```
import random
import string

class TextEncoding:

    @staticmethod
    def determine_alphabet(text):
        if text[0] in string.ascii_lowercase:
            return string.ascii_lowercase + string.digits
        else:
            return "абвгдеёжзийклмнопрстуфхцчшщъыьэюя" + string.digits

    @staticmethod
    def generate_key(size, alphabet):
        return "".join(random.choice(alphabet) for _ in range(size))

    @staticmethod
    def to_hex(coding):
        return " ".join(hex(ord(character))[2:] for character in coding)

    @staticmethod
    def encode_string(text, key):
```

```

        return "".join(chr(ord(char) ^ ord(key_char)) for char, key_char in zip(text, key))

    @staticmethod
    def find_possible_keys(text, fragment):
        key_length = len(fragment)
        possible_keys = []

        for index in range(len(text) - key_length + 1):
            key = [chr(ord(char) ^ ord(key_char)) for char, key_char in zip(text[index:index + key_length], fragment)]
            presumed_plaintext = TextEncoding.encode_string(text, key)

            if fragment in presumed_plaintext:
                possible_keys.append(''.join(key))

        return possible_keys

plaintext = input("Введите открытый текст: ")
alphabet = TextEncoding.determine_alphabet(plaintext)
key = TextEncoding.generate_key(len(plaintext), alphabet)

print(f"Ключ: {key}", f"Ключ в 16 бит: {TextEncoding.to_hex(key)}", sep='\n')

ciphertext = TextEncoding.encode_string(plaintext, key)
print(f"Зашифрованный текст: {ciphertext}", f"Зашифрованный текст в 16 бит: {TextEncoding.to_hex(ciphertext)}", sep='\n')

decrypted_text = TextEncoding.encode_string(ciphertext, key)
print("Расшифрованный текст:", decrypted_text)

```

```
known_fragment = input("Введите фрагмент открытого текста: ")
possible_keys = TextEncoding.find_possible_keys(ciphertext, known_fragment)
print("Возможные ключи для шифротекста:", possible_keys)
```

```
Введите открытый текст: С Новым Годом, друзья!
Ключ: 7ъэ0ю7амшсрувяэ7з4чпсф
Ключ в 16 бит: 37 44а 44d 30 44е 37 430 43с 448 441 440 443 432 44f 437 37 437 34 447 43f 441 444
Зашифрованный текст: ЖѡРŸ|Œ"М[="t}""ъ3Гwŷps""ѡ
Зашифрованный текст в 16 бит: 416 46а 50 40е 7с 47с с 41с 5b 7f 74 7d е 463 417 403 77 477 70 73 е
Расшифрованный текст: С Новым Годом, друзья!
Введите фрагмент открытого текста: Новым
Возможные ключи для шифротекста: ['\x0bТъЕр']
```

Рис. 2.1: Вывод программы для текста на русском

## **3 Вывод**

Освоил на практике применение режима однократного гаммирования.



## 4 Список литературы

[1] [https://esystem.rudn.ru/pluginfile.php/2090284/mod\\_resource/content/2/007-lab\\_crypto-gamma.pdf](https://esystem.rudn.ru/pluginfile.php/2090284/mod_resource/content/2/007-lab_crypto-gamma.pdf)