Лабораторная работа 1

Классические шрифты

Вариант 18

Раконяц Даниела

7 группа

2023

**Шифр Хилла** - полиграммный шифр подстановки (элементы исходного открытого текста заменяются зашифрованным текстом в соответствии с некоторым правилом), в котором буквы открытого текста заменяются группами с помощью линейной алгебры. Для латинского алфавита каждой букве сопоставляется число, например, A – 0, B – 1, C – 2, …, Z – 25. В общем случае соответствия “буква – число” можно выбрать произвольно.

## Шифрование

Открытый текст представляет собой n-мерный вектор. Ключ – квадратная матрица размера n x n. Для получения шифротекста ключ умножается на открытый текст по модулю выбранной числовой схемы.

**Шифр простой замены** - простой подстановочный шифр, моноалфавитный шифр — класс методов шифрования, которые сводятся к созданию по определённому алгоритму таблицы шифрования, в которой для каждой буквы открытого текста существует единственная сопоставленная ей буква шифр-текста.

Задание:

1)Шифровка

| Шифр | Ключ | Открытый текст |
| --- | --- | --- |
| Хилла | 26 30 20 32 | криптологи |

2)Расшифровка

| Шифр | Ключ | Шифр-текст |
| --- | --- | --- |
| Простой замены | ЪЗЙЩЛШУИРДЧЁСФЕТЦЭМАВПЯБЫЖЮОХНЬГК | ЪЗЭДЁТМ |

**Код:**

**from typing import Tuple**

**import numpy as np**

**alphabet = "абвгдеёжзийклмнопрстуфхцчшщъыьяюя"**

**open\_text = "криптологи"**

**m = 33**

**keys = np.array([[26, 30], [20, 32]])**

**size = len(open\_text)**

**y = [None] \* size**

**for i in range(0, size - 1, 2):**

**a = np.array([alphabet.find(open\_text[i]), alphabet.find(open\_text[i + 1])])**

**temp = a.dot(keys)**

**y[i], y[i + 1] = alphabet[temp[0] % m], alphabet[temp[1] % m]**

**encrypted\_text = ''.join(y)**

**print(encrypted\_text)**

**def simple\_decryption(alphabet: str, key: str, message: str) -> str:**

**if len(alphabet) != len(key):**

**raise ValueError(**

**f'Alphabet and key expected to have same length. But got alphabet: {len(alphabet)}, key: {len(key)} '**

**)**

**return "".join([alphabet[key.find(i)] for i in message])**

**print(**

**simple\_decryption(**

**alphabet="абвгдеёжзийклмнопрстуфхцчшщъыьяюя",**

**key="ЪЗЙЩЛШУИРДЧЁСФЕТЦЭМАВПЯБЫЖЮОХНЬГК".lower(),**

**message="ЪЗЭДЁТМ".lower(),**

**)**

**)**

**Результат:**

