Федеральное государственное автономное

образовательное учреждение

Высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт космических и информационных технологий |
| институт |
| Программная инженерия |
| кафедра |

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ**

|  |
| --- |
| Простые симметричные шифры |
| тема |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Преподаватель | |  |  |  | Ю.В. Потылицина |
|  | |  | подпись, дата |  | инициалы, фамилия |
| Студент | КИ21-17/1Б, 032156940 |  |  |  | Н.А. Самарин |
|  | номер группы, зачётной книжки |  | подпись, дата |  | инициалы, фамилия |

Красноярск 2023

**СОДЕРЖАНИЕ**

1 Задание............................................................................................................... 3

2 Вариант задания................................................................................................ 3

3 Описание Шифра Хилла.................................................................................. 3

4 Криптостойкость Шифра Хилла..................................................................... 3

5 Блоксхемы и реализация на Python Шифра Хилла....................................... 4

6 Описание, блоксхемы и реализация на Python Модифицированного  
 Шифра Хилла.................................................................................................... 11

7 Выводы.............................................................................................................. 19

**1 Задание**

Согласно вашему персональному варианту или индивидуальному  
заданию преподавателя разработайте и составьте в виде блоксхемы алгоритмы  
шифрования и дешифрования текста. Убедитесь в правильности составления  
алгоритмов и затем на языке программирования составьте программу, которая  
реализует данные алгоритмы. На ряде контрольных примеров открытого  
текста, состоящего из различного количества символов, проверьте  
правильность работы алгоритмов шифрования и дешифрования.  
Самостоятельно или с помощью преподавателя придумайте оригинальный  
способ модификации шифра с целью повышения его криптостойкости. Внесите  
изменения в исходный алгоритм и программу. Проверьте работоспособность  
алгоритма на тестовых примерах.

**2 Вариант задания**

Вариант 8(24) - Шифр Хилла.

**3 Описание Шифра Хилла**

Шифр Хилла – полиграммный шифр подстановки, 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.

Открытый текст представляет собой n-мерный вектор. Если текст по  
длине превышает n, он делится на блоки длиной n, не достающие символы  
можно заменить любым символом алфавита. Символы блоков и ключ  
заменяются на числа по индексу символа в алфавите. Ключ – квадратная  
матрица размера n x n. Для получения шифротекста ключ умножается на  
открытый текст по модулю выбранной числовой схемы, в случае латинского  
алфавита - 26. При наличии нескольких блоков текста, каждый из них  
умножается подобным образом на ключ. Результат умножения переводится  
обратно в символы алфавита, блоки складываются в единый шифротекст.

Чтобы расшифровать зашифрованный текст, ключ – матрица должна  
иметь обратную по модулю матрицу. Такое возможно, если детерминант  
матрицы не равен нулю и не имеет общих делителей с основанием модуля.  
Чтобы упростить задачу, можно выбрать в качестве основания модуля простое  
число, путем добавления в алфавит новых элементов, например, знаков  
препинания. Таким образом любой детерминант не будет иметь общих  
делителей с основанием модуля. Расшифровка происходит подобно  
шифрованию, но, вместо умножению текста на матрицу ключа, происходит  
умножение на обратную матрицу.

**4 Криптостойкость Шифра Хилла**

Атаковать грубой силой шифр Хилла сложно. При использовании  
латинского алфавита, если размерность матрицы – ключа n x n, то всего  
существует 26n^2 таких матриц. Но не каждая матрица обратима, поэтому  
область несколько сужается. Количество обратимых матриц можно рассчитать  
с помощью китайской теоремы об остатках. Например, количество обратимых  
матриц по модулю 26 равно: |K| = 26n^2(1 – 1/2)(1 – 1/22)…(1 – 1/2n) (1 –  
1/13)(1 – 1/132)…(1 – 1/13n) В результате эффективное пространство ключей  
составляет около 4.64n2 – 1.7. Для ключа размера 5x5 этот параметр будет  
равен 114 битам, что делает вариант полного перебора неприменимым.

Так же шифр Хилла не поддается частотному анализу, так как каждый  
символ открытого текста принимает участие в шифровании.

Шифр Хилла очень уязвим для атаки по открытому тексту, так как в нем  
используются линейные операции. Если знать m пар “открытое сообщение”/  
“зашифрованное сообщение”, то можно вычислить ключ. Предположим, что  
мы перехватили 3 пары “открытое сообщение”/ “зашифрованное сообщение”.  
Если составить матрицы P и C из этих пар, то можно найти ключ K. Если  
матрица P не будет являться обратимой, то необходимо задействовать другие m  
наборов пар “открытое сообщение”/ “зашифрованное сообщение”. K = P-1 \* C

**5 Блоксхемы и реализация на Python Шифра Хилла**

Составленные блоксхемы алгоритмов шифрования и дешифрования  
представлены на рисунках 1 и 2.

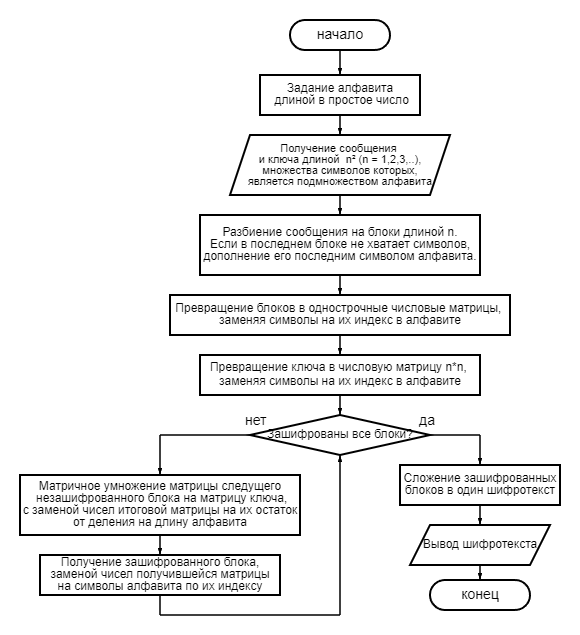


Рисунок 1 – Блоксхема шифрования Шифра Хилла

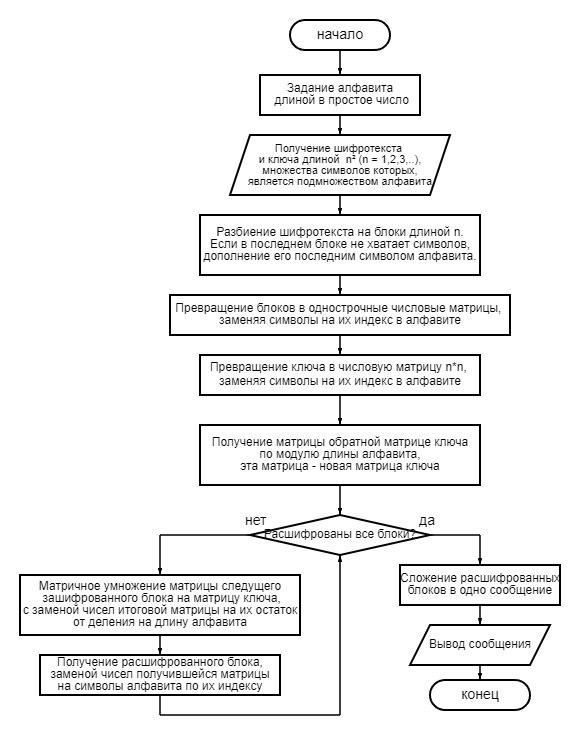


Рисунок 2 – Блоксхема дешифрования Шифра Хилла

Шифрование и дешифрование были реализованы в виде одной функции  
написанной на Python, для дешифрования нужно передать флаг decrypt=True. В  
качестве алфавита был выбран русский алфавит дополненный знаками "?., ",  
для того чтобы длина алфавита составляла простое число, для упрощения  
процесса дешифрования. Текст программы представлен в листинге 1.

Листинг 1 – Реализация Шифра Хилла

import numpy as np  
import math  
from sympy import Matrix  
alphabet = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя?., "  
alphabet\_len = len(alphabet)  
def hill\_cipher(message, key, decrypt=False):  
 if len(message) == 0 or len(key) == 0:  
 return "Неккоректный ввод"  
 n = math.sqrt(len(key))  
 if n % 1 == 0:  
 n = int(n)  
 else:  
 return "Неккоректный ввод"  
 if (not (set(message) <= set(alphabet))) or (not (set(key) <= set(alphabet))  
 ):

Окончание листинга 1

return "Неккоректный ввод"  
 splitted\_message = list([list(message[i:i + n]) for i in range(0, len(messag  
 e), n)])  
 while len(splitted\_message[-1]) < n:  
 splitted\_message[-1].append(alphabet[-1])  
 digit\_message = []  
 for i in range(0, len(splitted\_message)):  
 digit\_message.append([])  
 for j in splitted\_message[i]:  
 digit\_message[i].append(alphabet.index(j))  
 digit\_key = []  
 for i in key:  
 digit\_key.append(alphabet.index(i))  
 matrix\_key = np.array([digit\_key[i:i + n] for i in range(0, len(digit\_key),   
 n)])  
 if decrypt:  
 matrix\_key = np.array(Matrix(matrix\_key).inv\_mod(alphabet\_len))  
 ciphertext = ""  
 for block in digit\_message:  
 new\_block = [i % alphabet\_len for i in np.array(block).dot(matrix\_key)]  
 for i in new\_block:  
 ciphertext += alphabet[i]  
 return ciphertext

Примеры использования шифра представлены на рисунках ниже.

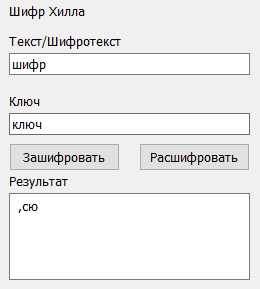


Рисунок 3 – Шифрование 1

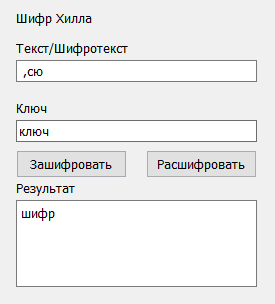


Рисунок 4 – Дешифрование 1

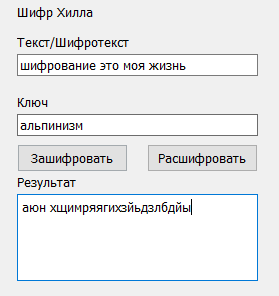


Рисунок 5 – Шифрование 2

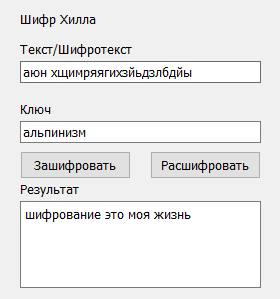


Рисунок 6 – Дешифрование 2

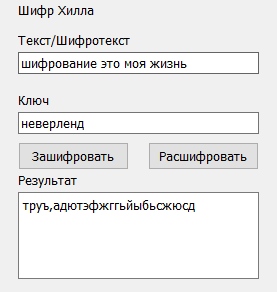


Рисунок 7 – Шифрование 3

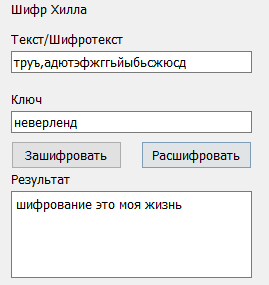


Рисунок 8 – Дешифрование 3

**6 Описание, блоксхемы и реализация на Python  
Модифицированного Шифра Хилла**

Путём модифицирования Шифра Хилла выбран вариант использования  
нескольких ключей. При использовании нескольких ключей каждый  
следующий блок исходного текста кодируется следующим ключем (если  
блоков больше чем ключей, ключи начинают идти по кругу). Выбран вариант с  
ключами одинаковой длины. Эта модификация увеличивает криптостойкость  
шифра. Так при грубом переборе ключей количество возможных вариантов при  
использовании нашего алфавита длиной 37 и 2ух ключей длиной 9(n\*n, n=3)  
равно 379 \* 379, в отличии от 379 при использовании оригинального шифра.  
Также устраняется главная уязвимость Шифра Хилла - атака по открытому  
тексту. При незнании длины ключей и их количества атака по открытому  
тексту используемая с оригинальным Шифром Хилла становится  
неприменимой. На рисунках ниже представлены составленные блоксхемы  
алгоритмов шифрования и дешифрования для модифицированного Шифра  
Хилла (части отличные от оригинального шифра выделены жирным шрифтом).

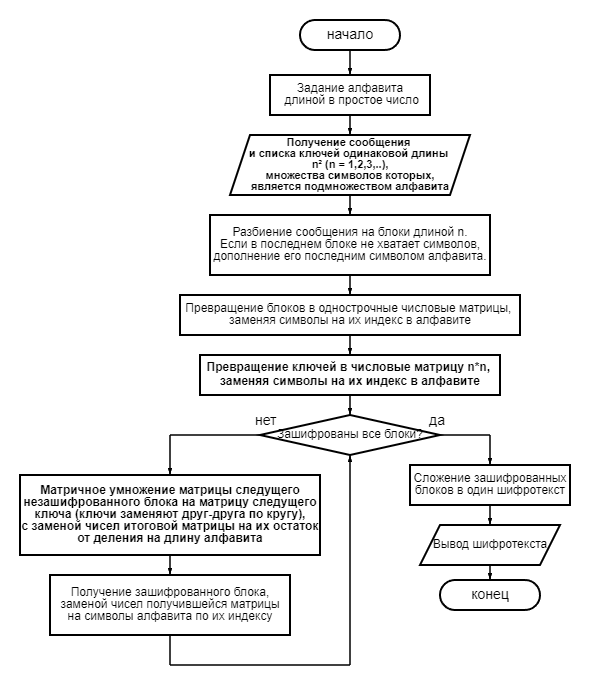


Рисунок 9 – Блоксхема шифрования модифицированного Шифра Хилла

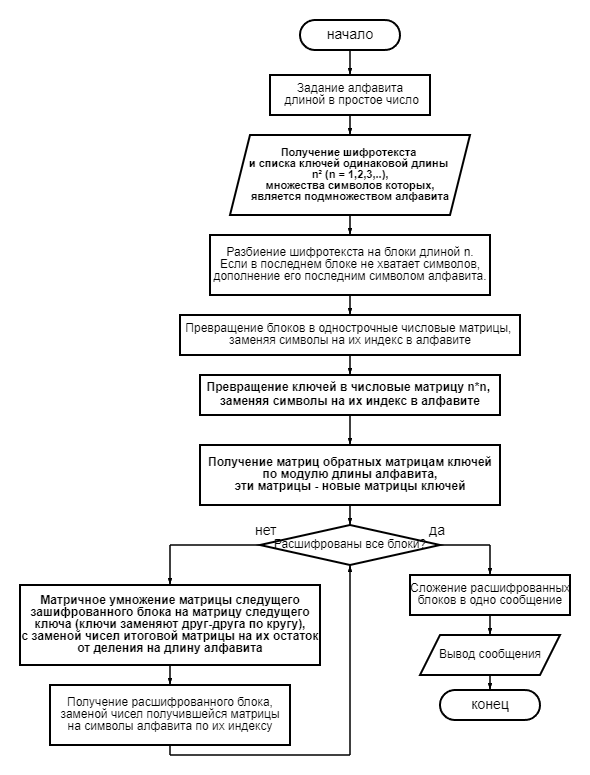


Рисунок 10 – Блоксхема дешифрования модифицированного Шифра Хилла

Реализация модифицированного Шифра Хилла была так же разработана  
на Python. Текст программы представлен в листинге 2.

Листинг 2 – Реализация модифицированного Шифра Хилла

import numpy as np  
import math  
from sympy import Matrix  
alphabet = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя?., "  
alphabet\_len = len(alphabet)  
def hill\_cipher\_modified(message, keys, decrypt=False):  
 if len(message) == 0 or min([len(i) for i in keys]) == 0:  
 return "Неккоректный ввод"  
 if type(keys) != list or len(keys) < 2:  
 return "Неккоректный ввод"  
 n = math.sqrt(len(keys[0]))  
 if n % 1 == 0:  
 n = int(n)  
 else:  
 return "Неккоректный ввод"  
 for i in range(1, len(keys)):  
 if len(keys[i]) != len(keys[0]):  
 return "Неккоректный ввод"

Окончание листинга 2

if (not (set(message) <= set(alphabet))) or (not (set("".join(keys)) <= set(  
 alphabet))):  
 return "Неккоректный ввод"  
 splitted\_message = list([list(message[i:i + n]) for i in range(0, len(messag  
 e), n)])  
 while len(splitted\_message[-1]) < n:  
 splitted\_message[-1].append(alphabet[-1])  
 digit\_message = []  
 for i in range(0, len(splitted\_message)):  
 digit\_message.append([])  
 for j in splitted\_message[i]:  
 digit\_message[i].append(alphabet.index(j))  
 digit\_keys = []  
 for key in keys:  
 digit\_keys.append([])  
 for i in key:  
 digit\_keys[-1].append(alphabet.index(i))  
 matrix\_keys = []  
 for key in digit\_keys:  
 matrix\_key = np.array([key[i:i + n] for i in range(0, len(key), n)])  
 matrix\_keys.append(matrix\_key)  
 if decrypt:  
 for i in range(0, len(matrix\_keys)):  
 matrix\_key = np.array(Matrix(matrix\_keys[i]).inv\_mod(alphabet\_len))  
 matrix\_keys[i] = matrix\_key  
 ciphertext = ""  
 for i in range(0, len(digit\_message)):  
 new\_block = [i % alphabet\_len for i in np.array(digit\_message[i]).dot(ma  
 trix\_keys[i % len(matrix\_keys)])]  
 for i in new\_block:  
 ciphertext += alphabet[i]  
 return ciphertext

Примеры использования модифицированного шифра представлены на  
рисунках ниже.

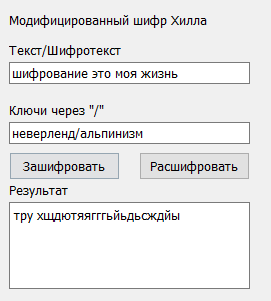


Рисунок 11 – Модифицированное шифрование 1

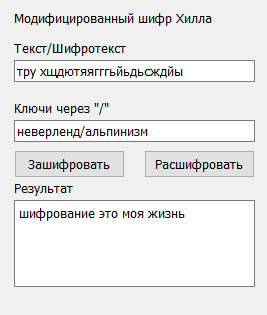


Рисунок 12 – Модифицированное дешифрование 1

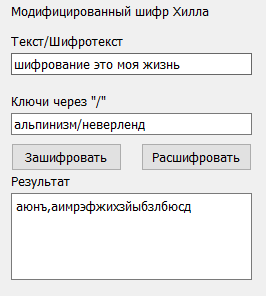


Рисунок 13 – Модифицированное шифрование 2

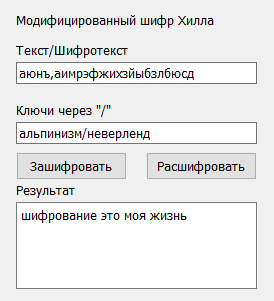


Рисунок 14 – Модифицированное дешифрование 2

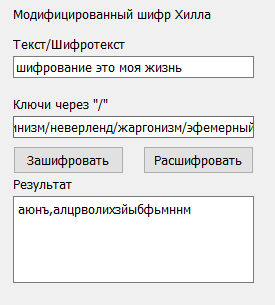


Рисунок 15 – Модифицированное шифрование 3

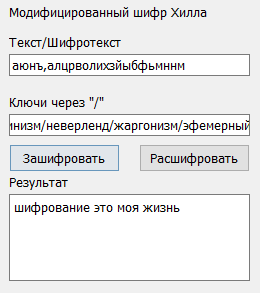


Рисунок 16 – Модифицированное дешифрование 3

**7 Выводы**

В процессе работы была изучена теория простых симметричных шифров  
и Шифра Хилла в частности, созданы блоксхемы его алгоритма и программа  
реализующая его. Также были изучены варианты модификации Шифра Хилла и  
реализован один из них.