Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
(УНИВЕРСИТЕТ ИТМО)

Факультет «Систем управления и робототехники»

# ОТЧЕТ
# О ЛАБОРАТОРНОЙ РАБОТЕ №1

По дисциплине «Практическая линейная алгебра»
на тему:
«Кодирование и шифрование»


Студент:
Зыкин Л.В.
группа R3335
ИСУ 470912

Преподаватель:
Егор Витальевич Догадин,


г. Санкт-Петербург
2024

# 1 Ход выполнения работы

## 1.1 Задание 1: Шифр Хилла

Возьмём русский алфавит, уберём из него букву "ё", а затем добавим пробел и всё цифры.

Получим:

абвгдежзийклмнопрстуфхцчшщъыьэюя 0123456789

Пронумеруем знаки получившегося алфавита числами от 0 до n − 1, где n - общее количество символов в алфавите. В нашем случае n = 43.

• Придумаем сообщение из 12 символов.

после 5 дома

• Придумаем три матрицы-ключа размеров 2 × 2, 3 × 3 и 4 × 4.

$$2 \times 2: \begin{bmatrix} 5 & 1 \\ 1 & 8 \end{bmatrix}$$

$$3 \times 3: \begin{bmatrix} 1 & 4 & 5 \\ 2 & 2 & 4 \\ 3 & 1 & 2 \end{bmatrix}$$

$$4 \times 4: \begin{bmatrix} 1 & 1 & 2 & 2 \\ 2 & 1 & 2 & 3 \\ 4 & 2 & 1 & 7 \\ 3 & 5 & 4 & 1 \end{bmatrix}$$

Проверим, чтобы определители матриц-ключей не имели общих делителей с числом 43.

2 × 2: det = 39

3 × 3: det = 12

4 × 4: det = -27

Общих делителей нет, значит матрица может быть обратима.

Зашифруем сообщение с помощью каждого из ключей, используя метод шифрования

Хилла.

1. Заменим каждый символ сообщения на его индекс в алфавите.

’п’ → 15, ’о’ → 14, ’с’ → 17, ’л’ → 11, ’е’ → 5, ’ ’ → 32, ’5’ → 38, ’ ’ → 32, ’д’ → 4,

’о’ → 14, ’м’ → 12, ’а ’ → 32.

Получили “после 5 дома”→ [15, 14, 17, 11, 5, 32, 38, 32, 4, 14, 12, 32]

2. Разделим сообщение на блоки по 2 символа:

[15, 14], [17, 11], [5, 32], [38, 32], [4, 14], [12, 32].

Разделим сообщение на блоки по 3 символа:

[15, 14, 17], [11, 5, 32], [38, 32, 4], [14, 12, 32]

Разделим сообщение на блоки по 4 символа:

[15, 14, 17, 11], [5, 32, 38, 32], [4, 14, 12, 32]

3. Умножим каждый вектор на матрицу ключей и найдём для каждого элемента матрицы остаток от деления на размер алфавита.

Для этого используем формулы:

Для 2 × 2:

$$R = \begin{bmatrix} v_1 m_{11} + \ v_2 m_{21} & v_1 m_{12} + v_2 m_{22} \end{bmatrix}$$

$$v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} - \text{вектор}, M = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} - \text{матрица} - \text{ключ } 2 \ \times \ 2$$

Для 3 × 3

$$R = \begin{bmatrix} v_1 m_{11} + \ v_2 m_{21} + v_3 m_{31} \\ v_1 m_{12} + \ v_2 m_{22} + v_3 m_{32} \\ v_1 m_{13} + \ v_2 m_{23} + v_3 m_{33} \end{bmatrix}$$

$$v = \begin{bmatrix} v_1 \\ v_1 \\ v_1 \end{bmatrix} - \text{вектор}, M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} - \text{матрица} - \text{ключ } 3 \ \times \ 3$$

Аналогично для матрицы 4 × 4.

Для матрицы 2 × 2 получим: [3, 41], [10, 19], [14, 3], [7, 36], [34, 30], [6, 10].

Для матрицы 3 × 3 получим: [27, 40, 7], [19, 31, 16], [14, 27, 25], [7, 8, 32].

Для матрицы 4 × 4 получим: [42, 25, 10, 22], [5, 42, 2, 15], [20, 13, 22, 33].

4. Преобразуем полученные индексы обратно в символы с помощью алфавита.

абвгдежзийклмнопрстуфхцчшщъыьэюя 0123456789

[3, 41], [10, 19], [14, 3], [7, 36], [34, 30], [6, 10] → "г8куогз31южк"

[27, 40, 7], [19, 31, 16], [14, 27, 25], [7, 8, 32] → "ы7зтяроыщзи "

[42, 25, 10, 22], [5, 42, 2, 15], [20, 13, 22, 33] → "9щкце9впфнц0"

5 Сымитируем вредоносное вмешательство в зашифрованные собщения. Заменим в каждом

из них по три символа на какие-то другие (случайные) символы из нашего алфавита.

2 × 2: "г8куогб31мбк"

3 × 3: "ы7зтяряы45и "

4 × 4: "9щуце9вп76ц0"

6 Расшифруем каждое из получившихся сообщений, используя обратные матрицы от матриц-ключей.

А именно:

1. Вычислим матрицу алгебраических дополнений $c_{ij}$, где каждый элемент $c_{ij}$ равен

$(-1)^{i+j} \cdot M_{ij}$ , а $M_{ij}$ - минор матрицы А.

2. Транспонируем матрицу алгебраических дополнений, чтобы получить матрицу $C_T$ .

3. Вычислим обратную матрицу по формуле:

$$A^{-1} = \frac{1}{|A|} \cdot C_T (\operatorname{mod} m)$$

Вычисления:

Для матрицы 2 × 2:

Ключ: $M = \begin{bmatrix} 5 & 1 \\ 1 & 8 \end{bmatrix}$

Определитель: det(M) = 5 · 8 − 1 · 1 = 39

Обратная матрица:

$$M^{-1} = \frac{1}{39} \begin{bmatrix} 8 & -1 \\ -1 & 5 \end{bmatrix} (mod\ 43)$$

Находим обратный элемент к 39 по модулю 43. Это число 32.

Теперь умножаем матрицу на 32:

$$M^{-1} = \begin{bmatrix} 8 * 32 & -1 * 32 \\ -1 * 32 & 5 * 32 \end{bmatrix} (mod\ 43)$$

$$M^{-1} = \begin{bmatrix} 256 & -32 \\ -32 & 160 \end{bmatrix} (mod\ 43)$$

$$M^{-1} = \begin{bmatrix} 41 & 11 \\ 11 & 31 \end{bmatrix}$$

Для матрицы 3 × 3:

Ключ:

$$M = \begin{bmatrix} 1 & 4 & 5 \\ 2 & 2 & 4 \\ 3 & 1 & 2 \end{bmatrix}$$

Аналогичные вычисления для матрицы 3 × 3:

$$M^{-1} = \begin{bmatrix} 0 & 32 & 32 \\ 15 & 24 & 22 \\ 14 & 26 & 21 \end{bmatrix}$$

Для матрицы 4 × 4:

Ключ:

$$M = \begin{bmatrix} 1 & 1 & 2 & 2 \\ 2 & 1 & 2 & 3 \\ 4 & 2 & 1 & 7 \\ 3 & 5 & 4 & 1 \end{bmatrix}$$

Аналогичные вычисления для матрицы 4 × 4:

$$M^{-1} = \begin{bmatrix} 3 & 16 & 38 & 24 \\ 23 & 37 & 21 & 40 \\ 13 & 10 & 30 & 35 \\ 39 & 28 & 5 & 19 \end{bmatrix}$$

Преобразуем слова в индексы алфавита:

2 × 2: "г8куогб31мбк" → [3, 41], [10, 19], [14, 3], [1, 36], [34, 12], [1, 10]

3 × 3: "ы7зтяряы45и " → [27, 40, 7], [19, 31, 16], [14, 31, 25], [37, 38, 32]

4 × 4: "9щуце9вп76ц0" → [42, 25, 19, 22], [5, 42, 2, 15], [40, 39, 22, 33]

После расшифровки получим следующие сообщения:

2 × 2 : [15, 14], [17, 11], [5, 32], [7, 9], [21, 15], [22, 20] →"после зйхпцф"

3 × 3 : [4, 27, 40], [8, 35, 34], [5, 18, 37], [5, 20, 41] → "ды7и21ет4еф8"

4 × 4 : [20, 41, 4, 5], [38, 15, 39, 33], [13, 41, 4, 4] → "9щуце9вп76ц0"

Видно, что при замене 3 символов не зависимо от их взаимного расположения друг к другу смысл сообщения из 12 символов теряется.

**1.2 Задание 2. Взлом шифра Хилла.**

У нас на руках есть два зашифрованных сообщения, в которых использовался шифр Хилла с одним и тем же ключом, который нам неизвестен. Но у нас есть расшифровка (оригинал) одного из этих сообщений.

Нужно найти способ расшифровать второе сообщение.

• Используем алфавит, который мы составили в предыдущем задании.

абвгдежзийклмнопрстуфхцчшщъыьэюя 0123456789

Выберем ключ размера 2 × 2.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

• Возьмем два различных сообщения из 12 символов и зашифруем их.

"ьшчдиг6птля4" "апъщзх  ахзу"

• "Забудем" одно из исходных сообщений. Имея на руках два зашифрованных сообщения и один оригинал, найдём способ расшифровать второе сообщение.

Для одного из сообщений расшифровкой будет:

"люблю читать"

• Расшифруем второе.

1. Текст → векторы

" ьшчдиг6птля4" → [28, 24] [23, 4] [8, 3] [39, 15] [18, 11] [31, 37].

" апъщзх  ахзу" → [0, 15] [26, 25] [7, 21] [32, 32] [0, 21] [7, 19].

" люблю читать" → [11, 30] [1, 11] [30, 32] [23, 8] [18, 0] [18, 28].

2. Для расшифрованного сообщения: возьмём 2 вектора, сложим матрицу 2x2 и найдём к ней обратную по модулю (модуль равен количеству элементов в алфавите - 43)

Для матрицы $\begin{bmatrix} 11 & 30 \\ 1 & 11 \end{bmatrix}$ обратной по модулю будет: $\begin{bmatrix} 28 & 37 \\ 17 & 28 \end{bmatrix}$.

3. Для каждого зашифрованного сообщения возьмём 2 вектора, сложим матрицу 2 × 2

и домножим её на обратную по модулю для расшифрованного сообщения. Затем

найдём mod полученной матрицы. Получим два ключа:

$$\begin{bmatrix} 12 & 40 \\ 28 & 8 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Теперь нам остаётся расшифровать каждое сообщение, как мы это делали в первом задании, используя сначала первый, а потом второй ключ.

После преобразования имеем:

**Первый ключ**

"ьшчдиг6птля4" → "4шг038зпжтт0"

" апъщзх  ахзу " → "лбюлязбцшк8д"

**Второй ключ**

" ьшчдиг6птля4" → "люблю читать"

" апъщзх  ахзу" → "порезал хлеб"

Видим, что правильный результат даёт только второй ключ. Именно с помощью него мы

получили расшифровку второго сообщения – "порезал хлеб".

1.3 Задание 3. Код Хэмминга.

• Возьмем русский алфавит из 32 букв и сопоставим каждой букве пятибитовый двоичный

номер (от 00000 до 11111).

| Буква | Код | Буква | Код | Буква | Код | Буква | Код |
|---|---|---|---|---|---|---|---|
| А | 00000 | И | 01000 | Р | 10000 | Ш | 11000 |
| Б | 00001 | Й | 01001 | С | 10001 | Щ | 11001 |
| В | 00010 | К | 01010 | Т | 10010 | Ъ | 11010 |
| Г | 00011 | Л | 01011 | У | 10011 | Ы | 11011 |
| Д | 00100 | М | 01100 | Ф | 10100 | Ь | 11100 |
| Е | 00101 | Н | 01101 | Х | 10101 | Э | 11101 |
| Ж | 00110 | О | 01110 | Ц | 10110 | Ю | 11110 |
| З | 00111 | П | 01111 | Ч | 10111 | Я | 11111 |

Таблица 1: Сопоставление буквам двоичных кодов

Придумаем слово из 4 букв, например, слово НОГА.

– Н: 01101

– О: 01110

– Г: 00011

– А: 00000

• Двоичный код для этого слова будет таким: 01101 01110 00011 00000

• Разберёмся в том, как работает код Хэмминга (7,4). Для этого составим порождающую матрицу G и проверочную матрицу H.

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

• Объяснение выбора матриц G и H

Матрица G кодирует 4-битные сообщения в 7-битные кодовые слова. Первые 4 столбца образуют единичную матрицу, что гарантирует включение исходного сообщения в кодовое слово. Оставшиеся 3 столбца добавляют избыточные биты, обеспечивая возможность исправления ошибок. Матрица H отвечает за проверку целостности данных. Она построена так, чтобы каждый возможный синдром был уникальным для каждого возможного положения ошибки. Образ матрицы G совпадает с ядром матрицы H, что гарантирует корректность алгоритма.

• Кодирование слова с использованием матрицы G

Разобьём 20-битное сообщение на 4-битные блоки: 0110, 1011, 1000, 0110, 0000. Кодируем каждый блок с помощью матрицы G:

Блок 1: 0110 → 1100110

Блок 2: 1011 → 0110011

Блок 3: 1000 → 1110000

Блок 4: 0110 → 1100110

Блок 5: 0000 → 0000000

Соединяем кодированные блоки в одно сообщение:

1100110 0110011 1110000 1100110 0000000

• Сымитируем вредоносное вмешательство в закодированное сообщение. Заменим на противоположный:

– 1 бит: 1100110 → 1100010 (бит на позиции 5).

– 2 бита: 0110011 → 0101011 (биты на позициях 3 и 4).

– 3 бита: 1110000 → 1101100 (биты на позициях 3, 4 и 5).

– 4 бита: 1100110 → 1011111 (биты на позициях 2, 3, 4 и 7).

– 5 бит: 0000000 → 1111100 (биты на позициях 1, 2, 3, 4 и 5).

• Декодируем каждое из "испорченных" сообщений, используя матрицу H, для поиска и исправления ошибочных битов.

– Испорченный блок 1: 1100010

Вычисляем синдром:

$$s_1 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \equiv \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Синдром $s_1$ = 101 указывает на ошибку во 5-м бите. Исправляем ошибку:

1100010 → 1100110

– Испорченный блок 2: 0101011

Вычисляем синдром:

$$s_2 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 3 \end{bmatrix} \equiv \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Синдром $s_2$ = 111 указывает на ошибку в 7-м бите. Исправляем ошибку:

0101011→ 0101010

– Испорченный блок 3: 1101100

Вычисляем синдром:

$$s_3 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} \equiv \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Синдром $s_3$ = 010 указывает на ошибку в 2-м бите.

Исправляем ошибку:

1101100→ 1001100

– Испорченный блок 4: 1011111

Вычисляем синдром:

$$s_4 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \\ 4 \end{bmatrix} \equiv \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Синдром $s_4$ = 010 указывает на ошибку в 2-м бите. Исправляем ошибку:

1011111 → 1111111

– Испорченный блок 5: 1111100

Вычисляем синдром:

$$s_5 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 2 \end{bmatrix} \equiv \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Синдром $s_5$ = 100 указывает на ошибку в 6-м бите. Исправляем ошибку:

1111100 → 1111110

• Переведём каждый из полученных результатов в слово из 4 букв.

После исправления ошибок возвращаемся к 4-битным блокам с помощью матрицы R, получаем:

– 1100110 → 0110

– 0101010 → 0010

– 1001100 → 1010

– 1111111 → 1111

– 1111110 → 1110

• Составим последовательность и выделим буквы:

01100 01010 10111 11110

– 01100 → "М"

– 01010 → "К"

– 10111 → "Ч"

– 11110 → "Ю"

В результате расшифровки и исправления ошибок мы получили слово

МКЧЮ

Это доказывает, что с помощью кода Хэмминга можно исправить только ошибки в одиночных битах.

2 Выводы и анализ результатов работы

В ходе выполнения данной лабораторной работы мы изучили кодирование и декодирование данных с помощью кода Хэмминга (7,4). Мы научились корректировать ошибки, возникшие в результате "вредоносного" вмешательства, и убедились, что код Хэмминга эффективно восстанавливает исходное сообщение.