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»

Студенты:
Бахтаиров Р.А. группа R3243
Сайфуллин Д.Р группа R3243
Симонов И.А. группа R3236
Холухина Д.Е. группа R3240

Проверил:
Догадин Егор Витальевич, ассистент

Санкт-Петербург
2024

Содержание

1 Введение

В данной лабораторной работе исследуются методы использования линейной алгебры в криптографии и теории кодирования. Основное внимание уделяется шифру Хилла, который использует матричные преобразования для шифрования сообщений, и коду Хэмминга, применяемому для обнаружения и исправления ошибок. Цель работы — продемонстрировать, как можно использовать эти методы на практике и оценить их эффективность.

2 Шифр Хилла

2.1 Подготовка алфавита

Мы используем русский алфавит. Исключим из него символы “Ь”, “Ъ”, “Ы”, и дополним пробелом. Так получаем набор из n=31 символа. Алфавит пронумерован от 0 до n-1.

2.2 Сообщение для шифрования

Выбранное сообщение: `ПРИВЕТ Я РОН`.

2.3 Матрицы-ключи

Выберем следующие матрицы-ключи:

$$A = \begin{pmatrix} 5 & 17 \\ 28 & 4 \end{pmatrix} \qquad B = \begin{pmatrix} 25 & 20 & 23 \\ 7 & 9 & 13 \\ 11 & 2 & 21 \end{pmatrix} \qquad C = \begin{pmatrix} 5 & 3 & 7 & 14 \\ 6 & 2 & 1 & 3 \\ 5 & 4 & 2 & 8 \\ 9 & 2 & 10 & 6 \end{pmatrix}$$

$$\det(A) = -456 \qquad \det(B) = 2040 \qquad \det(C) = -756$$

Определители всех матриц-ключей не имеют общих делителей с числом символов в алфавите, т. к. n - простое число.

2.4 Шифрование

Для каждого из ключей зашифруем сообщение методом Хилла. Для этого представим сообщение как набор чисел нашего алфавита и получим M = [16, 17, 9, 2, 5, 19, 30, 29, 30, 17, 15, 14]. Для разных матриц-ключей понадобится разное представление этого набора чисел (вектора), чтобы произведение было возможным и дало матрицу той же размерности.

$$M_A = AM \pmod{31} = \begin{pmatrix} 5 & 17 \\ 28 & 4 \end{pmatrix} \begin{pmatrix} 16 & 17 & 9 & 2 & 5 & 19 \\ 30 & 29 & 30 & 17 & 15 & 14 \end{pmatrix} \pmod{31} =$$

$$= \begin{pmatrix} 590 & 578 & 555 & 299 & 280 & 333 \\ 568 & 592 & 372 & 124 & 200 & 588 \end{pmatrix} \pmod{31} = \begin{pmatrix} 1 & 20 & 28 & 20 & 1 & 23 \\ 10 & 3 & 0 & 0 & 14 & 30 \end{pmatrix}$$

После шифрования получаем зашифрованное сообщение: `БУЮУБЦЙГААН□`

$$M_B = BM \pmod{31} = \begin{pmatrix} 25 & 20 & 23 \\ 7 & 9 & 13 \\ 11 & 2 & 21 \end{pmatrix} \begin{pmatrix} 16 & 17 & 9 & 2 \\ 5 & 19 & 30 & 29 \\ 30 & 17 & 15 & 14 \end{pmatrix} \pmod{31} =$$

$$= \begin{pmatrix} 1190 & 1196 & 1170 & 952 \\ 547 & 511 & 528 & 457 \\ 816 & 582 & 474 & 374 \end{pmatrix} \pmod{31} = \begin{pmatrix} 12 & 18 & 23 & 22 \\ 20 & 15 & 1 & 23 \\ 10 & 24 & 9 & 2 \end{pmatrix}$$

После шифрования получаем зашифрованное сообщение: `ЛСЦХУОБЦЙЧИВ`

$$M_C = CM \pmod{31} = \begin{pmatrix} 5 & 3 & 7 & 14 \\ 6 & 2 & 1 & 3 \\ 5 & 4 & 2 & 8 \\ 9 & 2 & 10 & 6 \end{pmatrix} \begin{pmatrix} 16 & 17 & 9 \\ 2 & 5 & 19 \\ 30 & 29 & 30 \\ 17 & 15 & 14 \end{pmatrix} \pmod{31} =$$

$$= \begin{pmatrix} 534 & 513 & 508 \\ 181 & 186 & 164 \\ 284 & 283 & 293 \\ 550 & 543 & 503 \end{pmatrix} \pmod{31} = \begin{pmatrix} 7 & 17 & 12 \\ 26 & 0 & 9 \\ 5 & 4 & 14 \\ 23 & 16 & 7 \end{pmatrix}$$

После шифрования получаем зашифрованное сообщение: `ЖРЛЩАИЕДНЦПЖ`

2.5 Вредоносное вмешательство

Симулируем вредоносное вмешательство, заменив три символа в каждом из зашифрованных сообщений на другие:

– Для сообщения `БУЮУБЦЙГААН□` $\rightarrow$ `БУЮУЛЦЙКААНХ` $\rightarrow M_A = \begin{pmatrix} 1 & 20 & 28 & 20 & 12 & 23 \\ 10 & 11 & 0 & 0 & 14 & 22 \end{pmatrix}$

– Для сообщения `ЛСЦХУОБЦЙЧИВ` $\rightarrow$ `ЛПЦХУОЕЦЙЧИШ` $\rightarrow M_B = \begin{pmatrix} 12 & 16 & 23 & 22 \\ 20 & 15 & 5 & 23 \\ 10 & 24 & 9 & 25 \end{pmatrix}$

– Для сообщения `ЖРЛЩАИЕДНЦПЖ` $\rightarrow$ `ЖРГЩЙИОДНЦПЖ` $\rightarrow M_C = \begin{pmatrix} 7 & 17 & 3 \\ 26 & 10 & 9 \\ 15 & 4 & 14 \\ 23 & 16 & 7 \end{pmatrix}$

2.6 Расшифровка

Теперь расшифруем взломанные сообщения, но для начала нужно найти обратные матрицы:

$$A^{-1} \pmod{31} = \frac{1}{\det(A)} \begin{pmatrix} 4 & -17 \\ -28 & 5 \end{pmatrix} \pmod{31} = 7 \begin{pmatrix} 4 & -17 \\ -28 & 5 \end{pmatrix} \pmod{31} =$$

$$= \begin{pmatrix} 28 & -119 \\ -196 & 35 \end{pmatrix} \pmod{31} = \begin{pmatrix} 28 & 5 \\ 21 & 4 \end{pmatrix}$$

$$B^{-1} \pmod{31} = \frac{1}{\det(B)} \begin{pmatrix} 163 & -374 & 53 \\ -4 & 272 & -164 \\ -85 & 170 & 85 \end{pmatrix} \pmod{31} =$$

$$= 5 \begin{pmatrix} 163 & -374 & 53 \\ -4 & 272 & -164 \\ -85 & 170 & 85 \end{pmatrix} \pmod{31} = \begin{pmatrix} 9 & 21 & 17 \\ 20 & 27 & 17 \\ 9 & 13 & 22 \end{pmatrix}$$

$$C^{-1} \pmod{31} = \frac{1}{\det(C)} \begin{pmatrix} -36 & -244 & 138 & 22 \\ 270 & 402 & -531 & -123 \\ 54 & 198 & -81 & -117 \\ -126 & -98 & 105 & 77 \end{pmatrix} \pmod{31} =$$

$$= 13 \begin{pmatrix} -36 & -244 & 138 & 22 \\ 270 & 402 & -531 & -123 \\ 54 & 198 & -81 & -117 \\ -126 & -98 & 105 & 77 \end{pmatrix} \pmod{31} = \begin{pmatrix} 28 & 21 & 27 & 7 \\ 7 & 18 & 10 & 13 \\ 20 & 1 & 1 & 29 \\ 5 & 28 & 1 & 9 \end{pmatrix}$$

И, наконец, займёмся расшифровкой, домножив слева матрицы сообщений на матрицы выше:

$$A^{-1}M_A \pmod{31} = \begin{pmatrix} 28 & 5 \\ 21 & 4 \end{pmatrix} \begin{pmatrix} 1 & 20 & 28 & 20 & 12 & 23 \\ 10 & 11 & 0 & 0 & 14 & 22 \end{pmatrix} \pmod{31} =$$

$$= \begin{pmatrix} 78 & 615 & 784 & 560 & 406 & 754 \\ 61 & 464 & 588 & 420 & 308 & 571 \end{pmatrix} \pmod{31} = \begin{pmatrix} 16 & 26 & 9 & 2 & 3 & 10 \\ 30 & 30 & 30 & 17 & 29 & 13 \end{pmatrix}$$

После шифрования получаем зашифрованное сообщение: `ПЩИВГЙ□□□РЯМ`

$$B^{-1}M_B \pmod{31} = \begin{pmatrix} 9 & 21 & 17 \\ 20 & 27 & 17 \\ 9 & 13 & 22 \end{pmatrix} \begin{pmatrix} 12 & 16 & 23 & 22 \\ 20 & 15 & 5 & 23 \\ 10 & 24 & 9 & 25 \end{pmatrix} \pmod{31} =$$

$$= \begin{pmatrix} 698 & 867 & 465 & 1106 \\ 950 & 1133 & 748 & 1486 \\ 588 & 867 & 470 & 1047 \end{pmatrix} \pmod{31} = \begin{pmatrix} 16 & 30 & 0 & 21 \\ 20 & 17 & 4 & 29 \\ 30 & 30 & 5 & 24 \end{pmatrix}$$

После шифрования получаем зашифрованное сообщение: `П□АФУДЯ□□ЕЧ`

$$C^{-1}M_C \pmod{31} = \begin{pmatrix} 28 & 21 & 27 & 7 \\ 7 & 18 & 10 & 13 \\ 20 & 1 & 1 & 29 \\ 5 & 28 & 1 & 9 \end{pmatrix} \begin{pmatrix} 7 & 17 & 3 \\ 26 & 10 & 9 \\ 15 & 4 & 14 \\ 23 & 16 & 7 \end{pmatrix} \pmod{31} =$$

$$= \begin{pmatrix} 1308 & 906 & 700 \\ 966 & 547 & 414 \\ 848 & 818 & 286 \\ 985 & 513 & 344 \end{pmatrix} \pmod{31} = \begin{pmatrix} 6 & 7 & 18 \\ 5 & 20 & 11 \\ 11 & 12 & 7 \\ 24 & 17 & 3 \end{pmatrix}$$

После шифрования получаем зашифрованное сообщение: `ЁЖСЕУККЛЖЧРГ`

2.7 Вывод

При замене хотя бы одной из букв в зашифрованном сообщении теряется всё исходное сообщение, так как во время матричного умножения буквы связываются между собой.

3 Взлом шифра Хилла

3.1 Подготовка

В этом задании будем использовать алфавит и сообщение из 1-го задания, второе сообщение будет - `АВТОМАТ□ШИФР`

3.2 Матрица-ключ

Для этого задания была написана простая программа, которая случайно генерирует ключ, скрывает его до конца расшифровки и шифрует сообщения. Результат работы программы:

$$M = [16, 17, 9, 2, 5, 19, 30, 29, 30, 17, 15, 14] \rightarrow M_K = [3, 15, 1, 14, 25, 23, 10, 12, 29, 3, 22, 2]$$

$$N = [0, 2, 19, 15, 13, 0, 19, 30, 25, 9, 21, 17] \rightarrow N_K = [21, 30, 27, 14, 19, 9, 30, 17, 21, 1, 3, 4]$$

Из уравнения $M_K = KM$, где матрица К является нашей матрицей-ключом мы можем составить:

$$\begin{pmatrix} 3 & 15 & 1 & 14 & 25 & 23 \\ 10 & 12 & 29 & 3 & 22 & 2 \end{pmatrix} = \begin{pmatrix} k_1 & k_2 \\ k_3 & k_4 \end{pmatrix} \begin{pmatrix} 16 & 17 & 9 & 2 & 5 & 19 \\ 30 & 29 & 30 & 17 & 15 & 14 \end{pmatrix} =$$

$$= \begin{pmatrix} 16*k_1+30*k_2 & 17*k_1+29*k_2 & 9*k_1+30*k_2 & 2*k_1+ \\ +17*k_2 & 5*k_1+15*k_2 & 19*k_1+14*k_2 & \\ 16*k_3+30*k_4 & 17*k_3+29*k_4 & 9*k_3+30*k_4 & 2*k_3+ \\ +17*k_4 & 5*k_3+15*k_4 & 19*k_3+14*k_4 & \end{pmatrix}$$

Решим данное уравнение с помощью простой программы и найдем матрицу-ключ:

$$K = \begin{pmatrix} 18 & 6 \\ 15 & 13 \end{pmatrix}$$
$$\det(K) = 144$$

3.3 Проверка

Для проверки возьмем обратную матрицу от матрицы K, умножим ее на зашифрованную матрицу, чтобы получить исходное сообщение:

$$K^{-1} \pmod{31} = \frac{1}{\det(K)} \begin{pmatrix} 13 & -6 \\ -15 & 18 \end{pmatrix} \pmod{31} = 14 \begin{pmatrix} 13 & -6 \\ -15 & 18 \end{pmatrix} \pmod{31} =$$

$$= \begin{pmatrix} 182 & -84 \\ -210 & 252 \end{pmatrix} \pmod{31} = \begin{pmatrix} 27 & 9 \\ 7 & 4 \end{pmatrix}$$

$$K^{-1} N_K \pmod{31} = \begin{pmatrix} 27 & 9 \\ 7 & 4 \end{pmatrix} \begin{pmatrix} 21 & 30 & 27 & 14 & 19 & 9 \\ 30 & 17 & 21 & 1 & 3 & 4 \end{pmatrix} \pmod{31} =$$

$$= \begin{pmatrix} 837 & 963 & 918 & 387 & 540 & 279 \\ 267 & 278 & 273 & 102 & 145 & 79 \end{pmatrix} \pmod{31} = \begin{pmatrix} 0 & 2 & 19 & 15 & 13 & 0 \\ 19 & 30 & 25 & 9 & 21 & 17 \end{pmatrix}$$

Проверим побуквенно:

$0 = \text{А}, 2 = \text{В}, 19 = \text{Т}, 15 = \text{О}, 13 = \text{М}, 0 = \text{А}, 19 = \text{Т}, 30 = \square, 25 = \text{Ш}, 9 = \text{И}, 21 = \text{Ф}, 17 = \text{Р}$

И получаем исходную фразу `АВТОМАТ□ШИФР`.

3.4 Вывод

Не имея на руках ключа, но имея оригинал и зашифрованное сообщение, можно выявить линейную зависимость шифра, получить матрицу-ключ и взломать шифр, чтобы в дальнейшем расшифровать любой зашифрованное сообщение.

4 Код Хэмминга

4.1 Подготовка алфавита

Первоначально каждому символу русского алфавита присвоим уникальный 5-битный двоичный код. Таким образом, для всех 32 букв русского алфавита формируется двоичная таблица:

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

Таблица 1: Таблица русского алфавита с 5-битным двоичным кодом

4.2 Матрица генератор кода G и матрица проверки H

Матрица G: Эта матрица используется для кодирования сообщения. Она состоит из 4 информационных бит и 3 проверочных бит. Общая структура сообщения после кодирования выглядит как 7-битная строка. Матрица G имеет размер 4x7 и применяется для умножения на исходное сообщение для получения кодового слова.

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

Матрица H: Эта матрица используется для проверки и исправления ошибок. Размер матрицы H — 3x7. Она позволяет обнаруживать ошибочные биты в переданном коде. Синдром, полученный после умножения матрицы H на кодовое слово, помогает идентифицировать ошибки и их местоположение.

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

4.3 Сообщение для шифрования

Выберем слово из 4 букв, например, `ВИТЯ`. Для каждой буквы этого слова с помощью таблицы алфавита найдем её 5-битный двоичный код. В результате слово будет представлено как строка из 20 двоичных символов.

$$\text{В} = 00010, \quad \text{И} = 01000, \quad \text{Т} = 10010, \quad \text{Я} = 11111$$

Получившееся двоичное представление слова `ВИТЯ`:

$$00010\ 01000\ 10010\ 11111$$

Для удобства представим наше слово в матричном виде. Для этого разделим наше сообщение на 5 групп по 4 бита и составим матрицу нашего сообщения:

$$M = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

4.4 Шифрование

Зашифруем сообщение с использованием порождающей матрицы G:

$$G^T M \pmod 2 = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix} \pmod 2 =$$

$$= \begin{pmatrix} 1 & 0 & 0 & 2 & 3 \\ 1 & 1 & 1 & 1 & 3 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 2 & 3 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix} \pmod 2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

4.5 Вредоносное вмешательство

Симулируем вредоносное вмешательство. Для этого, в закодированное сообщение намеренно вносим ошибки:

- Инверсия одного бита: $\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & \mathbf{0} & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix}$

- Инверсия двух бит: $\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & \mathbf{0} & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & \mathbf{1} & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix}$

- Инверсия трёх бит: $\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & \mathbf{0} & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ \mathbf{1} & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & \mathbf{1} & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix}$

- Инверсия четырёх бит: $\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & \mathbf{0} & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & \mathbf{1} & 1 \\ \mathbf{1} & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & \mathbf{1} & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix}$

4.6 Декодирование и исправление ошибок

Для исправления ошибок используем контрольную матрица H. Используя матрицу H, находим синдром ошибки:

$$H \cdot \text{Кодовое слово} = \text{Синдром}$$

Если синдром ненулевой, ошибка обнаружена, и по его значению можно определить, какой бит ошибочный.

Исправляем 1 «плохой» бит

$$S_1 = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix} \pmod 2 =$$

$$\begin{pmatrix} 2 & 0 & 0 & 2 & 4 \\ 2 & 1 & 2 & 2 & 4 \\ 2 & 2 & 2 & 2 & 4 \end{pmatrix} \pmod 2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Синдром S_1 - ненулевой, следовательно мы имеем ошибку во втором столбце нашего исходного сообщения. Чтобы найти порядковый номер ошибочного бита мы берем нужный столбец синдрома и находим его порядковый номер в матрице $H = 2$. Исправляем «плохой» бит во втором столбце на втором месте и расшифровываем сообщение.
Для расшифровки необходимо извлечь информационные биты. В коде Хэмминга (7,4) они находятся на позициях 2, 4-6 в кодовом слове. Тогда,

$$1101001 \to 0001$$

$$0101010 \to 0010$$

$$0101010 \to 0010$$

$$0100101 \to 0101$$

$$1111111 \to 1111$$

Наше сообщение:

$$00010\ 01000\ 10010\ 11111 \to \texttt{ВИТЯ}$$

Исправляем 2 «плохих» бита

$$S_2 = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix} \pmod 2 =$$

$$\begin{pmatrix} 2 & 0 & 0 & 2 & 4 \\ 2 & 1 & 2 & 3 & 4 \\ 2 & 2 & 2 & 3 & 4 \end{pmatrix} \pmod 2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Синдром S_2 - ненулевой, аналогично находим «плохие» биты - это (2, 2) и (6, 4). Исправляем ошибку и расшифровываем сообщение:

$$1101001 \to 0001$$

$$0101010 \to 0010$$

$$0101010 \to 0010$$

$$0100101 \to 0101$$

$$1111111 \to 1111$$

Наше сообщение:

$$00010\ 01000\ 10010\ 11111 \to \texttt{ВИТЯ}$$

Исправляем 3 «плохих» бита

$$S_3 = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix} \pmod 2 =$$

$$\begin{pmatrix} 3 & 0 & 0 & 2 & 4 \\ 2 & 1 & 2 & 3 & 4 \\ 3 & 2 & 2 & 3 & 4 \end{pmatrix} \pmod 2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Синдром S_3 - ненулевой. Ошибки находятся в координатах (5, 1), (2, 2), (6, 4). Исправим их и расшифруем наше сообщение

$$1101001 \to 0001$$

$$0001010 \to 0010$$

$$0101010 \to 0010$$

$$0100101 \to 0101$$

$$1111111 \to 1111$$

Наше сообщение:

$$00010\ 01000\ 10010\ 11111 \to \texttt{ВИТЯ}$$

Мы можем заметить, что Код Хэмминга отлично справляется с единычными ошибками. Это связано с тем, что в каждом столбце нашей матрицы M шифруется по одной букве нашего алфавита. Но что будет, если сделать 2 ошибки в одной зашифрованной букве? Проверим это в следующем пункте.

Исправляем 4 «плохих» бита

$$S_4 = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix} \pmod 2 =$$

$$\begin{pmatrix} 3 & 0 & 0 & 2 & 4 \\ 2 & 1 & 2 & 3 & 4 \\ 3 & 2 & 2 & 4 & 4 \end{pmatrix} \pmod 2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Синдром S_4 - ненулевой, следовательно мы имеем ошибки в первом, втором и четвертом столбцах. Попробуем исправить ошибки по известному нам алгоритму и расшифровать сообщение. Ошибки находятся в координатах (5, 1), (2, 2), (2, 4).

$$1101001 \to 0001$$

$$0001010 \to 0010$$

$$0101010 \to 0010$$

$$0001111 \to 0111$$

$$1111111 \to 1111$$

Наше сообщение:

$$00010\ 01000\ 10011\ 11111 \to \texttt{ВИУЯ}$$

Как мы видим сообщение расшифровалось неверно, следовательно можно сделать вывод о том, что Код Хэмминга не работает при возникновении более двух ошибок в одном слове.

4.7 Вывод

В данной задании был рассмотрен процесс кодирования и декодирования сообщения с использованием кода Хэмминга (7,4). Были продемонстрированы этапы работы с порождающей и с проверочной матрицами, введение ошибок в исходное сообщение и их исправление. Код Хэмминга показал эффективность в исправлении одиночных ошибок, но не справился с случаем, когда возникло более 1 ошибки в одной из букв.

5 Заключение

В ходе лабораторной работы было продемонстрировано, как матричные преобразования и коды Хэмминга могут применяться для решения задач шифрования и обнаружения ошибок. Шифр Хилла, благодаря своей основе на линейной алгебре, является простым, но эффективным методом шифрования. Код Хэмминга показал свою способность исправлять ошибки, что делает его полезным в системах передачи данных. Полученные навыки могут быть полезны в области компьютерной безопасности и защиты информации.