Задание 5

И. Герасимов

1 Использование программы

```
python3.8 main.py [-h] [-g -n [N] -p [PROBABILITY]]
[-c -f [FILE]] [-d -f [FILE] -y [Y]]
```

1.1 Генерация кода (указан флаг -g)

- `-n` — желаемая максимальная длина блока сообщения, передаваемая по каналу связи;
- `-p` — вероятность ошибки в канале связи для двоичного симметричного канала.

Если указанные параметры не позволяют сформировать код, удовлетворяющий прямой теореме Шеннона, то будет выполняться понижение n, пока не требования не будут соблюдены или не будет исчерпано множество возможных n. Во втором случае работа программы закончиться с выводом того, что невозможно найти подходящие параметры относительно прямой теоремы Шеннона.

Будут созданы 2 файла (в конце каждого файла указывается индекс, чтобы избежать перезаписываний):

- `code` — информация для кодера;
- `decode` — информация для декодера.

Информация, требуемая в соответствии с заданием указывается в файле для декодера.

1.2 Режим кодирования (указан -c)

- `-f` - файл для кодера (например, первый файл `code`, получаемый при генерации кода);

Будет создан следующий файл (в конце каждого файла указывается индекс, чтобы избежать перезаписываний):

- `code_result` — результат кодирования.

В конце файла также указывается длина паддинга для достижения кратности длины сообщения длине кодируемого блока.

1.3 Режим декодирования (указан `-d`)

- `-f` - файл для декодера (например, второй файл `decode`, получаемый при генерации кода);
- `-y` - файл с кодом, пришедшим из канала связи;

Для каждого кода выводится процесс декодирования и итоговый результат в байтовом представлении и представлении UTF-8. Если представление результата декодирования в кодировке UTF-8 не выполнима, то будет выведено уведомление.

2 Описание работы генерации кода

1. Определяется минимальное m и сответствующее s: $n = (2^m - 1)/s$;
2. Строятся циклотомические классы относительно элемента α^s;
3. Определяются всевозможные значения k относительно циклотомических классов. Количество элементов в классе определяет степень соответствующего минимального многочлена. Поскольку порождающий многочлен есть роизведение минимальных, его степень определяется степенями полиномов. Степень порождающего мнгочлена определяет k.
4. Для каждого возможного k определяется b такое, что δ максимально;
5. Берется наибольшее k такое, что выполняется прямая теорема Шеннона относительно заданного p;
6. Определяется многочлен, соответсвующий α через многочлен $x^m + 1$ (плюс, так как в $GF(2)$);
7. Строются многочлены, соответствующие циклотомическим классам;
8. Строится порождающий многочлен;
9. Строится порождающая матрица в соответствии с порождающим многочленом;
10. Строится проверочная матрица (в итоговой версии скрипта не используется);
11. Формируются файлы, описанные в 1.1.

Замечание 1: Поскольку порождающий многочлен определяется как наименьшее общее кратное минимальных многочленов, его степень вообще говоря, не будет равна сумме степеней минимальной. Однако минимальные многочлены являются приведенными многочленами и было решено использовать такой подход для k.

3 Описание работы кодера

При запуске будет инициирован запрос на ввод кодируемого сообщения.

Для каждого блока сообщения выполняется умножение $x = mG$ и накладывается ошибка $y = x \oplus e$. Ошибка генерируется в соответствии с указанным распределением.

Если длина сообщения не кратна длине кодируемого блока, выполняется паддинг (к концу сообщения). Длина паддинга будет указана в конце файла.

3.1 формат файла кодера

1. Список кодов y;
2. длина паддинга.

4 Описание работы декодера

1. Выполняется загрузка параметров;
2. Для каждого кода y вычисляется синдром;
3. По синдрому выполняется алгоритм Берлекэмпа-Месси для вычисления присоединенного полинома $\sigma(x)$ регитра сдвига;
4. Перебираются корни $\sigma(x)$;
5. По полученным корням снимается ошибка e;
6. Синдром полученного кода x вычисляется снова. Если синдром не равен нулевому вектору, то выдается уведомление о неуспехе декодирования;
7. Далее выполняется результат декодирования x посредством деления на попрождающий полином $g(x)$;
8. Выводится результат в байтовом представлении и в кодировке UTF-8. Если представить сообщение в UTF-8 не получается, будет выведено уведомление.

5 Примеры

5.1 Примитивный код БЧХ в узком смысле при $n = 15, p = 0.01$ (код Хэмминга)

Файл декодера должен выглядеть следующим образом:

```
t:
1
n:
15
k:
11
probability of the error in channel:
0.01
alpha polynomial:
11001
b:
1
g polynomial:
11001
```

5.2 Примитивный код БЧХ в не узком смысле $n = 15, p = 0.2$

Файл декодера должен выглядеть следующим образом ($b = 6$):

```
t:
2
n:
15
k:
3
probability of the error in channel:
0.2
alpha polynomial:
11001
b:
6
g polynomial:
1001001001001
```

5.3 Примитивный код БЧХ в узком смысле $n = 5, p = 0.1$

Файл декодера должен выглядеть следующим образом ($m = 4, s = 3$):

```
t:
2
n:
5
k:
1
```

```
probability of the error in channel:
0.1
alpha polynomial:
11001
b:
1
g polynomial:
11111
```