ИМИТАЦИОННАЯ МОДЕЛЬ ПРОГРАММНОГО СРЕДСТВА ДЛЯ РАБОТЫ С СОВЕРШЕННЫМИ КОДАМИ НА ОСНОВЕ КОДА ХЭММИНГА И КОДА ГОЛЕЯ.

Егоров Никита Валентинович

Аннотация. В данной статье проводится аналитический обзор помехоустойчивых кодов, а именно рассматриваются - совершенные коды. Совершенные коды могут быть использованы для кодирования и декодирования информации, так как обладают всеми желанными свойствами, а именно: устойчивость к помехам и простота реализации. В заключении разработана имитационная модель кодирования и декодирования информационного слова.

Annotation. This article presents an analytical review of noise-tolerant codes, namely, the perfect codes are considered. Ideal codes can be used to encode and decode information, as they have all the desired properties, namely, resistance to interference and ease of implementation. In conclusion, a simulation model of encoding and decoding the information word is developed.

Ключевые слова: Код Хэмминга, код Голея, коды повторения, помехоустойчивость, кодирование, декодирование, побитовый сдвиг, Совершенные коды.

Key words: Hamming code, Golay code, Repetition codes, Noise Immunity, Encoding, Decoding, Bitwise shift, Perfect codes.

Введение

Помехоустойчивые коды и помехоустойчивое кодирование является неотъемлемой частью процесса передачи информации. Ещё в середине 20 века передовики научной дисциплины были уверены, что будущее будет за каналами связи, которые могут в условиях множественных источников электромагнитных волн не только доставлять какие-либо данные, но и доставлять их в целости и сохранности.

Именно так и появились помехоустойчивые коды и помехоустойчивое кодирование которое является неотъемлемой частью процесса передачи информации. Одним из главных преимуществ является Главным преимуществом помехоустойчивых кодов является беспрецедентная надежность даже при минимальной мощности принимаемого сигнала.

Одним из примеров помехоустойчивых кодов могут стать совершенные коды. Совершенные коды могут быть использованы для кодирования и декодирования информации, так как обладают всеми желанными свойствами, а именно: устойчивость к помехам и простота реализации. [1]

В соответствии в заявленной целью необходимо решить следующие задачи:

- исследовать работу совершенного двоичного кода Голея;
- исследовать работу совершенного двоичного/троичного кода Хэмминга
- разработать имитационную модель приложения для кодирования введенного информационного слова и декодирования кодового слова с ошибкой.

Основная часть

В теории кодирования граница Хэмминга определяет пределы возможных значений параметров произвольного блокового кода, также она называется, как граница сферической упаковки. Коды, достигающие границы Хэмминга, называют совершенными или плотноупакованными.

Совершенный код есть код, для которого сферы некоторого одинаково радиуса вокруг кодовых слов, не пересекаясь, покрывают все пространство.

Свойства совершенного кода:

1. Для совершенного (m,n) -кода, исправляющего все ошибки веса, не большего k, выполняется соотношение $\sum_{i=0}^{k} C_n^i = 2^{n-m}$. Верно и обратное утверждение;
2. Совершенный код, исправляющий все ошибки веса, не большего k, в столбцах таблицы декодирования содержит все слова, отстоящие от кодовых на расстоянии, не большем k. Верно и обратное утверждение;
3. Таблица декодирования совершенного кода, исправляющего все ошибки в не более чем k позициях, имеет в качестве лидеров все строки, содержащие не более k единиц. Верно и обратное утверждение.

Совершенный код - это лучший код, обеспечивающий максимум минимального расстояния между кодовыми словами при минимуме длины кодовых слов. Совершенный код легко декодировать: каждому полученному слову однозначно ставится в соответствие ближайшее кодовое. Чисел m, n и k, $(1 < k < \frac{n-1}{2})$, удовлетворяющих условию совершенности кода очень мало. Но и при подобранных m, n и k совершенный код можно построить только в исключительных случаях.

Если m, n и k не удовлетворяют условию совершенности, то лучший групповой код, который им соответствует называется квазисовершенным. Квазисовершенный код - это код, у которого сферы радиуса t вокруг каждого кодового слова не пересекаются и все слова, не лежащие внутри сфер, лежат на расстоянии t+1

хотя бы от одного кодового слова.

Коды, достигающие границы Хэмминга, называют совершенными. Были открыты следующие типы совершенных кодов: коды Хэмминга и коды Голея. Имеются ещё тривиальные совершенные коды: двоичные коды с повторением нечётной длины[2].

У каждого из совершенных кодов есть свои границы параметров, например:

- д
- н
- коды повторения;
- двоичный код Голея (23,12,7);
- троичный код Голея (11,6,5).

Код Голея

Исследуем двоичный код Голея.

Двоичный код Голея - один из двух связанных друг с другом исправляющих ошибки линейных кодов:

- совершенный двоичный код Голея - совершенный двоичный код с параметрами (23,12,7);
- расширенный двоичный код Голея, получающийся из совершенного добавлением бита контроля чётности и имеющий параметры (24,12,8).

Далее будем рассматривать совершенный двоичный код Голея. Из таблицы биноминальных коэффициентов замечено равенство. Это равенство представляет собой необходимое, но достаточное условие существования совершенного, исправляющего 3-х кратные ошибки (23, 12) кода Голея. Так же из биноминального уравнения выяснилось, что: число точек внутри сферы декодирования равно 2^{11}; всего имеется 2^{12}сфер декодирования; всё пространство 2^{23}точек.

Такой код исправляет 3 ошибки, является совершенным, что означает все сферы некоторого одинакового радиуса вокруг кодовых слов не пересекаясь, перекрывают всё пространство точек, т.е. никаких точек между сферами нет. В зависимости от того, какой неприводимый многочлен используется для построения поля, могут быть получены два двойственных по отношению друг к другу порождающих многочлена g(x) кода Голея: $g(x) = x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1$ и $g(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$.

Чтоб разобраться лучше в принципе работы кода Голея, рассмотрим пример кодирования и декодирования информационных и кодовых слов. Код Голея характеризуется своим постоянным образующим полиномом - g(x)=(110001110101)[3].

– Получим кодовое слово, для этого необходимо информационное слово умножить на образующий полином g(x), т.е. перемножить соответствующие полиномы. Получиться полином с максимальной степенью 22, т.е. слово в 23 бита. При умножении полиномов сложение происходит по модулю 2. Для получения информационного слова нужно кодовое слово разделить на образующий полином, причем остаток от деления будет равен нулю (делим один полином на другой). Если в кодовое слово внести ошибки, то остаток от деления на g(x) уже не будет равен нулю. Задача декодирования заключается в поиске ошибочных бит, при исправлении которых остаток от деления на g(x) будет равен нулю. Далее представлен алгоритм декодирования.

– 1. Необходимо разделить кодовое слово на образующий полином, т.е. находим синдром s(x). Если вес синдрома равен нулю (т.е. g(x) делит кодовое слово без остатка) - это будет означать, что в кодовом слове ошибок нет. Если вес синдрома от 1 до 3 - это значит, что все (1, 2 или 3) ошибки в младших 11 битах кодового слова. Причем, единичные биты в синдроме соответствуют ошибочным битам в кодовом слове. Если вес больше 3, переходим к шагу 2.

– 2. Предполагаем, что в 18 бите ошибка, остальные (0, 1 или 2) ошибки в младших 11 битах. Для проверки предположения находим модифицированный синдром. Чтобы его найти необходимо сложить «нормальный» синдром с остатком от деления x^{17} на g(x). Остаток заранее посчитан: 11011001100 (старшие биты в этом примере всегда слева). Операция сложения эквивалентна побитовой операции xor (исключающее или). Если вес модифицированного синдрома равен 0, 1 или 2 - это значит, что в 18 бите ошибка, а все остальные ошибки в младших 11 битах. Причем, единичные биты в модифицированном синдроме соответствуют ошибочным битам в кодовом слове. Если вес больше 3, переходим к шагу 3.

– 3. Предполагаем, что в 17 бите ошибка, остальные (0, 1 или 2) ошибки в младших 11 битах. Далее аналогично 2 шагу. Остаток от деления x^{18} на g(x) равен 01101100110. Если вес больше 3, переходим к шагу 4.

4. Далее необходимо сделать циклический сдвиг кодового слова в сторону младших разрядов и начинаем с 1 шага. Когда на очередном этапе ошибки будут найдены, необходимо кодовое слово циклически сдвинуть в обратную сторону, на соответствующее количество бит.

Чтобы каждый раз на первом шаге не делить кодовое слово на g(x), существует правило для нахождения нового синдрома, после циклического сдвига кодового слова. Если в младшем разряде синдрома 0, то новый синдром получается простым циклическим сдвигом старого синдрома. Если в младшем разряде синдрома 1, то для нахождения нового синдрома нужно старый синдром сложить с g(x), а потом результат сдвинуть на 1 бит (не

циклически) в сторону младших разрядов. Чтобы найти информационное слово достаточно найденное кодовое слово разделить на порождающий полином g(x). В итоге получилось исправленное кодовое слово и нашли соответствующее ему информационное слово. Здесь мы подробно исследовали метод кодирования и декодирования совершенного двоичного кода Голея.

Коды с повторением

Исследуем коды повторения.

Код с повторением всей кодовой комбинации можно рассматривать как групповой код (n; k), где $n - k = k$, т.е. $n = 2 \cdot k$; $d_{min} = 2$ и позволяет обнаруживать ошибки любой кратности за исключением случаев, когда искажается один информационный и все соответствующие ему проверочные; два информационных и все соответствующие им проверочные.

Код содержит два кодовых слова: последовательность из n нулей и последовательность из n единиц. Первый символ слова можно назвать информационным, а остальные r - проверочными. Декодер может использовать следующее правило. Подсчитывается число нулей и число единиц в полученной последовательности. Если нулей получено больше, чем единиц, то выносится решение, что передаваемое кодовое слово состояло из нулей; если единиц получено больше, чем нулей, то выносится решение, что передаваемое кодовое слово состояло из единиц. Если число оказывается равным числу единиц, то решение не принимается.

Ясно, что это правило декодирования позволяет декодировать правильно во всех случаях, когда шум в канале искажает меньше половины символов в каждом блоке. Если шум канала искажает точно половину символов некоторого блока, то декодер фиксирует отказ от декодирования: он не может декодировать полученное слово ни в одно из возможно передававшихся сообщений. Если шум в канале искажает более половины символов некоторого блока, то в декодере произойдет ошибка декодирования: он неверно декодирует полученное слово. При редких ошибках в канале связи вероятность отказа или ошибки декодирования для кода с повторением с большой блоковой длиной, очевидно, очень мала.

Например, рассмотрим код с повторениями с блоковой длиной n=5. Предположим, что если полученная последовательность содержит не более двух единиц, то она кодируется, как нулевое кодовое слово. В противном случае, оно декодируется как единичное слово.

Код Хемминга

Исследуем основные характеристики кода Хэмминга.

Коды Хэмминга – наиболее известные и, вероятно, первые из самоконтролирующихся и самокорректирующихся кодов. Построены они применительно к двоичной системе счисления.

Другими словами, это алгоритм, который позволяет закодировать какое-либо информационное сообщение определённым образом и после передачи (например, по сети) определить появилась ли какая-то ошибка в этом сообщении (к примеру, из-за помех) и, при возможности, восстановить это сообщение. Сегодня, я опишу самый простой алгоритм Хемминга, который может исправлять лишь одну ошибку.

Также стоит отметить, что существуют более совершенные модификации данного алгоритма, которые позволяют обнаруживать (и, если возможно исправлять) большее количество ошибок. Сразу стоит сказать, что Код Хэмминга состоит из двух частей. Первая часть кодирует исходное сообщение, вставляя в него в определённых местах контрольные биты (вычисленные особым образом). Вторая часть получает входящее сообщение и заново вычисляет контрольные биты (по тому же алгоритму, что и первая часть). Если все вновь вычисленные контрольные биты совпадают с полученными, то сообщение получено без ошибок. [5]

Имитационная модель приложения для кодирования введенного информационного слова и декодирования кодового слова с ошибкой.

Первым действием при работе с приложением, можно выбрать один из совершенных кодов: Хемминга, тривиальный, Голея.

После того как выбран совершенный код, нужно определить параметры. У кода Хемминга необходимо ввести параметры кода q и r, по которым уже просчитаются параметры кода n и k. У Тривиального же кода нужно ввести только параметр n, так как k – const и k=1. А у двоичного кода Голея параметры уже определены заранее.

Теперь можно приступить к кодированию или декодированию информационных, или кодовых слов. Для этого необходимо ввести в определенные поля слова нужной длины. Длина проверяется компьютером, т.е. в случае не правильного ввода, под вводом будет написано, что именно вы ввели неправильно. При вводе слова, если вам не выдало никаких ошибок, нужно нажать на кнопку кодирование или декодирование. [4]

На рисунке 1 приведена структурная схема имитационной модели, которая демонстрирует структуру кодирования и декодирования сообщения 1-им из 3-х кодов.

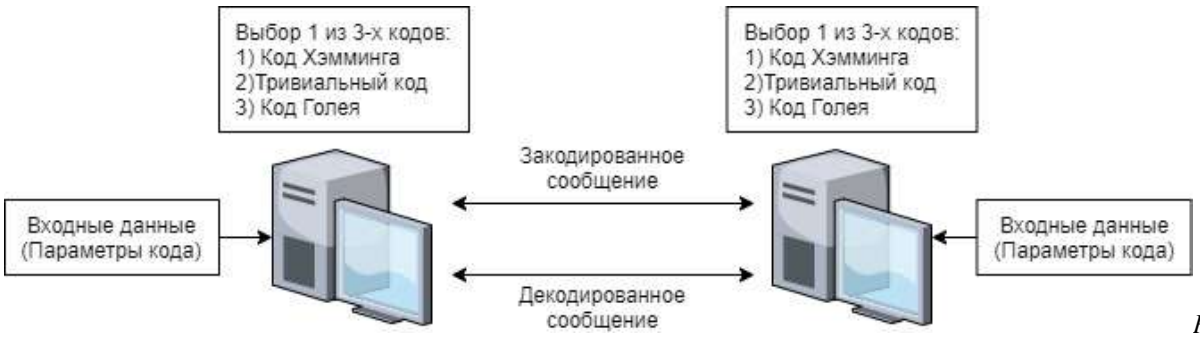

Рисунок 1 – Структурная схема имитационной модели

Мы рассмотрели алгоритмы имитационной модели программного средства для работы с совершенными кодами, необходимые для функционирования разрабатываемого приложения. Помимо использования уже существующих алгоритмов были разработаны собственные, выполняющие задачи в соответствии со спецификой совершенного кода. Например, методы для осуществления кодирования и декодирования кодов Хемминга и Голея. [6]

Заключение

В дальнейшем планируется разработка программного средства, позволяющего передавать содержимое сообщения клиентов на основе описанной выше имитационной модели.

Список источников

1. Технологии защиты данных. – Текст : электронный // SearchInfo : [сайт]. – 2020. - URL: https://searchinform.ru/informatsionnaya-bezopasnost/zaschita-informatsii/zaschita-dannykh/tekhnologii-zaschity-dannykh/ (дата обращения 19.04.2021 г.).

2. Деундяк, В.М., Маевский, А.Э., Могилевская, Н.С. Методы помехоустойчивой защиты данных: учебник / В.М. Деундяк, А.Э Маевский, Н.С. Могилевская; Южный федеральный университет. – Ростов-на-Дону: Издательство Южного федерального университета, 2014. – 309 с.

3. Хорев, Павел Борисович Методы и средства защиты информации в компьютерных системах: Учеб. пособие для студ. высш. учеб. заведений / П. Б. Хорев – Москва: Издательский центр «Академия», 2005. – 256 с.

4. Гробер Т. А., Савченко О. В. Создание имитационной модели поликлиники / Т. А. Гробер, О. В. Савченко – Ростов н/Д: Молодой исследователь дона ДГТУ, 2020. – 22 – 27с 5. Колесник В.Д.- учебное пособие по курсу «кодирование и декодирование сообщений» / В.Д. Колесник ; - Санкт-Петербург. 2005-2006.

5. Могилевская Н.С. - Введение в теорию информации, Ростов-на-Дону: ДГТУ, 2013. – 108 с.

6. Блейхут Р., - Теория и практика кодов, контролирующих ошибки / Р. Блейхут - М.: Мир, 1986. - 576 с.