# УМЕНЬШЕНИЕ ВЕРОЯТНОСТИ ВОЗНИКНОВЕНИЯ ИСКУССТВЕННЫХ ОШИБОК В ПОМЕХОУСТОЙЧИВЫХ КОДАХ С ПОМОЩЬЮ УВЕЛИЧЕНИЯ ОБЪЁМА ВКРАПЛЯЕМОГО СТЕГОСООБЩЕНИЯ

***Слипенчук Павел Владимирович***, *г. Москва*

*Статья посвящена проблеме уменьшения количества искусственных ошибок (artificial errors) при реализации стеганографии в помехоустойчивых кодах. Уменьшение количества искусственных ошибок позволяет усложнить стегоанализ и/или вкрапить большее количество данных. Автор предлагает в качестве решения данной актуальной проблемы использовать «коды уменьшения веса». В работе перечислены основные информационно-числовые характеристики «кодов уменьшения веса». Дано описание способа построения «простого кода уменьшения веса» с наибольшей добротностью. Обоснована актуальность изучения «непростых кодов уменьшения веса». Приведён пример использования кодов уменьшения веса в других классах стеганографии.*

***Ключевые слова****: стеганография в корректирующих кодах, коды уменьшения веса, искусственная ошибка, совершенная стегосистема.*

# REDUCTION OF ARTIFICIAL ERROR PROBABILITY IN ECC CODES BY INCREASE MEANS OF STEGANOGRAPHY DATA VOLUME

***Pavel Slipenchuk***, *Moscow*

*This work is devoted to a quantity reduction of artificial errors in ECC steganography. Quantity reduction of artificial errors allows to complicate stegoanalysis and/or to insert more amount of bits. The author offers to use "Weight Reduction Codes" (WRC) as the solution of this actual problem. Information-theoretic parameters of WRC are listed. The method of construct "Simple Weight Reduction Codes" (SWRC) with most possible Q factor is given. Relevance of studying of "Not Simple Weight Reduction Codes" (NSWRC) is explained. Examples of WRC use in other types of a steganography is given.*

***Keywords****: steganography in ECC, Weight Reduction Codes, artificial error, perfectly secure stegosystem.*

В настоящее время в области стеганографии осуществляются попытки построения строгих математических моделей, которые позволили бы формально дать описания процессам *вкрапления* и *извлечения* стегосообщения а так же ясно поставили бы задачу *стегоанализа*, задав строгое и формальное определение ***стойкости*** (или так называемой ***стегостойкости***).

В отличие от криптографии, которая имеет дело с формальными алгоритмами и работает с идеальными математическими объектами для построения шифров (SP-сеть, FSR, сеть Фейстеля и т. д.), стеганография является ***междисциплинарной*** наукой. Стеганографию невозможно представить в отрыве от техники, физики и иных наук, необходимые для изучения ***контейнеров***, в которые осуществляется вкрапление ***стегосообщения***. Без ясного, научного понимания того, что представляет собой *контейнер*, говорить о стеганографии как о научной дисциплине трудно, так как стегоанализ изучает именно *контейнеры*, пытаясь разделить то или иное множество *контейнеров* на ***пустые контейнеры*** и ***стегоконтейнеры*** (на не содержащие стегосообщения и содержащие стегосообщения). Если целью криптографии является создание шифра, то целью стеганографии – создание *алгоритма вкрапления* сообщения в уже заданные *контейнеры* (или создание алгоритма, порождающего *контейнеры* по определённому правилу, пытаясь «подражать» текстам и иным объектам, созданные человеком). Стеганография не создает новые «сущности», а как бы «приспосабливается» к уже созданным человеком и/или природой «сущностями», пытаясь вкрапить в них сообщение.

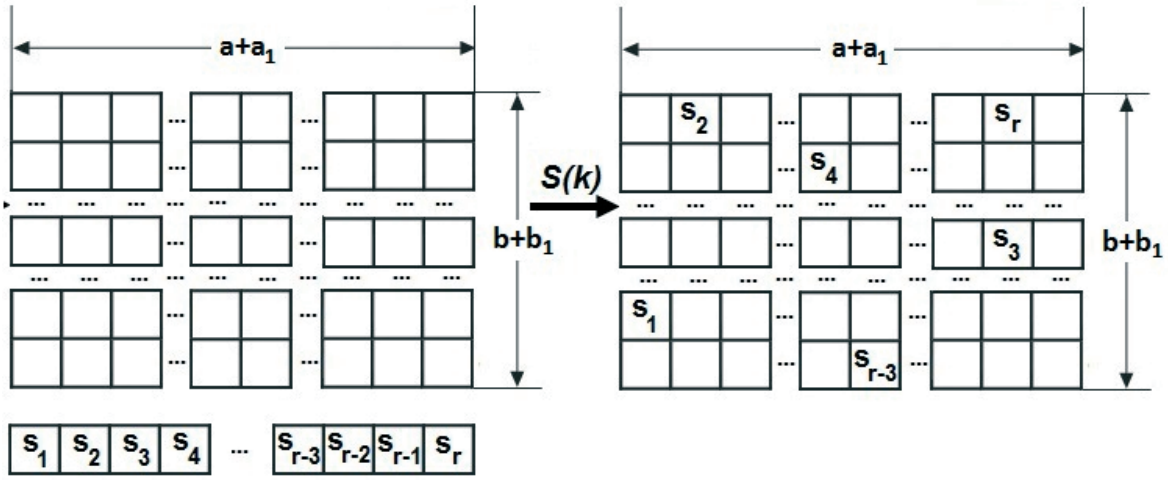


***Рис. 1**: Схема вкрапления стеганографических данных*

Именно по этой причине в настоящий момент стеганография как теоретическая дисциплина оторвана от практики и не смотря на огромный потенциал использования для гражданских (***цифровые отпечатки***[9], ***стеганографические водяные знаки***[9]) и военных целей (***скрытая передача данных***[9]), применяется не часто.

Таким образом, говоря языком философии, разумно определить различные ***стеганографические парадигмы***, отличающиеся друг от друга подходом к определению и подходом к изучению *контейнеров*. Одной из таких парадигм можно назвать «***информационную парадигму***», которая представляет контейнер как последовательность букв, символов или бит, связанных друг с другом по каким-либо законам. Эти законы могут иметь очень сложный вид и плохо поддаваться изучению; однако формально определив эти законы, можно попытаться построить различные информационно-теоретические модели стеганографии[1].

Начиная с «***дрезденской модели***»[3] существует ряд попыток построения информационно-теоретических моделей в рамках «*информационной парадигмы*». Одними из наиболее известных являются модели Кристиана Кашена [1], Томаса Миттельхольцера[4] и Святослава Волошиновского[6].

Введённая автором «***модель трех каналов***»[7,8] (МТК), в отличие от вышеперечисленных, не имеет общий характер и справедлива только для частного случая стеганографии – *стеганографии в кодах, исправляющих ошибки*. Однако для построения ***совершенной стегосистемы***[1], можно определить формальные критерии *стойкости*[8] для МТК.

Одной из подзадач в рассматриваемом нами классе стеганографии является задача уменьшения количества ***искусственных ошибок***[8]. Для реализации этой цели можно использовать специальный код, который без потери данных, при определенных методах вкрапления, позволяет уменьшить количество *искусственных ошибок*.

Данная статья посвящена формальному определению данного кода, названный автором «кодом уменьшения веса» (КУВ) и ряду характеризующих его параметров: *скорости*, *добротности*, *ёмкости*. В статье будут рассмотрены различные свойства и простые методы построения КУВ.

**Общая идея стеганографии в помехоустойчивых кодах.**

В различных носителях и каналах передачи данных могут возникать ошибки. Для передачи данных по каналу используют помехоустойчивые коды. Данные, подаваемые на вход помехоустойчивого кода, называют **информационным вектором**, а получаемые на выходе – **кодовым вектором**. По каналу с шумом передают *кодовый вектор*. Получатель отправляет на вход декодера *кодовый вектор*, в котором, возможно, возникли ошибки, и получает *информационный вектор*.

Суть стеганографии в помехоустойчивых кодах заключается во вкраплении битов стегосообщения в *кодовый вектор* перед отправкой данных в канал [7] (см рис.1). Позиции вкрапления задаются с помощью ключа. Вкрапляемые биты будем называть **искусственными ошибками (artificial errors)** [7], а ошибки, происходящие в канале передачи данных, будем называть **подлинными ошибками (genuine errors)**[7].

1 В научной литературе часто встречаются нелепые попытки строго определить понятие подсознательного канала (subliminal channel). Суть ошибки большинства исследователей заключена в том, что подсознательный канал не может в принципе быть определен, по крайней мере, в рамках информационной парадигмы.

**Актуальность уменьшения вероятности искусственной ошибки.**

Одной из проблем стеганографии в помехоустойчивых кодах является уменьшение количества *искусственных ошибок*. Большое количество *искусственных* ошибок нежелательно по двум причинам:

Возможно некорректное декодирование передаваемого контейнера;

Интуитивно понятно, что при стегоанализе, чем больше было вкраплено скрытой информации, тем проще определить наличие стегосообщения в контейнере.

Передаваемый по каналу стегоконтейнер содержит в себе искусственные и подлинные ошибки. Обозначим за $\boldsymbol{b}$ – кодовый вектор (кодовая последовательность бит) перед вкраплением стегосообщения; $\boldsymbol{d}$ – кодовый вектор после приёма из канала; $\boldsymbol{e}$ – вектор подлинных ошибок, $\boldsymbol{v}$ – вектор искусственных ошибок. Верна формула:

$$\boldsymbol{d} = \boldsymbol{b} + \boldsymbol{e} + \boldsymbol{v} \quad (1)$$

где $+$ означает побитовое сложение по модулю два

Под $v(j)$ и $e(j)$ будем обозначать -й бит в векторах *искусственной* и *подлинной* ошибки соответственно.

При использовании стеганографии, как канал ***скрытой передачи данных*** (СПД)[9], основная задача стегоанализа заключена в определении того, является ли определённая совокупность ошибок на носителе *подлинными ошибками* или *искусственными*? При решении данной задачи стоит обратить внимание на два вопроса:

1) как физически отличаются *искусственные* ошибки от *подлинных* ошибок?

2) по какому закону возникают физические ошибки и насколько правило выбора позиций для искусственных ошибок отличается от закона возникновения физических ошибок?

При решении первой задачи необходимо непосредственное изучение физического носителя информации или канала. Например, на оптических дисках ошибки возникают в результате старения, попадания пыли или возникновения царапин[5]. Если мы под микроскопом сможем увидеть царапины, которые являются возникновением подлинных ошибок, а при рассмотрении искусственных ошибок не увидеть ни царапин, ни следов старения, ни грязи, то можно подобным образом отличить подлинные ошибки от искусственных. Это очень похоже на ***атаку по побочным каналам*** (side channel attacks) в криптографии [2]. В стеганографии подобную атаку так же будем называть ***атакой по побочным каналам***.

При решении второй задачи мы абстрагируемся от конкретного носителя или канала передачи данных. Имея законы необходимо установить, является ли совокупность ошибок подлинными ошибками или среди них имеются искусственные ошибки, содержащие стегосообщение. Установка данных законов не всегда является тривиальной задачей. Во многих случаях необходимо разрабатывать математические модели возникновения ошибок.

Интуитивно понятно, что чем меньше возникнет искусственных ошибок, тем сложнее будет определить наличие стеганографии на носителе или в канале. При увеличении количества искусственных ошибок на носителе или в канале, третьей стороне легче, при прочих равных условиях, положительно определить наличие *стегосообщении* в *контейнере*.

Так же уменьшение вероятности искусственной ошибки может быть полезным при разработке *идеальной стегосистемы* [8,7]. В работе [8] есть доказательство, что *идеальные* стегосистемы являются ***совершенными (perfectly secure)***[1]. При использовании математической модели возникновения ошибок, приведённой в работе [8] при уменьшении вероятности искусственной ошибки можно вкрапить большее количество данных, чтобы стегосистема оставалась *совершенной*.

Таким образом, разумно разработать некий алгоритм, который уменьшил бы количество *искусственных ошибок*, но при этом позволил бы вкрапить ту же информацию в контейнер без потерь.

**Методы вкрапления искусственных ошибок.** Возьмём контейнер длины $N$. Предположим, что мы хотим вкрапить стегосообщение длины $l$. С помощью какого-либо ключа $K$ мы определяем позиции $\{a_1, a_2, a_3, a_4, \cdots, a_i\}$ в контейнере, в которые мы хотим вкрапить стегосообщение. Будем обозначать само стегосообщение за $\boldsymbol{s}$, а -й бит стегосообщения за $s_i$; $j$-й бит контейнера, до вкрапления стегосообщения, будем обозначать за $b(j)$ ; $j$-й бит стегоконтейнера, после вкрапления и прохождения через канал с шумом, будем обозначать за $d(j)$.

При этом мы можем руководствоваться следующими методами вкрапления.

**Способ 1: «Вкрапление замещением».** Если $s_i = 1$, то мы в позицию $a_i$ записываем 1. Если $s_i = 0$, то мы в позицию $a_i$ записываем 0.

В данном методе искусственная ошибка будет равна 1, если $b(a_i) = 0$ и $s_i = 1$ или $b(a_i) = 1$ и $s_i = 0$. Иначе говоря, верно равенство $b(a_i) + s_i = 1$. Если же $b(a_i) + s_i = 0(mod\ 2)$, то искусственная ошибка равна 0.

Извлечение бита стегосообщения происходит следующим способом. Вторая сторона просто берёт бит с позиции $a_i$ из стегоконтейнера $\boldsymbol{d}$:

$$\forall i = \overline{1,n} \Rightarrow s_i = d(a_i) = b(a_i) + \\ + v(a_i) + e(a_i)(mod\ 2) \quad (2)$$

Возможна ситуация, когда подлинная ошибка $e(a_i) = 1$. Для этого необходимо перед вкраплением стегосообщения подать стегосообщение в помехоустойчивый код. Если $\forall a \in \{a_1, a_2, a_3, a_4, \cdots, a_l\}$ вероятность $P(e(a_i) = 1)$ будет отлична от 0.5, то возможно разработать помехоустойчивый код для стегосообщений. Разумеется, из-за использования помехоустойчивого кода, объём записываемой скрытой информации уменьшится.

**Способ 2: «Вкрапление ошибкой».** Если $s_i = 0$, то ничего не делаем. Если $s_i = 1$, то вкрапляем в позицию $a_i$ ошибку. То есть $v(a_i) = 1$.

Извлечение бита стегосообщения происходит следующим способом. Вначале контейнер проходит декодирование. После декодирования определяются позиции, на которых произошли ошибки. По ключу $k$ вторая сторона определяет позиции $\{a_1, a_2, a_3, a_4, \cdots, a_l\}$. Для каждой позиции определяем, произошла ошибка или нет. Если ошибка не произошла ($d(a_i) = b(a_i)$), то $s_i = 0$ , если произошла ($d(a_i) \neq b(a_i)$), то $s_i = 1$. Следовательно, верна формула:

$$\forall i = \overline{1,n} \Rightarrow s_i = d(a_i) + b(a_i)\ (mod\ 2) = \\ = v(a_i) + e(a_i)\ (mod\ 2) \quad (3)$$

Аналогично, возможна ситуация, когда $e(a_i) = 1$. Данную проблему будем так же решать помехоустойчивым кодом, как в способе 1.

**Простые коды уменьшения веса.** Предположим, что будем вкраплять вторым способом (вкрапление ошибкой) биты стегосообщения в контейнер. Тогда выходит, что чем меньше единиц в стегосообщении, тем меньше *искусственных ошибок* произойдёт в контейнере.

Будем называть количество единиц в стегосообщении **весом** стегосообщения.

Возникает задача построения кода, который на выходе создавал бы стегосообщения меньшего *веса*, чем стегосообщение на входе.

Разобьём сообщение длины $A$ на $u$ равных частей, каждая часть которой имеет длину $m$. Каждую часть длины $m$ будем называть **входным вектором (или информационным вектором)**. Само сообщение будем называть **входным (или информационным)** сообщением. Зададим функцию $U$, которая каждому *входному вектору* однозначно ставит в соответствие некую другую последовательность длины $n$. Эту последовательность длины $n$ будем называть **выходным вектором (или кодовым вектором)**. Затем из $u$ *кодовых векторов* собираем новое сообщение длины $\frac{n}{m} \cdot A$. Данное сообщение будем называть **выходным (или кодовым) сообщением**.

Величину $r = \frac{m}{n}$ будем называть **скоростью (кода)**, так же как в теории кодирования**.**

Если функция $U$ является биекцией, то эту функцию и обратную к ней будем называть **простым кодом уменьшения веса из $m$ в $n$** или, для краткости, **кодом из $m$ в $n$**. Функцию $U$ будем называть **кодированием,** а обратную к ней **декодированием**.

Задача *кода уменьшения веса* в получении кодового стегосообщения меньшего веса, чем входное сообщение. При этом чем ближе скорость кода к единице, тем лучше, так как от скорости кода в минус первой степени зависит то, во сколько раз уменьшится максимально-допустимый объём скрытого сообщения в контейнере.

Сам *простой код из $m$ в $n$* можно задать таблицей из двух столбцов, с $2^m$ строками. В первом столбце («столбец $m$») перечисляются всевозможные последовательности из нулей и единиц длины $m$, а во втором («столбец $n$») — кодовые вектора, в которые переходят информационные вектора напротив. Данную таблицу будем называть **таблицей кода**. Таким образом, сам код можно задавать его *таблицей кода*.

Приведём пример *простого кода уменьшения веса из 3 в 4* с помощью *таблицы кода*:

| m | n |
|---|---|
| 000 | 0000 |
| 001 | 0001 |
| 010 | 0010 |
| 011 | 0100 |

| m | n |
|---|---|
| 100 | 1000 |
| 101 | 0011 |
| 110 | 0101 |
| 111 | 1001 |

Например, если входное сообщение 010-110-010-001-000-101-011 (дефисы расставлены для удобства), то выходное сообщение: 0010-0101-0010-0001-0000-0011-0100. Вес входного сообщения 9, вес выходного 8.

Будем считать, что вероятность возникновения единицы во *входном* сообщении равна 0.5. Таким образом, если длина *входного сообщения* равна $A$, то вес, в среднем, будет равен $0.5 \cdot A$. Рассмотрим вес *выходного сообщения*. *Вероятность возникновения единицы* из-за равной вероятности каждой из 8 возможных последовательностей длины 3 вычисляется как сумма всех единиц во всех последовательностях длины $n$, делённая на сумму всех единиц и нулей:

$$P_1 = \frac{0 \cdot 1 + 1 \cdot 4 + 2 \cdot 3}{4 \cdot 2^3} = \frac{10}{32} \quad (4)$$

Найдём *среднестатистический вес кодового сообщения* длины $A$. Для этого необходимо *вероятность возникновения единицы* умножить на *скорость кода* в минус первой степени и на длину входного сообщения:

$$W_{\text{код.сообщ}} = \frac{4}{3} \cdot P_1 \cdot A = \frac{4}{3} \cdot \frac{10}{32} \cdot A = \frac{5}{12} \cdot A \quad (5)$$

Величину, равную отношению *среднестатистического веса выходного сообщения* к длине *входного сообщения* будем называть **удельным весом (простого кода)**.

$$P = \frac{W_{\text{код.сообщ}}}{A} = \frac{5}{12} \quad (6)$$

*Удельный вес* должен быть меньше 0.5, иначе код не имеет практического смысла. Чем больше разница между 0.5 и *удельным весом*, тем в большее количество раз уменьшен вес стегосообщения.

Отношение среднестатистического веса *входного сообщения* длины $A$, к среднестатистическому весу *выходного сообщения* будем называть **добротностью кода,** при $A \to \infty$.

$$Q = \lim_{A \to \infty} \frac{1/2 \cdot A}{W_{\text{код.сообщ}}} = \lim_{A \to \infty} \frac{1/2 \cdot A}{P \cdot A} = \frac{1/2}{P} = \frac{6}{5} \quad (7)$$

*Добротность* показывает во сколько раз вес *выходного сообщения* меньше веса *входного сообщения*.

Таким образом, качество *простого кода уменьшения веса* характеризуется двумя величинами: *добротностью кода* и *скоростью кода*.

Заметим, что кроме п*ростых кодов уменьшения веса* можно рассмотреть коды с переменными длинами информационных и кодовых векторов.

**Лемма о простом коде с минимальным удельным весом.** Возьмём множество всех кодов уменьшения веса из $m$ в $n$. Рассмотрим код с минимальным удельным весом $P_{MIN}(n, m)$, а *таблицу кода* будем обозначать $A_{MIN}$. Если их несколько – возьмём любой из них.

Рассмотрим код, построенный по следующему принципу:

1) построим таблицу с двумя столбцами и с $2^m$ строками. В первом столбце будем писать информационные вектора, а во втором — кодовые вектора, в которые кодер переводит информационные вектора, стоящие на той же строке. В первом столбце перечислим всевозможные входные вектора длины m. Их всего $2^m$ и каждый из них равновероятен, так как вероятность появления единицы во входном сообщении равна 0.5.

2) Во второй столбец запишем один вектор длины $n$, веса 0, затем, будем заполнять оставшиеся столбцы векторами длины $n$, веса 1 так, чтобы эти вектора не повторялись. Когда не останется векторов веса 1, будем заполнять векторами веса 2. Когда «закончатся» вектора веса 2, будем заполнять векторами веса 3 и т.д.

Примером построенного данного кода является ранее построенный *код уменьшения веса из 3 в 4*.

*Удельный вес* полученного кода будем обозначать за $P(n, m)$, саму *таблицу кода* будем обозначать $A$. Данный код будем называть **оптимальным кодом (уменьшения веса из $n$ в $m$)**.

**Утверждение**. $P(n, m) = P_{MIN}(n, m)$.

◀ Если в таблице $A_{MIN}$ во втором столбце нет вектора нулевой длины, то мы заменяем любой кодовый вектор в таблице на вектор нулевой длины и получаем код с меньшой вероятностью возникновения единицы, и следовательно с меньшим *удельным весом*. Получили противоречие. Следовательно, в $A_{MIN}$ есть вектор нулевой длины.

Возьмём в $A_{MIN}$ кодовый вектор наибольшего веса. Будем считать, что этот вес равен $u$. Рассмотрим в таблице кодовые вектора меньшего веса $i$. Для каждого $i$ существуют $C_n^i$ различных комбинаций векторов длины $n$ с весом $i$. Если хотя бы одного из этих векторов не будет, то заменяем вектор веса $u$ на данный отсутствующий вектор. Таким образом, получим новую таблицу кода, удельный вес которого меньше $P_{MIN}(n, m)$. Противоречие. Следовательно, в таблице кода $A_{MIN}$ присутствуют все $C_n^i$ различных комбинаций векторов длины $n$ с весом $i$.

Следовательно, $A_{MIN}$ построен так же, как код $A$. ▶

Впредь удельный вес *оптимального кода из m в n* будем обозначать за $P(n, m)$. *Таблицу кода* любого *оптимального кода из m в n* будем обозначать за $A(n, m)$.

**Вычисление добротности произвольного оптимального кода.** При рассмотрении *оптимального кода*, за $W(n,m)$ будем обозначать количество единиц в правом столбце *таблицы кода*. Общая формула вычисления *удельного веса*:

$$P(n,m) = \frac{n}{m} \cdot \frac{W(n,m)}{n \cdot 2^m} = \frac{W(n,m)}{n \cdot 2^m} \quad (8)$$

Введём обозначения:

$$F(n,u) = 1 + C_n^1 + C_n^2 + \cdots + C_n^u \quad (9)$$

$$D(n,u) = 0 + 1 \cdot C_n^1 + 2 \cdot C_n^2 + \cdots + u \cdot C_n^u \quad (10)$$

Если код является *оптимальным*, то $W(n,m)$ вычисляется по формуле:

$$W(n,m) = D(n,i) + (i+1)(2^m - F(n,i)),$$

где $i \in \mathbb{N}$, такое, что:

$$F(n,i+1) \geq 2^m \quad (11)$$
$$F(n,i) < 2^m$$

Действительно, от одного вектора веса 0 получаем 0 единиц к величине $W(n,m)$; от $C_n^1$ векторов веса 1 получаем по 1 единице к величине ; от $C_n^2$ векторов веса 2 получаем по 2 единицы к величине $W(n,m)$ и т. д. Это и есть величина $D(n,i)$. Всего векторов $2^m$. Таким образом, в конце будем иметь $(2^m - F(n,i))$ векторов веса $(i+1)$.

Из формулы (8) и определения *добротности* (7) можно получить формулу вычисления *добротности* произвольного *оптимального кода уменьшения веса*:

$$Q(n,m) = \frac{1/2}{P(n,m)} = \frac{m \cdot 2^{m-1}}{W(n,m)} \quad (12)$$

Формулу (11) можно преобразовать в другой вид:

$$W(n,m) = (i+1) \cdot 2^m + \sum_{j=0}^{i}(j - (i+1)) \cdot C_n^j,$$

где $i \in \mathbb{N}$, такое, что:

$$F(n,i+1) \geq 2^m \quad (13)$$
$$F(n,i) < 2^m$$

**Некоторые свойства оптимальных кодов.** Как следует из доказанной выше леммы, для любого *не оптимального* простого кода уменьшения веса из $m$ в $n$ *удельный вес* будет больше *удельного веса* оптимального простого кода уменьшения веса из $m$ в $n$. Таким образом, оптимальные коды из $m$ в $n$ – это коды, обеспечивающие наибольшую *добротность*.

Для оптимальных кодов верно неравенство:

$$W(n+1,m) \leq W(n,m) \quad (14)$$

◀Из формулы (11) следуют формулы:

1) $W(n,m) = D(n,i_1) + (i_1+1)(2^m - F(n,i_1))$

2) $W(n+1,m) = D(n+1,i_2) +$
$+ (i_2+1)(2^m - F(n+1,i_2))$

Рассмотрим случай, когда $i_1 = i_2 = 0$. Тогда справедлива формула:

$$W(n+1,m) - W(n,m) =$$
$$= 0 + (2^m - 1) - (0 + (2^m - 1)) = 0 \quad (15)$$

Если $i_1 = i_2 = i \neq 0$, то вычитая из второй формулы первую и используя формулу (13), получаем:

$$W(n+1,m) - W(n,m) =$$

$$= \sum_{j=0}^{i}(j - (i+1)) \cdot (C_{n+1}^j - C_n^j) < 0 \quad (16)$$

Действительно, так как $j \leq i$, то $(j - (i+1)) < 0$, а $C_{n+1}^j - C_n^j \geq 0$. Таким образом, неравенство в формуле (16) верно.

Если $i_1 < i_2 = 1$, то $i_1 = i_2 + 1$. Действительно, при увеличении $n$ на единицу, $i$ не может уме[illegible] я из второй фо[illegible]у (13), получае

$$W(n+1,m) - W(n,m) = -C_{n+1}^i +$$
$$+ \sum_{j=0}^{i-1}((j - (i+1)) \cdot C_{n+1}^j) - ((j-i) \cdot C_n^j) <$$
$$< \sum_{j=0}^{i-1}((j-i) \cdot C_{n+1}^j) - ((j-i) \cdot C_n^j) =$$
$$= \sum_{j=0}^{i-1}(j-i) \cdot (C_{n+1}^j - C_n^j) \leq 0 \quad (17)$$

Случай, когда $i_1 > i_2$ невозможен.▶

Из формулы (14) следует, что при увеличении длины *кодового* вектора в *оптимальном* коде, добротность не уменьшается:

$$Q(n+1,m) \geq Q(n,m) \quad (18)$$

При $n = 2^m + 1$ величина $P(n,m)$ достигает наименьшего допустимого значения:

$$\forall n \geq 2^m + 1 \Rightarrow P(n,m) = P(2^m + 1, m) = \\ = \frac{2^m - 1}{m \cdot (2^m)} = \frac{1}{m} - \frac{1}{m \cdot 2^m} \approx \frac{1}{m} \quad (19)$$

Из формулы (19) следует верхняя граница для добротности *оптимальных кодов*:

$$Q(n,m) < \frac{m \cdot 2^{m-1}}{2^m - 1} \approx \frac{m}{2} \quad (20)$$

Из формул (13)(12) имеем:

$$Q(n,m) = \frac{m \cdot 2^{m-1}}{(i+1) \cdot 2^m + \sum_{j=0}^{i} (j - (i+1)) \cdot C_n^j} \geq \\ \geq \frac{m \cdot 2^{m-1}}{(i+1) \cdot 2^m} \quad (21)$$

Из (21) получаем для добротности нижнюю оценку:

$$Q(n,m) \geq \frac{m}{2 \cdot (i+1)}$$

где $i \in \mathbb{N}$, такое, что:

$$F(n, i+1) \geq 2^m \quad (22)$$
$$F(n,i) < 2^m$$

Обозначим за $Q_2(n,m)$ добротность какого-либо не оптимального кода из $m$ в $n$. Тогда, в силу леммы справедлива формула:

$$Q(n,m) > Q_2(n,m) \quad (23)$$

Коды с добротностью равной или меньшей единицы не имеют практического смысла. Не все *простые коды* имеют *добротность* большую единицы. Рассмотрим код из 2 в 4 представленный *таблицей кода*:

| m | n |
|---|---|
| 00 | 0001 |
| 01 | 0010 |

| m | n |
|---|---|
| 10 | 0100 |
| 11 | 1000 |

Для данного кода $P = \frac{4}{2} \cdot \frac{4}{4 \cdot 4} = \frac{1}{2}$. Таким образом, добротность равна 1. Для *оптимального* кода из 2 в 4, согласно формуле (12), имеем $Q(4,2) = \frac{4}{3}$.

**Способ определения максимальной добротности среди простых кодов в случае**

**постоянного максимального объёма вкрапляемого стегосообщения.** Предположим, что в контейнер можно записать не более $L$ бит данных стегосообщения. Данная величина постоянна. Размер стегосообщения, который мы хотим записать, равен l. Предполагаем, что $l < L$. Вкраплять будем *методом вкрапления ошибкой*. Задача найти простой код с наибольшей добротностью.

Возьмём величину $r_2 = \frac{l}{L}$. Возьмём и зафиксируем некое $m$. За $\lfloor x \rfloor$ будем обозначать целую часть от $x$. Для заданного $m$ построим оптимальный код с добротностью $Q(m, \lfloor \frac{m}{r_2} \rfloor)$. Рассматривать коды с *добротностью* $Q(m,n)$, где $n < \lfloor \frac{m}{r_2} \rfloor$ не имеет смысла, так как согласно формуле (18) *добротность* при уменьшении $n$ не увеличивается.

Величина $n$ не может быть больше либо равна $\lfloor \frac{m}{r_2} \rfloor + 1$. Поясним это. Обозначим за $L_2$ количество бит, полученные после *кода уменьшения веса*. Справедлива оценка:

$$L_2 = \frac{l \cdot n}{m} \geq \frac{l \cdot \left( \lfloor \frac{m}{r_2} \rfloor + 1 \right)}{m} > \frac{l \cdot \frac{m}{r_2}}{m} = \frac{l}{r_2} = L \quad (24)$$

Таким образом, $L_2 > L$.

Скорость кода при фиксированном $m$ будет равна:

$$r = \frac{m}{n} = \frac{m}{\lfloor \frac{m}{r_2} \rfloor} \approx r_2 \quad (25)$$

Таким образом, скорость лежит в пределах:

$$r_2 \leq r < \frac{m}{\frac{m}{r_2} - 1} = r_2 \cdot \frac{m}{m - r_2} \quad (26)$$

Неоптимальные коды в силу формулы (23) не имеет смысл рассматривать. Следовательно, для фиксированного $m$ имеет смысл рассматривать только код $A(m, \lfloor \frac{m}{r_2} \rfloor)$. Для каждого $m$, такого, что $2 \leq m \leq m_{MAX}$ найдём все коды $A(m, \lfloor \frac{m}{r_2} \rfloor)$ и определим их *добротность*. Выберем код с наибольшей добротностью.

Величину $m_{MAX}$ можно выбрать в соответствии с требованиями быстродействия или памяти. Действительно, память растёт как $O(2^m)$. Однако бы-

стродействие алгоритма зависит только от времени обращения в таблицу, то есть как $O(1)$.

Было бы актуально изучить коды уменьшения веса, кодирование и декодирование которых занимало бы меньше памяти и при этом не сильно увеличивало время обращения. Решение данной проблемы, возможно, позволит использовать оптимальные коды с большим $m$, что при табличном кодировании невозможно в силу нехватки памяти.

**Понятие «ёмкости кода». Лемма о совпадении ёмкости и добротности.**

Пусть $l_0$ – максимальное количество данных, которые мы можем вкрапить без использования кода уменьшения веса, а $l_1$ – максимальное количество входных данных, которые мы можем вкрапить с использованием рассматриваемого кода. Определим за **ёмкость (кода)** величину:

$$C = \frac{l_1}{l_0} \quad (27)$$

Пусть существует некий контейнер, в котором мы можем допустить не более **V** искусственных ошибок. Сделаем ряд предположений:

**Предположение 1.** Ошибки можно допустить в любом месте контейнера.

**Предположение 2.** Период алгоритма, определяющий позиции бит данных, в которые осуществляется стеганография, равен объёму контейнера.

**Предположение 3.** $U(l_1) < W$

Учитывая, что вероятность появления 1 во входных данных равно 0.5, мы получаем максимальное допустимое количество данных, которые можно записать без использования стеганографического алгоритма:

$$l_0 = \frac{V}{2} \quad (28)$$

Определим за $L$ – длину выходного сообщения кода уменьшения веса, при условии, что на вход подали сообщение длины $l_1$. Определим за $p_1$ – вероятность появления единицы в бите выходного сообщения, а за $W$ – среднестатистический вес выходного сообщения. Верна формула:

$$V = L \cdot p_1 = \frac{L \cdot W}{L} = W = \frac{2 \cdot l_1}{Q} \quad (29)$$

Из (29) и (28) следует, что при использовании кода можно вкрапить максимальное количество данных:

$$l_1 = l_0 \cdot Q \quad (30)$$

Следовательно, верно равенство:

$$C = Q \quad (31)$$

если верны предположения 1, 2 и 3.

Заметим, что в данном случае нам совершенно не важна скорость кода. На практике же очевидно, что $U(l_1) < W$.

Таким образом, увеличивая добротность кода, мы обеспечиваем возможность вкраплять большее количество данных. При верности предположений 1 и 2 можно утверждать, что чем больше добротность, тем больше данных можно вкрапить в контейнер. Скорость кода при этом может влиять только на быстроту кодирования и/или декодирования. На стойкость стеганографической системы скорость не оказывает никакого влияния.

Формула (31) является важным свойством добротности. Из этой формулы следует, что имея достаточно большой контейнер, уменьшая при этом $V$, можно увеличить $Q$ и вкрапить тот же объём данных. Формально говоря, можно исказив всего один бит вкрапить сообщение любой длины $l_1 < W$! Разумеется, при этом потребуется разработать код с невероятной скоростью. Скорее всего, кодирование/декодирование будет занимать огромные промежутки времени. Находя компромисс между добротностью и скоростью, мы можем вкрапить большее количество данных, искажая то же количество бит.

Ограничение на максимальное количество ошибок важно при построении ***совершенных стегосистем*** [1,8]. Таким образом, коды уменьшения веса могут увеличить максимальный объём вкрапляемых данных так, чтобы стегосистема оставалась ***совершенной***.

**Пример непростого кода уменьшения веса. Актуальность исследования непростых кодов.**

Рассмотрим оптимальный код из 3 в 7 (далее код A):

| m | n | m | n |
|---|---|---|---|
| 000 | 0000000 | 100 | 0001000 |
| 001 | 0000001 | 101 | 0010000 |
| 010 | 0000010 | 110 | 0100000 |
| 011 | 0000100 | 111 | 1000000 |

Скорость данного кода будет равна $r_A = 3/7$.

Рассмотрим непростой код из 3 в 7 (далее код B)

| m | n | m | n |
|---|---|---|---|
| 000 | 0000000 | 100 | 0001 |
| 001 | 0000001 | 101 | 001 |
| 010 | 000001 | 110 | 01 |
| 011 | 00001 | 111 | 1 |

Коды, для которых длина информационного вектора является постоянной величиной, а длина кодового вектора – переменной будем называть **кодами уменьшения веса с переменным выходом и постоянным входом длины** $m$; для краткости данные коды будем называть **кодами с переменным выходом**.

Данный код корректно кодируется и декодируется. Корректность кодирования очевидна в силу постоянной длины информационного вектора. Поясним корректность декодирования. При декодировании мы считываем бит до тех пор, пока не считали семь нулей или единицу. Полученную последовательность декодируем по таблице. В силу того, что у кода B нет кодовых векторов веса большего 1, декодирование всегда будет корректным.

Доопределим понятие скорости кода для *кодов с переменным выходом*. Определим за $\hat{n}$ величину как **среднестатистическую длину кодового вектора**. В случае, если каждая входная последовательность равновероятна, то $\hat{n}$ равна сумме длин всех возможных кодовых векторов на $2^m$. **Скоростью кода с переменным выходом** будем называть величину:

$$r = \frac{m}{\hat{n}} \tag{32}$$

Как нетрудно видеть, для кода B величина вычисляется по формуле:

$$\hat{n} = \frac{7+7+6+5+4+3+2+1}{2^3} = \frac{35}{8} \tag{33}$$

Заметим, что:

$$\frac{7+7+6+5+4+3+2+1}{2^3} < \frac{8}{}$$

$$< \frac{8+7+6+5+4+3+2+1}{2^3} = \frac{36}{8} = 4$$

Получаем значение скорости кода $B$:

$$r_B = \frac{m}{\hat{n}} = \frac{8 \cdot 3}{35} > \frac{8 \cdot 3}{36} = \frac{3}{4} \tag{34}$$

Таким образом, скорость кода B больше скорости кода A.

Обозначим за $Q_A$ и $Q_B$ – добротности кодов $A$ и $B$ соответственно.

**Утверждение.** $Q_B = Q_A$.

◀

Определим код $C$, выходной алфавит которого {0,1,x}:

| m | n | m | n |
|---|---|---|---|
| 000 | 0000000 | 100 | 0001xxx |
| 001 | 0000001 | 101 | 001xxxx |
| 010 | 000001x | 110 | 01xxxxx |
| 011 | 00001xx | 111 | 1xxxxxx |

Например, последовательность $s_0 = 001 - 011 - 100 - 111$ (дефисы расставлены для удобства) кодируется как $C(s_0) = 0000001 - 00001xx - 0001xxx - 1xxxxxx$.

Оптимальным кодом $A$ эта же последовательность кодируется как $A(s_0) = 0000001 - 0000001 - 0000100 - - 0001000 - 1000000$. Так как на вес влияет только количество единиц, то отличие кода A от кода C только в том, что на некоторых местах вместо 0 стоят x, что на добротность никак не влияет. Из этого следует, что добротность кода A равна добротности кода C:

$$Q_A = Q_C \tag{35}$$

Для того, чтобы из s получить кодовый вектор кода B, можно из кодового вектора $C(s)$ убрать все символы x. Данную операцию обозначим как $Des(C(s))$. Таким образом:

$$B(s) = Des(C(s)) \tag{36}$$

Для нашего примера $B(s_0) = 0000001 - 00001 - 0001 - 1$. Но на добротность влияет только количество единиц в информационном и в кодовом векторах! Устранение всех символов x никак не изменит добротность кода, следовательно:

$$Q_C = Q_B \tag{37}$$

Из (35) и (37) следует:

$$Q_B = Q_A \tag{38}$$

▶

Таким образом, код B имеет ту же добротность, что и ***оптимальный код*** A из 3 в 7, но в то же время большую *скорость*! Это значит, что при решении проблемы уменьшения вероятности возникновения искусственной ошибки, имеет смысл рассматривать не только *простые коды уменьшения веса*.

**Использование кодов уменьшения веса в других классах стеганографии**. Доопределим понятие *искусственной ошибки* в других классах стеганографии как биты стегоконтейнера, отличные от битов пустого контейнера. Можно ли использовать *коды уменьшения веса* в других типах стеганографии? Для этого необходимо разработать

способ, с помощью которого можно определить произошла ли запись в указанные позиции или не произошла.

Например, при стеганографии в *.bmp файле методом LSB представим, что пользователи обменялись не только ключами, но и множеством пустых контейнеров. Имея пару пустой контейнер – стегоконтейнер, сторона, принявшая стегоконтейнер, может определить позиции, в которых произошли ошибки.

Со стороны может показаться, что данная идея не имеет смысла, так как стороны преждевременно обмениваются не только коротким ключом, но и огромными данными, которые будут контейнерами в стегоканале. На самом деле, при современных носителях информации данная идея не является абсурдной. Формально, каждый пустой контейнер можно считать «одноразовым ключом». Стегоключом можно восстановить последовательность вкраплённых нулей и единиц, получив последовательность позиций, в которые было записано стегосообщение. Если бит отличен, то первая сторона вкрапила 1, если бит стегосообщения совпадает с битом пустого контейнера, то вкрапляли 0.

Уменьшение веса стегосообщения означает уменьшения количества бит, отличные от пустого контейнера. Стегоанализ усложнится.

Таким образом, коды уменьшения веса могут быть применимы не только в стеганографии в помехоустойчивых кодах.

**Выводы:**

1. Задача уменьшения вероятности искусственной ошибки актуальна для стеганографии в помехоустойчивых кодах, так как позволяет вкрапить большее количество данных, не уменьшая стойкость системы.
2. Простые коды уменьшения веса могут служить способом уменьшения вероятности искусственной ошибки путём увеличения объёма вкрапляемых данных.
3. Основными показателями качества кода уменьшения веса являются скорость кода, и добротность кода; чем больше каждый из этих параметров, тем лучше код.
4. Нахождение качественных непростых кодов уменьшения веса является актуальной проблемой.
5. Табличный метод имеет сложность кодирования и декодирования $O(1)$, но требует памяти $O(2^m)$, что при больших $m$ делает невозможным кодирование с помощью *таблицы кода*. Нахождение способов кодирования и декодирования, требующих меньше памяти и несущественно увеличивающие время кодирования и декодирования является актуальной задачей в стеганографии.
6. Если существует алгоритм определения для каждого бита стегоконтейнера, равен ли этот бит биту пустого контейнера, то коды уменьшения веса могут быть эффективны при реализации стеганографических алгоритмов для данного типа контейнера.

**Литература (References):**


1. Cachin C. An Information-Theoretic Model for Steganography // MIT Laboratory for Computer Science - 2002. Edition. – October, 2010. – P.31
2. P. Kocher, J. Jaffe, B. Jun Differential Power Analysis // Интернет-доступ: http://www.cryptography.com/public/pdf/DPA.pdf
3. J.Zöllner, H.Federrath, H.Klimant, A.Pfitzmann, R.Piotraschke, A.Westfeld, G.Wicke, G.Wolf Modeling the Security of Steganographic Systems // Berlin. Springer: Information Hiding 1998, LNCS 1525, pp. 344-354, 1998.
4. Thomas Mittelholzer An Information-Theoretic Approach to Steganography and Watermarking // Berlin. Springer Information Hiding, LNCS 1768, pp. 1–16, 2000
5. Standard Blu-ray Disk Format: 1.B Physical Format Specifications for BD-R. 5th Edition. – October,2010. – P.31
6. S.Voloshynovskiy, S.Pereira, V. Iquise, T. Pun "Attack Modeling: Towards a Second Generation Watermarking Benchmark," Sig. Processing. Special Issue on Information Theoretic Issues in Digital Watermarking, 2001, vol. 81, no. 6, pp. 1177-1214
7. Слипенчук П.В. Стеганография в кодах, исправляющих ошибки // М.: Вестник МГТУ, Специальный Выпуск №5, 2013. 249 с.
8. Слипенчук П.В. Простое построение совершенных стегосистем на основе различных ошибок в помехоустойчивых кодах в модели трех каналов. Инженерный журнал: наука и инновации, 2013, вып. 11. URL:http://engjournal.ru/catalog/it/security/998.html Дата обращения: 24.05.2014
9. Слипенчук П.В. Перспективы и практическое применение стеганографии в помехоустойчивых кодах. // М.: ВНИИ ПВТИ, журнал БИТ, № 3, 2014.