Учреждение образования

«Белорусский государственный технологический университет»

**Отчет к лабораторной работе № 13**

по дисциплине «Защита информации и надежность информационных систем»

Выполнил:

Студент 3 курса 1 группы ФИТ

Парибок Илья Александрович

2023 г.

**Исследование методов текстовой стеганографии**

**Цель:** изучение стеганографических методов встраивания/извлечения тайной информации с использованием электронного файла-контейнера текстового формата, приобретение практических навыков программной реализации методов (рассчитана на 4 часа аудиторных занятий: 2 часа – часть 1, 2 часа – часть 2).

**Задачи:**

1. Закрепить теоретические знания из области текстовой стеганографии, классификации, моделирования стеганосистем подобного вида и сущности основных методов.

2. Изучить основные алгоритмы встраивания/извлечения тайной информации на основе методов текстовой стеганографии, получить опыт практической реализации методов.

3. Разработать приложение для реализации алгоритмов встраивания/извлечения тайной информации на основе методов текстовой

стеганографии.

4. Познакомиться с методиками оценки стеганографической стойкости методов.

5. Результаты выполнения лабораторной работы (отдельно по каждой из 2 частей) оформить в виде описания разработанного приложения (для части 2), методики выполнения экспериментов с использованием приложений и результатов экспериментов.

**Теоретические сведения**

Для понимания сущности некоторых из методов полезно познакомиться с важнейшими особенностями и параметрами использования стилей (в том числе пространственно-геометрическими параметрами шрифтов), на основе которых строится текстовый файлконтейнер. На рисунке 1 показаны основные из параметров шрифта.

Изображение выглядит как текст, Шрифт, снимок экрана, диаграмма

Автоматически созданное описание

Рисунок 1 – Параметры шрифта

К синтаксическим методам компьютерной стеганографии, которые характеризуются сравнительно невысокой эффективностью (с точки зрения объема встраиваемой информации), относятся следующие (такие методы мы отнесем к числу базовых синтаксических методов):

• изменение расстояния между строками электронного текста (Line-Shift Coding); называется методом изменения межстрочных интервалов; сущность заключается в том, что используется текст с различными межстрочными расстояниями: выделяется максимальное и минимальное расстояния между строками, позволяющее кодировать соответственно символы «1» и «0» осаждаемого сообщения;

• изменение расстояния между словами в одной строке электронного текста (Word-Shift Coding); суть метода состоит в том, что осаждение информации основано на модификации расстояния между словами текста-контейнера; • изменение количества пробелов между словами (частный случай метода Word-Shift Coding); основан на том, что, например, чередование одинарного пробела и двойного (хх\_хх\_\_хх) кодирует «1», переход же с двойного пробела на одинарный кодирует «0» (хх\_\_хх\_хх);

 • на основе внесения специфических изменений в шрифты, т. е.начертания отдельных букв (Feature Coding); заключается в изменении написания отдельных букв используемого стандартного шрифта: визуально заметны различные образы, соответствующие буквам с верхними (например, l, t, d) или нижними (например, a, g) выносными элементами (см. рис. 1); например, букву «А» можно модифицировать, незначительно укорачивая длинную нижнюю часть буквы (рис. 2);

Изображение выглядит как зарисовка, дизайн, черно-белый

Автоматически созданное описание

Рисунок 2 – Пример применения метода Feature Coding:

а – пустой контейнер; б – заполненный контейнер

• изменение интервала табуляции; аналогичен вышеописанному методу изменения количества пробелов, только в этом случае меняется не количество пробелов, а соответственно расстояние между строками и интервал табуляции;

• Null Chipper (дословно – несуществующий, нулевой лепет); предполагает размещение тайной информации на установленных позициях слов или в определенных словах текста-контейнера, который, как правило, лишен логического смысла (как видно, действительно лепет);

• увеличение длины строки; предусматривает искусственное увеличение длины каждой строки за счет пробелов: например, нет пробела (определяется положением знака перехода на новую строку) – «0», один пробел – «1»;

• использование регистра букв; для обозначения бита секретного сообщения, представленного единицей, используется символ нижнего регистра, а нулем – верхнего (или наоборот);

• использование невидимых символов; знак «пробел» кодируется символом с кодом 32, но в тексте его можно заменить также символом, имеющим код 255 (или 0), который является «невидимым» и отображается как пробел.

Рассмотренные базовые методы могут применяться независимо и совместно, сохраняют исходный смысл текста, а обеспечиваемые ими показатели плотности кодирования при совмещении складываются. Еще одна важная особенность. Перечисленные методы работают успешно до тех пор, пока тексты представлены в коде ASCII.

Существуют также стеганографические методы, которые интерпретируют текст как двоичное изображение. Необходимо отметить, что данные методы не чувствительны к изменению масштаба документа, что обеспечивает им хорошую устойчивость к большинству искажений, которые могут иметь место при активных атаках. К числу основных лингвистических методов относятся [2, 52]:

 • метод синонимов; в качестве примера приведем подмножество синонимов: {«тайный», «секретный», «конфиденциальный», «доверительный»}. В приведенном подмножестве каждое слово имеет единственное одинаковое смысловое значение, что позволяет закодировать каждое слово своим уникальным кодом (т. е. выполнить операцию осаждения), например, «доверительный» – 00, «конфиденциальный» – 01, «секретный» – 10, «тайный» – 11. Подобное кодирование позволяет выбирать одно из четырех слов (как видим, они для удобства расположены по алфавиту) в зависимости от двух битов секретного сообщения. Отметим, что при этом, независимо какое из четырех слов будет выбрано, семантика сообщения не изменится. Очевидно, что при этом количество символов, соответствующих одному из синонимов используемого подмножества, зависит от общего числа элементов в подмножестве. Кроме того, обеим сторонам стеганосистемы должен быть известен общий алгоритм кодирования, т. е. один из ключей системы. Следует отметить, что в каждом подмножестве синонимов их упорядочивание должно выполняться по одному и тому же алгоритму и у отправителя сообщения, и у его получателя. В случае наличия слов с несколькими смысловыми значениями подобное кодирование оказывается невозможным. Также невозможно кодирование, если один из синонимов состоит из двух (или более) разделенных пробелом слов;

• метод переменной длины слова; основан на том, что длина слов в сообщении зависит от содержания секретного сообщения и способа кодирования слов: обычно одно слово текста-контейнера определенной длины кодирует два бита информации из стеганосообщения; например, слова текста длиной в 4 и 8 символов могут означать комбинацию битов «00», длиной в 5 и 9 – «01», 6 и 10 – «10», 7 и 11 букв – «11»; слова короче 4 и длиннее 11 букв можно вставлять где угодно для лексической и грамматической связки слов в предложении – программное приложение, которое декодирует принятое сообщение (извлекает сообщение из стеганоконтейнера), будет просто игнорировать их;

• метод первой буквы – программа-помощник в этом методе накладывает ограничение уже не на длину слова, а на первую (можно на вторую) букву; обычно одну и ту же комбинацию могут кодировать несколько букв, например, комбинацию «101» означают слова, начинающиеся с «А», «Г» или «Т;

• мимикрия; мимикрия генерирует осмысленный текст, используя синтаксис, описанный в Context Free Grammar (CFG), и встраивает информацию, выбирая из CFG определенные фразы и слова; грамматика CFG – это один из способов описания языка, который состоит из статических слов и фраз языка, а также узлов.

**Метод на основе апроша**

Апрош определяет расстояние между соседними символами текста. Фактически апрош состоит из двух таких расстояний – полуапрошей, являющихся как бы пространством, прилегающим к каждому из символов-соседей (см. рис. 13.1 и 13.4). Мы далее будем обращаться только к апрошу. Согласно существующим техническим правилам набора нормальный апрош должен быть равен половине кегля (размера) шрифта. Идея метода [69, 70] заключается в следующем. Встраивание сообщения в контейнер может быть основано на модификации базового (устанавливаемого текстовым процессором по умолчанию) значения апроша ао, его изменением от базового до некоторого максимального аmax (или минимального аmin), которое зрительно не должно отличаться от стандартного. Такое изменение производится с определенным шагом (дискретно) Δаi, каждому значению которого присваивается определенный бит или определенная комбинация битов.

Изображение выглядит как текст, Шрифт, Графика, графический дизайн

Автоматически созданное описание

Рисунок 3 – Изменение Апроша

Изменение величины апроша между двумя определенными символами текста относительно базового значения ао на небольшое расстояние (пункты (пт) или доли пункта) формально можно представить в следующем виде: at = ао + Δаt. (13.1) Такое изменение не должно вызывать визуально заметного уплотнения (Δаt <0) или разрежения (Δаt>0) групп символов. В текстовом процессоре MS Word апрош может принимать значения в диапазоне от 0 до 1584 пунктов.

Особенностью рассматриваемого метода является возможность одноразового размещения (в апроше одного символа) числа битов, определяемого дискретной разницей между минимальным и максимальным значениями Δа. Например, если отсчет вести от Δаmin до установленного интервала Δаt в виде параметра 0,1 nд (пт), то количество условных дискретных единиц nд, представленное в бинарном виде, определяет число битов, которые можно таким образом разместить; например, Δаmin= –0,5 пт, а Δаt = 0,3 пт. Разница между этими величинами составляет 0,8 пт: 8 · 0,1 или nt = 8 (в двоичном виде – 1000; в первом приближении именно такую бинарную комбинацию можно разместить (осадить) путем модификации конкретного апроша). На этой основе могут быть разработаны различные варианты кодировки осаждаемых комбинаций.

**Метод на основе кернинга**

В текстовых документах встречаются такие сочетания знаков, которые образовывают визуальные «дыры» либо «сгущения». Например, в текстах на основе кириллицы – это такие сочетания: «ГА», «TA», «ATA», «ЬТ» и т. п., на основе латиницы – «AY», «AV», «T;», «ff», а на основе греческого алфавита – «ΘΑ», «ΔΟ», «λκ» и др. Такие сочетания называются кернинговыми парами. Особенности «кернингования» приведены на рис. 4. Под кернингом обычно понимается процесс изменения межсимвольного расстояние между отдельными парами символов или кернинговыми парами (именно фактор парности отличает кернинг от апроша).

**Изображение выглядит как текст, Шрифт, белый, Графика

Автоматически созданное описание**

Рисунок 4 – Пояснение к понятию кернинга

Таким образом, технология кернинга, появившаяся в полиграфии после внедрения фотонабора (а затем и компьютерного набора), включает подбор межбуквенных интервалов для конкретных пар букв с целью улучшения внешнего вида и удобочитаемости текста. Такой избирательный подбор позволяет компенсировать неравномерности визуальной плотности текста, получаемой при использовании стандартных апрошей для каждой буквы.

С появлением цифрового фотонабора стало возможным хранить такие критические сочетания знаков (кернинговые пары) для некоторого условного шрифта, общее число которых мы обозначим Nk, в памяти компьютера с указанием величины (σ), на которую необходимо сдвинуть символы, чтобы визуально выровнять буквенные просветы. Как правило, текстовые редакторы или процессоры содержат встроенные средства настройки кернинга, который определяет стандартный межбуквенный интервал для того или иного шрифта. При этом σ устанавливается в соответствии со значениями из таблицы кернинговых пар, встроенной в вышеуказанный файл со шрифтом. Такая настройка позволяет выровнять шрифт и является стандартной. В некоторых шрифтах сейчас количество пар доходит до нескольких тысяч. Значение кернинга может быть как положительным (когда знаки раздвигаются, σ > 0), так и отрицательным (когда сдвигаются, σ < 0). Эта величина в программах верстки устанавливается в процентах от ширины символа пробела используемого шрифта.