МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №13 НА ТЕМУ:**

**Исследование методов текстовой стеганографии**

Выполнила студентка 3 курса 4 группы

Чёрная Яна

Минск 2025

**Цель:** изучение стеганографических методов встраивания/извлечения тайной информации с использованием электронного файла-контейнера текстового формата, приобретение практических навыков программной реализации методов.

**Теоретические сведения**

К текстовой стеганографии относятся методы, предусматривающие использование в качестве контейнера файла-документа текстового типа.

Текстовая стеганография:

1) Синтаксические методы:

– изменение расстояния между строками электронного документа;

– изменение расстояния между словами;

– изменение количества пробелов между словами;

– на основе внесения специфических изменения в шрифты;

– изменение интервала табуляции;

– Null Chipper;

– увеличение длины строки;

– использование регистра букв;

– использование невидимых символов.

Достоинства:

– Легко применяются к любому тексту (независимо от содержания, назначения, языка);

– Легко реализуются в программном коде (т.к. они полностью автоматические)

Недостатки:

– Невысокая эффективность (объем встраиваемой информации);

– Перечисленные методы работают успешно до тех пор, пока тексты представлены в коде ASCII;

– Неустойчивы к форматированию текса;

– Неустойчивы к изменению масштаба документа.

2) Лингвистические методы:

– Метод синонимов;

– Метод переменной длины слова;

– Метод первой буквы;

– Мимикрия.

**Практическая часть**

**Задание 1.** Разработать приложение, позволяющее использовать стенографический метод метода переменной длины слов.

**Принцип**: вставка разного количества пробелов после слов:

* Короткое слово (до 5 символов):
  + 1 → два пробела " "
  + 0 → один пробел " "
* Длинное слово:
  + 00 → " "
  + 01 → " "
  + 10 → " "
  + 11 → " "

Каждому одному или двум битам сообщения соответствует определённое количество пробелов после слова.

Тип слова (короткое или длинное) определяет, сколько битов закодировать и сколько пробелов вставить.



Рисунок 1.1 - Кодирование (encode\_message, метод variable)

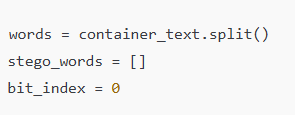


Рисунок 1.2 - Преобразует текст в бинарную строку (каждый символ — 8 бит).

Разбиваем текст контейнера на слова.

Будем вставлять пробелы между ними. Дальше — **основной цикл** по каждому слову. Если слово короткое — вставляется 1 или 2 пробела в зависимости от бита 0/1.



Рисунко 1.3 – Основной цикл

Метод использует изменение количества пробелов после слов для кодирования битов. В зависимости от длины слова кодируются один или два бита с помощью 1–5 пробелов.

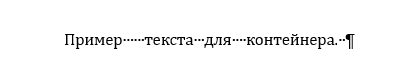


Рисунок 1.4 – Результат

Этот способ обладает высокой скрытностью, так как визуально текст почти не отличается от обычного, особенно при коротком сообщении. Однако реализация сложна: необходимо точно управлять пробелами и учитывать, что текстовые редакторы могут автоматически удалять лишние пробелы, что приведёт к потере данных.

**Задание 2.** Разработать приложение, позволяющее использовать стенографический метод апроша.

**Принцип**: добавление пробелов внутри слова, после символов.

* '1' → добавляется пробел после символа
* '0' → ничего не добавляется

Такой способ изменяет форму слова (межбуквенные пробелы), но визуально это не бросается в глаза.

- Проходит по каждому символу слова.

- Если очередной бит равен 1 — вставляет пробел после символа.

- 0 — ничего не делает.



Рисунок 2.1 - Кодирование (encode\_message, метод aprosha)

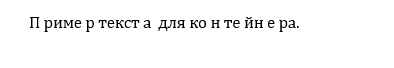


Рисунок 2.2 – Результат

Метод вставляет пробелы внутри слов — после символов, основываясь на бите 1. При этом каждый символ может нести 1 бит информации.  
Это упрощает реализацию, не требует различия между короткими и длинными словами, и легко восстанавливается: достаточно проверить, есть ли пробел между символами.

**Вывод:** В ходе лабораторной работы были изучены методы текстовой стеганографии и разработаны приложения для использования этих методов. Оба метода реализованы вручную без библиотек стеганографии. Первый — более скрытый, второй — проще реализовать, но легче обнаружить.