Лабораторная работа №6

Целостность информации. Простейшие методы стеганографии. Хэширование.

В данной работе предполагается рассмотрение методов проверки целостности информации на примере наиболее распространенных методов хэширования.

Стеганография

Цифровая стеганография – направление в стеганографии, основанное на сокрытии или внедрении дополнительной информации в цифровые объекты и использовании указанных объектов в качестве контейнеров при скрытом хранении или передаче информации, вызывающее незначительные искажения этих объектов.

LSB (Least Significant Bit, наименьший значащий бит) — суть этого метода заключается в замене последних значащих битов в контейнере (изображения, аудио или видеозаписи) на биты скрываемого сообщения. Разница между пустым и заполненным контейнерами должна быть не ощутима для органов восприятия человека.

Метод наименее значащих битов (LSB) считается наиболее популярным для цифровой стеганографии. Как уже говорилось, цифровая стеганография основывается на ограниченности способностей органов чувств человека и, как следствие, неспособности распознать незначительные вариации звука/цвета.

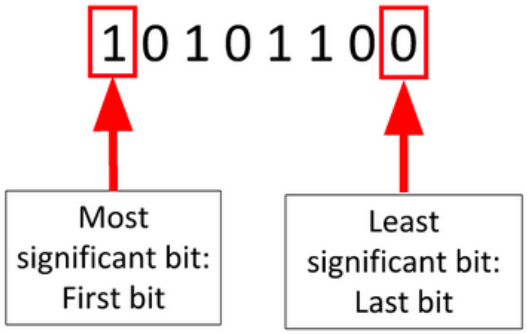

Рис. 1. «Значимость» битов цвета

Рассмотрим графический контейнер – изображение в формате BMP.В данном формате для описания каждой точки (пикселя) используются 3 байта, обозначающие в какой пропорции необходимо смешивать красный, зелёный и голубой цвета (цветовая схема RGB). Если произвести замену старших бит в этих байтах, цветовые изменения в картинке будут бросаться в глаза. Младшие же биты дают куда более незначительный вклад в изображение. Если использовать по одному младшему биту в каждом цвете для записи скрываемого сообщения, то распознать изменения человеческий глаз будет не способен.

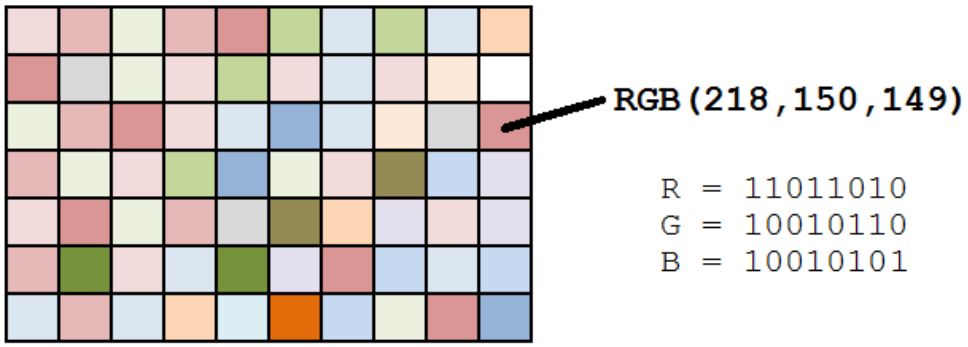

Рис. 2. Кодирование цветовой палитры RGB

Таким образом, в графическом изображении размером 117 Кбайт (200 на 200 пикселей) возможно спрятать, как минимум, 14 Кбайт скрытой информации.

Задание 1

Для выполнения работы допускается использование одной из готовых библиотек или реализаций LSB. Варианты:

Способ	Типы файлов	Описание	Сокрытие	Извлечение
openstego	PNG	Может использоваться не только для сокрытия данных, но и для водяных знаков. Использует RandomLSB — улучшенный алгоритм LSB с записью в Random Least Significant Bit. Поддерживает шифрование. Имеет также GUI.	openstego embed -mf secret.txt -cf cover.png -p password -sf stego.png	openstego extract -sf openstego.png -p abcd -xf output.txt
stegano	PNG	Работает не только с классическим LSB. Имеет гибкую настройку. Может использоваться как модуль Python.	stegano-lsb hide --input cover.jpg -f secret.txt -e UTF-8 --output stego.png	stegano-lsb reveal -i stego.png -e UTF-8 -o output.txt
cloackedpixel	PNG, JPG	Плохо справляется с большим сообщением. Поддерживает шифрование.	cloackedpixel hide cover.jpg secret.txt password	cloackedpixel extract cover.jpg-stego.png output.txt password
LSBSteg	PNG, BMP	Небольшая программа на Python с читабельным кодом.	LSBSteg encode -i cover.png -o stego.png -f secret.txt	LSBSteg decode -i stego.png -o output.txt

1. Примените LSB к одному из изображений из вашей Модели угроз, предварительно сохранив его в удобном формате.
2. Продемонстрируйте в отчёте работоспособность выбранного способа реализации LSB (скриншоты и описание к ним).

Требования:

1. *В изображение должен быть встроен текст методом LSB. Текст – любой, длина – любая, алфавит – любой.*
2. *На Портал загрузить архив, содержащий: изображение до встраивания скрытого текста, изображение со встроенным скрытым текстом, отчёт.*

На проходной балл по данной части лабораторной работы (2,5) достаточно использование онлайн-инструментов. Оценка 5 ставится при наличии работоспособного кода.

Хэширование

Хеш-функции используются в криптографических алгоритмах, электронных подписях, кодах аутентификации сообщений, обнаружении манипуляций, сканировании отпечатков пальцев, контрольных суммах (проверка целостности сообщений), хеш-таблицах, **хранении паролей** и многом другом.

Эти функции могут быть использованы для проверки дубликатов данных и файлов, проверки целостности данных при передаче информации по сети, безопасного **хранения паролей в базах данных** или, возможно, для какой-либо работы, связанной с криптографией.

Некоторые часто используемые хеш-функции:

- MD5: Алгоритм производит хеш со значением в 128 битов. Широко используется для проверки целостности данных. Не подходит для использования в иных областях по причине уязвимости в безопасности MD5 (подробнее - https://intuit.ru/studies/courses/28/28/lecture/20424?page=3).

- SHA: Группа алгоритмов, что были разработаны NSA Соединенных Штатов. Они являются частью Федерального стандарта обработки информации США. Эти алгоритмы широко используются в нескольких криптографических приложениях. Длина сообщения варьируется от 160 до 512 бит (подробнее - https://habr.com/ru/company/selectel/blog/530262/).

Модуль hashlib, включенный в стандартную библиотеку Python, представляет собой модуль, содержащий интерфейс для самых популярных алгоритмов хеширования и может быть использован в данной лабораторной работе.

Задание 2

1. **Используя одну из рассмотренных хэш-функций, показать различие или совпадение хэшей двух изображений из Задания 1.**

На проходной балл по данной части лабораторной работы (2,5) достаточно использование онлайн-инструментов. Оценка 5 ставится при наличии работоспособного кода.

Содержание отчёта:

1. Титульный лист.
2. Изображение до встраивания скрытого текста.
3. Изображение после встраивания скрытого текста.
4. Встроенный текст.
5. Листинг кода (при наличии).
6. Описание выбранного алгоритма хэширования (кратко)
7. Хэш-суммы изображений.
8. Листинг кода (при наличии).
9. Вывод.
10. Ответ на контрольные вопросы:
    a. В какой деятельности могла бы пригодиться стеганография для вас?
    b. Не используя сети Интернет попробуйте придумать свой способ скрыть сообщение в цифровом контейнере. Опишите в 3-5 предложениях.
    c. Какая хэш-функция наименее защищена от подбора исходного слова на основе хэша?
    d. В каких задачах наиболее применим md5?