Введение

В последние годы в мире программирования появился новый язык для создания веб-сайтов – CSS, который можно использовать вместе с HTML для разработки веб-интерфейсов. Сейчас эти два языка практически неразделимы и широко применяются для создания визуальной и функциональной составляющей веб-страниц. Веб-технологии развиваются стремительными темпами, и с увеличением объема цифровой информации возрастает и значение методов защиты данных. Веб-сайт является мощным инструментом распространения информации по всему миру, и потому он может быть использован для скрытой передачи данных. Одним из таких методов является стеганография – способ сокрытия информации внутри другого контента таким образом, чтобы он оставался незаметным для постороннего наблюдателя. Как клиентский скрипт, CSS виден всем пользователям в исходном виде, но его нельзя изменять, что открывает дополнительные возможности для скрытой передачи сообщений.

В данном реферате описана новая схема, использующая веб-инструменты, такие как CSS, для скрытия информации. Этот механизм секретной коммуникации основан на стеганографических техниках, позволяющих скрывать сообщения в файлах CSS, которые затем шифруются с помощью RSA, криптографического алгоритма с открытым ключом.

В то же время HTML, как основа веб-страниц, также предоставляет возможности для стеганографии. В данном реферате рассматриваются наиболее распространенные методы HTML-стеганографии: стеганография с перестановкой атрибутов, стеганография с использованием невидимых символов и стеганография с изменением регистра букв в тегах. Обнаружение и исследование этих методов позволяют выявлять скрытую информацию и углубить понимание возможностей стеганографии в веб-контенте.

Эти методы используют структуру HTML-документа для внедрения скрытых данных, оставаясь практически незаметными для пользователей. Изучение и обнаружение подобных методов являются актуальными задачами, особенно в контексте кибербезопасности и защиты данных в условиях современной цифровой среды. Понимание принципов HTML-стеганографии и механизмов ее обнаружения позволяет создавать более защищенные информационные системы, противостоящие потенциальным угрозам утечки данных и нарушению конфиденциальности.

1 Методы HTML-стеганографии

* 1. Метод перестановки атрибутов

Стеганография с перестановкой атрибутов (Attribute Permutation Steganography, APS) основывается на том, что порядок атрибутов в HTML-тегах не имеет значения для интерпретации браузером. Это означает, что один и тот же HTML-тег с разными порядками атрибутов будет одинаково отображаться на веб-странице. В листинге 1.1 приведен пример с HTML-тегом img, который содержит четыре атрибута: src, alt, width и height.

|  |
| --- |
| <img src=”w3schools.jpg” alt=”W3Schools.com” width=”104” height=”142”> |

Листинг 1.1 – Пример тега img

Здесь атрибуты могут быть переставлены в любом порядке, и браузер всё равно корректно отобразит изображение, не обращая внимания на очередность атрибутов. Возможность перестановки атрибутов открывает путь для кодирования информации: поскольку существует 24 возможных варианта порядка (4! = 4 × 3 × 2 × 1), можно использовать каждый из них для представления определенного символа или буквы. Таким образом, перестановки атрибутов позволяют закодировать частые буквы английского алфавита или другую информацию, скрытую от простого наблюдателя.

При этом такое изменение порядка не приводит к увеличению размера HTML-файла. Независимо от того, как расположены атрибуты, файл будет иметь одинаковый вес и визуально не изменится, что делает данный метод скрытия данных менее заметным для посторонних.

Для реализации APS в данном реферате рассматриваются три метода.

Механизмы, предложенные первым и вторым методами, во многом схожи, но отличаются способом выполнения перестановки атрибутов. В псевдокоде представлен общий алгоритм метода APS, используемого в этих работах. На вход алгоритма подаются веб-страница H и сообщение M, которое необходимо встроить в страницу. На выходе получается стеганографическая веб-страница H’. Основное отличие между методами состоит в порядке генерации перестановок. В втором методе используется лексикографический порядок (A-Z), тогда как в первом методе, применялся обратный лексикографический порядок (Z-A).

Введем обозначения:

* H: HTML-страница;
* M: Текстовое сообщение для скрытия в H;
* N: Число, представляющее ASCII-коды символов сообщения M;
* T: Тег с набором атрибутов в H, таких как a1, a2, a3, …, am​, где m — количество атрибутов;
* Q: Набор тегов в H, к которым применяется метод перестановки.

Шаги алгоритма:

1. Входные данные: получить HTML-страницу H и текстовое сообщение M для встраивания;
2. Подготовка сообщения: преобразовать сообщение M в большое число N, используя ASCII-коды символов;
3. Инициализация: присвоить M=N;
4. Выбор набора тегов: найти набор тегов Q, к которым можно применить перестановку атрибутов;
5. Начало цикла: пока M≠0:
   1. Для каждого тега T в Q:
      1. Разделить M на факториал m!, где m — количество атрибутов тега T;
      2. Определить число p как остаток от деления M на m!. Число p — число от 0 до m!−1, определяющее порядок атрибутов.
      3. Преобразовать p в конкретную перестановку атрибутов тега T.
      4. Заменить текущий порядок атрибутов T на новый, в соответствии с p.
   2. Обновить значение M.
6. Конец цикла: повторять шаги, пока значение M не станет равным нулю.

Третий метод использует альтернативный подход к стеганографии на основе перестановки атрибутов (APS), отличающийся от первого и второго методов. В данном подходе атрибуты тега переводятся в бинарную строку.

Алгоритм начинается с приема на вход веб-страницы и сообщения, которое необходимо встроить. Он выделяет часть сообщения, которая будет впоследствии скрыта. Это смещение информации необходимо для формирования бинарной строки (BS), представляющего собой бинарное кодирование атрибутов тегов.

Процесс генерации этой бинарной строки происходит через лексикографический анализ первых букв атрибутов. Если первая буква одного атрибута расположена перед первой буквой другого в алфавите, то соответствующий бит в BS становится равным нулю; если же буквы совпадают или расположены иначе, бит принимает значение единицы. Таким образом, каждый атрибут вносит свой вклад в формирование бинарной строки, создавая своеобразный код, который отражает их взаимосвязь.

После этого алгоритм создает множество перестановок для полученной бинарной строки, образуя набор G. Каждая перестановка в этом наборе представляет возможное распределение битов, и среди них будет найдена та, которая совпадает с подсообщением. Если такая перестановка найдена, она заменяет исходный атрибут на веб-странице, что позволяет скрыть сообщение без визуальных изменений.

Таким образом, метод APS не только использует преимущества HTML-структуры, но и создает сложные связи между атрибутами, что делает процесс стеганографии более изощренным и менее заметным для посторонних глаз.

* 1. Метод использования невидимых символов

Когда веб-браузер отображает веб-страницу HTML, он игнорирует невидимые символы, такие как: пробел (ASCII 0x20) и символ табуляции (ASCII 0x19). Подход стеганографии на основе невидимых символов (Invisible Characters Steganography, ICS) использует эту особенность для скрытия сообщений в HTML-файлах без визуальных изменений в отображаемой веб-странице. Этот вид стеганографии также известен как стеганография открытого пространства (Open Space Steganography, OSS).

Идея была впервые предложена в середине 1990-х годов. Хотя внедрение пробельных символов в текстовый файл увеличивает размер файла, эти методы имеют то преимущество, что в любом документе пробельные символы встречаются чаще, чем любые другие символы, поэтому наличие дополнительных пробелов вряд ли вызовет подозрения.

Кроме того, в зависимости от того, как применяется ICS и как просматривается исходный текст, в многих случаях добавление дополнительных пробельных символов не приведет к заметным изменениям в восприятии файла со стороны зрителей. Даже когда дополнительные пробелы видны, наблюдатель вряд ли обратит значительное внимание на их присутствие и не подумает, что эти пробелы представляют собой скрытое сообщение.

ICS использует один или несколько из следующих методов для скрытия сообщений: пространство между двумя последовательными словами, пространство между двумя последовательными предложениями, дополнительные пробелы в конце строк и пространство между двумя последовательными абзацами. Пример использования пробельных символов для скрытия сообщения показан в листинге 1.2.

|  |
| --- |
| <tag></tag> – означает 0,  <tag ></tag > – означает 1 |

Листинг 1.2 – Пример использования пробельных символов

В этом примере, когда теги написаны без вставленного пробельного символа, это отправляет 0, в то время как вставка пробельного символа отправляет 1 (при условии, что сообщения, которые нужно скрыть, представлены в двоичном виде).

Как и в случае APS, существует множество инструментов, доступных для ICS. Например SNOW, Spacemimiс, wbStego4open и WhiteSteg.

С помощью SNOW сообщение встраивается с использованием пробельных символов в конце строк. Когда файл-носитель слишком короткий для размещения сообщения, SNOW встраивает оставшуюся часть сообщения в конце файла-носителя. Перед скрытием каждый символ сообщения кодируется и записывается как тройка (по 3 бита за раз) путем добавления от 0 до 7 пробелов, символ табуляции используется как разделитель для этих троек.

При использовании инструмента стеганографии wbStego4open пробельные символы вставляются между словами и предложениями. С помощью wbStego4open пробельный символ используется для встраивания 1, а символ табуляции — для встраивания 0. В отличие от SNOW, перед скрытием сообщения wbStego4open проверяет, достаточно ли велик файл-носитель для размещения сообщения.

* 1. Метод изменения регистра букв в тегах

Как упоминалось ранее, HTML-теги и атрибуты нечувствительны к регистру. Эта особенность используется в стеганографии переключения регистра букв тегов для скрытия сообщений в HTML-файлах. На рисунке 1.1 представлен пример сокрытия сообщения.

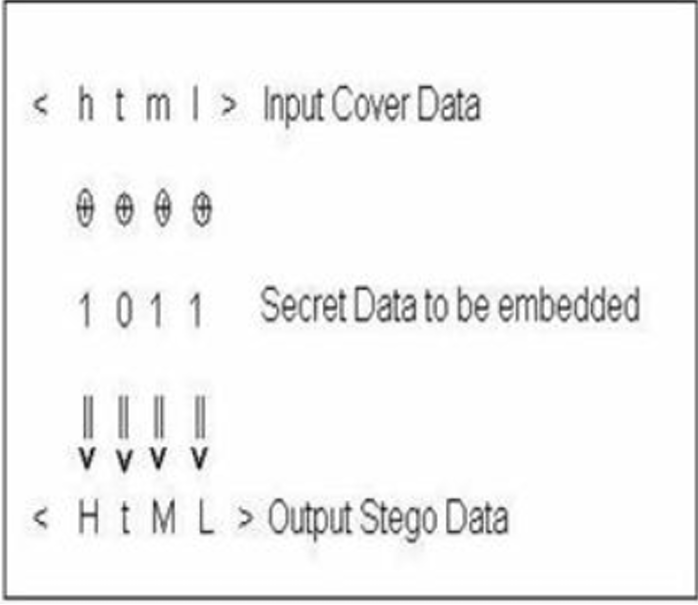


Рисунок 1.1 – Стеганография в HTML с использованием изменения регистра букв тегов

Алгоритмы, представленные в этом разделе, относятся к методу стеганографии, использующему изменение регистра букв в HTML-тегах для скрытия сообщений.

Первый алгоритм, начинается с определения входных данных: веб-страницы, в которую необходимо встроить сообщение, и самого сообщения. Алгоритм извлекает набор тегов, присутствующих на странице, и перебирает их. Если текущий бит сообщения равен 1, первая буква тега переключается на заглавную, что служит индикатором для скрытого сообщения. Если же бит равен 0, никаких изменений не происходит. Этот процесс продолжается, пока все биты сообщения не будут обработаны. Прежде чем встраивать сообщение, алгоритм проверяет, достаточно ли тегов для его размещения, используя формулу для расчета максимальной емкости встраивания.

Второй алгоритм во многом повторяет первый подход, но с важным отличием: он использует только первую букву каждого тега для скрытия информации. Это ограничение приводит к меньшему количеству данных, которые можно встроить, однако принцип проверки максимальной емкости остается прежним.

Общие черты обоих алгоритмов заключаются в том, что они не увеличивают размер HTML-файла, что делает их трудными для обнаружения. Однако использование заглавных и строчных букв в тегах может быть замечено в исходном коде, что создает риск выявления скрытого сообщения.

Таким образом, оба алгоритма представляют собой различные подходы к стеганографии, основанные на изменении регистра в тегах HTML. Первый алгоритм предлагает большую универсальность и возможность скрыть больше информации, в то время как второй метод, хотя и более ограниченный, все же остается эффективным в контексте стеганографии.

* 1. Методы обнаружения стеганографии в HTML-документах

Методы обнаружения стеганографии в HTML-документах можно классифицировать на основе различных подходов, направленных на выявление скрытых сообщений. Существует несколько ключевых техник, используемых для этой цели, которые фокусируются на анализе различных аспектов HTML-кода.

Первый метод связан с анализом атрибутов тегов, использующих перестановки для сокрытия информации, как в случае стеганографии с перестановкой атрибутов (APS). Этот подход предполагает, что изменение порядка атрибутов может не вызвать подозрений, однако для его обнаружения можно применять алгоритмы, которые анализируют структуру и порядок тегов. Например, наличие необычных перестановок в атрибутах может быть сигналом о том, что сообщение было вставлено.

Второй метод заключается в обнаружении невидимых символов, таких как пробелы и табуляции, которые используются для скрытия сообщений в HTML-документах. Этот метод, известный как стеганография с невидимыми символами (ICS), может быть обнаружен с помощью инструментов, которые анализируют наличие и распределение пробелов и табуляций в коде. Если в документе обнаруживается аномально высокое количество этих символов, это может указывать на попытку скрытия информации.

Третий метод сосредоточен на анализе регистра букв в тегах, что характерно для стеганографии с переключением регистра букв в тегах. Поскольку HTML-теги не чувствительны к регистру, изменения в написании тегов (например, использование заглавных букв для кодирования битов сообщения) могут быть неочевидными. Однако использование нестандартного регистра может быть выявлено через алгоритмы, которые ищут несоответствия в написании тегов и их стандартном формате.

Таким образом, методы обнаружения HTML-стеганографии сосредоточены на анализе структуры HTML-документов, выявлении аномалий в использовании атрибутов, символов и регистра, что позволяет определить наличие скрытой информации. Каждый из этих методов требует специфических алгоритмов и инструментов, способных обнаруживать различные формы стеганографии.

2 Методы CSS-стеганографии

Идея встраивания тайного сообщения в CSS-файл, или каскадные таблицы стилей, возникла как продолжение исследований о скрытии информации в XML-данных.

В этом контексте сообщения могут быть вписаны в каждое свойство стиля, аккуратно размещаясь после символов точки с запятой. Используя прием "интервала в конце строки", можно задействовать невидимые пробелы и табуляции для кодирования сообщения, встраивая их в конце строк свойств CSS. Такой подход он позволяет передавать сообщения, оставаясь в безопасности от посторонних глаз.

Предложенная схема передачи сообщений существенно снижает риск их перехвата и подделки злоумышленниками. Идея заключается в использовании криптосистемы с открытым ключом, что делает расшифровку сообщений значительно более сложной задачей для потенциальных нарушителей.

Графическое представление метода представлено на рисунке 2.1.

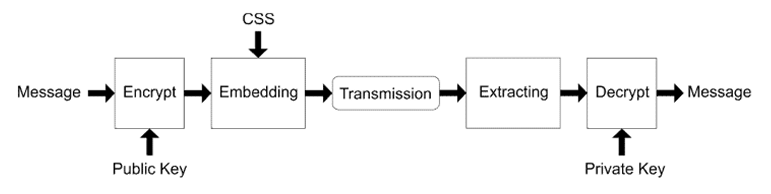


Рисунок 2.1 – Схема внедрения зашифрованного сообщения с CSS

Веб-коммуникации обладают большой пропускной способностью и, следовательно, могут быть использованы для секретной передачи информации. HTML и CSS являются двумя основными, но важными и универсальными инструментами для веб-разработки.

В данном методе предлагается новая схема скрытия информации, которая встраивается через каскадные таблицы стилей (CSS) с использованием интервала в конце строки (EOL) в каждом свойстве стиля CSS, точно после точки с запятой. Перед встраиванием в текст-носитель сообщение сначала шифруется с использованием алгоритма RSA, а затем передается получателю.

В литературе отмечается, что скрытие информации с использованием пробелов имеет большой потенциал, так как люди с трудом могут обнаружить наличие скрытых битов, которые появляются в пробелах между словами. Эта особенность делает данный метод особенно привлекательным для стеганографии, так как скрытые данные не привлекают внимание и могут быть эффективно использованы для передачи секретной информации.

В данных методах один пробел интерпретируется как "0", в то время как два пробела интерпретируются как "1". Однако использование двух пробелов между словами может сделать стеготекст более подозрительным. Одним из лучших способов преодолеть эту проблему является скрытие информации в конце строки.

Данный подход применялся к XML-файлам и дополнительно шифровался с использованием стандарта AES. Однако это все еще не обеспечивает должной безопасности, поскольку известны проблемы с алгоритмами симметричного шифрования, касающиеся распределения ключей. Поэтому в предложенной схеме будет применяться алгоритм асимметричного шифрования.

Далее представлена последовательность действий в процессе скрытия сообщения.

В первом шаге получатель генерирует и отправляет свой открытый ключ отправителю. Этот ключ необходим для шифрования секретного сообщения, которое будет передано. Пример представлен на рисунке 2.2.



Рисунок 2.2 – Генерация закрытого и открытого RSA ключей

На втором шаге отправитель использует алгоритм RSA для шифрования секретного сообщения. После успешного шифрования создается зашифрованный текст, который затем будет встроен в файл CSS. Результатом этого шага является стеганографический файл CSS, как показано на рисунке 2.3.

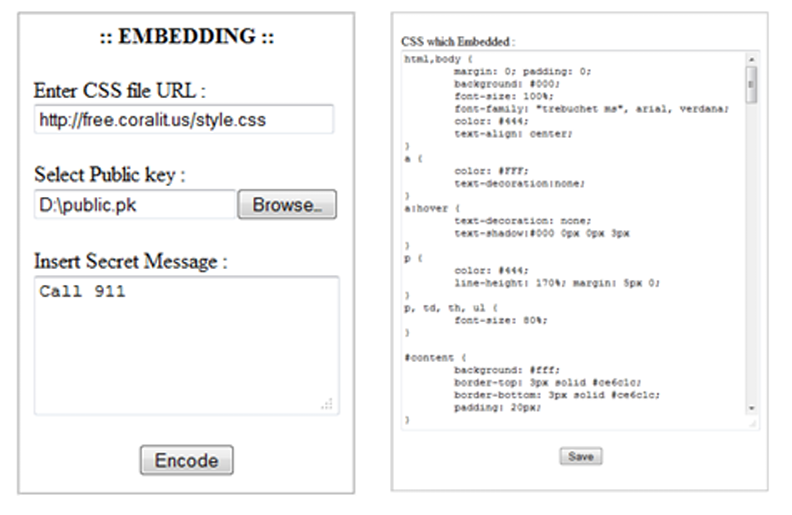


Рисунок 2.3 – Встраивание сообщения в CSS-контейнер

В процессе шифрования зашифрованное сообщение, полученное с помощью алгоритма RSA, переводится в двоичный код. Далее этот двоичный код преобразуется в пробелы, где один пробел представляет собой "0", а табуляция – "1". Затем алгоритм ищет символ точки с запятой в коде CSS и вставляет соответствующее количество пробелов или табуляций сразу после этого символа. Таким образом, информация аккуратно встраивается в конец свойств CSS, что позволяет скрыть ее от глаз посторонних.

На четвертом шаге отправитель отправляет полученный стеганографический файл CSS получателю. В финальном шаге получатель получает этот файл и извлекает зашифрованный текст. Затем он расшифровывает полученный шифротекст, чтобы обнаружить скрытое сообщение. Этот процесс показан на рисунке 2.4.

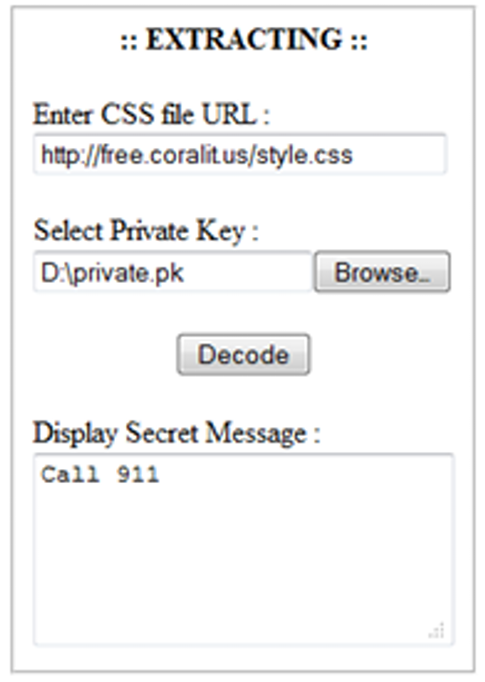


Рисунок 2.4 – Извлечение сообщения из CSS-контейнера

Таким образом, описанная схема позволяет надежно скрывать и передавать информацию, минимизируя риск ее перехвата и изменения.

Новый подход, использующий данную схему, демонстрирует, что стеганографический текст выглядит идентично оригинальному тексту. Применение техники "Конца строки" для процесса встраивания не приводит к заметным изменениям, как показано на рисунке 2.5.

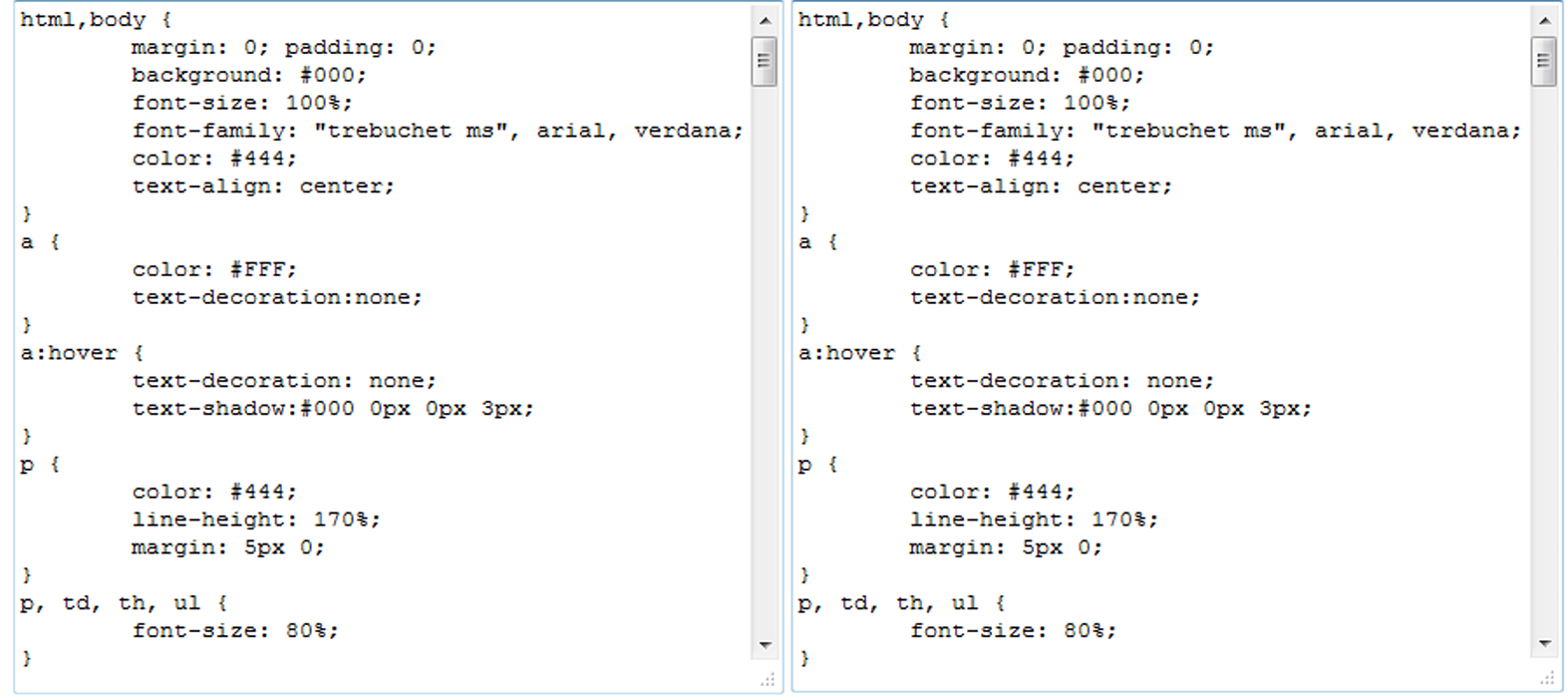


Рисунок 2.5 – Исходный CSS-файл (справа) и CSS-файл с внедренным сообщением (слева)

Использование криптографии с открытым ключом также повышает безопасность передачи скрытой информации. Поскольку CSS-файлы хранятся на веб-сервере, изменения данных третьими лицами невозможны. Однако у этого метода есть слабость – ограниченное количество символов, которое может быть встроено, что зависит от доступного количества точек с запятой.

Методы стеганографии в HTML и CSS представляют собой эффективные подходы к скрытому передаче информации, используя существующие элементы веб-разметки и стилей. Они базируются на ряде уникальных техник, позволяющих интегрировать скрытые данные в текст и стили веб-страниц, сохраняя при этом их визуальное и функциональное целостность.

Однако, несмотря на их преимущества, данные методы также имеют свои ограничения, такие как ограниченное количество скрываемых символов и возможность обнаружения стеганографического контента при внимательном анализе. Таким образом, хотя методы стеганографии в HTML и CSS представляют собой многообещающие инструменты для скрытого обмена информацией, их эффективность зависит от тщательной реализации и защиты передаваемых данных.