

СТЕГАНОГРАФИЧЕСКИЙ МЕТОД НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ ОСОБЕННОСТЕЙ ОТОБРАЖЕНИЯ ЭЛЕМЕНТОВ В ФОРМАТЕ SVG

Н. И. Уласевич

*Белорусский государственный технологический университет,
г. Минск, Беларусь, ulasevich.n@belstu.by*

Изучена возможность внедрения информации в контейнер в формате SVG основанный на правиле порядка отображения элементов на экране. Предложен метод внедрения дополнительных неотображаемых элементов, которые могут быть использованы для скрытой передачи данных или защиты авторского права при советуемой модификации метода

Ключевые слова: стеганография в векторных изображениях; svg-формат; порядок отображения элементов.

В данный момент векторная графика широко применяется для передачи и хранения графических данных, которые могут предоставить уникальную и простую среду для добавления скрытых данных используя стеганографические методы. Данный формат хранит объекты, которые описывают изображение и объект может описать как несколько пикселей, так и всё изображение. К таким объектам относят точки, сплайны, кривые Безье, окружности, многоугольники, эллипсы и текст посредством которых можно описать практически любое изображение.

Формат SVG (Scalable Vector Graphics) является наиболее простым для внедрения информации так-как использует структуру документа, основанную на XML, что позволяет легко внедрить дополнительные объекты не затрагивая существующие. Как пример, рассматривая объект `<path>` используемый для создания сложных графических объектов. В данном объекте в атрибуте `"d"` содержится последовательность команд и параметров. Команды в данном объекте обозначаются буквами: верхнего и нижнего регистра где заглавные буквы обозначают абсолютные координаты, а команды со строчной буквы означают относительные координаты, которые считаются от последней точки.

Способы стеганографического преобразования можно свести к двум большим группам. К первой группе можно отнести методы, которые изменяют существующие значения объектов как пример с использованием метода аналогичного LSB, что приводит к почти незаметному визуальному изменению изображения. Ко второй группе можно отнести методы, которые внедряют дополнительные точки или объекты, которые не влияют на изображение. Как пример внедрение дополнительной точки в объект `"polyline"` лежащей на изначальной линии не приведёт к изменению изображения.

Рассматривая особенности отображения изображений в формате SVG можно выделить два из них, которые имеют потенциал использования для сокрытия данных. Первая особенность – это свойство атрибута `viewbox` элемента `<svg>` согласно которому в данном атрибуте устанавливаются координаты области просмотра векторного изображения что возможно использовать для сокрытия данных [1]. Вторая особенность это правило отображения «элементы, которые отрендерились позднее, отображаются поверх предыдущих элементов. Чем дальше вниз элемент по порядку рендеринга и как следствие по коду то тем более видимым он будет» [2]. При анализе данного правила можно сделать вывод о возможности добавления объектов в файл, которые впоследствии будут не отображены с точки зрения стороннего наблюдателя или будут скрыты под другими элементами согласно правилу отображения. Также как следствие из этого правила необходимо, чтобы элемент, под которым будут расположены другие объекты, имел некоторую заливку. Рассматривая простейший пример части кода в формате `svg`, представленный на рис. 1 можно увидеть два объекта прямоугольник и

текст. Прямоугольник имеет координаты верхнего левого угла по координатам 20,20 с шириной 150 и высотой 50 и объект текст, который расположен по координатам 205,50 с размером 10. Также стоит заметить, что область просмотра никак не препятствует полному отображению объектов.

```
<svg version="1.0" xmlns="http://www.w3.org/2000/svg"
viewBox="0 0 250 250" width="250" height="250">
  <rect x="20" y="20" width="150" height="50" style="fill:none;stroke-
width:5;stroke:rgb(155,155,155)" />
  <text x="205" y="50" font-size="10" fill="black">Web Conf</text>
</svg>
```

Рис. 1. Пример кода в формате SVG

Исходя из ранее рассмотренного правила можно сделать вывод о возможности скрыть объект текста за прямоугольником, но при рассмотрении кода прямоугольника можно также обнаружить отсутствие заливки у центральной части что ограничивает перекрываемые области только обводкой прямоугольника. Для проверки изменим позиции теста на 18,22 с размером текста 5. На рис. 2 можно рассмотреть результат без изменения порядка отображения элементов.



Рис. 2. Пример отображения без изменения порядка рендеринга

Как уже упоминалось, чтобы скрыть данный текст его необходимо расположить раньше по коду, чем объект прямоугольника. Пример кода представлен на рис. 3.

```
<svg version="1.0" xmlns="http://www.w3.org/2000/svg"
viewBox="0 0 250 250" width="250" height="250">
  <text x="18" y="22" font-size="5" fill="black">Web Conf</text>
  <rect x="20" y="20" width="150" height="50" style="fill:none;stroke-
width:5;stroke:rgb(155,155,155)" />
</svg>
```

Рис. 3. Код со скрытым текстом “Web conf”

Описанный выше метод применим для практически для любого файла в формате svg. Для сокрытия текста целесообразно использовать часто используемый объект в данном файле и располагать данные элементы в зависимости под фигурами с заливкой. В зависимости от объекта возможны использования различных методов для внедрения информации и также возможно различное соотношение внедренной полезной информации к размеру внедряемого объекта.

Библиографические ссылки

1. Николайчук А. Н., Урбанович П. П. Стеганографический метод на основе использования особенностей отображения элементов в формате SVG // Труды БГТУ. Сер. 3, Физико-математические науки и информатика. 2023. № 1 (266). С. 64-70.
2. SVG учебное руководство. Начало работы [Электронный ресурс]. URL:https://developer.mozilla.org/ru/docs/Web/SVG/Tutorial/Getting_Started (дата обращения: 15.03.2024).