

ОБРАБОТКА И ПЕРЕДАЧА ИНФОРМАЦИИ PROCESSING AND TRANSMISSION OF INFORMATION

УДК 003.026:004.92

Н. И. Уласевич, Н. А. Жилияк

Белорусский государственный технологический университет

СТЕГАНОГРАФИЧЕСКИЙ МЕТОД НА ОСНОВЕ ВСТРАИВАНИЯ ДОПОЛНИТЕЛЬНЫХ ЭЛЕМЕНТОВ ПОД ЗАКРАШЕННЫЕ УЧАСТКИ В ИЗОБРАЖЕНИЯХ В ФОРМАТЕ SVG

Рассмотрены ключевые характеристики файлов векторной графики, включая уникальную структуру SVG и принципы формирования графических объектов в этом формате. Рассмотрено правило визуального ранжирования для отображения графических объектов. Содержимое файла SVG, отображаемое пользователю, ограничено областью просмотра. Однако при описании фигур в области просмотра необходимо учитывать возможность частичного наложения или полного перекрытия одного элемента другим графическим элементом. На основе анализа вышеперечисленных особенностей параметров отображения и правила отображения геометрических фигур в данной статье обоснован и описан новый стеганографический метод и реализующие его алгоритмы встраивания (извлечения) скрытых данных при использовании векторных изображений формата SVG в качестве стеганоконтейнера. Одним из важных типов элементов таких контейнеров является элемент эллипса. В качестве модифицируемых элементов используются координаты центра фигуры. Внедрение тайной информации осуществляется путем формирования множества ключевых точек, которые затем применяются при построении эллипсов. При этом координаты ключевых точек остаются в пределах существующих фигур, что обеспечивает сохранение визуальной целостности изображения. Метод может использоваться для скрытой передачи данных и защиты электронного контента от несанкционированного применения.

Ключевые слова: стеганография, векторные изображения, SVG-формат, эллипс, правило ранжирования.

Для цитирования: Уласевич Н. И., Жилияк Н. А. Стеганографический метод на основе встраивания дополнительных элементов под закрашенные участки в изображениях в формате SVG // Труды БГТУ. Сер. 3, Физико-математические науки и информатика. 2025. № 2 (296). С. 76–82.
DOI: 10.52065/2520-6141-2025-296-10.

M. I. Ulasevich, N. A. Zhilyak

Belarusian State Technological University

A STEGANOGRAPHIC METHOD BASED ON EMBEDDING ADDITIONAL ELEMENTS UNDER THE COLORED AREAS IN SVG FORMAT IMAGES

The key features of vector graphics files are considered, including the unique structure of SVG and the principles of forming graphic objects in this format. The rule of visual ranking for displaying graphic objects is considered. The content of SVG file displayed to the user is limited to the viewing area. However, when describing figures in the viewing area it is necessary to take into account the possibility of partial overlapping or full overlapping of one element by another graphic element. On the basis of the analysis of the above mentioned features of display parameters and the rule of displaying geometric figures, this paper substantiates and describes a new steganographic method and algorithms of embedding (extracting) hidden data when using vector images of SVG format as a steganographic container. One of the important types of elements of such containers is an ellipse element. The coordinates of the center of the figure are used as modifiable elements. Implementation of secret information is carried out by forming a set of key points, which are then applied in the construction of ellipses. In this case, the coordinates of key points remain within the existing figures, which ensures the preservation of visual integrity of the image. The method can be used for covert data transmission and protection of electronic content from unauthorized use.

Keywords: steganography, vector graphics, SVG, ellipse, ranking rule.

For citation: Ulasevich N.I., Zhilyak N. A steganographic method based on embedding additional elements under the colored areas in SVG format images. *Proceedings of BSTU, issue 3, Physics and Mathematics. Informatics*, 2025, no. 2 (296), pp. 76–82 (In Russian).

DOI: 10.52065/2520-6141-2025-296-10.

Введение. В настоящий момент информация, обладающая рядом свойств и особенностями использования, приобрела статус одного из важнейших экономических ресурсов. В связи с этим обеспечение безопасности информационных ресурсов является приоритетной задачей в сфере информационных технологий. Данные ресурсы требуют защиты от несанкционированного доступа, хищения, модификации или уничтожения, что обуславливается их высокой ценностью.

Одним из перспективных направлений решения данной проблемы является применение методов цифровой стеганографии, позволяющей осуществлять скрытую передачу данных посредством их внедрения в информационные объекты-носители (стеганоконтейнеры), таким образом, чтобы внесенные изменения не могли быть обнаружены при визуальном отображении [1]. Одним из таких стеганоконтейнеров являются файлы векторной графики, которые в последнее время становятся объектом изучения специалистов в области стеганографии и широко используются при создании web-ресурсов для иконок и логотипов, интерактивных элементов и графиков.

Существует множество форматов, которые хранят векторную графику, например, PDF, AI, EPS, CDR и SVG. Формат SVG (Scalable Vector Graphics) обладает уникальными свойствами из-за использования структуры, основанной на XML-документе, который является текстовым файлом. Из-за данной особенности к таким файлам можно применить методы текстовой стеганографии, методы, используемые для файлов разметки, включающие в себя методы подмены и перестановки атрибутов или метод замены регистров тегов [2–4]. Также возможно использование методов, разработанных для изображений, включающих в себя незначительное изменение существующих координат фигур или изменения цвета изображения [5].

Формат SVG обладает рядом преимуществ по сравнению с другими графическими форматами, включая динамическое отображение данных и интерактивность. Однако расширение сферы его применения в коммерческих web-приложениях для обмена данными или публикация контента сопровождается рисками несанкционированного копирования и распространения информации. Это создает серьезные проблемы в области защиты авторских прав для разработчиков и владельцев веб-ресурсов [6–9].

В данной статье предлагается новый стеганографический метод внедрения (извлечения) данных с использованием векторных изображений, а также описываются алгоритмы его реализации.

Основная часть. Несмотря на то, что SVG-формат является текстовым, его принципиальное значение определяется возможностью предоставлять графические элементы через систему математических координат и параметров. Основными графическими примитивами являются линия, прямоугольник, окружность эллипс, многоугольники и ломаные линии. Они в структуре документа реализуются через систему тэгов с полным набором атрибутов, которые описывают фигуру, учитывая ее особенности. Например, для описания эллипса необходимо использовать тег `<ellipse>` и установить значения атрибутов: sx – координата центра фигуры по оси X , sy – координата центра фигуры по оси Y , rx – величина радиуса по горизонтальной оси [3, 4]. Для создания сложных объектов применяется тег `<path>`, который использует атрибут `-d`, содержащий серию команд и параметров для определения линий фигуры.

Содержимое SVG-файла формируется из графических элементов, порядок отображения которых определяется их расположением в документе. Согласно правилу ранжирования или правилу отображения, элементы, которые были отображены позднее, могут отображаться поверх существующих элементов. Порядок отображения устанавливается расположением элементов в файле.

Для понимания и наглядности действия правила рассмотрим изображение, содержащее три элемента с разной заливкой с частичным наложением элементов.

На рис. 1 показано содержимое изображения из трех прямоугольников. Так как расположение элементов идет в порядке уменьшения размера, то все три фигуры будут отображены пользователю. Исходя из размеров и координат первой фигуры и из того, что она является первой в последовательности отображения, она может быть перекрыта остальными прямоугольниками. Вторая фигура в порядке отображения, исходя из размеров и координат, перекрывает первый прямоугольник. Третья фигура одновременно частично перекрывает первый и второй прямоугольники, которые были отображены ранее (рис. 2).

```

<defs>
<style type="text/css">
<![CDATA[
.str0 {stroke:#2B2A29;stroke-width:2;stroke-miterlimit:22.9256}
.fil1 {fill:#FFED00}
.fil0 {fill:#E31E24}
.fil2 {fill:#BBC274}
]]>
</style>
</defs>
<g id="Слой_x0020_1">
<metadata id="CorelCorpID_0Corel-Layer"/>
<rect class="fil0 str0" x="200" y="200" width="600" height="400"/>
<rect class="fil1 str0" x="250" y="250" width="300" height="300"/>
<rect class="fil2 str0" x="300" y="300" width="400" height="200"/>
</g>

```

Рис. 1. Часть текста файла SVG

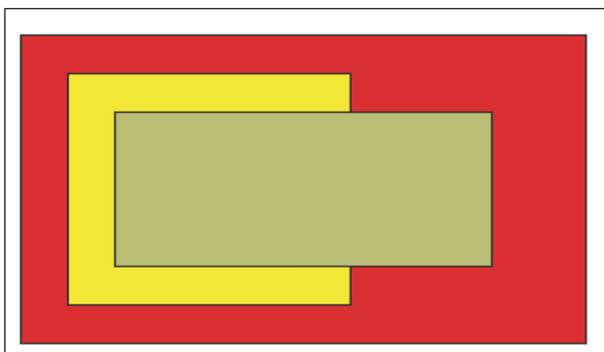


Рис. 2. Файл Svg с тремя прямоугольниками

Содержимое изображения на рис. 3 представлено такими же прямоугольниками, как на рис. 1. Однако во втором случае были добавлены три эллипса с координатами, принадлежащими области первого прямоугольника и расположенные таким образом, чтобы они отображались раньше прямоугольников (рис. 4).

```

<g id="Слой_x0020_1">
<metadata id="CorelCorpID_0Corel-Layer"/>
<ellipse class="fil3 str0" cx="750" cy="400" rx="40" ry="40"/>
<rect class="fil0 str0" x="200" y="200" width="600" height="400"/>
<ellipse class="fil5 str0" cx="500" cy="400" rx="40" ry="40"/>
<rect class="fil1 str0" x="250" y="250" width="300" height="300"/>
<ellipse class="fil4 str0" cx="300" cy="400" rx="40" ry="40"/>
<rect class="fil2 str0" x="300" y="300" width="400" height="200"/>
</g>

```

Рис. 3. Часть текста файла SVG с дополнительными эллипсами

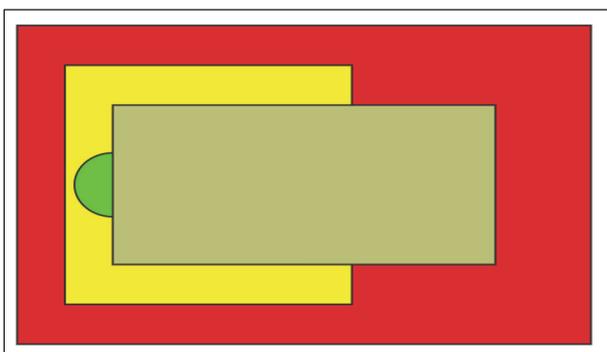


Рис. 4. Файл Svg с дополнительными эллипсами

Таким образом, два эллипса, описанные в содержимом файла, не отображены на экране, а третий эллипс был отображен частично. Как следствие, фигуры со сплошной заливкой скрывают элементы, расположенные в пределах их координат, эти элементы были раньше в порядке отображения.

Кроме того, стоит упомянуть о возможности отсутствия заливки, что приведет к отображению всех скрытых элементов за прямоугольниками (рис. 5). Данная особенность также может быть использована для простейшей стеганографии с простым добавлением элементов без цвета.

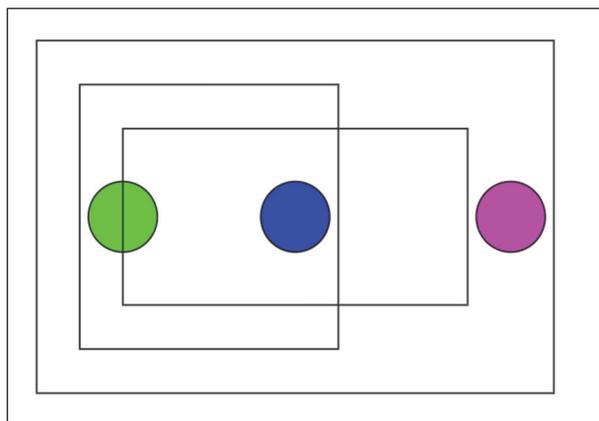


Рис. 5. Файл Svg без заливки прямоугольников

На основе использования вышеперечисленных особенностей отображения и видимости элементов предлагается новый метод встраивания данных в векторные изображения. В качестве встраиваемого элемента выбран эллипс, так как имеется возможность задействовать значения координат центра фигуры и значения величин радиусов для хранения данных.

Для усложнения метода встраивания информации вместо непосредственного внедрения информации в координаты предлагается использовать значения угла наклона прямой и расстояния между встраиваемой точкой и некоторой ключевой координатой. Исходя из предположения, что ключевая точка, от которой будут производиться вычисления расстояния и угла наклона, может быть значительно удалена от встраиваемой точки, целесообразно использовать дробные части вычисляемых величин для хранения данных как части, обладающей большей энтропией.

При небольшом числе элементов с заливкой и высоком разрешении файла процесс выбора каждой встраиваемой точки может занять значительное время. Для ускорения работы алгоритма предполагается использовать дополнительную точку, которая будет находиться на некотором удалении от геометрического центра фигуры.

Для получения проверяемых алгоритмом точек было решено использовать обход точек по

квадрату. Данный метод позволяет уменьшить объем вычислений по сравнению с традиционной круглой спиралью, кроме того, указанный алгоритм гарантирует равномерное покрытие области точками.

В качестве ключей в алгоритме было решено использовать значения двух точек и значения позиций, на которых будет расположена информация. Первое значение – это значение сдвига $D(\Delta x, \Delta y)$ относительно точки центра фигуры и относительно которой будет производиться получение точек для проверки. Вторая точка $C(x, y)$ – это точка, от которой производится вычисление значений угла и расстояния до встраиваемой точки H . Третьим значением ключа является массив элементов $L(x_1, x_2, \dots, x_n)$ с позициями цифр в дробной части расстояния от H до C , в которых будет содержаться информация. Третьим значением ключа является массив элементов $A(x_1, x_2, \dots, x_n)$ с позициями цифр в дробной части угла от H до C , в которых хранится информация.

Алгоритм внедрения дополнительных эллипсов может быть представлен в виде следующей последовательности шагов.

Шаг 1. Считываем файл и сохраняем код графических элементов.

Шаг 2. Преобразуем графические элементы в многоугольники (обозначим M_i).

Шаг 3. Вычисляем геометрические центры многоугольников (обозначим как $C_i(x, y)$).

Шаг 4. Преобразуем полученное секретное сообщение I во внедряемую цифровую последовательность (обозначим ее S).

Преобразование ведется посимвольно с преобразованием каждого символа.

Шаг 5. Создаем регулярные выражения R_L и R_A для значения угла и расстояния.

Регулярные выражения используются для проверки точки H .

Шаг 6. Устанавливаем значение удаленности r .

Шаг 7. Устанавливаем значения $g = 1$, где $g \in [1; 4 \cdot r^2)$.

Шаг 8. Вычисляем значения сдвига для вычисления проверяемой точки $F_g(r)$.

Шаг 9. Устанавливаем значения $i = 0$;

Шаг 10. Выбираем текущий многоугольник M_i .

Шаг 11. Получаем значение приведенного центра $P_i(x, y)$.

$$P_i = C_i + D. \quad (1)$$

Шаг 12. Вычисляем значение проверяемой точки $H(x, y)$.

$$H = P_i + F_g. \quad (2)$$

Шаг 13. Проверяем, принадлежит ли точка H многоугольнику M_i , если не перейти к п. 10 с увеличением i на единицу, иначе к п. 15.

Шаг 14. Вычисляем значения угла наклона прямой из точки H к точке C и оси X (обозначим как a).

$$a = \arctan((P.x - C.x), (P.y - C.y)). \quad (3)$$

Шаг 15. Вычисляем дистанцию от H к точке C (обозначим как l).

$$l = ((P.x - C.x)^2 + (P.y - C.y)^2)^{1/2}. \quad (4)$$

Шаг 16. Проверяем соответствие значения a регулярному выражению R_A и соответствие l регулярному выражению R_L . В случае несоответствия переходим к п. 10 с увеличением значения i на единицу.

Шаг 17. Добавляем позицию точки H в свойство HP объекта M_i .

Шаг 18. Если длина S не равна 0, переходим к п. 19, иначе к п. 22.

Шаг 19. Удаляем первые n символов из S , где n соответствует общему числу позиций в L и A .

Шаг 20. Создаем новые регулярные выражения R_L и R_A .

Шаг 21. Сортируем объекты M в порядке возрастания числа встроенных точек HP и переходим к п. 8 с увеличением значения g на единицу.

Шаг 22. Формируем эллипсы для всех точек, хранящихся в M_iHP .

Шаг 23. Сформируем SVG-файл.

Для извлечения сообщения I из стеганоконтейнера необходимо использовать следующий алгоритм.

Шаг 1. Считываем файл и сохраняем код графических элементов.

Шаг 2. Преобразуем графические элементы в многоугольники (обозначим M_i).

Шаг 3. Вычисляем геометрические центры многоугольников (обозначим как $C_i(x, y)$).

Шаг 4. Обнаруживаем эллипсы, расположенные под фигурами.

Шаг 6. Распределяем найденные эллипсы по M_i в зависимости от расстояния от центра эллипса до P_i .

Шаг 7. Сортируем найденные эллипсы в порядке возрастания расстояния от P_i до центра эллипса.

Шаг 8. Сортируем найденные эллипсы в порядке возрастания угла P_i до центра эллипса.

Шаг 9. Извлекаем данные по ключевым позициям A и L .

Шаг 10. Извлекаем данные по ключевым позициям A и L .

Шаг 11. Сформируем цифровую последовательность.

Шаг 12. Преобразуем сообщение из цифровой последовательности п. 11 в символьную строку.

Для демонстрации предложенного метода нами разработано консольное приложение. В качестве языка программирования был выбран C#.

Кроме того, использовалась библиотека System.Xml.XmlDocument для работы с XML-файлами, а также были созданы классы для работы с фигурами, что упрощает взаимодействие со значениями их атрибутов. Были созданы классы Embedder и Deembedder для реализации функций внедрения и извлечения сообщения. После запуска приложения пользователь может ввести путь к файлу, сообщение, параметры ключевой точки, параметры сдвига центральной точки и позиции значений угла и расстояния, используемые для внедрения информации. Результатом работы является файл с осажденной информацией. Набор эллипсов со встроеным сообщением «bgtu 2025» с использованием одной значащей позиции в значении угла и расстояния представлен на рис. 6.

```
<g id="hidden">
<ellipse class="filf str5" cx="3039962" cy="19313278" rx="40569" ry="38456"/>
<ellipse class="filf str5" cx="17124162" cy="17700571" rx="29945" ry="28428"/>
<ellipse class="filf str5" cx="4584712" cy="24829531" rx="24565" ry="37098"/>
<ellipse class="filf str5" cx="19148876" cy="11813661" rx="28209" ry="22072"/>
<ellipse class="filf str5" cx="12258134" cy="21732941" rx="22146" ry="26563"/>
<ellipse class="filf str5" cx="8004599" cy="18040862" rx="31311" ry="30519"/>
<ellipse class="filf str5" cx="14827320" cy="12409160" rx="24472" ry="46991"/>
<ellipse class="filf str5" cx="6422284" cy="9805992" rx="31748" ry="34145"/>
<ellipse class="filf str5" cx="6422282" cy="9805992" rx="46560" ry="45034"/>
</g>
```

Рис. 6. Набор эллипсов со встроеным сообщением

В представленном варианте использовались параметры ключевой точки $C(15000000, 15000000)$ и значение параметра сдвига $D(200000, 400000)$ и установлена пятая позиция для значений угла и расстояния для хранения информации. Для первой точки, представленной на рис. 6, значение угла наклона $a = 160,168626$ и значение расстояния $l = 12714042,475417$. Следовательно, скрытой информацией является комбинация цифр «21».

Представленный метод был протестирован для выбора оптимального числа позиций, используемых для встраивания информации. В табл. 1 даны замеры времени вычисления координат точек при встраивании информации отдельно в значение угла и расстояния.

Таблица 1
Зависимость времени вычисления точек от числа значимых позиций

Число позиций	Время при использовании значения угла, мс	Время при использовании значения расстояния, мс
1	1,45	8,45
2	4,36	38,63
3	31,28	151,02
4	205,85	1038,12
5	2340,79	8658,35
6	17218,64	78519,05

Исходя из полученных данных можно сделать вывод об оптимальности использования от

одной до четырех позиций для встраивания информации. Кроме того, важно отметить, что время, необходимое для вычисления точек, при использовании только значений угла меньше, чем при использовании значений расстояния.

Метод также был протестирован при использовании комбинации из позиций в значениях угла и расстояния, в которые осуществлялась встраивание информации. В данном случае на максимальное число позиций в одном значении было наложено ограничение, равное двум. Результаты замеров для части комбинаций представлены в табл. 2.

Таблица 2
Время вычисления точек при использовании комбинации значимых позиций

Позиции в значении угла	Позиции в значении расстояния	Время вычисления, мс
5	7	2,57
4, 9	7	14,01
3	9	15,74
9	3, 5	17,79
8, 9	4, 5	75,54
3, 4	9	93,74
8, 9	3, 9	78519,05

Полученные значения позволяют сделать вывод, что неверный выбор позиции может значительно увеличить время вычисления координат точек. Кроме того, исходя из среднего времени вычисления координат точек в комбинации позиций оптимально использовать от двух до четырех позиций.

Заключение. Цифровое изображение, содержащее большое число геометрических фигур с разнообразными стилями, являются хорошим стеганографическим контейнером. Данные файлы позволяют внедрить большой объем информации и устойчивы к изменениям.

Формат SVG позволяет добавлять элементы, которые не видны пользователю из-за перекрытия другими элементами. Расположение данных элементов в больших файлах также является вариативным, так как они должны быть добавлены не позднее, чем отображаемый элемент. Таким образом можно скрыть разнообразные элементы, которые не будут обнаружены при просмотре. На этой основе разработан описанный в статье стеганографический метод. Ввиду особенностей данного метода возможно скрыть практически неограниченный объем информации при высокой площади фигур с заливкой путем резкого увеличения объема файла.

Данный метод можно использовать для внедрения скрытой информации в векторные изображения. Кроме того, при незначительных доработках данный метод может быть применен

для защиты авторского права путем внедрения в векторное изображение водяных знаков.

Представляет научный и практический интерес дальнейшее расширение исследований в

данной предметной области, связанных с оценкой пропускной способности канала и уточнением уровня стеганографической стойкости метода к различным типам атак.

Список литературы

1. Урбанович П. П. Защита информации методами криптографии, стеганографии и обфускации. Минск: БГТУ, 2016. 220 с.
2. Blinova E., Shutko N. The use of steganographic methods in SVG format graphic files // Proc. of the 10th Intern. Conf. New Electrical and Electronic Technologies and their Industrial Implementation. Lublin, 2015. P. 45.
3. Text Steganography utilizing XML, HTML And XHTML Markup Languages / S. Imran [et al.] // International Journal of Computational Geometry & Applications. 2017. No. 3. P. 99–116.
4. Almutairi A. A Comparative Study on Steganography Digital Images: A Case Study of Scalable Vector Graphics (SVG) and Portable Network Graphics (PNG) Images Formats // (IJACSA). 2018. Vol. 9, no. 1. P. 170–175.
5. Topology-Preserving Watermarking of Vector Graphics / S. Huber [et al.] // International Journal of Computational Geometry & Applications. 2014. No. 1. P. 61–86. DOI: 10.1142/s0218195914500034.
6. Николаичук А. Н., Урбанович П. П. Стеганография в векторных изображениях // 73-я науч.- техн. конф. учащихся, студентов и магистрантов: сб. науч. работ, Минск, 18–23 апр. 2022 г. Минск: БГТУ, 2022. С. 947–949.
7. Zhou X. and Pan X. Watermark-Based Scheme to Protect Copyright of SVG // International Conference on Computational Intelligence and Security. 2006. P. 1199–1202. DOI: 10.1109/ICCIAS.2006.295245.
8. Badr Almutairi. A New Steganography Method for Scalable Vector Graphics (SVG) Images Based On An Improved LSB Algorithm // International Journal of Computer Science and Network Security. 2019. Vol. 19, no. 10. P. 99–104.
9. Kuznetsov A., & Kononchenko G. Steganographic methods in vector graphics // Radiotekhnika. 2021. No. 2 (205). P. 32–41. DOI: 10.30837/rt.2021.2.205.03.

References

1. Urbanovich P. P. *Zashchita informatsii metodami kriptografii, steganografii i obfuskatsii* [Information protection by cryptography, steganography and obfuscation methods]. Minsk, BG TU Publ., 2016. 220 p. (In Russian).
2. Blinova E., Shutko N. The use of steganographic methods in SVG format graphic files. *New Electrical and Electronic Technologies and their Industrial Implementation. Proc. of the 10th Intern. Conf.* Lublin, 2015, p. 45.
3. Imran S., Khan A., Ahmad B. Text Steganography utilizing XML, HTML And XHTML Markup Languages. *International Journal of Computational Geometry & Applications*, 2017, no. 3, pp. 99–116.
4. Almutairi A. A. Comparative Study on Steganography Digital Images: A Case Study of Scalable Vector Graphics (SVG) and Portable Network Graphics (PNG) Images Formats. *International Journal of Advanced Computer Science and Applications*, 2018, vol. 9, no. 1, pp. 170–175.
5. Huber S., Held, M., Meerwald, P., & Kwitt, R. Topology-preserving watermarking of Vector Graphics. *International Journal of Computational Geometry & Applications*, 2014, no. 1, pp. 61–86. DOI: 10.1142/s0218195914500034.
6. Nikolaichuk A. N., Urbanovich P. P. Steganography in vector images. *73-ya nauchno-tekhnicheskaya konferentsiya uchashchikhsya, studentov i magistrantov: sbornik nauchnykh rabot* [73rd Scientific and Technical Conference of students, undergraduates and undergraduates: collection of scientific papers]. Minsk, 2022, pp. 947–949 (In Russian).
7. Zhou X. and Pan X. Watermark-Based Scheme to Protect Copyright of SVG Data. *International Conference on Computational Intelligence and Security*, 2006, pp. 1199–1202. DOI: 10.1109/ICCIAS.2006.295245.
8. Badr Almutairi. A New Steganography Method for Scalable Vector Graphics (SVG) Images Based On An Improved LSB Algorithm. *International Journal of Computer Science and Network Security*, 2019, vol. 19, no. 10, pp. 99–104.
9. Kuznetsov A., & Kononchenko G. Steganographic methods in vector graphics. *Radiotekhnika*, 2021, no. 2 (205), pp. 32–41. DOI: 10.30837/rt.2021.2.205.03.

Информация об авторах

Уласевич Николай Иванович – магистрант кафедры информационных систем и технологий. Белорусский государственный технологический университет (ул. Свердлова, 13а, 220006, г. Минск, Республика Беларусь). E-mail: koliaulasevich@gmail.com

Жиляк Надежда Александровна – кандидат технических наук, доцент, доцент кафедры информационных систем и технологий. Белорусский государственный технологический университет (ул. Свердлова, 13а, 220006, г. Минск, Республика Беларусь). E-mail: Ghilyak@belstu.by

Information about the authors

Ulasevich Mikalai Ivanovich – Master’s degree student, the Department of Information Systems and Technologies. Belarusian State Technological University (13a Sverdlova str., 220006, Minsk, Republic of Belarus). E-mail: koliaulasevich@gmail.com

Zhilyak Nadezhda Aleksandrovna – PhD (Engineering), Associate Professor, Assistant Professor, the Department of Automation of Production Processes and Electrical Engineering. Belarusian State Technological University (13a Sverdlova str., 220006, Minsk, Republic of Belarus). E-mail: Ghilyak@belstu.by

Поступила 25.04.2025