
ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ

THEORETICAL FOUNDATIONS OF COMPUTER SCIENCE

УДК 004.056:004.42

СТЕГАНОГРАФИЧЕСКИЙ МЕТОД ВСТРАИВАНИЯ ИДЕНТИФИКАТОРА В ПРОСТРАНСТВЕННЫЕ ДААННЫЕ ЭЛЕКТРОННОЙ КАРТЫ

Е. А. БЛИНОВА¹⁾, И. Ю. СТАШЕВСКАЯ¹⁾, П. П. УРБАНОВИЧ^{1), 2)}

¹⁾Белорусский государственный технологический университет,
ул. Свердлова, 13а, 220006, г. Минск, Беларусь

²⁾Люблинский католический университет им. Иоанна Павла II,
ал. Рацлавицке, 14, 20-950, г. Люблин, Польша

Электронные карты, представляющие собой набор компьютерных файлов, являются основной формой представления географической информации конечному потребителю. Важной задачей выступает защита карт от несанкционированного использования или модификации. Приводится описание стеганографического метода встраивания

Образец цитирования:

Блинова ЕА, Стасhevская ИЮ, Урбанович ПП. Стеганографический метод встраивания идентификатора в пространственные данные электронной карты. *Журнал Белорусского государственного университета. Математика. Информатика*. 2023;1:76–87 (на англ.).
<https://doi.org/10.33581/2520-6508-2023-1-76-87>

For citation:

Blinova EA, Stashevskaya IYu, Urbanovich PP. A steganographic method of embedding an identifier into the spatial data of an electronic map. *Journal of the Belarusian State University. Mathematics and Informatics*. 2023;1:76–87.
<https://doi.org/10.33581/2520-6508-2023-1-76-87>

Авторы:

Евгения Александровна Блинова – старший преподаватель кафедры информационных систем и технологий факультета информационных технологий.

Ирина Юрьевна Стасhevская – магистрант кафедры информационных систем и технологий факультета информационных технологий. Научный руководитель – Е. А. Блинова.

Павел Павлович Урбанович – доктор технических наук, профессор; профессор кафедры информационных систем и технологий факультета информационных технологий¹⁾, профессор²⁾.

Authors:

Evgenia A. Blinova, senior lecturer at the department of information systems and technologies, faculty of information technologies.

evgenia.blinova@belstu.by
<https://orcid.org/0000-0001-7245-8721>

Irina Yu. Stashevskaya, master's degree student at the department of information systems and technologies, faculty of information technologies.

irina.stashevskaya@belstu.by
<https://orcid.org/0000-0001-8889-056X>

Pavel P. Urbanovich, doctor of science (engineering), full professor; professor at the department of information systems and technologies, faculty of information technologies^a, and professor^b.
p.urbanovich@belstu.by
<https://orcid.org/0000-0003-2825-4777>

невидимого цифрового водяного знака в пространственные данные электронных карт в целях защиты авторских прав на карты, обеспечения целостности последних или доказательства их подлинности. Метод применим к электронным картам в форматах Shapefile и GeoJSON. Он основывается на размещении дополнительных точек в пространственных объектах (полигонах), составляющих электронную карту. В данном случае карта служит контейнером, а координаты точек являются элементами ключевой информации стеганографического преобразования. При этом устанавливается связь между пространственными объектами карты, что обеспечивает их целостность. Рассматриваются алгоритмы прямого и обратного стеганографического преобразования. Описывается разработанный интернет-сервис *StegoMap* для реализации размещения и извлечения невидимого цифрового водяного знака на основе предложенного метода.

Ключевые слова: электронные карты; стеганография; авторское право; цифровой водяной знак; пространственные данные; формат GeoJSON; формат Shapefile.

Благодарность. Работа выполнена при финансовой поддержке государственной программы научных исследований «Цифровые и космические технологии, безопасность человека, общества и государства» на 2021–2025 гг.

A STEGANOGRAPHIC METHOD OF EMBEDDING AN IDENTIFIER INTO THE SPATIAL DATA OF AN ELECTRONIC MAP

E. A. BLINOVA^a, I. Yu. STASHEVSKAYA^a, P. P. URBANOVICH^{a, b}

^aBelarusian State Technological University, 13a Sviardlova Street, Minsk 220006, Belarus

^bThe John Paul II Catholic University of Lublin, 14 Raclawickie Alley, Lublin 20-950, Poland

Corresponding author: E. A. Blinova (evgenia.blinova@belstu.by)

Electronic maps (e-maps), which are a set of computer files, are the main form of representation of geographic information to the end user. An important task is protecting e-maps from unauthorised use or modification. This paper describes the steganographic method of embedding an invisible digital watermark into the spatial data of e-maps for protecting copyright, ensuring integrity of maps or proving their authenticity. The method is applicable to e-maps in the Shapefile and GeoJSON formats. It is based on the placement of additional points in spatial objects (polygons) of the e-map. In this case the e-map is used as a carrier object, and point coordinates are elements of the key information of the steganographic transformation. This establishes a relationship between the spatial objects of the e-map, which ensures their integrity. Algorithms for direct and inverse steganographic transformations are considered. The developed *StegoMap* Internet service for implementing the placement and extraction of an invisible digital watermark on the basis of the proposed method is described.

Keywords: e-maps; steganography; copyright; digital watermark; spatial data; GeoJSON format; Shapefile format.

Acknowledgements. This work was carried out with the financial support of the state program of scientific research «Digital and space technologies, security of man, society and state» for 2021–2025.

Introduction

Modern geoinformation technologies based on the methods and tools for studying, modelling and analysing relationships in geosystems, cartographic analysis and modelling, are the most important areas of geoinformatics [1]. These technologies are of great importance for the solution of a number of social problems. The level of such importance largely depends on the delimitation of access to a geographic information system (GIS), and the information included in the system [2–4].

Electronic cartographic images or electronic maps (e-maps) are the main and rather expensive form of representation of geographic information to the end user [5]. E-maps are a set of computer files containing cartographic images in a vector or raster format that can be rendered in the GIS. E-maps are widely used in environmental, social and economic applications, such as navigation, various land management tasks (creation of a land cadastre for real estate accounting), agromonitoring and equipment monitoring, creating a communications accounting system, as well as business planning. They are also used in military or security related applications [1–5].

The vector e-maps preparation requires significant costs and efforts. At that such methods are used as digitising images of raster maps, as well as space or aerial photography with adjustments if necessary. The value of these e-maps makes protecting them necessary not only to prevent an attacker from illegal use but also to prevent the use of e-map in a situation related to various security aspects.

Due to this, the task of developing effective means of protecting e-maps from illegal copying or use is relevant. The International Hydrographic Organization recommended standards for protecting e-map information are contained in publication [6]. This document defines the security structures and operational procedures that must be followed to ensure the proposed e-maps protection scheme, and also includes specifications to enable the creation of consistent systems for dealing with the data of the e-maps. This document recommends using cryptography as the primary tool for securing the e-map.

Since maps are often used in an open form, in accordance with the recommendations of publication [6] to solve the above problem, it is justified to use the methods that provide the placement of secret, copyrighted information, performing the function of a digital watermark (DWM), using steganographic transformations [7–10].

The main directions and possibilities of using steganographic methods for protecting e-map data presented in vector graphics formats are formulated in works [11–13]. The main features of the method for embedding watermarks in SVG vector image files based on adding points to Bezier curves are described in articles [14; 15]. A similar approach can be used to protect e-maps. The latter is the subject of the research in this paper.

Theoretical substantiation of the proposed steganographic method

E-map can be considered from the logical and physical sides. Logically it consists of a number of spatial objects that have additional characteristics (attributes). Spatial objects are often grouped into layers. A layer represents geographic data on a specific topic, such as roads, land plots, building footprints, etc. An e-map consists of an ordered collection of layers; often the map has only one layer. In addition to the spatial description the map contains a set of attributes that characterise the spatial areas, for example, soil type, building height, etc. The description of spatial regions is usually written in one of the following formats: WKT (well known text), WKB (well known binary), GML (geography markup language) or GeoJSON. Moreover, all formats, except the last one, are subsets of the XML markup language. Attributes are numeric or textual characteristics.

Physically the e-map is a file or a set of files linked together. Cartographic data processing is done in a GIS such as *ArcGIS* or *MapInfo*, each of which provides its own file format. In addition, most modern DBMS allow you to store and process spatial data in the appropriate database formats.

E-map Shapefile storage format. Let us consider the Shapefile format used by *ArcGIS*, which is one of the standards for vector e-maps. A map in the Shapefile format is a set of files that store the spatial and attribute values of objects. Let us analyse, for example, an e-map which contains a single spatial element, namely, a polygon representing the contour of the border of the Republic of Belarus. The set of files is shown in fig. 1.





Имени	Дата изменения	Тип	Размера
 Граница_Беларуси_полигон.dbf	20.12.2017 10:08	Файл "DBF"	1 КБ
 Граница_Беларуси_полигон.prj	20.12.2017 10:07	Файл "PRJ"	1 КБ
 Граница_Беларуси_полигон.shp	20.12.2017 10:07	Файл "SHP"	525 КБ
 Граница_Беларуси_полигон.shx	20.12.2017 10:07	Файл "SHX"	1 КБ

Fig. 1. A set of e-map files in the Shapefile format

The main file is in the .SHP format, which contains information about the spatial object. The file consists of a fixed length header and one or more variable length entries. The .SHX index file format makes a link between .DBF and .SHP files, and also defines the type of the spatial data binding. The most important are the three required files, namely, .SHP, .SHX and .DBF, that must be saved in the same directory. In addition to the main files, there may be additional files, including .SBN and .SBX spatial index files, which allow us to accelerate the processing of spatial data, or .AIH and .AIN attribute table index files that allow to speed up the attribute lookups.

In the .DBF file format the attribute information of geometric objects is saved. The .PRJ file specifies information about the spatial coordinate reference system. Figure 2, a, shows the display of the *Border_Belarus_polygon.shp* file. It is not possible to view the contents of a file directly. Figure 2, b, reflects the contents of the *Border_Belarus_polygon.dbf* file with a list of spatial data which shows that there is only one spatial object in the file that has no attributes. Figure 2, c, shows the contents of the *Border_Belarus_polygon.prj* file which specifies the GCS_WGS_1984 reference system.

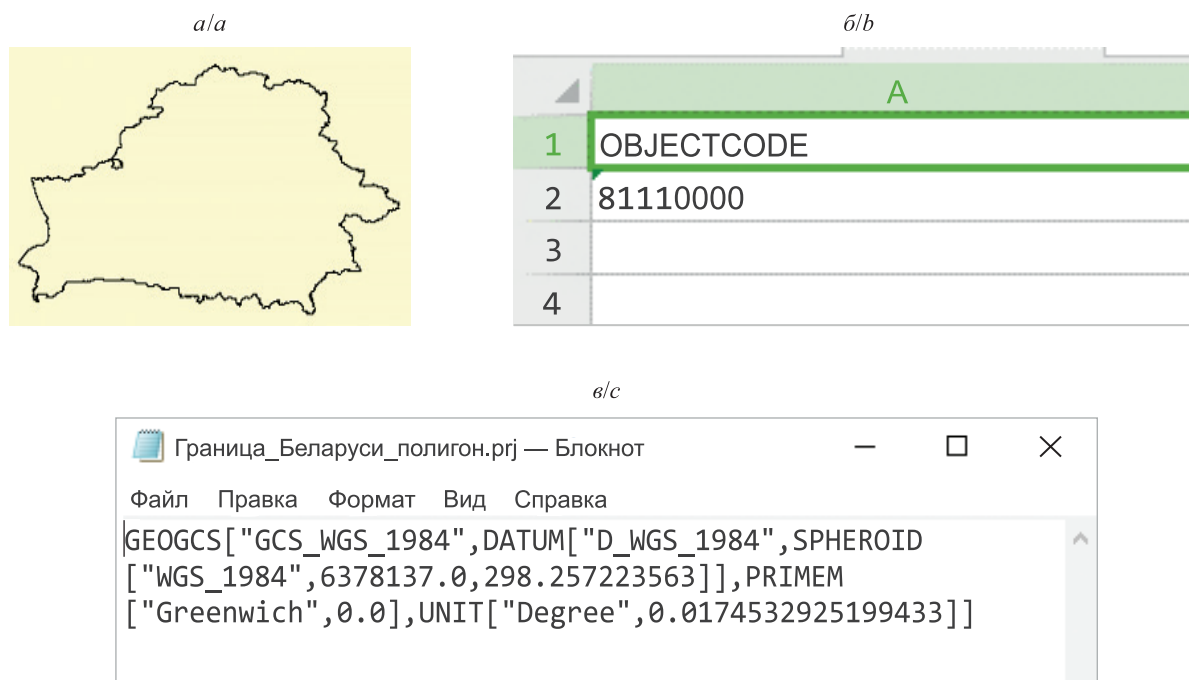


Fig. 2. Contents of e-map files in the Shapefile format

The .SHP format allows to store the following types of geometric objects: points (multipoints), lines (polylines), polygons, etc. When storing polygons, only the coordinates of the vertices are stored. Virtual straight lines between adjacent polygon vertices are named as edges. The edges are not described in any way, they are only displayed by the GIS software. A single file can only store objects of the same type. Each entry in the .SHP file can also have several attributes, such as name, height, terrain type, etc. Spatial objects descriptions can be displayed in the WKT format which is similar to path descriptions in the SVG files. Thus the techniques developed for embedding the hidden data in e-map files can be applied to the SVG files and vice versa. Figure 3, for example, shows the content of one spatial element of the e-map in the .SHP format. The element is a set of two polygons. It is important that according to the requirements of WKT format the last vertex of the polygon must coincide with the first vertex of the polygon. The specified coordinates depend on the used coordinate system which is the same for all objects of the e-map. The current map uses the spatial reference identifier SRID = 4326 which corresponds to the WGS_1984 geographic coordinate system.

```

MULTIPOLYGON (((24.121052730010657 52.537050075893916,
24.121040931537145 52.536891401261236, 24.121043226657839
52.536916149411823, 24.121052730010657 52.537050075893916))),
(((24.12108609988805 52.53756315704824, 24.121075745353686
52.537374418290831, 24.121079227484003 52.537423489707422,
24.12108609988805 52.53756315704824)))
    
```

Fig. 3. Spatial element of the e-map content

Vector e-map GeoJSON format. The GeoJSON format is also a way to describe spatial data. A GeoJSON object can generally be represented as a collection (*feature collection*). This object consists of the spatial object itself (*geometry*) and attributes (*properties*), as well as a set of key – value pairs named as properties. Each GeoJSON spatial object must have a property *type*. The value of this property is a string containing the GeoJSON object type. The GeoJSON format supports geometric types similar to WKT format: a point, a line, a polygon, as well as sets of these objects. Next, a mandatory object property is specified – *coordinates*, which are determined by an array of numbers. The order of the elements must be as follows: *x, y, z* (for data in a rectangular coordinate system – east offset, north offset, altitude; for data in a geographic coordinate system – longitude, latitude, altitude). In addition to the spatial description, optional properties of the type of key – value pairs can be included in the object. An example of a fragment of a file in the GeoJSON format is shown in fig. 4. This file is a spatial description of some objects in Baranovichy city (Republic of Belarus), and the fragment of the file demonstrates the structure of the GeoJSON format in the form of key – value pairs where the list of all vertices of a polygon that defines a spatial figure is sequentially set for the *coordinates* key.

```

        "type": "Feature",
        "properties": {
            "name": "UpPart",
            "town": "Baranovichi",
            "square": "15.9152"
        },
        "geometry": {
            "type": "Polygon",
            "coordinates" : [
                [
                    [
                        25.9552001953125,
                        53.11154464430509
                    ],
                    [
                        26.009445190429684,
                        53.140180585580396
                    ],
                    [
                        26.055450439453125,
                        53.156858919018774
                    ],
                    [
                        26.039657592773438,
                        53.15891752333123
                    ],
                    [
                        26.052017211914062,
                        53.16571821716968
                    ],
                    [
                        26.037940979003906,
                        53.17106127943977
                    ],
                    ]
                ]
            ]
        }
    ]
}
    
```

Fig. 4. GeoJSON file fragment example

Description of the proposed steganographic method. Now let us return to the fundamental task formulated above: the user needs not only to confirm his copyright on the e-map, but also to ensure the integrity of the data (both spatial and attributive ones). In this regard we will further consider the most important features of the steganographic method described in work [14] and adapt it to the GeoJSON format. The main idea of the steganographic method is that additional points set on edges of the spatial objects or lines are not visualised due to the way GIS displays. Therefore, you can set any number of additional points on segments of a spatial figure and place hidden copyright information in their location. Let us consider a little example. Table 1 gives a description of three spatial objects in the form of the WKT format description. The column «*Id*» lists a single attribute of the spatial area, and the column «Spatial object» lists this object’s description: in this case, all objects are polygons.

Table 1

Spatial figures with the same display

<i>Id</i>	Spatial object
1	POLYGON ((10 10, 10 20, 20 20, 20 15, 10 10))
2	POLYGON ((10 10, 10 15, 10 20, 20 20, 20 15, 10 10))
3	POLYGON ((10 10, 10 20, 15 20, 20 20, 20 15, 10 10))

Table 1 suggests that spatial figures have different descriptions, but they are all displayed in the same way. Figure 5 shows the display of each of the polygons from table 1 in the GIS. The additional vertex (10, 15) from polygon 2 is not displayed because it is on the same line with two neighboring vertices (10, 10) and (10, 20), and the additional vertex (15, 20) from polygon 3 is not displayed because it is on the same line with two neighboring vertices (10, 20) and (20, 20).

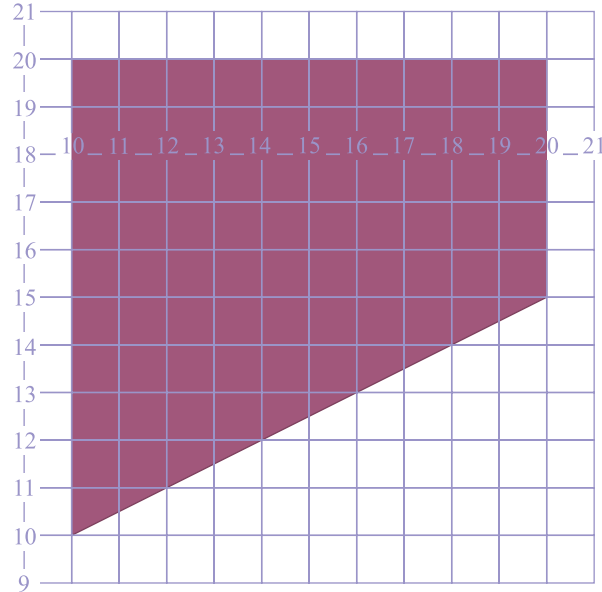


Fig. 5. Display of each of the polygons from table 1

Further only polygons will be considered. For real maps spatial areas are described using *polygon* or *multi-polygon* objects, and each of them can have hundreds, thousands and tens of thousands of vertices. For example, table 2 shows the number of vertices for the polygons that form the spatial areas of a small map of the water protection zones of the Republic of Belarus. In total, the map consists of 363 spatial objects.

Table 2

**Distribution of peaks in the map
of water protection zones of the Republic of Belarus**

The number of peaks	The quantity of objects	The percentage of total quantity of objects
101–1000	232	64
1001–10 000	124	34
>10 000	7	2

The main idea of the steganographic method proposed in work [14] is to set additional points on the edges of polygons in a certain ratio λ , $\lambda \in [0; 1]$, as shown in fig. 6. The additional vertex will have coordinates (X, Y) :

$$X = \frac{x_t + \lambda x_{t+1}}{1 + \lambda}, Y = \frac{y_t + \lambda y_{t+1}}{1 + \lambda},$$

where (x_t, y_t) , (x_{t+1}, y_{t+1}) are the vertices of the edge on which the additional point is set.

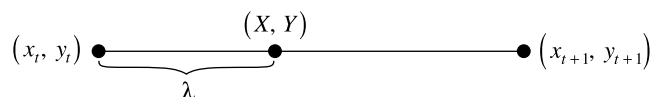


Fig. 6. Additional vertex (X, Y) set in ratio λ

Let us assume that the user has a GeoJSON (or Shapefile) format e-map in which all spatial objects are represented as polygons with a set of additional attributes, the first of which (Id) is the number of the object and it is named the *key attribute*. We will consider the e-map as a sequence of polygons in the order of the Id attribute as shown, for an example, in fig. 7.

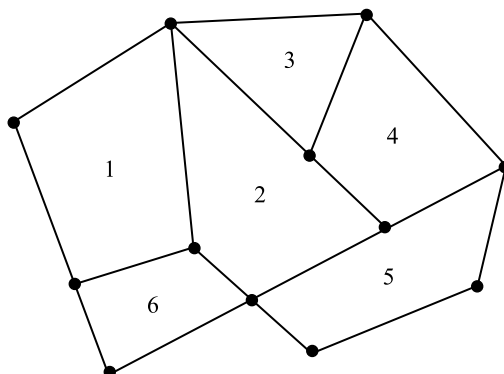


Fig. 7. E-map as a sequence of polygons

Let us also assume that the e-map consists of N spatial objects, each of which we will denote as G_i , $i \in [1; N]$. The object G_i is a structure and consists of the following set of fields:

- Id_i is the key attribute, the number of the spatial object G_i ;
- A_{i1}, \dots, A_{im} are additional attributes of the object G_i of the e-map;
- g_i is the description of the initial spatial region i .

The user also has to generate DWM as the identifier I to protect copyright, and ensure the integrity of the e-map. This could be the user's first and last name, the current date, another unique identifier, and so on as a text.

Let us define the date of implementation as D and the data on the e-map owner as O . The D and O values make the identifier I of the e-map owner, and I consists of two parts (variable D and constant O): $I = \{D, O\}$.

For each polygon, starting from the first one, a control value h_i should be calculated that ensures the data integrity in this polygon. This control value verifies that the polygon and its attributes are unchanged. As h_i it is proposed to use the hexadecimal value of the hash function H^1 from the concatenation of the spatial description of the polygon g_i , its attributes A_{i1}, \dots, A_{im} , user identifier $I = \{D, O\}$, and the number of polygons N , $i \in [1; N]$:

$$h_i = H^1(g_i \parallel \{A_{i1}, \dots, A_{im}\} \parallel I \parallel N).$$

For this method we suggest to use the MD5 hash function, the result of which is a 128-bit string written in hexadecimal notation.

To hide the DWM (I) it is necessary to convert the control value h_i to a set of additional vertices P_i embedded on the edges of the polygon g_i . Such a set P_i of the polygon g_i will be called as *secret vertices*. We write h_i in the form

$$h_i = \{h_{1i} h_{2i} \dots h_{32i}\},$$

where h_{ji} , $j \in [1; 32]$, is the next digit of the control value h_i in hexadecimal notation.

We propose to place the vertices P_i as follows: each vertex of the set P_i is a set in ratio

$$\lambda = \begin{cases} \frac{h_{ji}}{16}, & h_{ji} \neq 0, \\ \frac{1}{32}, & h_{ji} = 0. \end{cases}$$

For example, for the control value $h_i = \{68EF4\dots\}$ the first additional vertex is located in the ratio $\frac{6}{16}$, the second – $\frac{8}{16}$, the third – $\frac{14}{16}$, etc.

It is also necessary to select a set of edges E_i of the spatial object g_i , to which the vertices P_i will be added. Hiding each h_i in the polygon g_i will require 32 edges. In this method, we propose to use a pseudo-random edge selection principle with a random number generator. Further we will name such edges as *secret*. We propose generating its own list of vertices P_i for each polygon g_i . Figure 8 shows the installation of some of the secret vertices (vertices P_i are marked in red colour, edges E_i , selected as secret, are marked in blue colour).

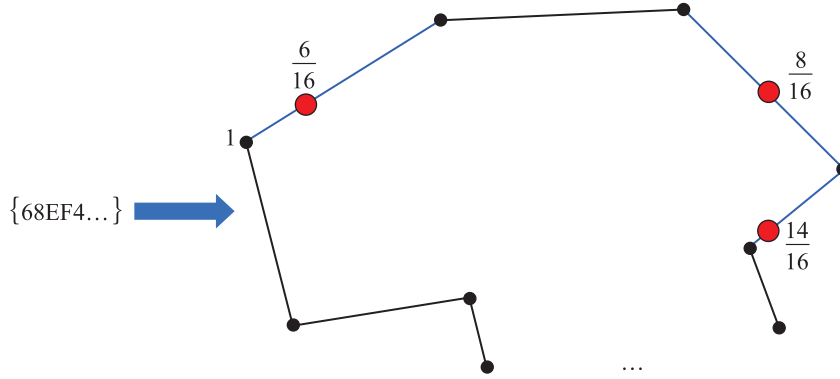


Fig. 8. Setting secret vertices P_i on the polygon g_i

After setting the vertices P_i the description of the spatial object g_i will change with the transformation function:

$$F_i^P: g_i \rightarrow gs_i. \quad (1)$$

For the reverse steganography transformation it is necessary to know the location of the set of secret vertices P_i in order to extract the identifier I from the spatial object gs_i . For this, it is proposed to add another set of vertices $R_i, i \in [2; N]$, to the next polygon e-map (g_{i+1}) from the data of the location of which it is possible to extract the numbers of secret edges $P_i, i \in [1; N]$. We will further name such vertices R_i as *control vertices*, and the edges T_i , on which they are installed, as *control edges*. The location of the control vertices R_i must be known to the author (owner) of the e-map and unknown to other (unauthorised) users. We propose to define the control edges T_i of the polygon g_{i+1} as the value H_i of the hash function H^2 from the concatenation of the identifier I and the key attribute Id_i of the spatial object:

$$H_i = H^1(I \parallel N \parallel Id_i).$$

Let us define the set $K_1 = \{D, O, N, H^1, H^2\}$ as a *key of the first kind*.

It is suggested to use the SHA-2 hash function, the result of which is a string with a length of 512 bits or 128 hexadecimal digits, as a function H^2 . Let us write H_i in the form $H_i = \{t_{1i}, t_{2i}, \dots, t_{128i}\}$, where $t_{ji}, j \in [1; 128]$, is a hexadecimal digit from 0 to F.

Each set of control edges $T_i, i \in [1; N]$, is formed as follows:

$$T_i = \{t_{1i}, t_{1i} + t_{2i}, t_{1i} + t_{2i} + t_{3i}, \dots, t_{1i} + t_{2i} + \dots + t_{128i}\}. \quad (2)$$

Figure 9 shows how the control edges T_i are located depending on the value of H_i :

$$\begin{aligned} t_{1i} = 3 &\Rightarrow T_{1i} = 3, \\ t_{2i} = 4 &\Rightarrow T_{2i} = 7, \\ t_{3i} = 1 &\Rightarrow T_{3i} = 8 \text{ and so on.} \end{aligned}$$

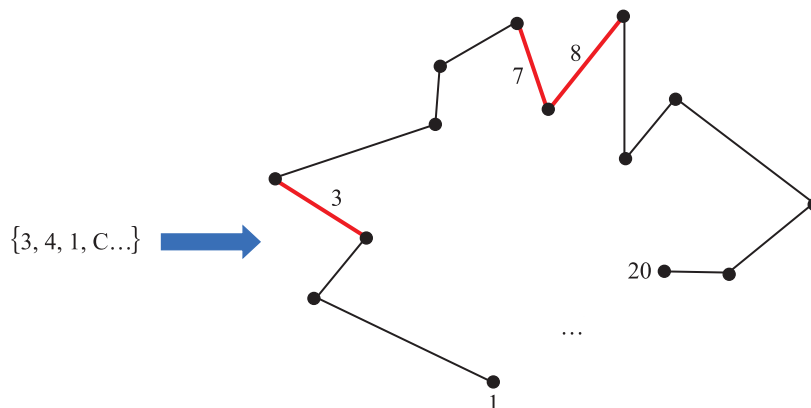


Fig. 9. Selection of a set of control edges T_i on a polygon g_{i+1}

The set of vertices P_i is known from polygon g_i . It is necessary to hide this set on control edges T_i . Since the number of secret vertices P_i is not always a single digit, it is necessary to separate the vertex numbers from each other by some symbol. To do this, we write P_i as a sequence φ separating the vertices with the number 0:

$$\varphi_i = [P_{1i}0P_{2i}0\dots0P_{32i}]. \quad (3)$$

For example, the set of secret vertices $P = \{1, 4, 17, 25, \dots\}$ becomes the sequence $\varphi = [104017025\dots]$. Here we can formulate a constraint on the sets P_i : vertices are generated randomly but cannot contain vertices marked with the number 0. It is also necessary that $l(H_i) > l(h_i)$, where l is the hash length.

On control edges T_i control points $R_i, i \in [2; N]$, will be set at a certain ratio. We propose to put additional points R_i in the ratio from the beginning of the edge according to the value of the next digit of the sequence to the number 16, and if the next digit is 0, then in relation to $\frac{1}{32}$. For the example of control edges T_i in fig. 9 the

sequence of the ratio would look like this: $\left[\frac{1}{16}, \frac{1}{32}, \frac{4}{16}, \frac{1}{32}, \frac{1}{16}, \frac{7}{16}, \frac{1}{32}, \frac{2}{16}, \frac{5}{16}, \dots \right]$. Figure 10 shows how a part of the control vertex set R_i is set.

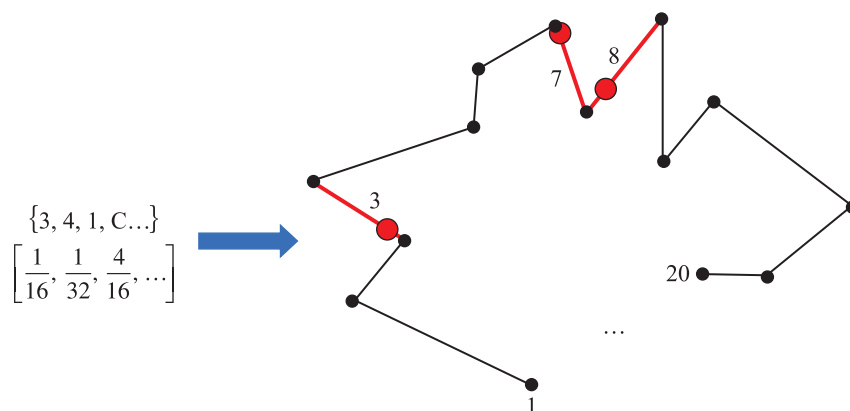


Fig. 10. Setting a set of control points R_i on a polygon g_{i+1}

Having set control vertices R_i the description of the spatial object g_{i+1} will change, so let us define the description of the new spatial object as g'_{i+1} , and the transformation function F_i^R as

$$F_i^R : g_{i+1} \rightarrow g'_{i+1}. \quad (4)$$

Next, for the region g'_{i+1} we perform the transformation F_{i+1}^P and obtain a spatial object g_{i+1} .

Thus each polygon g_i , except for the first one, goes through two transformations: first F_i^R , when control vertices R_i are added to the polygon, and then F_i^P , when secret vertices P_i are added to the polygon. The first polygon goes through only the F_1^P transformation. Moreover, the secret vertices P_i of the polygon g_i are written in the current polygon g_i , and its control vertices R_i – in the next polygon g_{i+1} , which allows us to sequentially connect all e-map objects like a blockchain system. After transforming the last polygon g_N we get the polygon g_N^S :

$$F_N^P : (F_{N-1}^R : g_{N-1}) \rightarrow g_N^S. \quad (5)$$

Let us name a set of secret vertices P_N the key of the second kind K_2 , that is

$$K_2 = \{P_{1N}, P_{2N}, \dots, P_{32N}\}.$$

After successive transformation of all polygons g_i we obtain an e-map with a placed identifier I and a key of the second kind K_2 . Then the resulting e-map can be distributed by the user. Using this method, all spatial regions g_i of the e-map become sequentially connected with each other. This allows you to control the integrity of the e-map and provide copyright protection.

So, according to formulas (4) and (5) for each spatial object g_i two transformations are performed sequentially:

$$F_{i-1}^R : g_i \rightarrow g'_i, \quad i > 1,$$

$$F_i^P : g'_i \rightarrow g_i^S.$$

Extraction of the identifier I occurs sequentially, starting from the last e-map object. The key of the second kind K_2 , contains the secret vertices P_N of the polygon g_{S_N} . Let us check which of the following values $\left\{ \frac{1}{32}, \frac{1}{16}, \frac{2}{16}, \dots, \frac{15}{16} \right\}$ represent the ratio in which the vertices $\{P_{1N}, P_{2N}, \dots, P_{32N}\}$ share edges between the vertices $\left\{ (P_{1N-1}, P_{1N+1}), (P_{2N-1}, P_{2N+1}), \dots, (P_{32N-1}, P_{32N+1}) \right\}$. Consistently saving the obtained values, we obtain the value h_N . Then we remove the secret vertices P_N of the polygon g_{S_N} and get the polygon g'_N . Then, using the key of the first kind K_1 , we determine the set of control edges T_{N-1} for the previous polygon $g_{S_{N-1}}$. We find the control vertices R_{N-1} and for each of them we determine at which ratio it is installed on the corresponding edge. The sequence of the ratio forms the sequence φ from which we obtain a set of secret vertices P_{N-1} of the polygon $g_{S_{N-1}}$.

Let us delete the control vertices R_{N-1} and obtain the polygon g_N . For the resulting polygon g_N we compare the previously obtained value h_N with the value of the hash function H^1 which we calculate for the obtained spatial object g_N . If the map has not been changed, then $h_N = H^1(g_N)$. After it we move on to the previous spatial object $g_{S_{N-1}}$, and continue comparing the obtained values of h_i with the values of the hash function H^1 for each spatial object g_i . We do this for all spatial objects.

Results and discussion

The proposed steganographic method is designed to hide identification information (invisible DWM) into electronic cartographic images. The peculiarity of the method is that it allows you to analyse the contents of the e-map not as a single, indivisible entity, but as a set of interrelated objects or areas. For each of these objects you can define a certain control parameter, with which you can control the integrity of the e-map. The method allows you to chain all e-map spatial objects with each other, placing the control value of the previous object in the next object, similar to the concept of blockchain. Thus, if any spatial area or the value of any of its attributes changes accidentally or intentionally, a control value mismatch will indicate an attempt to change the integrity of the e-map and perform evidentiary procedures related to copyright issues.

A brief outline of the method is as follows.

- Step 1:** representation of the e-map as an ordered set of polygons.
- Step 2:** building a list of vertices and edges of the current polygon.
- Step 3:** selection of random (secret) edges of the current polygon.
- Step 4:** calculation of the control value from the current polygon.
- Step 5:** setting secret vertices to secret edges defined in step 3 in the ratio defined in step 4.
- Step 6:** rebuilding a list of vertices and edges of the current polygon according to formula (1).
- Step 7:** making the sequence φ according to formula (3).
- Step 8:** move to the next polygon to add control vertices.
- Step 9:** building a list of vertices and edges of the current polygon.
- Step 10:** getting a list of control edges of the current polygon from formula (2).
- Step 11:** setting control vertices of the current polygon defined in step 10 in the ratio defined in step 7.
- Step 12:** go to step 2 if there are still unprocessed polygons.

Upon completion of the algorithm, the user receives a special key that allows him to check the e-map for integrity.

To apply the method, it is necessary for the number of vertices of any spatial object of the e-map to be more than 2048. This value was evaluated as a minimum of the number of control edges needed. If $l(H^2) = 512$ and it is assumed that each polygon can contain no less edges than is used in formula (3), so the number of vertices is $16 \cdot 128 = 2048$ (here first multiplier (16) is the max hexadecimal digit, and second multiplier (128) is the quantity of digits). If there are polygons with an insufficient number of vertices in the e-map, then such polygons must either be combined with other polygons with similar values of attribute columns into *multipolygon* objects or deliberately complicated by adding points to random edges.

The method involves placing secret vertices that hide the control value on random edges of spatial objects, which makes it difficult to create an algorithm for extracting them. The key of the first kind provided by the author of the e-map consists of a constant and variable parts, serves to extract such vertices, and must be kept secret. The key of the second kind, obtained as a result of hiding the DWM, allows you to sequentially obtain the original e-map, control the integrity of all objects and confirm the authorship of the card owner. The key information generated by the owner of the e-map for the steganographic system created on the basis of this method allows the owner of the key to perform all the necessary identification and evidentiary procedures like the use of keys in cryptographic systems.

To implement the steganographic method of placing hidden copyright labels on e-map files and checking the integrity of the e-map, the *StegoMap* software product was developed [16]. Using the application, information about the owner of the e-map is embedded in the map, and the polygon attributes are controlled. The application converts a Shapefile format to the GeoJSON format to create a steganographic container. The application is implemented using a microservice and client-server architecture in the form of the *StegoMap* Internet service. For implementation a set of classes was created, which includes methods for hiding information in the polygons of an e-map and extracting that hidden information, as well as several auxiliary classes. To place the hidden label, a hash value is calculated from the identifier of the e-map owner. This identifier is generated when a user registers in the system using standard Angular utilities. The application is implemented using the Java Spring Boot technology.

The user selects an e-map in the .SHP format. As soon as the files are selected from the hard drive, they are converted to the GeoJSON format using the Aspose.GIS library. Then the GeoJSON format file is sent to the Internet service, where the steganographic label is added, and the file with hidden information is sent to the client. The resulting steganographic container (carrier of a secret author's message) is placed in the database and displayed in the tab «Maps» of the application, where the user can upload it. When the map is uploaded, GeoJSON is converted by Aspose.GIS library back into Shapefile that the user can distribute.

To check the map for authenticity the user must click the button «Check map» on the main page and then select the Shapefile that he wants to check in the window that opens. The result of the check from the Internet service is sent to the client. If the map has been changed, the user sees information about these changes.

Conclusions

A steganographic method is considered that allows embedding and extracting hidden messages when using e-maps as steganographic containers. This method can be applied to the Shapefile or GeoJSON formats of the e-map. The method is based on the sequential modification of the parameters of spatial objects and it can be used to confirm the authorship and control the integrity of the e-map. The method provides the random placement of additional secret vertices containing the user's identifier, and the control of this placement by calculating the control value also using the identifier. The user's identifier consists of a constant and variable parts. All spatial areas of the e-map become connected similarly to the blockchain principle.

Further research is supposed to focus on hiding such an excess steganographic label in the e-map which allows us not only to determine the fact of violation of the integrity of the e-map but also to at least partially restore the spatial objects of the e-map.

Библиографические ссылки

1. Sheppard E. GIS and society: towards a research agenda. *Cartography and Geographic Information Systems*. 1995;22(1):5–16. DOI: 10.1559/152304095782540555.
2. Longley PA, Goodchild MF, Maguire DJ, Rhind DW. *Geographic information science and systems*. 4th edition. Hoboken: Wiley; 2015. XVI, 505 p.
3. Mukherjee F. GIS use by an urban local body as part of e-governance in India. *Cartography and Geographic Information Science*. 2018;45(6):556–569. DOI: 10.1080/15230406.2018.1448304.
4. Rzeszewski M. Geosocial capta in geographical research – a critical analysis. *Cartography and Geographic Information Science*. 2018;45(1):18–30. DOI: 10.1080/15230406.2016.1229221.
5. Захаров МС, Кобзев АГ. *Картографический метод и геоинформационные системы в инженерной геологии*. 3-е издание. Санкт-Петербург: Лань; 2021. 116 с.
6. IHO data protection scheme [Internet]. Monaco: International Hydrographic Organization; 2020 [cited 2022 November 15]. XII, 106 p. (IHO publication S-63; edition 1.2.1). Available from: https://iho.int/uploads/user/pubs/standards/s-63/S-63_2020_Ed1.2.1_EN_Draft_Clean.pdf.
7. López C. Watermarking of digital geospatial datasets: a review of technical, legal and copyright issues. *International Journal of Geographical Information Science*. 2002;16(6):589–607. DOI: 10.1080/13658810210129148.
8. Calagna M, Mancini LV. Information hiding for spatial and geographical data. In: Belussi A, Catania B, Clementini E, Ferrari E, editors. *Spatial data on the web: modeling and management*. Berlin: Springer; 2007. p. 235–258. DOI: 10.1007/978-3-540-69878-4_11.
9. Aybet J, Al-Saedy H, Farmer M. Watermarking spatial data in geographic information systems. In: Jahankhani H, Hessami AG, Hsu F, editors. *Global security, safety, and sustainability. Proceedings of the 5th International conference; 2009 September 1–2; London, England*. Berlin: Springer; 2009. p. 18–26 (Communications in computer and information science; volume 45). DOI: 10.1007/978-3-642-04062-7_3.
10. Neyman SN, Wijaya YH, Sitohang B. A new scheme to hide the data integrity marker on vector maps using a feature-based fragile watermarking algorithm. In: *Proceedings of 2014 International Conference on Data and Software Engineering (ICODSE); 2014 November 26–27; Bandung, Indonesia*. [S. l.]: Institute of Electrical and Electronics Engineers; 2014. p. 1–6. DOI: 10.1109/ICODSE.2014.7062486.

11. Блинова ЕА, Урбанович ПП. Защита целостности данных электронных карт стеганографическим методом. В: Галкин ИМ, Романчик ВС, Волков ВМ, Расолько ГА, редакторы. *Веб-программирование и интернет-технологии WebConf2018. Тезисы докладов 4-й Международной научно-практической конференции; 14–18 мая 2018 г.; Минск, Беларусь*. Минск: БГУ; 2018. с. 147.
12. Блинова ЕА, Урбанович ПП. Стеганографический метод на основе встраивания дополнительных значений координат в изображения формата SVG. *Труды БГТУ. Серия 3, Физико-математические науки и информатика*. 2018;1:104–109.
13. Блинова ЕА, Урбанович ПП. Сравнительные особенности использования стеганографических методов в электронных картах. В: Ковалев МЯ, Бибило ПН, Гривачевский АГ, Дудкин АА, редакторы. *Информационные технологии в промышленности, логистике и социальной сфере (ITI'2019). Тезисы докладов X Международной научно-технической конференции; 23–24 мая 2019 г.; Минск, Беларусь*. Минск: ОИПИ НАН Беларуси; 2019. с. 22–24.
14. Блинова ЕА. Стеганографический метод на основе встраивания дополнительных значений координат в картографические данные. *Труды БГТУ. Серия 3, Физико-математические науки и информатика*. 2019;1:69–74.
15. Блинова ЕА, Урбанович ПП. Стеганографический метод на основе встраивания скрытых сообщений в кривые Безье изображений формата SVG. *Журнал Белорусского государственного университета. Математика. Информатика*. 2021;3: 68–83 (на англ.). DOI: 10.33581/2520-6508-2021-3-68-83.
16. StegoMap [Internet]. 2023 [cited 2023 February 1]. Available from: <https://stashevskaya.belstu.by/about>.

References

1. Sheppard E. GIS and society: towards a research agenda. *Cartography and Geographic Information Systems*. 1995;22(1):5–16. DOI: 10.1559/152304095782540555.
2. Longley PA, Goodchild MF, Maguire DJ, Rhind DW. *Geographic information science and systems*. 4th edition. Hoboken: Wiley; 2015. XVI, 505 p.
3. Mukherjee F. GIS use by an urban local body as part of e-governance in India. *Cartography and Geographic Information Science*. 2018;45(6):556–569. DOI: 10.1080/15230406.2018.1448304.
4. Rzeszewski M. Geosocial capta in geographical research – a critical analysis. *Cartography and Geographic Information Science*. 2018;45(1):18–30. DOI: 10.1080/15230406.2016.1229221.
5. Захаров МС, Кобзев АГ. *Картографический метод и геоинформационные системы в инженерной геологии*. 3-е издание. Санкт-Петербург: Лань; 2021. 116 с.
6. IHO data protection scheme [Internet]. Monaco: International Hydrographic Organization; 2020 [cited 2022 November 15]. XII, 106 p. (IHO publication S-63; edition 1.2.1). Available from: https://iho.int/uploads/user/pubs/standards/s-63/S-63_2020_Ed1.2.1_EN_Draft_Clean.pdf.
7. López C. Watermarking of digital geospatial datasets: a review of technical, legal and copyright issues. *International Journal of Geographical Information Science*. 2002;16(6):589–607. DOI: 10.1080/13658810210129148.
8. Calagna M, Mancini LV. Information hiding for spatial and geographical data. In: Belussi A, Catania B, Clementini E, Ferrari E, editors. *Spatial data on the web: modeling and management*. Berlin: Springer; 2007. p. 235–258. DOI: 10.1007/978-3-540-69878-4_11.
9. Aybet J, Al-Saedy H, Farmer M. Watermarking spatial data in geographic information systems. In: Jahankhani H, Hessami AG, Hsu F, editors. *Global security, safety, and sustainability. Proceedings of the 5th International conference; 2009 September 1–2; London, England*. Berlin: Springer; 2009. p. 18–26 (Communications in computer and information science; volume 45). DOI: 10.1007/978-3-642-04062-7_3.
10. Neyman SN, Wijaya YH, Sitohang B. A new scheme to hide the data integrity marker on vector maps using a feature-based fragile watermarking algorithm. In: *Proceedings of 2014 International Conference on Data and Software Engineering (ICODSE); 2014 November 26–27; Bandung, Indonesia*. [S. l.]: Institute of Electrical and Electronics Engineers; 2014. p. 1–6. DOI: 10.1109/ICODSE.2014.7062486.
11. Blinova EA, Urbanovich PP. [Protecting the integrity of e-card data steganographic method]. In: Galkin IM, Romanchik VS, Volkov VM, Rasol'ko GA, editors. *Веб-программирование и интернет-технологии WebConf2018. Тезисы докладов 4-й Международной научно-практической конференции; 14–18 мая 2018 г.; Минск, Беларусь* [Web programming and internet technologies WebConf2018. Abstracts of the 4th International scientific and practical conference; 2018 May 14–18; Minsk, Belarus]. Minsk: Belarusian State University; 2018. p. 147. Russian.
12. Blinova EA, Urbanovich PP. A steganographic method based on the embedding of additional coordinates into images of SVG format. *Proceedings of BSTU. Issue 3, Physics and Mathematics. Informatics*. 2018;1:104–109. Russian.
13. Blinova EA, Urbanovich PP. [Comparative features of the use of steganographic methods in electronic maps]. In: Kovaliev MYa, Bibilo PN, Grivachevskii AG, Dudkin AA, editors. *Информационные технологии в промышленности, логистике и социальной сфере (ITI'2019). Тезисы докладов X Международной научно-технической конференции; 23–24 мая 2019 г.; Минск, Беларусь* [Information technologies in industry, logistics and social sphere (ITI'2019). Abstracts of the 10th International scientific and technical conference; 2019 May 23–24; Minsk, Belarus]. Minsk: United Institute of Informatics Problems, National Academy of Sciences of Belarus; 2019. p. 22–24. Russian.
14. Blinova EA. A steganographic method based on the embedding of additional coordinates into spatial data. *Proceedings of BSTU. Issue 3, Physics and Mathematics. Informatics*. 2019;1:69–74. Russian.
15. Blinova EA, Urbanovich PP. Steganographic method based on hidden messages embedding into Bezier curves of SVG images. *Journal of the Belarusian State University. Mathematics and Informatics*. 2021;3:68–83. DOI: 10.33581/2520-6508-2021-3-68-83.
16. StegoMap [Internet]. 2023 [cited 2023 February 1]. Available from: <https://stashevskaya.belstu.by/about>.

Received 15.12.2022 / revised 16.02.2023 / accepted 21.03.2023.