STEGANOGRAPHIC ALGORITHM FOR INFORMATION HIDING USING SCALABLE VECTOR GRAPHICS IMAGES

Branislav MADOŠ, Ján HURTUK, Marek ČOPJAK, Peter HAMAŠ, Michal ENNERT
*Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics,
Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic, tel. +421 55/602 3023,
e-mail: { branislav.mados, jan.hurtuk, marek.copjak, michal.ennert}@tuke.sk, peter.hamas@student.tuke.sk

ABSTRACT

Besides cryptography, the great attention is paid to the steganography, which is considered not only the science but also the art of the concealed communication. In steganography, the information is hidden within another piece of information, called stegomedia. This paper presents a new data hiding technique based on steganographic algorithms that is hiding information into vector images. The paper briefly introduces this technique and evaluates the benefits and drawbacks of the proposed approach.

Keywords: *steganography, vector image, SVG, scalable vector graphics*

1. INTRODUCTION

There has been an explosive growth of digital multimedia, communication and computer applications during the last decades. Wide distribution of digital information is related with easy data modification and duplication. The need to secure information and personal data is gaining importance not just in the military but also in our everyday lives.

Data hiding techniques received less attention from academic community than cryptography, but the situation has changed at the beginning of the 90's. The first academic conference about the subject of data hiding was held in 1996. Since then many other conferences took place followed by many articles in periodicals and journals.

It is often assumed that a communication channel is secure once cryptography is applied on the waging communication. This does not have to be always true, as the selected cryptography technique can have vulnerable points, for example unknown bugs in the protocol, continual increase in calculation speed of modern computers and related brute force methods, password phishing attacks, etc. Steganography, in contrast to cryptography, conceals the fact that two sides are exchanging information. If used along with cryptography, it hampers the attacker's efforts. Attacker has to discover if the communication is taking place before he can try to break the communication protocol.

The topic of this work is to present an overview of the modern steganography techniques, to classify the proposed steganography algorithm into the data hiding hierarchy and to analyse the benefits and drawbacks of the proposed algorithm based on defined criteria. The goal of this work is to define a new approach for hiding secret messages into Scalable Vector Graphics (SVG) images, aiming to create minimum modifications to the vector cover image.

If steganography is the art of hiding information into other information, then the art of detecting secret messages embedded using steganography is called steganalysis. The aim of steganalysis is to determine if a cover message contains the secret message and is based mostly on statistical methods. Compares the statistical deviation of what the cover image looks like and how it should look like and tries to predict the length of the secret message.

The less the cover image has been modified, the harder it is for steganalysis to determine if any secret communication is taking place and the steganographic method can be considered more successful. The design of the proposed algorithm concentrates mainly on this fact.

2. SUBJECT

Steganography is a part of techniques that specializes in information hiding (Fig. 1) [1]. Covert channels are defined as any kind of data transportation method, that was originally not intended for this purpose [2][3].

Watermarking is in some aspects similar to steganography, but in contrast to steganography it needs to be robust, ergo it should be insusceptible to manipulation or render the manipulated data useless in the process of destroying the watermark [4][5][6].

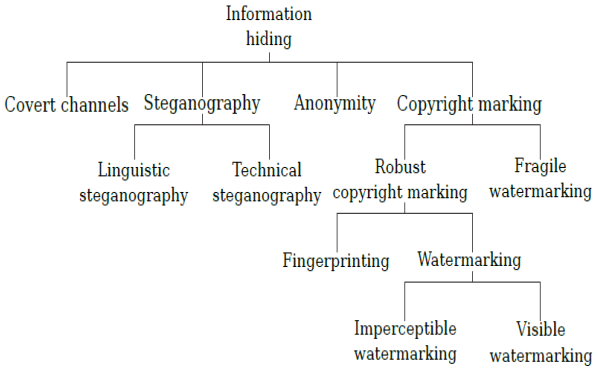

Fig. 1 Classification of information hiding techniques

Steganography can be further classified based on the chosen classification criteria. Classical steganography includes steganographic methods used before the computer era (Fig. 2) [7].

Linguistic steganography consists of methods that conceal messages into natural human language, commonly using cover media in the form of written text [8][9][10]. Examples are including acrostics, Javanais, Piglatin, semagrams which hide the message into small graphical details, cues, etc.

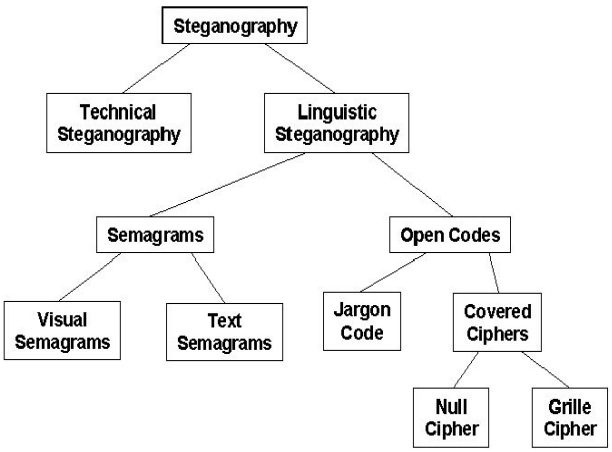

Fig. 2 Classification of classical steganographic methods

Technical steganography hides the message with the help of scientific methods using various types of cover mediums [7]. Examples include the German microdot and invisible ink.

Modern steganographic techniques use the advantages of computer systems. They can be divided based on the type of cover medium used, for example images, audio, video or text. Image steganography can be performed using raster or vector cover images. The basic type of raster image steganography is the LSB steganography, which hides the message into the least significant bits of the raster cover image. The changes are so small that they are invisible to the naked eye.

Vector image steganography methods can be divided into jittering and embedding [11][12]. Jittering is similar to LSB steganography, due to storing the secret message into the least significant digits of point coordinates in vector images.

Embedding stores the secret message into extra image points generated directly on other graphical objects. For example if there is a line consisting of two points, extra point can be put directly on the line without changing the visual form of the line (Fig. 3) [12].

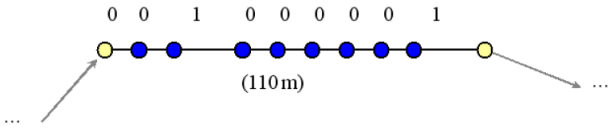

Fig. 3 Message hidden using embedding algorithm

The algorithm creates a collection of points along a line starting with a reference point. The length of the reference point is compared to the lengths of other points. The same length may represent the bit 0 and two times the length bit 1. It is possible to encode for example 8 bits in one extra image point by introducing 256 different lengths, etc. The algorithm is more robust than jittering by being resistant to modifications such as scaling, moving and rotation.

A new algorithm, proposed in this paper for vector steganography is based on the embedding algorithm (Fig. 4). The proposed algorithm transforms the secret message from byte array to a stream of decadic digits. The stream is divided into the smaller parts that are labeled m and are based on the precision of numbers in SVG vector image.

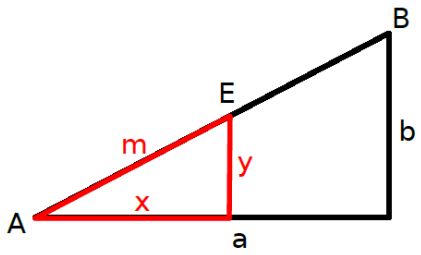

Fig. 4 Message hidden using embedding algorithm

The x and y coordinates are calculated using simple trigonometry and rounded based on the chosen precision. The value of the message m stored in point E can be restored using the values x, y and the Pythagorean Theorem.

The secret message can be transformed to the digit stream in different ways, depending on the chosen encoding method. In order to determine the most efficient encoding, with the least number of output digits generated, a program designed as the part of this work has been implemented and tested on a 12MB of random data secret message (see Table 1).

Bits column represents the number of bits to remove from the secret message in one step and digits column represents the number of decadic digits created. Extra bit is the chance to take one extra bit from the secrete message byte array. For example the 3 bit encoding needs to encode the numbers from 2 to 7 using three bits, but numbers 0, 1, 8 and 9 can be encoded using four bits, which shortens the message by one bit.

Total removes is the total number of steps it took to transform the whole message and total digits is the number of generated decadic digits. Not all bit encodings make sense, for example 1 and 2 bit encodings always generate more digits than 3 bit encoding, because they take less extra bits each step.

The shortest digit stream has been generated using the 63 bit encoding. Prolongation column represents the extension of the secret message, if it has been encoded using some other type of encoding.

Table 1 Selecting the encoding with least digits generated

Bits	Digits	Extra Bit	Total Removes	Total digits	Prolongation
63	19	16	1585189	30118591	0.00
53	16	20	1882855	30126160	0.03
59	18	85	1674082	30133476	0.05
49	15	87	2008992	30134880	0.05
39	12	90	2511348	30136176	0.06
43	13	24	2318192	30136496	0.06
29	9	93	3348632	30137688	0.06
19	6	95	5023250	30139500	0.07
9	3	98	10046963	30140889	0.07
56	17	56	1773433	30148361	0.10
33	10	28	3015284	30152840	0.11
46	14	59	2154195	30158730	0.13
36	11	63	2743109	30174199	0.18
23	7	32	4311812	30182684	0.21
26	8	66	3774969	30199752	0.27
16	5	69	6050948	30254740	0.45
13	4	36	7563853	30255412	0.45
6	2	72	15237923	30475846	1.19
3	1	40	30769748	30769748	2.16

3. SOLUTION AND RESULTS

This section focuses on the implementation of the proposed solution and summarizes attained results.

3.1. The design of the vector steganography algorithm

The proposed algorithm begins by preparing the secret message for encoding (Fig. 5). The message is transformed from byte array to a digit stream using the 63 bit encoding. The algorithm takes each step 63 or 64 bites and turns them into 19 decadic digits. The last 63 or less bits are an exception, since they can be turned into 20 down to 2 digits, depending on the minimum number of bits required to encode the number (the one extra digit is used for decoding purposes). The algorithm cycles through all the polygons of the vector cover image and gets all point coordinates for each polygon. If the coordinates are in an unsupported format, the polygon is being ignored. Otherwise it checks if the first supported polygon has finally been found and if yes, it initialises its first point with metadata. Next the polygon needs to be prepared. The algorithm encodes the points containing secret message on other vector objects, so there must not be any extra points on vector objects already present in the image before encoding. After the message encoding there needs to be precision correction, because the floating point coordinates of the extra points are always rounded based on the SVG standard.

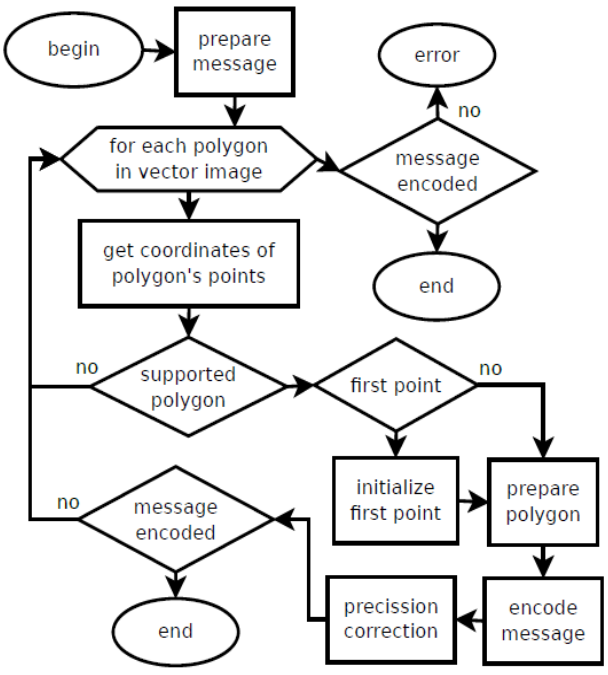

Fig. 5 Proposed vector steganography algorithm

3.2. Evaluation of the proposed algortihm

The proposed algorithm has been tested on random input messages of varying length and compared to the default algorithm, which uses the 8 bit encoding without using the extra bit. The results show that the proposed algorithm allows shortening of the secret message after encoding by up to 20% (Table 2).

Table 2 Number of generated digits comparison

Message Length	Default Algorithm Digits	Proposed Algorithm Digits
1B	3	2 - 4
7B	21	2 - 18
8B	24	2 – 21
9B	27	21 – 23
1KB	3 000	2 396 – 2 414
10KB	30 000	24 075 – 24 112
100KB	300 000	240 922 – 240 978
1MB	3 000 000	2 409 411 – 2 409 543
100MB	30 000 000	24 094 605 – 24 094 870

4. CONCLUSIONS

The proposed vector steganography algorithm shortens the input secret message after encoding, thanks to the effective conversion of message bytes into decadic digit stream, resulting in less extra points generated into vector images, making the images less suspicious and more secure. The algorithm uses 63 bit encoding, which is the limitation of fundamental integral types in the C++ language. It is interesting to research more effective message conversions using higher bit variables. Additional place for research is in the improvement of the precision correction with the goal of finding the most number of digits that may be encoded into a single point manipulating its coordinates accordingly.Eventually, the algorithm's security may be improved by placing the extra points onto other vector objects, for example using Beezier curves.

ACKNOWLEDGMENTS

This work was supported by the Slovak Research and Development Agency under the contract No. APVV-0008-10 and KEGA 008TUKE-4/2013 Microlearning environment for education of information security specialists. The projects are being solved at the Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice.

REFERENCES

[1] PETITCOLAS, F.A.P. - ANDERSON, R.J. - KUHN, M.G.: *Information hiding a survey*. Proceedings of the I.E.E.E. conference, July 1999, 87 (7): 1062 - 1078

[2] REILAND, K et al: Steganography and Covert Channels. PACISE conference. Bloomsburg, PA, 2005.

[3] LANDWEHR, C.E. - BULL, A.R., McDERMOTT, J.P. - CHOI, W.S.: A Taxonomy of Computer Program Security Flaws, with Examples. ACM Computing Surveys, Vol. 26, No. 3, September 1994, pp. 211-254.

[4] COX, I.J. et al: Digital Watermarking and Steganography. Second Edition. Burlington: Elsevier, 2008, ISBN 978-0-12-372585-1.

[5] CVEJIC, N. - SEPPANEN, T.: Increasing Robustness of LSB Audio Steganography Using a Novel

Embedding Method. ITCC, Vol. 2, pp. 533 to 537, 2004.

[6] BARAN, B. - GOMEZ, S. - BOGARIN, V.: Steganographic Watermarking for Documents. Proceedings of the 34th Annual Hawaii International Conference on System Sciences, IEEE CS Press, Los Alamitos, Calif., 2001.

[7] KESSLER, G.C. et al: An Overview of Steganography. Advances in Computers. Vol. 83, 2011, Num. 1, pp. 51-107.

[8] CHAPMAN, M. - DAVIDA, G. - RENNHARD, M.: A Practical and Effective Approach to Large-Scale Automated Linguistic Steganography. Proceedings of the Information Security Conference, October 2001, pp. 156-165.

[9] BENNETT, K.: Linguistic Steganography: Survey, Analysis, and Robustness Concerns for Hiding Information in Text. Purdue University, CERIAS Tech. Report, 2004.

[10] TILBORG, H.C.A: Encyclopedia of Cryptography and Security. New York: Springer-Verlag, 2005. ISBN 0-387-23473-X.

[11] HUBER, W.A.: GIS and Steganography Part 3 Vector Steganography.

[12] THOEN B.: GIS and Steganography Part 2 As applied to MapInfo and ArcView.

Received January 30, 2015 , accepted March 5, 2015

BIOGRAPHIES

Branislav Madoš (Ing., PhD.) was born on 20th May 1976 in Trebišov, Slovakia. In 2006 he graduated (Ing.) at the Department of Computers and Informatics at the Faculty of Electrical Engineering and Informatics of the Technical University of Košice. He defended his PhD. in the field of Computers and computer systems in 2009; his thesis title was "Specialized architecture of data flow computer". Since 2010 he is working as an Assistant Professor at the Department of Computers and Informatics. His scientific research is focused on the parallel computer architectures and control flow computer architectures.

Ján Hurtuk (Ing.) was born on 4th October 1988 in Kežmarok. In 2013 he graduated (MSc.) at the Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, TUKE. Since 2014 he is studying as a PhD. student at the Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice.

Marek Čopjak (Ing.) was born on 29th June 1987 in Kežmarok. In 2012 he graduated (MSc.) at the department of Cybernetics and Artificial Intelligence of the Faculty of Electrical Engineering and Informatics at Technical University in Košice. Since 2013 he is studying as PhD. Student at the department of Computers and Informatics at the Faculty of Electrical Engineering and Informatics the Technical University of Košice.

Peter Hamaš (Ing.) was born in Košice, Slovakia, in 1990. He received his master's degree in Informatics in 2014 from Faculty of Electrical Engineering and Informatics, Technical University of Košice. His research is focusing on the methods of concealing messages using data hiding and steganography. Since 2014 he is working as a programmer specializing in CAD systems for Ekosoft s.r.o., Košice.

Michal Ennert (Ing.) was born on 4th August 1987 in Revúca, Slovakia. In 2011 he graduated at the Department of Computers and Informatics of the Faculty of Electrical Engineering and Informatics at the Technical University of Košice and received the engineering degree. Since 2011 he is PhD. student. He is doing research and experiments mainly in the field of computer security with usage of GPGPU technology and in the field of distributed software architecture.