DE GRUYTER OPEN
RIGA TECHNICAL UNIVERSITY
Applied Computer Systems
ISSN 2255-8691 (online)
ISSN 2255-8683 (print)
December 2016, vol. 20, pp. 40–48
doi: 10.1515/acss-2016-0014
https://www.degruyter.com/view/j/acss

Development of Requirements Specification for Steganographic Systems

Dāvids Grībermans[1], Andrejs Jeršovs[2], Pāvels Rusakovs[3]
[1–3]*Department of Applied Computer Science, Riga Technical University, Latvia*

***Abstract* – The paper focuses on development of requirements specification for steganographic systems. The main concepts and fields of use of steganography are briefly explained. Criteria that can be used for various non-functional requirements are grouped and their possible metrics are given with examples and sample requirements. The authors provide an original systematic approach to develop requirements for steganographic systems based on the field of use and the importance of each criterion in the selected field. The approach also allows for automated selection of steganographic algorithms based on the requirements and is related to the concept of the Universal Stegoconstructor, which guarantees that clients receive the required steganographic system from the developers.**

Keywords – Copyright protection, data security, digital watermarking, requirements specification, steganography.

I. Introduction

The word "steganography" comes from the Greek language and means "covered writing". Steganography as a science studies the exchange of information in a way that the fact of the exchange remains unseen [1].

It shares a goal similar to cryptography – to protect information, but unlike cryptography, where the goal is to make the contents of the information unreadable for unauthorised persons, the goal of steganography is to hide the existence of information itself. Although steganography does not require computers to be performed, the given paper focuses on a form of modern steganography – digital steganography, which hides messages (a sequence of bits) into containers (usually a file, also a sequence of bits) resulting in a stegocontainer, a file with the message embedded into it. Digital pictures, videos, text documents and other digital files can be used as a container as long as they contain some redundant data. A simplified steganographic process is shown in Fig. 1.

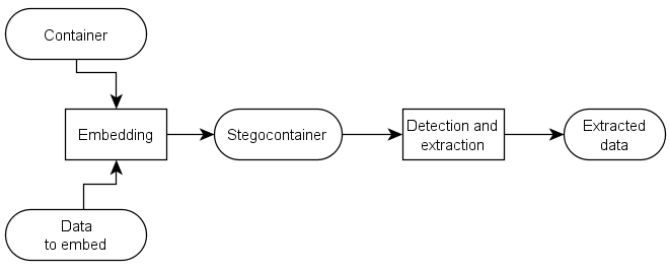

Fig. 1. A simplified steganographic process [1].

Steganography has multiple fields of use (see Section II), but the classic field of use for steganography is hidden communication. For example, Alice needs to send a message to Bob through a stegochannel that is monitored by a warden. The warden reads all the messages before they arrive to Bob. In cases when the warden finds something suspicious the message is blocked and will never reach Bob (see Fig. 2). In such a situation, cryptography will protect the message itself, but will raise the suspicion of the warden. Steganography would embed the secret message into another message or an image and the warden would ideally let it through to Bob without any suspicion.

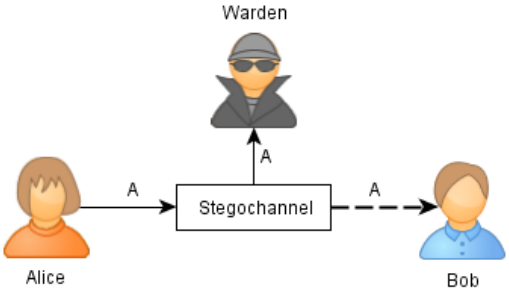

Fig. 2. Hidden communication example.

The field of digital steganography is growing in popularity due to the problems of communication privacy (various government surveillance scandals), copyright protection and possible encryption limitations, which can limit the freedom of speech [2], as well as already existing limitation in China [3]. Steganography is also a potentially growing threat due to its potential use by terrorists, which in turn boosts the field of steganalysis that is aimed at detecting the use of steganography. Along with the popularity of the field of steganography, the interest in developing steganographic systems and software could also increase in the future.

In order to develop a steganographic system for any purpose, a requirements specification is necessary. This way the client could define the goals that need to be achieved and the requirements that the developers need to consider during development.

Sommerville [4] defines requirements specification as "the process of writing down the user and system requirements in a requirements document". In [5], this document is called the requirements specification and is defined as an "unambiguous and complete specification document", which is needed for software customers to describe what they want to obtain and the developer to understand what the customer wants.

The requirements specification should contain functional and non-functional requirements, where functional requirements define the services the system should provide and non-functional requirements are constraints on those services. Although there are guides on how to develop requirements specifications [4], [6], the field of digital steganography has some specifics that should be considered and have not been researched previously in the context of requirements specification. The goal of the given paper is to provide a guide for the development of such requirements specifications for steganographic systems as well as show how the ideas can be used further for the automated selection of steganographic methods or algorithms based on the customer use scenario.

Section II introduces the fields of use of modern digital steganography. Section III presents the approach to the development of requirements specification for steganographic systems. Section IV is devoted to various requirements and criteria that can be defined for steganographic systems. Section V introduces the concept of the Universal Stegoconstructor and links it to the approach presented in Section III. Section VI briefly explains the possibilities of automated method selection based on the requirements. The last section provides a summary of the present paper as well as defines areas of further research.

II. Application Fields

Digital steganography has multiple fields of use. The fields summarised in the paper are based on [1] and [7] (see Fig. 3).

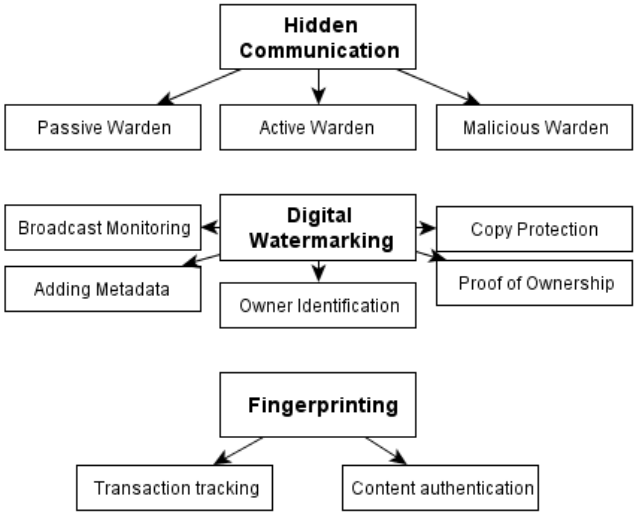

Fig. 3. The main application fields of digital steganography.

In the field of concealed communication, the goal is to hide the message from the attacker as the protection of the message is the top priority. In the case of passive warden (see Fig. 2), the attacker can only see the message and block it, if necessary. In case of the active warden, he is able to alter the message before Bob receives it. The malicious warden is able not only to modify the message, but also send fake messages in order to confuse Bob and Alice and find out their means of communication. This case is rarely considered in steganography methods [1].

The field of digital watermarking focuses on protection of copyright and intellectual property. Instead of a hidden message, an invisible digital watermark is embedded, which either identifies the author of the work, the owner of the copy or some other information, like metadata or copy protection data. In this field, the focus is on the robustness of the message – the watermark must still be readable even after various transformations of the original container (for example, compression). The "secrecy" of the message usually is not as important as in concealed communication, sometimes the fact that a watermark exists is no secret at all [1], [7].

The field of fingerprinting hides data with specific information that can be used to track individual media files, usually for content authentication (for example, finding copyrighted videos on video sharing sites) or transaction control, where each transaction is identified by the embedded message.

Each application field has its own goals related to the use of steganography. The criteria that are most important in each field also vary. When the client defines the requirements, the concept of the "magic" triangle should be noted (see Fig. 4). It features three corners: capacity, imperceptibility and robustness (all 3 criteria are introduced in Section IV). The main idea is that all 3 criteria cannot be implemented to a high degree at once and the client needs to consider his priorities and sometimes lower the minimum requirements.

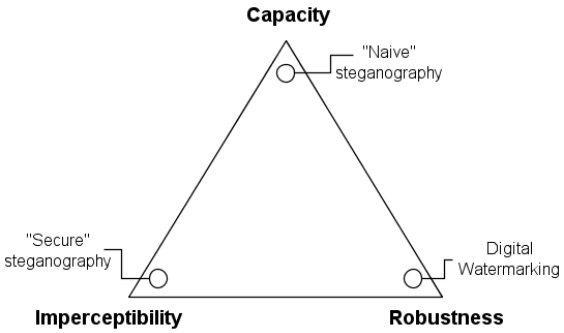

Fig. 4. The "magic" triangle of digital steganography [8].

The triangle is not the only possible figure, depending on the use case both the figure and the criteria can easily change, for example, paper [9] introduces a "magic" hexagon for digital image steganography. In any case, the requirements need to be defined in order to develop requirements specification.

III. Developing Requirements Specification

The authors suggest a systematic approach to develop requirements specification for steganographic systems: it starts with the definition of goals of the future system and ends with the selection of the possible algorithms (see Fig. 5 for the steps of this approach). The circles with A, B, C and D are links to another concept and will be introduced in Section V.

Step 1 is the definition of the general goal that the steganographic system must achieve, for example, protect video using watermarks or encode hidden data in video files. Based on that goal, the client must choose the application field

of digital steganography (see Section II). Based on the application field selected, the client can define the functional requirements and select criteria that are required (see Table VI demonstrating the importance of various criteria based on the application field), as well as define the non-functional requirements for the system accordingly (see Section IV for the possible criteria that are specific in the field of digital steganography). After defining all other requirements, including those that are not specific in the field of digital steganography (Step 5), the specification document can be created (Step 6) and the developers can use the specification to select (or develop) the possible steganographic algorithm (Step 7). This step can be automated (see Section VI). After the algorithm is selected, the system can be developed according to the specification.

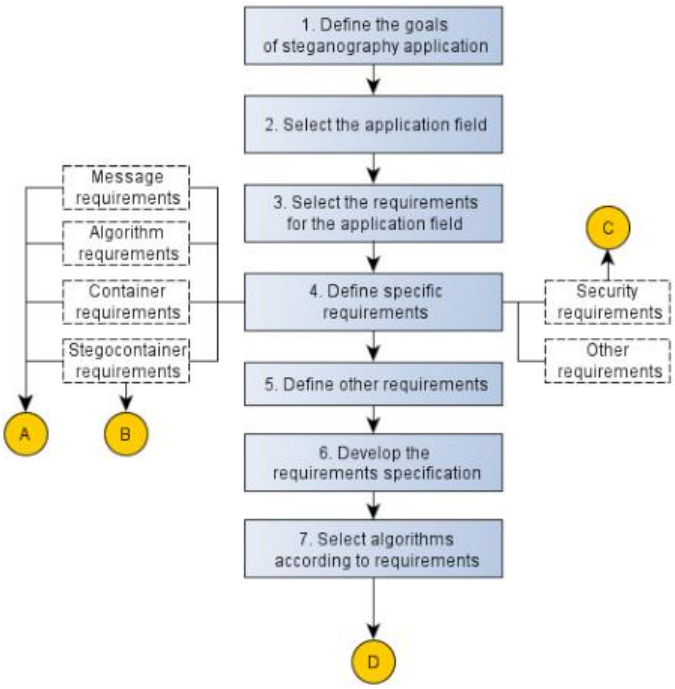

Fig. 5. The systematic approach to develop requirements specification for steganographic systems.

IV. Requirements for Steganographic Systems

The authors have compiled and grouped some of the possible criteria for steganographic systems that can be used to define requirements. The groups are as follows:

- Message requirements (see Table I);
- Container requirements (see Table II);
- Stegocontainer requirements (see Table III);
- Algorithm requirements (see Table IV);
- Security requirements (see Table V).

The corresponding tables provide the criteria, their short descriptions, possible metrics and examples of requirement. The criteria given are meant for use in the requirements specification and can be defined both for the algorithm used and the steganographic system itself as a whole.

Steganographic systems are not limited to the criteria mentioned – they can have other requirements that are not specific in the field of digital steganography. In [4] and [6], the process of defining these requirements is described. One such example – the cost of the steganographic system, where the cost of development, cost of encoding and decoding hardware as well as maintenance costs must all be considered during the requirements engineering phase [1]. This requirement is almost the same as in any other information system and will not be considered in the present paper.

Table VI shows the importance of the mentioned criteria in each of the application fields mentioned in Section II. Using this table, the client can select the intended field of use and see all the important criteria and requirements that need to be defined. The table has been created by the authors using descriptions of digital steganography and watermarking available in [1], [7] and [10] as well as relying on personal experience.

Some criteria are optional and are dependent on the specific use case – for example, embedding domain and stegocontainer size. The client must decide if it is necessary to define a requirement for this criterion or not.

Criteria that are marked as “very important” are almost obligatory in the application field and must be included in the requirements specification. Criteria marked as “important” should also be included, but they are not the main focus in the application field, and requirements for the criterion can be lower to meet the needs of other requirements.

Criteria that are marked as “not important” or “not needed” can be skipped unless needed by the specific use scenario.

As shown in the table, the field “Hidden Communication” focuses on the secrecy of the message (imperceptibility) and the size of the embedded message (maximum size and capacity). The container can be either selected or constructed (container selection type), as the user is not forced to embed in a specific cover container and choose another one if a specific one does not work well. Both public and private keys can be used.

Digital watermarking and fingerprinting focus on robustness to keep the hidden message or watermark safe from various transformations both by attackers and transformations during broadcast or compression. The container selection type is non-selective, which means that the system is not allowed to skip some containers as it should embed in them, which will in turn raise requirements for embedding effectiveness, which is also marked as important for this reason. The type of key for digital watermarking depends on the application – in some cases, only private or public keys can be used (for example, proof of ownership – only the owner should be able to read and write the message, or copy protection – where only the owner can write the message, but any software that reads the media should be able to read the embedded message back).

TABLE I

MESSAGE REQUIREMENTS

Criterion	Short description	Metrics	Example
Maximum message size	Describes the maximum size of the message that the system must be able to embed into a given container. Can be given in bits or some other unit (characters, pixels, etc.).	Bits	The system must be able to embed a message with the size up to 32 bits.
Message modification	Describes the possibility to modify an already embedded message without the need for the original container. One way to do this is to modify the changed bits again to new values or return the stegcontainer to the original condition and embed again [1].	Yes / No	The system must be able to change the embedded message without using the original container.
Multiple messages	Describes if it is necessary to be able to embed multiple messages into a single container [1]. This can be done with multiple messages in different parts of the container or by nesting stegocontainers by layers (see [11] – metadata are embedded into geographical data with other geographical data, which also have metadata embedded into them).	Yes / No	The system must be able to embed up to 2 messages with the maximum size into a container.

TABLE II

CONTAINER REQUIREMENTS

Criterion	Short description	Possible metrics	Example
Type of container	Describes the possible type of digital containers. Almost any type of file can be used, but the container should contain redundant data, which can be used for embedding [12]. It is also necessary to specify the nature of the digital file – it can be of fixed size (the size is known before embedding) or stream (the final size is unknown) [7]. The contents of the containers should also be described (photos of animals, people, technical documents, etc.).	Text, Image, Sound, Video, Other. Fixed size or stream.	The system must be able to embed messages into fixed size digital photos of city landscapes and mountains.
Supported file formats	Specifies the file formats that the system must support for embedding and extraction. If only one format is specified, file format specific methods can be used. More specific file formats can also be given, for example, XML files generated by PowerDesigner 4.2 from class diagrams. The client should consider that the attackers can change the file format of the stegocontainer possibly removing the hidden message.	File formats and specifications	The system must be able to embed and extract messages from and into JPG and PNG images.
Container selection type	Specifies if the system has a choice of the container to embed the message into [7]. If a file is given to the system and the message must be embed into it the selection type is non-selective. If the system has a number of files and can select the best one for embedding the selection type is selective [13]. Constructing type generates the container based on the message that needs to be hidden, for example, [14] generates a spam message with hidden data and no input container.	Non-selective; Selective; Constructing.	The system can select the file from a list based on the parameters.

TABLE III

ALGORITHM REQUIREMENTS

Criterion	Short description	Possible metrics	Example
Capacity	Describes how many bits can be hidden per unit of data. The unit of measurement is heavily influenced by the steganographic algorithm used and defining this criterion too specifically can limit the spectre of algorithms that the developer can use.	Bits per unit (Mb, pixel, DCT block, minute etc).	The system must be able to embed at least 1 bit per nonzero DCT coefficient.
Speed	Shows the speed at which the steganographic algorithm can perform.	As the speed of an algorithm is both hardware and implementation dependent it is suggested to use the Big-O Notation to describe the speed like regular information systems (see [15] and [4]). Maximum execution time in seconds for specific functions can also be defined (client should then define the hardware and containers that will be used for measurements).	The system must be able to embed a message of maximum size into a 500x500 pixel BMP format picture within 5 seconds.
Embedding domain	Defines the domain in which the data are embedded within the container.	Name of the embedding domain (Spatial domain, Transform domain, Wavelet domain, Time domain and others) as well as a range or frequency specifications.	The system must embed the message in the Spatial domain of the container.

TABLE IV

STEGOCONTAINER REQUIREMENTS

Criterion	Short description	Possible metrics	Example
Imperceptibility	Shows how much the stegocontainer has changed compared to the original container. If the changes are noticeable, the stegocontainer can raise suspicion and the message can be found. The changes can appear as various noises, artefacts, errors or changes in colour depending on the container and method used.	Peak Signal-to-Noise-Ratio (PSNR) for Video and images (see [16]), Signal-to-Noise-Ratio (SNR) for sound [7]. For text: percentage of changed units (letters, words, tags etc.).	The PSNR of the stegocontainer compared to original container must not exceed 35 dB.
Fidelity	Shows the perceptual difference between the original container and the stegocontainer. This is the difference in quality that viewers can notice after a video has been compressed with a video codec.	In [17] a methodology is presented that allows measuring quality and noise at 5 levels based on a group of respondents who have shown both original and modified work.	The stegocontainer can contain perceptible noise, but it must not be annoying.
Stegocontainer size	Defines the minimum and maximum size of the stegocontainer. The minimum size is needed because smaller sizes may have no redundant data to embed the message. The maximum size is needed to limit the size growth of the stegocontainer compared to the original container.	Bits; percentage of increase.	Both containers and stegocontainers must be of the size in between 2 MB and 7 MB. Stegocontainer must not be more than 10 % larger than the original container.
Embedding efficiency	Defines the probability that an embedded message can be extracted from the stegocontainer. Some methods cannot always guarantee that the message will be successfully embedded into the container and the message will not always be readable [7]. The parameter defines the minimum acceptable probability.	Probability – from 0 (never) to 1 (always).	The embedding efficiency must be 0.95 or more.
False Positive Rate	Shows the probability of a container to be detected as having a message when it does not have a message embedded; the probability of a message not being detected inside a stegocontainer.	Percentage or probability of false positive detections (see [18] and [1]).	False positive rate must not exceed 10^{-6}.
Extraction method	Defines how the hidden message is extracted. *Blind* – only the stegocontainer is needed for extraction. *Informed* – both the stegocontainer and the original container are needed for extraction. In [1], it is noted that for most use cases informed extraction is impossible, but when it is possible and the attacker has no access to the original container, informed methods can be more efficient than blind ones.	Blind; Informed.	The system must be able to extract the hidden message from the stegocontainer without having the original container.

TABLE V

SECURITY REQUIREMENTS

Criterion	Short description	Possible metrics	Example
Resistance against steganalysis	Steganalysis is a field that specialises in discovering the use of steganography and the detection of hidden messages, usually by using methods of statistics [10] (see [19], [20] and [21] for examples of such approaches). Steganalysis can be performed both manually or using some form of software (even completely automatically).	Paper [22] presents a scale of 1 to 3 to describe the resistance against steganalysis based on 1. the existence of steganalysis algorithms to break the steganographic algorithm; 2. statistical changes in the stegocontainer; 3. information that can be received by using incorrect keys with the stegocontainer during extraction.	The steganographic algorithm must not introduce any statistical changes to the container and should not have steganalysis methods against publicly available ones.
Robustness	Robustness in the context of steganography specifies the resistance of the message to various transformations of the stegocontainer [1]. For example, these transformations could be compression, format change printing and scanning, writing the video to VHS tape, change of sound pitch, etc. These changes could be done both by attacker and by the channel in which the media are sent (for example, online video services). These alterations can make the hidden message unreadable.	The client needs to specify whenever he needs a robust or fragile algorithm. In fragile algorithms the message is lost, whenever the stegocontainer undergoes changes, which is necessary in multiple fields of use. If the client specifies the need for a robust algorithm, it is needed to specify the types of transformations the container should be able to undergo without losing the hidden message [23].	The message embedded into the stegocontainer must be resistant to image rotation up to 1.5 degrees in any direction.
Resistance against attacks	Defines the resistance of the system to various attacks [10]. Defines 4 groups of attacks: 1. Attacks against the embedded message or watermark; 2. Geometric attacks; 3. Cryptographic attacks; 4. Attacks against the steganographic algorithm used or the system itself.	The actions that should be allowed and denied for all users should be specified, which can result in the list of possible attacks that need to be guarded against (see [1] and [10] for possible attack types).	The system must allow regular users to detect the embedded watermark, but only content owners can embed or change it.
Type of keys	The parameter specifies the type of keys used in the system. It must be noted that there are two types of keys: steganographic and cryptographic keys. The steganographic key specifies the way in which the algorithm hides the message, but cryptographic keys encrypt the message itself. In [24], it is defined that a secure cryptographic algorithm should not allow the message to be extracted without the key even when the method is known. The authors suggest that the same should be applied to the steganographic algorithm.	None, private, public keys or both. See [7] for details about the key types. The principles apply to both steganographic and cryptographic keys.	The system must use private steganographic keys. The message should be encrypted using public cryptographic keys.

TABLE VI

IMPORTANCE OF CRITERIA BY FIELDS OF USE AND POSSIBLE VALUES

	Hidden Communication			**Digital Watermarking**					**Fingerprinting**	
	Passive Warden	**Active Warden**	**Malicious Warden**	**Broadcast Monitoring**	**Owner Identification**	**Proof of Ownership**	**Copy Protection**	**Adding Metadata**	**Transaction tracking**	**Content authentication**
Maximum message size	Important			Optional	Not important			Optional	Not important	
Message modification	Not needed			Not important			Optional		Optional	No
Multiple messages	Not needed			No	Optional	No	Optional		Optional	No
Type of container	Optional			Optional					Optional	
Supported file formats	Optional			Optional					Optional	
Container selection type	Selective, Constructing			Non-selective					Non-selective	
Imperceptibility	Very important			Important					Important	
Fidelity	Not needed			Important					Important	
Stegocontainer size	Optional			Optional					Optional	
Embedding effectiveness	Not important			Important					Important	
False positive rate	Not important			Optional					Optional	
Extraction method	Blind, informed			Blind		Blind, informed	Blind		Blind, informed	Blind
Capacity	Important			Optional	Not important			Optional	Not important	
Speed	Not important			Important	Optional				Optional	
Embedding domain	Optional			Optional					Optional	
Resistance against steganalysis	Very important			Optional					Optional	
Robustness	No	Important		Very important					Important	
Resistance against attacks	No		Yes	Yes	No	Yes			Optional	
Type of keys	Public, Private			Public, Private, None	Public	Private	Public	Public, Private, None	Public, Private	

V. THE UNIVERSAL STEGOCONSTRUCTOR

A connected concept to the approach defined in Section III is the Universal Stegoconstructor, which is described in detail in [25] and [11]. It allows the customer to receive and the developer to select the most appropriate steganographic method based on the criteria defined. The concept is shown in Fig.ure 6. Multiple steps of the Universal Stegoconstructor need inputs, which can be provided by the approach defined in section III of this work (see A, B, C and D as inputs in Fig. 6 as well as outputs in Fig. 5).

Input A passes the information about the container and the size of the embedded message. Step B selects the necessary level of quality for embedding based on the user requirements. Step C is used to select the required level of robustness based on the channel noises and possible transformations. Step D was added in the given work and provides a library of steganographic methods that is described in Section VI.

By using both the approach presented and the Universal Stegoconstructor concept, the client can successfully define the requirements needed and the developers can select the most appropriate methods for development.

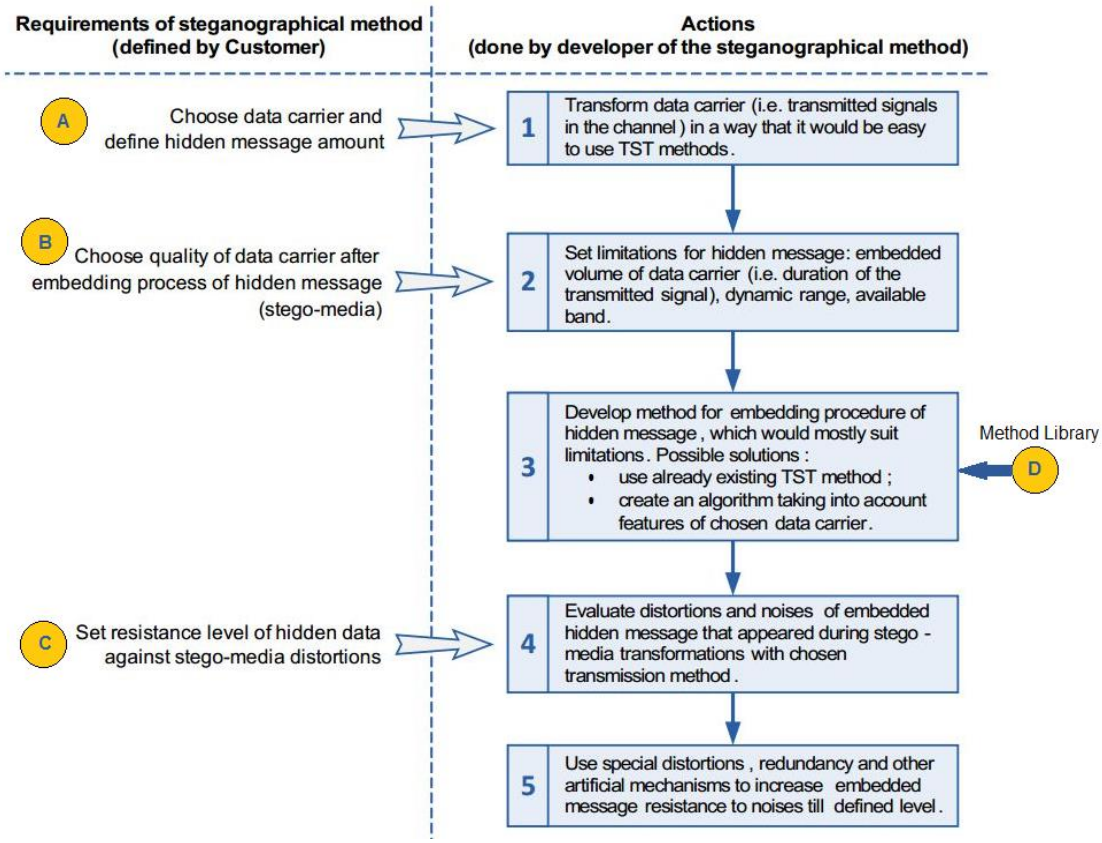

Fig. 6. The concept of the Universal Stegoconstructor [25] modified with the inputs from the approach introduced in the present research (A, B, C, D).

VI. Automated Selection of Steganographic Algorithms

Using the approach specified in Section III, it is possible to automatically select applicable steganographic methods or algorithms to the use case defined by the client.

The authors suggest that a knowledge base of steganographic algorithms is gathered (by the developers or possibly even the customer) with all characteristics, criteria and constraints. Then by using the requirements from the specification, the criteria for an algorithm for this use case can be defined.

Afterwards each method in the knowledge base is compared by each criterion whenever it is acceptable by the defined requirements. In this way methods can be filtered – ones that cannot be used for this use case will not satisfy minimum requirements. If no methods are usable for the client's specifications, then a new algorithm should be developed or the client needs to lower his requirements.

The mentioned automated selection is possible in a simple use case, but it has several challenges if applied practically:

- There are no universally accepted methods or guidelines to benchmark steganographic algorithms in all criteria (see [26] as an example of such guidelines), and the descriptions of algorithms usually are not directly comparable;
- Most descriptions of steganographic algorithms do not have publicly available implementations, which do not allow comparing methods empirically to fill the needed knowledge base;
- Some algorithms perform differently based on the container and message used, so all methods need to be tested on the same set of containers that the client needs, which can prove to be a challenge;
- Not all steganalysis methods are publicly available, which makes it hard to evaluate the security of the selected algorithm.

VII. Conclusion

The paper has presented guidelines for the development of requirements specification for steganographic systems using an original systematic approach based on the customer use case. The base criteria for non-functional requirements have been defined and grouped as well as their connections to various use cases of digital steganography. The approach further extends the previous research of the authors – the concept of the Universal Stegoconstructor in the context of developing requirements for the system. The research has also provided the option of automated method selection based on the criteria defined by the customer.

While developing the requirements specification, the customer should be aware of the possibilities that steganographic and digital watermarking systems can provide, and specific criteria that can be defined for the developers. Yet it is necessary to understand that if all criteria are set to high

values, the developer will be unable to find such an algorithm to satisfy all of them (see the "magical" triangle in Fig. 4), so each and every criterion must be weighted carefully.

Further research will focus on the development of steganographic systems based on specifications created using the presented guidelines, creation of a knowledge base of publicly available steganographic methods as well as improving the automatic method selection capabilities.

REFERENCES

[1] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich and T. Kalker, *Digital Watermarking and Steganography*, 2nd ed. Burlington: Morgan Kaufmann, 2008, 624 p.

[2] C. Everett, "Should encryption software be banned?" *Network Security*, vol. 2016, iss. 8, pp. 14–17, Aug. 2016. https://doi.org/10.1016/S1353-4858(16)30078-2

[3] P. Mozur, *The New York Times. New Rules in Chine Upset Western Tech Companies*. 2015. [Online]. Available: http://www.nytimes.com/2015/01/29/technology/in-china-new-cybersecurity-rules-perturb-western-tech-companies.html [Accessed: Oct. 9, 2016].

[4] I. Sommerville, *Software Engineering*, 10th ed. USA: Pearson, 2015, 816 p.

[5] *IEEE Recommended Practice for Software Requirements Specifications*, IEEE Std 830-1998, 1998. https://doi.org/10.1109/IEEESTD.1998.88286

[6] MITRE. (2016) *Develop System-Level Technical Requirements*. [Online]. Available: https://www.mitre.org/publications/systems-engineering-guide/se-lifecycle-building-blocks/system-design-and-development/develop-systemlevel-technical-requirements [Accessed: Oct. 8, 2016].

[7] G. Konahovich and A. Puzirenko, *Digital Steganography, Theory and Practise - Компьютерная стеганография, теория и практика*. Kiev: MK-Press, 2006, 286 p. (in Russian).

[8] N. Hamid, A. Yahya, R. B. Ahmad and O. M. AI-Qershi, "Image Steganography Techniques: An Overview," *International Journal of Computer Science and Security (IJCSS)*, vol. 6, no. 3, pp. 168–187, June 2012.

[9] A. H. Lashkari, A. A. Manaf and M. Masrom, *Magic Hexagon Image Steganography Evaluator*. [Online]. Available: http://www.academia.edu/1014807/Magic_Hexagon_Image_Steganography_Evaluator [Accessed: Oct. 8, 2016]

[10] V. Gribunin, I. Okov and I. Turincev, *Digital Steganography, Aspects of Protection - Цифровая стеганография, аспекты защиты*. Moscow: Solon-press, 2009, 265 p. (in Russian).

[11] A. Jersovs, V. Zabiniako and P. Semencuks, "Using Concatenated Steganography for Visual Analysis in GIS SOA," *Applied Computer Systems*, vol. 13, pp. 74–82, Nov. 2012. https://doi.org/10.2478/v10312-012-0010-6

[12] T. Morkel, J. H. P. Eloff and M. S. Olivier, "An Overview of Image Steganography," in *Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005)*, Sandton, Africa, June/July 2005.

[13] M. Kharrazi, H. T. Sencar and N. Memon, "Cover Selection for Steganographic Embedding," in *International Conference on Image Processing*, Atlanta, Georgia, USA, 2006, pp. 317–326. https://doi.org/10.1109/icip.2006.312386

[14] Spammimic. (2015). *Spam mimic*. [Online]. Available: http://www.spammimic.com/ [Accessed: Oct. 9, 2016].

[15] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms*, 3rd ed. Massachusetts: The MIT Press, 2009, 1312 p.

[16] M. K. Kundu and S. Das, "Lossless ROI Medical Image Watermarking Technique with Enhanced Security and High Payload Embedding," in *International Conference on Pattern Recognition*, Istanbul, Turkey, Aug. 23–26, 2010, pp. 1457–1460. https://doi.org/10.1109/icpr.2010.360

[17] International Telecommunication Union. (2012). *Methodology for the subjective assessment of the quality of television pictures. Recommendation ITU-R BT.500-13*. [Online]. Available: https://www.itu.int/rec/R-REC-BT.500-13-201201-I/en [Accessed: Oct. 8, 2016].

[18] R. Böhme, "Principles of Modern Steganography and Steganalysis," *Information Security and Cryptography* (Information Security and Cryptography), pp. 11–77, 2010. https://doi.org/10.1007/978-3-642-14313-7_2

[19] H. Sajedi, "Steganalysis based on steganography pattern discovery," *Journal of Information Security and Applications*, vol. 30, pp. 3–14, Oct. 2016. https://doi.org/10.1016/j.jisa.2016.04.001

[20] D. Lerch-Hostalot and D. Megias, "Unsupervised steganalysis based on artificial training sets," *Engineering Applications of Artificial Intelligence*, vol. 50, pp. 45–59, Apr. 2016. https://doi.org/10.1016/j.engappai.2015.12.013

[21] T. Sloan and J. Hernandez-Castro, "Steganalysis of OpenPuff through atomic concatenation of MP4 flags," *Digital Investigation*, vol. 13, pp. 15–21, June 2015. https://doi.org/10.1016/j.diin.2015.02.002

[22] G. Luo and X. Sun, "An Evaluation Scheme for Steganalysis-proof Ability of Steganographic Algorithms," in *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Haohsiung, Taiwan, Nov. 26–28, 2007, pp. 126–129. https://doi.org/10.1109/iih-msp.2007.85

[23] J. Cummins, P. Diskin, S. Lau and R. Parlett. (2004). *Steganography and Digital Watermarking*. [Online]. Available: http://www.cs.bham.ac.uk/~mdr/teaching/modules03/security/students/SS5/Steganography.htm [Accessed: Oct. 8, 2016].

[24] D. Kahn, *The Codebreakers*. New York: Scribner, 1996, 1224 p.

[25] A. Jersovs and P. Rusakovs, "Kutter Steganographical Method's Improvement and Concept of Universal Stegoconstructor," in *Computer Sciences*, vol. 38, pp. 198–208, Jan. 2009. https://doi.org/10.2478/v10143-009-0018-6

[26] M. Kharrazi, H. T. Sencar and N. Memon, "Benchmarking steganographic and steganalysis techniques," in *Proceedings of the SPIE*, vol. 5681, pp. 252–263, March 2005. https://doi.org/10.1117/12.587375

Dāvids Grībermans was born in 1990. The degrees obtained: Mg. sc. ing. (2015), Bc. sc. ing. (2013) – Riga Technical University (RTU). Diploma with distinction: Mg. sc. ing.
He is an Oracle Programmer at IT company "Lattelecom Technology". Fields of interest include steganography, hidden communications, copyright protection of digital media, database technologies.
E-mail: st16@griberman.com

Andrejs Jeršovs was born in 1984. The degrees obtained: Mg. sc. ing. (2008), Bc. sc. ing. (2005) – Riga Technical University (RTU), Institute of Applied Computer Systems.
He is an IT Project Manager at the Latvian company "ABC software". Fields of interests include computer science, data integration. Special interests: programming paradigms, distributed systems, Web technologies, steganography and steganalysis, cross-platform development, and signal processing.
E-mail: Andrejs.Jersovs@rtu.lv

Pāvels Rusakovs was born in 1972, in Riga, Latvia. The degrees obtained: Dr. sc. ing. (1998), Mg. sc. ing. (1995), Bc. sc. ing. (1993) – Riga Technical University (RTU). Diploma with distinction: Mg. sc. ing.
He is an Associated Professor at the Institute of Applied Computer Systems, RTU. He is the Head of Laboratory, responsible for the Professional Bachelor and Master Studies at the Department of Applied Computer Science. Field of interest is computer science. Special interest: programming paradigms, object-oriented approach to systems development, parallel computing, web technologies, distributed systems, computer graphics, and protection of information.
E-mail: Pavels.Rusakovs@cs.rtu.lv