RESEARCHING THE FEATURES OF PRACTICAL USE OF THE SOFTWARE COMPLEX FOR ATTACKING THE LINGUISTIC STEGOSYSTEM

Yaroslav Tarasenko, Oleg Piven Candidate of Physical and Mathematical Sciences, Associate Professor, Iryna Fedotova-Piven Candidate of Technical Sciences, Associate Professor

Ukraine, Cherkasy, Cherkasy State Technological University:
1 Postgraduate Student of Department of Information Security and Computer Engineering
2 Professor of Department of Information Security and Computer Engineering
3 Head of Department of Information Security and Computer Engineering

DOI: https://doi.org/10.31435/rsglobal_ws/12062018/5809

ARTICLE INFO
Received: 27 April 2018
Accepted: 24 May 2018
Published: 12 June 2018

KEYWORDS
software complex, linguistic steganography, counteracting the methods of steganography, semantic compression, automated linguistic steganalysis, practical value of attacking stegosystems by compression, removal of hidden message

ABSTRACT
The article is devoted to the practical aspects’ experimental study of application the developed software complex for attacking the linguistic stegosystem. In addition to the basic settings, 6 more combinations were selected to perform the most common tasks of cybersecurity in countering the computer linguistic steganography. These combinations were the basis for carrying out the study of the program's work and making conclusions about the field of each combination’s possible use. Based on the testing results in context of the initial settings combinations study, an idea about the dependence of the text modification time on its volume is given, which determines the value of selected parameters. Finally, it was made a conclusion on the possibility of developed software efficiency forced increase due to a flexible system of settings. Recommendations on the software complex application area are given and its practical value is proved.

Introduction. Today many spheres of modern people life are connected with the use of developing methods and computerized means of steganography. This is due to the need of hiding certain tags, copyright signatures, watermarks [1] or even messages in any files. Steganography can be used in data protection, for example, when it comes to media files copyright protection or prevention the falsification in patient medical records described in [2], as well as secret commercial or state information leak prevention: commercial espionage, terrorism [2] etc. In addition, one cannot even know about the use or dissemination of secret data with the proliferation of multi-machine data processing, which makes the steganography even more dangerous. However, the linguistic steganography cares the greatest threat. Its main directions are described in [3]. While lexical and phrase or syntactic transformation can be recognized by computer programs such as mail scanners, but if it comes to semantic methods or artificial text generation in order to hide a stegomessage, such means of protection are not effective, due to the possibility of easily overcoming protection by means of linguistic generation software, such as, for example, NICETEXT [4] which generates text, similar to the natural one. A software complex for attacking a linguistic stegosystem [5] was created to provide effective counteraction to the full range of linguistic steganography methods. It is based on the textual semantic compression and manipulations which remove any stegomessage, preserving the semantic structure of the original text. That is why testing of this software is an actual task.

The aim of the article is to test the work of the software complex, depending on the combinations of initial settings to identify the possibilities of its use in specific situations and to provide some practical recommendations for its application.
The following tasks were set for achieving the aim:
1. To select the most suitable combinations of initial settings of the software complex.
2. To conduct an experimental study of its work with each setting combination based on the execution time and the text volume.
3. To make appropriate conclusions about the area of use for each setting combination on the basis of testing.
4. To give recommendations on the software complex application area and its practical value as a result.

**Research results.** After a primary analysis of the operability and efficiency of the software complex, it was made a research on its practical use opportunity. Stress testing [6] was the main method of researching.

If it comes to the operations speed, then many factors affect it. First of all, if the necessary data is indicated before the system starts, or the system must determine the discourse characteristics itself. Secondly, the volume of the text directly influences the speed of data processing. In addition, if there is a need of extremely small texts compression, the number of which exceeds one thousand unrelated messages, then the speed will also significantly decrease. Thirdly, the combination of initial settings of the system also affects the speed of data processing. And of course, settings and specifications of the computer on which the program is installed also determine the speed of its execution. Thus, there is a combination of initial basic settings and custom sets available. In order to determine the most optimal combination from the operations time point of view, aimed at compression and modification of the text, the following possible combinations of settings were tested:

1. Basic settings, where style is not specified, intertextual verification functions, meaningless texts compression, the text purpose definition, one-time modification and the superficial compression tracking are enabled.
2. Combination 1. The style is not specified, the intertextual verification function, meaningless texts compression, the text purpose definition are disabled; one-time modification and the superficial compression tracking are enabled.
3. Combination 2. The style is specified, the intertextual verification function, meaningless texts compression, the text purpose definition, double modification are disabled; the superficial compression tracking is enabled.
4. Combination 3. The style is specified, the intertextual verification function, the text purpose definition are enabled; meaningless texts compression, double modification, the superficial compression tracking are disabled.
5. Combination 4. The style is not specified, the intertextual verification function, the text purpose definition, double modification, the superficial compression tracking are disabled; meaningless texts compression is enabled.
6. Combination 5. The style is not specified, the intertextual verification function, the text purpose definition, double modification, the superficial compression tracking are enabled; meaningless texts compression is disabled.
7. Combination 6. The style is not specified, the intertextual verification function, meaningless texts compression, the text purpose definition, the superficial compression tracking are disabled; double modification is enabled.

Thus, combinations where certain functions are disabled and other are enabled have been tested in comparison to the basic combination of system settings to guide the program in right direction for the most effective and rapid its execution according to a certain practical task or situation.

However, only in consideration that any of the mentioned functions may not be necessary in some cases their use is optional and changing the combination is applied to speed up execution of the program complex or when its operator decides to apply a special strategy as an exception to the rules. In such case using custom combination sets is possible, and this possibility can easily overcome any difficulty. Otherwise, an automatic modification is used to ensure that the stegomessage is fully deleted and cannot be restored.

Comparing the textual compression time depending on its volume in all noticed combinations of settings is shown on the graph (Figure 1), where the X-axis indicates the volume of text, and the Y-axis defines the execution time of text processing.
As it is shown on the graph, each combination has its own properties that influence the prospect of its usage. Thus, it can be assumed that the basic set of parameters is universal. It can be applied to texts of any volume, complexity and method of stegomessage hiding. However, it should be noted that the time of text modification will be the longest and will increase with the text volume increasing. Combination 1 is the best for analyzing brief personal messages or separate sentences. To study meaningful texts of large volume combination 2 is the most suitable because it demonstrates the smallest modification time of large texts. Combination 3 can be used to analyze texts in publicistic style, which are placed in the Internet, since it has the possibility of intertextual search in spite of longer execution time. Combination 4 is effective for studying a large textual data flow of various topics in order to identify among them meaningless texts which have a hidden stegomessage inside and neutralizing it. Combination 5 is necessary for compressing texts, where it is known about a very high chance of stegomessage presence. Combination 6 is required to modify the text array of ultra-low volume where texts are not related to each other.

Obviously, one can use other unique combinations of system settings for special tasks. However, described combinations illustrate the dependence of text modification time on the selected system setting and volume of the text. Taking into account the specific task that involves the study of target text or data set can be recommended for performing practical tasks. Discourse description such as text style, its main theme should also be pre-specified, instead of selecting settings by default. Thus, as it is shown at Figure 1, the execution time of the system can be significantly reduced and steganalysis efficiency can be improved in context of the particular situation needs. It follows that, although the effectiveness of the program complex is proved, but the opportunity for increasing the steganalysis efficiency and as a result the efficiency of attacking stegosystem is provided by the user before or in the course of the program execution.

Thus, the practical value of software complex is caused by the ability of using the developed program for automated detection of meaningless texts, as well as for automated computer attacks on linguistic stegosystem. That’s why a significant increase in speed and efficiency of processing large volumes of textual data is provided. At the same time the comprehensive approach to steganalysis offers a wide range of software application possibilities for solving lots of information security practical problems related to counteracting cyberterrorism, the leak of classified commercial and government information and providing individual, business and government agencies security.

Conclusions.
1. It was selected and described 6 combinations of software complex configuration for performing the most common practical tasks.
2. The dependence of the operations execution time on the text volume was determined for each combination as a result of experimental research.
3. The application area of all initial settings combinations is detected on the basis of testing.
4. Recommendations on the implementation area of the software complex are given.
5. The possibility of the efficiency forced increasing depending on the practical problem is proved.
6. The high practical value of the software complex is substantiated.
REFERENCES


