2021 IEEE 3rd International Conference on Advanced Trends in Information Theory (ATIT) | 978-1-6654-3847-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/ATIT54053.2021.9678855

Information Transmission Protection Using Linguistic Steganography With Arithmetic Encoding And Decoding Approach

Maksym Ivasenko
Software Systems and Technology Department
Taras Shevchenko National University of Kyiv
Kyiv, Ukraine
ivasenko182@gmail.com

Olha Suprun
Department of Cybersecurity
National Academy of the Security Service of Ukraine
Kyiv, Ukraine
o.n.suprun@gmail. com

Oleh Suprun
Department of Intelligent Software Systems
Taras Shevchenko National University of Kyiv
Kyiv, Ukraine
oleh.o.suprun@gmail.com

***Abstract* — Security of Internet of Things and Internet space in general is one of the main areas of research in the field of information technology today and steganography - one of the best methods of message encryption and transmission. Steganography algorithms are used to hide data in cover media, such as text, images, videos, etc., as cover media and change the structure and features of a text message so that it is not identified by human eyes and machines. However, the use of text media as accompanying text is relatively difficult compared to other methods of information coverage due to the lack of available redundant information in the text data. At the same time, text data is being transmitted the fastest and most frequently, so it is ideal for quickly transmitting short important messagesю In this research, an overview of existing approaches to textual steganography and our approaches to steganography methods are presented. New approaches to text steganography, which easily hide the text-coverage and also take less time to encrypt, are presented. A new method of steganography is shown and tested; a software implementation of this method is created. The advantages of using a new method of steganography are substantiated. It is confirmed that this algorithm is faster when working with large amounts of data, comparing to existing algorithms. At the same time, it allows to choose different parameters and settings, increasing its protective potential.**

Keywords — steganography, linguistic steganography, encryption, decryption, text coverage

I. Introduction

Steganography is a method of encrypting information that is rather often confused with cryptography, but their meanings are quite different. Steganography hides the entire secret message inside a data object known as a cover message, while cryptography uses symmetric or asymmetric encryption algorithms to hide the content of the message that must be sent without attention. In other words, using steganography approaches, the presence of a secret message can't be detected without deep analysis, so that no one except the sender and recipient know about its presence and can discover it, while cryptography algorithms don't hide, the presence of a secret text, therefore the third person knows about the presence of a secret message.

Steganography is often incorrectly called concealment of information, but it is the art of embedding secret messages (steganograms) into specific cover data transmit them in secret. The boundary between the two areas cannot be clearly demarcated, as their definitions are not clear, and there is no consistent classification of invented methods of secret communication that attributes them to specific areas of steganography or concealment. These mistakes can be explained by the recent growing interest in steganography that has been shown in the media, which has shed light on only a small part of the available methods. The reports and news about espionage and terrorist activity that have emerged in the last decade have largely contributed to the development of these methods of concealing information related to the Internet and digital steganography, using images as covering data.

This main advantage of steganography is the following. Using approaches like this, the message can be sent over various communication channels using different types of cover messages, like text, images or other media, to hide the message existence. At the same time this applies the major limitation in the cryptography techniques. Another reason for using steganography is that some countries, such as the United States and the British government, prohibited using encryption because a third party can learn that secret important messages are being transmitted that is inacceptable.

Thus, steganography techniques use much more efficient methods of hiding text in cover media. Steganography approaches are also used to change the structure and features of a secret message in different ways, so that it can't be identified by human itself [1,2]. However, the use of text media as accompanying text is rather difficult in comparison with other algorithms of information coverage due to the lack of redundant information, available in the covering text data [3,4].

In this paper, an analysis of modern approaches to textual steganography and our target approaches to different steganography methods are presented. The disadvantages of most existing methods in text steganography are the use of large text-covering for the encryption and decryption processes to hide simple text messages, as well as unacceptable time used for encryption and decryption.

New approaches to text steganography, which easily hides text and also requires less time to encrypt, are presented. Also the paper shows their comparison to other modern algorithms, demonstrating its competitive ability.

978-1-6654-3847-6/21/$31.00 © 2021 IEEE

II. Existing stenography methods.

For the first time, steganography was mentioned in the 5th century BC, in the annals of Herodotus talked about armed clashes between Greece and Persia in the 5th century BC. According to Herodotus, it was the art of secret writing that saved Greece from being enslaved by Xerxes, the "king of kings", the ruler of Persia.

Modern steganography can use every system of human communication and data, created by humans, in general.

The most common media are images, audio, text massages, or video samples and video files. Among these coverage possibilities, text data is preferred since it requires less memory than the other coverage files mentioned above. Text encryption is better and more widely used to encrypt more bytes of plaintext in less memory and transfer more important information with less operation time. However, text steganography is the most complex of all because it does not contain any duplicate information, such as an image, video or audio file. The diversity of the coverage file is more obvious in text steganography. Text coverage is faster and easier to read, making it more user-friendly than other steganography techniques. The classification of text steganography into three types is the following: format editing, random sequence and statistical, and finally, language-based algorithms.

A. Methods based on formatting

Formatting-based methods use covering text formatting to hide the secret data [5]. Not a word or sentence is changed in this method, so the original meaning of the covering text is the same. The white space, color changes, font's size or type, etc. is used to hide the required text.

Text formatting steganography is a method of open space and method of coding of text features. The use of spaces and deliberate distortion of spelling to hide classified information can be discovered by an analyzing program, while people perceive it as usual text. Similarly, a person can recognize changes in the type, size or color of text that the computer considers normal message. It should be noted that in the case of reformatting or reprinting, the text will lose confidential data.

B. Statistical and random generation

Random and statistical generation approaches are used to form the covering text in accordance with the statistical attributes of the native language. The method of random generation is applied in order to avoid a predicted attention of the third person on plain text. Random context-free changes in grammar are a widely used language tools for generating random coverage text.

The random sequence of coverage should seem random after hiding secret information for everyone except those who communicate. Another approach to this method is to create random words using the statistical properties of word length and letter frequency. Due to chance, both random sequences of characters and a random sequence of words are meaningless, which is why it can be very suspicious.

C. Linguistic approach in steganography

In the linguistic method the linguistic properties of the text are used to modify itself [6,7]. This approach uses the message linguistic structure to hide classified message.

The syntactic method is a method of linguistic steganography in which random punctuation marks, such as a comma (,) and a dot (.) are placed in the appropriate places in the covering message to hide data. This method requires correct usage of places where characters may be inserted.

Other linguistic approach of steganography is the semantic method. In the following, discuss some general existing approaches to textual steganography will be discussed. Also the linguistic steganography has the following sub-methods:

1) The method of open space – spaces are used to hide secret information. According to this method the meaning of the original text is not changed, and the third party considers additional gaps as plain text.

2) Line shift method – text lines are shifted vertically to some extent, and the message is hidden, creating an exeptional form of text. The up and down shift method is used to hide bits "0" and "1", respectively.

3) Word shift method – instead of string shift, words can be shifted horizontally according to the secret bits.

4) Syntactic method – this method is implemented by placing punctuation marks such as a comma (,) and a dot (.) in the appropriate places to hide confidential information.

5) Semantic method – This approach has an advantage over others that is based on hiding the information in case of reprinting [8]. The ynonyms of the real words of the accompanying text are used to conceal classified message. Bit "1" is hidden by the actual word, and bit "0" is covered by the synonyms of the word. This approach is widely used in text recognition programs.

These 5 methods change the appearance of the text, or its structure, so it can be obvious that there is a hidden massage inside.

In all of the above approaches, there is no method of verifying whether a third party intrusion is present or not. This puts our approach above all previous methods. In addition, both the need for memory to store cover text and the time required to perform this approach are very small. Thus, the proposed approach reduces the number of bytes of text coverage required to encrypt (hide) a message or information. In our approach, overlay text symbols are used to validate data on the recipient side. The very next section describes our approach and its implementation.

III. General description of the algorithm

The proposed modified algorithm (PECT) is based on simple operations with the UTF-8 code for each character in the coverage text and symbols from the classified message. For parallel verification of data on the recipient's side, a method of self-control based on vowels and special characters (hereinafter - all other characters of the table UTF-8 Ukrainian, Russian, English, German alphabets, except vowels) is presented. Different keys for every character of the classified message are being generated, and due to the different values of these keys, our method becomes complex and difficult to decrypt by attackers.

The PDAC approach in using the parallel method to reduce CPU consumption and small text coverage has helped to cover more private data in a small repository. But the PDAC has no verification methods. Therefore, in the case of damage to any data packet, the complete data must be sent once more. To resolve this situation, the PDAC is divided into many steps, and all of them are processed in parallel to reduce temporary overhead.

In the initial steps, a method of self-control based on special characters and vowels is used:

- The location of vowels and characters in the data is checked and bit arrangement according to it is made;
- Based on the resulting bit pattern, the bits are divided into groups of four to get a decimal number;
- The resulting decimal number is then converted to UTF-8 code after adding 60. This UTF-8 code forms the coverage text for personal data. Therefore, we do not need any additional space to verify our data on the recipient side. The sent message contains only the secret message and the accompanying text. Therefore, instead of random text-coverage, as in PDAC, in PECT, we have automatically generated text-coverage. The reason for this is that it would not increase the space. Thus, the method uses the same amount of space as PDAC.
- After receiving the text-coverage, the PDAC approach to obtain encrypted data follows. The decimal number comes from the alphanumeric text-cover. First of all, the text-coverage comes from the encrypted data every fifth character of the data.

Therefore, its decimal form is obtained. Then 400 is added and subtracted from the obtained four-digit number. The value 400 was chosen to avoid creating an encrypted message that would contain special UTF-8 encoding characters. When adding 400 to the decimal representation of the symbol, "Extended Latin - A" is used.

The decimal numbers of the cover text are added and subtracted independently. Now the obtained four digits are of exceptional importance for the operation of private data. The result is encrypted text, which is then encrypted with a key. The result is sent to the recipient by adding a cover text to each of them. The process begins with the generation of a random text-cover the size of 1/4 of our secret message, rounded up to the whole. We use parallelism to reduce time and hide large amounts of data in the text-coverage. After generating the encrypted message, it is hidden in the same text message-coverage in such forms that one character of the text-coverage with four characters of the encrypted message, which were encrypted using the same text-coverage.

Proposed method is complex compared to the existing steganography, approaches, so it would be difficult to calculate the original text from the encrypted text for the casual reader. This approach uses self-monitoring based on vowels, consonants, and characters with simple addition, subtraction, and X-OR operation to encrypt plain text and hide it in our random text-coverage so it's quick to calculate. The maximum of N/4 (rounded up to an integer) byte of coverage text is needed to hide N bytes of the secret data.

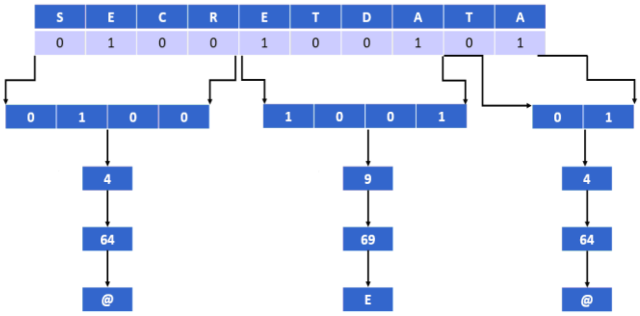
At the same time, using UTF-8 coding makes it quite easy to use, since this system is installed into every operation system. Furthermore, for additional complexity non-English characters can be used, like Japanese. This allows making the appearance of encoded text more random, or even looking like the massage in different language, that will make the identification impossible without further investigation.

IV. Algorithm Performance Demonstration

To show the algorithm efficiency, its process can be shown on a simple short message that consists of 2 words, let it be collocation “secret data”. In the algorithm two attributes of each letter are used – its number in UTF-8 code, and a boolean value, 1 if the letter is vowel and 0 for consonant letter.

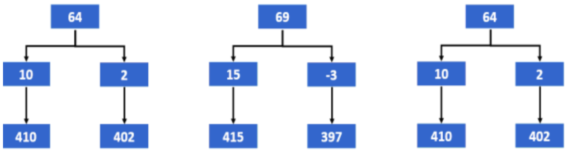
A. Encoding process

1) As the first step of coverage text creation, the vowel attribute of the letter is used:



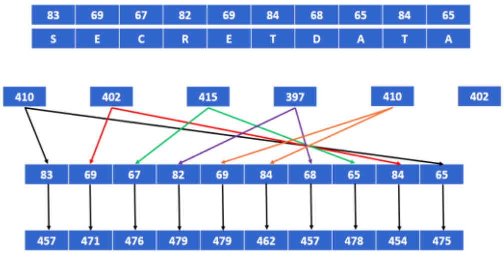
According to the algorithm, each four letters are used to form a set of 0 and 1 values that are transformed into decimal number system, 60 is added to this number, and the resulting value is transformed into a sign according to UTF-8 code. Also it must be noted that in the case when the number of received characters at the input is not a multiple of 4, we still create an additional stego-key for the cipher "incomplete" group. When converting a binary code, the "incomplete" group is counted from left to right. For example, in the above case, code 01 will have an equivalent in decimal notation in the decimal number system - 64: 0 * 2 ^ 3 + 0 * 2 ^ 2 = 0 + 4 = 4.

2) To obtain the stego-key, the UTF-8 code from step 1 is used:



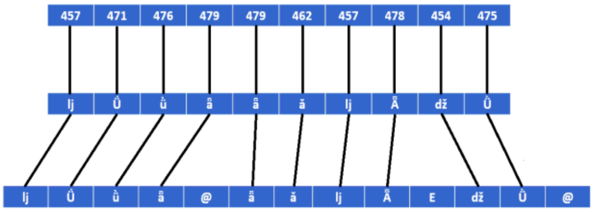
For each value of the code we get two numbers as follows: the first is obtained by adding the digits of the original value, the second – subtracting. Then 400 is added to each valueю

3) To encrypt the original message, its representation using UTF-8 and the key generated in the last step are used:



The resulting numbers are being calculated by parallel using operation X-OR with obtained values.

4) The resulting text is being obtained by combination of steps 3 and 1.

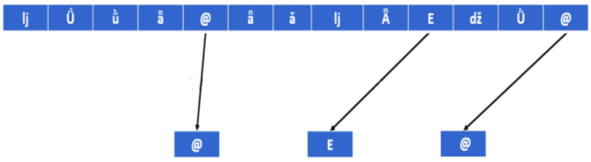


At first, the values from step 3 are transferred into characters, using UTF-8. After that the characters, obtained calculating the cover-text are inserted into every fifth position of the text.

Therefore, encoded text is ready for transfer.

B. Decoding process

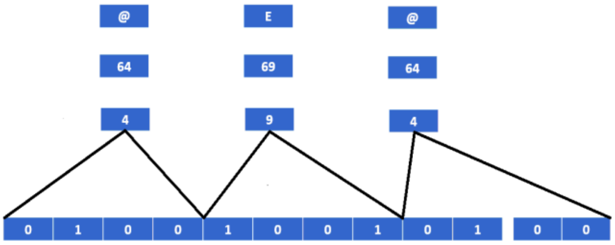
1) After reiciving the encoded text, on the first stape the characters of cover-text are being pulled out of the string:



These characters are also required on the further steps to get the stego-key.

In the case when the number of received characters at the input is not a multiple of 4, we still create an additional stego-key for the cipher "incomplete" group.

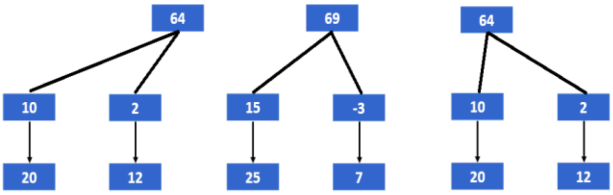
2) On the basis of the cover-text the bit template 0-1 is obtained:



The characters from previous step are transformed into numbers, using UTF-8, their value is subtracted from 60, after which the result is translated into a binary number system.

Note that the missing figures are added to the end of the line as 0.

3) The decryption process to obtain the UTF-8 code of the encrypted text is performed:

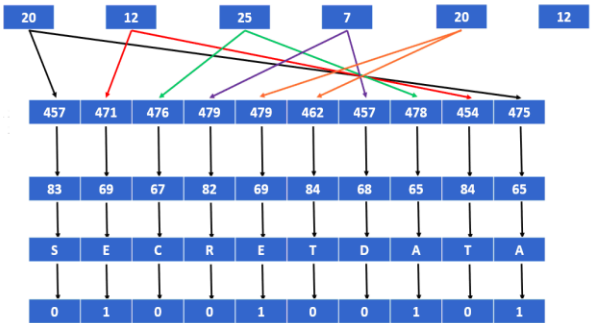


Two digits of each value of the cover-text symbol (according to UTF-8 from step 2) are added and subtracted from each other.

Also the string of values is obtained from initial text, excluding the characters of cover-text:

457	471	476	479	479	462	457	478	454	475

4) Finally, we write the encrypted text with the received numbers, using X.



In parallel, we also perform a 0-1 check to determine if the received message is corrupted.

V. Algorythm Implementation and Test Results

The project was performed using the following technologies:

- ASP .NET Core 3.1.4 + MVC
- Bootstrap 4\

The structure of the MVC architecture divides the application into three main groups of components: models, views, and controllers. This allows to implement the principles of division of tasks. According to this structure, user requests are sent to the controller, which is responsible for working with the model to perform user actions and (or) obtain the results of requests.

Top level is a view (UI) designed to interact with the user. Implemented by a separate project. ASP.NET Core MVC technology was used to create the Web-interface. At the UI level, only user interaction operations are implemented. The UI code is as simple as possible, not overloaded with a large number of operations. This part of the system (Front End) is implemented in this way – for stylistic design used Bootstrap framework. The data that the presentation level works with is stored in separate models of this level (meaning own classes (types) of the level, not borrowed from other levels). Provides control / verification of data entered by the user (text fields, length of entered data, etc. must be filled in).

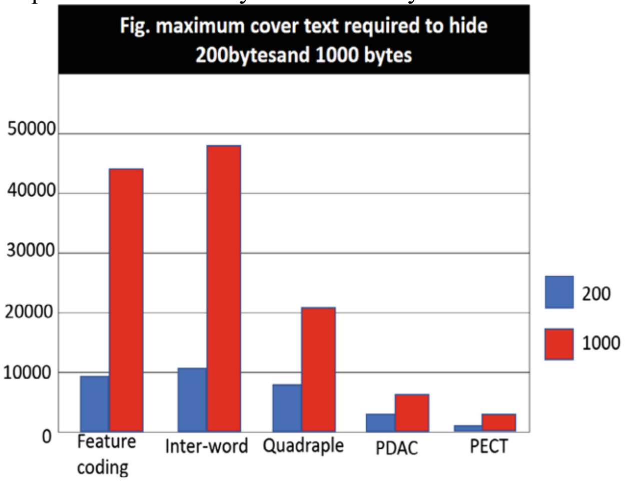
Experiments show that our approach has cases with different overhead times that have been calculated using compilation.

1. time of additional calculations is 6-8 MS, when the entire packet is damaged or an error is detected at the beginning or end of the message.
2. time is 12-15 MS, when an error occurs in the middle of the message.
3. time is 16-20 MS, when the message has no errors.

Thus, the average overhead time in the proposed PECT approach is 12-16 MS. Comparison to other methods is shown in the table below:

Text steganography method	Text massage size (in bytes)	Covering text size (in bytes)	Number of hidden bytes	Time (ms)
Feature coding	600	1980	52	20,289
Inter word space [13]	600	1980	45	22,604
CASE	600	1980	200	1,666
PDAC	600	1980	600	15-20
PECT	600	1980	600	12-16

Diagram below shows a comparison between different text steganography methods based on the text coverage required to hide 200 bytes and 1000 bytes.



Therefore the PECT approach is quite safe since it checks the symbols of the secret message received on the recipient's side. The binary pattern used for vowels and consonants is cross-examined to obtain the same pattern in the decoding process. In addition, after applying PECT, it can be made effective for big data. Now it can be used even on channels with high traffic, since only the part that is defective should be sent by the sender again, not the full message.

With the growth of technology and the wider use of digital technology, the steganography methods, which will work with big data and can be used on high-traffic channels, are required. In addition, the requirements for storage and temporal complexity of PECT are much lower compared to other methods of text steganography.

VI. Conclusions

A web application has been developed to demonstrate the performance of the algorithm on the ASP .NET Core 3.1.4 platform using the MVC template and the Bootstrap 4 framework. Using this application, the new method testing was conducted and results were compared to existing algorithms.

It is confirmed that the presented algorithm is faster working with large amounts of data, in contrast to existing algorithms. By simply adding, subtracting and comparing, we were able to dynamically encrypt the data. The time complexity of the overhead for comparison is ½ of the total number of bytes in the secret message. Due to concurrency, when we compare the first and last code of UTF-8 encrypted data together, the increased overhead time is n / 2. But we see that in the case of large messages, this method is useful for detecting any changes in the message before all decryption. In addition, the requirements for storage and temporal complexity of PECT are much lower compared to other methods of text steganography.

Also one of the benefits comparing to other methods is that presented method doesn't need any additional data to work, like transferring the encryption keys or other clues that can be very dangerous. For a third person, the transferred text seems to be illogical, like broken data that happens a lot in Internet.

Besides that, being rather simple, this algorithm doesn't require any additional knowledge to be performed, and at the same time it provides high security level. In the case of any data loss, it is rather easy to recover the needed data which is very important while working with big data.

Moreover, presented method can be easily modified and improved depending on current needs and requirements. For example, the number of parts, in which the text is divided, can be easily changed, therefore the decryption process must be started anew, and the text can't be decrypted without the knowledge about cover-text.

References

[1] M. Gaur, M. Sharma, "A new PDAC (Parallel Encryption with Digit Arithmetic of Cover Text) based text steganography approach for cloud data security," International Journal on Recent and Innovation Trends in Computing and Communication, vol. 3(3), 2015, pp. 1344–1352.

[2] S. Kataria, B.Singh, T.Kumar, H.S.Shekhawat, "PDAC (Parallel Encryption with Digit Arithmetic of Cover Text) Based Text Steganography," (2013)

[3] Chaudhary, S., Mathur, P., Kumar, T., Sharma, R.: A capital shape alphabet encoding (CASE) based text steganography. In: Conference on Advances in Communication and Control Systems, (CAC2S 2013), India (2013)

[4] Kataria, S., Singh, B., Kumar, T., Shekhawat, H.S.: An efficient text steganography using digit arithmetic. In: Fourth International Joint Conference of Advances in Engineering and Technology in Elsevier Science and Technology, NCR India, December 2013, vol. 6, pp. 155– 163 (2013)

[5] Agarwal, M.: Text steganographic approaches: a comparison. Int. J. Netw. Secur. Appl. (IJNSA) 5(1), 91–106 (2013) Kataria, S., Singh, K., Kumar, T., Nehra, M.S.: ECR (encryption with cover text and reordering) based text steganography. In: 2013 IEEE Second International Conference on Image Information Processing (ICIIP), Shimla, December 2013, pp. 612–616 (2013)

[6] Yudin, O., Suprun, O., Buchyk, S., Ziubina, R., Bondarenko, I. Devising a Method of Protection Against Zero-Day Attacks Based on an Analytical Model of Changing the State of the Network Sandbox: Eastern-European Journal of Enterprise Technologies, 2021, 1(9(109)), p. 50–57

[7] Shirali-Shahreza, M., Shirali-Shahreza, M.H.: Text steganography in SMS. In: 2007 International Conference on Convergence Information Technology, November 2007, pp. 2260–2265 (2007)

[8] Shirali-Shahreza,M.: Text steganography by changing words spelling. In: 2008 10th International Conference on Advanced Communication Technology, ICACT 2008, February 2008, vol. 3, pp. 1912–1913 (2008)