DOI: https://doi.org/10.53052/9788367652001.16

Ruslana ZIUBINA[1], Olga VESELSKA[2], Maksym HEDEON[3]

STEGANOGRAFICZNA OCHRONA INFORMACJI Z WYKORZYSTANIEM ZMODYFIKOWANEGO ALGORYTMU LSB

Streszczenie: Opracowano model matematyczny transformacji steganograficznej. Podano merytoryczne uzasadnienie wyboru algorytmu LSB przyjętego za podstawę pracy nad implementacją produktu programowego. Podano uzasadnienie wyboru danych wejściowych. Zasadniczą różnicą proponowanej metody jest wykorzystanie plików o różnych formatach jako ukrytej informacji.

Słowa kluczowe: steganografia, ochrona informacji, cyberbezpieczeństwo, WAV, LSB

STEGANOGRAPHIC PROTECTION OF INFORMATION USING MODIFIED LSB ALGORITHM

Summary: The mathematical model of steganographic transformation was worked out. A substantive justification for the choice of the LSB algorithm taken as the basis for the work on the software product implementation has been given. The rationale for the choice of input data was given. The main difference of the proposed method is the use of files of different formats as hidden information.

Keywords: steganography, information protection, cyber security, WAV, LSB

1. Introduction

A large number of active users of modern information technologies, the development of cybersecurity and cybercrime lead to the creation of the latest cryptographic and steganographic algorithms, their software implementation. The main purpose of creating such algorithms is to transmit confidential information in containers that inherently do not arouse suspicion among users (images, audio, video) and allow you to completely unobtrusively transmit any confidential information through open

[1] Akademia Techniczno-Humanistyczna w Bielsku-Białej, Wydział Budowy Maszyn i Informatyki, adiunkt: rziubina@ath.bielsko.pl

[2] Akademia Techniczno-Humanistyczna w Bielsku-Białej, Wydział Budowy Maszyn i Informatyki, asystent: oveselska@ath.bielsko.pl

[3] Taras Shevchenko National University of Kyiv: mhedeon@knu.ua

communication channels. This method of data transmission can be used to commit criminal intentions (1-3).

This is usually a complex process, and it is necessary to hide data transmission using different methods in different container formats. As it is known, the reliability of a hidden message in a container rapidly decreases as the size of the message increases. This means that an overflowing container can easily be identified and the data will be exposed (4-8).

2. Mathematical model of steganographic transformation

The process of steganographic transformation can be described as follows:

$$E: C \times M \rightarrow S, \tag{1}$$

$$D: S \rightarrow M, \tag{2}$$

$$S = \{(c_1, m_1), (c_2, m_2), \dots , (c_q, m_q)\} = \{s_1, s_2, \dots, s_q\}; \tag{3}$$

where: S– is a set of steganograms (filled containers).

Dependency (1) describes the process of hiding information, dependency (2) - the process of obtaining hidden information.

So, in general, a quilt system is a collection of containers (original and result), messages and transformations that connect them.

There are two ways to choose a container:

- arbitrary;
- selection of the most appropriate container in a particular situation.

3. Selection of input data

Uncompressed WAV audio files were chosen as a container for steganographic transformations. Taking into account the limitations of human hearing, data compression will not affect the perception of sound quality. And given the fact that the containers will be formed without compression - the end user will not feel the difference even if the container is full. Also, the undoubted advantage of this choice is that uncompressed audio files have a much larger output file size. This allows you to hide more information in the container, compared to the container without compression.

The size of such containers, which are uncompressed WAV audio files, directly depends on the sound depth, sound time and number of channels on the audio track.

As possible files for use as hidden information, files of any specific type or extension are not used. Since this significantly narrows the possibilities of using the software product and completely devalues its potential as a product that allows you to hide confidential information in WAV containers and transmit them through open communication channels. After all, information is not only a file of a particular type or extension, but it (meaning information in computer systems) is a certain set of bytes that is structured and logically loaded. Therefore, in the developed software product, the ability to use any type of file as confidential data is implemented.

4. WAV – files

WAV files are an example of the RIFF format created by IBM and Microsoft. The RIFF format is a "wrapper" for encoding audio in various formats.

Although WAV files can contain compressed audio, the most common WAV audio format is uncompressed Linear Pulse Code Modulation (LPCM) audio. LPCM is also the standard audio coding format for audio CDs. It is used to store two-channel LPCM audio. The sampling rate is 44100 Hz and each sample is 16 bits. Since LPCM does not compress and store all samples of an audio track, professional users or audio experts can use the WAV format with LPCM audio to get the best sound quality. WAV files can also be edited and processed relatively easily with software.

5. File integrity after the attack

The use of hash function is a typical method of data integrity verification that is widely used in various protocols and applications. It plays an important role in modern cryptography. The basic idea of a hash function is that the hash acts as a compact delegate, called a fingerprint or digital fingerprint of the input object. These functions in combination with the digital signature model are widely used to verify data integrity.

One of the main categories of hash functions is message authentication codes. This allows using a symmetric method to identify messages. This technique uses two main input parameters - message and key. The main goal is to simplify the combination with other data integrity mechanisms for various applications.

Checksum is one of the main methods of vertical verification. Its value depends on the comparison between the input object and the value obtained after encoding. This method is mainly used in conjunction with the calculation of hash functions. The method of comparing the checksum values of two objects helps to detect the integrity of the document. However, if the input and output checksums do not match, the document is considered to be modified and there is no possibility of its recovery. The stored checksum may be corrupted or modified. Another reason for the problem of recovering the checksum value is that when data cannot be recovered to obtain the checksum value, it is calculated using a one-way hash function.

6. Description of the method

The article presents the developed modification of LSB algorithm. The subspecies of LSB algorithm with changing only one least significant bit in the audio container was taken as a basis.

This choice was made in order to provide greater reliability in data hiding. Although this method leads to a directly proportional reduction in the possible file size that will be hidden in the audio container, it produces significantly less distortion and noise in the output filled container.

Most steganographic transformation algorithms based on the Last Significant Bit method have the ability to work only with text data streams. That is, it is either the ability to enter a limited amount of text from the keyboard, and then, when extracted, this text will only be displayed on the computer screen, or the ability to read only files with the extension .txt and extract hidden messages into the same files.

This creates quite significant limitations for the effective use of a software product that implements information hiding using steganographic methods. Therefore, in the process of preparing a container for hiding data in it - first read the header of the container itself, and then form an additional header.

First, we save the size of the file that you want to hide in the container - fileSize. This data takes 4 bytes. Next, save the size of the full file name that you want to hide in the container. This data will be needed when extracting the file from the container and takes 4 bytes. Next, save the full name of the file to be hidden, including its extension. These dates will also be needed to correctly extract the file from the container and occupy nameSize bytes. In general, such a header takes 4 + 4 + nameSize bytes.

This header is not written immediately after the main header of the WAV file, because then the data of the audio file used as a container would be damaged. This header will be added to the beginning of the main byte stream of the file to be hidden in the container and will be hidden in it together with the main file.

After the additional header is formed, the possibility of uniformly hiding the file in the container is calculated. Provided that such a possibility exists, the calculation of the uniform distribution of information in the container is performed. This possibility depends directly on the ratio of the number of bytes that can be hidden in the container to the size of the file to be hidden plus the size of the additional header.

$$N = \frac{\mathrm{Subchunk2Size} - \mathrm{extHeaderSize} * \mathrm{BitsPerSample}}{\mathrm{BitsPerSample}} \tag{4}$$

$$\mathrm{M} = \frac{\mathrm{N}}{\mathrm{HidingFileSize} + \mathrm{extHeaderSize}} \tag{5}$$

where:

- N – the maximum possible number of bytes that can be hidden in this container;
- M – coefficient for uniform distribution of the hidden file over the container;
- Sunchunk2Size – udio file data size (in bytes), excluding the header size, in which the audio track is stored;
- extHeaderSize – the size of the additional header (in bytes), which is formed in the process of preparing the container for steganographic transformations;
- BitsPerSample – audio stream depth (number of bits per sample);
- HidingFileSize – the size (in bytes) of the file that is hidden in the container.

The ability to distribute data hiding evenly over the entire possible length of the container does not allow to accumulate hidden data in one area of the container. This is possible provided that M > 0.

After all preparatory processes and calculations are completed, the process of hiding the file in the stegano container begins. To ensure greater reliability of the hidden file and container - we use cryptographic encryption of data before hiding them in the container.

As cryptographic encryption we use stream encryption based on a four-byte shift register with linear feedback.

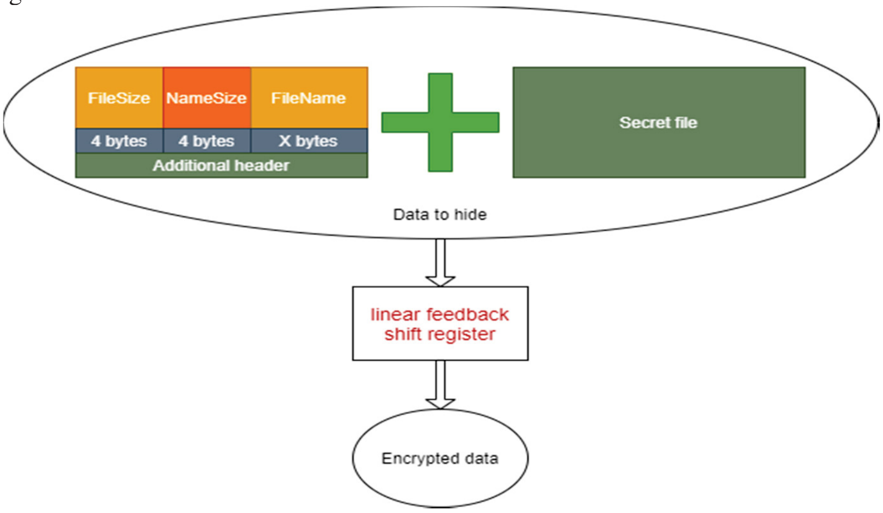

Figure 1. Logic scheme of the modified LSB algorithm

The undoubted advantages of choosing this method are:

high performance of the cryptographic algorithm, which allows streaming encryption of large amounts of memory, without overloading the working computer, and not slowing down the final software product;

- only the simplest addition and multiplication operations are used, which are implemented in hardware;
- good cryptographic properties.

7. Conclusions

In this article, the mathematical model of steganographic transformation was developed. A substantive justification for the choice of the LSB algorithm, which was taken as the basis for the work on the implementation of the software product, is given. The justification of the choice of input data was given. The main difference of the proposed method is the use of files of different formats as hidden information. This approach removes a number of restrictions for the user when creating steganocontainers, and also allows you to transfer documents with the desired formatting. The use of a cryptographic encryption method based on a four-byte shift register with linear feedback allows to increase the security of the transmitted confidential information on the one hand, and on the other - does not require significant resource costs.

REFERENCES

1. SIBURIAN R.: Steganography implementation on android smartphone using the LSB (least significant bit) to MP3 and WAV audio. 2017 3rd International Conference on Wireless and Telematics (ICWT), Yogyakarta 2017, 170-174.
2. INDRAYANI R.: Modified LSB on Audio Steganography using WAV Format. 2020 3rd International Conference on Information and Communications Technology (ICOIACT), Yogyakarta 2020, 466-470.
3. BINNY A., KOILAKUNTLA M.: Hiding secret information using LSB based audio steganography. 2014 International Conference on Soft Computing and Machine Intelligence, New Delhi 2014, 56-59.
4. MANE A., GALSHETWAR G., JEYAKUMAR A.: Data hiding technique: Audio steganographyusing lsb technique. International Journal of Engineering Research and Applications (IJERA), **2**(2012)3, 1123-1125.
5. ZHANG J., DU F., LI S.: Steganalysis of LSB Matching in WAV Audio. Conference of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013), Hangzhou 2013, 1027-1031.
6. PATIL R., PAWAR D.: Secure Audio Steganography by LSB for Hiding Information. International Journal of Innovations in Engineering Research and Technology, **3**(2015)4, 1-6.
7. ABOOD E. W.: Audio steganography with enhanced LSB method for securing encrypted text with bit cycling. Bulletin of Electrical Engineering and Informatics, **11**(2022), 185-194.
8. MARZUKI I.: Modification four bits of uncompressed steganography using least significant bit (LSB) method. 2012 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Depok 2012, 287-292.