A. V. Dudatiev, Cand. Sc. (Eng.), Ass. Prof.; P. V. Kozliuk; D. S. Oksymchuk

DEVELOPMENT OF ALGORITHM OF DIGITAL WATER SIGNS HIDING IN AUDIO FILES OF WAV FORMAT

There had been illustrated process of elaboration and analysis of stenographic algorithm, intended for hiding of text messages or binary data files in audio files of WAV format. Development of algorithm of digital water signs hiding in audio files of WAV format.

Key words: *DWS, stenography media files security, stenographic algorithms, data coding, C# programming.*

Introduction

Digital water sign (DWS) – is a technology created to protect the copy rights of multimedia files. DWS is mainly used for protection from copying and unauthorized usage [1]. The problem of protection of copyright and intellectual property presented in digital format became very actual and is caused by rapid development of multimedia technologies. Examples may be photos, audio and video recordings, etc. Advantages of message transfer in digital format may become useless due to stealing or message modification. Different measures of organization and technical character for information protection are developed. One of the most efficient technical means for multimedia information protection is built-in invisible lables in the object – DWS. Research in this sphere are carried out by the largest firms all over the world. Since these methods began to be developed quite recently, this sphere has many problems, to be considered. The name this method received from the well known method of securities protection, including money forgery (definition «digital watermarking») [5]. Unlike conventional watermark DWS may be not only visible but (as a rule) invisible. Invisible DWS are analyzed by special decoder, that makes decision regarding their correctness. DWS may contain authentic code, information about the proprietor or any control information. The most suitable objects to be protected with the help DWS are stationary images, files of audio and video data [4].

The aim of this article is elaboration of algorithm and software for hiding of digital water signs in audio files. Format wav is considered as a target format of audio file.

Let us consider the files structure of that format.

The first bytes in file WAV – are format identifiers.

```
typedef struct {
   char id[4];//- is file identifier = "RIFF" = 0x46464952
   long len;  // -is the length of file without this heading
} IDRiff;
```

The given code, described by C# language demonstrates programme realization of file heading structure. As in case of other containers, recognition is carried out namely by these bytes that is why there may be any file name extension.

Size and data format (makes up 24 bytes) and bitrate value (how many counts per second), number of channels (mono or stereo) are indicated in the simplest case after identification heading in WAV-file.

```
typedef struct {
        int type;   - type of sound data may be!!!
             1 – is simply a sample;
             0x101 - IBM mu-law;
                        0x102 - IBM a-law;
                        0x103 - ADPCM.
        int channels; - number of channels 1/2 - !!!
```

```
          long SamplesPerSec; - sample frequency- !!!
          long AvgBytesPerSec; - is a frequency of byte delivery
          int align; - aligning
          int bits; - is a bit number per sample - !!!
  } IDWave;
```

Subsequently the key word data followes, then data are allocated.

```
typedef struct {
     char id[4]; - identifier="data" =0x61746164
          long len;   - is a sample length (multiple 2 )
  } IDSampleWave;
```

If the format is without compressions then these data can represent 8-bit (one byte per each count) or 16-bit (two bytes per count) sound. If the number of channels is more than 1, then the counts for each channel are placed one by one, the first being the left channel and the second – the right. Such simple structure enables to use WAV for storing sequences of digitized signal, not only for audio usage but for other purposes (e.g. scientific-engineering usage).

Theoretical base of the algorithm

To start our algorithm elaboration we must determine requirements, to be met by our stegosystem. Namely these requirements will be criteria for creation of our methods, that is why let us consider them more thoroughly:

- hiden information must be immune to various coloured noises, compression with losses, filtration; analog-to- digital and digital-to-analog convertions;
- hidden information does not have to introduce in the signal distortions, perceived by human sensor system ;
- the attempt to remove hidden information must lead to visible damage of the container (for DWS);
- hidden information is not to introduce visible changes to container statistics.

Accordingly programme realization of the algorithm has to satisfy the above- mentioned requirements, maintaining the simplicity of usage and functionality. Let us consider theoretical essence of block-wise integration of DWS algorithm in target file by means of redundant bits substitution.

DWS is introduced in audiosignals(sequence of 8 or 16-bits counts) by means of minor amplitude change of every count. It is not necessary to have output signal for DWS identification [2].

Let audio signal consist of N counts x (i), i = 1, ..., N, where N value is not less than 88200 (accordingly, 1 second for stereo audio signal, quantized at the frequency of 44,1 kHz). To introduce DWS, function f(x(i), w(i)) is fulfilled, where w(i) – is DWS count, being changed within the boundaries [-α; α], α - is certain constant. Function f has to take into account peculiarities of human hearing system, to avoid tangible distortions of output signal. Count of resultant signal is obtained in such way

$$(i) = x(i) + f(x(i), w(i)). \quad (1.1)$$

Signal-noise ratio in this case is calculated as

$$\mathrm{SNR} = 10 \log_{10} \frac{\sum_n x^2(n)}{\sum_n [x(n) - y(n)]^2}$$

It is important to mention, that generator of random number , used in the circuit must have even distribution. In general case DWS stability is increased with DWS energy increase, but this increase is limited from top by acceptable signal-noise ratio.

DWS determination is fulfilled in the following way. Let us denote by S the next sum

$$S = \sum_{i=1}^{N} y(i)w(i) \quad . \tag{1.2}$$

Having combined (1.1) and (1.2) we will obtain the following

$$S = \sum_{i=1}^{N} \left[x(i)w(i) + f(x(i), w(i))w(i) \right] \quad . \tag{1.3}$$

The frst sum equals zero, if numbers at the output of RNG are distributed evenly and mathematical expectation of signal value equals zero. In the majority of cases certain difference is observed, that is denoted by Δw, it must be take into account.

Consequently, (1.3) obtains the form

$$S = \sum_{i=1}^{N} \left[x(i)w(i) + f(x(i), w(i))w(i) \right] \quad .$$

The sum $\sum_{i=1}^{N-\Delta w} x(i)w(i)$ approximately equals zero. If DWS has not been introduced in audio signal, then S will be approximately equal $\frac{\Delta w}{N} \sum_{i=1}^{N} x(i)w(i)$. On the other hand, if DWS has been introduced in audio signal, then S will be equal $\frac{\Delta w}{N} \sum_{i=1}^{N} x(i)w(i) + \sum_{i=1}^{N} f(x(i), w(i))w(i)$. But x(i) – is an output signal, according to condition it can not be used in DWS identification process. Signal x(i) can be substituted by y(i), this will lead to substitution of $\sum_{i=1}^{\Delta w} x(i)w(i)$ by $\frac{\Delta w}{N} S$, but the error will not be significant.

Consequently, subtracting the value $\frac{\Delta w}{N} S$ from S and having divided the result into $\sum_{i=1}^{N} f(y(i), w(i))w(i)$, we will obtain the result r, normalized to 1. DWS detector used in this method, calculates the value of r, set by the formula

$$r \hat{=} \frac{S - \frac{\Delta w}{N} |S|}{\sum_{i=1}^{N} f(y(i), w(i))w(i)} .$$

Threshold value of determination theoretically is within the range of 0 and 1, taking into account the approximation, this interval is reduced to [0 - ε; 1 + ε]. Empirically, it is determined, that to define whether certain DWS is located in the signal, threshold value of DWS must be greater than 0,7 [6]. If great validity is necessary for determination of DWS availability in the signal, threshold value is to be increased.

Fig 1 shows empiric function of probability density for audio signal with and without DWS

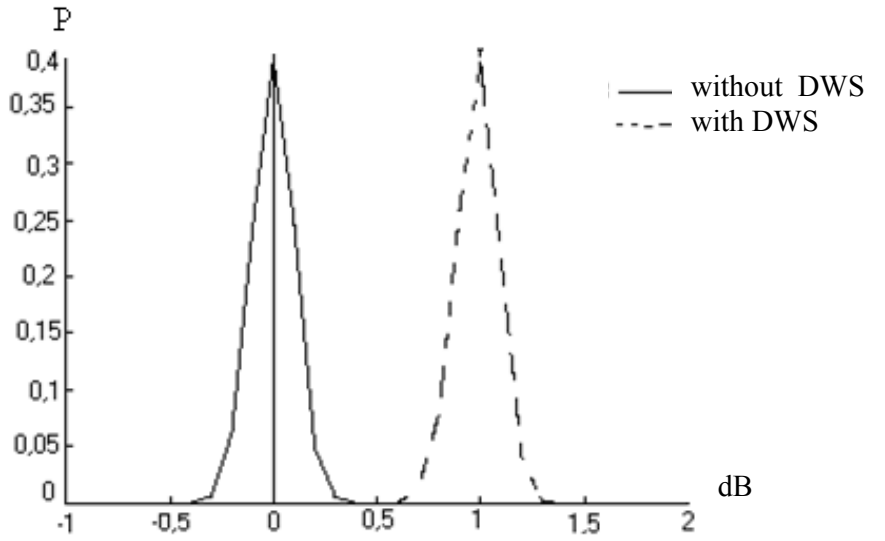

Fig. 1. Function of allocation density of determination value for signals with or without DWS

Empiric function of audio signal probability density without DWS is shown by continuous curve, dotted curve describes empiric function of audio signal probability density with embedded DWS. Both these allocations were calculated and 1000 different values of DWS at signal to noise ratio26 dB, were obtained.

Introduction of large amount of various DWS in one audio signal leads to increase of audibility distortions Maximum number of DWS is limited by the energy of each of them. Decoder is able to restore correctly each DWS on condition of usage the coder of unique keys. Fig 2 shows the example of DWS identification using different 1000 - th keys, only one being correct [1].

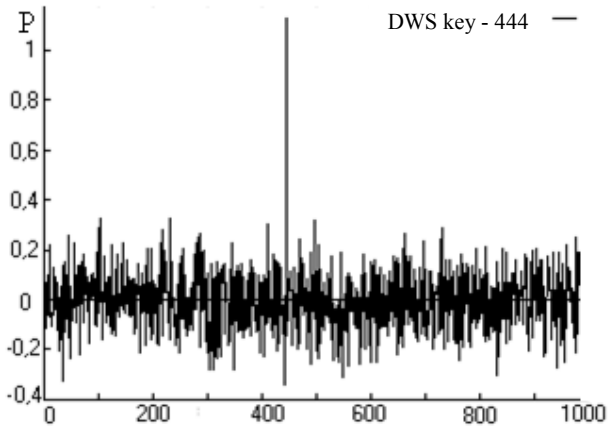

Fig. 2. Recognition of the set key of DWS embedding

Stability of DWS embedding algorithm to filtration is checked by applying of sliding filters of low and middle frequencies. Audio files with embedded DWS are filtrated by sliding filter of middle frequencies of 20 of length , that introduces significant distortions to audio information.

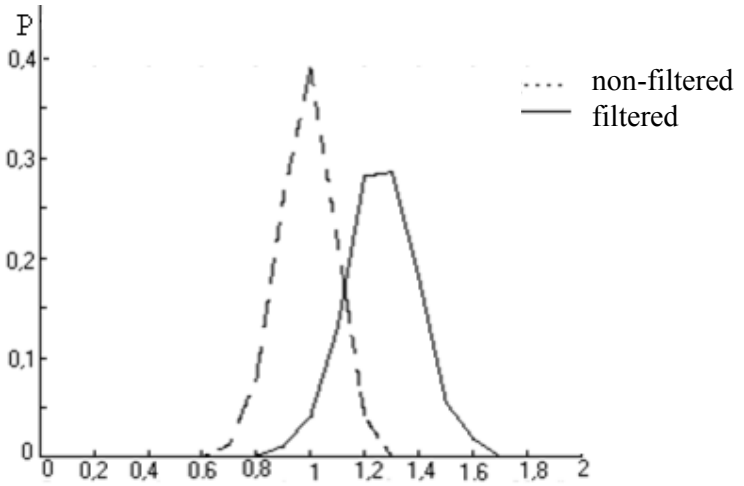

Fig. 3.The influence of mid frequencies sliding filter on DWS application to audio signals

Fig. 3 shows how threshold identification value is being changed while application of the above -described filter. Identification threshold is increased in filtered signals. It occurs, due to the fact that the function of signals allocation density after filtration is shifted to the right as compared compared with relative allocation function of non-filtered signals .

DWS is saved in case of application to audio signals of low frequency filter. However, while filtration of DWS- audio signals by Hamming low frequency filter of 25^{-th} order 2205 Hz frequency cut, the probability of DWS presence identification decreased.

While requantization of audio signal from 16-bit into 8bit one and vice versa the embedded DWS is saved , in spite of partial loss of information. Fig 4 shows the state of saved DWS in 1000 audio signals while their requantization into 8-bit counts and back into 16- bit ones.

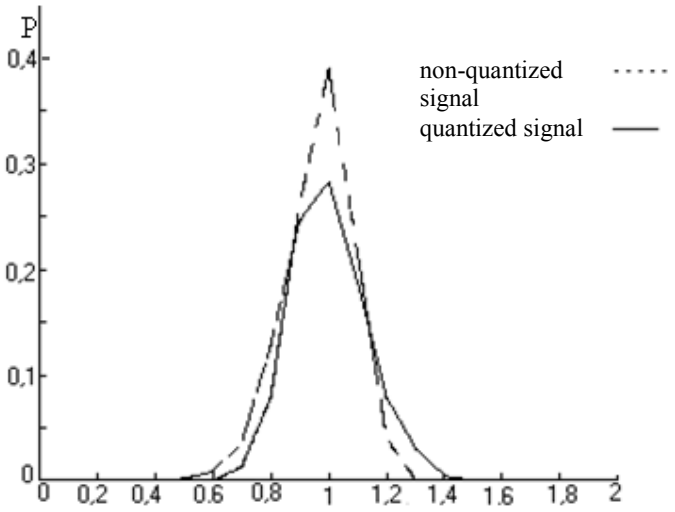

Fig. 4. Influence of signal quantization on DWS

Deviation of quantizeded signal allocation density function is increased as in the case of lower frequencies filter application and reduction of identification efficiency takes place.

Practical realization of the algorithm [4]

Let us consider programme realization of the above- mentioned algorithm and programme code fragments, described by C# language.

Bit-wise algorithm of DWS hiding is based on input file decomposition into blocks of the same size and recording of masked message bits in the last bytes of these blocks. Message recording is carried out due to increase or reduction of current byte of input file by the value, depending on the value, obtained as a result of logic addition of message byte and bit index, that corresponds to iterator value of the main cycle of the coding. The essence of logic operation is the following: in cycle *for* to the byte of message, obtained from the flow of file-message in the cycle *while,* the value of binary presentation of number 1, shifted to the left by the value, corresponding to current cycle iteration is logically added.

```
  public void Hide(Stream messageStream, Stream keyStream)
  {
  byte[] waveBuffer = new byte[bytesPerSample];
  byte message, bit, waveByte;
  int messageBuffer;
  int keyByte;

  while( (messageBuffer=messageStream.ReadByte()) >= 0 ){
  //read 1 byte from the message
  message = (byte)messageBuffer;

  //and now for each bit
  for(int bitIndex=0; bitIndex<8; bitIndex++){

  //read byte from the key
  keyByte = GetKeyValue(keyStream);

  for(int n=0; n<keyByte-1; n++){
  //copy one part
  sourceStream.Copy(waveBuffer,        0,        waveBuffer.Length,
destinationStream);
  }
  sourceStream.Read(waveBuffer, 0, waveBuffer.Length);
  waveByte = waveBuffer[bytesPerSample-1];

  //take the next bit of the message
  bit = (byte)(((message & (byte)(1 << bitIndex)) > 0) ? 1 : 0);
```

Comparing the obtained value with the value of 0, i. e., identifying the sign of the given number, we obtain the value, that will be necessary for the exchange of input file byte. Proceeding from the obtained value and the results of mod 2opeartion with current byte we perform increase or decrease of this byte.

```
  if((bit == 1) && ((waveByte % 2) == 0))
  {
     waveByte += 1;
  }
  else if((bit == 0) && ((waveByte % 2) == 1))
  {
     waveByte -= 1;
  }

  waveBuffer[bytesPerSample-1] = waveByte;
```

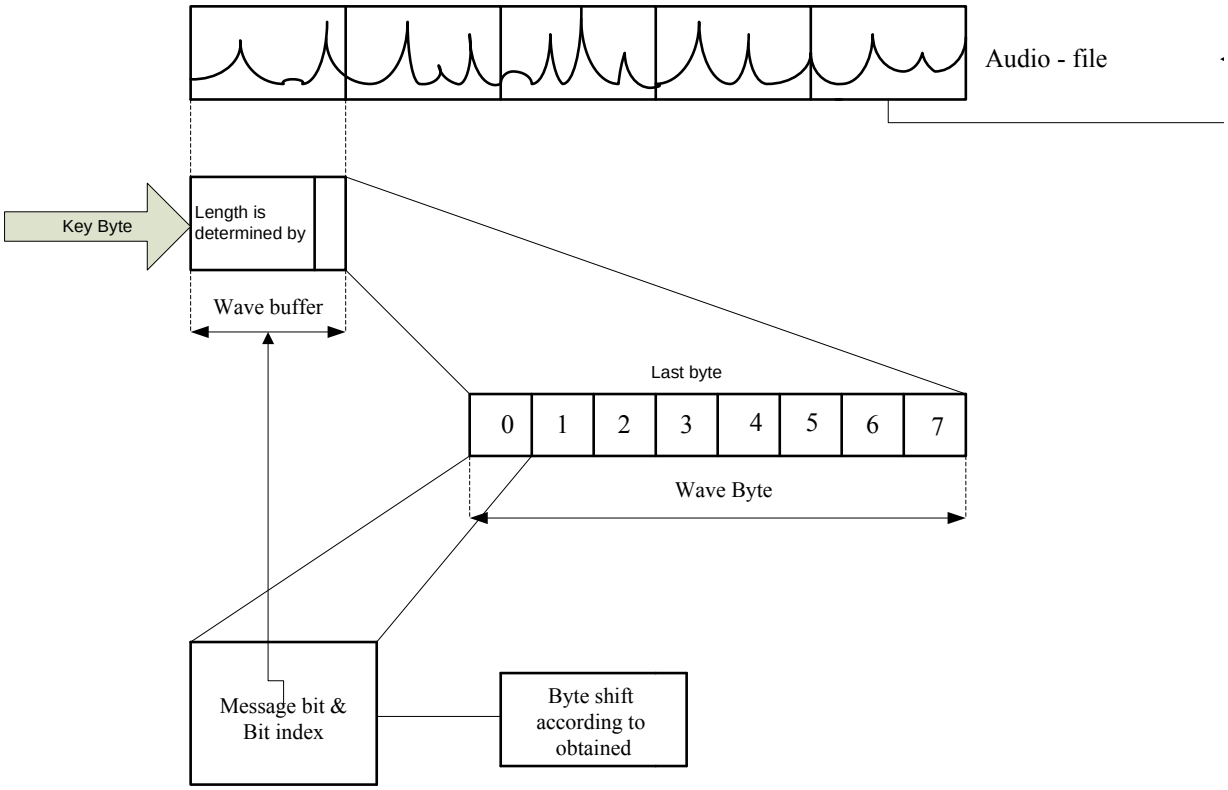

Fig. 5. Diagram of message hiding

Having message coded in this way, it is easy to carry out operation of reverse decoding, if correct key is available. Decoding algorithm is the following: input file is divided into blocks again, which are defined by the key value, the last byte is chosen for each block as a byte with hidden message. To obtain each of 8 bits, message byte consists of, mod 2 operation with current byte is performed. Obtained value is shifted by the value, corresponding to cycle iterator, that will determine digit of this bit. The obtained value is added to byte message in this cycle. After the exit from the cycle, the formed byte is introduced into message flow.

Listing of the basic cycle of hidden message reception is given below.

```
for(int bitIndex=0; bitIndex<8; bitIndex++)
{
//read key byte
keyByte = GetKeyValue(keyStream);
// Form block
for(int n=0; n<keyByte; n++)
{
sourceStream.Read(waveBuffer, 0, waveBuffer.Length);
}
//Receive the last byte of the  block
waveByte = waveBuffer[bytesPerSample-1];
//Obtain value of current bit
bit = (byte)(((waveByte % 2) == 0) ? 0 : 1);
//Write in the message byte the obtained bit in corresponding
digit
message += (byte)(bit << bitIndex);
}
```

Conclusions

The algorithm presented in the research, satisfies all the requirements, which define qualitative hiding of the DWS and minor risks of their revealing. Test program developed on the basis of algorithm, has showed the absence of target file size change, insignificant diffrence between input and output files as a result of analytic evaluation of files and visual comparison of amplitude - time characteristic graphs files.

REFERENCES

1. Барсуков В. С. Компьютерная стеганография: вчера, сегодня, завтра / В. С. Барсуков // Специальная Техника, – 2000. – № 5 .– С. 35.

2. Хорошко В. О. Основи комп'ютерної стеганографії: Навчальний посібник / В. О. Хорошко, М. Є. Шелест, О. Д. Азаров, Ю. Є. Яремчук. – Вінниця: ВДТУ, 2003 – 143 с.

3. Ткаченко О. М. Об'єктно-орієнтоване програмування мовою Java: Навчальний посібник / О. М. Ткаченко, В. А. Каплун. – Вінниця: ВНТУ, 2006. – 106 с.

4. Інформаційний ресурс наукової групи «CNews Analytics» [Електронний ресурс]. // Режим доступу: http://www/cnews.ru.

5. Cox J., Miller M., McKellips A. Watermarking as communications with side information // Proceedings of the IEEE. 1999. Vol. 87. № 7. P. 1127-41 [Електронний ресурс] // Режим доступу: http://www.autex.spb.ru/wavelet/books/stego.zip.

6. Козлюк П. В. Розробка ефективного дискретного перетворення для потокової обробки / П. В. Козлюк // Прогресивні інформаційні технології в науці та освіті. Збірник наукових праць. – Вінниця: Вінницький соціально-економічний інститут Університету «Україна». – 2007.– С. 42 – 46.

Dudatiev Andriy – Cand. Sc. (Eng.), Assistant Professor ,with the Department for Information Protection.

Kozliuk Petro – Assistant with the Department for Information Protection.

Oksymchuk Dmytro – Student.
Vinnytsia National Technical University.