ISSN 2224-087X. Електроніка та інформаційні технології. 2018. Випуск 10. С. 86–98
Electronics and information technologies. 2018. Issue 10. P. 86–98

UDC 539.681.7

DIGITAL SIGNATURE CREATIONS BY USING DISCRETE COSINE AND WAVELET TRANSFORMATIONS

I. Polovynko

Electronics and Computer technology Department,
Ivan Franko National University of Lviv,
50 Drahomanov St., UA–79005 Lviv, Ukraine
polovynkoi@gmail.com

The bringing on of digital signature (DS)–is a method of revealing unique owner sight and used first of all for identifications and protections copyrights.Depending on the method of data conversion, the technology of applying the DS can use spatial method, frequency method and spectral connections method. At present moment methods of DS constantly perfected. Int hisreport the applying of DS two algorithms are used-DCT and WLT Selection of algorithms depends from selections by users the types of DS. The DC being processed after selections, including transformations to the grayscale. For the realizations of DCT the algorithms Hsu was used . This algorithm is used in the case, where DS is more than twice smaller from the document. For the realizations of wavelets transformations, Pitas algorithm was used. This algorithm is useful in the case of DS with size more than half of image value.

Key words: digital signature, discrete cosine transformations, discrete wavelet transformations, algorithms Hsu, Pitas algorithm

Introduction

With the rapid development of high velocity computer technique (especially optical)[1-8] and Internet, the amount of transmitted information has increased significantly. It's a different type of documents, high quality multimedia and audio productions, and translations newspapers with computers and communication servers. All of these features lead to increased marketing of multimedia products on the computer market by means of e-commerce. Along with trade, technologies for protecting copyright and legal distribution of multimedia products are developing. Similarly, the need arises also in the transmission of documents, securities and newspapers.Consequently, there is a need to develop methods for depositing owner's identification marks on a multimedia document, which is a digital signature (DS) (watermark) in the form of a visible or invisible image.In general, as security criteria for digital data, data encoding, DS usage, data decoding, and DS verification are used. Key-algorithms are used for data encoding and their mixing. All methods of protection seek to satisfy the two sides of the requirement: the invisibility of the DS and its stability while modifying the image while preserving it in the desired quality.

The use of hidden securities is a method of inserting a unique copyright holder's signature into a multimedia product for identification or protection of copyright. Technology makes it possible to detect forgery and modification of data and claim copyrights. In other words, it is the protection of copyright for intellectual property, such as a multimedia product or digital

publication, which is readily available on the network by inserting a graphic sign or identifying information.

Depending on the method of data conversion, the technology of applying the DS can use spatial method, frequency method and spectral connections method. In the case of using the spatial method, the initial data in the spatial domain are analyzed and changes are made to the least significant bit for entering the digital signature data, where they will be felt to a minimum. Despite the fact that this technology makes it easy to insert a DS, it is an unstable replacement of data related to the cost of compression and filtering.In the case of using the frequency method, the initial data and the data of the DS are converted to the frequency domain, after which the modification of the coefficients is performed for the purpose of applying the DS in the initial multimedia data. In the case of using the frequency method, the discrete cosine transformations (DCT) initial data and data are converted to the frequency domain, after which a modification of the coefficients is performed in order to apply the DCT in the initial multimedia data. Since signed multimedia data often require compression and filtering, they must be resistant to such types of processing.

In the case of using the DCT, the pseudo-random number of the DS data in the initial data should be evaluated in the signature allocation process by measuring the correlation. According to this principle, the method proposed by Koksi is working. According to this method, the DS, placed perceptually important component of the signal, during operation strongly distorts the overall signal and badly affects its quality.

In fact, the modification of such a component during the removal of the DS will cause a significant distortion of the image itself. On the other hand, the perceptually important component may be slightly modified by the signature process without any loss of image quality. The applied DS is a set of data from a thousand random numbers $W = \{W_1 \dots W_{1000}\}$ that have normal (Gaussian) distribution N (0,1). A set of data is entered in the largest coefficients f_I using DCT with the following encoding function:

$$f_i^* = f_i + \alpha W_i f_i, \quad i = 1...1000 \ . \tag{1}$$

1. Discrete *cosine* transformations

Fourier series for any real and even continuous function contains only real coefficients corresponding to the cosine terms of the series [8]. Sharing this of this result on the discrete Fourier transformations (DFT) lead to discrete cosine transformations (DCT).DCT widely applied in systems of signals compression based on the orthogonal transformation method.

We can see that input analog signal $x(t)$ converted with help of analog digital converter (ADC) to digital sequence $x[n]$, which is remembered in buffer memory. Cumulated length sample Nenters the block of direct orthogonal transformation where consistency $X[k]$ is calculated. The direct orthogonal transformation must have such properties, that consistency $X[k]$ must to have a small number of significant members. Encoding only those members $X[k]$, such carry significant information.

Due to this, basically, compression of the signal is achieved. After encoding, the consistency $X[k]$ enters the communication channel or information storage device. In the decoder, reverse processes are performed.

A key point of these method realizations, is choice of orthogonal transformations and choice the way coefficient $X[k]$ encoding. The orthogonal transformations is considered as affective, when it has a fast computational algorithm, that provides to highest energy spectral concentration in a small numbers of $X[k]$ members and small distortion under their encode and decode.

The most complete, the DCT correspond toformulate requirements. Direct and vice versa DCT for the $x[n]$ consistency has the form [9]:

$$x[n]=\sum_{k=0}^{N-1}\lambda[k]X[k]\cos\left[\frac{(2n+1)k\pi}{2N}\right], \quad (2)$$

where N-long of sampling, $x[n]$; $X[k]$-coefficients DCT; $\lambda[k]$-a set of weight functions, and $\lambda[k]=\sqrt{2/N}$ when k=1,2,..N-1, $\lambda[k]=1/\sqrt{N}$ when k=0.

Set of basic functions DCT:

$$\left\{\sqrt{2/N},\ 1/\sqrt{N},\cos\left[\frac{(2n+1)k\pi}{2N}\right]\right\} \quad (3)$$

That accord to set of Chebyshev polynomials $T_N(z)=\cos(N\arccos z)$.

Show the zeroes of this polynomials: $\cos(N\arccos z)=0$;

It is: $N\arccos z=0$ or $N\arccos z=\pi/2+n\pi$ and $N\arccos z=\pi/2+n\pi$

We can normalizeChebyshev polynomials by using:

$$T_0(z)=1/\sqrt{N},T_k(z)=\sqrt{2/N}\cos(k\arccos z),k=1,2...N-1 \quad (4)$$

Substituting () in () we can calculate meaning of polynomials for k=1, 2, 3…N-1 in the points where the polynomials of N degree dive a zero. Taking this into account we can obtained:

$$T_0(z)=1/\sqrt{N},T_k(z)=\sqrt{2/N}\cos\left(k\arccos\left(\cos\frac{(2n+1)\pi}{2N}\right)\right)=\sqrt{2/N}\cos\frac{(2n+1)\pi k}{2N}. \quad (5)$$

Set of polynomials (5) are equivalent to set of basic functions of DCT. Its mean that DCT –is set of Chebyshev polynomials, calculated in the points where polynomial of N - degree dives zero.

Take into account that $X[k]$ coefficients in () are no complex. They have real values. Besides, to according the general principle of building DCT, sequence $x[n]$ is real and even. That's why DCT as it were applied to not to initial sample $x[n]$, but to transformed sample $y[n]$, that is even expansion of $x[n]$:

$y[n]=1/2\,x[n]$, when n=0,1,…N-1, and

$y[n]=1/2\,x[2N-1-n]$, when $n=N,N+1...2N-1$.

The sequence $y[n]$ must be considered as periodical with a period 2N.

The meanings of DFT $X_F[k]$ of 2*N*- points sequence $y[n]$ can be determine by formula:

$$X_F[k] = \sum_{n=0}^{N-1} y[n] e^{-j\frac{2\pi kn}{2N}} = e^{-j\frac{2\pi k}{2N}} \sum_{n=0}^{N-1} x[n] e^{-j\frac{\pi k(2N-1)}{2N}} \quad (6)$$

From (2) and (6) is follow that DCT coefficients $X_C[k]$ may be calculated from the DFT coefficient $X_F[k]$:

$$X_C[k] = \lambda[k] \mathrm{Re}\left\{ e^{-j\frac{\pi k}{2N}} X_F[k] \right\} \quad (7)$$

The existence of connection between Furies coefficients and DCT allows calculating DCT by using fast Furies transformation (FFT) algorithm.

2. Discrete wavelet transformations (DWLT)

Under DWLT the transformations of discreet meanings input signals take place, through wavelet and scaling functions. The signal is presented as sum of functions with corresponding weighed coefficients [9]:

$$f(x) = \sum_{i=1}^{n} c_i \psi(x) \quad (8)$$

Where $\psi(x)$ - basic functions, c_i - weight coefficients. In the case of wavelet functions, $\psi(x)$ must fulfilled some demands. For time of existence the function must be oscillatory. $\psi(x)$ fast increase and fast decrease in the direction of zero. The integral of function for time of its existence is equal zero: $\int_{-\infty}^{+\infty} \psi(x) dx = 0$. Permissible functions are broad-smoothed and cannot have as zero frequency as infinite frequency components. Such demands limit using basic wavelet functions insignificantly. A set of basic functions $\{\psi_i\}$ are built as scaled and shifted versions of maternal basic function $\psi_{(a,b)}$:

$$\psi_{(a,b)}(x) = a^{-1/2} \psi\left(\frac{x-a}{a} \right). \quad (9)$$

Function $\psi_{(a,b)}$ is scaled on a and shifted on $a^{-1}b$ in relation to basic wavelet basic function. In a case of DWLT take place such changes: $a = 2^{-j}$, $k = b/a$, $b = 2^{-j}k$. The scaling is realized by multiplying x on some scaling coefficient. In such case it is multiply 2. Then its turn out desirable cascade octave filters. In that case we have $\psi(2^j x)$ where *j*-is integer. Because the function ψ has accomplishing base, for coverage all signal, its necessary to shift along time axis. Such shift is written as

$$\psi_{j,k}(x) = 2^{j/2} \psi(2^j x - k) \quad (10)$$

Such expression has wide applying for calculating DWLT.Its transformations has analogy with discreet in time Fourier series relatively to discreet independent variables and vari-

ables of transformations. Its transformations is defined relatively "discreet base wavelet functions" *h (k)* and can be written as:

$$W_h f(m,n) = \frac{1}{\sqrt{a_0^m}} \sum_{k=-\infty}^{\infty} f(k) h\left(\frac{k - nb_0 a_0^m}{a_0^m}\right)$$
$$= a_0^{\frac{-m}{2}} \sum_{k=-\infty}^{\infty} f(k) h\left(a_0^{-m} k - nb_0\right) = \left\langle f, h_{m,n} \right\rangle = \left\langle f, U\left(a_0^m, nb_0 a_0^m\right) h \right\rangle \quad , \tag{11}$$

where *m*, *n*-numbers of discretization's scale steps and shifting accordingly. Scale *a* and shift *b* accordingly can write as $a = a_0^m, b = nb_0 a_0^m$., a_0, b_0 – are the step size of discreet scaling and shifting accordingly.

We will limit considerations by using most important algorithm, when coefficients are choosing on binary grate: $a = 2^j$, $b = 2^j k$ of time-scale presentations. Through this transformation take place multilevel decay of *x[n]* on j-octaves, with meaning *j*=1...*J* in the form of:

$$x[n] = \sum_{j=1}^{\infty} \sum_{k \subset Z} c_{j,k} h_j\left[n - 2^j k\right] + \sum_{k \in Z} b_{j,k} g_j\left[n - 2^j k\right] . \tag{12}$$

This equation display opposite (transverse) discrete wavelet transformation (TDLWT).the term $h_j\left[n - 2^j k\right]$ is presented synthesizing wavelet functions. For quality displaying the additional component is used. Its basic function is $g_i\left[2^{-j} k\right]$ and call synthesizing scaling function.

Under the direct DWLT the "wavelet coefficients" $c_{i,k}$ for *j*=1…*J* and scaling coefficients $b_{j,k}$ are calculated.

$$\text{DLWT } \left\{x[n]; 2^j, k2^j\right\} = c_{j,k} = \sum_n x[n] h_j^*\left[n - 2^j k\right] , \tag{13}$$

$$b_{j,k} = \sum_n x[n] g_j^*\left[n - 2^j k\right], \tag{14}$$

where $h_j^*\left[n - 2^j k\right]$ - analyzing discrete wavelet function, and $g_j^*\left[n - 2^j k\right]$- analyzing scaling function. TDLWT executes displaying of original signal from its coefficients by using expression (13).

Let us tike in consideration the analysis of functions. We take into account impulse response of two filters: for low frequency *g*[*n*] and high frequency *h*[*n*] .The wavelet and scaling function we can write as:

$$g_1[n] = g[n],\ h_1[n] = h[n],\ g_{j+1} = \sum_k g_j[k] g[n - 2k],\ h_{j+1}[n] = \sum_k h_j[k] h[n - 2k] \tag{15}$$

i.e. take please the transitions from one octave j to the next (*j*+1) by using interpolation operator:

$$f[n] \to \sum_k f[k] g[n - 2k] . \tag{16}$$

Its predict as discreet equivalent of expansion $f(t) \to 2^{-1/2} f\left(t/2\right)$.

Take into account () and () for the same level of decomposition we can write:

$$x[n] = \sum_{k \in Z} c_{j+1,k} h_{j+1}\left[n - 2^{j+1} k\right] + \sum_{k \in Z} b_{j+1,k} hg\left[n - 2^{j+1} k\right] \tag{17}$$

Take into account (13) and (14) in view of basic functions orthonormality we get:

$$c_{j+1,k} = \sum_k c_{j,k} h_j^* \left[n - 2^j k \right], \tag{18}$$

and

$$b_{j+1,k} = \sum_k b_{j,k} g_j^* \left[n - 2^j k \right]. \tag{19}$$

The expressions (18), (19) are used for fast algorithm of DWLT decomposition.

3. Realization

Practical part of this work was realized by using C#program language in the environment Visual Studio 2017 Community Edition. For working with images, the basic methods of NET-technology where used. Additionally, the library AForge.NET was used too.

The applying of DS two algorithms are used. Selection of algorithms depends from selections by users the types of DS. The DC being processed after selections, including transformations to the grayscale.

For the realizations of DCT the algorithms Hsu was used [11]. This algorithm is used in the case, where DS is more than twice smaller from the document. In this case for DS decoder the initial image is needed. Decoder doesn't reveal existence of DS, but select input data. As DS appear black-white image, with double lover content. Before inputting, on the image influence random. The DS is inputting in middle –frequency coefficients DCT (quarter part from general amount). These coefficients are situated along second diagonal of DCT matrix.

For inputting one bit of DS into coefficient, the sign of difference between coefficients of current block and corresponded to it from previous block. When is need to input 1, the coefficient change such, that sight of difference become positive, when 0 –the sight must be negative.

There are many ways to improve of base algorithm. Firstly instead of coefficient meaning it is prudent to use their absolute values. Secondly, instead coefficients from previous blocs can be used coefficients of current blocs. Another more improvements of this algorithm is the sort order, in which DS blocs are arranged by falling of numbers of units in themes. The blocs of output images are arranged by falling dispersions. After that, appropriative the data attachment is executed.

For the realizations of wavelets transformations, Pitas algorithm [12] was used. This algorithm is useful in the case of DS with size more than half of image value. There are exist numerous versions of the Pitas algorithm at the beginning inputting of bit in every image pixel was proposed. But later the blocs of 2x2 or 3x3 pixels were used. It's doing the algorithm more robust under compression or filtrations. In the next step, DS is attached with image. If digital image detector for built-in blocs is used, the average meaning of brightness is calculated. It's give the possibility of uneven inserting DS into pixels. So we can get digital image, that is optimized according to robust criteria and undergo to compresses procedure by using JPEG algorithm. In finish part, the8x8 element blocs in advance the "capacity" of each pixel. Such mask is suitable for all pictures.

4. Conclusions

Given in this work analyst show, that for effective bringing on and reading of DS it is necessary to using simultaneously no less two different methods. The more effectively reviled of using DCT and DWLT methods. These methods at present time are well worked out. Especially for DCT algorithms Hsu and for DWLT -Pitas algorithm are used. The program work occur in two schedules: applying and reading of image. The program is able to open any image format and safe encrypted image in any format too.

Supplement

Program code:

```
usingSystem;
usingSystem.Drawing;
usingSystem.Windows.Forms;
usingAForge.Imaging.Filters;
usingSystem.Collections;
usingSystem.Linq;

namespaceDigital_Watermark
{
publicpartialclass Form1 : Form
      {
Colordefault_color = Color.FromArgb(64, 64, 64);
Colorselected_color = Color.FromArgb(94, 94, 94);

public Form1()
{
            InitializeComponent();
            ApplySwitch.BackColor = selected_color;
}

privatevoidApplySwitch_Click(object sender, EventArgs e)
{
            ApplySwitch.BackColor = selected_color;
            ReadSwitch.BackColor = default_color;
            GoButton.Text = "Apply";
            WatermarkBox.Visible = true;
            Clear ();
}

privatevoidReadSwitch_Click(object sender, EventArgs e)
{
            ReadSwitch.BackColor = selected_color;
            ApplySwitch.BackColor = default_color;
            GoButton.Text = "Read";
            WatermarkBox.Visible = false;
            Clear ();
}

privatevoidClear()
{
```

```
                ImageBox.Image = null;
                WatermarkBox.Image = null;
                ResultBox.Image = null;
        }

        privatevoidExitButton_Click(object sender, EventArgs e)
        {
                Application.Exit();
        }

        privatevoidImageBox_Click(object sender, EventArgs e)
        {
                OpenFileDialogofd = newOpenFileDialog();
                ofd.Filter = "PngImage (.png)|*.png| BitmapImage (.bmp)|*.bmp| GifImage
(.gif)|*.gif| JPG
                Image (.jpg) |*.jpg";

                if (ofd.ShowDialog() == DialogResult.OK)
                {
                        ImageBox.ImageLocation = ofd.FileName;
                        ImageBox.SizeMode = PictureBoxSizeMode.StretchImage;
                }
        }

        privateBitmapToGreyScale(Bitmapbitmap)
        {
                intgrey, i, j;
                Colorcolor;
                Progress.Maximum = bitmap.Width * bitmap.Height;
                Progress.Step = 1;
                for (i = 0; i <bitmap.Width; i++)
                {
                        for (j = 0; j <bitmap.Height; j++)
                        {
                        color = bitmap.GetPixel(i, j);
                        grey = (int)((color.R + color.G + color.B) / 3);
                        bitmap.SetPixel(i, j, Color.FromArgb(grey, grey, grey));
                        Progress.PerformStep();
                        }
                }
                Progress.Value = 0;
                returnbitmap;
        }

        privatevoidWatermarkBox_Click(object sender, EventArgs e)
        {
                OpenFileDialogofd = newOpenFileDialog();
                ofd.Filter = "PngImage (.png)|*.png| BitmapImage (.bmp)|*.bmp| GifImage
(.gif)|*.gif | JPG
                Image (.jpg)|*.jpg";
                if (ofd.ShowDialog() == DialogResult.OK)
```

```
                {
                        Bitmapimg = newBitmap(ofd.FileName);
                        WatermarkBox.Image = ToGreyScale(img);
                        WatermarkBox.SizeMode = PictureBoxSizeMode.StretchImage;
                }
        }

        privatevoidResultBox_Click(object sender, EventArgs e)
        {
                if (ResultBox.Image == null) return;
                SaveFileDialogsfd = newSaveFileDialog();
                sfd.Filter = "BitmapImage (.bmp)|*.bmp";
                if (sfd.ShowDialog() == DialogResult.OK) ResultBox.Image.Save(sfd.FileName);
        }

        privatebytegetByte(byte[] bits)
        {
                StringbitString = "";
                for (int i = 0; i < 8; i++)
                bitString += bits[i];
                bytenewpix = Convert.ToByte(bitString, 2);
                intdePix = (int)newpix ^ 2;
                return (byte)dePix;
        }

        privatebyte[] getBits(bytesimplepixel)
        {
                intpixel = 0;
                pixel = (int)simplepixel ^ 2;
                BitArraybits = newBitArray(newbyte[] { (byte)pixel });
                bool[] boolarray = newbool[bits.Count];
                bits.CopyTo(boolarray, 0);
                byte[] bitsArray = boolarray.Select(bit => (byte)(bit ? 1 : 0)).ToArray();
                Array.Reverse(bitsArray);
                returnbitsArray;
        }

        privatevoidGoButton_Click(object sender, EventArgs e)
        {
                if (GoButton.Text.Equals("Aplly") && !(ImageBox.Image == null || Watermark-
Box.Image == null))
                {
                        Bitmapsimple = newBitmap(ImageBox.Image);
                        BitmapsecretGreyScale = newBitmap(WatermarkBox.Image);
                if (secretGreyScale.Height != simple.Height || secretGreyScale.Width != sim-
ple.Width)
                {
                ResizeBilinearresizeFilter = newResizeBilinear(simple.Width, simple.Height);
                secretGreyScale = resizeFilter.Apply(secretGreyScale);
                }
                ColorpixelContainerImage = newColor();
```

```
ColorpixelMsgImage = newColor();

byte[] MsgBits;
byte[] AlphaBits;
byte[] RedBits;
byte[] GreenBits;
byte[] BlueBits;
bytenewAlpha = 0;
bytenewRed = 0;
bytenewGreen = 0;
bytenewBlue = 0;
#regionEncryption
Progress.Maximum = simple.Height * simple.Width;
Progress.Step = 1;
for (int i = 0; i <simple.Height; i++)
{
for (int j = 0; j <simple.Width; j++)
{
pixelMsgImage = secretGreyScale.GetPixel(j, i);
MsgBits = getBits((byte)pixelMsgImage.R);
pixelContainerImage = simple.GetPixel(j, i);
AlphaBits = getBits((byte)pixelContainerImage.A);
RedBits = getBits((byte)pixelContainerImage.R);
GreenBits = getBits((byte)pixelContainerImage.G);
BlueBits = getBits((byte)pixelContainerImage.B);
AlphaBits[6] = MsgBits[0];
AlphaBits[7] = MsgBits[1];
RedBits[6] = MsgBits[2];
RedBits[7] = MsgBits[3];
GreenBits[6] = MsgBits[4];
GreenBits[7] = MsgBits[5];
BlueBits[6] = MsgBits[6];
BlueBits[7] = MsgBits[7];
newAlpha = getByte(AlphaBits);
newRed = getByte(RedBits);
newGreen = getByte(GreenBits);
newBlue = getByte(BlueBits);
pixelContainerImage = Color.FromArgb(newAlpha, newRed, newGreen, newBlue);
simple.SetPixel(j, i, pixelContainerImage);
Progress.PerformStep();
}
}
ResultBox.Image = simple;
ResultBox.SizeMode = PictureBoxSizeMode.StretchImage;
Progress.Value = 0;
#endregion
}
elseif (GoButton.Text.Equals("Read") &&ImageBox.Image != null)
{
        BitmapEncryptedImage = (Bitmap)ImageBox.Image;
```

```
                    BitmaphiddenImage = newBitmap(EncryptedImage.Width, EncryptedI-
mage.Height);
                    ColorpixelToDecrypt = newColor();

                    byte[] BitsToDecrypt = newbyte[8];
                    byte[] AlphaBits;
                    byte[] RedBits;
                    byte[] GreenBits;
                    byte[] BlueBits;
                    bytenewGrey = 0;
                    #regionDecryption
                    Progress.Maximum = EncryptedImage.Height * EncryptedImage.Width;
                    Progress.Step = 1;
                    for (int i = 0; i <EncryptedImage.Height; i++)
                    {
                    for (int j = 0; j <EncryptedImage.Width; j++)
                    {
                    pixelToDecrypt = EncryptedImage.GetPixel(j, i);
                    AlphaBits = getBits((byte)pixelToDecrypt.A);
                    RedBits = getBits((byte)pixelToDecrypt.R);
                    GreenBits = getBits((byte)pixelToDecrypt.G);
                    BlueBits = getBits((byte)pixelToDecrypt.B);
                    BitsToDecrypt[0] = AlphaBits[6];
                    BitsToDecrypt[1] = AlphaBits[7];
                    BitsToDecrypt[2] = RedBits[6];
                    BitsToDecrypt[3] = RedBits[7];
                    BitsToDecrypt[4] = GreenBits[6];
                    BitsToDecrypt[5] = GreenBits[7];
                    BitsToDecrypt[6] = BlueBits[6];
                    BitsToDecrypt[7] = BlueBits[7];
                    newGrey = getByte(BitsToDecrypt);
                    pixelToDecrypt = Color.FromArgb(newGrey, newGrey, newGrey);
                    hiddenImage.SetPixel(j, i, pixelToDecrypt);
                    Progress.PerformStep();
                    }
                    }
                    ResultBox.Image = hiddenImage;
                    ResultBox.SizeMode = PictureBoxSizeMode.StretchImage;
                    Progress.Value = 0;
                    #endregion
                    }
                }
            }
        }
```

REFERENCES

1. *Половинко І.* Фізика оптичних комп'ютерів / І. Половинко, Т. Кріль // Електроніка та інформаційні технології. – 2014. – Вип. 4. – С. 3-23.
2. *Polovynko I.* The Optico-Geometrical Analysis of the Planar Waveguides and Modulators // Proceedings of the Xth International Scientific and Practical Conference "Electronics

and Information Tecnolologies", August 30-September 2 2018, Lviv-Karpaty village, Ukraine. – P. A-151 – A-154
3. *Кріль Т.* Розрахунок параметрів оптичних планарних хвилеводів / Т. Кріль, І. Половинко, С. В. Рихлюк // Тези V наук.-практ. конф. ЕлІТ–2013 "Електроніка та інформаційні технології". – Львів-Чинадієво-Мукачево (Закарпатська обл., Україна). – 29 серпня–1 вересня 2012. – С. 121–123.
4. *Polovynko I.* To the possibility of the building integrating – optical computer / I. Polovynko, T. Kril //6-th Ukrainian-Polish Scientific Conference «Electronics and information technology» Lviv-Chynadiyevo, Ukraine 28-31 august 2014.– Book of abstracts. – C. 190-192.
5. *Половинко І.* Матеріали для елементів оптичних комп'ютерів / І. Половинко, Т. Кріль // "Relaxed, nonlinear and acoustic optical processes and materials" RNAOPM 14, Ukraine, Lutsk – Lake "Svitiaz", Int. Conf. June 8-12, 2014. – P.120.
6. *Polovynko T.* Calculation of the optical wave propagations in the solid state waveguides / T. Polovynko, S. Rykhlyuk, I. Polovynko // Book of abstracts of XX-th International Seminar on Physics and Chemistry of Solids ISPCS'15. – Lviv (Ukraine). – 13–16 September 2015. – P.85 .
7. *Polovynko T. I.* Optico-geometrical analysis of the GaAs wawegides / Polovynko T. I., Grynchuk G. Ya., Polovynko I. I. // FIRST INTERNATIONAL WORKSHOP "Actual problems of fundamental science" – APFS'2015 (oral session), dedicated to 75 Anniversary of prof. Olekseyuk I.D., May 30 – June 3, 2015, Lutsk–Lake "Svityaz'", UKRAINE. – P.79-82.
8. *Половинко Т. І.* Оптико-геометричний аналіз діелектричних хвилеводів / Половинко Т. І., Гринчук Г. Я., Половинко І. І. // Електроніка та інформаційні технології. – 2015. – Вип. 5. – С.126-131.
9. *Бондарев В.* Цифровая обработка сигналов: методы и средства: Учебное пособие для вузов / В. Бондарев, Г. Трэстер, В. Чернега – СЕВГТУ, 1999. – 398 с.
10. *Наконечний А. Й.* Цифрова обробка сигналів: навч. посібник / А. Й. Наконечний, Р. А. Наконечний, В. А. Павліш. – Львів: Видавництво Львівської політехніки, 2010. – 368 с.
11. Chloe Ching Yun Hsu,CrisUlams. A new algorithm for fast generalization DFTs. Computer Science, Data structure and Algorythms.Cornell University. 2017, v.1. P.121-124.
12. *Pitas I.* Digital Image Processing Algorithms And Applications. 2000, Canada. – 188 p.

НАНЕСЕННЯ ЦИФРОВИХ ПІДПИСІВ ІЗ ВИКОРИСТАННЯМ КОСИНУСНОГО ТА МАЛОХВИЛЬОВОГО ДИСКРЕТНИХ ПЕРЕТВОРЕНЬ

І. Половинко

Факультет електроніки та комп'ютерних технологій,
Львівський національний університет імені Івана Франка,
вул. Драгоманова 50, 79005 Львів, Україна
polovynkoi@gmail.com

У зв’язку із істотним зростанням кількості передачі електронних документів, виникає потреба в розробленні ефективних методів нанесення на них ідентифікаційних міток власника, які представляють собою цифровий підпис (ЦП) у формі видимого або невидимого зображення. ЦП повинні відповідати певним вимогам: Зокрема вони повинні бути непомітними для користувачів. Автор повинен мати можливість виявити несанкціоноване використання файлу. ЦП не може бути видалений сторонніми особами а також бути стійким до зміни формату і розмірів документу, його масштабування, стискування, повороту, фільтрації, введення спецефектів, монтажу, аналогових і цифрових перетворень.

ЦП хоча і відрізняються середньою надійністю, але при цьому має незначну інформаційну ємність. Представлена робота покликана підвищити надійність ЦП. Для цього нанесення ЦП здійснювалось двома алгоритмами, причому вибір алгоритму залежить від співвідношення розмірів самого ЦП та зображення. ЦП проходить обробку після вибору, включаючи перетворення в градації сірого. Для реалізації дискретного косинусного перетворення використовується алгоритм Хсю. Цей алгоритм є ефективним у випадку вибору розмірів ЦП вдвічі меншим розміру зображення.

Для реалізації малохвильового перетворення використовується алгоритм Пітаса. Цей алгоритм використовується у випадку вибору ЦП більше половини розміру зображення.

Практична частина роботи реалізована мовою програмування C# у середовищі VisualStudio 2017 CommunityEdition. Для обробки зображення використовуються базові методи технології .NET та додатково бібліотека обробки зображення AForge.NET.

Робота програми відбувається в двох режимах: нанесення і зчитування зображення. Програма, код якої представлений у додатку, дозволяє відкривати будь-який формат зображення і, відповідно, зберігати зашифроване зображення теж в будь-якому форматі.

Ключові слова: цифровий підпис, дискретне косинусне перетворення, малохвильове перетворення, алгоритм Хсю, алгоритм Пітаса.

Стаття: надійшла до редакції 07.11.2018,
доопрацьована 14.11.2018,
прийнята до друку 15.11.2018.