J. Basic. Appl. Sci. Res., 3(6)1193-1201, 2013

© 2013, TextRoad Publication

ISSN 2090-4304
Journal of Basic and Applied
Scientific Research
www.textroad.com

A New Secure Image Encryption Algorithm using Logical and Visual Cryptography Algorithms and based on Symmetric Key Encryption

Mohammad Soltani

Young Researchers Society, Department of Computer Engineering, Shahid Bahonar University, Kerman, Iran

Received: May 13 2013
Accepted: June 9 2013

ABSTRACT

Image applications have been increasing in recent years. Encryption is used to provide the security needed for image applications. There are many image encryption schemes have been proposed, each one of them has its strength and weakness. In this paper I suggested a new, secure and robust cryptography algorithm to prevent unauthorized access to contents of encrypted images. The main features of cryptography algorithm defined in this article are logical cryptography and visual cryptography together, the ability to encrypt the secret image in successive stages, changing the physical structure of the secret image, no limitation for the number of keys, using logical operation, creating six keys, Interdependence of all keys in all stages of encrypting and decrypting, bigger changes in the physical structure of the encrypted image In case of wrong decryption and to make the resulting keys and encrypted image unique after the cryptography process.

KEYWORDS: Security, Symmetric key, Secret Images, Encryption algorithm, Keys interdependent, Logical operation, Random numbers

1. INTRODUCTION

Cryptography in all of its applications, including data confidentiality, data integrity, and user authentication, is the most powerful tool for protecting information [1]. The field of encryption is becoming very important in the present era in which information security is of utmost concern. Security is an important issue in communication and storage of images, and encryption is one of the ways to ensure security [2]. Cryptography has a specific role to protect secret communication from unauthorized access and to prevent such attacks encryption technique is the best way [3, 4]. Since the celebrated Shannon's work, cryptography has become one of the fields of modern science to protect secret communication [5]. Due to the importance of cryptography in protecting secret communication, security of information has become a major issue during the last decades [5]. In other words cryptography was invented to protect communications, and the issue of trust was not addressed explicitly [6]. The encryption algorithm required keys [7] so In general, the algorithms used for cryptography applications are classified into two types, Asymmetric methods or public key cryptography and Symmetric methods or Symmetric key cryptography [5]. The goal of the study: In this paper I suggested a new robust cryptography algorithm based on symmetric keys to increase security and prevent from unauthorized access to the contents of encrypted images. This cryptography algorithm can lead to further image theft Prevention and debarment from detecting contents of the secret image. the major contribution of this paper and a comparison with another published works: logical cryptography and visual cryptography together, ability to encrypt the secret image in successive stages, changing the physical structure of the secret image, no limitation for the number of keys, Creating six keys, Interdependence of all keys in all stages of encrypting and decrypting, bigger changes in the physical structure of the encrypted image In case of wrong decryption and to make the resulting keys and encrypted image unique after the cryptography process. The rest of the paper is organized as follows. Section 2 discusses the type of selectable image for cryptography, the creation method of keys and resultantly the secret image cryptography using each of them. Section 3 discusses the algorithm to decrypt the image encrypted by each key. Section 4 discusses cryptography process. Section 5 discusses conclusion.

2. MATERIAL AND METHOD

According to the second principle of Auguste Kerckhoffs, cryptography algorithm must not include any secret and hidden point. In fact the only secret point is the secret key [8, 9]. The Cryptography algorithm defined in this paper aims at boosting the security of the secret image cryptography style based on symmetric keys.

2.1 The type of selectable image for cryptography

With regard the structure of images, due to the fact that a physical file is a group of bytes gathered physically in a disk [10], the cryptography algorithm defined in this article can be applied for the cryptography of all files with the same physical structure.

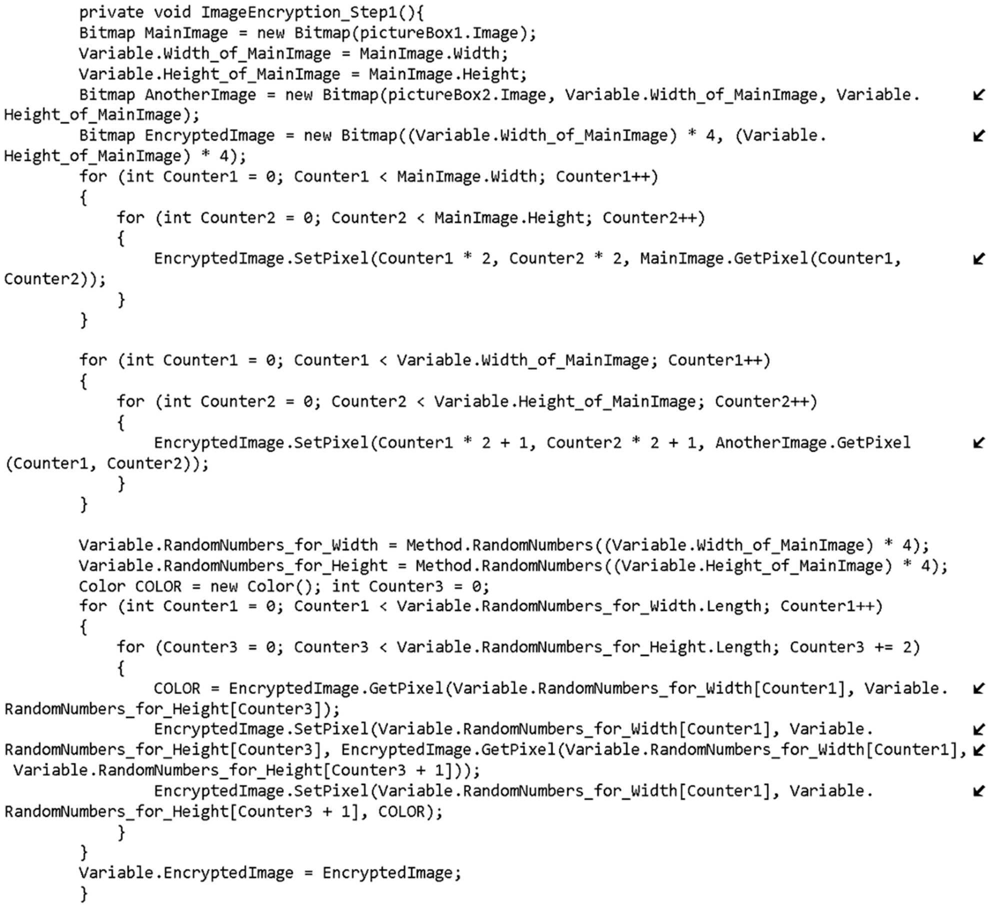
***Corresponding Author:** Mohammad Soltani, Young Researchers Society, Department of Computer Engineering, Shahid Bahonar University, Kerman, Iran. E-Mail: soltani.mohammad.edu@gmail.com Tel: 0098-913-2981497

2.2 Image encryption steps and the creation method of keys and resultantly the secret image cryptography using each of them

According to the second principle of Auguste Kerckhoffs, to stop decoding the content of the encrypted image through hacking the keys, there is no limitation for the number of keys to construct the cryptography algorithm defined in the preset article. In addition, the structures of all keys are interdependent while encryption and decryption. In case of lacking even a single key in the decrypting stage, bigger changes in the physical structure of the encrypted image are possible. The physical structure of the created keys for encryption and decryption are of 6 types.

2.2.1 The first step of the main image encryption using another image is done according to the fallowing source code. Another image is used to combine with the main image. It's done to obfuscate the main image

The first step of the main image encryption using another image:



```
        private void ImageEncryption_Step1(){
        Bitmap MainImage = new Bitmap(pictureBox1.Image);
        Variable.Width_of_MainImage = MainImage.Width;
        Variable.Height_of_MainImage = MainImage.Height;
        Bitmap AnotherImage = new Bitmap(pictureBox2.Image, Variable.Width_of_MainImage, Variable.     ↙
Height_of_MainImage);
        Bitmap EncryptedImage = new Bitmap((Variable.Width_of_MainImage) * 4, (Variable.              ↙
Height_of_MainImage) * 4);
        for (int Counter1 = 0; Counter1 < MainImage.Width; Counter1++)
        {
            for (int Counter2 = 0; Counter2 < MainImage.Height; Counter2++)
            {
                EncryptedImage.SetPixel(Counter1 * 2, Counter2 * 2, MainImage.GetPixel(Counter1,       ↙
Counter2));
            }
        }

        for (int Counter1 = 0; Counter1 < Variable.Width_of_MainImage; Counter1++)
        {
            for (int Counter2 = 0; Counter2 < Variable.Height_of_MainImage; Counter2++)
            {
                EncryptedImage.SetPixel(Counter1 * 2 + 1, Counter2 * 2 + 1, AnotherImage.GetPixel       ↙
(Counter1, Counter2));
            }
        }

        Variable.RandomNumbers_for_Width = Method.RandomNumbers((Variable.Width_of_MainImage) * 4);
        Variable.RandomNumbers_for_Height = Method.RandomNumbers((Variable.Height_of_MainImage) * 4);
        Color COLOR = new Color(); int Counter3 = 0;
        for (int Counter1 = 0; Counter1 < Variable.RandomNumbers_for_Width.Length; Counter1++)
        {
            for (Counter3 = 0; Counter3 < Variable.RandomNumbers_for_Height.Length; Counter3 += 2)
            {
                COLOR = EncryptedImage.GetPixel(Variable.RandomNumbers_for_Width[Counter1], Variable.  ↙
RandomNumbers_for_Height[Counter3]);
                EncryptedImage.SetPixel(Variable.RandomNumbers_for_Width[Counter1], Variable.          ↙
RandomNumbers_for_Height[Counter3], EncryptedImage.GetPixel(Variable.RandomNumbers_for_Width[Counter1], ↙
 Variable.RandomNumbers_for_Height[Counter3 + 1]));
                EncryptedImage.SetPixel(Variable.RandomNumbers_for_Width[Counter1], Variable.          ↙
RandomNumbers_for_Height[Counter3 + 1], COLOR);
            }
        }
        Variable.EncryptedImage = EncryptedImage;
        }
```

Figure 1. The first step of the main image encryption

The features of this source code are defined based on the following points:

1. Source code language is C Sharp (C#).
2. In the source code using Dot Net framework (.NET), Object Oriented Programming (OOP), Random numbers and Bitmap Object.

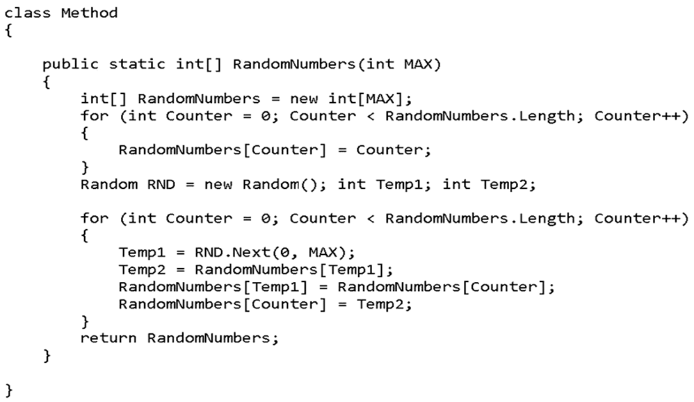
- An introduction to the .NET Framework: The .NET framework was designed to make life simpler for programmers by providing a common platform that can be used when writing programs in several different programming languages [11].
- An introduction to the object oriented programming: Object oriented programming is a method of programming that involves the creation of intellectuals objects [11],[12].
- We use two arrays to combine the two images. The Length of the first array(RandomNumbers_for_Height) is equal to the height of the main image and the length of the second array (RandomNumbers_for_Width) is equal to the Width of the main image. The elements of these arrays are random and non-repeating numbers.
- An introduction to the Bitmap Object: A bitmap is an array of bits that specify the color of each pixel in a rectangular array of pixels. The number of bits devoted to an individual pixel determines the number of colors that can be assigned to that pixel. For example, if each pixel is represented by 4 bits, then a given pixel can be assigned one of 16 different colors (2^4 = 16). The following table shows a few examples of the number of colors that can be assigned to a pixel represented by a given number of bits[13, 14]. A Bitmap object created from the stream displayed inside a picture box [12].

Table 1. Examples of the number of colors that can be assigned to a pixel represented by a given number of bits

Bits per pixel	Number of colors that can be assigned to a pixel
1	2^1 = 2
2	2^2 = 4
4	2^4 = 16
8	2^8 = 256
16	2^16 = 65,536
24	2^24 = 16,777,216

3. In the source code using Method.cs Class and Variable.cs Class

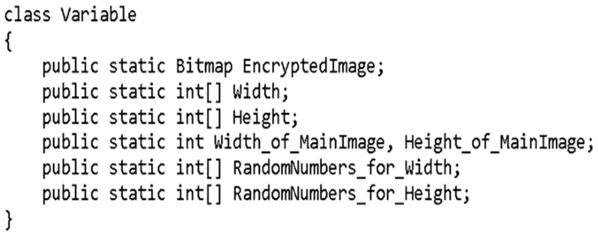
- An introduction to the Method.cs Class: According to the following source code In the Method.cs class random numbers is created using static RandomNumbers method.



```
class Method
{

    public static int[] RandomNumbers(int MAX)
    {
        int[] RandomNumbers = new int[MAX];
        for (int Counter = 0; Counter < RandomNumbers.Length; Counter++)
        {
            RandomNumbers[Counter] = Counter;
        }
        Random RND = new Random(); int Temp1; int Temp2;

        for (int Counter = 0; Counter < RandomNumbers.Length; Counter++)
        {
            Temp1 = RND.Next(0, MAX);
            Temp2 = RandomNumbers[Temp1];
            RandomNumbers[Temp1] = RandomNumbers[Counter];
            RandomNumbers[Counter] = Temp2;
        }
        return RandomNumbers;
    }

}
```

Figure 2. Method.cs Class

- An introduction to the Variable.cs Class: According to the following source code In the Variable.cs class, static data is created.



```
class Variable
{
    public static Bitmap EncryptedImage;
    public static int[] Width;
    public static int[] Height;
    public static int Width_of_MainImage, Height_of_MainImage;
    public static int[] RandomNumbers_for_Width;
    public static int[] RandomNumbers_for_Height;
}
```

Figure 3. Variable.cs Class

The table indicating the symmetric keys in the first step of image encryption:

According to The first step of the main image encryption, the following table indicating the symmetric keys in the first step of the main image encryption:

Table 2. The table indicating the symmetric keys in the first step of image encryption

Keys number	Keys Name
1	Width_of_MainImage
2	Height_of_MainImage
3	RandomNumbers_for_Width
4	RandomNumbers_for_Height

Sample of The first step of the main image encryption using another image:

Sample of the main image encryption using another image is done according to the fallowing figure

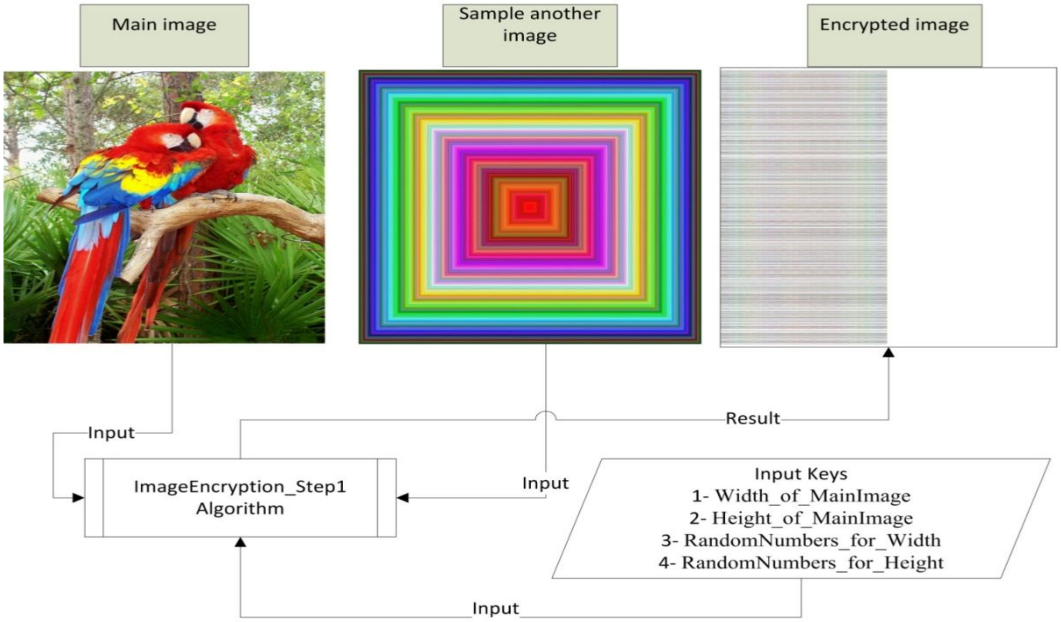


Figure 4. Sample of The first step of the main image encryption using another image

2.2.2 The second step of the main image encryption using the key determining the result of the XNOR logical operation defined based on the following Flowchart

The second step of the main image encryption using the key determining the result of the XNOR logical operation defined based on the following Flowchart [15]:

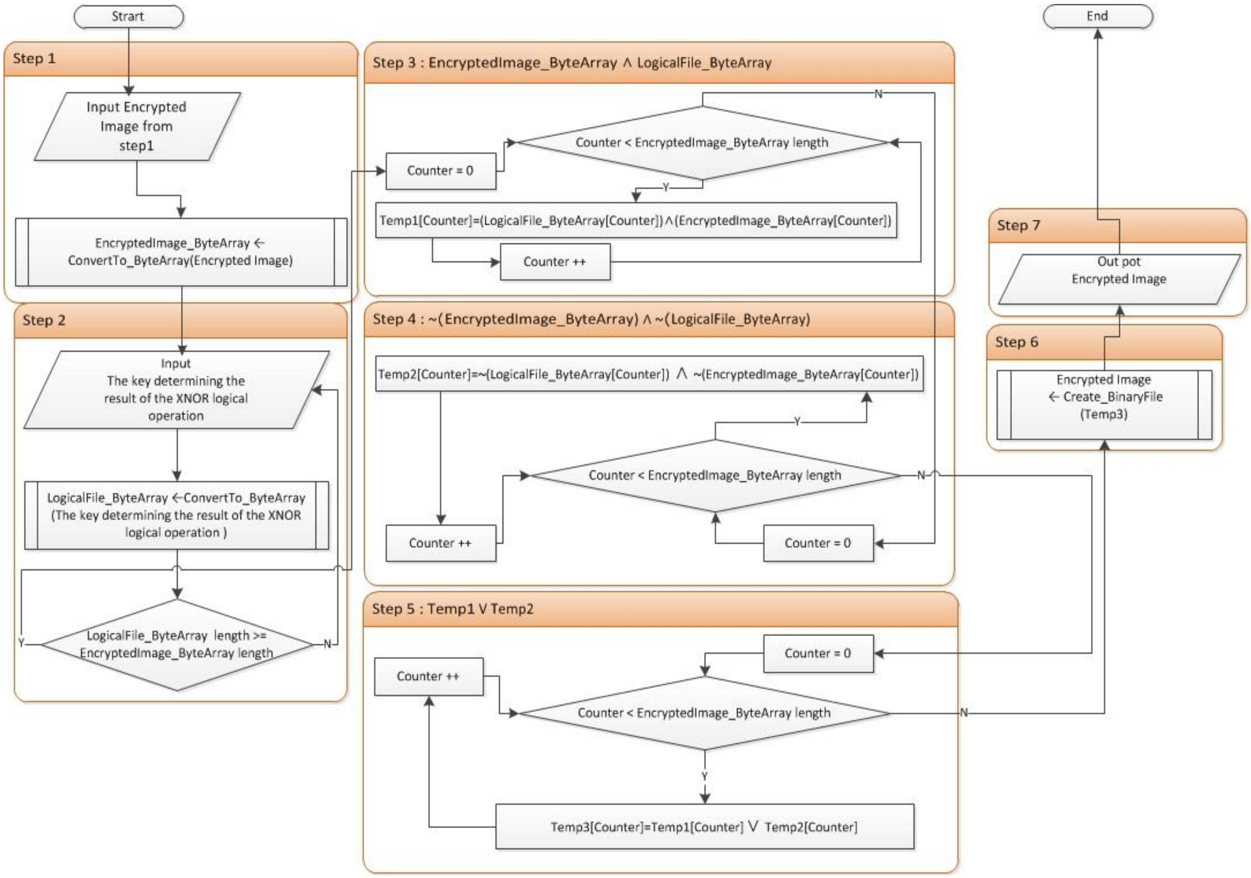

Strart
Step 1
Input Encrypted Image from step1
EncryptedImage_ByteArray ← ConvertTo_ByteArray(Encrypted Image)
Step 2
Input The key determining the result of the XNOR logical operation
LogicalFile_ByteArray ←ConvertTo_ByteArray (The key determining the result of the XNOR logical operation)
LogicalFile_ByteArray length >= EncryptedImage_ByteArray length
Y
N
Step 3 : EncryptedImage_ByteArray ∧ LogicalFile_ByteArray
Counter = 0
Counter < EncryptedImage_ByteArray length
Y
N
Temp1[Counter]=(LogicalFile_ByteArray[Counter])∧(EncryptedImage_ByteArray[Counter])
Counter ++
Step 4 : ~(EncryptedImage_ByteArray) ∧ ~(LogicalFile_ByteArray)
Temp2[Counter]=~(LogicalFile_ByteArray[Counter]) ∧ ~(EncryptedImage_ByteArray[Counter])
Y
Counter < EncryptedImage_ByteArray length
N
Counter ++
Counter = 0
Step 5 : Temp1 V Temp2
Counter = 0
Counter ++
Counter < EncryptedImage_ByteArray length
N
Y
Temp3[Counter]=Temp1[Counter] V Temp2[Counter]
Step 6
Encrypted Image ← Create_BinaryFile (Temp3)
Step 7
Out pot Encrypted Image
End

Figure 5. The second step of the main image encryption using the key determining the result of the XNOR logical operation

The features of this flowchart are defined based on the following points[15]:

1. This key is created by receiving a file from the user.
2. To create this key, the file format received from the user can be of any type and the number of physical elements of the file received from the user should be bigger than or equal to the physical elements of the image used for cryptography.
3. According to the first feature, the file received from the user is called the key determining the result of XNOR logical operation.
4. To encrypt the secret image using this key, each byte comprising the physical structure of the secret image is processed with its equivalent byte in the physical structure of the file determining the result of XNOR logical operation, using XNOR logical operation and the result of this processing replaces the byte comprising the physical structure of the secret image.
5. XNOR logical operation details: The XNOR gate or XNOR logical operation (sometimes spelled "exnor" or "enor"and rarely written NXOR) is a digital logic gate whose function is the inverse of the exclusive OR (XOR) gate. The two-input version implements logical equality, behaving according to the truth table to the right. A HIGH output (1) results if both of the inputs to the gate are the same. If one but not both inputs are HIGH (1), a LOW output (0) results[16].

Table 3. XNOR Truth Table

Input		**Output**
A	B	
0	0	1
0	1	0
1	0	0
1	1	1

6. The result of XNOR logical operation is used through the following formula:

$(A \leftrightarrow B) \Leftrightarrow (A \wedge B) \vee (\neg A \wedge \neg B)$

7. According to the fourth feature, the file determining the result of the XNOR logical operation is called the key determining the result of the XNOR logical operation.

According to the second step of the main image encryption, the following table indicating the symmetric keys in the second step of the main image encryption**:**

Table 4. The table indicating the symmetric keys in the second step of the main image encryption

Keys number	**Keys Name**
1	Type of logical operation
2	the key determining the result of the XNOR logical operation

2.3 The algorithm to decrypt the main image encrypted by each key

2.3.1 The algorithm to decrypt the main image encrypted by the key determining the result of the XNOR logical operation

The first step of the main image decryption using the key determining the result of the XNOR logical operation defined based on the following Flowchart [15]:

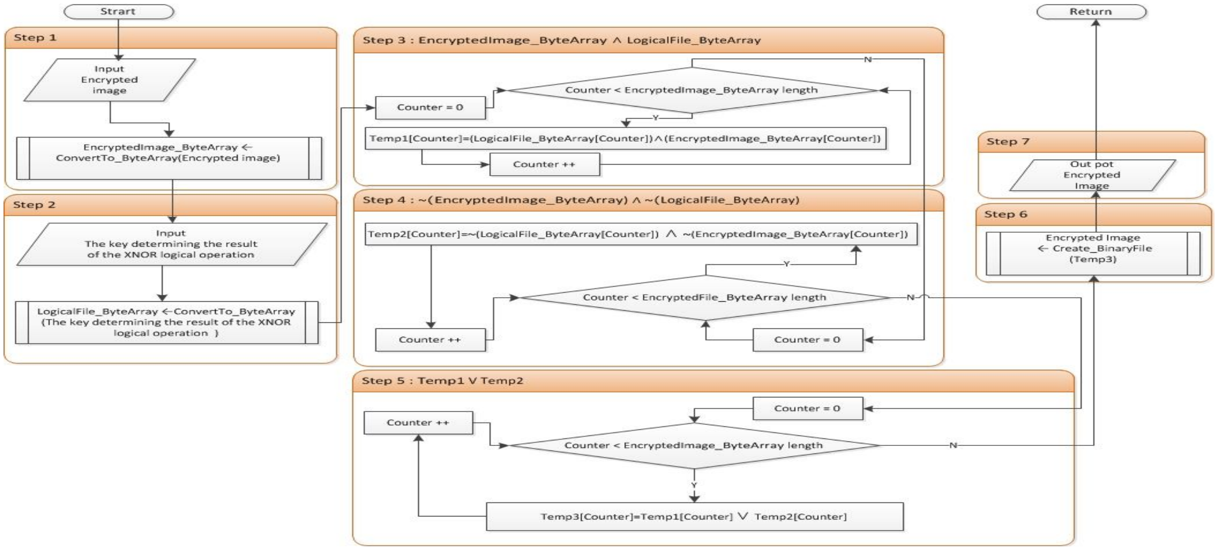
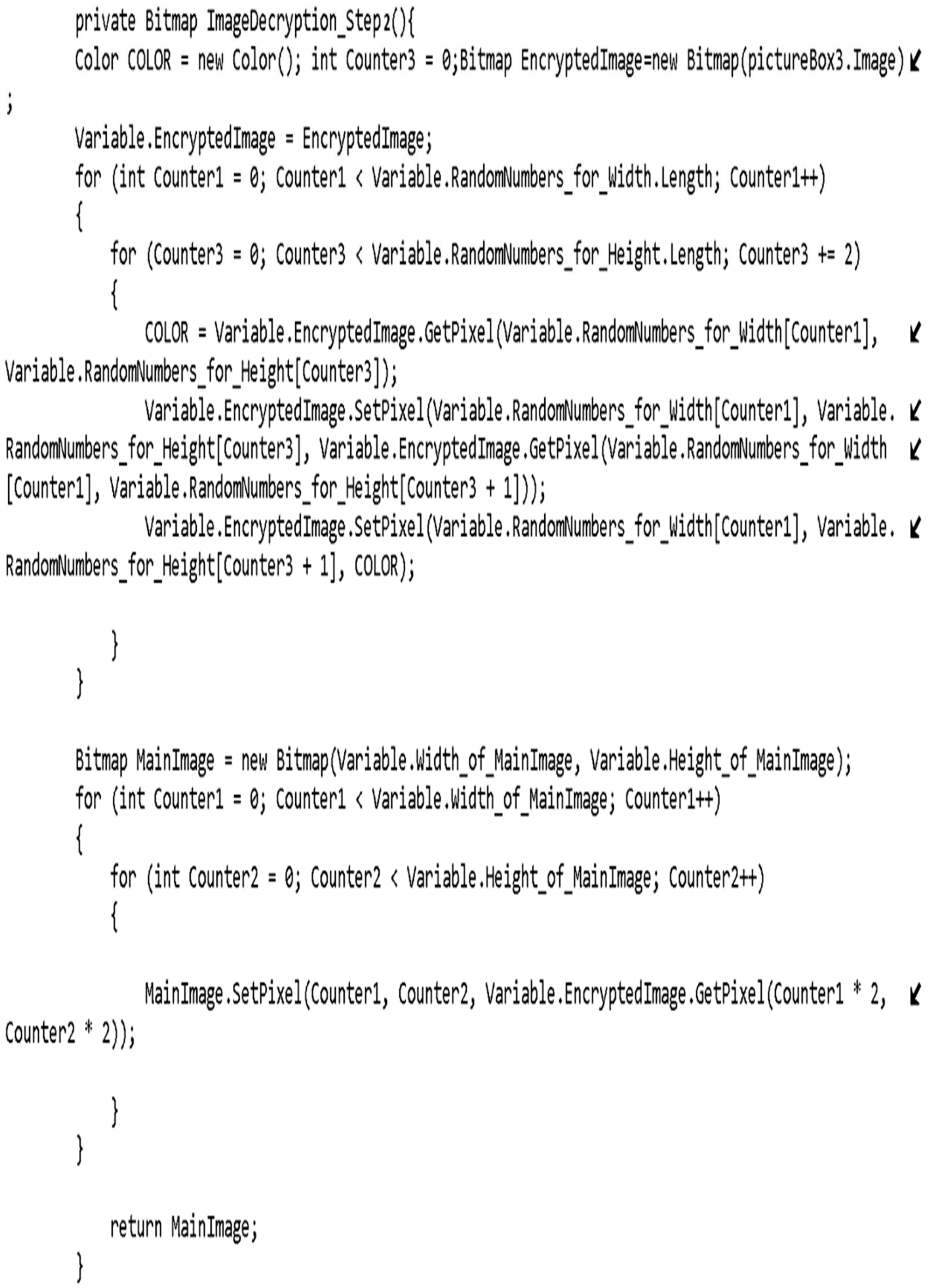

Strart
Step 1
Input Encrypted image
EncryptedImage_ByteArray ← ConvertTo_ByteArray(Encrypted image)
Step 2
Input The key determining the result of the XNOR logical operation
LogicalFile_ByteArray ←ConvertTo_ByteArray (The key determining the result of the XNOR logical operation)
Step 3 : EncryptedImage_ByteArray ∧ LogicalFile_ByteArray
Counter = 0
Counter < EncryptedImage_ByteArray length
Y
N
Temp1[Counter]=(LogicalFile_ByteArray[Counter])∧(EncryptedImage_ByteArray[Counter])
Counter ++
Step 4 : ~(EncryptedImage_ByteArray) ∧ ~(LogicalFile_ByteArray)
Temp2[Counter]=~(LogicalFile_ByteArray[Counter]) ∧ ~(EncryptedImage_ByteArray[Counter])
Y
Counter < EncryptedFile_ByteArray length
N
Counter ++
Counter = 0
Step 5 : Temp1 V Temp2
Counter = 0
Counter ++
Counter < EncryptedImage_ByteArray length
N
Y
Temp3[Counter]=Temp1[Counter] V Temp2[Counter]
Return
Step 7
Out pot Encrypted Image
Step 6
Encrypted Image ← Create_BinaryFile (Temp3)

Figure 6. The first step of the main image decryption using the key determining the result of the XNOR logical operation

2.3.2 The algorithm to decrypt the main image encrypted by the symmetric keys in the table 2

The second step of the main image decryption defined based on the following source code:



```
        private Bitmap ImageDecryption_Step2(){
        Color COLOR = new Color(); int Counter3 = 0;Bitmap EncryptedImage=new Bitmap(pictureBox3.Image)↙
;
        Variable.EncryptedImage = EncryptedImage;
        for (int Counter1 = 0; Counter1 < Variable.RandomNumbers_for_Width.Length; Counter1++)
        {
            for (Counter3 = 0; Counter3 < Variable.RandomNumbers_for_Height.Length; Counter3 += 2)
            {
                COLOR = Variable.EncryptedImage.GetPixel(Variable.RandomNumbers_for_Width[Counter1],    ↙
Variable.RandomNumbers_for_Height[Counter3]);
                Variable.EncryptedImage.SetPixel(Variable.RandomNumbers_for_Width[Counter1], Variable. ↙
RandomNumbers_for_Height[Counter3], Variable.EncryptedImage.GetPixel(Variable.RandomNumbers_for_Width  ↙
[Counter1], Variable.RandomNumbers_for_Height[Counter3 + 1]));
                Variable.EncryptedImage.SetPixel(Variable.RandomNumbers_for_Width[Counter1], Variable. ↙
RandomNumbers_for_Height[Counter3 + 1], COLOR);

            }
        }

        Bitmap MainImage = new Bitmap(Variable.Width_of_MainImage, Variable.Height_of_MainImage);
        for (int Counter1 = 0; Counter1 < Variable.Width_of_MainImage; Counter1++)
        {
            for (int Counter2 = 0; Counter2 < Variable.Height_of_MainImage; Counter2++)
            {

                MainImage.SetPixel(Counter1, Counter2, Variable.EncryptedImage.GetPixel(Counter1 * 2,  ↙
Counter2 * 2));

            }
        }

            return MainImage;
        }
```

Figure 7. The second step of the main image decryption: decrypt the image encrypted by the symmetric keys in the table2

2.4. Cryptography and decryption process

To encrypt and decrypt the secret image using the cryptography algorithm defined in this paper, the following table and the fallowing points should be taken into account:

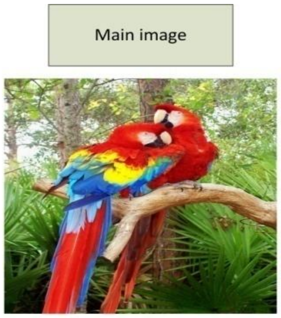
1. The cryptography algorithm defined in this paper is categorized as a symmetric key cryptography algorithm.
2. The physical structure of the created keys in the encryption and decryption are of 6 types and the number of (the key determining the result of the XNOR logical operation) is unlimited. To increase the number of keys, in the second step of main image encryption, the user can encrypt the secret image in the successive stages.

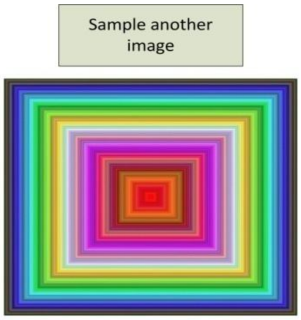
3. With regard to the second feature, in case the user encrypts the secret image more than once, he has to follow the reverse of the encrypting stages to decrypt it.

Table 5. Cryptography and decryption process

Number of independent algorithm	**Type of Process**	**Step 1 : Start**	**Step 2 : End**
1	Cryptography	Algorithm: 2.2.1	Algorithm: 2.2.2
	Input / Out put	M,S,W,H,R,T E	E,O E
	Decryption	Algorithm: 2.3.1	Algorithm: 2.3.2
	Input / Out put	E,O E	E,R,T D
letter	**The meaning of each letter used in the table cells**		
M	Main image		
S	Sample another image		
E	Encrypted image.		
D	Decrypted image.		
W	Width of main image		
H	Height of main image		
R	Random numbers for width		
T	Random numbers for height		
O	The key determining the result of the XNOR logical operation.		

Sample of main image encryption and decryption process according to the figure 8 and figure 9
Sample of main image encryption based on the following figure:





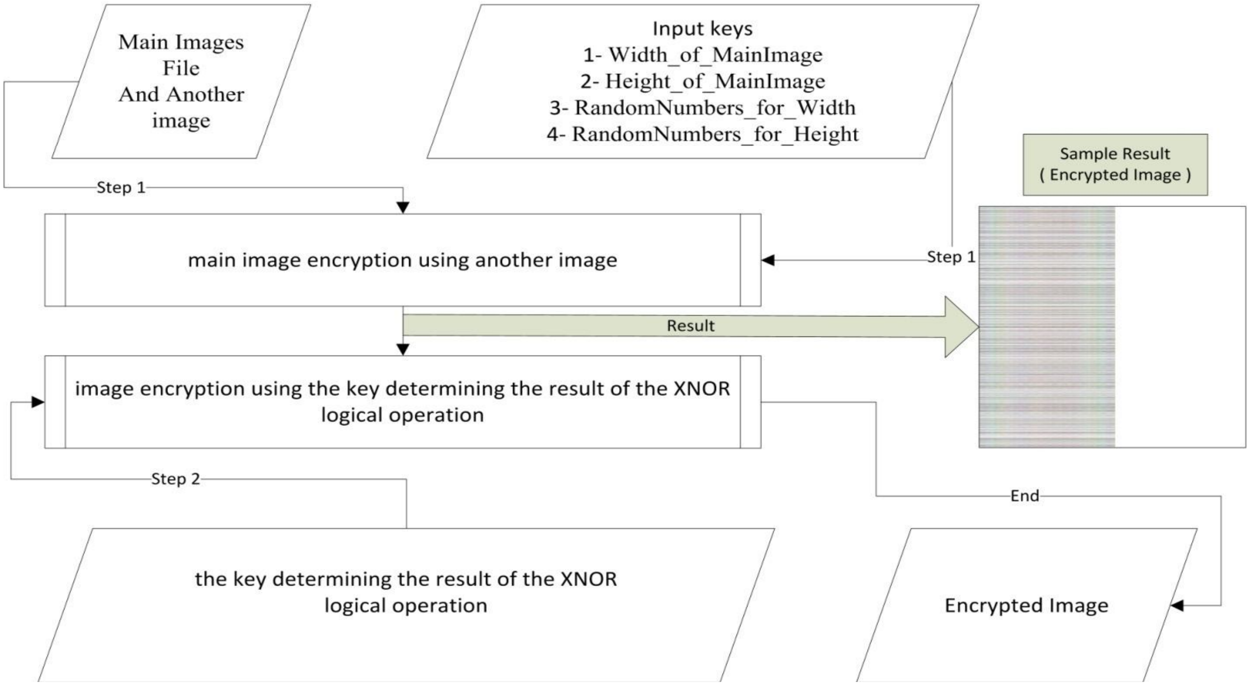

Main Images File And Another image
Input keys
1- Width_of_MainImage
2- Height_of_MainImage
3- RandomNumbers_for_Width
4- RandomNumbers_for_Height
Sample Result (Encrypted Image)
Step 1
main image encryption using another image
Step 1
Result
image encryption using the key determining the result of the XNOR logical operation
Step 2
End
the key determining the result of the XNOR logical operation
Encrypted Image

Figure 8. Sample of main image encryption

Sample of main image decryption based on the following figure:

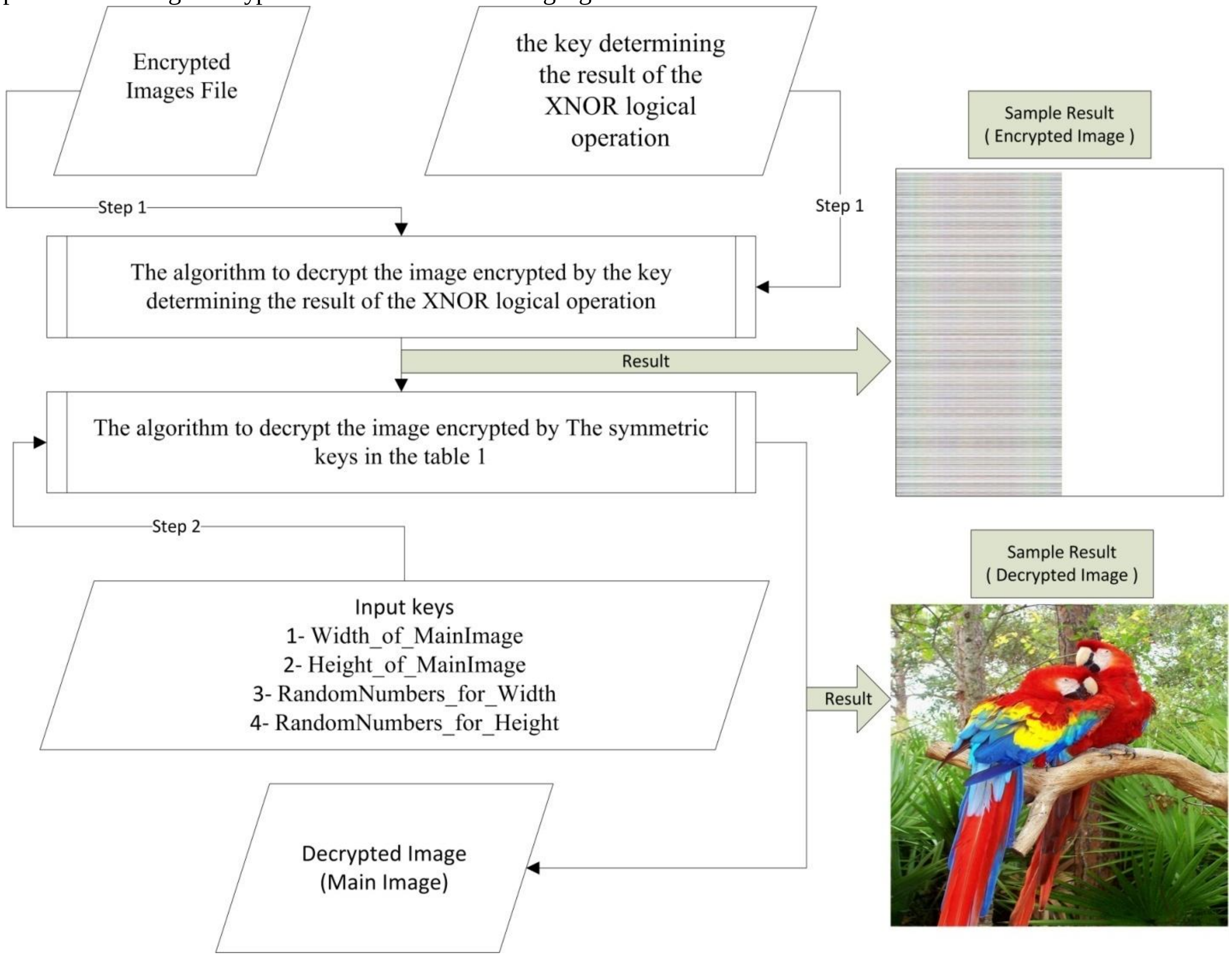

Encrypted Images File
the key determining the result of the XNOR logical operation
Step 1
Step 1
Sample Result (Encrypted Image)
The algorithm to decrypt the image encrypted by the key determining the result of the XNOR logical operation
Result
The algorithm to decrypt the image encrypted by The symmetric keys in the table 1
Step 2
Sample Result (Decrypted Image)
Input keys
1- Width_of_MainImage
2- Height_of_MainImage
3- RandomNumbers_for_Width
4- RandomNumbers_for_Height
Result
Decrypted Image (Main Image)

Figure 9. Sample of main image decryption

3. Conclusion

Visual cryptography technique is widely applied in cryptographic field [17] and also During the last decades, Visual cryptography schemes have been extensively investigated since their invention and extended to numerous applications such as visual authentication and identification, steganography, and image encryption[18]. Cryptography algorithms are classified in to two types, Symmetric-key producing and Public-key producing algorithms. In this paper I suggested a new robust and secure image encryption algorithm based on symmetric key encryption and also in this cryptography algorithm using visual cryptography and logical cryptography together. briefly mention the performed task in each section of the paper: Section 2.1 discusses the type of selectable image for cryptography, Section 2.2 discusses the main Image encryption steps and the creation method of keys and resultantly the secret image cryptography using each of them, Section 2.2.1 discusses The first step of the main image encryption using another image to combine with the main image and obfuscate the main image, Section 2.2.2 discusses the second step of main image encryption using the key determining the result of the XNOR logical operation. Section 2.3 discusses The algorithm to decrypt the main image encrypted by each key, Section 2.3.1 discusses the algorithm to decrypt the main image encrypted by the key determining the result of the XNOR logical operation, Section 2.3.2 discusses the algorithm to decrypt the main image encrypted by the symmetric keys in the table 2, Section2.4 discusses cryptography and decryption process.

REFRENCES

1. Sen, J., ed. *Theory and Practice of Cryptography and Network Security Protocols and Technologies*. 2013, InTech.
2. Sapna Sasidharan, D.S.P., *A FAST PARTIAL IMAGE ENCRYPTION SCHEME WITH WAVELET TRANSFORM AND RC4.* International Journal of Advances in Engineering & Technology, 2011. **1**(4): p. 322-331.

3. Sheeraz Arif, R.H., Syed Wasif Ali Shah, Ahmed Sikander, *Security Key Generation Algorithm for User Identification in Voice over IP (VOIP) Networks.* Journal of Basic and Applied Scientific Research, 2011. **1**(12): p. 3143-3148.
4. Amir Ghotbi, N.N.G., *Evaluating the Security Actions of Information Security Management System in the Electronic Stock Commerce, and Providing the Improvement Strategies.* Journal of Basic and Applied Scientific Research, 2012. **2**(3): p. 3046-3053.
5. S. Behnia, A.A., A. Akhavan, H. Mahmodi, *Applications of tripled chaotic maps in cryptography.* Chaos, Solitons & Fractals, 2009. **40**(1): p. 505-519.
6. Walton, R., *Cryptography and trust.* Information Security Technical Report, 2006. **11**(2): p. 68-71.
7. Chang-Doo Lee, B.-J.C., Kyoo-Seok Park, *Design and evaluation of a block encryption algorithm using dynamic-key mechanism.* Future Generation Computer Systems, 2004. **20**(2): p. 327–338.
8. Kerckhoffs, A., *la cryptographie militaire*. Journal des sciences militaires, 1883. **IX**: p. 5-83.
9. Kerckhoffs, A., *la cryptographie militaire*. Journal des sciences militaires, 1883. **IX**: p. 161-191.
10. Sinha, P.K., *Computer Fundamentals.* 2004: BPB Publications.
11. Kendal, S., *Object Oriented Programming using C#.* 2011: Ventus Publishing ApS.
12. Saurabh Nandu, W.M.L., *C# . NET, Web Developer's Guide*. Copyright © 2002, United States of America: Syngress Publishing, Inc.
13. Microsoft. *Types of Bitmaps*. 2013.
14. Ya-Lin, S. and B. Chen-Xi, *Research and Analysis of Image Processing Technologies Based on DotNet Framework.* Physics Procedia, 2012. **25**: p. 2131-2137.
15. Soltani, M., *A NEW ROBUST CRYPTOGRAPHY ALGORITHM BASED ON SYMMETRIC KEY TO PREVENT UNAUTHORIZED ACCESS TO CONTENTS OF ENCRYPTED FILES. IJCSI International Journal of Computer Science* Issues, 2013. **10**(2): p. 444-452.
16. Rao, G.S., *Discrete Mathematical Structures*. 2002: New Age International.
17. Lou, D.-C., H.-K. Tso, and J.-L. Liu, *A copyright protection scheme for digital images using visual cryptography technique.* Computer Standards & Interfaces, 2007. **29**(1): p. 125-131.
18. Tsai, D.-S., T.-H. Chen, and G. Horng, *A cheating prevention scheme for binary visual cryptography with homogeneous secret images.* Pattern Recognition, 2007. **40**(8): p. 2356-2366.

Mohammad Soltani was born in Kerman, Iran in May 1991. He is currently pursuing his B.S. degree in the department of computer engineering at Shahid Bahonar University of Kerman. His research interests include image processing, cryptography and modern physics. He was announced as the top young researcher in Mahani Scientific Festival based on his scientific curriculum vitae (CV) and articles. He was also accepted as a young scientific scholar in the ministry of science, research and technology in Iran. In addition, he managed to make his way to Khrazami Young Festival (Khwarizmi international award) owing to the results of his scientific studies.