Global Journal of Information Technology: Emerging Technologies

Volume 7, Issue 1, (2017) 187-195

www.gjit.eu

Pixel-sieve cryptographic primitives with LSB steganography

Arpad Incze[*],Department of Mathematics and Computer Science, 1 Decembrie 1918, University of Alba Iulia, 510009, Alba Iulia, Romania

Suggested Citation:
Incze, A. (2017). Pixel-sieve cryptographic primitives with LSB steganography. *Global Journal of Information Technology: Emerging Technologies*. *7*(1), 187-195

Received November 21,2016; revised February23, 2017; accepted April 20, 2017.
Selection and peer review under responsibility of Prof. Dr. Dogan Ibrahim, Near East University, Cyprus.

Abstract

This paper contains a brief description of new approach regarding LSB steganography. The novelty of the method resides in the combination of LSB (Least Significant Bits) steganography with some primitives of the pixel-sieve/bit-sieve cryptographic method. In short, we propose to use two or more carrier images and the sieving algorithm, borrowed from the pixel sieve primitive, to determine which carrier image will receive the next set of bits of the secret message. While in classic LSB steganography the secret message must be encrypted prior to embed the information into the carrier image, in our proposal the message is scrambled between the shares in a pseudo random way. An attacker will need all the carrier images and the sieving key in order to reconstruct the original message. Also we recommend an alternative method in which instead of simply replacing the last bit/bits we use them as XOR keys to further enhance the security.

Keywords: steganography, cryptography, secret sharing; visual cryptography, LSB.

* ADDRESS FOR CORRESPONDENCE: **Arpad Incze**, Department of Mathematics and Computer Science,1 Decembrie 1918, University of Alba Iulia, 510009, Alba Iulia, Romania. *E-mail address*: aincze@uab.ro / Tel.: +40-766655394

1. Introduction- Prerequisites

1.1. Steganography

Steganography is a branch of cryptography in which the secret information is hidden on sight. Most often this is done by hiding useful information on images. Such an example is described by Sir Robert Baden-Powell in Baden-Powell & Baron (2011). In figure 1 the picture of the butterfly also shows the position of artillery inside a military fortress.

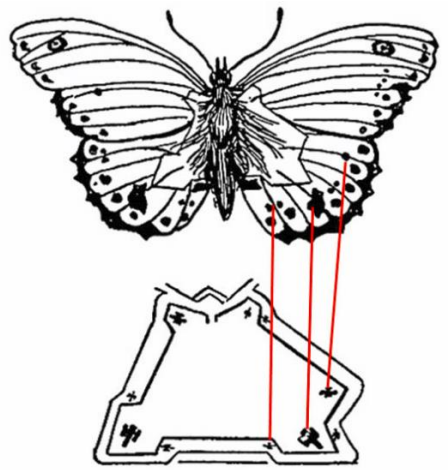

Fig. 1 Historical Steganography example from Baden-Powell & Baron (2011)

In computing steganography is achieved by hiding bits of the secret message in multimedia files. For ex; this pictures, movies, audio files are used as carriers (Joseph & Sundara, 2011; Gutte & Chincholkar, 2012). The most obvious reason to use steganography instead of plain cryptography is that while sending a file with interpretable information (eg. a picture) is far less suspicious than an encrypted file. Let's take a look at fig. 2. The image box from the right is obviously an encrypted one while the image of Lena on the right those not rise suspicion.

Fig. 2 Encrypted image a) vs steganography carrier b)

1.2. LSB Steganography

The most common form of image steganography is LSB steganography. LSB stand for Least Significant Bit (some times Last Significant Bit) The method consist in breaking down a picture to its

pixels and for each pixel the RGB color components are altered by changing the least (last) bit with bits of the secret message. The resulting image is altered in such an insignificant manner that a human eye cannot observe the change. For instance a 24 bit picture means that each color component has 8 bits. By changing the last bit of each component one pixel can carry 3 bits. By changing the last two bits, one pixel can carry 6 bits (Gutte & Chincholkar, 2012).

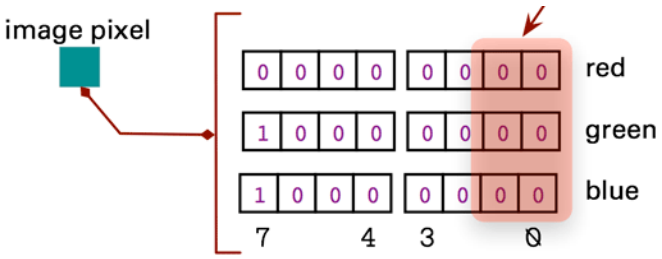

Fig. 3 LSB with 2 bits/color.*

Usually the change in color is insignificant for the human eye. Look carefully the two pictures in figure 4. Although they seem identical actually the left image contains the message „I should be able to hold 37 bytes". Because we replaced the last bit for every color component of every pixel and there are 10x10 pixels x3 color component = 300 last bits which can take 37 bytes =37*8=296 bits.

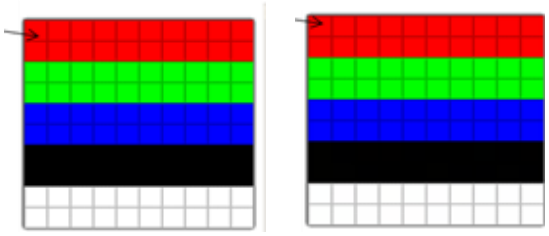

Fig. 4 Two Almost identical images by color.

The cryptanalysis of steganography is called steganalysis. It is based mostly in discovering altered pixels in pictures. Once a picture identified as steganographyc carrier it is easy to extract the hidden information. Therefore usually the information is encrypted with some known cryptographic method before being embedded into the carrier and sent over the network.

Thus a standard approach to LSB steganography consist in these steps:

1. encrypting the message
2. embedding the encrypted message into the carrier

To retrieve the information at the destination the user who receives the image will have to:

1. extract the bits and reconstruct the encrypted message
2. decrypt the message

* http://www.drdobbs.com/security/how-to-secure-and-authenticate-images-us/229400454

1.3. Pixel sieve/bit sieve cryptography

The pixel-sieve Incze (2010) and bit-sieve Incze, Moldovan & Muntean(2010) cryptographic methods are cryptographic primitives. This means that they are basic ideas around which a more complex cryptographic method can be built.

Briefly, the pixel/bit-sieve is a 2 by 2 secret sharing cryptographic method. The sieving process is used to copy the pixels/bits of the original image/message in a certain share according to the value of a bit of the key. Depending on the current bit of the key (0 or 1), the bits of the message are sent to *Share0* or *Share1*. Each time a share receives a bit from the original message a random bit is generated as noise (marked "x") and added to the other share as noise by this further concealing the useful information. The bit sieve process is presented in Fig. 5

Clear text	1	0	0	1	0	1	0	1	0	1	0	0	1	1	1	1
Key	0	0	0	0	1	1	0	1	0	1	1	0	1	0	1	0
Share 1	x	x	x	x	0	1	x	1	x	1	0	x	1	x	1	x
Share 0	1	0	0	1	x	x	0	x	0	x	x	0	x	1	x	1

Fig. 5. The basic principle of bit-sieving

During the testing of the method several weaknesses emerged especially in the pixel sieve version. To solve those weaknesses enhancements were proposed by the author himself and also by other researchers too.

Thus in Incze (2014) the author proposes an LFSR like key expansion algorithm. With the proposed method from an n length initial key an $2n^2$-1 length extended key can be generated.

Another enhancement in Incze (2014) solves an issue regarding an unequal distribution of the original data in the shares. Because the number of 0-s an 1-s of the key will determine how many bits/pixels will receive each share, if the ration between 0 and 1 is in favor of a certain share, that share will receive more data. In the case of pixel sieve this can mean a visually interpretable share. To solve this issue a threshold is introduced. Each time a share gets a bit/pixel a counter is incremented. If the counter reaches the threshold the shares are swapped so in the end each share will receive a fare amount of data.

2. Literature review

In the original pixel sieve method each pixel of the key sieve encrypts only the corresponding pixel in the original image. Any pixel of key does not affect the encryption or decryption process of other pixels. Hence, if we use a key with some incorrect pixels to decrypt the image, only corresponding pixels will be decrypted incorrectly, while other pixels will be decrypted successfully. To remove this problem key sieve shifting method is proposed by Choudhary, Kumar, Kumar & Singh (2011). Also a cross merge and key shifting is applied (Choudhary, Kumar, Kumar, & Singh, 2011).

Another team has embedded the sieving technique in a more advanced encryption algorithm (Venkatesh & Roopanjali, 2013). The algorithm is mainly divided into three steps they are: sieving, dividing and, shuffling (SDS). The sieving involves the secret image splitting into primary colors. The second important step is division, which involves the random division of the split image. In the third step, the divided Shares are shuffled randomly as in (Malik, Sardana & Jaya, 2012).

A modified version of pixel sieve method is proposed by Koteswari, Paul & Indrani(2012). Such as; possible use in bio-metric identification and protection using iris images to achieve more security. It uses the modified version of pixel sieve and is based on key shifting scheme.

In Patil & Udupi (2013), a new enhanced encryption method is introduced using visual photographic scheme which is a hybrid of the traditional VCS and the conventional image encryption schemes. A secret image is split into multiple random images using sieving and with minimum computation the original secret image can be retrieved back.

3. Sieving in multiple carriers

To further secure the LSB method we propose to use not one but two or more carriers. Considering the amount of pictures sent over the internet it would be not suspicious at all such an activity of sharing images.

In our approach we propose to adapt the sieving algorithm for steganography as follows: the bits of the binary key will determine which carrier will receive the next bit of the secret message. By this the bits of the secret are spread between the two images. Figure 6 illustrates the process.

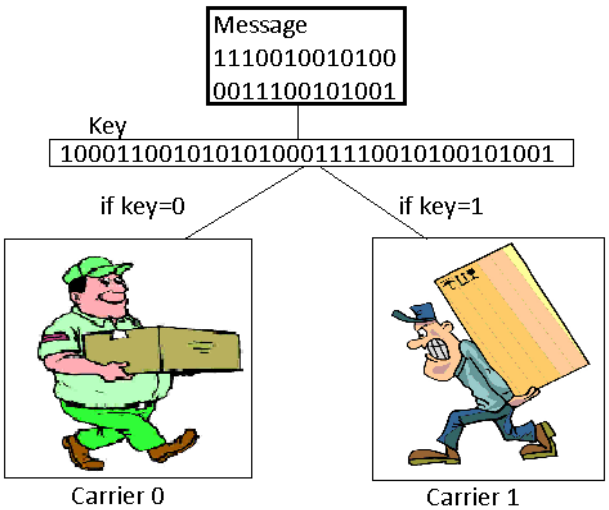

Fig. 6 Sieving the message between the carrier images

When we speak about current position in carrier we can mean the current pixel or the current color component. If we consider the current pixel this translates in weaker security because one pixel holds 3 or 6 consecutive bits from the secret message. For better security we strongly recommend the second option where in current position we mean the current color component. In this case we get a better spreading of the bits between the carriers. For the simple case of working with ASCII codes with one byte representation of characters the bits of the same byte can be spread on both of the carriers.

Especially if we store two bits in each color component it is possible that one component of one pixel will store bits from different bytes as shown in figure 7 where the pixels of both carriers stores bits from both bytes.

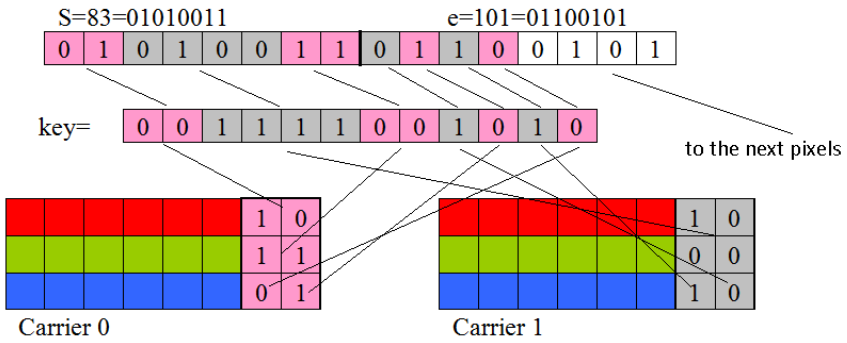

Fig. 7 spreading of the bits in the carriers

One issue of the pixel sieve primitive was if there is an advance in both shares regardless which share receives information or there is advance to the next position only in the share that receives data. Both situations are having advantages as well as disadvantages. Those issues were thoroughly discussed in Incze (2010) and Incze, Moldovan & Muntean (2010). The same question rises in the case of steganography with sieving primitive. We can have both versions in this case too with the ups and downs. In a nutshell if we have advance in both carriers the size requirements are higher but also the strength of the method is higher. If the user has some size limitations (eg. bandwidth) the no-advance implementation is recommended with arguably weaker security.

In both cases the security can be enhanced with XOR cryptography.

Several scenarios emerge depending from some initial assumptions regarding the images used as carriers. The question is: do the participants share or not an initial set of images ?

3.1. No initially shared images

In this approach the sender of the encrypted message can use any image. The receiver only needs the key to extract and decrypt the message. The value of the key shows us which carrier image contains the next correct bit/bits.

To further enhance the security of the method the bits of the message can be XOR-ed with the bits of the key before being inserted to the corresponding carrier image.

The carrier image which does not receive a bit thru the key will receive a random bit to preserve the advance in the carriers. In this case we have advance in both carriers. This is also to strengthen the method because noise is added.

With the above mentioned proposals we can formalize the followings (1)

Let $\mathbf{M_p}$ be a clear text (binary)

K key (also binary)

$\mathbf{I_{0p}}$ and $\mathbf{I_{1q}}$ the least significant bit of the color components of the pixels of the carrier

$$I_{0p} = \begin{cases} M_p \otimes K_i \ \ if \ \ K_i = 0 \\ Random(0,1) \ \ if \ \ K_i = 1 \end{cases} \qquad (1)$$

$$I_{1q} = \begin{cases} M_p \otimes K_i \ \ if \ \ K_i = 1 \\ Random(0,1) \ if \ \ K_i = 0 \end{cases}$$

To extract the useful information the recipient first will extract the bits from the right positions indicated by the key and then he will decrypt the message with simple XOR cryptography using the same key.

As you can see the key has two roles.

- To XOR encrypt/decrypt the message
- To determine which carrier image will contain the bits of the message

Using random bits the size requirements for the carrier images are higher, but the security of the method is stronger. In case that an attacker captures both the carriers, without the key he will have serious difficulties assembling the bits of the message in the right order before even be able to try to decrypt the secret message

Actually with this method, in certain conditions, we can skip the supplementary encryption of the data. One such condition would require from the key to have a fair spread of 0's and 1's. Otherwise if the key would have big blocks of 0's or 1's here is a risk that meaningful blocks of the secret message are inserted in only one carrier, allowing to an attacker to reconstruct parts of the message or to find the key in case of a clear text attack. A threshold-swap algorithm proposed in Incze (2014) will solve this issue to ensure a fair ration and spreading of 0's and 1's of a given key.

If the key has a fair amount of spreading we can also eliminate the need of random bits inserted into the carriers. By this the size requirements of the images can be reduced still preserving a fair amount of security.

3.2. Initially shared images

In this approach we will increase the number of items needed to correctly decrypt the message.

For this instead of simply replacing the last one or two bits of the color components of the pixels, we will also XOR the bits of the message with the bits of the key and with the bits of the carrier image.

M clear text (binary)
K key (also binary)
$\mathbf{I_{0p}}$ and $\mathbf{I_{1q}}$ the least significant bits of the color components of the pixels of the carrier images in positions p and q

$$I_{0p} = \begin{cases} (M_k \otimes K_i) \otimes I_{0p} & if\ K_i = 0 \\ Random(0,1) & if\ K_i = 1 \end{cases} \qquad (2)$$

$$I_{1q} = \begin{cases} (M_k \otimes K_i) \otimes I_{1q} & if\ K_i = 1 \\ Random(0,1) & if\ K_i = 0 \end{cases}$$

But this approach means that both participants share an initial set of images and a secret key.
To decrypt the information the receiver will have to do the following steps:

- Extract the useful bits from the carriers with the key
- Decrypt the information using the bits key and the bits of the original image
- The fact that the original image is also needed to decrypt the message gives the method some serious strength. In case that an attacker captures the carriers he will need not only the key used to scramble the information but the original images too for the XOR-ing part of decryption. In this case the initial images act like cryptographic keys.

4. Conclusion- Further research.

The proposed method brings some improvements for the LSB steganography. Also the supplementary encryption/decryption of the message is not mandatory. The added security resides in the fact that the secret is distributed among several carrier images. Also while a set of images can be identified by steganalysis as being steganographyc carriers, to rebuild the hidden message an attacker will need all the carriers AND the binary key to extract in the right order the information.

If a set of initially shared images are used the attacker is helpless without those images.

The images can have different sizes as long as they can harbor the required amount of data. Bigger images with random data inserted (padding) can fool an attacker making him to wrongly assume the size of the message.

There are multiple choices of approach depending on the needs of the user:

- Smaller images and without encryption for quick preparation and transmission of the message with acceptable security
- Big carrier images (when random bits are inserted) and supplementary encryption for high security.

Because the described method is an empirical one, a thorough theoretical and practical analysis is required. Also opinions, observations from the scientific community are welcome.

After a thorough theoretical analysis of the proposed method, the next step is to write a software application for live testing the method.

References

Baden-Powell, R., & Baron, R. S. S. B. P. (2011). My adventures as a spy. Courier Corporation.

Choudhary v., Kumar P., Kumar K., & Singh, D.S. (2011). An improved Pixel Sieve method for Visual Cryptography. *International Journal of Computer Applications, 12*(9)

Choudhary, V., Kumar, P., Kumar, K., & Singh, D. S. (2011). Modified pixel sieve method for visual cryptography. *Indian Journal of Computer Science and Engineering, 1*(4), 321-326

Gutte, R. S., & Chincholkar, Y. D. (2012). Comparison of Steganography at One LSB and Two LSB Positions. *International Journal of Computer Applications, 49*(11), 1-7.

Incze, A. (2010). *Pixel Sieve method for secret sharing & visual cryptography*. Proceedings of the 9th RoEduNet IEEE International conference, Sibiu, 24-25 June

Incze, A. (2014). Solutions regarding some cryptographic key issues for the pixel-sieve cryptographic method. *4rd World Conference on Innovation and Computer Sciences 2014*. *Procedia Information Technology & Computer Science, 4*

Incze, A., Moldovan Gr., & Muntean M. (2010). From pixel sieve to bit sieve. Bit level based secret sharing cryptographic method", in proceedings 11th International symposium CINTI, Budapest 18-20 November

Joseph, A., & Sundaram, V. (2011). Cryptography and steganography–A survey.

Koteswari, S., Paul, P. J., & Indrani, S. (2012). VC of IRIS Images for ATM Banking. *International Journal of Computer Applications, 48*(18), 1-5.

Malik, S., Sardana, A., & Jaya, J. (2012). A keyless approach to image encryption. In Communication Systems and Network Technologies (CSNT), 2012 International Conference on (pp. 879-883). IEEE.

Patil, V. S., & Udupi, R.(2013). A secure approach to image encryption of color image without using key.*International Journal of Current Engineering and Technology.* Retrieved from; http://inpressco.com/category/i jcet

Petitcolas, F.A.P., Anderson, R. J., & Kuhn, M.G. (1999) "Information Hiding -A Survey", Proceedings of the IEEE, Special issue on Protection of Multimedia Content, International Journal of Database Management Systems (IJDMS), 87(7), 1062- 1078.

Venkatesh, M.R., & Roopanjali, D. (2013). SDS Technique For Secret Image Encryption. *International Journal of Engineering Research & Technology (IJERT), 2*(4)