Improved Security Based on Combined Encryption and Steganography Techniques

Ştefan MOCANU, Andrei DULUŢĂ, Daniel MEREZEANU, Radu PIETRARU

Faculty of Automatic Control and Computer Science, University Politehnica of Bucharest,
Splaiul Independentei 313, 060042, Romania.
stefan.mocanu@upb.ro

Abstract: For some people a science, for other people just an art, steganography is in fact an ancient method of embedding a message into an apparently uninteresting carrier. Aiming to protect information, steganography is considered to be a close relative of encryption although they have different approaches. While encryption scrambles the message based on a certain algorithm so it can't be understood by anyone that does not hold the unscrambling key, steganography works on hiding the message without the need to deteriorate it. Minor alteration of the carrier file is accepted since steganography does not affect its useful content. Actually, steganography exploits file redundancy, file headers or, in case of multimedia content, replaces information that can't be used or perceived by the human eyes or ears. In this paper, a combination of image based steganography with encryption is presented. The encryption and decryption are based on two different files in which, by steganographic means, important information is hidden. The original message (the secret message) can be restored only if both files arrive safe at destination. There are no constraints regarding the image file types. The most important contribution resides in how the original information is encrypted and embedded into different graphic files, sent to the destination through different channels and then restored. Comparative performance tests were performed.

Keywords: Digital steganography, encryption, pseudo random matrix, message hiding.

1. Introduction

From the earliest times participants to remote conversations felt the need to ensure the privacy and confidentiality of that conversation. Today, an immediate solution may consist in encrypting the conversation using more or less complicated encryption procedures. This would lead to a visible deterioration of the original message which would be, in most cases, almost impossible to be decoded by a third party if this entity lacks the necessary information for decryption.

There are, however, situations in which some may prefer alternative techniques which can guarantee hidden transfer of information between a source and a destination without a noticeable deterioration of the carrier. This is the case of steganography (*gr. steganos* + *graphein* = hidden writing), a technique that allows embedding of sensitive content into messages/carriers that appear harmless and that raise no suspicions for an interceptor.

First steganographic techniques were used in Ancient times. From the wax tablets and slaves' scalps tattoos, steganography evolved a lot during the medieval times and The Second World War, a global event that had serious implications over the development of communication technologies and not only. The limited available resources of the times involved in hiding and extracting of information as well as the efficiency of this method turned steganography into a serious option for transferring secret information. The most popular form of steganography of that time is represented by the *null cipher*. An extensive analysis of various null ciphers is presented in [5].

Digital steganography is the normal evolution of the former analogical steganographic techniques. Digital steganography is a generic definition for all techniques that make possible the inclusion of various forms of information (text, image, video[3]) into various formats of carrier containers (static image or text files, video or sound files, network headers and other). Many studies [12], [13], [20] reveal the fact that multimedia files are preferred when steganography is implemented. This choice is based on particular characteristics of multimedia files [11], [16] that allow embedding information without a visible impact or modification of the carrier [17], [19].

There are many studies which reveal the fact that more than a lot of the multimedia/image files from the Internet contain various forms of hidden messages. Although most of the steganographic applications aim at protecting individual privacy, some authors [3], [21], [22] fear that steganography may be used for malicious or dangerous purposes.

As a normal consequence of the development of new and more robust steganographic techniques, the development and improvement of detection methods closely followed. It is the case of steganalysis, a term which denotes a wide variety of techniques that are able to detect the presence of steganographic content inside a digital file. This is not an easy task because we seldom have access to an original and a modified version of the same file. In such cases, by direct comparison, a difference between files could be easily detected. However, this is not the case we deal in reality. Moreover, most digital multimedia files can be modified without leaving many traces.

Steganalysis may be able to detect the existence of some additional content but the process of extracting it requires a lot of time and resources [17], [11], [23]. This task can be much more difficult if the embedded content was previously encrypted. The basic idea is that encrypted content itself can be identified as suspicious without many efforts even if this does not mean someone may easily decrypt it. Adding steganography over encryption should decrease the chances to discover encrypted content.

The main goal of this paper is to propose a robust and secure method of communication based on a combination of steganography with a light form of cryptography. The most original contribution resides in how the original information is encrypted and embedded into graphic files, divided into several different objects, sent to the destination through different channels and then restored. The rest of the paper is organized as follows: in chapter 2 related work is presented; chapter 3 presents theoretical aspects and implementation details for the proposed technique; chapter 4 is dedicated to experimental results while in chapter 5 conclusions are drawn.

Related Work

During the past years steganographic techniques were subject to investigation of many authors and researchers. The growth of Internet and use of digital multimedia objects offered practically limitless possibilities for implementing steganographic applications. In [6] the authors present the importance of using various techniques (steganography included) for improving the specific capabilities of an Augmented Reality Teaching Platform. Former analogical steganographic techniques dedicated to security applications found their equivalent in digital steganography [3].In networking, steganography can be used in order to implement QoS like mechanisms or to increase the security and privacy of the network data package [8], [9], [14].

In papers [1], [13], [15], [19] the authors present various steganographic techniques. In most cases, steganography exploits the redundancy that is present in various digital files and formats. We are talking about compressed or uncompressed multimedia files (images, films or sounds), or various forms of text files, executable files or databases.

In practice, there are many steganographic methods [2], [7] that rely on a strong mathematical background (Discrete Cosine Transform, Wavelet Transform, temporal or frequency masking) but they require a higher computational effort. However, there are other approaches that aim the modification of parameters such as: most insignificant bit [17], unused bits from files' headers[8], minor modifications of the compression level [20](in case of compressed files, especially multimedia ones), minor modifications of the content (the carrier itself), various conventions (such as: even length means "1", odd length means "0"). These techniques do not require a very complicated theoretical background nor high computational resources. In the same time, the results are more than satisfactory for a normal application.

The best files that can carry a high amount of additional data without visible effects are multimedia files. In their case, specific methods for encoding and compressing the content allows minor changes without significant effects. For instance, in case of audio content, it is widely known that the human ear cannot perceive sounds with a frequency higher than 22 KHz. As a consequence, audio files can carry additional data hidden under the form of high frequencies [2], [7]. In case of image files, the modification of one pixel's color cannot be observed by the human eye, especially when it comes to high resolution pictures. This is one of the reasons why multimedia files are preferred to steganography applications.

Paper [15] presents an extended and critical analysis of the characteristics of various steganographic algorithms. The qualitative

aspects taken into consideration are related to: hiding capacity, complexity, computational effort, robustness, and visible results of the hiding process.

In some certain cases, the privacy and security offered by steganography alone are not good enough. In those cases, additional techniques can be used, such as encryption. However, a balance between encryption/decryption efforts is hard to find when we talk about "classic" encryption methods. Symmetric encryption has the advantage of simplicity but it is more exposed to "brute force" attacks [4], [10]. Encryption based on public and private keys is more robust but it complicates the architecture of the communication system and raises the amount of computational resources required [4], [10].

The solution presented later in this paper is based on simple encryption methods in conjunction with steganography. The encryption and decryption are based on two different files in which, by steganographic means, important information is hidden. The original message (the secret message) can be restored only if both files arrive at destination.

2. Proposed Technique Overview

2.1. Theoretical Background

Encryption

The first stage of the algorithm presented in this paper is the encryption. In conjunction with the mechanisms used to hide the image inside a carrier image, the encryption fulfills an essential role in increasing the effort required for an unauthorized user to interpret the content of the message that was sent.

The main objective of the encryption stage is to process the message image such as the content becomes impossible to understand by simply viewing or using some known image editing techniques (geometrical: rotations, translations, scaling, color and brightness change, contrast, color palette, etc.).

In this respect, the alteration of the three channels of color (R, G, B) was aimed by using relatively simple mathematical procedures, whose application to bring the message closer to a "gray noise" image. These procedures combine the results known in Boolean algebra and trigonometry with practical tests in order to obtain an algorithm that modifies, at pixel level, the message picture using an encryption key that is known only by the sender and the recipient.

An encryption key was used in order to increase the security of communication. If increased robustness of the application is aimed, the encryption key can be obtained by an algorithm of RSA type [10], [18].

The key, the encryption algorithm and the encrypted image may be transmitted over separate communication channels, making more difficult any attempt to intercept communication in a way that favors the deciphering of the message. Basically, in order to restore the original message, the attacker must know the key, the encryption algorithm and the encrypted image.

The encryption algorithm consists of two steps: generating a pseudorandom matrix using encryption key and using this matrix to encrypt the message image.

Pseudo random matrix generation

Generating a set of numbers which are as close as possible to the concept of "random" using minimal computing resources was and still is an important problem both for the Mathematics and IT domains. The cryptography implemented into the studied algorithms and methods uses numbers whose appearance is very hard to estimate.

For the encryption algorithm proposed in this paper it was necessary to create a matrix containing pseudorandom values. Some of the requirements imposed on the matrix are:

- The size of the matrix should be the same with the size of the image to be encrypted
- The generated numbers must be within the range [0, 255]
- The generated values must depend on a number known a priori (the encryption key)
- Values must be the same for an encryption key and a fixed set of dimensions (number of rows and columns) regardless of the number of executions of the program (there is no dependence on any matter at runtime or similar)
- Values depend on the image's channel where they will be applied (R - 1, G - 2 or B - 3)

One of the most affordable solutions which was proved that successfully fulfils the above specifications involves the use of trigonometric functions and nonlinear combinations.

The sine and cosine functions

Very often used in engineering (signal processing, modeling, system synthesis and analysis, etc.), the trigonometric functions present several advantages that are recommending them for the described application. The most significant advantages are:

- Implementation done by optimized algorithms, developed by those from MathWorks.
- Low interdependence between successive values. The numbers are not generated recursively, which is why the search for correlations does not provide useful information for "breaking" the encryption method.
- Obtaining of pseudorandom values for positive integer arguments. The arguments used for these functions are natural numbers belonging to a very large range, as will be shown in the next chapter.
- The periodicity. Being not injective implies that there is no correspondence between the set of the arguments and the values, increasing the difficulty of "breaking" the encryption algorithm.
- Ease of implementation. The number generation is done using only a few lines of code.

Although mentioned issues are particularly important for this application, the use of trigonometric functions involves a number of drawbacks that need to be considered for the future developments:

- The calculation of actual values of these functions consumes significant resources, compared to basic algebraic operations (addition, multiplication, etc.)
- If the arguments are consecutive, a careful analysis of the obtained values can show the evolution of the sine (or cosine).

Nonlinear combinations

As described above, using trigonometric functions is tricky, especially when their integer arguments are very close. For such values, a careful analysis may indicate a harmonic development to an unauthorized person, providing useful clues in order to detect the encryption algorithm.

Based on a set of arguments (consecutive integers) and a corresponding set of values obtained by applying a harmonic function, the identification of the function that generated those values using "fitting" mechanisms can be achieved.

To illustrate the ease of associating a sinusoidal evolution to a set of values it is enough to do a simple test using the "Curve Fitting Toolbox" from the MATLAB package. The procedure is very simple: the sine function has been applied to a set of consecutive numbers (0, 1, 2, 3, 4) obtaining a set of values. Then, an attempt to identify the function that best approximates the obtained values was made and the result was:

$$f(x) = sin(x + 3.416 \cdot 10^{-13}) \qquad (2.1)$$

One can notice the precision with which the procedure of "reverse-engineering" worked. Of course, in case of a large set of values, finding groups that could be regarded as having such an evolution of oscillatory type is a difficult task. However, having the necessary resources, a search for such "templates" can narrow the set of these possible groups and thus can provide a satisfactory solution that can be exploited as shown above.

A possible solution could be the use of linear combinations of trigonometric functions, whose arguments can be various parameters (index of the line or column from the application). This way, getting the coefficients of the harmonic components would become a much more difficult task, in case of using only fitting algorithms. On the other hand, a widely-used instrument in signal processing is the Fourier transform. Applying a FFT algorithm to the values resulted from the use of linear combinations of trigonometric functions, would lead to finding information that could pose a significant starting point in rebuilding encryption algorithm.

In order to avoid these problems, the solution of using nonlinear combinations of trigonometric functions was chosen. To reduce the high consumption of resources involved by these mathematical operations (multiplication, logarithms, extracting the square root, etc.), in order to create those nonlinear combinations of trigonometric functions, the mathematical

operation that ensures a minimum consumption of processor resources (multiplication) was chosen.

Besides complicating the identification of coefficients and arguments of the used functions, using the nonlinear combinations in conjunction with harmonic functions provides an additional advantage. As it will be shown in the next chapter, the operation to generate the pseudorandom values will be achieved using a function like this:

$$f = \alpha \cdot sin(i) \cdot cos(j) = \\ = \alpha \cdot g_1(i) \cdot g_2(j) \quad (2.2)$$

where the arguments key, i, j will be detailed in the next chapter.

It is important to analyze the advantage of the multiply operation in terms of the sign of the resulted value, knowing the signs of the harmonic functions involved. Thus, assuming $\alpha > 0$and knowing the common ranges of *sine* and *cosine* functions we obtain:

Table 2.1.Sign of f, given g_1 and g_2

$\mathbf{sgn(g_1)}$	$\mathbf{sgn(g_2)}$	$\mathbf{sgn(f)}$
-	-	+
-	+	-
+	-	-
+	+	+

Based on the above table it can be concluded that the probability of getting a negative or positive number is the same. Thus, the sign of the result varies, depending on the signs of the values of the harmonic functions used and this way both positive and negative numbers can be equally obtained.

The use of XOR function

The non injective functions whose values are equiprobable have a particularly important role in cryptography, because they are very difficult to reverse, using either deterministic or non-deterministic methods.

The XOR function belongs to the set of functions that have the properties listed above, which recommend it for this kind of application.

To encrypt the message picture, the XOR function is used to perform a "mask" of all the pixels (for all 3 channels of color) with a matrix containing pseudorandom values generated as was detailed in the previous section.

As a result, because the generated values have an almost random distribution and the possible results obtained using the XOR function have the same probability of occurrence, the image obtained by masking is close to a “gray noise" image type.

Euclidean division

Once encrypted, the message image should be hidden in the carrier image. Knowing that it is intended to hide a high volume of information related to the carrier image, a method to alter as little as possible of its informational content has to be found.

Hiding the image itself was done by altering the values for each pixel from the carrier image based on the pixel values from the message image.

This modification must satisfy two essential conditions: to be reversible and least noticeable. To alter the image carrier, as little as possible, the Euclidean division was chosen for dividing the informational content of the message. Denoting by $m_{i,j,k}$the numerical value of the channel k (1-R, 2-G, 3-B) of the pixel from the line i and columnj, from the message picture we obtain:

$$m_{i,j,k} = q_{i,j,k} \cdot 16 + r_{i,j,k} \quad (2.3)$$

It is well known that$m_{i,j,k} \in [0,255]$. From the above relation, we obtain $q_{i,j,k}, r_{i,j,k} \in [0,15]$.

Thus, the range of the values that need to be hidden for each pixel was reduced from 256 to 16 for both the quotient and the remainder. However, the approach has a downside: while using the previous image encryption method, hiding the message in the carrier image was generating a single image as a result, the version proposed in this section leads to obtaining two images that contain the informational content of the message: an image in which the quotient is hidden and one that contains the remainder.

Steganography using subtraction

Initially, the hiding of the quotient and the remainder in the carrier image was done using the subtraction operation.

Considering$c_{i,j,k}$, $c_q_{i,j,k}$and$c_r_{i,j,k}$the numerical values of the channel k (1-R, 2-G, 3-B)

of the pixel from the line i, column j, from the carrier image, from the image where the

quotient is hidden and the one where the remainder is hidden. Then:

$$c\_q_{i,j,k} = \begin{cases} c_{i,j,k} - q_{i,j,k}, & c_{i,j,k} \geq q_{i,j,k} \\ c_{i,j,k} + q_{i,j,k}, & c_{i,j,k} < q_{i,j,k} \end{cases} \quad (2.4)$$

$$c\_r_{i,j,k} = \begin{cases} c_{i,j,k} - r_{i,j,k}, & c_{i,j,k} \geq r_{i,j,k} \\ c_{i,j,k} + r_{i,j,k}, & c_{i,j,k} < r_{i,j,k} \end{cases} \quad (2.5)$$

The procedure of concealing the information is simple but has some drawbacks:

- The procedure requires decision-making structures to determine whether the value of the quotient and the remainder is smaller than the original value of the pixel
- After modification, the resulting image is darker than the original image because most pixel values are smaller than they were in the unaltered image;
- To reduce the effects of the second issue mentioned above, a logical matrix was created:

$$m\_l_{i,j,k} = \begin{cases} 1, & c_{i,j,k} \geq q_{i,j,k} \\ 0, & c_{i,j,k} < q_{i,j,k} \end{cases} \quad (2.6)$$

- Although this matrix helps accelerate the steganography phase, through appropriate MATLAB mechanisms, it requires supplementary memory to be stored.

Steganography using XOR

The disadvantages of using the subtraction method were reduced or even eliminated in a later version of the algorithm, where instead of the subtraction the XOR operation was used. Thus, keeping the notation from the previous section:

$$c\_q_{i,j,k} = c_{i,j,k} \oplus q_{i,j,k} \quad (2.7)$$

$$c\_r_{i,j,k} = c_{i,j,k} \oplus r_{i,j,k} \quad (2.8)$$

This way the transforming operation can be performed regardless of $c_{i,j,k} \geq q_{i,j,k}$ or $c_{i,j,k} < q_{i,j,k}$, which eliminates the need of using a decision-making structure.

To illustrate the way that using the XOR operation contributes to an improved behavior of the program from the viewpoint of the second disadvantage mentioned in the previous section, let's consider four examples:

1) $c_{1,1,1} = 128, q_{1,1,1} = 15$
   SUB: $c\_q_{1,1,1} = 113$
   XOR: $c\_q_{1,1,1} = 143$
2) $c_{1,1,1} = 128, q_{1,1,1} = 5$
   SUB: $c\_q_{1,1,1} = 123$
   XOR: $c\_q_{1,1,1} = 133$
3) $c_{1,1,1} = 127, q_{1,1,1} = 15$
   SUB: $c\_q_{1,1,1} = 112$
   XOR: $c\_q_{1,1,1} = 112$
4) $c_{1,1,1} = 127, q_{1,1,1} = 5$
   SUB: $c\_q_{1,1,1} = 122$
   XOR: $c\_q_{1,1,1} = 122$

According to values $c_{1,1,1}$ and $q_{1,1,1}$, the result is either the sum or the difference between the two values. The overall effect of this observation is that the resulting image will not be darker than the original. In contrast to the first steganography method presented, the result can be either the sum or the difference of the operands involved, which leads to a balanced change of carrier image.

A particular advantage of using the XOR function is the need to use lower resource, as will be shown in the tests section.

2.2 Actual Implementation

In order to test the theoretical aspects described in the previous paragraphs, all the algorithms were embedded into a fully functional application.

The general architecture of the application is depicted in Figure 2.1.

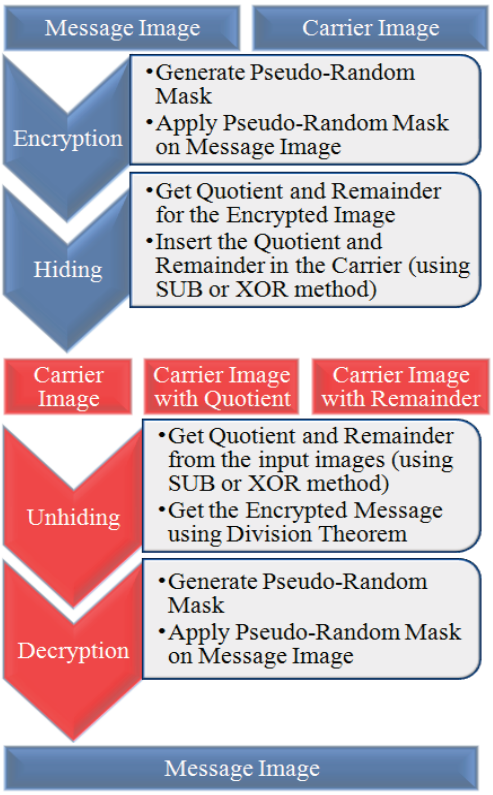

Figure 2.1. General Application Architecture

Message hiding

In this stage, two images are used as entries: the carrier and the message that will be embedded. The output will be represented by three images: the carrier, the carrier in which the quotient was embedded and the carrier in which the remainder was embedded.

In order to prove the flexibility, efficiency and precision of the application, the message is represented by an image that carries many different colors. This aims to validate the decomposition/composition algorithms which should not alter any of the colors. Figure 2.2 presents the message image

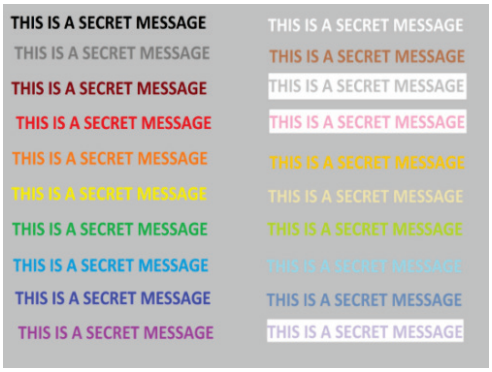

Figure 2.2. Message Image

The carrier image has no other particularities but the size. It is mandatory to have the same resolutions for the carrier and the message; however, there are no constraints as far as the graphic format is concerned. The unaltered carrier image is presented inFigure 2.3.

Figure 2.3. Carrier Image

Pseudo-random matrix generation

For the generation of the pseudo-random matrix, non-linear combinations of *sine* and *cosine* functions weighted by coefficients related to channel, line and column indexes were used.

Let $p\_m_{i,j,k}$ be the numerical value from the k channel (1-R, 2-G, 3-B) of the pixel located on row i and column$j$, in the pseudorandom matrix. Let keybe the encryption key and $\alpha_{i,j,k}$ a coefficient which depends on i, j and k. Then $p_m_{i,j,k}$can be computed as:

$$p\_m_{i,j,k} = \left(\alpha_{i,j,k} \cdot sin(i) \cdot cos(j)\right)\%256 \quad (2.9)$$

where % is the modulus operator.

Based on this expression, a mask that has the aspect of a grey noise is obtained. Regarding the pseudorandom matrix as an image in RGB space, one may find very useful to analyze the distribution of the pixel values, on each channel. The distribution for the red (R) channel is presented in Figure 2.4.

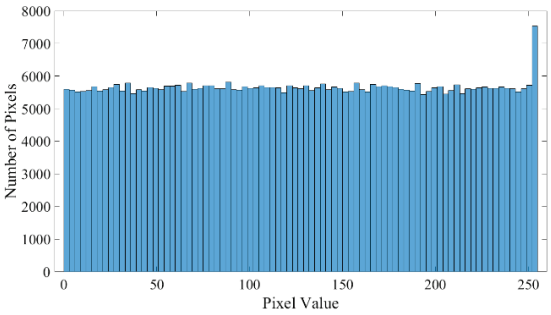

Figure 2.4. Pixel Value Histogram for R Channel in Pseudo-Random Generated Mask

A uniform distribution can be observed, except for the upper limit interval. This should not be considered a drawback since most of the applications dedicated to image processing intentionally disregard the first and the last 10% of the interval. This can be explained by the technical limitations of nowadays regular displays (TFT, LCD, LED) which cannot realistically render differences between very low or very high tones of color. However, this application is using the entire palette of values for encryption.

Applying the pseudo-random matrix over the message

XOR operation is applied between the mask image and the message image, pixel by pixel. The result is presented in Figure 2.5.

During the encryption stage the original message image (presented in Figure 2.2) completely changed its meaning, as one can see in Figure 2.5.

Hiding process

Once the encryption is finalized, the resulted image will be hidden into the carrier image. This will be achieved by using one of the methods previously described (subtraction or

XOR). The images resulted after this stage are presented in Figure 2.6 and Figure 2.7. One can observe little to no differences between these images and the original image presented in Figure 2.3.

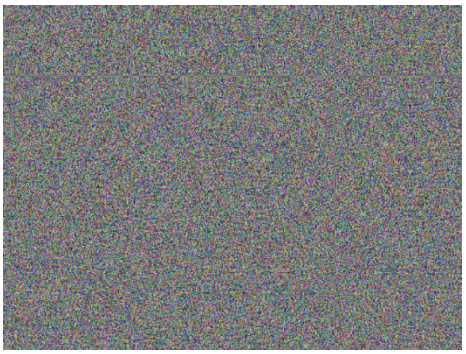

Figure 2.5. Encrypted Message

Figure 2.6. Carrier with Quotient (XOR Method)

Figure 2.7. Carrier with Remainder (XOR Method)

The two modified images and the original one will be sent to the destination where they will be used to extract and decrypt the message. In order to increase the safety of this method it is highly recommended to send the images on different communication channels or, at least, at different moments.

Message recovery

As one can see in the General Application Architecture (Figure 2.1), the receiver must follow the same steps from the encryption/hiding process, but in reverse order.

In order to lower the resources usage, a faster alternative for message extraction was implemented only for the subtraction method.

Instead of using the complementary relations of (2.4) and (2.5), the quotient and the remainder were extracted without relying on any decisional instructions. For instance:

$$q_{i,j,k} = \left|c_{i,j,k} - c\_q_{i,j,k}\right| \tag{2.10}$$

$$r_{i,j,k} = \left|c_{i,j,k} - c\_r_{i,j,k}\right| \tag{2.11}$$

This approach offers improved performances compared to the XOR based hiding method, as it will be presented later in the performance evaluation section.

Assuming that data transmission was error free, the extracted message at the receiver side will be identical with the original message from the sender. None of the stages of the proposed method modifies the entropy of the message. The extracted message is identical with the one presented in Figure 2.1.

3. Experimental Results

The application developed during this study was tested using PNG images with some of the most common resolutions used today. Figure 3.1presents the size of the pictures related to their resolution. This is an aspect closely related to the amount of RAM memory that will be used.

The main parameters monitored while running the individual algorithms and the overall behavior of the application were the CPU and RAM. Both tests (processor and memory) were conducted on a processing platform having the following components:

- Motherboard: Gigabyte B85-HD3
- Processor: Intel Pentium G3220
- RAM: 4GB, DDR3
- HDD: MAXTOR STM3500320AS ATA
- Video Card: NVIDIA GeForce GT 240
- Operating System: Windows 7 SP1, 64-bit
- MATLAB Version: R2016a.

MATLAB offers dedicated libraries and functions for monitoring hardware resources. Some of them were used for checking the amount of memory used during application execution.

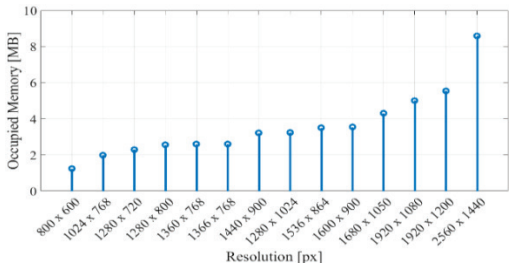

Figure 3.1. Memory vs. Resolution

The first step was to determine the amount of memory used by MATLAB prior to the execution of the steganography application. This value was set as reference. Next, the memory occupation was monitored after every instruction or function call. The maximum value was saved and, in the end, the reference value was subtracted. The results of the memory tests can be observed in Figure 3.2.

It can be observed that the evolution is similar with the one presented in Figure 3.1. Thus, the most important factor that influences the memory consumption is the size of the used pictures.

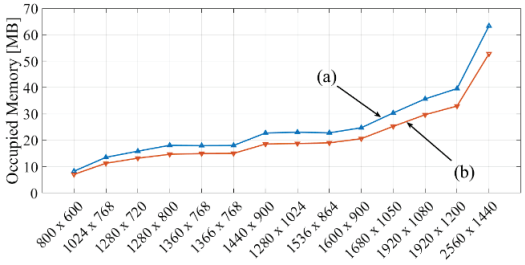

Figure 3.2. Memory Test Results vs. Resolution for Encryption and Hiding using (a) Subtraction Method and (b) XOR Method

One can also observe that the application requires 5-6 times more internal memory than the actual dimension of the images. There is a linear dependency between the required internal memory and the actual size of the images which can be explained by the co-existence of 6 variables that occupy same memory space as the initial image.

It is interesting to observe that XOR based hiding method requires about 10MB less memory in case of the biggest tested image. The difference in memory consumption is actually proportional with the resolution of the image; this offers a significant advantage in case of high quality (thus high resolution) images.

Memory consumption, in case of the reverse process, is lower and can be observed in Figure 3.3.

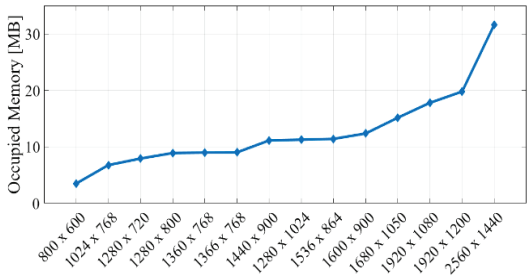

Figure 3.3. Memory Test Results vs. Resolution for Un-hiding and Decryption using Subtraction Method and XOR Method

The reason for this behavior derives from the extracting stage which requires a smaller number of matrixes associated with the images involved in the process. Moreover, it should be observed the fact that both methods that were used for obtaining the quotient and the remainder (subtraction and XOR) require the same amount of memory. This can be explained by subtracting procedure implemented as per (2.10) and (2.11)and without the auxiliary logical matrix as per (2.4) and (2.5).

MATLAB incorporates mechanisms that lower resources' usage while calling functions if the call is repetitive and consecutive. These mechanisms rely on Cache memory for storing variables or code segments. During the memory tests conducted over the described steganography application, reducing the influence of the Cache mechanisms was aimed.

In addition to memory consumption analysis, the execution time was monitored for images having similar resolutions as for the memory tests. For the same reasons previously presented, the use of MATLAB built-in functions was preferred.

In order to reduce the errors that may affect the results, the algorithm execution was made in strict conditions:

- The antivirus was uninstalled;
- The network connection was disabled;
- No user application was running except for MATLAB;
- For every image, the algorithm was run for 100 times; after each execution, the memory was cleared in order to disable the Cache mechanisms.

After running these tests, both maximum and minimum execution times were saved along with the medium execution time. The values are presented in Figure 3.4.

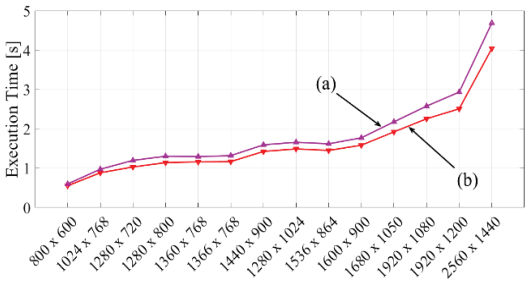

Figure 3.4. Execution Time Test Results vs. Resolution for Encryption and Hiding using (a) Subtraction Method and (b) XOR Method

Based on Figure 3.2 and Figure 3.4 several conclusions can be drawn:

- There is a certain similarity between execution time and memory demands and they are both highly related to image resolution.
- The XOR based method is slightly faster; the execution speed depends on image resolution. For a 2560x1440 picture, the XOR based method is 0.5 seconds faster.
- For most resolutions, a small execution time deviation was observed compared to the average of 100 tests presented in Figure 3.4.The execution time deviation was no higher than 0.35s, as presented in Figure 3.5.One can also observe that XOR based method produces a more uniform deviation as opposed to subtraction based method. It can be concluded that processing resources can be used without significant variations.

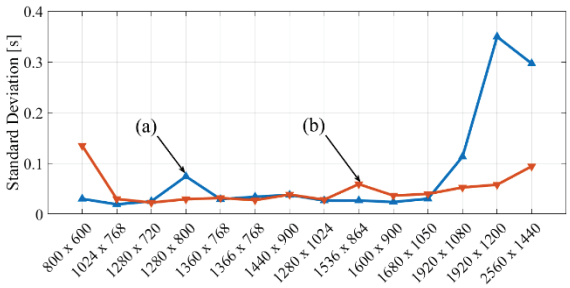

Figure 3.5. Standard Deviation of Execution Time for Encryption and Hiding using (a) Subtraction Method and (b) XOR Method

During the extraction procedure, the execution time dramatically decreases as presented in Figure 3.6.

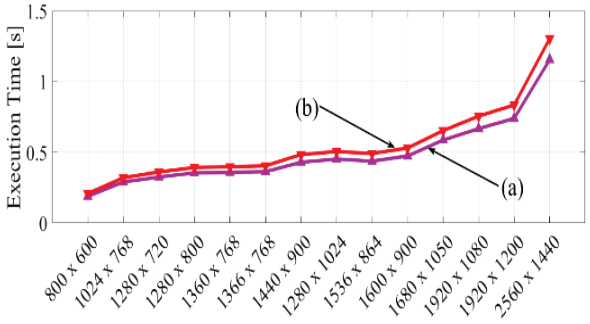

Figure 3.6. Execution Time Test Results vs. Resolution for Un-hiding and Decryption using (a) Subtraction Method and (b) XOR Method

The main reason resides in less disk writing operations. During the hiding stage, three images are generated while, in reconstruction stage, only the message image is written.

The overall performance of the application (hiding + reconstruction) was analyzed. The individual times for hiding and reconstruction stages were summed and the results are presented in Figure 3.7. The XOR based method exhibits a better time which, in case of high resolution images, may become significant.

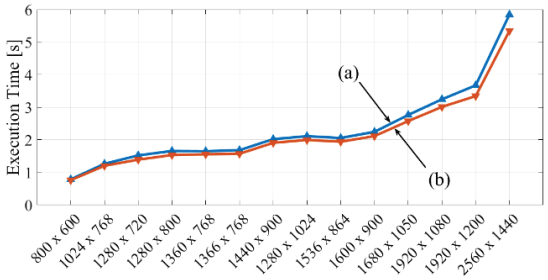

Figure 3.7. Execution Time Test Results vs Resolution for Entire Process using (a) Subtraction Method and (b) XOR Method

4. Conclusions and Future Work

In this paper a simple and yet very powerful steganography method is presented. This method is based on steganography improved by encryption. Unlike other steganography applications in which encryption is involved, it is not necessary to send the encryption key as it is. In exchange, 3 extremely look alike images are sent at different times or by different means from the sender to receiver. Should anyone intercept any of the images, he would not be able to recover the hidden message.

The target of this study was to develop a simple method for securely sending information with low hardware and software requirements. Of

course, a strong security was aimed, as close as possible to enhanced encryption.

During the encryption stage of the presented application two methods were tested and the results were similar. However, for big images, the XOR based method behaved a little better.

REFERENCES

1. Abboud, G., Marean, J. & Yampolskiy, R. (2010). Steganography and Visual Cryptography, In *Computer Forensics, IEEE Fifth International Workshop on Systematic Approaches to Digital Forensic Engineering*, pp. 25-32.

2. Djebbar, F., Ayad, B.& Meraim, K. (2012). Comparative study of digital audio steganography techniques, *EURASIP Journal on audio, speech and music processing*, 2012(1), pp. 1-16.

3. Doshi, R., Jain, P. & Gupta, L. (2012). Steganography and Its Applications in Security, *International Journal of Modern Engineering Research*, 2(6), pp. 4634-4638.

4. Elminaam, D., Kader, H. & Hadhoud, M. (2009). Performance Evaluation of Symmetric Encryption Algorithms, *Communications of the IBIMA*, 8, pp. 58-64.

5. Gleason, N. (1987). *Fun with codes and ciphers workbook*, Dover Publications Inc.

6. Iordache, D., Pribeanu, C., & Balog, A. (2012). Influence of Specific AR Capabilities on the Learning Effectiveness and Efficiency, *Studies in Informatics and Control*, 21(3), pp. 233-240.

7. Jayaram, P., Ranganatha, H., Anupama, H. (2011). Information hiding using audio steganography – a survey, *International Journal of Multimedia & Its Applications (IJMA)*, 3(3), pp. 86-96.

8. Mazurczyk, W. & Kotulski, Z. (2006). New security and control protocol for VoIP based on steganography and digital watermarking, *Annales UMCS, sectio AI – Informatica*, *5*, pp. 417-426.

9. Mazurczyk, W., Wendzel, S., Villares, I. & Szczypiorski, K. (2016). On importance of steganographic cost for network steganography, *Security and Communication Networks*, 9(8), pp. 781–790.

10. Mona, M., Chitra, S., Gayathri, V. (2014). A survey on various encryption and decryption algorithms, *Singapore Journal of Scientific Research*, *6*(6), pp. 289-300.

11. Nissar, A. & Mir, A. (2010). Classification of steganalysis techniques: A study, *Journal of Digital Signal Processing*, 20(6), pp. 1758-1770.

12. Saeed, M. (2013). A new technique based on chaotic steganography and encryption text in DCT domain for color image, *Journal of Engineering Science and Technology*, 8(5), pp.508–520.

13. Saha, B. & Sharma, S. (2012). Steganographic Techniques of Data Hiding using Digital Images, *Defence Science Journal, 62*(1), pp.11-18.

14. Saleh, S. (2013). A secure data communication system using cryptography and steganography, *International Journal of Computer Networks & Communications (IJCNC)*, 5(3), pp.125-137.

15. Singh, S. & Attri, V. (2015). State-of-the-art Review on Steganographic Techniques, *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 8(7), pp.161-170.

16. Subhedar, M. & Mankar, V. (2014). Current status and key issues in image steganography: A survey, *Computer Science Review*, 13–14, pp.95–113.

17. Thangadurai, K. & Devi, G. (2014). An analysis of LSB based image steganography techniques, In *IEEEInternational Conference on Computer Communication and Informatics*, pp.1-6.

18. Tripathi, R. & Agrawal, S. (2014). Comparative Study of Symmetric and Asymmetric Cryptography Techniques, *International Journal of Advance*

Foundation and Research in Computer, 1(6), pp.68-76.

19. Tseng, H. & Leng, H. (2013). A Steganographic Method Based on Pixel-Value Differencing and the Perfect Square Number, *Journal of Applied Mathematics*, *2013*, ID 189706, online.

20. Wang, K., Lu, Z. & Hu, Y. (2013). A high capacity lossless data hiding scheme for JPEG images, *Journal of Systems and Software*, *86*(7), pp.1965-1975.

21. Warkentin, M., Bekkering, E. & Schmidt, M. (2008). Steganography: Forensic, Security, and Legal Issues, *Journal of Digital Forensics, Security and Law*, *3*(2), article 2, online.

22. Wendzel, S., Mazurczyk, W., Caviglione, L. & Meier, M. (2014). Hidden and Uncontrolled - On the Emergence of Network Steganographic Threats, In *Securing Electronic Business Processes: Highlights of the Information Security Solutions Europe Conference*, pp. 1-11.

23. Zhang, J., Cox, I. & Doerr, G. (2007). Steganalysis for LSB Matching in Images with High-frequency Noise,In *IEEE 9th Workshop on Multimedia Signal Processing*, pp. 385-388.