

Combining Steganography and Cryptography: New Directions

Khalil Challita and Hikmat Farhat
Computer Science Department
Notre Dame University - Louaize, Lebanon
kchallita,hfarhat@ndu.edu.lb

Abstract

Our main goal in this paper is to give new insights and directions on how to improve existing methods of hiding secret messages, possibly by combining steganography and cryptography. We start by describing the main existing methods and techniques in steganography that allow us to hide the existence of a message, together with the mostly used steganalysis techniques to counter them. We then illustrate two different approaches that help us achieve a higher level of secrecy and security, together with their limitations. The first method is about combining steganography and cryptography in such a way to make it harder for a steganalyst to retrieve the plaintext of a secret message from a stego-object if cryptanalysis were not used. The second method does not use any cryptographic techniques at all and relies solely on steganographic ones.

Keywords : *Steganography, steganalysis, cover object, stego-object, cryptography.*

1. Introduction

Since the ancient times people have been interested in hiding secret messages. Both cryptography and steganography achieve this aim, but using different strategies as we next explain. We have stories from the antiquity on how the

Greeks received warning of Xerxes hostile intentions from a message underneath the wax of a writing tablet; about the use of invisible ink; about ancient Chinese wrote messages on fine silk, which was then crunched into a tiny ball and covered by wax that the messenger swallowed; or about a Roman general who shaved a slaves head and tattooed a message on it. After the hair grew back, the slave was sent to deliver the now-hidden message [13].

Steganography is concerned with sending a secret message while hiding its existence. The word *steganography* is derived from the Greek words *steganos*, meaning 'covered', and *graphein*, meaning 'to write'.

On the other hand, cryptography is not concerned with hiding the existence of a message, but rather its meaning by a process called *encryption*. The word *cryptography* is derived from the Greek word *kryptos*, meaning 'hidden'.

Several ciphers can be found in the literature such as the Caesar cipher, or the cipher of Mary Queen of Scots that was successfully broken by the linguist and cryptanalyst Phelippes, which led to the execution of Mary Queen of Scots in 1587. One may read [27] for more historical notes on the art of exchanging secret messages. A more scientific and formal approach can be found in [26, 21].

Cryptanalysis is the science that tries to defeat cryptography. Many ciphers were shown to be vulnerable to cryptanalytic attacks such as *fre-*

quency attack used to break mono-alphabetic ciphers.

Very often, a message is encrypted before being hidden in a message in order to achieve a better level of secrecy (which provides a basic example on how to combine cryptography and steganography).

Steganography embeds the secret message in a harmless looking cover, such as a digital image file [10, 11]. The need for steganography is obvious but what is less obvious is the need for more research in the field. Simple techniques are easily detectable and there is a whole field of defeating steganographic techniques called steganalysis [12]. As it is always the case, advances in steganography are usually countered by advances in steganalysis which makes it a constantly evolving field. Since most steganographic system use digital images as cover, the whole field has borrowed methods and ideas from the closely related fields of watermarking and fingerprinting which also manipulate digital audio and video, for the purpose of copyright. Even though, in principle, many aspect of images can be manipulated, in reality most stego systems aim for the preservation of the visual integrity of the image. Early stego systems goals was to make changes not detectable by the human eye [24]. This feature is not enough because statistical methods can detect the changes in the image even if it is not visible. Image compression also plays a role in steganography because it was found that on many occasions the result depend on the compression scheme used.

Steganographers struggle to find more efficient methods to embed a secret message in a cover object, only to be defeated by techniques derived by steganalysts. Anderson and Petitcolas [1] explore the theoretical limits of steganography, and show how much hard it is to get a scheme that gives unconditional covertness, in the sense that the one-time pad provides unconditional secrecy for cryptography. More recently, Hopper et al. [9] used cryptographic and complexity-theoretic proof techniques to show that the exist-

tence of one-way functions and access to a channel *oracle* are both necessary and sufficient conditions for the existence of secure steganography relative to any channel. They constructed a steganographic protocol that is provably secure and has nearly optimal bandwidth under these conditions when compared with known provably secure constructions, which is the first known example of a general provably secure steganographic protocol.

Despite this fact, we believe that a simple modification of existing steganographic protocols may enhance drastically the security of a steganographic system (from a practical point of view). The below ideas do not work in the case of watermarking or fingerprinting where we are required to embed the secret message in a cover object.

Our main 2 ideas are the following: (1) Embed the secret message in the cover object without modifying it; and (2) Embed the secret message in more than one cover object. In the latter example we need to use cryptography in order to extract the message from the stego-object. We already proposed such a scheme in [6], where the cover media remains intact and a separate file is sent to the receiver that allow him to retrieve the secret message from the stego-media.

To our knowledge no such methods are available in the literature. Their advantages and drawbacks will be discussed in more details in Sections 4 and 5.

This paper is divided as follows. In Section 2 we briefly talk about steganography and the most widely used steganographic techniques to hide messages in digital images using the least significant bit technique. In Section 3 we show how steganalysis can defeat all the methods discussed in the previous section. We propose in Section 4 a new steganographic protocol to embed a secret message. In Section 5 we explain how to make use of cryptographic techniques in order to achieve a higher level of security against well-known steganalytic attacks.

2. Steganography

Three different aspects in information-hiding systems contend with each other: capacity, security, and robustness [3]. Capacity refers to the amount of information that can be hidden in the cover medium, security to an eavesdroppers inability to detect hidden information, and robustness to the amount of modification the stego medium can withstand before an adversary can destroy hidden information.

Information hiding generally relates to both watermarking and steganography. A watermarking systems primary goal is to achieve a high level of robustness. Steganography, on the other hand, strives for high security and capacity.

Modern steganography attempts to be detectable only if secret information is knownnamely, a secret key. This is similar to Kerckhoffs Principle in cryptography [17], which holds that a cryptographic systems security should rely solely on the key material.

Classical steganography concerns itself with ways of embedding a secret message. The embedding usually uses a secret key as we just explained. The terminology researchers use was agreed at the First International Workshop on Information Hiding [22]. We use a cover object (e.g. image, text, etc.) to embed a secret message. The resulting object is called a stego object (e.g. stego-image).

2.1. Steganographic system

A steganographic system is a mechanism that embeds a secret message m in a cover object c using a secret shared k . The result is a stego object s which carries the message m . Formally we defined the stegosystem as a pair of mappings (F, G) with F serves as the embedding function and G as the extraction function.

$$s = F(c, m, k)$$

$$m = G(s, k)$$

If M is the set of all possible messages then the embedding capacity of the stegosystem is $\log_2 M$

bits. The embedding efficiency is defined as

$$e = \frac{\log_2 M}{d(c, s)}$$

The set of all cover objects C is sampled using a probability distribution $P(c)$ with $c \in C$, giving the probability of selecting a cover object c . If the key and message are selected randomly then the Kullback-Leibler distance

$$KL(P|Q) = \sum_{c \in C} P(c) \log \frac{P(c)}{Q(c)}$$

gives a measure of the security of the stegosystem. The three quantifiers defined above: capacity, efficiency and security are the most important requirements that must be satisfied for any steganographic system. In reality, determining the best embedding function from a cover distribution is an NP-hard problem[2]. In addition, combining cryptography and steganography adds another layer of security [18]. Before embedding a secret message using steganography, the message is first encrypted. The receiver then should have both the stego-key in order to retrieve the encrypted information and the cryptographic key in order to decrypt it.

2.2. Image as a cover media

Steganographic systems for the JPEG format seem more interesting than others (e.g. BMP) because the systems operate in a transform space and are not affected by visual attacks [29].

It would be helpful to review the encoding scheme of some image formats. The GIF format is a simple encoding of the RGB colors for each pixel using an 8-bit value. The color is not specified directly, rather the index into a 256 element array is selected. After the encoding the whole image is compressed using LZW lossless technique. In the JPEG format, first each color is converted from RGB format to $YC_B C_R$ where the luma (Y) component representing the brightness of the pixel is treated differently than the chroma

components ($C_B C_R$) which represent color difference. The difference of treatment is due to the fact that the human eye discerns changes in the brightness much more than color changes. Doing such a conversion allows greater compression without a significant effect on perceptual image quality. One can achieve higher compression rate this way because the brightness information, which is more important to the eventual perceptual quality of the image, is confined to a single channel. Once this is done for each component the discrete cosine transform (DCT) is computed to transform 8×8 pixel blocks of the image into DCT coefficients. The coefficients are computed as:

$$F(u, v) = \sum_{x=0}^7 \sum_{y=0}^7 G(x, y) \cos \frac{(2x+1)\pi u}{16} \cos \frac{(2y+1)\pi v}{16}$$

After the DCT is completed the coefficients $F(u, v)$ are quantized using elements from a table.

Many different steganographic methods have been proposed during the last few years. Most of them can be seen as substitution systems (which are based on the Least Significant Bit (LSB) encoding technique). Such methods try to substitute redundant parts of a signal with a secret message. Their main disadvantage is the relative weakness against cover modifications. Other more robust techniques fall within the *transform domain* where secret information is embedded in the transform space of the signal such as the frequency domain. We next describe some of these methods.

2.2.1 LSB

The most popular method for steganography is the Least Significant Bit (LSB) encoding [5]. Using any digital image, LSB replaces the least significant bits of each byte by the hidden message bits. Depending on the image format the resulting changes made by the least-significant bits are visually detectable or not [18]. For example, the GIF format is susceptible to visual attacks while

JPEG being in the frequency domain is less prone to such attacks.

2.2.2 Jsteg

The first publicly available steganographic system was JSteg [25]. Its algorithm replaces the least-significant bit of the DCT coefficients with the message data. Because JSteg does not require a key, an attacker knowing the existence of the message will be able to recover it. Due to its simplicity LSB embedding of JSteg is the most common method implemented today. However, many steganalysis techniques have been developed to counter JSteg [33]. One can show that there is JPEG steganographic limit with respect to the current steganalysis methods [7, 30, 29].

2.2.3 OutGuess

Created by Niels Provos, OutGuess is a steganographic system that improves the encoding step by using a pseudo-random number generator to select DCT coefficients at random. The least-significant bit of a selected DCT coefficient is replaced with encrypted message data. The χ^2 -test for JSteg does not detect data that is randomly distributed across the redundant data and, for that reason, it cannot find steganographic content hidden by OutGuess.

Other stegosystems include the Transform domain method [32, 15] which works in similar way as watermarking uses by using a large area of the cover image to hide messages which makes these method robust against attacks. The main disadvantage of such methods, however, is that one cannot send large messages because there is a trade-off between the size of the message and robustness against attack. What concerns us most in this paper is the fact that almost all steganographic methods applied on digital images change the structure and statistics of the images in when a hidden message is embedded in them.

3. Steganalysis

Steganalysis is the art of detecting messages hidden by stegosystems [14]. Steganalysis can be performed by examining the statistics of the cover image or by visual inspection.

There are different types of attacks against such systems [5, 20]. In one such attack, the *Known cover attack*, the original cover object and the stego-object are available for analysis. The idea in this attack is to compare the original media with the stego-media and note the differences. These differences may lead to the emergence of patterns that would constitute a signature of a known steganographic technique. A different approach to steganalysis is to model images using a feature vector as in blind steganalysis and capture the relationship between the change in the feature vector to the change rate using regression [19]. Yet another approach is based on the Maximum Likelihood principle [16]. The concept of steganographic security, in the statistical sense, has been formalized by Cachin [2] by using an information-theoretic model for steganography. In this model the action of detecting hidden messages is equivalent to the task of hypothesis testing. In a perfectly secure stegosystem the eavesdropper has no information at all about the presence of a hidden message.

We next show how steganalysis can defeat the steganographic methods described in the previous section. The attacks against JStet and OutGuess use statistical properties of the stego-image.

3.1. JSteg

Andreas Westfeld and Andreas Pfitzmann noticed that steganographic systems that change least-significant bits sequentially cause distortions detectable by steganalysis [29]. They used a χ^2 -test to determine whether the observed frequency distribution γ_i in an image matches a distribution γ_i^* that shows distortion from embedding hidden data. See [23] for more details.

Westfeld and Pfitzmann observed that for a

given image, the embedding of high-entropy data (often due to encryption) changed the histogram of color frequencies in a predictable way. In the simple case, the embedding step changes the least-significant bit of colors in an image. The colors are addressed by their indices i in the color table; we refer to their respective frequencies before and after embedding as n_i and n_i^* . Given uniformly distributed message bits, if $n_{2i} > n_{2i+1}$, then pixels with color $2i$ are changed more frequently to color $2i + 1$ than pixels with color $2i + 1$ are changed to color $2i$. As a result, the following relation is likely to hold: $|n_{2i} - n_{2i-1}| \geq |n_{2i}^* - n_{2i-1}^*|$.

In other words, embedding uniformly distributed message bits reduces the frequency difference between adjacent colors. The same is true in the JPEG data format. Instead of measuring color frequencies, we observe differences in the DCT coefficients frequency.

3.2. OutGuess

Using an extended χ^2 -test as explained in [23], one can detect pseudo-randomly distributed hidden data. Given a constant sample size, take samples at the beginning of the image and increase the sample position by 1 percent for every χ^2 calculation. Take the sum of the probability of embedding for all samples. If the sum is greater than the detection threshold, the test indicates that an image contains a hidden message. To find an appropriate sample size, one can select an expected distribution for the extended χ^2 -test that should cause a negative test result. Instead of calculating the arithmetic mean of coefficients and their adjacent ones, we take the arithmetic mean of two unrelated coefficients:

$$\gamma_i^* = \frac{n_{2i} + n_{2i-1}}{2}$$

3.3. Variations of the LSB method

It is worth noting that different schemes based on LSB steganography have been proposed. Two of them are known as LSB-I and LSB-II. But both these methods were shown to be weak against statistical attacks. Yu et al. [31] describe an effective method to detect what is called *Multiple LSB steganography*, which encompasses the detection of techniques that use both LSB-I and LSB-II we next briefly describe.

Let $b_r b_{r-1} \dots b_L \dots b_1$ denote the binary representation of the pixel value of an image, where r is the number of bits to represent image pixel value, b_r , b_L and b_1 are the Most Significant bit (MSB), L th-rightmost bit and Least significant bit (LSB), respectively. There are two distinct embedding paradigms for Multiple LSB Steganography. In the first embedding method, bits are embedded in the cover by selecting pixels and replacing all the L LSBs of each pixel with L corresponding bits of the payload. We call this as LSB-I embedding. As an alternative method of using multiple LSBs, bits can be embedded in the cover by selecting pixels and replacing only the LSBs of each pixel with a corresponding bit of the payload, then repeating with a new selection of pixels of which the next LSB is used. The iteration will stop till L th LSB. We call this as LSB-II embedding.

Yu et al. [31] implemented and tested their method for LSB-I and LSB-II, using different embedding rates (from 10% to 100%) and different values for L . They proved that their new approach performs better than existing ones.

3.4. Visual LSB detection

Many steganographic systems embed hidden messages inside the least significant bit layers of colour natural images. The presence of these messages can be difficult to detect by using statistical steganalysis. Experiments conducted by Watters et al. [28] showed that visual steganalysis by humans is more successful than statistical ones to detect least-significant bit steganography. Their

study used simple, single-layer embedding across the eight different color layers individually, to determine if there was any relationship between discriminability and the bit layer used. Basic substitution approaches to natural image steganography encode secret information by replacing insignificant parts of the original image with parts of the secret message. The embedding process consists of choosing a subset of cover elements and performing a substitution operation on them. In this operation, the LSB of the original images is replaced with a bit of the secret message. This algorithm can be extended by replacing more than one bit of the original image or by storing two message bits in each cover byte, for instance. Another extension involves distributing the bits over the image and not substituting each LSB bit, but every second, third, or by following a pseudorandom pattern.

Watters et al. [28] showed (empirically) that bit layers 13 appear to be highly resistant to visual steganalysis, whereas human being can detect hidden messages in the higher layers 4-8.

4. Combining steganography and cryptography

Our first suggestion in this paper is to improve steganographic techniques by combining them to cryptographic ones in a new way that is, as far as we know, not available in the literature. Indeed, most of the techniques that combine cryptography and steganography consist in encrypting the secret message before hiding its existence in a cover object.

As for us, we suggest to use totally different schemes such as the one suggested in [6].

Our idea is the following: both the sender and the recipient agree on a cover image to send a secret message. The protocol does not modify the cover image, rather it determines the bits of the secret message that match the ones in the cover image and stores their different locations (i.e. in the cover image) in a vector. This vector is then

sent (possibly encrypted using classical cryptography) to the recipient in a way we describe below.

A steganalyst in this case may intercept a vector of bits that is possibly encrypted, without knowing to which cover image it corresponds. So to defeat our scheme, a steganalyst has to intercept the secret message sent to the recipient and must know which cover image it corresponds to.

4.1. Static parsing steganography

SPS consists of 2 main steps.

1. A cover image (that both the sender and receiver share), and the secret message to be sent are converted into bits. Let us denote the output files by Image1 and Secret1, respectively.
2. In this step, we encode the secret message Secret1 based on Image1. The idea is based on the problem of finding the longest common substring of two strings using a generalized suffix tree, which can be done in linear time [8].

The algorithm uses a divide-and-conquer strategy and works as follows.

It starts with the whole bits of Secret1 and tries to find a match of all the bits of Secret1 in Image1. If this is the case, it stores the indexes of the start and end bits of Secret1 that occur within Image1 in an output file Output1. If not, the algorithm recursively tries to find a match of the first and second halves of Secret1 in Image1. It keeps repeating the process until all the bits of Secret1 have been matched with some bits of Image1.

We next give a pseudo-code on how the algorithm works.

Denote by LCS(S1, S2) the algorithm that finds the longest common subsequence of S1 that appears in S2, and returns true if the whole of S1 occurs in S2. We allow this modification of the

algorithm (i.e. LCS) in order to simplify the implementation of SPS we next describe.

```
SPS(secretMessage , coverImage);
    if LCS(secretMessage , coverImage) is true ,
        then
            store the positions of the indexes
            of the start and end bits of Secret
            that occur within Image the output
            file Output ,
        else
            SPS(LeftPart-secretMessage , coverImage)
            SPS(RightPart-secretMessage , coverImage)
    return Output ,
```

Example 1 Assume that the cover image is 100010101111 and that the secret message is 1010. Then the output file would be 58, since 1010 occurs in 100010101111 starting from index 5 (assuming that the first index is numbered 1).

Example 2 Assume that the cover image is Image = 110101001011000 and that the secret message is Secret = 11111010. This encoding requires 4 recursive calls of SPS. Indeed, the first call returns false since Secret does not appear in Image. After the first recursive call, we evaluate SPS(1111,110101001011000) and SPS(1010,110101001011000). The former requires 2 additional recursive calls: SPS(11,110101001011000), and the latter none, since 1010 appears in Image from index 4 to 7. The call SPS(11,110101001011000) returns 12. So the output file contains 121247.

4.2. Time complexity

The running time of SPS can be determined by the recurrence relation $T(n) = 2T(n/2) + O(n)$. This is because the recursive call divides the problem into 2 equal subproblems, and the local work which is determined by LCS requires $O(n)$ time. The solution of this recurrence is $\Theta(n \log n)$ [4].

5. Design of a new steganographic protocol

Our main contribution in this paper is a protocol that hides a secret message in more than one cover object. Steganography has always focused on developing techniques for hiding a secret message in one carrier, but as far as we know, never in more than one. Indeed, this would be possible since the aim of steganography is to hide the *existence of a message* as we said, no matter what the method is. Definitely, our method does not apply to watermarking or fingerprinting techniques, since we are forced here to store a message in one specific cover object.

The trade-off here is between a higher level of security and more stego-objects to handle.

Indeed, we can hide a message in $k > 0$ cover images. The greater the value of k , the harder it will be for a steganalyst to defeat our scheme. Moreover, hiding the secret message in k cover objects require the use of an algorithm (i.e. secret key) that both the sender and recipient should share.

Overall, the security of this new protocol relies on the (variable) number of stego-objects, and a secret key.

The idea of our algorithm **Multiple-Cover-Objects** (MCO) is given below:

```
MCO(M, k, cover objects, technique);  
  Hide the secret message M in k cover  
  objects using technique, and a secret  
  key (e.g. a mathematical function)  
  that distributes the bits of M over  
  all the cover objects.  
return the k stego-objects,
```

Example 3 Assume that we need to store the secret message $M = 110011$ in $k = 4$ cover images C_1, \dots, C_4 . We could for example use the well known LSB technique to store M as follows (i.e. the secret key):

Store the first bit in C_1 , the second in C_2 , and so on until $C_{k=4}$; and then go back to C_1 ...

For example, the first and fifth bits of M would be stored in C_1 , and only bit 0 in C_4 .

A cryptanalyst will have to determine 3 key elements in order to *completely* uncover the secret message:

1. The number of cover objects used (i.e. k).
2. All the stego-objects used to hide M .
3. The algorithm (i.e. secret key) used to hide M in the cover-objects.

Note that we did not include the *technique* used because it is usually well-known to the steganalyst.

We believe that this protocol achieves a very high level of confidentiality even if we use a relatively weak technique (such as LSB) to hide a message in several cover objects. This is because a steganalyst will not be able to recover M even if he/she suspects the presence of hidden information in some of the stego-objects used to hide M .

6. Conclusion

We described in this paper very well known steganographic techniques used to hide secret messages in stego objects that use the least significant bit method, together with known methods that stem from steganalysis on how to counter them.

Our main contribution in this paper can be found in Section 4 and Section 5, where we provide new steganographic protocols to hide secret messages. The first one does not modify the cover object and consists in sending a (possibly encrypted) vector that contains the different positions of the cover object that allow us to reconstruct the secret message from it. In this case, both the sender and the recipient should share a secret algorithm (or a key) on how to retrieve the secret message, given the cover object and the (secretly sent) vector.

The originality of this paper lies in the second

protocol that we called Multiple-Cover-Objects, where we suggest using more than one cover object to hide a secret message. Indeed, in order to recover the secret message, a steganalyst has to determine all the stego-objects and unravel the algorithm used to hide into them the secret message. As a future work we will implement and test this method for different number of cover images.

References

- [1] R. Anderson, F. Petitcolas. On the limits of steganography. In *IEEE Journal on Selected Areas in Communications*, volume 16, number 4, 1998.
- [2] C. Cachin. An information-theoretic model for steganography. In *Information Hiding*, volume 1525 of *Lecture Notes in Computer Science*, pages 306–318. Springer Berlin / Heidelberg, 1998.
- [3] B. Chen and G.W. Wornell. Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding. In *IEEE Trans. Information Theory*, volume 47, no. 4, pages 1423–1443, 2001.
- [4] Cormen, Leiserson, Rivest, and Stein. Introduction to algorithms. *The MIT press, second edition*, 2001.
- [5] B. Dunbar. A detailed look at steganographic techniques and their use in an open-systems environment. *Sans InfoSec Reading Room*, 2002.
- [6] H. Farhat, K. Challita, J. Zalaket. Static parsing steganography. In *Proceedings of Digital Information and Communication Technology and Its Applications*, (DICTAP 2011), pages 485–492, 2011.
- [7] J. Fridrich, T. Pevný, and J. Kodovský. Statistically undetectable jpeg steganography: dead ends challenges, and opportunities. In *Proceedings of the 9th workshop on Multimedia & security*, pages 3–14, New York, NY, USA, 2007. ACM.
- [8] D. Gusfield. Algorithms on strings, trees, and sequences. *Cambridge university press*, 1997.
- [9] N. Hopper and L. Von Ahn and J. Langford. Provably Secure Steganography. *IEEE Transactions on Computers*, volume 58, number 5, 2009.
- [10] W. Huaiqing and S. Wang. Cyber warfare: Steganography vs. steganalysis. *Communications of the ACM*, 47(10):76–82, 2004.
- [11] M. G. J. Fridrich. Practical steganalysis of digital images - state of the art. *Security and Watermarking of Multimedia Contents IV*, 4675:1–13, 2002.
- [12] N. Johnson and S. Jajodia. Steganalysis of images created using current steganography software. *Workshop on Information Hiding*, 1998.
- [13] N. Johnson and S. Jajodia. Exploring Steganography: Seeing the Unseen. *Computer*, vol. 31, no. 2, pp. 2634, 1998.
- [14] N. Johnson and S. Jajodia. Steganalysis: The investigation of hidden information. *Proc. Of the 1998 IEEE Information Technology Conference*, 1998.
- [15] Katzenbeisser and Petitcolas. Information hiding: Techniques for steganography and watermarking. *Artech House*, 2000.
- [16] A. D. Ker. A fusion of maximum likelihood and structural steganalysis. In *Proceedings of the 9th international conference on Information hiding*, IH'07, pages 204–219, Berlin, Heidelberg, 2007. Springer-Verlag.

- [17] A. Kerckhoffs. La Cryptographie Militaire (Military Cryptography). In *J. Sciences Militaires (J. Military Science, in French)*, 1883.
- [18] R. Krenn. Steganography and steganalysis. *Whitepaper*, 2004.
- [19] K. Lee, A. Westfeld, and S. Lee. Generalised category attack: improving histogram-based attack on jpeg lsb embedding. In *Proceedings of the 9th international conference on Information hiding, IH'07*, pages 378–391, Berlin, Heidelberg, 2007. Springer-Verlag.
- [20] E. T. Lin and E. J. Delp. A review of data hiding in digital images. *Proceedings of the Image Processing, Image Quality, Image Capture Systems Conference*, 1999.
- [21] Wenbo Mao. Modern cryptography: Theory and practice. *Prentic Hall, 1st edition*, 2003.
- [22] B. Pfitzmann. Information hiding terminology. in *Information Hiding, Springer Lecture Notes in Computer Science*, vol. 1174, pp. 347-350, 1996.
- [23] N. Provos and P. Honeyman. Hide and Seek: An Introduction to Steganography. *IEEE Computer Society, IEEE Security and Privacy*, 2003.
- [24] M. Shirali-Shahreza and S. Shirali-Shahreza. Collage steganography. In *Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2006)*, pages 316–321, Honolulu, HI, USA, July 2006.
- [25] D. Upham. Steganographic algorithm JSteg. <http://zooid.org/paul/crypto/jsteg>.
- [26] Bruce Schneier. Applied Cryptography: Protocols, Algorithms, and Source Code in C. In *Wiley, 2nd Edition*, 1994.
- [27] Simon Singh. The code book. In *Fourth Estate, London*, 1999.
- [28] P. Watters and F. Martin and H. Steffen Stripf. Visual Detection of LSB-Encoded Natural Image Steganography. In *ACM Transactions on Applied Perception*, Vol. 5, No. 1, Article 5, 2008.
- [29] A. Westfeld and A. Pfitzmann. Attacks on steganographic systems. In *Proceedings of the Third International Workshop on Information Hiding, IH '99*, pages 61–76, London, UK, 1999. Springer-Verlag.
- [30] X. Yu, Y. Wang, and T. Tan. On estimation of secret message length in jsteg-like steganography. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4 - Volume 04, ICPR '04*, pages 673–676, Washington, DC, USA, 2004. IEEE Computer Society.
- [31] X. Yu, N. Babaguchi. A Fast and Effective Method to Detect Multiple Least Significant Bits Steganography. In *SAC 2008, ACM*, 2008.
- [32] M. J. Z. Zahedi Kermani. A robust steganography algorithm based on texture similarity using gabor filter. *IEEE 5th International Symposium on Signal Processing and Information Technology*, 2005.
- [33] T. Zhang and X. Ping. A fast and effective steganalytic technique against jsteg-like algorithms. In *Proceedings of the 2003 ACM symposium on Applied computing, SAC '03*, pages 307–311, New York, NY, USA, 2003. ACM.