*Journal of Information Hiding and Multimedia Signal Processing* ©2020 ISSN 2073-4212
**Ubiquitous International** Volume **11**, Number **1**, March **2020**

# Side Attacks on Stegosystems Executing Message Encryption Previous to Embedding

Valery Korzhik, Cuong Nguyen, Ivan Fedyanin

Department of Protected Communication Systems
The Bonch-Bruevich Saint-Petersburg State University of Telecommunications, St. Petersburg, Russia
22 Prospekt Bolshevikov, St. Petersburg, Russia, 193232
val-korzhik@yandex.ru, cuong0111@gmail.com, ivan.a.fedyanin@gmail.com

Guillermo Morales-Luna

Computer Science, CINVESTAV-IPN, Mexico City, Mexico
CINVESTAV-IPN, Mexico City, Mexico
Av. IPN No. 2508 Col. San Pedro Zacatenco Mexico City, Mexico
gmorales@cs.cinvestav.mx

Received April 2018; revised December 2018

Abstract. *There are introduced two new steganalytic methods not depending on the statistics of the cover objects, namely side attacks stegosystems. The first one assumes that the plaintext, encrypted before embedding, is partly known by the attacker. In this case, the stegosystems detection is based on the calculation of mutual information between message and extracted encrypted data. For this calculation, a notion of the k-nearest neighbor distance is applied. The second method is applied to HUGO, one of the most efficient steganographic algorithms. In this case the stegosystems detection is based on a verification of the NIST tests to the extracted encrypted messages. Moreover, we show that the problem to find a submatrix of the embedding matrix determining a trellis code structure in the HUGO algorithm provides a search of the stegokey by the proposed method.*
**Keywords:** Stegosystem, mutual information, entropy, relative entropy, encryption, decryption, NIST tests

1. **Introduction.** Steganalysis (SGA) is an important part of steganography (SG) that is in its turn a technology to hide some confidential information into innocent (at a single glance) cover object (CO). CO can be presented as digital motionless and video images, files containing digital audio signals (like speech and music) and so on. The main goal of SG (Information Hiding (IH) in a wider sense) is to provide such transform of CO into SG that a detection of SG against CO is either at all impossibly or it entails a very hard procedure even for the case of completely known embedding and extraction algorithms of messages, except perhaps of the stegokey. (Here, the Kerckhoff's principle is extended from cryptography to steganography.)

There is a great collection of well-known steganographic algorithms and many of them were described in an excellent monograph [1]. For simplicity reasons, we will consider here only motionless gray scale images, although our methods can be applied also to video or audio CO.

It is obvious that SGA is a very important part of IH due to two main reasons.

Firstly, SGA should be taken into account during the design of any steganographic algorithm because such algorithm is useless if it can be easily detected by some SGA.

Secondly, SGA is very important to prevent a leakage of sensitive information outside of some contour because otherwise it can be arranged by SG with innocent CO. (See for the thing the well-known "Data Leakage Prevention (DLP) system" [2].)

An important aspect of SGA was stroked in a special chapter of the monograph [1], and it is a subject of many scientific papers. But not all problems of SGA are already solved. It is obvious that if stegokey is known or can be found easily [3, 4, 5] then an attacker can be able to detect the SG presence after extraction of meaningful messages if they were not previously encrypted by sufficiently strong cipher. We assume in our paper (as it is common in the majority of scientific papers) that such previous encryption of messages is used. We consider the attacks that execute some *side information* about SG and, in particular do not use the statistics of CO for the detection procedure, a situation that is atypical for conventional SGA.

In the next section such side information is known as messages before encryption and may be embedded in CO. In section 3, we have side information as a fact that HUGO algorithm could be used with very strong cipher for message encryption and maybe with an unknown stegokey. Section 4 concludes the paper.

2. **Detection of the stegosystem with partly known plaintext that was encrypted previous to embedding.** In this section let us consider the following scenario:

1. Any stegosystem can be used for embedding but the steganalyst does not know which one has been used.
2. The extraction algorithm is known or can be found by the steganalyst.
3. The message is encrypted before embedding in CO by a not very strong ciphering procedure.
4. Part of the plain-text message is known by steganalyst.

We claim that under the conditions provided above, it is possible to create a general stegoanalytic algorithm that can be executed for any stegosystem.

This scenario is comparatively uncommon, however information security is very important even in rare situations. Even more, it is very common to consider in cryptography the so called *chosen-plaintext attack* that tries to break *semantic security*. In steganography this scenario assumes that the embedded message is encrypted by some block cipher and that the message is known (even partly) but there remains an open problem: to decide whether this message was embedded or not in a given testing object. It is important to strike that we are not going to extract the embedded information because it can be encrypted by cipher, but to detect only a fact of embedding.

In Figure 1, the Heye's substitution-permutation cipher (SPC) [6] is presented, and in Table 1 and 2, the S-box and the permutation mapping transforms are presented. Although this cipher has $2^{80} \approx 1.2 \times 10^{24}$ secret keys, hence a brute force attack by key exhaustion is intractable, it can be easily broken, by linear or differential cryptanalysis.

However, sometimes in stegosystems sufficiently simple encryption algorithms can be used. Moreover, we use as well some extension of substitution-permutation cipher with block length 32. For this case, we extend the 16-bit cipher with addition of the second half to the first one and keeping previous transforms in S-boxes. The Table 2 for permutation mapping is changed to Table 3 shown below.

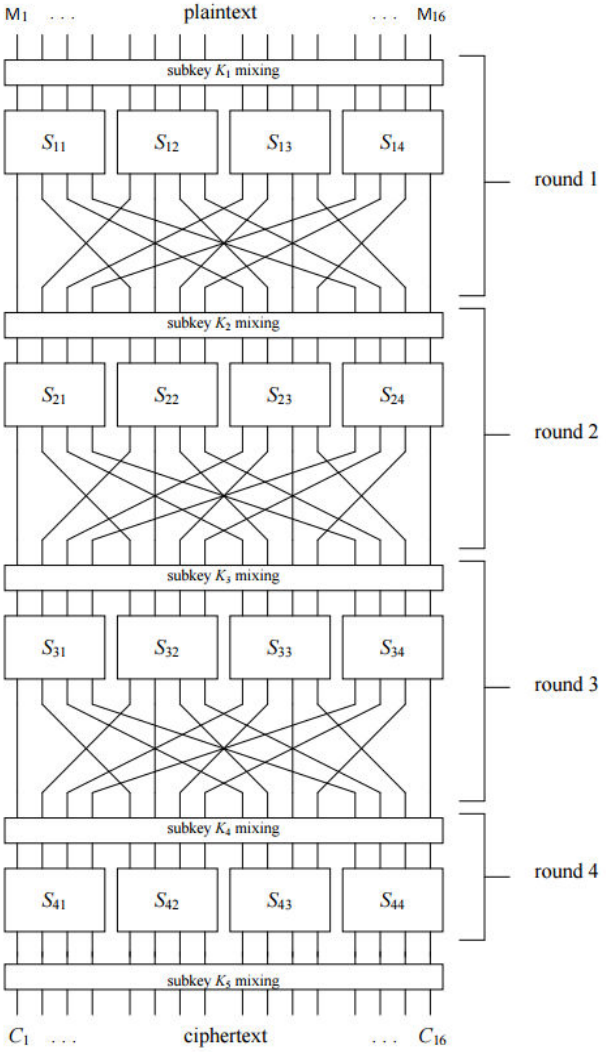


FIGURE 1. SPC cipher with block length 16 and four rounds

TABLE 1. S-box transforms for all S-boxes (they are presented in hexadecimal system

| Input | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10(A) | 11(B) | 12(C) | 13(D) | 14(E) | 15(F) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Output | E | 4 | D | 1 | 2 | F | B | 8 | 3 | A | 6 | C | 5 | 9 | 0 | 7 |

TABLE 2. Permutation mappings for all cipher rounds

| Input | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Output | 1 | 5 | 9 | 13 | 2 | 6 | 10 | 14 | 3 | 7 | 11 | 15 | 4 | 8 | 12 | 16 |

The inequality for mutual information between plaintext and ciphertext that should be valid for any cryptosystem is well known [7, 8]:

$$I(M^N, C^N) \geq H(M^N) - H(K^L) \tag{1}$$

where $M^N$ is a sequence of message symbols of the length $N$, $C^N$ is a sequence of ciphertext symbols of length $N$ (without loss of generality we may assume that these lengths are

TABLE 3. Permutation mappings for 32-bit block length cipher for all rounds

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Input | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| Output | 1 | 5 | 9 | 13 | 17 | 21 | 25 | 29 | 2 | 6 | 10 | 14 | 18 | 22 | 26 | 30 |
| Input | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| Output | 3 | 7 | 11 | 15 | 19 | 23 | 27 | 31 | 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 |

equal), and $K^L$ is the binary key string of the length $L$. We transform inequality (1) by dividing both its sides by $N$:

$$I'(M^N, C^N) \geq H'(M^N) - H'(K^L) \tag{2}$$

where the apostrophe ' means that we consider normalized values by $N$:

$$I'(M^N, C^N) \sim H'(M^N) > 0 \tag{3}$$

It follows from inequality (3) that if some message $M^N$ has been encrypted into the ciphertext $C^N$ with any unknown key $K^L$ of the limited length $L$ then for very large message length $N$ we get nonzero mutual information $I'(M^N, C^N)$. But this value approaches to zero if $M^N$ is not encrypted as $C^N$ with some key. Hence, we can take a decision about a choice of the message that is encrypted into the given ciphertext just by comparing the value $I'(M^N, C^N)$ with some threshold.

But the following problem appears - how to calculate a mutual information $I'(M^N, C^N)$? A solution of this problem, based on "binning" design, is very hard, generally speaking, but in [9], a method based on the notion of *k-nearest neighbor distance* was used. This approach can be termed as *fast mutual information calculation* (FMIC) between two $N$-dimension random vectors X and Y. It has been proved in [9] that FMIC can be performed by the following algorithm:

$$I(X, Y) = \psi(1) - \langle \psi(n_x + 1) + \psi(n_y + 1) \rangle + \psi(N) \tag{4}$$

where $X = \{x_1, x_2, ..x_N\}$ and $Y = \{y_1, y_2, ..y_N\}$ are vectors corresponding to $M^N$ and $C^N$, $\psi$ is the digamma function,

$$\forall x \in \mathbb{C}: \quad \psi(x) = \frac{1}{\Gamma(x)} \frac{\mathrm{d}\Gamma}{\mathrm{d}x}(x)$$

$\psi$ satisfies the recursion $\psi(x+1) = \psi(x) + \frac{1}{x}$ and $\psi(1) = C$, where $C = 0.5772156$ is the *Euler-Mascheroni constant*. For large $x \in \mathbb{R}^+$, $\psi(x) \approx \log(x) - \frac{1}{2x}$.

The value $n_x(i)$ is the number of points $x_j$ whose distance to $x_i$ is strictly less than $\varepsilon(i)/2$ and, similarly for $y$ instead of $x$. Here $\varepsilon(i)/2$ is the distance from $z_i = (x_i, y_i)$ to its neighbor and $\varepsilon_x(i)/2, \varepsilon_y(i)/2$ are distances between the same points projected into the $X$ and $Y$ subspaces. Obviously, $\varepsilon(i) = \max(\varepsilon_x(i), \varepsilon_y(i))$. $\langle .. \rangle$ is symbol that denotes an averaging over all $i \in \{1, 2, .., N\}$ and over all realizations of random samples. But in our case, we average only on all samples of integers $i \in [1, 2, .., N]$, namely $\langle .. \rangle = \frac{1}{N} \sum_{k=0}^{N} (\ldots)$.

In order to implement the relation (4) to estimate the left side of inequality (3), we map each of the plaintext blocks $M_i = (m_{i1}, m_{i2}, ..m_{in})$ into an integer $X_i$ and each of the ciphertext blocks $C_i = (c_{i1}, c_{i2}, ..c_{in})$ into an integer $Y_i$ according to trivial relations, respectively:

$$X_i = \sum_{j=0}^{n-1} x_{ij} 2^j; Y_i = \sum_{j=0}^{n-1} y_{ij} 2^j, i = 1, 2, .., N \tag{5}$$

where $n$ is the block cipher length; $x_{ij}, y_{ij}$ are binary symbols of the plaintext $M^N$ and the ciphertext $C^N$, respectively. (We assume of course that the block cipher is binary and

that it has the same length of input and output blocks.) The experimental investigation of the technique described above can be presented as follows.

There are produced two pseudo randomly generated binary sequences $M_I$ and $M_{II}$ both of length $n \cdot N$, where $n$=16 is the cipher block length and $N$ is the number of tested blocks. One of these sequences, say $M_I$ is encrypted by Heye's block cipher that gives $n.N$ ciphertext bits. (It is worth to note that in the case of meaningful plaintext, the entropy $H'(M^N)$ in (2) can be lesser than for truly random binary sequence but it still is not zero. Next, the mutual information $I(M_I, C)$ is calculated by (4) and (5), where the values $X_i$ are integers corresponding to $M_I$ and $Y_i$ are integers corresponding to $C = f(M_I, K)$, where $f(.)$ is the encryption function for Heye's 16-bit block cipher with 80-bit key chosen pseudo randomly. After that it is calculated also by (4) and (5) the mutual information between ciphertext $C$ obtained after encryption of plaintext $M_I$ and independent on it another plaintext $M_{II}$. The results of such calculations against the number of message bits $N$ are presented in Table 4.

TABLE 4. Mutual information between ciphertext and plaintext corresponding and no corresponding to given ciphertext against the plaintext bit length $N$

| N | $10^2$ | $10^3$ | $10^4$ | $2 \times 10^4$ | $4 \times 10^4$ | $8 \times 10^4$ | $3 \times 10^5$ | $10^6$ |
|---|---|---|---|---|---|---|---|---|
| $I'(M_I, C)$ | 0.3 | 1.200 | 5.52 | 7.057 | 8.77 | 10.30 | 12.650 | 14.24 |
| $I'(M_{II}, C)$ | −0.09 | 0.053 | 0.03 | 0.040 | 0.08 | 0.13 | 0.373 | 0.89 |

Within this table it can be seen that, in fact, the mutual information $I'(M_I, C)$ for valid plaintext $M_I$ encrypted into $C$ increases with $N$ and it approaches to a normalized entropy of truly random binary string of length 16. The mutual information $I'(M_{II}, C)$ between ciphertext (obtained for plaintext $M_I$) and the plaintext $M_{II}$ is close to 0. It is sufficient to select some threshold in order to distinguish between valid and invalid plaintexts for the already given ciphertext, for $N \geq 10^3$.

In Table 5 there are presented the results of calculation for cross correlation $R(M, C)$ between sequence $C$ and sequences $M_I$ and $M_{II}$ showing that such criteria cannot be used for a breaking the block cipher semantic security. (This is an obvious consequence of the presence of nonlinear transforms in the algorithm of Heye's block cipher contained into its S-boxes.)

TABLE 5. Cross correlation between ciphertext $C$ and plaintexts $M_I, M_{II}$ against the plaintext bit length $N$

| N | $4 \times 10^4$ | $8 \times 10^4$ | $3 \times 10^5$ | $10^6$ |
|---|---|---|---|---|
| $R(M_I, C)$ | −0.00068 | −0.038000 | 0.011 | −0.000977 |
| $R(M_{II}, C)$ | −0.00190 | −0.000556 | 0.003 | −0.000160 |

Next, a block cipher is considered with the same structure as Heye's cipher but with block length 32 and with round keys consisting from 32 bit each. The S-box transforms are shown in Table 1 and the permutation mapping is shown in Table 3. The experiment with such "extended cipher" was arranged similarly as for ordinary cipher described before with the only differences that two plaintext boxes $M_I$ and $M_{II}$ have the length 32 bits and the same length has ciphertext $C$. The results of simulations are presented in Table 6.

We see from Table 6, that despite the fact that mutual information $I'(M_I, C)$ grows much slower with respect to $N$ than the similar value for 16-bit block cipher (see Table 4), it is still exceeding the value $I'(M_{II}, C)$ when $N \geq 10^4$. This means that after a choice

TABLE 6. Mutual information between ciphertext and plaintext corresponding and not given ciphertext against the plaintext bit length $N$ for block cipher of the length 32

| N | $10^3$ | $10^4$ | $2 \times 10^4$ | $4 \times 10^4$ | $8 \times 10^4$ | $3 \times 10^5$ | $10^6$ |
|---|---|---|---|---|---|---|---|
| $I'(M_I, C)$ | $-0.065$ | 0.025 | 0.0380 | 0.078 | 0.0830 | 0.3626 | 0.9760 |
| $I'(M_{II}, C)$ | $-0.030$ | $-0.007$ | 0.0025 | $-0.012$ | 0.0055 | 0.0014 | 0.0017 |

of an appropriate threshold, it is possible to distinguish "valid" plaintext from "invalid" one for a given ciphertext. Thus, the proposed approach can break semantic security of at least for block ciphers with limited block length $n \leq 32$.

Our experiments with DES block cipher having block length 64 bits showed that this problem is unfortunately rather untractable, at least with the use of currently ordinary personal computer.

Finally, we can conclude that if steganalyst knows a part of plaintext that is expected to be embedded into some testing object and the length of plaintext/ciphertext are sufficiently to calculate mutual information, then comparing this value with some threshold, he/she be able to take a decision that the testing object belongs to SG, otherwise to CO.

It is worth to note that an opinion that part of plaintext knowledge is unrealistic seems to be naive. (We remember that namely similar situation, but of course connected with cryptography but not with steganography, was executed by US navy to break Japanese "purple cipher" during the War at Pacific'.)

3. **Detection of the HUGO stegosystem with side information about the fact of encryption with any strong block cipher of the messages previous to embedding.** Let us make the following assumptions:

1. Any stegosystem can be used for embedding, but steganalyst does not know which one has been used.
2. Extraction algorithm is known or can be found by steganalyst.
3. Message is encrypted before embedding in CO by strong cipher.

In contrast with the scenario described in previous section plaintext is not required. Next in the section we show how this steganalityc method can be implemented.

A new stegoanalytic algorithm has been proposed in [10]. It was based on the side information to which the messages are subjected, with encryption using a strong block cipher previous to embedding. Such assumption is not very limiting because otherwise, if the extraction stegoalgorithm is known for an attacker (or somewhat can be found) it is easy to decide whether in a tested object occurs SG, if the extracted message is meaningful (see Algorithm 16.2 in [1]), and it is CO if extracted message is meaningless.

If the extracted messages consist of ciphertext got by strong encryption, then it is very likely to think that they are close to a perfect pseudo-random bit sequence satisfying the so called *NIST-Tests* [11] listed in Table 7. Otherwise, if the extracted sequence does not pass some of the NIST tests then it is assumed to be a cover object. (In fact, it is unlikely that a clear CO satisfies all NIST tests.) In reality by selecting some threshold, it can be assumed that if the number of NIST tests exceeds this threshold then the testing object is SG, otherwise it is CO.

Similarly to an approach presented in [10] we are going to consider also a method of SVM-based classification with the use of p-values as the results of NIST tests. But in contrast to [10], we consider one of the most advanced steganographic algorithm HUGO [12] and include also a search of stego key for it.

TABLE 7. Titles of NIST tests on pseudo randomness

| N | Title of test |
|---|---|
| 1 | The frequency test |
| 2 | Frequency test within a block |
| 3 | The runs test |
| 4 | Tests for the longest-run-of-ones in a block |
| 5 | The binary matrix rank test |
| 6 | The discrete Fourier transform (spectral) test |
| 7 | The non-overlapping template matching test |
| 8 | The overlapping template matching test |
| 9 | Maurer's "Universal Statistical" test |
| 10 | The linear complexity test |
| 11 | The serial test |
| 12 | The approximate entropy test |
| 13 | The cumulative sums (cusums) test |
| 14 | The random excursion teat |
| 15 | The random excursions variant test |

But firstly, let us remember the HUGO SG. The core idea of HUGO's embedding algorithm (in line with [12, 19]) is the following: for a given bit sequence $m$, intended to be embedded into CO, and a given sequence of the least significant bits (LSB) $x$, find the LSB sequence $y$ after embedding, which satisfies the equation:

$$H \cdot y = m \tag{6}$$

with an additional condition to minimize the following value:

$$\triangle = \sum_{i=1}^{n} \rho_i \mid x_i - y_i \mid \tag{7}$$

where $\rho_i, i \in \{1, \ldots, n\}$ is the given weight function; $x_i, y_i, i \in \{1, \ldots, n\}$ are the $i$-th components of vectors $x$ and $y$, respectively; $n = n_1 \cdot n_2$, for $n_1 \times n_2$ sizes of motionless image taken as CO; and H is an $n \times N$ generator matrix of some trellis code.

The matrix H can be presented as a step-matrix obtained by consecutive shifting of a $t \times w$ submatrix $\widehat{H}$. The weight function $\rho_i$ can be calculated in line with SPAM-based functionals described in [12]. It has been shown in [12] that the solution of (6) given condition (7) can be obtained with the use of Viterbi Algorithm (VA) (in [13] there was proved that Generalised Viterbi Algorithm (GVA) is optimal in the considered case).

In order to provide the best security of SG it is necessary to select the matrix H (namely the submatrix $\widehat{H}$) giving a maximum of the error probability $P_e = \frac{1}{2}(P_m + P_{fa})$, as close as possible to $\frac{1}{2}$ where $P_m$ is the probability of SG missing and $P_{fa}$ is the probability of SG false alarm.

But it is an expensive time-consuming method.

A very nice idea was proposed by C. Cachin in [14]. It is based on the notion of *relative entropy* $D(P_w//P_c)$ (or *Kullbach-Leibler Divergence* (KLD)). This criterion asserts that for any steganalytic method the following inequality holds:

$$P_{fa} \log(\frac{P_{fa}}{(1-P_m)}) + (1-P_{fa}) \log(\frac{(1-P_{fa})}{P_m}) \leq D(P_w//P_c) \tag{8}$$

FIGURE 2. The picture "Underwater World" of the size 12992 × 6080 pixels before embedding.

where

$$D(P_w//P_c) = \sum_{x\in X} P_w(x)\log(\frac{P_c(x)}{P_w(x)}) \tag{9}$$

$P_w$ is the probability distribution after embedding, $P_c$ is the probability distribution of CO, and X is the set of image luminance.

But the problem is to calculate KLD even for such CO as motionless grey scale images.

Fortunately, it has been proposed in [15] a method for KLD calculation based on the notion of the *nearest neighbor distance* (NND) for given samples of both SG and CO. (Similar approach was presented in section 2.)

Let us assume that $(x_1, x_2, .., x_n)$ and $(y_1, y_2, .., y_m)$ are vectors of random samples with the probability distributions $P$ and $Q$, respectively. Then the distance from $x_i$ to the "nearest neighbor" $x_j$, $j \neq i$, is:

$$\rho_i = \min_{j=\overline{1,n}, j\neq i} (\parallel x_i - x_j \parallel) \tag{10}$$

where $\parallel . \parallel$ is $L^2$ norm in $R^d$. In a similar manner, the distance from $x_i$ to the "nearest neighbor" $y_j$, $j \neq i$,

$$\gamma_i = \min_{j=\overline{1,m}} (\parallel x_i - y_j \parallel) \tag{11}$$

Then KLD can be estimated as:

$$D_{n,m}(P//Q) = \frac{d}{n}\sum_{i=1}^{n}\log(\frac{\gamma_i}{\rho_i}) + \log(\frac{m}{n-1}) \tag{12}$$

It was proved in [15] that under some not very strong conditions, the following asymptotic relation holds:

$$\lim_{n,m\to\infty} E(D_{n,m}(P//Q)) = D(P//Q) \tag{13}$$

We note that in application to stegosystems the above procedure has to be specified to be valid even for a single testing image [16].

Let us take only one image and divide it into disjoint $(L\times L)$-pixel areas arranged as a *chess board*, where white areas correspond to the set X (no embedding) and black areas correspond to the set Y embedding by HUGO algorithm.

In Figure 2 the selected grey scale image of size 12992 × 6080 pixels is presented for the experiment. The size of disjoint areas $L$ was selected as 32. The results of KLD calculations are presented in Table 8. For a comparison we calculated also KLD for LSB-based embedding algorithm. We note that the parameter $p$ for LSB-based SG

TABLE 8. The results of KLD estimation based on NND for the image "Underwater World" with different embedding rate $R = p^{-1}$ for LSB and $R = \frac{1}{w}$ for HUGO

| R | 0 | 0.1 | 0.2 | 0.3 | 0.4 |
|---|---|---|---|---|---|
| HUGO | −0.07 | 7.52 | 22.46 | 48.13 | 83.48 |
| LSB | −0.07 | 32.16 | 85.22 | 143.89 | 199.70 |

means the probability of embedding into each pixel, whereas the 1-$p$ is the probability of no embedding (obviously, the SG owner has a stegokey and hence he/she knows in which pixels were the embeddings). The HUGO algorithm provides the embedding rate determined by its trellis code structure.

It can have seen from Table 8 that although the values of KLD are very large, hence it is useless for a calculation of the steganalytic security by (8). However, the following qualitative conclusions can be drawn:

- the grater is the embedding rate, the lesser is the SG security, and
- the HUGO SG is much more secure than the LSB-based SG.

These facts are well known in steganography but we are able to go further and investigate the best submatrix $\widehat{H}$, that provides the most secure HUGO SG. In fact, let us examine how the structure of submatrix $\widehat{H}$ affects on the SG HUGO detectability. We use here an image size 512 x 512 just for simplicity.

Table 9 shows a dependence between structure of the $2 \times 2$ submatrix $\widehat{H}$ and corresponding to them KLD averaged on many images.

TABLE 9. The dependence between $2 \times 2$ submatrix $\widehat{H}$ and the values KLD

| Submatrix $\widehat{H}$ | $\begin{bmatrix}1 & 1\\ 1 & 0\end{bmatrix}$ | $\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$ | $\begin{bmatrix}0 & 1\\ 1 & 1\end{bmatrix}$ | $\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}$ | $\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$ | $\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}$ |
|---|---|---|---|---|---|---|
| $D(X//Y)$ | 29.50 | 57.65 | 29.30 | 30.31 | 53.68 | 30.41 |

From this table it can be seen that the security of the HUGO-based SG depends on the submatrix $\widehat{H}$ structure. In the sequel we are going to confirm these results by the use of a new steganalytic algorithm and execute it also for a stegokey search.

Let us consider firstly a scenario where a stegokey of HUGO-SG (that is a submatrix $\widehat{H}$) is known exactly for an attacker. This means that he/she is able to extract the encrypted messages if the testing object was SG. If the testing object was a CO, then we extract also some sequences but we believe that they have more worse pseudo randomness than for the case of SG for the number are passed NIST tests. Next it is possible to select some thresholds and to take a decision that the testing object is SG if this threshold was exceeded, otherwise that it is CO.

We selected for our experiment 1000 grey scale images of the size 512 x 512 pixels from database used in [17]. In order to select optimal threshold were taken 500 both SG and CO. For testing we used 500 other images with or without embedding.

In Table 10 are presented the values of error SG detecting probabilities $P_e = \frac{1}{2}(P_m + P_{fa})$ for different submatrix $\widehat{H}$ of HUGO SG and for optimally chosen thresholds. (For a save of simplicity we denote each submatrix $\widehat{H}$ by vector, where each integer is representation of columns in binary form.)

TABLE 10. The probabilities of error $P_e$ of SG detection based on threshold method for different submatrices $\widehat{H}$

| | $\widehat{H}$ | R | D(X//Y) | $P_e$ based on Threshold method |
|---|---|---|---|---|
| 1 | [1 2] | 1/2 | 57.65 | 0.291 |
| 2 | [2 1] | 1/2 | 53.68 | 0.318 |
| 3 | [3 2] | 1/2 | 30.41 | 0.335 |
| 4 | [1 3] | 1/2 | 30.31 | 0.336 |
| 5 | [2 3] | 1/2 | 29.30 | 0.354 |
| 6 | [3 1] | 1/2 | 29.50 | 0.351 |
| 7 | [15 6] | 1/2 | 28.63 | 0.367 |
| 8 | [15 4] | 1/2 | 25.22 | 0.368 |
| 9 | [11 5] | 1/2 | 23.82 | 0.376 |
| 10 | [11 15] | 1/2 | 22.33 | 0.388 |

We can see from Table 10 that the best submatrix

$$\widehat{H} = \begin{bmatrix} 11 & 15 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \tag{14}$$

gives maximal value $P_e = 0.388$. This fact is very reasonable because it has the smallest $D(X//Y)$.

In order to improve SG HUGO detecting was used SVM. Such approach has been proposed before in the paper [16] but only against LSB-based SG. We will assume that each of NIST tests is presented by its $p$-value (A small $p$-value is an indication that the null hypothesis is false). The vector of $p$-values corresponding to all 15 NIST tests is used both at the training and detecting stages of SVM. We apply the most effective version of SVM known as nonlinear kernalized weighted one, where the kernel function is Gaussian: $K(x, x') = \exp(-\gamma \parallel x - x' \parallel^2)$ with $\gamma > 0$ being a parameter controlling the kernel width and $\parallel x \parallel$ is the Euclidean norm of $x$. After an optimization of both $\gamma$ and the penalization coefficient $C$ (similar to the same procedure considered more detail in [16]) we get the results presented in Table 11.

TABLE 11. The error probabilities $P_e$ of SG detection based on an execution of optimized procedure for SVM

| Submatrix $\widehat{H}$ | [1 2] | [2 1] | [3 2] | [1 3] | [2 3] | [3 1] | [15 6] | [15 4] | [11 5] | [11 15] |
|---|---|---|---|---|---|---|---|---|---|---|
| $P_e$ | 0.255 | 0.274 | 0.307 | 0.308 | 0.321 | 0.327 | 0.322 | 0.335 | 0.34 | 0.35 |

It can be seen from this table that the error detecting probability $P_e$ occurs slightly lesser than $P_e$ after the procedure when using the threshold presented in Table 10, but the optimal submatrix $\widehat{H}$ is the same. It seems that the error detecting probability occurs still sufficiently large in comparison with the best known cryptanalytic methods [17], but nevertheless, such approach opens a new direction in a stegokey searching (we remember that for HUGO-SG, a submatrix $\widehat{H}$ plays the role of stegokey). Then we assume that some submatrices $\widehat{H}$ were used for HUGO embedding procedure but in the detection procedure, the submatrix $\widehat{H}$ is unknown.

Let us consider two classification procedures to recognize a stegokey. The first procedure is based on a threshold technique with the use of NIST tests. This means that we use

TABLE 12. The number of images that was classified by different submatrices $\widehat{H}$ by threshold method given different embedding submatrices

| | | Submatrices for extraction | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | [1 2] | [2 1] | [3 2] | [1 3] | [3 1] | [2 3] | CO |
| Submatrices for embedding | [1 2] | 236 | 218 | 226 | 212 | 217 | 217 | 0 |
| | [2 1] | 144 | 265 | 235 | 217 | 268 | 252 | 0 |
| | [3 2] | 172 | 183 | 283 | 211 | 253 | 238 | 0 |
| | [1 3] | 163 | 182 | 222 | 284 | 208 | 233 | 4 |
| | [3 1] | 157 | 190 | 207 | 196 | 286 | 219 | 3 |
| | [2 3] | 149 | 178 | 213 | 198 | 230 | 256 | 1 |
| | CO | 130 | 154 | 164 | 166 | 184 | 182 | 136 |

TABLE 13. The error probabilities $P_e$ of classification given different submatrices and CO

| | $H_1$=[1 2] | $H_2$=[2 1] | $H_3$=[3 2] | $H_4$=[1 3] | $H_5$=[3 1] | $H_6$= [2 3] | CO |
|---|---|---|---|---|---|---|---|
| $P_e$ | 0.4165 | 0.419167 | 0.428167 | 0.416 | 0.440667 | 0.4675 | 0.365333 |

one by one all different submatrices for extraction and we take a decision about those submatrix $\widehat{H}$ providing the greatest number of images passing all NIST tests.

In Table 12 are presented the results of stegokey searching after testing of 500 different images.

We note that a decision about CO instead of some SG was taken if the number of passed tests was less than 13. The probabilities of error (a falsification of submatrices) are presented in Table 13. From Table 13 it can be seen that $P_e$ is sufficiently large. In order to make a submatrix recognition more reliable we try to improve the results using multi-class SVM for submatrix recognition. In line with proposal in [17], we execute the so-called *Max-Wins algorithm* considered in [18]. We explain this idea briefly for the case with 6 submatrices plus one (the 7th class corresponds to CO). The number of pair-wise combinations taken from 7 outcomes $\binom{7}{2} = 21$. Each pair out of 21 can be classified by 2 classes SVM as usually. A possible result of such classification is presented in Table 14.

TABLE 14. Example of pair-wise recognition for 7 classes

| | Results for binary classifiers | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Pair | 1-2 | 1-3 | 1-4 | 1-5 | 1-6 | 1-7 | 2-3 | 2-4 | 2-5 | 2-6 | 2-7 | 3-4 | 3-5 | 3-6 | 3-7 | 4-5 | 4-6 | 4-7 | 5-6 | 5-7 | 6-7 |
| Classified result | 2 | 3 | 4 | 1 | 6 | 1 | 2 | 2 | 5 | 2 | 2 | 3 | 5 | 3 | 3 | 5 | 4 | 7 | 6 | 5 | 7 |

After a perform of the first step (see above) it is necessary to arrange a "voting procedure". Namely, to count how many times is presented every element in Table 14. Finally, we take a decision about class with maximal vote. The results of the last procedure are presented in Table 15.

(It is obvious that the above presented algorithm can be extended for any number of classes.) Following to "Max-Wins" procedure we realized multi-class recognition for 6 submatrices and plus CO presented in Table 16.

It can be seen from this table that results of submatrix recognition for Max-Wins method are better than for the case of threshold procedure (See Table 12.)

TABLE 15. Results of "voting" on the Table 14. The maximal element is shown by hatching

| Class | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Classified quantity | 2 | 5 | 4 | 2 | 4 | 2 | 2 |

TABLE 16. The number of images that were classified by SVM following Max-Wins recognition for different embedding and extracted submatrices

| | | Submatrices for extraction | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | [1 2] | [2 1] | [3 2] | [1 3] | [3 1] | [2 3] | CO |
| Submatrices for embedding | [1 2] | 257 | 30 | 44 | 56 | 39 | 72 | 2 |
| | [2 1] | 107 | 147 | 63 | 69 | 25 | 89 | 0 |
| | [3 2] | 82 | 49 | 141 | 79 | 60 | 85 | 4 |
| | [1 3] | 96 | 43 | 55 | 187 | 43 | 70 | 6 |
| | [3 1] | 86 | 49 | 70 | 72 | 132 | 80 | 11 |
| | [2 3] | 89 | 53 | 56 | 72 | 52 | 176 | 2 |
| | CO | 70 | 31 | 43 | 66 | 49 | 80 | 161 |

To be more precisely we present in Table 17 the results of such recognition error probabilities $P_e$ for different embedding submatrices.

TABLE 17. The values of error probabilities $P_e$ for submatrix recognition

| | $H_1$=[1 2] | $H_2$=[2 1] | $H_3$=[3 2] | $H_4$=[1 3] | $H_5$=[3 1] | $H_6$= [2 3] | CO |
|---|---|---|---|---|---|---|---|
| $P_e$ | 0.331333 | 0.3955 | 0.414167 | 0.382 | 0.412667 | 0.403333 | 0.343167 |

Comparing Table 13 and 17 we can see that SVM-based method is slightly better than threshold-based approach.

Although the size of submatrix was chosen as 2 x 2, it is not a problem to extend it to more sizes of such submatrices. (But it worth to note that very large sizes for HUGO-based SG result in a growing of the embedding complexity.)

4. **Conclusions.** A very common situation arises when encryption of messages is performed before embedding them into cover objects. We have considered two scenarios that are very natural under this condition and can be termed as side attacks on stegosystem because they work only under message encryption and known (or easily found) extraction algorithm. For the first scenario it is assumed that the embedded message is at least partially known for a steganalyst, but it is unknown whether this message was embedded or not into given cover object. Detection algorithm based on fast calculation of mutual information was investigated in the current paper. Such attack is typical for *cryptography* as *known plaintext attack* but although on that context it is used to find the cryptographic key, in stegography it is required to prove a fact of embedding the given message into given encrypted text. We show that this problem can be solved but not for very strong block ciphers having large block lengths. Justification of the last condition may be a fact that in steganography often not very strong ciphers can be used in a hope that messages are hidden already by steganographic algorithm.

By the way, similar condition was taken in the paper [23] but it executed there by completely another way.

The second scenario apply on the contrary to sufficiently strong cipher with unknown messages but known (or computable) stegokey. If total search of stegokey is tructable

problem then it be also found by our approach. As an example, we considered search of submatrix for HUGO stegosystem that plays there the role of stegokey.

We considered the use of GOST algorithm for encrypted messages but our approach can be easily extended to any strong cipher like 3DES or AES.

In the near future we aim to investigate an approach with a modification of strong ciphers to such ones that keeping a good protection against cipher breaking but simultaneously does not satisfy to NIST tests on pseudo randomness that allows to protect SG against the proposed attack.

We are going also to try our universal approach as method of SGA for more contemporary SG [20, 21, 22].

## REFERENCES


[1] J. Fridrich, Steganography in Digital Media: Principles, Algorithms and Application, *Cambridge University Press*, 2009.

[2] M. Jain, S. K. Lenka, A Review on Data Leakage Prevention using Image Steganography, *International Journal of Computer Science Engineering (IJCSE)*, vol.5, no.2, pp.56–59, 2016.

[3] N. F. Johnson, S. Jajodia, Steganalysis: The investigation of hidden information, *Proc. of the 1998 IEEE Information Technology Conference, Information Environment for the Future*, New York, USA, pp.113–116, 1998.

[4] J. Liu, G. Tang, Stego Key Estimation in LSB Steganography, *Journal of Multimedia*, vol.7, no.4, pp.309–313, 2012.

[5] J. Liu, Y. Tian, T. Han, J. Wang, and X. Luo, Stego key searching for LSB steganography on JPEG decompressed image, *Science China Information Sciences*, vol.59, no.3, 2016.

[6] H. M. Heys, A Tutorial on Linear and Differential Cryptanalysis, *Cryptologia*, vol.26, no.3, pp.189–221,2002.

[7] C. E. Shannon, Communication theory of secrecy systems, *Bell Labs Technical Journal*, vol.28, no.4, pp.656–715, 1949.

[8] HCA. van Tilborg, Fundamentals of Cryptography, *Kluwer Academic Publishers*, 1999.

[9] A. Kraskov, H. Stgbauer, and P. Grassberger, Estimating mutual information, *Physical review E*, vol.69, no.6, pp.066138-1–066138-16, 2004.

[10] V. Korzhik, I. Fedyanin, A. Godlewski, and G. Morales-Luna, Steganalysis Based on Statistical Properties of the Encrypted Messages, in *Computer Network Security. MMM-ACNS 2017*, J. Rak, J. Bay, I. Kotenko, L. Popyack, V. Skormin, K. Szczypiorski (eds.), Warsaw, Springer-Cham, LNCS 10446, pp.288–298, 2017.

[11] L. E. Bassham III, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, ... and N. A. Heckert, Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications, *National Institute of Standards and Technology*, 2010.

[12] T. Pevn, T. Filler, and P. Bas, Using high-dimensional image models to perform highly undetectable steganography, in *Information Hiding. IH 2010*, R. Bhme, P. W. L. Fong, R. Safavi-Naini (eds.), Berlin, Springer-Heidelberg, LNCS 6387, pp.161–177, 2010.

[13] V. Korzhik, G. Morales-Luna, and I. Fedyanin, Using the generalised Viterbi algorithm to achieve a highly effective stegosystem for images, *Pro. of the Computer Science and Information Systems (FedCSIS)*, Lodz, Poland, pp.855-860, 2015.

[14] C. Cachin, An information-theoretic model for steganography, *Information and Computation*, vol.192, no.1, pp.51-56, 2004.

[15] Q. Wang, S. R. Kulkarni, and S. Verd, A nearest-neighbor approach to estimating divergence between continuous random vectors, *Proc. of the 2006 IEEE International Symposium on Information Theory*, Washington, USA, pp.242–246, 2006.

[16] V. Korzhik, I. Fedyanin, Steganographic applications of the nearest-neighbor approach to Kullback-Leibler divergence estimation, *Proc. of 2015 Third International Conference on Digital Information, Networking, and Wireless Communications (DINWC)*, Moscow, Russia, pp.133–138, 2015.

[17] P. Bas, T. Filler, and T. Pevn, Break Our Steganographic System: The Ins and Outs of Organizing BOSS, in *Information Hiding. IH 2011*, T. Filler, T. Pevn, S. Craver, A. Ker (eds.), Berlin, Springer-Heidelberg, LNCS 6958, pp.59–70, 2011

[18] T. Pevn, J. Fridrich, Towards multi-class blind steganalyzer for JPEG images, in *Digital Watermarking. IWDW 2005*, M. Barni, I. Cox, T. Kalker, H. J. Kim (eds.), Berlin, Springer-Heidelberg, LNCS 3710, pp.39–53, 2005.

[19] W. W. Liu, G. J. Liu, and Y. W. Dai, Alternative Syndrome-Trellis Codes With Reduced Trellis Complexity, *Journal of Information Hiding and Multimedia Signal Processing*, vol.5, no.4, pp.769–777, 2014.

[20] T. Denemark, J. Fridrich, Improving steganographic security by synchronizing the selection channel, *Proc. of the 3rd ACM Workshop on Information Hiding and Multimedia Security*, New York, USA, pp.5–14, 2015.

[21] R. Wazirali, Z. Chachzo, Hyper Edge Detection with Clustering for Data Hiding, *Journal of Information Hiding and Multimedia Signal Processing*, vol.7, no.1, pp.1–10, 2016.

[22] T. Denemark, J. Fridrich, Model based steganography with precover, *Electronic Imaging*, vol.2017, no.7, pp.56–66, 2017.

[23] J. Gan, J. Liu, X. Luo, C. Yang, and F. Liu, Reliable steganalysis of HUGO steganography based on partially known plaintext, *Multimedia Tools and Applications*, vol.77, no.14, pp.18007-18027, 2018.