9th International scientific conference
Technics and Informatics in Education – TIE 2022
16-18 September 2022

Session: *Engineering Education and Practice*
Professional paper
DOI: 10.46793/TIE22.171M

Implementation of embedded messages using steganography in the PHP software package

Marija Mojsilović[*], Selver Pepić and Goran Miodragović
Academy of Vocational Studies Šumadija, Department of Trstenik, Serbia
[*] mmojsilovic@asss.edu.rs

Abstract: *The term steganography is usually associated with hiding and concealing information and messages. People, and even IT professionals, very rarely come into contact with steganography and steganalysis. Only messages are protected by cryptographic protection, while steganography can be said to protect both messages and parties participating in the communication. Steganography also means hiding messages inside computer files and data streams. This paper provides an overview of the implementation of embedded messages using steganography in the PHP software package. Emphasis is placed on hiding information, i.e. messages in JPEG images. As well as decoding or reading a hidden message. The field of steganography is naturally linked to the field of steganalysis, the primary goal of which is the detection of a hidden message, and then its extraction from the object of the message carrier. The most commonly used method for hiding messages is LSB, it is a method that changes the least significant bits to match the secret message. Then, by passing the steganographic key, the message is encrypted using the RC4 algorithm.*

Keywords: *Cryptography; Steganalysis; Steganography; JPEG; PHP*

1. INTRODUCTION

Information is one of the most important resources in this world, so it must be protected from third parties. The method of protecting information content is cryptography. We use cryptography to prevent the leakage of classified information to third parties [1].

The rapid development of technology has had a major impact on the exchange of information. In this modern era, maintaining security during the exchange of information is essential. Many algorithms have been used to ensure that the data exchanged is confidential [2]. One way to protect information is to keep it secret. The secrecy of information can be preserved by encrypting it.

By using mathematical tools, it is possible to achieve that only authorized persons have access to confidential data.

Encryption moves the original message so that it becomes incomprehensible, and obtaining original information from the encrypted message is achieved by decrypting using the appropriate key. This type of confidentiality encroaches on the field of steganography. The field of steganography is naturally associated with the field of steganalysis, whose primary goal is the detection of a hidden message, and then its separation from the object of the bearer of the message. Steganography is the practice of concealing information or a message in secret communication that involves hiding information in any multimedia variant such as text, image, or video [3].

The paper describes the procedures and techniques of steganography and steganalysis of digital images. Emphasis is placed on BMP and JPEG image formats.

2. STEGANOGRAPHY AND STEGANALYSIS

2.1. Steganography

Steganography is a technique of hiding secret messages in such a way that no one except the transmitting and receiving sides is aware of the existence of communication. Hiding messages is based on disguising the message within images, movies, and text. The main advantage of steganography over cryptography is the fact that messages do not attract attention. We can say that steganography can avoid an attack since the attacker is not aware of the existence of communication in a communication channel [4].

Steganography has an advantage in countries where cryptographic techniques for encrypting messages are prohibited. Cryptographic protection protects only messages, while steganography can be said to protect both messages and parties involved in communication. Steganography also involves hiding messages within computer files and data streams. In the case of digital steganography, the message may be within a document, image, or film [4].

Steganography is the science and writing skills of a message in such a way that no one, except the sender and recipient of the message, doubts its existence [5].

Steganography is used to hide secret messages in other messages so that only existing secrets are hidden. In recent times, people are hiding messages in pictures. Replace the least significant bit of each byte of the image with the message bits. The image will not change significantly - in most graphic formats, more colors are displayed than the human eye can recognize - so the message can be unpacked at the end of its journey [6].

Image steganography is the technique of changing image colors as a mechanism for hiding data in an image. This is usually done by changing the smallest significant part, to change the color of the pixels very slightly. A pixel in an image is stored as a 24-bit binary number consisting of red, green, and blue channels. By choosing the least important bit, the color remains almost unchanged [13].

To convert a string to its ASCII equivalent using PHP, we need to run each character through the ord() function which will give us the integer for that character that represents its ASCII code. To convert a decimal number to binary in PHP we use the decbin() function which will return a string representing the binary number. The problem with this function is that it will return the binary number to the most significant digit. In other words, the 8-bit number we need to represent 'a' will be converted into a 7-bit number since the leading 0 will be removed. Since the return of this function is a string, we pass it through the str_pad() function to force all 8-bit numbers to be present [13].

In the continuation of the work, the PHP codes are shown.

2.1.1. Steganographic system

How safe a steganographic system is depends on how well it resists passive, active, and malicious attacks.

The steganographic system is robust if the hidden information can be changed only by major changes to the stego object [10].

A secure steganographic system meets four conditions:

- The hiding algorithm is public, but a secret key is used.
- Only the person with the secret key can detect, remove and prove the permanence of the secret message. No one else can detect any statistical trace of the existence of a secret message.
- Even when an attacker knows the content of one transmitted message, he is unlikely to decipher the content of the remaining messages.
- Detection of a secret message by a computer is too demanding [10].

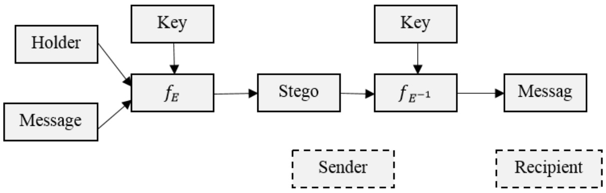


Figure 1. *Steganographic system*

Fig. 1, shows the way the steganographic system works, where:

- Holder - an image within which a secret message is hidden,
- Message - a secret message that needs to be hidden,
- Key - steganographic key, function parameter f E,
- f_E - steganographic function "embedding"
- Stego - steganographic file,
- f_(E^(-1)) - steganographic function "extraction"

2.2. Steganalysis and methods of detection

Steganalysis is a continuation of steganography. It is a science that deals with the detection of hidden messages. When a message is detected, the analysis determines the size of the hidden message and how to separate it from the existing object. With the development of technology and the Internet, cryptographic methods are being replaced by new digital methods, where there is less risk of revealing information during transmission. It is increasingly common to hide information in an image file, due to less suspicion and a large number of photos that are constantly transmitted.

Steganalysis techniques can be classified similarly to cryptanalysis techniques, depending on how much information we have:

- “Steganography-only” attack - when only steganographic media is available for analysis,
- “Known-holder” attack - when we have a carrier and medium for analysis,
- “Known-message” attack - when we know a hidden message,
- “Chosen-steganography” attack - when we know the medium and algorithm,
- "Chosen-message" attack - when we know the message and algorithm,
- “Known-steganography” attack - when we know the carrier, medium, and algorithm [11].

The process of discovering these contents is much more complicated and complex than in cryptanalysis because in cryptanalysis it is known that the subject file under investigation contains some data or information, while in steganalysis it is not known whether there is other content in the suspicious file. Such hidden content can be inserted anywhere (eg on the Internet on the web) in:

- Website text,
- Pictures on the web,
- Audio-video content on the website,
- Within any link (extended HTML).

2.3. LSB method

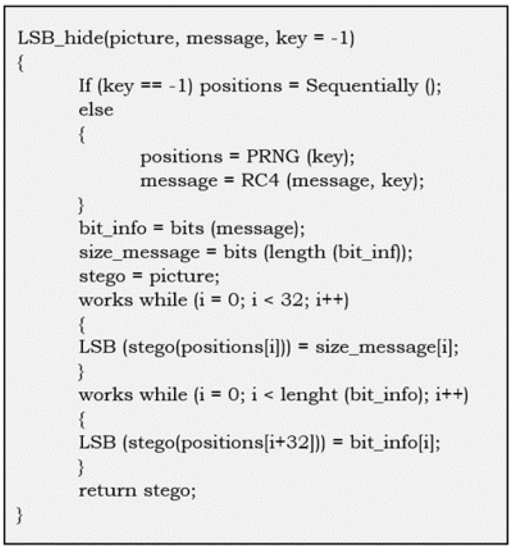
Most images on the Internet consist of a rectangular map of image pixels (represented as bits) where each pixel is located and it is color. These pixels are displayed horizontally row by row. The number of bits in a color scheme, called bit depth, refers to the number of bits used for each pixel. The minimum bit depth in current color schemes is 8, which means that 8 bits are used to describe the color of each pixel. Grayscale images use 8 bits for each pixel and can display 256 different colors or shades of gray. Digital color images are usually stored in 24-bit files and use the RGB color model. All color variations for the pixels of a 24-bit image are derived from three primary colors: red, green, and blue, and each primary color is represented by 8 bits. So in one given pixel, there can be 256 different amounts of red, green, and blue, which is more than 16 million combinations, resulting in more than 16 million colors.

The most commonly used steganographic technique is the use of LSB because it may contain hidden patterns. Statistical analysis of LSB data is a widespread method for detecting these samples. One of the most common patterns is the correlation between HOB (High Order Bits) and LSB, which is usually presented in hardware, such as a camera, which is used to generate original data. This type of attack is the most successful because most steganographic algorithms work under the assumption that LSB is random. Statistical analysis can detect changes made to the LSB [12].

The LSB method changes the least important bits to match the secret message. The following paper will explain how to hide a message in a 24-bit BMP image using the LSB method.

The pseudocode of the simple LSB method of hiding the implemented image within the work is given in Fig. 2. If a steganographic key is passed to the algorithm, then the key initializes PRNG, but also encrypts the message using the RC4 algorithm. If no key is specified, the message is hidden without encryption. The algorithm first hides the size of the message and then the message itself.

To understand the steganography algorithms that can be used when embedding data in the transform domain, one must first explain the type of file format connected with this domain. The JPEG file format is the most popular image file format on the Internet, because of the small size of the images. To compress an image into JPEG format, the RGB color representation is first converted to a YUV representation. In this representation, the Y component corresponds to the luminance (or brightness) and the U and V components stand for chrominance (or color) [6].



```
LSB_hide(picture, message, key = -1)
{
        If (key == -1) positions = Sequentially ();
        else
        {
                positions = PRNG (key);
                message = RC4 (message, key);
        }
        bit_info = bits (message);
        size_message = bits (length (bit_inf));
        stego = picture;
        works while (i = 0; i < 32; i++)
        {
        LSB (stego(positions[i])) = size_message[i];
        }
        works while (i = 0; i < lenght (bit_info); i++)
        {
        LSB (stego(positions[i+32])) = bit_info[i];
        }
        return stego;
}
```

Figure 2. *Pseudocode hiding of the message by the LSB method*

3. RC4 CRYPTOGRAPHIC ALGORITHM

RC4 is the most commonly used cryptographic data flow algorithm. The paper uses the RC4 algorithm, which, in addition to encryption, can also serve as a generator of pseudo-random numbers.

The algorithm works in OFB mode, the key sequence does not depend on the plaintext. It has an 8 * 8 S-box: S0, S1, ..., S255. The elements of the box are the permutation of numbers from 0 to 255, and the permutation is a function of the key of variable length. The algorithm has two counters, i and j, which are initialized with zero. To generate a random byte, do the following:

$$i = (i + 1) \bmod 256$$

$$j = (j + S_i) \bmod 256$$

replace Si and Sj

$$t = (S_i + S_j) \bmod 256$$

$$K = S_t$$

The open-source XOR operation is applied to byte K to obtain a code or with a code to obtain plaintext. Shifting is fast, about 10 times faster than when DES is used [6].

3.1. Encryption and decryption of RC4 algorithms

Encryption and decryption by the RC4 algorithm are identical. A pseudo-random number generator is used. An 8-bit pseudo-random number is generated for the 8-bit input, and then the XOR operation of those numbers is performed. The pseudocode of encryption and decryption is shown in Fig. 3.

```
S = RC4_initialize (key);
i = 0; j = 0;
While there is data in the input stream
{
        i = (i + 1) % 256;
        j = (j + S[i]) % 256;
        replace (&S[i], &S[j]);
        k = (S[i] + S[j]) % 256;
        exit = entrance XOR S[k];
}
```

Figure 3. *Pseudocode of encryption and decryption by RC4 algorithm [7]*

4. APPLICATION

4.1. Xampp

XAMPP is a completely free open source server package for easy installation of Apache servers on computers running Windows, Linux, or OSX. Xampp is intended for use in the local network not as a web server and is primarily used by developers to create a server to test their scripts. Xampp belongs to the family of WAMP software packages, where WAMP is an abbreviation of Windows, Apache, MySQL, and P can refer to PHP, Python, or Perl [8].

4.2. PHP programming language features

PHP is an open-source server-side programming language for dynamically generating HTML code. In other words, PHP is a programming language that can be used to create an HTML page on a server before it is sent to a client filled with dynamic content. In this way of generating the content, the client cannot see the code (script) that generated the content he is watching but has access to pure HTML code, Fig. 4.

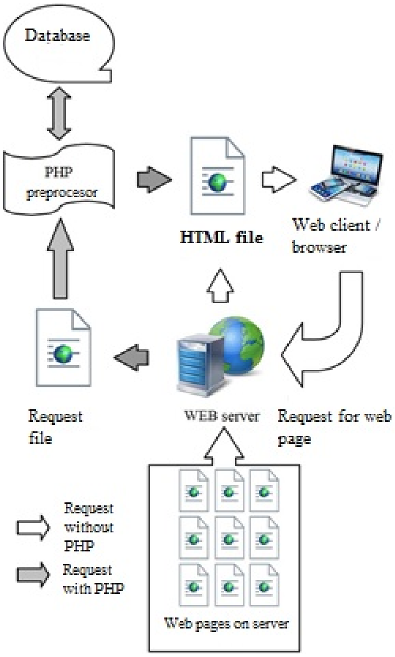


Figure 4. *Execution of dynamic HTML script by PHP program*

PHP is a server-side programming language and a very powerful tool for creating dynamic and interactive Web pages. PHP program code is executed on the server and the result of the execution is an HTML file that is sent to a Web browser. PHP files have a .php extension.

After receiving the request with the PHP document, the server executes the PHP code and based on it generates HTML code and sends it to the client. This means that the page displayed in the client's browser does not exist in that form anywhere on the server from where the client received it. This can create difficulties in positioning - ranking the created pages. In other words, PHP is a scripting language used to create an HTML page on a server before it, filled with dynamic the content, is sent to the client. In this way of generating content, the client cannot see the code (script) that generated the content he is watching but has access to pure HTML code [9].

4.3. Display results and code

After starting Xampp and within it the Apache web server, localhost is entered in the web browser, to access the local server and select the project. The application is launched by entering the following address in a web browser:

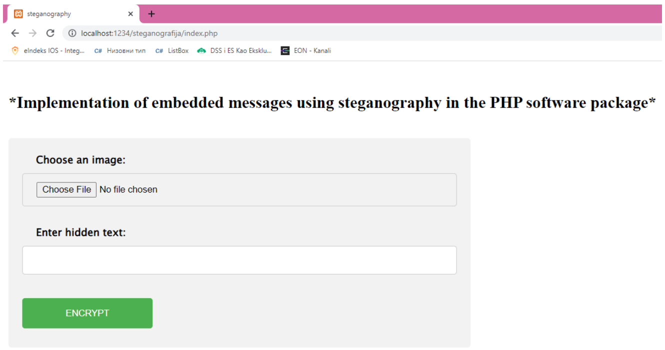
http://localhost:1234/steganografija/index.php

In addition to the main program: index.php, and Php-1, three more subroutines have been created, which are called from the main program.

These subroutines are:

- functions.php (Php-2),
- encrypt.php (Php-3) and
- descrypt.php (Php-4).

Fig. 5 shows the layout of the form, which was created in the Php-1 program code. By clicking on Choose File, the image in which the hidden message will be embedded is selected, while the desired hidden message is entered in the label and by clicking on the ENCRYPT button the message is placed inside the imported image, as shown in Fig. 6.



Implementation of embedded messages using steganography in the PHP software package

Choose an image:

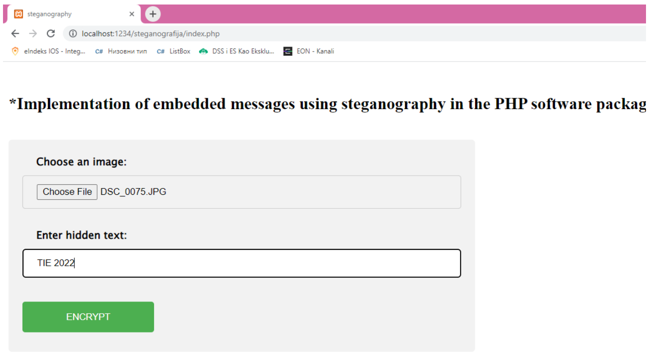
No file chosen

Enter hidden text:

Figure 5. *The appearance of the initial form*

Php-1.

```
<!DOCTYPE html>
<html>
<title> steganography </title>
<body>
<h2 align="left">*Implementation of embedded messages using steganography in the PHP software pack
<link rel="stylesheet" type="text/css" href="style.css" >
<div>
<form action="upload.php" method="post" enctype="multipart/form-data" >
<label class="label" for="fname">Choose an image:</label>
<input type="file" name="fileToUpload" id="fileToUpload"><br><br>
<label class="label" for="fname"> Enter hidden text:</label>
<input type="text" name = "tekst" id = "tekst">
<br><br>
<input type="submit" value="ENCRYPT" name="submit">
</form>
</div>
</body>
</html>
```



***Implementation of embedded messages using steganography in the PHP software packag**

Choose an image:

Choose File DSC_0075.JPG

Enter hidden text:

TIE 2022

ENCRYPT

***Figure 6**. Select an image and enter the desired message*

The ENCRYPT button uses the Php-3 code (encrypt.php), which includes the Php-2 program code (function.php) and thus opens the PHP image page that contains the hidden message.

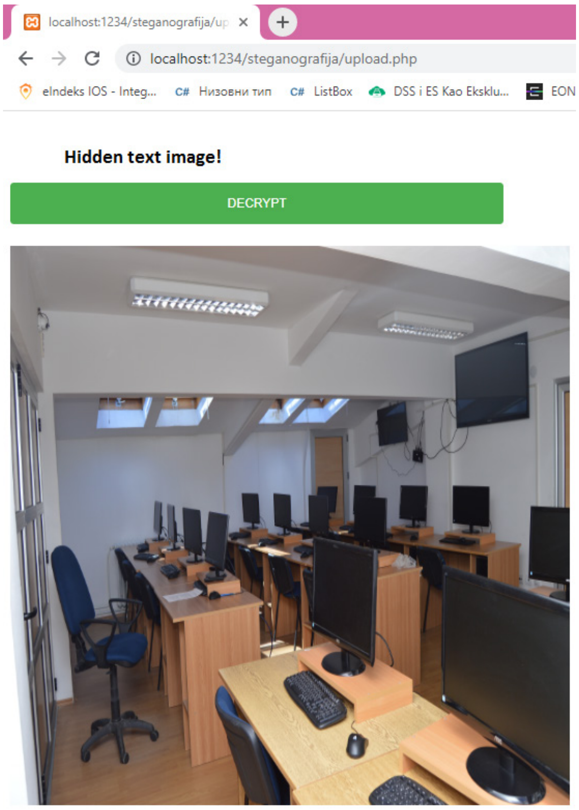
Php-3.

```
<link rel="stylesheet" type="text/css" href="style.css">
<?php
include('functions.php');
$msg = 'Test KRIPTOGRAFIJA.';
$src = 'DSC_0075.jpg';
$msg .='|';
$msgBin = toBin($msg);
$msglenght = strlen($msgBin);
$img = imagecreatefromjpeg($src);
list($width, $height, $type, $attr) = getimagesize($src);
if($msgLength>($width*$height)){
  echo('Message too long. This is not supported as of now.');
  die();
}
$pixelX=0;
$pixelY=0;
for($x=0;$x<$msgLength;$x++){
  if($pixelX === $width+1){
    $pixelY++;
    $pixelX=0;
  }
if($pixelY===$height && $pixelX===$width){
    echo('Max Reached');
    die();
  }
$rgb = imagecolorat($img,$pixelX,$pixelY);
  $r = ($rgb >>16) & 0xFF;
  $g = ($rgb >>8) & 0xFF;
  $b = $rgb & 0xFF;
$newR = $r;
  $newG = $g;
  $newB = toBin($b);
  $newB[strlen($newB)-1] = $msgBin[$x];
  $newB = toString($newB);
 $new_color = imagecolorallocate($img,$newR,$newG,$newB);
  imagesetpixel($img,$pixelX,$pixelY,$new_color);
  $pixelX++;
}
$randomDigit = rand(1,9999);
imagepng($img,'result' . $randomDigit . '.png');
echo('done: ' . 'result' . $randomDigit . '.png');
imagedestroy($img);
?>
```

Php-2.

```
<?php
function toBin($str)
{
$str = (string)$str;
    $l = strlen($str);
    $result = '';
    while($l--)
{
$result = str_pad(decbin(ord($str[$l])),8,"0",STR_PAD_LEFT).$result;
}
return $result;
}
function toString($str)
{
   $text_array = explode("\r\n", chunk_split($str, 8));
   $newstring = '';
for ($n = 0; $n < count($text_array) - 1; $n++)
{
       $newstring .= chr(base_convert($text_array[$n], 2, 10));
 }
   return $newstring;
}
```

After selecting an image, called slika.jpg, a PHP page opens with an image that implements hidden text that is not visible to the human eye, Fig. 7. An RGB color system is used in which each of these colors (red, blue, and green) has an 8-bit channel. The LSB method changes the least significant bits for each channel, which means that it is possible to hide 3 bits per image element. For each bit of the message, it is determined in which image element it will be inserted.



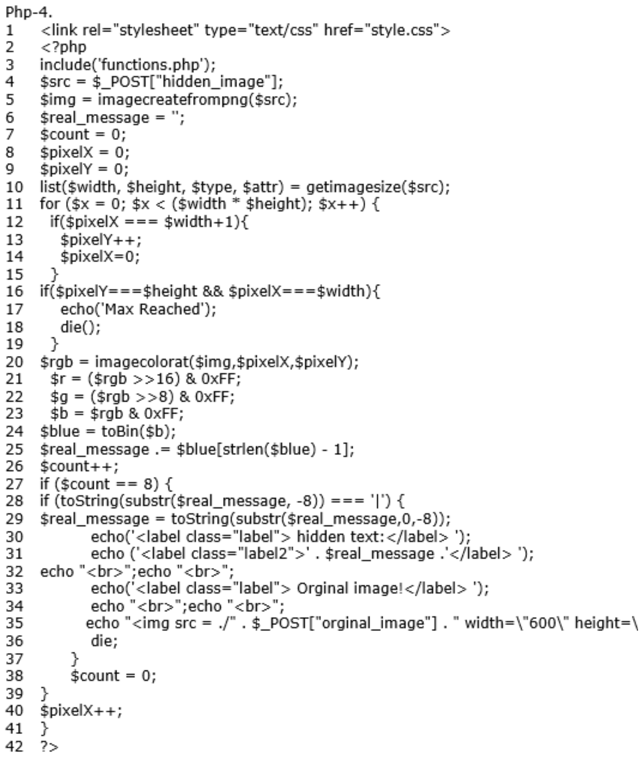
localhost:1234/steganografija/up
localhost:1234/steganografija/upload.php
eIndeks IOS - Integ...
Низовни тип
ListBox
DSS i ES Kao Eksklu...
EON

Hidden text image!

DECRYPT

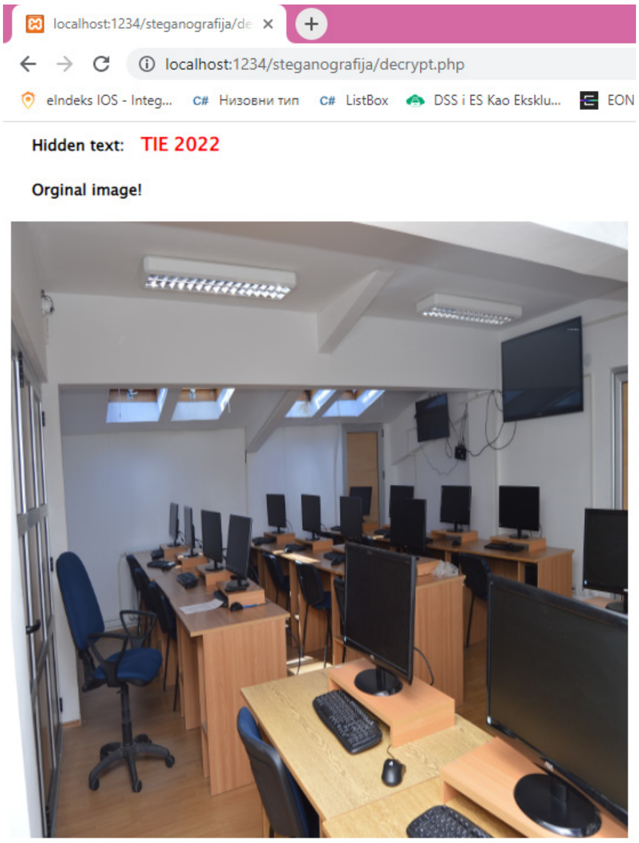
***Figure 7**. Showing an image with a hidden message*

The Php-4 code (decrypt.php) and the Php-2 code (function.php) allow you to click the DECRYPT button to display the decrypted message on the new PHP page, as well as the original image, shown in Fig. 8.



Php-4.

```
<link rel="stylesheet" type="text/css" href="style.css">
<?php
include('functions.php');
$src = $_POST["hidden_image"];
$img = imagecreatefrompng($src);
$real_message = '';
$count = 0;
$pixelX = 0;
$pixelY = 0;
list($width, $height, $type, $attr) = getimagesize($src);
for ($x = 0; $x < ($width * $height); $x++) {
  if($pixelX === $width+1){
    $pixelY++;
    $pixelX=0;
  }
if($pixelY===$height && $pixelX===$width){
    echo('Max Reached');
    die();
  }
$rgb = imagecolorat($img,$pixelX,$pixelY);
  $r = ($rgb >>16) & 0xFF;
  $g = ($rgb >>8) & 0xFF;
  $b = $rgb & 0xFF;
$blue = toBin($b);
$real_message .= $blue[strlen($blue) - 1];
$count++;
if ($count == 8) {
if (toString(substr($real_message, -8)) === '|') {
$real_message = toString(substr($real_message,0,-8));
          echo('<label class="label"> hidden text:</label> ');
          echo ('<label class="label2">' . $real_message .'</label> ');
echo "<br>";echo "<br>";
          echo('<label class="label"> Orginal image!</label> ');
          echo "<br>";echo "<br>";
         echo "<img src = ./" . $_POST["orginal_image"] . " width=\"600\" height=\
          die;
      }
      $count = 0;
}
$pixelX++;
}
?>
```



localhost:1234/steganografija/de
localhost:1234/steganografija/decrypt.php
eIndeks IOS - Integ...
C# Низовни тип
C# ListBox
DSS i ES Kao Eksklu...
EON

Hidden text: TIE 2022

Orginal image!

Figure 8. Display of the original image and hidden text

5. CONCLUSION

In the last few years, steganography has been the main topic of many discussions related to its abuse. For this reason, many legal bodies have raised concerns about the use of steganography to exchange illegal material through digital multimedia content on websites.

Particularly interesting are steganographic systems that use encrypted messages, which further improves the security of the system, even if the message is separated, it is still encrypted and incomprehensible to the attacker.

In this paper, the existing steganographic systems that use JPEG images as stego media and transmit encrypted, secret messages through them are analyzed.

Also, the paper presents codes for the appearance of the website, as well as codes for hiding and showing a secret message, which is not visible to the human eye.

REFERENCES

[1] Takashi, M., (2020) Misdirection steganography, *Soft Computing*, 24:16005–16010.

[2] Harianto, A., & Prasad, P. W. C. & Abeer, A., (2019), Implementation of cryptography in steganography for enhanced security Multimedia Tools and Applications, 78:32721–32734.

[3] Mukesh, D., Mamta, J., (2021), A survey on information hiding using video steganography Artificial Intelligence Review, 54:5831–5895.

[4] Veinović, M., Adamović, S., (2020) Kriptologija I, Univerzitet Singidunum, Beograd.

[5] Chandramouli, R., Subbalakshmi, K., (2004), Current trends in steganalysis: a critical survey, Control, Automation, Robotics, and Vision Conference, ICARCV 2004, Svezak 2, Str. 964-967.

[6] Bruce, S., (2007), Primenjena kriptografija: protokoli, algoritmi i izvorni kod na jeziku C, Mikro knjiga, Beograd.

[7] Zeljković, S., (2005), Steganografija, Hrvatski matematički elektronski časopis math.e, Broj 5, Lipanj; Dostupno na: http://e.math.hr/old/stegano/index.html

[8] Gautam, R., (2016), Analysis and implementation of WHOIS domain lookup. International Journal of Technical Research & Science.

[9] Julie C., M., (2018), PHP, MySQL i JavaScript, IV izdanje, Kompjuterska biblioteka, Beograd.

[10] Katzenbeisser, S., Petitcolas, F., (2020), Information Hiding Techniques for Steganography and Digital Watermarking, Arch House, Boston.

[11] Curran, K. Bailey, K., (2003), An evaluation of image-based steganography methods, International Journal of Digital Evidence.

[12] Ibrahim, A., (2007), Steganalysis in Computer Forensic, School of Computer and Information Science, Edith Cowan University.

[13]https://www.hashbangcode.com/article/steganography-images-php