A ESTEGANOGRAFIA EM IMAGENS DIGITAIS: UM PROCESSO DE GERAÇÃO E CODIFICAÇÃO PARA PROGRAMA DE COMPUTADOR

STEGANOGRAPHY IN DIGITAL IMAGES: PROCESS OF GENERATION AND CODING TO SOFTWARE

José Gladistone Rocha

RESUMO

A esteganografia é um ramo da criptologia que tem a finalidade de ocultação de mensagens em arquivos de imagem. Existem vários métodos de ocultação de mensagens, sendo a Least Significant Bit (LSB) ou Bit Menos Significativo uma delas e adotada para utilização por este trabalho. O objetivo aqui é apresentar o processo de criação de imagens esteganografadas por LSB; realizar sua posterior codificação em programa de computador para que seja possível averiguar sua eficiência. Espera-se com isso que possa se ter um conhecimento dessa técnica para seu aprimoramento futuro. Para se alcançar esse objetivo realizou-se uma compilação de trabalhos na literatura e posteriormente abrangeu-se a técnica do método LSB. O resultado obtido nesse experimento é que foi definido o processo para o método por LSB e sua posterior codificação em linguagem pascal.

Palavras-chave: Esteganografia; Segurança da informação; Processo de geração; Criptografia; Processo de ocultamento.

ABSTRACT

Steganography is a branch of cryptology for the purpose of embedded messages in image files. There are several methods of hiding messages, being the Least Significant Bit (LSB) one of them and adopted for use by this work. The goal here is to present the process of creating steganographed images by LSB, carrying out their subsequent encoding in a software so that it is possible to verify their efficiency. It is hoped that this way you can have a knowledge of this technique for future improvement. In order to achieve this objective, a compilation of works in the literature was carried out and, later, the technique of the LSB method was covered. The result obtained in this experiment is that the process for the method was defined by LSB and its subsequent encoding in Pascal language.

Keywords: *Steganography; Information security; Generation process; Cryptography; Concealment process.*

Introdução

O assunto segurança é uma preocupação constante entre os profissionais de tecnologia da informação, usuários comuns e empresas de diversos seguimentos. A Segurança da Informação (SI) é adotar controles físicos, tecnológicos e humanos personalizados, que viabilizem a redução e administração dos riscos, levando a empresa a atingir o nível de segurança adequado ao seu negócio (SÊMOLA, 2003, p.35) apud (AZEVEDO, FAVERI e NUNES, 2015).

A esteganografia é um dos ramos da criptologia. Trata-se de uma palavra de origem grega que significa a arte da escrita escondida (estégano = esconder e grafia

= escrita). É a arte da ciência da comunicação, na qual o objetivo é esconder dentro de outro arquivo aparentemente inofensivo alguma mensagem (AZEVEDO, FAVERI e NUNES, 2015), (FERREIRA, 2021). assim a esteganografia se propõe a esconder uma informação em uma imagem de cobertura. Essa poderosa tecnologia pode auxiliar na proteção e privacidade.

O uso de esteganografia em *software* tem um grande potencial, pois pode esconder dados em uma infinidade de mídias. Nas técnicas que utilizam o último bit de um byte para esconder mensagens, uma mensagem de 64Kbytes pode ser escondida em uma figura de 1024 x 1024 em tons de cinza ou imagens coloridas. Esta e outras novas técnicas, representam o estado da arte da esteganografia atual (JÚLIO, BRAZIL e ALBUQUERQUE 2007).

As aplicações de esteganografia incluem identificação de componentes dentro de um subconjunto de dados, legendagem (*captioning*), rastreamento de documentos e certificação digital (*time-stamping*) e demonstração de que um conteúdo original não foi alterado (*tamper-proofing*) (JÚLIO, BRAZIL e ALBUQUERQUE 2007).

Para o processo de geração da esteganografia em imagens digitais existem várias técnicas, como o *Least Significant Bit* ou Bit menos significativo, Filtragem e Mascaramento, Algoritmos e Transformações, Transformada de cosseno discreta, espelhamento de espectro e outras. Para efeito desse trabalho, a técnica a ser utilizada será a LSB.

Ante o exposto, este artigo visa apresentar aspectos da esteganografia e seu processo de funcionamento em imagens digitais. Para consecução deste objetivo, percorrem-se os dois passos mencionados a seguir. Inicialmente, realiza-se uma compilação de trabalhos recentes da literatura que tratam do assunto e em seguida faz-se a apresentação do processo e codificação da esteganografia. As principais contribuições advindas deste trabalho são permitir identificar o processo de execução da esteganografia como forma de ocultar mensagens em imagens digitais, reconhecer a codificação que envolve o mesmo e identificar se a técnica empregada é eficiente. Deixou-se de propósito os vários componentes de armazenamento na tela principal do programa para que fosse possível enxergar as várias etapas de processamento do método.

Para a demonstração do experimento de esteganografia foi utilizado o Windows 10 como Sistema Operacional e o ambiente *Integrated Development Environment* (IDE) ou Ambiente de Desenvolvimento Integrado na plataforma Lazarus com linguagem em Pascal para a criação do programa que realizará o processo de esteganografia e seu reverso em apresentar a mensagem em texto claro.

O restante deste texto está organizado da seguinte forma. A Seção 2 traz a compilação crítica de trabalhos da literatura sobre o tema em foco. A Seção 3 traz o detalhamento do processo de esteganografia e extração de mensagem. A Seção 4 trata das conclusões finais e de direcionamentos para trabalhos futuros.

2 Compilação de trabalhos da literatura

Há técnicas que são baseadas na modificação dos bits menos significativos (*Least Significant Bit*) dos valores de *pixel* no domínio espacial. Em uma implementação básica, estes *pixels* substituem o plano LSB inteiro com o stego-

dados (mensagem a ser oculta). Com esquemas mais sofisticados em que locais de inclusão são adaptativamente selecionados, dependendo de características da visão humana, até uma pequena distorção é aceitável. Em geral, a inclusão de LSB simples é suscetível a processamento de imagem, especialmente a compressão sem perda (JÚLIO, BRAZIL e ALBUQUERQUE, 2007).

Azevedo, Faveri e Nunes (2015) apresentam um processo de realização da esteganografia utilizando a web para interação com o usuário. O processo envolve: carregar imagem; definir a mensagem a qual será oculta; início da etapa de esteganografia e opção de fazer download da imagem processada. Há duas opções de compressão de arquivos ou dois tipos de algoritmos usados para tal, chamados de *Lossy* e *Lossless*. Na opção *Lossy* ocorre perda de dados, ou seja, reduz-se a quantidade de espaço que um arquivo consome. Já a compressão *Lossless*, consiste em reduzir o espaço de armazenamento exigido para informação digital. O algoritmo de compressão *Lossless* trabalha sem perda de dados.

Estevam (2017) apresenta um estudo de métodos e conceitos de esteganografia e criptografia como formas eficazes de garantir a inviolabilidade da informação. O uso de criptografia e esteganografia são requisitos de segurança, porém a união destas duas técnicas resulta no aumento da segurança dos dados. O objetivo do seu trabalho foi desenvolver uma ferramenta utilizando a técnica de esteganografia LSB em imagens, com o método de criptografia simétrica com o algoritmo DES (*Data Encryption Standard*). O autor detalha como funciona, em linhas gerais, a técnica LSB porém não entra em detalhes como tal técnica é implementada em programa de computador.

Vicentini et al. (2017) realizaram um estudo que teve como objetivo utilizar programas para ocultação de mensagens em imagens. Foram utilizados os programas *Jpeg Hide and Seek* (JPHS), *Camouflage*, *OpenPuff*, *Steganography-Tools* (S-Tools) para identificar qual *software* obteve a melhor performance. Concluíram que as ferramentas *Camouflage* e *OpenPuff*, obtiveram maior destaque se comparadas com as ferramentas JPHS e S-Tools.

Eduardo e Imamura (2019) apresentaram um trabalho que teve por proposta combinar a teoria da esteganografia com o processamento de imagens, a fim de contribuir para a computação forense na detecção da esteganografia. A estratégia adotada consiste em converter uma imagem em uma tabela de valores, onde cada valor representa uma cor da imagem e verificar se possui alteração entre os elementos, sabendo-se que geralmente os elementos vizinhos tendem a ter o mesmo valor de cor. Assim, se fosse possível verificar a alteração de cores na vizinhança de um certo elemento, possivelmente a imagem se tornaria suspeita e passiva de investigação.

Couto, Ferreira e Madeiro (2020) desenvolveram um aplicativo web que utiliza a técnica LSB até a MSB para ocultação da mensagem com a finalidade de apresentar aos alunos do curso de Engenharia e assim motivá-los a não evasão. Foi aplicado questionário aos alunos e 100% responderam estarem motivados com o curso. Essa técnica utilizada é a mesma empregada nesse trabalho.

Ferreira (2021) abordou estudo em que a esteganografia é aplicada em imagens digitais, bem como estudos que apresentam as ferramentas que são comumente utilizadas para a extração e/ou identificação de informações ocultadas neste tipo de arquivo. Em relação ao processo investigativo no ambiente virtual, buscou-se ressaltar os processos da perícia digital indicados nos estudos

disponíveis, assim como, os *softwares* e ferramentas utilizadas na solução dos crimes envolvendo mídias de armazenamento na área de informática.

Ainda há muitos outros trabalhos relacionados ao que se propõe nesse artigo que deixaram de serem apresentado por questões de espaço mas que pode-se apontar alguns como: Adonias et al. (2017); Almeida, Neto e Aquino (2017) e Silva, Carvalho e Martins (2020).

3 O Processo de geração e cifração da esteganografia

Essa Seção trata do processo de geração da esteganografia e da codificação do processo em programa de computador. A ideia é demonstrar o processo e apresentar os códigos fontes que atendem a esse processo.

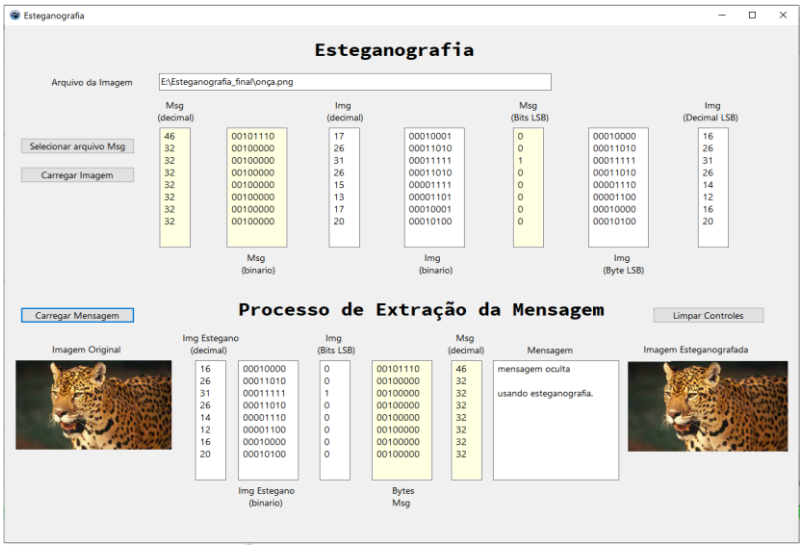
3.1 O Processo de geração da esteganografia

Há vários métodos de esteganografia para ocultação da mensagem, desde a inclusão de chaves criptográficas como o uso de criptografia durante as etapas do processo de cifração da esteganografia. O método que será apresentado neste trabalho é a esteganografia por meio de LSB.

O método de esteganografia LSB visa substituir o bit menos significativo, como por exemplo, de cada uma das três cores que formam um *pixel* (RGB – Red (Vermelho), Green (Verde) e Blue (Azul)), assim cada *pixel* aceita até 3 bits de informação; desse modo, cada imagem consegue armazenar até 3 vezes o número de *pixels* que possui (DEMACDOLINCOLN, 2013) apud (AZEVEDO, FAVERI e NUNES, 2015). O método aplicado garante a privacidade por meio da confidencialidade, como também a integridade por meio da utilização de cifras do tipo *HASH* para verificação da diferença de conteúdo de arquivos esteganografados (AZEVEDO, FAVERI e NUNES, 2015).

A Figura 1 apresenta a tela do programa, onde apresenta vários componentes do tipo lista a serem utilizados no processo de esteganografia, além de apresentar o processo utilizado na codificação do programa para gerar a Esteganografia e seu processo inverso de retirada da mensagem oculta.

Figura 1: Tela principal do programa Esteganografia



Fonte: o autor

Para efeito desse exemplo apresentado na Figura 1 foi utilizada uma imagem (onça.png) de tamanho igual a 127 Kbytes e como resultado da esteganografia da imagem resultou-se uma imagem de tamanho igual a 126 Kbytes.

O Quadro 1 apresenta o significado de cada uma dessas listas apresentas na Figura 1 para um melhor entendimento do código do programa, mais adiante.

Quadro 1: descrição dos componentes visuais da tela do programa

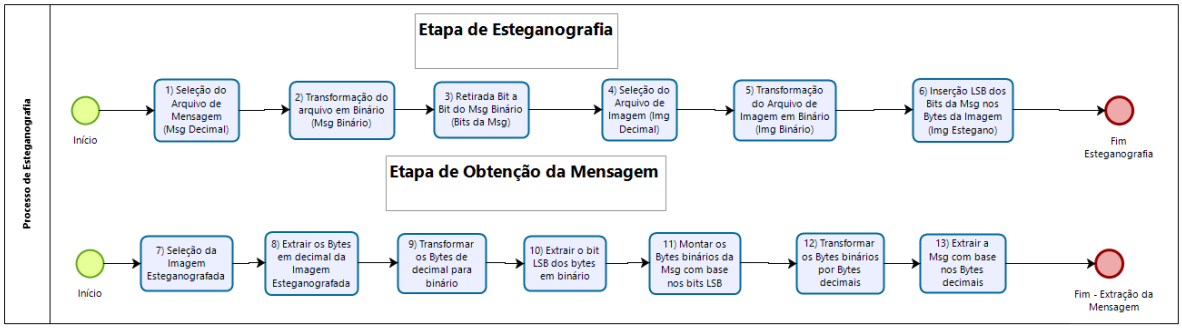
Componentes visuais (listas)	Descrição
Oriundos do processo de Esteganografia	
Msg (decimal)	Lista para armazenar os bytes em decimal da mensagem a ser oculta.
Msg (binário)	Lista para armazenar os bytes binários oriundos dos bytes decimal da mensagem.
Img (decimal)	Lista para armazenar os bytes, em decimal, da imagem.
Img (binário)	Lista para armazenar a transformação dos bytes decimal em bytes binários da imagem.
Msg (Bits LSB)	Lista para armazenar a extração, bit a bit, de todos os bytes binários da imagem.
Msg (Bytes LSB)	Lista para armazenar os bytes binários oriundos do processo de substituição LSB da imagem.
Img (Decimal LSB)	Lista para armazenar a transformação dos bytes binários em bytes decimal da Imagem.
Oriundos do processo de obtenção da mensagem oculta	
Img Estegano (decimal)	Lista para armazenar os bytes em decimal oriundos do arquivo de imagem esteganografado.
Imagem Estegano (binário)	Lista para armazenar a transformação dos bytes decimal em bytes binário do arquivo de imagem esteganografado.
Img Bits LSB	Lista para armazenar os bits LSB oriundos, bit a bit, dos bytes binário do arquivo de imagem esteganografado.
Bytes Msg	Lista para armazenar os bytes binários oriundos da junção de oito em oito bits da lista LSB para formar os bytes binários da mensagem.
Msg (decimal)	Lista para armazenar os bytes em decimal oriundos da lista de Bytes Msg.
Mensagem	Apresenta o texto em claro da mensagem oculta na imagem.

Fonte: o autor

O processo utiliza a técnica de LSB associada aos bytes da cor Azul do *Red-Green-Blue* (RGB) da imagem. A aplicação da técnica incide a partir do primeiro *pixel* da imagem nos bytes de sua cor azul e se estende até o tamanho de bytes do arquivo de mensagem a ser oculta. O processo poderia usar qualquer cor (verde e/ou vermelho) ou a combinação de todas, mas para efeito desse programa utilizou-se os bytes da cor azul.

A Figura 2 apresenta o processo de esteganografia e obtenção da mensagem oculta.

Figura 2: processo de esteganografia e obtenção de mensagem oculta



Fonte: o autor

Para dar continuidade na descrição do processo segue-se os seguintes passos como indicado na Figura 2 e apresentado no Quadro 2 onde mostra toda a etapa de esteganografia:

Quadro 2: descrição das etapas do processo de esteganografia LSB

1) **Seleção de Arquivo de Mensagem (Msg Decimal)**: trata-se da escolha do arquivo texto que contem a mensagem e a extração dos bytes em decimal do arquivo da mensagem (*embedded data*) – seu tamanho não pode exceder a 30% do tamanho da imagem a ser esteganografada, pois do contrário ocasionaria distorções visíveis a olho nu;
2) **Transformação do arquivo em binário (Msg binário)**: essa etapa os bytes em decimal são convertidos para bytes em binário do arquivo da mensagem;
3) **Retirada bit a bit do Msg binário (Bits da Msg)**: é criada uma lista contendo todos os bits de cada byte da mensagem – são todos colocados em ordem, um a um;
4) **Seleção do arquivo de imagem (Img Decimal)**: nessa etapa o usuário seleciona um arquivo de imagem (*cover-image*), também chamado de arquivo de cobertura, e submete ao programa. Este por sua vez extrai todos os bytes, em decimal, da imagem a ser trabalhada;
5) **Transformação do arquivo de imagem em binário (Img Binário)**: transformação de cada byte decimal em bytes binário do arquivo de imagem;
6) **Inserção LSB dos bits da Msg nos bytes da imagem (Img Estegano)**: retira-se de cada byte binário o último bit, ou bit menos significativo LSB e insere-se o bit oriundo da mensagem em binário – todos os bits da mensagem são inseridos nesse processo, um a um dentro de cada byte binário da mensagem. O resultado desse processo é uma lista de bytes da imagem inseridos os bits da mensagem no seu último bit ou bit menos significativo. Ao final dessa etapa tem-se a imagem esteganografada ou estego-objeto.

Fonte: o autor

Agora segue-se os seguintes passos como indicado na Figura 2 e apresentado no Quadro 3 onde mostra toda a etapa de obtenção da mensagem:

Quadro 3: Descrição das etapas do processo de esteganografia

7) **Seleção da imagem esteganografada**: o usuário seleciona a imagem esteganografada;
8) **Extrair os bytes em decimal da imagem esteganografada**: o programa extrai cada byte em decimal e inclui os bytes na lista respectiva;
9) **Transformar os bytes de decimal para binário**: o programa pega cada byte decimal da lista e transforma em binário e armazena na devida lista;
10) **Extrair o bit LSB dos bytes em binário**: retira o último bit de cada byte em binário da imagem;

11) **Montar os bytes binários da Msg com base nos bits LSB**: para cada 8 bits LSB da imagem em binário é montado cada byte binário da mensagem – esse processo se repete até que se obtenha todos os bytes em binário da mensagem;
12) **Transformar os bytes binários por bytes decimais**: nessa etapa todos os bytes binários da mensagem são transformados em bytes decimais e armazenados em sua devida lista (ver Quadro 1);
13) **Extrair a Msg com base nos bytes decimais**: essa última etapa realiza a transformação dos bytes decimais da mensagem em caracteres ASCII e apresenta na lista de mensagem.

Fonte: o autor

3.2 O Processo de cifração da esteganografia

Essa Seção apresenta a codificação em Pascal de cada etapa do processo descrito na Figura 2 e Quadro 1 para realizar a esteganografia e seu processo reverso ou obtenção da mensagem.

O processo de cifragem e decifragem do texto da mensagem oculta é de responsabilidade do programa que realizou a esteganografia e seu reverso, sendo assim, tal programa deve ser protegido por meio de criptografia simétrica ou assimétrica. Isso garante uma duplicidade no ocultamento da mensagem esteganografada.

A maior contribuição desse método de ocultação de mensagens está na sua utilização para comunicações confidenciais por canais ou ambientes não seguros. O processo de embutir e retirar a mensagem na figura é feita de forma simples, possibilitando que usuários com pouco conhecimento possam garantir a integridade das informações (AZEVEDO, FAVERI e NUNES, 2015).

A seguir são apresentadas as telas de código do programa. As partes mais importantes do código foram comentadas para facilitar o entendimento e a sequência de execução do programa.

A Figura 3 apresenta a codificação do botão “Selecionar Arquivo Mensagem”. Destaca-se as seguintes linhas de comando: 250 – armazena a imagem em decimal; 252 – armazena a imagem em binário; e 252 – armazena os bits LSB da mensagem.

Figura 3: tela do código do botão Selecionar Arquivo Mensagem

```
  procedure TFMenu.btnarqselecionarClick(Sender: TObject);
  var
    k:integer;
235 aByte : string;
  begin
237 if(odImg.Execute)then
    edcaminho.Text:=odImg.FileName;
  ArquivoMsg:=TFileStream.Create(odImg.FileName,fmOpenRead);
240 lbArquivo.Caption:='Arquivo da Mensagem';
  //showmessage('nome do arquivo = '+edcaminho.Text);
  ArquivoMsg.Position:=0;
  tamanhoMsg:= ArquivoMsg.Size;
  //showmessage(' tamanho do arquivo de mensagem: '+Inttostr(tamanhoMsg));
245 for i:=0 to tamanhoMsg -1 do
    begin
        b := ArquivoMsg.ReadByte;
        aByte:=DecToBin(b);
        // armazena a Msg em decimal
250     MemoMsgDec.Lines.Add(IntToStr(b));
        // armazena a Msg em binário
        MemoMsgBin.Lines.Add(aByte);
        for k:=1 to 8 do
          begin
255         // armazena os Bits LSB da Msg
            MemoMsgBits.Lines.Add(copy(aByte,k,1));
          end;
    end;
  btCarregarImagem.Enabled:= true;
260 ArquivoMsg.Free;
  end;
```

Fonte: o autor

A Figura 4 apresenta a codificação do botão “Carregar Imagem”. Destaca-se a seguintes linhas de comando, conforme apresentado no Quadro 4.

Quadro 4: Descrição dos comandos do botão Carregar Imagem

Linha	Descrição do comando
144	Inicializa-se o arquivo lógico de imagem ArqImg por meio da Classe pascal Tpicture;
145	Inicializa-se o arquivo lógico de imagem Img por meio da Classe pascal TPicture;
147	Carregamento da imagem selecionada para o arquivo lógico Img;
152	Obtém a cor do pixel de cada byte da imagem;
153	Obtém a cor azul do pixel da imagem de cada byte;
155	Armazena a imagem em bytes decimais;
157	Armazena a imagem em bytes binários;
162	Troca o bit LSB da imagem por bit da mensagem;
164	Armazena os bytes em binários esteganografados da imagem;
166	Armazena os bytes esteganografados em decimal;
176	Atribui a cor azul os bytes esteganografados da imagem;
181	Grava o arquivo de imagem esteganografado.

Fonte: o autor

Figura 4: Tela do Código do botão Carregar Imagem

```
.   procedure TFMenu.btCarregarImagemClick(Sender: TObject);
.   begin
144   ArqImg := TPicture.Create;
145   Img :=TPicture.Create;
.     if(opdImagem.Execute)then
.       Img.LoadFromFile(opdImagem.FileName);
.     h:=Img.Height;
.     w:=Img.Width;
150   for i:=0 to tamanhoMsg*8 - 1 do
.     begin
.       cor:=Img.Bitmap.Canvas.Pixels[i mod w,i div w];
.       b := Blue(cor);
.       // armazena a Img em decimal
155     MemoImgDec.Lines.Add(Inttostr(b));
.       // armazena a Img em binário
.       MemoImgBin.Lines.Add(DecToBin(b));
.     end;
.     for i:=0 to tamanhoMsg*8 - 1 do
160   begin
.       // troca o bit LSB da Img
.       Binario := TrocaBit(MemoImgBin.Lines[i],MemoMsgBits.Lines[i]);
.       // armazena a Img com Bytes LSB
.       MemoImgLSB.Lines.Add(Binario);
165     // armazena Img com Bytes LSB em decimal
.       MemoImgDecLSB.Lines.Add(IntToStr(BinToDec(Binario)));
.     end;
.     Img.Free;
.     ArqImg.LoadFromFile(opdImagem.FileName);
170   for i:=0 to tamanhoMsg*8 - 1 do
.     begin
.       cor:=ArqImg.Bitmap.Canvas.Pixels[i mod w, i div w];
.       r:=Red(cor);
.       g:=Green(cor);
175     // atribui a cor Blue os Bytes LSB em decimal da Img
.       b:= MemoImgDecLSB.Lines[i].ToInteger;
.       cor:=RGBToColor(r,g,b);
.       // arquivo de Img recebe a cor alterada
.       ArqImg.Bitmap.Canvas.Pixels[i mod w, i div w]:= cor;
180   end;
.     ArqImg.PNG.SaveToFile('imagem_esteganografada.png');
.     ImagemEsteganografada.Picture.LoadFromFile('imagem_esteganografada.png');
.     ArqImg.Free;
.     sFilename:=opdImagem.FileName;
185   edcaminho.Text:=sFilename;
.     lbArquivo.Caption:='Arquivo da Imagem';
.     Imagem.Picture.LoadFromFile(sFilename);
.     btCarregaMsg.Enabled:= true;
.   end;
```

Fonte:os autores

A Figura 5 apresenta a codificação do botão “Carregar Mensagem”. Serão apresentadas as descrições das principais linhas de código para um melhor entendimento do processo codificado. Destaca-se as seguintes linhas de comando conforme apresentado no Quadro 5:

Quadro 5: Descrição dos principais comandos do botão Carregar Mensagem

Linha	Descrição do comando
206	Armazena os bytes em decimal da imagem esteganografada;
208	Transforma os bytes decimais em binários e armazena os bytes em binário da imagem esteganografada;
210	Retira cada bit LSB dos bytes binários da imagem esteganografada e armazena numa lista;
212	Monta os bytes binários da mensagem com base nos bits LSB da imagem extraídos na etapa anterior;
224	Armazena os bytes em decimal da mensagem;
226	Extrai a mensagem em texto claro.

Fonte: o autor

Figura 5: Tela do código do botão Carrega Mensagem

```
   procedure TfMenu.btCarregaMsgClick(Sender: TObject);
.  aByte : String;
.  begin
195   aByte:='';
.     aImagem:=TPicture.Create;
.     opdImagem.Title:='Escolher arquivo esteganografado';
.     if(opdImagem.Execute)then
.       aImagem.LoadFromFile(opdImagem.FileName);
200   w := aImagem.Width;
.     for i:=0 to tamanhoMsg*8 - 1 do
.     begin
.       cor:=aImagem.Bitmap.Canvas.Pixels[i mod w,i div w];
.       b := Blue(cor);
205     // armazena Img Estegano em decimal
.       MemoEsteganoDec.Lines.Add(Inttostr(b));
207     // armazena Img Estegano em binário
.       MemoEsteganoBin.Lines.Add(DecToBin(b));
.       // Armazena Bytes da Img Estegano sem os Bits LSB
210     MemoBitsLSB.Lines.Add(RetiraBitLSB(DecToBin(b)));
.       // Monta os Bytes da Msg com os Bits LSB
.       aByte:=aByte+MemoBitsLSB.Lines[i];
.       if aByte.Length = 8
.          then
215       begin
.           // armazena os Bytes da Msg sem os Bits LSB
.           MemoByteMsg.Lines.Add(aByte);
.           aByte:='';
.         end;
220   end;
.     for i:= 0 to tamanhoMsg - 1 do
.     begin
.       // Armazena os Bytes em decimal da Msg
.       MemoMsgDecimal.Lines.Add(IntToStr(BinToDec(MemoByteMsg.Lines[i])));
225     // Monta a Msg com os Bytes em decimal
.       MemoMensagem.Text:=MemoMensagem.Text + Chr(StrToInt(MemoMsgDecimal.Lines[i]));
.     end;
.  aImagem.Free;
.  end;
```

Fonte: o autor

Acredita-se que com a definição do processo de esteganografia e a sua consequente codificação, pode-se ter condições de realizar um estudo do programa visando seus melhoramentos para aplicações mais especializadas por meio de emprego das demais técnicas de esteganografia comentadas anteriormente.

4. Conclusões Finais e Trabalhos Futuros

Foi realizado um estudo onde se iniciou com a apresentação do processo de ocultação de mensagens em imagens usando a técnica de LSB. Em seguida foi codificado o processo em linguagem pascal onde pode-se observar como o processo seria executado em *software*.

Assim, pode-se aferir se a codificação atendia ao processo e ainda possibilitar fazer um estudo dessa codificação para melhorá-la ao ser executada.

Como trabalhos futuros sugere-se: i) escrever o código usando chave de deslocamento; ii) usar criptografia para inserção dos bits LSB na imagem, tornando o processo mais seguro; e iii) aplicar a codificação em outros formatos de imagens.

REFERÊNCIAS

ADONIAS, G. L., FARIAS, E. S., SANTOS, W, C., REGIS, C. D. M. (2017). Análise Objetiva do Número de Bits Menos Significativos em Esteganografia de Imagens Digitais. Revista Brasileira de Sistemas de Informação, Rio de Janeiro, vol. 10, No. 3, pp. 24-35.

ALMEIDA, W. D., NETO, P. S., AQUINO, F. J. A. (2017). Estudo Comparativo e Implementação de Técnicas Esteganográficas para Ocultamento de Informações. Revista de Tecnologia da Informação e Comunicação, Vol. 7, No. 2, Agosto.

AZEVEDO, E., FAVERI, J. G., NUNES, S. E. (2015). Esteganografia. Rev. Ciências Exatas Tecnologia, v. 10, n. 10, p. 31-35.

COUTO, I. S., FERREIRA, F. A. B. S., e MADEIRO, F. (2020). A esteganografia poderia ser utilizada para aumentar o engajamento dos estudantes de engenharia? Revista Novas Tecnologias na Educação, V. 18 Nº 1, julho.

DEMACDOLINCOLN. 2013. Esteganografando com o steghide.

EDUARDO, W. F. E., IMAMURA, C. Y. M. (2019). Desafios da Detecção de Esteganografia em Imagens Digitais Através de Análise de Vizinhança. 10° Congresso de Inovação, Ciência e Tecnologia do IFSP, 27 e 28 de nov – Sorocaba-SP.

FERREIRA, J. S. (2021). Computação Forense e a Técnica De Esteganografia Aplicada em Imagens Digitais: um mapeamento sistemático. obtenção do título de Bacharela em Engenharia de Software.

JULIO, E. P., BRAZIL, W. G., ALBUQUERQUE, C. V. N. (2007). Esteganografia e suas Aplicações. VII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais.

SÊMOLA, M. (2003). Gestão da segurança da informação: visão executiva da segurança da informação aplicada ao Security Officer. Rio de Janeiro: Elsevier.

SILVA, W. G., CARVALHO, R. L. e MARTINS, G. A. S. (2020). Steganography Genetic Algorithm Hyperparameter Tuning through Response Surface Methodology. ISSN: 2675-3588.

VICENTINI, F. R. S., Oliveira, H. C., Godoy, M. A., Martins, H. P. (2017). Técnicas de Segurança Aplicação de Esteganografia em Imagens. Faculdade de Tecnologia de Bauru-SP.