

Esteganografia em imagem baseada na técnica do bit menos significativo.

Stephane de Freitas Schwarz
Hélio Pedrini

Resumo – Por longos séculos a humanidade busca meios para proteger seus dados através de códigos e cifras. Uma das técnicas mais antigas para preservar o conteúdo de uma informação é a esteganografia, ela basicamente oculta um dado em outro. Neste trabalho, são abordados os conceitos de esteganografia em imagens nos pixel menos significativos, e.g. 2^0 2^1 2^2 .

Palavras chave – *Segurança da informação, esteganografia e ocultação de dados.*

I. INTRODUÇÃO

A esteganografia é uma técnica antiga para camuflagem de dados, foi registrada pela primeira vez no fim do século XIV por Johannes Trithemius. Ela pode ser aplicada em imagens, vídeos e arquivos de texto. Suas aplicações são bem variadas, contudo em geral é usada para transmitir uma informação sigilosa sem levantar suspeitas. Uma das vantagens da utilização dessa técnica é que, se bem elaborada, dificilmente uma pessoa notará qualquer nuance no documento alterado.

II. MÉTODO PROPOSTO

A solução consiste em, dada uma imagem RGB, uma mensagem a ser ocultada e a posição do bit da imagem a ser alterado, o algoritmo combina os valores binários da imagem com os valores da mensagem. Mais especificamente, cada bit do pixel da imagem em uma determinada posição (e.g. 2^0 , 2^1 , ..., 2^8) é trocado por um bit da mensagem, esse procedimento é feito sequencialmente, tanto na imagem quanto no texto. Suponha uma imagem RGB qualquer e uma mensagem T contendo a sigla IC, na ilustração 1 é possível verificar separadamente cada canal de cor da figura, originalmente os valores valem 255, ou seja, todos os bits em nível lógico alto, contudo, após a esteganografia os valores de bit da mensagem

substituíram o último bit da imagem, isso porque o plano de bit escolhido foi 7.

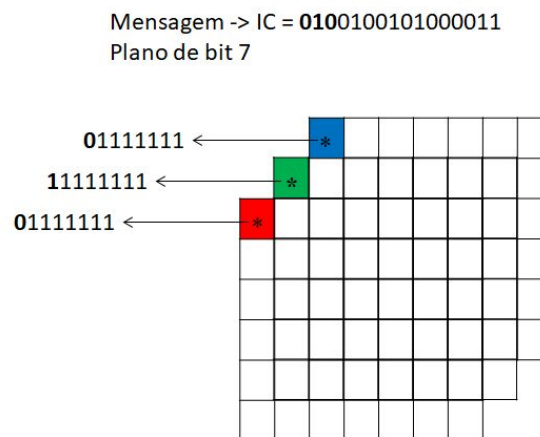


Figura 1: Ilustração do processo de esteganografia em imagem. Cada bit da mensagem substitui um bit da imagem na posição especificada pelo usuário. Neste caso, o bit mais significativo foi alterado.

O algoritmo foi desenvolvido seguindo alguns passos lógicos e simples, de modo geral, após o texto ser convertido para valores binários verifica-se o tamanho total da cadeia gerada, essa informação é fundamental para evitar a leitura de dados irrelevantes, para tanto, os primeiros 300 bits são reservados para armazenar o tamanho da mensagem. Outro ponto importante é certificar se o tamanho do texto não ultrapassa o tamanho da figura.

Uma vez que o ambiente está pronto basta rearranjá-lo, isso é, alocar o bit de dado na imagem. O modo mais simples de fazer isso é percorrendo toda a matriz alterando seus valores da imagem para o desejado como mostra o pseudocódigo abaixo.

1. para cada linha
2. para cada coluna
3. para cada dimensão
4. $im = im \& \sim(2^{**PB}) | BT \ll PB$.
5. # onde im representa um pixel da imagem.
6. # BT é um bit específico do texto
7. # e PB é o plano de bit

Para alterar um byte em uma posição específica é preciso manipular-lo de modo que as demais

posições não sejam comprometidas. Tomando como exemplo um byte de valor 187, sua representação binária é 10111011, para modificar apenas o terceiro bit é preciso fazer uma operação do tipo *and* entre o pixel e o valor invertido do bit correspondente, em seguida realizar uma operação *or* com o resultado obtido e o novo valor de bit. De maneira mais clara,

$$\begin{array}{rcl} 10111011 \wedge & & 10111011 \vee \\ \underline{11111011} \text{ (}\sim 2^2\text{)} & & \underline{00000100} \text{ (}1 \ll 2\text{)} \\ 10111011 & & 10111111 \end{array}$$

esse cálculo deve ser executado até o último bit da mensagem ser inscrita na imagem. O resultado final do processo de esteganografia no terceiro bit menos significativo pode ser observado na figura 2.

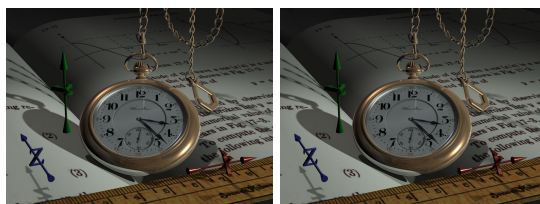


Figura 2: Imagem original (direita) e esteganografada (esquerda), neste caso a mensagem foi inserida no terceiro bit menos significativo.

Percebe-se uma pequena variação na iluminação da imagem, no entanto, é impossível constatar que há algo camuflado dentro dela analisando apenas tais figuras. Essa premissa contudo, não é verdadeira quando os bits alterados estão próximos do mais significativo, isso porque, como o nome sugere, são bits que contribuem consideravelmente no valor final do pixel. Na Ilustração 3, aplicando a esteganografia no sexto bit, é possível notar uma descontinuidade de textura e iluminação nas regiões que contém dados. E como o padrão de cores muda abruptamente, o sistema visual humano pode captar tais nuances sem grandes dificuldades.



Figura 3: Processo de esteganografia aplicado no bit de posição 5. Notória a presença de elementos ruidosos.

O processo inverso dessa técnica resulta na mensagem escondida dentro da imagem, entretanto identificar uma imagem esteganografada não é uma tarefa trivial, se o emissor foi cuidadoso no momento da elaboração do algoritmo provavelmente não haverá anomalias explícitas na figura, inviabilizando análise visual nas condições de recebimento.

Uma alternativa para identificar uma potencial imagem esteganografada é através da avaliação do plano de bits. Tomando como exemplo a figura 2, se o plano de bits dos pixel forem visualizados separadamente, é possível apontar uma região com anomalias acentuadas exatamente quando são apresentados os valores da posição do bit manipulado.

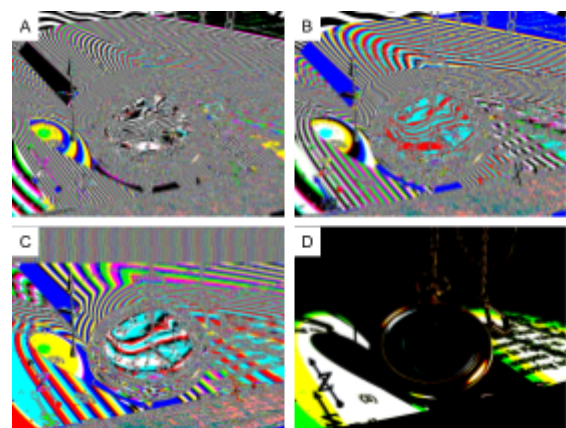


Figura 4: Exibe o comportamento dos valores dos bits em uma imagem esteganografada no bit de posição 2. A) plano de bit 0, B) plano de bit 1, C) plano de bit 2 e D) plano de bit 7.

Na figura 4.C nas primeiras linhas da imagem é evidente a presença de uma irregularidade pois os

bits deixam de seguir um padrão. Embora seja difícil ou impossível perceber essas variações na imagem inicial, quando os bits são plotados em um gráfico separadamente, dependendo do tamanho da mensagem (textos pequenos em geral não resultam em modificações expressivas), fica claro a presença de potenciais informações camufladas.

III. EXECUTAR O PROGRAMA

Abra o terminal ou prompt de comando e digite *python Ass#2_codificar.py imagem.png arquivo_texto.txt posicao_bit* para esteganografar uma imagem. Para decodificar a imagem digite *Ass#2_decodificar.py imagem.png posicao_bit*.

IV. REQUISITOS

O algoritmo foi construído para ser executado em qualquer computador desde que respeite as seguintes especificações: Python versão 3 mais bibliotecas *scipy*, *matplotlib* e *numpy*.

REFERÊNCIAS

[1] Introdução a esteganografia. [Online] Disponível em: <https://www.vivaolinux.com.br/artigo/Esteganografia-e-Esteganalise-transmissao-e-deteccao-de-informacoes-ocultas-em-imagens-digitais>