Centro Universitário de Brasília

Luiz Gustavo Lustosa Colombo

ESTEGANOGRAFIA – INSERÇÃO DE TEXTO EM IMAGEM GIF

Brasília/DF, junho de 2006

ESTEGANOGRAFIA – INSERÇÃO DE TEXTO EM IMAGEM GIF

Monografia apresentada ao Centro Universitário de Brasília – UniCEUB como um dos pré-requisitos para obtenção do título de Bacharel em Engenharia da Computação.

PROFº ORIENTADOR:
MC. Aderlon M. Queiroz

Brasília/DF, junho de 2006

Agradecimentos

A Deus,

Por sempre ter me dado força e bom ânimo.

Aos Meus pais,

Por me darem todo o suporte necessário para a realização dos meus sonhos.

A todos os Meus Familiares ,

Pela amizade e o incentivo.

Ao MC. Francisco Javier Obaldia,

Que me ajudou na escolha do projeto.

Ao meu orientador MC. Aderlon M. Queiroz,

Que me ajudou quando tive necessidade.

Aos meus amigos: Emanuel Cortes e Marco Antônio Soares

Que me ajudaram e incentivaram nesses tempos de faculdade.

A todos,

O meu gesto mais sincero de gratidão.

RESUMO

Este trabalho apresenta a implementação de um protótipo que faz a inserção de uma mensagem dentro de uma imagem do formato GIF, utilizando-se da técnica de esteganografia, tendo junto com ela o uso do algoritmo de Huffman, um algoritmo de compressão de texto.

Enfim, os resultados obtidos nas simulações com as imagens esteganografadas são mostrados através de histogramas.

Palavras chaves: esteganografia, bit menos significativo, algoritmo de Huffman, formato GIF.

ABSTRACT

This work presents the implementation of a prototype that inserts a message into a GIF image by using the technique of steganography. Together with this technique, there is the Huffman algorithm, an algorithm that works with text compression.

Finally, the results obtained in the simulations with the stego images are shown using histograms.

Keywords: steganography, least significant bit, Huffman Algorithm, GIF format.

Sumário

Lista de Figuras

Lista de Abreviaturas e Siglas

ASCII	”*American Standard Code for Information Interchange*”
BIT	“*Binary Digit*”
BMP	“*Bitmap* ou Mapa de Bits”
GIF	“*Graphics Interchange Format*”
JPEG	“*Joint Photographic Experts Group*”
LSB	“*Last Significant Bit*”
MB	“*Megabytes* correspondem à 1.024 *Kilobytes*”
GB	“*Gigabytes* correspondem à 1.024 *Megabytes*”
Pixel	“*Picture Elements*”
RAM	“*Radom-Access Memory*”
RGB	“*Red, Green e Blue*”

1. CAPÍTULO 1 - Introdução

1.1. - Motivação

Atualmente a informação vêm ganhando grande volume e velocidade. A quantidade de dados possíveis em uma transmissão e a velocidade com que eles podem ser transmitidos, tiveram um aumento significativo. Com isso, a questão da segurança torna-se um requisito extremamente necessário para uma transmissão de dados, principalmente no que tange a confidencialidade deles.

No entanto, esses acontecimentos somente foram possíveis por causa do surgimento da rede de computadores e posteriormente o da Internet.

A Internet fez com que a troca de informações passasse a ser muito mais rápida, tornando-se o meio de comunicação mais utilizado, não só para comunicação e pesquisas acadêmicas, mas também como meio de fornecer serviços, como por exemplo as lojas virtuais, que hoje representam um número bem significativo em todo o mundo.

Entretanto, o mais intrigante disso é que essa grande inovação não previa que teria um crescimento tão estrondoso, e ainda, que surgiriam pessoas especializadas em roubar informações e aplicá-las para o mal. Como por exemplo utilizar-se de um programa que faça o roubo de senhas de banco para conseguir dinheiro.

Uma das formas encontradas por pessoas que fazem essas coisas, numa tentativa de não serem descobertas, foi inovar nas técnicas de troca de informação. Os chamados piratas de computador ou hackers possuem nos dias de hoje uma das técnicas mais eficientes que é conhecida por esteganografia. Essa técnica através de textos, imagens, sons ou vídeos possibilita ocultar informações de forma que as mesmas passem despercebidas aos olhos humanos. Acredita-se que grandes

atentados terroristas como os de 11 de Setembro de 2001, tenham sido estruturados e planejados utilizando esteganografia como principal meio de comunicação. Além desse exemplo, podemos ter a pedofilia, pornografia e disseminação de vírus com o uso dessa técnica.

Mas, apesar de ter os usos ilegais ou maléficos, essa técnica também pode ser usada para o bem como, por exemplo, armazenar informações médicas de um paciente dentro de seu próprio raio X. Isso garante uma maior confidencialidade à informação.

1.2. - Metodologia

Para o desenvolvimento do projeto foi utilizada a linguagem de programação C em conjunto com a técnica do bit menos significativo (LSB), tendo ainda a possibilidade do uso do algoritmo de Huffman.

1.3. - Objetivo

Fazer uma das sugestões futuras contidas no projeto final do ex-aluno Adriano Delfino. No caso, umas de suas sugestões era a mudança de formato de imagem. Com isso foi desenvolvido um programa que utiliza a técnica de esteganografia para ocultar um texto dentro de uma imagem do formato GIF (Graphics Interchange Format).

2. Capítulo 2 - Processamento de imagens digitais

2.1. - Conceito

Uma imagem pode ser definida como uma função bidimensional, f(x, y), onde x e y são coordenadas espaciais (do plano), e a amplitude de f em qualquer par de coordenadas (x, y) é chamada de intensidade ou tons de cinza da imagem nesse ponto. Quando x, y, e os valores da amplitude de f são todos finitos, de quantidades discretas, nós chamamos a imagem de imagem digital. A área de processamento digital de imagens refere-se ao processamento de imagens por meio de técnicas digitais. Note que uma imagem digital é composta por um número finito de elementos, onde cada um tem seu valor e localização particular. Estes elementos são conhecidos como elementos da figura, elementos da imagem, pels[1] ou pixels[2]. [Gonzalez, 2002]

O pixel é o termo mais utilizado para denotar os elementos de uma imagem digital, pois o mesmo representa o conjunto de pontos que a formam. Segundo Lowell Thing, a cor específica que um pixel descreve é uma mistura dos três componentes do espectro de cores – o RGB (Red, Green e Blue). Até três bytes de dados podem ser alocados para especificar a cor do pixel, cada byte para uma das três cores básicas. Um sistema true color [3]de 24 bits[4] usa todos os três bytes. No entanto, a maioria dos sistemas usa apenas 8 bits (o que provê até 256 cores).

[1] Pels – vêm da contração da palavra inglesa “picture elements”
[2] Pixels – é uma outra contração da palavra inglesa “picture elements”
[3] True Color – cor verdadeira, ou seja, é a especificação da cor de um pixel em uma tela que usa o valor de 24 bits
[4] Bits- menor unidade de dados num computador

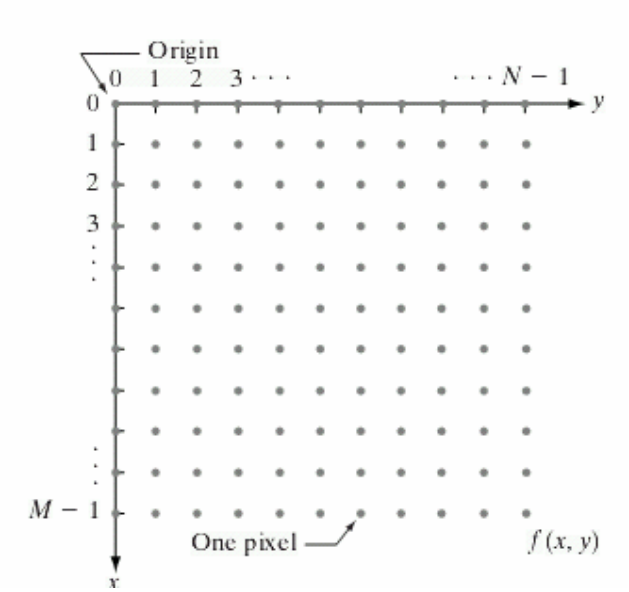

Figura 2.1 - Pixel

Fonte: Gonzalez

2.1.1. - Conceito de Cor

Um elemento que é necessário na criação de uma imagem é a cor.

Cor é um elemento relacionado especificamente a percepção visual do ser humano. Sabe-se que existem animais que não enxergam as cores, portanto a noção de cor está relacionada ao sistema visual do homem [Casacurta, 1998].

2.1.1.1. - Importância do conceito de cor

O conceito de Cor é de grande importância, não sendo indispensável somente para a Computação Gráfica, pois com o seu uso pode ser explorado um dos principais sentidos, a visão. Os seres humanos podem perceber cerca de 100 níveis de intensidade de luz. O desafio do sistema digital é reproduzir esta resposta convertendo a informação recebida em um número apropriado de cores. Sabemos que todas as cores que o olho humano pode detectar são combinações de três

cores primárias: Vermelho/Verde/Azul ou Red/Green/Blue (RGB[5]), como visto no espectro abaixo. [Parconsult, 2006]

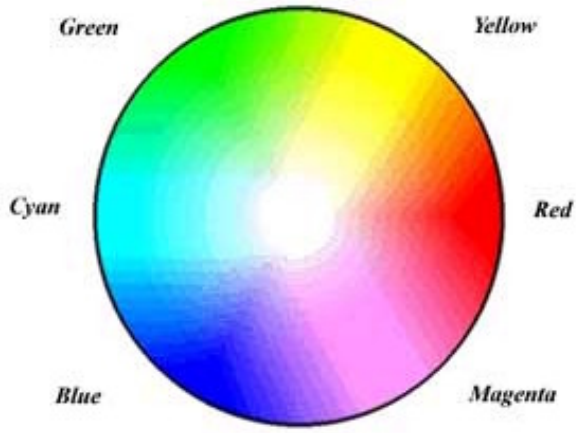

Figura 2.2 - Espectro

Fonte: Parconsult

2.2. - Imagem do tipo GIF

2.2.1. - Conceito

O GIF (Graphics Interchange Format) é um formato de imagem de mapa de bits. Ele trabalha com uma paleta de 256 cores, onde cada pixel tem um índice de cor. O conjunto de índices da imagem é comprimido pelo algoritmo LZW (Lempel-Ziv-Welch) sem perdas. Esse formato de imagem foi criado pela empresa CompuServe, em 1987. Por trabalhar com uma paleta de 256 cores, o GIF não é recomendável para figuras que precisam de aproximação da realidade, a não ser que a imagem em questão seja em preto e branco (escala de cinza). Devido a essa

[5] RGB – é a abreviatura do Inglês *Red, Green, Blue*, que utiliza três números inteiros para representar cada uma das cores primárias: vermelho, verde e azul.

característica, o uso do formato GIF é voltado para ícones ou imagens que não precisam de muitas cores (ilustrações, por exemplo).

Apesar deste formato parecer limitado devido ao número baixo de cores com que trabalha, o GIF é muito utilizado por alguns recursos que oferece. Um deles é a capacidade de utilizar fundo transparente. Com isso, é possível, por exemplo, que um site publique uma imagem em GIF e esta terá como fundo a cor da página. Além disso, o GIF permite que uma seqüência de imagens sejam salvas em um único arquivo, onde cada imagem surge no lugar da anterior após um tempo pré-determinado. Isso dá a sensação de animação.

Segundo Lowell Thing, GIF é um dos formatos mais comuns de arquivo de imagem gráfica na Web. Na Web e em outros locais na internet (por exemplo, servidores de quadros de avisos), o formato GIF acabou se tornando o formato padrão de arquivos de imagem.

O formato GIF utiliza o tipo de dados raster 2D e é codificado em sistema binário (binary). Há duas versões do formato, GIF87a e GIF89a. A versão 89ª (julho de 1989) traz a possibilidade de um GIF animado (animated GIF), que é uma seqüência curta de imagens em único arquivo GIF. Uma GIF89a pode também ser especificada para uma apresentação de GIF entrelaçado (interlaced GIF).

> "Para alguns tipos de imagens, o formato **GIF** é superior em qualidade de imagem, tamanho de arquivo, ou ambos. O que precisamos saber é para que tipo de imagens devemos plica-lo. De maneira geral, é melhor aplicado a imagens com poucas variações de cor ou coloridas com tabelas de até 256 cores ou níveis de cinza. Quando não é possível a perda de dados já que este é um formato sem perdas." [Jackson, 2006]

GIF e arquivos de 8 bits do tipo BMP empregam o que é conhecido por *lossless compression*, um método que permite que um programa reconstrua a imagem original sem perda de dados.

2.2.2. - Vantagens e desvantagens do Formato GIF

O formato GIF é adequado para distribuir imagens que sejam formadas por linhas e por áreas de cor uniforme. Ele não deve ser usado para distribuir fotografias nem qualquer imagem que contenha muitas cores porque só é capaz reproduzir 256 cores diferentes.

2.2.2.1. - Desvantagens do formato GIF

Figura 2.3 - Formato GIF (23KB)

Fonte: Google Imagens

Quando utiliza-se o formato GIF com uma imagem composta por muitas cores, o resultado pode ser aquele que se vê na imagem acima. No caso, pode-se perceber que o aspecto do céu não é muito natural. São visíveis anomalias no lado direito (mais evidentes na parte mais clara do céu) e um pouco por toda a imagem.

Figura 2.4 - Formato GIF (secção ampliada)

Fonte: Google Imagens

Estes efeitos resultam da aplicação de uma técnica de aproximação de cor chamada “dithering”.[6] A figura 4 mostra a ampliação de uma parte da imagem anterior e permite-nos ver claramente o modo como a técnica opera. Mais à frente nós vamos analisar esta técnica com maior detalhe.

Figura 2.5 - Formato JPEG (11KB)

Fonte: Google Imagens

Comparando-se a primeira das fotografias anteriores com a que está acima. Esta última tem um aspecto muito mais natural. Se considerarmos que a imagem em formato GIF (a primeira) tem 22538 bytes de tamanho enquanto que esta em formato JPEG tem apenas 10556 bytes, fica bastante claro que com esta imagem não é recomendado o uso do formato GIF.

2.2.2.2. - Vantagens do formato GIF

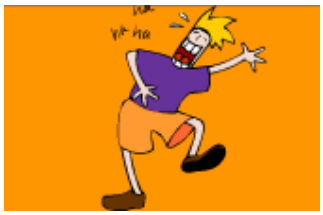

Figura 2.6 - Formato GIF (3KB)

Fonte: Google Imagens

[6] Dithering – técnica criada para compensar cores que faltam.

Se a imagem possuir áreas com cor uniforme, linhas bem definidas e um número de cores não superior a 256 então podemos usar o formato GIF. A figura 6 é um bom exemplo de um caso destes.

Quando uma imagem possui zonas com variação contínua de cor (gradientes de cor) ou formas complexas (não as linhas regulares como as da imagem anterior) o mais provável é o formato GIF produzir maus resultados.

2.2.2.3. - Imagens animadas

O formato GIF permite distribuir imagens animadas que funcionam em todos browsers. Não é exigido aos utilizadores que tenham nenhum plugin instalado nos seus browsers. A figura 7 é composta por 8 imagens estáticas que são substituídas repetidamente a intervalos de 0,1 segundos. O formato GIF é o único que nos permite distribuir imagens animadas na Web sem exigir a utilização de plugins (como o *Macromedia Flash*).

Figura 2.7 - Formato GIF (17KB)

Fonte: Google Imagens

2.2.2.4. - Imagens com partes transparentes

Quando guardamos uma imagem em formato GIF é permitido escolher uma cor (não mais do que uma única cor) que passará a ser transparente.

Figura 2.8 - GIF com transparência (2KB)

Fonte: Google Imagens

Na figura 8 escolhemos a cor preta para ser transparente. O resultado é que essa cor passou a ser invisível e em vez dela é mostrado aquilo que está por trás da imagem.

Por isso se quisermos obter bons resultados quando usamos cores transparentes, devemos ter o cuidado de fazer com que a imagem se misture suavemente com o fundo. Isso não acontece neste exemplo porque ele serve apenas para ilustrar um ponto.

2.2.2.5. - Dithering

Há imagens para as quais o formato GIF não consegue reproduzir todas as cores necessárias, isso devido a ele suportar até 256 cores.

O "dithering" é uma técnica que foi criada para compensar as cores que faltam. Ela consiste em misturar as cores disponíveis de forma a simular as cores

que faltam. O resultado é o aparecimento de pontinhos na imagem. Neste exemplo foi reduzido o número de cores para apenas 16 para que o efeito fosse mais acentuado.

Figura 2.9 - Dithering com 16 cores (3,3kb)

Fonte: Google Imagens

Se optarmos por não usar "dithering", a falta de algumas cores provoca o aparecimento de faixas artificiais na imagem, conforme a “figura 10” abaixo. Estas faixas não existiam na imagem original, que pode ser vista no lado direito em formato JPEG.

Figura 2.10 - Imagem sem dithering (1,9kb)

Fonte: Google Imagens

Figura 2.11 - Imagem Original

Fonte: Google Imagens

No entanto, além de não possuir o melhor efeito visual, essa técnica tem como um outro aspecto negativo a questão do aumento do tamanho da imagem, como pode ser visto nas figuras anteriores, elas apresentam tamanhos diferentes, de forma que a figura que utiliza o dithering apresenta o tamanho maior.

3. Capitulo 3 - Esteganografia

3.1 - Histórias

O primeiro uso confirmado da esteganografia está em "As Histórias" de Heródoto, onde remonta no século V a.C.: um tirano grego chamado Histiaeus, que foi aprisionado pelo rei Darius, Esse grego querendo fazer contato secreto com seu superior, o tirano Aristágoras de Mileto, escolheu um escravo fiel, onde esse teve sua cabeça raspada, de forma que seu couro cabeludo recebeu uma tatuagem com uma mensagem a qual Histiaceus queria enviar. Esperando o tempo necessário para que os cabelos crescessem, mandou o escravo ao encontro de Aristágoras com a instrução de que deveriam raspar seus cabelos. Dessa forma, seria possível a leitura da mensagem.[Muller, 2006]

Ainda nas "As Histórias" de Heródoto, consta que, para informar os espartanos de um ataque iminente dos persas, o rei Demeratos utilizou um estratagema[7] muito elegante: pegou tabletes, retirou-lhes a cera, gravou na madeira a mensagem secreta e recobriu-os novamente com cera. Deste modo, os tabletes, aparentemente virgens, não chamariam a atenção. O problema era que os gregos não sabiam do que se tratava quando Gorgo, mulher de Leônidas, teve a idéia de raspar a cera.[Muller, 2006]

No século XVI, o cientista italiano Giovanni Porta descobriu como esconder uma mensagem num ovo cozido: escrever sobre a casca com uma tinta contendo uma onça de alume (± 29 g) diluído em cerca de meio litro de vinagre. A solução penetra a casca e se deposita sobre a superfície branca do ovo. Depois, basta abrir o ovo para ler a mensagem. [Muller,2006]

[7] Estratagema – Ardil empregado na guerra para burlar o inimigo.

O historiador da Grécia antiga, Enéias, o Tático, teve a idéia de enviar uma mensagem secreta fazendo minúsculos furos em certas letras de um texto qualquer. A sucessão destas letras marcadas fornecia o texto secreto. Dois mil anos mais tarde, remetentes ingleses empregaram o mesmo método, não para garantir o segredo de suas cartas, mas para evitar o pagamento de taxas muito caras. Na realidade, antes da reforma do serviço postal ao redor de 1850, enviar uma carta custava cerca de um shilling para cada cem milhas de distância. Os jornais, no entanto, eram isentos de taxas. Graças a furinhos de agulha, os ingleses espertos enviavam suas mensagens gratuitamente. Este procedimento foi até utilizado pelos alemães durante a Primeira Guerra Mundial. Durante a Segunda Guerra, eles aperfeiçoaram o método marcando letras de jornais com tintas "invisíveis".[Muller, 2006]

Os espiões alemães da Segunda Guerra faziam uso de micropontos para que suas mensagens viajassem de maneira discreta. Eram fotografias do tamanho de um ponto (.) que depois tinham seu tamanho aumentado, de forma que a mensagem apareceria claramente. Era uma espécie de microfilme colocado numa letra, num timbre, etc. [Muller, 2006]

Em 1999, Catherine Taylor Clelland, Viviana Risca e Carter Bancroft publicaram na revista Nature "Hiding messages in DNA microdots" (escondendo mensagens em micropontos de DNA). Na verdade, qualquer material genético é formado por cadeias de quatro nucleotídeos (Adenina, Citosina, Guanina e Timina) que podemos comparar a um alfabeto de quatro letras: A, C, G e T. Além disso, os cientistas atualmente são capazes de fabricar cadeias de DNA com um conjunto predeterminado de nucleotídeos. Nada impede de atribuir a um grupo de três nucleotídeos uma letra do alfabeto, um número ou sinais de pontuação (por

exemplo, "A"=CGA, "B"=CCA, etc) e compor uma "mensagem genética". Para disfarçar as pistas, poderia-se misturar algumas outras seqüências aleatórias de nucleotídeos. O resultado é apenas visível ao microscópio eletrônico. Como possível aplicação, pode-se imaginar que uma empresa que produza uma nova espécie de tomate poderá incluir sua marca de fábrica nas moléculas do tomate afim de evitar as imitações. [Muller, 2006]

3.2 - Definição

A palavra esteganografia tem sua origem de duas palavras gregas: steganos (coberto) e graphein (escrever) que juntando significam escrita encoberta. Basicamente, esta técnica realiza uma comunicação secreta por ocultação de mensagem.

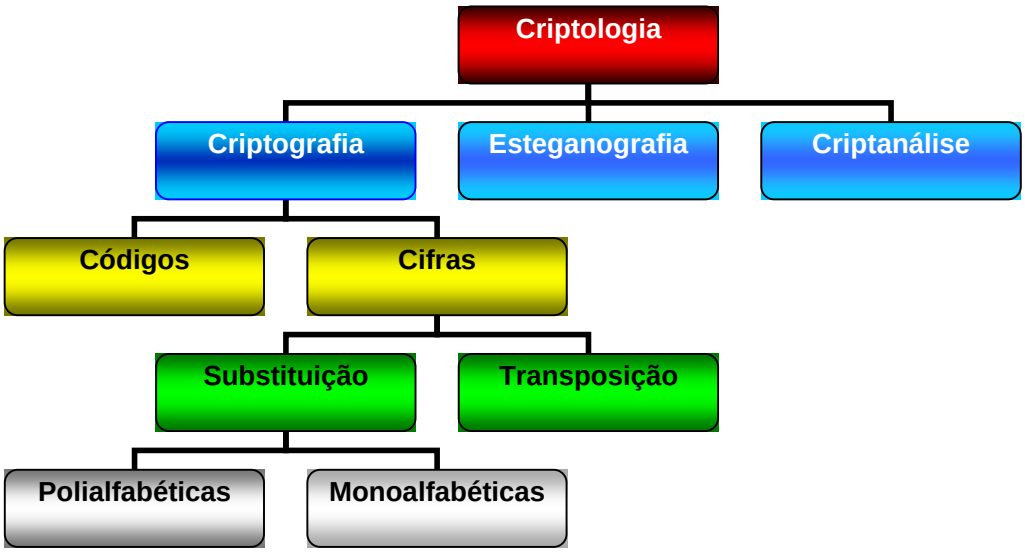

Figura 3.1 - Organograma de Esteganografia

Fonte: Autor

Diferentemente da criptografia, que cifra[8] as mensagens de modo torná-las incompreensíveis, a esteganografia esconde as mensagens através de artifícios, como imagens, vídeos, áudios e textos que tenham sentido mas que são usados apenas como suporte, ou seja essa técnica com o auxilio de um portador aparentemente inofensivo consegue camuflar a existência de uma mensagem secreta.

No entanto, apesar da esteganografia e a criptografia apresentarem suas diferenças, essas duas tecnologias de comunicação secreta podem ser usadas em conjunto, como por exemplo encriptando primeiro a mensagem, e então ocultando-a em outro arquivo para a transmissão. Com isso, teria-se uma forma mais robusta e eficiente para a manutenção da integridade e confidencialidade das informações.

Conforme o mundo vai se tornando mais apreensivo em relação ao uso de qualquer comunicação secreta, e o governo com a criação de regulamentos para limitar o uso da encriptação, o papel da esteganografia vêm ganhando destaque. [Cole, 2003]

3.3 - Onde os dados ocultos se escondem?

Ao contrário de um arquivo de palavras processadas onde seja provável a observação da falta de algumas letras, é possível alterar ligeiramente arquivos de som e imagem sem perder sua viabilidade total para quem vier a ver ou escutar esses arquivos. Com o de som, podem ser usados os bits do arquivo que contém som não audível para o ouvido humano. Já com imagens, pode-se remover os bits

[8] Cifrar – método de criptografar que oculta a legibilidade e significado de um texto.

de cor redundantes de uma imagem e ainda produzir uma figura que aparenta-se intacta para o olho humano, sendo difícil discernir da original. [Cole, 2003]

Então, quanto maior a qualidade da imagem ou do som, maior será a quantidade de dados redundantes, o que explica o porquê de um som de 16 bits e imagem de 24 bits serem popularmente "*hiding spots*". Se uma pessoa está espionando alguém e não possui o arquivo de imagem ou som original, de forma que possa comparar com o arquivo de esteganografado, ele geralmente nunca poderá dizer se o que está sendo transmitido não é um arquivo de som ou imagem original, e ainda, que há dados escondidos nele. [Cole, 2003]

3.4 - Princípios da Esteganografia

Segundo Eric Cole, a esteganografia envolve ocultação de dados em uma mensagem, de forma a dificultar para o adversário a detecção e remoção desses. Com base nesse objetivo, três princípios podem ser usados para medir a eficiência de uma dada técnica de esteganografia: quantidade de dados, dificuldade de detecção e remoção.

- Quantidade de dados sugere que quanto mais dados podem ser ocultados, melhor a técnica.
- Dificuldade de detecção está relacionada à facilidade de alguém saber a existência de uma mensagem escondida. Existe uma relação direta entre quantidade de dados que podem ser ocultados e a facilidade de alguém detectá-los. À medida que se aumenta a quantidade de informação que é escondida em um arquivo, maior a probabilidade de alguém descobrir que existe informação escondida no arquivo.

- Dificuldade de remoção envolve o princípio de alguém interceptar um arquivo e não conseguir remover os dados ocultos com facilidade.

3.4 - Terminologias utilizadas na Esteganografia

No “Information Hiding Workshop”, realizado em Cambridge, Inglaterra em abril de 1996 foi definida a terminologia para descrever um subconjunto de informação oculta. [Bento, Coelho]

De acordo com eles, a descrição de esteganografia (escrita oculta) pode ser realizada da seguinte forma:

O dado embutido (embedded data) é a informação que alguém deseja transmitir secretamente. Este é escondido em uma mensagem aparentemente inocente, conhecida por recipiente ou de objeto de cobertura (container ou cover-object), produzindo assim um estego-objeto (stego-object).

Esse processo pode ser representado através da seguinte fórmula:

Recipiente + Mensagem Embutida = Estego-Objeto

Figura 3.2 - Terminologias

Fonte: Autor

Segundo [Bento, Coelho], o termo recipiente é associado a qualquer tipo de informação digital que utiliza-se para a sua transmissão um sistema digital ou analógico, assim como arquivos de texto, áudio ou vídeo.

3.5 - Técnicas para codificar uma mensagem

As formas mais comuns para inserção de mensagens texto em imagens podem ser obtidas através das seguintes técnicas abaixo:

- Inserção do bit menos significativo

- Filtragem e mascaramento
- Algoritmo e transformação

3.5.1 - Inserção do Bit Menos Significativo

Como é colocado por Jascone (2003, p.38), “O método Last Significant Bit (LSB) é o mais comum utilizado para armazenar informação em imagens digitais. Consiste em utilizar o bit menos significativo de cada pixel (ou de cada cor) da imagem, para ocultar a mensagem”.

Uma conversão simples de um formato BMP[9] a um formato de compressão do tipo lossy como o JPEG pode destruir a informação escondida na imagem [Johnson, Jajodia, 1998].

Ao aplicar técnicas de LSB (método do bit menos significativo) a cada byte de uma imagem 8-bits, 1 bit pode ser codificado em cada pixel , como cada pixel é representado por um byte, todas as mudanças ocorridas no pixel serão imperceptíveis ao olho humano.

Como exemplo, poderia-se utilizar uma mensagem com valor binário de 10000011, onde essa seria escondida em 8 pixels. Suponha que os 9 pixels originais são representados pelas palavras de 8-bits abaixo:

00100111 11101001 11001000
00100111 11001000 11101001
11001000 00100111 11101001

Figura 3.3 – Representação de uma mensagem em binário

Fonte: Autor

[9] BMP - abreviação de bitmap.

Introduzir o valor binário de 10000011 da mensagem nos 9 pixels , partindo do byte esquerdo superior, resultaria em:

00100111 11101000 11001000
00100110 11001000 11101000
11001001 00100111 11101001

Figura 3.4 - Representação de uma mensagem utilizando o LSB

Fonte: Autor

Os bits sublinhados foram os que realmente sofreram alteração.

A vantagem principal da inserção de LSB é que os dados podem ser escondidos no menor e no segundo menor bit, e ainda, o olho humano seria incapaz de observar a degradação da imagem, além de ser fácil a implementação. [Johnson, Jajodia, 1998]

3.5.3 - Mascaramento e filtragem

Técnicas de mascaramento e filtragem escondem informações marcando uma imagem de maneira similar as marcas d'água. As técnicas de marca d'água são integradas na imagem, por isso, podem ser aplicadas, sem medo da destruição da imagem, na compressão do tipo lossy ou compressão com perda de dados. Cobrindo ou mascarando um sinal fraco mais perceptível com outro tornando o primeiro não-perceptível, explorando o fato que o sistema visual humano não pode detectar mudanças ligeiras em determinados domínios (temporal) da imagem [Anderson, Petitcolas, 1998].

Tecnicamente, marcas d'água não são fórmulas esteganográficas. Estritamente, a esteganografia esconde dados na imagem, a marca d'água estende

a informação da imagem e transforma-se em um atributo da imagem de cobertura, fornecendo detalhes da licença, da posse ou do copyright , podendo esconder ou não dados na imagem [Góis, 2003].

De acordo com [Johnson e Jajodia, 1998], “Técnicas de filtragem e mascaramento são restritas às imagens em tons de cinza. Estas técnicas, escondem a informação através da criação de uma imagem semelhante às marcações de copyright em papel”. Isto deve-se ao fato das técnicas de marca d’água impedirem que a marcação seja removida, mesmo se a imagem tiver sido alterada por métodos de compressão.

Filtragem e mascaramento são técnicas mais robustas que a inserção LSB no sentido de gerarem estego-imagens imunes a técnicas de compressão e recorte. Ao contrário das modificações LSB, filtragem e mascaramento trabalham com modificações nos bits mais significativos das imagens. [Popa, 1998]

As imagens de cobertura devem ser em tons de cinza porque estas técnicas não são eficientes em imagens coloridas [Rocha, 2003]. Isso é resultante das modificações nos bits mais significativos que em imagens coloridas geram alta quantidade de ruído, fazendo com que as informações possam ser detectadas.

3.5.4 - Algoritmo e Transformação

Por se tratar de imagens de elevada qualidade com boa compressão, é desejável usá-las no formato JPEG.

As imagens JPEG usam o discrete cosine transform (DCT) para conseguir a compressão. DCT é uma compressão do tipo lossy ou compressão com perda de dados, porque os valores de co-seno não podem ser calculados precisamente, e o arredondamento de erros pode ser introduzido. As variações entre os dados

originais e os dados recuperados dependem dos valores e dos métodos usados no cálculo do DCT. As imagens podem também ser processadas usando a transformação rápida de Fourier e a transformação de Wavelet [Anderson, Petitcolas, 1998].

Estes algoritmos são mais eficazes para processar imagens como o cropping, mas a custo do tamanho da mensagem. Mesmo se a imagem é cropped , há uma probabilidade que a marca d água ainda seja legível. [Anderson, Petitcolas, 1998]

3.6 - Usos Inapropriados

A esteganografia é uma poderosa ferramenta que caindo em mãos erradas pode ser usada de forma inapropriada. Uns dos usos indevidos que podem ser dados a essa ferramenta são: comunicação criminal, pornografia, vírus e pedofilia.

Um terrorista poderia através dessa técnica passar mensagens para outro de forma desapercebida através de páginas da internet, onde nessas mensagens conteriam planos ou mesmo resultados de ataques efetuados com sucesso.

Já com o vírus, pessoas mal intencionadas poderiam disseminar vírus que fazem operações sofisticadas, como roubo de senhas, tendo-se assim acesso para fazer tudo que lhe convier, como dinheiro e informações sigilosas que muitas das vezes possuem valores inestimáveis.

A pedofilia e pornografia poderiam ser cometidas por mascaramento, onde teríamos uma imagem aparentemente normal que de forma alguma aparentaria que por trás dela, estaria trazendo imagens com esse tipo de conteúdo, apenas o remetente e o destinatário teriam conhecimento disso.

4. Capítulo 4 - Criptografia

4.1 - Conceito

A criptografia é um ramo que faz parte do domínio da criptologia, o qual aborda maneiras para processamento computacional ou não de documentos com a finalidade de esconder seu conteúdo, validar sua integridade e impedir o acesso de pessoas não autorizadas. Na prática, criptografia é usada para manter secreto algo sigiloso, ou seja, ela codifica a informação de tal forma que ninguém pode lê-la, exceto a pessoa que possui a chave. Sendo que em técnicas mais avançadas, ela garante que a informação durante a transmissão não será modificada.

Segundo o escritor Bruce Schneier, a criptografia deve fornecer outros serviços além de confidencialidade. Dentre esses estão:

- Autenticação. Deve ser possível ao receptor de uma mensagem certificar-se da veracidade de sua origem, onde um intruso não deve ser capaz de mascarar-se como outro individuo.

- Integridade. Deve ser possível ao receptor de uma mensagem saber se o conteúdo foi alterado em trânsito, onde um intruso não conseguirá trocar uma mensagem legítima por uma falsificada.

- A não retratação. Um emissor não será capaz de negar o envio da mensagem

Ainda segundo Bruce Schneier, os métodos criptográficos estão divididos em:

- Sistemas de Cifras.

- Códigos.

- Esteganografia.

4.1.1 - Sistemas de Cifras

Os sistemas de cifras se diferenciam pelos métodos os quais são empregados nela. Transposição e Substituição.

O Método da Substituição, o valor normal ou convencional das letras do texto original é mudado, mas sua posição permanece a mesma. Já, na transposição, apenas a posição das letras do texto original é mudada, mantendo-se assim seu valor normal ou convencional sem qualquer modificação.

Como os métodos de encriptação são bem distintos, os princípios envolvidos na criptoanálise[10] dos dois métodos também são fundamentalmente distintos.

Palavra-chave	G	E	N	I
Texto Claro	INVENCAO	DA	PALAVRA	CHAVE
Texto Cifrado	TDMPDACX	QO	ECBCMGC	XQEBA

Figura 4.1 - Representação da cifragem de um texto

Fonte: Google Imagens

4.1.1.1 - Cifras de substituição monoalfabéticas

Na substituição monoalfabética, ou substituição simples, substitui-se cada um dos caracteres do texto original por outro, de acordo com uma tabela pré-estabelecida.

Este tipo de substituição pode ser feito com o auxilio de uma chave, a qual será um número que indica quantas posições deve-se avançar no alfabeto, para se obter o texto cifrado.

[10] Criptoanálise – é o ramo da criptografia que estuda formas de decodificar uma mensagem sem conhecer a chave. A técnica de criptoanálise é responsavel por quebrar o código da mensagem cifrada, e não em decifrá-lo.

É possível identificar que a freqüência de ocorrência dos símbolos que resultam na mensagem cifrada é a mesma das letras da língua usada na mensagem original.

Este tipo de substituição pode ser monográfica ou poligráfica.

A substituição monográfica consiste na substituição de cada letra da mensagem original por apenas um símbolo. Esta regra resultará na formação de uma mensagem cifrada de comprimento igual ao da mensagem original.

Já a substituição poligráfica, consiste na substituição de cada letra da mensagem original por mais de um símbolo, resultando na formação de uma mensagem cifrada de comprimento diferente da mensagem original.

Um exemplo deste tipo de técnica é a Cifra de César. Estima-se que em 60 a.C. ; o imperador Júlio César desenvolveu e empregou uma cifra de substituição para cifrar mensagens governamentais. Ele basicamente avançava a ordem das letras desviando-as em, por exemplo, em três posições; A se tornava D, B se tornava E, etc.

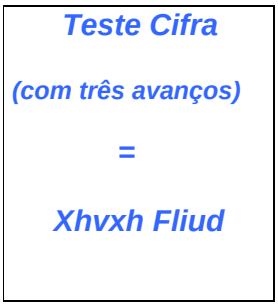

Figura 4.2 - Representação da cifra de César

Fonte: Autor

4.1.1.2 - Cifras de substituição polialfabéticas

As substituições polialfabéticas são aquelas em que um conjunto de letras é substituído por outro conjunto de letras. Os conjuntos de letras (alfabeto) no caso não precisam ser necessariamente de origens diferentes.

Desde que a ordem da seqüência das letras seja mudada já é considerado um “novo alfabeto”.

Um exemplo de um alfabeto de substituição seria l-y-n-x... e k-b-c-a. Se eles fossem utilizados para cifrar uma mesma mensagem, fazendo a substituição das letras originais, teria-se uma substituição polialfabética.

Pode-se obter a substituição polialfabética com palavra-chave ou com auto-chave. Num sistema de palavra-chave é essa que mostra quais os alfabetos cifrantes que devem ser usados.

Já com auto-chave há uma chave que indica a escolha inicial do alfabeto cifrante e depois a própria mensagem determina os alfabetos subseqüentes.(SUL03)

Um exemplo de cifras de substituição polialfabética é: o Cilindro de Jefferson. Segue figura abaixo:

Figura 4.3 - Cilindro de Jefferson

Fonte: Google Imagens

4.1.1.3 - Cifras de transposição

A transposição faz com que as letras das mensagens sejam embaralhadas de acordo com um padrão estabelecido, sem modificá-las.

É um criptograma[11] que pode tratar um a um ou em grupo de comprimento constante os caracteres do texto original, deslocando-os de acordo com um sistema predefinido e uma chave.

É importante ressaltar que os caracteres (letras, números ou símbolos) não são alterados, há apenas mudança de posição conforme a cifra.

Um exemplo histórico do uso do método de cifragem por transposição é o Bastão de Licurgo. Ele consiste num bastão no qual é enrolada uma tira de couro ou pergaminho. O remetente escreve a mensagem ao longo do bastão e depois desenrola a tira, a qual então se converteu numa seqüência de letras sem sentido. O mensageiro usa a tira como cinto, com as letras voltadas para dentro. O destinatário, ao receber o ”cinto”, enrola-o no seu bastão, cujo diâmetro é igual ao do bastão do remetente. Dessa forma a mensagem poderá ser lida, conforme a figura abaixo.

Figura 4.4 - Bastão de Licurgo

Fonte: Google Imagens

[11] Criptograma – texto criptografado de uma mensagem.

4.1.2 - Código

O código é uma espécie de linguagem secreta criada com intuito de esconder o significado de uma mensagem. Nesta técnica, há troca de uma palavra por outra ou a contextualização de cenários com personagens.

Na frase, ”Coronel, acabei de reconhecer o carro no estacionamento.”, é possível perceber que um agente (soldado) usa palavras-código para avisar o agente (Coronel) que o alvo (carro) está no local esperado (estacionamento).

Na prática, códigos de espionagem são chamados de números-código ao invés de palavras-código. Para isso há o uso de livros de código que possuem um dicionário de números com suas respectivas palavras. Por exemplo, a mensagem acima poderia ser codificada como:

92281 91466 00758 27001

Figura 4.5 - Representação de uma mensagem codificada

Fonte: Autor

Onde ”92281” seria (Coronel), ”91466” seria (soldado), ”00758” seria (carro) e ”27001” seria (estacionamento).

Um exemplo do uso dos sistemas de códigos em mensagens é o Código Morse. A transmissão das mensagens (informação) se dava por meio de um código que utilizava apenas dois símbolos (traços e pontos) para representar as letras do alfabeto. Ele é representado conforme a figura abaixo:

Figura 4.6 - Representação do Código Morse

Fonte: Google Imagens

4.1.3 - Esteganografia

Já definido no capítulo 3, é uma técnica que utiliza-se de um portador para ocultar uma informação. Esse pode ser tanto um arquivo de áudio, imagem, vídeo ou texto, onde esses para os olhos humanos serão as únicas coisas perceptíveis. Com isso, o que estiver por detrás desses arquivos ou “estego-objetos”, apenas o receptor e emissor serão capazes de saber.

5. Capítulo 5 - Compressão

5.1 - Compressão Estatística

A idéia da compressão estatística é fazer uma representação de caracteres ou grupos de caracteres de forma otimizada.

Os caracteres de maior freqüência de utilização são representados por códigos binários pequenos, e os de menor freqüência são representados por códigos proporcionalmente maiores. [Wikipedia, 2006]

O objetivo da codificação estatística é diminuir o comprimento médio dos códigos usados, consequentemente, aumentando-se a eficiência da compressão.

5.2 - Codificação de Huffman

A técnica de codificação de Huffman permite a representação em binário de seus caracteres a partir de sua probabilidade de ocorrência.

A idéia geral é que o codificador deve dar como saída, representações mais curtas para símbolos mais prováveis e mais compridas para símbolos menos prováveis. Uma consideração importante é a velocidade do codificador, pois pode haver um grande overhead se for requerida uma compressão ótima. Pode-se sacrificar um pouco da performance da compressão para reduzir o overhead.

Dado um texto cujos símbolos em ordem decrescente de probabilidade sejam: t,e,s,n,a,o,l, uma possível representação para os símbolos seria a seguinte:

Símbolos	**Representações**	**Probabilidades**
T	**00**	**0,3**
E	**10**	**0,2**
S	**010**	**0,2**
N	**110**	**0,1**
A	**111**	**0,1**
O	**0110**	**0,05**
L	**0111**	**0,05**

Utilizando-se da tabela acima em uma árvore, teremos a figura abaixo, que poderá ser usada para a decodificação. Para decodificar um símbolo, a árvore é percorrida transversalmente, começando da raiz até alguma folha. O caminho percorrido corresponde a uma representação na tabela acima.

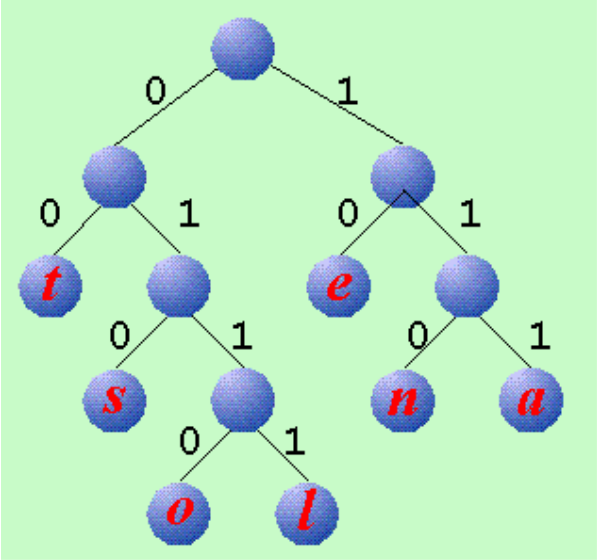

Figura 5.1 - Árvore Binária

Fonte: USP

O algoritmo de Huffman constrói a árvore de decodificação de baixo para cima. Como exemplo, sabendo as probabilidades dos símbolos, primeiro o algoritmo constrói um nó para cada símbolo. Depois os dois nós com menor probabilidade tornam-se irmãos na árvore, criando-se para isso um nó pai que tenha como probabilidade a soma dos seus dois filhos. A operação de combinação é repetida, escolhendo-se novamente os dois nós com menor probabilidade e ignorando-se os nós que já são filhos. [Hayashida, 2006]

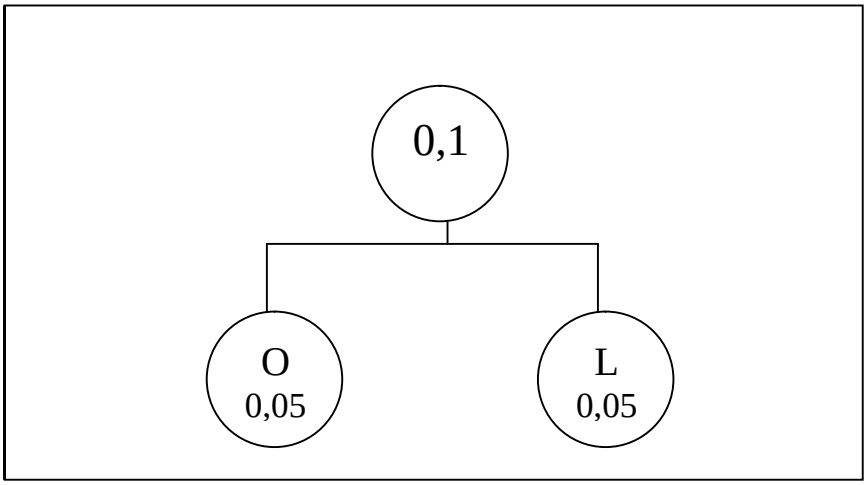

Figura 5.2 - Combinação dos nós O e L

Fonte: USP

O código de Huffman é geralmente rápido tanto para codificação como para decodificação, visto que as probabilidades são estáticas. Entretanto, há versões em que as probabilidades são adaptativas, o que diminui a velocidade devido a um maior overhead para a computação das probabilidades. [Hayashida, 2006]

6. Capítulo 6 - Protótipo

O protótipo tem como objetivo fazer a inserção de um texto de tamanho limitado dentro de imagem do formato GIF utilizando-se da técnica de esteganografia. Dessa forma é possivel garantir a segurança da informação se a mesma estiver em um ambiente hostil, pois as modificações que a imagem sofre, são imperceptíveis ao olho humano.

6.1 - Ferramentas

6.1.1 - Hardware

Foi utilizado para o desenvolvimento do protótipo um computador com a seguinte configuração: Processador: AMD Duron 1,1Ghz, 384 MB de memória RAM[12], disco rígido de 40 GB[13], placa mãe SIS 730S , placa de rede Realtek RTL 8139 Family PCI Fast Ethernet NIC, som e video onboard.

6.1.2 - Software

6.1.2.1 - Sistema operacional

Para a implementação do protótipo foi utilizado o sistema operacional Windows XP versão que contém já o SP2.

6.1.2.2 - Programas para visualização das imagens

Para fazer as comparações com a imagem original e a esteganografada foi utilizado um programa. No caso, esse programa é: o Adobe Photoshop 8.0, um programa para visualização e editoração de imagens.

[12] RAM - abreviação da palavra inglesa “Random Access Memory” ou memória de acesso aleatório.
[13] GB – abreviação de Gigabyte corresponde a 1024 Megabytes.

6.1.2.3 - Compilador

Para a geração do programa executável foi utilizado o compilador Dev-C++ 5 beta 9 release (4.9.9.2).

6.1.3 - Linguagem e Algoritmo

6.1.3.1 - Linguagem de Programação

Para o desenvolvimento do protótipo foi escolhida a linguagem de programação C, isso porque ela possui operadores lógicos que facilitam quando se trabalha a nivel de bit.

6.1.3.2 - Algoritmo

Foi escolhido para a parte de compressão do texto ou da mensagem inserida na imagem do formato GIF, o algoritmo de Huffman.

6.2 - Caracteristicas do Protótipo

As principais caracteristicas do protótipo são as seguintes:

- Interface de linha de comando;
- Fácil manuseio;
- Protótipo funciona em máquinas de diferentes arquiteturas;
- Manipulação de imagem do formato GIF 8 bits;
- O usuário tem a liberdade de poder digitar o texto que quiser, mas com limitação no tamanho;
- Será utilizado um programa em conjunto com o protótipo para visualização da imagem original e imagem esteganografada;
- Utilização da tabela ASCII e de uma tabela de Codificação Huffman.

6.3 - Execução do programa

Foram criados dois programas executáveis através do compilador Dev-C++ 5 beta 9 release (4.9.9.2) : o esteg.exe é o programa para inserção de textos e o extracao.exe é o programa utilizado para extrair a mensagem esteganografada.

Primeiramente, utiliza-se o programa executável de inserção de texto, **esteg.exe**, para inserir uma mensagem esteganografada na imagem gif.

A tela inicial do esteg.exe é mostrada abaixo:

Para a execução do programa deverá ser usado o programa executável criado através do compilador Dev-C++ 5 beta 9 release (4.9.9.2). Isso é feito clicando-se duas vezes com o botão esquerdo do mouse no ícone de inserção de texto do programa executável.

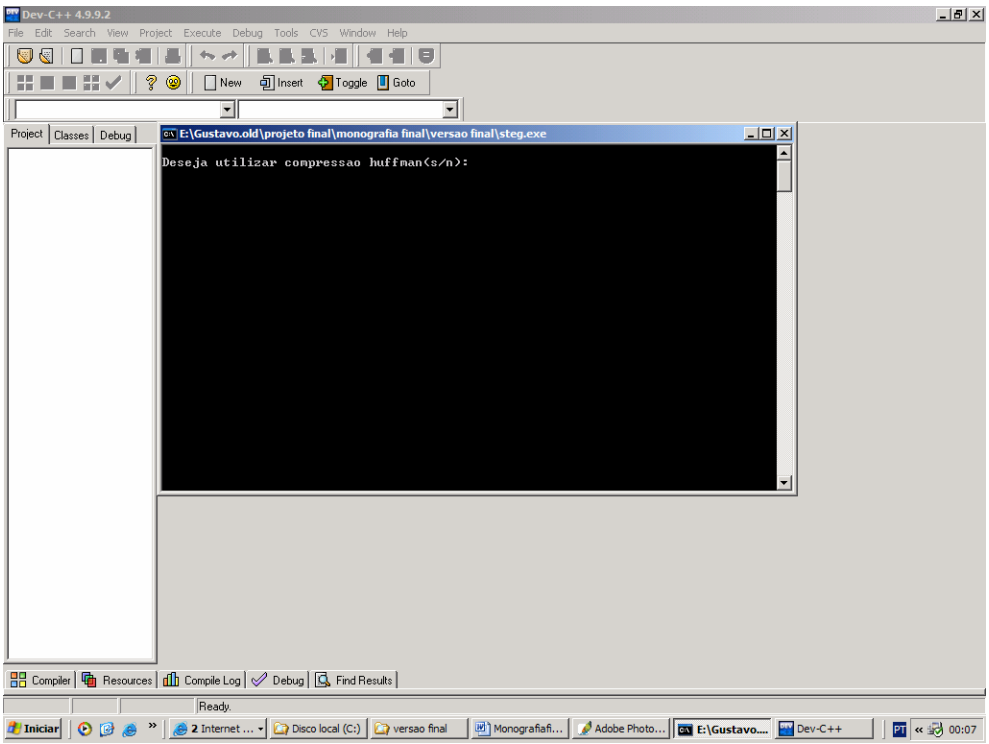

Figura 6.1 - Início do programa executável para inserção de texto

Fonte: Autor

Nessa tela inicial do programa será iniciado o processo de inserção de texto dentro de uma imagem GIF. Nela, inicialmente, é perguntado para o usuário se ele deseja ou não utilizar a codificação de Huffman. Em caso afirmativo, o usuário será levado para uma tela semelhante a seguinte.

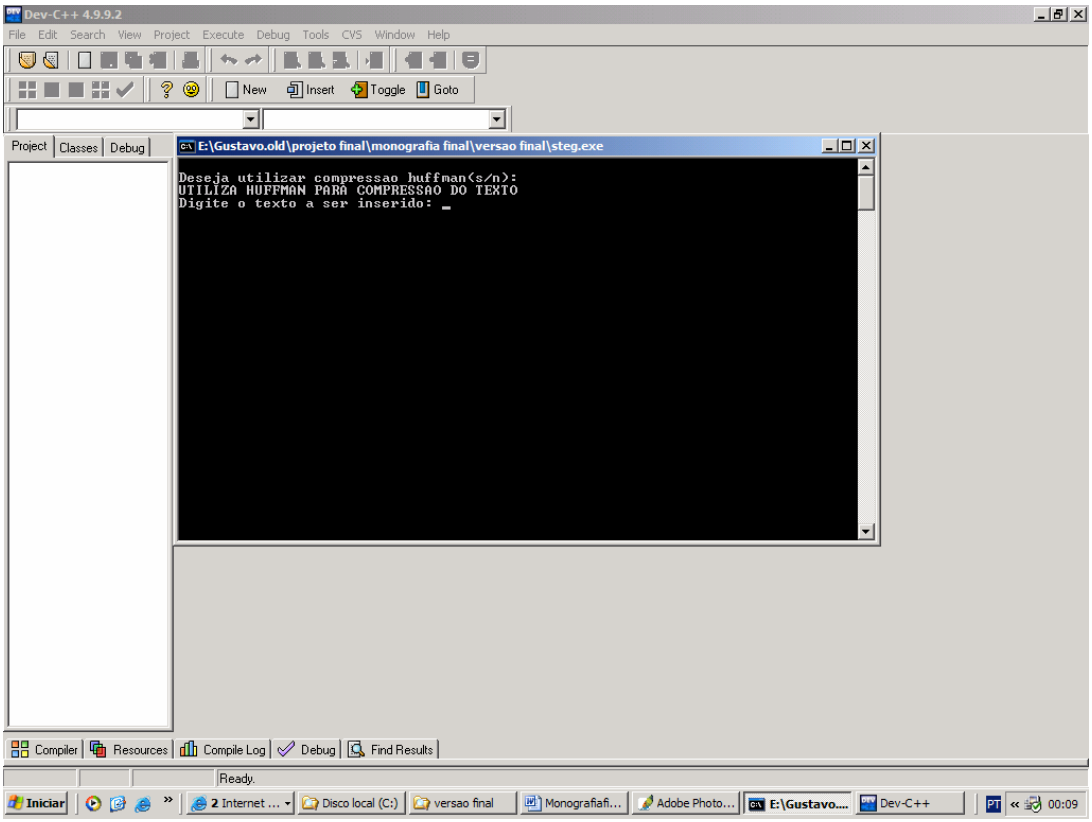

Figura 6.2 - Inserindo texto

Fonte: Autor

Nessa tela será pedido para o usuário digitar uma mensagem com no máximo 90 caracteres. Essa limitação existe porque foi usado como lógica a idéia de trabalhar com a tabela de cores do formato GIF.

O formato GIF possui em sua estrutura duas tabelas de cores sendo uma conhecida por Local e a outra por Global. Como o fluxo só pode ter uma tabela

global, e esta pode ser temporariamente sobrepujada por uma local, uma tabela local serve apenas para o gráfico que a sucede no fluxo, onde as mesmas são opcionais.

O que foi feito nesse caso foi replicar a tabela Global para a Local, pois como elas são opcionais, consegue-se obter um espaço na imagem para os dados. E como esse espaço que foi disponibilizado possui um tamanho máximo de 768 bytes, usando-se o método do bit menos significativo, é possivel inserir até 96 caracteres, pois o GIF trabalha com 8 bits.

No entanto, foram disponibilizados 6 bytes para o cabeçalho, esse sendo usado para fazer o reconhecimento da imagem após a inserção de um texto. O cabeçalho é composto por identificador de imagem esteganografada (2 bytes), identificador de codificação Huffman (1 byte) e quantidade de bits utilizados para inserção da mensagem (3 bytes). Assim é possivel saber se a imagem contém ou não dados ocultos nela e ainda se foi ou não utilizado a codificação de Huffman.

Após ter sido feita a inserção da mensagem pelo usuário, será exibida uma tela que mostrará para ele os bytes utilizados com a codificação de Huffman e sem ela. Além disso, será mostrada a taxa de compressão com o uso da codificação de Huffman. Segue abaixo a tela de exibição.

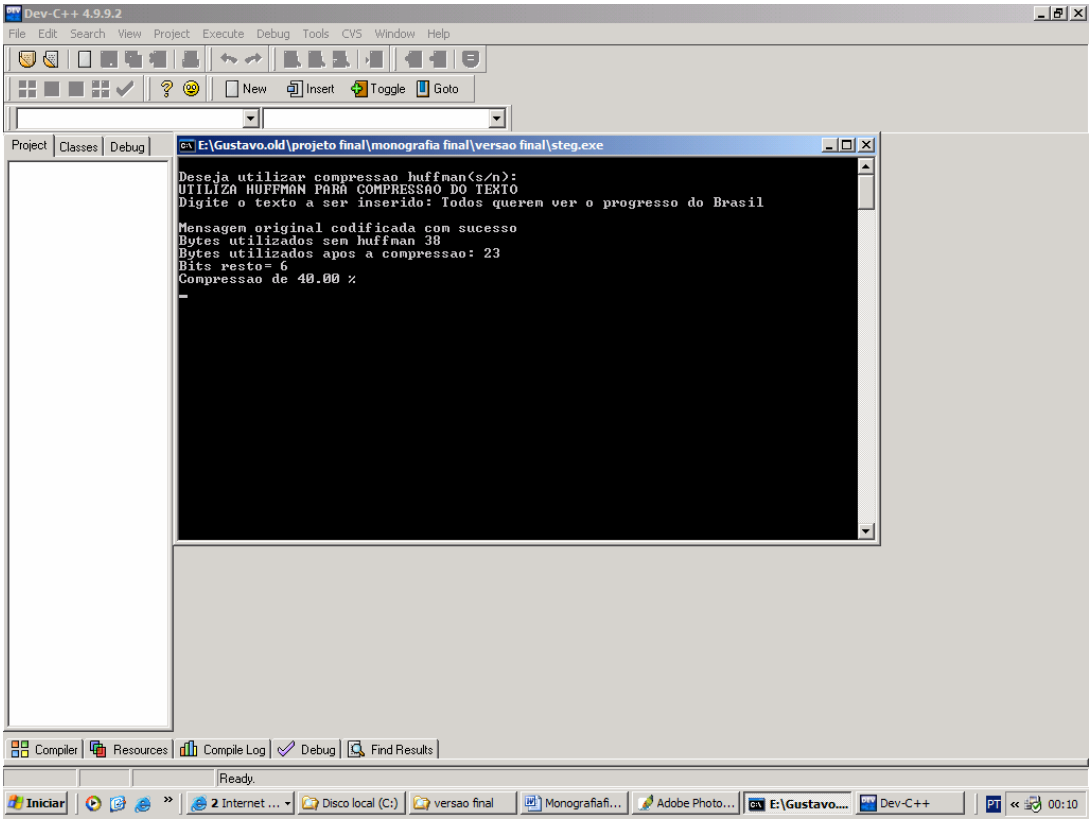

Figura 6.3 - Exibição da compressão através do uso da Codificação de Huffman

Fonte: Autor

O usuário tendo feito isso, será pedido a ele para informar o caminho o qual contém a imagem que ele deseja utilizar para ser esteganografada. O caminho deve conter o **“diretório + nome do arquivo”**, onde o nome do arquivo sempre ao final tem que ser inserido **.gif**. Uma tela como a seguinte será exibida.

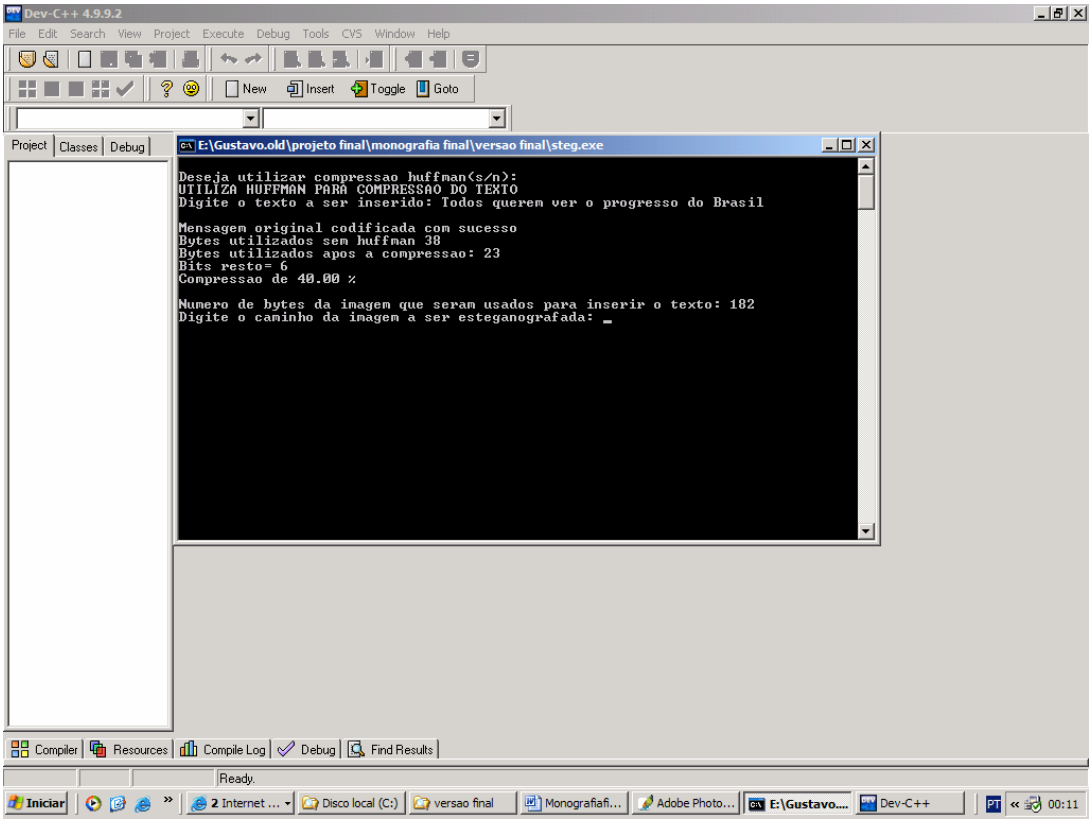

Figura 6.4 - Digitando o caminho da imagem a ser esteganografada

Fonte: Autor

Após isso, o programa insere na tela o cabeçalho, mais a mensagem a ser escrita na imagem que foi selecionada pelo usuário. Tem-se então o término do processo de inserção de texto na imagem. Isso é visto na tela abaixo.

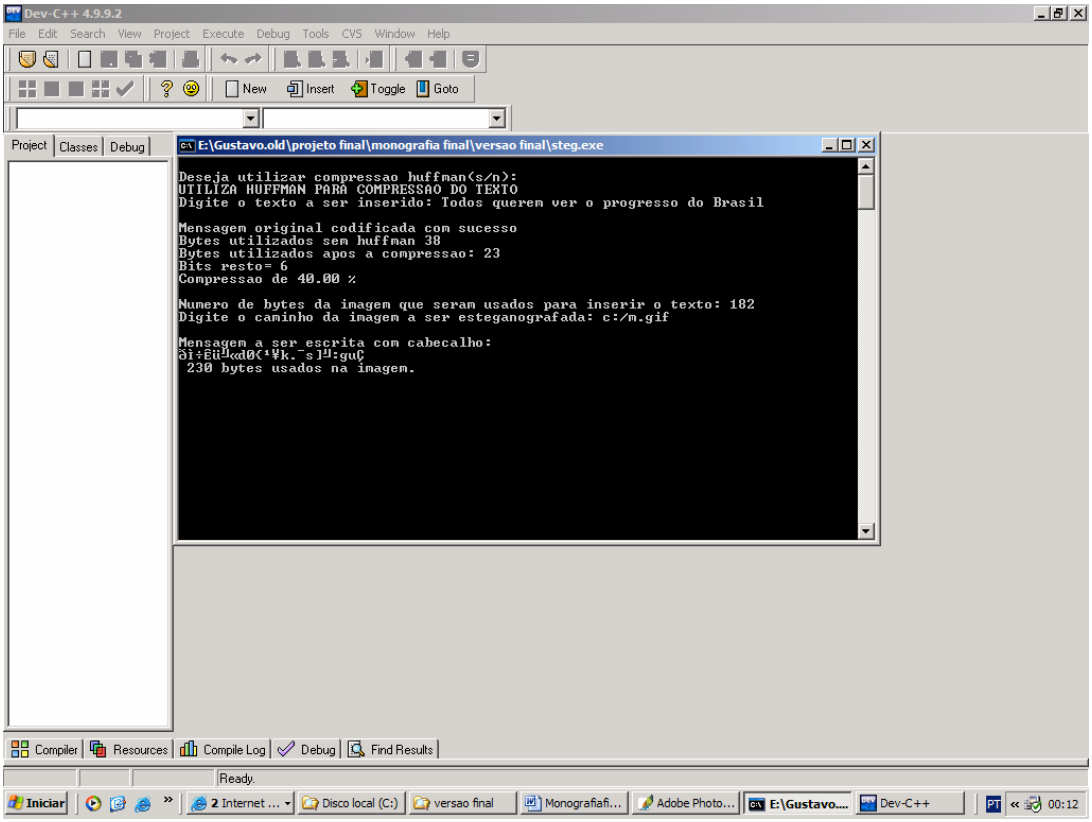

Figura 6.5 - Término do processo de inserção de texto

Fonte: Autor

Agora para saber que a mensagem realmente está dentro da imagem tem-se o processo inverso que é a extração do texto da imagem. Para esse processo, executa-se o programa executável **extracao.exe.**

O programa de extração quando inicia, pedirá ao usuário que digite o caminho da imagem que foi esteganografada para que possa ser extraida a mensagem. O caminho deve conter o **"diretório + nome do arquivo"**, onde o nome do arquivo sempre ao final tem que ser inserido **.gif**. Uma tela como a seguinte será exibida.

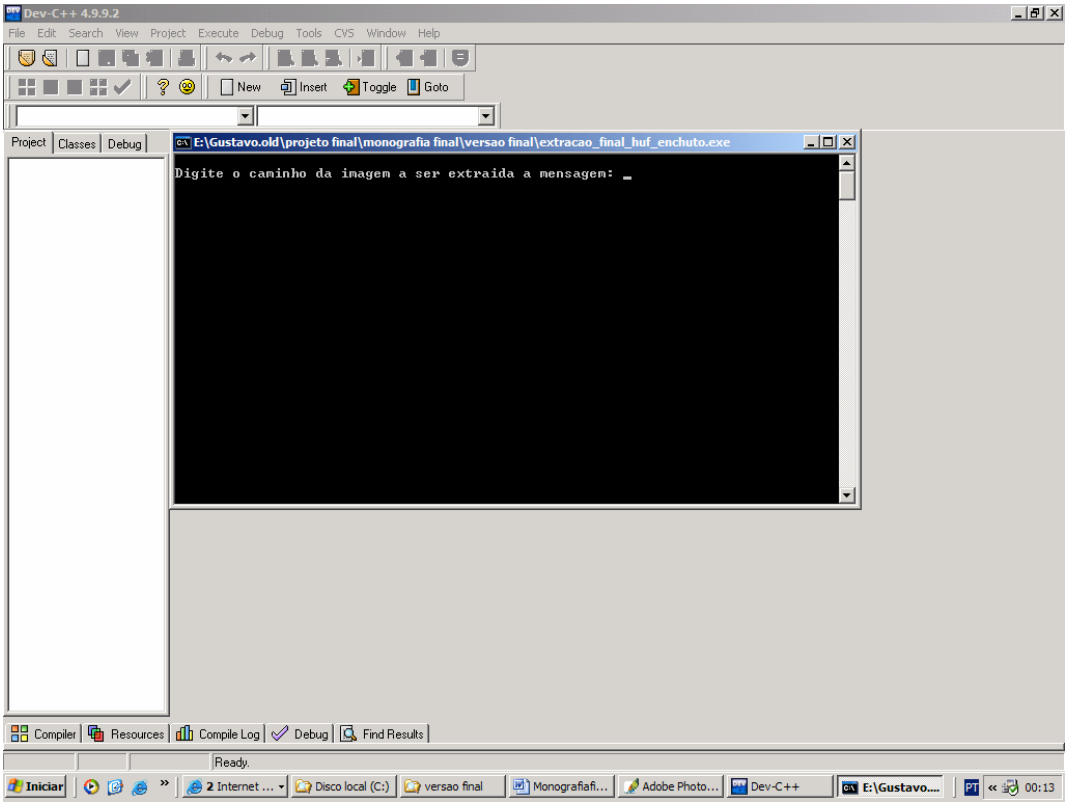

Figura 6.6 - Digitando o caminho da imagem a qual será extraída a mensagem

Fonte: Autor

O usuário tendo digitado o caminho correto, terá posteriormente uma tela que identifica a imagem como esteganografada, informando o seu tamanho em bits e se apresenta ou não a codificação de Huffman. Em caso afirmativo de conter a codificação de Huffman, é exibida a mensagem codificada, sendo essa posteriormente decodificada. Após a decodificação, será exibido na tela o tamanho da mensagem decodificada e em seguida a própria mensagem. Isso possibilitará a visualização da mensagem a qual foi inserida pelo usuário na imagem. Segue abaixo a tela de exibição.

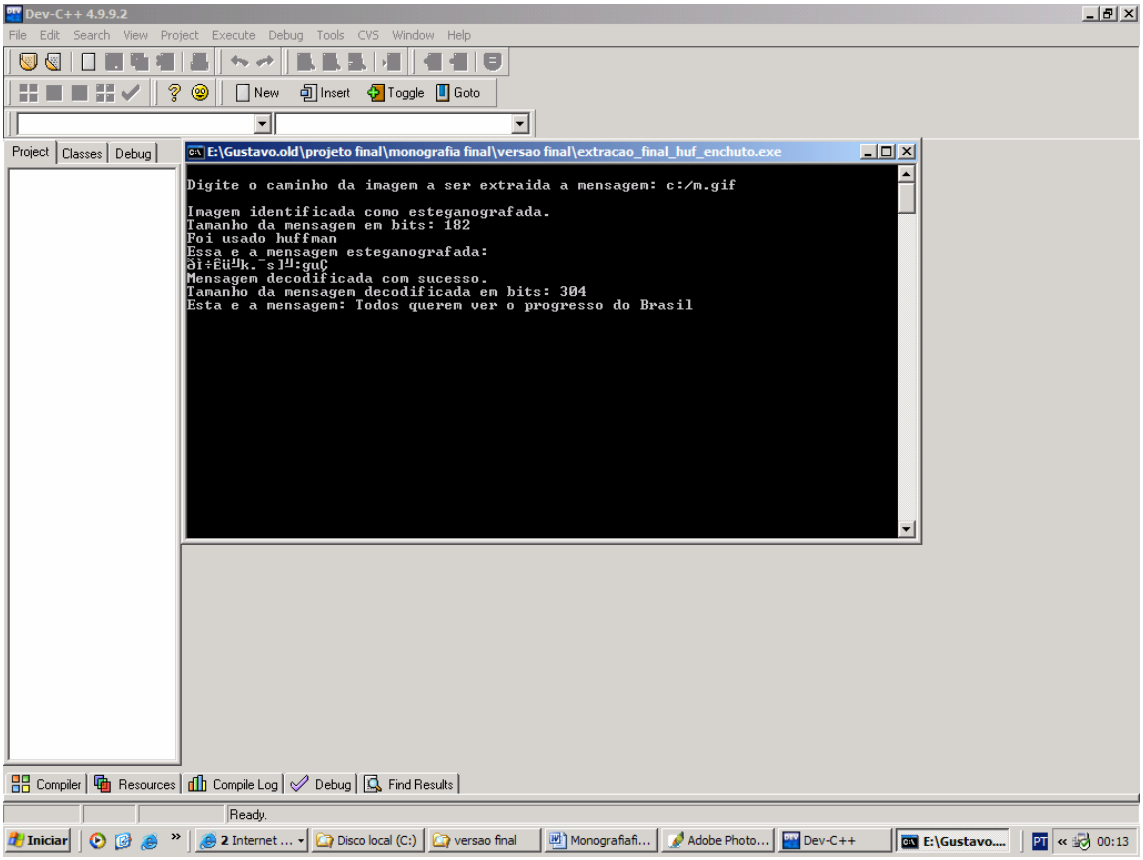

Figura 6.7 - Término do processo de extração da mensagem

Fonte: Autor

Tendo-se o processo de extração finalizado e o mesmo ocorrido com sucesso, pode-se concluir que a inserção também ocorreu com bom êxito.

7. Capitulo 7 - Testes e Simulações

Nas simulações e testes foram utilizadas imagens tanto monocromáticas quanto coloridas para mostrar que é possível trabalhar com ambas. Isso dá ao usuário a opção de escolher aquilo que lhe for mais conveniente ou de seu gosto utilizar.

Os resultados exibidos nas figuras abaixo tiveram o auxílio de um programa de visualização e editoração de imagens, no caso como já mencionado anteriormente foi utilizado o Photoshop 8.0. Com ele, foi possível criar os histogramas que nos dão uma noção das modificações ocorridas após a inserção de dados na imagem, podendo-se comparar as imagens originais com as esteganografadas.

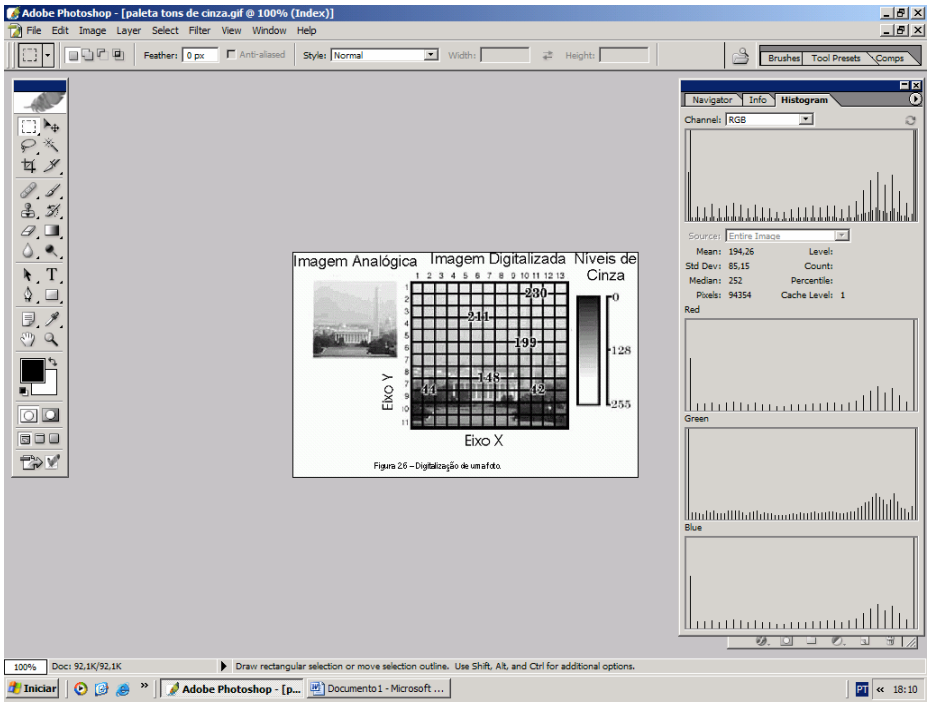

Figura 7.1 - Paleta tons de cinza (original)

Fonte: Autor

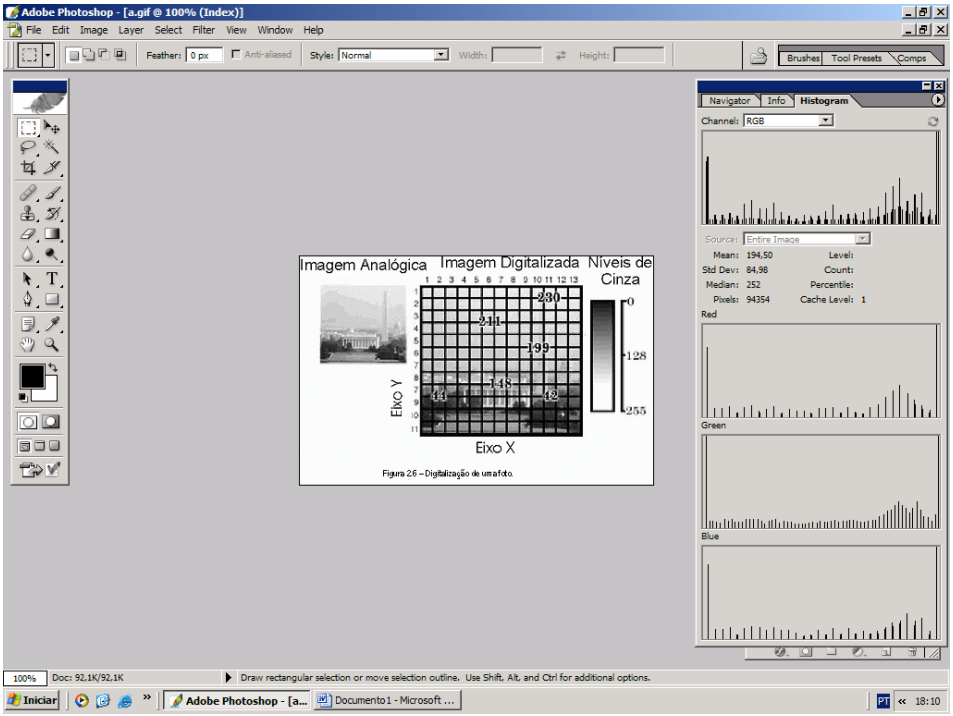

Figura 7.2 - Paleta em tons de cinza (esteganografada)

Fonte: Autor

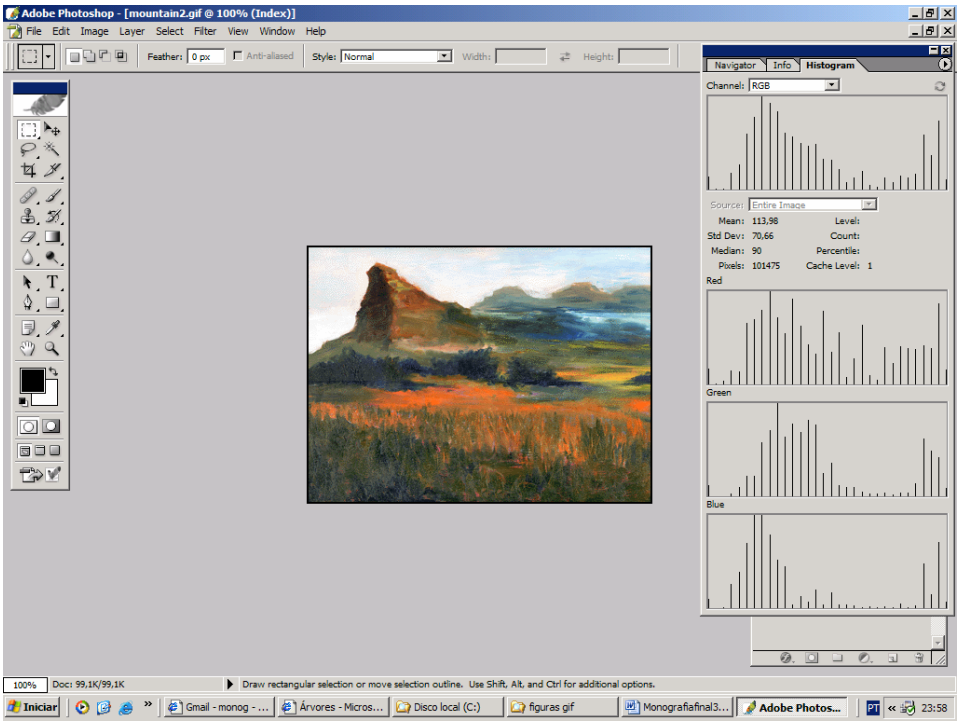

Figura 7.3 - Montanha (original)

Fonte: Autor

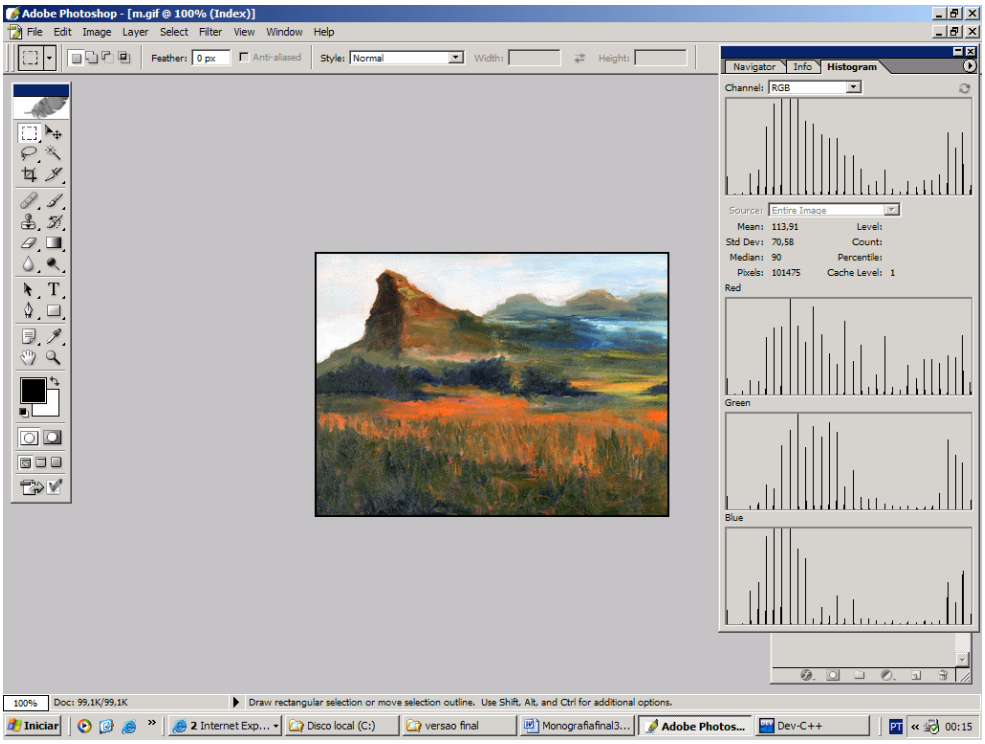

Figura 7.4 - Montanha (esteganografada)

Fonte: Autor

8. Capítulo 8 - Conclusão

8.1 - Considerações Finais

O projeto apresentou uma síntese com relação à esteganografia, desde a sua história a algumas de suas aplicações mais atuais. Foram mostradas algumas das principais técnicas, tendo-se em especial a do LSB em imagens.

Além disso, foi mostrado o protótipo que faz a inserção de texto em uma imagem do formato GIF que conta com o auxilio do algoritmo de Huffman para a compressão de texto.

O protótipo implementado como tem disponibilizado todo o código fonte, facilitará futuros trabalhos acadêmicos relacionados ao tema da esteganografia que é um assunto hoje em dia bem atual. Isso porque a segurança cada vez vêm sendo um dos requisitos essenciais ao se lidar com informações.

8.2 - Projetos Futuros

8.2.1 - Outros Tipos de Arquivos

A técnica de esteganografia pode ser utilizada além de arquivos de imagens, com arquivos de audio, video e texto. Esses tipos de arquivos seriam outras formas interessantes de estar empregando a técnica de esteganografia.

8.2.2 - Criptografia

O uso da criptografia em conjunto com a esteganografia garantirá maior segurança a informação, pois ficará muito mais difícil para pessoas não autorizadas terem acesso a informação. Isso porque a mensagem além de estar ilegivel, estará também ocultada.

8.2.3 - Códigos corretores de erros

A criação de códigos corretores de erros seria uma solução interessante para se recuperar uma mensagem contida numa imagem após ter sofrido um ataque geométrico.[14]

[14] Ataque geométrico – tipo de ataque que modifica as dimensões da imagem.

Referências Bibliográficas

AMIM, M. M.; SALLEH, M.; IBRAHIM, S.; KATMIN, M. R.; SHAMSUDDIN, M. Z. I. et. al. Information Hiding using Steganography. Artigo IEEE Computer Science 4º National Conference on Telecommunication Technology Proceedings, p. 21-25, 2003.

BENTO, Ricardo J.; COELHO, Laura M. Ferramentas de Esteganografia e seu uso na Infowar. ICCyber 2004 I Conferencia Internacional de Perícia em Crimes Cibernéticos.

CASACURTA, Alexandre. *Computação Gráfica - Introdução*. Disponível em: http://www.inf.unisinos.br/~osorio/CG-Doc/cg.pdf. Acesso em: 20 abril 2006.

COLE, Eric. *Hiding in Plain Sight:Steganography and the Art of Covert Communication.* Editora Wiley Publishing. Indiana, 2003.

CRUZ, Adriano. Algoritmo de Huffman. <http://equipe.nce.ufrj.br/adriano/c/apostila/arvore.htm#huffman> - Acesso em 15 maio 2006.

GONZALEZ, Rafael. *Digital Image Processing.* Editora Prentice Hall. New Jersey, 2002.

HAYASHIDA, Ulisses. *Código Huffman*. Disponível em: <http://www.linux.ime.usp.br/~cef/mac499-01/monografias/ulisses/#_5.1__Código> - Acesso em 15 maio 2006.

KESSLER, Gary. *Steganography*. Disponível em <http://www.garykessler.net/library/steganography.html> - Acesso em 9 abril 2006.

JASCONE, Fábio Luis Tavares. Protótipo de Software para Ocultar Texto Criptografado em Imagens Digitais. Blumenau, 2003. Trabalho de Conclusão de Curso – Ciências da Computação, Universidade Regional de Blumenau, p.33-40.

JOHNSON, N.; JAJODIA, S. Exploring steganography: seeing the unseen. In: IEEE Internet Computing. [S.l.: s.n.], 1998. v. 31, p. 26–34.

MISAGHI, Mehran. *Segurança e auditoria em informática, 2004*. Disponível em: http://www.vision.ime.usp.br/ mehran/ensino/20042.html. Acesso em: 10 abril 2006.

MÜLLER, Didier. *Stéganographie*. Disponível em: http://www.apprendre-en-ligne.net/crypto/stegano/ > - Acesso em 25 março 2006.

PARCONSULT. *Percepção Visual Humana*. Disponível em: <http://www.parconsult.com.br/uff/visao.htm> - Acesso em 12 abril 2006.

PETICOLAS, R. J.; ANDERSON, F. A. P. et. al. On the limits of steganography. Artigo IEEE Journal of Selected Areas in Comunications, Vol. 16, p. 474-481, março 1998.

POPA, R. An analysis of steganography techniques. Dissertação (Mestrado) — Department of Computer Science and Software Engineering of The Polytechnic University of Timisoara, Timisoara, Romênia, 1998.

SCHNEIER, Bruce. *Applied Cryptography*. John Wiley Sons, Inc., New Jersey,1996.

THING, Lowell. *The whatis?com Encyclopedia of Technology Terms*. Editora Futura. São Paulo, 2003.

WIKIPEDIA. *Codificação Huffman.* Disponível em <http://pt.wikipedia.org/wiki/Codifica%C3%A7%C3%A3o_de_Huffman> - Acesso em 18 maio 2006.

Anexo A - Estrutura do formato GIF

Bloco: **Header**

Tamanho: **6 bytes**

Identificador: **não tem**

Bytes	Formato	Conteúdo	Descrição
0 a 2	3 bytes alfabético	GIF	Identificação do arquivo.
3 a 5	3 bytes alfabético	87a ou 89a	Versão mínima que tenha funcionalidades necessárias para processar este fluxo.

Bloco: **DESCRITOR DE TELA LÓGICO**

Tamanho: **7 bytes**

Identificador: **não tem**

Bytes	Formato	Conteúdo	Descrição
0 e 1	inteiro sem sinal	Largura da imagem em pixels	Valor máximo 65.536
2 e 3	inteiro sem sinal	Altura da imagem em pixels	Idem
4	bit 0	Indicador de tabela de cores global	0 = não há 1 = vem a seguir
	bits 1, 2 e 3	Resolução de cores	Dá o número de bits (menos 1) que cada cor original tem. Assim, 7 aqui, indicará 8 bits para verde, 8 para azul e 8 para vermelho, ou seja true-color.
	bit 4	Indicador de classificação da tabela global de cores	Se =1, a tabela global está classificada com as cores mais freqüentes no começo
	bits 5, 6 e 7	Tamanho da tabela global de cores	Para conhecer este número calcule 2 elevado ao número que está aqui, mais 1. Assim, se tiver 0 lá, serão 2 cores (2^{0+1}=2).
5	byte	Índice da cor de background	Se o indicador de tabela de cores global é zero, este campo é ignorado e deve ser zero.

6	byte	Razão de aspecto	Número que varia entre 1:4 e 4:1 e determina a relação entre altura e largura da imagem original.

Bloco: **Tabela de Cores Global**

Tamanho: **3 x 2** **tamanho da tabela global de cores + 1**

Identificador: **não tem**

Bytes	Formato	Conteúdo	Descrição
Varia de 6 a 768	bytes	cores	Representação das cores

Bloco: **DESCRITOR DE IMAGEM**

Tamanho: **10 bytes**

Identificador: **0x2C**

Obrigatório ao menos 1

Bytes	Formato	Conteúdo	Descrição
0	byte, 0x2C	Separador de imagens	
1 e 2	inteiro sem sinal	Posição esquerda da imagem	Deslocamento em pixels em relação a margem esquerda da tela.
3 e 4	inteiro sem sinal	Posição do topo da imagem	Deslocamento em pixels em relação a margem superior da tela.
5 e 6	inteiro sem sinal	Largura da imagem	Em pixels.
7 e 8	inteiro sem sinal	Altura da imagem	Idem.
9	bit 0	Indicador de tabela de cores local	0 = não há 1 = vem a seguir deste descritor
	bit 1	Imagem entrelaçada	0 = não entrelaçada 1 = entrelaçada
	bit 2	Indicador de classificação da tabela de cores local	Se =1, a tabela local está classificada com as cores mais freqüentes no começo.

	bits 3 e 4	Reservado	Zeros.
	bits 5, 6 e 7	Tamanho da tabela local de cores	Para conhecer este número calcule 2 elevado ao número que está aqui, mais 1. Assim, se tiver 0 lá, serão 2 cores ($2^{0+1} = 2$).

Bloco: **Tabela Local de Cores**

Tamanho: **3 x 2** **tamanho da tabela local de cores + 1**

Identificador: **não tem**

Bytes	Formato	Conteúdo	Descrição
Varia de 6 a 768	bytes	cores	Representação das cores

Bloco: **Dados da Imagem**

Tamanho máximo dos sub-blocos: **255 bytes**

Tamanho total: **Depende da imagem**

Identificador: **não tem**

Bytes	Formato	Conteúdo	Descrição
Primeiro	bytes	Número inicial de bits usados pelo código LZW	
Máximo 255	bytes	Índices dentro da tabela de cores ativa para cada pixel.	Ordem: da esquerda pra direita, de cima pra baixo. Utiliza compressão LZW.

Bloco: **Extensão do Controle**

Tamanho: **8 bytes**

Identificador: **0x21**, **0xF9**

Bytes	Formato	Conteúdo	Descrição
0	byte	0x21	Identificador de extensão.
1	byte	0xF9	Identificador de extensão de controle.

0	byte	Tamanho deste bloco	Sempre igual a 4.
1	bit 0 a 2	Reservado.	Zeros.
	bits 3, 4 e 5	Método de substituição.	0 = nada a fazer 1 = não há substituição. O gráfico deve ser deixado onde está. 2 = a área usada pelo gráfico deve ser retornada à cor de background.
	bit 6	Aguardar ação do operador.	Se 0 não aguarda operação do operador. Se = 1, aguarda (enter, click, etc).
	bit 7	Cor para transparência.	Se 0 não é estabelecido uma cor para transparência. Se =1 há uma cor transparente.
2 e 3	Inteiro sem sinal	Retardo.	Especifica a quantidade de centésimos de segundo que se deve esperar antes de continuar a processar o fluxo.
4	byte	Cor transparente.	(Se bit7 = 1) indica a cor que, quando presente, deve deixar o que já existe naquele pixel, seguindo o processamento para o próximo.

0	0x00	Terminador do bloco.	

Bloco: **Extensão de Comentário**

Tamanho: **Depende da Imagem**

Identificador: **0x21**, **0xFE**

Usada para descrever textualmente informações na imagem.

0	1
0x21	0xFE

1 a 255
bloco

1 a 255
bloco

0
0x00