

**UNIVERSIDADE FEDERAL FLUMINENSE
GERALDO JOSÉ FERREIRA CHAGAS JÚNIOR**

ESTEGANOGRAFIA EM MÍDIAS DIGITAIS

**Niterói
2015**

GERALDO JOSÉ FERREIRA CHAGAS JÚNIOR

ESTEGANOGRÁFIA EM MÍDIAS DIGITAIS

Trabalho de Conclusão de Curso submetido ao Curso de Tecnologia em Sistemas de Computação da Universidade Federal Fluminense como requisito parcial para obtenção do título de Tecnólogo em Sistemas de Computação.

Orientador:

Joao Gabriel Felipe Machado Gazolla

NITERÓI

2015

GERALDO JOSÉ FERREIRA CHAGAS JÚNIOR

ESTEGANOGRÁFIA EM MÍDIAS DIGITAIS

Trabalho de Conclusão de Curso submetido ao Curso de Tecnologia em Sistemas de Computação da Universidade Federal Fluminense como requisito parcial para obtenção do título de Tecnólogo em Sistemas de Computação.

Niterói, 13 de Novembro de 2015.

Banca Examinadora:

Prof. Joao Gabriel Felipe Machado Gazolla MSc. – Orientador
UFF – Universidade Federal Fluminense

Prof. Daniel Luiz Alves Madeira DSc. – Avaliador
UFSJ – Universidade Federal de São João del-Rei

Dedico este trabalho aos meus afilhados Maria, Sophia e Ítalo; e também aos dois futuros membros da família; meus sobrinhos Paloma e Fernando, que em breve estarão a nosso lado.

AGRADECIMENTOS

A meu Orientador Gabriel Gazolla pelo estímulo e atenção que me concedeu durante o desenvolvimento do trabalho.

A meus pais Rita e Geraldo que nunca mediram esforços para garantir minha educação, entregando-me de forma antecipada a maior herança que poderiam deixar.

A minha irmã Renata (Xinha) que sempre esteve ao meu lado, independente da distância.

A minha esposa Vanessa, por todo apoio, paciência, incentivo e compreensão nesses anos de batalha, convivendo, por diversas vezes, com minha ausência, que se fazia necessária.

“Uma pessoa humilde é aquela que sabe que não sabe tudo. Aquela que sabe que não é a única que sabe. Aquela que sabe que outra pessoa sabe o que ela não sabe. Aquela que sabe que ela e outra pessoa saberão muita coisa juntas. Aquela que sabe que ela e outra pessoa nunca saberão tudo que pode ser sabido”.

Mário Sérgio Cortella

RESUMO

Este trabalho tem o intuito de realizar um breve estudo sobre o tema esteganografia, percorrendo sua história, apresentando técnicas e possibilidades futuras; descrevendo os formatos mais comuns de arquivos digitais utilizados atualmente; e desenvolvendo um *software* que colocará em prática alguns métodos esteganográficos permitindo uma análise dos resultados obtidos.

Palavras-chaves: esteganografia, ocultar, arquivos.

ABSTRACT

This paper aims to conduct a brief study on the subject steganography, through your history, showing techniques and future possibilities; describing the most common digital file formats currently used; and developing software that will implement some steganographic methods that will allow an analysis of the results.

Key words: **steganography, hide, files.**

LISTA DE ILUSTRAÇÕES

Figura 1: Exemplo de Esteganografia Genômica [24].	37
Figura 2: Anverso de uma cédula de R\$ 10,00.	40
Figura 3: Reverso de uma cédula de R\$ 10,00.	40
Figura 4: Marca d'água na cédula de R\$ 10,00 estampa C.	41
Figura 5: Microimpressões das letras "B" e "C" na cédula de R\$ 10,00.	41
Figura 6: Imagem latente das letras "B" e "C" na cédula de R\$ 10,00.	41
Figura 7: Fibras luminosas na cédula de R\$ 10,00.	42
Figura 8: Visualização dos pontos ocultos de uma impressão [28, p.15].	43
Figura 9: Parte da foto de Lena contendo a referência à propriedade da <i>Playboy</i> [26, p.21].	46
Figura 10: Ideia da câmera digital segura [29, p.1198].	48
Figura 11: Ampliação do círculo para visualização dos <i>pixels</i>	52
Figura 12: Maior densidade de <i>pixel</i> no círculo à esquerda.	53
Figura 13: FHSS aplicada a um compartilhamento de banda [44, p.184].	63
Figura 14: Obtenção do sinal espalhado utilizando-se a técnica DSSS, salientando o maior ritmo binário do código de espalhamento [44, p.185].	64
Figura 15: Diagrama de Caso de Uso	71
Figura 16: Diagrama de atividades do caso de uso ocultação do dado.	75
Figura 17: Diagrama de atividades do caso de uso de recuperação de dado.	76
Figura 18: Diagrama de classes do <i>software</i> proposto.	77
Figura 19: Diagrama de Sequência para a ocultação de mensagem texto utilizando a técnica de final de arquivo.	79
Figura 20: Diagrama de Sequência para a ocultação de arquivo utilizando a técnica de final de arquivo.	80
Figura 21: Diagrama de Sequência para a ocultação de mensagem texto utilizando as técnicas de LSB, LSB 2, LSB 3, LSB n, LSB Cíclico.	81
Figura 22: Diagrama de Sequência para a ocultação de arquivo utilizando as técnicas LSB, LSB 2, LSB 3, LSB n, LSB Cíclico.	83

Figura 23: Diagrama de Sequência para a recuperação de mensagem texto quando utilizada a técnica de final de arquivo.	84
Figura 24: Diagrama de Sequência para a recuperação de arquivo quando utilizada a técnica de final de arquivo.	85
Figura 25: Diagrama de Sequência para a recuperação de mensagem texto quando utilizada as técnicas LSB, LSB 2, LSB 3, LSB n, LSB Cílico.	86
Figura 26: Diagrama de Sequência para a recuperação de arquivo quando utilizada as técnicas LSB, LSB 2, LSB 3, LSB n, LSB Cílico.	87
Figura 27: Tela inicial do <i>software</i> com a guia "Ocultar Dados" selecionada.	92
Figura 28: Tela inicial do <i>software</i> com a guia "Recuperar Dados" selecionada.	94
Figura 29: Tela do <i>software</i> em processo de ocultação.	95
Figura 30: Tela do <i>software</i> em processo de recuperação de informação esteganografada.	96
Figura 31: Tela do <i>software</i> exibindo a realização da recuperação do dado.	97
Figura 32: Código em Java do procedimento utilizado para concatenar dois <i>arrays</i> de <i>bytes</i> .	99
Figura 33: Código em Java do procedimento utilizado para ocultar mensagem em final de arquivo.	100
Figura 34: Código em Java do procedimento utilizado para ocultar arquivo em final de arquivo.	100
Figura 35: Código em Java do procedimento utilizado para recuperar mensagem em final de arquivo.	100
Figura 36: Código em Java do procedimento utilizado para recuperar arquivo em final de arquivo.	101
Figura 37: Código em Java dos procedimentos utilizados para recuperar (getBit) e alterar (alteraBit) um <i>bit</i> qualquer em uma cadeia de <i>bytes</i> . O valor 1 é representado por <i>true</i> e 0 por <i>false</i> .	102
Figura 38: Código em Java do procedimento utilizado para ocultar <i>bytes</i> com a técnica LSB.	103
Figura 39: Código em Java do procedimento utilizado para recuperar <i>bytes</i> com a técnica LSB.	104
Figura 40: Código em Java do procedimento utilizado para ocultar <i>bytes</i> com a técnica LSB Cílico.	105

Figura 41: Código em Java do procedimento utilizado para recuperar <i>bytes</i> com a técnica LSB Cíclico.....	105
Figura 42: Código em Java do procedimento utilizado para ocultar <i>bytes</i> com a técnica LSB n.....	106
Figura 43: Código em Java do procedimento utilizado para recuperação <i>bytes</i> com a técnica LSB n.....	107
Figura 44: Arquivo BMP de 1.390 KB original.....	111
Figura 45: Arquivo BMP de 1.390 KB com um arquivo PDF de 60 KB esteganografado.....	111
Figura 46: Representação gráfica das ondas de áudio do arquivo WAV original....	111
Figura 47: Representação gráfica das ondas de áudio do arquivo WAV contendo um arquivo JPG de 10 KB esteganografado.....	111
Figura 48: Arquivo BMP de 215 KB original.....	112
Figura 49: Arquivo BMP de 215 KB com um arquivo JPG de 10 KB esteganografado.	112
Figura 50: Arquivo BMP de 1.390 KB original.....	113
Figura 51: Arquivo BMP de 1.390 KB com um arquivo JPG de 200 KB esteganografado.....	113
Figura 52: Representação gráfica das ondas de áudio do arquivo WAV original....	114
Figura 53: Representação gráfica das ondas de áudio do arquivo WAV contendo um arquivo JPG de 48 KB esteganografado.....	114
Figura 54: Arquivo BMP de 215 KB original.....	114
Figura 55: Arquivo BMP de 215 KB com um arquivo JPG de 48 KB esteganografado.	114
Figura 56: Arquivo BMP de 215 KB original.....	116
Figura 57: Arquivo BMP de 1.390 KB original.....	116
Figura 58: Arquivo BMP de 215 KB com um arquivo PDF de 61 KB esteganografado com a técnica LSB 3.....	117
Figura 59: Arquivo BMP de 1.390 KB com um arquivo WAV de 217 KB esteganografado com a técnica LSB 3.....	117
Figura 60: Arquivo BMP de 215 KB com um arquivo PDF de 61 KB esteganografado com a técnica LSB n; para $n = 4$	117
Figura 61: Arquivo BMP de 1.390 KB com um arquivo WAV de 217 KB esteganografado com a técnica LSB n; para $n = 4$	117

Figura 62: Arquivo BMP de 215 KB com um arquivo PDF de 61 KB esteganografado com a técnica LSB n; para $n = 5$.	118
Figura 63: Arquivo BMP de 1.390 KB com um arquivo WAV de 217 KB esteganografado com a técnica LSB n; para $n = 5$.	118
Figura 64: Arquivo BMP de 215 KB com um arquivo PDF de 61 KB esteganografado com a técnica LSB n; para $n = 6$.	118
Figura 65: Arquivo BMP de 1.390 KB com um arquivo WAV de 217 KB esteganografado com a técnica LSB n; para $n = 6$.	118
Figura 66: Arquivo BMP de 1.390 KB original.	120
Figura 67: Arquivo BMP de 1.390 KB com um arquivo DOC de 1.019 KB esteganografado.	120
Figura 68: Arquivo BMP de 215 KB original.	120
Figura 69: Arquivo BMP de 215 KB com um arquivo PDF de 48 KB esteganografado.	120
Figura 70: Arquivo BMP de 1.390 KB original.	122
Figura 71: Arquivo BMP de 1.390 KB com um arquivo DOC de 1.019 KB esteganografado.	122
Figura 72: Arquivo BMP de 215 KB original.	122
Figura 73: Arquivo BMP de 215 KB com um arquivo PDF de 48 KB esteganografado.	122
Figura 74: Arquivo beep.wav visualizado em um editor de texto.	129
Figura 75: Arquivo Esteganografia.txt visualizado em um editor de texto.	129
Figura 76: O arquivo beep.wav com a mensagem esteganografada.	129

LISTA DE TABELAS

Tabela 1: Codificação para Ave Maria de Trithemius.....	29
Tabela 2: Cabeçalho do arquivo no formato BMP.....	54
Tabela 3: Cabeçalho de mapa de <i>bits</i> do arquivo BMP.	54
Tabela 4: Paleta cores (opcional) do arquivo BMP.	55
Tabela 5: Cabeçalho do arquivo no formato WAV.	58
Tabela 6: Bloco fmt do arquivo WAV.....	58
Tabela 7: Bloco de dados do áudio do arquivo WAV.	59
Tabela 8: Tabela representando os campos e tamanho em <i>bytes</i> ocupados em um arquivo de mídia com a mensagem UFF esteganografada.....	89
Tabela 9: Tabela representando os campos e tamanho em <i>bytes</i> ocupados em um arquivo de mídia com um arquivo PDF esteganografado.	89
Tabela 10: Representação dos valores possíveis para o <i>byte</i> de identificação do tipo de esteganografia utilizada.	90
Tabela 11: Representação dos valores decimais, hexadecimais e binários do caracteres T,C,U e F.....	90

LISTA DE GRÁFICOS

Gráfico 1: Número anual de publicações sobre marca d'água e esteganografia realizadas pela IEEE [26, p.9] 39

LISTA DE ABREVIATURAS E SIGLAS

A – Adenina

A.C. – Antes de Cristo

AVI – *Audio Video Interleave*

BMP – *Bit map*

BPCS – *Bit-Plane Complex Segmentation*

C – Citosina

CD – *Compact Disc*

CPF – Cadastro de Pessoas Físicas

CSS – *Cascading Style Sheets*

D.C. – Depois de Cristo

DFT – *Discrete Fourier Transform*

DICOM – *Digital Imaging and Communications in Medicine*

DNA – *Deoxyribonucleic Acid*

DSSS – *Direct Sequence Spread Spectrum*

DVD – *Digital Versatile Disc*

FBI – *Federal Bureau of Investigation*

FHSS – *Frequency Hopping Spread Spectrum*

G – Guanina

GB – *Giga Bytes*

HTML – *HyperText Markup Language*

IBM – *International Business Machines*

ID – Identificador

IEEE – *Institute of Electrical and Electronics Engineers*

JPEG – *Joint Photographic Experts Group*

KB – *Kilo Byte*

LSB – *Least Significant Bit*

LZW – *Lempel-Ziv-Welch*

MB – *Mega Byte*

MM – Milímetro

MP3 – *MPEG-1/2 Audio Layer 3*

MPEG – *Moving Picture Experts Group*

NASA – *National Aeronautics and Space Administration*

OO – Orientação à Objetos

P2P – *peer-to-peer*

PC – *Personal computer*

PCM – *Pulse Code Modulation*

PDF – *Portable Document Format*

PNG – *Portable Network Graphics*

RGB – *Red, Green e Blue*

RLE – *Run-length encoding*

SAG – *Screen Actor's Guild*

SALT – *Strategic Arms Limitation Talks*

SVG – *Scalable Vector Graphics*

T – Timina

TIF – *Tag Image File Format*

TIFF – *Tag Image File Format*

UML – *Unified Modeling Language*

UV – Ultravioleta

VBI – *Vertical Blanking Interval*

WAV – *Waveform Audio Format*

WMA – *Windows Media Audio*

SUMÁRIO

RESUMO.....	7
ABSTRACT	8
LISTA DE ILUSTRAÇÕES	9
LISTA DE TABELAS	13
LISTA DE GRÁFICOS.....	14
LISTA DE ABREVIATURAS E SIGLAS	15
1 INTRODUÇÃO.....	21
1.1 MOTIVAÇÃO.....	22
1.2 OBJETIVO.....	24
1.3 ORGANIZAÇÃO DO TRABALHO	24
2 HISTÓRICO	26
2.1 ENÉAS, O TÁTICO	26
2.2 A ESTEGANOGRÁFIA DE TRITHEMIUS	28
2.2.1 AVE MARIA DE TRITHEMIUS.....	28
2.3 ESTEGANOGRÁFIA – PASSADO, PRESENTE E FUTURO.....	31
2.3.1 PASSADO DA ESTEGANOGRÁFIA.....	31
2.3.2 PRESENTE DA ESTEGANOGRÁFIA.....	32
2.3.3 FUTURO DA ESTEGANOGRÁFIA	35
3 APLICAÇÕES	39
3.1 SEGURANÇA MONETÁRIA.....	40
3.2 IMPRESSORAS MODERNAS.....	42
3.3 MONITORAMENTO DE TRANSMISSÃO	43
3.4 PROPRIEDADE INTELECTUAL	45
3.5 PROVA DE PROPRIEDADE	47
3.6 AUTENTICAÇÃO	47
3.7 CONTROLE DE CÓPIAS	48
3.8 COMUNICAÇÕES SECRETAS.....	50

4	FUNDAMENTAÇÃO TEÓRICA	52
4.1	IMAGEM DIGITAL	52
4.1.1	FORMATOS DE ARQUIVOS.....	53
4.2	AUDIO DIGITAL	57
4.2.1	FORMATO DO ARQUIVO WAV	57
4.3	TÉCNICAS DE ESTEGANOGRAFIA E SEU ALGORÍTMOS.....	59
4.3.1	FINAL DO ARQUIVO	59
4.3.2	LSB	60
4.3.3	ESTEGANOGRAFIA EM PALETA DE IMAGEM	62
4.3.4	ESPALHAMENTO DE ESPECTRO.....	62
4.3.5	EM ÁUDIO	65
4.4	ESTEGANÁLISE	66
5	IMPLEMENTAÇÃO	68
5.1	DEFINIÇÃO DO PROBLEMA.....	68
5.2	REQUISITOS	69
5.3	ESPECIFICAÇÃO	71
5.3.1	DIAGRAMA DE CASO DE USO	71
5.3.2	DESCRIÇÃO DE CASO DE USO	72
5.3.3	DIAGRAMA DE ATIVIDADE	75
5.4	MODELAGEM DE DADOS.....	76
5.4.1	DIAGRAMA DE CLASSES	77
5.4.2	DIAGRAMAS DE SEQUÊNCIA	78
5.5	VISÃO GERAL DO SOFTWARE.....	88
5.6	INTERFACE DO SOFTWARE.....	92
5.7	LIMITAÇÕES.....	98
5.8	ALGORÍTMOS.....	99
6	RESULTADOS DA IMPLEMENTAÇÃO.....	108
6.1	EXPERIMENTO 01	108
6.1.1	CENÁRIO.....	109
6.1.2	RESULTADOS.....	109
6.1.3	ANÁLISE DOS RESULTADOS	109
6.2	EXPERIMENTO 02	110
6.2.1	CENÁRIO.....	110
6.2.2	RESULTADOS.....	110

6.2.3	ANÁLISE DOS RESULTADOS.....	112
6.3	EXPERIMENTO 03	112
6.3.1	CENÁRIO.....	113
6.3.2	RESULTADOS.....	113
6.3.3	ANALISE DOS RESULTADOS.....	115
6.4	EXPERIMENTO 04	115
6.4.1	CENÁRIO.....	115
6.4.2	RESULTADOS.....	116
6.4.3	ANALISE DOS RESULTADOS.....	118
6.5	EXPERIMENTO 05	119
6.5.1	CENÁRIO.....	119
6.5.2	RESULTADOS.....	120
6.5.3	ANALISE DOS RESULTADOS.....	121
6.6	EXPERIMENTO 06	121
6.6.1	CENÁRIO.....	121
6.6.2	RESULTADOS.....	121
6.6.3	ANALISE DOS RESULTADOS.....	123
7	CONCLUSÃO E TRABALHOS FUTUROS	124
	REFERÊNCIAS BIBLIOGRÁFICAS	125
	ANEXOS	129
	ANEXO A – Demonstração de esteganografia em final de arquivo.....	129

1 INTRODUÇÃO

Esteganografia é uma palavra de etimologia grega proveniente das palavras *estegano* (*στεγανό*) que significa oculto, secreto, esconder, mascarar; e *grafí* (*γραφή*) que significa escrita. Apesar de etimologicamente originar-se do grego, em geral traduz-se do inglês *steganography* oriundo do livro de título *Steganographia* escrito por Johannes Trithemius em 1518. Dito isto, podemos definir a esteganografia como sendo a ocultação da escrita, ou seja, a arte de se esconder uma informação [3, p.418].

Apesar de em um primeiro momento a esteganografia se assemelhar com a criptografia, ambas têm paradigmas completamente diferentes¹. Enquanto a criptografia se baseia em tornar a informação ilegível [2, p.19], permitindo que um intermediário perceba a existência do tráfego, mas não possa interpretá-la, a esteganografia se preocupa em tornar a informação oculta no meio de transmissão, desta forma o objetivo é que agentes intermediários não consigam identificar a existência de uma mensagem camouflada entre o remetente e o destinatário [4].

A simples utilização da criptografia, em muitos casos, pode não ter o resultado desejado, já que permite que uma mensagem seja impedida de chegar ao seu destinatário, pois por melhor que seja a criptografia, essa não impede que um intermediário perceba a tentativa de comunicação entre as partes de origem e destino, sendo assim, mesmo que não possa compreender o conteúdo da informação, se possível e de interesse, a comunicação pode ser interrompida, não permitindo com que o caminho a ser percorrido seja completado. A esteganografia, por sua vez,

¹ É importante ressaltar que alguns autores como: Luis Pinochet em seu livro *Tecnologia da Informação e Comunicação* [1] e Eduardo Pagani Julio, Wagner Gaspar Brazil, Célio Vinicius Neves Albuquerque (UNIVERSIDADE FEDERAL FLUMINENSE) em *Esteganografia e suas Aplicações*. VII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais [9]; consideram a esteganografia como um ramo da criptografia. Este trabalho segue a vertente de autores que tratam esteganografia e criptografia como coisas distintas.

consegue ocultar essa tentativa de comunicação, porém, uma vez descoberta a tentativa, a mensagem fica exposta.

Uma solução para aumentar a segurança e corrigir os problemas da criptografia e esteganografia seria a união das duas técnicas de forma que a esteganografia seja realizada com uma informação criptografada [12].

“O único sistema verdadeiramente seguro é aquele que está desligado, fundido em um bloco de concreto, selado em uma sala revestida de chumbo com guardas armados – e mesmo assim tenho minhas dúvidas”. (Dewdney March, 1989) [5, p.21].

Dado o crescimento do armazenamento e tráfego de informação, a criptografia e a esteganografia vêm sendo largamente utilizadas por diversas aplicações – como: *sites de banco, watermarking e certificados digitais* - que necessitam garantir a segurança de seus dados [1].

1.1 MOTIVAÇÃO

A esteganografia vem sendo utilizada por milhares de anos, normalmente, através de técnicas baseadas mais na criatividade dos utilizadores do que em estudos científicos. Entretanto, com a disseminação da informação por meios digitais, a possibilidade de ocultação da informação em diversas formas de mídia tem substituído essas técnicas mais rústicas por técnicas mais sofisticadas - como utilização de teorias matemáticas - que criaram possibilidades, até então, improváveis de serem realizadas.

A Internet, uma grande invenção do século 20, está mudando a civilização do século 21. Seu poder cresce com os cabos de fibra óptica, cujos fios de vidro constituem os nervos da economia mundial. Nenhuma outra tecnologia permite maior velocidade de transmissão e gera mais ganhos de escala a tão baixo custo do que os cabos de fibra óptica. Graças a esse sistema, que armazena, organiza e compartilha informação no mundo todo, 90% dos dados disponíveis globalmente foram criados nos últimos dois anos. Em 2012, todos os dias 2,5 *exabytes* (1 seguido de 18 zeros) de dados foram transmitidos. Enquanto isso, e, junho de 2013, o número de usuários da Internet atingiu 2,4 bilhões, 34% da população do mundo, um aumento de 566% desde o ano 2000.

Esta enorme avalanche de dados deverá dobrar a cada dois anos até 2020, impulsionada pelo aumento do número de usuários da Internet e pelo crescente consumo e produção de vídeos, entre outros fatores. [6, p.4].

Datacenter, cloud computing, redes sociais, internet de alta velocidade, conectividade, essas palavras são determinísticas para a criação desta avalanche.

Atualmente, em uma realidade que se caracteriza pela grande quantidade de tráfego - em volume e velocidade - de informação, em um mundo cada vez mais interconectado, onde os dados podem ser interceptados, criando uma espécie de *reality show* digital visível aos especialistas que dominam as técnicas de análise de tráfego de dados e ao mesmo tempo invisível a olhos nus, o sigilo e segurança da informação tornam-se fundamentais para manutenção do modelo de comunicação utilizado.

A utilização de tecnologias digitais como: *scanner*, gravador de CD/DVD, placa de captura de vídeo, principalmente associados a computadores, possibilitou, também, novas formas de apropriação intelectual² não autorizada [1] de materiais, que assim como o tráfego de dados, necessitam de técnicas, que se não impedem, pelo menos minimizem a prática da pirataria³.

Neste contexto esteganografia encontra um ambiente interessante e vasto de utilização, tanto no meio acadêmico quanto no meio comercial, com uma diferença entre eles, enquanto o meio acadêmico defende a publicação e compartilhamento do conhecimento de técnicas e algoritmos a fim de refiná-los e aperfeiçoá-los, o ambiente comercial defende a ocultação dessas informações, para que não possam ser usadas de forma a atacar a vulnerabilidade de seus produtos [4].

² Propriedade intelectual - Trabalho criativo intangível que se materializa em forma física. Exemplo: uma invenção, um trabalho literário, uma arte, uma partitura musical, uma fotografia, entre outros. No campo da computação, o software tem sido protegido como propriedade intelectual por meio de direitos autorais e patentes. O hardware tem sido protegido apenas por patente [1].

³ Pirataria - refere-se à cópia, à venda ou a distribuição de material, com direitos autorais, de forma não autorizada. No Brasil, pirataria é considerada crime e possui lei específica [1][7].

1.2 OBJETIVO

Realizar um estudo da esteganografia demonstrando seus defeitos, qualidades e utilidades. Também faz parte desta produção a implementação e análise de um *software* que utiliza técnicas esteganográficas em diversos formatos de mídias digitais (como imagens e áudio) com intuito de demonstrar suas abrangências e limitações.

Ao final, espera-se alcançar as seguintes metas:

- Um entendimento sólido sobre esteganografia e suas aplicações.
- A implementação de um *software* que utiliza técnicas esteganográficas, que inclui um texto, imagem ou áudio em outro arquivo multimídia.
- A implementação de um *software* de recuperação do texto, imagem ou áudio cifrado em um arquivo multimídia. Uma análise detalhada sobre os sucessos, desafios, limitações e impossibilidades de cada caso implementado.

1.3 ORGANIZAÇÃO DO TRABALHO

O presente trabalho organiza-se sob a forma de sete capítulos sendo este, Capítulo 1, o capítulo de introdução. O Capítulo 2 traz um breve relato histórica sobre a esteganografia, mostrando algumas de suas funcionalidades desde os primórdios de sua utilização, percorrendo o tempo até apresentar possibilidades futuras para o uso das técnicas. O Capítulo 3 aprofunda-se nas aplicações utilizadas atualmente, mostrando o vasto campo onde esteganografia é inserida; suas técnicas, vulnerabilidades e observações. O Capítulo 4 apresenta toda fundamentação teórica necessária para implementação de soluções esteganográficas. Neste capítulo é descrito as estruturas de alguns arquivos de mídia; técnicas e algoritmos estegano-

gráficos; bibliotecas necessárias e esteganálise⁴. O Capítulo 5 se dedica a demonstrar o desenvolvimento da aplicação proposta neste trabalho, mostrando a parte relevante de sua construção. Nele será possível acompanhar pontos como: requisitos, interface, limitações e algoritmos relevantes. Aqui também serão apresentadas as dificuldades de seu desenvolvimento e suas limitações. O Capítulo 6 trará uma sequência de testes realizados com o aplicativo, mostrando vários cenários, seus resultados e a conclusão obtida em cada caso. Finalmente, no Capítulo 7, a conclusão deste trabalho e sugestões de trabalhos futuros.

⁴ Esteganálise – nome dado a técnicas que tentam identificar a existência da utilização da esteganografia. Esse termo será melhor elucidado no tópico 4.4 do presente trabalho.

2 HISTÓRICO

A esteganografia não é um conceito novo, seus relatos são encontrados em obras com milhares de anos e sua utilização presente até os dias atuais. Dentre os documentos conhecidos, dois autores merecem uma atenção exclusiva: Enéas por sua obra *Poliorketika* escrita no século IV A.C.; e Johannes Trithemius por sua obra *Steganographia* escrita no século XVI D.C..

2.1 ENÉAS, O TÁTICO

Enéas foi um o primeiro conselheiro militar ocidental a escrever sobre os métodos de criptografia. Escritor da Grécia antiga, tem como sua obra mais famosa *Poliorketika*, ou "Comentário Tático sobre como devem Defender-se dos Cercos", escrito em 357 A.C. em um dialeto comum helenístico⁵, o que o tornava popular e prático [8].

Além de que viveu durante o século IV A.C., muito pouco se sabe sobre a vida de Enéas. Há indícios de que tenha sido um general e que Tacticus seja um sobrenome honorário dado para distingui-lo dos demais, de mesmo nome, sendo seu nome original Enéas Stymphalos. Isso levaria a acreditar que Enéas, o Tático, tenha trabalhado como mercenário arcadiano⁶, antes de se tornar um comandante de tropas.

No livro, Enéas traz detalhes sobre a segurança da comunicação militar, embutindo conceitos de código e criptografia. Essas práticas já eram conhecidas

⁵ Período Helienístico – Época que começa com a morte de Alexandre Magno (323 A.C.) e que se estende até inícios do Império Romano [11].

⁶ Arcadianos - Estão entre os primeiros povos gregos a desenvolver uma classe de soldados pagos, mercenários [8].

desde a existência da inteligência militar, ou ainda anteriores, desde o contrabando de mercadorias preciosas; porém, foi ele, o primeiro autor a fornecer um guia completo sobre as técnicas de criptografia, recebendo, pela seção de esteganografia, grandes elogios.

Um exemplo de técnica esteganográfica descrita no livro era a ocultar a informação sob cera de quadro de mensagens. Em seu tempo, as mensagens eram escritas em cera ou em tábuas de madeira. A técnica consistia em escrever a mensagem secreta na tábua de madeira e aplicar uma camada de cera por cima. Sobre essa cera era escrita uma mensagem inócuia. Ao receber a tábua, o destinatário derretia a cera e a mensagem oculta poderia ser lida. Essa técnica era descrita nos escritos de Heródoto⁷ e utilizada para avisar sobre um ataque eminente à Grécia.

Um segundo método, parecido, consistia em escrever sobre a tábua com tinta preta e após secar aplicar um agente branqueador. Ao receber a placa, mergulhava-a em água para que a mensagem fosse revelada.

Outra técnica era a escrita de mensagem em bexiga de animal. Inflava-se a bexiga escrevia a mensagem, esvaziava e enviava em uma garrafa de óleo. Ao receber a bexiga, ela era inflada e a mensagem revelada. A mensagem poderia ser apagada e a resposta escrita na mesma bexiga.

Outra engenhosidade utilizada para burlar a inspeção utilizava uma atadura, de tiras de papiro, para transportar a informação. Essa atadura então era utilizada para cobrir ferimentos de soldados. Esse método utilizava o raciocínio de que poucos se atreveriam a desfazer a atadura ensanguenta de uma ferida infecionada.

Enéas descreve ainda a estratégia utilizada por Heródoto que tatuou uma mensagem na cabeça raspada de um escravo e enviou o escravo após o cabelo ter crescido, bastando raspar a cabeça para se conhecer a mensagem.

Peças de vestuário eram largamente utilizados, como esconder mensagens em brincos utilizados por mulheres ou inscrições dentro de sapatos.

Animais tinham grande preferência no transporte de mensagem, uma vez que não sucumbiam a pressão e não confessavam sob tortura.

⁷ Heródoto (485?-420 a.C.) foi um famoso geógrafo e historiador grego, nascido em Halicarnasso (hoje Bodrum, na Turquia) que escreveu sobre a invasão persa da Grécia. [10 p.56].

Enéas não se preocupava apenas com o envio de mensagem, mas também com a detecção da esteganografia, uma forma de impedir que estrangeiros em sua cidade recebessem mensagens inimigas era fazendo com que as correspondências passassem por sensores. Nos dias atuais, essa prática de examinar correspondências, ainda é utilizada em presídios e em agências de inteligência como a NASA [8].

Embora as técnicas apresentadas por Enéas, hoje, pareçam simples, em sua época elas se mostravam eficientes.

2.2 A ESTEGANOGRÁFIA DE TRITHEMIUS

Nascido em 1462 na Alemanha, Johannes Trithemius foi o primeiro autor a publicar uma obra sobre a arte da escrita oculta, sendo que do título de um dos seus livros, *Steganographia*, originou-se a palavra esteganografia [3].

Steganographia é uma obra escrita em três volumes, disfarçado de um livro de magia negra, mas que tratava tanto de esteganografia quanto de criptografia, detalhando técnicas para envio de mensagem de forma oculta. Por esses escritos, Trithemius foi julgado por praticar magia, sob alegação de que utilizava os espíritos para envio de mensagens secretas. Sua obra foi proibida, sendo publicada somente em 1608, noventa e dois anos após sua morte.

2.2.1 AVE MARIA DE TRITHEMIUS

Uma técnica desenvolvida por Trithemius, de fácil demonstração e entendimento, consiste em uma tabela com letras que referenciam frases religiosas. Com essa tabela é possível codificar e decodificar as mensagens apenas consultando-a e identificando a letras correspondentes. O resultado obtido se confunde com um poema religioso, uma oração, onde a esteganografia se apresenta de forma bem funcional, desde que a mensagem oculta seja curta.

A tabela original é complexa e composta por quarenta tabelas escritas em quatro idiomas: latim, alemão, italiano e francês [13]; porém, para facilitar a compreensão o exemplo abaixo utiliza uma tabela; veja Tabela 1; mais simplificada e escrita em português.

Tabela 1: Codificação para Ave Maria de Trithemius.

letra	frase correspondente
A	No céu
B	Para todo sempre
C	Um mundo sem fim
D	Numa infinidade
E	Perpetuamente
F	Por toda a eternidade
G	Durável
H	Incessantemente
I / J	Irrevogável
K	Eternamente
L	Na sua glória
M	Na sua luz
N	No paraíso
O	Hoje
P	Na sua divindade
Q	Em Deus
R	Na sua felicidade
S	No seu reino
T	Na sua majestade
U / V / W	Na sua beatitude
X	Na sua magnificência
Y	Ao trono
Z	Em toda eternidade

Um poema codificado poderia ser:

Perpetuamente no seu reino
Na sua majestade
Perpetuamente durável
Hoje no paraíso
Hoje durável
Na sua felicidade
No céu por toda a eternidade
Irrevogável no céu

Perpetuamente na sua luz
Na sua luz irrevogável
Numa infinidade irrevogável
No céu no seu reino

Numa infinidade irrevogável
Durável e irrevogável
Na sua majestade no céu
Irrevogável no seu reino

O resultado da decodificação da oração seria a frase: ESTEGANOGRAFIA EM MIDIAS DIGITAIS. Uma observação importante é que normalmente a codificação de letras é uma técnica criptográfica, porém, neste exemplo o resultado obtido é um texto legível, o que torna esta técnica também em uma forma de esteganografia.

2.3 ESTEGANOGRÁFIA – PASSADO, PRESENTE E FUTURO

2.3.1 PASSADO DA ESTEGANOGRÁFIA

Além dos casos já citados de Enéas, o Tático; e Johannes Tristhemius, o passado nos mostra vários casos de técnicas esteganográficas sendo utilizadas em diversos lugares e épocas da humanidade.

Na China antiga, normalmente, as mensagens transmitidas aos soldados e aos aliados diplomáticos eram enviadas fazendo-se uma pessoa memorizar a informação. Porém, em algumas ocasiões, as mensagens eram escritas em um pedaço de seda e enroladas em cera, criando pequenas bolas que eram engolidas pelos enviados [14 p.21].

Giovanni Porta, um cientista italiano nascido em 1535, deu uma contribuição para esteganografia ao descobrir que uma mistura formada por um onça de alumínio e cerca de meio litro de vinagre formava uma tinta que penetrava na casca do ovo cozido e depositava-se sobre a superfície branca do interior. Desta forma, era possível escrever sobre a casca que a mensagem só poderia ser lida após o ovo ser descascado [15 p.20].

Na Primeira Guerra Mundial, foi usada, com sucesso, uma técnica chamada de *Turning Grille* e aperfeiçoada com o nome de *Cardone Grille*, que consiste em um cartão com posições perfuradas de forma aleatória. Coloca-se o cartão sobre uma folha e escreve-se a mensagem na posição dos furos. Depois de preenchido os furos, o cartão é girado em noventa graus dando continuação à escrita, assim sucessivamente. Após a mensagem ser preenchida, escreve-se um texto inócuo com as letras desejadas já posicionadas. Para decodificar, basta o receptor utilizar um cartão com a mesma perfuração do utilizado na redação e posicioná-lo sobre o texto recebido [15 p.31].

Durante a Segunda Guerra Mundial, uma técnica chamada de microponto foi largamente utilizada. O processo de criação do microponto utilizava um equipa-

mento capaz de reduzir uma fotografia ao tamanho de aproximadamente um milímetro, deixando-a do tamanho de um ponto. Assim, poderia ser utilizado qualquer meio de transmissão como, por exemplo, um selo; bastando ser ampliada para ser compreendida [15 p.8].

O domínio de técnicas esteganográficas por grupos terroristas levou o FBI, após o Atentado de 11 de setembro, a gastar milhões de dólares em uma tentativa frustrada de varredura de todos os sites da internet. Após nove meses de trabalho, menos de 1% dos sites foram verificados, mostrando que com a tecnologia da época e mesmo com a atual, esse tipo de busca é inviável [16 p.28].

2.3.2 PRESENTE DA ESTEGANOGRÁFIA

Algumas técnicas de esteganografia do passado – como exemplo, as tintas invisíveis – continuam muito presentes na atualidade. Hoje, as tintas visíveis sob luz ultravioleta fazem parte da segurança de cédulas de diversos países, como Brasil e Suíça. Essa tecnologia também proporciona a identificação de cheques de viagens cancelados, fazendo com que a palavra ‘VOID’ apareça sob exposição à luz UV.

O mundo digital abre uma grande possibilidade para utilização da esteganografia. A precisão do computador e a limitação dos nossos sentidos permitem a inclusão de informação em imagens, áudios e vídeos de forma que não seja possível à identificação das informações pelos seres humanos.

Com relação às imagens, é possível incluir informação nos *bits* menos significativo das cores que estas passarão despercebidas aos olhos nus. Outra possibilidade é a inclusão da informação em áreas dos arquivos que não fazem parte da exibição da imagem, regiões que muitas vezes são utilizadas para controle, encontram-se reservadas para possíveis utilizações futuras, ou simplesmente são áreas descartadas na construção da imagem.

Um dos problemas da inclusão de informação no *bit* menos significativo das cores de uma imagem é que de cada oito *bits*, apenas um poderá conter a informação esteganografada. Sendo assim uma imagem poderá, aproximadamente,

conter no máximo 12,5% de seu tamanho em dado oculto. Uma alternativa a essa limitação seria utilizar a técnica BPCS.

O BPCS é um tipo de esteganografia, utilizada normalmente no formato de imagem BMP⁸ (*true color – 24 bits*⁹), que consegue armazenar uma grande quantidade de informação se baseando na dificuldade do sistema ótico humano em identificar alteração de cor na zona de conflito de um arquivo BMP. Essa zona de conflito acontece quando ao isolarmos em cada ponto da imagem o mesmo *bit* da cor a informação da imagem se perde por completo, se tornando uma grande quantidade de pontos coloridos de forma aleatória. Iniciando o isolamento do *bit* mais significativo para o menos significativo, poderá ser feita a inclusão da informação, sempre que o isolamento apresentar essa região de conflito. Essa técnica será detalhada no capítulo 4.

A inclusão de informação no *bit* menos significativo pode ser utilizada também para ocultar mensagens em arquivos de áudio e vídeo. Assim com na imagem, existem outras possibilidades de inclusão de informação nesses formatos de mídia, sendo que os vídeos, por se tratarem de uma combinação de imagens e áudios, possibilitam a utilização das técnicas de ambos os formatos, assim como suas combinações.

Os vídeos contêm uma peculiaridade, a percepção de movimento em sua visualização se deve a exibição em alta velocidade de *frames*¹⁰, de forma que o cérebro não consiga interpretá-los separadamente. Essa limitação do cérebro humano permite que mensagens sejam incluídas entre os *frames* sem ser percebidas pelos espectadores, sendo necessária uma desaceleração na exibição das imagens para que a informação possa ser vista [20].

A linguagem de marcação HTML, utilizada como padrão para exibição de páginas na *web*, possibilita a ocultação de informação de diversas formas. Uma possibilidade simples é a utilizar a mesma cor para o texto e para o fundo da página.

⁸ BMP – formato de imagem digital.

⁹ *True Color 24 bits* – informação de uma cor digital onde a cor é formada por três cores primárias (RGB): vermelho, verde e azul; onde cada cor primária é formada por 8 *bits*. Desta forma existem 256 tons de cada cor primária que combinadas possibilitam um total de 2^{24} variações de cores.

¹⁰ Frames – quadros ou uma imagem fixa.

Essa técnica, porém, é pouco eficiente, pois qualquer buscador identificaria a existência da mensagem oculta.

Outro método um pouco mais eficaz consiste na utilização da *tags* de comentários. No HTML o conteúdo delimitado pelas *tags* “`<!--`” e “`-->`” não são exibidos na página, desta forma a recuperação da informação é possível apenas exibindo o código fonte da página.

Pode-se citar ainda como forma de esteganografia em HTML a possibilidade de inclusão de informação em javascript¹¹, CSS¹² em atributos dos elementos de uma página, ou seja, o HTML disponibiliza um ambiente fértil para utilização de técnicas de ocultação de informação. Essa predisposição a esteganografia aliada à sua finalidade de disponibilizar informação pela internet torna as páginas web alvo de grande preocupação das agências de segurança, vide tópico 2.3.1§6.

Uma estratégia de esteganografia moderna denominada *winnowing and chaffing*, que se traduz como separar o joio do trigo, foi sugerida por Ron Rivest e é interessante para utilização em rede comutada¹³. O mecanismo consiste em enviar várias mensagens falsas, chamadas de *winnowing*, e apenas uma é verdadeira, chamada de *chaffing*. Assim ninguém além do destinatário, que terá a chave para a mensagem verdadeira, poderá identificar entre as falsas qual realmente interessa [18 p.558]. O nome *winnowing and chaffing* remete a uma técnica agrícola que permite separar o que se deseja do que não se deseja.

Uma nova técnica de envio de informação de forma transparente, utilizada atualmente, é injetar atrasos imperceptíveis em pacotes que são enviados pela rede. Este último método, que será relatado neste tópico, tem uma característica peculiar, pois permite o uso da esteganografia sem que seja aumentado o processamento ou tráfego na rede onde o processo está sendo utilizado, bastando ao receptor a sensi-

¹¹ Javascript – Linguagem de programação interpretada utilizada para manipulação de páginas HTML.

¹² CSS – “Folha de estilo em cascata”. Especificação que define o estilo com que os elementos serão exibidos em uma página HTML. Exemplo: cor, tamanho, fonte, bordas, posicionamento etc.

¹³ Rede comutada - “Uma rede comutada é formada por uma série de nós interligados, denominados comutadores. Tradicionalmente, existem três métodos importantes de comutação: comutação de circuitos, comutação de pacotes e comutação de mensagens.” [19 p.236]. Internet é um exemplo de rede comutada.

bilidade na percepção dos atrasos para reconhecimento da informação transmitida [20].

2.3.3 FUTURO DA ESTEGANOGRÁFIA

A esteganografia parece ter um lugar garantido no futuro. Várias técnicas e produtos estão sendo desenvolvidos com tecnologias esteganográficas, já que muitas vezes se faz necessário obter informações sobre uma determinada coisa sem que os dados informativos causem interferência no objetivo principal. A simples inclusão dos dados de forma visível pode não só afetar o *design* como ser inviável por sobrepor o verdadeiro conteúdo desejado ou, até mesmo, impossível de se inserido com a utilização de técnicas convencionais.

O código de barras é um problema para o *design* de embalagens, porém, o problema se agrava com a diminuição das medidas das embalagens, sendo muitas vezes necessária a utilização de embalagens significativamente maiores do que os produtos, para que essas possam comportar a informação. Para resolver esse problema, algumas empresas como a CIS e a Fujitsu vêm trabalhando na criação de código de barras invisível. Desta forma, seria possível a inclusão de uma quantidade grande de informação em uma embalagem, sem que esta tivesse influência na apresentação da mesma. Atualmente, vários canais de comunicação relatam o desenvolvimento de um código de barra invisível pela empresa japonesa Fujitsu e a CIS anuncia em seu site [21] um sensor capaz de ler, por infravermelho, códigos de barras invisíveis.

A indústria médica é outro ramo que pode se beneficiar com o uso da esteganografia.

Com o desenvolvimento de equipamentos que possibilitavam o diagnóstico por imagem, os profissionais da área médica ganharam uma importante ferramenta para avaliação dos quadros clínicos de seus pacientes. Como consequência, o armazenamento, transporte, recuperação de imagem e suas informações passam a fazer parte do cotidiano de uma unidade médica. Os diversos tipos de exames e aparelhos de variados fabricantes; associados à crescente utilização dessas tecno-

logias criaram a necessidade de padronização na troca de informação de imagem digital entre os aparelhos e computadores. Desta forma foi criado o padrão DICOM.

Apesar da padronização, as imagens e os dados dos pacientes continuam a ser manipulados de forma independentes, possibilitando que a relação entre ambos seja perdida. A esteganografia se encaixaria perfeitamente na resolução desse problema. Os dados do paciente poderiam ser inseridos de forma oculta nas imagens dos exames, desta forma, além de não correr o risco de ter imagem e informação separadas indevidamente, seria possível obter todo o histórico médico de um paciente dentro da própria imagem do exame.

A mesma técnica utilizada nas imagens médicas poderia ser utilizada para outros fins. Seria possível, por exemplo, armazenar dados geográficos locais em fotos de satélite. As câmeras digitais que incluem nas fotos informações sobre data e hora da captação da imagem poderiam adicionar essas informações de forma invisível, ampliando assim sua capacidade registro de dados além das fotos, pois esses dados não mais atrapalhariam a visualização da imagem.

2.3.3.1 *Genomic Steganography*

Na Segunda Guerra Mundial foi utilizada como forma de esteganografia a técnica de micropontos. Uma moderna versão dessa concepção foi recentemente proposta para esconder informações em DNA [25].

Em 1999, pesquisadores da *Mount Sinai of Medicine* em Nova York publicaram na revista *Nature* um artigo, intitulado “*Hiding messages in DNA microdots*” de tradução “escondendo mensagens em micropontos de DNA”, onde codificavam uma mensagem escondida em uma fita de DNA humano utilizando uma técnica descrita como *genomic steganography* [23].

A codificação proposta trabalha com a alteração das sequências de nucleotídeos - A, T, C e G - da estrutura do DNA permitindo que essa seja interpretada pelo conhecedor do algoritmo utilizado e com a utilização de equipamentos especiais. Essa técnica demonstra-se barata e confiável, podendo ser utilizada não apenas

no DNA, mas também nas bases de dados que descrevem a sequência do mesmo ou nas sequências estruturais e de dados de proteínas [22 p.76].

A Figura 1 apresenta um exemplo de utilização desta técnica esteganográfica, onde a frase “JUNE 6 INVASION: NORMANDY” está oculta no trecho com a sequência de nucleotídeos “AGT CTG TCT GGC TTA ATA ATG TCT CCT CGA ACG ATG GGA TCT GCT TGA TCA TCC CGA TCT TTG AAA”.

O problema do método de microponto em DNA é que a modificação de sua estrutura poderia causar grandes modificações no organismo ao qual o DNA pertence, porém, publicações científicas recentes projetam que grandes extensões de sequências não estão intimamente relacionadas com a função do gene. Sendo assim, a capacidade da base de dados do genoma humano para armazenar dados esteganográficos sem exceder a taxa normal de defeitos é muito grande.

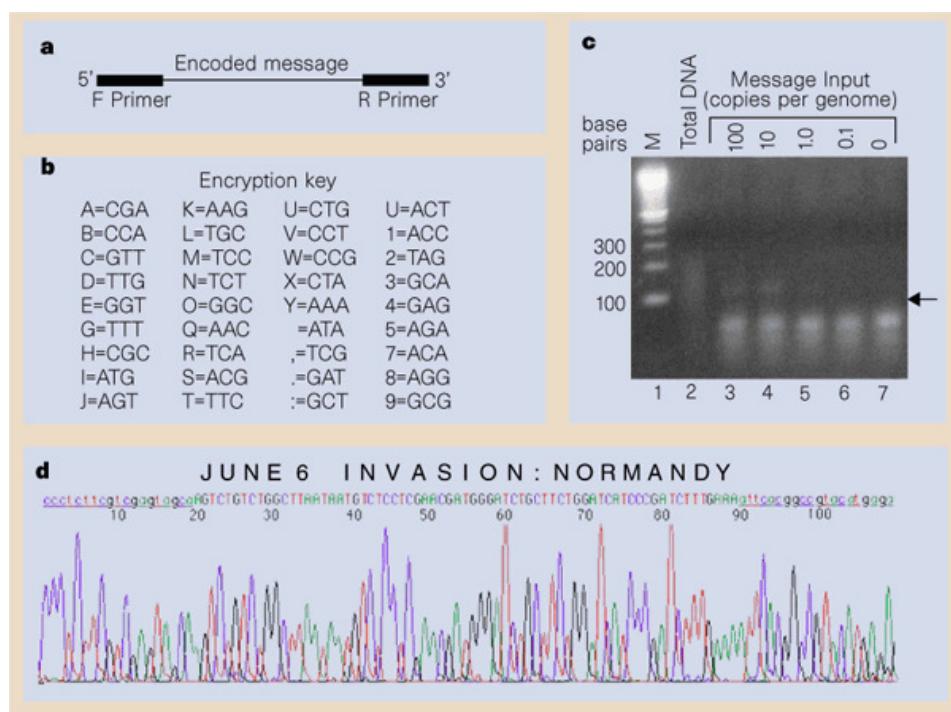


Figura 1: Exemplo de Esteganografia Genômica [24].

A utilização de DNA no armazenamento e transporte de informação tem aparentado um ramo promissor no futuro, tanto que fez a IBM¹⁴ desenvolver uma linguagem específica para armazenar a informação em dados de sequência de DNA.

¹⁴ IBM – Gigante multinacional norte americano do ramo de informática.

Para efeitos comparativos, um disco rígido de um PC, que trabalha com base binária¹⁵, consegue armazenar uma densidade de dados de aproximadamente 7 *gigabits* por polegada quadrada. Já o DNA, que armazena informação na base quatro¹⁶, consegue uma densidade de armazenamento, em potencial, de mais de 1 milhão de *gigabits* por polegada quadrada [22].

Prognosticando a utilização desta nova técnica, seria possível a assinatura de modificações genéticas, por seus criadores. Poderiam ser assinadas vacinas, células-tronco entre muitos outros produtos desenvolvidos em laboratórios. Produtos agropecuários poderiam carregar, em si próprio, informações relevantes sobre suas origens e qualquer outro dado que se julgar relevante. Entretanto, talvez a grande mudança de paradigma seria o fato de que qualquer organismo vivo se tornaria um potencial banco de dados com capacidade de armazenamento, até então, nunca visto.

¹⁵ Base binária – ou base 2. Apresenta dois estados de informação. É comum ser representada na informática pelos dígitos 0 e 1.

¹⁶ Base quatro – Apresenta quatro estados para representação da informação. No DNA representados pelas letras A, C, G e T.

3 APLICAÇÕES

A partir de meados dos anos 1990 a aplicabilidade da esteganografia começou a ser explorada para fins comerciais, aumentando o interesse pela sua utilização e impulsionando publicações acadêmicas sobre o tema. O Gráfico 1 ilustra o número de publicações anuais realizada pela IEEE¹⁷ sobre: marca d'água e esteganografia; desde o ano de 1990 até o ano de 2006.

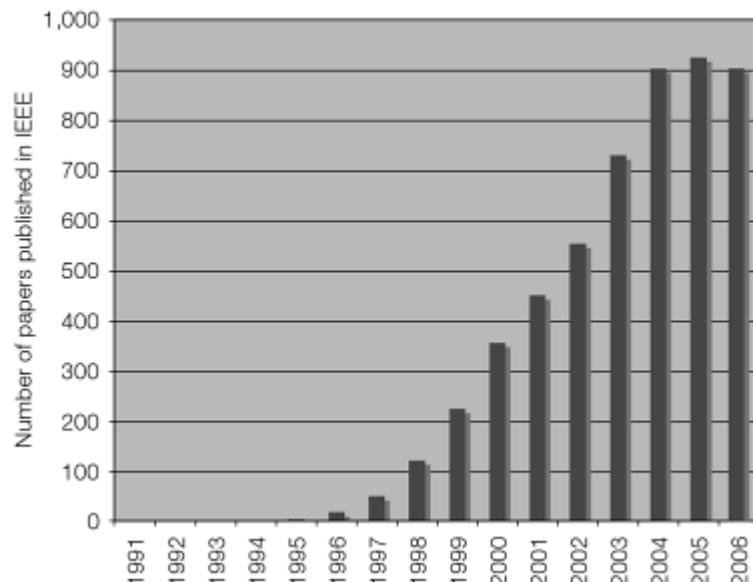


Gráfico 1: Número anual de publicações sobre marca d'água e esteganografia realizadas pela IEEE [26, p.9].

¹⁷ IEEE – “É a maior associação profissional do mundo dedicada ao avanço da inovação tecnológica e excelência para o benefício da humanidade. IEEE e seus membros estimulam uma comunidade global através de suas publicações altamente citadas, conferências, padrões tecnológicos, e atividades profissionais e educacionais” [27].

3.1 SEGURANÇA MONETÁRIA

Nas cédulas de dinheiro, por exemplo, são utilizadas diversas marcas ocultas como forma de segurança, impedindo que possam ser feitas cópias idênticas, possibilitando, assim, a identificação de falsificações [17].

As Figuras 2 e 3 ilustram o anverso e o reverso de uma cédula de R\$ 10,00 e as imagens demonstram algumas técnicas esteganográficas utilizadas para sua segurança.



Figura 2: Anverso de uma cédula de R\$ 10,00.



Figura 3: Reverso de uma cédula de R\$ 10,00.

Marca D'água (Figura 4) – Apresenta a imagem da Bandeira Nacional brasileira e só pode ser vista se visualizada contra a luz.



Figura 4: Marca d'água na cédula de R\$ 10,00 estampa C.

Microimpressões – Utilizando uma lente de aumento é possível ver a impressão de pequenas letras “B” e “C”, Figura 5, na faixa clara junto à efígie (anverso) e no interior do número 10 (anverso e reverso).

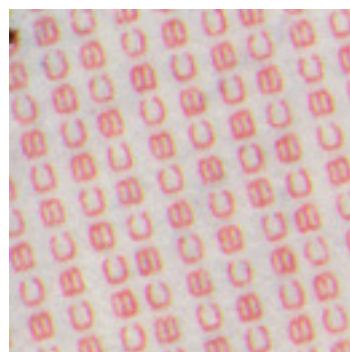


Figura 5: Microimpressões das letras "B" e "C" na cédula de R\$ 10,00.

Imagen latente – Observando a frente da cédula, posicionada a altura do olho, na posição horizontal e sob luz natural, é possível observar as letras “B” e “C” no canto inferior esquerdo, Figura 6.



Figura 6: Imagem latente das letras "B" e "C" na cédula de R\$ 10,00.

Fibras luminescentes – Pequenos fios espalhados pela cédula que podem ser vistos na cor lilás, Figura 7, quando exposta à luz ultravioleta.



Figura 7: Fibras luminosas na cédula de R\$ 10,00.

3.2 IMPRESSORAS MODERNAS

Em 2004, a *PCWorld* publicou um artigo – intitulado “Governo utiliza tecnologia de impressoras laser coloridas para rastrear documentos” ou “*Government Uses Color Laser Printer Technology to Track Documents*”, em seu título original [28] - que alertava a população sobre o fato de que algumas impressoras estavam imprimindo pontos ocultos, em cada página impressa, para controle. A existência dos pontos de controle na impressão era verdadeira, no entanto essa medida foi apresentada ao governo, na década de 1990, pela Xerox, em princípio, com intuito de mostrar ao governo que suas impressoras não seriam utilizadas para cópias ilegais.

As informações ocultas trazem dados relativos à impressão, como número serial da impressora, data e hora de impressão. Apesar de pouca informação, é possível identificar que a utilização desta forma de impressão vem se tornando um grande aliado dos técnicos forenses. Por outro lado, são alvos de críticas por grupos que consideram esse método invasivo.

Impressoras laser de fabricantes como HP, Canon e Xerox utilizam esta técnica esteganográfica que consistem em incluir pequenos pontos amarelos de aproximadamente 0,01 mm de tamanho e separados uns dos outros por uma distância de aproximadamente 1 mm. Ao ler a posição de cada ponto é possível recuperar a informação esteganografada.

A Figura 8 ilustra um exemplo de codificação utilizada pela impressora DocuColor da Xerox. Nela é possível visualizar os pontos amarelos, que em situação normal são invisíveis ao olho humano. A imagem também indica a posição onde se encontra cada informação e o valor de cada posição do ponto. A soma dos valores dos pontos de uma coluna indica o valor da informação referenciada pela coluna.

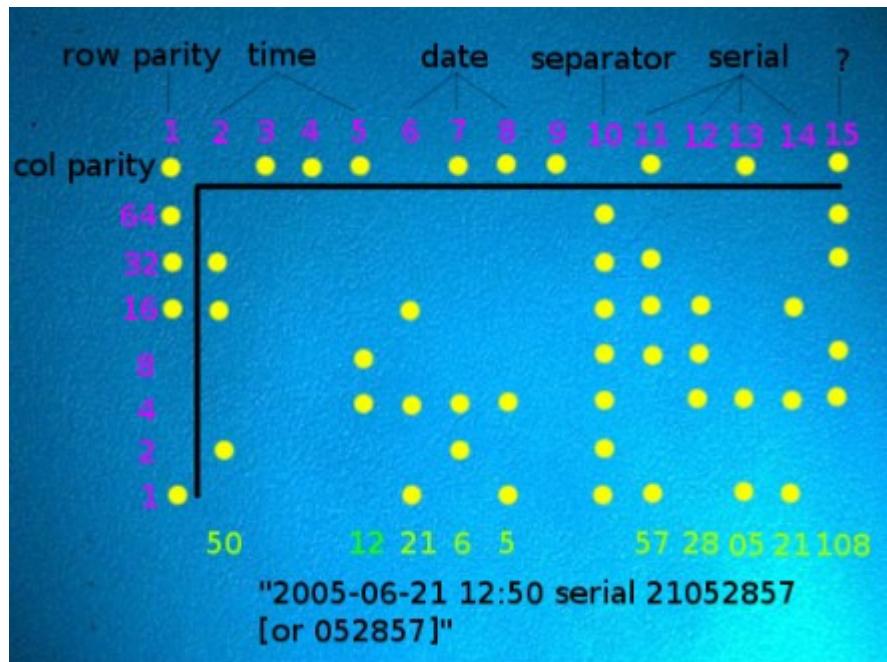


Figura 8: Visualização dos pontos ocultos de uma impressão [28, p.15].

Devido ao tamanho reduzido é necessária uma grande qualidade e precisão nas impressões, por esse motivo essa técnica esteganográfica é utilizada apenas por impressoras modernas.

3.3 MONITORAMENTO DE TRANSMISSÃO

Em 1997 um escândalo atingiu a televisão japonesa. Duas estações televisivas praticavam *overbooking*¹⁸ em seus tempos de transmissão. Desta forma, mi-

¹⁸ Overbooking – Termo em inglês que significa excesso de reservas. Atualmente, podemos considerar como venda de um produto em quantidade maior do que pode ser entregue.

Ihares de comerciais pagos nunca foram ao ar. Essa prática manteve-se por mais 20 anos sem ser detectada, em parte por não haver um sistema de monitoramento de transmissão. A partir desse episódio diversos tipos de organizações e interessados passaram a monitorar as transmissões com intuito de erradicar essa prática [26].

Os japoneses não foram os únicos a enfrentarem problemas de pagamentos inadequados relacionados à televisão. Em 1999 uma verificação realizada pela SAG¹⁹ concluiu que em média \$ 1.000,00 por hora eram pagos indevidamente de *royalties* de programação nos Estados Unidos [26].

Os proprietários de direitos autorais que desejam garantir que suas obras não sejam retransmitidas por estações piratas também fazem parte dos interessados no monitoramento das transmissões.

O primeiro monitoramento era simples, consistia em pessoas assistindo as transmissões e gravações, fossem elas televisivas ou de rádios, para identificação das irregularidades. Essa técnica mostrou-se pouco eficiente e logo foi substituída por métodos que utilizavam computadores [26].

Os computadores passaram a simular a inspeção realizada pelos humanos, comparando frames das transmissões com as informações guardadas em seus bancos de dados, para fazer o reconhecimento. Um dos problemas dessa técnica é que o volume de dados de vídeo é grande; seu armazenamento, busca e comparação necessitam de uma computação complexa acarretando um grande potencial de problemas [26].

Uma solução esteganográfica seria embutir marcas digitais informativas nas transmissões. A parte do sinal contendo dados, chamada de VBI, é enviada entre os *frames* sem que haja interferência na exibição da imagem. O *close caption* é propagado utilizando este método. Desta forma, as transmissões poderiam ser fiscalizadas, bastando analisar o VBI. O problema desta técnica é que em alguns países não existe uma legislação consistente para este tema, a legislação norte americana, por exemplo, não define claramente quem tem o controle desta faixa de sinal, não sendo obrigatório à controladora do *close caption* entregar qualquer outro conteúdo presente no VBI [26].

¹⁹ SAG – Sindicato de atores, nos Estados Unidos.

Neste contexto, outra forma de esteganografia surge como uma solução eficaz, tanto para vistoriar sinais de rádio quanto para vistoriar sinais de televisão; seja ele analógica ou digital. Informações necessárias para o monitoramento da transmissão podem ser inseridas no próprio conteúdo do vídeo ou do áudio, técnica conhecida como marca d'água digital. Desta forma, seria possível identificar se uma música que contenha a marca d'água é compatível com a estação de rádio que a está transmitindo, ou quantas vezes um canal de televisão exibiu uma propaganda contendo determinada marcação [26].

Esta última forma esteganográfica apresenta uma implementação um pouco mais complexa que a recuperação de informação no VBI, porém não necessita de manipulação de grande quantidade de dados de vídeos gravados em banco de dados. Apenas a manipulações do sinal seria suficiente para extração do conteúdo necessário ao monitoramento.

3.4 PROPRIEDADE INTELECTUAL

A utilização da esteganografia para auxiliar a proteção de direitos autorais é uma das aplicações mais citadas em publicações e trabalhos relacionados à ocultação de informação, sendo um assunto extenso e complexo.

Nos Estados Unidos, por exemplo, para que um autor tenha seus direitos garantidos, é necessário que nas obras protegidas por propriedade intelectual seja incluído um texto informativo sobre a propriedade. Porém acontece - e muitas vezes sem má fé - que uma cópia - mesmo que obtida legalmente - acaba danificando ou excluindo a parte da imagem onde se encontra a informação de propriedade, fazendo com que uma pessoa que tenha acesso à cópia não identifique a existência de um proprietário, ou o detentor da propriedade.

Um famoso caso de perda da informação de propriedade foi a foto de Lena Sjööblom – ilustrada na Figura 9 – que se tornou, talvez, a foto mais utilizada no estudo de processamento de imagem. A foto já apareceu em diversos artigos, jornais e conferências sem que a empresa *Playboy*, sua verdadeira proprietária tenha sido referenciada. Tudo começou quando em uma central da *Playboy*, para teste, a

foto foi digitalizada tendo a maior parte descartada, perdendo a referência de propriedade. A imagem foi espalhada mundialmente e muitos pesquisadores a incluíram em suas publicações. A *Playboy*, no entanto, decidiu ignorar a utilização generalizada da imagem [26, p.20].



Figura 9: Parte da foto de Lena contendo a referência à propriedade da *Playboy* [26, p.21].

Para identificar o direito autoral de uma mídia digital, poderia ser incluída, de forma esteganográfica, a informação com referência ao proprietário. Essa inclusão poderia ser realizada no cabeçalho do arquivo digital, porém as ferramentas de recortar, colar, incluir entre outras; possibilitam que a informação de cabeçalho seja perdida. Para resolver esses problemas, poderiam ser espalhadas, por todo o arquivo digital, várias repetições referentes ao proprietário, de forma que mesmo que o arquivo fosse recortado ou modificado, até um determinado nível, seria possível identificar no fragmento restante a informação de autoria.

3.5 PROVA DE PROPRIEDADE

A esteganografia é um grande aliado à proteção da propriedade intelectual por permitir a inclusão de informações dentro do arquivo, porém, estas técnicas servem também para provar o autor de uma determinada obra.

A forma mais adequada para comprovar os direitos sobre uma criação seria registrá-la nos órgãos responsáveis, porém, em alguns casos, como disponibilização de várias imagens na internet, o custo de registro pode ser tornar inviável, fazendo com que muitos autores dispensem as formas de registros convencionais.

Nestes casos, reunir provas que comprovem o verdadeiro autor de uma criação pode não ser tarefa tão simples. Uma alternativa é incluir informações invisíveis, de forma que apenas o próprio autor seja capaz de recuperar, isso facilita o processo de reconhecimento judicial, podendo ser decisivo para o convencimento dos tribunais.

3.6 AUTENTICAÇÃO

A adulteração de um arquivo digital pode ser de difícil identificação. Facilmente encontramos montagens de imagens sendo espalhadas pela rede. Revistas masculinas utilizam largamente editores de imagens para corrigir imperfeições em suas imagens publicadas. Apesar de terem sido mencionadas apenas imagens digitais, a alteração pode ocorrer também, em áudios e vídeos.

Os exemplos citados normalmente não causam grandes transtornos, porém se a mídia alterada for uma prova crítica de uma investigação policial poderá existir um grande problema.

Uma técnica apropriada para a questão seria a inclusão de marcas oculistas espalhadas pelo arquivo, em uma espécie de assinatura digital, de forma que qualquer alteração na mídia resultasse na alteração da assinatura. Para tornar essa assinatura mais eficiente, ela poderia ser protegida por uma criptografia. Desta for-

ma, a não ser que se conhecesse a chave, a assinatura não poderia ser reconstruída.

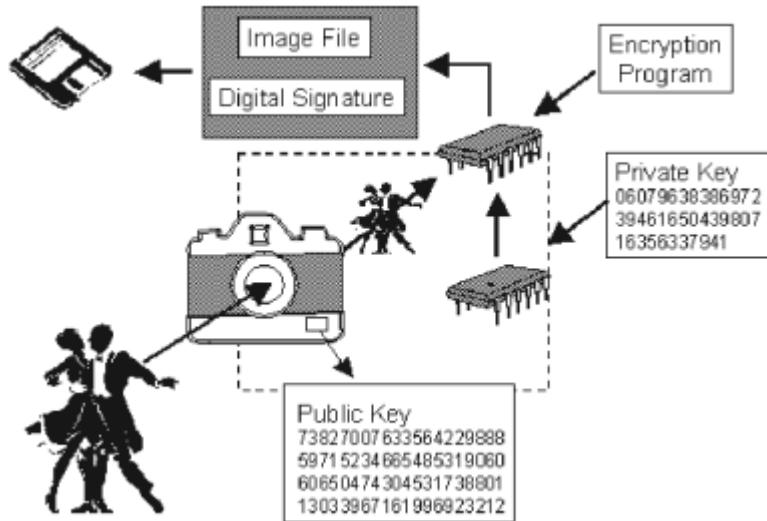


Figura 10: Ideia da câmera digital segura [29, p.1198].

Em 1993 um pesquisador da *University of North Carolina* - chamado William H. Fridman - associou a assinatura digital à câmera digital, propondo uma “*trustworthy digital camera*” ou “câmera digital segura”, em português. A proposta, ilustrada na Figura 10, utiliza um microprocessador programável que gera dois arquivos: uma imagem criptografada e uma chave necessária para decodificação da imagem gerada [29, p.1199].

Uma modificação no projeto de Fridman poderia realizar a inclusão de uma assinatura digital dentro do arquivo, de forma esteganográfica, autenticando-o. Desta forma, uma câmera de segurança, como exemplo, poderia gerar um vídeo que impossibilitaria sua alteração sem que esta fosse identificada.

3.7 CONTROLE DE CÓPIAS

A troca de arquivos digitais realizada por usuários da *web* traz constantes discussões acaloradas sobre a legalidade da prática que atualmente é popular, sobre tudo através dos mais jovens [30]. Não só as trocas de arquivos causam proble-

mas de legalidade, cópias de mídias físicas, como CD's e DVD's estão sujeitas a mesma violação de direitos. Como centro do problema, normalmente encontram-se os arquivos de música em especial o formato mp3²⁰.

Na década de 1990, com a consolidação as redes P2P²¹, vários sistemas para troca de arquivos surgiram tendo um grande impacto na indústria fonográfica [30]. A música adquirida por meio de arquivos na Internet se tornou um grande objeto de consumo na rede, tendo afetado sensivelmente as gravadoras musicais, que pareciam não estar preparadas para tamanha mudança de paradigma.

Pesquisa realizada pela Ipsos-Reid²² publicou que 37% dos norte-americanos entrevistados informaram ter feito cópias “ilegais” de CD's; 40%, entre 12 e 24 anos, já baixaram músicas da web; e 60% dos entrevistados que baixaram músicas classificam o interesse de tornar a baixar como “grande” ou “muito grande” [31].

Uma tentativa de inibir a cópia ilegal de um arquivo seria criar a possibilidade de identificar o proprietário do arquivo original. Para isso, os sites licenciados para comercialização das mídias poderiam utilizar dados dos clientes, que são exigidos ao se cadastrarem – como CPF e ID -, para identificar o arquivo que será disponibilizado. Esses dados poderiam ser inseridos nos arquivos de forma esteganográfica no momento do download. Inserindo também informações referentes ao revendedor, seria possível, ao analisar uma cópia, conhecer não apenas o responsável pelo arquivo original como, também, seu fornecedor.

A solução apresentada criaria um controle sobre cópias de originais obtidas sob forma de arquivos digitais. Porém outro problema é a cópia de mídias físicas como CD's e DVD's. A mesma técnica de inclusão de dados esteganográficos poderia ser utilizada, porém neste caso, necessitaria que o dispositivo utilizado proibisse a gravação no meio físico quando as informações ocultas fossem identificadas. Tal sistema tem sido imaginado para uso em DVD's pela *Technical Working Copy Protection Grupo* [26, p.31].

²⁰ MP3 – Formato compacto de arquivo digital de áudio.

²¹ P2P – Ou *peer-to-peer*. arquitetura de redes de computadores onde cada um computadores pertencentes a da rede funcionam tanto como cliente quanto como servidor.

²² Ipsos Reid – Companhia canadense destinada principalmente a pesquisa de *marketing*.

Para pôr essa tecnologia de limitação dos dispositivos em prática existe um problema não técnico que deve ser solucionado: uma dificuldade em convencer os fabricantes dos dispositivos a adotarem tal restrição, pois incluir detectores em seus dispositivos cria uma desvalorização do produto, já que o consumidor prefere adquirir um equipamento que permite a prática de cópias ilegais.

3.8 COMUNICAÇÕES SECRETAS

Um caso interessante de comunicação secreta foi descrito por Gustavus J. Simmons no início de 1980. Os Estados Unidos e a antiga União Soviética, de acordo com o tratado de SALT²³, concordaram em utilizar sensores em suas instalações nucleares para informação do número de mísseis. Para evitar acesso não autorizado a essa informação, foi utilizada assinatura digital na transmissão de seus sinal [26, p.34].

Existem várias razões para duas partes desejarem se comunicar secretamente. Sejam elas benignas, como dois amantes que desejam manter seu relacionamento em segredo; sejam elas políticas, como a comunicação de organizações dissidentes entre si ou com outras organizações internacionais; ou, sejam elas criminosas, como comunicações entre o crime organizado ou organizações terroristas; o fato é que a esteganografia estará entre os principais métodos utilizados para esta finalidade [26, p.34].

Analizando as organizações dissidentes, há diversos países no mundo onde a dissidência política não é tolerada, podendo até mesmo ser ilegal. Desta forma a comunicação entre os dissidentes e entre dissidentes e organizações internacionais, como Anistia Internacional, deve ser realizada com extrema cautela, pois, é comum vigilância entre as comunicações nesse tipo de regime.

²³ SALT – *Strategic Arms Limitation Talks* ou em português, Conversações sobre Limites para Armas Estratégicas. Tratado entre Estados Unidos e União Soviética para controle de armas nucleares.

Neste cenário normalmente existe três possibilidades para manutenção da privacidade das informações ou dos interlocutores: criptografia, remetente anônimo e esteganografia.

A criptografia pode manter a segurança da informação, mas sua utilização indica a necessidade dos envolvidos em ocultar a mensagem, o que pode acarretar no aprisionamento das partes e até a obtenção da informação com a utilização de torturas.

Utilizar de remetente anônimo impede, em princípio, a identificação dos envolvidos, mas sua prática, assim como a criptografia, indica a necessidade de uma ocultação – neste caso, dos pares –, podendo despertar o interesse dos opositores que montariam investigação para descoberta dos envolvidos.

Sendo assim, nestes casos, a esteganografia pode ser a melhor forma de comunicação entre os dissidentes, pois faria com que a comunicação pudesse ser realizada sem atrair suspeitas.

4 FUNDAMENTAÇÃO TEÓRICA

4.1 IMAGEM DIGITAL

A imagem exibida na tela do computador é formada por um conjunto de unidades discretas chamadas de *pixels*. Cada *pixel* pode assumir apenas uma coloração por vez, e são organizados sob a forma de uma matriz bidimensional [32]. Desta forma, cada unidade pode ser descrita na matriz por sua posição referente à linha e à coluna que ocupa, em uma espécie de plano cartesiano. Na Figura 11, à medida que a imagem é ampliada, é possível ver com mais clareza suas unidades mínimas de cor, *pixel*.

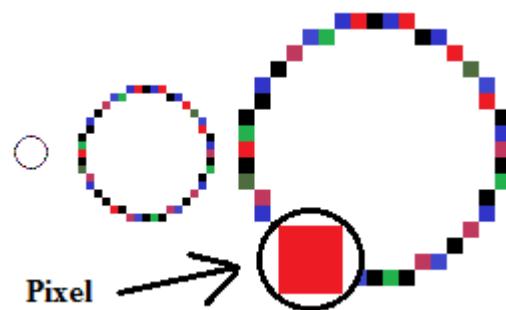


Figura 11: Ampliação do círculo para visualização dos *pixels*.

Quanto maior a densidade de *pixel*, maior é a resolução da imagem. Essa relação entre densidade de *pixel* e qualidade da imagem está representada na Figura 12. O círculo à esquerda é exibido em um ambiente com densidade de *pixel* significativamente maior do que o da direita.

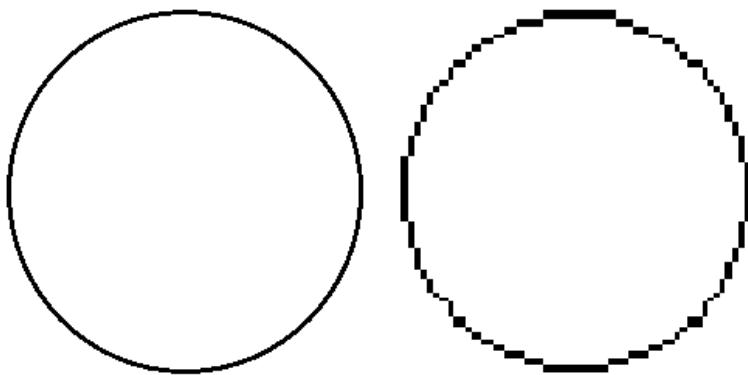


Figura 12: Maior densidade de *pixel* no círculo à esquerda.

4.1.1 FORMATOS DE ARQUIVOS

Uma vez entendida a exibição das imagens digitais, existe a necessidade de armazenar essas informações sob forma de arquivos. Para isso, vários formatos de arquivos – estruturas para armazenamento das informações – foram criados no decorrer dos anos. Alguns desapareceram enquanto outros ganharam larga aceitação.

O BMP é o formato de imagem nativo do *Microsoft Windows*, desde a versão 3. Neste formato cada *pixel* é representado por uma quantidade de *bits* que identificam sua cor. Quanto maior a quantidade de *bits* utilizada no armazenamento, maior será a quantidade de cor passível de representação. O tamanho da informação de cor pode ser de 1, 4, 8, 16, 24 e 32 *bits* por *pixel*, sendo que arquivos que utilizam 16 e 32 *bits* são raros. Este formato suporta RLE, que comprime as cores para 4 e 8 *bits* por *pixel*, no entanto essa compressão é de uso exclusivo para grandes blocos de cores idênticas, o que torna essa compressão limitada. Por esse motivo é raro ser encontrado o formato BMP na forma comprimida.

A representação *pixel* a *pixel*, sem compressão ou qualquer outra forma de tratamento faze com que os arquivos no formato BMP costumem ser grandes se comparados a outros formatos.

O formato geral do arquivo BMP é composto por três ou quatro partes, sendo a primeira o cabeçalho do arquivo com tamanho de 14 *bytes*, que contém in-

formações de assinatura, tamanho e *layout* do arquivo. A Tabela 2 demonstra as disposições e conteúdos dos *bytes* [36].

Tabela 2: Cabeçalho do arquivo no formato BMP.

Campo	Bytes	Descrição
<i>BfType</i>	2	Assinatura do arquivo: os caracteres ASCII "BM" ou (42 4D)h.
<i>BfSize</i>	4	Tamanho do arquivo em <i>bytes</i> .
<i>BfReser</i>	2	Campo reservado. Deve ser zero
<i>BfReser</i>	2	Campo reservado. Deve ser zero
<i>BfOffSetBits</i>	4	Especifica o deslocamento, em <i>bytes</i> , para o início da área de dados da imagem, a partir do início deste cabeçalho. - Se a imagem usa paleta, este campo tem tamanho=14+40+(4 x número de cores). - Se a imagem for <i>true color</i> , este campo tem tamanho=14+40=54.

A segunda parte, que é o cabeçalho do mapa de *bits*, tem tamanho de 40 *bytes* e contém informações sobre imagem contida no arquivo como dimensões, compressão e informações sobre as cores. A Tabela 3 demonstra as disposições e conteúdo dos *bytes* [36].

Tabela 3: Cabeçalho de mapa de *bits* do arquivo BMP.

Campo	Bytes	Descrição
<i>BiSize</i>	4	Tamanho deste cabeçalho. Sempre 40 <i>bytes</i> ou 28h.
<i>BiWidth</i>	4	Largura da imagem em <i>pixels</i> .
<i>BiHeight</i>	4	Altura da imagem em <i>pixels</i> .
<i>BiPlanes</i>	2	Número de planos de imagem. Sempre 1.
<i>BiBitCount</i>	2	Quantidade de <i>bits</i> por <i>pixel</i> : 1,4,8,24 ou 32.
<i>BiCompress</i>	4	Compressão usada. 0 (sem compressão), 1 (compressão RLE 8 <i>bits</i>), 2 (compressão RLE 4 <i>bits</i>).
<i>BiSizeImag</i>	4	Tamanho da imagem em <i>byte</i> .

<i>BiXPPMeter</i>	4	Resolução horizontal em <i>pixels</i> por metro.
<i>BiYPPMeter</i>	4	Resolução vertical em <i>pixels</i> por metro.
<i>BiClrUsed</i>	4	Número de cores usadas na imagem. Quando zero, indica o uso do máximo de cores possível pela quantidade de <i>bits</i> por <i>pixel</i> .
<i>BiClrImpor</i>	4	Número de cores importantes na imagem. É útil quando for exibida um dispositivo que suporte menos cores que a imagem possui.

A terceira parte, que é opcional, contém a paleta ou mapa de cores contém 4 *bytes* e é utilizada apenas para imagens de 16 ou 256 cores. A Tabela 4 demonstra as disposições e conteúdo dos *bytes* [36].

Tabela 4: Paleta cores (opcional) do arquivo BMP.

Campo	Bytes	Descrição
<i>Blue</i>	1	Intensidade de Azul. De 0 a 255.
<i>Green</i>	1	Intensidade de Verde. De 0 a 255.
<i>Red</i>	1	Intensidade de Vermelho. De 0 a 255.
Reservado	1	Campo reservado deve ser sempre zero.

Por último temos a área de dados da imagem, que contém a informação da cor referente a cada *pixel* da imagem. Seu tamanho é variável, dependendo da quantidade de *pixels* representados e da quantidade de *bits* utilizados para armazenamento de sua cor [36].

O JPEG, ou simplesmente JPG, tornou-se o formato mais utilizado para armazenamento de fotografias [32], no entanto, apesar de sua ampla utilização seu funcionamento interno possui uma compressão complexa que necessita de um esforço computacional superior a maioria dos outros formatos utilizados para obtenção da imagem armazenada.

JPEG é um acrônimo para "Joint Photographic Experts Group", organização que criou o padrão para compressão de imagens utilizada por esse tipo arquivo. A norma JPEG é complexa, porque, em vez de definir um formato de arquivo, ele define uma série de relacionamento entre técnicas de compressão de imagem, tor-

nando este padrão adequado para armazenamento de imagens fotográficas, pois permite, nestes casos, maior compressão que qualquer formato de mapa de *bits* em uso. Uma fotografia que ocupa 1 MB para ser armazenada em um arquivo BMP normalmente pode ser comprimido até 50 KB com o JPEG [32].

Apesar de sua grande capacidade de compactação, o JPEG não é adequado para algumas aplicações. Os métodos de compactação JPEG, em geral, causam perdas de informação, o que os tornam inadequados para armazenamento intermediário – quando edição e gravação são realizadas repetidamente –, pois a qualidade da imagem original vai diminuindo à medida que cada armazenamento é realizado. A utilização para desenhos também não é aconselhável, pois a eficiência da compactação não é a mesma que quando utilizada em fotografias.

O *Graphics Interchange Format* é extensamente usado na *Web*, com maior ênfase na arte de linha e não para imagens fotográficas. Conhecido como GIF, este formato acumula 256 cores de uma imagem em uma tabela chamada de paleta. Considerando que imagens têm milhões de cores, um programa para edição deste tipo de arquivo normalmente seleciona as melhores cores representativas do todo para realizar a gravação. Quando exibida, cada *pixel* na imagem é mostrado como uma das cores da paleta utilizada.

O PNG ou *Portable Network Graphics* foi desenvolvido para substituir o envelhecido formato GIF e é apoiado pelo *Microsoft Internet Explorer* e *Netscape Navigator*. O PNG, como o GIF é um formato de *lossless*, mas tem algumas características que o formato de GIF não possui.

RAW, ou cru em inglês, é um formato de arquivo digital que contém o máximo de dados captados pelo sensor da câmera fotográfica. Assim sendo, o RAW é uma sequência de números, resultante da codificação binária dos impulsos elétricos captados pelos sensores – referentes à incidência da luz recebida –, que independem de um tratamento que vá interpretá-los como imagens. Apesar de ter o tamanho como um de seus problemas, pois os arquivos ficam muito grandes, ele é disseminado entre os fotógrafos profissionais podendo ter diversas extensões, dependendo do fabricante da câmera: CR2, NEF ou DNG; e é o único formato imagem aceito nos tribunais de justiça brasileiros como prova, pois não apresentam nenhum processamento de imagem e apenas o possuidor da câmera fotográfica detém as imagens neste estado original [33 p.155].

O formato TIFF (ou TIF), *Tag Image File Format* ou Formato de Arquivo de Imagem Rotulado, foi desenvolvido originalmente pela Aldus *Corporation* para salvar imagens criadas por *scanners*, *frame grabbers* e programas que editam fotografia e é extremamente recomendável para a confecção de impressos profissionais, como revistas, folders, livros e qualquer outro material impresso em gráficas especializadas. Este formato foi aceito e apoiado amplamente na transferência de imagem não associada ao *hardware* de computador. Existem muitas variações do formato, chamadas extensões, podendo apresentar problemas ocasionais ao abrir um de outra fonte. Suportam até 24 *bits* de cores e algumas versões são comprimidas usando o LZW ou outros métodos de *lossless* [36].

O SVG é a sigla para *Scalable Vector Graphics* e consiste um padrão que trabalha com imagens vetoriais. É um formato poderoso que pode ser manipulado em um editor de texto ou inserido em uma página HTML, simplesmente, copiando e colando seu código [34 p.82]. Este formato foi desenvolvido pela W3C e que surgiu oficialmente em 2001. Por se tratar de um formato vetorial, a ampliação da imagem não compromete sua visualização.

4.2 AUDIO DIGITAL

O áudio digital é a codificação binária de um sinal elétrico que representa as ondas sonoras. Existem muitos formatos de arquivos de áudio, sendo os mais conhecidos: WAV, MP3 WMA e MID; e os menos conhecidos: AIFF, AU, AAC, AC3, AMR, AMF e MPC [35 p.12].

4.2.1 FORMATO DO ARQUIVO WAV

O WAV, abreviação de *Waveform Audio Format*, é um padrão sem compressão que resulta em arquivos grandes, porém com uma qualidade máxima. Por

sua qualidade de áudio, normalmente são utilizados para gravação de sons provenientes de fontes analógicas.

Apesar de poder conter dados comprimidos, o formato mais comum utilizado pelo arquivo WAV contém áudio em forma de modulação de pulso PCM, ou *Pulse Code Modulation*, onde uma técnica de armazenamento não comprimido é utilizada. Isso possibilita a manipulação arquivo com certa facilidade.

O formato WAV é composto por três blocos principais como se segue.

O primeiro bloco é o cabeçalho do arquivo com tamanho de 12 *bytes* que contém informações de assinatura e tamanho do arquivo. A Tabela 5 demonstra as disposições e conteúdos dos *bytes* [37][38].

Tabela 5: Cabeçalho do arquivo no formato WAV.

Campo	Bytes	Descrição
<i>ChuckID</i>	4	Assinatura do arquivo: os caracteres ASCII "RIFF"
<i>ChuckSize</i>	4	Tamanho do arquivo descontado os 8 primeiros <i>bytes</i>
<i>TypeID</i>	4	Os caracteres ASCII "WAVE"

O segundo bloco, “fmt” ou *subchunk1* descreve o formato dos dados do áudio. A Tabela 6 demonstra as disposições e conteúdos dos *bytes* [37][38].

Tabela 6: Bloco fmt do arquivo WAV.

Campo	Bytes	Descrição
<i>SubChuck1ID</i>	4	Os caracteres ASCII "fmt"
<i>SubChuck1Size</i>	4	16 para PCM
<i>AudioFormat</i>	2	PCM = 1, outros valores indicam outros formatos de compressão.
<i>NumChannels</i>	2	1 = mono / 2 = estéreo
<i>SampleRate</i>	4	Taxa de amostragem. Do áudio em Hz
<i>ByteRate</i>	4	Taxa de byte do áudio = $\text{SampleRate} * \text{NumChannels} * \text{BitsPerSample} / 8$
<i>BlockAlign</i>	2	Número de <i>bytes</i> para uma amostra, incluindo todos os canais = $\text{NumChannels} * \text{BitsPerSample} / 8$
<i>BitsPerSample</i>	2	Resolução do áudio. Número de <i>bits</i> por amostra.

O terceiro bloco, “*data*”, contém os dados reais do áudio. A Tabela 7 demonstra as disposições e conteúdos dos *bytes* [37][38].

Tabela 7: Bloco de dados do áudio do arquivo WAV.

Campo	<i>Bytes</i>	Descrição
<i>SubChuck2ID</i>	4	Os caracteres ASCII “ <i>data</i> ”
<i>SubChuck2Size</i>	4	Apresenta o tamanho do restante do bloco “ <i>data</i> ”
<i>Data</i>	–	Dados para reprodução do áudio

4.3 TÉCNICAS DE ESTEGANOGRAFIA E SEU ALGORÍTMOS

Em meados de 1990 foram criadas as primeiras técnicas de esteganografia em mídias digitais, essas técnicas eram baseadas em intuição e heurísticas no lugar de fundamentos específicos [40, p.59]. Os desenvolvedores se ocuparam em fazer a incorporações imperceptíveis de informação em vez indetectáveis. Este objetivo foi, sem dúvida, causado pela falta de métodos de esteganálise que usam propriedades estatísticas e como consequência, praticamente todos os primeiros esquemas de ocultação de dados foram descobertos com sucesso mais tarde.

Devido ao avanço das técnicas de esteganálise os métodos esteganográficos se tornaram mais sofisticados, que provocaram um novo interesse em pesquisas de esteganálise criando um ciclo de interesse.

4.3.1 FINAL DO ARQUIVO

Uma das técnicas mais simples de esteganografia em mídias digitais consiste em ocultar a informação no final dos arquivos [39]. Grande parte dos formatos digitais contém em sua estrutura informações sobre o tamanho do bloco de dados nele contido, sendo assim, qualquer informação que seja incluída a partir do final do

arquivo original é automaticamente descartada na reprodução da mídia. Nesse conceito se encontram formatos como BMP, JPG, WAV, MP3 e AVI.

A utilização desta técnica, muitas vezes, não necessita de *softwares* específicos, podendo ser utilizado comandos do próprio sistema operacional. Um exemplo é a utilização do comando *copy* do *Windows*.

```
copy /b <nome arquivo de mídia>+<nome arquivo a ser adicionado>
<nome do arquivo de destino>
```

Ex.: *copy /b audio.wav+mensagem.txt audioesteganografado.wav*

Além da facilidade na utilização, esta técnica tem como vantagem a possibilidade de inclusão de informação qualquer tipo e tamanho, porém como desvantagem o fato da mensagem de texto ser descoberta com uma simples visualização do arquivo em um editor de texto. O Anexo A demonstra a utilização desta técnica. As figuras 74, 75 e 76 mostram os arquivos de áudio e texto e o resultado da esteganografia, respectivamente, vistos em um programa de edição de texto.

4.3.2 LSB

A técnica LSB, acrônimo de *Least Significant Bit*, ou seja, *bit* menos significativo, consiste em utilizar os *bits* menos significativos das cores de uma imagem ou dos dados de um áudio para incluir as informações que se deseja ocultar. Essa técnica baseia-se na teoria de que a alteração de uma parte tão pouco significativa não causa perturbação grande o suficiente para ser percebida. Em uma imagem BMP *True Color*, por exemplo, onde a cor de *pixel* é formada por 24 *bits* é possível ter 3 *bits* de informação oculta para cada *pixel* da imagem.

Algoritmo para a técnica LBS para a imagem BMP *True Color*.

m = mensagem.

i = dados imagem.

```

nm = número de bits (m)
ni = número de bits (i)
para x = 0 até nm-1
    se 8 * x >= ni então vai para o fim // não pode ocultar toda mensagem.
    i[8*x+7] = m[x]
fim para

```

A modificação de apenas um *bit* nas cores primárias torna a estegoimagem fiel à imagem original. No entanto, sua capacidade de armazenamento de dados fica significativamente limitada. Alternativas como LSB 2 *bit*, que utiliza os dois últimos *bits* da cor; LSB 3 *bit* que utiliza os três últimos *bits* da cor ou até LSB n *bit* que usam *n bits* menos significativos da cor podem ser utilizados para aumentar a capacidade de armazenamento, contudo, quanto maior a quantidade de *bits* utilizados menor será a qualidade da estegoimagem.

Uma variação do LSB n seria a técnica LSB cíclico, onde apenas o *bit* menos significativa da imagem é utilizado. Chegando ao final da imagem, a mensagem não for completamente esteganografada, volta-se ao inicio da imagem, porém utilizando o segundo *bit* menos significativo e assim sucessivamente até que todos os dados sejam esteganografados.

Algoritmo LSB cíclico para imagem BMP *True Color*.

```

m = mensagem.
i = dados imagem.
nm = número de bits (m)
ni = número de bits (i)
y=0
offset=7
para x = 0 até nm-1
    se 8 * x > ni então vai para o fim // não pode ocultar toda mensagem.
    y=0
    offset=offset-1
fim se
se offset<0 então

```

```

i[8*y+offset] = m[x]
y = y +1
fim para

```

4.3.3 ESTEGANOGRAFIA EM PALETA DE IMAGEM

Alguns formatos de arquivos trabalham com uma tabela de cores, onde são selecionadas as cores mais representativas da imagem, conhecida como paleta. Essa paleta é adicionada ao arquivo e cada cor de *pixel* do arquivo é obtida referenciando-se ao índice da cor desejada na tabela. Desta forma, nestes formatos de imagem – conhecidos como imagens indexadas –, a ordem da cor na paleta não interfere na imagem, bastando ao *pixel* fazer referência ao índice correto.

Reordenando a paleta de imagem e reindexando a imagem é possível ocultar pequenas mensagens sem modificar sua aparência. Essa técnica pode ser utilizada em formatos com GIF e PNG. Esse método é implementado no programa gifshuffle [40, p.69] e pode esconder através da permutação até $\log_2 256 = 210$ bytes de informação – para paleta de 8 bits ou 256 cores –. Como a paleta de cores não necessita de ordenação, essa técnica baseia-se em utilizar ordenações específicas para transmitir significados. Uma desvantagem desta técnica é que a pequena capacidade de armazenamento independe do tamanho da imagem.

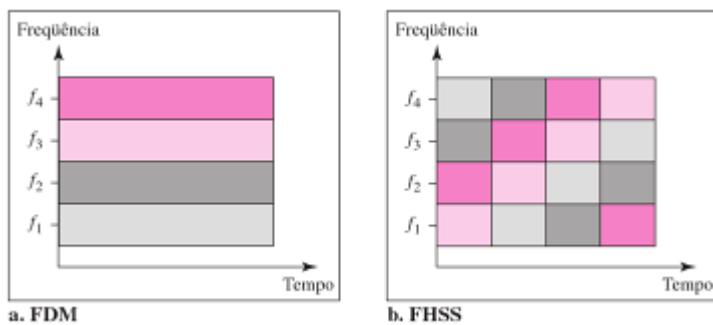
4.3.4 ESPALHAMENTO DE ESPECTRO

A técnica de espalhamento de espectro foi desenvolvida pelos militares que necessitavam transmitir informações seguras através de meio que podiam estar sob interferência ou bloqueio intencional [42], para isso a mensagem era espalhada por uma faixa grande de frequências com utilização de redundância. A impossibilidade de inspecionar grande quantidade de frequências de forma eficiente tornava essa técnica robusta.

Apesar do espalhamento espectral, como também é conhecido, ter sido desenvolvido para comunicação em rádio, o método se adapta facilmente a esteganografia em imagem possibilitando o aproveitamento de suas características. Se nas transmissões de rádio o método resiste à interferência ou bloqueio, nas imagens, ele é resistente a manipulações tais como compressão – JPEG e MPEG – ou filtragens [43].

A técnica consiste em utilizar estegoimagem como canal de transmissão por onde a mensagem a ser oculta deve ser espalhada de forma pseudoaleatória, utilizando-se uma estego-chave²⁴. Existem várias formas de espalhamento espectral, sendo as duas principais: Salto em Frequência – *Frequency Hopping* (FHSS) – e Seqüência Direta – *Direct Sequence* (DSSS) –.

A técnica FHSS utiliza um gerador de código pseudoaleatório, chamado ruído pseudoaleatório, para criar um padrão de salto de k -bits a cada intervalo de dados. A Figura 13 ilustra o FHSS sendo utilizado em um compartilhamento de banda de uma rede *wireless*.



No DSSS é utilizada uma sequência de *bits* aleatórios, chamado de sequência de espalhamento, ou código de espalhamento, com um ritmo binário muito superior ao da mensagem a ser ocultada, ou seja, para cada *bit* da mensagem é utilizado n *bits* do código de espalhamento; onde $n >> 1$. O dado a ser inserido na imagem, chamado de sinal espalhado, é obtido multiplicando-se cada *bit* da sequência de espalhamento ao *bit* equivalente contido na mensagem.

²⁴ Estego-chave – informação necessária para distribuir e recuperar a informação esteganografada no meio.

O sinal espalhado é então inserido na imagem em uma ou mais componentes de cor. Alguns autores sugerem a componente azul ou a componente de luminância. A componente de luminância é a mais utilizada, pois é de fácil acesso em grande quantidade dos formatos de imagem digital, além de que experimentalmente observar-se que está componente é mais robusta do que a componente azul.

A Figura 14 ilustra a obtenção do sinal espalhado a partir dos *bits* “101” de uma mensagem, sendo aplicado a um código de espalhamento aleatório.

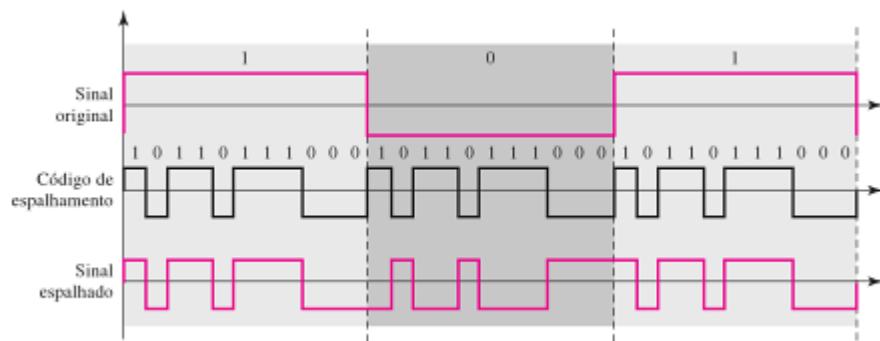


Figura 14: Obtenção do sinal espalhado utilizando-se a técnica DSSS, salientando o maior ritmo binário do código de espalhamento [44, p.185]

Podemos descrever o algoritmo DSSS como sendo a seguinte ordem [45]:

- 1) `IMG = binario (imagemContainer) /* bits dos dados da imagem que servira como canal */`
- 2) `MSG = binario (mensagem)`
- 3) `PA = geraPseudoAleatório()/* Geração da sequência pseudoaleatória (sequência de espalhamento) onde o ritmo binário deve ser muito superior ao da mensagem.*/`
- 4) `N = bitsPA_x_msg /* número de bits de PA para cada bit de MSG */`
- 5) `SE = xor (MSG, PA, N) /* Multiplicação (ou XOR) dos bits da mensagem pelos bits da sequência de espalhamento, obtendo o sinal espalhado.*/`
- 6) `SE2D = 2D (SE) /* Mapeamento do sinal espalhado em um espaço bidimensional.*/`

- 7) MD = fatorLocal (SE2D) /* Multiplicação do sinal bidimensional pelo fator local, que define em que região da imagem a informação será inserida, criando uma marca d'água.*/
- 8) IT = Transforma (IMG) /* Aplicação de uma transformação T a imagem original que receberá a informação, obtendo-se uma Imagem I_T .*/
- 9) IMI = Substitui (MD, IT) /* Adição do sinal bidimensional, aplicado o fator local obtido, à imagem transformada I_T .*/
- 10) IMF = Transformal (IMI) /* Aplicação da Transformação inversa para obtenção da imagem final.*/

4.3.5 EM ÁUDIO

O sistema auditivo humano possui, por característica, uma maior sensibilidade se comparado ao sistema visual, podendo trabalhar com faixas muito grandes de amplitude e frequências, além de captar ruídos muito baixos se comparado com o som ambiente. Porém, apesar de grandes faixas perceptíveis, essas são faixas relativas, de forma que se aumentar o ruído do ambiente, o limite mínimo perceptível também aumenta, de forma que sons altos tendem a mascarar sons tranqüilos. Situações análogas acontecem com a amplitude e a frequência. Além disso, distorções ambientais comuns tendem a ser ignoradas, na maioria dos casos.

Dentre as técnicas esteganográficas mais populares para áudio estão algumas já conhecidas como: *Bit Menos Significativo*, Espalhamento de Espectro, DTC; e técnicas que exploram as deficiências ou limitações auditivas como: Pontilhamento, *Perceptual Masking*, *Content-Adaptive*, além das mais populares Eco e Codificação em Fase.

O método de inclusão por Eco inclui em um áudio $A(t)$ o dado a ser esteganografado como um sinal de áudio com um atraso e amplitude significativamente pequenos $\alpha A(t - \Delta t)$ para torná-lo imperceptível, criando uma espécie de eco, de forma controlada. O sinal resultante pode ser então expresso como $A'(t) = A(t) + \alpha A(t - \Delta t)$. Nas técnicas mais básicas, o eco é codificado utilizando dois atrasos distintos $-\Delta t$ e $\Delta t'$ – que devem ser escolhidos cuidadosamente para possibilitar que o dado

esteganografado seja inaudível e recuperável. Como a diferença entre o original e o eco é pequena o ouvido humano não consegue distinguir entre os dois, podendo ser percebido como uma ressonância, dependendo de alguns fatores como: qualidade da gravação original, tipo de som ecoado e ouvinte. O deslocamento Δt normalmente tem valor em torno de um milissegundo e para ser inserido, o áudio original é dividido em vários blocos antes da codificação. Com essa técnica é possível incorporar uma taxa de cerca de 16 *bits* por segundo, sem que qualquer degradação do sinal seja percebida. Para a recuperação do dado, uma técnica conhecida como *cepstrum autocorrelation*, é utilizada.

Já a Codificação em Fase aproveita a menor sensibilidade do sistema auditivo humano às componentes de fase do som se comparadas às componentes de ruído, uma propriedade que também é explorada por alguns esquemas de compressão de áudio. A codificação em fase é um dos mais efetivos métodos de codificação em áudio, quando analisados os aspectos de percepção da ocultação. O método funciona intercalando fases do áudio original com segmentos de informações a ser ocultada, para isso, a Transformada de Fourier Discreta, conhecida como DFT, é utilizada. As fases são ajustadas para preservar uma relação evitando que a inclusão dos dados seja percebida. Para a extração da mensagem secreta, o receptor deve conhecer o tamanho dos segmentos. O receptor pode então usar a DFT para obter as fases e extrair a informação. Uma desvantagem desta técnica é a baixa taxa de transmissão dos dados.

4.4 ESTEGANÁLISE

A ocultação de informações dentro de mídia digitais acarreta a alterações das propriedades da mídia, podendo introduzir algumas formas de degradações ou características incomuns. Embora tais distorções possam ser de difícil percepção pelos sentidos humanos, suas características podem atuar como assinaturas que informam existência da mensagem incorporada, frustrando assim a finalidade da esteganografia. Ataques e análises sobre a informação escondida pode ocorrer de

várias formas: detectando, extraíndo, incapacitando ou destruindo as informações ocultas. Essas técnicas são denominadas esteganalises [41].

Ferramentas de esteganografia variam em seus métodos para ocultar informações, fazendo com que a detecção da informação oculta possa se tornar complexa. Apenas após avaliar um grande número de imagem sob ação da esteganografia, comparando-as com as originais – técnica conhecida como *cover* ataque –, é possível identificar a formação de padrões. Tais padrões normalmente apresentam uma disposição incomum de cor, paletas, relações entre as cores em índices de cor, luminosidade e ruídos exagerados. Desta forma é realizada a detecção da esteganografia [41].

Entre as possibilidades de ataque, a desativação ou remoção de informações escondidas em imagens é a mais simples de ser implantada, pois não é necessário o reconhecimento da utilização da esteganografia. A técnica se resume ao processamento de imagem. Técnicas de esteganografia como manipulação do *bit* menos significativo, por exemplo, são facilmente inutilizadas utilizando-se algoritmos de compressão *lessy*²⁵, tal como JPEG. Esse processo é suficiente para renderizar à mensagem incorporada tornando-a inútil.

Existem ferramentas para testar a robustez da esteganografia. Estas ferramentas automatizam processos como a deformação, recorte, rotação e perda de foco nos arquivos analisados.

A extração da informação pode variar de situações extremamente fáceis, quando descoberta da esteganografia, tais como leitura da informação em cabeçalho ou no final do arquivo a técnicas mais complexas, podendo não ser possível sua legibilidade quando associadas a criptografias e espalhamentos de dados.

²⁵ Compressão *lessy* – compressão com perda de informação.

5 IMPLEMENTAÇÃO

Este capítulo dedica-se a demonstrar o desenvolvimento da aplicação proposta neste trabalho utilizando o paradigma de Orientação a Objetos, também conhecido como OO, utilizando um ciclo de vida cascata. Serão mostradas as partes relevantes de sua construção desde a motivação para sua criação à suas limitações, abordando as fases de análise, projeto e codificação; sendo que na codificação, apenas os algoritmos relevantes serão tratados. O código por completo poderá ser obtido no endereço <https://db.tt/BJ410tyu>.

5.1 DEFINIÇÃO DO PROBLEMA

A utilização das técnicas esteganográficas em mídias digitais é de difícil aplicação para usuários, principalmente leigos em computação, sem a utilização de softwares específicos para esta finalidade. Conseguir implementar esses métodos é uma tarefa que, na maioria dos casos, necessita além de conhecimento da estrutura dos arquivos, de manipulação de seus dados em nível de *bits*, tarefa que é pouco recomendada e que não dispõe de programas nativos nos sistemas operacionais que permitam essa realização. Mesmo que a tarefa de ocultar a informação fosse realizada pelo remetente, o destinatário precisaria do mesmo nível de conhecimento, esforço e habilidade para conseguir extrair a mensagem desejada.

O software desenvolvido neste trabalho busca facilitar essa troca de informação esteganografada, possibilitando a inclusão e recuperação de dados em forma de textos e arquivos em mídias de imagem, áudio e vídeo. Para isso, serão utilizadas as técnicas de esteganografia em final de arquivo e LSB.

5.2 REQUISITOS

Existe apenas um ator que é o usuário do sistema, podendo se comportar de duas formas:

- 1) Como remetente – realizará a inclusão da informação dentro da mídia.
- 2) Como destinatário – realizará a recuperação da informação desejada inclusa na mídia recebida.

Requisitos funcionais:

- 1) O sistema deve possibilitar a inclusão de um texto, utilizando a técnica de final de arquivo, em: um arquivo de imagem – nos formatos BMP ou JPG –, áudio – nos formatos WAV ou MP3 –, ou vídeo – no formato AVI –.
- 2) O sistema deve possibilitar a inclusão de um arquivo, utilizando a técnica de final de arquivo, em: um arquivo de imagem – nos formatos BMP ou JPG –, áudio – nos formatos WAV ou MP3 –, ou vídeo – no formato AVI –.
- 3) O sistema deve possibilitar a inclusão de um texto, utilizando a técnica LSB, em: um arquivo de imagem no formato BMP ou áudio no formato WAV.
- 4) O sistema deve possibilitar a inclusão de um arquivo, utilizando a técnica LSB, em: um arquivo de imagem no formato BMP ou áudio no formato WAV.
- 5) O sistema deve possibilitar a inclusão de um texto, utilizando a técnica LSB 2, em: um arquivo de imagem no formato BMP ou áudio no formato WAV.
- 6) O sistema deve possibilitar a inclusão de um arquivo, utilizando a técnica LSB 2, em: um arquivo de imagem no formato BMP ou áudio no formato WAV.
- 7) O sistema deve possibilitar a inclusão de um texto, utilizando a técnica LSB 3, em: um arquivo de imagem no formato BMP ou áudio no formato WAV.

- 8) O sistema deve possibilitar a inclusão de um arquivo, utilizando a técnica LSB 3, em: um arquivo de imagem no formato BMP ou áudio no formato WAV.
- 9) O sistema deve possibilitar a inclusão de um texto, utilizando a técnica LSB n, em: um arquivo de imagem no formato BMP ou áudio no formato WAV.
- 10) O sistema deve possibilitar a inclusão de um arquivo, utilizando a técnica LSB n, em: um arquivo de imagem no formato BMP ou áudio no formato WAV.
- 11) O sistema deve possibilitar a inclusão de um texto, utilizando a técnica LSB cíclico, em: um arquivo de imagem no formato BMP ou áudio no formato WAV.
- 12) O sistema deve possibilitar a inclusão de um arquivo, utilizando a técnica LSB cíclico, em: um arquivo de imagem no formato BMP ou áudio no formato WAV.
- 13) O sistema não deve permitir que dados maiores do que o suportado pelo arquivo de mídia sejam esteganografados.
- 14) O sistema deve recuperar a informação esteganografada, caso exista, de arquivos BMP, JPG, WAV, MP3 e AVI; independente da técnica de esteganografia utilizada – final de arquivo, LBS, LSB 2, LSB 3, LSB n ou LSB cíclico – sem que haja necessidade do destinatário identificá-la.
- 15) Ao recuperar o dado esteganografado, o sistema deve exibir a informação se for uma mensagem de texto ou se for um arquivo, possibilitar sua extração e gravação em um diretório a ser informado pelo destinatário.

Requisitos não funcionais:

- 1) O sistema deve ser desenvolvido para ser executado localmente.
- 2) O sistema deve desenvolvido em Java.

5.3 ESPECIFICAÇÃO

As especificações deste *software* foram realizadas aplicando o padrão UML²⁶ e utilizando como ferramenta o Cacoo²⁷.

5.3.1 DIAGRAMA DE CASO DE USO

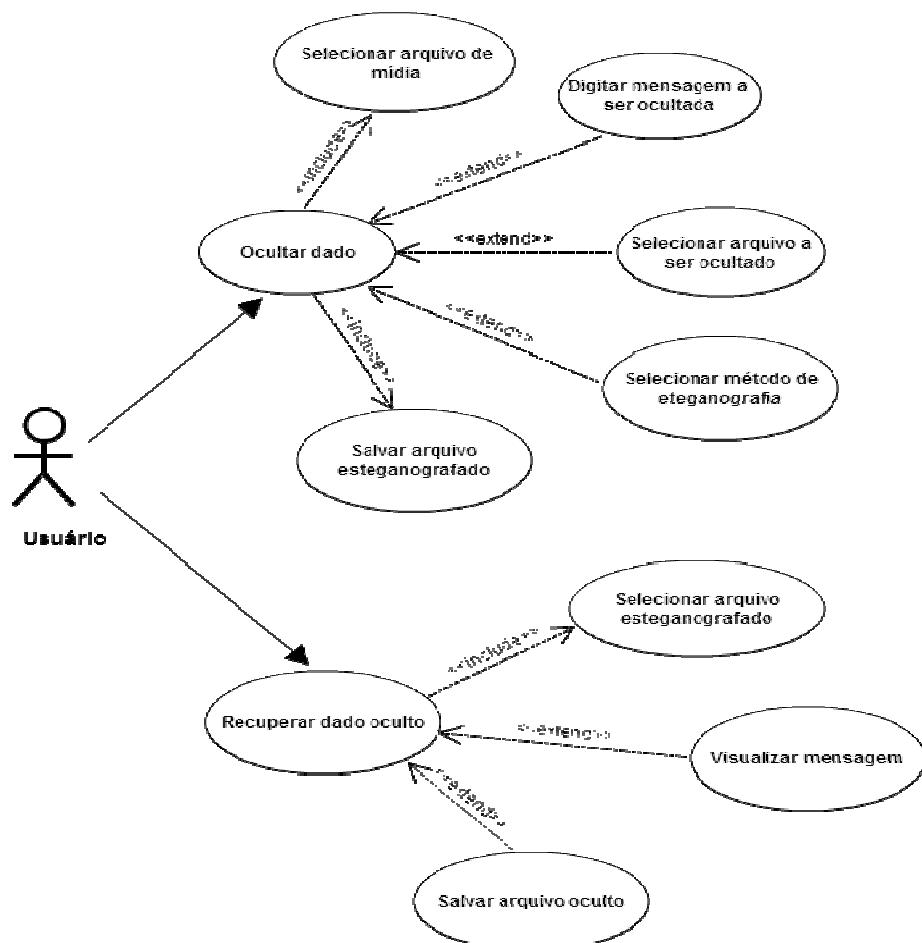


Figura 15: Diagrama de Caso de Uso

²⁶ UML – *Unified Modeling Language*, ou Linguagem de Modelagem Unificada, é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de *software* [46].

²⁷ Cacoo – Ferramenta de diagramação *online* disponível em https://cacoo.com/lang/pt_br/. Acessado em Setembro de 2015.

5.3.2 DESCRIÇÃO DE CASO DE USO

Descrição do caso de uso, cujo diagrama está representado na Figura 15, referente à operação de ocultação do dado.

Nome:	Ocultar dado
Objetivo:	Ocultar um arquivo ou mensagem de texto em arquivos de mídia
Atores:	Usuário do sistema
Pré-condições:	Nenhuma
Trigger:	Opção de ocultar dado escolhida pelo usuário
Fluxo Principal:	<ol style="list-style-type: none"> 1. O sistema exibe os campos Arquivo de Mídia; um grupo com os campos Mensagem e Arquivo; As opções de seleção Final de Arquivo, LSB, LSB 2, LSB 3, LSB n e LSB Cíclico; e o botão Esteganografar 2. O usuário clica em Arquivo de Mídia 3. O sistema abre uma janela para que o usuário selecione o arquivo de mídia 4. O usuário seleciona o arquivo de mídia 5. O usuário digita a mensagem no campo Mensagem 6. O usuário seleciona uma das opções disponíveis: Final de Arquivo, LSB, LSB 2, LSB 3, LSB Cíclico 7. O usuário clica no botão Esteganografar 8. O sistema realiza a esteganografia de acordo com as opções selecionadas. 9. O sistema apresenta uma janela para o usuário selecionar o local de gravação e digitar o nome do arquivo 10. O usuário seleciona o arquivo e local e clica em gravar 11. O sistema salva o arquivo esteganografado
Fluxo Alternativo:	4.1: O usuário cancela a seleção do arquivo de mídia. <ol style="list-style-type: none"> 1. O sistema retorna ao passo 1 do fluxo principal.

5.1: Seleciona a opção arquivo.

1. O sistema abre uma janela para que o usuário selecione o arquivo a ser ocultado

2. O usuário seleciona o arquivo a ser ocultado

3. Retorna ao passo 6 do fluxo principal

6.1: O usuário seleciona a opção LSB n

1. O sistema habilita um campo para que o usuário indique o número de *bits* da técnica LSB n

2. O usuário digita o número de *bits*

3. Volta ao passo 7 do fluxo principal

10.1: O usuário cancela gravação do arquivo

1. Volta ao passo 7 do fluxo principal

Extensões: A qualquer momento pode ser executada a opção de sair do sistema.

Pós-condições: Arquivo de mídia gravado com o dado esteganografado.

Regras de negócio: RN1: O sistema só deve permitir seleção de arquivo de Mídia nos formatos BMP, JPG, WAV, MP3 e AVI.

RN2: O sistema só deve permitir a digitação da mensagem ou a seleção do arquivo de forma exclusiva.

RN3: O sistema só deve permitir a seleção das opções de esteganografia LSB, LSB 2, LSB 3, LSB n ou LSB Cíclico caso o arquivo de mídia selecionado seja do tipo BMP ou WAV.

RN4: O sistema não deve permitir esteganografia de dados maior do que o suportado pelo arquivo de mídia.

RN5: O valor do campo *n* digitado na opção LSB n deve ser maior que três e menor que oito.

RN6: A extensão do arquivo esteganografado deve ser a mesma do arquivo de mídia selecionado.

Descrição do caso de uso, representado na Figura 15, referente à operação de recuperação de dado oculto.

Nome:	Recuperar dado
Objetivo:	Recuperar o dado de um arquivo de mídia quando identificar a existência de esteganografia
Atores:	Usuário do sistema
Pré-condições:	Nenhuma
Trigger:	Opção de recuperar dado escolhida pelo usuário
Fluxo Principal:	<ol style="list-style-type: none"> 1. O sistema apresenta o campo de Arquivo de Mídia e o botão Recuperar Dado. 2. O usuário clica em Arquivo de Mídia. 3. O sistema abre uma janela para que o usuário selecione o arquivo de mídia. 4. O usuário seleciona o arquivo de mídia. 5. O usuário clica no botão Recuperar Dado 6. O Sistema faz a recuperação do dado esteganografado 7. Se for uma mensagem de texto, o sistema exibe a informação.
Fluxo Alternativo:	<ol style="list-style-type: none"> 4.1: O usuário cancela a seleção do arquivo de mídia. <ol style="list-style-type: none"> 1. O sistema retorna ao passo 1 do fluxo principal 7.1: O dado esteganografado é um arquivo. <ol style="list-style-type: none"> 1. O sistema apresenta uma janela para o usuário selecionar o local de gravação e digitar o nome do arquivo 2. O usuário seleciona o arquivo e local e clica em gravar. 3. O sistema salva o conteúdo oculto como arquivo
Extensões:	A qualquer momento pode ser executada a opção de sair do sistema.
Pós-condições:	O dado recuperado em forma de arquivo ou mensagem de texto.
Regras de negócio:	RN1: O usuário não necessita saber o tipo de técnica este-

ganográfica foi utilizada para ocultação do dado.

RN2: O dado recuperado tem que ser idêntico ao dado oculto.

RN3: A extensão do arquivo recuperado deve ser a mesma do arquivo que foi oculto no processo esteganográfico.

5.3.3 DIAGRAMA DE ATIVIDADE

A Figura 16 ilustra o diagrama de atividades do caso de uso referente à operação de ocultação do dado.

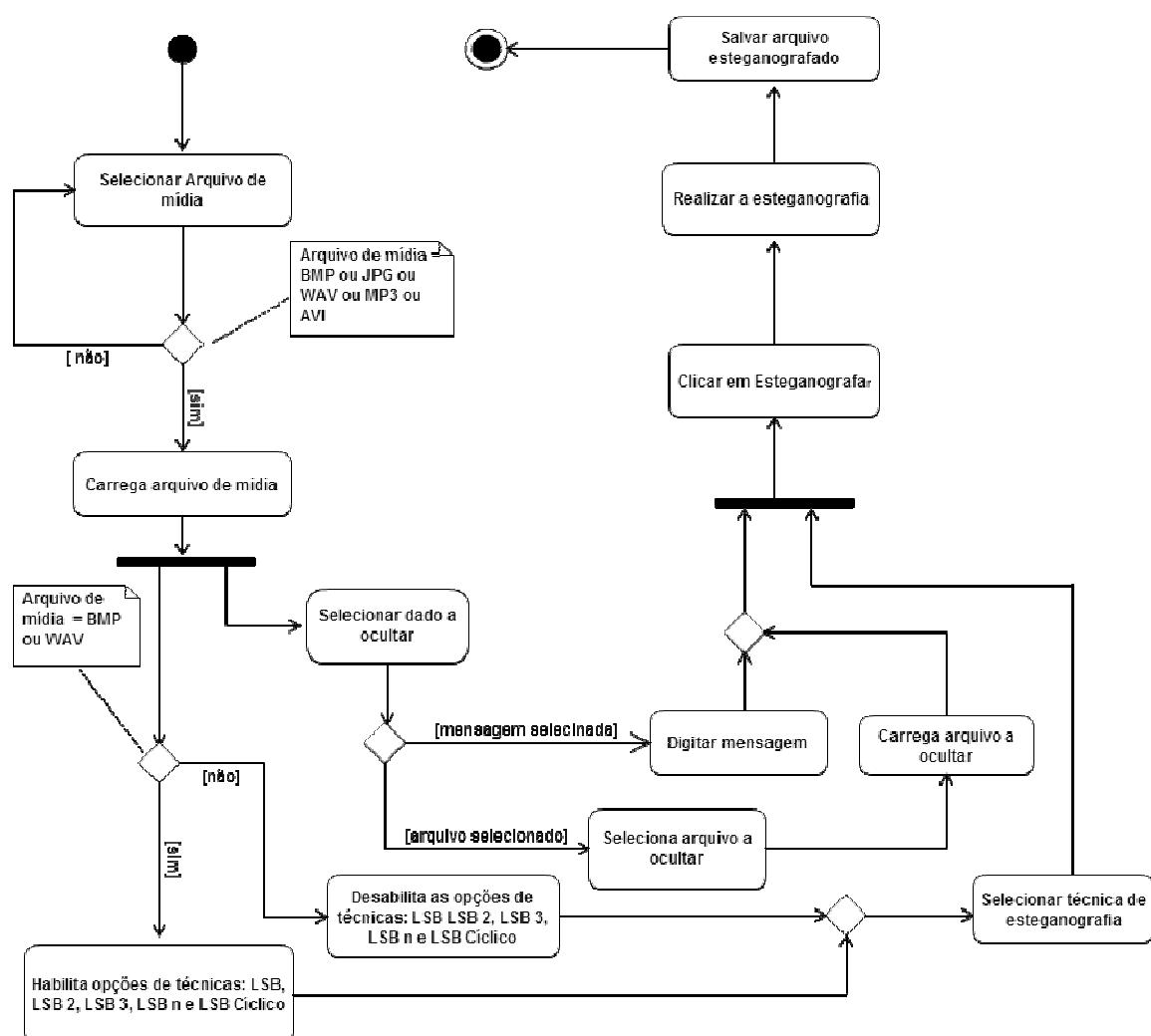


Figura 16: Diagrama de atividades do caso de uso ocultação do dado.

A Figura 17 ilustra o diagrama de atividades do caso de uso referente à operação de recuperação de dado ocultado.

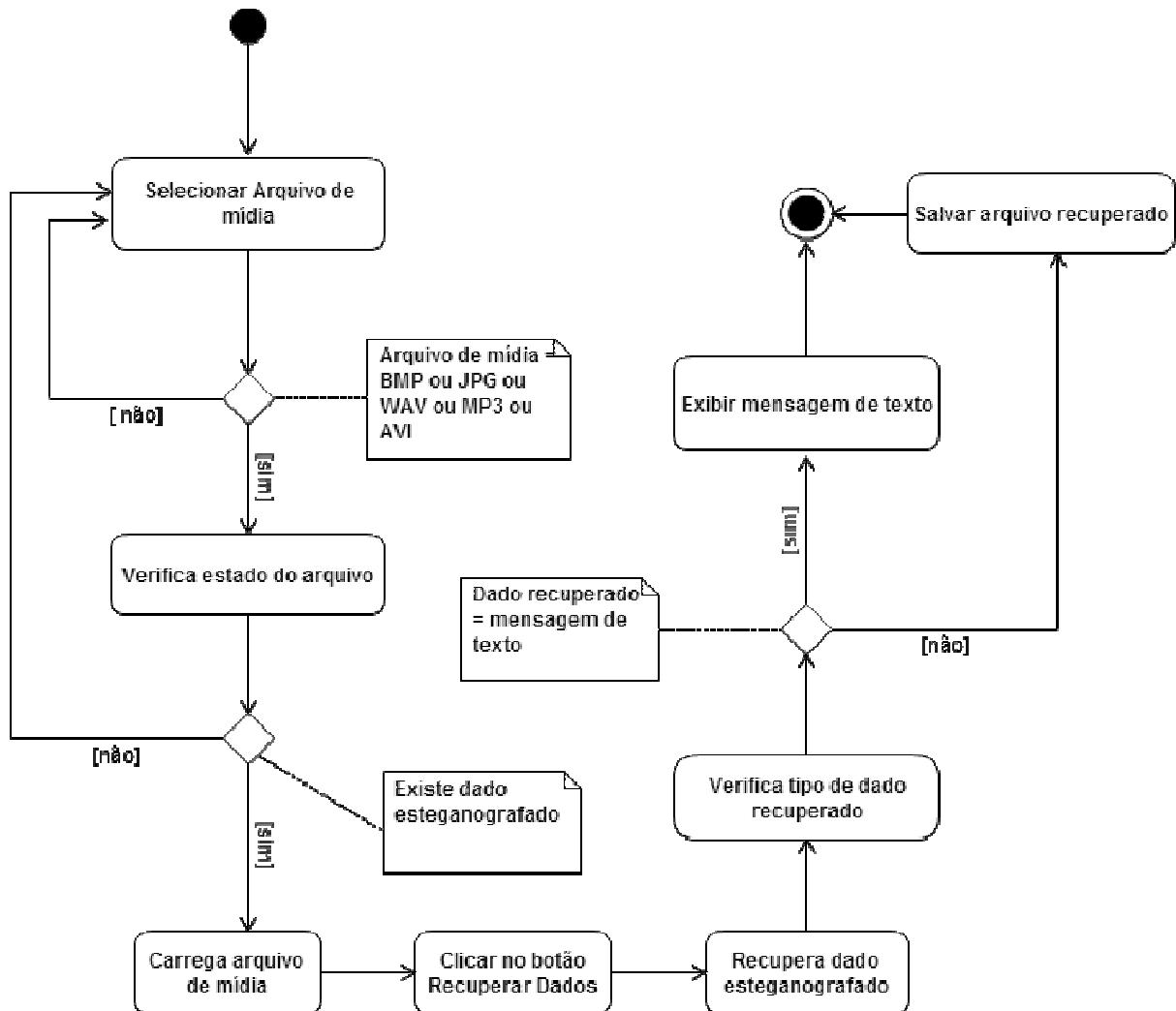


Figura 17: Diagrama de atividades do caso de uso de recuperação de dado.

5.4 MODELAGEM DE DADOS

A modelagem dos dados deste *software* foram realizadas aplicando o padrão UML e utilizando como ferramenta o Cacoo.

5.4.1 DIAGRAMA DE CLASSES

As classes utilizadas no desenvolvimento do sistema proposto estão modeladas através do diagrama na Figura 18.

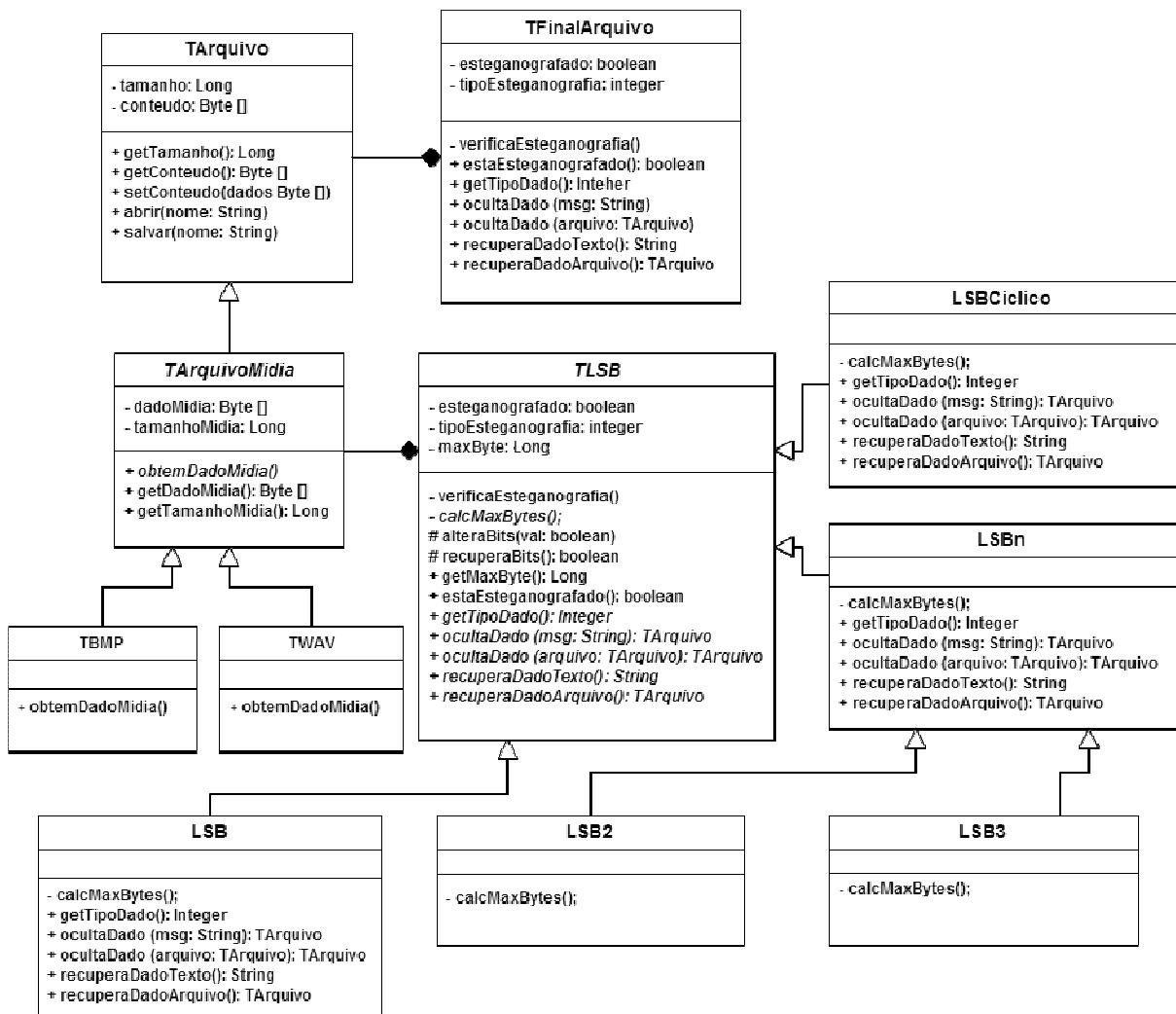


Figura 18: Diagrama de classes do *software* proposto.

O modelo de classe conceitual apresentado é composto pelas classes:

- 1) **TArquivo**, será responsável por abrir, salvar, e armazenar o conteúdo do arquivo;
- 2) **TFinalArquivo**, será responsável por realizar a esteganografia com a técnica final de arquivo;

- 3) **TArquivoMidia**, será responsável por armazenar a parte do arquivo de mídia referente aos dados, desconsiderando cabeçalhos e quaisquer outras informações que não serão manipulados. É uma classe *abstract*, pois a posição dos dados varia dependendo do formato do arquivo, por isso o método abstrato obtemDadoMidia, desta classe, será implementada em suas especializações;
- 4) **TBMP**, especialização de TArquivoMidia, implementa o método abstrato obtemDadoMidia;
- 5) **TWAV**, especialização de TArquivoMidia, implementa o método abstrato obtemDadoMidia;
- 6) **TLSB**, será responsável pela manipulação de forma livre dos *bits* dos dados dos arquivos de mídia, pelo método protegido alteraBits, por esse motivo é uma classe abstrata, para que suas especializações definam a regra de manipulação dos *bits*;
- 7) **LSB**, especialização de TLSB que utiliza o método alteraBits para realizar a alteração segundo a técnica LSB;
- 8) **LSBn**, especialização de TLSB que utiliza o método alteraBits para realizar a alteração segundo a técnica LSB n;
- 9) **LSB2**, especialização de LSBn especificando o $n = 2$, no construtor;
- 10) **LSB3**, especialização de LSBn especificando o $n = 3$, no construtor;
- 11) **LSBCiclico**, especialização de TLSB que utiliza o método alteraBits para realizar a alteração segundo a técnica LSB Cíclico.

5.4.2 DIAGRAMAS DE SEQUÊNCIA

O diagrama da Figura 19 é executado para realizar a esteganografia ocultando uma mensagem de texto no final de um arquivo de mídia. A sequência inicia com a criação de um objeto do tipo TArquivo que será responsável pela leitura do arquivo de mídia através do método abrir, onde é passado o nome do arquivo. Em seguida é criado o objeto do TFinalArquivo, que recebe o arquivo carregado. Esse objeto será responsável por realizar a esteganografia utilizando o método ocultarDa-

do. Esse método recebe como parâmetro a mensagem a ser ocultada, obtém o conteúdo do arquivo original, utilizando o método `getConteudo`. Os dados modificados são carregados no objeto `TArquivo` para que sejam salvos, utilizando o método `salvar`.

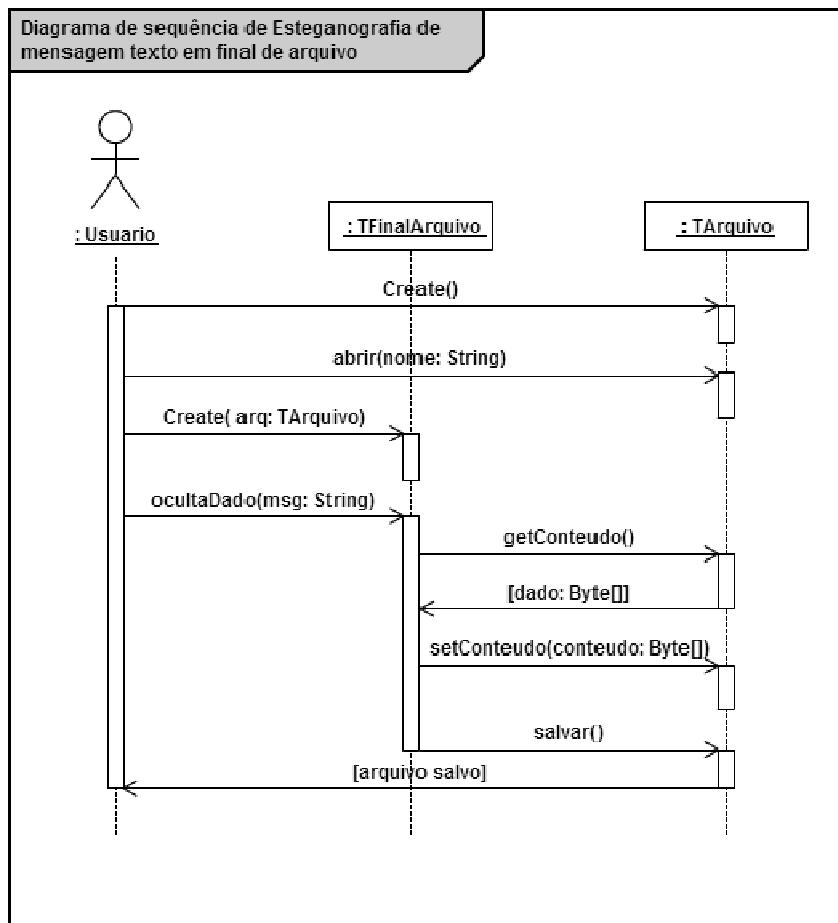


Figura 19: Diagrama de Sequência para a ocultação de mensagem texto utilizando a técnica de final de arquivo.

O diagrama da Figura 20 demonstra a sequência da esteganografia que oculta um arquivo dentro de uma mídia, incluindo o arquivo a ser ocultado no final do arquivo de mídia. São criados dois objetos do tipo `TArquivo`: um que carregará, utilizando o método `abrir`, o arquivo de mídia; e outro que carregará o arquivo a ser ocultado, também através da chamada do método `abrir` que recebe como parâmetro o nome do arquivo a ser carregado. Em seguida, um objeto do tipo `TFinalArquivo` é criado, recebendo como parâmetro o objeto contendo o arquivo de mídia. Em seguida, o método `ocultaDado` é chamado, e o objeto contendo o conteúdo do arquivo a ser ocultado é adicionado ao objeto do tipo `TFinalArquivo`. Para realizar a estegano-

grafia, o conteúdo dos dois arquivos são obtidos pelo método `getConteudo`. Após os dados serem manipulados, o conteúdo obtido é carregado no objeto `TArquivo` para que seu método `salvar` seja executado para realização da gravação do conteúdo esteganografado em forma de arquivo de mídia.

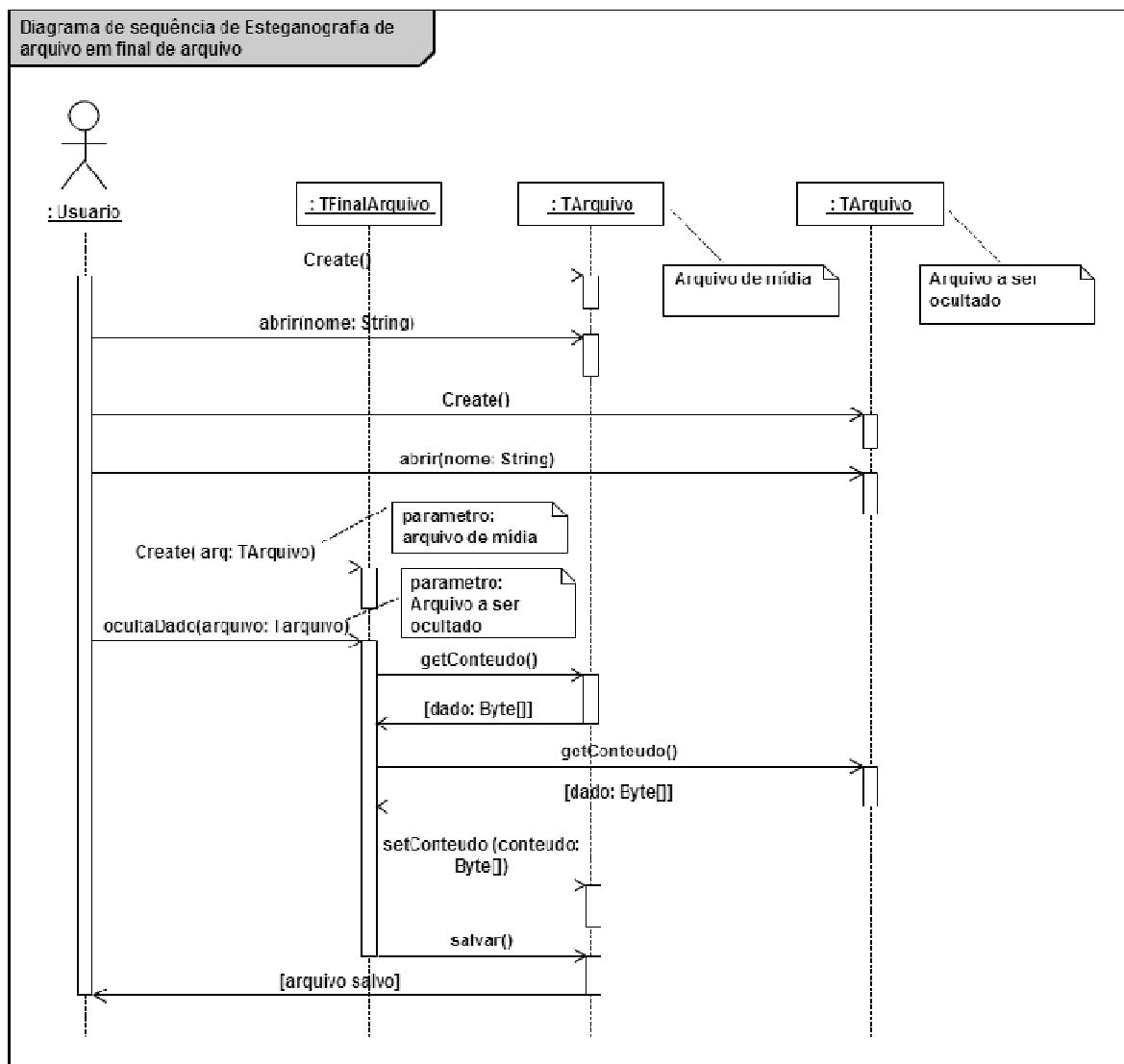


Figura 20: Diagrama de Sequência para a ocultação de arquivo utilizando a técnica de final de arquivo.

O diagrama da Figura 21 ilustra uma sequência genérica para realização da esteganografia de uma mensagem texto utilizando as técnicas LSB, LSB 2, LSB 3, LSB n e LSB Cíclico em um arquivo de mídia. Em específico, os tipos BMP e WAV. A sequência inicia com a criação de um objeto do tipo `TArquivoMidia` – que é uma generalização das classes `TBMP` ou `TWAV`, dependendo do arquivo de mídia

carregado –, que será responsável pela leitura do arquivo de mídia através do método abrir, onde é passado o nome do arquivo. Em seguida é criado o objeto do TLSB – que é uma generalização das classes LSB, LSB2, LSB3, LSBn ou LSBCiclico, dependendo apenas da escolha do tipo de técnica esteganográfica –, que recebe o arquivo carregado. Esse objeto será responsável por realizar a esteganografia utilizando o método ocultarDado.

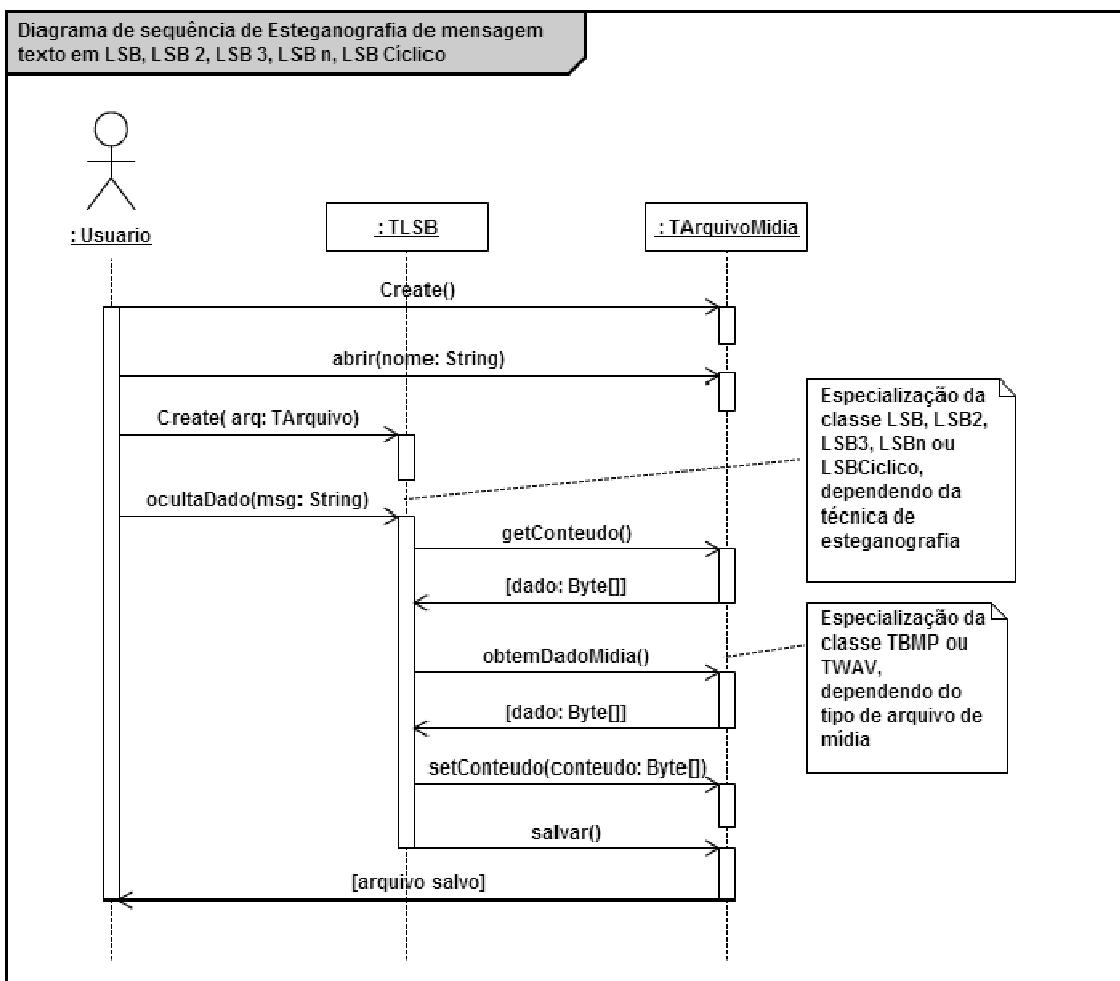


Figura 21: Diagrama de Sequência para a ocultação de mensagem texto utilizando as técnicas de LSB, LSB 2, LSB 3, LSB n, LSB Cíclico.

O método ocultarDado recebe como parâmetro a mensagem a ser oculta, obtém o conteúdo do arquivo original, utilizando o método getConteudo e destaca os dados manipuláveis da mídia utilizando o método obtemDadoMidia. O método obtemDadoMidia específico para cada tipo de arquivo de mídia, sendo implementado nas classes especializadas TBMP e TWAV. Os dados modificados são carrega-

dos no objeto TArquivoMidia pelo método setConteudo para que sejam salvos, utilizando o método salvar.

O diagrama da Figura 22 é referente a sequência da esteganografia de arquivo utilizando as técnicas LSB, LSB 2, LSB 3, LSB n e LSB Cílico, generalizadas pelo objeto do tipo TLSB, em uma arquivo de mídia. Neste caso, específicos para os formatos BMP e WAV que são generalizados pelo objeto TArquivoMidia. São criados dois objetos para carregar os arquivos: um do tipo TArquivoMidia que carregará, utilizando o método abrir, o arquivo de mídia; e outro do tipo TArquivo que carregará o arquivo a ser ocultado, também através da chamada ao método abrir que recebe como parâmetro o nome do arquivo a ser carregado. Em seguida, um objeto do tipo TLSB – que é uma generalização das classes LSB, LSB2, LSB3, LSBr ou LSBCiclico, dependendo apenas da escolha do tipo de técnica esteganográfica – é criado, recebendo como parâmetro o objeto contendo o arquivo de mídia. Em seguida, o método ocultaDado é chamado, e o objeto contendo o conteúdo do arquivo a ser ocultado é adicionado ao objeto do tipo TLSB. Para realizar a esteganografia, o conteúdo dos dois arquivos são obtidos pelo método getConteudo, sendo que a parte alterável do arquivo de mídia é destacado pelo método obtemDadoMidia. O método obtemDadoMidia específico para cada tipo de arquivo de mídia, sendo implementado nas classes especializadas TBMP e TWAV. Após os dados serem manipulados, o conteúdo obtido é carregado no objeto TArquivoMidia para que seu método salvar seja executado para realização da gravação do conteúdo esteganografado em forma de arquivo de mídia.

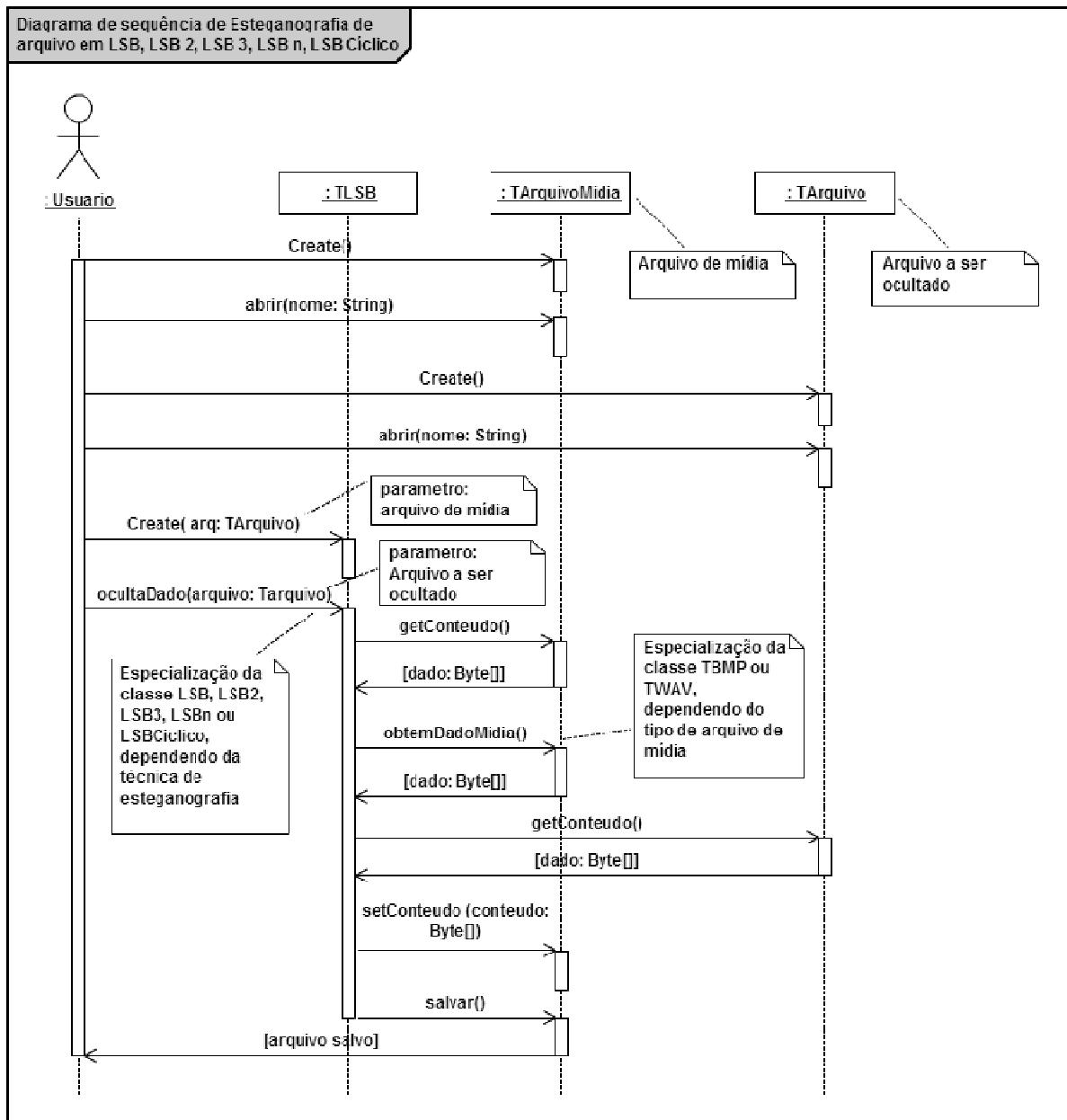


Figura 22: Diagrama de Sequência para a ocultação de arquivo utilizando as técnicas LSB, LSB 2, LSB 3, LSB n, LSB Cíclico.

A sequência do diagrama da Figura 23 é responsável pela recuperação de uma mensagem de texto esteganografada no final de um arquivo de mídia. A sequência é iniciada com a criação de um objeto do tipo TArquivo que será responsável pela leitura do arquivo de mídia, contendo a esteganografia, através do método abrir, onde é passado o nome do arquivo. Em seguida é criado o objeto do TFinalArquivo, que recebe o arquivo carregado. Esse objeto será responsável por realizar a recuperação da mensagem utilizando o método recuperarDadoTxt. O conteúdo do

arquivo de mídia é obtido utilizando o método `getConteudo`. Então, a mensagem extraída do conteúdo obtido.

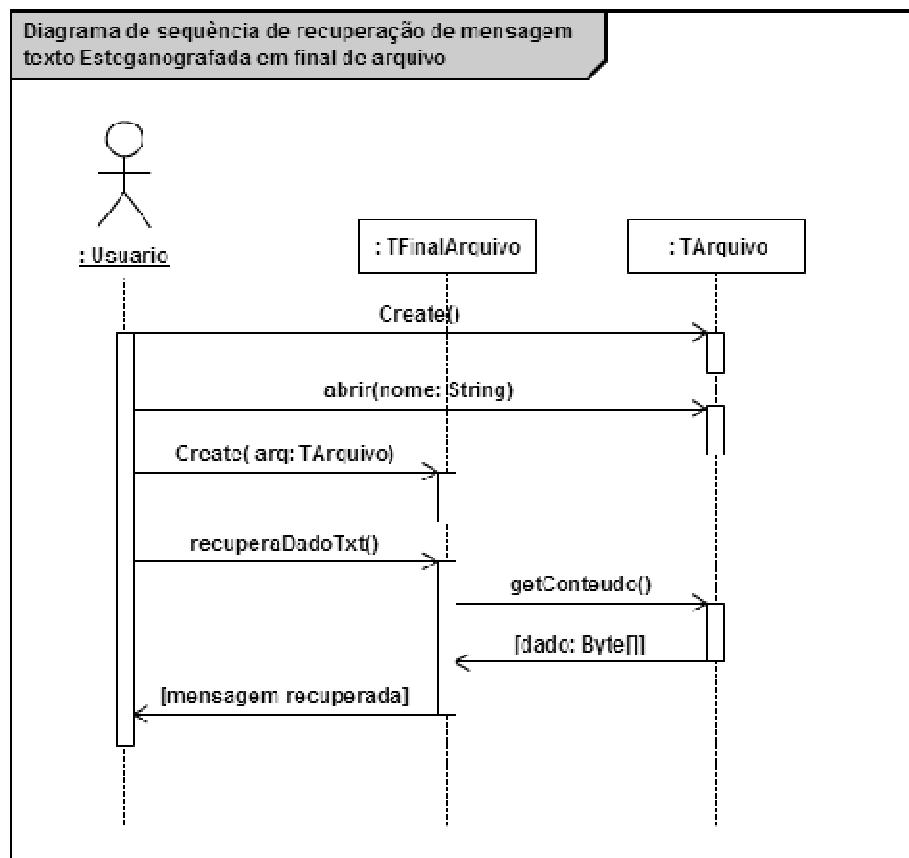


Figura 23: Diagrama de Sequência para a recuperação de mensagem texto quando utilizada a técnica de final de arquivo.

Na Figura 24, a sequência da recuperação de um arquivo, escondido no final de um arquivo de mídia e iniciada com a criação de um objeto do tipo `TArquivo` que será responsável pela leitura do arquivo de mídia, contendo a esteganografia, através do método `abrir`, onde é passado o nome do arquivo. Em seguida é criado o objeto do `TFinalArquivo`, que recebe o arquivo carregado. Esse objeto será responsável por realizar a recuperação da mensagem utilizando o método `recuperarDadoArquivo`. O conteúdo do arquivo de mídia é obtido utilizando o método `getConteudo`. O conteúdo do arquivo escondido é obtido e carregado no objeto do tipo `TArquivo` para que seja salvo pela utilização do método `salvar`.

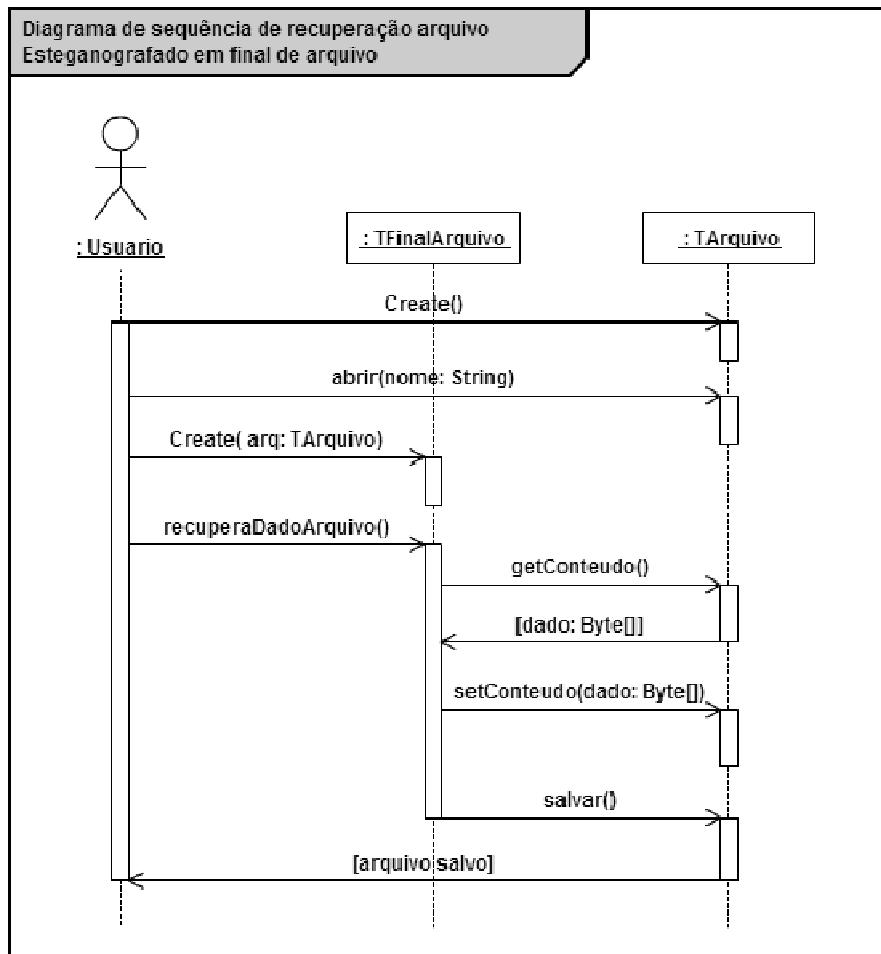


Figura 24: Diagrama de Sequência para a recuperação de arquivo quando utilizada a técnica de final de arquivo.

O diagrama da Figura 25 ilustra uma sequência genérica para realização da recuperação de uma mensagem de texto utilizando as técnicas LSB, LSB 2, LSB 3, LSB n e LSB Cílico em um arquivo de mídia. Em específico, os tipos BMP e WAV. A sequência inicia com a criação de um objeto do tipo `TArquivoMidia` – que é uma generalização das classes `TBMP` ou `TWAV`, dependendo do arquivo de mídia carregado –, que será responsável pela leitura do arquivo de mídia através do método `abrir`, onde é passado o nome do arquivo. Em seguida é criado o objeto do `TLSB` – que é uma generalização das classes `LSB`, `LSB2`, `LSB3`, `LSBn` ou `LSBCiclico`, dependendo apenas da técnica esteganográfica identificada –, que recebe o arquivo carregado. Esse objeto será responsável por realizar a recuperação da mensagem utilizando o método `recuperarDadoTxt`. Esse método obtém o conteúdo do arquivo original, utilizando o método `getConteudo` e destaca os dados manipuláveis da mídia utilizando o método `obtemDadoMidia`. O método `obtemDadoMidia` específico para

cada tipo de arquivo de mídia, sendo implementado nas classes especializadas TBMP e TWAV. Com as informações necessárias obtidas, a mensagem é recuperada.

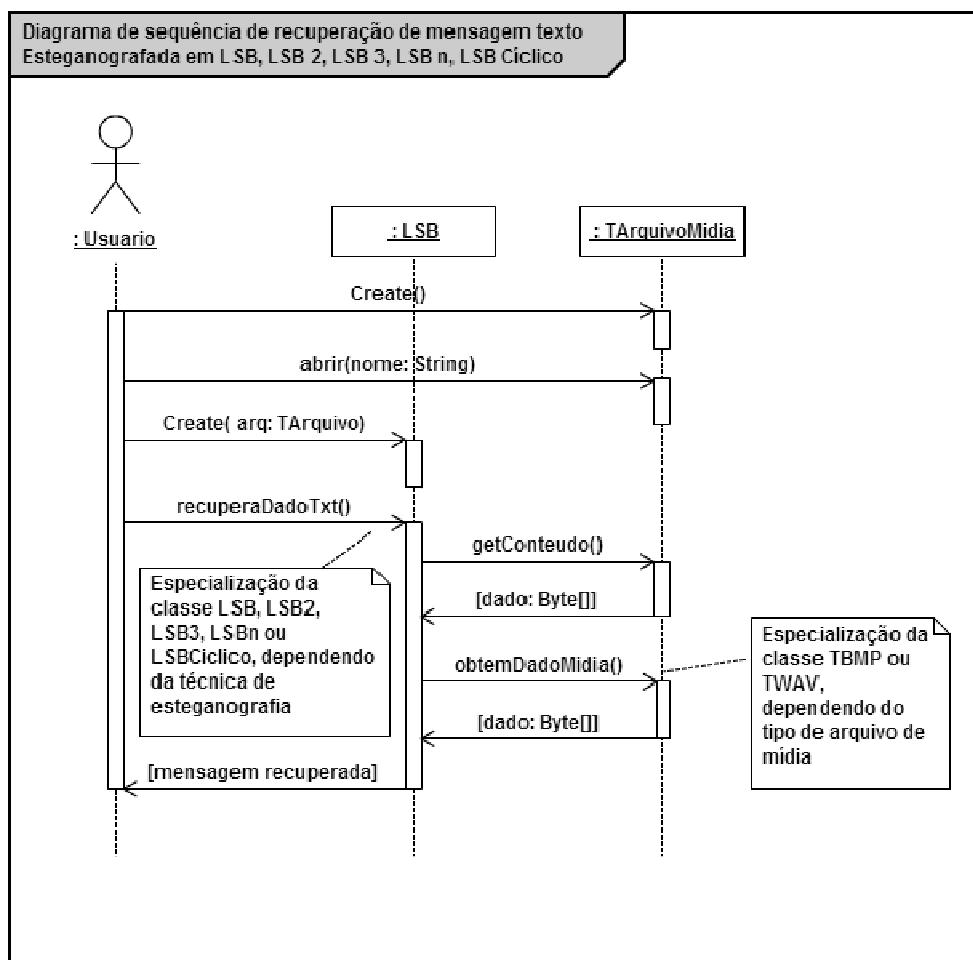


Figura 25: Diagrama de Sequência para a recuperação de mensagem texto quando utilizada as técnicas LSB, LSB 2, LSB 3, LSB n, LSB Cíclico.

Por último, o diagrama de sequência ilustrado na Figura 26 demonstra a recuperação de um arquivo ocultado pelas técnicas LSB, LSB 2, LSB 3, LSB n e LSB Cílico, generalizadas pelo objeto do tipo TLSB, em um arquivo de mídia que é generalizados pelo objeto TArquivoMidia. Um objeto é criado para carregar os arquivos, utilizando o método `abrir`, que recebe como parâmetro o nome do arquivo a ser carregado. Em seguida, um objeto do tipo é criado, recebendo como parâmetro o objeto contendo o arquivo de mídia. O método `recuperarDadoArquivo` é chamado, e o objeto contendo o conteúdo do arquivo original é adicionado ao objeto do tipo TLSB através do método `getConteudo`. A parte de interesse do arquivo de mídia é

destacado pelo método obtemDadoMidia que é específico para cada tipo de arquivo de mídia, sendo implementado nas classes especializadas TBMP e TWAV. Após recuperação do conteúdo escondido, este é carregado no objeto TArquivoMidia para que seu método salvar seja executado para realização da gravação.

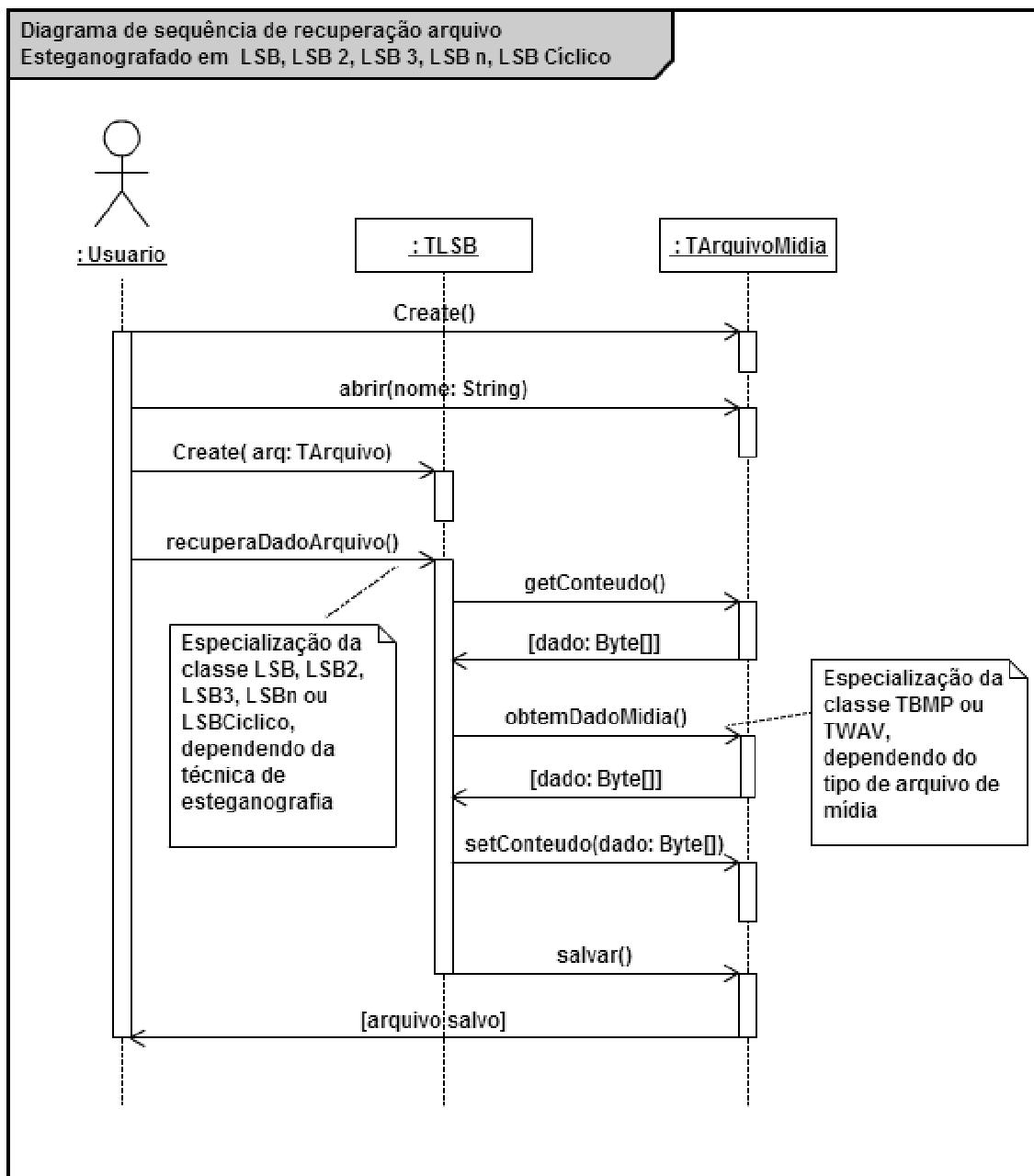


Figura 26: Diagrama de Sequência para a recuperação de arquivo quando utilizada as técnicas LSB, LSB 2, LSB 3, LSB n, LSB Cíclico.

5.5 VISÃO GERAL DO SOFTWARE

O *software* consiste em uma aplicação local desenvolvida em java que permite a esteganografia em arquivos de mídias digitais nos formatos: BMP, JPG, WAV, MP3 e AVI. Para realizar a ocultação, o programa implementa as técnicas de adição de informação no final do arquivo e a técnica LSB com suas variações: LSB 2, LSB 3, LSB n e LSB Cíclico.

Em todas as técnicas implementadas é possível ocultar tanto dado no formato de texto, quanto arquivos digitais; sendo que a técnica LSB e suas derivações poderão ser utilizadas apenas em arquivos no formato BMP e WAV.

Além do processo de esteganografia, o *software* possibilita a recuperação da informação esteganografada sem que haja necessidade de que o usuário conheça e informe o método que foi utilizado no processo de esteganografia. Para isso, além dos dados, é necessário incluir nas estegoimagens informações referentes ao tipo de esteganografia utilizada, tamanho do dado ocultado e em caso de ocultação de arquivo, o tipo do arquivo que foi escondido.

Desta forma, quando utilizada a técnica de final de arquivo, para inclusão das informações extras, é adicionado um rodapé que pode ser de 9 *bytes* quando a informação esteganografada é uma mensagem de texto e 13 *bytes* quando o conteúdo esteganografado é um arquivo digital. Em ambos os casos, os quatro últimos *bytes* serão a assinatura que identifica um arquivo contendo esteganografia realizada por este *software*. Essa assinatura é composta pelos caracteres em ASCII “TCUF”. Precedendo a assinatura, existirá 1 *byte* que identifica se o dado escondido refere-se a uma mensagem de texto, neste caso o conteúdo do *byte* receberá o caractere ASCII “M”; ou se o dado se refere a um arquivo, neste caso o conteúdo do *byte* será o caractere ASCII “A”. Completando os 9 *bytes* do rodapé, para mensagem em texto, os 4 *bytes* restantes conterão o tamanho da mensagem. Para arquivos ocultos, serão adicionados 4 *bytes* no início do rodapé, completando os 13 *bytes* informados. Esses *bytes* serão preenchidos com os caracteres, em ASCII, da extensão do arquivo ocultado.

Para exemplificar, dois arquivos de mídia com dados esteganografados foram considerados, tendo seus campos e tamanhos separados em tabelas, o pri-

meiro com a mensagem “UFF” esteganografada, representado pela Tabela 8; e segundo com um arquivo PDF de *10 bytes* esteganografado, representado na Tabela 9.

Tabela 8: Tabela representando os campos e tamanho em *bytes* ocupados em um arquivo de mídia com a mensagem UFF esteganografada.

Bytes do Arquivo de Mídia	Mensagem oculta	Rodapé adicionado - 9 Bytes		
		Tamanho do dado oculto	Tipo do dado oculto	Assinatura
X Bytes	3 Bytes	4 Bytes	1 Byte	4 Bytes
xxxxxxxxxx	UFF	3	M	TCUF

Tabela 9: Tabela representando os campos e tamanho em *bytes* ocupados em um arquivo de mídia com um arquivo PDF esteganografado.

Bytes do Arquivo de Mídia	Bytes do Arquivo PDF	Rodapé adicionado - 13 bytes			
		Extensão do arquivo oculto	Tamanho do dado oculto	Tipo do dado oculto	Assinatura
X bytes	10 bytes	4 bytes	4 bytes	1 byte	4 bytes
xxxxxxxxxx	yyyyyyyyyy	PDF	10	A	TCUF

Essa esteganografia é realizada pela classe TFinalArquivo e poderia ser aplicada a qualquer tipo de arquivo digital sem que seja necessário qualquer alteração na classe. A única restrição é que o arquivo original não se corrompa com adição de informação em seu final.

A técnica LSB e seus derivados utilizarão as mesmas informações extras que foram necessárias no método de final de arquivo, porém, cada informação contida em um *byte* do rodapé passará a ocupar 8 *bytes*, utilizando apenas o último *bit* de cada. Desta forma, o rodapé de 9 *bytes* estará distribuído nos últimos 72 *bytes* da área de dados do arquivo de mídia e o rodapé de 13 *bytes* ocupará os últimos 104 *bytes* da área de dados do arquivo. Outra diferença é que o *byte* utilizado para identificação do tipo de informação ocultada também será utilizado para identificar a variação do LSB utilizado. O *byte* será formado pelo número *n* da esteganografia LSB *n*, considerando LSB simples como *n=1*, e 8(oito) para LSB Cíclico. Esse valor será

deslocado para a esquerda e o *bit* 0(zero) – para mensagem de texto – ou 1(um) – para arquivo oculto – será adicionado. A Tabela 10 apresenta as possibilidades de valores para o *byte* e seus significados.

Tabela 10: Representação dos valores possíveis para o *byte* de identificação do tipo de esteganografia utilizada.

Técnica Esteganográfica	7 bits mais significativos	Bit menos significativo	
		Texto oculto	Arquivo oculto
LSB	0000001	0	1
LSB 2	0000010	0	1
LSB 3	0000011	0	1
LSB n; para $n=4$	0000100	0	1
LSB n; para $n=5$	0000101	0	1
LSB n; para $n=6$	0000110	0	1
LSB n; para $n=7$	0000111	0	1
LSB Cíclico	0001000	0	1

Para exemplificar a distribuição das informações pelos *bytes* do arquivo , na técnica LSB, será utilizado apenas o campo de assinatura presente em todo arquivo contendo dado ocultado por esse *software*. Utilizando a Tabela 11 que informa os valores decimais, hexadecimais e binários dos caracteres T,C,U e F; verifica-se que os últimos 32 *bytes* do conteúdo sempre seguirão o padrão:

xxxxxxxx0 xxxxxxxx1 xxxxxxxx0 xxxxx1 xxxxxxxx0 xxxxxxxx1 xxxxxxxx0 xxxxx0
 xxxxxxxx0 xxxxxxxx1 xxxxxxxx0 xxxxx0 xxxxxxxx0 xxxxxxxx0 xxxxxxxx1 xxxxx1
 xxxxxxxx0 xxxxxxxx1 xxxxxxxx0 xxxxx1 xxxxxxxx0 xxxxxxxx1 xxxxxxxx0 xxxxx1
 xxxxxxxx0 xxxxxxxx1 xxxxxxxx0 xxxxx0 xxxxxxxx0 xxxxxxxx1 xxxxxxxx1 xxxxx0
 ; onde cada bloco representa 1 *byte*, sendo x = 1 *bit* = 0 ou 1.

Tabela 11: Representação dos valores decimais, hexadecimais e binários dos caracteres T,C,U e F.

Caracter	Valor Decimal	Valor Hexadecimal	Valor Binário
T	84	54	0101 0100
C	67	43	0100 0011
U	85	55	0101 0101

F	70	46	0100 0110
---	----	----	-----------

A estruturação das classes no sistema foi projetada de forma a permitir que as técnicas LSB, LSB 2, LSB 3, LSB n e LSB Cíclico possam ser aplicadas a qualquer arquivo digital que delimita uma área de dados, bastando para isso conhecer a estrutura do arquivo e implementar o método obtemDadoMidia – que identifica a área de dados da mídia – em uma subclasse de TArquivoMidia.

Terminada a visão técnica e iniciando a visão operacional, o programa é composto por uma tela com duas guias: uma guia para realização da esteganografia e outra para recuperação da informação.

Para ocultar um dado, o usuário deverá selecionar a guia de ocultação, e o arquivo de mídia que receberá o dado esteganografado. Se o arquivo selecionado for do tipo JPG, MP3 ou AVI, apenas a opção de técnica em final do arquivo ficará disponível para realização da esteganografia. Se o arquivo selecionado for do tipo BMP ou AVI, as opções de LSB, LSB 2, LSB 3, LSB n e LSB Cíclico ficarão disponíveis, além da opção de final de arquivo.

O usuário deverá em seguida selecionar um arquivo a ser ocultado ou digitar uma mensagem. Essas opções são excludentes, ou seja, ao selecionar um arquivo, o campo de mensagem é apagado e ao digitar uma mensagem, o campo de arquivo é limpo.

Com a informação a ser escondida preenchida, bastará ao usuário escolher uma das técnicas esteganográficas disponíveis e executar a esteganografia. Será apresentada uma janela para que o usuário escolha o local e o nome para que o arquivo resultante do processo esteganográfico seja gravado. A extensão do arquivo resultante será sempre igual à extensão do arquivo de mídia original, não sendo permitido ao usuário alterá-la. Em todo momento, informações referentes ao processo como: tamanho dos arquivos, tipo dos arquivos e tamanho da mensagem; são informadas ao usuário através de exibições na tela.

O processo de recuperação da informação é simples. O usuário deverá selecionar o arquivo desejado, sendo possíveis apenas arquivos nos formatos BMP, JPG, WAV, MP3 e AVI. O sistema identificará a existência ou não de esteganografia e informará ao usuário através de exibição de informações. Se existir algum dado oculto, será habilitada a possibilidade de realizar a recuperação. Ao executar a recu-

peração, se o conteúdo for uma mensagem em texto, essa mensagem será exibida na tela, se for um arquivo, uma janela será apresentada ao usuário para que este possa salvar o arquivo no lugar desejado, não sendo permitida a alteração da extensão do arquivo.

5.6 INTERFACE DO SOFTWARE

A Figura 27 ilustra a tela inicial do *software* com a guia “Ocultar Dados” selecionada. Os principais objetos da tela foram numerados e suas funcionalidades listadas a seguir.

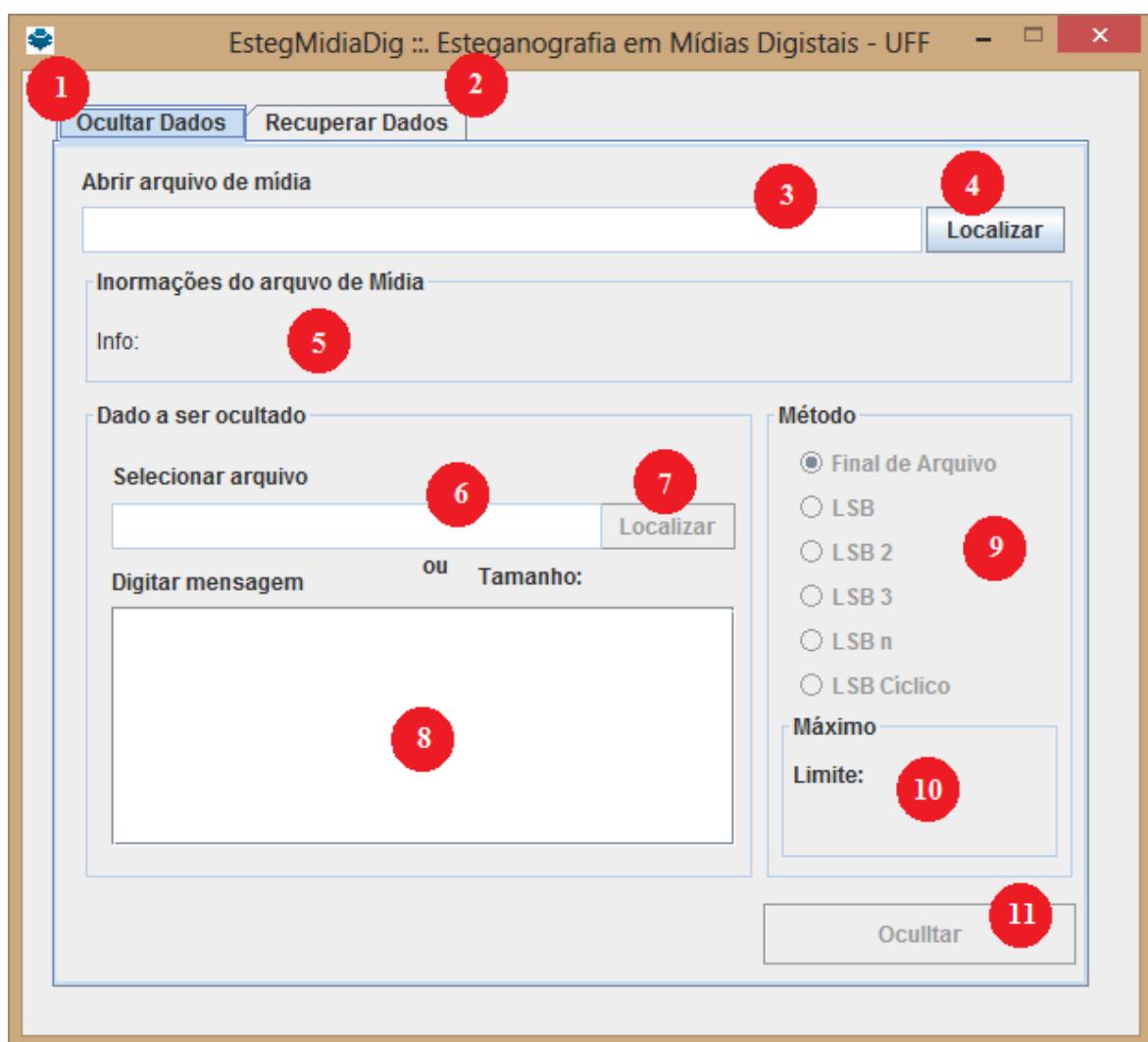


Figura 27: Tela inicial do *software* com a guia "Ocultar Dados" selecionada.

- 1 – Guia para seleção da opção de ocultação de dados.
 - 2 – Guia para seleção da opção de recuperação de dado oculto.
 - 3 – Campo de texto não editável que apresentará o local e nome do arquivo de mídia selecionado.
 - 4 – Botão que apresentará uma caixa de diálogo para seleção do arquivo de mídia a ser utilizado.
 - 5 – Área de texto onde serão apresentadas informações sobre o arquivo de mídia selecionado.
 - 6 – Campo de texto não editável que apresentará o local e nome do arquivo a ser esteganografado.
 - 7 – Botão que apresentará uma caixa de diálogo para seleção do arquivo a ser esteganografado.
 - 8 – Caixa de texto editável que permite a digitação da mensagem a ser esteganografada.
 - 9 – Opções de técnicas de esteganografia disponíveis.
 - 10 – Informação sobre o tamanho limite, do dado, possível de ser esteganografado com as opções selecionadas.
 - 11 – Botão que executa a esteganografia, considerando as informações apresentadas.
- A Figura 28 apresenta a tela recuperação de dado oculto, nela, cinco objetos foram selecionados e numerados e suas funcionalidades.
- 1 – Campo de texto não editável que apresentará o local e nome do arquivo de mídia selecionado que possivelmente contenha um dado esteganografado;
 - 2 – Botão que apresentará uma caixa de diálogo para seleção do arquivo de mídia com possível dado esteganografado.
 - 3 – Área de texto onde será informado se o arquivo contém dado oculto e em caso afirmativo, o tipo de técnica utilizada e o tipo de dado ocultado; além de outras informações sobre o arquivo de mídia selecionado.
 - 4 – Botão que será habilitado para realização da recuperação do dado oculto, quando este existir.
 - 5 – Caixa de texto, não editável, que apresentará o conteúdo recuperado, caso este seja uma mensagem de texto.

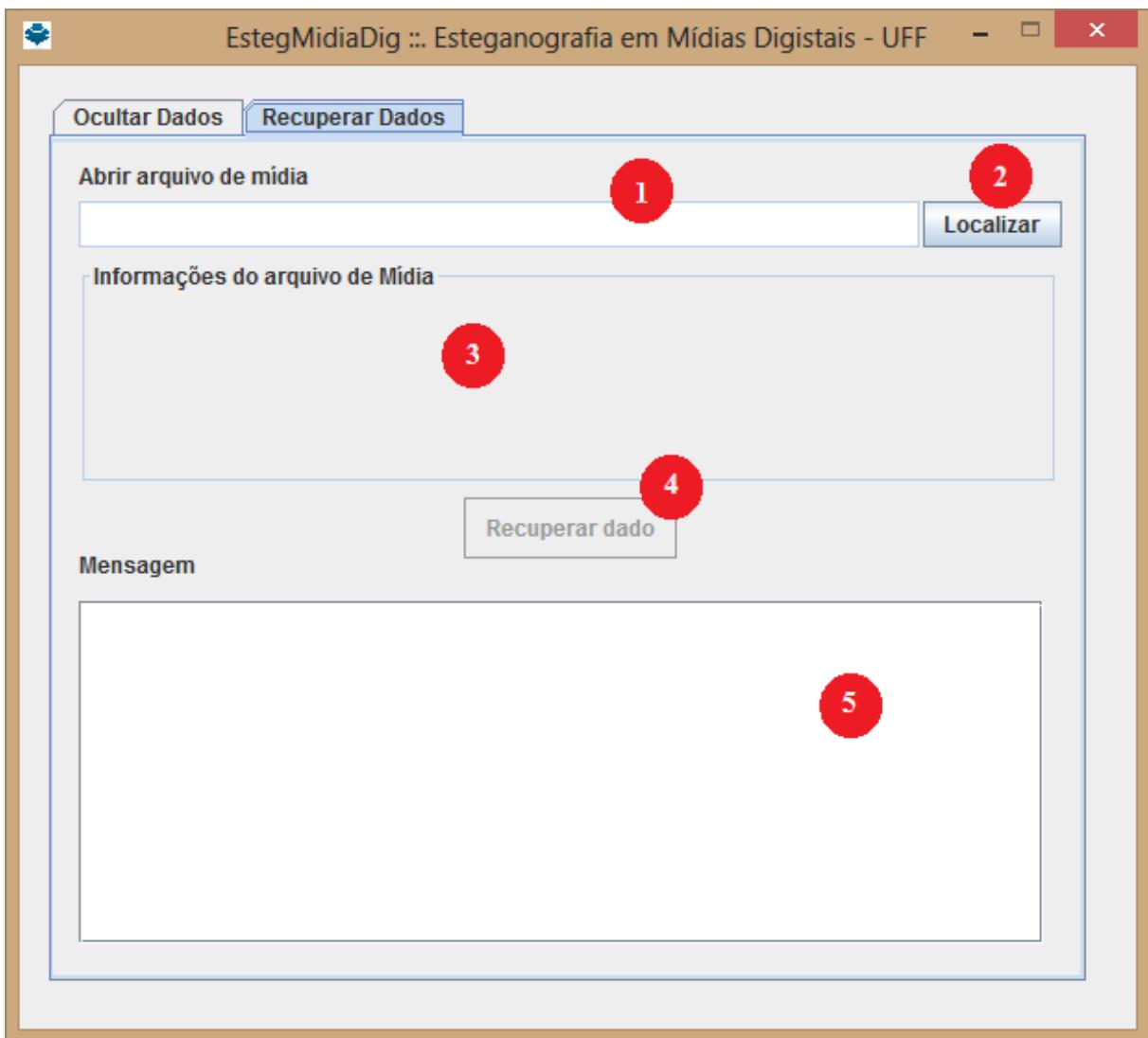


Figura 28: Tela inicial do *software* com a guia "Recuperar Dados" selecionada.

A Figura 29 exibe o *software* em processo de ocultação de mensagem em um arquivo JPG utilizando a técnica de final de arquivo. A seguir, algumas informações sobre os campos numerados.

- 1 – Caminho do arquivo de mídia selecionado.
- 2 – Extensão e tamanho do arquivo de mídia selecionado. Extensão JPG e tamanho de 1,89 Mb.
- 3 – Apenas a opção de esteganografia em final de arquivo aparece disponível, pois o arquivo de mídia tem o formato JPG.
- 4 – Tamanho da mensagem digitada. 504 bytes.
- 5 – Mensagem a ser ocultada.

6 – Tamanho máximo da mensagem permitida na esteganografia em final de arquivo. Aproximadamente 4 GB.

7 – Botão de liberado para ocultação da mensagem.

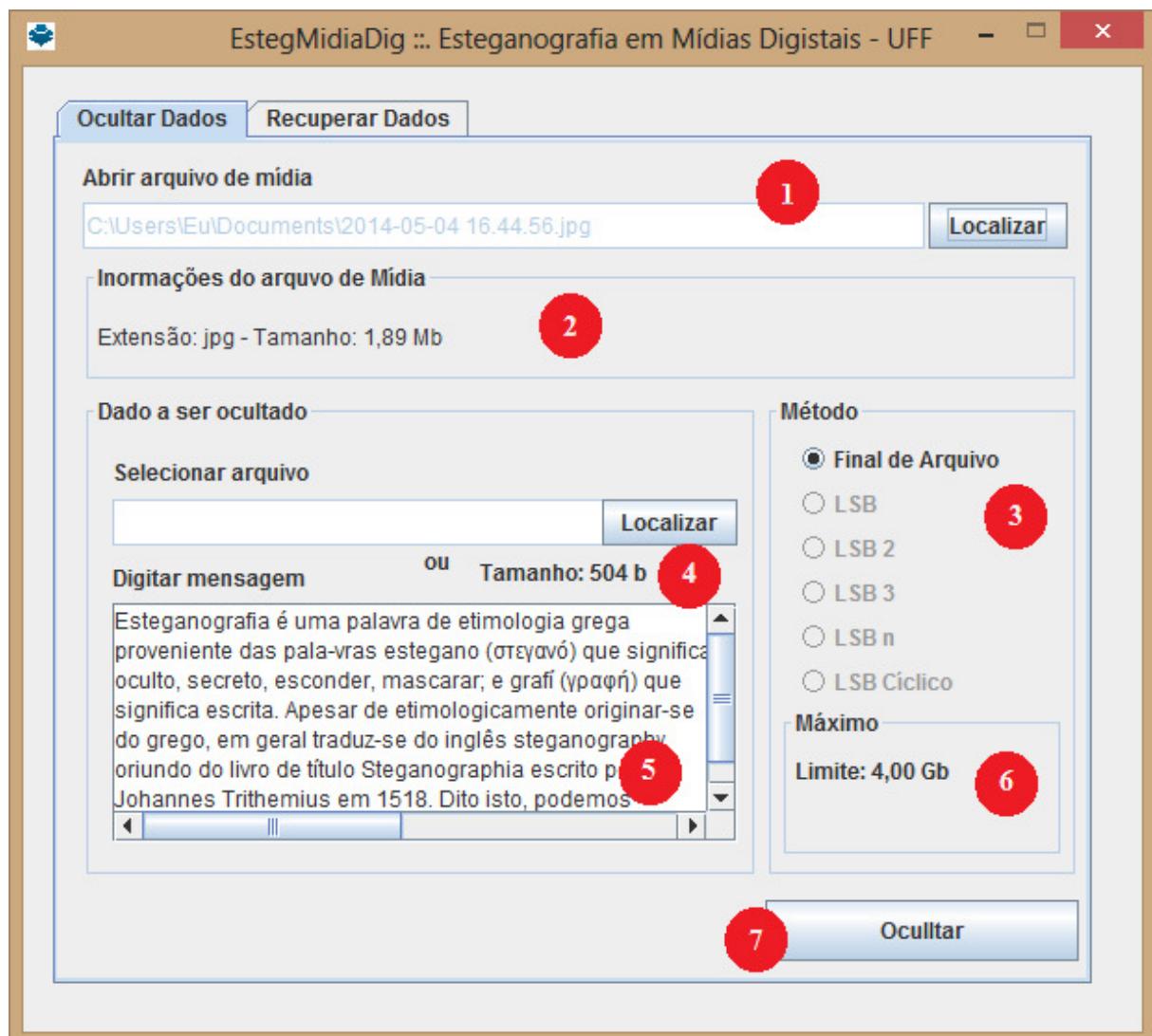


Figura 29: Tela do software em processo de ocultação.

A Figura 30 demonstra a guia "Recuperar Dados" e, processo de recuperação da informação oculta. Os campos importantes foram destacados e numerados; e suas explicações a seguir.

1 – Caminho do arquivo de mídia selecionado para recuperação da informação.

2 – Exibição das informações referentes à esteganografia encontrada no arquivo de mídia. Existe um arquivo de 1,8 MB escondido na imagem JPG que foi esteganografado com a técnica de final de arquivo.

3 – Botão de recuperação do dado disponibilizado.

4 – A caixa de texto não será preenchida, pois o dado oculto não se trata de uma mensagem de texto.

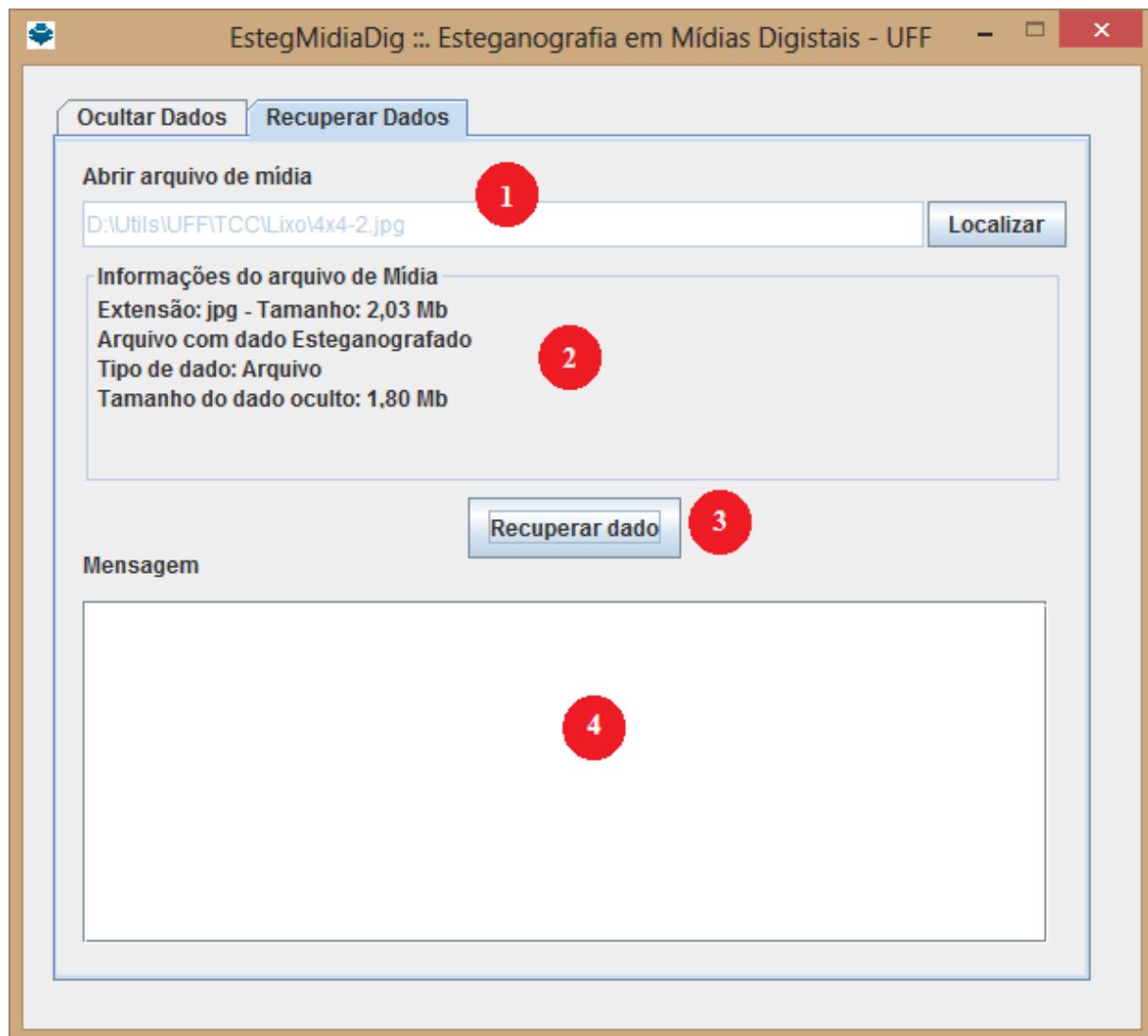


Figura 30: Tela do *software* em processo de recuperação de informação esteganografada.

A Figura 31 apresenta o final da realização de uma recuperação de dado esteganografado, onde se destacam os objetos a seguir.

1 – Caminho do arquivo de mídia transportou o dado oculto.

2 – Informações sobre a esteganografia utilizada. Trata-se de uma mensagem de texto oculta.

3 – Caixa de mensagem informando o sucesso da recuperação da mensagem esteganografada.

4 – Exibição da mensagem recuperada.

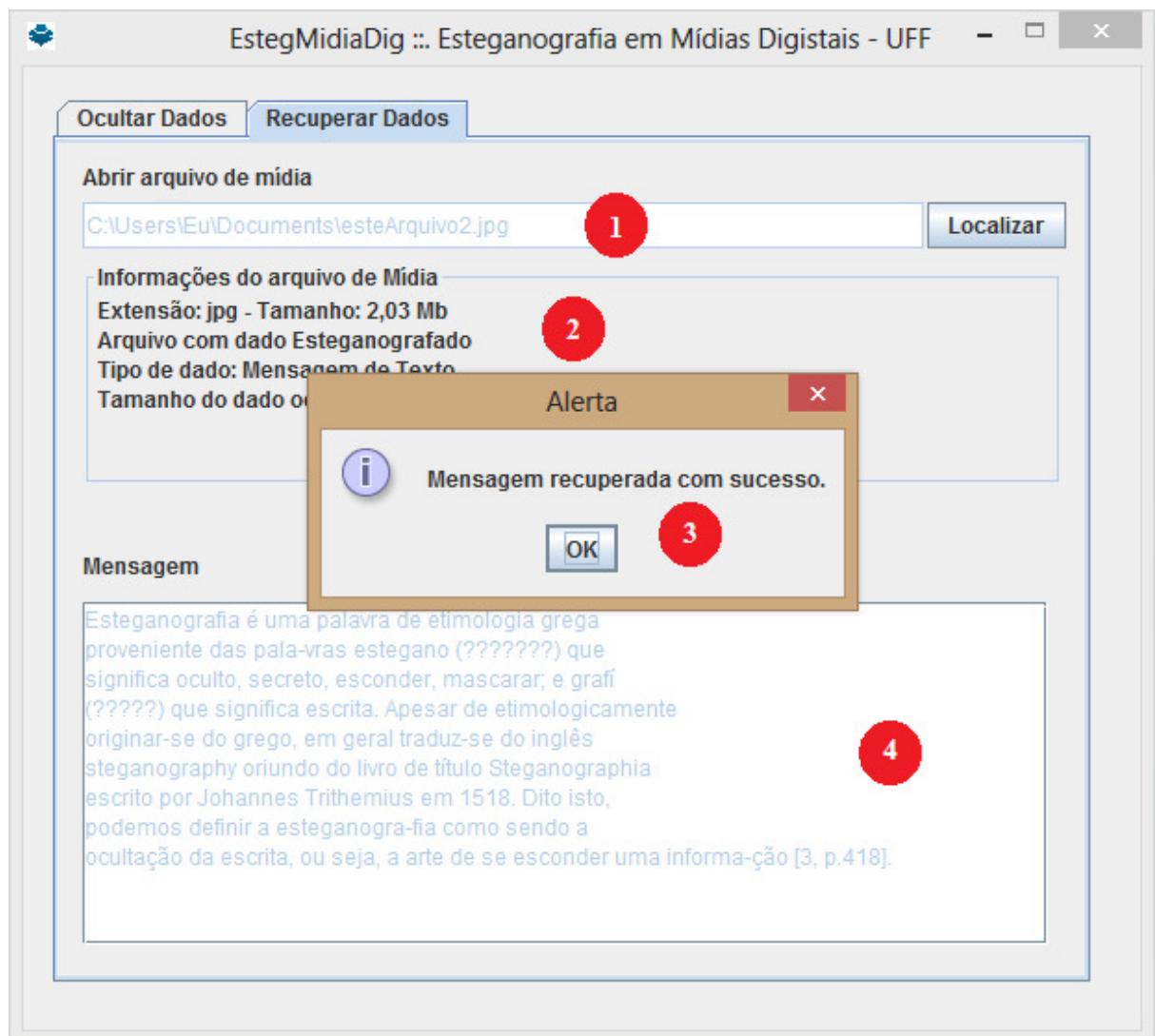


Figura 31: Tela do *software* exibindo a realização da recuperação do dado.

5.7 LIMITAÇÕES

Além das limitações referentes aos formatos de arquivos de mídia que podem ser utilizados para inclusão dos dados esteganografados o sistema contém as seguintes limitações:

- 1) A identificação do formato do arquivo é realizada sem a análise de seu conteúdo, sendo observada apenas a extensão do nome, desta forma, se uma extensão não corresponder, corretamente, ao tipo do arquivo, o *software* não perceberá essa divergência.
- 2) Embora o tamanho do conteúdo oculto em uma esteganografia de final de arquivo não tenha limite, o *software* desenvolvido limita seu tamanho em 4294967296 *bytes*, ou seja, aproximadamente 4 *Gbytes*. Esse tamanho refere-se a maior número inteiro possível de ser representado em 4 *bytes*, que é quantidade de *bytes* do campo que contém a informação referente ao tamanho do conteúdo esteganografado no rodapé que é adicionado ao arquivo final.
- 3) Na técnica LSB e suas derivações, o arquivo de mídia tem que conter no mínimo 80 *bytes* – 72 + 8 – de dados para inclusão de mensagem de texto e 112 *bytes* – 104 + 8 – de dados para inclusão de dados na forma de arquivo. Isso ocorre por causa da quantidade de dados ocupada pela informação extra contida no rodapé que será adicionado.
- 4) Além do limite de 4294967296 *bytes*, que se dá pelo mesmo motivo da limitação na técnica em final de arquivo, a quantidade máxima de informação esteganografada nas técnicas LSB depende da quantidade de *bytes* de dados disponível no arquivo de mídia. Para a técnica LSB em de uma mensagem de texto, o tamanho limite da mensagem é de $|(<\text{tamanho dos dados}>-72)/8|$. Para a ocultação de arquivo através da LSB, o tamanho limite é $|(<\text{tamanho dos dados}>-104)/8|$. Os limites para a LSB 2 são $|((<\text{tamanho dos dados}>-72)*2)/8|$ e $|((<\text{tamanho dos dados}>-104)*2)/8|$. Para a LSB 3, $|((<\text{tamanho dos dados}>-72)*3)/8|$ e $|((<\text{tamanho dos dados}>-104)*3)/8|$. Para LSB n, $|((<\text{tamanho dos dados}>-72)*n)/8|$ e $|((<\text{tamanho dos dados}>-104)*n)/8|$.

$\text{dos} > -72) * n / 8 |$ e $|((\text{tamanho dos dados} - 104) * n / 8|$. Por último, para LSB Cíclico, $|((\text{tamanho dos dados} - 72) * 7) / 8|$ e $|((\text{tamanho dos dados} - 104) * 7) / 8|$.

5.8 ALGORÍTMOS

Um dos principais algoritmos é o que realiza a concatenação de *arrays* de *bytes*. Ele cria um *array* de *bytes* com o tamanho das duas informações a serem concatenadas e realiza a cópia dos dados, sendo o primeiro copiado para a posição 0 (zero) e o segundo para o final da primeira cadeia de *bytes* copiados. A Figura 32 exibe a codificação desse algoritmo na linguagem Java.

```

18 public static byte[] byteArrayConcat (byte[] array1, byte[] array2)
19 {
20     //Novo array com tamanho total dos 2 array passados como parametro
21     byte[] arrayResult = new byte[array1.length + array2.length];
22
23     //Copiando o primeiro array para o novo a partir da posição 0
24     System.arraycopy(array1, 0, arrayResult, 0, array1.length);
25     //Copiando o segundo array para o novo a partir da posição array1.length
26     System.arraycopy(array2, 0, arrayResult, array1.length, array2.length);
27
28     return arrayResult;
29 }
```

Figura 32: Código em Java do procedimento utilizado para concatenar dois *arrays* de *bytes*.

A esteganografia em final de arquivo de uma mensagem é realizada convertendo a mensagem em uma *array* de *bytes*. Em seguida, é realizada a concatenação do conteúdo, em *bytes*, do arquivo de mídia – que servirá como container – e a mensagem. Ao resultado obtido, são adicionadas as informações do rodapé. A Figura 33 demonstra a codificação desse algoritmo em linguagem Java.

O algoritmo que realiza a ocultação de um arquivo utilizando a esteganografia em final de arquivo segue a mesma lógica do algoritmo de mensagem em final de arquivo, apenas substituindo o *array* de *bytes* da mensagem pelo *array* de *bytes* com do conteúdo do arquivo a ser ocultado. A Figura 34 apresenta essa codificação em linguagem Java.

```

108 public TArquivo ocultaDado (String msg) //Oculta mensagem Txt no arquivo
109 {
110     byte[] msgByte = msg.getBytes();
111     byte[] novoConteudo = TArquivo.byteArrayConcat(VCContainer.getConteudo(),
112                                                 msgByte);
113     novoConteudo = TArquivo.byteArrayConcat(novoConteudo,
114                                             rodapeTxt(msgByte.length));
115
116     VCContainer.setConteudo(novoConteudo);
117     return VCContainer;
118 }

```

Figura 33: Código em Java do procedimento utilizado para ocultar mensagem em final de arquivo.

```

120 public TArquivo ocultaDado (TArquivo arquivo) //Oculto arquivo no arquivo
121 {
122     byte[] novoConteudo = TArquivo.byteArrayConcat(VCContainer.getConteudo(),
123                                                 arquivo.getConteudo());
124
125     novoConteudo = TArquivo.byteArrayConcat(novoConteudo,
126                                             rodapeArq(arquivo.getTamanho(),
127                                                       arquivo.getExtensao()));
128
129     VCContainer.setConteudo(novoConteudo);
130     return VCContainer;
131 }

```

Figura 34: Código em Java do procedimento utilizado para ocultar arquivo em final de arquivo.

Para recuperar a mensagem de texto esteganografada em um arquivo de mídia, o rodapé da esteganografia é lido e identificado a posição de início e de fim da mensagem dentro do arquivo hospedeiro. De posse dessas informações, a cópia dos *bytes* contidos neste intervalo é realizada e convertida para texto. A Figura 35 apresenta a codificação deste algoritmo em linguagem Java.

```

133 public String recuperaDadoTexto () //Recupera mensagem Txt do arquivo
134 {
135     byte[] msgByte = new byte[(int)VPTamEsteg];
136     byte[] b = VCContainer.getConteudo();
137     System.arraycopy(b, (int)(b.length - VPTamEsteg - 9), msgByte,
138                      0, (int)VPTamEsteg);
139
140     return new String(msgByte);
141 }

```

Figura 35: Código em Java do procedimento utilizado para recuperar mensagem em final de arquivo.

A lógica do algoritmo para recuperação do arquivo ocultado em final de arquivo é a mesma utilizada na recuperação de mensagem ocultada em final de arquivo. A diferença no algoritmo é que o *array* de *bytes* obtido é convertido em um

arquivo que deve ser gravado em disco. A Figura 36 mostra a codificação deste algoritmo em linguagem Java.

```

143 public TArquivo recuperaDadoArquivo () //Recupera arquivo do arquivo
144 {
145     byte[] arqByte = new byte[(int)VPTamEsteg]; //Tamanho
146     byte[] b = VPContainer.getConteudo();           //Conteúdo oculto
147
148     System.arraycopy(b, (int)(b.length - VPTamEsteg - 13), arqByte,
149                      0, (int)VPTamEsteg);
150
151     TArquivo arOculto = new TArquivo();
152     arOculto.setConteudo(arqByte);
153     arOculto.setExtensao (VPExtensaoEsteg);
154     return arOculto;
155 }
```

Figura 36: Código em Java do procedimento utilizado para recuperar arquivo em final de arquivo.

Para todas as variações da técnica de esteganografia LSB são fundamentais dois algoritmos: um que altera qualquer *bit* dentro de um *array* de *bytes* e um que recupere qualquer *bit* dentro de um *array* de *bytes*. O funcionamento desses procedimentos, por sua vez, necessita apenas de duas informações. A posição do *byte* dentro do vetor e a posição do *bit* desejado dentro do *byte*. A posição do *byte* é referenciada diretamente através do índice do vetor, já o *bit* desejado depende do procedimento em execução, se é uma alteração ou recuperação de seu valor.

Na alteração do *bit*, Figura 37, se o desejo for alterar seu valor para 0 (zero), rotaciona-se *n* vezes para esquerda o número 127 – 01111111 em binário –; onde *n* é a posição – iniciando da direita para esquerda – do *bit* desejado e realiza-se um “AND” lógico do valor obtido com a rotação e o valor do *byte* selecionado. Se alteração do *bit* for para 1, o valor 128 – 10000000 em binário – é rotacionado *n* vezes para esquerda; sendo *n* a posição do *bit* e realiza-se um “OR” lógico entre o valor obtido com a rotação e o *byte* selecionado.

Para exemplificar a alteração do *bit*, podemos considerar o número 91 e realizar a alteração do segundo *bit* para 0. Neste caso, o número 127 seria rotacionado duas vezes para a esquerda, resultando no número binário 11111101, realizando um “AND” com o número 91 – 01011011 em binário – obteríamos como resultado o número binário 01011001, ou seja, o número 89.

O método alteraBit da Figura 37 demonstra a codificação desse algoritmo em linguagem Java.

Para recuperar o valor do *bit*, basta realizar a rotação para esquerda do valor 128 – 10000000 em binário – *n* vezes; onde *n* é a posição – iniciando da direita para esquerda – do *bit* desejado e realiza-se um “AND” lógico do valor obtido com a rotação e o valor do *byte* selecionado. Se o valor resultante for 0, o valor do *bit* é 0, em caso contrário, o valor do *bit* é 1.

Para exemplificar esse algoritmo, podemos considerar o número 91 e recuperar segundo *bit*. Neste caso, o número 128 seria rotacionado duas vezes para a esquerda, resultando no número binário 00000010. Realizando um “AND” com o número 91, em binário 01011011, obteríamos como resultado o número binário 00000010, ou 2 em decimal; logo, o segundo *bit* do número 91 é 1.

O método getBit da Figura 37 demonstra a codificação desse algoritmo em linguagem Java.

```

89 protected void alteraBit(boolean val)
90 {
91     int valByte;
92     if (val) valByte = 0b10000000;
93     else valByte = 0b01111111;
94
95     valByte = byteRotateLeft(valByte, VPPosBit);
96
97     if (val) VPConteudo[(int)VPPosByte] = (byte) (VPConteudo[(int)VPPosByte] |
98                                         valByte);
99
100    else VPConteudo[(int)VPPosByte] = (byte) (VPConteudo[(int)VPPosByte] &
101                                         valByte);
102 }
103
104 protected boolean getBit()
105 {
106     int valByte = 0b10000000;
107     valByte = byteRotateLeft(valByte, VPPosBit);
108     int val = 0xff & VPConteudo[(int)VPPosByte];
109     if ((val & valByte)==0) return false;
110     else return true;
111 }
```

Figura 37: Código em Java dos procedimentos utilizados para recuperar (getBit) e alterar (alteraBit) um *bit* qualquer em uma cadeia de *bytes*. O valor 1 é representado por *true* e 0 por *false*.

O algoritmo para ocultar dados utilizando a técnica LSB considera que os dados a serem escondidos, sejam eles textos ou arquivos, já tenham sido convertidos para *array* de *bytes*. A posição do *bit* a ser alterado no arquivo hospedeiro é

mantido como 1(um) durante todo o processo de esteganografia. A inclusão da informação é realizada percorrendo todos os *bytes* do dado através de um *loop*. Um *loop*, também, é utilizado para percorrer cada *bit* do *byte* atual e gerar um índice. O *bit* desejado é obtido pelo deslocamento para esquerda *n* vezes do valor do *byte* desejado; onde *n* é o índice obtido pelo segundo *loop*, e realizado um “AND” lógico com o valor 128 – 10000000 em binário –. Se o valor obtido for 0, o *bit* é zero, em caso contrário, o valor do *bit* é 1. Esse valor é passado para o procedimento de alteração de *bit* e a posição do *byte* no arquivo é incrementada (Figura 38).

```

35 private TArquivoMidia ocultaBytes (byte[] dado)
36 {
37     zeraPosByte();
38     setPosBit(1);
39
40     for (int i=0; i<dado.length; i++)
41     {
42         for (int j=0; j<8; j++)
43         {
44             //0b10000000 para pegar o bit mais significativo
45             int val = ((byte)(dado[i]<<j)) & ((byte)0b10000000);
46             alteraBit(val!=0);
47             incPosByte();
48         }
49     }
50     return altDadoMidia();
51 }
```

Figura 38: Código em Java do procedimento utilizado para ocultar *bytes* com a técnica LSB.

O processo de recuperação da informação esteganografada obtém um *array* de *bytes* que poderá ser convertido para texto ou para arquivo, dependendo da informação oculta. O tamanho do vetor é recuperado do rodapé que identifica a esteganografia. Um *loop* é realizado de zero até o tamanho do dado oculto. Esse *loop* representa cada *byte* da informação esteganografada. Para cada laço, um segundo loop representando os *bits* do *byte* é realizado; logo, esse segundo *loop* contém 8 laços e realiza a recuperação de um *bit* esteganografado e a posição do *byte* do arquivo hospedeiro é incrementado. O *bit* obtido é adicionado ao do *array* de *byte* que vai sendo criado. Para isso, o valor que está sendo criado é deslocado um *bit* para esquerda a cada interação. A codificação deste algoritmo em linguagem Java é demonstrada na Figura 39.

```

53  private byte[] recuperaBytes ()
54  {
55      long tamMsg = tamDadoOculto();
56      byte[] b = new byte[(int)tamMsg];
57
58      zeraPosByte();
59      setPosBit(1);
60
61      for(int i=0; i<tamMsg; i++)
62      {
63          b[i] = 0;
64          for(int j=0; j<8; j++)
65          {
66              boolean bit = getBit();
67              b[i] = (byte)(b[i]<<1);
68              if(bit) b[i] = (byte)(b[i] | 1);
69              else   b[i] = (byte)(b[i] | 0);
70
71              incPosByte();
72          }
73      }
74
75      return b;
76  }

```

Figura 39: Código em Java do procedimento utilizado para recuperar *bytes* com a técnica LSB.

O algoritmo de esteganografia LSB Cílico, tanto para ocultação da informação quanto para recuperação, segue a mesma lógica utilizada no algoritmo LSB, com uma diferença. No algoritmo LSB ao incrementar a posição do *byte* no arquivo hospedeiro o valor nunca ultrapassa o tamanho da área de dado disponível, já no algoritmo LSB Cílico esse valor poderá ser ultrapassado. Porém, quando isso acontece, a posição do *byte* no arquivo é reiniciada e a posição do *bit* é incrementada. A Figura 40 codifica o algoritmo de ocultação de dados em Java, enquanto a Figura 41 codifica o algoritmo de recuperação da informação.

Na codificação, a função *incPosByte* é responsável por incrementar a posição do *byte* no arquivo e reiniciar seu valor quando o mesmo ultrapassar o limite permitido. Essa função também retorna a posição do *byte* após ser incrementado, desta forma basta testar o valor retornado para saber se o ciclo foi completado. Em caso afirmativo, utiliza-se o método *incPosBit* para incrementar a posição do *bit* dentro do *byte*.

```

36  private TArquivoMidia ocultaBytes (byte[] dado)
37  {
38      zeraPosByte();
39      setPosBit(1);
40
41      for (int i=0; i<msgByte.length; i++)
42      {
43          for (int j=0; j<8; j++)
44          {
45              //0b10000000 para pegar o bit mais significativo
46              int val = ((byte)(msgByte[i]<<j)) & ((byte)0b10000000);
47              alteraBit(val!=0);
48              long proxByte = incPosByte();
49              if(proxByte==0) incPosBit();
50          }
51      }
52
53      return altDadoMidia();
54  }

```

Figura 40: Código em Java do procedimento utilizado para ocultar *bytes* com a técnica LSB Cíclico.

```

56  private byte[] recuperaBytes ()
57  {
58      long tamMsg = tamDadoOculto();
59      byte[] b = new byte[(int)tamMsg];
60
61      zeraPosByte();
62      setPosBit(1);
63
64      for(int i=0; i<tamMsg; i++)
65      {
66          b[i] = 0;
67          for(int j=0; j<8; j++)
68          {
69              boolean bit = getBit();
70              b[i] = (byte)(b[i]<<1);
71              if(bit) b[i] = (byte)(b[i] | 1);
72              else    b[i] = (byte)(b[i] | 0);
73
74              long proxByte = incPosByte();
75              if(proxByte==0) incPosBit();
76          }
77      }
78
79      return b;
80  }

```

Figura 41: Código em Java do procedimento utilizado para recuperar *bytes* com a técnica LSB Cíclico.

Os algoritmos de ocultação e recuperação de dados LSBo seguem as mesmas lógicas utilizadas nos demais algoritmos da variação LSB. A diferença desse algoritmo ocorre ao se incrementar as posições do *byte* e do *bit* desejados no arquivo hospedeiro. Neste algoritmo um número *n* também é predefinido com um valor

entre 1 e 7 que identifica a quantidade de *bits*, de cada *byte* de dado do arquivo hospedeiro, possíveis de serem utilizados na esteganografia.

Nas outras técnicas LSB, a primeira posição a ser incrementada é a posição do *byte*, podendo a posição do *bit* ser incrementada ou não dependendo das variantes obtidas. No LSB n a cada interação do laço a posição do *bit* é incrementada. Quando a posição do *bit* ultrapassa o valor *n*, predefinido, sua posição é reiniciada e a posição do *byte* é incrementada. A Figura 42 exibe o algoritmo LSB n para ocultação de dados codificado em Java.

```

48 private TArquivoMidia ocultaBytes (byte[] dado)
49 {
50     zeraPosByte();
51     setPosBit(1);
52
53     for (int i=0; i<dado.length; i++)
54     {
55         for (int j=0; j<8; j++)
56         {
57             //Ob10000000 para pegar o bit mais significativo
58             int val = ((byte)(dado[i]<<j)) & (byte)0b10000000;
59             alteraBit(val!=0);
60
61             if (getPosBit()>=VPN)
62             {
63                 setPosBit(1);
64                 incPosByte();
65             }
66             else
67             {
68                 incPosBit();
69             }
70         }
71     }
72
73     return altDadoMidia();
74 }
```

Figura 42: Código em Java do procedimento utilizado para ocultar *bytes* com a técnica LSB n.

A Figura 43 demonstra o algoritmo para recuperação de dados LSB n codificado em Java.

Os algoritmos LSB 2 e LSB 3 são os mesmos algoritmos LSB n com o valor de *n* sendo respectivamente 2 e 3.

```

76 private byte[] recuperaBytes ()
77 {
78     long tamMsg = tamDadoOculto();
79     byte[] b = new byte[(int)tamMsg];
80
81     zeraPosByte();
82     setPosBit(1);
83
84     for(int i=0; i<tamMsg; i++)
85     {
86         b[i] = 0;
87         for(int j=0; j<8; j++)
88         {
89             boolean bit = getBit();
90             b[i] = (byte)(b[i]<<1);
91             if(bit) b[i] = (byte)(b[i] | 1);
92             else   b[i] = (byte)(b[i] | 0);
93
94             if (getPosBit()>=VPN)
95             {
96                 setPosBit(1);
97                 incPosByte();
98             }
99             else
100             {
101                 incPosBit();
102             }
103         }
104     }
105
106     return b;
107 }
```

Figura 43: Código em Java do procedimento utilizado para recuperação *bytes* com a técnica LSB n.

O código fonte completo do *software* desenvolvido pode ser obtido no endereço <https://db.tt/BJ410tyu>.

6 RESULTADOS DA IMPLEMENTAÇÃO

Neste capítulo, serão apresentados uma variedade de experimentos realizados com o *software* desenvolvido afim de avaliar, na prática, seu comportamento e de alguns algoritmos estudados neste trabalho, em específico as técnicas de final de arquivo, LSB, LSB 2, LSB 3, LSB n e LSB Cíclico, permitindo assim, uma analise dos dados obtidos.

Para possibilitar comparação, os testes foram efetuados com arquivos de mídia, para hospedagem da esteganografia, pré-fixados: conforme se segue:

- 1) Um arquivo no formato BMP 24 *bits True Color* de 215 KB;
- 2) Um arquivo no formato BMP 24 *bits True Color* de 1.390 KB;
- 3) Um arquivo no formato JPG com tamanho de 35 KB;
- 4) Um arquivo no formato WAV de 218 KB;
- 5) Um arquivo no formato MP3 de 7 KB;
- 6) Um arquivo AVI de 40.029 KB.

Apesar do *software* desenvolvido poder realizar esteganografia tanto de mensagem de texto quanto arquivo, apenas esteganografia realizando a ocultação de arquivo foram considerados nos testes por apresentar de forma fácil um volume maior de informação e padronização. Os resultados dos testes realizados são apresentados nos experimentos na seção 6.1.

6.1 EXPERIMENTO 01

Foram utilizados todos os arquivos de mídia especificados.

6.1.1 CENÁRIO

Esteganografia em final de arquivo embutindo em cada arquivo de mídia um arquivo MP4 de 1.842 KB.

6.1.2 RESULTADOS

Os arquivos obtidos foram:

- 1) Um arquivo no formato BMP 24*bits True Color* de 2.056 KB.
- 2) Um arquivo no formato BMP 24 *bits True Color* de 3.231 KB.
- 3) Um arquivo no formato JPG com tamanho de 1.876 KB.
- 4) Um arquivo no formato WAV de 2.059 KB.
- 5) Um arquivo no formato MP3 de 1.848 KB.
- 6) Um arquivo AVI de 41.870 KB.

Em nenhum dos casos houve alteração funcional do arquivo, permanecendo todos com a mesma qualidade dos arquivos originais, porém, todos os arquivos tiveram seu tamanho aumentado em 1.885.244 *bytes* referentes à soma de 1.885.231 *bytes* do tamanho do arquivo oculto e 13 *bytes* referentes ao tamanho do rodapé adicionado.

6.1.3 ANÁLISE DOS RESULTADOS

A esteganografia em final de arquivo mostrou-se uma técnica eficaz, permitindo a inclusão da informação sem nenhuma alteração funcional do conteúdo original, porém, em caso de ocultação de informação muito grande em relação ao arquivo de mídia original, o resultado pode ser comprometido, pois o arquivo final obtido terá um tamanho significativamente maior do que o original. Em caso de oculta-

ção de informação relativamente pequena, se comparada ao tamanho do arquivo original, como o caso do esteganografia realizada utilizando o arquivo AVI, a técnica apresenta resultados satisfatórios.

6.2 EXPERIMENTO 02

Foram utilizados como hospedeiros os dois arquivos de mídia com formato BMP e o arquivo no formato WAV.

6.2.1 CENÁRIO

Esteganografia LSB, adicionado ao arquivo BMP de 1.390 KB um arquivo PDF de 60 KB; ao arquivo WAV foi adicionado um arquivo JPG de 10 KB; e ao arquivo BMP de 215 KB foi adicionado um arquivo JPG de 10 KB.

6.2.2 RESULTADOS

Em nenhum dos casos houve alteração no tamanho do arquivo original. A alteração nas cores do arquivo BMP de 1.390 KB não acarretou em mudanças visuais perceptivas em relação a imagem original. Essa constatação pode ser realizada comparando se a Figura 44 que ilustra a imagem original com a Figura 45 que contém a imagem com o arquivo PDF de 60 KB esteganografado.

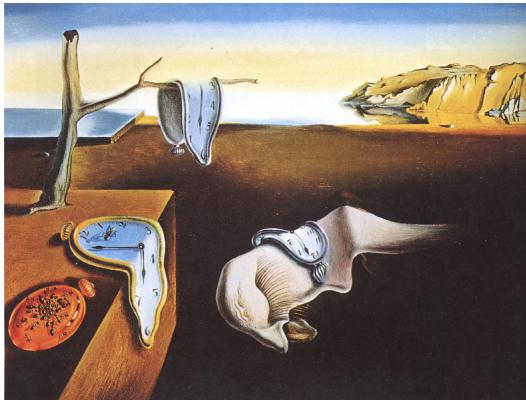


Figura 44: Arquivo BMP de 1.390 KB original.

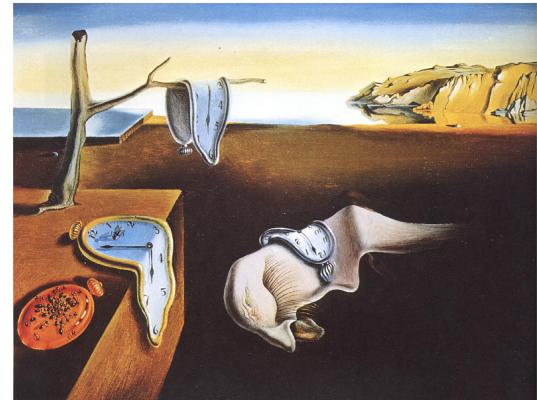


Figura 45: Arquivo BMP de 1.390 KB com um arquivo PDF de 60 KB esteganografado.

A esteganografia LSB realizada no arquivo WAV alterou de forma perceptível o áudio existente no arquivo original. Analisando a representação gráfica das ondas sonoras ilustradas pela Figura 46, que exibe o áudio original e a Figura 47 que exibe o áudio esteganografado, é possível perceber a existência da esteganografia e a seta em vermelho aponta a posição final do conteúdo esteganografado. O arquivo continua funcional, porém é possível escutar de forma clara ruídos durante a reprodução do áudio na parte esteganografada.

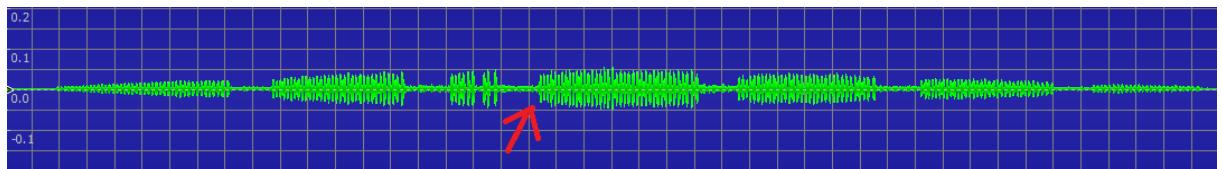


Figura 46: Representação gráfica das ondas de áudio do arquivo WAV original.

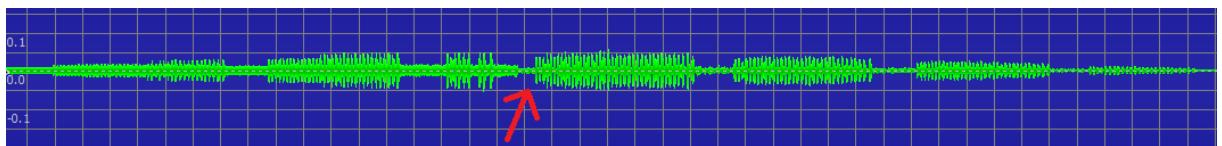


Figura 47: Representação gráfica das ondas de áudio do arquivo WAV contendo um arquivo JPG de 10 KB esteganografado.

A inclusão do arquivo JPG de 10 KB na imagem BMP também não causou nenhuma alteração visualmente perceptível na imagem original. A comparação pode ser realizada observando a Figura 48 que exibe a imagem original com a Figura 49 que exibe a imagem contendo a esteganografia.



Figura 48: Arquivo BMP de 215 KB original.



Figura 49: Arquivo BMP de 215 KB com um arquivo JPG de 10 KB esteganografado.

6.2.3 ANÁLISE DOS RESULTADOS

A técnica LSB apresentou um resultado satisfatório para esteganografia em imagens BMP. Além de não alterar o tamanho do arquivo, criou uma imagem que não pode ser distinguida visualmente da imagem original. O único problema encontrado foi a pequena capacidade de armazenamento. Já em relação ao arquivo WAV, a técnica criou ruídos perceptíveis no áudio original. O áudio continua comprehensível, fazendo com que a técnica não seja descartada, porém provavelmente acrescente limitações a sua utilização.

6.3 EXPERIMENTO 03

Foram utilizados como hospedeiros os dois arquivos de mídia com formato BMP e o arquivo no formato WAV.

6.3.1 CENÁRIO

Esteganografia LSB 2, adicionado ao arquivo BMP de 1.390 KB um arquivo JPG de 200 KB; ao arquivo WAV foi adicionado um arquivo JPG de 48 KB; e ao arquivo BMP de 215 KB foi adicionado um arquivo JPG de 48 KB.

6.3.2 RESULTADOS

Em nenhum dos casos houve alteração no tamanho do arquivo original. A alteração nas cores do arquivo BMP de 1.390 KB não acarretou em mudanças visuais perceptivas em relação a imagem original. Essa constatação pode ser realizada comparando se a Figura 50 que ilustra a imagem original com a Figura 51 que contém a imagem com o arquivo JPG de 200 KB esteganografado.



Figura 50: Arquivo BMP de 1.390 KB original.

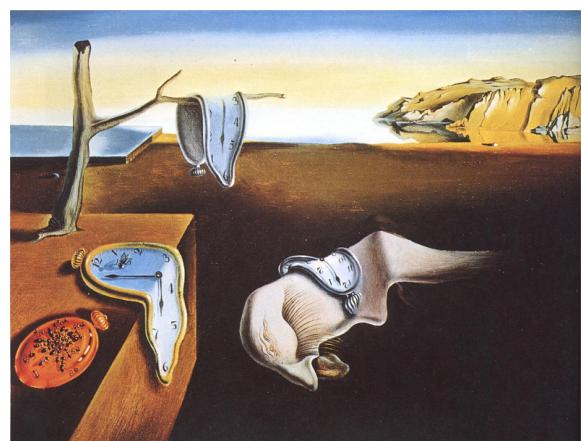


Figura 51: Arquivo BMP de 1.390 KB com um arquivo JPG de 200 KB esteganografado.

A esteganografia LSB 2 realizada no arquivo WAV alterou o áudio existente no arquivo original de forma a deixá-lo completamente distorcido. Analisando a representação gráfica das ondas sonoras ilustradas pela Figura 52, que exibe o áudio original e a Figura 53 que exibe o áudio esteganografado, é possível perceber a completa modificação das ondas sonoras que passou a conter apenas pequenas frações do áudio original.

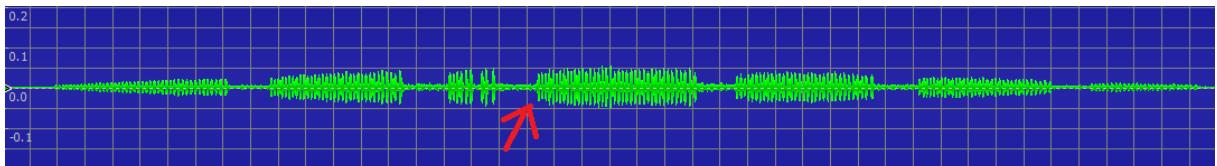


Figura 52: Representação gráfica das ondas de áudio do arquivo WAV original.

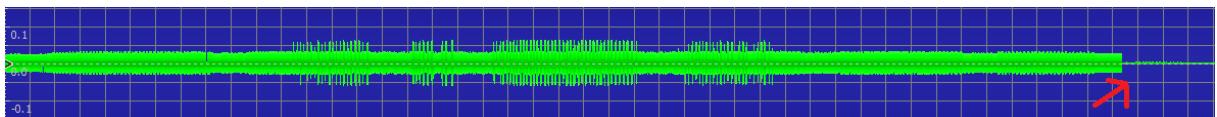


Figura 53: Representação gráfica das ondas de áudio do arquivo WAV contendo um arquivo JPG de 48 KB esteganografado.

A inclusão do arquivo JPG de 48 KB na imagem BMP também não causou nenhuma alteração visualmente perceptível na imagem original. A comparação pode ser realizada observando a Figura 54 que exibe a imagem original com a Figura 55 que exibe a imagem contendo a esteganografia.



Figura 54: Arquivo BMP de 215 KB original.



Figura 55: Arquivo BMP de 215 KB com um arquivo JPG de 48 KB esteganografado.

6.3.3 ANALISE DOS RESULTADOS

A técnica LSB 2, em imagens BMP apresentou um resultado semelhante ao da técnica LSB, ou seja, nenhuma alteração sensível a visão humana é apresentada. Sua utilização, ainda aumentou a capacidade de armazenamento da imagem; no entanto, a quantidade de ruídos acrescentado no arquivo WAV impossibilitando a compreensão do som original. Desta forma, utilizar a técnica LSB 2 e variações que utilizam maior quantidade de *bits* por *byte* é inviável quando o arquivo de transporte for do formato WAV.

6.4 EXPERIMENTO 04

Foram utilizados como hospedeiros os dois arquivos de mídia com formato BMP e o arquivo no formato WAV.

6.4.1 CENÁRIO

Esteganografia LSB3 e LSB *n*; para *n* entre 3 e 6. Na técnica LSB 3 foi adicionado ao arquivo BMP de 1.390 KB um arquivo WAV de 217 KB e ao arquivo BMP de 215 KB foi adicionado um arquivo PDF de 61 KB. Nas técnicas LSB *n*; para *n* de 4 a 6, foram adicionados ao arquivo BMP de 1.390 KB um arquivo JPG de 643 KB e ao arquivo BMP de 215 KB foi adicionado um arquivo PDF de 61 KB.

6.4.2 RESULTADOS

Em nenhum dos casos houve alteração no tamanho do arquivo original. A alteração nas cores dos arquivos BMP de 1.390 KB não sofreram alterações perceptíveis nas técnicas LSB 3, LSB n ; para $n = 4$ e $n = 5$. No BMP de 215 KB a esteganografia apresenta-se imperceptível em LSB 3 e LSB n ; para $n = 4$.

A esteganografia LSB n , para $n = 6$, no arquivo BMP de 1.390 KB; assim como a esteganografia LSB n , para $n = 5$, no arquivo BMP de 215 KB, apresentam pequenas alterações na visualização das imagens. As outras técnicas que completaram os testes apresentaram alterações significativas nas imagens, comprometendo a visualização.

Para visualização dos resultados obtidos, as Figuras 56, 58, 60, 62 e 64 mostram respectivamente o arquivo BMP de 215 KB em seu estado original e esteganografado com as técnicas LSB 3 e LSB n ; para $n = 4, 5$ e 6 . O arquivo BMP original de 1.390 KB e o resultado das técnicas esteganográficas LSB 3 e LSB n ; para $n = 4, 5$ e 6 podem ser visualizadas respectivamente nas Figuras 57, 59, 61, 63 e 65.



Figura 56: Arquivo BMP de 215 KB original.

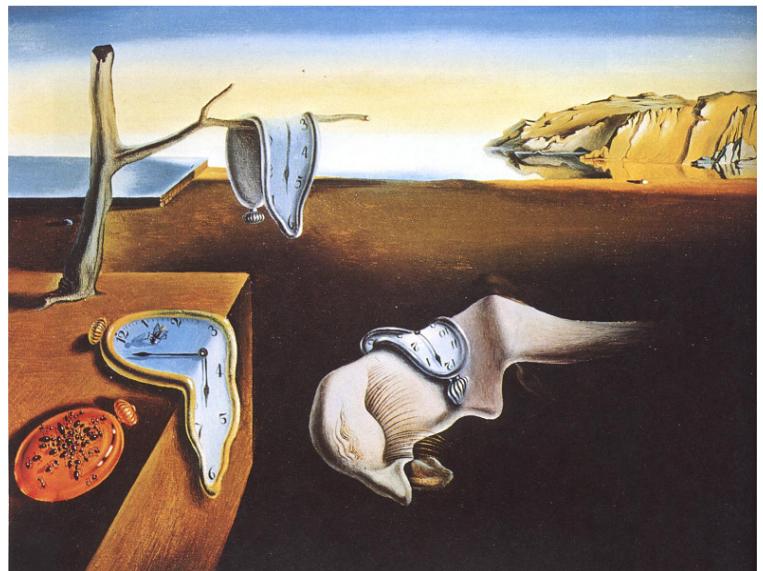


Figura 57: Arquivo BMP de 1.390 KB original.



Figura 58: Arquivo BMP de 215 KB com um arquivo PDF de 61 KB esteganografado com a técnica LSB 3.

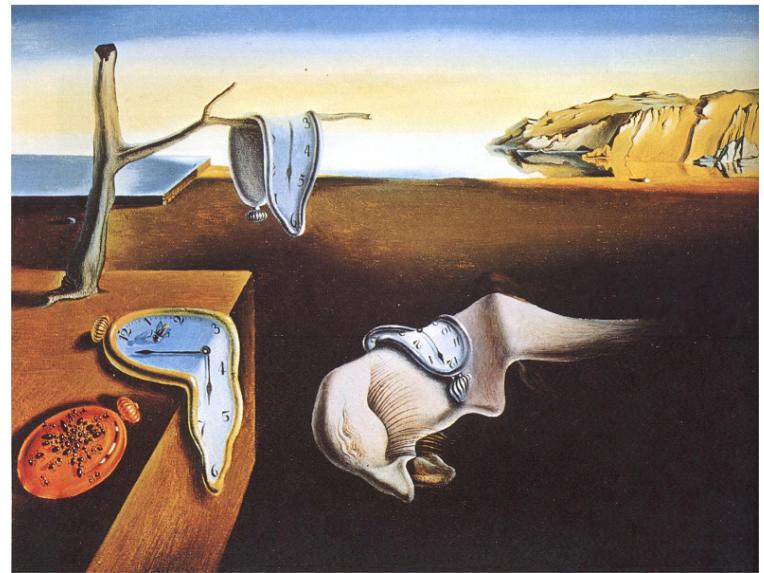


Figura 59: Arquivo BMP de 1.390 KB com um arquivo WAV de 217 KB esteganografado com a técnica LSB 3.



Figura 60: Arquivo BMP de 215 KB com um arquivo PDF de 61 KB esteganografado com a técnica LSB n; para $n = 4$.

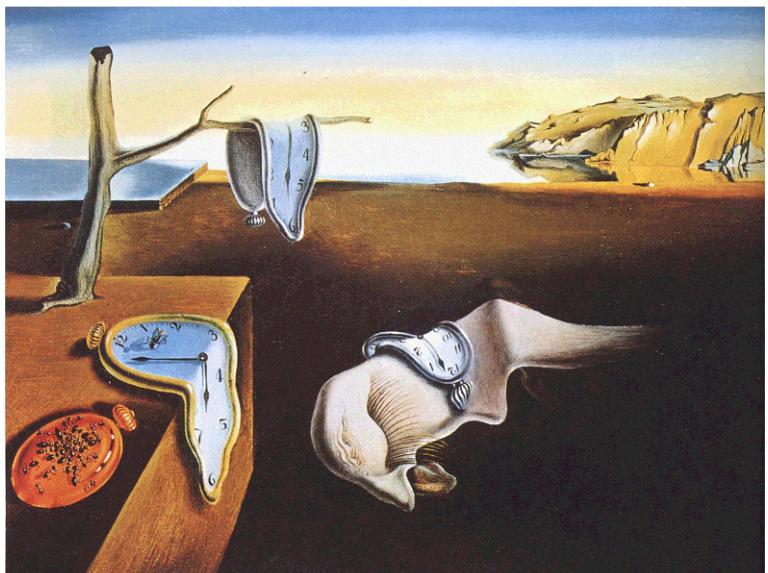


Figura 61: Arquivo BMP de 1.390 KB com um arquivo WAV de 217 KB esteganografado com a técnica LSB n; para $n = 4$.



Figura 62: Arquivo BMP de 215 KB com um arquivo PDF de 61 KB esteganografado com a técnica LSB n; para $n = 5$.

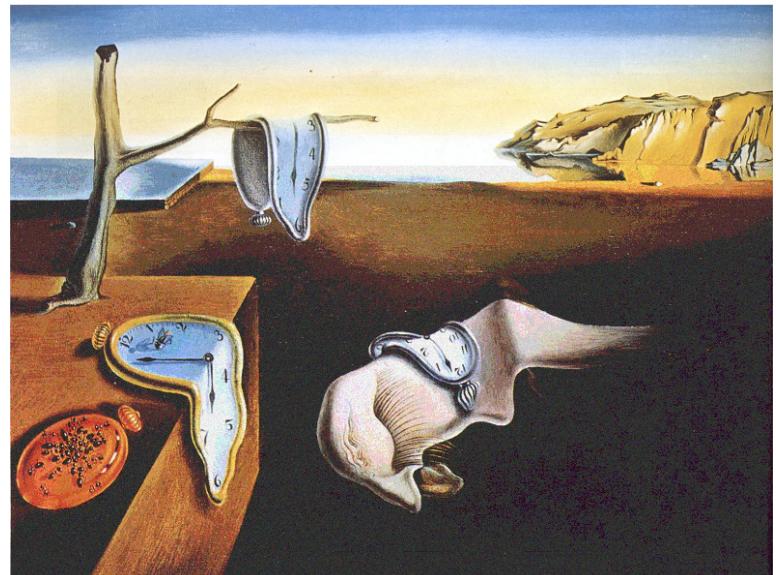


Figura 63: Arquivo BMP de 1.390 KB com um arquivo WAV de 217 KB esteganografado com a técnica LSB n; para $n = 5$.



Figura 64: Arquivo BMP de 215 KB com um arquivo PDF de 61 KB esteganografado com a técnica LSB n; para $n = 6$.

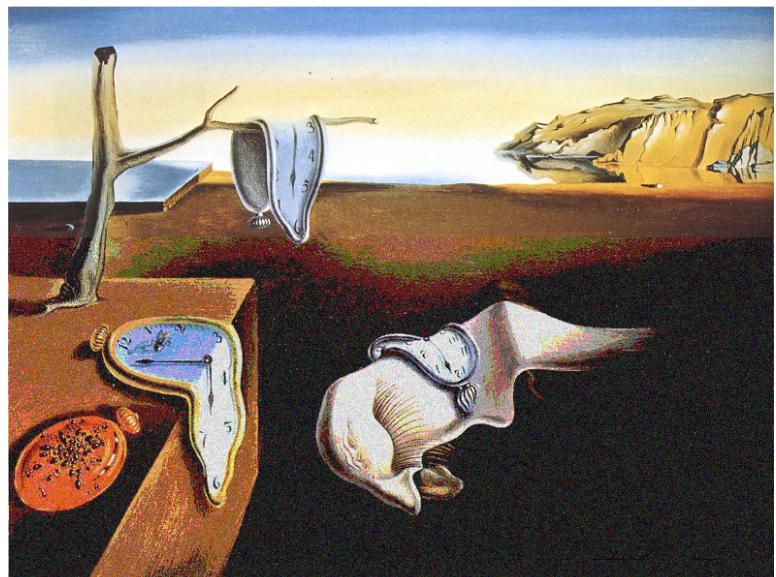


Figura 65: Arquivo BMP de 1.390 KB com um arquivo WAV de 217 KB esteganografado com a técnica LSB n; para $n = 6$.

6.4.3 ANALISE DOS RESULTADOS

Os testes realizados sugerem que o limite de *bits* possíveis de serem utilizados pela esteganografia em arquivo BMP sem que estes comprometam o conteúdo.

do da imagem é de aproximadamente quatro, ou seja, para um volume grande de dados a ser ocultado a melhor técnica seria LSB n , para $n = 4$. Nos casos em que está técnica não consiga suportar o tamanho do dado a ser ocultado, a solução seria utilizar como transporte um arquivo BMP maior e não aumentar o valor de n , sendo o ideal utilizar o menor valor possível para n .

Outra sugestão obtida com os testes é que as cores da imagem influenciam na qualidade da esteganografia, isso justificaria fato de ser obtido limites diferentes de n para as técnicas LSB n , pois as imagens que fizeram parte dos testes apresentam cores, tonalidades e distribuição distintas.

6.5 EXPERIMENTO 05

Foram utilizados como hospedeiros os dois arquivos de mídia no formato BMP.

6.5.1 CENÁRIO

Esteganografia LSB n ; para n valendo 7, adicionado ao arquivo BMP de 1.390 KB um arquivo DOC de 1.019 KB; e ao arquivo BMP de 215 KB foi adicionado um arquivo PDF de 48 KB.

6.5.2 RESULTADOS



Figura 66: Arquivo BMP de 1.390 KB original.



Figura 67: Arquivo BMP de 1.390 KB com um arquivo DOC de 1.019 KB esteganografado.

Em nenhum dos casos houve alteração no tamanho do arquivo original. A alteração nas cores dos arquivos foram significativas como pode ser observado na comparação da Figura 66 que ilustra a imagem BMP, de 1.390 KB, original com a Figura 67 que contém a imagem após a esteganografia; e a Figura 68, original, com a Figura 69 que apresenta comprometimento da imagem no arquivo BMP de 215 KB.



Figura 68: Arquivo BMP de 215 KB original.



Figura 69: Arquivo BMP de 215 KB com um arquivo PDF de 48 KB esteganografado.

6.5.3 ANALISE DOS RESULTADOS

A utilização da técnica LSB n , para $n = 7$, é completamente descartada para fins práticos, pois acarreta uma danificação significativa na imagem.

6.6 EXPERIMENTO 06

Foram utilizados como hospedeiros os dois arquivos de mídia no formato BMP.

6.6.1 CENÁRIO

Esteganografia LSB Cíclico com os mesmos arquivos utilizados no experimento LSB n , para $n = 7$; para fins comparativos.

6.6.2 RESULTADOS

Em nenhum dos casos houve alteração no tamanho do arquivo original. A alteração nas cores dos arquivos BMP de 1.390 KB apresenta-se demasiadamente danificada, como pode ser constatado comparando se a Figura 70 que ilustra a imagem original com a Figura 71 que contém a imagem com o dado esteganografado.



Figura 70: Arquivo BMP de 1.390 KB original.

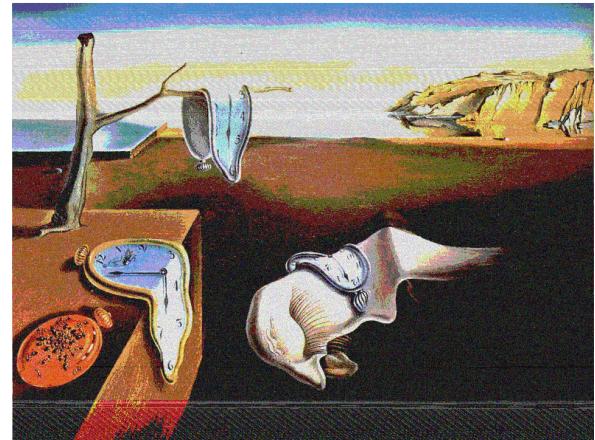


Figura 71: Arquivo BMP de 1.390 KB com um arquivo DOC de 1.019 KB esteganografado.

Já a esteganografia no arquivo BMP de 215 KB não causou alteração visualmente perceptível na imagem original. A comparação pode ser realizada observando a Figura 72 que exibe a imagem original com a Figura 73 que exibe a imagem contendo a esteganografia.



Figura 72: Arquivo BMP de 215 KB original.



Figura 73: Arquivo BMP de 215 KB com um arquivo PDF de 48 KB esteganografado.

6.6.3 ANALISE DOS RESULTADOS

Os dados que causaram distorção na imagem com a técnica LSB n , para $n = 7$, após serem esteganografados com o LSB Cíclico apresentaram resultados significativamente diferentes. Apesar do BMP de 1.390 KB continuar comprometido, a imagem BMP de 215 KB tornou-se idêntica a original, isso se deve ao fato do dado oculto ter sido distribuído de forma mais homogênea por toda a imagem. Mesmo a imagem danificada melhorou significativamente sua qualidade com a alteração da técnica.

O motivo que levou imagem tornar-se utilizável e a outra não, apenas com a troca da técnica se deve ao fato do tamanho do dado ocultado na primeira imagem ser próximo ao limite possível de ser esteganografado com a utilização de 7 *bits*, enquanto o segundo dado é significativamente menor do que o limite imposto pelo LSB n , para $n=7$.

A técnica LSB Cíclico demonstra ser interessante, pois utiliza sempre a menor quantidade de *bits* possível do *byte*, equiparando-se a técnica LSB quando o dado a ser ocultado é significativamente pequeno, porém, possibilitando um limite de tamanho igual à LSB n , para $n = 7$; contudo, considerando os testes realizados, o ideal seria limitar o LSB Cíclico à utilização de 4 *bits* por *byte*.

7 CONCLUSÃO E TRABALHOS FUTUROS

Nota-se através deste trabalho a existência de uma grande variedade de métodos para ocultar informação em mídias digitais. Sendo que para cada formato de arquivo existem métodos diferentes que tornam a esteganografia mais eficiente. Constatou-se que em todos os métodos analisados foram identificados diferentes pontos fortes e fracos; onde uma técnica carece de capacidade de carga, a outra carece de robustez.

Ao se considerar todos os métodos pesquisados, o formato do arquivo hospedeiro deve ser considerado junto com a informação a ser ocultada. Uma diferença no tamanho da informação, pode tornar a técnica mais eficaz ou comprometer sua eficiência.

Apesar das técnicas esteganográficas apresentarem limitações, ficou evidente que a esteganografia é uma técnica forte de segurança da informação, especialmente quando combinada com outras técnicas. O objetivo deste trabalho foi discutir e apresentar a esteganografia em mídias, sendo considerado na análise dos resultados apenas a percepção sensorial humana.

Não apenas robustez a ataques, mas a manipulações de processamento seriam assuntos interessantes de serem tratados futuramente, pois, a dificuldade de manter a informação esteganografada intacta após simples operações de processamento, incluindo os algoritmos de compactação com perda, são uma das grandes dificuldades encontradas nas técnicas esteganográficas. Os próximos trabalhos podem, também, propor soluções para as aplicações futuras descritas no tópico de aplicações. Um outro complemento para esse trabalho, seria o desenvolvimento de técnicas para ocultação de informação em mídia de áudio, onde o LSB apresentou problemas ou técnicas de ocultação em mídias de vídeo.

Os testes realizados foram satisfatórios e conclusivos ratificando a teoria elaborada no decorrer do trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

1. PINOCHET, Luis Hernan Cantreras. **Tecnologia da Informação e comunicação**, 1. Ed., Rio de Janeiro: Elsevier, 2014.
2. MARCACINI, Augusto Tavares Rosa. **Direito e Informática: uma abordagem jurídica sobre a criptografia**, São Paulo, 2010.
3. ÁLVAREZ, Benjamín Ramos; GARNACHO, Arturo Ribagorda. **Avances en Criptología y Seguridad de La Información**, Madrid: Díaz de Santos, 2004.
4. DUARTE, Otto Carlos Muniz Bandeira. **Esteganografia**, http://www.gta.ufrj.br/grad/09_1/versao-final/stegano/. Acessado em Julho de 2015.
5. ARABO, Abdullahi. **User-Centred and Context-Aware Identity Management in Mobile Ad-Hoc Networks**, Newcastle: Cambridge Scholars Publishing, 2013.
6. KNIGHT, Peter T.. **A Internet no Brasil: Origens, Estratégia, Desenvolvimento e Governança**, AuthorHouse, 2014.
7. PIMENTEL, Alex. **Tudo o que você precisa saber sobre economia**, São Paulo: Digerati Books, 2007
8. RANK, Michael. **Espiões, Espionagem e Operações Secretas -Da Grécia Antiga à Guerra Fria**, Babelcube, 2015.
9. JULIO, Eduardo Pagani. Brazil; WAGNER Gaspar. Albuquerque; CÉLIO Vini- cius Neves. **Esteganografia e suas Aplicações**. VII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais, 2007.
10. GONÇALVES, Alberto Cosme. **Gotas de reflexão**, Ribeirão Preto, 2011
11. CORDON, Juan Manuel Navarro; MARTINEZ, Tomás Calvo. **História da Filosofia**, Lisboa/ Portugal: Edições 70, 1995.

- 12.CANTANHEDE, Hugo Silva. **Esteganografia em Áudio e Imagem utilizando a técnica LSB**, Projeto Final de Curso -Bacharelado em Ciência da Computação, DCC-UFG, Catalão, GO, 2009.
- 13.HOSPODAR, Gabriel Mayrink da Rocha. **Algorítmos de Esteganografia via Restauração Estatística em Imagens Digitais**, Dissertação (Mestrado em Ciências em Engenharia Elétrica) -Curso de Pós-graduação em Engenharia Elétrica, COPPE-UFRJ, Rio de Janeiro, RJ, 2009.
- 14.TRIGUERO, Jesús J. Ortega; GUERRERO, Miguel Ángel López; CRESPO, Eugenio C. García del Castillo. **Introducción a la criptografía. Historia y actualidad**, Cuenca: Ediciones de la Universidad de Castilla-La Mancha, 2006
- 15.KIPPER, Gregory. **Investigator's Guide to Steganography**, CRC Press, 2003.
- 16.SCUDERE, Leonardo. **Risco Digital**, Rio de Janeiro: Elsevier, 2007
- 17.SCHÜRMANN, Henrique Augusto. **Criptografia Matricial Aplicada ao Ensino Médio**, Dissertação (Mestrado em Matemática) -Curso Matemática em Rede Nacional, UEL, Londrina, PR, 2013.
- 18.ORWANT, Jon; HIETANIEMI, Jarkko; MACDONALD, John. **Mastering Algorithms with Perl**, CA: O'Reilly & Associates, 1999
- 19.FOROUZAN, Behrouz A.. **Comunicação de Dados e Redes de Computadores**, São Paulo: McGraw-Hill Education, 2007
- 20.CASTELLO, Thiago. **Redes de Computadores II -Esteganografia** .
http://www.gta.ufrj.br/grad/07_2/thiago_castello/TcnicasModernas.html, Acessado em Agosto de 2015.
- 21.<http://www.cis.com.br/produtos/modulos-oem/item/mini-sr-ii-modulo>, Acessado em Agosto de 20015.
- 22.Committee on Defense Intelligence Agency Technology Forecasts and Reviews; National Research Council. **Avoiding Surprise in an Era of Global Technology Advances**, Washington DC: The National Academies Press, 2005.
- 23.JOHNSON, Neil F; DURIC, Zoran; JAJODIA, Sushil. **Information Hiding: Steganography and Watermarking-Attacks and Countermeasures**, Springer-Science+Business Media, LLC, 2001.

24. SINGH, Amardeep; GARG, Deepak. **Soft computing**, New Delhi: Allied Pub. Pvt. Ltd., 2005.
25. CLELLAND, Catherine Taylor; RISCA, Viviana; BANCROFT, Carter . **Hiding messages in DNA microdots**, Nature 399, 533-534 (10 June 1999).
26. COX, Ingemar J.; MILLER, Matthew L.; BLOOM, Jeffrey A; FRIDRICH, Jessica; KALKER, Ton. **Digital Watermarking and Steganography**, 2. Ed., Rio de Janeiro: Elsevier, 2008.
27. http://www.ieee.org/about/today/at_a_glance.html, Acessado em agosto de 2015.
28. RAGGO, Michael T.; HOSMER, Chet. **Data Hiding: Exposing Concealed Data in Multimedia, Operating Systems, Mobile Devices and Network Protocols**, Elsevier, 2012.
29. NEMATI, Hamid. **Information Security and Ethics: Concepts, Methodologies, Tools, and applications**, Hershey PA: IGI Global, 2007.
30. KARAGAINS, Joe; RENKEMA, Lennart. **Copy Culture in the US and Germany in the US & Germany**, California -USA: The American Assembly, 2013.
31. SANTINI, Rose Marie. **Admirável Chip Novo: A música na Era da Internet**, Rio de Janeiro: e-Papers, 2005.
32. MIANO, John. **Compressed image file formats: JPEG, PNG, GIF, XBM, BMP**, Massachusetts: Library of Congress, 1999.
33. BONI, Paulo César. **Fotografia: múltiplos olhares**, Londrina: Midiograf, 2011.
34. GRAÇA, Ricardo. **Produzindo Animações com Softwares Livres**, Rio de Janeiro: RME Comunicações e Idiomas ME, 2012.
35. FERREIRA, Silvio. **Tudo o que você precisa saber sobre Áudio e Vídeo digital**, São Paulo : Digerati Books, 2008.
36. GODSE, A.P.; GODSE, D.A.. **Computer Graphics And Multimedia, Technical Publications**, 2009.
37. TOULSON, Rob; WILMSHURST, Tim. **Fast and Effective Embedded Systems Design: Applying the ARM Mbed**, Elsevier, 2012.
38. TAKAHASHI, Fernando Yamaguti. **Produtor de Música Baseado em Microcontrolador ARM7 para Reprodução de Arquivos WAV Armazenados em Cartão de memória**, Trabalho de Conclusão de Curso -Curso de Engenharia

- Elétrica com Ênfase em Eletrônica, Escola de Engenharia de São Carlos, USP, São Carlos, SP, 2012.
39. JUNIOR, Elias Daher. **A Culpa é Da Informática**, Clube de Autores, 2015.
40. Fridrich, Jessica. **Steganography in Digital Media: Principles, Algorithms, and Applications**, UK: Cambridge, 2009.
41. JOHNSON, Neil F.; JAJODIA, Sushil. **Steganalysis: The Investigation of Hidden Information**, Virginia: George Mason University, 1998.
42. CHITODE, Dr.J.S.. **Analog And Digital Communication**, Technical Publications Pune, 2009.
43. PROAKIS, J. G.. **Digital Communications**, 2 Ed., NY: McGraw-Hill, 1989.
44. FOROUZAN, Behrouz A.. **Comunicação de Dados e Redes de Computadores**, 4 Ed., NY: McGraw-Hill, 2007.
45. BRANDÃO, Tomás Gomes da Silva Serpa. **Assinatura Digital de Imagens Baseada em Espalhameto de Espectro**, Dissertação (Mestrado em Engenharia Eletrotécnica e de Computadores) -Mestre em Engenharia Eletrotécnica e de Computadores, Instituto Superior Técnico -Universidade Técnica de Lisboa, Lisboa, Portugal, 2012.
46. BOOCHE, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: guia do usuário**, 2 Ed., Elsevier, 2005.

ANEXOS

ANEXO A – Demonstração de esteganografia em final de arquivo.

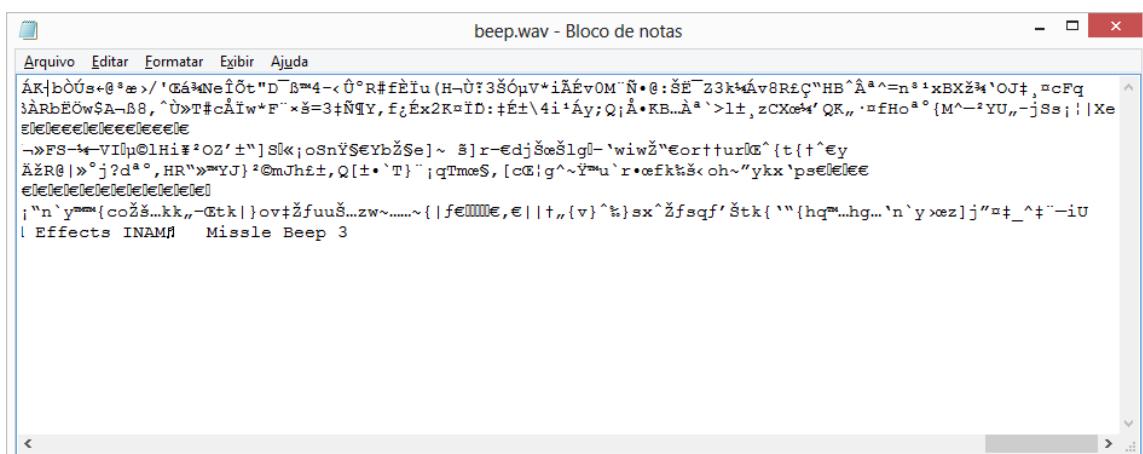


Figura 74: Arquivo beep.wav visualizado em um editor de texto.

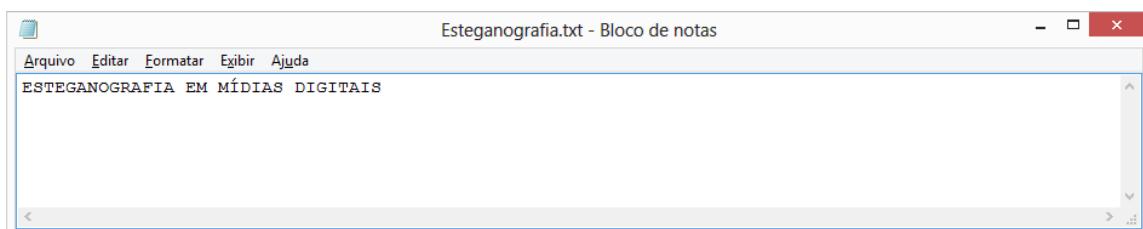


Figura 75: Arquivo Esteganografia.txt visualizado em um editor de texto.

```
copy /b beep.wav+Esteganografia.txt beepEsteganografia.wav
```

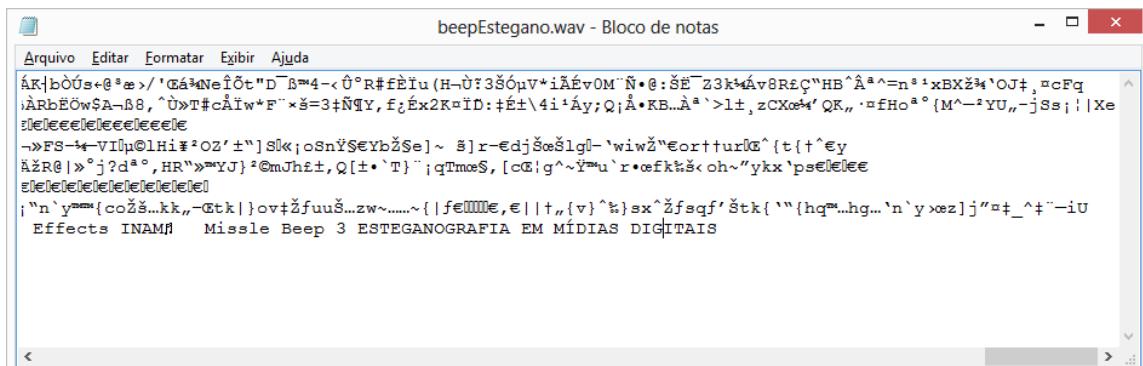


Figura 76: O arquivo beep.wav com a mensagem esteganografada.