UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

MARCIO RANIERI TEIXEIRA

Esteganografia em arquivos texto e técnicas de detecção

Trabalho de Graduação.

Prof. Dr. Raul Fernando Weber
Orientador

Porto Alegre, dezembro de 2011.

AGRADECIMENTOS

Primeiramente, gostaria de agradecer a minha família e a minha namorada, pois sempre me apoiaram e deram a sustentação necessária para que eu pudesse superar as barreiras impostas até a conclusão deste curso.

Agradeço ao professor Raul Fernando Weber pelas suas valiosas orientações e por partilhar seu conhecimento e experiência na produção deste trabalho.

Por último, agradeço aos professores e funcionários do Instituto de Informática da UFRGS por propiciarem diversos momentos de aprendizado.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

ASCII American Standard Code for Information Interchange
HTML HyperText Markup Language
GPL General Public License
INTERSNOW Aplicativo com interface para ferramenta Snow.
SPAM Mensagem de correio eletrônico com fins publicitários.
W3C The World Wide Web Consortium.

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

A Esteganografia é a arte de ocultar mensagens dentro de um objeto com aparência inocente sem que terceiros percebam a existência da mensagem original. Essa técnica foi utilizada e aperfeiçoada no decorrer da nossa história e, nos dias atuais, é encontrada em diversos objetos, entre eles: arquivos texto. Para camuflar mensagens, a Esteganografia em arquivos texto faz uso de muita criatividade combinada com algumas características normais encontradas neste tipo de objeto em questão. No caso de arquivos textos: espaços em branco, preposições e pontuação. Neste contexto, esse trabalho tem como objetivo demonstrar essas técnicas e, também, formas de detectar arquivos que possam conter essas mensagens. Para tanto, serão apresentados os conceitos fundamentais da Esteganografia, bem como suas principais características e aplicações. Juntamente com esses conceitos, será apresentado o funcionamento de um aplicativo de licença GPL chamada Snow que utiliza os espaços e as linhas em branco de um arquivo para esconder mensagens. A partir desta ferramenta, será criado um projeto de uma interface amigável para utilização da mesma, já que o Snow, até o momento, tem versão somente com uso em linha de comando. Além disso, acrescentar novas funcionalidades ao aplicativo pertinentes e este estudo proposto. Logo, pretende-se, com este trabalho e projeto, demonstrar de forma clara o uso da Esteganografia para ocultação e extração de mensagens ocultas dentro de arquivos texto, bem como sua detecção.

Palavras-Chave: Esteganografia, Programa Snow, Criptografia, Esteganálise.

Steganography in Text Files and Detection Techniques

ABSTRACT

Steganography is the art of hiding messages inside an innocent-looking object in such a way that an outsider doesn't perceive the existence of the original message. This technique was used and perfected over the course of our history. And today is found in several objects, including text files. To camouflage messages, text files Steganography uses a lot of creativity combined with some standard features found in this type of object in question (text files), for example, blank spaces, punctuation and prepositions. In this context, this study aims to demonstrate these techniques, as well as ways to detect files that could contain these messages. For this, we presented the fundamental concepts of Steganography, and its main characteristics and applications. Along with these concepts will be presented the operation of a GPL application called Snow that uses spaces and blank lines from a file to hide messages. This software will also be used to create a project with a friendly user interface, since Snow, to date, have only command line versions. The aim of this work and this project clearly demonstrate the use of Steganography for hiding and extracting a hidden message into text files.

Keywords: Steganography, Snow Software, Encryption, Steganalysis.

1 INTRODUÇÃO

Uma das características mais interessantes da Esteganografia é que ela procura esconder um segredo em algum tipo de objeto de tal forma que terceiros não consigam discernir a sua presença. Esse técnica, não deve ser confundida com a da sua "irma" criptografia que não se importa de aparecer e nem faz questão de esconder que carrega informações secretas, onde, o desafio é simplesmente encontrar a chave para extrair os dados ali contidos, diferente da Esteganografia que precisa ser percebida, desconfiada, examinada antes de ser atacada por alguma técnica de decodificação que revele os dados. O bom, é que nada impede que ambas sejam usadas em conjunto, ou seja, os dados camuflados pela esteganografia podem ser criptografados.

Hoje, a vasta disponibilidade de objetos digitais como arquivos texto, mp3, imagens, páginas Web, datagramas ip, códigos de programas, etc, contribuiu para chamar a atenção e difundir o uso e a pesquisa da esteganografia sobre esses objetos principalmente depois que foi aventado que essa técnica pode ter sido usada em arquivos de imagens para planejar os ataques de 11 de Setembro. Mesmo assim, a esteganografia em arquivos texto, ao contrário da em imagens, não possui ferramentas robustas disponíveis para que se possa explorar de uma forma aprazível esse assunto sem ter que recorrer aos não intuitivos *comandos de linha*.

Esse trabalho é voltado para estudantes ou amantes desta área de conhecimento e se propõe a apresentar de forma singela alguns conceitos e aplicações das técnicas de esteganografia em arquivos textos, juntamente com métodos de detecção e ataque. Em seguida, o funcionamento de uma ferramenta de licença livre (GPL) que trabalha em cima de uma dessas técnicas de esteganografia explicitadas. E, por fim, a especificação de um aplicativo que se propõe usar os conceitos manifestados nesta obra e que será modelo para criação um protótipo de uma interface amigável e intuitiva para mostrar, na prática, a conjunção do que aqui foi abordado.

2 CONCEITOS INICIAIS

Esse capítulo apresenta os primeiros registros históricos da Esteganografia, seu funcionamento básico, algumas técnicas esteganográficas mais usadas em arquivo texto, bem como, formas de detecção.

Basicamente, o conceito de Esteganografia é ocultar informações dentro de objetos, como por exemplo: arquivos de imagens, vídeos, musicas, textos, etc, sem levantar suspeita. Para tanto, utiliza algumas técnicas desenvolvidas através dos anos para que a comunicação fique camuflada e não seja vista. Por exemplo: Alguém deseja enviar uma mensagem para outra pessoa, porém não confia no transportador (o transportador não sabe disso). [POLLON 2007]

2.1 Histórico da Esteganografia

Esteganografia é uma palavra que vem do grego (Steganós = oculto, misterioso) e em seu conjunto significa "escrita oculta".

Diversos registros presentes na história do homem mostram a esteganografia sendo utilizada há muitos anos. Tanto que no século XV, o monge Johannes Trithemius (Figura 2.1) publicou um livro de três volumes, disfarçado como um livro de magia negra, chamado *Steganographia* onde detalhava várias técnicas para enviar mensagens sem que elas fossem percebidas.

Figura 2.1: Johannes Trithemius

Métodos como as famosas tintas “invisíveis” utilizando urina ou limão. Ou, escrever uma mensagem em um ovo cozido (século XVI, Giovanni Porta), marcar certas letras com furinhos (técnica utilizada na Primeira Guerra pelos alemães), gestuais em fotos e vídeos durante guerras e até mesmo tatuagens sob o cabelo de mensageiros, são alguns exemplos históricos do uso da esteganografia. [POLLON 2007]

Atualmente, a esteganografia é amplamente utilizada em nossa sociedade e pode ser encontrada na segurança monetária para garantir a autenticidade das cédulas e dificultar a falsificação. Grandes fabricantes de impressoras também a utilizam para garantir o não repúdio da impressão. Filmes e músicas, para garantir a autenticidade e proteger os direitos autorais. Imagens digitais, sistemas de arquivos, pacotes TCP/IP e páginas html. Nesta última, a mensagem secreta pode ser incluída no código fonte ou escrita com os caracteres da mesma cor do fundo da página, conhecido como *texto invisível,* utilizado para melhorar o posicionamento das páginas nos sites de busca já que o visitante não percebe o texto adicional, porém, são lidos e considerados pelos robôs de busca.

2.2 Resumo Esquemático da Esteganografia

A esteganografia funciona basicamente com os seguintes componentes: a mensagem secreta que se deseja enviar, o recipiente onde essa mensagem vai ser camuflada, que pode ser um áudio, vídeo, texto, imagem, etc. Um estego-objeto que é a união dos dois componentes anteriores (mensagem secreta + recipiente). Por fim, o método ou técnica que vai ser utilizada para controlar o processo de esconder e recuperar o dado secreto. A Figura 2.2 ilustra esse processo.

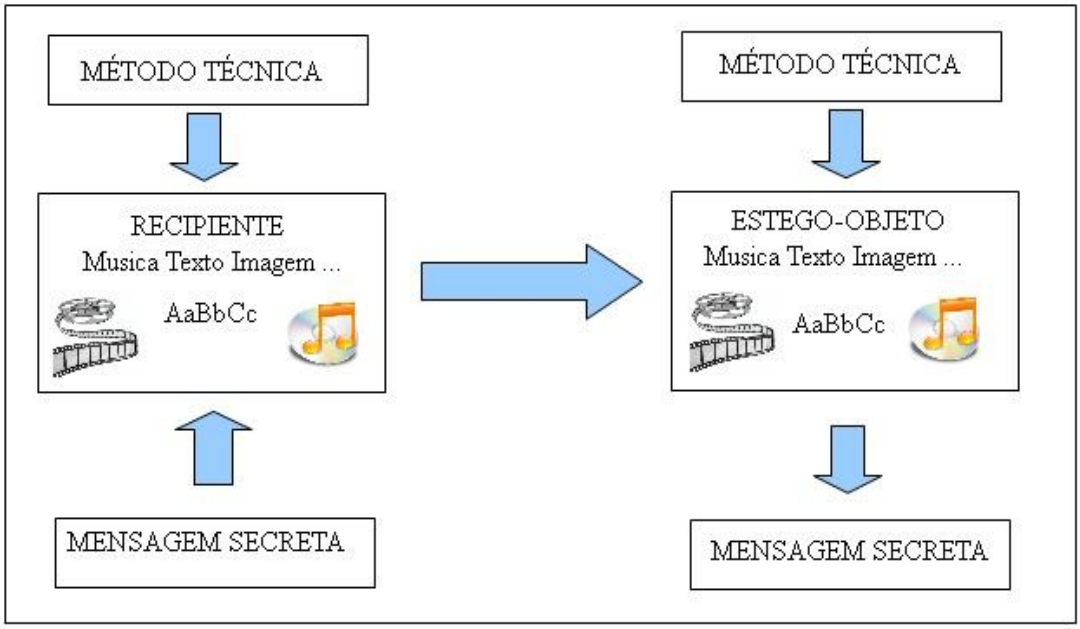


Figura 2.2 (PETRI 2004)

2.3 Técnicas em aquivo texto

As técnicas de Esteganografia usadas em arquivos texto mais conhecidas são:

- codificação por deslocamento de linhas verticalmente
- codificação por deslocamento de palavras em textos justificados
- codificação por adição de pontuações e preposições
- codificação por deslocamento do carácter de fim de linha
- XML

2.3.1 Codificação por deslocamento de linhas verticalmente

Essa técnica consiste em deslocar verticalmente uma pequena fração várias linhas de um texto. Dependendo do dicionário, para cima pode representar um valor de código 1 e para baixo um valor de código 0. Quando esse deslocamento é mínimo ele fica imperceptível aos olhos humanos porém um programa de computador pode medir a distância de cada uma das linhas facilmente. Um detalhe nesta técnica é a codificação diferencial, ou seja, se uma linha é deslocada, as linhas adjacentes não são movidas a fim de que o computador possa ter uma referência ao medir as distâncias entre elas. Uma limitação dessa técnica é a pouca informação que pode ser escondida em um texto. Uma página de formato A4, por exemplo, com espaçamento simples e caracteres tamanho 12, comporta aproximadamente 50 linhas para serem usadas.

2.3.2 Codificação por deslocamento de palavras em textos justificados.

Essa técnica é baseada no mesmo princípio da codificação por deslocamento de linhas verticalmente e consiste utilizar os espaços em branco entre as palavras de um texto, de forma que elas se desloquem horizontalmente e assim representar uma codificação. Podendo, por exemplo, **um** espaço significar o **bit 0** e **dois** espaços corresponder ao **bit 1**, procurando manter a aparência de espaçamento natural, já que muita distorção no texto pode levantar suspeita.

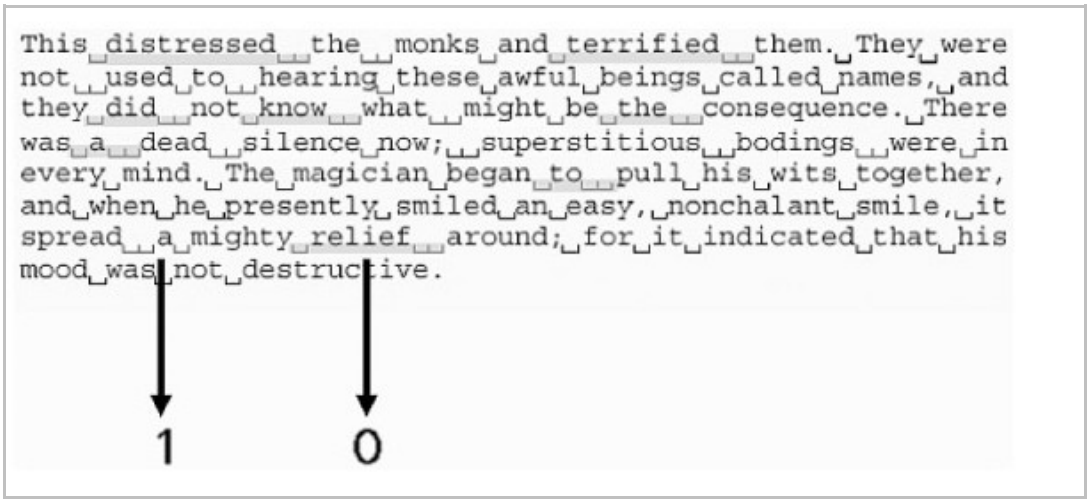


Figura 2.3 (KIPPER 2004)

Uma variação que torna essa técnica difícil de ser percebida aos olhos é deslocar as palavras para a direita ou para esquerda em “meio espaço” ou 0,5 pontos. Podendo o

deslocamento à esquerda representar o **bit 0** e a **direita** representar **bit 1**. No exemplo da Figura 1.3.2 a primeira linha usa um espaçamento padrão enquanto que na segunda linha cada palavra é deslocada para direita ou para esquerda a fim de codificar a sequência de 01000001 que em ASCII é '65' ou o carácter 'A'. Se utilizada em um texto justificado e sem ter a original para comparar, é provável que seja difícil perceber que os espaços carregam alguma informação secreta. (CUMMIS 2004)

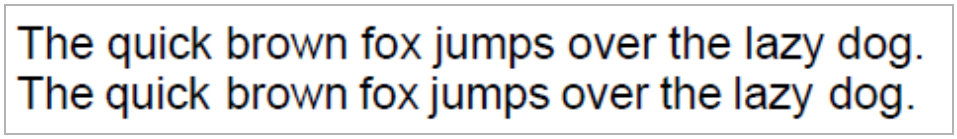


Figura 2.4 (CUMMIS 2004)

Esta técnica, embora funcione de maneira eficaz, possui alguns inconvenientes. Um deles é que se o documento for digital, qualquer processador de texto mais moderno poderia mostrar o espaçamento irregular ou reformatar o documento. Destruindo, assim a mensagem escondida. Outra desvantagem é, assim como a codificação por deslocamento de linhas verticalmente, a pouca informação que pode ser inserida neste documento, tornando esse método muito limitado. [KIPPER 2004]

2.3.3 Codificação por adição de pontuações e preposições

Consiste em ocultar informações utilizando a gramática de forma aleatória para criar um texto que, por vezes, não é lá muito coerente mas que aproveita objetos portadores já existentes e suas características para camuflar uma mensagem. Usa o recurso de pré-definir um significado para cada palavra (ou grupo de palavras) ou pontuações. Um bom exemplo dessa técnica que é usas nos dias atuais é o site SpamMimic [WAYNER 2003] e que é simplesmente uma implementação moderna do método “Ave Maria” (Figura 2.5) criado por Johannes Trithemius, já citado neste trabalho. O SpamMimic oculta informações em mensagens que aparentam ser aqueles e-mails não solicitados conhecidos com SPAM e que são propagados pela internet através de correio eletrônico. [CUMMIS 2004]

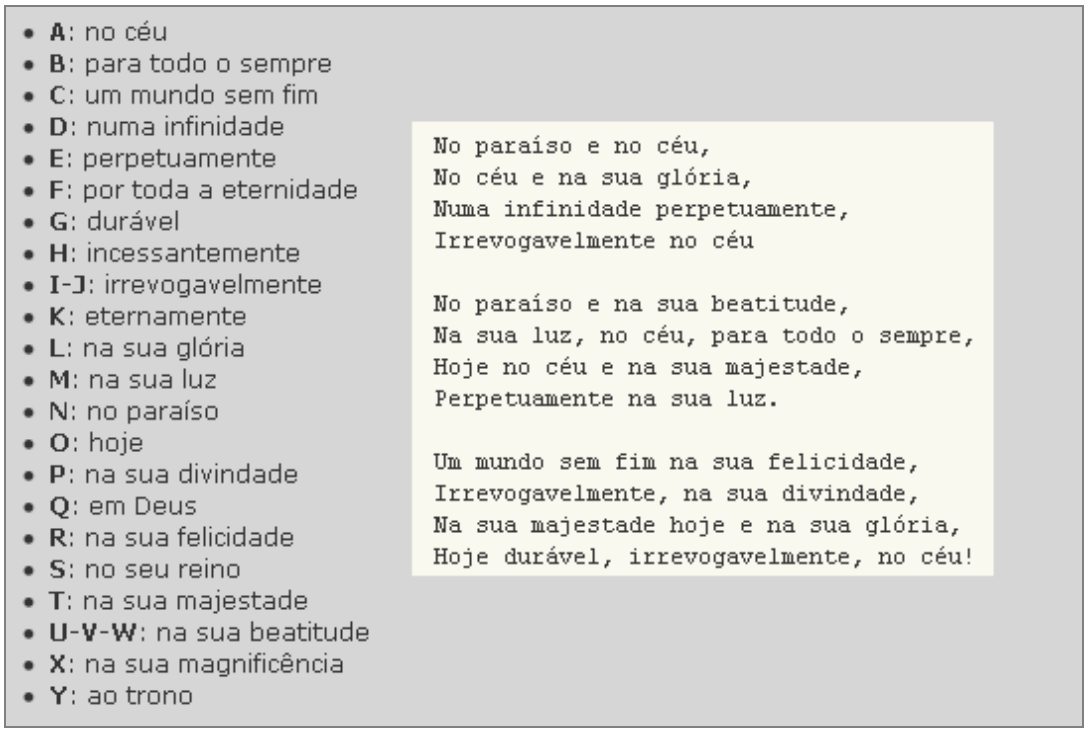


Figura 2.5 Ave Maria de Trithemius (Vicki 2007)

Na figura 2.5 um dos alfabetos de Trithemius ilustrando essa técnica de esteganografia. Depois de decifrar o poema da imagem, é encontrado a seguinte mensagem: "*Na Aldeia NumaBoa tem Criptologia***"** (Vicki 2007)

2.3.4 Codificação por deslocamento do carácter de fim de linha

Essa técnica consiste em utilizar as linhas em branco e o espaço que sobra no final das linhas de um texto para codificar uma mensagem, usando para isso, os caracteres "espaço" e "tab", onde, uma tabulação indica o início do texto secreto ou código que será codificado por espaços em branco. No caso exemplificado abaixo, a representação de "bits" utilizados na codificação são gravados de três em três e cada conjunto deles são separados por uma tabulação (podem ser utilizados zeros para completar o último trio).

São utilizados de 0 a 7 espaços para definir o estado (Ligado/Desligado) dos três bits e contanto que os espaços no final das linhas ocoram naturalmente, a existência do código não deve ser suficiente para alertar imediatamente um observador. Vejamos como fica essa codificação na Tabela 2.1 abaixo: [POLLON 2007]

Tabela 2.1: Tabela de codificação e espaços

Zero espaços	000
Um espaço	001
Dois espaços	010
Três espaços	011

Quatro espaços	100
Cinco espaços	101
Seis espaços	110
Sete espaços	111

Fonte: POLLON, 2007. p. 12.

A título de exemplo, a palavra secreta “ufrgs” será inserida em um suposto arquivo de texto utilizando essa técnica. O arquivo original ou recipiente contém o seguinte conteúdo:

> Este texto serve como exemplo para detecção.
>
> Iremos incluir no final de cada linha um código.
>
> Este código para quem está lendo é invisível.

A Tabela 2.2 e a Figura 2.6 mostram como cada carácter da palavra secreta será codificada com o seu respectivo valor Hexadecimal da Tabela ASCII em formato binário. Onde, no texto, os espaços representando o conjunto de 3 bits serão separados por uma Tabulação.

Tabela 2.2: Codificação caracteres para hexa e binário

Carácter	*Hexa*	*Binário*	*Descrição*
u	75	0111 0101	Letra u
f	66	0110 0110	Letra f
r	72	0111 0010	Letra r
g	67	0110 0111	Letra g
s	73	0111 0011	Letra s

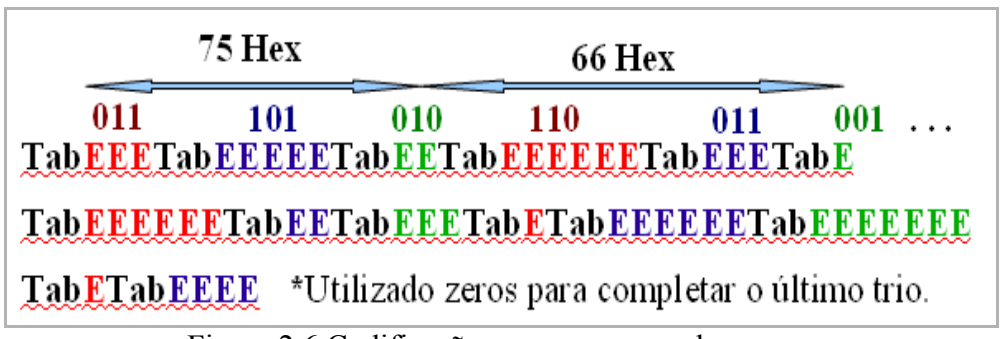


Figura 2.6 Codificação nos espaços em branco

Ao abrir o arquivo com um editor de texto do tipo Notepad, Word ou BrOffice, não deve aparecer o código inserido, já que este é formado com espaços e tabulações. Porém, ao habilitar a opção de "Visualizar caracteres não imprimíveis" do editor BrOfffice, por exemplo, o texto é aparece conforme a Figura 2.7. ou seja, os códigos inseridos nos espaços em branco do arquivo são expostos.

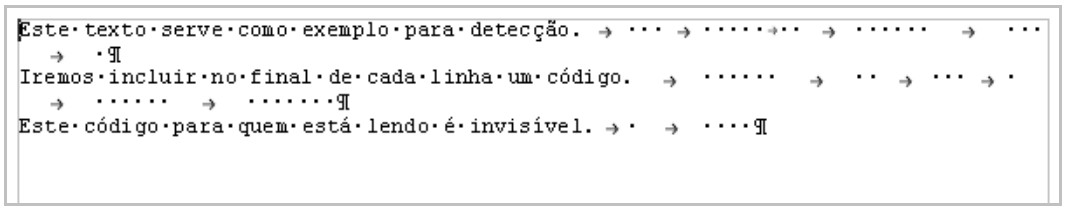


Figura 2.7 Texto com o código de tabulações e espaços visíveis

Uma alternativa interessante que pode ser usada com essa técnica e que permite inserir mensagens maiores e aproveitar melhor os espaços do arquivo é compactar o código secreto. Dependendo da taxa de compressão, a mensagem pode ser reduzida em 25% a 40% dependendo do texto. O aplicativo Snow utiliza a codificação por deslocamento do carácter de fim de linha e compressão facultativa no momento de inserir uma mensagem em um arquivo. Para tanto, usa o algoritmo de compressão Huffman em conjunto com a esteganografia .

2.3.5 XML

O XML é um formato amplamente utilizado na internet para troca de dados e criação de páginas Web, e que o torna um objeto propício para esconder informações, pois, com XML pode-se ter muitos arquivos com codificações diferentes para representar o mesmo conteúdo, ou seja, alterações nos códigos de uma página XML podem não influenciar na maneira com que ela é vista na tela de um navegador web. Uma das maneiras de fazer isso é usar, no código XML, *tags* diferentes das permitidas pela W3C mas que funcionam mesmo assim quando interpretadas pelos navegadores.

No exemplo abaixo, é usado uma técnica com *tags* duplas para representar um valor, no caso “0” e uma *tag* simples com barra para representar o valor “1”. Neste código as tags de *img* são válidas e podem ser usadas para esconder algum código.

Chave:

<img></img> ….0

<img/> …..... 1

Dados:

<img src=”foo1.jpg”></img> ---------- 0

 ---------- 1

 ---------- 1

 ---------- 1

<img src=”foo5.jpg”></img> ---------- 0

Dados inseridos:

01110

A inserção de espaços em branco depois do nome da *tag* é outra técnica que pode ser utilizada para esconder dados sem alterar a forma como é apresentado o documento XML. Abaixo, um exemplo desse método:

Chave:

<tag>, </tag>, or <tag/> ….. 0

<tag >, </tag >, or <tag /> … 1

Datos:

<user ><name>Alice</name ><id >01</id></user>

<user><name >Bob</name><id>02</id ></user >

Dados inseridos:

101100 010011

Um programa que manipule os espaços em branco dentro de arquivos XML pode ser criado de forma relativamente simples e assim, tornar essa tarefa de esconder e inserir dados no arquivo mais simples. [CUMMIS 2004]

2.4 Comparativo

Neste capítulo será feito um comparativo simples entre as técnicas de esteganografia em arquivo texto vistas neste trabalho objetivando demonstrar quais podem carregar mais informações. Será comparado também, alguns formatos de arquivos texto, visando descobrir os recipientes ou tipos de arquivo que podem absorver a maior carga útil sem despertar suspeitas.

2.4.1 Método utilizado para o comparativo entre as técnicas

Para esse comparativo utilizou-se como base uma página de texto retirada do documento "Normas para Apresentação de Trabalhos do Instituto de Informática e do PPGC", Capítulo 3, página 17. Disponível na página Web da Biblioteca da CIC. Esse documento tem tamanho A4 e não contém imagens, somente 34 linhas de 84 colunas de caracteres. O distanciamento de linhas é padrão. Não foi utilizado nenhuma ferramenta especial para essa análise. Somente o editor de texto BrOffice e calculadora.

Em todas as técnicas mostradas na tabela foi calculado o quando de unidades de informação ela podia armazenar e esse número, em alguns casos, foi convertido para bytes, ou seja, foi dividido por oito. Ou, pode ser visto como um caractere da mensagem que podemos formar.

Para estimativa de cálculo, na técnica de Adição de pontuações e preposições, foi utilizado o exemplo da Figura 2.5. Onde, 52 palavras representam as 25 letras do alfabeto. Resultando numa média de 2,36 palavras para cada letra. Já na técnica XML, a página do texto foi convertida para o formato HTML e suas *tags* foram contadas. Resultando em aproximadamente 134 *tags.*

2.4.2 Tabela

Tabela 2.3 Comparativo de Técnicas Esteganográficas em texto

Técnica	*Resultados*
Deslocamento de linhas verticalmente	Nas 34 linhas da página pode-se armazenar 33 unidades de informações. Ou, se convertido, **2 bytes/caracteres**.
Deslocamento de palavras em textos justificados	O texto contém 389 palavras. Onde se consegue armazenar 389-2 = 387 unidades de informação. Convertido, tem-se **48 bytes/caracteres**.
Adição de pontuações e preposições (Ave Maria)	O texto contém 389 palavras. Dividido por 2,36 (média de palavras para representar um caractere). Totalizando **164 caracteres**.
Deslocamento do caracter de fim de linha	O texto da página possui 9 linhas e meia de espaços em branco. Cada linha de 84 colunas. Podendo armazenar 798 unidades de informação. Ou seja, **100 bytes/caracteres.**
XML	As 134 *tags* pertencentes ao texto podem guardar o mesmo número de unidades de informação, que convertidas representam **16 bytes/caracteres**.

2.4.3 Conclusões

Embora a técnica de Adição de pontuações e preposições possua a maior capacidade de carregar caracteres escondidos no texto, ela tem a desvantagem de só poder representar 25 deles e de apresentar um texto sem sentido para o observador. O que a torna uma técnica de fácil detecção.

Nas demais técnicas, concluiu-se aquilo que já era esperado: com exceção da técnica de deslocamento do caractere de fim de linha, as demais possuem uma baixa capacidade de carregar informações. O que torna a técnica de deslocamento do caractere de fim de linha uma boa escolha, pois, além dessa característica citada, ela é de difícil percepção visual.

Se a opção de técnica escolhida for essa última, ela terá uma melhor capacidade de carga se utilizada em um texto não justificado, em virtude de sobrarem mais espaços no final de cada linha. Quando o formato do arquivo recipiente for XML/HTML isso não tem importância, já que, os dados serão inseridos no código do arquivo. Ou seja, justificado ou não, o plicativo que irá exibir esses arquivos, normalmente um Browser, irá desconsiderar aqueles espaços ali inseridos.

2.5 Detecção

Dependendo da técnica, a esteganografia pode fornecer vários níveis de segurança e sigilo à informação em um grau, por vezes, difícil de ser medido. Porém, longe de garantir que os dados secretos não sejam interceptados ou notados por terceiros. Quando a esteganografia é utilizada em um objeto, este sofre alterações onde a sua qualidade é degradada de tal forma que, dependendo da técnica ou *payload*[1] da informação, essa distorção pode ser perceptível aos sentidos humanos ou detectada por alguma ferramenta de esteganálise. [PETRI 2004]

Esteganálise é o nome que se dá ao estudo de métodos que visam detectar e revelar uma mensagem oculta por técnicas esteganográficas dentro de um objeto. Apontando onde a técnica esteganográfica não foi suficiente o bastante e falhou na camuflagem dos dados. O que nem sempre é fácil, devido as novas técnicas de esteganografia que podem não terem sido consideradas durante a esteganálise. Caso detectada, utiliza métodos conhecidos como **ataques** para tentar decodificar a mensagem, alterar ou até mesmo destruí-la.

Decodificar a mensagem e conseguir recuperar os dados ali contidos requer um esforço maior no processo de esteganálise até porque a mensagem pode estar criptografada. Devido a esse fato, para alguns esteganalistas, basta somente verificar se existe informações escondidas na mensagem e destruí-la. Ou, alterar o conteúdo para que a informação secreta se perca. O curioso é que, dependendo da situação, essa forma de ataque pode descobrir a existência ou o tipo de técnica usada. A título de exemplo, pode ser ditado um caso famoso da I Guerra Mundial em que uma mensagem foi interceptada e alterada de "Father is dead" para "Father is deceased" e reenviada so seu destinatário. Quando foi interceptada a resposta, esta veio com os dizeres: "Is Father is dead or deceased?" Neste momento os especialistas descobriram que se tratava de uma mensagem secreta. Neste mesmo contexto, outra forma de ataque pode ser feita alterando o formato do arquivo, já que diversos formatos armazenam seu conteúdo de diversas maneiras. Outra também, comprimindo o arquivo, pois a grande maioria dos algoritmos de compressão eliminam os dados extras.

Outras técnicas de ataque mais conhecidas em arquivo texto:

- Ataque contra estego-objeto.

  - É aquele ataque onde se tem somente o estego-objeto para tentar detectar e extrair a mensagem secreta.

- Ataque contra estego-objeto conhecido.

  - É o ataque onde se conhece o estego-objeto e o recipiente, de forma que uma comparação entre os dois pode se feita.

- Ataque contra estego-objeto com mensagem conhecida.

  - Esse ataque pressupõem que se conhece a mensagem e o estego-objeto, de forma que o interesse é descobrir a técnica que foi utilizada para inserir a mensagem secreta.

---

1 Termo utilizado para designar o tamanho da mensagem esteganográfica ou a carga útil de um estego-objeto.

- Ataque contra estego-objeto escolhido.

- É o ataque onde se tem tanto o estego-objeto e a ferramenta esteganográfica ou algoritmo.

- Ataque contra estego-objeto com mensagem escolhida.

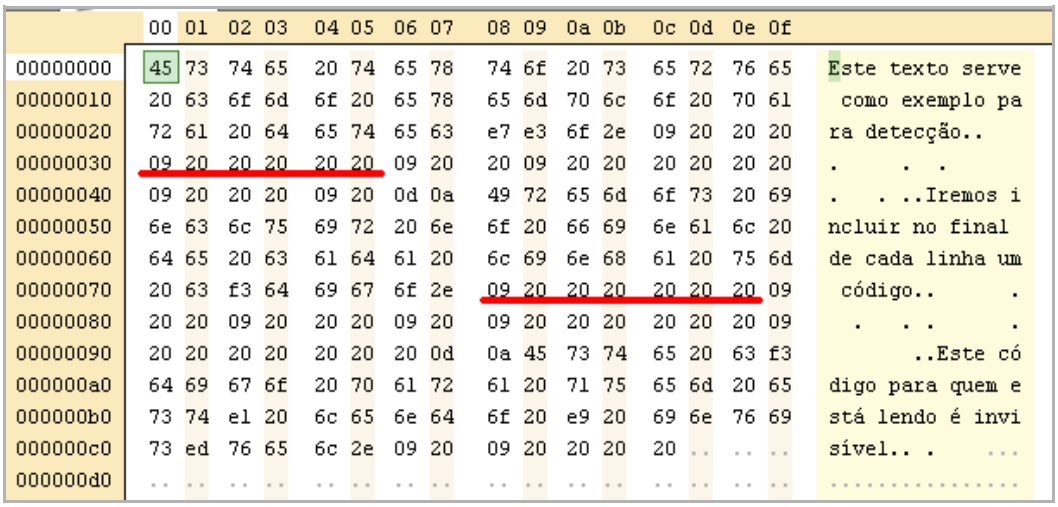
- É o ataque onde o esteganalista gera um estego-objeto a partir de uma mensagem utilizando uma ferramenta específica. Para com isso, buscar assinaturas que permitam detectar outros estego-objetos. [KIPPER 2004]

- Ataque visual contra estego-objeto.

- É o ataque onde o arquivo é inspecionado de forma visual procurando encontrar algumas distorções perceptíveis que sugerem algum tipo de manipulação.

Esse último tipo de ataque é justamente o que a esteganografia eficiente tenta evitar que tenha exito. Já que procura explorar a percepção e limitações dos olhos humanos para esconder mensagens.

Pode ser feito um ataque contra estego-objeto, que utilizou a técnica esteganografia de deslocamento do carácter de fim de linha, verificando o se o arquivo (estego-objeto) possui muitos espaços em branco em sequencia sempre após tabulações. Para ilustrar este exemplo, um arquivo que contém uma mensagem secreta foi editado de forma hexadecimal conforme a Figura 2.8.



	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
00000000	45	73	74	65	20	74	65	78	74	6f	20	73	65	72	76	65	Este texto serve
00000010	20	63	6f	6d	6f	20	65	78	65	6d	70	6c	6f	20	70	61	como exemplo pa
00000020	72	61	20	64	65	74	65	63	e7	e3	6f	2e	09	20	20	20	ra detecção..
00000030	09	20	20	20	20	20	09	20	20	09	20	20	20	20	20	20	. . .
00000040	09	20	20	20	09	20	0d	0a	49	72	65	6d	6f	73	20	69	. . ..Iremos i
00000050	6e	63	6c	75	69	72	20	6e	6f	20	66	69	6e	61	6c	20	ncluir no final
00000060	64	65	20	63	61	64	61	20	6c	69	6e	68	61	20	75	6d	de cada linha um
00000070	20	63	f3	64	69	67	6f	2e	09	20	20	20	20	20	20	09	código.. .
00000080	20	20	09	20	20	20	09	20	09	20	20	20	20	20	20	09	. . . .
00000090	20	20	20	20	20	20	20	0d	0a	45	73	74	65	20	63	f3	..Este có
000000a0	64	69	67	6f	20	70	61	72	61	20	71	75	65	6d	20	65	digo para quem e
000000b0	73	74	e1	20	6c	65	6e	64	6f	20	e9	20	69	6e	76	69	stá lendo é invi
000000c0	73	ed	76	65	6c	2e	09	20	09	20	20	20	20	..	..	..	sível.. . ...
000000d0	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	.....

Figura 2.8 Arquivo texto(estego-objeto) em hexadecimal

Nota-se que existem sequências de espaços em branco (código 20 em hexa) logo após tabulação horizontal (código 09 em hexa) mostrando indícios de esteganografia no texto mesmo se o código secreto for compactado antes de ser inserido nas linhas. Isso acontece, pois o padrão se mantem, fazendo com que a técnica esteganográfica seja facilmente descoberta por alguma ferramenta esteganográfica que conhece esse método.

Softwares já existentes que analisam histograma e repetições, editores hexadecimais, ferramentas de checksum e outros, podem dar suporte ao esteganalista para análise de documentos suspeitos.

3 INTERFACE “INTERSNOW”

O projeto vai se chamar “InterSnow” em virtude da 'Interface' usar como base o aplicativo 'Snow' de licença GPL disponível em http://www.darkside.com.au/snow/.

Neste capítulo será visto o funcionamento do programa Snow, criado por Matthew Kwan, com suas características, classes, técnicas, métodos, exemplos de utilização e suas limitações. Depois, partindo do que já foi desenvolvido, a especificação do projeto de uma interface simples e amigável para o mesmo, cujo objetivo é expandir as funcionalidades da ferramenta e facilitar a compreensão e demostração da esteganografia e esteganálise em arquivos texto.

3.1 Funcionamento do aplicativo Snow

3.1.1 Características

O aplicativo Snow funciona em *modo prompt* de comando ou *console* de Comando e a técnica de codificação utilizada é a de deslocamento de carácter de fim de linha, que foi explicada no Capítulo 1.3.4, e se baseia no aproveitamento de espaços e tabulações para esconder os dados, já que, geralmente as linhas que contêm essas entidades não aparecem para o observador do documento. Isso permite que mensagens sejam camufladas no texto sem prejudicar sua representação visual. A aplicação foi escrita em Java por Matthew Kwan em abril de 1997 e que pode ser contatado pelo endereço eletrônico: mkwan@darkside.com.au.

O aplicativo Snow é executado em dois modos: Ocultação de mensagens e extração de mensagens. Durante a ocultação, a mensagem pode sofrer compressão ou criptografia facultativa. Para extrair os dados do texto o programa inverte esse processo.

Para fazer a compressão o Snow utiliza a codificação de Huffman por ser um algoritmo de compressão de baixa carga. Dependendo do texto o programa pode obter uma taxa de compressão de 20 a 40 por cento.

Quando a criptografia, o Snow trabalha com o algoritmo ICE. Um bloco de 64 bits que foi projetado pelo autor do Snow. Ele roda no modo de operação CFB que, segundo o autor, é ineficiente, porém, fornece uma melhor segurança quando mensagens diferentes utilizam a mesma senha.

3.1.2 Classes utilizadas

Snow.Class - Classe principal que recebe e analisa os argumentos digitados durante a chamada do programa Snow. São esses: os nomes dos arquivos de entrada e saída, as opções de ocultação, extração, compressão, criptografia, bem como a mensagem que

será escondida e a chave de encriptação. É nela, também, que são invocadas as classes e os métodos responsáveis pelo trabalho, propriamente dito, de codificação e decodificação dos dados de entrada.

Como o projeto desse trabalho é uma interface gráfica para o aplicativo Snow que antes era *janela Prompt* de Comando, essa e outras classes do projeto original sofreram alterações. Essas modificações serão descritas e ilustradas no Capítulo 4 deste trabalho.

SnowEncode.Class – Classe invocada para codificar ou decodificar o arquivo de entrada. Tudo vai depender se, quando o Snow for invocado, ele tiver um texto na linha de comando indicando que é uma mensagem logo após o o argumento "*-m*". Se sim, o programa entende que deve codificar esta mensagem no arquivo de entrada. Se não, o Snow apenas tenta extrair ou decodificar do arquivo de entrada os dados secretos.

SnowCompress.Class - Classe invocada quando se quer, além de codificar o arquivo de entrada uma mensagem, comprimi-lo. Para tanto, requer o uso da opção "*-C*" nos argumentos da invocação do programa Snow. Essa opção também vale quando se pretende decodificar uma entrada.

SnowEncrypt.Class - Classe invocada quando se quer, além de codificar o arquivo de entrada uma mensagem, encriptá-lo. Para tanto, requer o uso da opção "*-p*" nos argumentos da invocação do programa Snow seguido de um texto indicando o uso de uma senha. Esses dois comandos devem ser usados quando se pretende decodificar uma entrada que foi encriptada.

As demais classes como *SnowOutput.class, SnowCompress.class, IceKey.class* e *BitFilter.class*e são utilizadas para manipulação de bits, compressão e encriptação das mensagens.

3.1.3 Funcionamento

No modo *prompt* de comando ou *console* é digitado os comando como ilustrado na Figura 3.1. Onde a mensagem "*mensagem secreta*" será codificada tendo como arquivo de entrada "*Texto.txt"* com compressão e encriptado com a chave "*senha1234*". O texto resultante será gravado no arquivo "*saida.txt"*.

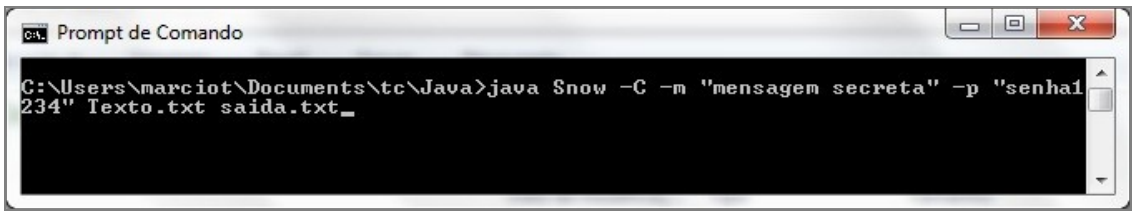


Figura 3.1 Comandos para esconder mensagens no Snow

Para extrair a mensagem devem ser digitados os comandos conforme ilustra a Figura 3.2 abaixo. Note que os comando de compactação "-C" e de encriptação "-p" devem estar presentes para que o Snow possa decodificar os dados e apresentar como saída a mensagem que foi escondia.

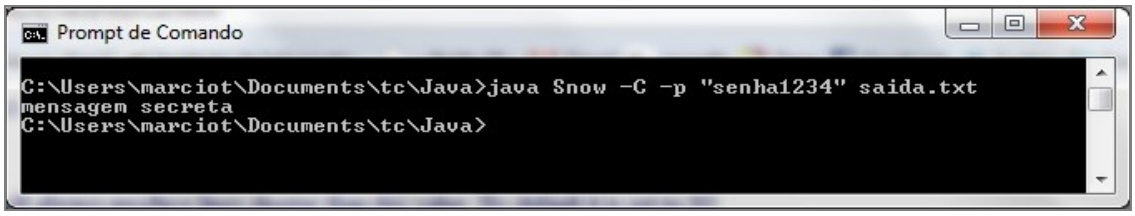


Figura 3.2 Comandos para extrair mensagem de arquivo codificado.

3.1.4 Limitações e Conclusão

Mesmo com um bom desempenho e eficácia na codificação e extração de mensagens em documentos textos, o aplicativo Snow fica engessado no modo *prompt* de comado onde, o usuário necessita digitar, além dos comandos DOS de deslocamento para chegar até a pasta do programa, toda a linha de caracteres para a invocação do aplicativo. Não esquecendo também que uma leitura prévia do manual do Snow, que é escrito em outro idioma, é recomendado para que se possa ter exito no processo como um todo.

Testes realizados utilizando como entrada arquivos de texto criados a partir do aplicativo BrOffice (extensão odt) e MicrosoftWord (extensão doc) mostraram que os arquivos gerados na saída do Snow perderam a formatação ou não puderam mais ser editados e nem recuperados. O que torna o programa ineficiente para arquivos de texto dessas duas plataformas.

Concluímos então, que falta ao programa Snow, além de uma compatibilidade aos principais programas editores de texto do mercado, uma interface mais amigável e intuitiva. Onde o usuário, ao alcance de um "*click*" do mouse, possa encontrar recursos que antes tinha que digitar em uma janela de *prompt*. Facilidades como: copiar e colar, escolher opções de saída, opções de técnicas, selecionar arquivos e tantas outras, façam parte de uma interface de janelas. Tornando, com isso, o processo de esteganografia e esteganálise mais claro para o usuário.

3.2 Especificação da Interface

Neste capítulo encontra-se a especificação da interface InterSnow. Embora o aplicativo não tenha uma complexidade significativa, a especificação segue o padrão UML (*Object Management Group)* que são notações gráficas que irão ajudar a transmitir as ideias deste projeto. Os diagramas das especificações foram desenvolvidas com o auxílio da ferramenta Astah.

3.2.1 Análise de requisitos

O objetivo dessa interface é prover ao usuário uma interface amigável e intuitiva de usar uma ferramenta de esteganografia em arquivos texto, possibilitando a escolha das técnicas mais conhecidas, bem como a criação de novas. Nesse sistema, deve ser possível gerar arquivos textos com mensagem secretas a partir de outros e também poder extraí-las. Os tipos de arquivos texto que essa interface deve manipular com sucesso são nos formatos TXT, RTF, HTML, XML e afins. Outra característica é que com essa ferramenta deve ser possível fazer análises de arquivos e pastas com o objetivo de encontrar textos secretos (esteganálise) e, neste caso, optar de decifrá-los, destruí-los ou não fazer nada.

3.2.2 Diagrama Hierárquico de Funções (DHF)

A partir da análise de requisitos e das funções já existentes na ferramenta Snow, foi criado o seguinte Diagrama Hierárquico de Funções:

- InterSnow
  - Modo Esteganografia
    - Alterar Técnica Esteganográfica
    - Esconder Mensagem
    - Extrair Mensagem
    - Gerenciar Técnicas de Esteganografia
      - ➔ Criar/Alterar/Excluir Técnicas
      - ➔ Tutorial
    - Limpar Dados
  - Modo Esteganálise
    - Escolha das Técnicas Esteganográficas
    - Escolher Pasta/Arquivo
    - Analisar Pasta/Arquivos
    - Gerar Relatório
    - Imprimir Relatório

3.2.3 Diagrama de casos de uso

O InterSnow não possui nenhum sistema de controle de acesso baseado em tipos de usuário. Desta forma, a Figura 3.3 mostra a configuração de funcionalidades que qualquer tipo de usuário tem acesso.

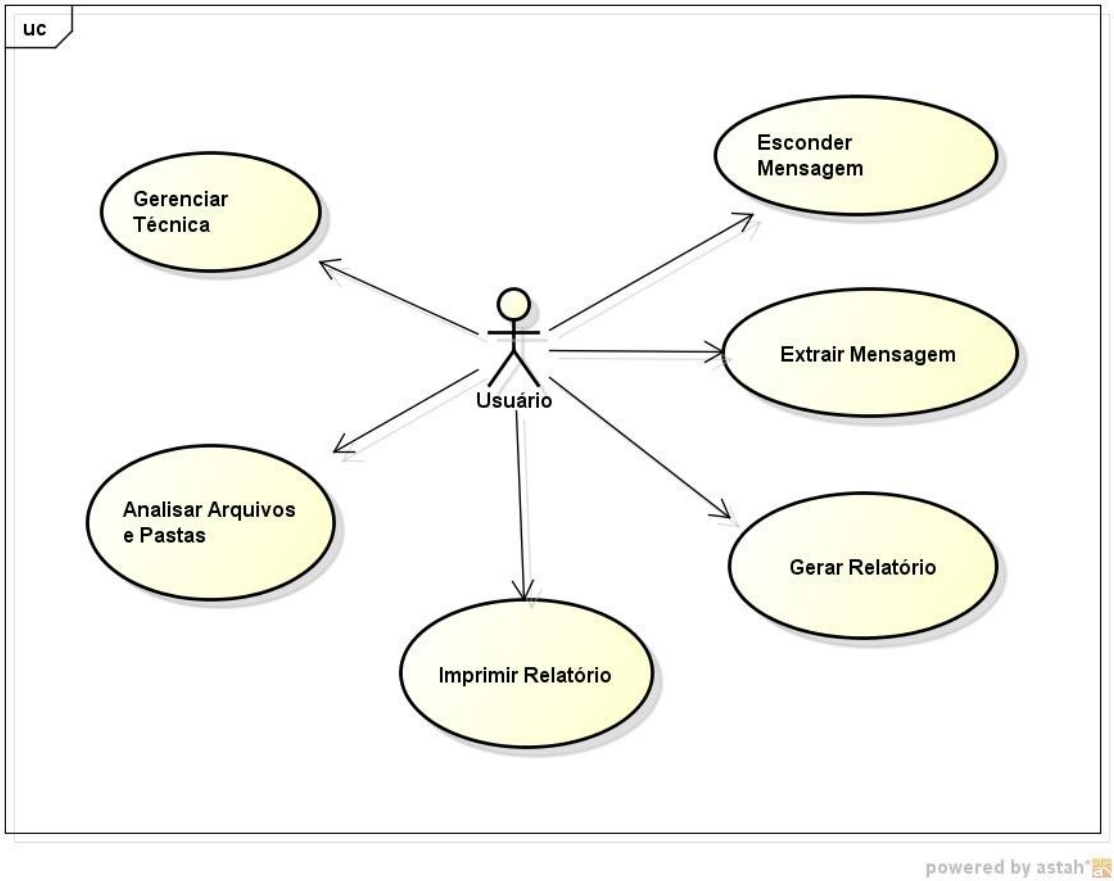


Figura 3.3 Diagrama de casos de uso do InterSnow

3.2.4 Descrição dos casos de uso

Aqui se encontram as descrições dos casos de uso do InterSnow, e tem como objetivo descrever toda a iteração que o usuário tem com o sistema. Adicionalmente, para permitir um melhor detalhamento, as descrições são acompanhadas por um fluxo básico dos eventos.

3.2.4.1 Gerenciar Técnicas Esteganográfica

Ator: Qualquer usuário do sistema.

Finalidade: Permitir ao usuário a escolha da técnica que deseja usar no modo Esteganografia ou Esteganálise.

Visão geral: Este caso se inicia quando o usuário escolhe qual técnica vai utilizar antes de esconder uma mensagem dentro de um arquivo texto. Ou, se sabe qual técnica foi utilizada, ter a opção de escolher uma específica antes extrair a mensagem que está escondida. No modo Esteganálise, o usuário pode ou não optar qual técnica utilizar para fins de performance. Todas as técnicas escolhidas no modo extrair ou esconder mensagem acompanha a opção de um tutorial explicativo.

Fluxo Básico:

1. Na página principal do aplicativo o ator escolhe o modo de trabalho (Esteganografia ou esteganálise).

2. O aplicativo, através da informação fornecida, altera a tela para o modo Esteganálise, caso tenha sido essa a escolha, senão no modo Esteganografia. (tela *default* do aplicativo)

3.2.4.2 Esconder mensagem

Ator: Qualquer usuário do sistema.

Finalidade: Permitir que o usuário esconda uma mensagem dentro de um arquivo texto.

Visão geral: Este caso se inicia quando o usuário deseja utilizar alguma técnica esteganográfica para esconder uma mensagem dentro de um arquivo.

Fluxo Básico:

1. O usuário seleciona o arquivo texto onde a mensagem será escondida. Esse processo pode ser efetuado através teclado digitando o caminho completo do arquivo ou através de um botão que aciona uma janela de navegação nas pastas do sistema operacional onde o aplicativo está executando.

2. O usuário escolhe através de menus e caixas de seleção as seguintes opções: (a) A técnica que deseja utilizar e se a mensagem será compactada no arquivo, se o arquivo de saída terá senha de acesso e se a saída será em arquivo ou na tela principal do aplicativo.

3. O usuário preenche um campo com o texto que deseja esconder ou seleciona um arquivo que contenha o texto através de um botão que aciona uma janela de navegação nas pastas do sistema operacional onde o aplicativo está executando. .

4. O usuário clica no botão que aciona um comando para que o aplicativo efetue a operação.

5. A ferramenta exibe uma mensagem indicando o sucesso na operação.

3.2.4.3 Extrair mensagem

Ator: Qualquer usuário do sistema.

Finalidade: Permitir que o usuário extraia de um arquivo texto uma mensagem que ali foi escondida através de alguma técnica de esteganografia.

Visão geral: Este caso se inicia quando o usuário em posse de um *estego-objeto* deseja obter a mensagem que ali está escondida.

Fluxo Básico:

1. O usuário seleciona o arquivo texto onde a mensagem está escondida. Esse processo pode ser efetuado através teclado digitando o caminho completo do arquivo ou através de um botão que aciona uma janela de navegação nas pastas do sistema operacional onde o aplicativo está executando.

2. O usuário escolhe através do menu de seleção qual técnica deseja utilizar para extrair a mensagem do arquivo. A técnica escolhida deve ser a mesma que foi utilizada para esconder a mensagem para que a extração ocorra sem erros. Ou seja, o usuário deve saber de antemão qual método de esteganografia que foi usado antes de extrair a mensagem.

3. O usuário clica no botão que aciona um comando para que o aplicativo efetue a operação de extração.

4. A ferramenta exibe uma mensagem indicando o sucesso na operação.

3.2.4.4 Gerenciar técnica de esteganografia

3.2.4.4.1 Alterar/Excluir/Visualizar as Técnicas

Ator: Qualquer usuário do sistema.

Finalidade: Permitir que o usuário modifique e visualize as técnicas e tutoriais de esteganografias utilizadas no aplicativo.

Visão geral: Este caso se inicia quando o usuário deseja alterar alguma configuração das técnicas utilizadas no aplicativo permitindo visualizar e editar o código dos métodos, alterá-los e recompilá-los, incluindo manuais e tutoriais explicativos dos mesmos.

Fluxo Básico:

1. O usuário seleciona a opção "*Gerenciar Técnicas"* no menu horizontal da janela principal do aplicativo.

2. O sistema procura as técnicas que estão sendo utilizadas e abre uma janela onde o usuário consegue visualizar esses dados.

3. O usuário seleciona a técnica desejada.

4. O sistema apresenta uma tela onde o usuário pode excluir, editar e visualizar as técnicas e tutoriais das mesmas no sistema.

5. Caso tenha sido efetuado alguma alteração nas técnicas ou tutoriais, o usuário precisa clicar na opção "salvar".

6. A ferramenta exibe uma mensagem indicando o sucesso na operação.

3.2.4.4.2 Inserir/Criar novas Técnicas

Ator: Qualquer usuário do sistema.

Finalidade: Permitir que o usuário insira ou crie técnicas de esteganografias.

Visão geral: Este caso se inicia quando o usuário deseja acrescentar novas técnicas para serem utilizadas no aplicativo. Permitindo que o usuário agregue novas funcionalidades ao programa sem precisar recorrer a ferramentas externas para recompilar o código, por exemplo.

Fluxo Básico:

1. O usuário seleciona a opção "*Gerenciar Técnicas"* no menu horizontal da janela principal do aplicativo (Modo Esteganografia).

2. O sistema apresenta uma tela onde o usuário escolhe a opção de "criar uma nova técnica".

3. O sistema apresenta uma outra tela onde o usuário pode carregar o arquivo do código que será adicionado no sistema ou inserir o texto diretamente em um campo.

5. Depois de carregar o novo código o usuário precisa clicar na opção “salvar”.

6. A ferramenta exibe uma mensagem indicando o sucesso na operação.

3.2.4.5 Limpar os dados

Ator: Qualquer usuário do sistema.

Finalidade: Permitir que o usuário limpe os dados utilizados e gerados na última consulta.

Visão geral: Este caso se inicia quando o usuário deseja efetuar um novo processo como esconder ou extrair mensagem. Caso, o aplicativo já tenha sido usado, precisa ser reiniciado para que suas variáveis, campos e *buffers* sejam limpos para não influenciar no novo processo.

Fluxo Básico:

1**.** O usuário, na tela principal do aplicativo, clica no botão “Limpar”.

2. O sistema esvazia todas as suas variáveis, *buffers* e campos da tela.

3. A ferramenta exibe uma mensagem indicando o sucesso na operação.

3.2.4.6 Analisar pastas ou arquivos

Ator: Qualquer usuário do sistema.

Finalidade: Permitir que o usuário procure por indícios de esteganografia em arquivos texto.

Visão geral: Este caso se inicia quando o usuário deseja verificar um arquivo ou varrer uma ou mais pastas dos sistema suspeitas de conter algum arquivo com mensagem escondida.

Pré-Requisitos: O usuário deve estar no modo Esteganálise.

Fluxo Básico:

1. O usuário clica no botão “Arquivo”ou digita no campo caminho o arquivo ou pasta que deseja que o aplicativo verifique.

2. No caso de clicar no botão “Arquivo”, o aplicativo abre uma janela de navegação nas pastas do sistema operacional em uso.

3. O usuário seleciona o arquivo ou pasta e aperta o botão “Pronto”.

4. O aplicativo retorna para a tela anterior.

5. O usuário escolhe uma ou mais técnicas existentes no aplicativo para efetuar a verificação.

6. O usuário clica no botão “Verificar”.

7. O aplicativo faz uma varredura nos arquivos ou pastas selecionadas e produz um relatório de saída com o resultado obtido em uma nova janela.

3.2.4.7 Geração de Relatório

Ator: Qualquer usuário do sistema.

Finalidade: Permitir que o usuário gere arquivos com relatórios das operações de esteganálise efetuadas.

Visão geral: Este caso se inicia quando o usuário, após analisar arquivos ou pastas em busca de indícios de esteganografia, resolve gerar um arquivo com o relatório estatístico detalhado do que já foi verificado pelo aplicativo.

Pré-Requisitos: O usuário deve ter usado a ferramenta de verificação e busca do modo esteganálise pelo menos uma vez. Além de estar na janela do modo Esteganálise.

Fluxo Básico:

1. O usuário clica na opção "Relatórios".

2. O sistema abre uma janela contendo um histórico dos arquivos já verificados pelo aplicativo.

3. O usuário seleciona na lista a opção desejada, escolhe um nome para o aquivo de saída e clica em "Gerar Relatório".

4. O sistema gere um relatório em formato PDF e exibe uma nova janela com uma mensagem indicando sucesso na operação.

3.2.4.7 Impressão de Relatórios

Ator: Qualquer usuário do sistema.

Finalidade: Permitir que o usuário imprima os relatórios das operações de esteganálise efetuadas.

Visão geral: Este caso se inicia quando o usuário, após analisar arquivos ou pastas em busca de indícios de esteganografia, resolve imprimir o relatório estatístico detalhado do que já foi verificado pelo aplicativo.

Pré-Requisitos: O usuário deve ter usado a ferramenta de verificação e busca do modo esteganálise pelo menos uma vez.

Fluxo Básico:

1. O usuário clica na opção do menu "Arquivo" - "Imprimir" ou no ícone da impressora disponível na tela no aplicativo.

2. O sistema abre uma janela contendo um histórico dos arquivos já verificados pelo aplicativo.

3. O usuário escolhe na lista o relatório desejado e clica em "Pronto".

4. O aplicativo em imprime na impressora padrão do sistema o relatório escolhido pelo usuário e gera uma mensagem que os dados forma enviados com sucesso para impressora.

3.2.5 Modelo de Classes

A seguir encontra-se o diagrama que representa o modelo de classes do aplicativo InterSnow mostrando apenas as classes com seus atributos e métodos.

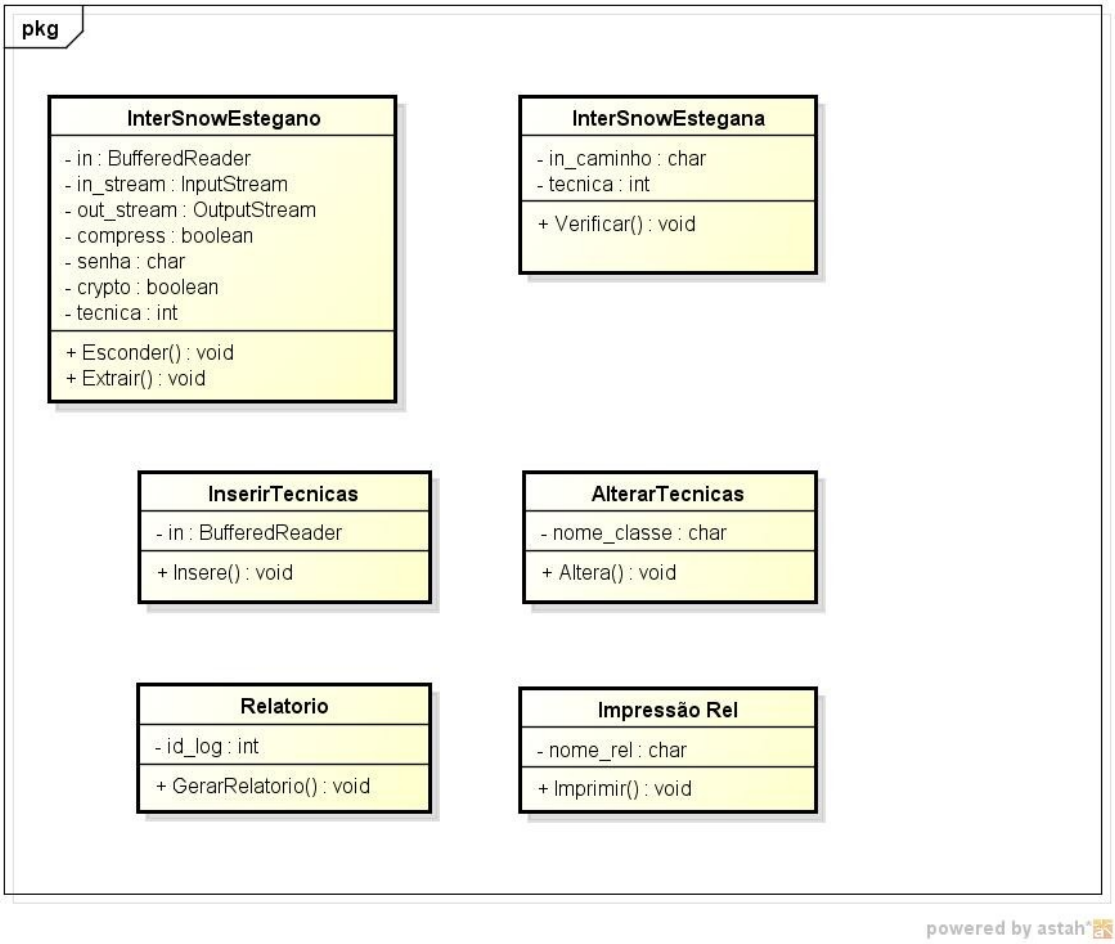


Figura 3.4 Diagrama de Classes do aplicativo com seus atributos e métodos

4 DESENVOLVIMENTO DO PROTÓTIPO

Neste capítulo, será apresentado as informações sobre a construção de um protótipo do aplicativo proposto neste trabalho. A tecnologias utilizadas, *layout,* telas*,* códigos, testes e operações. Por fim, uma análise sobre a inserção de mensagens com tamanhos diferentes em arquivos de formatos e tamanhos distintos.

4.1 Tecnologias utilizadas

Para o desenvolvimento do protótipo, foram utilizadas as seguintes tecnologias no sistema Windows XP versão 5.1 executando em x86:

- Programas:
  - NetBeans IDE 6.9.1 (Build 201007282301)
  - Notepad ++ Versão 5.7 (UNICODE)
- Linguagens
  - Java 1.6.0_24;
  - Java HotSpot(TM) Client VM 19.1-b02
- Códigos
  - Snow.java
  - SnowEncode.java

4.1.1 Por que utilizar Java

Basicamente, optou-se pela linguagem Java para desenvolver esse aplicativo pelas seguintes razões:

1) Já existir uma parte desenvolvida, porém, em *modo prompt* de comando ou *console* de uma ferramenta chamada "Snow" nesta linguagem. Neste caso o desafio seria estender esse código e adaptá-lo para uma interface mais iterativa e intuitiva com o objetivo de contribuir ao que já foi feito nesta área.

2) O aplicativo a ser de desenvolvido não exigia alto poder de processamento ou velocidade de resposta. Caso contrário, a linguagem Java não seria a linguagem recomendada, pois tem um desempenho menor em comparação com outras linguagens como C++, por exemplo.

3) Facilidade para a implementação da interface gráfica já que Java possui as duas maiores IDEs do mercado (Eclipse e NetBeans) com vasta documentação, ambas gratuitas. Fácil acesso aos acervos das documentações incluindo tutoriais e vídeos aula postados na rede mundial de computadores.

4.1.2 Apresentação das páginas e campos do protótipo

A figura 4.1 mostra um exemplo da página principal do protótipo InterSnow. Nela qualquer usuário tem acesso aos campos e botões que vão configurar a ferramenta para ativar os processos de "Esconder" ou "Extrair" mensagens em arquivos texto. O objetivo principal do *layout* desta tela, bem como da distribuição dos campos e formulários, é facilitar a visualização do processo acontecendo sem sobreposição de telas e menus, priorizando, desta forma, a distribuição em tela única de todas as ferramentas necessárias.

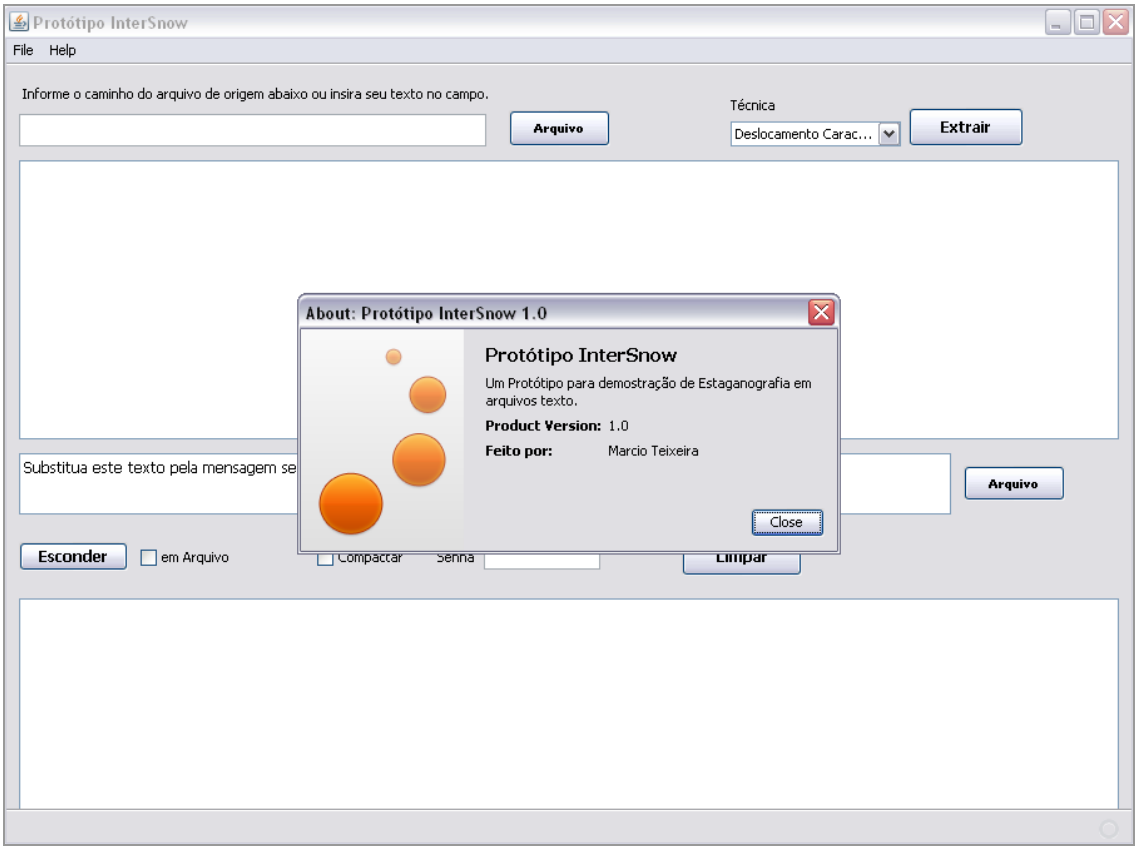


Figura 4.1 tela Principal InterSnow

4.1.2.1 Seleção do arquivo de entrada (recipiente)

A figura 4.2 mostra a tela com o campo onde o usuário deve inserir o caminho completo onde se encontra o arquivo que irá conter a mensagem. Para tanto, basta digitar no espaço indicado na janela a localização ou, caso prefira, usar o botão "Arquivo" para navegar pelas pastas do sistema operacional até o local do arquivo. Após esse passo, o aplicativo carrega o conteúdo do arquivo no painel superior da tela principal para que o usuário possa visualizar os dados originas antes da inserção da mensagem ou, quando for o processo inverso, os dados antes a extração da mensagem.

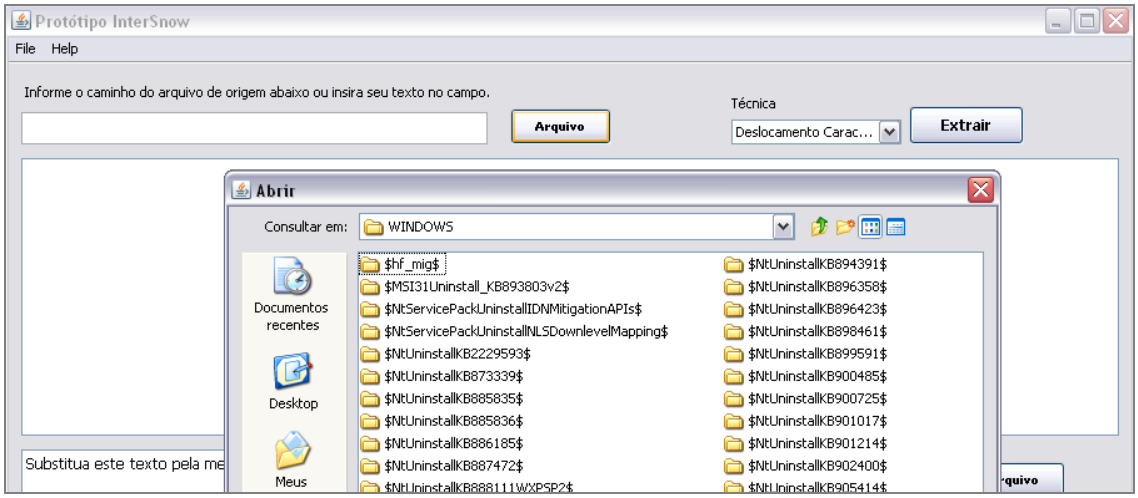


Figura 4.2 Seleção arquivo de entrada. (Recipiente)

4.1.2.2 Seleção da técnica de esteganografia e mensagem

A figura 4.3 mostra a tela onde o usuário seleciona a técnica a ser utilizada para esconder a mensagem no arquivo. A título de ilustração foram colocadas todas as técnicas citadas neste trabalho, porém no protótipo, a única que foi implementada é a de 'deslocamento de caractere de fim de linha'. Na figura 4.3 também mostra o campo onde deve ser escrito a mensagem secreta que será escondida no arquivo ou, através do botão 'Arquivo' ao lado direito desta área, indicado um arquivo que contenha um texto que servirá como mensagem. Caso o processo seja de extração, esse campo deve ficar em branco.

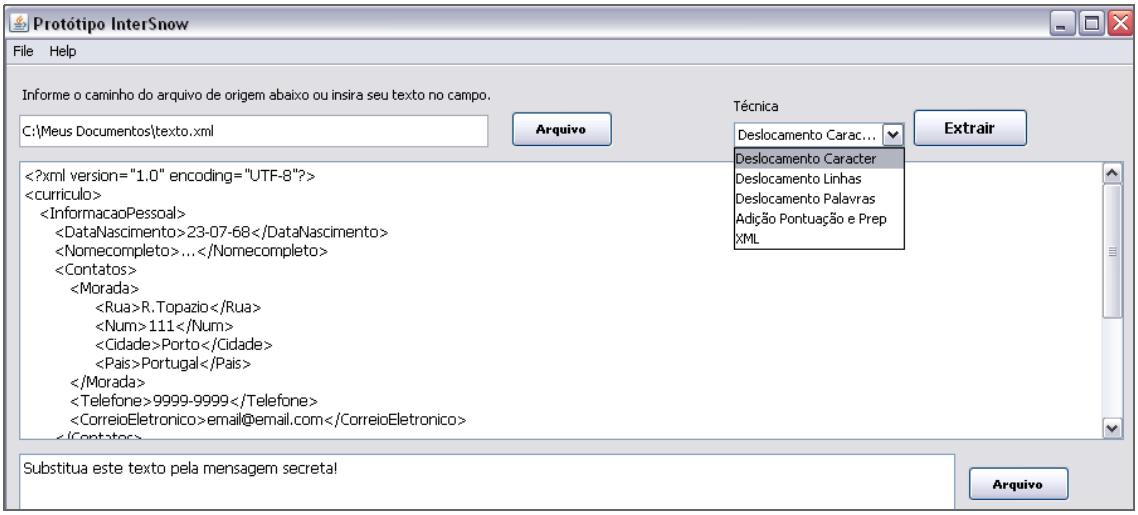


Figura 4.3 Seleção da técnica e mensagem

4.1.2.3 Esconder mensagem em arquivo

A figura 4.4 mostra a tela do aplicativo depois que uma mensagem foi escondida utilizando como *recipiente* um arquivo texto de nome 'texto.xml' e como mensagem a frase: “mensagem secreta!”. A imagem também mostra o conteúdo do arquivo de saída (*estego-objeto*), ou seja, depois que a mensagem foi inserida. Note que ao selecionar algumas das linhas contidas no painel inferior da tela, o final das as 8 (oito) primeiras estão preenchidas por algo que não aparece no editor do aplicativo. No caso, são os

espaços em branco e tabulações que são a codificação que vai guardar o código da mensagem que se quer esconder.

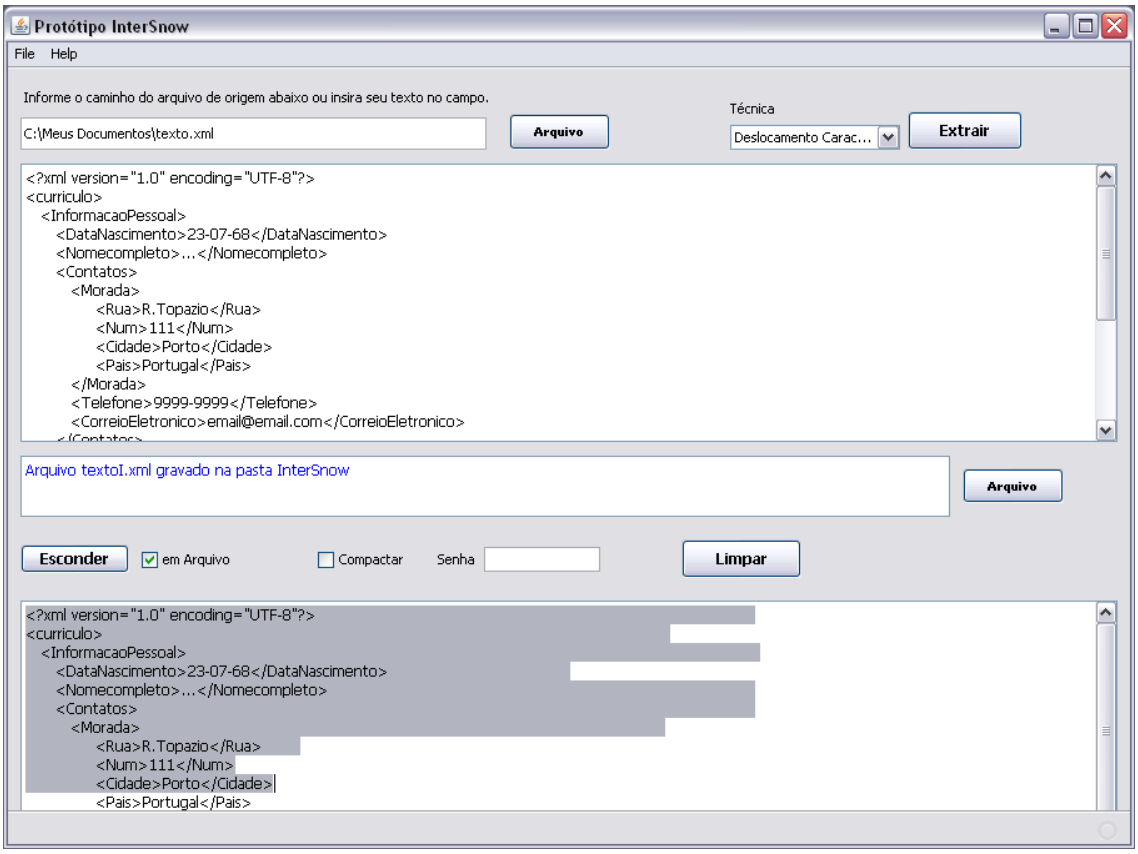


Figura 4.4 Tela do processo de esconder mensagem em arquivo

Por simplicidade o protótipo está configurado para efetuar o processo de esteganografia na própria tela sem a necessidade de salvar a saída em outro arquivo (*estego-objeto*). Caso o usuário opte em, além disso, criar esse arquivo que terá a mensagem secreta, deve selecionar a opção "em Arquivo" antes de Esconder a mensagem. Neste caso, o protótipo no final do processo exibe uma mensagem na cor azul no campo central informando: "Arquivo <*nome*>I.*ext* gravado na pasta InterSnow" onde '*nome*' e '*ext*' são, respectivamente o nome e a extensão do arquivo original.

4.1.2.3.4 Processo de extração de mensagem

A Figura 4.5 mostra a tela do protótipo InterSnow depois que uma mensagem foi extraída de um arquivo (*estego-objeto)* através da técnica 'deslocamento de caractere de fim de linha'. Para que esse processo se efetue, o usuário deve informar o caminho do arquivo que contém essa mensagem e clicar no botão "Extrair". Caso não ocorra erros, será informado no campo central do aplicativo um texto na cor vermelha com a mensagem que foi extraída. O processo de extração se concentra basicamente na parte superior da tela do protótipo nos campos de caminho, edição e mensagem ficando os demais campos sem alteração.

Assim como na Figura 4.4 foi selecionado, a título de demonstração, as primeiras linhas do arquivo de entrada, onde pode se verificar que no final de cada linha existe "algo" que a edição não exibe. Ou seja, a codificação da mensagem escondida.

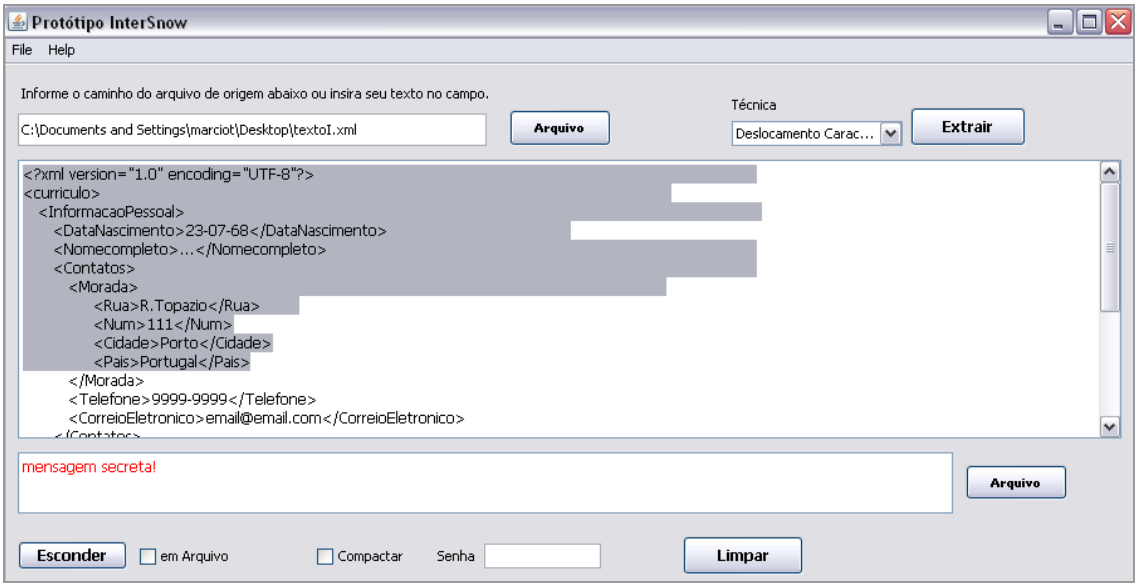


Figura 4.5 Extração de mensagem

4.1.3 Modificações no código original "Snow"

Para adaptar aplicativo "Snow", que é executado em *modo prompt* de comando ou *console*, com o protótipo da interface gráfica proposta neste trabalho, foi necessário efetuar algumas modificações no código fonte do "Snow" para permitir a comunicação entre essas duas partes.

As alterações mais relevantes foram: a retirada de 118 linhas que faziam parte da análise dos argumentos de entrada (*parsing*) que eram inseridos em linha de comando (Basicamente, um comando "switch" com vários "cases", um para cada argumento de entrada - vide anexo A deste trabalho); E, consequentemente, as funções que fazem parte da classe principal do aplicativo Snow que efetuam os processos de esconder e extrair as mensagem dos textos já que elas utilizam esses argumentos. A Figura 4.6 e Figura 4.7 mostram uma versão simplificada do antes e depois dos códigos destas funções.

```
    // Entry point to the program.
  public static void  main (String argv[]) {
    if (!parse_args (argv)) {
    System.err.print ("Usage: Snow.class [-C][-Q][-S]");
    System.err.print ("[-p passwd][-l line-len]");
    System.err.println (" [-f file][-m message]");
    System.err.println ("\t\t\t\t\t[infile [outfile]]");
    System.exit (1);
    }
```

Figura 4.6 Snow original

```
    // Entry point to the program.
Snow (
            BufferedReader  in, // Arquivo de entrada
            InputStream in_stream,  //Mensagem secreta
            OutputStream out_stream //Arquivo de saida

    ) {
       stream_in = in;
       stream_message = in_stream;
       stream_out = out_stream;
```

Figura 4.7 Snow adaptado

No código do Snow original, os argumentos eram digitados junto com a chamada do programa em linha de comando e processados como '*Strings*' através da variável '*argv*' que fica dentro da função principal. Já no código do Snow adaptado, esses argumentos agora são fornecidos pelos recursos da interface gráfica (campos, formulários, *flags*, etc) e a função principal do programa não faz mais parte da classe Snow e sim da classe que chama a interface gráfica.

4.2 Análise

Com as informações dos tamanhos dos arquivos coletadas após a inserção das mensagens secretas, que chamaremos de Estego-Objeto, foi possível montar a Tabela 4.1 que mostra o comparativo entre arquivos de tamanhos e formatos diferentes, que são os recipientes, versus arquivos com mensagens de texto com tamanhos diferentes.

Foi utilizado para essa análise os seguintes arquivos:

Recipiente1 - Arquivo 'html' de 3.305 bytes.

Recipiente2 - Arquivo 'txt' de 186.654 bytes.

Recipiente3 - Arquivo 'rtf' de 1.056.192 bytes.

Mensagem 1 – Arquivo 'txt' de 64 bytes.

Mensagem 2 – Arquivo 'txt' de 432 bytes.

Mensagem 3 – Arquivo 'txt' de 805 bytes.

Tabela 4.1: Comparativo

	Tamanho Recipiente em bytes	Tamanho Mensagem em bytes	Tamanho do Estego-Objeto em bytes	Percentual (%) de aumento bytes
Recip1/Mensg1	3.305	64	3.982	20,48
Recip1/Mensg2	3.305	432	8.199	148,07
Recip1/Mensg3	3.305	805	12.564	280,15
Recip2/Mensg1	186.654	64	187.162	0,27

Recip2/Mensg2	186.654	432	191.116	2,39
Recip2/Mensg3	186.654	805	195.187	4,57
Recip3/Mensg1	1.056.192	64	1.056.881	0,07
Recip3/Mensg2	1.056.192	432	1.060.733	0,43
Recip3/Mensg3	1.056.192	805	1.064.691	0,8

De acordo com as informações da Tabela 4.1 pode-se afirmar que quanto maior o arquivo recipiente menor é o impacto que ele sofre em tamanho quando informações nele são inseridas. É uma boa estratégia quando o assunto é Esteganografia onde o objetivo é justamente não chamar a atenção. Assim, um arquivo com uma quantidade pequena de texto mas com um tamanho em bytes desproporcional em virtude da quantidade informação que ali foram inseridas, poderia gerar suspeita. Logo, quando maior o recipiente, mais capacidade ele terá de carregar mensagens e a influencia destas no Estego-Objeto é proporcional ao recipiente.

Essas influencias que o tamanho do recipiente e das mensagens tem no Estego-objeto podem ser melhor observadas na inclinação da reta amarela nos gráficos mostrados na Figura 4.8 onde os recipientes 1 e 2 que foram submetidos a inserção de mensagens utilizando a técnica esteganográfica de 'deslocamento de caractere de fim de linha'. Os gráficos da Figura 4.8 foram construídos com base nas informações da Tabela 4.1 representando os **recipientes 1** de 3.305 bytes e o **recipiente 2** de 186.654 bytes em que foram inseridos menagens de 64, 432 e 805 bytes.

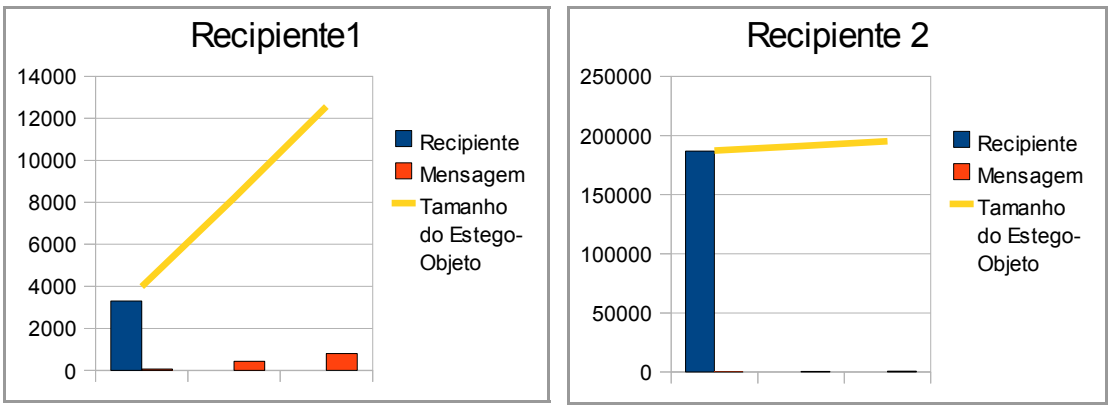


Figura 4.8 Gráficos Recipiente 1 e Recipiente 2

5 CONCLUSÃO

Neste trabalho foram apresentados não só os conceitos sobre a Esteganografia e Esteganálise com algumas das diversas técnicas desenvolvidas através dos anos, mas também, procurou apresentar algumas ferramentas já criadas dentro desse contexto e adaptar para uma nova o que já existia e que podia era considerado difícil de operar. Para tanto, foi feito a especificação de uma ferramenta que procura contemplar e criação de um protótipo.

Em termos de pesquisa histórica, foi encontrado muito conteúdo curioso e motivador, principalmente na rede mundial de computadores, mas que teve de ser filtrado, resumido e direcionado já que a intensão desse trabalho não era apenas essa. Porém, fica o registro de que muito já foi inventado pelo homem com o intuito de esconder mensagens, porém, optou-se em não sobrecarregar este documento com esses registros.

A esteganografia usa de muita criatividade na sua aplicação, uma fonte que podemos considerar inesgotável e que nos leva a crer que novas técnicas podem ser criadas, alteradas, combinadas, etc. Isso torna o trabalho de esteganálise mais difícil de ser implementado, porém, desafiador. Em arquivos texto, onde, na sua maioria, é aproveitado aqueles espaços existentes entre linhas e caracteres, esse trabalho não parece ser tão complicado para um programa de computador. O difícil é descobrir onde monitorar dentro da quantidade enorme objetos que estão disponíveis nos dias atuais e que podem ser usados para esse fim.

No desenvolvimentismo do protótipo, nem tudo que foi especificado foi implementado, até porque muitas ideias fogem do escopo principal do trabalho mas que seriam interessantes serem continuadas para que contribuam com o entendimento e estudo da esteganografia. O maior desafio dessa implementação foi a adaptação e um código escrito por outra pessoa para deixa-lo com uma nova aparência. Para que essa tarefa fosse executada, a pesquisa de soluções juntamente com o estudo dos códigos fontes e da linguagem de programação foram essenciais. Nesta momento, os conceitos adquiridos durante o curso de formação foram de vital importância e tiveram que ser colocados em pratica.

REFERÊNCIAS

[AMIN 2003] AMIN, M. M.; IBRAHIM, S.; SALLEH, M.; KATMIN , M. R. **Information Hiding Using Steganography** 2003 Department of Computer System & Communication Faculty of Computer Science and Information System

[CHIRIGATI 2009] CHIRIGATI, F. S.; KIKUSHI, R. S. A.; GOMES, T. L. **Esteganografia** Universidade do Rio de Janeiro. Disponível em <http://www.gta.ufrj.br/grad/09_1/versao-final/stegano/esteganalise.htm>. Acesso em Maio de 2011.

[CUMMIS 2004] CUMMIS, J; DISKIN, P; LAU, S.; PARLETT, R. **Steganography And Digital Watermarking.** School of Computer Science, The University of Birmingham.

[DUARTE 2008] DUARTE, O. C. M. B. **Esteganografia** Universidade Federal do Rio de Janeiro. UFRJ Disponível em <http://www.gta.ufrj.br/grad/08_1/estegano/TcnicasModernas.html>. Acesso em Setembro de 2011.

[GUENTER 2010] GUENTER, M. S. **Sistema de gerenciamento de projetos de pesquisa COMPESQ/INF** 2010 Trabalho de Graduação (Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.

[KIPPER 2004] KIPPER, Greg. **Investigator's Guide to Steganography.** Boca Raton London New York Washington, D.C. 2004.

[MAKINO] S. Inoue, K. Makino, I. Murase, O. Takizawa, T. Matsumoto and H.Nakagawa, **A Proposal on Information Hiding Methods using XML**, Disponível em <http://takizawa.gr.jp/lab/nlp_xml.pdf.> Acesso em junho 2011.

[PETRI 2004] PETRI, Marcelo. **Esteganografia**. 2004. 62 f. Trabalho de conclusão (Bacharelado em Sistemas de Informação) - Instituto Superior Tupy Joinville.

[POLLON 2007] POLLON, Vanderlei. **Esteganografia: A arte de ocultar arquivos dentro de arquivos –** 2007. Trabalho Individual (2º Seminário de Software Livre Tchelinux - Edição Porto Alegre) Disponível em < www.pollon.org >. Acesso em: abril. 2010.

[RAJGURE] RAJGURE, V. A.; GAIKWAD, V. T. **Steganography and Steganalysis: Different Approaches** Department of Computer Science and Engineering, Sipna College of Engg & Tech, Amravati Disponível em <http://www.interscience.in/IJESS_Vol1_Iss1/paper_32.pdf> Acesso em Outubro 2011.

[SELBACH 2010] SELBACH, F. C. **Sistema Web para Auxílio em Pesquisas e Gerência de Dados Clínicos** 2010 Trabalho de Graduação (Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.

[WAYNER 2003] Wayner, P; **SpamMimic** 2003, Disponível em <http://www.spammimic.com>. Acesso em Maio de 2011.

[WAYNER 2009] WAYNER, Peter. **Disappearing Cryptography – Information hiding: Steganography & Watermarking,** 3rd ed. Burlington, MA 01803, USA Elsevier Inc.