UNIVERSIDADE REGIONAL DE BLUMENAU

CENTRO DE CIÊNCIAS EXATAS E NATURAIS

CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

PROTÓTIPO DE SOFTWARE PARA OCULTAR TEXTOS COMPACTADOS EM ARQUIVOS DE ÁUDIO UTILIZANDO ESTEGANOGRAFIA

ANDRÉ KOBUSZEWSKI

BLUMENAU
2004

2004/2-06

ANDRÉ KOBUSZEWSKI

PROTÓTIPO DE SOFTWARE PARA OCULTAR TEXTOS COMPACTADOS EM ARQUIVOS DE ÁUDIO UTILIZANDO ESTEGANOGRAFIA

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Ciência da Computação — Bacharelado.

Prof. Francisco Adell Péricas

BLUMENAU
2004

2004/2-06

PROTÓTIPO DE SOFTWARE PARA OCULTAR TEXTOS COMPACTADOS EM ARQUIVOS DE ÁUDIO UTILIZANDO ESTEGANOGRAFIA

Por

ANDRÉ KOBUSZEWSKI

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente:	Prof. Francisco Adell Péricas – Orientador, FURB
Membro:	Prof. Sérgio Stringari – FURB
Membro:	Prof. Mauro Marcelo Mattos – FURB

Blumenau, 10 de dezembro de 2004

Dedico este trabalho aos meus pais e à minha família, que esteve sempre ao meu lado dando apoio e incentivo para a conclusão desta importante fase de minha vida.

"A vida só pode ser compreendida olhando-se para trás; mas só pode ser vivida olhando-se para a frente."

Soren Kierkegaard

AGRADECIMENTOS

Agradeço primeiramente a Deus, que me deu forças para subir mais este degrau de vida.

À minha família, principalmente aos meus pais, José e Marit, pelo amor, dedicação e confiança, estando presentes em todos os momentos na caminhada para a conclusão deste curso.

Ao meu orientador, Francisco Adell Péricas, por toda sabedoria, apoio e incentivo para a conclusão deste trabalho.

A todos os professores, colegas e amigos que de alguma forma participaram e colaboraram na conclusão de mais esta etapa em minha vida.

RESUMO

Este trabalho apresenta a especificação e implementação de um protótipo de software para incluir mensagens de texto compactadas em arquivos de áudio digitais, utilizando compactação de dados em conjunto com esteganografia. Utilizou-se do algoritmo de Huffman, técnica estatística de compressão de dados para a compactação de texos, em conjunto com o método de esteganografia *Last Significant Bit* (LSB), para ocultar mensagens compactadas em arquivos de áudio no formato *Wave.*

Palavras chaves: Esteganografia; Segurança de Dados; Criptografia; Compactação de Dados.

ABSTRACT

This work presents the specification and implementation of a software prototype for including compacted text messages in audio digital files, using data compression and steganography. It was used the Huffman code, a statistic technique for data compression, used to compact texts, together with the method Last Significant Bit (LSB) of steganography, to hide compacted messages in wave format audio files.

Key-Words: Steganography; Data Security; Cryptography; Data Compression.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

O espetacular crescimento de sistemas de multimídia interligados nas redes de computadores nos últimos anos tem apresentado um enorme desafio nos aspectos tais como propriedade, integridade e autenticação de dados digitais. A busca de mecanismos adequados para proteção das informações cresce na proporção da crescente disponibilidade de sistemas de telecomunicações, estes cada vez mais sofisticados e abrangentes, ressaltando a importância de sistemas de segurança.

A segurança está relacionada à necessidade de proteção contra o acesso ou manipulação, intencional ou não, de informações confidenciais por elementos não autorizados, e a utilização não autorizada do computador ou de seus dispositivos periféricos. A necessidade de proteção deve ser definida em termos das possíveis ameaças e riscos dos objetivos de uma organização, formalizados nos termos de uma política de segurança (SOARES, 1995).

Um dos métodos mais utilizados e eficientes para a transferência de informações, sem que haja a possibilidade de comprometimento do sigilo, é através da codificação ou cifragem das informações, tornando-as incompreensíveis, conhecida como criptografia.

A criptografia é uma ciência que estuda os princípios, meios e métodos para proteger a confidencialidade das informações através da codificação ou processo de cifração e que permite a restauração da informação original através do processo de decifração. Largamente aplicada na comunicação de dados. Esta ciência utiliza-se de algoritmos matemáticos e da criptoanálise para conferir maior ou menor proteção de acordo com sua complexidade e estrutura de desenvolvimento.

Outro método utilizado, a esteganografia, é uma antiga arte utilizada para embutir mensagens secretas em mensagens aparentemente inofensivas, de forma que previna sua detecção por parte de terceiros. Diferentemente da criptografia, a qual procura tornar a mensagem incompreensível, a intenção na esteganografia é esconder a existência da mensagem, mas o que não impede a utilização das duas técnicas em conjunto, aumentando assim segurança e dificultando ainda mais um possível ataque.

A esteganografia inclui uma vasta disposição de técnicas para ocultar mensagens em uma variedade de meios. Entre esses métodos estão tintas invisíveis, micropontos, assinaturas

digitais, canais escondidos e comunicação de espectro espalhado. Hoje, graças à tecnologia moderna, a esteganografia pode ser utilizada em textos, imagens, sons, sinais, e outros.

Segundo Tomás (2002), devido ao alcance do sistema auditivo humano, ocultar dados em sinais de áudio é especialmente desafiador. Os métodos de ocultamento de dados em áudio mais conhecidos são: codificação do *bit* menos significativo (LSB – *Last Significant Bit*), codificação de fases, *spread spectrum* e ocultação de dados no eco.

A fim de aumentar a capacidade do tamanho da mensagem a ser oculta, pode-se adicionar alguma técnica para compressão de dados. A compressão de Huffman é uma técnica de codificação estatística que diz respeito ao uso de um código curto para representar símbolos comuns e códigos longos para representar símbolos pouco freqüentes.

Como sugestão de extensão do trabalho de conclusão de curso de Jascone (2003), que utiliza a esteganografia para ocultar mensagens criptografadas em arquivos de imagens, neste trabalho foi utilizada uma técnica de esteganografia, a fim de desenvolver um protótipo de software para ocultar mensagens em arquivos de áudio. Foi utilizada também a compressão de Huffman para efetuar a compressão dos dados a serem ocultos no arquivo, possibilitando um aumento no tamanho da mensagem a ser incluída.

Um trabalho semelhante foi desenvolvido por John (2004), a qual desenvolveu um sistema capaz de ocultar dados contidos em arquivos de qualquer formato em arquivos de áudio no formato *wav*. Este projeto foi desenvolvido na linguagem C#, utilizando o método de esteganografia LSB em conjunto com uma técnica para codificação onde são utilizados *bytes* dispersos dentro do arquivo de áudio, de acordo com um arquivo chave escolhido pelo usuário. Para extrair a informação do arquivo de áudio é necessário além do arquivo de áudio contendo a mensagem, o arquivo chave, efetuando assim a extração da informação de forma correta.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho foi desenvolver um protótipo de software para ocultar textos em arquivos de áudio que utiliza técnicas de esteganografia em conjunto com a compressão de Huffman.

Os objetivos específicos do trabalho são:

a) compactar um texto utilizando o algoritmo de Huffman;
b) esconder o texto comprimido em um arquivo de áudio, utilizando o método LSB;
c) extrair o texto do arquivo de áudio;
d) descompactar o texto possibilitando sua visualização.

1.2 ESTRUTURA DO TRABALHO

Este está subdividido em capítulos que serão explicitados a seguir.

O primeiro capítulo apresenta a contextualização e justificativa para o desenvolvimento da proposta do trabalho.

O segundo capítulo aborda segurança da informação, detalhando alguns conceitos importantes sobre: ataques, mecanismos de segurança, criptografia, etc.

O terceiro capítulo explica a esteganografia, desde sua história até a aplicação na atualidade em seus diversos meios. Aborda as técnicas para esteganografia em arquivos de áudio, e métodos para compressão de dados, mais especificamente a compressão de Huffman, método estatístico bastante utilizado para a compressão de textos.

O quarto capítulo trata sobre o desenvolvimento do trabalho, mostrando os diagramas de classe, casos de uso e diagramas de seqüência. Este capítulo também explica a implementação do protótipo e os resultados obtidos.

2 SEGURANÇA DAS INFORMAÇÕES

Os computadores vêm assumindo uma crescente importância como meios de armazenamento, processamento e troca de informação entre várias instituições da nossa sociedade. Visando garantir sua capacidade operacional e de viabilização de negócios, as empresas estão investindo cada vez mais em tecnologia para suportar a crescente demanda de serviços na busca de sua diferenciação em um mercado altamente competitivo.

Um dos problemas mais árduos que a indústria de computadores enfrenta é o da segurança de dados. Atualmente vive-se em uma sociedade baseada em informações, onde se faz cada vez mais necessária uma proteção contra o acesso ou manipulação de informações confidenciais por elementos não autorizados, seja de forma intencional ou arbitrária. De acordo com Silva (1998), o principal objetivo da segurança é restringir o uso de informações no computador e dispositivos de armazenamento associados a indivíduos selecionados.

A comunicação segura é concebida de modo a garantir o sigilo, integridade e autenticação dos dados (KUROSE; ROSS, 1995):

a) **sigilo:** garantir que apenas usuários autorizados tenham acesso à informação, ou consigam torná-la inteligível;
b) **integridade**: garantir que os dados transmitidos cheguem ao seu destino íntegros, eliminando a possibilidade de terem sido modificados no caminho sem que isto possa ser detectado;
c) **autenticação:** verificar se a pessoa ou processo com quem se está comunicando é de fato a pessoa ou processo que alega ser.

Soares (1995) define segurança como sendo a tentativa de minimizar a vulnerabilidade de bens e recursos, sendo vulnerabilidade qualquer fraqueza que possa ser explorada para atacar um sistema ou as informações que ele contém. De forma mais abrangente, o grande objetivo a ser alcançado é dotar os sistemas de computação de, pelo menos, duas características básicas: confiabilidade e disponibilidade. Confiabilidade é definida como sendo a capacidade que um sistema tem em responder a uma dada especificação dentro de condições definidas e durante um certo tempo de funcionamento. Disponibilidade é a probabilidade de que o sistema esteja funcionando em um dado instante.

Vários mecanismos podem ser utilizados a fim de manter a segurança das informações. Um dos métodos mais eficientes e mais utilizados de se transferir informações, sem que haja a possibilidade de comprometimento do sigilo, é através da codificação ou cifragem das informações, de modo a torná-las incompreensíveis, conhecida como criptografia. Outro método que pode ser utilizado para a transferência de dados com maior segurança é a esteganografia. A intenção na esteganografia é esconder a existência da mensagem, enquanto que a criptografia codifica uma mensagem de modo que não se possa compreendê-la.

2.1 AMEAÇAS E ATAQUES

Ameaças à segurança consistem em violações da segurança de um sistema, as quais podem comprometer os principais fundamentos da segurança em relação às informações: integridade, confidencialidade e disponibilidade. Os comportamentos das ameaças, em relação às posições da origem e do destino da mensagem podem ser visualizadas na Figura 1.

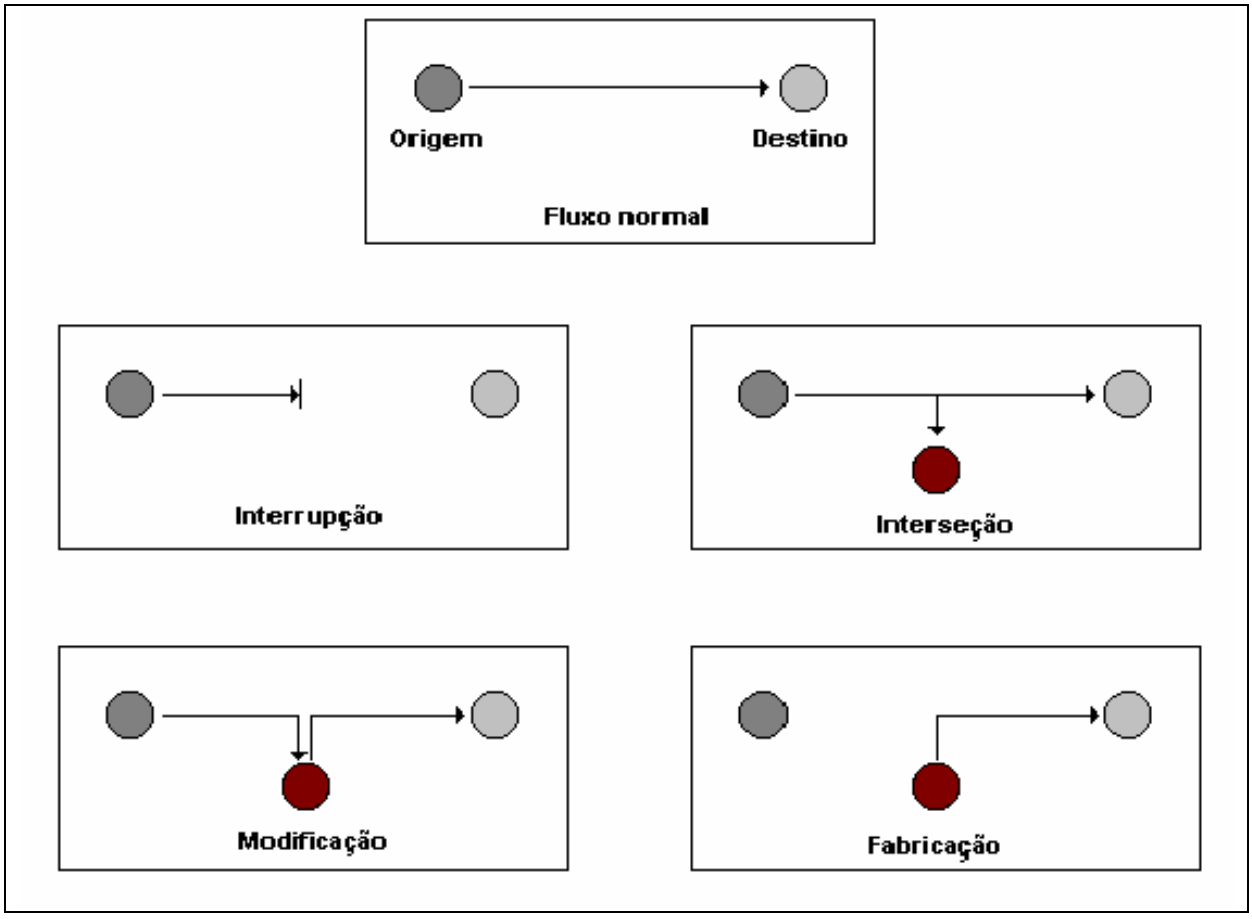


Figura 1 – Posição do intruso em relação à origem e ao destino

Veríssimo (2002) cita como sendo os principais objetivos de cada ameaça:

a) **interrupção**: interromper o fluxo de dados que parte da origem, deixando o dispositivo de destino sem receber pacotes;
b) **interseção**: tomar conhecimento de todo fluxo de dados que trafega por essa conexão;
c) **modificação**: escutar o tráfego, interceptar os dados e modificá-los, enviando-os para o destino;
d) **fabricação**: fabricar dados para enviar para o destino. O dispositivo destino não tem como saber quem está enviando esses dados.

As ameaças podem ser tanto acidentais, onde as mesmas não estão associadas à intenção premeditada, como descuidos operacionais e *bugs* de software, por exemplo, quanto intencionais, configurando um ataque, onde variam desde a observação de dados com ferramentas simples de monitoramento de redes, a ataques sofisticados baseados no conhecimento do funcionamento do sistema.

As ameaças podem ser classificadas também em ameaças passivas e ativas, onde nas passivas não há alteração ou modificação nas informações, enquanto as ativas são as quais provocam uma modificação nas informações, estados ou operações de um sistema (SOARES, 1995).

2.2 MECANISMOS DE PROTEÇÃO

A implementação de uma política de segurança pode ser feita através de diversos mecanismos, desde criação de cópias de segurança, até ferramentas de detecção de invasão. Dentre os mecanismos atualmente empregados, a prevenção de incidentes é a área com grande aceitação e maior penetração no contexto atual de segurança (CAMPELLO; WEBER, 2003).

Mecanismos de proteção são as diversas técnicas ou métodos utilizados a fim de tentar prever ou detectar um ataque de segurança.

2.2.1 CRIPTOGRAFIA

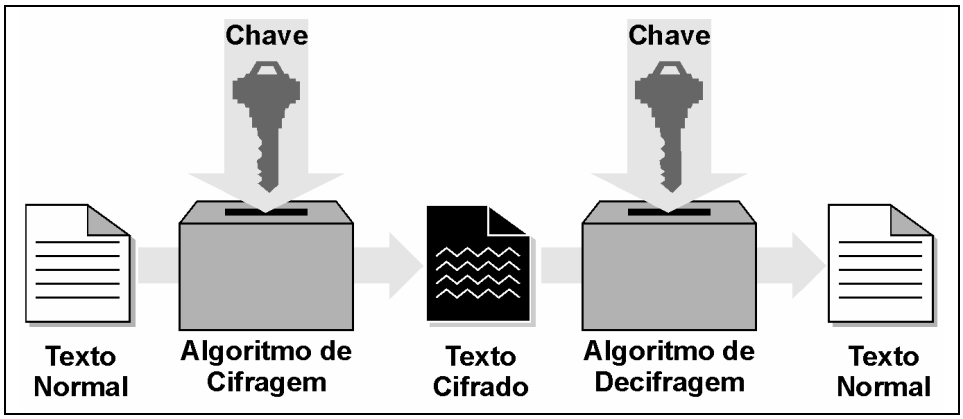
Criptografia é a arte ou ciência de escrever em cifra ou em códigos, de forma a permitir que somente o destinatário a decifre e compreenda a informação. Seu propósito é o

de garantir a privacidade, mantendo a informação escondida de qualquer pessoa não autorizada, mesmo que esta consiga visualizar os dados criptografados.

A criptografia não oferece proteção ao nível de garantir a disponibilidade da informação. Isso se deve a que, essencialmente, uma vez interceptada uma mensagem, alguém que deseje de fato prejudicar as entidades intervenientes na comunicação pode simplesmente danificar ou mesmo destruir essa mensagem (SILVA, 1998).

Segundo Soares (1995), a criptografia surgiu da necessidade de se enviar informações sensíveis através de meios de comunicação não confiáveis, ou seja, em meios onde não é possível garantir uma interceptação do fluxo de dados.

As mensagens a serem criptografadas, conhecidas como texto simples, são transformadas por uma função parametrizada por uma chave. Em seguida, a saída do processo de criptografia, conhecida como texto cifrado, é transmitida normalmente. Mesmo que os dados sejam interceptados, os mesmos não podem ser visualizados com muita facilidade, pois o intruso não conhece a chave para decriptografar a mensagem transmitida. Este procedimento pode ser visualizado na Figura 2.



Fonte: Macêdo; Trinta (1998)

Figura 2 – Cifragem e decifragem de uma mensagem

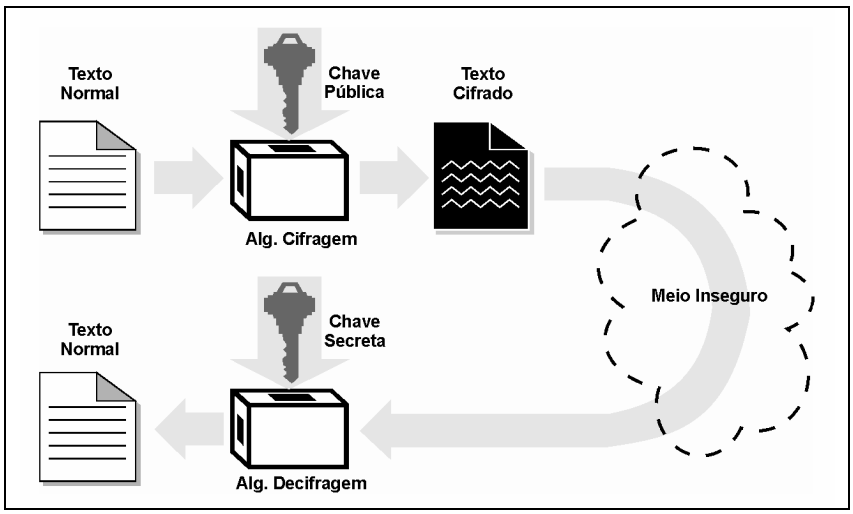
A segurança na criptografia não pode estar baseada nos algoritmos de codificação e decodificação, mas sim em um valor: a chave. O mecanismo deve ser tão seguro que mesmo o autor de um algoritmo não seja capaz de decodificar uma mensagem se não possuir a chave. Este é o princípio de Kerckhoff, onde a segurança de um criptossistema não deve depender da

manutenção de um criptoalgoritmo em segredo. A segurança depende apenas de se manter em segredo a chave (TANENBAUM, 2003).

Os métodos de criptografia podem ser divididos em duas categorias quanto ao tipo de cifra que trabalham: as cifras de substituição e as cifras de transposição. Em uma cifra de substituição, cada letra ou grupo de letras é substituído por outra letra ou grupo de letras, criando assim um “disfarce” na mensagem a ser transmitida, enquanto que em uma cifra de transposição as letras são reordenadas, e não sofrem nenhum disfarce.

Os modernos algoritmos de criptografia podem ser classificados de acordo com o tipo de chave que utilizam: os de chave simétrica e os de chave assimétrica.

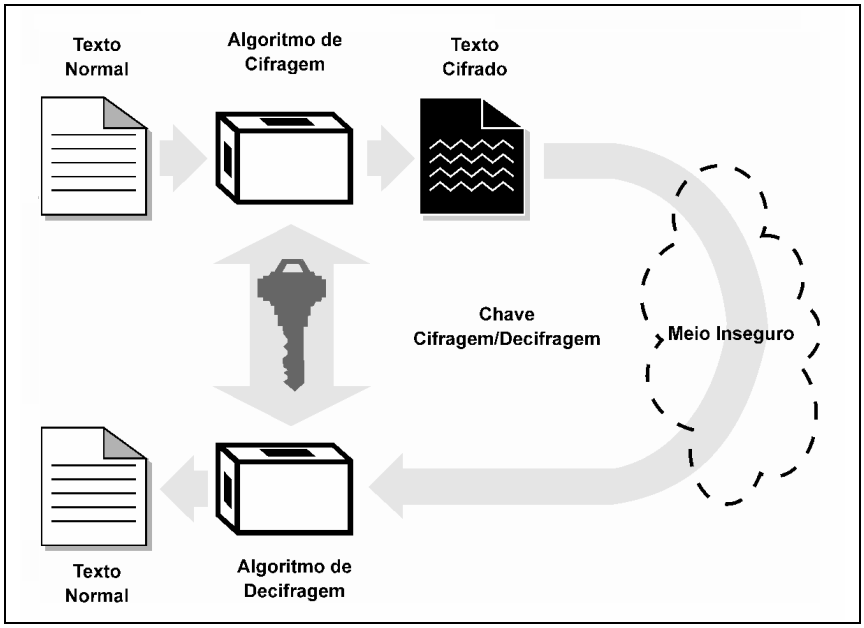
Os algoritmos de chave simétrica, também conhecidos como algoritmos de chave única ou de chave secreta, caracterizam-se por utilizar a mesma chave secreta, tanto para a cifragem como para a decifragem dos dados. Se a chave permanecer secreta, então ninguém mais além do transmissor e do receptor, poderá ler a mensagem. O exemplo mais difundido é o DES (*Data Encryption Standard*), um padrão de criptografia de chaves simétricas desenvolvido em 1977 e atualizado em 1993 pelo U.S. National Bureau of Standards para uso comercial e não confidencial do governo norte-americano. O DES cifra blocos de 64 bits, produzindo 64 bits de texto cifrado, utilizando uma chave de 56 bits (64 bits, onde 8 são de paridade). Assim, para cifrar escolhe-se uma chave e divide-se a mensagem em blocos de 64 bits, cifrando cada bloco separadamente (Figura 3).



Fonte: Macêdo; Trinta (1998)

Figura 3 – Imagem Criptografia Simétrica

Os algoritmos de chave assimétrica, também chamados de algoritmos de chave pública e privada, utilizam duas chaves: uma pública que pode ser divulgada e outra secreta conhecida somente por pessoas autorizadas. Se uma das chaves é usada para cifrar uma mensagem, então a outra deve ser usada para decifrar. O exemplo bastante conhecido de chave pública é o RSA, acrônimo de Rivest, Shamir e Adleman, seus autores, e sua segurança está baseada na dificuldade de fatorar grandes números: as chaves são calculadas matematicamente combinando dois números primos de tamanho grande (Figura 4).



Fonte: Macêdo; Trinta (1998)

Figura 4 – Imagem Criptografia Assimétrica

Analisando os dois métodos, pode-se observar que a criptografia por chave pública tem a vantagem sobre a chave secreta no sentido de viabilizar a comunicação segura entre pessoas comuns. Com a chave pública também acaba o problema da distribuição de chaves existente na criptografia por chave secreta, pois não há necessidade do compartilhamento de uma mesma chave, nem de um pré-acordo entre as partes interessadas. Com isto o nível de segurança é maior. A principal vantagem da criptografia por chave secreta está na velocidade dos processos de cifragem/decifragem, pois estes tendem a ser mais rápidos que os de chave pública.

Todos os sistemas criptográficos possuem níveis diferentes de segurança, dependendo da facilidade ou dificuldade com que os mesmos são quebrados. Só tem-se um sistema condicionalmente seguro quando ele for teoricamente inquebrável, ou seja, não importa a

quantidade de texto normal ou cifrado a disposição de um criptoanalista, ele nunca terá informação suficiente para se quebrar as cifras ou deduzir as chaves que foram usadas (MACÊDO; TRINTA, 1998).

2.2.2 CRIPTOANÁLISE

Criptoanálise é a ciência que estuda os princípios, processos e métodos para desvendar os segredos dos sistemas criptográficos existentes, onde o objetivo principal do atacante, também conhecido como criptoanalista, é ganhar acesso ou mesmo capacidade de alterar as informações ou dados pretensamente protegidos (PUTTINI, 2000).

A grande dificuldade que um criptoanalista encontra ao tentar quebrar um algoritmo está no número de chaves que esse utiliza. Quanto maior o número de chaves, maior a dificuldade de se conseguir obter sucesso na criptoanálise de um determinado sistema. Um sistema que não se consegue quebrar é conhecido como sistema de segurança perfeita.

A criptoanálise clássica envolve uma interessante combinação de raciocínio analítico, uso de ferramentas matemáticas, testes, paciência, determinação e sorte.

As tentativas de se quebrar os códigos de algoritmos são chamadas ataques, cujos possíveis tipos usando criptoanálise são (MACÊDO; TRINTA, 1998):

a) **ataque por texto conhecido:** o criptoanalista tem um bloco de texto normal e seu correspondente bloco cifrado, com objetivo de determinar a chave de criptografia para futuras mensagens;
b) **ataque por texto escolhido**: o criptoanalista tem a possibilidade de escolher o texto normal e conseguir seu texto cifrado correspondente;
c) **criptoanálise diferencial**: variação do ataque por texto escolhido, onde procura-se cifrar muitos textos bem parecidos e comparar seus resultados.

2.2.3 ESTEGANOGRAFIA

A esteganografia é a arte de comunicar-se secretamente, ocultando uma mensagem sigilosa dentro de outra informação sem importância, de maneira que não exista forma de detectar que há uma mensagem escondida. Na computação, essa outra informação pode ser um arquivo de som, imagem ou texto, onde em uma das técnicas utilizadas, parte dos dados destes arquivos as quais não são utilizadas, ou são pouco significativas são utilizadas para a

inclusão da informação, tornando-a oculta dentro do arquivo. Este assunto é detalhado no capítulo 3 deste trabalho.

2.2.4 AUTENTICAÇÃO

A autenticação é o método pelo qual um processo confirma se o parceiro na comunicação é quem deve realmente ser e não um impostor. Confirmar a identidade de um processo remoto exige protocolos complexos baseados no uso da criptografia (TANENBAUM, 2003).

A autenticação garante que os dados recebidos correspondem àqueles originalmente enviados, assim como garante a identidade do emissor.

2.2.5 ASSINATURA DIGITAL

A autenticidade de muitos documentos legais, financeiros e outros documentos é determinada pela presença de uma assinatura autorizada (TANENBAUM, 2003). Um documento digital não pode ser assinado no modo tradicional, através do qual o autor se identifica por meio de sua assinatura manuscrita. Porém, através da assinatura digital, é possível garantir a autenticidade destes documentos, garantindo ao destinatário de uma mensagem digital tanto a identidade do remetente quanto a integridade da mensagem.

Segundo Puttini (2000), as assinaturas digitais satisfazem os cinco critérios das assinaturas de papel:

a) a assinatura não pode ser falsificada: só o emissor conhece sua chave privada;
b) a assinatura é autêntica: quando o receptor verifica a assinatura com a chave pública do emissor, o receptor sabe que quem assinou a mensagem foi o emissor;
c) a assinatura não é realizável: a assinatura em um documento não pode ser transferida para qualquer outro documento;
d) o documento assinado é inalterável: qualquer alteração de um documento (se ele foi ou não codificado) ou assinatura, não é mais válido;
e) a assinatura não pode ser repudiada: o receptor precisa do auxílio do emissor para comprovar sua assinatura.

De acordo com Tanenbaum (2003), para a criação de uma assinatura digital, necessita-se de um sistema através do qual uma parte possa enviar uma mensagem “assinada” para outra de forma que:

a) o receptor possa verificar a identidade alegada pelo transmissor;
b) posteriormente, o transmissor não possa repudiar o conteúdo da mensagem;
c) o receptor não tenha a possibilidade de forjar ele mesmo a mensagem.

O processo de assinatura digital utiliza-se de algoritmos criptográficos para fundir um segredo a um conjunto de bytes (mensagem a ser assinada). A garantia é que somente quem conhece o segredo pode reproduzir o mesmo resultado. O processo de verificação da assinatura (reconhecimento de firma) utiliza-se de uma informação pública acrescida da mensagem original para verificar se a referida pessoa efetivamente assinou a mensagem.

3 ESTEGANOGRAFIA

Esteganografia é a arte de mascarar informações como uma forma de evitar a sua detecção. Esteganografia deriva do grego, onde *estegano* = esconder, mascarar e *grafia* = escrita. Logo, estaganografia é a arte da escrita encoberta.

A esteganografia inclui um vasto conjunto de métodos para comunicações secretas desenvolvidos ao longo da história. Dentre tais métodos estão: tintas “invisíveis”, micro-pontos, arranjamento de caracteres (*character arranjement*), assinaturas digitais, canais escondidos (*covert channels*), comunicações por espalhamento de espectro (*spread spectrum communications*), entre outras (TOMÁS, 2004).

3.1 HISTÓRICO

O primeiro uso confirmado da esteganografia está em "As Histórias" de Heródoto e remonta ao século V a.C. Um certo Histio, querendo fazer contato secreto com seu superior, o tirano Aristágoras de Mileto, escolheu um escravo fiel, raspou sua cabeça e escreveu na pele a mensagem que queria enviar. Esperou que os cabelos crescessem e mandou o escravo ao encontro de Aristágoras com a instrução de que deveriam raspar seus cabelos (HETZL; MÜLLER; SELLARS, 2002).

Outra história da Grécia antiga também chega a nós via Heródoto. O meio de escrita na época era texto, escrito em tabletes cobertos de cera. Demeratus, um grego, precisava avisar Esparta que Xerxes pretendia invadir a Grécia. Para evitar a captura, ele removeu a cera dos tabletes e escreveu a mensagem na madeira subjacente. Então ele cobriu os tabletes com cera de novo. Os tabletes pareciam estar em branco e sem uso, por isso passaram pela inspeção (TOMÁS, 2004).

Tintas invisíveis sempre foram uma forma bastante popular da Esteganografia. Os romanos costumavam escrever entre as linhas usando tintas invisíveis baseadas em substâncias como suco de frutas, urina e leite. Quando aquecidas, as tintas invisíveis escureciam e se tornavam legíveis.

Porém, foi durante o século 20 que a esteganografia realmente floresceu. A Segunda Guerra Mundial marcou um período de intensos experimentos esteganográficos. No início da guerra, a tecnologia esteganográfica consistia quase inteiramente em tintas invisíveis. Mais

tarde, cifras nulas (mensagens não encriptadas) foram usadas para esconder mensagens secretas. As cifras nulas, que geralmente pareciam ser mensagens inocentes sobre acontecimentos ordinários, não gerariam suspeitas, não sendo então interceptadas.

Um exemplo do uso de esteganografia é visto na seguinte mensagem, enviada por um espião alemão durante a Segunda Guerra: *"Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by products, ejecting suets and vegetable oils"*. Retirando a segunda letra de cada palavra, pode-se ler a mensagem de fato a ser transmitida: "*Pershing sails from NY June 1"* (TOMÁS, 2004).

Atualmente, com o surgimento da era da computação, a Esteganografia recebeu um grande impulso. Velhos métodos, como ocultar mensagens em imagens, ganharam vida novamente através do computador.

3.2 ESTEGANOGRAFIA X CRIPTOGRAFIA

A esteganografia é um ramo da Criptologia que consiste no estudo das formas de ocultação de uma mensagem. Ao invés de cifrá-la, técnica utilizada na criptografia de mensagens, o que se busca é tornar a sua presença imperceptível.

Os dois métodos (criptografia e esteganografia) podem ser combinados para aumento da segurança. Por exemplo, pode-se criptografar uma mensagem e em seguida, utilizar a técnica de esteganografia, trocando-se os bits menos significativos de uma imagem digitalizada pelos bits da mensagem criptografada, e então transmitir a imagem. Se a imagem for interceptada, o adversário primeiro precisará descobrir a mensagem dentre os bits da imagem, e, após isso, poderá tentar decriptografá-la (Figura 5).

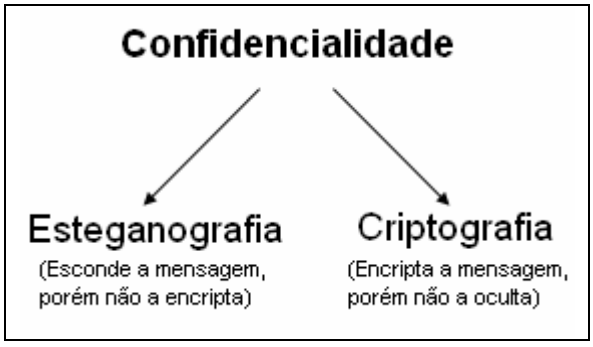


Figura 5 – Criptografia x Esteganografia

3.3 ESTEGANOGRAFIA EM IMAGENS

Para a confecção das imagens, utiliza-se o fato de o olho humano não ser capaz de identificar pequenas modificações em tons próximos. Com esse raciocínio, geralmente sugere-se que sejam utilizadas imagens em tons de cinza, pois estas possuem menores variações *byte* a *byte*.

A informação pode ser escondida de várias formas em imagens. As abordagens mais comuns são:

a) inserção no *bit* menos significativo;
b) máscara e filtragem;
c) algoritmos e transformações.

A técnica de inserção no *bit* menos significativo é o método mais comum para armazenamento de informações em arquivos de imagens, porém é uma das mais vulneráveis à manipulação da imagem. Esta técnica utiliza o último *bit* de cada *byte* da imagem, isto é, o *bit* menos significativo, para ocultar a mensagem desejada. No caso de imagens coloridas, utiliza-se 3 *bits* em cada *pixel*, sendo cada um deles pertencentes a cada uma das componentes RGB (*Red, Green, Blue*).

Técnicas de filtragem e mascaramento são restritas às imagens em tons de cinza (*grayscale*). Estas técnicas escondem a informação através da criação de uma imagem semelhante às marcas d'água em papel. Isto acontece porque as técnicas de *watermarking* garantem que, mesmo se a imagem for modificada por métodos de compressão, a marcação não será removida (ROCHA, 2003).

Os algoritmos de transformação geralmente trabalham com formas mais sofisticadas de manuseio de imagens como brilho, saturação, luminescência e compressão das imagens. As imagens podem também ser processadas usando a transformação rápida de Fourier e Wavelet.

3.4 ESTEGANOGRAFIA EM ÁUDIO

As amostras de sons são, por natureza, estimações inexatas de um valor correto de um som em um momento particular do tempo. Estas leves incorreções são aproveitadas para ocultar informações.

Um dos métodos mais utilizados para o uso de esteganografia distribui o padrão de bits correspondentes do arquivo que se deseja ocultar através dos bits menos significativos do arquivo de som. Outros métodos utilizados são codificação de fases, *spread spectrum* e ocultação de dados no eco.

O LSB é o *bit* de cada amostra, que menos informações oferece, ou seja, o último *bit*. Se em um número binário 1101 (13), modificar-se o seu 1º *bit*, tem-se 0101 (5), porém se modificar-se seu último *bit*, tem-se como resultado 1100 (12), o que implica em uma mudança do valor resultante, muito menor que o primeiro exemplo, e desta forma facilitando sua passagem despercebida.

Supondo que um arquivo de som tenha as seguintes informações armazenadas em alguma parte do mesmo, conforme a Tabela 1 (a), cujos respectivos valores binários podem ser visualizados na linha (b). Se por exemplo, queira-se ocultar o *byte* binário 01110101, correspondente ao valor 117 (c) dentro desta seqüência, simplesmente substituímos os *bits* menos significativos de cada *byte* do arquivo de som original, por cada um dos *bits* do *byte* a ser oculto no arquivo (d). Desta forma, após ocultar 8 *bits* de informação, em 8 amostras de som, apenas alguns dos valores da seqüência original sofrem alteração (e).

Tabela 1 – Funcionamento do Método LSB – *Last Significant Bit*

(a)	100	67	58	4	184	48	198	142
(b)	01100100	01000011	00111010	00000100	10111000	00110000	11000110	10001110
(c)	0	1	1	1	0	1	0	1
(d)	0110010**0**	0100001**1**	0011101**1**	0000010**1**	1011100**0**	0011000**1**	1100011**0**	1000111**1**
(e)	100	67	59	5	184	49	198	143

Pode-se verificar claramente, que os valores das amostras de som foram alteradas, no máximo, em um valor apenas. Caso seja aplicada a lei da probabilidade, chega-se à conclusão de que a quantidade de amostras alteradas tendem a 50%, já que a probabilidade de que o LSB da amostra coincida com a informação que deseja-se incluir é de 0,5 (possibilidade 1 ou 0).

Estas pequenas alterações nos arquivos são inaudíveis para o ouvido humano. Esta é a teoria na qual trabalham a maioria das ferramentas de esteganografia em arquivos de áudio.

3.4.1 ESTEGANÁLISE

Grande parte das técnicas de esteganografia possui falhas ou inserem artefatos detectáveis nos objetos de cobertura. Algumas vezes, basta um atacante, alguém interessado em descobrir indevidamente a mensagem, fazer um exame mais detalhado destes artefatos para descobrir que há mensagens escondidas (ROCHA, 2003).

A esteganálise é a arte de detectar mensagens escondidas nos mais diversos meios digitais. Além de fornecer técnicas robustas para detecção de mensagens escondidas, a esteganálise pode ser aplicada na criação de *watermarks* mais resistentes ao detectar e destruir marcações mais frágeis.

Segundo Rocha (2003), recuperar os dados escondidos, atualmente, está além das capacidades da maioria dos testes e técnicas de esteganálise, devido a muitos algoritmos de mascaramento utilizarem geradores aleatórios criptográficos muito seguros para misturar a informação no processo de mascaramento. As pesquisas de esteganálise estão mais concentradas em simplesmente identificar a presença de mensagens ocultas, ao invés de extraí-las.

3.5 ARQUIVOS DE ÁUDIO

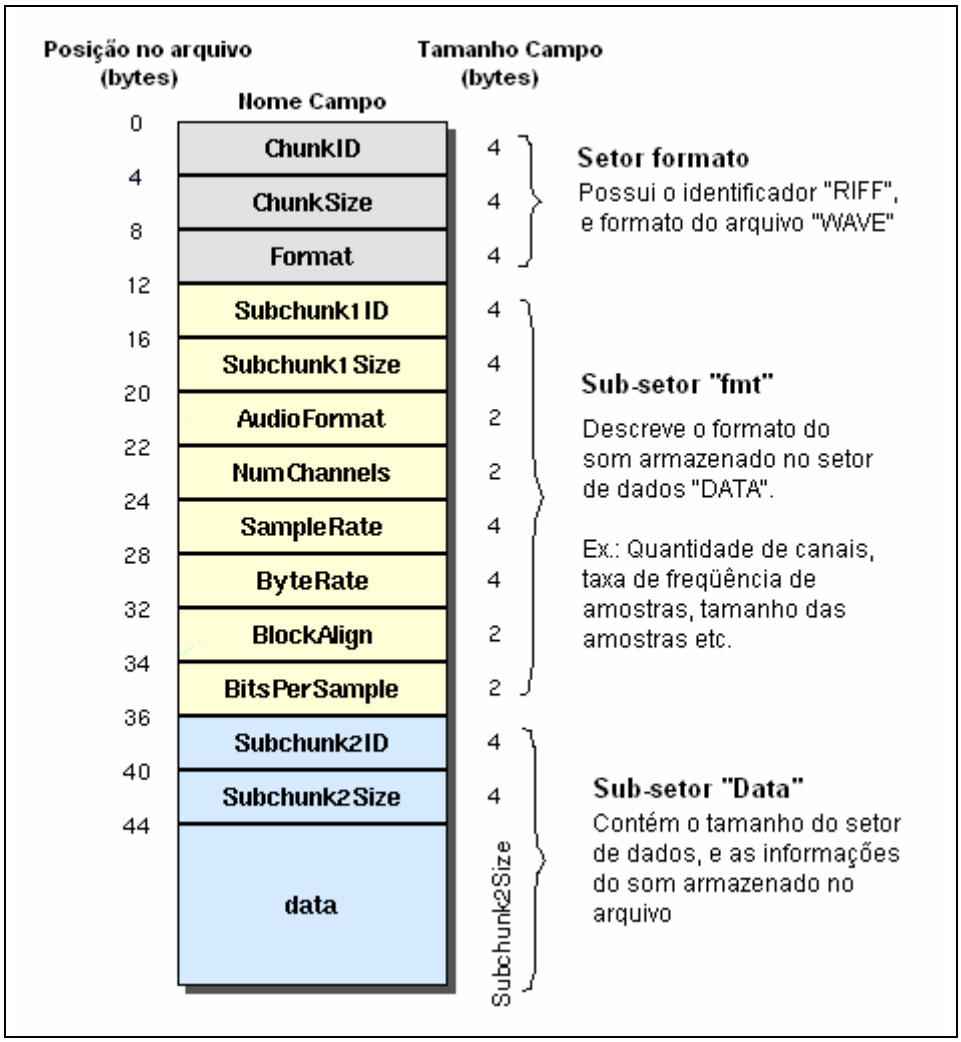
O som digital pode ser representado como amostras de som (*samples*), onde a cada fração de segundo uma amostra do som é capturada e armazenada no formato digital, ou seja, a informação é representada na forma de *bits* e *bytes*. A freqüência com que as amostras são capturadas determina a taxa de amostragem, e a quantidade de informações armazenadas a cada amostragem determina o tamanho da amostra, onde as freqüências mais utilizadas são 44,1 kHz e 22,05 kHz, e os tamanhos das amostragens são de 8 ou 16 *bits* (VAUGHAN, 1994).

A digitalização de sons é feita a partir de fontes analógicas, necessitando a conversão de analógico para digital. A conversão de um sinal analógico em um sinal digital envolve a captura de uma série de amostras da fonte analógica, onde a agregação das amostras forma o equivalente digital de uma onda sonora analógica (TAROUCO, 2003).

Um dos primeiros formatos utilizados ainda em uso para o armazenamento de sons digitais, é o formato *wav*. Este tipo de arquivo de som é composto de duas partes:

a) **cabeçalho**: contém a informação de como está digitalizado o som, como quantidade de canais, freqüência de amostras e tamanho de amostras;
b) **setor de dados**: aqui se encontram em formas de *bits*, as amostras de som do arquivo.

Quando é executado um arquivo *wav*, primeiramente é lido o cabeçalho para determinar o formato dos *bits* e desta forma efetuar a leitura dos mesmos corretamente. Uma vez lido o cabeçalho, é iniciada a leitura dos *bits* do setor de dados a fim de reproduzir o som armazenado no arquivo. O formato de um arquivo *wav* pode ser visualizado na Figura 6.



Fonte: Wilson (2003)

Figura 6 – Formato de um arquivo *wav*

3.6 COMPRESSÃO DE DADOS

Compressão de dados é o processo de codificar um corpo de informações digitais dentro de uma representação menor, da qual a original pode ser reconstituída em momento posterior. As técnicas de compressão de dados podem ser divididas em duas grandes famílias: com perda e sem perda (LYRA, 2003).

3.6.1 COMPRESSÃO SEM PERDA

Se a informação pode sempre ser reconstituída exatamente sem qualquer distorção ou perda de informação, o processo é denominado “sem perda” (*lossless*).

Os algoritmos sem perda, conhecidos como codificadores universais, mantém a integridade da informação codificada, porém as taxas de compressão são visivelmente mais modestas, devido ao limite entrópico da mensagem, isto é, a quantidade real de informação existente (BRUNO, 2002).

A compressão sem perda é baseada em técnicas que garantem uma cópia exata do fluxo de dados de entrada depois de um ciclo de compressão/expansão. Esse é o tipo de compressão usada para armazenar registros de banco de dados, planilhas eletrônicas ou arquivos de processadores de textos.

3.6.2 COMPRESSÃO COM PERDA

A compressão “com perda” (*lossy*), sacrifica um pouco da integridade da informação em troca de um grande incremento na compressão.

Os codificadores com perda são normalmente conhecidos como quantizadores, pois a informação original é submetida a um processo de quantização, permitindo altas taxas de compressão, ao custo da perda de fidelidade de informação (BRUNO, 2002).

A compressão de dados com perda é geralmente utilizada nos casos onde pode implicar em perda de qualidade, sem comprometer profundamente a interpretação do conteúdo, como na compressão de dados do tipo imagem, vídeo e som. Exemplos bastante conhecidos de seu uso são imagens com extensão jpeg e arquivos de áudio com extensão mp3.

3.6.3 CODIFICAÇÃO DE HUFFMAN

O código de Huffman é um método estatístico utilizado para codificar um texto de forma a obter uma compactação que seja ótima dentro de certos critérios. A construção desse código foi desenvolvida por David Huffman, que utilizou a estrutura de árvore binária, de forma a gerar um código binário.

A idéia da compressão estatística é realizar uma representação otimizada de caracteres ou grupos de caracteres. Caracteres de maior freqüência de utilização são representados por códigos binários pequenos, e os de menor freqüência são representados por códigos proporcionalmente maiores.

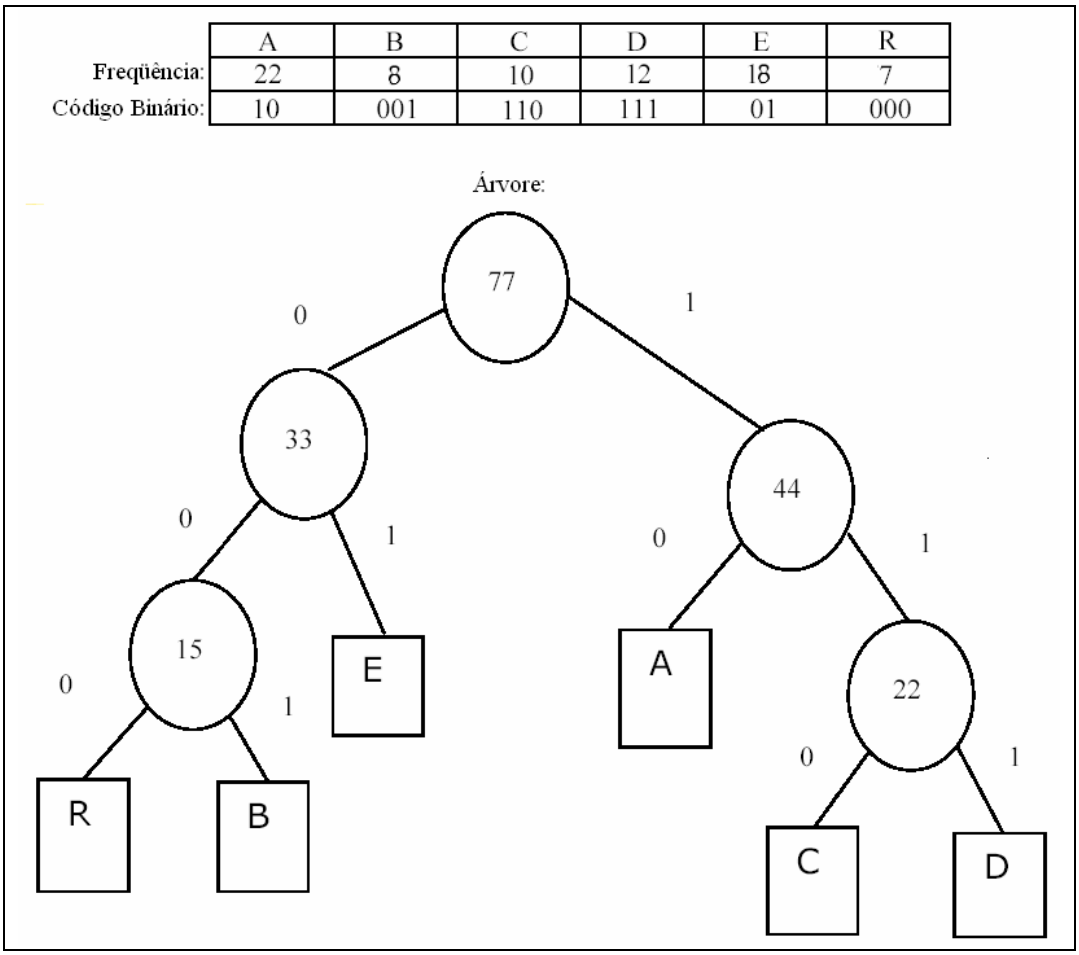
Neste tipo de compressão, portanto, não há necessidade de saber qual caractere vai ser comprimido, mas é necessário, porém, ter o conhecimento da probabilidade de ocorrência de todos os caracteres sujeitos à compressão. Caso não seja possível a tabulação de todos os caracteres sujeitos à compressão, utiliza-se uma técnica adequada para levantamento estatístico dos dados a comprimir, formando tabelas de probabilidades.

Supõe-se que o texto seja composto por um conjunto de símbolos (ou caracteres). É conhecida a freqüência com que cada símbolo aparece ao longo do texto. Deseja-se atribuir um código a cada símbolo, de modo a compactar o texto todo. A árvore estritamente binária de Huffman possui as arestas rotuladas, que serão usadas na determinação de cada código. Para cada nó interno a aresta que conduz ao filho esquerdo é rotulada com zero, enquanto o rótulo que conduz ao filho direito é igual a 1. Cada símbolo está associado a uma folha da árvore. O código de um determinado símbolo é igual à seqüência dos rótulos das arestas, do caminho desde a raiz até a folha correspondente ao símbolo.

A técnica de compressão Huffman permite a representação em binário de caracteres a partir de sua probabilidade de ocorrência. Esta representação é gerada por um sistema de codificação e decodificação em árvore binária, o que impede a ambigüidade na análise do código.

A Figura 7 ilustra a criação de uma árvore considerando uma mensagem composta pelos caracteres A, B, C, D, E, R, com freqüência de uso destas letras na mensagem de 22, 8, 10, 12, 18, e 7, respectivamente. A construção da árvore binária é baseada na freqüência de uso das letras de tal forma que as mais freqüentemente usadas apareçam mais perto da raiz que as

menos freqüentemente usadas. As letras serão representadas nas folhas e os seus vértices internos conterão um número correspondente à soma das freqüências dos seus descendentes.



Fonte: Cantarelli (2001)

Figura 7 – Exemplo de árvore binária de Huffman

Para descompactar o texto, basta percorrer a árvore a partir da seqüência binária, indo para a esquerda caso o caractere lido seja “0”, e para a direita caso “1”, até chegar a uma folha da árvore, correspondente ao caractere da seqüência.

4 DESENVOLVIMENTO DO PROTÓTIPO

Para a realização deste trabalho foram executados procedimentos de especificação e implementação visando o cumprimento dos objetivos utilizando-se de metodologias de desenvolvimento de software, ferramenta de especificação e ambiente de programação já consagrados tecnologicamente de acordo com a categoria em que o trabalho se enquadra.

Neste capítulo serão apresentados em síntese o modelo e especificação do protótipo proposto neste trabalho.

4.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

O protótipo desenvolvido neste trabalho utiliza a esteganografia para ocultar mensagens de texto em arquivos de áudio. É utilizada também uma técnica de compressão de dados, de forma a compactar a mensagem a ser transmitida, possibilitando esteganografar uma maior quantidade de texto.

Há dois tipos de usuários:

a) **emissor**: este usuário fará a inclusão de uma mensagem em um arquivo de áudio qualquer, podendo o texto estar na forma compactada ou não, tendo como saída do sistema o arquivo de áudio com a mensagem oculta;
b) **receptor**: a partir de um arquivo de áudio recebido, este usuário poderá visualizar a mensagem embutida no arquivo, efetuando a descompressão da mesma caso seja necessário.

Os requisitos funcionais (RF) e não funcionais (RNF) do protótipo desenvolvido são:

a) ocultar textos em arquivos de áudio (RF);
b) extrair textos esteganografados de arquivos de áudio, identificando se estão compactados e descompactando-os quando necessário (RF);
c) possibilitar a compactação dos textos a serem ocultos, permitindo assim um maior número de caracteres na mensagem (RF);
d) apresentar uma interface de fácil usabilidade, onde o usuário apenas seleciona o arquivo de áudio e em um campo no sistema insira o texto a ser compactado e incluído no arquivo (RNF);

4.2 ESPECIFICAÇÃO

Para fazer a especificação deste protótipo foi utilizada uma metodologia orientada a objetos, representada em diagramas da Unified Modeling Language (UML), utilizando como ferramenta o Rational Rose Enterprise Edition.

O primeiro diagrama utilizado na especificação é o de casos de uso, seguido pelo diagrama de classes, diagrama de atividades e por último os diagramas de seqüência especificando o funcionamento do protótipo.

4.2.1 DIAGRAMA DE CASOS DE USO

O protótipo possui quatro casos de usos (Figura 8):

a) **insere mensagem**: responsável por entrar com a mensagem a ser transmitida e ocultá-la no arquivo de áudio;

b) **compacta mensagem**: responsável por compactar a mensagem a ser oculta, caso seja de interesso do emissor;

c) **consulta mensagem**: responsável por ler a mensagem transmitida a partir de um arquivo de áudio;

d) **descompacta mensagem:** responsável por descompactar a mensagem recebida, quando a mesma tiver sido compactada na sua inclusão.

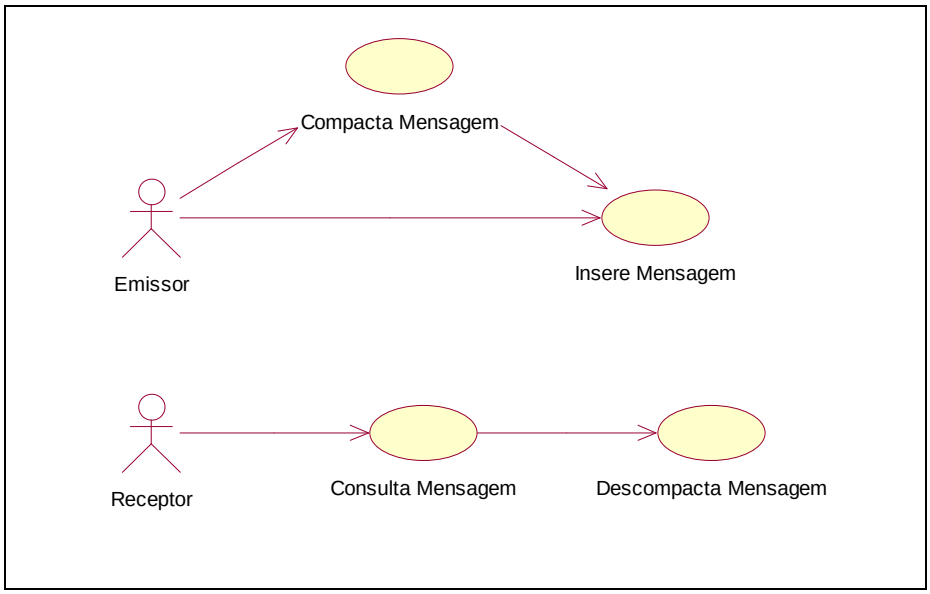


Figura 8 – Casos de Uso

4.2.2 DIAGRAMA DE CLASSES

As classes utilizadas no desenvolvimento do protótipo são (Figura 9):

a) **TWaveStream**: responsável pela manipulação dos arquivos de áudio no formato *wav*, fazendo a leitura das informações do arquivo em seu cabeçalho, utilizadas para a ocultação da mensagem, e do setor de dados do arquivo, onde será ocultada a mensagem;

b) **TMessage**: classe que contém informações da mensagem a ser esteganografada;

c) **TWaveUtility**: classe responsável pela esteganografia, ou seja, ocultar e extrair a mensagem do arquivo de áudio;

d) **THuffman**: classe escrita originalmente por Saju (2002), a qual sofreu algumas modificações para uma melhor adaptação no desenvolvimento do protótipo. Classe responsável pela compactação/descompactação da mensagem a ser esteganografada, utilizando para isso o algoritmo de Huffman.

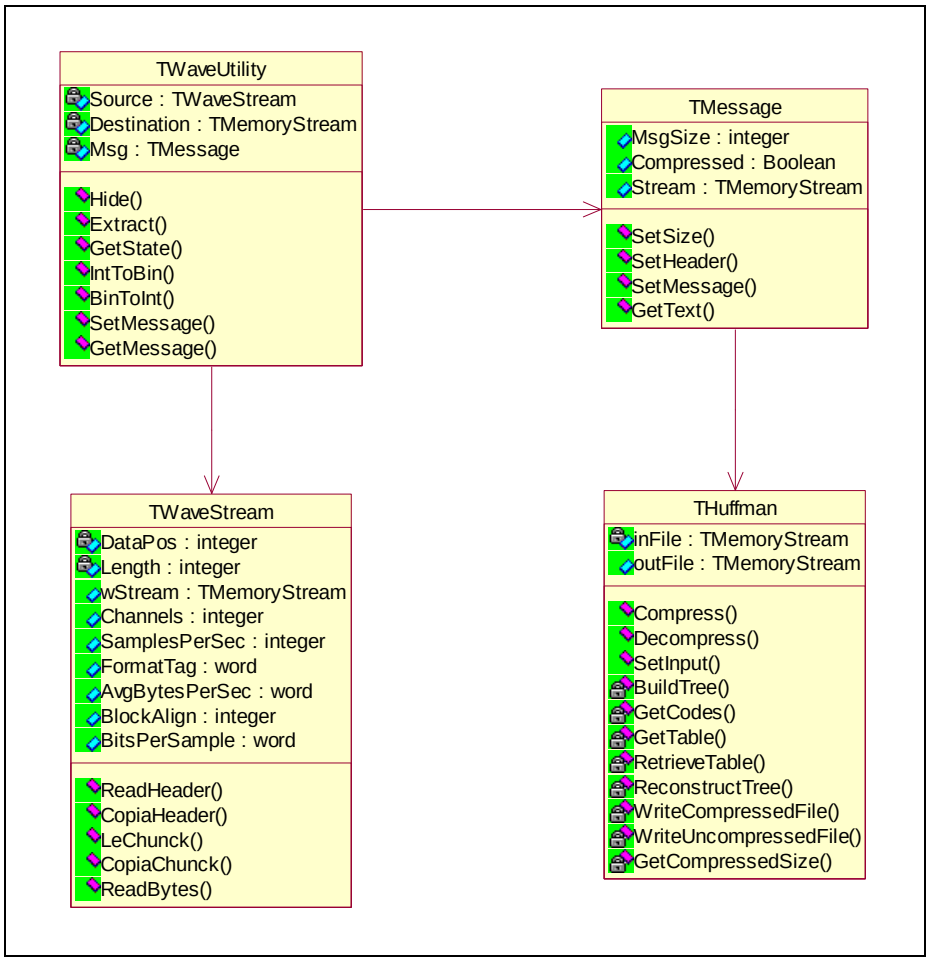


Figura 9 – Diagrama de Classes

4.2.3 DIAGRAMA DE ATIVIDADES

O diagrama de atividades visa representar o controle do fluxo de informações entre atividades de um sistema. Na Figura 10, é apresentado o diagrama de atividades do aplicativo, destacando os procedimentos que ocorrem tanto em um arquivo de áudio com texto esteganografado quanto sem texto oculto.

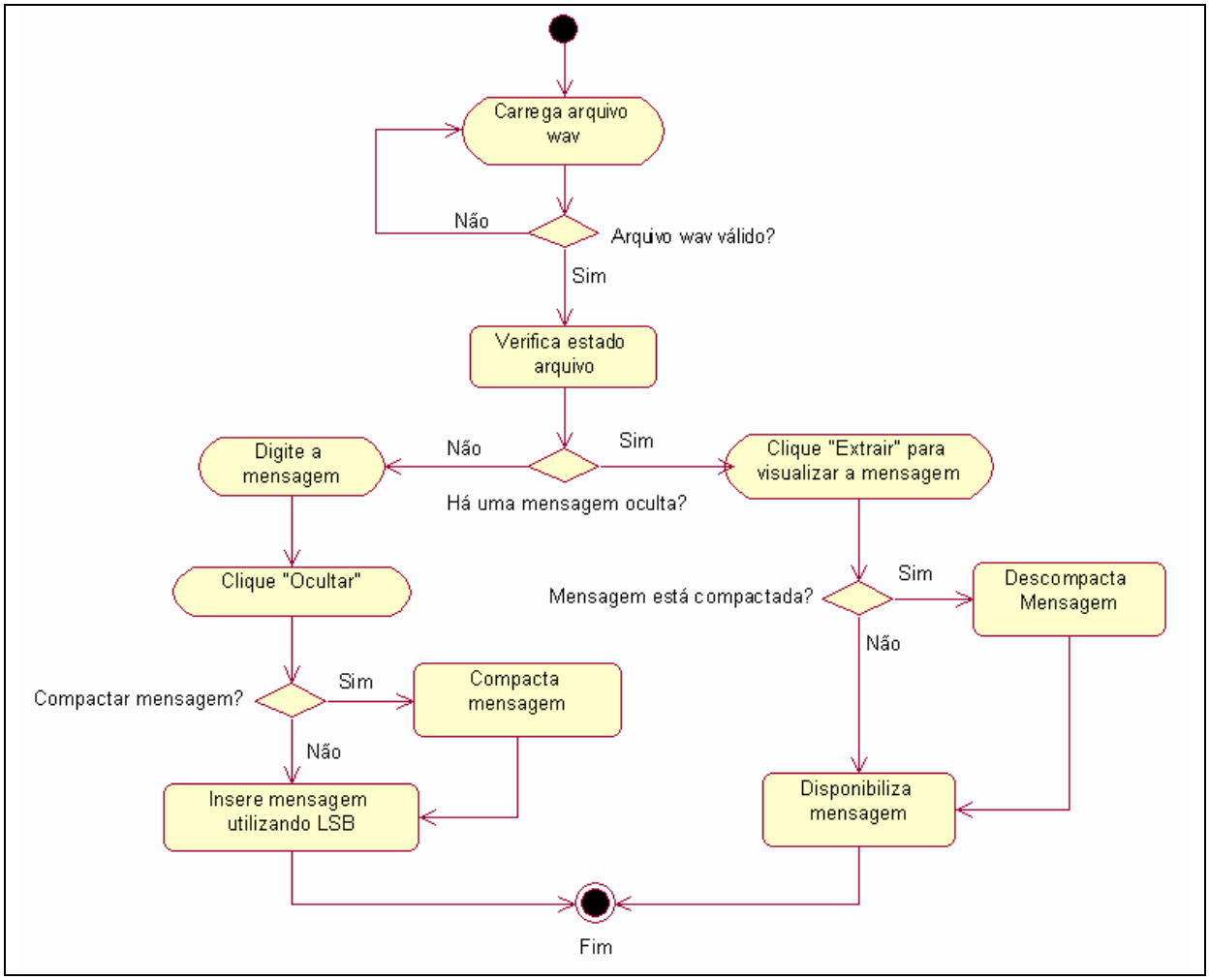


Figura 10 – Diagrama de atividades

4.2.4 DIAGRAMAS DE SEQÜÊNCIA

Os digramas de seqüência representam a seqüência das ações ocorridas em um conjunto de classes, demonstrando como ocorre a troca de mensagens entre as classes. Para cada caso especificado de uso, há um diagrama de seqüência, conforme detalhamento a seguir.

4.2.4.1 INÍCIO

Este diagrama de seqüência representado na Figura 11, mostra as ações executadas sempre que um arquivo de áudio é carregado no sistema. Ao carregar um arquivo, é instanciado um objeto da classe TWaveStream, responsável pela leitura do arquivo, e um objeto da classe TWaveUtility que inicialmente faz uma verificação no arquivo aberto, se já existe alguma mensagem oculta no mesmo.

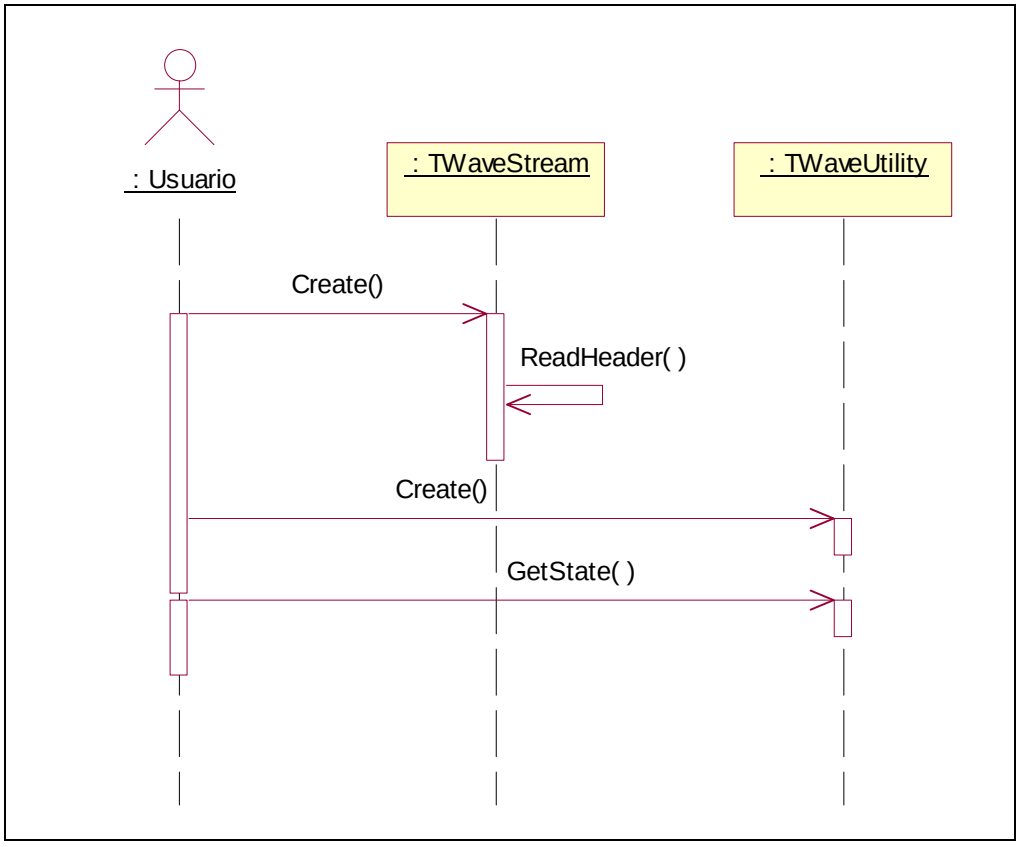


Figura 11 – Diagrama de seqüência “Início”

4.2.4.2 INSERE MENSAGEM

Este diagrama é executado quando o usuário inserir a mensagem no arquivo de áudio, onde primeiramente é executado o método “CopyHeader”, o qual faz uma cópia do cabeçalho do arquivo *wav* com as informações de formatação do arquivo original (Figura 12). Em seguida é executado o método “Hide” da classe TWaveUtility, responsável por ocultar a mensagem no arquivo de áudio, onde para a execução deste processo utiliza um objeto da classe TMessage, responsável pela formatação da mensagem para a inclusão, efetuando a compactação da mesma quando for de desejo do usuário. Para a compactação da mensagem, a classe TMessage define a entrada no objeto da classe THuffman, através do método “SetInput”, e efetua a compactação do texto através do método “Compress”.

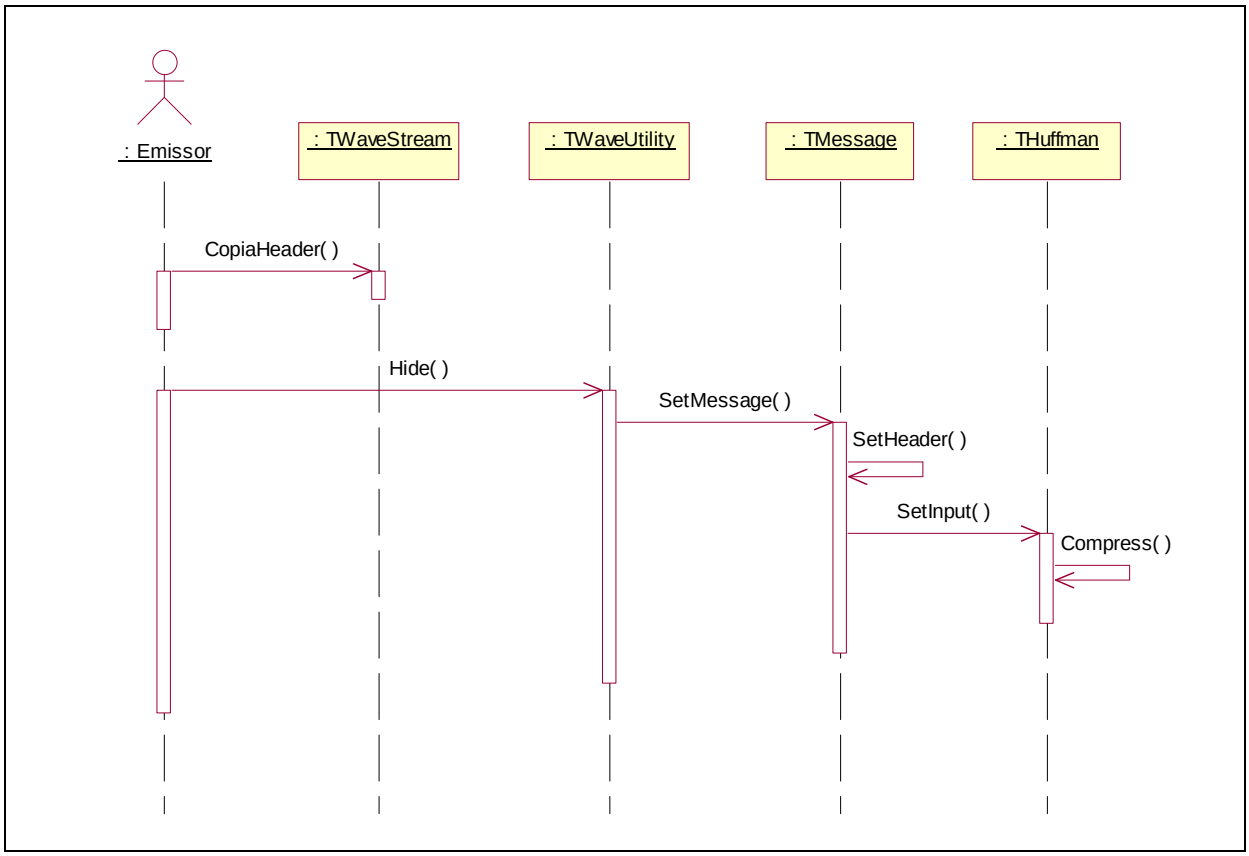
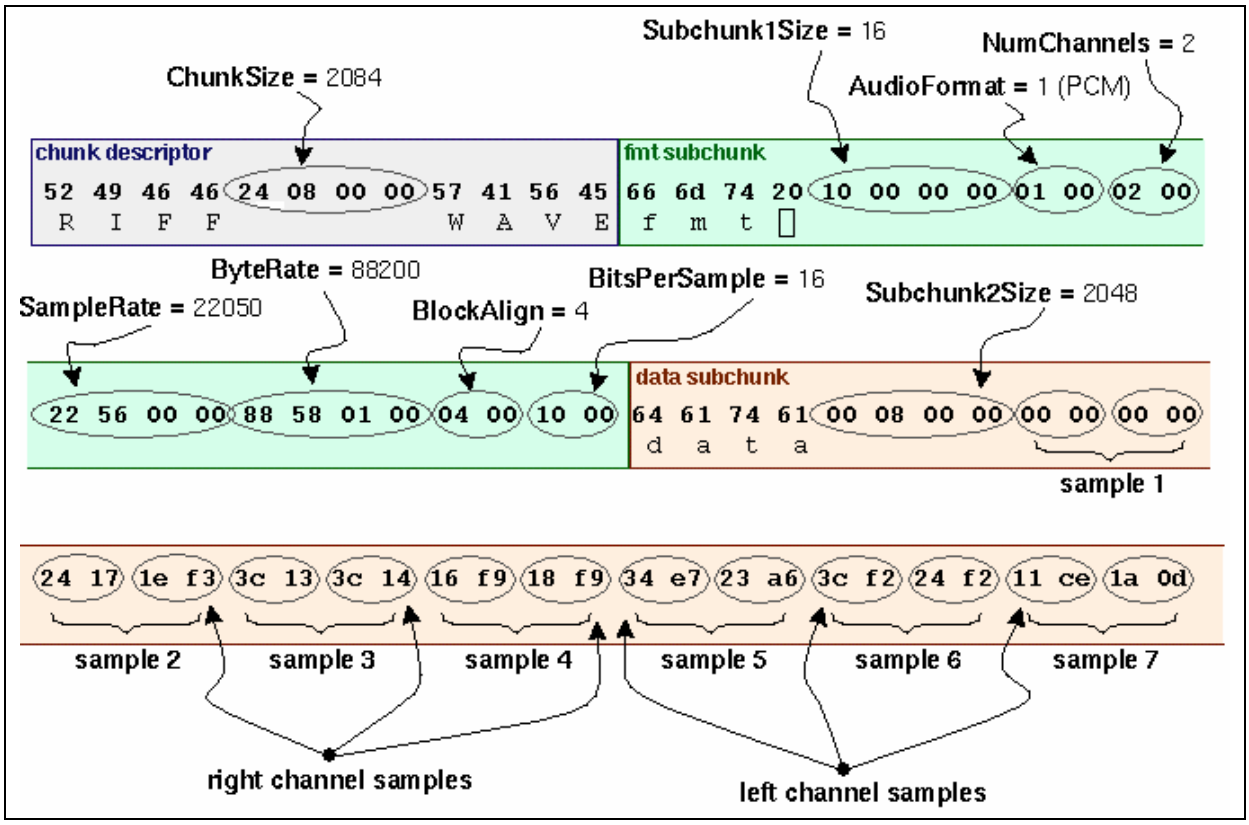


Figura 12 – Diagrama de seqüência “Insere Mensagem”

Para que se possa efetuar a esteganografia, é preciso primeiramente abrir um arquivo de áudio existente qualquer, desde que esteja no formato *wav*. Ao abrir o arquivo faz-se a leitura do seu cabeçalho e todas as informações são mostradas ao usuário. De acordo com

estas informações, é possível determinar a capacidade de armazenamento do arquivo, de acordo com o tamanho do setor de dados do arquivo (Figura 13).



Fonte: Wilson (2003)

Figura 13 – Exemplo de distribuição de um arquivo *wav*

Ao abrir o arquivo é feita também a verificação da existência de uma mensagem esteganografada no arquivo, através da leitura dos 64 primeiros *bits* menos significativos do setor de dados do arquivo, utilizando a técnica LSB, formando assim os 8 *bytes* do cabeçalho da mensagem.

4.2.4.3 CONSULTA MENSAGEM

Este diagrama é executado para extrair de um arquivo de áudio a mensagem ocultada, onde primeiramente é instanciado um objeto da classe TWaveUtility, chamando em seguida o método “Extract”, o qual extrai a mensagem do arquivo de áudio utilizando um objeto da classe TMessage (Figura 14). Ao extrair uma mensagem compactada, a classe TMessage fará a descompactação definindo uma entrada a um objeto da classe THuffman e chamando o método “Decompress”.

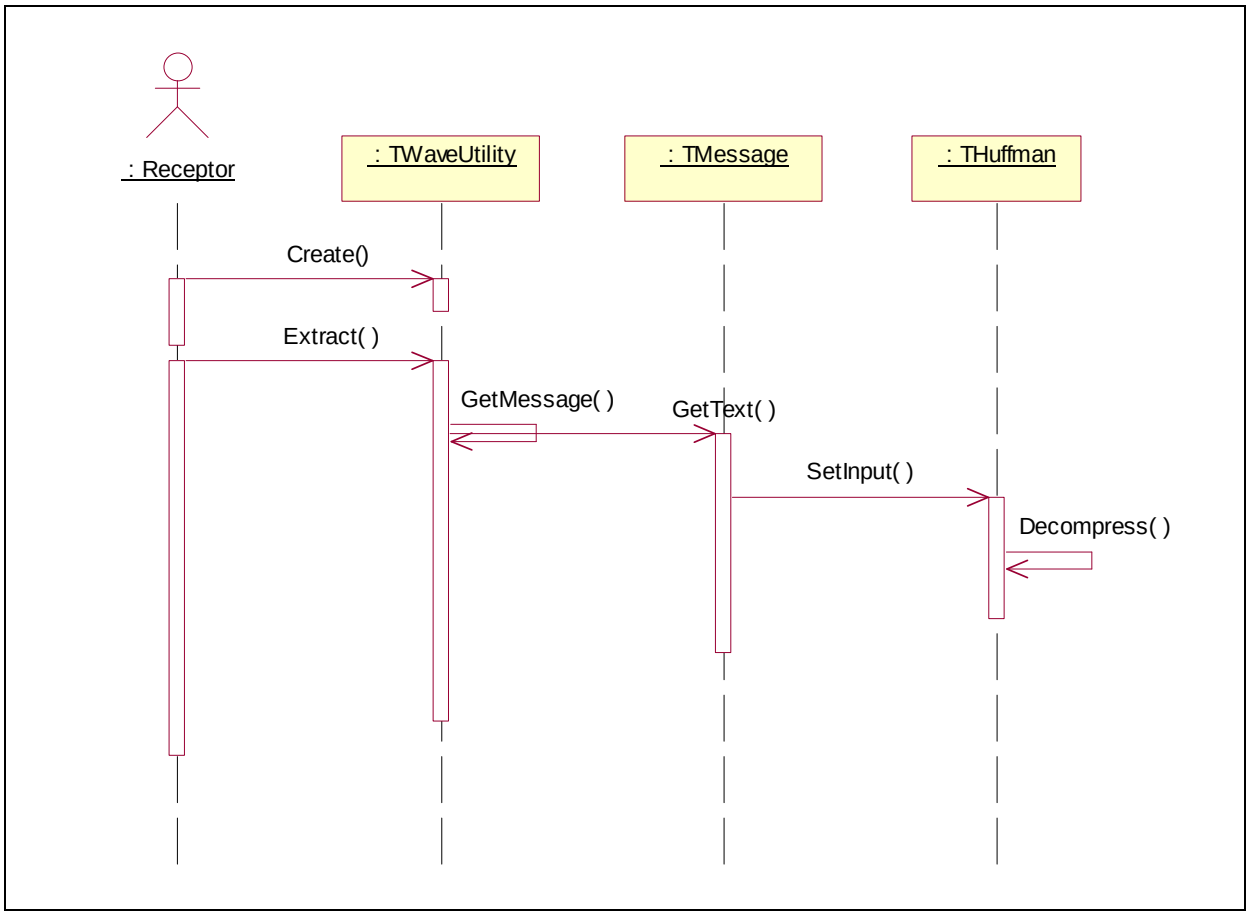


Figura 14 - Diagrama de seqüência “Consulta Mensagem”

4.3 IMPLEMENTAÇÃO

Neste capítulo são apresentados tópicos referentes à implementação do protótipo desenvolvido neste trabalho, utilizando o ambiente de programação Borland Delphi 7.0.

4.3.1 TÉCNICAS E FERRAMENTAS UTILIZADAS

O desenvolvimento do protótipo está dividido em 2 fases:

a) **esteganografia**: responsável por ocultar a mensagem de texto, seja esta compactada ou não, no arquivo de áudio. Para efetuar a ocultação da mensagem, é necessário ter como entrada no sistema um arquivo de áudio no formato *wav*, gerando como saída o arquivo com a mensagem esteganografada. Em outra situação, tendo como entrada um arquivo de áudio com uma mensagem já oculta, terá como saída a mensagem de texto, extraída do arquivo;

b) **compactação da mensagem**: através de um componente desenvolvido por terceiros, é possível efetuar a compressão da mensagem a ser oculta através do

algoritmo de Huffman, possibilitando assim a inserção de um maior número de caracteres na mensagem a ser esteganografada. Para efetuar a compactação é enviada ao componente a mensagem de texto a ser transmitida, o qual retorna a mensagem compactada.

A implementação da estaganografia através do protótipo se dá através de duas etapas. Na primeira é feita uma análise do cabeçalho do arquivo *wav*, o qual contém informações de sua formatação, para saber se este é um arquivo válido e para guardar as principais características que serão usadas para o processamento do algoritmo de codificação e decodificação.

Em uma segunda fase, tanto no processo de codificação quanto no de decodificação, são analisados os *bits* menos significativos de cada amostra sonora, ou *sample*, onde será inserida a mensagem ou então onde estarão os dados para a extração da mensagem a ser decodificada.

De acordo com informações de formatação do arquivo de áudio contidas em seu cabeçalho, pode-se verificar o tamanho de cada *sample* do arquivo, o qual pode possuir 8 ou 16 *bits*, isto é, 1 ou 2 *bytes*. A inclusão da mensagem é feita *bit* a *bit*, onde de cada *sample* lido do setor de dados do arquivo, é utilizado o bit menos significativo do último *byte* para a inclusão da informação a ser esteganografada.

Para a inclusão da mensagem, foi definido um cabeçalho da mensagem de 8 *bytes*, contendo um identificador de 4 *bytes*, indicando que há uma mensagem esteganografada no arquivo de áudio e se a mesma está de forma compactada ou não dentro do arquivo, seguido por outros 4 *bytes* contendo o tamanho da mensagem esteganografada (Figura 15).

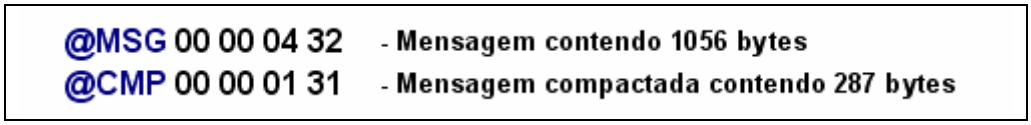


Figura 15 – Cabeçalhos de mensagens esteganografadas.

Ao ocultar qualquer mensagem em um arquivo, é inserido primeiramente seu cabeçalho. Desta forma, ao se abrir um arquivo de áudio qualquer pode-se fazer a verificação da existência de uma mensagem esteganografada a partir da leitura dos primeiros 32 *bits*

menos significantes do arquivo de áudio, formando assim os 4 *bytes* do identificador de uma mensagem. Caso a informação formada pelos quatro primeiros bytes lidos não corresponda a um dos identificadores, conclui-se que não há mensagem oculta no arquivo (Quadro 1).

```
// Calcula o tamanho de cada sample a ser lido no arquivo de áudio
SampleSize := SourceStream.BitsPerSample div 8;
// Le apenas os 4 primeiros bytes do arquivo através do método LSB
while (Stream.Size < 4) do
begin
   bit := 1;
   // Faz a leitura de cada sample do arquivo
   while bit <= 8 do
   begin
      SourceStream.wStream.Read(WavBuffer, SampleSize);
      WavBits := IntToBin(WavBuffer[SampleSize], 8);
      MsgBits[bit] := WavBits[8];
      Inc(bit);
   end;
   // Inclue o byte lido, após leitura dos 8 bits LSB
   MsgByte := BinToInt(MsgBits);
   Stream.Write(MsgByte, SizeOf(MsgByte));
   // Retira cabecalho da mensagem oculta ('@MSG' ou '@CMP')
   if (Header = '') and (Stream.Size = 4) then
   begin
      Stream.Position := 0;
      Stream.Read(Data, 4);
      for bit := 1 to 4 do
         Header := Header + char(Data[bit]);
      // Verifica se o arquivo contém mensagem oculta através de seus IDs
      if (Header = '@MSG') then
         result := Hidden
      else
      if (Header = '@CMP') then
         result := Compressed;
   end;
end;
```

Quadro 1 – Verificação da existência de uma mensagem em um arquivo *wav*

4.3.2 OPERACIONALIDADE DA IMPLEMENTAÇÃO

Para a utilização deste protótipo é necessária a existência de um arquivo *wav*. Tendo este arquivo um formato válido pode-se incluir ou então extrair uma mensagem do mesmo. Os processos para ocultar ou extrair uma mensagem ficam habilitados levando em consideração o estado do arquivo de áudio. Caso o arquivo de áudio possua um formato válido, o botão para incluir uma mensagem esteganografada fica habilitado. Se o arquivo possuir uma mensagem oculta, o botão para extraí-la do arquivo fica habilitado, bem como o botão para ocultar uma mensagem caso o usuário tenha necessidade de incluir uma nova mensagem esteganografada,

substituindo a existente. Caso o arquivo aberto não esteja em um formato adequado, os botões para inclusão e extração da mensagem permanecem desabilitados, informando ao usuário da incompatibilidade do arquivo.

A Figura 16 apresenta a tela principal do protótipo, onde a mesma pode ser dividida em três partes, descritas a seguir:

a) **informações do arquivo**: na parte esquerda superior da tela estão todas as informações do arquivo *wav* utilizado, seja este esteganografado ou não. Primeiramente pode-se visualizar as informações de formatação de áudio do arquivo, seguidas pelo tamanho do arquivo e sua capacidade de armazenamento de dados esteganografados, cálculo feito de acordo com o tamanho do arquivo e sua formatação. Abaixo são disponibilizadas informações sobre a existência e compactação de uma mensagem oculta no arquivo, caso exista;
b) ***wav player***: na parte inferior esquerda da tela encontra-se um pequeno *player*, onde é possível fazer a execução tanto do arquivo de áudio original, quanto o arquivo resultante do processo de esteganografia, isto é, o arquivo contendo a mensagem oculta;
c) **mensagem**: na parte direita do protótipo encontra-se a área para digitação da mensagem a ser oculta no arquivo. Nesta área será visualizada também a mensagem extraída do arquivo de áudio quando o mesmo contiver uma mensagem oculta. Na parte inferior da área destinada para a mensagem pode-se observar informações quanto ao tamanho ocupado pela mensagem, tanto na forma compactada quanto na forma normal.

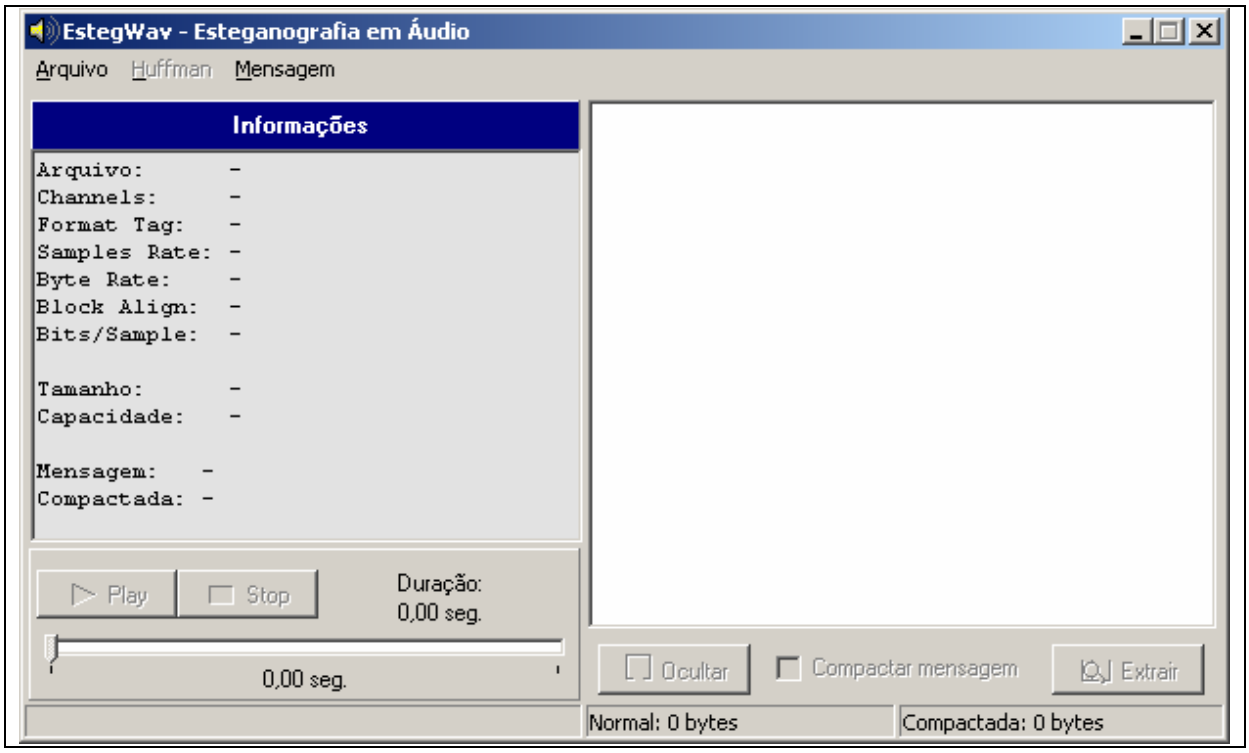


Figura 16 – Tela inicial do protótipo EstegWav.

As informações referentes ao arquivo *wav* são disponibilizadas ao usuário sempre que for aberto algum arquivo de áudio, sendo atualizadas também sempre que o usuário efetuar a esteganografia dos dados, através do botão “Ocultar”. O *wav player* também estará habilitado ao usuário assim que algum arquivo de áudio for carregado, possibilitando que o usuário ouça o arquivo sonoro original, bem como o arquivo contendo a mensagem oculta.

4.3.2.1 IMPLEMENTAÇÃO DO CASO DE USO “INSERE MENSAGEM”

O caso de uso “Insere mensagem” ocorre quando um usuário, neste caso o emissor, oculta uma mensagem de texto, seja esta compactada ou não, em um arquivo de áudio, a fim de enviar a uma outra pessoa, o receptor.

Para a inclusão da mensagem, pode-se utilizar o espaço disponível para digitação, ou então abrir um arquivo de texto existente. O limite de caracteres a serem ocultados depende da capacidade do arquivo, o qual é calculado e indicado ao usuário ao abrir um arquivo de áudio. Esta capacidade depende da quantidade de *bits* menos significativos disponíveis no arquivo, isto é, do tamanho do setor de dados do arquivo e da quantidade de *bits* de cada *sample*. A fim de aumentar a quantidade de caracteres da mensagem a serem transmitidos, o usuário tem a opção de compactação da mensagem através da opção de “Compactar

mensagem". Esta opção utiliza-se do método estatístico de compressão de Huffman, o qual gera a representação dos caracteres por um sistema de codificação em árvore binária, o que impede a ambigüidade na análise do código com a redução da redundância na codificação de símbolos.

Selecionado o arquivo de áudio e a mensagem de texto basta clicar no botão "Ocultar" (Figura 17) para ocultar a informação no arquivo de áudio, retornando uma mensagem informando algum problema ou o sucesso no processo de esteganografia. As informações do novo arquivo contendo a mensagem esteganografada serão atualizadas, onde o botão para extrair o texto passará a estar habilitado ao usuário.

Este processo para inclusão de uma mensagem pode ser visualizado na Figura 17, onde é incluída uma mensagem em um arquivo *wav* padrão.

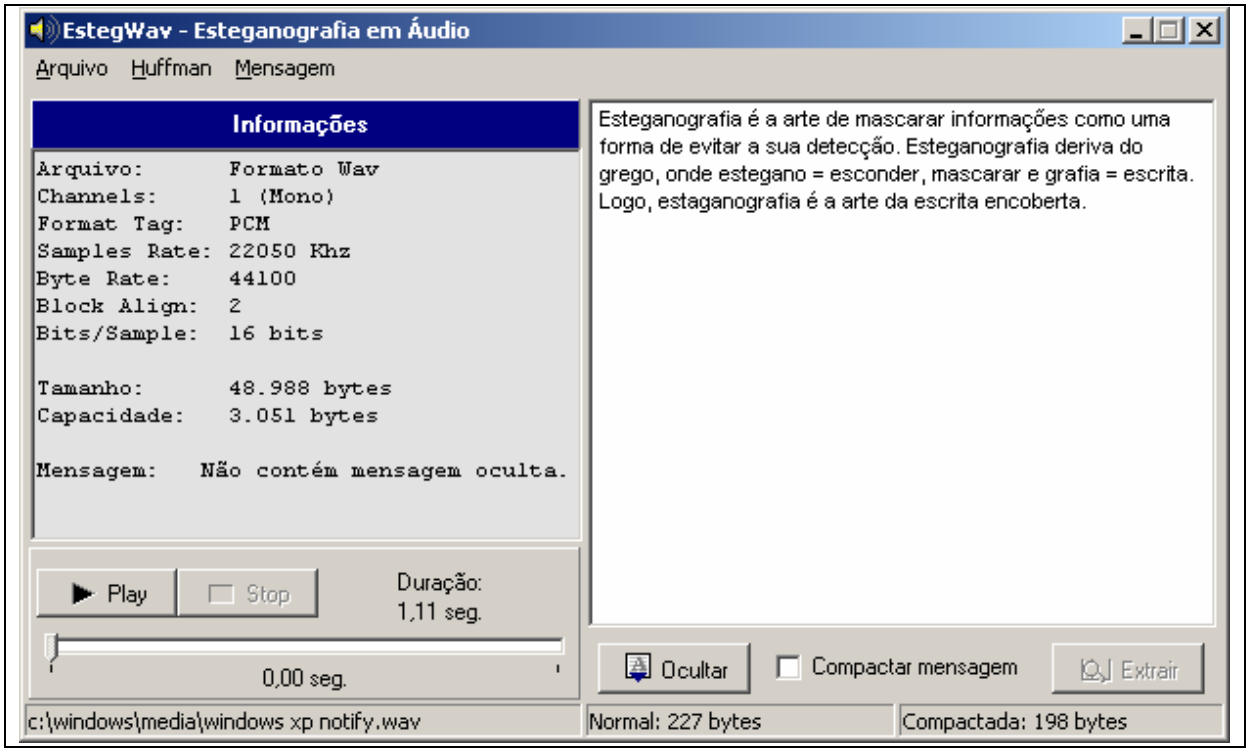


Figura 17 – Inclusão de uma mensagem de texto em um arquivo de áudio.

Finalizado o processo de esteganografia da mensagem, o usuário precisa apenas salvar o novo arquivo de áudio com a mensagem oculta.

4.3.2.2 IMPLEMENTAÇÃO DO CASO DE USO “CONSULTA MENSAGEM”

O caso de uso “Consulta Mensagem” ocorre quando um usuário receptor do arquivo de áudio abre o arquivo recebido. Com base nas informações lidas do cabeçalho do arquivo *wav* e dos 8 primeiros *bytes* formados pela leitura através do método LSB, é possível verificar se há uma mensagem esteganografada no arquivo e se a mesma encontra-se de forma compactada ou não.

Para efetuar e extração da mensagem esteganografada, basta apenas o usuário clicar no botão “Extrair” (Figura 18) e visualizar a mensagem no espaço reservado para a mesma. Caso a mensagem esteja compactada no arquivo, ao extraí-la é feita automaticamente a descompressão da mensagem, não precisando de nenhuma intervenção do usuário para este processo.

O processo de consulta de uma mensagem em um arquivo de áudio pode ser visualizado na Figura 18.

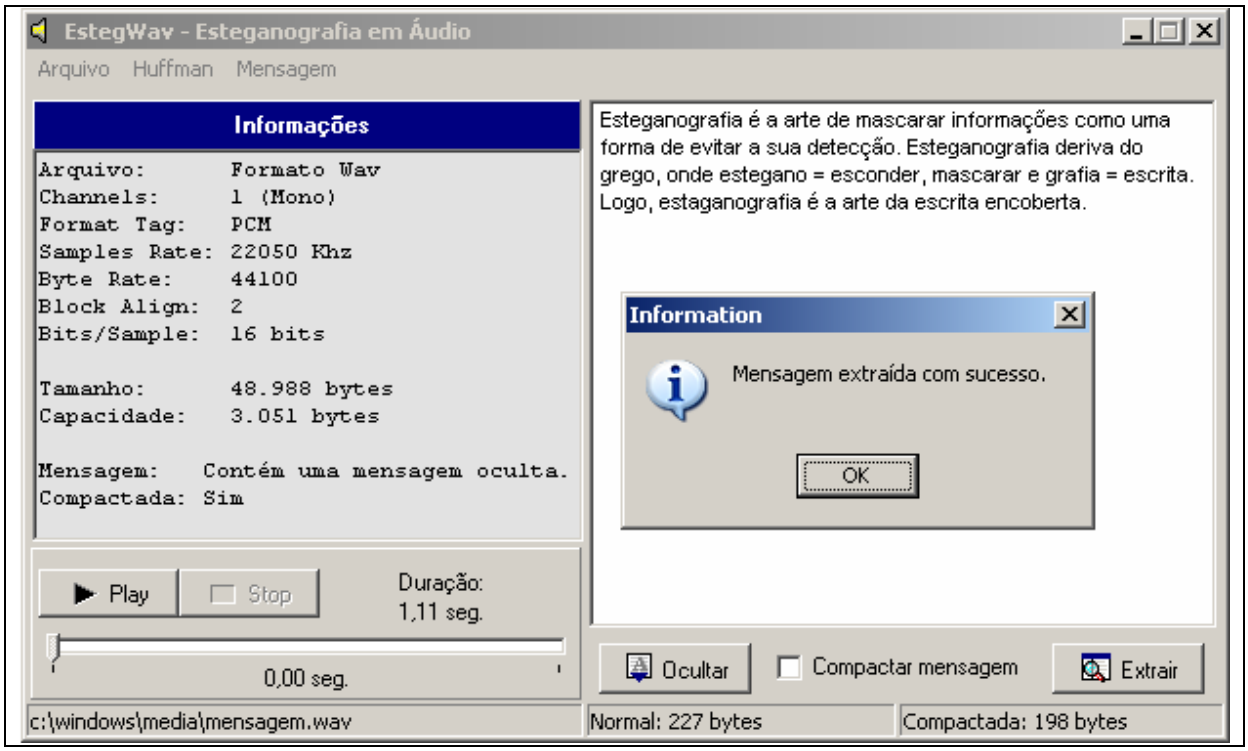


Figura 18 – Extração de uma mensagem oculta.

4.4 RESULTADOS E DISCUSSÃO

Para uma visualização mais clara dos resultados obtidos com o uso de esteganografia em arquivos de áudio, será demonstrado um exemplo prático do processo e uma comparação das ondas geradas a partir dos arquivos de áudio, antes e depois da esteganografia de uma mensagem.

Deseja-se ocultar a mensagem “ABC” em um arquivo. Definida a mensagem, é construído o cabeçalho da mesma, conforme visto anteriormente. O primeiro campo do cabeçalho contém o identificador “@MSG” ou “@CMP”, indicando se a mensagem será esteganografada de forma compactada ou não, seguido pelo tamanho da mensagem.

A representação binária para a mensagem não compactada “ABC”, precedida pelo seu identificador e tamanho é a seguinte: 01000000 01001101 01010011 01000111 00000000 00000000 00000000 00000011 01000001 01000010 01000011. Essa seqüência de *bits* será incluída no setor de dados do arquivo, o qual tem início logo após o cabeçalho do arquivo. Para isso é utilizado o último *bit* de cada amostra lida do arquivo. Em arquivos com tamanho da amostra de 2 *bytes* será utilizado o último *bit* apenas do segundo *byte* da amostra.

Para armazenar esta mensagem em um arquivo com tamanho de amostra de 2 *bytes*, são necessários 176 *bytes* (11 *bytes* x 8 *bits* x 2 *bytes*), enquanto em um arquivo com tamanho de amostra de 1 *bytes*, são necessários 88 bytes do arquivo de áudio (11 *bytes* x 8 *bits*).

Na esteganografia de uma mensagem compactada é utilizado o identificador “@CMP”, seguido pelo tamanho da mensagem e a mensagem em si. A mensagem compactada terá inicialmente uma lista de caracteres que formam a árvore de Huffman, a qual será necessária para a decodificação da mensagem. Em seguida é encontrada a seqüência de caracteres codificados que ao efetuar a descompressão da mensagem, formarão a mensagem original.

Na Figura 19 gerada a partir de um editor de arquivos de áudio, pode-se visualizar uma amostra das ondas geradas por um arquivo de áudio original, antes de efetuar a esteganografia. Este trecho do arquivo corresponde a uma pequena amostra de um arquivo *wav* padrão, cerca de 0,01 segundo, de um arquivo com freqüência de amostragem de 22,05 kHz e tamanho da amostra de 16 *bits*.

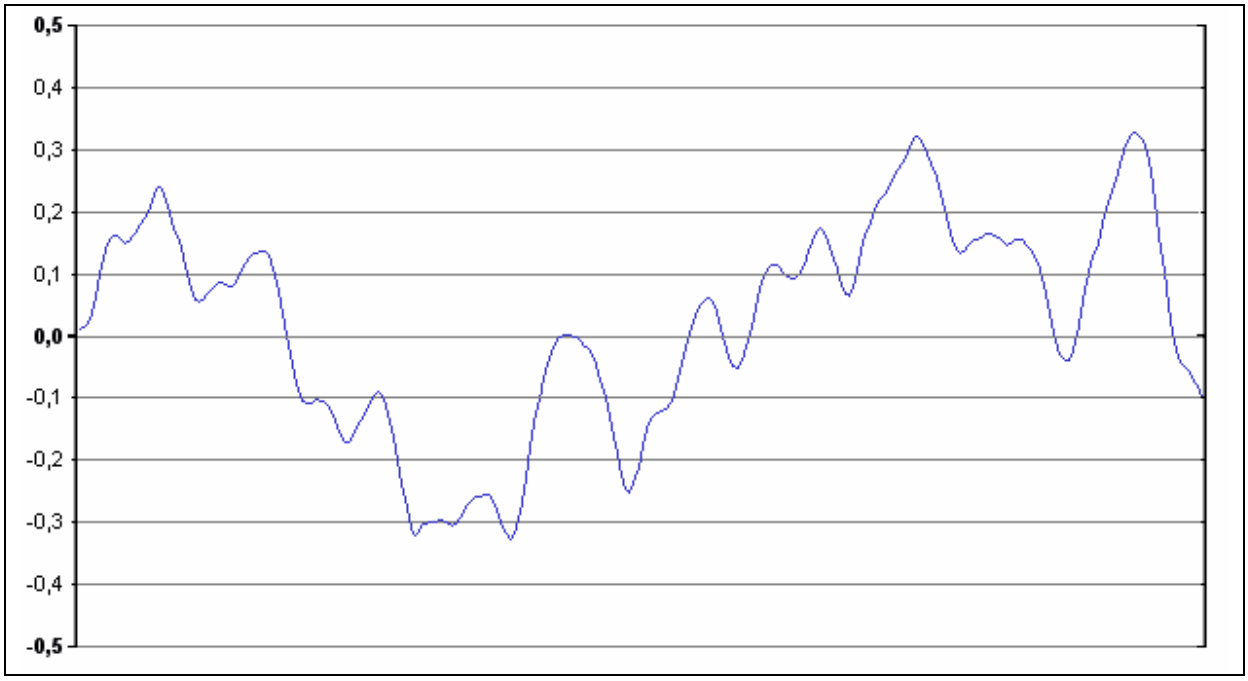


Figura 19 – Equalização gráfica do arquivo de áudio original

Após a esteganografia de uma mensagem de texto no arquivo percebe-se pequenas alterações nas ondas do arquivo de áudio, visualizados na Figura 20, onde as maiores alterações foram destacadas. Estas alterações causadas pelas modificações dos *bits* menos significativos das amostras de som não comprometem em nenhum aspecto o arquivo de áudio.

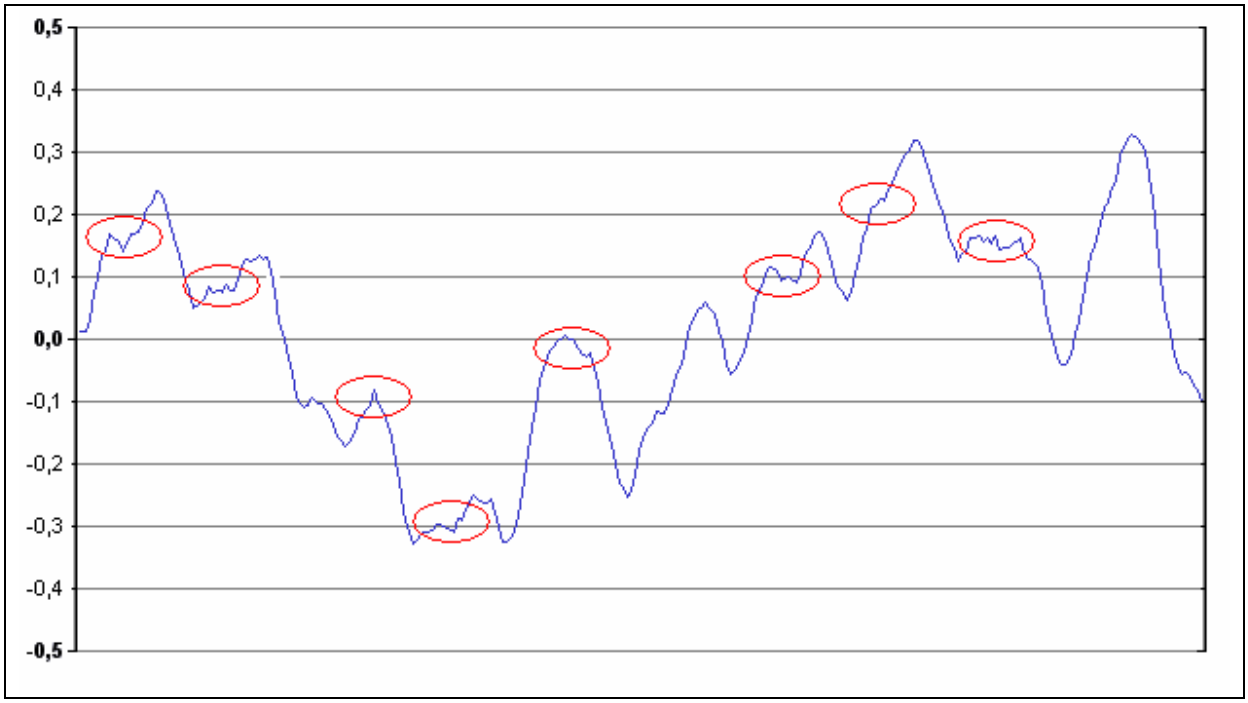


Figura 20 – Equalização gráfica do arquivo de áudio após esteganografia

Para o ouvido humano estas pequenas modificações são imperceptíveis, não havendo possibilidade sequer de distinção entre o arquivo original e o arquivo contendo a imagem estaganografada. Apenas através da análise de gráficos de ondas, ou através de aplicativos para comparação de arquivos ou editores em hexadecimal, pode-se verificar uma alteração dos dados do arquivo de áudio.

Para extrair a mensagem é feita a leitura do arquivo, uma amostra seguida da outra, retirando o *bit* menos significativo para formação da mensagem. A cada conjunto de 8 *bits* lidos, é formado um caractere da informação.

Os primeiros 4 *bytes* lidos formam a identificação da mensagem, seguidos por outros 4 *bytes* formando o tamanho da mensagem a ser lida, determinando assim a quantidade de *bytes* restantes para leitura, os quais formarão a mensagem esteganografada. No caso de uma mensagem compactada, indicada na leitura do primeiro identificador, a mensagem será lida por inteiro para em seguida efetuar a descompressão da mesma, disponibilizando ao usuário a mensagem decodificada.

5 CONCLUSÕES

Foram apresentados e analisados métodos de esteganografia para aplicação em arquivos de áudio, e técnicas para compressão de dados, para utilização em conjunto com a esteganografia.

A aplicação da esteganografia mostrou-se bastante eficaz na transmissão de dados de forma segura, tornando muito difícil sua percepção ou violação. A compressão dos dados esteganografados, além de permitir ocultar uma maior quantidade de informação, ajuda a tornar sua violação ainda mais difícil, pois os dados não ficam disponibilizados dentro do arquivo em uma seqüência lógica. Para que a informação possa ser visualizada de forma correta, a mesma tem que passar por um processo de decodificação, através do algoritmo de Huffman.

Uma das limitações do protótipo é a aplicação da esteganografia apenas em arquivos *wav*. Neste ponto, a compressão de dados mostrou-se bastante eficaz e muito útil no desenvolvimento do protótipo, pois este formato de arquivo de áudio tende a ocupar um espaço maior em disco. A compressão dos dados permite que sejam utilizados arquivos menores para o armazenamento de informações esteganografadas, facilitando assim a transmissão das informações.

Foram encontradas algumas dificuldades no levantamento de informações quanto a técnicas de esteganografia em arquivos de áudio, visto que a maioria do material encontrado trata de métodos para esteganografia em imagens.

5.1 EXTENSÕES

Como extensão deste trabalho pode-se estudar outras técnicas de esteganografia em arquivos de áudio, além da aplicação em arquivos áudio compactados, como o *mp3*, formato bastante utilizado atualmente para o armazenamento de áudio digital.

Outra sugestão é a utilização da esteganografia para ocultar além de mensagens de textos, outros formatos de arquivos.

Para garantir uma segurança ainda maior, pode-se acrescentar também a utilização de técnicas de criptografia em conjunto com a esteganografia.

REFERÊNCIAS BIBLIOGRÁFICAS

BRUNO, Sergio V. B. **Compressão de dados sem perda de informação usando algoritmos de recorrência de padrões**. Rio de Janeiro, maio 2002. Disponível em: <http://www.cetuc.puc-rio.br/~sbruno/files/reltz.pdf>. Acesso em: 05 set. 2004.

CAMPELLO, Rafael S; WEBER, Raul F. **Sistemas de detecção de intrusão**. Rio Grande do Sul, 2001. Disponível em: <http://www.inf.ufrgs.br/~gseg/producao/minicurso-ids-sbrc-2001.pdf>. Acesso em: 15 ago. 2004.

CANTARELLI, Elisa M. P. **Compressão de dados.** Rio Grande do Sul, 2001. Disponível em: <http://www.fw.uri.br/~elisa/compressao2.pdf>. Acesso em: 18 set. 2004.

HETZL, Stefan; MÜLLER, Didier; SELLARS, Duncan. **A esteganografia**. Paraná, out. 2002. Disponível em: <http://www.numaboa.com/criptologia/stegano/index.php>. Acesso em: 18 mar. 2004.

JASCONE, Fábio Luis T. **Protótipo de software para ocultar texto criptografado em imagens digitais.** 2003. 64 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

JOHN, Corinna. **Hiding Data in Wave Audio Files.** [S.I.], maio 2004. Disponível em: <http://www.codeproject.com/csharp/steganodotnet8.asp>. Acesso em: 20 out. 2004.

KUROSE, James F; ROSS, Keith W. **Redes de computadores e a internet:** uma nova abordagem. São Paulo: Pearson Brasil, 2003. 548 p.

LYRA, André Luiz et al. **Compressão sem perda:** método de Huffman e método de Lempel – ZIV. São Paulo, set. 2003. Disponível em: <http://www.dc.ufscar.br/~jander/ori203/grupo01c_2.pdf >. Acesso em: 07 abr. 2004.

MACÊDO, Rodrigo; TRINTA, Fernando. **Um estudo sobre criptografia e assinatura digital**. Pernambuco, set. 1998. Disponível em: <http://www.di.ufpe.br/~flash/ais98/cripto/criptografia.htm>. Acesso em: 19 mar. 2004.

PUTTINI, Ricardo S. **Criptografia e segurança de redes de computadores.** Brasília, dez. 2000. Disponível em: <http://www.redes.unb.br/security/seguranca.htm>. Acesso em: 25 jul. 2004.

ROCHA, Anderson de R. **Camaleão**: um software para segurança digital utilizando esteganografia. 2003. 108 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Federal de Lavras, Lavras.

SAJU, Vimil. **The Huffman Compression Algorithm**. Índia, ago. 2002. Disponível em: <http://www.howtodothings.com/showarticle.asp?article=313>. Acesso em: 19 set. 2004.

SILVA, Rogério da Guedes. **Segurança de redes de computadores**. Rio de Janeiro, jul. 1998. Disponível em: <http://www.cefetrio.hpg.com.br/ciencia_e_educacao/8/trabalhos/seguranca2/seguranca.htm>. Acesso em: 17 jul. 2004.

SOARES, Luiz F. G. **Redes de computadores**. Rio de Janeiro: Campus, 1995. 705 p.

STANG, David J.; MOON, Sylvia. **Segredos de segurança em rede**. Rio de Janeiro: Berkeley, 1994. xxvi, 986 p.

TANENBAUM, Andrew S. **Redes de computadores**. Rio de Janeiro: Campus, 2003. 786 p.

TAROUCO, Liane. **Grupo de Trabalho RNP**: aplicações educacionais em rede. Porto Alegre, jan. 2002. Disponível em: <http://penta3.ufrgs.br/RNP/cap3/3.2 Audio/>. Acesso em: 08 set. 2004.

TOMÁS, Gustavo Vieira et al. **Sistemas operacionais:** esteganografia. Goiás, nov. 2002. Disponível em: <http://www.inf.ufg.br/~eduardo/So/arq2002/esteganografia.doc>. Acesso em: 09 abr. 2004.

VAUGHAN, Tay. **Multimídia na prática**. São Paulo: Makron Books, 1994. 545 p.

VERÍSSIMO, Fernando. **Segurança em redes sem fio.** 2002. 90 f. Trabalho de Conclusão de Curso (Tópicos Especiais em Redes Integradas Faixa Larga) – Universidade Federal do Rio de Janeiro, Rio de Janeiro.

WILSON, Scott. **WAVE PCM soundfile format.** [S.I.], jan. 2003. Disponível em: <http://ccrma.stanford.edu/courses/422/projects/WaveFormat/>. Acesso em: 15 set. 2004.