FUNDAÇÃO EURÍPIDES SOARES DA ROCHA
CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA – UNIVEM
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

APARECIDO JOSÉ PEREIRA

ESTEGANOGRAFIA – IMPLEMENTAÇÃO DE UM SOFTWARE PARA OCULTAR MENSAGENS CRIPTOGRAFADAS EM IMAGENS

MARÍLIA
2009

APARECIDO JOSE PEREIRA

ESTEGANOGRAFIA - IMPLEMENTAÇÃO DE UM SOFTWARE PARA OCULTAR MENSAGENS CRIPTOGRAFADAS EM IMAGENS

Trabalho de Curso apresentado ao Curso de Bacharelado em Ciência da Computação da Fundação de Ensino "Eurípides Soares da Rocha", mantenedora do Centro Universitário Eurípides de Marília – UNIVEM, como requisito parcial para obtenção do grau de Bacharelado em Ciência da Computação.

Orientador:
Prof. Ms. PAULO AUGUSTO NARDI.

MARÍLIA
2009

PEREIRA, Aparecido José

Esteganografia - Implementação de um software para ocultar mensagens criptografadas em imagens / Aparecido José Pereira; orientador: Paulo Augusto Nardi. Marília, SP: [s.n.], 2009.

44 f.

Trabalho de Curso (Graduação em Bacharelado em Ciência da Computação) - Fundação de Ensino "Eurípides Soares da Rocha", mantenedora do Centro Universitário Eurípides de Marília – UNIVEM.

1. Esteganografia 2. Criptografia.

CDD: 005.82

CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO – AVALIAÇÃO FINAL

Aparecido José Pereira

ESTEGANOGRAFIA - IMPLEMENTAÇÃO DE UM SOFTWARE PARA OCULTAR MENSAGENS CRIPTOGRAFADAS EM IMAGENS

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação.

Nota: 8,5 (oito e meio)

Orientador: Paulo Augusto Nardi

1°. Examinador: Leonardo Castro Botega

2°. Examinador: Fábio Dacêncio Pereira

Marília, 02 de dezembro de 2009.

AGRADECIMENTOS

Ao Professor Paulo Augusto Nardi pelo direcionamento deste trabalho e a todos aqueles que me auxiliaram nesta etapa da minha vida e à força divina que me inspirou energias para a conclusão deste ciclo.

E por fim, àqueles que me trouxeram à vida e àqueles que em companhia de mim cresceram.

PEREIRA, Aparecido José **Esteganografia - Implementação de um software para ocultar mensagens criptografadas em imagens.** 2009. 44f. Trabalho de Curso (Bacharelado em Ciência da Computação) – Centro Universitário Eurípides de Marília, Fundação de Ensino "Eurípides Soares da Rocha", Marília, 2008.

RESUMO

Com o crescente uso das redes de computadores e da internet por pessoas e organizações para conduzir seus negócios, surge à necessidade de se utilizar mecanismos para garantir a segurança das transações com informações confidenciais. Duas Técnicas que podem ser utilizadas com o intuito de tornar confidencial a informação são: a Criptografia e a Esteganografia. Através da criptografia podemos ocultar o significado de uma informação e a esteganografia, diferindo da criptografia, oculta a existência da informação. Neste trabalho é apresentado um estudo sobre tais conceitos e é implementado um software para utilizar essas duas técnicas em conjunto. O software é modelado nos diagramas de Casos de Uso, Classes e Componentes da UML (Linguagem de Modelagem Unificada) e é implementado em linguagem Java.

Palavras-Chave: Esteganografia, Criptografia.

PEREIRA, Aparecido José **Esteganografia - Implementação de um software para ocultar mensagens criptografadas em imagens.** 2009. 44f. Trabalho de Curso (Bacharelado em Ciência da Computação) – Centro Universitário Eurípides de Marília, Fundação de Ensino "Eurípides Soares da Rocha", Marília, 2008.

ABSTRACT

With the increasing use of computer networks and the Internet by individuals and organizations to conduct their business, the use of mechanisms to ensure the security of transactions with confidential information. Two techniques that can be used in order to make confidential the information are cryptography and steganography. Through cryptography the meaning of information and steganography, differing in cryptography, hides the existence of information. This work presents a study on such concepts and a proposal for implementation of software to use these two techniques together. The software is modeled in UML Use Case, Class and Components of UML (Unified Modeling Language) and is implemented in programming language Java.

Keywords: Steganography, Cryptography.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

AES – Advanced Encryption Standard (Padrão de Criptografia Avançada)

DES – Data Encryption Standard (Padrão de Criptografia de Dados)

JPEG – Joint Photographic Expert Group (Grupo de Peritos Fotográfico Comum)

LSB – Last Significant Bit (Inserção no Bit Menos Significativo)

PNG – Portable Network Graphics (Gráficos de Rede Portátil)

UML – Unified Modeling Language (Linguagem de Modelagem Unificada)

SUMÁRIO

INTRODUÇÃO

Transferir mensagens e dados com segurança são preocupações constantes na vida das pessoas. Muitas vezes nos deparamos com a necessidade de transmitir uma mensagem secreta, e temos algum receio de que o conteúdo dessa mensagem seja descoberto por algum estranho (HINZ, 2000, p.12). Diante dessa situação surge a necessidade de se utilizar mecanismos para garantir a segurança das transações de informações confidenciais.

A segurança na transferência de dados é de extrema importância para o meio digital. A segurança na transferência de dados é bastante enfatizada, principalmente ao quando se imagina a possibilidade pessoas não autorizadas ter acesso as suas informações, pessoas essas que surgem com meios cada vez mais eficientes e sofisticados, com o objetivo de violar a segurança e a privacidade das comunicações.

Mensagens e dados devem ser protegidos para que somente as pessoas autorizadas consigam acessá-los.

Este trabalho tem por objetivo estudar e apresentar uma breve visão sobre esteganografia e criptografia, e utilizar o conhecimento adquirido com o estudo dessas duas técnicas, construir uma aplicação para ocultar mensagens criptografadas em imagens digitais. O software é modelado nos diagramas Casos de Uso, Classes e Componentes da UML (Linguagem de Modelagem Unificada) e é implementado em linguagem Java, com objetivo de garantir uma maior segurança na transferência de informações confidenciais.

Por fim é apresentado um estudo de caso demonstrando o funcionamento do software, comparando a imagem antes e após a inserção da mensagem.

Este trabalho está dividido da seguinte forma:

Primeiro é apresentado à introdução; depois é apresentado o primeiro capítulo – Esteganografia – descreve o que é esteganografia relatando sua origem e sua importância; o segundo capítulo – Criptografia – descreve o que é a criptografia relatando sua origem e importância; o terceiro capítulo – Inserção no Bit menos significativo(LSB) – este capítulo tem por objetivo detalhar o método de esteganografia adotado no desenvolvimento do software para ocultar mensagens criptografadas em imagens; o quarto capítulo – Algoritmo DES – este capítulo tem por objetivo detalhar o algoritmo de criptografia adotado no desenvolvimento do software para ocultar mensagens criptografadas em imagens; o quinto capítulo – Desenvolvimento do software para ocultar mensagens criptografadas em imagens – mostra a modelagem do software com a UML; o sexto capítulo – Apresenta um estudo de caso, demonstrando o software e seu funcionamento.

Por fim são apresentadas a Conclusão e as Referências usadas para a realização do trabalho.

CAPÍTULO 1 – ESTEGANOGRAFIA

1.1 Introdução

Esteganografia, é uma palavra que possui origem grega, onde Stegano significa escondido ou secreto e Grafia: escrita ou desenho("escrita escondida") (PETRI, 2004, p.10).

A esteganografia é a arte da comunicação secreta, onde uma mensagem sigilosa é ocultada dentro de outra informação sem importância, de maneira que não seja possível identificar que há uma mensagem escondida. Essa mensagem sem importância pode ser, por exemplo, um arquivo de imagem, texto ou som (JASCONE, 2003, P. 34).

Na esteganografia, é possível destacar um conjunto de métodos desenvolvidos para comunicações secretas ao longo da história. Entre os métodos de esteganografia é possível destacar: micro pontos, tintas invisíveis, arranjo de caracteres, canais escondidos, assinaturas digitais, entre outras.

1.2 Histórico

A seguir são relatados alguns métodos que foram criados ao longo da história, com a finalidade de esconder informações dentro de outros objetos, sempre com o objetivo de se obter privacidade na comunicação.

Segundo PETRI (2004, p.10), um dos primeiros registros sobre esteganografia relata sobre a utilização de uma lebre morta, onde era escondida uma mensagem secreta em suas entranhas, e que a mesma era enviada até o seu destino.

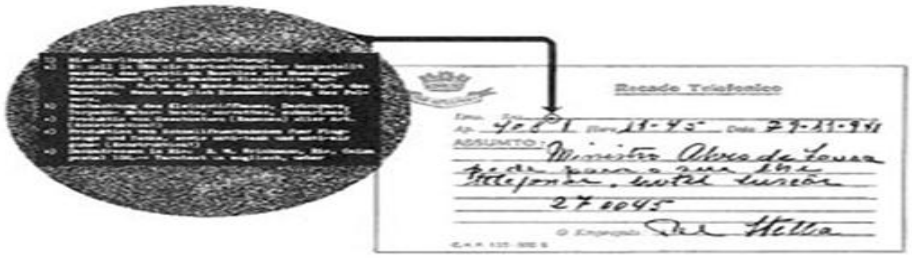
Outro método utilizado é o de cortar o cabelo de mensageiros, tatuando a mensagem em seu coro cabeludo, e após o crescimento do cabelo o mensageiro era enviado até o seu destino, onde novamente teria sua cabeça raspada para revelar a mensagem (PETRI, 2004, p.11).

Segundo ROCHA (2003, p.13), os métodos de esteganografia utilizados durante a Segunda Guerra Mundial eram baseados em tintas invisíveis. As tintas invisíveis, de forma geral, eram químicas que, ao serem misturadas com outras químicas, tornavam o resultado visível. Durante a Primeira Guerra Mundial, espiões alemães colocavam pequenos “pontos” de tinta invisível sobre letras de jornais e revistas de grande circulação. As folhas de revistas , quando aquecidas, revelavam a seqüência das letras ”pontuadas”.

Outro método que começou a ser utilizado na Segunda Guerra Mundial foi o de Micropontos, onde uma mensagem poderia ser fotografada e reduzida ao tamanho de um

ponto, podendo este ponto fazer parte de uma mensagem qualquer sendo o ponto de uma letra "i", ou um ponto final de frase de uma mensagem qualquer (JASCONE, 2003, p.33).

Figura 1.2.1: Método dos Micropontos.



Fonte: JASCONE (2003, p.33).

Outra método consiste em ocultar uma mensagem dentro de outra, um exemplo seria escrever uma mensagem utilizando a primeira letra de cada palavra de uma outra mensagem que não tenha importância (JASCONE, 2003, p.33).

Abaixo um exemplo da técnica descrita anteriormente:

O senhor Evandro quer usar este salão temporariamente. Relembre o fato ocorrido, isto poderia estragar relíquias, florais e imagens talhadas. Obrigado. Lendo apenas a primeira letra de cada palavra temos: O SEQUESTRO FOI PERFEITO.

No Brasil, algumas provas de concursos públicos, até meados da década de 80, eram corrigidas com a utilização de cartões perfurados. Ao colocar o cartão perfurado sobre os cartões-resposta dos candidatos, era revelado se o candidato acertou ou errou as questões da prova (ROCHA, 2003, p.16).

1.3 Esteganografia X Criptografia

Embora as duas tenham um objetivo em comum que é proteger a informação de terceiros, é importante salientar a diferença entre a esteganografia e a criptografia. Enquanto a criptografia oculta o significado da mensagem, a esteganografia oculta a existência da mensagem.

A esteganografia é um ramo da Criptologia que consiste em ocultar uma informação dentro de outra informação sem importância. Ao invés de cifrá-la, técnica utilizada na criptografia de mensagens, o que se busca é tornar imperceptível a sua presença.

De acordo com PETRI (2004, P.16), uma diferença entre essas duas técnicas é que na esteganografia a mensagem não está codificada, mas não é possível notar a sua presença, enquanto que na criptografia a mensagem está codificada, porém a sua presença pode ser identificada.

Essas duas técnicas podem ser combinadas para aumento da privacidade e segurança.

1.4 Aplicações

A seguir são apresentadas algumas aplicações de Esteganografia descritas por GOIS (apud PETRI, 2004, p. 19).

Proteção de direitos autorais - É quando um autor de uma imagem, texto, vídeo ou som deseja assinar sua obra, preservando dessa forma sua autoria.

Certificação e controle de acesso - Muito utilizado em documentos oficiais, por exemplo, carteiras de identidade e passaportes. Ela permite validar as informações presentes em um documento por meio da sua vinculação. A utilização de cartões de identificação seria um exemplo de utilização, onde o identificador do cartão é mostrado em forma textual e também escondido como uma marca na própria fotografia impressa no cartão.

Legenda em imagens e vídeos - Nesse caso as legendas seriam ocultadas em vídeos ou imagens, de uma forma que possam ser apresentadas ou extraídas quando houver necessidade. Esse método, por exemplo, também pode ser aplicado em som ou informações adicionais de um filme de DVD.

Autenticação e verificação da integridade - Imagens não podem ser aceitas como provas de julgamento, pois são facilmente manipuladas ou alteradas. Ao usar marcas digitais essa situação pode ser alterada, viabilizando o uso desse tipo de mídia, por exemplo, o uso de câmeras digitais.

Controle de cópias – Muitas empresas procuram métodos para controlar e restringir a possibilidade de se realizar cópias de seus produtos. Na fabricação de DVDs, por exemplo, filme comercial, poderia conter algum tipo de marca digital, para evitar ou restringir cópias do filme.

1.5 Técnicas de Esteganografia em Imagens

Inserção no bit menos significativo – O método mais utilizado para armazenar informações em imagens digitais é o LSB (*Last Significant Bit)*. O método consiste em alterar o bit menos significativo de cada pixel, ou de cada cor da imagem para ocultar a mensagem (JASCONE, 2003, P.28).

Técnicas de filtragem e mascaramento – As Técnicas de filtragem e mascaramento só podem ser aplicadas nas imagens em tons de cinza. Essas técnicas escondem a informação pela marcação de uma imagem, semelhante ao funcionamento das marcas d'água em papel. Uma das vantagens dessa técnica é que devido ao fato da marca d'água ser integrada na

imagem, não haverá problema se a imagem passar por algum método de compressão. Outra vantagem é que o olho humano não consegue distinguir as mudanças na imagem, sendo assim, essas técnicas não podem percebidas pelo olho humano (JASCONE, 2003, p.40).

Algoritmos e transformações - Os algoritmos de transformação trabalham com formas mais refinadas de manipulação de imagens, entre elas estão: o brilho, a saturação e a compressão das imagens. A compressão, que é o principal inimigo do LSB, não afeta as técnicas de transformações. Por esse motivo essas técnicas são tidas como as técnicas mais refinadas de mascaramento de informações em imagens (ROCHA, 2003, apud PETRI, 2004, p.23).

CAPÍTULO 2 – CRIPTOGRAFIA

2.1 Introdução

Do Grego kryptós, "escondido", e gráphein, "escrever", a criptografia é o estudo e aplicação de técnicas que tem por objetivo transformar uma informação original e legível para outra completamente ilegível, que a torna difícil de ser lida por alguém não autorizado. Sua interpretação só pode ser feita caso seja conhecida sua "chave secreta".

A criptografia pode ser entendida como um conjunto de métodos e técnicas para cifrar ou codificar informações legíveis por meio de um algoritmo, convertendo um texto original em um texto ilegível, sendo possível mediante o processo inverso recuperar as informações originais.

A criptografia é tão antiga quanto à própria escrita, uma vez que já estava presente de escrita hieroglífica dos egípcios. Os Romanos utilizavam códigos secretos para comunicar planos de batalha. Com as guerras mundiais e a invenção do computador a criptografia cresceu incorporando complexos algoritmos matemáticos (EDWARD; PEREIRA; CHIARAMONTE, 2005, p.22).

2.2 Histórico

A criptografia fornece a pessoas devidamente autorizadas, a determinados processos, serviços básicos de: autenticidade, confidencialidade, integridade e não repúdio para os dados. A criptografia tem como funcionalidade a alteração de uma mensagem original para outra cifrada. Para conseguir isso, a mensagem deve passar por um processo de cifragem, que consiste basicamente no emprego da mensagem a um mecanismo que, através de uma chave "específica", cifre a mesma. Ou seja, após a mensagem ser empregada ao algoritmo de criptografia com esta chave secreta a mesma será totalmente alterada ficando completamente ilegível. E um processo contrário, que é a decifragem onde, novamente, através da chave específica, pode-se converter os dados cifrados para a mensagem original.

A Figura 2.2.1 exemplifica a funcionalidade da criptografia:

Figura 2.2.1: Funcionalidade da criptografia.

Fonte: HINZ (2000, p.14)

Com intuitos históricos, de exemplificar técnicas de criptografia utilizadas antigamente, pode-se comentar sobre as cifras de substituição e cifras de transposição. O funcionamento das cifras de substituição consiste basicamente em “trocar os bits, caracteres ou blocos de caracteres por outros”, ou seja, troca-se um caractere por outro seguindo uma tabela de substituição. Por exemplo, se na tabela de substituição é indicado que se deve alterar os caracteres em cinco posições, a consoante B, após ser cifrada, irá assumir o valor da consoante G.

As técnicas de substituição é o tipo de cifragem mais simples, conhecido e fácil de ser quebrada. É necessário somente analisar a freqüência com que aparecem os caracteres e ir comparando com a freqüência dos caracteres em determinado idioma. Por exemplo, em português, o caractere que mais vezes aparece em textos é a vogal E; então, se a mensagem a ser analisada for em português, provavelmente o caractere que aparecer mais vezes no texto cifrado é a letra E, e assim realiza-se uma análise sobre toda a mensagem.

Segundo EDWARD, PEREIRA, CHIARAMONTE (2005, p.22), um exemplo famoso da utilização de cifras de substituição é a Cifra de César ou Código de César. Criado pelo Imperador Julio César com o objetivo de possibilitar uma comunicação de forma sigilosa com seus generais passando informações militares. A técnica consiste em alterar a posição dos caracteres em três posições à frente, ou seja, a vogal A se transformaria na consoante D após ser cifrada.

As cifras de transposição ou cifras de permutação, por sua vez, consistem, segundo, “em reorganizar a ordem dos bits, caracteres ou bloco de caracteres”, ou seja, consistem em trocar as posições dos caracteres no texto. Como nas cifras de substituição é importante definir uma ordem para realizar a troca das posições.

Por exemplo, se pegar um texto com 20 caracteres, dividem-se estes caracteres em um vetor de 20 posições, define-se uma ordem para realizar a transposição, de tal forma que todos os caracteres mudem de posição, ou seja, define-se que a posição um do vetor trocará de lugar com a posição 10, a posição dois com a posição 15, e assim por diante, até que toda a mensagem tenha trocado de posição.

Outro exemplo básico da utilização de transposição é, simplesmente, pegar uma palavra e embaralhar a posição dos caracteres.

2.3 Serviços Básicos

É necessário que alguns serviços básicos sejam assegurados para que a criptografia realmente seja um meio efetivo para proteção dos dados, são eles:

Autenticidade - Assegura a validade da transmissão, da mensagem e das entidades envolvidas. Conforme BURNETT e PAINE (2002, p. 10), "a autenticidade pode ser comparada à identificação de usuário - ID - requerido para retirar dinheiro de uma conta corrente".

Confidencialidade - Restringe o acesso à informação apenas às pessoas autorizadas, tornando, assim, o documento protegido. A confidencialidade seria a garantia de privacidade da comunicação.

Integridade - Tem como objetivo a garantia da não alteração dos dados, ou seja, garantia de que a mensagem recebida é exatamente igual à mensagem enviada, e não sofreu nenhuma alteração. Conforme BURNETT e PAINE (2002, p. 10), "A integridade é um mecanismo que informa se algo foi alterado, assim como um alarme da ferramenta".

Não-Repúdio - É a garantia de que determinada pessoa que participou de determinada operação realmente participou, não podendo negar a propriedade do ato. Ou "seja, um remetente não pode negar mais tarde que enviou determinada mensagem ou executou determinada ação", (SCHNEIER, 1996, p. 2).

2.4 Tipos de Criptografia

Existem atualmente dois tipos diferentes de criptografia: simétrica e assimétrica (PETRI, 2004, p. 29).

A criptografia simétrica utiliza uma única chave no processo de criptografia, e esta chave deve ser mantida em segredo.

A criptografia assimétrica utiliza duas chaves, matematicamente relacionadas, sendo uma delas para encriptar outra para decriptar. Uma das chaves é mantida em segredo, e a outra é divulgada.

A principal diferença encontrada nas duas técnicas é com relação às chaves utilizadas. A criptografia simétrica utiliza somente uma chave para realizar o processo de cifragem e decifragem da mensagem e a assimétrica utiliza duas chaves, uma pública e outra privada, sendo uma utilizada para cifrar e a outra para decifrar a mensagem.

Assim, observa-se que a criptografia assimétrica soluciona o problema de distribuição de chaves e facilita a comunicação segura entre quaisquer pessoas. Mas na criptografia

simétrica, o processo de cifragem e decifragem é extremamente mais rápido do que com a assimétrica.

2.4.1 Criptografia Simétrica

Conforme EDWARD, PEREIRA e CHIARAMONTE (2005, p. 37), na criptografia simétrica, os processos de cifragem e decifragem são feitos com uma única chave, ou seja, o destinatário e o remetente usam a mesma chave.

A Figura 2.4.1.1 ilustra o funcionamento de um algoritmo de criptografia simétrico:

Figura 2.4.1.1: Funcionamento da criptografia simétrica.

Fonte: SCHNEIER (1996, p. 28)

Se A quiser enviar uma mensagem para B, ela deverá empregar a mensagem que deseja enviar a um algoritmo e alimentá-lo com uma chave secreta. Assim, a informação vai ser encriptada e um atacante não pode visualizá-la; ela envia a mensagem para B e ele executa o processo reverso: emprega a mensagem cifrada ao algoritmo e o alimenta com a mesma chave secreta para poder decriptá-la.

A criptografia de chave simétrica até pode manter seguros seus dados, mas se precisar compartilhar com outras pessoas, é inevitável que você compartilhe a chave secreta também. Portanto, é extremamente necessário que se guarde e/ou transmita essa chave de forma segura, pois se um dos dois não armazenar de forma segura esta chave ou se a distribuição da mesma não for realizada de forma segura, ela pode ser roubada por um atacante, e utilizada para interceptar a mensagem ou até mesmo falsificar novas mensagens.

Existem dois tipos de algoritmos de criptografia simétrica, a cifragem de blocos que opera sobre blocos de dados e cifragens de fluxo que operam utilizando a técnica de criptografia de enchimento de uma única vez.

Conhece-se varias implementações de algoritmos simétricos, podendo citar como exemplos a cifra de César, cifras de transposição, algoritmo DES (Data Endrypition Standart), AES (Advanced Encryption Standard), 3DES dentre outros, sendo abaixo apresentando alguns dados históricos e conceitos básicos de dois algoritmos simétricos, o DES e o AES.

2.4.2 Criptografia Assimétrica

Conforme EDWARD, PEREIRA e CHIARAMONTE (2005, p. 37), na criptografia assimétrica cada pessoa têm um par de chaves, uma denominada chave pública, e outra denominada chave privada. A chave pública, como próprio nome diz, pode ser divulgada, enquanto a chave privada deve ser mantida em sigilo. Para enviar uma mensagem secreta, o transmissor usa a chave pública do destinatário para cifrar a mensagem, que utilizará sua chave privada para decifrar a mensagem, obtendo dessa forma a mensagem original.

Decorrente dos problemas com a distribuição das chaves, que é a necessidade de distribuir a mesma chave secreta entre as pessoas envolvidas no processo, tornou-se necessário a criação de ferramentas que pudessem fazer a distribuição das chaves de forma segura. Então, foram criados os algoritmos de criptografia de chave assimétrica ou de chave pública, como também são conhecidos.

Em meados da década de 70, o aluno Whitfield Diffie, graduado pela Stanford University, e o professor Martin Hellman investigaram a criptografia em geral e o problema de distribuição de chaves. Eles criaram o primeiro esquema de chave pública, no qual duas pessoas poderiam trocar informações utilizando meios de comunicação públicos, através de chaves secretas compartilhadas, ou seja, eles poderiam criar uma chave secreta trocando informações públicas. Este esquema ficou conhecido como DH (Diffie, Hellman), decorrente das iniciais dos nomes de seus criadores.

"O DH resolve o problema de distribuição de chaves, mas não é o algoritmo final. Ele não é efetivo para a criptografia, mas ainda é muito utilizado até hoje", cita BURNETT e PAINE (2002, p. 79).

Conforme SCHNEIER (1996, p. 31), pode-se definir a técnica da seguinte maneira: "A criptografia assimétrica utiliza duas diferentes chaves, uma pública e outra privada, de modo que é computacionalmente complexa e normalmente impraticável a dedução da chave privada por meio da chave pública".

Assim, qualquer pessoa pode cifrar a mensagem com a chave pública, mas jamais pode decifrá-la com a mesma chave, sendo necessário à chave privada para decifrá-la, que, teoricamente, só a pessoa autorizada, dona da chave pública informada e utilizada para cifrar a mensagem terá.

É relevante salientar que as duas chaves (pública e privada) detêm uma relação matemática entre elas. A geração das chaves (problema direto) é rápida e eficiente, já o problema reverso (a partir da chave pública definir a chave privada), é oneroso e proibitivo.

Segundo SCHNEIER (1996, p. 31), pode-se definir o processo encriptação da seguinte maneira:

"Matematicamente, o processo de encriptação é baseado nas funções de sentido único (one-Ways), a encriptação é o sentido fácil, sendo necessário somente à chave publica para realizar esta tarefa, e qualquer pessoa poderá realizá-la".

A decifração é a parte complexa da técnica de chave assimétrica, conforme SCHNEIER (1996, p. 31), "Uma pessoa com um computador Cray (super computador) poderia levar milhares de anos para conseguir decifrar a mensagem sem a chave privada, com a chave o processo é tão simples quanto a encriptação".

Figura 2.4.1.2: Funcionamento da criptografia assimétrica.

Fonte: SCHNEIER (1996, p. 31)

Ainda, segundo SCHNEIER (1996, p. 32), a técnica de criptografia de chave pública funciona basicamente da seguinte maneira:

- A deseja enviar uma mensagem de forma segura para B;
- A e B concordam com um sistema de criptografia de chave pública em comum;
- B fornece para a A sua chave pública, podendo enviá-la ou disponibilizá-la em qualquer meio de comunicação, não sendo necessário um meio seguro;
- A cifra a mensagem que ela deseja enviar com a chave pública do B e a envia para B;
- B ao receber a mensagem, a decifra utilizando sua chave privada e obtém a mensagem original enviada por A.

A Figura 2.4.1.2 exemplifica o funcionamento básico da técnica.

CAPÍTULO 3 – INSERÇÃO NO BIT MENOS SIGNIFICATIVO (LSB)

3.1 Introdução

Neste capítulo é apresentada uma visão geral sobre método LSB (Last Significant Bit), explicando seu funcionamento. O método LSB foi o escolhido para ser implementado no software para ocultar mensagens criptografadas em imagens.

Inserção no bit menos significativo - O método mais utilizado para armazenar informações em imagens digitais é o *LSB*. Esse método consiste em ocultar a informação no bit menos significativo de cada pixel, ou de cada cor, da imagem (JASCONE, 2003, P.28).

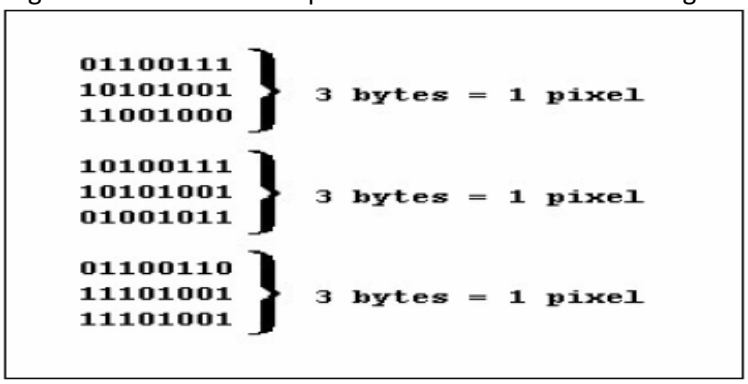
O algoritmo LSB utiliza uma *imagem pública*, para esconder uma determinada informação. Esta informação é escondida nos bits menos significativos da imagem, e tem como saída uma *imagem esteganografada* aparentemente semelhante à original, mas com a informação oculta. Através de um processo de *"desesteganografia"*, a informação oculta na imagem pode ser recuperada.

O método LSB é uma grande técnica de esteganografia de imagens digitais, mas infelizmente vulnerável aos ataques do tipo manipulação da imagem. Os métodos de *LSB* escondem eficientemente dados dos olhos humanos, entretanto são facilmente destruídas na utilização de algoritmos de compressão com perdas de dados. Se uma mensagem estiver escondida em uma imagem no formato *bitmap*, a mensagem será perdida se a imagem for convertida para o formato *JPEG*.

3.2 Funcionamento

A Figura abaixo ilustra os valores dos pixels de uma determinada imagem.

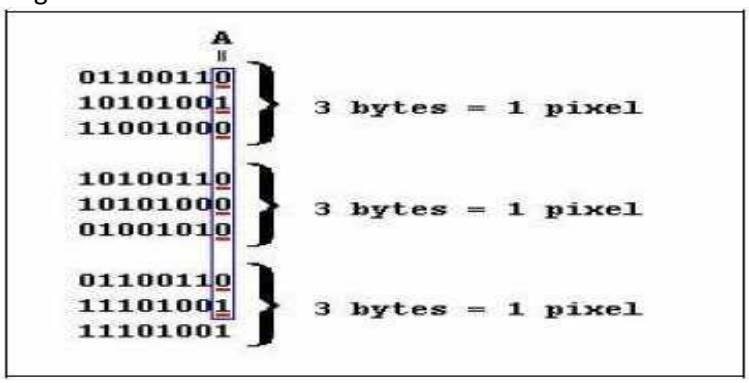
Figura 3.2.1: Valores dos pixels de uma determinada imagem.



Fonte: JASCONE (2003, p.29)

Supondo que o desejável seja armazenar a letra “A”. A letra “A” possui o código 65 da tabela ASCII com valor binário 0 1 0 0 0 0 0 1. Utlilizando a Método LSB a imagem resultante seria:

Figura 3.2.2: Letra “A” armazenada utilizando o método LSB.



Fonte: JASCONE (2003, p.40)

Onde os bits em destaque na imagem acima, são os que foram alterados. Vale lembrar que essa alteração na cor de cada pixel é imperceptível ao olho humano.

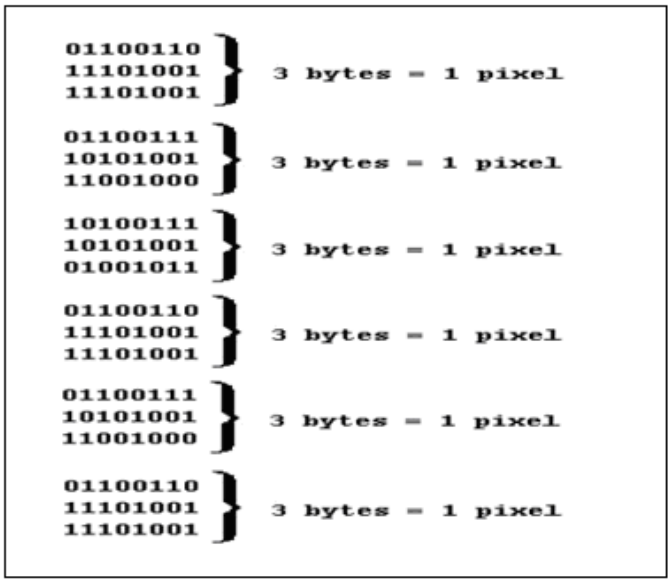
De acordo com JASCONE (2003, p.38), o tamanho máximo do arquivo a ser armazenado depende do tamanho da imagem. Para ocultar uma informação em uma imagem de 24 *bits*, onde cada *pixel* possui 3 *bytes*, utilizando o método LSB, podemos armazenar 3 bits em cada *pixel*, utilizando 1 *bit* de cada *byte* desse *pixel*, ou seja, para cada *byte* da informação a ser escondida, será necessário 8 *bytes* da imagem.

3.3 Customizações

Por ser a técnica mais utilizada, pode-se supor que a utilização da técnica LSB pode ser facilmente detectada, basta percorrer os bits menos significativos de uma imagem, por exemplo, e se os dados obtidos fizerem algum sentido a mensagem secreta foi descoberta. Para fugir do obvio algumas “customizações” podem ser implementadas a fim de se dificultar a obtenção da informação por parte de pessoas não autorizadas.

Uma delas é o uso da criptografia, que mesmo que uma pessoa obtenha a mensagem, a mesma não fará sentido algum, e para decifrá-la será necessário conhecer a chave criptográfica.

Figura 3.3.1: Valores dos pixels de uma determinada imagem.



Outra maneira de melhorar o LSB vem do fato de não ocultar a mensagem no início da imagem, por exemplo, a mensagem pode ser iniciada no meio, ou em outra posição qualquer da imagem. Novamente será utilizada a letra "A" nesta imagem, que é representada pelo código ASCII 65, em 0 1 0 0 0 0 0 1.Na figura 3.3.2 é ilustrada a técnica descrita acima.

Figura 3.3.2: Letra "A" armazenada utilizando o método LSB, começando em uma posição pré-determinada .

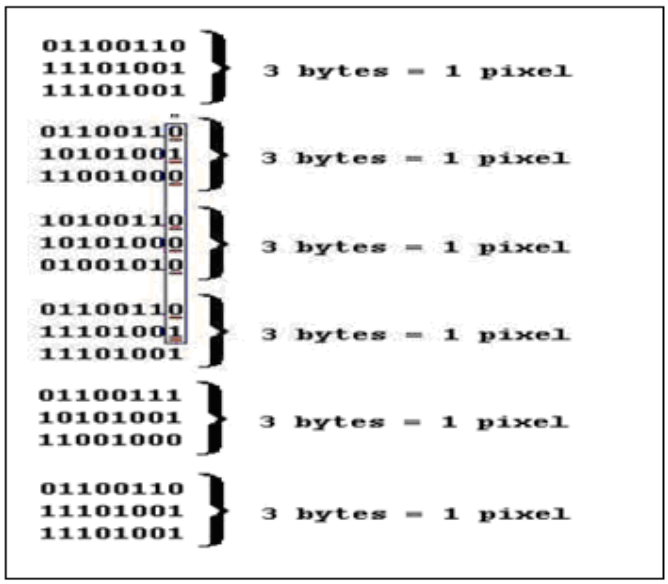
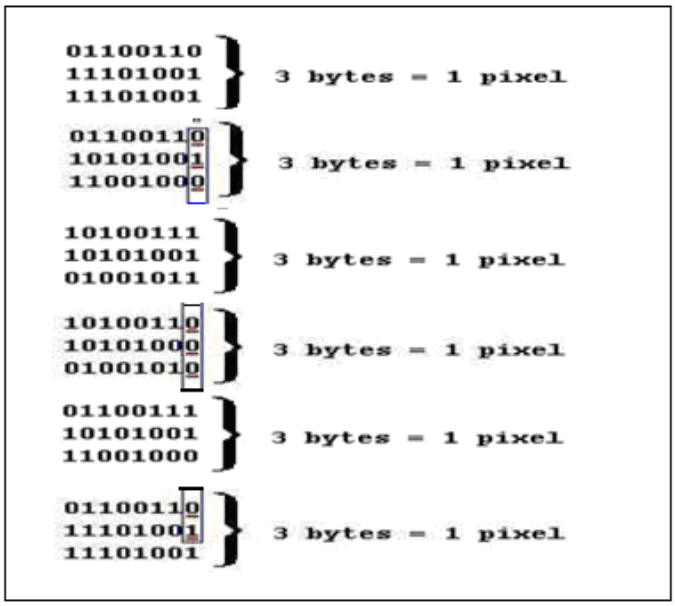


Figura 3.3.3: Letra “A” armazenada de forma alternada utilizando o método LSB.



Outro método que pode ser adotado é a utilização de números aleatórios, que seria responsável pela dispersão da mensagem na imagem. Sendo esses números gerados de forma pré-definida possibilitando sua recuperação no processo de extração, ou seja, os pixels alterados não estariam em bits adjacentes, mas sim espalhados de forma alternada, dificultando dessa forma a obtenção da mensagem original. Na figura 3.3.3 é ilustrada a técnica descrita acima.

As customizações apresentadas, são citadas apenas para conhecimento, as mesmas não foram utilizadas no desenvolvimento do software, ou seja no desenvolvimento a mensagem é armazenada em bits adjacentes e no inicio da imagem.

CAPÍTULO 4 – ALGORITMO DES

4.1 Introdução

Neste capítulo é apresentada uma visão geral sobre o algoritmo DES, explicando seu funcionamento teoricamente e na pratica através de um pequeno exemplo. Como explicado anteriormente o DES é um algoritmo simétrico, ou seja, a chave utilizada para cifrar os dados é a mesma utilizada para decifrar os dados. O algoritmo DES foi o escolhido para ser implementado no software para ocultar mensagens criptografadas em imagens.

Segundo EDWARD, PEREIRA e CHIARAMONTE (2005, p. 113), o algoritmo DES (Data Encryption Standard) é um dos algoritmos de cifragem mais utilizados no mundo. E apesar da Electronic Frontier Foundation ter conseguido criar uma maquina para quebrar mensagens cifradas com o algoritmo DES, ele continuara sendo útil para governos e bancos nos próximos anos, através da versão chamada triple-DES.

O processo de cifragem realizada pelo trile-DES, também conhecido como 3-DES, é idêntica á do algoritmo DES, só que é repetido três vezes.

4.2 Histórico

Em 15 de maio de 1973, durante o governo de Richard Nixon, o NBS (National Bureau of Standards) publicou uma notícia solicitando propostas de algoritmos criptográficos, que seriam utilizados para proteger dados durante transmissões e armazenamento. A notícia explicava por que a cifragem de dados era um assunto importante (EDWARD; PEREIRA; CHIARAMONTE, 2005, p. 113).

O NBS solicitava técnicas e algoritmos para cifragem de dados por computador e, também, técnicas para implementar as funções criptográficas de gerar, avaliar e proteger chaves criptográficas, manter arquivos cifrados com chaves que expiram, realizar atualizações parciais em arquivos cifrados e misturar dados cifrados com claros para permitir marcações, contagens, roteamentos, etc. O NBS, ao desempenhar o papel de estabelecer padrões e de auxiliar o governo e a industria no acesso à tecnologia, encarregar-se ia de avaliar os métodos de proteção para preparar as linhas de ação.

O NBS ficou esperando por resposta. Nenhuma apareceu até agosto de 1974, três dias antes da renúncia de Nixon, quando a IBM apresentou um algoritmo candidato que ela havia desenvolvido internamente, denominado Lucifer.

O algoritmo Lucifer passou por avaliações, avaliações realizadas com a ajuda da National Security Agency, após essa etapa a NBS adotou o algoritmo Lucifer com algumas modificações sob a denominação de DES em 15 de julho de 1977.

O DES foi rapidamente adotado pela mídia não digital, como as linhas telefônicas públicas. Depois de alguns anos, a Internacional Flavors and Fragrances começou a utilizar o DES para proteger as transmissões por telefone das suas fórmulas (EDWARD; PEREIRA; CHIARAMONTE, 2005, p. 114).

Neste meio-tempo, a indústria bancária, a maior usuária de criptografia depois do governo, adotou o DES como padrão para o mercado bancário atacadista

4.3 Funcionamento

De acordo com EDWARD, PEREIRA e CHIARAMONTE (2005, p. 114), o algoritmo DES é composto de operações simples, como: permutações, substituições, XOR e deslocamentos. O DES utiliza o processo de cifra de bloco para criptografar informações, os blocos utilizados possuem tamanho de 64 bits, retornando blocos de texto cifrado do mesmo tamanho, utilizando uma chave de 56 bits.

No algoritmo DES o processo principal é executado 16 vezes, e em cada iteração se utiliza uma subchave derivada da chave original.

Não serão exibidos detalhes do funcionamento, uma vez que para o desenvolvimento do software foi utilizada a API CRYPTO do Java que disponibiliza vários algoritmos de criptografia, e o DES é um deles.

4.4 Exemplo de Funcionamento

A seguir é apresentado um exemplo simples, ilustrando o funcionamento do algoritmo DES. O exemplo utilizado é citado por EDWARD, PEREIRA e CHIARAMONTE (2005, p. 121).

Já foi dito anteriormente que o algoritmo DES trabalha cifrando blocos de 64 bits d mensagem, isso significa 16 números hexadecimais, pois cada número hexadecimal representa 2^4=16, um grupo de 4 bits.

Para realizar o processo de cifrar dados, o DES utiliza chaves com cumprimento aparente de 16 números hexadecimais (64 bits). Porém, o algoritmo DES ignora cada oitavo bit da chave, deste modo a chave acaba tendo 56 bits de comprimento.

Se a mensagem original for 87878787878787h(apresentação hexadecimal) cifrada com a chave DES 0E328232EA6DOD73h, obtem-se o texto cifrado

0000000000000000h. Desta forma, decifrando o texto cifrado com a chave DES 0E328232EA6DOD73h, o resultado será o texto original 87878787878787878787h.

O exemplo citado é bem simples, uma vez que o texto original tinha o cumprimento de 64 bits. Porém a maioria das mensagens poderá não cair nessa categoria, pois nem sempre o comprimento da mensagem será múltiplo de 64 bits. Sendo necessário nesses casos, ajustar o ajuste do cumprimento da mensagem, adicionando bytes extras no final, para cifrar a mensagem. Bytes extras que são descartados após a mensagem ser decifrada. Existem vários métodos diferentes para adicionar bytes, um deles é o simples preenchimento com zeros.

CAPÍTULO 5 – DESENVOLVIMENTO DO SOFTWARE PARA OCULTAR MENSAGENS CRIPTOGRAFADAS EM IMAGENS

5.1 Introdução

Este capítulo apresenta o estudo de caso da modelagem do software com a UML (Linguagem de Modelagem Unificada). O software é modelado nos diagramas Casos de Uso, Classes e Componentes.

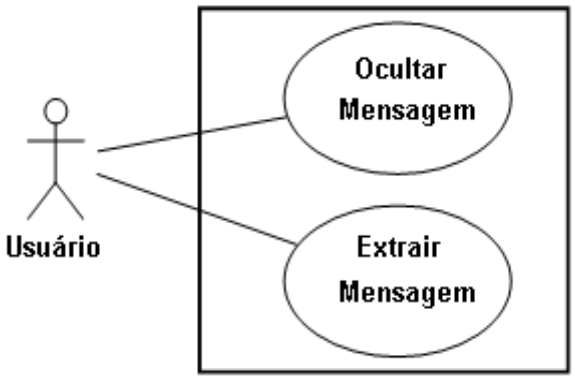
5.2 Modelagem do software com a UML

Antes de iniciar a modelagem do sistema são apresentados os requisitos do software para ocultar mensagens criptografadas em imagens:

- o usuário deve selecionar uma imagem para ocultar ou extrair a mensagem;
- o sistema deve permitir ao usuário informar a mensagem desejada;
- para ocultar ou extrair a mensagem o usuário deve informar a senha de criptografia utilizada;
- o sistema deve exibir uma mensagem apropriada quando falhar ou quando a mensagem exceder o tamanho máximo permitido;
- as imagens utilizadas devem estar no formato PNG;
- o sistema deve salvar a imagem com a mensagem embutida.

Ao analisar a descrição do problema acima, pode ser determinado que o usuário é quem faz a interação com o sistema enquanto as funções deste são ocultar e extrair mensagem. Portanto, pode-se definir o usuário como o Ator e Ocultar mensagem e Extrair mensagem como os Casos de Uso. O Diagrama de Casos de Uso é exibido na figura 5.2.1.

Figura 5.2.1: Diagrama de Caso de Uso do sistema.



Nesta figura 5.2.1 temos o Ator Usuário conectado aos Casos de Uso Ocultar mensagem e Extrair mensagem que estão no interior da fronteira do sistema. A tabela 5.2.1 descreve a documentação dos Casos de Uso Ocultar e Extrair mensagem.

Tabela 5.2.1: Documentação do Caso de Uso Ocultar Mensagem.

Nome do caso de uso	Ocultar Mensagem
Ator	Usuário
Resumo	Este caso de uso descreve as etapas percorridas pelo Usuário para ocultar uma mensagem em uma imagem.
Ações do Ator (Fluxo Normal)	**Respostas do Sistema (Fluxo Normal)**
1. Entrar com a mensagem que se deseja ocultar.	
2. Solicitar que a mensagem seja ocultada.	
3. Informar a Imagem	
4. Informar o nome da Imagem Modificada	
5. Informar a chave de criptografia.	
	6. Verificar formato da Imagem.
	7. Exibir a imagem com a mensagem embutida.
Restrições/Validações	
	1. A imagem deve estar no formato PNG

Tabela 5.2.2: Documentação do Caso de Uso Extrair Mensagem.

Nome do caso de uso	Extrair Mensagem
Ator	Usuário
Resumo	Este caso de uso descreve as etapas percorridas pelo Usuário para extrair uma mensagem de uma imagem.
Ações do Ator (Fluxo Normal)	**Respostas do Sistema (Fluxo Normal)**
1. Solicitar a extração da mensagem.	
2. Selecionar a imagem.	
3. Confirmar a extração da mensagem.	
4. Informar a chave de criptografia.	
5. Ler a mensagem.	
	6. Verificar a existência da mensagem.
	7. Exibir a mensagem extraída da imagem.
Restrições/Validações	
	1. A imagem deve estar no formato PNG.

O próximo passo é identificar as possíveis classes do sistema. Vamos separar os substantivos do Documento de Requisitos e analisá-los conforme a seguir:

esteganografia: pode-se determinar esta como uma classe do sistema e que nela sejam feitas às operações para ocultar e extrair as mensagens;

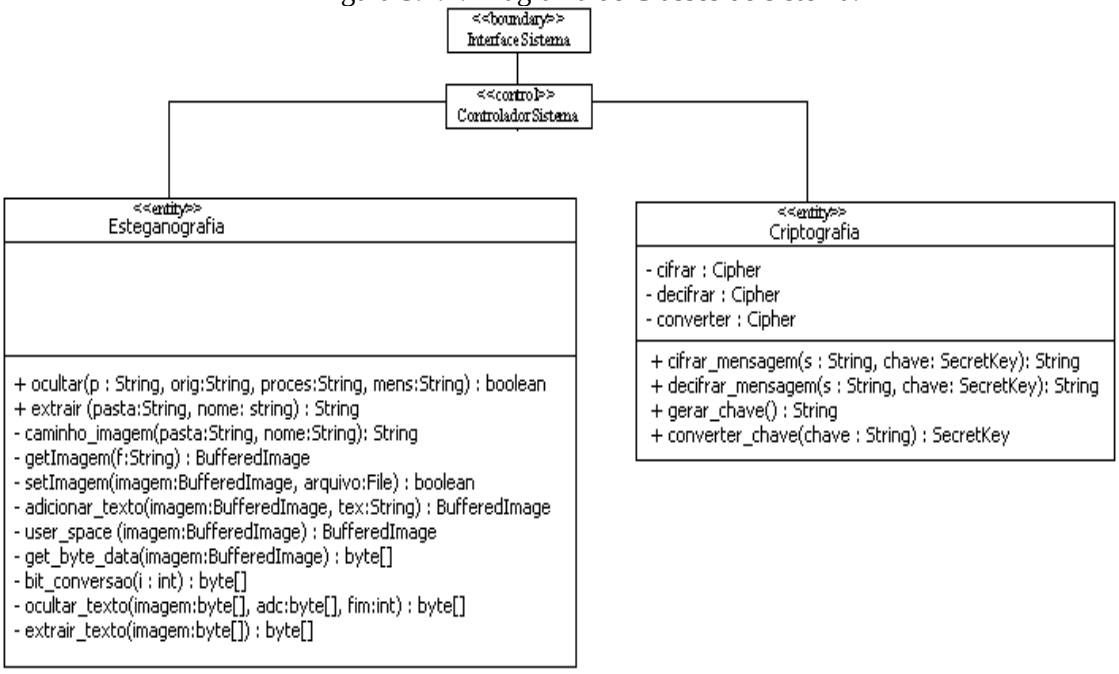
criptografia: pode-se determinar esta como uma classe do sistema e que nela sejam realizadas às operações para cifrar e decifrar a mensagem;

usuário: é quem interage com o sistema e, por não fazer parte deste, não é necessário modelá-lo como classe;

Criptografia e esteganografia foram definidas como classes do sistema. Além destas, serão criadas as classes InterfaceSistema e ControladorSistema com os estereótipos do tipo <<boundary>> e <<control>> respectivamente.

Uma possível solução para o nosso Diagrama de Classes é descrita na figura 5.2.2. Além dos atributos e métodos especificados acima, foram adicionados outros que foram necessários para a construção do Diagrama de Classes e, conseqüentemente para a implementação do sistema.

Figura 5.2.2: Diagrama de Classes do sistema.



InterfaceSistema: esta classe representa a comunicação entre o Usuário e o sistema. Por meio da InterfaceSistema o Usuário pode ocultar ou a extrair a mensagem.

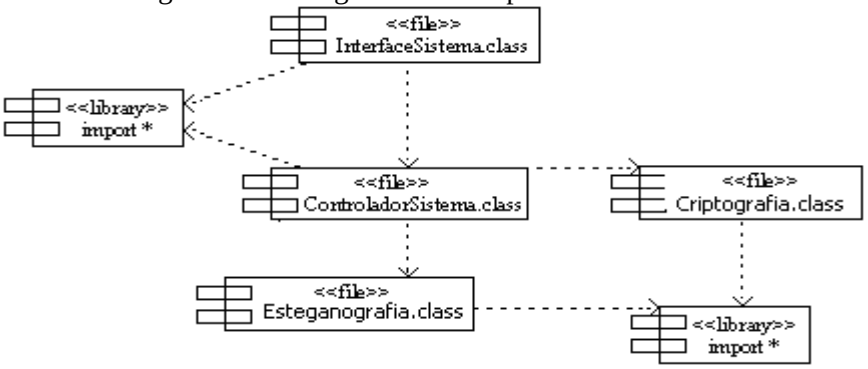
ControladorSistema: classe de controle responsável por traduzir a solicitação feita pelo Usuário e retransmiti-la às outras classes do sistema. Ela também é encarregada de informar o Usuário de possíveis erros de entrada ou falhas no sistema.

esteganografia: nesta classe são efetuados as operações de ocultar e extrair a mensagem da imagem.

Criptografia: esta classe é responsável pelas operações de cifrar mensagem, decifrar mensagem, gerar chave e converter chave.

Por último, modelamos o Diagrama de Componentes que é apresentado na figura 5.2.6.

Figura 5.2.6: Diagrama de Componentes do sistema.



No diagrama o componentes, da figura 5.2.6, o import * foi repetido para deixá-lo mais inteligível e não precisarmos interceptar linhas. Este componente tem o estereótipo <<library>> e são as bibliotecas usadas por todos os outros componentes. Estes outros possuem o estereótipo <<file>> e a extensão .class que contém a versão compilada dos componentes .java (código-fonte) (DEITEL; DEITEL, 2005, p.10). O ControladorSistema.class depende dos componentes Esteganografia.class e Criptografia.class para receber os resultados destes e retransmiti-los ao componente InterfaceSistema.class.

CAPÍTULO 6 – ESTUDO DE CASO – SOFTWARE EM FUNCIONAMENTO

6.1 Introdução

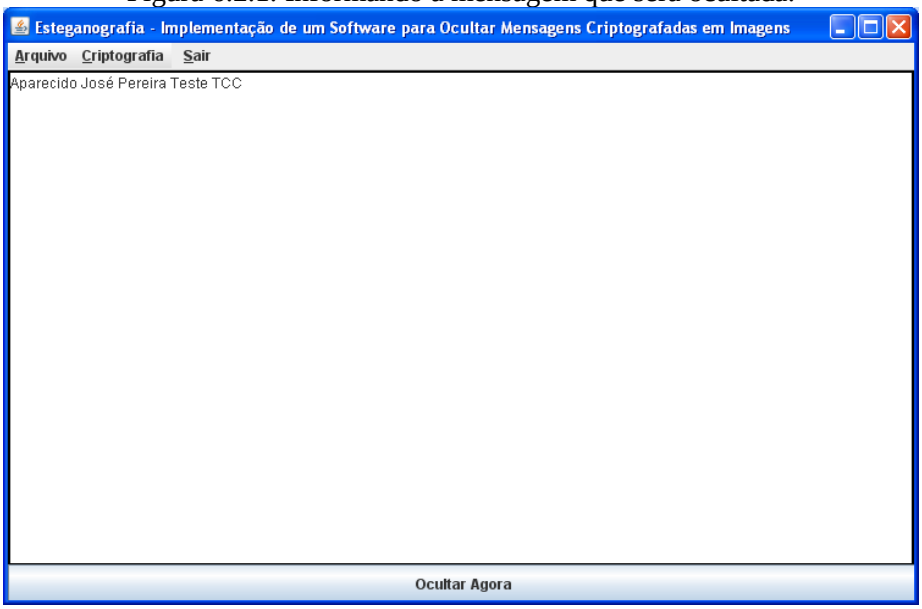
Neste capítulo é apresentado um estudo de caso mostrando detalhes do software em funcionamento, apresenta também uma comparação das imagens originais e processadas, a fim de observar diferenças no tamanho ou na aparência da imagem após a inserção da mensagem.

6.2 Utilizando o Software

O estudo de caso mostrando detalhes do software em funcionamento será realizado através de um exemplo de utilização do software, onde são realizadas as operações de ocultar mensagem e extrair mensagem.

Ao iniciar o software é apresentada ao usuário uma área de texto, para ocultar a mensagem deve-se informar nessa área à mensagem desejada, como é demonstrado na figura 6.2.1.

Figura 6.2.1: Informando a mensagem que será ocultada.



Em seguida deve-se clicar sobre o botão Ocultar Agora, que fina na parte inferior da janela principal, ao realizar essa operação o usuário devera selecionar a imagem onde a mensagem será ocultada, e deverá também informar o nome para salvar a imagem de saída, e em seguida devera informar a chave de criptográfica que será utilizada, como é ilustrado nas figuras 6.2.2, 6.2.3 e 6.2.4.

Figura 6.2.2: Selecionando a Imagem Original.

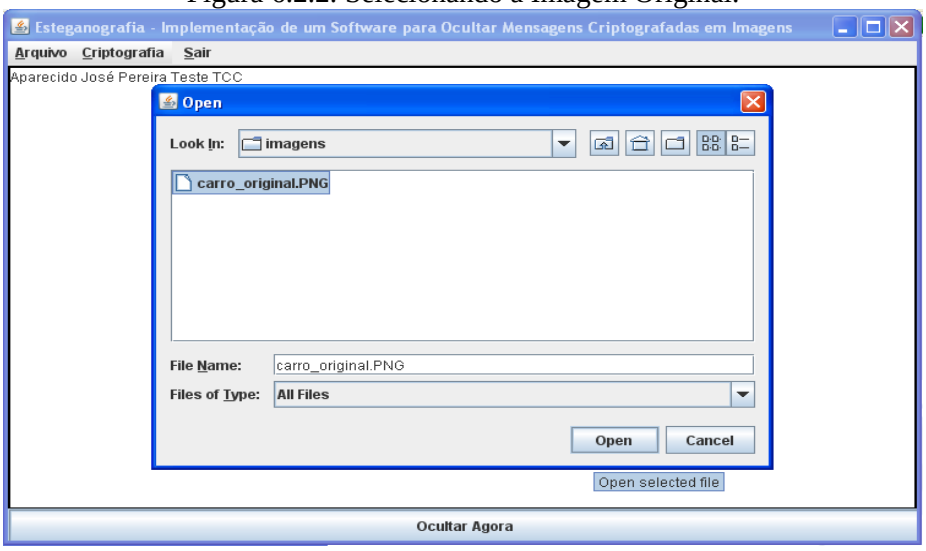
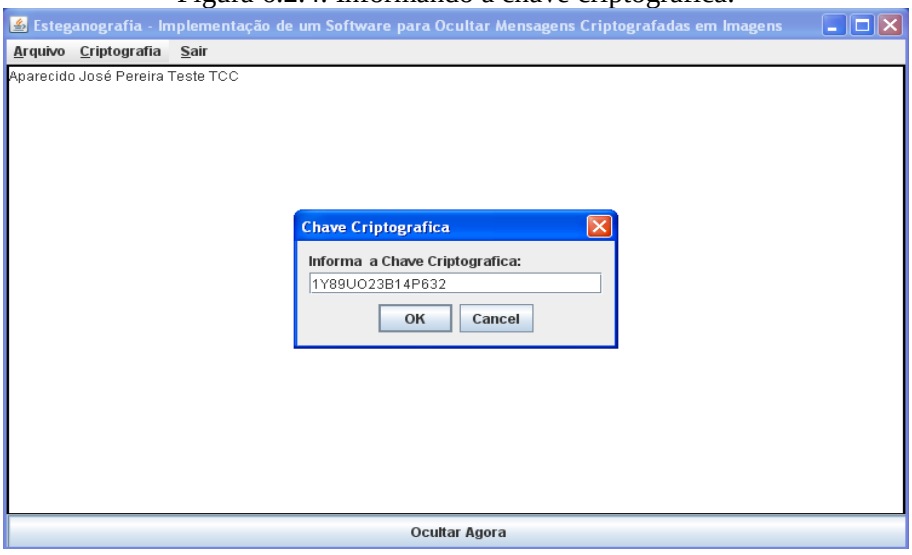


Figura 6.2.3: Informando o nome do arquivo de saída.

Figura 6.2.4: Informando a chave criptografica.

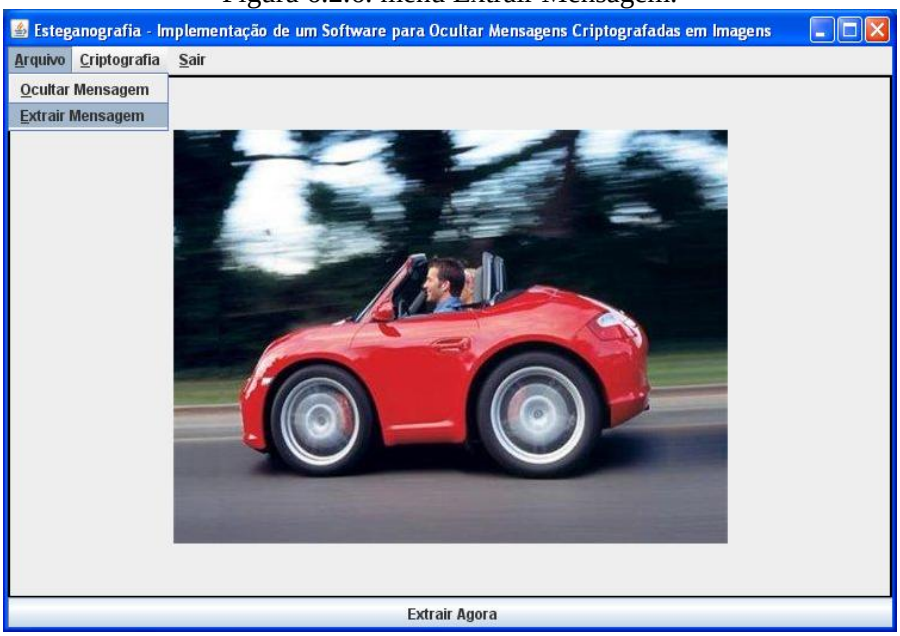


Se a mensagem foi ocultada com sucesso, o software apresentara uma mensagem indicando o sucesso da transação, e em seguida a imagem modificada será exibida na janela do sistema, conforme a figura 6.2.5.

Figura 6.2.5: Exibindo Imagem modificada.

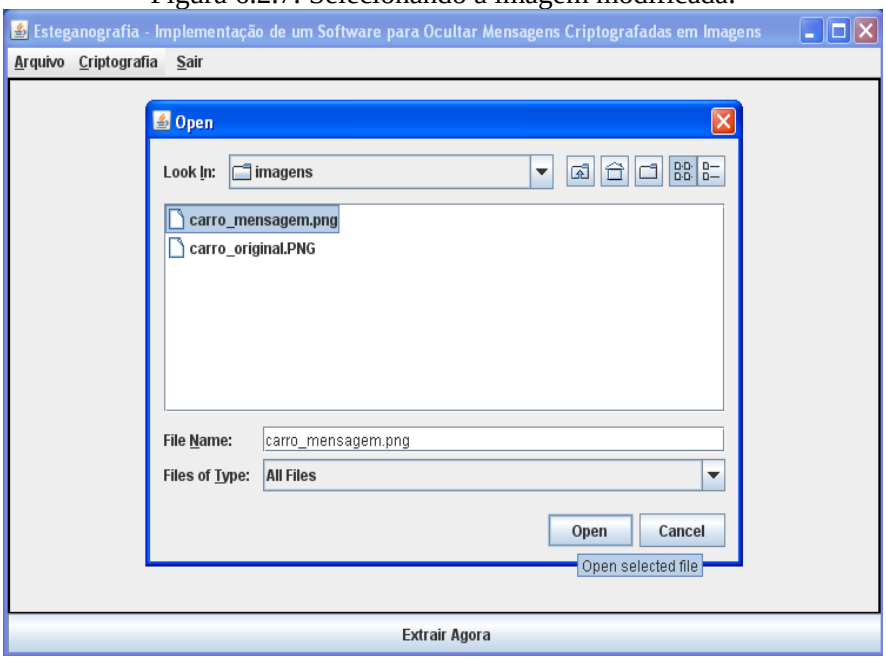
Para extrair a mensagem, basta clicar no menu Arquivo/Extrair Mensagem, que esta localizado na parte superior da janela, como é demonstrado na figura 6.2.6.

Figura 6.2.6: menu Extrair Mensagem.



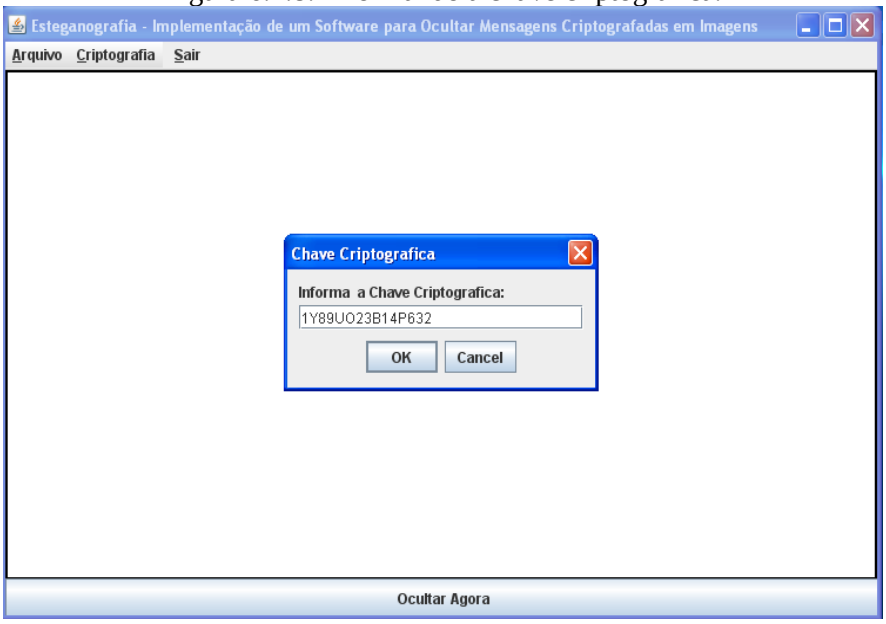
Ao escolher o menu Extrair Mensagem deverá ser indicada a imagem que contém a mensagem oculta. Essa operação é ilustrada na figura 6.2.7.

Figura 6.2.7: Selecionando a imagem modificada.



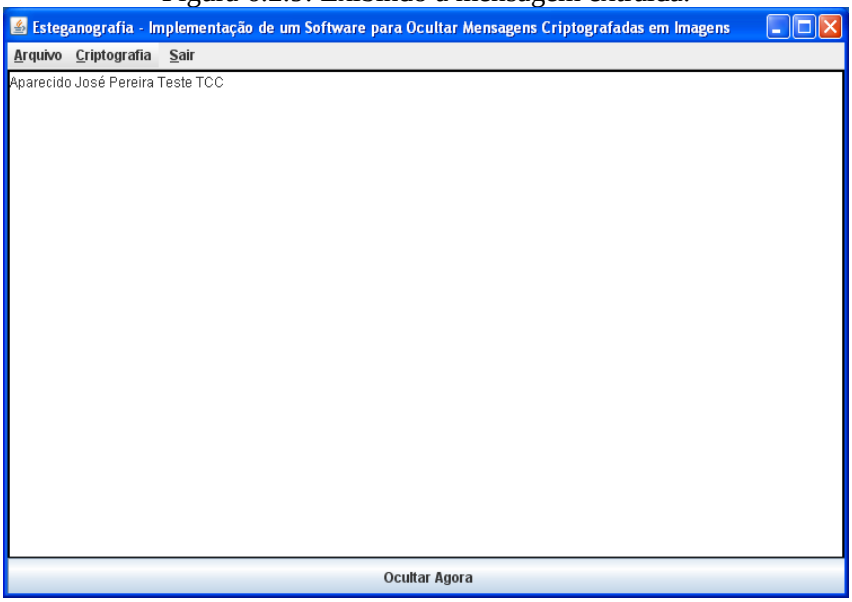
Após escolher a imagem modificada a mesma é exibida pelo software, para extrair a mensagem devemos clicar sobre o botão extrair mensagem que fica na parte inferior da Janela, neste momento é solicitada a chave criptográfica, que deve ser a mesma utilizada no processo de ocultar a mensagem, como é ilustrado na figura 6.2.8.

Figura 6.2.8: Informando a chave criptográfica.



Após informar a chave criptográfica a mensagem extraída e exibida ao usuário, conforme figura 6.2.9.

Figura 6.2.9: Exibindo a mensagem extraída.

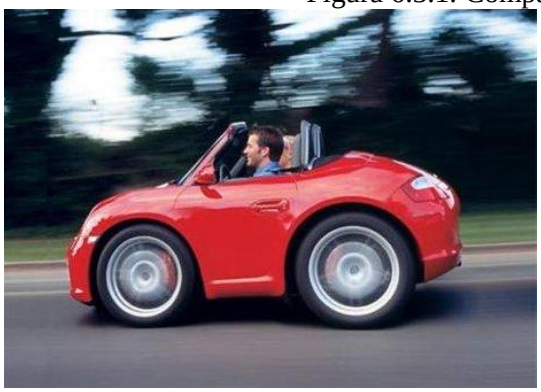


6.3 Comparando Resultados

Abaixo são realizadas algumas comparações entre as imagens originais e as que possuem algum texto oculto.

A figura 6.3.1 apresenta a imagem de um carro, a original possui o tamanho de 306KB, e após inserir uma mensagem na mesma e salva-la com o nome de carro_mensagem o tamanho da imagem permaneceu o mesmo 306KB e visivelmente não é possível detectar nenhuma diferença entre as imagens.

Figura 6.3.1: Comparação das imagens 1.



Tamanho: 306 KB

Tamanho: 306 KB

A figura 6.3.2 apresenta uma nova comparação entre imagens, agora utilizamos a figura de uma paisagem.

Figura 6.3.2: Comparação das imagens 2.

Tamanho: 892 KB

Tamanho: 897 KB

A imagem original possui o tamanho de 892KB, e após inserir uma mensagem na mesma e salva-la com o nome de paisagem_mensagem houve uma alteração pequena no tamanho da imagem que passou a ter 897KB, visivelmente não foi possível detectar nenhuma diferença entre as imagens.

A figura 6.3.3 apresenta uma nova comparação entre imagens, agora utilizamos uma imagem que possui a cor preta.

Figura 6.3.3: Comparação das imagens 3.

Tamanho: 5,13 KB

Tamanho: 6,24 KB

Conforme ilustrado na figura 6.3.3, a imagem original possui o tamanho de 5,13KB, e após inserir uma mensagem na mesma e salva-la, houve uma alteração pequena no tamanho da imagem que passou a ter 6,24 KB, visivelmente não foi possível detectar nenhuma diferença entre as imagens.

Nas comparações realizadas pode ser observado que não é possível identificar nenhuma alteração na imagem, ou seja, o sistema visual humano não pode detectar nenhuma diferença entre as imagens, porém pode haver diferenças no tamanho das imagens, diferença essa que se deve ao fato da compressão PNG levar em conta o contraste de cores da imagem na hora da compressão, uma vez que o método LSB altera o valor dos pixels da imagem, diferença que não é percebida pelo olho humano, porém pode causar alterações no tamanho na imagem.

CONCLUSÕES

No desenvolvimento deste trabalho destacam-se o conhecimento teórico e prático obtido com a modelagem e implementação do software para ocultar mensagens criptografadas em imagens.

Neste trabalho foram apresentados conceitos sobre esteganografia, descrevendo o que é, e relatando sua origem e sua importância, foram apresentados conceitos sobre criptografia, foi apresentado o que é a criptografia relatando sua origem e importância, foram apresentados detalhes sobre a Inserção no Bit menos significativo (LSB), tendo como objetivo detalhar o método de esteganografia adotado no desenvolvimento do software para ocultar mensagens criptografadas em imagens, também foram apresentados detalhes sobre o algoritmo DES que foi o algoritmo de criptografia utilizado, foi apresentada a modelagem do software em UML e foi apresentado um estudo de caso, com o intuito de demonstrar o software e seu funcionamento.

Durante o projeto pode ser concluído que a criptografia e a esteganografia , quando bem utilizadas, fornecem meios eficientes e eficazes na busca por proteção digital. Embora a esteganografia e criptografia sejam diferentes, elas possuem um objetivo em comum que é a proteção da informação, podendo ambas ser utilizadas em conjunto, aumentando dessa forma o grau de segurança nas transferências de informações.

Pode ser concluído também que as imagens processadas pelo software desenvolvido podem sofrer pequenas alterações em seu tamanho, porém não foi possível detectar diferenças visuais entre as imagens.

Num trabalho futuro, podem ser estudadas outras técnicas de esteganografia, utilizando outros formatos de arquivos como música e textos, também é possível implementar outras técnicas de criptografia, possibilitando dessa forma que o usuário escolha o método de criptografia que deseja utilizar.

Ainda em um trabalho futuro podem ser estudadas técnicas de esteganálise, que tem por objetivo detectar a presença da esteganografia, adequando o software para que não seja detectada a presença da mensagem, aumentando a segurança do mesmo.

REFERÊNCIAS

BOOCH, Grady; JACOBSON, Ivar; RUMBAUGH, James. **UML:** guia do usuário. Rio de Janeiro: Elsevier, 2005.

BURNETT, S.; PAINE, S. (2002) **"Criptografia e Segurança:** O Guia Oficial RSA". Rio de Janeiro: Campus, 2002.

COELHO, Laura C. M.; BENTO, Ricardo J**. Ferramentas de Esteganografia e seu uso na Infowar**. *Evidência Digital Magazine*, Rio de Janeiro, Edição-3, Ano 1.

DEITEL, Harvey. M.; DEITEL, Paul. J. **Java:** como programar. 6. ed. São Paulo: Pearson Prentice Hall, 2005.

EDWARD , David M.; PEREIRA, Fabio D.; CHIARAMONTE,Rodolfo B. . **"Criptografia Em Software e Hardware"**, 2005.

FOWLER, Martin. **UML essencial:** um breve guia para a linguagem-padrão de modelagem de objetos. 3. ed. Porto Alegre: Bookman, 2005.

GUEDES, Gillianes T. A. **UML:** uma abordagem prática. 2. ed. São Paulo: Novatec Editora, 2006.

HINZ, Marco Antônio Mielke. **Um estudo descritivo de novos algoritmos de criptografia**, Rio Grande do Sul, dez. 2000.

JASCONE, Fabio Luis Tavares. **Protótipo de Software para ocultar texto em imagens digitais**, Blumenau, nov. 2003.

LARMAN, Craig. **Utilizando UML e padrões:** uma introdução à análise e ao projeto orientados a objetos. Porto Alegre: Bookman, 2002.

PETRI, Marcelo. **Esteganografia**, Joinville, dez. 2004.

POPA, R. (1998). **An analysis of steganography techniques**. Tese de Mestrado, Departamento de Ciência da Computação e Engenharia de Software da "Politécnica", Universidade da Timisoara, Romênia.

ROCHA, Anderson Rezende. **Desenvolvimento de um software para segurança digital utilizando esteganografia**, Lavras, jul. 2003.

SCHNEIER, Bruce **. Applied Cryptography**,1997.