FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA – UNIVEM
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

RODRIGO MARTINS DA CONCEIÇÃO

ESTEGOCHIP: ESTEGANOGRAFIA EM HARDWARE UTILIZANDO VHDL E CIRCUITOS PROGRAMÁVEIS (FPGA)

MARÍLIA
2010

RODRIGO MARTINS DA CONCEIÇÃO

EstegoChip:
Esteganografia em Hardware
Utilizando VHDL e Circuitos Programáveis (FPGA)

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 Bacharel em Ciência da Computação.

Orientador:
Prof. Dr. FÁBIO DACÊNCIO PEREIRA

MARÍLIA
2010

CONCEIÇÃO, Rodrigo Martins

EstegoChip: Esteganografia em Hardware Utilizando VHDL e Circuitos Programáveis (FPGA) / Rodrigo Martins da Conceição; orientador: Dr. Fábio Dacêncio Pereira. Marília, SP: [s.n.], 2010.

60 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, Marília, 2010.

1. Esteganografia 2. Imagem Digital 3. Segurança Digital 4. FPGA

CDD: 004.0684

CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO – AVALIAÇÃO FINAL

Rodrigo Martins da Conceição

ESTEGOCHIP: ESTEGANOGRAFIA EM HARDWARE UTILIZANDO VHDL E CIRCUITOS PROGRAMÁVEIS (FPGA)

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: 10 (Dez)

Orientador: Fábio Dacêncio Pereira

1°. Examinador: Juliana de Oliveira

2°. Examinador: Rodolfo Barros Chiaramonte

Marília, 03 de dezembro de 2010.

DEDICATÓRIA

Dedico a conclusão de mais esta etapa:

A toda minha família e amigos pelo eterno apoio e incentivo.

Aos meus pais por todo amor e dedicação a mim dedicados, fornecendo as bases para minha formação.

À Daniela, minha esposa e companheira, que compreende a minha dedicação para com os estudos com muito carinho e paciência.

À minha mãe, Luzinete, por estar sempre ao meu lado valorizando o meu esforço e trabalho.

E à pequena Beatriz, minha bebê, por ter entrado na minha vida, trazendo-me alegria e vontade de melhorar a cada dia.
E também por ter me ajudado a concluir o curso, quando, sentada em meu colo, estudava comigo!

AGRADECIMENTOS

Meus sinceros agradecimentos:

Aos amigos da faculdade, companheiros de estudo e divertimento. Sempre estavam presentes para tirar aquela dúvida de última hora!

A todos os professores do curso que lecionaram com vontade e dedicação, contribuindo direta ou indiretamente para minha formação acadêmica.

Ao professor Botega que, de forma humilde e atenciosa, sempre esteve à disposição para ajudar no que fosse preciso. Nos momentos de dúvidas, ele sempre me incentivou a seguir em frente.

E especialmente ao meu professor e orientador Dacêncio por compartilhar suas experiências e conhecimentos. Com muita competência e sabedoria, ele soube indicar quais os melhores caminhos a seguir, ponderando as dificuldades e benefícios de cada um.

Estes dois professores, além de educadores, foram amigos. Profissionais que admiro demais e tive o maior prazer em té-los conhecido.

"O problema da segurança de dados existe desde o momento em que alguém possua determinada informação e queira protegê-la".

José Ricardo Campelo Arruda

CONCEIÇÃO, Rodrigo Martins da. **EstegoChip: Esteganografia em Hardware Utilizando VHDL e Circuitos Programáveis (FPGA)**. 2010. 60 f. Trabalho de Curso (Bacharelado em Ciência da Computação) – Centro Universitário Eurípides de Marília, Fundação de Ensino "Eurípides Soares da Rocha", Marília, 2010.

RESUMO

Esteganografia significa escrita encoberta, ou seja, é a arte de esconder uma informação dentro de outra. Ao contrário do que possa parecer, essa é uma técnica antiga que foi bastante utilizada para fins militares, servindo de estratagema para a troca de mensagens secretas. Atualmente essa técnica vem ganhando espaço no campo digital, adaptando-se às novas necessidades e se utilizando das novas tecnologias disponíveis. Devido ao desenvolvimento tecnológico das últimas décadas e o uso dos sistemas compartilhados e das redes de computadores, está se tornando cada vez mais difícil proteger informações importantes das mais variadas formas de ameaças. Nesse contexto, a segurança digital desponta como uma área de grande potencial para pesquisa e desenvolvimento, atuando na criação de ferramentas e técnicas que provêem a proteção da informação e das comunicações. Sendo que a essência da segurança de tráfico é esconder informação, a esteganografia se apresenta como um mecanismo de proteção adequado para os dias atuais. A técnica de esteganografia utilizada foi o *Least Significant Bit* (LSB), que consiste em inserir a informação nos bits menos significativos de cada byte dos pixels da imagem. A descrição do hardware foi feita com a linguagem VHDL, utilizando a tecnologia de circuitos programáveis (FPGA).

Palavras-chave: Esteganografia. Imagem Digital. Segurança Digital. FPGA.

CONCEIÇÃO, Rodrigo Martins da. **EstegoChip: Esteganografia em Hardware Utilizando VHDL e Circuitos Programáveis (FPGA)**. 2010. 60 f. Trabalho de Curso (Bacharelado em Ciência da Computação) – Centro Universitário Eurípides de Marília, Fundação de Ensino "Eurípides Soares da Rocha", Marília, 2010.

ABSTRACT

Steganography means writing shelter, in other words, it is the art of hiding an information inside other. On the contrary what it apparenthy, that is an old technique that was quite used for military ends, serving as stratagem for the exchange secret messages. Now that technique is conquesting space in the digital field, adapting to the new needs and if utilizing of new available technologies. Due to the technological development of the last decades and the use of shared systems and of the nets of computers, it is turning if more and more difficult to protect important information in the most varied ways of threats. In that context, the digital safety blunts as an area of great potential for research and development, acting in the creation of tools and techniques that provide the protection of the information and of the communications. And the safety's of traffic essence is to hide information, the steganography it comes as an appropriate protection mechanism for the current days. The technique of used esteganografia was Least Significant Bit (LSB), that consists of inserting the information in the less significant bits of each byte of the pixels of the image. The description of the hardware was made with the language VHDL, using the technology of programmable circuits (FPGA).

Keywords: Steganography. Digital Image. Digital Security. FPGA.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AES: *Advanced Encryption Standard* (Padrão de Criptografia Avançado)

BMP: *Bitmap*

CLB: *Configurable Logic Block* (Bloco Lógico Configurável)

DES: *Data Encryption Standard* (Padrão de Criptografia de Dados)

FPGA: *Field Programmable Gate Array* (Dispositivo Lógico Programável)

HDL: *Hardware Description Languages* (Linguagem de Descrição de Hardware)

IOB: *In/Out Block* (Bloco de Entrada e Saída)

LSB: *Least Significant Bit* (Bit Menos Significativo)

PIXEL: *Picture Element* (Menor Elemento da Imagem)

RGB: *Red, Gren, Blue* (Vermelho, Verde, Azul)

RSA: Algoritmo desenvolvido por Ron Rivest, Adi Shamir e Len Adleman

SB: *Swith Box* (Caixa de Conexão)

VHDL: *Very High Speed Integrated Circuit* (Circuito Integrado de Alta Velocidade) + HDL (Linguagem de Descrição de Hardware com Ênfase em Circuitos Integrados de Alta Velocidade)

SUMÁRIO

INTRODUÇÃO

Nas últimas décadas houve uma verdadeira popularização do uso de computadores. Os ambientes informatizados passaram a fazer parte do cotidiano das pessoas e das empresas, e a informação digital se tornou de vital importância, ocupando o lugar da informação antes armazenada em papel (FERREIRA, 2003, p. 01-02).

Dessa forma, é possível identificar algumas mudanças importantes no que tange aos requisitos de segurança da informação. Duas dessas mudanças são apontadas por Stallings (2008, p. 03):

1) Antes da generalização dos computadores a segurança da informação de uma organização era fornecida por meios físicos (grandes armários com fechaduras de segredo) e administrativos (seleção de pessoal na contratação). Com a introdução do microcomputador no cotidiano das pessoas, torna-se essencial o uso de ferramentas automatizadas para proteger os arquivos digitais. A necessidade cresce com os sistemas compartilhados (*time-sharing*) e aumentam ainda mais com o uso das redes de telefonia, de dados ou Internet;

2) Outra mudança acontece com o desenvolvimento dos sistemas distribuídos e a expansão das redes e recursos de comunicação para transmitir dados entre o usuário do terminal e o computador e entre diferentes computadores. A segurança de redes torna-se necessária para proteger os dados durante sua transmissão através da rede ou, ainda, através de diversas redes interconectadas.

Dessa forma, Stallings (2008) coloca de forma bastante clara uma transição da **segurança da informação** para a **segurança de computador** e desta última para a **segurança de rede** (ou segurança de inter-rede).

No mesmo sentido, Terada (2008, p. 13) argumenta que devido ao extraordinário avanço na disseminação e popularização da rede mundial de computadores na década de 1.990 foi possível o advento e o rápido crescimento do comércio eletrônico (lojas virtuais) e das transações bancárias (*home-bankings*) feitas através da Internet. E acrescenta, ainda, que essa situação gerou uma necessidade muito grande em cada cidadão de uma proteção cada vez maior da privacidade, pois dados pessoais trafegam pela rede estando vulneráveis a diversos tipos de ataques de pessoas mal intencionadas.

Diversos programas e métodos de intrusão estão disponíveis em livros, revistas e Internet adicionando recursos e conhecimento a qualquer pessoa que possa se interessar pelo assunto. Apesar dos inúmeros e incontestáveis benefícios, os sistemas informatizados têm

seus pontos fracos, sendo vulneráveis a diversos tipos de ameaças, como ataques, invasões e vírus. Nesse contexto, desponta um grande problema a respeito da segurança da informação, pois diante de todas essas facilidades, aumenta também a necessidade de sistemas eficientes de proteção de dados, de ferramentas e de tecnologias novas que sejam capazes de proteger informações importantes, mantendo um nível adequado de segurança para o usuário. Sendo que a essência da segurança de tráfego é esconder informação (PETITCOLAS; ANDERSON; KUHN, 1999, p. 1062), a esteganografia se apresenta como um dos mecanismos de proteção pertinente aos dias atuais.

A esteganografia é a arte de ocultar uma mensagem de forma que ninguém a encontre, ou melhor, de forma que ninguém saiba de sua existência (JOHNSON; JAJODIA, 1998, p.26).

Ao longo da história foram desenvolvidos vários métodos para manter certas comunicações secretas, como tintas “invisíveis”, micro-pontos ou trocas de letras de uma mensagem (ROCHA; COSTA; CHAVES, 2003, p. 02; JOHNSON; JAJODIA, 1998, p. 26; PETITCOLAS; ANDERSON; KUHN, 1999, p. 1062). Este trabalho trata da esteganografia digital, mais especificamente da esteganografia em imagens. Para isso, faz uso de um algoritmo bastante conhecido e utilizado, denominado LSB (*Least Significant Bit* – Bit Menos Significativo) *insertion*. De acordo com essa técnica, utiliza-se sempre o bit menos significativo de cada byte de determinados pixels da imagem para armazenar a informação desejada. Todos os módulos do projeto foram descritos em VHDL e implementados com a tecnologia de circuitos programáveis (FPGAs).

Problemática e Justificativa

Com a popularização do uso de computadores – tanto no ambiente doméstico como no empresarial – principalmente a partir da década de 1990, houve uma mudança no tratamento da informação. Se até esse momento a informação era gravada em papel e podia ser protegida por um cofre ou um simples armário trancado, a partir daí cresce, cada vez mais, o processo de informatização, transformando a informação de papel em informação digital, exigindo uma estrutura de segurança muito mais sofisticada e complexa (FERREIRA, 2003, p. 01-02).

Além disso, nesse período começa a se disseminar o uso da rede mundial de computadores, a Internet. Por um lado, essa tecnologia traz muito conforto para as pessoas,

pois oferece muitos serviços como *home bankings* e lojas virtuais, além de muita informação das mais diversas áreas. Mas por outro lado, quando usada de maneira inadequada, toda essa comodidade pode servir como porta de entrada para grandes problemas, facilitando o trabalho de pessoas mal-intencionadas a fim de capturar informações sigilosas, como uma senha bancária, por exemplo (TERADA, 2008, p. 13).

Para resolver esse problema muito pode ser feito, incluindo políticas de segurança ou algoritmos criptográficos. A proposta deste trabalho é implementar uma maneira de ocultar informações de modo que elas passem despercebidas por essas pessoas mal-intencionadas. Isto é possível utilizando uma técnica de esteganografia, cujos detalhes serão descritos no decorrer do trabalho.

Objetivos

Este trabalho pode ser dividido em duas partes principais, sendo uma teórica e uma prática.

O objetivo da primeira parte é apresentar ao leitor alguns conceitos básicos sobre os temas abordados no projeto como um todo. Dessa forma, foram expostos alguns conceitos a respeito de criptografia, esteganografia, imagens, circuitos programáveis (FPGA) e linguagem VHDL. Também para fins de contextualização foi realizado um levantamento de alguns trabalhos e aplicativos já desenvolvidos referentes à esteganografia. É importante salientar que o cerne deste trabalho é a esteganografia e por esta razão este tema obteve maior destaque, sendo discorrido sobre sua história, seus tipos, seus algoritmos (ou técnicas de implementação) e suas aplicações.

No que se refere ao desenvolvimento, o objetivo principal deste trabalho é criar um *core* de segurança, buscando um diferencial com relação a outros trabalhos da área de segurança digital, pois este pretende seguir uma abordagem voltada para implementação em hardware. Para isso foi proposto a implementação de um chip, denominado EstegoChip, capaz de aplicar uma técnica de esteganografia em uma imagem digital. O hardware foi descrito através da linguagem VHDL e posteriormente sintetizado utilizando a tecnologia de circuitos programáveis (FPGA), sendo implementada a técnica LSB (*Least Significant Bit*) *insertion* para a inserção de informação nos bits menos significativos.

Metodologia

Na primeira fase deste projeto foi realizada uma pesquisa bibliográfica a respeito de esteganografia, bem como todos os outros temas que envolvem a área de segurança digital e que também foram abordados neste trabalho. O material selecionado nessa fase e utilizado posteriormente para a escrita da parte teórica (e alguns também utilizados na implementação) contempla alguns livros, dissertações e artigos de estudiosos da área de segurança digital.

No intuito de estudar a linguagem VHDL e a ferramenta de desenvolvimento ISE 9.1i da fabricante *Xilinx* e, posteriormente, para a descrição dos módulos do EstegoChip, foram utilizados livros, tutoriais e apostilas de mini-cursos.

É importante ressaltar que a pesquisa e leitura de novos materiais ocorreram do início ao fim do trabalho, acontecendo sempre de forma simultânea e auxiliar às outras etapas.

Após a leitura de uma parte do material selecionado, iniciaram-se a escrita da parte teórica do trabalho e a descrição dos módulos do EstegoChip, sendo que concomitantemente a essas etapas, foram realizados testes e simulações para validação do *core* de segurança.

Na fase de desenvolvimento, os módulos do projeto foram descritos na seguinte ordem: memória ROM, memória RAM, módulo de inserção de dados, módulo de recuperação dos dados, unidade de controle e módulo top. A cada módulo finalizado eram realizados testes através de simulações com a ferramenta de desenvolvimento. Os resultados eram analisados para verificar se o módulo estava funcionando corretamente podendo ser, então, validado.

Mais detalhes a respeito dos módulos, testes e análise de resultados serão abordados no item “4.5 Arquitetura do EstegoChip” e no “CAPÍTULO V – ANÁLISE DE RESULTADOS”.

Organização do Trabalho de Conclusão

Este trabalho foi organizado da seguinte maneira:

O capítulo I aborda o conceito de criptografia, uma caracterização dos sistemas criptográficos, os principais métodos de criptografia desenvolvidos ao longo do tempo e uma explanação sucinta sobre funções de *hash*.

O capítulo II apresenta a esteganografia (tema central deste trabalho), sua história, áreas de aplicação e técnicas de implementação.

No capítulo III é realizada uma explanação a respeito das imagens digitais, suas formas de representação e seus formatos; também é discorrido sobre os fundamentos das cores e seus sistemas.

O capítulo IV apresenta o EstegoChip, seu algoritmo esteganográfico, a linguagem e tecnologia utilizadas, os trabalhos correlatos existentes e a arquitetura criada.

No capítulo V são mostrados o teste de validação do EstegoChip e alguns de seus resultados como as taxas de ocupação e propagação, utilizadas como métricas para a análise de ocupação e desempenho, respectivamente.

Por fim, é apresentada a conclusão do trabalho e uma proposta para um trabalho futuro, agregando funcionalidades e mais segurança ao EstegoChip.

CAPÍTULO I – CRIPTOGRAFIA

Este capítulo descreve os principais métodos criptográficos e a função de *hash*. O estudo da criptografia, associado ao uso de funções *hash*, é de fundamental importância, visto que ela é um dos principais mecanismos de segurança de dados, sendo talvez o mais pesquisado e utilizado.

1.1 Conceitos

Em se tratando de criptografia alguns conceitos são fundamentais. Uma mensagem original (antes de ser codificada) é chamada texto claro (ou *plaintext*), enquanto que a mensagem após a codificação é conhecida como texto cifrado (ou *ciphertext*). O processo de transformar o texto claro em texto cifrado é denominado cifragem, sendo o processo inverso conhecido como decifragem. Os esquemas utilizados para cifrar uma mensagem constituem a área de estudo conhecida como criptografia (STALLINGS, 2008, p. 18). Trata-se, portanto, de um conjunto de conceitos e técnicas visando codificar uma informação para que somente o emissor e o receptor da mensagem consigam extrair seu significado.

1.2 Sistemas Criptográficos

A criptografia é baseada em transformações, sendo que cada transformação gera um sistema de criptografia, denominado criptossistema.

Stallings (2008, p. 19-20) caracteriza os sistemas criptográficos em três dimensões:

1. Tipo de operações utilizadas na transformação do texto claro em texto cifrado: as transformações podem ser através de substituição (troca das letras) ou transposição (inversão das posições das letras – anagrama);
2. Número de chaves usadas: quando uma única chave é usada para cifrar e decifrar uma mensagem, o sistema é considerado de criptografia simétrica; quando os processos de cifragem e decifragem necessitam de chaves distintas para serem concluídos com sucesso, o sistema é considerado de criptografia assimétrica;
3. Modo de processamento do texto claro: a cifra de bloco processa um bloco de elementos do texto de entrada de cada vez, gerando um bloco de elementos do

mesmo tamanho. A cifra em fluxo processa e gera como saída um elemento (um bit ou byte) por vez.

Além disso, um algoritmo seguro precisa atender a um ou a ambos os critérios a seguir:

- Custo para quebrar a cifra superior ao valor da informação codificada;
- Tempo exigido para quebrar a cifra superior ao tempo de vida útil da informação.

De acordo com suas características, portanto, ao longo do tempo foram surgindo diferentes métodos ou modelos criptográficos. Os principais são descritos a seguir.

1.2.1 Criptografia Aberta

Os primeiros métodos criptográficos existentes usavam apenas um algoritmo de codificação. Assim, bastava que o receptor da informação conhecesse esse algoritmo para poder extraí-la. No entanto, se um intruso tivesse posse desse algoritmo, também poderia efetuar um processo de decifragem, caso capturasse os dados criptografados.

Outro problema é o descrito a seguir: uma pessoa A manda uma mensagem criptografada para a pessoa B e outra mensagem diferente para a pessoa C. A pessoa B não deve ter conhecimento da mensagem enviada para a pessoa C e vice-versa. Porém, as duas mensagens podem ser decifradas usando o mesmo algoritmo, que tanto a pessoa B quanto a pessoa C possuem. Dessa forma, basta ter posse da mensagem alheia para conseguir decifrar tal mensagem.

1.2.2 Criptografia de Chave Privada ou Simétrica

Existem dois requisitos para o uso seguro da criptografia de chave privada (ou simétrica), segundo Stallings (2008, p. 18):

- O algoritmo de criptografia deve ser forte o suficiente para que ninguém seja capaz de decifrar um texto cifrado ou descobrir a chave, mesmo tendo posse de textos cifrados e seus respectivos textos claros originais;
- A chave precisa estar muito bem guardada e protegida, pois se alguém tiver conhecimento da chave e do algoritmo poderá ler toda a comunicação feita usando essa chave.

Na criptografia de chave privada, o algoritmo não precisa ser secreto; apenas a chave precisa ser secreta. Essa é a característica que torna viável o uso generalizado desse tipo de criptossistema. Dessa forma, o principal problema de segurança desse sistema é manter o sigilo da chave (por isso a denominação de chave privada).

O funcionamento desse algoritmo é simples: o emissor produz uma mensagem em texto claro. Uma chave é gerada. Se a chave for gerada na origem da mensagem, então ela precisará ser fornecida ao destino por meio de um canal seguro. Como alternativa, um terceiro poderia gerar a chave e oferecê-la com segurança à origem e ao destino. Essa chave, que é única, é utilizada tanto na cifragem como na decifragem da mensagem, como mostra a Figura 1. O algoritmo criptográfico precisa, então, ser alimentado com a mensagem (texto claro) e com a chave para gerar o texto cifrado. No caminho inverso, o algoritmo precisa ser alimentado com o texto cifrado e com a mesma chave utilizada na cifragem para ser decifrado corretamente, ou seja, a mesma chave é compartilhada entre o emissor e o receptor da mensagem.

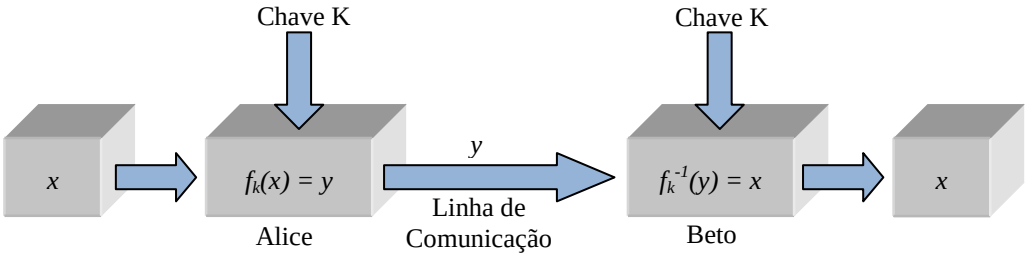

Figura 1 - Criptografia Simétrica. Adaptado de Stallings (2008, p. 18).

Segundo Terada (2008, p. 42), o algoritmo mais usado internacionalmente (até a data de publicação de sua obra – 2008) é o DES (*Data Encryption Standard*), originalmente desenvolvido pela IBM e adotado como padrão nos Estados Unidos em 1977. Foi um avanço significativo por ter sido o primeiro algoritmo de criptografia cujo conhecimento se tornou público, visto que até então todos os algoritmos eram secretos.

O DES trabalha dividindo a mensagem em blocos de 64 bits (8 caracteres) e cifrando cada um desses blocos com uma chave de 56 bits (mais 8 bits de paridade, completando 64 bits). Esse algoritmo parecia ser invencível para a tecnologia da época, contudo, com o grande aumento do poder de processamento dos novos computadores, o DES acabou sendo vencido. Por algum tempo, o 3DES tornou-se um novo padrão popular. Essa técnica utiliza o mesmo algoritmo DES, porém com três chaves de 56 bits, gerando uma força total de 168 bits na chave (STALLINGS, 2008, p. 46-48). Mais detalhes sobre o funcionamento do DES,

incluindo outros algoritmos de chave privada, podem ser encontrados em Terada (2008) e Stallings (2008).

Com a desvantagem de ser uma técnica que exige um maior poder computacional e após também ter sido vencido, solicitou-se, em 1997, um novo *Advanced Encryption Standard* (AES). O vencedor foi Rijndael, algoritmo criado por Vincent Rijmen e Joan Daemen, que logo se tornou o novo padrão de criptografia. No algoritmo AES, o tamanho da chave pode ser de 128, 192 ou 256 bits, porém o tamanho do bloco limita-se a 128 bits (STALLINGS, 2008, p. 92).

1.2.3 Criptografia de Chave Pública ou Assimétrica

O conceito de criptografia por chave pública surgiu em 1.976, quando Whitfield Diffie e Martin E. Hellman publicaram um artigo intitulado "*New directions in criptography*"[1]. Este artigo representou um grande avanço e inspirou diversos algoritmos de chave pública, desenvolvidos a partir de um novo modelo, onde cada usuário possui um par de chaves relacionadas matematicamente (TERADA, 2008, p. 95).

Segundo Stallings (2008, p. 182), "o desenvolvimento da criptografia de chave pública é a maior e talvez a única verdadeira revolução na história da criptografia", e explica que este tipo de criptografia oferece uma mudança radical por dois motivos principais: primeiro, os algoritmos desse modelo são baseados em funções matemáticas[2], diferentemente dos anteriores (algoritmos de chave privada) baseados na substituição e permutação; e segundo, os algoritmos assimétricos envolvem o uso de duas chaves distintas, sendo uma para cifrar e outra para decifrar a mensagem.

O algoritmo funciona da seguinte maneira: se Alice deseja enviar uma mensagem secreta para Beto, basta utilizar a chave pública de Beto para cifrar a mensagem. Quando receber a mensagem, Beto utiliza sua chave privada (que somente Beto conhece) para decifrá-la com segurança. A Figura 2 ilustra esse funcionamento.

Neste modelo, o problema de distribuição de chaves não existe, porque as chaves privadas de cada usuário são geradas localmente por cada um deles. Outra vantagem é que a chave privada pode ser alterada a qualquer momento, sendo necessária apenas a publicação de sua chave pública correspondente. Por outro lado, este modelo utiliza algoritmos que exigem

[1] W. Diffie and M. E. Hellman, *"New directions in cryptography," IEEE Trans. Inform. Theory*, vol. IT-22, pp. 644-654, Nov. 1976.
[2] Grande parte da teoria dos criptossistemas de chave pública baseia-se na teoria dos números.

um poder de processamento muito grande, devido à utilização de uma matemática mais complexa. Este fato gera uma perda de desempenho considerável quando comparados aos algoritmos de chave privada.

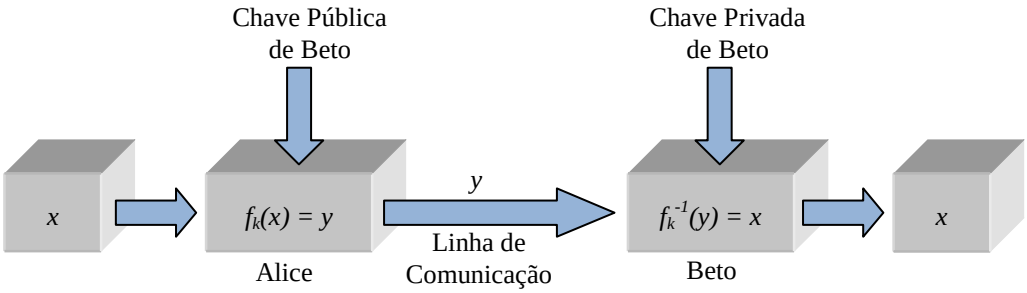

Figura 2 - Criptografia Assimétrica. Adaptado de Stallings (2008, p. 184).

Uma característica importante desse criptossistema é que tendo o conhecimento apenas do algoritmo de criptografia e da chave de criptografia é computacionalmente inviável determinar a chave correta para a decifragem. Além disso, alguns algoritmos (como o RSA) reúnem a seguinte qualidade: qualquer uma das duas chaves pode ser usada para cifrar, sendo a outra, relacionada com a primeira, usada para decifrar.

O RSA foi o primeiro criptossistema de chave pública a ser implementado e, segundo Stallings, o mais utilizado até a publicação de sua obra (2008). Tem esse nome devido aos seus desenvolvedores Ron Rivest, Adi Shamir e Len Adleman. A segurança desse sistema baseia-se na dificuldade de fatorar grandes números, sendo computacionalmente inviável fatorar o produto de dois números primos (STALLINGS, 2008, p. 188-189). Maiores informações a respeito do algoritmo RSA, dentre outros de chave pública, podem ser encontradas em Terada (2008).

1.3 Funções de *Hash*

As funções de *hash* são funções que recebem dados com um número arbitrário de bits, compactando-os e retornando um número fixo de bits, de tal forma que o resultado *hash* forneça sempre uma identidade única para uma mensagem. Caso esta função satisfaça requisitos adicionais poderá ser usada em aplicações criptográficas, protegendo, por exemplo, a autenticidade de mensagens enviadas por canais inseguros (TKOTZ, 2009[a], p. 01).

Dentre as mais diversas aplicações de funções *hash* (adicionadas a outras tecnologias), destacam-se:

- Autenticação de mensagens: fornece proteção em dois aspectos, sendo o primeiro a integridade dos dados (garantia de que os dados não foram alterados) e o segundo a autenticação da origem dos dados (garante a fidelidade da origem ou fonte dos dados);
- Assinaturas digitais: são usadas para proteção de autenticidade oferecendo, adicionalmente, serviço de não-repúdio. Ou seja, é impossível para o remetente negar a autoria de uma mensagem autenticada;
- Datação de documentos: é um serviço que fornece uma autenticação temporal de um documento, fornecendo provas da existência de certos fragmentos da informação antes da data e hora indicada na datação.

As funções *hash* mais conhecidas são: SNEFRU (função de mão única que cria resultados *hash* de 128 ou 256 bits), N-HASH (usa blocos de 128 bits de mensagem e produz um resultado *hash* de 128 bits), MD4 (função *hash* de mão única que produz um valor *hash* de 128 bits, onde MD vem de *Message Digest*), MD5 (versão melhorada do MD4 e também produz um resultado *hash* de 128 bits), SHA (o *Secure Hash Algorithm* foi desenvolvido pelo NIST e pela NSA e produz um *hash* de 160 bits), RIPE-MD (variação do MD4, desenvolvido para o projeto RACE da Comunidade Européia) e HAVAL (variação do MD5, sendo uma função *hash* de mão única e tamanho variável) (TKOTZ, 2009[a], p. 08-09).

CAPÍTULO II – ESTEGANOGRAFIA

Este capítulo traz uma explanação a respeito da esteganografia, abordando conceitos, história, áreas de aplicação e principais técnicas. Este é um capítulo de fundamental importância, pois a esteganografia é o tema central deste trabalho. É importante, também, á medida que acrescenta material de pesquisa a um campo ainda pouco explorado.

2.1 Introdução

De acordo com Sellars (1999 apud ZANELLA, 2002, p. 22) a esteganografia era considerada, até pouco tempo atrás, como o "primo pobre" da criptografia. Contudo, esta situação está sendo revista, à medida que a esteganografia ganha notoriedade e importância, devido à crescente demanda da indústria por proteção dos direitos autorais, através de marcas d'água e impressões digitais.

Ao longo da história foi desenvolvido um vasto conjunto de métodos e técnicas para prover comunicações secretas, como será visto mais adiante, no item "2.4 Esteganografia na História". Atualmente, devido ao avanço tecnológico, a esteganografia é utilizada, inclusive, em diferentes mídias, como texto, imagem, vídeo, áudio entre outras (estas técnicas serão descritas no item "2.6 Técnicas Esteganográficas").

2.2 Conceitos

O modelo geral de esconder dados em outros dados pode ser descrito conforme a Figura 3. A grande área de pesquisa é denominada *Information Hiding* (Ocultação da Informação). No segundo nível da hierarquia estão os canais secretos, a esteganografia, o anonimato e a marcação de *copyright.* É possível observar, no terceiro nível, que a esteganografia se divide em dois tipos, sendo a linguística (quando o processo de ocultar a mensagem depende de propriedades linguísticas) e a técnica (quando a mensagem é fisicamente escondida). É na esteganografia linguística que se encontra a esteganografia digital, tema central do presente trabalho.

O termo "esteganografia" é de origem grega. A palavra "*estegano*" significa oculto, escondido; enquanto que "*grafia*" significa escrita. De acordo com Johnson e Jajodia (1998, p. 26), esteganografia significa literalmente meios de 'escrita encoberta', sendo, portanto, a

arte de esconder informação de um modo que previna a descoberta da mensagem escondida. Kuhn (1995 apud ZANELLA, 2002, p. 22) observa que o objetivo da esteganografia é embutir uma mensagem dentro de outra sem importância para que um inimigo não detecte a sua presença. Em outras palavras, é uma forma de se escrever algo que não será notado, ou seja, tem o objetivo de passar despercebido aos olhares daqueles que não deveria, por algum motivo, ter acesso àquilo que foi escrito.

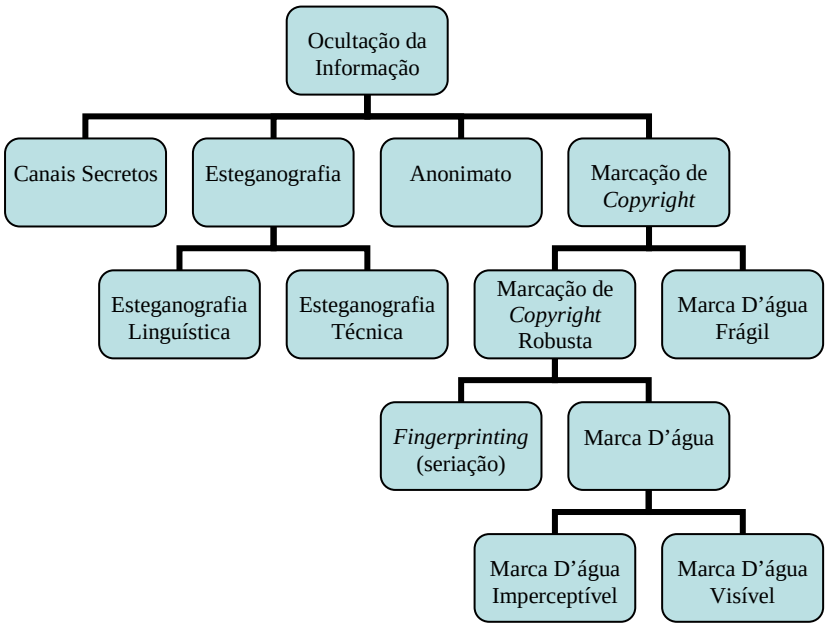

Figura 3 - Classificação da Técnica de Ocultação da Informação. Adaptado de Petitcolas; Anderson; Kuhn (1999, p. 1063).

O dado embutido (*embedded data*) é a mensagem que alguém deseja enviar secretamente. Normalmente a mensagem é escondida em um arquivo inócuo chamado objeto de cobertura (*cover-object*) ou imagem de cobertura (*cover-image*) para arquivos de imagem, produzindo – após o processo de embutir a mensagem – o estego-objeto (*stego-object*) ou a estego-imagem (*stego-image*).

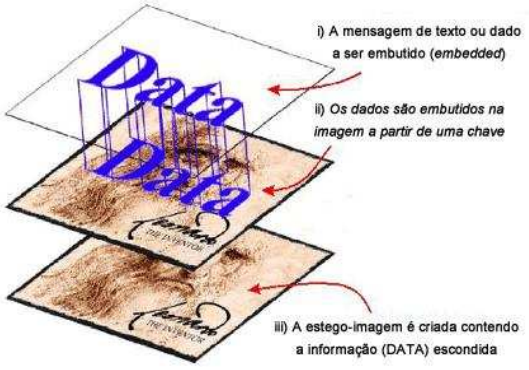

Figura 4 - Ocultando uma Mensagem (ROCHA; COSTA; CHAVES, 2004, p. 03).

Opcionalmente uma estego-chave (*stego-key*) é usada para controlar o processo de ocultação de forma a restringir a detecção e/ou recuperação dos dados embutidos (PETITCOLAS; ANDERSON; KUHN, 1999, p. 1063), conforme mostra a Figura 4.

2.3 Esteganografia X Criptografia

Johnson e Jajodia argumentam que diferentemente da criptografia, que embaralha uma mensagem dificultando o entendimento do seu conteúdo, a esteganografia esconde a mensagem tornando-a 'invisível' não despertando suspeita de sua existência (JOHNSON; JAJODIA, 1998, p. 26). No mesmo sentido Tkotz (2009[b], p. 01) ressalta essa diferença explicando que a esteganografia camufla a mensagem mascarando a sua presença, ou seja, enquanto que a criptografia pretende tornar a mensagem ininteligível, a esteganografia procura esconder a existência dessa mensagem.

2.4 Esteganografia na História

Ao contrário do que possa parecer, a esteganografia não é um artifício recente, ou moderno. Este recurso já era utilizado a milhares de anos atrás, principalmente na área militar.

Segundo Tkotz (2009[b], p. 01), o primeiro uso confirmado da esteganografia remonta ao século V a.C, quando o filósofo grego Heródoto, no seu livro "As Histórias", relata a história de um subordinado de Aristágoras de Mileto que para manter um contato secreto com seu superior, raspa a cabeça de um escravo e escreve nela uma mensagem. Após esperar que os cabelos do escravo crescessem novamente, mandou-o para o encontro de Aristágoras com a instrução de que deveriam raspar seus cabelos.

Outro caso tirado de "As Histórias" e lembrado por Tkotz (2009[b], p. 01) diz respeito ao rei Demaratos, que aproveitou um transporte de caixas de cera com destino à Grécia para enviar aos gregos um aviso secreto sobre um ataque iminente dos persas. Demaratos retirou, então, a cera de alguns tabletes, gravou na madeira a mensagem e recobriu os tabletes com cera novamente. Os gregos rasparam a cera e leram a mensagem no fundo da caixa, o que garantiu a vitória dos gregos sobre os persas.

Também se observou casos de esteganografia na China antiga, segundo Tkotz (2009[b], p. 01). A autora relata que as mensagens eram escritas sobre uma seda fina que era

transformada em uma bolinha, recoberta por cera e depois engolida por um mensageiro. Quando o mensageiro chegava ao seu destino, expelia a bolinha com a mensagem.

De acordo com Coutinho (2008), por volta do ano de 1.500, um abade alemão chamado Johannes Trithemius, considerado o pai da criptografia, escreveu um trabalho de três volumes intitulado "*Steganographia*", onde descrevia técnicas de criptografia nos dois primeiros volumes e escrevia sobre ocultismo e astrologia no terceiro. Mais tarde, o Dr. Thomas Ernst (professor do *La Roche College* - Estados Unidos) e o Dr. Jim Reeds (Laboratórios da AT&T - Estados Unidos) descobriram que a obra foi escrita utilizando técnicas esteganográficas e mostra maneiras de se ocultar mensagens secretas em textos aparentemente inofensivos.

A historiadora Tkotz relata também que o cientista Giovanni Porta, ainda no século XVI, descobriu como ocultar uma mensagem em um ovo cozido. Em tal façanha o cientista "escrevia sobre a casca do ovo cozido com uma tinta contendo uma onça de alume (aproximadamente 29 gramas) diluída em cerca de meio litro de vinagre. A solução penetrava na casca e se depositava sobre a superfície branca do ovo. Depois, bastava o destinatário abrir o ovo para ler a mensagem" (TKOTZ, 2009[b], p. 01-02).

Tkotz (2009[b], p. 02) também faz referência a uma técnica aperfeiçoada ao longo da história. O historiador grego Enéias teve a idéia de fazer minúsculos furos em certas letras de um texto qualquer e a sucessão dessas letras marcadas fornecia a mensagem secreta. Dois mil anos mais tarde, os remetentes ingleses empregavam o mesmo método para evitar o pagamento das taxas de correios. Como o envio de uma carta ficava muito caro na época, os ingleses faziam furinhos de agulha nos jornais – que eram isentos da taxa – e os enviavam com suas mensagens. Esse método também foi utilizado pelos alemães na Primeira Guerra Mundial, sendo aperfeiçoado na Segunda Guerra marcando-se as letras de jornais com tintas "invisíveis" ao invés de se fazer furos. Também na Segunda Guerra, os espiões alemães se utilizaram dos micropontos para enviar suas mensagens secretas. Eram fotografias do tamanho de um ponto, que após serem ampliadas revelavam uma mensagem.

Numa época muito mais recente, em 1999, Catherine Taylor Clelland, Viviana Risca e Carter Bancroft publicaram um artigo interessante na revista *Nature* intitulado "*Hiding messages in DNA microdots*" (escondendo mensagens em micropontos de DNA). Tal artigo descreve como é possível se utilizar de material genético, através de cadeias de DNA, para compor uma "mensagem genética". Este método esteganográfico permite, por exemplo, que uma determinada empresa produza um produto que contenha, em suas próprias moléculas, a marca da empresa a fim de evitar imitações (TKOTZ, 2009[b], p. 01).

Nos dias atuais cogita-se a hipótese de que grupos terroristas se utilizem de técnicas modernas de esteganografia para se comunicarem. Coutinho (2008) menciona duas reportagens publicadas pelo jornal americano *"USA Today"* de 5 de fevereiro de 2001, afirmando (embora sem provas concretas) "que o grupo terrorista Al-Qaeda estaria se comunicando usando mensagens ocultas em imagens de sites da Internet e em mensagens de *spam*, lixo eletrônico". Menciona também uma reportagem do dia 30 de outubro de 2001, no jornal "*The New York Times"*, dizendo "que os ataques de 11 de setembro de 2001 teriam sido organizados pela Internet utilizando técnicas esteganográficas".

2.5 Áreas de Aplicação

A esteganografia possui inúmeras aplicações, podendo estar presente em diversas áreas como, por exemplo, a militar, a médica, a artística etc.

Lin e Delp (2002 apud ZANELLA, 2002, p. 24), destacam três importantes aplicações no que se refere a imagens: proteção de direitos autorais, classificação e comunicações secretas.

Na proteção dos direitos autorais é possível, através da esteganografia, inserir uma marca registrada ou assinatura com marca d'água em uma imagem (ou também documentos, vídeos, músicas, programas) com a finalidade de identificá-la como uma propriedade intelectual. A marca d'água permite a detecção de uma possível alteração da imagem, garantindo, também, a sua autenticidade (LIN; DELP, 2002 apud ZANELLA, 2002, p. 24). No tocante a músicas, um programa pode procurar no sinal de rádio por marcas inseridas nas mesmas. Caso a marca seja inteiramente decodificada do sinal sabe-se que a emissora de rádio tocou a música sem autorização (JULIO; BRASIL; NEVES, 2007, p. 80).

A classificação refere-se a informações ou descrições - como títulos, datas ou anotações - inseridas em uma mídia. Com essa técnica é possível deixar um mecanismo de busca mais eficiente, inserindo uma palavra chave nas imagens de uma base de dados, por exemplo. Pode-se, também, inserir marcadores de tempo nas imagens pertencentes a uma sequência de vídeos para que se consiga uma sincronização com o áudio (LIN; DELP, 2002 apud ZANELLA, 2002, p. 25).

No que se refere às comunicações secretas, Zanella observa que o uso da criptografia no processo de envio de uma mensagem secreta poderia chamar a atenção de um possível interceptador dessa mensagem. Nesse contexto, o uso da esteganografia seria muito

apropriado, pois protegeria o remetente, o destinatário e a própria mensagem (LIN; DELP, 2002 apud ZANELLA, 2002, p. 25).

Julio, Brasil e Neves (2007, p. 79-80) salientam que as técnicas de ocultação de informações também podem ser utilizadas de forma maliciosa. Em uma rede de computadores militares, onde existem vários níveis de segurança, um vírus ou programa malicioso se propaga dentro do sistema passando de níveis de segurança inferiores para os superiores. Quando seu objetivo é alcançado, as informações sigilosas são passadas para os níveis de seguranças inferiores com o auxílio de técnicas de ocultação de informações, que conseguem burlar o sistema escondendo informações confidenciais em arquivos comuns.

Na área da indústria médica também existe importantes aplicações de técnicas esteganográficas. Há uma forma de comunicação padrão (DICOM – *Digital Imaging and Communications in Medicine*) que separa a imagem médica de um paciente de informações importantes como nome, data, médico etc. Em alguns casos, a ligação entre essa imagem e seus dados é perdida, o que não aconteceria se essa informação estivesse oculta dentro da própria imagem (FILHO et al., 2005 apud JULIO; BRASIL; NEVES, 2007, p. 27).

Coutinho (2008) mostra que a esteganografia está presente, também, em equipamentos como máquinas fotográficas, filmadoras e impressoras[3]. Utilizando-se de uma de suas técnicas podem inserir números de série, datas e horários de criação nas mídias que produzem com a finalidade de tornar as fraudes rastreáveis. Segundo este autor, até mesmo os cheques de viagem estão protegidos por esteganografia. Para garantir a autenticidade do documento, a palavra ‘VOID’ é escrita no cheque com tinta invisível, revelando, através de luz ultravioleta, que o cheque está cancelado.

2.6 Técnicas Esteganográficas

As técnicas de esteganografia existentes são inúmeras, ficando a cargo apenas da criatividade humana o desenvolvimento de outras. A seguir, segue uma breve descrição de alguns dos tipos mais utilizados.

[3] As impressoras a laser coloridas de marcas como HP e Xerox acrescentam pontos amarelos minúsculos em documentos impressos por elas.

2.6.1 Esteganografia em Texto

O envio de um simples documento pela Internet pode ocultar um grande problema de segurança. Zanella (2002, p. 26) observa que a utilização de técnicas de esteganografia nesse tipo de mídia (arquivo texto) “diversificam-se desde o envio de mensagens embutidas no conteúdo do documento em texto, até controles para fins de autenticidade do mesmo”.

Um problema típico é a distribuição ilegal de documentos via *e-mail*, sem o devido pagamento dos direitos legais ao autor do documento original. Com o objetivo de combater a pirataria, por exemplo, foi proposto um método de esteganografia em texto, onde “seria embutida uma palavra-código para marcar um documento impresso, identificando o destinatário do documento quando o mesmo fosse recuperado” (ZANELLA, 2002, p. 26). A inserção da palavra-código no documento é feita através da alteração das características particulares do próprio documento. Tais características é o que determina qual o método de codificação será utilizado. Os tipos de codificação podem ser *line-shift*, *word-shift* e de características; os detalhes de cada um não serão abordados neste trabalho, podendo ser encontrados em Sellars (1999).

2.6.2 Esteganografia em Imagem

Essa técnica somente foi possível, segundo Zanella (2002, p. 29), devido à evolução computacional, juntamente com o crescente desenvolvimento do processamento de imagens digitais.

A esteganografia em imagem é muito utilizada devido ao grande tráfego de imagens pela Internet. Como o fluxo de troca de imagens ou fotos é bastante grande na rede, uma foto (aparentemente comum, mas contendo uma informação sigilosa), por exemplo, acaba passando despercebida aos olhares de um usuário mal intencionado.

Essa técnica faz uso de uma limitação do sistema visual humano (HVS), que é incapaz de perceber pequenas mudanças em padrões de cor. Dessa forma, qualquer informação que pode ser representada por um conjunto de bits, esteja ela na forma de texto, imagem, gráfico etc., pode ser inserida dentro de uma imagem sem ser detectada. Os bits da informação sigilosa são inseridos nos setores da imagem onde não irão afetar a aparência da mesma, podendo também ser inseridos de maneira dispersa por toda a imagem, de forma a

tornar a detecção da mensagem oculta mais difícil (SELLARS, 1999 apud ZANELLA, 2002, p. 29).

Dentre as diversas maneiras existentes para ocultar uma informação dentro de uma imagem, as técnicas mais utilizadas são a inserção nos bits menos significativos (LSB *insertion*), filtragem e mascaramento e algoritmos e transformações (JOHNSON; JAJODIA, 1998, p. 28; WAYNER, 2000 apud JULIO; BRASIL; NEVES, 2007, p. 59; SELLARS, 1999 apud ZANELLA, 2002, p. 30). Tais técnicas serão abordadas a seguir.

2.6.2.1 Inserção nos Bits Menos Significativos (LSB)

O algoritmo LSB (*Least Significant Bit*), é a técnica mais comum em se tratando de esteganografia em imagem digital. Talvez por esse motivo, existem diversas maneiras diferentes de se implementar tal algoritmo. Contudo é uma técnica frágil porque a informação pode se perder após um simples processo de compressão de dados com perda (compressão *lossy*) (JOHNSON; JAJODIA, 1998, p.28).

Na sua forma mais básica a informação secreta é inserida (bit a bit) nos bits menos significativos de cada pixel da imagem. Esse algoritmo é simples, mas desperdiça espaços potenciais para ocultação de dados.

Outra versão do LSB é a inserção nos bits menos significativos de cada byte de cada pixel. Essa forma de implementação acarreta mais mudanças na aparência da imagem, porém não é expressiva a ponto de ser percebida pelo olho humano. Por outro lado, dessa forma é possível ocultar uma quantidade maior de dados na mesma imagem. Um esquema mais sofisticado de implementação desse algoritmo é selecionar os locais de inclusão dos dados. Esse tipo de técnica é uma forma de mascaramento em imagens muito difícil de ser detectada, pois utiliza inserção de dados em pixels não sequenciais (PETITCOLAS; ANDERSON; KUHN, 1999; POPA, 1998 e WAYNER, 2002 apud JULIO; BRASIL; NEVES, 2007, p. 60-61).

O algoritmo usado no EstegoChip foi o LSB *insertion*, embutindo dados sempre nos bits menos significativos de cada byte de uma imagem Bitmap, codificada em 24 bits por pixel. Ou seja, com este algoritmo cada pixel tem a capacidade de armazenar três bits de informação (o item “4.2 Implementação do LSB” fornece mais detalhes sobre a implementação deste algoritmo).

2.6.2.2 Filtragem e Mascaramento

Com este tipo de técnica a informação é escondida através da marcação de uma imagem, similarmente ao funcionamento das marcas d'água em papel. Segundo Wayner (2002 apud JULIO; BRASIL; NEVES, 2007, p. 61), são técnicas mais robustas do que a LSB *insertion* porque "geram estego-imagens imunes a compressão e recorte. No entanto, são técnicas mais propensas a detecção". As técnicas de filtragem e mascaramento alteram os bits mais significativos da imagem, por essa razão devem ser utilizadas imagens em tons de cinza, evitando-se as coloridas, situação em que tais técnicas não seriam eficazes.

Zanella aponta para uma divergência relevante entre dois importantes autores, Duncan Sellars (SELLARS, 1999) e David Holmes (HOMES, 2002), a respeito da marca d'água ser ou não considerada uma forma de esteganografia. Sellars afirma que a marca d'água acrescenta uma informação – que pode ficar visível – à imagem, enquanto que a esteganografia oculta uma informação. De outro lado, Holmes defende a ideia de que a marca d'água pode ser implementada de forma a ficar oculta, sendo utilizada para a proteção de *copyright* e ferramentas de licença (ZANELLA, 2002, p. 33).

2.6.2.3 Algoritmos e Transformações

Um problema encontrado na técnica LSB, segundo Julio; Brasil; Neves (2007, p. 61), é a compressão de imagens. As técnicas baseadas em algoritmos e transformações conseguem tirar proveito deste obstáculo, utilizando a Transformada Discreta de Fourier, Transformada Discreta de Cosseno e Transformada Z. Maiores detalhes sobre transformadas podem ser encontradas em Gonzalez; Woods (2000).

Os dados embutidos no domínio de transformação residem em áreas mais robustas, fornecendo maior resistência contra processamento de sinal. Esse tipo de técnica é amplamente utilizado para marca d'água robusta, configurando-se como uma das mais sofisticadas técnicas de mascaramento de informações conhecidas (POPA, 1998 apud JULIO; BRASIL; NEVES, 2007, p. 61).

No geral, essas técnicas aplicam uma transformação em blocos de 8x8 pixels da imagem, selecionando, em cada um, os coeficientes redundantes ou menos significantes. Estes coeficientes são utilizados para atribuir a mensagem a ser embutida na imagem (POPA, 1998 apud JULIO; BRASIL; NEVES, 2007, p. 61).

2.6.3 Esteganografia em Vídeo

A forma de esconder uma informação dentro de um vídeo é similar à esteganografia em imagem, exceto pelo fato de esconder a informação em cada *frame* (quadro) do arquivo de vídeo. De acordo com Julio, Brasil e Neves (2007, p. 69), para se ocultar uma informação dentro de um vídeo normalmente é utilizado o método TDC (Transformada Discreta de Cosseno), comumente utilizado na compressão de imagens digitais.

Segundo Coutinho (2008), “é possível utilizar uma sequência pseudo-aleatória de semente secreta para determinar os quadros que conterão dados ocultos” tornando a técnica ainda mais segura. Caso o arquivo possua áudio, pode-se utilizar a esteganografia para sons, podendo também haver uma combinação das duas técnicas, ou seja, esteganografia nos quadros do vídeo e no áudio.

2.6.4 Esteganografia em Áudio

A utilização desse tipo de mídia é bastante complexa devido ao sistema auditivo humano (SAH), que consegue trabalhar em uma vasta faixa de freqüência. As técnicas devem, portanto, explorar as vulnerabilidades do ouvido humano para obterem sucesso e alcançarem seus objetivos (JULIO; BRASIL; NEVES, 2007, p. 69).

De acordo com Julio, Brasil e Neves (2007, p. 69), “uma das primeiras considerações a serem feitas é o ambiente onde o som trafegará entre a origem e o destino” e destaca que, para a escolha do método de esteganografia adequado, devem ser considerados a representação digital do sinal que será usado e o caminho de transmissão do sinal.

Alguns métodos principais são a codificação *low-bit* (bits baixos), codificação em fase, *spread spectrum* e escondendo informações com eco. Mais detalhes em Julio; Brasil; Neves (2007, p. 70-74) e Sellars (1999 apud ZANELLA, 2002, p. 37-39).

CAPÍTULO III - CARACTERIZAÇÃO DE IMAGENS

Este capítulo explica como é representada uma imagem digital, descreve a estrutura dos formatos dessas imagens e cita quais são os principais, explicando com mais detalhes o formato BMP no item “3.3.1 Formato Bitmap (BMP)”; em seguida traz uma introdução às cores e seus sistemas, reservando um item para o sistema RGB (3.4.1.2.1 Sistema RGB de Cores). Estes temas fornecem as informações adicionais e necessárias para um melhor entendimento deste trabalho, visto se tratar de esteganografia em imagens digitais.

3.1 Introdução

Existem duas grandes categorias de imagens, sendo as de formato *raster* e as de formato vetorial. A imagem de formato *raster* é a forma mais comum de representação de imagem encontrada em uso atualmente, pois pode representar os efeitos de cor, luz e sombra nos objetos e permite manipular o menor detalhe da figura (ZANELLA, 2002, p. 04). A imagem de formato vetorial é descrita em termos de equações matemáticas, ou seja, os pontos, linhas, polígonos, círculos, elipses e outras formas geométricas complexas são representados através de parâmetros e coeficientes matemáticos. Por esse motivo, a representação através de vetores é mais adequada para imagens com predominância de linhas, como diagramas e gráficos simples (FILHO; NETO, 1999, p. 243). O formato vetorial não será contemplado neste trabalho, porém alguns aspectos podem sem encontrados, de forma sucinta, em Filho; Neto (1999).

3.2 Representação de Imagens Digitais

De acordo com Gonzalez e Woods (2000, p. 4-5):

> O termo *imagem monocromática*, ou simplesmente *imagem*, refere-se à função bidimensional de intensidade da luz *f(x,y)*, onde *x* e *y* denotam as coordenadas espaciais e o valor de *f* em qualquer ponto *(x,y)* é proporcional ao brilho (ou *níveis de cinza*) da imagem naquele ponto.

Considera-se como imagem digital uma matriz cujos índices de linhas e de colunas identificam um ponto na imagem, sendo que o valor correspondente desse elemento da matriz identifica o nível de cinza naquele ponto. Os elementos dessa matriz (vide Figura 5) são

chamados de elementos da imagem, elementos da figura, “pixels” ou “pels”. Ou seja, um pixel é a menor parte de uma imagem e contém informações que determinam suas características; quanto mais pixels por polegada, melhor é a qualidade da imagem. Com relação às cores, quanto maior o número de bits por pixel (profundidade de bits), maior é a matiz de tons presente na imagem.

$$f(x,y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & & & \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

Figura 5 - Imagem Digital. Adaptado de Gonzalez e Woods (2000, p. 22).

A qualidade de uma imagem, portanto, é definida por duas variáveis: a densidade de pontos na matriz, normalmente definida em pixels por polegada (DPI); e a resolução espectral ou número de cores, normalmente definido por um número de bits disponíveis para sua codificação. Um exemplo de pixel e seus respectivos valores R, G e B é ilustrado com a Figura 6.

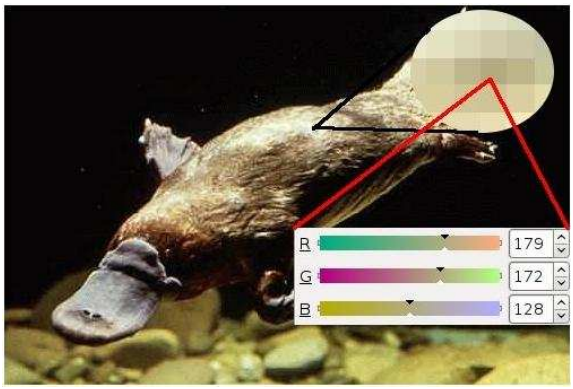

Figura 6 - Um Pixel e seus Valores RGB (COUTINHO, 2008).

Neste trabalho serão usadas imagens no formato BMP de 24 bits por pixel, sendo um byte para cada canal de cor do modelo RGB (explicado adiante), logo cada cor terá 256 tons possíveis, com um total de 16.777.216 tons de cores para cada pixel. Contudo a técnica de esteganografia abordada neste trabalho (LSB - Inserção nos Bits Menos Significativos) pode ser implementada em qualquer outro formato, porém com a ressalva de que os formatos que

usam compressão *lossy* – JPEG, por exemplo – não são recomendados devido ao sistema de compressão com perdas. De acordo com Johnson e Jajodia (1998, p. 27) os softwares de esteganografia não recomendam a utilização de imagens JPEG, indicando o uso de imagens BMP de 24 bits.

3.3 Formatos de Imagens Digitais

Existe uma gama de formatos diferentes de arquivos. Esses arquivos variam em seus detalhes, mas compartilham da mesma estrutura geral, que consiste em um cabeçalho, dados da imagem e outras informações (MURRAY, 1994 apud ZANELLA, 2002, p. 06).

O cabeçalho encontra-se no início do arquivo. É iniciado com um identificador do arquivo que geralmente é único e é atribuído pelo próprio criador do formato. Normalmente é composto por campos fixos, possuindo informações como mapa de cores, identificação do arquivo, versão do arquivo, número de linhas por imagem, número de pixel por linha, número de bits por pixel, tipo de compressão utilizado etc. (ZANELLA, 2002, p. 06).

Os dados da imagem acomodam o mapa de cores ou outra estrutura de dados presente. Geralmente vêm logo após o cabeçalho, porém isto não é uma regra; quando esta estrutura não é seguida, o cabeçalho indica a posição inicial destes dados (ZANELLA, 2002, p. 06).

Outra estrutura que pode compor um arquivo de imagem é o rodapé: similar ao cabeçalho, mas se encontra no final do arquivo. Normalmente é acrescido ao formato quando se necessita de uma extensão para acomodar novos tipos de dados. Como o rodapé sempre vem antes dos dados sua localização é determinada com a fixação de um *offset* no final do arquivo (ZANELLA, 2002, p. 06).

Dentre os principais tipos de formatos de imagens digitais encontram-se BMP, GIF, TIFF, PNG e JPEG dentre outros. Este item descreve o formato Bitmap (BMP). Uma descrição sucinta a respeito de outros formatos pode ser encontrada em Filho; Neto (1999) e Zanella (2002).

O formato Bitmap (BMP) – conhecido também como DIB (*Device Independent Bitmap*) – é proprietário da Microsoft compatível ao sistema operacional Windows (LOPES, 2002 apud ZANELLA, 2002, p. 07). Ele permite descrever imagens a cores de 1, 4, 8, 16, 24 ou 32 bits por pixel, podendo representar imagens com 2, 16, 256, 2^{16}, 2^{24} ou 2^{32} tonalidades de cores respectivamente, utilizando um mapa de cores nos três primeiros casos e uma

máscara para cada cor nos outros. O BMP ainda permite a compressão opcional sem perdas do conteúdo das imagens com 16 ou 256 cores através do algoritmo RLE (*Run Length Encoding*) adaptado ao número de bits por pixel (MURRAY, 1996 apud ZANELLA, 2002, p. 08).

Toda informação pertencente a um arquivo BMP está estruturada em três blocos colocados no arquivo na seguinte ordem, obrigatoriamente (LOPES, 2002 apud ZANELLA, 2002, p. 08):

- *Bitmap File Header*: descreve o arquivo;
- *Bitmap Info*: descreve o bitmap da imagem;
- Dados da Imagem: formados pelos pixels da imagem.

A Figura 7 ilustra um exemplo de estrutura de um arquivo BMP.

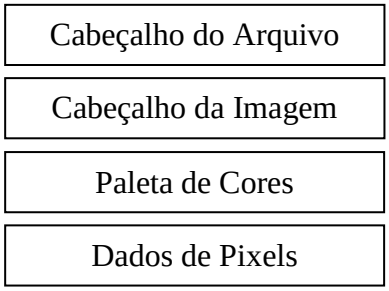

Figura 7 - Organização do Arquivo Bitmap. Adaptado de Agostini (1999 apud ZANELLA, 2002, p. 08).

Uma vez que este trabalho trata de imagens codificadas em 24 bits por pixel, mais detalhes deste formato específico serão fornecidos mais adiante (vide item “4.2 Implementação do Algoritmo LSB”**)**.

3.4 Fundamentos de Cores

Como as imagens coloridas, em geral, são mais atraentes do que as monocromáticas entraremos no assunto das cores para melhor entendermos as imagens coloridas. Segundo Gonzalez e Woods (2000, p. 157) uma “motivação para o uso de cores é que o olho humano pode discernir milhares de tons e intensidades de cores, comparado a cerca de apenas duas dúzias de tons de cinza”.

“O espectro de cores pode ser dividido em seis amplas regiões: violeta, azul, verde, amarelo, laranja e vermelho”, porém, conforme Figura 8, “quando visto em cores reais

nenhuma cor no espectro termina abruptamente, mas cada cor mistura-se suavemente com a próxima" (GONZALEZ; WOODS, 2000, p. 157). Este fenômeno foi demonstrado pelo cientista Isaac Newton, em 1666, quando o mesmo passou um raio de luz solar através de um prisma de vidro produzindo um arco-íris; em seguida, Newton transferiu seu arco-íris para um segundo prisma que reconstituiu o raio de luz branca original. Com isso, concluiu-se que as cores não estão no vidro, mas na luz; e a luz branca, na verdade, é uma mistura de todas as cores de espectro visível (FILHO, 1997 apud ZANELLA, 2002, p. 15).

Figura 8 - Espectro Visível de Energia Eletromagnética (GONZALEZ; WOODS, 2000, p. 157).

"A luz visível é composta de uma banda de frequência relativamente estreita no espectro de energia eletromagnética", sendo que a luz cromática abarca desde aproximadamente 400 até 700 nm desse espectro (GONZALEZ; WOODS, 2000, p. 157-158). As frequências do espectro eletromagnético podem ser observadas na Figura 9.

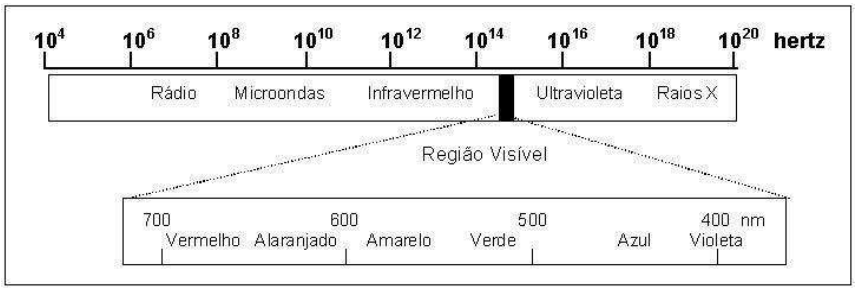

Figura 9 - Frequências do Espectro Eletromagnético (AZEVEDO; CONCI, 2003, p. 186).

Com base neste conhecimento podem-se tirar algumas vantagens do uso das cores, principalmente em se tratando de imagens. Por exemplo, a cor pode melhorar a legibilidade da informação, possibilitar a geração de imagens mais realistas, focar a atenção do observador e passar emoções. Enfim, segundo Azevedo e Conci (2003, p. 181) "o uso de cores torna o processo de comunicação mais eficiente", tendo as cores o poder de interferir nos sentidos, emoções e intelecto das pessoas.

Outro conceito importante é o de sistema de cores. Um sistema de cores, segundo Azevedo e Conci (2003, p. 189), “é um modelo que explica as propriedades ou o comportamento das cores num contexto particular, facilitando a especificação das cores em uma forma padrão”. Como não existe um sistema capaz de explicar todos os aspectos relacionados à cor, são utilizados sistemas diferentes para ajudar a descrever as diferentes características da cor que são percebidas pelo ser humano.

Alguns exemplos de sistemas de cores são RGB (usados em monitores coloridos e processamento de imagens), CMYK ou CMY (impressoras coloridas e processamento de imagens), YIQ (transmissão de TV colorida), HSI e HSV (ambos para manipulação de imagens coloridas). O sistema RGB será explicado com maiores detalhes mais adiante. Mais informações a respeito dos outros sistemas podem ser encontradas em Azevedo; Conci (2003) e Gonzalez; Woods (2000).

Os sistemas de cores podem ser subtrativos ou aditivos.

3.4.1 Subtrativos

As cores primárias são as duas ou três que um sistema utiliza para produzir outras cores. Nos sistemas subtrativos, segundo Gonzalez e Woods, “uma cor primária é definida como sendo aquela que subtrai ou absorve uma cor primária da luz e reflete ou transmite as outras duas”. Dessa forma as cores primárias dos pigmentos desse sistema são magenta, ciano e amarelo e as cores secundárias são vermelho, verde e azul. Uma combinação apropriada dos três pigmentos primários, ou um secundário com o seu primário oposto, produz o preto (GONZALEZ; WOODS, 2000, p. 158). Essas cores são mostradas na Figura 10.

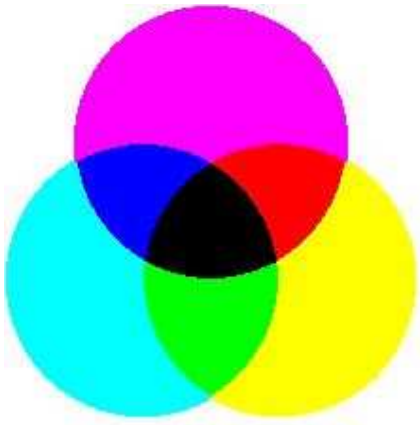

Figura 10 - Mistura de Pigmentos – Primárias Subtrativas (GONZALEZ; WOODS, 2000).

3.4.2 Aditivos

Nos sistemas aditivos as cores primárias são o vermelho, verde e azul, sendo que as intensidades das cores primárias são adicionas para produzir outras cores. As cores primárias podem ser adicionadas para produzir as cores secundárias da luz – magenta (vermelho e azul), ciano (verde e azul) e amarelo (vermelho e verde) (GONZALEZ; WOODS, 2000, p. 158). Zanella explica que, neste sistema, “o preto é gerado pela ausência de qualquer cor, indicando que nenhuma luz está sendo transmitida; e o branco, que é a mistura de todas as cores, indica que uma quantidade máxima de vermelho, verde e azul está sendo transmitida” (ZANELLA, 2002, p. 17). A Figura 11 ilustra as três cores primárias e suas combinações para produzir as cores secundárias.

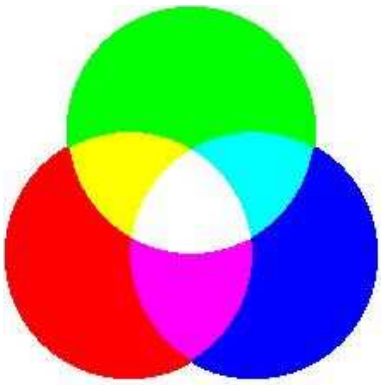

Figura 11 - Mistura de Cores – Primárias Aditivas (GONZALEZ; WOODS, 2000).

O RGB é um padrão pertencente ao sistema aditivo de cores. Baseia-se num sistema de coordenadas cartesianas, onde cada cor aparece nos seus componentes espectrais primários vermelho (*red*), verde (*green*) e azul (*blue*). O subespaço de cores é o cubo da Figura 12,

> [...] no qual os valores RGB estão nos três cantos; ciano, magenta e amarelo estão nos outros três cantos; preto está na origem; e branco está no canto mais distante da origem. Nesse modelo, a escala de cinza estende-se do preto até o branco ao logo da linha juntando estes dois pontos, e as cores são pontos sobre ou dentro do cubo, definidas por vetores estendendo-se a partir da origem (GONZALEZ; WOODS, 2000, p. 160).

Assume-se que todos os valores de R, G e B estão no intervalo [0,1].

As imagens desse modelo consistem em três planos de imagens independentes, um para cada cor primária; quando alimentadas num monitor RGB, as três imagens combinam-se sobre a tela para produzir uma imagem de cores compostas. Para Gonzalez e Woods “um dos

melhores exemplos da utilização do modelo RGB está no processamento de dados de imagens multiespectrais aéreas e de satélite” (GONZALEZ; WOODS, 2000, p. 160).

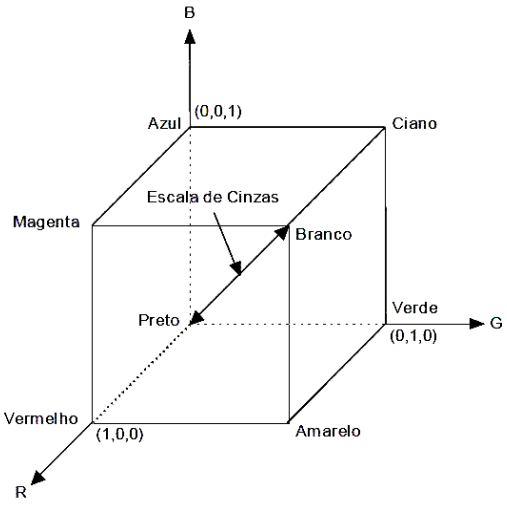

Figura 12 - Cubo de Cores RGB (GONZALEZ; WOODS, 2000, p. 161).

CAPÍTULO IV – ESTEGOCHIP

Este capítulo apresenta o EstegoChip, o algoritmo implementado, a linguagem e a tecnologia empregada, os trabalhos correlatos e a arquitetura desenvolvida. É um capítulo de extrema importância porque está diretamente relacionado com o desenvolvimento do projeto, ou seja, a descrição e implementação do hardware. Com relação à arquitetura, foi colocado em prática o conteúdo absorvido ao longo da pesquisa.

4.1 Introdução

O EstegoChip foi descrito em linguagem VHDL e implementado em FPGA, em um chip *Spartan2E* XC2S200E do pacote PQ208. Possui uma arquitetura projetada para executar uma técnica de esteganografia em imagem denominada LSB, com a finalidade de minimizar os problemas de segurança através de uma solução desenvolvida em hardware. Todo o projeto foi desenvolvido com a ferramenta de descrição, simulação, síntese e implementação ISE 9.1i da fabricante *Xilinx*. A Figura 13 exibe uma placa de FPGA com um chip *Spartan2E* do fabricante *Xilinx*.

Figura 13 - Placa de FPGA com Chip *Spartan2E* (SPARTAN2E, 2010).

4.2 Circuitos Programáveis (FPGAs) e Linguagem VHDL

“A implementação de circuitos digitais complexos era uma ciência dominada apenas por grandes empresas ou universidades de renome internacional” (ORDONEZ et al., 2003, p. 02). Isso acontecia porque a maioria dos circuitos digitais era projetada a partir de equações

booleanas, baseada em *flip-flops* (memória de um bit) e portas lógicas. Apesar de qualquer sistema poder ser representado por equações booleanas, essa técnica é impraticável para projetos grandes que contenham muitos componentes devido ao grande número de equações (GRUPO DE MICROELETRÔNICA, [s.d], p. 01). A Figura 14 ilustra a composição de um circuito pela técnica booleana.

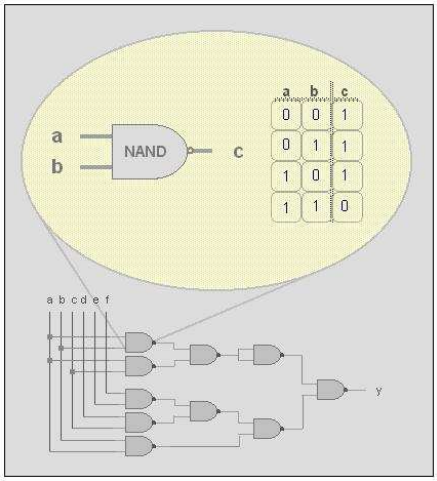

Figura 14 - Composição de um Circuito pela Técnica Booleana (GRUPO DE MICROELETRÔNICA, [s.d], p. 03).

Entretanto, Ordonez et al. lembram que devido o avanço da tecnologia surgiram os FPGAs e afirma que “esta tecnologia inovadora está viabilizando a construção e prototipação de circuitos digitais complexos sem a necessidade de muitos recursos computacionais e financeiros” (ORDONEZ et al., 2003, p. 02). Um ambiente relativamente simples e de baixo custo está popularizando, cada vez mais, esta tecnologia.

O grupo de microeletrônica da Universidade Federal de Itajubá acrescenta que “a maior dificuldade nos métodos tradicionais de projeto é a conversão manual da descrição do projeto em um conjunto de equações Booleanas”. Quando se utiliza as linguagens de descrição de hardware – HDL (*Hardware Description Languages*) – esta dificuldade é eliminada, podendo, por exemplo, implementar um circuito usando HDL a partir de uma tabela verdade, ou da descrição de uma máquina de estado (GRUPO DE MICROELETRÔNICA, [s.d], p. 01).

As linguagens HDLs mais conhecidas são: VHDL, VERILOG, AHDL, ABEL entre outras. A seguir, será explicado um pouco sobre os circuitos programáveis, denominados FPGAs, utilizados neste trabalho para a implementação do chip esteganográfico e, na sequência, um pouco sobre a linguagem VHDL, utilizada para a descrição dos algoritmos.

4.2.1 Circuitos Programáveis (FPGAs)

Os FPGAs (*Field Programmable Gate Array*) “são circuitos programáveis compostos por um conjunto de células lógicas ou blocos lógicos alocados em forma de uma matriz” (ORDONEZ et al., 2003, p. 05). Os blocos lógicos são nomeados pelos seus fabricantes, podendo até existir mais de um nome para um mesmo fabricante. Este trabalho utiliza os conceitos do fabricante *Xilinx*.

Apesar de existir uma grande variação na estrutura de um FPGA, três elementos fundamentais são mantidos, de acordo com Ordonez et al. (2003, p. 06):

- CLB: *Configurable Logic Block* ou Bloco Lógico Configurável é a unidade lógica de um FPGA, podendo possuir recursos como *flip-flops* ou registradores;
- IOB: *In/Out Block* ou Bloco de Entrada e Saída é responsável pela interface com o ambiente e;
- SB: *Swith Box* ou Caixa de Conexão (chamada também de *Switch Matrix* – Matriz de Conexão) é responsável pela interconexão entre os CLBs, permitindo o seu roteamento.

A Figura 15 mostra a estrutura de um FPGA.

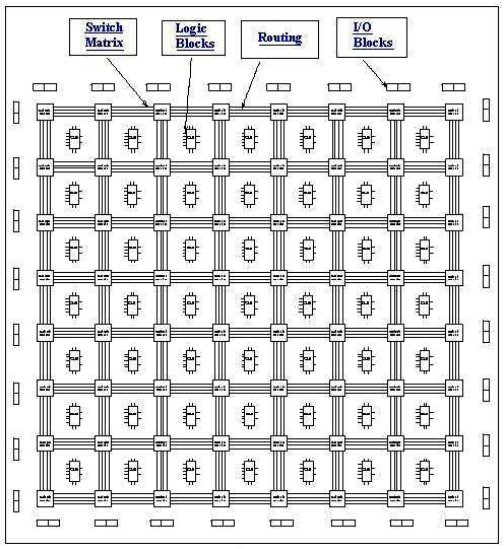

Figura 15 - Estrutura de um FPGA (TERROSO, 1998, p. 04).

Ordonez et al. (2003, p. 07) ressaltam que “nos últimos anos, a quantidade de portas lógicas disponíveis num FPGA tem crescido num ritmo muitíssimo acelerado, possibilitando

a implementação de arquiteturas cada vez mais complexas". Os FPGAs também permitem uma rápida prototipagem, reconfigurabilidade e baixo custo de desenvolvimento, além de permitir a utilização de linguagens de alto nível para a descrição do hardware, como é o caso do VHDL.

Essa tecnologia permite uma enorme variedade de aplicações, como por exemplo: decodificador de áudio digital, equipamentos médicos, robótica, roteadores, vídeo conferência, sistemas de computadores, microprocessadores, processadores embarcados, multimídia, impressoras, scanners etc.

Outro ponto importante a ser mencionado é a possibilidade de se mensurar o desempenho dos circuitos digitais implementados em FPGA. Dentre os diversos parâmetros existentes para tal mensuração, destacam-se dois:

- Ocupação espacial: determina a quantidade de componentes necessários para implementar o circuito e;
- Desempenho temporal: determina o tempo de atraso do sinal através do circuito.

É desejável que um circuito digital seja rápido e ocupe pouco espaço no FPGA. Contudo, satisfazer esses dois critérios é uma tarefa muito difícil e nem sempre é possível. Para auxiliar o projetista no cumprimento dessa meta, existem algumas ferramentas de síntese e vários algoritmos de otimização de circuitos específicos. Esses recursos têm a capacidade de tornar os circuitos digitais mais confiáveis e otimizados.

Para se implementar um chip em FPGA o sistema deve ser descrito através de uma linguagem de descrição de hardware. Esta descrição é um modelo do sistema de hardware e determina o quê o hardware deve fazer e como deve se comportar. Como mencionado anteriormente, neste projeto utilizou-se a linguagem VHDL.

4.2.2 Linguagem VHDL

O VHDL é um padrão para descrição de hardware adotado pela tecnologia FPGA. A letra V vem do inglês *"Very High Speed Integrated Circuit"*, que se traduz por "Circuito Integrado de Alta Velocidade" e as letras HDL significam *Hardware Description Language*, ou seja, Linguagem de Descrição de Hardware. Segundo Terroso (1998, p. 08), o "VHDL é uma forma de se descrever, através de um programa, o comportamento de um circuito ou componente digital".

Ordonez et al. vão mais além, colocando que o “VHDL é uma linguagem padronizada para descrever componentes digitais, permitindo a transferência de componentes ou projetos para qualquer tecnologia em construção de hardware existente ou que ainda será desenvolvida” (ORDONEZ et al., 2003, p. 10). É uma linguagem flexível e de altíssima velocidade, muito utilizada na indústria e, também, nas universidades para fins acadêmicos. Oferece, também, uma rica variedade de construções que permitem modelar o hardware em um elevado nível de abstração (ORDONEZ et al., 2003, p. 11).

Na sua origem, o VHDL visava simulação, modelagem e documentação, passando mais tarde a adquirir a possibilidade de síntese, com o objetivo de se automatizar o projeto de circuitos. Foi desenvolvida pelo Departamento de Defesa Americano (DoD) em 1983, pretendendo-se desenvolver uma metodologia comum de desenvolvimento de circuitos e que pudesse ser reutilizável em novas tecnologias. Em 1987 o *Institute of Eletrical and Electronics Engineers* (IEEE) ratificou o VHDL como IEEE Padrão 1076, assegurando o sucesso da linguagem.

Ordonez et al. (2003, p. 11) salientam a importância de uma linguagem de descrição de hardware em diversos aspectos do projeto, como na documentação do sistema (a própria descrição do sistema já é uma forma de documentação); simulação em diversos níveis (o circuito pode ser simulado a qualquer momento); simplicidade de migração tecnológica (facilidade de re-sintetização em outras tecnologias); e reutilização de recursos (construção de bibliotecas ou módulos).

Como esses circuitos são reprogramáveis, eles são muito utilizados em protótipos, gerando uma economia de tempo e dinheiro. É o que afirma o engenheiro Terroso (1998, p. 03), colocando que “a primeira utilização destes circuitos é naturalmente nos projetos de prototipagem. Tendo em vista que grande parte destes circuitos podem ser reprogramados, o seu uso nas fases preliminares de projeto possibilita uma grande economia de tempo e dinheiro” [sic].

Com relação à linguagem VHDL algumas vantagens são encontradas, como redução do tempo e do custo de desenvolvimento, maior nível de abstração, projetos independentes da tecnologia, facilidade de atualização dos projetos, verificação (através de simulação) do comportamento do futuro sistema digital entre outras (ORDONEZ et al., 2003, p. 12-13). Em contra partida, no início do projeto devem ser levados em conta algumas desvantagens do VHDL, como o fato do hardware gerado ser menos otimizado (TERROSO, 1998, p. 10), a dificuldade em representar um comportamento desejado e o fato de muitas funcionalidades não serem sintetizáveis.

4.3 Implementação do Algoritmo LSB

Em se tratando de esteganografia em imagens, a técnica mais comum para embutir dados dentro de um objeto de cobertura é a inserção no bit menos significativo (*Least Significant Bit* - LSB) (JOHNSON; JAJODIA, 1998, p.28). Wayner (2002) e Petitcolas; Anderson; Kuhn (1999) apud Julio; Brasil; Neves (2007, p. 59) ainda acrescentam que "o método de inserção no bit menos significativo é provavelmente uma das melhores técnicas de esteganografia em imagem".

Na sua forma mais básica os dados são escondidos no plano LSB inteiro, porém é possível selecionar os locais de inserção com esquemas mais sofisticados (JULIO; BRASIL; NEVES, 2007, p. 60). Porém, Johnson e Jajodia observam que esse tipo de técnica é vulnerável quando se trabalha com compressão. Quando se converte, por exemplo, uma imagem do formato GIF ou BMP para um formato JPEG, pode-se perder toda a informação embutida na imagem (JOHNSON; JAJODIA, 1998, p. 28).

A partir da Figura 16 é possível fazer um comparativo entre duas imagens, sendo uma sem informação embutida (imagem da esquerda) e outra após a inserção da informação (imagem da direita). A comparação de ambas as imagens ilustra a eficácia da técnica LSB, demonstrando que é impossível, a olho nu, notar alguma diferença entre elas.

Figura 16 - Comparação Entre Imagem de Cobertura e Estego-imagem (ROCHA; COSTA; CHAVES, 2004, p. 09).

No desenvolvimento deste trabalho foi descrito o algoritmo LSB visando a manipulação de imagens no formato BMP codificadas em 24 bits (3 bytes), onde cada byte representa uma cor do padrão RGB. Com uma imagem desse tipo é possível armazenar 3 bits de informação em cada pixel (um bit em cada byte), portanto uma imagem de 1024 x 768 pixels (2.359.296 bytes ou 2.304 Kbytes de tamanho), por exemplo, tem potencial para

esconder um total de 2.359.296 bits (294.912 bytes) de informação. Caso os dados sejam comprimidos antes de serem embutidos é possível esconder uma quantidade maior de informação, e, ainda assim, a estego-imagem resultante parecerá idêntica à imagem de cobertura (JOHNSON; JAJODIA, 1998, p. 28).

Por exemplo, se quisermos esconder a letra A, cujo valor binário é 10000011, precisaríamos utilizar 3 pixels da imagem. O dado *raster* original para 3 pixels poderia ser:

(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)

Inserindo o valor binário de A nos 3 pixels teríamos como resultado a seguinte porção de imagem:

(00100111 11101000 11001000)
(00100110 11001000 11101000)
(11001001 00100111 11101001)

É possível verificar que dos 8 bytes usados apenas os 4 bits destacados em vermelho foram realmente modificados, sendo que os outros bits não sofreram modificações. A informação do exemplo anterior foi inserida somente nos bits menos significativos de cada byte (são os bits dentro dos quadrados azuis), contudo ainda que fosse utilizado o segundo bit menos significativo, o olho humano não seria capaz de discernir qualquer diferença entre a imagem de cobertura e a estego-imagem (JOHNSON; JAJODIA, 1998, p. 28-29).

Isto é possível porque cada canal de cor consegue expressar valores de 0 a 255, ou seja, 256 tons diferentes de cores ($2^8 = 256$). Dessa forma, um pixel pode ter 256 valores de vermelho, 256 valores de verde e 256 valores de azul, gerando uma combinação de 16.777.216 (256 x 256 x 256 ou 2^{24}) tonalidades distintas de cores possíveis.

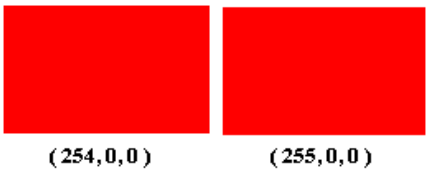

Figura 17 - Comparação Entre Dois Tons de Vermelho (Fonte Própria).

Esta é uma característica interessante para ser aproveitada em um algoritmo de esteganografia, pois se o bit menos significativo de cada byte for alterado para embutir uma informação, o valor da cor é alterado apenas em 1/256, resultando em uma diferença muito

insignificante, que não pode sequer ser detectada a olho nu. Conforme exemplificado na Figura 17, a cor da esquerda com os valores 254, 0 e 0 para vermelho, verde e azul, respectivamente; e a outra (cor da direita) com os valores 255, 0 e 0 para as mesmas cores são praticamente iguais, sendo quase impossível notar alguma diferença.

4.4 Trabalhos Correlatos

Todos os trabalhos sobre esteganografia encontrados foram desenvolvidos em software, reforçando a importância de um projeto focado em uma implementação por hardware, almejando-se um diferencial na linha de pesquisa e no desenvolvimento do projeto.

Um dos softwares de esteganografia em imagem encontrado com mais facilidade talvez seja o Camaleão (vide Figura 18). Seu objetivo principal, segundo os autores, seria "desenvolver um produto de software capaz de permitir a comunicação segura pela internet por fazer uso de técnicas esteganográficas em imagens digitais" (ROCHA; COSTA; CHAVES, 2004, p. 01). O produto proporciona, por ser desenvolvido em Java, um ambiente multiplataforma, tendo sido testado nos sistemas operacionais Linux, Windows 9x, Windows XP e Mac OS X; um ambiente bilíngüe, podendo ser usado em português ou inglês; código aberto sob licença de uso GPL, ou seja, o software pode ser alterado e utilizado livremente desde que se mantenham as referências aos autores; tipos de mascaramento e recuperação de qualquer arquivo binário dentro de outras imagens de extensão JPG ou PNG, sendo que a imagem de saída será de extensão PNG; e robustez, pois o sistema permite geração de chaves de deslocamento configuráveis (ROCHA; COSTA; CHAVES, 2004, p. 07).

Figura 18 - Tela Inicial do Sistema Camaleão (ROCHA; COSTA; CHAVES, 2004, p. 08).

Outro trabalho encontrado foi o intitulado "Protótipo de Software para Inserção e Extração de Mensagens em Arquivos *Raster* Através de Esteganografia". O objetivo deste trabalho é implementar um protótipo de software que permita inserir uma mensagem em um

arquivo *raster* utilizando-se da técnica de esteganografia LSB *insertion* (ZANELLA, 2002, p. 02). O algoritmo usado altera o bit menos significativo de cada byte do pixel, sendo que os pixels a serem alterados são selecionados randomicamente (na verdade os números são gerados de forma pré-definida), para que mensagem embutida fique dispersa na imagem. Foi desenvolvido um protótipo de software no ambiente de programação Visual Basic 6.0 (vide Figura 19), que recebe como parâmetros de entrada uma imagem e uma senha para efetuar o processo de inserção da mensagem, sendo que a mesma senha é necessária para a realização do processo inverso, ou seja, a recuperação de tal mensagem. As imagens de entrada podem ser de extensão GIF, JPEG ou BMP, porém gerando sempre estego-imagens no formato BMP.

Figura 19 - Tela Inicial do Protótipo do Sistema de Zanella (ZANELLA, 2002, p. 54).

Outra implementação em linguagem Java encontra-se no trabalho de título “Esteganografia – Implementação de um Software para Ocultar Mensagens Criptografadas em Imagens”, utilizando as técnicas de criptografia e esteganografia juntas. O software foi modelado com UML, sendo que o trabalho apresenta diagrama de caso de uso, diagrama de classes, diagrama de componentes e um estudo de caso. O usuário deve digitar uma mensagem, escolher uma imagem de formato PNG e uma senha para o processo de ocultar a mensagem; no processo inverso, ou seja, recuperação de mensagem, o usuário deve selecionar a imagem e digitar a mesma senha. A Figura 20 mostra uma tela deste software.

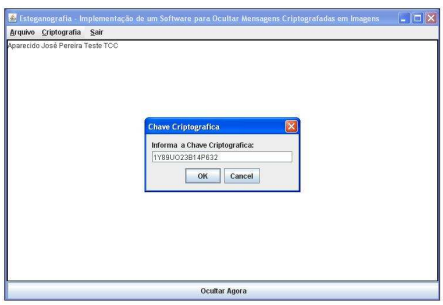

Figura 20 - Tela do Software em Funcionamento (PEREIRA, 2009, p. 35).

Julio, Brasil e Neves (2007, p. 84-96) elencam vários aplicativos de esteganografia, dentre eles estão:

- Ezestego: desenvolvido em Java e limitada a imagens indexadas de 8 bits no formato GIF. Pode ser executado em Unix, Linux, Windows e MAC/OS;
- Revelation: desenvolvido em Java e esconde arquivos em imagens de cobertura do tipo bitmap de 24 bits, podendo, também, ser executado em Unix, Linux, Windows e MAC/OS;
- Hide and Seek: insere uma lista de arquivos em uma imagem no formato JPEG. Seu ponto fraco é não utilizar criptografia;
- Outgess: propõe-se a melhorar o passo da codificação da imagem JPEG através de um gerador pseudo-randômico de números. O bit menos significativo dos coeficientes da transformada de cosseno selecionados é substituído pela mensagem cifrada;
- SignIt: esconde números de série em imagens de vários formatos. O número é escondido em todos os lugares da imagem, sendo impossível removê-lo sem alterar a imagem em um modo visível. Este software pode ser utilizado para proteger os direitos autorais de imagens e localizar cópias ilegais. Além disso, é possível verificar se uma imagem está protegida por alguma marca;
- GWatermarker: escrito em Virtual C++.NET, o software insere marca d'água de forma visível ou invisível a olho nu de forma robusta. Utiliza algoritmos próprios para inserção e remoção das marcas, algoritmo RC4 para inserção da chave secreta (com 6 a 56 caracteres) e algoritmo *hash* MD5.

4.5 Arquitetura do EstegoChip

O EstegoChip possui uma arquitetura própria com duas entradas e uma saída. A primeira entrada (denominada DADOS) é a informação do usuário que será embutida em uma porção de imagem e deve ter um byte de tamanho; a segunda entrada (denominada MODO) tem um bit de tamanho e informa a operação a ser realizada, ou seja, indica se é um processo de inserção dos dados (esteganografia) ou se é o processo inverso – a recuperação dos dados. A SAIDA exibe para o usuário a informação embutida na estego-imagem.

O EstegoChip é composto por seis módulos: ROM, RAM, INSERE DADOS, RECUPERA DADOS, UNIDADE DE CONTROLE e TOP, conforme Figura 21. A seguir é descrito a função de cada um destes componentes:

- ROM: memória somente de leitura utilizada para armazenar a imagem de cobertura, ou melhor, o trecho desta imagem.
- RAM: memória de leitura e escrita utilizada para armazenar a estego-imagem, ou seja, o trecho de imagem após a inserção dos dados.
- INSERE DADOS: módulo de inserção de dados, responsável pela execução do algoritmo LSB. Tem como parâmetros de entrada um bit (referente à informação do usuário a ser embutida) e mais um byte (referente à imagem de cobertura). Este componente envia para a saída o byte da imagem com o seu bit menos significativo alterado, contendo o bit da mensagem do usuário.
- RECUPERA DADOS: módulo responsável pela recuperação dos dados. Tem como parâmetro de entrada um byte (da estego-imagem) e envia para a saída somente o bit menos significativo.
- UNIDADE DE CONTROLE: utiliza uma máquina de estados para controlar os quatro primeiros módulos (descritos acima), sincronizando os dados de entrada e saída, fazendo com que eles consigam se comunicar corretamente.
- TOP: módulo responsável pela interligação de todos os outros módulos, relacionando todas as saídas e entradas.

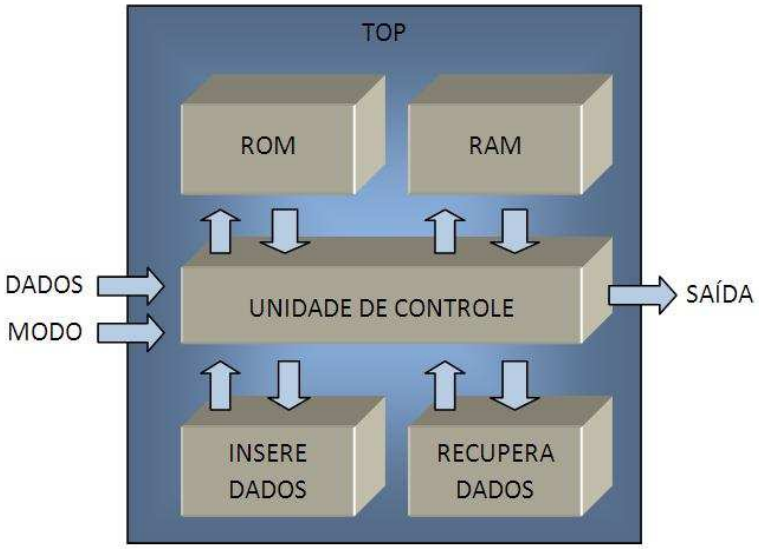

Figura 21- Diagrama da Arquitetura Geral do EstegoChip (Fonte Própria).

O funcionamento do processo de ocultação de dados segue os seguintes passos:

1. Entrada MODO recebe valor ‘1’ e entrada DADOS recebe a informação que será embutida na imagem de cobertura;
2. A UNIDADE DE CONTROLE separa um bit da informação, busca um byte da imagem de cobertura (na memória ROM) e envia estes dois parâmetros para o módulo INSERE DADOS;
3. O módulo INSERE DADOS executa o algoritmo LSB e envia um byte da estego-imagem para a UNIDADE DE CONTROLE;
4. A UNIDADE DE CONTROLE grava os dados na memória RAM;
5. Os passos 2, 3 e 4 são repetidos até que toda a informação esteja oculta gerando a estego-imagem.

Neste processo, nenhuma saída é exibida.

O processo inverso, ou seja, a recuperação dos dados funciona da seguinte maneira:

1. Entrada MODO recebe o valor ‘0’;
2. A UNIDADE DE CONTROLE busca um byte da estego-imagem (na memória RAM) e envia-o para o módulo RECUPERA DADOS;
3. O módulo RECUPERA DADOS extrai o bit menos significativo do byte e envia-o para a UNIDADE DE CONTROLE;
4. Os passos 2 e 3 são repetidos até que toda a informação seja recuperada;
5. A UNIDADE DE CONTROLE une todos os bits e envia-os para a saída.

Neste processo, não é necessária a utilização da entrada DADOS.

CAPÍTULO V – ANÁLISE DE RESULTADOS

Este capítulo mostra o cenário de teste e validação, a taxa de ocupação física do FPGA e o desempenho alcançado. Ele é importante porque exibe os resultados obtidos com o EstegoChip.

5.1 Cenário de Teste e Validação

Para testar e validar os módulos do EstegoChip foi utilizado a ferramenta de simulação ISE 9.1i da fabricante *Xilinx*. Primeiramente foram testados todos os módulos separadamente para verificação dos resultados parciais. Em seguida foi testado o módulo top – responsável pela interligação de todos os módulos do EstegoChip – para validar o funcionamento geral do *core* de segurança.

A Figura 22 ilustra a simulação do módulo top, onde o EstegoChip recebe um valor de entrada (8'h5D) e após 10100 ns este mesmo valor é mostrado na saída. Na verdade, o valor de entrada e saída é '5D' (01011101 em binário ou 93 em decimal), sendo que o '8' indica um valor de 8 bits e o 'h' indica que o valor está representado no sistema hexadecimal.

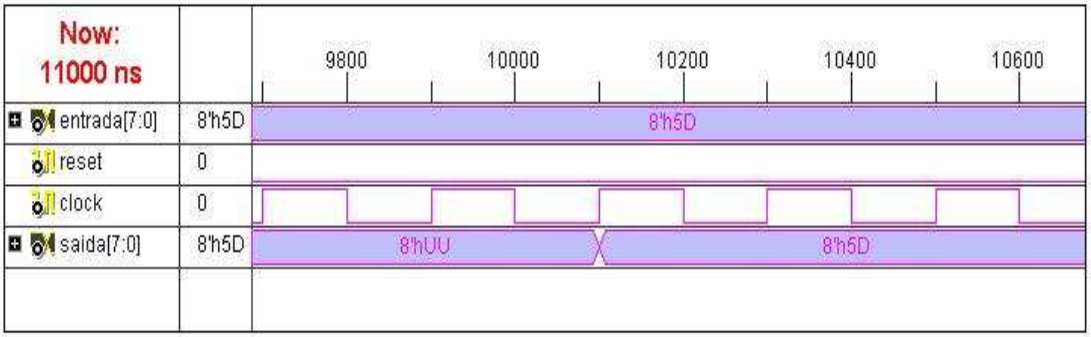

Figura 22 - Simulação e Validação do EstegoChip (Fonte Própria).

O valor de entrada '5D' é, na realidade, a informação do usuário que será embutida na imagem de cobertura. Esta simulação já contém os processos de inserção e recuperação de dados, pois não seria possível realizar uma simulação para inserir dados e depois utilizar o resultado em outra simulação de recuperação destes dados. Desta forma, esta simulação valida todo o *core* de segurança proposto, demonstrando o seu correto funcionamento.

5.2 Taxa de Ocupação

Uma das principais métricas para análise de um FPGA é a taxa de ocupação. Ela refere-se à ocupação física do FPGA em relação à área total disponível. Em se tratando de ocupação física os principais dados analisados são as quantidades de *Flip-Flops*[4], LUTs[5] e *Slices*[6] consumidos. A ocupação do EstegoChip pode ser observado na Tabela 1, onde é exibida a quantidade de componentes utilizados e a porcentagem ocupada da área total disponível, comprovando uma baixa utilização de recursos.

Tabela 1 - Taxa de Ocupação do EstegoChip.

COMPONENTES	*FLIP-FLOPs*	LUTs	*SLICEs*
QUANTIDADE	68	158	100
PORCENTAGEM	1%	3%	4%

Fonte Própria.

5.3 Taxa de Propagação

A taxa de propagação é uma das mais importantes métricas para avaliar o desempenho de um circuito. As medidas utilizadas são o tempo de propagação e frequência máxima do circuito.

Tabela 2 - Estimativas de Tempos de Propagação.

TAMANHO DA INFORMAÇÃO	1 byte	100 bytes	500 bytes	1 Kbyte	1 Gbyte
TEMPO DE PROPAGAÇÃO	558,96 ns	55.896 ns	279.480 ns	≈ 572.375 ns	≈ 0,586 seg

Fonte Própria.

O EstegoChip, de acordo com a especificação de sua arquitetura, requer uma entrada de dados limitada em 8 bits (1 byte), sendo que o tempo de propagação de cada *clock* é de

[4] Memória capaz de armazenar somente um bit.
[5] Look-Up Table: bloco lógico de armazenamento utilizado para implementar pequenas funções lógicas.
[6] Menor unidade lógica configurável. Pode ser configurado para operar como LUT.

10,96 ns, operando em uma frequência de 91,241 MHz. Sabendo que o EstegoChip necessita de 51 *clock's* para executar os processos de inserção e recuperação de uma informação com tamanho de 1 byte, deve-se multiplicar 10,96 por 51 para se chegar ao tempo total de execução dos referidos processos. A partir dessas informações é possível criar novos dados com entradas de diferentes tamanhos, gerando estimativas de desempenho para estes novos valores. A Tabela 2 mostra um comparativo entre algumas entradas de diferentes tamanhos e suas respectivas estimativas de tempos de propagação. É possível observar que para uma informação de 1 Kbyte, o processo de inserção e recuperação da informação seria realizado em uma fração de segundos (aproximadamente 572.375 nanosegundos) e mesmo uma informação de 1 Gbyte exigiria um tempo de processamento muito pequeno (aproximadamente 0,586 segundos).

CONCLUSÃO E TRABALHOS FUTUROS

Partindo do princípio de que o processo de informatização e o aumento do uso das redes de computadores tornaram as informações (dados digitais) muito mais vulneráveis, foi proposto e implementado, em hardware, um *core* de segurança capaz de embutir uma informação dentro de um trecho de imagem. Dessa forma, vislumbra-se que uma informação sigilosa, juntamente com o uso da criptografia convencional, seja ocultada em um arquivo inócuo para trafegar pela rede de forma despercebida.

Este projeto foi embasado em uma pesquisa bibliográfica, onde foram levantados conceitos, técnicas, aplicações e trabalhos correlatos, dentre outros assuntos pertinentes ao tema. Esta pesquisa gerou a primeira dificuldade do projeto, pois foram poucos os trabalhos de especialistas encontrados na literatura, principalmente na área de hardware. Entretanto, de acordo com este estudo verificou-se que o algoritmo mais comum para este tipo de aplicação é a inserção de dados nos bits menos significativos (LSB) de cada byte dos pixels. Este algoritmo foi implementado utilizando-se um trecho de imagem pré-definido na memória ROM do *core* de segurança, gerando um trecho de estego-imagem na sua memória RAM.

Uma arquitetura específica para este *core* foi criada e validada através de testes e simulações realizados com o software ISE 9.1i do fabricante *Xilinx*, comprovando o seu correto funcionamento. Através desta mesma ferramenta de desenvolvimento foram gerados dados referentes à ocupação física e de desempenho do hardware, fornecendo métricas importantes para a análise de seus resultados.

Uma dificuldade encontrada na parte de desenvolvimento foi a utilização de um arquivo de imagem como parâmetro de entrada para o EstegoChip, gerando em seguida outro arquivo de imagem, a estego-imagem. Isto poderia ter sido programado e simulado, entretanto não seria possível a sintetização deste hardware, visto que uma desvantagem da linguagem VHDL é a sua limitação com relação à sintetização de alguns comandos.

Para um trabalho futuro, poderia ser utilizada uma senha no processo de esteganografia, sendo esta necessária no processo de recuperação dos dados embutidos. Estes dados poderiam ser comprimidos para diminuir a quantidade de bits a serem embutidos, obtendo, assim, um ganho na capacidade de armazenamento. E, por fim, os dados poderiam ser cifrados antes de serem embutidos na imagem, visto que a esteganografia e a criptografia funcionam muito bem juntas, uma complementando a outra.

REFERÊNCIAS

AGOSTINI, L. V. **Estudo de Padrões de Compressão de Imagens para Aplicações VLSI.** Porto Alegre, 1999. Disponível em: <http://www.inf.ufrgs.br/~agostini/TI_Image.pdf >. Acesso em: 11 ago. 2002.

AZEVEDO, E; CONCI, A. **Computação Gráfica – Teoria e Prática**. Rio de Janeiro. Elsevier, 2003.

COUTINHO, P. S. **Esteganografia**. Universidade Federal do Rio de Janeiro, atualizado em 06/06/2008. Disponível em: <http://www.gta.ufrj.br/grad/08_1/estegano/EsteganografiaeCriptografia.html>. Acesso em: 08 maio 2010.

FERREIRA, F. N. F. **Segurança da Informação**. Rio de Janeiro. Editora Ciência Moderna Ltda, 2003.

FILHO, A. G. **Física e Realidade**. São Paulo: Sciplione, 1997.

FILHO, O. M.; NETO, H. V. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999.

FILHO de L. et al. ***Electrocardiographic Signal Compression Using Multiscale Recurrent Patterns***. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, v. 52, n. 12, p. 2739–2753, 2005.

GONZALEZ, R. C.; WOODS, R. E. **Processamento de Imagens Diditais**. Tradução: Roberto Marcondes Cesar Junior, Luciano da Fontoura Costa – São Paulo: Edgard Blücher, 2000.

GRUPO DE MICROELETRÔNICA. **Tutorial VHDL**. Universidade Federal de Itajubá, [s.d]. Disponível em: <http://www.microeletronica.unifei.edu.br/ELT502/VHDL-color.pdf>. Acesso em: 26 fev. 2010.

HOLMES, D. P. ***Introduction to Digital Image Steganography***, [S.1.], 2002. Disponível em: <http://www.giac.org/practical/David_P_Holmes_GSEC.doc>. Acesso em: 11 out. 2002.

JOHNSON, N. F.; JAJODIA, S. ***Exploring Steganography: Seeing the Unseen***. *George Mason University. Computing Practices IEEE*, 1998. p. 26-34. Disponível em:

<http://scholar.google.com.br/scholar?hl=pt-BR&q=Exploring+steganography%3A+Seeing+the+unseen&lr=&as_ylo=&as_vis=0>. Acesso em: 24 fev. 2010.

JULIO, E. P.; BRASIL, W. G.; NEVES, C. V. **Esteganografia e suas Aplicações**. In: VII SIMPÓSIO BRASILEIRO EM SEGURANÇA DA INFORMAÇÃO E DE SISTEMAS COMPUTACIONAIS. Universidade Federal Fluminense. Departamento de Ciência da Computação – Centro Tecnológico, 2007. p. 54-102. Disponível em: <http://www.dcc.ic.uff.br/~celio/papers/minicurso-sbseg07.pdf>. Acesso em: 23 fev. 2010.

LIN, E. T.; DELP, E. J. ***A Review of Data Hiding in Digital Images***, Indiana, [2002?]. Disponível em: <http://debut.cis.nctu.edu.tw/~yklee/Research/Steganography/Edward_J_Delp/PICS99.pdf>. Acesso em: 14 out. 2002.

LOPES, J. M. B. **Computação Gráfica: Formatos de Imagem**. Portugal, 2002. Disponível em: <http://mega.ist.utl.pt/~ic-cg/programa/livro/FormatosdeImagem.pdf>. Acesso em: 15 nov. 2002.

MURRAY, J. E. **Formatos de Arquivos Gráficos**, [S.l.], 1994. Disponível em: <http://www.di.ufpe.br/~if291/documentos/formatos_graficos/formatos.htm>. Acesso em: 15 set. 2002.

ORDONEZ, E. D. M. et al. **Projeto, Desempenho e Aplicações de Sistemas Digitais em Circuitos Programáveis (FPGAs)**. Pompéia: Bless, 2003.

PEREIRA, A. J. **Esteganografia - Implementação de um Software para Ocultar Mensagens Criptografadas em Imagens**. UNIVEM – Centro Universitário Eurípides de Marília, Fundação de Ensino “Eurípides Soares da Rocha”, 2009.

PETITCOLAS, F. A. P.; ANDERSON R. J.; KUHN, M. G. ***Information Hiding - A Survey***. *Proceedings of the IEEE*, v. 87, n. 7, p. 1062-1078, 1999. Disponível em: <http://scholar.google.com.br/scholar?hl=pt-BR&q=PETITCOLAS&lr=&as_ylo=&as_vis=0>. Acesso em: 24 fev. 2010.

POPA, R. ***An Analysis of Steganography Techniques***. Dissertação (Mestrado) - *The Polytechnic University of Timisoara*, Timisoara, Romênia, 1998.

ROCHA, A. R.; COSTA, H. A. X.; CHAVES, L. M. **Camaleão: um Software para Segurança Digital Utilizando Esteganografia**. Instituto de Computação - Universidade Estadual de Campinas (Unicamp), (2004). Disponível em:

<http://www.liv.ic.unicamp.br/~undersun/pub/papers/Camaleao_um_Software_para_Seguranca_Digital_Utilizando_Esteganografia.pdf>. Acesso em: 16 nov. 2009.

SELLARS, D. ***An Introduction to Steganography***, [S.1.], [1999]. Disponível em: <http://www.cs.uct.ac.za/courses/CS400W/NIS/papers99/dsellars/stego.html>. Acesso em: 23 jun. 2002.

SPARTAN2E, 2010. Disponível em: <http://www.ece.unm.edu/xup/spartan2emicroblaze.htm>. Acesso em: 08 out. 2010.

STALLINGS, W. **Criptografia e Segurança de Redes: Princípios e Práticas**. Tradução: Daniel Vieira; Revisão Técnica: Graça Bressan, Ákio Barbosa e Marcelo Succi. 4ª Ed. São Paulo. Pearson Prentice Hall, 2008.

TERADA, R. **Segurança de Dados – Criptografia em Redes de Computadores**. Editora Edgard Blücher Ltda, 2008. 2ª Edição. São Paulo.

TERROSO, A. R. **Dispositivo Lógico Programável (FPGA) e Linguagem de Descrição de Hardware (VHDL)**. In: VII SEMANA DA ENGENHARIA - PUCRS. Mini curso II. Porto Alegre, setembro de 1998. Disponível em: <http://www.ee.pucrs.br/~terroso/fpgavhdl6.pdf>. Acesso em: 26 fev. 2010.

TKOTZ, V. **As Funções *Hash***, 2009[a]. ALDEIA NUMABOA. Disponível em: <http://www.numaboa.com/criptografia/hashes/338-hash?showall=1>. Acesso em: 25 out. 2010.

TKOTZ, V. **O que é Esteganografia**, 2009[b]. ALDEIA NUMABOA. Disponível em: <http://www.numaboa.com/criptografia/esteganografia/614-esteganografia>. Acesso em: 02 mar. 2010.

WAYNER, P. ***Disappearing Cryptography: Information Hiding: Steganography and Watermarking*** *(2nd Edition). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc*., 2002. ISBN 1558607692.

ZANELLA, D. **Protótipo de Software para Inserção e Extração de Mensagens em Arquivos *Raster* Através de Esteganografia**. Universidade Regional de Blumenal, 2002. Disponível em: <campeche.inf.furb.br/tccs/2002-II/2002-2danielzanellavf.pdf>. Acesso em: 02 jul. 2010.