Universidade Federal Fluminense

Eduardo Pagani Julio

Uma Arquitetura de Sistemas de Detecção de Intrusão em Redes Ad Hoc Sem Fio usando Esteganografia e Mecanismos de Reputação

NITERÓI

2007

Universidade Federal Fluminense

Eduardo Pagani Julio

Uma Arquitetura de Sistemas de Detecção de Intrusão em Redes Ad Hoc Sem Fio usando Esteganografia e Mecanismos de Reputação

Dissertação de **Mestrado** *submetida* ao "Programa de Pós-Graduação em Computação" da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre. Área de concentração: Processamento Paralelo e Distribuído.

Orientador:
Prof. Célio V. N. Albuquerque, Ph.D.

NITERÓI

2007

Uma Arquitetura de Sistemas de Detecção de Intrusão em Redes Ad Hoc Sem Fio usando Esteganografia e Mecanismos de Reputação

Eduardo Pagani Julio

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre. Área de concentração: Processamento Paralelo e Distribuído.

Aprovada por:

Prof. Célio V. N. Albuquerque, Ph.D. / IC-UFF
(Orientador)

Prof. Julius C. B. Leite, Ph.D. / IC-UFF

Prof. Artur Ziviani, Dr. / LNCC

Niterói, 21 de Dezembro de 2007.

Se, a princípio, a idéia não é absurda, então não há esperança para ela.

Albert Einstein

À minha esposa Alessandreia e ao nosso futuro filho, ou filha, que foi adiado em função deste trabalho.

Agradecimentos

A meu pai Köber que não pode acompanhar a minha caminhada de perto, mas que é presença constante em minha vida.

À minha mãe Cilea pelo amor que me levou a perseguir a realização de meus sonhos.

Aos meus familiares pelo apoio e carinho.

À minha esposa e fiel companheira Alessandreia pelo carinho, compreensão, força e incentivo. Sem você não conseguiria chegar até aqui.

Ao Professor Célio por ter acreditado em minha capacidade e pelo prazer de sua orientação.

Aos professores do IC pelos ensinamentos.

Aos professores Julius Leite e Artur Ziviani por aceitarem o convite para participar da minha banca.

Aos meus amigos que também foram responsáveis por mais essa conquista, em especial ao amigo Edelberto.

A Deus que tem manifestado seu carinho em todos os momentos felizes e menos felizes de minha vida.

Resumo

Este trabalho apresenta uma arquitetura para um Sistema de Detecção de Intrusão (IDS) em redes sem fio *ad hoc* usando esteganografia para troca de mensagens e mecanismos de reputação. Esta solução elimina a exigência de um canal de comunicação seguro entre os nós, onde o uso de criptografia é custoso em se tratando de processamento e bateria escassas, além da necessidade de uma infra-estrutura de chave pública. A esteganografia é usada para transportar mensagens no padrão IDMEF (*Intrusion Detection Message Exchange Format*) trocadas entre os IDSs. Os mecanismos de reputação são usados para classificar as mensagens trocadas entre os IDSs formando uma opinião local sobre uma determinada evidência de invasão global. Este trabalho apresenta ainda os resultados sobre a identificação dos formatos e freqüência de alertas que são enviados pela arquitetura e das dimensões e formatos de imagens que podem servir como stego-imagem para o IDS. Além disso, este trabalho verifica a possível utilização do tráfego real de imagens de uma rede como meio de transporte dos alertas e expõe o estudo do consumo de energia da arquitetura proposta, fazendo uma comparação com uma arquitetura de infra-estrutura de chave pública para redes *ad hoc*.

Palavras-chave: Segurança, detecção de intrusão, esteganografia, redes *ad hoc*, reputação.

Abstract

This work presents an architecture for a Intrusion Detection System (IDS) in wireless ad hoc networks using reputation mechanisms and steganography for IDS alert exchange. This solution eliminates the requirement of a safe communication channel between nodes, where the use of criptography becomes expensive in regards to scarce processing and energy, in addition to the need for a public key infrastructure. Steganography is used to carry messages in IDMEF (*Intrusion Detection Message Exchange Format*) standard exchanged between the IDSs. The reputation mechanisms are used to classify the messages exchanged between the IDSs forming a local opinion on one determined evidence of global invasion. This work also presents the results on the identification of formats and frequency of alerts that are sent by the architecture and the size and the format of images that can serve as stego-image for the IDS. Furthermore, this work notes the possible use of traffic actual images of a network as a means of transporting the alerts and exposes the study of energy consumption of the proposed architecture, making a comparison with one architecture for public key infrastructure in ad hoc networks.

Keywords: Security, intrusion detection, steganography, *ad hoc* networks, reputation.

Abreviações

ADM	:	Módulo de Descoberta de Anomalia
AIFF	:	Audio Interchange File Format
AODV	:	Ad Hoc On-Demand Distance Vector Routing
BPP	:	Bits por Pixel
CA	:	Certification Authority
CBR	:	Constant Bit Rate
DCT	:	Discrete Cosine Transform
DICOM	:	Digital Imaging and Communications in Medicine
DoS	:	Deny of Service
DSDV	:	Destination Sequenced Distance Vector
DSR	:	Dynamic Source Routing
DSSS	:	Direct-Sequence Spread Spectrum
FHSS	:	Frequency Hopping Spread Spectrum
FRAC	:	Flow-based Route Access Control
GZip	:	GNU Zip
HIDS	:	Host-based Intrusion Detection System
IDCT	:	Inverse Discrete Cosine Transform
IDMEF	:	Intrusion Detection Message Exchange Format
IDRM	:	Intrusion Detection and Response Model
IDS	:	Intrusion Detection System
IDXP	:	Intrusion Detection Exchange Protocol
IETF	:	The Internet Engineering Task Force
IRM	:	Intrusion Response Model
LDA	:	Análise Discriminante Linear
LIDS	:	Local Intrusion Detection System
LSB	:	Least Significant Bit
MANET	:	Mobile Ad Hoc Network
MDCT	:	Modified Discrete Cosine Transform

MDM	:	Módulo de Descoberta por Assinatura
MIBs	:	Bases de Gerenciamento de Informações
NIDS	:	Network-based Intrusion Detection System
PKI	:	Public Key Infrastructure
PoVs	:	Pair of Values
QIM	:	Quantization Index Modulation
RLE	:	Run Length Encoding
SAH	:	Sistema Auditivo Humano
SNMP	:	Simple Network Management Protocol
SS	:	Spread Spectrum
SSD	:	Stationary Secure Database
TIARA	:	Techniques for Intrusion-Resistant Ad hoc Routing Algorithms
VANETs	:	Vehicular Ad hoc NETworks
WAV	:	Windows Audio-Visual
XML	:	Extensible Markup Language

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Devido à natureza cooperativa das redes sem fio *ad hoc*, diversas vulnerabilidades são expostas, tornando-as suscetíveis a várias ameaças, sejam estas passivas como a espionagem ou ativas como as utilizadas por atacantes que objetivam alterar ou interromper o encaminhamento de mensagens de roteamento, comprometendo o funcionamento geral de toda a rede.

Assim, diferentemente das redes cabeadas, onde os atacantes devem inicialmente obter o acesso físico à rede ou ultrapassar diversas linhas de defesa tais como filtros de roteadores, *proxies* e *firewalls*, numa rede *ad hoc*, o atacante pode lançar o seu ataque contra qualquer um dos nós independentemente da direção da mensagem. Isto se deve à natureza compartilhada do meio sem fio e à ausência de um ponto único de distribuição de tráfego ou *gateway*. Além disso, as redes *ad hoc* estão submetidas às mesmas vulnerabilidades observadas em redes cabeadas, onde são possíveis ataques como falsidade ideológica (*spoofing*), gravação e reprodução (*replay*), e negação de serviço (*denial of service*). Diante disso, surgem os sistemas de detecção de intrusão (IDSs) que têm por finalidade a detecção automatizada e a geração subseqüente de um alarme para alertar o aparato de segurança em um local. Porém este sistema sozinho não oferece nenhuma garantia de eliminar esta possibilidade.

Diversos sistemas de detecção de intrusão para redes *ad hoc* foram propostos em [Zhang et al. 2003, Okazaki et al. 2002, Kachirski e Guha 2002, Marti et al. 2000] [Ramanujan et al. 2003, Bhargava e Agrawal 2001], onde toda troca de mensagens entre IDSs na rede *ad hoc* é feita através de um canal seguro, normalmente usando criptografia. Mas a ausência de pontos de acesso centralizados e de mecanismos de autorização, como uma infraestrutura de chave pública ou autoridades certificadoras, torna difícil a sua utilização. Assim, é altamente desejável a troca de mensagens entre IDSs de maneira

simplificada, onde não seja necessário o uso de criptografia.

Além disso, as medidas de prevenção de intrusão com o uso de criptografia e autenticação somente podem prevenir ataques externos, mas pouco podem fazer quando nós internos são comprometidos. Diante desse cenário, mecanismos de incentivo a colaboração e de reputação podem auxiliar os IDSs. Em sistemas de detecção colaborativos, quando uma anomalia é detectada em um nó local, ou quando existe uma evidência não conclusiva, os vizinhos participam cooperativamente em ações de detecção de intrusão global. Nesse processo, pode-se introduzir um mecanismo de reputação, definido como uma medida de quão confiável e seguro um nó é considerado por outro, fundamentado em suas interações diretas e também na opinião de outros pares que interagiram com este. Assim, pode-se receber respostas de nós vizinhos sobre uma possível intrusão e, baseado nas reputações desses nós, formar a opinião local sobre essa. Com a combinação dessas técnicas, objetiva-se substituir um canal seguro por um transporte de mensagens onde não seja necessária criptografia.

O objetivo deste trabalho é verificar a viabilidade de utilização de um sistema de detecção de intrusão colaborativo em uma rede *ad hoc*, utilizando esteganografia e mecanismos de reputação, onde cada nó pode trabalhar cooperativamente informando a outros IDSs sobre evidências de ataques locais. Em um ambiente sem fio *ad hoc*, como o consumo de bateria é um fator crítico, este trabalho apresenta um estudo sobre o consumo de energia dos principais passos da arquitetura proposta, identificando em que tipo de ambiente ela pode ser empregada.

Este trabalho está organizado como segue. O Capítulo 2 descreve os sistemas de detecção de intrusão em redes *ad hoc*. O Capítulo 3 apresenta uma extensa revisão sobre esteganografia, método este que é utilizado no modelo proposto para transporte de alertas dos IDSs. O Capítulo 4 apresenta o modelo de arquitetura de IDS proposto por este trabalho e os mecanismos de esteganografia e reputação utilizados. No Capítulo 5, estão descritos os testes e as simulações realizados, os resultados obtidos e uma comparação com uma proposta de uso de criptografia em redes *ad hoc*. Finalmente no Capítulo 6 são apresentadas as considerações finais e algumas propostas para trabalhos futuros.

Capítulo 2

Sistemas de Detecção de Intrusão

A natureza de uma rede *ad hoc* a faz insegura. O grau de comprometimento entre seus membros é alto, já que todos dependem uns dos outros para o pleno funcionamento da rede. A qualidade conseguida depende do trabalho de cada nó. A partir desses comentários pode-se perceber que o mau funcionamento de um único nó pode trazer grande prejuízo para toda a rede. Os protocolos de roteamento desenvolvidos inicialmente não se preocuparam com os aspectos de segurança. Dessa maneira, as vulnerabilidades intrínsecas de uma rede *ad hoc* devido ao alto grau de dependência entre seus membros, tornaram-se falhas de segurança para os protocolos de roteamento.

Sendo assim, percebe-se que a forma como uma rede desse tipo deve ser protegida não é a mesma adotada em redes cabeadas. Cada um de seus membros deve estar preparado para enfrentar um adversário, garantindo indiretamente maior grau de segurança para toda a rede. Sabe-se que em redes de outros tipos, onde o meio físico é compartilhado, a segurança total da rede depende, também, das ações preventivas tomadas por cada membro. Porém em redes *ad hoc* essas ações têm um significado ainda mais forte.

Neste capítulo é apresentada uma visão dos Sistemas de Detecção de Intrusão, bem como o seu funcionamento e os requisitos para que sejam empregados em redes *ad hoc*.

2.1 Definição

A detecção de intrusão pode ser definida como a detecção automatizada e a geração subseqüente de um alarme para alertar o aparato de segurança em um local, se intrusões aconteceram ou estão acontecendo. Os IDSs são sistemas automáticos baseados em mecanismos de monitoramento em tempo real que observam o comportamento do tráfego de rede em ambientes operacionais, que em seu conjunto procuram identificar padrões

considerados hostis que possam comprometer a segurança do sistema.

Os principais benefícios do uso desta tecnologia estão na antecipação e acompanhamento dos ataques, bem como no auxílio à compreensão das estratégias utilizadas pelos atacantes. A parte de prevenção pode envolver alertas emissores como também a tomada de medidas preventivas diretas, como bloquear uma conexão suspeita. Em outras palavras, a detecção de intrusão é um processo de identificar e responder à atividade maliciosa disparada contra os computadores e os recursos da rede. Além disso, ferramentas IDS são capazes de distinguir entre ataques internos originados de nós dentro da rede e ataques externos. Diferentemente de *firewalls* que são a primeira linha de defesa, IDSs funcionam de forma reativa a uma intrusão quando um nó e/ou uma rede foram comprometidos. É por isso que eles podem ser considerados uma segunda linha de defesa.

Os modelos de reconhecimento de ameaças presentes nos IDSs atualmente estão distribuídos em três grandes grupos [Mishra et al. 2004]. O primeiro é baseado em assinaturas de ataque, ou seja, monitora o tráfego de rede coletado, buscando a ocorrência de determinadas seqüências presentes em um repositório especial. Esta seqüência é descrita por meio de uma linguagem específica que tem a função de caracterizar o tráfego hostil. A vantagem desta abordagem está em apresentar um nível baixo de falsos positivos, porém mostra-se ineficiente para detectar ataques baseados em novas técnicas ou abordagens cuja representação de suas características não está em seu repositório de assinaturas.

O segundo modelo é centrado no entendimento do comportamento do perfil padrão do nó, ou seja, utiliza técnicas que procuram identificar diferenças baseadas na comparação de padrões de tráfego considerados seguros, isto é, livres de qualquer traço de tentativas de intrusão com um padrão anômalo de rede ou comportamento atípico do nó.

Por fim o terceiro modelo é baseado em especificações e está orientado a acompanhar o fluxo de dados buscando características específicas do comportamento de uma aplicação ou protocolo. Esta abordagem apresenta o mais baixo nível de falsos positivos, porém sua aplicabilidade é restrita a alguns cenários, atualmente presente em aplicações militares e de operações globais de transações financeiras.

Os modelos atuais de IDSs são concebidos de maneira modular, onde podem ser associadas diversas funcionalidades que possibilitem a visão completa da segurança. Os IDSs podem ser entendidos como uma solução composta por uma coleção de módulos especializados que em seu conjunto ou individualmente, definem características como a capacidade de coletar e analisar dados, detectar e responder a uma intrusão.

2.2 Tipos de IDSs

Os IDSs também podem ser classificados, de acordo com a sua área de atuação, em IDSs baseados em rede (NIDS - *Network-based Intrusion Detection System*) e em IDSs baseados em *host* (HIDS - *Host-based Intrusion Detection System*).

2.2.1 IDS baseado em rede

Este tipo de IDS tem por objetivo detectar ataques pela análise dos pacotes que trafegam pela rede através de uma escuta em um segmento de rede. Com isso, um IDS tem a capacidade de monitorar o tráfego de todos os nós que estão conectados neste segmento, protegendo-os [Bace e Mell 2001].

IDSs baseados em rede geralmente consistem de um conjunto de sensores colocados em vários pontos da rede. Estas unidades monitoram o tráfego, realizando uma análise local do mesmo e relatando ataques a um console central de gerenciamento. Como os sensores são limitados a executarem somente o IDS, eles podem ser mais facilmente protegidos contra ataques [Bace e Mell 2001].

Este tipo de IDS garante que com poucos IDSs instalados, mas bem posicionados, pode-se monitorar uma grande rede. Geralmente é simples adicionar esse tipo de IDS a uma rede e são considerados bem seguros contra ataques. Porém apresentam algumas desvantagens, como a dificuldade em processar todos os pacotes em uma rede grande e sobrecarregada, logo, eles podem falhar no reconhecimento de um ataque lançado durante períodos de tráfego intenso; além de muitas das vantagens dos IDSs baseados em rede não se aplicarem às redes mais modernas baseadas em *switches*, onde somente seria possível o monitoramento em equipamentos com porta monitor para fins de gerência, onde esta faz uma cópia de todo o tráfego agregado de todas as portas da rede. Outra grande desvantagem do IDS baseado em rede é a de não poder analisar informações criptografadas [Bace e Mell 2001].

Um exemplo de colocação em uma rede de um IDS baseado em rede pode ser visto na Figura 2.1.

2.2.2 IDS baseado em *host*

Este tipo de IDS é instalado em um *host* que é alertado sobre ataques ocorridos contra a própria máquina. Este IDS avalia a segurança deste *host* com base em arquivos de *logs*

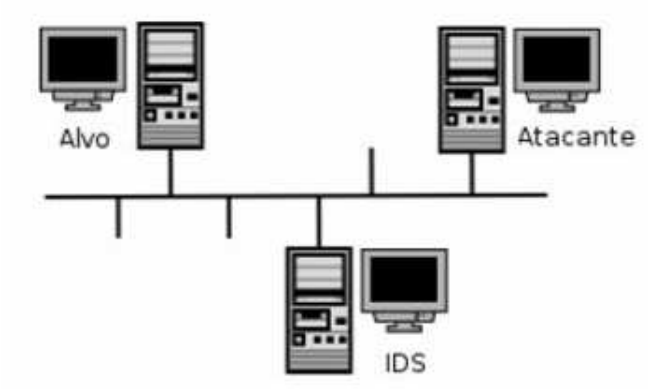


Figura 2.1: Exemplo de NIDS

de Sistema Operacional, de acesso e de aplicações. Tem grande importância, pois fornece segurança a tipos de ataques que o *firewall* e um IDS baseado em rede não detectam, como os baseados em protocolos de criptografia [Zhang et al. 2004]. Um exemplo de sua utilização pode ser visto na Figura 2.2, onde um IDS baseado em *host* avisa aos demais sistemas de IDSs dos outros nós sobre a presença de um intruso.

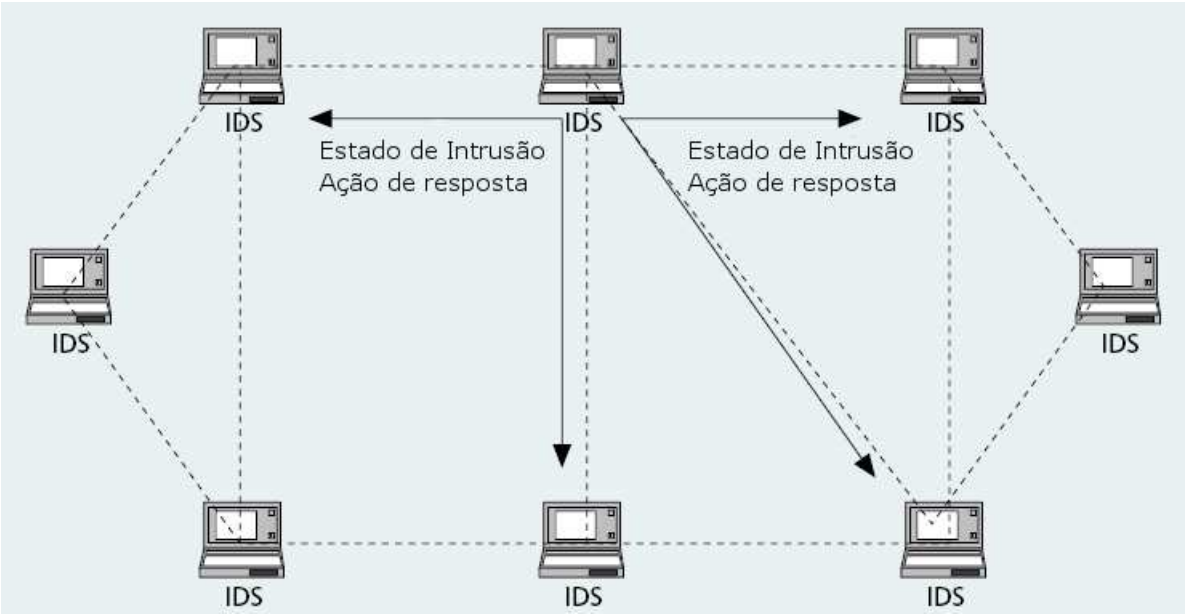


Figura 2.2: Exemplo de HIDS

O IDS baseado em *host* monitora as conexões de entrada no *host* e tenta determinar se alguma destas conexões pode ser uma ameaça. Monitora também arquivos, sistema de arquivos, *logs*, ou outras partes do *host* em particular, que podem ter atividades suspeitas representando uma tentativa de intrusão ou até mesmo uma invasão bem sucedida [Zhang et al. 2004].

Alguns IDS de *host* possuem a capacidade de interpretar a atividade da rede e detec-

tar ataques em todas as camadas do protocolo, aumentando assim a sua capacidade de bloqueio a determinados ataques que não seriam notados pelo *firewall* ou pelo IDS de rede, tais como pacotes criptografados. Esta análise é restrita a pacotes direcionados ao *host* protegido pelo IDS. Um exemplo de tentativa suspeita que é detectada pelo IDS baseado em *host* é o *login* sem sucesso em aplicações que utilizam autenticação de rede. Desta forma, o sistema IDS informa ao administrador de rede que existe um usuário tentando utilizar uma aplicação que ele não tem permissão [Zhang et al. 2004].

2.3 Trabalhos Relacionados: IDS para Redes *Ad Hoc*

Diversos trabalhos sobre detecção de intrusão para redes cabeadas tradicionais já foram realizados. Porém, aplicar esta pesquisa em redes sem fio não é uma tarefa automática por causa das diferenças-chave da arquitetura *ad hoc*, principalmente a falta de infraestrutura, que facilita a tarefa do atacante, sendo mais fácil espionar o tráfego de rede em um ambiente sem fio.

As redes sem fio *ad hoc*, devido a suas vulnerabilidades, fornecem um desafio maior para se projetar um IDS. Sem pontos de auditoria centralizados como roteadores e *gateways*, um IDS para redes *ad hoc* é limitado a usar somente o tráfego de entrada e saída do nó como dados de auditoria. Outro requisito chave é que os algoritmos que o IDS usa devem ser distribuídos por natureza, e deve-se levar em conta o fato que um nó pode ver somente uma porção do tráfego da rede.

Além disso, as redes *ad hoc* são dinâmicas e nós podem se mover livremente, existindo a possibilidade de um ou mais nós serem capturados e comprometidos, especialmente se a rede está em um ambiente hostil. Se os algoritmos do IDS são cooperativos, é importante saber em quais nós se pode confiar. Então, sistemas de detecção de intrusão em redes *ad hoc* devem ser cautelosos sobre ataques e ações maliciosas geradas por nós da própria rede. Por outro lado, nós em redes móveis não podem se comunicar tão freqüentemente quanto em redes cabeadas para descobrir intrusões a fim de preservar recursos de largura de banda. A largura de banda e outras questões como tempo de vida e consumo de bateria compõem um problema adicional. A disponibilidade de dados parciais de auditoria torna ainda mais difícil distinguir um ataque de um uso normal da rede.

A seguir são apresentados os mais recentes modelos conceituais de IDS para redes sem fio *ad hoc*. Por serem modelos conceituais sabe-se da não obrigatoriedade de sua implementação e pode-se dizer que nenhum dos presentes neste trabalho tem sua imple-

mentação conhecida.

2.3.1 IDS distribuído

O trabalho de [Zhang et al. 2003] introduz a utilização de IDS em redes móveis *ad hoc* (MANETs), descrevendo a detecção distribuída e cooperativa onde cada nó na rede participa na detecção da intrusão e na resposta. Neste modelo, um agente de IDS funciona em cada nó móvel, executa o levantamento de dados local e a detecção local, cooperando para a detecção e a resposta global à intrusão que podem ser provocados quando um nó relatar uma anomalia. Os autores consideram dois cenários separados de ataque: atualizações anormais das tabelas de roteamento e atividades anormais em outras camadas que não a de roteamento.

Os agentes de IDS são estruturados em seis partes, como mostrado na Figura 2.3. Cada nó faz a detecção local da intrusão independentemente, e os nós vizinhos trabalham colaborativamente em uma escala maior. Os agentes individuais de IDS colocados em cada nó monitoram as atividades locais (incluindo as do usuário, dos sistemas, e de comunicação provenientes do alcance do rádio), detectam as intrusões locais e iniciam as respostas.

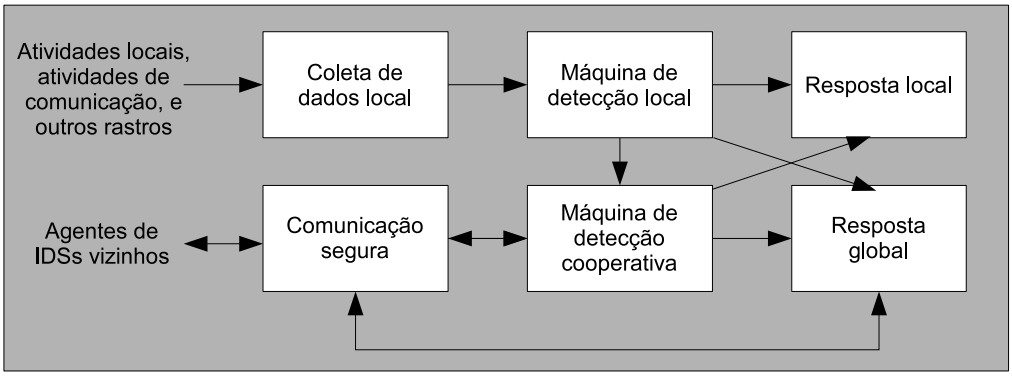


Figura 2.3: O IDS para MANETs [Mishra et al. 2004]

Os agentes de IDSs vizinhos participam cooperativamente na detecção de intrusão global quando uma anomalia é detectada em dados locais ou quando há uma evidência ainda conclusiva. Os recolhimentos de dados locais e registros de atividades pelo módulo de coleta de dados local são usados para detectar o local da anomalia. Métodos de detecção necessitam de uma série de dados ou colaboração entre os agentes. Os agentes de IDS usam uma detecção cooperativa. Os módulos fornecem uma resposta local e global como resposta à intrusão [Mishra et al. 2004].

O módulo de resposta local provoca ações locais ao nó móvel (por exemplo, com o

agente de IDS alertando o usuário local), já de forma global coordena ações entre os nós vizinhos, efetuando na rede uma ação corretiva. Um módulo de comunicação segura garante uma alta confiança no canal de comunicação entre os agentes IDS [Mishra et al. 2004].

2.3.2 IDS baseado no protocolo AODV

[Bhargava e Agrawal 2001] propuseram um modelo de detecção e resposta à intrusão (IDRM - *Intrusion Detection and Response Model*) utilizando o protocolo de roteamento reativo, ou seja, sob demanda, AODV (*Ad hoc On-Demand Distance Vector Routing*). O modelo de detecção proposto por estes autores é uma extensão do modelo descrito na Seção 2.3.1.

A Figura 2.4 ilustra como o IDRM provê a segurança do protocolo AODV. Neste esquema cada nó emprega o IDRM utilizando informações de sua vizinhança para detectar um comportamento considerado incorreto de seus vizinhos. Quando a contagem de comportamentos incorretos (*Malcount*) excede um valor predefinido para o nó, as informações são enviadas para outros nós como parte da resposta global. Os outros nós recebem estas informações, verificam sua *Malcount* local para este nó malicioso e adicionam seus resultados no início da resposta. No modelo de resposta à intrusão (IRM - *Intrusion Response Model*) um nó identifica o outro nó como comprometido quando sua *Malcount* excede o valor limite pré-estipulado. Neste caso é propagada uma mensagem para a rede inteira contendo estas informações por um tipo especial de pacote chamado MAL. Se outro nó também suspeitar que o nó descoberto está comprometido, ele reporta sua suspeita para toda a rede com um novo tipo de pacote, o REMAL. Se dois ou mais nós reportarem sobre um nó em particular, outro pacote especial chamado PURGE é transmitido para isolar o nó malicioso da rede. Todos os nós que têm uma rota com o nó comprometido buscam novas rotas e todos os pacotes recebidos do nó comprometido são descartados [Mishra et al. 2004].

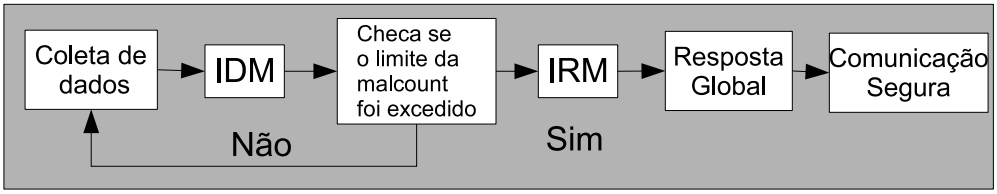


Figura 2.4: Manipulação de ataques internos [Mishra et al. 2004]

Alguns dos ataques internos incluem a distribuição de requisições de rota, a negação de serviço (*DoS - Deny of Service*), a personificação e o comprometimento de um destino.

[Bhargava e Agrawal 2001] identificam os seguintes modos de ataques internos:

- *Pedidos falsos de rota distribuídos*: um nó malicioso pode enviar desnecessariamente vários pedidos de descoberta de rota. Quando os nós da rede recebem um número de requisição de rotas maior do que um limite pré-estabelecido por uma fonte para um destino em um intervalo de tempo particular, o nó é declarado malicioso;

- *Negação de serviço (DoS)*: um nó malicioso lança o ataque de DoS transmitindo falsos pacotes de controle e usando todos os recursos da rede. O DoS pode ser lançado transmitindo falsas mensagens de roteamento ou falsos pacotes de dados. Pode ser identificado se um nó está gerando pacotes de controle em um tempo menor do que o intervalo de tempo pré-estipulado;

- *Destino comprometido*: este ataque é identificado quando a fonte não recebe um pacote de resposta do destino após certo intervalo de tempo. Os vizinhos geram pacotes de *probe/hello* para atestar a conectividade;

- *Personificação*: pode ser evitado se o remetente codificar seu pacote de envio com a chave privada e os outros nós decodificarem com a chave pública do remetente. Se o receptor não puder decriptografar o pacote, o remetente não é uma fonte real; Conseqüentemente o pacote é descartado.

2.3.3 IDS baseado em *Watchdog* e *Pathrater*

Conforme descrito em [Marti et al. 2000], o trabalho apresenta a junção de duas técnicas: *watchdog* e *pathrater*. O *watchdog* trabalha identificando nós com comportamentos inadequados através de observações feitas nas transmissões e mantém uma pontuação equivalente a cada transmissão coordenada pelo nó que está utilizando o *watchdog*. No momento em que um limite pré-estabelecido é atingido, o nó de origem que está utilizando a rota recebe uma mensagem de alerta.

O *pathrater* realiza um trabalho complementar ao *watchdog*, pois além de avaliar as informações obtidas pelo comportamento dos nós, também verifica possíveis rotas entre uma origem e um destino, de acordo com sua disponibilidade, dando maior prioridade para aquelas rotas que apresentarem melhores condições para utilização.

Essa técnica possui algumas limitações, como: colisões ambíguas (Figura 2.5), colisões no receptor (Figura 2.6), força de transmissão limitada, falso comportamento, conspiração

e descarte parcial. A consipração se refere a dois ou mais nós, impedindo que os demais membros da rede saibam de seu mau comportamento através da cumplicidade. Por serem cúmplices, o nó com *watchdog* nunca sabe se o pacote foi realmente encaminhado ou não. Já o descarte parcial pode ser compreendido pela técnica de se descartar apenas alguns pacotes, a fim de impedir sua classificação pelo *watchdog* como mau comportamento.

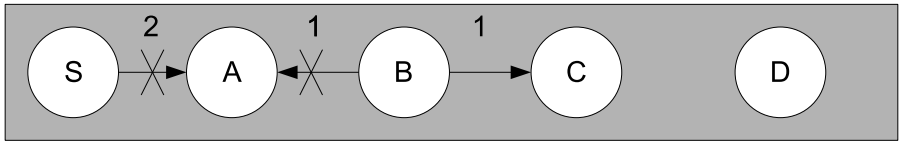


Figura 2.5: O nó A não ouve o pacote 1 que B encaminha a C, porque a transmissão de B colide em A com o pacote 2 que S enviou [Mishra et al. 2004].

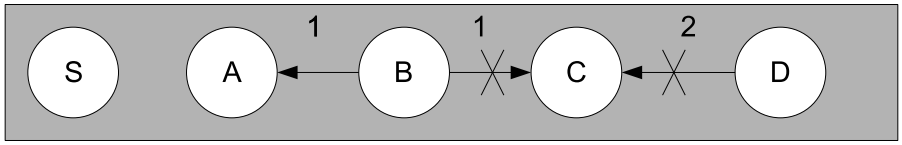


Figura 2.6: O nó A acredita que B enviou o pacote 1 a C, embora C nunca receba o pacote devido a uma colisão com pacote 2 [Mishra et al. 2004].

2.3.4 Técnica para Resistência de Intrusão em Algoritmos de Roteamento *ad hoc*

O TIARA, do inglês *Techniques for Intrusion-Resistant Ad hoc Routing Algorithms*, é apresentado por [Ramanujan et al. 2003] como um conjunto de técnicas que visam a proteção a ataques de negação de serviço. O TIARA funciona de forma que seja possível trabalhar com níveis aceitáveis de operações na rede mesmo durante tais ataques. Sua medida de precaução provê resistências a ataques que visem o roteamento de dados, indo de encontro aos intrusos que os promovem. As técnicas oriundas do TIARA visam a utilização principalmente dos protocolos reativos, como o AODV e o DSR.

Como princípios desta técnica, podem ser citados: o denominado controle de acesso a rotas baseado em fluxo (*Flow-based Route Access Control* - FRAC), que utiliza uma lista de controle de acesso armazenada em cada nó, identificando as mensagens trafegadas que possuem autorização para serem encaminhados. Como uma segunda técnica tem-se a que utiliza o roteamento por múltiplos caminhos, onde os procedimentos de descoberta e manutenção de rotas constantemente trabalham para que todas as rotas que tenham como destino certo nó sejam encontradas e mantidas, na intenção de se tornar mais tolerante a falhas causadas por nós intrusos. O terceiro princípio, denominado roteamento de fluxo

iniciado pela origem (*Source-Initiated Flow Routing*), diz que a origem de cada pacote deve inserir um rótulo que indique qual dos múltiplos caminhos conhecidos deve ser tomado.

O TIARA também utiliza um mecanismo de monitoramento de fluxo de dados, que exige que o nó de origem transmita periodicamente mensagens que indiquem o estado do fluxo de pacotes. Com isso, o nó de destino, que deve monitorar os fluxos ativos dos quais ele participa, pode armazenar os pacotes recebidos entre as mensagens de estado do fluxo e caso a diferença entre o número de pacotes recebidos pelo destino e o de pacotes enviados pela origem seja muito grande ou o tempo de espera da mensagem de estado do fluxo ultrapasse um limite estipulado, é assumido que houve uma falha na rota.

A autenticação dos pacotes neste protocolo é feita por um mecanismo denominado autenticação rápida (*Fast Authentication*), que obriga os nós a inserir uma espécie de rótulo do caminho em um local secreto a cada pacote transmitido. Essa localização secreta é determinada no estabelecimento da rota entre os nós comunicantes e deve ser diferente para cada nó. O TIARA utiliza também um controle por números de seqüência a fim de evitar ataques de replicação.

O último mecanismo proposto, denominado mecanismo de alocação de recursos baseado em referência, determina a quantidade máxima de recursos que cada nó pode alocar para a transmissão de um determinado fluxo. Alocações adicionais de recursos somente são permitidas no caso de o nó de origem apresentar recomendações oriundas de nós confiáveis que garantam a autenticidade do seu pedido.

2.3.5 IDS baseado em Agentes Móveis

São tipos especiais de agentes que têm a habilidade de se mover por grandes redes. Em sua movimentação podem interagir com os nós, colecionando informações e executando tarefas atribuídas a eles. Oferecem vantagens como a redução do tráfego de mensagens na rede, alcançada por meio da eliminação da necessidade de se enviar grandes quantidades de dados pela rede por programas de análise para auditoria. Quando partes do IDS são destruídos ou separados, devido ao particionamento da rede, eles continuam a trabalhar, aumentando assim o nível de tolerância à falhas da rede. Estes tendem a ser independentes de arquiteturas de plataformas, e deste modo habilitam a utilização de IDSs baseados em agentes em ambientes com sistemas operacionais diferentes [Mishra et al. 2004].

2.3.5.1 Sistema de Detecção de Intrusão Local

O sistema de detecção de intrusão local (LIDS - Local Intrusion Detection System) é distribuído por natureza e utiliza agentes móveis em cada nó da rede *ad hoc* móvel [Albers et al. 2002]. O LIDS é executado em diversos nós colaboradores e tem uma preocupação com a intrusão local visando a segurança global. Neste modelo são utilizados dois tipos de dados, os de segurança, que servem como complemento às informações dos nós vizinhos, e os de alerta, responsáveis por informar os outros nós sobre intrusões localmente descobertas.

Os LIDS utilizam o protocolo SNMP (*Simple Network Management Protocol*), o que diminui muito o custo da coleta de dados, dados esses localizados em bases de gerenciamento de informações (MIBs) utilizados como fonte de auditoria. Alguns LIDS são capazes de delegar uma missão específica para um agente.

O agente LIDS local pode usar detecção por assinatura ou anomalia. Conforme o mecanismo de resposta, assim que um LIDS descobre uma intrusão local, informa aos outros nós da rede a fim de desencadear uma resposta global, além de agir localmente autorizando o nó a recusar conexões destes nós suspeitos. A estrutura deste modelo pode ser vista na Figura 2.7.

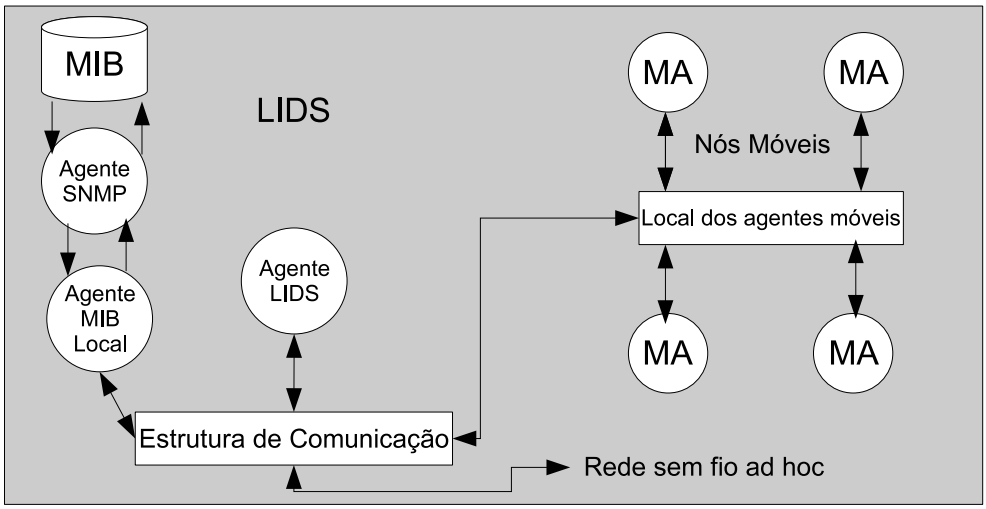


Figura 2.7: Sistema de detecção de intrusão local [Mishra et al. 2004]

O LIDS utiliza um padrão de formato de alerta, denominado formato de troca de mensagem de detecção de intrusão (IDMEF - *Intrusion Detection Message Exchange Format*) transportado pelo protocolo de troca de mensagens de detecção de intrusão (IDXP - *Intrusion Detection Exchange Protocol*). Com a utilização destes padrões é assegurado um grande número de plataformas suportadas, podendo haver troca de informações

relacionadas à intrusão.

2.3.5.2 Detecção de intrusão distribuída usando agentes móveis

[Kachirski e Guha 2002] propuseram um sistema distribuído de detecção de intrusão para redes sem fio *ad hoc* baseado na tecnologia de agentes móveis. Tendo como principal característica a auditoria em múltiplos sensores de rede, este modelo trabalha com nós chaves, restringindo as intensas análises a alguns nós, e assim visando a segurança da rede como um todo. Os nós chave são dinamicamente eleitos impedindo que a segurança da rede como um todo dependa de um nó em particular.

O IDS proposto é fundamentado em um agente móvel conforme a Figura 2.8 e em um sistema não monolítico que emprega vários tipos de sensores que desempenham funções específicas, como: monitoramento da rede, monitoramento por nó, tomada de decisão e ação. Os agentes podem ser classificados em três categorias: monitoramento, tomada de decisão e agentes de ação. Alguns encontram-se presentes em todos os nós, enquanto outros são distribuídos para um grupo de nós.

A rede é dividida em grupos (*clusters*), onde se elege um líder (*cluster-head*), responsável por monitorar todos os pacotes que trafegam dentro daquele agrupamento. Os sensores de monitoramento de rede colecionam todos os pacotes dentro do seu alcance de comunicação e os analisam em busca de padrões de ataque. Os agentes de monitoramento são os responsáveis por monitorar os pacotes, as atividades de usuários e de sistema. Os agentes de descoberta estão presentes em cada nó e em sensores de monitoramento realizando a detecção por anomalia. Os agentes de descoberta local procuram por atividades suspeitas no nó, como alocações incomuns de processos na memória e tentativas inválidas de *login*.

Se uma anomalia é descoberta com uma forte evidência, um agente de descoberta local termina o processo suspeito ou trava a saída do usuário iniciando uma re-autenticação por meio de chaves de segurança para toda a rede, porém se a atividade anômala descoberta em um nó por um agente de monitoramento não é conclusiva, o nó é reportado para o agente de decisão do mesmo agrupamento ao qual o nó é integrante. Se há fortes evidências reunidas sobre este nó, vindo de qualquer fonte (incluindo monitoramento de pacotes, resultante de um agente de monitoramento de rede), a ação é empreendida pelo agente daquele nó.

Os agentes de decisão estão localizados no mesmo nó que os agentes de monitoramento

de pacotes. Um agente de decisão contém o estado de todos os nós residentes no agrupamento. Se a evidência de uma atividade anômala existe para cada nó, o agente então decide que o nó foi comprometido a partir dos relatórios dos próprios agentes de monitoramento local e informações de monitoramento de pacotes pertencentes ao nó. Quando certo nível de ameaça é alcançado para um nó em questão, o agente de decisão despacha um comando que define uma ação a ser empreendida pelos agentes locais daquele nó.

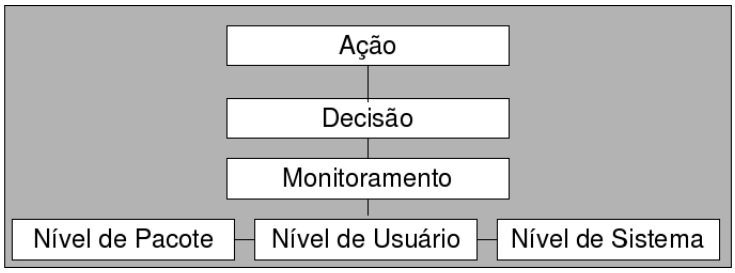


Figura 2.8: Arquitetura modular de detecção de intrusão[Mishra et al. 2004]

2.3.5.3 Arquitetura de Detecção de Intrusão em uma Base Estacionária

Proposto como um IDS distribuído pela Universidade do Estado de Mississippi, este modelo propõe a utilização de agentes móveis em cada nó da rede. Os agentes de IDS trabalham cooperativamente por meio de um algoritmo de descoberta de intrusão a fim de decidir quando e como a rede está sendo atacada. Sua arquitetura é dividida em duas partes: o agente de IDS móvel, que reside em cada nó e na rede, e a base de dados segura e estacionária, que contém assinaturas globais de padrões de ataques conhecidos e padrões de atividades do usuário normal em um ambiente não hostil [Mishra et al. 2004].

Agentes móveis de IDS estão presentes em cada nó da rede, rodando por todo o tempo. Este agente é responsável por detectar intrusões baseado em dados locais de auditoria e a partir de algoritmos cooperativos com outros IDSs. Cada agente pode ser dividido em cinco partes: uma tentativa de auditoria local, uma base de dados de intrusão local (LID), um módulo de comunicação segura, módulos de descoberta de anomalia (ADMs) e módulos de descoberta por assinatura (MDMs), como apresentado na Figura 2.9.

O LID é uma base de dados local que armazena todas as informações necessárias para o agente do IDS tal como, os arquivos de assinatura de ataques conhecidos, os padrões de utilização do usuário e o fluxo normal de tráfego na rede. Os ADMs e os MDMs se comunicam diretamente com o LID para determinar se uma intrusão está acontecendo.

Neste modelo é utilizado um módulo de comunicação segura entre os agentes de IDS, transmitidos codificadamente que permite que os MDM e ADM usem algoritmos coopera-

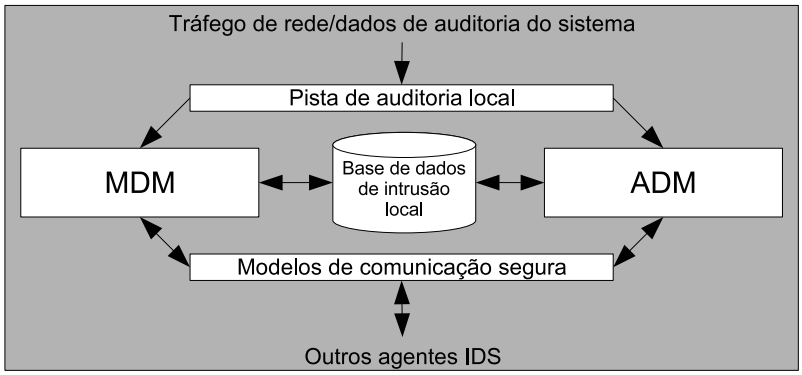


Figura 2.9: Arquitetura de detecção de intrusão em uma base estacionária [Mishra et al. 2004]

tivos na descoberta de intrusos. Além disso, pode desencadear uma resposta global desde que um ou vários agentes de IDS descubram intrusões.

Os ADMs são os responsáveis pela descoberta de um novo tipo de anomalia. Pode haver um ou mais ADMs em cada agente móvel de IDS, cada um trabalhando separadamente e cooperativamente.

Os MDMs identificam padrões conhecidos de ataques, que por sua vez são especificados no LID. Como no ADM, os MDMs podem determinar, com base nos dados de auditoria disponíveis, que se uma intrusão está acontecendo localmente, isso já é suficiente para desencadear uma resposta, sendo também possível a utilização de um algoritmo cooperativo para a identificação de intrusão.

A base de dados segura estacionária (*Stationary Secure Database* - SSD) serve como uma base de consulta aos nós móveis para obter padrões recentes de atividade do usuário normal a fim de encontrar assinaturas de ataques. É considerado que a base não será atacada ou comprometida, sendo armazenada em um área física de alta segurança. Os agentes móveis colecionam e armazenam dados de auditoria, como tráfego de rede, comandos do usuário, etc.

Em sua fase de coleção de dados são transferidas as informações coletadas assim que se acoplarem ao SSD, então ele usa estas informações para a mineração de dados e criação de novas regras para detecção de ataques por anomalia. O SSD também pode ser administrado por um especialista, que pode especificar novas assinaturas. Quando os agentes estão acoplados ao SSD, eles atualizam seus padrões de assinaturas inseridos recentemente. Com a utilização de um SSD, a tendência é que a comunicação entre os agentes das MANETs para troca de informações sobre novas assinaturas, seja bem menor.

Apesar de várias vantagens em se manter um SSD na arquitetura de agentes móveis, há também desvantagens em se manter uma base fixa de informações aos IDSs, como no caso da perda de mobilidade dos agentes enquanto acoplados à SSD e também pelo fato da SSD ser considerada uma base segura e não levam em consideração nenhum risco ou vulnerabilidade.

2.4 Conclusão

Neste capítulo foram apresentados os principais conceitos de IDS e os modelos existentes propostos para redes *ad hoc*. Todos esses modelos citam algum modo de comunicação segura para troca de mensagens entre os IDSs, mas não abordam a sua implementação. Essa comunicação segura normalmente envolve criptografia, recaindo na questão principal de como distribuir chaves de maneira segura entre os membros da rede.

A Tabela 2.1 apresenta as características principais e a metodologia utilizada para detecção de intrusão dos modelos de IDS descritos neste capítulo.

Tabela 2.1: Principais características dos modelos de IDS apresentados [Mishra et al. 2004].

Modelos	**Características**	**Metodologia**
IDS distribuídos	Uso de estatísticas de detecção de anomalias para detecção local e global	Detecção por anomalia cooperativa
IDS baseado no protocolo AODV	Esquema colaborativo baseado em pontuação onde os nós trabalham em cooperação com seus vizinhos em busca de atividades maliciosas. Se dois ou mais nós relatam sobre um nó em particular este é isolado da rede.	Detecção por anomalia cooperativa
TIARA	Apresenta técnicas independentes de roteamento que podem ser incorporados em MANETS para uma maior resistência a falhas e ataques.	Detecção por anomalia cooperativa
Watchdog e Pathrater	Esquema baseado em pontuação onde os nós trabalham analisando seus vizinhos em busca de atividades maliciosas. Uma vez que a pontuação é mínima o nó é excluído da rede.	Detecção por anomalia cooperativa
continua na próxima página		

Tabela 2.1 – continuação da página anterior

Modelos	**Características**	**Metodologia**
Sistema de Detecção de Intrusão Local	Os agentes móveis usam dados SNMP situados em uma base de informações gerenciáveis como fonte na detecção de intrusão. Utiliza o IDMEF como padrão de troca de mensagens sobre o IDXP, que é responsável pelo transporte das mensagens, a fim de fazer a troca de mensagens entre os agentes em uma escala maior e garantir o sucesso da detecção.	Detecção por anomalia cooperativa baseada em agentes móveis.
Detecção de Intrusão Distribuída utilizando Agentes Móveis	Examina dados de vários sensores, analisa toda a faixa da rede ad hoc a fim de descobrir intrusões em níveis múltiplos, tenta impedir os ataques.	Baseado em agentes móveis de detecção por combinação
Arquitetura de Detecção de Intrusão em uma Base Estacionária	Os dados de auditoria usados são coletados por vários sensores, implementando um esquema de largura de banda consciente para a detecção de intrusão distribuída usando agentes móveis.	Detecção por anomalia baseado em agentes móveis

Uma infra-estrutura de chave pública (PKI) pode ser utilizada para tal, mas pode implicar uma centralização do mecanismo de troca de chaves, tornando a rede vulnerável. A distribuição de responsabilidades entre diversos servidores em um PKI foi proposto em [Zhou e Haas 1999], mas introduz um *overhead* grande de troca de mensagens na rede. Modificações foram propostas por [Brazil 2007], tentando minimizar o volume de mensagens trocadas na rede.

No Capítulo 3 é apresentada uma alternativa para a criptografia, a esteganografia, onde o objetivo principal é a troca de mensagens escondidas em outras mensagens transmitidas na rede, visando eliminar a necessidade de troca de chaves criptográficas.

Capítulo 3

Esteganografia

A segurança digital é uma área com grande potencial para pesquisa e desenvolvimento. Sistemas de detecção de intrusão, anti-vírus, *proxies* e *firewalls* ultimamente aparecem muito na mídia em geral e estão se tornando ferramentas de uso doméstico. É cada vez maior o número de pessoas que tentam a todo custo ludibriar as defesas para ter acesso a um dos bens mais preciosos da sociedade moderna: a informação. Por outro lado, existem outras pessoas que buscam o desenvolvimento e o estudo de técnicas para proteção das comunicações. As ferramentas e técnicas que provêem a segurança da informação são inúmeras e a criptografia está entre elas há muitos anos.

Esse capítulo aborda a esteganografia, um dos ramos do ocultamento de informações, onde o objetivo principal é esconder uma mensagem em outra, bem como a evolução da esteganografia ao longo da história e suas aplicações modernas com a chamada esteganografia digital. São mostradas as principais técnicas de mascaramento e, em especial, mascaramento em imagens.

3.1 Introdução

De origem grega, a palavra esteganografia significa a arte da escrita escondida (estegano = esconder e grafia = escrita). A esteganálise por sua vez é a arte de detectar mensagens escondidas nos mais diversos meios de comunicação. A esteganografia inclui um amplo conjunto de métodos e técnicas para prover comunicações secretas desenvolvidos ao longo da história. Dentre as técnicas se destacam: tintas invisíveis, micropontos, arranjo de caracteres (*character arrangement*), assinaturas digitais e canais escondidos (*covert channels*) [Petitcolas et al. 1999, Petitcolas e Katzenbeisser 1999, Johnson e Jajodia 1998].

As aplicações de esteganografia incluem a identificação de componentes dentro de

um subconjunto de dados, a legendagem (*captioning*), o rastreamento de documentos e certificação digital (*time-stamping*) e a demonstração de que um conteúdo original não foi alterado (*tamper-proofing*). Entretanto, como qualquer técnica, a esteganografia pode ser usada correta ou incorretamente. Há indícios de que a esteganografia tem sido utilizada para divulgar imagens de pornografia infantil na Internet, além das mensagens de redes terroristas como a Al-Qaeda [Morris 2004, Hart et al. 2004].

3.1.1 Terminologia

Há um interesse cada vez maior, por diferentes comunidades de pesquisa, no campo da esteganografia, marcas d'água e seriação digitais. Com certeza, isso leva a uma certa confusão na terminologia. A seguir, encontram-se alguns dos principais termos utilizados nestas áreas e ilustrados na Figura 3.1:

- dado embutido ou *embedded data* - é o dado que é enviado de maneira secreta, normalmente em uma mensagem, texto ou figura;
- mensagem de cobertura ou *cover-message* - é a mensagem que serve para mascarar o dado embutido. Esta mensagem de cobertura pode ser de áudio (*cover-audio*), de texto (*cover-text*) ou uma imagem (*cover-image*);
- estego-objeto ou *stego-object* - após a inserção do dado embutido na mensagem de cobertura se obtém o estego-objeto;
- estego-chave ou *stego-key* - adicionalmente pode ser usada uma chave para se inserir os dados do dado embutido na mensagem de cobertura. A esta chave dá-se o nome de estego-chave;
- número de série digital ou marca *fingerprinting* - consiste em uma série de números embutidos no material que é protegido a fim de provar a autoria do documento.

A esteganografia pode ser dividida em dois tipos: técnica e lingüística. O primeiro tipo se refere às técnicas utilizadas quando a mensagem é fisicamente escondida, como por exemplo escrever uma mensagem em uma tábua de madeira e cobri-la com cera, como faziam alguns povos na antigüidade. A esteganografia lingüística se refere ao conjunto de técnicas que se utilizam de propriedades lingüísticas para esconder a informação, como por exemplo *spams* e imagens.

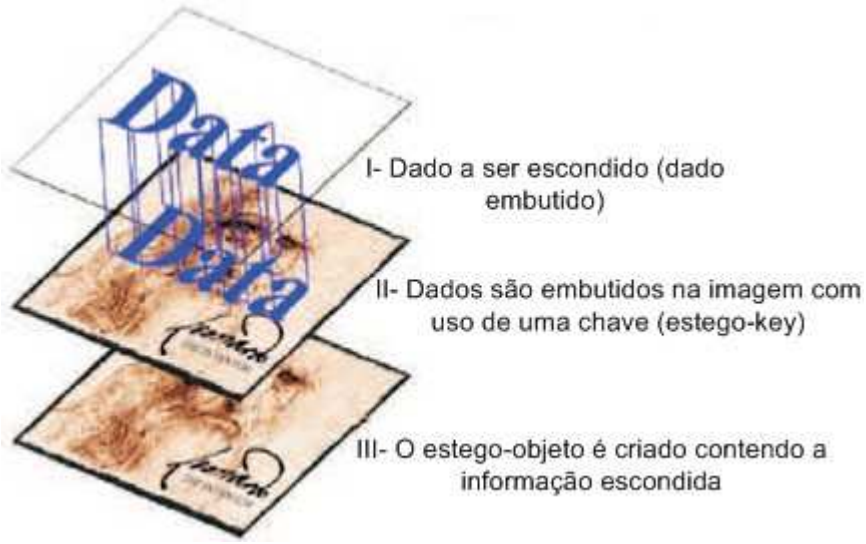


Figura 3.1: Escondendo uma imagem [Petitcolas et al. 1999].

Os sistemas de marcação visam proteger a propriedade intelectual sobre algum tipo de mídia (eletrônica ou não). Estes sistemas de marcação são conhecidos também como *watermarking* (marca d'água). Apesar de aparecerem quase sempre em conjunto com esteganografia, os sistemas de marcação não pertencem ao ramo da esteganografia. Ambos pertencem a uma área de pesquisa conhecida como **ocultamento da informação** ou ***information hiding***, cuja taxonomia é apresentada na Figura 3.2.

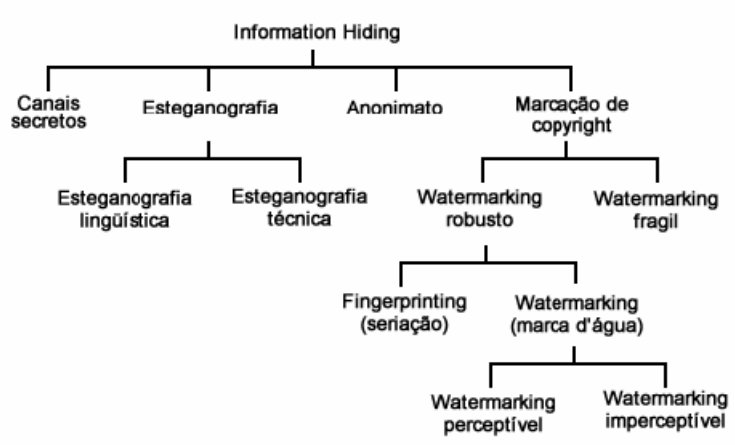


Figura 3.2: A taxonomia do ocultamento de informação [Rocha 2006].

O sistema de marcação tipo marca d'água se refere a métodos que escondem informações em objetos que são robustos e resistentes a modificações. Neste sentido seria impossível remover uma marca d'água de um objeto sem alterar a qualidade visual do mesmo. Por outro lado a esteganografia se propõe a esconder uma informação em uma imagem de cobertura. Se a imagem for destruída ou afetada a mensagem é perdida. Uma outra diferença clara entre esteganografia e técnicas de marca d'água é que enquanto o

dado embutido da esteganografia nunca deve ficar aparente, a marca d'água pode ou não aparecer no objeto marcado, dependendo da aplicação que se queira atender.

Neste sentido pode-se classificar os sistemas de marcação de acordo com a sua robustez e a sua aparência. Segundo sua robustez, podem ser classificados como:

- **robustos** - são aqueles em que mesmo após a tentativa de remoção a marca permanece intacta;
- **frágeis** - são os sistemas em que qualquer tentativa de modificação na mídia acarreta a perda da marcação. É muito útil para verificação de cópias ilegais. Quando se copia um objeto original, a cópia é feita sem a marca.

Já quanto à sua aparência, os sistemas de marcação podem ser classificados como:

- **de marcação imperceptível** - são os sistemas onde a marca encontra-se no objeto ou material, porém não é visível diretamente;
- **de marcação visível** - neste sistema a marca do autor deve ficar visível para comprovar a autoria visualmente. Um bom exemplo deste sistema são as marcas d´água em cédulas de dinheiro e em selos.

3.1.2 Aspectos Históricos

A esteganografia é uma arte antiga. Suas origens remontam à antigüidade. Os gregos já a utilizavam para enviar mensagens em tempos de guerra [Kahn 1996]. Nas "Estórias de Herodotus", existem muitas passagens mostrando o uso da esteganografia. Em uma estória, um mensageiro se disfarçou de caçador para enviar uma mensagem ao rei escondendo-a dentro de uma lebre. Como o mensageiro estava disfarçado, passou despercebido pelos portões do palácio e o rei pôde receber a mensagem.

Mensagens também foram enviadas através de escravos de confiança. Alguns reis raspavam as cabeças de escravos e tatuavam as mensagens nelas. Depois que o cabelo crescesse, o rei mandava o escravo pessoalmente com a mensagem [Kahn 1996]. Ninguém suspeitaria onde a mensagem se encontrava, a menos que soubesse exatamente onde procurar. Neste caso o segredo com a localização da mensagem deveria ser mantido. Outro exemplo de esteganografia na Grécia antiga era furar buracos em livros acima das letras que formavam a mensagem desejada. Quando o destinatário recebesse o livro poderia

procurar pelos buracos sobre as letras para reconstruir as mensagens. Para quem não soubesse do código, o livro pareceria ter apenas seu conteúdo escrito pelo autor.

Os chineses e egípcios também criaram seus métodos de esteganografia na idade antiga. Os chineses escreviam mensagens em finas folhas de papel de seda que eram depois enroladas como uma bola e cobertos com cera. Esta bola era então escondida em algum lugar do corpo ou engolida para prevenir sua detecção. Os egípcios usavam ilustrações para cobrir as mensagens escondidas. O método de escrita egípcio conhecido como hieróglifo era uma técnica comum para esconder mensagens. Quando um mensageiro egípcio era pego com um hieróglifo que continha algum código, o inimigo não suspeitava e a mensagem podia ser entregue sem problemas ao destinatário.

Durante a idade média, a esteganografia foi mais estudada e desenvolvida. Em 1499, um monge chamado Tritheimius escreveu uma série de livros chamados "Steganographia" (Figura 3.3) nos quais ele descreveu várias técnicas diferentes. Uma delas, desenvolvida na idade média, foi a grade de Cardano [Kahn 1996]. Criada por Girolamo Cardano, a grade era uma lâmina que randomicamente definia retângulos. A quantidade e o posicionamento dos retângulos era o segredo da grade. O remetente escrevia as palavras da mensagem secreta nos retângulos. Depois a grade era removida e o remetente preenchia os espaços remanescentes com letras ou palavras para criar a mensagem que seria enviada (mensagem de cobertura). Uma vez entregue a mensagem, o destinatário colocaria a grade, que era a mesma do emissor, sobre o papel ou superfície que continha a mensagem e podia lê-la sem problemas, lendo os caracteres que estariam dentro dos retângulos.

Figura 3.3: Exemplar de *"Schola Steganographica"* publicado em 1680 [Petitcolas e Katzenbeisser 1999].

Os primeiros experimentos com tintas invisíveis também começaram na idade média. Giovanni Porta escreveu vários livros de história natural. Dentro destes livros estavam receitas de tintas secretas que poderiam ser usadas para escrever sobre a pele humana e outras superfícies. Este tipo de tinta foi desenvolvido e usado mais tarde no fim dos anos

de 1700 e foi a chave para comunicações secretas.

Tintas invisíveis também foram muito usadas em esteganografia nos tempos mais modernos e são utilizadas até hoje. Estas tintas foram utilizadas por espiões durante a primeira e a segunda grande guerra com o desenvolvimento de reagentes químicos específicos para cada tinta. Textos eram escritos em jornais, revistas ou livros com tintas invisíveis para serem passados de forma segura até seus destinatários. Uma outra utilização era escrever a mensagem com tinta invisível sobre um papel, cortá-lo em alguns pedaços e depois rejuntá-los no destinatário [Kahn 1996].

Outros métodos modernos de esteganografia incluem cifradores nulos e micro pontos. Cifradores nulos são mensagens nas quais certas letras devem ser usadas para formar a mensagem e todas as outras palavras ou letras são consideradas nulas. Para o uso do cifrador nulo, ambos os lados da comunicação devem usar o mesmo protocolo de uso das letras que formam a mensagem. Por exemplo, usar sempre a primeira letra de cada palavra para compor a mensagem. Este método é difícil de implementar, pois a mensagem de cobertura deve ter algum sentido, do contrário um inimigo desconfia e quebra o código. Um exemplo de um código utilizando cifrador nulo é mostrado a seguir [Johnson e Jajodia 1998].

"News Eight Weather: tonight increasing snow. Unexpected precipitation smothers eastern towns. Be extremely cautious and use snowtires especially heading east. The highways are knowingly slippery. Highway evacuation is suspected. Police report emergencies in downtown ending near Tuesday".

Usando as primeiras letras de cada palavra o texto que aparece é: **"Newt is upset because he thinks he is president"**.

A técnica de Micro-pontos é também uma outra forma de esteganografia usada atualmente. Um micro-ponto é uma fotografia da mensagem secreta que deve ser entregue. Com a tecnologia avançando rapidamente, é possível tirar uma foto de uma mensagem e reduzi-la a uma fotografia circular de 0,05 polegadas ou 0,125 cm de diâmetro. Esta minúscula fotografia é então colada em um sinal de pontuação de uma frase ou no "pingo" de uma letra "i" de uma outra mensagem qualquer que será entregue. Somente aqueles que sabem onde procurar o micro-ponto poderão detectar sua presença.

Atualmente, novas técnicas de ocultamento de informação são produzidas para serem utilizadas em conjunto com a esteganografia nos novos meios de comunicação. Por exemplo, hoje em dia muitos artistas e gravadoras estão utilizando a marca d'água para

proteger suas obras. Com o crescente aumento da pirataria e de *sites* na Internet onde se pode baixar filmes, músicas e vídeos, esta técnica tem se mostrado uma aliada na proteção dos direitos autorais. O uso de esteganografia em software tem um grande potencial, pois pode esconder dados em uma infinidade de mídias. Nas técnicas que utilizam o último bit de um byte para esconder mensagens, uma mensagem de 64Kbytes pode ser escondida em uma figura de 1024 x 1024 em tons de cinza ou imagens coloridas. Esta e outras novas técnicas, representam o estado da arte da esteganografia atual e são apresentadas a seguir.

3.2 Técnicas de Esteganografia

As imagens são a mídia de cobertura mais popular para esteganografia e podem ser armazenadas em um formato bitmap direto (como BMP) ou em um formato comprimido (como JPEG). Imagens de palheta de cores estão normalmente no formato GIF. O ocultamento de informações é realizado ou no domínio espacial ou no domínio de freqüência. Em termos de esquemas de inserção, vários métodos (como substituição, adição e ajuste) podem ser usados. Uma abordagem de ajuste é a QIM (*Quantization Index Modulation*), que usa diferentes quantizadores para transportar diferentes bits dos dados secretos [Sullivan et al. 2004].

As abordagens mais comuns de inserção de mensagens em imagens incluem técnicas de inserção no bit menos significativo, técnicas de filtragem e mascaramento e algoritmos e transformações. Cada uma destas técnicas pode ser aplicada à imagens, com graus variados de sucesso. O método de inserção no bit menos significativo é provavelmente uma das melhores técnicas de esteganografia em imagem [Petitcolas et al. 1999, Wayner 2002].

3.2.1 Requisitos para Sistemas Esteganográficos

Os três requisitos mais importantes que devem ser satisfeitos para qualquer sistema esteganográfico são:

- segurança - a fim de não levantar suspeita, enquanto tenta criar uma blindagem contra um algoritmo de descoberta, o conteúdo escondido deve ser invisível tanto perceptivelmente quanto por meios estatísticos [Buccigrossi e Simoncelli 1999]. Algumas definições baseadas em informações teóricas para um sistema seguro perfeito assumem conhecimento detalhado das estatísticas da cobertura e exigem muitos

recursos computacionais. Estas condições não são estritamente encontradas em aplicações esteganográficas reais. Por exemplo, relativo a conhecimento estatístico, pode-se estimar um conjunto particular de sinais freqüentemente utilizados por um certo grupo de pessoas e estabelecer um modelo para descoberta. Mas tais modelos não tem sentido se o erro de estimação excede a extensão de modificações causadas por inclusão. Além disso, a complexidade computacional de qualquer ferramenta de esteganografia útil não pode ser infinitamente grande. Em termos de praticidade, um sistema pode ser considerado seguro, ou esteganograficamente forte [Duda et al. 2000], se não for possível descobrir a presença de stego-conteúdo usando qualquer meio acessível;

- carga útil - diferentemente de marca d'água, que precisa embutir somente uma quantia pequena de informações de direitos autorais, a esteganografia é direcionada à comunicação escondida e portanto normalmente exige capacidade de inclusão suficiente. Os requisitos para capacidade significativa de dados e segurança são freqüentemente contraditórios. Dependendo dos argumentos de aplicação específica, um compromisso deve ser buscado;

- robustez - embora robustez contra ataques não seja uma prioridade importante, como em marcas d'água, ter a capacidade de resistir a compressão é certamente desejável, pois a maioria das imagens JPEG coloridas são comprimidas antes de serem colocadas on-line.

3.2.2 LSB

Estas técnicas são baseadas na modificação dos bits menos significativos (*Least Significant Bit* - LSB) dos valores de *pixel* no domínio espacial. Em uma implementação básica, estes *pixels* substituem o plano LSB inteiro com o stego-dados. Com esquemas mais sofisticados em que locais de inclusão são adaptativamente selecionados, dependendo de características da visão humana, até uma pequena distorção é aceitável. Em geral, a inclusão de LSB simples é suscetível a processamento de imagem, especialmente a compressão sem perda.

Técnicas baseadas em LSB podem ser aplicadas a cada *pixel* de uma imagem codificada em 32 bits por *pixel*. Estas imagens possuem seus *pixels* codificados em quatro bytes. Um para o canal alfa (*alpha transparency*), outro para o vermelho (*red*), outro para o verde (*green*) e outro para o azul (*blue*). Seguramente, pode-se selecionar um bit (o menos significativo) em cada byte do *pixel* para representar o bit a ser escondido sem causar alteracões

perceptíveis na imagem. Estas técnicas constituem a forma de mascaramento em imagens mais difícil de ser detectada pois podem inserir dados em *pixels* não sequenciais, tornando complexa a detecção [Popa 1998, Petitcolas et al. 1999, Wayner 2002].

3.2.3 Filtragem e Mascaramento

As técnicas de esteganografia baseadas em filtragem e mascaramento são mais robustas que a inserção LSB. Estas geram estego-imagens imunes à compressão e recorte. No entanto, são técnicas mais propensas a detecção [Wayner 2002]. Ao contrário da inserção no canal LSB, as técnicas de filtragem e mascaramento trabalham com modificações nos bits mais significativos das imagens. As imagens de cobertura devem ser em tons de cinza porque estas técnicas não são eficazes em imagens coloridas [Popa 1998]. Isto deve-se ao fato de que modificações em bits mais significativos de imagens em cores geram muitos artefatos tornando as informações mais propensas à detecção.

Estas técnicas são semelhantes à marca d'água visível em que valores de *pixel* em áreas mascaradas são aumentados ou diminuídos por um pouco de porcentagem. Reduzindo o incremento por um certo grau faz a marca invisível.

3.2.4 Algoritmos e Transformações

As técnicas de esteganografia baseadas em algoritmos e transformações conseguem tirar proveito de um dos principais problemas da inserção no canal LSB que é a compressão. Para isso são utilizadas: a transformada de Fourier discreta, a transformada de cosseno discreta e a transformada Z [Gonzalez e Woods 2002].

Sendo embutido no domínio de transformação, os dados escondidos residem em áreas mais robustas, espalhadas através da imagem inteira e fornecem melhor resistência contra processamento de sinal. Configuram-se como as mais sofisticadas técnicas de mascaramento de informações conhecidas, embora sofisticação nem sempre implique em maior robustez aos ataques de esteganálise. A inclusão de dados apresentados no domínio de transformação é amplamente usada para marca d'água robusta.

De forma geral, estas técnicas baseadas em algoritmos e transformações aplicam uma determinada transformação em blocos de 8x8 *pixels* na imagem. Em cada bloco, devem ser selecionados os coeficientes que são redundantes ou de menor importância. Posteriormente, estes coeficientes são utilizados para atribuir a mensagem a ser escondida em um processo em que cada coeficiente é substituído por um valor pré-determinado para o bit

0 ou 1 [Popa 1998].

Para melhor entendimento do funcionamento destas técnicas, é explicada a seguir a transformada de cosseno discreta que é muito utilizada nas compressões dos padrões JPEG e MPEG.

3.2.4.1 Transformada de Cosseno Discreta

A transformada de cosseno discreta (DCT - *Discrete Cosine Transform*) é uma transformada matemática baseada em cossenos, muito utilizada em processamento digital de imagens e compressão de dados. O valor da função da DCT de um vetor p de *pixels* de comprimento n é:

$$G_f = \frac{1}{2} C_f \sum_{t=0}^{n-1} p_t \cos\left(\frac{(2t+1)f\pi}{2n}\right), \tag{3.1}$$

$$onde:\ C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0 \\ 1 & f > 0 \end{cases} \text{ para } f = 0, 1, ..., n-1.$$

A matriz dessa transformada é composta de vetores ortonormais, sendo por isso uma matriz de rotação. Na compressão de dados, esta transformada é muito utilizada pois transfere a maior parte da informação contida para os primeiros elementos do vetor, otimizando o armazenamento (para compressão sem perdas) e facilitando a quantização dos valores (para compressão com perdas), conforme mostrado na Figura 3.4.

A recuperação dos dados transformados pode ser feita com a operação inversa, chamada de IDCT (*Inverse Discrete Cosine Transform*), que é dada pela fórmula:

$$p_t = \frac{1}{2} \sum_{j=0}^{n-1} C_f G_j \cos\left(\frac{(2t+1)j\pi}{2n}\right), \text{para } t = 0, 1, ..., n-1. \tag{3.2}$$

Em compressão de imagens e vídeos a maioria dos padrões usa a transformada discreta de cosseno do vetor p com o tamanho $n = 8$ (JPEG e MPEG).

Sabendo que os *pixels* de uma imagem tem correlação com seus vizinhos nas duas dimensões da imagem, e não apenas em uma dimensão, a DCT para ser usada na compressão de imagens também deve ser uma transformada bidimensional. A fórmula para

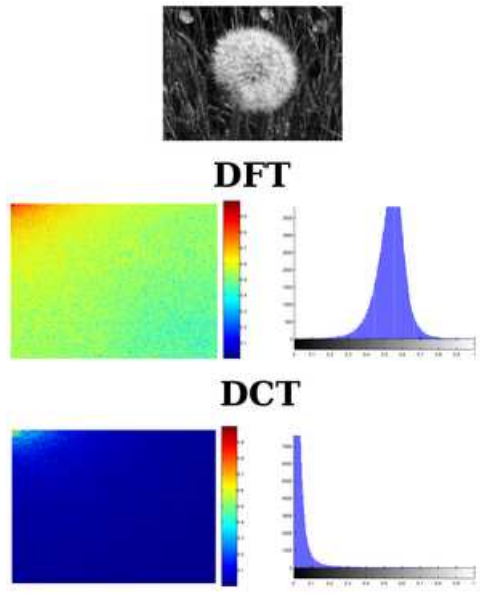


Figura 3.4: Comparação entre a Transformada de Fourier discreta e a DCT.

uma matriz (ou seja uma imagem) p de tamanho n x n é:

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum y = 0^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \tag{3.3}$$

$$\text{para } 0 \leq i, j \leq n-1; \textit{onde } C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0 \\ 1, & f > 0 \end{cases}.$$

Essa transformada pode ser considerada como uma rotação (ou duas rotações consecutivas, uma em cada dimensão), ou ainda como uma base ortogonal em um espaço vetorial de n dimensões. A recuperação dos dados transformados pode ser feita usando a transformação inversa, conhecida como IDCT bidimensional:

$$p_{xy} = \frac{1}{4} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \cos\left(\frac{(2x+1)i\pi}{2n}\right) \cos\left(\frac{(2y+1)j\pi}{2n}\right). \tag{3.4}$$

Analogamente à transformada unidimensional, a transformada bidimensional resulta em uma matriz onde os coeficientes mais significativos se acumulam no canto superior esquerdo (início da matriz) e os demais coeficientes são de pequeno valor podendo ser mais facilmente armazenados ou mesmo quantizados para proporcionar uma compressão com perdas.

Apesar de ser relativamente fácil de implementar em qualquer linguagem de programação, a compressão de imagens demanda um grande poder de processamento e por isso precisa ser otimizada ao máximo. O uso da DCT em imagens grandes, apesar de apresentar ótimos resultados, exige um processamento muito grande. Por isso na prática a estratégia que se adota é dividir a imagem em blocos de tamanho menor (em geral de tamanho 8x8 *pixels*, como no JPEG), levando a uma primeira otimização:

- otimização 1 - a imagem a ser tratada deve ser dividida em blocos menores facilitando a computação das transformadas. Outra justificativa para esta abordagem é que, apesar de existir bastante correlação com os vizinhos próximos, existe pouca ou nenhuma correlação entre pontos distantes de uma mesma imagem. Os ganhos de processamento com esta abordagem suplantam em muito as perdas em termos de compressão.

O cálculo das funções de cosseno, por ser uma função transcendental, também exige bastante poder de processamento. Verificando a fórmula da DCT pode-se pré-calcular todos os valores de cosseno a serem utilizados e, depois disto, apenas realizar operações aritméticas de soma e multiplicação, o que leva à segunda otimização:

- otimização 2 - os cossenos usados devem ser pré-calculados e armazenados, realizando-se assim apenas operações aritméticas ao se calcular a fórmula da transformada.

Com um pouco de esforço algébrico, pode-se provar que a somatório dupla da fórmula da DCT bidimensional na Equação 3.3 corresponde ao produto matricial CPC^T, onde P é a matriz 8x8 representando o bloco de imagem a ser comprimido, C é a matriz definida por:

$$C_{ij} = \begin{cases} \frac{1}{\sqrt{8}}, & i = 0 \\ \frac{1}{2}\cos\left(\frac{(2j+1)i\pi}{16}\right), & i > 0 \end{cases} \tag{3.5}$$

e C^T é a sua transposta. Essa multiplicação matricial exige menor número de multiplicações e somas que a fórmula original, reduzindo ainda mais o tempo de execução da transformada. E isso leva à terceira otimização:

- otimização 3 - aplicação da transformada de cosseno discreta sob a forma matricial CPC^T para reduzir ainda mais o número de operações.

Uma última otimização é a utilização de aritmética de ponto fixo (número fixo de casas decimais). Esta técnica aproveita o fato de que muitos computadores executam as instruções de ponto fixo com mais rapidez do que as de ponto flutuante, acelerando assim o cálculo da transformada. Entretanto, esta técnica introduz uma quantização forçada, mas que no contexto da compressão de dados pode ser desprezada.

- otimização 4 - uso de aritmética de ponto fixo para aproveitar a maior velocidade desse tipo de cálculo na maioria dos computadores.

Ao aplicar a DCT, os coeficientes mais significativos se acumulam no início do vetor (ou matriz) dos dados, ficando o restante com valores muito pequenos e carregando pouca informação. Este tipo de distribuição já é suficiente para que uma técnica de redução de redundância (como os algoritmos LZ77, LZ78 ou LZW) ou uma codificação otimizada (como codificação de Huffman ou codificação aritmética) produzam melhores resultados do que na imagem ou nos dados originais. Entretanto, por se trabalhar sempre com uma precisão finita nas representações numéricas utilizadas, tem-se uma perda nos dados. Portanto, mesmo sem aplicar nenhuma forma de quantização, a compressão usando transformada de cosseno discreta é uma compressão com perdas.

Entretanto, a forma mais comum e que gera melhores resultados, é a aplicação de uma operação de quantização nos dados gerados pela transformada, e apenas o armazenamento dos dados quantizados. Essa quantização permite uma maior eficiência das técnicas de codificação e eliminação de redundância utilizada. Algumas formas de quantização normalmente utilizadas com a DCT são:

- eliminação dos componentes menos significativos - determina-se um patamar de valor ou mesmo de posição na matriz de resultados da transformada, e elimina-se ou substitui-se esses valores por 0;
- divisão inteira dos valores por um coeficiente de quantização fixo - assim pode-se usar menos dígitos, ou bits, para se representar os valores;
- divisão inteira por uma matriz de coeficientes de quantização - esta técnica é a empregada pela maioria dos padrões de compressão de dados, pois é mais flexível e permite que se ajuste a matriz a qualidade desejada da imagem.

O padrão JPEG usa esta última técnica, e a tabela de coeficientes utilizada deve ser gravada junto com o arquivo comprimido da imagem. A escolha das matrizes no padrão JPEG pode ser da seguinte forma:

1. uso das tabelas padronizadas de quantização fornecidas pelo padrão JPEG; ou

2. uso de uma tabela de quantização Q personalizada, em geral calculada com uma fórmula simples que pode ser parametrizada para melhor ou pior qualidade de imagem. Uma fórmula bem comum é a seguinte, que usa um valor inteiro R como parâmetro:

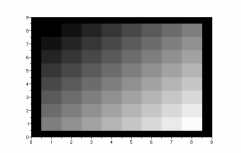
$$Q_{ij} = 1 + (i + j) \times R \tag{3.6}$$

Ainda no padrão JPEG, os coeficientes quantizados são separados (o coeficiente mais significativo de cada bloco 8x8 é separado dos demais para efeito de maior compressão) e comprimidos usando-se uma combinação de RLE (*Run Length Encoding*) e codificação de Huffman. O padrão prevê também a compressão através de uma variante das codificações aritméticas, chamada de codificação QM. Entretanto, a codificação QM, assim como a maioria das codificações aritméticas está protegida por patentes, e é preciso de uma licença do detentor das patentes para ser utilizada. Esta restrição das patentes fez com que a maioria dos compressores de JPEG utilize apenas a codificação de Huffmann, ignorando o uso do QM.

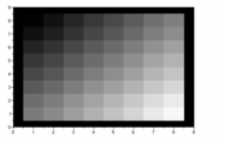
A Figura 3.5 apresenta alguns exemplos de imagens transformadas usando DCT (tamanho 8x8 *pixels*), quantizadas com a tabela recomendada pelo padrão JPEG, e destransformadas para recompor a imagem descomprimida. Note que as imagens onde as transições de tons são mais suaves proporcionam uma melhor recomposição da imagem.

Essa característica de suavizar as bordas, que pode ser notada nas imagens da Figura 3.5, é o que faz o DCT ser amplamente utilizado em compressão de fotos, pois nesse tipo de imagem, a presença de bordas e mudanças abruptas é mais rara. Para a compressão de desenhos e textos escaneados, esta técnica não é tão boa pois "borra" ligeiramente as bordas das linhas retas, como pôde ser visto nos dois últimos conjuntos de imagens.

Para demonstrar a capacidade de compressão proporcionada, usa-se a matriz que gerou a imagem em degradê e mostra-se aqui todos os passos da compressão usando DCT. Primeiro tem-se a matriz original. A seguir, pode-se ver que essa matriz possui vários valores distintos, não alcançando bons resultados apenas com a eliminação das repetições.

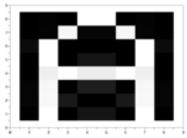


(a) Degradê cinza sem DCT

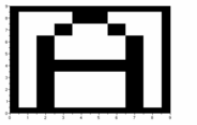


(b) Degradê cinza após DCT

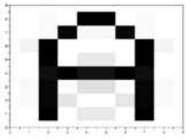
(c) Letra com fundo preto sem DCT



(d) Letra com fundo preto após DCT



(e) Letra com fundo branco sem DCT



(f) Letra com fundo branco após DCT

Figura 3.5: Efeito da DCT em imagens.

$$\begin{pmatrix} 1. & 19. & 37. & 55. & 73. & 91. & 109. & 127. \\ 19. & 37. & 55. & 73. & 91. & 109. & 127. & 145. \\ 37. & 55. & 73. & 91. & 109. & 127. & 145. & 163. \\ 55. & 73. & 91. & 109. & 127. & 145. & 163. & 181. \\ 73. & 91. & 109. & 127. & 145. & 163. & 181. & 199. \\ 91. & 109. & 127. & 145. & 163. & 181. & 199. & 217 \\ 109. & 127. & 145. & 163. & 181. & 199. & 217 & 235. \\ 127. & 145. & 163. & 181. & 199. & 217 & 235. & 253. \end{pmatrix} \quad \textbf{(1)}$$

Quando se aplica o DCT, tem-se a matriz seguinte. Esta matriz já tem vários valores zerados, que podem ser eliminados na compressão por alguma técnica de remoção de repetições, como RLE. A matriz a seguir está arredondada em duas casas decimais.

$$\begin{pmatrix} 1016. & -327.99 & 0. & -34.29 & 0. & -10.23 & 0. & -2.58 \\ -327.99 & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ -34.29 & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ -10.23 & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ -2.58 & 0. & 0. & 0. & 0. & 0. & 0. & 0. \end{pmatrix} \textbf{(2)}$$

O passo seguinte é aplicar a quantização. Nesse momento pode-se ver que o número de posições zeradas aumenta ainda mais, e os valores restantes são todos relativamente pequenos, podendo ser representados em um arquivo com número menor de bits do que os valores maiores do arquivo original:

$$\begin{pmatrix} 63. & -30. & 0. & -2. & 0. & 0. & 0. & 0. \\ -27. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ -2. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \end{pmatrix} \textbf{(3)}$$

Após a destransformação da matriz quantizada, usando IDCT, observa-se que os valores quase não mudaram em relação ao arquivo original, com a maior diferença entre eles na posição (2,8) que é de 6 (em um máximo de 256), ou seja, menos de 3%.

$$\begin{pmatrix} 4. & 18. & 39. & 57. & 74. & 93. & 113. & 128. \\ 17. & 32. & 52. & 71. & 88. & 106. & 127. & 141. \\ 37. & 52. & 72. & 91. & 107 & 126. & 146. & 161. \\ 56. & 70. & 91. & 109. & 126. & 144. & 165. & 179. \\ 73. & 87. & 108. & 126. & 143. & 161. & 182. & 196. \\ 91. & 106 & 126. & 145. & 161. & 180. & 200. & 215 \\ 111. & 125. & 146. & 164. & 181. & 200. & 220 & 235. \\ 124. & 139. & 159. & 178. & 195. & 213. & 234. & 248. \end{pmatrix} \textbf{(4)}$$

Para imagens onde as variações dos tons são graduais, a técnica de DCT mostra excelentes resultados, e por isso é adotada nos padrões mais usados hoje em dia.

O padrão MPEG usa para a compressão de áudio uma variante da DCT conhecida como MDCT (*Modified Discrete Cosine Transform*). Esta transformada é bastante parecida com a transformada de cosseno unidimensional, e sua fórmula é:

$$S_i = \sum_{k=0}^{n-1} x_k \cos\Big(\frac{\pi}{2n}\left[2k+1+\frac{n}{2}\right](2i+1)\Big), i = 0, 1, .., \frac{n}{2} - 1. \tag{3.7}$$

E a sua inversa, conhecida como IMDCT é dada por:

$$x_k = \sum_{i=0}^{n/2-1} S_i \cos\Big(\frac{\pi}{2n}\left[2k+1+\frac{n}{2}\right](2i+1)\Big), k = 0, 1, .., n-1. \tag{3.8}$$

Maiores detalhes podem ser obtidos em [Salomon 2000].

3.2.5 Técnicas de Espalhamento de Espectro

Na técnica de espalhamento de espectro (como o espalhamento de freqüência), os dados escondidos são espalhados ao longo da imagem de cobertura. Uma stego-chave é usada para selecionar randomicamente os canais de freqüência. A *White Noise Storm* é uma ferramenta popular usando esta técnica. Em outra pesquisa [Marvel et al. 1999], dados embutidos como objeto a ser transmitido, a imagem de cobertura é visualizada como interferência em um *framework* de comunicação de cobertura. Os dados embutidos são primeiramente modulados com pseudo ruído e então a energia é espalhada sobre uma faixa de freqüência larga, alcançando somente um nível muito baixo de força de inclusão. Isto é valioso para alcançar a impercepitibilidade.

3.2.6 Técnicas de Esteganografia em Vídeo

Como já foi dito anteriormente, a esteganografia tira proveito de qualquer meio ou tipo de informação para esconder uma mensagem. No mundo digital atual, há grande quantidade de áudio e vídeo circulando principalmente pela Internet. E a esteganografia tira proveito deste vasto domínio de cobertura.

Quando informações são escondidas dentro de um vídeo, normalmente é usado o

método da DCT. Sendo assim, esteganografia em vídeo é muito similar a esteganografia em imagens, exceto pelo fato de que as informações são escondidas em cada frame do arquivo de vídeo. Da mesma forma que nas imagens, quanto maior for a quantidade de informação a ser escondida no vídeo, maior será a possibilidade do método esteganográfico ser percebido.

Maiores detalhes sobre trabalhos de esteganografia em vídeos podem ser encontrados em [Langelaar et al. 1998, Qiao e Nahrstedt 1998, Kalker et al. 1999, Linnartz et al. 1999, Hartung e Girod 1996].

3.2.7 Técnicas de Esteganografia em Áudio

Esconder imagens em sinais de áudio é algo desafiante, pois o sistema auditivo humano (SAH) pode trabalhar em uma faixa muito grande de freqüências. O SAH pode captar até um bilhão de potências diferentes de sinais (altura) e até mil freqüências de sinais distintas. A sensitividade a ruído também é muito apurada. Uma perturbação em um arquivo de som pode ser detectada tão baixa quanto uma em 10 milhões de partes ou 80 dB em um ambiente comum. Apesar de ser tão poderoso para captar sinais e freqüências, o SAH não consegue fazer diferenciação de tudo que recebe. Sendo assim, sons mais altos tendem a mascarar sons mais baixos. Além disso, o SAH não consegue perceber um sinal em fase absoluta, somente em fases relativas. Também existem algumas distorções do ambiente muito comuns que são simplesmente ignoradas pelo ouvido na maioria dos casos.

As técnicas de esteganografia exploram muitas destas "vulnerabilidades" do ouvido humano, porém sempre têm que levar em conta a extrema sensibilidade do SAH. Para se desenvolver um método de esteganografia em áudio, a representação do sinal e o caminho de transmissão devem ser considerados nesta escolha. A taxa de dados é muito dependente da taxa de amostragem e do tipo de som que está sendo codificado. Um valor típico de taxa é 16 bps, mas este valor pode variar de 2 bps a 128 bps.

Outros trabalhos relacionados à esteganografia em áudio podem ser encontrados em [Boney et al. 1996, Bassia e Pitas 1998, Prandoni e Vetterli 1998, Swanson et al. 1999] [Su et al. 1999, Swanson et al. 1998, Lu et al. 2000, Li e Yu 2000, Kim 2000].

3.3 Técnicas de Esteganálise

Grande parte das técnicas de esteganografia possuem falhas ou inserem padrões que podem ser detectados. Algumas vezes, basta um agressor fazer um exame mais detalhado destes padrões gerados para descobrir que há mensagens escondidas. Outras vezes, o processo de mascaramento de informações é mais robusto e as tentativas de detectar ou mesmo recuperar ilicitamente as mensagens podem ser frustradas. A pesquisa de métodos para descobrir se há alguma mensagem escondida por esteganografia é chamada de **esteganálise**.

Recuperar os dados escondidos está além da capacidade da maioria dos testes atuais, uma vez que muitos algoritmos de mascaramento utilizam geradores aleatórios muito seguros para esconder a informação durante o processo de mascaramento. Muitas vezes, os bits são espalhados pelo objeto de cobertura. Desta forma, os melhores algoritmos de esteganálise podem não ser capazes de dizer onde está a informação, mas devem dizer se há dados escondidos.

3.3.1 Tipos de Ataques

Existem diversas abordagens para se detectar a presença de conteúdo escondido em imagens digitais. Estas abordagens podem ser divididas em três tipos, descritos a seguir:

- ataques aurais - estes ataques consistem em retirar as partes significativas da imagem como um meio de facilitar aos olhos humanos a busca por anomalias na imagem. Um teste comum é mostrar os bits menos significativos da imagem. Câmeras, *scanners* e outros dispositivos sempre deixam alguns padrões nos bits menos significativos;

- ataques estruturais - a estrutura do arquivo de dados algumas vezes muda assim que outra mensagem é inserida. Nesses casos, um sistema capaz de analisar padrões estruturais seria capaz de descobrir a mensagem escondida. Por exemplo, se mensagens são escondidas em imagens indexadas (baseadas em paletas de cores), pode ser necessário usar diferentes versões de paletas. Este tipo de atitude muda as características estruturais da imagem de cobertura, logo as chances de detecção da presença de uma mensagem escondida aumentam [Wayner 2002];

- ataques estatísticos - os padrões dos *pixels* e seus bits menos significativos freqüentemente revelam a existência de uma mensagem secreta nos perfis estatísticos. Os novos dados não têm os mesmos perfis esperados. Muitos dos estudos de Matemática

e Estatística têm por objetivo classificar se um dado fenômeno ocorre ao acaso. Cientistas usam estas ferramentas para determinar se suas teorias explicam bem tal fenômeno. Estas técnicas estatísticas também podem ser usadas para determinar se uma dada imagem e/ou som possui alguma mensagem escondida. Na maioria das vezes, os dados escondidos são mais aleatórios que os dados que foram substituídos no processo de mascaramento ou inserem padrões que alteram as propriedades estatísticas inerentes do objeto de cobertura [Westfeld e Pfitzmann 2000, Provos e Honeyman 2003, Wayner 2002].

3.3.2 Principais Técnicas de Esteganálise

A seguir, são apresentadas algumas das principais técnicas de esteganálise baseadas em ataques estatísticos existentes.

1. Esteganálise por teste do χ^2 (*Chi-Square Test Approach*).

   O teste Chi-quadrado permite verificar igualdade (semelhança) entre categorias discretas e mutuamente exclusivas (por exemplo, diferenças de comportamento entre homens e mulheres). Cada indivíduo ou item deve pertencer a uma e somente uma categoria.

   As seguintes suposições precisam ser satisfeitas:

   (a) os dois grupos são independentes;

   (b) os itens de cada grupo são selecionados aleatoriamente;

   (c) as observações devem ser freqüências ou contagens;

   (d) cada observação pertence a uma e somente uma categoria;

   (e) a amostra deve ser relativamente grande (pelo menos 5 observações em cada célula e no caso de poucos grupos (2 x 2), pelo menos 10).

   A hipótese *H0* é que não existe diferença entre as freqüências (contagens) dos grupos. A hipótese alternativa é que existe diferença. Para se testar as hipóteses é preciso testar se existe diferença significativa entre as freqüências observadas e as esperadas em cada extrato.

   Deseja-se, por exemplo, saber se existe diferença na percepção de homens e mulheres em relação a uma afirmativa feita. As categorias são homens e mulheres, e número total de mulheres é diferente do número total de homens. Cada item pertence a

uma e somente uma destas categorias. Da mesma maneira, cada indivíduo pode responder somente de uma forma. O resultado deve ser comparado com o que seria obtido se não houvesse diferença entre os grupos. Para ilustrar, supõe-se 99 homens e 99 mulheres na amostra. Neste caso, se os grupos se comportassem da mesma forma e respondessem igualmente para cada situação, o resultado seria 33 pessoas em cada célula.

Em geral os grupos não são igualmente distribuídos. O valor esperado de cada célula é uma proporção do valor total. Um caso real é apresentado na Tabela 3.1.

Tabela 3.1: Tabela exemplo para o teste χ^2.

	Homens	Mulheres	Total
Concorda	58	35	93
Neutro	11	25	36
Não concorda	10	23	33
Total	79	83	162

Os valores esperados para cada célula são obtidos multiplicando o percentual da coluna pelo total da linha, isto é, total da linha x (total coluna/total). Por exemplo: 45,35 = 93 x 79/162, conforme Tabela 3.2.

Tabela 3.2: Cálculo do χ^2.

Esperado				
		Homens	Mulheres	Total
	Concorda	45,35185	47,64815	93
	Neutro	17,55556	18,44444	36
	Não concorda	16,09259	16,90741	33
	Total	79	83	162
Chi				
		3,527434	3,357437	
		2,447961	2,329987	
		2,306632	2,195469	
Chi Tabelado =	2			

O valor de chi-quadrado para cada célula é a diferença ao quadrado entre o valor esperado e o valor medido dividido pelo valor esperado, conforme fórmula a seguir.

$$\chi^2 = \frac{(ValorEsperado - ValorMedido)^2}{ValorEsperado} \tag{3.9}$$

O chi total é a soma dos valores de cada célula. O valor de χ^2 calculado deve ser comparado com o valor de chi tabelado, quanto maior o valor de chi calculado, maior

a diferença. Para obter o valor de chi tabelado (tabela de distribuição χ^2) deve-se escolher o valor do nível de significância (alfa) adequado para a situação.

Em esteganografia, as funções de cobertura de alguns softwares, por exemplo o Ezstego [Stego e Ezstego 2007] reescrevem os bits menos significativos dos bytes sorteados para tal fim guardando seus índices. Isso gera valores modificados de bytes que somente diferem, quando diferem, no último bit. Este par de valores (iniciais e transformados) é chamado de PoVs (*Pair of Values*). Se os bits usados para escrever no bit menos significativo são igualmente distribuídos, a freqüência dos valores de cada PoV se torna igual. A idéia dos ataques estatísticos é comparar uma distribuição de freqüência teórica esperada com um histograma com algumas distribuições observadas em possíveis imagens que podem ter sido modificadas. A distribuição de freqüência teórica é obtida com o chi tabelado usando o nível de significância adequado (alfa).

Um ponto crítico é como obter a distribuição de freqüência teórica. Esta freqüência não deve ser derivada da amostra que está sendo analisada pois ela pode ter sido modificada por esteganografia. O problema é que na maioria dos casos não se tem a amostra original para comparar. Na amostra original, a freqüência teórica esperada é a média aritmética das duas freqüências de um PoV. Isso porque a função mascaramento do método esteganográfico sobrescreve os bits menos significativos e isso não muda a soma destas duas freqüências (freqüência de um PoV). A contagem dos valores de freqüência pares é transferida para o valor ímpar correspondente de freqüência em cada PoV e vice-versa. Este fato permite obter a distribuição de freqüência esperada da amostra analisada, não necessitando da original para o teste.

2. Análise RS.

   Apresentada por [Fridrich e Goljan 2002], esta técnica consiste na análise das inter-relações entre os planos de cores presentes nas imagens analisadas. A classificação é feita pontualmente, sem utilização de treinamento e é dependente do contexto da imagem analisada.

   Este é um dos métodos de detecção mais robustos disponíveis. Para análise podem ser utilizadas imagens coloridas ou em tons de cinza. Não existe distinção na profundidade de cores na imagem analisada, isto pode ser válido tanto para imagens de 8 bpp (bits por *pixel*) quanto para imagens de 32 bpp.

   As definições feitas por [Rocha 2006] ressaltam os seguintes aspectos:

   "Seja IMG a imagem testada. IMG possui MxN *pixels* e cada *pixel* tem os seus

valores dados por um conjunto P. Como exemplo, para uma imagem de 8 bpp, tem-se P = 0,....255. Então, divide-se IMG em grupos de *pixels* disjuntos G de n *pixels* adjacentes, onde G = $(x_1, ..., x_n)$.

Como exemplo, pode-se escolher grupos de n = 4 *pixels* adjacentes. Feito isso, define-se uma função de discriminação f responsável por atribuir um número real $f(x_1, ..., x_n)$ para cada grupo de *pixels* G = $(x_1, ..., x_n)$. Quanto mais aleatório for o grupo, maior o valor da função de discriminação, dada por $f(x_1, x_2, ..., x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$.

Finalmente, define-se uma operação inversível F sobre P chamada *flipping*. Por *flipping* entende-se a permutação dos níveis de cores e consiste em 2 ciclos. Assim, F tem a propriedade que F^2 = Identidade ou F(F(x)) = x para todo $x \in P$. A permutação $F_1 : 0 \Leftrightarrow 1, 2 \Leftrightarrow 3, ..., 254 \Leftrightarrow 255$ corresponde a negar o LSB de cada nível de cor. Adicionalmente pode-se definir uma função de *shifting* (deslocamento) $F_{-1} : -1 \leftrightarrow 0, 1 \leftrightarrow 2, ..., 255 \leftrightarrow 256$, ou $F_{-1}(x) = F_1(x + 1)$-1 $\forall x$.

Para completar, define-se F_0 como sendo a permutação de identidade F(x)= x $\forall x \in P$. Estas operações são utilizadas para classificar os grupos de *pixels* em três categorias diferentes R, S e U:

- grupos regulares: $G \in R \Leftrightarrow f(F(G)) > f(G)$;
- grupos singulares: $G \in S \Leftrightarrow f(F(G)) < f(G)$;
- grupos não-usáveis: $G \in U \Leftrightarrow f(F(G)) = f(G)$.

Nestas expressões, F(G) significa que a função de *flipping* F foi aplicada para os componentes do vetor G = $(x_1, ..., x_n)$. Caso seja desejado aplicar diferentes *flippings* em diferentes *pixels*, deve-se usar uma máscara M que denota quais os *pixels* devem sofrer alterações. A máscara M é uma n-tupla com valores -1, 0, 1. Define-se o grupo alterado GA como: GA = $(F_{M(1)}(x1), F_{M(2)}(x2), ..., F_{M(n)}(xn))$.

O objetivo da função F é perturbar os *pixels* de uma forma pouco significativa, tal como aconteceria no processo de mascaramento de uma mensagem."

O método descrito a seguir também é proposto por Rocha [Rocha 2006] baseado em [Fridrich e Goljan 2002]:

"Seja R_M a percentagem do número de grupos regulares em relação ao total de grupos para a máscara M. De forma similar, S_M denota o número relativo de grupos singulares. Tem-se que $R_M + S_M \leq 1$ *e* $R_{-M} + S_{-M} \leq 1$, para a máscara negativa.

A hipótese estatística para o método é que, em imagens típicas, o valor esperado de R_M é aproximadamente igual ao de R_{-M} e o mesmo é verdade para S_M *e* S_{-M}. A equação definida em $R_M \cong R_{-M}$ *e* $S_M \cong S_{-M}$, foi empiricamente comprovada [Fridrich e Goljan 2002]. A randomização do plano LSB força a diferença entre R_M *e* S_M para zero à medida que o tamanho m da mensagem escondida cresce. Depois de alterar os LSBs de 50 por cento dos *pixels* (é o que acontece quando se esconde uma mensagem aleatória em todos os *pixels*), obtém-se $R_M \cong S_M$, isto é o mesmo que dizer que a capacidade de mascaramento no plano LSB agora é zero. O fato surpreendente é que um efeito contrário acontece com R_{-M} *e* S_{-M}, sua diferença aumenta proporcionalmente ao tamanho da mensagem escondida."

Desta forma, ao analisar a imagem testada, esta provavelmente estará escondendo uma mensagem se:

- condição 1: $R_M - R_{-M} =$ i e i é muito grande;
- condição 2: $R_M - S_M =$ k e k é muito grande.

Valores muito grandes para i acontecem quando $i \geq 2,5\%$ do total de grupos. Valores muito grandes para k acontecem quando $k \geq 25\%$ do total de grupos. Um mascaramento detectável ocorre toda vez que a primeira condição é verdadeira. Caso apenas a segunda condição seja verdadeira, há apenas uma suspeita de que houve um mascaramento [Fridrich e Goljan 2002].

3. Métricas de qualidade de imagens (*Image Quality Metrics*).

   Métricas de qualidade de imagem são utilizadas, de forma geral, na avaliação de codificação de artefatos, predição de desempenho de algoritmos de visão computacional, perda de qualidade devido a inadequabilidade de algum sensor, entre outras aplicações.

   Nesta abordagem proposta por [Avcibas et al. 2001], essas mesmas métricas são utilizadas para construir um discriminador de imagens de cobertura (sem conteúdo escondido) de estego-imagens (com conteúdo escondido) através da utilização de regressão multi-variada. A classificação é feita por um discriminante linear após um certo treinamento (estabilização dos coeficientes da regressão multi-variada).

4. Métricas de tons contínuos e análise de pares de amostragem (*Continuous Tone Metrics and Sample Pair Analysis*).

Esta abordagem [Dumitrescu e Wu 2002] consiste em analisar as relações de identidade estatística existentes sobre alguns conjuntos de *pixels* considerados. As identidades observadas são muito sensíveis ao mascaramento LSB e as mudanças nestas identidades podem indicar a presença de conteúdo escondido.

3.4 Aplicações

Em atividades militares, a descoberta de comunicações secretas pode levar a um ataque imediato do inimigo. Mesmo com a criptografia, a simples detecção do sinal é fatal pois descobre-se não somente a existência de inimigos como também a sua posição. Unindo o conceito de ocultamento de informação com técnicas como modulação em espalhamento de espectro torna-se mais difícil de os sinais serem detectados ou embaralhados pelo inimigo.

Várias técnicas relacionadas a ocultamento de informação levam em consideração sistemas com níveis de segurança. Em uma rede de computadores militares existem vários níveis de segurança. Um vírus ou um programa malicioso se propaga dentro do sistema passando de níveis de segurança inferiores para os superiores. Uma vez que alcança seu objetivo, tenta passar informações sigilosas para setores de nível de segurança menores. Para isso, ele se utiliza de técnicas de ocultamento para esconder informações confidenciais em arquivos comuns de maneira que o sistema lhe permita ultrapassar níveis de segurança diferentes.

Existem situações onde se deseja enviar uma mensagem sem que seja possível descobrir quem a enviou. Geralmente, esse tipo de situação é mais uma característica de atividades ilegais onde os criminosos envolvidos não desejam ser descobertos se sua mensagem for rastreada. Entretanto, essa situação também tem aplicações em atividades legais onde se deseja que a privacidade do remetente seja mantida. Alguns exemplos dessas situações são: registros médicos ou votações online.

Um tema importante a ser considerado pelo criador das técnicas de ocultamento de informação é a ética. Assim como os conhecimentos apresentados podem ser usados para garantir privacidade de dados médicos, votações ou prover serviços online com segurança, algumas pessoas podem encontrar meios de se aproveitar das vantagens dessa 'comunicação invisível e não rastreável' para executar ações ilícitas como difamações, chantagens ou seqüestros. É um dever dos desenvolvedores de sistemas de ocultamento de informação prestar atenção aos abusos que podem ocorrer.

Existe também grandes aplicações na área da indústria médica no que diz respeito

a imagens médicas. Normalmente, é usada uma forma de comunicação padrão chamada DICOM (*digital imaging and communications in medicine*) que separa a imagem das informações relativas ao paciente e ao exame como o nome, data e o médico. Em alguns casos, a ligação entre os dados e a imagem é perdida. Então, se as informações fossem ocultadas dentro da própria imagem, não haveria risco de a imagem se separar dos dados. Ainda não existem pesquisas aprofundadas sobre o efeito que tais inserções de dados na imagem poderiam causar alteração na precisão do diagnóstico. Estudos recentes na área de compressão de imagens médicas revelam que tais procedimentos não atrapalham, o que pode indicar uma certa robustez do diagnóstico a pequenas alterações como as causadas pelas técnicas de ocultamento de informação [Filho et al. 2005].

Em alguns casos, se deseja monitorar um dado arquivo com direitos autorais que está sendo distribuído na Internet, por exemplo. Para isso, utiliza-se um programa robô que procura em *sites* a divulgação desses arquivos. Ele baixa os arquivos, tenta retirar qualquer informação que possa estar escondida e compara com a informação do arquivo original. Caso as informações sejam compatíveis, sabe-se que o arquivo está sendo distribuído de maneira ilegal. O mesmo pode ser feito com músicas tocadas em transmissões via rádio. O programa procura no sinal do rádio marcas inseridas propositalmente nas músicas a serem protegidas. Se em um dado momento a marca é inteiramente decodificada do sinal, sabe-se que a estação de rádio investigada tocou a música sem autorização.

Pode-se também inserir pedaços de informações dentro dos dados que estão sendo transmitidos para que o público que a receba possa usar. Como exemplo, pode-se ter informações de um dado produto anunciado por uma rádio onde o cliente, com um simples apertar de botão, pode descobrir o preço, local de venda mais próximo ou fabricante. Essas informações são enviadas sem a necessidade de se usar outra banda para transmissão pois ela é inserida no próprio sinal de rádio sem prejudicar a qualidade do mesmo.

Outra aplicação seria inserir uma forma de indexação de músicas a serem armazenadas no banco de dados de uma estação de rádio para que elas sejam acessadas de maneira mais fácil. Pode-se inserir também dados da transmissão como país de origem, autor e produtora.

Atualmente a esteganografia tem sido também explorada em ramos de sistemas de detecção de intrusão [Sieffert et al. 2004] e em sistemas de arquivos [Hirohisa 2007].

Outras aplicações de esteganografia incluem as técnicas de autenticação, criptografia e rastreamento de documentos, que por serem utilizadas normalmente em conjunto com a técnica de marca d'água, são mencionadas a seguir.

3.4.1 Marcas D'Água

O grande crescimento dos sistemas de multimídia interligados pela rede de computadores nos últimos anos apresenta um enorme desafio nos aspectos tais como propriedade, integridade e autenticação dos dados digitais (áudio, vídeo e imagens estáticas). Para enfrentar tal desafio, o conceito de marca d'água digital foi definido.

Uma marca d'água é um sinal portador de informação, visualmente imperceptível, embutido em uma imagem digital. A imagem que contém uma marca é dita imagem marcada ou hospedeira. Apesar de muitas técnicas de marca d'água poderem ser aplicadas diretamente para diferentes tipos de dados digitais, as mídias mais utilizadas são as imagens estáticas.

Existe uma certa confusão entre as marcas d'água imperceptíveis e as visíveis utilizadas em cédulas de dinheiro, por exemplo. As visíveis são usadas em imagens e aparecem sobrepostas sem prejudicar muito a percepção da mesma. São usadas geralmente para que se possa expor imagens em locais públicos como páginas na Internet sem o risco de alguém copiá-la e usá-la comercialmente, pois é difícil remover a modificação sem destruir a obra original. É possível também inserir digitalmente marcas visíveis em vídeo e até audíveis em música.

3.4.1.1 Marcas Robustas e Frágeis

As marcas d'água digitais são classificadas, de acordo com a dificuldade em removê-las, em robustas, frágeis e semifrágeis. Esta classificação também normalmente determina a finalidade para a qual a marca será utilizada.

As marcas robustas são projetadas para resistirem a maioria dos procedimentos de manipulação de imagens. A informação embutida em uma imagem através de uma marca robusta poderia ser extraída mesmo que a imagem hospedeira sofresse rotação, mudança de escala, mudança de brilho/contraste, compactação com perdas com diferentes níveis de compressão, corte das bordas (*cropping*), etc. Uma boa marca d'água robusta deveria ser impossível de ser removida, a não ser que a qualidade da imagem resultante deteriore a ponto de destruir o seu conteúdo visual. Isto é, a correlação entre uma imagem marcada e a marca robusta nela inserida deveria permanecer detectável mesmo após um processamento digital, enquanto a imagem resultante do processamento continuar visualmente reconhecível e identificável como a imagem original. Por esse motivo, as marcas d'água robustas são normalmente utilizadas para a verificação da propriedade (*copyright*) das

imagens. Apesar de muitas pesquisas, parece que ainda não foi possível obter uma marca d'água robusta realmente segura.

As marcas frágeis são facilmente removíveis e corrompidas por qualquer processamento na imagem. Este tipo de marca d'água é útil para checar a integridade e a autenticidade da imagem, pois possibilita detectar alterações na imagem. Em outras palavras, uma marca d'água frágil fornece uma garantia de que a imagem marcada não seja despercebidamente editada ou adulterada. As marcas frágeis de autenticação detectam qualquer alteração na imagem. Às vezes, esta propriedade é indesejável. Por exemplo, ajustar brilho/contraste para melhorar a qualidade da imagem pode ser um processamento válido, que não deveria ser detectado como uma tentativa de adulteração maliciosa. Ou então, compactar uma imagem com perdas (como JPEG ou JPEG2000) em diferentes níveis de compressão deveria ser uma operação permitida. Ainda, imprimir e escanear uma imagem não deveria levar à perda da autenticação. Assim, foram criadas as marcas d'água semifrágeis.

Uma marca semifrágil também serve para autenticar imagens. Diferentemente, estas procuram distinguir as alterações que modificam uma imagem substancialmente daquelas que não modificam o conteúdo visual da imagem. Uma marca semifrágil normalmente extrai algumas características da imagem que permanecem invariantes através das operações "permitidas" e as insere de volta na imagem de forma que a alteração de uma dessas características possa ser detectada.

3.4.1.2 Tipos de Marcas de Autenticação

Pode-se subdividir as marcas de autenticação (tanto frágeis como semifrágeis) em três subcategorias: sem chave, com chave secreta (cifra simétrica) e com chave pública/privada (cifra assimétrica).

Uma marca de autenticação sem chave é útil para detectar as alterações não intencionais na imagem tais como um erro de transmissão ou de armazenamento. Funciona como uma espécie de *checksum*. Se o algoritmo de autenticação sem chave estiver disponível publicamente, qualquer pessoa pode inserir este tipo de marca em qualquer imagem e qualquer pessoa pode verificar se uma imagem contém uma marca válida.

A marca de autenticação com chave secreta (cifra simétrica) é usada para detectar uma alteração que pode ser inclusive intencional ou maliciosa. Este tipo de marca é similar aos códigos de autenticação de mensagem, sendo que a única diferença é que o código de autenticação é inserido na imagem ao invés de ser armazenado separadamente.

Os algoritmos para inserção e detecção deste tipo de marca podem ser disponibilizados publicamente, e uma chave secreta é usada em ambas as fases.

As marcas de autenticação com chave pública (cifra assimétrica) utilizam a criptografia de chave pública para inserir uma assinatura digital na imagem. Usando uma cifra de chave pública, a autenticidade de uma imagem pode ser julgada sem a necessidade de se tornar pública qualquer informação privada.

3.4.1.3 Marca de Autenticação em Imagens de Tonalidade Contínua e Imagens Binárias

Existe uma forma "natural" de embutir as marcas de autenticação em imagens de tonalidade contínua (*contone*) não compactadas, que é inserir os dados nos bits menos significativos. Alterar os LSBs afeta muito pouco a qualidade da imagem, ao mesmo tempo em que se conhece exatamente os bits que serão afetados pela inserção da marca.

Não ocorre o mesmo com as imagens binárias, onde cada *pixel* consiste de um único bit, de forma que não existe LSB. Isto traz dificuldades especiais para projetar marcas de autenticação para este tipo de imagem. Inserir uma marca de autenticação em imagens *contone* compactadas com perdas também apresenta dificuldades especiais.

Entre os três tipos de marca de autenticação, a de chave pública é a que oferece mais recursos. Alguns possíveis usos de uma marca de autenticação de chave pública são mostrados a seguir:

- câmera digital segura - costuma-se citar o artigo de [Friedmann 1993] como o trabalho que inspirou os primeiros trabalhos de marca d'água de autenticação. A câmera digital proposta produz dois arquivos de saída para cada imagem capturada: a primeira é a própria imagem digital capturada pela câmera em algum formato; e a segunda é uma assinatura digital produzida aplicando a chave privada da câmera (que deve estar armazenada de forma segura em um circuito integrado dentro da câmera). O usuário deve tomar cuidado para guardar os dois arquivos, para que se possa autenticar a imagem mais tarde. Uma vez que a imagem digital e a assinatura digital são geradas pela câmera e armazenadas no computador, a integridade e a autenticidade da imagem pode ser verificada usando um programa para decodificar a assinatura digital, que pode ser distribuído livremente aos usuários. O programa de verificação recebe como entrada a imagem digital, a assinatura digital e a chave pública da câmera. Ele calcula a função *hash* da imagem digital, decriptografa a

assinatura digital e verifica se as duas impressões digitais obtidas são iguais. O esquema proposto por Friedman poderia ser melhorado de duas formas. A primeira seria embutir a assinatura digital no arquivo da imagem, o que eliminaria a necessidade de armazenar dois arquivos para cada imagem. Alguns formatos de imagem permitem armazenar alguns dados adicionais no cabeçalho ou rodapé do arquivo. Mas o mais interessante seria embutir a assinatura digital na própria imagem. A segunda seria permitir a localização da região alterada. Isto poderia ser interessante, por exemplo, para descobrir a intenção do falsificador ao adulterar a imagem. A marca d'água de autenticação de chave pública pode ser usada para incorporar essas melhorias à câmera de Friedman;

- autenticação de imagens distribuídas pela rede - uma agência de notícias necessita distribuir pela Internet uma fotografia jornalística, com alguma prova de autenticidade de que a foto foi distribuída pela agência e que ninguém introduziu alterações maliciosas na foto. A agência pode inserir uma marca d'água de autenticação na imagem e distribui a foto marcada;

- FAX confiável - uma "máquina de FAX confiável" poderia conter internamente uma chave privada e inserir uma marca d'água em todos os documentos transmitidos por ela. O receptor de FAX, usando a chave pública da máquina transmissora, poderia verificar que o documento foi originado de uma máquina específica de FAX e que o documento não foi manipulado.

3.4.1.4 Extração de Marca D'água

Com relação a extração da marca d'água, tem-se três tipos de sistemas diferentes. Cada um deles se diferencia pela sua natureza ou combinação de entradas e saídas:

- marca d'água privada (também chamada de "não-cega") - esse sistema requer a marca d'água original. Dentro desse esquema, existem 2 tipos. No primeiro, é necessário o arquivo original para achar pistas de onde se localiza a marca dentro do arquivo marcado. O sistema do segundo tipo necessita das mesmas informações do anterior, mas ele somente tenta responder a seguinte pergunta: o arquivo contém a marca d'água? Sim ou não?. Espera-se que este sistema seja mais robusto já que transporta pouca informação e requer acesso a dados secretos;

- marca d'água semiprivada ou semi-cega - diferente da anterior, não utiliza o arquivo original na extração. Entretanto, tenta responder a mesma questão. Algumas

aplicações onde se poderia utilizar esse esquema seria para provar a propriedade em côrte ou em mecanismos de controle de cópia como em aparelhos de DVDs. No último caso, a chave poderia ser guardada dentro da memória do DVD e qualquer disco que fosse colocado no aparelho somente poderia ser decodificado se a marca d'água pudesse ser extraída dos dados de vídeo do anterior. Como não se pode colocar os dados originais de todos os possíveis vídeos a serem usados no aparelho, não se pode usar o esquema de marcas d'água privadas descrito no item anterior;

- marca d'água pública ou cega - não requer nem o arquivo original nem a marca. A intenção do esquema é tentar retirar a marca do dado sem pistas de onde ele se localiza ou como ele seria.

Um estudo sobre diversos algoritmos de marca d'água está descrito em [Meerwald 2001].

Para complementar, a tabela 3.3 apresenta um comparativo indicando o objetivo, as especificações e os detalhes de detecção e extração dos algoritmos de marca d'água e esteganografia .

Tabela 3.3: Tabela comparativa entre Esteganografia e Marca D'água [Wang e Wang 2004].

	Exigências	Marca d'agua		Esteganografia
		Privado	Público	
Objetivo	Proteção de direitos de propriedade intelectual	++++		–
	Transmissão da mensagem secreta sem levantar suspeita	–		++++
Especificações	Invisibilidade Perceptível	++++		+++++
	Estatístico ou Algoritmo de Invisibilidade	+		+++++
	Robustes em relação à remoção, destruição, ou falsificação maldosa	+++++		–
	Resistência em relação ao processo de um sinal normal	++++		+
	Capacidade de resistência a compressão comum	++++		++
	Alto Custo	++		++++
Detecção/ Extração	Extração/Detecção sem objeto inserido	–	++++	++++
	Extração somente com presença do objeto inserido	++++	–	–
	Exigência de baixa complexidade na extração/detecção	++		+++
	capacidade opcional de download automático do objeto	+		++
Nota: Crucial: +++++ necessário: ++++ importante: +++ desejável: ++ útil: + desnecessário ou irrelevante: –				
Os esquemas públicos de marca d'água não necessitam do objeto original na detecção/extração; os esquemas confidenciais requerem a presença do original.				

3.4.2 Aplicativos Existentes

Atualmente as redes de computadores provêem um canal de fácil utilização para a esteganografia. Vários tipos de arquivo podem ser utilizados como imagem de cobertura incluindo imagens, sons, texto e até executáveis. Por isso é grande o número de aplicativos já criados para tentar usar esta facilidade. Por outro lado, existem também alguns softwares de esteganálise que tentam localizar os dados embutidos nas diversas mensagens

de cobertura. Tais aplicações podem ser encontradas na Internet e funcionam em várias plataformas, desde o DOS, Windows passando por MAC/OS até o Unix/Linux. Esta subseção apresenta alguns exemplos destes softwares e seu funcionamento.

As ferramentas *Ezstego* e *Stego Online* [Stego e Ezstego 2007] foram desenvolvidas em Java por Romana Machado e limitadas a imagens indexadas de 8bits no formato GIF. Outra aplicação em Java fácil de usar é o *Revelation* [Revelation 2007], que esconde arquivos em imagens de cobertura no formato *bitmap* de 24 bits. Por serem escritas em Java as ferramentas são altamente portáveis, podendo ser executadas em Linux, Unix, Windows e MAC/OS.

O *Hide and Seek* [Hide e Seek 2007] foi desenvolvido por Colin Maroney e é capaz de inserir uma lista de arquivos em uma imagem no formato JPEG. A ferramenta porém não faz uso de criptografia. Uma outra ferramenta parecida chamada *Jphide and Seek* [Jphide e Seek 2007] desenvolvida por Allan Latham, contém na verdade dois arquivos: *jphide.exe* e *jpseek.exe*. O primeiro esconde a mensagem em um arquivo JPEG e o segundo extrai a mensagem. O programa utiliza criptografia de chave simétrica e o usuário é obrigado a fornecer uma *pass phrase*.

O Jphide and Seek também é de simples operação. A opção *Hide* esconde o dado embutido (arquivo de entrada) na imagem de cobertura com formato JPEG. É interessante notar que o aplicativo analisa a imagem de cobertura e diz qual o tamanho máximo que o arquivo de entrada deve ter para que o processo seja seguro. A opção *Seek* decodifica o dado embutido usando a imagem de cobertura e o salva em um arquivo de saída.

Niels Provos desenvolveu o *Outguess* [Outgess 2007]. Este software se propõe a melhorar o passo da codificação da imagem JPEG através de um gerador pseudo-randômico de números. Os coeficientes da transformada de cosseno são escolhidos também de maneira randômica para serem substituídos pelos número gerados aleatoriamente. O bit menos significativo dos coeficientes selecionados é substituído pela mensagem cifrada. Testes estatísticos de primeira ordem não são capazes de detectar mensagens mascaradas com o *Outguess*. O pseudo-código gerado a partir do Outguess é apresentado na Figura 3.6.

Os softwares de esteganálise se dispõem a descobrir se os arquivos usados como mensagem de cobertura contém algum dado embutido e se possível identificar o software utilizado no processo de esteganografia. Um destes softwares é o *StegSpy* [StegSpy 2007], que permite a identificação de um arquivo que serve como mensagem de cobertura. O programa detectará a esteganografia e o software utilizado para esconder o dado embutido. A versão atual do software também identifica a localização da mensagem embutida

```
Input: message, shared secret, cover image
Output: stego image
initialize PRNG with shared secret
while data left to embed do
   get pseudo-random DCT coefficient from cover image
    if DCT !=  0 and DCT != 1 then
      get next LSB from message
      replace DCT LSB with message LSB
    end if
    insert DCT into stego image
end while
```

Figura 3.6: Pseudo-código do OUTGUESS [Provos e Honeyman 2003].

dentro do arquivo de cobertura. O *StegSpy* atualmente identifica os programas Hiderman, JPHide and Seek, Masker, JPegX e Invisible Secrets.

Outra ferramenta de esteganálise é o *StegDetect* [Stegdetect 2007] que foi desenvolvida pelo mesmo autor do Outguess (Niels Provos). Este software se propõe a detectar conteúdo esteganográfico gerado pelo softwares Jsteg, JP Hide and Seek, Invisible Secrets, versões mais antigas do Outguess, F5, AppendX, e Camouflage. A versão mais atual do StegDetect suporta análise discriminante linear (LDA) para detectar qualquer estego sistema.

3.5 Conclusão

Tanto a esteganografia quanto a marca d'água descrevem técnicas que são usadas na intenção de ocultar uma comunicação dentro de uma informação disfarce. Entretanto, esteganografia se refere tipicamente a uma comunicação ponto-a-ponto. Por isso, o método geralmente não é robusto contra modificações ou tem somente uma robustez limitada que a protege de pequenas alterações que possam ocorrer em termos de transmissão, armazenamento, mudanças de formato, compressão ou conversões digital-analógicas.

Em marcas d'água, por outro lado, o foco está na robustez. Não existe comunicação ponto-a-ponto, mas deseja-se que a marca inserida em um dado seja recuperada de algum modo depois da imagem circular por quaisquer canais típicos da aplicação. Por exemplo, pode-se marcar uma imagem que deseja-se proteger contra cópias sem autorização. Caso alguém a copie e utilize técnicas de processamento de imagem para tentar apagar a marca, ainda assim deve ser possível decodificar a marca da imagem alterada. Isso provaria quem é o verdadeiro autor ou proprietário da imagem. A questão da detecção não é tão importante, apesar de que, se o observador não perceber a marca, ele talvez nem tente removê-la.

Um exemplo de aplicação oposta seria marcar uma imagem para verificar se ela sofrerá alterações. Caso a imagem seja modificada de alguma forma, a marca será destruída, mostrando que o ato realmente aconteceu. A robustez ou a ausência dela define a aplicação da marca utilizada. As marcas d'água robustas devem resistir a ataques e alterações na imagem. As marcas frágeis devem ser destruídas caso a imagem sofra alterações.

Atualmente existem estudos para proteger a esteganografia das técnicas de esteganálise. Em [Provos 2001]são apresentados novos métodos que permitem esconder mensagens de forma segura e resistentes a análise estatística.

Técnicas esteganográficas têm uso legal e ilegal. Como uso legal no presente e no futuro, esteganografia tem sido usada e será cada vez mais utilizada na proteção de direitos intelectuais, principalmente quando se considera as novas formas de comercialização utilizando mídia digital. Neste sentido as técnicas de marca d'água parecem ser um campo profícuo de pesquisa e aplicações no futuro.

Por outro lado, há o uso ilegal de técnicas esteganográficas, que cresce cada vez mais, em virtude da facilidade de acesso a Internet. Usar esteganografia para transitar mensagens ou até pequenas imagens de pornografia ou pedofilia é possível e provável. Um relatório de crimes de tecnologia [The High Technology Crime Advisory Committee 2007] lista alguns tipos de crime comuns utilizando alta tecnologia:

- comunicações criminosas;
- fraudes;
- *hacking*;
- pagamentos eletrônicos;
- pornografia e pedofilia;
- ofensas a propriedade intelectual;
- propagação de vírus e cavalos de tróia.

Um exame preliminar desta lista mostra vários casos de mau uso da esteganografia, principalmente no que se refere à comunicação criminosa. Em termos de segurança da informação há também outras áreas de interesse. Uma área com uso potencial em várias aplicações é o desenvolvimento de protocolos que usam esteganografia para burlar censura. Em [Haselton 2000], o coordenador da organização peacefire.org, uma organização que

se opõe à censura na Internet a menores de 18 anos, descreve um protocolo que seria "indetectável" por sensores.

Há também a possibilidade de ataques de vírus utilizarem técnicas de esteganografia. As técnicas e ferramentas esteganográficas podem ser utilizadas em conjunto com outras aplicações para automaticamente extrair informações escondidas sem a intervenção do usuário. Um cenário possível para um ataque de vírus poderia ser o envio de uma mensagem escondida em uma imagem enviada por e-mail. Um cavalo de tróia instalado na máquina poderia então extrair o vírus da imagem e infectar várias máquinas.

A esteganografia, quando bem utilizada, fornece meios eficientes e eficazes na busca por proteção digital. Associando criptografia e esteganografia, as pessoas têm em mãos o poder de comunicar-se em segredo pela rede mundial de computadores mantendo suas identidades íntegras e secretas.

Capítulo 4

Modelo de Arquitetura Proposto

Neste capítulo descreve-se o modelo proposto para uma arquitetura de IDS, onde toda a troca de mensagens está baseada em dois pontos principais: a *esteganografia* como forma de transmissão de mensagens e a *reputação* para ponderar as mensagens de alertas recebidas dos IDSs vizinhos, que tem como base a teoria dos jogos aplicada a redes *ad hoc*.

A arquitetura proposta permite que cada nó participante do modelo de detecção de intrusão funcione independentemente de seus vizinhos. A troca de alertas para a formação de uma opinião global sobre uma ou mais detecções locais é um dos principais pontos e utiliza a esteganografia como método para o envio dos alertas, utilizando-se ainda de um mecanismo de reputação para ponderar os alertas recebidos. Caso um nó não tenha boa reputação, seus alertas são desprezados. A reputação é calculada em função das interações entre os nós, principalmente repasse de pacotes (roteamento), onde um mecanismo monitora os repasses dos pacotes de um vizinho, determinando assim se ele está colaborando com a rede como um todo ou não.

O modelo de arquitetura proposto, bem como sua estrutura e funcionalidades, é apresentado na Figura 4.1.

- *Coleta de Dados Local*: Coleta dados de sensores locais, como registros de atividades locais (acesso não autorizado), atividades de comunicação (varredura de portas, DoS, falhas de repasse de pacotes) e outros dados locais que possam formar evidências de ataques. Além disso, envia os dados coletados para a *Máquina de Detecção Local*;

- *Máquina de Detecção Local*: Os dados colhidos pela *Coleta de Dados Local* são repassados à *Máquina de Detecção Local* para análise e agrupamento dos mesmos,

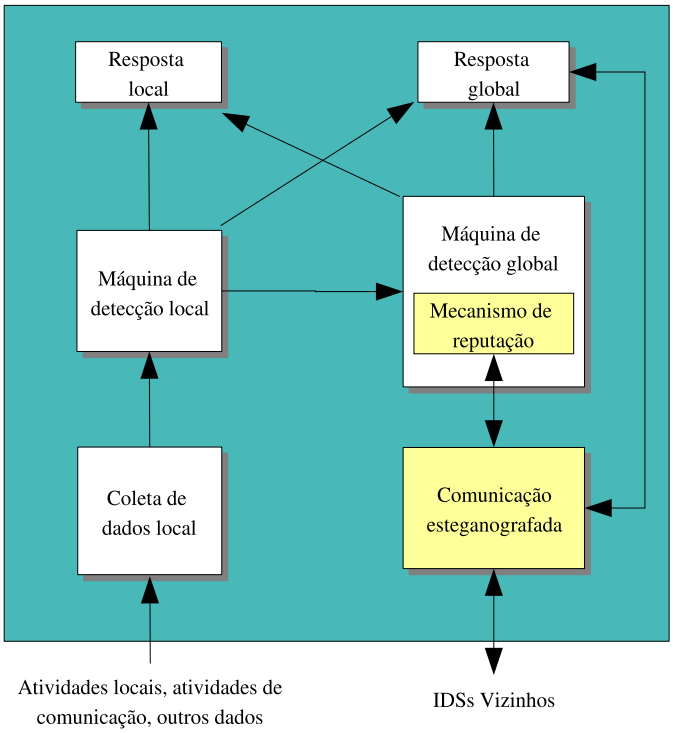


Figura 4.1: Arquitetura proposta

gerando informações sobre ataques. Essas informações são repassadas à *Máquina de Detecção Global*, para que sejam verificadas em conjunto com informações recebidas de outros IDSs vizinhos. Caso um ataque seja detectado localmente, a *Máquina de Detecção Local* repassa as informações para a *Resposta Local* e para a *Resposta Global*, para que uma providência sobre o atacante seja tomada (exclusão do nó do roteamento ou bloqueio de pacotes vindo do nó atacante, por exemplo) localmente e globalmente. Também é responsável por recolher o comportamento dos vizinhos, principalmente na questão de repasse de pacotes (*watchdog*). Assim, o *Subsistema de Reputação* é alimentado, recalculando a reputação desses vizinhos;

- *Resposta Local*: Caso um ataque seja detectado localmente pela *Máquina de Detecção Local* ou globalmente pela *Máquina de Detecção Global*, a *Resposta Local* é acionada para executar uma ação local a respeito do atacante, como por exemplo a exclusão do nó do roteamento ou o bloqueio de pacotes oriundos desse nó;

- *Resposta Global*: No caso de um ataque ser detectado localmente, a *Máquina de Detecção Local* gera um alerta de invasão e o envia à *Resposta Global* para que os demais IDSs da rede sejam avisados, gerando assim uma *Resposta Global*. O mesmo

acontece com a *Máquina de Detecção Global*, que pode repassar alertas globais gerados por outros nós;

- *Máquina de Detecção Global com Subsistema de Reputação*: É responsável por analisar os dados recebidos via comunicação esteganografada e ponderá-los, com base na reputação do nó que enviou o alerta. Assim, mesmo que uma máquina que enviou o alerta seja comprometida, os alertas enviados por ela são ponderados em cada máquina, individualmente. Nesse caso, a máquina comprometida tem uma reputação baixa perante as demais, tendo então seus alertas ignorados. Além disso, é responsável por ativar as respostas, local e global, caso um ataque seja detectado globalmente;

- *Comunicação Esteganografada*: Subsistema utilizado para a comunicação com os demais IDSs da rede, responsável por esteganografar e enviar as mensagens de alertas ou consultas sobre reputação de outros nós, assim como receber e desesteganografar as mensagens recebidas, repassando-as para a *Máquina de Detecção Global* para serem analisadas pelo *Subsistema de Reputação*. A *Resposta Global* também envia alertas para a *Comunicação Esteganografada* informando aos demais IDSs sobre invasões. Nesse subsistema também são comprimidos os alertas antes de serem esteganografados e enviados e descomprimidos após a desesteganografia no destino.

As mensagens trocadas entre os agentes de IDS utilizam o formato IDMEF (*Intrusion Detection Message Exchange Format*) [Curry e Debar 2002], padrão este formalizado pelo IETF (*The Internet Engineering Task Force*) e baseado em XML (*Extensible Markup Language*). Criado com a intenção de ser um padrão na troca de mensagens de detecção de intrusão, o modelo aqui proposto utiliza este formato com o intuito de se tornar um sistema interoperante com outros IDSs baseados neste mesmo padrão.

4.1 Esteganografia

Nesse trabalho, a esteganografia tem o papel de ocultar os alertas no padrão IDMEF em imagens JPEG. Essas imagens contendo os alertas são então enviadas para os outros IDSs da rede *ad hoc*. O uso de imagens como objeto de cobertura no sistema esteganográfico da proposta não é obrigatório, podendo ser utilizados outros objetos como áudio e vídeo, entre outros, conforme visto no Capítulo 3.

Em [Provos e Honeyman 2003] é exemplificado o uso de uma chave compartilhada para a criptografia e decriptografia, porém, neste trabalho não é utilizada nenhuma técnica de criptografia em cooperação à esteganografia, uma vez que se deseja a maior independência possível dos nós e a minimização do custo aplicado ao envio das mensagens de alerta. Um esquema criptográfico empregado em conjunto com a esteganografia implica no uso de PKI para distribuição de chaves.

[Johnson e Jajodia 1998] mostrou que as técnicas de esteganografia utilizando imagens baseadas em paletas, como no caso das BMP, são facilmente detectadas, exatamente pelas distorções geradas com a inserção dos dados.

Para a criação de cada cor em imagens JPEG, são utilizados grupos divididos em blocos de 8 x 8 pixels, por meio da transformada discreta de cosseno formando assim a geração do coeficiente 64 DCT. A DCT ocupa mais espaço do que a matriz original, pois, enquanto os valores são do tipo *byte* ou inteiro, os coeficientes da DCT são reais. Para reduzir esses valores, é realizada a quantização, que é simplesmente o processo de reduzir o número de *bits* necessários para armazenar um valor, reduzindo à precisão de um inteiro. Transformando-se o LSB do coeficiente DCT, tem-se modificações em todos os 64 *pixels* da imagem, resultando no padrão de ocultação em imagens por meio da esteganografia.

Em alguns formatos de imagens, como o GIF, existe um prejuízo com a mudança do LSB, tornando-as, muitas vezes suscetíveis a ataques devido a perda de qualidade. Isto acontece por trabalharem no domínio espacial, sem a conversão por meio da DCT. Logo, o mesmo não acontece com imagens que trabalham no domínio de freqüência, com DCT, como no caso das imagens JPEG.

A técnica da transformada discreta de cosseno utiliza como aliada a compressão, configurando-se como uma das técnicas mais eficientes de mascaramento de informações em imagens. Neste trabalho foram utilizadas imagens JPEG, por serem comuns (principalmente na Internet) e aderirem bem aos sistemas esteganográficos, como no caso das ferramentas *steghide* e *outguess* que são usadas para validação da proposta.

4.2 Compressão de Alertas para Economia de Energia

Conforme [Tsiftes 2007], softwares de compressão são de grande valia na diminuição considerável do tamanho de vários tipos de arquivos, especialmente texto (alertas no formato IDMEF são escritos em XML). Como se sabe, a troca de dados em uma rede sem fio *ad*

hoc é algo muito custoso, devido a baixa autonomia das baterias dos dispositivos móveis hoje existentes, e é visando a redução do consumo de energia que este trabalho apresenta a relação de consumo equivalente à compressão e à descompressão de arquivos similares aos alertas gerados pelos IDS [Barr e Asanović 2003, Xu et al. 2003, Tsiftes 2007]. Nos experimentos de [Tsiftes 2007] foram comprovadas as reduções de até 50% no consumo de energia despendidos para a transmissão por rádio. O compressor *gzip* [Jean-loup e Mark 2003] foi o que melhor se portou nos testes de compressão e descompressão. O *gzip* utiliza o algoritmo DEFLATE [Deutsch 1996], que é uma combinação entre o código de Huffman e a janela deslizante de Ziv-Lempel.

A ferramenta de compressão e descompressão, *gzip*, é uma entre várias outras comparadas em [Barr e Asanović 2003]. São apresentados softwares baseados nos mais diversos algoritmos de compressão e se conclui que nem sempre aquele que é mais eficaz na compressão é o que apresenta o menor consumo de energia. Após a simulação em um hardware específico, similar a um *handheld* iPAQ da Compaq, conclui-se que o algoritmo LZ77, base para o *zlib*, biblioteca principal de *gzip*, tem um melhor custo/benefício, conseguindo uma boa taxa de compressão (não a melhor) e um baixo consumo de energia.

O algoritmo de compressão (DEFLATE), apresenta níveis que variam de 1 a 9, onde o valor mais alto representa um maior nível de compressão. O *gzip* utiliza por padrão o nível de compressão 6, podendo ser configurado para utilizar os níveis entre 1 e 9, ao invés do nível padrão. Em [Barr e Asanović 2003] foram efetuados testes com os níveis 1, 6 e 9. Os resultados mostram que o o nível 1 é o mais rápido e consome menos energia, sendo ainda assim eficiente na compressão.

Sabe-se também que, em geral, a descompressão é uma ação menos custosa à energia do nó em relação à compressão. Há ainda a possibilidade de se utilizar tipos de compressão e descompressão distintos nas duas pontas (origem e destino), conforme testes realizados em [Barr e Asanović 2003] e considerado por seus autores muito eficaz, principalmente em ambientes heterogêneos, onde estações de trabalho com poder de processamento maior e sem exigências de energia podem usar algoritmos de compressão mais eficientes e conseqüentemente mais custosos. Em [Xu et al. 2003] foi identificado que ótimos compressores, até mesmo o *gzip*, podem despender um consumo desnecessário de energia ao se comprimir arquivos muito pequenos, isso se deve ao fato de se ter somado ao envio dos dados, um consumo de energia extra com uma compressão insignificante. A solução para excluir este consumo extra com a energia é a avaliação do tamanho do arquivo e sendo este considerado um arquivo pequeno é então enviado sem compressão,

de forma direta.

A compressão dos dados a serem trocados dentro da rede é uma solução viável para a troca de alertas entre os IDS por meio da esteganografia. Sabe-se que para cada 1 bit transmitido por uma rede sem fio, têm-se o consumo de energia 1000 vezes maior em relação ao mesmo processamento em uma computação simples de 32 bits. Se os dados a serem trocados pelos IDS puderem ser reduzidos de forma que se tenha um menor volume desses dados, tem-se um valor ideal, onde a diminuição do consumo de energia é motivante e viável. Para que a compressão dos dados seja válida, é necessário que: $\frac{c}{n-m} < w$, onde n bits é o resultado da compressão de m bits, então $n > m$, c é o custo da compressão e descompressão e w é o custo da transmissão e recepção.

Para o envio dos alertas do modelo de IDS proposto neste trabalho, emprega-se a compressão por meio do *gzip*, utilizando o nível 1 de compressão (mais rápido). Com a compressão dos alertas antes mesmo se serem esteganografados, espera-se uma diminuição do consumo de energia para envio e recebimento destes, mesmo considerando o *overhead* gerado pela compressão e descompressão. No Capítulo 5 são apresentados os testes realizados para a medição do consumo de energia para a compressão e descompressão de alertas do IDS.

O algoritmo *gzip* (GNU zip) [Jean-loup e Mark 2003] foi criado para ser uma alternativa ao *compress*. Como já citado, o *gzip* utiliza o algoritmo de compressão DEFLATE e o algoritmo de descompressão INFLATE, ambos descritos nesta seção. São abordados todos os processos de forma geral, tanto de compressão quanto descompressão utilizados pelo *gzip*.

4.2.1 Algoritmo de compressão

O algoritmo de compressão utilizado pelo *gzip* é uma variação do LZ77. Seu principal objetivo é encontrar *strings* duplicadas nos dados de entrada a serem manipulados. Quando esta *string* é encontrada, sua segunda ocorrência é substituída por um ponteiro que indica onde se encontra a *string* anterior, que é formado por pares compostos por distância e comprimento. A distância é dada pelo número de bits necessários a serem percorridos até que se encontre a seqüência inicial e o comprimento representa o número de caracteres onde a seqüência é idêntica. Distâncias são limitadas por 32Kb e comprimentos a 258 *bytes*. Quando uma *string* não ocorre em nenhum lugar nos 32Kb anteriores, ela é emitida como uma seqüência literal de *bytes* [Jean-loup e Mark 2003].

Duas árvores baseadas no algoritmo de Huffman são geradas, onde uma é reservada a armazenar os literais e os comprimentos das cadeias encontradas por LZ77 e a outra armazena a distância entre as cadeias e a posição atual. As árvores são armazenadas de forma compacta no início de cada bloco. Os blocos podem ser de qualquer tamanho, desde que haja memória disponível para tal armazenamento. O processo é totalmente armazenado e executado em memória principal. Um bloco é terminado quando o algoritmo DEFLATE determina que deve ser conveniente começar outro bloco com árvores novas.

Para a localização de *strings* duplicadas, são utilizadas tabelas *hash*, onde *strings* de entrada que tenham tamanho 3 são inseridas. O índice *hash* é utilizado para o cálculo dos próximos 3 bytes. Caso o encadeamento de *hash* para esse índice não seja vazio, todas as *strings* nesse encadeamento são comparadas à *string* de entrada atual, e a maior combinação é selecionada. Na busca do encadeamento de *hash*, utiliza-se as *strings* mais recentes, a fim de favorecer as menores distâncias e assim tirar o maior proveito da codificação de Huffman. Encadeamentos de *hash* são unidos um a um, além de não haver remoções nos encadeamentos de *hash*, uma vez que o algoritmo simplesmente descarta combinações consideradas ultrapassadas.

Para evitar uma situação de pior caso, encadeamentos de *hash* são truncadas arbitrariamente quando um certo comprimento é atingido. Este parâmetro limite de comprimento é encontrado levando-se em consideração o tempo de execução. Assim, o algoritmo DEFLATE nem sempre acha a maior combinação possível, mas geralmente encontra aquela que é extensa o bastante.

4.2.1.1 Algoritmo de Compressão LZW

O algoritmo de compressão LZ77 encontra *strings* repetidas. Define-se o termo "janela deslizante" (*sliding window*) como aquele que, para qualquer ponto dos dados de entrada, armazena um histórico dos caracteres que ocorreram antes. Por exemplo, uma janela deslizante de 32K armazena o equivalente ao registro dos últimos 32768 (32 * 1024) caracteres.

Quando a próxima seqüência de caracteres a ser comprimida é idêntica a uma seqüência qualquer encontrada nos registros contidos na janela deslizante, a seqüência de caracteres é então substituída por dois números de referência, onde o primeiro representa a distância que deve ser percorrida para trás na janela deslizante, a fim de se encontrar o início da primeira seqüência. O outro valor é o comprimento, que representa o número de caracteres onde esta seqüência é idêntica à encontrada.

Um exemplo muito conhecido na apresentação do algoritmo de compressão LZ77 é o da seqüência de caracteres "Blah blah blah blah blah!", que pode ser muito comprimida ao se utilizar os conceitos deste algoritmo.

Passo 1 - Avalia-se a seqüência de caracteres "B", "l", "a", "h", " " e "b", onde o espaço entre as letras é considerado um caracter;

Passo 2 - Nota-se claramente que o os próximos 5 caracteres: "lah b" são idênticos aos 5 caracteres já "processados". Então, substitui-se essa seqüência pelo par (distância, comprimento);

Passo 3 - Até o momento pode-se comprimir a relação da seguinte forma "Blah b[D=5,L=5]". Porém, se continuar a ler os próximos caracteres, descobre-se que os próximos 18 caracteres que iniciam no segundo caractere são idênticos aos 18 caracteres que iniciam no sétimo caractere. Desse modo, os dados de entrada podem ser ainda mais comprimidos, resumindo-se em "Blah b[D=5,L=18]!".

4.2.2 Algoritmo de Descompressão

O algoritmo de descompressão INFLATE busca decodificar o mais rapidamente possível a árvore de Huffman gerada na etapa de compressão. Para tal, é importante priorizar os menores códigos, que além de mais rápidos apresentam maior número de ocorrências que os códigos longos.

Este algoritmo se organiza montando uma tabela de primeiro nível que compreende alguns números de bits de entrada menores que o tamanho do maior código. Esta etapa é realizada utilizando-se de uma quantidade de bits existentes no fluxo (*stream*), e procurando-os na tabela. A tabela vai dizer se o próximo código tem o mesmo número de bits ou menos. Se o número é menor então ele indica quanto, e se igual, ele diz o valor. Caso nenhuma das duas possibilidades ocorra, aponta para o próximo nível da tabela, e o algoritmo INFLATE seleciona mais bits e tenta decodificar um código maior.

A quantidade de bits selecionada para a realização da primeira busca na tabela é uma comparação entre o tempo que o algoritmo leva para decodificar e o tempo necessário para a construção da tabela, que é equivalente a quantidade de memória disponível. Caso esta seja "infinita", pode-se concluir até o código mais longo em uma tabela de primeiro nível. Um dos pontos negativos desta consideração é o fato de que códigos curtos são replicados por muitas vezes em uma tabela como esta. Em resumo, o funcionamento do algoritmo DEFLATE é transformar o número de bits da primeira tabela em uma variável, e setar a

velocidade máxima para esta.

O INFLATE gera novas árvores freqüentemente, o que é conveniente para uma tabela com número de níveis menor do que uma aplicação que tem somente uma árvore para todos os dados. Para este algoritmo, que tem 286 códigos possíveis para a árvore de literais ou comprimentos, o tamanho da primeira tabela é de 9 bits. Já a árvore de distância tem 30 valores possíveis, e o tamanho da primeira tabela é de 6 bits.

4.2.3 A Tabela de Busca

A tabela de busca não é uma árvore de Huffman, pois ela busca apenas os primeiros 9 bits de um símbolo de Huffman. O símbolo pode variar desde o menor tamanho, que é de 1 bit, até o maior valor, de 15 bits. Se um símbolo em particular é menor que 9 bits, então a tradução desse símbolo é duplicada em todas aquelas entradas que começarem com os bits desse símbolo. Por exemplo, se o símbolo é de 4 bits, então ele é duplicado 32 vezes em uma tabela de 9 bits, porém, se o símbolo tem 9 bits, ele aparece na tabela uma única vez. Caso o símbolo seja maior que 9 bits, então este tem uma entrada para uma segunda tabela similar, que armazena os bits restantes. Desta forma, existem entradas duplicadas, como necessário. A idéia principal da compressão de dados de primeiro plano, que é empregada neste modelo, é que na maior parte do tempo, os símbolos sejam encurtados de forma que haja somente uma busca na tabela. Para símbolos compridos que sejam menos freqüentes são feitas duas buscas. No algoritmo INFLATE, dois níveis de busca são o bastante para todas as possibilidades, já que a entrada da tabela ou aponta para outra tabela ou contém uma tradução para o símbolo e o número de bits utilizados.

O método utilizado na tabela de busca visa a diminuição do tempo gasto com o preenchimento de entradas de símbolos duplicados ao invés de realmente estar decodificando, pelo menos para a saída do algoritmo DEFLATE que gera novas árvores a cada kbytes. Por exemplo, se estiver decodificando vários milhões de símbolos, demoraria muito até ter tabelas de tamanho 2^{15} preenchidas para códigos de 15 bits. No outro extremo, poderia criar uma nova tabela para cada novo bit no código. Na verdade, isso seria uma árvore de Huffman, mas então gastaria-se tempo demais percorrendo a árvore, mesmo para símbolos curtos. Então, o número de bits para a tabela de busca é um balanceamento entre o tempo de preenchimento da tabela pelo tempo gasto durante as buscas nos níveis dois e superiores da tabela.

Exemplificando o processo comentado anteriormente, onde o código a seguir está sendo decodificado, com 10 símbolos, de 1 até 6 bits de comprimento, tem-se:

A: 0, B: 10, C: 1100, D: 11010, E: 11011, F: 11100, G: 11101, H: 11110, I: 111110, J: 111111.

A geração da primeira tabela com 3 bits de comprimento (totalizando 8 entradas) é estabelecida como segue:

000: A,1 (utilizou 1 bit);

001: A,1;

010: A,1;

011: A,1;

100: B,2 (utilizou 2 bits);

101: B,2;

110: tabela X (utilizou 3 bits);

111: tabela Y (utilizou 3 bits).

Cada entrada mostra o que é decodificado e quantos bits são utilizados ou o ponto de entrada para outra tabela, com o número de bits utilizados implícito no tamanho da tabela. A tabela X tem tamanho igual a dois desde que o maior código começando com 110 tenha 5 bits:

00: C,1; 01: C,1; 10: D,2; 11: E,2.

A tabela Y tem tamanho igual a 3 desde que o maior código começando com 111 tenha 6 bits:

000: F,2; 001: F,2; 010: G,2; 011: G,2; 100: H,2; 101: H,2; 110: I,3; 111: J,3.

Então, o que se tem aqui são três tabelas, com um total de 20 entradas a ser construída. Isso é comparado com 64 entradas para uma tabela simples, ou com 16 entradas para uma árvore de Huffman (6 tabelas de duas entradas e um tabela de 4 entradas). Assumindo que o código representa a probabilidade dos símbolos, idealmente, isso representa uma média de 1,25 buscas por símbolo, comparado com uma busca para uma tabela simples ou 1,66 buscas por símbolo na tabela de Huffman.

Assim, tem-se uma idéia de como o algoritmo trabalha. Para o algoritmo INFLATE, o significado de um símbolo em particular é geralmente mais do que apenas uma letra. Isso pode ser um byte (um "literal"), ou um comprimento ou ainda a distância que indica um valor base e um número de bits para trazer após o código que é adicionado ao valor

base. Ou simplesmente pode ser um código especial de fim de bloco. A estrutura de dados criada no algoritmo tenta codificar toda a informação de modo mais compacto possível nas tabelas.

Até aqui foram comentados os processos de compressão e descompressão utilizados pela ferramenta *gzip*, assim como os algoritmos e técnicas envolvidas. A intenção deste trabalho não é dar ênfase aos algoritmos de compressão, mas sim a utilização da compressão das alertas do modelo de IDS, além da medição da energia despendida neste processo. Com a compressão dos alertas antes mesmo de serem esteganografados, tem-se uma diminuição do consumo de energia para envio e recebimento dos mesmos. No Capítulo 5 são apresentados os testes realizados para a medição do consumo de energia para a compressão e descompressão dos alertas afim de calcular o *overhead* inserido por este processo.

4.3 Mecanismos de Incentivo à Colaboração e de Reputação

Quando todos os membros de uma rede *ad hoc* cooperativa agem de forma justa e de acordo com os protocolos predefinidos, os nós podem confiar completamente uns nos outros, já que, em tais ambientes, uma requisição somente não é atendida devido a eventos inesperados, tais como falhas, ataques ou congestionamentos. Em um ambiente *ad hoc* sem fio real, a situação se mostra bem diferente, pois é possível que existam nós oportunistas e mal intencionados, que não aderem às políticas da rede, colocando em risco a confiabilidade dos serviços oferecidos pelo sistema, além de falhas freqüentes de comunicação.

Esse problema pode ser visto pela perspectiva da confiança, e no caso de IDS para redes *ad hoc* os alertas podem ser forjados ou mascarados, por exemplo. Nesse sentido, esses nós são entendidos como membros da rede em quem não se pode confiar, já que não retribuem de forma correta às requisições de seus vizinhos, não visando o bem global. Ao se aplicar o conceito de reputação no cenário estudado tem-se uma medida de quão justo e confiável um nó é. Um nó racional rapidamente percebe que não há benefícios em acreditar nos nós não-confiáveis e o mecanismo de reputação é usado justamente para determinar quando um nó deve ou não confiar em um determinado par. A reputação de um nó determina a probabilidade que este possui de ter suas requisições atendidas, e quanto maior sua reputação, maiores são as chances de sucesso. Dessa maneira, os nós

possuem um incentivo claro em se mostrarem confiáveis e manterem a melhor reputação possível.

A reputação de cada nó é construída com base em suas experiências individuais ou nos depoimentos dos pares sobre ele. A experiência individual é o valor que um nó assume para outro. Normalmente esta experiência é adquirida por meio de uma situação de interação entre os nós, ou quando é solicitado a pontuação do nó em questão a outros nós. O valor encontrado é provido a partir das experiências individuais de cada vizinho, que após ponderadamente calculada, é considerada como sua experiência individual relativo aquele nó distante. Essa ponderação pode ser obtida por meio de uma definição formal que leva em consideração o valor dos nós delatores (vizinhos questionados sobre um determinado nó), sendo que esses, por sua vez, foram avaliados anteriormente e remetem a um peso em suas declarações.

Neste trabalho são propostas formas para que a pontuação encontrada para um nó (sua reputação) seja um mecanismo que defina a real intenção deste junto à rede, além de informar se esta é uma fonte de delação confiável ou não.

4.3.1 Experiência Individual

A experiência individual é a visão pessoal que um nó i possui a respeito da confiabilidade de um outro nó j, baseada apenas nas interações passadas entre eles. Conceitualmente, sempre que i requisita um serviço qualquer a outro nó j, por exemplo, o encaminhamento de pacotes, o recebimento de pacotes e o teste de conectividade entre os IDS, conhecido como *heartbeat*, o nó i consegue calcular qual o grau de confiança atualizando sua experiência pessoal sobre j, aumentando-a se j age justamente e diminuindo-a caso contrário.

Desta forma, para se definir formalmente a experiência individual, é necessário antes delimitar o conceito de justiça nesse contexto. Como justiça pode ser um conceito muito subjetivo, [Rocha et al. 2006] propôs uma política para ser usada em redes de roteamento sobrepostas (*overlay*), onde o principal objetivo é eliminar nós oportunistas, usando experiência individual e reputação. No presente trabalho, os conceitos de [Rocha et al. 2006] foram simplificados de forma a se adaptar ao contexto de um mecanismo de reputação para IDS em redes *ad hoc*.

Definição 1: para $S^t_{i(j)}$ representando a quantidade de serviços que o nó i forneceu ao nó j até o tempo t. $i \Longleftrightarrow j$ define a relação de requisição (r) e provimentos (p) de

serviços entre i e j. Desta forma, se $r > p$ conclui que j não é totalmente confiável, uma vez que não atendeu a todas as requisições enviadas por i.

Conforme a definição anterior sabe-se quando um nó é mais ou menos confiável, sendo que posteriormente é definida a pontuação máxima e mínima a ser alcançada pelos nós. A intenção deste trabalho é visar a qualidade da rede por meio da reputação como base para o envio de alertas e confiança entre os nós, não abordando como o protocolo de roteamento repassa qual nó atendeu às requisições anteriormente comentadas.

Um nó é considerado injusto se compreender a definição 1, desta forma não são levadas em conta as falhas ocasionalmente ocorridas, como falhas de *hardware*, congestionamentos, falta de recursos, entre outros, uma vez que todos esses fatores, mesmo que ocorridos não intencionalmente, tendem a comprometer o grau de confiabilidade da rede.

Sobre os valores mínimos e máximos de uma experiência individual de um nó i em relação a um nó j, tem-se:

Definição 2: Considerando $I_{i(j)}^{t_0}$ a experiência individual do nó i em relação ao nó j no momento atual t_0 e estando a mesma compreendida como $0 \leq I_{i(j)}^{t_0} \leq 1$. Sendo r a quantidade de serviços requisitados por i, e p a quantidade de serviços provida por j, além de n o número de vezes em que j falhou em processar completamente as requisições de i, compreendendo $n \geq 1$. Então o máximo e mínimo da pontuação para o momento futuro requisitante, $I_{i(j)}^{t_0+t}$ é definido como:

$$I_{i(j)}^{t_0+t} = \begin{cases} \max(I_{i(j)}^{t_0} + \alpha \cdot p, 1) & \text{, se} \quad p = r, \\ \\ \min\left(I_{i(j)}^{t_0} - (1 - \frac{p}{r}) \cdot \alpha \cdot n^2, 0\right) & \text{, se} \quad p < r, \quad \text{sendo} \quad n = r - p. \end{cases}$$

Se $1 - \frac{p}{r} = \frac{r-p}{r} = \frac{n}{r}$, tem-se:

$$I_{i(j)}^{t_0+t} = \begin{cases} \max(I_{i(j)}^{t_0} + \alpha \cdot p, 1) & \text{, se} \quad p = r, \\ \\ \min\left(I_{i(j)}^{t_0} - \frac{n^3}{r} \cdot \alpha, 0\right) & \text{, se} \quad p < r, \quad \text{sendo} \quad n = r - p. \end{cases}$$

A variável t_0 corresponde ao instante em que foi realizada a última avaliação, sendo que $t_0 + t$ é o instante da avaliação imediatamente posterior. A nova avaliação é realizada a cada dez requisições, não sendo possível e viável portanto, definir o tempo t, uma vez que este varia de acordo com a freqüência f, medida em requisições por unidade de tempo.

Portanto encontra-se o tempo t a partir da função $t = \frac{r}{f}$.

Por exemplo:

Com $f = 100req/m$ e $r = 10req$, tem-se: $t = \frac{r}{f} = 0,1min = 6seg.$

Com $f = 20req/s$ e $r = 10req$, tem-se: $t = \frac{r}{f} = 0,5seg.$

Ou seja, fixando o tempo a partir do número de requisições igual a dez, é possível definir $\alpha = 0,01$ de tal modo que quando todas as requisições forem providas, a experiência individual de um nó em relação a um outro é acrescida de 0,01, enquanto se $p < r$, a experiência individual decresce $0,001 \cdot n^3$ (Onde $\frac{n^3}{r} \cdot r = \frac{n^3}{10} \cdot 0,01 = 0,001 \cdot n^3$) podendo, portanto, perder até o valor de um, ou seja, toda a sua credibilidade em apenas um intervalo de tempo, caso as negue totalmente.

4.3.2 Experiência Individual Relativa a Nós Distantes

Para que o nó i tenha sua experiência individual calculada de um nó k qualquer, distante do sinal de seu rádio, deve ser feita a definição por meio de uma média da ponderação entre os nós informantes, uma vez que o nó i tem que perguntar a seus vizinhos quais são suas experiências individuais em relação ao nó k. Portanto, conforme a experiência individual de cada nó delator para com i é encontrada a experiência individual de i em relação a k. O nó i confia nos depoimentos dos pares sobre o nó k e para isto pondera sua experiência individual com cada nó.

Considera-se que cada nó deve confiar primeiramente em si para definir qual a reputação de um nó. Portanto, pode-se dizer que $I^{t_0}_{i(j)} = R^t_{i(j)}$, onde R é a reputação que aquele nó tem no momento. Para descobrir qual experiência individual assumir quando o nó se encontra distante do raio do nó requisitante, tem-se:

Definição 3: $T^t_{i(k)}$ como o depoimento dos pares sobre k, e N como o número de nós a serem questionados, definida a partir de:

$$T^t_{i(k)} = \frac{\Sigma_{k \in N} I^{t_0}_{j(k)} R^t_{i(j)}}{N}$$

Os depoimentos dos pares são informações dos outros nós da rede sobre a confiabilidade de um nó em particular, ou em outras palavras, a opinião da comunidade sobre um nó específico. O depoimento de cada nó é ponderado pela sua própria reputação. Assim, a opinião de nós com altas reputações possui maior impacto que aquela de nós

com baixas reputações, o que é importante, pois nós com baixas reputações podem ser nós oportunistas que estão tentando maliciosamente difamar nós justos (por não aceitar suas requisições, por exemplo). Como a contribuição de cada nó para a reputação é proporcional à sua própria reputação, a opinião dos nós oportunistas é diluída e a difamação é evitada. Assim, um nó pode contar tanto com sua experiência individual na interação com outro par quanto na opinião dos demais membros da rede para inferir a presente confiabilidade daquele par.

Levando em consideração que o valor da reputação de um nó está compreendido entre $0 \leq R^t_{i(k)} \leq 1$, sendo $R^t_{i(k)}$ a reputação de um nó i em relação a um nó k. Para tal, um nó deve ter pontuação conforme a equação anterior, já que o nó k é inalcançável a i. Os nós com valores de reputação $0 \leq R^t_{i(j)} < 0,5$ são considerados menos confiáveis, porém seus alertas enviados são considerados e devidamente ponderados conforme a definição 3.

4.3.3 Teoria dos Jogos como Auxílio à Reputação

A teoria dos jogos é um ramo da matemática aplicada que estuda situações estratégicas onde jogadores escolhem diferentes ações na tentativa de melhorar seu retorno. Outros exemplos de matemática aplicada são: probabilidade e estatística, cálculo numérico, matemática financeira, criptografia entre outros. A teoria dos jogos estuda as escolhas de comportamentos ótimos quando o custo e o benefício de cada opção não é fixo, mas depende, sobretudo, da escolha dos outros indivíduos [Rocha 2005].

A partir da teoria de Von Neumman que apresenta o chamado "Jogo de soma zero", cuja premissa é: ao vencedor tudo, ao perdedor nada, o trabalho de John Nash, seu discípulo, vem como um complemento com a criação do teorema muito conhecido e que leva seu nome: o "Equilíbrio de Nash". A motivação principal desta linha de pesquisa sempre foi tentar encontrar os princípios do comportamento racional. As aplicações iniciais foram em ciências sociais, economia, ética, entre outras. Seu principal objetivo é determinar, dentre um conjunto de possíveis atitudes ou estratégias, qual a melhor possível para uma determinada situação.

No trabalho em questão, a teoria dos jogos é utilizada baseando-se em seus princípios mais básicos, onde aquele que visa somente o bem próprio é excluído ou severamente punido. O dilema do prisioneiro pode ser apresentado como um exemplo para a utilização a que se propõe o emprego de reputação neste trabalho. Este problema foi originalmente formulado por Merrill Flood e Melvin Dresher em 1950. Mais tarde, Albert W. Tucker o formalizou com o tema da sentença de tempo de prisão e deu ao problema geral esse

nome específico [Rocha 2005].

O Dilema do Prisioneiro clássico funciona como segue:

"Dois suspeitos, A e B, são presos pela polícia. A polícia tem provas insuficientes para condená-los, mas, separando os prisioneiros, oferece a ambos o mesmo acordo: se um dos prisioneiros testemunhar para a procuradoria contra o outro e o outro permanecer em silêncio, o dedo-duro sai livre enquanto o cúmplice silencioso cumpre 10 anos de sentença. Se ambos ficarem em silêncio, a polícia somente pode condená-los a 6 meses de cadeia cada um. Se ambos traírem o comparsa, cada um leva 2 anos de cadeia. Cada prisioneiro faz sua decisão sem saber que decisão o outro vai tomar e nenhum tem certeza da decisão do outro. A questão que o dilema propõe é: Como o prisioneiro vai reagir?

O fato é que pode haver dois vencedores no jogo, sendo esta última solução a melhor para ambos, quando analisada em conjunto. Entretanto, os jogadores confrontam-se com alguns problemas: confiam no cúmplice e permanecem negando o crime, mesmo correndo o risco de serem colocados numa situação ainda pior, ou confessam e esperam ser libertados, apesar de que, se ele fizer o mesmo, ambos ficam numa situação pior do que se permanecessem calados?"

Na realidade, não importa os valores das penas, mas o cálculo das vantagens de uma decisão cujas conseqüências estão atreladas às decisões de outros agentes, onde a confiança e traição fazem parte da estratégia em jogo.

Neste trabalho, como mencionado anteriormente, o cálculo da reputação é apresentado como meio de julgamento que cada nó deve ter de seu vizinho. Esta reputação varia conforme o depoimento dos pares e do seu próprio conhecimento, levando em consideração o valor da reputação local em relação ao relato de vizinhos. Esse cálculo foi simplificado de [Rocha et al. 2006], onde é usada uma função de utilidade que mede a latência relativa da conexão, isto é, a latência da conexão com um nó j comparada com a latência da melhor conexão possível que o jogador i poderia estabelecer na rede sobreposta. Essa função não foi usada nesse trabalho justamente por não representar uma métrica adequada para uma rede *ad hoc*.

Se este cenário for visualizado pela perspectiva da teoria dos jogos, uma rede *ad hoc* sem fio se comporta como um jogo não-cooperativo no que diz respeito a utilização dos recursos da rede, porém é necessário saber em quais alertas, de quais nós, pode-se confiar e para tal aplica-se a este modelo de IDS a estratégia de incentivo à colaboração e reputação. Ao iniciar o jogo, é delegado o valor de 0,5 a todos os nós que estão na

rede, e posteriormente aos que ingressarem. O valor de reputação 0,5 representa um valor intermediário, que expressa uma confiança inicial mas que pode cair rapidamente e ser compreendida pelo nós como menos confiáveis, aqueles com $R_{\min(i)}$ contidos em $0 \leq R_{\min(i)} < 0,5$. Se $R^t_{i(j)} < R_{\min(i)}$ então i não considera as alertas de j.

Nós com o $R^t_{i(j)} < R_{\min(i)}$ tendem a ser excluídos da rede. Técnicas devem avaliar o quanto isso influencia a rede, podendo ou não delegar maneiras que controlem a volta de um nó excluído após certo período de tempo.

4.4 Conclusão

Nesse capítulo foi apresentado um modelo de arquitetura para sistemas de detecção de intrusão para redes *ad hoc*, usando troca de mensagens esteganografadas e os mecanismos de incentivo a colaboração e reputação. O modelo descrito trabalha em um sistema colaborativo onde cada nó recebe alertas dos demais sistemas de IDS, ponderando esses alertas em função da reputação que este outro nó tem consigo. Cada alerta é enviado a seus vizinhos em mensagens esteganografadas em imagens.

A idéia central da arquitetura é que em cada nó, o sistema de detecção de intrusão funciona independentemente dos demais nós, trocando alertas para formar uma opinião global sobre uma ou mais detecções locais. Como os alertas são enviados esteganografados, utiliza-se um mecanismo de reputação para ponderar os alertas recebidos. Caso um nó não tenha boa reputação, seus alertas são desprezados. A reputação é calculada em função das interações entre os nós, principalmente repasse de pacotes (roteamento), onde um mecanismo monitora os repasses dos pacotes de um vizinho, determinando assim se ele está colaborando com a rede ou não.

Os alertas são comprimidos antes de serem enviados com o objetivo de diminuir o tamanho da imagem necessária para transportar esse alerta esteganografado, além de economizar energia na transferência.

O Capítulo 5 apresenta os testes efetuados e os resultados obtidos para a validação dessa proposta.

Capítulo 5

Resultados Obtidos

Este capítulo apresenta os resultados obtidos por essa proposta. Foram efetuados testes iniciais com o objetivo de: identificar características principais dos alertas do IDS; encontrar o tamanho de imagem ideal para transportar os alertas usando esteganografia; verificar a viabilidade de uso do tráfego real de imagens para transportar os alertas.

A partir dos testes iniciais, também foram efetuados outros testes para validação da proposta, como: simulações de tráfego do IDS em redes *ad hoc* no simulador de rede NS-2 para avaliação da probabilidade de entrega dos pacotes; avaliação do consumo de energia da proposta, identificando o *overhead* inserido; comparação com a arquitetura ICPAH, apresentando cenários onde a presente proposta consome menos energia que uma arquitetura usando criptografia.

5.1 Testes com IDS

Nesta seção são apresentados os testes com um IDS para identificar as características de alertas tais como tamanho, formato e freqüência desses em um ambiente real, realizados em um servidor em produção de um Provedor de Serviços Internet na cidade de Juiz de Fora (MG).

O IDS usado para estes testes é o Prelude-IDS [Vandoorselaere 1998], um IDS híbrido, isto é, um sistema que permite a unificação em um sistema centralizado a vários tipos de aplicações (chamados de sensores), sejam de código-fonte aberto ou proprietário. A fim de implementar tal tarefa, o Prelude-IDS utiliza o padrão IDMEF [Curry e Debar 2002] que permite que diferentes tipos de sensores gerem eventos utilizando uma mesma linguagem.

O Prelude-IDS não considera confiável acreditar em uma única fonte de informação a fim de fazer a análise de segurança, já que métodos diferentes de análise têm vantagens

diferentes, então o Prelude-IDS une estes métodos em um único sistema, produzindo uma ferramenta detalhada de análise de segurança [Vandoorselaere 1998]. O Prelude-IDS tem a habilidade de encontrar atividades maliciosas em conjunto com outros tipos de sensores, como o *Snort* [Westphal 2000], *honeyd* [Provos 2003], *nessus* [Deraison 1999], *samhain* [Wichmann 2006], diversos tipos de registros de sistema (*syslog*), entre outros, a fim de melhorar sua evidência ao apontar um ataque [Vandoorselaere 1998].

Foram coletados dados durante 4 dias consecutivos, gerando um número médio de 8090 alertas por dia (mínimo de 6960 e máximo 9100 alertas por dia). Dos alertas coletados, de acordo com o padrão de alertas do IDMEF [Curry e Debar 2002], 77,3% são de nível baixo, 17,8% médio, 2,4% alto e 2,5% dos alertas são do tipo *heartbeat*. Esses valores consideram atividades do servidor de emails do provedor. Os valores apresentados na Tabela 5.1 contabilizam os alertas gerados, desconsiderando os alertas do sensor de *anti-spam*, por não representar um dado relacionado à atividade de redes *ad hoc*. Um exemplo de um alerta no formato IDMEF é apresentado na Figura 5.1.

Tabela 5.1: Alertas no padrão IDMEF gerados pelo Prelude-IDS

Dia	1			2			3			4		
	Alertas	%	Freqüência por Minuto	Alertas	%	Freqüência por Minuto	Alertas	%	Freqüência por Minuto	Alertas	%	Freqüência por Minuto
Baixo	2662	73,33	1,85	2668	45,16	1,85	2666	50,30	1,85	2657	63,23	1,85
Médio	357	9,84	0,25	2926	49,53	2,03	2096	39,55	1,45	1180	28,08	0,82
Alto	419	11,54	0,29	122	2,06	0,08	346	6,53	0,24	173	4,12	0,12
Heartbeat	192	5,29	0,13	192	3,25	0,13	192	3,62	0,13	192	4,57	0,13
Total	**3630**		**2,52**	**5909**		**4,10**	**5301**		**3,68**	**4204**		**2,92**

Tabela 5.2: Relação entre tamanho e freqüência média dos alertas gerados pelo Prelude-IDS

Nível do Alerta	Tamanho Médio do Alerta (Bytes)	Freqüência Média (Alertas/minuto)	Bytes Médios por minuto
Baixo	2.055	1,85	3.800,68
Médio	2.396	1,14	2.728,36
Alto	2.233	0,18	410,93
Hearteat	866	0,13	115,47
Total	**2.270**	**3,31**	**7.055,44**

A Tabela 5.1 é usada como base para a construção da Tabela 5.2, onde se identifica que a partir dos alertas coletados, o alerta médio do sistema de IDS possui o tamanho de **2.270 bytes**. Além disso, tem-se a freqüência média de alertas em 3,31 alertas por minuto, chegando ao valor médio por minuto de **7.055,44 bytes** de tráfego na rede inserido pelos alertas, ou seja, **940,73bps**.

```
<IDMEF-Message>
  <Alert messageid="5460404080">
    <Analyzer analyzerid="2330050988037005" name="prelude-manager"
manufacturer="http://www.prelude-ids.com" model="Prelude Manager" version="0.9.0-rc8"
class="Concentrator" ostype="Linux" osversion="2.6.15-26-server">
      <Node category="unknown">
        <name>servidor2</name>
        <Address category="ipv4-net-mask">
          <address></address>
        </Address>
      </Node>
      <Process>
        <name>prelude-manager</name>
        <pid>4027</pid>
        <path>/usr/local/bin/prelude-manager</path>
      </Process>
      <Analyzer analyzerid="3706377553007457" name="prelude-lml"
manufacturer="http://www.prelude-ids.com" model="Prelude LML" version="0.9.0" class="Lc
Analyzer" ostype="Linux" osversion="2.6.15-26-server">
        <Node category="unknown">
          <name>servidor2</name>
          <Address category="ipv4-net-mask">
            <address></address>
          </Address>
        </Node>
        <Process>
          <name>prelude-lml</name>
          <pid>4039</pid>
          <path>/usr/bin/prelude-lml</path>
        </Process>
        <Analyzer name="kernel" class="Kernel">
          <Node category="unknown">
            <name>servidor2.domain.invalid</name>
            <Address category="ipv4-addr">
              <address>192.168.0.3</address>
            </Address>
          </Node>
          <Process>
            <name>kernel</name>
          </Process>
        </Analyzer>
      </Analyzer>
    </Analyzer>
    <CreateTime ntpstamp="0xc921a5f0.0x62b85000">2006-12-06T18:26:24.385625-
02:00</CreateTime>
    <DetectTime ntpstamp="0xc921a5ec.0x00000000">2006-12-06T18:26:20.00-
02:00</DetectTime>
    <AnalyzerTime ntpstamp="0xc921a5f0.0x63325000">2006-12-06T18:26:24.387486-
02:00</AnalyzerTime>
    <Target decoy="unknown" interface="lo">
      <Node category="unknown">
        <name>servidor2.domain.invalid</name>
        <Address category="ipv4-addr">
          <address>192.168.0.3</address>
        </Address>
      </Node>
      <Process>
        <name>kernel</name>
      </Process>
    </Target>
    <Classification text="Promiscuous mode detected"/>
    <Assessment>
      <Impact severity="low" completion="succeeded" type="other">A sniffer is probably
running on this machine</Impact>
    </Assessment>
    <AdditionalData type="string" meaning="Log received
from">/var/log/messages</AdditionalData>
    <AdditionalData type="string" meaning="Original Log">Dec  6 18:26:20 servidor2
kernel: [42949911.020000] device lo entered promiscuous mode</AdditionalData>
  </Alert>
</IDMEF-Message>
```

Figura 5.1: Exemplo de alerta no formato IDMEF.

5.2 Testes de esteganografia com imagens genéricas

Nesta seção são apresentados os testes de esteganografia utilizando imagens JPEG, visando encontrar o tamanho ideal de uma imagem para transportar alertas do IDS.

Os testes foram realizados em um ambiente Linux Ubuntu 6.10 utilizando a ferramenta *convert* [ImageMagick 2007] (pertencente ao pacote *ImageMagick*) para a geração de imagens de diversos tamanhos e padrões diferentes e as ferramentas *steghide* [Hetzl 2003] e *outguess* [Provos 2002] responsáveis pela inserção das mensagens de alertas nas imagens geradas, ou seja, responsáveis pela esteganografia dos alertas, usando imagens como stego-objeto.

Com o objetivo inicial de se determinar o tamanho médio de uma imagem para esteganografar diversos tipos de alertas do sistema de IDS, foram geradas imagens de quatro padrões diferentes, conforme visto na Figura 5.2. Para cada padrão, foram geradas imagens de tamanhos variando entre 50x50 *pixels* até 200x200 *pixels*.

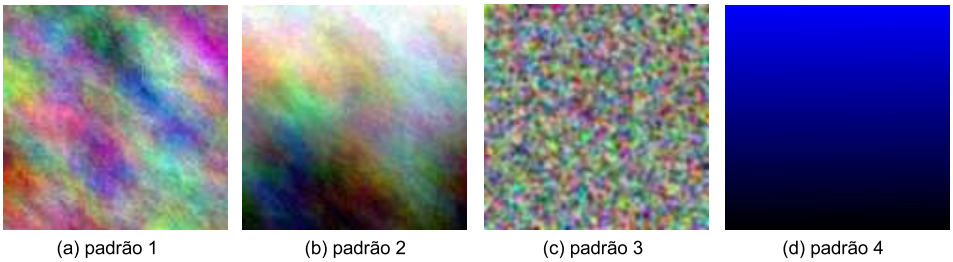


Figura 5.2: Exemplo das imagens utilizadas no processo de esteganografia e criadas com a ferramenta *convert*

A Tabela 5.3 sumariza os resultados encontrados utilizando o software *steghide*, indicando o tamanho mínimo das imagens que conseguiram esteganografar cada tipo de alerta, divididos por padrão de imagem. Pode-se observar que imagens com um total de 16200 pixels são capazes de transportar alertas de qualquer nível, inserindo um *overhead* médio de 10 vezes o tamanho do alerta, isto é, para transportar um alerta de 2409 bytes, é necessário transmitir uma imagem de 23.641 bytes. As imagens do padrão 3 necessitam de mais pixels para transportar o mesmo tamanho de alerta do que as imagens do padrão 1 e 2. As imagens do tipo 4, por não ter uma grande variação de cores, não conseguiu transportar nenhum tipo de alerta, nem mesmo os de tamanho reduzido (1079 bytes). O próprio *steghide* faz a compressão do alerta (gzip nível 6) e por isso não foram feitos testes em separado sem compressão com esse software.

A Tabela 5.4 sumariza os resultados encontrados agora com a ferramenta *outguess* e com os alertas comprimidos com gzip (nível de compressão 1 - rápida). Assim, com uma simples compressão, os alertas reduzem em média 57,4% de tamanho. Conseqüentemente, o tamanho das imagens necessárias para transportar os alertas caem praticamente pela metade, chegando a uma imagem de 16KBytes como ideal para transportar qualquer tipo de alerta. Verifica-se também que quanto menor o tamanho do alerta, maior é o *overhead*

Tabela 5.3: *Overhead* inserido na utilização da esteganografia com o *steghide* de quatro tipos de alertas em cada padrão de imagem

Nível do Alerta	Imagem	Pixels (WxH)	Tamanho do Alerta Original (*bytes*)	Tamanho do Alerta Esteganografado (*bytes*)	*Overhead* (x)
Baixo	1	14000 (200x70)	2057	19145	9,30
Médio	1	15300 (170x90)	2409	20768	8,62
Alto	1	15000 (150x100)	2257	21204	9,39
Heartbeat	1	8400 (70x120)	1079	11665	10,81
Baixo	2	14000 (200x70)	2057	16154	7,85
Médio	2	16200(180x90)	2409	18727	7,77
Alto	2	15300 (170x90)	2257	17760	7,86
Heartbeat	2	8400 (70x120)	1079	10174	9,42
Baixo	3	15000(150x100)	2057	22160	10,77
Médio	3	16200 (180x90)	2409	23641	9,81
Alto	3	16200(180x90)	2257	23637	10,47
Heartbeat	3	9000 (150x60)	1079	13604	12,60
Baixo	4	-	2057	-	-
Médio	4	-	2409	-	-
Alto	4	-	2257	-	-
Heartbeat	4	-	1079	-	-

inserido (8,67 vezes o tamanho do alerta no pior caso apresentado).

Assim, são levados em consideração os dois tamanhos de mensagens esteganografadas (**16KBytes e 23KBytes**) como o tamanho de cada mensagem para os demais cálculos e simulações. O objetivo de considerar os dois tamanhos de alerta é o de trabalhar com mais de um sistema esteganográfico e não deturpar os resultados utilizando somente o menor valor encontrado. O *outguess*, confome apresentado pelo Capítulo 3, é uma ferramenta bastante consolidada, porém já existem ferramentas de esteganálise que detectam a esteganografia executada por ela. Já o *steghide*, de acordo com os testes executados, não foi detectado por nenhuma ferramenta de esteganálise.

Com base nesses dados, observa-se que com a freqüência média em 7.055,44 bytes por minuto (940,73bps) e o *overhead* de esteganografia do alerta em 10 vezes o seu próprio tamanho, chega-se ao número de 70.554,40 bytes por minuto, ou seja, uma média de 1.175,91 bytes por segundo (9.407,25bps) de *overhead* inserido pela proposta, utilizando o *steghide* para a esteganografia. Com o *outguess*, consegue-se uma taxa de 6,5 vezes o tamanho do alerta original (não compactado), chegando a 45.695,73 bytes por minuto, conseqüentemente, uma média de 761,59 bytes por segundo (6.092,76bps).

Tabela 5.4: *Overhead* inserido na utilização da esteganografia com o *outguess* de quatro tipos de alertas compactados em cada padrão de imagem

Nível do Alerta	Imagem	Pixels (WxH)	Tamanho do Alerta Compactado (*bytes*)	Tamanho do Alerta Esteganografado (*bytes*)	*Overhead* Compactado (x)	*Overhead* Original (x)
Baixo	1	10000 (200x50)	870	13427	15,43	6,53
Médio	1	11400 (190x60)	966	14643	15,15	6,08
Alto	1	10800 (120x90)	936	14201	15,17	6,29
Heartbeat	1	6000 (120x50)	502	8589	17,10	7,96
Baixo	2	11400 (190x60)	870	12472	14,33	6,06
Médio	2	13000 (130x100)	966	14567	15,08	6,05
Alto	2	12600 (180x70)	936	13930	14,88	6,17
Heartbeat	2	6500 (130x50)	502	8037	16,01	7,45
Baixo	3	10500 (150x70)	870	14830	17,04	7,21
Médio	3	12800 (160x80)	966	16285	16,85	6,76
Alto	3	12100 (110x110)	936	16124	17,22	7,14
Heartbeat	3	6500 (130x50)	502	9358	18,64	8,67
Baixo	4	-	870	-	-	
Médio	4	-	966	-	-	
Alto	4	-	936	-	-	
Heartbeat	4	-	502	-	-	

5.3 Testes de Esteganografia em Tráfego Real de Imagens

Com base também no tamanho do alerta médio e em um *log* diário de acessos de um *webcache* real, foram feitos testes de esteganografia em imagens reais acessadas pelos clientes do *webcache* real do mesmo provedor de Juiz de Fora, para verificar a possibilidade de utilização do próprio tráfego real de imagens na rede servir como meio de transporte para os alertas do IDS. Verificou-se que apenas 4,3% das imagens reais (5262 de um total de 123736) conseguiriam transportar tais alertas usando o software *steghide*. Apesar do número reduzido de imagens que tem a capacidade de transportar tais alertas, verifica-se com base na Tabela 5.1 que o número máximo de alertas transportados por dia foi de 5909, número esse muito próximo do total das imagens reais que conseguiram transportar os alertas. Isso mostra que seria necessário a criação de poucas imagens extras, diminuindo significativamente o *overhead* da arquitetura.

Novos testes foram efetuados com imagens reais, agora se baseando em um novo conjunto de *logs* do *webcache*, no tamanho dos alertas de cada tipo e utilizando esteganografia com o *outguess*. A Tabela 5.5 sumariza os resultados, onde pode-se verificar que mesmo com uma porcentagem reduzida de imagens que conseguem esteganografar os alertas (2,68% do total da imagens do dia no pior caso), ainda assim é possível aproveitar o próprio tráfego normal da rede para diminuir a necessidade de geração de imagens. Quanto maior o tráfego de imagens na rede, menor é essa necessidade.

Tabela 5.5: Esteganografia de imagens reais para transporte de alertas.

Tamanho do Alerta (**bytes**)	**Dia 1**		**Dia 2**		**Dia 3**		**Dia 4**		**Dia 5**	
	imagens	%	imagens	%	imagens	%	imagens	%	imagens	%
870	3344	3,26	2723	2,98	3282	3,85	3916	5,68	3369	4,65
966	2979	2,91	2454	2,68	2896	3,40	3493	5,07	3048	4,21
936	3086	3,01	2511	2,75	3019	3,54	3586	5,2	3141	4,34
502	5700	5,56	4533	4,96	5752	6,75	7013	10,17	6350	8,77
Total de imagens por dia	**102480**		**91424**		**85169**		**68932**		**72407**	

Para a utilização do tráfego de imagens já existentes na rede é necessário avaliar a freqüência e a latência com que essas trafegam, além da possível invasão de privacidade nessa utilização, já que o uso de esteganografia implica em modificação dessas imagens, podendo inclusive destruir uma esteganografia já existente.

5.4 Resultados das Simulações

Foram feitas cerca de 1500 simulações para calcular a taxa de entrega dos alertas do IDSs na rede, variando o número de nós, o protocolo de roteamento, a velocidade dos nós e a quantidade de tráfego (número de conexões). Estas simulações são comparadas com os valores do trabalho ICPAH descrito em [Brazil 2007]. Neste conjunto de simulações o número de nós com IDS foi mantido fixo em nove devido ao tamanho dos cenários e ao alcance dos nós, além de servir como base de comparação. A velocidade dos nós variou entre zero, 1,5m/s e 10m/s e o número de conexões variou entre 10, 15 e 20 conexões. O tempo das simulações foi de 300 segundos. O tamanho dos cenários variou de 670x670, 900x900 e 1000x1000m^2. Foram testados o envio de alertas usando como fonte e destino os nove nós fixos e uma taxa de 2, 5 e 9 IDSs enviando alertas a cada 30 segundos. O padrão de movimentação dos nós foi aleatório. Foram usadas as ferramentas "setdest" para geração dos cenários, gerando quatro cenários de movimentações diferentes para cada tamanho de cenário, e o script "cbrgen.tcl", para geração do tráfego CBR das simulações (ambos setdest e cbrgen.tcl ferramentas integrantes do NS-2). Para a geração dos alertas do IDS, o script "cbrgen.tcl" serviu como base para a criação de um novo script (idsgen.tcl) para a geração aleatória de envio dos alertas a cada 30 segundos, enviando alertas com tamanhos de 16KB e 32KB. Além disso, foram construídos scripts em linguagem "shell", "awk" e "perl" com o intuito de calcular as taxas de entrega de pacotes.

A variação das velocidades nos valores citados anteriormente tenta modelar três tipos de usuários:

- velocidade igual a zero - cenários de redes de sensores estáticos;

- velocidade igual a 1,5m/s - cenários de redes com pessoas se deslocando e levando consigo um sensor ou computador portátil;
- velocidade igual a 10m/s - cenários de redes com veículos se deslocando e levando consigo os nós da rede sem fio.

Em todas as simulações foram usados o protocolo reativo DSR (*Dynamic Source Routing*) e o protocolo pró-ativo DSDV (*Destination Sequenced Distance Vector*), protocolo MAC 802.11b, nós se comunicando numa taxa CBR (*Constant Bit Rate*) de 16Kbps com tamanho de pacote igual a 512 bytes. Esta taxa, bem como o tamanho de pacote utilizado, são bastante utilizados para modelar protocolos de rede sem fio bem como em aplicações militares [Pereira e Pedroza 2004]. O modelo dos nós utilizado foi simulando um rádio Wave Lan Lucent 914 Mhz com antenas a 1,5m de distância do solo, alcance de 250 metros e taxa máxima de 2Mbps.

O objetivo das simulações foi validar parâmetros como taxa de entrega de alertas dos IDSs na rede. Assim também é possível medir o consumo adicional de energia dos nós da rede e o *overhead* introduzido pela proposta. Estes valores foram analisados e comparados com a proposta ICPAH.

Diversos parâmetros dos cenários avaliados foram configurados de forma similar aos trabalhos ICPAH e de [Zhou e Haas 1999] para efeito de comparação.

Foram escolhidos três tipos de cenários para as simulações:

- cenário esparso - formado por 30 nós de rede, 9 IDSs com dimensão de 1000x1000m^2;
- cenário denso - formado por 50 nós de rede, 9 IDSs com dimensão de 900x900m^2;
- cenário muito denso - formado por 50 nós de rede, 9 IDSs com dimensão de 670x670m^2.

Outra informação variável nas simulações foi a quantidade de alertas enviados em cada cenário. Cada IDS enviou alertas de dois tamanhos diferentes: um alerta de 23KBytes (baseados nos resultados obtidos com o *steghide*) e um alerta de 16KBytes (baseados nos resultados obtidos com o *outguess*), em um intervalo de 30 em 30 segundos, ou seja, 10 alertas a cada simulação. Com isso, foram criadas as seguintes situações:

- cenário brando - formado por 2 nós com IDSs enviando alertas e um total de 20 alertas na simulação;

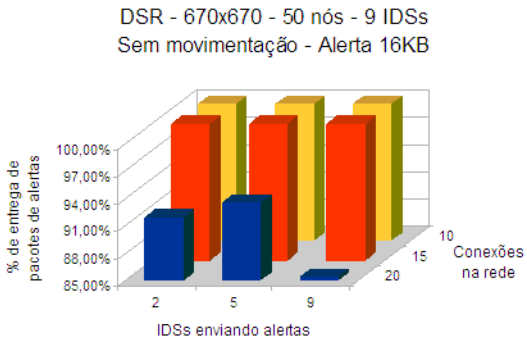
- cenário médio - formado por 5 nós com IDSs enviando alertas e um total de 50 alertas na simulação;
- cenário hostil - formado pelos 9 nós com IDSs enviando alertas e um total de 90 alertas na simulação.

Os primeiros resultados mostram que o protocolo DSR foi melhor na taxa de entrega dos alertas em 70% contra 30% do protocolo DSDV, comparando exatamente o mesmo cenário de simulação. A taxa de entrega de pacotes vai diminuindo conforme aumenta o número de conexões já existentes nas simulações, ou seja, a taxa de entrega dos alertas do IDS diminui em função da carga da rede. Os gráficos apresentados nas Figuras 5.3, 5.4 e 5.5 apresentam os resultados com as simulações com alertas de 16KB. Os gráficos apresentados nas Figuras 5.6, 5.7 e 5.8 apresentam os resultados com as simulações com alertas de 23KB. Todos os gráficos apresentam o valor médio das 4 simulações feitas com o mesmo conjunto de váriáveis, mudando somente o padrão de movimentação.

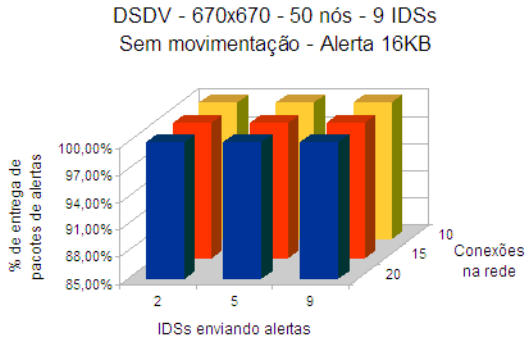
É importante salientar que não é o objetivo deste trabalho avaliar qual o melhor protocolo de roteamento *ad hoc*, e sim avaliar o desempenho da proposta quando usada com protocolos tradicionais de roteamento.

A Figura 5.3 apresenta os resultados com alertas de 16KB no cenário muito denso (670x670m^2 com 50 nós móveis). Nota-se que em cenários sem movimentação (Figuras 5.3(a) e 5.3(b)) e em cenários com mobilidade baixa (Figuras 5.3(c) e 5.3(d)) o DSDV apresenta uma entrega total dos alertas, mesmo nos casos com alto tráfego na rede (20 conexões já existentes na rede). Já nos cenários com grande movimentação (Figuras 5.3(e) e 5.3(f)), o protocolo DSR passa a ser melhor em alguns casos (envio de 2 e 9 alertas, com tráfego de 10 e 15 conexões na rede). Isso se deve ao fato do protocolo DSR ser reativo, onde a alta mobilidade implica em maiores mudanças em rotas. No DSDV, por ser pró-ativo, a alta mobilidade acaba atrapalhando o estabelecimento prévio de rotas.

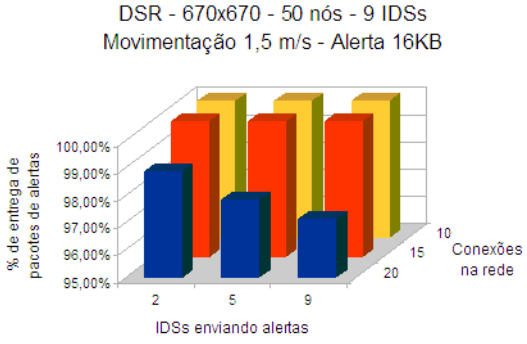
A Figura 5.4 apresenta os resultados com alertas de 16KB no cenário denso (900x900m^2 com 50 nós móveis). Nesse caso, o protocolo DSDV se apresenta com uma entrega melhor do que o DSR no cenário sem movimentação (Figuras 5.4(a) e 5.4(b)). Em alguns cenários, o DSR se torna uma opção muito ruim (Figura 5.4(a) com tráfego de 20 conexões e 9 alertas e Figura 5.4(e) com tráfego de 15 e 20 conexões, com 9 alertas). Esses casos podem ser explicados pela falta de conexão em parte do cenário em determinados momentos, comprometendo a entrega total dos alertas. Nesses casos, o DSDV se apresenta melhor, com entregas superiores a 87% dos alertas.



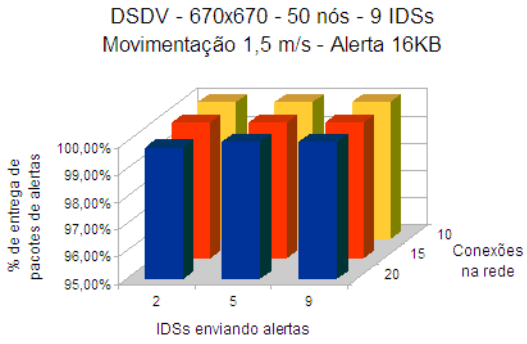
(a) DSR - sem movimentação



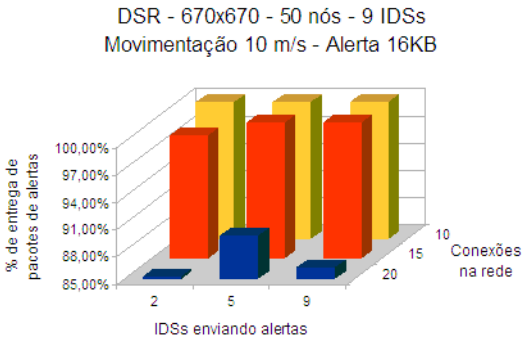
(b) DSDV - sem movimentação



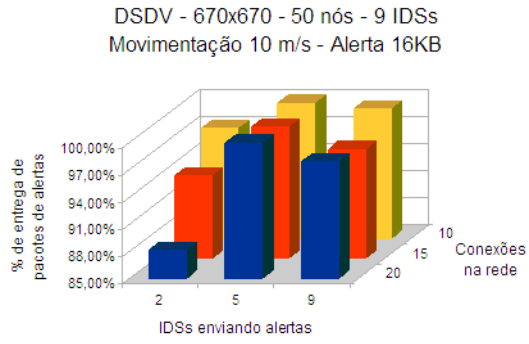
(c) DSR - movimentação 1,5 m/s



(d) DSDV - movimentação 1,5 m/s



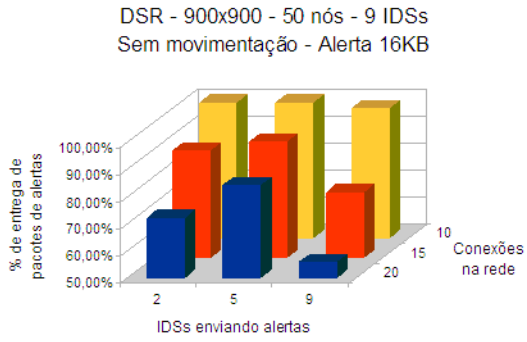
(e) DSR - movimentação 10 m/s



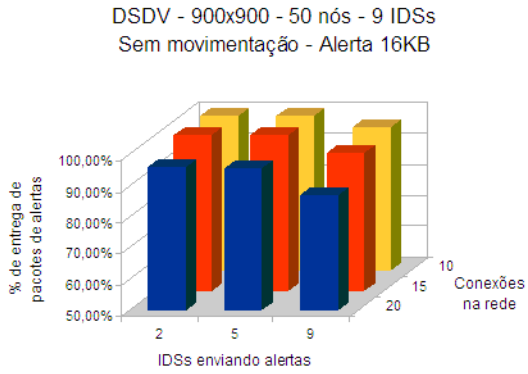
(f) DSDV - movimentação 10 m/s

Figura 5.3: Porcentagem de entrega de pacotes no cenário de 670x670, com 50 nós e 9 IDSs, nos protocolos DSR e DSDV, enviando alertas de 16KB

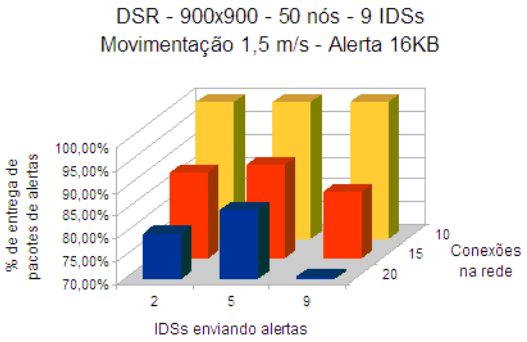
A Figura 5.5 mostra os resultados com alertas de 16KB no cenário esparso de 1000x1000m^2 com 30 nós móveis, o mais esparso. Aqui, em função da falta de conexão nos cenários, o DSR apresenta alguns cenários com entrega de pacotes inferior a 50%. Esse pode ser um fator preocupante, principalmente em cenários hostis, onde a entrega de alertas se torna fundamental. O DSDV, principalmente em cenários com alta mobi-



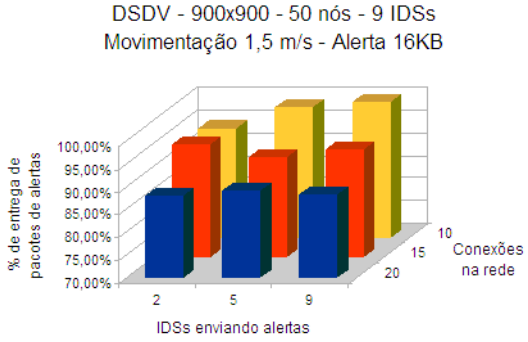
(a) DSR - sem movimentação



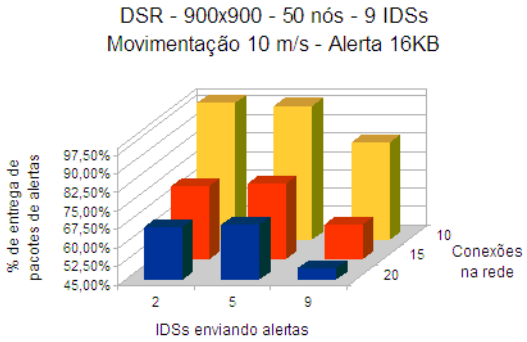
(b) DSDV - sem movimentação



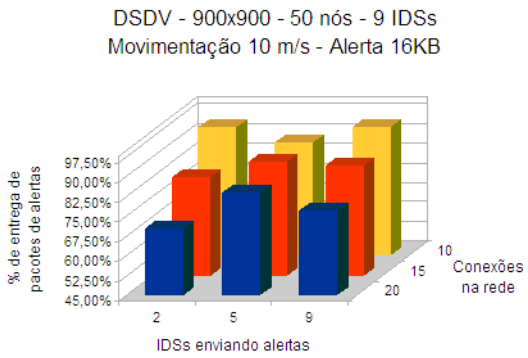
(c) DSR - movimentação 1,5 m/s



(d) DSDV - movimentação 1,5 m/s



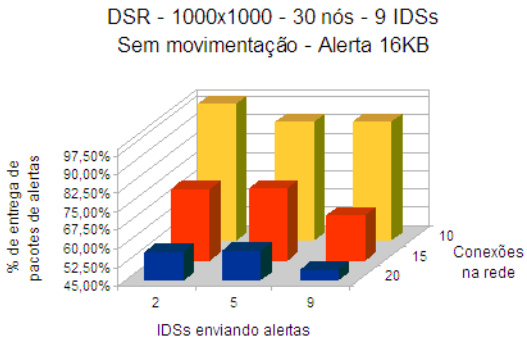
(e) DSR - movimentação 10 m/s



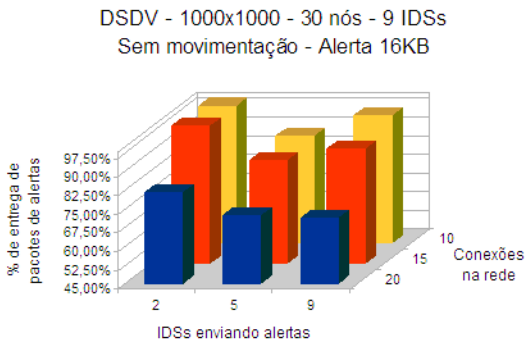
(f) DSDV - movimentação 10 m/s

Figura 5.4: Porcentagem de entrega de pacotes no cenário de 900x900, com 50 nós e 9 IDSs, nos protocolos DSR e DSDV, enviando alertas de 16KB

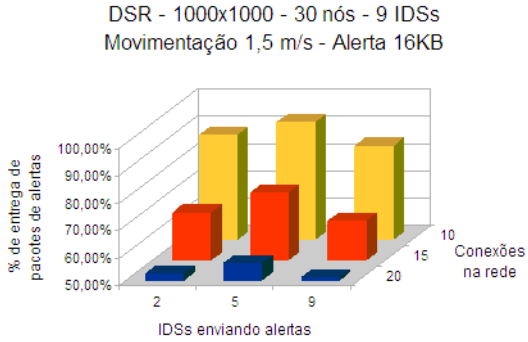
lidade (Figura 5.5(f)) apresenta as piores taxas de entrega, chegando a menos de 15% no caso de 20 conexões e 2 IDSs enviando alertas. Nesses casos, a falta de conectividade se apresenta por mais da metade do tempo de simulação, gerando esse resultado ruim. Nos demais cenários (principalmente nas Figuras 5.5(c) e 5.5(d)) o DSDV se apresenta melhor em praticamente todos os casos, inclusive nos cenários com muitas conexões já



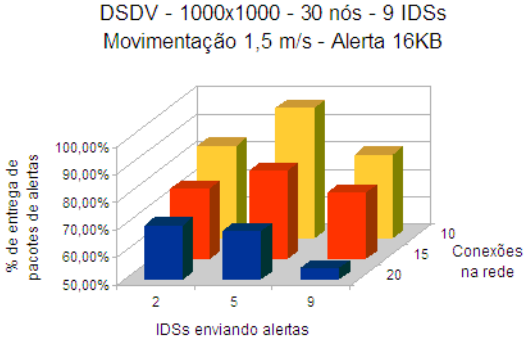
(a) DSR - sem movimentação



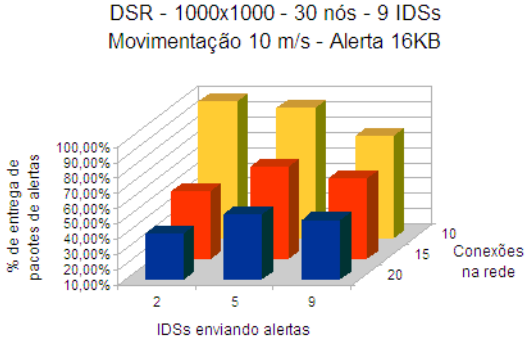
(b) DSDV - sem movimentação



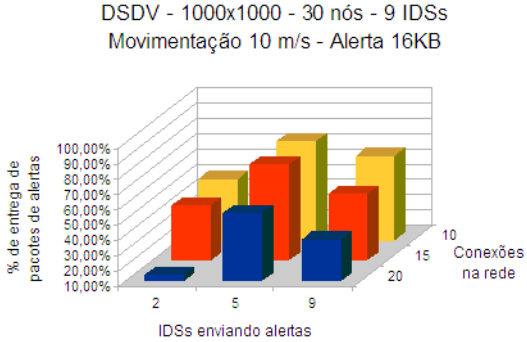
(c) DSR - movimentação 1,5 m/s



(d) DSDV - movimentação 1,5 m/s



(e) DSR - movimentação 10 m/s

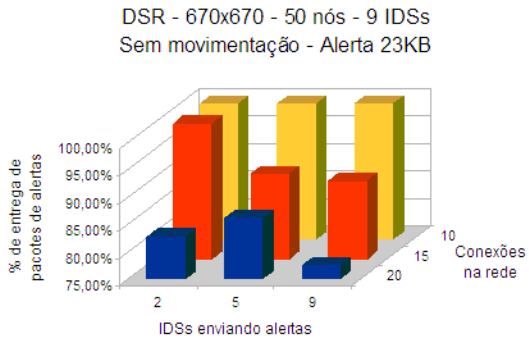


(f) DSDV - movimentação 10 m/s

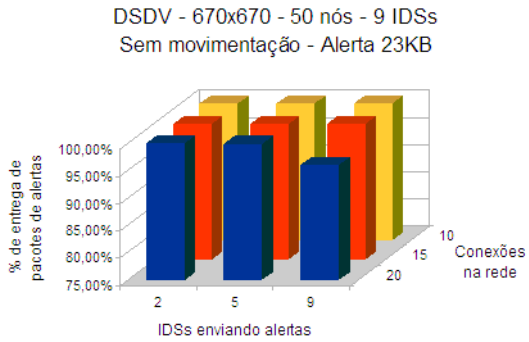
Figura 5.5: Porcentagem de entrega de pacotes no cenário de 1000x1000, com 30 nós e 9 IDSs, nos protocolos DSR e DSDV, enviando alertas de 16KB

existentes (20 conexões existentes).

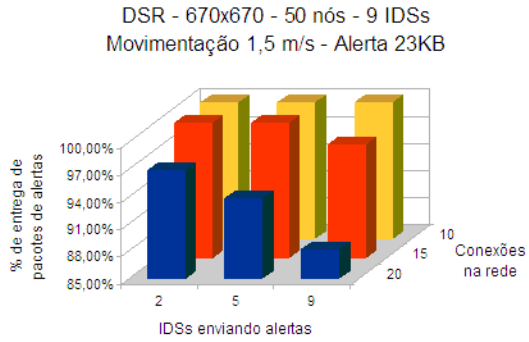
A Figura 5.6 apresenta os resultados com alertas de 23KB no cenário muito denso (670x670m^2 com 50 nós móveis). Com esse tamanho de alerta, o protocolo DSDV foi melhor em todos os casos sem movimentação (Figuras 5.6(a) e 5.6(b)). Quando é inserida a movimentação, o DSR somente é melhor nos casos com baixa carga na rede



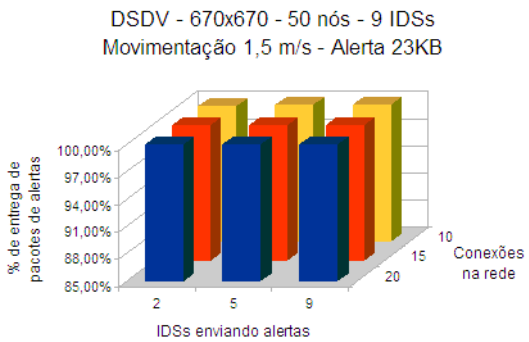
(a) DSR - sem movimentação



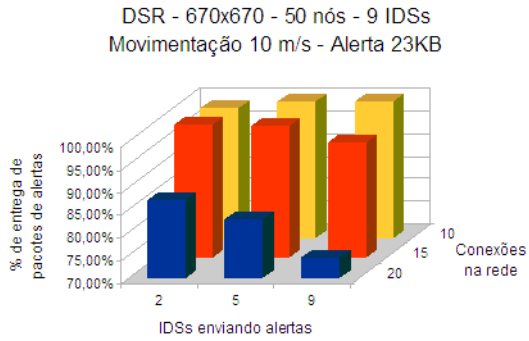
(b) DSDV - sem movimentação



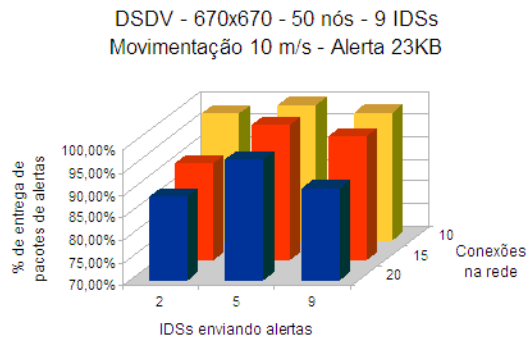
(c) DSR - movimentação 1,5 m/s



(d) DSDV - movimentação 1,5 m/s



(e) DSR - movimentação 10 m/s

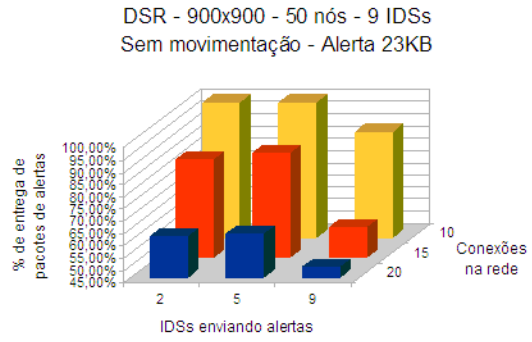


(f) DSDV - movimentação 10 m/s

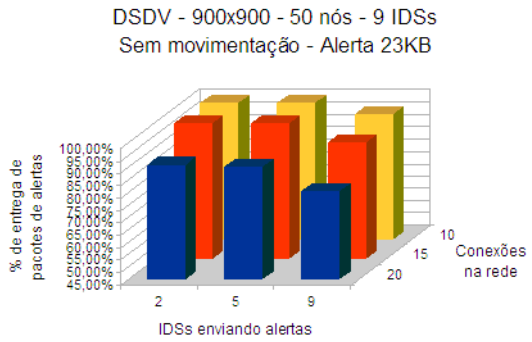
Figura 5.6: Porcentagem de entrega de pacotes no cenário de 670x670, com 50 nós e 9 IDSs, nos protocolos DSR e DSDV, enviando alertas de 23KB

(10 e 15 conexões existentes) e com baixa taxa de alertas (2 IDSs enviando alertas). Nos demais, o DSDV faz a entrega dos alertas melhor.

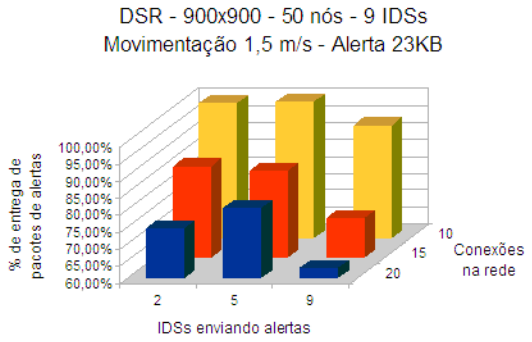
No cenário denso com alertas de 23KB (Figura 5.7) o protocolo DSDV se comporta melhor em todos os casos, independente da mobilidade, da carga existente na rede e da taxa de IDSs enviando alertas. Nesses casos, o DSR volta a ter taxas preocupantes,



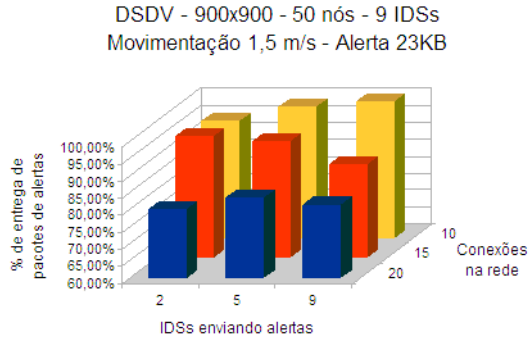
(a) DSR - sem movimentação



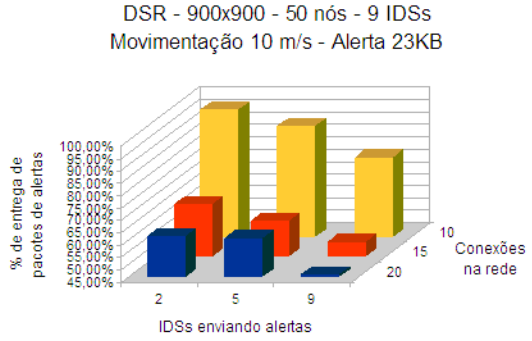
(b) DSDV - sem movimentação



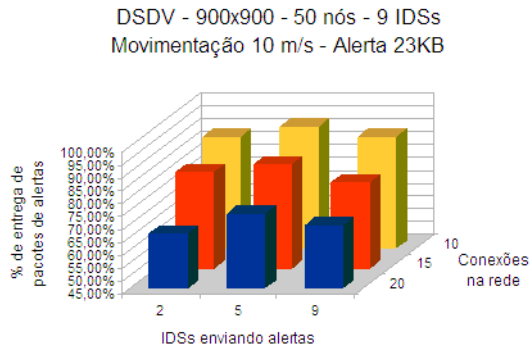
(c) DSR - movimentação 1,5 m/s



(d) DSDV - movimentação 1,5 m/s



(e) DSR - movimentação 10 m/s

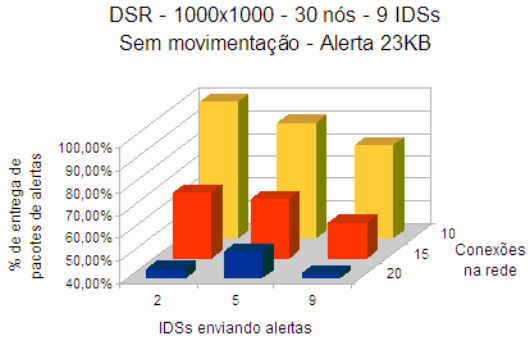


(f) DSDV - movimentação 10 m/s

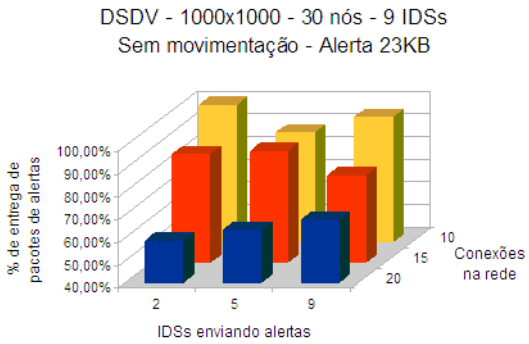
Figura 5.7: Porcentagem de entrega de pacotes no cenário de 900x900, com 50 nós e 9 IDSs, nos protocolos DSR e DSDV, enviando alertas de 23KB

com entrega de pacotes de IDS inferior a 50% nos cenários com maior carga na rede (20 conexões e 9 IDSs enviando alertas).

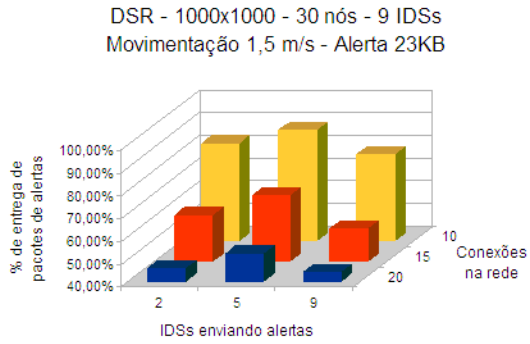
No cenário esparso e com alertas de 23K (Figura 5.8) tem-se os piores resultados com o DSDV (taxa inferior a 25%) devido a falta de conectividade. Já o DSR se comporta melhor e tem uma taxa de entrega melhor do que o DSDV (Figuras 5.8(e) e 5.8(f)).



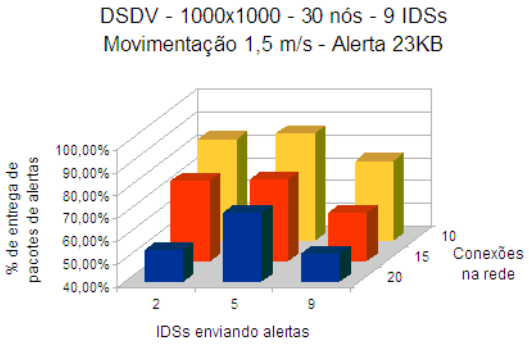
(a) DSR - sem movimentação



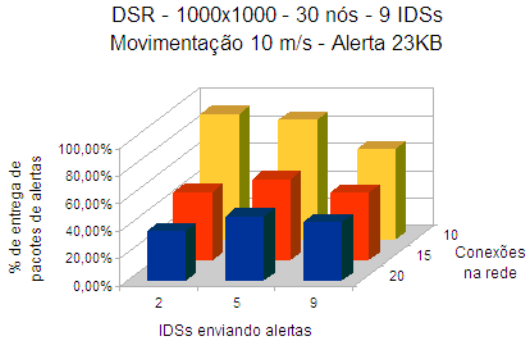
(b) DSDV - sem movimentação



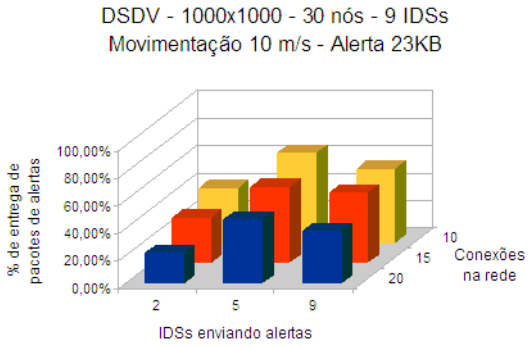
(c) DSR - movimentação 1,5 m/s



(d) DSDV - movimentação 1,5 m/s



(e) DSR - movimentação 10 m/s



(f) DSDV - movimentação 10 m/s

Figura 5.8: Porcentagem de entrega de pacotes no cenário de 1000x1000, com 30 nós e 9 IDSs, nos protocolos DSR e DSDV, enviando alertas de 23KB

A partir desses resultados apresentados, chega-se a conclusão que em cenários com conectividade, o DSDV se comporta melhor que o DSR, e em cenários com problemas de conectividade e alta mobilidade, o DSR consegue uma taxa de entrega de pacotes melhor que o DSDV. Sendo assim, não foi fixado para a arquitetura um único protocolo de roteamento como o ideal para a transferência de alertas. A utilização de um protocolo ruim

para um determinado cenário pode comprometer a entrega dos alertas e conseqüentemente a opinião de outros nós sobre um determinado nó. Isso não prejudica a arquitetura em si, já que o IDS em cada nó pode operar de maneira independente, mesmo sem receber alertas de outros nós.

5.5 Consumo de Energia da Arquitetura Proposta

Para analisar o consumo de energia da arquitetura proposta é preciso contabilizar duas questões:

- compressão e descompressão dos alertas;
- esteganografia e desesteganografia dos alertas comprimidos.

A questão de compressão e descompressão é explorada em [Barr e Asanović 2003], onde é descrito o consumo de energia de cinco algoritmos, testados em um Compaq iPAQ com processador 233MHz StrongARM SA-110, rede sem fio Enterasys 802.11b, 32MB de DRAM, rodando ARM/Linux 2.4.2-rmk1-np1-hh2, instalado em 4MB de memória *flash*.

Os testes foram realizados com 2 tipos de arquivos para compressão: "dados *web*" (HTML, XML, Javascript e CSS) e "dados em texto" (primeiro MB de livros). Além disso, a compressão e descompressão foi analisada com cinco aplicativos populares: bzip2, compress, LZO, PPMd e gzip, utilizando parâmetros característicos de cada algoritmo e utilizando 1MB de dados (*web* ou texto). Em resumo, o gzip nível 1 foi a aplicação que conseguiu comprimir e descomprimir dados *web* com melhor custo/benefício.

Por se assemelhar ao formato dos dados dos alertas (em XML), considera-se nesse trabalho os valores de compressão dos dados *web*, com compressão e descompressão feita pelo gzip no seu nível mais rápido (nível 1), por consumir menos energia e ainda sim ter uma boa compressão.

A Tabela 5.6 apresenta os dados de consumo de energia na compressão e descompressão de dados, baseados em [Barr e Asanović 2003]. Para os tamanhos dos alertas descritos no trabalho, tem-se um consumo de **1,506mJ** para o maior alerta e **0,675mJ** para o menor alerta.

Com relação a esteganografia, durante a realização dessa pesquisa, não foram encontrados trabalhos relacionados ao consumo de energia de algoritmos de esteganografia.

Tabela 5.6: Consumo de energia de compressão e descompressão

Tamanho	Compressão	Descompressão	Total
1MB	0,44J	0,2J	0,64J
2,409KB	1,035mJ	0,471mJ	1,506mJ
1,079KB	0,464mJ	0,211mJ	0,675mJ

Como apresentado no Capítulo 3, esteganografia e marca d'água trabalham com conceitos semelhantes, mas com objetivos diferentes. Assim, esse trabalho apresenta cálculos de energia extraídos de algoritmos de marca d'água descritos em [Kejariwal et al. 2004].

O trabalho de [Kejariwal et al. 2004] inspeciona o consumo de energia de dez algoritmos de marca d'água (descritos por Bruyndonckx [Bruyndonckx et al. 1995], Corvi [Corvi e Nicchiotti 1997], Cox [Cox et al. 1997], Dugad [Dugad et al. 1998], Fridrich [Fridrich 1998], Kim [Kim e Moon 1999], Koch [Koch e Zhao 1995], Xia [Xia et al. 1998], Xie [Xie e Arce 1998], Wang [Wang et al. 1998] e Zhu [Zhu et al. 1999]).

Dentre os algoritmos apresentados, este trabalho baseou os cálculos nos que trabalham com DCT (Cox, Fridrich e Koch). O consumo de energia de cada algoritmo é apresentado na Tabela 5.7. Esses valores representam a inserção/remoção de uma marca d'água de 32Bytes em uma imagem de 512x512 pixels (total de 262.144 pixels). Com isso, a cada bloco DCT é inserida uma marca de 32Bytes, representando um total de 128KBytes inseridos na imagem.

Tabela 5.7: Consumo de energia dos algoritmos de marca d'água [Kejariwal et al. 2004]

Algoritmo	Inserir Marca (J)	Extrair Marca (J)
Cox	126	121
Fridrich	196	191
Koch	2,19	0,61

Baseados nos valores das Tabelas 5.3, 5.4 e 5.7, são apresentadas as Tabelas 5.8 e 5.9 com o consumo de energia da inserção de alertas em imagens, usando os algoritmos referenciados.

Tabela 5.8: Consumo de energia da para inserir alertas nas imagens

Algoritmo	Alerta (Bytes)	Consumo (mJ)		Compressão (mJ)	Consumo Final (mJ)	
		outguess	steghide		outguess	steghide
Cox	2409	113,075	143,111	1,035	114,110	144,146
	1079	50,647	64,100	0,464	51,111	64,564
Fridrich	2409	175,895	222,617	1,035	176,930	223,652
	1079	78,784	99,711	0,464	79,248	100,175
Koch	2409	1,965	2,487	1,035	3,000	3,522
	1079	0,880	1,114	0,464	1,344	1,578

Tabela 5.9: Consumo de energia da para remover alertas das imagens

Algoritmo	**Alerta**	**Consumo (mJ)**		**Descompressão**	**Consumo Final (mJ)**	
	(Bytes)	**outguess**	**steghide**	**(mJ)**	**outguess**	**steghide**
Cox	2409	108,588	137,432	0,471	109,059	137,903
	1079	48,637	61,556	0,211	48,848	61,767
Fridrich	2409	171,408	216,938	0,471	171,879	217,409
	1079	76,774	97,167	0,211	76,985	97,378
Koch	2409	0,547	0,693	0,471	1,018	1,164
	1079	0,245	0,310	0,211	0,456	0,521

Pode-se observar que, para cada alerta, o algoritmo de Koch é o que consome menos energia em todos os casos (4,018mJ para o maior alerta e 1,8mJ para o menor, utilizando a ferramenta *outguess*), considerando o processo de compressão, esteganografia, desesteganografia e descompressão. Isso se deve ao fato desse algoritmo utilizar uma técnica muito simples de inserção de marca d'água, conseqüentemente mais frágil e rápido. No pior caso tem-se o algoritmo de Fridrich com um total de 441,061mJ para o maior alerta e 197,553mJ para o menor, utilizando a ferramenta *steghide*. Assim, considerando somente alertas de maior tamanho e utilizando o *steghide*, a geração do alerta esteganografado consome **441,061mJ** no pior caso e **4,686mJ** no melhor caso. Já com o *outguess*, o consumo é **348,809mJ** no pior caso e **4,018mJ** no melhor caso.

De acordo com [Karri e Mishra 2002], um sensor típico utilizando um chip Motorola MC68328 consome 21,5mJ para transmitir e 14,3mJ para receber 1024bits. Estes valores também são usados no cálculo adicional de energia necessário para a proposta. Como os alertas possuem 16KB (*outguess*) e 23KB (*steghide*), tem-se um total de 2,752J para transmitir com o *outguess* e 3,956J para transmitir com o *steghide*, e um total de 1,8304J para receber com o *outguess* e 2,6312J para receber com o *steghide*. A Tabela 5.10 apresenta o consumo total de envio e recebimento dos alertas esteganografados (compressão + esteganografia + transmissão + recepção + desesteganografia + descompressão).

Tabela 5.10: Consumo de energia total para enviar e receber alertas esteganografados

Algoritmo	Alerta	Transmissor (J)		Receptor (J)		Total (J)	
	(Bytes)	outguess	steghide	outguess	steghide	outguess	steghide
Cox	2409	2,866	4,100	2,861	4,094	5,727	8,194
	1079	2,803	4,021	2,801	4,018	5,604	8,038
Fridrich	2409	2,929	4,180	2,924	4,173	5,853	8,353
	1079	2,831	4,056	2,829	4,053	5,660	8,110
Koch	2409	2,755	3,960	2,753	3,957	5,508	7,917
	1079	2,753	3,958	2,752	3,957	5,506	7,914

5.6 Comparação da arquitetura proposta com um mecanismo de autenticação de rotas

O trabalho descrito em [Brazil 2007] propõe uma arquitetura de Infraestrutura de Chave Pública de alta disponibilidade e robusta de forma a minimizar os ataques do tipo buraco negro e falsificação de identidade contra redes *ad hoc*, chamada ICPAH. A arquitetura se propõe a trocar um número de mensagens de controle reduzido aumentando assim a sua escalabilidade e desempenho. Para atingir este objetivo, os diversos nós de uma rede *ad hoc* utilizam o serviço distribuído de certificação digital para autenticar e cifrar suas mensagens, particularmente os protocolos de acesso ao meio e os de roteamento devem usar a autenticação e cifragem para se protegerem dos ataques citados anteriormente garantindo assim confidencialidade, autenticidade e integridade na troca de mensagens.

Este trabalho é baseado em [Zhou e Haas 1999], que usa o conceito de limite criptográfico (*threshold cryptography*), utilizando cifragem por limiar para dividir a chave privada do serviço de ICP entre as diversas autoridades certificadoras (AC) em um esquema de certificação cruzada. O trabalho descrito em [Brazil 2007] propõe um protocolo que diminui o número de mensagens trocadas entre as ACs e entre os nós e as ACs.

O presente trabalho faz uma comparação entre o consumo de energia do envio de mensagens de IDS com a energia consumida na proposta descrita em [Brazil 2007], visando identificar o limite do número de mensagens enviadas por IDSs em uma rede *ad hoc*, sem que haja um consumo de energia maior do que uma proposta com criptografia. No caso dos alertas do IDS o consumo de energia é constante visto que a geração do alerta é constante. O impacto na energia consumida se deve ao sistema usado para esteganografar o alerta (*outguess* ou *steghide*), pois cada sistema necessita transmitir um tamanho de imagem.

Apesar das propostas não terem o mesmo objetivo, essa comparação se torna válida a fim de mostrar cenários onde a criptografia se torna muito custosa, apresentando o IDS proposto como uma alternativa à sua utilização.

5.6.1 Consumo de energia do ICPAH

De acordo com [Brazil 2007], o consumo de energia para o estabelecimento de rotas é de 164,22mJ relativo ao transmissor, 102,91mJ relativo ao receptor e 0,91mJ relativo ao serviço ICPAH (trabalho das ACs), para um conjunto de 9 ACs. Isso gera um total de

268,04mJ para cada autenticação de rota.

As Tabelas 5.11, 5.12 e 5.13 apresentam o consumo de energia de acordo com o número de autenticações em cada protocolo de roteamento, de cada cenário já descrito na Seção 5.4.

Tabela 5.11: Consumo de energia do ICPAH para o cenário de 670x670m^2

Velocidade (m/s)	**Conexões**	**Autenticações**		**Energia (J)**	
		DSR	**DSDV**	**DSR**	**DSDV**
0	10	270	1084	72,370	290,554
	15	376	1111	100,782	297,791
	20	616	1165	165,112	312,265
1,5	10	346	1918	92,741	514,098
	15	443	1960	118,741	525,355
	20	816	1945	218,719	521,335
10	10	1190	3065	318,966	821,538
	15	1443	3100	386,779	830,919
	20	1647	3069	441,459	822,610

Tabela 5.12: Consumo de energia do ICPAH para o cenário de 900x900m^2

Velocidade (m/s)	**Conexões**	**Autenticações**		**Energia (J)**	
		DSR	**DSDV**	**DSR**	**DSDV**
0	10	202	679	54,144	181,998
	15	381	706	102,123	189,235
	20	595	760	159,483	203,709
1,5	10	511	1355	136,968	363,192
	15	818	1406	219,255	376,862
	20	1973	1422	528,840	381,151
10	10	1495	2306	400,717	618,097
	15	2345	2339	628,550	626,942
	20	2802	2346	751,044	628,818

Tabela 5.13: Consumo de energia do ICPAH para o cenário de 1000x1000m^2

Velocidade (m/s)	**Conexões**	**Autenticações**		**Energia (J)**	
		DSR	**DSDV**	**DSR**	**DSDV**
0	10	351	317	94,081	84,968
	15	439	319	117,669	85,504
	20	772	352	206,926	94,350
1,5	10	364	445	97,566	119,277
	15	476	464	127,586	124,370
	20	1051	505	281,708	135,359
10	10	978	966	262,142	258,925
	15	1060	966	284,121	258,925
	20	1140	1000	305,564	268,038

Da Tabela 5.11 conclui-se que, para o cenário mais denso, o protocolo DSDV necessita de um número muito maior de autenticações, chegando a usar 5,5 vezes mais autenticações, no mesmo cenário, do que com o protocolo DSR. Conclui-se também que o protocolo DSDV se mantém mais constante ao número de autenticações com relação a movimentação

e ao tráfego da rede do que o protocolo DSR, que tem um aumento significativo quando o tráfego da rede aumenta.

A Tabela 5.12 apresenta o cenário denso, com resultados semelhantes ao da tabela 5.11, exceto pelo fato de que em cenários com tráfego e movimentações altas o protocolo DSR necessita de um número maior de autenticações que o protocolo DSDV.

Já no cenário esparso (Tabela 5.13), o número de autenticações nos 2 protocolos é praticamente a mesma, somente com um número maior de autenticações necessárias pelo DSR nos cenários com maior tráfego na rede (15 e 20 conexões existentes).

5.6.2 Comparação entre o consumo de energia das propostas

Com base nos dados apresentados nas Tabelas 5.10, 5.11, 5.12 e 5.13, as Tabelas 5.14, 5.15 e 5.16 apresentam a comparação de energia entre as duas propostas, em função do número de alertas que podem ser enviados, no mesmo cenário. Como para a arquitetura proposta a variação entre a energia consumida para a geração do maior alerta e o menor alerta é em média menor do que 150mJ, usa-se o pior caso (maior alerta) usando-se os algoritmos de Fridrich (maior consumo) e Koch (menor consumo) combinados pelo uso do *outguess* e do *steghide*.

Tabela 5.14: Comparação do consumo de energia das propostas, em função do número de alertas no cenário de 670x670m^2

Velocidade	Conexões	ICPAH (J)	Fridrich (Alertas)		Koch (Alertas)		ICPAH (J)	Fridrich (Alertas)		Koch (Alertas)	
		DSR	outguess	steghide	outguess	steghide	DSDV	outguess	steghide	outguess	steghide
0	10	72,370	12	9	13	9	290,554	50	35	53	37
	15	100,782	17	12	18	13	297,791	51	36	54	38
	20	165,112	28	20	30	21	312,265	53	37	57	39
1,5	10	92,741	16	11	17	12	514,098	88	62	93	65
	15	118,741	20	14	22	15	525,355	90	63	95	66
	20	218,719	37	26	40	28	521,335	89	62	95	66
10	10	318,966	54	38	58	40	821,538	140	98	149	104
	15	386,779	66	46	70	49	830,919	142	99	151	105
	20	441,459	75	53	80	56	822,610	141	98	149	104

Pode-se concluir que, em cenários muito densos e com alta mobilidade, o alto consumo de energia do ICPAH favorece a utilização da presente proposta, onde pode-se enviar até 149 alertas (comparando com o ICPAH usando DSDV, com velocidade de 10m/s e 20 conexões já existentes). Isso gera uma média de 3,31 alertas enviados por cada um dos 9 IDSs, número esse apresentado na Tabela 5.2 como a freqüência média de alertas por minuto, considerado um cenário muito hostil.

Também conclui-se que a diferença entre o uso do algoritmo de Koch e Fridrich é

Tabela 5.15: Comparação do consumo de energia das propostas, em função do número de alertas no cenário de 900x900m^2

Velocidade	Conexões	ICPAH (J)	Fridrich (Alertas)		Koch (Alertas)		ICPAH (J)	Fridrich (Alertas)		Koch (Alertas)	
		DSR	outguess	steghide	outguess	steghide	DSDV	outguess	steghide	outguess	steghide
0	10	54,144	9	6	10	7	181,998	31	22	33	23
	15	102,123	17	12	19	13	189,235	32	23	34	24
	20	159,483	27	19	29	20	203,709	35	24	37	26
1,5	10	136,968	23	16	25	17	363,192	62	43	66	46
	15	219,255	37	26	40	28	376,862	64	45	68	48
	20	528,840	90	63	96	67	381,151	65	46	69	48
10	10	400,717	68	48	73	51	618,097	106	74	112	78
	15	628,550	107	75	114	79	626,942	107	75	114	79
	20	751,044	128	90	136	95	628,818	107	75	114	79

Tabela 5.16: Comparação do consumo de energia das propostas, em função do número de alertas no cenário de 1000x1000m^2

Velocidade	Conexões	ICPAH (J)	Fridrich (Alertas)		Koch (Alertas)		ICPAH (J)	Fridrich (Alertas)		Koch (Alertas)	
		DSR	outguess	steghide	outguess	steghide	DSDV	outguess	steghide	outguess	steghide
0	10	94,081	16	11	17	12	84,968	15	10	15	11
	15	117,669	20	14	21	15	85,504	15	10	16	11
	20	206,926	35	25	38	26	94,350	16	11	17	12
1,5	10	97,566	17	12	18	12	119,277	20	14	22	15
	15	127,586	22	15	23	16	124,370	21	15	23	16
	20	281,708	48	34	51	36	135,359	23	16	25	17
10	10	262,142	45	31	48	33	258,925	44	31	47	33
	15	284,121	49	34	52	36	258,925	44	31	47	33
	20	305,564	52	37	55	39	268,038	46	32	49	34

mínima (9 alertas extras utilizando o *outguess* e 6 utilizando o *steghide*, considerando a maior diferença). Já a diferença entre o uso do *outguess* ou *steghide* tem um impacto considerável (46 alertas no pior caso), visto que o valor se aproxima do número total de alertas enviados em cenários moderados (5 IDSs enviando alertas). Essa diferença se deve ao tamanho da imagem necessária pelo *steghide* para transportar tais alertas e em virtude do maior consumo de energia ser a própria transmissão via rede.

Em cenários sem movimentação, ou com pouca movimentação, a presente proposta somente se torna utilizável em cenários brandos, com poucos alertas sendo gerados. A medida que o tráfego da rede aumenta, as autenticações do ICPAH aumentam e tornam a proposta deste trabalho mais viável. No cenário esparso, o número de autenticações necessárias pelo ICPAH é baixo, conseqüentemente, a presente proposta somente poderia ser utilizada em cenários médios ou brandos, considerando o tráfego já existente na rede mais alto.

5.7 Conclusão

Foram apresentados os resultados dos testes de esteganografia utilizando os aplicativos *steghide* e *outguess* identificando qual o tamanho ideal de uma imagem JPEG para transportar os tipos de alertas conforme o padrão IDMEF. Além disso, foram criadas imagens com o aplicativo *convert* para identificar qual padrão de imagem consegue transportar os alertas.

Também foram explanados os testes com um grupo de imagens reais para identificar se um tráfego real de imagens em uma rede pode servir de stego-objeto para envio dos alertas, bem como os testes com o Prelude-IDS a fim de identificar o tamanho médio de um alerta no padrão IDMEF.

Nas Seções 5.4 a 5.6 foram apresentados os resultados obtidos das simulações do envio de alertas dos IDSs em diversos cenários, além de uma comparação com um trabalho que apresenta uma proposta alternativa utilizando rotas autenticadas para garantir a segurança na rede. Apresentou-se também o estudo sobre o consumo de energia inserido pela proposta, este também sendo comparado com a proposta ICPAH.

De acordo com os cálculos apresentados, conclui-se que a escolha do algoritmo de esteganografia não tem um grande impacto no consumo final de energia. O maior impacto é do tamanho da stego-imagem para a transportar os alertas.

Outro fator importante é a compressão do alerta antes da esteganografia. Isso representa uma economia de energia, visto que a imagem necessária para a esteganografia é reduzida.

Dos resultados apresentados conclui-se que é viável a utilização da presente proposta em determinados cenários, principalmente com alta mobilidade e alto tráfego, pelo ponto de vista do consumo de energia, onde esta proposta se mostra mais econômica que o ICPAH.

Capítulo 6

Considerações Finais

A segurança em redes sem fio *ad hoc* representa um grande desafio em virtude da natureza desses tipos de redes. Nesses ambientes, os sistemas de detecção de intrusão distribuídos são indispensáveis para aumentar a segurança de todos os nós.

Este trabalho apresenta um modelo de IDS para redes *ad-hoc* baseado em *esteganografia* e *reputação* como alternativa à utilização de um canal seguro para a transmissão de alertas entre IDSs. A utilização deste canal seguro em uma rede *ad-hoc* implica no uso de PKI e criptografia, ambos custosos para nós com pouca capacidade de processamento e energia de bateria.

Como contribuições iniciais, este trabalho faz uma revisão sobre os modelos de arquitetura para sistemas de detecção de intrusão para redes *ad hoc*, além de apresentar uma revisão sobre esteganografia e marca d'água digital, resultado de um minicurso apresentado no SBSEG 2007 [Julio et al. 2007].

As contribuições deste trabalho foram:

1. apresentação de uma arquitetura para um sistema de detecção de intrusão que troca mensagens com outros sistemas de detecção, visando a substituição de um canal criptográfico pela transmissão de alertas esteganografados. Devido a ausência de autenticação e de criptografia, é usado um mecanismo de reputação para ponderar os alertas recebidos por outros IDSs, com o objetivo de ignorar alertas recebidos de um nó comprometido;

2. testes realizados apresentaram a identificação dos tipos de alertas que são enviados por um IDS, além dos formatos e freqüência desses alertas. Também foram apresentados os testes realizados com a esteganografia em imagens, identificando que tipo de imagem e qual formato de imagem pode servir como uma stego-imagem para o

IDS. Além disso, verificou-se a possível utilização do próprio tráfego real de imagens de uma rede como meio de transporte dos alertas;

3. avaliação sobre a probabilidade de entrega de pacotes do IDS, simulada no NS-2 com o objetivo de definir em que tipo de cenário a arquitetura pode ser aplicada;

4. estudo sobre o consumo de energia da arquitetura. Esse consumo leva em consideração o processo de compactar, esteganografar e transmitir o alerta, bem como o processo inverso;

5. comparação da arquitetura proposta com uma arquitetura de infra-estrutura de chave pública para autenticação de rotas em redes *ad hoc*, objetivando encontrar cenários onde o consumo de energia da utilização do IDS seja menor do que com o uso de criptografia.

Como resultados obtidos, tem-se uma freqüência média de envio de 3,31 alertas por minuto por IDS, identificando a média dos tipos de alertas no formato IDMEF, com cada alerta com tamanho máximo de 2409Bytes. Estes dados foram colhidos a partir de testes realizados com o IDS Prelude em uma rede real que gera alertas no formato IDMEF. Também identifica-se o tamanho da imagem para transportar alertas em 16.200 *pixels*(23KBytes), suficiente para transmitir qualquer tipo de alerta, já considerando a imagem esteganografada, utilizando o software *steghide*. Já para o software *outguess*, tem-se uma imagem de 12.800 pixels (16KBytes). Os testes foram realizados criando imagens de formatos e tamanhos diferentes, inserindo alertas de diferentes tamanhos, compactados ou não.

Com base nesses resultados, chega-se a um *overhead* de 1175,91Bytes por segundo (*steghide*) e 761,59Bytes (*outguess*) inserido por um IDS na rede.

Um outro resultado obtido pelo presente trabalho é a capacidade de imagens reais do tráfego colhido por um *webcache* transportarem os alertas do IDS. Obteve-se um valor entre 2,6% e 5% de imagens reais que conseguem transportar alertas. Apesar de uma taxa muito baixa, dependendo do tráfego da rede, esse valor se aproxima com a quantidade de alertas totais transmitindo, sendo possível aproveitar o tráfego de imagens já existente na rede e diminuir o *overhead* de criação de imagens da proposta.

Com base nas simulações executadas no NS-2, obtem-se os resultados da taxa de entrega de pacotes do sistema de IDS, em diversos cenários: com ou sem movimentação; com carga existente baixa, média e alta; esparsos, densos e muito densos; com IDSs

enviando alertas em cenários brandos, médios e hostis. De acordo com as simulações, a taxa de entrega de pacotes em cenários com alta mobilidade e com a carga da rede também alta, o protocolo de roteamento DSR se comporta melhor que o protocolo DSDV. E em cenários mais estáticos e com carga mais baixa, o DSDV tem uma qualidade de entrega melhor. Nos cenários com carga alta, a taxa de entrega de pacotes cai, podendo compromenter a troca de informações entre IDSs, mas não prejudicando a arquitetura em si, pois os nós podem operar independentemente, coletando informações de sensores locais.

Sobre a energia consumida, os cálculos mostram que o consumo de energia dos algoritmos apresentados tem uma variação da ordem de mJ, não impactando no consumo final, visto que o maior consumo é da transmissão dos alertas. No melhor caso, a arquitetura proposta consome aproximadamente 5,5J para enviar e receber alertas esteganografados. No pior caso, 8,36J aproximadamente. Vale ressaltar que são valores constantes, já que o tamanho da stego-imagem é o principal fator de influência no consumo de energia. Esses valores devem ser levados em conta, já que atualmente a maioria das baterias de notebooks são do tipo AA com tecnologia "Lithium Ion" e possuem capacidade de 4.000mAh(Mili-Amper/hora). Estas baterias podem trabalhar em até 14,8 Volts totalizando uma carga total de 59.200 Joules ou 59,2KJ (dados obtidos a partir de um notebook Sony Vaio). Já os sensores típicos segundo [Karri e Mishra 2002] possuem capacidade de energia de aproximadamente 26KJ. A autonomia de uso varia muito de acordo com as aplicações que rodam nas máquinas. No caso do uso típico de um notebook, a autonomia da bateria é de 4,5 horas.

Por fim, os resultados obtidos da comparação da presente proposta com a arquitetura ICPAH, onde é apresentado o consumo de energia relativo as duas propostas. Esses resultados mostram que em cenários com alta mobilidade e também com alto tráfego, há uma necessidade de um número relevante de autenticações, favorecendo a utilização da arquitetura aqui apresentada. Em cenários com pouca movimentação, a utilização desta proposta passa a ser viável somente com um tráfego alto na rede.

É importante salientar que a utilização de esteganografia como alternativa à criptografia não deixa o sistema mais inseguro, pois algoritmos genéricos de esteganálise são frágeis na descoberta de imagens esteganografadas, visto que cada método de esteganografia requer um algoritmo diferente de detecção [Provos e Honeyman 2003], ou seja, o processamento gasto na análise de tais mensagens é muito custoso para nós com baixo poder de processamento ou pouca bateria, sendo assim inviável a esteganálise com diversos

algoritmos diferentes [Kejariwal et al. 2004].

Como trabalho futuro propõe-se a implementação da arquitetura aqui apresentada, assim como a investigação de sua utilização em outros tipos de redes *ad hoc*, como em redes *MESH* [Akyildiz et al. 2005b], VANETs (*Vehicular Ad hoc NETworks*) [Golle et al. 2004], *Wireless Underground Sensor Networks* [Akyildiz e Stuntebeck 2006] e *Wireless Underwater Sensor Networks* [Akyildiz et al. 2005a].

Referências Bibliográficas

[Akyildiz et al. 2005a] Akyildiz, I. F., Pompili, D., e Melodia, T. (2005a). Underwater acoustic sensor networks: research challenges. *Ad Hoc Networks*, 3(3):257–279.

[Akyildiz e Stuntebeck 2006] Akyildiz, I. F. e Stuntebeck, E. P. (2006). Wireless underground sensor networks: Research challenges. *Ad Hoc Networks*, 4(6):669–686.

[Akyildiz et al. 2005b] Akyildiz, I. F., Wang, X., e Wang, W. (2005b). Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487.

[Albers et al. 2002] Albers, P., Camp, O., Percher, J., Jouga, B. Ludovic, M., e Puttini, R. (2002). Security in ad hoc networks: A general intrusion detection architecture enhancing trust based approaches. Em *First International Workshop on Wireless Information Systems, 4th International Conference on Enterprise Information Systems*.

[Avcibas et al. 2001] Avcibas, I., Memon, N., e Sankur, B. (2001). Steganalysis based on image quality metrics. Em *Proceedings of the Fourth Workshop on Multimedia Signal Processing*, pp. 517–522, USA. IEEE.

[Bace e Mell 2001] Bace, R. e Mell, P. (2001). Nist special publication on intrusion detection system. Technical report, NIST (National Institute of Standards and Technology). Special Publication 800-31.

[Barr e Asanović 2003] Barr, K. e Asanović, K. (2003). Energy aware lossless data compression. Em *MobiSys '03: Proceedings of the 1st international conference on Mobile systems, applications and services*, pp. 231–244, New York, NY, USA. ACM.

[Bassia e Pitas 1998] Bassia, P. e Pitas, I. (1998). Robust audio watermarking in the time domain. Em *9th European Signal Processing Conference (EUSIPCO'98)*, pp. 25–28, Island of Rhodes, Greece.

[Bhargava e Agrawal 2001] Bhargava, S. e Agrawal, D. P. (2001). Security enhancements in aodv protocol for wireless ad hoc networks. Em *VTC 2001 Fall*, volume 4, pp. 2143–2147.

[Boney et al. 1996] Boney, L., Tewfik, A. H., e Hamdy, K. N. (1996). Digital watermarks for audio signals. Em *International Conference on Multimedia Computing and Systems*, pp. 473–480.

[Brazil 2007] Brazil, W. G. (2007). Protegendo redes ad hoc com certificados digitais e limite criptográfico. Dissertação de Mestrado, Instituto de Computação, Universidade Federal Fluminense, Niterói, Brasil.

[Bruyndonckx et al. 1995] Bruyndonckx, O., Quisquater, J.-J., e Macq, B. (1995). Spatial method of copyright labeling of digital images. Em *IEEE Workshop on Nonlinear Images/Signal Processing, Thessal*.

[Buccigrossi e Simoncelli 1999] Buccigrossi, R. W. e Simoncelli, E. P. (1999). Image compression via joint statistical characterization in the wavelet domain. *IEEE Trans Image Proc*, 8(12):1688–1701.

[Corvi e Nicchiotti 1997] Corvi, M. e Nicchiotti, G. (1997). Wavelet based image watermarking for copyright protection. Em *Scandinavian Conference on Image Analysis*.

[Cox et al. 1997] Cox, I., Kilian, J., Leighton, T., e Shamoon, T. (1997). Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687.

[Curry e Debar 2002] Curry, D. e Debar, H. (2002). Em *Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition*.

[Deraison 1999] Deraison, R. (1999). The nessus project. http://www.nessus.org/ documentation.html.

[Deutsch 1996] Deutsch, L. P. (1996). Deflate compressed data format specification. *RFC 1951*.

[Duda et al. 2000] Duda, R. O., Hart, P. E., e Stork, D. G. (2000). *Pattern Classification (2nd Edition)*. Wiley-Interscience.

[Dugad et al. 1998] Dugad, R., Ratakonda, K., e Ahuja, N. (1998). A new wavelet-based scheme for watermarking images. Em *IEEE International Conference on Image Processing*, pp. 419–423.

[Dumitrescu e Wu 2002] Dumitrescu, S. e Wu, X. (2002). Steganalysis of lsb embedding in multimedia signals. Em *Proceedings of the Intl. Conference on Multimedia and Exp*, volume 3, pp. 581–584, USA. IEEE.

[Filho et al. 2005] Filho, E. B. L., da Silva, E. A. B., de Carvalho, M. B., e Waldir S. S. Júnior, J. K. (2005). Electrocardiographic signal compression using multiscale recurrent patterns. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 52(12):2739–2753.

[Fridrich 1998] Fridrich, J. (1998). Combining low-frequency and spread spectrum watermarking. Em *SPIE Symposium on Optical Science, Engineering and Instrumentation, San Diego, USA*.

[Fridrich e Goljan 2002] Fridrich, J. e Goljan, M. (2002). Practical steganalysis of digital images - state of the art.

[Friedmann 1993] Friedmann, G. L. (1993). The trustworthy digital camera: Restoring credibility to the photographic image. *IEEE Transactions on Consumer Electronics*, 39(4):905–910.

[Golle et al. 2004] Golle, P., Greene, D., e Staddon, J. (2004). Detecting and correcting malicious data in vanets. Em *VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pp. 29–37, New York, NY, USA. ACM.

[Gonzalez e Woods 2002] Gonzalez, R. C. e Woods, R. E. (2002). *Digital Image Processing*. Prentice-Hall, Boston, MA, USA, 2nd edition.

[Hart et al. 2004] Hart, S. V., Ashcroft, J., e Daniels, D. J. (2004). Forensic examination of digital evidence: a guide for law enforcement. Technical report, Department of Justice - Office of Justice Programs, USA. Technical Report NCJ 199408.

[Hartung e Girod 1996] Hartung, F. e Girod, B. (1996). Digital watermarking of raw and compressed video. Em *Proc. European EOS/SPIE Symposium on Advanced Imaging and Network Technologies*, Berlin, Germany.

[Haselton 2000] Haselton, B. (2000). A protocol that uses steganography to circumvent network level censorship. Em *DEF CON*.

[Hetzl 2003] Hetzl, S. (2003). Steghide software.

[Hide e Seek 2007] Hide e Seek (2007). Hide and seek software.

[Hirohisa 2007] Hirohisa, H. (2007). Crocus: a steganographic filesystem manager. Em *ASIACCS '07: Proceedings of the 2nd ACM symposium on Information, computer and communications security*, pp. 344–346, New York, NY, USA. ACM Press.

[ImageMagick 2007] ImageMagick (2007). Convert software.

[Jean-loup e Mark 2003] Jean-loup e Mark (2003). Gzip software.

[Johnson e Jajodia 1998] Johnson, N. F. e Jajodia, S. (1998). Exploring steganography: Seeing the unseen. *IEEE Computer*, 31(2):26–34.

[Jphide e Seek 2007] Jphide e Seek (2007). Jphide and seek software.

[Julio et al. 2007] Julio, E. P., Brazil, W., e Albuquerque, C. (2007). Esteganografia e suas aplicações. Em *SBSEG'2007: Mini Curso do VII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pp. 54–102, Rio de Janeiro, Brasil. Sociedade Brasileira de Computação.

[Kachirski e Guha 2002] Kachirski, O. e Guha, R. (2002). Intrusion detection using mobile agents in wireless ad hoc networks. Em *KMN '02: Proceedings of the IEEE Workshop on Knowledge Media Networking*, page 153, Washington, DC, USA. IEEE Computer Society.

[Kahn 1996] Kahn, D. (1996). The history of steganography. Em *Proceedings of the First International Workshop*, Cambridge, UK.

[Kalker et al. 1999] Kalker, T., Depovere, G., Haitsma, J., e Maes, M. J. (1999). Video watermarking system for broadcast monitoring. volume 3657, pp. 103–112. SPIE.

[Karri e Mishra 2002] Karri, R. e Mishra, P. (2002). Minimizing energy consumption of secure wireless session with qos constraints. Em *Int. Conf. Communications*, pp. 20–53.

[Kejariwal et al. 2004] Kejariwal, A., Gupta, S., Nicolau, A., Dutt, N., e Gupta, R. (2004). Proxy-based task partitioning of watermarking algorithms for reducing energy consumption in mobile devices. Em *DAC '04: Proceedings of the 41st annual conference on Design automation*, pp. 556–561, New York, NY, USA. ACM Press.

[Kim 2000] Kim, H. (2000). Stochastic model based audio watermark and whitening filter for improved detection. Em *ICASSP '00: Proceedings of the Acoustics, Speech, and Signal Processing, 2000. on IEEE International Conference*, pp. 1971–1974, Washington, DC, USA. IEEE Computer Society.

[Kim e Moon 1999] Kim, J. R. e Moon, Y. S. (1999). A robust wavelet-based digital watermarking using level-adaptive thresholding. Em *ICIP (2)*, pp. 226–230.

[Koch e Zhao 1995] Koch, E. e Zhao, J. (1995). Towards robust and hidden image copyright labeling. Em *Proc. of 1995 IEEE Workshop on Nonlinear Signal and Image Processing*, pp. 452–455, Halkidiki, Greece.

[Langelaar et al. 1998] Langelaar, G. C., Lagendijk, R. L., e Biemond, J. (December 1998). Real-time labeling of mpeg-2 compressed video. *Journal of Visual Communication and Image Representation*, 9:256–270(15).

[Li e Yu 2000] Li, X. e Yu, H. H. (2000). Transparent and robust audio data hiding in subband domain. Em *ITCC '00: Proceedings of the The International Conference on Information Technology: Coding and Computing (ITCC'00)*, page 74, Washington, DC, USA. IEEE Computer Society.

[Linnartz et al. 1999] Linnartz, J.-P., Kalker, T., e Haitsma, J. (1999). Detecting electronic watermarks in digital video. Em *ICASSP '99: Proceedings of the Acoustics, Speech, and Signal Processing, 1999. on 1999 IEEE International Conference*, pp. 2071–2074, Washington, DC, USA. IEEE Computer Society.

[Lu et al. 2000] Lu, C., Liao, H., e Chen, L. (2000). Multipurpose audio watermarking. Em *15th Int. Conf. on Pattern Recognition, Barcelona, Spain, Vol. III*, pp. 286–289.

[Marti et al. 2000] Marti, S., Giuli, T. J., Lai, K., e Baker, M. (2000). Mitigating routing misbehavior in mobile ad hoc networks. Em *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 255–265, New York, NY, USA. ACM Press.

[Marvel et al. 1999] Marvel, L. M., Jr., C. G. B., e Retter, C. T. (1999). Spread spectrum image steganography. *IEEE Transactions on Image Processing*, 8(8):1075–1083.

[Meerwald 2001] Meerwald, P. (2001). Digital image watermarking in the wavelet transform domain. Dissertação de Mestrado, Department of Scientific Computing, University of Salzburg, Austria.

[Mishra et al. 2004] Mishra, A., Nadkarni, K., e Patcha, A. (2004). Intrusion detection in wireless ad hoc networks. Em IEEE, editor, *Wireless Communication*, pp. 48–60. IEEE.

[Morris 2004] Morris, S. (2004). The future of netcrime now (1) - threats and challenges. Technical report, Home Office Crime and Policing Group, USA. Technical Report 62.

[Okazaki et al. 2002] Okazaki, Y., Sato, I., e Goto, S. (2002). A new intrusion detection method based on process profiling. Em *SAINT '02: Proceedings of the 2002 Symposium on Applications and the Internet*, pp. 82–91, Washington, DC, USA. IEEE Computer Society.

[Outgess 2007] Outgess (2007). Outgess software.

[Pereira e Pedroza 2004] Pereira, I. C. M. e Pedroza, A. C. P. (2004). Redes móveis ad hoc aplicadas a cenários militares. Em *4o Congresso Brasileiro de Computação (CBComp 2004)*, Rio Grande do Sul, Brasil.

[Petitcolas et al. 1999] Petitcolas, F. A. P., Anderson, R. J., e Kuhn, M. G. (1999). Information hiding — A survey. *Proceedings of the IEEE*, 87(7):1062–1078.

[Petitcolas e Katzenbeisser 1999] Petitcolas, F. A. P. e Katzenbeisser, S. (1999). *Information hiding techniques for steganography and digital watermarking*. Artech House Books, 1st edition.

[Popa 1998] Popa, R. (1998). An analysis of steganography techniques. Dissertação de Mestrado, The Polytechnic University of Timisoara, Timisoara, Romênia.

[Prandoni e Vetterli 1998] Prandoni, P. e Vetterli, M. (1998). Perceptually hidden data transmission over audio signals. Em *ICASSP98*, pp. 3665–3668.

[Provos 2001] Provos, N. (2001). Defending against statistical steganalysis. Em *10th USENIX Security Symposium*.

[Provos 2002] Provos, N. (2002). Outguess software, http://www.outguess.org.

[Provos 2003] Provos, N. (2003). Honeyd — A virtual honeypot daemon. Em *10th DFN-CERT Workshop,*, Hamburg, Germany.

[Provos e Honeyman 2003] Provos, N. e Honeyman, P. (2003). Hide and seek: An introduction to steganography. *IEEE Security and Privacy*, 1(3):32–44.

[Qiao e Nahrstedt 1998] Qiao, L. e Nahrstedt, K. (1998). Watermarking methods for MPEG encoded video: Towards resolving rightful ownership. Em *International Conference on Multimedia Computing and Systems*, pp. 276–285.

[Ramanujan et al. 2003] Ramanujan, R., Kudige, S., e Nguyen, T. (2003). Techniques for intrusion-resistant ad hoc routing algorithms (tiara). Em *DISCEX (2)*, pp. 98–100.

[Revelation 2007] Revelation (2007). Revelation software.

[Rocha 2006] Rocha, A. R. (2006). Randomização progressiva para esteganálise. Dissertação de Mestrado, Universidade Estadual de Campinas, Campinas, Brasil.

[Rocha 2005] Rocha, B. G. (2005). Estratégias para aumentar a confiabilidade em redes sobrepostas com nós egoístas. Dissertação de Mestrado, Universidade Federal de Minas Gerais, Belo Horizonte, Brasil.

[Rocha et al. 2006] Rocha, B. G., Almeida, V., e Guedes, D. O. (2006). Strategies to improve reliability in routing overlay networks with selfish nodes. Em *Anais do 24o. Simpósio Brasilieiro de Redes de Computadores, 2006.*

[Salomon 2000] Salomon, D. (2000). *Data Compression: The Complete Reference*. Springer, Nova Iorque, segunda edição edition.

[Sieffert et al. 2004] Sieffert, M., Forbes, R., Green, C., Popyack, L., e Blake, T. (2004). Stego intrusion detection system. *Digital Forensic Research Workshop*.

[Stegdetect 2007] Stegdetect (2007). Stegdetect software.

[Stego e Ezstego 2007] Stego e Ezstego (2007). Stego e ezstego softwares.

[StegSpy 2007] StegSpy (2007). Stegspy software.

[Su et al. 1999] Su, P.-C., Houng-Jyh, W., Mike, K., e Jay, C.-C. (1999). Digital image watermarking in regions of interest. Em *PICS*, pp. 295–300.

[Sullivan et al. 2004] Sullivan, K., Bi, Z., Madhow, U., Chandrasekaran, S., e Manjunath, B. (2004). Steganalysis of quantization index modulation data hiding. Em *IEEE International Conference on Image Processing*, pp. 1165–1168.

[Swanson et al. 1999] Swanson, M. D., Zhu, B., e Tewfik, A. H. (1999). Current state of the art - challenges and future directions for audio watermarking. Em *ICMCS, Vol. 1*, pp. 19–24.

[Swanson et al. 1998] Swanson, M. D., Zhu, B., Tewfik, A. H., e Boney, L. (1998). Robust audio watermarking using perceptual masking. *Signal Processing*, 66(3):337–355.

[The High Technology Crime Advisory Committee 2007] The High Technology Crime Advisory Committee (2007). High technology crime in california. Em *Annual Report on High Technology Crime in California*.

[Tsiftes 2007] Tsiftes, N. (2007). Poster abstract: Compressing software for energy-efficient reprogramming of wireless sensor networks. Em *The Contiki Workshop*, Kista, Sweden.

[Vandoorselaere 1998] Vandoorselaere, Y. (1998). Prelude project, http://www.prelude-ids.org.

[Wang et al. 1998] Wang, H., Su, P.-C., e Kuo, C.-C. J. (1998). Wavelet-based digital image watermarking. *Opt. Express*, 3(12):491–496.

[Wang e Wang 2004] Wang, H. e Wang, S. (2004). Cyber warfare: steganography vs. steganalysis. *Commun. ACM*, 47(10):76–82.

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

[Westfeld e Pfitzmann 2000] Westfeld, A. e Pfitzmann, A. (2000). Attacks on steganographic systems. Em *IH '99: Proceedings of the Third International Workshop on Information Hiding*, pp. 61–76, London, UK. Springer-Verlag.

[Westphal 2000] Westphal, K. (2000). Snort — A look inside an intrusion detection system. *Sys Admin: The Journal for UNIX Systems Administrators*, 9(9):46, 48, 50, 52–53.

[Wichmann 2006] Wichmann, R. (2006). Samhain file integrity / intrusion detection system.

[Xia et al. 1998] Xia, X., Boncelet, C., e Arce, G. (1998). Wavelet transform based watermark for digital images. *Opt. Express*, 3(12):497–511.

[Xie e Arce 1998] Xie, L. e Arce, G. (1998). Joint wavelet compression and authentication watermarking. Em *IEEE International Conference on Image Processing*, volume 2, pp. 427–431.

[Xu et al. 2003] Xu, R., Li, Z., Wang, C., e Ni, P. (2003). Impact of data compression on energy consumption of wireless-networked handheld devices. Em *ICDCS*, pp. 302–311. IEEE Computer Society.

[Zhang et al. 2004] Zhang, X., Li, C., e Zheng, W. (2004). Intrusion prevention system design. Em *CIT*, pp. 386–390. IEEE Computer Society.

[Zhang et al. 2003] Zhang, Y., Lee, W., e Huang, Y.-A. (2003). Intrusion detection techniques for mobile wireless networks. *Wireless Networks*, 9(5):545–556.

[Zhou e Haas 1999] Zhou, L. e Haas, Z. J. (1999). Securing ad hoc networks. *IEEE Network*, 13(6):24–30.

[Zhu et al. 1999] Zhu, W., Xiong, Z., e Zhang, Y. Q. (1999). Multiresolution watermarking for images and video. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(4):545–550.