Ataques e Mecanismos de Segurança em Redes Ad Hoc

Introdução

As redes ad hoc móveis (*Mobile Ad hoc NETworks - MANETs*) são constituídas por dispositivos móveis que utilizam comunicação sem fio. A principal característica dessas redes é a ausência de infra-estrutura, como pontos de acesso ou estações-base, existentes em outras redes locais sem fio ou ainda nas redes de telefonia celular. A comunicação entre nós que estão fora do alcance de transmissão do rádio é feita em múltiplos saltos através da colaboração de nós intermediários. Além disso, a topologia da rede pode mudar dinamicamente devido à mobilidade dos nós.

As redes ad hoc sem fio possuem como grande vantagem o baixo custo de instalação e facilidade de configuração. Por outro lado, o meio de comunicação sem fio, a ausência de infra-estrutura e o roteamento colaborativo em múltiplos saltos as tornam alvos potenciais de diversos tipos de ataques. Assim, a segurança é um ponto crucial das redes ad hoc.

A utilização do ar como meio de transmissão torna a rede susceptível a diversos ataques vão desde uma simples escuta clandestina (espionagem) passiva das mensagens até interferências ativas com a criação, modificação e destruição das mensagens. As redes cabeadas são consideradas mais seguras, pois um atacante tem maior dificuldade para obter um acesso ao meio físico e também para transpor as barreiras formadas pelos *firewalls*. Os ataques às redes sem fio, podem vir de várias direções e alvejar qualquer nó da rede, basta que o nó atacado esteja no alcance da transmissão do nó atacado. Dessa maneira, é possível que um nó malicioso tenha acesso a informações sigilosas, possa alterar mensagens em trânsito ou ainda tentar se passar por outros nós da rede. Portanto, o preço que se paga pelas facilidades oferecidas pela comunicação sem fio é a ausência de uma barreira de defesa clara. Assim, cada nó da rede deve estar preparado para lidar direta ou indiretamente com ações maliciosas.

Outro aspecto importante a ser considerado nas redes ad hoc é a ausência de centralização e de infra-estrutura. Portanto, não existem dispositivos dedicados a tarefas específicas da rede como, por exemplo, prover algumas funcionalidades básicas. Apesar de a descentralização ter como vantagem a robustez, devido a inexistência de pontos únicos de falha, a ausência de infra-estrutura dificulta a aplicação das técnicas convencionais de autorização de acesso e de distribuição de chaves. Isto dificulta a tarefa de distinguir os nós confiáveis dos nós não-confiáveis, pois nenhuma associação segura prévia pode ser assumida.

Devido a ausência de infra-estrutura, as redes ad hoc exigem a colaboração distribuída dos nós da rede para o encaminhamento das mensagens. Nas redes ad hoc, todos os nós participam do protocolo de roteamento, pois também desempenham a função de roteador. Além disso, estes nós roteadores estão sob o controle dos usuários da rede, e não de administradores. Isso possibilita a criação de novos ataques que visam as vulnerabilidades dos algoritmos cooperativos. Ou seja, as principais particularidades das redes ad hoc estão na camada de rede. Desta forma, os protocolos de roteamento das redes ad hoc devem ser robustos a novos tipos de ataques.

As redes ad hoc móveis introduzem outros obstáculos importantes à implementação de mecanismos de segurança devido às constantes alterações na topologia da rede.

Esta dinamicidade implica novos nós que se tornam vizinhos e antigos nós que deixam de ser vizinhos e pode até causar o particionamento da rede. Assim, os mecanismos de segurança devem se adaptar dinamicamente às mudanças na topologia da rede e ao movimento dos nós entrando e saindo da rede. Além disso, as redes ad hoc móveis são em geral compostas por dispositivos portáteis, portanto com restrições de energia, processamento e memória. Com isso, as MANETs estão sujeitas a diferentes ataques de negação de serviço que visam esgotar os recursos dos nós a fim de prejudicar o funcionamento da rede.

Desta forma, as redes ad hoc móveis possuem vulnerabilidades específicas ligadas principalmente ao meio de comunicação sem fio, à ausência de infra-estrutura e o roteamento colaborativo. A maior parte dos novos ataques concentra-se na camada redes e, conseqüentemente, também a maioria dos mecanismos de defesa específicos das redes ad hoc.

Este capítulo está organizado da seguinte forma. A Seção 2.2 revisa as ferramentas básicas utilizadas na implementação de segurança em redes de computadores. A Seção 2.3 descreve as principais formas de ataques às redes ad hoc móveis. A Seção 2.4 apresenta os principais mecanismos de segurança que podem ser utilizados para combater estes ataques, enquanto a Seção 2.5 apresenta os protocolos especialmente projetados para prover segurança em redes ad hoc. Finalmente, a Seção 2.7 analisa as tendências futuras na área de segurança em redes ad hoc.

Fundamentos de Segurança em Redes

Para o projeto de protocolos seguros, é necessário definir os objetivos que os mecanismos de segurança a serem implementados na rede devem buscar. Os requisitos de segurança clássicos que devem ser observados são a autenticação, a confidencialidade, a integridade, o não-repúdio e a disponibilidade. A autenticação garante que uma dada entidade é realmente quem ela diz ser, enquanto que o não-repúdio impede que o emissor de uma mensagem negue a sua autoria. A confidencialidade garante o sigilo das informações trocadas por dois nós e a integridade permite afirmar que as informações recebidas por um nó não foram alteradas durante o trânsito ao longo da rede. A disponibilidade trata de garantir que os recursos da rede estarão disponíveis quando forem necessários.

A criptografia é uma ferramenta fundamental para prover segurança, pois por meio dela, é possível atender a todos os requisitos clássicos. A maioria dos ataques a redes poderia ser solucionada pela utilização de um mecanismo criptográfico seguro.

Tradicionalmente, a criptografia é separada em dois ramos: simétrica e assimétrica. A criptografia simétrica é caracterizada pela existência de um segredo, chamado de chave secreta, compartilhado entre os nós que desejam se comunicar. Esta chave é utilizada em operações que alteram os dados a transportar, enviando um texto criptografado ao invés de um texto em aberto. As principais operações realizadas pelos algoritmos simétricos são o ou-exclusivo, a troca de colunas, a troca de linhas, a permutação, a rotação e a expansão, que são operações de baixo custo computacional. Apesar de serem simples, as combinações dessas operações devem ser capazes de tornar difícil a descoberta da mensagem para quem não possui a chave secreta. Por essa razão, a eficiência desses algoritmos é medida pelo seu custo computacional e pela capacidade de modificar a

saída dada uma pequena mudança na entrada. Os algoritmos simétricos mais conhecidos são o DES [National Bureau of Standards, 1977] e o AES [Daemen e Rijmen, 2002]. Na criptografia assimétrica existem duas chaves, a chave pública e privada. A chave pública deve ser distribuída aos membros da rede, enquanto que a privada deve ser mantida em segredo pelo nó. Esse tipo de criptografia possui maior custo computacional que a simétrica, por fazer uso de operações como o logaritmo discreto, curva elíptica e fatoração de inteiros, aliadas as considerações de segurança da Teoria dos Números. O objetivo principal é que, a partir de uma das chaves, não seja possível encontrar a outra, o que é obtido quando se usa para o cálculo funções que são simples de calcular, mas quase impossíveis de se reverter computacionalmente. Outras funcionalidades, como a distribuição de chaves de forma segura e a assinatura de mensagens são possíveis com o uso de criptografia assimétrica. Os algoritmos mais conhecidos são o RSA [Kaliski e Staddon, 1998], o Diffie-Hellman [Rescorla, 1999], que utilizam números primos entre si muito grandes para gerar as suas chaves, e mais recentemente a Criptografia de Curva Elíptica (*Elliptic Curve Cryptography* (ECC)) [IEEE, 2000], considerada a mais segura.

Em redes cabeadas, é comum utilizar as características dos dois tipos de criptografia para garantir uma comunicação segura, o que é conhecido como criptografia híbrida. Primeiramente, é trocado um segredo entre os nós por intermédio das chaves públicas. Este segredo servirá como chave secreta para criptografar a comunicação posterior usando criptografia simétrica, de menor custo computacional. A Figura 2.1 mostra como funciona a criptografia híbrida. Primeiramente, ambos os nós trocam suas chaves públicas. Em seguida, o nó A gera uma chave secreta, a criptografa com chave pública de B e a envia. O nó B, então, decriptografa a mensagem com sua chave privada e gera uma mensagem contendo a chave secreta, criptografada com a chave pública de A, para confirmar que conseguiu obter o segredo. É importante notar que um esquema como esse não é suficiente para garantir a autenticação e confiabilidade das mensagens, pois um nó malicioso poderia realizar o Ataque do Homem do Meio (*Man in the Middle Attack*), forjando a comunicação para os dois nós.

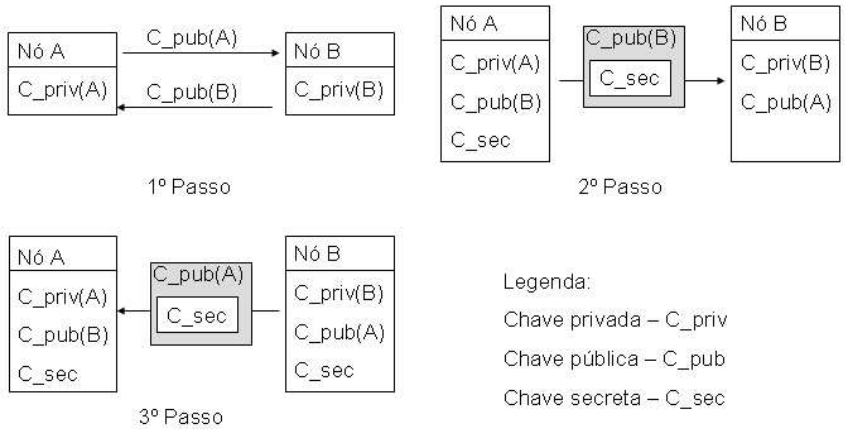

Figura 2.1. Criptografia híbrida.

O uso de uma Infra-Estrutura de Chave Pública (*Public-Key Infrastructure*-PKI) e assinatura digital permite solucionar o problema de interceptação, além de garantir a

autenticação e o não-repúdio na rede. No PKI, existe uma terceira entidade, chamada autoridade certificadora (AC), capaz de garantir a quem pertence realmente uma chave pública, através da emissão, validação e revogação de certificados. Um certificado deve conter a identificação e a chave pública do nó, criptografados com a chave privada da AC. A assinatura digital será feita criptografando a mensagem com a chave privada do emissor e enviando junto com ela um certificado emitido pela AC para aquele emissor. Assim, ao receber a mensagem, o destino decriptografa o certificado com a chave pública da AC, que deve ser conhecida por todos os nós da rede, obtendo a chave pública do emissor, que será a única chave capaz de decriptografar a mensagem. Assim, é possível garantir a autenticação e o não-repúdio, pois a chave privada deve ser mantida sempre em segredo pelos nós. É importante ressaltar, que apesar dessas características, tal mecanismo não garante a privacidade dos dados, o que poderia ser feito criptografando todo o pacote com a chave pública do receptor ou ainda com uma chave secreta obtida por criptografia híbrida.

Uma outra entidade importante para o funcionamento do PKI é a Autoridade Registradora, que realiza o cadastro dos usuários que desejam ser certificados. Um usuário sem registro não pode assinar mensagens, pois a AC não será capaz de emitir certificados para ele.

Uma outra ferramenta importante que pode ser utilizada para garantir integridade dos dados são as funções *hash*. Uma função *hash* é definida como uma função H que mapeia uma seqüência de bits de tamanho arbitrário em uma de tamanho fixo. O conceito de função *hash* unidirecional foi introduzido em [Diffie e Hellman, 1976]. De maneira informal, uma função *hash* unidirecional deve ser simples de calcular, porém computacionalmente impossível de ser invertida. Para o uso em criptografia simétrica, ainda se exige que a função *hash* seja resistente a colisões, ou seja, é computacionalmente impossível encontrar duas seqüências distintas x e y tais que $H(x) = H(y)$. Os famosos algoritmos MD5 [Rivest, 1992] e SHA-1 [National Institute of Standards, 2000] foram projetados para possuírem essas propriedades. Além dessas propriedades básicas, as funções *hash* criptográficas geralmente possuem propriedades aleatórias, como a uniformidade dos valores de saída ao longo do conjunto imagem, a independência entre a entrada e a saída, a impossibilidade de inferência da saída ainda que partes da seqüência de entrada sejam conhecidas, etc. Devido a essas propriedades, essas funções também são conhecidas como funções de espalhamento.

A disponibilidade não é tratada diretamente com o uso de criptografia. De fato, para garantir esse conceito, é necessário evitar ataques de negação de serviço. A autenticação dos usuários permite que seja dado acesso à rede ou aos serviços disponíveis apenas aos nós autorizados, o que já ajuda a reduzir o acesso dos atacantes à rede.

Criptografia Simétrica x Assimétrica: Vantagens e Desvantagens

A criptografia assimétrica trouxe inúmeros avanços, resolvendo questões como a autenticação e o não-repúdio através do PKI, que eram problemas em aberto para algoritmos simétricos. Mesmo os mecanismos que utilizam autenticação com chaves secretas e funções *hash* não são capazes de garantir quem foi o gerador da mensagem, pois o segredo é conhecido por todos os nós que fazem parte da mesma comunicação.

Outro ponto é que para se manter uma comunicação segura utilizando apenas criptografia simétrica, é necessário que cada par de nós possua uma chave para comunicação. Dessa forma, em uma rede com n nós, seriam necessários $(n * (n - 1))/2$ pares de chaves em cada nó. Utilizando criptografia assimétrica, o número de valores armazenados cai para n. Cabe ressaltar que o tamanho da chave utilizada com a criptografia assimétrica tradicional é muito superior ao tamanho da chave da criptografia simétrica, o que pode ser um problema para nós com restrições de memória. Neste caso, é aconselhado o uso de criptografia elíptica, pois suas chaves são menores que a dos demais algoritmos assimétricos devido à alta complexidade da inversão da função elíptica.

Apesar de todas as vantagens do PKI, o seu uso em redes ad hoc não é simples. A primeira razão são as fortes restrições de processamento e memória dos nós, o que torna muito complicado o cálculo para criptografar e decriptografar mensagens. Mesmo em redes com mais recursos, como as cabeadas, não se utiliza criptografia assimétrica para criptografar todas as informações, devido ao seu alto custo. Em redes de sensores sem fio, essa restrição é ainda maior, pois seus recursos são muito reduzidos, devendo ser poupados ao máximo. Medidas com sensores MICA mostraram que estes se tornam entre 100 e 1000 vezes mais lentos utilizando criptografia assimétrica [Kulkarnia et al., 2006]. Medidas utilizando *hardwares* específicos para os algoritmos RSA e DES mostraram um desempenho 1500 vezes mais lento do RSA [Nichols e Lekkas, 2002]. Os recursos das redes ad hoc costumam ser menos restritos que os das redes de sensores, mas, ainda assim, o uso de criptografia assimétrica deve ser evitado ao máximo, tentando buscar formas de equilibrar suas vantagens com a facilidade da criptografia simétrica.

Outro ponto importante a ser observado é que a utilização de PKI exige uma terceira entidade para certificar a comunicação, o que é, geralmente, feito por um servidor. No entanto, redes ad hoc não possuem pontos centrais. Assim, um outro problema a ser solucionado é como distribuir as tarefas da AC pelos nós da rede e como fazer o registro das chaves públicas em uma rede que deve ser auto-configurável.

Principais Formas de Ataques

Os ataques a redes ad hoc móveis podem ser divididos em passivos ou ativos [Murthy e Mano, 2004]. Os ataques passivos não afetam a operação da rede, sendo caracterizados pela espionagem dos dados sem alterá-los. Por outro lado, os ataques ativos são aqueles em que o atacante cria, altera, descarta ou inviabiliza o uso dados em trânsito. Os ataques ativos são os mais numerosos, podendo atuar em diferentes camadas do modelo OSI.

Os atacantes podem ser classificados como internos ou externos. Atacantes internos são aqueles que conseguem de alguma forma se passar por membros da rede, enquanto que os externos são aqueles que influenciam, mas não participam da rede. De fato, a eficiência e as possibilidades de ataques variam de acordo com o acesso que o atacante tem à rede. Se de alguma forma ele conseguir obter chaves ou for incluído na lista de vizinhos válidos, passando a ser um atacante interno, poderá causar mais problemas.

Ataques Passivos

Nos ataques passivos, o atacante não interfere no funcionamento da rede, mas pode escutá-la e analisar o seu tráfego. O atacante tem acesso à informação, porém não a altera ou destrói. Os ataques passivos são de difícil detecção por não influírem no comportamento da rede.

Espionagem

A espionagem (*eavesdropping*[2]) caracteriza-se pela escuta do tráfego sem modificação dos dados. O atacante aproveita-se do meio inseguro apenas para roubar informações.

Quando o atacante utiliza o tráfego observado para aprender a localização dos recursos críticos da rede, o ataque é chamado de Revelação de Informações Críticas (*Homing*/*Information Disclosure*) [Wood e Stankovic, 2002]. Uma vez que esses pontos são encontrados, essas informações são passadas para outros nós maliciosos que poderão realizar ataques ativos. Protocolos de roteamento que utilizam encaminhamento geográfico são ainda mais expostos a esse ataque, pois a posição exata dos nós críticos é passada para os atacantes ativos, facilitando a localização e ataque ao nó.

A proteção contra espionagem costuma ficar sob responsabilidade das camadas superiores, que, em geral, cuidam do sigilo das informações. No entanto, como a espionagem de informações de roteamento pode levar à exposição da topologia da rede para o atacante, o sigilo através da criptografia passa a ser uma necessidade do roteamento [Karlof e Wagner, 2003].

Ataques Ativos

Os ataques ativos, em sua maior parte, têm como alvo a vulnerabilidade de alguma camada específica do modelo OSI. Esta seção considera cada uma das camadas, descrevendo os seus principais ataques.

Camada Física

Na camada física estão os ataques de tratamento mais difícil, pois eles exercem uma utilização indevida do meio sem fio por dispositivos que não participam da rede. Os ataques são mais fáceis que nas redes cabeadas, pois não há a necessidade de conexão "física" ao meio de comunicação. Cabe ressaltar que os ataques da camada física são característicos do meio físico utilizado, e não específicos das redes ad hoc.

Interferência Contínua

Esse é um ataque muito conhecido, que consiste em sujar continuamente com interferências a freqüência de comunicação da rede sem fio. Um adversário pode impedir totalmente o funcionamento de uma rede com *N* nós utilizando *k* nós maliciosos distribuídos randomicamente, onde $k \ll N$ [Wood e Stankovic, 2002]. Para redes com uma única

[2]A origem do termo vem da expressão "*hide out in the eavesdrop of a house*" [Wikipedia - The Free Encyclopedia, 2006], que tem como similar em português o "ouvir atrás da porta".

freqüência de comunicação, esse é um ataque simples e muito efetivo.

A detecção da interferência contínua também é simples, pois basta que o nó atacado ou algum nó próximo à área de interferência observe que um nível constante de energia, e não a falta de uma resposta, impede a comunicação. Se o nó está no alcance da interferência, mas não totalmente imerso nela, ele pode comunicar ao resto da rede que evite rotas por aquela área. Embora o resultado do ponto de vista da aplicação seja o mesmo devido à perda de pacotes ou devido à interferência, esta é mais grave por impedir que o nó envie ou receba pacotes, ou mesmo que se faça uma notificação para algum nó de monitoramento.

A interferência é um ataque de difícil combate. Entre algumas das propostas estão o espalhamento de espectro, descrito mais adiante, e o descobrimento de novas rotas isolando a área de interferência. Para os nós que estão dentro dessa área só resta dormir e checar periodicamente até que a interferência acabe.

Interferência Esporádica ou Exaustão por Interferência

Este tipo de interferência pode ser ainda pior que a contínua. Ela consiste em gerar interferências por curtos períodos, o que já é suficiente para impedir a comunicação. A grande eficiência dos ataques com interferência esporádica se deve a eles poderem causar grandes prejuízos no consumo de bateria do nó atacado, que deve fazer retransmissões, e ao mesmo tempo terem um custo mínimo para o nó atacante, que realiza apenas pequenas transmissões de tempos em tempos. Além disso, como o ataque é esporádico, é de mais difícil detecção. A melhor forma de minimizar os efeitos desse ataque é através do método de espalhamento de espectro.

Método de Espalhamento de Espectro

Algumas tecnologias de transmissão sem fio impõem um nível de dificuldade a mais a ataques da camada física [Stallings, 2004]. Um exemplo são as técnicas de modulação baseadas em um espectro de freqüências espalhado (*Spread Spectrum*). Técnicas de espalhamento de espectro como DSSS (*Direct Sequence Spread Spectrum*) e FHSS (*Frequency Hopping Spread Spectrum*) foram projetadas com o objetivo de maior resistência a interferências de fontes de faixa de freqüências estreita. A idéia básica é utilizar uma largura de banda maior do que a que é realmente necessária para transmitir dados a uma velocidade específica. No FHSS, o espalhamento de espectro é obtido saltando-se continuamente de uma freqüência de portadora para outra, minimizando, desta forma, interferências. Se um atacante não conhecer qual a seqüência em que as portadoras são utilizadas, ele não é capaz de obter acesso ou sujar a informação sendo transmitida. Já o DSSS literalmente aumenta a taxa de dados de um sinal, mapeando cada bit de informação em uma cadeia de bits a transmitir, chamada de seqüência de *chips*. O efeito é espalhar um bit de informação no tempo, o que aumenta a robustez a interferências. Para cada valor binário 0 ou 1 a transmitir, uma seqüência de chips é transmitida. Os códigos utilizados como seqüência de chips para transmitir os bits de dados fornecem a segurança inerente do DSSS.

É importante notar que as técnicas de modulação por espalhamento de espectro possuem um nível de segurança inerente, mas não fornecem qualquer proteção criptográfica. A segurança vem apenas do fato de manter os códigos (seqüência de portadoras ou

de chips) secretos. Uma vez que estes códigos não são protegidos, e, normalmente, são bem conhecidos ou fáceis de descobrir, o nível de segurança fornecido é mínimo. Assim, estes códigos provêem pouca ou nenhuma proteção contra ataques de negação de serviço na camada física através das interferências, embora ainda seja o melhor método para preveni-las.

Camada Enlace

A camada enlace é a responsável pela transmissão confiável de dados ponto a ponto. Desta forma, nesta camada, os ataques visam à retransmissão de quadros, a prioridade de mensagens e os códigos corretores de erro, características especificas do IEEE 802.11, a tecnologia mais usada nas redes ad hoc. A solução para esses ataques consiste de uma implementação mais robusta do protocolo de enlace, prevendo o comportamento malicioso.

Exaustão de Bateria por Colisão

Neste ataque, o nó malicioso tem como objetivo consumir a bateria do nó atacado gerando retransmissões continuamente, através de uma implementação maliciosa da camada de enlace. As retransmissões são geradas por recursos como, ao ouvir o início de uma transmissão, gerar uma colisão tardia no fim do quadro. No caso desta colisão intencional ocorrer com um pacote de ACK, isso poderia acarretar em um aumento exponencial do *back-off* em alguns protocolos MAC [Wood e Stankovic, 2002]. O uso repetido desse método culmina na exaustão da bateria do nó atacado, pois a transmissão é uma operação muito custosa, e que só deve ser feita quando estritamente necessário.

Uma variação deste ataque é conhecida como o ataque da interrogação. Neste, o nó atacante, que é um nó suicida, explora características da interação de protocolos da subcamada MAC que utilizam o *Request To Send* (RTS), *Clear To Send* (CTS) e mensagens de dado e ACK. O nó malicioso faz pedidos de alocação do canal com RTS repetidos, forçando inúmeras respostas CTS da vítima, levando ambos à morte.

A exaustão de bateria é de difícil tratamento, pois necessariamente a camada de enlace conta com certa confiança entre os nós participantes. Um nó malicioso poderia negar acesso ao canal repetidamente, impedindo o funcionamento da rede sem ter um grande gasto de energia. Soluções para essa variação do ataque são obtidas na reformulação dos protocolos, tornando-os mais robustos a comportamentos inadequados [Wood e Stankovic, 2002].

Alteração de ACK

A maioria dos algoritmos de roteamento confia de alguma forma em ACKs da camada de enlace. Uma vez que o meio é o ar, um atacante pode forjar esses ACKs com a finalidade de enganar o receptor a respeito de dados como a qualidade do canal, ou ainda dizer que um nó que já desativado ainda está ativo. Isso implicaria na escolha de rotas por enlaces inapropriados ou passando por nós que não participam mais da rede [Karlof e Wagner, 2003].

Camada Transporte

Os ataques à camada de transporte [Wood e Stankovic, 2002], visam vulnerabilidades do TCP, na fase de sincronização e na retransmissão de pacotes.

Inundação de *Sync*

Para realizar a comunicação utilizando o TCP, é necessário um período de tempo para o estabelecimento da conexão. Cada processo de conexão ocupa um espaço de memória no nó até que seja concluído. Este ataque visa explorar essa característica, gerando vários pedidos de conexão para a vítima. Cada um desses pedidos, que nunca é completado, provoca a alocação de mais recursos, até o momento que acontece um estouro de memória. A limitação do número de conexões pode impedir que o ataque consiga a exaustão de recursos, mas não pode impedir que conexões reais com nós legítimos sejam perdidas devido a inúmeros pedidos de conexões falsos na fila. Outra possível solução é o uso de desafios para diminuir a velocidade que o nó malicioso gera os pedidos de conexão.

Dessincronização

Neste ataque um terceiro nó influi em uma conexão entre dois nós legítimos. O atacante envia mensagens falsas pedindo retransmissões, a partir da observação do número de seqüência que está sendo utilizado na comunicação. Para o funcionamento efetivo do ataque, é necessário que o atacante mantenha um controle preciso do momento de envio das mensagens, para evitar que os nós legítimos troquem informações úteis. O tratamento para esse ataque exige autenticação e criptografia das mensagens.

Seqüestro de Sessão (*Session Hijacking*)

O seqüestro de sessão, descrito em [Murthy e Mano, 2004], é um ataque onde o adversário toma o controle de uma sessão entre dois nós. Uma vez que a maioria dos processos de autenticação só é feita no início da sessão, após essa fase, o atacante pode se passar por uma das extremidades, se comunicando com o outro nó como se fosse o nó legítimo. Para evitá-lo deve-se utilizar criptografia ou assinatura digital em todas as mensagens trocadas.

Camada Rede

Na camada rede acontece a maior parte dos ataques, devido tanto às características críticas da rede, quanto às vulnerabilidades dos protocolos de roteamento. Muitos dos ataques a essa camada possuem soluções eficientes, com métodos preventivos capazes de reduzir bastante a interferência do atacante na rede.

Ataque Bizantino

Neste ataque, geralmente ligado a problemas de tolerância a falhas, um ou mais nós maliciosos trabalham em conluio para gerar problemas como *loops* de roteamento, pacotes de roteamento falsos, escolha de caminhos não-ótimos, entre outros, utilizando mensagens de controle dos protocolos que estão sendo utilizados. Murthy cita esse problema com o nome de mensagens de roteamento alteradas, restringindo-o apenas aos

problemas de roteamento. Além disso, os nós também podem executar um encaminhamento seletivo [Murthy e Mano, 2004]. Esse tipo de ataque é de difícil detecção, pois para os nós comuns, o funcionamento estará correto, embora, de fato, esteja apresentando anomalias.

O nome ataque bizantino tem uma origem curiosa. A idéia é baseada no problema dos generais [Lamport et al., 1982] bizantinos, distribuídos em campo com suas tropas para organizar o ataque à cidade inimiga. A comunicação entre eles é feita apenas por mensagens e isso deve ser suficiente para organizar o ataque. No entanto, um ou mais generais podem ser traidores tentando confundir os demais, o que gera a necessidade de um algoritmo capaz de garantir que os generais leais conseguirão chegar a um acordo.

Tem-se como objetivos que todos os generais leais devem decidir pelo mesmo plano de ação e um pequeno número de generais maliciosos não deve levar os generais leais a adotar um plano ruim. Para satisfazer estes dois objetivos, é necessário que todos os generais leais recebam a mesma informação, e se um general é leal, então sua informação deve ser utilizada por todos os generais leais. Como ambas as condições levam ao mesmo ponto sobre como um general envia sua ordem, é possível simplificar o problema a um general e dois tenentes que devem receber a sua ordem. No caso de apenas um general e um tenente, a solução é trivial, pois a comunicação é direta e não existem mais versões sobre o que foi dito. Assim, o problema acontece a partir de três generais. Por simplificação, a mensagem só pode ser de atacar ou bater em retirada. O problema pode ser caracterizado como na Figura 2.2, onde há um general e dois tenentes. No primeiro caso, temos um tenente traidor, e o tenente leal receberá duas mensagens, uma do general mandando atacar e uma do traidor dizendo que as ordens do general são de bater em retirada. No segundo caso, mostrado na Figura 2.3, o general é traidor, e o mesmo grupo de informações chega ao tenente leal, de forma que, com três entidades, sendo uma traidora, não há solução. Generalizando o problema, se cada entidade representasse m generais/tenentes, ainda assim, não seria possível resolver o problema, pois seria necessário mais algum testemunho para concluir qual é a informação que deve ser usada. Isto leva a regra de que são necessários pelo menos $3m + 1$ generais, sendo m o número de generais mentirosos, para que os generais leais cheguem a uma solução única e verdadeira. A solução matemática para o problema é descrita em [Pease et al., 1980].

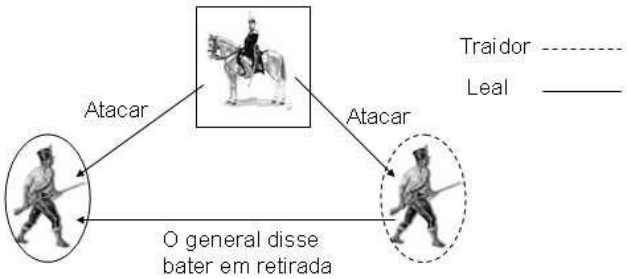

Figura 2.2. Ataque Bizantino onde um tenente é traidor.

As soluções para o ataque bizantino podem ser a assinatura digital, o uso de múltiplos caminhos e ainda a autorização, mecanismos que serão descritos na Seção 2.4.

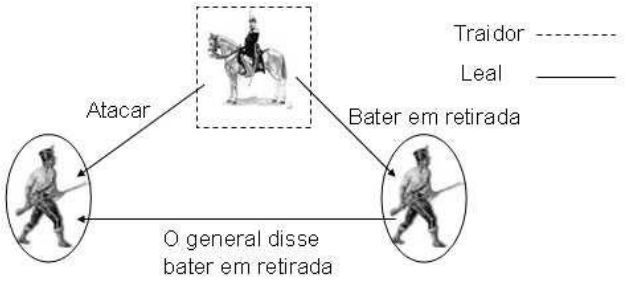

Figura 2.3. Ataque Bizantino onde o general é traidor.

Estouro (*Overflow*) da Tabela de Roteamento

Este ataque se baseia no fato de os protocolos de roteamento ad hoc pró-ativos armazenarem todas as rotas anunciadas pelos seus vizinhos. Nestes protocolos, o nó armazena em sua tabela de roteamento todas as mensagens de rota que recebe periodicamente. A estratégia deste ataque é anunciar diversas rotas para nós inexistentes, de modo a aumentar progressivamente o tamanho da tabela de roteamento, até que ela estoure e o nó não possa mais armazenar as rotas reais. Os protocolos reativos que armazenam diversas rotas para um mesmo destino também estão expostos a esse tipo de ataque, pois o nó malicioso poderia enviar rotas passando pelos nós inexistentes.

Esse ataque é grave no caso de redes ad hoc que possuem nós com escassos recursos, onde tanto o gasto de energia com a recepção de um número excessivo de mensagens, quanto o estouro de *buffer* são cruciais. Para preveni-lo, deve-se limitar o número máximo de rotas nas tabelas de roteamento, além de só aceitar entradas de nós autenticados.

Replicação de Pacotes

Este ataque possui dois objetivos principais: ocupar o meio de transmissão e levar os nós à exaustão. Esta é a versão de nível roteamento do ataque de exaustão de bateria da camada enlace. Para conseguir seus objetivos, o atacante envia réplicas de pacotes de roteamento antigos.

As soluções propostas para esse tipo de ataque são limitadas, pois ainda que se deduza que o mesmo nó envia mensagens de roteamento antigas, através da observação do número de seqüência, o máximo que poderia ser feito é retirar aquele nó das rotas, mas nada poderia impedi-lo de continuar as suas réplicas, assim como acontece no ataque da interferência na camada física.

Envenenamento de *Cache*

Muito semelhante ao ataque do estouro da tabela de roteamento, este ataque visa envenenar o *cache* de roteamento fazendo anúncios falsos de rotas para nós reais. Esse ataque se aproveita em especial de protocolos sob demanda, como o *Ad Hoc On demand Distance Vector* (AODV) [Perkins et al., 2003], que mantêm rotas para nós que foram aprendidas em um passado recente. Estes protocolos estão mais susceptíveis que os protocolos pró-ativos pelo fato de anunciarem para onde desejam mandar o pacote sempre que não tem rota, permitindo ao nó invasor anunciar a rota falsa antes do nó confiável. No caso dos protocolos pró-ativos, esse ataque também é possível, mas seria necessá-

rio mandar anúncios de rota falsos em todas as rodadas de atualização, para todos os possíveis destinos. Para evitar esse tipo de comportamento dos nós maliciosos, deve-se utilizar sistemas de confiabilidade baseados em monitoramento e punição. Outra forma de identificação deste ataque é a utilização de pacotes de investigação 2.4.

Ataque da Pressa (*Rushing Attack*)

Este ataque permite a formação de um buraco negro, se aplicando a protocolos de roteamento sob demanda que guardam apenas uma rota para cada destino em sua tabela. Ao receber um *Route Request*, o atacante o envia de forma mais rápida aos demais nós da rede, de forma que todas as respostas passem por ele. Assim, como ele será o primeiro a responder, as demais respostas provenientes dos outros vizinhos serão descartadas. Desta forma, as rotas sempre passariam pelo nó malicioso, tornando a rede vulnerável.

A detecção de tal ataque é difícil, dado que, para o protocolo de roteamento, tudo está transcorrendo de forma normal. A solução seria tentar identificar os métodos para conseguir enviar a mensagem de forma mais rápida, como por exemplo, um abuso do protocolo de enlace. De fato, existem vários métodos para acelerar o envio da mensagem, e que não exigem muitos recursos do nó malicioso [Hu et al., 2003b]. Em geral, os protocolos MAC (*Medium Access Control*) impõem atrasos entre o momento no qual o pacote é recebido e a transmissão. Um exemplo são os protocolos MAC que utilizam divisão de tempo para acesso ao meio, onde o nó precisa esperar até a sua vez de transmitir, ou ainda os protocolos de acesso ao meio que utilizam *Carrier-Sense Multiple Access* (CSMA), onde é utilizado um *backoff* para evitar colisões. Outro espaço de tempo que também pode ser burlado por um nó malicioso é aquele que pode ser usado pelo roteamento entre a recepção de um RREQ e o encaminhamento do mesmo, para evitar colisões. Assim, um atacante que deseja enviar um pacote mais rápido que outros nós pode simplesmente ignorar um ou mais destes tempos de espera. Uma outra forma de executar esse ataque seria, por exemplo, provocar a criação de filas nas interfaces dos nós vizinhos, de forma a que o nó malicioso repasse o pacote enquanto seus vizinhos estão processando os pacotes das filas. Esse tipo de atitude do nó malicioso é mais fácil em sistemas que utilizam autenticação da mensagem, pois ele poderia gerar várias mensagens com defeito, levando os vizinhos a perder tempo verificando as mensagens. No caso de autenticação por chave pública esse problema é ainda maior, devido ao alto custo computacional para realizar a verificação. Outros métodos para transmitir os pacotes mais rápido que os vizinhos também são possíveis, como a utilização de uma potência de transmissão maior, ou, ainda, através da utilização de um túnel de minhoca (*Wormhole*).

A melhor solução para este ataque é o uso de múltiplas rotas disjuntas ou trançadas, que garantiriam que mesmo que o atacante atraísse o trafego para si em uma das rotas, as outras permaneceriam seguras.

Direcionamento Falso (*Misdirection*)

O direcionamento falso consiste na fabricação de mensagens visando gerar negação de serviço para um determinado nó. Assim, são enviadas mensagens de modo a direcionar tráfego para uma determinada região que se deseja atacar. Na versão da Internet desse ataque, conhecida como Ataque *Smurf*, o atacante forja pacotes *echo*, colocando como emissor o nó vítima, que irá receber inúmeros *echo-backs*.

Esse ataque pode ser realizado por mecanismos além do uso de *echos*. O caso do protocolo *Dynamic Source Routing* (DSR) [Johnson e Maltz, 1996] é um exemplo, onde o atacante pode responder às requisições de rotas com caminhos falsos que incluem o nó que se deseja atacar.

Inundação de *Hellos*

Este ataque inicialmente foi considerado para redes de sensores. De fato, ele também se aplica as redes ad hoc, desde que o atacante possua uma potência de transmissão maior que os demais nós da rede. A Inundação de *Hello* só se aplica a protocolos que utilizam a mensagem de *hello* para identificação dos vizinhos, embora não façam a verificação de bidirecionalidade do enlace.

Para realizar o ataque, o nó malicioso envia *hellos* com alta potência, informando que o nó possui enlaces muito bons com determinados destinos. Assim, ele atinge um grande número de nós, que por terem ouvido a mensagem, o colocam na sua lista de vizinhos e podem escolhê-lo para encaminhamento de dados. No entanto, apesar dos nós ouvirem o nó malicioso, o nó malicioso não é capaz de escutá-los, de forma que vários nós da rede irão apontar suas rotas de encaminhamento para um nó inalcançável, como ilustrado na Figura 2.4.

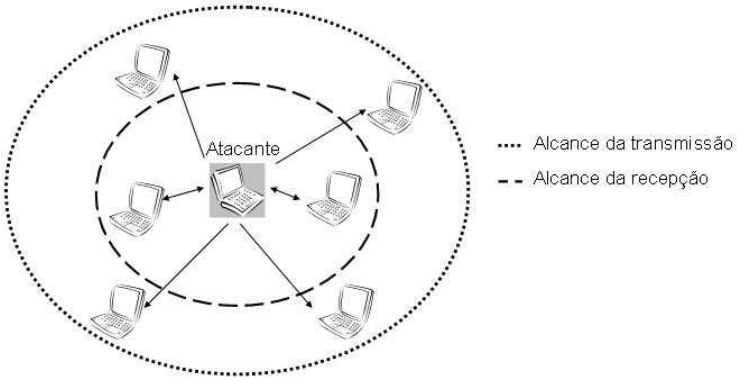

Figura 2.4. Inundação de *Hellos*.

A inundação de *hello* funciona melhor quando é realizada em conjunto com uma revelação de informações críticas, de forma a permitir que o nó malicioso descubra quais rotas ele deve tentar interceptar.

Esse ataque pode ser evitado pela verificação de bidirecionalidade do *link*. Cabe observar que apesar de ser uma solução simples, muitos protocolos de roteamento não a aplicam, assumindo que os enlaces são bidirecionais.

Ganância (*Greed*)

A Ganância é caracterizada quando um nó dá uma prioridade injusta às suas mensagens. O nome representa o fato de o nó, apesar de não prejudicar o funcionamento da rede de forma explicita, atrapalhar por tentar ter sempre a maior fatia de tempo para transmissão.

A detecção de um nó ganancioso é difícil, pois um nó que já esteja com falhas de bateria apresenta um comportamento semelhante. Assim, a melhor forma de evitar esse

ataque é a utilização de redundâncias, garantindo que, ainda que pacotes sejam perdidos por demora ao enviar, uma segunda rota poderá garantir a sua entrega.

Encaminhamento Seletivo ou Buraco Cinza

Uma das características principais das redes ad hoc é a confiança nos vizinhos para o encaminhamento de dados. No entanto, um vizinho malicioso pode encaminhar apenas alguns pacotes. Neste caso não se deseja prejudicar todos os nós, ou uma determinada área da rede. O nó malicioso pode escolher não passar alguma ou todas as mensagens pra um determinado nó alvo, ou pode optar por passar as mensagens de roteamento, mas impedir a transmissão de dados, impedindo o funcionamento da aplicação.

Um tipo especial de encaminhamento seletivo é chamado de egoísmo, no qual o nó não encaminha nenhuma mensagem dos vizinhos, passando apenas as suas próprias. O egoísmo nem sempre é um ataque, podendo ser uma escolha de um nó por um comportamento não cooperativo. Tal decisão pode ser tomada, por exemplo, em momentos nos quais o nó deseja se poupar.

Diferentemente de um buraco negro, esse ataque é de difícil detecção, assim como a Ganância, pois nós com pouca energia e perdas de pacotes normais podem gerar um quadro muito semelhante.

Buraco Negro

Este é caso extremo do Encaminhamento Seletivo, onde todos os pacotes são atraídos até o nó e são descartados. Este ataque, dependendo da posição do atacante, pode ter um efeito totalmente destrutivo na rede, impedindo todo o seu funcionamento. Por outro lado, ao contrário do Encaminhamento Seletivo, sua detecção é fácil, pois em muito pouco tempo todo um ramo da rede deixará de funcionar.

Uma segunda conseqüência do buraco negro, é que, como ele atrái muito tráfego em sua direção, ele acaba consumindo os recursos dos nós à sua volta, tanto em termos do meio, que fica excessivamente ocupado, assim como em termos de recursos dos nós. No caso de exaustão desses nós, o resultado poderia acabar particionando a rede.

Assim como o Encaminhamento Seletivo, uma premissa de funcionamento do ataque é que o nó malicioso se torne atrativo durante a escolha de rotas. Para tanto, vários métodos podem ser utilizados.

A solução mais simples para os ataques de descarte de pacotes é a utilização de múltiplas rotas. Outros métodos que ajudam a detectar e prevenir esse comportamento são a investigação e a autorização.

Túnel de Minhoca (*Wormholes*)

No ataque do túnel de minhoca, dois atacantes criam um túnel de comunicação por um enlace de baixa latência, através do qual irão trocar informações da rede, replicando-as do outro lado do túnel, de forma a tornar excepcionalmente atrativo o enlace formado pelos dois. Assim, os nós maliciosos podem convencer nós da rede que eles podem se comunicar com determinado destino por apenas um salto, ao invés de utilizar os vários saltos que existem realmente entre o nó e o destino.

O túnel é um canal seguro e de baixa latência entre os dois nós maliciosos, que per-

mite que os nós vizinhos sejam incapazes de perceber que o ataque está sendo realizado. Uma forma simples de obter esse resultado é utilizar uma conexão por fio entre os dois nós, fazendo uma transmissão mais rápida que o encaminhamento por múltiplos saltos [Hu et al., 2003b]. Outra possibilidade seria a utilização de um enlace direcional sem fio de longa distância, para conseguir maior velocidade que comunicações que normalmente utilizariam mais que um salto [Hu et al., 2003a]. Uma terceira forma seria utilizar um canal diferente do utilizado na comunicação com uma potência de transmissão superior, o que também permitiria a maior velocidade sem que os vizinhos notassem. Além disso, uma técnica que pode ser usada é o envio dos bits diretamente, sem aguardar a chegada do pacote completo para começar a transmissão.

É interessante notar que, no caso de o atacante construir o seu túnel de forma honesta e confiável, nenhum prejuízo direto é causado a rede. Pelo contrário, um serviço é prestado ao melhorar a eficiência da conexão da rede. No entanto, o ataque coloca os nós maliciosos em uma posição privilegiada, que os permite gerar, no momento que desejarem, diversos tipos de prejuízos à rede.

Ainda que a rede implemente autenticidade e confiabilidade, o ataque ainda pode ser realizado, pois, normalmente, o nó não precisa se autenticar para encaminhar um pacote. É importante observar que, para as camadas superiores, o ataque é invisível, e mesmo para a camada de roteamento, a princípio é complicado perceber a presença de um Túnel de Minhoca.

Ataques Multicamadas

Existem alguns ataques que não estão ligados a uma camada específica do modelo OSI, mas que podem afetar diversas camadas.

Exaustão de Bateria

Neste ataque, o nó malicioso tem como objetivo consumir a bateria do nó atacado, até que o nó fique inativo. De fato, esse ataque pode se aplicar a várias camadas. No caso de fazer essa atividade por meio de interferências, se trataria de um problema de camada física. Já se a interferência for gerada com o objetivo de gerar retransmissões, trata-se de um problema da camada enlace. O ataque pode também retransmitir mensagens reais da rede, dificultando sua detecção.

Tantas versões para o mesmo ataque se justificam pela importância da vida útil da bateria para dispositivos móveis, e, pela mesma razão, várias metodologias para poupar bateria já foram desenvolvidas. Por essa razão aplicativos de segurança que exigem modo promíscuo são muito criticadas, pois a ação de escutar a rede continuamente gasta muita energia do nó, sendo mais recomendado colocá-lo dormindo sempre que possível. Outros nomes dados a esse ataque são *Sleep Deprivation Attack* [Wood e Stankovic, 2002] e *Spam Attacks* [Sancak et al., 2004].

Negação de serviço

O conceito de negação de serviço é muito amplo. O ataque de negação de serviço pode ser definido como qualquer ação que reduza ou elimine a capacidade da rede de reali-

zar uma de suas funções esperadas [Wood e Stankovic, 2002]. Assim sendo, a negação de serviço não seria causada apenas por ataques, mas por qualquer evento que prejudicasse a rede, como falhas de *hardware*, defeitos de programas, exaustão de recursos intencional ou não, condições ambientais não favoráveis ou qualquer interação entre esses fatores. Dessa forma, todos os ataques ativos poderiam gerar uma negação de serviço na rede, o que dá a esse ataque a classificação de multicamadas.

Uma forma mais severa deste ataque é a negação de serviço distribuída. Nesta, vários adversários estão espalhados pela rede fazendo um conluio para impedir que usuários legítimos tenham acesso aos serviços. Este ataque tem um efeito muito mais rápido sobre a rede, podendo impedir totalmente o seu funcionamento sem grandes dificuldades.

Sybil[3]

O ataque Sybil se baseia no fato de que é praticamente impossível, em sistemas computacionais distribuídos, que nós que não se conhecem apresentem identidades distintas convincentes. Sem a existência de um ponto central para controlar a associação de uma identidade a uma entidade, é sempre possível para uma entidade desconhecida apresentar múltiplas identidades. Assim, o ataque sybil acontece quando um único *hardware* assume múltiplas identidades em uma rede [Newsome et al., 2004].

Este ataque tem grande importância por muitos sistemas utilizarem sistemas de réplicas de dados armazenados, para ter garantia contra violação de integridade, e sistemas de fragmentação de tarefas, para impedir a violação da privacidade. Em ambos os casos, a redundância, mecanismo explorado pelo ataque, é um ponto chave. Assim, devido ao nó malicioso assumir múltiplas personalidades, o sistema poderia escolher o mesmo nó para guardar todas as réplicas ou fragmentos, o que acabaria com toda a segurança adquirida com o mecanismo.

O Sybil pode ser utilizado para atacar não só armazenamentos distribuídos. Uma outra possibilidade que utiliza redundância é o roteamento com múltiplos percursos. Em geral, protocolos que utilizam essa técnica buscam escolher caminhos disjuntos ou trançados para diminuir a possibilidade de existir um atacante na rota. O ataque sybil pode ser feito de tal forma a colocar uma identidade falsa em cada rota, de forma que todos os caminhos continuarão passando pelo nó malicioso. Ainda no campo de roteamento, outro possível problema que não tem relação com redundância é o ataque ao roteamento geográfico. Neste caso, o nó malicioso anunciará sempre uma de suas identidades sybil como o nó mais próximo ao destino, fazendo com que todos os pacotes de roteamento passem por ela.

Outro ataque possível é a utilização dos nós sybils para falsificar resultados de votações na rede. Sempre que existir algum mecanismo cooperativo para tomada de decisões na rede, o nó malicioso pode gerar diversas identidades para votar sempre a seu favor. Outro ataque é a alocação injusta de recursos, que pode ocorrer em redes que fazem divisão temporal para acesso ao meio. Neste caso, o nó malicioso utiliza todas as suas

[3]A primeira descrição do ataque Sybil foi feita em [Douceur, 2002], para redes *peer-to-peer*. O nome do ataque foi inspirado em um caso famoso nos EUA, onde uma mulher sofria de múltiplas personalidades, num total de 16 diferentes personalidades. Sybil Isabel Dorsett foi o pseudônimo criado pela autora Flora Schreiber para proteger a identidade real da paciente, em seu livro "Sybil", Warner Books, 1973.

identidades falsas para obter um maior tempo de acesso. Por fim, uma outra utilização para os nós sybils acontece em redes que utilizam mecanismos de confiabilidade. Em tais redes, a índole do nó é dada pela observação de suas ações. Um nó só é considerado malicioso se cometer diversas ações consideradas ruins ou se cometer uma grande ação ruim. Assim, duas estratégias podem ser utilizadas. A primeira seria o espalhamento da culpa, na qual o nó sybil utiliza cada uma de suas identidades para fazer pequenas ações ruins, de forma que nenhuma delas possa ser considerada maliciosa. A outra estratégia seria utilizar uma identidade para realizar uma ou mais ações ruins até que ela fosse expulsa, classificada como maliciosa. Quando isso acontecesse o nó geraria uma nova identidade e a usaria para continuar atacando.

Existem diversas propostas de defesas para o Sybil, sendo a maioria delas baseadas em métodos de autenticação ou de validação de chaves distribuídos. Estes serão descritos na Seção 2.4.1.

Identidade Falsa (*Impersonating*) e Ataque da Replicação

Estes ataques se assemelham muito ao Sybil. Nestes, nós maliciosos assumirão uma ou mais identidades da rede, porém desta vez, todas as identidades são reais, e cada identidade estará ligada a um ou mais hardwares diferentes, caracterizando respectivamente a Identidade Falsa e o Ataque da Replicação. A Replicação serve para inserir vários nós maliciosos, sem ter a dificuldade de se roubar várias identidades. Desta forma, os nós maliciosos replicam alguma identidade roubada e utilizam as réplicas simultaneamente dentro da rede [Chan et al., 2003].

No caso dessas réplicas serem muito numerosas, os adversários podem dominar a rede através de um conluio para ter vantagens em casos votação, ou ainda tirar vantagem apenas por estarem participando da rede. Deve-se notar que esses ataques, que, em geral, acontecem após uma violação ou uma quebra de algoritmo criptográfico, fazem com que o atacante tenha o segredo da rede, podendo participar de todas as suas atividades como um nó legítimo. Assim, ele pode, por ter se tornado um atacante interno, executar a maioria dos ataques já descritos, com a facilidade do conluio com as outras réplicas.

Cabe ressaltar, que apesar da gravidade do efeito causado, a Replicação é de fácil detecção, devido a uma mesma identidade se anunciar em diversos pontos da rede. Algumas propostas eficientes já foram feitas, embora elas não estejam incluídas nos protocolos mais populares. No caso da identidade falsa, se o nó legítimo tiver sido destruído, a detecção é muito mais complicada, pois a identidade é única na rede.

Uma outra variação da Identidade Falsa é o Ataque do Homem no Meio (*Man-in-the-Middle*) [Murthy e Mano, 2004]. Neste, o nó malicioso intercepta uma comunicação, enganando os dois nós que deveriam se comunicar, como pode ser visto na Figura 2.5. Uma vez que ele se passa por x para y e por y para x, ele está assumindo duas identidades reais da rede. Esse tipo de ataque só pode ocorrer em redes que não possuem um terceiro ponto para autenticar a comunicação entre os dois primeiros.

Violação

Este é um dos ataques mais preocupantes para redes onde os nós ficam desprotegidos. Ele consiste da violação física dos nós com o fim de obter informações e segredos,

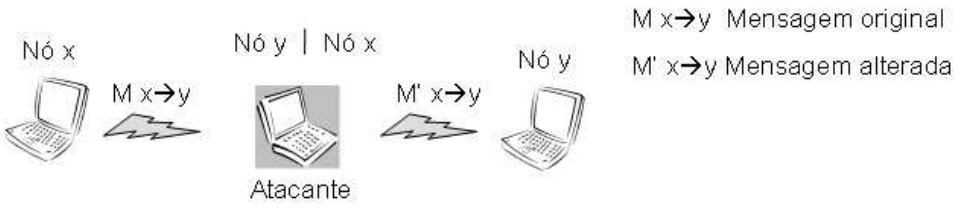

Figura 2.5. Ataque do Homem no Meio.

além de também comprometer o nó, com a inserção de códigos maliciosos ou ainda pela troca de partes do *hardware*. A maioria dos protocolos desenvolvidos para prover segurança falha em ambientes onde é possível ocorrer a violação.

De fato, não é simples garantir a segurança de todos os nós quando tratamos de redes de larga escala. Em especial pelo fato de que, em geral, existem muitas falhas de comunicação e períodos de sono, que tornam impraticável distinguir uma falha de um nó propositalmente desligado ou destruído. A proposta de defesa é a resistência à violação, descrita a seguir.

Resistência à Violação

O termo resistência à violação (*tamper proofing*), diz respeito à dificuldade imposta por um mecanismo de segurança à violação da informação, seja por software ou por hardware. O conceito pode ser estendido à capacidade da rede em resistir a ataques onde um usuário não autorizado se apossou de um dos nós da rede.

Desta forma, as técnicas de resistência à violação podem ser baseadas em hardware, em software, ou em ambos. Uma forma de aumentar a resistência à violação baseada em software é a utilização de associações de segurança temporárias. Por exemplo, toda vez que um nó for enviar uma requisição de rota em um protocolo de roteamento reativo, o sistema operacional pode exigir que o usuário entre com uma senha, ou forneça sua impressão digital para se autenticar. Se o equipamento for perdido ou roubado, o usuário não autorizado não conseguirá gerar pedidos de rota, e quando esta tentativa for feita, o sistema operacional do nó pode tomar alguma atitude adicional, como,por exemplo destruir qualquer chave armazenada no sistema, aumento sua resistência à violação [Yi et al., 2001]. Esse tipo de medida tem o problema de ser considerada como inconveniente pela maioria dos usuários.

A utilização de hardware resistente à violação possui, naturalmente, um problema de custo-benefício. Quanto mais seguro o hardware, mais caro [Anderson e Kuhn, 1996]. Além disso, como na segurança do software, não existe sistema de hardware inviolável. Considerem-se os *smart-cards*, utilizados, por exemplo, em alguns decodificadores de TV por assinatura para controlar os canais aos quais o usuário tem direito de acesso. *Smart-cards* típicos são constituídos de um micro-processador, memórias ROM, EEPROM e RAM, além de portas de entrada e saída. Normalmente, o apagamento de um bit da EEPROM exige uma voltagem alta. Ataques podem se basear no bloqueio desta voltagem mais alta, para evitar o apagamento de alguma informação. Por exemplo, em algumas empresas de TV por assinatura, o *smart-card* é gravado por padrão com todos os canais

de TV habilitados. No *set-top Box* do usuário, um sinal é emitido pela operadora de forma a "apagar"os canais não contratados do smart-card. Um ataque simples consiste em bloquear a geração de voltagem mais alta para esta operação, no set-top box, garantindo o acesso a todos os canais. Uma solução para aumentar a segurança contra este tipo de ataque é incluir o circuito de geração da voltagem alta dentro do próprio smart-card, o que aumenta o custo do cartão, e *dificulta* a ação do atacante: o circuito para gerar a voltagem mais alta utiliza um capacitor, que pode ser identificado e destruído com a ajuda de um microscópio. Para aumentar a segurança, um encapsulamento inviolável pode ser utilizado, o que aumenta ainda mais o custo do sistema.

Mecanismos de Segurança Específicos de Redes Ad Hoc

A utilização dos mecanismos de segurança deve levar em consideração a relação custo/benefício de cada solução. Nesta seção são descritos os principais mecanismos de seguranças que podem ser utilizados nas redes ad hoc.

Distribuição de Chaves

O principal objetivo do gerenciamento de chaves é compartilhar uma chave com um grupo de participantes. Para tanto, quatro operações podem ser necessárias: a pré-distribuição, o transporte, a arbitração e o acordo de chaves [Murthy e Mano, 2004].

A Pré-Distribuição de Chaves consiste da distribuição das chaves pelos nós interessados antes do início da comunicação. Isto exige que todos os nós da rede sejam previamente conhecidos, embora não seja exigido que todos participem sempre da rede. Uma vez que esta fase concluída, não é possível inserir novos nós ou trocar chaves. Grupos de comunicação, que devem ter uma chave própria também devem ser estabelecidos nesta fase.

No Transporte de Chaves, as entidades trocam chaves para se comunicar. O método mais simples para essa fase se chama *Key Encription Key* (KEK), e consiste em criptografar a nova chave com o segredo compartilhado, e apenas os nós que possuírem esse segredo podem obter a nova chave. No caso de não existir uma chave previamente conhecida por um grupo, mas existir uma infra-estrutura de chave pública, essa nova chave pode ser trocada criptografando-a com a chave pública do nó que irá recebê-la.

A Arbitração de Chaves utiliza um arbitrador central para criar e distribuir chaves entre os participantes, o que a torna uma especialização da fase de transporte. Em sistemas infra-estruturados, um ponto central é escolhido para exercer a função de arbitrador. No entanto, em redes ad hoc, esta função centralizada de arbitrador é proibitiva por causa da ausência de infra-estrutura e restrições de recursos. Entre esses está a necessidade do arbitrador estar sempre ativo e acessível, sob pena de negação de serviço caso o nó se mova ou saia da rede, ou ainda tornar um único ponto vulnerável a ataques. A utilização de réplicas da base de dados para resolver o problema da negação de serviço aumentaria o número de nós guardando os segredos da rede, gerando mais pontos de vulnerabilidade, além de ser uma solução mais dispendiosa em termos de recursos.

Por fim, o Acordo de Chaves corresponde à troca de chaves posterior ao início da rede. Aqui serão estabelecidos segredos entre nós através de chaves assimétricas, se elas

estiverem disponíveis. Isto é necessário para realizar uma comunicação segura dentro da rede, embora seja uma operação muito custosa.

A seguir serão descritos os principais métodos propostos na literatura para distribuição de chaves, tanto simétricos quanto assimétricos.

Criptografia de Limiar

A criptografia de limiar (*threshold criptography*) é aplicada para solucionar o problema das Autoridades Certificadoras (AC). Através desse tipo de criptografia, um segredo D é dividido em n partes $(D_1, D_2, \ldots, D_i, \ldots, D_n)$, de maneira que o conhecimento de k ou mais D_i partes facilita o cálculo de D, enquanto que o conhecimento de $k-1$ ou menos partes não permitem determinar D. Um esquema como esse é chamado de esquema de limiar (k, n), e se aplica à distribuição da função de certificação de uma autoridade certificadora por um grupo de nós. Isto é eficiente em redes ad hoc, onde a existência de uma única autoridade certificadora introduz um ponto único de falha, enquanto o uso de réplicas da autoridade certificadora para melhorar a acessibilidade produz mais vulnerabilidades. Utilizando criptografia de limiar (k, n), onde $n = 2k-1$, obtém-se um esquema de segurança robusto, pois mesmo que metade dos nós da rede fiquem comprometidos, ainda é possível reconstruir a chave k. O compromisso que se busca com a criptografia de limiar é entre segurança e conveniência, pois o ideal para a segurança é que todos os pedaços fossem necessários, embora para conveniência o ideal fosse utilizar o menor número de pedaços possível. Em outras palavras, no ambiente ad hoc se busca o equilíbrio entre a disponibilidade e tolerância à intrusão. Assim, um adversário precisa destruir $(n-k+1)$ nós para indisponibilizar o serviço, ou ainda roubar o segredo de k nós para obter a chave secreta [Kong et al., 2001].

A criptografia de limiar é ideal para aplicações nas quais um grupo de indivíduos com interesses conflitantes mutuamente suspeitos devem cooperar. O esquema de criptografia de limiar proposto em [Shamir, 1979] é baseado na interpolação polinomial de Lagrange com complexidade $O(nlog^2(n))$. Neste esquema, cada compartilhamento é calculado de um grupo de polinômios com grau k em um esquema (k, n). Usando os k valores em um sistema de equações lineares é possível encontrar o segredo, e usando menos que k equações, se obtém uma indeterminação. Um outro esquema baseado nos espaços euclidianos pode ser encontrado em [Blakley, 1979], onde o número de espaços é dado por k e o segredo compartilhado é dado por um ponto no espaço. Cada compartilhamento é um plano contendo o segredo, e a intercessão de k planos determina o ponto. Se $k = 3$, na ausência de alguns compartilhamentos, se obterá um plano ou uma linha, com infinitas possibilidades, o que torna o segredo indeterminável [Gahlin, 2004].

Uma vez que em redes ad hoc não se deseja que nenhum nó possua o segredo da autoridade certificadora, pois ele se tornaria um alvo para ataques, não é adequado que a chave possa ser recuperada, mesmo juntando k pedaços da mesma. O que se utiliza é a criptografia de limiar para fazer assinaturas. Nessa criptografia, cada nó em posse de um pedaço da chave fará uma assinatura parcial da mensagem, de forma a que, quando todas as k assinaturas parciais forem obtidas, seja possível construir a assinatura completa. Na Figura 2.6 a chave S é dividida em n pedaços, que são distribuídos entre n nós. Um

nó que deseje uma assinatura para a mensagem *m* deverá enviá-la aos *n* nós, e ainda que *n*-*k* assinaturas se percam, ele será capaz de reconstituir a assinatura de *m*. Uma multi-assinatura de limiar através da combinação do RSA com a criptografia de limiar é proposta em [Frankel e Desmedt, 1992].

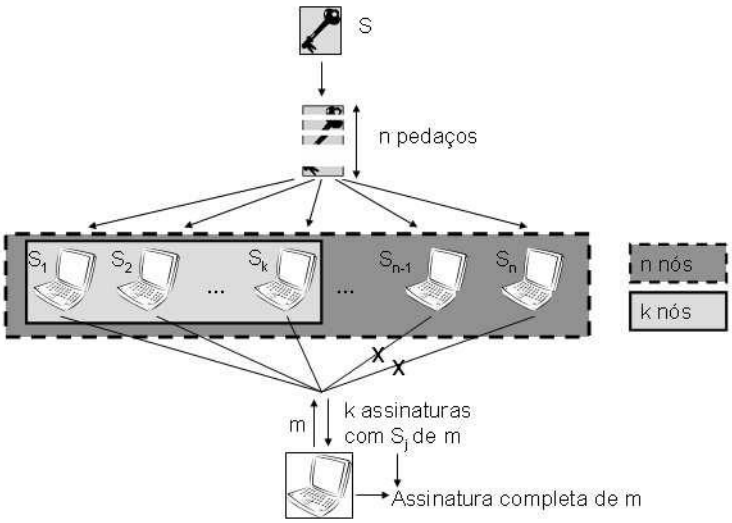

Figura 2.6. Esquema de assinatura com criptografia de limiar.

O maior problema desse tipo de criptografia é que os nós maliciosos enviarão assinaturas parciais inválidas, o que geraria uma assinatura completa inválida. Assim, o nó que desejar que sua mensagem seja validada deve ser capaz de testar todas as possibilidades de combinações de chaves até que ele obtenha uma assinatura completa válida. Esquemas mais robustos para essa composição de chaves também foram propostos, como o *Digital Signature Standard* (DSS) e são baseados em redundâncias entre as chaves [Zhou e Haas, 1999] [Gennaro et al., 1996].

O DSS é um esquema para melhorar o desempenho da criptografia de limiar. Uma primeira proposta mostrou que era possível proteger uma rede com *n* nós, sendo *t* nós maliciosos, dede que $n = t^2 - t + 1$. Isso significa que para *n* servidores, é possível garantir a resistência para $\sqrt{n}$ partes corruptas. Gennaro et al. mostraram um esquema de assinatura DSS, onde para obter um limiar de segurança *t* são necessários $2t + 1$ compartilhamentos ativos durante o cálculo da assinatura, obtendo limites de até $\frac{n-1}{2}$ [Gennaro et al., 1996]. Utilizando mecanismos para detectar e corrigir assinaturas erradas, foi criado um sistema robusto a *t* falhas, ou seja, que suporta até *t* invasores. Este sistema é capaz de se proteger contra nós que se recusam a cooperar, desde que $t < \frac{n}{3}$, ou ainda contra nós que se comportam de forma arbitrária e maliciosa, e nesse caso tem-se como premissa $t \leq \frac{n}{4}$. O DSS também suporta pró-atividade, como um meio de ser mais resistente a ataques.

Os sistemas de atualização de chaves, ou pró-ativos [Zhou e Haas, 1999] têm como princípio impedir que os atacantes consigam obter *k* compartilhamentos da chave. Uma vez que um atacante leva um tempo até conseguir roubar um segredo compartilhado de um nó, se os segredos nunca mudarem, ele consegue, após um longo tempo, roubar todos os *k* segredos. Se for aplicado um sistema de atualização de chaves, isso poderá ser impedido. Nos sistemas de criptografia de limiar pró-ativos, os novos segredos são com-

putados a partir dos antigos, por meio de colaboração dos servidores sem que o sistema de certificação deixe de funcionar. Os novos segredos compõem um compartilhamento $(n, t+1)$ da chave secreta, e não podem ser descobertos ainda que o invasor consiga obter todos os compartilhamentos antigos. O processo de obtenção dos novos segredos começa com todos os *n* nós gerando *n* compartilhamentos S_{ij} do compartilhamento S_i que ele possui do segredo *S*. Em seguida, por meio de um canal seguro, cada nó j receberá as chaves $S_{1\,j}$ até $S_{n\,j}$ dos seus vizinhos, podendo gerar $S^{j}_{j} = S_{j} + \sum_{i=1}^{n} S_{i\,j}$.

Pode-se, então, dizer que a técnica da criptografia de limiar trouxe melhorias às redes ad hoc, permitindo dar integridade e confidencialidade aos dados, autenticação, não-repúdio e disponibilidade do serviço de certificação.

Criptografia por ID

O sistema de criptografia por ID, proposto em [Shamir, 1984], tem como motivação inicial simplificar o gerenciamento de certificados de e-mail. A idéia é demonstrada pelo seguinte exemplo. Se Alice desejasse enviar um e-mail para Bob, ela não precisaria buscar o certificado para a chave pública de Bob, mas apenas criptografaria a mensagem utilizando a *string* bob@provedor.com. Quando Bob recebesse essa mensagem, caso ele não tivesse ainda gerado o seu segredo, ele se dirigiria ao *Private-Key Generation service* (PKG) e obteria o seu segredo. Dessa forma, Alice não seria impedida de enviar o e-mail caso Bob não tivesse obtido sua chave ou caso seu certificado já tivesse sido revogado [Boneh e Franklin, 2001].

Em um esquema de criptografia baseado em ID, os nós da rede terão como chave pública sua própria identificação, que pode ser qualquer valor arbitrário, e sua chave privada será gerada pela entidade PKG. Outra função da PKG é gerar as chaves mestras pública e privada, necessárias para criptografia e decriptografia, onde se assume que a chave mestra é conhecida por toda a rede. Para obter um nível maior de segurança, a PKG pode ser implementada utilizando criptografia de limiar. A grande vantagem deste esquema é que não é necessário que cada nó gere e divulgue a sua chave pública [Khalili et al., 2003].

As duas maneiras mais conhecidas para se realizar a criptografia de ID são o BF-IBE (Boneh and Franklin ID-Based Encription scheme) ou *Weil Paring* e o *Gap Diffie-Hellman* (GDH). O BF-IBE é um esquema baseado em mapas bilineares em curvas elípticas e é considerado o primeiro exemplo prático de uma criptografia por ID. O GDH tem sua segurança baseada na alta dificuldade de resolver o *Computational Diffie-Hellman Problem* (CDHP). Utilizando o GDH obtido pelo pareamento bilinear, obtém-se uma assinatura baseada em ID com os mesmos parâmetros do BF-IBE, com mesma eficiência, embora se saiba que o *Bilinear Diffie-Hellman Problem* (BDHP) seja mais complexo que o CDHP [Cha e Cheon, 2003].

O problema das assinaturas baseadas em ID é a dificuldade de prover a propriedade do não-repúdio e autenticação. Para tanto, é necessário proteger o PKG, o que pode ser feito através de criptografia de limiar, e possuir um sistema de arbitração de identificação. Caso a identidade do usuário não importe, bastando apenas conhecer o *hardware* que está dando origem às mensagens, a emissão de segredos pode ser feita para qualquer

ID arbitrário que faça o pedido.

Criptografia Comutativa

Para realizar o transporte de chaves na ausência de uma entidade central ou de um segredo compartilhado, existe um esquema proposto por Shamir chamado de protocolo dos três passes (*three-pass protocol*) [Shamir et al., 1978], baseado em criptografia comutativa.

Na criptografia comutativa existem duas funções, chamadas de f e g, que podem ser compostas, ou seja, existe $f(g(x))$, e que são reversíveis. Supondo que o nó X e o nó Y desejam se comunicar, o nó X irá escolher uma chave K que ele deseja utilizar para se comunicar com Y. Ele irá gerar também uma chave K_x para criptografar K com a função f, e envia o resultado para Y. O nó Y, por sua vez, ao receber essa mensagem gera uma chave K_y, criptografa tudo com K_y usando a função g e envia o valor resultante para X. O nó X, então, decriptografa a mensagem usando K_x e a inversa de f, enviando o resultado para Y. Este, por sua vez, utiliza a inversa de g e K_y para obter o valor K. Este processo pode ser visto esquematicamente na Figura 2.7.

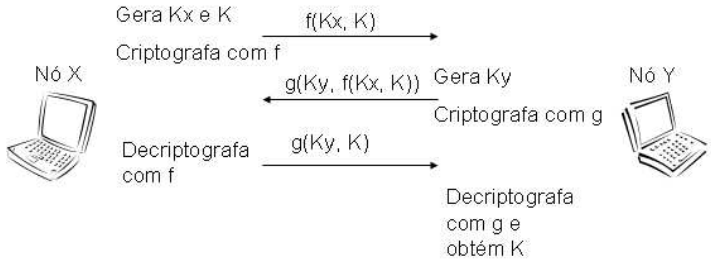

Figura 2.7. Criptografia comutativa [Murthy e Mano, 2004].

Esquemas de pré-distribuição de chaves

O princípio dos esquemas de pré-distribuição de chaves é fazer com que, a partir de um grupo de segredos pré-estabelecidos, os nós possam se comunicar baseados em segredos que possuem em comum. São necessários ainda mecanismos capazes de garantir que mesmo que dois nós não possuam segredos em comum, eles sejam capazes de se comunicar através de um caminho por pares que compartilham segredos.

Em [Eschenauer e Gligor, 2002] o primeiro esquema de pré-distribuição de chaves foi proposto. Neste esquema, uma fase de inicialização atribui um grupo de chaves S de um total de m chaves para cada nó. O número de chaves em S é escolhido de forma a garantir uma probabilidade p de encontrar pelo menos uma chave em comum entre quaisquer dois nós vizinhos. Ao entrar na rede, os nós tentam descobrir entre seus vizinhos se existem chaves em comum, através de desafios e identificadores de chaves. A chave que for compartilhada entre os dois nós se torna a chave para criptografia daquele enlace. Depois desta descoberta de chaves, um grafo conectado de enlaces seguros é formado, e

a partir deste são buscadas rotas para os nós vizinhos que não possuem uma chave em comum.

Em [Chan et al., 2003] são propostos três mecanismos para pré-distribuição de chaves, chamados de *q-composite random key predistribution scheme*, para redes de pequena escala, *multi-path key reinforcement scheme*, e *random-pairwise keys scheme*.

O esquema *q-composite random key predistribution* é uma melhoria para o esquema básico de Eschenauer e Gligor. A modificação consiste em checar a existência de q chaves em comum, ao invés de apenas uma. Aumentando o valor de q, é possível aumentar a resistência contra ataques de violação. Por outro lado, com este novo esquema é necessário buscar um equilíbrio, pois apesar do aumento de q melhorar a resistência contra o roubo de chaves, ele também força o uso de mais chaves em S para manter a probabilidade p, o que permite que o atacante consiga mais chaves com menos violações. O *multi-path key reinforcement scheme* foi outra melhoria proposta. Uma vez que a chave usada para criptografar comunicação entre dois nós faz parte do espaço de chaves, a violação de um nó da rede pode permitir obter o segredo da comunicação de diversos pares. A idéia do *multi-path key reinforcement scheme* é gerar uma chave de comunicação entre os nós a partir de um ou-exclusivo de vários números randômicos, transmitidos de uma extremidade a outra da comunicação por meio dos múltiplos caminhos existentes no grafo entre os dois nós. A chave não é passada apenas por um canal seguro porque o nó malicioso pode possuir a chave desse par e descobrir o segredo. Por outro lado, é pouco provável que o invasor possua as chaves de todas as rotas seguras. O terceiro esquema proposto, o *random-pairwise keys scheme*, traz como proposta permitir autenticação na comunicação. Com o esquema inicial, dois nós podem possuir o mesmo S, de forma que a comunicação destes com um terceiro nó pode estar utilizando a mesma chave, não permitindo que se identifique a origem exata da informação enviada. Assim, foi proposto que cada chave seja utilizada no máximo em dois grupos de chaves, de forma que ao escolher k para ser a chave criptográfica da comunicação, ela seja ligada a um ID em cada um dos lados da comunicação, garantindo a autenticação, já que nenhum outro nó possuirá aquela chave. Outra vantagem adicionada a esse método é a utilização da menor probabilidade p tal que a probabilidade de todo o grafo estar conectado ser c, permitindo que um número menor de chaves seja guardado.

Em [Newsome et al., 2004] são propostos três métodos de pré-distribuição de chaves para evitar o ataque Sybil, associando sempre grupos de chaves únicos a identidades, além de utilizar mecanismos de validação de chaves, tanto por um nó, quanto pela cooperação de vários nós da rede. O primeiro método proposto consiste de modificar o esquema básico utilizando funções pseudo-randômicas e *hash*, de forma a dificultar a criação de um ID a partir do roubo de um grupo de chaves. A segunda proposta é a adaptação do esquema de *random-pairwise keys* para utilizar grupos de chaves associados a identificações. Um terceiro esquema de *random-pairwise keys* pode ser feito utilizando uma informação pública U_i e uma privada V_i para cada nó, de forma que o segredo da comunicação do nó i com o nó j será dado por $f(V_i, U_j)$, o que é igual a $f(V_j, U_i)$, devido a características especiais da função f. Esse esquema impõe a restrição de no máximo λ nós comprometidos. O terceiro esquema proposto por Newsome et al. é o *Multi-Space Pairwise Key Distribution*, que é uma combinação dos métodos *random-pairwise keys* e básico modificado, aplicados de forma a dar uma maior segurança contra o roubo de

segredos de nós e a criação de novas identidades.

Algumas Considerações

Vários sistemas para gerenciar chaves têm sido recentemente propostos, utilizando os esquemas citados na seção anterior. No entanto, esses sistemas utilizados isoladamente não são capazes de garantir todos os requisitos de segurança da rede. Primeiramente, deve-se levar em conta que todos os sistemas que assumem a existência de um segredo pré-estabelecido supõem a atuação prévia de um administrador, o que não é condizente com o cenário de redes ad hoc, embora em muitos casos seja necessário. Além disso, a maioria dos sistemas apresentados consegue autenticar a origem das mensagens, embora não exista uma relação com o usuário que está naquela origem. Essa relação é essencial para a maioria das redes de computadores, e não existe nenhum mecanismo, senão a interferência de um administrador, de implementar estes esquemas isto em redes ad hoc.

Muitos estudos foram feitos sobre os esforços para quebrar os sistemas propostos, e a escolha por cada um deles deve levar em consideração tanto o custo de mensagens inseridas na rede quanto o custo dos recursos utilizados, além do custo de quebra, de acordo com o cenário utilizado. É evidente que o nível de segurança de um cenário militar é totalmente diferente do de uma rede ad hoc domiciliar, e isso deve ser considerado na escolha do mecanismo de gerenciamento de chaves.

Funções *Hash*

Dada a possibilidade de limitação de recursos dos nós de uma rede ad hoc, foram propostos mecanismos mais eficientes do que os tradicionais mecanismos de criptografia assimétrica. Esses novos mecanismos utilizam funções *hash* unidirecionais a fim de prover segurança nas tarefas de roteamento. Nessa seção, são descritos os Códigos de Autenticação de Mensagem, as cadeias de *hash* e as árvores de *hash*. Ao final, é apresentado o protocolo de autenticação TESLA, que utiliza estes mecanismos em seus esquemas de segurança.

Códigos de Autenticação de Mensagem

O uso de Códigos de Autenticação de Mensagem (*Message Authentication Code - MAC*) é uma maneira de garantir a integridade e autenticidade das informações trocadas por duas entidades através de um canal de comunicação inseguro. Tal mecanismo se baseia no compartilhamento de uma chave secreta entre as entidades. Quando uma das entidades deseja enviar uma mensagem à outra, ela anexa à mensagem um valor autenticador, denominado valor MAC ou simplesmente MAC, calculado em função da mensagem enviada e da chave secreta. Na recepção, a outra entidade, usando o mesmo procedimento e a mesma chave, recalcula o valor autenticador e compara com o valor MAC anexo à mensagem recebida. Somente se os valores forem iguais a informação recebida é considerada inalterada durante o trânsito pelo canal de comunicação. O objetivo desse mecanismo é impedir que, sem o conhecimento da chave secreta, um adversário seja capaz de forjar o valor MAC de uma nova mensagem, mesmo que muitas mensagens anteriores e seus

valores MAC correspondentes sejam conhecidos. Assim, os algoritmos MAC protegem as mensagens contra tentativas de falsificação, ou seja, tentativas de calcular o valor MAC sem o conhecimento da chave secreta.

A maioria dos algoritmos MAC foi construída usando cifradores de bloco (*block ciphers*) como o famoso DES. O algoritmo MAC desse tipo mais popular é conhecido como CBC MAC [Bellare et al., 1994]. Surgiram, porém, propostas de construção de MACs usando funções *hash* criptográficas como MD5 e SHA-1. Uma vantagem dessa abordagem é a sua simplicidade e eficiência, pois as funções *hash* populares são mais rápidas do que cifradores de bloco em implementações em *software*. Outra vantagem é que essas implementações são de domínio público. Os algoritmos que se baseiam nessa abordagem são denominados algoritmos HMAC [Bellare et al., 1996] e, devido as suas vantagens, são largamente utilizados para proteção das mensagens de roteamento em redes ad hoc.

Cadeia de *Hash*

Uma cadeia de *hash* (*hash chain*) é definida como uma seqüência, gerada a partir da aplicação sucessiva de uma função *hash* a uma semente, geralmente um número gerado aleatoriamente. Dado um dos elementos da cadeia de *hash*, pode-se garantir que os valores seguintes fazem parte da mesma cadeia, aplicando-se a função *hash* novamente sobre o elemento conhecido um número adequado de vezes. A unidirecionalidade da função *hash* impede que se obtenha os elementos anteriores da cadeia.

Para criar uma cadeia de *hash* de n elementos, um nó deve gerar uma semente aleatória h_0 e calcular a lista de valores $h_0, h_1, h_2, h_3, \ldots, h_n$, onde $h_i = H(h_{i-1})$ para $0 < i \leq n$. Dessa maneira, ao inicializar a cadeia de *hash*, os valores da lista são gerados da esquerda para direita. Em seguida, esses elementos podem ser utilizados para garantir a segurança da atualização das mensagens de roteamento, por exemplo. Nesse caso, para autenticar os campos atualizados da mensagem, o nó deve seguir da direita para a esquerda. Dado um elemento previamente autenticado da cadeia de *hash*, é possível verificar a pertinência dos elementos posteriores. Por exemplo, supondo conhecido o valor autenticado h_i, pode-se autenticar o elemento h_{i-3} calculando-se $H(H(H(h_{i-3})))$ e verificando-se se o valor calculado é igual ao valor previamente autenticado h_i.

Árvore de *Hash*

Outro mecanismo de segurança bastante usado pelos protocolos de roteamento seguro é o esquema de autenticação através de árvore de *hash* [Merkle, 1980]. Para autenticar os valores $v_0, v_1, \ldots, v_{w-1}$, estes são colocados como nós-folha de uma árvore binária, suposta balanceada por simplicidade. Em primeiro lugar, é aplicada uma função *hash* H a todos os valores v_i, isto é, $v'_i = H(v_i)$. Em seguida, é utilizada a construção de Merkle, ilustrada na Figura 2.8, na qual cada nó interno da árvore é obtido a partir de seus dois nós filhos.

Para se obter o elemento m a partir de seus nós filhos da esquerda $m_{esquerda}$ e

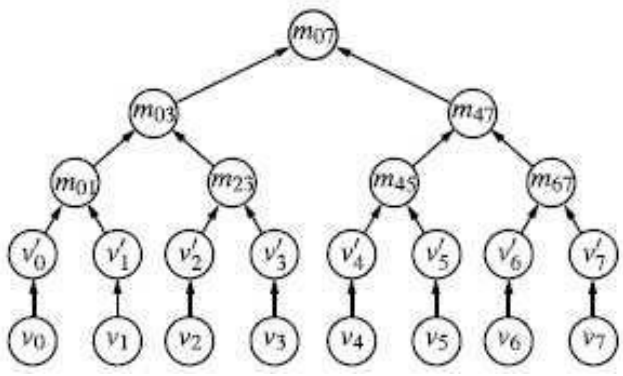

Figura 2.8. Exemplo de árvore de *hash*, extraído de [Hu et al., 2002].

da direita $m_{direita}$ deve-se calcular $H(m_{esquerda} || m_{direita})$, onde $||$ significa concatenação. Assim, os diversos níveis da árvore de *hash* são obtidos recursivamente a partir dos nós-folha da árvore. Por exemplo, na figura, $m_{01} = H(v'_0 || v'_1)$ e $m_{03} = H(m_{01} || m_{02})$. O nó-raiz da árvore pode ser utilizado para autenticar os nós-folha. Para autenticar o elemento v_i, deve-se divulgar os valores i, v_i e todos os nós-irmão dos nós da rota entre v_i e o nó-raiz. Por exemplo, para autenticar o elemento v_2 da Figura 2.8, deve-se conhecer os valores v'_3, m_{01}, m_{47} para calcular $H(H(m_{01} || H(H(v_2) || v'_3)) || m_{47})$. Caso o valor calculado seja igual a m_{07}, é assumido que o elemento v_2 é autêntico.

Autenticação TESLA

O protocolo TESLA [Perrig et al., 2002] é bastante eficiente para autenticação de mensagens e adiciona somente um único valor MAC à mensagem a ser transmitida para obter autenticação em difusão (*broadcast*). Em comunicações ponto-a-ponto, a utilização de algoritmos MAC para garantir a autenticação das mensagens é simples. Porém, em comunicações em difusão, os diversos destinatários teriam que conhecer a chave MAC, o que possibilitaria a ocorrência do ataque da Identidade Falsa. Por isso, a autenticação em difusão necessita de primitivas assimétricas. O protocolo TESLA difere dos protocolos assimétricos tradicionais, como RSA, pois a assimetria é obtida através de sincronização de relógios e atraso na divulgação da chave, ao invés de realizar operações que exigem grande poder computacional.

O protocolo TESLA determina que cada nó deve gerar uma cadeia de *hash* a partir de uma semente aleatória. Os elementos da cadeia serão utilizados como chaves para a autenticação das mensagens. O nó emissor deve divulgar o último valor da cadeia de *hash* gerada e, a partir daí, deve usar a cadeia no sentido inverso da geração para autenticar suas mensagens. Então, ao enviar uma mensagem, o nó emissor deve calcular o tempo médio que essa mensagem deve levar para chegar ao destino, divulgando a chave utilizada depois de decorrido esse tempo. Assim, os nós destinatários receberão a chave logo após terem recebido a mensagem. Com isso pode-se garantir que somente o nó emissor conhecia a chave TESLA utilizada para autenticar a mensagem recebida. Para verificar se a chave recebida está correta, deve-se aplicar a função *hash* sobre a chave um número adequado de vezes e comparar o resultado com o último elemento da cadeia de *hash*, que foi divulgado

pelo nó emissor. Se houver atraso no recebimento da mensagem ou a chave for divulgada antes que a estação de destino receba a mensagem, a mensagem deve ser descartada.

Um Novo Método de Confiança

As redes ad hoc dependem da colaboração dos nós para o seu bom funcionamento. No entanto, o comportamento de cada nó é dinâmico e depende dos seus objetivos e das suas limitações. Desta forma, cada nó da rede tende a decidir o que é melhor para si mesmo, sempre tentando maximizar seus objetivos. Entretanto, os nós da rede deveriam levar em consideração uma colaboração mínima, como em uma sociedade. Por isso, em uma rede ad hoc, uma ingênua dependência pode provocar baixa eficiência, alto consumo de energia e até mesmo ataques de nós maliciosos.

Existem alguns trabalhos que visam incentivar a colaboração dos nós em redes ad hoc através de sistemas de punição e incentivo [He et al., 2004] [Zhong et al., 2003] [Buttyan e Hubaux, 2000] [Buttyan e Hubaux, 2003]. O objetivo destes trabalhos é evitar a presença de nós egoístas criando um sistema de incentivo à colaboração e punição ao comportamento não colaborativo. Alguns trabalhos utilizam um sistema de crédito no qual cada nó recebe certa quantidade de unidades de crédito ao realizar uma ação que favoreça a um outro nó e o nó favorecido deve pagar pelo serviço que utilizou com seus créditos. Assim, nós egoístas seriam obrigados a colaborar a fim de receber uma quantidade suficiente de créditos que os permitam utilizar a rede. O grande problema destes sistemas é a necessidade de existir *hardwares* resistentes a alterações ou bancos virtuais em que todos os nós da rede possam confiar. Outra possibilidade para o sistema de incentivo/punição é a utilização de um esquema baseado na reputação dos nós. Neste caso, os nós devem ter mecanismos para avaliar e propagar a reputação dos outros nós da rede. Um sistema de reputação pode ser visto como um sistema de confiança. Apesar do estímulo à cooperação ser um ponto importante, não é suficiente para maximizar a eficiência da rede, por que os nós continuam dependendo de seus vizinhos de maneira ingênua.

Dentro deste contexto, a confiabilidade aparece como uma importante alternativa para viabilizar uma rede mais eficiente. A idéia é prover aos nós um mecanismo de confiança que os torne capaz de avaliar o grau de confiabilidade de seus vizinhos. Assim, os nós poderão sempre interagir com os vizinhos mais confiáveis, ignorando os vizinhos menos confiáveis.

A confiança é um conceito abrangente que engloba diversas definições. McKnight e Chervany [McKnight e Chervany, 2000] apresentam uma classificação conceitual da confiança. Neste caso, são abordados dois tipos de confiança. O primeiro está relacionado com o encaminhamento de pacotes, que permitirá aos nós decidir qual vizinho tem maior probabilidade de entregar um pacote corretamente, dado um determinado nó destino. O segundo é relacionado com a veracidade das informações recebidas de terceiros, que irá viabilizar a troca de informações entre os nós da rede de maneira eficiente e consistente.

Existem diversos trabalhos [Liu et al., 2004] [Pirzada e McDonald, 2004] que tratam da questão da confiança em redes ad hoc. No entanto, a maioria deles está focada apenas nos problemas de roteamento e de identificação de nós maliciosos.

Um novo modelo de confiança para redes ad hoc foi proposto por Velloso et al. O modelo visa simular as relações humanas de confiança e é baseado no aprendizado dos nós [Velloso et al., 2006]. A abordagem do modelo difere de outros trabalhos preocupados apenas com aspectos convencionais de segurança da rede, como a detecção de nós maliciosos, entre outros. O principal objetivo do modelo proposto por Velloso *et al.* é proporcionar aos nós de uma rede ad hoc uma maneira de avaliar e manter uma opinião sobre seus vizinhos, que servirá de base para a interação e a tomada de decisões entre eles. Assim, proporcionar um ambiente confiável não é um dos objetivos, mas sim capacitar os nós a reconhecer o ambiente ao qual pertencem. Isto é alcançado através da avaliação da confiabilidade de seus vizinhos. A informação de confiança será utilizada não apenas para o aprendizado e tomada de decisões, mas também poderá ser utilizada para a detecção e o isolamento de nós maliciosos.

O sistema proposto é distribuído e baseia-se na confiança que diferentes nós da rede possuem sobre um determinado nó sendo avaliado. O processo de avaliação do grau de confiança considera não somente o grau de confiança entre os nós adjacentes, mas também a precisão do grau de confiança e a maturidade do seu relacionamento. A fim de viabilizar a troca de recomendações, também foi proposto o protocolo REP (*Recommendation Exchange Protocol*), que simplifica a troca de informações de confiança na rede.

O modelo proposto pode ser representado por duas entidades distintas, como mostra a Figura 2.9. A entidade de Aprendizado é responsável por coletar e converter informações em conhecimento. A entidade de Confiança define como avaliar a confiança de um nó vizinho de acordo com o conhecimento adquirido pela entidade de aprendizado. Ambas as entidades interagem com todas as camadas. A entidade de Aprendizado considera o contexto do nó, que inclui o estado atual, as condições da rede, o lugar, a mobilidade e as ações de nós vizinhos para ajustar os parâmetros do modelo de confiança.

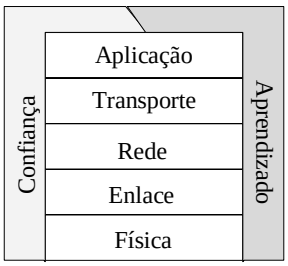

Figura 2.9. O modelo de confiança.

Velloso et al. consideram que os nós de uma rede realizam determinadas ações, como por exemplo, o envio de pacotes de dados, o encaminhamento de mensagens de roteamento, o descarte de pacotes de terceiros, entre outras possibilidades. O modelo parte do princípio que a ação de um nó tem sempre como objetivo maximizar o seu grau de satisfação. Desta forma, o grau de satisfação de um determinado nó é definido como o quão perto ele está do seu objetivo. O conjunto de ações de um nó determina o seu comportamento.

O grau de confiança é baseado nas experiências anteriores e na contribuição dos

nós vizinhos. As experiências anteriores resultam do julgamento das ações dos outros nós, realizado pela entidade de Aprendizado. Uma ação pode produzir um impacto positivo, negativo, ou nenhum impacto nos nós adjacentes. Os dois primeiros tipos geram uma reação que poderá disparar uma atualização do grau de confiança e eventualmente provocar uma mudança de comportamento. A capacidade de percepção das ações dos nós está diretamente relacionada com a eficiência da entidade de Aprendizado. Por exemplo, nós com sérias restrições de consumo de energia podem apresentar uma baixa eficiência por não poderem operar em modo promíscuo.

A contribuição dos nós vizinhos pode ser considerada no cálculo do grau de confiança. A contribuição é definida como o conjunto das recomendações dos nós vizinhos. Uma recomendação inclui o grau de confiança, a precisão do grau e a maturidade do relacionamento, que reflete a duração do relacionamento de confiança entre dois nós. Este conceito permite aos nós atribuir maior importância a recomendações baseadas em relacionamentos de mais longa duração. A precisão do grau de confiança pode ser vista como a confiabilidade na medida realizada, ou seja, a precisão representa a variação do grau de confiança sobre um determinado nó ao longo do tempo. Assim, para considerar as recomendações dos vizinhos, os nós devem utilizar o protocolo de troca de recomendações (*Recomendation Exchange Protocol* - REP).

Cada nó computa um grau de confiança para cada vizinho, o qual é atualizado sempre que necessário. Os nós são inteiramente responsáveis pelos próprios processos de avaliação do grau de confiança. Assim, o cálculo do grau de confiança é dividido em duas parcelas, como mostra a Equação 1, onde o valor do grau de confiança é uma variável continua limitada no intervalo $[0, 1]$, onde o valor 1 representa o grau mais confiável.

$$T_a(b) = (1 - \alpha)Q_a(b) + \alpha C_a(b), \quad (1)$$

onde α permite escolher o fator mais relevante. A primeira parcela, $Q_a(b)$, representa a capacidade de um nó de avaliar o grau de confiança baseado nas suas próprias informações, ou seja, utilizando apenas informações locais. A segunda parcela representa a contribuição dos nós vizinhos. A Equação 2 mostra como obter $Q_a(b)$, no modelo proposto.

$$Q_a(b) = \beta E_T + (1 - \beta)T_a(b), \quad (2)$$

onde E_T representa um valor de grau de confiança obtido através do julgamento das ações de terceiros. O parâmetro $T_a(b)$ é o valor antigo de grau de confiança armazenado na tabela de confiança. O parâmetro β permite escolher o termo mais relevante. Isto significa que o parâmetro β depende de qual evento desencadeou a atualização do grau de confiança. Por exemplo, supondo que o nó *a* começou uma atualização sobre o nó *b*, desencadeada por uma recomendação do nó vizinho *c*, mas o nó *a* não notou nada de estranho no comportamento do nó *b*. Neste caso, o nó *a* pode ignorar o primeiro termo da Equação 2. Por outro lado, caso a atualização tenha sido desencadeada por uma ação, o nó *a* pode escolher $\alpha 0$ e $\beta = 1$, ignorando a contribuição dos nós vizinhos (Equation 1) e o valor antigo para o grau de confiança sobre o nó *b* (Equação 2).

A contribuição é o conjunto das recomendações de todos os nós vizinhos. Assim, a parcela $C_a(b)$ representa a contribuição de todos os nós $i \in K_a$ sobre o nó *b* ponderada

pelo grau de confiança do nó a ($T_a(i)$) sobre o nó i, como mostra a Equação 3.

$$C_a(b) = \frac{\sum_{i \in K_a} T_a(i) M_i(b) X_i(b)}{\sum_{j \in K_a} T_a(j) M_j(b)}. \quad (3)$$

K_n define o grupo de nós do qual as recomendações serão consideradas no cálculo da contribuição de outros nós. Assim, K_n é um subgrupo dos vizinhos de n (N_a) que inclui todos os nós que satisfaçam uma determinada condição. Dentre as possíveis condições para seleção de K_n, duas são consideradas neste trabalho:

$$K_n = \{\forall i \in N_a | T_a(i) \geq T_{th}\}. \quad (4)$$

onde T_{th} representa o valor de confiança limiar a partir do qual um vizinho será considerado nas contribuições. Uma outra opção seria selecionar os r primeiros nós pertencentes à N_a de acordo com o grau de confiança. A seleção de K_n é uma decisão importante para o processo de avaliação da confiança, que depende de muitos parâmetros. A relevância da recomendação de cada nó ($T_i(b)$) é fortemente relacionada à seleção de K_a. Quanto mais confiável for K_a mais útil será a recomendação dos nós vizinhos. A contribuição inclui não somente o grau de confiança ($T_a(b)$), como também a precisão desta medida e a maturidade da relação, que representa há quanto tempo os nós se conhecem. A maturidade do relacionamento ($M_i(b)$) é representada em segundos por uma variável contínua e X é uma variável aleatória de distribuição normal que pode ser expressa por

$$X_i(b) = N(T_i(b), \sigma_i(b)) \quad (5)$$

onde σ representa a precisão e é definida como o desvio padrão, similar ao trabalho de Theodorakopoulos e Baras [Theodorakopoulos e Baras, 2004].

Cada valor na tabela de grau de confiança do nó i ($T_i(b)$) está associado a um valor de desvio padrão ($\sigma_i(b)$), que se refere à variação do valor do grau de confiança que o nó i observou. Assim, após uma atualização do grau de confiança do nó i sobre o nó b, o nó i deve atualizar o valor de $\sigma_i(b)$, que é definido como:

$$\sigma_i(b) = \sqrt{\frac{\sum_{j=1}^{k} (\bar{S}_k - S_j)^2}{k-1}}, \quad (6)$$

onde S_k representa o conjunto dos k últimas amostras de grau de confiança sobre o nó b, dado $k \in \mathbb{N} \mid k \geq 2 \leq 10 \leq S_k$ é o valor médio. O parâmetro σ expressa a confiabilidade da medida do grau de confiança. Um valor grande de σ pode demonstrar a dificuldade do nó de avaliar o grau de confiança ou a instabilidade do comportamento do nó que está sendo avaliado.

A recomendação do nó i sobre o nó b é ponderada pela maturidade $M_i(b)$. Isto significa que quanto maior for o tempo que os nós se conhecem, maior será a relevância da sua opinião para o valor da contribuição final de todos os nós vizinhos. Nós maliciosos podem tentar falsificar graus de confiança por diversas razões. Por exemplo, um nó pode querer difamar um outro vizinho, ou pode querer convencer os outros vizinhos que um determinado nó malicioso é, na verdade, um nó de boa índole, ou ainda apenas querer confundir os outros nós. Assim, bastaria colocar um valor alto para a maturidade do

relacionamento de um grau de confiança forjado, para que esta recomendação tivesse um grande peso no processo de atualização dos nós vizinhos. Para minimizar este efeito, cada nó deve definir um limiar para o valor de maturidade da relação (M_{max}) de tal forma que a maturidade pode ser expressa por:

$$\bar{M}(b) = \begin{cases} M_i(b), & \text{if } M_i(b) < M_{max} \\ M_{max}, & \text{if } M_i(b) \geq M_{max}. \end{cases} \quad (7)$$

O valor de (M_{max}) deve ser baseado na média dos valores de maturidade de relacionamento de todos seus vizinhos.

Para viabilizar a troca de recomendações foi concebido o protocolo de troca de recomendações (REP - *Recomendation Exchange Protocol*). Quando dois nós se encontram pela primeira vez, isto é, quando nós não vizinhos tornam-se vizinhos, uma mensagem de pedido de grau de confiança (*Trust Request - TREQ*) deve ser enviada em difusão. Por exemplo, quando o nó *a* e o nó *b* se encontram pela primeira vez, o nó *a* enviará uma TREQ com o identificador de nó igual a *b* ($T\ REQ_a^b$). O nó *b*, por sua vez, enviará uma $T\ REQ_b^a$. Os nós que recebem uma TREQ e possuem grau de confiança sobre o nó requisitado devem responder com uma mensagem de resposta de confiança (Trust Reply - TREP). Uma mensagem TREP possui a recomendação do nó remetente sobre o nó cujo grau de confiança foi requisitado. Após o envio de uma TREQ, o nó requerente deve esperar as respostas (TREP) de seus vizinhos por uma determinada quantidade de tempo (*timeout*). Caso o nó requerente não receba nenhuma TREP de seus vizinhos, a parcela da Equação 1 relativa à contribuição dos vizinhos deve ser ignorada, igualando o parâmetro α a 0.

Por último, a mensagem de anúncio de grau de confiança (*Trust Advertisement* - TA) é uma recomendação não solicitada. Uma mensagem TA inclui uma recomendação do nó que está anunciando sobre um de seus vizinhos. O envio de uma TA acontece sempre que uma atualização de grau de confiança gera um novo grau cuja diferença para a última TA enviada for maior que um determinado limiar ($TA_{threshold}$). A recepção de uma TA não implica necessariamente uma atualização do grau de confiança, mas apenas a atualização da recomendação recebida, para o caso em que o nó em questão seja também seu vizinho.

Um caso específico da Equação 1 acontece quando um nó toma conhecimento de um novo vizinho, para o qual não existe ainda nenhuma informação armazenada. Neste caso, a primeira parcela da equação, que representa as informações locais é substituída por um valor (F_a) previamente definido. Este valor está associado com a estratégia de atribuição do primeiro grau de confiança. Desta forma, reescrevendo a Equação 1 obtém-se:

$$T_a(b) = (\alpha - 1) \cdot F_a + \alpha \cdot C_a(b), \quad (8)$$

O valor de F_a dependerá das condições da rede, da mobilidade, do lugar e do estado atual do nó que irá atribuir.

A atualização do grau de confiança é um processo que pode ser desencadeado a qualquer momento, desde que o primeiro grau de confiança já tenha sido atribuído previamente. O modelo proposto considera que uma atualização é sempre desencadeada por um evento, no entanto, a ocorrência de um evento não implica necessariamente a

atualização do grau de confiança. A definição de um evento consiste na recepção de uma mensagem de recomendação ou na percepção de uma ação realizada por um dos vizinhos.

Outros Métodos

Existem alguns métodos simples para aumentar a segurança que podem ser implementados no projeto de protocolos de roteamento. Entre estes métodos estão a utilização de redundância, a verificação de bidirecionalidade do enlace, a autorização e a investigação.

Redundância

A redundância pode ser utilizada para prevenir diversos ataques em redes ad hoc, devido a sua característica de manter alternativas seguras, dando maior robustez à rede. O único ataque que pode inviabilizar a robustez gerada por redundância é o Sybil.

A utilização de múltiplas rotas é uma forma de se obter redundâncias no encaminhamento de dados. Embora as múltiplas rotas sejam usadas para balanceamento de carga e prevenção contra congestionamentos, elas também podem prover segurança, através do envio repetido dos dados pelos múltiplos caminhos. Assim, ainda que existam nós maliciosos em um caminho, descartando ou alterando pacotes, a mensagem real conseguirá ser entregue através de outra rota que não inclua esse nó. Para tanto, a escolha das rotas redundantes deve ser cuidadosa, escolhendo rotas disjuntas, ou seja, totalmente independentes, de forma a garantir que mesmo que um nó malicioso esteja em uma rota, ele não estará nas outras. O grande problema da utilização de rotas disjuntas é a dificuldade de encontrá-las em redes ad hoc, devido à baixa conectividade. Para resolver esta questão existem duas propostas. A primeira é ao invés de buscar rotas disjuntas, buscar os nós que são confiáveis na rede e tentar traçar múltiplas rotas passando por esses nós, independente de elas se sobreporem ou não [Ye et al., 2003]. A segunda proposta é a de utilizar caminhos trançados [Ganesan et al., 2001], que podem ter nós em comum, mas não possuem enlaces em comum, o que pode prover proteção probabilística contra o encaminhamento seletivo [Karlof e Wagner, 2003].

Apesar das vantagens trazidas pelo uso de múltiplos caminhos, é preciso observar também os contrapontos dessa medida. O uso de múltiplos caminhos pode sobrecarregar a rede, pois o tráfego total irá ser multiplicado pelo número de rotas entre pares, o que restringe o uso dessa técnica a redes que não trabalham próximas a saturação.

A utilização de armazenamento distribuído na rede é outro tipo de redundância [Newsome et al., 2004]. Este mecanismo, quando utilizado para prover segurança, constitui-se de distribuir várias réplicas de uma base de dados por diversos nós, de forma que a falha de um nó não comprometa a disponibilidade das informações.

Verificação de Bidirecionalidade do Enlace

Esta verificação está diretamente ligada à forma como o protocolo de roteamento determina quais são os seus vizinhos. Apesar de freqüentemente o roteamento executar acima da camada de enlace IEEE 802.11, nem sempre a informação gerada por ele através do ACK de enlace sobre quem são os vizinhos reais é utilizada. É comum que protocolos de roteamento assumam que todos os enlaces são bidirecionais, o que não é verdade em redes sem fio. Assim, eles determinam que os vizinhos de um nó são todos os nós que

ele é capaz de ouvir. Isso gera vulnerabilidades que permitem, por exemplo, o ataque da Inundação de *Hello*.

A verificação do estado do enlace é simples e não traz grandes custos ao funcionamento da rede. Basta que todos os nós enviem mensagens periodicamente, anunciando quem são os nós que eles são capazes de escutar. Qualquer nó que escute uma mensagem em que ele esteja listado como um possível vizinho pode considerar o emissor da mensagem como seu vizinho real.

Autorização

A autorização [Wood e Stankovic, 2002] é uma defesa contra os ataques do Direcionamento Falso, Encaminhamento Seletivo e Buraco Negro, que se baseia em escolher nós que serão os únicos autorizados a encaminhar informações de roteamento. Para tanto, é necessária a existência de autoridades certificadoras ou *Private-Key Generation Services*, para que seja possível autenticar os nós autorizados. Assim, qualquer pacote de roteamento que chegue por meio de um nó não-autorizado deve ser imediatamente descartado.

Para o bom desempenho da Autorização é necessário um bom sistema de confiabilidade, para que apenas nós que possam ser considerados de boa índole sejam escolhidos para essa função.

Investigação

A investigação é um teste que qualquer nó pode realizar para avaliar a conectividade da rede. Em particular em redes onde se tem o conhecimento da estrutura física, é possível enviar pacotes que atravessem o diâmetro da rede, funcionando como sondas. Essas sondas realizarão uma investigação sobre a rede, avaliando a existência ou não de áreas desconectadas. As sondas não são capazes de distinguir regiões de ataques de regiões de falhas, mas já trazem boas contribuições para a determinação das rotas a serem utilizadas.

É importante notar que um pacote de sonda deve ser indistinguível de pacotes normais, para que os nós que descartam pacotes maliciosamente não o encaminhem apenas para reforçar a validade da rota que o inclui. Para um bom desempenho, esses pacotes devem ser enviados periodicamente, tanto para manter um mapa das rotas disponíveis atualizado, quanto para aumentar a probabilidade de descobrir quais são os nós que realizam encaminhamento seletivo, egoísmo e ganância.

Compressão

Técnicas de compressão podem ser utilizadas para aumentar a segurança durante a atualização de bases de dados entre dois nós. Um tipo de compressão especial para essa função é chamado de Compressão Delta, e funciona através remoção de redundâncias das diferenças entre as duas bases de dados que serão atualizadas, de forma a diminuir ao máximo o tráfego na rede. A conseqüência disto é que a previsibilidade dos dados que estão sendo transmitidos cai muito, dificultando a ação de espiões (*eavesdroppers*), mesmo na ausência de algoritmos criptográficos [Bing, 2006].

Protocolos de Roteamento Seguros

Os protocolos de roteamento inicialmente propostos para redes ad hoc sem fio consideravam apenas cenários onde os nós eram confiáveis, premissa que compromete a sua eficiência em ambientes hostis. Para suprir a necessidade de segurança, foram projetados novos protocolos, que serão descritos nessa seção.

Cabe ressaltar aqui a diferença entre roteamento seguro e transmissão segura. As mensagens encaminhadas em uma rede ad hoc podem ser divididas em dois tipos: mensagens de controle e mensagens de dados. Entre as mensagens de controle, destacamos as de roteamento, trocadas pelos nós da rede a fim de estabelecer uma rota entre os nós que desejam se comunicar. Para isso, essas mensagens devem ser processadas em cada nó e, possivelmente, modificadas antes de serem encaminhadas, de acordo com as regras definidas pelo protocolo de roteamento. Já as mensagens de dados só podem ser trocadas após o estabelecimento da rota e seu conteúdo deve ser mantido, preferencialmente em sigilo, até o destino. Assim, as mensagens de roteamento e as de dados possuem naturezas e propósitos distintos, exigindo requisitos de segurança diferentes. Em geral, para ambas as mensagens se deseja garantir a integridade e autenticação, pois cada nó deseja receber dados corretos e precisa confiar nas informações de roteamento recebidas.

Os protocolos de roteamento de redes ad hoc móveis existentes estão expostos tanto a ataques passivos como ativos (Seção 2.3). Os ataques ativos mais freqüentes são os ataques de fabricação, alteração e modificação de pacotes, Identidade Falsa (*Spoofing* ou *Impersonating*) e o túnel de minhoca (*wormhole*). Com a publicação destes ataques, tornou-se evidente a fragilidade dos protocolos, e percebeu-se que propostas de segurança como IPSec (*IP Security*) [Atkinson, 1995] somente são válidas para autenticação fim-a-fim e para prover segurança entre entidades que já possuem roteamento estabelecido entre si, não garantindo roteamento seguro. Então, foram propostos novos protocolos que visam garantir a integridade, a autenticidade e o não-repúdio das mensagens de roteamento. Os novos protocolos projetados para prover segurança, expostos a seguir, utilizam os mecanismos descritos na Seção 2.4 a fim garantir segurança no roteamento.

ARAN

O ARAN (*Authenticated Routing for Ad hoc Networks*) [Sanzgiri et al., 2002] é um protocolo de roteamento seguro reativo que garante autenticidade, integridade e não-repúdio das mensagens de roteamento, baseado em criptografia assimétrica. Para isso, é suposta a existência de um servidor certificador confiável, cuja chave pública deve ser conhecida por todos os nós da rede, responsável pela manutenção e distribuição dos certificados de todos os nós da rede.

Após um processo preliminar de certificação, pelo qual todos os nós devem passar antes de ingressarem na rede, o protocolo ARAN prevê dois procedimentos. O primeiro é um procedimento obrigatório que garante autenticação fim-a-fim no processo de descoberta de rota. O segundo procedimento é opcional e tem por objetivo garantir de maneira segura que o caminho obtido no processo de descoberta de rota é o menor caminho até o destino.

O primeiro procedimento é iniciado quando o nó de origem envia um pacote de

descoberta de rota (*Route Discovery Packet - RDP*) por difusão (*broadcast*) para os seus vizinhos a fim de determinar uma rota para um dado destino. O pacote RDP possui o endereço IP do nó de destino, o certificado do nó de origem, o número de seqüência e um *timestamp*. Antes de ser enviado, todo o pacote RDP é assinado com a chave privada do nó de origem. O próximo nó que receber o pacote RDP deve validar a assinatura usando o certificado do nó de origem, e, em seguida, atualiza sua tabela de roteamento com os endereços dos nós de origem e destino, assina a mensagem com sua chave privada e a envia para seus vizinhos, juntamente com o seu certificado. As próximas estações que receberem a mensagem devem repetir o mesmo procedimento, retirando sempre a assinatura do nó anterior antes de assinar a mensagem e reenviá-la para os nós vizinhos. Dessa maneira, as estações intermediárias vão encaminhando o pacote até o destino final. O encaminhamento só não é feito se alguma condição de segurança não for respeitada, como a assinatura do nó anterior não ser válida, indicando um ataque de identidade falsa. Outra condição é que os nós não devem encaminhar mensagens que tiverem o mesmo número de seqüência e o mesmo destino de uma mensagem já encaminhada, evitando ataques de replicação. O número de seqüência do pacote RDP é incrementado monotonicamente toda vez que um nó inicia um processo de descoberta de rota. O nó de destino responde somente ao primeiro pacote RDP recebido, descartando os demais que possuam a mesma origem e o mesmo número de seqüência, o que não garante o melhor caminho. O pacote de resposta (*Reply Packet - REP*), que contém o endereço do nó de origem, o certificado do nó de destino, o mesmo número de seqüência e o mesmo *timestamp* do pacote RDP recebido, deve ser assinado antes de ser enviado ao nó de origem. A resposta, então, segue a rota reversa usando um processo similar ao utilizado na descoberta de rota, exceto que agora a comunicação entre os nós é ponto-a-ponto. O nó de origem, ao receber o pacote de resposta, verifica sua autenticidade usando o número de seqüência e a assinatura do nó de destino.

O segundo procedimento previsto pelo protocolo ARAN visa assegurar que a descoberta do caminho ótimo. Assim como no procedimento anterior, o nó de origem envia por difusão um pacote de confirmação de caminho ótimo (*Shortest Path Confirmation - SPC*) assinado para os seus vizinhos, contendo os mesmos campos do RDP mais dois campos adicionais para acomodar o certificado do nó de destino e a mensagem criptografada usando a chave pública do nó de destino. É utilizada uma espécie de roteamento cebola (*onion routing*) [Syverson et al., 1997], onde cada nó que receber a mensagem deve assiná-la, incorporar o seu certificado e criptografar novamente a mensagem usando a chave pública do nó de destino. Quando os pacotes SPC chegam ao nó de destino, este verifica todas as assinaturas e responde ao primeiro pacote SPC e a todos os pacotes posteriores que tiverem percorrido uma rota menor.

O protocolo ARAN ainda possui esquemas que garantem o não-repúdio das mensagens de erro e um processo seguro de revogação de certificados. Assim, em conjunto, os mecanismos de defesa do protocolo oferecem alta proteção contra a maioria dos ataques, embora o uso de criptografia assimétrica exija um alto poder computacional e o ataque do túnel de minhoca não tenha sido solucionado.

Ariadne

O Ariadne [Hu et al., 2005] é um protocolo de roteamento seguro baseado no protocolo reativo DSR (*Dynamic Source Routing*) [Johnson e Maltz, 1996]. O Ariadne se baseia em criptografia simétrica, tendo como vantagem a alta eficiência e simplicidade desse mecanismo. O protocolo provê ainda autenticação ponto-a-ponto das mensagens de roteamento usando um Código de Autenticação de Mensagem (*Message Authentication Code - MAC*) e uma chave secreta compartilhada pelas duas entidades. No entanto, para autenticar mensagens enviadas por difusão é utilizado um esquema de autenticação TESLA similar ao descrito na Seção 2.4.2.

Para garantir a autenticação fim-a-fim do esquema de descoberta de rota, o nó de origem calcula um MAC da mensagem usando a chave secreta que somente os nós de origem e destino conhecem. Dessa maneira, assegura-se que a mensagem veio realmente do nó de origem e que as informações de roteamento não foram alteradas. Antes de enviar a mensagem, o nó de origem estima um tempo máximo para o atraso fim-a-fim e inclui esta informação na mensagem, juntamente com uma lista de nós e uma lista de MACs, ambas inicialmente vazias. Após o término do tempo estimado, o nó de origem irá divulgar sua chave TESLA. Assim, quando uma estação intermediária recebe a mensagem, ela verifica se o tempo de divulgação da chave já expirou. Se o resultado for positivo, ela descarta a mensagem, e, se não for, insere seu endereço na lista de nós. A integridade da lista de endereços é obtida através do mecanismo de cadeias de *hash*, usando um esquema idêntico ao utilizado pelo SAODV. Assim, nesse momento é calculado um novo *hash* sobre o campo *Hash Chain*. A estação intermediária ainda utiliza sua chave TESLA atual para computar o MAC da mensagem, que é inserido na lista de MACs. Finalmente, a mensagem modificada é reenviada para os vizinhos, como no DSR, procedimento que é repetido recursivamente pelas estações intermediárias até chegar ao nó de destino. Quando o nó de destino recebe a mensagem, ele verifica se o valor final da cadeia de *hash* está correto e se as chaves TESLA já foram divulgadas. Caso a mensagem recebida seja válida, o nó de destino calcula o MAC da resposta usando a chave secreta compartilhada com o nó de origem e envia a mensagem de resposta para a origem. Ao final da rota reversa, a origem autentica a resposta antes de aceitá-la.

O protocolo ainda prevê o uso da autenticação TESLA para impedir a fabricação de mensagens de erro falsas e um mecanismo para autenticação da descoberta de rota que permite aos nós limitar a taxa de requisições de descoberta de rota vindas de outros nós. Além disso, uma versão avançada do protocolo possui proteção contra o ataque do túnel de minhoca (Seção 2.3.2), usando o protocolo TIK (*TESLA with Instant Key disclosure*) [Hu et al., 2003a]. Apesar de tantas vantagens, o Ariadne exige alguns pré-requisitos para ser utilizado, como algum mecanismo seguro de distribuição das chaves TESLA dos nós, um esquema para o estabelecimento das chaves secretas compartilhadas pelos nós comunicantes e ainda é requerida uma sincronização de tempo fraca entre os nós que possibilite estimar o tempo de transmissão fim-a-fim para outro nó da rede.

SRP

O SRP (*Secure Routing Protocol*) [Papadimitratos e Haas, 2002] foi projetado para manter o roteamento correto em redes ad hoc onde ocorrem mudanças freqüentes e onde

pode haver nós maliciosos, porém que não agem em conluio. O protocolo foi concebido como uma extensão, que pode ser aplicada em diversos protocolos de roteamento reativos existentes, em particular o DSR [Johnson e Maltz, 1996] e o IERP (*Interzone Routing Protocol*) [Haas et al., 2001]. No SRP, o nó iniciador do procedimento de descoberta de rota é capaz de identificar e descartar respostas contendo informações de roteamento falsas ou ainda evitar recebê-las, garantindo a obtenção de informações topológicas corretas. Para isso, é suposta a existência de uma Associação de Segurança (*Security Association - SA*) entre os nós comunicantes, como uma chave simétrica compartilhada. Além disso, é suposto que os nós possuem uma única interface de rede, com uma correspondência biunívoca entre os endereços IP e o da interface. Sob essas hipóteses, os autores provam que o protocolo é robusto.

Ao iniciar o procedimento de descoberta de rota, o nó de origem deve gerar um MAC, usando uma função *hash* com chave que recebe como argumentos de entrada o cabeçalho IP, os campos básicos da mensagem de roteamento e a chave secreta compartilhada entre os nós de origem e destino. As estações intermediárias são responsáveis por encaminhar a mensagem até o seu destino final. Quando o nó de destino recebe a mensagem de roteamento, ele não somente verifica a integridade da mensagem, como também assegura a autenticidade da origem, uma vez que o MAC só pode ser calculado pelos nós que possuem a chave secreta, garantindo que somente a origem poderia ter computado o MAC recebido. Caso a mensagem recebida seja autêntica e íntegra, o nó destino envia uma mensagem de resposta ao nó de origem realizando o mesmo procedimento feito pelo nó de origem. O nó de origem, ao receber a mensagem de resposta, verifica sua integridade usando o MAC computado pelo nó de destino e descarta a resposta se ela não tiver o mesmo identificador da mensagem inicial.

O protocolo ainda possui um interessante mecanismo de regulação das requisições de descoberta de rota. Cada nó mede as freqüências de requisições realizadas pelos seus vizinhos e mantém uma fila na qual a prioridade de atendimento às requisições é inversamente proporcional à freqüência com que elas são feitas. Assim, fica caracterizado um mecanismo de *feedback* negativo que controla a freqüência de requisições realizadas pelos nós vizinhos, impedindo ataques nos quais o nó malicioso inunda a rede com requisições de descoberta de rota, já que o atacante será atendido por último ou ignorado, dada a sua baixa prioridade de atendimento.

Uma das principais vulnerabilidades do SRP é a ausência de autenticação das mensagens de erro, embora no esquema proposto o nó malicioso só consiga prejudicar rotas às quais ele pertence. Outra desvantagem é que, como o protocolo não previne ações maliciosas em conluio, ele não está imune aos ataques de atração e descarte de pacotes. Apesar disso, o SRP possui a grande vantagem da imunidade aos ataques que modificam a origem do pacote ou simulam identidades. Isso se deve ao protocolo NLP (*Neighbor Lookup Protocol*) de descoberta de vizinhos, integrante do SRP, que mantém um mapeamento dos endereços da subcamada de acesso ao meio (*Medium Access Control*) e da camada de rede dos nós da rede.

SEAD

Os mesmos autores do Ariadne propuseram também mecanismos para garantir a segurança de protocolos de roteamento pró-ativos, desenvolveram o SEAD (*Secure Efficient Ad hoc Distance vector*) [Hu et al., 2002], que é baseado no protocolo de vetor distância DSDV (*Destination-Sequenced Distance Vector routing*) [Perkins e Bhagwat, 1994]. O SEAD foi projetado para suportar a existência de nós com capacidade de processamento limitada e possui defesas contra ataques de negação de serviço nos quais o atacante visa esgotar recursos da vítima, como banda passante e processamento. Para isso, são utilizados mecanismos de segurança reconhecidamente eficientes, como funções *hash* unidirecionais e criptografia simétrica. Foi provada a robustez do protocolo contra ataques ativos não coordenados ou efeitos causados por nós comprometidos.

O SEAD utiliza a técnica de cadeia de *hash* para autenticar os campos número de saltos e número de seqüência. A cadeia é criada ao se aplicar repetidas vezes uma função *hash* a um valor inicial aleatório, e é assumida a existência de algum mecanismo que permita que um nó distribua um elemento autenticado da cadeia para seus vizinhos. Um elemento autenticado da cadeia de *hash* é utilizado para garantir uma atualização segura das mensagens de roteamento. Os elementos posteriores da cadeia podem ser autenticados aplicando-se sobre eles a função *hash* um número adequado de vezes. A distribuição do elemento autenticado não é parte do protocolo, embora os autores sugiram o uso de uma autoridade certificadora confiável para distribuição das chaves públicas dos nós. Dessa maneira, um dado nó poderia usar sua chave pública para assinar o elemento da cadeia de *hash* e distribuí-lo.

São propostos ainda dois mecanismos para autenticação dos nós vizinhos. O primeiro mecanismo utiliza autenticação TESLA (Seção 2.4.2), exigindo sincronização entre os nós da rede. O outro mecanismo supõe a existência de uma chave secreta compartilhada para cada par de nós que desejam se comunicar. Essa chave secreta é usada com um código de autenticação de mensagem MAC para garantir a autenticidade dos nós vizinhos.

O protocolo SEAD não possui proteção contra o ataque do túnel de minhoca (Seção 2.3.2). Porém, como no Ariadne, os autores sugerem o uso do protocolo TIK (*TESLA with Instant Key disclosure*) [Hu et al., 2003a] para detectar esse tipo de ataque. Além disso, dada a natureza unidirecional da função *hash*, o uso da cadeia de *hash* impede que nós maliciosos diminuam o campo de número de saltos, tentando forjar rotas melhores do que elas realmente são. Outra vantagem provém do uso de mecanismos de autenticação de vizinhos, que protegem contra ataques de Identidade Falsa.

SAODV

O *Secure AODV (SAODV)* [Zapata, 2002] é uma extensão do protocolo AODV (*Ad Hoc On-Demand Distance Vector Routing Protocol*) [Perkins et al., 2003], que garante segurança no processo de descoberta de rota. Zapata e Asokan propõem adicionar algumas mensagens chamadas de "Extensão de Assinatura" (*Signature Extension*), no intuito de acrescentar funcionalidades ao protocolo que permitam assegurar a integridade, a autenticação e o não-repúdio das informações de roteamento.

O SAODV possui dois mecanismos de proteção das mensagens de roteamento:

assinatura digital e cadeias de *hash*. A assinatura digital garante autenticação fim-a-fim dos campos imutáveis da mensagem, que devem ser assinados pelo nó de origem antes do envio da mensagem. No entanto, o campo Número de Saltos (*Hop Count*) deve ser decrementado a cada salto pelas estações intermediárias, o que impossibilita a adoção da abordagem anterior. Nesse caso, a cadeia de *hash* é usada para autenticar o campo Número de Saltos a cada salto, garantindo a integridade do vetor de distância. Ao enviar uma mensagem, o nó de origem deve inicializar o campo *Hash* com uma semente aleatória, o campo Número Máximo de Saltos com o valor desejado e o campo *Top Hash* com o resultado da aplicação da função *hash* sobre a semente um número de vezes igual a Número Máximo de Saltos . Quando um nó intermediário receber a mensagem, poderá autenticar o campo Número de Saltos, aplicando a mesma função *hash* um número de vezes igual a diferença entre Número Máximo de Saltos e Número de Saltos sobre o campo *Hash* e comparando o resultado com o campo *Top Hash*. Se os campos foram iguais, pode-se assegurar que o campo Número de Saltos não foi alterado e, então, o nó intermediário deverá incrementar o campo Número de Saltos em 1 e aplicar a função *hash* no campo *Hash* antes de reencaminhar a mensagem. Caso contrário, a mensagem deverá ser descartada. Dessa maneira, o mecanismo de cadeias de *hash* impede que um nó malicioso diminua o Número de Saltos da mensagem de roteamento, uma vez que não se pode obter os valores anteriores da seqüência, dada a propriedade unidirecional da função *hash*.

O SAODV ainda prevê extensões de assinaturas duplas (*Double Signature Extension*), que permitem que nós intermediários respondam imediatamente à origem quando eles já possuírem uma rota atualizada para o destino, assim como é previsto no AODV original. Além disso, o SAODV propõe que todos os nós devem possuir um meio de armazenar os seus números de seqüência, mesmo quando são reiniciados, a fim de evitar ataques de número de seqüência que atrairiam mais tráfego para o nó malicioso, por suas rotas parecerem mais atualizadas. Finalmente, o protocolo ainda propõe um mecanismo de autenticação das mensagens de erro.

SLSP

O protocolo seguro baseado em estado do enlace denominado SLSP (*Secure Link State Routing Protocol*) [Papadimitratos e Haas, 2003] foi proposto pelos mesmos autores do SRP. É suposta a existência de um par de chaves assimétricas para cada interface de rede de um nó e de um sistema de gerenciamento das chaves públicas dos nós. O SLSP é composto por três procedimentos: distribuição de chaves públicas, descoberta de vizinhos e atualização dos estados dos enlaces.

O procedimento de distribuição das chaves públicas dos nós é feito de forma distribuída, a fim de evitar o uso de um servidor central de gerenciamento de chaves. Cada nó envia por difusão (*broadcast*) um pacote de distribuição de chaves públicas (*Public Key Distribution packet - PKD*) assinado para os vizinhos da sua zona. Esse pacote contém o certificado de chave pública do nó de origem que poderá ser usado posteriormente para autenticar os pacotes oriundos daquela fonte.

Assim como no SRP, é utilizado o protocolo NLP (*Neighbor Lookup Protocol*) de descoberta de vizinhos. Cada nó envia aos seus vizinhos uma mensagem de *hello* assinada contendo seus endereços MAC (*Medium Access Control*) e IP. Ao receberem a

mensagem, os vizinhos validam sua assinatura e, em seguida, a aceitam. Dessa maneira, o protocolo NLP mantém um mapeamento dos endereços MAC e IP dos nós da rede e fica responsável por enviar notificações no caso de ocorrência de discrepâncias, como o mesmo MAC possuir dois ou mais endereços IP ou um nó tentando obter o endereço MAC de um nó existente.

O procedimento de atualização dos estados dos enlaces utiliza pacotes denominados LSU (*Link State Update*), identificados pelo endereço IP do nó de origem e um número de seqüência. Assim como no SEAD e no SAODV, o número de saltos, indicado pelo campo *hops_traversed*, é autenticado usando a técnica de cadeia de *hash*. Assim, ao receber um pacote LSU, os nós devem atualizar o campo *hops_traversed*, substituindo-o pelo seu valor *hash* e, em seguida, reencaminhá-lo. Antes disso, porém, a assinatura do pacote recebido deve ser verificada usando a chave pública obtida no procedimento de distribuição de chaves públicas.

Os três procedimentos e o protocolo NLP, em conjunto, garantem segurança no processo de descoberta de topologia e proteção contra ataques que modificam a origem do pacote ou simulam identidades. O protocolo SLSP também possui um mecanismo de regulação da taxa de recepção de pacotes idêntico ao do SRP, oferecendo proteção contra ataques de negação de serviço. Apesar de todas essas vantagens, o SLSP é vulnerável a ataques em conluio nos quais os atacantes fabricam enlaces inexistentes entre eles e inundam a rede com as informações falsas.

Extensões a Protocolos de Roteamento

Esta seção apresenta algumas extensões para os protocolos de roteamento existentes. Apesar de nem todas as propostas constituírem um protocolo completo, elas podem ser aplicadas aos protocolos que já existem de forma a garantir segurança nas tarefas de roteamento.

SOLSR

O SOLSR (*Secure Optimized Link State Routing Protocol*) [Hafslund et al., 2004] é uma versão segura do protocolo de roteamento OLSR [Jacquet et al., 2001], cuja idéia é assinar, usando chaves simétricas, cada pacote de controle do OLSR a fim de garantir a autenticidade das mensagens.

Uma vantagem do SOLSR é que a autenticação é feita salto a salto. Isso significa que é garantida a segurança até de campos que devem ser atualizados pelos nós intermediários, como o número de saltos e o campo TTL (*time-to-tive*). Além disso, somente é necessária uma assinatura por salto, já que muitas mensagens de roteamento são encapsuladas em um único pacote do OLSR. Em compensação, a abordagem salto a salto não garante assinaturas fim-a-fim, já que um pacote recebido por um nó não terá sido assinado pelo nó de origem, mas pelo nó anterior. Apesar disso, o protocolo determina que os nós só devem encaminhar pacotes oriundos de nós confiáveis. Conseqüentemente, os nós de uma dada rota serão confiáveis, dois a dois. O processo de assinatura digital utiliza uma função *hash* com chave, de forma que um nó que não tenha acesso à chave secreta não poderá reproduzir a assinatura do nó emissor.

O SOLSR possui mensagens próprias para acomodar as assinaturas, de forma a

garantir a compatibilidade com nós que não estejam operando a versão segura do OLSR. Além disso, para evitar ataques de replicação, o SOLSR utiliza a técnica de *timestamp*. Foi proposto um mecanismo bidirecional de troca de *timestamps* que possui a vantagem de não exigir sincronização dos nós.

SAR

O protocolo SAR (*Security-aware Ad hoc Routing*) [Yi et al., 2001] torna seguros os processos de descoberta e manutenção de rotas de protocolos de roteamento reativos como o AODV e o DSR. A idéia básica da proposta é incorporar métricas de segurança às mensagens de roteamento no intuito de estabelecer um nível de "qualidade de segurança" que será usado na comunicação entre os nós da rede ad hoc. O SAR trata os requisitos de segurança de forma semelhante àquela com que os requisitos de qualidade de serviço (*Quality of Service - QoS*) são tratados.

As métricas de segurança do protocolo podem ser especificadas por níveis hierárquicos de confiança entre os nós ou por requisitos de segurança como autenticidade e não-repúdio. Os níveis de confiança podem ser definidos a partir de um sistema de distribuição de chaves ou de compartilhamento de uma chave secreta entre os nós. Assim, somente os nós que pertencem a um determinado nível de confiança podem trocar mensagens entre si, uma vez que nós de outros níveis de confiança não terão como decriptografar as mensagens daquele nível. Por outro lado, os requisitos de segurança podem ser implementados usando técnicas conhecidas como certificação e números de seqüência.

Uma implementação do protocolo SAR baseada no AODV também foi proposta pelos autores. O *Security-aware AODV* acrescenta às mensagens de roteamento do AODV campos que definem a métrica de segurança utilizada. Assim, somente os nós que possuem o nível de segurança especificado podem ser incorporados durante o processo de descoberta de rota. Dessa forma, ao receber um pacote de requisição de rota, o nó de destino pode ter a certeza de que existe uma rota para o nó de origem e que os componentes dessa rota podem respeitar o nível de segurança estabelecido pelo nó de origem.

TIARA

A proposta chamada TIARA (*Techniques for Intrusion Resistant Ad hoc Routing Algorithms*) [Ramanujan e Edin, 2000] consiste em um conjunto de princípios de projeto e técnicas que podem ser aplicadas aos protocolos de roteamento atuais, em especial aos protocolos reativos como o AODV e o DSR, para prover resistência a ataques de negação de serviço.

O primeiro princípio, denominado controle de acesso a rotas baseado em fluxo (*Flow-based Route Access Control - FRAC*), utiliza uma lista de controle de acesso, armazenada em cada nó, que identifica os fluxos de mensagens que possuem autorização para serem encaminhados. O roteamento por múltiplos caminhos é outra técnica utilizada pelo TIARA. Nesse caso, os procedimentos de descoberta e manutenção de rota compulsoriamente trabalham para que todas as rotas para um dado fluxo sejam encontradas e mantidas, o que assegura tolerância 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 deve ser tomado. O TIARA também utiliza um mecanismo de mo-

nitoramento de fluxo, 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. Caso a diferença entre o número de pacotes recebidos pelo destino e o número 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. Para autenticação dos pacotes, o protocolo utiliza um mecanismo denominado autenticação rápida (*Fast Authentication*), que obriga os nós a colocar o rótulo do caminho numa localização secreta em 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 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 só são permitidas no caso de o nó de origem apresentar recomendações oriundas de nós confiáveis que garantam a autenticidade do seu pedido.

BISS

O protocolo BISS (*Building secure routing out of an Incomplete Set of Security associations*) [Capkun e Hubaux, 2003] é um conjunto de otimizações que podem ser aplicadas aos protocolos de roteamento existentes. A proposta dos autores é a construção de um protocolo de roteamento seguro onde não haja necessidade de associações seguras entre todos os pares de nós, mas apenas entre uma fração deles. Para atingir tal objetivo, é proposto que a autenticação dos nós intermediários ao longo do processo de descoberta de rota deve ser realizada não somente através das associações de segurança pré-existentes, mas também através da troca de certificados de chave pública entre os nós.

Em uma aplicação do BISS ao DSR, os pacotes de requisição de rota devem conter o certificado da origem assinado por uma autoridade certificadora e chave pública do nó de origem, que deve assinar o pacote antes de enviá-lo aos seus vizinhos. Os nós intermediários devem verificar a assinatura do nó anterior e, usando uma associação segura pré-existente, autenticar o destino. Assim como no Ariadne, ao reencaminhar o pacote, os nós intermediários devem calcular o MAC do pacote usando a chave secreta que compartilham com o destino, permitindo a este a verificação da autenticidade dos nós da rota. Em seguida, o nó de destino deve enviar um pacote de resposta ao nó de origem. Se o destino compartilhar uma chave secreta com a origem, a resposta será protegida com um MAC calculado com essa chave. Caso contrário, o pacote de resposta deve ser assinado para que o nó de origem possa autenticá-lo. Os nós intermediários devem proceder da mesma forma. Assim, o nó de origem é capaz de autenticar o pacote de resposta, verificando os MACs e as assinaturas incluídas pelo nó de destino e os nós intermediários. De forma análoga, as mensagens de erro serão autenticadas por um MAC, se os nós compartilharem uma chave secreta, ou usando assinatura digital, caso não possuam uma associação de segurança pré-existente, mas possam usar os certificados de chave pública correspondentes.

O BISS possui a vantagem de aumentar o número de associações seguras da rede, pois chaves e certificados anteriormente desconhecidos podem ser distribuídos ao longo do processo de descoberta de rota. Esse método permite também o estabelecimento de

chaves simétricas entre os nós, que podem ser utilizadas para futuras verificações de mensagens.

SMT

O projeto de um sistema completo de segurança deve abranger as tarefas de roteamento e encaminhamento de dados em uma rede ad hoc. Apesar de ser uma condição necessária, apenas o roteamento seguro não é suficiente para atingir tal objetivo, pois os protocolos de roteamento não garantem que os nós de uma rota corretamente descoberta irão encaminhar as mensagens de dados posteriores da forma esperada. Um nó malicioso pode enganar o sistema de segurança, fornecendo informações corretas durante o processo de descoberta de rota, mas se comportando indevidamente durante a fase de encaminhamento de dados.

O SMT (*Secure Message Transmission protocol*), proposto por Papadimitratos e Haas, também é uma extensão que se aplica aos protocolos de roteamento, mas tendo como principal objetivo garantir o encaminhamento seguro dos dados, após o estabelecimento da rota entre a origem e o destino, utilizando informações de roteamentos para determinar quais são as rotas seguras. O SMT aproveita algumas vantagens do ambiente ad hoc para conseguir um encaminhamento de dados seguro e tolerante a falhas, utilizando as redundâncias topológicas características da rede ad hoc e criptografia simétrica para alcançar tal objetivo. Além disso, é suposta uma associação segura somente entre os nós de origem e destino.

O SMT utiliza as informações do protocolo de roteamento seguro operante para determinar um conjunto de caminhos que ligam o nó de origem ao de destino. Em seguida, a mensagem a ser transmitida é dividida em partes, segundo o algoritmo de inserção de redundância limitada de Rabin [Rabin, 1989], de forma que mesmo a recepção de apenas uma fração da mensagem original possibilitará sua reconstrução pelo nó de destino. Cada parte da mensagem original é transmitida por um caminho diferente e possui um cabeçalho com informações de criptografia que permitem ao nó de destino garantir a autenticidade e a integridade dos dados recebidos, além de assegurar proteção contra ataques de replicação. Após receber algumas partes da mensagem original, o nó de destino envia mensagens de reconhecimento ao nó de origem informando quais partes permaneceram intactas e quais rotas são confiáveis. Para garantir a robustez desse mecanismo de *feedback*, as mensagens de reconhecimento são protegidas da mesma forma que as mensagens de dados e são uniformemente distribuídas ao longo das rotas reversas confiáveis. Dessa maneira, o retorno de uma única mensagem de reconhecimento é suficiente para o correto funcionamento do protocolo. Caso o nó de destino não receba um número suficiente de partes para reconstruir a mensagem original, o nó de origem deve retransmitir as partes restantes através das rotas confiáveis. Além disso, se o protocolo determinar que as rotas inicialmente escolhidas não foram suficientemente boas, um novo conjunto de rotas deve ser escolhido.

O protocolo SMT tem como vantagem o fato de não ser exigido nenhum processamento adicional dos nós intermediários, que devem simplesmente reencaminhar os pacotes recebidos. Além disso, os autores mostram que o protocolo consegue 100% de sucesso de recepção das mensagens transmitidas, ainda que haja um ambiente altamente hostil, como o caso em que 20% dos nós da rede são maliciosos. Por outro lado, o uso de

múltiplos caminhos e, conseqüentemente, o grande número de nós envolvidos na transmissão de uma única mensagem é o preço pago para se obter a robustez desejada.

Protocolos Baseados em Reputação

Uma defesa contra ações egoístas e maliciosas é a adoção de um sistema de reputação. O objetivo dessa abordagem é incentivar o comportamento que leve a uma confiança crescente entre os nós. Para isso, são realizadas decisões baseadas na confiança entre as entidades. Em geral, as propostas utilizam monitoramento passivo das transações, troca de recomendações entre nós e algum mecanismo de geração de mensagens de alarme. A seguir, são descritos os principais protocolos baseados em reputação encontrados na literatura.

OSRP

O OSRP (*On-demand Secure Routing Protocol Resilient to Byzantine Failures*) [Baruch Awerbuch e Rubens, 2002] é um protocolo reativo que provê resistência contra falhas bizantinas (Seção 2.3.2) causadas por nós individuais ou em conluio. A idéia básica do protocolo é atribuir pesos crescentes aos caminhos onde houver detecção de falhas bizantinas, de forma que se possa descobrir um caminho livre de falhas, caso exista, ainda que haja nós maliciosos agindo em conluio na rede. Assim, ao invés de identificar o nó malicioso em si, o OSRP visa atribuir um peso alto aos seus enlaces. O protocolo é dividido em três partes: descoberta de rota livre de colisões, detecção de falhas bizantinas e gerenciamento de peso dos enlaces.

A primeira fase é responsável por encontrar a rota de menor peso entre os nós de origem e de destino. Para isso, é usada a técnica de assinatura digital, que requer a existência de uma infra-estrutura de chave pública que certifique a autenticidade das chaves públicas dos nós da rede. Ao iniciar o processo de descoberta de rota, o nó de origem envia, por difusão (*broadcast*), uma mensagem de requisição de rota assinada para todos os seus vizinhos. Essa mensagem contém os endereços dos nós de origem e destino, um número de seqüência e uma lista de pesos. O próximo nó que receber a mensagem irá verificar sua assinatura, adicionar a requisição de rota a sua lista e reencaminhar a mensagem, procedimento que será repetido por todos os nós intermediários até que a mensagem atinja seu destino final. Quando receber a mensagem de requisição de rota, o nó de destino irá verificar sua assinatura e, em seguida, enviar uma mensagem de resposta assinada aos seus vizinhos. A mensagem de resposta irá percorrer a rota reversa e cada nó intermediário deverá calcular o peso total do caminho até o nó em questão. Caso o valor calculado seja menor que aquele encontrado em uma mensagem de resposta anterior com o mesmo número de seqüência, então o nó intermediário deverá verificar todas as assinaturas, anexar o seu endereço, assinar a mensagem e reenviá-la em difusão. O nó de origem deve realizar esse mesmo procedimento ao receber uma mensagem de resposta, atualizando o destino da rota caso o caminho seja menor que o atual.

A fase de detecção de falhas bizantinas tem como objetivo descobrir enlaces com falhas no caminho obtido na fase anterior. Para isso, em cada pacote de dados é inserida uma lista de nós, denominados nós de teste (*probe nodes*). Os nós de teste devem enviar ao nó de origem um reconhecimento para cada pacote recebido. Caso o número

de pacotes sem reconhecimento ultrapassar um limite estipulado, uma falha é registrada. Para garantir a autenticidade e a integridade dos reconhecimentos, o protocolo requer a existência de chaves secretas compartilhadas entre o nó de origem e cada nó de teste.

A terceira fase do OSRP é responsável por manter uma lista de pesos dos enlaces de acordo com o resultado da fase anterior. Essa lista de pesos é utilizada pela primeira fase do protocolo para evitar os caminhos com falhas. Nessa fase, o peso dos enlaces com falha é dobrado, de forma a penalizar o comportamento bizantino do nó daquele enlace.

As três fases do protocolo OSRP são executadas seqüencialmente e a resposta de uma fase é utilizada como entrada para outra. Sob as hipóteses impostas, os autores provam que o protocolo é resistente a falhas bizantinas. Porém, o OSRP é incapaz de prevenir ataques de atração de tráfego, caso eles sejam realizados sem um comportamento bizantino.

CONFIDANT

O CONFIDANT (*Cooperation Of Nodes: Fairness In Dynamic Ad hoc NeTworks*) [Buchegger e Boudec, 2002] é outro protocolo baseado em reputação encontrado na literatura recente. O protocolo é composto por quatro módulos e foi proposto como uma extensão de segurança para o protocolo de roteamento DSR [Johnson e Maltz, 1996].

O primeiro módulo é responsável pelo monitoramento passivo dos reconhecimentos de cada mensagem que um nó encaminha. Quando um nó transmite uma mensagem, ele tenta monitorar, usando o modo promíscuo da sua interface de rede, a transmissão do próximo nó a fim de assegurar que a mensagem foi reencaminhada devidamente. O próximo módulo, denominado gerenciador de confiança (*trust manager*), é responsável pelo envio e pelo recebimento de mensagens de alarme. Essas mensagens são trocadas por nós que foram pré-definidos como confiáveis e visam informar comportamentos indevidos na rede. O terceiro módulo consiste em um sistema de reputação que reúne as informações geradas pelos módulos anteriores e atribui notas aos nós, mantendo uma tabela de reputações para cada nó da rede. As notas são atribuídas de acordo com um critério que dá maior importância às experiências locais do que as mensagens de alerta recebidas de outros nós. Se a nota de certo nó ultrapassar um limite de segurança, o último módulo, denominado de gerenciador de rotas, é chamado para remover os caminhos que contiverem o nó com comportamento suspeito. Para isso, ele utiliza a tabela de reputação dos nós, escolhendo o melhor caminho a ser tomado. Além disso, são ignoradas as mensagens de roteamento oriundas de nós maliciosos e são gerados alertas para os nós legítimos avisando quando é feito um pedido de requisição de rota que usa um caminho comprometido.

É importante notar que o protocolo CONFIDANT suporta somente experiências negativas, que são apagadas após um determinado tempo sem detecção de ações suspeitas. Além disso, os nós da rede ad hoc devem operar os quatro módulos do protocolo para o seu correto funcionamento. Dessa maneira, o CONFIDANT é capaz de detectar, alertar e reagir a ataques no encaminhamento das mensagens de dados e de roteamento, embora a utilização do modo promíscuo esteja sujeita a críticas, devido às limitações de energia dos nós da rede.

Tendências Futuras

Este capítulo apresentou as principais vulnerabilidades e ataques às redes ad hoc, classificando-os segundo os efeitos que causam e as camadas da pilha de protocolos nas quais eles atuam. Foram então apresentados os mecanismos de segurança propostos na literatura, ressaltando-se as vantagens e desvantagens de cada um. Destaca-se entre os mecanismos um novo método de confiabilidade, que permite que um grau maior de segurança seja obtido através da avaliação do comportamento dos nós vizinhos. Por fim, foram selecionados e descritos alguns dos principais protocolos recentemente propostos. Essa seção conclui este trabalho descrevendo as linhas de pesquisa atuais e as tendências futuras da segurança de redes ad hoc.

Antes de apresentar as tendências futuras da segurança de MANETs, deve-se salientar algumas limitações das técnicas atuais. Em primeiro lugar, os mecanismos de segurança propostos recentemente foram projetados para proteger a rede ad hoc somente contra ataques conhecidos. No entanto, espera-se que surjam novos ataques que explorem as vulnerabilidades das técnicas existentes. Dessa maneira, não há uma solução completa, sendo necessários novos esforços cada vez que novas vulnerabilidades foram descobertas. Em segundo lugar, cada mecanismo de segurança acrescenta *overhead* e complexidade. Conseqüentemente, a limitação de recursos dos nós da MANET impede a ativação simultânea de todos os mecanismos existentes. Por último, as técnicas que usam chaves criptográficas necessitam de um sistema distribuído, robusto e seguro que garanta a geração, a distribuição e o gerenciamento dessas chaves. Essa é outra questão de segurança ainda aberta, podendo-se encontrar diversos trabalhos publicados recentemente nessa área de pesquisa.

O rápido crescimento das redes ad hoc móveis nos últimos anos, provendo soluções satisfatórias para os problemas técnicos a que se propunham, leva a crer num futuro promissor para essa tecnologia. Entretanto, há questões de segurança ainda não completamente solucionadas, em especial às que se referem as características do meio físico e do IEEE 802.11. Entre os ataques específicos de redes ad hoc, as soluções propostas são eficientes, embora a premissa de existir um método de autenticação automático e seguro ainda seja um problema. Além disso, os requisitos de segurança são dependentes das aplicações e dos ambientes, que podem variar significativamente. Por exemplo, as MANETs podem ser utilizadas tanto para prover conexão sem fio flexível e de curto alcance para estabelecimentos comerciais, como para conectar dispositivos militares capazes de realizar operações criptográficas de alto desempenho. Enquanto que aplicações militares podem exigir altos níveis de segurança, outras aplicações precisam de requisitos de segurança simples. Além disso, acredita-se que, no futuro, diversos dispositivos e usuários terão de coexistir numa rede ad hoc grande, aberta, ubíqua e autônoma. Dessa forma, uma linha de pesquisa atual visa encontrar soluções eficientes para esse problema.

Outra linha de pesquisa recente procura mecanismos que consigam proteger a rede ad hoc contra nós comprometidos ou com comportamento indesejável. Esses mecanismos não devem exigir alto poder computacional, devem fazer suposições limitadas acerca das relações de confiança entre os nós e ainda agregar um baixo *overhead*. Nesse sentido, destacam-se os sistemas de confiança, como o novo método apresentado na seção 2.4.3.

Outra questão ainda aberta é o roteamento seguro (seção 2.5). Mecanismos exis-

tentes, como o IPSec, não oferecem uma solução completa para o problema. A literatura apresenta duas abordagens para resolver essa questão: esquemas pró-ativos de combate aos ataques existentes ou sistemas de detecção de intrusão. Porém as propostas existentes agregam bastante *overhead* e computação aos protocolos, sem conseguir oferecer uma solução satisfatória para todos os ataques conhecidos.

Alguns estudos que vêm sendo feitos atualmente visam inserir redundâncias nos canais de comunicação de forma a reduzir o impacto de ataques de negação de serviço e combater comportamentos maliciosos, usando múltiplos caminhos entre a fonte e o destino do fluxo de dados.

Outros pontos em aberto estão ligados ao controle de acesso aos serviços da rede (nível de aplicação) e aos grupos de encaminhamento de pacotes (nível de transporte), ao estabelecimento de chaves de grupo para uma rede com topologia arbitrária, ou ainda à aplicação das técnicas atuais aos campos emergentes das redes de sensores ad hoc e das redes ad hoc veiculares (*Vehicular Ad hoc NETworks - VANETs*).

Finalmente, conforme argumentado na Seção 2.1, a natureza das MANETS possui vulnerabilidades intrínsecas que não podem ser removidas. Assim, continua sendo um campo interessante a busca por soluções que possam reunir as propostas atuais de forma a garantir um nível mínimo de segurança que cada aplicação exige, sem gerar uma sobrecarga da rede.