CONTROLE DE ACESSO DISTRIBUÍDO PARA REDES AD HOC

Natalia Castro Fernandes

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

_____

Prof. Otto Carlos Muniz Bandeira Duarte, Dr. Ing.

_____

Prof. Joni da Silva Fraga , Dr.

_____

Prof. Luís Henrique Maciel Kosmalski Costa , Dr.

RIO DE JANEIRO, RJ - BRASIL
FEVEREIRO DE 2008

FERNANDES, NATALIA CASTRO

Controle de Acesso Distribuído para Redes Ad Hoc [Rio de Janeiro] 2008

XV, 117 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia Elétrica, 2008)

Dissertação - Universidade Federal do Rio de Janeiro, COPPE

1. Segurança

2. Redes ad hoc

3. Controle de acesso

I. COPPE/UFRJ II. Título (série)

À minha família e ao meu querido João Kleber.

Agradecimentos

Agradeço, inicialmente, a Deus pela oportunidade de estar defendendo a minha tese de mestrado. Agradeço também à minha família, pelo incentivo e apoio em todas as horas. Aos amigos e ao meu namorado, também fico grata pelas horas de diversão e pelas sugestões recebidas ao longo da tese. Agradeço, em especial, aos meus amigos do GTA pela companhia e pelas sugestões recebidas. Destaco entre esses, o Danilo, a Carina e o Reinaldo, pelos trabalhos que realizamos juntos e aos meus companheiros de sala, pela disponibilidade em todos os momentos.

Agradeço também a todos os professores da COPPE/UFRJ por todos os conhecimentos e orientações recebidos ao longo do mestrado, que me foram de grande utilidade para o desenvolvimento da tese. Em especial, agradeço ao professor Otto, meu orientador de iniciação científica e de mestrado, pela amizade, conselhos e lições aprendidas e pela oportunidade de ter entrado no GTA. Agradeço também ao professor Luís Henrique, pela amizade, atenção e ajudas durante o desenvolvimento da tese. Por participar da minha banca examinadora, agradeço novamente ao professor Luís Henrique e também ao professor Joni.

Agradeço aos órgãos de fomento à pesquisa FAPERJ, CAPES, CNPQ e RNP, pelos recursos financeiros recebidos, que permitiram o curso do mestrado com bolsa e a participação em congressos da área. Também aos funcionários do Programa de Engenharia Elétrica da COPPE/UFRJ, pela presteza no atendimento na secretaria do Programa.

Por fim, um obrigado especial a todos que, embora não estejam citados aqui, me incentivaram, contribuindo de forma direta ou indireta, para a minha formação acadêmica e profissional.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

CONTROLE DE ACESSO DISTRIBUÍDO PARA REDES AD HOC

Natalia Castro Fernandes

Fevereiro/2008

Orientador: Otto Carlos Muniz Bandeira Duarte

Programa: Engenharia Elétrica

As redes ad hoc sem fio têm como vantagens a ausência de infra-estrutura, a auto-configuração, o baixo custo de instalação e também a possibilidade de mobilidade dos usuários. Por outro lado, essas redes possuem muitas vulnerabilidades, devido à ausência de um elemento central, ao roteamento colaborativo e às freqüentes desconexões da rede. Devido a essas características, os mecanismos de segurança adotados em redes cabeadas não se aplicam às redes ad hoc. Este trabalho aborda o problema de segurança de uma rede ad hoc e em particular o controle de acesso. São propostos mecanismos de segurança que têm por objetivo identificar, autenticar e monitorar os nós, utilizando poucos recursos dos dispositivos. Os resultados mostram que, separadamente, os mecanismos propostos apresentam vantagens com relação a outras propostas. Além disso, quando utilizados em conjunto, os mecanismos propostos oferecem um sistema de controle de acesso robusto e adequado às características das redes ad hoc.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

DISTRIBUTED ACCESS CONTROL IN AD HOC NETWORKS

Natalia Castro Fernandes

February/2008

Advisor: Otto Carlos Muniz Bandeira Duarte

Department: Electrical Engineering

Ad hoc wireless networks advantages are the absence of infrastructure, the auto-configuration, the low installation cost, and also the possibility of user mobility. These networks, however, have many vulnerabilities due to the absence of central entities, the collaborative routing, and the frequent network disconnections. Due to these characteristics, security mechanisms for wired networks do not apply for ad hoc networks. This work broaches the ad hoc network security quest, particularly the access control. We proposed three security mechanisms, whose objectives are to identify, authenticate and monitor nodes, using a few node resources. The results show that, individually, each of the mechanisms has advantages over other proposals. Besides, the use of them simultaneously creates a robust access control which is adequate to ad hoc networks characteristics.

Sumário

Lista de Figuras

Lista de Acrônimos

AES :	*Advanced Encryption Standard*;
AMORA :	Autenticação e MOnitoração em Redes Ad hoc;
AODV :	*Ad Hoc On-Demand Distance Vector routing protocol*;
AREP :	*Address Reply*;
AREQ :	*Address Request*;
AUFIRA :	AUtoconfiguração de endereços baseado em FIltros para Redes Ad hoc;
BD :	*Burmester-Desmedt*;
CHARADAS :	CHAve de grupo no Roteamento Através de Distribuição Assimétrica Segura;
DAD :	*Duplicate Address Detection*;
DAP :	*Dynamic Address assignment Protocol in mobile ad-hoc networks*;
DECA :	*Distributed, Efficient Clustering Approach protocol*;
DHCP :	*Dynamic Host Configuration Protocol*;
DTN :	*Delay and Disruption Tolerant Networks*;
GDH-3 :	*Group Diffie-Hellman*;
HMAC :	*keyed-Hash Message Authentication Code*;
ID :	Identificação;
IEEE :	*Institute of Electrical and Electronics Engineers*;
IP :	*Internet Protocol*;
IPv4 :	*Internet Protocol version 4*;
IPv6 :	*Internet Protocol version 6*;
MAC :	*Medium Access Control*;
MANET :	*Mobile Ad hoc NETworks*;
MPR :	*Multipoint Relay*;

OLSR : *Optimized Link State Routing protocol*;

PDA : *Personal Digital Assistant*;

PGP : *Pretty Good Privacy*;

PKG : *Private Key Generator*;

PKI : *Public Key Infrastructure*;

RSA : *Rivest-Shamir-Adleman*;

SAODV : *Secure Ad hoc On demand Distance Vector*;

SDI : Sistema de Detecção de Intrusão;

SOLSR : *Secure Optimized Link State Routing protocol*;

TCP : *Transmission Control Protocol*;

VANET : *Vehicular Ad Hoc Networks*;

WMN : *Wireless Mesh Network*.

Capítulo 1

Introdução

O cenário atual da Internet e das redes locais permitiu o desenvolvimento de redes móveis e a sua popularização. As tecnologias para redes móveis, hoje, são inúmeras e atingem grande parte da população, principalmente devido às redes de celulares. O avanço da Internet e a necessidade dos usuários manterem-se sempre conectados provocam investimentos para a instalação de redes sem fio em ambientes empresariais, industriais, domiciliares e comunitários. Os investimentos previstos para os próximos anos atingem grandes cifras e projetam-se novas aplicações como o controle automático do estoque doméstico, assistência e monitoração remota de funções vitais de idosos e convalescentes, jogos, lazer, entre outros.

Estuda-se hoje o modelo a ser aplicado à "Internet do Futuro", que deve ser escalável, autônoma, ubíqua e segura. Nesta rede, homens e também dispositivos (coisas) com capacidade de processamento e comunicação poderão se conectar à Internet em qualquer lugar e a qualquer momento, justificando sua denominação Internet de Coisas (*Internet of Things*) [1]. Os desafios a serem vencidos por esta "nova Internet" são enormes. As redes sem fio assumirão um papel fundamental para garantir mobilidade e ubiqüidade. Para permitir o amplo acesso à Internet, além das redes sem fio infra-estruturadas, algumas redes de nova geração, como as redes ad hoc móveis (*Mobile Ad Hoc Networks* - MANETs) [2], as redes ad hoc veiculares (*Vehicular Ad Hoc Networks* - VANETs) [3], as redes em malha sem fio (*Wireless Mesh Networks* - WMNs) [4] e as redes tolerantes a atrasos e desconexões (*Delay and Disruption Tolerant Networks* - DTNs) [5] deverão ser

desenvolvidas. No entanto, a popularização destas redes requer um avanço significativo na segurança.

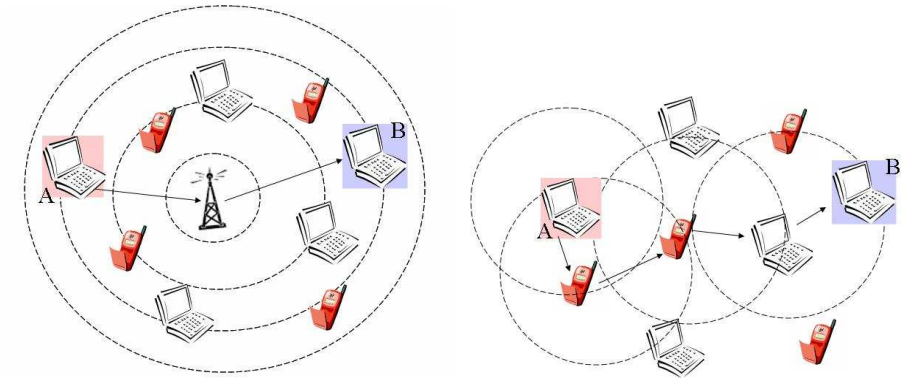
A segurança de redes atrai investimentos pesquisa, pois dela dependem a disponibilidade e confiabilidade dos serviços oferecidos pela rede. As ameaças são inúmeras, variando desde espionagens até ações maliciosas para fraudes bancárias ou, ainda, ataques para indisponibilizar serviços de grandes empresas. Tais problemas provocam prejuízos de bilhões de dólares e atingem empresas e usuários domésticos.

A segurança de redes começou a ser estudada há décadas atrás, mas cada nova tecnologia de rede trás consigo novas características e, conseqüentemente, novos desafios a serem vencidos. Assim, a segurança é um tema atual, em especial para as redes sem fio. Ao contrário das redes cabeadas, nas redes sem fio, qualquer pessoa que esteja no alcance de transmissão da antena pode escutar todos os dados que foram transmitidos. Além disso, nas redes sem fio sem infra-estrutura, como é o caso das redes ad hoc, prover segurança é complexo devido ao roteamento colaborativo. Nessas redes, além de impedir a espionagem, é preciso ainda estimular a cooperação entre os nós e excluir da rede todos os que agem de forma maliciosa, pois apenas um nó malicioso pode impedir o funcionamento da rede [6, 7].

1.1 Redes Ad Hoc

As redes ad hoc móveis, também conhecidas como *Mobile Ad hoc NETworks - MANETs*, são redes sem fio não infra-estruturadas que utilizam roteamento colaborativo. A ausência de infra-estrutura significa que essas redes não possuem pontos de acesso ou estações-base, como acontece, por exemplo, nas redes de celulares. O roteamento colaborativo permite que nós que não estejam no alcance de transmissão de rádio se comuniquem através de múltiplos saltos, através da colaboração de nós intermediários que encaminham a mensagem do nó origem até o nó destino. Outra característica importante das redes ad hoc sem fio é que a topologia da rede pode mudar dinamicamente devido à mobilidade dos nós. Nas Figuras 1.1(a) e 1.1(b), estão representadas, respectivamente, uma rede com ponto de acesso, na qual os nós, para se comunicarem, precisam estar no

alcance de um ponto de acesso, e uma rede ad hoc, na qual os nós comunicam-se através de múltiplos saltos.



(a) Rede infra-estruturada. (b) Rede ad hoc.

Figura 1.1: Exemplos de redes.

As redes ad hoc sem fio possuem como principais vantagens o baixo custo de instalação, a facilidade de configuração e o maior alcance devido aos múltiplos saltos. Entre as aplicações dessas redes estão o uso em conferências, recuperação de desastres, campos de batalhas [8–10], ambientes inóspitos ou hostis e também para provimento de acesso a Internet a baixo custo em áreas carentes e em domicílios [11–13]. Cabe observar que, para uma rede ad hoc prover acesso à Internet, é necessário um *gateway* e protocolos de roteamento específicos que permitam a qualquer nó ser um *gateway*.

A segurança nas redes ad-hoc militares é mandatória, pois tanto a espionagem quanto a perda de dados podem ter conseqüências graves. Nas redes ad hoc não militares, a segurança também é um requisito essencial, pois o comportamento malicioso de um nó pode impedir o funcionamento de toda a rede. Dessa forma, para garantir a disponibilidade da rede, medidas de segurança como o controle de acesso e a monitoração dos nós devem ser tomadas.

Devido a características como o meio de comunicação sem fio, a ausência de infra-estrutura e o roteamento colaborativo em múltiplos saltos, as redes ad hoc se tornam alvos potenciais de diversos tipos de ataques. Assim, a provisão de segurança é um dos seus desafios mais críticos. A utilização do ar como meio de transmissão torna a rede susceptível a diversos ataques, que vão desde uma simples espionagem das mensagens

até interferências com a criação, modificação e destruição das mensagens em trânsito. As redes cabeadas são consideradas mais seguras, pois um atacante tem maior dificuldade para obter 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, bastando que o nó atacado esteja no alcance da transmissão do nó atacante. 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, as facilidades oferecidas pela comunicação sem fio têm como contrapartida a ausência de uma barreira de defesa clara. Assim, todo nó da rede deve estar preparado para lidar direta ou indiretamente com ações maliciosas.

A ausência de infra-estrutura nas redes ad hoc requer colaboração distribuída dos nós da rede para o encaminhamento das mensagens. Por essa razão, nas redes ad hoc, todos os nós também desempenham a função de roteador. No entanto, 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, o que significa que as principais particularidades das redes ad hoc estão na camada de rede. Outro aspecto importante a ser considerado nas redes ad hoc é a ausência de centralização e de infra-estrutura. Não existem dispositivos dedicados a tarefas específicas da rede como, por exemplo, realizar a autenticação ou distribuir endereços dinamicamente. Apesar de a descentralização ter como vantagem a disponibilidade, devido à inexistência de pontos únicos de falha, a ausência de infra-estrutura inviabiliza a aplicação das técnicas convencionais de controle 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.

A mobilidade introduz outros obstáculos importantes à implementação de mecanismos de segurança, devido às constantes alterações na topologia da rede, que podem causar até particionamentos na rede. Outro problema que surge é que, com nós móveis, não há como saber acessar diretamente servidores e nem a divisão clássica de redes e sub-redes se aplica corretamente, pois o nó, ao mudar de posição, poderia sair da área de acesso da sua sub-rede e perder conexão. 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, deve-se ressaltar que as redes ad hoc móveis são, em geral, compostas por dispositivos portáteis, possuindo, assim, restrições de energia, processamento e memória. Com isso, as MANETs estão também sujeitas a diferentes ataques de negação de serviço que visam esgotar os recursos dos nós a fim de prejudicar o funcionamento da rede.

Em suma, as redes ad hoc móveis possuem vulnerabilidades justamente nas suas principais características, como a utilização do ar como meio físico, as alterações dinâmicas da topologia de rede, a cooperação entre os nós, a distribuição de tarefas e a ausência de um ponto central de monitoramento e gerenciamento. Essas vulnerabilidades são inerentes às MANETs, havendo, portanto, a necessidade de criar mecanismos de defesa específicos. É importante notar que a solução desses problemas não resolve algumas das questões clássicas de segurança, como proteção contra vírus ou mal-uso das aplicações, mas apenas visam colocar as redes ad hoc com o mesmo grau de segurança que uma rede cabeada. Assim, são ainda necessários todos os métodos de segurança da camada de aplicação para garantir uma rede segura.

1.2 Motivação

As propostas para prover segurança em redes ad hoc procuram excluir os nós maliciosos ou não-cooperativos da rede. Para prover segurança no roteamento, muitos protocolos propõem a utilização de assinaturas de mensagens para garantir que o conteúdo não foi modificado e para validar a origem da mensagem. Dessa forma, os autores das atitudes maliciosas poderiam ser identificados e excluídos da rede. Esse é o caso dos protocolos de roteamento seguro, tais como o *Secure Optimized Link State Routing protocol* (SOLSR) [14] e o *Secure Ad hoc On demand Distance Vector* (SAODV) [15], que se servem de assinaturas para garantir que nós não autorizados não interfiram na rede. Além das assinaturas, sistemas de detecção de intrusão (SDI) também são utilizados para detectar ações maliciosas e sistemas de confiança são utilizados para realizar a troca de informações, obtidas com o SDI, entre os nós. Novamente se faz necessário garantir que o conteúdo e a origem das mensagens não sejam forjados. Portanto, a segurança em redes

ad hoc depende da existência de métodos eficazes de identificação e autenticação.

A utilização apenas de sistemas de autenticação e monitoração dos nós, no entanto, não é eficiente, pois é necessário garantir que, uma vez excluído, o nó malicioso não possa voltar à rede. Assim, além de identificar, autenticar e monitorar cada um dos nós da rede, é preciso também estabelecer uma política de autorização, responsável por determinar os recursos que cada nó pode acessar. Por meio de uma política de autorização é possível realizar um controle de acesso completo na rede, ou seja, é possível restringir e controlar o acesso de nós à rede. Para obter esse controle, cada entidade tentando obter acesso deve, primeiramente, ser identificada/autenticada, para que os direitos de acesso sejam dados a essa entidade. O controle de acesso também é responsável por impedir o acesso sempre que o direito de uma determinada entidade de acessar a rede for cassado. Portanto, para obter um controle de acesso é necessário identificar, autenticar e monitorar. Nas redes ad hoc, essas tarefas devem ser realizadas de forma distribuída, pois não existe uma entidade centralizadora que possa atuar como administrador. Além disso, esses sistemas distribuídos devem também ser robustos às desconexões, uma vez que as redes ad hoc apresentam freqüentes desconexões. Esses sistemas distribuídos de identificação/autenticação/monitoração são o foco desta tese.

1.3 Objetivo

O objetivo desse trabalho é apresentar um sistema de controle de acesso para redes ad hoc. Assim, as propostas nessa tese visam garantir a identificação, a autenticação, a monitoração e a autorização em redes ad hoc. Portanto, o objetivo é desenvolver um sistema que realize todos os passos necessários para realizar o controle de acesso da rede, como mostrado na Figura 1.2. Primeiramente, é preciso definir uma regra para determinar quais são os usuários que podem acessar a rede, ou seja, qual é o grupo de usuários que tem autorização para utilizar os recursos disponíveis. Em seguida, após definir o grupo de usuários que tem direito de acesso, é preciso dar uma identidade para cada nó. O fornecimento da identidade será importante nas atividades como o monitoramento, além de ser ligada ao material criptográfico. Após a identificação, é preciso registrar e

autenticar os nós. A autenticação pode ser feita através do uso de criptografia simétrica ou assimétrica e permite que os usuários provem a sua identidade ao utilizar a rede. Por fim, é preciso ainda monitorar os nós e excluir da rede todos aqueles que agem de forma maliciosa ou não-cooperativa, redefinindo o grupo de usuários autorizados.

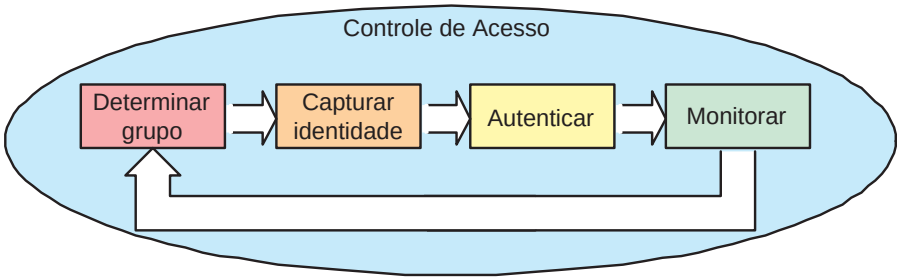


Figura 1.2: Funções do sistema de controle de acesso.

Executando todos os passos do esquema descrito, é possível controlar de acesso na rede, permitindo um ambiente seguro e privado. Nesse trabalho, são apresentadas contribuições para cada um dos passos citados de forma a permitir o controle de acesso, através de protocolos distribuídos que se adequam às características das redes ad hoc. Assim, propõe-se um protocolo para a identificação dos nós, um mecanismo de autoridade certificadora distribuída baseado em cadeia de confiança, um protocolo para distribuição de chaves de grupo e um mecanismo de monitoramento baseado nos mesmos princípios que a autoridade certificadora. Com essas propostas, é possível realizar o controle de acesso sem tomar muitos recursos dos nós e sem emitir uma grande carga de controle.

1.4 Contribuições

Nesse trabalho são apresentadas novas abordagens que tornam o processo de controle de acesso em redes ad hoc mais eficiente e robusto através de sistemas de autoconfiguração de endereços, para identificação dos nós, de autenticação e de monitoração.

A proposta de identificação dos nós permite um controle preciso dos endereços utilizados na rede, evitando duplicidade de endereços e, portanto, garantindo uma identificação única para cada nó. A proposta prevê como contribuições um protocolo de divulgação de endereços e um mecanismo de armazenamento compacto dos endereços utilizados pe-

los nós. O mecanismo de armazenamento compacto das informações proposto procura reduzir a capacidade de memória necessária e também o tamanho das mensagens de controle a serem enviadas. O protocolo de autoconfiguração dos endereços procura reduzir o número de mensagens de controle a serem tocadas.

A proposta de autenticação prevê uma autoridade certificadora distribuída baseada em cadeias de confiança, que registra os endereços dos nós, associando a cada endereço uma chave pública, e emite certificados. A segurança dessa proposta é baseada em testemunhas, que são nós escolhidos por meio de funções *hash* para monitorar um determinado nó. Por meio dessas testemunhas é possível controlar a entrada de nós na rede e punir nós maliciosos com sua exclusão através da revogação do certificado. A contribuição deste trabalho na autenticação é uma forma inovadora de controlar as autorizações e de distribuir os certificados dos nós, permitindo o registro dos usuários de forma distribuída e, ao mesmo tempo, impedindo que os nós maliciosos excluídos da rede retornem. Nas redes convencionais esta função é realizada de forma centralizada por um "administrador".

Por fim, a terceira contribuição apresentada é um protocolo de gerenciamento de chaves de grupo baseado na existência de uma autoridade certificadora. O gerenciamento das chaves de grupo é importante para controlar o acesso de nós às atividades de um determinado grupo. No caso do roteamento seguro ad hoc, o grupo representa todos os nós autorizados à acessar a rede e a chave de grupo é utilizada para identificar as mensagens de controle provenientes dos nós autorizados. O protocolo apresentado nesse trabalho para o gerenciamento de chaves de grupo é projetado para o protocolo SOLSR, muito embora possa ser generalizado para qualquer protocolo, mantendo uma eficiência com relação à energia gasta por cada nó durante a distribuição.

1.5 Organização da Tese

Esta tese está organizada da seguinte forma. O Capítulo 2 trata a identificação dos nós em uma rede ad hoc, descrevendo o estado da arte e uma proposta para distribuição eficiente de endereços em termos de tráfego de controle. O Capítulo 3 apresenta uma proposta para autenticar, certificar e monitorar os nós que já possuem um endereço IP, enquanto o

Capítulo 4 apresenta um protocolo para distribuição de chaves de grupo. Finalmente, a o Capítulo 5 apresenta as conclusões da tese.

Capítulo 2

Autoconfiguração de Endereços

IDEALMENTE , o endereçamento dos nós deve ser compatível com o *Internet Protocol* (IP), que é o protocolo utilizado pela maioria das aplicações e disponível em todos os sistemas operacionais. Dessa forma, cada nó deve ter um endereço IP único, contendo um prefixo de rede e um sufixo de máquina. A alocação automática é necessária para redes criadas para eventos, conferências e recuperação de desastres, nas quais não é possível fazer uma alocação manual de endereços, porque não se conhece, *a priori*, informações sobre os dispositivos que irão utilizar a rede. Além disso, a alocação dinâmica de endereços, que difere da automática por realizar o controle dos endereços que não estão sendo mais utilizados, é fundamental para redes onde o espaço de endereçamento disponível é menor que o número de dispositivos que podem ser interconectados. No contexto de controle de acesso, o IP também serve como a identificação a ser associada à chave pública durante o registro do nó.

Nas redes com infra-estrutura, como as redes cabeadas e as redes em malha, a alocação automática/dinâmica de endereços pode ser feita de forma centralizada por um servidor responsável por controlar os endereços disponíveis. Nas redes sem infra-estrutura, como é o caso das redes ad hoc, nenhum nó da rede está sempre disponível a todos os outros nós para atuar como servidor de alocação de endereços. Devido à mobilidade e também à má qualidade das condições de enlaces sem fio, segmentações da rede ad hoc são comuns. Assim, se torna necessário um protocolo para a autoconfiguração dos endereços que funcione de forma robusta e distribuída. Esse protocolo também deve consumir

poucos recursos, pois muitos dispositivos sem fio, como sensores, assistentes digitais pessoais (*Personal Digital Assistant* - PDA) e computadores portáteis, possuem restrições de processamento, memória, armazenamento e bateria.

Nesse capítulo é proposto o protocolo de AUtoconfiguração de endereços baseado em FIltros para Redes Ad hoc (AUFIRA). No AUFIRA, cada nó controla, de forma distribuída, os endereços disponíveis através do uso de vetores (filtros) que possuem a capacidade de representar informações de forma compacta. A utilização dos filtros no AUFIRA permite também observar a união de partições na rede, situação comum em redes ad hoc. O protocolo proposto também reduz o número de mensagens para resolução de colisões de endereços, que ocorrem quando se aloca um endereço já em uso. Os resultados das simulações mostram que, mesmo com perdas de pacote, o protocolo proposto consegue distribuir os endereços e resolver as colisões, pois mensagens de controle perdidas são identificadas no filtro, evitando que colisões de endereço não sejam detectadas. Além disso, o protocolo proposto atende às restrições de economia de energia, processamento, memória e armazenamento, comuns em dispositivos portáteis.

2.1 Autoconfiguração de Endereços

A utilização de protocolos de alocação de endereços baseados no paradigma cliente/servidor convencionalmente empregados em redes infra-estruturadas, como o *Dynamic Host Configuration Protocol* (DHCP) [16], não é viável em redes sem infra-estrutura devido à ausência de nós com características específicas de servidores e às freqüentes desconexões dos nós. Uma proposta para solucionar esse problema foi a adoção de endereçamento baseado em *hardware*, como sugerido na autoconfiguração de endereços sem estados do IPv6 [17]. A proposta é utilizar um endereço formado por um prefixo conhecido de rede e um sufixo baseado no endereço MAC do dispositivo. Um fator restritivo para essa proposta é que, com esse tipo de configuração, o espaço de endereçamento disponível tem que ser maior ou igual ao número de nós.

Outra questão relevante do uso do endereço MAC para determinação do endereço IP é a facilidade de rastreamento de um dispositivo, que pode gerar um problema de privaci-

dade, uma vez que com simples monitoramento do IP nas conexões é possível conhecer a movimentação do usuário entre os diferentes locais nos quais ele esteve com o dispositivo [18]. Além disso, sabe-se que os protocolos TCP/IP devem funcionar independentes da implementação da camada de enlace. De fato, nem todos os dispositivos em uma rede ad hoc utilizam placas de interface de rede com endereços MAC únicos distribuídos pelo IEEE com 48 bits, além de existirem casos conhecidos de placas do mesmo fabricante com o endereço MAC repetido. Por fim, é possível trocar manualmente o endereço MAC. Portanto, a alocação dinâmica de endereços realizada de forma não centralizada que garanta a unicidade do endereço alocado e que evite o rastreamento por questões de privacidade não é um problema trivial.

Um dos principais desafios da alocação distribuída de endereços é a garantia da unicidade do endereço. Supondo uma escolha aleatória, a probabilidade de colisões de endereço, $P(E_c)$, é determinada pelo espaço de endereçamento disponível e pelo número de nós disputando esses endereços. Essa probabilidade de colisão pode ser calculada com analogia ao conhecido paradoxo do aniversário [19] e pode ser estimada pela inequação dada por

$$P(E_c) > 1 - e^{\frac{-(n(n-1))}{2r}}, \tag{2.1}$$

onde n representa o número de nós disputando endereços e r representa o espaço de endereços disponível. Suponha uma rede utilizando endereços IP versão 4 (IPv4) com endereços classe C. Pela Figura 2.1, que representa a Equação 2.1, é possível observar que, com um espaço de endereçamento de 256 endereços e com 42 destes endereços ocupados, o que representa cerca de um sexto do total, a probabilidade de acontecer colisões de endereço é maior que $0,96$, ou seja, é quase certo que haverá colisão de endereço. Assim, fica claro que é necessária a utilização de um protocolo de alocação de endereços que resolva as colisões.

Perkins *et al.* propuseram um protocolo totalmente distribuído para a detecção de endereços duplicados (*Duplicate Address Detection* - DAD) [20]. Este protocolo é baseado em mensagens de requisição de endereço (*Address Request* - AREQ) e resposta (*Address Reply* - AREP). Assim, cada nó, ao entrar na rede, escolhe um endereço e inunda a rede

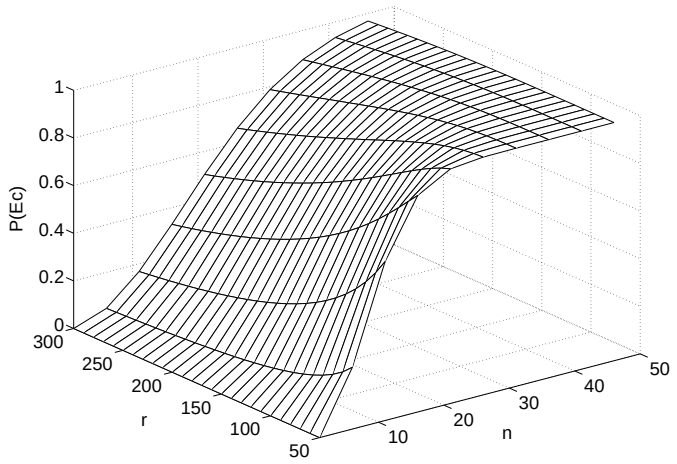


Figura 2.1: Probabilidade de colisão de endereços.

com AREQs por uma quantidade determinada de vezes. Se algum nó da rede já possuir um endereço igual, ele deve enviar uma mensagem de AREP para o novo nó que está realizando a atribuição de endereço. Na ausência de respostas, o novo nó assume que nenhum nó da rede possui o endereço atribuído e, portanto, o utiliza. Caso contrário, ao receber um pacote de resposta, o nó escolhe um novo endereço e repete o processo de inundação. A proposta de Perkins *et al.*, no entanto, não trata dos casos de união de partição, o que impede o uso do protocolo em redes com baixa conectividade.

Baseado na detecção de endereços duplicados (DAD), outros protocolos foram propostos, inserindo mensagens de HELLO e identificadores de rede para detectar partições na rede. Os identificadores de rede são números escolhidos aleatoriamente para identificar grupos de nós e devem ser trocados sempre que uma partição se formar e quando partições se unirem. Os identificadores de rede são importantes porque, quando a rede sofre partições, o prefixo de rede não deve ser modificado, já que os nós continuam pertencendo ao mesmo domínio, embora estejam temporariamente desconectados. Assim, diferentes valores de identificadores de redes servem para identificar as partições formadas. Quando um nó recebe um HELLO com um identificador de rede diferente do seu próprio identificador, é observada a existência de duas partições. Fan e Subramani propuseram um protocolo baseado no DAD que detecta união de partições sempre que um nó receber um HELLO com um identificador de rede diferente do seu ou ainda quando

ocorrerem mudanças no conjunto de vizinhos [21]. Fazio *et al.* propuseram um protocolo também baseado em identificadores de rede, mas que funciona de forma reativa [22]. Ao invés de enviar AREQs e AREPs em todas as entradas de nós ou uniões de partições, o que permitiria acabar com as colisões no primeiro momento no qual elas possam ser detectadas, o protocolo faz a identificação de colisões apenas quando existe a necessidade de troca de dados. Dessa forma, a verificação de um determinado endereço só será realizada se algum nó tentar enviar dados para aquele endereço. Embora reduza o número de mensagens trocadas, esse protocolo, por ser reativo, acarreta um atraso na transmissão dos dados. Além disso, o mecanismo usado se aplica apenas a protocolos de roteamento reativos, pois, nos protocolos pró-ativos, como as rotas são formadas em avanço, colisões de endereço causam a formação de rotas erradas.

Algumas propostas são baseadas na alocação de conjuntos de endereços para cada nó. O protocolo *Dynamic Address assignment Protocol in mobile ad-hoc networks* (DAP) subdivide o seu conjunto de endereços disponíveis à medida que novos nós entram na rede [23]. Quando um nó fica sem endereços extras para a alocação de novos nós, é feito um pedido que gera uma realocação dos endereços disponíveis. O inconveniente dessa proposta é a realocação de endereços, pois é difícil detectar que um determinado endereço não está mais sendo utilizado e pode ser disponibilizado para os novos nós. Outra proposta semelhante é o *Prophet*, que, para endereçar os nós, utiliza um gerador de números aleatórios com alta entropia, ou seja, uma baixa probabilidade de repetição do mesmo valor [24]. O primeiro nó da rede, chamado de nó profeta, escolhe a semente da seqüência de números aleatórios e atribui endereços para os novos nós que o contatam. Esses novos nós passam a distribuir endereços a partir de pontos diferentes da seqüência de números aleatórios, formando uma árvore de distribuição. O protocolo gera poucas mensagens, mas, em compensação, não soluciona completamente a união de partições e também não trata a realocação de endereços.

Outro tipo de abordagem de autoconfiguração de endereços é a formação de células de alocação de endereço baseadas na localização dos nós [25]. Existem, ainda, propostas que utilizam informações de roteamento para detectar colisões [26, 27]. Essas propostas, no entanto, acarretam atrasos significativos na detecção de colisões ou são soluções específicas para determinados protocolos.

2.2 O Protocolo AUFIRA

O objetivo do protocolo proposto, denominado AUFIRA, é a autoconfiguração de endereços de forma dinâmica, identificando e resolvendo as colisões de endereço de forma eficiente. O AUFIRA é apropriado para redes sem infra-estrutura nas quais os nós conhecem o prefixo da rede e, assim, necessitam apenas determinar um sufixo de máquina para possuir um endereço completo. O AUFIRA também visa gerar uma baixa carga de mensagens de controle e manter o conjunto de endereços utilizados sem colisões mesmo quando novos nós entram na rede ou quando ocorrem uniões de partições. Para reduzir o número de mensagens a serem trocadas e a memória necessária em cada nó, o protocolo proposto utiliza filtros de endereço, que são estruturas capazes de armazenar informações de forma compacta. Esses filtros devem ter capacidade de comprimir os dados inseridos neles. Embora muitas soluções possam ser dadas na área de compressão, neste artigo são apresentadas duas estruturas de dados que podem funcionar como filtro de endereços e que se adequam a cenários diferentes. A primeira estrutura é o Filtro de Bloom [28], que é baseado em funções *hash*, enquanto que a segunda estrutura, chamada de Filtro Simplificado, comprime os dados baseado na seqüencialidade dos endereços.

2.2.1 Filtros de Bloom

Um Filtro de Bloom é um vetor formado por m bits que representa um determinado conjunto $A = \{a_1, a_2, a_3, \ldots, a_n\}$ formado por n elementos. Devido à sua alta capacidade de compressão, os Filtros de Bloom vêm sendo utilizados em diversos tipos de aplicações, como rastreamento de pacotes [29] e *web cache* [28]. Para gerar o filtro, k funções *hash* independentes (h_1, h_2, ..., h_k) com um alcance m são utilizadas. Inicialmente, o vetor de bits que forma o filtro deve ser zerado e, em seguida, cada um dos elementos $a_i \in A$ deve ser aplicado a cada uma das k funções *hash*. O resultado de cada função *hash* representa uma posição do vetor que deve ser trocada de 0 para 1, como mostrado na Figura 2.2(a). Para verificar se um elemento a_j pertence ao conjunto A de elementos inseridos no filtro, deve-se aplicar as k funções *hash* ao elemento e, se alguma das posições $h_1(a_j)$, $h_2(a_j)$, ..., $h_k(a_j)$ corresponder a um bit em 0 no filtro, o elemento

certamente não pertence ao conjunto A. Caso todos os bits correspondam a bits em 1 no filtro, assume-se o elemento pertence ao conjunto A. No entanto, existe uma probabilidade de ocorrerem falso-positivos, ou seja, de que elementos que não foram inseridos anteriormente apareçam como presentes após uma verificação. A partir da expressão da probabilidade de um bit permanecer em 0 após a inserção de n elementos (P_0), dada por

$$P_0 = \left(1 - \frac{1}{m}\right)^{kn}, \tag{2.2}$$

é possível concluir que a probabilidade de falso-positivo, P_{fp} é dada por

$$P_{fp} = (1 - P_0)^k = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k. \tag{2.3}$$

Pode-se observar que quanto menor o número de elementos, n, do conjunto A e quanto maior o tamanho do filtro, m, menor a probabilidade de falso-positivos. Além disso, por essa expressão, é possível obter o valor de k que minimiza a probabilidade de falsos positivos, através da derivada em relação à k da Equação 2.3, que é dada por

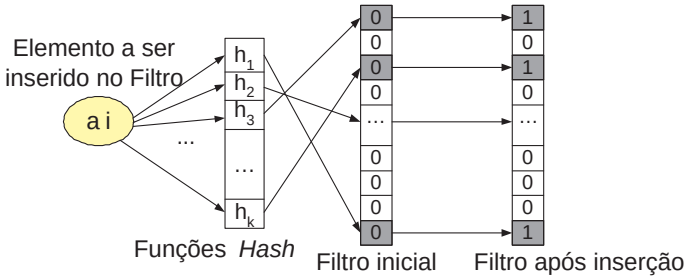
$$\frac{d(P_{fp})}{dk} = \left[ln\left(1 - e^{-\frac{kn}{m}}\right) + \frac{(kn)e^{\frac{-kn}{m}}}{m(1 - e^{\frac{-kn}{m}})}\right]\left(1 - e^{\frac{-kn}{m}}\right)^k \tag{2.4}$$

Igualando a derivada a zero, deduz-se que o valor de k para minimizar a probabilidade de falso-positivos é dado por

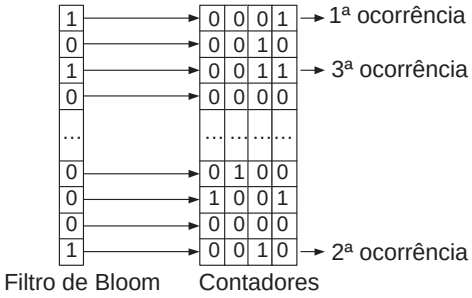
$$k = \frac{m \cdot ln(2)}{n}. \tag{2.5}$$

O valor de k deve ser escolhido como um valor inteiro levando em consideração o esforço computacional necessário em cada inserção ou verificação no filtro, o tamanho máximo que o filtro pode ocupar e a probabilidade de falso-positivos máxima determinada pela aplicação. Dessa forma, para uma probabilidade de falso-positivos de aproximadamente 5%, utilizando o valor de k ideal, o filtro precisa ter um tamanho mínimo dado por $m/n = 6$.

Em Filtros de Bloom tradicionais, não é possível retirar elementos do filtro, pois, se os bits relativos a um elemento fossem zerados, poderiam ocorrer falso-negativos. Isso



(a) Inserção de elementos no filtro.



(b) Contadores do filtro ($i = 16$).

Figura 2.2: Filtro de Bloom.

ocorre porque os bits zerados podem ter sido mudados para 1 por mais que um elemento, e ao zerá-los, todos esses outros elementos também seriam excluídos do filtro. Assim, para aplicações onde é necessário retirar elementos do filtro é utilizado um contador para cada bit do filtro, como ilustrado na Figura 2.2(b). Dessa forma, sempre que um elemento a_j for incluído, os contadores dos bits $h_1(a_j)$, $h_2(a_j)$, ..., $h_k(a_j)$ são incrementados, e, quando um elemento é excluído, os contadores são decrementados. Assim, é possível saber o número de vezes que um determinado bit foi selecionado por alguma das funções *hash*. Além disso, somando os valores de cada contador e dividindo pelo número de funções *hash*, é possível determinar o número exato de elementos no filtro. Em [28], os autores mostram que a probabilidade de um contador ser maior ou igual a i ($P(c \geq i)$) é dada por

$$P(c \geq i) \leq m\left(\frac{e \cdot n \cdot k}{i \cdot m}\right)^{i}. \tag{2.6}$$

Ao se usar contadores é importante que se garanta que não ocorra transbordo do contador, pois isto implicaria na geração de falso-negativos. Assim, supondo contadores com quatro bits ($i = 16$), $m/n = 6$ e $k = ln(2) \cdot m/n \approx 4$, a probabilidade de transbordo do contador

é menor que $6,6 \cdot 10^{-13}$, o que pode ser considerado desprezível.

A idéia chave da utilização do filtro no protocolo de alocação de endereços proposto é a representação de forma compacta de todos os endereços já atribuídos. Assim, sempre que um novo nó entra na rede, ele solicita a um vizinho qualquer o seu filtro, escolhe aleatoriamente um endereço e verifica se o endereço já foi alocado. Caso o endereço já tenha sido alocado, o teste de pertinência no filtro é positivo e o nó repete o processo. Caso o teste de pertinência seja negativo é certo que este endereço não foi inserido no filtro e, portanto, não foi ainda alocado a nenhum nó. Neste caso, ele aloca este endereço para si, insere este novo endereço no filtro e difunde o novo endereço alocado para todos os nós. Com isso, o procedimento proposto realiza apenas uma inundação ao contrário dos protocolos baseados no DAD [20] que são obrigados a fazer novas inundações toda vez que recebem uma mensagem informando que o endereço atribuído já está em uso, ou seja, ocorre uma colisão.

A utilização de Filtros de Bloom também simplifica a detecção de partições. Uma união de partições é detectada sempre que nós vizinhos possuem filtros de Bloom diferentes, pois filtros diferentes indicam conjuntos de nós diferentes. Assim, se os nós puderem comparar os seus filtros e eles forem diferentes, isto significa que cada um possui um conjunto de nós diferente, o que é indicador de existência de partição. Com os Filtros de Bloom, é possível, ainda, estimar qual é a menor partição através da observação da quantidade de nós inserida em cada filtro. Com base na informação sobre o tamanho das partições, apenas os nós da partição menor cujos endereços estiverem presentes no filtro da partição maior deverão escolher um novo endereço não utilizado como sua nova identificação. Portanto, a proposta de utilização de filtros de Bloom melhora a eficiência da alocação de endereços na rede e reduz o volume de tráfego de controle.

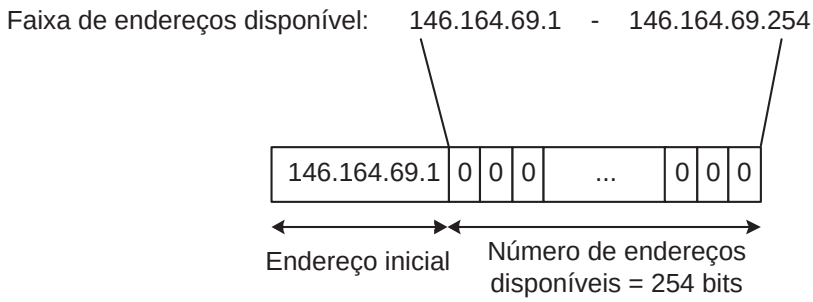
A idéia chave do uso do Filtro de Bloom é possibilitar que cada nó possua a informação de quais endereços já foram usados e, assim, evite colisões de endereços. Não havendo colisões, uma única inundação informando o novo endereço alocado é suficiente. A informação do conjunto de elementos que pertencem ao Filtro de Bloom também facilita identificação de união de sub-redes segmentadas. A função do Filtro de Bloom é "representar de forma compacta" o conjunto dos nós e permitir um simples processo de

verificação de pertinência. O Filtro de Bloom permite o teste de pertinência mas, devido à compactação, não permite a identificação de quais elementos estão sendo representados pelo filtro. Para que seja possível o procedimento de alocação dinâmica, os endereços dos nós que saem da rede devem poder ser realocados. Assim, extensão de Filtros de Bloom com contadores foi usada para permitir a "retirada" de elementos do Filtro. Portanto, a proposta de utilização de filtros de Bloom melhora a eficiência da alocação de endereços na rede, reduz o volume de tráfego de controle e facilita o processo de identificação de uniões.

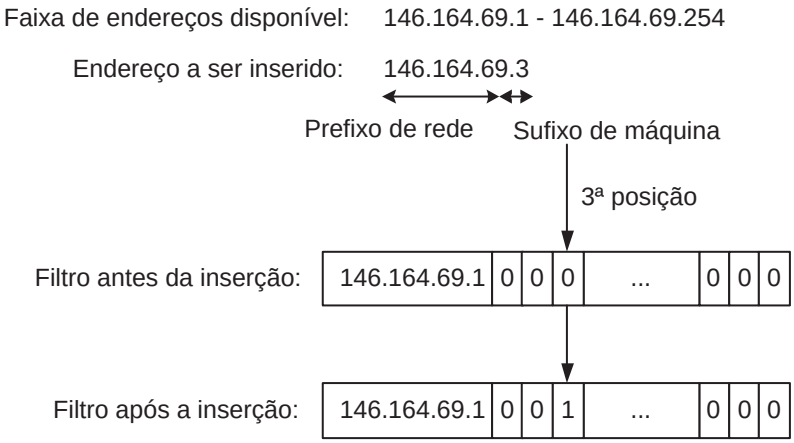
2.2.2 O Filtro Simplificado

Esta seção propõe o Filtro Simplificado, que é uma estrutura de compressão para armazenar endereços e que se baseia no fato do espaço de endereçamento ser seqüencial. Assim, nessa estrutura proposta, cada endereço é representado pelo prefixo de rede e por um bit. Assim, em uma classe de endereços tipo C (254 endereços de máquinas) utilizam-se 254 bits para representar as máquinas. Portanto, a posição do bit corresponde ao endereço seqüencial, como mostrado na Figura 2.3(a). A inserção de elementos no Filtro Simplificado está ilustrada na Figura 2.3(b).

O Filtro Simplificado mantém as mesmas características que o Filtro de Bloom com relação ao protocolo de autoconfiguração de endereços, uma vez que contém a representação dos endereços que estão sendo usados. Assim como no Filtro de Bloom, a estrutura do Filtro Simplificado também permite identificar o número de nós ativos na rede, além de simplificar a detecção de partições e reduzir o número de mensagens trocadas para a solução de colisões. Uma vantagem da estrutura do Filtro Simplificado com relação ao Filtro de Bloom é que os dados não ficam escondidos no Filtro, ou seja, é mais simples verificar a presença de um nó e retirar os nós ausentes do Filtro. Além disso, a probabilidade de ocorrer falso-positivos ou falso-negativos no Filtro Simplificado é sempre zero.



(a) Estrutura do Filtro Simplificado.



(b) Inserção de elementos no Filtro Simplificado.

Figura 2.3: Filtro Simplificado.

2.2.3 Escolha de Filtros

Como mencionado anteriormente, a probabilidade de falso-negativos no Filtro de Bloom é nula, ou seja não existe a possibilidade de um teste de pertinência de endereço que foi inserido no filtro resultar negativo. No entanto, existe a possibilidade de falso positivos, ou seja, o teste de pertinência pode indicar que um endereço já foi alocado sem que este endereço tenha sido realmente alocado. É importante observar que os falsos positivos não acrescentam sobrecarga de mensagens devido a inundações, pois o nó deve repetir o processo até que o teste de pertinência do Filtro de Bloom indique que o endereço não foi ainda alocado. Assim, o falso positivo apenas faz o nó repetir o processo de escolha do novo endereço e atrasa o processo de alocação. O número de vezes que é necessário repetir a escolha do endereço e, conseqüentemente, o processamento gasto aumentam a

medida que o número de endereços inseridos no filtro cresce. Assim, o tamanho do Filtro de Bloom deve ser determinado de forma a reduzir a probabilidade de falso-positivos, mas com o compromisso de ser compacto, mantendo m não muito grande, e não aumentando muito o processamento com o uso de muitas funções *hash*.

O tamanho do Filtro de Bloom deve ser determinado de forma a reduzir a probabilidade de falso-positivos, conforme mostrado na Seção 2.2.1. Para manter a probabilidade de erro em $0,05$, a relação entre o número de bits no filtro (m) e o número de entradas (n) deve ser $m/n = 6$ e o número de funções *hash* deve ser igual a 4. Uma vez que contadores estão sendo utilizados no Filtro para permitir a saída de nós, o tamanho do Filtro de Bloom deve ser dado por

$$m = 6 \cdot n \cdot 4. \tag{2.7}$$

O Filtro de Bloom não possui falso-negativos, mas possui falso-positivos. Além disso, a identificação de um endereço que pertença ao conjunto de endereços alocados não é possível, o que dificulta o processo de identificação e exclusão de um nó. O tamanho do Filtro de Bloom pode ser fixo e pequeno, mas à medida que aumenta o número de endereços alocados representados no filtro aumenta a probabilidade de falso-positivos. Para manter a mesma probabilidade de falso-positivo, o tamanho do Filtro de Bloom deve aumentar com o número de endereços representados pelo filtro. Cabe observar que o número de entradas, n, no Filtro de Bloom corresponde ao número médio de nós ativos na rede. De fato, a probabilidade de falso-positivo no Filtro de Bloom não é dada pelo espaço de endereçamento, mas pelo número de entradas inseridas no filtro.

Diferentemente do Filtro de Bloom, o Filtro Simplificado é totalmente determinístico não ocasionando falso-positivos nem falso-negativos. No Filtro Simplificado, é fácil e imediato identificar um endereço e seu tamanho depende diretamente do espaço de endereçamento. Assim, o total de bits no Filtro Simplificado é dado por

$$m = T_e + r, \tag{2.8}$$

onde T_e representa o tamanho em bits do endereço e r representa o número de endereços

no espaço de endereços.

As Figuras 2.4 e 2.5 representam o tamanho dos Filtros de Bloom e Simplificado de acordo com tamanho do espaço de endereços e com o número médio de endereços ocupados, supondo o uso de endereços IPv4 (32 bits de endereço) e que a probabilidade de falso-positivos no Filtro de Bloom é mantida fixa em $0,056$, ou seja, mantendo a relação $m/n = 6$ e fazendo $k = 4$. Por esses gráficos é possível observar que o Filtro de Bloom representa vantagem quando o espaço de endereços é muito grande, mas o número de endereços ocupados é pequeno. Como exemplo, se o número médio de endereços utilizados for 50 e o espaço de endereços for três vezes maior ($r = 150$), a lista de endereços sem compressão ocuparia 200 octetos, enquanto que o Filtro de Bloom ocuparia 88 octetos e o Filtro Simplificado, 23 octetos. Por outro lado, se o espaço de endereços for 20 vezes maior que o número médio de endereços utilizados ($r = 1000$), o Filtro de Bloom ocuparia os mesmos 88 octetos, enquanto que o Filtro Simplificado ocuparia 129 octetos. Portanto, para obter o melhor desempenho, a escolha do Filtro deve ser feita de acordo com o ambiente da rede.

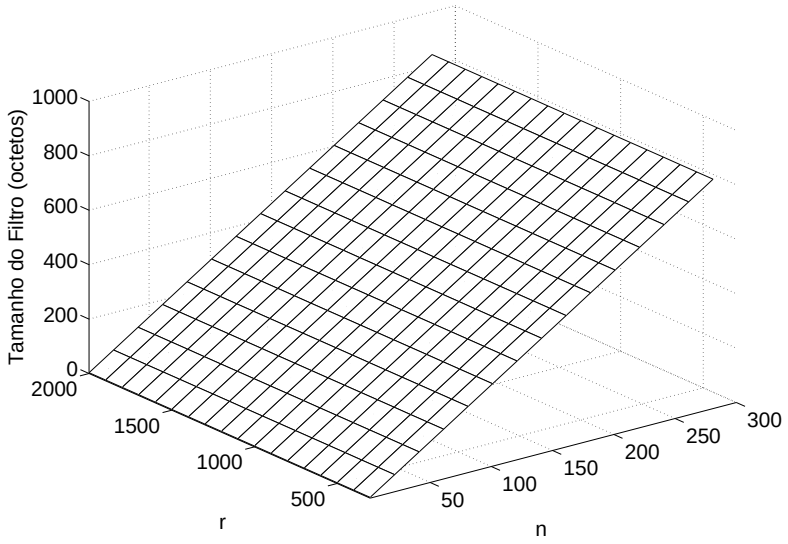


Figura 2.4: Tamanho do Filtro de Bloom.

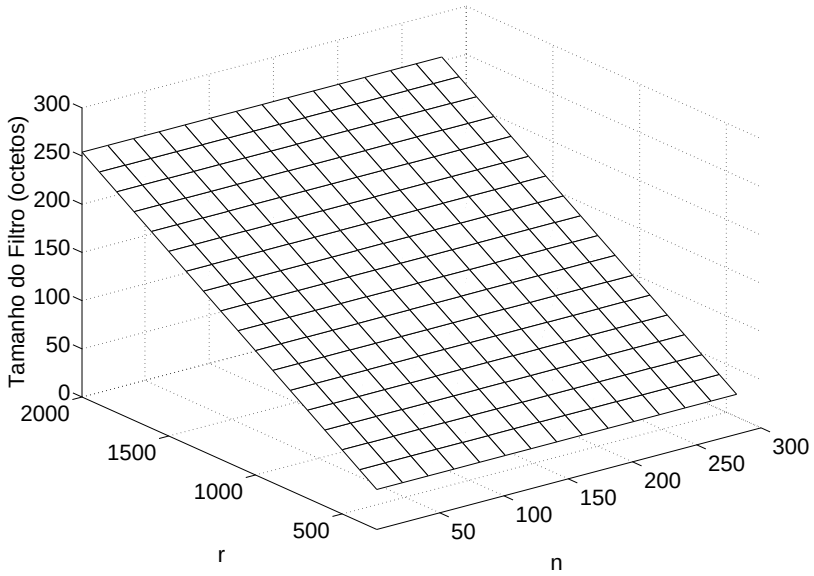


Figura 2.5: Tamanho do Filtro Simplificado.

2.2.4 Descrição do Protocolo AUFIRA

O protocolo AUFIRA pode ser modelado através de uma máquina de estados, como mostrado na Figura 2.6. Primeiramente, o nó deve observar a rede para identificar se está na inicialização ou se a rede já está formada. Após obter o seu endereço, o nó passa para o estado de funcionamento da rede, no qual irá tratar a entrada e saída de outros nós e a formação de partições. Ocasionalmente, se o filtro estiver cheio, o nó pode voltar a fase de inicialização, para excluir nós ausentes que não notificaram sua saída.

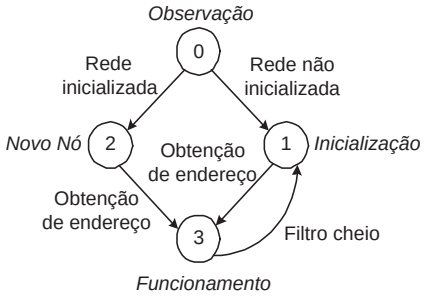


Figura 2.6: Máquina de estados do AUFIRA.

Inicialização da rede

O processo de inicialização da rede refere-se a como o conjunto inicial de nós consegue obter os seus endereços. A inicialização da rede pode ocorrer gradativamente, com longos intervalos entre a entrada de um nó e o seguinte, ou todos os nós podem entrar simultaneamente. Os protocolos de alocação de endereços devem funcionar independentemente da forma como os nós são ativados, embora a maioria dos protocolos suponha que exista um intervalo grande entre a entrada do primeiro nó e a entrada do segundo nó. Um exemplo é o protocolo de Fan e Subramani [21], no qual o primeiro nó, após identificar que está sozinho, escolhe um identificador de rede e começa a emitir mensagens de HELLO. Os próximos nós a entrar na rede são tratados como um nó novo pelo primeiro nó da rede.

O AUFIRA trata tanto a entrada gradativa como a simultânea de nós, utilizando mensagens de HELLO e *Address Request* (AREQ), como representado na máquina de estados da Figura 2.7. Ao entrar na rede, o nó observa o meio por um período T_O. Caso não escute HELLOs de nós que já tenham obtido um filtro, o nó escolhe um endereço aleatoriamente, respeitando os bits do prefixo de rede e passa para a fase de inicialização da rede. Nessa fase, o nó anuncia o seu endereço através da inundação de AREQs por um número N_R de vezes e permanece neste estado por um período de tempo esperando por outros anúncios. Após um período T_I sem escutar AREQs, o nó deixa a fase de inicialização e insere no seu filtro todos os endereços recebidos em AREQs. A partir daí, o nó passa a emitir HELLOs indicando que já possui um filtro. Se durante a fase de inicialização o nó receber um endereço igual ao seu em um AREQ, o nó deve esperar por um período T_T, escolher um novo endereço e enviar novamente os AREQs. O período T_T é importante para permitir que o nó receba mais AREQs informando outros endereços. Assim, mais colisões são evitadas, pois o nó terá um conhecimento mais amplo dos endereços que já foram alocados.

A Figura 2.8 mostra as mensagens de AREQ e HELLO do AUFIRA. O bit *I* das mensagens de AREQ e HELLO indica se o nó está ou não na fase de inicialização da rede. Assim, se um nó A recebe um HELLO com o bit *I* em 0 de um nó B, o nó A sabe que o nó B ainda está na inicialização e que não possui filtro. Esse campo nas duas mensagens é importante para que os novos nós saibam se eles devem passar para a fase

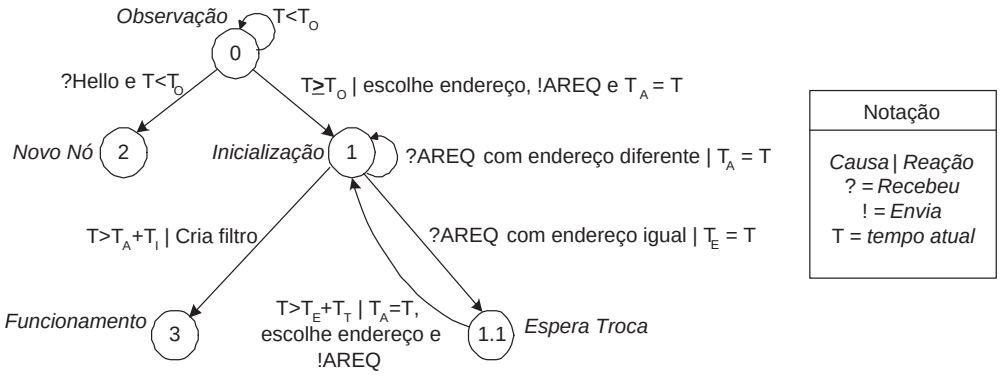
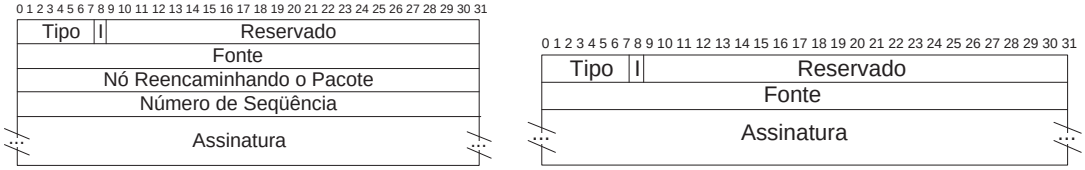


Figura 2.7: Máquina de estados da inicialização do AUFIRA.



(a) Formato da mensagem AREQ.

(b) Formato da mensagem HELLO.

Figura 2.8: Formato das mensagens da fase de inicialização do AUFIRA.

de inicialização da rede ou se devem se comportar como novos nós na rede. O campo assinatura contém o *hash* do filtro, para identificar o filtro atual.

Entrada de Novos Nós

Após a inicialização, os nós começam a emitir HELLOs com o bit *I* em 1, para indicar que já possuem um filtro. No AUFIRA, qualquer nó que já possua um filtro pode receber um pedido de entrada de um novo nó.

A Figura 2.9 mostra a máquina de estados para a entrada de novos nós. Quando um novo nó fica ativo, ele deve escutar o meio durante o período T_O. Como a rede já está formada, o novo nó escutará algum HELLO com o bit *I* em 1 e escolherá o primeiro nó escutado para lhe transmitir o filtro da rede. Para obter o filtro, o novo nó envia uma mensagem Requisição, representada na Figura 2.10(a), para o nó "escutado". Ao receber uma Requisição, o nó observa o bit *I*, para identificar se o pedido contido na mensagem veio de um novo nó. Caso confirme que é um processo de entrada de novos nós, o nó envia como resposta à Requisição uma nova mensagem Requisição com o bit *R* igual a 1. Ao

receber a Requisição em resposta, o novo nó deve guardar o filtro recebido e escolher um endereço que não esteja sendo utilizado. Em seguida, o novo nó envia um AREQ para a rede anunciando que um novo endereço foi alocado. Cada nó, ao receber essa mensagem, insere o endereço indicado no filtro e atualiza a assinatura de seu filtro.

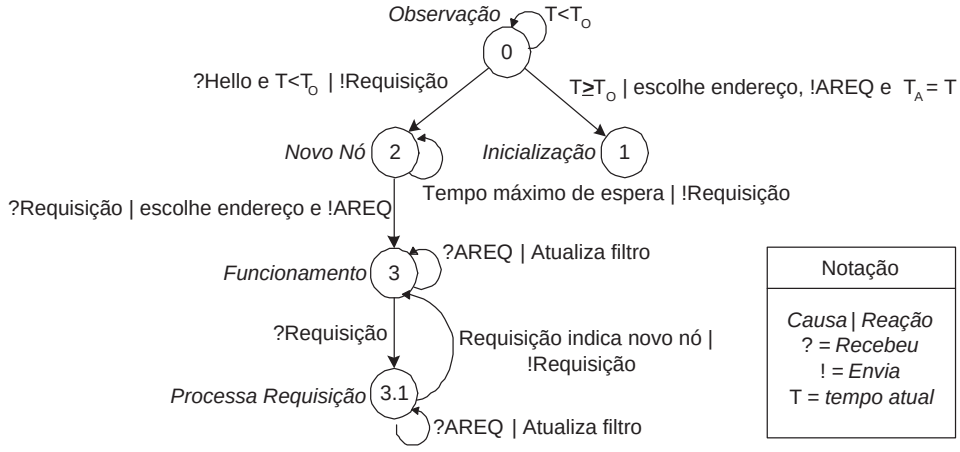
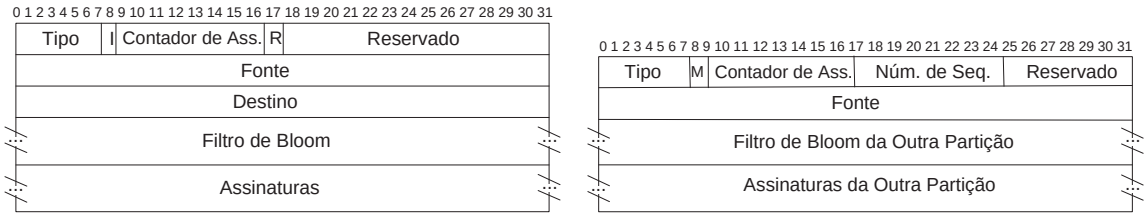


Figura 2.9: Máquina de estados da entrada de novos nós no AUFIRA.



(a) Formato da mensagem Requisição.

(b) Formato da mensagem Partição.

Figura 2.10: Formato de mensagens do protocolo AUFIRA.

A mensagem Requisição traz em seu corpo, além do filtro que está sendo utilizado na assinatura do HELLO, todas as assinaturas de filtros de Bloom válidas atualmente para o nó emissor da Requisição. Essas assinaturas extras de filtros são guardadas por pouco tempo e devem ser mantidas para evitar a falsa detecção de união de partições durante o processo de entrada de novos nós ou de uniões de partições. Como exemplo, suponha a situação demonstrada pela Figura 2.11(a). Nesta figura, o nó A recebeu um AREQ anunciando que o nó C entrou na rede em t_1. O nó A deve encaminhar essa mensagem para o nó B. No entanto, antes de reencaminhar a mensagem, em t_2, o nó A atualiza a sua assinatura de filtro e envia um HELLO. O nó B, ao receber o HELLO de A, irá verificar que a assinatura do HELLO difere da sua própria assinatura, indicando um processo de

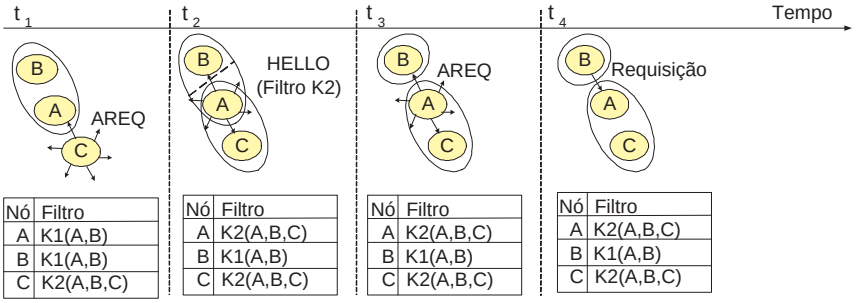
união de partição. Assim, em t_4, um falso processo seria disparado, implicando em uma emissão desnecessária de mensagens.

Na Figura 2.11(b), o nó A, ao invés de atualizar a assinatura ao receber o AREQ de C, guarda a assinatura do filtro K1 e insere em K1 o endereço de C. Dessa forma, o nó B não detecta partição e tem tempo para receber o AREQ de C. Portanto, o falso processo de partição é evitado. Para completar a transição de filtros, o nó A deve esperar um tempo T_F para deixar de usar a assinatura de K1 nos HELLOs e passar a utilizar a assinatura de K2, que é o filtro mais novo. O tempo T_F deve ser grande o suficiente para garantir que todos os vizinhos de A já receberam o AREQ e já criaram um próximo filtro igual a K2. Desta forma, após o nó A fazer a troca da assinatura de K1 pela assinatura de K2, o nó B não detectará partição, pois identificará que o filtro usado no HELLO de A, K2, está presente na sua memória como o seu próximo filtro. Da mesma forma, após o nó A trocar a sua assinatura para K2, ele deve guardar a assinatura de K1 novamente por um tempo T_{b1}. Assim, o nó A poderá reconhecer que o HELLO de B também não é uma partição até que o nó B realize a troca para a assinatura do filtro mais atualizado, K2. Cabe observar que cada nó guarda apenas o filtro mais atualizado. Assim, por exemplo, em t_4, o nó A guarda o filtro K2 e apenas a assinatura de K1, para evitar ocupar muitos recursos dos nós.

Detecção de Colisões em Uniões de Partições

Nós que estão em partições diferentes selecionam endereços baseados apenas no conjunto de endereços alocados na sua partição. Portanto, nós em diferentes partições podem selecionar o mesmo endereço, gerando colisões durante a união de partições. No AUFIRA, a detecção das partições é feita através da assinatura nos HELLOs. Ao receber um HELLO, o nó deve verificar se a assinatura presente na mensagem corresponde à assinatura do seu filtro ou a alguma outra assinatura de filtro guardada. Caso seja diferente, significa que aquele nó possui um conjunto de endereços utilizados diferente e, assim, um processo de união de partição deve ser iniciado.

A máquina de estados referente à união de partições está na Figura 2.12. Quando dois nós detectam a existência de uma partição através da mensagem de HELLO, eles devem



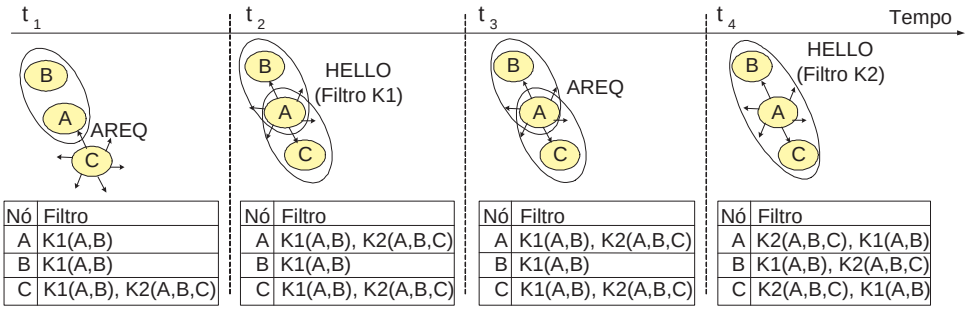
Nó	Filtro
A	K1(A,B)
B	K1(A,B)
C	K2(A,B,C)

Nó	Filtro
A	K2(A,B,C)
B	K1(A,B)
C	K2(A,B,C)

Nó	Filtro
A	K2(A,B,C)
B	K1(A,B)
C	K2(A,B,C)

Nó	Filtro
A	K2(A,B,C)
B	K1(A,B)
C	K2(A,B,C)

(a)



Nó	Filtro
A	K1(A,B)
B	K1(A,B)
C	K1(A,B), K2(A,B,C)

Nó	Filtro
A	K1(A,B), K2(A,B,C)
B	K1(A,B)
C	K1(A,B), K2(A,B,C)

Nó	Filtro
A	K1(A,B), K2(A,B,C)
B	K1(A,B)
C	K1(A,B), K2(A,B,C)

Nó	Filtro
A	K2(A,B,C), K1(A,B)
B	K1(A,B), K2(A,B,C)
C	K2(A,B,C), K1(A,B)

(b)

Figura 2.11: Exemplo de entrada de novos nós.

trocar mensagens Requisição para enviar o filtro e as assinaturas de sua partição. Em seguida, ambos os nós devem inundar a sua partição com a mensagem Partição, representada na Figura 2.10(b), para atualizar os outros nós com as novas informações recebidas na Requisição. Com base no filtro recebido, cada nó da rede pode determinar se está na menor ou na maior partição. Apenas os nós na menor partição devem verificar se existem réplicas do seu endereço e, caso existam réplicas, deve-se escolher um novo endereço não utilizado e enviar os AREQs para toda a rede, anunciando uma nova alocação de endereços. Caso as partições tenham o mesmo tamanho, a partição do nó que enviou a Requisição com $R = 1$ verifica as colisões de endereços e isso é anunciado para as duas partições através do bit M da mensagem Partição. Após receber a mensagem Partição, os nós devem atualizar o filtro, fazendo a união entre o seu filtro e o filtro da outra partição, e atualizar a assinatura do filtro.

Para evitar que múltiplos processos de união de partição relativos a uma única união de partição se iniciem ao mesmo tempo, deve-se observar a seguinte regra. Se o nó A detecta

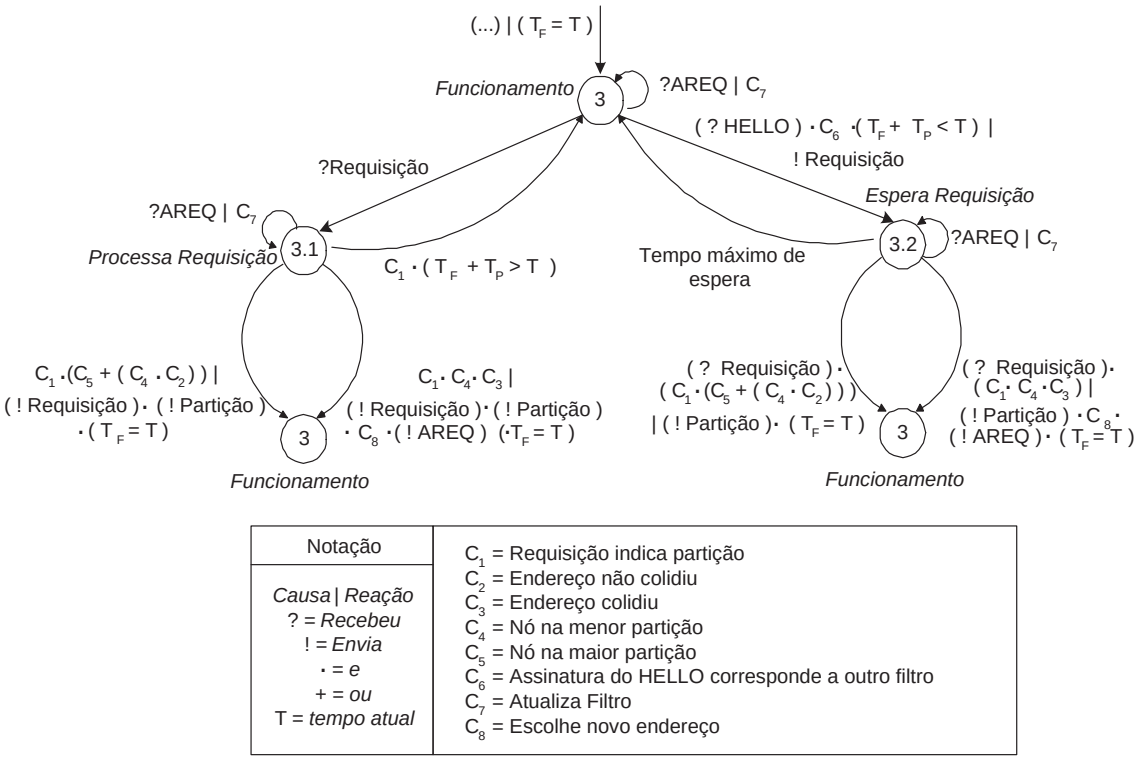


Figura 2.12: Máquina de estados da união de partições no AUFIRA.

uma partição com o nó B e se o endereço de A é maior que o de B, então A não deve iniciar o processo de união de partições. Isso evita que A e B enviem mensagens de Requisição simultaneamente. Além disso, deve-se esperar um período mínimo entre processos de união de partição para evitar ataques de negação de serviço e para evitar instabilidade do protocolo em casos de múltiplos processos de união de partição concomitantes.

Saída de Nós da Rede

A saída de nós da rede deve implicar na devolução do endereço utilizado para o espaço de endereços disponível. A saída pode acontecer através de uma notificação para toda a rede ou de forma repentina, causada, por exemplo, por uma falta de energia. Quando a saída não é repentina, cada nó, ao receber a notificação de saída, exclui o nó indicado do filtro. Quando a saída é repentina, o endereço permanece alocado no filtro. Após diversas saídas repentinas, o filtro pode passar a ter poucos ou nenhum endereço disponível, o que é identificado pelo percentual de bits em 1 no filtro.

Para solucionar o problema da devolução de endereços, basta que, ao receber uma notificação de partição ou de entrada de novo nó, cada nó verifique o percentual de bits em 1 no seu filtro. Se o percentual chegar a um nível crítico, então todos os nós devem voltar à fase de inicialização, zerando os seus filtros e enviando novos AREQs. Embora essa situação pareça dispendiosa em termos de número de mensagens enviadas, ela corresponde apenas a uma união de partições nos protocolos sem estado, ou seja, que não guardam informações sobre o conjunto de endereços alocado, como é o caso do protocolo de Fan e Subramani [21]. Para evitar constantes renovações do filtro em redes com o espaço de endereços muito ocupado, deve-se esperar um período mínimo entre as renovações de filtro.

2.3 Ambiente de Simulação

O desempenho do AUFIRA foi avaliado utilizando o ns-2, que é um simulador amplamente usado na avaliação de desempenho de protocolos para redes ad hoc. Como protocolo de roteamento utilizou-se o *Ad Hoc On-Demand Distance Vector routing protocol* (AODV), enquanto que na camada MAC utilizou-se o IEEE 802.11g. O modelo de propagação utilizado foi o *Shadowing* com parâmetros relativos a uma rede comunitária. Foi assumido que os nós possuem um alcance médio de $18,5$m e um limiar de interferência da portadora máximo de 108m, para representar valores comerciais típicos de equipamentos IEEE 802.11. A densidade foi definida como $\approx 0.0121\ nós/m^2$, para manter os parâmetros de uma rede comunitária [11].

O AUFIRA foi comparado com o protocolo proposto por Perkins *et al.* [20], porque o mecanismo do *Duplicated Address Detection* (DAD) é a base para muitos outros protocolos, embora não solucione as uniões de partições. Além disso, o AUFIRA também foi comparado com uma adaptação da proposta de Fan e Subramani [21], um protocolo sem estado, que utiliza identificadores de rede para identificar partições e o DAD para permitir a entrada de novos nós e a solução de colisões durante a união de partições. A proposta inicial do protocolo de Fan e Subramani foi modificada porque sugeria que os identificadores de rede fossem somados a cada união de partições. Essa estratégia, no

entanto, causa instabilidade no protocolo e foi substituída para permitir uma comparação mais justa com o AUFIRA. A adaptação feita foi, ao invés de somar os identificadores, utilizar o maior identificador de rede sempre que acontecer alguma união de partições. Ao comparar o AUFIRA com a proposta de Fan e Subramani, é possível determinar o impacto do uso de filtros de endereço na redução do volume de tráfego de controle enviado e na detecção correta de partições da rede.

Os parâmetros do AUFIRA e dos outros protocolos utilizados na simulação estão na Tabela 2.1. Os parâmetros foram escolhidos com base em experimentos, de forma a melhorar a eficiência dos três protocolos. O número de transmissões das mensagens inundadas, N_R, foi variado entre 1 e 3, para avaliar o impacto do aumento do número de retransmissões. Os HELLOs utilizados pelo AUFIRA e pela proposta de Fan e Subramani possuem os mesmos campos e o identificador de rede foi escolhido do mesmo tamanho que a assinatura do filtro, composta por 4 octetos. O intervalo entre HELLOs foi escolhido como 1 s. Optou-se por utilizar o Filtro Simplificado, pois o espaço de endereços disponíveis escolhido foi de 150 endereços. Esse valor foi escolhido para representar um espaço de endereços cerca de três vezes maior que o número médio de nós utilizando a rede. Assim, o filtro ocupa um espaço de 23 octetos. Os resultados apresentados possuem um intervalo de confiança de 95% e o tempo de simulação para cada rodada foi de 40 s.

Tabela 2.1: Parâmetros do AUFIRA.

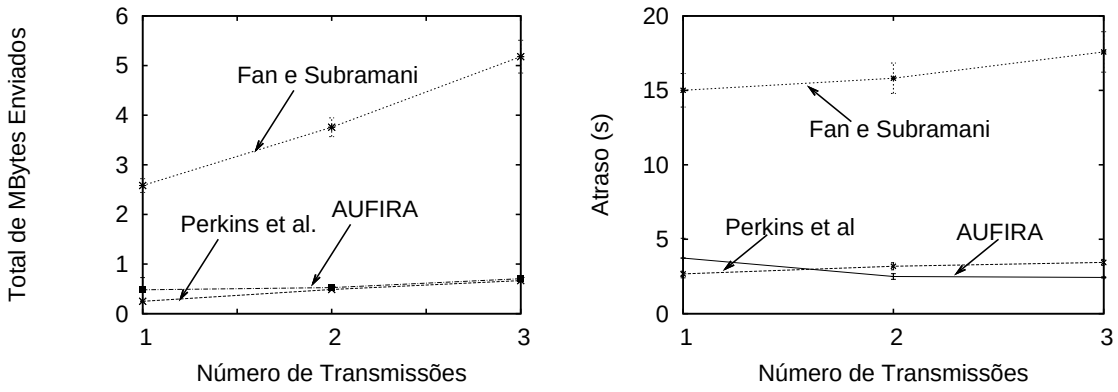
Variável	**Descrição**	**Valor**
T_O	Tempo de observação do meio na entrada na rede	1s
T_P	Tempo de espera entre partições	3s
T_T	Tempo de espera para troca de endereços	0,3s
T_I	Tempo de espera por AREQs na inicialização	1,2s
T_R	Intervalo entre retransmissões	0,3s
T_{B1}	Tempo de manutenção de filtros backup de outra partição	3s
T_{B2}	Tempo de manutenção de filtros backup de própria partição	0,5s
T_F	Tempo para troca de filtro	1s

2.4 Resultados

A eficiência do procedimento de inicialização da rede é analisada na primeira simulação, na qual utilizou-se um cenário em grade com 49 nós. Os endereços são escolhidos aleatoriamente por cada nó com base em geradores pseudo-aleatórios com uma semente diferente para cada nó. Na simulação, todos os nós entram simultaneamente na rede. O número de colisões médio no conjunto inicial de endereços foi de aproximadamente 8 colisões e, ao final da simulação, todos os protocolos solucionaram todas as colisões.

O total de octetos emitidos por transmissão ou retransmissão de mensagens de controle dos protocolos de autoconfiguração de endereço é representado na Figura 2.13(a). Por essa figura, é possível observar que, mesmo o protocolo de Perkins *et al.* sendo um protocolo simples que não trata união de partições e não transmite HELLOs, a sua carga não foi muito inferior à emitida pelo AUFIRA. Isso se deve ao fato de o AUFIRA não necessitar de mensagens de *Address Reply* (AREP). Uma vez que não existe nenhum tipo de prioridade para o uso dos endereços, é mais vantajoso trocar de endereço ao receber um *Address Request* (AREQ) do que enviar um AREP e esperar que o outro nó troque de endereço e envie um novo AREQ. Outro resultado interessante é que, com apenas uma transmissão, o desempenho do AUFIRA com relação ao protocolo de Perkins *et al.* diminui. Com apenas uma transmissão, nem todos os AREQs alcançam todos os nós da rede, de forma que o AUFIRA compensa essas perdas com falsos processos de união de partição que permitem que todos os nós sejam notificados sobre todos os endereços. Os outros dois protocolos não possuem essa função, e, assim, fazem uso da probabilidade de a mensagem perdida não ser relativa a algum caso de colisão de endereços.

O protocolo de Fan e Subramani apresentou o pior resultado, devido a também utilizar mensagens de HELLO, ser sem estado e não tratar a entrada simultânea. De fato, o protocolo de Fan e Subramani supõe a existência de um intervalo significativo entre a entrada do primeiro e do segundo nó na rede. Como os nós entram simultaneamente, cada nó escolhe um identificador de rede próprio, e os nós passam a encarar os vizinhos como partições. Por essa razão, o protocolo de Fan e Subramani apresentou uma carga até 7 vezes maior que o AUFIRA.



(a) Carga de controle na inicialização.

(b) Atraso na inicialização.

Figura 2.13: Carga provocada por mensagens controle e atraso na inicialização da rede.

Os atrasos com o processo de autoconfiguração de endereços estão representados na Figura 2.13(b). Esses atrasos foram medidos como o tempo até a última recepção de mensagens de controle relativas à inicialização da rede. Uma vez que durante a autoconfiguração de endereços os nós podem trocar de endereços algumas vezes, não seria interessante permitir o estabelecimento de conexões até o fim do processo. Os resultados dessa análise mostram que para duas ou três repetições, o AUFIRA tem o melhor resultado, apresentando um ganho de até 1 s com relação ao protocolo de Perkins *et al.* e 15 s com relação ao protocolo de Fan e Subramani. Com apenas uma transmissão, o AUFIRA apresenta um atraso maior devido aos falsos processos de união de partição, que aumentam o atraso até a estabilização do processo de inicialização da rede.

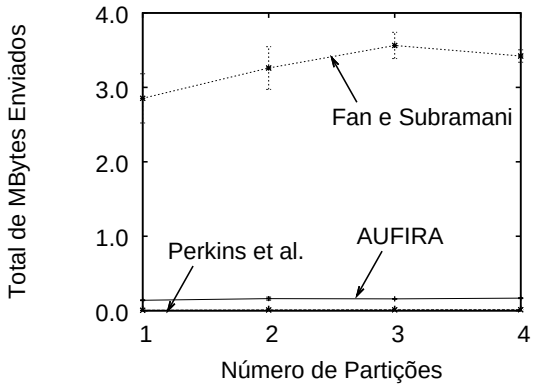
Com base nessa primeira análise, é possível afirmar que com relação ao número de colisões, todos os protocolos se mostraram satisfatórios, solucionando todos os conflitos. Com relação ao volume de tráfego de controle, o protocolo de Perkins *et al.* apresenta o melhor resultado, embora a carga imposta pelo AUFIRA seja menos do que 7% maior que a do protocolo de Perkins para duas ou três repetições. Por outro lado, o AUFIRA se apresenta como um protocolo mais robusto contra a perda de pacotes. Ao se utilizar a repetições de mensagens, a robustez das outras propostas é aumentada, mas em compensação, os atrasos se tornam muito maiores do que no AUFIRA.

A análise seguinte foi relativa à formação de partições. Em um cenário estático formado por 50 nós, variou-se a quantidade de uniões de partições através da ativa-

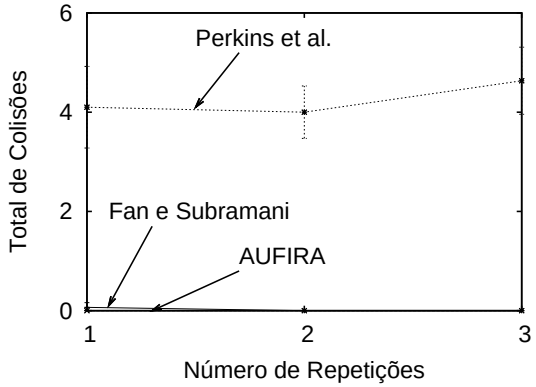
ção/desativação de determinados nós responsáveis por conectar as partições isoladas. Neste cenário, utilizou-se $N_R = 2$, ou seja, cada mensagem de controle inundada é transmitida duas vezes. Os nós nas partições são ligados simultaneamente e os nós que interconectam as partições são ligados em intervalos pré-determinados, ficando ativos até o fim da simulação. O gráfico representando a quantidade de octetos transmitidos ou retransmitidos após o início das uniões de partições está na Figura 2.14(a). Por esse gráfico, é possível perceber que o protocolo de Perkins *et al.* não enviou nenhum tipo de mensagem de controle, pois esse protocolo não é capaz de detectar partições e solucionar colisões formadas durante a união de partições. Assim, o protocolo é ineficiente em ambientes com partições. Esse fato pode ser observado na Figura 2.14(b), que mostra o número de colisões de endereço no final da simulação com uma união de partição. Por esse gráfico, observa-se que o protocolo de Perkins não soluciona nenhuma das colisões formadas após a união de partições e o protocolo de Fan e Subramani não é sempre eficiente, pois existiram alguns casos em que, após o mecanismo, ainda existiam colisões. Essa situação não ocorre no AUFIRA devido ao uso dos filtros de endereço, que permitem indicar as partições com precisão e corrigir as perdas de mensagens na rede.

Durante a união de partições, o AUFIRA produziu uma carga até 22 vezes menor que a do protocolo de Fan e Subramani, pois evita o excesso de AREQs durante a união de partições. De fato, com o AUFIRA, apenas metade dos nós envolvidos em colisões durante as partições troca de endereço e envia AREQs, enquanto que, no protocolo de Fan e Subramani, todos os nós verificam seus endereços com AREQs.

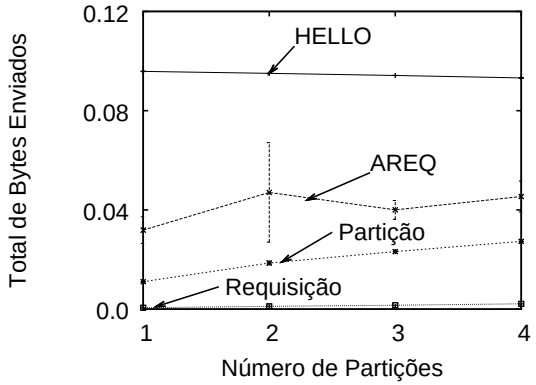
Na Figura 2.14(a), observa-se que o total de octetos transmitidos não aumenta com o número de uniões de partições no AUFIRA. Isso se explica devido à carga de controle na união de partições ser composta, em sua maioria, por mensagens de HELLO, cujo número não varia devido às partições. Além disso, as mensagens de Requisição, que aumentam de número com o número de uniões de partições, são enviadas em *unicast* e, assim, não causam impactos. A quantidade de mensagens de AREQ também não aumenta com o número de partições porque essas mensagens são enviadas por inundação e quanto mais partições, menor o número de nós em cada partição. A distribuição das mensagens do AUFIRA durante a união de duas partições está na Figura 2.14(c).



(a) Carga de controle na união de partições.



(b) Número de colisões na união de partições.



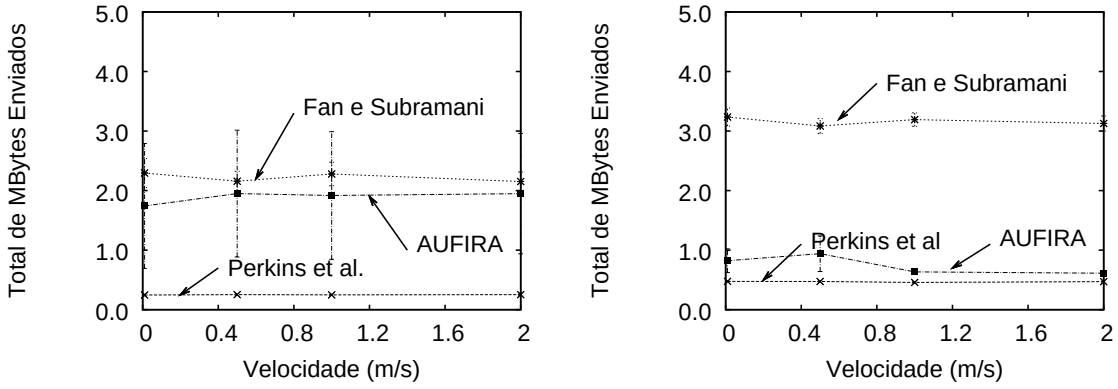
(c) Mensagens na união de partições.

Figura 2.14: Efeito da união de partições.

O efeito da mobilidade também foi avaliado em dois cenários de 50x50m com baixa probabilidade de formação de partição e compostos por 49 nós. Os nós se movem segundo o modelo *random-way point*, com tempo de pausa de 1 s.

No primeiro cenário, cujos resultados estão na Figura 2.15(a), para uma transmissão, e na Figura 2.15(b), para duas transmissões da mesma mensagem, os nós foram ligados simultaneamente. Os três protocolos não sofreram efeitos com o aumento da mobilidade, mas o AUFIRA apresentou uma carga inferior ao aumentar o número de repetições. A inserção de mobilidade sem repetição das mensagens acaba levando mais nós a não receber todos os AREQs e assim o AUFIRA precisa trocar mais mensagens para garantir que todos os nós receberam todos os anúncios de endereço. Enquanto os nós não recebem todos os anúncios de endereço, os filtros ficam diferentes e falsos processos de partição são iniciados até que todos os filtros se igualem. Quando os filtros ficam iguais signi-

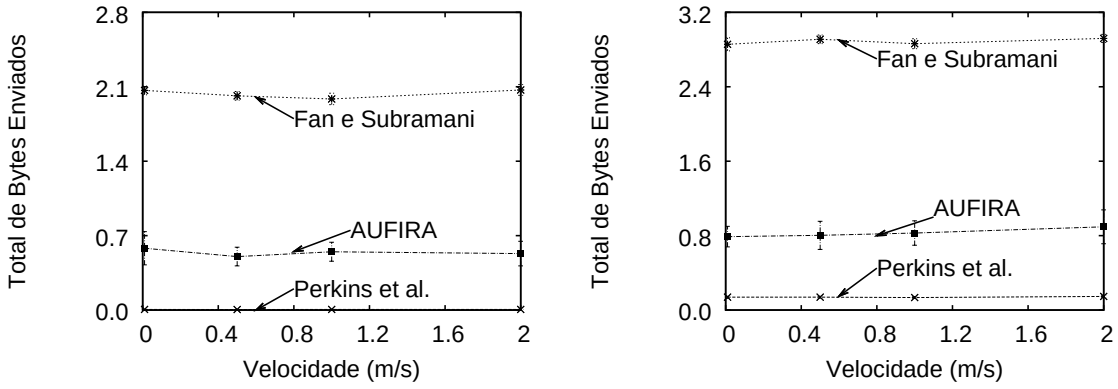
fica que todos os nós possuem o mesmo conjunto de endereços alocados. Além disso, a mobilidade pode causar, ocasionalmente, partições reais na rede que são detectadas tanto pelo AUFIRA quanto pelo protocolo de Fan e Subramani, enquanto que o protocolo de Perkins não sofre alterações por não monitorar partições. Com apenas uma transmissão, o protocolo de Fan e Subramani apresenta uma carga até $1,3$ vezes maior que a carga do AUFIRA e com duas transmissões, essa diferença chega a aproximadamente 5 vezes.



(a) Carga de controle com uma transmissão. (b) Carga de controle com duas transmissões.

Figura 2.15: Efeito da mobilidade com entrada simultânea.

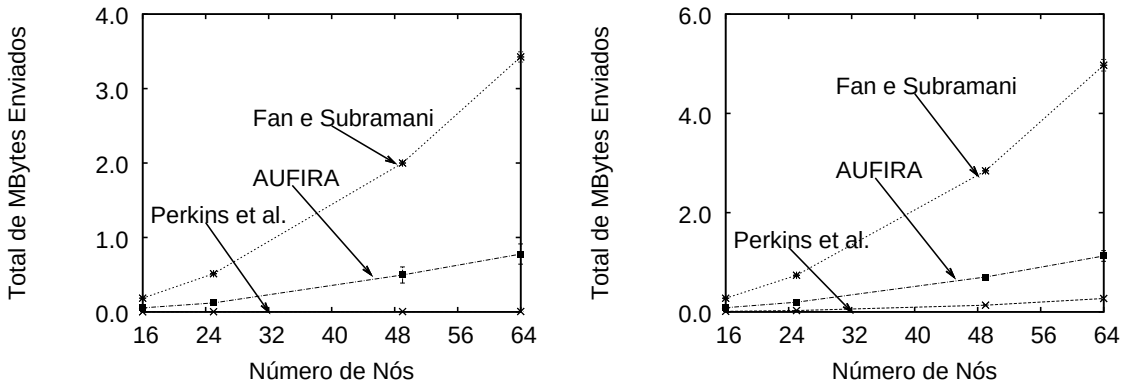
No segundo cenário, cujos resultados estão na Figura 2.16(a) e 2.16(b), os nós entram na rede de forma não simultânea. O que se observa é que a entrada não simultânea traz vantagens para todos os protocolos, pois o número de nós na rede aumenta gradativamente, e, assim, as primeiras inundações implicam em menos retransmissões de pacotes.



(a) Carga de controle com uma transmissão. (b) Carga de controle com duas transmissões.

Figura 2.16: Efeito da mobilidade com entrada não-simultânea.

A última análise foi referente ao efeito do número de nós na rede. O cenário é formado por nós estáticos organizados em grade. O momento de entrada de cada nó é escolhido aleatoriamente entre 1 e 20 s, enquanto que o momento de saída, entre 20 e 40 s. Os resultados dessa análise para uma e duas transmissões das mensagens de controle que são inundadas estão representados nas Figuras 2.17(a) e 2.17(b). Por esses gráficos, é possível observar que o aumento do número de nós tem grande impacto sobre os protocolos, devido à utilização de inundações. O AUFIRA é menos afetado pelo aumento do número de nós que a proposta de Fan e Subramani, que chega a ter uma carga até $4,4$ vezes maior que a do AUFIRA. Isso se deve ao AUFIRA reduzir a emissão de mensagens que são inundadas na rede durante a inicialização.



(a) Carga de controle com uma transmissão.

(b) Carga de controle com duas transmissões.

Figura 2.17: Efeito da variação do número de nós.

2.5 Considerações Finais

Os nós de uma rede ad hoc devem receber um endereço único antes de poderem se comunicar. Idealmente, esse processo deve ser automático, rápido, sem erros e deve poupar recursos do nó. Nesse capítulo, foram analisadas as principais propostas para a autoconfiguração de endereços em redes as hoc, mostrando seus pontos positivos e negativos. O protocolo AUFIRA foi proposto com o objetivo de atender melhor aos requisitos de uma alocação de endereços em redes ad hoc. A utilização de filtros de endereço trouxe grandes vantagens em termos de redução do número de mensagens de controle trocadas e atrasos

durante a fase de autoconfiguração de endereços, ao custo do armazenamento em cada nó de uma pequena quantidade de octetos.

Os resultados das simulações mostram que o AUFIRA é capaz de solucionar todas as colisões de endereço mesmo em situações de união de partições, enviando poucas mensagens de controle. Além disso, os resultados mostram que o AUFIRA sofre um impacto menor com o aumento do número de nós na rede que os protocolos semelhantes à proposta de Fan e Subramani. O AUFIRA apresenta uma carga durante a inicialização da mesma ordem de grandeza que a do protocolo de Perkins *et al.*, que não é capaz de monitorar as partições. Quando comparado aos protocolos sem estado que tratam a união de partições, o AUFIRA possui uma carga de controle até 22 vezes menor. Outro resultado interessante é que o AUFIRA reduz o atraso na autoconfiguração de endereços quando se usa pelo menos uma repetição de cada uma das mensagens inundadas. Por fim, o AUFIRA se mostrou um protocolo mais robusto contra perda de pacotes, o que é de primordial importância em ambientes com muita perda. Portanto, o AUFIRA é um protocolo completo, eficiente e adequado às características das redes ad hoc, como as freqüentes partições da rede. De fato, a economia de energia devido às poucas mensagens emitidas é de grande importância para a maioria dos dispositivos móveis e a robustez à perda de pacotes é essencial para garantir um processo de alocação seguro.

Capítulo 3

Autenticação e Monitoração

NO contexto de segurança de redes, o controle de acesso é a habilidade de limitar e controlar o acesso de dispositivos às máquinas da rede e às aplicações. Para alcançar esse controle, cada dispositivo tentando obter acesso deve, primeiramente, ser identificado e em seguida autenticado, de forma que os direitos de acesso possam ser concedidos ao dispositivo.

O serviço de identificação pode ser provido por protocolos como o AUFIRA, descrito no Capítulo 2, onde o endereço IP atribuído é um identificador do nó. Já o serviço de autenticação diz respeito à garantia de que uma comunicação é autêntica, ou seja, que os nós emissores e receptores de mensagens são realmente quem dizem ser. Um método muito conhecido para realizar esse tipo de proteção é a utilização de assinaturas criptográficas nas mensagens. Normalmente, a utilização do serviço de autenticação permite também a proteção das mensagens contra modificação do conteúdo. Para tanto, o emissor deve gerar um *hash* da sua mensagem e criptografá-lo com o seu material criptográfico. Esse é o princípio para as assinaturas digitais, que são amplamente utilizadas em redes para a garantia de segurança.

A assinatura digital tem como base a utilização de chaves públicas e privadas. Através dessas chaves, um nó pode garantir que é o autor de uma determinada mensagem e que o conteúdo não foi modificado, conforme ilustrado na Figura 3.1. No entanto, para evitar que nós maliciosos se passem pelo nó emissor da mensagem, é necessário que exista uma

entidade confiável que relacione cada chave pública com uma determinada identificação. Essa entidade é chamada de autoridade certificadora e tem como principais funções registrar nós, emitir certificados que associam a identidade à chave pública e fazer o controle dos certificados, mantendo uma lista de revogações que contém todos os certificados que não devem mais ser aceitos, embora ainda estejam dentro da validade.

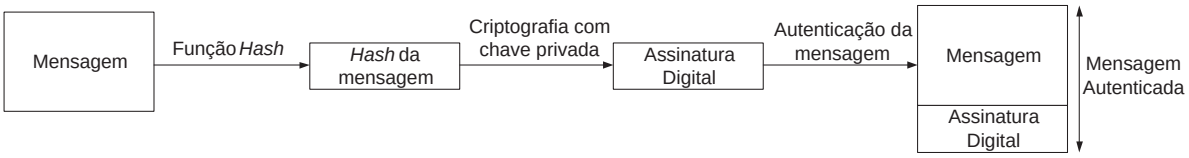


Figura 3.1: Assinatura digital de mensagens.

Nas redes ad hoc, a identificação dos nós pode ser feita por meio do IP conforme descrito no Capítulo 2. A autenticação deve ser realizada por uma entidade confiável, mas, devido às características da rede, essa entidade deve ser distribuída. De fato, ao contrário das redes cabeadas, onde existe um servidor que realiza a autenticação, nas redes ad hoc não existe nenhum tipo de infra-estrutura fixa disponível em todos os momentos para qualquer nó da rede. Assim, se faz necessário o desenvolvimento de sistemas distribuídos de autenticação e certificação digital nas redes ad hoc.

3.1 Autenticação em Redes Ad Hoc

O acesso às redes ad hoc pode ser feito de duas formas: sem autorização e com autorização. No acesso sem autorização, a rede é totalmente auto-organizada e não possui nenhum tipo de autoridade *online* ou *offline* [30]. Essas redes são criadas pelos usuários finais em um modo totalmente ad hoc. Nesse caso, não existe nenhum tipo de relação de segurança pré-estabelecida entre os usuários e não existe nenhum tipo de material criptográfico previamente compartilhado. Assim, quando a rede se torna operacional, não existe nenhum tipo de entidade que realize o controle de acesso na rede. Dessa forma, o funcionamento da rede depende inteiramente da cooperação e confiança entre os nós [31]. Nesse tipo de rede, qualquer nó pode participar a qualquer momento da rede, embora o contraponto dessa facilidade seja que se torna muito difícil impedir ataques, já que nós excluídos podem retornar à rede com uma identidade distinta. No acesso com autorização, alguma

entidade precisa autorizar o acesso. Nesse tipo de rede, os nós possuem algum tipo de relação de confiança pré-estabelecida, que é formada antes da inicialização ou a qualquer momento. No caso das relações de confiança formadas antes da inicialização da rede todo material criptográfico e alguns parâmetros universais são distribuídos para todos os nós. No caso das relações de confiança formadas a qualquer momento, a entidade autenticadora é distribuída e fica disponível em todos os momentos, permitindo o provimento de todos os serviços de uma autoridade certificadora ao longo do funcionamento da rede, tais como o registro de novos usuários, a emissão e a revogação de certificados [32]. Esse segundo tipo de rede ad hoc é o foco deste trabalho, pois se considera importante o controle de acesso para garantir um bom desempenho na rede e evitar o retorno indiscriminado de nós maliciosos já excluídos.

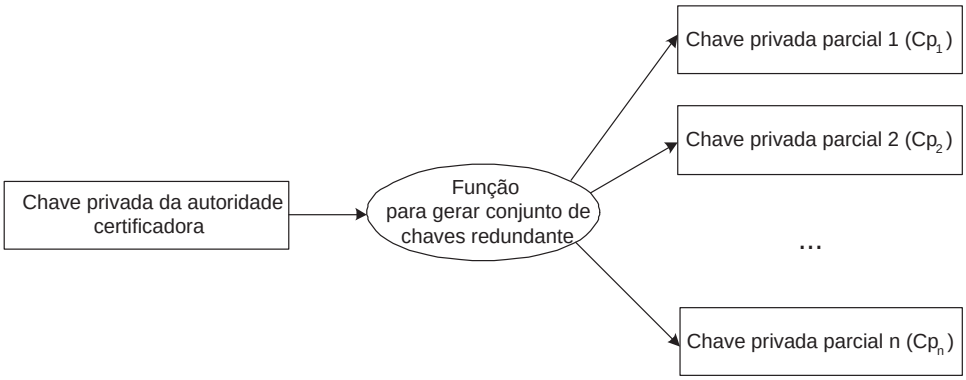
3.1.1 Entidades Autenticadoras em Redes Ad Hoc

Para realizar a autenticação em redes ad hoc, podem-se utilizar alguns tipos de entidades distribuídas capazes de verificar as identidades dos nós. A seguir, são descritos alguns dos principais sistemas de autenticação que utilizam esses diferentes tipos de entidade distribuída.

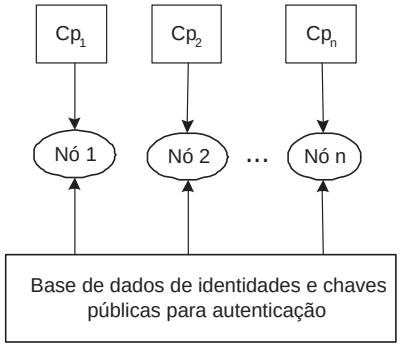
Autoridades Certificadoras Parcialmente Distribuídas

Os protocolos enquadrados nesse grupo distribuem a confiança na autoridade certificadora para um subconjunto de nós na rede. Uma das principais propostas nesse grupo foi feita por Zhou e Haas [33]. Nessa proposta, a chave privada da autoridade certificadora é distribuída entre n nós por meio de criptografia de limiar [34]. A Figura 3.2 mostra a geração das chaves parciais e a inicialização dos nós que representam a autoridade certificadora. As chaves parciais são geradas de forma que uma combinação de k assinaturas com chaves parciais gere uma assinatura idêntica à que seria gerada pela chave privada da autoridade certificadora. Assim, cada um dos n nós pré-selecionados para formar a autoridade certificadora distribuída recebe um compartilhamento Cp_i da chave privada. A autoridade certificadora distribuída emite certificados através da união de pelo menos k

assinaturas parciais, como mostrado na Figura 3.3. A desvantagem dessa proposta é que é necessário que exista um administrador *offline* que deve escolher e pré-determinar quais os n nós escolhidos, além de ser responsável pelo gerenciamento do controle de acesso, determinando quais usuários podem entrar na rede e quais certificados devem ser revogados. Além disso, é preciso que se garanta que pelo menos k dos n nós estarão sempre disponíveis para todos os nós, o que é uma premissa forte em redes onde é freqüente a entrada e a saída dos nós, assim como a formação de partições. Outra questão é que, em sistemas baseados em criptografia de limiar, é preciso assumir que atacantes móveis não são capazes de comprometer k dos n nós selecionados, o que exporia toda a rede.



(a) Criação das chaves para criptografia de limiar.



(b) Inicialização dos nós.

Figura 3.2: Inicialização da autoridade certificadora com criptografia de limiar.

Outro exemplo de autoridade certificadora parcialmente distribuída que apresenta características semelhantes é apresentada nos trabalhos de Yi e Kravets [35], cuja principal diferença para o trabalho de Zhou e Haas é a ausência de um nó que realiza a união dos compartilhamentos de assinatura. Pereira *et al.* também tratam de autoridades certificadoras distribuídas, mas considerando aspectos como a reconfiguração do grupo de

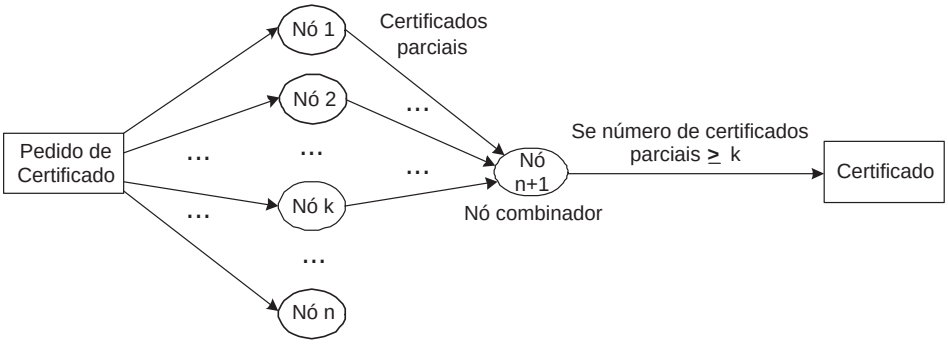


Figura 3.3: Geração dos certificados.

servidores para melhorar a disponibilidade do serviço na rede [36].

Autoridades Certificadoras Totalmente Distribuídas

Esse grupo de protocolos preserva a igualdade entre os nós da rede distribuindo igualmente a responsabilidade do gerenciamento de chaves por todos os nós [37, 38]. Estes protocolos também utilizam a criptografia de limiar para gerar os certificados, mas, nesse caso, n é igual ao número de nós na rede. Essa diferença dá maior robustez ao protocolo no sentido de disponibilidade da autoridade certificadora, embora implique em uma maior vulnerabilidade, pois agora basta comprometer k nós quaisquer da rede para obter a chave privada da autoridade certificadora. Para reduzir o impacto desse problema, as propostas de autoridades certificadoras totalmente distribuídas incluem algoritmos para renovação periódica da chave.

Os problemas da autoridade certificadora parcialmente distribuída são quase todos mantidos para a autoridade totalmente distribuída. Primeiramente, a autoridade *offline* precisa conhecer todos os nós antes da formação da rede, registrando um certificado para cada identificação. Além disso, a solução tem como restrição a necessidade de conhecer todas as identidades *a priori*.

Gerenciamento de Chaves Distribuído Baseado em Identidade

Esse conjunto de protocolos possui características semelhantes às apresentadas pelas autoridades certificadoras parcialmente distribuídas por também utilizar a criptografia de

limiar. A diferença entre os dois grupos é que, no gerenciamento de chaves baseado em identidade, é utilizada criptografia baseada em ID [39] para reduzir os requisitos de armazenamento comuns aos cripto-sistemas de chaves públicas. Assim, a identidade de cada nó é dada pela chave pública do nó, evitando a necessidade de guardar duas listas, uma com as identificações e outra com as respectivas chaves públicas. Nos sistemas baseados em ID, a autoridade certificadora é substituída pelo *Private Key Generator* (PKG), responsável por gerar as chaves privadas de cada nó. Todos os nós devem conhecer a chave pública do PKG para realizar as operações criptográficas na rede [40].

O PKG distribuído, adequado a redes ad hoc, é apresentado por Khalili *et al.* [41]. Essa proposta, no entanto, ainda possui alguns pontos em aberto, como a distribuição para cada nó de forma segura da chave privada gerada pelo PKG. Como pontos negativos, o PKG distribuído herda todas as desvantagens da autoridade certificadora distribuída. Além disso, o comprometimento do PKG é mais grave que o comprometimento da autoridade certificadora, uma vez o PKG conhece todas as chaves privadas de todos os nós.

Gerenciamento de Chaves Baseado em Cadeias de Certificados

Nesse grupo de protocolos, os nós se autenticam por meio da busca por cadeias de certificados em comum. Assim, se o nó A deseja se comunicar com o nó C, com o qual o nó A nunca teve contato, o nó A precisa autenticar o certificado de C. Supondo que o nó A conheça um terceiro nó, B, no qual A confia. Se B já conhece C, pode informar a A que C é autêntico e, dessa forma, A passa a confiar também em C. Portanto, diferentemente das outras propostas, no gerenciamento de chaves baseado em cadeias de certificados, não existe nenhum tipo de entidade na qual todos os nós precisam confiar, o que torna esse tipo de solução mais adequado às redes ad hoc.

No gerenciamento de chaves baseado em cadeias de certificados, cada nó deve gerar o seu próprio certificado, de forma semelhante ao funcionamento do *Pretty Good Privacy* (PGP) [42], embora, neste caso, não exista um repositório central de certificados. De fato, com o gerenciamento de chaves baseado em cadeias de certificado, cada nó possui um repositório parcial de certificados formado por nós na sua vizinhança. Quando dois nós desejam validar seus certificados, eles trocam os seus repositórios e tentam achar uma

cadeia de certificados em comum [43, 44].

Esse esquema foi desenvolvido para as redes ad hoc totalmente auto-organizadas, ou seja, que não possuem nenhum tipo de relação de segurança entre os nós. Assim, comparar essa proposta com as anteriores não tem muito sentido, uma vez que os propósitos são diferentes. Um problema dessa proposta é que as relações de confiança entre nós em redes ad hoc levam tempo para se formar, uma vez que elas exigem observação da atitude dos nós, o que pode gerar um problema durante a inicialização da rede.

Capkun *et al.* sugeriram que fosse utilizado um segundo canal, como uma interface com infravermelho, para a troca de material criptográfico. Assim, enquanto os nós se movimentam, os usuários poderiam utilizar o contato visual para identificar o material recebido [30]. Essa solução, no entanto, ainda apresenta o problema do atraso até que se obtenha um número suficiente de relações de confiança. Além disso, nos sistemas baseados em cadeias de certificados, por não existir um controle de acesso, mesmo com um sistema capaz de detectar ações maliciosas e excluir os nós mal-intencionados, os nós maliciosos poderiam retornar à rede criando uma nova identidade e um novo certificado auto-assinado. Outra questão é que uma cadeia depende da segurança de cada um de seus elos. Assim, basta um nó comprometido para gerar uma falsa autenticação dentro da rede.

Gerenciamento Baseado em Pré-Distribuição de Chaves

Esse tipo de gerenciamento de chaves conta com uma autoridade que distribui material criptográfico antes da formação da rede. Assim, uma entidade gera um conjunto de chaves simétricas, o qual é distribuído em pequenos subconjuntos com interseções, como ilustrado na Figura 3.4. Esses subconjuntos são distribuídos entre os nós da rede de forma aleatória. Quando a rede é inicializada, os nós vizinhos verificam se possuem chaves em comum, formando um grafo onde as arestas representam a existência de pelo menos uma chave simétrica em comum, o que significa que o par de nós pode se comunicar de forma segura, como mostrado na Figura 3.5. Esse esquema não garante que existirá pelo menos uma chave em comum entre todos os nós, mas que todos os nós podem se comunicar através de um caminho seguro com alta probabilidade [45–47].

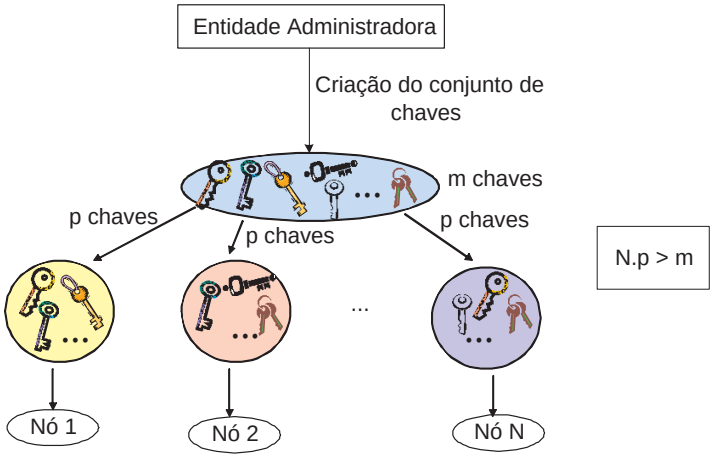
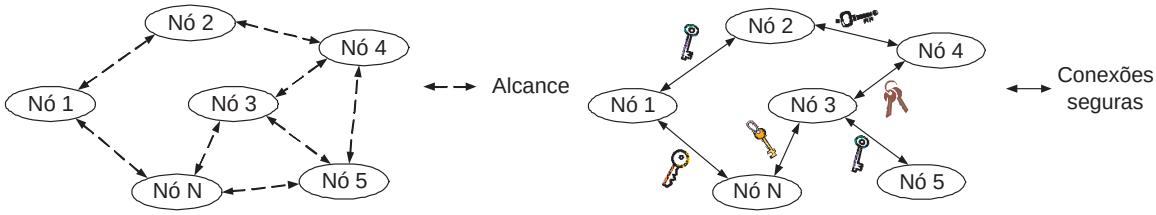


Figura 3.4: Inicialização da pré-distribuição de chaves.



(a) Grafo de conectividade.

(b) Grafo com conexões seguras.

Figura 3.5: Grafos da pré-distribuição de chaves.

A pré-distribuição de chaves tem por objetivo autenticar um grupo de nós, ou seja, por meio dela, é possível dizer se um determinado nó pertence ou não ao grupo. Em geral, a pré-distribuição de chaves é utilizada em redes com restrições de energia, armazenamento e processamento, como por exemplo em redes de sensores, pois os sistemas de autenticação anteriores possuem um alto consumo energético devido às diversas trocas de mensagens. No entanto, um dos principais problemas da pré-distribuição de chaves simétricas é a vulnerabilidade, pois o comprometimento de um nó dá acesso ao seu subconjunto de chaves simétricas, expondo a rede à possibilidade de formar conexões entre nós legítimos e nós maliciosos. Se não existir um mecanismo de renovações periódicas de chaves o comprometimento de um único nó prejudica para sempre o grupo.

3.2 Monitoração dos nós

Em uma rede com nós identificados e autenticados, é possível monitorar as ações dos nós, determinando quais nós estão agindo de forma maliciosa ou não-colaborativa, para que seja possível excluir esses nós da rede. Dessa forma, o sistema de monitoração deve observar a rede e notificar ao sistema de controle de acesso quando um nó deve ser excluído.

A monitoração dos nós em redes cabeadas é feita por um sistema de detecção de intrusão localizado em uma única máquina, pois todos os nós ligados em um mesmo segmento de rede podem ser escutados por todas as máquinas nesse segmento. Assim, a máquina responsável pela detecção de intrusão pode observar todas as outras máquinas e avaliar o comportamento de cada uma. Nas redes sem fio ad hoc, essa detecção não pode ser feita de forma centralizada, não apenas porque a rede não tem infra-estrutura, mas também porque os nós conseguem escutar apenas os seus vizinhos. Assim, cada nó só pode ter um resultado de monitoração local, de acordo com a observação das atitudes de cada vizinho [48–50]. Portanto, em redes ad hoc, a monitoração é local e, para excluir nós da rede, é necessário que aconteça uma troca de informações entre os nós, o que é uma das tarefas dos sistemas de confiança.

Os sistemas de confiança são responsáveis por organizar a troca de informações entre os nós e por determinar quais nós são realmente maliciosos ou não-cooperativos, devendo ser excluídos da rede. Essa troca de informações deve ser robusta à inserção de informações falsas por nós maliciosos que acusam nós não-maliciosos com o objetivo de excluí-los da rede. Assim, os sistemas de confiança devem considerar fatores como a própria observação do nó, os comentários dos demais nós na rede, a confiança do nó avaliador na informação difundida, o tempo de contato entre os nós, entre outros fatores. Os sistemas de monitoração e confiança devem também considerar os falso-positivos na detecção de ações maliciosas, pois tanto a mobilidade quanto as freqüentes desconexões da rede podem levar a falsas detecções de ações maliciosas.

3.2.1 Sistemas de Monitoração e Confiança da Literatura

Alguns trabalhos incentivam a colaboração dos nós em redes ad hoc através de sistemas de punição e incentivo. O objetivo destes sistemas é evitar a presença de nós egoístas criando incentivos à colaboração e punição ao comportamento não colaborativo [51, 52]. 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 outro nó e o nó favorecido deve pagar pelo serviço que utilizou com seus créditos [53, 54]. 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 existirem *hardwares* resistentes a alterações ou bancos virtuais em que todos os nós da rede possam confiar.

Existem trabalhos [55] [56] 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 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 [57]. 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.

3.3 Sistema de Autenticação e Monitoração Proposto

O sistema proposto, chamado de Autenticação e MOnitoração em Redes Ad hoc (AMORA), realiza o controle de acesso à rede por meio da autenticação e monitoração dos nós. O objetivo do AMORA é criar um sistema de autenticação adequado às características das redes ad hoc. Ao contrário das propostas com autoridades certificadoras distribuídas, ou baseados na pré-distribuição de chaves, no sistema proposto, não

é necessário um administrador que seja responsável pela entrada de todos os nós. De fato, no AMORA, a responsabilidade de permitir a entrada de um nó, o monitoramento do funcionamento da rede e a exclusão dos nós são feitos de forma totalmente distribuída. Dessa forma, não existe nenhum nó ou usuário central que tenha controle sobre tudo o que se passa na rede. Essa é uma característica importante, porque dá maior robustez, pois inexistem pontos centrais de falha. Além disso, faz com que a proposta se adeque às redes ad hoc, já que essas, por princípio, não tem infra-estrutura ou administrador, sendo caracterizadas por serem redes auto-organizáveis.

Para manter o controle de acesso evitando a necessidade de um administrador, o AMORA utiliza cadeias de confiança. Assim, cada nó irá cooperar apenas com os usuários em que ele confia, ou ainda, nos usuários em que seus amigos confiam. Diferentemente das propostas de gerenciamento de chaves baseado em cadeia de certificados [43, 44], no sistema proposto, existem relações de confiança estabelecidas *offline* que permitem ou não a entrada de usuários na rede. Além disso, não é necessária a atualização de repositórios de certificados no AMORA, já que o sistema é formado por uma autoridade certificadora distribuída capaz de emitir certificados com base apenas na verificação do usuário que autorizou a entrada do nó na rede.

No sistema proposto, cada nó cria o seu próprio par de chaves, que deve ser relacionado ao seu IP e certificado pelas suas testemunhas. As testemunhas são nós escolhidos com base no IP e em funções *hash* para monitorar o funcionamento dos nós. Cada nó possui um conjunto de testemunhas que serão responsáveis por emitir o seu certificado, autorizar a entrada dos seus filhos na rede e monitorar as suas ações com base nas recomendações recebidas de outros nós.

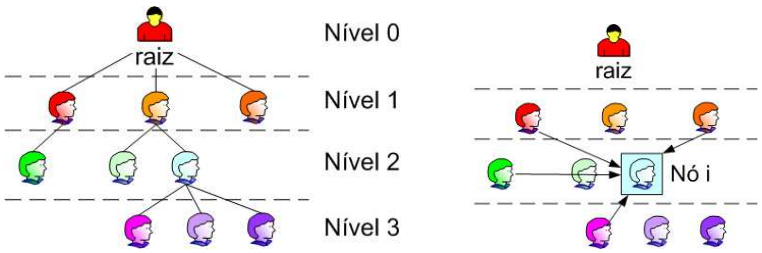
No AMORA, a autenticação e a monitoração foram colocadas na mesma entidade, representada pelas testemunhas, devido à necessidade de se revogar o certificado dos nós que se comportam mal na rede. Assim, o uso de testemunhas faz com que não seja necessário que cada nó guarde os dados de confiança sobre todos os outros nós. A idéia de utilizar testemunhas já havia sido citada para a detecção do ataque da replicação [19], mas os autores não determinaram como escolher o conjunto de testemunhas e nem como manter esse conjunto caso algumas das testemunhas se ausentassem da rede ou partições

fossem formadas.

3.3.1 Cadeia de Confiança

O sistema de autenticação e monitoração proposto é baseado em cadeias de confiança. Diferentemente dos trabalhos nos quais os nós buscavam cadeias em comum para validar os certificados, nessa proposta, a cadeia funciona como uma árvore, como ilustrado na Figura 3.6. Por simplicidade, durante a descrição do AMORA, o termo pai será utilizado para denominar o usuário que autorizou a entrada de um novo usuário e o termo filho será utilizado como referência ao novo usuário na rede.

No AMORA, a verificação com o pai e suas testemunhas é suficiente para garantir que o filho é autêntico e pertence à cadeia. De fato, o uso das testemunhas evita a necessidade de subir recursivamente pela cadeia até chegar a sua raiz para validar um determinado certificado.



(a) Cadeia de confiança. (b) Testemunhas do nó i.

Figura 3.6: Cadeias de confiança e testemunhas.

A cadeia de confiança é formada pela identidade dos usuários e não pela identidade dos nós. Dessa forma, cada nó/usuário possui dois pares de chaves assimétricas, uma correspondente à identidade do usuário e uma correspondente à identidade do nó. A identidade de cada nó é dada pelo seu IP, que deve ser escolhido segundo o protocolo descrito no Capítulo 2, que permite atribuir um endereço IP único para cada nó. A identidade dos nós serve para determinar o conjunto de testemunhas responsáveis pela monitoração dos nós. Assim, quando um nó deseja entrar na rede, ele deve escolher um endereço não utilizado, determinando o seu conjunto de testemunhas. Em seguida, para que cada testemunha valide a identificação de um novo nó, ela deve buscar as testemunhas que estão

responsáveis pelo usuário pai do novo nó. Dessa forma, os nós podem verificar relações de confiança entre os usuários.

Para um usuário pai autorizar a entrada de um usuário filho, ele deve transmitir ao filho uma autorização, ilustrada na Figura 3.7. A autorização é a forma de estabelecer as relações de confiança no mecanismo de autenticação proposto, e, dessa forma, é o ponto chave da cadeia de confiança. O primeiro campo da autorização determina qual o tipo de autorização que está sendo dada ao filho. Os valores com os quais esse campo pode ser preenchido estão indicados na Tabela 3.1.

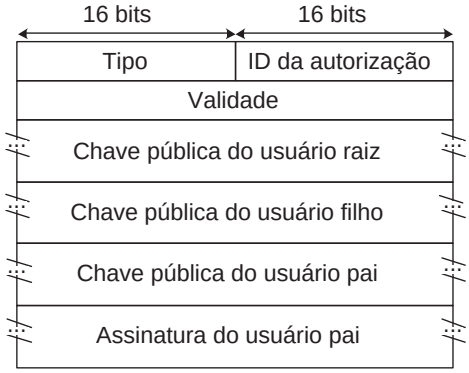


Figura 3.7: Formato da autorização.

Tabela 3.1: Tipos de Autorização.

Tipo	**Descrição**
-1	Número indefinido de filhos
0	Nó não pode ter filhos
n	Número de filhos = n

3.3.2 Testemunhas e Monitoramento

As testemunhas dos nós são responsáveis pela emissão de certificados, pelo monitoramento das ações maliciosas e pela entrada e saída dos nós na rede. Cada nó tem um conjunto formado por m testemunhas, que funciona de forma semelhante ao esquema da criptografia de limiar [39], para dar maior robustez ao esquema na saída de nós e na formação de partições. Portanto, basta que apenas k testemunhas, onde $k \leqslant m < 2k$,

se manifestem para que determinada atitude seja tomada dentro da rede. Desta forma, é necessário pelo menos k votos de testemunhas para que um nó seja expulso da rede, assim como é necessário k assinaturas de testemunhas para que um certificado se torne válido.

Existem dois tipos de conjuntos de testemunhas: as de usuário e as de nó. As testemunhas de nó são responsáveis por guardar e julgar os dados de monitoramento das atitudes do nó na rede. Assim, se um nó observa ações maliciosas de seu vizinho, ele deve contatar as testemunhas de nó do vizinho, notificando o que foi observado. Dessa forma, evita-se que todos os nós guardem informações sobre atitudes maliciosas, evitando sobrecarga de armazenamento. As testemunhas dos usuários são responsáveis por monitorar os usos e emissões de autorizações. Essas testemunhas impedem que uma mesma autorização seja usada simultaneamente diversas vezes para a criação de identidades falsas, além de controlar a entrada dos filhos do nó monitorado na rede.

Seleção das Testemunhas

As testemunhas de nó são selecionadas por meio de funções *hash*, para impedir que o nó possa manipular a escolha de seu conjunto de testemunhas, evitando o conluio na rede. Assim, cada nó deve obter um conjunto de m_n testemunhas de nó de acordo com o seu IP e um conjunto de m_u testemunhas de usuário baseado na chave pública do usuário pai. O uso da chave pública do pai é importante para evitar que a mesma autorização seja usada para obter várias identidades simultaneamente. Se um nó tentar criar uma nova identidade usando a mesma autorização, as testemunhas de nó não poderão detectar a ação maliciosa, pois os IPs das identidades são diferentes, o que implica em um conjunto de testemunhas diferente. Por outro lado, as testemunhas de usuário serão as mesmas, pois elas são escolhidas de acordo com a chave pública do pai, que está na autorização. Essas testemunhas identificarão o uso malicioso da autorização e não emitirão os certificados parciais para a segunda identidade.

As testemunhas são escolhidas utilizando uma função *hash*. A testemunha i de nó, onde $1 \leqslant i \leqslant m_n$, é escolhida segundo a equação

$$T_i = hash^i(IP), \tag{3.1}$$

e a testemunha j de usuário, onde $1 \leqslant j \leqslant m_u$, pela equação

$$T_j = hash^j(C_{pai}), \tag{3.2}$$

onde

$$hash^k(X) = hash(hash(\cdots(hash(X))\cdots)), \tag{3.3}$$

IP é o endereço IP do nó e C_{pai} é a chave pública do usuário pai.

Caso a testemunha selecionada não esteja presente no filtro de endereços da rede, o nó deve escolher uma nova testemunha. Essa nova testemunha deve ser escolhida no filtro de endereços como a que possuir o menor IP maior que o IP da testemunha ausente. Esse nó deve ser mantido como testemunha até que a testemunha ausente retorne à rede.

Cada nó é responsável por manter o seu conjunto de testemunhas ativas. Caso o nó não realize essa tarefa, o seu certificado será descartado, já que o nó que recebeu o certificado não conseguirá contatar as testemunhas para verificar a lista de revogação. Assim, cada nó deve enviar periodicamente pacotes de teste para suas m testemunhas. Caso uma testemunha deixe de responder ao pacote de teste, o nó deve anunciar à rede a ausência da testemunha e, em seguida, substituí-la.

A nova testemunha, escolhida para substituir a ausente, deve receber do nó a autorização. A nova testemunha deve proceder como se aquela fosse a entrada de um novo nó. Além disso, as testemunhas $i-1$ e $i+1$ devem repassar as suas informações armazenadas sobre o nó para a nova testemunha. Com base nas informações recebidas, a nova testemunha pode criar a sua própria base de dados sobre o nó e iniciar o monitoramento. Caso a testemunha selecionada pelo *hash*, que estava inicialmente ausente, retorne à rede, a testemunha que a substituiu deve repassar os dados que tiver armazenado para a testemunha inicial. Isto é importante para que o conjunto de testemunhas de qualquer nó possa ser determinado por todos os nós que desejarem enviar acusações ou validar certificados.

3.3.3 Entrada de Nós na Rede

A entrada de nós na rede é feita por um mecanismo que permite que o nó obtenha um par de chaves, se autentique e receba um certificado. Após a entrada, o nó passa a ser

monitorado por suas testemunhas e a monitorar outros nós.

Para entrar na rede, primeiramente, o novo usuário deve obter uma autorização com o usuário pai. Essa autorização deve ser obtida por fora da rede, embora não exista necessidade de ter sido emitida antes do momento de formação da rede.

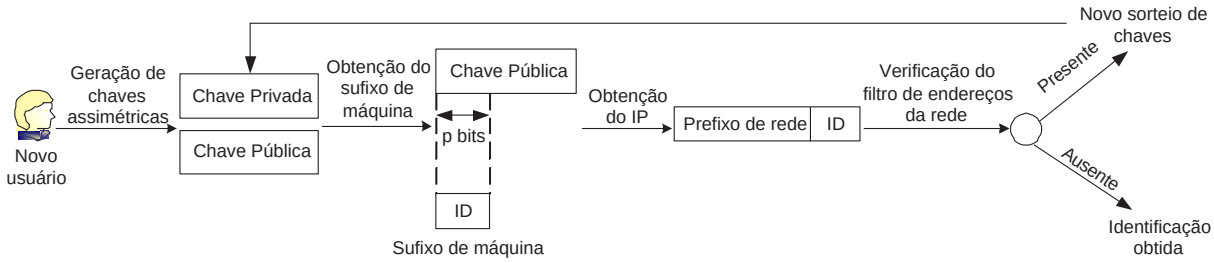
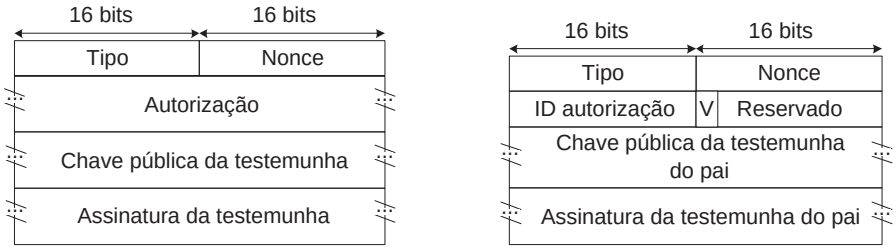


Figura 3.8: Obtenção do endereço IP.

Ao entrar na rede, o novo nó deve obter um endereço IP, conforme o protocolo descrito no Capítulo 2, que permite atribuir um endereço IP único para cada nó. Para tanto, o novo nó deve escolher um par de chaves assimétricas e fixar os primeiros p bits da chave pública como o sufixo de máquina do seu endereço IP, como ilustrado na Figura 3.8. Em seguida, o nó verifica se o IP selecionado já está presente no filtro de endereços. Caso esteja, um novo par de chaves deve ser selecionado e o processo deve ser repetido. Caso não esteja, o nó deve contatar as testemunhas correspondentes ao IP selecionado, enviando a cada uma a autorização recebida do pai. As testemunhas do novo nó devem validar a autorização junto às testemunhas do usuário pai, através da mensagem ilustrada na Figura 3.9(a).



(a) Mensagem com a autorização. (b) Mensagem com a validação.

Figura 3.9: Autorização e validação.

A troca de mensagens para validar a autorização do usuário filho está descrita na Figura 3.10. As testemunhas do usuário pai, após receberem a mensagem das testemunhas

do novo nó, verificarão se a autorização foi realmente gerada pelo pai e se ele ainda tem direito de emitir autorizações. Caso as testemunhas do usuário pai validem a autorização, através da mensagem ilustrada na Figura 3.9(b), as testemunhas do nó filho devem anunciar para toda rede que um novo endereço foi alocado. Cada nó da rede, após receber pelo menos k anúncios de novo endereço alocado (AREQ) assinados por testemunhas correspondentes ao IP anunciado, deve inserir o novo endereço no filtro de endereços. Os AREQs enviados pelas testemunhas devem ser assinados e devem conter o certificado das testemunhas do novo nó, embora não seja necessário que os nós validem os certificados recebidos. Além de alocar o novo endereço, as testemunhas do novo nó também devem enviar o certificado parcial, ilustrado na Figura 3.11, para o novo nó.

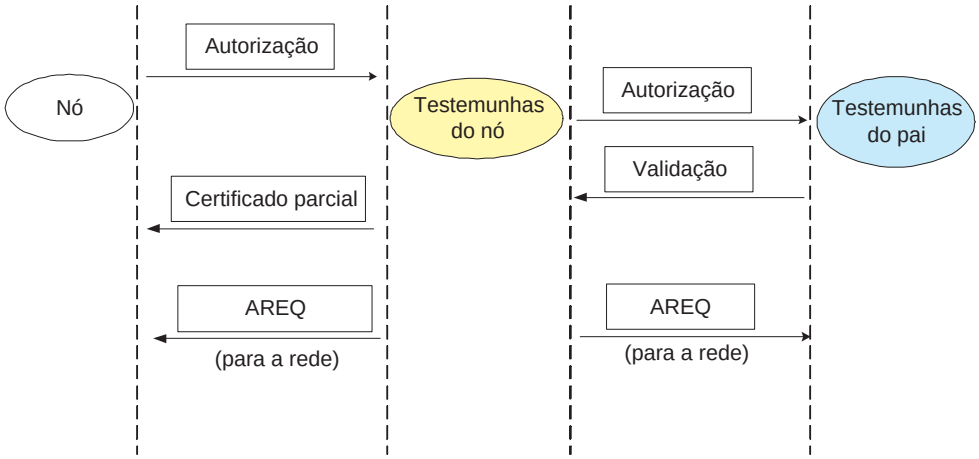
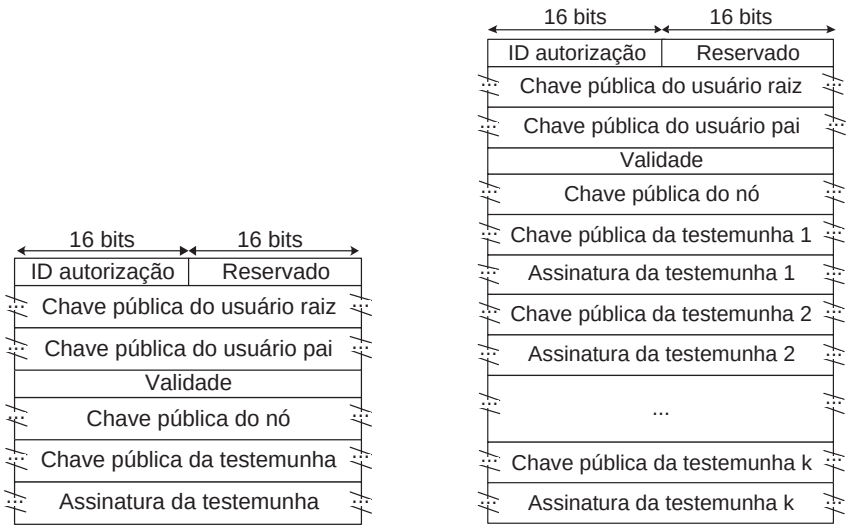


Figura 3.10: Obtenção do certificado.

Para obter o certificado completo, o novo nó deve ainda contatar as suas testemunhas de usuário, enviando para cada uma delas a autorização e obtendo como resposta os certificados parciais. O certificado completo é composto por pelo menos k_n assinaturas de certificados parciais de testemunhas do nó e pelo menos k_u assinaturas de testemunhas de usuário, onde $k = k_n + k_u$.



(a) Certificado parcial. (b) Certificado completo.

Figura 3.11: Certificados.

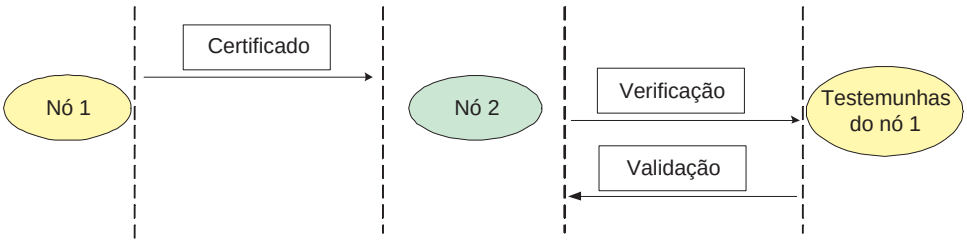
3.3.4 Emissão e Revogação de Certificados

Assim como nas autoridades certificadoras tradicionais, o sistema proposto também emite certificados e mantém uma lista de revogação para identificar os certificados que estão na validade, mas que correspondem a nós já excluídos da rede. A diferença da proposta para as autoridades tradicionais é que não existe nenhum tipo de servidor ou repositório central que pode ser consultado por qualquer nó. No sistema proposto, o papel do servidor central é cumprido pelas testemunhas de cada nó.

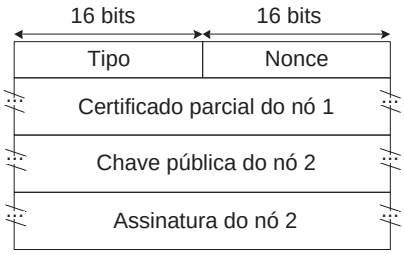
Para validar um certificado, o nó deve buscar em seu filtro de endereços as testemunhas do nó. Caso as testemunhas indicadas no certificado sejam referentes ao nó e estejam todas presentes, o nó deve fazer um pedido de verificação de certificado, como ilustrado na Figura 3.12. As testemunhas deverão responder a esse pedido informando se o certificado é válido ou não, através da mensagem Validação (Figura 3.9(b)).

3.3.5 Exclusão de nós

A exclusão dos nós é importante para impedir que nós identificados como maliciosos ou não-cooperativos acessem a rede, além de garantir que o espectro de endereços não



(a) Processo de verificação do certificado.



(b) Mensagem de verificação do certificado.

Figura 3.12: Verificação de certificados.

ficará ocupado por endereços já liberados. Dessa forma, existem dois tipos de exclusão de nós que podem ser realizadas: por mau comportamento ou por ausência. A exclusão por ausência de um nó A pode ser indicada pelas testemunhas do nó A ou pelos nós que estavam sendo monitorados pelo nó A. A ausência é detectada porque cada nó e suas testemunhas devem trocar periodicamente pacotes de teste, para que o conjunto de testemunhas ativas fique sempre atualizado.

A exclusão por mau comportamento só pode ser realizada pelas testemunhas de nó, após o voto de pelo menos k_n testemunhas. Cada testemunha deve armazenar as informações recebidas sobre ações maliciosas ou não-cooperativas do nó observado. Essas informações devem ser avaliadas segundo um sistema de confiança, para analisar se o nó é ou não malicioso ou não-cooperativo, devendo ser excluído da rede.

No AMORA, sempre que um filho é excluído, o pai e todos os nós acima do pai na cadeia perdem pontos na confiança. Esse mecanismo é importante para impedir que nós maliciosos criem filhos falsos para realizar as ações maliciosas sem serem identificados. Com esse mecanismo, o poder de criação de filhos falsos fica comprometido, pois após a criação de certa quantidade de filhos falsos maliciosos, o pai acabará sendo excluído. Além disso, o sistema proposto também determina que se um nó é excluído, também

devem ser excluídos todos os seus descendentes, o que garante a legitimidade de todos os nós na cadeia.

O sistema de monitoração do AMORA pode funcionar com diferentes tipos de sistemas de detecção de intrusão e de confiança. Por questões de simplicidade, nessa descrição será utilizado um sistema de avaliação de comportamento simplificado, que pode ser substituído por outros sistemas de avaliação para dar maior robustez.

Sempre que um nó observa uma ação maliciosa, ele deve contatar as testemunhas do nó, enviando a denúncia. As testemunhas, ao receberem uma denúncia, devem verificar a assinatura e o certificado recebidos e, caso sejam válidos, devem armazenar essa informação. No sistema de avaliação simplificado, cada testemunha aceita apenas uma denúncia de cada nó durante um período t_{den}. Isso evita que as testemunhas sejam sobrecarregadas com um excesso de mensagens de denúncia, além de evitar a sobrecarga no armazenamento das informações. Supondo que todas as denúncias tenham um peso igual, a reputação do nó (R_i) pode ser atualizada a cada denúncia por

$$R_i = R_{i-1} - 1. \tag{3.4}$$

Após um período t_a sem denúncias, a reputação é atualizada para

$$R_i = R_{i-1} + 1. \tag{3.5}$$

A variável R_i é limitada por 0 e R_{max}, não devendo ultrapassar esses valores. Se a reputação alcançar um limiar L, a testemunha deve enviar para a rede um voto de exclusão do nó. Se até k testemunhas enviarem esse voto, o nó deverá ser excluído, o endereço liberado e o certificado deverá entrar para a lista de certificados revogados. As testemunhas devem guardar a informação de certificado revogado enquanto a validade do certificado não vencer.

O valor de L indica a rigidez com a qual o nó está sendo avaliado. Um L grande evita que nós bem comportados sejam excluídos injustamente, mas é menos eficiente na punição dos nós maliciosos. Por outro lado, um L pequeno exclui os nós mal comportados, mas acaba punindo injustamente nós bem comportados. O valor de L também pode ser usado para aumentar a rigidez na avaliação dos nós com relação à exclusão dos filhos.

Sempre que um nó filho for excluído, L deve ser atualizado por

$$L = L - \frac{t}{nc \cdot N}, \tag{3.6}$$

onde t é o número de filhos que o nó pode inserir na rede, N é o número máximo de nós na rede e nc é o número de níveis na cadeia de confiança entre o nó e o descendente excluído. Se o número de filhos for ilimitado t vale N. Dessa forma, o nó que tem direito a inserir mais filhos recebe uma punição maior por cada filho malicioso. O uso de um L variável gera uma maior robustez contra nós maliciosos que criam identidades para realizar as ações maliciosas. Assim, quando um filho é expulso, o pai também é punido.

De acordo com as Equações 3.4 e 3.5, a reputação de cada nó é formada de forma recursiva. Assim, é necessário que cada nó possua uma reputação inicial. O nó raiz da árvore, por possuir a confiança de todos os outros nós, deve iniciar com confiança igual a 1. Os filhos, por sua vez, devem herdar a reputação atual do pai, assim como o seu L.

3.3.6 Formação de Partições

Como já foi discutido, uma das características das redes ad hoc é a existência de freqüentes partições na rede. Esse problema foi tratado na distribuição de endereço através da detecção de união de partições e troca dos endereços colididos. Os endereços, no entanto, correspondem à identificação do nó, e, assim, ao trocar de endereço, o certificado de um nó deixa de ser válido. Dessa forma, a união de partições também deve ser tratada de forma especial no sistema de autenticação proposto.

Primeiramente, a troca de endereços implica na troca das testemunhas. Isso significa que toda vez que acontecer uma união de partição e o nó precisar trocar de endereço, ele deve encontrar um par de chaves cujos primeiros p bits não estejam alocados como endereço de nenhum outro nó e notificar as testemunhas do endereço antigo o novo endereço. As testemunhas deverão, então, contatar automaticamente as testemunhas correspondentes ao novo IP, repassando os dados armazenados sobre o nó.

O segundo problema que pode ocorrer durante a união de partições é que as testemunhas sejam separadas do seu nó observado. Nesse caso, as testemunhas que forem deslocadas para outra partição aonde o nó observado não está presente acabarão considerando

o nó como ausente e apagarão todos os dados, exceto listas de revogação e autorizações, referentes ao nó. Da mesma forma, o nó na outra partição perceberá a ausência das testemunhas, alocando novos nós para essa atividade. Caso o nó seja separado de todas as suas testemunhas, novas testemunhas devem ser alocadas como se o nó estivesse entrando na rede naquele momento.

Após uma união de partições, as testemunhas dos nós que estavam em partições diferentes devem trocar informações observadas nas suas partições. Isso é importante para que informações de exclusão não sejam perdidas devido às partições. Além disso, as testemunhas de usuário devem manter o conjunto de testemunhas ativas autonomamente, independente do usuário, para que as informações de autorizações emitidas e revogadas não sejam perdidas.

3.3.7 Inicialização da Rede

A inicialização da rede pode ocorrer com vários nós entrando simultaneamente ou com os nós entrando gradativamente. Em ambos os casos, não há como garantir que $n \geqslant (k+1)$, onde n é o número de nós na rede e k é o número mínimo de testemunhas para emitir um certificado. Se $n < (k+1)$, não há como formar o conjunto de testemunhas. Assim, nesses casos, que podem ser detectados pelo filtro de endereços, todos os nós devem operar como testemunhas e os certificados podem ser validados mesmo com um número de assinaturas inferior a k.

Outra restrição com relação à inicialização é que o usuário raiz da cadeia de confiança ou os seus filhos imediatos precisam estar na rede no momento da inicialização. Um usuário não pode entrar na rede enquanto o seu usuário pai não tiver entrado, pois a entrada exige a verificação das testemunhas do pai. Como todos os nós conhecem a chave pública da raiz, todos os nós no nível 1 da cadeia de confiança podem validar a autorização dada pela raiz com qualquer nó da cadeia. As demais autorizações precisarão ser validadas gradativamente, com a entrada dos nós filhos e seus descendentes. Cabe observar que basta que o pai tenha entrado em algum momento na rede para que seus filhos possam também entrar. As testemunhas de usuários devem guardar os dados referentes ao nó

monitorado mesmo que ele não esteja na rede, permitindo que os filhos acessem a rede na ausência do pai.

3.4 Análise do Sistema Proposto

Nessa seção são analisados os possíveis casos de tentativa de falsificação de certificado e o impacto da variação de alguns parâmetros do AMORA. O número de testemunhas de cada nó (m) e o limiar de votação (k), que é o número de testemunhas necessárias para certificar um nó, são parâmetros que interferem no desempenho do sistema de autenticação e monitoramento. O número de testemunhas e o limiar de votação determinam, entre outros fatores, a robustez do protocolo a ataques em conluio, a carga de armazenamento em cada nó e a probabilidade de uma partição ficar sem nenhuma testemunha de um determinado usuário. Outro ponto discutido nessa seção são as cadeias de confiança, que representam o ponto central do sistema de controle de acesso.

3.4.1 Robustez contra conluio na Votação para Prejudicar Nó Não-Malicioso

O AMORA emite/valida certificados e exclui nós com base na votação das testemunhas. Se em uma rede existem muitos nós maliciosos, o resultado da votação pode ser influenciado, caso os nós maliciosos se unam para enviar votos falsos que comprometam um determinado nó. Assim, os parâmetros relativos às testemunhas devem ser escolhidos de forma a minimizar a chance de sucesso em um conluio na votação.

O limiar de votação corresponde ao parâmetro k, que determina o número mínimo de testemunhas para validar uma requisição. A utilização de um valor de k grande permite uma maior robustez ao conluio nas votações, pois é necessário pelo menos k nós comprometidos para modificar o resultado de uma votação. Por outro lado, esse parâmetro interfere na disponibilidade do serviço na rede. Se o k é muito grande, pode-se tornar difícil manter o conjunto de testemunhas ativas necessárias para se emitir um certificado. Além disso, um valor de k muito alto aumenta o número de mensagens trocadas para se

obter ou validar um certificado. Dessa forma, k deve ser escolhido como o menor valor que garanta uma probabilidade de sucesso de conluio pequena durante as votações.

A probabilidade de sucesso de um conluio na votação ($P(E_{sc})$) para prejudicar um nó não-malicioso pode ser calculada com base no número de nós na rede (N), no número de nós comprometidos na rede (M), no número de testemunhas por nó (m) e em k. Supondo que o conluio seja feito para impedir a validação de um certificado legítimo e que $N > m+1$, a $P(E_{sc})$ pode ser calculada com base na combinação de conjuntos de testemunhas com pelo menos k nós maliciosos e é dada por:

$se\ M < k$:

$$P(E_{sc}) \quad = \quad 0; \tag{3.7}$$

$se\ k \leq M \leq m\ e\ (N-1-M) \geqslant (m-k)$:

$$P(E_{sc}) \quad = \quad \frac{C_k^M \cdot C_{m-k}^{N-1-M} + C_{k+1}^M \cdot C_{m-k-1}^{N-1-M} \ \cdots \ C_{k+a}^M \cdot C_{m-k-a}^{N-1-M}}{C_m^{N-1}}, \quad \forall a | a \in Z\ e\ (k+a) \leq M; \tag{3.8}$$

$se\ k \leq M\ e\ (N-1-M) < (m-k)$:

$$P(E_{sc}) \quad = \quad \frac{C_{k+b}^M \cdot C_{m-k-b}^{N-1-M} + C_{k+b+1}^M \cdot C_{m-k-b-1}^{N-1-M} \ \cdots \ C_{k+a}^M \cdot C_{m-k-a}^{N-1-M}}{C_m^{N-1}}, \quad \forall a, b | a \in Z,\ b \in Z,\ (k+a) \leq M,\ (k+a) \leqslant m, \quad (m-k-b) \leq (N-1-M)\ e\ a > b; \tag{3.9}$$

$se\ m < M\ e\ (N-1-M) \geqslant (m-k)$:

$$P(E_{sc}) \quad = \quad \frac{C_k^M \cdot C_{m-k}^{N-1-M} + C_{k+1}^M \cdot C_{m-k-1}^{N-1-M} \ \cdots \ C_m^M \cdot C_{m-m}^{N-1-M}}{C_m^{N-1}}. \tag{3.10}$$

O gráfico correspondente a Equação 3.10, que supõe que o número de nós maliciosos é maior que m, está representado na Figura 3.13, onde $N = 50$ e $m = 7$. Por esse

gráfico é possível observar que com $k = 7$, mesmo com metade da rede comprometida, ou seja, que metade da rede seja composta por nós maliciosos dispostos a enviar votos falsos nas votações, a probabilidade de sucesso de um conluio na votação para validação de requisições é de $0,005$, ou seja, com um valor adequado de k pode-se obter um sistema robusto. Por outro lado, o valor do número total de testemunhas para cada nó também influencia esse resultado, pois, com o aumento de m, a probabilidade de sucesso de conluio aumenta, como pode se observar na Figura 3.14, onde $N = 50$ e $m = 14$. De fato, a probabilidade de existirem pelo menos k nós maliciosos em um conjunto de testemunhas aumenta se o conjunto de testemunhas também aumentar. Dessa forma, é preciso haver um equilíbrio entre m e k para que o AMORA possa permanecer robusto mesmo com muitos nós maliciosos na rede que tentam prejudicar nós não-maliciosos.

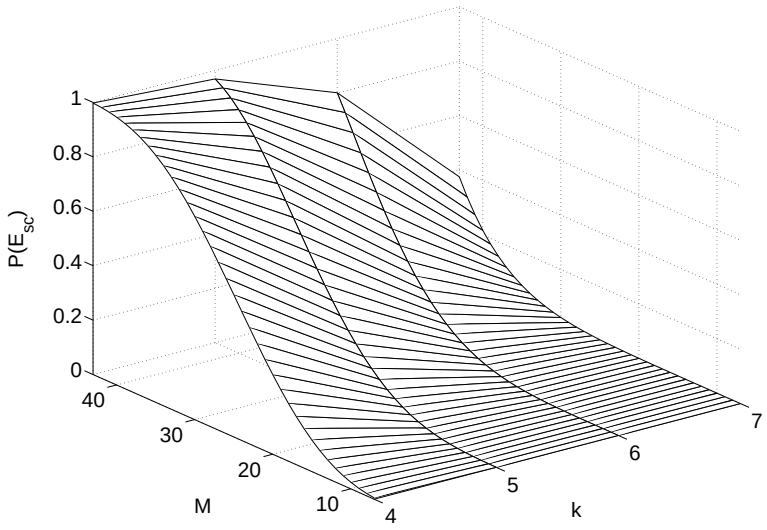


Figura 3.13: Probabilidade de sucesso de um conluio com $m = 7$.

3.4.2 Carga de Armazenamento para Monitoração dos Nós

Os parâmetros do AMORA devem ser escolhidos de forma a obter uma robustez ao conluio na votação e a não sobrecarregar os nós. É importante evitar a sobrecarga de dados de monitoramento, pois muitos dispositivos sem fio possuem restrições de armazenamento. Dessa forma, é importante reduzir ao máximo o número de nós monitorados por cada nó e a carga de dados por nó monitorado.

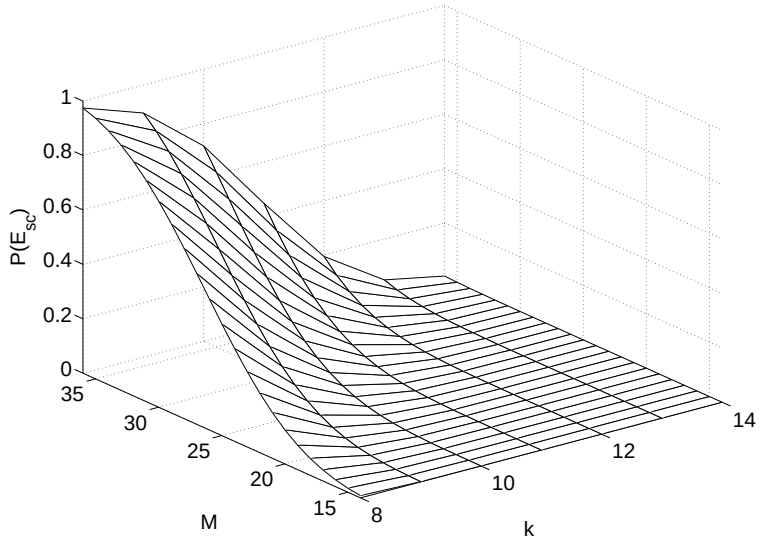


Figura 3.14: Probabilidade de sucesso de um conluio com $m = 14$.

A quantidade de nós selecionando um determinado nó como testemunha depende do valor de m e influencia diretamente o volume de dados que cada nó deverá armazenar para que o sistema de monitoração permaneça funcional. Supondo que todos os nós tenham igual probabilidade de serem escolhidos como testemunha e que N é o número de nós na rede, pode-se afirmar que o número médio de nós monitorados por cada nó é dado por

$$N_m = \frac{m \cdot N}{N} = m. \tag{3.11}$$

Portanto, o aumento de m, além de levar a uma maior propensão a ataques em conluio, implica em uma maior carga de armazenamento em cada nó, pois quanto mais nós são monitorados, mais dados precisam ser guardados. Dessa forma, a carga total média (C_m) que deve ser armazenada em média por cada nó no AMORA vale

$$C_m = m_n \cdot C_s + m_u \cdot C_a, \tag{3.12}$$

onde m_n é o número de testemunhas por nó, m_u é o número de testemunhas por usuário, C_s é a carga imposta pelo sistema de confiança por cada nó monitorado e C_a é a carga devido ao armazenamento da autorização, da lista de identificações de autorizações concedidas para filhos e da lista de IPs das outras testemunhas de usuário do nó monitorado.

3.4.3 Formação de Partições

No AMORA, é importante que em todas as partições sempre existam nós testemunhas de usuário de todos os usuários, pois, caso as testemunhas de usuário não estejam presentes, nenhum nó filho poderá se unir àquela partição. No caso das testemunhas de nó, a formação de partições também pode trazer impactos, pois se um nó é separado de suas testemunhas na formação da partição, toda a sua reputação é perdida, pois as novas testemunhas não terão acesso ao histórico de monitoração.

O valor de m influencia a probabilidade de que, após uma partição, todos os nós permaneçam com testemunhas ativas nas duas partições. Supondo que o número de nós na partição formada seja P, o número de nós na rede seja N e que $N \geqslant m + 1$ pode-se afirmar que a probabilidade da partição não possuir testemunhas de um determinado nó que está na partição ($P(E_p)$) é dada por:

$se\ (N - m) \geqslant P$

$$P(E_p) \quad = \quad \frac{C_{P-1}^{(N-m-1)}}{C_{P-1}^{N-1}} = \frac{(N-m-1)!(N-P)!}{(N-m-P)!(N-1)!}, \tag{3.13}$$

$se\ (N - m) < P$

$$P(E_p) \quad = \quad 0. \tag{3.14}$$

A Figura 3.15 mostra a probabilidade de o nó não estar presente na partição em função da relação entre P e m, supondo que $N = 50$. O que se observa, como esperado, é que para valores maiores de m, essa probabilidade cai. A partir desse resultado, pode-se concluir que, como é desejável que as testemunhas do usuário estejam presentes sempre para controlar a entrada dos nós filhos, o m_u deve ser grande. Por outro lado, como as testemunhas do nó podem ser perdidas com a formação de partições sem grandes prejuízos para a rede, o m_n deve ser pequeno, evitando a sobrecarga de armazenamento e dificultando conluios.

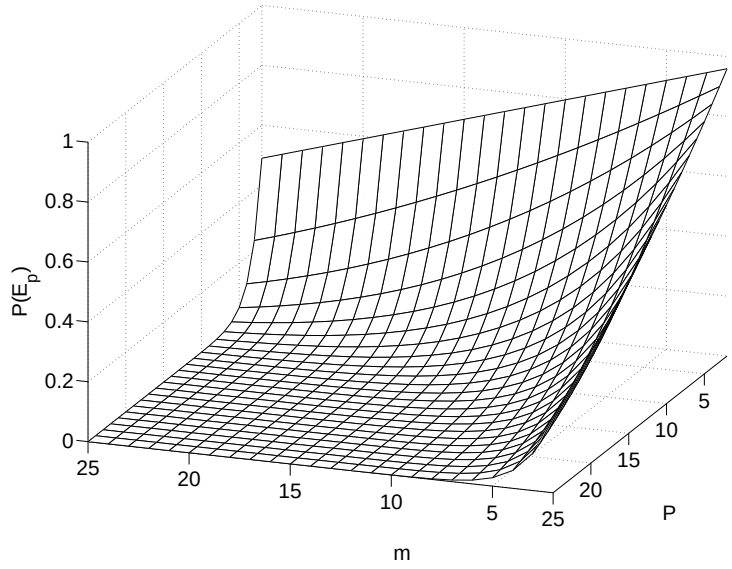


Figura 3.15: Probabilidade de perda de testemunhas após formação de partição.

3.4.4 Cadeia de Confiança

Nas propostas baseadas em cadeias de certificados para realizar o controle de acesso em redes ad hoc, um nó precisa verificar todos os saltos da cadeia para verificar se um determinado nó é confiável. Esse é o caso da proposta de Capkun *et al.* [43, 44], na qual os nós precisam estabelecer um repositório de certificados para determinar se um nó é confiável. No AMORA, cada testemunha verifica apenas o usuário pai para emitir um certificado, pois se supõe que, se um usuário já possui um certificado válido, ele já conseguiu provar anteriormente que seu pai também é parte da cadeia. Esse tipo de relação é importante para que a entrada de novos nós não seja impedida quando todos os seus antecessores na cadeia não estiverem presentes na rede. Por outro lado, basta que um usuário ilegítimo consiga acesso à rede através de um certificado falsificado para comprometer toda a rede. A seguir, será analisada a probabilidade de sucesso na falsificação de um certificado que leve à obtenção dos direitos de acesso. Essa análise difere da probabilidade de sucesso de conluio na votação (Seção 3.4.1), pois aqui não existem nós autorizados cooperando. De fato, está sendo analisada a probabilidade de um nó conseguir entrar na cadeia sem ajuda de nenhum nó autorizado.

Uma vez que a verificação não é realizada até a raiz da cadeia de certificação, a pro-

babilidade de falsificar um certificado é avaliada de acordo com a probabilidade de um nó forjar todas as suas testemunhas. Assim, falsificar um certificado corresponde a encontrar um par de chaves correspondente ao IP de cada testemunha, gerar as assinaturas das testemunhas e garantir, que durante a consulta a lista de certificados revogados para validação do certificado, nenhuma das testemunhas reais utilizadas irá responder.

Primeiramente, avaliou-se a probabilidade de um par de chaves assimétricas sorteado corresponder a identificação de uma determinada testemunha. Supondo que a probabilidade de um bit da chave pública ser igual a 1 seja igual à probabilidade do bit ser igual 0, a probabilidade de uma chave sorteada possuir os mesmos primeiros bits da identificação da testemunha que se deseja forjar (P_{ID}) é

$$P_{ID} = \left(\frac{1}{2}\right)^p, \tag{3.15}$$

onde p é o número de bits no endereço que identificam uma máquina. Pode-se deduzir também que o número médio de sorteios até encontrar as k testemunhas vale

$$\begin{aligned} N_s &= \frac{1}{m\left(\frac{1}{2}\right)^p} + \frac{1}{(m-1)\left(\frac{1}{2}\right)^p} \cdots \frac{1}{(m-k+1)\left(\frac{1}{2}\right)^p} \\ &= 2^p \left(\frac{1}{m} + \frac{1}{m-1} \cdots \frac{1}{m-k+1}\right) \\ &= 2^p \sum_{i=0}^{k-1} \left(\frac{1}{m-i}\right) \end{aligned} \tag{3.16}$$

$$\tag{3.17}$$

A Equação 3.16, supondo $m = 2 \cdot k - 1$, está representada na Figura 3.16, que mostra que o número de tentativas chega a ordem de 10^7 para um sufixo de máquina (p) com 25 bits.

O tempo médio para encontrar as testemunhas vale

$$T_t = T_g \cdot N_s, \tag{3.18}$$

onde T_g é o tempo médio para gerar 1 par de chaves.

Para avaliar a dificuldade de gerar o certificado com testemunhas falsas, fez-se um teste do tempo médio para gerar um certificado falso. Assim, utilizou-se o OpenSSL0.9.8g para gerar pares de chaves assimétricas em uma máquina bi-processada com $2,6$GHz de *clock* e 2GB de memória usando sistema operacional Debian. No teste, realizou-se 10

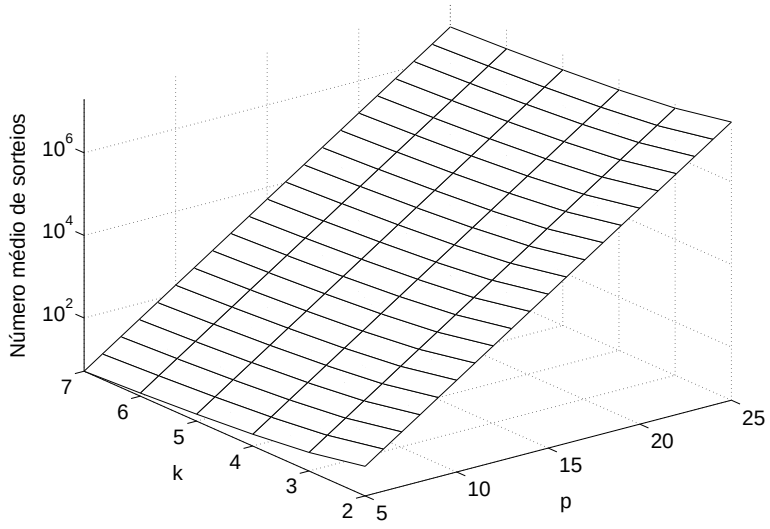


Figura 3.16: Numero médio de tentativas para forjar as k testemunhas.

rodadas de 1000 repetições de geração de chave e mediu-se o tempo com o comando *time*, obtendo-se que o tempo médio para gerar um par de chaves vale $0,031$s. Dessa forma, supondo-se que $k = 3$ e $p = 16$, o tempo médio para obter chaves públicas correspondentes às k testemunhas seria de aproximadamente 21 minutos. Por outro lado, se $p = 8$, o tempo cai para aproximadamente 5 segundos. Dessa forma, a proteção contra entrada não autorizada do protocolo não pode depender exclusivamente da dificuldade em se encontrar as chaves das testemunhas.

Supondo que o atacante consegue gerar sem dificuldades chaves correspondentes às testemunhas, ainda assim ele não conseguiria validar um certificado. Quando um pedido de validação de certificado for feito, tanto as testemunhas falsas quanto as reais responderão, embora com respostas diferentes para o pedido. Com uma verificação dos certificados apresentados pelas testemunhas, seria possível afirmar quais são as testemunhas reais e concluir que o certificado do nó é falso. Dessa forma, para o ataque ser funcional, é preciso que todas as testemunhas estejam presentes no Filtro de endereços, que mostra todos os nós ativos na rede, mas estando, de fato, ausentes da rede. Assim, a segurança da cadeia do AMORA leva em consideração não apenas o tempo para gerar k testemunhas, mas também o tempo para gerar uma identidade que tenha pelo menos k testemunhas ausentes com ausência não identificada no filtro de endereço e o tempo para a detecção de

uma ausência. Para analisar a relação entre essas três variáveis, inicialmente, calculou-se a probabilidade de uma ausência não ser detectada. A maior chance de ter todas as testemunhas ausentes sem que a ausência tenha sido detectada é durante a formação de uma partição. Quando as partições se formam, o filtro de endereço deve ser atualizado e existe uma probabilidade de uma ausência não ser detectada ($P(E_a)$), que é dada por

$$P(E_a) = P(E_{t_u}) \cdot P(E_{t_n}) \cdot P(E_m) \cdot P(E_p), \tag{3.19}$$

onde $P(E_{t_u})$ corresponde à probabilidade de um nó não ter testemunhas de usuário após uma formação de partição, $P(E_{t_n})$ corresponde à probabilidade de um nó não ter testemunhas de nó após uma formação de partição, $P(E_m)$ representa a probabilidade de o nó não estar monitorando nenhum nó na partição e $P(E_p)$ é a probabilidade de o nó ficar na outra partição. As duas primeiras parcelas da equação são dadas por

$se\ (N - m) \geqslant P$:

$$P(E_{t_u}) \quad = \quad \frac{C_P^{(N-m_u-1)}}{C_P^{N-1}} = \frac{\prod_{b=1+P}^{m_u+P}(N-b)}{\prod_{a=1}^{m_u}(N-a)}, \tag{3.20}$$

$se\ (N - m) < P$:

$$P(E_{t_u}) \quad = \quad 0. \tag{3.21}$$

e

$se\ (N - m) \geqslant P$:

$$P(E_{t_n}) = \frac{C_P^{(N-m_n-1)}}{C_P^{N-1}} = \frac{\prod_{b=1+P}^{m_n+P}(N-b)}{\prod_{a=1}^{m_n}(N-a)}, \tag{3.22}$$

$se\ (N - m) < P$:

$$P(E_{t_n}) \quad = \quad 0. \tag{3.23}$$

A probabilidade de o nó ficar na outra partição é dada por

$$P(E_p) = 1 - \left(\frac{P}{N}\right). \tag{3.24}$$

Por fim, $P(E_m)$, que é a probabilidade de o nó não estar monitorando nenhum nó na partição, é composta por duas parcelas, sendo elas a probabilidade de nenhum nó que estava sendo monitorado ficar na partição analisada ($P(E_{tp})$) e a probabilidade de que após a formação da partição o nó não seja selecionado por nenhum outro como testemunha ($P(E_{ts})$). Supondo que antes da formação da partição, o nó monitorasse m nós (Equação 3.11), o que corresponde à média de nós monitorados na rede, pode-se afirmar que:

$se\ (N-m) \geqslant P$:

$$\begin{aligned} P(E_{tp}) &= \frac{C_P^{(N-m_u-1)}}{C_P^{N-1}} \cdot \frac{C_P^{(N-m_n-1)}}{C_P^{N-1}} \\ &= \frac{\prod_{b=1+P}^{m_u+P}(N-b)}{\prod_{a=1}^{m_u}(N-a)} \cdot \frac{\prod_{b=1+P}^{m_n+P}(P-b)}{\prod_{a=1}^{m_n}(N-a)}, \end{aligned} \tag{3.25}$$

$se\ (N-m) < P$:

$$P(E_{tp}) = 0, \tag{3.26}$$

onde P é o número de nós na partição e N é o número de nós na rede. Supondo que $P(E_s)$ é a probabilidade do nó A ser selecionado pelo nó B como testemunha, dado que A está no filtro de endereços, mas, de fato, está ausente da partição e B está presente na partição e é dada por

$$P(E_s) = \frac{C_{m-1}^{P-1}}{C_m^P} = \frac{m}{P}. \tag{3.27}$$

A probabilidade de um nó não ser escolhido como testemunha, $P(E_{ts})$, é dada por

$se\ P \geqslant (m+1)$

$$\begin{aligned} P(E_{ts}) &= (1 - P(E_s))^P \\ &= \left(1 - \frac{m}{P}\right)^P, \end{aligned} \tag{3.28}$$

$se\ P < (m+1)$:

$$P(E_{ts}) = 0, \tag{3.29}$$

Supondo que m para as testemunhas de usuário vale 5, que m para as testemunhas do nó vale 3, que N vale 50 e $P = N/7 \approx 7$, a probabilidade de uma ausência não ser detectada é da ordem de 10^{-8}, o que é um valor que pode ser considerado desprezível.

Considerando que todas as ausências serão detectadas, o tempo máximo até a determinação da ausência é dado pelo valor do intervalo entre os pacotes de teste, T_{pt}. Os pacotes de teste permitem que os nós monitorados e as suas testemunhas verifiquem mutuamente a presença na rede. Uma vez que T_{pt} é um parâmetro do sistema, este valor deve ser determinado de acordo com o número de bits para identificar cada máquina (p), de forma que a ausência dos nós seja sempre detectada antes que um nó malicioso possa tentar forjar um certificado utilizando testemunhas falsas para substituir as testemunhas reais que estão ausentes, mas cuja ausência ainda não foi detectada.

Com base no valor escolhido para m e k, o intervalo entre pacotes de teste, T_{pt}, pode ser determinado como o maior valor que atenda a inequação

$$T_t > T_{pt} \tag{3.30}$$

onde T_t é o tempo médio para encontrar as chaves das testemunhas falsas. Substituindo os valores, obtém-se que

$$T_g \cdot 2^p \left(\frac{1}{m} + \frac{1}{m-1} \cdots \frac{1}{m-k+1}\right) > T_{pt} \tag{3.31}$$

onde e T_g é o tempo para gerar um par de chaves assimétricas.

Um estudo de caso mais detalhado sobre as tentativas de forjar o certificado e de manipular o sistema de certificação e monitoração levando em consideração os dados apresentados será mostrado a seguir. Nessa análise também será considerado o caso pouco provável de o nó conseguir forjar o seu certificado enquanto a ausência das testemunhas não é identificada.

Entrada de Nós

Para que um novo nó forje um certificado durante a entrada na rede, é preciso que as testemunhas estejam presentes no filtro de endereços, mas inativas. O nó precisa das testemunhas para anunciar o endereço alocado para a rede. Se as testemunhas não estão no filtro, todos os nós descartarão o anúncio de entrada, pois verificarão que essas não são as testemunhas corretas do nó. Se as testemunhas estão presentes na rede e o nó envia anúncios forjados para a rede, todas as testemunhas detectarão a ação maliciosa e poderão impedir o uso do certificado falso. Além disso, mesmo que o nó não alocasse o endereço no Filtro, seu certificado nunca poderia ser utilizado, já que as testemunhas não validariam algo que não foram elas que emitiram e a lista de revogação precisa ser consultada sempre que um nó recebe um novo certificado. Assim, o nó malicioso precisaria que as suas testemunhas estivessem listadas no Filtro de endereços como ativas, embora estivessem inativas, para poder forjar e validar o seu certificado. Uma vez que a probabilidade de isso ocorrer é desprezível, em especial em situações onde não aconteceram partições, não é possível forjar o certificado para permitir a entrada de um novo nó.

Considerando a possibilidade de as testemunhas estarem ausentes com a ausência ainda não detectada e de o nó conseguir entrar na rede antes de as ausências das testemunhas forjadas serem detectadas, ainda assim o nó não manteria esse acesso por muito tempo, pois, após a detecção da ausência, que ocorre em no máximo T_{pt}, o nó teria que trocar de testemunhas e as novas testemunhas não aceitariam uma autorização falsa. Se o nó tivesse inserido algum filho antes da troca para as testemunhas reais, esse filho também seria excluído, pois a exclusão de um nó implica na exclusão de todos os seus descendentes.

Manutenção de Testemunhas

Cada nó é responsável por manter o seu conjunto de testemunhas. Se o nó malicioso optar por não manter o seu conjunto de testemunhas atualizado para não ser penalizado com as monitorações, o seu certificado perderá a validade, já que cada nó, ao receber um certificado, deve verificar a lista de revogação. Como a lista de revogação fica com as testemunhas, o nó que recebeu o certificado deve buscar cada uma das testemunhas listadas no certificado para verificar se elas validam as suas assinaturas. Se as testemunhas estiverem ausentes, o certificado não poderá ser validado, e o nó malicioso não poderá utilizar nenhum serviço da rede. Caso o nó malicioso opte por atualizar suas testemunhas apenas quando for emitir um certificado, para evitar que suas ações maliciosas sejam monitoradas, essa ação maliciosa também poderá ser detectada. Como os nós que vão denunciar ações maliciosas devem contatar todas as testemunhas reais, as testemunhas não contatadas detectarão a ação maliciosa.

Falsificação de Autorização

Outra possibilidade para tentar entrar na rede sem autorização é falsificar uma autorização. Se o nó malicioso que deseja entrar na rede falsificar a autorização, assinando-a com uma chave que não seja a do usuário pai, a ação seria facilmente detectada pelas testemunhas do usuário pai, que não fariam a validação para as testemunhas do usuário filho. Outra possibilidade seria utilizar várias vezes a mesma autorização para obter diferentes identidades. Novamente, o ataque não teria efeito, pois as testemunhas de usuário podem identificar quais autorizações estão sendo utilizadas e por quais nós. Caso as testemunhas do usuário pai tenham ficado todas separadas em uma partição diferente, o ataque também não teria efeito, pois o novo nó não poderá entrar sem contatar as testemunhas do pai.

Personificação de Outro Nó

Uma estratégia do nó malicioso para utilizar a rede poderia ser utilizar a identidade de algum nó que esteja presente na rede, falsificando um par de chaves que corresponda àquela identificação e um par de chaves para cada uma das k testemunhas corresponden-

tes. Novamente esse ataque não teria efeito devido à necessidade de consulta à lista de revogação.

O ataque seria iniciado quando o nó repassasse o seu certificado forjado para um nó qualquer da rede. Esse nó, ao receber o certificado, faria um pedido de validação para as testemunhas. Uma vez que o nó malicioso possui chaves falsas correspondentes às suas testemunhas, poderia responder ao pedido de validação se passando também pelas testemunhas. No entanto, ao receber as respostas da validação, o nó que recebeu o certificado falso verificaria que recebeu duas respostas diferentes de cada uma das k testemunhas. Assim, o nó deve requisitar os certificados para verificar qual é a verdadeira testemunha, impedindo o nó malicioso de utilizar uma identidade falsa.

Outra estratégia do nó malicioso poderia ser tentar se passar por um nó que acabou de sair, enviando os pacotes de teste para as testemunhas. Esse ataque, no entanto, também não funcionaria, porque os pacotes de teste também são assinados e as testemunhas notariam que o par de chaves foi modificado.

3.4.5 Resultados da Análise

Pelos dados obtidos com o estudo dos parâmetros do AMORA e de alguns casos de tentativa de forjar certificados, é possível observar que o AMORA é um sistema robusto contra a entrada de nós maliciosos. Foi mostrado na análise que, mesmo em situações onde metade da rede está comprometida, o sistema ainda tem baixa probabilidade de sofrer um ataque. Essa característica é importante para garantir o sucesso do controle de acesso nas redes ad hoc e também para dar confiabilidade à autoridade certificadora utilizada.

Além da alta robustez contra ataques para a geração e validação de certificados falsos e ataques para invalidar certificados verdadeiros, o AMORA permite também adaptar a quantidade de dados que cada nó deve guardar de acordo com a aplicação, através da variação do valor de m. A possibilidade de variar os parâmetros do sistema permite uma maior adaptabilidade do AMORA para diferentes cenários, sob o custo de aumentar ou diminuir a robustez do sistema contra invasões.

3.5 Considerações Finais

As redes ad hoc seguras dependem de um controle de acesso distribuído, por meio de autenticação e monitoração dos nós. Os sistemas de autenticação propostos na literatura, apesar de realizarem o controle de acesso, não atendem a todas as características das redes ad hoc. Muitos sistemas contam com uma pré-inicialização da rede, na qual um administrador conhece todas as identidades e todos os nós que desejam acessar a rede. Assim, o sistema de autenticação é pré-inicializado e, a partir da inicialização da rede, todos os nós previamente cadastrados podem utilizar o sistema. Esse é o caso das autoridades certificadoras de limiar e dos sistemas de gerenciamento de chaves baseados em pré-distribuição. Essas características de necessidade de administrador e de identidades autorizadas antes da inicialização da rede não são adequadas a redes auto-organizáveis, restringindo muito o uso das redes ad hoc. Os sistemas baseados em cadeias de confiança mostram uma nova perspectiva de uma rede totalmente auto-configurável. Essa estratégia, no entanto, é pouco robusta por não ser capaz de impedir a volta de usuários maliciosos à rede, já que não existe nenhum sistema que restrinja a entrada de usuários com base em relações pré-estabelecidas de segurança.

O sistema de autenticação e monitoração proposto traz uma nova estratégia para o controle de acesso em redes ad hoc, permitindo que o administrador passe a ser uma entidade distribuída e evitando a necessidade do conhecimento das identidades de todos os usuários antes da inicialização da rede. Além disso, o sistema impede a volta dos usuários maliciosos por meio das autorizações que limitam quantos filhos cada nó pode ter e pela punição de usuários pais cujos filhos executam ações maliciosas. Assim, a cadeia de confiança estabelecida também implica em uma cadeia de responsabilização, onde usuários que permitem a entrada de nós não-confiáveis também são excluídos por tornar a rede vulnerável. Outro ponto interessante é que a proposta permite a criação de um sistema de certificação sem que seja necessário percorrer toda a cadeia de confiança para emitir e validar certificados.

O sistema proposto se adequa a outros sistemas de confiança previstos na literatura, embora possa funcionar com um módulo de confiança básico descrito nesse capítulo. Assim, a proposta é adaptável a outros sistemas já existentes e pode ter sua robustez

aumentada com a utilização de sistemas de confiança mais complexos, que só devem ser utilizados com nós com maior capacidade de processamento e armazenamento.

Devido às características apresentadas e à análise de segurança realizada, é possível afirmar que o sistema proposto é robusto e auto-configurável, dependendo apenas na estabilização *offline* da cadeia de confiança entre os usuários. Essa não é uma premissa forte, já que, quando se fala de redes ad hoc com controle de acesso, se espera que exista um grupo de usuários que desejam restringir a entrada de outros, seja por questões de privacidade, de utilização de recursos da rede, ou, ainda, por não confiarem nesses usuários. Por essas razões, é possível afirmar que o sistema proposto se adequa às características das redes ad hoc e é uma alternativa viável para muitas aplicações que exigem segurança.

Capítulo 4

Distribuição de Chaves Simétricas

AS redes ad hoc são baseadas em roteamento colaborativo, o que significa que nós atuando de forma maliciosa podem comprometer toda a rede. Por essa razão, muitos protocolos foram propostos para prover segurança no roteamento [15, 58–61] e encaminhamento de dados [62]. O *Secure Optimized Link State Routing protocol* (SOLSR) [63] é um desses protocolos seguros, desenvolvido com base no *Optimized Link State Routing protocol* (OLSR) [64]. O SOLSR utiliza chaves de grupo para identificar os nós que têm direito de acesso à rede. A chave de grupo é uma chave secreta cuja cópia é entregue a cada um dos membros do grupo. Dessa forma, toda mensagem de roteamento que não é assinada com a chave secreta compartilhada pelo grupo de nós autorizados é descartada. O SOLSR, assim como os demais protocolos baseados em criptografia simétrica, supõe a existência de um sistema para gerenciar as chaves de grupo.

O gerenciamento de chaves secretas possui desafios semelhantes ao gerenciamento de chaves públicas e privadas nas redes ad hoc, pois não é possível garantir que algum recurso específico, tal como um servidor de distribuição de chaves, estará disponível para todos os nós em todos os momentos [65]. Portanto, o gerenciamento de chaves nas redes ad hoc não pode ser baseado em infra-estrutura fixa e centralizada, como o Kerberos [66] ou a Infra-estrutura de Chaves Públicas (*Public Key Infrastructure* - PKI) [67], que são convencionalmente usados nas redes cabeadas. Outra questão é que as redes ad hoc, em geral, são compostas por dispositivos com restrições de bateria. Assim, a segurança deve ser provida sem gerar um grande consumo de energia, pois alguns nós podem não ser

capazes de executar operações criptográficas freqüentes.

Nesse capítulo é especificado um protocolo para gerenciamento de chaves de grupo para ambientes ad hoc que utilizam o SOLSR ou algum protocolo de roteamento seguro similar. Com este protocolo de gerenciamento de chaves associado aos mecanismos propostos nos capítulo anteriores, o ambiente seguro em redes ad hoc passa a atender tanto as aplicações que utilizam criptografia assimétrica quanto simétrica, pois todos os nós são capazes de obter o material criptográfico específico da aplicação desejada, são identificados de forma única, e são monitorados, para que todos os nós possam confiar na cooperatividade dentro da rede.

O protocolo proposto para distribuição de chaves simétricas usa um pequeno número de mensagens durante a distribuição das chaves para reduzir o consumo de energia e é chamado de CHAve de grupo no Roteamento Através de Distribuição Assimétrica Segura (CHARADAS) [65]. O CHARADAS foi desenvolvido para a utilização no roteamento com o SOLSR, embora possa ser estendido, através de algumas modificações, para qualquer outra aplicação de chaves de grupo. O protocolo proposto é composto por três mecanismos principais: distribuição de chave de grupo; união de partições e entrada de nós; e, por fim, detecção de falha e substituição de líderes de rodada. Como foi explicado anteriormente, a vantagem da chave simétrica de grupo é o baixo custo computacional se comparada à criptografia assimétrica. No entanto, o inconveniente é o comprometimento de todo o grupo a partir da revelação da chave que pode ser obtida com a violação de apenas um nó. Portanto, no CHARADAS, a chave de grupo é periodicamente renovada com o objetivo de excluir nós não-autorizados. Além disso, evita-se, com a troca periódica, que a mesma chave de grupo seja utilizada em mais do que certa quantidade de dados, especialmente quando técnicas criptográficas fracas estão sendo utilizadas. Se o algoritmo criptográfico escolhido for fraco para atender a requisitos de processamento dos dispositivos, é importante que a chave seja trocada periodicamente.

A proposta para gerenciamento de chaves de grupo é compatível com as características de redes ad hoc, como a ausência de infra-estrutura fixa e as freqüentes partições da rede. O protocolo proposto não só simplifica o gerenciamento de chaves, mas também evita a necessidade de uma fase de inicialização na qual o administrador prepara alguns nós com

certos segredos que, se expostos, podem levar ao comprometimento de toda a rede. No CHARADAS, todos os nós precisam apenas possuir um par de chaves pública e privada e possuir um certificado, que podem ser obtidos como descrito no Capítulo 3. Assume-se que todos os nós que pertencem à cadeia de confiança podem obter a chave de grupo. Para aplicações nas quais a chave de grupo deve ser passada para um grupo pré-determinado que corresponda a um subconjunto dos nós autorizados a utilizar a rede, deve-se utilizar uma lista de nós autorizados para repassar a chave de grupo.

4.1 Gerenciamento de Chaves de Grupo na Literatura

Muitas propostas foram feitas de gerenciamento de chaves de grupo em redes ad hoc, o que resultou em um número considerável de protocolos baseados em diferentes abordagens. O gerenciamento de chaves centralizado tradicional utiliza um servidor para criar e distribuir as chaves [68] [69]. Essa abordagem, no entanto, não é adequada às redes ad hoc devido à ausência de infra-estrutura e à baixa conectividade da rede.

Protocolos baseados em acordo contributório de chaves [70] [71] [72] também foram propostos, mas estes também sobrecarregam a rede devido ao alto número de mensagens trocadas para formar uma nova chave. As propostas de acordo contributório, em geral, são baseadas no Diffie-Hellman [73]. O Diffie-Hellman é um algoritmo para distribuição de chaves públicas, baseados em dois parâmetros principais: p, que é um número primo, e a, que corresponde à raiz primitiva de p. A raiz primitiva de um número primo p é a potência que gera todos os inteiros de 1 até $p-1$, de forma que os números dados por

$$a \bmod p, a^2 \bmod p, \ \ldots, a^{(p-1)} \bmod p \tag{4.1}$$

são distintos e correspondem aos inteiros de 1 até $1-p$ em alguma permutação. Os dois parâmetros, p e a, são públicos e podem ser utilizados para transportar um segredo. Para tanto, cada nó deve gerar um segredo privado (S_{p_i}), onde $S_{p_i} < p$, e calcular um valor público (V_{p_i}) dado por

$$V_{p_i} = a^{S_{p_i}} \bmod p. \tag{4.2}$$

Em seguida, os nós i e j devem trocar V_{p_i} e V_{p_j} para que a chave secreta (C_s) possa ser

calculada por

$$C_s = (V_{p_j})^{S_{p_i}} \ mod\, p, \tag{4.3}$$

para o usuário i e

$$Cs = (V_{p_i})^{S_{p_j}} \ mod\, p, \tag{4.4}$$

para o usuário j. Assim, ambos os usuários passam a compartilhar o mesmo segredo. Os protocolos baseados em contribuição para gerar uma chave generalizam esse esquema para que um grupo de usuários possa contribuir para a formação da chave final k.

Uma das propostas para obter a chave de forma contributória baseado no Diffie-Hellman é chamada de Burmester-Desmedt (BD) [74]. O BD utiliza uma estrutura em anel na qual após o usuário U_n encontra-se o usuário U_1. Na primeira rodada do BD, cada usuário U_i escolhe um valor privado S_{p_i} e gera o V_{p_i} segundo a Equação 4.2 e o envia para toda a rede. Na segunda rodada, todos os usuários calculam o valor X_i dado por

$$X_i = \left(\frac{Vp_{i+1}}{Vp_{i-1}}\right)^{Sp_i} = a^{Sp_i \cdot Sp_{i+1} - Sp_i \cdot Sp_{i-1}} \ mod\, p \tag{4.5}$$

e o enviam para toda a rede. Por fim, cada usuário computa a chave de grupo (C_g) com a equação

$$\begin{aligned} C_g &= \left(\left(V_{p_{i-1}}^{S_{p_i}}\right)^n \cdot {X_i}^{n-1} \cdot {X_{i+1}}^{n-2} \ \cdots \ X_{i-2}\right) \ mod\, p \\ &= a^{S_{p_1} \cdot S_{p_2} + S_{p_2} \cdot S_{p_3} \ \cdots \ S_{p_n} \cdot S_{p_1}} \ mod\, p. \end{aligned} \tag{4.6}$$

É possível perceber que a chave de grupo é formada pela contribuição de todos os nós, o que dá uma segurança maior contra a escolha e o uso de chaves secretas fracas.

Outra proposta semelhante ao BD para gerar uma chave com base na contribuição de todos os nós é o *Group Diffie-Hellman* (GDH-3) [72]. O GDH-3 foi desenvolvido para atuar em ambientes onde alguns nós possuem pouca capacidade de processamento. Assim, $n-1$ nós realizam poucas exponenciações para obter a chave, enquanto que o $n-ésimo$ nó realiza um total de n exponenciações. O protocolo possui quatro estágios. No primeiro estágio, o usuário U_1 deve enviar o valor $V_1 = a^{S_{p_1}}$ para o usuário U_2. Em seguida, o usuário U_2 deve passar o valor $V_2 = a^{S_{p_1} \cdot S_{p_2}}$ para o usuário U_3. Essa passagem de valores deve seguir de forma que U_i deve repassar para U_{i+1} o valor $V_i = V_{i-1}^{S_{p_i}}$. Esse processo deve seguir até o usuário U_{n-1}, onde n é o número de usuários. Nesse

ponto, inicia-se o segundo estágio, no qual o usuário U_{n-1} deve calcular $V_{n-1} = V_{n-2}^{Sp_{n-1}}$ e repassar esse valor para todos os nós da rede. No terceiro estágio, cada nó i divide o expoente de V_{n-1} pelo seu próprio segredo S_{p_i} e envia o valor obtido para o usuário U_n. O usuário U_n, que deve possuir um poder computacional maior que os outros nós, deve elevar cada um dos valores recebidos a S_{p_n} e repassar esses valores para todos os nós. Com base nos valores recebidos de U_n, é possível que cada nó calcule a chave secreta do grupo.

Um ponto negativo dos mecanismos baseados em acordo contributório é a ausência de soluções para determinar a ordem que os nós devem seguir para a formação da chave, ou qual deve ser o usuário n, no caso do GDH.3. Além disso, tanto o GDH.3 quanto o BD assumem que todos os nós conseguem se escutar, ou, no caso das redes ad hoc, que possuem múltiplos saltos, todos os nós conhecem as rotas necessárias para realizar os *unicasts*. Essa pode ser uma suposição muito forte, em especial quando a chave de grupo está sendo utilizada para tornar o roteamento seguro, como é o caso do *Secure Optimized Link State Routing protocol* (SOLSR). Assim, para esses casos, todas as mensagens de controle de protocolo precisam ser inundadas, aumentando os gastos com a transmissão de mensagens.

A pré-distribuição de chaves, já discutida na Seção 3.1.1 é outro tipo de solução para o problema do gerenciamento de chaves de grupo. Nessa abordagem, uma entidade deve selecionar um conjunto de chaves e, a partir desse conjunto, criar subconjuntos de chaves para distribuir entre os nós. Os nós que possuem chaves em comum podem se comunicar de forma segura, formando um grafo de comunicação que permite que todos os nós se comuniquem com segurança através de múltiplos saltos. Utilizando as chaves em comum que receberam com os subconjuntos, os nós podem trocar dados criptografados. Assim, a chave de grupo pode ser passada pelos canais seguros para todos os nós da rede. O ponto negativo de se utilizar a pré-distribuição de chaves é que o comprometimento de um determinado nó expõe todas as chaves de um subconjunto, permitindo que um nó malicioso se comunique com outros nós se passando por um nó legítimo. Além disso, para realizar a distribuição das chaves, é necessário um administrador que conheça todos os nós da rede antes da rede ser inicializada.

Luo *et al.* propuseram um sistema de distribuição de chave de grupo baseado na pré-distribuição de chaves e no acordo contributório de chaves [75]. Nesse protocolo, os nós precisam manter uma lista de todos os nós excluídos e as chaves que os nós excluídos possuíam devem ser descartadas. Um ponto negativo desse sistema é que a rede pode ter um problema de conectividade no caso de muitos nós excluídos.

Muitos protocolos propõem tornar o gerenciamento de chaves escalável através da clusterização dos nós [74] [76]. O *Distributed, Efficient Clustering Approach protocol* (DECA) é um exemplo de protocolo que forma *clusters* para distribuir chaves em redes ad hoc [77]. A desvantagem desse protocolo é o alto custo energético devido às freqüentes transmissões de mensagens de controle para gerenciar os *clusters*. Outros protocolos para distribuir chaves de grupo são baseados na informação de localização dos nós [78] [79] [80].

O protocolo proposto nesse capítulo lida com desafios como o acesso freqüente de nós não autorizados e partições na rede. O CHARADAS utiliza os certificados obtidos com a autoridade certificadora distribuída proposta no Capítulo 3 para realizar o controle de acesso e utiliza informações de roteamento para melhorar o seu desempenho. As principais vantagens do CHARADAS são o baixo consumo de energia, devido ao pequeno número de mensagens de controle usadas para distribuir a chave de grupo, e a simplicidade para detecção de ações maliciosas. O CHARADAS não precisa conhecer rotas da rede, o que é importante para as aplicações de chave de grupo para proteger o roteamento. O protocolo proposto evita a sobrecarga de mensagens, diferentemente dos protocolos baseados no acordo contributório dos nós e em clusterização. Além disso, o CHARADAS não depende do estabelecimento de segredos antes da rede ser inicializada [75, 81], pois essa premissa poderia restringir os cenários aos quais o protocolo se aplica, além de inserir vulnerabilidades na rede. O CHARADAS necessita apenas de informações públicas para funcionar, que são os certificados e a lista de nós autorizados. Assim, mesmo que algum nó autorizado seja comprometido por um atacante, nenhum segredo da rede é exposto.

4.2 Modelo do Sistema

O CHARADAS funciona sobre a premissa de que os nós se movem livremente e colaboram uns com os outros para permitir a operação da rede. Portanto, partições podem ocorrer em qualquer momento e os membros que compõem o grupo que compartilha uma chave podem mudar com freqüência.

O grupo é definido como o conjunto de nós que pode se comunicar através de rotas de um ou mas saltos. Nós que estejam no mesmo grupo compartilham a mesma chave de grupo para trocar mensagens de controle com assinatura. Foi suposto que o grupo é formado por todos os nós ativos autorizados a utilizar a rede, para analisar o caso de uso de chaves de grupo para tornar seguro o roteamento. Além disso, foi suposto que os nós utilizam o protocolo SOLSR ou algum protocolo similar de roteamento.

O SOLSR é um protocolo de roteamento pró-ativo, de forma que todas as rotas da rede são conhecidas por todos os nós. Além disso, o SOLSR utiliza um mecanismo de controle de inundação chamado *Multipoint Relaying* (MPR), o qual reduz o número de retransmissões em uma inundação. Nesse mecanismo, apenas nós selecionados como MPRs reencaminham mensagens de controle. Os MPRs são selecionados entre os vizinhos por um salto de um nó de forma a atingir todos os vizinhos por dois saltos em uma inundação. Outras características do SOLSR são que os nós podem descobrir o atraso aproximado entre seus relógios e que todas as mensagens de controle são assinadas com a chave de grupo.

4.2.1 Modelo do Adversário

Foi considerado como adversário qualquer nó não-autorizado que tenta acessar a rede. Nós não-autorizados podem agir de forma maliciosa, causando danos à rede através da criação, modificação ou descarte de pacotes, ou apenas funcionar de forma correta, cooperando com todos os nós. O nó não-autorizado que não age de forma maliciosa deve ser excluído da rede por consumir recursos aos quais não tem direito, tais como a banda disponível ou os serviços oferecidos pela rede. Por essa razão, todo tipo de nó não-autorizado

deve ser excluído da rede, o que significa descartar todas as mensagens destinadas ou provenientes do nó não-autorizado, além de impedir que o nó participe do mecanismo de distribuição de chaves de grupo.

Outro tipo de adversário é o nó autorizado que realiza ações maliciosas ou não-cooperativas na rede. Esse tipo de adversário é difícil de detectar e excluir da rede baseado apenas na observação das mensagens de roteamento assinadas com chaves de grupo. Portanto, ao utilizar chaves de grupo, os ataques de roteamento podem ser detectados, mas nenhum nó pode ser acusado, já que a chave de grupo não autentica a identidade do nó, mas apenas indica se o nó pertence ou não ao grupo. Dessa forma, esse tipo de adversário deve ser excluído com base na observação de dados trocados em outras camadas e excluí-lo não está no escopo do CHARADAS. O CHARADAS é capaz de tratar apenas nós autorizados que expõem a chave de grupo, embora não realizem ações maliciosas.

Na análise do CHARADAS não existem premissas sobre o poder de processamento dos adversários. Assim, supõe-se que os adversários são capazes de obter a chave de grupo através de pontos fracos nos algoritmos criptográficos, por força bruta ou invadindo um nó autorizado. Além disso, os adversários são capazes de realizar conluios, embora sejam sempre minoria na rede.

4.3 Protocolo CHARADAS

A principal função do CHARADAS é distribuir a chave de grupo para todos os nós fazendo uso de criptografia assimétrica. O CHARADAS trata das questões como os campos necessários na mensagem para obter um *handshake* de autenticação correto e a construção da árvore de distribuição da chave de grupo.

Para realizar a distribuição da chave de grupo, o CHARADAS utiliza três características do SOLSR: os *multipoint relays* (MPR), o conhecimento do atraso aproximado entre os relógios dos nós da rede e o conhecimento do número de nós na rede. Apesar de o CHARADAS utilizar algumas características do SOLSR, o protocolo proposto não depende de informações de roteamento. O uso de protocolos de roteamento seguro ba-

seado em chaves de grupo, como o SOLSR, implica na necessidade de um sistema de gerenciamento de chaves. Esse sistema de gerenciamento, no entanto, não pode depender de informações como rotas para construir a árvore de distribuição da chave ou para enviar mensagens de controle, pois as mensagens de roteamento só serão trocadas após a distribuição da chave de grupo.

O protocolo proposto é baseado em três mecanismos. O primeiro mecanismo, chamado de distribuição de chave de grupo, é responsável pelo estabelecimento de uma nova chave de grupo no caso de nós serem excluídos e no caso de renovação automática da chave de grupo. Diferentemente de outros sistemas de gerenciamento de chaves de grupo, o roteamento seguro não requer a troca da chave quando um novo nó entra na rede, pois, nesse caso, a chave de grupo não provê confidencialidade, mas identifica quais nós pertencem ao grupo. Além disso, como a chave de grupo só é utilizada para assinar as mensagens de controle, a exposição de chaves de grupo antigas também não influencia no roteamento, pois as novas chaves não possuem correlação com as chaves antigas e o uso de uma chave antiga para assinar uma mensagem provocaria apenas o descarte da mensagem.

O segundo mecanismo do CHARADAS trata como nós que entram na rede podem obter a chave de grupo, como lidar com a união de partições e como realizar a inicialização da rede. O terceiro mecanismo trata a detecção de falha do nó líder e a reposição do líder. No CHARADAS, a distribuição da chave de grupo é inicializada em cada rodada por um líder que deve ser trocado a cada distribuição. Se o líder da rodada falhar e não inicializar a distribuição de chaves, é necessário substituir automaticamente esse nó para continuar a distribuição de chaves.

4.3.1 Distribuição da Chave de Grupo

O mecanismo de distribuição de chaves de grupo substitui a chave de grupo sempre que é chamado. Uma utilização deste mecanismo é para substituir a chave de grupo após a exclusão de um nó, evitando que o nó excluído possa continuar assinando mensagens de controle. Outra possibilidade é a substituição automática da chave, na qual uma distri-

buição periódica é realizada para excluir adversários que possuem a chave de grupo, mas não uma chave privada. Isso pode ocorrer, por exemplo, em redes comunitárias, nas quais um usuário autorizado pode passar para algum amigo não autorizado a chave de grupo da rede para permitir que esse amigo também tenha acesso aos recursos da rede. O mecanismo de distribuição de chaves também pode ser disparado por um sistema de detecção de intrusão (SDI). Quando o SDI envia um alerta, significa que existe um adversário na rede que deve ser excluído.

A Figura 4.1 ilustra o mecanismo de distribuição de chaves. O líder da rodada inicia a distribuição da chave de grupo através do *broadcast* da mensagem Anúncio, a qual indica a existência de uma nova chave de grupo. Quando os vizinhos do líder de rodada escutam a mensagem Anúncio, eles enviam a mensagem Pedido, para requisitar o envio em *unicast* da nova chave de grupo. O líder responde cada um dos Pedidos com a mensagem Resposta, que contém a nova chave de grupo criptografada com a chave pública do vizinho. Em seguida, os vizinhos que são *Multipoint Relays* devem retransmitir o Anúncio e os vizinhos por dois saltos do líder devem escolher um MPR para obter a nova chave de grupo e enviar a mensagem Pedido. Todos os MPRs da rede devem repetir esse mecanismo para garantir que todos os nós receberão a nova chave de grupo.

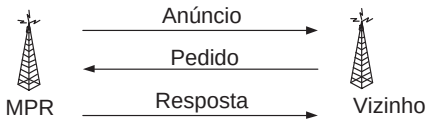
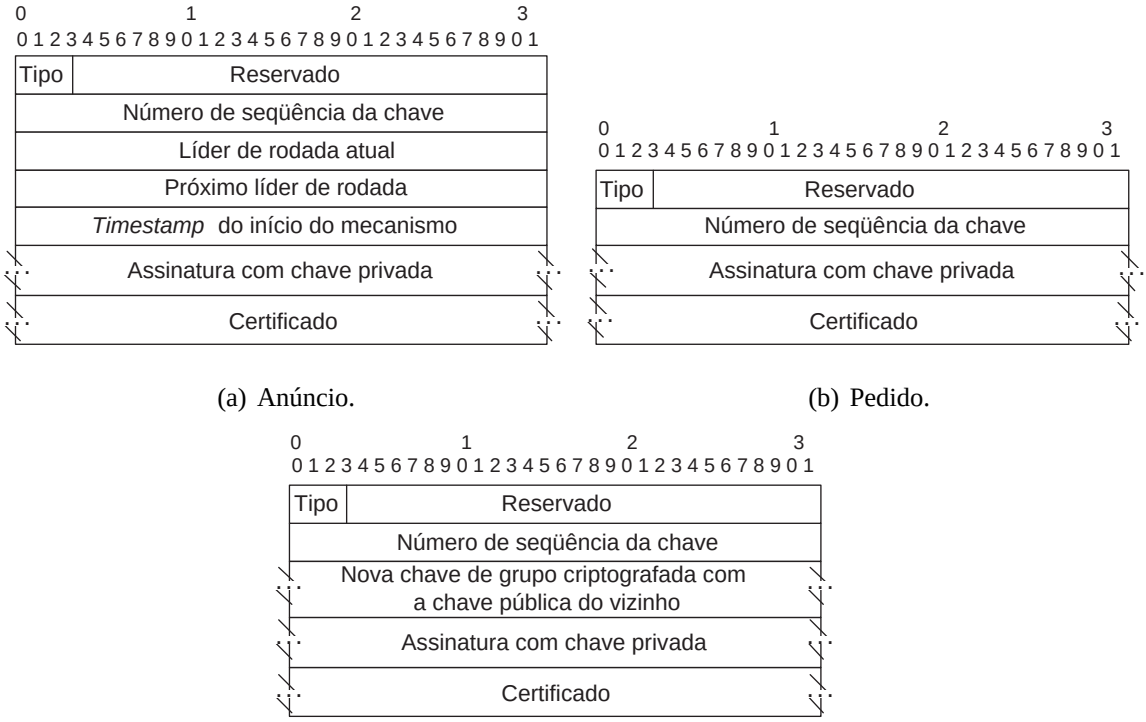


Figura 4.1: Mecanismo de distribuição de chave de grupo.

As mensagens utilizadas no mecanismo de distribuição de chaves de grupo estão na Figura 4.2. Os campos para assinatura, certificado e chave criptografada têm tamanho variável, dependendo da função *hash*, algoritmos criptográficos e tamanhos de chave selecionados. O certificado e a assinatura são importantes para provar a identidade do nó que está enviando a mensagem. Além disso, esses campos garantem a integridade do conteúdo. Sem a assinatura e o certificado nas mensagens de Anúncio e Pedido, a distribuição de chave de grupo não deve ser realizada, pois é necessário provar que ambos os nós estão na lista de nós autorizados e que eles são quem dizem ser.

Os nós devem iniciar o uso da nova chave de grupo aproximadamente ao mesmo



(a) Anúncio.

(b) Pedido.

(c) Resposta.

Figura 4.2: Mensagens do mecanismo de distribuição de chave de grupo.

tempo. Por essa razão, cada nó calcula o tempo esperado para começar a utilizar a nova chave de grupo, T_e, dado por

$$T_e = T_i + T_p \cdot N_{s_{max}}. \tag{4.7}$$

Nessa equação, T_i é o momento aproximado no qual a distribuição de chaves de grupo iniciou, o qual pode ser obtido na mensagem Anúncio, T_p representa a estimativa do tempo máximo que um MPR leva para transmitir a nova chave de grupo para seus vizinhos e $N_{s_{max}}$ representa o número de saltos entre o líder da rodada e o nó mais distante na rede.

Os nós começam a utilizar a nova chave de grupo após T_e, embora classifiquem como legítimas todas as mensagens assinadas com a chave nova e com a chave antiga durante o período dado por $T_e - \alpha$ e $T_e + \alpha$, onde α é a constante que representa a tolerância ao atraso. Depois de $T_e + \alpha$, todas as mensagens que não estejam assinadas com a nova chave de grupo devem ser descartadas. Nós que não receberam a nova chave de grupo antes de $T_e + \alpha$ são tratados como novos nós e podem obter a nova chave de grupo através

do mecanismo de entrada de novos nós, como descrito na Seção 4.3.2. Devido ao α e ao mecanismo de entrada de novos nós, a sincronização exigida pelo CHARADAS é fraca.

4.3.2 Entrada de Novos Nós e União de Partições na Rede

O mecanismo de distribuição de chaves de grupo trata da exclusão de nós, embora não lide com a entrada de nós na rede. Se um nó autorizado entra na rede, ele precisa obter a chave de grupo atual para assinar suas mensagens de controle de roteamento. De forma semelhante, quando duas partições restauram um enlace em comum, elas precisam estabelecer uma chave em comum, de forma que todas as mensagens de roteamento se tornem válidas para os nós em ambas as partições.

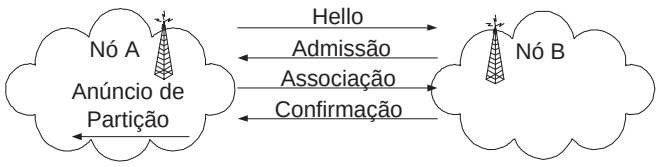
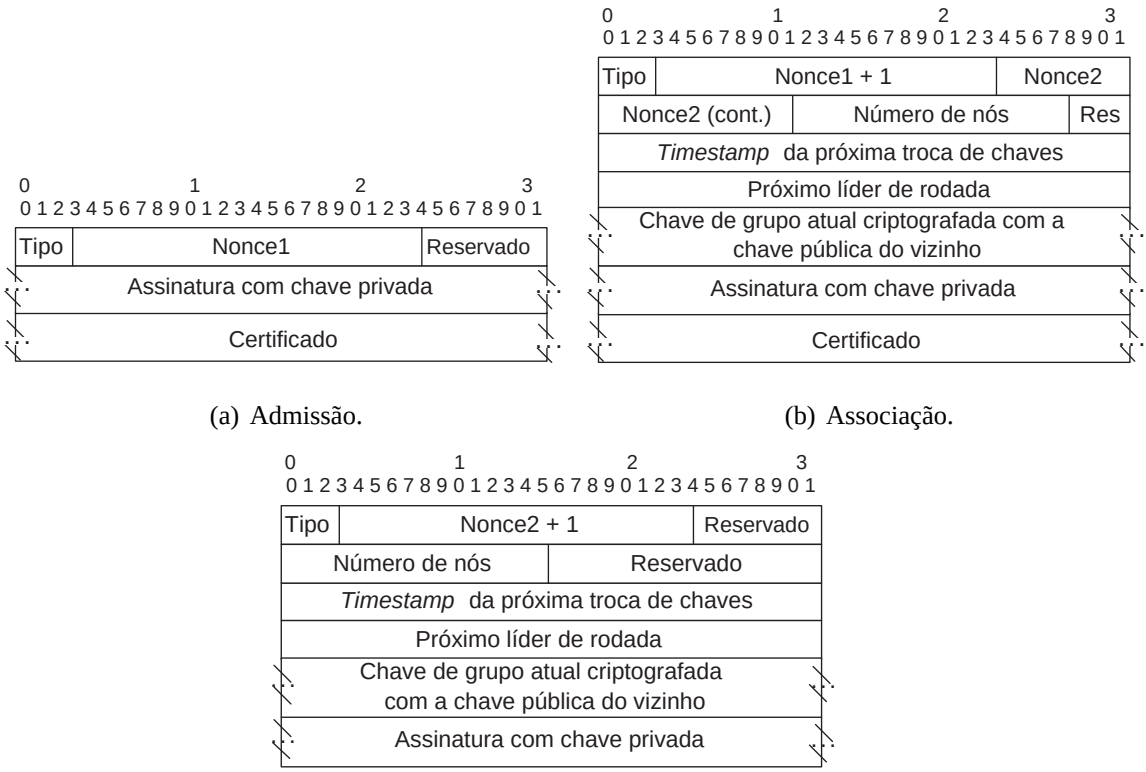


Figura 4.3: Mecanismo de união de partições.

A Figura 4.3 ilustra o mecanismo de entrada após a detecção de uma partição. O mecanismo inicia quando o nó B recebe uma mensagem de um nó autorizado assinada com uma chave de grupo diferente. O nó B, então, envia em *unicast* a mensagem Admissão para sinalizar a detecção de partição. Quando o nó A recebe a mensagem de Admissão, verifica se o nó B está na lista de nós autorizados e se o nó B está ausente da lista de nós ativos. Em seguida, caso ambos os testes tenham sido positivos, o nó A responde com a mensagem Associação, a qual informa para B os parâmetros atuais na partição de A, como a chave de grupo e o número de nós na partição. Após receber a Associação, B envia uma mensagem Confirmação, indicando a recepção da chave e enviando as informações sobre a partição de B. As mensagens Admissão, Associação e Confirmação estão apresentadas na Figura 4.4.

Após a troca de chave entre os nós, as partições devem compartilhar a mesma chave de grupo. Dessa forma, o nó na menor partição se anuncia como líder de rodada imediato e distribui a nova chave de grupo através da mensagem Anúncio de Partição, descrita na



(a) Admissão.

(b) Associação.

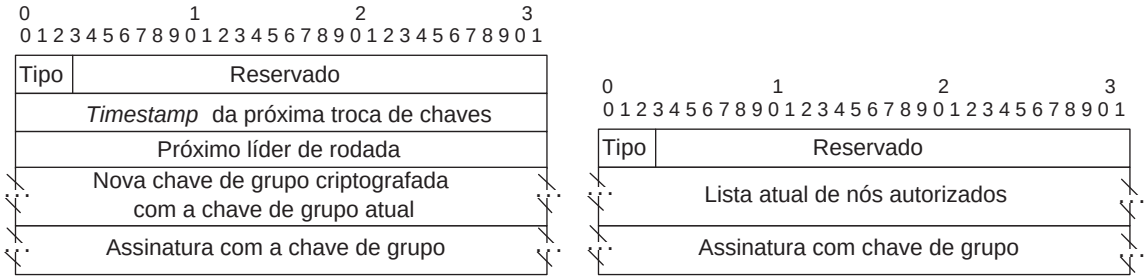
(c) Confirmação.

Figura 4.4: Mensagens para a entrada de novos nós e união de partições.

Figura 4.5(a). Essa mensagem é inundada na menor partição para transmitir a nova chave de grupo para todos os nós. Na união de partições, não é necessário autenticar cada nó como no mecanismo de distribuição de chave de grupo, pois todos os nós com a chave de grupo na união de partições são confiáveis. Portanto, na mensagem Anúncio de Partição, a nova chave de grupo é criptografada com a chave de grupo antiga ao invés das chaves públicas/privadas.

O mecanismo para um nó novo entrar na rede é similar ao mecanismo de união de partição. O novo nó pode obter a chave de grupo com qualquer nó da rede trocando as mensagens Admissão, Associação e Confirmação.

Quando o CHARADAS é utilizado com a autoconfiguração de endereços proposta no Capítulo 2 e com a autoridade certificadora distribuída proposta no Capítulo 3, o processo de obtenção da chave grupo pelo novo nó termina após a recepção da Confirmação. No caso do uso ser feito com outros protocolos, é necessário ainda repassar a lista de nós au-



(a) Anúncio de Partição. (b) Lista.

Figura 4.5: Mensagens para a união de partições.

torizados ativos para o novo nó, através da mensagem Lista, apresentada na Figura 4.5(b).

Inicialização da Rede

Nas redes ad hoc que utilizam as chaves de grupo para prover segurança no roteamento, os nós não podem trocar mensagens de controle até o estabelecimento da chave de grupo. Assim, um dos procedimentos que deve ser realizado durante a inicialização da rede é a distribuição da chave de grupo. O CHARADAS trata a inicialização através do mecanismo de entrada de novos nós e de união de partições. Assim, cada nó forma grupos com seus vizinhos por um salto e, em seguida, os grupos vão se unindo como se cada grupo fosse uma partição. Um exemplo de inicialização da rede em uma topologia genérica está disposto na Figura 4.6.

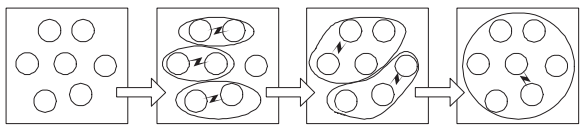


Figura 4.6: Exemplo de distribuição de chave de grupo na inicialização da rede.

O mecanismo para inicialização da rede descrito acima é vulnerável a *loops*, pois vários nós na mesma partição podem iniciar o mecanismo de união de partições com diferentes partições ao mesmo tempo. Para solucionar isso, o CHARADAS utiliza um processo de decisão baseado em duas regras. A primeira regra determina que, se existe mais do que uma união de partição ao mesmo tempo, a mensagem de Anúncio de Partição da menor partição é sempre descartada e a chave de grupo da partição com maior número

de nós é adotada. A segunda regra trata partições com o mesmo tamanho, determinando que, se duas partições possuem o mesmo número de nós, a chave que deverá ser adotada é a da partição cujo líder possuir o maior IP. Com essas duas regras, é possível afirmar que todos os nós iram obter a mesma chave após a fusão de todas as partições da rede.

4.3.3 Detecção de Falha e Substituição de Líder de Rodada

O líder de rodada é responsável por escolher a nova chave e iniciar a distribuição de chave de grupo. Além disso, cada líder de rodada é responsável por escolher o próximo líder, pois é importante evitar que o mesmo nó seja sempre o líder, o que pode implicar em uma sobrecarga para o nó, além de tornar o nó um alvo de ataques. Outro problema que pode ocorrer é que se um nó malicioso fosse escolhido como líder e o líder não fosse trocado, o nó malicioso poderia escolher sempre chaves de grupo fracas, comprometendo a segurança da rede.

Cada líder de rodada seleciona o próximo líder de rodada ordenando os IPs dos nós ativos e selecionando o nó após o seu IP na lista. No caso de se utilizar o protocolo AUFIRA, proposto no Capítulo 2, com Filtros de Bloom, a lista dos nós ativos não fica disponível de forma simples. Nesse caso, o próximo líder deve ser a primeira testemunha do nó.

O líder de rodada, apesar de ser trocado após cada distribuição, ainda representa um ponto de falha, pois é um elemento centralizador. Se o líder de rodada falhar na inicialização da distribuição de chave de grupo, o gerenciamento de chaves ficaria comprometido. Então, o mecanismo de detecção de falha e substituição de líder deve ser utilizado para garantir o funcionamento correto do protocolo CHARADAS.

Os nós detectam a falha de líder de rodada quando uma distribuição de chave de grupo deveria começar, mais após T_c nenhum vizinho enviou a mensagem Anúncio. O momento T_c é calculado por

$$T_c = T_{in} + T_p \cdot N_s + \delta, \tag{4.8}$$

onde N_s representa o número de saltos desde o líder de rodada até o nó e δ representa a tolerância ao atraso. A variável T_p representa uma estimativa do atraso máximo na distribuição da chave de grupo de um MPR para todos os seus vizinhos e T_{i_n} representa o momento esperado para o início da distribuição de chaves. T_{i_n} pode ser calculado pela soma do intervalo de substituição automática de chaves com o momento no qual a última distribuição foi iniciada. O líder de rodada é considerado ausente se após T_c nenhuma chave nova é recebida.

Quando um nó detecta a falha de líder de rodada, ele seleciona o próximo líder colocando em ordem os IPs dos nós ativos e selecionando o próximo nó após o líder. No caso de se utilizar Filtros de Bloom, o próximo líder deve ser a primeira testemunha ativa do líder atual. Todos os nós calculam o novo T_c considerando o atraso até que o novo líder de rodada detecte que o líder atual está ausente. Da mesma forma, o momento para início do uso da chave, T_e (Equação 4.7), é reiniciado para o novo líder de rodada. O mecanismo de substituição de líder de rodada é finalizado para um nó quando ele obtém a nova chave de grupo. Se o nó obtiver chaves de grupo diferentes com atrasos menores que T_e e maiores que T_c, o nó aceita como chave de grupo a chave enviada pelo líder de rodada mais antigo e atualiza o T_p. Nessa situação, as chaves de grupos enviadas pelos líderes mais novos devem ser descartadas.

4.4 Análise do Protocolo

4.4.1 Análise com Redes de Petri

O protocolo proposto foi modelado em redes predicado-ação, como mostrado nas Figuras 4.7, 4.8 e 4.9, cuja legenda está na Tabela 4.1, para analisar o funcionamento do protocolo. Essa rede foi convertida para uma rede de Petri para avaliar se o protocolo atende às três propriedades clássicas: ser limitado, vivo e reiniciável [82] [83]. A ferramenta Analisador de Rede de Petri (ARP) versão 2.3 [84] foi utilizada nessa análise.

O resultado obtido foi que o protocolo atende as três propriedades desejadas. A rede ser limitada indica que o protocolo possui um número finito de estados, o que permite que

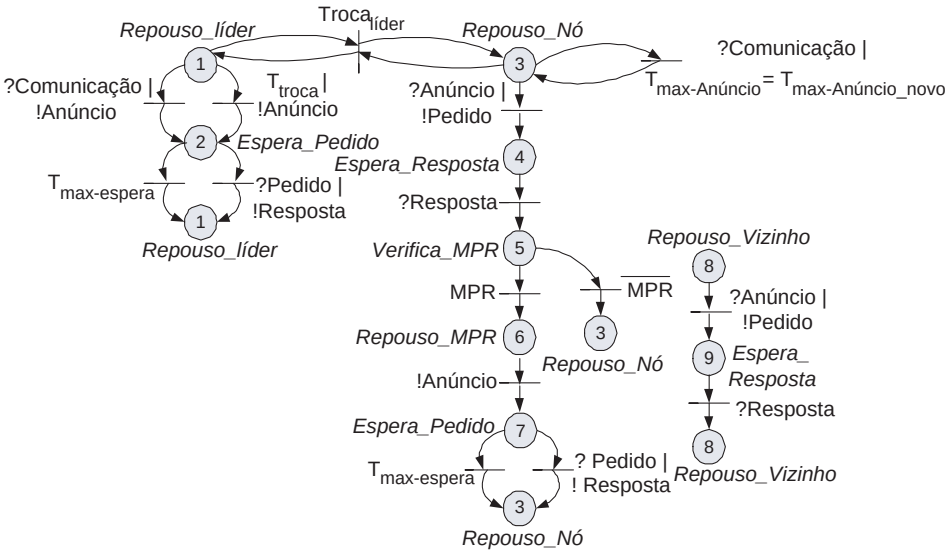


Figura 4.7: Distribuição de chaves de grupo.

ele seja implementado fisicamente. A propriedade de rede viva indica que o protocolo consegue chegar a qualquer um dos estados a partir do estado inicial, o que indica que todas as ações que se deseja realizar com o protocolo são possíveis de se realizar a partir do estado inicial. Por fim, a rede ser reiniciável significa que estando em qualquer estado, é possível voltar ao estado inicial, o que indica que o protocolo não possui *loops* ou pontos de onde não é possível avançar para outro estado.

4.4.2 Análise de Segurança

Nessa seção, são discutidas algumas questões de segurança e como o CHARADAS aliado a um sistema de detecção de intrusão lida com elas. Analisou-se os efeitos da exposição das chaves de grupo e privada devido a invasões ou a usuários autorizados que repassam a chave para usuários não-autorizados.

Exposição da Chave de Grupo

Se um nó não-autorizado obtém a chave de grupo atual CG_n, ele pode assinar mensagens de controle. Suponha que ID_{conj} é o conjunto de identificações dos nós autorizados. Se o nó não-autorizado escolhe aleatoriamente uma identificação ID_k, com

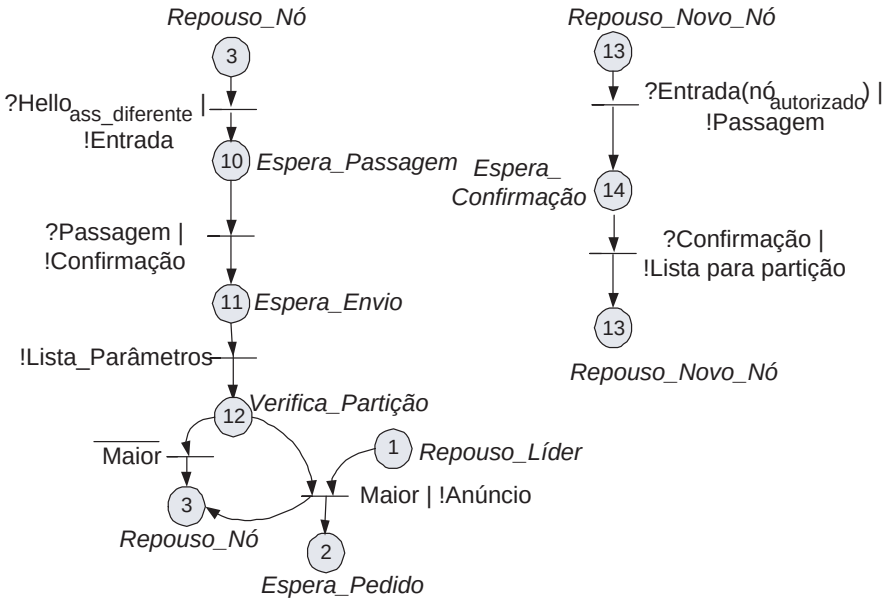


Figura 4.8: Entrada de novos nós e união de partições.

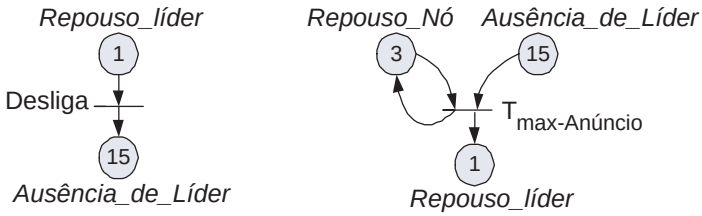


Figura 4.9: Substituição de líder de rodada.

$ID_k \notin ID_{conj}$, esse nó não poderá acessar a rede, pois todos os nós conhecem a lista de nós autorizados, ID_{conj}. Por outro lado, se o nó não-autorizado conhece uma identificação de um nó autorizado que esteja ausente ID_a, $ID_a \in ID_{conj}$, os nós autorizados não podem reconhecer imediatamente a intrusão. No entanto, supondo que f_{rep} é a freqüência da troca automática de chave de grupo, os nós não-autorizados serão excluídos da rede em um período inferior à $1/f_{rep}$, pois o nó não-autorizado não possui a chave privada CP_a e o certificado $Cert_a$ requeridos pela mensagem Ordem na distribuição de chave de grupo. Além disso, se o nó não-autorizado realiza ações maliciosas, o sistema de detecção de intrusão pode enviar um alarme antes da próxima troca automática de chave de grupo. Esse alarme dispara uma distribuição de chave de grupo, excluindo o nó não-autorizado rapidamente.

Caso a autoridade certificadora distribuída descrita no Capítulo 3 esteja sendo usada,

Tabela 4.1: Legenda das Redes Predicado-Ação

Termo	Significado
$Comunicação$	Mensagem notificando detecção de ação maliciosa
T_{troca}	Final do tempo de espera para troca automática de chave de grupo
$T_{max-espera}$	Final do tempo de espera por mensagem Pedido
$Troca_{líder}$	Verificação do campo próximo líder da mensagem Anúncio
MPR	Nó é MPR
$\overline{MPR}$	Nó não é MPR
$T_{max-Anúncio}$	Tempo máximo de espera pela mensagem Anúncio
$Hello_{ass-diferente}$	Hello com assinatura feita com outra chave de grupo
$Maior$	Nó na maior partição
$\overline{Maior}$	Nó na menor partição

outra possibilidade de exclusão do nó é que ele não será capaz de enviar os pacotes de testes assinados com a chave privada para as testemunhas. Assim, a primeira testemunha a detectar a ausência do nó legítimo na rede o retiraria da lista de nós autorizados, impedindo a ação do nó malicioso.

Exposição da Chave Privada

Um caso pior do que a exposição da chave de grupo é a exposição da chave privada de um nó. Neste caso, o nó não-autorizado não só possui a chave de grupo atual, CG_n, mas uma identidade ID_b, $ID_b \in ID_{conj}$, a chave privada CP_b, a chave pública CPu_b e o certificado $Cert_b$, pertencentes a um nó autorizado. Com esse material, o nó não-autorizado pode assinar qualquer mensagem de controle e se autenticar na distribuição da chave de grupo. Além disso, mesmo o disparo de trocas de chaves devido a ações maliciosas não poderia excluir o nó da rede. Por outro lado, se a rede possui um SDI, e o nó autorizado que está sendo forjado na rede estiver presente, o uso inapropriado

do material criptográfico pode ser identificado durante a troca de chave de grupo, pois a mesma identidade irá receber a chave de grupo duas vezes. Essa identidade pode, então, ser bloqueada e um novo processo de distribuição pode ser inicializado.

Ataques Internos

Um ataque interno acontece quando um nó autorizado passa a agir de forma maliciosa. O uso de chaves privadas para assinar todas as mensagens de controle ajuda a evitar esse tipo de ataque. Por outro lado, a criptografia assimétrica possui um custo computacional muito alto e deve ser evitada. Como resultado, atacantes internos devem ser detectados em nível de aplicação, pois a utilização de chaves de grupo no roteamento impede a identificação de atacantes internos. O uso do SDI e do CHARADAS, nesse caso, ajuda apenas na identificação da existência de um atacante interno, pois constantes distribuições de chave de grupo serão disparadas pelo SDI sem conseguir excluir o nó malicioso.

4.4.3 Análise de Desempenho

Na seção anterior, foi analisado o desempenho do CHARADAS com relação à capacidade de exclusão de nós não-autorizados. Nesta seção é analisado o consumo de energia do CHARADAS, que é comparado com outros mecanismos de estabelecimento de chave de grupo. Os resultados mostram que o CHARADAS possui o menor consumo de energia e é eficiente mesmo em cenários desfavoráveis.

Cenário

O CHARADAS foi desenvolvido para funcionar em cenários formados por dispositivos com restrição de energia. Por essa razão, a análise realizada foi uma estimativa do consumo médio de energia com transmissão de mensagens e operações criptográficas, utilizando o CHARADAS com o protocolo de roteamento SOLSR. A ferramenta utilizada na análise foi o Matlab 7.

O cenário utilizado na análise é mais denso que o de uma rede comunitária [11] e é

composto por aproximadamente 1000 nós utilizando a tecnologia IEEE 802.11. A densidade e o número de nós foram escolhidos de forma a representar um cenário adverso com relação ao número de mensagens trocadas. Foi considerado para a análise que o número de vizinhos de cada nó fica aproximadamente constante mesmo considerando-se a mobilidade.

As taxas do SOLSR foram escolhidas segundo a recomendação [64], sendo um HELLO para cada dois segundos e um TC para cada cinco segundos. A menos que se afirme o contrário, os parâmetros do CHARADAS são duas trocas automáticas de chave por dia, dez novos nós entrando na rede por semana e dez nós excluídos por semana. Como esses parâmetros dependem da freqüência das ações maliciosas e de entrada/saída de nós, eles foram variados para analisar o seu impacto sobre o protocolo. Além disso, para representar o pior caso em termos de energia para o CHARADAS na análise realizada, ao ocorrerem uniões de partições, o nó analisado está sempre na menor partição.

Os custos energéticos considerados nessa análise são referentes ao Rockwell Scientific WINS sensor com o microprocessador SA-1110 “StrongARM”. Os custos energéticos considerados estão representados na Tabela 4.2 e 4.3 [74, 85]. Foram escolhidos como algoritmos criptográficos o RSA [86] com chave de 1024 bits, o *Advanced Encryption Standard* (AES) [87] com chave de 128 bits e o *keyed-Hash Message Authentication Code* (HMAC) com chave de 128 bits, pois esses algoritmos são amplamente utilizados e testados.

Tabela 4.2: Custos criptográficos

Algoritmo	Ação	Custo
RSA	Criptografar/Verificar	0.74 mJ/1024-bits de mensagem
RSA	Decriptografar/Assinar	15 mJ/1024-bits de mensagem
AES	Criptografar/Decriptografar	0.00217 mJ/128-bits de mensagem
HMAC	Assinar/Verificar	0.0108 mJ/1024-bits de mensagem

Tabela 4.3: Custos de Transmissão

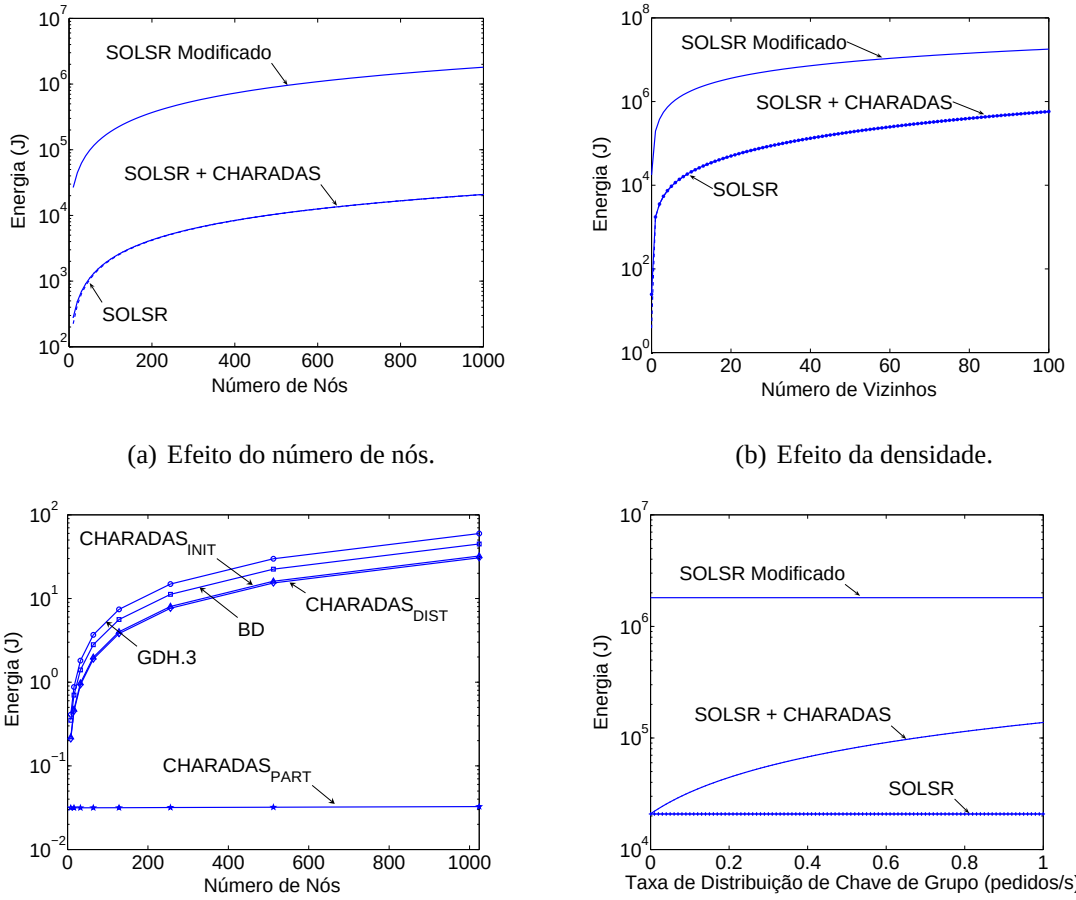
Ação	Custo
Transmissão	$10,8\ \mu$ J/bit
Recepção	$7,51\ \mu$ J/bit

Impacto do CHARADAS

O uso da chave privada para assinar todas as mensagens de controle em protocolos de roteamento seguro simplifica a identificação de nós maliciosos. A criptografia assimétrica, no entanto, consome mais energia do que a criptografia simétrica. Portanto, foi comparado o consumo de energia com operações criptográficas entre uma versão modificada do SOLSR utilizando criptografia assimétrica (SOLSR modificado), o SOLSR tradicional utilizando criptografia simétrica e o CHARADAS.

As Figuras 4.10(a) e 4.10(b) mostram a energia consumida por um nó após uma semana funcionando com diferentes números de nós e diferentes densidades. Tanto o número de nós quanto a densidade influenciam no desempenho do SOLSR e o do CHARADAS, pois esses parâmetros da rede aumentam a quantidade de bits trocados entre os nós. De acordo com a Figura 4.10(a), o SOLSR modificado tem um consumo aproximadamente 86 vezes maior que o SOLSR com o CHARADAS em uma rede com 1000 nós com aproximadamente 10 vizinhos por nó. Quando a densidade é variada para aproximadamente para 100 vizinhos por nó, o SOLSR modificado tem um consumo 31 vezes maior que o SOLSR com o CHARADAS. Dessa forma, o uso freqüente de criptografia assimétrica deve ser evitado devido ao alto consumo de energia. Em uma rede com 1000 nós e aproximadamente 100 vizinhos para cada nó, o custo do CHARADAS representa menos do que $0,07\%$ do consumo de energia do sistema formado pelo SOLSR e o CHARADAS. Portanto, pode-se afirmar que o número de nós na rede e a densidade influenciam o CHARADAS, mas o custo total com operações criptográficas do CHARADAS tem um impacto muito pequeno no sistema.

Na Figura 4.10(c), é mostrada a soma de gastos energéticos de todos os nós com criptografia para distribuir a chave de grupo no mecanismo de distribuição de chave de grupo



(a) Efeito do número de nós.

(b) Efeito da densidade.

(c) Comparação entre EGSR, BD e GDH.

(d) Distribuição de chave de grupo para um nó.

Figura 4.10: Energia consumida com operações criptográficas.

(CHA_{DIST}), no mecanismo de união de partições (CHA_{PART}) no caso de duas partições com o mesmo número de nós e o pior caso da inicialização do CHARADAS (CHA_{INI}). Nessa Figura também é apresentada a soma de todos os gastos de energia com criptografia nos sistemas baseados em acordo contributório para obter a chave de grupo Burmester-Desmedt (BD) [74] e *Generalized Diffie-Hellman* (GDH.3) [72]. Ambos os sistemas são baseados na generalização do Diffie-Hellman para grupos e, assim, o custo com operações de criptografia assimétrica são diferentes do considerado para o CHARADAS, que utiliza o RSA para distribuir a chave de grupo. No BD, todos os nós gastam a mesma quantidade de energia, enquanto que no GDH.3 existe um nó especial responsável por executar mais operações criptográficas, assumindo que pelo menos um nó da rede possui um equipamento com maior processamento e energia. Tanto o BD quanto o GDH.3 assumem que todos os membros do grupo podem escutar qualquer mensagem. Como em nossa análise

as rotas da rede ainda não foram estabelecidas, todas as mensagens desses protocolos são inundadas.

A análise na Figura 4.10(c) descarta a energia consumida com a autenticação no CHARADAS, uma vez que o BD e o GDH.3 tratam apenas das operações criptográficas para se obter uma chave compartilhada, sem discutir como identificar quais nós são realmente membros do grupo. Dessa forma, nesta análise foi avaliada apenas a energia consumida por todos os nós na distribuição/acordo de chave de grupo. Por essa análise é possível concluir que mesmo a fase de inicialização da rede do CHARADAS, que é a fase que consome maior quantidade de energia, é menos custosa que o BD e o GDH.3. O algoritmo do BD consome aproximadamente $1,3$ vezes mais energia do que o CHARADAS na fase de inicialização, $1,4$ vezes mais energia que o mecanismo de distribuição de chave de grupo e 356 vezes mais energia que o mecanismo de união de partição do CHARADAS. Além disso, tanto o BD quanto o GDH.3 devem inundar todas as mensagens, enquanto que o CHARADAS inunda apenas uma mensagem no caso do mecanismo de união de partições. Todas as outras mensagens do CHARADAS são enviadas em *unicast*.

Como o mecanismo de distribuição de chave de grupo consome mais energia do que o mecanismo de união de partições e entrada de novos nós, foi analisado o impacto do aumento da taxa média de chamadas ao mecanismo de distribuição de chave de grupo. Essa taxa média de chamadas representa a troca automática de chaves, a exclusão de nós e a detecção de ações maliciosas, as quais disparam o mecanismo de distribuição de chave de grupo. A Figura 4.10(d) apresenta a energia consumida com operações criptográficas por um nó após uma semana utilizando o CHARADAS com o SOLSR, quando é variada a taxa média de troca de chave de grupo. Pelo gráfico, pode-se afirmar que o CHARADAS não implica em um grande gasto de energia mesmo em cenários com uma troca de chave por segundo, especialmente quando comparado ao uso constante de criptografia assimétrica no SOLSR. O volume de mensagens de controle de roteamento que cada nó recebe é muito superior ao volume de mensagens geradas pelo CHARADAS. Assim, mesmo com uma taxa de uma troca de chave por segundo, o SOLSR com chave de grupo e o CHARADAS possui um consumo de energia 19 vezes menor que o SOLSR com chaves assimétricas (SOLSR modificado).

4.5 Considerações Finais

Nesse capítulo foi apresentado e avaliado o protocolo CHAve de grupo no Roteamento Através de Distribuição Assimétrica Segura (CHARADAS). O uso do protocolo proposto restringe a entrada de usuários não-autorizados em ambientes que utilizam chave de grupo no roteamento, além de automatizar o gerenciamento de chave de grupo de forma distribuída. O uso do CHARADAS com o SOLSR torna o roteamento nas redes ad hoc mais seguro sem causar grande impacto no consumo de energia, mesmo em redes nas quais existe conluio entre nós autorizados e/ou não-autorizados. Além disso, o protocolo sincroniza o uso da nova chave de grupo e é robusto com relação à falha em nós e à formação de partições na rede. O uso de um sistema de detecção de intrusão (SDI) aumenta a segurança provida pelo CHARADAS, pois nós não-autorizados que utilizam a chave privada de um nó autorizado para obter a chave de grupo também são excluídos da rede.

A análise do protocolo indica que ele funciona corretamente e é implementável. O protocolo também é adequado para dispositivos com restrições de bateria e processamento e simplifica a detecção e exclusão de nós não-autorizados em ambientes cujo roteamento é baseado apenas em chaves de grupo. A análise realizada mostra que o CHARADAS consome menos energia que o BD e o GDH.3, que são protocolos conhecidos para a obtenção de chave pelo acordo do grupo. Além disso, o CHARADAS funciona em cenários com mobilidade e conectividade variáveis.

O CHARADAS foi apresentado nesse capítulo para distribuir chave de grupo para o roteamento. No entanto, esse protocolo pode ser generalizado para uso em qualquer aplicação que necessite de chave de grupo. Para tanto, basta que exista uma lista de nós autorizados a participar do grupo, e que as mensagens de Anúncio e Pedido tragam um campo a mais com a estampa de tempo de cada nó. Esse dado permite conhecer a diferença entre os relógios dos nós e permite estimar o momento da última troca de chave. Caso o protocolo de roteamento não seja o SOLSR, a árvore de distribuição pode ser construída como se todos os nós fossem MPRs. Assim, o protocolo torna-se adaptável para qualquer tipo de aplicação e as características de segurança analisadas se mantêm. Ao utilizar o CHARADAS com os protocolos descritos nos Capítulos 2 e 3, o seu uso se torna menos custoso para os dispositivos, pois a lista de nós autorizados não precisa

mais ser guardada e transmitida. Assim, o uso integrado dos três protocolos permite uma solução completa e que não sobrecarrega os nós.

Capítulo 5

Conclusões

AS redes sem fio são parte da Internet do Futuro, pois através delas é possível obter a baixo custo ambientes ubíquos e sistemas de comunicações autoconfiguráveis em cenários inóspitos e hostis. No entanto, as redes sem fio possuem inúmeras vulnerabilidades e garantir a segurança dessas redes é primordial para o sucesso da sua implantação.

As redes ad hoc se apresentam como uma solução de baixo custo ideal para muitas aplicações, em especial, onde a grande quantidade de dispositivos permite uma boa conectividade sem gastos com infra-estrutura. Por outro lado, os desafios para garantir a segurança nesse tipo de rede são maiores do que nas redes cabeadas ou nas redes sem fio infra-estruturadas, pois além de não existir infra-estrutura, não é possível contar com elementos centralizadores na rede e é preciso estimular a cooperação entre os nós para obter uma qualidade de serviço. Outro ponto negativo para as redes sem fio móveis é que os dispositivos contam tipicamente com recursos limitados em termos de energia, operando através de baterias. Essa é mais uma restrição que as redes sem fio devem considerar, utilizando protocolos que poupem energia e processamento.

Muitas soluções apresentadas para redes ad hoc solucionam parcialmente o problema da segurança, tratando a segurança no roteamento, ou a autenticação, ou a monitoração dos nós, fazendo abstrações com relação ao modelo de segurança. De fato, essas abstrações recaem, em sua grande maioria, sobre a dificuldade de controlar o acesso em uma rede não infra-estruturada que não possui elementos centralizadores e na qual até a idéia

de um administrador não é totalmente compatível com as características da rede. Essa tese apresentou uma proposta para o controle de acesso em redes ad hoc, que trata a identificação do nó, o gerenciamento de chaves e a monitoração das ações dos nós.

O primeiro problema de controle de acesso abordado foi a identificação dos nós. Em uma rede funcionando sobre o protocolo TCP/IP, cada nó deve possuir um endereço único para poder estabelecer conexões. A proposta de autoconfiguração de endereços apresentada utiliza filtros para a identificação dos nós, o que significa que, apesar da proposta ser baseada em estados, a capacidade de armazenamento exigida é pequena. Deve-se observar que apesar de existirem propostas de autoconfiguração sem estados, a proposta apresentada neste trabalho é mais robusta que outras propostas da literatura e emite menos mensagens de controle. Além disso, a proposta de autoconfiguração é integrada ao protocolo de gerenciamento de chaves que também se serve da informação que é armazenada por cada nó. Assim, o fato de ser necessário armazenar estados nos nós não representa um acréscimo nos requisitos de memória dos dispositivos. Dessa forma, a proposta de distribuição de endereços, além de ser automática, rápida, distribuída e poupar recursos do nó, é também robusta a perdas de pacote e desconexões na rede. Os resultados da análise do protocolo de autoconfiguração de endereços mostram que a proposta é eficiente e robusta, atendendo a todos os requisitos de uma rede ad hoc. Mesmo quando existem uniões de partições, o protocolo proposto soluciona todas as colisões de endereço enviando poucas mensagens de controle. Quando comparado a uma solução sem estado completa, o protocolo proposto apresenta uma redução de até 22 vezes no número de mensagens de controle trocadas, além de possuir uma resposta rápida às requisições de endereço. Além disso, a proposta elimina todos os casos de colisão de endereços, o que não ocorreu com os demais protocolos comparados.

A autenticação e a monitoração dos nós através do uso de criptografia assimétrica e testemunhas foi outro ponto de controle de acesso para redes ad hoc abordado. Sistemas de autenticação dos nós, em geral, utilizam uma pré-inicialização dos nós ou com a existência de um administrador que controla a entrada e a saída dos nós da rede. Ambas as características não são plenamente compatíveis com os princípios das redes ad hoc. As propostas que mais se adequam aos princípios das redes ad hoc estão relacionadas ao uso de cadeias de confiança e repositórios de certificados. Esse tipo de abordagem, no

entanto, não consegue impedir o retorno à rede de nós maliciosos que tenham sido excluídos anteriormente. Este trabalho propõe uma autoridade certificadora distribuída troca a idéia de um administrador que controla o conjunto de nós que tem acesso à rede para uma entidade administradora formada por todos os nós que controla a entrada dos nós na rede através das relações de confiança entre os usuários. A proposta é interessante não só por inibir a volta de nós maliciosos à rede, uma vez que pais de nós maliciosos também são punidos, mas também por criar um esquema que torna desnecessária a verificação de toda a cadeia para atestar se um elemento é, de fato, parte dessa cadeia. Portanto, a utilização de cadeias de confiança com testemunhas torna a autenticação em redes ad hoc eficiente e compatível com as características da rede.

O sistema de monitoração apresentado também é importante para o controle de acesso, pois uma vez que todos os nós foram identificados e autenticados, é necessário avaliar o seu comportamento e excluir todos aqueles que não agirem de forma a cooperar com o funcionamento da rede. De fato, o sistema de monitoração proposto pode ser utilizado em conjunto com as propostas de detecção de intrusão e confiança já existentes, atuando como uma forma de diminuir a quantidade de dados armazenados por cada nó e dando uma maior robustez na presença de adversários como foi mostrado na análise matemática. Vale ressaltar que para valores adequados dos parâmetros do sistema, ainda que metade dos nós esteja comprometida, a chance de prejudicar algum nó não-malicioso no que diz respeito às atividades da autoridade certificadora distribuída proposta é menor do que 1%.

Por fim, foi também abordado o gerenciamento das chaves de grupo. O protocolo CHAve de grupo no Roteamento Através de Distribuição Assimétrica Segura (CHARADAS) foi proposto e seu desempenho foi avaliado, mostrando uma economia de energia na distribuição de chaves quando comparado a protocolos de distribuição de chaves baseados em acordo contributório. A análise de segurança do CHARADAS mostrou que, com o uso de um sistema de detecção de intrusão, o CHARADAS simplifica a detecção e a exclusão de nós não autorizados, devido à forma como é feito o seu sistema de autenticação. Além disso, a análise formal do protocolo utilizando-se redes de Petri mostrou que ele está livre de *loops* e é implementável. O CHARADAS foi desenvolvido inicialmente para realizar a distribuição da chave de grupo no roteamento utilizando o *Secure Optimized Link State Routing protocol* (SOLSR). No entanto, foi mostrado que o protocolo pode

ser estendido a qualquer tipo de aplicação. De fato, a utilização do CHARADAS com o sistema de identificação de nós propostos é ainda mais fácil devido ao uso dos filtros de endereço, que indicam os nós autorizados que estão presentes na rede. Além disso, o CHARADAS também é simplificado com o uso da autoridade certificadora proposta, pois a questão da determinação do conjunto de nós autorizados que tem direito a obter a chave de grupo é solucionada pela entidade administradora distribuída e os requisitos de memória de cada nó são relaxados, pois os nós não precisam mais guardar a lista de nós autorizados.

As contribuições apresentadas lidam com as características das redes ad hoc, sendo totalmente distribuídas, além de robustas às partições na rede e às freqüentes entradas e saídas de nós. Outro aspecto é que o uso em conjunto das propostas apresentadas garantem ao sistema uma maior robustez como um todo. Dessa forma, problemas e até ataques que poderiam ocorrer ao se utilizar cada uma das propostas em separado são solucionados pelo uso das três em conjunto. Como exemplo, os protocolos para autoconfiguração de endereços nas redes ad hoc, em sua maioria, são vulneráveis aos ataques de negação de serviço, no qual um nó malicioso aloca todos os endereços disponíveis para indisponibilizar o serviço na rede. Ao se utilizar a autoridade certificadora distribuída, esse ataque não pode mais ser realizado, já que apenas os nós que possuírem uma autorização podem acessar a rede. Da mesma forma, a distribuição de chaves simétricas também depende da lista de nós autorizados. Com a utilização da autoridade certificadora distribuída, essa lista não precisa mais ser guardada, evitando sobrecarregar os nós da rede que possuem restrições de armazenamento. Além disso, a utilização de filtros para verificar os endereços utilizados na autoconfiguração de endereços também simplifica a verificação dos nós ativos na distribuição de chaves de grupo e na autoridade certificadora. Dessa forma, é possível afirmar que os elementos dos sistemas propostos estão correlacionados, de forma que certas estruturas devem ser utilizadas em conjunto para dar um maior desempenho aos protocolos sem sobrecarregar os nós.

Além das vantagens da utilização em conjunto, cada uma das propostas apresenta contribuições originais. As propostas têm como foco principal a simplicidade devido às características dos dispositivos sem fio móveis que possuem recursos escassos e, portanto, deve-se evitar ao máximo desperdiçar energia com o envio de mensagens de controle ou

com o excesso de processamento criptográfico. Além disso, o excesso de mensagens de controle representa também um desperdício de banda que também é um recurso escasso.

Com base nas três propostas apresentadas, é possível afirmar que um controle de acesso eficiente para redes ad hoc é obtido, atendendo a todas as características da rede.

Como trabalhos futuros, deve-se desenvolver módulos de simulação para a autenticação e monitoração dos nós, para que seja possível avaliar os valores ideais para os parâmetros dos protocolos. Além disso, testes podem ser realizados verificando o volume de mensagens de controle, o processamento e o armazenamento necessários para cada nó da rede ao se utilizar os três sistemas em conjunto.

Referências Bibliográficas

[1] INTERNATIONAL TELECOMUNICATION UNION. ITU Internet reports 2005: The Internet of Things - executive summary, novembro de 2005.

[2] FERNANDES, N. C., MOREIRA, M. D. D., VELLOSO, P. B., COSTA, L. H. M. K., E DUARTE, O. C. M. B. Ataques e mecanismos de segurança em redes ad hoc. Em *Minicursos do Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg'2006)* (agosto de 2006), pág. 49–102.

[3] RAYA, M., PAPADIMITRATOS, P., AAD, I., JUNGELS, D., E HUBAUX, J.-P. Eviction of misbehaving and faulty nodes in vehicular networks. *IEEE Journal on Selected Areas in Communications (JSAC) 25*, 8 (outubro de 2007), 1557–1568.

[4] BRAUNSTEIN, B., TRIMBLE, T., MISHRA, R., MANOJ, B. S., E RAO, R. On the traffic behavior of distributed wireless mesh networks. Em *2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks (WOWMOM'06)* (2006), IEEE Computer Society, pág. 581–586.

[5] OLIVEIRA, C. T., MOREIRA, M. D. D., RUBINSTEIN, M. G., COSTA, L. H. M. K., E DUARTE, O. C. M. B. Redes tolerantes a atrasos e desconexões. Em *Minicursos do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC'2007)* (2007).

[6] WOOD, A., E STANKOVIC, J. Denial of service in sensor networks. *Computer 35*, 10 (outubro de 2002), 54–62.

[7] KARLOF, C., E WAGNER, D. Secure routing in wireless sensor networks: attacks and countermeasures. *IEEE International Workshop on Sensor Network Protocols and Applications 2003* (maio de 2003), 113–127.

[8] MCGRATH, S. P., ENTIN, E. B., GRAY, R. S., E SHAY, L. The ActComm project: mobile agents and ad hoc routing meeting military requirements for information superiority. Em *Military Communications Conference (MILCOM 2001)* (outubro de 2001), vol. 1, pág. 413–417.

[9] PLESSE, T., LECOMTE, J., ADJIH, C., BADEL, M., JACQUET, P., LAOUITI, A., MINET, P., MUHLETHALER, P., E PLAKOO, A. OLSR performance measurement in a military mobile ad-hoc network. Em *24th International Conference on Distributed Computing Systems Workshops* (2004), pág. 704–709.

[10] CHOUDHARY, M., SHARMA, P., E SANGHI, D. Secure multicast model for ad-hoc military networks. Em *12th IEEE International Conference on Networks (ICON 2004)* (novembro de 2004), vol. 2, pág. 683–688.

[11] CAMPISTA, M. E. M., MORAES, I. M., ESPOSITO, P., AMODEI JR., A., COSTA, L. H. M. K., E DUARTE, O. C. M. B. The ad hoc return channel: a low-cost solution for Brazilian interactive digital TV. *IEEE Communications Magazine 45*, 1 (janeiro de 2007), 136–143.

[12] ANTAS, R. B. Z., FERNANDES, N. C., TAVEIRA, D. M., CAMPISTA, M. E. M., COSTA, L. H. M. K., E DUARTE, O. C. M. B. Análise da qualidade de voz em uma rede ad hoc comunitária. Em *XI Workshop de Gerência e Operação de Redes e Serviços (WGRS'2006)* (maio de 2006).

[13] BEURAN, R., ICHI CHINEN, K., LATT, K. T., MIYACHI, T., NAKATA, J., NGUYEN, L. T., SHINODA, Y., E TAN, Y. Application performance assessment on wireless ad hoc networks. Em *Asian Internet Engineering Conference (AINTEC)* (novembro de 2006), pág. 128–138.

[14] HAFSLUND, A., TØNNESEN, A., ROTVIK, R. B., ANDERSSON, J., E ØIVIND KURE. Secure extension to the OLSR protocol. Em *OLSR Interop and Workshop* (agosto de 2004), pág. 1–4.

[15] ZAPATA, M. G. Secure ad hoc on-demand distance vector (SAODV) routing. *ACM Mobile Computing and Communications Review 6*, 3 (julho de 2002), 106–107.

[16] DROMS, R. *Dynamic host configuration protocol*. RFC 2131, março de 1997.

[17] THOMSON, S., E NARTEN, T. *IPv6 stateless address autoconfiguration*. RFC 2462, dezembro de 1998.

[18] NARTEN, T., E DRAVES, R. *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*. RFC 3041, janeiro de 2001.

[19] PARNO, B., PERRIG, A., E GLIGOR, V. Distributed detection of node replication attacks in sensor networks. Em *IEEE Symposium on Security and Privacy* (maio de 2005), pág. 49–63.

[20] PERKINS, C. E., ROYERS, E. M., E DAS, S. R. *IP address autoconfiguration for ad hoc networks*. Internet Draft, julho de 2000.

[21] FAN, Z., E SUBRAMANI, S. An address autoconfiguration protocol for IPv6 hosts in a mobile ad hoc network. *Computer Communications 28*, 4 (março de 2005), 339–350.

[22] FAZIO, M., VILLARI, M., E PULIAFITO, A. IP address autoconfiguration in ad hoc networks: design, implementation and measurements. *Computer Networks 50*, 7 (2006), 898–920.

[23] KIM, H., KIM, S. C., YU, M., SONG, J. K., E MAH, P. DAP: Dynamic address assignment protocol in mobile ad-hoc networks. Em *IEEE International Symposium on Consumer Electronics (ISCE 2007)* (junho de 2007), IEEE, pág. 1–6.

[24] ZHOU, H., NI, L., E MUTKA, M. Prophet address allocation for large scale MANETs. Em *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM 2003)* (março de 2003), vol. 2, IEEE, pág. 1304–1311.

[25] GIRUKA, V. C., E SINGHAL, M. A localized IP-address auto-configuration protocol for wireless ad-hoc networks. Em *4th international workshop on Wireless*

Mobile Applications and Services on WLAN Hotspots (WMASH'06) (2006), ACM, pág. 101–108.

[26] WENIGER, K. PACMAN: passive autoconfiguration for mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications (JSAC) 23*, 3 (março de 2005), 507–519.

[27] ERIKSSON, J., FALOUTSOS, M., E KRISHNAMURTHY, S. V. DART: dynamic address routing for scalable ad hoc and mesh networks. *IEEE/ACM Transactions on Networking 15*, 1 (2007), 119–132.

[28] FAN, L., CAO, P., ALMEIDA, J., E BRODER, A. Z. Summary cache: A scalable wide-area web cache sharing protocol. *IEEE/ACM Transactions on Networking 8*, 3 (junho de 2000), 281–293.

[29] LAUFER, R. P., VELLOSO, P. B., CUNHA, D. O., MORAES, I. M., BICUDO, M. D. D., MOREIRA, M. D. D., E DUARTE, O. C. M. B. Towards stateless single-packet IP traceback. Em *32nd IEEE Conference on Local Computer Networks (LCN'2007)* (outubro de 2007), pág. 548–555.

[30] CAPKUN, S., HUBAUX, J. P., E BUTTYAN, L. Mobility helps peer-to-peer security. *IEEE Transactions on Mobile Computing 5*, 1 (janeiro de 2006), 43–51.

[31] BUTTYAN, L., E HUBAUX, J.-P. Stimulating cooperation in self-organizing mobile ad hoc networks. *Mobile Networks and Applications 8*, 5 (2003), 579–592.

[32] MERWE, J. V. D., DAWOUD, D., E MCDONALD, S. A survey on peer-to-peer key management for mobile ad hoc networks. *ACM Computing Surveys 39*, 1 (2007).

[33] ZHOU, L., E HAAS, Z. J. Securing ad hoc networks. *IEEE Network 13*, 6 (1999), 24–30.

[34] FRANKEL, Y., E DESMEDT, Y. Parallel reliable threshold multisignature. TR 92-04-02, University of Wisconsin, 1992.

[35] YI, S., E KRAVETS, R. MOCA: mobile certificate authority for wireless ad hoc networks. Em *2nd Annual PKI Research Workshop (PKI 2003)* (abril de 2003).

[36] PEREIRA, F. C., DA SILVA FRAGA, J., NOTOYA, A. E., E CUSTÓDIO, R. F. Autoridade certificadora dinâmica para redes ad hoc móveis. Em *Anais do 25o. Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)* (maio de 2007), pág. 191–204.

[37] KONG, J., ZERFOS, P., LUO, H., LU, S., E ZHANG, L. Providing robust and ubiquitous security support for mobile ad-hoc networks. Em *Ninth International Conference on Network Protocols (ICNP'01)* (novembro de 2001), pág. 251–260.

[38] JOSHI, D., NAMUDURI, K., E PENDSE, R. Secure, redundant, and fully distributed key management scheme for mobile ad hoc networks: an analysis. *EURASIP Journal on Wireless Communications and Networking 5*, 4 (2005), 579–589.

[39] SHAMIR, A. How to share a secret. *Communications of the ACM 22*, 11 (1979), 612–613.

[40] BONEH, D., E FRANKLIN, M. Identity-based encryption from the weil pairing. Em *21st Annual International Cryptology Conference on Advances in Cryptology - CRYPTO '01* (2001), pág. 213–229.

[41] KHALILI, A., KATZ, J., E ARBAUGH, W. A. Toward secure key distribution in truly ad-hoc networks. Em *Applications and the Internet Workshops (SAINT'03 Workshops)* (2003), pág. 342–346.

[42] MIT PRESS. *The Official PGP User's Guide*. Cambrigde, 1995.

[43] CAPKUN, S., BUTTYAN, L., E HUBAUX, J.-P. Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing 2*, 1 (março de 2003), 25–64.

[44] HUBAUX, J.-P., BUTTYÁN, L., E CAPKUN, S. The quest for security in mobile ad hoc networks. Em *2nd ACM international symposium on Mobile ad hoc networking & computing (MobiHoc '01)* (2001), ACM, pág. 146–155.

[45] ESCHENAUER, L., E GLIGOR, V. D. A key management scheme for distributed sensor networks. Em *9th ACM Conference on Computer and Communication Security* (novembro de 2002), pág. 41–47.

[46] CHAN, H., PERRIG, A., E SONG, D. Random key predistribution schemes for sensor networks. Em *IEEE Symposium on Security and Privacy* (maio de 2003), pág. 197–213.

[47] NEWSOME, J., SHI, E., SONG, D., E PERRIG, A. The sybil attack in sensor networks: Analysis & defenses. Em *3rd IEEE/ACM Information Processing in Sensor Networks 2004 - IPSN 04* (abril de 2004), pág. 259–268.

[48] MARTI, S., GIULI, T. J., LAI, K., E BAKER, M. Mitigating routing misbehavior in mobile ad hoc networks. Em *6th annual international conference on Mobile computing and networking (MobiCom'00)* (2000), ACM, pág. 255–265.

[49] FOURATI, A., E AGHA, K. A. On the traffic behavior of distributed wireless mesh networks. Em *Wireless Communications and Networking Conference (WCNC 2007)* (2007), IEEE Computer Society, pág. 2169–2624.

[50] ISLAM, M. M., POSE, R., E KOPP, C. An intrusion detection system for suburban ad-hoc networks. Em *IEEE TENCON 2005* (novembro de 2005), IEEE Computer Society, pág. 1–6.

[51] HE, Q., WU, D., E KHOSLA, P. SORI: A secure and objective reputation-based incentive scheme for ad-hoc networks. Em *Proc. IEEE Wireless Communications and Networking Conference (WCNC 2004)* (março de 2004).

[52] BUCHEGGER, S., E BOUDEC, J.-Y. L. Performance analysis of the CONFIDANT protocol. Em *3rd ACM international symposium on Mobile ad hoc networking & computing (MobiHoc'02)* (2002), ACM, pág. 226–236.

[53] ZHONG, S., CHEN, J., E YANG, Y. R. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. Em *IEEE INFOCOM* (abril de 2003).

[54] BUTTYAN, L., E HUBAUX, J. P. Stimulating cooperation in self-organizing mobile ad hoc networks. *ACM/Kluwer Mobile Networks and Applications (MONET) 8*, 5 (outubro de 2003), 579–592.

[55] LIU, Z., JOY, A. W., E THOMPSON, R. A. A dynamic trust model for mobile ad hoc networks. Em *IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04)* (maio de 2004), pág. 80–85.

[56] PIRZADA, A. A., E MCDONALD, C. Establishing trust in pure ad-hoc networks. Em *27th Australasian Computer Science Conference (ACSC'04)* (outubro de 2004), pág. 47–54.

[57] VELLOSO, P. B., LAUFER, R. P., DUARTE, O. C. M. B., E PUJOLLE, G. HIT: A human-inspired trust model. *8th IFIP IEEE International Conference on Mobile and Wireless Communication Networks (MWCN'2006)* (agosto de 2006), 35–46.

[58] HU, Y.-C., PERRIG, A., E JOHNSON, D. B. Ariadne: A secure on-demand routing protocol for ad hoc networks. *Wireless Networks 11*, 1–2 (janeiro de 2005), 21–38.

[59] SANZGIRI, K., DAHILL, B., LEVINE, B. N., E BELDING-ROYER, E. M. A secure routing protocol for ad hoc networks. Em *10th IEEE International Conference on Network Protocols* (novembro de 2002), pág. 78–87.

[60] HU, Y.-C., JOHNSON, D. B., E PERRIG, A. SEAD: Secure efficient distance vector routing in mobile wireless ad hoc networks. Em *Fourth IEEE Workshop on Mobile Computing Systems and Applications* (WMCSA '02) (junho de 2002), pág. 3–13.

[61] PAPADIMITRATOS, P., E HAAS, Z. Secure link state routing for mobile ad hoc networks. Em *IEEE CS Workshop on Security and Assurance in Ad hoc Networks* (janeiro de 2003), pág. 379–38.

[62] PAPADIMITRATOS, P., E HAAS, Z. Secure data transmission in mobile ad hoc networks. Em *ACM Workshop on Wireless Security (WiSe)* (setembro de 2003), pág. 41–50.

[63] TØNNESEN, A. Implementing and extending the optimized link state routing protocol. Tese de Mestrado, University of Oslo, agosto de 2004.

[64] CLAUSEN, T., E JACQUET, P. *Optimized Link State Routing Protocol (OLSR)*. RFC 3626, outubro de 2003.

[65] FERNANDES, N. C., E DUARTE, O. C. M. B. CHARADAS: Uma proposta para uso de chave de grupo no roteamento através de distribuição assimétrica segura. Em *XXV Simpósio Brasileiro de Telecomunicações (SBrT'07)* (setembro de 2007).

[66] NEUMAN, B. C., E TS'O, T. Kerberos: an authentication service for computer networks. *IEEE Communications Magazine 32*, 9 (setembro de 1994), 33–38.

[67] SLAGELL, A., BONILLA, R., E YURCIK, W. A survey of PKI components and scalability issues. Em *25th IEEE International Performance, Computing, and Communications Conference (IPCCC 2006)* (2006).

[68] GOUDA, M., WONG, C., E LAM, S. Secure group communications using key graphs. Em *ACM Special Interest Group on Data Communications (ACM SIGCOMM'98)* (1998), pág. 68–79.

[69] HARNEY, H., E MUCKENHIRN, C. *Group key management protocol (GKMP) architecture*. RFC 2094, julho de 1997.

[70] JUNG, S.-J., LEE, J.-H., E CHUNG, T.-M. The effective group key agreement protocol for ad-hoc networks for medical emergency environments. Em *SICE-ICASE International Joint Conference 2006* (2006), pág. 1127–1130.

[71] AMIR, Y., KIM, Y., NITA-ROTARU, C., SCHULTZ, J. L., STANTON, J., E TSUDIK, G. Secure group communication using robust contributory key agreement. *IEEE Transactions on Parallel and Distributed Systems 15*, 5 (maio de 2004), 468–480.

[72] STEINER, M., TSUDIK, G., E WAIDNER, M. Key agreement in dynamic peer groups. *IEEE Transactions on Parallel and Distributed Systems 11*, 8 (agosto de 2000), 769–780.

[73] DIFFIE, W., E HELLMAN, M. E. New directions in cryptography. *IEEE Transactions on Information Theory IT-22*, 6 (1976), 644–654.

[74] TEO, J. C. M., E TAN, C. H. Energy-efficient and scalable group key agreement for large ad hoc networks. Em *2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks (PE-WASUN'05)* (2005), pág. 114–121.

[75] LUO, L., SAFAVI-NAINI, R., BAEK, J., E SUSILO, W. Self-organised group key management for ad hoc networks. Em *ACM Symposium on Information, Computer and Communications Security (ASIACCS'06)* (março de 2006), pág. 138–147.

[76] BOUASSIDA, M. S., CHRISMENT, I., E FESTOR, O. Efficient group key management protocol in MANETs using the multipoint relaying technique. Em *Fifth International Conference on Networking and the International Conference on Systems (ICN/ICONS/MCL 2006)* (abril de 2006), vol. 4, pág. 64 – 71.

[77] LI, J. H., LEVY, R., YU, M., E BHATTACHARJEE, B. A scalable key management and clustering scheme for ad hoc networks. Em *First International Conference on Scalable Information Systems (INFOSCALE'06)* (2006).

[78] LIU, J., SACCHETTI, D., SAILHAN, F., E ISSARNY, V. Group management for mobile ad hoc networks: design, implementation and experiment. Em *6th international conference on Mobile data management (MDM'05)* (2005), ACM Press, pág. 192–199.

[79] LI, D., E SAMPALLI, S. An efficient group key establishment in location-aided mobile ad hoc networks. Em *2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks (PE-WASUN'05)* (2005), pág. 57–64.

[80] LAZOS, L., E POOVENDRAN, R. Power proximity based key management for secure multicast in ad hoc networks. *Wireless Network 13*, 1 (janeiro de 2007), 127–148.

[81] LUO, H., KONG, J., ZERFOS, P., LU, S., , E ZHANG, L. URSA: Ubiquitous and robust access control for mobile ad hoc networks. *IEEE/ACM Transactions on Networking 12*, 6 (dezembro de 2004), 1049–1063.

[82] RAMAMOORTHY, C. V., E YAW, Y. A petri net reduction algorithm for protocol analysis. Em *ACM SIGCOMM'86* (1986), pág. 157–166.

[83] LAMCH, D. Verification and analysis of properties of dynamic systems based on petri nets. Em *International Conference on Parallel Computing in Electrical Engineering (PARELEC'02)* (2002), pág. 92–94.

[84] MAZIERO, C. A. *The ARP tool*. http://www.ppgia.pucpr.br/~ maziero/diversos/petri/arp.html, março de 2000.

[85] CARMAN, D. W., KRUUS, P. S., E MATT, B. J. Constraints and approaches for distributed sensor network security (final). Tech Report 00-010, NAI Labs, setembro de 2000.

[86] KALISKI, B., E STADDON, J. *PKCS #1: RSA Cryptography Specifications Version 2.0*. RFC 2437, outubro de 1998.

[87] DAEMEN, J., E RIJMEN, V. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer-Verlag, 2002.