UM MECANISMO DE EXCLUSÃO ACURADO E PRECISO BASEADO EM CONFIANÇA PARA CONTROLE DE ACESSO EM REDES AD HOC

Lyno Henrique Gonçalves Ferraz

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Otto Carlos Muniz Bandeira Duarte

Rio de Janeiro
Dezembro de 2011

UM MECANISMO DE EXCLUSÃO ACURADO E PRECISO BASEADO EM CONFIANÇA PARA CONTROLE DE ACESSO EM REDES AD HOC

Lyno Henrique Gonçalves Ferraz

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

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

Prof. Luís Henrique Maciel Kosmalski Costa, Dr.

Prof. Célio Vinicius Neves de Albuquerque, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
DEZEMBRO DE 2011

Ferraz, Lyno Henrique Gonçalves

Um Mecanismo de Exclusão Acurado e Preciso baseado em Confiança para Controle de Acesso em Redes Ad Hoc/Lyno Henrique Gonçalves Ferraz. – Rio de Janeiro: UFRJ/COPPE, 2011.

XV, 98 p.: il.; 29, 7cm.

Orientador: Otto Carlos Muniz Bandeira Duarte

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2011.

Referências Bibliográficas: p. 92 – 98.

1. Redes Ad Hoc. 2. confiança. 3. Segurança. I. Duarte, Otto Carlos Muniz Bandeira. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

À minha família.

Agradecimentos

Agradeço aos meus pais e toda minha família pelo constante apoio e eterno carinho.

Ao professor Otto, meu orientador, e aos professores Luís Henrique, Miguel, Pedro Velloso e Igor pelos diversos conselhos durante todo o trabalho.

A todos os amigos do GTA, em especial Natalia, Carlo, Diogo, Hugo, Pisa, Pedro C., Rodrigo, pelos conselhos e pela grande ajuda.

A todos os amigos que sempre estiveram do meu lado.

A todos que contribuíram direta ou indiretamente para a minha formação.

À CAPES, CNPq e FINEP pelo financiamento desta pesquisa.

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

UM MECANISMO DE EXCLUSÃO ACURADO E PRECISO BASEADO EM CONFIANÇA PARA CONTROLE DE ACESSO EM REDES AD HOC

Lyno Henrique Gonçalves Ferraz

Dezembro/2011

Orientador: Otto Carlos Muniz Bandeira Duarte

Programa: Engenharia Elétrica

Redes ad hoc móveis são baseadas na cooperação de nós para seu funcionamento. Esse tipo de rede é sujeita a partições de rede frequentes devido ao canal não confiável, nós móveis e entrada e saída de nós na rede. Desse modo, para garantir segurança e justiça, as redes ad hoc dependem de um mecanismo controle de acesso robusto e distribuído. Esta dissertação propõe um mecanismo de controle de acesso baseado em um modelo de confiança para excluir nós não cooperativos da rede. Assim, todos nós participam do controle da rede assumem papéis diferentes, ao monitorar os nós da vizinhança e votar para punir os nós não cooperativos. O mecanismo proposto identifica de maneira acurada os nós não cooperativos em cenários com diferentes densidades e até em cenários sujeitos a partições. A combinação de um modelo de confiança com um mecanismo de votação garante boa classificação mesmo em condições de detecção falha e imprecisão de ações. Resultados de simulações mostram que o mecanismo exclui corretamente um nó malicioso com grande precisão e acurácia.

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

AN ACCURATE AND PRECISE MALICIOUS NODE EXCLUSION MECHANISM FOR AD HOC NETWORKS

Lyno Henrique Gonçalves Ferraz

December/2011

Advisor: Otto Carlos Muniz Bandeira Duarte

Department: Electrical Engineering

Mobile ad hoc networks are based on the cooperation between nodes. These networks are prone to frequent network partitions due to the fading channels, mobile nodes, and frequent membership changes. Hence, to ensure security and fairness, ad hoc networks depend on distributed and robust access control mechanism. In this dissertation, we propose a distributed access control mechanism based on a trust model to exclude non-cooperative nodes. Thereby, all nodes participate in the network control and play multiple roles, monitoring the nodes, voting to punish non-cooperative nodes. Our mechanism accurately identifies non-cooperative nodes in different density scenarios and even in partition likely scenarios. The combination of voting and trust mechanisms guarantees good classification even though in scenarios of low and imprecise neighbor action detection. Simulation results show that mechanism excludes nodes with high accuracy and precision.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

$J_{\mathbf{d}}$	Júri ou conjunto de jurados de um réu **d**., p. 53
$Q_{\mathbf{w}}(\mathbf{d})$	Avaliação do comportamento de um réu **d** realizada pela testemunha **w**., p. 53
$R_{\mathbf{w}}(\mathbf{d})$	Recomendações que uma testemunha **w** possui sobre um réu **d**, dos seus vizinhos em comum com **w**., p. 53
R_{max}	Valor máximo de reputação., p. 59
T_E	Intervalo de tempo no qual o jurado aceita a mensagem de evidência de uma testemunha., p. 59
T_R	Intervalo de tempo sem receber mensagens de evidências que o jurado deve esperar para aumentar a reputação., p. 59
$T_{\mathbf{w}}(\mathbf{d})$	A confiança de uma testemunha **w** em um nó réu **d**., p. 53
$W_{\mathbf{d}}$	Conjunto de testemunhas de um réu **d**., p. 53
$\mathbf{d}$	Nó réu., p. 53
$\mathbf{j}$	Nó jurado., p. 53
$\mathbf{w}$	Nó testemunha., p. 53
u	Unidade de decremento/incremento da reputação., p. 59

Lista de Abreviaturas

AES	Advanced Encryption Standard, p. 6
AODV	Ad hoc On-Demand Distance Vector routing protocol, p. 10
ARAN	Authenticated Routing for Ad Hoc Networks, p. 10
ARP	Address Resolution Protocol, p. 58
BBDS	Bad Behavior Detection System, p. 25
CASTOR	Continuously Adapting Secure Topology-Oblivious Routing, p. 44
CA	Certificate Authority, p. 7
CBE	Certificate-Based Encryption, p. 24
CBTRP	Cluster Based Trust-aware Routing Protocol, p. 34
CCS	Credit Clearance Service, p. 39
CL-PKC	Certificateless Public Key Cryptography, p. 24
COFFEE	COntext FrEE, p. 43
DES	Data Encryption Standard, p. 6
DHCP	Dynamic Host Configuration Protocol, p. 58
DSR	Dynamic Source Routing, p. 31
DoD	Denial of Decryption, p. 24
DoS	Denial of Service, p. 5
ECC	Elliptic Curve Cryptography, p. 7, 24
FAP	Filter-based Addressing Protocol, p. 64
HIP	Host Identity Protocol, p. 9, 64

ID-PKC	Identity-based Public Key Cryptography, p. 24
IP	Internet Protocol, p. 9
IPsec	IP Security, p. 9
KGC	Key Generator Centre, p. 24
LISP	Locator Id Separation Protocol, p. 64
MANETs	Mobile Ad hoc NETworks, p. 1
MARCH	Money And Reputation sCHemes, p. 42
MiM	Man in the Middle, p. 7
OLSR	Optimized Link State Routing protocol, p. 10
P2P	Peer to Peer, p. 42
PKG	Private Key Generator, p. 24
PKI	Public Key Infrastructure, p. 7
REP	Recommendation Exchange Protocol, p. 36
RREP	Route Reply, p. 10
RREQ	Route Request, p. 10
RSA	Rivest-Shamir-Adleman Cryptography, p. 64
SAODV	Secure Ad hoc On-Demand Distance Vector, p. 10
SEAD	Secure Efficient Ad hoc Distance vector, p. 10
SEAR	Secure Efficient Ad hoc Routing protocol, p. 10
SGC-PKC	Self-Generated-Certificate Public Key Cryptography, p. 24
SOLSR	Secure Optimized Link State Routing protocol, p. 10
SRP	Secure Routing Protocol, p. 10
Sprite	Simple cheat-pRoof credIT-based SystEm, p. 39
TTL	Time to Live, p. 10

Capítulo 1

Introdução

As redes de computadores se desenvolveram durante o século XX, e com o advento da Internet, se tornaram um grande sucesso que é atualmente. Com a evolução das redes de computadores, apareceram diversos requisitos antes inexistentes, tais como segurança, mobilidade, qualidade de serviço [1]. Nesse sentido surgiram as redes ad hoc sem-fio, que são redes que utilizam o ar como canal de comunicação. As redes ad hoc sem-fio são descentralizadas e não possuem infraestruturas de redes convencionais, como roteadores e pontos de acesso. Uma variante desse tipo de rede são as redes ad hoc móveis (*Mobile Ad hoc NETworks* - MANETs), que além de todas as características das redes ad hoc sem-fio, os nós se movimentam e assim causam mudanças na topologia da rede [2]. Nessa rede, os nós ao alcance do rádio uns dos outros se comunicam diretamente, e quando um nó está fora do alcance do outro, eles se comunicam com a ajuda de nós intermediários para encaminhar os pacotes. As MANETs são atrativas por justamente não necessitar de infraestrutura, assim possuem baixo custo de instalação, funcionam em ambientes diversos e mesmo enquanto os elementos da rede se movem. Entretanto, a ausência de infraestrutura, a natureza do canal sem-fio e a mobilidade trazem diversos desafios relacionados à segurança, confiabilidade e disponibilidade.

A falta da infraestrutura específica faz com que os próprios nós assumam os papéis de roteador, servidor e também cliente. Assim, quanto mais nós participarem da rede, maior o número de rotas possíveis entre os nós e a banda disponível, e menor a possibilidade de ocorrer partições na rede. Desta forma, para se conseguir esses atributos, os nós devem cooperar em prol da rede. Contudo, os nós podem falhar em cooperar por estarem sobrecarregados, com defeito, ou podem comportar-se mal e assumir um comportamento egoísta para economizar os próprios recursos, ou malicioso para atrapalhar a rede. Portanto, um sistema seguro que garanta a cooperação entre nós é indispensável para a operação correta da rede.

A segurança das redes ad hoc é normalmente realizada através do uso de um sistema de controle de acesso com autenticação. Assim, somente os nós autorizados

podem participar e usufruir da rede. Entretanto, a utilização de um sistema de controle de acesso não garante que os nós na rede cooperem sempre. Mesmo se somente um grupo seleto de nós for autorizado a participar da rede, isso não os impede de depois mudarem seus comportamentos e atrapalhar a rede seja intencionalmente ou devido limitações de recursos. Assim, o sistema de controle de acesso deve possuir um mecanismo de que seja capaz de identificar os nós não cooperativos e proibir o acesso aos recursos da rede.

Dessa maneira, o sistema de controle de acesso inclui um mecanismo de exclusão de nós que tem um objetivo importante, ele deve ser capaz de analisar e julgar a melhor atitude em prol da rede, pois o mecanismo de exclusão de nós deve proibir o acesso de nós que prejudiquem a rede e ao mesmo tempo deve garantir a máxima participação dos nós na rede. Então, o mecanismo de exclusão deve ser altamente preciso e acurado em identificar os comportamentos dos nós para que possa fazer o julgamento.

Neste sentido, esse trabalho propõe a construção de um mecanismo de exclusão de nós não cooperativos. O mecanismo realiza o controle de acesso em dois níveis, um nível local que faz a análise de comportamentos da vizinhança do nó, e um nível global distribuído na rede toma as decisões. O nível parte local do mecanismo tem como objetivo realizar uma análise acurada e precisa dos comportamentos dos nós com pouco impacto na rede. Quando o nível local detectar um comportamento inadequado, ele contata o nível global para que tome uma atitude em relação ao nó mal comportado. O nível global é importante para que as decisões sejam tomadas de maneira imparcial. Além disso, grupo que compõe o nível global é específico para um nó e é escolhido de maneira aleatória, de forma que dificulta a formação de conluios.

Assim, o mecanismo proposto pode ser caracterizado por um tribunal de júri, no qual cada nó assume múltiplos papéis: tanto de testemunha e jurado, quanto de réu. O papel de testemunha é realizado no nível local, quando um nó monitora seus vizinhos e avalia o nível de confiança neles. As testemunhas devem reunir as informações sobre o comportamento de seus vizinhos para enviar ao nível global como evidências de comportamento. Outro papel que o nó exerce é o papel de jurado. Ele exerce esse papel ao ser escolhido para julgar se certo nó deve ou não ser punido. Em caso positivo, o jurado vota pela exclusão do nó da rede, e a decisão de exclusão é tomada com base no voto da maioria dos nós do júri. Finalmente, cada nó assume papel de réu, assim ele é sujeito ao tribunal do júri.

Para analisar a proposta foram realizadas simulações que comparam a proposta com trabalhos relacionados. As simulações avaliam a eficácia do mecanismo em excluir nós mal comportados. Além disso, analisou-se a eficiência do mecanismo quando o monitoramento do ambiente não é ideal, de maneira que não é possível

fazer o monitoramento de todas as ações dos vizinhos e quando o mecanismo de monitoramento confunde-se e erra na classificação de comportamento.

O restante da dissertação está organizado como se segue. A Seção 2 descreve os principais trabalhos relacionados segurançaca e cooperação em redes ad hoc móveis. No Capítulo 3 apresenta-se a descrição da arquitetura do mecanismo de exclusão. O Capítulo 4 apresenta uma análise do modelo de exclusão. No Capítulo 5 as simulações realizadas e resultados obtidos são descritos, e finalmente o Capítulo 6 conclui a dissertação e apresenta direções futuras.

Capítulo 2

Segurança e Cooperação em Redes Ad Hoc Móveis

As redes ad hoc móveis (*Mobile Ad Hoc Networks* - MANETs) são naturalmente descentralizadas e não necessitam qualquer infraestrutura específica. Dessa maneira, deve-se ter a máxima colaboração de nós da rede nas tarefas de transferência de pacotes para ter a maior disponibilidade de caminhos e de banda agregada. Então, essas redes contam com a máxima participação e cooperação de nós para as tarefas de encaminhamento e roteamento de pacotes. Assim, a situação ideal é que todos os nós da rede estejam dispostos a cooperar no encaminhamento dos pacotes, uma vez que a não cooperação nas tarefas de encaminhamento coloca em risco a alcançabilidade da rede. Entretanto, um nó pode não participar do encaminhamento, pois não tem disponível poder de processamento, memória, banda, ou até por estar com defeito no *software* ou *hardware*. O nó pode também assumir um comportamento egoísta e não participar do encaminhamento e roteamento de pacotes por esperar que outros nós o façam, ou para economizar seus recursos como bateria, poder de processamento, memória. Além disso, um nó malicioso pode realizar diversos ataques á rede para espioná-la ou prejudicá-la [3].

Outra característica importante das MANETs é a utilização do canal de comunicação sem-fio. Uma das vantagens da comunicação sem-fio é a universalização do acesso, no qual os nós para se comunicarem precisam basicamente de uma antena e nenhuma instalação de infraestrutura como cabos e outros aparelhos. Entretanto, por causa do acesso ao canal ser fácil, a comunicação sem-fio possui problemas de segurança. Isso ocorre, pois qualquer dispositivo que esteja ao alcance do sinal de rádio, pode realizar diversos tipos de ataques. Esses ataques são desde espionagem passiva a interferência ativa, como geração, modificação e destruição de mensagens. No cenário de comunicação sem-fio não existe uma barreira de defesa clara, então todos os nós devem estar preparados para se defender de possíveis ataques. Além disso, pelo fato da rede ser descentralizada, não existe elementos com funções es-

pecíficas de controle e segurança da rede. Essas tarefas devem ser realizadas pelos próprios nós de maneira colaborativa. Outra importante vantagem das redes ad hoc é a possibilidade dos nós se moverem, saírem da rede e se conectarem. Essas características fazem com que a topologia da rede seja dinâmica, com mudanças frequentes de vizinhos que pode causar partições na rede. Deste modo, os mecanismos de controle e segurança da rede devem ser projetados funcionar nessas condições [4].

2.1 Segurança em Redes

O tópico de segurança em redes, não só para MANETs, mas para redes de computadores em geral, é dividido em diversos atributos que devem ser atendidos: disponibilidade, integridade, autenticação, confiabilidade, não-repúdio e autorização [5].

A disponibilidade se refere aos serviços da rede que devem permanecer disponíveis apesar de nós comportarem-se de maneira inadequada. Exemplos disso são quando nós da rede assumem comportamentos egoístas ou quando realizam ataques de negação de serviço (*Denial of Service* - DoS). A disponibilidade é um atributo que deve ser protegido em diversas camadas de protocolos diferentes, pois uma negação de serviço pode vir tanto da camada física através de interferências no sinal de rádio, quanto da camada de rede ao atrapalhar o protocolo de roteamento, ou ainda da camada de aplicação com a abertura de diversas sessões em um servidor *web*.

A integridade garante que as mensagens trocadas entre os nós na rede permanecerão inalteradas. As alterações podem ocorrer por causa de falhas na comunicação sem-fio ou por ataques de nós maliciosos. Além disso, a integridade pode ser vista no contexto de uma conexão, de forma a garantir que as mensagens não são removidas, reordenadas ou repetidas.

O atributo autenticação é a capacidade de se garantir a identidade do nó par com quem se comunica. Sem a autenticação um nó não autorizado pode fingir ser outro para ganhar acesso a recursos e informações ou até interferir com a operação de outros nós.

A confidencialidade garante que a informação permaneça vedada a entidades não autorizadas. Confidencialidade é importante para a segurança de informações sensíveis como informações militares estratégicas ou táticas.

Não repúdio garante que o nó que enviou a mensagem não possa negar esse fato. O não repúdio é importante para detectar e isolar nós comprometidos.

Finalmente, a autorização estabelece regras para o acesso dos usuários, quais os recursos e informações que cada um pode ter acesso. Esse atributo é importante quando existem diferentes grupos com permissões distintas.

Para se conseguir oferecer segurança às redes com esses atributos, normalmente utiliza-se a criptografia. Em geral, a criptografia é uma ferramenta importante para

a provisão de segurança em redes, e em redes ad hoc isso não é exceção. Uma das ferramentas da criptografia são as funções *hash* criptográficas, que são importantes para a garantia de integridade dos dados. Basicamente, são funções que mapeiam quaisquer dados binários em uma sequência de *bits* com tamanho fixo. O cálculo direto da função *hash* deve ser simples de ser realizado, mas o cálculo inverso deve ser impossível computacionalmente. Ou seja, a partir da um valor de saída da função *hash* deve ser impossível obter os dados binários que deram origem àquela saída. Além disso, as funções *hash* criptográficas devem evitar colisões de valores de saída. Os algoritmos MD5 [6] e SHA-1 [7], uns dos mais utilizados, possuem essas propriedades. As funções *hash* também são conhecidas por outras propriedades como a uniformidade de valores saída no conjunto imagem, independência dos valores de saída em relação à entrada, de maneira que pequenas modificações nos valores da entrada causem grande mudança nos valores de saída.

Com a criptografia é possível codificar a mensagem de forma que dificulta quem não possua uma chave criptográfica a ler a mensagem. A criptografia computacional é dividida em dois ramos, a criptografia simétrica e a criptografia assimétrica. Na criptografia simétrica, as partes que desejam comunicar-se de forma secreta compartilham uma mesma chave secreta. Essa chave é utilizada tanto na codificação quanto da decodificação da mensagem. Os algoritmos de criptografia simétrica normalmente utilizam operações de ou-exclusivo, a troca de colunas, a troca de linhas, a permutação, a rotação e a expansão. Essas operações possuem baixo custo computacional, mas combinadas são capazes de dificultar bastante a decodificação da mensagem para quem não tem acesso à chave secreta. Os algoritmos de criptografia simétrica mais utilizados são o DES (*Data Encryption Standard*) [8] e AES(*Advanced Encryption Standard*) [9]. Um dos maiores problemas desse tipo de criptografia é a distribuição das chaves, que deve se realizar antes da comunicação segura.

No outro ramo da criptografia computacional, a criptografia assimétrica, cada usuário possui um par de chaves criptográficas: a chave privada e a chave pública. Uma mensagem codificada pela chave privada só pode ser decodificada pela chave pública correspondente e vice-versa. Dessa maneira, um usuário mantém sua chave privada em segredo, e pode distribuir sua chave pública para outros usuários. Assim, os outros usuários podem decodificar mensagens codificadas pela chave privada, ou podem enviar mensagens codificadas com a chave pública para que só quem possui a chave privada correspondente conseguir decodificá-las. Com a criptografia assimétrica é possível se fazer a distribuição segura de chaves, que resolve um dos problemas da criptografia simétrica, e também a assinatura digital com o uso conjunto com funções *hash*, que garante a autenticação e não repúdio. Um de seus problemas é que seus algoritmos utilizam operações de exponenciação, portanto, requerem maior processamento computacional se comparados com os algoritmos de

criptografia simétrica. Algoritmos de criptografia assimétrica mais conhecidos e utilizados são RSA [10] (*Rivest Shamir Adleman*) e Diffie-Hellman [11], e mais recente ECC (*Elliptic Curve Cryptography*) [12].

As soluções para segurança em redes normalmente utilizam a criptografia híbrida, que utiliza as características tanto da criptografia simétrica quanto da assimétrica. Basicamente, as partes informam suas chaves públicas uns para os outros, e as utilizam para criar um segredo compartilhado entre as partes. Posteriormente, esse segredo será utilizado para codificar as mensagens com menor custo computacional com a criptografia simétrica. Entretanto, essa abordagem está sujeita a ataques do homem do meio (*Man in the Middle* - MiM), no qual um nó intermedeia a comunicação entre as partes e finge ser a outra parte para ambas. Ataques MiM podem ser evitados com o uso de infraestruturas de chaves públicas (*Public Key Infrastructure* - PKI) [13] e com esquemas de assinaturas digitais e certificados. No PKI, existe uma entidade terceira confiável, que conhece as partes e comprova suas identidades. Essa entidade, chamada de autoridade certificadora (*Certificate Authority* - CA) gerencia certificados, que são assinaturas das próprias chaves públicas das partes codificadas pela chave privada da CA. Assim, quando uma parte anuncia sua chave pública, a outra parte pode verificar se ela é válida com a comparação com o certificado.

Entretanto, a criptografia não soluciona todos os atributos de segurança em redes. O atributo disponibilidade não é tratado diretamente com a criptografia. Normalmente, utilizam-se outros mecanismos para se garantir a disponibilidade, como a redundância e distribuição dos serviços. Ainda assim, mecanismos de autenticação podem limitar o acesso de atacantes aos serviços da rede, e, consequentemente, diminuir a possibilidade de ataques de negação de serviço.

2.2 Segurança em Redes Ad Hoc Móveis

As redes ad hoc móveis (MANET) estão sujeitas diversos ataques, que se classificam em dois tipos. O primeiro tipo é o ataque passivo, nos quais os atacantes não interferem com a rede, simplesmente espionam a rede sem alterar seu funcionamento. O segundo tipo é ataque ativo, no qual os nós interferem ativamente na rede, possivelmente com a alteração, criação e descarte de dados em trânsito. Os ataques ativos podem normalmente exploram vulnerabilidades específicas de alguma camada específica da pilha de protocolos. A camada de rede é a principal camada alvo dos ataques às MANETs, devido às vulnerabilidades dos protocolos de roteamento que consideram unicamente cenários não hostis, e também devido ao fato do alcance dos ataques serem amplos, podem influenciar a rede toda.

Um importante ataque à camada de rede é o túnel de minhoca (*wormhole*), no qual dois nós atacantes criam um túnel de comunicação em um enlace de baixa

latência para difundir para a rede um caminho atrativo de encaminhamento de mensagens. Os nós maliciosos ficam com uma posição privilegiada e atraem para si o papel de encaminhador de pacotes de muitas rotas de comunicação entre nós. Assim, os atacantes podem causar danos à rede quando quiserem ao deixar de encaminhar pacotes. Outro ataque comum é a falsificação de identidades, no qual o atacante gera ou rouba identidades para explorar a redundância das redes ad hoc. Esse ataque não é exclusividade da camada de rede, ele pode afetar diversas camadas e impactar diversos protocolos. Nesse ataque, o nó malicioso assume outras identidades, existentes na rede ou novas. Dessa maneira, o atacante pode usar as identidades para diversos fins. Um exemplo é no caso de um sistema cuja segurança dos dados é obtida com a distribuição de fragmentos do dado a identidades distintas. Assim, um atacante que possua diversas identidades pode ser escolhido para guardar diversos fragmentos de um dado e conseguir decifrar o dado. Outra possibilidade com esse tipo de ataque baseia-se no fato de que alguns protocolos de roteamento utilizam múltiplos percursos para evitar rotas que possuam atacantes. Entretanto, com a falsificação de identidades, as identidades de um nó malicioso poderiam ser escolhidas justamente para fazer parte dos percursos escolhidos pelo protocolo de roteamento. Ainda, quando se utiliza mecanismos de reputação e confiança, o atacante com várias identidades pode distribuir a culpa de suas más ações ou pode utilizar outras identidades para zerar seu histórico. Além disso, múltiplas identidades possibilitam o atacante alterar quaisquer mecanismos de votações na rede ou distribuição de recursos em seu favor, realizar ataques em conluio contra os mecanismos e protocolos da rede.

2.2.1 Protocolos de Roteamento Ad Hoc Seguros

Os protocolos de roteamento convencionais para redes ad hoc móveis assumem que todos os nós são cooperativos, então não estão preparados para combater situações hostis. Com o intuito de reduzir as vulnerabilidades da camada de rede, desenvolveram-se diversos protocolos de roteamento seguros.

Uma das formas de se proteger a integridade e autenticidade dos dados trafegados é a com a utilização de protocolos que funcionam na comunicação fim a fim como o IPsec [14] e HIP [15]. Entretanto, mesmo com a utilização desses protocolos, o roteamento permanece vulnerável. Para se ter o roteamento seguro, deve-se proteger as mensagens de controle dos protocolos de roteamento. Para tal, diversos protocolos de roteamento que protegem as mensagens de controle e que garantem sua autenticidade e integridade foram desenvolvidos para assegurar o roteamento seguro.

Um protocolo de roteamento seguro é o SAODV (*Secure Ad hoc On-Demand*

Distance Vector) [16] que estende o protocolo AODV (*Ad hoc On-Demand Distance Vector*) [17] para assegurar o processo de descoberta de rotas. SAODV utiliza mensagens de assinatura chamadas de "Extensão de Assinatura" (*Signature Extension*) para garantir os atributos integridade, autenticação e não-repúdio para o protocolo de roteamento. O SAODV utiliza dois mecanismos para proteger as mensagens de roteamento: a assinatura digital e a cadeia de *hash*.

A assinatura digital garante a autenticação e não repúdio dos pacotes. Dessa maneira, o nó de origem assina a mensagem e atesta seus dados. Durante o procedimento de descoberta de rota, alguns valores de campos das mensagens do protocolo de roteamento são modificados tais como o campo Número de Saltos das mensagens de requisição e resposta de rota (*Route Request* - RREQ e *Route Reply* - RREP respectivamente). Portanto, a assinatura da mensagem que é feita pela origem das mensagens não pode ser realizada sobre esses campos que se alteram.

Os campos que têm seus valores alterados durante o processo de descoberta de rota também devem ser protegidos, pois garantem a consistência da rota, e se forjados, comprometem o funcionamento do roteamento. Por exemplo, um nó malicioso responder um pedido de rota com número de saltos menor que o real é uma forma de obrigar as mensagens a passar por este nó. Desse modo, esses campos são protegidos por meio de uma cadeia de *hash*. Para tal, são utilizados três campos: Número Máximo de Saltos, *Hash* e *Top Hash*. O nó de origem cria a mensagem define um valor máximo de saltos e coloca esse valor no campo Número Máximo de Saltos. Em seguida, o nó de origem gera uma semente aleatória e coloca no campo *Hash*, e coloca no campo *Top Hash* o resultado da aplicação recursiva da função *hash* sobre a semente aleatória um número de vezes igual ao máximo de saltos definido. Os nós intermediários que encaminharem as mensagens da descoberta de rota alteram o campo *Hash* da mensagem para o resultado da aplicação da função *hash* sobre esse campo. Assim, qualquer nó pode verificar o campo Número de Saltos ao se comparar o valor do campo *Top Hash* com o resultado da aplicação da função *hash* ao campo *Hash* um número de vezes igual à diferença entre Número Máximo de Saltos e Número de Saltos. Se os valores forem iguais, garante-se que o campo número de saltos não foi alterado.

Já o protocolo SOLSR (*Secure Optimized Link State Routing protocol*) [18, 19] é uma extensão do protocolo OLSR (*Optimized Link State Routing protocol*) [20]. O SOLSR adiciona um campo de Assinatura que garante a autenticidade das mensagens de controle, no qual se coloca assinaturas digitais baseadas em chaves simétricas. As mensagens são assinadas salto a salto, inclusos os campos que alteram a cada salto como o Número de Saltos e TTL (*Time to Live*). Somente uma assinatura é necessária por salto, pois as mensagens podem ser agrupadas em um único pacote OLSR. A assinatura é feita com um *hash* da chave secreta, assim, os nós

que não tem acesso à essa chave secreta não podem produzir a assinatura correta. Como a assinatura é realizada salto a salto, o protocolo não faz a autenticação fim a fim diretamente, mas forma uma cadeia de autenticação salto a salto. O SOLSR utiliza *timestamps* para evitar ataque de replicação, além de um mecanismo de troca de *timestamps* para a conexão inicial entre os nós. Tanto as assinaturas quanto os *timestamps* são posicionados no corpo das mensagens OLSR para manter a compatibilidade com o protocolo sem segurança.

Além do protocolo de roteamento SAODV e do protocolo SOLSR existem diversos outros que provêem segurança para o roteamento como ARAN (*Autenticathed Routing for Ad Hoc Networks*) [21], Ariadne [22], SRP (*Secure Routing Protocol*) [23, 24], SEAD (*Secure Efficient Ad hoc Distance vector*) [25], SLSP (*Secure Link State Protocol*) [26] e SEAR (*Secure Efficient Ad hoc Routing protocol*) [27]. Todos esses protocolos se baseiam em assinaturas digitais, criptografia assimétrica ou compartilhamento de chaves secretas. Entretanto, nenhum desses protocolos de roteamento seguros possui um mecanismo para a gerência e distribuição das chaves e certificação das identidades, então esses protocolos dependem de sistemas de gerenciamento que realizam essas funções. Além disso, esses protocolos não possuem nenhum sistema de controle de acesso para permitir e proibir o acesso, e não exige atitudes altruístas e cooperativas dos nós.

2.2.2 Controle de Acesso e Autenticação em Redes Ad Hoc

A maioria dos serviços seguros da rede como protocolos seguros de roteamento em MANETs e comunicação segura fim a fim, se baseiam em mecanismos criptográficos como criptografia assimétrica e compartilhamento de chaves. Então, para uma rede segura, é necessário um sistema de gerenciamento de chaves.

Entretanto, a gerência de chaves não garante a segurança por completo, pois além de se distribuir e validar chaves, deve-se validar a associação das chaves com identidades reais. Além disso, as redes possuem propósitos usuários-alvo diferentes, então somente quem for permitido deve utilizar os serviços da rede e os outros devem ser privados do acesso. Dessa maneira, mecanismos de controle de acesso e autenticação são de extrema importância para o funcionamento correto da rede, e devem integrar ou acompanhar os mecanismos de geração e validação de chaves criptográficas.

Tradicionalmente, as autoridades certificadoras (*Certification Authorities* - CAs) são utilizadas para garantir a validade das identidade como em PKI e Kerberos [28]. Elas certificam as chaves públicas dos nós, então são responsáveis tanto pelo controle de acesso ao gerar certificados somente a quem for permitido, como também pela autenticação das identidades aos gerar os certificado em si. Entretanto, o uso de CAs

em redes ad hoc sem-fio não é recomendado, pois requer infraestrutura especializada e uma entidade de controle central e confiável. A utilização de um ponto único de distribuição de um serviço nas redes ad hoc móveis é desaconselhada, pois o serviço pode não ficar sempre disponível. A rede está sujeita à ocorrência de partições devido à mobilidade, perda de comunicação, congestionamento de enlaces. Além disso, o serviço da entidade central poder ficar indisponível devido a ataques de adversários, que podem impedir o encaminhamento de pacotes à entidade central ou podem fazer ataques de negação de serviço para extinguir recursos.

Uma solução para o problema de se ter uma entidade central é a sua abolição. Assim, todos os nós da rede ao iniciar já deveriam possuir todos os certificados dos outros nós. Essa abordagem, portanto, limita muito o escopo da rede, que deve ser fechada, de curta duração e com poucos nós. Além disso, essa rede teria certificados permanentes uma vez que não poderia compreender nenhum mecanismo de atualização de certificados, pois este tipo de mecanismo requer uma autoridade certificadora.

Alternativamente, uma abordagem para prover as funcionalidades de uma autoridade certificadora (CA) de forma distribuída em redes ad hoc é fazer com que os próprios nós da rede certifiquem as chaves públicas dos outro nós. Um esquema utiliza a criptografia de limiar [29, 30]. Com a criptografia de limiar, um grupo de n nós são aptos a gerar um certificado parcial, e com pelo menos k certificados parciais é possível gerar um certificado completo. Dessa maneira um nó que queira um certificado difunde sua solicitação e se pelo menos k nós responderem essa solicitação o nó pode obter o certificado completo.

Distribuição da Autoridade Certificadora em Servidores Especializados

Um esquema distribuído no qual os próprios nós realizam a funcionalidade da autoridade certificadora de certificação de identidades foi proposto por Zhou e Haas [31]. Esse esquema adapta as funcionalidades da CA para o ambiente de MANETs, assim, aumenta a disponibilidade e tolerância a falhas dos serviços ao distribuir a CA em um grupo de nós servidores. Na proposta, a CA distribuída é equipada com um par de chaves pública/privada, cuja chave pública é conhecida por todos os nós da rede e a chave privada é dividida entre servidores responsáveis pelo gerenciamento de certificados. A distribuição da CA não é, portanto, uma simples réplica, cada um dos servidores possui seu próprio certificado e par de chaves e eles colaboram entre si para a emissão de novos certificados. Nesse esquema, os servidores compartilham a tarefa de emissão de certificados, para isso assinam coletivamente os certificados com o uso da criptografia de limiar. A utilização da criptografia de limiar com um esquema de (k, m) (com $k < m$) permite que pelo menos k de m servidores possam fazer operações criptográficas colaborativamente.

Para isso, a chave privada é dividida em m partes distribuídas para os m servidores. Quando uma assinatura precisa ser gerada, cada servidor utiliza sua parte da chave e realiza uma assinatura parcial. As assinaturas parciais podem ser combinadas por um servidor especial para a geração da assinatura do certificado quando pelo menos k de m servidores realizaram a assinatura parcial. Assim, no esquema proposto por Zhou e Haas é possível que o certificado seja emitido mesmo que até $m - k$ servidores estejam inacessíveis ou comprometidos. Se adversários controlarem menos de k servidores o serviço não é afetado, e os outros servidores podem detectar o comportamento duvidoso dos servidores comprometidos. O número de servidores pode ser calculado para satisfazer as condições de disponibilidade da rede em relação ao tamanho da rede e frequência de mudanças de topologia. Além disso, é possível garantir que atacantes não atuem por longos períodos de tempo com uma a utilização de uma técnica de atualização periódica das partes da chave (*share refreshing*) que calcula novas partes de chave independentes a partir das antigas, sem que a chave privada seja revelada. Assim, os nós teriam se comportar corretamente para que consigam obter novos certificados válidos para continuar na rede. O tempo de atualização pode ser pequeno para diminuir as vulnerabilidades do sistema, mas não deve ser muito pequeno para evitar sobrecarga excessiva.

Geração Local de Certificados

Luo *et al.* propuseram um sistema de autenticação distribuído para redes, no qual cada nó da rede executa funcionalidades de segurança localmente e, assim, todos os nós colaborativamente realizam a segurança geral da rede [32, 33]. O sistema baseia-se no fato que para a provisão de atributos de segurança como confidencialidade, autorização, integridade, não-repúdio e também disponibilidade, é necessário um sistema eficiente para a provisão de autenticação. Após a criação de um canal de comunicação autenticado, os outros atributos de segurança podem ser obtidos com mecanismos de troca de chaves. Então, a construção de um sistema de autenticação é imprescindível para uma rede segura. Esse sistema realiza a autenticação dos nós com um mecanismo de emissão local de certificados. A emissão de certificados é realizada com o uso da criptografia de limiar, cujas partes das chaves são divididas entre nós em um escopo local. O sistema proposto utiliza um modelo de confiança com escopo local que caracteriza as questões de segurança em redes ad hoc. Além disso, o sistema possui uma arquitetura de autenticação local e distribuída, para garantir disponibilidade ubíqua para nós móveis mesmo com a ocorrência de ataques de negação de serviço e intrusões. Todos os nós estão equipados com mecanismos para a detecção local de mau comportamento de seus vizinhos, e também com ferramentas de verificação de certificados.

A noção de entidade confiabilidade é utilizada com diversos mecanismos de au-

tenticação, como no caso de uma autoridade certificadora (CA) que valida a propriedade de chaves. Então, a autenticação deve basear-se em entidades confiáveis. No modelo de confiança utilizado por Luo *et al.*, um nó é considerado confiável em certo período, se os seus vizinhos considerados confiáveis disserem que o nó é confiável naquele período. E assim, o nó é considerado confiável globalmente.

No sistema de Luo *et al.*, existe um par de chaves assimétricas de uma entidade considerada "autoridade confiável", cuja chave pública é conhecida por todos. A chave privada da "autoridade confiável" é divida entre todos os nós segundo um esquema polinomial de ordem $k-1$ de compartilhamento de chave [34], de maneira que pelo menos k nós possam coletivamente reaver a chave privada e assinar certificados que serão aceitos em toda rede. Nesse sistema, todos os nós da rede possuem um certificado assinado pela chave privada da "autoridade confiável", e aqueles que não possuem não podem participar da rede. Todas as vezes que um nó precisa renovar seu certificado ele faz uma requisição em *broadcast* para seus vizinhos. Cada nó que recebe a requisição e considera o nó pedinte confiável faz uma assinatura parcial com a sua chave privada parcial. Assim, o nó pode gerar um certificado completo com k partes recebidas, que inclusive podem ser acumuladas conforme o nó se move.

Quando um nó recebe um pedido de emissão de certificado ele deve decidir se assina ou não o certificado com sua chave privada parcial. Entretanto, o nó pode não ter informações necessárias para realizar essa decisão, pois a relação entre os nós pode ainda ser recente. Em uma abordagem, o nó pode descartar o pedido de emissão de certificado até que o nó pedinte demonstre bom comportamento, mas isso faria com que nós legítimos não consigam certificados prontamente e esperarem em um local até conseguirem o certificado antes se moverem. Outra abordagem, a qual é utilizada, é aceitar os pedidos de certificados e assiná-los, sob risco de ser um nó malicioso que se moveu para obter um certificado novo. Nesse caso, os nós maliciosos devem ser identificados e excluídos da rede.

A eliminação dos nós maliciosos é realizada com mecanismos distribuídos de revogação de certificados. Para tal, um nó condena outro se o monitoramento direto e a avaliação própria considerá-lo malicioso. Quando um nó condena outro, ele não assina mais os certificados do nó malicioso e difunde uma acusação para seus vizinhos. A acusação tem alcance de propagação controlado para evitar sobrecarga de mensagens na rede, de maneira que o nó malicioso não consiga sair da área de propagação antes de seu certificado expirar. Se um nó receber k ou mais acusações acerca de um nó, ele também condena o nó, não mais assina seus certificados e difunde uma acusação.

Na inicialização da rede, os nós precisam obter suas chaves privadas parciais para assinar os certificados parciais. Inicialmente, um nó cria as chaves parciais e as distribui para os primeiros k nós. Posteriormente, os nós utilizam as chaves privadas

parciais e criam outras chaves parciais para os nós novos que entram na rede. Além disso, os nós atualizam periodicamente as suas chaves privadas parciais para evitar que adversários consigam ter acesso a k chaves parciais.

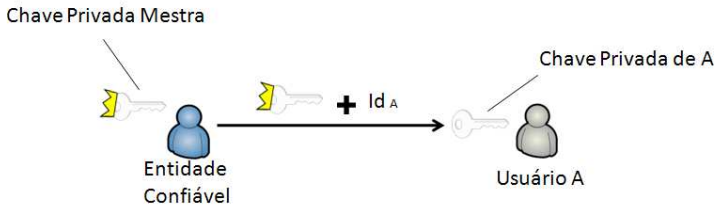
O sistema de Luo *et al.* não sobrecarrega a rede inteira com mensagens de controle, então escala com o tamanho da rede. Além disso, a decisão e assinatura de certificados são feitas localmente, e assim provê acesso ubíquo à rede.

Chaves Públicas baseadas em Identidades

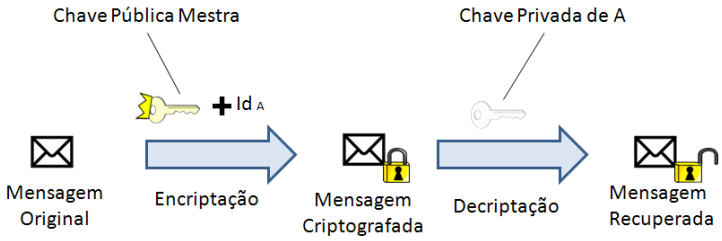
As identidades de usuários e suas chaves criptográficas são normalmente validadas por meio da utilização de certificados. Nesses sistemas, uma entidade confiável terceira emite os certificados que validam a relação de uma identidade de um usuário com suas chaves criptográficas, como por exemplo, a chave pública. Alternativamente, poderia ser utilizado um esquema criptográfico no qual a própria chave pública certifica implicitamente a identidade do dono da chave pública.

Uma proposta que faz a própria chave pública certificar implicitamente a identidade do dono da chave pública, é a utilização do próprio identificador dos usuários como chave pública. Assim, as próprias chaves públicas (que são os identificadores) certificam as identidades. Esse tipo de criptografia, a criptografia de chaves públicas baseadas em identidades (*Identity-based Public Key Cryptography* - ID-PKC), foi iniciada por Shamir em 1984 [35]. O objetivo desse esquema é disponibilizar uma infraestrutura de chaves públicas (PKI) sem a utilização de certificados. Nesse esquema, existe uma entidade confiável terceira, a geradora de chaves privadas (*Private Key Generator* - PKG). Ela possui duas chaves, a chave pública mestra que deve ser conhecida por todos da rede para encriptar as mensagens, e a chave privada mestra que é utilizada para gerar as chaves privadas para cada usuário como mostrado na Figura 2.1(a). Assim, as mensagens são criptografadas com a utilização de somente de um identificador arbitrário do destinatário e a chave pública mestra, e a decriptação da mensagem é realizada com uma chave privada pessoal obtida com a PKG. Dessa forma, um usuário contata a PKG, se autentica e requisita sua chave privada, para assim poder decriptar as mensagens. Esse procedimento é mostrado na Figura 2.1(b).

Com o esquema de criptografia baseada em identidades, a expiração de chaves públicas pode ser realizada diretamente. Desse modo, diferentemente de certificados que possuem data de expiração, a própria chave pública utilizada para criptografar o dado pode estar com a validade embutida (ex. a chave pública poderia ser "ID || validade"). Assim, o emissor da mensagem não precisa obter um novo certificado válido do receptor todas as vezes que o receptor atualiza sua chave pública, como acontece no caso de certificados. A revogação de chaves públicas é realizada quando a PKG não envia a um nó uma nova chave privada de uma chave pública com



(a) A entidade confiável usa a chave privada mestra e a identidade do usuário para gerar a chave privada do usuário.



(b) A encriptação é realizada através da chave pública mestra e do identificador do usuário (que funciona como chave pública do usuário). Para decriptar a mensagem criptografada, o usuário utiliza sua chave privada recebida da entidade confiável.

Figura 2.1: Criptografia baseada em identidades.

nova validade. Nesse caso, a PKG pode ser instruída para parar de enviar a chave privada de um nó quando necessário. De maneira semelhante, a chave pública pode além incluir a validade da chave, pode incluir credenciais para garantir que somente usuários com acesso ao serviço (e as credenciais) podem decodificar a mensagem. Além disso, o esquema criptográfico baseado em identidades é eficiente, pois pode ser feito através de criptografia de curvas elípticas (*Eliptic Curve Cryptography* - ECC) [36], que gera mensagens cifradas e assinaturas pequenas, além de ser leve computacionalmente [37].

O esquema criptográfico baseado em identidades (ID-PKC) evita a emissão de certificados e os problemas associados, como a verificação e transferência de certificados, e assim economiza recursos de armazenamento, rede e processamento. Entretanto, esse esquema não apresenta uma solução para renovação e revogação direta de chaves.

Gerenciamento de Chaves Públicas baseadas em Criptografia de Identidades com Criptografia de Limiar

Para aumentar a disponibilidade, flexibilidade e eficiência da distribuição de chaves públicas em ambientes ad hoc Khalili *et al.* [38] propuseram um protocolo

para gerenciamento e autenticação para ad hoc com ID-PKC e criptografia de limiar. Nessa abordagem, não existe a priori chaves e material criptográfico, nem relações de confiança e segurança entre os nós. Essas relações e materiais criptográficos são estabelecidos quando se cria a rede. No momento de criação da rede, os nós participantes criam colaborativamente de maneira distribuída uma chave pública para o sistema de ID-PKC que deve ser conhecida por todos os integrantes da rede. Também no momento de criação da rede cria-se a chave privada, a qual é dividida através da criptografia de limiar (k, m) entre os m participantes iniciais da rede, e é recuperada ao se juntar ao menos k partes de chave privada. Assim, um nó entra na rede e após um processo de autenticação, ele é identificado e recebe do grupo de nós iniciais sua chave privada para decriptar as mensagens destinadas a ele.

Assim, nesse esquema, a distribuição da PKG previne ponto único de falha e adiciona resistência de até k nós comprometidos. Além disso, se menos de k nós iniciais estiverem acessíveis, o nó poderia acumular as partes da sua chave privada e se aproveitar da mobilidade para encontrar outros nós iniciais e obter outras partes chave privada. Entretanto, a utilização de criptografia de limiar adiciona complexidade ao sistema e introduz sobrecarga de troca de mensagens ao sistema.

Apesar de esquemas criptográficos baseados em identificadores (ID-PKC) serem interessantes por sua propriedade de auto-autenticação das chaves públicas, a falsificação de identidades é um problema desse esquema. Nesse sentido, o procedimento de emissão de chaves poderia requisitar uma prova não falsificável de sua identidade, como informações definidas direto no *hardware*. Em outra abordagem para prevenir a falsificação de identidades, assume-se que as identidades são aleatórias e impossíveis de serem previstas, e permitir a emissão das chaves somente uma vez para cada identidade. Dessa maneira é impossível de prever uma identidade e obter a chave pessoal de um nó se ele já a possui [38].

Revogação de Chaves Públicas baseadas em Criptografia de Identidades

Hoeper e Gong apresentaram um esquema auto-organizado [39] para realizar a renovação e revogação de chaves com o esquema criptográfico baseado em identidades (ID-PKC) para a geração das chaves privadas em redes ad hoc móveis (MANETs). Primeiramente, os autores definem um padrão para a chave pública dos nós, que inclui a identificação do nó, a data de expiração definida em intervalos regulares e assim previsível para os outros nós, e a versão da chave dentro do intervalo de validade. A inclusão da versão é necessária para a renovação da chave dentro de um período de expiração. Com esse padrão de chave pública, todos os nós conhecem a priori as chaves públicas dos nós, com a necessidade de redistribuição de chaves públicas somente quando uma chave pública foi comprometida e outra versão de chave é redistribuída dentro de um período de expiração.

Para prover as funcionalidades de revogação de chaves, Hoeper e Gong apresentam três algoritmos: vigília da vizinhança, *harakiri* e esquema de acusação. No primeiro algoritmo, a vigília da vizinhança, cada nó está no modo promíscuo e monitora comportamentos suspeitos de seus vizinhos de um salto. Tais comportamentos podem ser frequentes descartes de pacotes, ou um número grande de mensagens enviadas. Assim, cada nó mantém um registro de seus vizinhos que indica se eles tiveram comportamentos suspeitos dentro do período de validade da chave pública (data de expiração e versão). Caso um dos vizinhos tenha um comportamento suspeito, o nó marca uma acusação para a chave pública vigente do nó. Em seguida, o nó envia uma mensagem de acusação que informa aos seus vizinhos de até um salto de distância todas suas acusações realizadas para seus vizinhos.

O segundo algoritmo, o algoritmo de *harakiri*, é executado por um nó quando este detecta que sua chave privada foi comprometida. Assim, ele envia uma mensagem de *harakiri* a nós de até m-saltos de distância[1], que é a revogação de sua chave pública, além de informar sua identidade, chave pública e privada e a validade da chave pública. Os nós que receberem essa mensagem verificam se as informações são válidas, ou seja, se a chave privada corresponde de fato ao nó que enviou a mensagem. Se as informações forem válidas, os nós adicionam uma marcação de comportamento suspeito relacionado à chave pública correspondente à chave privada.

No terceiro algoritmo, o esquema de acusação, os nós criam suas próprias listas de revogação para uma vizinhança de até m-saltos de distância. Esse algoritmo é dividido em três partes: a criação de listas de revogação, a propagação segura de listas de revogação e como as mensagens de propagação de listas de revogação e de *harakiri* modificam as listas de revogação. Na primeira parte do terceiro algoritmo, a criação das listas de revogação, cada nó cria uma tabela que contém informações de nós de até m-saltos de distância. As informações armazenadas são a identidade, a validade das chaves públicas e suas versões, um *flag* que indica se a chave pública foi revogada, e uma lista de todas as acusações dos nós até m-saltos. O *flag* de revogação de chave pública é '1' quando o nó percebe um comportamento malicioso durante a vigília da vizinhança, ou quando expira a validade da chave pública, ou quando o nó recebe uma mensagem de *harakiri*, ou se a soma de acusações na vizinhança for maior que δ. A escolha de δ deve ser realizada de maneira coerente com a previsão de vizinhança da rede. Sua escolha indica o número de nós que em conluio tentam revogar uma chave pública que o esquema resiste.

Na segunda parte do esquema de acusação, a propagação segura de listas de revogação, o nó envia uma mensagem de acusação com sua lista de revogação para cada um de seus vizinhos diretos todas as vezes que ele realizar uma modificação em sua lista de revogação. As mensagens de acusação são enviadas a cada um

[1]Há um compromisso entre escalabilidade e desempenho com relação à escolha de m.

de seus vizinhos de maneira *unicast*, e são criptografadas com um chaves secretas compartilhadas entre o nó e um vizinho. A chave compartilhada com outro nó é obtida diretamente a partir da chave pública do outro nó e a própria chave privada, sem a necessidade de interações. A obtenção da chave compartilhada é possível por causa do uso da criptografia baseada em identidades.

Por fim, a terceira parte do esquema de acusações evidencia como as mensagens de propagação de listas de revogação e *harakiri* influenciam as listas de revogação. Se um nó recebe uma mensagem de *harakiri* válida, ele adiciona uma acusação para o nó remetente e modifica o valor do *flag* de revogação do nó para '1'. No caso do nó receber uma acusação de outro nó sobre terceiros, o nó verifica se a chave pública do acusador foi revogada e verifica a validade chave compartilhada. Em caso positivo para ambas as verificações, o nó que recebeu a mensagem de acusação armazena a lista de revogação recebida. Quando o nó receber um mínimo de ϵ^2 mensagens de acusações, ele compara as listas de revogação recebidas e verifica a validade das acusações. Assim, o nó atualiza um valor de acusação de sua própria lista de revogação somente quando esse valor for a maioria das listas de revogação recebidas. Esse mecanismo evita a acusações falsas ou equivocadas de serem propagadas.

No esquema de Hoeper e Gong, a obtenção das chaves privadas é realizada de maneira *offline* para uma geradora de chaves privadas (PKG), assim como a renovação dessas chaves quando a primeira expira. Entretanto, essa PKG não consegue distinguir entre nós honestos que tiveram sua chave revogada por terem seus dados privados comprometidos ou por comportamento malicioso. Assim, para minimizar o poder desses nós, um nó só pode requisitar novas versões da chave por um número limitado de vezes. Nesse caso, os nós utilizam suas identidades próprias para fazer essas operações, de maneira que o problema de falsificação e roubo de identidades deve ser tratado através de outros mecanismos.

Chaves Públicas com Certificados Implícitos

O esquema criptográfico baseado em identidades (ID-PKC) possui um problema intrínseco relacionado à entidade geradora de chaves privadas. Como a PKG gera as chaves privadas de todos os usuários, ela possui acesso a todas as chaves criptográficas, então pode decodificar qualquer mensagem criptografada e forjar a assinatura de qualquer usuário. Assim, existe uma forte relação de confiança entre o usuário e a PKG. Al-Riyami e Paterson desenvolveram a criptografia de chaves públicas sem certificados (*Certificateless Public Key Cryptography* - CL-PKC), que se assemelha à ID-PKC para garantir a autenticidade de identidades sem que uma entidade tenha acesso a todo material criptográfico [40, 41]. Nesse sentido, a CL-PKC é um

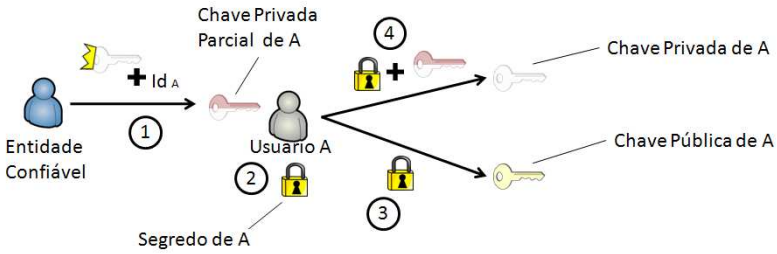
[2] ϵ é um parâmetro de segurança da rede definido no início da rede. Sua escolha indica a resistência de nós honestos a acusações falsas.

esquema criptográfico que se situa entre a ID-PKC e a criptografia tradicional que usa certificados.

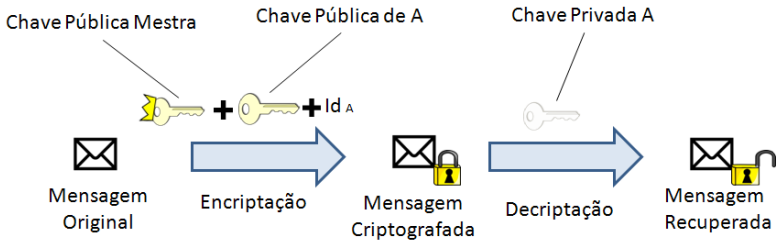
O modelo de criptografia de chaves públicas sem certificados também usa uma entidade confiável geradora de chaves (*Key Generator Centre* - KGC). Diferentemente da PKG, a KGC não possui acesso às chaves privadas dos usuários. A KGC fornece uma chave parcial para um usuário baseada no identificador dele. O fornecimento de chaves parciais deve ser realizado de maneira segura para os usuários correspondentes. Ao receber a chave parcial da KGC, o usuário usa a chave parcial para criar sua chave privada. Dessa maneira, a KGC não tem acesso à chave privada dos usuários. Para a geração das chaves públicas, o usuário combina suas informações secretas com parâmetros da KGC sem a necessidade de gerar sua chave secreta antes. A Figura 2.2(a) mostra o processo de obtenção das chaves privada e pública. É importante notar que a chave pública não pode ser obtida diretamente do identificador e, portanto, não é baseada em identificadores. A forma da divulgação da chave pública segue o padrão tradicional, deve ser transmitida em mensagens ou publicada em um diretório de chaves públicas. A encriptação das mensagens é realizada com a utilização da chave pública mestra, a chave pública do usuário e seu identificador, como mostrado na Figura 2.2(b). Dessa maneira, o uso de certificados é descartado, pois existe a garantia que somente o dono do identificador que foi utilizado para criptografar a mensagem pode decriptar a mensagem, dado que somente ele utilizou a chave parcial privada para gerar a chave privada. Da mesma maneira que na ID-PKC, pode-se incluir a data de expiração nas chaves além do identificador do usuário para limitar a validade e forçar a renovação das mesmas.

Como o esquema criptográfico não possui autenticação de chaves públicas, um atacante poderia gerar chaves públicas e tentar se passar por outro usuário. Esse tipo de ataque falha, pois o procedimento de encriptação utiliza o identificador da entidade, então um atacante é incapaz de decriptar mensagens destinadas a outro usuário, já que não é possível gerar uma chave pública que funcione com o identificador de outro usuário devido à chave parcial entregue pela KGC. Nesse caso, o KGC deve necessariamente ser confiável, pois como ele tem acesso às chaves parciais das entidades ele poderia gerar chaves públicas e privadas correspondentes e se passar por qualquer outra entidade. A diferença entre a KGC e a PKG do esquema criptográfico baseado em identidades (ID-PKC) é no nível de confiança que se deve ter na entidade terceira. A confiança necessária na KGC se limita a acreditar que ela não vai tentar se passar por outra entidade e divulgar chaves públicas falsas. No caso da ID-PKC, as entidades devem acreditar que a PKG não deve abusar do conhecimento das chaves privadas e fazer ataques passivos ou ativos.

Semelhante à criptografia de chaves públicas sem certificados (CL-PKC), Gentry [42] desenvolveu uma noção de encriptação baseada em certificados (*Certificate-*



(a) 1: A partir da chave privada mestra e o identificador do usuário, a entidade confiável gera uma chave privada parcial para um usuário e transmite para ele. 2: O usuário gera um segredo independentemente de qualquer outra chave. 3: O usuário utiliza o segredo para gerar sua chave pública. 4: O usuário gera sua chave privada a partir da chave privada parcial e do seu segredo.



(b) A encriptação é realizada através da chave pública do usuário, seu identificador e a chave mestra. Para decriptar a mensagem, o usuário utiliza sua chave privada.

Figura 2.2: Criptografia sem certificados.

Based Encryption - CBE), na qual a certificação de chaves está implícita na própria chave. Essa abordagem utiliza um certificado emitido por uma autoridade certificadora (CA) como chave de decriptação. Assim, para decriptar mensagens, um usuário precisa tanto da sua chave privada quanto do certificado atualizado. Nesse esquema de criptografia, as próprias entidades geram suas chaves privadas e públicas e requisitam à CA para gerar um "certificado", que funciona como uma segunda chave privada. Assim, os certificados são gerados como as chaves privadas do ID-PKC. A encriptação é realizada da mesma maneira que na CL-PKC, com o uso da chave pública mestra, da chave pública e identificador do usuário. A decodificação é realizada com as duas chaves, a chave privada do usuário e o seu "certificado".

A CL-PKC e a CBE resolvem o problema de certificação explícita, entretanto é vulnerável a um ataque chamado de negação de decriptação (*Denial of Dercryption* - DoD) [43]. Nesse tipo de ataque, um adversário substitui a chave pública de um usuário e divulga para outros usuários que desejam enviar uma mensagem criptografada. Dessa maneira, a entidade que deseja enviar a mensagem criptografada, criptografa a mensagem com o identificador do usuário de destino, mas usa

uma chave pública errada. Assim, o usuário de destino não consegue decriptar a mensagem e o usuário de origem desconhece esse fato. Liu *et al.* propuseram criptografia de chaves públicas com certificados autogerados (*Self-Generated-Certificate Public Key Cryptography* - SGC-PKC) para resolver esse problema sem a perda das vantagens da CL-PKC. Da mesma maneira que a CL-PKC, os usuários recebem uma chave parcial do centro gerador de chaves (KGC) e criam seu par de chaves pública/privada. Além disso, os usuários devem gerar um certificado com sua própria chave privada. O objetivo desse certificado autogerado é similar ao certificado tradicional, atrelar sua identidade e informações pessoais a uma chave pública, de maneira que outra entidade pode verificar que uma chave pública condiz com a identidade anunciada.

A criptografia sem certificados (CL-PKC) foi construída com base o esquema de encriptação da criptografia baseada em identidades (ID-PKC), ou seja, utilizam o mesmo esquema de encriptação baseado Boneh e Franklin de 2001 [36]. Essas operações são consideradas custosas em relação às operações "básicas" de campos finitos como exponenciação modulares. Baek *et al.* [44] desenvolveram um método diferente de encriptação para a CL-PKC que é mais eficiente e seguro.

Lai e Kou [45] adaptaram a encriptação de Baek *et al.* para a criptografia com certificados autogerados (SGC-PKC). Cada usuário gera inicialmente um par de chaves pública e privada primárias, e em seguida envia sua chave pública primária e seu identificador para a KGC. A KGC, de posse da chave pública primária e do identificador do usuário, gera uma chave privada parcial e uma chave pública parcial e envia para o usuário. O usuário cria uma chave privada composta com a chave privada parcial e a chave privada primária. A chave pública composta do usuário é conseguida a partir da chave pública parcial que a KGC gerara com a chave pública primária, e além disso, inclui-se uma assinatura das chaves públicas com sua chave privada composta. Desse modo, um usuário pode verificar a pertinência das chaves públicas em relação a um identificador. O processo de criação de chaves está evidenciado na Figura 2.3. A encriptação e desencriptação são realizadas de maneira semelhante à CL-PKC, exceto que se utiliza a chave pública e a privada composta para tais processos.

Com o esquema SGC-PKC, a chave pública é certificada e atestada. Assim, se atinge o nível 3 de confiança de Girault [46], que diz que além da entidade terceira geradora de chaves (no caso a KGC) não ser capaz de computar chaves privadas de usuários da rede, é possível detectar fraudes da entidade terceira caso ela tente criar chaves de outros usuários. Isso ocorre, pois somente a entidade confiável poderia gerar outra chave pública parcial e falsificar uma chave pública composta (pois somente ela possui a chave privada mestra).

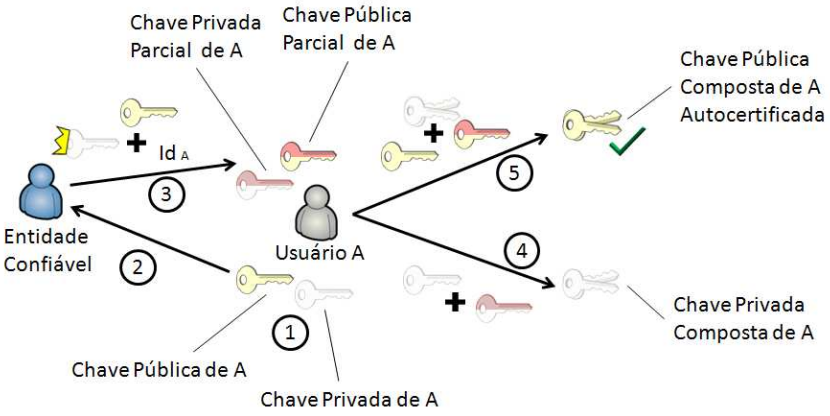


Figura 2.3: 1: O usuário gera um par de chaves pública/privada primárias independentemente de outras chaves. 2: O usuário envia sua chave pública primária para a entidade confiável. 3: A entidade confiável gera as chaves privada e pública parciais do usuário a partir da chave privada mestra, do identificador do usuário e da chave pública primária, que gera independentemente suas chaves públicas e privadas primárias. A partir das chaves privadas primária e parcial, o usuário gera sua chave privada composta. A chave pública composta é gerada a partir da chave pública primária e da chave pública parcial recebida da entidade confiável. Além disso o usuário assina com o uso da chave privada composta a chave pública composta, para certificar que aquela chave pública composta pertence a ele.

Gerenciamento de Chaves Públicas com Certificados Autogerados e Criptografia de Limiar

Lai *et al.* apresentam em [47] um esquema para gerenciamento de chaves e autenticação para redes ad hoc baseados em criptografia de chaves públicas com certificados autogerados (SGC-PKC). Para isso, assume-se que todos os nós possuem uma identidade única e imutável, podem descobrir seus vizinhos de um salto e podem obter as identidades de outros nós na rede. Além disso, a rede possui um par de chaves pública/privada mestras, para o serviço de geração de chaves. A chave pública deve ser conhecida em toda a rede e a chave privada mestra deve ser compartilhada entre todos através de um modelo de criptografia de limiar (k, m) sem a necessidade de uma entidade confiável como em [48]. A geração da chave privada mestra é realizada com a contribuição de todos os nós inicias da rede, de maneira que nenhum deles conheça a chave privada mestra por completo. De maneira semelhante, a chave pública é criada através da junção de partes anunciadas pelos nós iniciais. Deste modo, cada um dos nós iniciais possui a chave pública mestra e uma parte da chave privada mestra. Assim, as operações criptografas que utilizam a chave privada mestra podem ser obtidas por um grupo de pelo menos k nós que possuem as partes de chaves privadas, que podem ser verificadas pela chave pública mestra.

A geração de chaves privadas parciais (do esquema SGC-PKC) para cada um

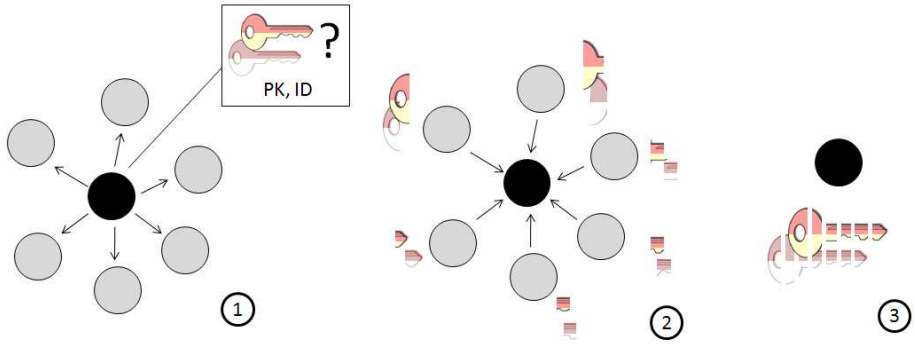


Figura 2.4: 1: Um usuário que entre na rede envia um pedido de chave parcial que inclui seu identificador e sua chave pública aos seus vizinhos. 2: Quando os vizinhos recebem o pedido de chave privada parcial, eles geram uma parte das chaves parciais publica/privada a partir da sua parte da chave privada da rede, com base no identificador e a chave pública do usuário. Em seguida enviam as partes das chaves parciais publica/privada geradas para o usuário. 3: De posse de ao menos k partes de chaves parciais, o nó pode gerar suas chaves parciais que são utilizadas na geração de suas chaves pública e privada compostas.

dos nós é realizada de maneira distribuída. Assim, após o nó criar suas chaves primárias, ele faz uma requisição de chave privada parcial aos seus vizinhos, que inclui seu identificador e sua chave pública. Ao receberem a requisição de chave privada do nó, os vizinhos computam através da parte da chave privada mestra que possuem uma parte da chave privada parcial e uma parte de chave pública parcial, e em seguida a enviam ao nó. De posse de ao menos k partes de chaves privadas e k partes de chaves públicas parciais recebidas de seus vizinhos, o nó pode computar as chaves privada e pública parciais. Assim, o nó pode gerar suas chaves compostas para serem usadas na rede. Esse procedimento está evidenciado na Figura 2.4.

Em um segundo procedimento o nó requisita uma parte de chave privada mestra para poder participar do processo de emissão de chaves para outros nós. Nesse processo, ao menos k nós que possuem uma parte da chave privada mestra enviam um pedaço da parte da chave parcial para o nó, que as junta e forma sua parte da chave privada mestra.

A revogação de chaves é realizada através de quatro mecanismos, a revogação da própria chave, a revogação de nós suspeitos ou comprometidos, um mecanismo de anúncio da lista de revogação e um mecanismo de anúncio de chaves públicas válidas. Esses mecanismos podem ser realizados como o esquema de Hoeper e Gong em [39].

Assim, nesse sistema, por causa do uso desse esquema criptográfico com certificados autogerados, nenhum nó pode obter uma chave privada de outro, mesmo se nós maliciosos em conluio tentarem obter a chave privada. Vale ressaltar também

que mais de k nós maliciosos em conluio podem gerar uma chave pública válida e verificável, mas nesse caso, a existência de duas chaves públicas de uma mesma identidade implicaria automaticamente em que nó maliciosos que participam da PKG distribuída geraram a chave pública. Esse caso é equivalente quando uma autoridade certificadora (CA) falsifica um certificado em uma infraestrutura de chaves pública tradicional, a existência de dois certificados válidos implica que a CA não agiu adequadamente.

Controle de Acesso baseado em Grupos Dinâmicos e Auto-organizados (*A Controller-node-based Access-Control mechanIsm for Ad hoc networks* - ACACIA)

Como a autenticação em redes ad hoc móveis deve ser feita sem qualquer tipo de controle centralizado, entidades certificadoras distribuídas não são suficientes para garantir a segurança em redes ad hoc. Fernandes *et al.* [49] propõe ACACIA (*A Controller-node-based Access-Control mechanIsm for Ad hoc networks*), um sistema distribuído de controle de acesso e autenticação sem a necessidade de uma CA. O sistema é auto-organizado e gerencia as chaves público-privadas e as identidades, além de controlar a entrada de nós na rede e punir os nós não colaborativos. ACACIA evita o uso de um administrador central para controlar o acesso dos nós com o uso de cadeias de delegação que controlam a entrada de nós na rede. Além disso, o sistema autogerencia o grupo de controle na inicialização e partição da rede.

A criação de uma rede ad hoc é motivada por um grupo com interesse ou relações comuns, como trabalhadores de uma mesma empresa, militares, amigos, etc. Eles utilizam suas relações sociais para criar a cadeia de delegação. A cadeia de delegação realiza controle de acesso não centralizado baseado nas relações entre os usuários. Assim, qualquer um pode criar uma nova rede e se tornar o nó raiz da cadeia de delegação, ou pode se associar a uma rede já existente ao obter um convite de outro usuário. Desta forma, cada usuário tem um número determinado de convites que ele pode distribuir para novos membros, que se tornam seus filhos na cadeia de delegação. O usuário envia um convite *offline* para o novo membro e transfere alguns de seus próprios convites (ou nenhum) para o novo membro de acordo com a confiança que tem nele. Esse procedimento limita o número de novos membros que um usuário e seus convidados podem chamar para rede, portanto, reduz a possibilidade de um usuário não confiável ganhar convites para convidar novos membros.

Esse sistema associa a cada nó da rede um grupo de nós controladores, que avalia o comportamento do nó, emite certificado para que o nó possa participar da rede e exclui o nó da rede quando esse se comporta inadequadamente. O grupo de nós controladores de um nó é escolhido randomicamente para evitar manipulação, e é redefinido a cada entrada ou saída de nós. Para cada nó, são definidos dois grupos de

controladores, os controladores de nó que avaliam o comportamento do nó e decidem se ele deve ser expulso da rede ou não, e os controladores de usuário que controlam a cadeia de delegação. Todas as decisões são tomadas por meio de votações, nas quais a maioria deve concordar para que se faça uma ação.

O grupo de controladores de usuário tem como função garantir a validade da cadeia de delegação. Então, o grupo verifica a validade dos convites e controlam o número de filhos. Assim quando um nó deseja conectar-se com a rede, ele cria uma identidade (chave pública) requisita um certificado para seus controladores de usuário, que por sua vez verificam a consistência do convite, da relação da cadeia de delegação do pai para poder gerar o certificado. Nesse esquema, o certificado é a agregação de assinaturas dos controladores em cima da identidade do nó, de maneira que reúna assinaturas de pelo menos a maioria dos controladores de usuário.

Todos os nós executam um sistema detector de mau comportamento (*Bad Behavior Detection System* - BBDS) que deve perceber os ataques de seus vizinhos. Quando um dos vizinhos do nó detectar que o nó se comporta mal, ele envia uma mensagem de acusação a respeito do nó aos controladores de nó. Os controladores de nó executam um sistema de reputação que diminui o valor de reputação de acordo com as mensagens de acusação recebidas. Quando o valor de reputação é menor que um limiar mínimo, os controladores de nó votam pela exclusão do nó da rede. O voto e inundado na rede, e quando os nós obtiverem votos da maioria do grupo de controladores, eles ignoram todos os pedidos e mensagens do nó excluído. Quando a grupo de controladores de usuário excluir o nó, eles revogam o certificado, e por sua vez votam pela exclusão dos filhos do nó. O procedimento é repetido até que todos os descendentes do primeiro nó excluído também o sejam. Além disso, nós cujos descendentes foram excluídos sofrem uma punição, ao ter um acréscimo do seu limiar mínimo de reputação.

Assim, o sistema provê segurança para a rede com o uso de grupos de controle auto-organizados e dinâmicos. O sistema controla a entrada dos nós, monitora e expulsa os nós que não se comportem de acordo.

Comparação das Propostas de Controle de Acesso e Autenticação

A Tabela 2.1 mostra uma comparação das diversas propostas de controle de acesso e autenticação. Como pode-se perceber, todas as propostas utilizam mecanismos criptográficos, que tem como objetivo proteger a comunicação entre os nós. Além disso, todas as propostas possuem algum tipo de certificado para o controle de acesso à rede, que relacionam as chaves criptográficas às identidades. Assim, somente de posse do certificado, o nó pode participar da rede. Então, os mecanismos para exclusão da rede normalmente baseiam-se em revogação ou expiração dos certificados.

Zhou e Haas e Luo *et al.* utilizam a RSA com criptografia de limiar (k, m) para fazer a emissão de certificados. A segurança da certificação das identidades desses esquemas está justamente no fato de que a geração de certificados é distribuída, e somente quando k de m geram os certificados parciais consegue-se o certificado completo. Contudo, nessas propostas basta que k nós maliciosos se juntem para gerar certificados na rede para qualquer nó. Em ambos mecanismos, os certificados possuem um tempo de validade, de maneira que os nós devem renovar periodicamente os certificados. Assim, se um nó deixar de se comportar bem, pode-se excluí-lo da rede ao deixar de gerar um novo certificado. Além desse mecanismo, a proposta de Luo *et al.* possui um mecanismo explícito para a revogação de certificados baseado em mecanismo de monitoramento, para acelerar o processo de exclusão.

Hoeper e Gong e Khalili *et al.* usam a criptografia baseada em identidades. Dessa maneira, a certificação das identidades é implícita, pois as mensagens são criptografadas com as identidades e assim, somente quem possuir a chave privada relacionada à identidade pode decriptar as mensagens. Ambas as propostas incluem mecanismos de monitoramento para a revogação explícita de chaves. As propostas incluem também mecanismos de expiração de certificados, de maneira que a entidade geradora de chaves privadas (PKG) deve gerar periodicamente novas chaves privadas para os nós. Na proposta de Hoeper e Gong, a entidade geradora de chaves privadas é externa ao mecanismo, então ela não tem como distinguir pedidos de renovação de chaves de nós cujo certificado foi expirado normalmente dos nós cujo certificado foi revogado, e assim nós mal comportados podem se associar novamente na rede. Na proposta de Khalili *et al.* a entidade distribuída na rede através da criptografia de limiar (k, m), na qual cada um do grupo de nós iniciais possui uma chave parcial. Dessa maneira, um conluio de pelo menos k nós pode comprometer a segurança da rede.

Na proposta de Lai *et al.* utiliza-se a criptografia com certificados autogerados. Assim, os próprios nós atestam suas identidades, e só podem decriptar as mensagens se receberam o material criptográfico gerado pela entidade geradora de chaves (KGC). Assim como a proposta de Hoeper e Gong, o esquema de Lai *et al.* apresenta um mecanismo para a revogação explícita de chaves e expiração de chaves, com a diferença que a KGC é distribuída através da criptografia de limiar. Então, assim como as outras propostas, um conluio de pelo menos k nós pode comprometer a segurança da rede.

Fernandes *et al.* apresentam um sistema de controle de acesso baseado em cadeias de delegação e grupos de controle distribuídos aleatoriamente. Assim como outras propostas, a participação é garantida através de certificados e pode ser revogada através de mesanismos de monitoramento. Nesse sistema, cada nó possui seu próprio grupo de controle que gera e revoga os certificados, o qual é escolhido aleatoriamente

na rede e é modificado a cada entrada e saída da rede. Assim, o conluio para geração e revogação de certificados é bem dificultado. Outra característica desse sistema é que o controle de acesso é baseado nas relações entre os usuários. Assim, um usuário só pode participar da rede caso tenha sido convidado, e assim forma uma cadeia de delegação. Para evitar a banalização do convite, o sistema faz com que o número de convites seja limitado, e também pune os nós cujos descendentes tenham sido punidos por mau comportamento com a justificativa de serem responsáveis pelo convite do nó mal comportado. Assim, os nós realizam a distribuição de convites com mais responsabilidade para não serem eles mesmos punidos.

Tabela 2.1: Comparação das propostas de controle de acesso e autenticação

Propostas	Criptografia	Certificação da Identidade	Controle de Acesso
Zhou e Haas	RSA com criptografia de limiar	Geração de cerificados com criptografia de limiar distribuidamente em servidores especializados *on-line*	Expiração de certificados (renovação periódica de chaves parciais privadas para reemitir certificados)
Luo *et al.*	RSA com criptografia de limiar	Geração de certificado com criptografia de limiar distribuída localmente	Expiração e revogação explícita de certificados com uso de monitoramento e modelo de confiança localizado
Khalili *et al.*	Criptografia de identidades com criptografia de limiar	Implícita	Expiração de certificados e mecanismos de monitoramento de mau comportamento com revogação de certificados
Hoeper e Gong	Criptografia de Identidades	Implícita	Geração de chaves privadas *off-line*, expiração de certificados e mecanismos de monitoramento de mau comportamento com revogação de certificados (*harakiri* e esquema de acusação)
Lai *et al.*	Chaves Públicas com Certificados Autogerados e Criptografia de Limiar	Certificado autogerado validado localmente	Expiração de certificados e mecanismos de monitoramento de mau comportamento com revogação de certificados (*harakiri* e esquema de acusação)
Fernandes *et al.* (ACACIA)	RSA	Controladores de usuário e cadeia de delegação	Monitoramento, controladores de nó e de usuário, cadeia de delegação e sistema de reputação

2.3 Cooperação em Redes Ad Hoc

A rede ad hoc é uma rede formada, criada, operada e gerenciada pelos próprios nós. Os nós ajudam-se uns aos outros com o encaminhamento de dados e mensagens de controle de um nó para outro, normalmente para um nó de destino além do alcance do rádio do nó de origem. Dessa maneira, a execução e sobrevivência da rede depende somente da natureza cooperativa e confiável dos nós [50]. Entretanto, a dependência de nós intermediários faz a rede ad hoc ficar vulnerável a diversos ataques, passivos e ativos, e mesmo que sistemas e mecanismos criptográficos sejam utilizados, não há garantia de cooperação e entre os nós.

Basicamente, existem dois tipos de nós que possuem comportamento não cooperativo: nós maliciosos/defeituosos e nós egoístas [51]. Os nós maliciosos/defeituosos são aqueles que ou estão com defeitos e não podem seguir os protocolos, ou podem intencionalmente ser maliciosos e tentar atacar o sistema, os mecanismos e protocolos da rede. Os nós egoístas são nós economicamente racionais cujos objetivos é maximizar a eficiência de suas ações, de maneira a maximizar o benefício próprio e ao mesmo tempo reduzir os custos das ações. Então, como ações de encaminhamento de pacotes possuem custos (de energia e outros recursos) e não trazem benefício direto ao nó egoísta, ele precisará de estímulo para realizar esse tipo de ação.

A possibilidade da presença de nós egoístas nas redes ad hoc cria a necessidade de um mecanismo que estimule os nós da rede cooperarem, e prevenir que os nós adquiram comportamentos egoístas. Assim diversos mecanismos de estímulo à cooperação foram desenvolvidos para tentar evitar a presença de nós egoístas. Além disso, os mecanismos de estímulo à cooperação devem incluir segurança, para evitar que nós maliciosos possam atrapalhar o funcionamento da rede e dos mecanismos de estímulo à cooperação.

Os mecanismos de estímulo à cooperação normalmente utilizam sistemas de reputação/confiança e de troca de créditos que estimulam o encaminhamento de pacotes nas redes ad hoc. A seguir alguns mecanismos de estimulo à cooperação são apresentados baseados em sistemas de reputação, confiança, trocas de créditos e alguns alternativos.

2.3.1 Sistemas de Reputação/Confiança

Um dos grandes problemas de redes ad hoc é o fato de entidades distribuídas realizarem ações sem a confirmação da colaboração por parte de outras. Quando as entidades não sabem como podem confiar em outras, elas normalmente acreditam ingenuamente nas boas intenções das outras entidades, e por causa disso podem sofrer diversos tipos de ataques. Sem o quesito de confiança, os nós devem confiar tarefas da rede para nós possivelmente não confiáveis, que pode causar falhas críticas

na rede, como o próprio roteamento [52].

A classificação acurada do comportamento de nós pode ser obtida através do uso de sistemas e modelos de confiança e reputação [52]. A ideia de sistemas e modelos de confiança e reputação é ter um valor para quantificar a confiabilidade e a competência de um nó da rede, baseado em técnicas de monitoramento. Essa informação é então utilizada para avaliar a cooperatividade dos nós da rede, além de quesitos como segurança, disponibilidade, eficiência do roteamento, etc. E assim, os nós que possuem comportamentos inadequados para as aplicações da rede são punidos.

A reputação e a confiança, apesar de serem termos relacionados, são conceitos distintos. Martignon *et al.* [53] definem que a reputação é a percepção que um nó cria sobre outros nós a partir de informações de terceiros que possuem interações passadas com os nós avaliados. Por sua vez, o conceito de confiança representa a expectativa que um nó possui das ações futuras de outros nós. Assim, a confiança é medida a partir de uma avaliação própria de suas interações com outros nós e outros fatores, como o tempo decorrido desde a última medição de reputação.

Nos sistemas de reputação, nós da rede monitoram outros nós, definem valores de confiança uns para os outros, e trocam opiniões entre eles para formar as reputações. A reputação é, portanto, um valor atribuído a um nó por terceiros, ou seja, a "opinião" que terceiros têm sobre um nó. Um nó que não se comporta adequadamente tem um valor de reputação reduzido, e assim, ele pode ser isolado pelos outros nós. As críticas desses tipos de sistemas são que os nós devem propagar uma opinião de segunda mão, então trazem novos problemas de segurança. Outro ponto é que esse tipo de sistema não resiste a ataques em coluio [54]. Além disso, os sistemas de confiança e reputação estão sujeitos a cinco tipos de ataque: ataques de mentirosos, ataques de bom e mau comportamentos alternados, ataque de comportamento conflitante, ataque de Sibil e de recém-chegado [52].

No ataque de mentirosos, um nó malicioso "mente", passa uma informação falsa, e se aproveita do fato de que outros nós consideram sua opinião para calcular os valores de reputação. Assim, ele faz recomendações falsas para diminuir a confiabilidade de um nó cooperativo ou aumentar a de um nó malicioso ou não cooperativo. Já o ataque de bom e mau comportamentos alternados, o nó comporta-se bem e mal alternadamente para evitar que o sistema de confiança e reputação o acuse de ter um mau comportamento. De maneira semelhante, no ataque de comportamento conflitante, o nó assume comportamentos distintos para diferentes grupos de usuários. Assim, ele tenta sabotar o sistema de reputação e confiança ao fazer as opiniões dos outros nós que dizem respeito a ele serem conflitantes. Já o ataque de Sibil e recém-chegado, os nós criam identidades falsas para tentar alterar a reputação de um nó através das recomendações, ou as utilizam para apagar o histórico de seu

comportamento.

A seguir, alguns esquemas de estímulo a cooperação baseado em sistema de reputação e confiança são descritos com mais detalhes.

Watchdog e Pathrater

Marti *et al.* [3] propuseram a criação de extensões para um protocolo de roteamento para detectar e eliminar os comportamentos egoístas e maliciosos no roteamento. Os autores apresentam dois módulos extras para o protocolo de roteamento DSR (*Dynamic Source Routing*), *watchdog* e *pathrater*. O *watchdog* identifica os nós mau comportados e envia para o *pathrater*, que evita o roteamento de pacotes por esses nós.

O módulo de *watchdog* de um nó verifica se o próximo salto do caminho também encaminha os pacotes. Para tal, o *watchdog* coloca a interface de rede sem-fio do nó no modo promíscuo e verifica se o próximo salto encaminha de fato o pacote. Se o nó percebe que o próximo salto não encaminhou os pacotes que deveria, ele considera o próximo salto está com mau comportamento. Dessa maneira, cada nó possui uma lista com os pacotes que foram encaminhados recentemente e seus respectivos próximos saltos, caso o pacote não tenha o próximo salto como destino. Quando o nó escuta um vizinho retransmitir um dos pacotes da lista, ele o remove da lista. Caso o pacote permaneça na lista por determinado tempo, o nó remove o registro da lista e incrementa um contador de falhas para o nó que deveria ter encaminhado. Ao perceber que as falhas ultrapassam um limiar de banda máximo, o nó avisa ao nó de origem dos pacotes do mau comportamento. Apesar do módulo de *watchdog* permitir a detecção de mau comportamento a nível de encaminhamento, existem alguns problemas que prejudicam seu funcionamento. Colisões de quadros e baixa potência de envio de pacotes podem fazer com que nós não consigam detectar se o vizinho realizou encaminhamentos e retransmissões de pacotes corretamente. Além disso, nós em uma rota poderiam se juntar e combinar de não avisar o nó de origem que um deles descarta pacotes.

O *pathrater* usa a informação enviada pelo *watchdog* para escolher o caminho no qual o pacote tem a maior probabilidade de ser entregue corretamente. Cada nó mantém uma avaliação da confiabilidade de cada outro nó que ele conhece na rede, e no cálculo de rotas ele considera a confiabilidade dos nós do caminho. Ao realizar a descoberta de rotas, o algoritmo de cálculo de rotas atribui valores de confiabilidade neutros para os novos nós descobertos nesse processo. Esse procedimento garante a escolha da rota mais curta até o destino, pois todos os nós têm o mesmo valor de confiabilidade. O algoritmo de cálculo de rotas também incrementa periodicamente o valor de confiabilidade de caminhos ativos, pois historicamente representaram uma boa escolha. Da mesma maneira, o algoritmo decrementa o valor de confiabilidade

dos nós quando detecta uma falha no enlace e o nó fica inalcançável. As rotas calculadas que contêm que os nós considerados mal comportados são sempre descartadas pelo algoritmo, e como o mau comportamento de um nó pode ser causado por falhas pontuais, com o passar do tempo o algoritmo reintegra os nós mal comportados no cálculo de rotas.

Os módulos de *watchdog* e *pathrater* podem ser anexados em diferentes protocolos de roteamento, mas funcionam melhor em protocolos de roteamento nos quais se conhece todos os nós do caminho, como roteamento por fonte e por estado de enlace. No caso do *watchdog*, um nó poderia descobrir comportamentos maliciosos ao verificar se o próximo salto encaminha para o próximo corretamente. No caso do *pathrater*, o nó deve conhecer todos os nós pelos quais os pacote passam para calcular corretamente a métrica do caminho.

É importante ressaltar que os mecanismos *watchdog* e *pathrater* detectam os nós não cooperativos e os retiram das rotas. Entretanto, os nós mal comportados ainda participam da rede e podem enviar e receber pacotes, além de economizarem energia. Não há nenhum mecanismo que os obrigue a cooperarem, e quando não cooperam, eles não sofrem nenhuma punição. Assim, se o cenário da rede for tal que os nós querem maximizar os próprios ganhos, e que existem nós que realizam ações intencionais para atrapalhar a rede, a rede não funcionaria corretamente. Um mecanismo que puna e até exclua os nós com mal comportamento se faz necessário.

O Mecanismo CORE

Michiardi e Molva propuseram CORE, um mecanismo de reputação colaborativa para estimular a cooperação entre os nós baseado em uma técnica de monitoramento colaborativo [55]. CORE foi proposto para ser um mecanismo genérico que possa ser integrado com qualquer funcionalidade da rede, como encaminhamento de pacotes, descoberta de rotas, gerência de rede e de local. Cada entidade da rede com CORE mantém uma reputação para as outras entidades, que é calculada a partir do monitoramento realizado pela própria entidade e de informações enviadas por outras entidades envolvidas em cada operação da rede. O modelo colaborativo do mecanismo proposto evita ataques de negação de serviço baseados em difamação, em difusão de avaliações negativas para nós legítimos.

O mecanismo utiliza um modelo de reputação para avaliar a cooperatividade de uma entidade. Quando uma entidade possuir reputação alta, pode utilizar os recursos da rede, e quando possuir um baixo valor de reputação, a entidade é excluída da comunidade. A reputação é composta de três tipos: a reputação subjetiva, a reputação indireta e a reputação funcional. A reputação subjetiva é calculada diretamente das observações de um indivíduo. O valor da reputação subjetiva depende da média ponderada das observações, na qual o passado possui maior relevância para evitar

que eventos esporádicos recentes de mau comportamento tenham importância no cálculo da reputação. A reputação indireta representa como a opinião de outras entidades afeta a reputação de certo indivíduo. O outro tipo de reputação evidencia o fato de uma entidade ter comportamentos distintos na realização das diferentes tarefas da rede. É assim representado pela reputação funcional, que compreende da soma das avaliações de reputação subjetivas e indiretas em relação a uma função da rede. Deste modo, as entidades possuem um conjunto de tabelas de reputação para cada função na rede, que armazenam as reputações subjetivas e as indiretas recentes. O valor global de reputação de uma entidade é obtido com a soma ponderada das reputações funcionais, e a confiabilidade desse resultado de reputação é representa pelo número de avaliações a variância das avaliações usadas para o cálculo de reputação.

CORE possui um mecanismo de monitoramento, que funciona como o *watchdog* [3]. O mecanismo de *watchdog* tem para cada função da rede uma lista uma ações esperadas para as entidades monitoradas. Quando uma entidade faz uma requisição de uma função para outra entidade, ela adiciona a ação esperada na lista da função e espera o resultado da execução da função. Se a entidade monitoradora perceber que a função foi executada como esperado, ela a retira da lista. A reposta da execução contém todas as entidades que cooperam e participaram corretamente para a realização da função, então a entidade atualiza positivamente a reputação indireta delas. A contribuição da reputação indireta deve ser positiva para evitar uma possível difamação contra nós legítimos. Se a função não for executada como esperado ou permanecer por muito tempo na lista, a entidade avalia negativamente a entidade monitorada e atualiza a reputação subjetiva. Quando a reputação geral de uma entidade é negativa, as entidades recusam quaisquer requisições da entidade com má reputação, então para participar da rede a entidade deve realizar as funções requisitadas. Além disso, a reputação positiva só é conseguida através da reputação indireta, ou seja, quando a função foi realizada com sucesso. Assim a boa reputação é difícil de ser conseguida, e por isso as entidades devem cooperar ainda mais.

O protocolo CONFIDANT (*Cooperation Of Nodes: Fairness In Dynamic Ad hoc NeTworks*)

O CONFIDANT (*Cooperation Of Nodes: Fairness In Dynamic Ad hoc NeTworks*) [56] é um protocolo baseado em reputação para encontrar e isolar nós maus comportados que não cooperam e atrapalham o protocolo de roteamento. O protocolo CONFIDANT consiste de quatro módulos: monitor, sistema de reputação, gerente de confiança e gerente de caminhos. O módulo de monitoramento detecta passivamente maus comportamentos de outros nós da rede. Uma forma de se fazer isso é a realização do monitoramento local do encaminhamento dos pacotes como

em [3]. Assim, o nó verifica se o próximo salto encaminha corretamente os pacotes, e também verifica se não modifica os pacotes encaminhados.

Outro módulo, o gerente de confiança, envia e recebe mensagens de alarme que informam sobre maus comportamentos detectados. Assim, quando um nó percebe um mau comportamento, ele envia uma mensagem de alarme para nós pré-determinados em uma lista de nós confiáveis. Um nó que receba um alarme checa a confiabilidade do nó que enviou o alarme, para aceitá-lo ou não.

O terceiro módulo, o sistema de reputação, reúne as informações de monitoramento do módulo de monitoramento e os alarmes recebidos pelo gerente de confiança, e a partir desses dados, calcula valores de reputação para os nós. Além disso, o sistema de reputação contém listas negras, para os nós que não podem ser utilizados para certas funções da rede como encaminhamento e anúncio de rotas. As listas e valores de reputações podem ser trocados entre os nós, mas para o cálculo do valor de reputação, os nós sempre dão mais valor à própria experiência com os nós.

Por fim, o módulo gerente de caminho analisa os possíveis caminhos tomados pelos pacotes em relação à segurança. Assim, o módulo recusa mensagens de roteamento proveniente de nós considerados maliciosos pelo sistema de reputação. Esse módulo funciona melhor com protocolos de roteamento por fonte, nos quais o nó de origem tem o controle do caminho completo por onde o pacote passa.

O protocolo CONFIDANT considera que as experiências negativas são exceção ao funcionamento normal da rede, então só as considera para o cálculo da reputação, e após um determinado tempo sem detecção de ações suspeitas descarta as experiências negativas. Para o protocolo CONFIDANT funcionar corretamente, os nós da rede ad hoc devem operar os quatro módulos, e assim, o protocolo é capaz de detectar, alertar e evitar nós maliciosos no encaminhamento das mensagens de dados e de controle.

O Protocolo de Roteamento CBTRP (*Cluster Based Trust-aware Routing Protocol*)

Os protocolos de roteamento de redes ad hoc móveis normalmente assumem que os nós da rede possuem comportamentos benevolentes e não consideram relações de confiança entre os nós, o que deixa a rede vulnerável a comportamentos maliciosos e egoístas. Entretanto, como a segurança da comunicação de um nó depende somente da escolha do caminho para se chegar ao destino, é importante que um nó conheça a confiabilidade dos nós que formam os caminhos dos pacotes que envia. Assim, Safa *et al.* [57] propuseram um protocolo de roteamento consciente da confiança baseado em *clusters* (*Cluster Based Trust-aware Routing Protocol* - CBTRP), que protege os pacotes de nós maliciosos intermediários ao tentar rotear os pacotes somente através de nós confiáveis.

No CBTRP, os nós desenvolvem uma relação de confiança que evolui com o tempo e com a frequência das interações entre os nós. A confiança em outro nó é estabelecida com base na informação que se pode extrair sobre o outro, como a análise de pacotes enviados e encaminhados. A confiança que um nó tem em um vizinho depende de três fatores, a probabilidade de o vizinho fazer uma ação positiva, a probabilidade de o vizinho fazer uma ação negativa e a incerteza dessas medidas. As probabilidades são calculadas a partir das ações observadas do vizinho, tanto as positivas e negativas.

Outro ponto do CBTRP é a rede ser organizada em *clusters* disjuntos de um salto, nos quais cada nó elege o vizinho mais qualificado e confiável para ser o chefe do *cluster*. Os nós membros do *cluster* só enviam pacotes para esses chefes de *cluster* para garantir o caminho seguro de pacotes, e assim que o chefe do *cluster* deixa de ser confiável, um novo chefe é eleito. Ao deixarem de ser confiáveis, os nós são excluídos da rede, seus pedidos agora não são processados, além de não se encaminhar os pacotes provenientes e destinados a eles. Como a rede é dividida em *clusters*, os nós podem ser membros ou o chefe de um *cluster*. O chefe do *cluster* armazena a confiança de todos os nós membros do *cluster*, e estes armazenam a confiança do nó chefe e a monitoram constantemente. A escolha inicial do chefe de cluster depende do número de vizinhos, da distância média para os vizinhos, velocidade do nó e da energia da bateria como em [58].

Dessa maneira, a rede é organizada em *clusters* nos quais cada um tem um nó eleito mais confiável que é o chefe de *cluster*. Como cada *cluster* é separado com até um salto de distância, os pacotes são encaminhados entre os chefes de *cluster*, ou seja, seguem o caminho pelos nós mais confiáveis da rede. Assim, com esse protocolo de roteamento baseado em *cluster* é possível a construção de rotas confiáveis até o destino.

O Sistema ACACIA (*A Controller-node-based Access-Control mechanIsm for Ad hoc networks*)

O sistema ACACIA descrito em 2.2.2 realiza o controle de acesso e autenticação por meio do grupos de controladores. Para realizar o controle de acesso, o sistema de utiliza dois grupos, os controladores de usuário que geram certificados para a participação da rede e os controladores de nós que excluem um nó da rede de acordo com o seu comportamento. O grupo de controladores de nó decide se o nó deve ser excluído de acordo com um sistema de reputação, portanto o sistema é considerado um sistema de estímulo à cooperação baseado em reputação. Assim, os nós devem se comportar adequadamente e agir cooperativamente para continuar na rede, pois caso não o façam são expulsos da rede.

Contudo, para a exclusão de nós que sejam realmente não cooperativos, o sistema

depende fortemente do sistema de detecção de mau comportamento (BBDS). Assim, os nós enviam mensagens de acusação todas as vezes que seus BBDSs acusarem um de seus vizinhos. Essa abordagem causa um número grande de mensagens de acusação para gerar os valores de reputação, que são custosas por serem destinadas a nós espalhados na rede.

Modelo de Confiança baseado em Interações Humanas

Velloso *et al.* [59] apresentam um modelo de confiança flexível para ser usado em redes ad hoc móveis, e assim permitir a escolha de somente os nós dignos de confiança para a interação. Assim, a autonomia da rede é assegurada, pois nós mal comportados não conseguem participar da rede.

O modelo de confiança proposto por Velloso *et. al* é baseado em no conceito de confiança entre humanos, cujas relações de confiança são construídas ao longo do tempo. Assim, os nós da rede constroem uma relação de confiança entre seus vizinhos, que se baseia na experiência individual passada e nas recomendações de seus vizinhos acerca dos outros. As recomendações melhoram o processo de avaliação de confiança para os nós que falham em observar as ações de seus vizinhos, e devido à limitações de recursos e interrupções na comunicação. A capacidade de avaliar a confiança de seus vizinhos permite os nós isolarem os mal comportados, e permite a escolha de interação com os vizinhos mais confiáveis de maneira que os nós devem agir corretamente para que possam participar da rede.

Uma importante característica do modelo é a atuação unicamente local, pois os nós interagem e armazenam informações somente dos seus vizinhos. Desse modo, os nós não precisam armazenar informações de todos nós na rede e assim se economiza recursos dos nós. A característica local também faz com que a atuação seja limitada na vizinhança e, portanto, diminui o impacto na rede. Além disso, diminui-se a probabilidade de recomendações falsas, pois a troca de recomendações é realizada entre vizinhos diretos sem intermediários para aumentar a incerteza da informação. Outro ponto importante desse modelo é a introdução do conceito de maturidade da relação que aumenta a eficiência do processo de cálculo da confiança quando há mobilidade. Dessa maneira, utiliza-se o tempo da relação entre o nó recomendador e o no recomendado como métrica para pesar a recomendação. Os autores ainda propõem um protocolo para a troca de recomendações (*Recommendation Exchange Protocol* - REP).

Dessa maneira, o modelo de confiança proposto pelos autores utiliza-se da própria experiência com seus vizinhos e da troca local de experiência para calcular um valor de confiança para seus vizinhos. Contudo, o caráter local desse modelo permite que nós maliciosos se movam para novas localidades e construir novas relações de confiança com outro nós. Além disso, o sistema depende de outros sistemas de

controle de acesso para garantir a consistência das permissões de participação da rede e da ação de rejeição de nós não confiáveis.

2.3.2 Sistemas de Trocas de Créditos

Outro tipo de sistemas de estímulo à cooperação são os sistemas baseados em trocas de créditos. Esses sistemas baseiam-se na premissa de que certas operações em redes não trazem benefícios diretos para os participantes, assim os sistemas criam formas artificiais de beneficiar os que cooperam. Para tal, os sistemas baseados em trocas de créditos utilizam uma unidade monetária para a execução de serviços de rede, que são obtidos somente com a cooperação. Um exemplo de serviço na rede que deve ser incentivado e pode ser trocado por créditos é o encaminhamento de pacotes. Nesse caso, os nós ganham créditos quando realizam encaminhamentos de pacotes e podem utilizá-los para comprar de outros nós o encaminhamento dos seus próprios pacotes. Assim, se um nó não cooperar com o encaminhamento de pacotes dos outros, ele não consegue com que seus próprios pacotes sejam encaminhados porque não obtém créditos para pagar a outros nós para encaminharem seus pacotes.

Esses sistemas de troca de créditos precisam de sistemas de gerência de créditos como "*internet banking*", que normalmente são muito complexos. Além disso, um nó que tenha créditos suficientes para encaminhar os próprios pacotes, pode começar a descartar os pacotes dos outros [54]. É importante ressaltar que como os sistemas de trocas de créditos visam evitar comportamentos egoístas, normalmente não possuem mecanismos para evitar comportamentos maliciosos. A seguir alguns mecanismos de estímulo à cooperação baseados em trocas de créditos são descritos com mais detalhes.

Mecanismo de Estímulo com Processador Seguro

Com o progresso da tecnologia, é possível criar redes ad hoc móveis para aplicações civis, como comunicação em áreas remotas. Nessas redes, os nós normalmente são independentes e não pertencem a uma autoridade só, então usam a rede com objetivos distintos, portanto, não se pode assumir que os nós cooperem entre si. Na realidade, ocorre justamente o contrário, para economizar seus próprios recursos, os nós agem egoisticamente. Butyan e Hubaux abordam a questão de cooperação de redes ad hoc móveis para aplicações civis [60, 61].

Uma primeira abordagem possível é fazer com que os nós sejam invioláveis, e que o comportamento não pode ser alterado. Entretanto, essa suposição é irreal, se não for também impraticável. Assim, foi proposta uma abordagem que utiliza um módulo de *hardware* inviolável, chamado de módulo de segurança. O módulo de segurança pode ser um *smart card* ou um processador seguro [62]. Assim, usuário

pode modificar o comportamento do nó, mas não pode alterar o módulo seguro. Uma das funções mais básicas de redes ad hoc que necessita de cooperação para funcionar corretamente é o encaminhamento de pacotes, então o módulo seguro tem como objetivo estimular essa funcionalidade. Para isso, todos os pacotes que passam pelo nó, tanto gerados, quanto encaminhados, devem passar pelo módulo seguro. O módulo seguro tem um contador, cujo valor é decrementado quando o nó quer enviar pacotes, e incrementado quando o nó encaminha um pacote. O encaminhamento de um pacote incrementa o contador de uma unidade, e ao enviar um pacote, o contador decrementa de um número calculado como uma estimativa do número de saltos que o pacote precisa para chegar ao destino. Assim, o comportamento egoísta pode ser evitado, pois o nó deve encaminhar para que seu contador seja positivo, para que o nó consiga enviar seus pacotes. Além disso, esse esquema estimula a manutenção dos nós ligados para encaminhar os pacotes dos outros e desestimula a geração de muitos pacotes para nós distantes, propriedade desejável uma vez que a banda disponível por nó diminui com o aumento de nós na rede (se considerar que o tráfrego não é essencialmente local) [63].

Cada módulo de segurança vem com um par de chaves privada e pública, e um certificado emitido pelo fabricante do módulo de segurança embutidos. Dessa maneira, é possível validar as chaves públicas e criar associações seguras entre os nós. Outra funcionalidade do módulo seguro é garantir o envio correto de pacotes e a proteção contra a manipulação do contador. Para isso, o módulo de segurança tem uma função especial para o encaminhamento de pacotes. Quando dois nós se tornam vizinhos, eles fazem uma associação segura através de seus módulos de segurança, associação tal que contém o identificador do vizinho, uma chave de seção e o número de pacotes enviados e recebidos. Assim, um pacote para ser transmitido, tanto o pacote originado no nó quanto o encaminhado, passa antes no módulo de segurança. O módulo de segurança decrementa seu contador caso seja a origem do pacote e cria um cabeçalho de segurança com as informações da associação segura com o próximo salto. Quando o pacote chega ao módulo de segurança do próximo salto, ele verifica a validade das informações do cabeçalho de segurança, e incrementa um valor de pacotes corretamente recebidos do salto anterior. Periodicamente em uma etapa de sincronização de contadores, os nós enviam os valores de pacotes corretamente recebidos para seus vizinhos, para que atualizem seus contadores. Dessa maneira, os contadores são incrementados somente na etapa de sincronização de contadores, e não imediatamente na hora do encaminhamento, o que permite a atualização de contadores quando de fato ocorreu o encaminhamento e o pacote chegou corretamente no próximo salto.

Esse mecanismo estimula o encaminhamento de pacotes por meio de um módulo de segurança feito em *hardware*. Assim, é necessário que todos participantes da rede

possuam um módulo de segurança, o que diminui a flexibilidade da rede, e aumenta o custo devido à aquisição do módulo. Além disso, todos os pacotes enviados devem passar pelo módulo de segurança e assim aumenta o consumo de energia e tempo, pois aumenta a carga de processamento com a criptografia, e sobrecarga de comunicação uma vez que esse mecanismo exige a criação de uma associação segura com os vizinhos.

O Sistema Sprite (*Simple Cheat-proof Credit-based System*)

Algumas propostas utilizam a troca de créditos com o uso de um módulo de *hardware* inviolável específico para fazer o estímulo à cooperação. Apesar de ter a possibilidade da segurança desses módulos, o uso de *hardwares* invioláveis específico deve ser evitado, pois esse requisito pode dificultar a aceitação da tecnologia. Assim, Zhong *et al.* propuseram Sprite (*Simple Cheat-proof Credit-based System*), um sistema de troca de créditos que evita a trapaça sem se servir de um módulo de *hardware* inviolável [51]. O sistema, assim como outros de trocas de créditos, baseia-se no consumo de créditos para o envio de pacotes e obtenção de créditos ao encaminhar pacotes de outros. Ao encaminhar pacotes de outros nós, um nó deve receber o suficiente de créditos para enviar os próprios pacotes, para que assim ocorra o estímulo à cooperação. Outro ponto é que como o sistema não utiliza módulos de *hardware* invioláveis, ele deve ter mecanismos seguros de trocas de informações para evitar a falsificação de mensagens. Esse sistema utiliza recibos de comprovação de encaminhamento, assim, o nó armazena um recibo para cada mensagem recebida e encaminhada como prova da ação. Depois, o nó contata uma entidade que possui o serviço de compensação de créditos (*Credit Clearance Service* - CCS) e envia seus recibos de mensagens encaminhadas e recebidas para contabilizar e atualizar os créditos ganhos. O CCS calcula os gastos e ganhos de créditos dos nós envolvidos na transmissão do pacote de acordo com os recibos. O custo do envio de pacotes poderia ser cobrado tanto do remetente quanto do destinatário, entretanto, o custo é cobrado somente do remetente das mensagens, para evitar possíveis ataques de negação de serviço. e envio de mensagens inúteis. Esse ataque de negação de serviço poderia ser feito com o objetivo de consumir os créditos do destinatário, para que esse fique indisponível. Os nós que ganham créditos são todos que participaram do encaminhamento do pacote, e somente se o pacote chegou corretamente no próximo salto. O CCS verifica a consistência do encaminhamento através dos recibos trocados durante a entrega de pacotes.

Para incentivar nós que são egoístas a encaminhar mensagens, a quantidade de créditos pagos a nós que encaminham é maior em relação aos nós que não encaminham. O último nó a receber o pacote, (o destinatário, deve receber uma quantidade de créditos para incentivá-lo a prestar contas com a CCS e validar a ação de recebi-

mento de pacotes. Dessa maneira, todos nós que participaram do encaminhamento do pacote enviam os recibos para o serviço de compensação de créditos (CCS), que de posse dos recibos pode determinar o último nó do caminho e os intermediários e pedir para o remetente pagá-los de acordo. O sistema ainda desestimula os nós deixarem de enviar recibos e previne que nós troquem recibos falsos. Entretanto, essa proposta tem a grande desvantagem de utilizar uma entidade centralizada que deve ser confiável e com alta disponibilidade para contabilizar os custos e ganhos de créditos, o que é de difícil implementação em redes ad hoc.

O Sistema SCAN

Yang *et al.* propuseram SCAN, um sistema para assegurar de maneira unificada tanto o plano de controle da rede (protocolos de roteamento), como também o plano de dados (encaminhamento de pacotes) [64]. SCAN utiliza uma abordagem reativa para proteger as mensagens, na qual os nós vizinhos colaborativamente sustentam, monitoram e reagem a ataques. No SCAN, os nós devem possuir uma ficha para participar da rede. As fichas são entregues por nós vizinhos legítimos, que já participam da rede. As fichas são assinadas através de criptografia assimétrica distribuída, na qual a chave pública é amplamente conhecida na rede, como no sistema distribuído de autenticação em [31, 33]. Assim, somente quando um determinado número de nós concordarem, a ficha é assinada. Além disso, os nós podem aumentar o tempo de validade de suas fichas pelo tempo de permanência na rede. Assim, como um prêmio de bom comportamento, a cada renovação bem sucedida de ficha os nós ganham bônus de tempo para a ficha, diminuem a frequência de renovação e, consequentemente, reduzem a sobrecarga de controle.

O monitoramento das mensagens é realizado graças à natureza de difusão do envio de mensagens em ad hoc sem fio. Assim, cada nó monitora os pacotes de dados trafegados para saber se são entregues corretamente, e se anúncios de rotas são consistentes. O monitoramento é feito através de uma técnica chamada de validação cruzada, na qual os nós observam a vizinhança e validam a consistência do encaminhamento de pacotes e de anúncios de rotas. O monitoramento é realizado individualmente, mas os nós precisam agir em conjunto com a vizinhança para condenar nós suspeitos. Quando um nó detecta um comportamento malicioso, ele envia uma mensagem em difusão denunciando o nó malicioso. Quando um mesmo nó é denunciado por determinado número de nós, o nó é considerado malicioso é colocado em uma lista de revogação de fichas, perdendo assim o direito ao acesso às funções da rede.

A renovação e revogação de fichas são realizadas de maneira colaborativa com quaisquer nós que possuam fichas válidas. As fichas contêm a identificação do nó, a estampa de tempo (*timestamp*) da assinatura e da expiração da ficha, e quando

os nós querem renovar a ficha, eles enviam uma requisição de renovação de fichas para os nós vizinhos, de até dois saltos de distância, com a estampa de tempo desta requisição de renovação e a ficha atual. Ao receber uma requisição de renovação e ficha, o nó verifica se a ficha do requisitante é válida e se não foi revogada, e então assina a nova ficha parcialmente com sua parte da chave secreta. A estampa de tempo da assinatura é a mesma enviada na requisição, e a estampa de tempo de expiração é a diferença da estampa de tempo de assinatura e da expiração da ficha anterior, acrescida de um intervalo definido.

Assim, para participar da rede, os nós precisam de uma ficha, uma ficha limpa ou um certificado de boa conduta. A emissão de fichas é controlada colaborativamente, assim como sua revogação, de modo que somente se um número determinado de nós concordarem em fazer essas operação ela pode ser realizada. Nesse sistema, todos os nós monitoram seus vizinhos e julgam o comportamento deles, e quando necessário, trocam mensagens sobre mau comportamento de nós para revogar suas fichas. Desta maneira, o SCAN controla o acesso à rede ad hoc de maneira segura e distribuída.

O Sistema MARCH (*Money And Reputation sCHemes*)

Como os usuários de redes par-a-par (*Peer to Peer* - P2P) estão cada vez mais diversos, deve-se existir um mecanismo para incentivar a cooperação entre eles. Outro problema desse tipo de rede é a falta de entidades confiáveis centralizadas com alta disponibilidade. Semelhante às redes ad hoc nesses aspectos, as redes P2P também possuem ausência das entidades centralizadas e se baseiam no altruísmo dos nós, no qual os nós oferecem seus recursos e em troca podem usar os recursos de outros nós. Entretanto, com o crescimento da rede e diversificação dos usuários da rede, diversos usuários consomem os recursos sem contribuir para a comunidade. Consequentemente, o altruísmo é quebrado, o que leva o sistema colapsar. Assim, de maneira semelhante às redes ad hoc, as redes par-a-par devem possuir um sistema que estimule a cooperação dos nós.

Para resolver o problema de estímulo a cooperação em redes par-a-par, Zhang *et al.* propuseram um esquema distribuído que combina reputação e dinheiro virtual (*Money And Reputation Schemes* - MARCH) [65]. O esquema limita o dano causado por nós maliciosos e grupos maliciosos que agem em conluio. Uma das características do esquema é que o benefício dos nós é limitado pela contribuição dos nós ao esquema. Outro ponto, é que nós em conluio não conseguem aumentar o dinheiro deles e a reputação, independentemente do tamanho do grupo em conluio. Além disso, nós maliciosos que só podem atacar ao custo de seus próprios lucros. O esquema possui uma infraestrutura de autoridade distribuída para gerenciar o histórico das informações com baixa sobrecarga e alta segurança, e possui também um protocolo de compartilhamento de chaves e verificação de contratos baseado em

criptografia de limiar.

O sistema MARCH se baseia na concepção de que em transações reais, o provedor gostaria de saber se o consumidor possui dinheiro suficiente para pagar pelo serviço, e o consumidor gostaria de saber a reputação do provedor. Com essas informações, os pares podem avaliar o risco de uma transação. No caso do provedor enganar o consumidor, ele pode ser processado pelo consumidor. Se o consumidor difamar o provedor, ele somente o fará depois de pagar pelo serviço, então a difamação possui custos, e consequentemente é limitada pela quantidade de dinheiro que o consumidor malicioso possui. Assim, o esquema se baseia em dois parâmetros: reputação e dinheiro. Os provedores ganham dinheiro e reputação ao servir os consumidores que pagam pelos serviços. Se o consumidor achar que não foi atendido adequadamente, ele informa à autoridade a quantidade de dinheiro paga a mais pelo serviço. Se a autoridade puder afirmar quem está mentindo, o mentiroso é punido. Se a autoridade não puder definir o mentiroso, ela diminui a reputação do provedor, e congela o dinheiro declarado pago a mais, de maneira que nem o provedor nem o consumidor podem mais utilizá-lo. Esse processo de congelamento de dinheiro desestimula a difamação, pois se um nó quiser difamar outro terá de pagar por isso.

A implementação da autoridade distribuída é realizada por meio de grupos de nós chamados de delegações. Cada nó possui uma delegação associada a ele, de maneira que os integrantes da delegação armazenam os valores de dinheiro e reputação do nó. A delegação constitui-se de x nós escolhidos de modo distribuído e aleatório na rede. Uma maneira de se fazer isso é aplicar x funções *hash* no identificador do nó para se obter identificadores derivados. Caso o identificador derivado não pertencer a nenhum nó na rede, o nó mais próximo pode ser escolhido no seu lugar, como no esquema do Chord [66]. As informações de um nó são legitimadas pelas respectivas delegações com base em voto de maioria. Assim, a probabilidade de nós maliciosos forjarem informações é reduzida, pois além dos nós que compõem a delegação serem distribuídos na rede, os nós maliciosos devem ser a maioria da delegação.

O dinheiro e reputação dos nós no esquema MARCH, são armazenados pela autoridade distribuída. O parâmetro dinheiro subdivide-se em três outros: 1) total de dinheiro, que evidencia o dinheiro ganho menos o dinheiro pago em transações; 2) dinheiro pago a mais ao nó, que é o dinheiro declarado pelos consumidores que foi pago a mais pelo serviço; 3) e dinheiro disponível, que é o total de dinheiro menos o dinheiro pago a mais ao nó. A reputação avalia a qualidade do serviço prestado pelo nó, que é a porcentagem do dinheiro disponível em relação ao total de dinheiro, ou seja, considera o dinheiro ganho que não foi congelado devido a reclamações de consumidores. Outro ponto interessante nesse esquema é que os provedores podem tentar aumentar suas reputações ao oferecer serviços a preços mais baratos. Desse modo, os consumidores avaliam os riscos de se fazer transações de provedores de

menor reputação por um menor preço.

2.3.3 Sistemas Alternativos a Reputação e Troca de Créditos

Alguns protocolos não se baseiam diretamente de sistemas de reputação, confiança ou troca de créditos para fazer o estímulo à cooperação. A seguir, apresentam-se algumas dessas propostas.

O Protocolo *C*Ontext FrEE (COFFEE)

Diversas abordagens baseadas em sistemas de reputações e trocas de créditos foram estudadas para fazer o estímulo à cooperação em redes ad hoc. A maioria das soluções são fundamentalmente baseada em contexto, ou seja, devem monitorar o ambiente e outros nós para identificar corretamente comportamentos maliciosos e egoístas e puni-los. O monitoramento e classificação correta de comportamento são tarefas muito difíceis de serem realizadas corretamente, pois ações podem não ser detectadas, ou podem ser mal interpretadas. Dessa maneira, Song e Zhang propuseram um protocolo de roteamento para o estímulo a cooperação livre de contexto (*COntext FrEE* - COFFEE) que permite a transmissão de pacote sem a dependência das informações de outros nós [54]. Um protocolo de roteamento ser livre de contexto significa que dado um caminho e um pacote, o protocolo deve fazer com que seja possível transmitir os pacotes com sucesso, sem a necessidade de informações de transmissões de outros pacotes. Nessa abordagem livre de contexto, o protocolo não precisa trocar, armazenar e atualizar informações sobre o contexto de segurança, não precisa rastrear os pacotes, e não precisa verificar o tráfego de outros nós. Uma consequência de o protocolo ser livre do contexto é que por não armazenar o histórico de transmissões de pacotes, os nós possuem informações limitadas para a para decidir o encaminhamento de pacotes. Então, o protocolo tem como premissas que todos os nós gostariam de receber pacotes e poder enviá-los. Além disso, o comportamento egoísta dos nós ainda representa que são racionais no momento de encaminhar os pacotes, e que o motivo de descartar os pacotes é somente economia dos próprios recursos. Dessa maneira, para evitar o comportamento egoísta, os nós não sabem claramente o destinatário dos pacotes, pois como eles mesmos podem ser os destinatários dos pacotes, não podem assim descartá-los.

O protocolo COFFEE possui três propriedades principais. A primeira das propriedades é que durante o encaminhamento dos pacotes, a identidade do destinatário deve ficar escondida de todos nós, tanto dos nós intermediários quanto do destinatário. Assim, um nó ao enviar um pacote remove todas as informações sobre o caminho do pacote e o destinatário e criptografa os dados do pacote. Outra propriedade prin-

cipal do protocolo é que o destinatário também participa do encaminhamento dos pacotes. Uma forma de se obter isso é fazer com que o caminho do pacote envolva o destinatário como nó intermediário. Para tal, o remetente do pacote acha um vizinho do destinatário e faz com que o caminho do pacote passe pelo destinatário até seu vizinho, e depois volte para o destinatário. Por fim, a terceira propriedade principal é a identidade do destinatário só é revelada depois que todos os nós encaminharem o pacote cooperativamente. Assim, o remetente do pacote criptografa o conteúdo e o destinatário do pacote com as chaves públicas dos nós que participam do encaminhamento na ordem reversa. Com essas propriedades, qualquer pacote que um nó receba pode ser destinado para ele mesmo, então o nó deve sempre encaminhar para evitar a perda de pacotes. Desse modo, com essas propriedades o protocolo COFFEE evita o monitoramento de ações, classificação de comportamento e armazenamento de informações de comportamentos e históricos de ações sobre outros nós da rede.

O protocolo COFFEE evita o comportamento egoísta com uma proposta livre de contexto. Apesar de não tratar diversos aspectos de cooperação e segurança e redes ad hoc como o próprio comportamento malicioso, o protocolo COFFEE é uma abordagem alternativa que realiza o estímulo à cooperação.

O Protocolo *Continuously Adapting Secure Topology-Oblivious Routing* (CASTOR)

Normalmente, a forma de fazer a comunicação de redes ad hoc com segurança é através de protocolos de comunicação segura em cima de protocolos de descoberta de rota seguros. Nessa abordagem, o gerenciamento é complexo quando as redes possuem grande dinamicidade e quando devem evitar adversários poderosos. Desse modo, atender a esses requisitos e preservar a comunicação segura é um desafio, e este desafio é ainda maior quando as redes aumentam de tamanho. Assim, Galuba *et al.* criaram CASTOR, um protocolo de roteamento seguro escalável (*Continuously Adapting Secure Topology-Oblivious Routing* - CASTOR) [67]. No protocolo CASTOR, cada nó acompanha a confiabilidade de somente seus vizinhos, então o estado local não é difundido na rede toda. Além disso, os pacotes de rotas não carregam a confiabilidade dos nós do caminho, então não crescem com a rede. Cada nó da rede opera autonomamente baseando suas decisões de roteamento independente de outros nós e somente com o conhecimento de sua vizinhança. Fundamentalmente, CASTOR baseia-se em um modelo de comunicação, no qual todos pacotes trafegados seguem normalmente até o destino, que sempre reconhecerá o sucesso da recepção com um pacote de resposta pelo caminho reverso até a fonte. Assim, cada nó analisa os fluxos que passam por ele, os pacotes e suas respectivas respostas. Esse procedimento deve ser protegido contra falsificações de pacotes de respostas, o que

pode ser assegurado por meio de esquemas criptográficos, como chaves assimétricas certificadas.

Os nós armazenam métricas de confiabilidade para os fluxos que os atravessam por vizinho. Para isso, um nó mantém uma relação dos fluxos que trafegam por ele e estatísticas das mensagens que foram respondidas ou não. As métricas de confiabilidade são atualizadas constantemente de acordo com os pacotes de resposta trafegados e também pela expiração de temporizadores correspondentes ao tempo de espera dos pacotes de resposta. Assim, os nós geram um valor de confiabilidade para os vizinhos, e utilizam este valor como métrica para a escolha do próximo nó, sendo o próximo salto aquele de maior valor de confiabilidade. No caso de falta de dados ou inexistência de nós confiáveis, o pacote é enviado em difusão pra tentar achar uma rota alternativa para o destino. CASTOR mede a confiabilidade de seus vizinhos através de um estimador de confiabilidade, cujo valor é definido pela média aritmética de dois estimadores: um de transmissões bem-sucedidas e outro de mal-sucedidas. Além do estimador de confiabilidade, CASTOR utiliza uma segunda métrica para escolha de rotas que dá preferência pelo vizinho que enviou a resposta antes. A escolha do próximo salto com essa propriedade evidencia a rota com menor tempo de resposta, e assim melhora o desempenho da rede. Se o vizinho com menor tempo de resposta for considerado não confiável o nó escolhe outro próximo salto que seja confiável.

Como o protocolo baseia-se na confiabilidade para a escolha das rotas, ele detecta e reage a diversas causas de perda de pacotes, independentemente da natureza da perda, e seja maligna ou não. Como o protocolo age em métricas locais de confiabilidade, ele evita localmente nós não confiáveis o que resulta em uma convergência global para rotas confiáveis e adaptabilidade à mobilidade. Além disso, como o protocolo só armazena informações locais, ações locais como falhas não causam inundações na rede inteira. Visto que as decisões de roteamento são tomadas com base nas informações de confiabilidade que os nós possuem de seus vizinhos, os nós são indiferentes ao que acontece além de sua vizinhança, o que torna o protocolo escalável.

Assim, o protocolo CASTOR, apesar de ter um funcionamento simples, baseado em mensagens de dados e de reconhecimento, é um protocolo de roteamento seguro e escalável, pois armazena somente informações sobre seus vizinhos e não cresce com o tamanho da rede. Entretanto, algumas questões não foram resolvidas, como a interação com um protocolo de comunicação seguro, a prevenção e punição contra nós egoístas e maliciosos.

2.4 Vantagens e Desvantagens dos Mecanismos Seguros e de Estímulo à Cooperação

A rede ad hoc móvel (MANET) possui características bem peculiares que trazem diversos desafios em relação à segurança e justamente por causa da ubiquidade e a falta de controle de acesso da rede. Além disso, a rede baseia-se na cooperação dos nós, sem a qual o funcionamento da rede fica comprometido. Dessa maneira é de suma importância um mecanismo que estimulo os nós a cooperarem para que a rede funcione corretamente.

Diversos mecanismos de estímulo a cooperação foram propostos, os quais se categorizam em mecanismos baseados em mecanismo de troca de crédito e sistemas de reputação/confiança. Os mecanismos baseados em trocas de créditos são interessantes por apresentar uma solução que força os nós cooperarem sem a necessidade de monitoramento. Entretanto, os sistemas de trocas de créditos requerem um serviço de "*internet banking*" que normalmente são muito complexos. Além disso, esses mecanismos focam em comportamentos egoístas e, portanto, não possuem meios de punir nós maliciosos. Da mesma maneira, os mecanismos alternativos também dispensam o monitoramento, mas não incluem meios de evitar os nós maliciosos. Assim, os mecanismos baseados em sistemas de reputação/confiança monitoram o ambiente e inferem valores de que representam o comportamento do nós, assim é possível tomar decisões para evitar os nós mal comportados, sejam eles egoístas ou maliciosos.

A construção de um mecanismo seguro para MANETs, depende de diversos atributos que em geral são alcançados com o uso de mecanismos criptográficos. As soluções normalmente se baseiam em uma entidade confiável para a geração de certificados que comprovem o material criptográfico dos usuários. No entanto, essa abordagem não é aplicável em redes ad hoc sem-fio, pois não há garantia de disponibilidade e requer o uso de entidades especializadas. Uma possível solução é o uso de criptografia de limiar para distribuir essas funcionalidades. Contudo, a utilização da criptografia de limiar implica em maior gasto de recursos para realizar as operações criptográficas. Além disso, na criptografia de limiar (k, m), basta k nós maliciosos agir em conluio para comprometer todas operações criptográficas na rede.

O sistema ACACIA por sua vez, utiliza uma abordagem na qual todos os nós organizam-se em grupos de nós controladores distribuídos na rede. Assim, existe um grupo de controladores específico para cada nó, o que dificulta a ação de nós maliciosos em conluio. Além disso, as operações criptográficas são assinaturas, menos custosas em relação às operações de geração de certificados por criptografia de limiar. O sistema ainda executa um mecanismo de exclusão de nós para forçar nós cooperarem na rede para que possam participar. O controle de acesso é reali-

zado através de dois mecanismos, o controle de entrada de nós através das cadeias de delegação e expulsão de nós mal comportados através do sistema de reputação. Contudo, o sistema de reputação e o mecanismo de exclusão implicam em uma grande sobrecarga de número de mensagens trafegadas e falta de precisão na exclusão de nós, pois não possui um mecanismo preciso de avaliação de comportamento. Dessa maneira, se faz necessário o uso de um sistema preciso e acurado de avaliação comportamental que tenha baixa sobrecarga.

Capítulo 3

A Arquitetura de Segurança para Redes Ad Hoc Proposta

Para atender às condições e ambientes das MANETs, propõe-se um mecanismo de controle de acesso que identifica e gradua comportamentos distintos e faça a punição de nós de maneira acurada em relação aos graus de comportamentos. O mecanismo baseia-se no sistema ACACIA textitet al. [49] ao herdar o grupo de controle que realiza o controle de acesso. Assim, é possível fazer controle de acesso distribuído e auto-organizado. O sistema ACACIA apresenta um mecanismo de punição baseado em um sistema de reputação. Contudo, o sistema de reputação quantifica a reputação baseado na taxa de recepção de uma mensagem especial de acusação de comportamentos suspeitos, o que faz com que o sistema não seja capaz de identificar comportamentos distintos e puni-los com acuradamente, e também acarreta em uma grande sobrecarga de mensagens. Assim, propõe-se a integração do sistema com um modelo de confiança que gradue os comportamentos distintos com acurácia. Para tal, utiliza-se o modelo de confiança proposto por Velloso *et al.* [59], que constrói um valor de confiança para seus vizinhos com alta acurácia e baixa sobrecarga.

Dessa maneira, o mecanismo controla a entrada de nós na rede, monitora o comportamento de nós, e caso detecte um nó que se comporta de maneira inadequada, o expulsa da rede. Para realizar o controle de acesso na rede, o sistema baseia-se em grupos de nós totalmente distribuídos na rede. Não existem nós com funções específicas, pois qualquer nó pode pertencer a este grupo de controle escolhido aleatoriamente e que é altamente dinâmico, de maneira que cada entrada ou saída de nós da rede faz com que os grupos sejam modificados para evitar que nós maliciosos consigam se juntar em um grupo de controle e realizar ataques em conluio.

O mecanismo realiza o controle de acesso através de um sistema de reputação/confiança de dois níveis baseado em um tribunal do júri. O mecanismo divide a tarefa controle em dois contextos, um contexto local que age na vizinhança do nó

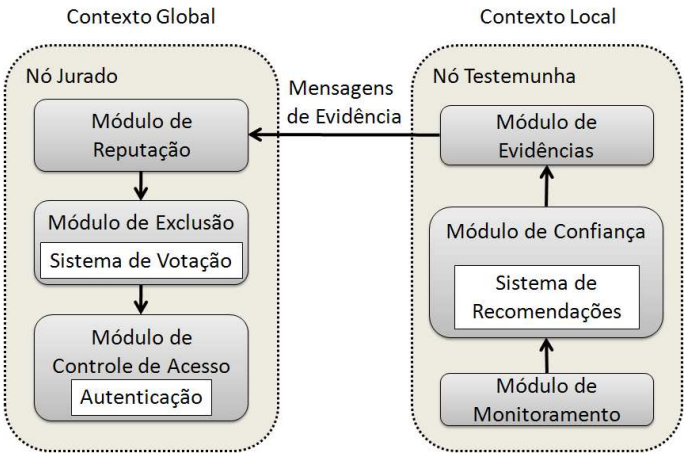


Figura 3.1: Visão geral da arquitetura do mecanismo de controle de acesso de dois níveis, que descreve a avaliação do comportamento dos nós e o mecanismo de exclusão.

ao monitorar e avaliar o comportamento de seus vizinhos, e um contexto global que verifica a permissão da participação dos nós no escopo da rede global. Assim, no nível global, o sistema cria um grupo de nós distribuídos na rede para cada nó, que controla tanto a admissão na rede quanto a punição desse nó. Esse grupo de nós que controla a presença de um nó na rede julga se o nó está apto a permanecer na rede, assim esse grupo tem o papel de júri de um nó específico. O júri realiza suas decisões de maneira distribuída, através de um esquema de votação de maioria, sem a presença de uma autoridade centralizada e também sem a presença de nós especiais que moderem, controlem, ou definam os resultados da votação. Todos os nós estão sujeitos às decisões de seus próprios júris e, portanto, todos são réus e podem ser julgados e punidos. Os integrantes do júri são escolhidos de maneira aleatória e totalmente distribuída, de maneira que todos os nós participam igualmente de júris sem a possibilidade da escolha de qual júri participar.

O júri baseia suas decisões com base em evidências de um grupo de nós que é responsável pela análise comportamental do nó réu. Assim, em um nível local os nós realizam o monitoramento da vizinhança e análise do comportamento dos nós vizinhos. Deste modo, todos os nós tornam-se testemunhas uns dos outros.

A arquitetura do mecanismo de controle de acesso está representada pela Figura 3.1, que mostra os módulos que executam nos nós para exercerem os papéis de jurados e testemunhas. O contexto local possui três módulos que são responsáveis tanto pelo monitoramento das ações realizadas pelos seus vizinhos quanto pela análise dos seus comportamentos e a notificação ao contexto global. O contexto global possui três módulos que decidem pela permanência ou pela exclusão de um nó a partir da análise comportamental informada pelo contexto local e também decidem a admissão de novos nós na rede.

A seguir descreve-se com mais detalhes os módulos e mecanismos envolvidos que integram o sistema de controle de acesso.

3.1 Contexto Local

O sistema de controle de acesso possui uma parte que age localmente para fazer o monitoramento e a análise comportamental da vizinhança. Assim, o contexto local exprime a relação entre o réu e as testemunhas. A Figura 3.1 mostra os módulos que executam nos nós para exercerem os papéis de testemunhas, que são os módulos de monitoramento, de confiança e de evidências. O módulo de monitoramento tem como funcionalidade básica monitorar as ações dos nós vizinhos e avaliar seus comportamentos. Além disso, o sistema possui um módulo de confiança que utiliza as informações do módulo de monitoramento para gerar um valor de confiança de cada vizinho. Ademais, como o monitoramento da vizinhança pode não ser perfeito, o módulo de confiança utiliza a opinião de vizinhos em comum para suprir possíveis falhas no monitoramento e acelerar a convergência do valor de confiança. Por fim, o módulo de evidências analisa os níveis de confiança de seus vizinhos para decidir se informam ao contexto global sobre vizinhos não confiáveis. Os módulos do contexto local são descritos a seguir.

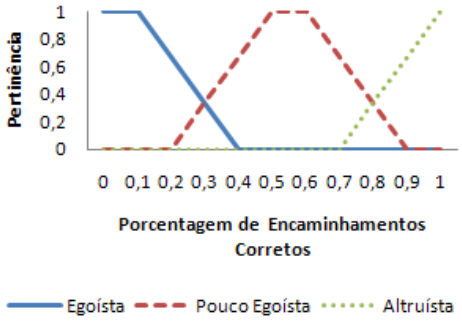
3.1.1 Módulo de Monitoramento

Todos os nós da rede possuem a função de monitorar sua vizinhança, e assim permitir somente nós que se comportem adequadamente permaneçam na rede. Desse modo, cada nó deve executar um mecanismo que inspeciona as ações dos outros nós localmente, e atribui um certo valor para o comportamento do nó. Esse valor de comportamento pode ser medido através de um valor que evidencie comportamentos distintos como um nó que ocasionalmente falhe em cooperar, ou um nó que seja egoísta e tente de qualquer maneira economizar seus recursos, ou ainda um nó que tente realizar ataques para interromper a rede. Desse modo, é necessário um mecanismo de monitoramento e classificação de comportamento. Um mecanismo que realiza essas funcionalidades é o *watchdog* proposto em [3] e utilizado também em [55]. Nesse mecanismo, o *watchdog* verifica se as ações dos outros nós foram realizadas como o esperado. Assim, todas as vezes que um nó i precisa monitorar uma ação (a) realizada por outro nó j, um de seus vizinhos, ele aciona um *watchdog* para a ação específica. Em seguida, o *watchdog* coloca a ação esperada $e_j(a)$ em um *buffer* de ações esperadas do nó j. Se a ação observada $o_j(a)$ for equivalente à ação esperada $e_j(a)$, ele remove a ação esperada $e_j(a)$ do *buffer*. Se em todo caso, a ação observada $o_j(a)$ for diferente da esperada $e_j(a)$, $e_j(a)$ permanece no *buffer*.

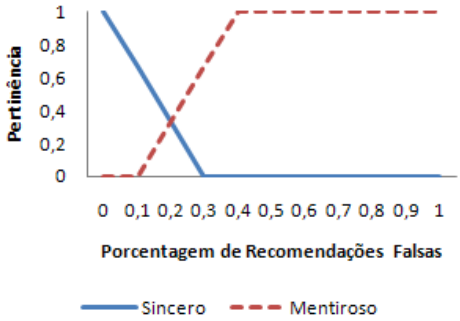
Caso a ação esperada $e_j(a)$ permaneça no *buffer* por determinado tempo, um fator de observação negativa é reportado para a entidade j, e um novo valor de confiança é calculado para o nó. Entretanto, esse mecanismo de monitoramento não consegue detectar falhas em algumas situações, como no caso que enquanto o nó monitora o próximo salto ele recebe um pacote e ocorre uma colisão, ou quando o próximo salto nó não reencaminha o pacote quando ocorre colisão no receptor, ou a potência de transmissão é suficiente para chegar ao nó que monitora, mas não no próximo nó do caminho. Se o nó percebe que o próximo salto não encaminhou os pacotes que deveria, ele considera o próximo salto está com mau comportamento.

O módulo de monitoramento pode ser adaptado para funcionar em nível de rede com o uso de um protocolo de roteamento que utilize pedidos de repetição automáticos (ARQs), como o CASTOR [67]. No protocolo CASTOR, cada mensagem é retornada com um ACK para a origem da mensagem, que indica o recebimento correto pelo destinatário. Assim, os nós podem verificar se os nós do caminho enviado encaminham os pacotes corretamente, ou se, por outro lado, não encaminham corretamente comportando-se de maneira egoísta.

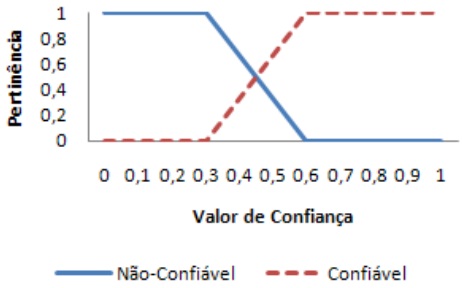
Além desses mecanismos de monitoramento, o módulo de monitoramento pode integrar outros mecanismos de monitoramento e de classificação comportamental que funcionem ao mesmo tempo. A forma com que os mecanismos se integram para gerar um valor de comportamento é um outro tópico que deve ser estudado. Um modelo genérico poderia utilizar a lógica nebulosa para agregar as informações dos diversos mecanismos de monitoramento sobre o valor de comportamento. Nesse caso, o comportamento poderia ser separado em conjuntos que definem comportamentos distintos, como por exemplo, um conjunto cujo comportamento define que o nó age cooperativamente de acordo com os protocolos, como no caso de encaminhamento de pacotes, e outro conjunto no qual o nó age maliciosamente para prejudicar o funcionamento da rede, como falsificação de recomendações sobre comportamentos de outros nós. Deste modo, classifica-se a pertinência em relação a esses conjuntos e utilizar as regras de inferência para classificar um valor final para o valor de comportamento do nó [68]. Assim, define-se um modelo com lógica nebulosa que define o valor de comportamento baseado em dois mecanismos de detecção de comportamento: a porcentagem de ações de encaminhamento que foram realizadas corretamente baseado em detectores do tipo *watchdog* para encaminhamento; e um detector de porcentagem de mensagens falsificadas, como recomendações do módulo de confiança e mensagens de evidências. Uma possível implementação em relação a recomendações é considerar falsas as recomendações de um nó X que dizem respeito a um nó Y que diferem mais de um valor limite $\pm\epsilon$ do que a própria opinião de confiabilidade no nó Y. No caso de evidências, um nó que possui um vizinho comum com outro nó poderia considerar falsas as evidências do outro nó a respeito desse



(a) Função pertinência para avaliar egoísmo.



(b) Função pertinência para avaliar comportamento mentiroso.



(c) Função pertinência para avaliar confiabilidade.

Figura 3.2: Funções de pertinência para avaliar os comportamentos dos nós.

vizinho, que informem o vizinho ser malicioso caso o nó não concorde. Desse modo o modelo com lógica nebulosa infere um valor de comportamento baseado nos valores desses detectores.

Primeiramente, aplicam-se as funções de pertinência que mapeiam os valores dos mecanismos de detecção de comportamento em valores nebulosos, os quais são usados no controlador nebuloso. Tais funções podem ser definidas pelo nó que cria a rede de acordo com os objetivos e requisitos da rede. A Figura 3.2(a) mostra uma função de pertinência possível para os valores de porcentagem de encaminhamentos corretos. Assim, de acordo com a porcentagem de encaminhamentos executados corretamente, comportamento pode ser classificado em *egoísta*, *pouco egoísta* ou *altruísta*. Já a Figura 3.2(b) mostra uma função de pertinência para definir um comportamento mentiroso de acordo com a porcentagem de recomendações falsas. A função de pertinência para o resultado, o valor de confiança pode ser representado pela Figura 3.2(c) que define se o nó é confiável ou não confiável. Em seguida, a confiança pode ser inferida de acordo com as seguintes regras SE→ENTÃO:

- SE sincero E altruísta ENTÃO confiável;

- SE não-mentiroso E pouco egoísta ENTÃO confiável;
- SE não-mentiroso E egoísta ENTÃO não-confiável;
- SE mentiroso ENTÃO não-confiável.

As saídas das regras são combinadas (ex. centróide da soma), e obtém-se o valor de confiança final proveniente do módulo de monitoramento. Assim, esse módulo de monitoramento baseado em detectores de comportamentos diversos e lógica nebulosa pode ser facilmente integrado a quaisquer outros detectores de comportamento.

3.1.2 Módulo de Confiança

Os mecanismos de monitoramento monitoram e avaliam pragmaticamente os comportamentos de outros participantes da rede. Contudo, os mecanismos de monitoramento não são acurados, seja por falhas na detecção de ações ou por má interpretação dos dados de monitoramento. Essas falhas são supridas pelo uso de modelos de confiança e reputação, que trazem como benefício a definição de como a informação do monitoramento pode ser integrada e interpretada para prever o comportamento de nós. Outra vantagem do uso de modelos de reputação e confiança é a geração de um valor quantitativo do grau de confiabilidade de um nó, que pode ser utilizado para prever o comportamento futuro dos nós. Esse grau de confiabilidade é utilizado para punir os nós que tenham graus de confiabilidade baixos, e portanto, os nós devem agir altruisticamente para que tenham um grau de confiabilidade alto e não sejam punidos. Desse modo, os modelos de reputação e confiança estimulam os nós a agirem de acordo com os protocolos e a cooperarem em prol da rede.

Dessa maneira, o módulo de confiança usa um modelo de confiança no qual os nós utilizam os dados do monitoramento local dos seus vizinhos e avaliam seus comportamentos. Nesse contexto, os nós são somente testemunhas das ações dos seus vizinhos e não cabe a eles a função de julgar ou punir. O motivo para se utilizar um modelo de confiança local é a avaliação do comportamento e da natureza dos nós. Assim, com a utilização de um modelo de confiança local, as testemunhas podem trocar informações, para acelerar e convergir na avaliação do confiança de um réu de maneira.

O modelo de confiança constrói um valor de confiabilidade para um nó vizinho a partir das avaliações de comportamento do módulo de monitoramento. Além disso, os nós com vizinhos comuns trocam suas experiências entre si para convergir conjuntamente para um valor de confiança. A seguir descreve-se o modelo de confiança.

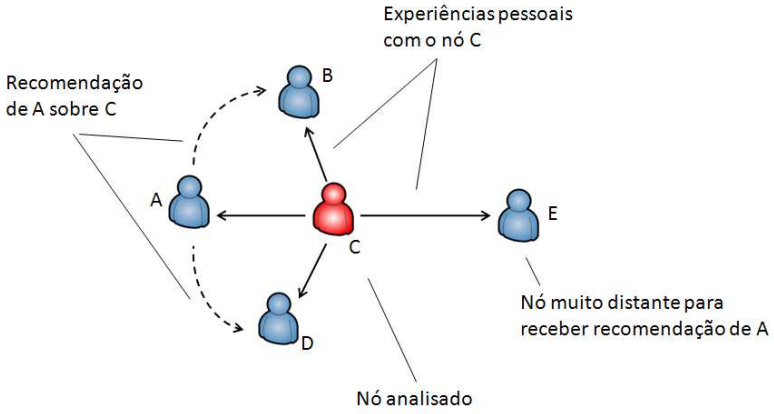


Figura 3.3: As recomendações são trocadas somente pelos nós que tem a mesma vizinhança. Assim, apesar dos nós A, B D e E analisarem o comportamento do nó C, uma recomendação de A só é difundida para seus vizinhos B e D.

Modelo de Confiança

O mecanismo proposto usa o modelo de confiança de Velloso *et al.* [59], que permite os nós avaliarem o valor de confiança de seus vizinhos, que representa uma previsão do comportamento futuro dos nós a partir do comportamento inferido pelo módulo de monitoramento. Além disso, os nós trocam recomendações para compensar problemas de monitoramento devido à incapacidade ou restrição de recursos. Recomendações são opiniões dos nós sobre um vizinho comum como mostrado na Figura 3.3. As recomendações são enviadas em difusão, portanto não são encaminhadas. Nesse modelo, o valor de confiança é um valor entre 0 e 1, onde 1 representa o maior valor de confiança, e 0 o menor. A confiança, $T^i_{\mathbf{w}}(\mathbf{d})$, de uma testemunha, $\mathbf{w}$, em um réu, $\mathbf{d}$, no momento i, é dada pela soma ponderada de sua própria avaliação do comportamento, $Q^i_{\mathbf{w}}(\mathbf{d})$, e as recomendações, $R^i_{\mathbf{w}}(\mathbf{d})$, dos vizinhos que também têm $\mathbf{d}$ como vizinho. O cálculo está descrito pela Equação a seguir.

$$T^i_{\mathbf{w}}(\mathbf{d}) = (1-\alpha)Q^i_{\mathbf{w}}(\mathbf{d}) + \alpha R^i_{\mathbf{w}}(\mathbf{d}). \tag{3.1}$$

onde α representa o peso das recomendações em relação à própria avaliação da testemunha em relação ao comportamento do réu. O valor de $R^i_{\mathbf{w}}(\mathbf{d})$ é calculado a partir das recomendações recebidas, e leva em conta a confiança do recomendador. A avaliação de comportamento $Q^i_{\mathbf{w}}(\mathbf{d})$ feita pela testemunha é dada pela expressão

$$Q^i_{\mathbf{w}}(\mathbf{d}) = \beta E^i_{\mathbf{w}}(\mathbf{d}) + (1-\beta)T^{i-1}_{\mathbf{w}}(\mathbf{d}). \tag{3.2}$$

onde $T^{i-1}_{\mathbf{w}}(\mathbf{d})$ representa o valor de confiança no momento anterior, $E^i_{\mathbf{w}}(\mathbf{d})$ representa a avaliação de comportamento estimado pelo módulo de monitoramento, é ponderada por β em relação ao valor de confiança estimado anteriormente.

Para o cálculo do valor agregado das recomendações, somente recomendações de vizinhos confiáveis são levadas em conta. Assim é definido um subconjunto $K_{\mathbf{w}}$ dos nós vizinhos que possuam um valor mínimo de confiança, cujas recomendações serão utilizadas no cálculo de $R^i_{\mathbf{w}}(\mathbf{d})$. O valor de $R^i_{\mathbf{w}}(\mathbf{d})$ é dado por

$$R^i_{\mathbf{w}}(\mathbf{d}) = \frac{\sum_{k \in K_{\mathbf{w}}} T^i_{\mathbf{w}}(k) M_k(\mathbf{d}) X_k(\mathbf{d})}{\sum_{j \in K_{\mathbf{w}}} T^i_{\mathbf{w}}(j) M_j(\mathbf{d})} \tag{3.3}$$

As recomendações levam em conta a confiança no vizinho k que fez a recomendação ($T_{\mathbf{w}}$), a maturidade da relação do vizinho k com $\mathbf{d}$ (M_k) e do valor de confiança do vizinho k em $\mathbf{d}$ (X_k). A recomendação de k em relação ao nó $\mathbf{d}$ é ponderada pela maturidade da relação ($M_k(\mathbf{d})$) medida em k. Essa medida depende do tempo que os dois nós se conhecem. A maturidade dá maior peso para nós que possuem relação mais antiga com o nó $\mathbf{d}$, pois espera-se que o valor de confiança deste já tenha convergido. Além disso, é definido um valor máximo para a maturidade (M_{max}), para diminuir o impacto de recomendações de nós que mentem sobre a maturidade para suas recomendações terem mais peso. O valor de confiança $X_k(\mathbf{d})$ é dado por uma variável aleatória normal cuja média é o valor de confiança do nó k em relação ao nó d, e $\sigma_k(\mathbf{d})$, o desvio padrão dos últimos valores de confiança do nó k em relação ao nó d. Dessa maneira, o valor de confiança considera a precisão das medidas anteriores e evita avaliações equivocadas e tendenciosas do valor de confiança:

$$X_k(\mathbf{d}) = N(T_k(\mathbf{d}), \sigma_k(\mathbf{d})) \tag{3.4}$$

3.1.3 Módulo de Evidências

O módulo de evidências gere a relação das testemunhas com o júri. Desse modo, as testemunhas enviam as mensagens de evidência para informar o júri sobre o mau comportamento de um réu que não age de acordo com as exigências da rede. Seja esse mau comportamento malicioso, egoísta ou simplesmente não cooperativo suficiente.

As mensagens de evidência são notificações enviadas pelas testemunhas que informam que um de seus vizinhos possui mau comportamento. Elas contêm informações básicas de quem é a testemunha e o réu do qual a testemunha reclama. Além disso, as mensagens são assinadas e possuem número de sequência para evitar ataques de repetição.

Em uma primeira abordagem, as mensagens de evidência são simples notificações de que um nó realizou uma ação prejudicial à rede detectada pelo módulo de monitoramento. Por conseguinte, uma ação prejudicial de um nó $\mathbf{d}$ faz com que cada vizinho $\mathbf{w} \in W_{\mathbf{d}}$ envie uma mensagem de evidência para o júri $J_{\mathbf{d}}$. O sistema ACACIA, explicado na Seção 2.2.2, utiliza essa abordagem, na qual os nós enviam

mensagens para um grupo de nós controladores todas as vezes que um nó perceber uma ação não cooperativa de outro nó, detectada pelo módulo de monitoramento.

O mecanismo de exclusão proposto neste trabalho se serve do módulo de monitoramento e também do módulo de confiança para construir um valor representativo do comportamento de seus vizinhos baseado em tanto sua própria experiência quanto a troca de opiniões com outros nós. De fato, as ações dos vizinhos podem ser mal interpretadas ou não percebidas. Assim, busca-se reunir um conjunto de evidências mais consistente e avaliar os nós de acordo com um modelo de confiança, para que a informação enviada ao júri seja mais precisa e, consequentemente, evitar punições indevidas e evitar também a impunidade. O sistema ACACIA não utiliza um módulo de confiança e, portanto não constrói uma medida representativa de comportamento. Com a proposta de se usar um módulo de confiança acurado baseado no modelo de confiança proposto por Velloso *et al.*, explicado na Seção 2.3.1, o módulo de evidências somente envia mensagens de evidência ao júri em *unicast* quando o valor de confiança é menor que limiar mínimo de confiança para permanecer na rede. Por essa razão, o módulo de evidências evita o envio desnecessário de mensagens de evidência antes de reunir um conjunto de evidências consistente quanto à confiabilidade do nó e espera-se com isto uma diminuição da sobrecarga de mensagens de controle. É importante ressaltar que com um modelo de confiança mais acurado, o limiar mínimo de confiança deve ser configurado de acordo com os propósitos e objetivos da rede, para que tolere ou não determinados perfis de comportamentos.

3.2 Contexto Global

O mecanismo de exclusão, além de possuir uma parte que age localmente para fazer o monitoramento e a análise do comportamento dos vizinhos, possui uma parte que age globalmente. Assim, cada nó da rede possui um grupo associado a ele que controla seu acesso à rede. Este grupo de nós é denominado de júri. Nesse modelo, o júri cumpre esse papel de analisar as informações adquiridas localmente para agir em um escopo global. O júri realiza a emissão de certificados para permitir o acesso do nó e também revoga seu acesso através do mecanismo de exclusão. Esse grupo é dinâmico e auto-organizado, assim, com a entrada e saída de nós, o grupo se reconfigura automaticamente. Além disso, as decisões do júri são realizadas através de votações de maioria, para distribuir as decisões e aumentar a disponibilidade. Esse grupo é configurado de maneira aleatória na rede, para evitar que adversários formem conluios e prejudiquem os serviços da rede.

Os Nós com Função de Júri

O júri é o grupo de nós que garante a validade global na rede das informações

sobre um determinado nó, denominado réu. O júri é calculado da mesma maneira que o grupo de nós controladores do sistema ACACIA, e assim o júri deve ser reconhecível e acessível por todos na rede para conferir as informações. Portanto, o júri é obtido a partir da lista de nós da rede disseminada por algum mecanismo ou protocolo. Uma forma de se obter a lista de nós que participam da rede é através de protocolos de roteamento que disseminam globalmente a lista de nós na rede. Exemplos de protocolos de roteamento que realizam essa funcionalidade são os protocolos de roteamento pró-ativos para redes ad hoc como o SOLSR [18, 19] ou FAP (*Filter-based Addressing Protocol*) [69]. Assim, nesse protocolo os nós possuem os endereços IPs uns dos outros, que podem ser utilizados como os identificadores dos nós.

A escolha do júri de um nó é realizada por um algoritmo totalmente aleatório baseado em funções *hash*. Primeiramente, ordena-se os identificadores dos nós em uma lista com uma ordenação padrão L_o. A ordenação padronizada permite a todos os nós que possuem a mesma lista de identificadores obterem o mesmo resultado do algoritmo. Além disso, as funções de *hash* utilizadas devem ser as mesmas utilizadas por todos os nós. Após a ordenação, escolhe-se uma *chave* para ser usada como semente do algoritmo a qual deve ser única para cada nó e deve ser identificável por qualquer outro que queira obter o júri de um nó. Dessa maneira, uma possível escolha é o próprio identificador do nó, ou seja, o endereço IP. Em seguida, aplica-se a função *hash* para obter o índice I_1 do primeiro jurado na lista de tal forma que

$$I_1 = mod_N\left(hash\left(chave\right)\right) \tag{3.5}$$

onde N é o número de nós na rede, e a expressão $mod_N(\cdot)$ garante que o índice está associado a um nó real. Assim, o primeiro nó jurado é $L_o[I_1]$. O índice do jurado seguinte é calculado reaplicando-se a função *hash* no índice do primeiro jurado I_1. Para cada nó são atribuídos m jurados, onde o valor de m é escolhido na criação da rede. O valor de m é um compromisso entre a segurança e disponibilidade e a sobrecarga de mensagens de controle que são necessárias para a função de júri. Para a obtenção dos m jurados diferentes o procedimento é repetido

$$I_2 = mod_N\left(hash\left(I_1\right)\right), \tag{3.6}$$

$$I_i = mod_N\left(hash\left(I_{i-1}\right)\right) = mod_N\left(hash^i\left(chave\right)\right). \tag{3.7}$$

Como o parâmetro *chave* é diferente para cada nó, os jurados são diferentes para cada réu, e não possuem controle de quem participa. Além disso, como os nós têm acesso a tanto a lista de nós, quanto ao identificador de um réu, os nós podem sempre calcular o júri de qualquer réu.

As decisões do júri são realizadas através da maioria de votos. Desse modo, mesmo que adversários formem conluio para atacar essa estrutura, existe pouca probabilidade de ao menos $k = \left\lfloor \frac{m}{2} \right\rfloor + 1$ adversários integrarem o júri de um mesmo réu [49].

Gerenciamento de Identidades

A identidade é o conceito abstrato que identifica unicamente uma entidade. Dessa maneira, a entidade nó deve possuir uma identidade que deve ser associada unicamente a ela [70]. Para remeter à identidade, utiliza-se um identificador, que é o padrão binário utilizado para identificar a identidade. Deste modo, os nós da rede só encaminham e enviam mensagens para os nós corretamente identificados e que sejam colaborativos, assim só permitem que nós cooperativos participem da rede. Assim, um nó poderá identificar-se e ser identificado por qualquer nó da rede, através de uma identidade única. Entretanto, a identidade única em uma rede ad hoc é um desafio, pois a rede é dinâmica e não existem entidades que regulem a distribuição de identificadores, assim não são raras as ocorrências de novas conexões, desconexões, formações e junções de partições. Outro ponto é que as redes ad hoc não terem um servidor central que realiza a distribuição e certificação de identificadores e localizadores como nas redes estruturadas, pois nem sempre o servidor pode estar disponível. Então, tanto a gerência quanto a difusão da identidade são desafios para sistema de gerência de redes ad hoc.

Por simplicidade, utiliza-se o endereço IP como identificador do nó. A utilização do endereço IP como identificador causa problemas implícitos. O endereço IP nesse caso possui dois papéis, além de ser um endereço topológico, é um identificador. Nesse caso, uma mudança topológica como uma mudança de rede, implica na mudança de identificador e a perda de todas as conexões. Soluções para esses problemas são o protocolo de identidade de estações (*Host Identity Protocol* - HIP) [15] ou protocolo de separação de localizador e identificador (*Locator Id Separation Protocol* - LISP) [71]. Utiliza-se a criptografia RSA (*Rivest-Shamir-Adleman Cryptography*) para realizar a comunicação segura. Como a rede é altamente dinâmica, a configuração manual de endereços IP torna-se inviável, pois há chances de escolha repetida. Assim, é indispensável a utilização de um protocolo de autoconfiguração de endereços IP. No caso de redes infraestruturadas, normalmente utiliza-se os protocolos *Dynamic Host Configuration Protocol* (DHCP) com *Address Resolution Protocol* (ARP) [72–74] para realizar a distribuição de endereços IP para as estações. Contudo, esses protocolos não se adéquam às condições de redes ad hoc, pois necessitam de um servidor com acesso direto a todas as estações e não consideram as frequentes desconexões. O gerenciamento de identidades nas redes ad hoc deve ser automático, dinâmico e distribuído, para que eventos na rede como entrada,

saída e partições na rede, tenham o mínimo de impacto na gerência de identidades. Além disso, o gerenciamento de identidades deve consumir poucos recursos, pois os dispositivos das redes ad hoc normalmente possuem recursos limitados. Uma solução que atende a esses requisitos é o protocolo de endereçamento baseado em filtros (*Filter Addressing Protocol* - FAP) [69], o qual garante a consistência de endereços frente a entrada e saída de nós, e formação e junção de partições. Então, esse protocolo pode ser utilizado no mecanismo proposto para realizar a distribuição de endereços.

A seguir descrevem-se os módulos executados pelo júri: o módulo de reputação, exclusão e controle de acesso.

3.2.1 Módulo de Reputação

O módulo de reputação serve como elemento final da conexão do contexto global com o contexto local. Dessa forma, o módulo de reputação recebe as mensagens de evidência enviadas pelo módulo de evidências do contexto local, e a partir dessas gera um valor de reputação. Assim, cada jurado possui um valor de reputação para seu réu, e ele atualiza o valor de reputação do réu através das mensagens de evidências. Quando a reputação do réu atingir um valor mais baixo que um limiar, o jurado considera que este já não pode mais participar da rede e notifica o módulo de exclusão.

O módulo de reputação tem seu funcionamento gerido pelas mensagens de evidência. Assim, o jurado mantém a reputação de seu réu, e quando receber uma mensagem de evidência, ele reduz o valor da reputação, pois, naturalmente a evidência é uma notificação de mau comportamento. Contudo, caso o jurado não receber nenhuma mensagem de evidência por certo tempo, o valor da reputação é incrementado automaticamente em intervalos regulares, pois a ausência de mensagens de evidência significa justamente um bom comportamento.

A definição do funcionamento do sistema se segue: um nó jurado de $\mathbf{d}$ ($\mathbf{j} \in J_{\mathbf{d}}$), mantém a reputação de $\mathbf{d}$ no momento i, denotada por $R^i_{\mathbf{d}|\mathbf{j}}$. Se o jurado receber uma evidência da testemunha $\mathbf{w}$ ele atualiza o valor da reputação para

$$R^i_{\mathbf{d}|\mathbf{j}} = \max\left(R^{i-1}_{\mathbf{d}|\mathbf{j}} - u, 0\right) \tag{3.8}$$

onde u é a unidade de decremento/incremento da reputação. Para evitar a sobrecarga de mensagens de evidência, o jurado só aceita uma mensagem de evidência de uma testemunha dentro de um período T_E. Esse valor pode ser definido baseado no intervalo médio entre ações de um nó para que se consiga capturar a evolução do comportamento dos nós. Além disso, o valor de reputação cresce periodicamente para reduzir o impacto de evidências falsas. Após um período T_R sem evidências a

reputação é atualizada para

$$R^{i}_{\mathbf{d}|\mathbf{j}} = \min\left(R^{i-1}_{\mathbf{d}|\mathbf{j}} + u, R_{max}\right) \tag{3.9}$$

onde R_{max} é a reputação máxima permitida. Caso a reputação caia e atinja o limiar $T_{\mathbf{d}|\mathbf{j}}$, o jurado $\mathbf{j}$ notifica o módulo de exclusão do nó $\mathbf{d}$. O voto é difundido na rede, e quando o réu possuir $\mathbf{k}_J$ votos, ele é excluído da rede e os nós da rede não consideram mais suas mensagens.

3.2.2 Módulo de Exclusão

O módulo de exclusão é responsável por realizar o procedimento de exclusão de nós uma vez que o módulo de reputação notifica que a reputação do réu é insuficiente para a permanência na rede. Dessa maneira, o módulo lida com o sistema de votação para excluir o nó.

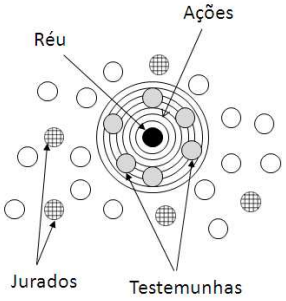
Ao receber a notificação do módulo de reputação, o módulo de exclusão envia um voto de exclusão do réu. O voto é inundado na rede para que todos os nós fiquem cientes do mesmo. Os votos são assinados e têm número de sequência para evitar ataques de repetição. Além disso, o voto é reenviado após certo tempo para reiterar sua opinião caso o nó não tenha sido excluído ainda.

Quando um nó receber votos válidos de mais da metade do número de jurados, ele marca o nó como condenado e informa o módulo de controle de acesso. O procedimento completo de exclusão é mostrado na Figura 3.4. Primeiro, na Figura 3.4(a), as testemunhas percebem ações do nó réu, e classificam seu comportamento como prejudicial. Em seguida, as testemunhas enviam mensagens de evidência para o júri periodicamente enquanto o comportamento do réu continuar classificado como prejudicial, como mostrado na Figura 3.4(b). Finalmente, ao receberem as mensagens de evidência, os jurados realizam uma votação para julgar se o nó réu deve ser expulso da rede.

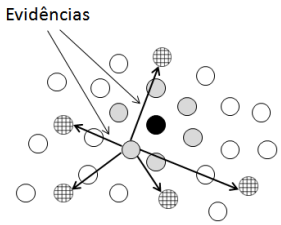
3.2.3 Módulo de Controle de Acesso

O módulo de controle de acesso tem duas funcionalidades básicas: permitir o acesso de novos nós e negar o acesso de nós mal comportados. Quando o módulo de controle de acesso é informado pelo módulo de exclusão que um réu foi condenado, ele deve retirá-lo da lista de participantes da rede. Assim, todas as rotas que o utilizam são excluídas, ele não é mais escolhido como jurado, e é adicionado em uma lista de nós condenados, caso ele tente se reconectar na rede no futuro.[1]

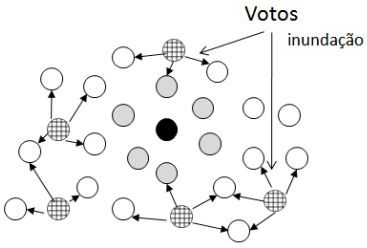
[1] A permanência de um nó na lista de nós condenados pode ser temporária, para permitir o acesso em um prazo maior.



(a) O nó réu realiza ações que são percebidas pelo módulo de monitoramento de seus vizinhos, suas testemunhas. A partir da avaliação de comportamento do módulo de monitoramento, as testemunhas inferem um valor de confiança no nó réu através do módulo de confiança.



(b) As testemunhas, ao julgarem que o nó réu não é confiável, seja por sua natureza ser maliciosa, egoísta, ou por estar com limitações de recursos, enviam mensagens de evidência ao júri.



(c) Um jurado, baseado nas mensagens de evidência recebidas julga se o réu deve ser excluído da rede. Caso decida pela exclusão, ele inunda a rede com seu voto. Se mais da metade dos jurados votarem pela exclusão, o réu é excluído da rede, e passa a ser ignorado por todos os outros nós.

Figura 3.4: O processo de exclusão de nós.

A outra funcionalidade do módulo de controle de acesso é permitir o acesso de um nó que deseje se conectar na rede, através de certificados do júri que permita o acesso. O júri verifica se o nó pode participar da rede de acordo com a lista de nós condenados e realiza um procedimento de autenticação para decidir se concede ou não o certificado. O procedimento de emissão de certificados é descrito a seguir.

Emissão de Certificados

Ao entrar na rede, o nó deve possuir um certificado emitido pelo seu júri que o permita atuar na rede. O certificado é a comprovação de que o acesso do usuário foi permitido. O certificado é apresentado quando outro nó requisita, em geral periodicamente e em novas conexões com nós. Dessa maneira, ao obter a lista de identificadores de nós, calcula seu próprio júri e envia um pedido de emissão de certificado. Os jurados ao receberem esse pedido autenticam o nó entrante e geram

certificados parciais, que juntos formam um certificado completo.

Todos os nós da rede enviam mensagens periódicas de HELLO que identificam a partição da rede para que os nós possam identificar a presença da uma outra partição. Assim, quando o nó detecta a presença de uma rede, ele pode iniciar um procedimento de entrada na rede e obtenção de certificado. A identificação da partição deve ser única, de maneira que partições tenham identificadores diferentes e possam ser diferenciadas quando se encontrar. Uma proposta é realizar a identificação com base na lista de nós da rede de maneira simplificada através de filtros de Bloom [75], ou filtros simplificados como apresentado em [69]. Nessa proposta, além do identificador da partição ser coerente e único, ele também pode ser utilizado para descobrir se um identificador de nó já é utilizado por outro, justamente pelo fato do identificador da partição representa a lista de nós daquela partição.

Dessa maneira, quando o nó deseja entrar na rede, após obter uma mensagem de HELLO de um nó, ele faz uma requisição da lista de nós para o nó que enviou a mensagem de HELLO. Nesse momento, o novo nó ainda não possui um endereço IP, então a mensagem com a lista de nós participantes da rede deve ser enviada em difusão para o novo nó. Com a posse da lista de nós, o nó pode escolher seu identificador único na rede, que no caso é o próprio endereço IP. O nó deve também escolher suas chaves criptográficas e associá-las ao seu identificador. Isso pode ser realizado diretamente através do uso de criptografia baseada em identidades (ID-PKC), de modo que o próprio identificador é a chave pública. Outra abordagem é com o uso de criptografia com certificados auto-gerados (SGC-PKC), no qual os próprios nós certificam uma chave pública para ser utilizada com certo identificador. Essas propostas requerem o uso de entidades específicas para a geração de chaves privadas e chaves parciais. Essas entidades podem estar fora da rede, de modo que a obtenção dessas chaves privadas e chaves parciais é realizada antes do acesso à rede, ou podem ser distribuídas na rede como [37, 47]. Outra proposta desenvolvida em [49], é utilizar fazer com que a chave pública escolhida possua os primeiro p bits do sufixo do identificador, como no caso o endereço de estação do seu endereço IP.

Para conseguir um endereço IP o nó passa por um processo de autenticação que deve averiguar a identidade do nó. O processo de autenticação deve dificultar a entrada de nós maliciosos e deve evitar que adversários façam diversos pedidos de autenticação para realizar um ataque de negação de serviço aos nós que autenticam novos nós. Assim, utiliza-se um sistema de autenticação que concentra as operações e processamento nos nós que querem se autenticar como em [70], o que faz com que nós que estiverem dispostos a gastar seus recursos podem se autenticar e participar da rede. O procedimento de emissão de certificados está representado na Figura 3.5.

A primeira mensagem serve como um gatilho para o processo. Quando o jurado recebe o pedido, ele verifica se está na lista de nós já condenados, e se o não

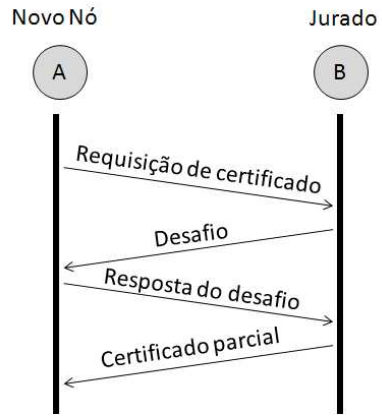


Figura 3.5: Procedimento de emissão de certificado para um nó que entra na rede.

foi excluído anteriormente o jurado dá continuidade ao procedimento. A segunda mensagem contém um numero de sequência, e um desafio para que o novo nó se mostra interessado em resolver e gaste seus recursos para isso. O desafio deve estar identificado por um contador de gerações, para evitar ataques de repetições e recusar resolução de desafios antigos. O desafio pode ser o mesmo usado em [70], no qual apresenta-se um número aleatório X e um número inteiro D que representa a dificuldade do desafio. Na terceira mensagem, o novo nó envia a solução S do desafio, a qual é obtida quando os D primeiros bits são zerados ao aplicar-se o *hash* da concatenação dos identificadores, X e S. O novo nó então deve testar diversos S até obter a solução. Por fim, quando o jurado receber a resposta e comprovar a solução do desafio (ao se verificar zerados, os D primeiros bits da aplicação do *hash* aos identificadores concatenados com X e S), ele gera um certificado parcial. O certificado parcial é uma assinatura do jurado sobre o identificador do novo nó e um número de sequência.

Assim, os jurados que aceitarem o pedido do novo nó enviam um certificado parcial para o nó entrante. Quando o nó entrante acumular ao menos $k = \left\lfloor \frac{m}{2} \right\rfloor + 1$ de seus m jurados, pode criar um certificado completo, que compõe-se de k certificados parciais juntos.

Ao se conectarem pela primeira vez, os nós trocam os certificados para verificar se seus pares realmente participam da rede. Além disso, os nós refazem a troca de certificados periodicamente para garantir a validade do acesso dos pares à rede. Dessa maneira, o certificado deve estar sempre consistente com os nós que estão na rede. Assim, quando um nó sai da rede, os certificados devem permanecer válidos e, portanto os outros devem verificar as mudanças do próprio júri e requisitar quando necessário a geração dos certificados parciais para que o certificado completo esteja atualizado e possa ser verificado.

Os nós podem a qualquer momento decidir verificar o certificado. Para verificar

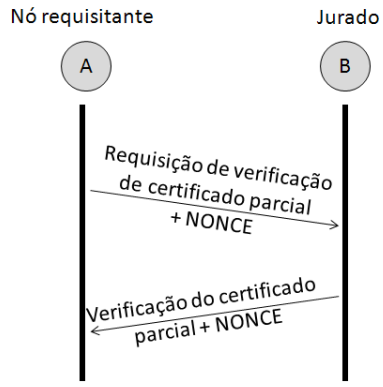


Figura 3.6: Procedimento de verificação de certificado.

um certificado, o nó desmembra o certificado nos certificados parciais e envia um pedido de verificação do certificado parcial para o jurado que o gerara. O pedido de verificação tem outro campo onde se coloca uma cadeia de caracteres arbitrária (*nonce*), cujo objetivo é evitar ataques de repetição nos quais mensagens de validação antigas são utilizadas para validar novos pedidos de verificação de certificados. Os jurados que receberem esse pedido de verificação assinam a mensagem de validação do certificado parcial, junto com o *nonce* recebido e enviam o nó que pediu pela verificação do certificado parcial. A Figura 3.6 mostra o procedimento de verificação de certificados. Quando o nó requisitanteda validação do certificado parcial receber a validação de todos os certificados parciais, ele pode ter certeza que o certificado apresentado à ele naquele momento é válido.

Apesar de um certificado ter sido validado, o nó pode requisitar a verificação periódica do certificado. Assim, após ter validado um certificado, o nó o considera válido por um tempo T_{CERT} que julgue suficiente. Esse tempo é definido pelos próprios nós de acordo com suas expectativa sobre o comportamento alheio. O procedimento de validação de certificados tem como objetivo manter a coerência de certificados, para evitar que perdas de mensagens do procedimento de punição de nós tenham grandes impactos.

Além da lista com os nós presentes na rede, existe uma outra lista com os nós condenados e que não podem mais participar da rede. Cada nó, constrói sua própria lista de revogação ao receber votos de punição de réus e quando o júri não atesta o certificado no processo de verificação de certificado.

Apesar de se usar um procedimento de autorização que dificulta certos tipos de ataques, esse procedimento não impede que nós maliciosos forjem identidades para prejudicar a rede sem serem punidos. A garantia de unicidade poderia ser utiliza através de módulos de *hardware* invioláveis como em [61]. Contudo, apesar dessa proposta garantir a unicidade da identidade devido a um controle prévio das

identidades, a diversidade dos usuários e de dispositivos torna essa proposta inviável. Outra proposta é utilizar as relações sociais para criar uma cadeia de confiança entre os nós baseado em convites para a participação na rede como no sistema ACACIA [49]. Nessa proposta os nós utilizam suas relações sociais para criar uma cadeia de delegação que realiza o controle de acesso de novos usuários. Cada nó tem um número determinado de convites que ele pode distribuir para novos membros, que se tornam seus filhos na cadeia de delegação. O nó pode enviar um de seus convites de maneira *offline* para o novo membro, e ao mesmo tempo, transfere alguns de seus próprios convites (ou nenhum) para o novo membro de acordo com a confiança que tem nele. Esse procedimento limita o número de novos membros que um usuário e seus convidados podem chamar para rede, portanto reduz a possibilidade de um usuário não confiável convidar novos membros. O convite contém a identificação e assinatura do nó que realiza o convite, e é apresentado aos nós que realizam a autenticação. Assim, pode-se verificar a consistência do convite de acordo com a cadeia de delegação e atualizar a contagem de convites dos usuários.

Além disso, a cadeia de delegação pode ser usada como uma proteção contra ataques de Sibil. Nesse tipo de ataque, o nó malicioso cria diversas identidades para dividir ou levar a culpa de suas ações maliciosas. Assim, a forma de proteção contra esse tipo de ataque é dificultar a obtenção de identidades. Neste sentido, utiliza-se um valor variável de limiar de reputação $T_{\mathbf{d}|\mathbf{j}}$ que o jurado $\mathbf{j}$ possui para o réu $\mathbf{d}$. Esse limiar é modificado todas as vezes que um descendente na cadeia de delegação for expulso da rede para

$$T_{\mathbf{d}|\mathbf{j}} = \max\left(T_{\mathbf{d}|\mathbf{j}} + \frac{F_{\mathbf{d}}}{N \cdot h} \cdot c, 1\right) \tag{3.10}$$

onde $F_{\mathbf{d}} = \min(F_{\mathbf{d}_{max}}, N)$, $F_{\mathbf{d}_{max}}$ é o número máximo de descendentes do réu d, N o número de nós na rede e h o número de saltos na cadeia de delegação entre o nó réu $\mathbf{d}$ e seu descendente expulso da rede. O parâmetro c é usado para ajustar a importância do crescimento do limiar. Dessa maneira, a cada descendente expulso da rede, o limiar de reputação mínimo aumenta. Assim, a atualização do limiar desencoraja ataques de Sibil, pois aumenta a responsabilidade ao convidar novos nós e distribuir convites para seus descendentes.

Capítulo 4

Análise do Mecanismo

Neste capítulo apresenta-se uma análise do mecanismo de exclusão de nós da rede. Na Seção 4.1 discute-se a escolha dos parâmetros do mecanismo de exclusão proposto e também do sistema ACACIA. Posteriormente na Seção 4.2, discute-se possíveis ataques ao sistema e defesas dos mesmos.

4.1 Escolha de Parâmetros

No sistema de reputação de segundo nível, a escolha dos parâmetros é importante para o funcionamento correto do mecanismo, pois o valor de reputação depende da frequência de recepção de mensagens de evidência. Assim, os parâmetros usados nas Equações 3.8 e 3.9 precisam ser customizadas para cenários específicos.

A partir da Equação 3.8, é possível notar que o valor de u determina o número mínimo[1] de mensagens de evidência necessárias para que um jurado $\mathbf{j}$ vote pela exclusão do réu $\mathbf{d}$. O número mínimo de evidências $\mathbf{E}_{min}$ é dado por

$$\mathbf{E}_{min} = \left\lceil \frac{R_{max} - T_{\mathbf{d}|\mathbf{j}}}{u} \right\rceil \tag{4.1}$$

se o valor inicial de reputação é máximo antes de receber as mensagens de evidência. Assim, o parâmetro u pode ser usado para determinar o número mínimo de mensagens de evidência necessárias para o jurado votar pela exclusão. O valor de u não deve ser muito pequeno pois fará com que seja necessário um número muito grande de mensagens de evidência para a exclusão de réus. Da mesma maneira, o valor de u não deve ser muito grande para evitar que poucas mensagens de evidência sejam capazes de gerar a exclusão de um réu, pois tais mensagens podem ser falsas (ou enviadas por falhas do módulo de monitoramento). Podemos escolher u de maneira

[1] Como a reputação reduz a cada mensagem de evidência recebida e aumente quando o jurado não recebe mensagens de evidência por determinado tempo. Assim, o número de evidências é mínimo quando a reputação só decresce, o mecanismo de incremento da Equação 3.9 não atua.

que o número mínimo de mensagens de evidências seja um número fixo ξ. Assim, se o número médio de testemunhas é $\overline{\mathbf{W_d}}$, u pode ser determinado pela expressão:

$$\frac{\mathbf{E}_{min}}{\overline{\mathbf{W_d}}} = \xi \tag{4.2}$$

$$u = \frac{R_{max} - T_{\mathbf{d|j}}}{\xi \cdot \overline{\mathbf{W_d}}}. \tag{4.3}$$

Como o jurado considera somente uma mensagem de evidência de uma testemunha em um período T_E, quanto mais testemunhas o nó possui, a taxa de redução de reputação é maior. Desse modo, a taxa máxima de envio de mensagens de evidências de uma testemunha é $\frac{1}{T_E}$, então quando o réu $\mathbf{d}$ tem $\mathbf{W_d} = |W_\mathbf{d}|$ testemunhas, a taxa máxima de redução de reputação é

$$r_{red_{MAX}}(\mathbf{d}) = \frac{\mathbf{W_d}}{T_E}. \tag{4.4}$$

O valor real de evidências que o jurado recebe é dependente do modelo de confiança do primeiro nível. No caso do sistema ACACIA, no qual as evidências são enviadas assim que as testemunhas percebem uma ação não cooperativa, a quantidade de evidências depende da frequência das ações não cooperativas. Para representar a frequência das ações boas e más define-se um conceito chamado natureza η que representa a porcentagens das ações boas executadas por um nó. Dessa maneira a taxa de ações más é $(1-\eta)\ T_A$, onde T_A é a taxa de realização de ações. Assim, se a taxa de ações más é menor que a taxa máxima de envio de evidências $\frac{1}{(1-\eta)\ T_A} < \frac{1}{T_E}$, a taxa de redução de reputação $r_{red|\text{ACACIA}}$ é

$$r_{red|\text{ACACIA}} = (1-\eta)\ \frac{\mathbf{W_d}}{T_A} < \frac{\mathbf{W_d}}{T_A}. \tag{4.5}$$

No modelo com confiança local descrito na Seção 3.1.2, as mensagens são enviadas com a taxa máxima aceita pelos jurados, ou seja, são enviadas com intervalo mínimo de T_E. Essa abordagem, portanto, não depende da natureza do nó réu, diferente do modelo de confiança simples utilizado no sistema ACACIA. A taxa de redução de reputação com esse modelo de confiança é

$$r_{red|proposta} = r_{down_{MAX}}(\mathbf{j}, \mathbf{d}) = \frac{\mathbf{W_d}}{T_E}. \tag{4.6}$$

Semelhantemente, a partir da análise da Equação 3.9, é possível notar que a reputação de um réu para um jurado cresce continuamente caso o jurado não receba mensagens de evidência. Em outras palavras, a reputação incrementa u unidades após T_R unidades de tempo desde a última atualização, seja um crescimento da reputação ou redução devido ao recebimento de evidências. Dessa maneira, pode-se

definir uma taxa com a qual a reputação cresce. Essa taxa é máxima quando o jurado não recebe mensagens de evidência, e pode ser representada por

$$r_{cres} \leq r_{cres_{MAX}} \quad | \quad r_{cres_{MAX}} = \frac{1}{T_R}. \tag{4.7}$$

onde T_R é o período com o o qual aumenta-se o valor de reputação caso o jurado não receba mensagens de evidências. A taxa de crescimento de reputação é sempre menor que o inverso do período de crescimento de reputação T_R. Isso ocorre, pois se o jurado receber uma mensagem de evidência, ele aumentará o valor de reputação T_R unidades de tempo após o recebimento da evidência, e não desde o último incremento do valor de reputação.

Assim, pode-se aproximar a dinâmica do valor de reputação do réu para certo jurado como a diferença entre taxa de crescimento da taxa de redução da reputação

$$r = r_{cres} - r_{red}. \tag{4.8}$$

Para o ACACIA,

$$r_{\text{ACACIA}} = r_{cres} - r_{red|\text{ACACIA}} \leq \frac{1}{T_R} - (1-\eta)\frac{\mathbf{W_d}}{T_A}. \tag{4.9}$$

Para a proposta,

$$r_{proposta} = r_{cres} - r_{red|proposta} = r_{cres} - r_{red_{MAX}} \leq \frac{1}{T_R} - \frac{\mathbf{W_d}}{T_E}. \tag{4.10}$$

onde r_{ACACIA} e $r_{proposta}$ são as taxas para o sistema ACACIA e com o modelo de confiança descrito na Seção 3.1.2. Uma consequência importante desses valores é que o número de testemunhas possui grande influência no cálculo de reputação de um réu. Assim, os parâmetros devem ser ajustados para o mecanismo funcionar corretamente com certo número médio de vizinhos na rede. Então, considera-se um número médio de vizinhos na rede, e que os valores de T_E e u foram escolhidos com base no intervalo médio de ações realizadas e do número mínimo de evidências necessárias para o voto de um jurado respectivamente. Dessa maneira, quando $r < 0$, pode-se determinar o tempo τ até um jurado votar pela exclusão do réu por

$$\tau = -\frac{R_{max} - T_{\mathbf{d|j}}}{r \cdot u}. \tag{4.11}$$

No caso do mecanismo de exclusão proposto, pode-se estimar o tempo mínimo até a execução do voto pela exclusão $\tau_{min|proposto}$. Suponha que o jurado recebe mensagens de evidência constantemente a uma $r = r_{min} = -\frac{\overline{\mathbf{W_d}}}{T_E}$ e não o mecanismo de crescimento de reputação não age. Considera-se o número médio $\overline{\mathbf{W}}$ de testemunhas,

então $\tau_{min|proposto}$ é calculado pela expressão

$$\tau_{min|proposta} = \frac{R_{max} - T_{\mathbf{d}|\mathbf{j}}}{\frac{\overline{\mathbf{W}}}{T_E} \cdot u}. \tag{4.12}$$

Utiliza-se a Equação 4.3 e substitui a expressão $\frac{R_{max} - T_{\mathbf{d}|\mathbf{j}}}{u \cdot \overline{\mathbf{W}}}$ por ξ:

$$\tau_{min|proposta} = \xi \cdot T_E. \tag{4.13}$$

Então T_E deve ser escolhido para gerar um $\tau_{min|proposta}$ grande o suficiente de modo que o modelo de confiança de primeiro nível tenha tempo para convergir para um valor consistente, e se evite assim a exclusão durante esse tempo devido à flutuações do valor. Pode-se também definir um valor mínimo para $r_{max|proposta}$ ao se fixar um tempo máximo τ_{max}

$$r_{max|proposta} = -\frac{\xi \cdot \overline{\mathbf{W}}}{\tau_{max}}. \tag{4.14}$$

Consequentemente, o T_R define o número mínimo de testemunhas ω necessários que fazem a reputação no júri diminuir. Isso ocorre quando as testemunhas enviam mensagens de evidência a uma taxa constante e o tempo entre as mensagens de evidência que o jurado percebe é maior que T_R, e possibilita assim a taxa de atualização da reputação ser máxima ($r = r_{max|proposta}$). Assim, T_R é definido pela expressão

$$r_{max|proposta} = \frac{1}{T_R} - \frac{\omega}{T_E} \tag{4.15}$$

$$T_R = \frac{1}{r_{max|proposta} + \frac{\omega}{T_E}} \tag{4.16}$$

onde $r_{max|proposta}$ foi definido por 4.14.

No caso do sistema ACACIA, a escolha dos parâmetros do sistema de reputação afeta a acurácia e precisão. Isso ocorre, pois a reputação é diretamente dependente da natureza, uma vez que a reputação depende da taxa de recepção de evidências, que por sua vez depende da taxa de ações más que o nó realiza. Nesse modelo, a acurácia da classificação de comportamento é realizada pelo sistema de reputação de segundo nível, que também influi na classificação de comportamento, pois o valor de reputação de um réu para um jurado depende diretamente da natureza do réu. Assim, como para um réu ser excluído o valor de r deve ser negativo ($r < 0$), ou seja, $r_{cres|\text{ACACIA}} - r_{red\text{ACACIA}} < 0$. Então, reorganizados os termos:

$$T_R > \frac{T_A}{(1-\eta)\ \mathbf{W_d}}. \tag{4.17}$$

Desta forma, ao escolher um T_R, então outros valores de natureza também causam

a exclusão enquanto $T_R > \frac{T_A}{(1-\eta)\,\mathbf{W_d}}$ for satisfeita. Por outro lado, se $T_R - \frac{T_A}{(1-\eta)\,\mathbf{W_d}}$ muito próximo de zero, o valor de $|r|$ seria muito pequeno, que causaria um tempo grande até o jurado votar pela exclusão do nó. Portanto, é possível concluir que com esse modelo não é possível obter acurácia para a exclusão.

4.2 Ataques e Proteção

Nessa seção, apresentam-se alguns ataques e a proteção oferecida pelo sistema. A Tabela 4.2 mostra a comparação entre a proteção oferecida com proposta e trabalhos relacionados.

Ataque de Sibil

Ataque de Sibil ocorre quando um nó malicioso pode criar diversas identidades para compartilhar a culpa de suas ações. Da mesma maneira, o nó pode criar uma nova identidade e apagar o seu histórico de ações. A defesa contra esses tipos de ataques reside na dificuldade de se criar identidades para a participação na rede.

Para dificultar esses ataques, o sistema utiliza o ajuste de limiar de reputação. Assim, o jurado aumenta o limiar de reputação para votar pela exclusão do réu. Isso ocorre quando um descendente do réu na cadeia de delegação é expulso da rede. O limiar de reputação é ajustado de acordo com a Equação 3.10.

Essa abordagem desestimula o ataque de Sibil, mas não evita completamente. Desse modo, para se manter na rede, o pai na cadeia de delegação deve evitar que seus descendentes sejam expulsos, assim, ele terá preferência para distribuir convites somente para filhos confiáveis. Assim, a prevenção contra esse tipo de ataque reside na maior responsabilização na distribuição de convites.

Ataque de Nós Mentirosos

Nesse tipo de ataque, um nó mente sobre o nível de confiança de outro para diminuir a confiabilidade de um nó cooperativo ou aumentar a de um nó malicioso ou não cooperativo. Esse ataque pode ser minimizado ao se diminuir a avaliação de comportamento de nós que enviam recomendações com valores de confiança muito díspares do valor que a própria testemunha possui do réu, descrito no módulo de monitoramento (Seção 3.1.1). Dessa maneira, diminui-se o valor de confiança e consequentemente o peso das recomendações dos nós mentirosos. Além disso, pode-se definir um limiar de confiança para se considerar as recomendações de outras testemunhas.

Um nó pode também tentar sabotar o mecanismo de exclusão ao enviar evidências falsas. As testemunhas desse nó podem monitorar as mensagens de evidência, e ao perceberem que a evidência foi enviada indevidamente, elas

diminuem o valor de confiança do nó que enviara a falsa mensagem de evidência. Assim, o envio de falsas mensagens de evidência resulta na exclusão do nó mentiroso.

Ataque de Conflito de Comportamento

Ataques de conflito de comportamento acontecem quando um nó comporta-se de maneira diferente para diferentes nós. O modelo de confiança utilizado constrói a confiança localmente e todas as testemunhas percebem as ações, então o impacto desse tipo de ataque é mínimo.

Ataque Bom e Mau Comportamentos Alternados

Nesse tipo de ataque o nó tem comportamento bom e mau alternadamente, e assim espera não ser detectado e identificado como maliciosos. O modelo de confiança de primeiro nível utilizado considera a opinião de outros nós, assim o valor de confiança converge rapidamente. Além disso, o passado do nó é considerado no cálculo do valor de confiança, então se o comportamento do nó mudar repentinamente, o passado ainda é considerado.

Ataque em Conluio

Nesse ataque, nós maliciosos podem se juntar para expulsar um nó da rede. O mecanismo de exclusão utiliza votação para expulsar nós, que requer pelo menos a maioria dos nós do júri concordem com a decisão de exclusão. Assim, para excluir um nó, é necessário que mais da metade do júri do nó seja maliciosa e aja em conluio. Além disso, a escolha aleatória de nós para compor o júri diminui a probabilidade de nós maliciosos que ajam em conluio integrar o mesmo júri.

Tabela 4.1: Proteção contra ataques

Ataques	ACACIA (Fernandes *et al.*) [49]	Proposta
Ataque de Sibil	sim	sim
Ataque de Mentirosos	não	sim
Ataque de conflito de comportamento	não	sim
Ataque liga-desliga	não	sim
Ataque em conluio	sim	sim

Capítulo 5

Simulações e Resultados Obtidos

Neste capítulo são apresentados os cenários das simulações realizadas e resultados obtidos. As simulações foram realizadas com o *Network Simulator 3* (NS-3) [76] e comparam a proposta que utiliza o sistema de reputação/confiança de dois níveis com o sistema ACACIA com diferentes configurações. A comparação com o sistema ACACIA permite avaliar o impacto do uso do mecanismo de exclusão proposto. As simulações avaliam a capacidade do mecanismo proposto para realizar a exclusão de nós mal comportados em diversos cenários com diferentes números de testemunhas. Ademais, as simulações avaliam também a robustez do mecanismo de exclusão de nós maliciosos quando testemunhas não são capazes de perceber todas as ações por limitações de ambiente de propagação e erros dos mecanismos de monitoramento, e também quando ocorrem erros de classificação de comportamento do módulo de monitoramento. Além disso, avalia-se a sobrecarga de mensagens do mecanismo de exclusão proposto assim como o tempo desde a entrada de um nó mal comportado até sua exclusão. A seguir o modelo é descrito e em seguida as simulações e resultados são apresentados.

5.1 Descrição da Simulação

Todas as simulações utilizam uma rede de 64 nós com topologia fixa em grade oito por oito como representado pela Figura 5.1. O modelo de canal utiliza: perda de sinal de propagação com distribuição log-normal; e velocidade constante. Assim, os nós possuem alcance máximo de $r = d\sqrt{2}$, onde d é a distância entre os nós horizontalmente e verticalmente. Desse modo, os nós alcançam os nós que se posicionam acima, abaixo, aos lados e nas diagonais de suas posições e, portanto, dependendo da posição do nó avaliado, possuem um numero diferente de vizinhos diretos. A Figura 5.2 mostra os respectivos alcances para diferentes posições possíveis da topologia em grade. A Figura 5.2(a) mostra o alcance de um vizinho posicionado em um dos vértices da grade, e as Figuras 5.2(b) e 5.2(c) mostram o alcance dos nós

que posicionam-se nas arestas e dentro da grade, respectivamente.

Para simular comportamentos distintos, as simulações utilizaram um modelo de comportamento no qual um nó pode realizar dois tipos de ações: ações boas ou más. Nesse contexto, utiliza-se o conceito de natureza para quantificar a taxa de ações boas e más que um nó faz. A natureza é então um valor entre 0 e 1 que define a frequência das ações boas. Assim, se um nó possui natureza igual a $0,60$, isso significa que 60% das ações são boas e 40% são más. Desse modo, o tipo de ação realizada é determinada por uma variável aleatória uniforme $U(0,1)$, que se for menor que a natureza é boa, senão é má. As ações boas e más são consideradas apenas pelo mecanismo de exclusão, portanto não alteram o funcionamento de nenhum outro protocolo ou mecanismo que executam nos nós. Todos nós realizam ações com intervalo exponencialmente distribuído de valor médio de 1 unidade de tempo ($\lambda = 1$).

Para integrar-se com o módulo de confiança, o módulo de monitoramento precisa enviar uma avaliação comportamental. Desse modo, o módulo de monitoramento foi modelado para estimar a natureza dos nós baseado em ações passadas, assim determina a porcentagem de ações boas realizadas como a avaliação comportamental. Assim, a cada ação realizada, o módulo de monitoramento gera um novo valor de comportamento como a porcentagem de ações boas em relação a todas as ações realizadas.

Para as simulações realizadas, o limiar de natureza mínima permitida para que um nó permaneça na rede foi definido como $0,3$, assim qualquer nó que possua natureza inferior a esse valor é considerado como uma ameaça à rede e deve ser excluído. Esse valor pode ser modificado de acordo com os requisitos da rede para garantir uma confiabilidade mais alta ou permitir uma confiabilidade mais baixa. As testemunhas enviam evidências ao júri ao detectarem que o réu possui menor valor de confiança que o limiar mínimo de natureza configurado.

Para modelar possíveis imperfeições no módulo de monitoramento como falhas

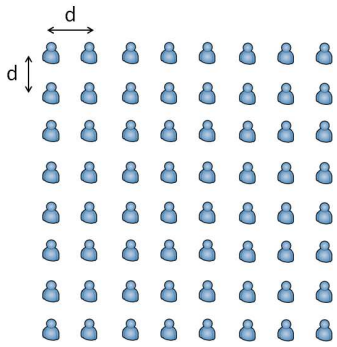
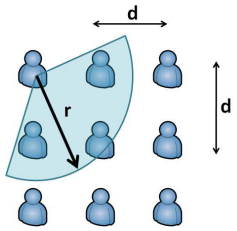
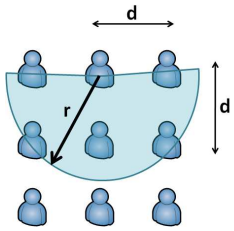


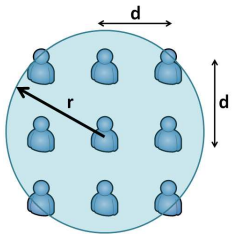
Figura 5.1: Topologia utilizada nas simulações.



(a) Três vizinhos de um nó que posiciona-se em um dos vértices da grade.



(b) Cinco vizinhos de um nó que posiciona-se nas arestas da grade.



(c) Oito vizinhos de um nó que posiciona-se dentro da grade.

Figura 5.2: Alcance máximo e número de vizinhos dos nós posicionados em grade. A distância entre os nós da grade e d e o alcance de rádio é $r = d\sqrt{2}$, de maneira que os nós também alcançan os nós posicionados nas sua diagonais além de alcançar os nós acima, abaixo e aos lados.

em detectar ações dos vizinhos e interpretações erradas quanto às ações dos vizinhos, foram definidos dois parâmetros: a percepção e a probabilidade de erro de classificação de ações. A percepção é um valor que indica a probabilidade de um nó detectar ações de outros, e a probabilidade de erro de classificação de ações indica a probabilidade de classificar uma ação boa como má e vice-versa.

O mecanismo de exclusão proposto é comparado com o mecanismo ACACIA, e como a acurácia do mecanismo ACACIA depende dos parâmetros do sistema de reputação, utilizaram-se três diferentes configurações para esse mecanismo. As configurações modificam o parâmetro T_R, que define o tempo entre incrementos de reputação caso o nó não receba mensagens de evidência. Logo, um aumento no parâmetro T_R causa em um crescimento mais lento da reputação de um réu enquanto o jurado não receber evidências daquele réu. Assim, as três configurações ACACIA_1, ACACIA_2 e ACACIA_3, consideram os valores $0,225$, $0,300$ e $0,500$ para T_R. Como na configuração ACACIA_1 a reputação do réu no júri aumenta rapidamente, espera-se que um réu só seja excluído com uma natureza mais baixa. Da mesma maneira com as configurações ACACIA_2 e ACACIA_3, espera-se que o réu seja excluído com valores de natureza mais altos.

5.2 Resultados

5.2.1 Avaliação de Desempenho

Primeiramente simulou-se a capacidade do mecanismo de exclusão proposto de identificar e excluir um nó específico da rede. Assim, escolhe-se um nó para ser analisado, atribui-se a este nó valores de natureza de 0 a 1 com passos de 0,05, e

avalia-se se o mecanismo tem sucesso em detectá-lo como um nó malicioso e excluí-lo, quando sua natureza é menor que a tolerada na rede. Como a posição do nó influencia na quantidade de vizinhos, foram analisados os três casos possíveis, o nó posicionado na grade 6x6 interna, posicionado em um dos vértices e posicionado sobre uma das arestas da grade. Todos os outros nós da rede possuem natureza igual a 1, portanto só realizam ações consideradas boas. Essas simulações tiveram 100 rodadas e a barra de erro corresponde a um intervalo de confiança de 95%. As simulações não consideraram as imperfeições do módulo de monitoramento, assim a percepção dos nós é máxima, assim como a taxa de erro de classificação de ações.

A Figura 5.3 mostra a posição do nó analisado na simulação com o nó posicionado na grade 6x6 interna, assim o nó possui oito vizinhos diretos, ao alcance de transmissão de rádio.

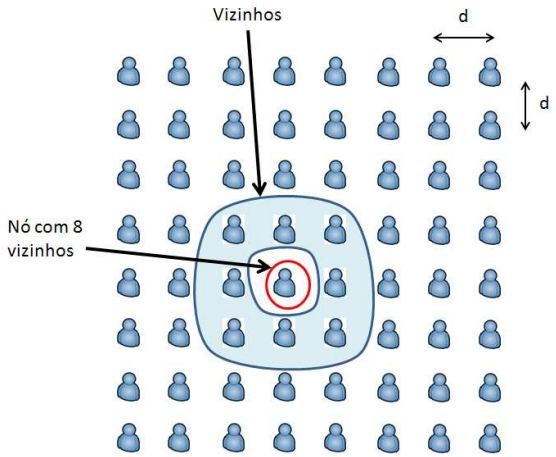
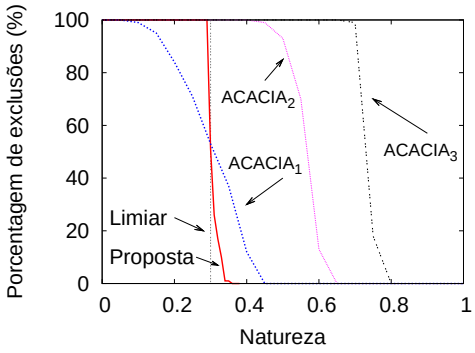
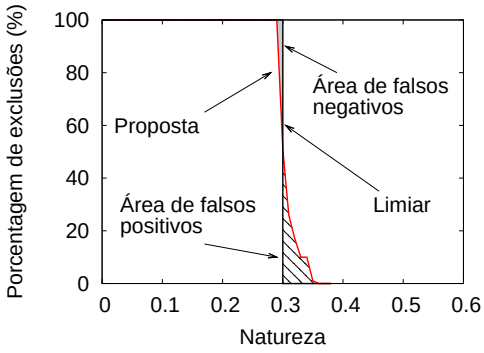


Figura 5.3: A posição do nó analisado na simulação com oito vizinhos.

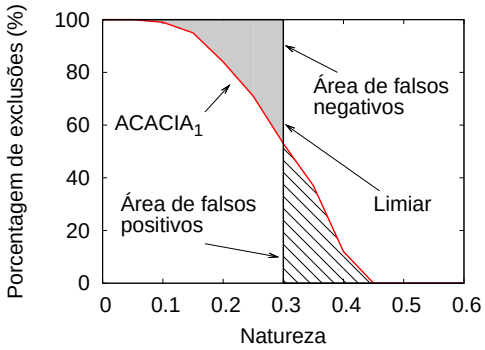
A Figura 5.4(a) mostra a porcentagem das rodadas de simulação nas quais o nó analisado com oito vizinhos foi excluído da rede, ou seja, a maioria do júri votou pela exclusão do nó analisado. Pode-se perceber a partir da Figura 5.4(a), que a utilização do mecanismo de reputação/confiança de dois níveis proposto neste trabalho possui uma baixa taxa de erro das exclusões, pois a curva de porcentagem de exclusão praticamente coincide com o valor 0,3 de limiar de natureza mínima. É importante ressaltar que toda exclusão de nós com natureza menor que este limiar é considerada como verdadeiro positivo. Consequentemente, toda exclusão de nós com natureza maior é um falso positivo e corresponde a região do gráfico para valores de natureza maior que 0,3 e menores que a curva de percentual de exclusão. Da mesma maneira, uma não exclusão de nós com natureza maior que este limiar é considerada como verdadeiro negativo e menor, região do gráfico a esquerda entre o valor 0,3 de natureza e a curva, é um falso negativo, que corresponde a região do gráfico para valores de natureza menor que 0,3 e maiores que a curva de percentual de exclusão.



(a) Porcentagem de rodadas que houve exclusão de nós.



(b) Detalhes da porcentagem de rodadas que houve exclusão de nós para o mecanismo proposto.



(c) Detalhes da porcentagem de rodadas que houve exclusão de nós para o sistema acacia na configuração ACACIA_1.

Figura 5.4: Porcentagem de exclusão de um nó com analisado com oito vizinhos para diferentes valores de natureza.

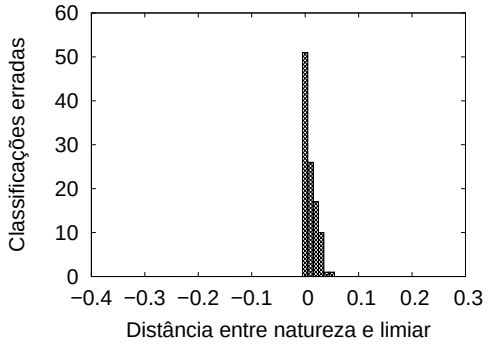
Logo, o mecanismo proposto exclui com uma baixa taxa de erro, pois possui baixa taxa de falsos positivos e falsos negativos. Isso ocorre justamente devido à utilização do módulo de confiança local, que antes de notificar o júri, avalia localmente o comportamento do réu. Quando o módulo de confiança das testemunhas avalia o réu com valor de confiança menor que o tolerado na rede, elas enviam mensagens de evidência a uma taxa fixa para o júri do réu. A pequena imprecisão do sistema proposto, pequena taxa de falsos positivos e falsos negativos, decorre do modelo de confiança utilizado que apresenta uma pequena variação do valor de confiança devido à característica interativa do cálculo de confiança (vide Equação 3.1 e [59]). Dessa maneira, a confiança do réu pode ficar um pouco maior que sua natureza, e vice-versa. Esse fato pode ser verificado na Figura 5.4(b) que mostra com mais detalhes a exclusão do nó analisado e em cinza os falso positivos e em hachurado os falsos negativos. Para a garantia da precisão das medidas do mecanismo proposto, utilizou-se o passo de 0,01 de natureza entre 0,25 e 0,40 de natureza. Nos demais intervalos e assim como o ACACIA com diferentes configurações o passo foi de 0,05

de natureza.

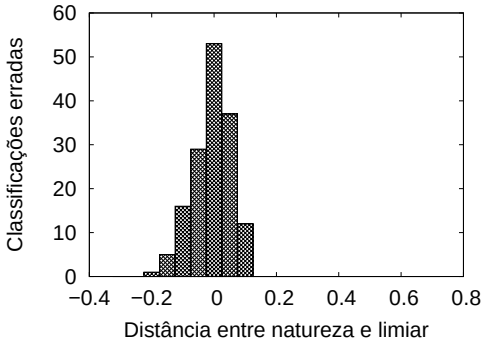
Por outro lado, o sistema ACACIA apresenta uma taxa de erros maior, possui maior taxa de falsos positivos e negativos. O ACACIA envia evidências todas as vezes que o módulo de monitoramento detecta uma ação considerada má. Desse modo, como a reputação do réu para o júri depende da taxa de recepção de mensagens de evidência, ela acaba por depender da taxa de ações más realizadas, ou seja, da natureza do réu. Uma maneira de controlar o valor da reputação do réu que o júri possui é alterando-se o valor do parâmetro T_R, que é o temporizador do contador que incrementa a reputação. Dessa maneira quando o parâmetro T_R é menor, a reputação no júri cresce mais rapidamente, então para ser excluído um nó precisa realizar ações más com maior frequência. Assim, com o parâmetro T_R menor, o réu é excluído somente quando possui natureza menor. A Figura 5.4(a) mostra que com a configuração de ACACIA_1, o mecanismo exclui o réu quando sua natureza é menor que o limiar e não o exclui quando sua natureza é maior que o limiar. No entanto, a exclusão não é precisa e ocorrem falsos negativos e falsos positivos, correspondentes às regiões cinza e hachurada respectivamente mostradas na Figura 5.4(c). Ao se aumentar o valor de T_R, como nas configurações ACACIA_2 e ACACIA_3, a reputação no júri cresce mais vagarosamente e portanto, o réu é excluído até quando possui natureza mais alta que o limiar de natureza da rede.

Para realizar a comparação da acurácia e precisão dos mecanismos de exclusão serão considerados os erros de falsos positivos e negativos. A Figura 5.5 mostra histogramas dos erros dos diferentes mecanismos de exclusão em relação à distância entre o limiar de natureza da rede e a natureza de fato do nó analisado. Assim, todo erro à esquerda de 0 é resultado de falso negativo e à direita de falso positivo. A Figura 5.5(a) mostra o histograma dos erros do mecanismo proposto, a partir do qual conclui-se que os erros do mecanismo são poucos e bem concentrados no limiar. Já o ACACIA_1 os erros são em torno de 0 o que indica a ocorrência de tanto falsos positivos quanto negativos, mas são mais dispersos como mostrado na Figura 5.5(b) se comparados com o mecanismo proposto. Conformes as Figuras 5.5(c) e 5.5(d), tanto o ACACIA_2 e o ACACIA_3 tem erros somente a partir de 0, o que indica unicamente erros de falsos positivos somente.

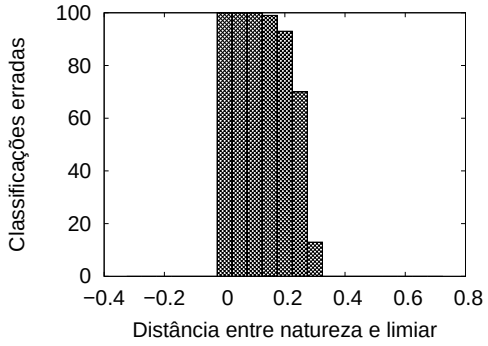
Para realizar a comparação da qualidade desses mecanismos, definem-se métricas de acurácia e precisão de modo que a acurácia signifique a capacidade do mecanismo de acertar a exclusão e não-exclusão de nós relativamente ao limiar de natureza da rede, e a precisão signifique a dispersão da exclusão para diferentes valores de natureza. Assim, utiliza-se os histogramas da Figura 5.5 normalizados, que os transforma em gráficos de densidades de probabilidade de erro da exclusão em relação à distancia da natureza e o limiar de natureza. Em seguida, mede-se a média e o desvio padrão do gráfico de densidade de probabilidade de erro e obtém-se a Figura 5.6.



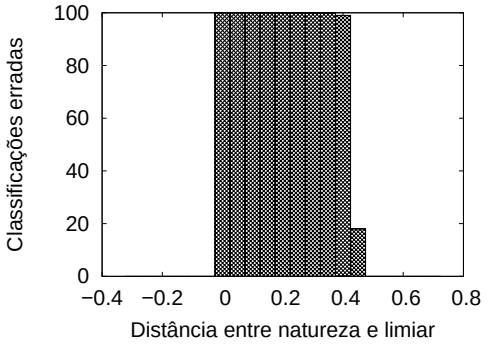
(a) Histograma de erros do mecanismo proposto.



(b) Histograma de erros do ACACIA_1.



(c) Histograma de erros do ACACIA_2.



(d) Histograma de erros do ACACIA_3.

Figura 5.5: Histograma dos erros de exclusão que o mecanismo proposto comete em relação à distância entre o limiar de natureza e a natureza do nó.

Quanto menor o valor da média dos erros em módulo, o mecanismo de exclusão é mais acurado ao excluir os nós, pois não apresenta erros de falsos positivos e negativos longe do limiar de natureza, o mecanismo acerta as exclusões e não-exclusões com naturezas nas redondezas do limiar. Por sua vez, quanto menor o desvio padrão, o mecanismo é mais preciso, pois o mecanismo concentra os erros próximos à média. Dessa maneira, a partir da Figura 5.6, conclui-se que tanto o mecanismo proposto e o ACACIA_1 possuem alta acurácia, pois as médias dos erros estão bem próximas 0, portanto, próximas ao limiar de natureza. Além disso, o mecanismo proposto possui alta precisão, como visto pelo baixo valor do desvio padrão dos erros, e assim concentra os erros bem próximos da média e não apresenta erros de falsos positivos e negativos muito dispersos. Assim, o mecanismo proposto consegue diferenciar com acurácia e precisão os diferentes valores de natureza dos nós. O mesmo não ocorre com o ACACIA_1 que possui baixa precisão como verificado pelo alto valor de desvio padrão. Tanto ACACIA_2 e ACACIA_3 concentram seus erros em falsos positivos como verificado pelo alto deslocamento do valor da média para o eixo positivo, também possuem baixa precisão pois cometem os erros de falsos positivos até altos valores de natureza.

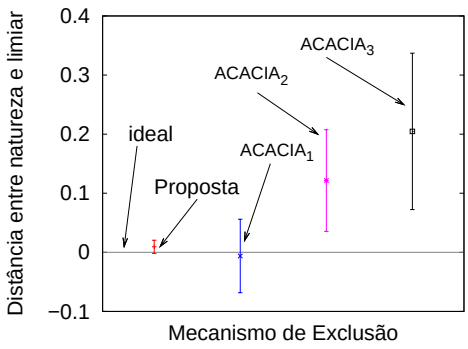
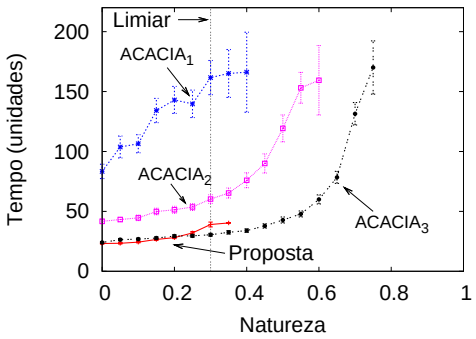


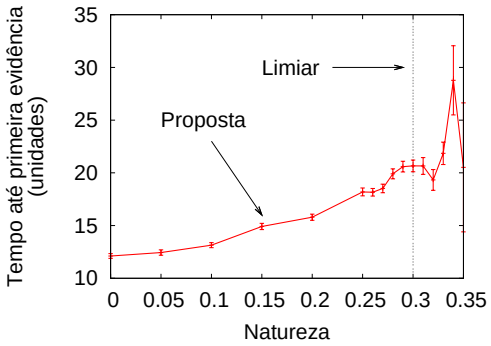
Figura 5.6: Média e desvio padrão do histograma normalizado de erros do mecanismo de exclusão proposto e ACACIA com diferentes configurações. O valor da média é traduzido como a acurácia e o desvio padrão como a precisão do mecanismo de exclusão.

No mecanismo proposto, o tempo de exclusão e a quantidade de evidências pode ser configurado de acordo com os parâmetros u, T_R e T_E como visto na Seção 4.1. Assume-se uma configuração que um jurado deve receber pelo menos uma soma de 80 mensagens de evidência de todas as testemunhas, por um tempo de 10 unidades de tempo para votar pela exclusão. Dessa maneira, no cenário que o nó tem 8 vizinhos, para o jurado excluir o nó, ele deve receber seguidamente 10 mensagens de evidência de cada vizinho, de forma que não dê tempo para a reputação aumentar. Assim, uma eventual redução do valor de confiança abaixo do limiar tolerado não causa a exclusão imediata. Para a reputação diminuir a ponto de o jurado votar pela exclusão do nó, o valor de confiança deve permanecer abaixo do limiar por certo tempo, para que sejam enviadas mensagens de evidência suficientes para tal. A Figura 5.7(a) mostra que o tempo necessário para a exclusão do nó é sempre maior que 20 unidades de tempo. Isso ocorre, pois o modelo de confiança utilizado não avalia um valor de confiança até que o módulo de monitoramento transmita ao menos 10 avaliações de comportamento para evitar grandes flutuações no valor de confiança. Como o módulo de monitoramento avalia um valor de comportamento a cada ação realizada, são necessárias 10 ações para se ter 10 valores de comportamento, que em média são realizadas em 10 unidades de tempo. Além disso, quando a natureza do nó analisado é maior, a porcentagem de ações más é menor e, portanto, o modelo de confiança demora mais tempo para convergir para um valor de confiança menor que o limiar, e causa um tempo maior até os jurados votarem pela exclusão do réu. A Figura 5.7(b) mostra o instante de tempo que as testemunhas enviam a primeira mensagem de evidência a respeito do nó analisado. Nesse instante é quando o modelo de confiança converge para um valor de confiança menor que o limiar de natureza configurado para a rede. Pode-se perceber que quanto menor a natureza a mensagem evidência é enviada com mais antecedência. Isso ocorre, pois os valores

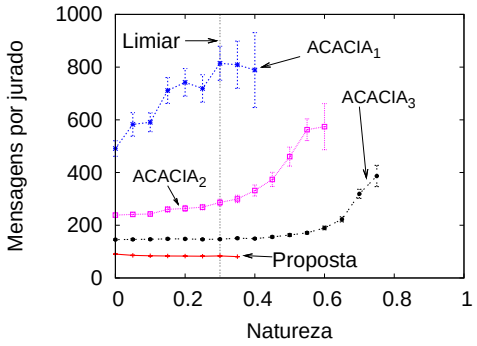
de comportamento avaliados pelo mecanismo de monitoramento são menores devido à maior taxa de ações más, assim o modelo de confiança converge mais rapidamente para um valor menor que o tolerado na rede. Quando a natureza do nó analisado é 0, as mensagens de evidência são enviadas em aproximadamente 12 unidades de tempo, o que corresponde a aproximadamente 10 ações más. Os instantes que o sistema ACACIA envia as primeiras mensagens de evidência não foram apresentados na Figura 5.7(b), pois representam somente os instantes que as testemunhas detectam uma ação má.



(a) Tempo até a exclusão do nó analisado.



(b) Tempo até o envio da primeira mensagem de evidência do mecanismo de exclusão proposto.



(c) Sobrecarga de mensagens de controle relativo ao número médio de mensagens de evidências recebidas por jurado até a exclusão do nó analisado.

Figura 5.7: Tempo necessário para se conseguir a exclusão nó analisado e sobrecarga de mensagens de controle, relativas ao número de mensagens de evidência com oito vizinhos.

A Figura 5.7(c) mostra a sobrecarga de mensagens de controle relativas ao número médio de mensagens de evidências que cada jurado recebe até a exclusão do nó analisado, para o caso do nó ter uma natureza menor que o limiar e ter que ser excluído. Essas mensagens causam grande impacto na rede, pois são destinadas aos jurados que estão distribuídos na rede. O mecanismo proposto foi configurado para que o jurado receba pelo menos 80 mensagens de evidência para votar pela exclusão.

Quando o valor de confiança local das testemunhas acerca do nó analisado é menor que o limiar de natureza, as testemunhas enviam evidências em intervalos determinados. Como as testemunhas possuem valores semelhantes de confiança acerca do nó réu devido às recomendações, elas enviam as evidências em geral ao mesmo tempo. Como consequência, o jurado recebe constantemente evidências até o valor de reputação ser o necessário para que ele vote na exclusão do réu, de maneira que a reputação não aumente. Assim, o número de evidências recebidas por jurado é em geral o previsto, ou seja, 80. Mais uma vez, quando a natureza do nó analisado fica perto do limiar de natureza, o valor de confiança é por vezes maior que o limiar. Assim, as testemunhas param de enviar mensagens de evidências, e retornam a enviá-las assim que a confiança é novamente menor que o limiar. Logo, o número de mensagens de evidência enviadas aos jurados é maior.

No sistema ACACIA, como o T_R deve ser pequeno para controlar a acurácia da exclusão de réu, a taxa de crescimento de reputação causada pelo T_R atua ao mesmo tempo da taxa de redução de reputação causada pela recepção das mensagens de evidência. Assim, quanto maior a taxa de crescimento de reputação (causada pela diminuição de T_R ACACIA$_1$), a taxa de redução de reputação resultante é menor. Logo, a exclusão de um nó ocorre em maior tempo, o que causa maior número de mensagens de evidência, uma vez que a taxa de envio não muda. Ademais, como maiores valores de natureza causam uma taxa de envio de evidências menor, o tempo até a exclusão é maior e consequentemente o número de mensagens de evidências maior.

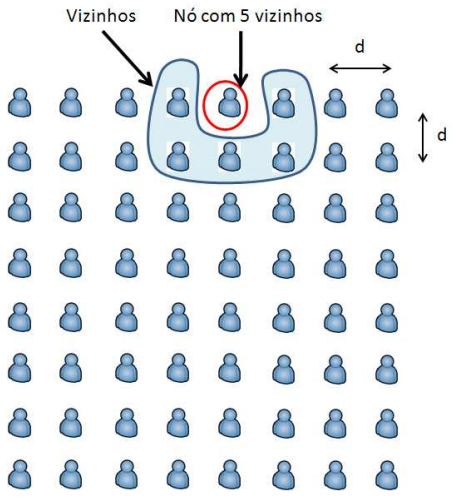
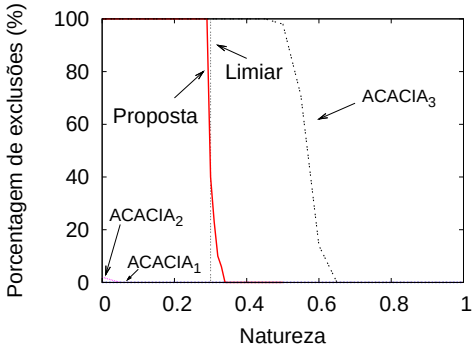


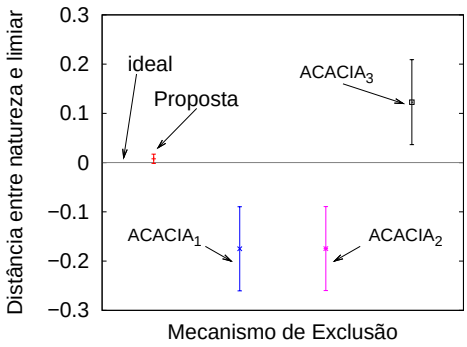
Figura 5.8: A posição do nó analisado em uma das arestas, com cinco vizinhos.

A Figura 5.8 mostra a posição do nó analisado na simulação com o nó posicionado em uma das arestas e, portanto, o nó possui cinco vizinhos diretos. Assim, como nessa simulação somente cinco testemunhas monitoram o nó analisado, espera-se que

o modelo de confiança do mecanismo de exclusão de dois níveis demore mais para convergir para um valor. Além disso, como são menos testemunhas que enviam mensagens de evidência, menor é a taxa de redução de reputação no júri, então demora mais para a reputação ser reduzida ao ponto de se votar pela exclusão do réu.



(a) Porcentagem de rodadas que houve exclusão de nós.

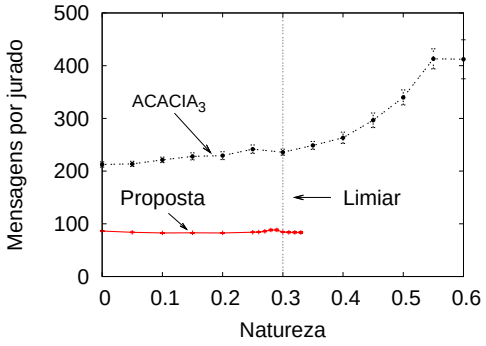


(b) Média e desvio padrão do histograma normalizado dos erros de exclusão.

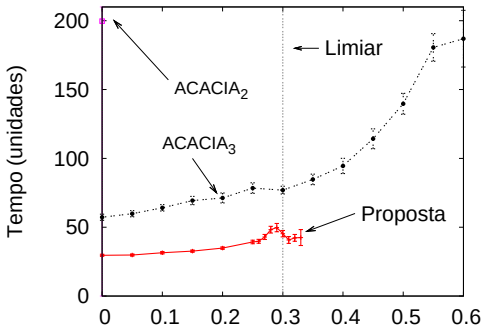
Figura 5.9: Resultados de desempenho de exclusão de um nó com cinco vizinhos.

A Figura 5.9(a) mostra a porcentagem das rodadas que são necessárias para a maioria dos jurados decidirem pela exclusão do nó analisado e a Figura 5.9(b) mostra a média e desvio padrão dos erros dos mecanismos de exclusão. O mecanismo de reputação/confiança de dois níveis proposto permanece acurado e preciso e com baixa taxa de erros, apesar da mudança do número de vizinhos com muito poucos falsos positivos e falsos negativos. Por outro lado, o sistema ACACIA se mostrou extremamente dependente do número de vizinhos, pois, como a quantidade de vizinhos é menor, a taxa de recebimento de mensagens de evidência no júri é menor e, consequentemente, também é menor a taxa de redução de reputação. Assim, o tempo até a exclusão também é maior como mostrado na Figura 5.10(b). Por sua vez, como mostrado na Figura 5.10(a), a quantidade total de mensagens de evidência recebidas é aproximadamente a mesma, pois a taxa de crescimento de reputação no júri não é alterada. Isso ocorre, pois o parâmetro T_R foi configurado com um valor alto, e mesmo com menos vizinhos enviando mensagens de evidência, o tempo entre a recepção não é o suficiente para que se incremente a reputação. Observa-se que tanto ACACIA$_1$ quanto ACACIA$_2$ não chegam nem a conseguir excluir o nó malicioso analisado porque o número de testemunhas não consegue enviar as evidências necessárias.

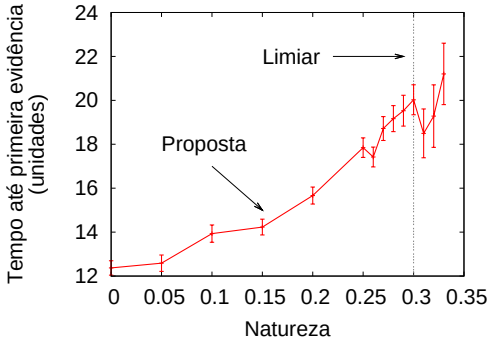
No sistema ACACIA, a acurácia depende diretamente da taxa de recepção de mensagens de evidência. Dessa maneira, quando um réu possui um valor de vizinhos diferente, a taxa conjunta de envio de mensagens de evidência é alterada e consequentemente a taxa de recepção de mensagens de evidência no júri. Como o



(a) Número médio de mensagens de evidências recebidas por jurado até a exclusão do nó analisado.



(b) Tempo até a exclusão do nó analisado.



(c) Tempo até o envio da primeira mensagem de evidência do mecanismo de exclusão proposto.

Figura 5.10: Resultados de desempenho de tempo e número de mensagens até a exclusão de um nó com cinco vizinhos.

número de vizinhos é menor, a taxa de recepção de mensagens de evidência no júri também é menor e consequentemente a taxa de redução de reputação. Ademais, a taxa de crescimento tende a ser maior, pois com menos evidências recebidas, é maior a probabilidade de ter um intervalo suficiente para aumentar a reputação. Essa mudança é análoga a um aumento da taxa de crescimento da reputação. Desse modo, o sistema ACACIA$_1$ e ACACIA$_2$, que estão configurados com taxa de crescimento de reputação maiores, não são capazes de excluir o réu, mesmo quando sua natureza é baixa. O sistema ACACIA$_3$, que com oito vizinhos exclui o réu com natureza até $0,73$ em média, com cinco vizinhos exclui o réu com natureza até $0,57$ em média. Esse comportamento é semelhante ao sistema ACACIA$_2$, assim como o tempo e a quantidade de mensagens de evidência enviadas até a exclusão.

A Figura 5.11 mostra a posição do nó analisado na simulação com o nó posicionado em um dos vértices da grade, assim o nó possui três vizinhos diretos.

Nessa configuração, o nó analisado possui ainda menos testemunhas, mas o mecanismo proposto ainda assim é capaz de excluir acurada e precisamente como mos-

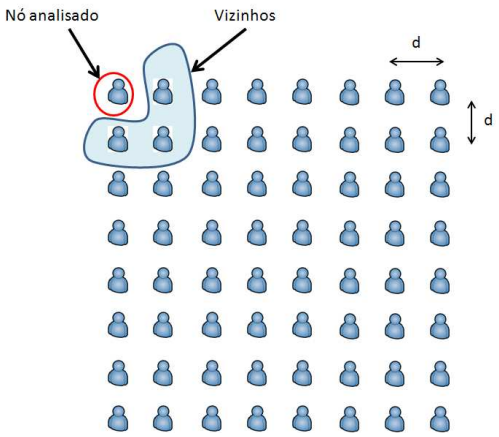
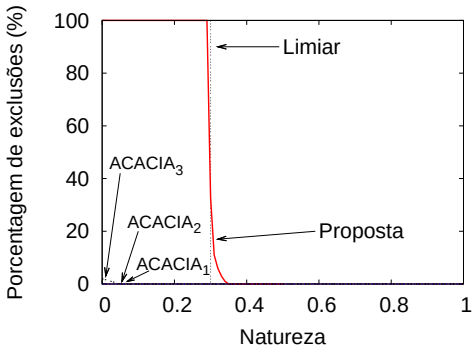
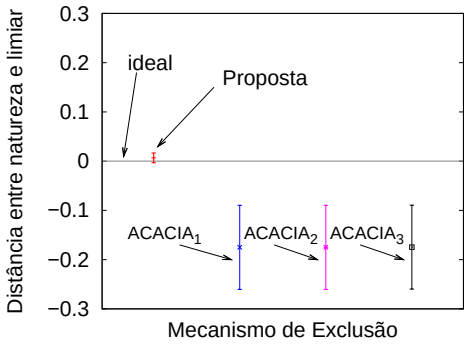


Figura 5.11: A posição do nó analisado na simulação com três vizinhos.



(a) Porcentagem de rodadas que houve exclusão de nós.



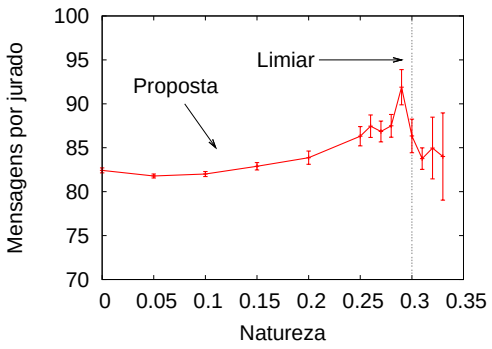
(b) Média e desvio padrão do histograma normalizado dos erros de exclusão.

Figura 5.12: Resultados de desempenho de exclusão de um nó com três vizinhos.

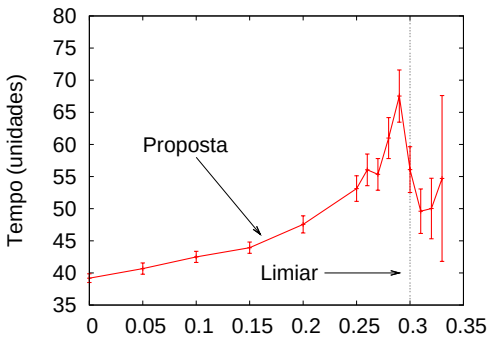
trado pelas exclusões realizada como na Figura 5.12(a), e pelos erros cometidos 5.12(b). A Figura 5.13(b) mostra que, como no caso do nó analisado com cinco vizinhos, com três vizinhos o tempo até a exclusão é ainda maior. Ademais, assim como no caso do nó analisado com cinco vizinhos, a quantidade total de mensagens de evidência recebidas pelos jurados é aproximadamente a mesma, como mostrado na Figura 5.13(a).

No sistema ACACIA nessa configuração, as testemunhas não são suficientes para ocorrer a exclusão. Nesse caso, a taxa de redução de reputação no júri não é pequena em módulo, e a taxa de crescimento aumenta ainda mais pelos mesmos motivos explicados no caso com cinco vizinhos. Assim, a taxa de variação total da reputação é positiva (na maioria dos casos), e o sistema não é capaz de realizar a exclusão.

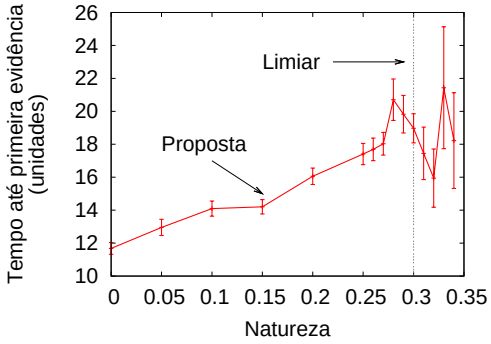
Esses resultados mostram que o mecanismo proposto exclui os nós com natureza menor que o limiar tolerado na rede acurada e precisamente. A exclusão acurada e precisa ocorre mesmo se a quantidade de vizinhos for modificada o que não acontece com o sistema ACACIA.



(a) Número médio de mensagens de evidências recebidas por jurado até a exclusão do nó analisado.



(b) Tempo até a exclusão do nó analisado.



(c) Tempo até o envio da primeira mensagem de evidência do mecanismo de exclusão proposto.

Figura 5.13: Resultados de desempenho de tempo e número de mensagens até a exclusão de um nó com três vizinhos.

5.2.2 Avaliação de Robustez

As simulações anteriores avaliam o desempenho da exclusão de um nó com certa natureza. Essas simulações consideram que o módulo de monitoramento funciona perfeitamente, ou seja, todas as ações que os vizinhos realizam são percebidas e corretamente classificadas. Contudo, os mecanismos de monitoramento nem sempre são capazes de monitorar completamente o ambiente, ou ainda podem ser configurados para funcionar intermitentemente para economizar os recursos do dispositivo. Dessa maneira, foram realizadas simulações que avaliam a robustez do mecanismo de exclusão de nós no caso de falhas no mecanismo de monitoramento. Dois tipos de falhas do mecanismo de monitoramento foram consideradas: falhas na detecção de ações e erros na classificação de ações.

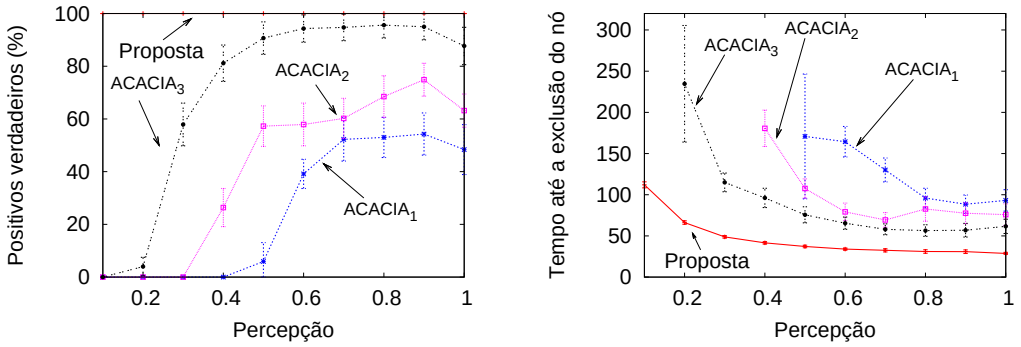
A topologia utilizada nas simulações de robustez é a mesma de 64 nós em grade. Nesse caso, não é escolhido um único nó para ser analisado. Nessas simulações, 10% dos nós da rede são considerados maliciosos e possuem natureza 0. Os nós maliciosos são escolhidos aleatoriamente, e podem estar posicionados tanto no quadrado

6x6 central, quanto nas arestas ou vértices. Todos os outros nós são considerados altruístas e possuem natureza 1. Essas simulações tiveram 20 rodadas e a barra de erro corresponde a um intervalo de confiança de 95%.

Falhas de Detecção de Ações

As falhas de detecção de ações foram modeladas pelo parâmetro percepção que indica a porcentagem das ações que o módulo de monitoramento considera. Desse modo, todas as ações detectadas dos vizinhos têm uma probabilidade de serem consideradas ou não. As simulações avaliam a eficiência em excluir os nós maliciosos e o tempo até suas exclusões, para diferentes valores de percepção dos nós.

A Figura 5.14(a) mostra a eficiência em excluir os nós maliciosos. Como pode-se perceber, o mecanismo de exclusão de dois níveis consegue excluir todos os nós maliciosos apesar de uma percepção baixa. Entretanto, como mostrado na Figura 5.14(b), o tempo até a exclusão dos nós maliciosos é maior em relação ao sistema com percepção alta, ex. demora 112 unidades de tempo em média para excluir quando os nós têm 10% de percepção. Como apenas 10% das ações são consideradas para gerar os valores de comportamento e o modelo de confiança exige a mesma quantidade de valores de comportamento para gerar a confiança dos nós, então espera-se que os valores de comportamento sejam gerados em um tempo 10 vezes maior. Dessa maneira, o módulo de confiança demora por volta de 100 unidades de tempo para gerar um valor de confiança, mas ao concluir que o valor de confiança é menor que a tolerada na rede, o módulo de evidências envia prontamente as mensagens de evidência com a taxa determinada. Ao se aumentar a percepção, o tempo até a exclusão diminui até chegar no tempo configurado para condições ideais.



(a) Taxa de positivo verdadeiro. (b) Tempo até a exclusão dos nós maliciosos.

Figura 5.14: Simulação de robustez a falhas de detecção do módulo de monitoramento.

No sistema ACACIA por sua vez, como o envio de evidências depende somente da taxa de ações más percebidas pelas testemunhas, a falta de percepção faz com

que a taxa de envio de mensagens de evidências seja semelhante a uma taxa que seria enviada no caso de um nó com natureza maior. Assim, o monitoramento com 10% de percepção de um nó cuja natureza é 0 (só realiza ações más), faz com que a taxa de ações más seja um décimo da taxa real. As equações abaixo mostram o cálculo da natureza percebida:

$$\eta_{\text{ACACIA}} = 1 - \lambda_{AM} \tag{5.1}$$

$$\eta_{p|\text{ACACIA}} = 1 - \rho \cdot \lambda_{AM} \tag{5.2}$$

onde λ_{AM} é a taxa de ações más realizadas, η_{ACACIA} é natureza calculada pelo sistema ACACIA e $\eta_{p|\text{ACACIA}}$ a natureza percebida, e ρ o valor de percepção. Desse modo, com o monitoramento com 10% de percepção a natureza percebida é de $0,9$ como mostrado pela Equação 5.3. A percepção de um valor de natureza maior que a real acontece para todas as configurações do sistema ACACIA, e assim, o sistema fica com altos valores de falsos negativos. Os tempos até a exclusão dos nós com o sistema ACACIA também ficam equivalentes ao tempo gasto para excluir um nó com natureza mais alta. Além disso, mesmo com a percepção alta, a posição dos nós maliciosos pode fazer com que o nó tenha menos vizinhos e assim o desempenho da exclusão de nós é afetado, e pode inclusive resultar em uma maior taxa de falsos negativos no caso do sistema ACACIA, como mostrado nas simulações de desempenho com menos vizinhos.

$$\eta_p = 1 - \rho \cdot \lambda_{AM} = 1 - 0,1 \cdot 1 = 0,9 \text{ se } \rho = 0,1 \text{ e } \lambda_{AM} = 1 \tag{5.3}$$

Falhas de Classificação de Ações

As falhas de classificação de ações foram modeladas pelo parâmetro probabilidade de erro de classificação de ações que indica uma porcentagem das ações que são classificadas de forma errada pelo módulo de monitoramento. Desse modo, todas as ações boas detectadas dos vizinhos têm uma probabilidade de serem consideradas más e vice-versa. Nessas simulações, a percepção é máxima, ou seja, $1,0$. As simulações avaliam a eficiência em excluir os nós maliciosos e o tempo até suas exclusões, para diferentes valores da probabilidade de erro de classificação de ações.

A classificação errada das ações resulta na percepção errada do valor de natureza, e assim das taxas de execução de ações boas e más. As equações abaixo mostram as taxas de execução das ações boas e más percebidas baseadas na probabilidade de erro de classificação:

$$\lambda_A = \lambda_{AB} + \lambda_{AM} \tag{5.4}$$

$$\lambda_{ABe} = (1 - \kappa)\lambda_{AB} + \kappa \cdot \lambda_{AM} \tag{5.5}$$

$$\lambda_{AMe} = \kappa \cdot \lambda_{AB} + (1 - \kappa)\lambda_{AM} \tag{5.6}$$

onde λ_A é a taxa de ações realizadas, λ_{AB} é a taxa de ações boas realizadas, λ_{AM} é a taxa de ações más realizadas, λ_{ABp} e λ_{AMp} são as taxa de ações boas e más percebidas, e κ a probabilidade de erro de classificação. O mecanismo proposto considera os dois tipos de ações para gerar um valor de confiança, portanto a natureza percebida do mecanismo proposto é

$$\eta = \frac{\lambda_{AB}}{\lambda_A} \tag{5.7}$$

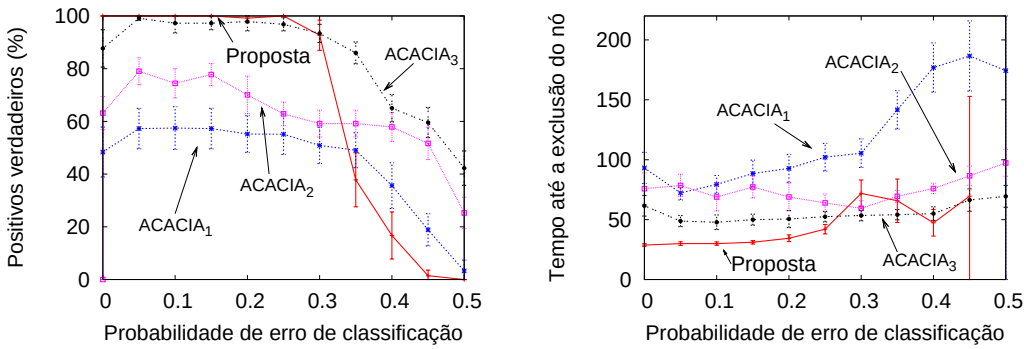
$$\eta_p = \frac{\lambda_{ABe}}{\lambda_A} = \frac{(1 - \kappa)\lambda_{AB} + \kappa \cdot \lambda_{AM}}{\lambda_{AB} + \lambda_{AM}} \tag{5.8}$$

Dessa maneira, a natureza percebida se aproxima do extremo oposto, ou seja, $1-\eta$ de acordo com um aumento da probabilidade de erro de classificação. Nessa simulação, os nós possuem duas naturezas possíveis, 0 ou 1. Desse modo, no caso de um erro de classificação de ações de 30%, significa que a natureza percebida pelo mecanismo é $0,3$ como mostrado pela Equação 5.9.

$$\eta_p = \frac{(1 - \kappa)\lambda_{AB} + \kappa \cdot \lambda_{AM}}{\lambda_A} = \frac{0,7 \cdot 0 + 0,3 \cdot 1}{1} = 0,3 \text{ se } \kappa = 0,3 \text{ e } \lambda_A = \lambda_{AM} = 1 \tag{5.9}$$

Esse resultado pode ser verificado pela Figura 5.15(a), na qual a taxa de positivos verdadeiros reduz quando a probabilidade de erro do classificador é 30%. Isso ocorre devido ao fato de a partir desse ponto o mecanismo percebe uma natureza maior que o limiar, e portanto, não considera o nó como maliciosos e não o exclui.. A Figura 5.15(b) mostra que o tempo até a exclusão dos nós maliciosos aumenta com a probabilidade de erro de classificação de ações. Isso ocorre pelo mesmo motivo que aumenta com a natureza, pois o aumento na probabilidade de erro de classificação causa aumento da natureza percebida.

No caso do sistema ACACIA a natureza percebida depende unicamente das taxas de ações más realizadas. As equações mostram como o sistema percebe a natureza quando ocorrem erros de classificação de ações:



(a) Taxa de positivo verdadeiro.

(b) Tempo até a exclusão dos nós maliciosos.

Figura 5.15: Simulação de robustez a erros de classificação de ações do módulo de monitoramento.

$$\eta_{p|\text{ACACIA}} = 1 - \lambda_{AMe} = 1 - \kappa \cdot \lambda_{AB} + (1 - \kappa)\lambda_{AM} \tag{5.10}$$

A natureza percebida também se aproxima de $1 - \eta$ de acordo com um aumento da probabilidade de erro de classificação, mas como o sistema não é muito preciso, a queda na taxa de positivos verdadeiros é mais sutil.

Então como mostrado, o mecanismo proposto ao utilizar o sistema de confiança de dois níveis, realiza a exclusão de nós não cooperativos com robustez a falhas de percepção das ações dos vizinhos e classificações das ações. Além disso, o mecanismo aprimora a eficiência e a acurácia das exclusões, pois o mecanismo proposto é capaz de detectar nós com naturezas diferentes, enquanto o sistema ACACIA não distingue nós que realizam ações boas e más de nós que realizam só ações más com a mesma frequência de ações más.

Capítulo 6

Conclusões

6.1 Considerações Finais

As redes ad hoc móveis (MANETs) são redes de comunicação sem-fio cujos nós são móveis, e não possuem qualquer infraestrutura física. Essas características fazem esse tipo de rede ser atrativo, mas também acompanham certas questões em relação à segurança da rede. Nesse tipo de rede, os nós devem gastar seus próprios recursos para servir outros sem um benefício direto, devem assim agir cooperativamente. Além disso, a facilitação do acesso à rede pode trazer nós maliciosos para a rede, então a rede deve ser capaz de se proteger contra possíveis ameaças. Dessa maneira, utiliza-se um sistema de controle de acesso que controla a entrada de nós na rede e baseado em um sistema de reputação pune os nós mal comportados da rede para forçá-los a cooperar. Assim, após os nós entrarem na rede, eles devem comportar-se adequadamente e agir de maneira cooperativa para manter uma reputação alta e não serem privados do acesso à rede. Entretanto, a identificação dos nós maliciosos é uma questão que deve ser tratada com cautela, pois deseja-se o máximo de participação de nós, e ao mesmo tempo deseja-se que nenhum nó mal comportado permaneça na rede. Então, deve-se evitar a punição excessiva de nós da rede e expulsar da rede somente nós que realmente mereçam. Dessa maneira, o mecanismo de exclusão de nós deve ser altamente acurado e preciso.

Assim, a dissertação tem como objetivo apresentar um mecanismo que faça o controle de acesso e realize a exclusão de nós não cooperativos de maneira acurada e precisa. O mecanismo foi inspirado em um tribunal de júri, de maneira que todos os nós da rede possuam múltiplos papéis. Assim, os nós são responsáveis pelo monitoramento da vizinhança (papel de testemunha), exclusão de nós (papel de jurado), e também estão sujeitos ao tribunal de júri (papel de réu). O mecanismo proposto baseia-se no sistema ACACIA [49] e no modelo de confiança [59], então realiza o controle de acesso de maneira totalmente distribuída, dinâmica, resistente

a conluios. Ao mesmo tempo, o mecanismo faz uma análise acurada, precisa e com baixa sobrecarga de mensagens, do comportamento dos nós da rede para decidir se eles devem ou não permanecer na rede. Além disso, o mecanismo é capaz de fazer a exclusão acurada dos nós mal comportados, mesmo quando o módulo de monitoramento falha.

6.2 Trabalhos Futuros

O modelo de confiança de segundo nível utilizado não considera a confiabilidade das testemunhas, assim as evidências de uma testemunha que não seja muito confiável têm o mesmo peso de testemunhas confiáveis. Além disso, o valor de reputação é uma medida indireta da confiabilidade do nó, baseada na frequência de evidências recebidas. Para considerar esse aspecto, pode-se aplicar o modelo de confiança apresentado na Seção 3.1.2 no modelo de confiança de segundo nível.

Desse modo, as testemunhas enviam suas recomendações nas mensagens de evidência. Uma vez que o júri não participa do monitoramento direto do réu, o júri não pode ter uma avaliação própria acerca do réu. Então, para o cálculo do valor de reputação, só se considera as recomendações das testemunhas, ou seja, a reputação do réu no júri é obtida através da Equação 3.1 com o parâmetro $\alpha = 1$. A equação a seguir mostra o cálculo da reputação:

$$Reputação_{\mathbf{d|j}} = Recomendações_{\mathbf{w}}(\mathbf{d}) \tag{6.1}$$

Assim, um jurado recebe as mensagens de evidência com o valor local de confiança, a acurácia e maturidade da relação, e faz o cálculo da reputação. O jurado obtém a reputação das testemunhas ao perguntar para os júris delas. Dessa maneira, evidências de testemunhas com baixa reputação têm menos importância no cálculo de reputação.

Entretanto, essa abordagem requer muitas mensagens de controle para a obtenção de reputação das testemunhas, cada jurado deve enviar requisições de reputação para o júri de cada testemunha. Assim, esse cenário cresce com o número médio de testemunhas e jurados que compõem o júri. A equação a seguir mostra o número necessário de mensagens de controle C, para um jurado calcular uma iteração da reputação de um réu, desconsideradas possíveis iterações posteriores:

$$C = \nu + \nu \cdot 2|J| \tag{6.2}$$

onde ν é o número médio de vizinhos na rede. Para diminuir a quantidade de mensagens de controle, as testemunhas enviam aleatoriamente as evidências para um grupo menor de jurados. Da mesma maneira, para obter as reputações das testemu-

nhas, os jurados consultam aleatoriamente um grupo de jurados das testemunhas menor que o júri completo de cada uma delas.

Referências Bibliográficas

[1] MOREIRA, M. D. D., FERNANDES, N. C., COSTA, L. H. M. K., et al. "Internet do Futuro: Um Novo Horizonte", *Minicursos do Simpósio Brasileiro de Redes de Computadores - SBRC'2009*, pp. 1–59, 2009.

[2] CAMPISTA, M. E. M., FERRAZ, L. H. G., MORAES, I. M., et al. "Interconexão de Redes na Internet do Futuro: Desafios e Soluções", *Minicursos do Simpósio Brasileiro de Redes de Computadores - SBRC'2010*, pp. 47–101, 2010.

[3] MARTI, S., GIULI, T. J., LAI, K., et al. "Mitigating routing misbehavior in mobile ad hoc networks". In: *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, pp. 255 –265, 2000.

[4] FERNANDES, N. C., MOREIRA, M. D. D., VELLOSO, P. B., et al. "Ataques e Mecanismos de Segurança em Redes Ad Hoc", *Minicursos do Simpósio Brasileiro de Redes de Computadores - SBRC'2006*, pp. 1–54, 2006.

[5] PAPADIMITRATOS, P., HAAS, Z. J. "Securing Mobile Ad Hoc Networks". In: *in Handbook of Ad Hoc Wireless Networks, M. Ilyas, Editor. 2002.* CRC Press, 2002.

[6] RIVEST, R. L. *The MD5 Message-Digest Algorithm.* IETF, RFC 1321, abr. 1992.

[7] NATIONAL INSTITUTE OF STANDARDS. *Secure hash standard.* FIPS 180-2, ago. 2000.

[8] NATIONAL BUREAU OF STANDARDS. *Data Encryption Standard.* FIPS-Pub.46, jan. 1977.

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

[10] KALISKI, B., STADDON, J. *PKCS #1: RSA Cryptography Specifications Version 2.0.* IETF, RFC 2437, out. 1998.

[11] RESCORLA, E. *Diffie-Hellman Key Agreement Method*. IETF, RFC 2631, jun. 1999.

[12] IEEE. *IEEE Standard Specifications for Public-Key Cryptography*. IEEE Std 1363-2000, ago. 2000.

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

[14] KENT, S., SEO, K. *Diffie-Hellman Key Agreement Method*. IETF, RFC 4301, dez. 2005.

[15] MOSKOWITZ, R., NIKANDER, P. *Host Identity Protocol Architecture*. IETF, RFC 4423, maio 2006.

[16] ZAPATA, M. G. "Secure Ad hoc On-Demand Distance Vector (SAODV) Routing", *ACM Mobile Computing and Communications Review*, v. 6, n. 3, pp. 106 –107, jul. 2002.

[17] PERKINS, C. E., M.BELDING-ROYER, E., DAS, R. S. *Ad Hoc On-Demand Distance Vector Routing*. RFC: 3561, jul. 2003.

[18] TØNNESEN, A. *Implementing and Extending the Optimized Link State Routing Protocol*. Tese de Mestrado, University of Oslo, ago. 2004.

[19] ADJIH, C., CLAUSEN, T. H., JACQUET, P., et al. "Securing the OLSR protocol". In: *IFIP Med-Hoc-Net*, pp. 1 –10, jun. 2003.

[20] JACQUET, P., MUHLETHALER, P., CLAUSEN, T., et al. "Optimized link state routing protocol for ad hoc networks". In: *INMIC 2001*, pp. 62 –68, dez. 2001.

[21] SANZGIRI, K., DAHILL, B., LEVINE, B. N., et al. "A secure routing protocol for ad hoc networks". In: *10th IEEE International Conference on Network Protocols*, pp. 78 –87, nov. 2002.

[22] HU, Y.-C., PERRIG, A., JOHNSON, D. B. "Ariadne: A secure on-demand routing protocol for ad hoc networks", *Wireless Networks*, v. 11, n. 1–2, pp. 21 –38, jan. 2005.

[23] PAPADIMITRATOS, P., HAAS, Z. J., SAMAR, P. "The secure routing protocol (SRP) for ad hoc networks". In: *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, v. 31, 2002.

[24] PAPADIMITRATOS, P., HAAS, Z., SAMAR, P. *The Secure Routing Protocol (SRP) for Ad Hoc Network*. IETF, dez. 2002.

[25] HU, Y.-C., JOHNSON, D. B., PERRIG, A. "SEAD: Secure Efficient Distance Vector Routing in Mobile Wireless Ad Hoc Networks". In: *Fourth IEEE Workshop on Mobile Computing Systems and Applications* (WMCSA '02), pp. 3 –13, jun. 2002.

[26] PAPADIMITRATOS, P., HAAS, Z. "Secure Link State Routing for Mobile Ad Hoc Networks". In: *IEEE CS Workshop on Security and Assurance in Ad hoc Networks*, pp. 379 –38, jan. 2003.

[27] LI, Q., ZHAO, M., WALKER, J., et al. "SEAR: a secure efficient ad hoc on demand routing protocol for wireless networks", *Security and Communication Networks*, v. 2, n. 4, pp. 325 –340, 2009.

[28] NEUMAN, B. C., TS'O, T. "Kerberos: an authentication service for computer networks", *IEEE Communications Magazine*, v. 32, n. 9, pp. 33 –38, set. 1994.

[29] DESMEDT, Y., FRANKEL, Y. "Threshold cryptosystems". In: *Advances in Cryptology CRYPTO' 89 Proceedings*, pp. 307 –315, 1990.

[30] DESMEDT, Y. "Threshold cryptography", *European Transactions on Telecommunications*, v. 5, n. 4, pp. 449 –458, 1994.

[31] ZHOU, L., HAAS, Z. "Securing ad hoc networks", *Network, IEEE*, v. 13, n. 6, pp. 24 –30, nov. 1999.

[32] LUO, H., LU, S. *Ubiquitous and robust authentication services for ad hoc wireless networks*. Relatório técnico, Citeseer, 2000.

[33] LUO, H., LU, S., ZHANG, L. "Providing robust and ubiquitous security support for mobile ad hoc networks". In: *Proceeding of The 9th International Conference on Network Protocols*, 2001.

[34] SHAMIR, A. "How to share a secret", *Communications of the ACM*, v. 22, n. 11, pp. 612 –613, 1979.

[35] SHAMIR, A. "Identity-based cryptosystems and signature schemes". In: *Advances in cryptology*, pp. 47–53, 1985.

[36] BONEH, D., FRANKLIN, M. "Identity-based encryption from the Weil pairing". In: *Advances in Cryptology CRYPTO 2001*, pp. 213 –229, 2001.

[37] KAPIL, A., RANA, S. "Identity-Based Key Management in MANETs using Public Key Cryptography", *International Journal of Security (IJS)*, v. 3, n. 1, pp. 1, 2009.

[38] KHALILI, A., KATZ, J., ARBAUGH, W. A. "Toward Secure Key Distribution in Truly Ad-Hoc Networks". In: *Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*, SAINT-W '03, pp. 342 –, 2003.

[39] HOEPER, K., GONG, G. "Key revocation for identity-based schemes in mobile ad hoc networks", *Ad-Hoc, Mobile, and Wireless Networks*, pp. 224 –237, 2006.

[40] AL-RIYAMI, S., PATERSON, K. "Certificateless public key cryptography", *Advances in Cryptology-ASIACRYPT 2003*, pp. 452 –473, 2003.

[41] AL-RIYAMI, S. S., PATERSON, K. G., EX, T. "CBE from CL-PKE: A generic construction and efficient schemes". In: *Public Key Cryptography - PKC 2005, Lecture Notes in Comput. Sci*, pp. 398 –415, 2005.

[42] GENTRY, C. "Certificate-based encryption and the certificate revocation problem". In: *Eurocrypt*, pp. 272 –293, 2003.

[43] LIU, J., AU, M., SUSILO, W. "Self-Generated-Certificate Public Key Cryptography and certificateless signature/encryption scheme in the standard model: extended abstract". In: *Proceedings of the 2nd ACM symposium on Information, computer and communications security*, pp. 273 –283, 2007.

[44] BAEK, J., SAFAVI-NAINI, R., SUSILO, W. "Certificateless public key encryption without pairing". In: *Computers and Operations Research*, pp. 134 –148, 2005.

[45] LAI, J., KOU, W. "Self-generated-certificate public key encryption without pairing", *Public Key Cryptography–PKC 2007*, pp. 476 –489, 2007.

[46] GIRAULT, M. "Self-certified public keys". In: *Proceedings of the 10th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'91, pp. 490 –497, 1991.

[47] LAI, J., KOU, W., CHEN, K. "Self-generated-certificate public key encryption without pairing and its application", *Information Sciences*, v. 181, n. 11, pp. 2422 –2435, 2011.

[48] PEDERSEN, T. "A threshold cryptosystem without a trusted party". In: *Proceedings of the 10th annual international conference on Theory and application of cryptographic techniques*, pp. 522 –526, 1991.

[49] FERNANDES, N., MOREIRA, M., DUARTE, O. "A Self-Organized Mechanism for Thwarting Malicious Access in Ad Hoc Networks". In: *INFOCOM, 2010 Proceedings IEEE*, pp. 1 –5, 2010.

[50] PIRZADA, A., MCDONALD, C. "Establishing trust in pure ad-hoc networks". In: *Proceedings of the 27th Australasian conference on Computer science*, v. 26, p. 54. Australian Computer Society, Inc., 2004.

[51] ZHONG, S., CHEN, J., YANG, Y. "Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks". In: *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, v. 3, pp. 1987 –1997, mar. 2003.

[52] SUN, Y., HAN, Z., LIU, K. "Defense of trust management vulnerabilities in distributed networks", *Communications Magazine, IEEE*, v. 46, n. 2, pp. 112 –119, fev. 2008.

[53] MARTIGNON, F., PARIS, S., CAPONE, A. "A framework for detecting selfish misbehavior in wireless mesh community networks". In: *Proceedings of the 5th ACM symposium on QoS and security for wireless and mobile networks*, Q2SWinet '09, pp. 65 –72, 2009.

[54] SONG, C., ZHANG, Q. "COFFEE: A Context-Free Protocol for Stimulating Data Forwarding in Wireless Ad Hoc Networks". In: *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09. 6th Annual IEEE Communications Society Conference on*, pp. 1 –9, jun. 2009.

[55] MOLVA, R., MICHIARDI, P. "Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks". In: *proc. The 6th IFIP Communications and Multimedia Security Conf*, 2002.

[56] BUCHEGGER, S., LE BOUDEC, J.-Y. "Performance analysis of the CONFIDANT protocol". In: *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '02, pp. 226 –236, 2002.

[57] SAFA, H., ARTAIL, H., TABET, D. "A cluster-based trust-aware routing protocol for mobile ad hoc networks", *Wireless Networks*, v. 16, n. 4, pp. 969 –984, 2010.

[58] CHLAMTAC, I., CONTI, M., LIU, J. "Mobile ad hoc networking: imperatives and challenges", *Ad Hoc Networks*, v. 1, n. 1, pp. 13 –64, 2003.

[59] VELLOSO, P., LAUFER, R., DE O CUNHA, D., et al. "Trust Management in Mobile Ad Hoc Networks Using a Scalable Maturity-Based Model", *Network and Service Management, IEEE Transactions on*, v. 7, n. 3, pp. 172 –185, set. 2010.

[60] BUTTYAN, L., HUBAUX, J.-P. "Enforcing service availability in mobile ad-hoc WANs". In: *Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. 2000 First Annual Workshop on*, pp. 87 –96, 2000.

[61] BUTTYAN, L., PIERRE HUBAUX, J. "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks", *ACM/Kluwer Mobile Networks and Applications (MONET*, v. 8, pp. 579 –592, 2003.

[62] CHHABRA, S., SOLIHIN, Y., LAL, R., et al. "An analysis of secure processor architectures", *Transactions on computational science VII*, pp. 101 –121, 2010.

[63] GUPTA, P., KUMAR, P. "The capacity of wireless networks", *Information Theory, IEEE Transactions on*, v. 46, n. 2, pp. 388 –404, mar. 2000.

[64] YANG, H., SHU, J., MENG, X., et al. "SCAN: self-organized network-layer security in mobile ad hoc networks", *Selected Areas in Communications, IEEE Journal on*, v. 24, n. 2, pp. 261 –273, fev. 2006.

[65] ZHANG, Z., CHEN, S., YOON, M. "MARCH: A Distributed Incentive Scheme for Peer-to-Peer Networks". In: *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 1091 –1099, maio 2007.

[66] STOICA, I., MORRIS, R., LIBEN-NOWELL, D., et al. "Chord: a scalable peer-to-peer lookup protocol for internet applications", *IEEE/ACM Trans. Netw.*, v. 11, pp. 17 –32, fev. 2003.

[67] GALUBA, W., PAPADIMITRATOS, P., POTURALSKI, M., et al. "Castor: scalable secure routing for ad hoc networks". In: *INFOCOM, 2010 Proceedings IEEE*, pp. 1 –9, 2010.

[68] ZADEH, L. "Fuzzy sets*", *Information and control*, v. 8, n. 3, pp. 338 –353, 1965.

[69] FERNANDES, N., MOREIRA, M., DUARTE, O. "An Efficient Filter-based Addressing Protocol for Autoconfiguration of Mobile Ad Hoc Networks". In: *INFOCOM 2009, IEEE*, pp. 2464 –2472, abr. 2009.

[70] FERRAZ, L. H. G. *Uma Avaliação do Protocolo HIP para Provisão de Mobilidade na Internet*. Projeto de Fim de Curso, Universidade Federal do Rio de Janeiro, mar. 2010.

[71] FARINACCI, D., FULLER, V., MEYER, D., et al. *Locator/ID Separation Protocol (LISP)*. cisco systems, draft-ietf-lisp-15.txt, jul. 2011.

[72] DROMS, R. *Dynamic Host Configuration Protocol*. Network Working Group, RFC 2131, fev. 2006.

[73] LEMON, T., SOMMERFIELD, B. *Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)*. Network Working Group, RFC 4361, fev. 2006.

[74] ARKKO, J., PIGNATARO, C. *IANA Allocation Guidelines for the Address Resolution Protocol (ARP)*. Network Working Group, RFC 5495, abr. 2009.

[75] LAUFER, R., VELLOSO, P., DUARTE, O. "Generalized bloom filters", *Electrical Engineering Program, COPPE/UFRJ, Tech. Rep. GTA-05-43*, 2005.

[76] "The ns3 Network Simulator". Acessado em http://www.nsnam.org/, jul. 2006. http://www.nsnam.org/.