UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

UM MODELO DE SEGURANÇA PARA REDES MÓVEIS AD HOC

RICARDO STACIARINI PUTTINI

ORIENTADOR: RAFAEL TIMÓTEO DE SOUSA JÚNIOR

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO: 004/2004

BRASÍLIA / DF: OUTUBRO/2004

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

UM MODELO DE SEGURANÇA PARA REDES MÓVEIS AD HOC

RICARDO STACIARINI PUTTINI

TESE DE DOUTORADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR.

APROVADA POR:

RAFAEL TIMÓTEO DE SOUSA JÚNIOR, Doutor, Universidade de Brasília – DF – Brasil (UnB)
(ORIENTADOR)

LUDOVIC MÉ, Docteur, École Supérieure d'Électricité – França (Supélec)
(EXAMINADOR EXTERNO)

WILLIAM FERREIRA GIOZZA, Doutor, Universidade de Salvador – BA – Brasil (UniFacS)
(EXAMINADOR EXTERNO)

ANTÔNIO JOSÉ MARTINS SOARES, Doutor, Universidade de Brasília – DF – Brasil (UnB)
(EXAMINADOR INTERNO)

CLÁUDIA JACY BARENCO ABBAS, Doutor, Universidade de Brasília – DF – Brasil (UnB)
(EXAMINADOR INTERNO)

PAULO HENRIQUE PORTELA DE CARVALHO, Doutor, Universidade de Brasília – DF – Brasil (UnB)
(SUPLENTE)

DATA: BRASÍLIA/DF, 18 DE OUTUBRO DE 2004.

FICHA CATALOGRÁFICA

PUTTINI, RICARDO STACIARINI
Um Modelo de Segurança para Redes Móveis Ad Hoc [Distrito Federal] 2004, 191 pp., 297 mm (ENE/FT/UnB, Doutor, Engenharia Elétrica, 2004).

Tese de Doutorado – Universidade de Brasília, Faculdade de Tecnologia, Departamento de Engenharia Elétrica.

1. Redes Móveis Ad Hoc 2. Certificação Digital Distribuída 3. Autenticação em Redes Móveis *Ad Hoc* 4. Detecção de Intrusão em Redes Móveis *Ad Hoc*

I. ENE/FT/UnB. II. Um Modelo de Segurança para Redes Móveis Ad Hoc

REFERÊNCIA BIBLIOGRÁFICA

PUTTINI, R. S. (2004). Um Modelo de Segurança para Redes Móveis Ad Hoc. Tese de Doutorado, Publicação 004/2004, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 191 pp..

CESSÃO DE DIREITOS

NOME DO AUTOR: Ricardo Staciarini Puttini
TÍTULO DA DISSERTAÇÃO: Um Modelo de Segurança para Redes Móveis Ad Hoc.
GRAU/ANO: Doutor/2004.

Ricardo Staciarini Puttini
SQN 303 Bloco H Apto. 118 – Asa Norte
CEP 70735-080 – Brasília – DF – Brasil

À minha filha Luiza.

AGRADECIMENTOS

Ao meu orientador Prof. Dr. Rafael Timóteo de Sousa Júnior, pelo constante apoio, incentivo, dedicação e amizade, essenciais para o desenvolvimento deste trabalho e para o meu desenvolvimento como pesquisador.

Ao Prof. Ludovic Mé, da École Supérieure d'Électricité (Supélec), verdadeiro co-orientador deste trabalho, pelas sábias e incomensuráveis orientações e pela acolhida e amizade durante minha estadia na França. Merci Beaucoup, Ludo!

Aos Prof. Paulo Henrique Portela de Carvalho e Antônio José Martins Soares, pelas conversas enriquecedoras, colaboração e amizade.

À minha família, em especial aos meus pais Cláuzio e Katea, meus irmãos Simone, Marcelo, Mônica e Solange, minha querida Sílvia e minha amada filha Luiza, pelo eterno amor e carinho.

A todos do Departamento de Engenharia Elétrica da UnB e aos meus colegas do LabRedes, os meus sinceros agradecimentos.

O presente trabalho foi realizado com o apoio da CAPES, uma agência do Governo Brasileiro voltada ao incentivo à capacitação profissional e científica, tendo sido o autor bolsista desta agência durante o período de realização do trabalho de pesquisa no exterior, entre os meses de setembro de 2001 e dezembro de 2002.

À Fundação de Empreendimentos Científicos e Tecnológicos – FINATEC, pelo apoio para participações em conferências internacionais.

RESUMO

Nesta tese é apresentado um modelo para segurança de redes móveis ad hoc (Manet). Uma combinação de serviços de segurança preventiva e corretiva permite o uso do modelo em diversos cenários de aplicação de Manet, inclusive aqueles onde a probabilidade de existência de nodos comprometidos não possa ser desconsiderada. O modelo possui essencialmente três serviços de segurança que interagem entre si. Primeiramente, um serviço de certificação digital permite impor uma política de segurança específica para discriminar nodos confiáveis e não confiáveis na rede, enquanto provê uma forma segura de identificação para os nodos. Um serviço de autenticação possibilita assegurar que mensagens advindas de nodos não confiáveis sejam tratadas de acordo com a política de segurança, podendo ser simplesmente descartadas ou processadas com ressalvas. Finalmente, um serviço de detecção de intrusão permite identificar e eliminar nodos comprometidos na rede. A proposta é imediatamente aplicada na segurança dos protocolos de roteamento e autoconfiguração. Uma implementação real dos serviços propostos serve como prova de conceito para a eficácia dos serviços propostos.

ABSTRACT

This thesis presents a model for securing mobile ad hoc networks (Manet). A combination of preventive and corrective security services allows the use of our model in a broad range of Manet application scenarios, including those where the probability of node compromising should not be neglected. Our model has three security services that interact among each other. First, a digital certification service provides an effective way for node identification and separation of trusted and untrusted nodes. Second, an authentication service ensures that messages originated in untrusted node are processed in conformance with the security policy, which can be even discarded or processed with restrictions. Finally, an intrusion detection service deals with identification and elimination of compromised nodes. The proposal has been applied for securing routing and autoconfiguration Manet protocols. An actual proof-of-concept implementation shows the effectiveness of the proposed services.

ÍNDICE

ÍNDICE DE TABELAS

ÍNDICE DE FIGURAS

LISTA DE ACRÔNIMOS

AAFID – Autonomous Agents For Intrusion Detection
AC – Autoridade Certificadora
ACD – Autoridade Certificadora Distribuída
API – Application Program Interface
AODV – Ad Hoc On Demand Distance Vector
ARAN (*Authenticated Routing for Ad hoc Networks*)
AREQ – Address Request
AREP – Address Reply
CBRP – Cluster Based Routing Protocol
CERT – Computer Emergency Response Team
CRL – Certificate Revocation List
DAD – Duplicated Address Detection
DDoS – Distributed Deny of Service
DHCP – Dynamic Host Configuration Protocol
DS – Digital Signature
DSDV – Destination-Sequenced Distance-Vector
DCDP – Dynamic Configuration Distribution Protocol
DEF – Diagrama de Estado Finito
DSR – Dynamic Source Routing Protocol
ESM – Estação de Suporte à Mobilidade.
f.d.p. – função densidade de probabilidade
f.d.a. – função de densidade acumulativa
FSR – Fisheye State Routing
GMM – Gaussian Mixture Model
HC – Hash Chain
HNA – Host and Network Association
ICP – Infra-estrutura de Chaves Públicas
IDA – Intrusion Detection Agent
IDS – Intrusion Detection System (sistema de detecção de intrusão)
IDWG – Intrusion Detection Working Group
IEEE – Institute of Electrical and Electronics Engineers.
IETF – Internet Engineering Task Force.
IP – Internet Protocol
IPSec – IP Security
IREQ – Initiator Request
KI_{AC} – Chave privada da AC
KU_{AC} – Chave pública da AC
L-Cert – Serviço Local de Certificação
L-IDS – Serviço Local de Detecção de Intrusão
L-SPM – Serviço Local de Gerência da Política de Segurança
LAN – Local Area Network
MAC (1) – Media Access Control
MAC (2) – Message Authentication Code
MAE – Manet Authentication Extension
MAIDS – Mobile Agent Intrusion Detection System
Manet – Mobile Ad Hoc Networks
MIB – Management Information Base
MID – Multiple Interface Declaration

MPR – Multipoint Relay
MS – MPR-Selector Set
NIST – National Institute of Standards and Technology
OLSR – Optimized Link State Routing Protocol
PAN – Personal Area Network
PCA – Principal Component Analysis
PDA – Personal Digital Assistant.
PGP – Pretty Good Privacy
PKIX – Public *Key Infrastructure* (PKI) baseada no X.509.
RF – Rádio Freqüência
RFC – Request for Comments
RREQ – Route Request
RREP – Route Reply
RREP-ACK – Route Reply Acknowledgment
RSA – Rivest-Shamir-Adelman
SAODV – Secure AODV
SPARTA – Security Policy Adaptation Reinforced Through Agents
SEAD (*Secure Efficient Ad hoc Distance vector*)
SNMP – Simple Network Management Protocol
SPR – Secure Routing Protocol
SSL – Secure Sockets Layer
TBRPF – Topology Dissemination Based on Reverse-Path Forwarding
TC – Topology Control
TCP – Transmission Control Protocol
TLS – Transport Layer Security
TTL – Time To Live
UDP – User Datagram Protocol
ZRP – Zone Routing Protocol

1. INTRODUÇÃO

Redes móveis *ad hoc* (Manet[1]) são redes sem fio nas quais nodos móveis trocam informação sem auxílio de uma infra-estrutura de rede pré-definida [26]. Nestas redes, também conhecidas como redes espontâneas, os nodos se comunicam diretamente, uns com os outros, em uma arquitetura de comunicação ponto-a-ponto. Como não há infra-estrutura assumida, os serviços de roteamento são estabelecidos de maneira cooperativa e cada nodo participante da rede atua como um possível roteador para os outros. Assim, quando um nodo necessita se comunicar com outro que não esteja dentro do alcance de seu enlace, ele encaminha seus pacotes através de um nodo vizinho que esteja mais próximo do destinatário, que por sua vez encaminhará o pacote adiante. Portanto, as Manets são redes móveis multi-salto onde a conectividade entre os nodos é assegurada através deste roteamento colaborativo [24,100].

Uma Manet é essencialmente constituída de nodos móveis com uma ou mais interfaces de rede sem fio. De uma maneira geral, enlaces sem fio continuarão a ter uma capacidade consideravelmente menor que os enlaces cabeados. Isso se deve não apenas a diferenças e limitações em termos de vazão nominal de uma interface (i.e. a taxa de transferência máxima do enlace de rádio é, em geral, menor que a taxa nominal em enlaces cabeados), mas também a outros fatores próprios a redes que utilizam transmissão sem fio, tais como: efeitos de múltiplo acesso, desvanecimento (*fading*), ruídos e interferências decorrentes de fontes eletromagnéticas exógenas ao sistema, entre outros. Além disso, os nodos móveis devem ser alimentados por fontes de energia portáteis (baterias) que se exaurem com o tempo. Assim, critérios de projeto importantes para otimização dos recursos e serviços projetados para Manet são o uso eficiente da banda e da energia disponíveis.

Em uma Manet, os nodos podem, continuamente e a qualquer tempo, aparecer, desaparecer ou mover-se dentro da rede. Como resultado, a adesão dos nodos com a Manet são construídas dinamicamente e a topologia da rede está sujeita a mudanças freqüentes e imprevisíveis. Essa característica relacionada com a mobilidade dos nodos, aliada à confiabilidade e largura de banda limitadas dos enlaces sem fio, faz com que a disponibilidade de um nodo específico não possa ser assegurada. Desse modo, os serviços em uma Manet não podem ser concentrados em entidades centralizadas. Ao contrário e a exemplo dos serviços de

[1] Do inglês, *Mobile Ad hoc NETwork*.

roteamento, os serviços em uma Manet devem ser providos de maneira distribuída e auto-organizada, através de colaboração entre os nodos da rede. Essa colaboração normalmente faz uso das redundâncias naturais resultantes do modelo de comunicação, o que proporciona, de certo modo, uma compensação pela ausência de confiabilidade acerca da disponibilidade dos nodos individualmente.

Pela caracterização de Manet apresentada acima, identificam-se dois serviços básicos, necessários à formação destas redes: roteamento [25,58,83,88] e autoconfiguração [80,81,87]. O serviço de roteamento está relacionado com a natureza multi-salto das Manets. Assim, o projeto do protocolo de roteamento deve levar em consideração as constantes mudanças na topologia da rede em função da mobilidade dos nodos. Já o serviço de autoconfiguração está relacionado com a associação dos nodos à rede, permitindo uma implantação rápida e com pouca ou nenhuma intervenção dos usuários.

Neste trabalho, é apresentado um novo modelo de segurança projetado para atender os requisitos específicos de ambientes de rede *ad hoc*. Tem-se por objetivo definir e desenvolver um conjunto integrado de serviços de segurança que sejam completamente distribuídos e possam ser providos através da colaboração entre nodos de uma Manet. Além disso, esse conjunto de serviços deve ser flexível a ponto de permitir a definição de níveis diferenciados de segurança para contextos de aplicações de Manet distintos. O modelo proposto é diretamente aplicado e desenvolvido para prover segurança aos serviços essenciais de uma Manet, isto é, roteamento e autoconfiguração.

Segurança em redes ad hoc é um tema ainda recente na literatura técnica especializada, embora muitos trabalhos tenham sido publicados, nos últimos anos, sobre o assunto. Em sua maioria, essas iniciativas consistem em propostas isoladas que tratam o problema da segurança pontualmente, tendo como foco a provisão de alternativas de segurança que se aplicam a um protocolo específico (e.g. um protocolo de roteamento) [28,41,48,49,84] ou a um contexto de utilização particular [7,36,91]. Este trabalho difere-se dessas iniciativas pela proposição de uma solução de segurança que seja aplicável ao contexto *ad hoc* como um todo, antes de ser direcionada para prover segurança a um protocolo específico, e de modo a prover níveis de segurança que sejam adaptáveis aos requisitos da aplicação da Manet, possibilitando sua utilização em contextos com políticas de segurança distintas.

1.1. APLICAÇÕES DE MANETS

Existem diversas demandas atuais e futuras para a tecnologia de redes *ad hoc* [26]. A área emergente da computação móvel e nômade, atualmente com ênfase na operação via *Mobile IP*[2], está gradualmente evoluindo e começa a requerer tecnologias de rede altamente adaptáveis que permitam o gerenciamento de *clusters* multi-saltos de redes *ad hoc* que operem autonomamente ou conectadas em um ou mais pontos à Internet. Em especial, o uso de tecnologias de Manet está relacionado com a formação espontânea de redes. De fato, a noção de auto-organização faz das redes *ad hoc* uma alternativa flexível para a formação de redes, permitindo que redes móveis sejam rapidamente estabelecidas sem a necessidade de implantação de infra-estrutura prévia. Dentro deste contexto, existe uma multiplicidade de cenários de uso das Manets em aplicações comerciais, industriais, acadêmicas, governamentais ou militares, dentre as quais pode-se citar:

- Comunicação inter-grupo e trabalho cooperativo: formação dinâmica de grupos colaborativos de trabalho, em ambientes empresariais, acadêmicos e comerciais, entre outros.
- Redes de área pessoal (*Personal Area Network – PAN*): estabelecimento de comunicação em rede para ambientes de dimensão reduzida através de comunicação máquina-a-máquina, eliminando ou reduzindo a necessidade de instalação de dispositivos para ligação e inter-ligação em rede.
- Intervenções em sítios sem infra-estrutura ou cuja infra-estrutura tenha sido destruída: aplicação em cenários onde se requeira que o estabelecimento rápido de comunicações através de redes dinâmicas e com sobrevivência, como por exemplo em operações de resgate em sítios de acidentes ou atentados, em incêndios, em desabamentos, em procedimentos de manutenção em sítios remotos, entre outros.
- Redes de sensores: formação de redes entre diversos sensores, que eventualmente se encontram em movimento, para troca e processamento de informações relacionadas com as medidas que estão sendo realizadas.
- Redes em movimento: redes constituídas por sistemas em movimento, tais como aviões, carros em uma estrada ou tropas em um campo de batalha.

[2] O charter do grupo de trabalho MobileIP do IETF pode ser encontrado em: http://www.ietf.org/html.charters/mobileip-charter.html

Adicionalmente, uma outra área que têm oferecido destaque para as tecnologias de Manet é a comunicação pervasiva e o estabelecimento de redes com acesso ubíquo. Assim, redes móveis *ad hoc* baseadas em malhas podem ser operadas como alternativa ou complemento robusto e barato a redes móveis celulares.

1.2. VULNERABILIDADES DAS REDES AD HOC

Muitas das vulnerabilidades existentes em arquiteturas de redes tradicionais são também possíveis em Manets. Entretanto, algumas das características especiais das Manets enfatizam tais vulnerabilidades, ao possibilitar novas maneiras de explorá-las. Além disso, as Manets têm vulnerabilidades que lhe são próprias e que não estão presentes em outras arquiteturas de rede [66,119]. Entre as características especiais das Manets que enfatizam vulnerabilidades já conhecidas nas redes tradicionais ou que implicam em novas vulnerabilidades específicas do contexto *ad hoc*, destacam-se:

- a natureza sem fio do serviço de enlace - os nodos são capazes de monitorar a utilização da rede por nodos próximos que estejam dentro do alcance de seu receptor[3];
- o modelo de comunicação descentralizado/ponto-a-ponto - os nodos são capazes de se comunicarem diretamente, uns com os outros;
- a mobilidade - a topologia de rede muda dinamicamente;
- o modelo colaborativo de comunicação - os nodos dependem uns dos outros para estabelecimento e manutenção da conectividade na rede; e
- o uso freqüente de fontes de energia que se extinguem com o uso - os nodos móveis usam fontes de energia portáteis.

Essas características tornam as Manets mais vulneráveis que as redes cabeadas a um largo espectro de ataques, tais como escuta passiva, personificação ou *spoofing* (uma entidade assume a identidade de outra) e negação de serviço; pois um adversário pode prontamente explorá-las para:

[3] Neste trabalho, utiliza-se a definição de Manet dada na RFC 2501 [26]. Assim, o acesso ao serviço de enlace de dados é assumido para cada nodo que participa da Manet. Não se elabora sobre a segurança do nível de enlace de dados, tanto no que diz respeito à análise de vulnerabilidades quanto à definição de serviços de segurança. Neste sentido, caso o serviço de enlace utilize algum tipo de autenticação ou outra proteção criptográfica (e.g. protocolo WEP em redes IEEE 802.11, por exemplo), assume-se que um nodo comprometido ou malicioso seja capaz de autenticar-se e de cifrar/decifrar quadros de enlace de dados e, portanto, seja também capaz de monitorar as comunicações de seus vizinhos.

- escutar promiscuamente transmissões vindas de nodos próximos;
- comunicar-se diretamente com qualquer nodo que esteja dentro de seu alcance de transmissão;
- mover-se para coletar informações sobre a atividade de outros nodos mais distantes ou para escapar da monitoração de nodos próximos;
- praticar a não-colaboração (e.g. para economizar sua própria bateria ou para provocar disfunções no encaminhamento de pacotes na rede); e
- provocar a realização de atividades desnecessárias com objetivo de acelerar a exaustão das fontes de alimentação de outros nodos.

Além disso, em redes tradicionais, os serviços tais como roteamento e autoconfiguração estão delegados a entidades que são projetadas para fins específicos de segurança (e.g. roteadores e servidores de autoconfiguração). Essas entidades executam um conjunto controlado de funcionalidades e possuem um posicionamento especial na topologia da rede, que lhes oferece proteção cuidadosa. Assim, estas entidades apresentam um conjunto reduzido de vulnerabilidades, devido à inexistência de funções genéricas, à possibilidade de desativação de funcionalidades desnecessárias e à facilidade de se ativar proteções, lógicas ou mesmo físicas, relacionadas com o seu posicionamento dentro da arquitetura da rede, geralmente em pontos de concentração/centralização dentro de partes controladas da rede. Enquanto isso, em Manets, os serviços básicos, assim como os demais serviços de rede, são providos de forma descentralizada e com participação potencial de todos os nodos da rede. Esses nodos são implementados, muitas vezes, em equipamentos de computação com hardware e software genéricos que estão sujeitos a uma série de vulnerabilidades relacionadas ao seu sistema operacional, a defeitos de software (*bugs*), portas de fundos (*backdoors*), vírus, entre outras. Além disso, um nodo de uma Manet sem proteção física adequada está sujeito a ser capturado [71].

Desse modo, não raramente haverá entidades com mau funcionamento ou comprometidas na rede. Em se considerando a existência de entidades incorretas, um nodo que esteja realizando ataques contra a Manet pode ainda mover-se, seja para realizar ataques em outras partes da rede ou para escapar da monitoração de seus vizinhos. Essa característica dificulta a detecção dos ataques e a identificação do(s) nodo(s) incorreto(s) pelos demais nodos que permanecem corretos na rede.

1.3. REQUISITOS PARA A SOLUÇÃO DE SEGURANÇA EM REDES AD HOC

A solução de segurança considerada neste trabalho envolve a definição de um conjunto integrado de serviços de segurança para Manets, que provêm proteção aos demais serviços da rede. Nesse sentido, dois tipos de requisitos são considerados nesta seção. Primeiramente, são apresentados os requisitos que devem ser atendidos, de uma maneira geral, por todos os serviços em rede de uma Manet, inclusive pelos serviços de segurança que forem definidos para operação em rede. Em seguida, são apresentados os requisitos de segurança propriamente ditos, que permitem identificar e projetar os diversos serviços de segurança considerados em nosso modelo.

Uma análise das características próprias às Manets permite estabelecer um conjunto de requisitos básicos a serem considerados no projeto de qualquer serviço voltado para esse tipo de rede. A seguir, são apresentadas as principais características das Manets e os requerimentos sobre os serviços em rede que delas decorrem:

- Ausência de pontos de concentração e de garantia da disponibilidade dos nodos individualmente: os serviços de uma Manet devem seguir uma abordagem distribuída.
- Mobilidade e dinâmica da topologia de rede: a distribuição deve se dar de forma auto-organizada e colaborativa, de modo a evitar interrupções dos serviços quando ocorrem alterações na conectividade dos nodos e a aproveitar as redundâncias do modelo de comunicação para otimizar a disponibilidade do serviço.
- Limitações da banda passante e da alimentação de energia: os serviços de não devem gerar *overhead* excessivo na rede, de modo que os serviços devam ser prestados, sempre que possível, na vizinhança local, evitando o encaminhamento e a retransmissão de mensagens.

De uma maneira geral, os requisitos apresentados acima devem ser atendidos também quando do projeto de serviços de segurança. Vale ressaltar que, com exceção de alguns serviços de segurança que executam isoladamente no *host* local, a maior parte das técnicas de segurança e proteções usadas em redes tradicionais não são adaptadas para as redes *ad hoc* [52,66,119]. Em arquiteturas convencionais, serviços como controle de acesso, autenticação, autorização, monitoração e gerenciamento da segurança estão relacionados com dispositivos claramente definidos, tais como servidores de autenticação ou sistemas *firewall*. Esses componentes não podem existir na forma de dispositivos únicos nas Manets. Desse modo, os

serviços de segurança nessas redes devem seguir uma abordagem de distribuição através da colaboração com auto-organização. Além disso, sempre que possível esta colaboração deve ser completamente localizada, restringindo o *overhead* de comunicação e de processamento à vizinhança dos nodos envolvidos.

Diversas abordagens vêm sendo apresentadas para definição de requisitos de segurança em Manet [7,18,28,36,41,48,49,52,66,84,91,119]. Neste trabalho, considera-se a combinação de dois requisitos básicos: a diferenciação de nodos confiáveis e não confiáveis em uma Manet, e a identificação e conseqüente isolação de nodos comprometidos ou mal-comportados.

A diferenciação entre nodos se dá pela definição e aplicação de um modelo de confiança, que define as condições de adesão dos nodos à rede. Esta adesão, assim definida como uma relação de confiança mútua entre os nodos, é imposta aos nodos como uma primeira linha de proteção para os serviços colaborativos, requerendo-se que os nodos estejam previamente associados à rede e restringindo-se à colaboração entre os nodos que são membros da rede. Neste cenário, as trocas de informações de controle e o encaminhamento de pacotes se dão apenas no âmbito do conjunto de nodos que confiam mutuamente entre si. O requisito de uma adesão prévia explícita à rede justifica-se, em especial, pela natureza promíscua das comunicações sem fio, permitindo que qualquer dispositivo configurado com uma interface sem fio possa se comunicar na rede. Este acesso facilitado, característico das redes sem fio, é ainda mais crítico em cenários de Manet, devido à natureza espontânea dessas redes. Uma vez estabelecida a adesão, um nodo deve ser capaz de provar para os outros membros que ele aderiu à rede, assim como deve ser possível aos demais nodos da rede a constatação e verificação da existência da adesão.

Outro requisito fundamental está ligado à existência de entidades com mau funcionamento ou comprometidas em uma Manet. Dado que a ocorrência de tais entidades não pode ser negligenciada, os serviços de segurança devem ser projetados de modo a se manterem robustos mesmo na presença de nodos comprometidos ou mal-comportados. Nesse sentido, admite-se que a degradação de desempenho decorrente da existência de nodos que interagem incorretamente com a rede seja apenas temporária, devendo os nodos incorretos serem identificados e isolados dos serviços colaborativos antes que a robustez do serviço esteja comprometida.

Finalmente, como já descrito anteriormente, existem diversos contextos de aplicação para as Manets. Seguramente, contextos diferentes devem requerer níveis de segurança distintos. Assim, uma Manet estabelecida para realização de trabalho cooperativo em uma

sala de aula exige níveis de segurança diferentes do requeridos em uma Manet estabelecida para prover serviços de comunicação e informação para uma operação de resgate em local onde tenha ocorrido um desastre. Do mesmo modo, os níveis de segurança seriam ainda mais severos em uma Manet estabelecida entre os combatentes de uma tropa que se encontre em movimento no campo de batalha. De uma maneira geral, esses requisitos podem ser expressos em termos de uma política de segurança que especifica os níveis de segurança exigidos em cada caso. É intenção deste trabalho, portanto, fazer com que os serviços de segurança possam ser prontamente ajustados em conformidade com as definições da política de segurança para cada contexto de aplicação específico.

1.4. MODELO DE SEGURANÇA PARA REDES AD HOC

Este trabalho consiste na definição de um modelo de segurança que seja capaz de amenizar ou eliminar as vulnerabilidades de uma Manet, sejam essas vulnerabilidades comuns a outros tipos de rede, sejam vulnerabilidades próprias do contexto Manet. Para isso, o modelo de segurança proposto define um conjunto de serviços de segurança integrados, que são providos em conformidade com os requisitos impostos pela arquitetura Manet, discutidos na seção anterior. Desse modo, a **primeira contribuição** deste trabalho consiste na proposição de um modelo de cooperação compatível com a arquitetura *peer-to-peer* que mantenha a distribuição e a auto-organização como características fundamentais para o projeto e desenvolvimento de serviços em Manet. Assim, todos os serviços de segurança discutidos são projetados de acordo com esse modelo, eliminando completamente a necessidade de entidades centralizadas, mesmo em fases de iniciação da rede.

A **segunda contribuição** deste trabalho consiste na realização de uma análise de vulnerabilidades que resulta na identificação de requisitos de segurança para o contexto Manet e na definição de um conjunto de serviços de segurança a serem desenvolvidos, bem como da interação entre esses serviços. Em especial, no modelo de segurança apresentado neste trabalho, propõem-se mecanismos de interação entre serviços de segurança preventiva e corretiva. Enquanto os serviços de segurança preventiva evitam que ataques possam ocorrer ou que tentativa de ataques sejam bem sucedidas, os serviços de segurança corretiva asseguram a anulação dos efeitos indesejáveis de ataques pela eliminação de nodos previamente confiáveis que tenham sido comprometidos ou que passem a se comportar incorretamente.

Um serviço de autenticação baseado em um serviço de certificação distribuída [93] provê uma solução básica de segurança preventiva, permitindo diferenciar nodos confiáveis e não confiáveis, enquanto um sistema de detecção de intrusão (IDS[4]) distribuído [95] provê a solução de segurança corretiva, ao realimentar o serviço de certificação com informações sobre nodos mal comportados, permitindo o isolamento destes nodos como principal contra-medida. Essa interação entre serviços de segurança preventiva (e.g. certificação) e corretiva (e.g. IDS) é a característica mais relevante deste trabalho [96,97], uma vez que em grande parte dos trabalhos relacionados publicados [28,41,48,49,52,66,84,119] tem-se por objetivo a proposição de um serviço de segurança específico, normalmente de natureza preventiva, e fundamentado em técnicas de autenticação criptográfica.

O modelo de segurança proposto é ainda diretamente aplicado para prover segurança aos protocolos de roteamento [25,58,83,88] e de autoconfiguração [80,81,87], conforme apresentado na Figura 1-1. A definição de uma Extensão de Autenticação para Manet (MAE[5]) permite que as mensagens dos protocolos de roteamento e autoconfiguração sejam utilizadas sem alterações. Isso é de grande importância, tendo em vista que a fase de definição dos padrões experimentais da Internet para protocolos de roteamento em Manet foi concluída recentemente pela força-tarefa Manet da Internet Engineering Task Force (IETF)[6] e os aspectos de segurança não foram considerados diretamente nestas primeiras versões. Serviços locais de certificação (L-Cert) e detecção de intrusão (L-IDS) são executados colaborativamente em todos os nodos da Manet. Para que a colaboração ocorra, basta que exista comunicação disponível entre os nodos. A auto-organização é conseqüência direta da possibilidade de se iniciar uma colaboração imediata a qualquer tempo e com quaisquer nodos. Finalmente, para manter-se a robustez da solução, é requerido que um número mínimo de nodos (limiar) concordem (noção de política de segurança) e cooperem (noção de comunicação coordenada) para que um novo nodo seja admitido na rede (serviço de certificação) ou uma acusação contra um nodo comprometido inicie uma resposta (sistema de detecção de intrusão).

[4] Do inglês, *Intrusion Detection System*.

[5] Do inglês, *Manet Authentication Extension*.

[6] A força-tarefa Manet da IETF definiu, em 2003, os protocolos AODV [88], OLSR [25], DSR [58] e TBRPF [83] como padrões experimentais da Internet para protocolos de roteamento em Manet, após quase quatro anos de trabalho sobre o tema. Para maiores detalhes, o *charter* oficial da força-tarefa Manet pode ser acessado em **http://www.ietf.org/html.charters/manet-charter.html**.

O projeto e análise desses serviços de segurança (i.e. o serviço de certificação distribuída e de autenticação como mecanismos preventivos, e o sistema de detecção de intrusão distribuído como mecanismo corretivo) e de sua interação no contexto da segurança dos protocolos de roteamento e de autoconfiguração consistem a **contribuição final** deste trabalho.

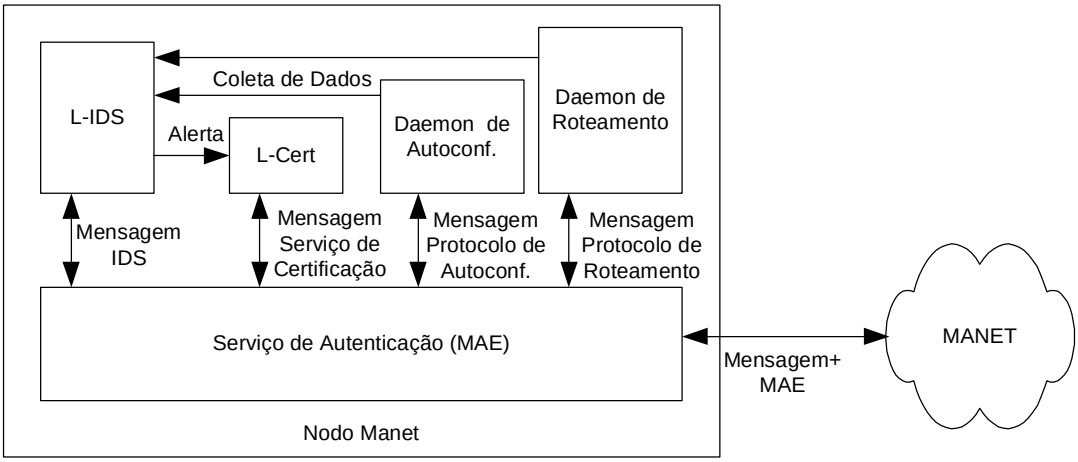


Figura 1-1 – Modelo de Segurança Aplicado aos Serviços de Roteamento e Autoconfiguração

No que diz respeito ao projeto do serviço de certificação, as funções de certificação, tipicamente executadas por autoridades certificadoras (AC) em arquiteturas convencionais, estão distribuídas entre os membros da Manet [66,119]. Esses nodos devem cooperar, em um número mínimo (limiar), para prover de forma colaborativa os serviços de certificação. Tais serviços são estabelecidos com uso de criptografia de limiar, proposta originalmente por Shamir [102]. Essa técnica foi originalmente usada para definir um esquema de certificação distribuída adaptado para o contexto das Manets por L. Zhou *et al.* [119,120], que foi posteriormente completado por Kong et al. [66]. Neste trabalho faz-se uma extensão, com correções, de [66], com as seguintes contribuições [93]:

- proposição de um mecanismo de iniciação do serviço, evitando a necessidade de um nodo centralizado (*dealer*) na fase de constituição da rede existente em [66,119]. Nosso mecanismo foi desenvolvido simultaneamente com [62] mas tem princípios de concepção e operação diferenciados.
- abordagem baseada em política de segurança, com definição de critérios objetivos para operação do serviço de certificação, corrigindo vulnerabilidades ao ataque de Sybil [32], e permitindo uma adequação do serviço a contextos com requisitos de segurança diferentes.

- projeto completo dos protocolos de certificação, incluindo a definição de sintaxe, seqüência e semântica de mensagens, a especificação de mecanismos de *cache* de certificados válidos e da lista de certificados revogados (CRL[7]) locais, assim como a implementação completa do protocolo para validação.

O serviço de autenticação é provido pela Extensão de Autenticação para Manet (MAE) [93], que é projetada para prover autenticação em serviços orientados a mensagens[8], tais como os protocolos de roteamento e autoconfiguração. A contribuição deste trabalho consiste em definir tal extensão de maneira independente do protocolo de roteamento ou autoconfiguração adotado. Esta é uma abordagem alternativa a trabalhos recentes que propõem a segurança dos protocolos AODV [28,41] e DSR [48,49] através de modificações de sintaxe ou mesmo nos mecanismos dos protocolos em questão.

Finalmente, no que diz respeito ao sistema de detecção de intrusão, propõe-se uma nova arquitetura para o IDS que seja adaptada aos requisitos do contexto Manet, com as seguintes característica [94,95]:

- distribuição completa do processo de detecção de intrusão, possibilitando que a detecção ocorra de forma colaborativa em qualquer das fases de detecção (coleta de dados, análise de dados e gerenciamento de alertas);
- auto-organização, pelo uso de uma plataforma de agentes;
- arquitetura modular, permitindo a utilização simultânea de técnicas de detecção de intrusão por uso incorreto (*misuse*) e por modelagem de comportamento[9], bem como a utilização de diferentes fontes de dados de auditoria no processo de detecção.

Ainda que detecção de intrusão seja um assunto corrente na literatura de segurança de redes, vale ressaltar que poucos trabalhos foram publicados a respeito de detecção de intrusão no contexto Manet, a exemplo de [42,51,73,79,111,117].

O IDS projetado foi implementado e avaliado para detecção de ataques contra o nível de rede (protocolo de roteamento) e de aplicação (ataque de cadeia de sessões *telnet* ou ataque

[7] Do inglês, *Certificate Revocation List.*

[8] Para serviços orientados a conexão, pode-se usar os protocolos SSL/TLS.

[9] Existem duas vertentes clássicas em detecção de intrusão [29]: a detecção de uso incorreto (*misuse*), onde os ataques são identificados através do reconhecimento de padrões previamente relacionados com os ataques e formalmente definidos na forma de uma assinatura, e por comportamento, quando são reconhecidas condições de operação que diferem significativamente das condições de operação normal previamente modeladas.

stepping stone), utilizando informações provenientes da MIB (*Management Information Base*) e da análise de pacotes de rede selecionados.

1.5. ORGANIZAÇÃO DO TRABALHO

Este trabalho está assim organizado: O Capítulo 2 é dedicado à apresentação do estado da arte da segurança em Manets, com ênfase principal para as contribuições relacionadas à certificação, detecção de intrusão e à segurança de protocolos de roteamento e autoconfiguração. No Capítulo 3, é apresentado o modelo de segurança proposto, juntamente com a análise de vulnerabilidade e a definição de requisitos de segurança que embasam a concepção do modelo. O Capítulo 4 contém uma descrição detalhada dos serviços de segurança preventiva, isto é, os serviços de certificação e de autenticação em Manets, este último com uma análise de sua aplicação aos quatro protocolos de roteamento para Manet definidos como padrões experimentais da Internet (i.e. AODV, OLSR, TBRPF e DSR) e ao protocolo de autoconfiguração DCDP (*Dynamic Configuration Distribution Protocol*), estudado no contexto deste trabalho. O Capítulo 5 traz a apresentação do Sistema de Detecção de Intrusão, assim como do projeto de assinatura de ataques contra os nodos de uma Manet, com foco especial em ataques realizados contra o protocolo de roteamento. Esse Capítulo apresenta ainda o mecanismo de resposta à intrusão, provendo o serviço de segurança corretiva. O Capítulo 6 mostra os experimentos e resultados realizados com o objetivo de validar o modelo proposto, consistindo tanto de experimentos realizados a partir de implementações reais dos serviços de segurança, quanto de experimentos com simulação usados essencialmente para validar os efeitos da mobilidade dos nodos no desempenho dos serviços projetados. Finalmente, o Capítulo 7 conclui esta tese com considerações finais e trabalhos futuros.

2. SEGURANÇA EM REDES MOVEIS AD HOC: ESTADO DA ARTE

A segurança em Manet é um assunto bastante discutido na literatura especializada recente, por tratar-se de contexto e problemática novos, com muitos pontos ainda em aberto. De uma maneira geral, os trabalhos publicados e os esforços correntes para definição de técnicas de segurança para essas redes podem ser divididos em dois grupos: a definição de modelos de confiança [7,17,18,36,52,62,66,71,91,119], permitindo a distinção de nodos confiáveis e não confiáveis na rede, e a segurança de protocolos relacionados com os serviços fundamentais desse tipo de rede, com especial destaque para a segurança do protocolo de roteamento [28,41,48,49,84]. Por outro lado, até a presente data, existe um número reduzido de publicações acerca de técnicas de segurança corretiva, tais como sistemas de detecção de intrusão especificamente projetados para Manet, apesar da apresentação recente de alguns poucos trabalhos sobre este importante tópico [42, 51,73,79,111,117].

Neste capítulo, discutem-se os principais resultados apresentados na literatura acerca da segurança em Manet, abordando os tópicos relacionados com a definição de modelos de confiança, com a segurança de protocolos de roteamento e autoconfiguração, e com o projeto de sistemas de detecção de intrusão nesses ambientes de rede. A intenção deste capítulo consiste em, ao mesmo tempo, retratar o estado da arte de segurança em redes móveis *ad hoc* e posicionar as contribuições deste trabalho em relação a outros trabalhos relacionados.

2.1. MODELOS DE CONFIANÇA E SERVIÇOS DE CERTIFICAÇÃO PARA MANET

De uma maneira geral, as principais propostas de segurança para protocolos e serviços em redes móveis *ad hoc* utilizam uma noção de separação lógica dos nodos da rede em confiáveis e não confiáveis. Desse modo, a utilização dos serviços e protocolos com segurança deve, em geral, ser precedida pelo estabelecimento de uma relação de confiança entre os nodos da rede. Além disso, uma vez acordada a confiança, isto é, quando os nodos concordam mutuamente em confiarem entre si, esta precisa ser estabelecida formalmente de maneira verificável. Isso pode ocorrer pela distribuição de fichas de filiação (*token*) [116], pelo estabelecimento de associações de segurança (compartilhamento de chaves criptográficas) [48,49,84], pelo uso de certificados digitais em esquemas similares a uma

infra-estrutura de chaves públicas (ICP) [28,41], ou por outra forma qualquer de expressão verificável da relação estabelecida.

As arquiteturas tradicionais de certificação incluem Kerberos [65], padrão X.509 [46] e PKIX [4]. Nestes padrões, duas entidades se autenticam através de uma autoridade certificadora. Esse tipo de arquitetura, entretanto, só funciona adequadamente em redes com infra-estrutura definida. Em redes *ad hoc*, o funcionamento não é satisfatório devido aos seguintes aspectos:

- O custo elevado para manter servidores centralizados em uma rede *ad hoc* de grande escala;
- Os servidores AC de uma rede *ad hoc* são vulneráveis a ataques;
- A mobilidade dos nodos gera a necessidade de autenticações constantes, o que gera problemas de escalabilidade e congestionamento em torno dos servidores AC;
- Uma comunicação multi-hop em um canal sem fio propenso a erros expõe a transmissão dos dados a altas taxas de erros.

Algumas variações, como ACs hierarquizadas e delegações de ACs [71] podem amenizar, mas não solucionam o problema de robustez do sistema e disponibilidade dos serviços dentro da rede.

O projeto de serviços de certificação em Manet deve seguir uma abordagem de distribuição, auto-organização e localização. Duas iniciativas se destacam como alternativas para esse tipo de serviço. J. Hubaux *et al.* [17,18,52] apresentam uma proposta de modelo onde as relações de confiança são estabelecidas entre pares de nodos (*peer-to-peer*), em um esquema similar ao sistema PGP (*Pretty Good Privacy*) [121]). Cada nodo gera localmente um par de chaves pública/privada. A chave privada é, então, usada para assinar certificados para outros nodos confiáveis, enquanto a chave pública é usada na verificação destes certificados. Um nodo deve ter diferentes certificados ligando sua chave pública com sua identidade, cada um tendo sido assinado por um outro nodo da Manet que confie nele. A distribuição do serviço de certificação é obtida pelo uso de repositórios de certificados mantidos localmente, que armazenam os certificados dos nodos que se encontram nas proximidades. A proposta apresenta ainda os algoritmos para construção e atualização desses repositórios. O ponto forte desta abordagem consiste na adoção de um modelo de confiança claramente *peer-to-peer*, que não requer o uso de nenhuma entidade externa nem mesmo para iniciação do serviço. Desse modo, o esquema é naturalmente auto-organizado.

A validação dos certificados é feita estabelecendo-se múltiplas cadeias de certificação (i.e. caminho de certificação de uma chave pública para outra) a partir da chave pública do

nodo executando a validação para a chave pública que está sendo validada. Para avaliar quantitativamente a confiabilidade do processo de validação, são projetadas e utilizadas métricas de autenticação atribuídas a cada cadeia de certificados.

Essa abordagem, no entanto, possui duas vulnerabilidades importantes, que dificultam seu uso em cenários mais genéricos e com requisitos de segurança mais restritos. Primeiramente, o uso de métricas de autenticação é útil para lidar com nodos mal-comportados que emitem certificados falsos. Entretanto, esta técnica não é bem sucedida para tratar a exposição de certificados válidos a nodos comprometidos, pois as métricas de certificação estão projetadas para lidar com a emissão errônea de certificados, mas não com certificados emitidos corretamente. Finalmente, essa técnica não é resistente a ataques de Sybil [32], onde um nodo forja múltiplas identidades para construir cadeias de certificação fictícias e distribui estes certificados aos nodos próximos, aumentando, assim, os valores aferidos para as métricas de autenticação no processo de validação. Desse modo, um nodo pode ganhar confiança de toda a rede, tendo conquistado a confiança de apenas um nodo confiável ou mesmo comprometendo um único nodo legítimo da rede.

A alternativa existente para a proposta de P. Hubaux consiste na definição de uma autoridade de certificação distribuída (ACD) [66,71,119]. Nesta abordagem, a chave privada da AC (KI_{AC}) é usada para assinar certificados para todos os nodos na Manet. Um certificado assinado com KI_{AC} pode ser prontamente verificado pelo uso da chave pública da AC (KU_{AC}), que é notória para todos os nodos da rede. A distribuição das facilidades e serviços providos pela AC é conseguida pelo compartilhamento da chave privada entre os nodos da rede pelo uso de técnicas de criptografia de limiar [102]. Cada nodo da Manet (v_i) mantém uma parte da chave privada ($KI_{AC,i}$) e qualquer coalizão de K (uma constante do sistema, usualmente definida em termos do número médio de vizinhos na rede) portadores de partes da chave privada podem coletivamente exercer a função de AC. A KI_{AC}, entretanto, não pode ser recuperada por nenhum destes nodos. A revogação de certificados é realizada pela emissão de contra-certificados, que também precisam ser assinados com KI_{AC}. Desse modo, a lista de certificados revogados pode ser mantida localmente por cada nodo da Manet, que fazem *cache* de todos contra-certificados emitidos. Obviamente, K portadores de partes da chave privada precisam concordar mutuamente para que um contra-certificado seja emitido contra um nodo qualquer. Essa dualidade equilibra a emissão e a revogação de certificados, estando ambos estes processos condicionados ao tamanho da coalizão (K).

A auto-organização é obtida pela definição de um protocolo para o estabelecimento dinâmico de coalizões de K portadores de partes da chave privada. Essas coalizões são

formadas para prestação de três serviços básicos: (1) assinatura de certificados, usada na emissão, renovação e revogação de certificados; (2) emissão de novas partes da chave privada, usadas na iniciação de nodos que chegam na rede; e (3) na atualização das partes da chave privada, que deve ocorrer periodicamente para evitar que um adversário possa progressivamente comprometer nodos distintos da Manet até quebrar o sistema depois de comprometer o K-ésimo nodo. Este último serviço está diretamente relacionado com uma noção de tolerância à intrusão, no sentido de que a segurança total do sistema está condicionada a um mecanismo de detecção de intrusão que seja capaz de rastrear e eliminar (por meio de revogação de certificados) nodos mal-comportados da rede, antes de um mesmo adversário, ou um grupo de adversários que cooperem entre si, comprometer *K* nodos distintos.

Tal serviço de certificação com base em uma ACD foi originalmente proposto por L. Zhou and Z. J. Haas [119]. Nesta proposta original, os portadores de partes da chave privada estão restritos a um conjunto de nodos "especiais", que devem ser previamente iniciados. Isto é, o sistema não é completamente auto-organizado, pois é necessário um distribuidor (*dealer*) centralizado que distribua *off-line* as partes da chave privada aos nodos portadores. Um esquema para atualização das partes da chave privada foi igualmente proposto. Em uma contribuição posterior, J. Kong *et al.* [66] propõem a generalização deste modelo original, permitindo que qualquer nodo detentor de um certificado válido possa participar nos serviços da ACD. Neste cenário, um nodo que não tenha uma parte da chave privada pode recebê-la de outros *K* nodos da rede que já detenham partes da chave privada. Este procedimento reduz os requisitos de iniciação (*bootstrap*) da rede, fazendo com que a distribuição centralizada de partes da chave privada seja requerida apenas para a iniciação dos primeiros *K* nodos da Manet. Finalmente, em [71], uma versão melhorada do procedimento de atualização de partes da chave privada é apresentada.

Outra diferença entre estes trabalhos diz respeito à política de emissão e revogação de certificados. Em [119], novos certificados devem ser emitidos por um procedimento *out-of-band*, normalmente executado por uma AC centralizada que também é responsável pela distribuição das partes da chave privada. Enquanto esta AC centralizada não é necessária para renovação e revogação de certificados, ela deve estar sempre presente para prover a emissão de certificados para nodos sem certificados válidos. Em [66,71], a emissão de novos certificados é realizada pela ACD (i.e. qualquer coalizão de *K* portadores de partes da chave privada). Entretanto, enquanto em [66] a política para emissão de certificados para nodos sem certificados válidos não é mencionada, em [71], a requisição para emissão de certificados é

sempre assinada, ao menos que exista alguma restrição explícita contra o nodo requerente (e.g. um contra-certificado). Essa política torna o sistema vulnerável a ataques de Sybil [32], pois um nodo forjando múltiplas identidades pode facilmente obter *K* certificados válidos e, conseqüentemente, *K* partes da chave privada distintas, quebrando definitivamente o sistema.

No que diz respeito a serviços de certificação, a proposta deste trabalho [93] é baseada em [66,71], com as seguintes contribuições:

- A emissão e renovação de certificados em [66,71] segue alguma regra pré-estabelecida. Neste trabalho, ao invés de se ter tal regra inflexível, propõe-se uma abordagem para a emissão e renovação de certificados de acordo com a política de segurança, que pode ser traduzida em um conjunto de parâmetros e opções configuráveis que tornam a emissão e renovação de certificados, assim como a emissão e atualização de partes da chave privada flexíveis e adaptáveis a diversos cenários de utilização de Manet. Essa abordagem pela definição de uma política de segurança permite ainda definir condições apropriadas para iniciação de novos nodos, corrigindo as vulnerabilidades relacionadas a ataques de Sybil.
- As propostas em [66,71] não especificam como certificados válidos são adquiridos e armazenados em cada nodo. Adicionalmente, a construção da CRL local é progressivamente construída acumulando-se os contra-certificados que são assinados e imediatamente distribuídos por *flooding* em toda a rede, entretanto, a sincronização de nodos novos que chegam na rede, e precisam iniciar suas CRL vazias, não é mencionada. Neste trabalho propõe-se um conjunto de mecanismos para distribuição e manutenção localizada das bases de dados locais contendo os certificados válidos (*cache* de certificados) e revogados (CRL). Esses mecanismos possuem ainda uma série de opções configuráveis que permitem a adaptação dos serviços de sincronização das bases de dados às demais políticas de roteamento e distribuição de informação na rede. Essas opções incluem a escolha entre métodos pró-ativos ou sob-demanda para a construção e manutenção da base de dados de certificados válidos e da CRL, a definição de temporizadores para manutenção de certificados válidos sem uso da *cache*, e tamanhos máximos para a *cache* de certificados válidos.
- Em [66,71] é considerado o uso de uma única ACD. Neste trabalho, estende-se este modelo para suporte a múltiplas ACD, com os respectivos mecanismos para construção dos caminhos de certificação. Essa característica é fundamental para

cenários onde se admita a junção (*merging*) de duas ou mais Manets que tenham sido iniciadas de forma independente.

- No que diz respeito ao projeto dos protocolos usados nos serviços de certificação, em [66,71] apenas é especificada a seqüência das mensagens em cada procedimento. O projeto do protocolo é completado neste trabalho, com a proposição de sintaxe e semântica precisa para as mensagens.

Finalmente, adota-se um modelo de confiança implementado na forma de um serviço de certificação distribuído [93]. Como será discutido com mais detalhe adiante neste trabalho, uma das justificativas para a escolha deste modelo consiste na possibilidade de se construir um modelo de confiança que suporte a revogação da confiabilidade para os nodos que estejam comprometidos e/ou mal-comportados. Para tanto, torna-se necessário o uso de primitivas de criptografia assimétrica.

2.2. SEGURANÇA DOS PROTOCOLOS DE ROTEAMENTO

Como mencionado no capítulo anterior, os serviços de roteamento e autoconfiguração são fundamentais na concepção da tecnologia de Manet. Nesta seção, faz-se uma revisão dos principais trabalhos acerca de segurança de protocolos para roteamento e autoconfiguração.

2.2.1. Protocolos de Roteamento

O roteamento em redes *ad hoc* é bastante diferente do roteamento utilizado em redes cabeadas. Roteamento em redes *ad hoc* depende de muitos fatores incluindo topologia, seleção de roteadores e características específicas que podem ser heurísticas na hora de encontrar o caminho melhor ou o mais rápido para a entrega dos dados. As principais características peculiares das Manets que possuem impacto direto no projeto de protocolos de roteamento são listadas a seguir:

- Escassez de recursos, tais como largura de banda e energia. Os algoritmos de roteamento devem prover uma utilização eficaz da banda disponível, permitindo a economia de energia sempre que possível.
- Enlaces simétricos e assimétricos. Um enlace é dito simétrico quando dois nodos estão dentro da área de transmissão um do outro, possuindo as mesmas características de roteamento em ambas as direções. Quando isso não ocorre, o enlace é denominado assimétrico tornando o roteamento uma difícil tarefa. A

maioria dos protocolos de roteamento em redes *ad hoc* se baseia em enlaces simétricos uma vez que o enlace assimétrico deve ser evitado.

- Padrões de mobilidade. Alguns nodos são altamente móveis enquanto outros podem ser fixos ou moverem-se vagarosamente. É difícil predizer o padrão de movimento dos nodos, além do que, o número de nodos em uma rede pode ser muito grande e a tarefa de encontrar uma rota para o destino acarretará em uma freqüente troca de informações de controle entre os nodos. Desta maneira, a alta mobilidade dos nodos pode implicar em sobrecarga na manutenção das rotas dos protocolos de roteamento, de tal forma que não sobrará banda para a transmissão de pacotes de dados.
- Escalabilidade. Um protocolo de roteamento deve ainda funcionar efetivamente desde as redes pequenas, com dezenas de nodos, até as redes de larga escala, com centenas de nodos e topologia *multi-hop*.

Devido às particularidades do ambiente de redes *ad hoc*, o grupo de trabalho Manet foi formado em 1997 pela IETF com o intuito de pesquisar e desenvolver especificações para o roteamento em Manet e introduzi-los como padrões para a Internet, viabilizando um roteamento ponto-a-ponto em um ambiente puramente móvel e sem fio.

Para julgar o mérito e o desempenho de um protocolo de roteamento, o grupo de trabalho Manet enumera algumas métricas que os protocolos devem seguir [26]. Essas métricas foram divididas em qualitativas e quantitativas e devem ser avaliadas independentes de qualquer protocolo de roteamento. A Tabela 2-1 descreve as métricas qualitativas de um protocolo de roteamento para Manet. A Tabela 2-2 mostra os pontos quantitativos [26] que devem ser observados para analisar o desempenho de um protocolo de roteamento *Manet.*

Obter a eficiência de um protocolo de roteamento de uma rede ad hoc não é uma tarefa simples. Devem-se considerar vários fatores, tais como, o tamanho da rede em número de nodos, a quantidade de vizinhos que cada nodo possui, a velocidade com que a topologia muda, a freqüência em que os nodos entram e saem do estado de inatividade, entre outros. Levando em conta esses fatores, pode se medir a eficiência de um protocolo de roteamento de uma rede ad hoc com as seguintes razões [26]:

- Bits de dados transmitidos / Bits de dados entregues: esta medida representa a eficiência dos bits de dados entregues dentro da rede. Indiretamente, essa medida fornece também a média de saltos percorridos pelos pacotes de dados.

Tabela 2-1 – Métricas qualitativas para um protocolo de roteamento para Manet	
Métrica	Descrição
Operação distribuída	Propriedade essencial para o roteamento nas redes *ad hoc*, uma vez que a centralização de informações é inviável neste contexto.
Livre de *loops*	Para que os pacotes não fiquem trafegando durante um período de tempo relativamente grande na rede, pode ser usada como solução uma variável do tipo TTL (*time to live*), entretanto uma abordagem mais estruturada é indicada.
Operação sob demanda	O algoritmo de roteamento deve ser adaptável às condições de tráfego; se isto for feito de forma inteligente, os recursos de energia e largura de banda são utilizados de forma mais eficiente.
Operação pró-ativa	Em alguns momentos, a latência adicionada pela operação sob demanda poderá ser inaceitável; se os recursos de energia e largura de banda permitirem, operações pró-ativas são desejáveis.
Segurança	Se as camadas de rede e de enlace não garantirem segurança, os protocolos de roteamento estarão vulneráveis a muitas formas de ataque; é necessário que haja mecanismos adicionais para inibir modificações nas operações dos protocolos.
Operação no período de inatividade (*sleeping mode*)	Como resultado da necessidade de conservação de energia, os nodos devem parar de transmitir e/ou receber pacotes durante períodos de inatividade, sem que isto resulte problemas significativos para o roteamento.

Tabela 2-2 - Métricas quantitativas para um protocolo de roteamento para Manet	
Métrica	Descrição
Atraso e desempenho de dados fim a fim	Dados estatísticos como variância, média e distribuição são muito importantes na avaliação da eficácia de um protocolo de roteamento.
Tempo de descobrimento da rota	Uma forma particular de medir o atraso do pacote fim a fim no que diz respeito aos algoritmos de roteamento sob demanda é o tempo requerido para estabelecer rotas quando requisitadas.
Porcentagem dos pacotes entregue fora da ordem	Medida externa para avaliar a performance do roteamento de protocolos da camada de transporte como TCP, que entregam os pacotes na ordem correta.
Eficiência	Se a eficácia do roteamento é uma medida externa na avaliação da performance, a eficiência é uma medida interna de sua efetividade. Se o tráfego de pacotes de dados e de controle deve compartilhar o mesmo meio, e a capacidade dos meios é limitada, então o tráfego excessivo dos pacotes de controle causará impacto na performance do roteamento.

- Bits de controle transmitidos / Bits de dados entregues: esta medida representa a eficiência do protocolo no uso entre os pacotes de controle sobre os pacotes de dados entregues. Vale a pena ressaltar, que essa medida não deve incluir somente os bits dos pacotes de controle de roteamento, mas também os bits inclusos no

cabeçalho dos pacotes de dados. Em outras palavras, tudo que não é pacote de dados é pacote de controle e deve ser contado no algoritmo.

- Pacotes de controle e pacotes de dados transmitidos / Pacotes de dados entregues: ao invés de medir a eficiência em número de bits do protocolo de roteamento, esta medida tenta capturar a eficiência de acesso ao canal do protocolo.

O projeto de protocolos de roteamento para Manet é um tópico de pesquisa bastante ativo nos últimos anos. Diversas propostas para a construção de protocolos de roteamento foram apresentas. A Figura 2-1 apresenta uma classificação dos principais protocolos de roteamento propostos. Essa classificação considera os seguintes critérios:

Quanto à política de descobrimento de rotas:

- Protocolos reativos: Determinam as rotas a serem usadas somente sob demanda, ou seja, somente quando é requerida uma rota é que o protocolo inicia o processo de descobrimento da rota. A principal vantagem do algoritmo reativo é que ele possibilita economia de banda e de energia. Entretanto o atraso para determinar uma rota pode ser significativamente alto.
- Protocolos pró-ativos: Propagam periódica e continuamente informações de roteamento, mantendo um conhecimento atualizado de todas as rotas, para que, quando um pacote necessitar de encaminhamento, a rota já seja conhecida e possa ser imediatamente utilizada. Possuem a vantagem de ter um atraso mínimo quando uma rota é solicitada devido à rota ser imediatamente selecionada da tabela de roteamento. Entretanto, para se manter a consistência e a topologia atualizada, há uma utilização contínua da rede para a troca de pacotes e de informações de roteamento. Essas atualizações são iniciadas por mecanismos de tempo e mesmo com uma rede em equilíbrio há uma troca constante de informações.
- Protocolos híbridos: Possuem estrutura hierárquica, em que parte das informações de roteamento é atualizada de maneira pró-ativa, sendo o processo completado por descobrimento de rotas sob demanda (reativo).

Quanto ao algoritmo de roteamento:

- Algoritmos de vetor de distância: As rotas são construídas com base nas informações de distâncias (e.g. número de saltos) entre origem e destino, mantidas por cada nodo/roteador.
- Algoritmos de estado de enlace: As rotas consideram todos os enlaces na topologia da rede para cálculo das rotas ótimas entre origem e destino.

§ Algoritmos de roteamento na origem: As rotas são estabelecidas para um par origem-destino e estão disponíveis na origem dos pacotes enviados.

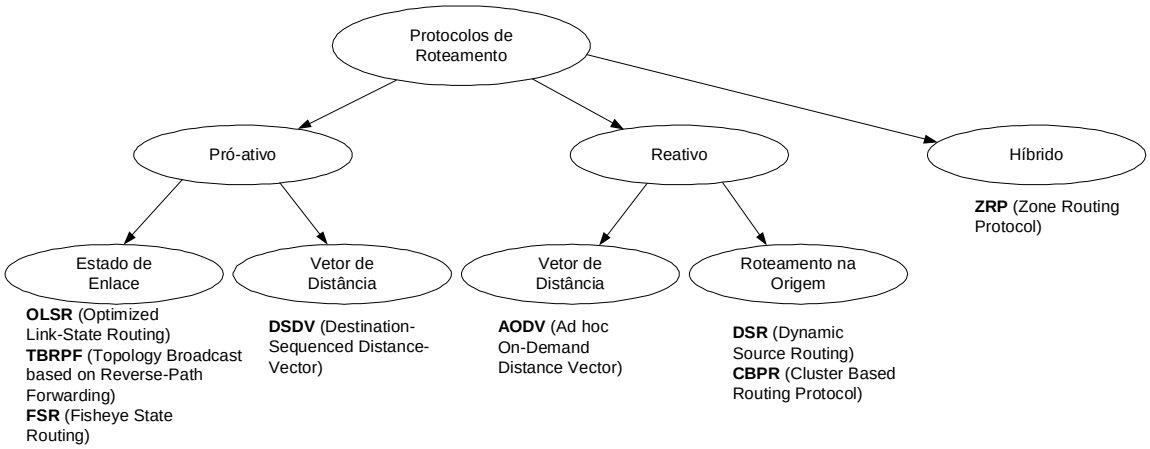


Figura 2-1 – Classificação dos Protocolos de Roteamento para Manet

Nenhum protocolo projetado possui características ótimas para todos os cenários [26,100]. Assim, os trabalhos coordenados pelo grupo de trabalho Manet identificaram recentemente um conjunto protocolos de roteamento para Manet que deverão constituir um núcleo de protocolos que provejam, com abrangência e flexibilidade, o serviço de roteamento nos diversos cenários de aplicação das Manets. Estes protocolos são: Ad Hoc on-Demand Distance Vector Routing (AODV) [88], Optimized Link State Routing Protocol (OLSR) [25], Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) [83] e Dynamic Source Routing (DSR) [58]. A seguir, é apresentada uma breve descrição destes protocolos.

2.2.1.1. Ad Hoc on-Demand Distance Vector Routing (AODV)

O AODV é um protocolo de roteamento reativo que utiliza um algoritmo de roteamento do tipo vetor de distância. De forma geral, o AODV tenta eliminar a necessidade da difusão das mensagens de roteamento, o que limita sua escalabilidade. Outro ponto importante do protocolo AODV é tentar minimizar a latência quando novas rotas são requisitadas. O AODV procura uma solução intermediária entre o roteamento reativo e o pró-ativo. No primeiro a latência é grande, já que é necessário esperar o tempo da resposta da requisição de rotas. No segundo, o volume de informações trocadas pode ser muito grande para uma rede *ad hoc* com muita mobilidade. O AODV quando comparado com os algoritmos clássicos de roteamento como vetor de distâncias e estado de enlace, apresenta uma grande redução no número de mensagens de roteamento propagadas na rede. Isto é devido à sua abordagem reativa. A forma de funcionamento do AODV é semelhante à de algoritmos tradicionais, o que pode facilitar no caso de uma possível interconexão da rede *ad hoc* a uma

rede fixa. Mesmo funcionando de forma semelhante aos algoritmos tradicionais, o AODV pode suportar tráfego *multicast* e *unicast*. Entretanto, o protocolo apresenta uma única rota para cada destino, o que pode não ser uma boa característica.

2.2.1.2. Optimized Link State Routing Protocol (OLSR)

OLSR é um protocolo de roteamento pró-ativo que utiliza um algoritmo de roteamento do tipo estado de enlace. O conceito chave desse protocolo é o uso de *multipoint relays* (MPR), que são nodos selecionados para encaminhar as mensagens de difusão no processo de inundação (*flooding*) do protocolo de roteamento. Somente os nodos selecionados como MPRs fazem essa difusão de informações na rede. O uso de MPRs combinado com a eliminação local de duplicidade é usado para minimizar o número de pacotes de controle enviados na rede. O OLSR é projetado para trabalhar em redes de larga escala, onde o tráfego é randômico e esporádico entre um conjunto de nodos específicos. Como um protocolo pró-ativo, OLSR é também adequado para os cenários onde pares de nodos que se comunicam mudam constantemente.

Os nodos que executam o OLSR usam mensagens HELLO, trocadas entre vizinhos de um salto, para detectar e atualizar o seu conjunto de vizinhos. Cada nodo, periodicamente faz uma difusão dessas mensagens anunciando informações sobre interfaces de vizinhos que são escutadas e o estado dos enlaces com cada interface. O estado pode ser “symmetric” (a comunicação bi-direcional foi verificada para este enalce), “asymmetric” (a comunicação foi verificada apenas em um sentido), MPR (o nodo vizinho anunciado foi escolhido pelo anunciante como um de seus MPR, neste caso o enlace de ser igualmente simétrico) ou “lost” (vizinho moveu-se e não está mais sendo escutado). Mensagens HELLO não são encaminhadas para outros nodos.

Cada nodo seleciona, de maneira independente, seu próprio conjunto de MPR, entre seus vizinhos com os quais ele possui um enlace simétrico. O MS deve ser computado de modo que, através dos nodos contidos neste conjunto, seja possível se atingir todos os vizinhos a dois saltos.

Para prover rotas para nodos distantes a mais de 2 saltos, cada nodo mantêm informações sobre a topologia da rede. Essa informação é adquirida através de mensagens *topology control* (TC). Os nodos que foram selecionadas como MPR por outros nodos periodicamente geram mensagens TC, anunciando a lista de todos os nodos seletores (MS). Mensagens TC são disseminadas (*flooding*) em toda a rede pelos MPRs. Um campo de

número de seqüência de mensagem (SN) é usado para evitar o processamento duplicado de mensagens. Este campo é gerado como uma seqüência de números inteiros, incrementada monotonicamente a cada mensagem gerada.

2.2.1.3. Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)

TBRPF é um protocolo pró-ativo que fornece roteamento passo a passo ao longo dos caminhos mais curtos para cada destino. Utiliza um algoritmo de roteamento do tipo estado de enlace onde cada nodo gera uma árvore de origem baseada na informação da topologia, usando uma modificação do algoritmo de Dijkstra. Para minimizar o processamento na rede, cada nodo reporta somente uma parte de sua árvore de origem para os nodos vizinhos. O protocolo usa uma combinação de diferentes atualizações periódicas para manter todos os vizinhos informados de seu envio de parte da árvore de origem. Cada nodo também tem a opção de enviar informações adicionais de topologia (árvore completa) para oferecer robustez em ambientes altamente móveis. TBRPF realiza descoberta dos vizinhos através de mensagens diferenciadas, denominadas *Hello*, que contém somente a mudança do estado dos vizinhos. Essa modificação resulta em mensagens *Hello* muito menores que as usadas em outros protocolos de estado de enlace como o OSPF. Por fim, o protocolo é constituído de dois módulos principais. O primeiro é o de "descoberta dos vizinhos" e o segundo módulo é o de "roteamento" – que realiza a descoberta da topologia da rede e computa as rotas para cada destino.

2.2.1.4. Dynamic Source Routing Protocol (DSR)

DSR é um protocolo de roteamento reativo (sob demanda), composto de dois principais mecanismos: "descobrimento de rotas" e "manutenção de rotas" que trabalham juntos. Utiliza um algoritmo de roteamento na origem que permite múltiplas rotas para um determinado destino e permite que cada remetente selecione e controle as rotas usadas para o envio de seus pacotes. Provê roteamento livre de laços, suporte a enlaces unidirecionais e uma rápida convergência quando a topologia da rede é alterada. Foi desenvolvido para redes móveis *ad hoc* de tamanho médio e desenhado para suportar altas taxas de mobilidade.

2.2.2. Segurança dos Protocolos de Roteamento

O processo de padronização dos protocolos de roteamento ainda está em curso e muitos aspectos de segurança destes protocolos não foram tratados ainda. Na prática, nenhum

desses protocolos considera, de maneira sistemática, aspectos de segurança. Não obstante, a maior parte dos trabalhos apresentados sobre segurança em Manet está relacionada com a proteção dos protocolos de roteamento [28,41,48,49,84,116].

Uma análise de vulnerabilidades de proteções possíveis para o AODV é apresentado nas referências [28,41,116]. Aspectos de segurança do protocolo DSR são analisados em [49,84]. Nesta última referência, é proposto um protocolo de roteamento seguro (*Secure Routing Protocol* – SPR) que é diretamente aplicado como extensão dos protocolos DSR e ZRP. Em [48], os autores discutem a análise de segurança de protocolos de roteamento por vetor de distância, pró-ativos, com foco no protocolo DSDV.

A maioria dos mecanismos de segurança propostos tem base em algum tipo de extensão de autenticação especificada para o protocolo de roteamento. Dahill *et al.* [28] propõem o protocolo ARAN (*Authenticated Routing for Ad hoc Networks*), uma versão modificada do protocolo AODV, onde são utilizados certificados digitais definidos de maneira particular para autenticar as mensagens do protocolo de roteamento. A autenticação é obtida pela inclusão de assinatura(s) digital(is) em cada mensagem do protocolo. Os certificados são especificados para prover uma ligação entre o endereço IP de um nodo e sua chave pública[10], que é usada na validação das assinaturas digitais incluídas nas mensagens. Um servidor centralizado que tem a confiança de todos os nodos da Manet provê os certificados. Como existem nas mensagens do protocolo informações que são modificadas durante o encaminhamento entre origem e destino, as mensagens são assinadas não apenas pelo nodo remetente. A contrário, todo nodo que encaminha tais mensagens com informação mutável (e.g. mensagens *route discovery* e *route reply*) deve assinar igualmente a mensagem. Esta solução demanda, portanto, um consumo elevado dos recursos de computação, além de provocar o aumento significativo do tamanho das mensagens a cada salto. Em uma abordagem similar, M. Zapata e N. Asokan propõem o protocolo SAODV (Secure AODV) [41]. O SAODV usa uma extensão de segurança, que é enviada juntamente com cada mensagem do AODV. A especificação original das mensagens AODV é preservada. Em oposição ao protocolo ARAN, SAODV requer que a mensagem seja assinada apenas pelo emitente. Para prover proteção aos campos com informação mutáveis (e.g. "*hop count*"), são usadas cadeias de *hash* (resumo) [50]. A proposta explora a natureza previsível dos campos

[10] O uso de certificados digitais para ligar o endereço IP de um nodo a sua chave pública parece inadequado, porque o endereçamento em redes *ad hoc* deve seguir a tendência atual da adoção de alocação dinâmica de endereços e autoconfiguração [41].

mutáveis, que são incrementados monotonicamente. Uma breve discussão sobre o uso da extensão de segurança para securizar outros protocolos de roteamento para Manet, especialmente o DSR, é apresentada igualmente. Ambas as abordagens descritas em [28,41] dependem da existência de serviços de certificação, que são assumidos como parte da iniciação da rede e dos nodos (*bootstrap*) [28] ou apenas discutidos *en passant* [41].

Uma alternativa ao uso de criptografia assimétrica, como nos trabalhos descritos acima, consiste no estabelecimento de associações de segurança entre nodos, permitindo o uso de primitivas de criptografia simétrica. Essas associações podem ser derivadas da sincronização de nodos, como em [89] ou diretamente a partir da mobilidade, possibilitando associações de segurança local apenas [19]. As abordagens em [48,49,84] também fazem uso de extensões de autenticação de mensagens, mas com uso de criptografia simétrica. Em [84], P. Papadimitratos e Z. Haas apresentam o SRP, projetado para securizar protocolos de roteamento reativos, com enfoque nos protocolos DSR e IERP. No SRP, para cada descoberta de rotas, origem e destino devem possuir uma associação de segurança estabelecida entre eles (i.e. compartilhem uma chave secreta). Além disso, SRP não provê qualquer proteção para mensagens de erro de rota (*route error*), permanecendo este protocolo vulnerável a ataques que utilizem estas mensagens. Nos protocolos ARIADNE [49] e SEAD (*Secure Efficient Ad hoc Distance vector*) [48], propostos por Y. Hu *et al.*, as chaves de autenticação são extraídas de um protocolo de autenticação por difusão (*broadcast*), denominado TESLA [89]. Este protocolo, no entanto, requer algum nível de sincronização de relógio entre os nodos da rede *ad hoc*, o que nem sempre é uma hipótese realista para Manet. ARIADNE é projetado para prover segurança em protocolos de roteamento reativos, com especial enfoque para DSR e IERP. SEAD propõe mecanismos de segurança com aplicação em protocolos de roteamento do tipo vetor de distância, com análise detalhada do protocolo DSDV, combinando autenticação por criptografia simétrica e o uso de cadeias de *hash*[11] para autenticação de campos mutáveis que são incrementados monotonicamente (neste caso, “hop count” e “sequence number”). As estratégias de segurança para concepção dos protocolos ARIADNE e SEAD são generalizadas por Y. Hu *at al.* em [50].

Uma abordagem diferente é proposta por H. Yang *et al.* [116]. O protocolo projetado é o AODV-S, que consiste em uma versão modificada do protocolo AODV (adiciona-se o “next hop” - próximo salto - às mensagens RREQ e as mensagens RREP são inundadas (*flooding*) na rede, ao invés de serem enviadas em *unicast* para o nodo requerente). Não há autenticação

[11] Cadeias *hash* são usadas aqui de maneira similar ao SADOV [41]

para as mensagens do AODV-S. Alternativamente, uma ficha de filiação (*token*) é continuamente emitida e renovada para cada nodo membro, sendo essas fichas usadas para nodos mal-comportados/comprometidos da troca de mensagens do protocolo de roteamento. Devido à natureza difusora (*broadcast*) do canal de comunicação sem-fio, todos os nodos que executam o AODV-S monitoram promiscuamente[12] as mensagens do protocolo, na tentativa de detectar ataques contra o protocolo de roteamento. Infelizmente, o projeto deste protocolo está baseado em uma hipótese equivocada, de que um adversário não pode personificar a identidade de outro nodo, caso essa seja expressa pelo endereço de acesso ao meio (MAC).

Uma revisão dos trabalhos discutidos acima leva à conclusão de que o estabelecimento de uma relação de confiança[13] entre os nodos precede o uso dos protocolos seguros. De fato, em muitos trabalhos esta fase é presumida como parte da iniciação da rede ou deixada de lado, assumindo-se a existência prévia das relações de confiança entre os nodos [28,41,84].

A Tabela 2-3 apresenta um comparativo das soluções de segurança apresentadas acima, destacando as principais técnicas utilizadas.

A contribuição deste trabalho no que diz respeito à segurança de protocolos de roteamento consiste no projeto de uma extensão de autenticação para Manet (*Manet Authentication Extension* – MAE), que incorpora e generaliza diversas técnicas utilizadas em outras propostas. Esta solução apresenta as seguintes vantagens [93]:

- Independência quanto ao protocolo de roteamento: A especificação para a extensão de autenticação é fornecida em termos de sintaxe e semântica para uso com qualquer protocolo de roteamento, sem alteração da especificação original do protocolo. Discute-se o uso de nossa MAE com os quatro protocolos de roteamento em processo de padronização pelo IETF (i.e. AODV, OLSR, TBRPF e DSR) e define-se precisamente os objetos de autenticação necessários para autenticar seguramente cada uma das mensagens utilizadas. O objetivo consiste em propor uma solução que seja flexível e extensível, de modo a possibilitar as extensões que atendam às necessidades específicas de autenticação de cada protocolo.

[12] A interface da rede *ad hoc* opera no modo promíscuo e cada mensagem "escutada" é copiada e analisada pelo módulo de monitoração.

[13] A relação de confiança pode ser estabelecida por um modelo de certificação digital com distribuição de chaves de criptografia assimétrica, como em [28,41], ou pelo uso de associações de segurança, com o compartilhamento de chaves secretas e o uso de criptografia simétrica, como em [48,49,84].

Tabela 2-3 – Comparação entre as soluções de segurança para protocolos de roteamento

Sistema	Protocolo(s) analisado(s)	Alterações no protocolo de roteamento original	Modelo de confiança / gerenciamento de chaves	Sistema de autenticação	Outras técnicas
ARAN [28]	AODV, DSR	SIM	distribuição de certificados presumida	múltiplas assinaturas digitais, com uso de certificação	-
SAODV [41]	AODV	NÃO	distribuição de certificados presumida	assinatura digital do emissor, com uso de certificação	cadeias de *hash* para autenticação de campos mutáveis
SRP [84]	DSR, IERP	NÃO	associação de segurança presumida	autenticação com criptografia simétrica	-
ARIADNE [49]	DSR	SIM	chaves criptográficas derivadas do TESLA	autenticação com criptografia simétrica	-
SEAD [48]	DSDV	NÃO	chaves criptográficas derivadas do TESLA	autenticação com criptografia simétrica	cadeias de *hash* para autenticação de campos mutáveis
AODV-S [116]	AODV	SIM	fichas de associação com identificação dos nodos	-	monitoração pró-ativa das mensagens e eliminação de nodos mal-comportados

- Adaptação ao modelo de confiança: A MAE é completamente adaptada ao modelo de segurança adotado, que provê a distribuição de certificados de forma auto-organizada. O sistema de autenticação prevê o uso de assinaturas digitais e o encapsulamento do certificado(s) da(s) entidade que assina(m) a mensagem[14]. Não obstante, a MAE projetada permite, alternativamente, o uso de primitivas de criptografia simétrica, compatibilizando seu uso com sistemas de autenticação tais como TESLA [89] ou autenticação derivada da mobilidade [19], entre outros. A escolha do mecanismo de autenticação adotado é configurável, permitindo a adaptação da solução a diferentes cenários de uso das Manets e a diferentes políticas de segurança.
- Segurança de campos com informação mutável: Em consonância com outros trabalhos similares [50], o uso de cadeias de *hash* para securização de campos mutáveis é suportado.

[14] Em geral, apenas o emissor da mensagem deve assiná-la. Entretanto, é possível carregar assinaturas digitais e certificados de mais de uma entidade, caso sejam necessárias assinaturas adicionais.

- Finalmente, discute-se e avalia-se o uso da MAE na provisão do serviço de autenticação para protocolos de roteamento pró-ativo do tipo estado de enlace, como é o caso do TBRPF e do OLSR. No caso específico do OLSR, são apresentadas ainda algumas otimizações decorrentes de interações diretas entre o protocolo de roteamento e o protocolo do serviço de certificação (e.g. mecanismo de *flooding* do OLSR pode ser usado para inundação de mensagens do serviço de certificação, aproveitando-se das otimizações introduzidas pelo uso de MPRs).

2.3. SEGURANÇA DOS PROTOCOLOS DE AUTOCONFIGURAÇÃO

Enquanto um esforço considerável tem sido dispensado ao projeto e padronização de protocolos de roteamento para Manet, o projeto de protocolos de autoconfiguração está ainda em seus primeiros estágios. Como conseqüência, propostas para melhoramentos de segurança para protocolos de roteamento para Manet estão aparecendo rapidamente, contudo a literatura sobre protocolos seguros para autoconfiguração em Manet ainda é rara. Nesta seção são apresentadas algumas propostas para a solução da questão da autoconfiguração em Manet.

2.3.1. Protocolos de Autoconfiguração

A alternativas tradicionais para autoconfiguração em redes TCP/IP envolvem a distribuição dinâmica de endereços pelo uso do protocolo DHCP [33] e, mais recentemente, a autoconfiguração por alocação aleatória de endereços [23], como proposto pelo grupo de trabalho *zeroconf*[15] do IETF. Entretanto, a utilização do protocolo DHCP requer a existência de um servidor centralizado para a distribuição de informações como endereço IP, máscara de rede, *gateway* padrão e outras informações adicionais de rede. Como já visto, o uso de servidores centralizados representa um problema em redes *ad hoc*. Por outro lado, os protocolos do grupo de trabalho *zeroconf* ainda estão em estágios preliminares.

Desse modo, os mecanismos adotados nas redes tradicionais não estão adaptados ao uso em ambientes *ad hoc*. Nesse sentido, um conjunto de novas propostas para soluções de autoconfiguração especialmente projetadas e adaptadas para Manet começa a aparecer.

[15] http://www.ietf.org/html.charters/zeroconf-charter.html

O protocolo de autoconfiguração deve permitir a alocação automática de endereços IP[16] (e de outros parâmetros da rede, tais como endereços de servidores de DNS). Para se desenvolver um protocolo de autoconfiguração que esteja adaptado aos diversos cenários de aplicação das Manets, foram identificadas as seguintes características qualitativas [118], mostradas na Tabela 2-4.

H. Zhou *et al.* [118] definem algumas métricas que podem ser usadas para analisar a performance de um protocolo de autoconfiguração em redes *ad hoc*, mostradas na Tabela 2-5 a seguir.

Os protocolos de autoconfiguração podem ser classificados de duas formas. Uma forma é com relação ao processo de autoconfiguração e a segunda forma diz respeito aos mecanismos utilizados para a detecção de endereços duplicados, atentando-se em como e quando esses endereços duplicados são detectados. A seguir são apresentadas as duas classificações:

Quanto ao processo de autoconfiguração:

- Independente (*stateless*): Permite que o nodo construa seu próprio endereço IP, baseado ou no identificador do hardware ou em um número randômico. Esse processo não depende de uma segunda entidade para fazer a autoconfiguração. Após a construção do endereço IP, um mecanismo de detecção de endereços duplicados é necessário para assegurar a unicidade do endereço gerado.
- Dependente (*stateful*): Requer que cada nodo na rede mantenha um conjunto de endereços IP. Isso implica na necessidade de participação de uma segunda entidade no processo de associação de um novo endereço IP. Além disso, a manutenção de uma estrutura comum e distribuída entre todos os nodos da rede, requer consumo de largura de banda principalmente na presença freqüente de junção e partição de redes *ad hoc*.

Quanto ao processo de detecção de endereços duplicados (DAD)[17]:

- Alocação com Detecção de Conflitos: A detecção de conflitos adota a política de tentativa e erro. O nodo escolhe um endereço IP por tentativa e faz uma requisição esperando pela aprovação de todos os nodos da rede *ad hoc*. Se algum nodo na rede responder negativamente, significa que esse endereço IP já está sendo usado.

[16] Ao contrário das redes com base em infra-estrutura, em Manet a configuração da máscara de sub-rede e do *gateway* padrão não são geralmente necessárias, uma vez que o roteamento é feito nodo-a-nodo e um protocolo de roteamento deve estar sempre ativo.

[17] Do inglês, *duplicated address detection.*

Tabela 2-4 – Métricas qualitativas para um protocolo de autoconfiguração para Manet	
Métrica	Descrição
Operação distribuída	Propriedade essencial das redes *ad hoc*, uma vez que a centralização de informações é inviável neste contexto.
Unicidade dos endereços IP	Assegurar que dois ou mais nodos não obtenham o mesmo endereço IP.
Saída de nodos da rede	Um endereço IP é associado a um nodo somente pelo tempo em que ele permanece na rede. Quando um nodo deixa a rede, o seu endereço IP deve ficar disponível para ser associado a outros nodos.
Perda de mensagens	Caso algum nodo falhe ou ocorra perda de mensagens, o protocolo deve agir rápido o suficiente para que não ocorra de dois ou mais nodos possuírem o mesmo endereço IP.
Operação multi-hop	Um nodo só não será configurado com um endereço IP quando não houver nenhum endereço IP disponível em toda a rede. Sendo assim, se qualquer nodo da rede possuir um endereço IP livre, este endereço deve ser associado ao nodo que está solicitando um endereço IP, mesmo que esteja a dois saltos ou mais de distância.
Suporte a partição e junção (*merging*) de redes	O protocolo deve ser capaz de ajustar a distribuição de endereços, quando ocorre a junção de duas ou mais redes *ad hoc* distintas, assim como a partição de uma rede em redes menores.
Segurança	Se as camadas de rede e de enlace não garantirem segurança, os protocolos estarão vulneráveis a muitas formas de ataque; é necessário que haja mecanismos adicionais para inibir modificações nas operações dos protocolos.
Operação no período de inatividade (*sleeping mode*)	Durante o período de inatividade, os endereços alocados aos nodos adormecidos podem ser considerados liberados pela rede. Por isso, é necessário que os nodos sejam capazes de notificar a rede sobre uma eminente transição para o estado de inatividade.

Tabela 2-5 - Métricas quantitativas para um protocolo de autoconfiguração para Manet	
Métrica	Descrição
Tempo para detecção de endereços duplicados	Dois ou mais nodos não podem possuir o mesmo endereço IP. Caso ocorram conflitos, estes devem ser detectados no menor período de tempo possível.
Complexidade	Levando em conta a quantidade limitada de memória e um baixo poder computacional dos nodos móveis, a solução deve ser a mais simples possível.
Eficiência	Protocolos baseados em difusão acarretam em consumo excessivo de banda e devem ser evitados, quando possível. A comunicação somente entre nodos vizinhos (localização) é preferível.
Uniformidade	Caso o protocolo distribua uniformemente os endereços, a probabilidade de conflitos é baixa, resultando em menos processamento na rede.
Latência	Latência é o tempo entre o início do processo de autoconfiguração e a sua conclusão, quando é associado um endereço IP livre a um novo nodo.

- Alocação por Melhor Esforço: Os nodos da rede são responsáveis pela associação de endereços IP para os novos nodos, tentando associar um endereço livre que não esteja sendo usado por nenhum nodo na rede. Todos os nodos da rede mantêm uma tabela dos endereços IP que estão em uso ou livres na rede. Assim, quando um novo nodo chega à rede, o seu vizinho mais próximo vai escolher um endereço IP livre para associar a ele. O problema é que dois ou mais nodos podem chegar simultaneamente e os nodos podem oferecer o mesmo endereço IP. A vantagem desse protocolo é que ele funciona muito bem com protocolos de roteamento pró-ativo, pois os nodos freqüentemente realizam difusão com as informações dos endereços já usados na rede.
- Alocação Livre de Conflitos: Usa o conceito de divisão binária, que significa que cada nodo possui conjuntos disjuntos de endereços IP. Cada nodo pode associar um endereço IP sem a necessidade de consultar outros nodos para se obter aprovação. Assim, todos os nodos da rede são responsáveis pelo processo de associação de um endereço IP. Esse mecanismo possui a vantagem de não necessitar de difusão para associar um endereço IP.

Alguns autores classificam os mecanismos de detecção de endereços duplicados de uma outra maneira. Em [114] os protocolos são classificados como ativos e passivos. Os protocolos ativos são aqueles que distribuem informação adicional na rede, sendo necessários pacotes adicionais de controle para que o protocolo funcione perfeitamente. Já os protocolos passivos são aqueles que detectam endereços duplicados, mas sem a necessidade de disseminar pacotes adicionais de controle na rede. Tudo é feito apenas monitorando o tráfego do protocolo de roteamento. Entretanto, esse último método tem um período de tempo em que pode haver a entrega de pacotes para o destino errado, sendo chamado de período de vulnerabilidade.

A autoconfiguração em Manet é um assunto bastante novo no ambiente acadêmico e incipiente no mercado de redes de comunicação de dados. Pesquisas feitas sobre esse tema mostram poucos trabalhos já realizados. Não existe nenhum padrão adotado e todas as fontes que estão disponibilizadas na Internet ainda estão em formato de proposta (*draft*) ou são artigos publicados em congressos especializados da área. A seguir, apresentamos algumas propostas para a autoconfiguração de endereços em redes *ad hoc*.

2.3.2. Alocação com Detecção de Endereços Duplicados

Perkins *et al.* propõem [87] um protocolo com base em uma política de tentativa e erro. Um nodo escolhe um endereço IP aleatoriamente e executa uma requisição de detecção de endereços duplicados (mensagem *Address Request* - AREQ), disseminada para todos os nodos da rede. O requerente aguarda, então, por um período fixo de tempo, por respostas positivas (mensagem *Address Reply* - AREP), indicando que o endereço escolhido já está alocado. Para esse processo de requisição e resposta, o nodo solicitante utiliza um endereço temporário escolhido de um bloco de endereços reduzido, que é reservado exclusivamente para o processo de autoconfiguração. Isso permite que a mensagem de resposta seja encaminhada diretamente ao solicitante por *unicast*. Para esse endereço temporário não há tratamento quanto a endereços duplicados, pois o endereço é alocado apenas por um período curto de tempo. Este protocolo não suporta partição e junção de redes e utiliza um processo de inundação (*flooding*) para obtenção da aprovação de todos os nodos da rede para a escolha de endereços. A solução é pouco escalável e pode acarretar em grandes latências na autoconfiguração ou mesmo uma sobrecarga excessiva na rede.

O protocolo proposto por J. Jeong *et al*. [57] trabalha com dois mecanismos de detecção de endereços duplicados. O primeiro é chamado detecção forte de endereços duplicados (*Strong* DAD), associando tempos limitados para a troca de mensagens. O segundo é denominado detecção fraca de endereços duplicados (*Weak* DAD) e utiliza as mensagens do protocolo de roteamento para checar a unicidade dos endereços. A checagem de DAD ocorre durante a atribuição do endereço para nodos entrantes (*Strong* DAD) e durante a junção de redes (*Weak* DAD). Este protocolo é similar ao usado em [87]. Entretanto, a partição e a junção de redes são consideradas. Necessita de integração com um protocolo de roteamento pró-ativo.

Finalmente, na proposta de K. Weniger [114], a detecção de endereços duplicados é feita de uma forma passiva, realizada somente pelo monitoramento do tráfego do protocolo de roteamento. Baseando-se em protocolos de roteamento pró-ativo, três formas diferentes são empregadas. Uma forma é utilizar o número de seqüência utilizado pelos protocolos de roteamento. A segunda forma é baseada no princípio da localização, devido ao fato que nos protocolos de roteamento pró-ativos, os nodos movem-se com velocidade limitada. Já na terceira forma, a vizinhança é a característica a ser explorada visto que um nodo conhece toda a sua vizinhança e também a vizinhança da origem do pacote de roteamento. O protocolo necessita de mecanismos adicionais para trabalhar com junção e partição de redes *ad hoc*.

2.3.3. Alocação por Melhor Esforço

Na proposta de S. Nesargi e R. Prakash [81] é apresentada uma alternativa ao uso de endereços IP temporários [87]. Cada nodo já configurado mantém uma estrutura de dados que contém os endereços IP associados (*allocated*) e os endereços IP que ainda estão pendentes (*pending*) por estar em processo de autoconfiguração. Um nodo solicitando um endereço IP envia uma mensagem (*broadcast*) para seus vizinhos que já fazem parte da rede. Um desses vizinhos responde à solicitação e passa a atuar como "procurador" do nodo solicitante. O procurador escolhe um endereço que não esteja na lista *allocated* e dissemina (*flooding*) um pedido de autorização para todos os nodos da rede (mensagem *Initiator_Request* - IREQ). Todos os nodos devem responder positivamente, autorizando a alocação do novo endereço. Caso alguma resposta seja negativa, o nodo procurador deve, então, escolher outro endereço e refazer o processo de DAD, obtendo aprovação de todos os outros nodos da rede para que a associação do endereço escolhido seja concluída. O protocolo trabalha com junção e partição de redes *ad hoc*.

De maneira semelhante, Boleng [10] propõe o uso de procuradores (aqui denominados nodos de atadura) para a escolha do endereço a ser atribuído a um novo nodo e para a realização do DAD. Os procuradores fazem a escolha dos endereços utilizando uma numeração seqüencial, o que permite armazenar apenas o último endereço alocado. Entretanto, para permitir o reuso de endereços alocados para nodos que deixam a rede, os nodos configurados mantêm também um *cache* de endereços disponíveis, que é construído a partir de notificações de saída enviadas em *broadcast* pelos nodos que partem. A característica saliente desta proposta consiste na adoção de um espaço de endereçamento com tamanho variável (e.g. endereços de tamanho variável), com objetivo de minimizar o *overhead* na transmissão de dados. O espaço de endereçamento aumenta e diminui de acordo com a necessidade (entrada e partida de nodos). Entretanto, esta abordagem exclui a possibilidade de uso de outros softwares que utilizem explicitamente o formato de endereçamento do IP. A partição e junção de redes é tratada igualmente.

2.3.4. Alocação Livre de Conflitos

Na alocação livre de conflitos, segue-se uma abordagem completamente diferente das soluções discutidas anteriormente. Nesse tipo de protocolo de autoconfiguração, cada nodo possui um conjunto de endereços IP que são usados para configurar novos nodos que chegam

à rede sem precisar consultar qualquer outro nodo já configurado da rede. Dentro de uma mesma rede *ad hoc*, esses conjuntos são disjuntos.

O DCDP (*Dynamic Configuration Distribution Protocol*), proposto por A. Misra *et al.* [78] é um exemplo desse tipo de protocolo e utiliza o modelo *buddy system* [102], para fornecer conjuntos disjuntos de endereços IP aos nodos da rede. Esse modelo adota um mecanismo de divisão binária do bloco de endereços. Um nodo desejando se associar a uma rede (nodo cliente), faz uma requisição (*broadcast*) por um endereço IP (mensagem *Address_Request*). Os nodos vizinhos já configurados respondem à solicitação (mensagem *Address_Reply*), informando o tamanho do seu bloco de endereços disponíveis (*free_ip_block*). O nodo cliente, então, seleciona o vizinho com o maior *free_ip_block* (mensagem *Server_Pool*). O vizinho selecionado (nodo servidor) divide seu conjunto de endereços IP em duas metades e envia uma metade para o nodo cliente (mensagem *IP_Assigned*), mantendo a outra metade consigo para atender futuras requisições. Quando o cliente recebe seu conjunto de endereços, ele associa o primeiro a si mesmo e mantém o resto como um conjunto de endereços disponíveis. Feito isso, o nodo cliente envia uma mensagem confirmando o sucesso da operação (mensagem *IP_Assignment_OK*). O processo é ilustrado na Figura 2-2.

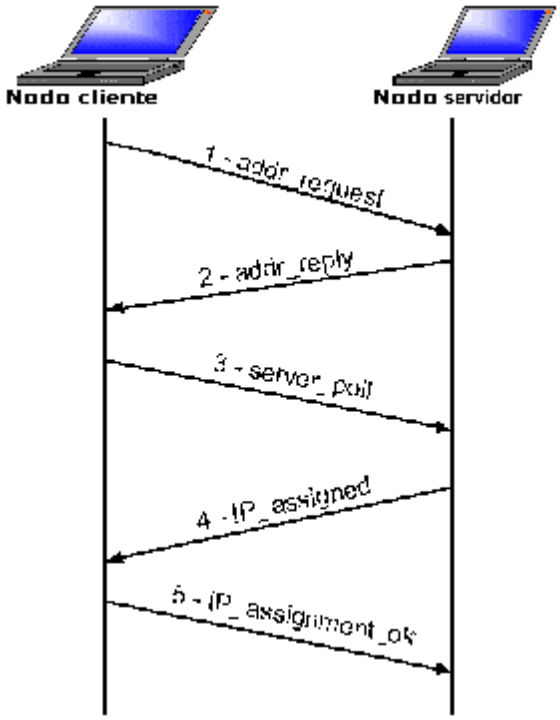


Figura 2-2 – DCDP - Associação de um endereço IP a um novo nodo
(Fonte: F. Buiati [14] p. 57)

Em trabalho recente, F. Buiati [14] apresenta uma especificação completa para o protocolo DCDP, tendo por base o trabalho feito em [78] com os melhoramentos propostos

em [80]. Esta proposta trabalha com junção e partição de rede *ad hoc* por manter conjuntos distintos de endereços IP nos nodos configurados da rede.

2.3.5. Segurança dos Protocolos de Autoconfiguração

A segurança do processo de autoconfiguração em redes *ad hoc* é assunto escasso, senão inexistente até pouco tempo atrás na literatura técnica. Em [13] é apresentada uma primeira abordagem para este tema, que será também discutida mais adiante neste trabalho. Esta abordagem consiste na adoção do modelo de confiança desde a entrada de um novo nodo na rede. Nesse sentido, é requerido que um nodo obtenha inicialmente a confiança da rede (i.e. obtenha um certificado), através do uso dos serviços de certificação distribuídos. Somente após completar esta etapa, o nodo realiza o processo de autoconfiguração de endereço IP. As mensagens do serviço de autoconfiguração devem ser todas autenticadas com a extensão de autenticação para Manet (MAE) [93]. Toda a comunicação neste processo, incluindo a autoconfiguração de endereços, ocorre entre vizinhos a um salto de distância e não necessita de um endereço IP previamente configurado[18]. A proposta de autoconfiguração segura neste trabalho tem por base o protocolo DCDP e detalha as vulnerabilidades deste protocolo e o modelo de proteção aplicado a ele. Não obstante, a solução é geral e pode facilmente ser utilizada com outros protocolos de autoconfiguração, desde que a certificação dos novos nodos ocorra antes da execução do processo de autoconfiguração. Para tanto, basta que as mensagens do protocolo de autoconfiguração carreguem em anexo uma MAE com a informação apropriada para autenticá-las (no caso do DCDP, basta uma assinatura digital, já que não há campos mutáveis nas mensagens).

Um trabalho simultâneo e, de certo modo, semelhante ao apresentado neste trabalho é apresentado por A. Cavalli e J. Orset [20]. Entretanto, nesta proposta os autores se limitam em definir uma extensão de autenticação para as mensagens do protocolo de autoconfiguração, assumindo a distribuição prévia de certificados, sem especificar como esta é realizada.

2.4. DETECÇÃO DE INTRUSÃO EM MANET

A maioria da pesquisa corrente em segurança de Manet está devotada à provisão de proteção preventiva dos protocolos básicos (e.g. roteamento), através de um mecanismo de autenticação similar ao nosso [28,41,84]. Como uma regra geral, essas soluções isoladamente

[18] Alternativamente, pode-se permitir que o nodo novato selecione um endereço de um bloco reservado para o processo de autoconfiguração, como em [87].

não são tolerantes à presença de nodos comprometidos na rede. Esses mecanismos de segurança podem ser reforçados por serviços de segurança pró-ativos, tais como sistemas de detecção de intrusão.

Sistemas de detecção de intrusão têm por objetivo detectar ataques contra sistemas de computação e redes, ou de uma forma geral, contra sistemas de informação. De fato, é consideravelmente difícil, senão impossível, prover sistemas de informação provavelmente seguros e mantê-los neste estado de segurança durante toda sua vida e utilização.Assim, os sistemas de detecção de intrusão têm a finalidade de monitorar o uso destes sistemas para detectar a aparição de estados inseguros.

De uma maneira geral, o interesse acerca da detecção de intrusão atualmente pode ser dividido em três processos básicos: coleta de dados, projeto do algoritmo de detecção (análise) e gerenciamento de alertas. O grupo de trabalho sobre detecção de intrusão do IETF (IDWG)[19] define os componentes que executam essas tarefas [115], conforme mostrado na Figura 2-3. O sensor coleta dados brutos acerca da operação do sistema monitorado (e.g. traços de auditoria, pacotes de rede). Esses dados são pré-processados e resultam em eventos que são enviados ao analisador, onde os eventos gerados são avaliados em termos de um mecanismo de detecção de intrusões. Se esses eventos forem sensitivos, o analisador gera alertas, que são repassados ao gerenciador. Este componente, por fim, além de realizar a correlação e classificação dos alertas – com objetivo de refinar a análise prévia, provê as informações necessárias para a resposta aos ataques detectados.

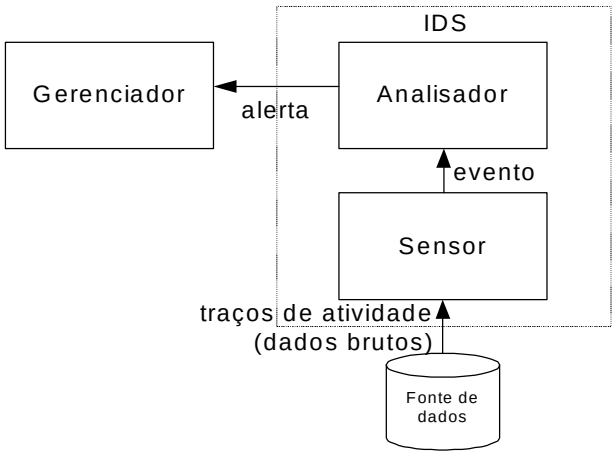


Figura 2-3 – *Framework* de detecção de intrusão do IDWG

[19] Do inglês, Intrusion Detection Working Group. O sítio oficial do grupo é **http://www.ietf.org/html.charters/idwg-charter.html**

H. Debar *et al.* [29] propõem uma taxionomia para os sistemas de detecção de intrusão atuais, mostrada na Figura 2-4. Esses critérios de classificação são discutidos abaixo:

Método de detecção: Descreve as características do analisador. Quando o IDS usa informações acerca do comportamento normal do sistema monitorado, buscando detectar variações deste estado normal, o IDS é dito com base em comportamento. Se o IDS utiliza informações acerca dos ataques que podem ser detectados (assinaturas de ataques), diz-se tratar de um IDS por uso incorreto.

Comportamento em caso de detecção: Descreve a resposta do IDS aos ataques detectados. Quando o sistema reage ativamente a um ataque executando ações corretivas (fechando brechas) ou pró-ativas (registrando possíveis atacantes, fechando serviços), o sistema é classificado com ativo. Se o sistema meramente gera e envia alertas (incluindo *pager*, etc.), ele é dito passivo.

Fonte de dados: Discrimina os IDS com base no tipo de informação de entrada que eles analisam. Essa informação pode ser trilhas de auditoria (e.g. *log* de sistema) em um computador, pacotes de rede, *log* de aplicativos ou mesmo alertas gerados por outros sistemas de detecção de intrusão.

Paradigma de detecção: Descreve o mecanismo de detecção usado pelo IDS. Estes sistemas podem avaliar estados (seguro/inseguro) ou transições (de seguro para inseguro).

Freqüência de uso: Certos IDS são usados na monitoração contínua e em tempo real do sistema alvo, enquanto outros são executados periodicamente.

Vale ressaltar que IDS têm requisitos específicos no contexto Manet que não são compatíveis com abordagens tradicionais para detecção de intrusão. Isto é, assim como os demais serviços de segurança, o IDS deve ser distribuído, auto-organizado e, se possível, operar localizadamente. Como o projeto de IDS para Manet é um assunto bastante recente, apresentamos a seguir um apanhado das principais iniciativas de concepção e de projeto de sistemas de detecção de intrusão que atendam a esses requisitos, mesmo que esses sistemas não tenham sido especificamente projetados para nosso ambiente alvo. Essa leitura é importante pois permite identificar conceitos chave na concepção de um IDS particularmente projetado para ambientes Manet.

Em primeira análise, pode-se dizer que muito se tem feito e estudado recentemente sobre sistemas de detecção de intrusão distribuídos. A Tabela 2-6 apresenta as principais propostas de sistemas de detecção de intrusão distribuídos, que já atingiram maturidade suficiente em seu desenvolvimento para permitir a validação de princípios e direções consideradas em cada projeto.

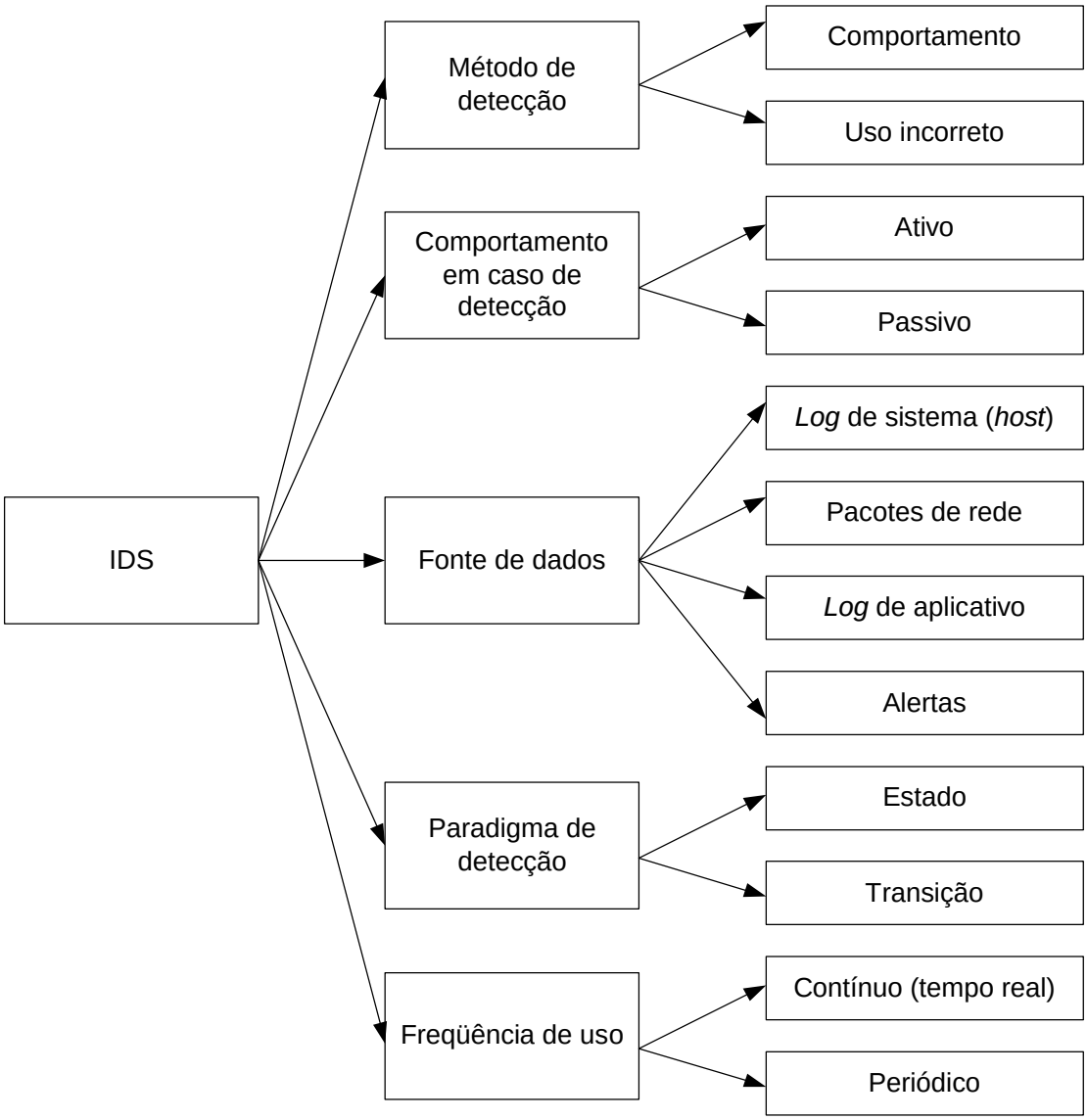


Figura 2-4 – Taxionomia dos Sistemas de Detecção de Intrusão

Tabela 2-6 – Principais propostas de IDS distribuídos

IDS	**Fonte de Dados**	**Método de Detecção**	**Pré-processamento distribuído**	**Detecção centralizada**	**Análise em tempo-real**	**Tipo de Resposta**
AAFID [103**]**	Sistema	Uso incorreto	Sim	Sim	Sim	Passivo
DIDS [106**]**	Sistema/Rede	Híbrido	Sim	Sim	Sim	Passivo
Grids [104**]**	Sistema/Rede	Híbrido	Sim	Sim	Não	Passivo
CSM [112**]**	Sistema	Anomalia	Sim	Não	Sim	Ativa
JiNao [37**]**	MIB/Rede	Híbrido	Sim	Sim	Sim	Passivo
EMERALD [90**]**	Sistema/Rede	Híbrido	Sim	Não	Sim	Ativo
IDA [5**]**	Sistema	Uso incorreto	Agentes móveis	Sim	Sim	Passivo
SPARTA [67**]**	Sistema/Rede	Uso incorreto	Agentes móveis	Não	Sim	Passivo

Todas as propostas apresentadas na Tabela 2-6, exceto CSM, EMERALD e SPARTA, são hierarquicamente organizadas em torno de um nodo central. Este nodo central é o cerne do IDS e usa informações coletadas de forma distribuída para detectar intrusões. Nestas arquiteturas, a distribuição está restrita ao processo de coleta de dados e, elas não são, portanto, adequadas ao contexto das Manets.

A arquitetura do CMS é completamente distribuída. Um IDS local é instalado em cada nodo cooperando para um identificação colaborativa do originador de conexões de rede. Já a arquitetura do EMERALD foi especialmente projetada para acomodar necessidades de escalabilidade em redes grandes. Este IDS é feito de nodos genéricos comunicantes, denominados monitores, que são instalados em cada sistema. Entre as arquiteturas na Tabela 2-6 apenas SPARTA foi especificamente projetado para ambientes de rede sem fio. Entretanto, esse sistema é projetado para detectar ataques contra aplicações distribuídas e não considera ataques contra a camada de rede.

No que diz respeito ao critério da auto-organização, o uso de plataformas de agentes oferece uma alternativa ao modelo de distribuição cliente-servidor. Em especial, pode-se considerar o uso de agentes móveis, que são agentes que se movem de um nodo a outro carregando dados e código executável. Com um projeto cuidadoso, esse tipo de agente permite reduzir consideravelmente a quantidade de dados trocados na rede, o que torna a arquitetura do IDS particularmente interessante em ambientes Manet, onde a largura de banda é limitada e os enlaces são pouco confiáveis. O uso de agentes móveis, em oposição às abordagens tradicionais de distribuição de aplicações onde os dados são transportados em direção ao sistema de computação que os processa, permite que o código mova até os dados. Adicionalmente, um nodo que despacha um agente móvel não precisa esperar por sua volta, antes de continuar seu processamento normal, uma vez que tais agentes podem ser despachados novamente ou destruídos em outros nodos, sem ter que mover-se de volta ao nodo criador.

O uso de agentes móveis na concepção de IDS é um tópico bastante aquecido nos últimos anos. A Tabela 2-7 resume as principais iniciativas neste campo.

W. Jasen [55] do projeto *Mobile Agent Security* do NIST (*National Institute of Standards and Technology*, EUA) provê uma análise elusiva dos possíveis benefícios e desvantagens do uso de agents móveis em detecção de intrusão, enquanto P. Mell *et al.* [77] definem uma arquitetura de IDS usando agentes móveis, resistente a ataques de negação de serviço. Entretanto, esta arquitetura é fortemente ligada à infra-estrutura da rede, o que não é observado em Manet. Uma abordagem similar pode ser encontrada em [21].

Tabela 2-7 IDS projetados com uso de agentes móveis

IDS	Fonte de dados	Método de detecção	Coleta de dados e Pré-processamento	Análise (detecção)	Correlação de alertas
AAFID [103**]**	Sistema (*host*)	Uso incorreto	Agentes móveis	Centralizada	Não
IDA [5**]**	Sistema (*host*)	Uso incorreto	Agentes móveis	Centralizada	Não
MAIDS [43**]**	Sistema/Rede	Híbrido	Agentes móveis	Centralizada	Sim
SPARTA [67**]**	Sistema/Rede	Uso incorreto	Agentes móveis	Agentes móveis	Não

E. Spafford *et al.* [103] (Purdue University, EUA) projetaram o sistema AAFID (*Autonomous Agents For Intrusion Detection*) e M. Asaka et al. [5] (Information-Technology Promotion Agency – IPA, Japão) propuseram o sistema IDA (Intrusion Detection Agent). Ambos os sistemas possuem uma arquitetura hierárquica organizada em torno de um nodo central. Este nodo é o núcleo do IDS e usa informações coletadas de maneira distribuída. Similarmente, G. Helmer *et al.* [43] (Iowa State University, USA) propõem o sistema MAIDS (*Mobile Agent Intrusion Detection System*), que envolve um IDS baseado em agentes inteligentes. Agentes móveis são usados para permitir vários tipos de agentes especializados, referidos como agentes de *baixo nível*, que viajam entre pontos de coleta de dados e implementam detecção simples de atividades suspeitas. Em cada instância do MAIDS, agentes colaborativos estáticos (agentes de *auto nível*), que especializam em uma categoria específica de intrusão e colaboram localmente uns com os outros para realização de inferências sobre a detecção de intrusão. A noção de cooperação entre instâncias do MAIDS é definida em termos de um sistema de *datawarehouse* centralizado. Outras abordagens tendo entidades centrais que realizam a análise de dados e a inferência sobre a intrusão pode ser encontrado em [8,9]. Em todas essas arquiteturas, a distribuição está restrita ao processo de coleta de dados e, portanto, não são conformes com o contexto Manet.

Uma vez mais, volta-se ao sistema SPARTA (*Security Policy Adaptation Reinforced Through Agents*), proposto por Krügel *et al.* [67] (Technical University – Vienna, Austria). Nesta proposta, desenvolvida concorrentemente com a proposta apresentada neste trabalho, agente móveis são usados para realizar eventos e consultas (*queries*), ambos sendo formalmente especificados em uma linguagem tipificada. Cada nodo em um IDS local autônomo, formado por sensores locais e uma plataforma de agentes móveis. Infelizmente, a arquitetura SPARTA possui um nodo central único, denominado console de gerenciamento. Apesar desta entidade central não estar diretamente envolvida no processo de detecção de intrusão, ela tem um papel importante na arquitetura do sistema, pois cada nodo que se junta

ao serviço de detecção colaborativo deve se registrar no console de gerenciamento. Essa última característica torna o SPARTA incompatível com o contexto Manet.

Finalmente, uma abordagem completamente diferente para o uso de agentes móveis em detecção de intrusão é proposta em trabalhos independentes de S. Hofmeyr *et al.* [45] (University of New Mexico, EUA) e S. Fenet *et al.*[35] (Université Claude Bernard Lyon, França). Em [45], a arquitetura IDS emula o sistema imunológico biológico pelo uso de agentes móveis especializados (cada padrão de intrusão é mapeado por um agente diferente) que perambulam pelo sistema monitorado procurando por traços de intrusões. Em [35] os agentes móveis imitam o comportamento de insetos sociais. Infelizmente, o padrão de mobilidade é pobremente descrito em ambos os sistemas, tornando difícil avaliar a usabilidade destas proposições em contextos Manet.

O projeto de IDS para Manet não é uma questão completamente nova e este assunto já foi tratado recentemente [3,51,73,79,111,117]. Y Zhang e W. Lee [117] introduzem os requisitos básicos para este tipo especial de IDS. Em um trabalho prévio, introduzimos os conceitos preliminares de arquitetura [3,85,94].

Em [42], S. Gwalani *et al.* propõem um IDS para Manet que é essencialmente projetado para reforçar a segurança do protocolo de roteamento. Entretanto, este IDS tem uma arquitetura centralizada.

V. Mittal e G. Vigna [79] apresentam um IDS formado por vários sensores para detecção de ataques contra o protocolo de roteamento, que monitoram promiscuamente os enlaces de rede. Noções de colaboração aparecem neste IDS, porém o mecanismo de detecção supõe que informações globais de topologia estão disponíveis. Em Manet, o mais apropriado seria usar informações de topologia localizada, pois a topologia é dinâmica e as informações de topologia globais podem não estar completamente atualizadas.

Y. Huang *et al.* [51] e C.-Y. Tseng *et al.* [111] apresentam projetos de IDS para Manet com base em uma estratégia de detecção por anomalia. O inconveniente desses trabalhos reside na ausência de cooperação entre os nodos, sendo que cada nodo age isoladamente na detecção de ataques.

Uma estratégia de detecção e resposta à intrusão para lidar com nodos não cooperantes em redes *ad hoc* é apresentado por S. Marti *et al.* em [73]. Entretanto, não há nenhuma noção de serviços de segurança colaborativa nesta abordagem. No trabalho em [116] é apresentada uma solução de segurança com base em uma versão modificada do AODV, que utiliza um mecanismo de detecção de intrusão combinado com um sistema de fichas de filiação (*tokens*) que são usadas para garantir o acesso dos nodos aos serviços de roteamento. Entretanto, a esta

solução não incorpora nenhuma proteção preventiva (autenticação). Ao invés, apenas um mecanismo simples de verificação de vizinhança é usado. Infelizmente, como mencionado anteriormente, este mecanismo é baseado em uma hipótese errônea de que endereços MAC não podem ser personificados. Adicionalmente, o mecanismo de detecção de intrusão está restrito apenas à inundação de mensagens RREP, não generalizando para lidar com todos os ataques descritos em termos de fabricação, modificação e personificação de outras mensagens do protocolo de roteamento.

Neste trabalho, é apresentado um sistema de detecção de intrusão completamente distribuído [95], no sentido que os processos de coleta de dados, análise (detecção) e gerenciamento de alertas ocorrem de maneira distribuída. Nenhuma entidade central é requerida. Neste projeto, um IDS local (LIDS) é colocado em cada nodo da Manet. Os LIDS intercomunicam-se usando um mecanismo que leva em consideração as restrições quanto à limitação de largura de banda e conectividade. Para prover a auto-organização, usa-se uma plataforma de agentes móveis. Adicionalmente, o processo de detecção colaborativa é feito basicamente na vizinhança ou em um número restrito de nodos envolvidos em um ataque com multi-fases, mantendo o *overhead* de comunicação e processamento, localizado ou entre nodos selecionados da rede. Finalmente, ao contrário da maioria dos trabalhos discutidos anteriormente, o IDS apresentado neste trabalho permite uma efetiva estratégia de resposta ativa ao ataques detectados, estando completamente integrado com os demais serviços de segurança.

3. MODELO DE SEGURANÇA PARA MANET

Neste capítulo, é apresentado o modelo de segurança proposto para as Manets. Para definir as propriedades do modelo, é discutido inicialmente o modelo de vulnerabilidades e dos adversários. Isso é importante, pois define o escopo da proteção oferecida pelo modelo – não há solução de segurança com abrangência completa e de escopo ilimitado. A caracterização das vulnerabilidades e adversários permite estabelecer os requisitos de segurança para as redes, que são considerados na concepção do modelo.

No que diz respeito ao modelo de segurança propriamente dito, este é constituído por uma combinação de serviços de segurança preventivos (certificação/autenticação) e corretivos (detecção e resposta à intrusão), que também são projetados no contexto deste trabalho, com projeto especialmente voltado para ambientes Manet. Na concepção do modelo, inclui-se ainda a integração com serviços de segurança existentes e disponibilizados nos principais ambientes operacionais (e.g. filtro de pacotes, túneis criptográficos entre aplicativos – SSL/TSL – e entre redes ou nodos – IPSec). Finalmente, discute-se a extensão do modelo para incorporar outros serviços de segurança que ainda não foram projetados (e.g. *firewall* distribuído, gerência de política de segurança distribuída).

3.1. MODELO DE VULNERABILIDADES E DE ADVERSÁRIOS

Existe um espectro relativamente largo de possíveis falhas de segurança e fraquezas nos sistemas computacionais, em especial aqueles que operam em rede, que levam a extensivas listas de vulnerabilidades. Diversos autores têm discutido a questão das vulnerabilidades de sistemas computacionais, a exemplo de Landwehr *et al.* [68], que apresentam um apanhado de vulnerabilidades potenciais de sistemas de computação e propõem uma taxonomia para essas falhas. Sem o objetivo de fazer um apanhado completo acerca das vulnerabilidades de segurança em sistemas computacionais, pode-se citar a iniciativa do CERT[20] (*Computer Emergency Response Team*) que mantém um repositório ativo das principais vulnerabilidades observadas e reportadas em incidentes de segurança da informação.

A exemplo do trabalho de Landwehr *et al.* [68] e da iniciativa do CERT, a maioria dos apanhados acerca de vulnerabilidades analisa sistemas em ambientes de rede convencionais.

[20] **http://www.cert.org**

Claramente, muitas das vulnerabilidades existentes nesses ambientes são igualmente possíveis em redes *ad hoc*. Entretanto, o ambiente de redes *ad hoc*, devido às suas características particulares, possui vulnerabilidades que lhe são próprias e não são observadas em outros tipos de rede. Além disso, muitas vulnerabilidades existentes em ambientes de redes convencionais possuem formas de exploração particulares em ambientes de redes *ad hoc*, constituindo-se, de fato, em novas vulnerabilidades que são exclusivas destas redes. Neste trabalho, o foco da discussão está nas vulnerabilidades que são decorrentes do ambiente *ad hoc*. Em especial, são analisadas as diversas falhas de segurança relacionadas com os serviços básicos dessas redes, i.e. os protocolos de roteamento e autoconfiguração.

3.1.1. Modelo de Vulnerabilidades

Para a caracterização de vulnerabilidades específicas de redes *ad hoc*, torna-se importante identificar quais as características dessas redes que fazem com que os principais aspectos de segurança devam ser tratados de forma diferenciada com relação às redes tradicionais. Esses aspectos são:

- Redes *ad hoc* utilizam essencialmente comunicações sem fio. A ausência de um atamento físico entre o nodo da rede e uma infra-estrutura possibilita uma série de ações:
  - Escuta: o canal de comunicação sem fio é passível de escuta telemática, bastando para isso que o dispositivo que faz escuta esteja no raio de alcance do(s) transmissor(es) sem fio.
  - Comunicação direta: em uma rede *ad hoc* basta que dois nodos estejam suficientemente próximos (alcance da transmissão sem fio) para que seja possível a comunicação entre eles.
  - Mobilidade: nodos podem mover-se livremente (com velocidade limitada) em uma rede *ad hoc*, saindo do alcance de comunicação direta com alguns nodos e entrando no de outros.
- Os nodos colaboram entre si para estabelecer e manter a conectividade na rede (i.e. roteamento), assim como os demais serviços essenciais, já que não existe uma entidade centralizada na rede. Desse modo, ao contrário das redes tradicionais, onde os serviços essenciais são localizados em pontos específicos da rede (e.g. *gateways*, servidores de autoconfiguração) que possuem proteção apropriada e acesso controlado, em Manet esses serviços são distribuídos entre os diversos nodos que devem colaborar entre si para assegurar o correto funcionamento da

rede. Assim, a não colaboração de um ou alguns nodos, seja por mau funcionamento ou por egoísmo (e.g. um nodo não colabora para economizar sua bateria), pode comprometer as funcionalidades da rede como um todo.

- Nodos usam fonte de energia portátil (bateria). Esse tipo de alimentação de energia se extingue com o uso. Assim, diversos serviços e protocolos são projetados para operarem de modo eficiente no que diz respeito ao consumo de energia. É possível, no entanto, provocar um comportamento errôneo em um ou mais nodos de uma Manet, fazendo com que estes passem a consumir sua bateria mais rapidamente, reduzindo seu tempo útil de atuação na rede.

Finalmente, um outro aspecto fundamental na concepção das vulnerabilidades das redes *ad hoc* consiste na possível existência de nodos comprometidos. Definimos nodo comprometido com um nodo que tenha adquirido a confiança prévia da rede e que começa a agir de forma errônea em algum momento. Esse comportamento inadequado pode ser resultado de uma falha individual, seja esta intencional ou não, ou mesmo decorrente do comprometimento por agentes externos à rede. Assim, as vulnerabilidades que não podem ser normalmente exploradas por nodos não confiáveis, continuam existindo como falhas potenciais para exploração por nodos comprometidos que possuam a confiança da rede.

Nosso modelo de segurança considera as seguintes vulnerabilidades, que são exploradas em maiores detalhes no caso dos serviços de roteamento e autoconfiguração:

- Modificação: mensagens são modificadas de maneira incorreta ao serem encaminhadas por nodos intermediários entre o remetente e o destinatário.
- Personificação: neste caso, um nodo qualquer utiliza a identidade de outro nodo para gerar mensagens na rede.
- Fabricação: consiste na geração de mensagens falsas.
- Não-colaboração: ocorre quando um nodo concorda em colaborar com seus pares na rede, mas não cumpre seu papel quando sua colaboração é requerida[21].

3.1.2. Modelo de Adversários

Definimos como adversário qualquer agente que execute ações com objetivo de corromper os serviços da Manet (ataques). O adversário pode ser igualmente interno (nodo

[21] É fundamental diferenciar não-colaboração de não-participação. Neste caso, um nodo está fisicamente presente, mas não concorda explicitamente em participar dos serviços colaborativos da Manet. Isto é, este nodo não executa o protocolo de roteamento ou de autoconfiguração, por exemplo.

confiável que age incorretamente) ou externo (nodo comprometido por um agente externo). Em especial, admitimos que um nodo legítimo (confiável) possa ser comprometido por agentes externos através da exploração de vulnerabilidades do sistema operacional ou de serviços executados normalmente em um nodo, ou mesmo pela captura (física) de um nodo que não possua proteção adequada.

Admitimos que o comprometimento de um nodo por um agente externo acarreta na revelação de todas as informações secretas mantidas no nodo (e.g. chaves secretas, chaves privadas, senhas, etc.). O adversário pode, então, lançar ataques desde o nodo comprometido ou simplesmente personificá-lo.

Considerando as características específicas do ambiente *ad hoc* e as vulnerabilidades definidas no escopo deste trabalho, identificamos as principais ações que um adversário pode executar no sentido de comprometer as funcionalidades da rede:

- Escutar promiscuamente o canal de comunicação sem fio e obter informações a partir do tráfego gerado em e destinado para seus vizinhos.
- Comunicar-se diretamente com qualquer nodo dentro do seu alcance de transmissão.
- Mover-se, com velocidade limitada, para coletar informações sobre nodos mais distantes, comunicar-se diretamente com outros nodos que não estejam na sua vizinhança ou mesmo escapar da monitoração de nodos próximos.
- Não-colaborar com nodos próximos, mesmo tendo concordado em fazê-lo (e.g. execução do protocolo de roteamento ou de autoconfiguração).

Finalmente, consideramos que múltiplos adversários possam co-existir na rede e estes podem se coordenar para comprometer os mecanismos de segurança colocados em funcionamento. Não diferenciamos adversários internos e externos do ponto de vista dos serviços de segurança, pois ainda que seja possível diferenciar nodos confiáveis e não confiáveis, o ambiente Manet não está garantidamente livre de nodos comprometidos por agentes externos.

3.1.3. Requisitos de Segurança

Uma análise dos modelos de vulnerabilidades e de adversários mostrada na seção anterior permite definir um conjunto de requisitos que devem ser atendidos pelo modelo de segurança pretendido para Manet. Nesta seção, discutimos estes requisitos:

- Nodos confiáveis *versus* nodos não-confiáveis: Deve existir uma maneira eficaz de verificação imediata para identificar se uma mensagem é proveniente de um nodo

confiável ou não. Mensagens provenientes de nodos não confiáveis devem ser processadas de acordo com a política de segurança, podendo ser prontamente descartadas de maneira silenciosa.

- Proteção contra modificação: Uma mensagem não pode ser modificada de maneira incorreta na rede por nodos não confiáveis. Igualmente, a modificação errônea de mensagens por nodos confiáveis deve ser passível de detecção.
- Proteção contra personificação: Um nodo confiável deve possuir uma identidade única. Essa identidade não pode ser usurpada por nodos não confiáveis ou mesmo por outros nodos confiáveis[22].
- Proteção contra fabricação: Mensagens fabricadas não podem ser injetadas na rede por nodos não confiáveis. Igualmente, a fabricação de mensagens falsas por nodos confiáveis deve ser passível de detecção.
- Proteção contra não-colaboração: Como a não-colaboração não pode ser efetivamente evitada, esta deve ser passível de detecção.
- Eliminação de nodos comprometidos: Nodos confiáveis que estejam comprometidos devem ser eliminados dos serviços colaborativos da rede, pela revogação da confiança que lhes foi outorgada.

3.2. MODELO DE SEGURANÇA

Nesta seção é apresentado o modelo de segurança proposto para redes *ad hoc*. Este modelo considera os requisitos genéricos de Manet, ao definir serviços de segurança que operam segundo um modelo de serviços auto-organizados e distribuídos. Em seguida, é apresentado um modelo de confiança distribuído que permite contemplar o requisito de diferenciação entre nodos confiáveis e não-confiáveis em um ambiente Manet. Esse modelo tem base em um serviço de certificação distribuído, que provê igualmente uma maneira eficaz de identificação unívoca dos nodos. Finalmente, é proposta a integração de um conjunto de serviços de segurança, com destaque para um serviço de autenticação e um serviço de detecção de intrusão, que interagem entre si para prover os demais requisitos de segurança apontados.

[22] O uso de endereços IP ou endereços MAC para esta finalidade é inadequado, uma vez que é comprovadamente possível personificar essas informações.

3.2.1. Modelo de Serviços Auto-organizados e Distribuídos

Uma arquitetura geral para serviços distribuídos, auto-organizados e colaborativos em Manets é mostrada na Figura 3-1. Cada nodo da Manet tem uma instância autônoma e ativa do serviço. Essas instâncias são chamadas genericamente L-Service (do inglês, *Local Service*). Um L-Service colabora com L-Services de nodos próximos através de um protocolo de colaboração. A colaboração pode se iniciar a qualquer tempo, e ser estabelecida entre quaisquer outros L-Services que estejam disponíveis no instante em que a colaboração se inicia. Está é uma característica fundamental, uma vez que a disponibilidade de nodos individuais não pode ser assegurada, dado que um nodo pode simplesmente mover-se para fora do alcance de comunicação da rede. Esta noção de auto-organização é exatamente a mesma usada na concepção do serviço de roteamento em Manet, o L-Service sendo representado pelo *daemon*, que é executado autonomamente em cada nodo da Manet, e o protocolo de cooperação sendo representado pelo protocolo de roteamento. Assim, cada um dos serviços de segurança definidos em nosso modelo, juntamente com outros serviços da Manet, inclusive os serviços de roteamento e autoconfiguração considerados nesse trabalho, enquadram-se nessa arquitetura geral. Nota-se que nenhuma entidade centralizada é requerida na concepção dos serviços que seguem o modelo da Figura 3-1.

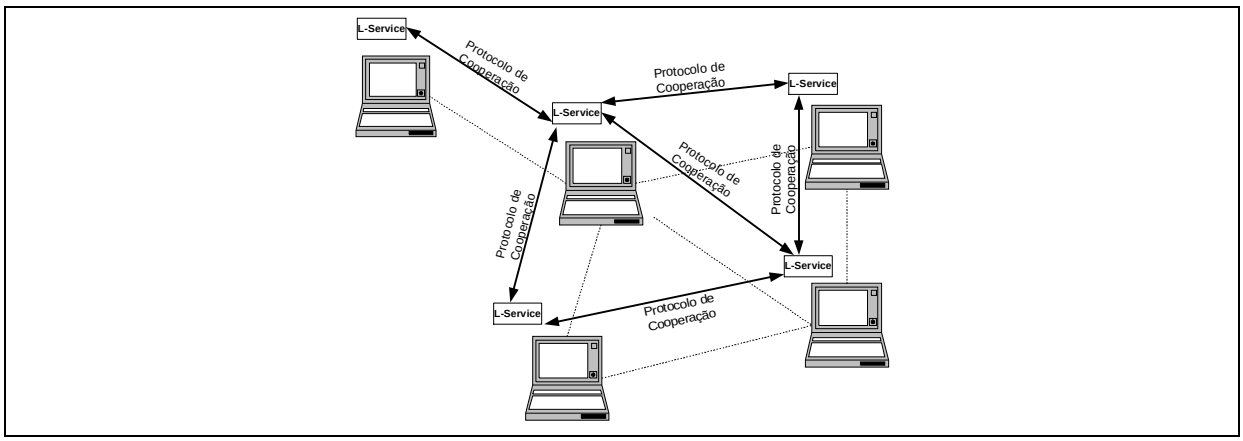


Figura 3-1 – Modelo de Serviços Distribuídos, Colaborativos e Auto-organizados

Na colaboração ilustrada na Figura 3-1, os nodos de uma Manet trocam informações entre si através de protocolos de colaboração definidos para cada serviço. Além disso, essa colaboração deve se estender até o serviço de encaminhamento de pacotes, uma vez que os nodos dependem uns dos outros para manter a conectividade na rede.

Finalmente, o projeto do protocolo de cooperação deve ser cuidadosamente realizado com o objetivo de manter a cooperação localizada e a interação restrita à vizinhança (1-*hop*) dos nodos ou a um número limitado e selecionado de nodos. Assim, limita-se o *overhead* de

comunicação entre os L-Services. Este requisito é devido às restrições de banda passante e de fornecimento de energia existentes nestes ambientes de rede.

3.2.2. Modelo de Confiança Distribuída

Uma questão fundamental no que diz respeito à segurança dos mecanismos colaborativos das Manets consiste na especificação adequada da filiação dos nodos com a rede, permitindo diferenciar nodos que são confiáveis daqueles que não o são. Esta associação assim definida (i.e. relacionada com a noção de confiança mútua entre os nodos) pode ser imposta aos nodos como uma primeira linha de proteção para os serviços colaborativos, requerendo-se que os nodos estejam previamente associados à rede e restringindo-se a cooperação apenas entre os nodos que são membros da rede. Neste cenário, a troca de informações através dos protocolos de colaboração definidos para cada serviço e o encaminhamento de pacotes se dá no âmbito de um conjunto de nodos que confiam mutuamente entre si.

Há três aspectos importantes em relação à definição e imposição de uma associação com a Manet apresentada acima. Primeiramente, deve-se considerar o próprio processo de estabelecimento de confiança (e.g. de associação com a rede). Uma alternativa consiste na definição de uma relação de confiança entre cada par de nodos que estejam interagindo (modelo *peer-to-peer*). Ainda que essa solução possa ser eficaz em alguns casos específicos, ela não é escalável para situações onde uma Manet é formada por algumas dezenas ou centenas de nodos. Assim, de uma maneira geral, esse processo também deve seguir uma abordagem colaborativa. As associações com a Manet devem poder ser revogadas no caso de se detectar algum nodo comprometido na rede. Em segundo lugar, existe a imposição da associação ao modelo de colaboração. Uma vez estabelecida a associação, um nodo deve ser capaz de provar para os outros membros que ele está associado à rede, assim como deve poder constatar e verificar a filiação reclamada por outros nodos. Para que isso seja possível, um serviço básico de autenticação da origem[23] deve estar disponível em todos os protocolos de colaboração. Finalmente, o terceiro aspecto a ser considerado diz respeito à própria noção de confiança, que varia para cada Manet em particular. Cenários diferentes podem ser

[23] Quando se considera os requisitos de segurança associados à eliminação de nodos comprometidos, deve-se requerer que esta autenticação proveja igualmente o não-repúdio, uma vez que a detecção de comportamento incorreto por nodos vizinhos é considerada como base para emissão de acusação contra esses nodos.

considerados, variando de situações completamente abertas (todos são confiáveis) a situações extremamente restritivas (ninguém é confiável).

Em nossos trabalhos, consideramos que esse modelo de confiança é realizado por meio de um serviço de certificação distribuída. A associação com a rede é expressa, então, na forma de certificados digitais que são emitidos em favor dos nodos considerados confiáveis em uma Manet. Como será discutido com maiores detalhes no próximo capítulo, a confiança é estabelecida de forma colaborativa, com a formação de uma autoridade de certificação distribuída onde o segredo de certificação (i.e. a chave privada da autoridade de certificação) é compartilhado entre os nodos participantes.

O modelo de confiança implementado por este serviço de certificação distribuída tem as seguintes vantagens:

- As relações de confiança em Manet são colaborativamente estabelecidas e mantidas, de forma escalável. O cancelamento da relação de confiança no caso de nodos comprometidos pode ser realizado simplesmente pela revogação do certificado do nodo.
- O uso de certificados digitais permite que cada nodo possa assinar digitalmente as mensagens que gera, possibilitando um serviço de autenticação da origem com não-repúdio. Os próprios certificados digitais são usados para identificação unívoca dos nodos e como proteção contra personificação por outros nodos não confiáveis.
- O uso do modelo em cenários de aplicação de Manet distintos é facilmente ajustado de acordo com a política de segurança, por meio da definição de parâmetros de operação que refletem os diversos requisitos de segurança de cada aplicação.

3.2.2.1. Modelo de Confiança K-de-N

O compartilhamento da chave privada da autoridade de certificação entre os nodos participantes é feito por meio da técnica de criptografia de limiar [102]. Com essa técnica, o segredo de certificação pode ser quebrado em tantas partes quanto se queira (e.g. N partes). Assim, potencialmente todos os nodos da rede podem ser portadores de uma parte da chave privada de certificação, se a política de segurança assim o permitir. Entretanto, apenas um número determinado de nodos (K) é requerido para que os serviços de certificação sejam realizados.

Este número K (o limiar) é uma constante importante do sistema e representa um compromisso entre o nível de segurança requerido e a escalabilidade do sistema. Quanto maior o valor de K, mais nodos devem participar da certificação colaborativa (i.e. confirmar sua confiança) para tornar disponíveis os serviços de certificação. Um sistema com K grande é, portanto, bastante robusto com relação à existência de nodos comprometidos, pois são requeridos pelo menos K nodos comprometidos e colaborando entre si para que a segurança do serviço de certificação seja quebrada, enquanto que K/2 nodos comprometidos podem colaborar para provocar falhas bizantinas no serviço [62]. No caso limite, quando K = N, todos os nodos da rede que sejam portadores de partes da chave privada devem dar sua aprovação para que os serviços de certificação sejam completados. Em contrapartida, valores grandes de K podem comprometer a disponibilidade e a escalabilidade do sistema. Assim, quanto menor o valor de K, menos *overhead* de comunicação é requerido para a provisão dos serviços. No caso limite, onde K = 1, o sistema é potencialmente inseguro, pois um nodo único pode isoladamente quebrar o sistema (i.e. todos os portadores conhecem a chave privada de certificação). Uma abordagem interessante é sugerida em [66] e consiste em escolher-se K com um valor próximo ao tamanho da vizinhança (1-*hop*), permitindo que os serviços de certificação estejam localizados e as comunicações sejam realizadas entre vizinhos dos nodos que necessitam de serviços de certificação.

Este modelo de confiança é do tipo “K-para-N”, onde K indica o número de nodos que devem confiar em um outro nodo para que este possa ser admitido na rede e N é o número (não fixo) de nodos que possuem uma parte da chave privada de certificação. Cada nodo que possua uma parte da chave privada da ACD pode, então, emitir certificados parciais para os nodos nos quais ele confia. Estes nodos, por sua vez, recuperam quaisquer K certificados parciais emitidos em seu favor e podem combiná-los para obter o certificado completo. A distribuição de partes da chave privada de certificação e a revogação de certificados são igualmente realizadas colaborativamente.

A provisão dos serviços de certificação é feita através de serviços locais de certificação (L-CERT), que colaboram entre si. Esses L-CERT formam dinamicamente coalizões de K nodos. O protocolo para os serviços básicos de certificação distribuída (emissão e revogação de certificados e distribuição de partes da chave privada de certificação) é ilustrado na Figura 3-2 a seguir. Os passos 1 a 3 consistem no estabelecimento de uma coalizão dinâmica de K nodos, que inicia-se com a requisição do serviço (passo 1), seguida pela coleta de respostas dos nodos que concordam em servir esta requisição (passo 2) e pela notificação da formação da coalizão, pelo requisitante, para todos os participantes (passo 3).

Finalmente, o serviço é completado quando são coletadas K respostas (resultados parciais) de todos os participantes da coalizão, devidamente assinadas com as partes das chaves privadas de cada um desses nodos (passo 4).

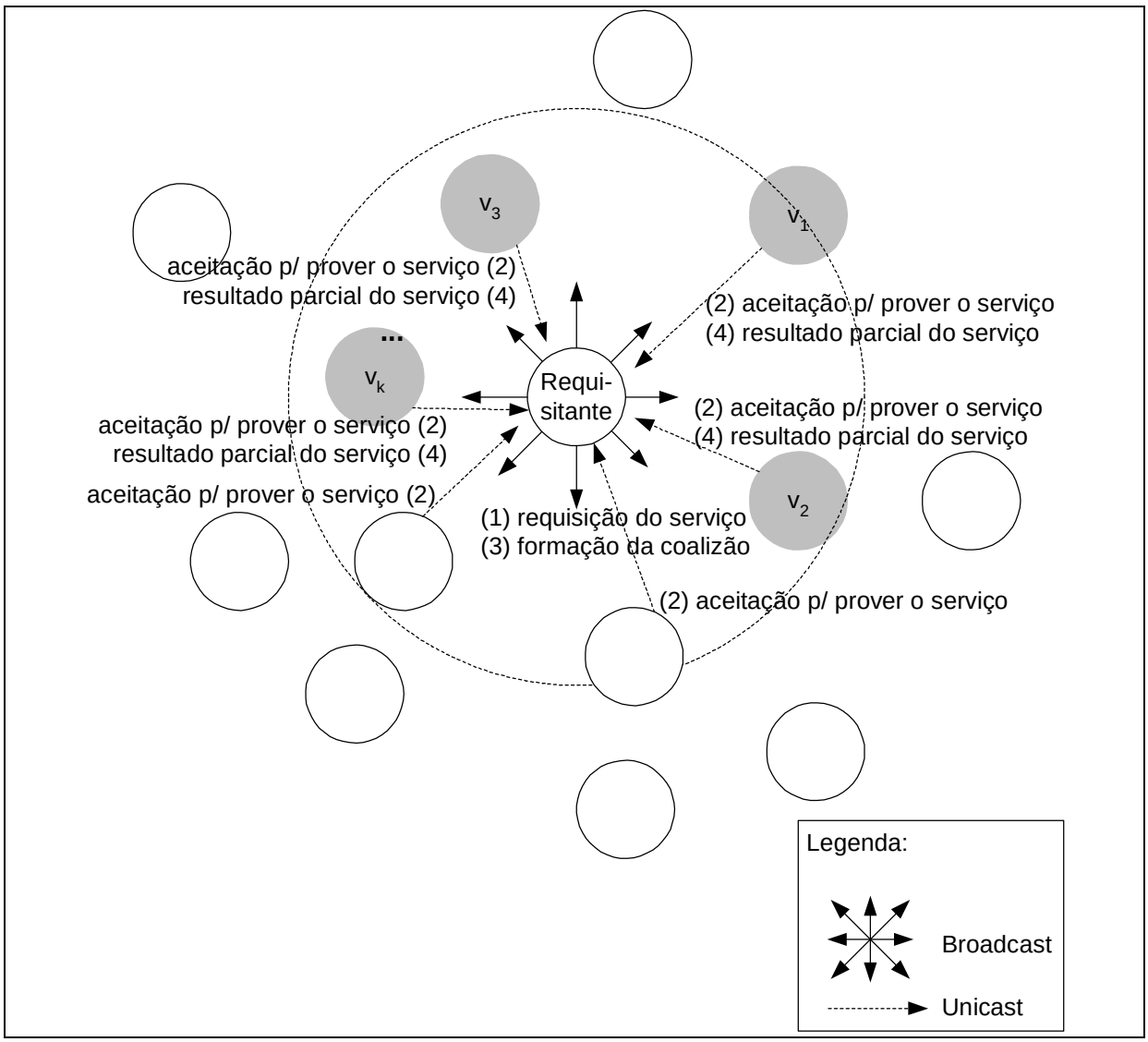


Figura 3-2 – Protocolo de Colaboração para Serviços de Certificação Distribuída

Finalmente, discute-se a revogação de certificados, utilizada para a eliminação de nodos comprometidos da rede. Nodos confiáveis que detectem atividades maliciosas de outros nodos podem gerar acusações contra estes. Desse modo, K nodos que geram acusações contra um nodo podem colaborar para revogar o seu certificado, eliminando-o da rede. Os mecanismos de detecção e acusação também devem ser executados colaborativamente, evitando que um único nodo comprometido gere acusações contra nodos corretos, tornando os serviços da rede indisponíveis para eles. Os certificados são revogados por meio da emissão de contra-certificados, que são assinados por K nodos que acusam colaborativamente um

nodo de comprometimento. Contra-certificados são armazenados localmente, formando uma lista de certificados revogados local (L-CRL).

3.2.2.2. Identificação de Nodos em uma Manet

Certificados digitais contêm uma ligação entre uma identidade e uma chave pública, que é usada em esquemas de criptografia assimétrica. A identificação expressa em um certificado pode ser um nome de uma entidade de rede (*host* ou serviço), um identificador de usuário (e.g. e-mail), entre outros. A chave pública, por sua vez, está associada a uma chave privada que é conhecida apenas pelo proprietário do certificado. Esta chave privada pode ser usada para autenticar por meio de assinaturas digitais as mensagens trocadas na rede.

Neste trabalho, utilizam-se os próprios certificados digitais para identificação dos nodos que são confiáveis na rede. Desse modo, os certificados são a única identificação considerada do ponto de vista dos serviços de segurança. Esta escolha deve-se ao fato de não haver uma identificação – para nodos ou usuários – que seja livre de personificação. No que diz respeito a endereços IP, a identificação usualmente considerada nos serviços de rede, ressalta-se que, além de ser este identificador facilmente usurpado, um nodo tende a ter seu endereço IP configurado dinamicamente, de modo que este endereço é potencialmente trocado, com freqüência. A associação de identificadores dos nodos com endereços MAC é mais perene, pois esses endereços vêm configurados de fábrica. Mesmo assim, um nodo pode mudar de endereço MAC ao trocar o *hardware* de sua interface de rede (e.g. um *notebook* pode trocar a placa de rede *wireless* PCMCIA). Adicionalmente, a personificação de endereços MAC é geralmente possível, uma vez que esses endereços podem ser alterados manualmente[24], além de ser possível a geração de quadros formados de maneira particular (e.g. API *rawsocket*) onde os endereços MAC são definidos por software. Neste caso, apenas alguns fabricantes reforçam o uso do endereço MAC do *hardware*, evitando que os quadros assim formados saiam na rede com o endereço MAC adulterado. Finamente, identificações típicas para usuários e nodos são nomes atribuídos pelos próprios usuários ou administradores de sistema e não possuem qualquer garantia de unicidade ou proteção quanto à personificação.

O uso de certificados digitais como identificação, do ponto de vista da segurança, oferece proteção criptográfica ao identificador, enquanto permite que a identidade seja

[24] Endereços MAC podem ser administrados localmente. Neste caso, o endereço único que foi atribuído à interface quando de sua fabricação não é utilizado.

verificada no processo de autenticação de mensagens pela origem. Com o uso de assinaturas digitais, essa autenticação é ainda não-repudiável. Não fazemos distinção entre o uso de nomes de entidades de rede (*host*) ou de usuários (e-mail) no identificador contido no certificado digital utilizado. Isso permite que um mesmo nodo possa ser usado por usuários diferentes, em momentos diferentes. Entretanto, para um nodo que pode ser usado por mais de um usuário, deve-se definir qual o certificado será usado para autenticar os serviços básicos da Manet (e.g. roteamento e autoconfiguração).

3.2.2.3. Considerações sobre a Política de Segurança

Um dos pontos mais críticos, e ainda aberto, no que diz respeito ao estabelecimento de confiança em Manet consiste na tradução da política de segurança adotada no ambiente de aplicação da rede em critérios objetivos que permitam impor esta política aos usuários através dos mecanismos de segurança adotados. Isso fica evidente quando se analisa as diversas propostas de modelos de confiança para Manet através de certificação digital [62,66,119]. Em muitas dessas propostas, a solução é estruturada admitindo-se previamente a política de segurança, principalmente no que diz respeito à emissão de novos certificados.

Em [119] os certificados devem ser emitidos previamente por uma autoridade de certificação centralizada e apenas a renovação de certificados ocorre colaborativamente. Pouco se trata de política de segurança, sendo esta deixada a critério da AC. Em [66], a AC centralizada emite apenas os certificados dos primeiros K nodos. Os certificados dos demais nodos são emitidos colaborativamente, segundo uma política que deve envolver algum tipo de verificação *off-line* com apresentação de provas físicas da identidade do requerente. Já em [71,116], dos mesmos autores de [66], uma requisição por um certificado sempre é servida por todos os nodos, exceto se o requerente já estiver identificado como um nodo mal comportado. Essa abordagem não impede que um atacante, sempre que descoberto, possa trocar de identidade e continuar na rede, ou mesmo forjar múltiplas identidades e recuperar o segredo de certificação após obter K chaves privadas parciais distintas. No trabalho de [52], os certificados são emitidos em uma base *peer-to-peer*, não havendo qualquer critério que uniformize a política de emissão de certificados em toda a rede.

No que diz respeito à renovação de certificados, as abordagens em [66,71,116] admitem que a renovação seja servida apenas se o requisitante for um nodo "bem comportado", i.e. se não existir acusações contra ele. Neste caso, não há mecanismos para garantir que o serviço de certificação distribuída continue a fazer valer esta política na

emissão de certificados. O problema é ainda mais crítico quando se trata de emissão de partes da chave privada de certificação.

Em qualquer caso, os requisitos mínimos para a política de segurança dos serviços de certificação distribuída são:

- A emissão de certificados requer a identificação unívoca do requisitante, pois, caso contrário, a solução está sujeita a quebras por um adversário que forje múltiplas identidades (ataques de Sybil). Como esta condição é difícil para nodos que não tenham um certificado prévio, a política de segurança deve estabelecer quais os critérios que devem ser usados para verificação da identidade do requisitante.
- A emissão de partes da chave privada de certificação e a renovação de certificados devem ocorrer apenas para nodos que já possuam um certificado válido. Este certificado deve autenticar as requisições para esses serviços, sendo esta a garantia mínima exigida para provisão do serviço pelos demais nodos.

Esses requisitos mínimos podem ser suficientes para cenários de aplicação de Manet com baixa probabilidade de comprometimento dos nodos, tais como um grupo de estudantes em uma sala de aula. Já em situações onde o comprometimento de nodos é mais provável, ou mesmo onde o impacto de um incidente de segurança é mais crítico, a política de segurança pode requerer uma verificação mais completa da identidade do requisitante na prestação quaisquer destes serviços, pois a apresentação de um certificado válido pode não ser suficiente para garantir que a política de certificação esteja obedecida o tempo todo, uma vez que o requerente pode ser um nodo comprometido.

Na proposta deste trabalho, os diversos parâmetros que definem os requisitos de segurança e de performance do sistema de certificação são configuráveis de acordo com a política de segurança adotada, sempre se respeitando os requisitos mínimos definidos anteriormente. Como exemplo para esses parâmetros configuráveis, podem-se citar as políticas de emissão e renovação de certificados, as disciplinas para distribuição e armazenamento de contra-certificados (i.e. certificados revogados), o tempo de validade de um certificado emitido/validado, entre outros. Essa abordagem permite que a solução de segurança seja adaptada à política de segurança, ao invés de restringir o escopo de utilização do modelo em ambientes com políticas pré-definidas.

3.2.3. Extensão de Autenticação para Manet (MAE)

O serviço de certificação distribuída apresentado na seção anterior provê uma solução robusta e eficiente para a distribuição da confiança. Entretanto, essa confiança tem que ser

imposta a todas as transações via rede que se deseja securizar. Este é o caso das mensagens dos protocolos de roteamento e de autoconfiguração, por exemplo. Para que isso seja possível, torna-se necessário anexar às mensagens em questão uma extensão de autenticação, contendo o(s) objeto(s) autenticador(es) das mensagens. Existem protocolos padronizados que são eficientes para troca de mensagens *peer-to-peer*, quando uma relação de confiança entre os pares já existe. Este é o caso do protocolo IPSec [109], para comunicação segura entre duas redes ou dois *hosts*, e do SSL/TLS [31], para estabelecimento de uma sessão de comunicação segura entre duas aplicações. Entretanto, essas soluções são adequadas apenas para casos onde a autenticação ocorre fim-a-fim, entre duas entidades que já tenham roteamento entre si. Por esses motivos, IPSec e SSL/TLS não se adaptam às necessidades de segurança dos protocolos de roteamento e autoconfiguração em Manet. Além disso, alguns protocolos de roteamento considerados possuem campos que são alterados na medida em que as mensagens atravessam a rede. Esses campos também precisam ser autenticados, requerendo-se o projeto de objetos autenticadores especiais e o uso de múltiplos objetos de autenticação, para esses casos.

Neste contexto, projetamos uma extensão de autenticação para Manet (MAE) que permite utilizar o nosso modelo de confiança para autenticar aplicações orientadas a mensagens que tenham requisitos que extrapolem as possibilidades de uso do IPSec. Nossa MAE, discutida em maiores detalhes no próximo capítulo, permite a incorporação de múltiplos objetos de autenticação para uma única mensagem autenticada e está completamente adaptada a operar juntamente com os certificados do modelo de confiança.

3.2.4. Detecção e Resposta a Intrusões em Manet

Enquanto os serviços de certificação e autenticação provêem condições básicas de segurança preventiva, evitando que nodos não confiáveis realizem ataques contra a Manet, um serviço de segurança corretiva ainda é necessário para considerar os requisitos de detecção de ataques realizados por nodos comprometidos ou por adversários que estejam personificando esses nodos, bem como a eliminação desses nodos dos serviços colaborativos da rede.

No que diz respeito aos mecanismos corretivos, duas abordagens básicas vem sendo estudadas e desenvolvidas. Primeiramente, quando o(s) nodo(s) adversário(s) pode(m) ser explicitamente identificado(s), notadamente quando este realiza ataques que envolvem ações tais como fabricação, modificação e personificação de mensagens dos protocolos de autoconfiguração e roteamento, assim como ataques de não-colaboração. A contra-medida corretiva que visa restabelecer o funcionamento normal da rede e dos serviços atacados

consiste na eliminação dos nodos adversários da rede, através da revogação de certificados. Estes nodos passam, então, a não poder mais participar e corromper os serviços colaborativos da Manet, pois suas mensagens não podem mais ser autenticadas. Alternativamente, em casos onde a origem dos ataques não pode ser identificada precisamente, pode-se tentar mitigar o efeito de ataques evitando-se a utilização de caminhos de rede que tenham apresentado problemas ou aplicando-se filtros de pacotes (nível de rede) ou de mensagens (nível de aplicação) para os fluxos de informações considerados anômalos. Enquanto, no primeiro caso, pretende-se uma identificação positiva do ataque e do adversário, no segundo realiza-se detecção de condições anômalas/degradadas de funcionamento.

Esses dois cenários podem ser prontamente relacionados às duas vertentes nas técnicas de detecção de intrusão atualmente em discussão. Na detecção de intrusão por uso incorreto (*misuse*), identifica-se a ocorrência de um ataque pelo reconhecimento de traços característicos do mesmo, previamente identificados e formalmente expressos em uma assinatura de ataque. Neste caso, a identificação do ataque é positiva e, se realizada com precisão, possibilita as condições necessárias para a acusação e eliminação de adversários. Na detecção de intrusão por comportamento, por sua vez, detecta-se condições de operação que estão em desvio com relação a condições normais previamente identificadas e formalmente expressas em um modelo de comportamento do sistema. A existência de ataques não é realizada explicitamente, mas as condições degradadas de operação podem ser mitigadas evitando-se caminhos de rede ou nodos que apresentem problemas/falhas de operação, através da aplicação de filtros seletivos em mensagens de roteamento, ou evitando-se o encaminhamento e/ou processamento de pacotes que façam parte de um perfil de utilização anômalo, pelo uso de filtros de pacotes (*firewall*). Essas contra-medidas são eficazes no caso de ataques de inundação (*flooding*) de pacotes (e.g. DDoS) e a utilização de *scanners* de rede.

Um serviço de detecção e resposta a intrusões (IDS) especialmente projetado para Manet provê o mecanismo de segurança corretivo que completa a nossa solução de segurança, atendendo a todos os requisitos discutidos na seção 3.1.3. Este IDS opera de maneira completamente distribuída. Um IDS local (L-IDS) é posicionado em cada nodo da rede e estes colaboram com L-IDS dos nodos vizinhos para coletar informações, executar o algoritmo de detecção e coordenar a resposta à intrusão. O L-IDS tem arquitetura modular e pode ser facilmente estendido para executar diferentes tipos de coletas de dados, estratégias de detecção e mecanismos de resposta.

O protocolo básico de resposta colaborativa à intrusão é constituído de três etapas, que são ilustradas na Figura 3-3. Inicialmente, os L-IDS colaboram para detectar um ataque,

movimentando agentes móveis. Cada vez que um ataque é detectado, um L-IDS gera uma acusação (alerta) assinada contra o adversário e a dissemina em toda a rede (passo 1). Esse processo é repetido por todos os L-IDS que detectam o mesmo ataque (passo 2). Quando o mecanismo de correlação de alertas de um L-IDS identifica K acusações contra um mesmo nodo, proveniente de L-IDSes diferentes, é iniciado um processo de revogação do certificado do acusado. Para tanto, uma requisição de formação de coalizão dinâmica é emitida para todos os L-IDS identificados como acusadores do nodo comprometido (passo 3). Esses nodos assinam um contra-certificado parcial contra o nodo comprometido e enviam-no para o requerente (passo 4). Este faz a reconstituição do contra-certificado completo, a partir dos K contra-certificados parciais recebidos e dissemina-o em toda a rede, completando a revogação do certificado (passo 5). Todos os nodos atualizam, então, sua CRL local.

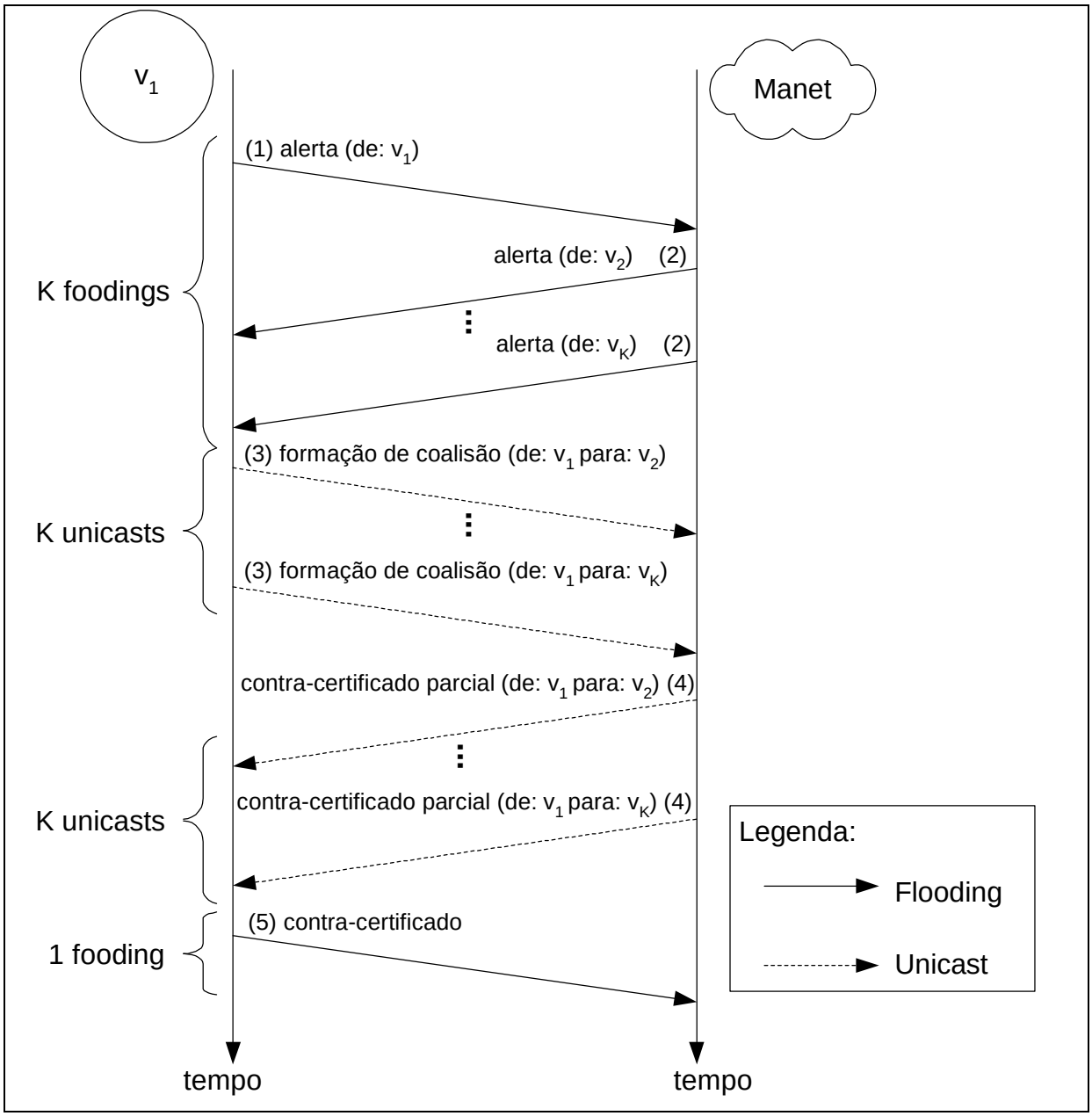


Figura 3-3 – Protocolo de Colaboração para Resposta à Intrusão

3.2.5. Serviços Integrados de Segurança

A solução de segurança apresentada neste trabalho consiste de um conjunto de serviços integrados que provêem conjuntamente uma solução de segurança distribuída e colaborativa que atenda aos requisitos gerais de Manet e aos requisitos de segurança discutidos na seção 3.1.3. Em especial, os mecanismos de certificação (L-CERT) e autenticação (MAE) provêem a segurança preventiva, enquanto os mecanismos de detecção e resposta à intrusão (L-IDS) fornecem a segurança corretiva. Essa combinação, ilustrada na Figura 3-4 é a característica saliente do modelo de segurança proposto.

Além dos serviços mencionados explicitamente no parágrafo anterior, o modelo de segurança pode ser estendido para promover a interação com outros serviços de segurança já projetados, tais como:

- serviço de *firewall* local, que provê um filtro de pacotes em nível de rede disponível localmente para configuração e re-configuração de acordo com a política de segurança e a estratégia de resposta à intrusão, assim como o estabelecimento de túneis criptográficos usando o protocolo IPSec para ligação privada entre duas redes através de uma rede virtual privada (VPN);
- serviço SSL/TLS, que utiliza o L-CERT para criação de sessões seguras (*peer-to-peer*) entre aplicativos.

Uma visão de implementação do modelo de segurança proposto é mostrada na Figura 3-4. Essa visão pode ser complementada com o modelo de arquitetura de protocolos para a solução proposta, mostrada na Figura 3-5.

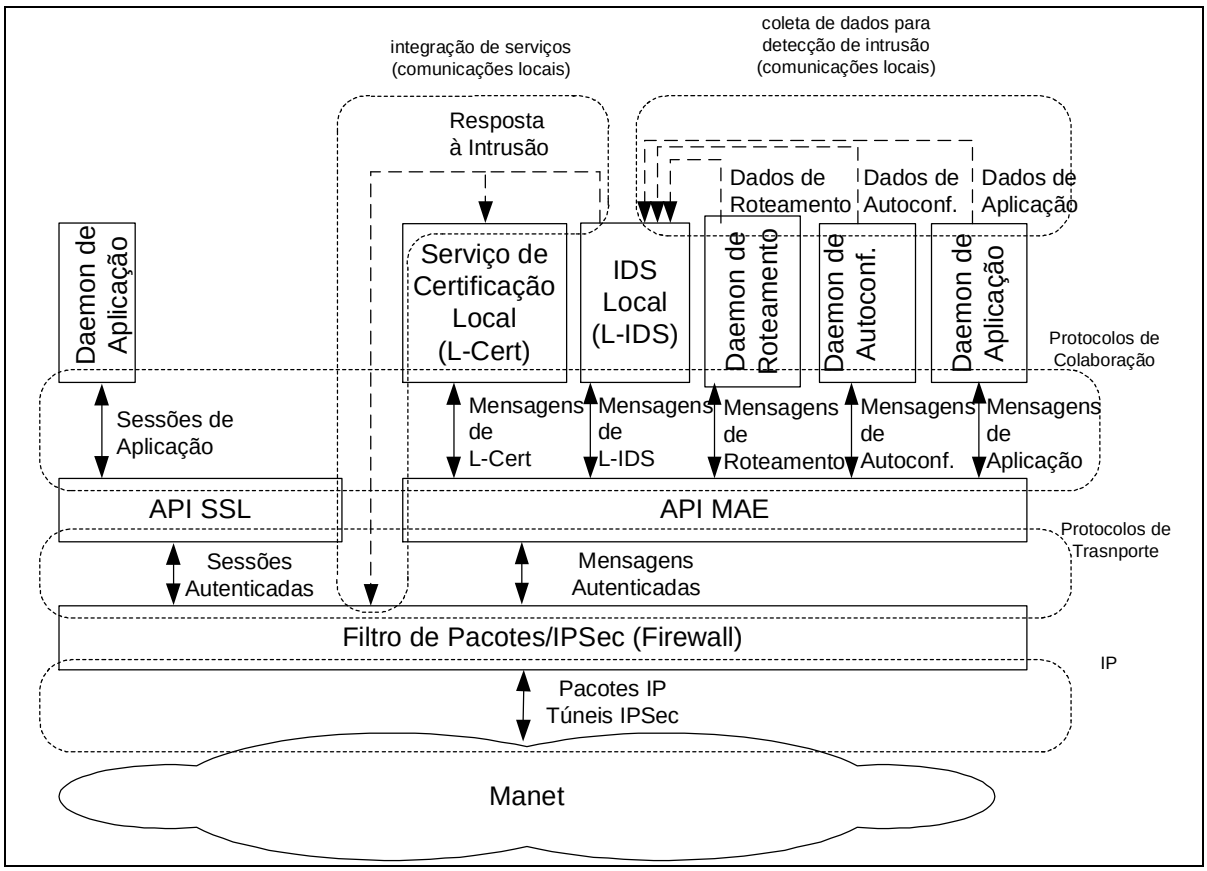


Figura 3-4 – Visão de Implementação do Modelo de Segurança

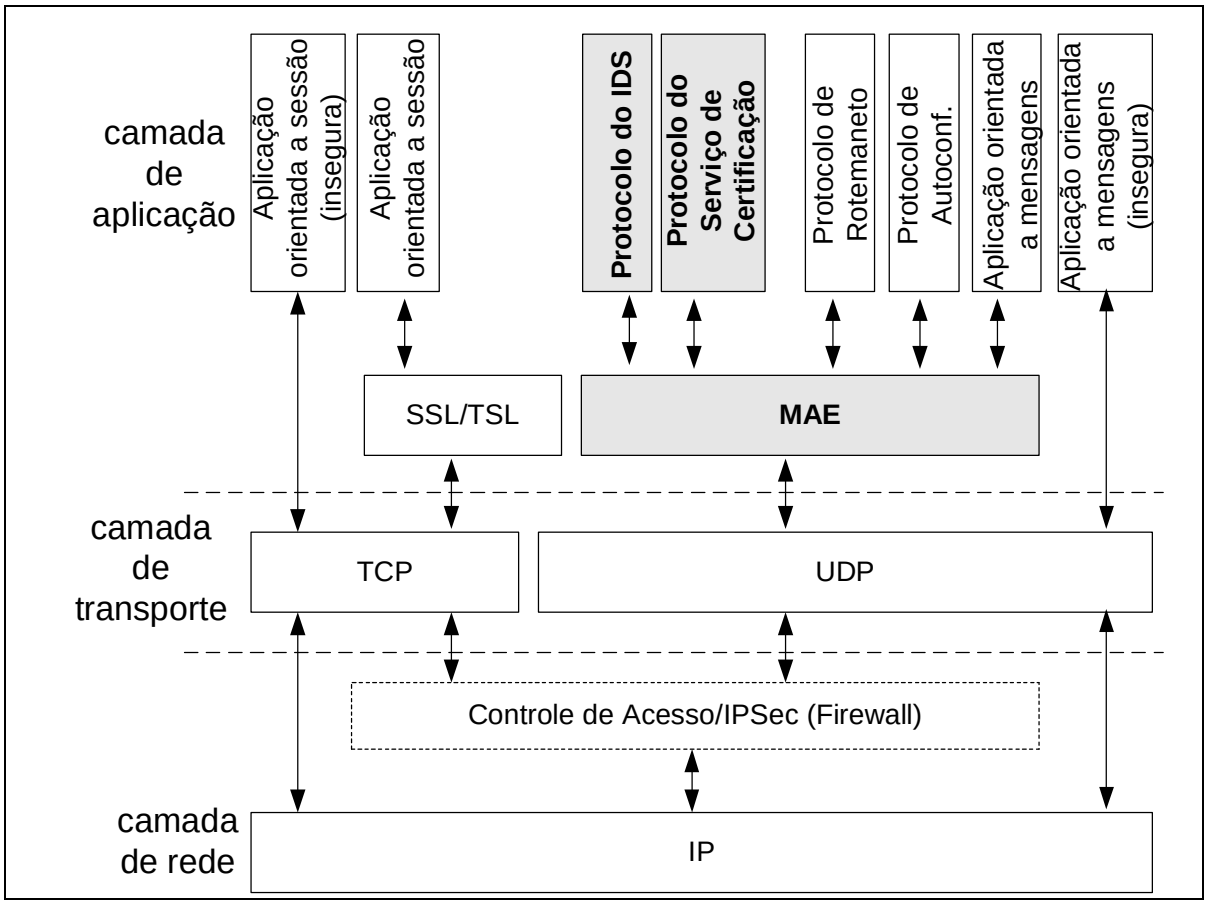


Figura 3-5 – Arquitetura de Protocolos do Modelo de Segurança

Finalmente, ainda que não se trate especificamente destes serviços neste trabalho, pode-se considerar o projeto e desenvolvimento de outros serviços de segurança distribuídos, tais como:

- serviço de *firewall* distribuído (L-Firewall), que provê uma resposta colaborativa para re-configuração dos filtros de pacotes em diversos nodos diferentes com objetivo de mitigar ataques que envolvam tráfego de pacotes atravessando mais de um nodo da rede (e.g. DDoS);
- serviço de gestão da política de segurança distribuída (L-SPM), que tem por principal objetivo estabelecer e distribuir, de maneira segura e cooperativa, atualizações na política de segurança e nas bases de configurações dos serviços de segurança (e.g. parâmetros do serviços de certificação distribuída e da autenticação, bases de dados de assinaturas de ataques, regras para os *firewall*, etc.).

Uma visão de implementação desse modelo de segurança estendido, incluindo os serviços L-Firewall e L-SPM é mostrado Figura 3-6.

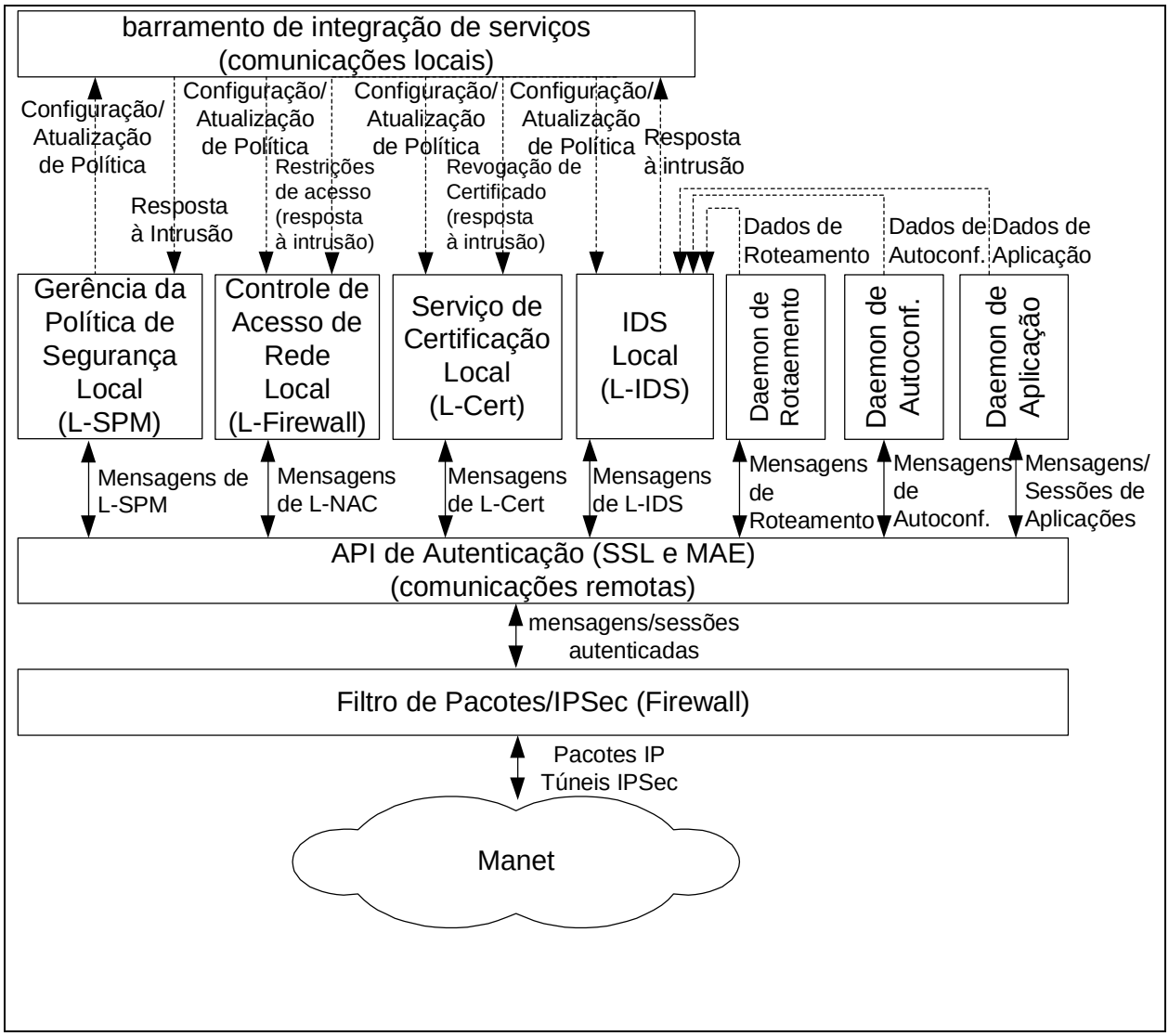


Figura 3-6 – Visão de Implementação do Modelo de Segurança Estendido

4. CERTIFICAÇÃO E AUTENTICAÇÃO EM MANET

Neste capítulo são apresentados os detalhes de projeto dos protocolos e algoritmos de certificação e autenticação adotados no modelo de segurança proposto. Esses mecanismos são, então, aplicados na segurança preventiva dos protocolos de roteamento e autoconfiguração. Em ambos os casos, um conjunto de ataques envolvendo vulnerabilidades relacionadas à modificação, fabricação e/ou personificação de mensagens do protocolo são discutidas e a proteção preventiva contra esses ataques é detalhada.

4.1. SERVIÇO DE CERTIFICAÇÃO EM MANET

O serviço de certificação apresentado nesta seção é uma adaptação de [66,71], que oferece algoritmos e protocolos escaláveis para distribuir de forma segura, os serviços de certificação entre os nodos de uma Manet. O serviço fica disponível para qualquer nodo da rede desde que seja possível localizar uma coalizão com um número mínimo de nodos (K). Os objetivos básicos deste serviço de certificação para Manet são:

- Distribuição e localização: Enquanto a distribuição está ligada à ausência de entidades centralizadas em Manet, a localização está ligada aos requisitos de desempenho e escalabilidade.
- *Bootstrapping* e auto-iniciação: Deve-se minimizar, ou mesmo eliminar, a necessidade de uma entidade centralizada (negociador) para iniciação de novos nodos. Caso um negociador seja requerido, isto deve ocorrer apenas na fase de *bootstrap* da rede.
- Atualização pró-ativa do segredo de certificação: Como um conjunto de K nodos pode realizar todos os serviços de certificação, o comprometimento deste quantitativo de nodos por um mesmo adversário (ou por adversários que colaborem entre si) implica na quebra da segurança do sistema. Com a atualização periódica das partes da chave privada dos nodos, a solução apresenta uma relativa tolerância à intrusão entre cada ciclo de atualização.

A distribuição das funcionalidades de uma autoridade certificadora é baseada no protocolo criptográfico RSA [98] e realizada através do compartilhamento da chave privada da AC entre todos os nodos participantes da rede utilizando a técnica de criptografia de limiar [102] (para maiores detalhes, vide o Anexo II). Considera-se uma rede *ad hoc* em que cada

nodo v_i possui um par de chaves RSA, $\{KI_i , KU_i\}$, onde $KI_i = <d_i , n_i >$ é a chave privada e $KU_i = <e_i , n_i>$ é a chave pública, sendo n_i o módulo da computação RSA.

Igualmente, a ACD do sistema deve possuir um par de chaves RSA, $\{KI_{AC} , KU_{AC}\}$, em que $KI_{AC} = <d_{AC} , n_{AC}>$ é a chave privada de certificação, usada para assinar todos os certificados dos nodos da rede. Qualquer um desses certificados pode ser verificado pela chave pública do sistema (KU_{AC}), que é de conhecimento geral.

De acordo com o modelo de confiança colaborativa K-de-N, introduzido no capítulo anterior, d_{AC} é repartida pelos nodos da rede através do uso da criptografia de limiar. Cada nodo v_i, além de seu par de chaves, possui uma parte de d_{AC} ($KI_{AC,i}$) que deve ser mantida privada, com segurança equivalente à de sua própria chave privada. Qualquer grupo de K nodos, dentre os N portadores de partes de d_{AC}, pode formar uma coalizão de certificação e funcionar como uma autoridade certificadora. Do mesmo modo, não é possível a nenhum nodo isoladamente o conhecimento de KI_{AC}. Nem mesmo uma coalizão de K nodos é capaz de recuperar a chave privada do sistema, a não ser que troquem suas partes de KI_{AC} entre si.

O limiar K representa um compromisso entre a disponibilidade do serviço e a tolerância à intrusão, ou seja, um grupo de adversários precisa destruir $(N – K + 1)$ portadores de partes da chave privada para inviabilizar o serviço (já que impediria uma auto-iniciação) e comprometer, pelo menos, K partes da chave privada para roubar KI_{AC}. Quando da construção do sistema, deve-se fazer uma análise cuidadosa para a escolha de K. Quanto menor o valor K, maior a facilidade de quebra de KI_{AC}. Entretanto, quanto maior K, ao mesmo tempo em que se aumenta a segurança do sistema, diminui-se sua tolerância a falhas. Com a escolha apropriada para K, as coalizões de certificação são estabelecidas dinamicamente na vizinhança *1-hop* para fornecer o serviço de certificação, mantendo a característica de localização da solução.

Os certificados gerados têm a finalidade de certificar, como em um sistema criptográfico comum, as chaves públicas de cada nodo da rede. Assim, no mecanismo de segurança utilizado, cada nodo da rede possui um certificado assinado pela chave secreta KI_{AC} no formato $CERTIFICATE v_i = <v_i , KU_i , T_{sign} , T_{expire} , CERT_i>$, onde:

- v_i é o identificador do nodo;
- KU_i é sua chave pública;
- T_{sign} é um selo de tempo (*timestamp*) com a data e hora do início de validade do certificado; e
- T_{expire} é um selo de tempo com a data e hora da expiração do certificado.
- $CERT_i$ é a assinatura do certificado com KI_{AC}.

Para controlar a validade do certificado, são empregados dois métodos:

- Revogação implícita de certificado: Cada entidade deve renovar o seu certificado a cada período T_{renew}, antes da expiração ($T_{expire} \leq T_{sign} + T_{renew}$).
- Revogação explícita de certificado: Contra-certificados são assinados colaborativamente pela ACD para revogar certificados de nodos comprometidos independentemente do seu tempo de validade. Os contra-certificados possuem o formato $COUNTER\_CERTIFICATE v_i = < \perp v_i , KU_i , T^{\perp}_{sign} , T_{expire} , CERT^{\perp}_i >$, onde $\perp$ denota contra-certificação e $T^{\perp}_{sign}$ é o momento em que o contra-certificado foi criado. O contra-certificado é igualmente assinado com KI_{AC} ($CERT^{\perp}_i$). Os contra-certificados são mantidos localmente por todos os nodos da Manet em uma lista de certificados revogados (CRL). Apenas os contra-certificados para certificados revogados que não expiraram ainda necessitam estar na CRL.

4.1.1. Serviços Básicos de Certificação

4.1.1.1. Emissão e Renovação de Certificado

Um nodo sem certificado ou necessitando renovar seu certificado deve solicitar um novo certificado para outros nodos que ofereçam o serviço de certificação. A política de certificação utilizada deve especificar como os nodos que recebem a requisição de certificados devem servir tal requisição. Diferentes políticas podem ser especificadas para emissão e renovação de certificados, tais como:

- Servir, de acordo com uma política específica de identificação do requerente;
- Servir manualmente, perguntando-se ao usuário do nodo que recebe a solicitação para que este decida se o certificado deve ser servido;
- Negar o serviço explicitamente, gerando uma mensagem de retorno para o nodo requisitante;
- Negar o serviço silenciosamente, descartando a requisição sem enviar nenhum aviso para o nodo requisitante;
- Outras.

Tecnicamente, não há diferença entre os processos de emissão e a renovação de um certificado. Entretanto, as políticas para tais serviços podem ser especificadas separadamente.

Afinal, os critérios para a assinatura de um certificado de uma nova entidade devem ser muito mais cuidadosos do que para a assinatura de um certificado de uma entidade já pertencente à rede que não possui um histórico de comportamento incorreto.

Conforme discutido no capítulo anterior, o serviço é realizado em quatro etapas (Figura 3-2): (1) e (2) formação da coalizão, (3) requisição do serviço, e (4) coleta e processamento das respostas. Essas etapas são descritas a seguir.

(1) Quando um nodo (v_i) necessita receber um certificado, ele dissemina uma mensagem de requisição de coalizão (COALITION_REQ) contendo REF_N, onde REF_N é o número de referência da transação, que é gerado aleatoriamente e deve estar presente em todas as demais mensagens do processo.

(2) Qualquer nodo que possui uma parte de KI_{AC} deve responder à requisição enviando uma mensagem autenticada de notificação de coalizão (COALITION_ACK) ao requerente, contendo sua identidade v_j e REF_N.

(3) O nodo requerente coleta respostas (COALITION_ACK) até que seja possível formar uma coalizão de K nodos[25]. O próprio requerente pode fazer parte da coalizão, caso ele possua uma parte de KI_{AC}. O conjunto de identidades dos nodos da coalizão é dado por $b = \{v_j / v_j \in \text{coalizão}\}$. Em seguida, o nodo dissemina uma mensagem de requisição de certificado (CERT_REQ), contendo a requisição do certificado, $<v_i , KU_i , T_{sign} , T_{expire} , CERT\_REQ_i >$, onde $CERT\_REQ_i$ é a assinatura da requisição do certificado com KU_i, juntamente com b e REF_N. No caso de emissão de um certificado para um nodo que não tenha um certificado válido, as informações de identidade do nodo requeridas pela política de certificação são adicionadas à requisição. Alternativamente, na renovação de um certificado, o certificado ainda válido (i.e. não expirado) é adicionado.

(4) Ao receber CERT_REQ, os nodos que fazem parte da coalizão identificam sua própria identidade na coalizão. Se um certificado válido é enviado junto com a mensagem, a requisição é tratada como uma renovação de certificado. Caso contrário, considera-se uma requisição de emissão de certificado para um nodo não certificado. A política de certificação apropriada é aplicada e cada nodo da coalizão decide se atende à requisição. Em caso afirmativo, o certificado parcial

[25] O nodo solicitante aguarda por mensagens COALITION_ACK durante um período $T_{coalition}$. Caso não seja possível formar uma coalizão com as respostas recebidas neste tempo, uma nova COALITION_REQ é realizada e um processo de *exponential-backoff* é adotado para o período de espera $T_{coalition}$.

$CERT_{i,j}$ é calculado e enviado ao requerente, em uma mensagem PARTIAL_CERT.

O cálculo de $CERT_{i,j}$ é feito da seguinte maneira. Cada nodo v_j que resolve atender a uma requisição de v_i, calcula sua chave aditiva $KI_{j,\beta}$, conforme Eq. 4-1:

$$KI_{j,b} = KI_{CA,j} L_{j,b}(0) = KI_{CA,j} \prod_{r \in b, r \neq j} \frac{v_r}{v_r - v_j} \bmod n_{AC} \quad \text{Eq. 4-1}$$

onde: $L_{j,b}(v_i)$ são os coeficientes de Lagrange para interpolação do polinômio gerador $f(x)$, dados por (Eq. 4-2):

$$L_{j,b}(v_i) = \prod_{r \in b, r \neq j} \frac{v_i - v_r}{v_j - v_r} \quad \text{Eq. 4-2}$$

$KI_{j,\beta}$ é chamada chave aditiva, pois pela interpolação de Lagrange tem-se (Eq. 4-3):

$$\sum_{r \in b} KI_{r,b} = \sum_{r \in b} KI_{AC,r} L_{r,b}(0) = d_{AC} \bmod n_{AC} = t \cdot n_{AC} + d_{AC} \quad \text{Eq. 4-3}$$

onde: $0 \leq t < k$.

Em seguida, v_j calcula o certificado parcial de v_i ($CERT_{i,b}$), assinando o *hash* do novo certificado ($cert_i$) com a chave aditiva $KI_{j,\beta}$ (Eq. 4-4):

$$CERT_{i,j} = (cert)^{KI_{j,b}} \bmod n_{AC} \quad \text{Eq. 4-4}$$

Finalmente, após receber os K certificados parciais, v_i combina-os para gerar um candidato para a assinatura do certificado ($CERT_i$'), conforme a Eq. 4-5:

$$CERT_i' = \prod_{r \in b} CERT_{i,r} \bmod n_{AC} = (cert_i)^{\sum_{r \in b} KI_{i,r}} \bmod n_{AC} = (cert)^{t \cdot n_{AC} + d_{AC}} \bmod n_{AC} \quad \text{Eq. 4-5}$$

Observando que $CERT_i = (cert)^{d_{AC}} \bmod n_{AC}$, verifica-se que este candidato $CERT_i$' é diferente de $CERT_i$ por uma constante. Esta polarização pode ser removida pelo algoritmo abaixo:

Algoritmo 1 – Cálculo de $CERT_i$

Entradas: $CERT_i$' (candidato a assinatura do certificado) e $cert_i$ (*hash* do certificado a ser assinado).

Saída: $CERT_i$ (assinatura do certificado).

1: $Z \coloneqq (cert_i)^{-n_{AC}} \bmod n_{AC}$
2: $r \coloneqq 0, Y \coloneqq CERT_i'$
3: *while* $j < K$ *do*
4: $Y \coloneqq Y \cdot Z \bmod n_{AC};\ j \coloneqq j + 1$
5: *if* $(cert_i = Y^{e_{AC}} \bmod n_{AC})$ *then*
6: *break while*
7: *end if*
8: *end while*
9: *saída*: $Y = CERT_i$

Caso ocorra uma falha em algum dos nodos da coalizão e um dos certificados parciais não seja recebido, os certificados parciais dos outros nodos tornam-se sem utilidade, sendo necessário reiniciar todo o processo. Para que esse processo seja executado somente através de comunicações *1-hop*, necessita-se de que o nodo requisitante tenha pelo menos *K* nodos vizinhos.

Em [71] é apresentado uma procedimento alternativo que eliminaria a necessidade de se fazer o estabelecimento explícito da coalizão. Neste caso, ao invés de usar chave de cifração $KI_{j,\beta}$ na Eq. 4-4, utiliza-se diretamente $KI_{AC,j}$. Isso evita os passos (1) e (2). Entretanto, no desenvolvimento deste trabalho foi constatada uma falha nesta abordagem, pois não há garantias matemáticas de que o algoritmo para o cálculo de $CERT_i$ retorne um valor correto em no máximo *K* iterações. De fato, para se obter o valor correto para $CERT_i$ neste caso, o número de iterações do referido algoritmo é potencialmente elevado, o que torna seu uso computacionalmente inviável.

4.1.1.2. Revogação de Certificado

Se de acordo com a política de segurança da rede o certificado de um nodo v_i for considerado comprometido, este deve ser revogado. A revogação de um certificado ocorre com a assinatura de um contra-certificado para este nodo. Quando um contra-certificado é assinado ($CERT^{\perp}_{i}$), ele deve ser disseminado (*flooding*) em toda a rede e adicionado na CRL local de todos o nodos. Este contra-certificado é mantido na CRL até que chega o momento T_{expire}. Dependerá da política de re-emissão de certificados definir se um nodo que tem seu certificado revogado terá um certificado re-emitido após o tempo $T^{\perp}_{sign}$.

A revogação de certificados ocorre de maneira análoga à emissão de novos certificados. Inicialmente é necessário se formar uma coalizão de *K* nodos (mensagens COALITION_REQ e COALITION_ACK). Em seguida, é gerada uma requisição de revogação de certificado (CERT_REVOKE_REQ), que contém o certificado a ser revogado, juntamente com os dados da coalizão (*b*). Finalmente, os nodos da coalizão assinam o contra-certificado parcial ($CERT^{\perp}_{i,j}$) com sua chave aditiva $KI_{j,\beta}$ (Eq. 4-1), conforme mostrado na equação Eq. 4-4. Os contra-certificados parciais são devolvidos ao requisitante (CERT_REVOKE_ACK), que recupera o contra-certificado solicitado, de maneira análoga ao processo de recuperação de certificados mostrado na Eq. 4-5 e no Algoritmo 1, e o dissemina na rede.

A revogação de certificados pode ser solicitada em dois casos: auto-revogação e revogação por intrusão detectada. A auto-revogação é feita quando um nodo decide revogar seu próprio certificado (e.g. por considerar que sua chave privada tornou-se insegura). Neste caso, o próprio nodo assina a solicitação da revogação de seu certificado com a sua chave privada, sendo esta imediatamente aceita pelos nodos da coalizão, que podem verificar a assinatura da requisição. No caso da revogação por detecção de intrusão, conforme discutido no capítulo anterior, é necessário que um dos nodos da Manet (requerente) colete, pelo menos, *K* acusações assinadas (alertas) por nodos diferentes contra o nodo comprometido. Este nodo forma uma coalizão com demais nodos que geram acusações. Após verificar as acusações, os participantes da coalizão assinam o contra-certificado e o enviam de volta ao requerente.

4.1.1.3. Validação de Certificado

A validação de certificados deve ocorrer sempre que uma mensagem assinada por um certificado que é visto pela primeira vez por um dos nodos da Manet é recebida. De uma maneira geral, a verificação local da validade de um certificado envolve:

(1) verificação das datas de emissão (T_{sign}) e expiração (T_{expire});

(2) consulta à CRL local para checar se o certificado não foi revogado explicitamente; e

(3) verificação da assinatura do emissor.

Para a verificação da assinatura do emissor (passo 3), é necessária a construção do caminho de certificação entre o certificado que está sendo validado e o certificado do emissor que o assina (usualmente, a ACD da rede). Os certificados das ACs confiáveis são mantidos em um *cache* de certificados válidos. Caso o emissor seja uma AC confiável, a verificação da assinatura é feita extraindo-se a chave pública do emissor de seu certificado (usualmente KI_{AC}). Caso o certificado que está sendo validado tenha sido assinado por uma AC que não é diretamente confiada pelo nodo que faz a validação, realiza-se a construção do caminho de certificação recursivamente, até se encontrar um certificado de alguma AC no caminho de certificação que seja confiável.

Como o processo de validação acarreta um custo computacional não desprezível, é conveniente que os certificados que tenham sido validados recentemente sejam mantidos em uma *cache* local de certificados válidos para consultas posteriores. Obviamente, certificados revogados ou expirados devem ser removidos da *cache* local de certificados válidos.

4.1.2. Iniciação do Sistema (*Bootstrap*) com um Negociador

Para que os serviços de certificação estejam disponíveis, é necessário que exista na Manet, pelo menos, K nodos portadores de partes de KI_{AC}. A maneira convencional de se iniciar esses primeiros nodos implica no envolvimento de uma entidade centralizada no momento da formação da rede, denominado negociador (*dealer*). O negociador gera o par de chaves RSA $\{KI_i , KU_i\}$ e um polinômio randômico $f(x)$ de grau $K - 1$ (Eq. 4-6):

$$f(x) = d_{AC} + a_1 x + \ldots + a_{K-1} x^{K-1} \qquad \text{Eq. 4-6}$$

onde:

$f(0) = d_{AC}$; e os coeficientes $a_1, \ldots, a_{K-1}$ são gerados aleatoriamente ($a_1, \ldots, a_{K-1} < n_{AC}$).

As chaves parciais das entidades v_i são dadas por (Eq. 4-7):

$$KI_{AC,i} = (f(v_i) \bmod n_{AC}) \qquad \text{Eq. 4-7}$$

Cada um dos K nodos iniciais da rede recebe uma chave parcial $KI_{AC,i}$ e, com isso a rede já pode funcionar. O negociador distribui ainda um conjunto de valores usados na verificação dos certificados parciais, denominados testemunhas do polinômio $f(x)$, isto é, $\{g^{d_{AC}}, g^{a_1}, ..., g^{a_{K-1}}\}$, onde g é conhecido por toda a rede, assim como a chave pública KU_{AC}. Após a rede ser iniciada, nem o nodo negociador e nem o polinômio são necessários mais para o funcionamento do sistema e são descartados. A partir deste momento, os nodos já iniciados são responsáveis pela emissão de certificados e de partes de KI_{AC} para nodos novos.

4.1.3. Emissão e Atualização Pró-ativa de Partes da Chave Privada

4.1.3.1. Emissão de Partes da Chave Privada

Todo nodo v_i que possuir um certificado válido na rede pode adquirir uma parte de KI_{AC} (e.g. $KI_{AC,i}$). Por questão de segurança e robustez do sistema, o nodo negociador do sistema que tem apenas a função de iniciação não permanece conectado à rede quando esta entra em funcionamento. Assim, um mecanismo de emissão auto-organizada de partes da chave privada de certificação é necessário. A idéia é que as entidades já iniciadas (e.g. que possuam uma parte da chave privada) iniciem as novas entidades, que, por sua vez, também poderão participar do processo de iniciação de novos nodos. Elimina-se, com isso, a necessidade de um nodo central no sistema. Esse processo é apresentado nessa seção.

Para a geração de uma nova parte da chave privada ($KI_{AC,i}$) o nodo v_i deve escolher uma coalizão de K nodos $b = \{v_j / v_j \in \text{coalizão}\}$ e enviar a eles uma requisição de parte da chave privada (SECRET_SHARE_REQ), juntamente com b e seu certificado válido. Quando v_j recebe a requisição, ele confere o certificado de v_i e a lista de revogações CRL. Se ele decidir servir a requisição, calcula sua chave parcial $P_{j,b}$ para v_i (Eq. 4-8).

$$P_{j,b} = KI_{AC,j} L_{j,b}(v_i) \bmod n_{AC} \qquad \text{Eq. 4-8}$$

A partir das chaves parciais de cada um dos K nodos, v_i pode construir $KI_{AC,i}$ por interpolação de Lagrange Eq. 4-9:

$$KI_{AC,i} = f(v_i) = \sum_{r \in b} KI_{AC,r} L_{r,b}(v_i) = \sum_{r \in b} P_{r,b} \bmod n_{AC} \qquad \text{Eq. 4-9}$$

Porém, não é seguro que os nodos mandem essas chaves parciais diretamente, pois $L_{j,b}(v_i)$ depende apenas dos IDs da coalizão e o nodo v_i pode facilmente descobrir a parte da chave privada de v_j ($KI_{AC,j}$) a partir da chave parcial $P_{j,b}$. Se v_i receber K chaves parciais quaisquer (e.g. $P_{1,b}$, ..., $P_{K,b}$) ele pode descobrir as chaves parciais da coalizão e recuperar a chave privada do sistema (KI_{AC}).

Para se resolver este problema, v_i deverá receber apenas o somatório das chaves $\{P_1, P_2, \ldots, P_k\}$. Para isso, é adotado um esquema de embaralhamento, com os seguintes passos:

(1) Para a geração de uma nova parte da chave privada, $KI_{AC,i}$, o nodo v_i escolhe uma coalizão de K nodos, disseminando uma mensagem de requisição de coalizão (COALITION_REQ) contendo REF_N, juntamente com seu certificado válido.

(2) Qualquer nodo que possui uma parte de KI_{AC} pode responder à requisição. A política de certificação é aplicada e cada nodo da coalizão decide se atende à requisição. Em caso afirmativo, envia-se uma mensagem autenticada de notificação de coalizão (COALITION_ACK) ao requerente, contendo sua identidade v_j e REF_N.

(3) O nodo requerente coleta respostas (COALITION_ACK) até que seja possível formar uma coalizão $b = \{v_j / v_j \in \text{coalizão}\}$ de K nodos. Em seguida, o nodo dissemina uma mensagem de requisição de parte da chave privada (SECRET_SHARE_REQ1), contendo b e REF_N.

(4) Ao receber SECRET_SHARE_REQ1, os nodos que fazem parte da coalizão identificam sua própria identidade na coalizão. Cada nodo v_j gera um fator randômico $d_{r,j}$ para cada membro v_r de b cujo o ID é menor do que o seu próprio. Cada fator $d_{r,j}$ é cifrado com a chave KU_r do membro que irá recebê-la e enviado para o nodo requerente (v_i) em mensagens SECRET_SHARE_ACK.

(5) O nodo requerente encaminha os fatores $d_{r,j}$ cifrados para seus destinatários (SECRET_SHARE_REQ2).

(6) Cada v_j, após decifrar todos os fatores recebidos, calcula uma chave parcial embaralhada ($\overline{P_{j,b}}$), subtraindo de sua chave parcial $P_{j,b}$ todos os fatores recebidos (i.e. enviados por nodos com ID maior que o seu) e soma a esta mesma chave os fatores enviados por ele aos nodos de ID menor que o seu próprio. Por fim, envia o resultado de volta para v_i (PARTIAL_SECRET_SHARE).

O cálculo de $\overline{P_{j,b}}$ é mostrado na Eq. 4-10.

$$\overline{P_{j.b}} = P_{j,b} + \sum_{r \in b, r \neq j} sign(v_r - v_j) d_{r,j} \qquad \text{Eq. 4-10}$$

onde: $sign(v_r - v_j) = 1$ se $(v_r - v_j) > 0$ e $sign(v_r - v_j) = -1$ se $(v_r - v_j) < 0$.

Depois que recebe as chaves parciais embaralhadas de toda a coalizão, o nodo v_i calcula sua chave secreta $KI_{AC,i}$ como mostrado na Eq. 4-11.

$$KI_{AC,i} = \sum_{r \in b} \overline{Pj, b} = \sum_{s \in b} \left(P_{j,b} + \sum_{s \in b, s \neq j}^{k} sign(v_r - v_j) d_{r,j} \right) \qquad \text{Eq. 4-11}$$

$$= \sum_{r \in b} P_{j,b} + \sum_{r \in b} \sum_{s \in b, s \neq j} sign(v_r - v_j) d_{r,j}$$

$$= \sum_{r \in b} P_{j,b}$$

A emissão de partes da chave privada é mostrada na Figura 4-1.

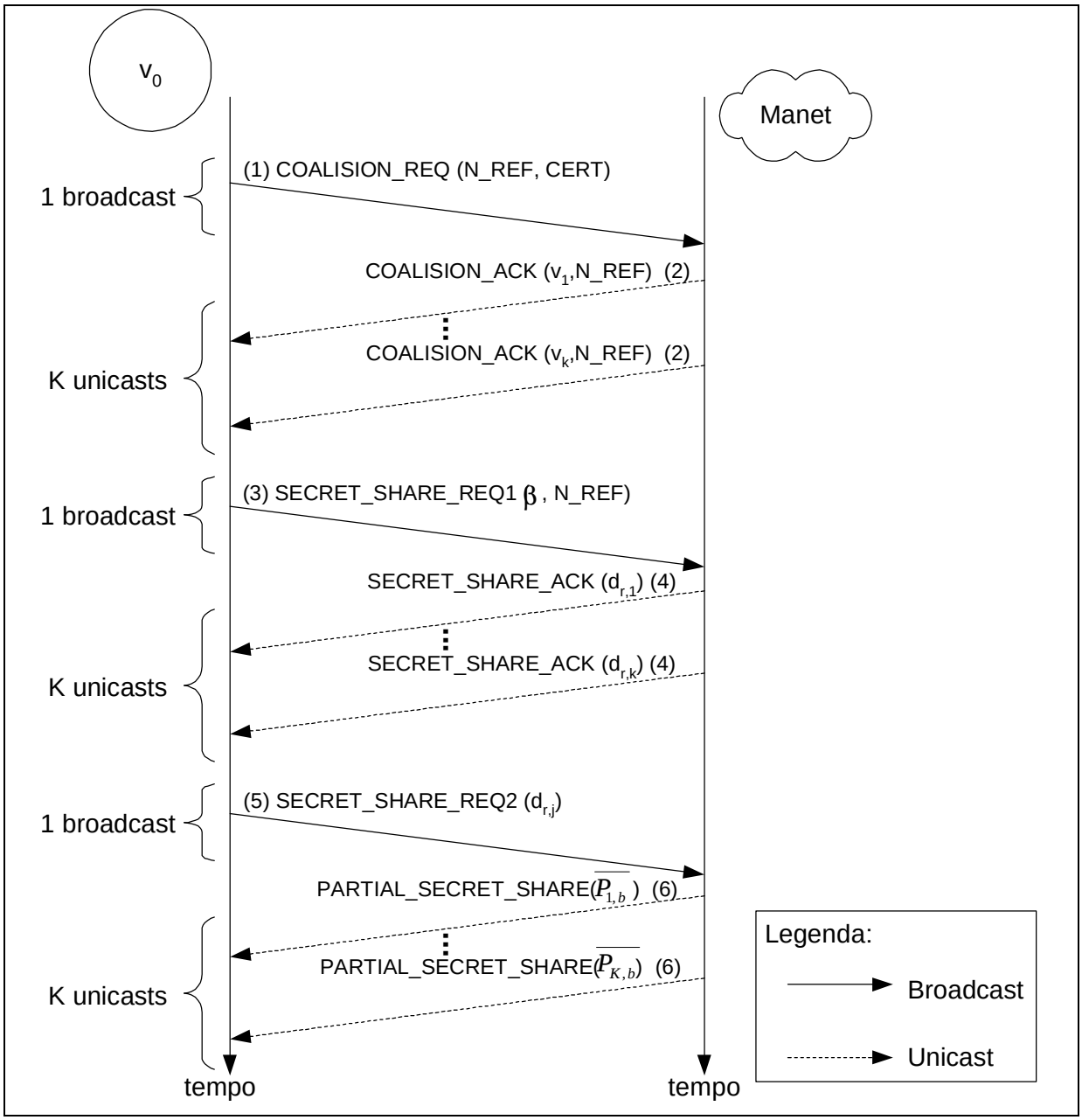


Figura 4-1 – Emissão de Partes da Chave Privada

4.1.3.2. Atualização Pró-ativa de Partes da Chave Privada

Existem duas formas de proteger o sistema da quebra do seu segredo no decorrer do tempo: trocar as chaves do sistema $\{KI_{AC}, KU_{AC}\}$ de tempos em tempos ou atualizar todas as partes da chave privada periodicamente. A primeira solução pode ser realizada reiniciando o sistema periodicamente, entretanto, esta não é a melhor opção, já que toda a rede deverá ser

atualizada ao mesmo tempo, tanto no que diz respeito a seus certificados como no que diz respeito às suas chaves parciais.

Pela segunda solução, todas as partes da chave privada são atualizadas periodicamente. Assim, o sistema tolera até $K - 1$ nodos comprometidos entre essas atualizações, já que KI_{AC} só pode ser obtido com K partes da chave privada. Em [71] são propostas duas alternativas para se conseguir uma atualização eficiente e escalável das partes da chave privada. A primeira é um simples processo seqüencial baseado na emissão de partes da chave privada apresentada no item anterior. Inicialmente, uma coalizão de K nodos atualiza suas próprias partes de chaves parciais aplicando os protocolos propostos em [44]. Então, o protocolo de emissão de partes da chave privada atualiza os demais nodos da rede. A segunda aproximação caracteriza-se por atualização paralela das chaves parciais de toda a rede por convergência rápida.

Neste segundo caso, o tempo é dividido em períodos (T_{update}) e cada um destes é composto por uma fase de atualização das chaves parciais e por uma fase operacional. Durante a fase operacional, os nodos renovam seus certificados periodicamente.

No começo da fase de atualização, uma coalizão escolhida de K nodos do sistema gera um polinômio de atualização $f_u(x) = f_{u,1}x + ... + f_{u,K-1}x^{K-1}$ onde $f_u(0) = 0$. f_u é então cifrado com KU_{AC} e assinado pela coalizão com suas partes de KI_{AC}. Com isso, um adversário é impedido de descobrir f_u e de simular uma coalizão para fazer uma falsa atualização de chaves parciais.

O polinômio cifrado, juntamente com sua assinatura, é propagado por toda a rede. Assim que for recebido por um nodo, este verifica a assinatura com KU_{AC} e solicita o serviço de atualização de chaves parciais a K nodos, que não precisam ter atualizado suas chaves para prestar o serviço. Os K nodos calculam $U_i = f_u(v_i)$ e enviam a v_i, que simplesmente soma este valor com a sua parte da chave privada atual $KI_{AC,i}$ para gerar a nova chave $KI_{AC,i}$' (Eq. 4-12). A parte da chave privada antiga será conservada durante a transição e eliminada após a fase de atualização.

$$KI_{AC,i}' = KI_{AC,i} + U_i = f(v_i) + f_u(v_i) = f_{novo}(v_i) \qquad \text{Eq. 4-12}$$

Note que, definindo-se $f_{novo} = f + f_u$, tem-se $f_{novo}(0) = f(0) + f_u(0) = d_{AC}$. Com isso, o polinômio é atualizado e KI_{AC} permanece inalterada.

4.1.4. Verificação de Chaves e Certificados

Quando um nodo v_i é iniciado, seja pelo negociador, seja pela rede, ele verifica a validade de sua parte da chave privada $KI_{AC,i}$ testando a igualdade mostrada na equação Eq. 4-13:

$$g^{KI_{AC,i}} = g^{d_{AC}} \cdot (g^{a_1})^{v_i} \cdot (g^{a_2})^{v_i^2} \cdot \mathbf{K} \cdot (g^{a_{K-1}})^{v_i^{K-1}} \quad \text{Eq. 4-13}$$

onde: $\{g^{d_{AC}}, g^{a_1}, ..., g^{a_{K-1}}\}$ são as testemunhas do polinômio $f(x) = d_{AC} + a_1 x + ... + a_{K-1} x^{K-1}$ e g é um valor conhecido por toda a rede.

No caso da emissão colaborativa de partes da chave privada, se a igualdade da Eq. 4-13 não for verificada, v_i saberá que a parte calculada não é válida. Isso pode ser resultado de um nodo corrompido controlado por um adversário ou de um nodo que cometeu um erro. Entretanto, v_i não tem como identificar qual das chave parciais $P_{j,b}$ está corrompida. Para solucionar o problema, quando a coalizão troca os fatores embaralhadores $d_{r,j}$, eles também enviam a v_i uma testemunha $g^{d_{r,j}}$. Para cada $\overline{P_{j,b}}$ embaralhado, v_i verifica a igualdade da equação Eq. 4-14:

$$g^{KI_{CA,j}} = \prod_{r \in b, r \neq j} (g^{d_{r,j}})^{sign(v_r - v_j)} \quad \text{Eq. 4-14}$$

onde: $g^{KI_{CA,j}} = g^{d_{CA}} \cdot \left(g^{a_1}\right)^{v_j} \cdot \left(g^{a_2}\right)^{v_j^2} \cdot ... \cdot \left(g^{a_{K-1}}\right)^{v_j^{K-1}}$ é a testemunha da parte da chave privada de v_j.

Caso a Eq. 4-14 não seja verificada, $\overline{P_{j,b}}$ é defeituoso. Então, v_i marca v_j como comprometido e envia a acusação e todas as provas contra v_j para a rede.

No caso da verificação de certificados, v_j prepara $\{A_1, A_2, r\}$ como prova para um nodo v_i da validade de seu certificado parcial $CERT_{i,j}$. Para isso, v_j gera um valor randômico u e calcula $A_1 = g^u$ e $A_2 = (cert_i)^u$. Então, v_j calcula $c = hash(g^{KI_{AC,j}} \mid CERT_{i,j} \mid A_1 \mid A_2)$ e $r = u - cKI_{AC,j}$. Então, $\{A_1, A_2, r\}$ é assinado por v_j e apresentado a v_i. Este, por sua vez, sendo o destinatário de $\{A_1, A_2, r\}$ ou apenas um nodo vizinho fazendo um monitoramento do

comportamento de v_j, calcula c e verifica se $A_1 = g^r \cdot (g^{KI_{CA,j}})^c$ e $A_2 = (cert_i)^r \cdot (CERT_{i,j})^c$. Se todas as respostas corresponderem, o certificado é válido.

4.1.5. Base Local de Dados de Certificação

Alguns dados de certificação devem ser mantidos permanentemente em cada nodo v_i que participa do serviço de certificação (L-Cert). Esses dados incluem: a chave privada do nodo (KI_i) e seu próprio certificado ($CERTIFICATEv_i$), certificados válidos de cada uma das ACs confiáveis, a sua parte da chave privada de certificação ($KI_{AC,i}$) com a versão atual do compartilhamento de chave, um polinômio cifrado com KU_{AC} e sua respectiva versão de atualização do compartilhamento.

Adicionalmente, os L-Certs mantêm dinamicamente uma tabela com os certificados validados recentemente (*cache* de certificados válidos) e uma tabela com os certificados revogados (CRL local). Registros nestas tabelas devem ser automaticamente removidos quando expirados (i.e. tempo atual = T_{expire}). Os registros da tabela de certificados válidos são mantidos durante um período de tempo máximo (T_{out}), configurável de acordo com a política de certificação. Quando um certificado nesta tabela é consultado, o momento de expurgo do registro da tabela é alterado para o *min*(tempo atual + T_{out}, T_{expire}). Essa tabela tem ainda um tamanho máximo (N_{CACHE}), que também é configurável de acordo com a política de certificação. Quando esse número máximo é atingido, os registros com momento de expurgo mais próximo são removidos da tabela para liberar espaço para novos registros na tabela.

Neste trabalho, a construção, gerenciamento e sincronização da *cache* de certificados válidos e a CRL local são configurados dependendo do tipo de protocolo de roteamento utilizado. As opções para o gerenciamento das bases de certificados são a construção pró-ativa ou sob-demanda das tabelas.

Na distribuição pró-ativa de certificados válidos, o remetente deve anexar seu certificado às mensagens transmitidas. Se os certificados são transmitidos em todas as mensagens, tem-se a disponibilidade máxima dos certificados. Entretanto, esta abordagem implica em *overhead* considerável na rede. Uma otimização possível consiste em anexar o certificado apenas em algumas mensagens transmitidas. Alternativamente, os nodos podem explicitamente requisitar o certificado de outros nodos, quando estes são necessários e não estão disponíveis na *cache* de certificados válidos. A exemplo do que ocorre com protocolos de roteamento sob-demanda, o proprietário do certificado requisitado não é o único que pode

servir a esta requisição. Outros nodos que possuam o certificado requisitado em sua *cache* podem responder à requisição.

Uma estratégia de sincronização para a CRL local é necessária, permitindo que nodos ambulantes ou recém-chegados tenham sua CRL local atualizada. Novamente, as abordagens pró-ativas e reativas são possíveis. Na abordagem pró-ativa, cada nodo troca periodicamente (e.g. a cada T_{CRL_SYNC}) as informações em sua CRL com seus vizinhos para que todos possuam a lista inteira de certificados revogados da rede. Na abordagem reativa, um nodo explicitamente requisita a seus vizinhos uma cópia atualizada da CRL.

4.1.6. Iniciação de um Novo Nodo antes da Autoconfiguração e Roteamento

A emissão e renovação dos certificados necessitam de que pelo menos K nodos estejam disponíveis e funcionem colaborativamente como autoridade certificadora. Todo nodo que não possui um certificado, passa por esse procedimento para obter um certificado válido e assim usufruir dos serviços básicos da rede (e.g. autoconfiguração e roteamento). Entretanto, nesse procedimento, é assumido que o nodo já tenha um endereço IP para se comunicar com os demais nodos da rede e assim obter um certificado. Essa hipótese não se verifica, se o nodo necessitar do serviço de autoconfiguração para obter o seu endereço IP, pois este serviço só estará disponível após a autenticação. Duas soluções são propostas para solucionar este problema.

Considerando que as mensagens necessárias para a emissão do certificado do novo nodo são transmitidas apenas na vizinhança (1-*hop*) do nodo solicitante, uma solução simples consiste no nodo solicitante escolher randomicamente um endereço IP temporário e realizar o procedimento de solicitação do certificado usando este endereço. Os nodos que servem a requisição enviam as mensagens de resposta ao requerente com o TTL do pacote IP configurado com o valor 1 (um), assegurando que o pacote que contém a resposta não seja encaminhado além dos nodos vizinhos. O nodo solicitante coleta todas as mensagens de resposta dos seus nodos vizinhos, mesmo que estas não contenham seu endereço MAC. Isso é possível, pois existe a possibilidade do endereço escolhido estar duplicado. Depois de recuperar o certificado, o solicitante inicia o processo de autoconfiguração. O endereço IP temporário é, então, descartado e não é mais usado por esse nodo.

A alocação de endereços IP temporários duplicados é possível, já que a escolha é feita randomicamente. Entretanto, como os nodos não iniciados enviam a mensagem de solicitação do serviço de certificação apenas para seus nodos vizinhos, o problema só ocorre se endereços

IP duplicados estão na mesma vizinhança do nodo solicitante. Isso pode ser evitado adotando algum tipo de protocolo de roteamento que utilize mensagens *HELLO*. O solicitante monitora essas mensagens e identifica todos os endereços IP seus nodos vizinhos. Outra solução pode ser a utilização de um escopo especial de endereços IP para alocação temporária aos nodos, reduzindo a probabilidade da alocação duplicada de endereços IP quando mais de um nodo não iniciado solicitar o serviço de certificação ao mesmo tempo, utilizando o mesmo endereço IP.

Caso não haja nodos suficientes na vizinhança (i.e ao menos *K* nodos), o serviço de certificação não pode ser realizado exclusivamente com comunicações na vizinhança 1-*hop*. Neste caso, o nodo solicitante poderia incrementar o TTL das mensagens do serviço de certificação progressivamente e utilizar *flooding* para disseminar suas requisições, ao invés de *broadcast*. Entretanto, como as respostas dos serviços de certificação são encaminhadas em *unicast* para o solicitante, este só poderá recebê-las caso já tenha um endereço IP configurado e esteja participando do serviço de roteamento, uma vez que ele se encontra a mais de 1-*hop* de distância dos nodos servidores.

Para nodos recém-chegados à Manet, estes não terão acesso aos serviços de autoconfiguração e roteamento antes de obterem um certificado válido. Adicionalmente, a estratégia de escolher de um endereço IP aleatório não é suficiente, pois um nodo sem certificado continua sem acesso ao serviço de roteamento. Uma solução simples consiste em requerer que os nodos que encaminham as mensagens de requisição do solicitante sirvam como procuradores para ele quando as respostas forem transmitidas em suas vizinhanças. Obviamente, esse requisito precisa ser aplicado apenas para as mensagens de emissão de um novo certificado e se a política de certificação permitir que essas mensagens extrapolem a vizinhança de 1-hop.

4.1.7. Utilização com Múltiplas ACDs

Quando duas ou mais Manets que foram iniciadas por processos distintos se juntam para formar uma única rede, há uma autoridade certificadora distribuída – com sua respectiva chave privada de certificação – para cada uma delas. Desse modo, os certificados dos nodos oriundos de Manets diferentes não são verificáveis entre si, pois são assinados com chaves privadas diferentes. Isto é, os nodos da nova rede ficam com certificados diferentes, vindos de diferentes ACDs. Uma solução simplista para esse problema consiste em exigir que todos os nodos oriundos de uma dessas redes adquiram um certificado emitido pela ACD da outra. Entretanto, essa abordagem acarreta um *overhead* excessivo de comunicação, pois todos os

nodos que desejam conquistar o novo certificado devem fazer uma requisição de certificação. Uma solução alternativa consiste em estabelecer uma nova relação de confiança, não entre os nodos individualmente, mas entre as redes que se juntam. Para tanto, propomos o estabelecimento de uma relação de confiança cruzada entre as autoridades certificadoras distribuídas.

Sem perda de generalidade, apresenta-se o estabelecimento da relação de confiança cruzada entre nodos de duas ACDs diferentes (e.g. ACD_1 e ACD_2). Considera-se que os números de nodos da coalizão dinâmica de ACD_1 e ACD_2 sejam, respectivamente, K_1 e K_2. Um dos nodos provenientes ACD_1 deve fazer uma requisição para os nodos de ACD_2, solicitando que estes assinem o certificado de ACD_1. É requerido que, pelo menos, K_2 nodos provenientes de ACD_2 aceitem confiar na rede de ACD_1, formando uma coalizão dinâmica para assinar o certificado de ACD requerido. O nodo responsável pela requisição do certificado de confiança cruzada recebe os certificados parciais de ACD_1, assinados com partes da chave privada de ACD_2, e reconstitui o novo certificado de ACD_1, assinado por ACD_2. Finalmente, o requisitante deve fazer a disseminação do novo certificado e dos contra-certificados contidos em sua CRL local em toda a rede, completando a formação da confiança cruzada. Do mesmo modo, K_1 nodos provenientes de ACD_1 devem assinar um certificado para ACD_2, usando a chave privada de certificação de ACD_1.

Nota-se que estes certificados de confiança cruzada têm propriedades e objetivos diferentes de certificados assinados para nodos individuais, pois a confiança estabelecida reflete-se para todos os nodos provenientes de uma mesma ACD. Assim, o estabelecimento dessa relação também deve ser controlado por políticas de certificação própria.

4.2. EXTENSÃO DE AUTENTICAÇÃO PARA MANET (MAE)

A extensão de autenticação para Manet (MAE), discutida nessa seção, consiste em um protocolo para autenticação de aplicações orientadas a mensagens, onde as comunicações não seguem um modelo cliente-servidor clássico. Em especial, este é o caso dos protocolos de roteamento e autoconfiguração para Manet. MAE é igualmente usada para autenticar mensagens dos serviços de segurança, tais como L-Cert e L-IDS.

MAE é anexada às mensagens dos protocolos autenticados. Esta MAE deve conter todas as informações necessárias para assegurar corretamente a autenticidade e a integridade dessas mensagens, garantindo a proteção contra ataques de fabricação, modificação e personificação. O objetivo consiste em projetar a MAE de maneira flexível e adaptável, para

que ela possa ser usada na securização de diferentes protocolos e aplicações de Manet. Em especial, discutimos seu uso na securização dos protocolos de roteamento AODV, OLSR, TBRPF e DSR e no protocolo de autoconfiguração DCDP. Em especial, no que diz respeito aos protocolos de roteamento – que já se encontram em uma fase madura de padronização, a idéia é permitir que a MAE possa ser usada sem necessidade de alteração das mensagens do protocolo que se deseja proteger, i.e. preservando a sintaxe e seqüência das mensagens sem modificações. Essa abordagem difere de outros trabalhos relacionados [e.g,28,41].

Objetos de Autenticação: O projeto da MAE segue a tendência moderna de se construir protocolos extensíveis pela adição sucessiva de objetos com finalidades específicas nas mensagens, como é o caso do IPv6, entre outros. Assim, a MAE é formada por um conjunto de objetos de autenticação, que provêm os serviços de autenticação adaptados às necessidades dos diferentes protocolos que se deseja securizar. Como a finalidade básica é prover a autenticação, uma MAE contém sempre um objeto de autenticação obrigatório, que carrega alternativamente uma assinatura digital (DS)[26] ou um código de autenticação de mensagem (MAC)[27]. A escolha do tipo de objeto de autenticação obrigatório depende da política de autenticação escolhida para a Manet.

Assinaturas digitais são usadas juntamente com certificados digitais para prover autenticação com uso de criptografia assimétrica. Esse sistema de autenticação é completamente integrado com o modelo de confiança e os serviços de certificação propostos neste trabalho. Uma assinatura digital é simplesmente computada pela cifração do resumo (*hash*) da mensagem protegida com a chave privada do originador da mensagem. Essa alternativa provê, além dos serviços de autenticação da origem e controle de integridade das mensagens, o serviço de não-repúdio – fundamental para os serviços de proteção corretiva do modelo de segurança proposto.

Por outro lado, o uso de MAC possibilita a aplicação de MAE em ambientes onde são adotados mecanismos de autenticação com chaves criptográficas compartilhadas e primitivas de criptografia simétrica. Um MAC é calculado como o resumo (*hash*) da mensagem protegida concatenada com a chave secreta de autenticação. O caso mais simples consiste no uso de uma única chave secreta (chave de grupo), compartilhada por todos os nodos da Manet. Note que, este sistema não é resistente a intrusos, uma vez que o comprometimento de um único nodo acarreta na exposição do segredo usado por todos. Do mesmo modo, ainda

[26] Do inglês, *digital signature.*

[27] Do inglês, *message authentication code*.

que seja possível detectar ataques neste cenário, como não há não-repudiação, a remoção de nodos comprometidos do sistema por meio de revogação da chave criptográfica é inviável. Entretanto, as vantagens oferecidas pelo uso de criptografia simétrica podem compensar o uso de tal tipo de sistema de autenticação, como é o caso dos trabalhos em [48,49]. Finalmente, existem alternativas para distribuição de chaves criptográficas especialmente adaptadas para o contexto Manet, como por exemplo, o uso do protocolo TESLA e o uso de informações da mobilidade dos nodos para derivação das chaves de autenticação, que torna o uso da criptografia simétrica ainda mais promissor nestes cenários. Entretanto, esses mecanismos não permitem o não-repúdio das mensagens, dificultando a eliminação de nodos comprometidos do sistema. Por isso, esse sistema de autenticação tem aplicação apenas em cenários onde não se admite a necessidade de eliminação pró-ativa de nodos comprometidos.

O objeto de certificação obrigatório (i.e. DS ou MAC) é utilizado para autenticar todos os campos não mutáveis de uma mensagem, incluindo os campos da própria MAE. Como regra geral, os dados mutáveis das mensagens são zerados para o cálculo do objeto de autenticação obrigatório.

Outros objetos de certificação são utilizados para prover serviços adicionais. As opções correntemente definidas são[93]:

Certificados (CERT): Objetos de certificado são usados para carregar, juntamente com um objeto DS, o certificado do signatário da mensagem e, opcionalmente, os demais certificados necessários para se estabelecer o caminho de certificação desse certificado.

Cadeias de *Hash* (HC): Um desafio comum na autenticação de protocolos de roteamento consiste da existência de campos mutáveis nas mensagens dos protocolos, i.e. campos que são alterados por nodos intermediários entre a origem e o destino [28,41]. Este é o caso dos protocolos DSR e AODV, que possuem mensagens que são alteradas progressivamente na medida em que são encaminhadas por nodos intermediários [88,58]. Alguns métodos para proteger campos mutáveis típicos desses protocolos (e.g. *hop count* e traçados de endereços IP) com uso de cadeias de *hash* são apresentados em [50]. Essa estratégia é adotada neste trabalho, através da inclusão de objetos de autenticação do tipo HASH_CHAINS na MAE que autentica as mensagens contendo esse tipo de campos mutáveis.

Proteção Anti-Repetição (SEQNUM): Um ataque típico em ambientes onde existe autenticação de mensagens consiste em repetir mensagens com autenticação válida em momentos posteriores à sua transmissão legítima (ataque de *reply*). A maioria dos protocolos que propagam suas mensagens para além da vizinhança de 1-*hop*, possui proteção contra o

processamento repetido de mensagens. Entretanto, em geral, essa proteção consiste em um número de seqüência que é incrementado monotonicamente, o que se mostra vulnerável à alguns ataques de *reply* [84]. MAE suporta uma proteção opcional contra esse tipo de ataque, baseada em números de seqüência aleatórios que são autenticados juntamente com os campos não mutáveis e incluídos na informação de autenticação como objetos do tipo SEQNUM.

4.3. AUTENTICAÇÃO DO PROTOCOLO DE ROTEAMENTO

4.3.1. Vulnerabilidades do Protocolo de Roteamento

Ataques contra os protocolos de roteamento estão usualmente relacionados com a injeção de informação de roteamento incorreta, na tentativa de perturbar a operação normal do protocolo. Este é o caso para ataques de modificação (alteração incorreta da mensagem do protocolo de roteamento durante o encaminhamento), fabricação (geração de mensagens de roteamento falsas) e personificação (usurpação da identidade de outra entidade da rede). A combinação dessas operações básicas é igualmente possível e provê um espectro mais largo de ataques. As vulnerabilidades descritas nesta seção são ataques contra os protocolos de roteamento para Manet que estão baseados em ações de fabricação, modificação e/ou personificação de mensagens do protocolo.

4.3.1.1. Vulnerabilidades do Protocolo OLSR

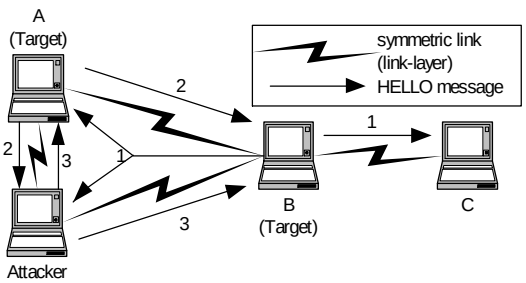
Como apresentado anteriormente, o protocolo OLSR possui dois tipos fundamentais de mensagens: HELLO e TC. Enquanto as mensagens HELLO estão restritas à vizinhança de 1 salto (i.e. não são encaminhadas pelos nodos), as mensagens TC são propagadas por toda a rede. Desse modo, os ataques envolvendo modificação de mensagens não se aplicam a mensagens HELLO, mas apenas a mensagens TC. Adicionalmente, as mensagens TC não possuem campos mutáveis, i.e. a mensagem sai pronta do nodo de origem e percorre toda a rede, sem modificações. Desse modo, todo ataque de modificação é também um ataque de personificação, já que as alterações realizadas são interpretadas como informação fornecida pelo nodo de origem. Ataques de fabricação são possíveis com ambas as mensagens HELLO e TC, podendo estes ataques estar combinados com ataques de personificação. As vulnerabilidades aqui apresentadas consideram a especificação original do protocolo OLSR [25], sem a proteção por uma MAE. Esses ataques podem ser facilmente mitigados com o uso de uma MAE apropriada. A Tabela 4-1 apresenta os ataques mostrados nesta seção.

Tabela 4-1 – Vulnerabilidades do Protocolo OLSR

Ataque	**Mensagem OLSR**	**Informação de Roteamento Falseada**	**Informação de Origem na Mensagem Corrompida**
Fabricação	HELLO	Neighbor List	Qualquer*
Fabricação + Personificação	HELLO	Link-status	Endereço IP do nodo personificado
Fabricação	TC	MS list	Qualquer*
Modificação + Personificação	TC	Sequence Number	Endereço IP do originador

*Em geral, o adversário vai tentar direcionar o tráfego da rede para si próprio, utilizando seu endereço IP na identificação do originador da mensagem. Alternativamente, o atacante pode direcionar o tráfego para um nodo inexistente, com objetivo de provocar negação de serviço. Neste caso, utiliza-se um endereço IP não alocado como originador da mensagem. Finalmente, o adversário pode personificar um outro nodo para forçar que todo o tráfego seja direcionado a ele.

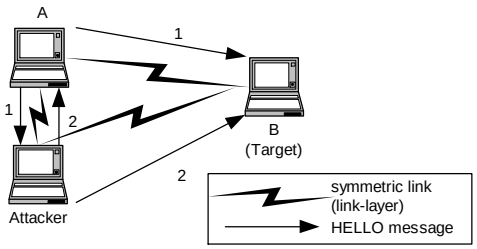
Fabricação de Mensagens HELLO: Neste ataque, mostrado na **Figura 4-2**, o adversário fabrica uma mensagem HELLO anunciando todos os nodos previamente divulgados em qualquer mensagem HELLO recebida por ele, juntamente com um endereço adicional não utilizado, com status de enlace simétrico. Ao receber essa mensagem, todos os vizinhos do adversário escolhem-no como único MPR. Assim, todo o tráfego em direção dos nodos que não sejam vizinhos diretos de um desses nodos passa a ser encaminhado para o adversário.



Antes do ataque: A (alvo) tem B (alvo) como MPR, o qual encaminha o tráfego de A para C. C está a uma distância de 02 saltos de A e do adversário.
1 – Mensagem HELLO de B: adversário identifica A e C como vizinhos de B.
2 – Mensagem HELLO de A: adversário identifica B como vizinho de A.
3 – Depois de receber uma nova mensagem HELLO de B, o adversário fabrica uma mensagem HELLO anunciando A, B, C e X (um endereço adicional não alocado) com status de enlace simétrico. Isto faz com que A e B selecionem o adversário como MPR, impedindo qualquer tráfego de A para C.

Figura 4-2 – Fabricação de Mensagens HELLO

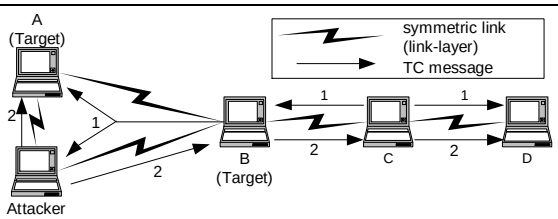
Fabricação + Personificação de Mensagens HELLO: A Figura 4-3 ilustra este ataque, onde um adversário fabrica uma mensagem HELLO personificada, depois de receber uma mensagem HELLO legítima, anunciando todos os nodos que estavam na mensagem correta com status de enlace perdido (“lost”). O nodo personificado é o mesmo nodo que originou a mensagem HELLO correta. Quando seus vizinhos recebem a mensagem HELLO fabricada, todos os nodos que tiveram seus enlaces anunciados erroneamente com status perdido (“lost”) mudam o status de seu enlace com o nodo personificado para assimétrico (“heard”). Assim, nenhum tráfego é encaminhado para o nodo personificado por esses enlaces.



Antes do ataque: A (alvo) e B (alvo) aprenderam um enlace com status simétrico entre si, através de mensagens HELLO prévias.
1 – Mensagem HELLO de A: Adversário identifica B como vizinho de A.
2 – Depois de receber a mensagem HELLO de A, o adversário fabrica uma mensagem HELLO personificando A, a qual anuncia B com status de enlace perdido. Esta mensagem faz B alterar o status de seu enlace com A para assimétrico, evitanto que qualquer tráfego seja encaminhado por esse enlace.

Figura 4-3 – Fabricação + Personificação de Mensagens HELLO

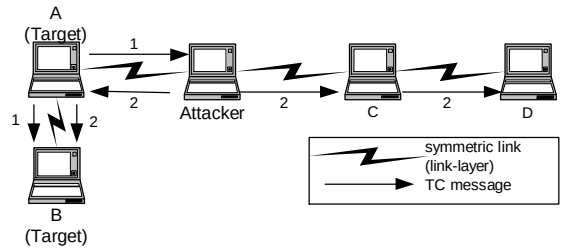
Fabricação de Mensagens TC: Neste ataque (Figura 4-4), o adversário fabrica uma mensagem TC anunciando nodos distantes (2 saltos ou mais) como parte do seu conjunto MS. Isso faz com que os seus vizinhos escolham rotear tráfego para esses nodos anunciados através do atacante.



Antes do ataque: A (alvo) tem uma rota para D, com custo de 03 saltos, através de B (alvo).
1 – Mensagem TC de C: adversário identifica D a uma distância de 03 saltos.
2 – Após receber a mensagem TC, o adversário fabrica outra mensagem TC, anunciando D como patê de seu MS (vizinhos diretos). Isso faz com que A pare de rotear o tráfego para D através de B e passe a fazê-lo para o adversário.

Figura 4-4 – Fabricação de Mensagens TC

Modificação + Personificação de Mensagens TC: Neste ataque (Figura 4-5), o adversário altera o campo "message sequence number" de uma mensagem TC, antes de encaminhá-la. O campo "message sequence number" é incrementado de um valor inteiro. Este ataque faz com que o processamento e encaminhamento de mensagens TC provenientes do nodo anunciado como originador da mensagem modificada seja interrompido em toda a rede[28].



1 – Mensagem TC de A (alvo): "message sequence number" = x
2 – Adversário encaminha a mensagem TC com o campo "message sequence number" modificado para x+i (um inteiro arbitrário). Isso provoca a paralização do processamento e encaminhamento das mensagens TC provenientes de A em C e D enquanto essas mensagens tiverem o campo "message sequence number" < x+i.

Figura 4-5 – Modificação + Personificação de Mensagens TC

4.3.1.2. Vulnerabilidades do Protocolo TBRPF

O protocolo TBRPF tem uma característica bastante diferente dos demais protocolos de roteamento para Manet, pois suas mensagens são enviadas apenas na vizinhança. Assim, não há qualquer tipo de encaminhamento de mensagens, o que torna ataques de modificação não aplicáveis.

No TBRPF pode-se fazer fabricação, com ou sem personificação, de mensagens HELLO ou de mensagens com anúncio da sub-árvore da topologia (*topology update*). No caso de fabricação de mensagens HELLO, o efeito é parecido com os ataques de fabricação e fabricação + personificação de mensagens OLSR: altera-se o conjunto de vizinhos a um e dois saltos aprendido pelos nodos participantes ou altera-se o status dos enlaces aprendidos entre dois vizinhos, respectivamente. Na fabricação de mensagens topology update (TU), um adversário pode alterar o cálculo das sub-árvores de topologia mantida pelos nodos da vinhança. Essas alterações serão repassadas em anúncios TU dos vizinhos, propagando as informações incorretas de roteamento por toda a rede.

[28] O revide (*fight back*) contra esse tipo de ataque não está definido para o OLSR, apesar de esta ser uma prática comum em outros protocolos de roteamento de estado de enlace (e.g. OSPF).

4.3.1.3. Vulnerabilidades dos Protocolos AODV e DSR

Os protocolos AODV e DSR são protocolos de roteamento reativos, isto é, as informações acerca das rotas são obtidas sob demanda, através de solicitações explícitas. Assim, as duas principais mensagens nesse tipo de protocolo de roteamento são: *route request* (RREQ) e *route reply* (RREP), indicando respectivamente a solicitação por uma rota e a resposta a essa solicitação. Esses protocolos usam ainda as mensagens *route error* (RERR) e *route reply acknowledgement* (RREP-ACK) como mensagens auxiliares para informar erros de roteamento (e.g. indisponibilidade de rotas) e para confirmar o recebimento de respostas RREP, respectivamente. A diferença básica entre o AODV e o DSR consiste no fato do primeiro utilizar um algoritmo de roteamento do tipo vetor de distância, enquanto o segundo utiliza um roteamento na origem. Assim, o conteúdo das mensagens RREP é diferente em cada caso. Para o AODV, tem-se apenas as os endereços de origem e destino, além de um contador de salto (*hop count*) que é alterado a cada encaminhamento da mensagem. Já no DSR, as informações dos endereços IP de todos os nodos intermediários são adicionados à mensagem à medida em que ela é encaminhada pelos nodos da rede, desde a origem até o destino. A seguir, apresenta-se um apanhando das principais vulnerabilidades dos protocolos AODV e DSR, no que diz respeito a ataques de fabricação, modificação e personificação de mensagens [28].

Ataques de Modificação

- Modificação de número de seqüência de destino em mensagens RREP (protocolo AODV): Os números de seqüência de rotas são incrementados monotonicamente, para evitar o processamento de mensagens (rotas) duplicadas/antigas. Um adversário pode modificar esse número, aumentando de um inteiro grande, fazendo com que as atualizações mais recentes sejam consideradas mais antigas. Assim, as mensagens verdadeiras não são mais processadas. Esse é um ataque semelhante ao ataque de modificação de números de seqüência em mensagens TC (OLSR).
- Modificação de número de saltos (protocolo AODV): ao modificar (para mais ou para menos) o *hop count* de mensagens RREQ ou RREP, altera-se o cálculo da tabela de roteamento de todos os nodos que utilizam esta mensagem.
- Modificação das rotas em mensagens RREQ e RREP (protocolo DSR): ao se excluir, incluir ou modificar endereços na lista de endereços que compõem as

rotas entre origem e destino, altera-se o caminho que será executado pela mensagem no roteamento pela origem.

Ataques de Personificação + Fabricação

- Fabricação de mensagens RREP, personificando nodos na rede: Um adversário pode escutar mensagens RREP trocadas entre os nodos e fabricar mensagens RREP alterando rotas entre dois nodos.
- Fabricação de mensagens RERR: mensagens RERR podem ser fabricadas, personificando nodos da rede que estão no caminho que se deseja quebrar.

Ataques de Fabricação

- Fabricação de mensagens RREP para "envenenamento" do *cache* de rotas (protocolo DSR): como as mensagens RREQ e RREP contêm a lista de endereços IP em uma rota entre dois nodos, uma otimização para evitar requisições por rotas consiste em se escutar promiscuamente as mensagens trocadas na rede e colocar as rotas divulgadas em uma *cache* de rotas que é consultada antes de se enviar requisições de rotas. Assim, ao se fabricar mensagens RREP e RREQ anunciando rotas falsas, essas rotas podem ser aprendidas por nodos que as escutam promiscuamente, passando a executar o roteamento contido nas mensagens falsas.

4.3.2. MAE para os Protocolos de Roteamento

4.3.2.1. MAE para o Protocolo OLSR

O protocolo OLSR permite que múltiplas mensagens sejam agrupadas em um único pacote OLSR para transmissão em cada salto. Normalmente, algumas das mensagens OLSR dentro de um pacote são processadas apenas localmente, enquanto outras são também encaminhadas para outros nodos como parte do mecanismo de *flooding*. Além disso, nem todas as mensagens em um pacote são provenientes de um mesmo originador. Assim, a proteção pela MAE é definida para ser usada em nível de mensagem e não em nível de pacote. Isto é, cada mensagem OLSR deverá ser autenticada por uma MAE. Como os campos do pacote não são efetivamente usados pelo algoritmo de roteamento e servem apenas para encapsulamento de múltiplas mensagens em transmissões de 1-hop, esses campos não precisam ser protegidos por uma MAE.

Nenhuma das mensagens OLSR (e.g. HELLO, TC, MID, HNA and FRR) tem qualquer campo mutável. Entretanto, o cabeçalho genérico de mensagens possui campos “hop count” e “time to live” que são mutáveis. Mensagens HELLO e FRR são difundidas (*broadcast*) apenas para a vizinhança de 1-hop, enquanto mensagens TC, MID e HNA são inundadas (*flooding*) em toda a rede. Dado que esses campos mutáveis não são usados no cálculo da tabela de roteamento, mas apenas pelo algoritmo de *flooding* (o qual já é robusto por si mesmo, devido às redundâncias naturais na topologia da rede), nenhuma proteção adicional é requerida para autenticação desses campos. Desse modo, a MAE para autenticação do OLSR consiste simplesmente de um único objeto de autenticação contendo uma assinatura digital ou um código de autenticação de mensagem. Todos os campos das mensagens são autenticados, exceto os campos “hop count” e “time to live”, que devem ser zerados para o cálculo de DS ou MAC. Um objeto de certificado pode ser acrescentado em cada mensagem, caso o sistema de autenticação seja DS. Neste caso, uma implementação básica de um mecanismo de distribuição de certificados pró-ativo pode ser definida, requerendo-se que o certificado do signatário seja incluído em mensagens HELLO e TC. Uma otimização ainda pode ser realizada, incluindo-se o certificado em mensagens HELLO apenas quando um novo vizinho for detectado.

4.3.2.2. MAE para o Protocolo TBRPF

O TBRPF é um protocolo de roteamento pró-ativo que usa um algoritmo do tipo estado de enlace. Desse modo, este protocolo não possui campos mutáveis em suas mensagens que sejam efetivamente usadas pelo algoritmo de roteamento. Em especial, as mensagens do protocolo são encaminhadas apenas entre os vizinhos. A MAE para proteger o protocolo TBRPF é simplesmente construída usando-se um único objeto de autenticação contendo uma assinatura digital ou um código de autenticação de mensagem. Um mecanismo de distribuição pró-ativa de certificados eficiente consiste em exigir que os certificados dos signatários sejam incluídos apenas em mensagens HELLO.

4.3.2.3. MAE para o Protocolo AODV

O AODV tem campos mutáveis nas mensagens *route request* (RREQ) e *route reply* (RREP). Esses campos contêm métricas de roteamento do tipo “hop count”, alteradas a cada vez que a mensagem é processada e encaminhada pelos nodos entre a origem e o destino da mensagem. Um objeto do tipo cadeia de *hash* (HC) [41] é incluído e atualizado cada vez que

esses campos são alterados (i.e. a cada novo encaminhamento que a mensagem sofre). Esta proteção evita que um adversário possa decrementar o *hop count*. Mensagens *route error* (RERR) são autenticadas apenas pelo nodo que as encaminha. Mensagens *route reply acknowledgments* (RREP-ACK) não possuem campos mutáveis e são autenticadas apenas pelo originador. Os certificados podem ser eficientemente distribuídos, sob demanda, em mensagens RREQ e RREP.

4.3.2.4. MAE para o Protocolo DSR

A segurança do DSR é consideravelmente mais complexa, e uma solução completa deve requerer que todos os nodos intermediários que encaminham as mensagens autentiquem-nas. Uma segurança limitada pode ser obtida combinando-se o objeto de autenticação obrigatório, com um objeto HC, implementando um resumo salto-a-salto do traçado de endereços IP em mensagens RREQ. Isto evita que um adversário falsifique o nodo iniciador ou remova endereços IP corretos da lista de roteamento [49]. Mensagens RREP poderiam ser simplesmente autenticadas pelo destino da descoberta de rotas (i.e. o nodo gerando a mensagem RREP). De maneira semelhante ao AODV, os certificados podem ser eficientemente distribuídos, sob demanda, em mensagens RREQ e RREP.

4.3.2.5. Avaliação da Proteção da MAE

A Tabela 4-2 ilustra as principais características dos protocolo de roteamento para Manet e os requisitos da MAE para cada um deles.

Tabela 4-2 – MAE dos Protocolos de Roteamento para Manet

Protocolo de Roteamento	Descoberta de Rotas	Algoritmo de Roteamento	Mensagens Relevantes	Objetos de Autenticação na MAE
DSR	sob-demanda	roteamento na origem	RREQ	DS+HC
			RREP	DS
AODV	sob-demanda	vetor de distância	RREQ	DS+HC
			RREP	DS+HC
			RERR	DS
			RREP-ACK	DS
OLSR	pró-ativa	estado de enlace	Hello, TC, MID, HNA	DS
TBRPF	pró-ativa	estado de enlace	Hello, Topology Update	DS

4.4. AUTENTICAÇÃO DO PROTOCOLO DE AUTOCONFIGURAÇÃO

Ao contrário dos protocolos de roteamento, o processo de padronização de soluções para autoconfiguração em redes *ad hoc* está em seus estágios iniciais atualmente. Isso dificulta a análise de vulnerabilidades das alternativas correntes, pois trata-se apenas de propostas preliminares que têm por objetivo ilustrar conceitos chaves e a aplicabilidade de métodos de autoconfiguração. Assim, essas propostas não apresentam um nível detalhado de especificação e amadurecimento suficientes que permitam tratá-las como um candidato sério a um protocolo padrão em seus estágios atuais. Não obstante, F. Buiati [14] fez um trabalho relevante de especificação e implementação do protocolo DCDP. Em trabalho recente [13], propusemos mecanismos de segurança aplicáveis a este protocolo que utilizam os mesmos conceitos do modelo de confiança via L-Cert e autenticação com MAE apresentados neste trabalho. Portanto, restringimos a discussão acerca de vulnerabilidades e proteção do protocolo de autoconfiguração ao escopo do protocolo DCDP. Entretanto, como os diversos protocolos de autoconfiguração têm escalonamento semelhante ao protocolo DCDP (i.e. basicamente os mesmos tipos de mensagem são usadas – e.g. requisição de serviço e resposta de serviço), acreditamos que esta análise poderá ser facilmente estendida a outros protocolos.

4.4.1. Vulnerabilidades do Protocolo DCDP

Os protocolos de autoconfiguração têm dois tipos básicos de mecanismos: o processo de autoconfiguração propriamente dito e os mecanismos de manutenção e atualização das bases de dados de autoconfiguração (sincronização). Focalizamos nossa análise de vulnerabilidades no processo de autoconfiguração, uma vez que os mecanismos de sincronização ainda estão muito prematuramente especificados.

O processo de autoconfiguração envolve comunicações entre um solicitante (cliente) e um servidor, que atende a essas solicitações. Isso posto, classificamos os ataques de fabricação contra o protocolo de autoconfiguração em ataques de solicitante, onde o adversário exerce o papel de cliente, e de servidor, onde o adversário responde maliciosamente às solicitações de clientes da rede.

As comunicações do processo de autoconfiguração do DCDP evoluem essencialmente na vizinhança de 1-hop. Assim, não se aplicam ataques envolvendo modificação de mensagens do protocolo, nesta etapa (ainda que na etapa de sincronização esses ataques possam fazer sentido). No que diz respeito aos ataques de cliente, são possíveis igualmente ataques de fabricação simples (e.g. um nodo solicita um bloco de endereços IP e torna o bloco

alocado indisponível para alocação), assim como ataques combinando fabricação e personificação (e.g. um nodo solicita a liberação de um bloco de endereços que ainda está em uso, provocando possíveis alocações duplicadas). Por outro lado, no caso de ataques de servidor, ataques de personificação devem vir acompanhados de um ataque de negação de serviço sobre o nodo personificado. Caso contrário, mesmo que as mensagens personificadas sejam entregues, as mensagens corretas também o serão, o que limita a eficácia desse tipo de ataque. Assim, sem perda de generalidade, consideramos essencialmente ataques de fabricação de mensagens para este caso. Essas mensagens podem ser eventualmente resultantes de um processo combinado de fabricação e personificação, sem, contudo, que estes apresentem efeitos mais relevantes que aqueles resultantes de ataques de fabricação tão somente.

Ataques de solicitação: Neste tipo de ataque, os adversários atuam como clientes e fabricam mensagens requerendo o serviço de autoconfiguração. Como exemplo, um adversário pode requerer a alocação de um endereço IP, tornando indisponível para outros nodos, o bloco de endereços IP que foi associado a ele (ataque de fabricação). Outro exemplo é a possibilidade de um adversário solicitar a liberação de algum bloco de endereços IP, permitindo que esses endereços IP sejam usados para atender a novas requisições, mesmo que os endereços IP ainda estejam sendo usados por alguns nodos na rede (ataque de fabricação + personificação).

Ataques de servidor: Neste tipo de ataque, os adversários atuam como servidores na rede e atendem a requisições solicitadas respondendo com falsas mensagens. Como exemplo, um adversário pode responder a uma mensagem de requisição de endereço e fornecer endereços IP que já estejam sendo usados por outros nodos participantes da rede *ad hoc*, resultando em conflito de endereços. Alternativamente, o adversário pode responder a uma requisição de liberação de endereço IP para um nodo que deseja sair da rede, confirmando a saída do nodo, entretanto não liberando o endereço IP para atender a novas requisições.

4.4.2. MAE para o Protocolo DCDP

A MAE do protocolo DCDP é essencialmente constituída de um objeto de autenticação obrigatório e de um objeto CERT, caso o sistema de autenticação seja DS. A distribuição de certificados pode ficar a cargo do protocolo de roteamento, caso se esteja utilizando um protocolo pró-ativo. Caso contrário, objetos CERT devem ser igualmente incluídos nas mensagens de sincronização das bases de dados de autoconfiguração, pois estes operam de modo pró-ativo. No caso de junção de duas redes iniciadas com ACDs diferentes,

o estabelecimento da relação de confiança cruzada deve anteceder a detecção e resolução de conflitos de endereços. Nestes casos, recomenda-se que os certificados sejam incluídos nas mensagens dos protocolos de autoconfiguração, pois o roteamento entre nodos provenientes de redes diferentes não será efetivo até que o processo de resolução de conflitos de endereços esteja terminado.

4.5. POLÍTICA DE SEGURANÇA E CONFIGURAÇÕES PARA OS SERVIÇOS DE CERTIFICAÇÃO E AUTENTICAÇÃO

Manets podem ser usadas em diferentes contextos, os quais possuem requisitos de segurança distintos. Por exemplo, uma rede aberta para compartilhamento de dados em uma sala de estudos, na qual os usuários podem livremente entrar e sair, tem requisitos de segurança diferentes de uma Manet formada para uma missão de resgate em uma área atingida por um desastre natural ou mesmo de uma Manet militar no campo de batalha. Enquanto no primeiro caso, a filiação à rede não é pré-determinada, nos outros dois casos os nodos podem ter sido previamente iniciados (com certificados e partes da chave privada de certificação) de maneira a prover tanto identificação quanto controle de acesso de maneira adequada nestes cenários.

Um conjunto de opções de configurações permite que o serviço de certificação e de autenticação providos em nossa solução de segurança seja prontamente adaptável para cenários tão diferentes quanto à política de segurança. O objetivo é permitir que os requisitos de segurança possam ser mapeados em configurações específicas determinando o comportamento dos diferentes mecanismos de seguranças implantados. Assim, no cenário de "rede aberta", exemplificado pela rede da sala de estudos, o sistema de autenticação pode ser configurado simplesmente para DS, e a política de certificação de novos nodos pode ser "servir desde que o requerente se identifique", por exemplo, mostrando sua carteira de estudante a usuários próximos que são seus vizinhos na rede. Restrições à certificação podem estar relacionadas com comportamento inadequado anterior. Em cenários de resgate, um grupo bem definido de nodos está cooperando na Manet e a probabilidade de um adversário interferir intencionalmente nesta rede é baixa. Igualmente, os usuários que cooperam na rede se conhecem e confiam mutuamente em si, não necessitando de um mecanismo para identificar nodos comprometidos. Assim, todos os nodos podem ser iniciados com uma chave secreta compartilhada e o sistema de autenticação configurado para MAC. Nodos novos não são admitidos na rede, exceto aqueles previamente configurados com a "chave de grupo".

Finalmente, no caso do cenário da Manet militar, tem-se uma rede com alta probabilidade de comprometimento de nodos, inclusive por captura física. O acesso à rede deve ser rigorosamente controlado, de modo que a política de certificação seja "rejeitar silenciosamente sempre". Entretanto, renovações de certificados são possíveis para nodos que não apresentaram histórico de mau comportamento (possível comprometimento) e possam comprovar sua identidade. A Tabela 4-3 ilustras esses conceitos de política de segurança aplicados aos parâmetros de configuração da solução, para os cenários mencionados.

Tabela 4-3 – Tipos de Políticas de Autenticação

Tipo de Política	Exemplo de Cenário	Sistema de Autenticação	Política de Certificação	Política de Renovação de Certificados
rede aberta	sala de estudos	DS	de acordo com verificação de identidade	servir sempre
rede gerenciada	resgate	MAC c/ chave de grupo	-	-
rede gerenciada em ambiente hostil	campo de batalha	DS	rejeitar silenciosamente	de acordo com verificação de identidade

A Figura 4-6 resume os parâmetros de configuração para a política de autenticação e certificação considerados em nosso trabalho [93].

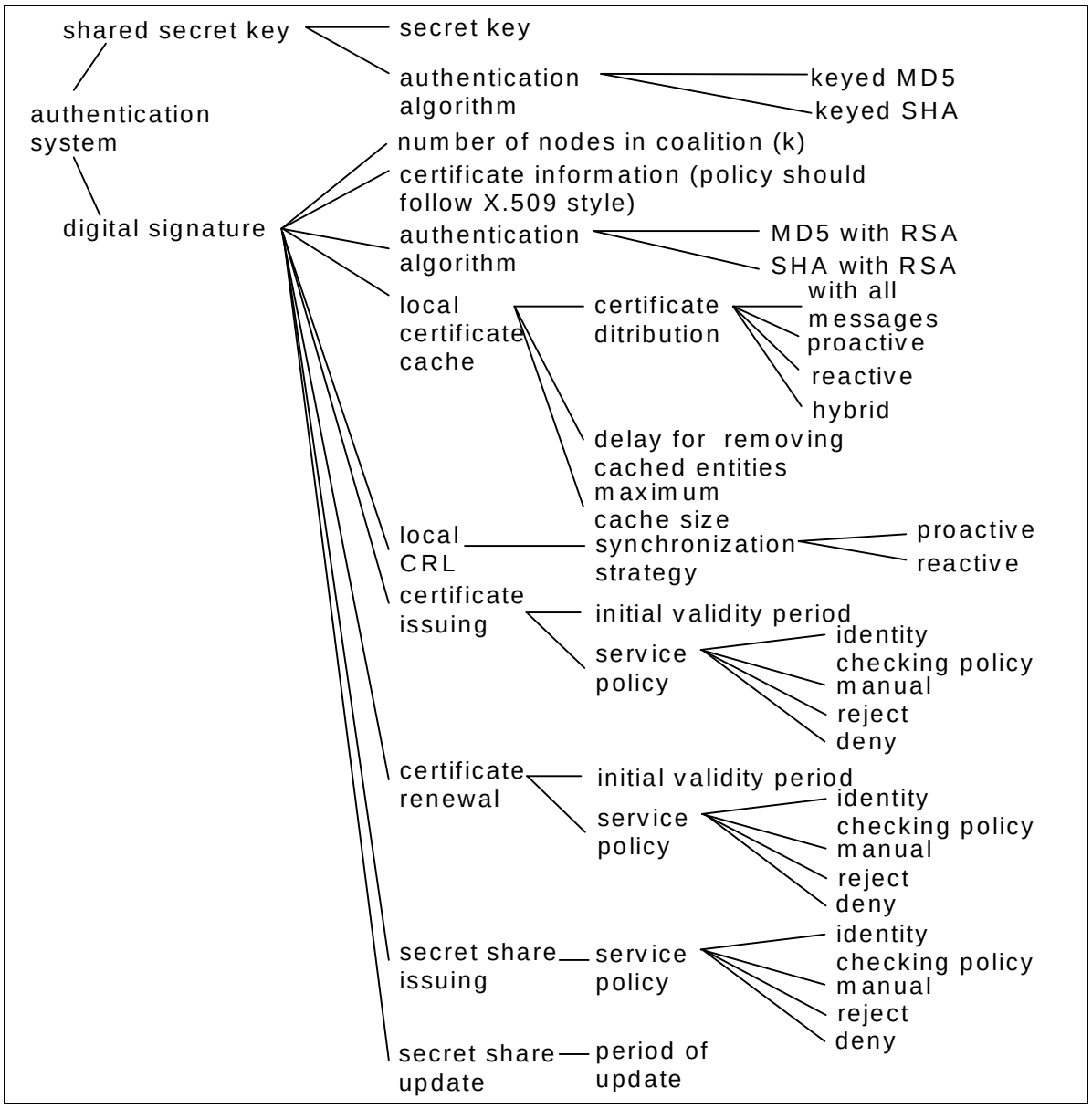


Figura 4-6 – Parâmetros para Política de Autenticação e Certificação

5. DETECÇÃO E RESPOSTA ÀS INTRUSÕES EM MANET

O sistema de detecção de intrusão tem uma função fundamental no modelo de segurança proposto neste trabalho, fornecendo uma monitoração pró-ativa do estado da segurança da Manet. Esta monitoração tem por objetivo identificar violações da política de segurança (ataques). Além disso, através de uma resposta também pró-ativa à detecção de ataques, o IDS interage com outros serviços de segurança (e.g. autenticação, certificação, controle de acesso) para eliminar as causas de um ataque (e.g. revogando o certificado de nodos que violam a política de segurança) ou mitigar os seus efeitos (e.g. reconfigurando a filtragem de pacotes para evitar o encaminhamento de pacotes que violam a política de segurança).

Este capítulo apresenta a especificação de um sistema de detecção de intrusão projetado para Manet. Define-se uma nova arquitetura para IDS, derivada dos requisitos específicos do ambiente Manet. Isto é, o IDS projetado tem características de distribuição, auto-organização e localização. São usadas as características de mobilidade e autonomia associadas à tecnologia de agentes móveis [3,85], que são incorporados ao IDS para prover uma solução eficiente e flexível às limitações de largura de banda e conectividade das Manets. Além disso, a concepção do IDS considera uma completa distribuição do processo de detecção de intrusão, ao invés de limitar a distribuição às tarefas de coleta de dados, que é a abordagem mais comum na maioria dos projetos de IDS baseados em agentes móveis. Finalmente, ressalta-se a modularização da arquitetura do IDS distribuído, permitindo a imediata extensão das capacidades do sistema em termos da derivação e incorporação de novos módulos ao IDS.

5.1. IDS COMPLETAMENTE DISTRIBUÍDO

O processo de detecção de intrusão consiste na coleção e análise de dados de auditoria provenientes da rede, do sistema ou das aplicações. Quando detectas, as intrusões devem ser reportadas à gerência de segurança. Do mesmo modo, a detecção pode disparar uma resposta automática, com objetivo de eliminar as causas ou efeitos da intrusão. Dada a ausência de centralização, a mobilidade dos nodos e os enlaces sem fio das Manets, algumas (senão todas) as tarefas requeridas no processo de detecção de intrusão descritas acima devem ser executadas de forma distribuída e cooperativa [117]. No projeto deste trabalho, um IDS local

(L-IDS) é colocado em cada nodo da Manet. Os L-IDS comunicam-se entre si através de um mecanismo que leva em conta as restrições de contexto Manet, i.e. banda passante limitada ou conectividade pobre. Esta arquitetura foi inicialmente definida em [117] como um requisito básico para IDS em ambientes Manet. O uso de agentes móveis é proposto como uma alternativa flexível e autônoma de interação entre os L-IDS [3].

Se um L-IDS falha em cooperar durante algum período de tempo (e.g. o nodo moveu-se, falhou ou está comprometido), o serviço de detecção de intrusão não pode ser degradado. As redundâncias inerentes às Manets servem de compensação para os nodos que não estão cooperando no processo de detecção, sendo possível que mais de um nodo possa acompanhar e detectar a evolução de um mesmo ataque.

A tecnologia de agentes é uma alternativa ao modelo de distribuição cliente-servidor [55]. O uso de agentes móveis, em oposição a abordagens tradicionais onde os dados são transportados em direção aos pontos de computação, permite que o código seja transportado em direção aos dados. Desse modo, agentes cuidadosamente projetados podem reduzir a quantidade de dados trocados através da rede, enquanto provêm uma maneira flexível de distribuição. Adicionalmente, um nodo despachando um agente não precisa esperar que este retorne para prosseguir seu processamento normal, uma vez que os agentes podem ser despachados ou mesmo destruídos em outros nodos sem ter que voltar ao nodo originário.

Um L-IDS coopera com outros despachando agentes móveis para outros nodos e processando as tarefas embutidas em agentes recebidos. No projeto deste trabalho, a cooperação pode ser feita nas diferentes etapas da detecção de intrusão. Durante a coleta de dados, os nodos trocam informações sobre eventos para construir evidências de intrusões. Na execução do algoritmo de detecção, os L-IDS trocam informações sobre o estado atual das estruturas que implementam o algoritmo de detecção. Finalmente, a correlação de alertas é igualmente possível, quando um nodo usa alertas provenientes de outros nodos para reforçar as evidências de atividades suspeitas detectadas localmente. Em qualquer dos casos, a cooperação é executada por meio de agentes móveis que movem-se de um sistema para outro. Agentes móveis também são um primeiro elemento de resposta para a natureza dinâmica da topologia e das adesões em Manet. De fato, quando um nodo ingressa na rede, ele o faz com um L-IDS em execução que contém uma plataforma de agentes móveis. Ele pode, portanto, tomar parte no processo colaborativo de detecção de intrusão imediatamente. Do mesmo modo, quando um nodo deixa a rede, outros nodos da vizinhança podem prover colaborativamente as informações necessárias para a execução do processo de execução.

Manter a colaboração entre um número restrito de nodos está relacionado com os requisitos de uso da banda passante e de escalabilidade. A razão de se executar algumas das tarefas de detecção apenas na vizinhança local é duplamente justificada pela natureza das Manets. Um nodo destas redes deve sempre coletar e manter informações sobre seus vizinhos, qualquer que seja o protocolo de roteamento que esteja em uso. Além disso, qualquer informação indo para ou vindo de um determinado nodo deve ser roteada através de um de seus vizinhos, quando o seu encaminhamento pode ser promiscuamente monitorado devido à natureza difusora dos enlaces sem fio [119]. Assim, vizinhos de nodo que está sendo atacando são as fontes primárias de informação sobre o estado do nodo que sofre os efeitos de um ataque, naturalmente elegíveis. Os vizinhos são também elegíveis como pares para a colaboração com objetivo de descobrir novas informações que não estão disponíveis localmente, acerca de uma intrusão em curso.

5.2. ARQUITETURA MODULAR DO L-IDS

Nesta seção é proposta a arquitetura modular do L-IDS, conforme ilustrado na Figura 5-1. Esta arquitetura consiste em uma adaptação do *framework* de detecção de intrusão proposto pelo grupo de trabalho sobre detecção de intrusão do IETF [115], formado por sensores, analisadores e gerentes. As adaptações visam estender este *framework* para atender os requisitos do contexto Manet.

Em cada L-IDS, o módulo **sensor** coleta dados a partir das fontes de dados de auditoria e os sintetiza, o módulo **analisador** processa os dados sintetizados para detectar situações que podem constituir-se violações da política de segurança, enquanto o módulo **gerente** realiza a interface de gerenciamento de todo processo, além de executar as tarefas de correlação de alertas e ativação de resposta automática às intrusões. As restrições do ambiente Manet são consideradas através dos módulos **gerente de distribuição** e **plataforma de agentes**.

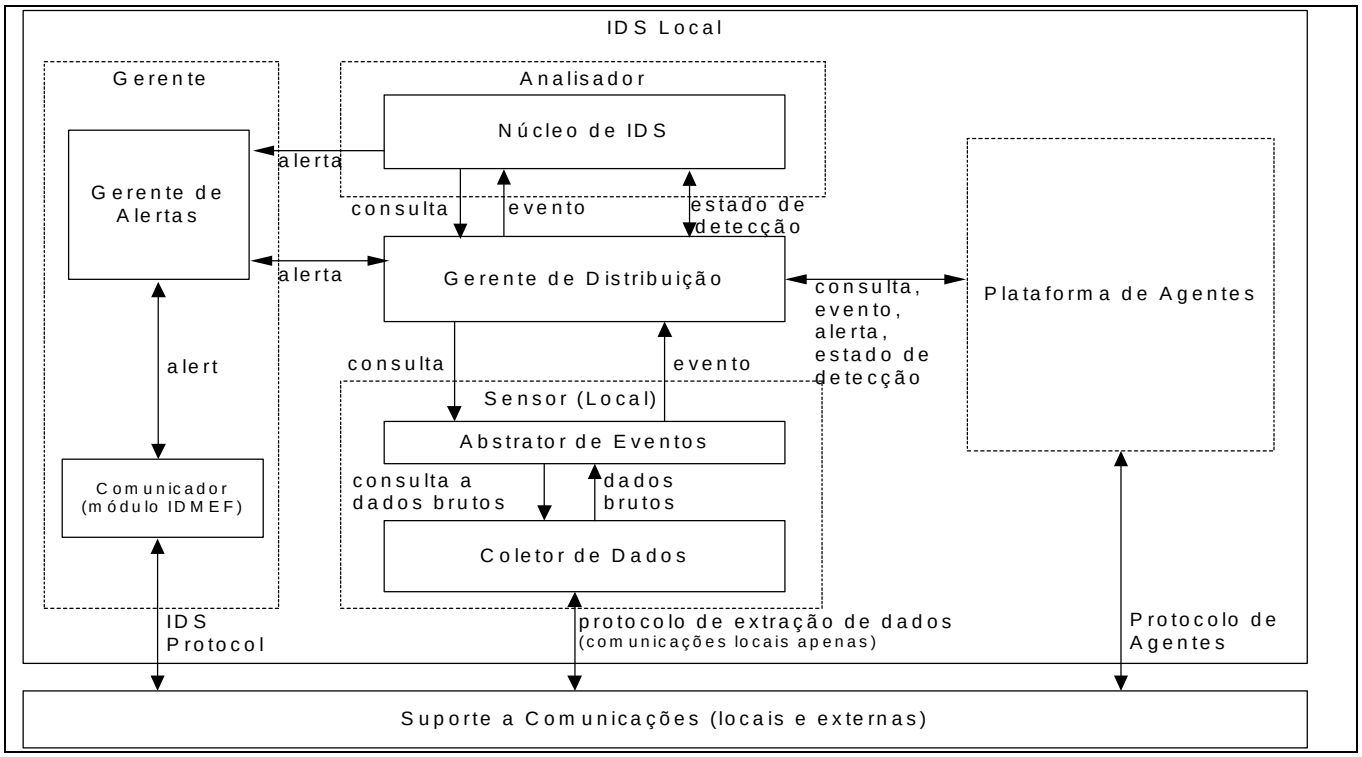


Figura 5-1 – Arquitetura Modular do L-IDS

5.2.1. *Framework* de Detecção de Intrusão

Os **dados** de auditoria utilizados pelo L-IDS podem ser coletados a partir de diferentes fontes, tais como uma interface de captura de pacotes de rede (nível de rede), um sistema de *log* (nível de sistema) ou mesmo uma MIB (nível de rede, sistema ou aplicativo). Estes dados são usualmente brutos e têm pouca semântica. Além disso, eles estão disponíveis em um formato que depende da fonte de dados. Assim, um pré-processamento é aplicado esses dados, transformando-os em mensagens semanticamente mais ricas para uso pelo algoritmo de detecção, denominadas **eventos**. Esta transformação realizada nos dados brutos é denominada abstração de eventos. A abstração de eventos pode ser feita através da aplicação de diversas técnicas, tais como casamento de padrões [53], mineração de dados [70] ou correlação estatística [117].

Neste projeto, o módulo **sensor** é decomposto em dois: **Extrator de Eventos** e **Coletor de Dados**. Esses módulos separam as tarefas de recuperação de dados e de abstração de eventos em duas entidades diferentes. A idéia é permitir múltiplas implementações para o coletor de dados, que podem operar simultaneamente coletando dados de fontes de auditoria diferentes. Assim, permite-se que o processo de abstração de eventos tenha regras de abstração que consumam dados brutos originados em múltiplas fontes. Do mesmo modo,

múltiplas implementações para o extrator de eventos são possíveis, permitindo o uso simultâneo de diferentes técnicas de abstração.

O **analisador** processa os eventos de acordo com alguma estratégia de detecção definida. Pelo menos duas metodologias de detecção de intrusão estão correntemente em discussão: detecção por uso incorreto e por desvio de comportamento (anomalia) [29]. De uma maneira geral, é correto afirmar que essas metodologias são complementares entre si. É objetivo deste projeto manter uma estratégia de detecção de intrusão híbrida, combinando técnicas de detecção por uso incorreto e por anomalia. Assim, na arquitetura do L-IDS, cada implementação de um algoritmo de detecção específico é encapsulada em um módulo **núcleo de IDS**. É igualmente possível que se tenha múltiplas instâncias deste módulo, cada qual implementando um algoritmo de detecção específico.

Uma representação concisa do estado corrente de execução do algoritmo de detecção consiste na mensagem **estado de detecção**. Esta informação pode relacionar-se com o estado de detecção de um determinado ataque, em detecção por uso incorreto, ou com o estado de estimulação do modelo de comportamento, em detecção por desvio de comportamento. O analisador pode ainda realizar uma **consulta** a um sensor local ou remoto, para aferir a ocorrência de eventos específicos em uma trilha de detecção que esteja sendo percorrida, sob demanda. Quando detecta algum tipo de atividade que é considerada intrusiva pelo algoritmo de detecção, é gerado um **alerta** ao módulo gerente.

O módulo **gerente** está dividido em dois módulos: **gerente de alertas** e ***framework* de comunicação**. Enquanto o gerente de alertas realiza as tarefas de interpretação de alertas, eliminação de falsos positivos e falsos negativos, correlação de alertas e resposta à intrusão, o ***framework* comunicação** provê uma interface de comunicação entre o IDS e outros serviços de segurança, inclusive para interação com outros sistemas de detecção de intrusão. Essa comunicação pode se dar usando formatos padronizados de mensagens de IDS, tais como o IDMEF [69].

5.2.2. Restrições do Ambiente Manet

Na maioria das arquiteturas de IDS distribuídos, a distribuição está restrita à coleta de dados. Em IDS para redes *ad hoc*, isto é obrigatório, dado que a coleta de volumes importantes de dados brutos a partir de entidades remotas é proibitiva devido à banda passante limitada dos enlaces sem fio. Além da coleta local de dados, o IDS projetado neste trabalho permite uma completa distribuição das tarefas do IDS, possibilitando que tanto a execução do algoritmo de detecção quanto o gerenciamento de alertas sejam igualmente distribuídos.

Assim, neste IDS, a coleta de dados é sempre localizada. Os dados trocados entre os L-IDSes consistem exclusivamente de informações concisas (e.g. eventos) resultantes do pré-processamento local de dados brutos. A cooperação na execução do algoritmo de execução, por sua vez, é realizada através da troca de mensagens de estado de execução, enquanto a colaboração no gerenciamento de alertas é feito através da troca da distribuição de alertas.

Em todos os casos, a distribuição e cooperação são realizadas através de agentes que geram mensagens de alto nível do IDS (e.g. eventos, consultas, estado de detecção e alertas). A distribuição dessas mensagens aos módulos locais e remotos é gerenciada pelo módulo **gerente de distribuição**. Os agentes, por sua vez, são despachados e gerenciados no módulo **plataforma de agentes**.

5.2.3. Mensagens geradas pelos L-IDS

As mensagens geradas pelo L-IDS podem ser especificadas em termos das seguintes cláusulas:

- As mensagens referem-se a um conjunto de entidades do sistema, possivelmente vazio, que é formado por pares atacante↔alvo. Um **atacante** consiste de um conjunto, possivelmente vazio, de identificadores de entidades da rede que são consideradas como possíveis causadoras do ataque. Do mesmo modo, um **alvo** consiste de um conjunto, possivelmente vazio, de identificadores de entidades da rede que são consideradas como possíveis afetados pelo ataque.
- Mensagens **evento**, **consulta**, **alerta** e **estado de detecção** possuem os seguintes atributos: identificador, identificador da entidade de origem e um conjunto, possivelmente vazio, de pares atacante↔alvo.
- Uma mensagem **consulta periódica** é um tipo especial da mensagem consulta, que é gerada automaticamente de maneira periódica. Esta mensagem possui um atributo adicional: período, indicando de quanto em quanto tempo tal mensagem será gerada.
- Mensagens **alerta** e **estado de detecção** possuem um atributo adicional: identificador do ataque, identificando qual o ataque está sendo monitorado quando da geração destas mensagens.
- Uma mensagem **estado de detecção** possui um atributo *booleano* adicional: isClone, indicando se os pares atacante↔alvo da mensagem devem ser eliminados

no núcleo do IDS de origem (isClone = false) ou se estes devem ser clonados na mensagem (isClone = true) e mantidos ativos no L-IDS de origem.

- As mensagens **evento**, **consulta**, **consulta periódica**, **alerta** e **estado de detecção** possuem ainda um atributo *booleano* isLocal, indicando se a mensagem deve ser consumida localmente (isLocal = true) ou deve ser despachada para L-IDS remotos (isLocal = false). No primeiro caso, mensagens **evento** e **estado de detecção** são consumidas pelo módulo analizador; mensagens **consulta** e **consulta periódica** são consumidas pelo módulo sensor; e mensagens **alerta** são consumidas pelo módulo gerente. Alternativamente (isLocal = false), as mensagens devem ser despachadas para L-IDS remotos, através do módulo gerente de distribuição.
- Para que as mensagens possam ser despachadas pelo gerente de distribuição (isLocal = false), estas devem possuir ainda: um conjunto, possivelmente vazio, de identificadores das entidades de destinos, um atributo *booleano* isFlooded e um atributo inteiro positivo TTL (*time to live*). A mensagem sempre deve ser despachada para os L-IDS cujos identificadores estejam no conjunto de identificadores das entidades de destino. Caso isFlooded = false, o conjunto de identificadores das entidades de destino não pode ser nulo. Caso contrário, a mensagem será difundida na rede em um raio medido em termos do TTL. Isto é, se TTL = 1, tem-se uma difusão na vizinhança a um salto de distância da origem da mensagem. Do mesmo modo, se TTL = 2, tanto os vizinhos a um salto quanto os vizinhos a dois saltos recebem a mensagem, e assim por diante.

5.3. DETECÇÃO DE INTRUSÃO POR USO INCORRETO

Para a detecção de ataques pelo método do uso incorreto adota-se um mecanismo de casamento de padrões onde as assinaturas dos ataques são modeladas em termos de um diagrama de estados finitos (DEF) com temporização, semelhante ao esquema proposto em [53]. Neste esquema, o módulo núcleo do IDS é composto por uma máquina de estados finitos, que processa os eventos recebidos para identificar transições entre os estados dos DEF que definem as assinaturas dos ataques monitorados.

Um DEF possui um estado inicial, um estado final e estados intermediários. Um conjunto de pares atacante↔alvo é mantido em cada estado do DEF, exceto para os estados inicial e final. Assim, toda vez que um evento dispara uma transição do estado inicial para

outro estado do DEF, este recebe o(s) par(es) atacante↔alvo contido(s) no evento. Isso permite que um mesmo ataque possa ser rastreado, ao mesmo tempo, em proveniência de origens diferentes. Do mesmo modo, um par atacante↔alvo é removido do DEF sempre que uma transição for disparada para o estado final, contendo este par atacante↔alvo.

A permanência de um par atacante↔alvo em um estado não inicial pode ser limitada em termo de uma temporização, disparando uma transição para outro estado ou auto-transições, caso não sejam detectadas transições disparadas por eventos para este par atacante↔alvo em um período de tempo pré-determinado. Assim, uma transição é sempre disparada para um par atacante↔alvo, seja esta devido à ocorrência de um evento ou a um estouro do temporizador.

Os eventos e as consultas podem ser originados localmente ou remotamente, permitindo a colaboração na coleta de dados. No que diz respeito à colaboração na execução do algoritmo de detecção, as mensagens de estado de detecção correspondem às informações sobre o estado corrente de um DEF, que representa uma assinatura de ataque. Assim, ao distribuir essas informações, um nodo pode fazer com que o processo de detecção continue em outro nodo, a partir da última transição detectada localmente. Em cada transição em um dos DEF, pode-se executar uma ação que pode ser: a realização de uma consulta, a distribuição de um estado de execução ou a geração de um alerta, quando as transições indicam um reconhecimento positivo de um padrão de ataque. Essas mensagens podem ser consumidas localmente ou despachadas em agentes para outros nodos.

O Kernel de IDS para esse tipo de detecção pode ser especificado em termos das seguintes cláusulas:

- Uma **assinatura de ataque** deve ser especificada em termos de um DEF com temporização. O módulo núcleo do IDS deve manter um conjunto de DEF.
- Um **DEF** é definido como um autômato formado por um estado inicial, um estado final, conjunto finito de estados intermediários e um conjunto finito de transições, constituindo um grafo orientado onde os estados são os nodos e as transições são as arestas. Um DEF contém: identificador de ataque, um conjunto de estados e um conjunto de transições.
- Um **estado** está associado a uma determinada situação ou status (possivelmente abstrato) assumido pelo sistema monitorado em um determinado instante de tempo, referindo-se a um determinado conjunto, possivelmente vazio, de entidades

do sistema formado por pares atacante↔alvo. Estados iniciais e finais têm sempre o conjunto de pares atacante↔alvo vazio.

- Uma **transição** consiste em uma alteração de um estado para outro, sendo definida em termos do estado de origem (atual), do estado de destino (novo estado), de um (único) fator que dispara a transição e de um conjunto, possivelmente vazio, de ações que devam ser executadas. Uma transição que tenha um mesmo estado como origem e destino é denominada uma alto-transição. O fator que dispara uma transição pode ser um evento ou um temporizador. O fator que dispara a transição e a própria transição são especificados para um mesmo conjunto, possivelmente vazio, de pares atacante↔alvo.
- Cada estado possui um **temporizador** que monitora o tempo máximo ($Time_{max}$) que um par atacante↔alvo pode permanecer em um determinado estado sem sofrer uma transição (mesmo que seja uma auto-transição). Este tempo é medido a partir da entrada de um par atacante↔alvo no conjunto de entidades do sistema de um estado. Toda vez que um par atacante↔alvo permanece em um estado por ($Time_{max}$), uma transição por temporizador é disparada para este par atacante↔alvo. $Time_{max}$ pode ser igual a 0 (zero), indicando que uma transição por temporizador deve ser disparada imediatamente. Alternativamente, $Time_{max}$ pode assumir um valor negativo, indicando que o temporizador não expira nunca.
- Toda vez que uma transição é disparada para um par atacante↔alvo este deve ser removido do conjunto de pares atacante↔alvo do estado de origem e acrescentado no conjunto de pares atacante↔alvo do estado de destino, com o temporizador incializado, exceto quando se tratar de transições para o estado final, quando o par atacante↔alvo deve apenas ser removido.
- A execução de uma ação consiste na geração de uma nova mensagem que pode ser: uma consulta, um estado de detecção ou um alerta. Esta mensagem é sempre gerada para o mesmo conjunto de pares atacante↔alvo da transição que disparou a execução da ação. Mensagens alerta e estado de detecção recebem como identificador de ataque o mesmo identificador do DEF e como identificador o mesmo identificador do estado de destino da transição.
- Um **evento** é consumido pelo módulo núcleo do IDS verificando-se se existe alguma transição disparada por ele em todos os estados de todos os DEF.

§ Uma mensagem **estado de detecção** é consumida pelo módulo núcleo do IDS adicionando-se todos os pares atacante↔alvo da mensagem ao conjunto de pares atacante↔alvo do estado cujo identificador for o mesmo do identificador da mensagem estado de execução, em um DEF cujo identificador de ataque seja o mesmo do identificador de ataque da mensagem.

Em seguida são apresentados exemplos de ataques contra o protocolo de roteamento e contra aplicações distribuídas que podem ser detectados usando-se um núcleo do IDS que utiliza uma abordagem de detecção por uso incorreto, conforme definição acima. Três exemplos de desenvolvimento completo de assinaturas de ataques são desenvolvidos, ilustrando as possibilidades de colaboração nos processos de gerenciamento de alertas (Fabricação + Personificação de mensagens HELLO do protocolo de roteamento OLSR, Figura 4-3), execução do algoritmo de detecção (Fabricação de mensagens HELLO do protocolo de roteamento OLSR, **Figura 4-2**) e coleta de dados (cadeias de conexões *telnet*).

5.3.1. Ataques e Assinatura de Ataques contra o Protocolo de Roteamento

Esta seção discute a detecção de ataques contra o protocolo de roteamento OLSR. Estes ataques foram apresentados na seção 4.3.1.1. A Tabela 5-1 mostra o tipo de assinatura de ataque que é desenvolvida para cada ataque em termos de DEF.

Tabela 5-1 – Assinaturas de Ataques contra o Protocolo OLSR

Ataque	**Mensagem OLSR**	**Informação de Roteamento Falseada**	**Informação de Origem na Mensagem Corrompida**	**Assinatura do Ataque**
Fabricação	HELLO	Neighbor List	Qualquer	Inconsistências na informação de roteamento entre mensagens HELLO diferentes.
Fabricação + Personificação	HELLO	Link-status	Endereço IP do nodo personificado	Anomalias no escalonamento do protocolo.
Fabricação	TC	MS list	Qualquer	Inconsistências na informação de roteamento entre mensagens HELLO diferentes.
Modificação + Personificação	TC	Sequence Number	Endereço IP do originador	Anomalias no escalonamento do protocolo.

Para a detecção dos três primeiros ataques mostrados na Tabela 5-1, a informação que necessita ser coletada (módulo sensor) é a tabela de vizinhos com os respectivos estados dos enlaces (symmetric, asymmetric), a tabela de vizinhos a dois saltos com os respectivos MPR e a tabela de MPRs. Estas tabelas são mantidas pelo *daemon* de roteamento OLSR e devem ser coletadas cada vez que houver uma mudança no seu conteúdo (e.g. mudança do estado de um

enlace, adição de um novo vizinho, remoção de um vizinho, etc.). Para a detecção do quarto ataque, pode-se coletar a tabela de números de seqüência de mensagens, igualmente mantida pelo *daemon*. Uma alternativa para coleta destas informações seria utilizar a MIB experimental SNMP definida para o OLSR [94][29]. Uma alternativa seria a utilização de uma interface de captura de rede, que possa capturar as mensagens HELLO e TC, promiscuamente, e as processe para obter as informações desejadas. O inconveniente desta abordagem consiste no fato de se repetir, no módulo sensor do L-IDS, o processamento que já é realizado pelo *daemon* de roteamento OLSR.

De uma maneira geral, as assinaturas para cada um dos ataques mostrados na Tabela 5-1 podem ser identificadas considerando-se:

- Fabricação de mensagens HELLO: Este ataque pode ser identificado através da verificação de inconsistências entre as tabelas de vizinhos OLSR de nodos vizinhos. De fato, qualquer nodo que possa escutar mensagens HELLO provenientes do atacante e de algum outro nodo que não seja escutado por esse pode detectar as inconsistências, pois o atacante anuncia o nodo que não é escutado por ele como seu vizinho, sendo que este não o é. Os candidatos para detecção do ataque são os antigos MPR dos nodos que selecionaram o atacante como novo MPR.
- Fabricação + Personificação de Mensagens HELLO: Este ataque pode ser identificado por anomalias no escalonamento do protocolo de roteamento, que ocorrem devido ao aparecimento de mais de uma mensagem HELLO em um mesmo período HELLO_INTERVAL, tendo um mesmo originador e anunciando o estado do enlace de um mesmo vizinho. Neste caso, o status do enlace deste vizinho altera-se entre “asymmetric” e “symmetric” ou “MPR” em um mesmo período HELLO_INTERVAL.
- Fabricação de Mensagens TC: Este ataque pode ser caracterizado pela presença de inconsistências anunciadas simultaneamente por nodos diferentes. Alguns nodos anunciados no MS do originador (i.e. o atacante) da mensagem falsa não são vizinhos deste nodo. Como a mensagem TC fabricada deve ser disseminada

[29] Uma MIB experimental para o OLSR foi proposta pelo autor em um trabalho prévio [94]. Entretanto, esta MIB não está padronizada, pois o IETF não iniciou ainda os trabalhos de padronização de MIB para protocolos de roteamento *ad hoc*. Entretanto, a se julgar pelo padrão MIB-II para outros protocolos de roteamento (e.g. OSPF) [75], é bastante provável que a tabela de vizinhos esteja na MIB a ser padronizada para o OLSR ou mesmo para todos os outros protocolos de roteamento *ad hoc*.

(*flooding*) a toda a rede, esta mensagem eventualmente deve chegar a esses nodos que não possuem o atacante em sua vizinhança. Esses nodos podem identificar o ataque.

- Modificação de Mensagens TC: Este ataque pode ser caracterizado por anomalias no escalonamento de mensagens do protocolo de roteamento, dado que tanto a mensagem correta quanto a mensagem modificada são disseminadas na vizinhança do nodo que originou a mensagem correta em um mesmo período TC_INTERVAL. Assim, tanto o originador real da mensagem quanto seus vizinhos podem detectar o ataque.

A seguir, o processo de especificação completa da assinatura de ataques, em termos da abstração de eventos e da definição do DEF, é mostrado para os dois primeiros ataques da Tabela 5-1.

5.3.1.1. Fabricação + Personificação de Mensagens HELLO

Este ataque é mostrado na Figura 4-3 do capítulo anterior, onde um adversário fabrica uma mensagem HELLO personificada, depois de receber uma mensagem HELLO legítima, anunciando todos os nodos que estavam na mensagem correta com status de enlace perdido ("lost"). O nodo personificado é o mesmo nodo que originou a mensagem HELLO correta. Este ataque é semelhante a ataques previamente identificados contra o protocolo de roteamento OSPF [113]. Este ataque é denominado NHOP, pois afeta os nodos a N = {1, 2, 3, ...} saltos do atacante.

A) Eventos

Durante um período de tempo HELLO_INTERVAL, uma das opções seguintes podem acontecer: (1) Nenhuma informação acerca de um determinado vizinho é recebida em mensagens HELLO (normal), (2) uma atualização acerca do status de enlace de um determinado vizinho é recebida durante um intervalo HELLO_MESSAGE (normal), ou (3) duas ou mais atualizações acerca do status de enlace de um determinado vizinho são recebidas em mensagens HELLO, sendo que uma delas acarreta em uma transição do status de enlace para "assymetric" e outra para "symmetric", em um mesmo período de tempo HELLO_INTERVAL. Pode-se, então, definir-se os seguintes eventos:

- NHOP_E1: O status do enlace foi alterado de "asymmetric" para "symmetric" (i.e. uma atualização informando o status "MPR" ou "symmetric" para um determinado

vizinho foi recebida). O atacante é o próprio nodo que teve seu status de enlace alterado. O alvo é um conjunto vazio. Isto é, o par atacante↔alvo é dado por: {identificador do nodo que teve o status do enlace alterado}↔{ }.

§ NHOP_E2: O status do enlace foi alterado de “symmetric” para “asymmetric” (i.e. uma atualização informando o status “lost” para um determinado vizinho foi recebida). O par atacante↔alvo é dado por: {identificador do nodo que teve o status do enlace alterado}↔{ }.

B) Abstração de eventos

A abstração de eventos é bastante simples: cada vez que uma atualização no status do enlace de um nodo é recebida, esta é comparada com o status anterior do enlace deste mesmo nodo.

C) Especificação da assinatura de ataque (DEF)

A **Figura 5-2** apresenta o DEF para a assinatura deste ataque. Ao receber um evento NHOP_E1, uma transição do estado inicial para o estado NHOP_S1 é disparada. Este estado possui um temporizador com $TIME_{max}$ igual a HELLO_INTERVAL. Uma transição para o estado final é disparada caso não sejam recebidos novos eventos indicando mudanças no status de enlace do nodo identificado como atacante (i.e. evento NHOP_E2). Se um ataque for realizado, um evento NHOP_E2 será gerado dentro do período HELLO_INTERVAL e uma outra transição para o estado final é gerada, desta vez executando uma ação que consiste na geração do alerta NHOP_A1 que indica a detecção positiva do ataque.

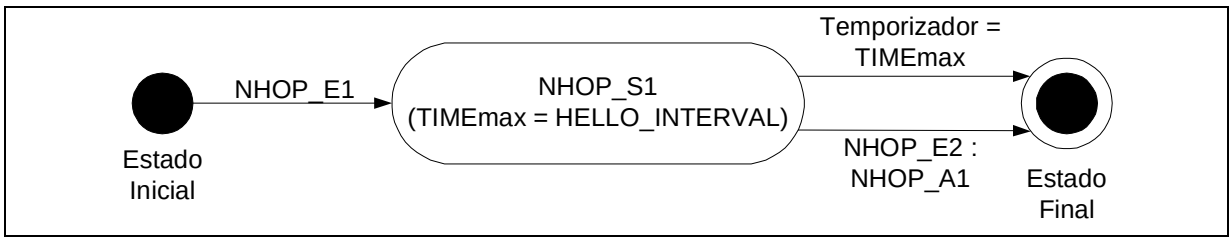


Figura 5-2 – Assinatura de Ataque: Fabricação + Personificação de mensagem HELLO

D) Cooperação no gerenciamento de alertas: correlação de alertas

Este ataque pode ser detectado por cada L-IDS que seja vizinho tanto do atacante quanto do nodo que está sendo personificado (nodo comprometido). Assim, se o módulo gerente de alertas gerar um alerta remoto que é difundido na rede, por exemplo, com um TTL = 2, todos os nodos que detectam localmente o ataque também recebem o alerta remoto

referindo-se ao mesmo ataque e ao mesmo par atacante↔alvo. Esses alertas podem ser facilmente correlacionados, pois se trata do mesmo tipo de ataque detectado por nodos diferentes, com evidencias contra um mesmo nodo atacante. Assim, esses alertas podem ser combinados em um único alerta de grupo, que identifica todos os nodos que coletam evidências contra um mesmo atacante, possibilitando uma resposta colaborativa à intrusão, como discutido mais adiante na seção 5.3.3.

5.3.1.2. Fabricação de Mensagens HELLO

Este ataque é mostrado na Figura 4-2 do capítulo anterior, onde um adversário fabrica uma mensagem HELLO anunciando todos os nodos previamente divulgados em qualquer mensagem HELLO recebida por ele, juntamente com um endereço adicional não utilizado, com status de enlace simétrico. Ao receber essa mensagem, todos os vizinhos do adversário escolhem-no como único MPR. Na detecção desse ataque é ilustrada a cooperação na execução do algoritmo de detecção. Este ataque é denominado N+1HOP, pois afeta os nodos a N + 1 = {2, 3, 4, ...} saltos do atacante.

A) Eventos e Estados de Detecção

A detecção deste ataque começa nos nodos que têm o seu conjunto de MPR alterado. Esses nodos geram um evento local (isLocal = true) N+1HOP_E1, indicando a alteração no conjunto de MPR, e têm como atacante o nodo MPR e como alvo todos os nodos a dois saltos com os quais o nodo MPR tem um enlace simétrico. Isto é, este evento tem o seguinte par atacante↔alvo: {identificador do nodo MPR}↔{identificadores dos vizinhos de dois saltos com enlace simétrico com o nodo MPR}. Este evento é consumido localmente e gera uma transição para um estado de detecção N+1HOP_S1 que é despachado na vizinhança de um salto do nodo (isLocal = false, isFlooded = true e TTL = 1, isClone = false). A detecção é interrompida neste nodo e prossegue nos nodos de destinos de N+1HOP_S1.

N+1HOP_S1 possui um temporizador com $TIME_{max}$ = 0, indicando que uma transição será disparada assim que este estado for ativado nos nodos de destino. Ao chegar nos nodos de destinos, N+1HOP_S1 dispara uma transição para N+1HOP_S2 (isLocal = true) e a execução de uma consulta local N+1HOP_C1, tendo o mesmo conjunto de par atacante↔alvo que N+1HOP_E1. Esta consulta pode gerar três tipos de eventos locais (isLocal = true):

- N+1HOP_E2: o nodo local (que executa a consulta) está na lista de alvos desta consulta, mas o nodo MPR não está entre os seus vizinhos de um salto. O seu par

atacante↔alvo é o seguinte: {identificador do nodo MPR}↔{identificadores dos vizinhos de dois saltos com enlace simétrico com o nodo MPR}.

- § N+1HOP_E3: o nodo local (que executa a consulta) está na lista de alvos desta consulta e o nodo MPR está entre os seus vizinhos de um salto. Este evento tem conjuntos vazios para atacante e alvo. Isto é, o seu par atacante↔alvo é o seguinte: {identificador do nodo MPR}↔{identificadores dos vizinhos de dois saltos com enlace simétrico com o nodo MPR}.
- § N+1HOP_E4: o nodo local não está na lista de alvos desta consulta. O seu par atacante↔alvo é o seguinte: {identificador do nodo MPR}↔{identificadores dos vizinhos de dois saltos com enlace simétrico com o nodo MPR}.

B) Abstração de eventos

Cada vez que ocorre uma alteração (inserção ou exclusão) na tabela de MPRs, gera-se o evento N+1HOP_E1 identificando quais são os vizinhos de dois saltos que têm o novo MPR como vizinho de um salto (tabela de vizinhos a dois saltos).

A abstração de eventos N+1HOP_E2, N+1HOP_E3 e N+1HOP_E4 consiste de verificar se o nodo local está listado em alvos da consulta N+1HOP_C1. Caso negativo o evento N+1HOP_E4 é gerado. Caso afirmativo, verifica-se a tabela local de vizinhos a um salto, buscando-se pelo identificador do novo MPR. Caso este se encontre nesta tabela, o evento N+1HOP_E3 é gerado. Caso contrário, gera-se o evento N+1HOP_E2.

C) Especificação da assinatura de ataque (DEF)

A Figura 5-3 apresenta o DEF para a assinatura deste ataque. Ao receber um evento N+1HOP_E1, uma transição do estado inicial para o estado N+1HOP_S1 é disparada. Este estado é despachado na vizinhança e a detecção prossegue nos nodos remotos que recebem N+1HOP_S1. Ao receber N+1HOP_S1, o nodo remoto executa uma transição imediata para N+1HOP_S2, fazendo a consulta N+1HOP_C1 (transição de temporizador com $TIME_{max} = 0$ para N+1HOP_S1). Caso seja recebido um evento N+1HOP_E3 ou N+1HOP_E4, é executada uma transição para o estado final. Em se recebendo um N+1HOP_E2, dispara-se uma auto-transição e executa-se a geração do alerta N+1HOP_A1 que indica a detecção positiva do ataque.

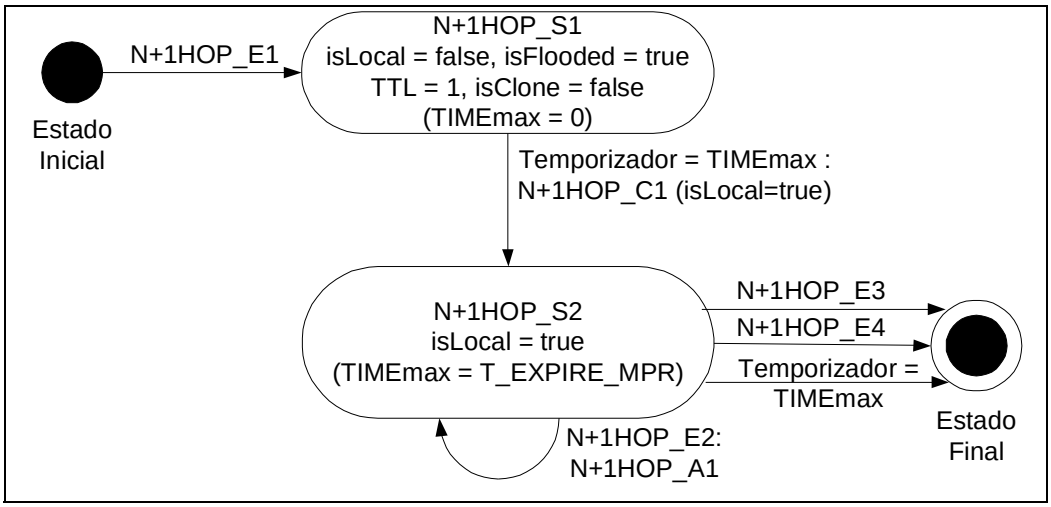


Figura 5-3 – Assinatura de Ataque: Fabricação de mensagem HELLO

5.3.2. Ataque e Assinatura de Ataques contra Aplicações Distribuídas

Esta seção discute a detecção por uso incorreto de ataques do tipo *stepping-stone*. O ataque em questão consiste no estabelecimento de uma cadeia de sessões *telnet.* Um nodo raiz (atacante) estabelece uma sessão com outro nodo. A partir desta sessão, estabelece-se então uma nova sessão para um outro nodo, e assim por diante. Esse tipo de ataque precede a realização de ações mais invasivas [105] (nível de aplicação). Todos os nodos que participam da cadeia colaboram para a detecção deste ataque na fase de coleta de dados.

A) Eventos

A detecção de ataques de cadeias de sessões de *telnet* é usualmente dividida em duas etapas. Primeiramente, quando um nodo recebe um pedido de abertura de sessão *telnet* (evento local STEPSTONE_E1). Este nodo faz, então, uma consulta remota (STEPSTONE_C1) ao nodo de origem da sessão (isLocal = false, isFlooded = false, destino = endereço do nodo que abre a sessão *telnet*), tendo como atacante o nodo de origem da conexão de *telnet* entrante local e como alvo o seu próprio endereço. A consulta STEPSTONE_C1 procura identificar se existem sessões de *telnet* entrantes no nodo que a executa. Se este não possui sessões *telnet* entrantes, um evento remoto (STEPSTONE_E2) é gerado tendo como destino todos os nodos alvo de STEPSTONE_C1. Caso exista alguma conexão entrante, um evento remoto (STEPSTONE_E3) é despachado para todos os nodos alvo de STEPSTONE_C1. Este evento tem como atacante o nodo de origem da conexão de *telnet* entrante local e como alvos o seu próprio endereço e os demais alvos da consulta

STEPSTONE_C1 anterior. O processo continua até que se identifique a origem da cadeia de sessões *telnet*, identificando-se o atacante.

O processo de detecção descrito acima diz apenas que existe uma cadeia de conexões entrantes e saintes desde um nodo raiz (atacante) até o último nodo da cadeira (que não abriu conexões *telnet* para outros nodos). Entretanto, não há garantias de que essas sessões nesta suposta cadeia estejam relacionadas entre si. Assim, a segunda etapa na detecção deste ataque consiste na avaliação da existência de correlações entre as sessões da cadeia. Descreve-se aqui apenas a assinatura de ataque para execução da primeira etapa da detecção. A segunda etapa pode ser realizada através de algum correlator estatístico tal como em [105].

B) Abstração de eventos

A abstração de eventos, para este ataque, pode ser feita realizando-se consultas periódicas STEPSTONE_PC1 (isLocal=true, período = T_{PC1}) para identificar a existência de conexões de *telnet* entrantes ativas. Para tanto, bastaria recuperar a tabela "tcpConnTable" a partir da MIB local do nodo, pois esta tabela é uma variável MIB padronizada [75]. Alternativamente, pode-se monitorar os pacotes TCP que chegam a um nodo com bit de SYN setado e tendo como porta de origem a porta notória do protocolo *telnet*. O procedimento para execução da SETPSTONE_C1 é exatamente o mesmo, tendo apenas como ressalva o fato desta consulta ser realizada em um nodo remoto.

C) Especificação da assinatura de ataque (DEF)

A Figura 5-4 apresenta o DEF para a assinatura deste ataque. Ao receber um evento STEPSTONE_E1, uma transição do estado inicial para o estado STEPSTONE_S1 é disparada e executa-se uma consulta STEPSTONE_C1 despachada para o atacante no evento STEPSTONE_E1. Ao se receber um evento STEPSTONE_E2, faz-se uma transição para o estado final. Caso um evento STEPSTONE_E3 ocorra, é feita uma transição para o estado STEPSTONE_S2. Neste estado, caso se receba um evento STEPSTONE_E2, declara-se um ataque, pela geração do alerta STEPSTONE_A1 ao se fazer uma transição para o estado final. Caso o evento recebido seja STEPSTONE_E3, uma auto-transição é executada, sem geração de alertas, mas com a geração de uma nova consulta STEPSTONE_C1.

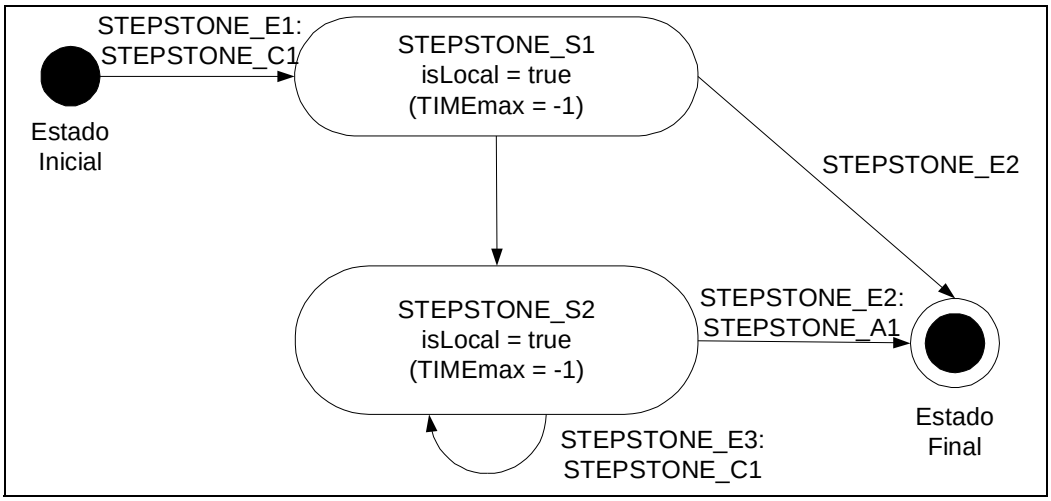


Figura 5-4 – Assinatura de Ataque: Stepping Stone para sessões *telnet*

5.3.3. Resposta a Intrusões

Nos três ataques para os quais uma assinatura de ataque é completamente descrita na sessão anterior, mais de um nodo pode detectar o ataque gerado por um mesmo adversário. Assim, uma correlação de alertas simples pode ser definida para implementação em um módulo gerente de alertas, permitindo reduzir a probabilidade de ocorrência de eventuais falsos positivos [116]. Este processo consiste em armazenar, por um período configurável de tempo, todos os alertas locais e/ou remotos que se referem a um mesmo ataque e a um mesmo atacante. Quando K alertas são coletados por um mesmo nodo, este pode gerar uma requisição de formação de coalizão, que contém os identificadores de todos os nodos que geraram alertas, formalizando o processo de resposta à intrusão através da revogação do certificado do atacante (Figura 3-3).

5.4. DETECÇÃO DE INTRUSÃO POR COMPORTAMENTO

As técnicas de detecção de intrusão baseadas em comportamento assumem que uma intrusão pode ser detectada pela observação de um desvio do comportamento normal ou esperado de um sistema ou usuário. O comportamento normal ou válido é extraído de informações prévias de referência sobre o sistema monitorado. Então, o IDS compara o modelo do comportamento de referência com a atividade corrente e gera alarmes cada vez que uma divergência em relação ao modelo original é observada. Isto significa que todo comportamento observado que não pode ser ajustado ao modelo de referência admitido previamente passa ser considerado como anomalia e possivelmente representam a ocorrência de um ataque.

Muitos sistemas baseados em modelagem do comportamento tem sido propostos e testados e, ainda que eles tenham diferentes características e arquiteturas, sua concepção e desenvolvimento podem ser, de maneira geral, descritos em termos de três fases [29]:

§ Construção de um modelo de comportamento normal ou válido: Este estágio consiste na modelagem do comportamento de referência do sistema. Neste estágio, a maioria das hipóteses acerca das fontes de informação que serão usadas e de como esses dados podem ser processados para construir um modelo que descreva, de forma consistente e desejavelmente completa a operação do sistema. De uma maneira sistemática, mas possivelmente particular, este estágio pode ser dividido em:

- Identificação de qual tipo de informação de auditoria deve ser usada para descrever o comportamento normal do sistema. Em geral, o mesmo tipo de informação usado neste estágio de modelagem deve ser usado (como entrada) no estágio de detecção. É importante notar ainda que um pré-processamento das informações coletadas é comumente requerido.
- Construção do modelo de comportamento. Vários tipos de modelos podem ser usados para descrever o comportamento normal de um sistema. Muitos sistemas foram desenvolvidos usando-se modelagem estatística [56], redes neurais ou algoritmos genéticos, além de diversas outras técnicas. Estes modelos têm, em geral, uma arquitetura prévia com diversos parâmetros que devem ser ajustados automaticamente usando-se algum tipo de algoritmo de aprendizagem ou otimização. Em alguns casos, a arquitetura e o algoritmo de aprendizado estão fortemente ligados, sendo este realizado através de um processo iterativo e adaptativo para o ajuste progressivo dos parâmetros do modelo. Nesse tipo de modelo existe ainda a possibilidade de se atualizar os parâmetros previamente ajustados para acompanhar as mudanças que ocorrem no comportamento normal do sistema. Outras abordagens, baseadas em sistemas especialistas, são igualmente possíveis, mas a atualização automática do modelo devida a mudanças do comportamento pode ser mais difícil.
- Obtenção de informação prévia de referencia (treinamento). Mesmo depois de se definir o tipo de informação a ser usado e a arquitetura do modelo, obter-se um bom conjunto de informações iniciais de referência não é tarefa evidente. De fato, as exigências que esses dados iniciais não

contenham qualquer tipo de utilização anômala e que esse conjunto de dados seja representativo de todo o comportamento normal do sistema são condições usualmente difíceis de satisfazer.

- Detecção: Esta fase consiste na realização de inferências acerca do estado de operação do sistema, comparando-se informações adquiridas do uso corrente do sistema com o modelo de comportamento ajustado no estágio anterior. Esses novos dados relativos à utilização posterior do sistema são apresentados ao IDS. A concepção do algoritmo de detecção pode variar em função do tipo de informação usado ou da arquitetura do sistema, mas também deve considerar outros critérios como desempenho e robustez, caso trate-se de um sistema que realize a detecção em tempo real. Independentemente do tipo de modelo utilizado, o algoritmo deve permitir uma clara definição para o desvio a ser avaliado. O desvio pode ser definido binariamente, isto é, para toda informação nova apresentada ao IDS este vai discriminá-la em normal ou anômala. Alternativamente, o desvio pode assumir a forma de um teste de significância, i.e. um comportamento observado pode ser avaliado em válido/inválido com uma probabilidade dada. Em muitos IDS que operam com técnicas de detecção por anomalia, requer-se que os alertas gerados nesta fase sejam pós-processados com objetivo de se eliminar falsos positivos. Isto ocorre, essencialmente, porque em detecção de intrusão por comportamento, ao contrário da detecção de intrusão por uso incorreto, não se tem o reconhecimento positivo de um ataque, mas apenas a indicação de uma atividade não observada durante a modelagem ocorreu.
- Atualização do modelo de comportamento assumido. Na medida em que o comportamento de utilização de um sistema muda, o modelo de comportamento deve ser atualizado para se evitar a indicação (alertas) errônea de anomalias pelo IDS. Esta atualização pode ser realizada continuamente, mas atualizações periódicas podem ser toleradas, mesmo em sistema que operem em tempo real. Usualmente, as atualizações do modelo de comportamento ocorrem gradativamente e em longo termo, evitando-se a ocorrência de adaptações distorcidas devido a uma utilização errônea por um curto período de tempo. Assim, se uma grande mudança do comportamento de utilização do sistema estiver para ser realizada, é necessário um reinício do sistema, construindo-se novamente o modelo de comportamento do sistema a partir de novas informações de referência que reflitam o novo comportamento, ou mesmo alterando-se aspectos da

arquitetura do modelo de comportamento do sistema para adaptá-lo à nova realidade. É importante salientar que essa atualização gradativa do sistema dá a oportunidade a um adversário de progressivamente induzir um comportamento errôneo ao sistema que será aprendido com um comportamento aceitável pelo mecanismo de atualização do modelo de comportamento válido. Esta é uma das desvantagens mais marcantes da abordagem de detecção de intrusão por comportamento.

Neste trabalho, deseja-se projetar um IDS por comportamento que seja complementar ao sistema de detecção e resposta a intrusões descrito na seção anterior. Utiliza-se uma abordagem de modelagem estatística para a construção do modelo de comportamento. Nesse tipo de abordagem, é usualmente necessário mapear-se os eventos de auditoria disponíveis para coleta e análise em variáveis aleatórias, isto é, em domínios numéricos, ainda que alguns eventos de auditoria já sejam observáveis nesta forma. Em um primeiro exercício, pretende-se observar valores numéricos que reflitam as condições de tráfego e uso da banda passante por determinados tipos de aplicações e protocolos em uma Manet. Nesse sentido, vale ressaltar que a modelagem estatística é usualmente complexa, uma vez que aplicações e protocolos diferentes possuem regras estatísticas bem distintas. Assim, decide-se construir um modelo de mistura de distribuições e ajustá-lo ao conjunto de dados candidato a caracterizar o tráfego normal de uma Manet.

Obviamente, a caracterização do que seria um perfil de tráfego normal para uma Manet ainda é um problema aberto e pouco consenso foi construído a este respeito. Assim, nossa modelagem deve ser ajustada a um perfil de tráfego considerado normal para uma Manet específica, cujas condições de operação estejam claramente definidas. Isto faz com que o trabalho ora apresentado tenha um caráter preliminar. Não obstante, e a exemplo de trabalhos similares nesta mesma ceara [16], os resultados obtidos preliminarmente permitem discriminar com sucesso ataques que se caracterizam por mudanças significativas nos padrões de tráfego, tais como ataques de DDoS e de *scanner* de rede. Desse modo, fica sendo este o objetivo principal do IDS baseado em comportamento descrito nesta seção.

5.4.1. Modelos de Mistura de Distribuições para Caracterização Estatística do Comportamento

Um modelo de mistura de distribuições é usualmente utilizado para modelar a função densidade de probabilidade (f.d.p.) de uma variável aleatória **y** d-dimensional[30] (cujas realizações são extraídas do domínio de informações de auditoria) e pode ser formalmente definido como se segue:

Seja $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n]^T$ um vetor de realização observável de **y**. Um modelo de mistura de distribuições para esses dados é definido expressando-se a f.d.p. dos dados como a combinação linear de funções nucleares básicas conforme mostrado na Eq. 5-1:

$$p(\mathbf{y}_i) = \sum_{k=1}^{K} w_k g_k(\mathbf{y}_i, \boldsymbol{\theta}_k) \qquad \text{Eq. 5-1}$$

onde: g representa cada função nuclear, w_k são os fatores de ponderação de cada função nuclear e q_k são os parâmetros das funções nucleares. O vetor $\boldsymbol{\Psi} = [w_1, w_2, ..., w_k, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_k]$ representa todos os parâmetros desconhecidos do modelo de mistura, os quais se deseja ajustar para enquadrar $\boldsymbol{\Psi}$ à **Y**. O número K é a ordem do sistema, geralmente fixo.

Este modelo de mistura finito vem sendo usado para modelar distribuições de diversos fenômenos supostamente aleatórios [76]. Um algoritmo iterativo para otimização, por um critério de máxima verossimilhança (ML) foi apresentado em [30] e é denominado algoritmo de estimação-maximização (EM).

Como o conjunto de dados de referência deve conter informações sobre diferentes comportamentos válidos, é normalmente útil que tais dados sejam *clusterizados*. O uso de modelos de mistura em *custerização* automática de dados é imediato, pela adoção de um modelo de mistura parametrizado [22,99]. Este modelo é definido assumindo-se que cada função nuclear individualmente representa a f.d.p. de cada *cluster* no conjunto de dados. Assim, um modelo de mistura de ordem K é diretamente aplicável em situações onde $\bar{Y}$ pode ser identificado como originário de uma mistura populacional de K grupos. Nestes casos, os coeficientes w_k equivalem à probabilidade de cada *cluster* p(k). Do mesmo modo, as probabilidades posteriores de cada realização $p(\mathbf{y}_i \mid k)$ podem ser obtidas, dado os valores de cada função nuclear em $\bar{y}$. Uma vez que $p(\mathbf{y}_i)$ pode ser diretamente estimada de Eq. 5-1, pelo teorama de Bayes, pode-se obter uma estimativa para as probabilidades posteriores na forma

[30] Um formalismo similar pode ser definido considerando-se um modelo de mistura de distribuições para a f.d.p. conjunta de "d" variáveis aleatórias unidimensionais.

$p(k|\ \mathbf{y}_i)=p(\mathbf{y}_i\ |k)p(k)/p(\mathbf{y}_i)$. O conhecimento prévio da ordem do modelo pode não estar disponível e é conveniente que este possa ser inferido automaticamente. Neste trabalho, desenvolve-se um algoritmo para determinação automática de K, baseado em uma otimização por um critério de maximização de entropia. Este algoritmo, adaptado de [99] para modelos paramétricos, é descrito na seção seguinte.

Para dados multivariados, o caso especial de funções nucleares gaussianas multivariadas forma um modelo conhecido como modelo de mistura de gaussianas (GMM). Este modelo, em particular, pode ser facilmente ajustado iterativamente pelo algoritmo EM, pois existem formas fechadas para a computação realizada em cada iteração. Além disso, o algoritmo possui boas propriedades de convergência, dado uma correta estimativa de K. Assim, neste trabalho, considera-se o caso de um GMM. Portanto, a descrição do algoritmo EM apresentada aqui está particularizada para este caso. Em [30,76] pode-se encontrar os detalhes de uma descrição mais genérica do algoritmo EM. Esta abordagem pode parecer um pouco restritiva, mas alguns pontos precisam ser destacados acerca do GMM. Em análise de *clusterização*, a aplicação de GMM parametrizados é largamente adotada, pois os *clusters* assumem formato elíptico. Entretanto, para um conjunto de dados contendo um grupo ou grupos de observações derivados de um número de populações normais maior que a ordem do sistema ou observações que não têm característica normal, modelos mais gerais precisam ser usados. Modelos paramétricos mais genéricos, usando as distribuições uniforme, gaussiana, gaussiana com deslocamento e escalonamento, além de distribuições t, podem ser facilmente derivados, pois o algoritmo EM já foi definido para esses casos [76,22,2]. Outro tipo importante de função nuclear para a qual seria interessante derivar-se o algoritmo EM são as distribuições de Pareto, largamente usadas para modelagem de tráfego intermitente, em rajada. Outra alternativa possível consiste na adoção de modelos de mistura semi-paramétricos [99], onde um modelo de mistura de ordem superior é ajustado aos dados e diferentes misturas das funções nucleares ajustadas pelo algoritmo EM são otimizadas para descrever a f.d.p. de cada *cluster* (i.e. a f.d.p. de cada *cluster* é formada por diferentes modelos de mistura de ordem alta, permitindo que a ordem do modelo seja mais alta e, portanto, mais genérica que o número de *clusters* nos dados.

Para o caso de GMM, g_k em Eq. 5-1 é substituído por $f(\mathbf{y}_i, \boldsymbol{\mu}_k, \mathbf{R}_k)$, que denota uma f.d.p. normal multivariada com média m_k e matriz de covariância $\mathbf{R}_k$. A Eq. 5-1 pode ser reescrita como Eq. 5-2:

$$p(\mathbf{y}_i) = \sum_{k=1}^{K} w_k f(\mathbf{y}_i, \boldsymbol{\mu}_k, \mathbf{R}_k) \qquad \text{Eq. 5-2}$$

onde: $\boldsymbol{\Psi} = [w_1, w_2, ..., w_K, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_K, \mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_K]$

5.4.1.1. Algoritmo EM

A estimação por máxima verossimilhança consiste de encontrar-se uma estimativa $\boldsymbol{\Psi}^*$ para $\boldsymbol{\Psi}$ que maximize a verossimilhança de $\mathbf{y}$ para um conjunto de observações $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n]^T$. Assumindo-se que $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n$ sejam realizações independentes[31] do vetor característico **Y**, a função de verossimilhança logarítmica como função de $\boldsymbol{\Psi}$ é dada por Eq. 5-3:

$$\log L(\boldsymbol{\Psi}) = \sum_{j=1}^{n} \log\left(\sum_{k=1}^{K} w_k f(\mathbf{y}_j, \boldsymbol{\mu}_k, \mathbf{R}_k) \right) \qquad \text{Eq. 5-3}$$

Uma estimativa de $\boldsymbol{\Psi}$ com máxima verossimilhança é dada pelas raízes da Eq. 5-3, que corresponde a um máximo local de Eq. 5-4:

$$\frac{\partial \log L(\boldsymbol{\Psi})}{\partial \boldsymbol{\Psi}} = 0 \qquad \text{Eq. 5-4}$$

Dado que é difícil otimizar $\boldsymbol{\Psi}$ diretamente, são introduzidas variáveis ocultas (não observadas) z_{jk}, onde z_{jk} é definido como 1 ou 0 se $\mathbf{y}_j$ é proveniente ou não do k-ésimo componente do modelo de mistura (j =1 ,..., n; k =1 ,..., K). O vetor de dados completo (não observado) **X** é formado por $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]^T$, onde $\mathbf{z}_j = [z_{j1}, z_{j2}, ..., z_{jK}]^T$ são os vetores de variáveis ocultas para uma realização $\mathbf{y}_j$ com $\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_n$ sendo realizações independentes de uma distribuição multinomial consistindo de um experimento em K categorias, com respectivas probabilidades $w_1, ..., w_K$. As realizações $\mathbf{x}_1 = (\mathbf{y}_1^T, \mathbf{z}_1^T)^T, ..., \mathbf{x}_K = (\mathbf{y}_K^T, \mathbf{z}_K^T)^T$ são consideradas independentes e identicamente distribuídas.

Para esta especificação, a função de verossimilhança logarítmica para o vetor complete **X** é dada por:

$$\log L_c(\boldsymbol{\Psi}) = \sum_{k=0}^{K} \sum_{j=1}^{n} z_{jk} \log\{w_k f(\mathbf{y}_j, \boldsymbol{\mu}_k, \mathbf{R}_k)\} \qquad \text{Eq. 5-5}$$

[31] Um modelo estocástico é mais realista em alguns casos (e.g. modelo de Markov), mas este modelo complica consideravelmente a computação, não sendo considerado neste estágio de desenvolvimento do trabalho.

O algoritmo EM [30] é efetivo quando maximizar a verossimilhança do vetor de dados completes (**X**) é mais simples que maximizar a verossimilhança dos dados incompletos (Eq. 5-3). O algoritmo EM é executado iterativamente e consiste de dois passos em cada iteração: passo E (estimação) e passo M (maximização). Considerando $\boldsymbol{\Psi}^i$ como uma estimativa de $\boldsymbol{\Psi}$ na i-ésima iteração, o passo E requer o cálculo de Eq. 5-6:

$$Q(\boldsymbol{\Psi};\boldsymbol{\Psi}^i) = E_{\boldsymbol{\Psi}^i}(\log L_c(\boldsymbol{\Psi}) \mid \mathbf{Y}) \qquad \text{Eq. 5-6}$$

onde: $Q(\boldsymbol{\Psi};\boldsymbol{\Psi}^i)$ é o valor esperado condicional de $\log L_c(\boldsymbol{\Psi})$, dado os dados observados **Y** e o ajuste atual $\boldsymbol{\Psi}^i$ para $\boldsymbol{\Psi}$.

Uma vez que $\log L_c(\boldsymbol{\Psi})$ é uma função linear das variáveis ocultas z_{jk}, o passo E é executado simplesmente substituindo-se z_{jk} por seu valor esperado condicional, dado $\mathbf{y}_j$, usando-se $\boldsymbol{\Psi}^i$ para $\boldsymbol{\Psi}$. Isto é, z_{jk} é substituído em Eq. 5-6 por Eq. 5-7:

$$t_k(\mathbf{y}_j;\boldsymbol{\Psi}^i) = E_{\boldsymbol{\Psi}^i}(z_{jk} \mid \mathbf{y}_j) = \frac{w_k^i f(\mathbf{y}_j, \boldsymbol{\mu}_k^i, \mathbf{R}_k^i)}{\sum_{k'=1}^{K} w_{k'}^i f(\mathbf{y}_j, \boldsymbol{\mu}_{k'}^i, \mathbf{R}_{k'}^i)} \qquad \text{Eq. 5-7}$$

Pode-se reconhecer $t_k(\mathbf{y}_j;\boldsymbol{\Psi}^i)$ na Eq. 5-7 como a estimativa corrente da probabilidade posterior da j-ésima realização ($\mathbf{y}_j$) ter vindo do k-ésimo grupo, i.e. p(k | $\mathbf{y}_j$). A Eq. 5-7 pode ser, então, reescrita como:

$$p(k \mid \mathbf{y}_i) = \frac{w_k^i f(\mathbf{y}_i, \boldsymbol{\mu}_k^i, \mathbf{R}_k^i)}{\sum_{k'=1}^{K} w_{k'}^i f(\mathbf{y}_i, \boldsymbol{\mu}_{k'}^i, \mathbf{R}_{k'}^i)} = \frac{p(k)\,p(\mathbf{y}_i \mid k)}{p(\mathbf{y}_i)} \qquad \text{Eq. 5-8}$$

Esta equação é una expressão do teorema de Bayes, reconhecendo-se que a estimativa da probabilidade *a priori* de cada *cluster* ($p(k)$) é dada pela estimativa corrente do fator de ponderação w_k^i.

Substituindo-se Eq. 5-8 em Eq. 5-7, obtêm-se a expressão para o passo E:

$$Q(\boldsymbol{\Psi};\boldsymbol{\Psi}^i) = \left(\sum_{j=1}^{n} \log \sum_{k=1}^{K} w_k^i f(\mathbf{y}_j, \boldsymbol{\mu}_k^i, \mathbf{R}_k^i) \right) / n = \left(\sum_{j=1}^{n} \log(p(\mathbf{y}_j)) \right) / n \qquad \text{Eq. 5-9}$$

No passo M na (i +1)-ésima iteração, o objetivo é escolher $\boldsymbol{\Psi}^{i+1}$ que maximize $Q(\boldsymbol{\Psi};\boldsymbol{\Psi}^{i})$. Assim, o ajuste atual para as proporções de mistura (w_k^{i+1}), os componentes de média ($\boldsymbol{\mu}_k^{i+1}$) e as matrizes de covariância ($\mathbf{R}_k^{i+1}$) são dadas explicitamente por Eq. 5-10:

Eq. 5-10

$$w_k^{i+1} = \sum_{i=1}^{n} p(k \mid \mathbf{y}_i)/n$$

$$\boldsymbol{\mu}_k^{i+1} = \sum_{i=1}^{n} p(k \mid \mathbf{y}_i)\mathbf{y}_i / \sum_{i=1}^{n} p(k \mid \mathbf{y}_i)$$

$$\mathbf{R}_k^{i+1} = \sum_{i=1}^{n} p(k \mid \mathbf{y}_i)(\mathbf{y}_i - \boldsymbol{\mu}_k^{i+1})(\mathbf{y}_i - \boldsymbol{\mu}_k^{i+1})^T / \sum_{i=1}^{n} p(k \mid \mathbf{y}_i)$$

Uma característica interessante do algoritmo EM é que a verossimilhança da mistura $L(\boldsymbol{\Psi})$ não pode nunca decrescer após uma seqüência EM. Assim, $L(\boldsymbol{\Psi}^{i+1}) \geq L(\boldsymbol{\Psi}^{i})$, implicando na convergência de $L(\boldsymbol{\Psi})$ para um certo valor L*, se a seqüência de valores para a verossimilhança for limitada. Os passos E e M são alternados repetidamente até a verossimilhança ou as estimativas para os parâmetros mudem de um valor arbitrariamente pequeno, indicando a convergência do algoritmo.

O algoritmo EM pode ser sintetizado da seguinte forma:

Algoritmo 2 – EM
1: Inicia-se $\boldsymbol{\Psi}^0$ com valores aleatórios para $w_1^0, w_2^0, ..., w_K^0, \boldsymbol{\mu}_1^0, \boldsymbol{\mu}_2^0, ..., \boldsymbol{\mu}_K^0, \mathbf{R}_1^0, \mathbf{R}_2^0, ..., \mathbf{R}_K^0$
2: Para i = 0, calcula-se L^i conforme a Eq. 5-9.
3: Para i =i+1, calcula-se $\boldsymbol{\Psi}^{i+1}$ (Eq. 5-10) e L^{i+1} (Eq. 5-9)
4: Se $L^{i+1} - L^i > \delta$ (constante de convergência), repete-se 3
5: Atualiza-se os valores reais dos parâmetros $\boldsymbol{\Psi}^* = \boldsymbol{\Psi}^i$.

5.4.1.2. Principais problemas na aplicação do algoritmo EM e soluções propostas

O primeiro problema na aplicação do algoritmo EM, conforme descrito na última seção, está relacionado com o fato da função de verossimilhança ter, em geral, múltiplos máximos locais. Assim, diferentes iniciações podem levar a diferentes modelos ajustados, correspondentes a máximos distintos da função. Isto é especialmente crítico no caso de iniciações aleatórias, uma vez que qualquer máximo local pode ser atingido resultando em ajustes sub-ótimos ou mesmo inadequados. Para tratar deste problema, diversos procedimentos de iniciação são propostos [76,39,22]. Entre eles, uma solução imediata consiste em se fazer um número fixo (C_{max}) de iniciações aleatórias para cada aplicação do

algoritmo EM, e utilizar aquela que resulta em um máximo valor para o passo E, após a convergência. A utilização de uma *pré-clusterização* [76,39] também pode ser usada (o que provê valores estimados para as probabilidades, *a priori*, de cada *cluster*).

Em adição, no caso de componentes das matrizes de covariância não terem restrições (i.e. considera-se que esta matriz é uma matriz cheia e não uma matriz diagonal, por exemplo), a função de verossimilhança não é limitada, dado que cada ponto de dados acarreta em uma singularidade no vértice do espaço de parâmetros [76]. Além disso, cuidados especiais devem ser tomados para os casos onde um componente (*cluster*) ajustado possui uma variância generalizada (i.e. o determinante da matriz de covariâncias) muito pequena (não nula), o que acarreta em valores relativamente grandes para o máximo local. Este componente corresponde a um *cluster* contendo poucos pontos, que estão relativamente próximos uns dos outros. Existe, portanto, uma necessidade de se monitorar o tamanho relativo das proporções das misturas ajustadas, os componentes das variâncias generalizadas com objetivo de eliminar estes máximos locais espúrios. Existe também uma necessidade de se monitorar as distâncias euclidianas entre as médias de componentes ajustados, com objetivo de verificar se os *clusters* implicados representam uma separação real entre as médias ou se trata-se de um ou mais *clusters* que caíram quase em um sub-espaço do vetor característico original [76].

Dadas as adaptações e cuidados mencionados nos parágrafos anteriores, uma versão modificada do algoritmo EM é sumarizada a seguir:

Algoritmo 3 – EM Modificado
1: Inicia-se um contador c = 0
2: c = c+1
3: Inicia-se $\boldsymbol{\Psi}^0$ com valores aleatórios para $w_1^0, w_2^0, ..., w_K^0, \boldsymbol{\mu}_1^0, \boldsymbol{\mu}_2^0, ..., \boldsymbol{\mu}_K^0, \mathbf{R}_1^0, \mathbf{R}_2^0, ..., \mathbf{R}_K^0$
4: Para i = 0, calcula-se L^i conforme a Eq. 5-9.
5: Para i =i+1, calcula-se $\boldsymbol{\Psi}^{i+1}$ (Eq. 5-10) e L^{i+1} (Eq. 5-9)
6: Se $L^{i+1} - L^i > \delta$ (constante de convergência), repete-se 5
7: Se o determinante de qualquer uma das matrizes de covariâncias < ε (uma constante pequena), repete-se 2
8: Se (c = 1) ou ($L^i > L_{opt}$) então faz-se $L_{opt} = L^i$ e $\boldsymbol{\Psi}_{opt} = \boldsymbol{\Psi}^i$
9: Se c <= C_{max}, repete-se 2
10: Atualiza-se os valores reais dos parâmetros $\boldsymbol{\Psi}^* = \boldsymbol{\Psi}_{opt}$.

5.4.1.3. Estimação automática da ordem ótima do modelo

Para os propósitos do algoritmo EM, a ordem do modelo (K) deve ser assumida *a priori*. Considerando, no entanto, que, em muitos casos, o número de partições não é

conhecido *a priori*, é útil que se tenha um mecanismo para se descobrir o número de partições mais provável, para um dado modelo. O objetivo aqui consiste em se construir uma estimativa para K que implique em uma "partição ideal", isto é, p(k | $\mathbf{y}_i$) é próximo da unidade para um valor de k e próximo de zero, para todos os outros valores, para cada realização $\mathbf{y}_i$. Como descrito em [99], esta partição ideal deve ser obtida pela minimização da entropia de Shannon dado os dados observados[32] (**Y**), que deve ser avaliada para cada observação como Eq. 5-11:

$$H_K = -\sum_{k=1}^{K} p(k \mid \mathbf{y}_i) \log(p(k \mid \mathbf{y}_i)) \qquad \text{Eq. 5-11}$$

O valor esperado desta entropia é avaliado tirando-se a média de H_K sobre todos os dados observados Eq. 5-12:

$$E^*(H_K) = -\sum_{i=1}^{n} \sum_{k=1}^{K} p(k \mid \mathbf{y}_i) \log(p(k \mid \mathbf{y}_i))/n \qquad \text{Eq. 5-12}$$

onde: E^* denota o estimador da esperança.

Então, procede-se ao ajuste de K_{max} modelos com ordens diferentes (K = 1, 2, ..., K_{max}) e avalia-se a entropia esperada (Eq. 5-12) para cada um deles. O modelo que resultar em uma medida mínima é considerado como o modelo ótimo.

O algoritmo EM com estimação automática de ordem ótima é sumarizado a seguir:

Algoritmo 4 – EM com Estimação de Ordem Ótima
1: Inicia-se K = 0, H_{opt} = 0, K_{opt} = 1
2: K = K+1
3: Ajusta-se o modelo de ordem K aos dados **Y**, usando-se o algoritmo EM modificado (algoritmo 3).
4: Estima-se a esperança de H_K (Eq. 5-12)
5: Se (K = 1) ou (H_K < Hopt) então faz-se H_{opt} = H_K ; K_{opt} = K; e $\mathbf{\Psi}_{opt} = \mathbf{\Psi}^*$
6: Se K < K_{max} (uma constante fixa), repete-se 2
7: Atualiza-se a ordem real do modelo com o valor ótimo: K = K_{opt}
8: Atualiza-se os valores reais dos parâmetros $\mathbf{\Psi}^* = \mathbf{\Psi}_{opt}$.

5.4.1.4. Algoritmo de detecção

Durante a fase de detecção, o modelo de comportamento já está computado e disponível para a realização de inferências sobre nodos dados apresentados ao sistema. O

[32] Este argumento pode ser facilmente verificado por simples inspeção da expressão para a entropia. Um tratamento formal pode ser encontrado em [9].

objetivo consiste em definir alguma penalidade λ (e.g. $0 \leq \lambda \leq 1$), indicando o grau de normalidade de uma realização de certamente anômalo ($\lambda = 0$) a certamente normal ($\lambda = 1$).

Muitas abordagens diferentes para definir este critério para o modelo estatístico de comportamento representado pela Eq. 5-1 são possíveis. Neste trabalho, define-se um procedimento de detecção formado por duas etapas: uma classificação (Bayesiana) e uma inferência acerca da pertinência em um determinado *cluster*.

A classificação é direta para modelos de mistura parametrizados e consiste da avaliação das probabilidades posteriores de cada *cluster* condicionadas ao novo dado **y'**, isto é, p(k | **y'**) (Eq. 5-8) para k = (1, 2,..., K).

A inferência acerca da pertinência a um *cluster* específico é um pouco mais complexa. Todas as funções nucleares de distribuição usadas em nosso modelo são contínuas por natureza. Assim, considerar-se a probabilidade posterior do novo dado, condicionada à probabilidade do *cluster* p (**y'** | k) pela simples avaliação da f.d.p. no novo ponto não tem significado prático. Uma abordagem mais realista consiste em avaliar a probabilidade do novo dado estar contido em algum intervalo de pertinência (Π_k), definido como uma função da nova observação **y'** e dos parâmetros da distribuição do *cluster* (e.g. $\mathtt{m}_k$ e $\mathbf{R}_k$), o que pode ser formalmente expresso como Eq. 5-13:

$$p(\mathbf{y}' \in \Pi_k \mid k) = \int_{\Pi_k} g_k(\mathbf{y}, \boldsymbol{\theta}_k) d\Pi_k \qquad \text{Eq. 5-13}$$

De fato, a probabilidade definida na Eq. 5-13 se parece com uma função de distribuição acumulativa (f.d.a), se Π_k for definido conforme mostrado na Eq. 5-14 abaixo [59]:

$$\Pi_k = \left\{ \mathbf{y} \in \Re^d \mid \frac{\left\| (\mathbf{y} - \boldsymbol{\mu}_k) \right\|^2}{\left\| \mathbf{R}_k \right\|} \geq g^2 \right\} \qquad \text{Eq. 5-14}$$

onde: $\| \ \|^2$ e $\| \ \|$ denotam algum tipo de operadores para calculo da norma e γ é uma constante que depende de **y'**.

Finalmente, a função de penalidade para a detecção pode ser definida, conforme a Eq. 5-15:

$$l(\mathbf{y}') = \sum_{k=1}^{K} p(k \mid \mathbf{y}') p(\mathbf{y}' \in \Pi_k \mid k) \qquad \text{Eq. 5-15}$$

5.4.1.5. Algoritmo de detecção para operação em tempo-real com GMM

O procedimento para a construção do modelo de comportamento de referência é usualmente executado *off-line*. Restrições acerca da complexidade computacional não são severas neste estágio. Entretanto, é usualmente desejável que os estágios de detecção e atualização do modelo de comportamento possam ser executados continuamente. Assim, os algoritmos para detecção e atualização do modelo devem ser projetados para operação em tempo-real. Nesta seção, mostra-se como o processo de detecção pode ser computado, em tempo real.

A Eq. 5-13 não pode ser sempre avaliada analiticamente. Uma solução geral seria avaliar esta equação integral numericamente, mas isto pode ser proibitivo, pois tal avaliação numérica é computacionalmente intensiva mesmo nos casos unidimensional ou bidimensional, fazendo a execução em tempo-real difícil ou mesmo impossível [40]. De fato, a Eq. 5-13 pode ser difícil para funções nucleares arbitrárias g_k, um algoritmo computacionalmente eficiente para avaliação desta integral por ser estabelecido no caso especial de distribuições Gaussianas. Assim, quando utiliza-se um GMM, a avaliação da Eq. 5-13 pode ser feita escolhendo-se convenientemente os elementos indefinidos desta equação, isto é, o operador de norma e γ. Define-se, portanto, Π_k como o espaço complementar (côncavo) da elipsóide de isodensidade (em $\hat{A}^d$), cuja fronteira contém **y'** e tem como centro de gravidade m_k. Isso significa que Π_k é limitado internamente por uma superfície elipsoidal d-dimensional, formada por todos os pontos que possuem o mesmo valor de densidade que **y'** (i.e. $f(\mathbf{y},\boldsymbol{\mu}_k,\mathbf{R}_k) = f(\mathbf{y}',\boldsymbol{\mu}_k,\mathbf{R}_k)$). Assim, reescrevendo-se a Eq. 5-14, tem-se a Eq. 5-16:

$$\Pi_k = \left\{ \mathbf{y} \in \Re^d \mid \sum_{ab} (y_a - m_a)\left[R_k^{-1}\right]_{ab} (y_b - m_b) \geq g^2 \right\} \qquad \text{Eq. 5-16}$$

onde: $\mathbf{y} = (y_1, y_2, ..., y_d)^T$; $\boldsymbol{\mu} = (m_1, m_2, ..., m_d)^T$; $\left[R_k^{-1}\right]_{ab}$ é o elemento da α-ésima linha e da β-ésima coluna da matriz de covariância inversa, e γ é dada pela Eq. 5-17:

$$g^2 = \sum_{ab} (y'_a - m_a)\left[R_k^{-1}\right]_{ab} (y'_b - m_b) \qquad \text{Eq. 5-17}$$

Essa estratégia pode ser ilustrada para os espaços uni e bidimensionais, conforme mostrados nas Figura 5-5 e Figura 5-6, respectivamente. Esta última foi desenhada para uma distribuição Gaussiana bivariada, com matriz de covariância diagonal (não correlacionada).

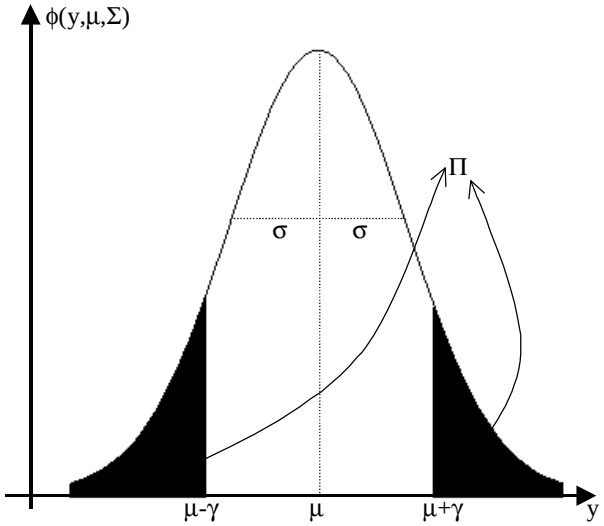


Figura 5-5 - Π para um *cluster* com distribuição Gaussiana unidimensional

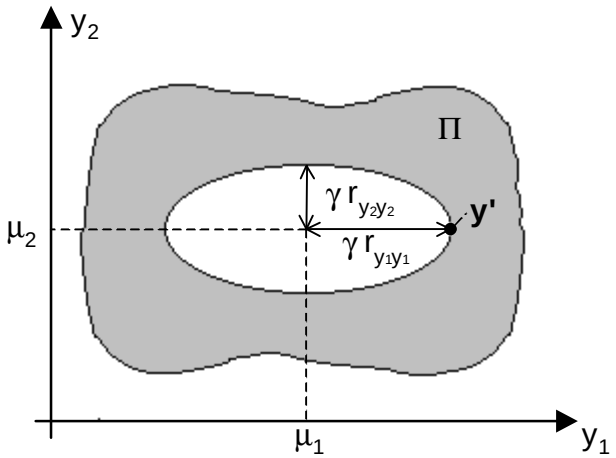


Figura 5-6 - Π para um *cluster* com distribuição Gaussiana bivariada e matriz de covariância diagonal

Este procedimento pode ser usado inclusive no cada de distribuições Gaussianas multivariadas, com matriz de covariância sem restrições, dado que é sempre possível encontrar uma transformação linear que mapeie uma distribuição Gaussiana multivariada qualquer em uma distribuição Gaussiana descorrelacionada (matriz de covariância diagonal) com o mesmo valor de γ[33].

Como os dados observados podem pertencer a um espaço multidimensional ($\hat{A}^d$), uma distância generalizada γ' é definida na Eq. 5-18. Isto possibilita a normalização das

[33] Este procedimento é chamado de análise de componente principal (PCA) [60] e é equivalente à rotação dos eixos coordenados nas direções principais. A matriz de rotação é formada pelos autovetores da matriz de covariância original, cujos autovalores correspondem às variâncias da nova distribuição. A translação usada para posicionar a origem dos eixos principais no ponto de média é implícita na Eq. 5-17.

probabilidades expressas na Eq. 5-13 para dados pertencentes a espaços dimensionais diferentes, o que permite a redução da computação para o espaço unidimensional, o que pode ser realizado por um procedimento simples de *lookup* em tabela.

$$g' = g / \sqrt{d} \qquad \text{Eq. 5-18}$$

5.4.1.6. Atualização recursiva dos parâmetros ajustados do modelo

Como o comportamento na utilização dos sistemas de informação muda constantemente, o modelo de comportamento de referência deve também ser atualizado para evitar falsos positivos. A atualização deve ser considerada como uma adaptação do modelo original com objetivo de acomodar variações suaves no comportamento do sistema, dado que o modelo pode tornar-se inválido ou incompleto no caso de mudanças expressivas.

Na abordagem proposta neste trabalho, considera-se a possibilidade de se atualizar o modelo de comportamento através da atualização recursiva e contínua dos parâmetros do modelo. Assim, a atualização é realizada nas probabilidades dos *clusters* (w_k) e nos parâmetros das distribuições nucleares. Utiliza-se estimadores usuais para a contínua estimação destas estatísticas do modelo [110]. É importante salientar que tanto a verossimilhança logarítmica quanto a entropia podem ser igualmente estimadas e comparadas com valores prévios (e.g. valores obtidos após a fase de treinamento), dado que esses valores podem fornecer uma idéia de "quão bom" está o novo modelo, quando comparado com o modelo de referência. Os estimadores para a atualização recursiva e contínua das probabilidades, *a priori*, de *clusters* e dos momentos de primeira e segunda ordem da distribuição são mostrados na Eq. 5-19.

$$\begin{aligned} w^{i+1} &= w^i(1-h_1) + h_1 p(z_k \mid \mathbf{y}_i) \\ \boldsymbol{\mu}_k^{i+1} &= \boldsymbol{\mu}_k^i(1-h_2 p(z_k)^{i+1}) + h_2 p(z_k)^{i+1}\mathbf{y}_i \\ \mathbf{R}_k^{i+1} &= \mathbf{R}_k^i(1-h_3 p(z_k)^{i+1}) + h_3 p(z_k)^{i+1}(\mathbf{y}_i - \boldsymbol{\mu}_k^{i+1})(\mathbf{y}_i - \boldsymbol{\mu}_k^{i+1})^T \end{aligned} \qquad \text{Eq. 5-19}$$

Estes estimadores são aplicáveis apenas nos casos onde as mudanças do comportamento acontecem a longo termo e o sistema, aplicação e/ou usuários mantêm-se estáveis. As constantes η_1, η_2 e η_3 devem ser cuidadosamente escolhidas para evitar instabilidades ((1/n pode ser uma primeira escolha para η_1 e valores ainda menores devem ser usados para η_2 e η_3, pois mudanças nos momentos da distribuição têm mais energia que mudanças na probabilidades *a priori* de *clusters*).

5.4.2. Caracterização de Tráfego Normal em uma Manet e Construção do Modelo de Comportamento Normal

Deseja-se construir um modelo de comportamento para caracterizar as condições normais de tráfego em uma Manet. De uma maneira geral, não há um consenso sobre qual seria um perfil de tráfego que possa ser considerado típico em Manet. De fato, com exceção de alguns protocolos de *controle* e sinalização da rede (e.g. roteamento) que estão presentes em quase todas as Manets, é provável que cada rede desse tipo tenha um perfil de tráfego que seja dependente da aplicação para a qual a rede foi projetada. Assim sendo, a caracterização do que seria um tráfego normal para uma Manet deve ser realizada caso a caso, ajustando-se o modelo de comportamento normal a uma situação específica, referente a uma aplicação definida da rede.

Outro aspecto que merece destaque consiste no fato de ser difícil se obter amostras reais do tráfego de uma Manet em operação, que sejam comprovadamente livres de traços de possíveis intrusões. Assim, uma alternativa que tem sido muito adotada em trabalhos similares acerca de Manet consiste na realização de simulações. Uma grande vantagem desse tipo de abordagem consiste em se simular diversos fatores tais como mobilidade e utilização da rede, de forma repetível e controlada. Entretanto, sempre cabe questionar a validade de simulações quando se pretende na realidade a modelagem de ambientes reais, que possuem muitos fatores que não tenham sido considerados de maneira adequada na simulação. Pela dificuldade de se montar uma Manet real, utiliza-se neste trabalho uma caracterização do perfil de tráfego normal derivado de uma simulação. Cabe ressaltar que o processo de treinamento do modelo de comportamento e o processo de detecção de intrusão são exatamente os mesmos casos os dados reais acerca do tráfego na rede estejam disponíveis. Assim, pretende-se validar um processo de detecção de intrusão por comportamento, usando-se dados simulados e, em trabalhos posteriores, aplicar-se este processo a situações mais reais, onde os dados reais de tráfego para treinamento e detecção de intrusão estejam disponíveis.

Por se tratar de uma simulação, três aspectos são definidos para caracterização do tráfego gerado na Manet: tráfego de controle, as aplicações que são executadas nos nodos e o modelo de mobilidade dos nodos:

- Tráfego de controle: consiste basicamente do tráfego gerado pelo protocolo de roteamento (UDP), além do tráfego ARP. Não se considera o tráfego gerado pelo protocolo de autoconfiguração, pois a rede simulada tem um número fixo de nodos, não sendo consideradas novas adesões ou partidas da rede. Também não considera-se tráfego de DNS, pois esta é uma questão ainda aberta em Manet.

- Aplicações executadas nos nodos: para se ter um cenário suficientemente representativo, considera-se a utilização de quatro tipos de tráfego gerado por diferentes aplicações em todos os nodos da rede simulada. São elas: sessão remota simples (e.g. telnet), transferência de dados em rajadas (e.g. FTP), transferência contínua de dados com taxa de bits constante (CBR) (e.g. videoconferência ou áudio-conferência) e aplicação simples de pergunta-resposta assíncrona (e.g. ping). Para cada um desses tipos de tráfego são definidas ainda algumas condições para distribuição do tráfego em toda a rede. Esses parâmetros são ajustados para se ter uma ocupação média dos enlaces sem fio em torno de 20% da sua capacidade.
    - Sessão remota simples (telnet)
        - utiliza o TCP;
        - o tráfego gerado é bidirecional;
        - intervalo entre mensagens: processo de Poisson;
        - múltiplas sessões entre origens/destinos diferentes, sendo os nodos de origem e destino (uniformemente distribuído), o tempo de início (processo de Poisson) e a duração da seção (normalmente distribuída) aleatoriamente definidos.
    - Transferência de dados em rajada (FTP)
        - utiliza o TCP;
        - tamanho do “arquivo” aleatório (normalmente distribuído);
        - múltiplas transferências entre origens/destinos diferentes, sendo os nodos de origem e destino (uniformemente distribuído) e o tempo de início (processo de Poisson).
    - Transferência de dados cbr (videoconferência)
        - utiliza o UDP;
        - taxa cbr fixa de 128kbps;
        - múltiplas transferências entre origens/destinos diferentes, sendo os nodos de origem e destino (uniformemente distribuído), o tempo de início (processo de Poisson) e a duração da seção (normalmente distribuída) aleatoriamente definidos.
    - Aplicação simples de pergunta-resposta assíncrona (ping)
        - utiliza o ICMP;
        - sempre envia 4 requisições, espaçadas no tempo de 1 segundo;
        - sempre envia-se resposta;

- múltiplas transferências entre origens/destinos diferentes, sendo os nodos de origem e destino (uniformemente distribuído) e o tempo de início (processo de Poisson) aleatoriamente definidos.

- Modelo de mobilidade: adota-se o modelo *random waypoint algorithm* desenvolvido pela CMU que possibilita uma distribuição uniforme dos nodos em uma área pré-definida, usualmente retangular. Utiliza-se, para efeitos de simulação, uma Manet com 50 nodos em uma área de 250m x 250m e um alcance de transmissão de 50m, resultando em uma vizinhança média de 6,28 nodos.
- O tempo de simulação é de 1.000 segundos.

Um modelo de mistura de gaussianas pode ser ajustado para as condições de tráfego geradas de acordo com as premissas definidas acima, bastando para isso definir as variáveis que são monitoras. Exemplos de variáveis que podem ser monitoradas são: definidas acima, considerando-se as seguintes variáveis monitoradas: taxa de conexões/sessões entrantes, duração de uma sessão, número de pacotes recebidos com erros, etc. Pode-se observar que essas variáveis são correlacionadas de maneira característica para as aplicações consideradas.

5.4.3. Detecção de Ataques de DDoS e Scanner de Portas

Diversos tipos de ataques de DDoS vem sendo criados nos últimos tempos. De um modo geral, para realização desses ataques, um adversário deve ter comprometido um determinado número de alvos, nos quais são instaladas as ferramentas de geração do ataque. O ataque, propriamente dito, ocorre em uma segunda fase, quando as ferramentas instaladas em todos os alvos disponíveis geram um tráfego excessivo contra um novo alvo particular, inundando-o de pacotes de tráfego espúrio e provocando a indisponibilidade do alvo por sobrecarga na rede e em sua capacidade de processamento. Este trabalho interessa-se em detectar apenas a segunda fase do ataque, isto é, quando diversos nodos estão gerando tráfego espúrio para um mesmo nodo alvo que se deseja tornar indisponível.

Um apanhado das principais ferramentas e respectivos ataques de DDoS conhecidos[34] pode ser visualizado na Tabela 5-2.

No caso de ataques de scanner de portas, tem-se, de maneira similar, uma geração excessiva de tráfego ilegítimo contra o alvo que está sendo escaneado, com a diferença que este tráfego não é necessariamente originado em múltiplos nodos, como no caso do DDoS.

[34] Uma lista atualizada de ferramentas para ataques de DDoS, análises técnicas e links úteis pode ser acessada em http://staff.washington.edu/dittrich/.

Entretanto, é possível que se tenha um ataque de scanner de portas distribuído, onde as informações acerca do nodo escaneado são coletadas a partir de nodos distintos. A caracterização do tráfego gerado por ataques de scanner de porta é mostrada na Tabela 5-3.

Tabela 5-2 – Caracterização do Tráfego Gerado por Ataques de DDoS

Ataque de DDoS	**Tipo de Tráfego Gerado**
smurf	inundação de pacotes ICMP echo-reply
trinoo	inundação de datagramas UDP em portas aleatórias
TFN e TFN2K	inundação de pacotes ICMP, UDM e TCP syn (flag syn setado); pacotes errôneos; smurf
TFN2K (ping *flood*)	inundação pacotes ICMP e smurf
TFN2K Targa 3	Pacotes IP inválidos
stacheldraht v.2.666	inundação de pacotes ICMP, UDP, TCP syn (flag syn setado), TCP null (nenhum flag setado), TCP ack (flag ack setado) e smurf
shaft	inundação de pacotes ICMP, UDP, TCP syn (flag syn setado)
mstream	inundação de pacotes TCP ack (flag ack setado)

Tabela 5-3 – Caracterização do Tráfego Gerado por Scanner de Portas

Ataque de Scanner de Portas	**Tipo de Tráfego Gerado**
Scanner de portas TCP	pedidos sucessivos de abertura de conexões TCP, em portas diferentes
Scanner de portas UDP	datagramas UDP sucessivos, em portas diferentes.

Para que se possa fazer a detecção destes ataques usando-se o modelo de detecção por comportamento proposto nas seções anteriores, são criados modelos de comportamento (e detecção) separados para o tráfego TCP, UDP, ICMP e IP. Conforme mostrado na Tabela 5-4, para cada modelo monitora-se um conjunto de variáveis pertinentes. No que diz respeito às variáveis consideradas para monitoração, utiliza-se variáveis padronizadas da MIB II [16], facilitando a coleta dessas informações, pois tais informações podem estar facilmente disponíveis nos nodos da Manet com a instalação de agentes SNMP padronizados, além de se utilizar um módulo **sensor** similar ao definido para a detecção de intrusão por uso incorreto.

A Tabela 5-4 mostra ainda quais ataques se pretende detectar utilizando-se um modelo de comportamento normal do sistema do tipo GMM e tendo como dados de referência o tráfego simulado, gerado conforme as premissas da seção anterior. Utiliza-se o mecanismo de detecção por comportamento proposto, com objetivo de discriminar a ocorrência do ataque em relação ao tráfego normal da rede.

Finalmente, pode-se utilizar uma monitoração colaborativa, onde um sensor é definido para escutar promiscuamente o meio de comunicação sem fio e sintetizar informações sobre o tráfego de vizinhos. Entretanto, deve-se destacar que essa abordagem não é muito efetiva no caso de DDoS, pois o nodo que monitora na vizinhança do nodo alvo tornar-se-ia igualmente indisponível. Para o caso de ataques de scanner, este método pode ser efetivo, permitindo que os vizinhos detectem ataques contra um nodo alvo.

Tabela 5-4 – Modelos de Comportamento e Variáveis Monitoradas

Modelo de Comportamento	**Variáveis a serem monitoradas**	**Ataques possivelmente detectados**
TCP	- número/taxa de conexões ou entrantes - duração de uma conexão - tcpInErrs* - tcpNoPorts*	- TFN e TFN2K - stacheldraht - shaft - mstream - scanner de TCP
UDP	- udpInDatagrams - udpInErrs* - udpNoPorts*	- trinoo - TFN e TFN2K - stacheldraht - shaft - scanner de UDP
ICMP	- icmpInEchos - icmpOutEchos - icmpInErrs*	- smurf - TFN (ping *flood*) - stacheldraht - shaft
IP	- ipReasmFails*	- TFN2K (Targa 3)

* estas variáveis são observadas com média e variância zero nos dados de construção do modelo de referência, pois trata-se de condições de erro não verificadas no tráfego gerado de forma simulada. Portanto, para este caso, o uso dessas variáveis gera singularidades na função de maximização do algoritmo EM, devendo ser evitado. Em redes reais, entretanto, essas variáveis apresentam valores não nulos que refletem as falhas ocasionais do sistema/rede monitorado.

5.4.4. Resposta a Intrusões

No caso de ataques de DDoS, a origem dos ataques (adversário) não pode ser claramente identificada nos pacotes de tráfego espúrio que são gerados, pois estes contém informações errôneas, na maioria do casos. A alternativa de defesa que se apresenta para esse tipo de ataque consiste em se evitar o encaminhamento do tráfego espúrio, o que exige uma colaboração de todas as entidades da rede que encaminham o tráfego desde seus pontos de origem (i.e. os nodos que tenham sido vítimas de comprometimento e possuam as ferramentas do ataque instaladas em seus sistema) até o alvo final (o nodo que sofre o ataque de DDoS). No caso de Manet, essa colaboração já existe, por princípio, devido à presença de um L-IDS em cada nodo. Pretende-se, como continuação deste trabalho, se investigar a possibilidade de se correlacionar alertas gerados por diversos nodos que estão no caminho dos pacotes de

tráfego espúrio, com objetivo de identificar-se o caminho destes pacotes, que poderiam ser filtrados de forma automática, enquanto o ataque durar.

Este método de resposta à intrusão, que não foi desenvolvido no contexto deste trabalho por limitações no tempo da pesquisa, deve ser ainda mais eficaz no caso de ataques de scanner de portas, pois, nestes casos, a origem dos pacotes é verdadeira e pode inclusive ser identificada. Isto possibilitaria um outro tipo de resposta – baseada na revogação do certificado da origem do ataque. A identificação da origem do ataque não aconteceria propriamente pela detecção de intrusão por comportamento proposta, mas sim pelo mecanismo de correlação de alertas que permitiria identificar o caminho dos pacotes espúrios desde sua origem ao seu destino (nodo alvo).

6. EXPERIMENTAÇÃO E RESULTADOS

Para validação do modelo de segurança proposto, os serviços de segurança são implementados e testados em uma Manet experimental, composta inicialmente de 10 nodos móveis. Um gerador de ataques é igualmente desenvolvido e permite executar o papel de adversários na rede experimental. Os mecanismos de segurança são aplicados na proteção dos protocolos de roteamento e autoconfiguração. Utiliza-se o protocolo de roteamento OLSR e o protocolo de autoconfiguração DCDP, com as otimizações propostas em [14]. Um modelo de simulação para validação dos serviços de segurança em ambientes de distribuição e topologia mais genéricas é também proposto. Este modelo é usado para avaliar o IDS por comportamento, descrito na seção 5.4.

6.1. PLATAFORMA EXPERIMENTAL

Para formação da Manet experimental utiliza-se 10 computadores, cada um com uma interface de rede IEEE802.11b (WiFi) (Anexo I) configurada para operação no modo *ad hoc*. Esses *hosts* possuem o sistema operacional com *kernel* GNU/Linux (versão 2.4.7), distribuição Red Hat (versão 9.0) instalado. Nos experimentos, não se utiliza o WEP (*wired equivalent privacy*) (Anexo I), mecanismo de proteção do nível de enlace, sem prejuízo para a validação dos mecanismos de segurança em consideração, uma vez que estes mecanismos operam em camadas superiores da rede. Não obstante, em redes reais, este mecanismo de segurança de nível de enlace de dados deve ser usado.

Utiliza-se a implementação do protocolo OLSR realizada pela Univerisdade de Oslo (Unik), Noruega [108], versão 0.4.7 (**uolsrd**). Esta implementação, linguagem de programação C, tem uma licença BSD[35] e pode ser usada livremente, inclusive para modificações. Esta versão pode ser compilada para ambientes GNU/Linux e Windows.

Não foram encontradas implementações para o protocolo de autoconfiguração DCDP disponíveis para *download* e utilização, uma vez que a definição de um protocolo de autoconfiguração para Manet ainda é processo que está sendo iniciado. Assim, uma implementação própria foi desenvolvida [14]. Esta implementação contém apenas a entrada e saída de nodos da rede. Os mecanismos para junção e partição não foram implementados ainda. A implementação do DCDP está integrada ao *daemon* de roteamento *uolsd* (linguagem

[35] Disponível em http://www.olsr.org/ (acessado em agosto/2004).

C), o que permite utilizar as otimizações com o uso de MPRs para as inundações (*floods*) requeridas pelo protocolo.

Para que as informações sobre os serviços de roteamento e autoconfiguração possam estar disponíveis ao sistema de detecção de intrusão, ou mesmo para outros serviços quaisquer, é desenvolvido ainda um agente SNMP (**OLSRAgent**) que implementa a MIB experimental para o OLSR, proposta neste trabalho. Este agente é desenvolvido com uso do pacote *Agent API*[36], construído em linhagem Java2. Este pacote é licenciado sob a Licença Pública GNU (versão 2), e pode ser usado livremente inclusive para modificações em seu código fonte. Para a comunicação entre o agente SNMP e o *daemon* de roteamento-autoconfiguração (trata-se do mesmo pacote) utiliza-se o protocolo SMUX, proposto pelo W3C como um protocolo experimental de multiplexação para gerenciamento de sessões. Este protocolo provê um canal de comunicação "leve" para a camada de comunicação no topo de uma conexão TCP. A especificação da MIB Experimental OLSR implementada encontra-se no Anexo VI. Como o agente SNMP é construído em Java2, este agente pode ser executado em qualquer plataforma que possua uma máquina virtual Java compatível com a API Java da Sun, versão 1.4.0 ou superior.

O agente SNMP do projeto NetSNMP[37] (**snmpd**) também é utilizado, para a coleta de informações da MIB-II. Este agente, construído em linguagem C, pode ser compilado para várias plataformas, inclusive Unix/Linux e Windows.

O L-IDS é (**LIDS**) implementado separadamente, em Java2, com uso da plataforma de agentes móveis *aglet* desenvolvida pela IBM [63].

É implementado ainda um pacote para geração de ataques contra os protocolos de roteamento e autoconfiguração (**attack**). Este software deve ser capaz de escutar promiscuamente o meio, para coletar informações sobre os nodos vizinho que possam ser usadas nos ataques e gerar mensagens na rede, sejam elas mensagens recebidas e encaminhadas com modificação (ataque de modificação) ou mensagens novas injetadas na rede (ataque de fabricação). Qualquer endereço IP ou MAC[38] pode ser usado como endereço de origem. As mensagens podem ser transmitidas em *unicast* ou *broadcast*, conforme o endereço IP de destino especificado.

---

[36] Disponível em http://nms.estig.ipb.pt/agentapi.web/index.jsp (acessado em agosto/2004).

[37] Disponível em http://netsnmp.org (acessado em agosto/2004).

[38] Algumas interfaces de rede não permitem que um quadro seja transmitido com um endereço MAC diferente de seu endereço *built in* (e.g. Lucent). Nesses casos, a personificação limita-se ao endereço IP.

Para a implementação do gerador de ataques utiliza-se a biblioteca de captura de pacotes *pcap*[39], desenvolvida como parte do projeto *tcpdump*. Para a geração de pacotes na rede utiliza-se a interface *socket* padrão do Unix/Linux, com a criação de um *raw socket* que permite a criação e envio de quadros de enlace de dados (MAC). Este módulo é construído em linguagem C e está disponível apenas para ambientes Unix/Linux.

6.2. TOPOLOGIA DA REDE E MOBILIDADE

A reprodução de padrões topológicos e de mobilidade com a Manet experimental da seção anterior é bastante limitada, além de ser difícil repetir-se uma realização de um experimento com as mesmas condições. De fato, a topologia de uma Manet pode variar de maneira muitas vezes imprevisível. A versatilidade da propagação de ondas eletromagnéticas na presença de obstáculos, da atenuação com a distância e da mobilidade é a fonte de dificuldades de modelização, tornando o problema muitas vezes intratável. Em especial, deve-se notar que a mobilidade não se refere apenas à mobilidade dos nodos da Manet, mas também à dinâmica do espaço de propagação. Assim, quando uma porta se abre em um ambiente *indoor*, ou quando um automóvel passa entre dois nodos, a distribuição de enlaces se altera.

Para avaliação dos aspectos de topologia e de mobilidade dos serviços propostos, adota-se modelos de simulação de distribuição de enlaces (propagação) para redes móveis, construídos a partir de simplificações dos cenários reais. Dois tipos de modelos simplificados para a definição do arranjo topológico da rede, considerando-se a mobilidade, são usados nos experimentos [12]. O primeiro deles, denominado modelo de grafo aleatório, consiste em um modelo aplicável a ambientes *indoor*. Neste modelo, admite-se que o enlace entre dois nodos quaisquer existe, em um determinado instante de tempo, com probabilidade p_1. A mobilidade é simulada através de um processo simples que define as transições do estado do enlace entre existente e não existente, e vice-versa. Se existente, o enlace pode passar a não mais existir, em um determinado instante de tempo, com probabilidade p_2. Igualmente, um enlace inexistente pode passar a existir, em um determinado instante de tempo, com probabilidade p_3 (usualmente, $p_2 \approx p_3$ faz com que o número de enlaces existentes não se aproxime de zero, quando $p_2 > p_3$, ou do número total de enlaces possíveis, quando $p_3 > p_2$). As probabilidades p_1, p_2 e p_3 são sempre as mesmas para todos os nodos, independentemente das posições em que eles se encontram. Esse modelo admite que a existência do enlace depende muito mais

[39] Disponível em http://tcpdump.org (acessado em agosto/2004).

das condições de propagação e dos obstáculos existentes entre dois nodos (e.g. paredes, móveis, etc.) do que da posição ocupada por ele no espaço.

O outro modelo, denominado modelo de grafo de unidade aleatória, é aplicável para ambientes *outdoor* ou ambientes *indoor* sem muitos obstáculos. Neste modelo, o enlace entre dois nodos existe apenas se a distância entre eles (d) for menor que um determinado valor, representando o alcance do enlace das interfaces sem fio. Diversas formas podem ser usadas para simular mobilidade nesse modelo, sendo a foram adotada nas simulações deste trabalho o modelo de mobilidade caminho de pontos aleatórios [12]. De acordo com este modelo, ao chegar em um determinado ponto de destino, um nodo permanece nesse ponto por um intervalo de tempo constante (t_{stop}) e em seguida um novo ponto de destino é escolhido aleatoriamente (uniformemente distribuído em todo domínio do espaço simulado). A velocidade de deslocamento é escolhida aleatoriamente, tendo uma distribuição uniforme entre um valor máximo e mínimo (V_{max} e V_{min}). Se a distribuição inicial dos nodos no espaço for inicialmente uniforme, ela manterá essa característica em regime estacionário. Neste caso, a densidade de nodos, em regime estacionário, pode ser facilmente calculada dividindo-se a quantidade total de nodos na simulação pela área (2D) ou volume (3D) da região do espaço que está sendo simulada.

O ambiente de simulação usado é o Network Simulator 2 (NS-2)[40]. Este ambiente foi compilado com as extensões para redes móveis *ad hoc* desenvolvidas no contexto do projeto Monarch CMU (Carnegie Mellon University – EUA)[41], e as extensões para simulação do OLSR desenvolvidas pelo IRINA – França[42].

São desenvolvidos ainda dois módulos para conversão dos arquivos de *trace* do NS-2 para o formato de analisadores de rede (ns2tcpdump) e para identificação de valores de variáveis importantes da MIB-II a partir dos pacotes que entram e saem de um nodo (tcpdump2mib). Com esses módulos é possível utilizar os dados simulados para verificar a operação dos serviços de segurança implementados. Esses módulos são construídos em linguagem C e utilizam partes de códigos das bibliotecas *libpcap* e *ucd-snmp* (NetSNMP).

[40] Disponível em http://www.isi.edu/nsnam/ns/ (acessado em agosto/2004).

[41] Disponível em http://www.monarch.cs.cmu.edu/cmu-ns.html (acessado em agosto/2004).

[42] Disponível em http://hipercom.inria.fr/OOLSR/ (acessado em agosto/2004).

6.3. EXPERIMENTAÇÃO DE VULNERABILIDADES DOS PROTOCOLOS OLSR E DCDP

O *daemon* uolsr, que implementa os protocolos de roteamento e autoconfiguração, pode ser utilizado com ou sem o serviço de autenticação (i.e. MAE). Para se executar o *daemon* de roteamento utiliza-se o comando[43]:

./uolsr [-i <interface> -cert [<certitifcado do nodo>] -share [<parte da chave privada>] -autoconf]

onde:

<interface>: identificador das interfaces de rede onde o *daemon* estará sendo executado (e.g. eth0), podendo ser uma lista de interfaces;
<certificado do nodo>: certificado digital usado para assinar as mensagens geradas;
<parte da chave privada>: parte da chave privada da ACD a ser usada pelo nodo.

A opção -cert controla o uso da MAE. Caso esta opção não esteja presente, os serviços de autoconfiguração e roteamento são executados sem que as mensagens geradas possuam uma MAE. Quando a opção –cert é especificada, todas as mensagens geradas são assinadas. Esta opção pode ou não receber parâmetros. Caso ela seja especificada sem parâmetros, o nodo deve obter um certificado pelo processo colaborativo (i.e. via L-Certs) antes de começar a gerar e receber mensagens dos protocolos de autoconfiguração e roteamento. Caso contrário, essa opção recebe como parâmetro um arquivo, em formato PEM, que contém, além do certificado do nodo/usuário propriamente dito, o certificado da autoridade certificadora que o assina (usado igualmente para verificar a MAE das mensagens recebidas) e a chave privada associada ao certificado, para assinatura das mensagens. Neste caso, apenas a renovação do certificado será realizada colaborativamente.

A opção –share controla a atuação do nodo nos serviços de certificação colaborativa. Caso esta opção não esteja presente, o nodo não poderá fazer parte de coalizões para prestação de serviços colaborativos de certificação, isto é, o nodo não possuirá uma parte da chave privada da ACD. Quando a opção –share é especificada, o nodo passa a executar, juntamente com o *uolsr*, uma instância de L-Cert. Esta opção também pode ou não receber parâmetros. Caso ela seja especificada sem parâmetros, o nodo deve obter sua parte da chave privada da ACD pelo processo colaborativo (i.e. via L-Certs) antes de começar a fazer parte

[43] Apenas as opções de comando sensíveis à parte que foi acrescentada ao programa são mostradas. O uolsrd tem várias outras opções de linha de comando que não são mostradas aqui.

de coalizões para prestação do serviço de certificação. Caso contrário, essa opção recebe como parâmetro um arquivo, em formato PEM, que contém sua parte da chave privada. Neste caso, apenas a atualização da parte da chave privada será realizada colaborativamente.

Finalmente, a opção –autoconf indica que o procedimento de autoconfiguração deve ser executado por este nodo. Caso esta opção não seja especificada, apenas o protocolo de roteamento é executado e o nodo não pode se configurar e nem servir requisições usando o protocolo de autoconfiguração. Em sendo especificada, a opção –autoconf indica que o serviço de autoconfiguração deverá ser executado nas interfaces listadas, iniciando a autoconfiguração de endereço IP destas interfaces logo após a completa iniciação do serviço *uolsr*. Nesta iniciação, está incluído o processo de obtenção de um certificado se este for requerido e não tiver sido especificado como atributo da opção –cert.

Os ataques contra o protocolo OLSR, definidos na seção 4.3.1.1, e os ataques de cliente e servidor contra o protocolo DCDP, definidos na seção , são implementados no gerador de ataque (*attack*) e podem ser executados a partir de qualquer um dos nodos da rede experimental. Para a execução dos ataques, utiliza-se o comando:

./attack [-i <interface> -a <tipo do ataque> -ip <endereço IP do alvo a ser personificado> -mac <endereço MAC do alvo a ser personificado> -cert <certificado digital do emissor de mensagens> -v -p]

onde:

- <interface>: identificador da interface de rede onde o ataque será gerado (e.g. eth0);
- <tipo do ataque>: *string* identificando o tipo do ataque, conforme mostrado na Tabela 5-1;
- <endereço IP do alvo a ser personificado>: endereço IP a ser personificado;
- <endereço MAC do alvo a ser personificado>: endereço MAC a ser personificado;
- <certificado digital do emissor de mensagens>: certificado digital usado para assinar as mensagens geradas/encaminhas, devendo ser o mesmo certificado usado pelo nodo que detém o endereço IP (e MAC) que será personificado, caso a opção "-IP e -MAC" tenham sido especificadas.

Tabela 6-1 – Ataques implementados no programa *attack*

string de identificação	**ataque**	**protocolo atacado**	**opções que devem especificadas**
nhop	Fabricação + Personificação HELLO	OLSR	-a, -IP [-MAC]
nhop+1	Fabricação HELLO	OLSR	-a
nhop+2	Fabricação TC	OLSR	-a
tcseqnum	Modificação + Personificação TC	OLSR	-a -IP [-MAC]
dcdpclient	Ataque cliente	DCDP	-a
dcdpserver	Ataque servidor	DCDP	-a

O programa funciona ainda como um analisador de protocolo para os protocolos OLSR e DCDP, mostrando as mensagens recebidas e enviadas pelo software em formato texto compreensível por seres humanos. A opção -v tem a finalidade de habilitar a exibição verborrágica das mensagens recebidas e enviadas pelo programa. A opção -p tem a função de evitar que a escuta de mensagens seja feita em modo promíscuo.

Com a opção -i deve-se especificar apenas uma interface de rede, isto é, para se gerar ataques simultaneamente em mais de uma interface, mais de uma instância do programa devem ser criadas.

As mensagens dos ataques podem ser encaminhadas/gerados com ou sem o uso de uma MAE. A opção -cert controla este processo. Caso esta opção não esteja presente, os ataques são gerados sem a MAE. Quando a opção –cert é especificada, todas as mensagens geradas são assinadas. Esta opção recebe como parâmetro um arquivo, em formato PEM que contém, além do certificado do emissor propriamente dito, o certificado da autoridade certificadora que o assina (usado igualmente para verificar a MAE das mensagens recebidas) e a chave privada associada ao certificado, para assinatura das mensagens. Nenhuma das opções de comando é obrigatória e se o comando for acionado sem especificar opções, este atuará como um mero analisador de rede, mostrando as mensagens recebidas e enviadas, em tempo real, na tela.

Um primeiro experimento realizado consiste em executar os protocolos OLSR e DCDP conforme estão especificados, isto é, sem a proteção de uma MAE. Neste caso o *uolsrd* deve ser invocado sem as opções –cert e –share. Todos os ataques mostrados na Tabela 5-1 foram executados neste cenário e com resultado observa-se o mal funcionamento dos protocolos provocado pelos ataques. Os efeitos são basicamente de DoS, mas outros efeitos podem ser observados conforme discutido anteriormente. Este experimento permite verificar, na prática, a existência das vulnerabilidades apontadas. No que diz respeito ao protocolo OLSR, a implementação utilizada (*uolsrd*) possui uma interface gráfica que permite verificar facilmente as disfunções de roteamento provocadas pelos ataques (e.g. quebra de simetria de enlaces, alterações nos conjuntos de MPR). No tocante ao DCDP, tanto no caso dos ataques de cliente quanto dos ataques de servidor, o efeito dos ataques observado quando nodos tentando se configurar na vizinhança do adversário não conseguem completar o processo.

6.4. MAE E L-CERT

Para implementação das funções de criptografia e manipulação de certificados digitais na implementação da MAE, usa-se a biblioteca *crypto* do projeto OpenSSL[44]. A MAE é implementada em uma biblioteca pré-compilada (**mae**) que possui funções para verificação e geração de assinaturas digitais (objetos DS) e manipulações em certificados digitais (e.g. extração de chaves públicas, verificação de assinatura, etc.). Esta biblioteca contém também as funções específicas para geração e verificação da MAE para os protocolos OLSR (mae-olsr) e DCDP (mae-dcdp). A extensão da MAE para outros serviços pode ser feita pela adição de novos módulos compilados separadamente e lincados junto com a biblioteca *mae*. Desse modo, a implementação da MAE é feita na forma de uma API. Esta API é usada para inserir a geração e verificação de mensagens recebidas nas implementações dos protocolos OLSR e DCDP, isto é, no programa *uolsrd*.

O L-Cert também utiliza a biblioteca *crypto* do projeto OpenSSL e, assim como o serviço de autoconfiguração, está integrado ao *daemon* de roteamento (*uolsrd*). Essa integração justifica-se pela intenção de fazer a distribuição de certificados em toda a rede de maneira integrada com o protocolo de roteamento, que se encarrega, por sua vez, de distribuir as rotas para todos os nodos. Além disso, as otimizações implementas nos protocolos de roteamento para realização de *flooding* (e.g. MPRs no caso do OLSR), também são aproveitadas pelo serviço de certificação.

A política de autenticação e certificação (Figura 4-6) a ser adotada deve ser especificada em um arquivo de configuração do *daemon uolsrd* denominado “policy.conf”. Este arquivo deve estar no mesmo diretório de onde o comando para iniciar o *olsrd* é chamado. Alternativamente, pode-se especificar a opção –policy <arquivo de configuração> informando como parâmetro o caminho e o nome do arquivo a ser utilizado.

Como identificadores dos nodos (e dos certificados) usa-se um resumo MD5 da assinatura digital do certificado, realizada por seu emissor. Este identificador é usado nos processos de formação de coalizões e como índice na *cache* de certificados válidos e na CRL. Este também é usado como identificador único dos nodos, do ponto de vista dos serviços de segurança.

Para a iniciação do serviço de certificação distribuída, é utilizado software negociador (**dealer**), desenvolvido em linguagem C. Este programa recebe como parâmetros o tamanho

[44] Disponível em http://openssl.org (acessado em agosto/2004).

da coalizão (*K*) e as requisições de certificados para os primeiros nodos (arquivos com extensão ".csr"). Após a geração da chave privada da ACD, o *dealer* assina todas as requisições de certificados e gera, para cada novo certificado assinado, uma parte da chave privada. Os resultados desta operação são gravados em arquivos no formato PEM, tendo como nome os mesmos nomes dos arquivos de requisição de certificado, mas com as extensões ".pem" e ".share.pem" para o novo certificado e sua parte da chave privada da ACD, respectivamente. O comando para invocação do *dealer* é mostrado a seguir.

./dealer <tamanho da coalizão> <arquivos .csr>

6.4.1. Parâmetros da Experimentação

Para a execução dos testes na Manet experimental, consideram-se os seguintes parâmetros (política):

- Tamanho da chave de certificação (ACD): Este parâmetro tem uma influencia importante no tamanho dos certificados, na computação requerida pelos serviços de certificação e na verificação da validade de objetos CERT (certificados) presentes em uma MAE, quando estes não estiverem presentes na cache de certificados válidos. Por outro lado, ele representa a segurança de certificação de todo o sistema. Nos experimentos, considera-se KIA_{CD} com 1024, 2048 ou 4096 bits. Não se considera seguro o uso de chaves RSA com 512 bits ou menos. Para utilização por períodos longos de tempo (e.g. 30 dias ou mais), não se recomenda o uso de chaves com 1024 bits.
- Tamanho da chave dos nodos: Este parâmetro tem influência importante no tamanho dos certificados, no tamanho dos objetos DS (assinatura digital) presentes na MAE e na computação requerida para se gerar as assinaturas digitais. Por outro lado, esse parâmetro representa a segurança das assinaturas digitais que autenticam as mensagens. Para redes que se foram e se desfazem em um curto período de tempo (e.g. 24 horas), o uso de chaves RSA de 512 bits pode ser tolerado. Nos demais casos, deve-se considerar o uso de chaves maiores. Nos experimentos, consideram-se chaves de 512 a 4096 bits.
- Tamanho da Coalizão (*K*): Este parâmetro representa o compromisso entre a disponibilidade dos serviços e a segurança do sistema. Ele tem influência fundamental no custo computacional (e de rede) dos serviços de certificação distribuídos. Como regra geral, este parâmetro deve ser comparável ao quantitativo médio de vizinhos de

um salto da rede. Considerando-se esta regra, adota-se $K = 3$ nos experimentos realizados.

- Informação nos certificados: Por simplicidade e padronização, utiliza-se certificados conformes com o padrão X.509v3. Com objetivo de diminuir o tamanho dos certificados, pois estes são carregados nas MAE de todas as mensagens caso se adote um esquema de distribuição de certificados pró-ativo, o mínimo de informações é colocado no certificado. Assim, os certificados utilizados contêm: nome (*distinguished name*) do proprietário do certificado, nome (*distinguished name*) do emissor (i.e. a AC), chave pública do proprietário, *timestamp* de emissão, *timestamp* de expiração e assinatura digital do emissor. Os certificados são transmitidos nas MAE e nas mensagens do protocolo L-Cert codificados em ASN.1, conforme determina o padrão X.509.
- Política de *cache* de certificados local: Como o protocolo OLSR é um protocolo de roteamento pró-ativo, adota-se uma política de distribuição pró-ativa dos certificados, com inclusão dos mesmos em todas as mensagens. Isto é, toda MAE tem, além do objeto mandatório (DS), um objeto CERT. Não se limita o tamanho da cache e os certificados válidos só são retirados da cache em caso de expiração ou revogação do certificado.
- Política de CRL: Utiliza-se uma estratégia de sincronização de CRL reativa, onde os nodos que chegam na rede devem solicitar explicitamente pela versão atual da CRL a seus vizinhos. No caso da emissão de um novo certificado, o nodo com o maior ID da coalizão envia automaticamente a atualização, quando da emissão do certificado parcial em favor do novo nodo certificado.
- Política para emissão de certificados: Os certificados são emitidos segundo uma política de verificação manual da identidade do emissor. Assim, ao receber uma requisição de emissão de um novo certificado, o usuário é perguntado se deseja ou não servir esta requisição. Este usuário deve, então, se certificar da identidade do solicitante, através de procedimento definido pela política de segurança. Os certificados são sempre emitidos com validade de 3600 segundos (01 hora).
- Política para renovação de certificados: Adota-se uma renovação automática de certificados, onde uma requisição de renovação de certificado cujo certificado anterior (ainda válido) não tenha sido revogado ou contra os qual não se tenha detectado, localmente, ações errôneas de seu proprietário (i.e. via IDS), são assinadas automaticamente. Nos demais casos, trata-se a requisição como se fosse uma emissão

de um novo certificado. Os certificados são sempre renovados com validade de 3600 segundos (01 hora).

- Política para emissão de partes da chave privada da ACD: Adota-se uma política de serviço manual, similar à política para emissão de certificados.
- Política para renovação de partes da chave privada da ACD: Adota-se um período de 24 horas para renovação pró-ativa de partes da chave privada.

No que diz respeito aos aspectos de topologia da rede e mobilidade, assume-se um modelo de simulação do tipo grafo por unidade aleatória, em uma Manet formada por um determinado número de nodos (10, 50, 100 e 1000), que se encontram uniformemente distribuídos em uma região bi-dimensional de 250m x 250m. A velocidade máxima de deslocamento é definida em 5m/s, para considerar movimentos devido a deslocamentos de seres humanos caminhando ou de veículos automotores se deslocando em baixa velocidade.

6.4.2. Avaliação do Overhead de Comunicação

O overhead provocado na rede pela adição da MAE às mensagens do protocolo de roteamento OLSR pode ser avaliado em termos comparativos com o overhead de controle das mensagens OLSR sem qualquer autenticação. Para que se possa comparar essas duas situações é necessário identificar a quantidade de bytes da MAE para diversas configurações da política de certificação, já que a MAE tem um tamanho fixo para cada mensagem, assim como a quantidade de bytes contida nas mensagens OLSR para Manet em diferentes tamanhos. A razão entre esses dois valores fornece o overhead adicional, em relação ao overhead de controle que existe de todo modo em uma rede com roteamento OLSR.

A Tabela 6-2 mostra os tamanhos, em bytes, para os elementos que compõem uma MAE que possui um objeto DS (assinatura digital) e um objeto CERT (certificado do emissor da mensagem). Esta tabela apresenta ainda, em sua penúltima coluna, o tamanho de uma MAE contendo apenas um objeto DS. Esta coluna tem por objetivo evidenciar que a maior contribuição para o tamanho da MAE é devida claramente ao objeto CERT. Este, por sua vez, tem seu tamanho influenciado tanto pelo tamanho da chave de certificação, pois ele carrega a assinatura da ACD, quanto pelo tamanho da chave de certificado, haja vista que ele carrega uma chave pública.

Tabela 6-2 – Tamanho dos Elementos da MAE (OLSR e DCDP)

Chave Certificado (bits)	Chave CA (bits)	Cabeçalho MAE (bytes)	Cabeçalho Obj. DS (bytes)	Objeto DS (bytes)	Cabeçalho Obj.CERT (bytes)	Objeto CERT (bytes)	MAE (s/CERT) (bytes)	MAE (c/CERT) (bytes)
512	**4096**	4	4	64	4	724	72	800
1024	**4096**	4	4	128	4	788	136	928
2048	**4096**	4	4	256	4	916	264	1184
512	**2048**	4	4	64	4	468	72	544
1024	**2048**	4	4	128	4	532	136	672
2048	**2048**	4	4	256	4	660	264	928
512	**1024**	4	4	64	4	340	72	416
1024	**1024**	4	4	128	4	404	136	544
2048	**1024**	4	4	256	4	532	264	800

A avaliação do overhead de controle do protocolo OLSR foi realizada com base nos resultados da simulação. A Tabela 6-3 indica a quantidade média de endereços IP anunciados em mensagens HELLO e TC para diferentes densidades de nodos. No caso de mensagens HELLO este valor corresponde ao número médio de vizinhos que os nodos possuem no momento da geração da mensagem. No caso de mensagens TC, este valor refere-se ao número de MS.

Tabela 6-3 – Número médio de endereços anunciados em mensagens HELLO (vizinhos) e TC (MS)

Número de Nodos na Rede	X (m)	Y (m)	Alcance do enlace sem fio (m)	HELLO	TC
10	250	250	50	1,26	4,81
50	250	250	50	6,28	8,23
100	250	250	50	12,57	10,37
1000	250	250	50	125,7	22,35

O tamanho da mensagem varia de acordo com o tamanho do endereço IP utilizado. Para IPv4, este tamanho é igual a 04 bytes e para IPv6 tem-se 16 bytes por endereço. Assim, a Tabela 6-4 mostra o tamanho médio (teórico) das mensagens HELLO em cada uma das densidades de nodos no espaço considerado.

Tabela 6-4 – Tamanho médio, em bytes, de mensagens HELLO (sem MAE)

Número de nodos na rede	**10**	**50**	**100**	**1000**
Tamanho médio da vizinhança	1,26	6,28	12,57	125,66
Tamanho médio da mensagem (IPv4)	33,03	53,13	78,27	530,65
Tamanho médio da mensagem (IPv6)	48,11	128,53	229,06	2038,62

O *overhead* provocado na rede pela inserção de uma MAE em uma mensagem OLSR corresponde ao tamanho da MAE, mostrado na última coluna da Tabela 6-2. Este overhead representa um aumento do tamanho da mensagem que varia entre 400 e 1200 bytes. O Tamanho médio das mensagens OLSR, sem MAE, no entanto, varia entre 30 e 500 bytes, no

caso de IPv4, e de 58 a 2058 bytes, no caso de IPv6 (Tabela 6-4). Isso representa um aumento de até 40 vezes no tamanho da mensagem. Entretanto, deve-se considerar que antes de se transmitir uma mensagem em uma rede sem fio em modo *ad hoc* é necessário a arbitração do meio compartilhado. Esse processo não tem eficiência total, de modo que a banda disponível em um enlace não corresponde à banda nominal. Por exemplo, no caso da tecnologia IEEE 802.11b, usada nestes experimentos, usa-se o protocolo CSMA/CA (vide descrição deste protocolo no Anexo I), cuja eficiência pode ficar abaixo de 50%, para enlaces em uma vizinhança com mais de cinco nodos executando o protocolo no modo *ad hoc*. Assim, o overhead representado pelo aumento do tamanho de uma mensagem, que não acarrete na quebra dessa mensagem em mais de um pacote não representa grandes perdas na ocupação do meio de transmissão. A quebra de uma mensagem (HELLO[45] ou TC) deve ocorrer sempre que essa mensagem ultrapassar o MTU (*maximum transfer unit*) da rede. Como exemplo, considera-se novamente o caso de IEEE 802.11b, modo *ad hoc*, onde um quadro de enlace de dados pode carregar até 2.304 bytes. Em IPv4, descontando-se os cabeçalhos IP (20 bytes), UDP (8 bytes) e do pacote OLSR (4 bytes) o MTU desta rede está é de 2272 bytes. Já no caso de IPv6, o cabeçalho do IP tem um tamanho que depende das opções presentes, mas, em situações usuais, na ultrapassa 48 bytes. Assim, o MTU da rede em IPv6 está em torno de 2260 bytes. Em cada mensagem HELLO, deve-se adicionar uma MAE, com objetos CERT (no caso da distribuição pró-ativa de certificados). As Tabela 6-5 e Tabela 6-6 mostram os tamanhos médio das mensagens HELLO para endereçamento IPv4 e IPv6, respectivamente, contendo uma MAE com um objeto CERT por mensagem, para cada uma das densidades de nodos consideradas, em função dos tamanhos das chaves da ACD e do certificado.

Tabela 6-5 – Tamanho médio (bytes) de mensagens HELLO – IPv4 (MAE com objeto CERT)

Chave Certificado (bits)	**Chave CA (bits)**	**Número de nodos na rede**			
		10	**50**	**100**	**1000**
512	**4096**	833	853	878	1331
1024	**4096**	961	981	1006	1459
2048	**4096**	1217	1237	1262	1715
512	**2048**	577	597	622	1075
1024	**2048**	705	725	750	1203
2048	**2048**	961	981	1006	1459
512	**1024**	449	469	494	947
1024	**1024**	577	597	622	1075
2048	**1024**	833	853	878	1331

[45] Mensagens TC tem tamanhos menores que mensagens HELLO, pois anunciam apenas parte da vizinhança de um nodo, isto é, aqueles vizinhos que escolheram este nodo como MPR.

Tabela 6-6 – Tamanho médio (bytes) de mensagens HELLO – IPv6 (MAE com objeto CERT)

Chave Certificado (bits)	Chave CA (bits)	Número de nodos na rede			
		10	**50**	**100**	**1000**
512	**4096**	848	929	1029	2839
1024	**4096**	976	1057	1157	2967
2048	**4096**	1232	1313	1413	3223
512	**2048**	592	673	773	2583
1024	**2048**	720	801	901	2711
2048	**2048**	976	1057	1157	2967
512	**1024**	464	545	645	2455
1024	**1024**	592	673	773	2583
2048	**1024**	848	929	1029	2839

No caso de endereçamento IPv4 (Tabela 6-5), o tamanho das mensagens HELLO adicionadas de uma MAE ocorre, em média, apenas no caso de uma alta densidade de nodos (1000 nodos em uma área de 250m x 250m) e para uma chave de ADC de 4096 bits e uma chave de certificado de 2048 bits. Assim, o aumento do tamanho de mensagens é perfeitamente tolerável, nesse caso. No caso de IPv6 (Tabela 6-6), esta mesma análise permanece válida mesmo em redes com densidades de nodos moderada (100 nodos em uma área de 250m x 250m). Já no caso de redes com alta densidade de nodos, a mensagem HELLO, sem MAE já ultrapassa o limite do MTU da rede, devendo ser necessariamente quebrada. Nestes casos, duas soluções são possíveis: acrescentar uma MAE para cada parte da mensagem ou utilizar uma única MAE para toda a mensagem. Na primeira opção, pode-se começar a se processar cada parte da mensagem na medida em estas são recebidas. Entretanto, o número de quadros necessários para enviar todas as partes de mensagens HELLO chega a 04, para a maioria dos casos considerados, o que pode ser proibitivo. Na segunda opção o problema consiste em se perder uma das partes da mensagem e se ter que descartar toda a mensagem, pois não seria possível verificar a assinatura da mensagem incompleta.

Em qualquer caso, pode-se considerar ainda otimizações no mecanismo de distribuição dos certificados, uma vez que os certificados tem um período de validade longo, em relação ao intervalo de HELLO. Assim, a *cache* local de certificados pode ser usada para evitar que se distribua um certificado em cada mensagem enviada. Uma otimização possível consiste em se carregar objetos CERT na MAE somente quando se detectar a presença de novos nodos na vizinhança (e.g. um nodo cujo certificado não esteja na *cache* de certificados locais). Entretanto, esta alternativa faz com que haja uma falta de certificado para a primeira

mensagem HELLO proveniente de um nodo que não tenha o seu certificado na *cache* de seus vizinhos.

As Tabela 6-7 e Tabela 6-8 mostram os tamanhos médios das mensagens para IPv4 e IPv6, respectivamente, contendo uma MAE sem o objeto CERT. Analisando-se os dados dessa tabela verifica-se que a inserção desta MAE não acarreta em geração de quadros adicionais na rede, qualquer que seja a configuração de rede considerada.

Tabela 6-7 – Tamanho médio (bytes) de mensagens HELLO – IPv4 (MAE sem objeto CERT)

Chave Certificado (bits)	**Chave CA (bits)**	**Número de nodos na rede**			
		10	**50**	**100**	**1000**
512	**4096**	105	125	150	603
1024	**4096**	169	189	214	667
2048	**4096**	297	317	342	795
512	**2048**	105	125	150	603
1024	**2048**	169	189	214	667
2048	**2048**	297	317	342	795
512	**1024**	105	125	150	603
1024	**1024**	169	189	214	667
2048	**1024**	297	317	342	795

Tabela 6-8 – Tamanho médio (bytes) de mensagens HELLO – IPv6 (MAE sem objeto CERT)

Chave Certificado (bits)	**Chave CA (bits)**	**Número de nodos na rede**			
		10	**50**	**100**	**1000**
512	**4096**	120	201	301	2111
1024	**4096**	184	265	365	2175
2048	**4096**	312	393	493	2303
512	**2048**	120	201	301	2111
1024	**2048**	184	265	365	2175
2048	**2048**	312	393	493	2303
512	**1024**	120	201	301	2111
1024	**1024**	184	265	365	2175
2048	**1024**	312	393	493	2303

No que diz respeito ao protocolo de autoconfiguração, vale ressaltar que este serviço gera tráfego de controle apenas em determinados momentos, quando ocorre a chegada de um novo nodo na rede. Além disso, as mensagens do protocolo considerado (DCDP) não ultrapassam 70 bytes de tamanho (sem MAE). Desse modo, a inclusão da MAE em mensagens deste protocolo é prontamente tolerável, no tocante ao overhead da rede.

No que diz respeito ao overhead de comunicação gerado pelos serviços de certificação distribuídos, a Tabela 6-9 mostra o número de comunicações *broadcast* e *unicast* necessárias para cada um dos serviços básicos de certificação. Uma análise da Tabela 6-9 indica que o overhead de comunicação para a obtenção de um serviço de certificação aumenta linearmente com *K*. Entretanto, deve-se ressaltar que, assim como o protocolo de autoconfiguração, estes

serviços geram tráfego quando um novo nodo chega à rede e requisita um certificado e uma parte da chave privada ou através dos processos periódicos de renovação de certificados e de atualização das partes da chave privada da ACD. Vale ressaltar que a periodicidade de realização desses processos é consideravelmente mais longa que a periodicidade de geração de mensagens de controle do protocolo de roteamento. Desse modo, o overhead de comunicação gerado pelo protocolo de certificação distribuída é muito menor que o overhead de controle gerado pelo OLSR, mesmo no caso de não se usar qualquer tipo de autenticação. Além disso, com uma escolha conveniente de *K*, este overhead está localizado na vizinhança do nodo que requisita os serviços de autenticação, o que confere escalabilidade aos serviços projetados.

Tabela 6-9 – Overhead de Comunicação L-Cert

Serviço de Certificação	Número de *broadcasts* ou *floodings*	Número de *unicasts*
emissão ou renovação de certificado	2	2 * *K*
emissão de partes da chave privada	3	3 * *K*
revogação de certificados*	1	2 * *K*

* não inclui a correlação de alertas, pois este é overhead do L-IDS.

6.4.3. Avaliação do Overhead Computacional

A primeira métrica para avaliação do overhead computacional provocado pela MAE é o tempo dispensado na construção e verificação da MAE de todas as mensagens geradas e recebidas, respectivamente, durante um intervalo de HELLO (i.e. o tempo médio entre o envio de duas mensagens HELLO). Obviamente, plataformas computacionais com desempenho diferentes apresentam diferentes valores absolutos para essa métrica. Além disso, o número médio de mensagens TC e HELLO recebidas durante esse intervalo de tempo varia de acordo com a dinâmica da topologia da rede. De fato, o número de mensagens HELLO recebidas para processamento aumenta na medida em que aumente o tamanho da vizinhança (e.g. chegada de um novo vizinho). Por esses motivos, faz-se uma estimativa indireta do tempo de computação necessário para processamento das MAE geradas e recebidas. Como a geração de uma assinatura RSA e a verificação de uma assinatura RSA são, indiscutivelmente, as operação mais computacionalmente dispendiosas nos processos de geração de uma nova mensagem e de verificação da MAE de mensagens recebidas, estima-se o número de operações desse tipo realizadas na emissão e recepção de mensagens, para cenários típicos de Manet.

A Tabela 6-10 mostra os resultados da simulação realizada, indicando a quantidade média de mensagens HELLO e TC enviadas por um nodo, durante um intervalo de HELLO, considerando-se um intervalo de TC igual a três vezes o intervalo de HELLO. Do mesmo modo, essa tabela mostra o número médio de mensagens HELLO e TC recebidas neste mesmo intervalo, lembrando que as mensagens HELLO são geradas apenas na vizinhança de um salto e as mensagens TC são disseminadas por *flooding* na Manet.

Tabela 6-10 – Número médio de mensagens enviadas e recebidas (HELLO e TC)

Número de nodos na rede	**10**	**50**	**100**	**1000**
Número médio de mensagens enviadas (HELLO e TC)	1,16	1,05	1,03	1,01
Número médio de mensagens enviadas (HELLO e TC)	6,07	14,52	22,94	148,01

Considerando que cada MAE deve possuir um objeto DS (assinatura digital) e um objeto CERT (certificado), considera-se como overhead de computação as seguintes operações:

- 01 geração de assinatura digital RSA com a chave pública de certificado, para cada mensagem enviada;
- 01 verificação de assinatura digital RSA com a chave pública de certificado, para cada mensagem recebida.

A etapa de verificação da validade do certificado do signatário da mensagem, que envolve 01 verificação de assinatura digital RSA com a chave pública de ACD pode ser desconsiderada, pois admitindo-se a existência de uma *cache* de certificados válidos, todos os nodos aprendem os certificados uns dos outros em regime estacionário.

As Tabela 6-11 e Tabela 6-12 apresentam o tempo médio de overhead para a geração e verificação de assinaturas RSA, respectivamente, para diferentes tamanhos de chave de certificado. Esses valores foram avaliados em três plataformas com desempenho computacional diferentes: Plataforma 1 - *laptop* com processador Pentium IV de 1,7GHz de freqüência de núcleo e 256Mbytes de RAM; Plataforma 2 – *laptop* com processador Pentium III de 850MHz e 256Mbytes de RAM; Plataforma 3 – PalmTop Compaq iPAQ com processador Intel StrongARM de 206MHz e 64Mbytes de RAM.

Tabela 6-11 – Tempo médio (ms) de geração da assinatura RSA

Chave Certificado (bits)	**Plataforma 1**	**Plataforma 2**	**Plataforma 3**
512	0,91	2,31	32,8
1024	4,36	9,2	150
2048	26,2	68,1	850

Tabela 6-12 – Tempo médio (ms) de verificação da assinatura RSA

Chave Certificado (bits)	Plataforma 1	Plataforma 2	Plataforma 3
512	0,165	0,293	3,51
1024	0,328	0,701	11,5
2048	0,928	2,01	29,90

Para se calcular o overhead (tempo de processamento) total, para cada plataforma, multiplica-se o tempo gasto para a geração de uma assinatura pelo número médio de mensagens que são geradas e soma-se este resultado ao produto do tempo gasto para verificar uma assinatura pelo tempo médio de verificação do certificado. Esses resultados podem ser normalizados pelo intervalo de HELLO. As Tabela 6-13, Tabela 6-14 e Tabela 6-15 mostram o overhead total relativo a um intervalo HELLO de 2s.

Tabela 6-13 – Overhead total de processamento em relação ao intervalo de HELLO mensagens enviadas + recebidas (HELLO e TC) – Plataforma 1

Número de nodos na rede		**10**	**50**	**100**	**1000**
Chave Certificado (bits)	**512**	0,103%	0,168%	0,236%	1,267%
	1024	0,352%	0,467%	0,601%	2,648%
	2048	1,801%	2,049%	2,414%	8,191%

Tabela 6-14 – Overhead total de processamento em relação ao intervalo de HELLO mensagens enviadas + recebidas (HELLO e TC) – Plataforma 2

Número de nodos na rede		**10**	**50**	**100**	**1000**
Chave Certificado (bits)	**512**	0,223%	0,334%	0,455%	2,285%
	1024	0,746%	0,992%	1,278%	5,652%
	2048	4,560%	5,035%	5,813%	18,314%

Tabela 6-15 – Overhead total de processamento em relação ao intervalo de HELLO mensagens enviadas + recebidas (HELLO e TC) – Plataforma 3

Número de nodos na rede		**10**	**50**	**100**	**1000**
Chave Certificado (bits)	**512**	2,968%	4,270%	5,715%	27,632%
	1024	12,190%	16,224%	20,916%	92,681%
	2048	58,375%	66,332%	78,070%	> 100%

Uma análise dos valores das tabelas permite verificar o alto dispêndio de computação provocado pelo uso da criptografia assimétrica. Mesmo assim, no caso da plataforma 1 (Pentium IV 1,8GHz, Tabela 6-13), o overhead total de computação não ultrapassa 2,6% para quase todas as situações avaliadas, exceto no caso extremo de uma alta densidade de nodos e com o uso de uma chave de certificado de 2048 bits, quando o overhead computacional chega a pouco mais de 8%. Entretanto, em plataformas computacionais de menor capacidade, como é o caso da plataforma 3 (Tabela 6-15), a utilização dos mecanismos propostos só mostra-se viável para tamanhos de chave de certificado de 512 bits e em redes com densidade moderada de nodos. Assim, como sugere a Tabela 6-13, com o aumento rápido da capacidade

computacional dos processadores projetados para plataformas computacionais móveis, mesmo o uso de criptografia assimétrica pode ser tolerado. Porém, como o custo computacional ainda é bastante elevado, deve-se procurar alternativas para o esquema de criptografia RSA (e.g. criptografia de curva elíptica) ou se adotar um esquema de criptografia simétrica, perdendo-se a possibilidade de se fazer uma autenticação com não-repúdio – o que limita as possibilidades de resposta à intrusão do modelo.

O overhead computacional para cálculo e verificação de MAE no caso do protocolo de autoconfiguração é mínimo, comparado ao overhead provocado pelo protocolo de roteamento, uma vez que o DCDP gera muito menos tráfego de controle que o OLSR.

Para avaliação do custo computacional dos serviços básicos de certificação distribuídos, avalia-se a complexidade desses processos no nodo requisitante e nos nodos que participam da coalizão, conforme mostrado na Tabela 6-16. Deve ser notado que a operação "básica" é diferente em cada caso. Cada uma dessas operações requer um tempo diferente para sua computação, conforme é ilustrado na Tabela 6-17.

Tabela 6-16 – Operações básicas e complexidade computacional do L-Cert

Serviço	Operação básica	Complexidade	
		Nodo Requisitante	Nodo da Coalizão*
Verificação de Certificado	verificação de assinatura RSA (chave pública da ACD)	$O(1)$	-
Emissão, renovação e revogação de certificado	verificação de assinatura RSA (chave pública da ACD)	$O(K)$	
	geração de assinatura RSA (parte da chave privada da ACD)**		$O(1)$
	multiplicação na interpolação de Lagrange (Eq. 4-5)	$O(K)$	
Emissão de partes da chave privada e atualização de partes da chave privada	geração de assinatura RSA (parte da chave privada da ACD)**	-	$O(1)$
	verificação de assinatura RSA (chave pública de certificado)	-	$O(K\text{-}1)$***
	geração de assinatura RSA (chave privada de certificado)	-	$O(K\text{-}1)$***
	soma na computação da parte da chave privada (Eq. 4-10)	$O(K)$	-

*Custo computacional, por cada nodo. Para se avaliar o custo computacional total em todos os nodos, este valor deve ser multiplicado por *K*. Entretanto, como o custo computacional é medido em tempo de processamento, está se considerando apenas uma vez esse tempo de processamento nos nodos da coalizão, pois essa computação é realiza em paralelo nos *K* nodos.

**O custo computacional de se realizar uma assinatura RSA com uma parte da chave privada ou uma cifração com a chave RSA pública é maior que o custo computacional de se realizar uma assinatura RSA padrão, pois, neste último caso, pode-se utilizar otimizações no algoritmo de computação devido ao conhecimento da fatoração do módulo em seus fatores primos.

***Apenas um dos nodos da coalização realiza *K*-1 operações. Cada um dos demais nodos da coalizão realiza um número menor de operações.

Tabela 6-17 – Tempo médio (ms) de computação das operações básicas do L-Cert				
Operação básica	Tamanho da chave	Plataforma 1	Plataforma 2	Plataforma 3
verificação de assinatura RSA (chave pública da ACD)	1024	0,328	0,701	11,5
	2048	0,928	2,01	29,90
	4096	2,91	7,00	92,2
geração de assinatura RSA (parte da chave privada da ACD)	1024	4,36	9,2	150
	2048	26,2	68,1	850
	4096	81,5	197	2670
verificação de assinatura RSA (chave pública de certificado)	512	0,165	0,293	3,51
	1024	0,328	0,701	11,5
	2048	0,928	2,01	29,90
geração de assinatura RSA (chave privada de certificado)	512	0,91	2,31	32,8
	1024	4,36	9,2	150
	2048	26,2	68,1	850
multiplicação na interpolação de Lagrange	1024	1,65	3,26	67,1
	2048	2,02	5,01	91,1
	4096	6,5	14,7	109,9
soma na computação da parte da chave privada	1024	0,000023	0,000061	0,00189
	2048	0,000022	0,000063	0,00208
	4096	0,000029	0,000081	0,00305

6.4.4. Avaliação de Desempenho do L-Cert

O tempo total de computação para cada um dos serviços básicos do L-Cert são mostrados na Tabela 6-18. Esses dados são coletados na Manet experimental (chave de ACD com 4096 bits e chave de certificado com 1024 bits) e referem-se a um valore de K = 3, executando os algoritmos em nodos com *hardware* do tipo da plataforma 1 e 2. Esses tempos são medidos entre o envio da mensagem que confirma a formação da coalizão (passo 3, Figura 3-2) e a conclusão da computação do novo certificado ou parte da chave privada. Não se considera o tempo para formação da coalizão, pois, durante essa fase, aplica-se a política de certificação, o que pode levar um tempo indeterminado. Como os serviços de certificação são realizados apenas quando ocorrem novas adesões à rede ou quando os nodos precisam renovar seus certificados, o que ocorre em períodos muito superiores aos tempos mostrados na Tabela 6-18, considera-se que esses tempos são perfeitamente aceitáveis.

Tabela 6-18 – Desempenho do L-Cert			
Serviço	Tempo médio para obtenção do serviço (s)	Número de mensagens broadcast	Número de mensagens unicast
Emissão, renovação e revogação de certificado	0,91	2	6
Emissão de partes da chave privada e atualização de partes da chave privada	1,23	3	9

Considerando os tempos utilizados na computação das operações de cada um desses serviços, pode-se verificar que o tempo do processo completo é consideravelmente superior à soma dos tempos necessários para todas as comutações necessárias. Assim, concluí-se que uma parte significativa do tempo necessário para obtenção do serviço é devida às comunicações.

Um último aspecto a ser analisado é o efeito da mobilidade na disponibilidade dos serviços de certificação distribuídos. De uma maneira geral, os nodos escolhidos para fazerem parte da coalizão devem se manter na vizinhança do nodo requisitante, pelo menos até o recebimento da mensagem (broadcast) que confirma a formação da coalizão. Caso um desses nodos se mova antes de receber esta mensagem, ele não saberá que foi escolhido como membro da coalizão e não responderá ao solicitante, fazendo com que toda computação realizada pelos outros nodos servidores seja perdida. Para evitar isso, o nodo solicitante pode aguardar por respostas dos membros da coalizão por um intervalo de tempo pré-determinado ($T_{timeout\text{-}c}$) após o qual ele pode decidir enviar uma mensagem *unicast* para cada um dos nodos que fazem parte da coalizão, mas não responderam à mensagem informando sua formação. Isso permite que esses nodos sirvam a requisição, mesmo se estiverem a mais de um salto de distância do requisitante. Entretanto, caso um dos nodos da coalizão falhe ou torne-se indisponível, o processo deve ser recomeçado com a formação de uma nova coalizão.

6.4.5. Avaliação da CRL Local e da *Cache* de Certificados Válidos

Conforme discutido nas seções precedentes, o uso da *cache* de certificados é essencial para o desempenho dos serviços de segurança preventivos (MAE e L-Cert). No que diz respeito à *cache* de certificados válidos é interessante verificar a quantidade de memória necessária para armazenar os certificados coletados pelo processo de distribuição adotado. A Tabela 6-19 mostra a quantidade de memória necessária quando os certificados de todos os nodos da Manet (simulação) estiverem carregados na *cache*.

Tabela 6-19 – Tamanho da *Cache* (kbytes)

Chave Certificado (bits)	**Chave CA (bits)**	**Número de nodos na rede**			
		10	**50**	**100**	**1000**
512	**4096**	7,1	35,4	70,7	707,0
1024	**4096**	7,7	38,7	77,3	773,4
2048	**4096**	9,0	45,1	90,2	902,3
512	**2048**	7,1	35,4	70,7	707,0
1024	**2048**	7,7	38,7	77,3	773,4
2048	**2048**	9,0	45,1	90,2	902,3
512	**1024**	7,1	35,4	70,7	707,0
1024	**1024**	7,7	38,7	77,3	773,4
2048	**1024**	9,0	45,1	90,2	902,3

No que diz respeito à CRL local, esta tem tamanho reduzido, considerando-se que ela poderá conter em um determinado instante, no máximo, *K*-1 contra-certificados. Caso contrário, o sistema foi quebrado pela existência de, pelo menos, *K* nodos comprometidos na rede.

6.5. L-IDS: DETECÇÃO POR USO INCORRETO

Na implementação desenvolvida, a detecção de intrusão é baseada em informações coletadas da MIB, mantida localmente em cada nodo da Manet por agentes SNMP (**OLSRAgent** e **ucd-snmp**). O uso desse tipo de fonte de informação é duplamente justificado. Primeiramente, existe um conjunto extenso de MIBs padronizadas, cobrindo uma grande variedade de aplicações e serviços de rede, a exemplo da MIB-II e da MIB RMON. Essas MIBs padronizadas estão implementadas em agentes SNMP desenvolvidos para diferentes plataformas, o que confere uma boa portabilidade ao L-IDS. Em segundo lugar, a informação da MIB é composta por informações que descrevem entidades de diferentes níveis da arquitetura do sistema. Assim, é possível monitorar, simultaneamente, os níveis de rede, sistema e mesmo algumas aplicações.

O módulo *coletor de dados* consiste de um software capaz de realizar consultas SNMP (e.g. SNMP GET, SNMP GETNEXT, etc.) e de receber *traps* (SNMP TRAP). Os dados brutos resultantes correspondem aos valores das variáveis MIB no momento da realização da consulta ou do envio da *trap*.

As regras de abstração de dados são realizadas em classes especializadas da classe eventAbstractorRule (regra de abstração de evento), devendo implementar a interface ruleProcessing (processamento de regra), contendo uma função processRule() que recebe como parâmetro os dados brutos coletados/recebidos pelo módulo *coletor de dados*. Assim, as regras de abstração podem ser escritas com toda a flexibilidade oferecida pela linguagem Java2, sem quebrar a estrutura modular do software. Essas classes especializadas podem ainda ser compiladas em tempo de execução, sob demanda. Para a detecção dos ataques descritos na 5.3.1, são implementadas as seguintes regras de abstração, mostradas na Tabela 6-20:

O *núcleo de IDS* para implementação da detecção de intrusão por uso incorreto consiste de uma máquina de estados finitos capaz de executar e manter múltiplas instâncias de DEF, uma para cada ataque que está sendo monitorado.

Tabela 6-20 – Regras de Abstração

Regra de abstração	Dados brutos	Consulta	Eventos Gerados	Descrição
trapOLSRNeighborLinkStatusChanged	olsrNeighborEntry	-	NHOP_E1, NHOP_E2	Implementa o processamento de mensagens que indicam a mudança de estado do enlace com um vizinho.
trapOLSRNeighborMPRSetChanged	olsrNeighborEntry, olsr2hopNeighborTable	-	N+1HOP_E1	Implementa o processamento de mensagens que indicam a mudança no conjunto de MPRs.
queryOLSRNeighborTable	olsrNeighborTable	N+1HOP_C1	N+1HOP_E2, N+1HOP_E3, N+1HOP_E4	Verifica se um nodo está anunciando o nodo local como seu vizinho se o ser, de fato.
queryTCPConnTable	tcpConnTable	STEPSTONE_PC1 STEPSTONE_C1	STEPSTONE_E1 STEPSTONE_E2 STEPSTONE_E2	Verifica a existência de conexões telnet, em cadeia, em um determinado nodo.

O *gerente de alertas* implementa uma correlação simples, verificando se múltiplos alertas recebidos foram gerados por nodos diferentes. Este processo armazena também todas as acusações (alertas) assinadas localmente ou recebidas de outros nodos. Já o módulo *comunicador* implementa o protocolo de comunicação (SMUX) para envio de informações ao L-Cert (uolsrd). Através desse módulo são enviados os alertas de grupo com os endereços de todos os nodos que detectaram um ataque proveniente de um mesmo adversário. Esse módulo serve também para que o L-Cert consulte sobre a existência de acusações geradas no L-IDS contra algum nodo para o qual se está solicitando a assinatura de um contra-certificado.

O módulo *plataforma de agentes* possui um servidor *aglet* que pode gerar e receber agentes móveis, destruí-los ou despachá-los para outros nodos. A criação de agentes móveis é sempre solicitada pelo módulo *gerente de distribuição*, sempre que uma consulta, evento, estado de detecção ou alerta deva ser executada em um nodo remoto. Do mesmo modo, quando a *plataforma de agentes* recebe um agente proveniente de outro nodo, extrai-se a mensagem a ser executada localmente, que é passada ao *gerente de distribuição* para processamento local. A Figura 5-1 ilustra o L-IDS implementado.

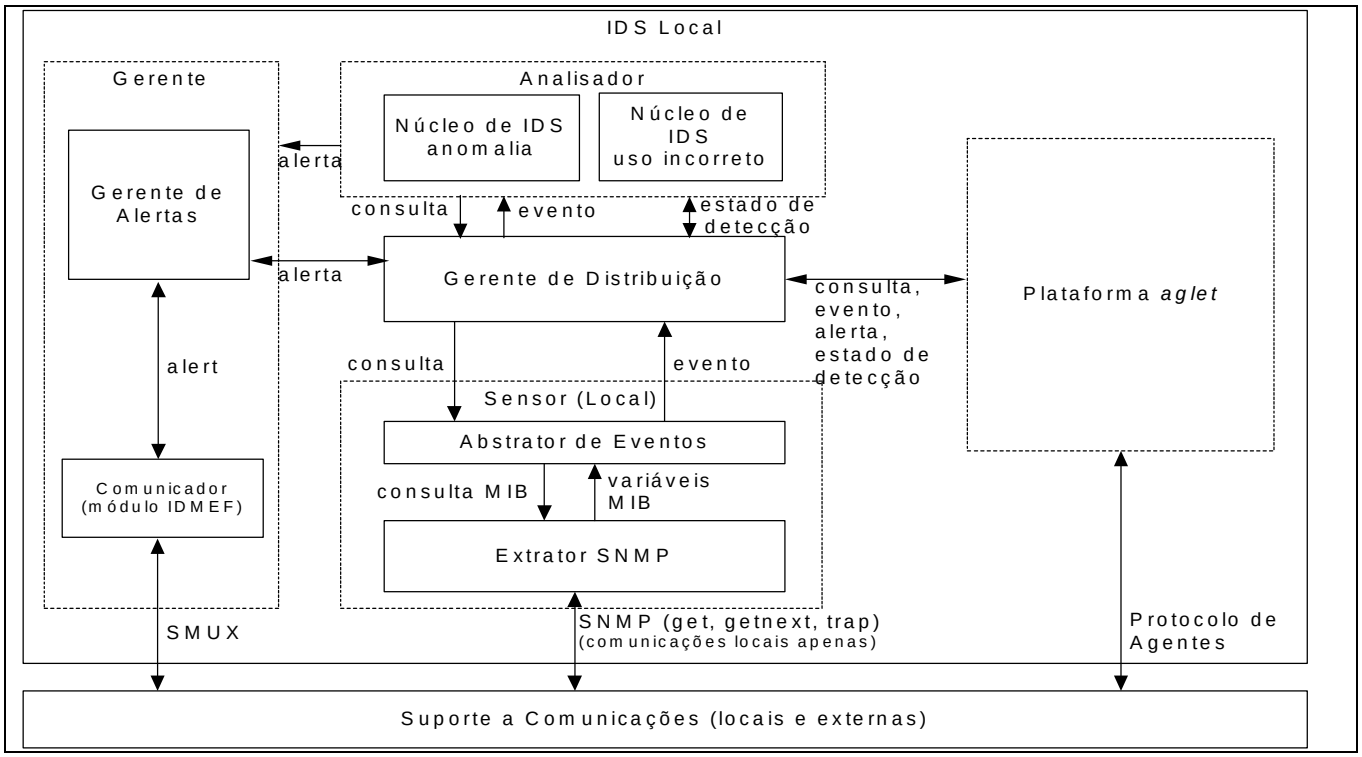


Figura 6-1 – Arquitetura Modular do L-IDS Implementado

6.5.1. Considerações de Desempenho

Comparativamente ao serviço de autenticação do protocolo de roteamento, o L-IDS consome muito menos memória e tempo de processador. Enquanto o uso de CPU do processo uolsrd (*daemon* OLSR, *daemon* DCDP e L-Cert) oscila entre 0,1% e 2% em um computador com hardware do tipo plataforma 1 (Pentium 4 1,8GHz, 256M RAM), o processo que executa o L-IDS consome sempre menos que 0,1% do tempo de CPU. Para o quesito memória, enquanto a memória média alocada para o processo *uolrd* é de 5,8Mbytes, a memória consumida pelo processo do L-IDS é sempre inferior a 2Mbytes. O overhead na rede também é pequeno, haja vista que o processo de detecção de intrusão ocorre sempre na vizinhança dos nodos e só é disparada uma investigação colaborativa (despacho de agentes móveis para nodos remotos) quando algum evento que possa estar associado a um ataque é detectado localmente. Assim, o L-IDS parece ter boa escalabilidade, pois as comunicações geradas pelo L-IDS não dependem da interação com nodos distantes a mais de dois saltos na Manet.

O maior overhead provocado na rede pelo L-IDS consiste na inundação de mensagens de alerta, quando uma intrusão é detectada. Os L-IDS precisam compartilhar esse tipo de informação para gerar a resposta colaborativa à intrusão. Entretanto, essas mensagens ocorrem apenas quando se detecta a presença de um adversário e apenas até que o certificado deste adversário seja revogado.

6.6. AVALIAÇÃO DA SEGURANÇA

Para validação dos serviços de segurança implementados, utiliza-se uma Manet experimental com 10 nodos, descrita na seção 6.1. Dois desses nodos são usados para o papel de nodos bem comportados, i.e. executam corretamente o OLSR e o DCDP e os serviços L-Cert e L-IDS. Os dois nodos restantes fazem o papel de adversários, gerando os ataques descritos anteriormente.

O processo de avaliação é divido em três etapas. Primeiramente, a rede é formada com os protocolos de roteamento e autoconfiguração sendo executados juntamente com o serviço de detecção de intrusão apenas. O objetivo consiste em avaliar a eficácia de detecção do IDS construído. Os nodos adversários executam os ataques definidos contra esses protocolos e os efeitos dos ataques são observados de maneira semelhante ao que se observa quando não há qualquer mecanismo de proteção (seção 6.3). Os efeitos dos ataques ocorrem mesmo que eles sejam detectados, pois o mecanismo de proteção corretiva (interação entre L-IDS e L-Cert) não está ativado. Para ilustrar o processo detecção, a Figura 6-2 mostra uma topologia particular assumida pela Manet durante a experimentação. Nesta figura, os nodos bem comportados são A, B, C, D, E, F, G e H, enquanto os nodos adversários são X e Y.

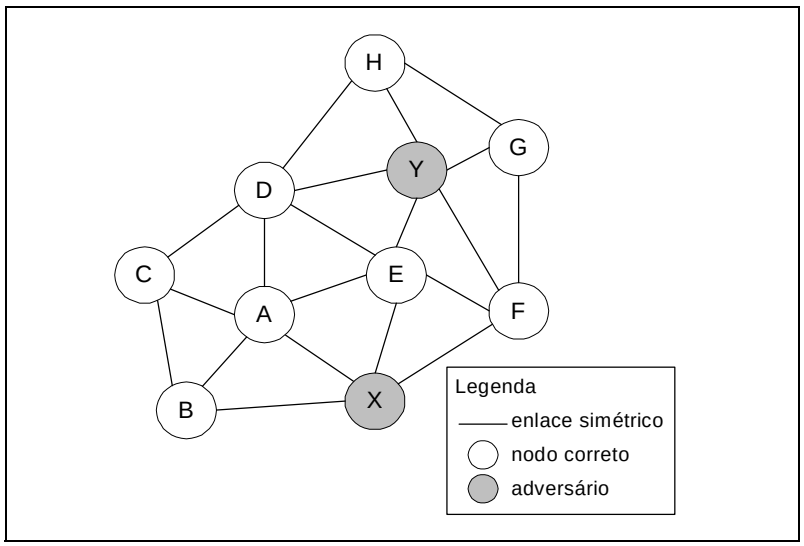


Figura 6-2 – Exemplo de Topologia da Manet Experimental

Cada um dos quatro ataques definidos na seção 4.3.1.1 são executados simultaneamente pelos nodos adversários X e Y. Como resultado, observa-se que, pelo menos 03 dos L-IDSes executando em nodos corretos são capazes de detectar os ataques gerados por cada um dos adversários, sem nenhum falso negativo. A Tabela 6-21 mostra os efeitos dos ataques gerados por X, observados pelo nodo correto A.

Tabela 6-21 – Detecção de Ataques contra o Protocolo OLSR

Ataque	**Mensagem fabricada/modificada**	**Conjunto MPR antes do ataque**	**Conjunto MPR durante o ataque**	**Nodos que detectam o ataque**
N+1HOP	HELLO, com B,C,D,E,F,H,Z = "sym"	D,E	X	C,D,G
NHOP	HELLO, com B,C,D,E = "lost", personificando E	D,E	D	A,B,F
TC_MS	TC, com F,G,H	D,E	X	A,D,F,G,H
TC_SEQNUM	TC de E, com número de seqüência modificado	D,E	D,X	A,B,C,D,E,F,G,H

Como a topologia experimental não sofre variações abruptas, permanecendo quase estática durante todo o experimento, não se observa também nenhum falso positivo para as assinaturas de ataque definidas. Esse resultado indica que o uso de um sistema de detecção de intrusão por uso incorreto, com assinaturas de ataque precisamente especificadas (baixa taxa de falsos positivos) é uma alternativa coerente para iniciação de respostas automáticas à intrusão, pela revogação de certificados, conforme propõe o modelo desenvolvido nesse trabalho. Entretanto, ainda que esse primeiro resultado seja bem sucedido, ele não é ainda conclusivo, pois a avaliação do comportamento na presença de mobilidade não foi ainda devidamente caracterizada.

Outro aspecto importante consiste em observar que a especificação precisa de assinaturas de ataques é uma tarefa complexa e que só pode ser realizada para ataques que são previamente conhecidos. Entretanto, as assinaturas podem ser generalizadas para identificar ocorrências de estados proibidos na execução dos protocolos que estão sendo monitorados, como é o caso das assinaturas dos ataques NHOP e N+1HOP, onde anomalias no escalonamento do protocolo e na topologia local da rede disparam os alertas. Essas assinaturas podem ser completadas para modelar um conjunto mais completo de ataques, mesmo aqueles que não estejam ainda explicitamente definidos, que passam a ser identificados pelo aparecimento de condições de execução proibidas. Essa abordagem, comumente denominada de detecção de intrusão baseada em especificação [64,111] é um outro campo para maiores investigações em trabalhos futuros.

A segunda etapa no processo de validação consiste em executar o OLSR e o DCDP com a proteção preventiva (MAE), juntamente com o L-Cert. Assim, nesta etapa, apenas o L-IDS não está iniciado. Os oito nodos bem comportados têm os seus certificados e partes da chave privada distribuídos *off-line* pelo negociador. Como mostrado na Figura 6-2, qualquer nodo correto na topologia experimental tem uma vizinhança de, no mínimo três vizinhos

igualmente corretos. Assim, configura-se o tamanho das coalizões para K = 3. Obviamente, os ataques gerados a partir dos nodos adversários sem o uso da autenticação pela MAE não fazem mais efeito, pois essas mensagens são descartadas no processamento realizado pelos *daemons* OLSR e DCDP. Entretanto, como não se tem ainda o serviço de IDS, ataques gerados por nodos certificados (comprometidos) ainda podem ser efetivos em gerar distúrbios na rede. Para simular a existência de nodos comprometidos, permite-se que os dois nodos adversários recebam um certificado usando o processo de certificação distribuída (L-Cert). Desse modo, pode-se verificar o correto funcionamento dos serviços de certificação e de autoconfiguração com certificação: os nodos adversários solicitam primeiramente um certificado, alocando um endereço IP temporário e, em seguida, solicitam e recebem um bloco de endereços IPs pelo uso correto do DCDP. De posse dos certificados e de um endereço IP, estes nodos passam a gerar ataques, que são uma vez mais bem sucedidos em provocar distúrbio nos serviços de roteamento e autoconfiguração.

Na terceira e última etapa do processo de validação, todos os serviços de segurança implementados são ativados. Assim, ao se gerar ataques contra o protocolo OLSR, os L-IDS passam a detectar a presença dos nodos comprometidos e revogam seus certificados (K = 3). Com a revogação dos certificados dos adversários, os efeitos dos ataques são completamente mitigados. Nos experimentos realizados, o tempo máximo observado para revogação do certificado de ambos os adversários ocorreu no caso dos ataques TC_MS (fabricação de mensagem TC), onde o processo levou cerca de 12 segundos para ser completado (incluindo a revogação do certificado).

A Tabela 6-22 mostra o tempo médio de geração do alerta de grupo para cada um dos ataques considerados. Esse tempo representa o tempo que o adversário tem para “escapar” a monitoração dos nodos vizinhos. Entretanto, as acusações contra um nodo ficam armazenadas nos módulos *gerente de alertas* até que os certificados dos nodos comprometidos sejam revogados, passando depois para um “log” de detecção (em arquivo). Assim, se um nodo conseguir escapar da monitoração de seus vizinhos, mas voltar a agir incorretamente, seu certificado pode ainda ser revogado com base na informação armazenada. Além disso, os nodos que tenham gerado acusações contra os adversários se negarão a renovar os certificados para estes nodos.

Tabela 6-22 – Tempo médio para Início do Processo de Resposta à Intrusão

Ataque	**Tempo médio para geração do alerta de grupo (s)***
N+1HOP	2,2
NHOP	3,1
TC_MS	3,5
TC_SEQNUM	1,4

*Não inclui a revogação do certificado, apenas a geração da mensagem de formação de coalizção.

Outro ponto importante diz respeito à escolha do parâmetro *K*. Claramente, existe um compromisso entre a segurança e o desempenho/disponibilidade dos serviços nesta escolha. Se *K* é escolhido com um valor menor do que o tamanho da vizinhança, todos os serviços podem ser providos de maneira localizada. Além disso, para que o processo de resposta à intrusão seja efetivo, é preciso que haja, pelo menos *K* nodos detectando ataques gerados por um determinado adversário. Para valores de *K* maiores que o tamanho da vizinhança, é possível que o número de vizinhos que conseguem detectar as atividades disturbadoras do nodo comprometido seja insuficiente para provocar a revogação do certificado do adversário, permitindo que este continue a gerar ataques localmente sem ter o seu certificados anulado. Nestes casos, os nodos que detectam os ataques devem ter alguma forma de "pedir ajuda" a outros nodos corretos para que estes possam se aproximar da vizinhança e detectar o ataque para completar o número suficiente de nodos na coalizão que revoga o certificado.

É possível ainda que existam nodos comprometidos em uma vizinhança que não colaboram no processo de detecção e resposta à intrusão. Assim, mesmo que *K* seja menor que o tamanho da vizinhança, pode-se chegar a situações onde os nodos de uma vizinhança que estão cooperando no processo de detecção de intrusão não sejam em número suficiente. Uma heurística bastante simples pode ser usada para se definir o valor de *K*. Seja *ns* tamanho médio da vizinhança. O número máximo de nodos comprometidos que pode-se ter em uma rede é igual a $K - 1$. Assim, fazendo-se $K = ns - (K - 1)$, tem-se $K = (ns + 1)/2$. Nota-se que este é justamente o caso para a topologia da Figura 6-2, pois considerando-se os nodos X e Y o menor número de nodos em uma vizinhança (incluindo o próprio nodo) é 5. Portanto, K = 3 satisfaz à heurística proposta, neste caso.

6.7. L-IDS: DETECÇÃO POR MODELAGEM DE COMPORTAMENTO

Diversas implementações de algoritmo EM estão disponíveis a partir de bibliotecas científicas *on-line* tais como a *statlib*[46] e outras fontes especializadas de recursos *on-line*.

[46] http://lib.stat.cmu.edu.

Entre elas, encontram-se implementações largamente utilizadas, tais como o programa *autoclass* [22] (originalmente em Fortran, recentemente portada para C), EEMIX [76] (Fortran) e MCLUST [39] (ambiente S-Plus). Não se encontrou nenhuma implementação em Java para o algoritmo EM para v.a. multivariadas. Como o L-IDS está todo escrito em Java (e.g. sensor, gerente de alertas, etc.), torna-se necessário fazer a implementação do núcleo de IDS por modelagem de comportamento também em Java. Em [2], existe uma implementação do algoritmo EM para o caso bi-dimensional generalizado aplicado a distribuições Gaussianas. Apesar de estar pobremente documentada, esta implementação foi completamente reescrita para o caso n-dimensional.

As Figura 6-3 e Figura 6-4 ilustram uma simulação do algoritmo de modelagem de comportamento e detecção de intrusão, para uma nova amostra considerada normal e para outra considerada anômala, respectivamente. Os dados de referência foram gerados a partir de 03 distribuições gaussianas, independentes e bem separadas, e misturados em iguais proporções. O algoritmo EM, juntamente com o algoritmo de estimação automática da ordem do GMM paramétrico foram bem sucedidos em identificar os parâmetros das distribuições e a ordem ótima do modelo, *a posteori*, conforme pode ser verificado pelas figuras. Os valores calculados para a entropia dos GMM paramétricos ajustados com o algoritmo EM de primeira, segunda, terceira e quarta ordens são igualmente mostrados. Esses gráficos ilustram a discriminação clara da ordem do modelo.

Para a detecção de intrusão, considera-se normal um evento para o qual o valor de λ (Eq. 5-15) é maior ou igual a 0,012, pois este é o menor valor de λ dentre todos as realizações observadas durante a fase de treinamento. Assim, para o caso da Figura 6-3, o novo evento (ponto vermelho) que corresponde a $\lambda = 0{,}3279$ é considerado normal. Por outro lado, o evento da Figura 6-4, para o qual $\lambda = 0{,}000157$, é considerado um sinal de intrusão.

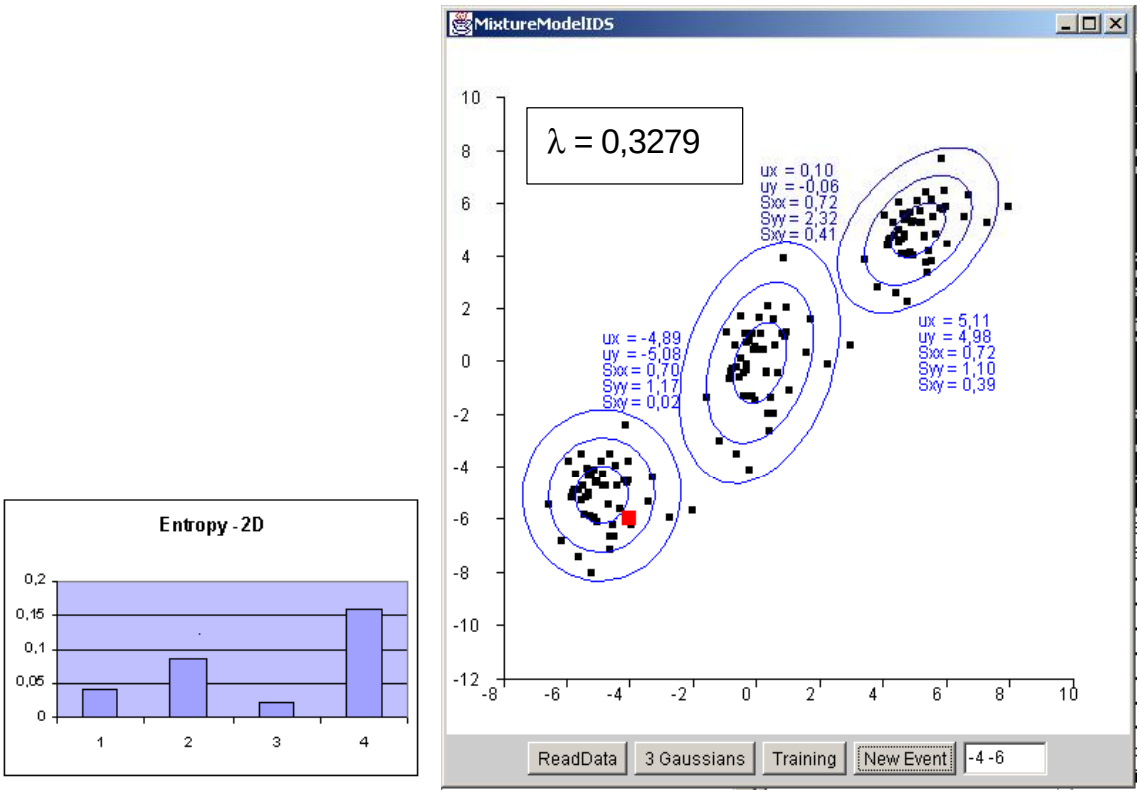


Figura 6-3 – Modelo de comportamento com 03 *clusters* e reconhecimento de um novo dado refletindo um comportamento normal.

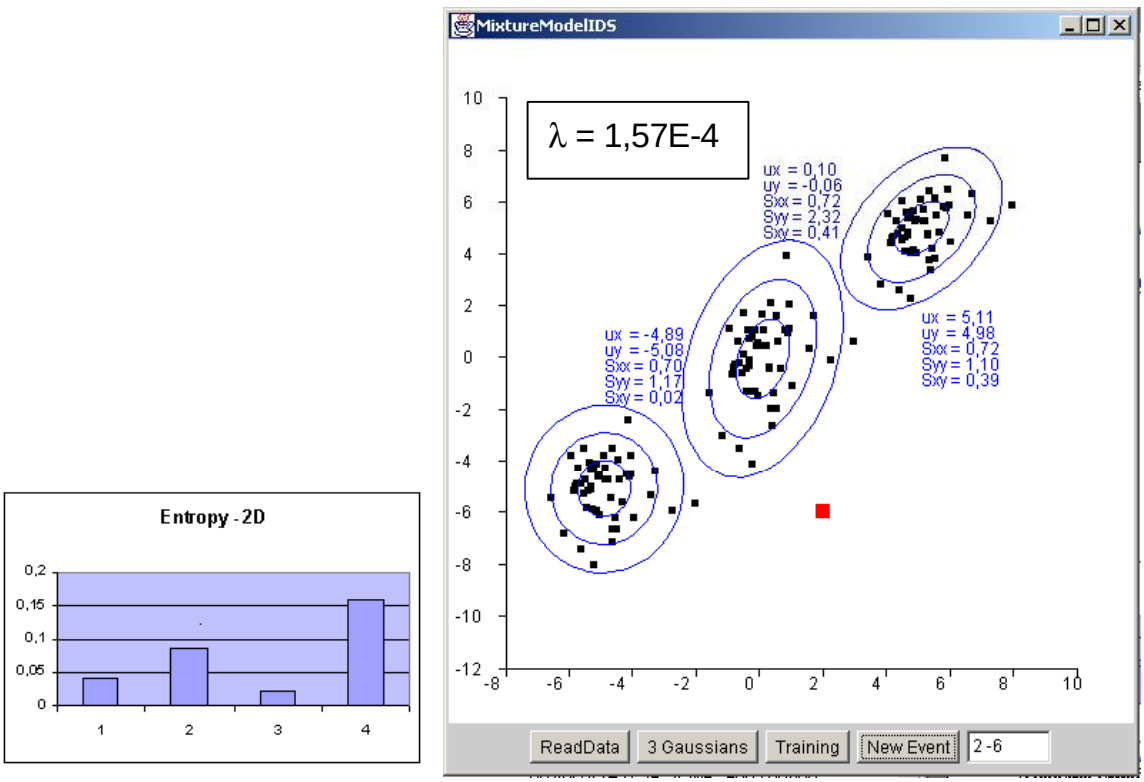


Figura 6-4 – Modelo de comportamento com 03 clusters e reconhecimento de um novo dado refletindo um comportamento anômalo.

Os dados apresentados nas figuras anteriores são úteis apenas do ponto de vista demonstrativo, pois foram gerados artificialmente a partir de distribuições que têm exatamente a mesma distribuição que compõe as funções nucleares do modelo de mistura em uso. Assim, é bastante natural que se observe um perfeito casamento entre os dados de referência e o modelo a eles ajustado. Na prática, no entanto, esse casamento não vai sempre existir.

A Figura 6-5 ilustra o modelo geração e processamento de dados de simulação para verificação da aplicabilidade das técnicas de detecção de intrusão por comportamento, apresentadas na seção 5.4, à detecção de ataques contra redes Manet. Primeiramente, o *script trafficgen* permite que sejam ajustados os modelos da simulação (e.g. Manet com 50 nodos, área de 250m x 250m, alcance de transmissão de 50m, etc.). O ns-2 é usado como ferramenta de simulação e gera um arquivo de *trace* contendo todos os pacotes gerados, encaminhado e recebidos em todos os nodos da rede (*trafego.out*). No entanto, as variáveis MIB devem ser mantidas e monitoradas em cada nodo. Desse modo, esse arquivo é decomposto em vários outros arquivos, um por nodo da rede, através do programa *ns2tcpdump*. Em cada arquivo são colocados apenas os pacotes gerados, recebidos ou encaminhados por este nodo. Assim, este arquivo corresponde a um *dump* de pacotes capturados por um analisador de rede, com a interface de captura em modo não promíscuo. Este arquivo transforma ainda os pacotes abstratos gerados pelo ns-2 em pacotes que se parecem com aqueles capturados por um analisador de redes: todos os campos dos protocolos de camada 3 e 4 são preenchidos (inclusive endereços IPv4 de 4 bytes) e atribui-se um *timestamp* absoluto, compatível com a medida de tempo relativa usada pelo *ns*-2, a cada pacote. O resultado do programa *ns2tcpdump* são arquivos *.pcap, em formato compatível com o formato de *dump* de pacotes brutos da biblioteca *libpcap*. Como esse formato é largamente suportado por diversos analisadores de rede, como exemplo, o ethereal[47], os arquivos *.pcap podem ser visualizados e analisados por essas ferramentas. Em seguida, cada um desses arquivos é processado pelo programa *tcpdump2mib* que produz como saída (arquivos *.mib) uma lista de amostras para os valores das variáveis MIB, amostradas em um intervalo de tempo que pode ser definido por passagem de parâmetros na chamada do comando. Finalmente, um módulo do L-IDS *extrator MIB* modificado permite injetar essas informações no L-IDS. Esse coletor de dados, desenhado para processamento de dados *off-line*, executa *consultas periódicas* que retornam, para os instantes de tempo quando a consulta é executada, os valores assumidos pelas

[47] Disponível em http://www.ethereal.com.

variáveis MIB que encontram-se armazenados no arquivo *.mib. Deve-se notar que o período de amostragem passado ao programa *tcpdump2mib* (i.e. para geração do arquivo *.mib) não precisa ser o mesmo período das *consultas periódicas* usadas pelo módulo *extrator MIB* do L-IDS. Na prática, o período das consultas é muito maior que o período usado pelo *tcpdump2mib*.

Dois modelos de tráfego são analisados mais de perto: TCP e UDP. O uso desses modelos em separado faz com que exista uma discriminação implícita entre todo tráfego UDP e TCP gerados. Assim, o modelo de comportamento usando UDP servirá para modelar apenas a aplicação de videoconferência e o protocolo de roteamento. Já no caso do TCP, modela-se o tráfego gerado pelas aplicações Telnet e FTP.

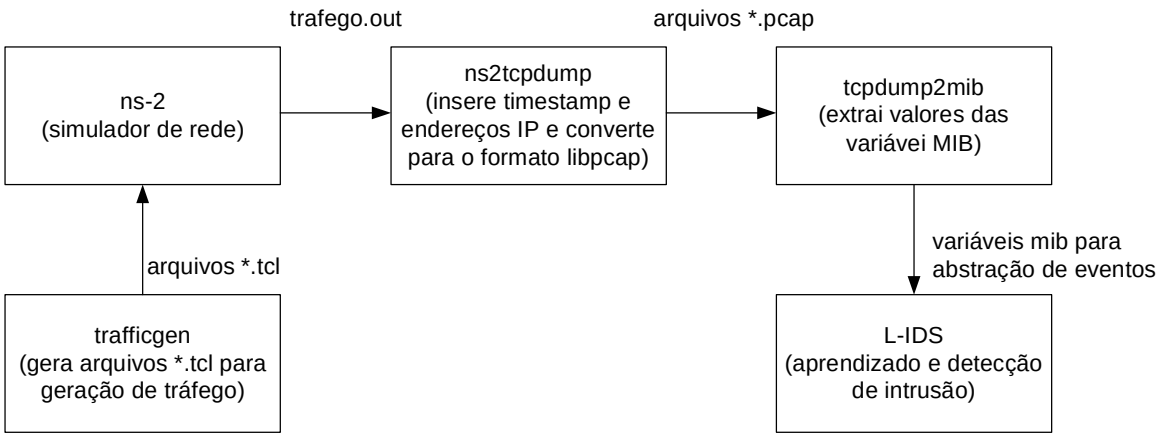


Figura 6-5 – Processo de geração da simulação

No caso do modelo UDP, apenas as variáveis udpInDatagrams (datagramas UDP que entram em um nodo são usados) e ipForwDatagrams (datagramas IP encaminhados pelo nodo). Como essas variáveis são monotonicamente crescentes, defini-se como resultado da abstração de eventos a geração de um evento de aprendizado (realização), cujo valor (udpIn ; ipForw) é obtido subtraindo-se do valor atual de (udpInDatagrams ; ipForwDatagrams) (consulta periódica corrente) o seu valor precedente (consulta periódica antecedente). Ajustou-se o período das consultas para o mesmo intervalo de TC (igual a três vezes o intervalo de HELLO) do OLSR. Para fazer o treinamento e o ajuste do modelo, todos os eventos de treinamento gerados em todos os nodos são consumidos em um mesmo L-IDS, procedendo-se ao ajuste do GMM aos dados de referência (eventos). O resultado dessa etapa são os parâmetros do modelo de misturas, que são colocados em uma mensagem *estado de detecção* e distribuídos para todos os L-IDS da rede.

Nesse instante (recebimento da mensagem *estado de detecção*) o processo de detecção de intrusão pode começar.

No que diz respeito ao ajuste do GMM aos dados resultantes da simulação para o modelo UDP, observa-se a formação de dois *clusters* bem definidos: o primeiro, com média em (52,3 ; 93,9) datagramas e desvio padrão de (10,2 ; 39,7) datagramas. Certamente, esse cluster indica as condições de tráfego de um nodo que não está recebendo nenhum pacote da aplicação de videoconferência, podendo estar ou não encaminhado datagramas de videoconferências (alto desvio padrão na variável ipForwDatagrams). Um outro cluster, com média em (203 ; 101) datagramas e com desvio padrão de (21,1 ; 47,1) datagramas resulta da modelagem do tráfego de videoconferência (fonte CBR a 128kbps). Obviamente, existe uma contribuição do tráfego do protocolo OLSR no valor da média e do desvio desse *cluster*. A correlação entre as variáveis é positiva, mas pequena (36,7 datagramas).

Para a geração do ataque de DDoS, simula-se a geração de um tráfego UDP CBR (2Mbps) em quatro nodos de origem escolhidos aleatoriamente, em direção a um único nodo de destino. Aplicando-se o modelo detecção, são detectadas situações anômalas em todos os nodos que encaminham o tráfego desde a origem até o destino. Esse resultado, bastante interessante do ponto de vista de detecção de DDoS só é possível graças a análise conjunta de duas variáveis udpInDatagrams e ipForwDatagrams.

Outra análise em relação a este ataque de DDoS faz-se ainda necessária: obviamente o nodo que recebe todo o tráfego gerado (de todos os seus vizinhos) vai tornar-se rapidamente indisponível (o próprio ns-2 acusa a geração de vários erros de encaminhamento e descarte de pacotes na vizinhança do nodo de destino). Entretanto, os nodos que estão distantes, apesar de estarem gerando/encaminhando uma quantidade expressiva de dados, não estão necessariamente quebrados com o ataque. Como o sistema de detecção de intrusão identifica anomalias em todos os nodos do caminho, sugere-se que, caso haja uma interação entre esses nodos intermediários, pode-se bloquear o encaminhamento dos pacotes vindos dessa origem. Esse encaminhamento deve ser bloqueado com base nos endereços de enlace e não nos endereços de destino do datagrama IP, pois esses são facilmente falsificados e, nos ataques de DDoS mais avançados, são constantemente alterados (a cada pacote).

No caso do modelo TCP, utiliza-se como variáveis MIB tcpPassiveOpens (número de conexões abertas passivamente no nodo) e tcpInSegs (número de segmentos, inclusive com erro e para abertura de conexão, recebidos). Do mesmo modo que no caso do UDP, define-se uma consulta com período igual a 5*(intervalo de TC), onde a abstração de eventos calcula

(tcpPO ; tcpIN) como a diferença entre o valor de (tcpPassiveOpens ; tcpInSegs) na consulta atual e precedente. Para se evitar singularidades (i.e. formação de um cluster com média zero e variância pequena para tcpPassiveOpens), os eventos onde tcpPassiveOpens foi igual a zero são descartados como normais tanto no processo de aprendizagem quanto no processo de detecção de intrusão. No que diz respeito ao ajuste, observa-se, neste caso, a formação de dois clusters com médias em (1,11 ; 38,41) e em (1,05 ; 97,11), modelando respectivamente o telnet e o FTP.

Os eventos gerados em todos os nodos são consumidos em um único L-IDS e a mensagem *estado de detecção* contendo os parâmetros ajustados do modelo é distribuída. Para a geração de um ataque de scanner, escolhe-se um par origem-destino aleatoriamente e faz com que esta origem envie pedidos de conexão TCP ao destino, em uma taxa de 10 pedidos por segundo. No destino, faz-se um dreno que, a cada 30 pedidos de conexões uma é aceita (i.e. indicando um “match” com uma porta que esteja respondendo). Logo que os valores das variáveis MIB começam a refletir esse tráfego adicional, o ataque é detectado pelo destino.

Esses dados obtidos com simulação encorajam o prosseguimento das investigações nessa direção. Vários aperfeiçoamentos são possíveis no modelo estatístico e na própria coleta de dados. Além disso, deve-se validar os resultados com dados de redes reais.

7. CONCLUSÕES

Este trabalho a apresenta a concepção e implementação de um modelo de segurança para redes móveis *ad hoc*, completamente distribuído e auto-organizado, que tem na combinação de serviços de segurança preventiva e corretiva sua característica mais relevantes. A prevenção se dá por meio do estabelecimento de um modelo de confiança que é usado como elemento chave em um serviço de autenticação. Este modelo é concretizado na forma de um serviço de certificação distribuído e auto-organizado. A correção se dá através de um sistema de detecção de intrusão, igualmente distribuído e auto-organizado. A principal contribuição ao estado da arte deste modelo consiste na definição de mecanismos precisos de interação entre esses serviços preventivos e corretivos para prover uma resposta automática e balanceada às intrusões.

No que diz respeito à concepção do serviço de certificação e à definição dos mecanismos de autenticação, a proposta deste trabalho tem alguns trabalhos prévios de outros autores como ponto de partida, mas acrescenta contribuições relevantes a esses, tais como: a definição de mecanismos de formação e manutenção de CRL e *cache* de certificados válidos; a reestruturação de parâmetros que definem o tipo de política de segurança de certificação a ser adotada, permitindo adaptar os mecanismos de certificação a diferentes contextos de aplicação de Manets – com especial enfoque na correção de vulnerabilidades relacionadas a ataques de Sybil; a definição de relações de confiança cruzadas entre ACDs diferentes – possibilitando a junção entre redes Manet iniciadas em momentos e condições diferentes; e a proposição de uma extensão de autenticação para Manet (MAE) que incorpora mecanismos de segurança capazes de securizar os diversos protocolos de roteamento e autoconfiguração definidos para esse tipo de ambiente de rede.

O modelo de confiança e o uso de técnica de autenticação ambos com base em certificados digitais é uma característica importante do modelo, na medida que provê uma identificação não repudiável das entidades que executam determinadas ações na rede. Na existência dessa característica baseia-se a interação entre os serviços corretivos e preventivos permitindo conceber-se uma resposta à intrusões com base na revogação de certificados. O contra-ponto desse modelo reside na imperatividade de se usar extensivamente primitivas de criptografia assimétrica. Essas operações representam ainda hoje um alto custo computacional que pode não ser tolerado em alguns ambientes. Entretanto, com o avanço da tecnologia computacional e o aumento gradativo da capacidade de processamento e de memória dos

sistemas, aliada a uma escolha coerente dos parâmetros dos serviços (e.g. tamanhos de chave e de coalizão, entre outros), o uso da criptografia assimétrica não deve ser impeditivo para a aplicação do modelo de segurança em vários cenários de aplicação de Manets. Além disso, os resultados alcançados motivam a procura por outras técnicas de criptografia assimétrica (e.g. criptografia de curvas elípticas), alternativas ao clássico RSA, que possibilitem um menor overhead computacional e de rede.

No tocante à concepção e implementação do sistema de detecção de intrusão, duas abordagens para o processo de detecção são usadas em conjunto, permitindo a detecção de diversos tipos de ataques em diferentes níveis da arquitetura do sistema. Uma abordagem por uso incorreto permite definir assinaturas precisas de ataque contra os protocolos de roteamento, gerando informações confiáveis acerca da existência de atividade maliciosas ou errôneas na rede com identificação positiva de sua origem. Desse modo, a confiança atribuída colaborativamente a alguns nodos/usuários pode lhes ser tirada dessa mesma forma, caso estes venham a ser comprometidos ou passem a agir incorreta ou faltosamente. Por outro lado, uma abordagem de detecção de intrusão por modelagem de comportamento permite não só identificar a ocorrência de ataques complexos como DDoS, mas fornece *insights* acerca de como os efeitos de tal ataque podem ser mitigados através dos mesmos mecanismos colaborativos já definidos para o modelo de segurança como um todo.

Da implementação real completa do modelo de segurança para os protocolos de roteamento OLSR e de autoconfiguração DCDP, tiram-se lições valiosas acerca do modelo proposto:

- A escolha dos parâmetros de implementação dos serviços de certificação e autenticação e a definição de políticas de segurança mais precisas para esses serviços acarretam em *trade-offs* importantes entre critérios de desempenho e overhead computacional e de rede, e requisitos de robustez da segurança dos próprios mecanismos auto-organizados e colaborativos.
- O uso combinado de certificação de detecção de intrusão reforça a segurança provida pelo primeiro, permitindo corrigir distorções posteriores através da eliminação de entidades mal comportadas da rede. Aproveitando-se as características de redundâncias e conectividade ponto-a-ponto das Manet, esses mecanismos de interação e colaboração podem ser projetados para que a colaboração ocorra, e complete com sucesso, de maneira localizada. Desse modo, tanto IDS quanto certificação são plenamente escaláveis mesmo para grandes redes ou redes com altas densidades de nodos.

- A escolha da MIB como fonte de informação para o serviço de detecção de intrusão permite a monitoração de aspectos diversos do funcionamento do sistema e da rede, senão também das aplicações. Este é um mecanismo que pode ser pronta e facilmente estendido para casos onde não se tenha as informações necessárias como parte da MIB padronizadas (e.g. MIB experimental OLSR), além de dispor de uma variedade representativa de informações acerca do sistema monitorado, a baixo custo.
- As experimentação em ambientes reais não possibilitam a avaliação de todos os aspectos requeridos para se validação completa dos mecanismos, dadas as dificuldades em se reproduzir as mais diversas condições de mobilidade, propagação e perfis de tráfego. Assim, os simuladores de rede, que encontram-se bastante avançados em termos de recursos para simulação de redes móveis, oferecem visões complementares às visões dos exercícios com as Manets reais. Em especial, os exercícios de simulação devem permitir avaliar os efeitos da dinamicidade da topologia da rede (e.g. mobilidade e propagação) nos serviços de segurança projetados.

Em relação à abordagem de detecção de intrusão por anomalia, é proposto um novo modelo para modelagem estatística do comportamento da rede, usando um modelo paramétrico de misturas de gaussianas, com detecção de anomalias por uso de critérios de classificação Bayesianos (cálculo de probabilidades *a posteori*). Esse modelo tem por objetivo permitir a modelagem simultânea de diferentes tipos de eventos (e.g. aplicações) que se reflitam em cima de um mesmo conjunto de variáveis disponíveis para monitoração. Os resultados experimentais preliminares indicam que esse tipo de modelo pode ser adequado, com uma escolha cuidadosa das variáveis a serem modeladas e monitoradas. Não obstante, esse modelo, além de precisar de uma maior validação com dados de referência provenientes de redes reais, possui algumas limitações importantes que precisão ser investigadas e flexibilizadas, entre elas a dificuldade de se modelar dados não numéricos ou numéricos com características especiais (e.g. dados modulares tais como hora do dia ou dia da semana, etc.). Além disso, o modelo de misturas gaussianas paramétrico não é adequado para modelar dados de natureza mais complexa que não tenham a característica estatística da normalidade.

Em fim, as questões de segurança em redes móveis *ad hoc* ainda encontram-se em estágios iniciais de concepção e desenvolvimento, muito havendo ainda por se pesquisar acerca desse tópico importante e atual. Como trabalhos futuros que decorrem diretamente dos resultados apresentados nesta tese, pode-se destacar:

- Definição (formal) e implementação de um modelo para iniciação auto-organizada do serviços de certificação. Os resultados produzidos neste trabalho utiliza a figura de um negociador para fazer a distribuição inicial dos certificados e partes da chave privada da ACD entre os primeiros (*K*) nodos da rede. Existem mecanismos para geração cooperativa de uma chave privada RSA por entidades que não detêm nenhum conhecimento completo da chave gerada [38], assim como para a distribuição desta chave compartilhada, na forma de partes da chave similares às utilizadas nos serviços de certificação projetados. Torna-se necessário adaptar esses mecanismos aos requisitos de geração e distribuição do modelo de confiança proposto neste trabalho. Isso poderá eliminar a necessidade de se ter um negociador no estágio inicial da rede, tornado os mecanismos de certificação, de fato, completamente auto-organizados.
- Implementação dos modos de operação com criptografia simétrica. Os serviços de segurança estão projetados em detalhe e implementados apenas para o uso de criptografia assimétrica. Os resultados obtidos nesse trabalho comprovam que essa técnica ainda representa um alto custo computacional que pode ser proibitivo em ambientes com nodos de menor capacidade de processamento. A implementação dos serviços de segurança pode ser, portanto, completada com mecanismos de autenticação usando criptografia simétrica, através de protocolos de autenticação especialmente projetados e adaptados para ambientes espontâneos, como é o caso do protocolo TESLA.
- Definição de políticas de segurança e de técnicas de aplicação e imposição dessas políticas para os processos de certificação distribuída. Ainda que o modelo desenvolvido neste trabalho tenha flexibilizado a adoção de diferentes políticas de certificação distribuída, os métodos de verificação e imposição dessas políticas continuam indefinidos. Essa importante questão, ainda em aberto na literatura técnica especializada, merece aprofundamento.
- Extensão dos mecanismos de segurança para outros protocolos de roteamento Manet. Este trabalho apresenta uma análise de vulnerabilidades que considera os quatro principais protocolos de roteamento para Manet (AODV, OLSR, TBRPF e DSR), assim como uma extensão de autenticação para Manet (MAE), definida em conjunto com um serviço de certificação distribuída, que permite securizar preventivamente estes protocolos. Entretanto, os mecanismos de proteção corretiva do modelo de segurança são desenvolvidos apenas para o caso do protocolo OLSR. Para se estender o serviço de detecção e resposta a intrusão a outros protocolos de roteamento, ou

mesmo ao protocolo de autoconfiguração DCDP considerado neste trabalho, pode-se: (1) revisar a análise de vulnerabilidades para identificação e especificação de ataques concretos que possam ser implementados no gerador de ataques; (2) identificar e especificar as assinaturas desses ataques, em termos dos mecanismos de detecção já existentes ou incorporando recursos novos aos atualmente implementados; (3) definir e implementar os agentes de coleta local de dados (e.g. MIB e a comunicação *daemon* ↔ agente snmp); e (4) testar e validar o processo.

- Simulações para verificar o efeito da mobilidade nos serviços de segurança e, em especial, no desempenho e efetividade do IDS. A validação do serviço de detecção de intrusão no contexto deste trabalho está limitada a topologias quase estáticas, com baixa mobilidade. Entretanto, Manets são redes móveis por natureza! Assim, os resultados ora apresentados carecem de validação adicional se estudar os efeitos da mobilidade nos processos e na eficácia dos mecanismos. Em especial, deve-se levantar ainda a efetividade do IDS em termos de falsos positivos e falsos negativos decorrentes da mobilidade. O processo de resposta a intrusões deve passar por esse mesmo tipo de análise, para se verificar se não existem condições factíveis onde os adversários possam se movimentar para escapar da revogação de certificados.
- Serviço de controle de acesso ao nível de rede colaborativo (L-Firewall). Os resultados na detecção de intrusão por modelagem de comportamento mostraram pelo menos um tipo de ataque (DDoS) onde a colaboração entre nodos da rede para filtrar fluxos de pacotes indesejados pode ser uma resposta efetiva à detecção desse tipo de ataque, mitigando seus efeitos. Como as informações acerca do endereço de rede de origem (IP) dos fluxos não são confiáveis, os nodos devem cooperar para identificar os caminhos utilizados por esses fluxos com informações salto-a-salto, muitas vezes derivadas do endereçamento da camada de enlace. Essa é uma das funcionalidades que podem ser desenvolvidas através de serviços L-Firewall que cooperam entre si.
- Serviço de gestão da política de segurança (L-SPM). As bases de assinaturas de ataques, regras de filtragem de pacotes, listas de acesso, entre outras informações importantes para os serviços de segurança precisam de uma rotina de atualização compatível com a natureza das Manets. Um serviço auto-organizado de atualização da política de segurança e de seus reflexos nos parâmetros dos mecanismos de controle da política ainda é um grande desafio para as pesquisas correntes.
- Detecção de intrusão por modelagem do comportamento: O modelo de detecção de intrusão por anomalias de comportamento proposto ainda está em seus primeiros

estágios de desenvolvimento, tendo sido usado apenas com dados sintéticos que não representam necessariamente o comportamento real de uma rede. Esse modelo precisa ser validado com experimentos que utilizam dados reais. Além disso, uma série de melhorias e flexibilização de pré-condições da concepção do modelo podem ser realizadas, tais como a utilização de outros tipos de funções nucleares, a utilização de modelos de mistura semi-paramétricos, a adoção de modelos estocásticos (e.g. Markov) para eliminação da pré-condição de independência estatística entre realizações (eventos), entre outros.

Este trabalho foi validado através da apresentação de trabalhos em importantes congressos internacionais e da publicação em periódicos com grande visibilidade internacional [3,13,85,92,93,94,95,96,97]. Pretende-se disponibilizar, com licença GPL ou FreeBSD, os programas e melhoramentos em programas produzidos para obtenção dos resultados dessa pesquisa através do sítio http://www.manet.redes.unb.br.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Mühlethaler, and D. Raffo, “Securing the OLSR Protocol”, Med-Hoc-Net 2003, Mahdia, Tunisia, June 25-27, 2003.

[2] S. Akaho, “Mixture Model for Image Understanding and the EM Algorithm”, ETL Technical Report TR-95-13, 1995.

[3] P. Albers, O. Camp, J. Percher, B. Jouga, L. Mé, and R. Puttini - Security in Ad hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches. WIS 2002, Ciudad Real Spain, April 2002.

[4] A. Aresenault and S. Turner – Internet X.509 public key infrastructure, IETF, draft-ietf-pkix-roadmap-06.txt, 2000.

[5] M. Asaka – The Implementation of IDA: An Intrusion Detection Agent System. Proceedings of the 11th Annual FIRST Conference on Computer Security Incident Handling and Response, Brisbane, Australia, June, 1999.

[6] J. Balasubraniyan, J. Fernandez, D. Isacoff, E. Spafford, D. Zamboni - AAFID - Autonomous Agents For Intrusion Detection, Technical report 98/05, COAST Laboratory Purdue University, June 1998.

[7] D. Balfanz, D. Smetters, P. Stewart, and H. Wong. – Talking to strangers: Authentication in adhoc wireless networks. In Proceedings of the ISOC Network and
Distributed Systems Security Symposium, Feb. 2002.

[8] J. Barrus, N. Rowe - A Distributed Autonomous-Agent Network-Intrusion Detection and Response System. In the Proceedings of the 1998 Command and Control Research and Technology Symposium, Monterey, CA, June-July 1998.

[9] M. C. Bernardes, E. Moreira – Implementation of an intrusion detection system based on mobile agents. In Proceedings of 2000 International Symposium on Software Engineering for Parallel and Distributed Systems, pp. 158-164, 2000.

[10] J. Boleng, “Efficient network layer addressing for mobile ad hoc networks”, Tech. Rep. MCS-00-09, The Colorado School of Mines, 2000.

[11] K. Bradley, S. Cheung, N. Puketza, B. Mukherjee and R. Olsson - Detecting disruptive routers: a distributed network monitoring approach, Proceedings of the IEEE Symposium on Security and Privacy, pp. 115 –124, 1998.

[12] Broch et al. 1998, "A performance comparison of multi-hop wireless ad hoc network routing protocols".

[13] F. Buiati; R. Puttini; C. Barenco and R. de Sousa – Secure Autoconfiguration for Mobile Ad Hoc Networks, International Journal of Wireless and Mobile Computing, 2004.

[14] F. Buiati – Protocolo Seguro para Autoconfiguração de Endereços de Redes Móveis Ad Hoc. Dissertação de Mestrado, Publicação ENE.DM 180A/04, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 2004.

[15] L. Buttyan and J. P. Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. ACM Journal for Mobile Networks (MONET), special issue on Mobile Ad Hoc Networks, 2002.

[16] J. Cabrera, L. Lewis, R. Prasanth, X. Qin, W. Lee, and R. Mehra – Proactive detection of distributed denial of service attacks using MIB traffic variables – a feasibility study. Proceedings of the 7th IFIP/IEEE International Symposium on Integrated Network Management, Seattle, WA, USA, may 2001.

[17] S. Capkuny, L. Buttyan and J. Hubaux - Self-Organized Public-Key Management for Mobile Ad Hoc Networks, Swiss Federal Institute of Technology Lausanne (EPFL) Technical Report 2002-34, June , 2002.

[18] S. Capkuny, L. Buttyan, and J.-P. Hubaux - Self-organized public-key management for mobile ad hoc networks. IEEE Transactions on Mobile Computing, 2(1), January-March 2003.
[19] S. Capkuny, J. Hubaux, and L. Buttyan – Mobility Helps Security in Ad Hoc Networks. In Proceedings of MobiHoc'03.
[20] A. Cavalli and J. Orset – Secure Hosts Autoconfigurarion in Mobile Ad hoc Networks, in Proceedings of WWAN 2004.
[21] P.C. Chan, V. K. Wei - Preemptive distributed intrusion detection using mobile agents. In Proceedings of the Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2002), pp. 103-108, 2002.
[22] P. Cheeseman and J. Stutz, "Bayesian classification (AutoClass): theory and results. Advances in Knowledge Discovery and Data Mining, U. M. Fayyad, G. Piatetsky-Shapiro, R. Smyth and R. Uthurusamy (Eds.), Menlo Park, California: The AAAI Press, pp. 61-83, 1996.
[23]S. Cheshire, B. Aboba and E. Guttman - Dynamic Configuration of IPv4 Link-Local Addresses, IETF Internet Draft, draft-ietf-zeroconf-ipv4-linklocal-17.txt, July 2004.
[24] Y. Chun, L. Qin, L. Yong and Shi MeiLin – Routing protocols overview and design issues for self-organized network. Proceedings of 2000 IEEE International Conference on Communication Technology (ICCT 2000), pp. 1298-1303, 2000.
[25] T. Clausen and P. Jacquet - Optimized Link State Routing Protocol (OLSR) - IETF RFC 3626 (Experimental), October 2003.
[26] S. Corson and J. Marker – Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation consideration. RFC 2501 (informational), IETF, 1999.
[27] D. Curry, H. Debar, and Merrill Lynch - Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML). IETF Internet draft. June 2002.
[28] B. Dahill, K. Sanzgiri, B. N. Levine, C. Shields and E. Royer, "A secure routing protocol for ad hoc networks". In the Proceedings of the 2002 IEEE International Conference on Network Protocols (INCP 2002), Nov. 2002.
[29] H. Debar, M. Dacier and A. Wespi - A revised taxonomy for intrusion-detection systems, IBM Research Report, Zurich, 1999.
[30] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm" (with discussion). Journal of the Royal Statistical Society B 39 ,1-38, 1977.
[31] T. Dierks and C. Allen – The TLS Protocol Version 1.0 – IETF RFC 2246, 1999.
[32] J. Douceur – The Sybil Attack. 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), February 2002.
[33] R. Droms - Dynamic Host Configuration Protocol – IETF RFC 2131, March 1997.
[34] T. Droste - Weighted communication in a security compound,. Proceedings of the 5th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Service, 2001 (TELSIKS 2001), pp. 463-466, vol.2, Yugoslavia, Sept. 2001.
[35] S. Fenet, S, Hassas – A Distributed Intrusion Detection and response System Based on Mobil Autonomous Agents Using Social Insects Communication Paradigms. In First International Workshop on Security of Mobile Multiagent Systems (SEMAS2001), Montreal, Canada, May, 2001.
[36] L. Feeney, B. Ahlgren, and A. Westerlund. – Spontaneous networking: an application-oriented approach to ad hoc networking. IEEE Communications Magazine, June 2001.
[37] Y.F. Fou, F. Gong, C. Sargor, X. Wu, S. F. Wu, H. C. Chang, F. Wang - JINAO-Design and Implementation of a Scalable Intrusion Detection System for the OSPF Routing Protocol, Advanced Networking Research, MCNC Computer Science Dept, NC State University, February, 1999.

[38] P.-A. Fouque and J. Stern. Fully Distributed Threshold RSA under Standard Assumptions. Asiacrypt'01
[39] C. Fraley and A. E. Raftery, “MCLUST: Software for Model-Based Cluster and Discriminant Analysis”, Technical Report No.342, Department of Statistics, University of Washington, 1998.
[40] A. Genz, “Numerical Computation of Multivariate Normal Probabilities”, J. Comp. Graph Statistics vol.1, pp. 141-149 (1992)
[41] M. Guerrero and N. Asokan, “Securing Ad Hoc Routing Protocols”, in the Proceedings of 2002 ACM Workshop on Wireless Security (WiSe’2002), in conjunction with the ACM MOBICOM2002, September, 2002.
[42] S. Gwalani, E. Royer, G. Vigna, R. Kemmerer – AODVSTAT: Intrusion Detection in AODV. (work in progress)
[43] G. Helmer, J. Wong, V. Honavar, L. Miller, Y. Wang – Lightweight Agents For Intrusion Detection. To be published in The Journal of Systems and Software.
[44] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung - Proactive secret sharing or: how to cope with perpetual leakage," extended abstract, IBM T.J.Watson Research Center, November 1995.
[45] S. Hofmeyr, S. Forrest – Architecture of an Artificial Immune System. Evolutionary Computation 7(1), Morgan-Kaufmann, San Francisco, CA, pp. 1289-1296 (2000).
[46] R. Housley; W. Ford; W. Polk and D. Solo - Internet X.509 Public Key Infrastructure: Certificate and CRL Profile - RFC 3280, IETF Network Working Group, April 2002.
[47] Y. Hu, A. Perrig, and D. B. Johnson. Wormhole detection in wireless ad hoc networks. Technical Report TR01-384, Department of Computer Science, Rice University, December 2001.
[48] Y. C. Hu, D. Johnson, and A. Perrig. SEAD: Secure eficient distance vector routing for mobile wireless ad hoc networks. In Fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '02), June 2002, pages 3--13, June 2002.
[49] Y. C. Hu, A. Perrig, and D. Johnson, “Ariadne: A secure On-demand routing protocol for ad hoc networks”, in the Proceedings of ACM MobiCom 2002, Sep. 2002.
[50] Y. Hu; A. Perrig and D. Johnson - Efficient Security Mechanisms for Routing Protocols. IETF NDSS2003.
[51] Y. Huang, W. Fan, W. Lee, and P. Yu. Cross-feature analysis for detecting ad-hoc routing anomalies. In The 23rd International Conference on Distributed Computing Systems, May 2003.
[52] J. Hubaux, L. Buttyan, and S. Capkun. The quest for security in mobile ad hoc networks. In Proc. ACM MobiHOC, 2001.
[53] K. Ilgun, R. A. Kemmerer, and P. A. Porras – State Transition Analysis: A Rule-Based Intrusion Detection Approach. IEEE Transactions on Software Engineering, pp. 181-199, March 1995.
[54] P.Jacquet, A. Laouiti, P. Minet and L. Viennot, “Performance Analysis of OLSR Multipoint Relay Flooding in Two Ad Hoc Wireless Network Models”, Research Report-4260, INRIA, September 2001, RSRCP journal special issue on Mobility and Internet.
[55] W. Jansen. Intrusion Detection with Mobile Agents. To be published in Computer Communications Journal, Special Issue on Intrusion Detection.
[56] A. S. Javits and A. Valdets, “The SRI IDES Statistical Anomaly Detector”, Proc. of IEEE Symposium of Research on Security and Privacy, pp. 316-326, may 1991.
[57] J. Jeong, H. Cha, J. Park and H. Kim, “Ad Hoc IP Address. Autoconfiguration”, draft-jeong-adhoc-ip-addr-autoconf-00.txt, Internet Engineering Task Force, MANET Working Group, May 2003.

[58] D. B. Johnson, David A. Maltz and Yih-Chun Hu - The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR), INTERNET DRAFT, MANET working group, version 10, Jul. 2004.
[59] R. A. Johnson, D. A. Wichern, D. W. Wichern, "Applied Multivariate Statistical Analysis – 4th Edition", Prentice-Hall, 1998.
[60] I.T. Jolliffe , "Principal Component Analysis", Springer Series in Statistics, May 1986.
[61] V. Kawadia, Y. Zhang, and B. Gupta. System services for ad-hoc routing: Architecture, implementation and experiences. In The First International Conference on Mobile Systems, Applications, and Services (MobiSys 2003), San Francisco, California, May 2003.
[62]A. Khalili, J. Katz and W. Arbaugh - Toward Secure Key Distribution in Truly Ad-Hoc Networks, in Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT-w'03).
[63] J. Kiniry and D. Zimmerman - Special Feature: A Hands-On Look at Java Mobile Agents. IEEE Internet Computing, Vol. 1, No. 4, July/August 1997.
[64] C. Ko, M. Ruschitzka, and K. Levitt. Execution Monitoring of Security-Critical Programs in Distributed Systems: A Specification-based Approach. Proceedings of the 1997 IEEE Symposium on Security and Privacy, 1997.
[65] J. Kohl and B. Neuman – The Kerberos network authentication service (version 5), IETF, RFC 1510.
[66] J. Kong, P. Zerfos, H. Luo, S. Lu and L. Zhang, "Providing robust and ubiquitous security support for MANET," IEEE ICNP 2001, 2001.
[67] Christopher Krügel, Thomas Toth - Flexible, Mobile Agent Based Intrusion Detection for Dynamic Networks. Proceedings of European Wireless (EW2002), Italy, February 2002.
[68] Carl E. Landwehr, Alan R. Bull, John P. McDermott, and William S. Choi - A taxonomy of computer program security flaws. ACM Computing Surveys, 26(3):211-254, September 1994.
[69] D. Curry, H. Debar, and Merrill Lynch - Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML). IETF Internet draft. June 2002.
[70] W. Lee; S. J. Stolfo; and K. W. Mok - A data mining framework for building intrusion detection models. Proceedings of the 1999 IEEE Symposium on Security and Privacy, 1999.
[71] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang – Self-securing Ad Hoc Wireless Networks. Proceedings of the Seventh IEEE International Symposium on Computers and Communications (ISCC'02), 2002.
[72] H. Luo and S. Lu, "Ubiquitous and robust authentication services for ad hoc wireless networks," UCLA Computer Science Technical Report 200030, Oct. 2000.
[73] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehaviour in mobile ad hoc networks. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking, Boston, MA, August 2000.
[74] S. Martino - A mobile agent approach to intrusion detection, technical report, Joint Research Centre Institute for Systems, Informatics and Safety, Italy, June 1999.
[75] K. McCloghrie; and A. Bierman - Entity MIB (Version 2). IETF Request for Comment 2737, December 1999.
[76] G. J. McLachlan, D. Peel, K. E. Basford and P. Adams, "The EMMIX Software for the Fitting of Mixtures of Normal and t –Components", Journal of Statistical Software, v. 04, 1999.
[77] P. Mell, D. Marks, M. McLarnon – A Denial-of-Service Resistant Intrusion Detection Architecture. Computer Networks, Special Issue on Intrusion Detection, Elsevier Science BV, November 2000.

[78] A. Misra, S. Das, A. McAuley, and S. K. Das – Sun - Autoconfiguration, Registration and Mobility Management for Pervasive Computing. IEEE Personal Communications, vol. 08, Issue 04, Aug. 2001.
[79] V. Mittal and G. Vigna. Sensor-based intrusion detection for intra-domain distance-vector routing. In R. Sandhu, editor, Proceedings of the ACM Conference on Computer and Communication Security (CCS'02), Washington, DC, November 2002. ACM Press.
[80] M. Mohsin and R. Prakash – IP Address Assignment in a Mobile Ad Hoc Network. IEEE Milcom 2002.
[81] S. Nesargi and R. Prakash, "MANETconf: Configuration of hosts in a mobile ad hoc networks" INFOCOM, 2002.
[82] P. Ning and K. Sun. How to misuse AODV: A case study of insider attacks against mobile ad-hoc routing protocols. In Proceedings of the 4th Annual IEEE Information Assurance Workshop, pp. 60-67, June 2003.
[83] R. Ogier, F. Templin and M. Lewis - Topology Dissemination Based on Reverse-Path Forwarding (TBRPF), IETF RFC 3684 (Experimental), Feb. 2004.
[84] P. Papadimitratos and Z. J. Haas. Secure routing for mobile ad hoc networks. SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), Jan 2002.
[85] JM. Percher; R. Puttini; L. Mé; O. Camp; B. Jouga; P. Albers - Un système de détection d'intrusion distribué pour réseaux ad hoc, TSI, France, 2004.
[86] Charles E. Perkins and Pravin Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", in Proceedings of the SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications, pages 234–244, August 1994.
[87] C. Perkins, J. Malinen, R. Wakikawa, E. Royer and Y. Sun - IP Address Autoconfiguration for Ad hoc Networks, draft-ietf-manet-autoconf-01.txt, Internet Engineering Task Force, MANET Working Group, November 2001.
[88] C. E. Perkins and E. M. Royer - Ad hoc on-demand distance vector (AODV) Routing. IETF RFC 3561 (Experimental), July 2003.
[89] A. Perrig, R. Canetti, J.D. Tygar, and D. Song - Efficient Authentication and Signing of Multicast Streams over Lossy Channels. In IEEE Symposium on Security and Privacy, pages 56-73, May 2000.
[90] Phillip A. Porras, Peter G. Neumann - EMERAL - Event Monitoring Enabling Responses to Anomalous Live Disturbances, Conceptual Overview, December, 1996.
[91] N. Prigent, C. Bidan, J.P. Andreaux and O. Heen – Secure Long Term Communits in Ad Hoc Networks. ACM SASN 2003.
[92] R. Puttini, Z. Marrakchi and L. Mé - Bayesian Classification Model for Real-Time Intrusion Detection, 22th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering (MAXENT'2002). August 2002.
[93] R. Puttini; L. Me; R. de Sousa – MAE - Manet Authentciation Extension for Securing Manet Routing Protocols. In Proceedings of 5th IFIP TC6 International Conference on Mobile and Wireless Communications Networks, 2003.
[94] R. Puttini; J.M Percher; L. Me; O. Camp; R. de Sousa - A Modular Architecture for a Distributed IDS for Mobile Ad Hoc Networks. International Conference on Computer Science and Applications in Lecture Notes on Computer Science 2669:91-113, Springer, 2003.
[95] R. Puttini; J.M. Percher; L. Me; R. de Sousa - A Fully Distributed IDS for Manet. In Proceedings of 9th IEEE International Symposium on Computers Communications, 2004.
[96] R. Puttini; L. Me; R. de Sousa - Preventive and Corrective Protection for

Mobile Ad Hoc Network Routing Protocols. In Proceedings of 1st International Conference on Wireless On-demand Network Systems in Lecture Notes on Computer Science, Springer, 2004.
[97] R. Puttini; L. Me; R. de Sousa – On the Vulnerabilities and Protection of Mobile Ad Hoc Routing Protocols. In Proceedings of 3rd IEEE International Conference on Networks, 2004.
[98] R. Rivest, A. Shamir and L. Adleman - A method for obtaining digitial signatures and public key cryptosystems, Communications of the ACM, Feb. 1978.
[99] S. J. Roberts, R. Everson and I. Rezek, "Maximum Certainty Data Partitioning", Pattern Recognition, 33:5, pp. 833-839, 1999.
[100] E. Royer and C. Toh. – A review of current routing protocols for ad hoc mobile wireless networks. IEEE Personal Communications Magazine, pp. 46-55, 1999.
[101] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou. Specication-based anomaly detection: a new approach for detecting network intrusions. In ACM Computer and Communication Security Conference (CCS), 2002.
[102] A. Shamir – How to Share a Secret. Communications of the ACM, 22(11):612-613, 1979.
[103] Eugene H. Spafford and Diego Zamboni – Intrusion detection using autonomous agents. Computer Networks, 34(4):547-570, October 2000.
[104] S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, D. Zerkle - GrIDS- A Graph Based Intrusion Detection System for Large Networks, Computer Security Laboratory,Department of Computer Science,University of California, Davis, 1996.
[105] S. Staniford-Chen, and L. Heberlein – Holding Intruders Accountable on the Internet. Proceedings of the 1995 IEEE Symposium on Security and Privacy, 1995.
[106] Steven R. Snapp, , James Brentano, Gihan V. Dias, Terrance L. Goan, L. Todd Heberlein, Che-Lin Ho, Karl N. Levitt, Biswanath Mukkherjee, Stephen E. Smaha, Tim Grance, Daniel M. Teal, and Doug Mansur - DIDS-Distributed Intrusion Detection System, Computer Security Laboratory, Department of Computer Science, University of California, Davis, June 1992.
[107] F. Silveira e M. Hanashiro – Serviços de Certificação para redes móveis ad hoc. Monografia de projeto final, publicação UnB.LabRedes.PFG.02/2003, Departamento de Engenharia de Redes de Comunicação, Universidade de Brasília, Brasília – DF, 2003.
[108] A. Tønnesen - Impementing and extending the Optimized Link State Routing Protocol, Master Degree Thesis, University of Oslo (UniK), 2004.
[109] R. Thayer, N. Doraswamy and R. Glenn – IP Security Document Roadmap – IETF RFC 2411 (Informational), November 1998.
[110] D. M. Titternington, "Recursive Parameter Estimation using Incomplete Data", J. R. Statist. Soc. B, n.o 46, pp. 257-267.
[111] C.-Y. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt. A specification-based intrusion detection system for AODV. In ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03), October 2003.
[112] Gregory B White, Eric A. Fish and Udo Pooch - CSM - Cooperating Security Managers: a peer based intrusion detection system, IEEE Networks, pages 20-23, January/February 1996.
[113] F. Wang, F. Wu – On the vulnerabilities and Protection of OSPF Protocol. Proceedings of 1998 International Conference on Computer Communications and Networks, 1998.
[114] Kilian Weniger - Passive Duplicate Address Detection in Mobile Ad Hoc Networks, IEEE WCNC 2003, March 2003.

[115] M. Wood and M. Erlinger. - Intrusion Detection Message Exchange Requirements, IETF Internet Draft, October 22, 2002. http://www.ietf.org/internet-drafts/draft-ietf-idwg-requirements-10.txt
[116] H. Yang, X. Meng and S. Lu, "Self-Organized Network Layer Security in Mobile Ad Hoc Networks", in the Proceedings of ACM Workshop on Wireless Security – 2002 (WiSe'2002), in conjunction with the ACM MOBICOM2002, September, 2002.
[117] Y. Zhang and W. Lee – Intrusion detection in wireless ad hoc networks. Proceedings of 6th ACM Annual International Conference on Mobile Computing and Networking (MOBICOM 2000), ACM Press, New York, pp. 275-283, 2000.
[118] H. Zhou, L. Ni and M. Mutka - Prophet Address Allocation for Large Scale MANETs. In Proceedings of IEEE INFOCOM 2003.
[119] L. Zhou and Z. J. Haas. Securing ad hoc networks. IEEE Network Magazine, 13(6):24-30, November/December 1999.
[120] L. Zhou, F. Schneider and R. Renesse – COCA: A Secure Distributed Online Certification Authority, in ACM Transactions on Computer Systems, 20(4): 329–368, 2002.
[121] P. Zimmermann. The Official PGP User's Guide. MIT Press, 1995.

ANEXO I – TECNOLOGIAS DE REDE SEM FIO IEEE 802.11B OU WI-FIAO MODO AD HOC[48]

Esta é, de longe, a tecnologia de redes sem fio (WLAN) mais amplamente utilizada hoje em dia. Trata-se de uma tecnologia que tem custos bastante atraentes e com velocidades de até 11Mbps. Sua freqüência de operação é de 2,4GHz, com 11 canis disponíveis para comunicação, sendo que apenas três são os canais que se operados em paralelo não exercem nenhuma interferência inter-canal significativa: os canais 1, 6 e 11. Na prática, esses canais, se usados simultaneamente em uma mesma região, não geram degradação do sinal que possa afetar a velocidade final do enlace. A tecnologia IEEE 802.11b foi endossada por um grupo representativo de fabricantes e uma convenção foi adotada para facilitar a identificação de produtos interoperáveis nesta tecnologia. A essa certificação deu-se o nome de Wi-Fi, que significa que *Wireless Fidelity* (fidelidade sem fio).

O alcance conseguido com o IEEE 802.11b depende muito do tipo de antena utilizada, que pode variar em ganho e tecnologia, porém a regra para uma antena padrão de 1dBi é de um alcance de 100m para áreas fechadas (*indoor*) e até 300m para áreas abertas ou externas (*outdoor*). Alterando-se a antena usada, pode-se conseguir distâncias da ordem de quilômetros e, por isso, hoje já existem enlaces que operam a essas distâncias.

No protocolo IEEE 802.11b, o mecanismo fundamental de acesso ao meio é chamado de *distributed coordination function* (DCF). Este mecanismo é baseado em um esquema de acesso aleatório usando detecção de portadora evitando-se colisões. Desse modo, esse protocolo configura-se como um método de acesso CSMA/CA (*carrier sense multiple access with collision avoidance*). Nesse protocolo, sempre que uma estação tem algum pacote para transmitir, ela monitora a atividade do canal. Se o canal estiver ocioso por um período maior que o tempo entre quadros distribuídos (*distributed interframe space* – DIFS), a estação transmite o pacote. Senão, ela monitora o canal até que este esteja ocisoso por um período de tempo igual a DIFS e, então, inicia um contador de duração aleatória (*backoff*) antes de iniciar sua transmissão, tentando minimizar a probabilidade de uma nova colisão. Além disso, para que uma única estação não monopolize o canal, esta precisa iniciar ser contador sempre que transmitir dois ou mais pacotes seguidos. Como uma estação não tem como detectar se houve uma colisão, um reconhecimento (ACK) é transmitido pela estação de destino logo após um

[48] Adaptado de http://www.brasilmobile.com

curto período de tempo chamado *short interframe space* (SIFS), sempre que um pacote é recbido sem erros. Se, após o período de tempo igual a ACKtimeout um ACK não for recebido, a estação transmissora assume que ocorreu um erro (e.g. colisão) e re-agenda a transmissão, de acordo com o tamanho da janela de *backoff*.

O mecanismo de segurança em nível de enlace das redes IEEE 802.11b é conhecido como *wired equivalent privacy* (WEP). WEP se encarrega de cifrar os dados transmitidos através da rede. Existem dois padrões WEP, de 64 e de 128 bits. O padrão de 64 bits é suportado por qualquer ponto de acesso ou interface que siga o padrão Wi-Fi, o que engloba todos os produtos comercializados atualmente. O padrão de 128 bits por sua vez não é suportado por todos os produtos. Para habilitá-lo, será preciso que todos os componentes usados na rede suportem o padrão, caso contrário os nós que suportarem apenas o padrão de 64 bits ficarão fora da rede.

Na verdade, o WEP é composto de duas chaves distintas, de 40 e 24 bits no padrão de 64 bits e de 104 e 24 bits no padrão de 128. Por isso, a complexidade criptográfica usada nos dois padrões não é a mesma que seria em algoritmos criptográficos com chaves de 64 e 128 de verdade. Além do detalhe do número de bits nas chaves de cifração, o WEP possui outras vulnerabilidades. Alguns programas já largamente disponíveis são capazes de quebrar as chaves de cifração caso seja possível monitorar o tráfego da rede durante algumas horas e a tendência é que estas ferramentas se tornem ainda mais sofisticadas com o tempo.

O WEP vem desativado na grande maioria dos pontos de acesso, mas pode ser facilmente ativado através do utilitário de configuração. O mais complicado é que é preciso definir manualmente uma chave de criptografia (um valor alfanumérico ou hexadecimal, dependendo do utilitário) que deverá ser a mesma em todos os pontos de acesso e estações da rede. Nas estações, a chave, assim como o endereço ESSID e outras configurações de rede, pode ser definida através de outro utilitário, fornecido pelo fabricante da placa.

ANEXO II – CRIPTOGRAFIA DE LIMIAR

Em 1979, A. Shamir [102] propôs um modelo que permite a divisão de uma informação *D* em *N* partes, de forma que *D* pode ser reconstruído a partir de *K* partes, mas o conhecimento de $K-1$ partes não revela nenhuma informação sobre *D*. Essa técnica foi proposta para a implementação de um sistema criptográfico de gerenciamento de chaves eficiente.

Este modelo foi chamado de esquema de limiar (*K*,*N*), onde *N* é o número total de partes de *D* e *K* é o número mínimo de partes necessárias para se reconstruir *D*. Segundo Shamir, um esquema eficiente de limiar pode ser muito útil no gerenciamento de chaves criptográficas, já que permite que se proteja uma chave criptográfica.

O esquema é baseado na interpolação polinomial: dados *K* pontos em um plano de duas dimensões $(x_1, y_1),...,(x_k, y_k)$, com diferentes valores de x_i, existe um e somente um polinômio $q(x)$, de grau $K-1$, com $q(x_i) = y_i$ para todo *i*. Para dividir *D* em *N* partes D_i, gera-se um polinômio randômico de grau $K-1$, conforme mostrado na equação 1.

$$q(x) = a_0 + a_1 x + ... + a^{k-1} x^{k-1} \quad (1)$$

onde: $a_0 = D$.

Em seguida, calcula-se D_i (equação 2):

$$D_i = q(x_i) \bmod p \quad (2)$$

onde: *p* é um número primo, x_i são chamados identificadores de cada uma das partes de *D* e o módulo tem a finalidade de fornecer maior precisão por garantir um campo onde a interpolação é possível.

Este número primo é maior que *D* e *N*. Os coeficientes $a_1,...,a_{k-1}$ são randomicamente escolhidos a partir de uma distribuição uniforme de inteiros em $[0, p)$.

Dado qualquer subconjunto de *K* valores de D_i, com os respectivos identificação (x_i), pode-se calcular os coeficientes $q(x)$ pode meio de interpolação e depois se calcular $D = q(0)$. Porém, o conhecimento de apenas $K-1$ valores não é suficiente para calcular *D*.

Shamir definiu algumas propriedades para o esquema de limiar (*K*,*N*). O tamanho da parte D_i não deve ser maior do que o tamanho do dado original *D*. Quando é fixado um valor para *K*, as partes D_i podem ser dinamicamente acrescentadas ou eliminadas sem que as outras partes sejam afetadas. Além disso, é fácil trocar as partes D_i sem modificar *D*, desde que se crie um novo polinômio com o mesmo *D*.

Realizar esta troca freqüentemente acrescenta segurança ao sistema já que as partes quebradas não podem ser acumuladas, a não ser que todas elas sejam originadas de um mesmo polinômio $q(x)$.

Por fim, pode-se criar um esquema hierárquico em que o número de partes para determinar *D* depende da importância das mesmas. Por exemplo, se o presidente de uma empresa recebe três valores de D_i, o vice recebe dois valores, os executivos recebem apenas um valor e define-se o esquema (3,*N*). Com isso, são necessários três executivos para acessar o sistema ou um executivo e o vice-presidente ou apenas o presidente.

ANEXO III – SINTAXE DAS MENSAGENS DO PROTOCOLO DE CERTIFICAÇÃO

A sintaxe de mensagens do serviço de certificação para Manet e dos objetos de certificação encapsulados nessas mensagens é mostrada nas Figura A3.1 e Figura A3.2 abaixo, respectivamente.

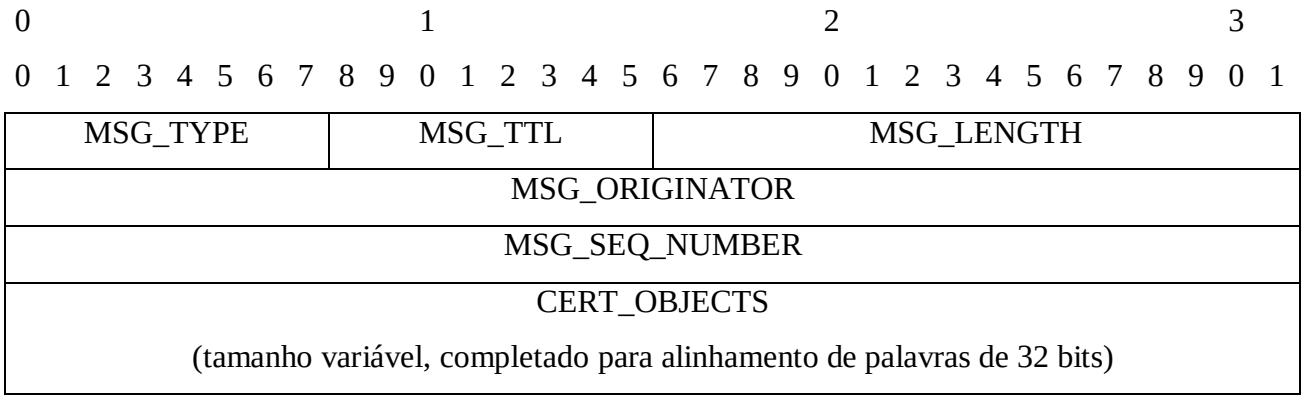


Figura A3.1 – Sintaxe de Mensagens do L-Cert

Onde:

- MSG_TYPE: campo usado para diferenciar a mensagem de outros tipos de mensagens encapsuladas;
- MSG_TTL: *time-to-live*, usado para definir o alcance, em número de saltos, da disseminação (*flooding*) da mensagem; (e.g. 0 se na vizinhança de um salto e 255 se toda a rede);
- MSG_LENGTH: tamanho, em bytes, da mensagem;
- MSG_ORIGINATOR: originador da mensagem (e.g. endereço IP);
- MSG_SEQ_NUMBER: número de seqüência da mensagem, para evitar o processamento de mensagens duplicadas ou antigas (cada nodo inicia um gerador de números pseudo-aleatório que gera os números de seqüência para suas mensagens);
- CERT_OBJECTS: um ou mais objetos de certificação.

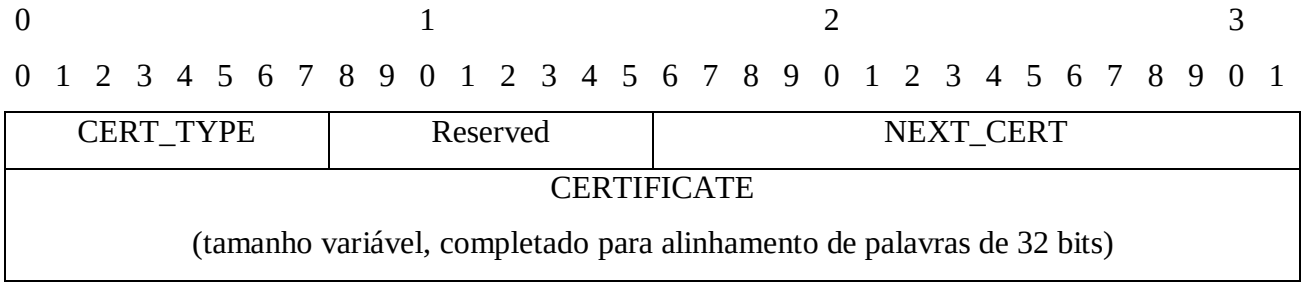


Figura A3.2 – Sintaxe de cada CERT_OBJECT

Onde:

- CERT_TYPE: identificador do tipo de objeto;
- CERT_LENGTH: tamanho, em bytes, do objeto;
- CERT: objeto de certificação, conforme o campo CERT_TYPE.

ANEXO IV – SINTAXE DA EXTENSÃO DE AUTENTICAÇÃO PARA MANET (MAE)

As sintaxes propostas para a MAE e para seus objetos de autenticação são mostradas nas Figura A4.1 e Figura A4.2, respectivamente.

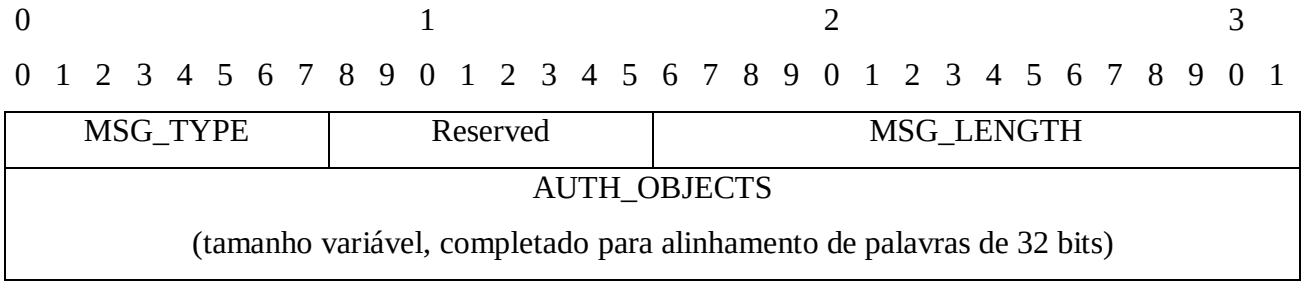


Figura A4.1 – Sintaxe da MAE

Onde:

- MSG_TYPE: campo usado para diferenciar a mensagem de outros tipos de mensagens encapsuladas;
- MSG_LENGTH: tamanho, em bytes, da mensagem;
- AUTH_OBJECTS: um ou mais objetos de autenticação.

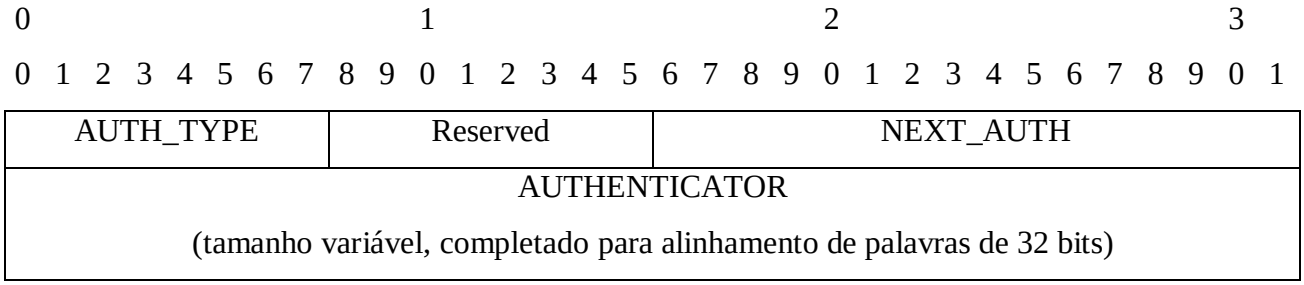


Figura A4.2 – Sintaxe de cada Objeto de Autneticação

Onde:

- AUTH_TYPE: tipo do objeto de autenticação;
- NEXT_AUTH: tamanho, em bytes, do objeto;
- AUTHENTICATOR: objeto de autenticação (e.g. DS, MAC, HC, etc.).

ANEXO V – ESPECIFICAÇÃO XML (DTD) PARA AS MENSAGENS DO L-IDS

```
<!-- DTD for L-IDS -->

<!ELEMENT MESSAGE
(ID,ID_ENT_ORIG,PAR_ATAC_ALVO*,TIPO_MSG,IS_LOCAL,ID_ENT_DEST,IS_FLO
ODED,TTL)>

   <!ELEMENT ID (#PCDATA)>
   <!ELEMENT ID_ENT_ORIG (#PCDATA)>

   <!ELEMENT PAR_ATAC_ALVO (ATACANTE,ALVO)?>
      <!ELEMENT ATACANTE (ID_ENT_REDE)?>
         <!ELEMENT ID_ENT_REDE (#PCDATA)>
      <!ELEMENT ALVO (ID_ENT_REDE)?>
         <!ELEMENT ID_ENT_REDE (#PCDATA)>

   <!ELEMENT TIPO_MSG
(MSG_EVENT|MSG_QUERY|MSG_QUERY_PERIODIC|MSG_ALERT|MSG_STATE_D
ETECT)>

   <!ELEMENT MSG_EVENT (#PCDATA)>

   <!ELEMENT MSG_QUERY (#PCDATA)>

   <!ELEMENT MSG_QUERY_PERIODIC (PERIOD)>
      <!ELEMENT PERIOD (#PCDATA)>

   <!ELEMENT MSG_ALERT (ID_ATAQUE)>
      <!ELEMENT ID_ATAQUE (#PCDATA)>

   <!ELEMENT MSG_STATE_DETECT (ID_ATAQUE,ISCLONE)>
      <!ELEMENT ID_ATAQUE (#PCDATA)>
      <!ELEMENT IS_CLONE (#PCDATA)>

   <!ELEMENT ID_ENT_ORIG (#PCDATA)>

   <!ELEMENT IS_LOCAL (#PCDATA)>

   <!ELEMENT IS_FLOODED (#PCDATA)>

   <!ELEMENT TTL (#PCDATA)>

   <!ATTLIST IS_CLONE
      valor (TRUE|FALSE) #REQUIRED>
   <!ATTLIST IS_LOCAL
```

```
    boolean (TRUE|FALSE) #REQUIRED>
<!ATTLIST IS_FLOODED
    boolean (TRUE|FALSE) #REQUIRED>
```

ANEXO VI – MIB EXPERIMENTAL PARA O PROTOCOLO OLSR EM FORMATO ASN-1

```
RAHMS-OLSR-MIB DEFINITIONS ::= BEGIN

IMPORTS
   MODULE-IDENTITY, OBJECT-TYPE
   experimental, IpAddress        FROM SNMPv2-SMI ;

rahmsOlsrMIB MODULE-IDENTITY
   LAST-UPDATED "0207051145Z"
   ORGANIZATION "ESEO"
   CONTACT-INFO "rahms@eseo.fr"
   DESCRIPTION  "The MIB module for RAHMS networks"
   ::= { experimental 6060 }

rahms  OBJECT IDENTIFIER ::= { experimental 6262 }

-- the olsr group
olsr      OBJECT IDENTIFIER ::= { rahms 1 }

-- OLSR Neighbor Table
-- The OLSR Neighbor Table contains information concerning this entity's
-- existing neighbors and the status of the link between this host and
-- each of its neighbors

olsrNeighborTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF olsrNeighborEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION "A table containing OLSR neighbor information."
   ::= { olsr 1 }

olsrNeighborEntry OBJECT-TYPE
   SYNTAX      olsrNeighborEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION "A conceptual row of the olsrNeighborTable containing
               Information about the connection towards a particular
               OLSR neighbor. Each row of this table is transient, in
               that it ceases to exist when (or soon after) the
               connection with a neighbor is lost."
   INDEX   { olsrNeighborAddress}
   ::= { olsrNeighborTable 1 }
```

```
olsrNeighborEntry ::= SEQUENCE {
      olsrNeighborState     INTEGER,
      olsrPreviousNeighborState INTEGER,
      olsrNeighborAddress   IpAddress
   }

olsrNeighborState OBJECT-TYPE
   SYNTAX      INTEGER { ASYM(1),SYM(2),MPR(3),LOST(4) }
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION "The state of this OLSR neighbor connection."
   ::= { olsrNeighborEntry 1 }

olsrPreviousNeighborState OBJECT-TYPE
   SYNTAX      INTEGER { ASYM(1),SYM(2), MPR(3),LOST(4) }
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION "The previous state of this OLSR neighbor connection."
   ::= { olsrNeighborEntry 2 }

olsrNeighborAddress OBJECT-TYPE
   SYNTAX      IpAddress
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION "This neighbor IP address"
   ::= { olsrNeighborEntry 3 }
END
```