Universidade Federal Fluminense

Wagner Gaspar Brazil

Protegendo Redes ad hoc com Certificados Digitais e Limite Criptográfico

NITERÓI

2007

Universidade Federal Fluminense

Wagner Gaspar Brazil

Protegendo Redes ad hoc com Certificados Digitais e Limite Criptográfico

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

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

NITERÓI

2007

Protegendo Redes ad hoc com Certificados Digitais e Limite Criptográfico

Wagner Gaspar Brazil

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

Aprovada por:

Prof. Célio Vinicius Neves Albuquerque, Ph.D. / IC-UFF
(Orientador)

Prof. Eugene Francis Vinod Rebello, Ph.D. / IC-UFF

Prof. Bruno Richard Schulze, D.Sc. / LNCC

Niterói, 8 de Outubro de 2007.

"O estudo em geral, a busca da verdade e da beleza são domínios em que nos é consentido ficar crianças toda a vida"

Albert Einstein

À minha esposa Márcia incansável companheira.

Agradecimentos

Toda vez que se chega ao fim de um ciclo é importante reconhecer o valor das pessoas que nos ajudaram a chegar aos nossos objetivos. Neste país de memória curta em que vivemos, as pessoas que têm valor geralmente são esquecidas ou então só são lembradas pós-morte. Da minha parte, prefiro receber todas as homenagens em vida mesmo.

Sendo assim, não posso deixar de lembrar de agradecer primeiramente a Deus por me dar saúde e disposição para completar este curso. Como humilde servo, vejo que hoje, com os conhecimentos adquiridos, caminhei um pouco mais na direção da sua magnânima sabedoria.

Minha esposa Márcia Valéria deveria ganhar um capítulo à parte nesta dissertação. Desde o início me deu apoio e nos momentos difíceis de dúvidas não me fez desviar do caminho. Peço desculpas e agradeço pelos dias de sol sem praia, pelas noites sem sair com os amigos, pelas horas passadas nos estudos em que não pude lhe dar atenção. Esta vitória é nossa !

Neste momento é impossível não se emocionar e não lembrar de nossa família, principalmente meus pais e avós. Eles que me criaram, me deram amor, carinho e atenção. Eles que me ensinaram o valor das coisas, me deram princípios morais e o mais importante: educação. Sempre aprendi com eles que a única coisa que não se tira de uma pessoa é sua educação. Ninguém pode roubar seus conhecimentos ou sua educação. A educação liberta para o mundo e nos faz pessoas melhores. Melhores para nós mesmos e melhores para os outros. Por isso hoje, eu agradeço a eles por me darem a oportunidade de ter estudado em ótimas escolas às custas de seus sacrifícios.

Gostaria de agradecer também aos professores da UFF que me acolheram e me ensinaram. Fiz minha graduação nesta escola e desde os idos de 1990 aprendi a amar esta instituição. Aqui me formei, aqui me casei e fiz meus melhores amigos. Os professores sempre foram o grande diferencial desta universidade. Alguns me acompanharam desde a graduação como as professoras Anna Dolesji, Lúcia Drummond e Cristina Boeres. Todas

três me acompanharam na graduação e agora na pós-graduação. Realmente sou uma pessoa de sorte por ter tido mestres tão bons e especiais. Faço um agradecimento especial às professoras Anna Dolesji e Lúcia Drummond por terem me incentivado a começar o curso me dando ótimas cartas de referência.

Todos os professores que já passaram na minha vida foram importantes até agora e terão sempre um grande espaço no meu coração. Todos sem exceção me deram sempre mais do que eu esperava. Gostaria de citar a professora Simone Martins de arquitetura de computadores, professor Michael Stanton de redes, professor Vinod de arquiteturas paralelas. A todos vocês meu muito obrigado.

Neste ponto não posso me esquecer do meu orientador que tanto tem me ajudado. O professor Célio foi desde o início uma pessoa que conseguiu tirar o melhor de mim. Desde a disciplina de Redes de Computadores começou, mesmo que de maneira não oficial, a me orientar nos meus estudos e juntos conseguimos bons resultados publicando trabalhos, definição do tema e desenvolvendo esta dissertação. Hoje tenho certeza que ganhei mais um amigo.

Preciso destacar também o apoio dos colegas do Proderj que precisaram trabalhar um pouco mais para compensar minhas ausências. Também do Proderj tenho que agradecer ao vice-presidente, Dr. Paulo Coelho e à minha antiga diretora, Mônica Moreira por terem me liberado, mesmo que parcialmente, para assistir às aulas. Sem o apoio e a compreensão deles eu não teria nem como começar o curso.

Meu muito obrigado a meus colegas da UFF. Em especial Alexandre Sena, Aline Nascimento e Etienne Oliveira. Todos futuros doutores. Não esmoreçam. Vocês conseguem e merecem.

Por fim, gostaria de agradecer ao povo brasileiro que através dos seus impostos me proporcionou a oportunidade de concluir meu curso superior e um curso de mestrado numa instituição pública tão renomada. No fim de tudo, é para o povo que nossos conhecimentos devem ser aplicados. Para melhorar a vida deste povo tão bom e tão sofrido. Do fundo do coração meu muito obrigado.

Resumo

Este trabalho propõe uma arquitetura de ICP de alta disponibilidade e robusta de forma a minimizar os ataques do tipo buraco negro e falsificação de identidade contra redes ad hoc. A arquitetura se propõe a trocar um número de mensagens de controle reduzido aumentando assim a sua escalabilidade e desempenho. Para atingirmos este objetivo propomos que os diversos nós de uma rede ad hoc utilizem o serviço distribuído de certificação digital para autenticar e cifrar suas mensagens. Particularmente os protocolos de roteamento devem usar a autenticação e cifragem para se protegerem dos ataques citados acima garantindo assim confidencialidade, autenticidade e integridade na troca de mensagens. Resultados mostram uma redução de até 92% no número de mensagens no protocolo de renovação de certificados comparados com alguns métodos existentes.

Palavras-chave: Segurança em sistemas computacionais, Confiabilidade de Redes de Computadores, Criptografia de Chave Pública, Rede Local Sem Fio.

Abstract

In this paper we propose a PKI architecture with high availability and resilience to mitigate the Black Hole and Spoofing attacks against ad hoc networks. The architecture is designed to reduce the amount of control messages exchanged, increasing its scalability and performance. In order to reach this goal, we propose that the various nodes within an ad-hoc network use a distributed digital certification service to authenticate and encrypt their messages. Particularly the network protocols must use the authentication and cryptography to protect the network from the above-mentioned attacks, granting confidentiality, authenticity and integrity when exchanging messages. Results show a decrease of up to 92% in the certificate renew protocol messages number when compared to existing approaches.

Keywords: Computer system security, Computer network reliability, Public key cryptography, Wireless LAN.

Abreviações

AP	:	Access Point
RF	:	Rádio-Freqüência
WEP	:	Wired Equivalent Privacy
WPA	:	Wi-Fi Protected Access
OSA	:	Open System Authentication
SKA	:	Shared Key Authentication
SSID	:	Service Set Identifier
OSI	:	Open System Interconnection
CNAC	:	Closed Network Access Control
IP	:	Internet Protocol
WLAN	:	Wireless Local Area Network
WI-FI	:	Wireless Fidelity
PDA	:	Personal Data Assistants
CDMA	:	Code Division Multiple Access
GSM	:	Groupe Spécial Mobile
AES	:	Advanced Encryption Standard
ASN.1	:	Abstract Syntax Notation 1
CBR	:	Constant Bit Rate
DER	:	Digital Encoded Representation
DES	:	Data Encryption Standard
3DES	:	Triple DES
DSDV	:	Destination-Sequenced Distance Vector
DSR	:	Dynamic Source Routing
MAC	:	Media Access Control
MD-5	:	Message Digest 5
SHA	:	Secure Hash Algorithm
ICP	:	Infra-Estrutura de Chave Pública

CRL : Lista de Certificados Revogados

AC : Autoridade Certificadora

AR : Autoridade Registradora

Sumário

Lista de Figuras **xiii**

Lista de Tabelas **xiv**

1 Introdução **1**

1.1 Classificação das Redes Sem Fio 2

1.1.1 Redes Pessoais 2

1.1.2 Redes Locais sem Fio 3

1.1.3 Redes Metropolitanas sem Fio 4

1.2 Histórico 6

1.2.1 One Time Password - OTP 7

1.2.2 RC4 7

1.2.3 WEP 8

1.2.4 Padrão IEEE 802.1x (Controle de Acesso Baseado em Porta) 9

1.2.5 WPA 10

1.3 Motivação 11

1.4 Tipos de Ataques 11

1.5 Objetivos 16

1.6 Organização do Texto 17

2 Conceitos e Técnicas Criptográficas **19**

2.1 Características da Informação 19

2.2 Criptografia de Chave Simétrica e Assimétrica 20

2.2.1 Algoritmos de Chave Simétrica 20
2.2.2 Algoritmos Assimétricos 22
2.3 Funções Hash 26
2.4 Assinatura Digital 27
2.5 Certificado Digital 29
2.6 Infra-Estrutura de Chave Pública 30
2.7 Limite Criptográfico 31
2.8 Criptografia por ID 32

3 Trabalhos Relacionados **33**
3.1 Perda de Pacotes em Redes ad hoc 33
3.2 Como Compartilhar um Segredo 35
3.3 Um Modelo para Redes Ad Hoc com Limite Criptográfico 36
3.4 Uma Proposta para ICP em Redes ad hoc 37
3.5 Outra Proposta para ICP em Redes ad hoc 38
3.6 Comparação entre os trabalhos 38

4 Proposta **40**
4.1 Arquitetura Proposta 41
4.2 Inicialização das ACs e Distribuição dos Certificados 42
4.3 Listas de Certificados Revogados e de ACs Confiáveis 44
4.4 Funcionamento do protocolo de autenticação 46
4.5 Renovação de certificados 49
4.6 Análise de Complexidade dos Algoritmos 53

5 Resultados **56**
5.1 Análise dos Parâmetros Utilizados no Protocolo 56
5.1.1 Modelo de Erros 56

5.2 Ambiente de simulação 58
5.3 Consumo de espaço em disco 64
5.4 Análise e Comparações 64
5.4.1 Quantidade de Mensagens do Protocolo de Autenticação 67
5.4.2 Quantidade de Mensagens na Renovação de Certificados 69
5.4.3 Consumo de Energia 72

6 Conclusão e Trabalhos Futuros **79**

Apêndice A **82**

Apêndice B **88**

Referências **91**

Lista de Figuras

1.1 Cenário para Wimax. Fonte: Revista Wimax 6

1.2 Funcionamento básico do WEP. Fonte: Wikepedia 9

2.1 Algoritmo DES. 21

2.2 Função de Feistel do Algoritmo DES. 22

2.3 Esquema de utilização de serviços e autenticação usando KDC 26

2.4 Certificado padrão X.509. 29

2.5 Esquema de Infra Estrutura de Chaves Públicas. 31

3.1 Representação para o limite criptográfico com K=2. 35

4.1 Exemplo de Arquitetura. 42

4.2 Sub-camada de segurança. 42

4.3 Estabelecimento de autenticação entre nós na rede 47

4.4 Protocolo de autenticação dos nós. 49

4.5 Protocolo de renovação de certificados. 52

5.1 Variação da probabilidade média de perda de pacotes e a probabilidade de falha em uma rodada. 58

5.2 Taxas de entrega de pacotes. IC é o intervalo de confiança 61

5.3 Autenticações - DSR 62

5.4 Autenticações - DSDV 63

5.5 Lista de ACs Confiáveis e Campos do Certificado Digital. 64

5.6 Probabilidade que nenhuma AC seja contatada depois de cada rodada - Probabilidade de perda de pacotes igual a 32,7%. 70

5.7 Distribuição da Energia gasta para transmitir 64Kbytes - 3DES. 77

Lista de Tabelas

1.1 Padrão IEEE 802.11 - Características 3

1.2 Métodos de autenticação EAP 10

5.1 Número de Mensagens Trocadas em cada Autenticação, p=0,32 69

5.2 Comparação do ganho entre diferentes Kno 69

5.3 Comparação entre os métodos em carga média 72

5.4 Porcentagem de ganho em carga média 72

5.5 Energia gasta em algoritmos de HASH. Fonte[36] 73

5.6 Consumo de Energia em algoritmos assimétricos para uma entrada de 1024 bits. A assinatura RSA é utilizada cifrando e decifrando a mensagem diretamente sem necessidade de gerar o hash da mesma. Usada quando a mensagem é pequena. Fonte[36] 73

5.7 Consumo de Energia em algoritmos simétricos. Fonte[36] 73

5.8 Resumo dos valores de consumo de energia que serão usados para calcular o consumo adicional dos protocolos propostos. 74

5.9 Energia gasta pelo serviço ICPAH para renovação de certificados com $Kno = \lceil N/Log_2 2T \rceil$ e $Kac = 3 \times (T-1)/2$ 76

5.10 Energia gasta por um nó em cada operação - $Kno = \lceil N/Log_2 2T \rceil$ e $Kac = 3 \times (T-1)/2$ 77

5.11 Energia gasta por um nó em cada operação - Kno= N = DIFUSÃO 77

Capítulo 1

Introdução

A utilização de redes móveis tem crescido de maneira acelerada nos últimos anos. Com o rápido crescimento e avanço das tecnologias, principalmente no que tange à mobilidade, tão desejada para quebrar o conceito de que, para estar conectado, deve-se estar à frente de um desktop em uma sala fechada.

Em paralelo ao crescimento da demanda por dispositivos que suportem a mobilidade e de tecnologias associadas, percebe-se, ao longo dos anos, que o principal protocolo que proporciona a conectividade global, o Internet Protocol (IP), se consolidou, e que é o principal ator para a convergência das redes e das redes de próxima geração.

O termo rede sem fio ou rede wireless refere-se a uma rede de computadores ou dispositivos móveis interligada sem fios, isto é, por canais de comunicação como radio-freqüência, infravermelho ou laser. Um tipo de rede sem fio que se tornou muito popular é o Wi-Fi. O termo Wi-Fi vem da abreviação de *Wireless Fidelity* e trata-se de um conjunto de padrões de compatibilidade para "*wireless local area networks*"(WLAN) baseado nas especificações IEEE 802.11 [32].

Pode-se perceber o grande potencial de utilização das redes sem fio como última milha de instalações onde seria muito custoso o lançamento de cabos. Também em construções tombadas pelo patrimônio histórico onde não são possíveis mudanças nos imóveis as redes sem fio são umas das poucas opções para ligação de dispositivos. Este trabalho foca no uso de redes ad hoc ou sem infra-estrutura. Estas redes são amplamente utilizadas em instalações militares, campos de batalhas, campus universitários, laboratórios de pesquisa entre outros.

Porém, existem algumas importantes diferenças entre as redes sem fio e redes com fio. É importante entender bem algumas destas diferenças pois elas são usadas como forma de exploração visando comprometer a segurança das redes sem fio. Os seguintes problemas

inerentes de redes sem fio podem ser citados:

1. Atenuação - A radiação eletromagnética é atenuada quando passa pelo meio de transmissão resultando na diminuição da potência do sinal recebido à medida que a distância entre o emissor e o receptor aumenta;

2. Interferência de outras fontes - Interferência pode ocorrer quando outros dispositivos transmitem na mesma freqüência ou no mesmo slot ou fatia de tempo (FDM e TDM) ou quando ruído eletromagnético é gerado no ambiente (um motor próximo por exemplo);

3. Propagação multi-caminho - Ocorre quando a onda eletromagnética reflete em objetos e na terra gerando caminhos de diferente comprimento de onda entre o emissor e o receptor;

4. Difusão usando o "ether" como meio - Pelo próprio uso do "ether" como meio de transmissão e a inexistência de cabeamento onde trafegam os sinais, a falta de privacidade é inerente às redes sem fio.

Esta dissertação propõe utilização de uma infra-estrutura de chaves públicas com limite criptográfico para proteger as rede ad hoc de ataques do tipo buraco negro e falsificação de identidade. Nas próximas subseções deste capítulo serão apresentados aspectos como a classificação das redes sem fio, histórico sobre segurança em redes sem fio e também alguns tipos de ataques que podem ser desferidos contra redes ad hoc.

1.1 Classificação das Redes Sem Fio

Didaticamente as redes sem fio podem ser divididas de diversas formas. Este trabalho propõe a seguinte classificação baseada no alcance das mesmas: redes pessoais, redes locais sem fio e redes metropolitanas sem fio.

1.1.1 Redes Pessoais

Redes de alcance limitado a poucas dezenas de metros. Muito utilizadas de maneira doméstica e em aparelhos eletro-eletrônicos de uso pessoal.

- Infra Vermelho. Redes bastante usadas, mas são limitadas pelo seu pequeno alcance e baixa velocidade (115 Kbps). Encontram-se principalmente em computadores portáteis, PDA's (Personal Data Assitants), telefones celulares e algumas impressoras.

- Bluetooth [29]. Usado para comunicação entre pequenos dispositivos de uso pessoal, como podem ser os PDAs, celulares de nova geração e alguns computadores portáteis. Opera dentro da banda dos 2,4 GHz. A sua principal desvantagem é a incompatibilidade de dispositivo de comunicação de diferentes fabricantes.

- Redes 802.15 [33]. Têm seus níveis físicos e de enlace baseados no Bluetooth e são consideradas como *Wireless Personal Networks* (WPAN). Estas redes operam também na faixa de 2.4 GHz usando multiplexação por divisão no tempo (TDM) com frequency-hopping spread spectrum (FHSS). Redes 802.15 podem prover velocidade de acesso de até 712 Kbps. Esta rede pode ter até oito dispositivos ativos ao mesmo tempo sendo que um deles é designado "master"e os outros são denominados "slave"ou escravos. Em adição aos nós escravos podem existir até 255 dispositivos "estacionados"ou "parked devices". Estes dispositivos não podem se comunicar até que seu estado seja modificado de "parked"para ativo pelo nó "master".

1.1.2 Redes Locais sem Fio

Existem muitos padrões 802.11 para redes sem fio mas todos têm algumas características em comum. Usam o mesmo protocolo de acesso ao meio - CSMA/CA (*Carrier Sense Multiple Access / Colision Avoidance*), a mesma estrutura de frame, diminuem a taxa de transmissão para alcançarem maiores distâncias e trabalham em modo ad hoc e infra-estruturado. A Tabela 1.1 mostra um sumário das taxas máximas de transmissão e freqüências usadas em alguns padrões 802.11. Nota-se que todos trabalham com uma banda de 85 MHz dividida em 11 canais. Estes canais apresentam sobreposição no espectro de freqüência e apenas os canais 1, 6 e 11 não têm sobreposição quando usados em conjunto.

Padrão	Faixa de freqüência	Taxa de transmissão
802.11b	2.4-2.485 GHz	Até 11 Mbps
802.11a	5.1-5.8 GHz	Até 54 Mbps
802.11g	2.4-2.485 GHz	Até 54 Mbps

Tabela 1.1: Padrão IEEE 802.11 - Características

O componente fundamental da arquitetura é o BSS (*Basic Service Set*). Um BSS contém uma ou mais estações sem fio e uma estação base também conhecida como *Access Point* (AP). Para uma estação poder fazer parte de uma rede sem fio é necessário que ela se associe a um AP (modo infra-estrutura) ou às outras estações (modo ad hoc) usando um identificador chamado SSID (*Service Set Identifier*).

- Topologia ad hoc

  Cada dispositivo pode se comunicar com todos os demais sem a necessidade de um ponto central de sincronização. Cada nó faz parte da rede ponto a ponto para a qual vai ser necessário um identificador - SSID - igual para todos. É importante não ultrapassar um número razoável de dispositivos para não degradar o desempenho da rede. Redes ad hoc têm protocolos de roteamento distintos dentre os quais podemos destacar DSDV (*Destination Sequence Distance Vector*) [51], AODV (*Ad Hoc On-Demand Distance Vector*) [50], OLSR(*Optimized Link State Routing*) [35] e vários outros.

- Topologia com Infra-estrutura

  Nesta topologia existe um nó central através do qual todos os outros nós se comunicam. Não há comunicação direta entre os nós. Este nó central serve também para encaminhar as ligações feitas na rede sem fio para outras redes distintas. Para poder estabelecer a comunicação, todos os nós devem estar dentro da zona de cobertura do nó central ou AP (Access Point).

- Redes Mesh

  As redes Mesh combinam as topologias wireless. Basicamente, são redes com topologia infra-estrutura que se comunicam de modo ad hoc entre si. Normalmente as redes em modo infra-estruturado são ligadas a alguma rede com fio ou roteador de acesso.

1.1.3 Redes Metropolitanas sem Fio

Atualmente observamos uma verdadeira explosão no uso de celulares no mundo. O termo celular vem do fato que a área de cobertura é composta de subáreas chamadas células. Cada célula contém uma estação rádio-base (base station). As tecnologias mais usadas atualmente são CDMA [52](Padrão nos EUA e parte da Ásia) e GSM [6](Padrão na Europa e Ásia). Estas duas tecnologias estão classificadas como segunda geração de

celulares ou 2G. GSM combina FDM (frequency division Multiplex) com TDM e consiste de bandas de freqüência de 200 kHz com cada banda suportando oito chamadas TDM. As conversações são codificadas a 13 kbps e 12.2 kbps no GSM.

Já o CDMA utiliza divisão por código para o acesso ao meio de transmissão. Todos os usuários compartilham a mesma faixa de freqüência porém cada um tem um código gravado no chip do celular. Com o uso de CDMA não é preciso fazer alocação de bandas de guarda de freqüência e também há o fato de que CDMA é mais imune a interferências.

Ainda pouco conhecido no mercado, o padrão wireless 802.16 pode revolucionar a indústria de acesso de banda larga sem fio. Esta tecnologia está sendo especificada pelo grupo do IEEE que trata de acessos de banda larga para última milha em áreas metropolitanas, com padrões de desempenho equivalentes aos dos tradicionais meios tais como DSL, Cable modem ou E1/T1.

O WIMAX Fórum (*Worldwide Interoperability for Microwave Access*) [24] é uma organização sem fins lucrativos, formada por empresas fabricantes de equipamentos e de componentes, e tem por objetivo promover em larga escala a utilização de redes ponto multiponto, operando em freqüências entre 2GHz e 11GHz, alavancando a padronização IEEE 802.16 e garantindo a compatibilidade e interoperabilidade dos equipamentos que adotarem este padrão. O WIMAX Fórum é o equivalente, ao Wi-Fi Alliance, responsável pelo grande desenvolvimento e sucesso do Wi-Fi em todo o mundo. O Wimax pode facilitar a implantação de diversas aplicações de banda larga sem fio conforme a Figura 1.1.

O WIMAX possui taxas de transmissão com longo alcance além de ser também tolerante às reflexões de sinais. A velocidade de transmissão dos dados varia entre 1 Mbps e 75 Mbps, dependendo das condições de propagação, sendo que o raio típico de uma célula WIMAX é de 6 km a 9 Km. O padrão 802.16 apresenta qualidade de serviço que permite a transmissão de voz e vídeo, que requerem redes de baixa latência. Além disso, características de privacidade e criptografia estão previstos no padrão 802.16, permitindo transmissões seguras incluindo os procedimentos de autenticação.

Todas as tecnologias utilizadas pelos diversos consórcios e fabricantes tentam ofertar serviços de maneira segura. Alguns protocolos e padrões de segurança já vêm sendo usados para este fim. Na próxima seção é apresentado um histórico de como os fabricantes têm implantado tais características.

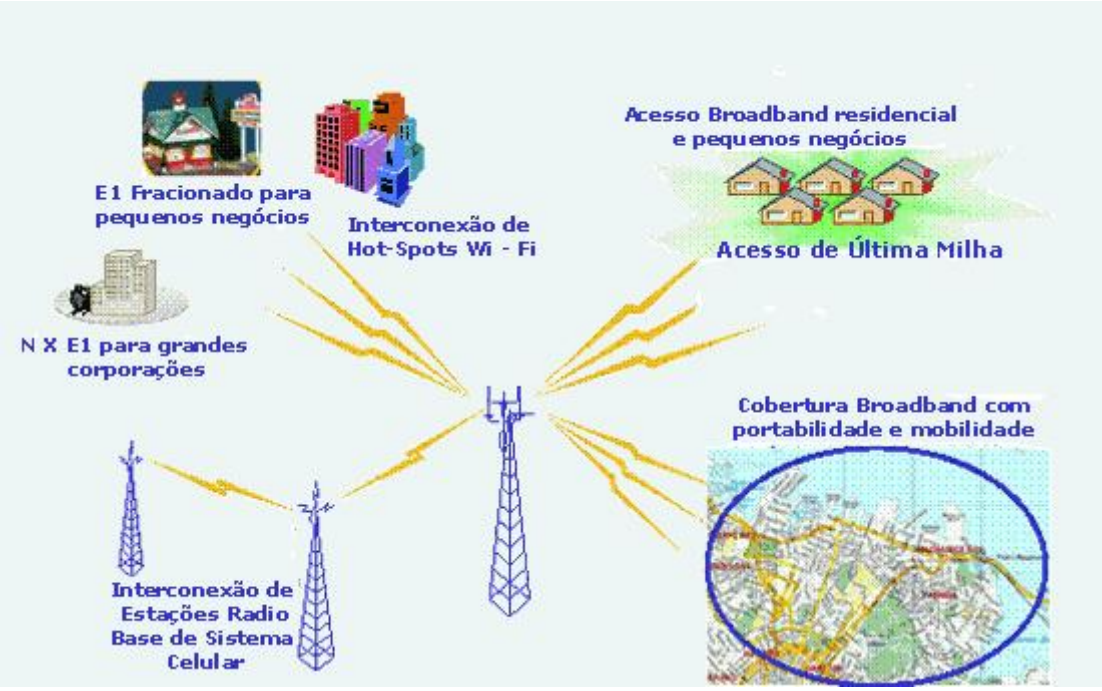


Figura 1.1: Cenário para Wimax. Fonte: Revista Wimax

1.2 Histórico

Esta seção dará ênfase no histórico de técnicas de segurança em redes 802.11, pois este é um dos tipos de rede que pode ser usada em modo ad hoc, o alvo das propostas deste trabalho.

Inicialmente algumas medidas simples podem ser tomadas sem a utilização de técnicas criptográficas. Uma destas medidas é mudar o modo de associação dos dispositivos junto aos AP's desabilitando nos AP's a difusão do SSID. Desta forma os nós devem apresentar o SSID ao AP quando do pedido de associação. Outra medida é a escolha de um nome não trivial para o SSID da rede que funciona como uma senha de autenticação entre os AP's e os nós. Adicionalmente também é possível fazer controle por endereço MAC dos nós e dos AP's que participam da rede, embora seja possível a falsificação dos endereços MAC por um agressor.

Ao longo do tempo várias técnicas têm sido utilizadas, muitas delas em conjunto, para tentar garantir a segurança das redes sem fio. As senhas geradas para serem utilizadas uma única vez (OTP), protocolos de criptografia como o RC4, o WEP e seu sucessor o WPA. Também são utilizados protocolos que fazem o controle de acesso à rede baseado nas portas de conexão físicas como o IEEE 802.1x. Estas técnicas e protocolos vêm ao longo do tempo sendo melhoradas e serão brevemente explicadas nas subseções a seguir.

1.2.1 One Time Password - OTP

A idéia principal no uso de senhas (passwords) que só são utilizadas uma vez é reduzir o risco de sua revelação e o uso de senhas fracas. Abaixo são descritos dois métodos básicos para se gerar "*One Time Passwords*"[30]:

- Implementação usando um algoritmo matemático e função hash

  A partir de uma "semente"aleatória são geradas as senhas utilizando-se uma função hash criptográfica. Maiores detalhes sobre funções hash são apresentados no capítulo 2. Se uma série muito grande de senhas é necessária, então é escolhida uma nova semente aleatória e o processo continua.

- Implementação através de sincronização de tempo

  Geralmente esta implementação está relacionada a tokens e dispositivos físicos que são dados a cada usuário. O token gera uma senha aleatória baseada no relógio interno que é sincronizado com o relógio de um servidor de autenticação. Um exemplo de dispositivos deste tipo são os tokens RSA SecureID.

1.2.2 RC4

O RC4 [38] é um algoritmo de cifragem de fluxo amplamente utilizado em software e em vários protocolos tais como SSL e WEP. Algoritmos de fluxo cifram as mensagens bit a bit e são de chave simétrica (no capítulo 2 será apresentada uma descrição da cifragem com chave simétrica).

Os algoritmos de fluxo utilizam uma chave de tamanho "n"igual ao do fluxo de bits que será cifrado. Então é feito um XOR (OU exclusivo) entre a mensagem em texto claro e a chave. O resultado é enviado e no remetente é feito um XOR entre a mensagem cifrada e a mesma chave gerando a mensagem original. A dificuldade é gerar chaves "aleatórias"com o tamanho do fluxo de bits. O protocolo RC4 é uma maneira de gerar bytes aleatórios a partir de uma chave de tamanho variável. Estes bytes são usados para cifrar uma mensagem da maneira descrita neste parágrafo (através do XOR). O destinatário executa o RC4 na mesma chave do remetente, obtendo assim os mesmos bytes aleatórios, podendo desta forma decifrar a mensagem.

A vantagem do RC4 é que ele gera seqüências pseudo-aleatórias fortes com mais rapidez quando comparado com outros algoritmos de fluxo, já que o sistema funciona

basicamente por permutações e somas de valores inteiros, o que torna este algoritmo muito simples e rápido.

1.2.3 WEP

Acrônimo para *Wired Equivalent Privacy*. Foi introduzido para tentar dar segurança na autenticação, proteção e confiabilidade na comunicação entre os dispositivos sem fio. Exemplos: WEP64 (40 bits reais) e WEP128 (104 bits reais). É inseguro devido a sua arquitetura. Um problema conhecido na implementação do vetor de bits de inicialização que é usado para gerar as chaves de criptografia do WEP tornou este protocolo passível de ser quebrado.

O algoritmo WEP foi o primeiro padrão de segurança proposto para comunicações wireless LAN. Trata-se de um algoritmo simples, que utiliza um gerador de número pseudo-randômico (PRNG) e a cifragem RC4. A segurança fornecida pelo algoritmo WEP reside na dificuldade de se descobrir uma chave secreta. Isso é relacionado com o tamanho da chave secreta e com a freqüência da mudança de chaves. A criptografia é feita através de uma operação "ou exclusivo (XOR)", com uma máscara pré-determinada. O usuário configura os rádios com uma chave de, por exemplo, 40 bits. Os rádios criam uma máscara composta dos 40 bits configurados, mais 24 bits (vetor de inicialização) gerados randomicamente. O WEP é um algoritmo simétrico, no qual a mesma chave é usada para cifrar e decifrar. O vetor de inicialização é usado para que a chave criptográfica não se repita com freqüência no momento da transmissão. Assim, ele deve ser transmitido no frame, de forma clara e não-cifrada, permitindo que o receptor possa uni-lo à sua chave secreta a fim de compor a chave criptográfica e decifrar a mensagem. O Vetor de Inicialização (IV) é então concatenado com a chave secreta, resultando na chave criptográfica propriamente dita. A saída do processo de criptografia é a mensagem contendo o endereço MAC de destino e o IV, ambos não-cifrados, mais o texto cifrado.

O IV estende a vida de uma chave secreta uma vez que esta permanece constante, enquanto o IV muda periodicamente. Cada novo resultado do IV gera uma nova chave Quando o IV e a mensagem cifrada são transmitidos, uma estação que estiver escutando o tráfego poderá determinar partes da seqüência-chave gerada pelo par "chave secreta/IV". Se o mesmo par "chave secreta/IV"for transmitido repetidamente a cada dado, o efeito da privacidade do WEP será reduzido, permitindo a um atacante recuperar uma série de dados do usuário sem qualquer conhecimento da chave secreta. Assim, a mudança de IV a cada dado a ser transmitido é um método simples de preservar a eficiência do WEP.

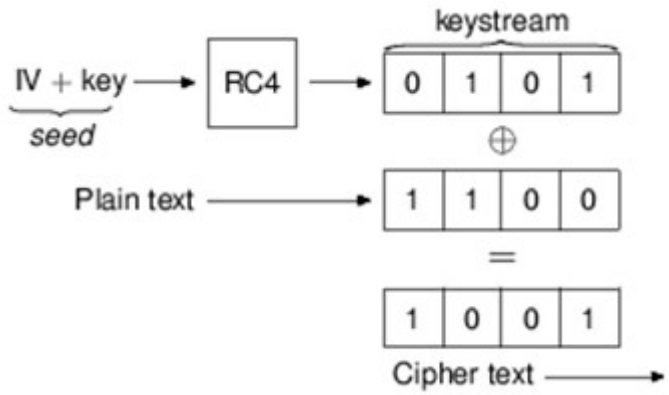


Figura 1.2: Funcionamento básico do WEP. Fonte: Wikepedia

Porém o WEP permite que o IV seja reutilizado (em média, a cada cinco horas). Esse recurso facilita muito um ataque, pois a repetição do IV garante que o invasor terá algum texto codificado repetido para analisar. Além disso, o WEP não oferece nenhuma maneira de mudar as chaves automaticamente. Como resultado, a única forma de reatribuir chaves ao AP e às estações é manualmente; portanto, por uma questão prática, ninguém muda as chaves, expondo assim a rede a ataques passivos que coletam o tráfego e violam as chaves.

1.2.4 Padrão IEEE 802.1x (Controle de Acesso Baseado em Porta)

O 802.1x é o padrão do IEEE para controle de acesso à rede baseado em portas. Usado em conjunto com os métodos EAP (Extensible Authentication Protocol) [3] para fornecer controle de acesso a redes com e sem fios que pode ser configurado para exigir autenticação mútua entre o cliente e a rede. Se não houver autenticação, as comunicações não são permitidas. O 802.1x trabalha com o EAP para autenticar o cliente com a rede e a rede com o cliente, garantindo que ambos os lados se comuniquem com entidades reconhecidas.

Um cliente faz uma conexão inicial para um autenticador (um switch de rede ou um ponto de acesso sem fio). O autenticador é configurado para exigir a utilização do 802.1x de todos os clientes e irá ignorar qualquer conexão de entrada que não se adequar. O autenticador solicita ao cliente sua identidade, a qual ele passará adiante para o "authentication server"(normalmente um servidor RADIUS). O RADIUS [54] segue qualquer mecanismo necessário para autenticar o cliente que está se conectando. Em geral, isto envolve a instalação de uma comunicação EAP entre o cliente e o servidor de autenticação (o autenticador é apenas um dispositivo de passagem) além do estabelecimento

de um método de autenticação dentro da comunicação EAP. Os métodos de comunicação EAP são mostrados na tabela abaixo. Assim que o RADIUS tenha autenticado o cliente, o cliente pode se comunicar com a rede passando pelo autenticador (switch LAN ou o ponto de acesso sem fio). O tráfego dos clientes autenticados passa pela porta controlada, que bloqueia o tráfego de clientes não autenticados. Durante o processo de autenticação o autenticador deve se comunicar com o servidor RADIUS, o que ocorre através da porta não controlada. Após a autenticação de um cliente, a porta controlada passa para o estado "conectado"para este determinado cliente. A identificação do cliente normalmente é feita pelo endereço MAC do mesmo.

1	Identity
2	Notification
3	Nak (somente resposta)
4	MD5-Challenge
5	OTP (RFC1938)
6	Generic Token Card
7	TLS (RFC 2716)

Tabela 1.2: Métodos de autenticação EAP

1.2.5 WPA

O *Wi-fi Protected Access* é desenhado para uso com um servidor de autenticação padrão IEEE 802.1X, o qual distribui chaves diferentes de criptografia para cada usuário. Entretanto o WPA pode ser usado no modo pre-shared key, onde a mesma passphrase ou "frase-password"é dada para todos os usuários da rede. O WPA foi criado pela Wi-Fi Aliance e este consórcio já antecipou o lançamento do WPA2 baseado no documento final do IEEE 802.11i.

Os dados são cifrados usando o algoritmo RC4 com chave de 128 bits e um vetor de inicialização. Adicionalmente o WPA define o uso do AES (Advanced Encription Standard)[12], como uma substituição opcional para criptografia WEP. Um dos grandes melhoramentos do WPA sobre o WEP é o uso do protocolo TKIP (*Temporal Key Integrity Protocol*). O TKIP dinamicamente troca as chaves dos usuários de tempos em tempos. Quando combinado com um vetor de inicialização maior, o TKIP torna o WPA imune ao ataque de key-recovery ao qual o WEP é vulnerável.

1.3 Motivação

A segurança é um dos maiores desafios das redes sem fio. Desde seu surgimento, os fabricantes vêm tentando disponibilizar protocolos que garantam a integridade, disponibilidade e confidencialidade das comunicações de maneira que não diminua o desempenho e nem a facilidade de uso. A questão da segurança deve ser muito bem analisada quando se utiliza um sistema em rede com vários usuários. Logicamente, como se tratam de tecnologias que possuem características próprias e/ou únicas, cada uma delas tem seus prós e contras. O grande desafio é saber quais são estes prós e contras e a que nível esses pontos podem afetar a transmissão e recepção dos dados.

Uma rede sem fio transmite seus dados através de várias técnicas. Duas destas técnicas são a transmissão por meio de rádio freqüência (RF) ou por pulsos de infravermelho. A faixa de freqüência de transmissão dos dados por RF dos padrões mais usados (IEEE 802.11 B/G), fica entre 2.4 - 2.4835 GHz, mas o sinal dessa transmissão via rádio pode ser interceptado por receptores colocados fora da área física da rede ou na direção em que as ondas se propagam, o que colocaria em risco uma informação. A transmissão por pulsos de infravermelho opera nas faixas de 300 - 428,000 GHz. Esta faixa é diferente da faixa das redes de RF mais utilizadas, mas o sinal é afetado pela luz do sol e por obstáculos, dificultando a transmissão.

Conforme aumentam o uso e a importância das redes sem fio, aumentam também os ataques contra este tipo de solução. Os ataques de negação de serviços tipo buraco negro e falsificação de identidade estão entre as maiores ameaças à segurança de redes sem fio e sem infra-estrutura fixa. Tais ataques ocorrem também em redes com fio, porém a arquitetura sem fio e sem infra-estrutura torna tais ataques mais difíceis de serem detectados e de serem contidos.

1.4 Tipos de Ataques

Nesta seção são apresentados alguns tipos de ataques que podem ser desferidos contra redes ad hoc e que podem ter seus efeitos mitigados utilizando-se as técnicas e propostas que serão mostradas neste trabalho.

Os ataques a redes ad hoc podem ser divididos em dois tipos: ativos e passivos. Os ataques passivos não afetam a operação da rede, sendo caracterizados pela espionagem dos dados sem alterá-los. Já os ataques ativos são aqueles onde o atacante cria, altera, descarta

ou inviabiliza o uso de dados em trânsito. Os ataques ativos são os que apresentam o maior número de tipos e podem ocorrer em vários níveis da arquitetura de redes.

Um dos principais ataques passivos é a espionagem ou *eavesdropping*. Neste ataque o atacante aproveita-se da falta de segurança para espionar, coletar e roubar informações. Normalmente um ataque de espionagem é seguido por um ataque ativo aproveitando-se das informações coletadas. A forma de proteção contra ataques de espionagem normalmente é feita pela camada de aplicação utilizando-se cifragem fim a fim. Também alguns algoritmos de roteamento (OSPF[48], RIPv2 [45], OLSR[35]) se preocupam em proteger suas informações de rotas e evitam que um atacante possa descobrir a topologia da rede.

Os ataques ativos em sua maioria têm como alvo alguma vulnerabilidade de alguma camada no modelo de comunicação. Abaixo serão descritos alguns ataques ativos amplamente conhecidos tais como o buraco negro, o de falsificação de identidade, o ataque Sybil e o Jelly-Fish.

1. O Ataque *Buraco Negro*

   Este tipo de ataque se caracteriza quando um ou vários nós da rede deliberadamente descartam os pacotes que passam por eles após o estabelecimento de rota. O buraco negro pode também funcionar como um espião, apenas copiando os pacotes para uma base de dados interna, sem descartá-los, para posterior análise.

   É importante ressaltar que o mau comportamento do nó atacante só começa após o estabelecimento das rotas. O nó atacante participa normalmente do protocolo de estabelecimento de rotas. Por esta razão, é importante que os nós de uma rede tenham confiança e autenticação entre si antes de estabelecerem suas rotas. Para a segunda variante do ataque, espionagem, o uso adicional de cifragem fim a fim dificulta a quebra de confidencialidade.

2. O Ataque de *Falsificação de Identidade*

   Em uma das formas de ataque de falsificação de identidade, também conhecido como *spoofing*, um nó pode se anunciar como sendo o nó atacado para receber as mensagens endereçadas à sua vítima. São conhecidos vários tipos de ataques de falsificação de identidade e eles podem ser lançados nos diversos níveis da arquitetura de comunicação.

   Uma variação do ataque de falsificação de identidade é o *Man-in-the-midle*. No *Man-in-the-midle* o atacante se interpõe na comunicação entre as partes capturando os dados de ambos os lados e se fazendo passar por ambas as partes da comunicação.

Neste caso, dois nós (X e Y) se comunicam sem perceber que existe um terceiro, que se faz passar por Y na comunicação com X e se faz passar por X na comunicação com Y.

Outra variação de *spoofing* é o *Replay-attack* ou *Playback-attack*. Nesta variante o atacante captura algum dado de autenticação de uma comunicação anterior e tenta usá-lo novamente para uma nova autenticação no sistema. Normalmente para defender o sistema deste tipo de ataque é usado um número aleatório gerado na comunicação entre os nós chamado *nonce* que não deve se repetir, dificultando assim o reuso de dados de autenticação já usados.

3. O Ataque *Sybil*

   O Sybil ocorre quando uma AC (Autoridade Certificadora - ver definição no próximo capítulo) não autorizada, por exemplo, consegue gerar certificados para os nós da rede [17]. O ataque *Sybil* se baseia no fato de que é praticamente impossível, em sistemas distribuídos, que nós que não se conhecem apresentem identidades distintas convincentes. Sem a existência de um ponto central para controlar a associação de uma identidade a uma entidade, é sempre possível para uma entidade desconhecida apresentar múltiplas identidades. Assim, o ataque *Sybil* acontece quando um único hardware assume múltiplas identidades em uma rede.

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

   Por fim, uma outra utilização para os nós *Sybil* acontece em redes que utilizam mecanismos de confiança baseados em reputação [62]. Em tais redes, a reputação do nó é dada pela observação de suas ações. Um nó só é considerado malicioso se cometer diversas ações consideradas ruins ou se cometer uma grande ação ruim. Assim, duas estratégias podem ser utilizadas pelo nó atacante. A primeira seria o espalhamento da culpa, na qual o nó *Sybil* utiliza cada uma de suas identidades para fazer pequenas ações ruins, de forma que nenhuma delas possa ser considerada maliciosa. A outra estratégia seria utilizar uma identidade para realizar uma ou mais ações ruins até que ela fosse expulsa, classificada como maliciosa. Quando isso acontecesse o nó geraria uma nova identidade e a usaria para continuar atacando.

Normalmente a defesa contra este ataque consiste em autenticação e uso de chaves distribuídas, por isso a proposta deste trabalho também pode ser usada para contra-atacar o ataque Sybil.

4. O Ataque *Jelly-Fish*

   O Jelly-Fish [1] explora vulnerabilidades do algoritmo de controle de congestionamento do TCP e age de três maneiras: desordenando pacotes, fazendo um descarte periódico de pacotes em um determinado tempo e em sua terceira forma atrasando randomicamente os pacotes. O objetivo deste ataque é reduzir a vazão recebida (*goodput*) de todos os fluxos TCP para próximo de zero.

   Um nó malicioso *Jelly-Fish* usa um buffer de reordenamento ao invés de entregar os pacotes de maneira FIFO. Desta forma, se um nó *Jelly-Fish* estiver no meio de um fluxo TCP, a origem do fluxo começará a receber ACK's duplicados do nó destino por conta da chegada fora de ordem dos pacotes. Recebimento de ACK's duplicados pode levar o nó de origem a inferir que existe perda no fluxo TCP. Assim o nó de origem vai diminuindo a taxa de envio de pacotes para poder se ajustar. Este decréscimo pode levar a vazão recebida do fluxo a valores próximos de zero.

   Para melhor entender a terceira modalidade do *Jelly-Fish* é preciso uma pequena explicação sobre o modelo de controle de fluxo do TCP [22]. Fundamental para o mecanismo de *timeout* e retransmissão é a estimativa do tempo total de transmissão de ida e volta (RTT - *round trip time*) em uma determinada conexão. Como esta estimativa muda com o tempo, devido a mudanças de rotas e padrões de tráfego, o TCP deve monitorar estas mudanças e modificar o tempo de timeout apropriadamente.

   Faz-se então necessário monitorar a variância nas medidas do RTT, em adição ao mecanismo que faz as estimativas atenuadas. Calculando o RTO (*Round Trip Timeout* ou *timeout* de retransmissão) baseado tanto na média, quanto na variância, obtém-se uma resposta muito melhor às altas flutuações no RTT, do que calcular o RTO como somente uma constante vezes a média. Como descrito por Jacobson [34], o desvio médio é uma boa aproximação do desvio padrão, porém mais fácil de calcular. Isto leva a seguinte série de equações que são aplicadas a cada nova medida do RTT:

$$
\begin{aligned}
Err &= RTT - A \\
A &= A + g \times Err \\
D &= D + h(|Err| - D) \\
RTO &= A + 4D
\end{aligned}
$$

Onde A é o RTT atenuado (uma estimativa da média) e D é o desvio padrão atenuado. Err é a diferença entre o último valor medido e a estimativa corrente. Tanto A como D são usados para calcular o próximo valor do *timeout* de retransmissão (RTO). O ganho g é igual a 1/8 (0,125). O ganho para o desvio é igual a h e tem valor 0,25. Quanto maior o ganho para o desvio, maior será o RTO em resposta às mudanças do RTT. As variáveis A e D são inicializadas com os valores 0 e 3 respectivamente.

Na modalidade de descarte de pacotes, o *Jelly-Fish* descarta pacotes uma vez a cada RTO (Round Trip Timeout). No primeiro evento de perda causada pelo *Jelly-Fish* o fluxo sofre *timeout* pois o descarte é suficientemente longo para gerar múltiplas perdas. Quando o fluxo for sair do *timeout* após "RTO"segundos, o Jelly-Fish começa o descarte de novo. Como o *Jelly-Fish* é que induz o *timeout*, o nó atacante sabe que "RTO"segundos depois o fluxo tentará sair da condição de *timeout* e então começa novo descarte.

Finalmente na modalidade de atraso randômico, o nó atacante serve o fluxo TCP de maneira FIFO porém espera um tempo randômico antes de tratar e despachar o pacote do fluxo. Altas variações de atraso no fluxo podem:

- Fazer o TCP enviar tráfego em rajadas, levando ao aumento de colisões e perdas;
- Causar má estimação da banda disponível para o controle de congestionamento baseado em atraso (TCP Vegas por exemplo);
- Levar a um valor excessivamente alto do RTO;

Além destes, vários outros ataques contra redes com fio podem ser observados em uma rede ad hoc. Como exemplo, podemos citar o *Denial of Service* (DOS) e *Distributed Denial of Service* (DDOS). DOS e DDOS são ataques que visam atingir a disponibilidade de algum nó ou serviço fazendo que o mesmo não consiga responder a requisições. Normalmente visam sobrecarregar o nó atacado com requisições com o objetivo de parar seus

serviços. O DDOS é lançado a partir de várias máquinas que são denominadas *zumbis*, as quais são coordenadas pelo atacante para desferir o ataque.

1.5 Objetivos

Neste ambiente hostil e com proteção fraca, é altamente recomendado que a arquitetura de segurança da rede seja distribuída. A introdução de qualquer ponto central de controle é uma vulnerabilidade pronta para ser explorada. Se o ponto central é comprometido, toda a rede é comprometida.

Devido à mobilidade dentro de uma rede ad hoc, a entrada e saída de nós pode ocorrer de maneira rápida e por isso nenhuma solução de segurança que possua uma configuração estática pode ser efetiva neste ambiente.

Por outro lado, uma rede ad hoc possui vários nós possivelmente no alcance uns dos outros e por isso há potencialmente vários caminhos ou rotas entre dois nós. Esta característica pode e deve ser aproveitada para aumentar tanto o desempenho da rede usando seus múltiplos caminhos [61] quanto a sua segurança. Qualquer solução de segurança deve ser escalável para suportar redes pequenas e grandes, densas ou esparsas. Esta solução deve também se aproveitar da característica de diversos caminhos e nós na rede para prover uma solução de alta disponibilidade e robusta a ataques.

Serviços de infra-estrutura de chaves públicas (ICP) têm sido utilizados para garantir, em diversos níveis, confidencialidade, integridade e também garantia de não repúdio em diversas aplicações. O capítulo 2 descreve de maneira mais detalhada, porém sucinta, uma infra-estrutura de chaves públicas. Desta forma este trabalho propõe a utilização de um serviço de ICP conforme recomendado em [63] para autenticar e cifrar, quando necessário, as mensagens dos protocolos de roteamento de uma rede ad hoc. O trabalho se propõe a melhorar as propostas em [59], [40] e [44] utilizando cifragem por limiar para dividir a chave privada do serviço de ICP entre diversas autoridades certificadoras (AC) em um esquema de certificação cruzada. Baseado no trabalho de [63], é proposto um protocolo que diminui o número de mensagens trocadas entre as ACs e entre os nós e as ACs.

Assim sendo, este trabalho tem como objetivos principais:

- Proposição de protocolos distribuídos para funcionarem em redes ad hoc que minimizem o número de mensagens trocadas entre os nós no estabelecimento seguro de

rotas e nos processos de renovação de certificados digitais;

- Demonstrar as estruturas de controle adicionais para o funcionamento dos protocolos;

- Mostrar um estudo de funções baseadas em modelos de erros e perdas de pacotes em redes ad hoc para tornar os protocolos propostos eficientes;

- Mostrar os campos mínimos necessários num certificado digital que devem ser usados para o funcionamento dos protocolos e

- Com os protocolos propostos diminuir o gasto de energia de nós em redes ad hoc, uma vez que menos mensagens de controle, para uso da Infra Estrutura de Chaves Públicas, são trocadas.

Com a utilização de técnicas que garantem autenticidade e confidencialidade pode-se evitar que nós maliciosos participem das comunicações, mitigando os efeitos dos ataques de buraco negro e de falsificação de identidade.

A seguir é feita uma breve explicação sobre a organização do texto para melhor contextualizar o problema, os trabalhos relacionados mais importantes, a proposta, resultados e conclusões.

1.6 Organização do Texto

O capítulo 2 descreve as principais técnicas e algoritmos criptográficos usados na proposta para um melhor entendimento do leitor.

No capítulo 3 são apresentados os trabalhos relacionados mais relevantes nos quais o trabalho é baseado para propor melhoramentos. São apresentados trabalhos que baseiam a comunicação do protocolo de segurança em técnicas de difusão. Apesar de ser de fácil implementação a difusão é por vezes custosa quando consideramos o número de mensagens enviadas com a quantidade ideal de mensagens que deveria ser enviada. Alguns trabalhos que utilizam limite criptográfico para implementar uma infra-estrutura mais robusta de ICP (Infra-Estrutura de Chaves Públicas) são mostrados e comentados neste capítulo. Este trabalho também utiliza ICP com limite criptográfico porém faz algumas adaptações no sentido de diminuir o número de mensagens trocadas pelos protocolos de cifragem.

O capítulo 4 descreve a proposta do trabalho com os algoritmos, cenário de implementação e protocolos. Este capítulo mostra em detalhes os algoritmos propostos e faz análises dos possíveis cenários de implementação levando em conta o tamanho da rede, a distribuição dos nós dentro da mesma e os protocolos de roteamento usados.

O capítulo 5 demonstra os resultados com análise matemática em comparação com outros trabalhos, custo de mensagens, consumo de energia e ambiente de simulação. Também é mostrada a análise de complexidade dos algoritmos descritos na proposta. Os protocolos não foram implementados no simulador utilizado (NS-2) [49], mas foram feitas simulações para validar as análises matemáticas e a arquitetura proposta. Também é mostrado um estudo das funções utilizadas para o envio de mensagens dos protocolos de segurança e a discussão de um modelo de erros e de probabilidades de perda de pacotes em redes sem fio.

Finalmente no capítulo 6 são apresentados as conclusões e os trabalhos futuros que podem ser desenvolvidos no sentido de complementar e cobrir os pontos que ficaram pendentes nesta proposta.

Capítulo 2

Conceitos e Técnicas Criptográficas

Neste capítulo são apresentados brevemente alguns conceitos e técnicas utilizadas no escopo da proposta. O objetivo é introduzir o leitor no conjunto de técnicas conhecidas e largamente utilizadas em diferentes níveis para prover segurança da informação em redes de computadores.

2.1 Características da Informação

Antes de discutir os principais conceitos e técnicas criptográficas é preciso definir alguns termos usados em segurança da informação. Estes termos na verdade designam propriedades desejáveis que a informação deve possuir em todo seu ciclo de vida (criação, armazenamento, manuseio e descarte) e também processos pelos quais se consegue alcançar e manter tais propriedades. Estas propriedades estão listadas em [2] e são as seguintes:

- Confidencialidade. Propriedade na qual os dados são revelados apenas a pessoas autorizadas.
- Integridade. Propriedade de uma informação que não foi adulterada ou destruída por alguém não autorizado durante todo o seu ciclo de vida (criação, uso, armazenamento e descarte).
- Disponibilidade. Condição que a informação deve possuir para as pessoas autorizadas a qualquer tempo. A informação deve estar acessível sempre que necessário.
- Autenticidade. Propriedade de uma informação autêntica, isto é, foi gerada por agente competente. É verdadeira.

- Autenticação. Processo de confirmação da identidade de um indivíduo ou organização, ou da comprovação da posse ou integridade de certas informações.
- Incontestabilidade ou Não repúdio. Garantia de que o conteúdo da informação é verdadeiro e foi criado por quem de direito. Não há possibilidade de ser contestada.

Uma das formas mais conhecidas para garantir segurança da informação é a utilização de algoritmos criptográficos, funções *hash*, assinaturas digitais, certificados digitais, infraestruturas de chave publica (ICP), técnicas criptográficas que utilizam limite criptográfico e também criptografia por ID ou identificação. Neste sentido torna-se necessário uma pequena introdução sobre estas técnicas nas seções subseqüentes.

2.2 Criptografia de Chave Simétrica e Assimétrica

2.2.1 Algoritmos de Chave Simétrica

Quando algoritmos de chaves simétricas são usados, ambas as partes interessadas na comunicação usam a mesma chave de criptografia para cifrar e decifrar. Para prover confidencialidade, esta chave necessita ser mantida secreta. Uma vez que alguém mais consiga descobrir a chave, a mesma não é mais segura. Um dos grandes problemas deste tipo de abordagem é a geração de forma segura das chaves e também a sua distribuição. Já que a chave simétrica é um segredo, como distribuir o segredo para o parceiro da comunicação? Se existe um meio seguro de se distribuir a chave, provavelmente este mesmo meio pode ser usado para passar a informação, não necessitando assim de chave de criptografia. Além disso se existirem vários participantes na comunicação, pode ser necessário gerar um par de chaves para cada par de participantes. Seja o número de participantes igual a N, o número de chaves que cada participante deve gerenciar é igual a N-1. A gerência deste número elevado de chaves é um problema que deve ser resolvido quando se usam algoritmos simétricos.

Algoritmos simétricos têm a vantagem de não exigir muito poder de computação. Isso ocorre porque os algoritmos simétricos trabalham com deslocamentos e permutas sobre blocos de dados que serão cifrados usando chaves de 56 a 256 bits. Por isso também estes algoritmos são chamados de algoritmos de blocos. Na Figura 2.1 o diagrama de fluxo de um dos algoritmos simétricos mais conhecidos, o DES [19].

O algoritmo cifra blocos de 64 bits com uma chave de 56 bits (na verdade existem mais oito de paridade dando um total de 64 bits) e uma função de embaralhamento chamada

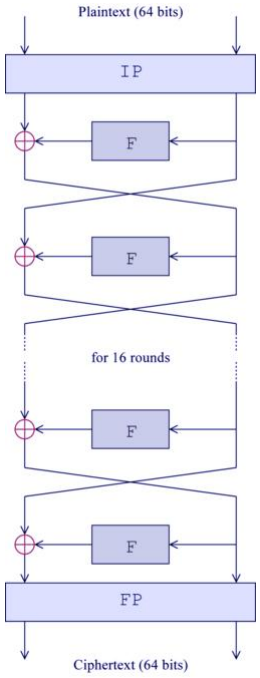


Figura 2.1: Algoritmo DES.

função Feistel. É feita uma permutação inicial no bloco de 64 bits e depois o bloco é dividido em duas metades. Após a divisão, o seguinte procedimento é repetido dezesseis vezes:

1. Uma metade do bloco serve de entrada para a função de embaralhamento (função de Feistel) juntamente com uma sub-chave de 48 bits derivada da chave principal de 56 bits. Esta mesma metade é enviada para a próxima rodada e será entrada para o XOR juntamente com a saída da função de Feistel. Existe um total de 16 sub-chaves, uma para cada rodada;

2. A função de Feistel expande os 32 bits de entrada para 48 bits e faz um XOR com a sub-chave;

3. Os 48 bits de saída do XOR são separados em oito conjuntos de seis bits;

4. É feita então uma transformação não linear que transforma os seis bits de cada conjunto em quatro bits de saída para cada conjunto;

5. Finalmente os 32 bits de saída do passo anterior sofrem uma permutação e este é o resultado de saída da função;

6. É feito um XOR entre a saída da função de Feistel e a outra metade do bloco de entrada do passo 1.

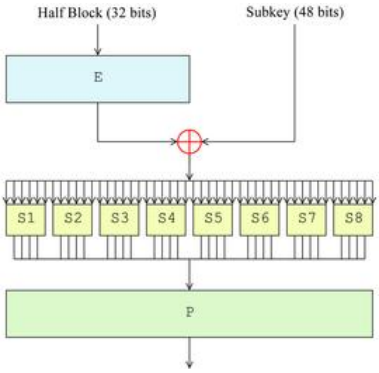


Figura 2.2: Função de Feistel do Algoritmo DES.

Após as dezesseis rodadas é feita uma permutação final no bloco de 64 bits gerando o texto cifrado.

Outros algoritmos simétricos conhecidos são: Triple-DES (3DES)[47], IDEA [46], CAST5 [4], BLOWFISH [57] e TWOFISH [58], AES [12].

2.2.2 Algoritmos Assimétricos

Algoritmos assimétricos também são conhecidos como algoritmos de chave pública. Isso porque usam pares de chaves. Uma chave é conhecida somente pelo dono do par de chaves e a outra é conhecida por todos os participantes da comunicação. A chave que somente o dono tem conhecimento deve ser guardada de forma segura e é chamada de "chave privada"ou "chave secreta", enquanto que a outra chave que é enviada para todos os participantes da comunicação é chamada de "chave pública". Este par de chaves pode ser usado de duas formas:

- Qualquer um que tenha a chave pública de alguém pode enviar mensagens cifradas para o dono da chave secreta pertencente ao par. A chave secreta irá decifrar a mensagem. A chave secreta não pode ser reconstruída a partir da chave pública e vice-versa;
- O dono da chave secreta pode cifrar uma mensagem com esta chave e quem receber a mensagem terá certeza da autenticidade do emissor pois só poderá ler a mensagem usando a chave pública do mesmo.

Com os algoritmos assimétricos, a chave secreta não deve ser compartilhada e o risco da mesma ser revelada é muito menor quando comparado com o uso de algoritmos simétricos. Qualquer usuário só precisa guardar uma chave secreta em segurança e

ter a coleção de chaves públicas dos pares com os quais irá trocar informações. As chaves públicas devem ser protegidas contra a ameaça de serem trocadas por um atacante malicioso que desta forma pode se passar pelo dono da chave. Ou seja, os algoritmos assimétricos resolvem o problema de distribuição de chave que os algoritmos simétricos possuem, mas trazem o problema de publicar e proteger as chaves públicas.

Por outro lado, algoritmos de chave assimétrica são computacionalmente custosos pois trabalham com chaves de 1024 a 4096 bits utilizando operações exponenciais com números primos muito grandes (tipicamente com mais de 100 algarismos decimais). Alguns algoritmos assimétricos conhecidos são RSA [56], DSA [41] e ELGAMAL [20]. A idéia sobre algoritmos assimétricos foi publicada pela primeira vez em 1976 por Diffie e Hellman. A seguir é mostrado o uso de cifragem e decifragem usando RSA.

Em primeiro lugar deve-se escolher dois números primos aleatórios p e q. Então se faz:

n=(pq)

φ =(p-1)(q-1)

Escolhe-se agora um número aleatório $e > 1$ tal que mdc(e,φ)=1 i.e. *"e"* e φ devem ser primos entre si. Calcula-se o número d, sendo $1 < d < \varphi$ tal que:

$ed \bmod \varphi = 1$

A chave pública é formada por [n,e] e a chave privada é o par [n,d]. Para ter uma boa idéia do funcionamento é mostrado um exemplo:

Seja um vetor formado pelos números (123,659,971).

Para facilitar, serão usados números primos pequenos. Toma-se p=17 e q=59. Tem-se então que n=$p.q = 17 \times 59 = 1.003$ e que $\varphi = (p-1) \times (q-1) = 16 \times 58 = 928$. Escolhe-se e=637 e assim d vale 405 pois $(637 \times 405) \bmod 928 = 1$. Na prática pode-se sempre escolher e como sendo um valor fixo, pequeno e primo entre si com φ. Um valor muito comum é e=(2^{16} +1). Isso não interfere em nada com a segurança do RSA e faz com que a cifragem se torne mais rápida. Sendo assim, a chave pública será [1003,637] e a chave privada é o par [1003,405].

Para cifrar uma mensagem m faz-se: c=$m^e \bmod n$

Para decifrar faz-se: m=$c^d \bmod n$. Isso porque

$$m^{e^d} \; mod \; n = m^{ed} \; mod \; n \tag{2.1}$$

Além disso, é preciso usar um resultado da teoria dos números que diz o seguinte:

Se p e q são primos e n=pq, então x^y mod n é o mesmo que $x^{(y \; mod \; (p-1)(q-1))}$ mod n [37]. Com este resultado tem-se que:

$$m^{e^d} \; mod \; n = m^{(ed \; mod \; (p-1)(q-1))} \; mod \; n \tag{2.2}$$

Porém, *e* e *d* foram escolhidos de forma que *ed*-1 é divisível por (p-1)(q-1) o que é equivalente a dizer que *ed* é divisível por (p-1)(q-1) com um resto igual a 1 . Assim, $ed \; mod \; (p-1)(q-1) = 1$. A conclusão é que:

$$m^{e^d} \; mod \; n = m^1 \; mod \; n = m \tag{2.3}$$

$$m^{e^d} \; mod \; n = m \tag{2.4}$$

Assim, conclui-se que:

$$m^{e^d} \; mod \; n = m = m^{d^e} \; mod \; n \tag{2.5}$$

Continuando com o exemplo, a cifragem do vetor ocorrerá conforme mostrado abaixo:

$$\begin{aligned} C1 &= 123^{637} \; mod 1003 = 956 \\ C2 &= 659^{637} \; mod 1003 = 183 \\ C3 &= 971^{637} \; mod \; 1003 = 389 \end{aligned}$$

O vetor cifrado será (956,183,389). Para decifrá-lo faz-se:

$$\begin{aligned} W1 &= 956^{405} \; mod \; 1003 = 123 \\ W2 &= 183^{405} \; mod \; 1003 = 659 \\ W3 &= 389^{405} \; mod \; 1003 = 971 \end{aligned}$$

Algoritmos assimétricos vieram resolver o problema de compartilhamento de chaves que existia com os algoritmos simétricos. No mundo real é interessante usar as duas técnicas em conjunto tirando o que há de melhor em ambas. Embora os algoritmos assimétricos sejam muito mais lentos que os simétricos, sua utilização em conjunto é amplamente difundida. Existem algumas formas usadas para troca de chaves simétricas utilizadas atualmente. São elas:

- As chaves assimétricas são usadas para autenticação e depois uma ou mais chaves simétricas são geradas e trocadas usando o canal de comunicação, cifradas com a chave assimétrica. Desta forma as vantagens de ambos algoritmos podem ser usadas. Exemplos típicos deste procedimento são as combinações RSA/IDEA e PGP2 [11] ou DSA/BLOWFISH [57]; ou

- Utilizando o Algoritmo de desafio Diffie e Hellman.

  O acordo de chaves Diffie-Hellman [16] foi inventado em 1976 e foi o primeiro método prático para estabelecer uma chave secreta sobre um canal de comunicação desprotegido.

  Um resumo do protocolo pode ser descrito da seguinte forma:

1. Alice e Bob concordam com um grupo cíclico finito (um conjunto de números da forma g^n para todo n). Escolhem a base g e o expoente n pertencentes a G, sendo g e n públicos.
2. Alice escolhe um número natural randômico a e envia A= $g^a mod\ n$ para Bob.
3. Bob escolhe um número natural randômico b e envia B= $g^b mod\ n$ para Alice.
4. Alice faz $B^a mod\ n$.
5. Bob faz $A^b mod\ n$.

Agora Alice e Bob estão de posse do valor de $g^{ab} mod\ n$ o qual serve como chave secreta pois $g^{b^a} = g^{a^b} = g^{ab}$; ou

- Utilizando um KDC (*Key Distribution Center*)

  Um KDC ou Centro de distribuição de chaves é um servidor que compartilha uma chave simétrica com cada usuário registrado. Sempre que um usuário (Alice) quer trocar mensagens cifradas com outro (Bob), o procedimento descrito na Figura 2.3 é executado.

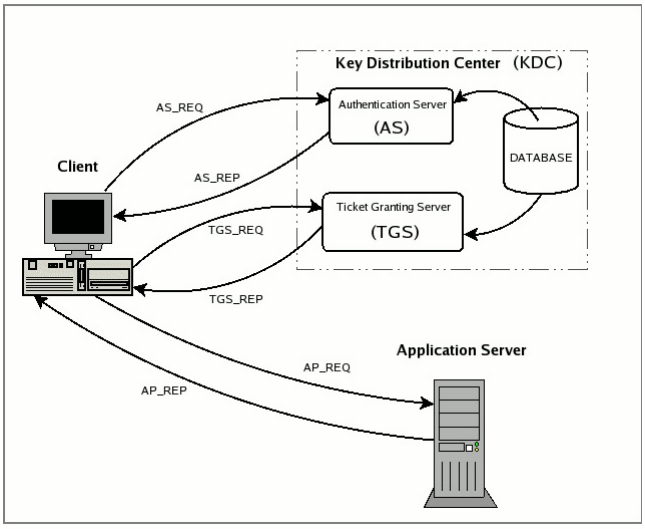


Figura 2.3: Esquema de utilização de serviços e autenticação usando KDC

Um exemplo de um esquema de troca de chaves com criptografia é o SSL [25] (Secure Socket Layer). Quando uma sessão SSL é estabelecida, o servidor começa anunciando sua chave pública para o cliente. Inicialmente não há criptografia e por isso o cliente (e também qualquer intruso) pode ler esta chave. Mas agora o cliente pode enviar mensagens para o servidor de forma que somente o servidor pode decifrá-las. O cliente gera então 46 bytes randomicamente e os transforma num número muito grande de acordo com o formato PKCS1 [42](PKCS são padrões divulgados em 1991 como parte dos acordos que estabeleceram os padrões de Criptografia por Chave Pública. O PKCS1 fala sobre criptografia RSA. A versão 1.5 de 1993 descreve os procedimentos básicos do SSL). Este número será a chave simétrica usada na comunicação. Desta forma o cliente cifra a chave simétrica com a chave pública do servidor e a envia para o mesmo. Somente o servidor pode decifrar a mensagem contendo a chave simétrica e esta chave simétrica será usada para gerar um conjunto de chaves RC4 para cifrar o restante da sessão.

2.3 Funções Hash

Uma função hash "H"é uma transformação que usa uma entrada "m"e retorna um string de tamanho fixo, o qual é chamado de valor de hash "h". Ou seja, h=H(m). Funções hash com apenas esta propriedade têm grande variedade de uso em computação como por exemplo CRC (Cyclic Redundancy Check) e Checksum, mas quando usadas em criptografia, as funções hash devem ter algumas propriedades adicionais.

Os requisitos básicos para uma função hash criptográfica são os seguintes:

- A entrada pode ser de qualquer tamanho.
- A saída tem tamanho fixo. O valor hash, ou saída da função, representa de maneira concisa a mensagem ou documento que serviu de entrada para a função hash. Este valor é chamado de *"message digest"* ou resumo da mensagem. Um *"message digest"* pode ser imaginado como uma impressão digital do documento ou mensagem.
- H(x) é relativamente fácil de computar para qualquer "x". As funções devem ser rápidas.
- H(x) é "one-way"ou não inversível. Uma função hash H é dita não inversível quando é computacionalmente difícil de inverter, onde "difícil de inverter"significa que dado um valor de hash "h", é computacionalmente improvável de achar alguma entrada "x"tal que H(x)=h.
- H(x) é livre de colisões ou possui baixa probabilidade de colisões. Se com uma dada mensagem "x"for computacionalmente difícil achar uma mensagem "y"diferente de "x"tal que H(x)=H(y), então H é dita uma função fracamente livre de colisões. Numa função fortemente livre de colisões é computacionalmente difícil achar quaisquer duas mensagens "x"e "y"tais que H(x)=H(y) [13].

Talvez o papel principal de uma função hash criptográfica seja prover checagem de integridade de mensagens e assinatura digital. Como funções hash são geralmente mais rápidas do que algoritmos de criptografia, é comum computar a assinatura digital de um documento ou mensagem, cifrando apenas o valor hash do documento o qual, normalmente, é muito menor quando comparado com o documento todo. Adicionalmente, um valor de hash pode ser publicado sem revelar o conteúdo do documento com o qual foi gerado. Isto é importante na marcação digital de tempo (*timestamp*) onde usando funções hash, alguém pode obter um documento com marcação de tempo sem que seu conteúdo seja revelado para o serviço de marcação de tempo.

Exemplos de funções hash bem conhecidas são MD5 [55] e SHA-1 [18].

2.4 Assinatura Digital

Uma assinatura digital é a garantia de não repúdio para quem recebe uma mensagem ou um arquivo. Em última instância tem o mesmo valor que uma assinatura a mão autenticada em cartório no mundo digital. Isso ocorre porque a assinatura digital é gerada

usando-se a chave privada de um usuário, a qual somente ele tem acesso e conhecimento. Desta forma o usuário que assinou alguma mensagem não tem meios de dizer que não o fez pois ninguém mais conhece sua chave privada em condições normais.

Por este motivo é muito importante que a chave privada seja guardada segura e que a sua chave pública correspondente fique em entidade confiável. Os problemas que podem advir da quebra da chave privada são muitos e vão desde a quebra de confidencialidade na comunicação decifrada com a chave privada até o envio de mensagens forjadas e assinadas por um intruso como se fosse o dono da chave.

A assinatura digital é conseguida cifrando o hash de um documento ou mensagem com a chave privada de um usuário. O receptor ao receber a mensagem ou documento decifra o hash com a chave pública do emissor e então tem a garantia de que a mensagem foi enviada realmente pelo dono da chave privada.

Uma outra técnica bastante usada e que combina criptografia simétrica e assimétrica é o envelopamento digital. Nesta técnica os seguintes passos são executados:

- O emissor gera uma chave simétrica de sessão e cifra a mensagem a ser enviada com esta chave;
- O emissor gera um hash da mensagem cifrada e cifra o hash com a sua chave privada;
- O emissor cifra a chave simétrica de sessão com a chave pública do receptor e a envia junto com a mensagem para o receptor;
- O receptor recebe a mensagem e decifra a chave simétrica com a sua chave privada e o hash com a chave pública do emissor. Assim obtém a chave simétrica e a garantia de que foi realmente o emissor que enviou a mensagem;
- O receptor calcula o hash para verificar a integridade da mensagem;
- Se o hash gerado for igual ao anexado na mensagem então a mensagem é decifrada com a chave simétrica de sessão.

Esta técnica é utilizada por exemplo no PGP [11] garante a troca de chaves simétricas, a autenticidade, confidencialidade, integridade e o não repúdio da mensagem envolvida na comunicação.

2.5 Certificado Digital

Certificados digitais são equivalentes eletrônicos às provas físicas de identificação, como passaporte e cédulas de identidade, que auxiliam a autenticação de usuários em redes de comunicações. O certificado digital é um arquivo binário que contém pelo menos os campos mostrados na Figura 2.4.

O certificado digital pode estar armazenado em uma estação, um disquete, um dispositivo de segurança como um smart-card ou um token. São elementos essenciais em uma ICP. Na verdade todo certificado digital é assinado digitalmente por uma AC que lhe dá garantia de autenticidade.

Figura 2.4: Certificado padrão X.509.

Atualmente o uso de certificados é bastante difundido visando dar maior segurança no processo de autenticação. Os certificados permitem que se faça uma autenticação baseada em pelo menos dois fatores: o que se tem (o certificado) e o que se sabe (uma senha) ou mesmo o que se é (através de biometria).

Os certificados possuem um tempo de vida especificado no seu campo de validade e podem ser revogados por diversas razões. Dentre elas pode-se destacar:

- Comprometimento da AC que o assinou ou de alguma AC superior na hierarquia da ICP;
- Comprometimento da Chave Privada;
- Mudança de Status;
- Suspensão;

- Outras razões administrativas de acordo com a política de certificação.

Existe uma lista de certificados revogados que deve ser mantida pela ICP. Esta lista normalmente fica publicada num servidor de diretório. A lista de certificados revogados é chamada de CRL (Certificate Revocation List). As CRL's são emitidas de tempos em tempos (tempo este definido no documento de política de certificados da ICP) e assinadas pela AC. Em qualquer operação com certificados a CRL deve sempre ser checada pela entidade que recebe o certificado de um parceiro na comunicação.

2.6 Infra-Estrutura de Chave Pública

Uma infra-estrutura de chaves públicas (ICP) é uma combinação de produtos de hardware e software, procedimentos e políticas de segurança. Infra-estrutura de chaves públicas tem sido usada largamente para garantir confidencialidade, integridade e garantia de não repúdio em transações onde parceiros que não se conhecem podem trocar informações de modo seguro através de cadeias de credibilidade. A ICP utiliza certificados de chave pública ou certificados digitais que vinculam os usuários a uma chave pública.

Este esquema é baseado em cifragem assimétrica onde cada usuário possui uma chave pública e uma chave privada. A chave pública é conhecida de todos e a chave privada fica com o usuário. Conforme demonstrado na seção de algoritmos assimétricos, tudo que é cifrado com a chave pública é decifrado com a chave privada e vice-versa.

A infra-estrutura é composta minimamente dos seguintes papéis conforme a Figura 2.5:

- Autoridade Certificadora - É a entidade que gera os certificados e faz a verificação da validade dos mesmos através de sua chave pública e da Lista de Certificados Revogados;
- Autoridade Registradora - É a entidade que faz o cadastro dos usuários do serviço de certificação e verifica a identidade correta do usuário;
- Lista de certificados revogados (CRL) - Uma lista dos certificados que foram revogados e devem ser considerados fora de uso;
- Serviço de diretório - Serviço normalmente implantado em estrutura hierárquica que contém os usuários cadastrados e seus certificados;

Figura 2.5: Esquema de Infra Estrutura de Chaves Públicas.

- Usuários do serviço - Qualquer entidade que utilize a infra-estrutura.

Os serviços de AC, AR e Diretório podem estar em um mesmo servidor ou podem ser separados conforme o tamanho da ICP. Em um esquema de certificação cruzada, uma autoridade certificadora confia nos certificados gerados por outra autoridade certificadora e vice-versa.

A ICP é uma solução para o problema de distribuição de chaves públicas de algoritmos assimétricos citado anteriormente. As chaves públicas podem ficar disponíveis em máquinas "confiáveis"(certificadas) pertencentes à infra-estrutura onde podem ser consultadas por qualquer participante envolvido na comunicação.

2.7 Limite Criptográfico

O limite criptográfico ou limiar criptográfico é usado para mitigar o problema de chaves centralizadas (chave privada do serviço numa ICP por exemplo) num sistema de segurança. Se o servidor principal ou a chave do serviço forem comprometidos então todo o sistema é exposto.

Um segredo, especificamente uma chave SK, é particionada e enviada para N nós da rede de acordo com um polinômio randômico de ordem K-1. Um conjunto de K nós, com K compartilhamentos, pode recuperar SK por interpolação de Lagrange [15], enquanto que qualquer conjunto de K-1 nós não consegue recuperar qualquer informação a respeito de SK. Neste caso, K é definido como o limite criptográfico, sendo esta a quantidade

mínima de partes para gerar SK.

Com esta técnica o sistema torna-se mais robusto e tolerante a falhas. Existem outras aplicações para o limite criptográfico. Um outro exemplo é a divisão de um segredo para votação num grupo. Vamos supor que uma chave secreta foi quebrada em oito partes e são necessárias pelo menos quatro partes para reconstruir a chave secreta. O presidente de uma empresa poderia ter três partes do segredo. Cada diretor da empresa poderia ter uma parte. O número total de diretores é de cinco. Para aprovar uma proposta ou uma decisão seriam necessárias pelo menos quatro partes (o presidente mais um diretor ou quatro diretores por exemplo).

2.8 Criptografia por ID

A técnica de limiar criptográfico ou limite criptográfico, pode ser usada com outras técnicas de criptografia assimétrica. Uma delas é a criptografia por ID [39], [8], [28]. Nesta técnica a chave pública usada por cada entidade é um identificador qualquer reconhecido por um serviço chamado PKG (Private Key Generator). O sistema usa uma chave pública mestra conhecida de todos e uma chave privada mestra. As chaves secretas de cada entidade participante são geradas após o nós se autenticarem no PKG que gera então a chave privada da entidade solicitante. Este esquema também pode usar a técnica de limiar criptográfico para dividir a chave privada do serviço (*master private key*) a fim de evitar o ponto central de falha do sistema, que neste caso é o PKG. Note que o processo de autenticação no PKG é altamente crítico e têm sido fonte de estudo de vários autores [8] e [28].

A vantagem desta última técnica é que não é necessário um serviço de infra-estrutura de chaves públicas pois a identidade possui um vínculo automático com seu dono. No entanto, é inerente ao modelo baseado em identidades a característica, quase sempre indesejável, de custódia de chaves, em que uma entidade de confiança gera (e portanto conhece) as chaves secretas de todos os usuários (neste caso o PKG). Isso enfraquece alguns aspectos de segurança como a incontestabilidade que por sua vez acaba com a garantia de não repúdio.

De posse dos conceitos apresentados é possível apresentar agora os trabalhos relacionados mais importantes nos quais as propostas deste trabalho foram baseadas.

Capítulo 3

Trabalhos Relacionados

Neste capítulo são apresentados os trabalhos relacionados mais importantes que serviram de base para esta dissertação. O objetivo é dar ao leitor o conhecimento resumido dos referidos trabalhos para um melhor entendimento das propostas aqui apresentadas.

Inicialmente será apresentado um estudo sobre perda de pacotes em redes ad hoc. Este estudo é importante na medida em que permite que sejam conhecidas características do comportamento de uma rede ad hoc, no que se refere à perda de pacotes, quando submetida a variados tipos de carga e diferentes tipos de protocolos. Este estudo permitirá calcular as probabilidades de perda de pacotes que serão usadas na análise de desempenho da proposta da dissertação.

Logo depois será apresentada uma técnica básica de criptografia para compartilhar um segredo de chaves distribuídas em diversos servidores. Esta técnica será explorada por mecanismos para prover comunicação segura em redes ad hoc nas seções subseqüentes. Também será apresentado um modelo proposto por Zygmunt Haas e Lidong Zhou que se utiliza da técnica de limite criptográfico para prover segurança em redes ad hoc. Depois do modelo proposto por Haas e Zhou serão apresentadas três propostas que se utilizam da combinação de técnicas de criptografia para prover segurança em redes ad hoc. Duas das propostas são feitas pelo mesmo grupo de pesquisdores liderados por J.Kong e a última proposta é feita por Seung Yi e Robin Kravets.

3.1 Perda de Pacotes em Redes ad hoc

Para melhor entender o comportamento e a eficiência de qualquer protocolo proposto, é importante contextualizá-lo num ambiente com perda de pacotes como é o ambiente em redes ad hoc. O mais difícil porém é encontrar um modelo que seja bom o suficiente para

acomodar as flutuações de carga da rede e medir a perda de pacotes nestas situações.

Em [43] é feito um estudo com simulações onde são estudadas diversas situações de perda de pacotes em redes ad hoc. São feitos experimentos com alguns protocolos conhecidos (AODV, DSR e DSDV), diferentes taxas de transmissão e diferentes tipos de fluxos (CBR e TCP).

O trabalho mostra as diversas causas de perda de pacotes em redes ad hoc e suas conseqüências para os protocolos de roteamento estudados. Podem ser citadas como principais causas de perda de pacotes a movimentação de nós, congestionamento por falta de buffers e congestionamento por conta do uso do canal. O trabalho possui um cenário muito similar ao que é proposto nesta dissertação e por isso os dados do estudo foram tomados como base para as análises matemáticas apresentadas na proposta. O cenário que serviu de base apresenta conexões CBR, protocolo DSDV, tamanho de pacote de 4Kbytes e taxa de transmissão de quatro pacotes por segundo.

O ambiente de simulação é o NS-2 (versão 2.1b9). Cada "host" utiliza uma antena omni direcional e a interface de rede é a WaveLan DSS 914 MHz da Lucent. A capacidade máxima de transmissão é de 2Mbps com alcance médio de 250 metros. Como protocolo MAC, é usado o IEEE 802.11 com função de coordenação distribuída (DCF) juntamente com o CSMA/CA.

As conclusões mais importantes do estudo [43] são as seguintes:

- As perdas de pacotes variam de acordo com o protocolo de roteamento usado. Mobilidade é a causa dominante para perdas com AODV e perdas por congestionamento afetam mais o DSDV;
- DSDV perde de 10 a 20 por cento mais pacotes que AODV em tráfego UDP. Nos tráfegos TCP as perdas são similares.

Além disso, o estudo mostrou certo padrão na perda de pacotes no tempo dos experimentos. Considerou-se nesta dissertação que os valores de perda de pacotes apresentados em [43] são típicos para cenários de redes ad hoc. A partir deles, foram calculadas as probabilidades de perda em diversos cenários a serem apresentados no capítulo 5.

3.2 Como Compartilhar um Segredo

No trabalho ‘‘How to Share a Secret”[60], Shamir propõe um método de dividir um segredo (uma chave) em N partes. Para recuperar este segredo são necessárias K partes. Esta técnica já foi citada no capítulo anterior e é conhecida como limite criptográfico.

O esquema é baseado em interpolação polinomial (Método de Lagrange) onde dados K pontos no plano bidimensional (x_i, y_i),...(x_k, y_k) com x_i's distintos, existe um e apenas um polinômio $Q(x_i) = y_i$ para todo i. O segredo D é dividido em N partes D_i, tomando-se um polinômio randômico de grau k-1 no qual $a_0 = D$ na forma:

$Q(X) = a_0 + a_1X + ... + a_{k-1}X^{k-1}$

Pode-se calcular:

$D_1 = Q(1), ..., D_N = Q(N)$, onde $K < N$;

Dado qualquer subconjunto de K dos valores D_i juntamente com seus índices, pode se achar os coeficientes de Q(x) por interpolação e então calcular D=Q(0)=a_0.

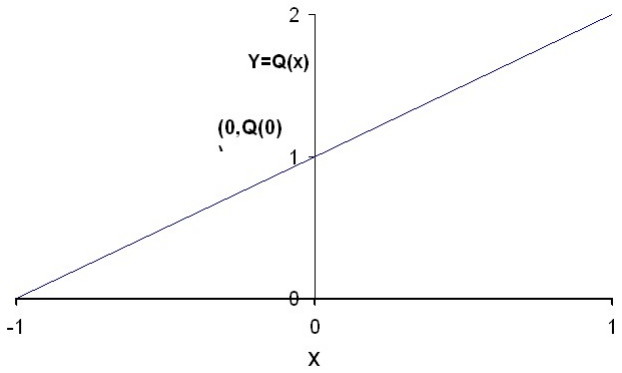


Figura 3.1: Representação para o limite criptográfico com K=2.

Um caso especial onde k=2 (isto é, apenas duas partes são requeridas para calcular o segredo) é mostrado na Figura 3.1. O polinômio é uma reta e o segredo é o ponto onde a reta intercepta o eixo Y. Este ponto é o ponto (0,Q(0))=(0,a_0). Cada parte ou compartilhamento é um ponto da reta. Cada dois pontos determinam a reta e por conseguinte o segredo. Com apenas um ponto a reta pode ser qualquer reta que passe por este ponto e assim o segredo não pode ser calculado.

Segundo Shamir existem algoritmos eficientes de ordem O($Nlog^2N$) para interpolação polinomial, mas mesmo algoritmos de ordem quadrática são rápidos o suficiente para esquemas como estes. Algumas propriedades do esquema de limite criptográfico são:

1. O tamanho de cada parte do segredo não pode exceder o tamanho do segredo original;

2. Quando o limite "K" é mantido fixo, as partes D_i podem ser dinamicamente adicionadas ou removidas. Tudo que é preciso para mudar tais partes sem trocar o segredo original "D" é um novo polinômio Q(x) com o mesmo termo independente. Com isso é feita a atualização pró-ativa dinâmica das partes aumentando a segurança.

Através desta técnica é possível compartilhar partes de um segredo.

3.3 Um Modelo para Redes Ad Hoc com Limite Criptográfico

Em seu trabalho "Securing ad hoc Networks"[63], Zygmunt Haas e Lidong Zhou sugerem um modelo de segurança para redes ad hoc. Este modelo é baseado em limite criptográfico dividindo a chave secreta da ICP em vários nós. O foco da proteção é dado nos protocolos de roteamento a fim de proteger a rede de ataques de negação de serviço desferidos por adversários internos ou externos.

A proposta dita que a comunicação de controle seja cifrada e autenticada pelo serviço de ICP com limite criptográfico. Neste modelo existem vários nós na rede com funções para assinar e verificar com sua parte da chave do serviço as mensagens de roteamento. Também é proposta a atualização pró-ativa dos compartilhamentos (partes da chave secreta do serviço) conforme descrito na seção anterior, para aumentar a segurança contra ataques que visem se apropriar das partes das chaves alcançando o limite criptográfico. O servidor que faz a combinação das assinaturas parciais para assinar ou cifrar as mensagens chama-se AC Combinadora. Não é especificado como esta AC combinadora é utilizada na arquitetura, não estando claro se podem ser utilizadas múltiplas ACs combinadoras ou se qualquer AC pode desempenhar a função de combinadora.

Uma AC ou nó atacado pode se tornar indisponível ou apresentar comportamento Bizantino (executar seu protocolo de forma errada levando a situações de erro na rede). Desta forma os autores também assumem um modelo de comunicação assíncrono pois qualquer sincronismo é uma vulnerabilidade que pode ser explorada. Como o modelo é assíncrono, é difícil (às vezes impossível) distinguir entre um servidor comprometido e um servidor não comprometido que apresenta alto tempo de resposta e lentidão [23]. Por este motivo, o modelo trabalha com a noção de consistência fraca [7]. Uma rede ad hoc

pode se encaixar no modelo de sistema distribuído que se comunica dentro de um grupo. Os elementos ou nós entram e saem do grupo ou da rede e é preciso garantir algum tipo de consistência nas variáveis e estruturas de dados através de um conjunto de primitivas para envio, recebimento e controle de mensagens. Não é necessário que todos os nós estejam com suas variáveis de controle e estruturas consistentes com relação aos outros nós após cada operação, mas é necessário que um número suficiente de nós esteja com suas informações atualizadas e operando normalmente dentro do sistema. Estas informações podem ser tabelas de roteamento, listas de controle ou estruturas de dados auxiliares. O estado das variáveis de um membro do grupo pode estar levemente diferente ou conter desvios quando comparado ao estado global do sistema.

A idéia central é que os nós ou roteadores da rede, troquem as informações de controle (principalmente as informações de roteamento) de maneira segura. Isso se dá através do uso do serviço de ICP onde um roteador se autentica com seu vizinho usando o par de chaves (pública e privada), antes de começar a transmitir as informações de controle.

O trabalho de Haas e o da seção anterior de Adir Shamir, são as bases da proposta desta dissertação conforme será visto no próximo capítulo.

3.4 Uma Proposta para ICP em Redes ad hoc

O trabalho [40] também usa ICP com limite criptográfico, e atualização pró-ativa da chave secreta do serviço de ICP. Os autores propõem um serviço de ICP com várias ACs onde um nó só poderia solicitar os serviços da ICP se tiver pelo menos "K"(o limite criptográfico) ACs vizinhas a uma distância de um salto.

O trabalho [44] é uma extensão dos mesmos autores do trabalho [40]. Em ambos os trabalhos, a comunicação com o serviço de ICP é feita por difusão mas a reposta do serviço de ICP para os nós é feita via *unicast* pelas ACs da infra-estrutura. Uma contribuição importante foi o protocolo de construção da lista de certificados revogados ou *CRL*.

No primeiro trabalho foram propostos dois mecanismos de revogação de certificados. O primeiro mecanismo é baseado em um parâmetro chamado T_{renew} que é o tempo para a renovação do certificado. Se o nó não renovar o certificado num tempo menor que T_{renew} então o certificado é revogado. O segundo mecanismo de revogação se propõe a invalidar certificados comprometidos em tempo real. O protocolo que marca um nó como comprometido só é descrito no segundo trabalho em [44] e a atualização da CRL é feita através de um pacote com a identificação do nó comprometido e o tempo em que a infra-

estrutura verificou sua invalidade. Este pacote é enviado a todas as ACs. Requisições ou certificados assinados pelo nó após este tempo são desconsiderados e assim a lista de certificados revogados é construída em cada nó.

O protocolo para revogação e verificação de certificados de [44] será aproveitado e modificado na proposta contida nesta dissertação.

3.5 Outra Proposta para ICP em Redes ad hoc

No relatório "Practical PKI for ad hoc Wireless Networks", Seung Yi e Robin Kravets também usam o conceito de limite criptográfico proposto por Shamir. Assim como no trabalho de Haas os nós geram assinaturas parciais que são reconstruídas para gerar a assinatura do serviço de ICP em um dado pacote. O trabalho descreve em linhas gerais o protocolo de renovação de certificados e um estudo entre o número de pacotes enviados (CREQ) e pacotes recebidos (CREP) no processo de renovação de um certificado. Analisam o comportamento destes pacotes variando o número de AC's, tempo de pausa e velocidade dos nós.

Em [59], os autores usam difusão para tentar contatar todas as ACs quando de um pedido de certificado (pacote CREQ) usando o esquema de cifragem por limiar. Através de simulações é mostrado que cada nó recebe em média $2N/3$ respostas, sendo N o número de ACs.

Este trabalho é relevante, pois o protocolo de renovação de certificados será modificado na proposta desta dissertação visando diminuir o número de mensagens de controle trocadas desnecessariamente.

3.6 Comparação entre os trabalhos

Conforme já citado, em [59] os autores usam difusão para entrar em contato com o serviço de ICP. A proposta desta dissertação modifica o protocolo de renovação de certificados tornando-o mais eficiente. Esta técnica de inundação, utilizada por Seung e Kravets no pedido de certificados, gera mais pacotes de controle do que o necessário e os próprios autores colocam o seguinte cenário:

Para 1000 requisições de certificados com 30 ACs, são gerados 119125 pacotes CREQ quando na verdade o número mínimo de pacotes deste tipo seria $1000 \times T$ (limite crip-

tográfico). Com T=10, por exemplo, o número de pacotes CREQ mínimo seria de 10000, bem inferior aos 119125 pacotes gerados.

Em [40] e [44] é apresentado um modelo de ICP com cifragem por limiar, ou limite criptográfico, onde a CRL é construída progressivamente através de contadores de certificados assinados digitalmente que são propagados por inundação quando da sua assinatura. Nesta dissertação será proposto um modelo que guarda em cada AC, a lista de certificados revogados. Além disso, uma lista de ACs confiáveis é guardada em cada nó e em cada AC. A atualização das CRL's de cada AC é feita por comunicação *unicast* (em oposição à inundação) entre as ACs assim que um certificado é revogado.

A proposta em [40] e [44] é baseada em um esquema de serviço único de ACs distribuídas. A proposta desta dissertação pode acomodar múltiplos serviços de ACs distribuídas bastando para isso, em um futuro trabalho, estabelecer o protocolo de controle e manutenção de confiança entre as ACs de diferentes serviços.

Também em [40] e [44] os autores partem do princípio de que cada nó deverá ter pelo menos K (limite criptográfico) ACs vizinhas a um salto de distância. Caso contrário o nó deverá esperar até conseguir ter este número de vizinhos e fazer a requisição ao serviço de ICP. Pela proposta constante nesta dissertação isso não é necessário pois os nós agem como roteadores e passam as requisições para as ACs da ICP mesmo que estejam a mais de um salto de distância, aproveitando assim os múltiplos caminhos existentes numa rede ad hoc.

Em [63], a técnica de cifragem por limiar é usada na ICP e é proposto pelos autores o uso de um servidor denominado "combinador", cuja função é computar a assinatura para os certificados gerados pela infra-estrutura. Nesta dissertação qualquer servidor da ICP pode fazer o papel de AC combinadora aumentando assim a disponibilidade do serviço.

Com este breve histórico e explicações preliminares é possível agora explicar de forma detalhada as propostas desta dissertação no próximo capítulo.

Capítulo 4

Proposta

A proposta desta dissertação consiste em apresentar um serviço de ICP para redes ad hoc denominado ICPAH com o objetivo de protegê-la de ataques do tipo falsificação de identidade e buraco negro. Este serviço é formado por "N" ACs onde cada uma confia nos certificados emitidos pelas outras ACs que estão em uma lista de ACs confiáveis denominada LAC, presente em cada AC e cada nó.

Inicialmente os nós da rede ad hoc obtém a renovação dos seus certificados utilizando o serviço ICPAH. A utilização da técnica de limite criptográfico faz com que o conjunto de ACs se comporte no serviço ICPAH como se fosse uma única AC. Os nós se autenticam com seus vizinhos que verificam a validade dos certificados e a CRL antes de qualquer operação ou troca de dados. Os protocolos de autenticação, de renovação de certificados, manutenção da LAC e da CRL serão descritos nas seções subseqüentes.

Cada AC e cada nó possui seu par de chaves (pública e privada) e o serviço de certificação possui também seu par: a chave pública é conhecida de todos os nós e a chave privada é dividida entre as "N" ACs. Para que uma requisição de certificado seja processada pelo serviço, é preciso pelo menos "T" (limite criptográfico [60]) partes desta chave privada para reconstruir a chave do serviço como um todo. Este limite pode variar e é configurado na inicialização das ACs. Quanto maior o limite mais confiável e seguro o serviço, porém menos tolerante a falhas, pois mais ACs deverão fornecer sua parte da chave privada corretamente. Neste trabalho porém, o limite configurado na inicialização é T=(N+1)/2 conforme sugerido em [60].

Quando se usa a técnica de limite criptográfico conforme detalhada em [15] e [63], é preciso proteger a infra-estrutura contra servidores comprometidos. Por exemplo: um servidor (AC) comprometido poderia gerar uma assinatura parcial incorreta e o uso da mesma geraria uma assinatura inválida. A validade de uma assinatura parcial pode

ser verificada usando a chave pública do serviço [15, 31, 26, 53]. No caso de falha da verificação de uma assinatura parcial, outra assinatura será necessária até que se alcance o limite criptográfico necessário para reconstruir a chave privada do serviço. A AC que faz a combinação das assinaturas parciais é denominada AC combinadora. Por conta disso a AC combinadora requisita uma quantidade de assinaturas parciais maior do que o limite criptográfico.

4.1 Arquitetura Proposta

No modelo proposto os nós e as ACs precisam ser inicializados com seus certificados digitais, sua chave privada, com a chave pública do serviço e com a LAC do ICPAH. As chaves públicas dos nós e das ACs são enviadas nos certificados durante o funcionamento dos protocolos e algoritmos da proposta.

Em um primeiro cenário para esta dissertação, as ACs são fixas e têm posições pré-estabelecidas dentro da rede. Esta distribuição das ACs não é mandatória, o que significa que não há necessidade das mesmas serem fixas. Esta configuração foi feita apenas como exemplo para modelar o uso de ACs distribuídas em roteadores fixos de uma rede em malha sem fio. A distribuição do serviço de AC deve ser feita em computadores pertencentes à infra-estrutura por questões de confiança da manutenção das partes da chave do serviço. Num cenário com segurança mais fraca qualquer nó da rede poderia fazer o papel de AC. Também é considerado apenas um serviço de certificação. Não são considerados mais de um serviço cada qual com seu par de chaves pública/privada.

Para prover o serviço de autenticação e troca de mensagens de roteamento de forma segura, é proposta a criação de um sub-nível na camada de rede do modelo OSI. Esta sub-camada é responsável por executar os protocolos de autenticação e renovação de certificados contidos nesta dissertação além de outras atividades de controle e segurança. O principal componente de software instalado nesta sub-camada é um Agente de segurança responsável por executar os protocolos citados. Após a autenticação provida pelo Agente de Segurança, os protocolos de roteamento de redes ad hoc são usados normalmente utilizando comunicação cifrada com uma chave simétria de sessão.

Além de um algoritmo de cifragem de chave simétrica, que visa dar maior desempenho, é usado um esquema de troca de números de seqüência aleatórios na comunicação (*nonce*) visando proteção contra ataques do tipo repetição.

A política padrão de requisição de novos certificados foi estabelecida como “NE-

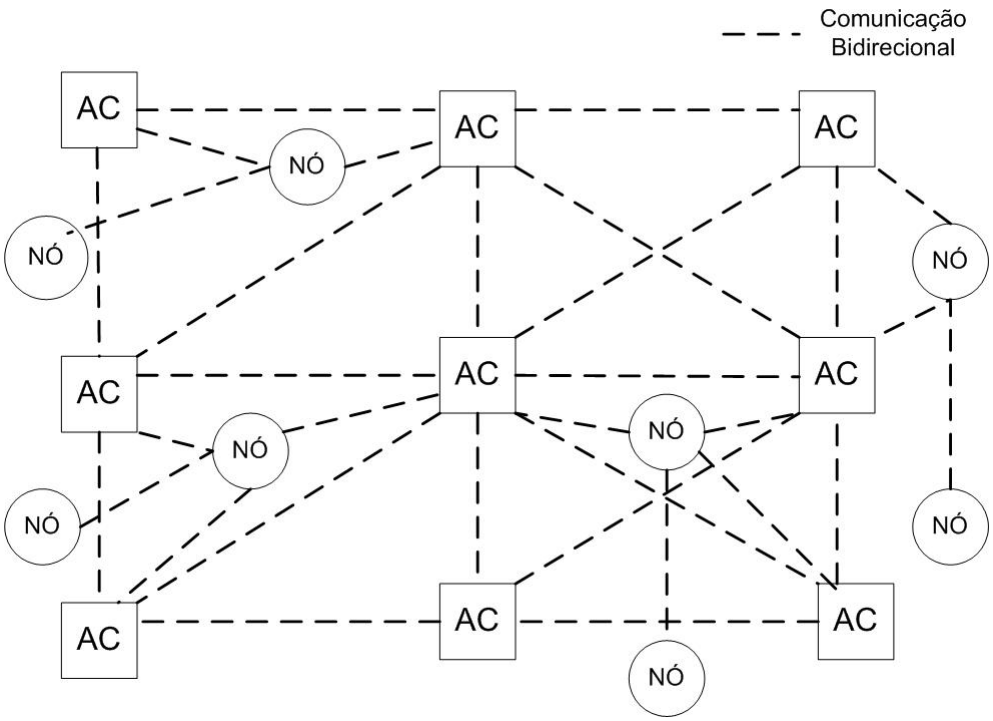


Figura 4.1: Exemplo de Arquitetura.

GADA", ou seja, nenhum novo certificado pode ser requerido visando impedir que novos nós entrem na rede. Já a política de renovação de certificados foi definida como "ACEITA", ou seja, as ACs aceitam pedidos de renovação de certificados já gerados por elas.

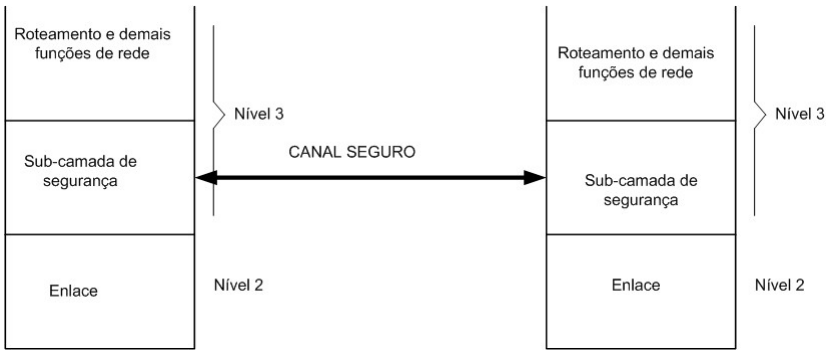


Figura 4.2: Sub-camada de segurança.

4.2 Inicialização das ACs e Distribuição dos Certificados

Visando dar maior confiabilidade e para diminuir a complexidade, as ACs e os nós recebem de maneira "*off-line*" (fora da rede) seus certificados iniciais. Também de maneira *off-line*

é gerado o par de chaves do serviço de certificação e a chave privada do serviço é então dividida entre as ACs do sistema. Este procedimento tenta garantir que o modelo com ACs descentralizadas apresente as mesmas características que uma AC centralizada, evitando ataques de ACs externas que não pertençam ao conjunto ou que foram comprometidas [17] (Sybil attack).

Pode não ser prático em determinados cenários fazer a inicialização do sistema de maneira "*off-line*". Em cenários mais estáticos com maior controle a inicialização "*off-line*" não seria problemática, porém em cenários mais abertos e com necessidades de segurança menores, esta restrição pode se tornar um sério entrave. A atribuição, registro e distribuição de certificados ou de identidades de maneira "*on-line*" pode ser endereçada em algum trabalho futuro, porém para o escopo desta proposta isso não será tratado.

Seria possível alterar a política de requisição de novos certificados para "ACEITA" se os novos nós já dispusessem de um certificado emitido por outra autoridade certificadora confiável (certificação cruzada). Desta forma, o novo nó entraria na rede se autenticando com o certificado de outra AC e pediria a geração de um certificado do serviço ICPAH.

A entrada de novos nós na rede pode ser feita também através da geração de certificados assinados com a chave privada do ICPAH que estaria numa AC *off-line*. Desta forma, após o nó receber seu certificado poderia começar a estabelecer rotas e trocar mensagens utilizando a arquitetura proposta.

Na inicialização do sistema ICPAH, são geradas duas constantes que são utilizadas nos protocolos da proposta. A primeira constante é resultado da função F_{ac} que é uma relação entre N (número de ACs) e T (limite criptográfico). Esta constante será batizada de Kac e estabelece o número de mensagens enviadas por uma AC para as outras ACs do serviço a fim de reconstruir a chave privada do serviço ICPAH. Esta primeira constante deve sempre ser maior que o limite criptográfico para compensar possíves perdas de mensagens enviadas para as outras ACs.

A segunda constante gerada na inicialização também pode ser obtida como função dos mesmos parâmetros (N e T) e denota a quantidade de ACs que um nó deverá contatar para fazer a verificação da chave pública de outro nó (protocolo de autenticação) e para pedir a renovação do seu próprio certificado. Esta segunda constante será batizada de Kno. É importante que esta constante seja escolhida de forma a também compensar possíveis perdas da rede evitando rodadas adicionais dos protocolos de autenticação e renovação de certificados como será mostrado mais a frente.

4.3 Listas de Certificados Revogados e de ACs Confiáveis

A lista de certificados revogados é mantida replicada em cada AC. Está sendo proposta a criação e manutenção de uma lista de ACs "confiáveis" (LAC) do sistema que é mantida em cada AC e em cada nó. Na inicialização das ACs a lista contém todas as ACs do sistema.

Quando da geração e instalação do certificado digital no nó, é gravada a lista de ACs confiáveis que servirá para o nó poder aceitar ou não certificados de ACs que porventura tenham sido comprometidas ou que estejam fora de serviço.

Existem duas maneiras de uma AC ser removida da LAC. A primeira ocorre quando a AC não é alcançável e fica fora da tabela de roteamento das outras ACs. Isso quer dizer que se uma AC não puder ser alcançada através do protocolo de roteamento, mesmo que passando por nós intermediários, a mesma será removida da LAC. Esta condição de falha deve ser comprovada por pelo menos "T" (limite criptográfico) ACs. Esta situação pode ocorrer se alguma AC sofrer um ataque de negação de serviço ou por algum motivo for desligada ou removida da rede. Não faz parte do escopo deste trabalho descrever as medidas que protegem o sistema de um ataque tipo *DOS* ou *DDOS*. O sistema deve prever medidas para estancar e prevenir tais ataques para que o mesmo não fique indisponível, porém este não é o foco desta dissertação.

A segunda maneira da AC ser removida da LAC ocorre quando a mesma envia a sua assinatura parcial corrompida ou errada. Como será visto na seção 4.4, a comunicação entre as ACs é cifrada com uma chave simétrica, evitando assim que um intruso capture pacotes, altere-os e faça uma reinserção dos mesmos na rede.

Nas duas situações acima a AC é removida da LAC e fica fora de serviço até que o administrador intervenha para recuperá-la. A AC que fez a verificação e removeu uma outra AC da lista de confiáveis, imediatamente altera sua lista e a envia por *unicast* para as outras ACs confiáveis da lista através de seus vizinhos na rede ad hoc.

A lista de ACs de cada nó pode ser implementada usando um número inteiro sem sinal - unsigned long int - de 32 bits, que é usado como um mapa de bits onde o valor "1"em determinada posição "k", representa que a AC correspondente é confiável. Quando uma AC se torna não confiável o valor "0"é preenchido na posição correspondente da AC. Por exemplo, se a AC número 10 se torna não confiável, o algoritmo coloca um "0"na posição 10 do string de 32 bits. Desta forma, esta dissertação assume um número máximo de 32 ACs, número este que foi mais que suficiente para os cenários simulados.

A lista de mensagens trocadas na utilização do protocolo de manutenção da lista de ACs confiáveis é a seguinte:

- Acusada: Mensagem enviada por uma AC para outra, acusando uma terceira AC. Composta por um código de identificação e o identificador da AC acusada;
- NovaLista: Mensagem que indica uma nova lista para atualização. Dentro desta mensagem vai a nova ListaConfiável.

Existem também as duas condições de remoção de AC da LAC já explicadas anteriormente.

Condição 1: Se existir uma assinatura parcial inválida gerada por uma AC, remover o mais rápido possível esta AC da lista de confiáveis e enviar a nova lista para todas as ACs pertencentes à lista. Esta condição é verificada pela AC combinadora;

Condição 2: Se uma AC_i não receber uma mensagem de resposta de uma AC_j (quando da requisição de assinatura parcial usando a constante Kac, ou caso exista falha no estabelecimento da rota para AC_j, então enviar mensagem "Acusada" $\forall N_i$ sendo N_i uma AC confiável e $N_i \neq AC_j$.

As condições 1 e 2 são checadas durante o protocolo de autenticação dos nós e de renovação de certificados.

O pseudocódigo do algoritmo de manutenção de ACs confiáveis é mostrado nos Algoritmos 1.1 e 1.2 do Apêndice A e será explicado abaixo:

A) Se ocorrer a CONDIÇÃO 1 então remove AC da LAC e envia nova LAC para todas as outras ACs confiáveis.

B) Se ocorrer CONDIÇÃO 2 então:

1. Uma AC "i" recebe uma mensagem "Acusada"vinda de outra AC "k"acusando uma terceira AC "j".
2. A AC que recebeu a mensagem checa se as ACs "k"e "j"são confiáveis e verifica se a AC "k"já acusou a AC "j"anteriormente.
3. Se é a primeira acusação, então esta acusação é contabilizada.
4. A AC "i" verifica se o contador de acusações de "j"já alcançou o limite criptográfico. Se o limite foi alcançado, então a lista de ACs confiáveis é modificada e é enviada

uma cópia da nova lista para cada AC confiável cifrada com a chave pública de cada AC destinatária. Com isso as demais ACs que recebem a lista decifram-na e executam uma operação de "AND bit a bit" entre a sua lista e a lista que estão recebendo. O resultado é a nova LAC. Este último passo é mostrado no algoritmo 1.2 do Apêndice A. Fica claro que as ACs só aceitam comunicação de vizinhos confiáveis conforme mostrado no passo 1 do algoritmo 1.2.

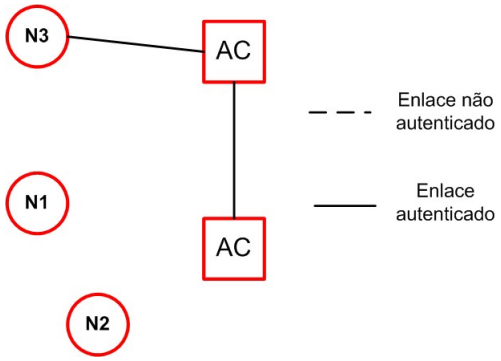
Caso uma AC_i consiga se comunicar com uma outra AC_j e o contador de acusações da AC_i contra a AC_j é diferente de zero, a AC_i deverá zerar seu contador de acusações contra a AC_j e enviar uma mensagem para todas as outras ACs confiáveis de sua lista solicitando que todas as ACs zerem o contador de acusações contra a AC_j caso este esteja diferente de zero. Este artifício é utilizado para que um número grande de ACs não seja considerada não confiável, num intervalo curto de tempo, por conta de perdas de pacotes na rede.

4.4 Funcionamento do protocolo de autenticação

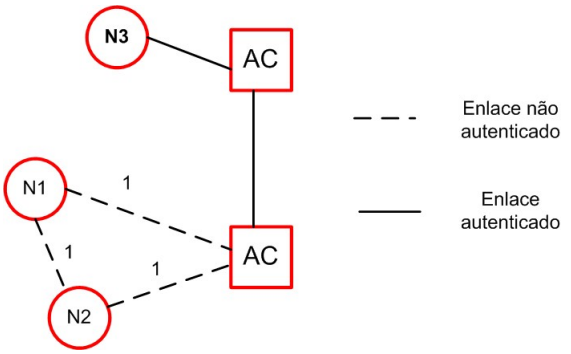
Sempre que um nó N1 se aproximar de um outro nó N2 que não esteja autenticado, deverá haver autenticação mútua entre os dois nós. Assim, um nó de origem N1 enviará para cada nó vizinho a requisição de autenticação. Por sua vez, o nó N2 fará o mesmo no estabelecimento do caminho reverso.

Estas autenticações ocorrem antes do estabelecimento das rotas entre dois nós quaisquer N1 e N2 (Figura 4.3), e é feita com a chave privada de cada nó do caminho que é verificada com sua chave pública por cada vizinho. Adicionalmente é enviado junto com a requisição de autenticação (AUTHENTICATION REQUEST) um *nonce* para evitar ataques de repetição. A cada autenticação os nós devem verificar se os certificados estão válidos checando a CRL em uma AC confiável usando a constante *Kno* para receber respota de pelo menos uma AC. Por este motivo é necessário que durante a autenticação, haja conectividade entre os nós e pelo menos uma AC.

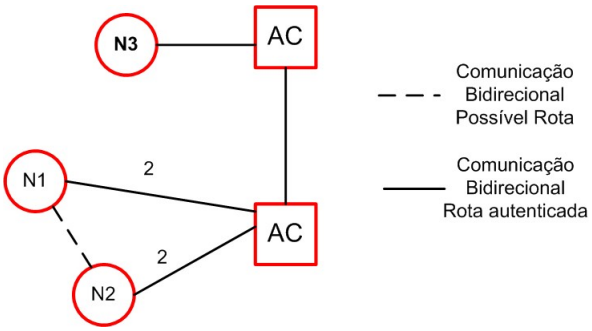
Após a autenticação, o receptor gera um número aleatório (*nonce*), uma chave simétrica de sessão de um algoritmo do tipo AES-128 bits ou DES e o pacote de AUTHENTICATION REPLY com a resposta ao AUTHENTICATION REQUEST. A chave de sessão é regerada a cada nova autenticação. Uma nova autenticação deve ocorrer após certo período de tempo máximo, configurado na inicialização dos nós e das ACs, quando se tornaria perigoso um ataque de quebra da chave simétrica.



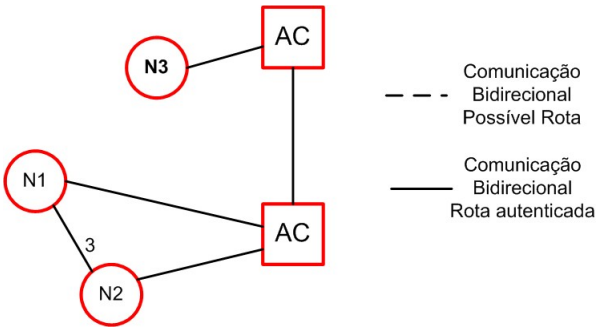
(a) Apenas as ACs e o nó N3 estão autenticados. N1 e N2 entram na rede



(b) N1 e N2 fazem requisição de autenticação com seus vizinhos



(c) N1 e N2 se autenticam com a AC vizinha



(d) N1 e N2 podem finalmente se autenticar

Figura 4.3: Estabelecimento de autenticação entre nós na rede

As mensagens trocadas entre o serviço ICPAH e os nós são as seguintes:

- Mensagem VerificaCert - Composta pelo certificado do nó que terá a chave pública verificada, a identificação do nó dono da chave pública a ser verificada e um código de 1 byte que conterá a identificação da mensagem. O último bit do código da mensagem serve para responder ao nó requisitante se a chave pública constante do certificado é válida.
- Mensagem CertVerificado - Composta pela identificação do nó dono da chave pública a ser verificada e o código da mensagem com o último bit representando a resposta do serviço ICPAH. Caso o bit tenha valor 1, a chave pública é válida, caso contrário inválida.

O algoritmo de autenticação é inspirado no SSL e segue os seguintes passos:

1. O emissor envia em difusão para todos os vizinhos, a requisição de autenticação cifrada com sua chave privada.
2. Os receptores decifram a mensagem com a chave pública do emissor. A chave pública do emissor é verificada no serviço ICPAH enviando a mensagem "VerificaCert" para *Kno* ACs.
3. Os receptores podem receber várias respostas do serviço ICPAH através da mensagem CertVerificado.
4. Os receptores enviam cifrados com a chave pública do emissor os seguintes parâmetros: a chave de sessão, o *nonce* e a resposta de requisição de autenticação. O receptor deve ainda cifrar o hashing da mensagem com sua própria chave privada. Esta assinatura digital provê a autenticação do receptor com o emissor.
5. O emissor então decodifica os parâmetros com sua chave privada e verifica as chaves públicas do receptor no serviço ICPAH. Os passos 2 e 3 são repetidos agora pelo emissor para verificar as chaves públicas de cada receptor.
6. O emissor responde com um reconhecimento (ACK) indicando que aceita a comunicação com o *nonce* e a chave simétrica escolhida. Neste ponto os nós já estão autenticados e com a chave simétrica escolhida.

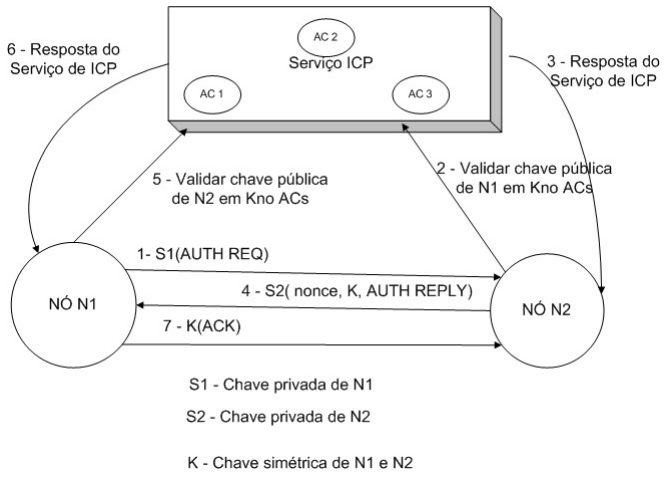


Figura 4.4: Protocolo de autenticação dos nós.

A partir daí as trocas de mensagens entre os nós se darão usando a chave simétrica acordada. Este protocolo é detalhado de forma gráfica na Figura 4.4.

É importante ressaltar que as autenticações ocorrem antes do estabelecimento das rotas e por isso é preciso estudar o impacto da inclusão deste protocolo na pilha de protocolos, abaixo dos protocolos de roteamento de redes ad hoc. Por este motivo são estudados o comportamento das redes, dentro do modelo proposto, utilizando um protocolo pró-ativo (DSDV) e um protocolo reativo (DSR). Variáveis como quantidade de nós na rede, mobilidade e o próprio protocolo de roteamento utilizado são avaliadas no próximo capítulo quando são mostrados os resultados, as simulações e as análises. Estas variáveis implicam no consumo de energia, número de autenticações necessárias e por conseqüência o número de mensagens de controle trocadas.

4.5 Renovação de certificados

O par de chaves do serviço ICPAH deve ser trocado ("*rekey*) sempre que o número de ACs confiáveis da LAC se aproximar do limite criptográfico. Sempre que o número de ACs confiáveis se aproxima do limite, o sistema pode não ter ACs suficientes para regerar a chave privada do serviço ICPAH. Esta condição pode ser verificada através de monitoração da LAC ou pelo envio de mensagem para o administrador informando este estado crítico. Neste caso, o serviço ICPAH deve ser inicializado novamente. Estas e outras funções de controle podem ser desempenhadas pelo Agente de segurança instalado nos nós da rede ad hoc. Desta maneira é promovida a renovação das chaves e o limite criptográfico pode ser alterado ou não. É importante ressaltar que no modelo ICPAH esta operação

implica na parada da rede. A chave privada do serviço também poderia ter suas partes ou "*shares*" modificados automaticamente pelas ACs conforme previsto em [63] numa técnica conhecida como "*dynamic shares update*" ou atualização pró-ativa, porém para simplificação do estudo esta técnica não é utilizada nesta dissertação.

Um nó da rede precisa trocar seu certificado ("*rekey*) quando o mesmo está próximo de expirar. Para determinar o tempo de validade dos certificados dos nós, deve ser levado em consideração o grau de confidencialidade e o tipo de aplicação que está sendo executada nos nós. Também deve ser levado em conta o tipo de ambiente em que os nós se encontram, pois em ambientes mais hostis é importante que os certificados sejam trocados com mais freqüência evitando assim que possam haver ataques bem sucedidos visando a quebra da chave privada dos nós. De qualquer forma, esta operação não deve ser muito freqüente quando comparada às operações de troca de mensagens e estabelecimento de rotas.

A lista de mensagens trocadas na execução do protocolo de renovação de certificados é mostrada abaixo. Logo depois o protocolo será explicado passo a passo.

- Renova: Mensagem vinda do Agente de segurança indicando a necessidade de renovação do certificado do nó;
- CertRenova: Enviada por um nó para as ACs solicitando a renovação de seu certificado;
- IniciaTempo: Mensagem interna enviada para o Agente de segurança. Inicia a contagem do tempo de expiração para receber uma resposta de uma AC;
- Reply: Mensagem enviada por uma AC que recebeu uma mensagem CertRenova. A AC está apta a renovar o certificado do nó;
- Ack: Reconhecimento. Um nó envia para uma AC que lhe enviou uma mensagem de Reply;
- Cert: Certificado digital que deve ser renovado. Enviado por um nó para uma AC;
- ResetTempo: Mensagem interna enviada por um nó para o Agente de segurança pedindo para parar a contagem do tempo de espera por resposta de uma AC;
- TempoExpirado: Mensagem vinda do Agente indicando que o tempo de resposta para uma ação expirou;

- AssinaParcial: Mensagem que pede a assinatura parcial no certificado do nó "i". Enviada por uma AC combinadora. Formada pela identificação do nó ("id") e pelo certificado ("cert");
- CertParcial: Mensagem enviada por uma AC para a AC combinadora. Formada pelo certificado assinado parcialmente ("certp") e pela identificação do nó ("id");
- FinalCert: Certificado assinado e renovado pelo serviço ICPAH. Enviado pela AC combinadora para o nó requisitante.

Para cada passo do protocolo de renovação de certificados pode haver mais de um algoritmo. Estes algoritmos estão detalhados no Apêndice A. No momento de renovar seu certificado o nó recebe uma mensagem interna do Agente de segurança indicando esta necessidade (mensagem Renova).

Para a renovação do seu certificado, o nó adota o procedimento descrito na Figura 4.5 que é explicado abaixo:

1. O nó vai escolher a AC combinadora pesquisando a lista de ACs confiáveis. A AC combinadora será a primeira AC a responder ao pedido do nó que utiliza a constante *Kno* para contatar um conjunto de ACs. Levando em consideração que o nó só precisa receber resposta de uma AC e que segundo [59], em média cada nó recebe 2/3 do número de mensagens enviadas quando usa a técnica de difusão, a constante *Kno* será usada para diminuir a quantidade de mensagens trocadas achando um valor entre 1 e N que será o número de ACs contatadas. A constante *Kno* escolhida deve ser tal que $PR \times Kno > 1$, sendo PR a probabilidade de receber mensagens de resposta em relação ao número de mensagens enviadas. Se não receber nenhuma mensagem de resposta o nó escolhe outras *Kno* ACs até conseguir comunicação ou até se esgotarem as ACs da sua LAC.
2. Após enviar as mensagens (mensagem tipo CertRenova) o nó inicia um temporizador enviando para o Agente de segurança uma mensagem interna tipo "TempoExpirado". Esta mensagem controla a quantidade de tempo que o nó irá esperar pelas respostas das ACs selecionadas. Quando o serviço ICPAH não responde ao nó requisitante num período menor que o especificado pela mensagem "TempoExpirado" outras *Kno* ACs são escolhidas (se a lista ainda não chegou ao fim) e o processo recomeça.

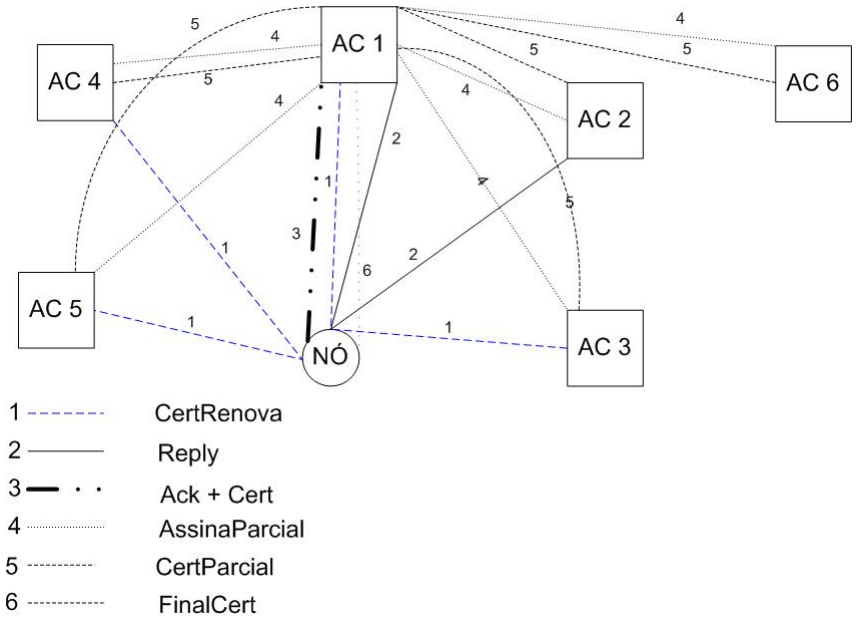


Figura 4.5: Protocolo de renovação de certificados.

3. Quando a AC responde com uma mensagem tipo "Reply"ao nó requisitante, o mesmo verifica se a mensagem está sendo esperada. Se é a primeira mensagem deste tipo, então a AC é escolhida como combinadora pelo nó.

4. O nó envia uma mensagem "Ack" para a AC.

5. A AC escolhida recebe a mensagem "Ack"vinda do nó requisitante e se torna a combinadora para a requisição.

6. A AC combinadora seleciona então outras *Kac* ACs para contatar, a fim de renovar o certificado do nó requisitante, solicitando suas assinaturas parciais no certificado. Cada AC também possui o Agente que controla o "timeout"para as respostas. A constante *Kac* deve sempre ser inicializada com valores maiores que "T" (limite criptográfico) para compensar possíveis perdas de mensagens enviadas para as outras ACs. Um estudo mais detalhado sobre os valores da constante *Kac* será mostrado no próximo capítulo.

7. Quando existe um "timeout"(mensagem TempoExpirado) enquanto a AC combinadora espera pelas respostas das outras ACs, a condição 2 [1] descrita anteriormente é verificada e são enviadas ou não as mensagens acusando as ACs que não responderam. Estas acusações dependem da condição 2. A AC combinadora faz então a nova seleção de outras ACs que serão contatadas. Esta seleção é feita verificando

[1]Se uma AC_i não receber uma mensagem de resposta de uma AC_j quando da requisição de assinatura parcial, ou caso exista falha no estabelecimento de rota para a AC_j

se a AC não foi escolhida anteriormente na mesma rodada de renovação. As novas requisições são enviadas e um novo temporizador é inicializado (mensagem interna "TempoExpirado" é enviada para o Agente). Este valor para controle de tempo é menor que o configurado para que os nós esperem pela resposta do serviço de certificação.

8. Uma AC que recebe a requisição para assinatura parcial do certificado (mensagem AssinaParcial) faz a assinatura parcial do certificado e responde para a AC combinadora (mensagem CertParcial).

9. Finalmente quando a AC combinadora recebe a resposta de pelo menos T-1 ACs, com as assinaturas parciais corretas, o certificado é renovado através da interpolação polinomial das assinaturas parciais das ACs. Neste momento a chave privada do serviço ICPAH é reconstruída para assinar o novo certificado. O certificado digital assinado é enviado para o nó (mensagem FinalCert) que agora possui um novo par de chaves.

Como premissa da proposta foi adotado que a AC combinadora sempre descarta a chave privada do serviço ICPAH de sua memória após a renovação do certificado. Isso é feito para que, se por acaso, uma AC seja comprometida ou invadida, o atacante não tenha acesso à chave privada do ICPAH. Num outro cenário com menos restrições, as ACs não necessitariam descartar tal chave e o ganho na diminuição do número de mensagens seria ainda maior, pois o protocolo para geração da chave privada do ICPAH só seria executado uma vez para cada AC. Nas próximas renovações a AC combinadora já teria a chave privada do ICPAH e renovaria diretamente o certificado pedido.

Este protocolo evita a inundação por difusão como em [59]. Será mostrado no próximo capítulo que o número de mensagens trocadas no processo de certificação diminui drasticamente. É certo que no passo 1 deste protocolo existe uma possibilidade, já descrita, de que nenhuma AC responda ao pedido do nó. Isso implicaria em uma segunda rodada do primeiro passo para selecionar outras ACs, o que poderia levar a um maior tempo de processamento e de espera quando comparado com a opção que utiliza difusão.

4.6 Análise de Complexidade dos Algoritmos

Os algoritmos de autenticação e renovação de certificados têm complexidade de mensagens da ordem das funções que geram as constantes Kno ou Kac, pois estas funções represen-

tam em última instância o número de mensagens que serão trocadas na execução dos algoritmos. Estas funções serão explicadas no próximo capítulo, mas podemos adiantar que as mesmas são sempre da ordem de O(N). As funções utilizadas nesta dissertação foram $\lceil N/Log_2 2T \rceil$ e $3/2 \times (T-1)$, sendo T=(N+1)/2.

O algoritmo de autenticação dos nós é O(N) também pois um máximo de $2 \times \lceil N/Log_2 2T \rceil$ mensagens são enviadas em cada autenticação.

A complexidade de tempo do algoritmo de autenticação é constante pois não depende do número de nós nem do limite criptográfico. Desta forma, o algoritmo é O(1), pois o número de passos executados é constante e igual a seis. O algoritmo de renovação de certificados, no pior caso, executa $N/(N/Log_2 2T)$ vezes o passo 1. Esta situação ocorre somente quando a lista de ACs confiáveis é percorrida até o fim. Após isso, o algoritmo executa mais cinco passos. Neste caso pode-se dizer que o algoritmo tem complexidade de $Log_2 2T + 5$ ou O(LogT) que é equivalente a O(LogN).

Sendo "p" a probabilidade de perda de pacotes na rede, o algoritmo de renovação de certificados sempre envia:

$Kno + (1-p) \times Kno$ *ou* $N/Log_2 2T$ e;

$(p \times (T-1)) + (p^2 \times (T-1)) + (T-1)$ ou $3 \times (T-1)/2$ mensagens.

Conclui-se que o algoritmo de renovação de certificados tem complexidade de mensagem de O(N), relacionada ao número de nós da rede (no pior caso quando $Kno = \lceil N/Log_2 2T \rceil$) e O(T) relacionado ao limite criptográfico.

Há também o custo do algoritmo para manter a lista de ACs atualizada e pronta para uso pelos diversos nós da rede. Esta lista será usada também para selecionar as ACs que irão tomar parte no processo de autenticação (constante Kno). Este algoritmo pertence ao processo de manutenção da arquitetura e é da ordem de O(LogN) para a busca (busca binária) e O(N/T) ou simplesmente O(N) para a criação do conjunto de ACs que serão selecionadas durante a renovação de certificados e durante a autenticação dos nós.

Por fim existe o algoritmo para computar e verificar a chave privada do serviço que executa uma interpolação polinomial. De acordo com [60] existem algoritmos eficiente da ordem de $O(NLog^2 N)$ para executar esta tarefa.

Com o uso dos protocolos descritos na arquitetura proposta, consegue-se mitigar os efeitos dos ataques buraco negro e de falsificação de identidade no nível de rede. O atacante pode se anunciar como um nó válido no ataque de falsificação de identidade, porém não

consegue obter sucesso na comunicação, visto que não consegue estabelecer comunicação no plano de rotas devido à impossibilidade de se autenticar usando o esquema de ICP proposto.

Com o atacante usando o buraco negro ocorre o mesmo. O atacante não consegue apresentar o mau comportamento descrito na introdução deste trabalho, nem descartar pacotes, visto que estes comportamentos só ocorrem após o estabelecimento das rotas. Como o atacante não consegue se autenticar corretamente para estabelecer as rotas, o ataque é frustrado.

Nesta proposta não é necessário que os nós da rede tenham várias ACs a seu alcance. Na verdade para a entrada do nó na rede basta apenas que o nó seja vizinho de uma AC confiável para o protocolo funcionar. No próximo capítulo será mostrado que o número de mensagens de controle trocadas aumenta na mesma proporção em que aumenta o número de ACs.

Defender uma estação de uma contaminação por algum vírus ou cavalo de tróia que apresente o comportamento e as características do buraco negro, não é o objetivo da proposta e por isso a mesma não endereça esta vulnerabilidade. Para resolver este problema seria necessário algum tipo de detecção de intrusão ou uso de antivírus, pois o usuário teria a sua chave privada e a estaria usando sem saber do intruso. Uma outra premissa do trabalho é que a chave privada de cada AC nunca fica exposta ou é decifrada por um atacante.

No próximo capítulo serão apresentadas as análises matemáticas que comprovam a viabilidade da proposta, o cálculo das probabilidades de perda de pacotes, a comparação com outros trabalhos no que tange ao consumo de energia e a explicação das simulações executadas.

Capítulo 5

Resultados

Neste capítulo são apresentadas análises e experimentos com o objetivo de validar a arquitetura proposta e compará-la com outras propostas existentes na literatura. É descrito o ambiente de simulação, feita a análise matemática para verificação do número de mensagens de controle trocadas e do consumo de energia. Também é explicado como foram calculadas as probabilidades de perda de pacotes da rede baseado no trabalho [43].

5.1 Análise dos Parâmetros Utilizados no Protocolo

As constantes *Kno* e *Kac* propostas nesta dissertação podem ser definidas baseadas na probabilidade de perda de pacotes da rede para otimizar ainda mais o número de mensagens trocadas e para ajustar o protocolo às condições de carga da rede. Desta forma, a dissertação adota um modelo de erros e irá mostrar o comportamento das funções dentro deste modelo.

5.1.1 Modelo de Erros

O grande incômodo na comunicação sem fio são os erros. De forma geral, comunicação sobre um canal sem fio apresenta e está exposta a muito mais erros, de diferentes tipos e características, quando comparada a comunicações com fio.

Entender as características destes erros é muito importante por muitas razões: para avaliação de desempenho baseada em simulação de protocolos sem fio, ou para explorar o conhecimento sobre as características de erros sobre um protocolo que dinamicamente adapta seu comportamento a estas características alterando o tamanho de pacotes por exemplo. Em ambos os casos uma representação compacta das características dos erros, ou seja um modelo de erros do canal, é requerida.

Perdas de pacotes em redes ad hoc são muito mais freqüentes porque canais sem fio são sujeitos a diferentes erros de transmissão e colisão. Além disso a topologia pode mudar dinamicamente, o que introduz mais complexidade à rede. Um pacote pode ser perdido por conta de erros de transmissão,interferência, ausência de rota para o destino, canais sem conexão, congestionamento e colisões dentre outros motivos. Os efeitos destas causas estão fortemente associados com o contexto da rede (mobilidade dos nós, número de conexões, carga de tráfego etc). Até mesmo construir um modelo aproximado para analiticamente avaliar perda de pacotes é difícil [27] [21]. Em redes ad hoc erros de transmissão do canal sem fio, mobilidade e congestionamento tendem a ser as maiores causas de perda de pacotes [43]. A perda de pacotes em virtude de erros de transmissão é afetada pelas condições físicas do canal, o terreno onde a rede está localizada, dentre outras. Congestionamento em rede ocorre sempre que a demanda excede a capacidade máxima de um canal de comunicação. Já as colisões ocorrem especialmente quando múltiplos nós tentam acessar simultaneamente um meio de transmissão compartilhado. Mobilidade pode causar perda de pacotes de diferentes maneiras. Um pacote pode ser descartado na origem se uma rota para o destino não está disponível. Conforme um nó se movimenta podem ocorrer mudanças nas tabelas de rota e alguma rota pode deixar de existir.

Por estas razões esta dissertação se baseia nos dados contidos no trabalho [43] para estudar a probabilidade de perda de pacotes. Esta probabilidade inclui os erros causados pela mobilidade, erros de transmissão e congestionamento em cenários de carga leve, média e alta na rede. Em [43] os autores usaram cenários com conexões CBR, protocolos AODV, DSDV e DSR, tamanho de pacote de 512 bytes e taxa de transmissão de quatro pacotes/segundo ou 16Kbps. Estas condições são muito similares às simulações apresentadas nesta dissertação. Estas taxas de transmissão são encontradas em alguns modelos de simulação de redes ad hoc e também em aplicações militares [14].

Conforme mostrado na Figura 5.1, o parâmetro *Kno* é fundamental para diminuir a probabilidade de falha de comunicação em uma rodada do protocolo, quando o nó está tentando entrar em contato com uma AC para para estabelecer as autenticações dos protocolos ou para renovar seu certificado. Quanto maior a probabilidade de perda de pacotes, maior a probabilidade de falha na primeira rodada.

Da Figura 5.1 pode-se notar que quanto maior a probabilidade de perda de pacotes na rede, representada pelo eixo X, maior a probabilidade de falha em uma rodada (representada no eixo Y), como era de se esperar. Porém esta probabilidade de falha em

uma rodada se comporta de maneira diferente de acordo com as funções *Kno* escolhidas. A Seção 5.4 apresentará as conclusões acerca do comportamento desta função e de sua importância para os protocolos da proposta.

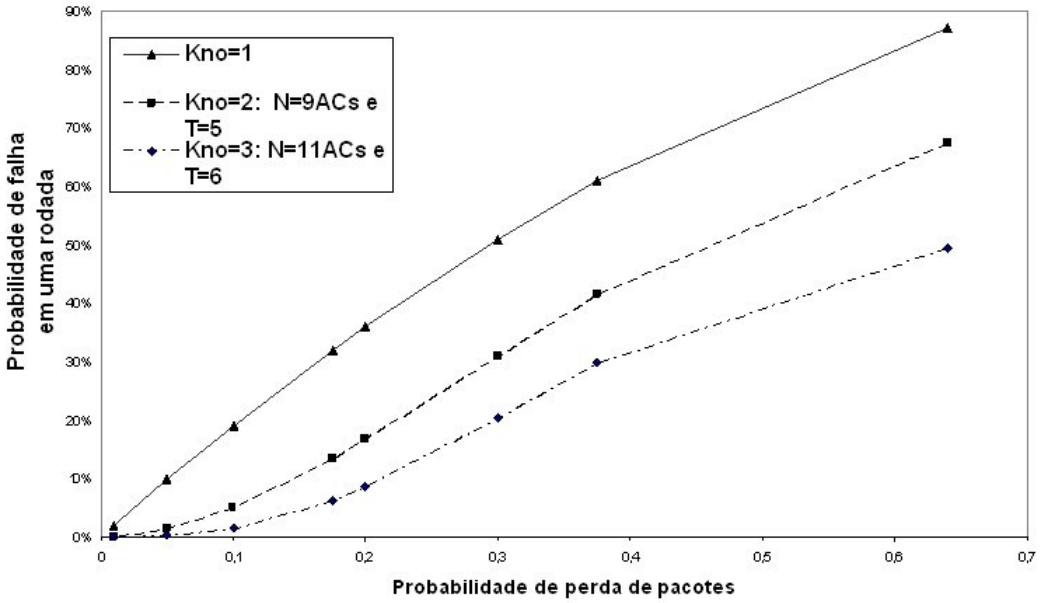


Figura 5.1: Variação da probabilidade média de perda de pacotes e a probabilidade de falha em uma rodada.

5.2 Ambiente de simulação

Foram feitas várias simulações para calcular a probabilidade de perda de pacotes na rede e para calcular o número de autenticações necessárias do protocolo de autenticação dos nós, variando o número de nós, o protocolo de roteamento, a velocidade dos nós e a quantidade de tráfego (número de conexões). Este cálculo será comparado com os valores do trabalho de [43]. Neste conjunto de simulações o número de ACs foi mantido fixo em nove devido ao tamanho dos cenários e ao alcance dos nós. A velocidade dos nós variou entre zero, 1,5m/s e 10m/s. O número de conexões variou entre 10, 15 e 20 conexões. O tempo das simulações foi de 300 segundos. O tamanho dos cenários variou de 670x670, 900x900 e 1000x1000m^2. O padrão de movimentação dos nós foi aleatório. Foram usadas as ferramentas "setdest" para geração dos cenários e o script "cbrgen.tcl" para geração do tráfego das simulações. Além disso, foram construídos scripts em linguagem "shell" e "awk" com o intuito de calcular as taxas de entrega de pacotes e também o número de autenticações das simulações. O arquivo de "trace" gerado pelo NS-2 (simulador) foi configurado para gerar pacotes no nível "MAC".

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

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

Do trabalho [43] foi utilizado o cenário de simulações com velocidade máxima de 20m/s, tempo de pausa igual a zero e tamanho do cenário de 1000x1000 com trinta nós. Foi então extraída a probabilidade de perda de pacotes (p) em três situações:

Carga leve (p=0,175) - Simulação com dez conexões;

Carga média (p=0,375) - Simulação com quinze conexões;

Carga alta (p=0,640) - Simulação com vinte conexões.

Estes valores descritos acima, o valor apresentado em [59] (que corresponde à probabilidade de perda fixa de 0,33) e os valores encontrados nas simulações desta dissertação (que serão mostrados na próxima seção) foram usados nas análises comparativas.

Em todas as simulações foram usados os protocolos DSR (*Dynamic Source Routing*) e DSDV (*Destination Sequenced Distance Vector*), protocolo MAC 802.11b, nós se comunicando numa taxa CBR (Constant Bit Rate) de 16 kbps com tamanho de pacote igual a 512 bytes. Esta taxa bem como o tamanho de pacote utilizado, são bastante utilizados para modelar protocolos de rede sem fio bem como em aplicações militares [14]. O modelo dos nós utilizado foi simulando um rádio Wave Lan Lucent 914 Mhz com antenas a 1,5m de distância do solo, alcance de 250 metros e taxa máxima de 2Mbps.

O objetivo das simulações foi validar parâmetros como probabilidade de perda de pacotes da rede e o número médio de autenticações necessárias para os nós se comunicarem usando os protocolos da proposta. Assim também é possível medir o consumo adicional de energia dos nós da rede e o número de mensagens de controle quando utilizados os protocolos propostos. Estes valores foram analisados e comparados com outras propostas existentes. Os protocolos em si não foram implementados no simulador NS-2. As simulações deram apoio às análises matemáticas feitas nesta dissertação.

Foram escolhidos três tipos de cenários para as simulações:

Cenário esparso - Formado por 30 nós de rede, 9 ACs com dimensão de 1000x1000m^2;

Cenário denso - Formado por 50 nós de rede, 9 ACs com dimensão de 900x900m^2;

Cenário muito denso - Formado por 50 nós de rede, 9 ACs com dimensão de 670x670m^2;

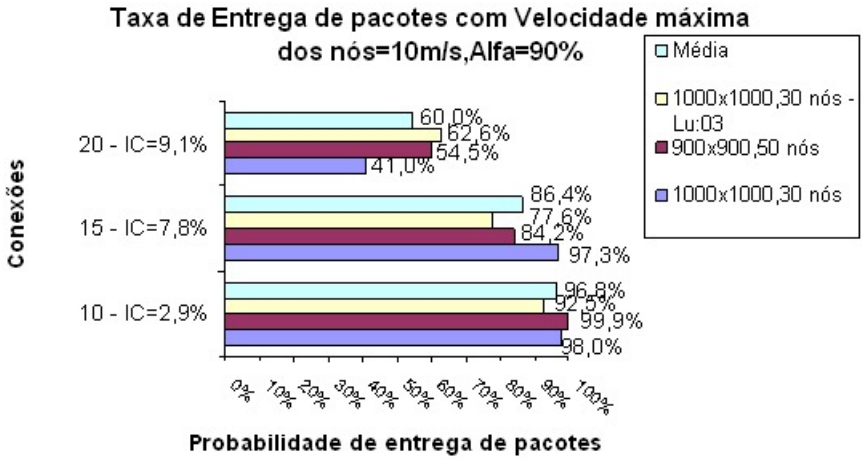
Nas Figuras 5.2(a) e 5.2(b) são mostradas as probabilidades de entrega de pacotes (em percentual) variando-se o número de conexões que corresponde à carga da rede. Estas figuras comparam os cenários esparso e denso com o cenário de 1000x1000 com 30 nós de [43]. A partir destes três cenários foi calculada a média das probabilidades de entrega de pacotes para cada valor de número de conexões. Para fins de análise foi usado um intervalo de confiança com o valor de alfa igual a 90%. A taxa de entrega de pacotes vai caindo conforme aumentam o número de conexões das simulações, ou seja, a carga da rede. Para os cenários analisados, nota-se que o protocolo DSR teve desempenho melhor quando são comparadas as médias das séries de 10,15 e 20 conexões, com as mesmas séries do protocolo DSDV.

As Figuras 5.3(a) a 5.3(f) e as Figuras 5.4(a) a 5.4(f) apresentam o número de autenticações e o número de autenticações/nó nos cenários citados acima, variando o número de conexões entre 10, 15 e 20 conexões por simulação. As séries das figuras mostram sempre os nós da rede se movendo com velocidades de 0, 1,5 e 10 m/s. A Figura 5.3 mostra o comportamento do protocolo reativo DSR, enquanto que a Figura 5.4 mostra o protocolo de roteamento pró-ativo DSDV. Podem ser notadas diferenças significativas no comportamento dos gráficos mostrados nas figuras, o que demonstra que um protocolo se ajusta melhor às necessidades de determinado cenário.

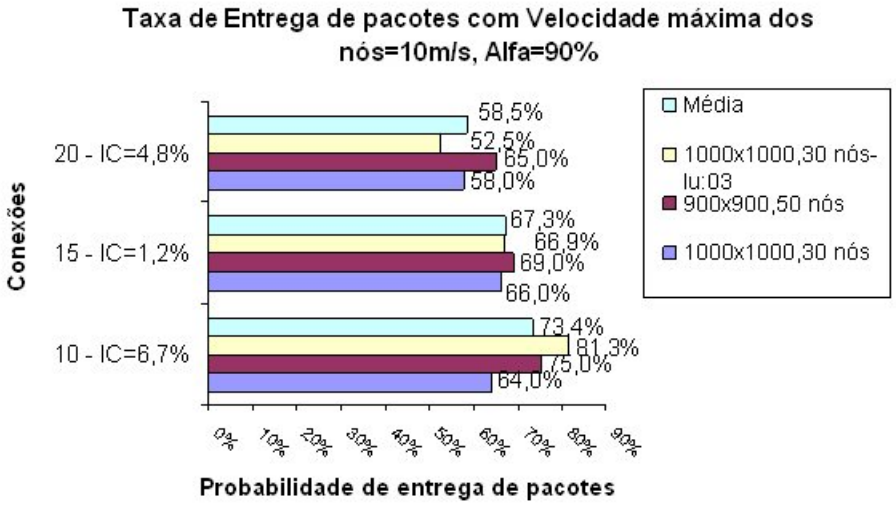
Pode-se notar que o protocolo DSDV é menos sujeito às variações do número de conexões das simulações, apresentando um número muito próximo de autenticações nos cenários de 10, 15 ou 20 conexões nas Figuras 5.4(a),5.4(c) e 5.4(f). O protocolo DSR apresentou uma variação muito grande no número de autenticações quando o número de conexões varia, principalmente no cenário de 900x900 das Figuras 5.3(c) e 5.3(d). A quantidade de autenticações das simulações também foi menor usando-se DSDV nos cenários esparso e denso, o que indica que este protocolo seria mais apropriado para ser usado nestes cenários. Apenas no cenário muito denso o protocolo DSR foi mais eficiente do que o DSDV. Nos outros cenários, com condições similares ao cenário apresentado em [43], o protocolo DSDV apresentou menor número de autenticações por nó e também menor número de autenticações totais na simulação.

Em todas as simulações ficou claro que quanto maior a velocidade de deslocamento

dos nós, maior foi o número de autenticações realizadas. Isto indica que a mobilidade realmente interfere em como os nós trocam mensagens de controle na rede, aumentando ou diminuindo o número de mensagens trocadas e por conseqüência o consumo de energia. Quanto maior a mobilidade, maior será o consumo de energia e maior a quantidade de mensagens de controle na rede.

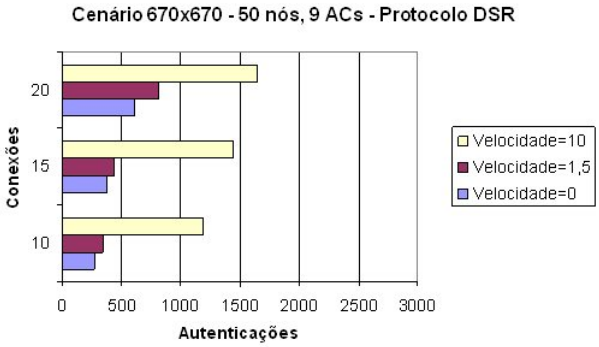


(a) Taxas de entrega de pacotes - DSR

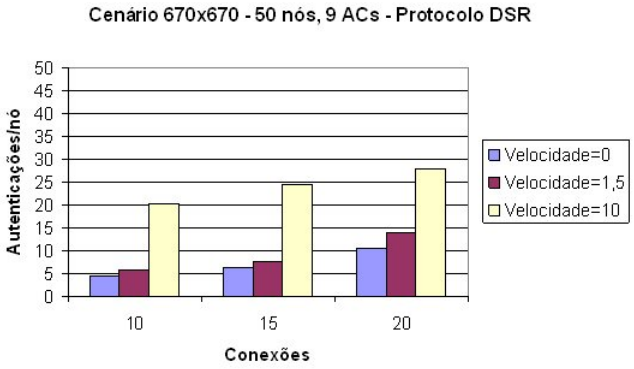


(b) Taxas de entrega de pacotes - DSDV

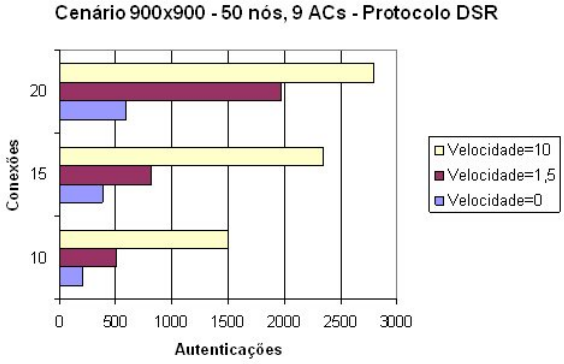
Figura 5.2: Taxas de entrega de pacotes. IC é o intervalo de confiança



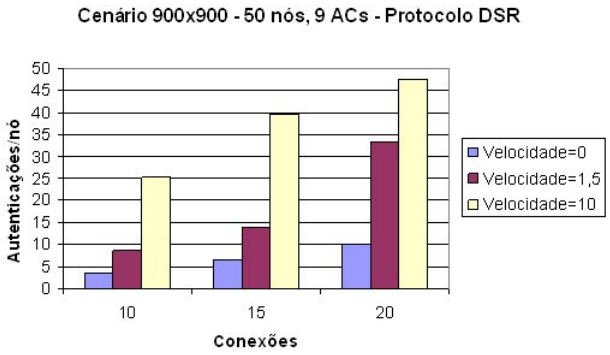
(a) Total de Autenticações,670X670,50 nós, 9ACs - DSR



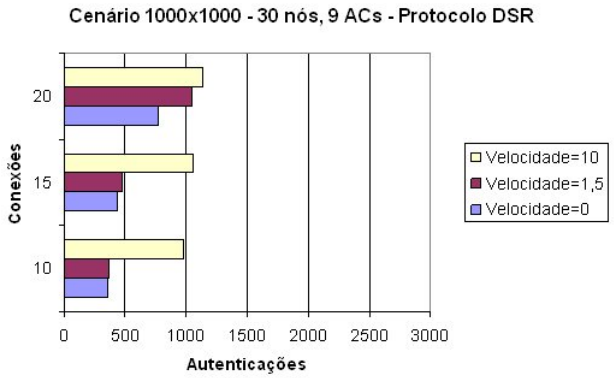
(b) Autenticações / nó,670X670,50 nós, 9ACs - DSR



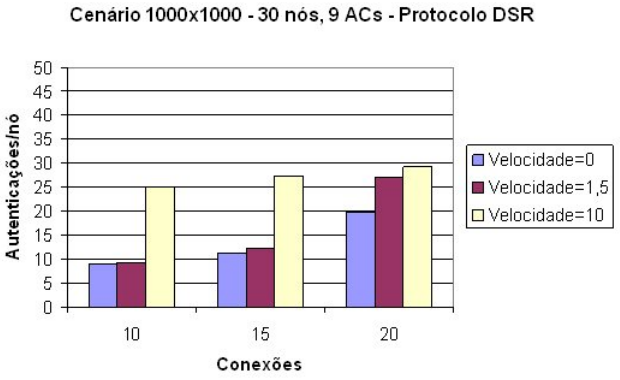
(c) Total de Autenticações,900X900,50 nós, 9ACs - DSR



(d) Autenticações / nó,900X900,50 nós, 9ACs - DSR

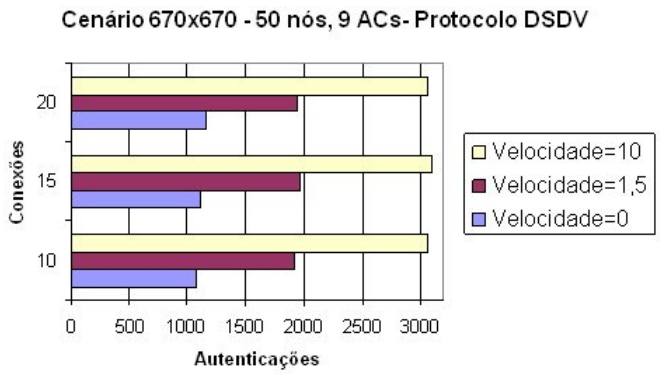


(e) Total de Autenticações,1000X1000,30 nós, 9ACs - DSR

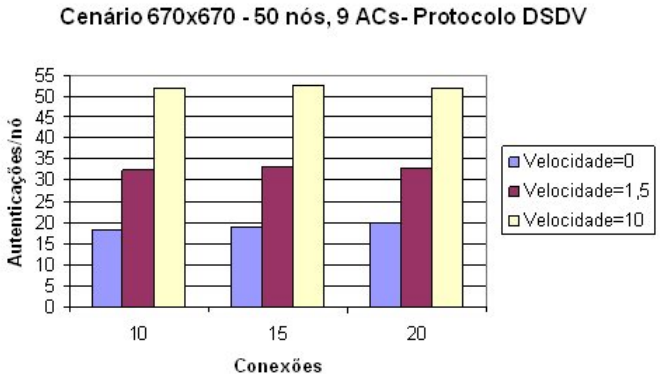


(f) Autenticações / nó,1000X1000,30 nós, 9ACs - DSR

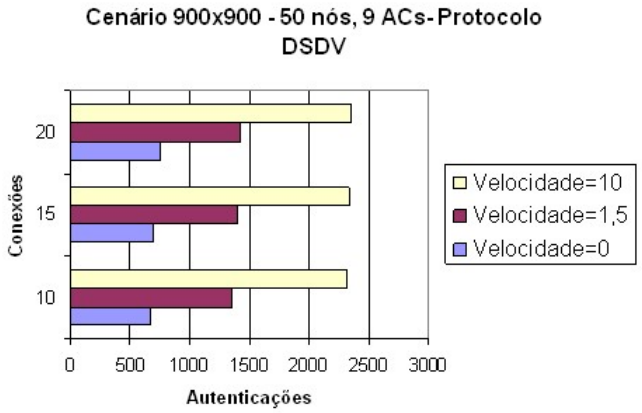
Figura 5.3: Autenticações - DSR



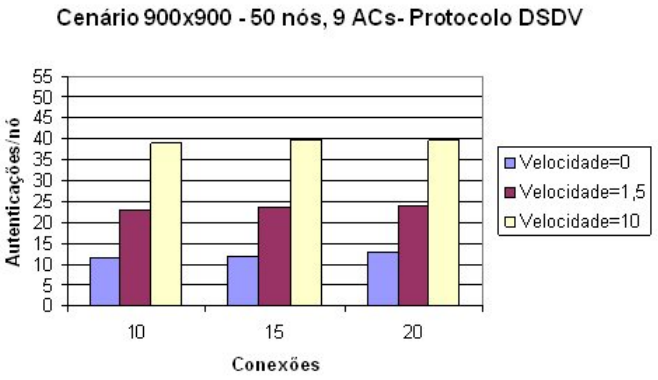
(a) Total de Autenticações,670X670,50 nós, 9ACs - DSDV



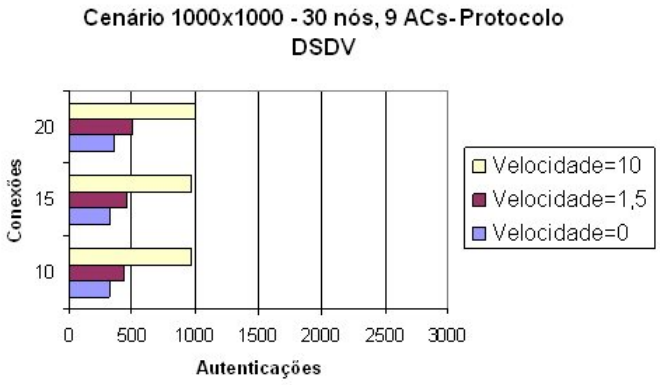
(b) Autenticações / nó,670X670,50 nós, 9ACs - DSDV



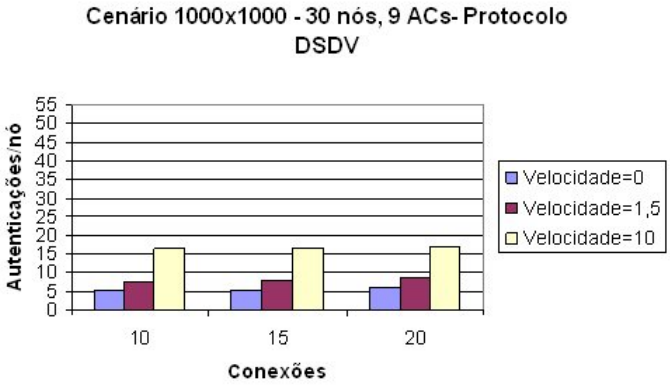
(c) Total de Autenticações,900X900,50 nós, 9ACs - DSDV



(d) Autenticações / nó,900X900,50 nós, 9ACs - DSDV



(e) Total de Autenticações,1000X1000,30 nós, 9ACs - DSDV



(f) Autenticações / nó,1000X1000,30 nós, 9ACs - DSDV

Figura 5.4: Autenticações - DSDV

5.3 Consumo de espaço em disco

Existe um consumo de espaço em disco causado pelas estruturas de controle e certificados propostos nesta dissertação. Por isso foram selecionados certificados X.509 com chave RSA de 1024 bits com o mínimo possível de campos perfazendo um total de 300 bytes (Figura 5.5). Este total não inclui os bits necessários para representar as regras de interoperabilidade de certificados usando codificação DER-ASN.1. Além disso, os nós armazenam os certificados recebidos dos outros nós e a LAC. As ACs conforme dito anteriormente, guardam também a lista de certificados revogados.

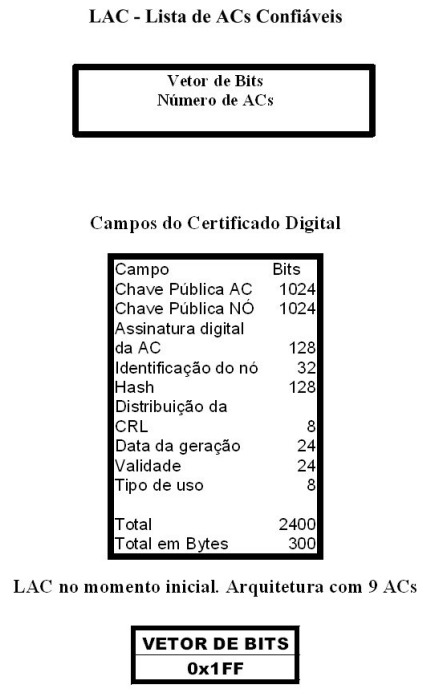


Figura 5.5: Lista de ACs Confiáveis e Campos do Certificado Digital.

Além disso, existe o consumo adicional de espaço que é a assinatura digital no fim do cabeçalho do protocolo de roteamento. São apenas 128 bits pois o protocolo MD-5 foi utilizado nas análises. Outro protocolo de *hash* mais novo (SHA ou SHA-1 por exemplo) poderia ser utilizado, porém o MD-5 foi escolhido por apresentar menor consumo de energia quando comparado com protocolos mais novos (Tabela 5.5).

5.4 Análise e Comparações

Uma métrica adotada nesta dissertação é a quantidade de mensagens de controle trocadas no processo de certificação e na autenticação dos nós. As análises foram divididas em duas

partes: número de mensagens trocadas durante a autenticação de dois nós e o número de mensagens para renovação de certificados. Uma outra métrica é o consumo de energia dos nós que será explicado na Seção 5.4.3.

No modelo proposto por [59] os autores inicialmente contaram a mensagem de difusão para renovação de um certificado como sendo apenas uma mensagem. Por outro lado, usando o modelo de difusão, as mensagens CREQ (mensagens enviadas para requisição de certificados) diminuem quando o número de ACs aumenta. Isso ocorre porque em cada simulação são solicitados mil certificados e o número de mensagens do tipo CREQ diminui à medida que o cenário se torna mais denso. Provavelmente este comportamento é explicado pelo maior número de ACs, o que diminui a quantidade de retransmissões de mensagens deste tipo. Deve-se levar em conta também que a mensagem de difusão é contada como sendo uma mensagem comum, sem diferença para uma mensagem "unicast". Além disso não existe uma AC combinadora no modelo por difusão, diferentemente do que é proposto por [63]. O próprio nó constrói a chave privada do serviço. Isto é um problema de segurança grave, pois o próprio cliente faz a verificação de assinatura da chave privada do serviço de certificação executando a interpolação polinomial. Além da exposição da chave privada para os nós, o nó é penalizado pois precisa executar o algoritmo de interpolação.

Esta dissertação propõe comunicação "unicast" onde o número de mensagens CREQ aumenta à medida que o número de ACs aumenta. Além disso, os nós não necessitam executar a interpolação polinomial e não podem conhecer a chave privada do serviço ICPAH, pois existe o papel de AC combinadora que é desempenhado por uma das ACs da infra-estrutura.

Para a primeira comparação entre os métodos serão usados dois tipos de parâmetros *Kno*. No primeiro, não linear, $Kno = \lceil N/Log_2 2T \rceil$ e no segundo, constante, Kno=6. Para o parâmetro *Kac* será feita a análise levando-se em conta uma função linear com $K_{ac} = 3/2 \times (T-1)$. Em ambos os casos, N é o número de ACs e T é o limite criptográfico. A relação entre T e N é N=2T-1 de [60]. Respeitando o limite explicado no capítulo anterior para comparar com o método que usa difusão, onde $2/3 \times Kno > 1$ pode-se verificar que:

$$
\begin{aligned}
Kno &> 3/2; \\
N/Log2T &> 3/2;\ \textit{Substituindo Kno} \\
2N/3 &> Log_2 2T;\ \textit{Substituindo T por } (N+1)/2 \\
2N/3 &> Log_2 2(N+1)/2;\ \textit{Fazendo 2 elevado a ambos os lados temos} \\
2^{2N/3} &> N+1; \\
2^{2/3^N} &> N+1;\ \textit{Rearrumando o expoente} \\
1,59^N &> N+1;
\end{aligned}
$$

A inequação é verdadeira $\forall N,\ tal\ que\ N \geq 3$.

Para qualquer outro valor de Kno constante, basta que $Kno \geq 3/2$.

Para a constante Kac, análise similar deve ser feita. A constante Kac deve gerar mensagens suficientes para compensar as perdas da rede. Assim, Kac pode variar conforme a probabilidade de perda de pacotes. Para esta primeira comparação com o método que utiliza difusão, $2/3 \times Kac \geq T-1$. Sendo assim, será usada o parâmetro $Kac = 3/2 \times (T-1)$ pois:

$$
\begin{aligned}
2/3 \times Kac &\geq T-1; \\
Kac &\geq 3/2 \times (T-1).
\end{aligned}
$$

A escolha deste valor para Kac visa gerar a menor quantidade possível de mensagens de controle para o funcionamento dos protocolos.

Para melhorar o desempenho das funções, será usada a probabilidade de perda em carga média da rede mostrada na Figura 5.2. Para esta análise é importante receber uma resposta com sucesso do serviço ICPAH na primeira tentativa ou rodada. Esta premissa é aplicável à verificação de assinatura e à renovação de certificados. A seguinte inequação deve ser respeitada:

$$
Kno\ -(Kno\ \times P1r) \geq 1 \tag{5.1}
$$

(Sendo P1r denota a probabilidade de falha na primeira tentativa.)

Em outras palavras, o número de mensagens enviadas (Kno) menos o número de respostas não recebidas ($Kno \times P1r$) deve ser maior do que um.

Isso porque o nó solicitante deve receber pelo menos uma resposta do serviço ICPAH. Observando os gráficos das Figuras 5.1 e 5.6, será usado Kno igual a pelo menos 2 (dois) e p=0,327, que é a probabilidade média obtida conforme Figura 5.2 (a qual é próxima a probabilidade obtida em [59] que é igual a 1/3). O valor de P1r é calculado como se segue.

Sendo X a variável que denota a probabilidade de perda de pacotes quando o nó envia mensagens para as ACs e Y a variável que denota a probabilidade de perda de pacotes quando as ACs respondem para os nós:

$$P1r = (P(X = 0) \times P(Y = 2)) + (P(X = 1) \times P(Y = 1)) + P(X = 2) \tag{5.2}$$

P(X) e P(Y) seguem distribuições binomiais de probabilidade pois o evento (ocorrência de erro de transmissão) pode apresentar dois estados: ocorrer ou não ocorrer.

$$P(X = k)\ ou\ P(Y = k) = \binom{m}{k} \times p^k \times (1 - p)^{m-k} \tag{5.3}$$

onde p é a probabilidade média de perda da rede retirada das simulações e m=Kno=2.

Desta forma P1r=0,34 e $2 - (2 \times 0.34) = 1.32 \geq 1$ respeitando 5.1).

5.4.1 Quantidade de Mensagens do Protocolo de Autenticação

O total de mensagens trocadas no protocolo de autenticação não poderá ser comparado com nenhum trabalho, visto que não foram encontrados trabalhos que tenham feito este tipo de análise. De qualquer forma, um protocolo de roteamento no estabelecimento de rotas, possui mensagens do tipo "Requisição de Rota" (RouteRequest) e mensagens do tipo "Resposta de Rota" (RouteReply). A proposta é utilizar o Agente de segurança (sub-camada de rede)para autenticar as comunicações antes das mensagens de roteamento começarem a trafegar entre os nós da rede.

No estabelecimento de autenticação entre dois nós quaisquer da rede, são trocadas as seguintes mensagens:

1. Transmissor envia uma mensagem de AuthenticationRequest para o receptor.
2. Receptor envia *Kno* mensagens VerificaCert para o ICPAH a fim de verificar a chave pública do transmissor.
3. Serviço ICPAH envia $(1-p)^2 \times Kno$ mensagens para o nó requisitante, sendo p a probabilidade de perda de pacotes da rede. Isso ocorre porque o serviço ICPAH só recebe $(1-p) \times Kno$ mensagens das *Kno* mensagens enviadas pelo nó. O nó só deverá receber então $(1-p)^2 \times Kno$ devido às perdas da rede.
4. Receptor envia o *nonce*, a chave simétrica e o AuthenticationReply cifrados numa mensagem.
5. O transmissor verifica a chave pública do receptor no serviço ICPAH. Mesma operação realizada no passo 2.
6. Resposta do serviço ICPAH. Mesma operação realizada no passo 3.
7. Transmissor envia Ack para o receptor.

Sendo assim, são trocadas em cada estabelecimento de autenticação entre dois nós um total de mensagens igual a

$$MensagensAutenticacao = 3 + 2 \times (Kno + (1-p)^2 \times Kno). \tag{5.4}$$

Como no processo de autenticação existem duas autenticações pois os nós se autenticam mutuamente, o total de mensagens trocadas é igual a (AUTENT ×Mensagens Autenticacao)/2, sendo AUTENT o número total de autenticações da rede. As tabelas abaixo mostram o número de mensagens trocadas para autenticação nos diversos cenários de simulação, utilizando os valores das Figuras 5.2 e 5.3. As Figuras 5.2 e 5.3 mostram o número de autenticações das simulações executadas para esta dissertação.

O número de mensagens adicionais em cada autenticação é $Kno + (1-p)^2 \times Kno$ conforme dito anteriormente. Mais uma vez é importante utilizar uma constante *Kno* bem ajustada e levar em conta a probabilidade de perda de pacotes da rede.

As Tabelas 5.1 e 5.2 mostram a comparação entre o número de mensagens de roteamento trocadas no protocolo de autenticação quando *Kno* assume os valores N (método de difusão), $\lceil N/Log_2 2T \rceil$ e um valor constante igual a três. A probabilidade de perda de

pacotes foi mantida fixa e igual a 1/3 (conforme o método de difusão de [59]). Variando-se o número de ACs da infra-estrutura obtém-se o número de mensagens para cada cenário.

Pode-se concluir pelas Tabelas 5.1 e 5.2 que o número de mensagens do protocolo de autenticação pode ser otimizado variando *Kno* dependendo da probabilidade de perdas da rede. Para tirar proveito desta variação é preciso conhecer bem o padrão de tráfego da rede ajustando as funções para ter melhor desempenho. Além disso o número de mensagens cresce numa taxa menor quando comparado com o acréscimo do número de ACs ou se mantém constante quando *Kno* é constante.

Tabela 5.1: Número de Mensagens Trocadas em cada Autenticação, p=0,32

Número de ACs	A) N (DIFUSÃO)	B) $\lceil N/Log_2 2T \rceil$	C) 3
15	46,47	13,87	11,69
30	89,93	24,73	11,69
50	147,89	39,22	11,69

Tabela 5.2: Comparação do ganho entre diferentes *Kno*

Número de ACs	Método B sobre A	Método C sobre A	Método C sobre B
15	70,16	74,84	15,67
30	72,50	87,00	52,72
50	73,48	92,09	70,19

5.4.2 Quantidade de Mensagens na Renovação de Certificados

Toda requisição de renovação de certificado apresenta os seguintes passos:

- *Kno* mensagens CertRenova enviadas;

- Pelo menos $(1-p)^2 \times Kno$ ACKS recebidos sendo "p"a probabilidade média de perda de pacotes, pois são recebidas $(1-p) \times Kno$ pelo serviço ICPAH e recebidas então $(1-p) \times (1-p) \times Kno$ mensagens de ACK pelo nó;

- Uma mensagem CERT;

- *Kac*AssinaParcial mensagens enviadas pela AC combinadora;

- *Kac* CertParcial mensagens enviadas pelas ACs para a AC combinadora;

- Uma mensagem FinalCert para o nó que requisitou a renovação de certificado.

Para comparar com o cenário de [59], serão utilizados os parâmetros N=19, T=10, usada a equivalência N=2T-1 de [60], 1000 requisições e p=0,3 ou 1/3. Além disso, temos que:

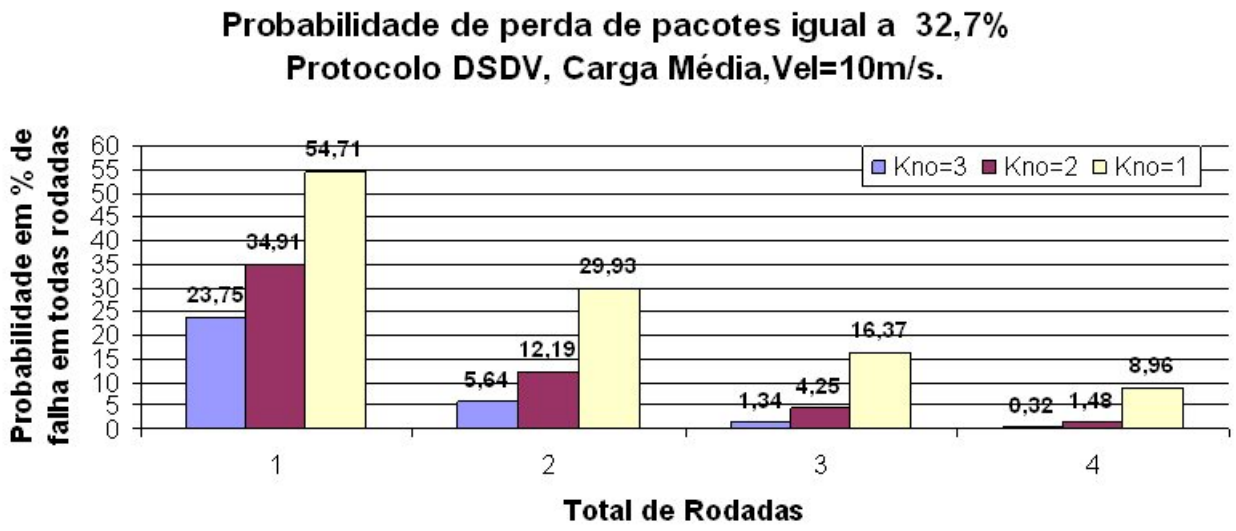


Figura 5.6: Probabilidade que nenhuma AC seja contatada depois de cada rodada - Probabilidade de perda de pacotes igual a 32,7%.

$Kno = \lceil N/Log_2 2T \rceil \rightarrow 4$ e

$Kac = 3/2 \times (T-1) \rightarrow 13{,}5$.

Total: $(4 + 1,81 + 13,5 + 13,5) \times 1000 = 32810$ mensagens.

Este resultado quando comparado com as 119125 mensagens CREQ em [59] que usa o método de difusão, apresenta um ganho de aproximadamente 73%. Com o método proposto nesta dissertação, apenas uma AC é necessária para responder ao nó requisitante. Usando o mesmo princípio, 1/3 de perda de pacotes para selecionar uma entre as ACs contatadas em apenas uma rodada, a chance de sucesso (1-P1r) é de 90% pois

$$P1r = (P(X=0) \times P(Y=4)) + (P(X=1) \times P(Y=3)) + (P(X=2) \times P(Y=2)) + (P(X=3) \times P(Y=1)) + P(X=4)) \tag{5.5}$$

Fazendo a mesma análise para o número de mensagens recebidas pelos nós (chamadas CREPs em [59]):

Total: $(2/3 \times 4 + 1) \times 1000 = 3666$ mensagens.

Comparando com as 29776 mensagens recebidas pelos nós em [59] o ganho é de aproximadamente 88%.

Estes resultados ainda podem ser melhorados refinando-se os valores de *Kno* e *Kac*. Sendo o resultado de *Kno* igual a 3 (três), utilizando-se a probabilidade média de perda da rede "p" igual a 0,32, temos que a probabilidade de falha na primeira rodada é de 12%. Este valor será considerado aceitável para a análise, pois em 88% dos casos não haverá necessidade de rodada adicional para um nó contatar o serviço ICPAH quando da renovação do seu certificados. Para *Kno* podemos considerar que a AC combinadora deve compensar as perdas da rede, pois como deve receber T-1 mensagens com assinaturas parciais das outras ACs, deve enviar

$$Kac = p^2 \times (T-1) + p \times (T-1) + (T-1) \tag{5.6}$$

mensagens, sendo $p \times (T-1)$ o número provável de mensagens perdidas enviadas pela AC combinadora para as outras ACs, $p^2 \times (T-1)$ o número provável de mensagens perdidas enviadas pelas ACs com as assinaturas parciais como resposta para a AC combinadora.

Refazendo os mesmos cálculos com os valores de *Kno*=3, p=0,32 e $Kac=p^2 \times (T-1) + p \times (T-1) + (T-1)$ temos:

CREQ - $(3 + 1,39 + 11 + 11) \times 1000 = 26390$ mensagens.

CREP - $(0,68 \times 3) + 1) \times 1000 = 3040$ mensagens.

Comparando novamente com as 29776 mensagens recebidas o ganho agora é de 90% e com as 119125 mensagens enviadas o ganho é de 77%. Isso tudo com uma garantia de 88% de chance de não serem necessárias novas rodadas do algoritmo devido às perdas da rede.

A Tabela 5.3 compara os métodos propostos com o método usado em [59]. A Tabela 5.4 mostra o ganho ou perda percentual (indicada pelo sinal negativo) entre os métodos. Fica claro que *Kno* e *Kac* podem ser ajustadas baseadas na probabilidade de perda de pacotes e na carga da rede. Em baixa carga, $Kno = \lceil N/Log_2 2T \rceil$ e $Kac = 3/2 \times (T-1)$ são menos eficientes porque os nós enviam mensagens desnecessárias. Em alta carga porém, os nós enviam um número inferior de mensagens ao necessário para garantir sucesso na primeira tentativa de comunicação com o serviço ICPAH. Uma nova rodada do protocolo pode causar atrasos e retransmissões. A conclusão é que $Kno = \lceil N/Log_2 2T \rceil$, $Kno = 3 \times [(N+1)/T]$ ou simplesmente *Kno*= 3 e $Kac = 3/2 \times (T-1)$ são boas estimativas para situações de carga média da rede.

A quantidade de mensagens de controle tipo CREQ aumenta de forma linear conforme aumenta o número de ACs quando $Kno = \lceil N/Log_2 2T \rceil$ ou mesmo quando $Kno = 3 \times$

$[(N+1)/T]$. Já as mensagens tipo CREP variam menos mesmo utilizando a função linear $Kac = 3/2 \times (T-1)$. Quando Kac é mantido contante as mensagens CREP também se mantém contantes. Isso indica que a proposta é escalável pois o número de mensagens aumenta linearmente com o número de ACs. Mais uma vez é preciso ressaltar que não é necessário possuir ACs dispostas de maneira a cobrir todo o cenário. O número de ACs só deve ser aumentado após se perceber que muitas ACs confiáveis só são alcançáveis pelos nós após um número elevado de saltos, pois esta condição atrasaria a execução dos protocolos gerando atrasos e retransmissões.

Na Tabela 5.3 são considerados os métodos de transmissão para renovação do certificado, sendo:

- MÉTODO A → Difusão;
- MÉTODO B → $Kno = N/Log_2 2T,\ Kac = 3 \times (T-1)/2$;
- MÉTODO C → $Kno = 3, p = 0,32, Kac = p^2 \times (T-1) + p \times (T-1) + (T-1)$;

Tabela 5.3: Comparação entre os métodos em carga média

Número de ACs	MÉTODO A		MÉTODO B		MÉTODO C	
	MSGS CREQ	MSGS CREP	MSGS CREQ	MSGS CREP	MSGS CREQ	MSGS CREP
15	134.694	14.953	25.000	3.000	26.390	3.100
30	119.125	29.776	85.000	5.000	46.390	3.100
50	98.962	49.447	81.000	6.333	76.390	3.100

Tabela 5.4: Porcentagem de ganho em carga média

Número de ACs	Método B sobre Método A		Método C sobre Método A		Método C sobre Método B	
	MSGS CREQ	MSGS CREP	MSGS CREQ	MSGS CREP	MSGS CREQ	MSGS CREP
15	81,44	79,94	80,41	79,27	-5,56	-3,33
30	28,65	83,21	61,06	89,59	45,42	38,00
50	18,15	87,19	22,81	93,73	5,69	51,05

5.4.3 Consumo de Energia

O *overhead* adicionado pelo uso de chaves assimétricas para a troca de mensagens deve ser considerado principalmente em redes ad hoc. Este consumo e o consumo provocado pelo uso de chaves públicas influencia o consumo de energia dos nós. Este consumo é um fator

decisivo em redes ad hoc porque os nós necessitam de baterias para seu funcionamento. Nas Tabelas 5.5, 5.6 e 5.7 são mostrados os valores de consumo de energia quando são usados alguns algoritmos simétricos, assimétricos e funções *hash*.

Tabela 5.5: Energia gasta em algoritmos de HASH. Fonte[36]

Algoritmo	MD2	MD4	MD5	SHA	SHA1
ENERGIA (μJB)	4,12	0,52	0,59	0,76	1,16

O consumo está relacionado também à mobilidade dos nós. Se os nós têm pouca mobilidade, então menor será o consumo. Por outro lado, se os nós constantemente trocarem pacotes de estabelecimento de novas rotas, então maior será o consumo. Este comportamento é mostrado claramente nas Figuras 5.2 e 5.3 que mostram que quanto maior a velocidade dos nós, nos mesmos cenários, maior o número de autenticações feitas pelos nós para estabelecer comunicação.

Tabela 5.6: Consumo de Energia em algoritmos assimétricos para uma entrada de 1024 bits. A assinatura RSA é utilizada cifrando e decifrando a mensagem diretamente sem necessidade de gerar o hash da mesma. Usada quando a mensagem é pequena. Fonte[36]

Algoritmo	Tam. Chave (Bits)	Geração(mJ)	Assinatura(mJ)	Verificação(mJ)
RSA	1024	270,13	546,50	15,97
DSA	1024	293,20	313,60	338,02
ECDSA	163	226,65	134,20	196,23

Tomando por base os dados da Tabela 5.6, chega-se a conclusão que, utilizando-se o protocolo RSA com chave de 1024 bits, o consumo para cifar um byte é de **4,27mJ** e o consumo para decifrar um byte é de **0,12mJ** fazendo uma regra de três simples.

Segundo [36], um sensor típico utilizando um chip Motorola MC68328 consome 21,5mJ e 14,3mJ para transmitir e receber respectivamente 1024 bits. Estes valores também serão usados no cálculo adicional de energia necessário para os protocolos propostos. Um resumo dos valores que serão utilizados daqui para frente encontra-se na Tabela 5.8

Tabela 5.7: Consumo de Energia em algoritmos simétricos. Fonte[36]

	DES	3DES	CAST	AES	RC5
KEY SETUP (μJ)	27,53	87,04	37,63	7,87	66,54
Enc/Dec (μJ/byte)	2,08	6,04	1,47	1,21	0,79

Por esta razão, esta dissertação propõe a diminuição do número de mensagens de controle trocadas entre os nós e pela infra-estrutura de ICP. Nas Tabelas 5.9 e 5.10 são mostrados os valores do consumo de energia nos processos de autenticação e renovação de

Tabela 5.8: Resumo dos valores de consumo de energia que serão usados para calcular o consumo adicional dos protocolos propostos.

Gerar Chave RSA/1024	270,3mJ
Cifrar RSA/1024	4,27mJ/Byte
Decifrar RSA/1024	0,12mJ/Byte
Gerar Chave AES-128	7,87μJ
Cifrar/Decifrar AES-128	1,21μJ/Byte
MD-5	0,59μJ/Byte
Transmissão	0,17mJ/Byte
Recepção	0,11mJ/Byte

certificados. Este consumo foi calculado usando os algoritmos RSA-1024, AES-128 e MD-5 (que produz um hash de 128 bits). Para calcular o consumo de energia na autenticação e na renovação de certificados, foi considerado que os nós devem executar a função *hash* sobre o certificado, o qual tem 300 bytes (0,177mJ), assinar e verificar as mensagens usando o protocolo RSA, além de cifrar e decifrar as mensagens necessárias com a chave simétrica de sessão.

No consumo de energia para autenticação foi considerado que os nós devem se autenticar mutuamente usando o protocolo de autenticação mostrado na Figura 4.2 com N=15 (número de ACs) e T=8 (limite criptográfico). A constante *Kno* utilizada foi $Kno = \lceil N/Log_2 2T \rceil$ e a constante *Kac* foi $Kac = 3 \times (T-1)/2$. Conseqüentemente os passos abaixo descritos são executados:

1. O transmissor deve assinar digitalmente o pacote de requisição de autenticação (1 byte) e enviá-lo junto com seu certificado digital que contém sua chave pública. Neste caso não será gerado *hash*, apenas a assinatura RSA por se tratar de apenas um byte;

2. O receptor irá contatar $Kno = \lceil N/Log_2 2T \rceil$ ACs para verificar e validar a chave pública do outro nó;

3. As ACs do serviço ICPAH irão verificar a chave pública recebida. Depois irão enviar a resposta para o nó receptor;

4. O receptor irá receber as respostas do serviço de criptografia;

5. O nó receptor deve cifrar o *nonce* gerado (1 byte), a chave simétrica escolhida (128 bits ou 16 bytes) e o pacote de resposta de autenticação (1 byte) com a chave pública do transmissor. O receptor deve cifrar o *hashing* da mensagem com sua chave privada;

6. O transmissor irá receber a mensagem;
7. O transmissor irá contatar $Kno = \lceil N/Log_2 2T \rceil$ ACs para verificar e validar a chave pública do outro nó;
8. As ACs do serviço ICPAH irão verificar a chave pública recebida. Depois irão enviar a resposta para o nó receptor;
9. O transmissor irá receber as respostas do serviço de criptografia;
10. O transmissor decifra o *nonce*, a chave simétrica e o pacote Authentication Reply com sua chave privada e decifra o hashing com a chave pública do receptor;
11. O transmissor cifra o reconhecimento (ACK) com a chave simétrica de sessão e o envia para o receptor;

O total de energia gasta pelo nó transmissor é de 215,63mJ , o total gasto pelo receptor é de 154,28mJ e o total gasto pelo serviço ICPAH é de 1,36mJ. O detalhamento destes cálculos pode ser observado no Apêndice B.

Os itens 1, 6, 7, 9, 10 e 11 correspondem aos passos que levam o transmissor (o nó que enviou a requisição de rota) a gastar energia adicional com o protocolo. Os itens 2, 4 e 5 são os passos que fazem com que o receptor da requisição gaste energia adicional com o protocolo. Os passos 3 e 8 indicam que o serviço ICPAH ou o conjunto de ACs é que gasta energia adicional com o protocolo.

Para o processo de renovação de certificados com N=15 (número de ACs) e T=8 (limite criptográfico) tem-se:

1. Geração do novo par de chaves pelo nó;
2. Envio da mensagem CertRenova pelo nó para *Kno* ACs;
3. Recebimento das mensgens pelo serviço ICPAH
4. Resposta do Serviço ICPAH, envio dos ACKs ($2/3 \times \lceil N/Log_2 2T \rceil$);
5. $2/3\times$ a quantidade de ACKs recebidos no nó;
6. Uma mensagem Cert enviada para o ICPAH;
7. AC combinadora recebe mensagem Cert;

8. AC combinadora envia AssinaParcial para *Kac* ACs;
9. ACs recebem o certificado e executam assinatura parcial;
10. ACs enviam resposta para a AC combinadora;
11. A AC combinadora recebe as respostas das outras ACs;
12. A AC combinadora calcula chave privada do ICPAH e assina certificado;
13. A AC combinadora envia certificado (FinalCert);
14. O nó requisitante recebe FinalCert.

O total de energia gasto pelo nó requisitante é de 354,89mJ e o total gasto pelo serviço ICPAH (conjunto de ACs) é 1334,62mJ. Considerando que neste caso, *Kno*=3 e *Kac*=10, a média de consumo de energia por AC é de 1334,62/13=102,66mJ. A Tabela 5.9 mostra os diferentes consumos de energia quando variam o número de ACs da rede. Mais uma vez a escolha de *Kno* e *Kac* influenciam também no consumo total de energia da solução.

Para calcular a energia gasta com a troca de mensagens é suficiente assumir que as mensagens não serão cifradas gerando um consumo constante (*hashing* do cabeçalho da mensagem mais a cifragem do *hashing* com a chave simétrica de sessão):

1. 20 bytes (tamanho usual do cabeçalho no DSDV) $\times 0,59 \rightarrow 11,80\mu J$;
2. Cifragem do *hashing* com a chave simétrica: 16 $bytes \times 1,21\mu J \rightarrow 19,36\mu J$;

Somando (1) e (2) chega-se ao total de $31,16\mu J$ ou 0,031mJ. O consumo parece baixo, porém quanto maior a quantidade de mensagens de controle usadas, maior será o consumo.

Tabela 5.9: Energia gasta pelo serviço ICPAH para renovação de certificados com $Kno = \lceil N/Log_2 2T \rceil$ *e* $Kac = 3 \times (T-1)/2$

	N=15, T=8	N=30, T=16	N=50,T=26
Nó Requisitante	0,35J	0,35J	0,35J
ICPAH	1,36J	2,72J	3,63J
Média por AC	0,10J	0,08J	0,08J

Como era de se esperar a operação de renovação de certificado é mais custosa porém ocorre com freqüência muito menor do que as operações de troca de mensagens de roteamento e também as de autenticação. Comparando as informações das Tabelas 5.10 e 5.11 podemos ver o ganho do método proposto quando comparado ao método de difusão. Os

Tabela 5.10: Energia gasta por um nó em cada operação - $Kno = \lceil N/Log_2 2T \rceil$ *e* $Kac = 3 \times (T-1)/2$

	N=15, T=8	N=30, T=16	N=50,T=26
Renovação	0,35J	0,35J	0,35J
Estabelecimento de autenticação (transmissor)	0,21J	0,36J	0,47J
Estabelecimento de autenticação (receptor)	0,15J	0,30J	0,41J
Troca de Mensagens de Roteamento	0,03mJ	0,03mJ	0,03mJ

Tabela 5.11: Energia gasta por um nó em cada operação - *Kno*= N = DIFUSÃO

	N=15, T=8	N=30, T=16	N=50,T=26
Renovação	0,35J	0,35J	0,35J
Estabelecimento de autenticação (transmissor)	0,83J	1,0J	2,63J
Estabelecimento de autenticação (receptor)	0,77J	1,54J	2,57J
Troca de Mensagens de Roteamento	0,03mJ	0,03mJ	0,03mJ

valores foram calculados utilizando-se a mesma probabilidade de perda de pacotes fixa igual a 2/3.

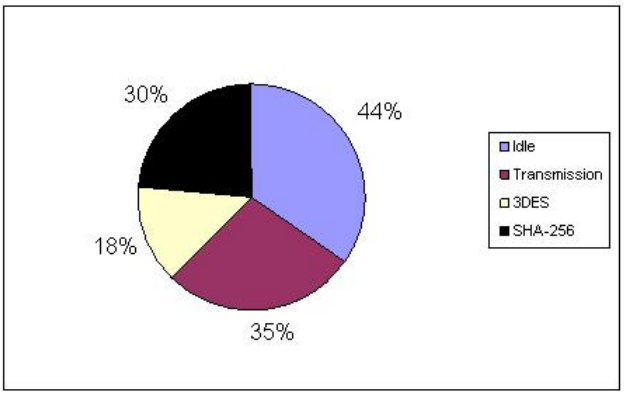


Figura 5.7: Distribuição da Energia gasta para transmitir 64Kbytes - 3DES.

A maioria das baterias de notebooks são do tipo AA com tecnologia "Lithium Ion" e possuem atualmente capacidade de 4.000mAh(Mili-Amper/hora) [1]. Estas baterias podem trabalhar em até 14,8 Volts totalizando uma carga total de 59.200 Joules ou 59,2KJ ($4.000mAh \times 14,8Volts$). Já os sensores típicos segundo [36] possuem capacidade de energia de aproximadamente 26KJ. A autonomia de uso varia muito de acordo com as aplicações que rodam nas máquinas. No caso do uso típico de um notebook a autonomia da bateria é de 4,5 horas. Isso mostra que o uso de certificados digitais e algoritmos criptográficos deve ser feito com cuidado. Porém com o uso da técnica proposta nesta dissertação este custo cai bastante. Por exemplo: Com N=9 ACs e T=5 (limite criptográfico),

[1]Dados retirados de um notebook Sony Vaio

o consumo de energia para cada nó em uma autenticação é na média de 0,13J. Utilizando como base o cenário esparso (1000x1000) com 30 nós, protocolo DSDV e calculando-se o percentual de energia gasto pelos nós no processo de autenticação (levando-se em conta o número médio de autenticações por nó neste cenário), chega-se a 0,01% sobre o consumo total de um sensor (26KJ), somente quando a velocidade é de 10m/s. Com velocidades de deslocamento iguais a 0 e 1,5m/s o consumo foi desprezível (menor que 0,01%). Para os outros cenários desta proposta os valores não se alteram muito chegando ao máximo de 0,03% do consumo máximo de um sensor (26KJ).

Em conclusão, o uso dos algoritmos criptográficos não é impeditivo para realizar comunicação segura quando o mesmo é limitado e otimizado. A Figura 5.7 mostra uma distribuição média de energia consumida por um nó de rede para transmitir 64Kbytes utilizando o algoritmo simétrico 3DES. Além disso, existem propostas como em [36] para otimizar o consumo de energia de nós sem fio, mas estas propostas não são o foco desta análise. Se os nós têm pouco movimento, menor é o consumo de energia relacionado à criptografia. Por outro lado se os nós trocam informações de rota constantemente, maior é o consumo adicionado. Por isso esta dissertação propõe redução no número de mensagens de controle trocadas entre os nós.

Capítulo 6

Conclusão e Trabalhos Futuros

Este trabalho abordou alguns problemas de segurança em redes ad hoc causados por diferentes tipos de ataques conhecidos. Especificamente foi proposta uma infra-estrutura de chaves públicas distribuída combinada com o uso de limite criptográfico para autenticação de nós móveis. Através de uma inundação restrita e controlada de pedidos de autenticação e de renovação de certificados, este trabalho mostrou que é possível diminuir a quantidade de mensagens de controle trocadas entre os nós de uma rede ad hoc na implementação de algoritmos que visam dar segurança na comunicação entre os nós desta rede. Em alguns casos, consegue-se uma diminuição de até 92% no número destas mensagens de controle conforme mostram as Tabelas 5.1 e 5.2. O serviço de infra-estrutura de chaves públicas proposto produz menos *overhead* do que alguns métodos baseados em *difusão*.

Também ficou evidenciado que o desempenho melhor ou pior dos algoritmos propostos depende da mobilidade dos nós e dos protocolos de roteamento conforme mostrado nas Figuras 5.1, 5.2 e 5.3. Quanto maior a mobilidade, maior é o número de mensagens de controle trocadas. Para certas situações, conforme os cenários variam em termos de quantidade de nós, número de conexões e velocidade, um protocolo de roteamento se adapta melhor do que outro. Apesar disso, nota-se que nas simulações apresentadas, na grande maioria das vezes, o protocolo pró-ativo (DSDV) apresentou melhores resultados.

Diferentemente dos métodos citados que utilizam difusão na comunicação entre as ACs e entre os nós e as ACs, o trabalho propõe uma maneira mais eficiente (via *unicast*) para tratar do envio e recebimento da lista de certificados revogados. A criação de uma lista de ACs confiáveis bem como a sua manutenção, também foram pontos importantes a se destacar neste trabalho. Esta lista de ACs é consultada e utilizada para otimizar os protocolos de autenticação e renovação de certificados.

Todos os algoritmos da proposta foram apresentados e também foram feitas as análises

de complexidade de tempo e do número de mensagens trocadas por estes algoritmos. Chega-se à conclusão de que a complexidade dos algoritmos é aceitável quando se leva em conta o benefício que os mesmos trazem à solução. A descrição dos algoritmos encontra-se no Anexo A.

Também foi discutido o tamanho mínimo que um certificado digital deveria ter de maneira que a solução possa ser implementada usando o padrão X.509. Além dos certificados de cada nó, todas as estruturas de controle auxiliares (LAC, LCR e certificados de outros nós) que necessitam ficar guardadas nos nós da rede foram analisadas e descritas. Chega-se à conclusão de que a quantidade de bits utilizados por estas estruturas é pequena quando comparada ao tamanho das mensagens trocadas pelas diversas aplicações que podem funcionar numa rede sem fio.

Os ataques descritos no capítulo 2 - *Black Hole, Spoofing, Sybil* e *Jelly Fish* - podem ser mitigados usando-se as técnicas e algoritmos aqui descritos. Mais uma vez é importante lembrar que o objetivo do trabalho não é bloquear ou detectar ataques, mas inibir seus efeitos protegendo a infra-estrutura e as comunicações.

A solução proposta pode ser utilizada em várias plataformas de nós móveis pois seu consumo de energia também é otimizado. Além de ser mais eficiente do que a proposta apresentada em [59], diminui o número de mensagens de controle trocadas e conseqüentemente diminui o consumo de energia. A comparação entre os métodos foi mostrada no Capítulo 5 e evidenciada nas tabelas de consumo de energia. Além disso, este consumo de energia pode ser melhorado com algumas soluções propostas em [36] as quais sugerem otimizações nos protocolos de criptografia quando utilizados em redes ad hoc. O consumo de energia de vários algoritmos de criptografia e *hashing* foi mostrado para ilustrar o problema e também para que as comparações e análises devidas fossem feitas.

Cabe salientar que este estudo já deu origem a duas publicações até o presente momento. A primeira no 6o Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais - SBSEG/2006 [10] e a segunda no 8o Simpósio de Segurança em Informática - SSI/2006 [9].

Finalmente, a solução proposta pode também ser usada por múltiplos serviços de autoridade certificadora utilizando limite criptográfico e certificação cruzada. O protocolo de manutenção das ACs confiáveis neste novo esquema deverá ser o objetivo principal de um trabalho futuro. Outro tema para trabalhos futuros seria a comparação das técnicas aqui propostas com outras técnicas que visam garantir comunicação segura numa rede ad hoc sem estabelecimento de um canal seguro utilizando outras técnicas como *Messages*

Authentication Codes, criptografia por ID e esteganografia por exemplo [5].

APÊNDICE A

Nesta apêndice são apresentados brevemento os algoritmos da proposta. A forma dos algoritmos apresentados [7] mostra uma maneira melhor de implementar tais algoritmos em um ambiente distribuído orientado a troca de mensagens do que a usada normalmente em algoritmos estruturados.

Algoritmo 1.1: Manutenção da lista de ACs confiáveis

Input: Msg (Acusada) vindo de N_k acusando N_j

Ação:

1: **se** N_k AND N_j são confiáveis **então**
2: Pesquisar se é a primeira acusação de K contra J
3: **se** é a primeira acusação **então**
4: Contabilizar a acusação
5: Atualizar contador de acusações contra J
6: **se** contador de acusações == T **então**
7: Remover J da lista de ACs confiáveis
8: Cifrar uma cópia da lista com a chave pública das ACs destino (uma cópia para cada AC)
9: Enviar "NovaLista" $\forall Elemento$ da nova lista de ACs confiáveis
10: **fim se**
11: **fim se**
12: **fim se**

Algoritmo 1.2: Manutenção da lista de ACs confiáveis

Input: Msg (NovaLista) venda de N_j

Ação:

1: **se** N_j é confiável **então**
2: Decifrar "NovaLista" com chave privada
3: Atualizar a Lista de ACs confiáveis
4: **fim se**

Algoritmo 2: Mensagens de renovação de certificado vindas do Agente.

Representa o passo 1 do protocolo de renovação de certificados.

Variáveis:

D : Conjunto de ACs escolhidas

Input: Msg (Renova) vinda do Agente

Ação:

1: Escolher *Kno* ACs Lista de Confiáveis e criar o conjunto "D" de ACs
2: Enviar mensagem "CertRenova" $\forall$*Elemento em D*
3: Enviar mensagem "IniciaTempo" para o Agente
4: Aguardar pelas mensagens de "Reply"

Algoritmo 3: Recebendo respostas das ACs.

Corresponde ao Passo 3 do protocolo de renovação de certificados.

Input Msg: (Reply) venda de uma AC N_j

Ação:

1: **se** estava esperando por mensagem de N_j **então**
2: **se** é a primeira mensagem de "Reply" a chegar **então**
3: Escolher N_j como AC Combinadora
4: Enviar "ResetTempo" para o Agente
5: Gera novo par de chaves

6: **fim se**
7: **se** N_j é a AC Combinadora **então**
8: Enviar "Ack" e "Cert" para N_j
9: **senão**
10: Enviar "Ack" para N_j
11: **fim se**
12: **se** Contador de acusações contra N_j é diferente de zero **então**
13: Zerar contador de acusações contra N_j
14: Enviar mensagem "ZeraAcusação" para J $\forall k$ sendo N_k confiável
15: **fim se**
16: **fim se**

Algoritmo 4: Tempo expirado no passo 3 do protocolo de renovação de certificados.

Input Msg: TempoExpirado vinda do Agente

Variáveis:

D : Conjunto de ACs escolhidas

Ação:

1: **enquanto** a lista de ACs confiáveis não chegar ao fim **faça**
2: Escolher *Kno* ACs e colocá-las no conjunto D
3: Enviar "IniciaTempo" para o Agente
4: **para todo** N_j in D **faça**
5: Aguardar pela resposta de N_j
6: **fim para**
7: **fim enquanto**

Algoritmo 5: AC contatada envia "Reply"para o nó.

Passo 2 do protocolo de renovação de certificados.

Input Msg: CertRenova vinda de N_j

Ação:

1: Enviar "Reply" to N_j

Algoritmo 6: A AC combinadora requisita assinatura parcial para *Kac* ACs.

Passo 4 do protocolo de renovação de certificados.

Input Msg: (Cert) vinda de N_i.

Ação:

1: Tornar-se a AC combinadora para esta requisição
2: Selecionar *Kac* ACs
3: **para todo** N_j tal que N_j é AC contatada) **faça**
4: Aguardar resposta de N_j
5: Enviar "AssinaParcial" para N_j
6: Enviar "IniciaTempo" para o Agente
7: **fim para**

Algoritmo 7: Tempo expirado. A AC combinadora não recebeu todas as assinaturas parciais.

Passo 4 do protocolo de renovação de certificados.

Input Msg: (TempoExpirado) vinda do Agente

Ação:

Requer: Condição 2 é verdadeira

1: **para todo** ACs que não respondem **faça**
2: Marcar as ACs
3: **fim para**
4: Selecionar outras *Kac* ACs
5: **para todo** N_j tal que N_j é nova AC a ser contatada **faça**
6: Enviar "AssinaParcial" para N_j
7: Enviar "IniciaTempo" para o Agente

8: **fim para**

Algoritmo 8: Requisição de assinaturas parciais.

Passo 5 do protocolo de renovação de certificados.

Input Msg: (AssinaParcial) vinda de N_j

Ação:

1: Gerar assinatura parcial no Certificado
2: Enviar "CertParcial" para N_j

Algoritmo 9: Recebendo assinaturas parciais e renovando certificados.

Passo 6 do protocolo de renovação.

Input Msg: (CertParcial) vinda de N_j

Action:

1: **se** ainda não recebeu Certificado Parcial de N_j **então**
2: **se** Condição 1 == False **então**
3: $FinalCert \leftarrow FinalCert + Cert$
4: Decrementar contador de partes de certificado
5: Marcar N_j como AC que já enviou Certificado Parcial
6: **senão**
7: Remover N_j da lista de ACs confiáveis
8: Cifra NovaLista
9: SEND "NovaLista" $\forall N_i$ tal que N_i *pertencente a* $NovaLista$
10: **fim se**
11: **se** contador de partes de certificado == 0 **então**
12: Gerar assinatura com as "T"partes
13: Enviar "FinalCert" para N_k (nó que requisitou a renovação)
14: **fim se**
15: **fim se**

Algoritmo 10: AC recebe mensagem "ZeraAcusação"contra a AC_j.

Input Msg: ZeraAcusação vinda de N_j

Ação:

1: Zerar contador de Acusações contra a AC_j

APÊNDICE B

Nesta apêndice são apresentados de forma mais detalhada os cálculos que mostram o gasto de energia dos algoritmos da proposta, bem como o cálculo do gasto aproximado do algoritmo de difusão.

No consumo de energia para estabelecimento de rotas foi considerado que os nós devem se autenticar mutuamente usando o protocolo de autenticação mostrado na Figura 4.2 com N=15 (número de ACs) e T=8 (limite criptográfico). A constante Kno utilizada foi $Kno = \lceil N/Log_2 2T \rceil$ e Kac foi $Kac = 3 \times (T-1)/2$. Conseqüentemente os passos abaixo descritos são executados:

1. O transmissor deve assinar digitalmente o pacote de requisição de autenticação (1 byte) e enviá-lo junto com seu certificado digital que contém sua chave pública: $1 \times 4,27$ (custo por byte para cifrar com protocolo RSA) +(300(certificado)+20(cabeçaho)+16(hash $\times 0,17 \rightarrow$ 61,39mJ;

2. O receptor irá contatar $Kno = \lceil N/Log_2 2T \rceil$ ACs para verificar e validar a chave pública do outro nó: $\lceil N/Log_2 2T \rceil \times 0,17mJ \times 302$(tamanho do certificado mais a mensagem VerificaCert) $\rightarrow 154,02mJ$;

3. As ACs do serviço ICPAH irão verificar a chave pública recebida. Depois irão enviar a resposta para o nó receptor: $2/3 \times 0,17 \times \lceil N/Log_2 2T \rceil$ (número de ACs que irão responder) $\times 2$ (Mensagem CertVerificado com o ID do nó cuja chave foi verificada e a resposta) $\rightarrow 0,68mJ$;

4. O receptor irá receber as respostas do serviço de criptografia: $2/3 \times \lceil N/Log_2 2T \rceil \times 0,11$ (verificação da assinatura da AC) $\rightarrow 0,22mJ$;

5. O nó receptor deve cifrar o *nonce* gerado (1 byte), a chave simétrica escolhida (128 bits ou 16 bytes) e o pacote de resposta de autenticação (1 byte) com sua chave privada. O receptor deve cifrar o *hashing* da mensagem com sua chave privada também:

- Gerar chave simétrica $\rightarrow 7,87\mu$J;
- Cifrar - $18 \times 1,21\mu J \rightarrow 21,78\mu$J;
- Gerar Hash - $18 \times 0,59\mu J \rightarrow 10,62\mu$J;
- Transmitir - $(18 + 16(hashing)) \times 0,17 \rightarrow 5,78$mJ.

Total de 5,83mJ;

6. O transmissor irá receber a mensagem: $34 \times 0,11 \rightarrow 3,74$mJ.
7. O transmissor irá contatar $Kno) = \lceil N/Log_2 2T \rceil$ ACs para verificar e validar a chave pública do outro nó: $\lceil N/Log_2 2T \rceil \times 0,17mJ \times 302$(tamanho do certificado mais a mensagem VerificaCert) $\rightarrow$ 154,02mJ;
8. As ACs do serviço ICPAH irão verificar a chave pública recebida. Depois irão enviar a resposta para o nó receptor: $2/3 \times 0,17 \times \lceil N/Log_2 2T \rceil$(número de ACs que irão responder) $\times 2$ (Mensagem CertVerificado com o ID do nó cuja chave foi verificada e a resposta) $\rightarrow 0,68mJ$;
9. O transmissor irá receber as respostas do serviço de criptografia: $2/3 \times \lceil N/Log_2 2T \rceil \times 0,11$ (verificação da assinatura da AC) $\rightarrow 0,22mJ$;
10. O transmissor cifra o reconhecimento (ACK) com a chave simétrica de sessão e o envia para o receptor:$1,21 + 170 \rightarrow 171,21\mu$J.

O total de energia gasta pelo nó transmissor é de 215,63mJ , o total gasto pelo receptor é de 154,28mJ e o total gasto pelo serviço ICPAH é de 1,36mJ.

Os itens 1,6,7, 9 e 10 correspondem aos passos que levam o transmissor (o nó que enviou a requisição de rota) a gastar energia adicional com o protocolo. Os itens 2,4 e 5 são os passos que fazem com que o receptor da requisição gaste energia adicional com o protocolo. Os passos 3 e 8 indicam que o serviço ICPAH ou o conjunto de ACs é que gasta energia adicional com o protocolo.

Para o processo de renovação de certificados com N=15 (número de ACs) e T=8 (limite criptográfico) tem-se:

1. Geração do novo par de chaves pelo nó : 270,13 mJ;
2. $\lceil N/Log_2 2T \rceil \times 0,17 \rightarrow 0,51mJ$ (CertRenova enviados pelo nó);
3. $2/3 \times \lceil N/Log_2 2T \rceil \times 0,22 \rightarrow 0,44mJ$ (recebimento da solicitação nas ACs);

4. $2/3 \times \lceil N/Log_2 2T \rceil \times 0,17 \rightarrow 0,68mJ$ (Resposta do serviços ICPAH, envio de ACKs);
5. $2/3 \times \lceil N/Log_2 2T \rceil \times 0,11 \rightarrow 0,22mJ$ (ACKs recebidos);
6. $301 \times 0,17mJ \rightarrow 51,17mJ$ (Cert enviado);
7. $301 \times 0,11mJ \rightarrow 33,11mJ$ (AC Combinadora recebe o certificado e a mensagem Cert enviado);
8. $301 \times 0,17 \times 3 \times (T-1)/2 \rightarrow 537,29mJ$ (Ac combinadora envia AssinaParcial para Kac);
9. $(301 \times 0,11 \times 3 \times (T-1)/2) + (3 \times (T-1)/2 \times 4,27) \rightarrow 392,49mJ$(ACs recebem certificado e executam assinatura parcial);
10. $301 \times 0,17 \times (1-0,33) \rightarrow 34,28$ (ACs enviam resposta para combinadora);
11. $301 \times 0,11 \times (1-0,33)^2 \rightarrow 14,86$ (AC Combinadora recebe as respostas);
12. 270,3 mJ (AC combinadora calcula chave privada do ICPAH e assina certificado);
13. $301 \times 0,17 \rightarrow 51,17$ AC combinadora envia certificado (FinalCert);
14. Nó requisitante recebe FinalCert: $301 \times 0,11 \rightarrow 33,11mJ$.

O nó requisitante gasta 354,89mJ, e o serviço ICPAH como um todo gasta 1334,62mJ, nesse caso como serão no máximo treze ACs envolvidas ($Kno + Kac$) a média de consumo entre as ACs é de 102,66mJ.

Para calcular a energia gasta com a troca de mensagens é suficiente assumir que as mensagens não serão cifradas gerando um consumo constante (*hashing* do cabeçalho da mensagem mais a cifragem do *hashing* com a chave simétrica de sessão):

1. 20 bytes (tamanho usual do cabeçalho no DSDV) $\times 0,59 \rightarrow 11,80\mu J$;
2. Cifragem do *hashing* com a chave simétrica: $16\ bytes \times 1,21\mu J \rightarrow 19,36\mu J$;

Somando (1) e (2) chega-se ao total de $31,16\mu J$ ou 0,031mJ.

Referências

[1] I. Aad, J. Hubaux e E.W. Knightly. Denial of service resilience in ad hoc networks. Em *In 10th Annual International Conference On Mobile Computing and Networking, Mobicom'04*, pp. 502–205, Philadelphia, USA, 2004.

[2] ABNT. Assosiação brasileira de normas técnicas. norma abnt iso 17799. 2005.

[3] B. Aboba, L. Blunk, J. Vollbrecht, C. Carlson e H. Levkowetz. Eap, Junho 2004. RFC 3748.

[4] C. M. Adams. Constructing symmetric ciphers using the cast design procedure. Em *Designs, Codes, and Cryptography*, volume 12, pp. 283–316, 1981.

[5] Célio Vinícius Neves Albuquerque, Eduardo Pagani Júlio e Wagner Brazil. Minicurso: Esteganografia e suas aplicações. Em *7o Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2007)*, Rio de Janeiro, RJ, Brasil, Agosto 2007.

[6] GSM Association. Gsm world. 2007. Disponível em http://www.gsmworld.com/index.shtml.

[7] Valmir C. Barbosa. *An Introduction to distributed algorithms*. MIT-Press, 2nd edição, 1996.

[8] Dan Boneh e Matt Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal of Computing*, 32(3):586–615, 2003.

[9] Wagner Brazil e Célio Vinícius Neves Albuquerque. Protecting ad hoc networks using digital certificates and threshold cryptography. Em *8th International Symposium on Systems and Information Security (SSI 2006)*, São José dos Campos, SP, Brasil, Novembro 2006.

[10] Wagner Brazil e Célio Vinícius Neves Albuquerque. Protegendo redes ad hoc com certificados digitais. Em *6o Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2006)*, Santos, SP, Brasil, Agosto 2006.

[11] J. Callas, L. Donnerhacke, H. Finney e R. Thayer. Open pgp message format. Novembro 1998. RFC 2440.

[12] Joan Daemen, Steve Borg e Vincent Rijmen. The design of rijndael: Aes. Em *The Advanced Encryption Standard*. Springer-Verlag, 2002.

[13] Daniel Balparda de Carvalho. *Segurança de Dados com Criptografia. Métodos e Algoritmos*. Book Express, Rio de Janeiro, RJ, Brasil, segunda edição, 2001.

[14] Ivana Cardial de Miranda Pereira e Aloysio de Castro P. Pedroza. Redes móveis ad hoc aplicadas a cenários militares. Em *4o Congresso Brasileiro de Computação (CBComp 2004)*, Rio Grande do Sul, Brasil, 2004.

[15] Y. Desmedt. Threshold cryptography. 5(4):449–457, Julho 1994.

[16] W. Diffie e M. E. Hellman. New directions in cryptography. Em *IEEE Transactions on Information Theory*, volume IT-22, pp. 644–654, 1976.

[17] John R. Douceur. The sybil attack. Em *In Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02)*, pp. 251–260, Cambridge, MA, USA, Março 2002.

[18] D. Eastlake e P. Jones. Us - secure hash algorithm 1 (sha1). Setembro 2001. RFC3174.

[19] William Friedrich Ehrsam, Carl H. W. Meyer, Robert Lowell Powers, John Lynn Smith e Walter Leonard Tuchman. Product block cipher system for data security. 1975. U.S. Patent 3.962.539.

[20] Taher ElGamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. Em *IEEE Transactions on Information Theory*, volume IT-31, pp. 469–472, 1985.

[21] E. O. Elliot. Estimates of error rates for codes on burst-noise channels. Em *Bell System Technical Journal*, volume 42, pp. 1977–1997, 1963.

[22] Reinaldo Penno Filho. Desvendando o tcp. Em *Boletim bimestral sobre tecnologia de redes*, volume 2, Abril 1988.

[23] M. J. Fisher, N. A. Lynch e M. S. Paterson. Impossibility of distributed consensus with one faulty processor. *Journal of the ACM*, 32(2):374–382, Abril 1985.

[24] Wimax Forum. Wimax, 2007. Disponível em http://www.wimaxforum.org.

[25] A. Freier, P. Karlton e P. Kocher. Secure socket layer. Março 1996.

[26] Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk e Tal Rabin. Robust threshold DSS signatures. *Lecture Notes in Computer Science*, 1070:354–371, 1996.

[27] E. N. Gilbert. Capacity of a burst-noise channel. Em *Bell System Technical Journal*, volume 39, pp. 1253–1265, 1960.

[28] Denise Hideko Goya. Propostas sob o modelo de criptografia de chave pública sem certificado. Dissertação de Mestrado, Universidade de São Paulo, Instituto de Matemática e Estatística, Departamento de Ciência da Computação, 2006.

[29] Bluetooth Special Interest Group. Bluetooth. 2007. Disponível em http://www.bluetooth.com.

[30] N. Haller, P. Nesser e M. Straw. One time password. Fevereiro 1998. RFC 2289.

[31] Min-Shiang Hwang e Ting-Yi Chang. Threshold signatures: Current status and key issues. Em *International Journal of Network Security*, volume 1, pp. 123–137, Hsinchu, Taiwan, Novembro 2005.

[32] IEEE802.11. Wireless lan medium access control (mac) and physical layer(phy) specifications sponsor lan man standards committee of the ieee computer society reaffirmed 12 june 2003 ieee-sa standards board. Relatório técnico, 1999. IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks- Specific requirements- Part 11.

[33] IEEE802.15. Wireless medium access control (mac) and physical layer (phy) specifications for wireless personal area networks (wpans(tm)). Relatório técnico, 2005. IEEE Standard for information technology–Telecommunications and information exchange between systems– Local and metropolitan area networks–Specific requirements. Part 15.1.

[34] Van Jacobson. Congestion avoidance and control. Em *ACM SIGCOMM '88*, pp. 314–329, Stanford, CA, Agosto 1988.

[35] P. Jacquet, P. Muhlethaler e A. Qayyum. Optimized link state routing protocol. Novembro 1998. in IETF MANET: Internet Draft.

[36] R. Karri e P. Mishra. Minimizing energy consumption of secure wireless session with qos constraints. Em *In Proceedings of the IEEE International Conference on Communications (ICC)*, volume 4, pp. 2053–2057, New York, USA, Maio 2002.

[37] C. Kaufman. Dass: distributed authentication security service. 1993. IETF - Request for Comments 1507.

[38] K. Kaukone e R. Thayer. A stream cipher encryption algorithm. Dezembro 1999. ARC-FOUR.

[39] A. Khalili, J. Katz e W. Arbaugh. Toward secure key distribution in truly ad-hoc networks. Em *In Proceedings of IEEE Security and Assurance in Ad-Hoc Networks at Int'l Symp. on Applications and the Internet (SAINT'03)*, pp. 342–346, Seoul, Corea, 2003.

[40] J. Kong, P. Zerfos, H. Luo e S. Lu. Providing robust and ubiquitous security support for wireless mobile networks. Em IEEE CS Press, editor, *Proceedings of the 9 Conference on Network Protocols (ICNP)*, pp. 251–260, Riverside, California, Novembro 2001. IEEE.

[41] David W. Kravitz. Federal information processing standards publication 186-2. Julho 1991. U.S. Patent 5.231.668.

[42] RSA Labs. Pkcs1 v1.5 - rsa laboratories technical note. 1993. Disponível em https://www.rsa.com/rsalabs/node.asp?id=2125.

[43] Yi Lu, Yuhui Zhong e Bharat Bhargava. Packet loss in mobile ad hoc networks. Relatório técnico, Department of Computer Sciences, Purdue University, USA, Abril 2003.

[44] H. Luo, P. Zerfos, J. Kong, S. Lu e L. Zhang. Self-securing ad hoc wireless networks. Em *In 7th IEEE Symp. on Computers and Communications (ISCC '02)*, pp. 567–576, Taormina, Itália, Julho 2002.

[45] Gary Scott Malkin. Rip version 2. Novembro 1998. RFC 2453.

[46] J. L. Massey e X. Lai. Device for the conversion of a digital block and use of same. Maio 1993. U.S. Patent 5.214.703.

[47] Ralph Merkle e Martin Hellman. On the security of multiple encryption. Em *Communications of the ACM*, volume 24, pp. 465–467, Julho 1981.

[48] John Moy. Ospf version 2. Abril 1998. RFC 2328.

[49] NS-2. The network simulator. 2007. Disponível em http://www.isi.edu/nsnam/ns/.

[50] Charles Perkins. Ad-hoc on-demand distance vector routing. 1997. Ad-hoc on-demand distance vector routing, in MILCOM '97 panel on Ad Hoc Networks, Nov. 1997.

[51] Charles Perkins e Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. Em *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, pp. 234–244, 1994.

[52] Qualcom. Cdma, 2007. Disponível em http://www.qualcomm.com/technology/.

[53] T. Rabin. A simplified approach to threshold and proactive rsa. Em *Advances in Cryptology - Crypto - 98*, pp. 89–104, New York, 1998.

[54] C. Rigney, S. Willens, A. Rubens e W. Simpson. Radius. Junho 2000. RFC 2865.

[55] Ron Rivest. The md5 message-digest algorithm. Abril 1992. RFC 1321.

[56] Ron Rivest, Adi Shamir e L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Em *Communications of the ACM*, volume 21, pp. 120–126, 1978.

[57] Bruce Schneier. Fast software encryption. Em *Cambridge Security Workshop Proccedings*, pp. 191–204. Springer-Verlag, Dezembro 1994.

[58] Bruce Schneier. Twofish: A 128-bit block cipher. Relatório técnico, Junho 1998. Disponível em http://www.schneier.com/paper-twofish-paper.pdf.

[59] Yi Seung e Robin Kravets. Moca:mobile certificate authority for wireless ad hoc networks. Em *Proceedings of 2nd Annual PKI Research Workshop*, 2002.

[60] Adi Shamir. How to share a secret. *Communication of ACM*, 22(11):612–613, 1979.

[61] Bernardo A.M. Villela e Otto Carlos M.B. Duarte. Maximum throughput analysis in ad hoc networks. Em Springer Berlin / Heidelberg, editor, *Proceedings of Third International IFIP-TC6 Networking Conference*, volume 3042, pp. 223–234, Athens, Greece, 2004.

[62] G. Zacharia, A. Moukas e P. Maes. Collaborative reputation mechanisms in electronic marketplaces. Em *32nd Annual Hawaii International Conference*, volume 8, pp. 7, Maui, HI, USA, Janeiro 1989.

[63] Lidong Zhou e Zygmunt J. Haas. Securing ad hoc networks. *IEEE Network Magazine*, 13(6):24–30, Novembro 1999.

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Administração
Baixar livros de Agronomia
Baixar livros de Arquitetura
Baixar livros de Artes
Baixar livros de Astronomia
Baixar livros de Biologia Geral
Baixar livros de Ciência da Computação
Baixar livros de Ciência da Informação
Baixar livros de Ciência Política
Baixar livros de Ciências da Saúde
Baixar livros de Comunicação
Baixar livros do Conselho Nacional de Educação - CNE
Baixar livros de Defesa civil
Baixar livros de Direito
Baixar livros de Direitos humanos
Baixar livros de Economia
Baixar livros de Economia Doméstica
Baixar livros de Educação
Baixar livros de Educação - Trânsito
Baixar livros de Educação Física
Baixar livros de Engenharia Aeroespacial
Baixar livros de Farmácia
Baixar livros de Filosofia
Baixar livros de Física
Baixar livros de Geociências
Baixar livros de Geografia
Baixar livros de História
Baixar livros de Línguas

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo