UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ALEXANDRE AUGUSTO GUEDES GUIMARÃES

PROPOSTA DE UM MODELO DE SEGURANÇA PARA VPNs NA INTERLIGAÇÃO DE REDES CORPORATIVAS

RECIFE

Fevereiro de 2004

ALEXANDRE AUGUSTO GUEDES GUIMARÃES

PROPOSTA DE UM MODELO DE SEGURANÇA PARA VPNs NA INTERLIGAÇÃO DE REDES CORPORATIVAS

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Engenharia Elétrica do Centro de Tecnologia e Geociências/Escola de Engenharia de Pernambuco, Universidade Federal de Pernambuco.

Orientador: Prof. Dr. Rafael Dueire Lins.

RECIFE

Fevereiro de 2004

ALEXANDRE AUGUSTO GUEDES GUIMARÃES

PROPOSTA DE UM MODELO DE SEGURANÇA PARA VPNs NA INTERLIGAÇÃO DE REDES CORPORATIVAS

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Engenharia Elétrica do Centro de Tecnologia e Geociências/Escola de Engenharia de Pernambuco, Universidade Federal de Pernambuco.

Aprovado em 27 de fevereiro de 2004

BANCA EXAMINADORA

_____

Prof. Dr. Rafael Dueire Lins
Orientador

_____

Prof. Dr. Valdemar Cardoso da Rocha Jr.

_____

Prof. Dr. Francisco Heron de Carvalho Jr.

A minha esposa Lyvia e minha filha Luana
pelo nosso grande amor

Agradecimentos

A Deus por ter tornado esse sonho uma realidade e por ter abençoado toda a minha família com saúde, união e felicidade. Agradeço por ele ter me presenteado com uma esposa que me ama, me incentiva e que me ilumina todos os dias. Agradeço por ter nos concedido o maior de todos os presentes: uma filha linda e saudável. Agradeço a Deus por ter possibilitado a cura do câncer da minha querida mãe. E finalmente agradeço a Deus por me sentir uma pessoa realizada, saudável e feliz.

A meus pais, Delzuite e Honório, pela minha formação e educação que possibilitou que eu alcançasse todas as minhas realizações, como ser humano e como profissional.

A Lyvia, minha amada esposa, pelo seu amor, carinho, compreensão e incentivo. Agradeço e dedico este trabalho a você, por todos os momentos que tivemos que renunciar durante a realização deste trabalho.

A minha filha Luana, de dois anos de idade, que me enche de alegrias, que me transformou em uma pessoa mais sensível e preocupada com o próximo, que me fez perceber tudo aquilo que havia de melhor guardado dentro de mim e que me deu forças para prosseguir neste desafio.

Aos pais da minha esposa, Adauto e Flávia (Anjinha), por serem verdadeiramente meus segundos pais, me incentivando e me amando em todos os momentos. Agradeço especialmente a Anjinha por ter cuidado com tanto amor e carinho da minha filha durante todo o período desse trabalho, que de certa forma, preencheu a minha ausência.

Aos professores da Universidade Federal de Pernambuco que participaram desse Mestrado, em especial ao Professor Rafael Dueire Lins, por toda orientação, apoio e atenção, fornecidas durante o desenvolvimento deste trabalho.

Aos meus colegas de Mestrado, em especial a minha amiga Laura Jane pelo incentivo e apoio.

Ao Instituto de Tecnologia da Amazônia - UTAM por ter oportunizado o curso de Mestrado.

A Dataprev pelo apoio e suporte para o desenvolvimento deste Trabalho.

Aos meus colegas da Dataprev que direta, ou indiretamente, me ajudaram na conclusão deste Trabalho, em especial ao colega e amigo Sérgio Dantas Silvestre pelo seu apoio e compreensão.

A Fundação de Amparo a Pesquisa do Estado do Amazonas - FAPEAM por ter me concedido uma bolsa de incentivo a Pós-graduação.

A todos os professores da UTAM que me incentivaram na conclusão desse Mestrado.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

ACL	*Access Control Lists*
AH	*Authentication Header*
ATM	*Asynchronous Transfer Mode*
CA	*Certification Authority*
CDP	*Cisco Discovery Protocol*
CHAP	*Challenge Handshake Authentication Protocol*
CRL	*Certificate Revocation List*
DDoS	*Distributed Denial of Service*
DMZ	*DeMilitarized Zone*
DNS	*Domain name service*
DoS	*Denial of Service*
EAP	*Extensible Authentication Protocol*
EAP	*Extensible Authentication Protocol*
ESP	*Encapsulating Security Payload*
GPL	*General Public License*
GRE	*Generic Routing Encapsulation*
HMAC	*Hashing Message Authentication Code*
IDS	*Intrusion Detection System*
IETF	*Internet Engineering Task Force*
IKE	*Internet Key Exchange*
IP	*Internet Protocol*
IPSec	*IP Security*
IPv4	*Internet Protocol versão 4*
IPv6	*Internet Protocol versão 6*
ISAKMP	*Internet Security Association and Key Management Protocol*
ISDN	*Internet Service Digital Network*
ISP	*Internet Service Provider*
ITU-T	*International Telecommunication Union – Telecommunication Sector*
L2F	Layer 2 Forwarding
L2TP	*Layer 2 Tunneling Protocol*
LAC	L2TP *Access Concentrator*

LCP	*Link Control Protocol*
LDAP	*light-weight Directory Access Protocol*
LDP	*Label Distribution Protocol*
LNS	*L2TP Network Server*
LSP	*Label Switch Path*
LSR	*Label Switch Router*
MD5	*Message Digest 5*
MPLS	*Multiprotocol Label Switching*
MPPE	*Microsoft Point-to-point Encryption*
NAS	*Network Authentication Service*
NAT	*Network Address Translation*
NCP	*Network Control Protocol*
NIST	*National Institute os Standards and Technology*
NSP	*Prove*
OE	*Opportunistics Encryption*
PAP	*Password Authentication Protocol*
PKI	*Public Key Infrastructure*
PPP	*Point-to-Point Protocol*
PPTP	*Point-to-Point Tunneling Protocol*
PVC	*Permanent Virtual Circuits*
RA	*Registration Authority*
RAS	*Remote Access Services*
RFC	*Request for Comments*
RPM	*Red Hat Packet Manager*
SA	*Security Associations*
SAD	*Security Association Database*
SHA	*Secure Hash Algorithm*
SPD	*Security Policy Database*
VPN	*Virtual Private Network*
WCCP	Web Cache Communication Protocol

Resumo

Uma *Virtual Private Network* - VPN, ou Rede Privada Virtual, é uma conexão de rede protegida, construída para uso privado de uma Empresa, estabelecida sobre uma infra-estrutura de rede pública e compartilhada. Uma VPN utiliza protocolos de segurança e tecnologias de criptografia e autenticação para garantir características de uma rede privada e dedicada às Corporações que utilizam uma infra-estrutura não-confiável, como a Internet, para interligação de suas redes ou de usuários remotos a estas. Portanto, corporações interessadas no uso desta tecnologia devem preocupar-se com vários aspectos de segurança envolvidos na interligação de suas redes através de uma infra-estrutura não-confiável. De maneira geral, poucas referências tratam essa questão de maneira completa, restringem-se muitas vezes a contextos isolados, sem a preocupação na adoção de outros mecanismos de segurança que possam ser combinados a uma infra-estrutura de VPN, com o objetivo de propiciar maior segurança no perímetro externo de uma rede corporativa.

Este trabalho investigou uma questão relevante ao cenário atual – a utilização de VPNs para interligação de redes corporativas através de um ambiente de rede não-confiável, a Internet, a qual já possui uma infra-estrutura montada e de grande disponibilidade e escalabilidade. Atualmente, empresas do mundo inteiro querem aproveitar a infra-estrutura da Internet para interligar suas redes corporativas (intranet), inclusive com redes de parceiros de Negócio (extranets). Querem buscar assim, uma alternativa viável e de baixo custo, que venha contribuir com reduções significativas nos custos referentes a contratação de circuitos privados de dados. Porém, desejam, ao mesmo tempo, uma solução que propicie segurança às informações do seu negócio.

Desta forma, este trabalho apresenta um estudo dos aspectos de segurança envolvidos na construção, manutenção e utilização de Redes Privadas Virtuais entre ambientes de rede corporativos. Além de ser desenvolvida a base conceitual sobre a qual se estabelece uma arquitetura de VPN, são detalhados os elementos dessa estrutura, bem como os serviços de segurança que devem ser combinados a *Gateways* VPN para propiciar o fortalecimento do Perímetro Externo de uma rede corporativa que utiliza a Internet, uma rede não confiável, para o estabelecimento de conexões VPN.

Para tal, modelos de segurança são propostos, os quais possuem a finalidade de estabelecer uma topologia que possa garantir um nível de segurança aceitável para VPNs, de tal sorte que os elementos e serviços de segurança, existentes nas redes corporativas, possam se somar, possibilitando a construção de uma arquitetura de defesa em profundidade. Desta forma, efetuamos um estudo das diversas formas de posicionamento dos elementos e serviços de segurança para a utilização de VPNs, de modo a possibilitar a análise e a construção de topologias adequadas às necessidades de cada Empresa. A proposta de uma VPN, combinada a serviços que propiciam defesa em profundidade, possui a finalidade de dificultar ataques e ameaças que possam vir a comprometer a integridade, sigilo, autenticidade e disponibilidade das informações de uma Empresa. Em conseqüência dessa questão, também realizamos um estudo de caso que se propôs a desenvolver um modelo de segurança para a rede da Previdência Social e que deverá ser submetido à apreciação de sua Diretoria.

Dentro desse contexto de redução de custos e aumento de segurança, outra questão investigada neste trabalho é sobre a adequação e viabilidade da utilização de *softwares* e ferramentas livres para o estabelecimento de uma solução de segurança completa para a construção de redes VPN. Em decorrência disso, foi implementado um cenário de uma VPN, como experimento, totalmente baseado em plataforma de *software* livre.

Palavras-chave: Redes Privadas Virtuais, Segurança, Defesa em Profundidade, *software* livre.

Abstract

A Virtual Private Network - VPN is a protected connection, built for the private use of the enterprise, over a shared public network. A VPN uses security protocols, cryptography technologies and authentication to guarantee characteristics of a private and dedicated network to enterprises that use a not-trustworthy infrastructure, as the Internet, for interconnection between its networks or remote users to these. Therefore, corporations interested in the use of this technology must be worried about some security aspects involved in the interconnection of its networks through a not-trustworthy infrastructure. General way, few references deal with this question in complete way, restrict many times the isolated contexts, without the concern in the adoption of other security mechanisms that can be combined to a VPN infrastructure, objectifying to propitiate greater security in the external perimeter of a corporative network.

This work investigated an excellent question to the current scene - the use of VPNs for interconnection the corporative networks through a not trustworthy environment, the Internet, which already owns an infrastructure mounted and with great availability and scalability. Currently, companies of the entire world want to use the Internet's infrastructure to establish connection with its corporative networks (Intranets), also with Partners of Business' Internet (extranets). Thus, they want to search an alternative viable, and low cost, that can to contribute with significant reductions in the referring costs to hire private circuits of data. However, they wish, at the same time, a solution that increases security to the information of its business.

This way, in this work we present a study of the involved security aspects in the construction, maintenance and use of Virtual Private Networks between Corporative Networks. Besides being developed the conceptual base on which it establishes a VPN architecture, the elements of this structure are detailed, as well as the security services that must be combined with gateways VPN to increase the security of corporative network's External Perimeter that uses the Internet, a not trustworthy network, for the establishment of connections VPN.

For that, security models are considered, which has the purpose to establish a topology that can guarantee a level of acceptable security for VPNs, this way the elements and existing security services in the corporative networks can be added, making the construction of security architecture in depth possible. As such, we studied many forms of elements positioning and security services in the use of VPNs, in order to make the analysis and the construction of adequate topologies to the necessities of each Company possible. The proposal of a VPN, combined the services that provide security in depth, has the purpose to difficult attacks and threats, which can come to compromise the integrity, secrecy, authenticity and availability of the information of a Company. In consequence of this question, we also carry through a case study that intends to develop a model of security for the Official Social Security network of Brazil and that the appreciation of its Direction will have to be submitted.

In this context of costs reduction and security increasing, another question investigated in this work is on the adequacy and viability of using free software and free tools as completely safe solution for the construction of VPNs. In result of this, was implemented a VPN scene, as experiment, total based in free software platform.

Key words: Virtual Private Networks, Security, Defense in depth, Free software

1 INTRODUÇÃO

É impraticável compreender o avanço e o desenvolvimento tecnológico que ocorreram nos últimos anos em Tecnologia da Informação e Telecomunicações, sem mencionar a influência que essas áreas tiveram com o advento e o crescimento da Internet. Atualmente, é inconteste que praticamente todas as Organizações possuem conexão com essa grande rede. Ela tornou-se de alcance global. Desta forma, torna-se bastante coerente propor a utilização de sua infra-estrutura, já montada, para interligar as redes corporativas. Obviamente, o grande desafio é em garantir segurança nos dados trafegados na Internet, uma vez que esta é uma rede compartilhada, baseada no protocolo TCP/IP, que não garante segurança das informações trafegadas. Dentro dessa abordagem, surgiu um novo paradigma: ***usar a rede pública como meio de interconectar redes privadas, dentro de critérios rígidos de segurança***.

Criou-se assim a *Virtual Private Network* (VPN) ou Rede Privada Virtual baseada na Internet como uma das formas de se interconectar diferentes redes privadas (SILVA, 2003) para o tráfego de dados entre elas, utilizando-se como meio, uma rede pública, a infra-estrutura da Internet (ORTIZ, 2003). Também são denominadas de VPNs IP. Sua característica principal é criar “túneis virtuais” de comunicação entre essas redes em uma infra-estrutura de rede não-confiável, de forma que os dados trafeguem criptografados por esses túneis e de forma transparente para as entidades ou redes que estão se comunicando, aumentando a segurança na transmissão e na recepção dos dados.

Definimos assim, uma Rede Privada Virtual, como sendo uma seção de rede protegida que forma um canal de comunicação com acesso controlado, permitindo conexões seguras

para apenas uma determinada comunidade, fazendo-se uso de uma infra-estrutura de rede não-confiável ou compartilhada.

Para que se tenhamos uma idéia do potencial e da importância dessa tecnologia no contexto atual, em março de 2003, a publicação *WorldTelecom* da editora *International Data Group* (IDG), publicou uma recente pesquisa da *Infonetics Research* que divulgava que as vendas de *software* e *hardware* para redes virtuais privadas (VPN) e sistemas de *firewall* tinham chegado a US$ 735 milhões no quarto trimestre de 2002, totalizando US$ 2,7 bilhões no ano, havendo uma expectativa para esse segmento de atingir um desempenho de US$ 5 bilhões em 2006, o que demonstra, claramente, o grande crescimento do interesse das VPNs pelo mercado corporativo. Esse mesmo artigo menciona o crescimento explosivo do mercado mundial de *gateways* para aplicações VPN, e traça uma projeção anual para 2006 de US$ 1 bilhão, apontando a *Cisco*, a *Check Point/Nokia* e a *NetScreen* como os três grandes fornecedores no segmento de VPN/*firewall*, com a *NetScreen* apresentando o crescimento mais rápido em vendas.

Diante desse conceito, este trabalho apresenta um estudo abrangente sobre os aspectos e as tecnologias existentes na área de segurança de redes de computadores envolvidos na construção e manutenção de redes VPN baseadas na Internet[1], com a finalidade maior de apresentar um modelo de segurança adequado e viável para a implementação de redes VPN, combinada a outros serviços de segurança, como por exemplo, *firewalls* (FIGUEIREDO, 2003), *proxys* (DOWNES et al., 2003), Sistemas de Detecção de Intrusos (CASWELL et al., 2003), entre outros, em um ambiente baseado na plataforma de *software* livre (STALLMAN, 2003), demonstrando inclusive o funcionamento e a viabilidade técnica dos produtos e sistemas existentes nessa plataforma para essa finalidade.

Esta dissertação também apresentará um estudo de caso para o uso de uma VPN IP como solução de interligação entre redes corporativas, utilizando-se como meio uma rede não-confiável, que é o caso da Internet. Ela investiga principalmente os impactos financeiros e técnicos decorrentes da construção e utilização de VPNs em substituição aos circuitos dedicados, especificamente nas redes da Previdência Social. O interesse em tal, decorre do fato do autor desta dissertação trabalhar como Analista de Suporte na Empresa de Tecnologia

[1] As VPNs baseadas na Internet também são denominadas de VPN IP, devido ao protocolo de rede usado na Internet, ou ainda, VPN de nível 3 ou VPN de nível de rede.

da Informação da Previdência Social - Dataprev, que é responsável por toda a área de comunicação e processamento de dados dessa Instituição Previdenciária.

1.1 Motivação

Um fato motivador dessa dissertação é que diversas Empresas e Instituições, mesmo geograficamente distribuídas e com acesso a Internet, ainda não possuem qualquer solução segura e de baixo custo para interligação de suas redes. Como exemplo, podemos citar as redes corporativas da Universidade Estadual do Amazonas que, atualmente, possui cinco unidades na capital e duas unidades no Interior do Estado, as quais não possuem qualquer interligação, apesar de todas terem acesso a Internet, e as redes dos Escritórios Estaduais da Empresa de Tecnologia da Previdência Social – Dataprev que existem em todos os Estados da Federação, e as quais são interligadas através de circuitos dedicados com os Centros de Processamento de Dados localizados no Rio de Janeiro, em São Paulo e em Brasília, isto é, interligadas através de circuitos interestaduais de altíssimo custo agregado.

Uma vez que ainda não há qualquer solução segura e de baixo custo para interligação das referidas redes, este trabalho procura realizar um estudo abrangente sobre os aspectos e as tecnologias de segurança existentes, com a finalidade de demonstrar a viabilidade na implementação e utilização de uma VPN IP, combinada a serviços do *Firewall*, *Proxys* e Sistemas de Detecção de Intrusos, em um ambiente baseado na plataforma de *software* livre.

Dessa forma, esta pesquisa se predispõe a buscar uma alternativa viável e de baixo custo, que venha contribuir com reduções significativas nos custos referentes à contratação de circuitos privados de dados, substituindo circuitos dedicados, em especial os circuitos *Frame Relay* interestaduais, de altíssimo custo ao tesouro do Governo Federal, e, concomitantemente, propiciar uma opção para as conexões *host-rede*.

Além da questão financeira, que é uma das grandes motivações existentes para a implementação de qualquer rede VPN, podendo trazer reduções significativas no custo em assinaturas de circuitos de comunicação interestaduais, outra grande vantagem é a possibilidade de expandir a quantidade de computadores que podem ter acesso à rede corporativa, isto é, um aumento na escalabilidade da rede sem investir em infra-estrutura extra, permitindo inclusive o suporte a usuários móveis, sem a utilização de bancos de

modems ou servidores de acesso remoto, ajudando a aumentar a flexibilidade e diminuir os gastos com equipamentos extras.

Como anteriormente mencionado, este estudo também deseja demonstrar a viabilidade técnica na implementação e utilização de uma VPN, combinada a serviços de *Firewall*, *Proxys,* Listas de Controle de Acesso e Sistemas de Detecção de Intrusos, em um ambiente baseado na plataforma de *software* livre, avaliando a eficácia e eficiência desta solução combinada, na segurança das informações trafegadas na rede, demonstrando quais os protocolos mais adequados para garantir o máximo de segurança, dentro de padrões de qualidade aceitáveis, e que possam também garantir um bom desempenho das aplicações corporativas na rede.

A utilização de *software* livre neste trabalho foi um fato bastante motivador, pois os sistemas de *software* livre em geral são seguramente um novo paradigma da nova era da Computação que influencia significativamente o comportamento das Organizações que utilizam Tecnologia da Informação como recurso estratégico e como mecanismo impulsionador dos seus negócios. A aplicação de *software* livre pelas Empresas nas soluções em comunicação de dados, em especial, na construção de redes VPN IP, se encaixa perfeitamente nas pretensões e requerimentos de reduções de custos exigidos, pois o *software* livre pode reduzir significativamente o custo na aquisição de licenças de *softwares* e ferramentas, bem como, pode propiciar a estas Empresas um aumento de segurança expressivo, uma vez que o código fonte dos sistemas livres é também disponibilizado. Desta forma, o código pode ser verificado, atualizado e modificado pelas próprias empresas, a qualquer momento e diante de qualquer necessidade específica. Portanto, é coerente que se pretenda buscar soluções mais econômicas e seguras para a implementação de uma VPN IP, combinada a serviços e protocolos de segurança que definem um modelo de Defesa em Profundidade, através de ferramentas e sistemas livres.

1.2 Questões de Pesquisa

A utilização de redes VPN IP em ambientes corporativos, traz uma série de problemáticas e situações que devem ser tratadas e resolvidas adequadamente a fim de garantir um nível de segurança satisfatório e confiável, pois a implementação de uma infra-estrutura VPN sobre uma rede não-confiável como a Internet exige, necessariamente, a

aplicação de outros serviços de segurança combinados, isto é, não basta apenas criar túneis criptografados para garantir segurança. Isto é um total equívoco. Deve existir a preocupação desde a segurança física do *site*, da capacitação e orientação dos Recursos Humanos de uma Empresa até a aplicação correta de serviços como *firewall*, *Proxys*, Sistemas de Detecção de Intrusos (SDI), Servidores de Backup, Servidores de Autenticação, Servidores de Antivírus, Servidores de Certificação, *Honey-Pots*, entre outros. A aplicação de segurança só não pode ter um custo maior que o prejuízo que pode ser causado por um invasor. Desta forma, esta dissertação, além de dar uma fundamentação teórica sobre os aspectos de segurança envolvidos em redes VPN, foi elaborada de forma a responder a algumas questões de pesquisa. A primeira a saber é:

1. "Quais são os protocolos de segurança adequados para suportar soluções VPN IP na Interligação de redes corporativas?"

No que diz respeito a essa questão, este trabalho investigará os diversos protocolos existentes para a implementação de redes VPN e definirá um protocolo mais adequado à interligação de redes corporativas. Não é foco deste trabalho discorrer profundamente sobre as soluções para acesso remoto, embora que a topologia a ser sugerida nesta dissertação sirva para este fim.

Outro ponto a ser pesquisado, o qual já comentado anteriormente, diz respeito a implementação e configuração de outros componentes de segurança que devem coexistir em uma solução VPN de modo a fortalecê-la. Pois apesar de sabermos que uma solução VPN baseada na Internet, que aplica princípios de tunelamento e criptografia, pode proporcionar serviços de integridade, privacidade e autenticidade durante uma conexão VPN, em uma rede não-confiável como a Internet, é inconcebível se adotar esta solução isoladamente, sem a adoção de outros mecanismos de segurança que possam garantir a integridade da rede corporativa. A ausência desta preocupação pode facilitar ataques e a invasão da rede corporativa, inclusive nos próprios servidores VPN e, conseqüentemente, facilitar o acesso de um invasor às chaves de criptografia, o que seria um desastre se não detectado a tempo ou impedido, acarretando na perda da privacidade e integridade na comunicação entre as redes corporativas. Por esta razão, deve-se estudar um modelo de segurança que atenda outros requisitos de segurança e que possa principalmente fortalecer a conexão VPN, bem como todo o perímetro externo de uma rede corporativa. Desta forma, houve a motivação de um outro questionamento para pesquisa:

2. "Quais os serviços ou componentes de segurança essenciais para garantir um nível de segurança aceitável em redes VPN?"

Ao longo deste trabalho serão discutidos vários serviços de segurança que devem ser utilizados em conjunto com uma solução VPN, os principais, a saber, são: *firewalls*, *Proxys*, Sistemas de Detecção de Intrusos (SDI), Servidores de Autenticação, Servidores PKI, Listas de Controle de Acesso e Servidores de Antivírus. Na verdade, a investigação dessa questão resultará em uma topologia de segurança (modelo proposto) para a implementação de uma solução VPN entre redes corporativas. Neste estudo, haverá outros desdobramentos, os quais esta dissertação discutirá posteriormente, que são decorrentes da questão de pesquisa supracitada, entre as quais cabe destacar:

2.1 *"Qual a disposição ideal desses componentes em relação ao Serviço de VPN?"*

2.2 *"Quais são as deficiências, riscos e as possíveis falhas de segurança em uma solução VPN nos modelos propostos?"*

2.3 *"Quais são as alternativas disponíveis dos diversos componentes de segurança a fim de se assegurar uma estrutura confiável?"*

2.4 *"Qual é o nível de escalabilidade do modelo?"*

2.5 *"Qual a redução de custos com o modelo de solução VPN?"*

2.6 *"É viável se implementar uma Defesa em Profundidade?"*

Outra questão de pesquisa que deve ser investigada nesta dissertação é a relacionada à aplicabilidade de soluções em *software* livre para a estruturação de um modelo de VPN adequado. O grande questionamento é:

3. *"É possível se estabelecer uma solução de segurança completa, que possa garantir um nível de segurança satisfatório, através da combinação de softwares livres para a interligação de redes corporativas através da Internet?"*

No que diz respeito a essa questão, vale destacar que os relatos encontrados até então na literatura se restringiam muitas vezes a contextos isolados, explorando a utilização de alguma solução VPN isoladamente, sem com isso apresentar um modelo de topologia completo que possa combinar outros serviços de Segurança no fortalecimento do perímetro de

uma rede corporativa. A situação ainda é mais precária na literatura nacional que apresenta pouca disponibilidade de livros sobre o assunto. Daí porque surgiu a necessidade de se investigar o uso de ferramentas livres combinadas para a implementação de soluções VPN completas que possam garantir um nível de segurança satisfatório para interligação de redes corporativas. Com isso, pode-se aferir a eficácia e eficiência das ferramentas livres para construção de um modelo que implemente soluções VPN.

As pesquisas oriundas destes questionamentos irão permitir contribuições significativas entre as quais destacam-se:

- Identificação dos aspectos e requisitos de segurança necessários e desejáveis para a utilização de redes VPN na conexão de redes corporativas.

- Identificação de um modelo de segurança adequado para a implementação de redes VPN em um modelo de defesa em profundidade.

- Avaliação dos sistemas e ferramentas livres para a implementação de redes VPNs, bem como para os outros serviços de segurança necessários nessa solução, avaliando a eficácia e eficiência de uma solução combinada.

- Avaliação de custos envolvidos na implantação de uma infra-estrutura VPN específica, principalmente diante dos aspectos de custo e disponibilidade (facilidades) de pontos e serviços de rede das concessionárias de telecomunicações, ou de provedores de Internet, que podem suprir as necessidades de interligação de redes corporativas.

Relacionadas a essas questões de pesquisa, destacam-se as seguintes hipóteses que norteiam a elaboração deste trabalho:

Hipótese 1: Todo e qualquer modelo de segurança que se aplique em uma solução VPN, por melhor que seja, ainda terá possibilidade de ser invadido e corrompido. Porém, a aplicação correta de um modelo de defesa em profundidade diminui a probabilidade da ocorrência de ataques bem sucedidos.

Hipótese 2: As soluções de segurança em *software* livre são mais adequadas e eficientes que as soluções proprietárias, principalmente por terem seus códigos fontes disponíveis e pela possibilidade de alteração e recompilação de todos os seus componentes, atendendo assim as necessidades de segurança das corporações mais exigentes.

Hipótese 3: Não há possibilidade de se prever todos os tipos de ataques que uma Empresa pode sofrer, principalmente quando a mesma utiliza uma infra-estrutura de rede não-confiável como a Internet.

Hipótese 4: Não existe sistema 100% seguro, principalmente porque os sistemas são operados e configurados por seres humanos, os quais são passíveis de falhas.

1.3 Objetivos

Com a realização desta pesquisa, pretende-se alcançar diversos objetivos que serão explicitados a seguir.

1.3.1 Objetivos Gerais

- Investigar os aspectos e requisitos de segurança necessários e desejáveis para a construção de redes VPN baseadas na Internet para a interligação de redes corporativas.

- Avaliar a adequação de outros componentes de segurança que possam ser combinados a uma VPN de forma a fortalecer a segurança no perímetro das redes conectadas, identificando um modelo de defesa em profundidade que possa propiciar um nível de segurança apropriado para a utilização de VPNs.

- Investigar a viabilidade de se estabelecer uma solução de segurança completa para a construção de redes VPN corporativas através da utilização de sistemas e ferramentas livres que possam garantir um nível de segurança satisfatório.

1.3.2 Objetivos Específicos

Os objetivos específicos a serem alcançados neste trabalho são:

- Desenvolver um experimento (estudo de caso) para a implementação de uma solução VPN que utilize ferramentas e sistemas livres, com o intuito de demonstrar o funcionamento e a viabilidade técnica dos produtos existentes nessa plataforma para este fim.

- Definir um modelo de segurança adequado para a implementação de redes VPN em uma infra-estrutura de defesa em profundidade.

- Definir uma normatização e orientação específica para aplicação de Listas de Controle de Acesso nos roteadores de redes corporativas que se interligam a Internet para a construção de VPNs.

- Identificar os impactos financeiros e técnicos decorrentes da construção de VPNs em substituição aos circuitos dedicados. Para isso, utilizar-se-á como referência as redes existentes na Previdência Social.

- Investigar deficiências e possíveis falhas de segurança no modelo proposto.

1.4 Metodologia

Para o desenvolvimento deste trabalho, a primeira atividade realizada foi uma revisão bibliográfica para se investigar todos os conceitos e aspectos tecnológicos acerca de redes VPN, apurando-se, em seguida, as pesquisas e trabalhos desenvolvidos na área de Segurança em Redes de Computadores envolvidos na construção de redes VPN, em especial, na interligação de redes corporativas utilizando a Internet. Posteriormente, foi desenvolvido um levantamento sobre os equipamentos, ferramentas e sistemas utilizados para construção de redes VPN, avaliando características de segurança, escalabilidade, aplicabilidade, facilidade de instalação e manutenção, estabilidade, suporte, documentação e custo. Nesta avaliação, concluiu-se que uma solução isolada para VPN não seria uma alternativa viável, pois existiam outros requisitos e serviços de segurança que necessariamente deveriam fazer parte de uma

solução VPN completa. Desta forma, houve a necessidade de se pesquisar outros sistemas de segurança, os quais são propostos e detalhados neste trabalho. Houve também uma preferência pela utilização de ferramentas e sistemas livres. Primeiramente, para se investigar o funcionamento e a viabilidade técnica de soluções livres para a construção de redes VPN, que é um dos objetivos deste trabalho. Em segundo, pelo domínio e conhecimento do autor desta dissertação nos sistemas GNU/Linux. Como resultado dessa pesquisa, ocorreu a escolha de um *software* livre para a construção de uma rede VPN de experimento para uso na investigação.

As respostas aos questionamentos deste trabalho não foram completamente obtidas mediante simples pesquisa e observação da literatura corrente, mesmo porque, as informações até então conhecidas não descrevem resultados do uso em conjunto de soluções e produtos de *software* livre para a construção de redes VPN, nem tão pouco, os problemas e as dificuldades encontradas na sua implementação. Fez-se necessário, portanto, a implementação e avaliação de uma solução VPN experimental para simular uma VPN baseada na Internet, antes mesmo, da execução em produção de uma solução VPN completa.

1.5 Estrutura da Dissertação

Esta dissertação está estruturada da seguinte forma:

No Capítulo 2, são apresentados os principais conceitos, serviços e tecnologias em segurança de redes, a fim de propiciar uma fundamentação teórica sobre as diversas tecnologias e serviços relacionados a segurança em VPNs corporativas que serão abordados ao longo deste trabalho.

No Capítulo 3, são descritos os conceitos relacionados a uma VPN e tunelamento, bem como, o seu funcionamento, modos de interconexão e tipos de VPNs.

No Capítulo 4, são apresentados os principais protocolos disponíveis para a construção de redes VPN, descrevendo inclusive o funcionamento, aplicabilidade e aspectos de segurança de cada um.

No Capítulo 5, são discutidas diversas topologias para interligação de redes corporativas utilizando-se VPN. Neste, analisa-se também cenários de *gateways* VPN com a

utilização de múltiplas redes desmilitarizadas e o posicionamento desses servidores em relação a outros serviços de segurança, como, por exemplo, o *firewall* e servidores PKI. Este capítulo também apresenta a topologia proposta para interligação de unidades corporativas através de VPN, juntamente com um estudo de caso e uma análise de custos desta proposta.

No capítulo 6, é apresentado o experimento que implementa uma solução VPN combinada a serviços de *firewall* utilizando ferramentas e sistemas livres, avaliando e demonstrando o funcionamento e viabilidade técnica na construção de VPNs nessa plataforma.

No Capítulo 7, discute-se as limitações da investigação, os trabalhos futuros e as considerações finais. Seguem-se as Referências Bibliográficas utilizadas para embasar esta dissertação e os Apêndices onde são apresentados conceitos de *software* livre, conceitos do Ambiente GNU/Linux, arquivos de configuração do FreeS/Wan, ferramenta escolhida para implementação de VPNs com o uso de IPSec, e relatórios dos testes realizados na ferramenta *Sniffer Pro*, os quais constatam o funcionamento do túnel VPN (IPSec) feito no experimento.

2 SEGURANÇA DE REDES

Para os propósitos deste trabalho, é necessário que se definam os principais conceitos, serviços e tecnologias em segurança de redes, assim como uma nomenclatura e uma estrutura comum a ser utilizada ao longo deste estudo, a fim de propiciar uma fundamentação teórica sobre as diversas tecnologias e serviços que serão abordados sobre segurança de informação em redes VPN, homogeneizando o entendimento acerca dos principais componentes de segurança de redes e dos termos que serão apresentados ao longo dessa dissertação.

2.1 Conceitos de Segurança

As empresas cada vez mais possuem informações sigilosas disponíveis em seus computadores, fazendo com que certos cuidados sejam necessários, a fim de protegê-las, como limitar o acesso físico e lógico aos computadores, principalmente para ambientes computacionais compartilhados ou que se utilizem de algum meio de comunicação público. Estes cuidados podem ser aplicados através da implantação de um conjunto de mecanismos de segurança para proteção de arquivos e de outras informações armazenadas, inclusive com a automatização de processos e ferramentas de segurança.

Outro complicador que afeta diretamente as questões de segurança nos ambientes corporativos, e também nos sistemas pessoais, é a introdução de facilidades de comunicação e de redes. A segurança nestes sistemas é necessária para proteger as informações que estão trafegando no ambiente de rede. Essa preocupação tornou-se ainda maior com a popularização

da Internet nas empresas, onde o risco das informações serem acessadas ou alteradas é grande, pois qualquer pessoa conectada à Internet pode vir a tomar posse destas informações, caso não estejam bem protegidas.

Para se avaliar efetivamente as necessidades de uma organização quanto aos aspectos de segurança adequados para cada Empresa, como por exemplo, definição de produtos e políticas de segurança que satisfaçam os requerimentos da organização, deve-se adotar uma abordagem que possa facilitar esse processo, e que possa avaliar os impactos, custos e benefícios à Organização. De acordo com (STALLINGS, 1999a), qualquer metodologia ou abordagem adotada em uma Organização deve considerar três aspectos de segurança da Informação:

- ***Ataques de Segurança***: são quaisquer ações que possam comprometer a disponibilidade, integridade, sigilo e autenticidade de uma informação pertencente a uma organização.

- ***Mecanismos de Segurança:*** são mecanismos projetados para se detectar, prevenir ou se recuperar de um ataque de segurança.

- ***Serviços de Segurança:*** são funções que aumentam o nível de segurança dos sistemas de processamento de dados e das transmissões de informação em uma Organização. Estes serviços podem utilizar um, ou mais, mecanismos de segurança.

2.2 Serviços de Segurança

Em princípio, os serviços de segurança de Informação podem ser associados aos mesmos tipos de funções de segurança aplicados a documentos físicos (STALLINGS, 1999a). Documentos tipicamente possuem assinaturas e datas; eles podem ser confidenciais; eles podem precisar de proteção contra falsificações, rasuras, possíveis intempéries ou destruição; eles podem precisar de autenticações em cartório ou de testemunhas para confirmar a autenticidade do documento; eles podem ser xerocopiados, armazenados ou licenciados, e assim por diante.

Com a evolução dos sistemas de informação, várias organizações passaram a ter seus escritórios totalmente automatizados, com a utilização mínima de papel. Com isso, as informações e documentos importantes ao negócio da Organização passaram cada vez mais a constar em mídia eletrônica, ao invés de documentos em papel. Desta forma, todos os procedimentos e funções típicas associadas à segurança de documentos em papel devem ser estendidos aos documentos em mídia eletrônica. Porém, a execução dessas funções de segurança em mídia eletrônica não é tão simples quanto parece, de acordo como (STALLINGS, 1999a), existem vários aspectos e desafios que devem ser analisados e tratados, como por exemplo:

- É usualmente possível se diferenciar um documento original em papel de uma cópia xerográfica. Entretanto, um documento eletrônico é meramente uma seqüência de bits, não há diferença entre um original e uma cópia. Além disso, uma alteração em um documento em papel original pode ser facilmente evidenciada através de rasuras ou outros sinais na superfície do papel. Enquanto que, a alteração de bits de um documento eletrônico não deixa sinais físicos para serem rastreados.

- Qualquer processo de “prova” associado a um documento físico necessitará identificar a autenticidade do documento em papel através da análise da assinatura contida no documento. Porém, a autenticidade de documento eletrônico deverá ser baseada na presença de alguma evidência interna contida no próprio documento.

Felizmente, a tecnologia de segurança da informação já “resolveu” muitos dos problemas e desafios relacionados as questões de integridade, privacidade, autenticidade e disponibilidade de informações para documentos eletrônicos, porém, deve-se entender que quando se trata de segurança, não existe uma solução totalmente segura, imune a ataques; o que existe, é uma informação, ou documento, que possui um nível de segurança maior que outro. E quanto maior o nível de segurança, maior será a confiança depositada na integridade, privacidade e autenticidade de dada informação.

Desta forma, é necessário compreender os serviços, mecanismos e ataques de segurança existentes, a fim de possibilitar uma melhor definição e implantação de uma

proposta para uma política e topologia de segurança viável, e que esteja em conformidade com requisitos econômicos e estratégicos de uma organização.

A literatura existente atualmente, acerca do tema em questão, não possui uma concordância sobre a classificação de muitos dos termos usados em Segurança na área da computação. Desta forma, este trabalho descreve a seguir uma classificação para os serviços de seguranças mais usualmente utilizados, semelhantes às descritas por Donn Park (PARK, 1994), Stallings (STALLINGS, 1999a), Tanenbaum (TANENBAUM, 2003) e por Lino Sarlo Silva (SILVA, 2003). Essa classificação de serviços de segurança é a seguinte:

- **Privacidade** – serviço que permite acesso à informação apenas a pessoas autorizadas, limitando o acesso às informações geralmente através do uso de criptografia. Uma pessoa não autorizada, mesmo de posse da informação capturada, não visualiza os dados, pois estes estarão criptografados. Um outro aspecto que pode ser aplicado ao serviço de privacidade é a proteção à análise do fluxo do tráfego, que tem o objetivo de impedir que uma entidade não autorizada possa ser capaz de observar a origem, o destino (Endereço IP origem e destino dos pacotes, por exemplo) e outras informações das características do tráfego em um canal de comunicação. Este serviço também é comumente denominado de confidencialidade ou sigilo.

- **Autenticidade** – esse serviço trata de assegurar que a comunicação é autêntica. Ele verifica se a entidade com quem está se trocando informações sigilosas é realmente quem deveria ser. Em uma comunicação entre duas entidades na rede, este serviço deve garantir que as duas entidades que trocam informações são autênticas e que a conexão não sofrerá interferência de uma terceira entidade que pode mascarar uma entidade legítima, não autorizando a transmissão e a recepção de informações. Este tipo de serviço pode utilizar assinaturas ou certificados digitais para aferir a autenticidade de uma entidade dentro da rede.

- **Integridade** – este serviço assegura que os dados não serão alterados durante uma transmissão sem o conhecimento do receptor, ou seja, nenhuma modificação no pacote original pode ser feita no decorrer de uma transmissão sem que exista a detecção da entidade receptora do pacote de que houve algum

tipo de violação. Desta forma, este serviço assegura que as mensagens sejam recebidas exatamente como foram enviadas, garantindo que as mensagens não sofrerão nenhuma inserção, duplicação, modificação, reorganização, classificação ou ataques de *replays* (salva de pacotes transmitidos por uma comunicação entre duas entidades para serem utilizados posteriormente na tentativa de forjar uma nova comunicação legítima). O Serviço de Integridade pode possuir mecanismos automáticos de recuperação ou apenas serviços de detecção de violação.

- **Não-repúdio** - este serviço é uma etapa posterior à autenticidade, podendo também ser um atributo desse serviço. Este serviço permite que quando uma mensagem é enviada, a entidade recebedora da mensagem pode provar que a mensagem foi de fato enviada pela entidade de origem alegada. De maneira similar, quando uma mensagem é recebida, a entidade que a enviou pode provar que a mensagem foi de fato recebida pela entidade recebedora alegada.

- **Controle de Acesso** – este serviço limita e controla o acesso a utilização de recursos, sistemas e *hosts* apenas a pessoas autorizadas. Cada entidade tentando conseguir acesso a algum recurso deve primeiramente se identificar, ou se autenticar, assim, os direitos de acesso são moldados para a entidade particular.

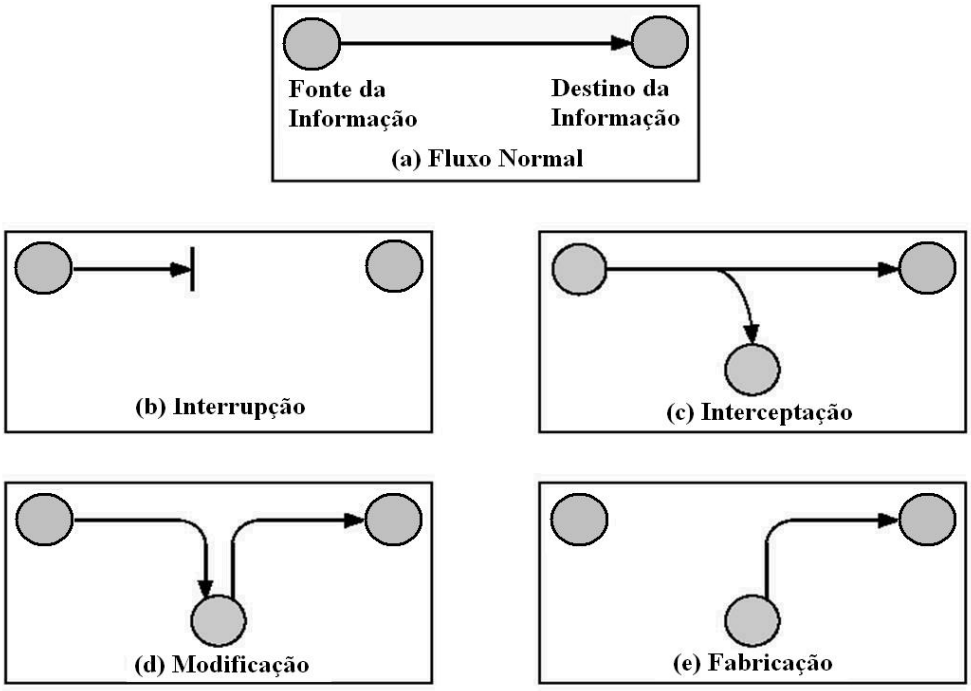
- **Disponibilidade** – este serviço tenta evitar a perda de disponibilidade dos elementos de um sistema distribuído, mesmo em caso de ataques. Atualmente, existe uma variedade muito grande de ataques que podem resultar na perda ou redução da disponibilidade.

2.3 Ataques de Segurança

Para tornar as redes mais seguras e confiáveis, deve-se estar atento às principais ameaças que podem comprometer a integridade, privacidade e autenticidade das informações de uma empresa. Estas ameaças podem ter origem interna ou externa, pois, ao contrário do que se pensa, nem sempre o principal “inimigo” está fora da rede, como um *hacker* ou

cracker[2], mas sim dentro dela, como um funcionário mal intencionado ou muito insatisfeito, que geralmente possui livre acesso aos recursos disponíveis, e que pode comprometer a integridade e a privacidade de informações estratégicas da Empresa, como por exemplo, através da destruição ou alteração de informações e o envio de informações sigilosas da Empresa a concorrentes.

(STALLINGS, 1999a) sugere que um ataque a segurança de um sistema computacional ou a uma rede de computadores pode ser melhor visualizado através de grafos direcionados que representam os fluxos de informações entre as entidades participantes da comunicação. Em geral, existe um fluxo de informação de um sistema origem, enviando um arquivo, ou alguma informação contida em uma área da memória, para um destino, como um outro arquivo ou usuário. Este fluxo normal é descrito na figura 2.1a, enquanto que os quatro esquemas restantes da figura representam categorias de ataque, que são ataques por: interrupção, interceptação, modificação e fabricação. A seguir discutiremos estas categorias de ataques.



Fonte: (STALLINGNS, 2000, p. 7)

Figura 2.1 - Ameaças de Segurança

[2] Um *haker* é um indivíduo hábil em enganar os mecanismos de segurança de sistemas de computação. Enquanto que um *cracker* é um *haker* que possui o intuito de roubar, alterar ou apagar as informações do sistema invadido.

2.3.1 Ataque de Interrupção

O Ataque de Interrupção visa destruir ou interromper o serviço oferecido, ou seja, ataca-se a disponibilidade das informações, conforme a Figura 2.1b.

O principal tipo de ataque conhecido e classificado como interrupção é o ***Denial of Service*** - **DoS**, ou negação de Serviço, que é o envio de requisições em massa para um determinado computador, de modo que o mesmo passe a não conseguir mais responder a todas elas, ficando sobrecarregado, fazendo com que os serviços de uma determinada máquina ou servidor pare de funcionar.

Outra forma comum desse tipo de ataque é a destruição de componentes de *hardware*, como por exemplo, discos rígidos, ou ainda, a interrupção proposital de circuitos de comunicação (*links*).

2.3.2 Ataque de Interceptação

O Ataque de Interceptação tem como objetivo capturar o que está sendo transmitido sem que o sistema perceba, ou seja, ataca-se a privacidade das informações. Este ataque gera cópias de informações, arquivos, ou programas não autorizados. A entidade não autorizada pode ser uma pessoa, um programa ou um computador. Um dos principais tipos de ataques desta categoria é o ***man-in-the-middle***, onde o invasor simula ser o parceiro de ambas as partes envolvidas na conexão, assumindo a identidade de um usuário válido. Esta categoria de ataque está representada pela figura 2.1c.

Outra forma bastante usual nesse tipo de ataque é a utilização de analisadores de protocolos ou ferramentas *Sniffer* que conseguem obter todos os pacotes que estão trafegando em um determinado canal de comunicação. Desta forma, um invasor pode facilmente obter ou adulterar informações confidenciais de uma Corporação, beneficiando-se das fragilidades encontradas em um ambiente de rede, como por exemplo, a ausência de criptografia nas informações confidenciais e senhas de acesso que trafegam na rede.

2.3.3 Ataque de Modificação

O Ataque de Modificação é quando existe alteração da informação que está sendo transmitida, ou seja, ataca-se a integridade da mesma. Um exemplo de ataque desta classificação é o ***Replay***, onde parte de uma transmissão da rede é copiada e reproduzida posteriormente, simulando um usuário autorizado. Este tipo de ataque está representado pela figura 2.1d.

2.3.4 Ataque de Fabricação

No Ataque de Fabricação, o atacante tem como finalidade se passar por um usuário do sistema, a fim de obter informações para transmitir dados na rede, ou seja, ataca-se a autenticidade das informações, conforme pode se ver na figura 2.1e. O tipo de ataque mais comum de fabricação é o ***IP Spoofing***, que consiste na substituição do endereço IP do computador do invasor, fazendo com que ele se passe por um computador confiável da rede, podendo assim obter privilégios na comunicação.

2.3.5 Ataques Ativos e Passivos

As ameaças também podem ser classificadas em dois tipos: ***Passivas***, onde o sistema continua a operação sem a percepção de ter um invasor na rede e, geralmente, acontece roubo de informações; e ***Ativas***, onde o invasor prejudica o sistema, atingindo os dados ou degradando os serviços (STALLINGS, 1999a), envolve geralmente modificações nos dados ou a criação de falsos fluxos de dados.

Existem dois tipos de ataques passivos. O primeiro é através dos *conteúdos de mensagens liberadas*, que é a simples “escuta” não autorizada e não perceptiva das informações contidas nas mensagens que trafegam na rede, quando duas ou mais entidades se comunicam numa rede. O segundo tipo de ataque passivo é a *análise de trafego*, que mesmo que as informações estejam criptografadas, este ataque realiza análises nos cabeçalhos dos pacotes de rede, e com isso, a entidade não autorizada pode determinar a localização e a identidade dos *hosts* que estão se comunicando; bem como, observar a freqüência e o tamanho das mensagens, o que pode ser utilizado para se determinar qual a natureza da comunicação.

Os ataques passivos são mais difíceis de serem detectados, pois os mesmos não envolvem alteração ou fabricação de dados.

2.3.6 Defesa em Profundidade

É necessário que se entenda que nenhum componente único poderá garantir um sistema de segurança adequado para uma rede corporativa, e que possa defendê-la com perfeição contra ataques (NORTHCUTT et al., 2002). Desta forma, este estudo propõe a utilização de um modelo de Defesa em Profundidade (NORTHCUTT et al., 2002) para uma Rede Virtual Privada, na qual projetam-se componentes de segurança dispostos em camadas, de forma a dificultar invasões que possam comprometer a integridade, a autenticidade e o sigilo das informações que trafegam em uma rede IPv4, definindo componentes com base nas necessidades específicas de cada empresa, no nível de segurança que se deseja atribuir a informação e nas restrições técnicas, orçamentárias e práticas.

Northcutt et al. propõe que se pense na segurança de uma rede como uma cebola:

> Quando você descasca a camada mais externa, muitas camadas permanecem por baixo dela (NORTHCUTT, 2002).

Este é o conceito que norteia a Defesa em Profundidade, o qual será sugerido neste estudo.

2.4 Política de Segurança

De acordo com Kevin Downes et al. (2000, p. 494), uma política de segurança de rede visa o controle do tráfego de rede e de sua utilização. Ela visa basicamente definir o que é permitido e o que é proibido em uma infra-estrutura de rede ou em sistema. Identifica os recursos da rede e as possíveis ameaças, define usos e responsabilidades e detalha planos de ação destinados a situações em que ocorram violações da política de segurança. Assim, esta deve ser reforçada de maneira estratégica através da definição de limites que possam ser estabelecidos na rede. Estes limites são chamados ***perímetros de rede***.

Existem basicamente duas filosofias por trás de qualquer política de segurança:

- a ***proibitiva***, onde tudo que não é expressamente permitido é proibido e

- a ***permissiva***, onde tudo que não é expressamente proibido é permitido.

Este trabalho preconiza que as redes corporativas devem sempre adotar uma abordagem proibitiva. Uma política deve descrever exatamente quais operações são permitidas em um sistema. Qualquer operação que não esteja descrita de forma detalhada na política de segurança deve ser considerada ilegal ao sistema.

Uma outra visão sobre os serviços de segurança foi elaborado por Donn Park em (PARK, 1994) onde descreve seis elementos que devem ser contemplados em qualquer política de segurança e que também devem ser considerados para a segurança de redes de computadores. São eles:

- *Disponibilidade* – O sistema deve estar disponível para uso quando o usuário precisar. Dados críticos devem estar disponíveis de forma ininterrupta;

- *Utilização* – O sistema e os dados devem ser utilizados para as devidas finalidades;

- *Integridade* – O sistema e os dados devem estar completamente íntegros e em condições de serem utilizados;

- *Autenticidade* – O sistema dever ter condições de verificar a identidade do usuário, e este deve ter condições de verificar a identidade do sistema;

- *Confidencialidade* – Dados privados devem ser apresentados somente para os donos dos dados ou para o grupo de usuários para o qual o dono dos dados permitir;

- *Posse* – O dono do sistema deve ter condições de controlá-lo.

2.5 Classificação de Redes quanto a Confiabilidade

Kevin Dowenes et al. (2000) descreve que à medida que se define uma política de segurança em uma corporação, cada rede que compõe a topologia precisa ser classificada como um dos três tipos de redes: **confiável**, **não-confiável** e **desconhecida**.

As ***Redes Confiáveis*** são normalmente aquelas localizadas no perímetro de segurança da rede, as quais possuem recursos que necessitam de proteção. Nas corporações, normalmente existe algum funcionário responsável pela administração da rede e pela aplicação das políticas de segurança definidas pela Empresa. Uma exceção para esta regra é a inclusão de uma VPN, pois esta se caracteriza como uma rede confiável, fora do perímetro de segurança da Empresa, transmitindo dados por uma infra-estrutura de redes não-confiáveis.

As ***Redes não-confiáveis*** são aquelas que se sabe estarem fora do perímetro de segurança, as quais não possuem controle da administração ou das políticas de segurança. Estas, por exemplo, são explicitamente identificadas durante a configuração de um servidor *firewall*.

As ***redes desconhecidas*** são redes que não são confiáveis, nem não-confiáveis. São aquelas desconhecidas para um *firewall* ou *Gateway* VPN, pois não é possível informar, de modo explícito, se a rede é confiável ou não-confiável. Porém, estas últimas são consideradas não-confiáveis para os propósitos de segurança.

2.6 Perímetros de Segurança e seus Componentes

Stephen Northcutt et al. (NORTHCUTT; ZELTSER; WINTERS; FREDERICK; RITCHEY, 2002) definem ***Perímetro*** como sendo a ***borda fortificada*** de uma rede de computadores, que pode incluir os seguintes componentes: Roteadores, *Firewalls*, *Proxys*, Sistemas de Detecção de Intrusos (SDI), Dispositivos VPN, *software*, DMZs e Screened Subnets. Todos esses componentes serão apresentados neste capítulo.

Porém, essa definição é um tanto quanto simplista, pois, na realidade, as redes corporativas podem conter vários perímetros dentro de um perímetro de segurança (NORTHCUTT et al., 2002). Normalmente, é estabelecido um conjunto de perímetros de

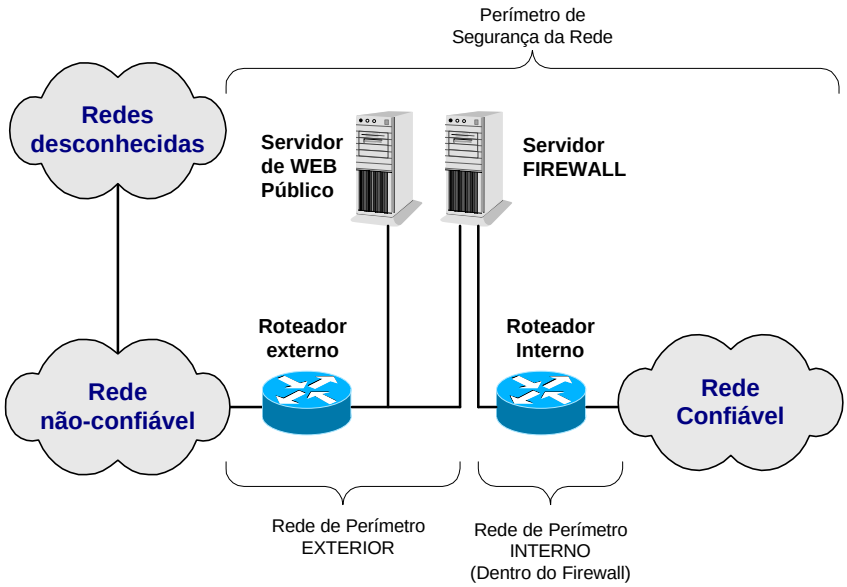
redes. Para isso, é necessário designar as redes que serão protegidas e definir os mecanismos de segurança de rede que exercerão esta proteção para cada perímetro.

Em geral, são encontrados dois tipos de perímetros de rede: o ***perímetro exterior*** e o ***perímetro interno***. É comum, a existência de mais de um perímetro interno para uma mesma rede. Estes perímetros internos são relacionados a um recurso particular que se pretende proteger, podem ainda representar limites adicionais, nos quais existem outros mecanismos de segurança.

Normalmente, o ***perímetro exterior*** representa o ponto de separação entre os recursos que estão sob controle e os recursos que não estão sob controle, com exceção da rede VPN, onde o circuito virtual estabelecido, que utiliza uma infra-estrutura pública e compartilhada de comunicação, necessita de controle e segurança fim-a-fim.

(DOWNES et al., 2000) define que para dispor de um perímetro de segurança de rede bem sucedido, o *firewall* precisará ser o gateway para todas as comunicações entre redes confiáveis e não-confiáveis.

A figura 2.2 apresenta um diagrama que representa um exemplo de uma estrutura de segurança de rede com dois perímetros, os quais são definidos pela posição dos roteadores internos e externos e do servidor *Firewall*.



Fonte: (DOWNES et al., 2002, p. 495)

Figura 2.2 - Exemplo de uma estrutura de rede de dois perímetros

Observando agora o esquema apresentado na figura 2.2, o qual o *firewall* se posiciona entre os roteadores interno e externo, pode-se concluir que a inclusão desse *firewall* nesta posição proporciona uma proteção adicional contra ataques que possam ocorrer em qualquer um dos lados. Além disso, existe um aumento de desempenho do servidor de *firewall*, pois há uma redução no tráfego que o *firewall* precisa avaliar em função do seu posicionamento. O tráfego que se origina de redes não-confiáveis destinado ao Servidor WEB, por exemplo, não é avaliado pelo *firewall*. Neste caso, dizemos que a rede exterior é uma rede desmilitarizada, conceito que será explicitado posteriormente. No momento, cabe apenas perceber que esta área desmilitarizada terá maior probabilidade a ataques devido a sua facilidade de acesso.

Vale ressaltar que caberá às próximas seções dessa dissertação um detalhamento maior sobre cada elemento de um perímetro de segurança. A partir daí, poderá se compreender melhor todos os elementos envolvidos em uma topologia de segurança baseada em perímetros. A seguir, os principais componentes de um perímetro de segurança serão detalhados.

2.6.1 Roteador de borda

É o roteador do perímetro exterior, ou seja, é o último roteador que se pode controlar antes da rede não-confiável, como a Internet. Em uma corporação que acessa a Internet, todo o tráfego de rede que possui origem ou destino à Internet passa por este roteador. Desta forma, ele funciona como a primeira e a última linha de defesa de uma rede através da filtragem de pacotes inicial e final.

2.6.2 Firewalls

Tanenbaum faz uma analogia interessante sobre *firewalls*, ele descreve:

> Os *firewalls* são apenas uma adaptação moderna de uma antiga forma de segurança medieval: cavar um fosso profundo em torno do castelo. Este recurso forçava todos aqueles que quisessem entrar ou sair do castelo a passar por uma única ponte levadiça, onde poderiam ser revistados por guardas (TANENBAUM, 2003, p. 776).

Aproveitando essa analogia, verifica-se que o *firewall* funciona como a ponte levadiça, pois todos os pacotes, entrando ou saindo do perímetro interno, devem passar obrigatoriamente pelo *firewall*, sendo submetidos às regras de segurança e filtragem definidas

neste *firewall*, enquanto os processos que aplicam as políticas de segurança podem ser vistos como os guardas do castelo que cumprem as determinações do Monarca.

Posteriormente, discutiremos algumas regras de filtragem que acreditamos serem obrigatórias quando do uso de VPNs.

Existem vários tipos de *firewalls* diferentes, inclui-se na lista: os ***filtros de pacote estático***, ***firewalls com estado*** e os ***firewalls Proxy***.

2.6.2.1 Filtros de Pacotes Estáticos

Os filtros de pacotes estáticos são componentes em uma rede que inspecionam as informações básicas dentro de um pacote. Os filtros de pacotes podem ser roteadores, como os roteadores da CISCO, ou *firewalls*, como o *IPTables* existente nas distribuições do Linux.

Este dispositivo, que geralmente é um roteador, é a primeira camada de fora para dentro e a última camada de dentro para fora em uma topologia de segurança de redes. Desta forma, os filtros de pacotes contribuem para a defesa em profundidade, filtrando o tráfego antes que ele entre ou saia da rede.

Os filtros de pacotes estáticos, como os roteadores, são mais rápidos para filtrar tráfego do que os *firewalls* com estado ou *Proxy*. Essa velocidade é importante e útil principalmente quando já se está sob ataque ou quando o *firewall* já está sobrecarregado.

2.6.2.2 Firewalls com Estado

Os *firewalls* com estado monitoram as conexões em uma tabela de estado, a qual armazena o seu banco de regras, bloqueando todo o tráfego que não esteja em sua tabela de conexões estabelecidas. Este banco de regras determina o IP e a porta de origem e de destino que são permitidos para estabelecer conexões.

2.6.2.3 Firewalls Proxy

O *firewall Proxy* é o tipo mais avançado de *firewall* atualmente existente. Ele possui todas as características e funcionalidades do *firewall* com estado, porém, possui serviços mais avançados e de alto nível, pois impede que os *hosts* interno e externo se comuniquem diretamente. Em vez disso, o *firewall* Proxy age como um intermediário entre os *hosts*.

Os *firewalls Proxy* examinam de forma mais minuciosa os pacotes para garantir que apenas o tráfego concordante com o protocolo atravesse o *firewall*, diminuindo, com isso, a possibilidade de tráfego malicioso que esteja entrando ou saindo da rede.

2.6.3 Sistemas de Detecção de Intrusos - SDI

O *Intrusion Detection System* (IDS), ou Sistema de Detecção de Intrusos (SDI), tem como principal objetivo analisar o tráfego a fim de detectar tentativas de invasões, como atividades de reconhecimento ou tentativas de se explorar alguma vulnerabilidade. Esta ferramenta roda constantemente em *background* e somente gera uma notificação quando detecta alguma situação que seja suspeita ou ilegal. Tradicionalmente, os sistemas de detecção de intrusão não interferem no tráfego da rede. Ao contrário de um *Firewall*, que toma decisões sobre qual tráfego permitir, um sensor de detecção de intrusão é, na realidade, um farejador de pacotes que também realiza análise. Porém, alguns sistemas SDI atuais estão começando a dar respostas ativas para o tráfego suspeito, como terminar conexões TCP suspeitas.

Os SDIs, ou IDSs, são excelentes mecanismos que podem ser adicionados na arquitetura de Defesa em Profundidade de uma rede. Eles podem ser utilizados para identificar vulnerabilidades e fraquezas em seus dispositivos de proteção de perímetro, como por exemplo, *firewalls* e roteadores.

Atualmente, um sistema de *firewall* que bloqueia portas e serviços, infelizmente, não é suficiente, pois muitos dos ataques e problemas de segurança existentes, que vão desde vulnerabilidades e anomalias até problemas da própria política de segurança, estão postados justamente onde é necessário que o acesso seja permitido, além disso, em um ambiente que utiliza apenas *firewall*, não há como saber o que está trafegando pelos sistemas em rede, se é hostil ou não, ou se é permitido ou não (OLIVA, 2001). O SDI vai um pouco mais além do *firewall*, no sentido de desconfiar sempre, mesmo daquele tráfego que já fora permitido pelo *firewall*, pois é possível que algum tráfego malicioso possa fingir-se de material apropriado e se aderir ao protocolo, enganando assim um *firewall*. Porém, este tráfego poderá ser analisado pelo SDI posteriormente.

(NORTHCUTT et al., 2002) e (CASWELL et al., 2003) exemplificam que uma solução de SDI bem configurada pode identificar e alertar o Administrador de uma rede diante das seguintes situações de ataques:

- *Ataques de transbordamento (estouro) de buffer*. Atualmente, estes ataques apresentam uma grande porcentagem das explorações em redes corporativas. Geralmente, os transbordamentos estão relacionados a falhas de programação específicas a um Sistema Operacional ou a um *software* aplicativo. Quando ocorre o ponto de falha, isto é, o *estouro de uma pilha*, como é denominado por muitos, o invasor tem a possibilidade de inserir um código danoso neste ponto de falha (na pilha de processos do computador) e conseguir o controle do sistema. Esta é uma das ameaças mais destrutivas e dispendiosas existentes.

- *Tentativas de exploração do Servidor DNS*. Esta detecção é feita pelo SDI através da "leitura" de tentativas de pedidos de transferência de zona DNS e tentativas de verificação da versão e nome DNS vindos de *hosts* não-autorizados. As entradas de um DNS possuem nomes de componentes de rede internos, endereços IPs e outras informações privadas sobre a rede que, sendo acessadas por um invasor, podem facilitar, e muito, o acesso a recursos da rede corporativa como: Bancos de Dados, Servidores de Arquivos, entre outros.

- O SDI pode detectar tentativas de *login* de Administrador e reconhecer programas de adivinhação de senhas, bem como o acesso a aplicativos críticos, notificando o Administrador de possíveis falhas ou tentativas de invasão.

- Tentativas de execução de programas de controle remoto na rede, como os cavalos de tróia.

- Tentativas de acesso a Banco de Dados. O ***Snort 2***, um conhecido SDI para Linux, possui um conjunto de regras projetadas para proteger contra explorações em Bancos de Dados como o ORACLE, MySQL, entre outros.

- Detecção e Varredura de conteúdos de e-mails com vírus. Alguns SDI, como o ***Snort***, podem detectar conteúdos de e-mails com vírus e ainda proteger

simultaneamente o servidor de e-mail contra ataques diretos. Os servidores de e-mail estão geralmente acessíveis para a Internet e, assim, ficam vulneráveis a ataques.

- Varreduras de exploração de rede.

- Detectar vulnerabilidades e fraquezas em seus dispositivos de proteção de perímetro, como os *firewalls* e roteadores.

- Detecção e Monitoramento de Ataques a servidores Web.

- Propagação de vermes.

É inconteste a importância de um SDI dentro do contexto de defesa em profundidade. O SDI age como mais um nível de segurança. Sabe-se que muitos ataques a um *host*, por exemplo, podem não danificar informações contidas neste, mas podem simplesmente extrair informações importantes, como um arquivo de senhas, por exemplo. Sem a detecção, o administrador da rede não perceberá esses eventos até que seja tarde demais. Um atacante pode, por exemplo, disparar ataques contra um servidor DNS de uma rede a fim de buscar "*estouros de buffer*" e, conseqüentemente, caso ocorra sucesso, o atacante pode ser capaz de realizar ações não autorizadas. É importante mencionar que este não é um cenário de ataque hipotético, várias ferramentas de ataque e vermes utilizam exatamente esta técnica para sondar, atacar e comprometer servidores.

De acordo com o seu campo de ação, os sensores ainda podem ser classificados de dois tipos:

- os ***Sensores de Rede***, ou **SDIR**, que monitoram um determinado segmento de rede, os quais devem localizar-se em segmentos estratégicos, observando o tráfego da rede, o formato de pacotes, entre outros fatores. Normalmente o SDIR opera em *modo promíscuo* para monitorar o tráfego da rede, isto é, analisa todos os pacotes que trafegam na rede, inclusive aqueles destinados a outros endereços de controle de acesso ao meio (Endereço MAC).

- os ***sensores de hosts***, também denominados sensores **SDIH**, que ficam dentro de alguns servidores críticos, observam as ações realizadas no sistema operacional, as ações dos serviços e o comportamento da pilha TCP/IP, protegendo apenas o sistema *host* em que ele reside. Ele opera em modo não-promíscuo. Uma das vantagens no uso deste tipo de sensor é a capacidade de personalizar o conjunto de regras para uma necessidade específica.

Os sensores devem interagir entre si a fim de construírem uma matriz de eventos que tem por objetivo a qualificação do padrão de ataque, minimizando, desta forma, a ocorrência de alertas falsos.

Outras características fundamentais de um SDI são: o gerenciamento centralizado, a possibilidade do sensor interagir com outros elementos de rede como *firewall*, roteadores e consoles de gerenciamento; e a possibilidade de construir uma base de conhecimento centralizada de forma a permitir uma visão ampla do nível de segurança da rede.

Desta forma, quando algum ataque for detectado pelos sensores, tornam-se possíveis ações de contra-ataque que podem ser: envio de e-mail para o administrador, ativação de alertas nas estações de gerência via SNMP, reconfiguração de elementos de rede como *firewall* e roteadores, e até mesmo o encerramento da conexão através do envio de pacotes de *reset* (flag RST do TCP) para a máquina atacante e para a máquina atacada, com o objetivo de descarregar a pilha TCP.

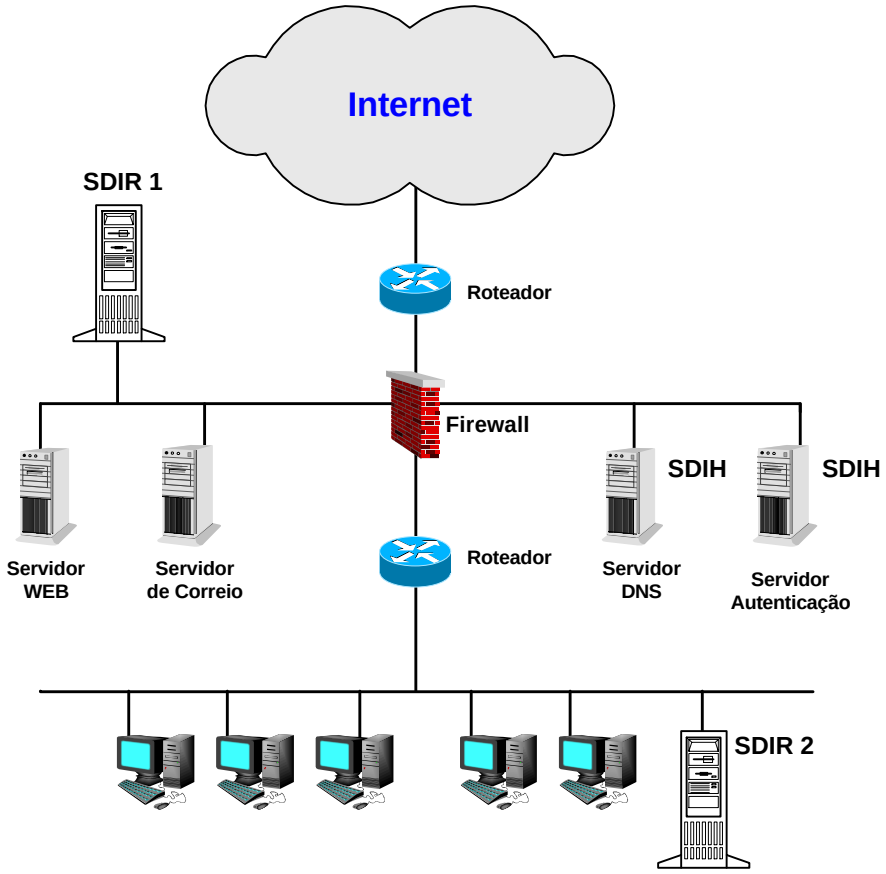
Os alertas de um SDI normalmente são gerados por dois métodos que serão posteriormente explorados neste trabalho:

- A *detecção de anomalia,* conta com a análise estática para identificar o tráfego que esteja fora daquilo que é previsto no ambiente de rede.

- A *detecção de assinatura,* procura detectar um ataque através de comparações entre padrões de assinaturas de ataque e o tráfego da rede. Quando os padrões de assinatura para um ataque correspondem ao tráfego observado na rede, um alerta é gerado.

Na figura 2.3 apresentamos uma rede usando dois SDIRs que foram colocados em segmentos estratégicos da rede, podendo monitorar os dispositivos ou servidores que estão

nesses segmentos; neste caso, o SDIR 1 está monitorando o Servidor WEB e o Servidor de Correio e o SDIR 2 está monitorando e protegendo os sistemas host internos, diminuindo assim a exposição ao comprometimento interno. Ainda neste cenário, a rede apresentada na figura utiliza dispositivos SDIH nos servidores de DNS e de Autenticação, protegendo apenas estes sistemas.



Fonte: Adaptada de (CASWELL, 2003, p. 4 - 5)

Figura 2.3 - Cenário de uma rede utilizando dispositivos SDIR e SDIH.

O *software* de SDI proposto neste trabalho é o ***Snort v.2***, disponível para o Sistema Linux. O *Snort* é um Sistema de Detecção de Intrusos baseado em Rede (SDIR) de código fonte aberto. Ele é baseado em Assinaturas e usa regras para verificar a existência de pacotes suspeitos em um segmento de rede. Deixaremos a sua instalação e configuração para um trabalho futuro. Informações mais detalhadas sobre o referido *software* podem ser encontradas em (CASWELL, 2003) ou no site www.snort.org .

2.6.4 DMZs e Screened Subnets

Uma DMZ (*DeMilitarized Zone*), ou Zona Desmilitarizada, e a *Screened Subnet* são pequenas redes que geralmente contém serviços públicos que são conectados diretamente ao *firewall* ou a outro dispositivo de filtragem e que recebem proteção desse dispositivo. Algumas bibliografias usam os termos DMZ e *Screened Subnets* indistintamente; outras, como (NORTHCUTT; ZELTSER; WINTERS; FREDERICK; RITCHEY, 2002), definem que uma DMZ é uma área desprotegida localizada na frente do *firewall*, enquanto uma *Screened Subnet* está atrás dele, conectada a uma interface dedicada de um *firewall* ou outro dispositivo de filtragem, hospedando normalmente serviços públicos como Servidores de DNS, Correio Eletrônico e WEB. Nesta visão, servidores em uma *Screened Subnet* estão bem mais protegidos, quando comparados a servidores na DMZ.

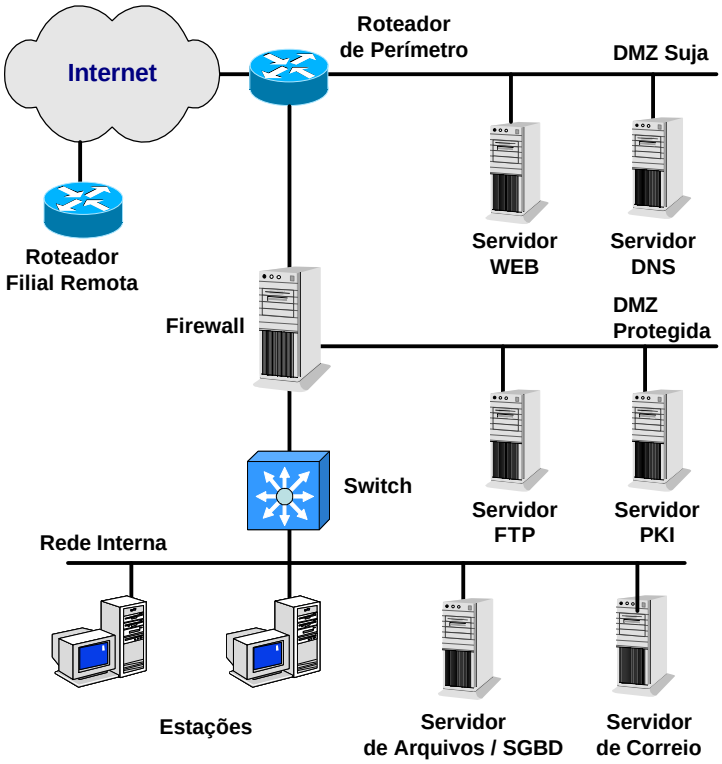
Michael Wenstrom, em (WENSTRON, 2002), já não faz distinção entre DMZs e *Screened Subnet*, porém, descreve uma sutil diferença de nomenclatura de uma DMZ, em relação ao posicionamento da rede desmilitarizada e o *firewall*, apresenta a **DMZ suja** e a **DMZ protegida**. Fazendo uma analogia com a definição de Northcutt et al. (2002), a DMZ protegida é exatamente uma *Screened Subnet*, enquanto que a DMZ suja é exatamente a DMZ localizada após o *firewall*. Neste trabalho, trataremos indistintamente DMZs e *Screened Subnets*, faremos referência apenas às DMZs, ou Zonas Desmilitarizadas, salvo quando explicitado.

É recomendável que se coloque os seus servidores acessíveis externamente, como servidores Web, FTP, correio eletrônico, DNS, entre outros, em uma DMZ. A principal importância disso é proteger a rede interna contra ataques provenientes dos servidores externos, pois sempre deve existir a precaução contra a eventualidade de que um destes servidores seja comprometido. Por exemplo, suponha que um atacante invada o servidor Web e instale um *sniffer* na rede. Se este servidor Web estiver na rede interna, a probabilidade dele conseguir capturar dados importantes (tais como senhas ou informações confidenciais) é muito maior do que se ele estiver em uma rede isolada.

É importante ressaltar que a segurança do perímetro pode ser implementada de diversas maneiras diferentes. Não existe um modelo rígido. O que definirá a topologia e o posicionamento dos servidores de rede será a própria política de segurança da Empresa. Ela definirá o que deverá ser protegido, qual o nível de segurança necessário para cada

componente da rede, qual o orçamento disponibilizado para a segurança, entre outros fatores. Obviamente, cada situação deverá ser avaliada a fim de se adotar a melhor topologia possível dentro dos requisitos de segurança, de disponibilidade da rede e diante dos recursos disponíveis. É evidente que quanto maior o nível de segurança a ser implementado, maior será a complexidade das configurações dos componentes da rede e, consequentemente, menor o desempenho. É uma questão de avaliar riscos, custos e benefícios.

A figura 2.4 apresenta uma topologia de uma rede fictícia que possui um roteador de perímetro que estabelece o ponto de demarcação entre a rede desprotegida (Internet) e uma rede protegida. Nessa topologia, existem duas DMZs, a primeira, logo após o roteador de perímetro, é uma zona desmilitarizada semiprotegida, identificada como DMZ suja; a segunda, que é um segmento de rede a partir do *firewall*, é uma DMZ protegida. Nesta zona, já existe a atuação do *firewall*, portanto, é um segmento mais protegido contra ataques do que o primeiro. Abaixo do *firewall*, temos o domínio interno da rede corporativa.



Fonte: Adaptado de (WENSTROM, 2002, p. 212)

Figura 2.4 - Exemplo de um sistema de segurança de perímetro

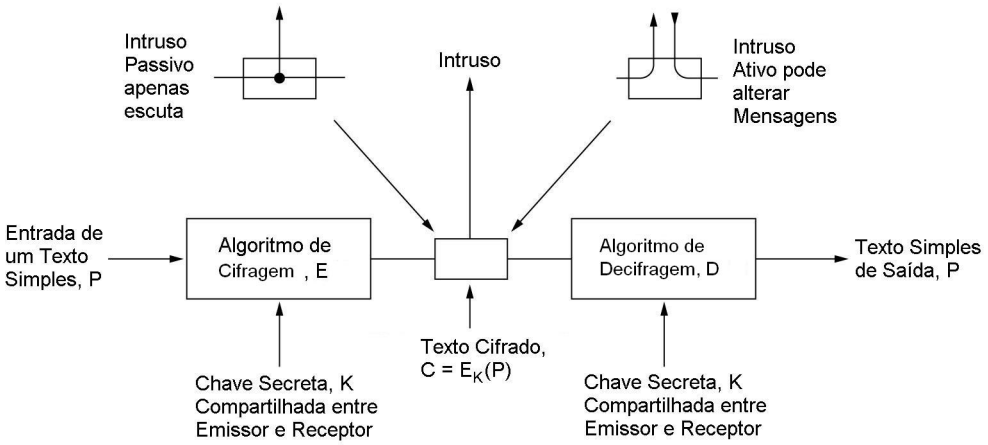
2.7 Fundamentos de Criptografia

A eficiência e eficácia dos serviços de segurança em ambientes de redes, como a privacidade, autenticidade, integridade, não-repúdio e controle de acesso, está diretamente relacionada a técnicas de criptografia utilizadas.

A palavra criptografia significa “escrita secreta”, cujo nome vem do grego *kryptos*, que significa oculto ou secreto, e *graphen*, que significa escrever (KAUFMAN; PERLMAN; SPENCINER, 2002). A criptografia pode ser definida como arte ou ciência que utiliza códigos e cifras para ocultar uma informação. Já a palavra cifra vem do hebraico *saphar*, que significa dar números. Uma cifra é uma transformação de caractere por caractere ou de bit por bit, sem levar em conta a estrutura lingüística da mensagem. Em contraste, um código substitui uma palavra por outra palavra (TANENBAUM, 2003). Atualmente, os códigos não são mais usados.

O processo de criptografia pode ser descrito da seguinte forma: um emissor gera uma mensagem original chamada *plain text*, ou texto simples[3], e utilizando-se de uma chave e um algoritmo de cifragem, gera um *cipher text*, ou texto cifrado, o qual é transmitido para um receptor. Ao chegar ao receptor, este texto passa pelo processo inverso, chamado de decifragem, resultando no texto simples original. Presume-se, então, que a mensagem cifrada seja incompreensível para quem não tem autorização de lê-la, pois tal intruso, ao contrário do destinatário pretendido, não possui a chave para decifrar a mensagem cifrada, portanto, não poderá fazê-lo com muita facilidade. Em algumas situações, este intruso além de escutar o que passa em um canal de comunicação (intruso passivo), poderá gravar as mensagens e reproduzi-las posteriormente antes que estas cheguem ao receptor (intruso ativo) (TANENBAUM, 2003). Na figura 2.5, é apresentado um modelo de criptografia simplificado e convencional que exemplifica este cenário; posteriormente, será denominado e entendido que este esquema refere-se a uma criptografia de chave simétrica.

[3] Algumas literaturas utilizam o termo *texto plano*, uma tradução ao pé da letra de *plain text*. Porém, neste trabalho será utilizado esta denominação por acharmos mais conveniente e usual.



Fonte: Adaptada de (TANENBAUM, 2003, p. 725)

Figura 2.5 – Modelo de Criptografia (cifra de chave simétrica)

De acordo com (STALLINGS, 1999a), existem dois requerimentos básicos para o uso seguro de um modelo de Criptografia convencional apresentado na figura 2.5 (criptografia de chave simétrica):

- Precisa-se de um algoritmo forte, onde um invasor não deverá ser capaz de decifrar o texto cifrado ou de descobrir a chave secreta, mesmo que ele conheça o algoritmo de cifragem ou que possua um número de textos cifrados, juntamente com seus textos simples que produziram cada texto simples.

- Tanto o emissor, quanto o receptor devem obter cópias das chaves secretas através de um meio seguro e que estas devem permanecer seguras, pois se alguém tiver acesso a esta chave e conhecer o algoritmo de cifragem, todas as informações trafegadas poderão ser lidas.

2.7.1 Notação

Este trabalho utilizará uma notação comumente utilizada para estabelecer uma relação entre texto simples, texto cifrado e as chaves utilizadas. Utilizar-se-á de uma notação $\mathbf{C = E_K(P)}$ para representar que a cifragem do texto simples **P** usando uma chave **K** gera o texto cifrado **C**. De maneira similar, $\mathbf{P = D_K(C)}$ representa a decifragem de **C** para se obter o texto simples **P**, utilizando-se a chave ***K***. Nesta notação, entende-se que **E** e **D** representam as funções de cifragem e decifragem respectivamente.

2.7.2 Princípio de Kerckhoff

Já foi dito que um dos requerimentos para o uso seguro de um modelo de criptografia é a utilização de algoritmos fortes. Aparentemente, pode parecer que um algoritmo secreto tende a ser mais forte que um algoritmo amplamente conhecido, porém, esta técnica, chamada de ***segurança pela obscuridade***, não parece que funcione adequadamente em todos os casos (TANENBAUM, 2003). A segurança não deve repousar na obscuridade do sistema. Tornar o algoritmo público possibilita que a segurança do método seja verificada e homologada por vários criptólogos. Uma vez que especialistas tenham tentado decodificar o algoritmo por vários anos seguidos, sem sucesso, prova-se efetivamente que o algoritmo é sólido. Portanto, a utilização de um algoritmo secreto de critptografia pode gerar a falsa impressão de que o mesmo é bastante seguro, simplesmente pelo fato dele não ser conhecido.

O fato, em questão, é que o segredo deve residir nas chaves, e que o algoritmo de cifragem pode ser de conhecimento público. Esta idéia foi descrita pelo holandês August Kerckhoff em (KERCKHOFF, 1983) onde ele descreve que uma cifra deve permanecer segura, mesmo que um criptoanalista inimigo conheça todos os detalhes dos algoritmos de cifragem e decifragem empregados (KERCHHOFF, 1983). Isto significa que os *algoritmos podem ser públicos, porém, as chaves devem ser secretas*. Outros princípios importantes, que norteiam um processo de cifragem, são:

- O sistema de criptografia deve ser, se não teoricamente inquebrável, não quebrável na prática. A idéia deste princípio é que a chave poderá até ser descoberta, porém, o esforço e o custo demandado para se quebrar essa chave é tão grande que não compensa tamanho esforço, mesmo que o criptoanalista possua conhecimento de detalhes do algoritmo, de textos cifrados e textos simples correspondentes.

- O compromisso com a criptografia não deve gerar inconvenientes para as entidades participantes do Sistema.

- A chave deve ser facilmente recuperável e de fácil substituição.

A seguir, detalharemos essa classificação.

2.7.3 Classificação dos Sistemas de Criptografia

A criptografia pode ser genericamente classificada em três diferentes dimensões:

- quanto aos ***tipos de cifras utilizadas***, isto é, quanto aos tipos de operações utilizadas na transformação do texto simples para o cifrado;
- quanto a ***simetria das chaves utilizadas***: Criptografia Simétrica e Assimétrica, e
- quanto ao ***modo de operação de cifra***, isto é, a maneira como o texto simples é processado.

2.7.4 Tipos de Cifras

Todos os algoritmos de cifragem são baseados em dois tipos de categorias de operações, as quais transformam o texto simples em texto cifrado: as ***cifras de substituição***, onde cada elemento do texto simples (bit, letra, grupo de bits ou letras) é substituído (mapeado) para um outro elemento correspondente, sendo sempre o mesmo elemento, e as ***cifras de transposição***, onde os elementos do texto simples são reordenados (embaralhados).

As cifras de substituição preservam a ordem dos símbolos no texto cifrado, mas disfarçam esses símbolos. Por outro lado, as cifras de transposição reordenam os elementos do texto, mas não as disfarçam (TANENBAUM, 2003).

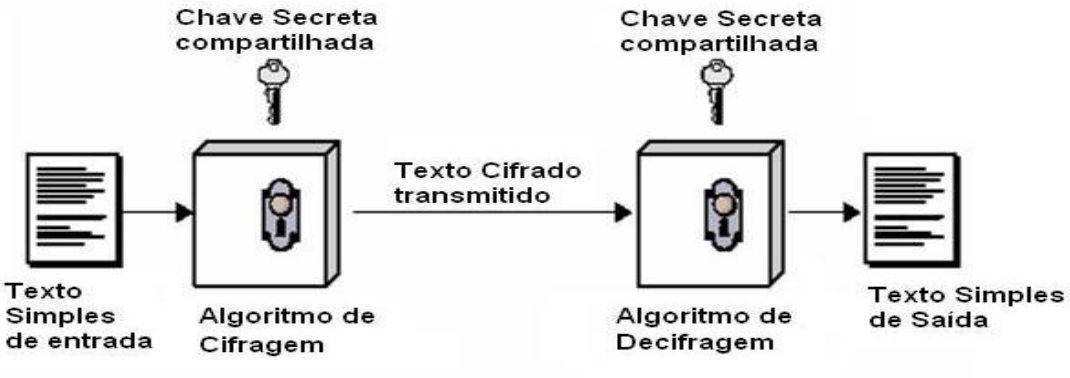
2.7.5 Criptografia Simétrica e Assimétrica

Se tanto o emissor, quanto o receptor em uma comunicação criptografada utilizam a mesma chave, o sistema é denominado de Sistema de Criptografia Simétrica, de única-chave ou de sistema de criptografia convencional. Se as entidades dessa comunicação utilizam diferentes chaves, o sistema é referenciado como um Sistema de Criptografia Assimétrica, de duas chaves ou de Criptografia de chave pública.

2.7.5.1 Criptografia Simétrica

A criptografia simétrica, como o nome sugere, baseia-se na simetria das chaves, ou seja, a mesma chave utilizada para criptografar será usada para decifrar a mensagem. A figura

2.6 apresenta este cenário, onde existe uma chave secreta compartilhada entre emissor e receptor, porém, essa chave é previamente trocada entre o emissor e o receptor por um canal de comunicação seguro.



Fonte: (STALLINGS, 1999a, p. 22)

Figura 2.6 – Modelo de Criptografia Simétrica

A criptografia simétrica é também comumente referenciada como criptografia de chave secreta (*secret key*), de chave única (*single key*) ou criptografia convencional.

As desvantagens deste processo devem-se ao fato de que como apenas uma chave é utilizada para cada par de entidades comunicantes, a segurança em cima dela deve ser rígida e, se o número de entidades que queiram comunicar-se de forma segura for muito grande, serão necessárias inúmeras cópias de chaves distribuídas, o que dificultará ainda mais a gerência das mesmas.

Pode-se citar os seguintes algoritmos simétricos e os respectivos tamanhos das chaves geradas por cada um deles:

- *Data Encryption Standard* –DES (Algoritmo *Lucifer*), com chaves de 56 bits. Criptografa em blocos de 64 bits, utilizando 16 estágios (rodadas). Consiste em um dos algoritmos mais rápidos atualmente em uso.

- *Triple Data Encryption Standard* -3DES, que utiliza duas chaves de 56 bits (chave de 112 bits). Foi uma adaptação do algoritmo DES para torná-lo mais forte, utilizando três chamadas do próprio algoritmo DES e duas chaves de criptografia. Maiores informações sobre a proposta deste algoritmo podem ser encontradas em (TUCHMAN, 1979).

- *Advanced Encryption Standard* – AES (Algoritmo ***Rijndael***), utiliza chaves de 128 bits até 256 bits. É um algoritmo muito forte. Para maiores detalhes sobre este algoritmo deve-se consultar (DAEMEN, RIJMEN, 2002). O site oficial do Algoritmo, acessado pela URL www.esat.kuleuven.ac.be/~rijmen/rijndael/, também trás muitas informações sobre o mesmo, inclusive a possibilidade de realizar *download* da documentação e de implementações em C e em Java desse algoritmo de criptografia.

- *International Data Encryption Algorithm – IDEA*, – utiliza chaves de até 448 bits. Considerado um bom algoritmo, mas patenteado (TANENBAUM, 2003).

- *Twofish*, que utiliza chaves de 128, 192 ou 256 bits. Algoritmo de Criptografia muito forte e amplamente utilizado.

- *Serpent*, que utiliza chaves de 128, 192 ou 256 bits. Algoritmo de Criptografia muito forte.

Cabe ressaltar que este trabalho não possui objetivo de descrever detalhes sobre os algoritmos de criptografia; bem como, não possui interesse, neste momento, em realizar estudos comparativos entre os algoritmos de criptografia, analisando questões de desempenho, de ocupação de memória, entre outros fatores. Porém, também ressalta a grande importância e relevância desse tipo de pesquisa em trabalhos futuros com o objetivo de investigar a aplicabilidade e a adequação de algoritmos de criptografia em soluções VPN.

Observa-se, entretanto, que os algoritmos simétricos mais conhecidos, destacando-se os algoritmos do DES-3 (Lúcifer), o AES (Rinjdael), o Twofish e o Serpent, são implementados nos principais *softwares* de VPN, inclusive no *FreeS/Wan* (*software* VPN analisado neste trabalho). A razão principal dessa grande variedade reside principalmente no fato em que muitas soluções VPN adotam suporte ao IPSec e, consequentemente, são abertos a qualquer algoritmo de criptografia.

2.7.5.2 Criptografia Assimétrica

A criptografia assimétrica, por sua vez, envolve o uso de duas chaves distintas, uma privada e outra pública. Pode-se utilizar qualquer uma das chaves para cifrar a mensagem.

Entretanto, somente a chave inversa deve ser utilizada para decifrá-la. Por exemplo, se um emissor utiliza a chave pública do receptor para cifrar a mensagem, o receptor deve utilizar a sua chave privada para decifrá-la.

A criptografia assimétrica foi proposta e publicada pela primeira vez por dois pesquisadores da Universidade de Standford, Diffie e Hellman em 1976 (STANDFORD; DIFFIE; HELLMAN, 1976). Essa publicação propôs um sistema de criptografia radicalmente novo, no qual as chaves de criptografia e descriptografia eram diferentes, revolucionando assim a criptografia, que até então era uma ciência baseada somente em chaves simétricas. A proposta de chaves assimétricas estabeleceu um progresso significativo na criptologia e, por conseguinte, propiciou avanços nos sistemas de criptografia, nas suas aplicações e na literatura. A utilização de chaves assimétricas teve profundas conseqüências nos serviços de privacidade, na distribuição de chaves e nos serviços de autenticação. Em sua proposta, o algoritmo de criptografia **E** e o algoritmo de descriptografia **D**, ambos chaveados por suas chaves assimétricas ($\mathbf{K_1}$ e $\mathbf{K_2}$), tinham que atender aos seguintes requisitos:

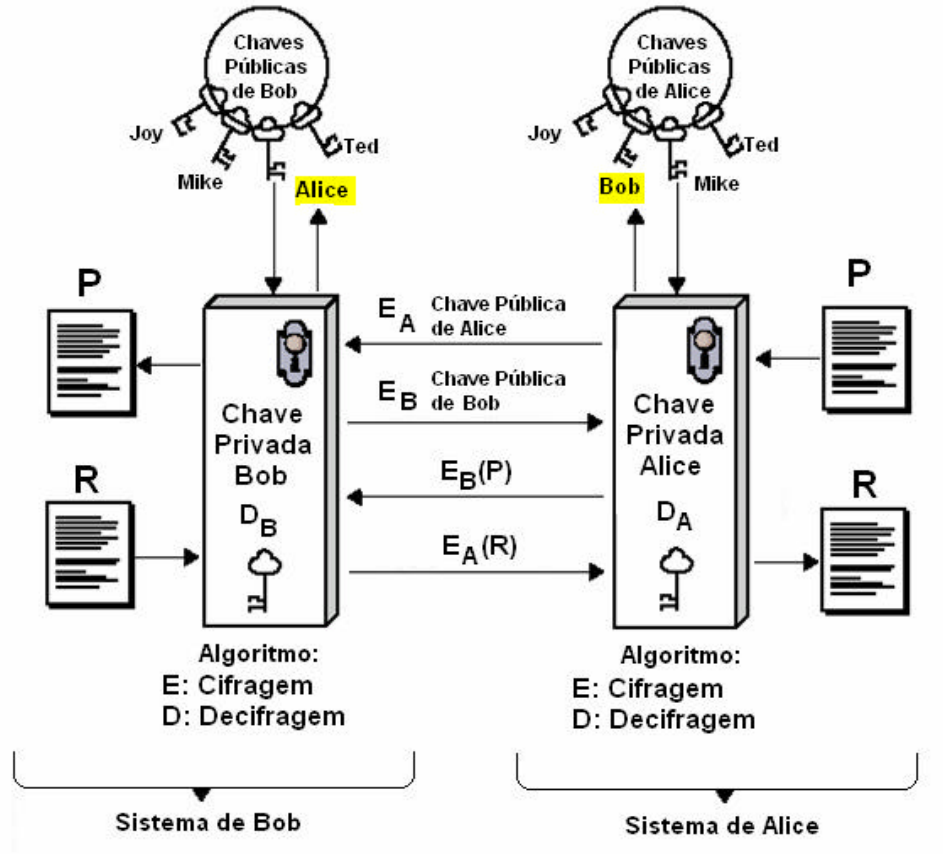
- $\mathbf{D_{K1}(E_{K2}(P)) = P}$: Aplicando-se **D** a uma mensagem criptografada, $E_{K2}(P)$, obtém-se novamente o texto simples **P**.

- Deve ser extremamente difícil deduzir a chave $\mathbf{K_1}$ conhecendo-se a chave $\mathbf{K_2}$, e vice-versa.

- O texto cifrado não pode ser decifrada por ataque de ***texto simples escolhido***.

A figura 2.7 exemplifica o problema de se estabelecer um canal seguro entre dois usuários em uma rede, no exemplo, Alice e Bob, através do uso de chave pública. Neste exemplo, Alice e Bob desejam trocar mensagens secretas. Desta forma, a chave pública de Alice e o seu algoritmo de cifragem tornam-se públicos, assim como os de Bob. A notação utilizada é $\mathbf{E_A}$ para representar o algoritmo de cifragem parametrizado com a chave pública de Alice. De forma similar, utilizar-se-á a notação $\mathbf{D_A}$ para indicar o algoritmo de decifragem com a chave privada de Alice. Bob, agora, faz o mesmo, publicando $\mathbf{E_B}$, porém, mantendo secreta a chave $\mathbf{D_B}$. Agora Alice pode obter a sua primeira mensagem **P**, calcular $\mathbf{E_B(P)}$ e a enviar para Bob. Em seguida, Bob a descriptografa aplicando a sua chave secreta $\mathbf{D_B}$, ou seja, aplica $\mathbf{D_B(E_B(P)) = P}$. E ninguém mais poderá ler a mensagem $\mathbf{E_B(P)}$, pois é extremamente

difícil derivar $\mathbf{D_B}$ da chave $\mathbf{E_B}$ (requisito da Criptografia Assimétrica). Para Bob enviar uma resposta **R** para Alice, Bob transmite $\mathbf{E_A(R)}$.



Fonte: Adaptado de (STALLINGS, 1999a, p. 63)

Figura 2.7 – Criptografia Assimétrica

Contudo, se o emissor utilizar a chave privada para cifrar a mensagem, qualquer pessoa poderá decifrá-la, uma vez que a chave pública é de conhecimento de todos, o que garantirá apenas a autenticidade da mensagem.

Como exemplo de algoritmo assimétrico, pode-se citar o **RSA**, algoritmo criado em 1978 por um grupo de pesquisadores do MIT, chamados **R**ivest, **S**hamir e **A**dleman, motivo pelo qual esse algoritmo chama-se RSA, que é uma composição das iniciais dos nomes dos seus criadores. Esse algoritmo é composto por chaves de 512, 768, 1024 ou 2048 bits, e é a base da maioria das aplicações de criptografia assimétricas utilizadas atualmente, pois seus mecanismos dificultam a obtenção da chave utilizada. Para obter detalhes sobre o funcionamento deste algoritmo, bem como seus princípios teóricos, deve-se consultar (RIVEST et al., 1978).

2.7.6 Modos de Operação Cifra

Em geral, quando os algoritmos de cifragem se deparam com *plain texts* muito longos para criptografar, estes algoritmos têm que, necessariamente, quebrar esse texto em unidades menores e realizar a criptografia para cada unidade individualmente para se obter o texto criptografado completo. Porém, os algoritmos de cifragem podem utilizar diferentes técnicas, ou modos, para realizar esse processamento (cifragem), sendo os modos mais comuns os seguintes:

- *Modo Electronic Code Book (ECB)*
- *Modo de Encadeamento de Blocos de Cifras*
- *Modo de Feedback de Cifra*
- *Modo de Cifra de Fluxo*

2.7.6.1 Modo Electronic Code Book

Neste modo, o texto simples é geralmente dividido em blocos de 64 ou 128 bits, porém, o tamanho do bloco mais usual é o de 64 bits, por ser de fácil representação. Desta forma, a maneira direta de usar um algoritmo baseado no Modo *Electronic Code Book* (ECB) para codificar um longo fragmento de texto simples é dividi-lo em blocos consecutivos de 8 bytes (64 bits) e codificá-los uns após outros com a mesma chave, sendo que o último bloco de texto simples é completado com até 64 bits, se necessário.

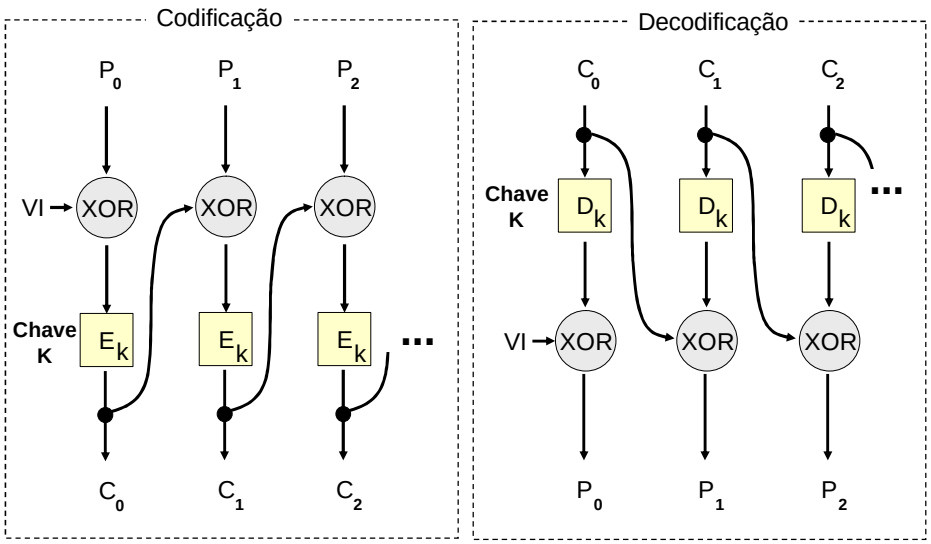
Este modo de cifra apresenta falhas de segurança, pois um invasor, com certa experiência, poderia substituir blocos cifrados, por outros previamente gravados, mesmo sem saber exatamente o conteúdo do bloco, a fim de obter alguma vantagem, e esta substituição poderia facilmente passar desapercebida, pois esta operação não afetaria o restante dos blocos cifrados, uma vez que não há relação entre os blocos no processo de cifragem.

2.7.6.2 Modo de Encadeamento de Blocos de Cifras

O Modo de Encadeamento de Blocos de Cifras é semelhante ao ECB, porém, neste modo, o bloco de texto simples é submetido a uma operação XOR com o bloco de texto

cifrado anterior, antes de ser codificado. Conseqüentemente, o mesmo bloco de texto simples não é mais mapeado para o mesmo bloco de texto cifrado, isto é, o mesmo bloco de texto simples não resultará no mesmo bloco de texto cifrado. Desta forma, a criptoanálise torna-se mais difícil e o problema de segurança no modo de cifra ECB, descrito no item anterior, não ocorre no Modo de Encadeamento de blocos de Cifras.

A figura 2.8 ilustra a codificação e decodificação no encadeamento de blocos de cifra. As caixas $\mathbf{E_K}$ e $\mathbf{D_K}$ representam os processos de criptografia e descriptografia respectivamente, usando uma chave **K**. Na codificação, os blocos $\mathbf{P_i}$ (texto simples) são submetidos a uma operação **XOR** (OU Exclusivo) com o bloco de texto cifrado $\mathbf{C_{i-1}}$ (bloco C anterior) antes de serem cifrados. Observa-se que o primeiro bloco é submetido a uma operação XOR com um vetor de inicialização – **VI**, escolhido ao acaso, transmitido (em texto simples) juntamente com o texto cifrado. A decodificação é o processo inverso.



Fonte: Adaptado de (TANENBAUM, 2003, p. 747)

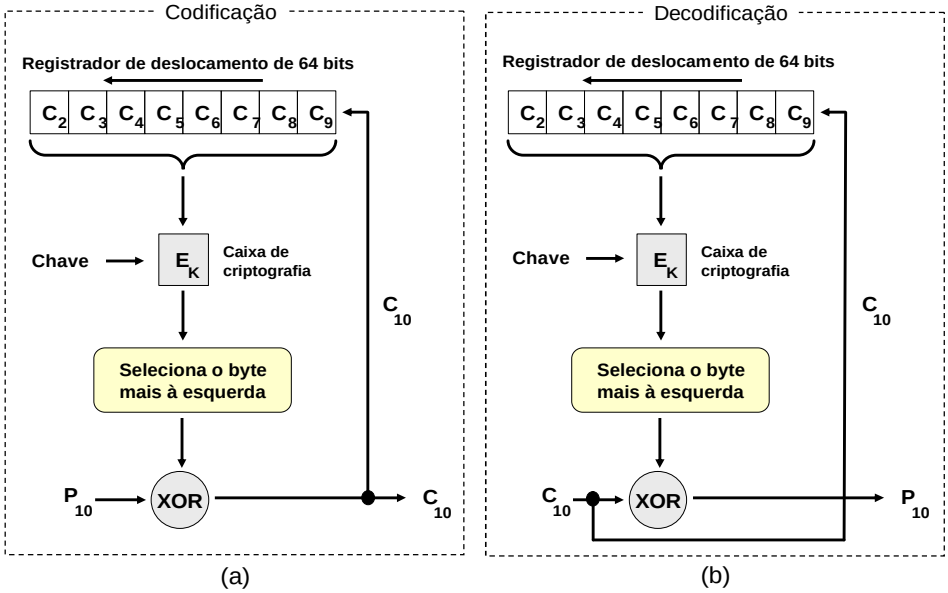
Figura 2.8 - Codificação e Decodificação no modo de Encadeamento de blocos de cifra

2.7.6.3 Modo de Feedback de Cifra

Os modos de cifras anteriores possuem a desvantagem de exigir a chegada de um bloco de 64 bits inteiro para poder iniciar a cifragem. Em ambientes interativos, nos quais os usuários podem digitar senhas, *logins*, ou linhas de comando com menos de oito caracteres e

parar a espera de uma resposta, estes modos mostram-se inadequados. Para aplicações ou ambientes que necessitem codificação *byte* a *byte*, é usado o modo de *feedback de cifra*.

Este modo é semelhante ao modo de bloco de cifra, porém, realiza a cifragem *byte* a *byte*. Neste modo, existe um registrador de deslocamento, geralmente de 64 ou 128 bits, no qual é utilizado para armazenar os últimos *bytes* cifrados. O *byte* mais à esquerda desse registrador é sempre extraído e utilizado na operação XOR com o byte corrente (texto simples). Posteriormente, este byte é encaminhado à linha de transmissão. Em cada rodada, o registrador desloca-se 8 bits à esquerda.



Fonte: Adaptado de (TANENBAUM, 2003, p. 748)

Figura 2.9 – Codificação e Decodificação no modo de feedback de cifra

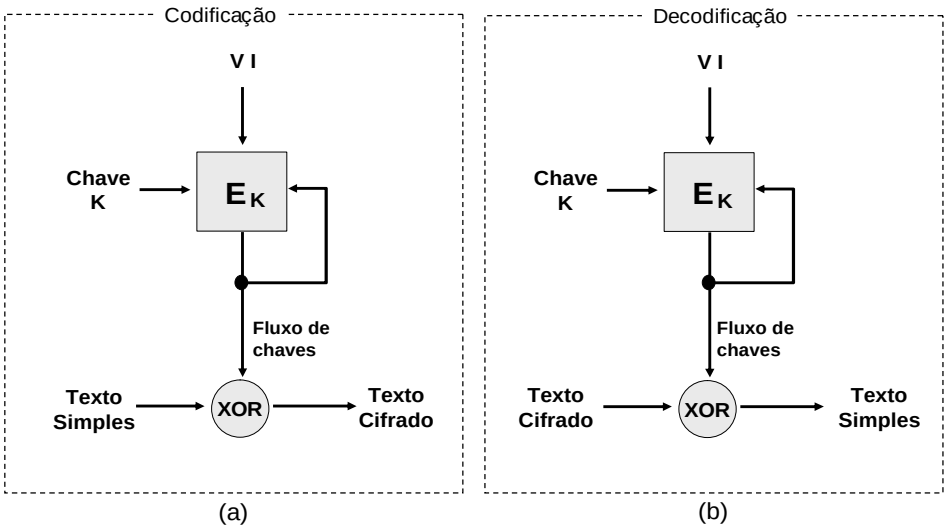
A figura 2.9(a) ilustra um registrador de 64 bits para gerar um texto cifrado de 64 bits. O byte mais à esquerda ($\mathbf{C_2}$) é extraído e submetido na operação de XOR com $\mathbf{P_{10}}$. Posteriormente, o registrador é deslocado 8 bits à esquerda, sendo descartado o $\mathbf{C_2}$, e $\mathbf{C_{10}}$ entra na posição que fica vaga, na extremidade direita do registrador. A decodificação neste modo de cifra funciona semelhante à codificação, inclusive, o conteúdo do registrador de deslocamento também é codificado e não decodificado. A figura 2.9 (b) ilustra a decodificação no modo *feedback* de cifra.

O problema deste modo de cifra é que se um bit for invertido acidentalmente durante a transmissão, os 8 bytes codificados ficarão danificados enquanto o byte defeituoso estiver no registrador, isto é, 64 bits de texto estarão danificados.

2.7.6.4 Modo de Cifra de Fluxo

Neste modo, o *texto simples* é submetido a uma operação XOR com uma seqüência de blocos chamada ***fluxo de chaves***. Este fluxo é obtido a partir de uma interação, onde inicialmente é codificado um *Vetor de Inicialização* com uma chave para se obter um bloco de saída. Este bloco gerado (a partir do VI) é então codificado usando-se a chave para se obter um próximo bloco de saída, até que se tenha uma seqüência de blocos de saída suficientemente grande para realizar a operação de XOR com todo o texto simples. Este fluxo de chaves é tratado como uma ***chave única***.

A figura 2.10 apresenta a codificação e decodificação no modo de cifra de fluxo. Na decodificação, o fluxo de chaves deve ser o mesmo da codificação.



Fonte: Adaptado de (TANENBAUM, 2003, p. 749)

Figura 2.10 – Codificação e Decodificação no modo de cifra de fluxo

A vantagem deste modo de cifra é que como o fluxo de chaves depende apenas da chave e do Vetor de Inicialização ele não é afetado por erros de transmissão. Desta forma, um erro ocasionado em um bit no fluxo do texto cifrado transmitido gera apenas o erro de um bit no texto simples.

O cuidado que deve existir nesse modo de cifra é não se utilizar o mesmo par [chave e VI] duas vezes, pois isso gerará o mesmo fluxo de chaves o tempo todo, fato que pode expor o texto cifrado a um ataque de reutilização de fluxo de chaves (TANENBAUM, 2003).

2.8 Assinaturas Digitais

Uma assinatura digital é um código binário baseado em dois aspectos: o documento em si e alguma informação que o ligue a uma certa pessoa ou conjunto de pessoas. Essa ligação é denominada autenticação. As assinaturas digitais tentam resolver o problema de se criar um substituto para as assinaturas em documentos escritas à mão.

Em um sistema no qual as entidades trocam mensagens, as Assinaturas Digitais têm como objetivos garantir (TANENBAUM, 2003):

a) que o receptor da mensagem possa verificar e certificar a identidade alegada pelo transmissor.

b) que o transmissor não possa repudiar o conteúdo da mensagem (propriedade de *não-repúdio*). Esta propriedade protege o receptor da mensagem.

c) que o receptor não possa forjar ele mesmo a mensagem. Esta característica pode proteger o transmissor de possíveis tentativas de fraudes.

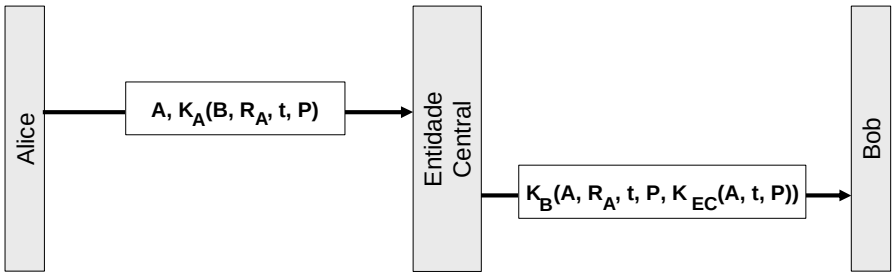
Comumente, a Assinatura Digital faz uso da criptografia assimétrica, utilizando um par de chaves, uma privada e outra pública. A chave privada serve para assinar o documento, enquanto que a pública serve para verificar a assinatura. Porém, a utilização de chaves simétricas também é utilizada.

2.8.1 Assinatura Digital de Chave Simétrica

Nas Assinaturas Digitais com o uso de criptografia simétrica, uma estratégia comum é ter uma autoridade central a qual todas as entidades confiam e que possua conhecimento de todas as chaves secretas dessas entidades. Neste cenário, somente a entidade comunicante (transmissor ou receptor da mensagem) e a entidade centralizadora conhecem a chave secreta.

A figura 2.11 apresenta um esquema onde Alice envia uma mensagem para Bob através de uma entidade centralizadora, fazendo uma assinatura de sua mensagem usando sua chave secreta K_A, gerando a mensagem criptografada K_A (B, R_A, t, P), onde **B** é a identidade de Bob, $\mathbf{R_A}$ é um número aleatório gerado por Alice, ***t*** é um timbre de hora para evitar ataques de repetição e **P** é a mensagem enviada por Alice. A entidade central (aquela em que todos

confiam) descriptografa a mensagem de Alice, verifica o destinatário (no caso é Bob) e envia a Bob a mensagem criptografada com a sua chave privada: K_B (A, R_A, t, P, K_{EC} (A, t, P)), onde **A** é a identidade de Alice, assinalando que Alice é quem enviou a mensagem. Notemos que a entidade centralizadora também envia K_{EC} (A, t, P), onde $\mathbf{K_{EC}}$ é a chave de criptografia da Entidade Centralizadora, com o objetivo de impedir que Alice possa repudiar a mensagem posteriormente, de maneira que Bob possa, se necessário, provar que Alice realmente enviou a mensagem P com atributo ***t***.



Fonte: (TANENBAUM, 2003, p. 757)

Figura 2.11 - Assinatura Digital de Chave Simétrica usando uma Entidade Central

2.8.2 Assinatura Digital de Chave Pública

O grande problema com o uso de criptografia simétrica para Assinaturas Digitais é que as entidades comunicantes precisam confiar plenamente em uma autoridade central que pode, inclusive, ler todas as mensagens trafegadas por ela. Este contexto, dentro de um mesmo ambiente organizacional, possivelmente, não causaria maiores transtornos, porém, a participação de entidades de outras organizações dentro desse processo de segurança não inspira confiança.

Desta forma, o uso de um mecanismo que não exigisse a presença de uma autoridade central para a utilização de Assinaturas Digitais é imprescindível. Felizmente, o uso de criptografia de chave pública para Assinaturas Digitais é possível e pode contribuir com a solução desta problemática.

A figura 2.12 apresenta o mesmo cenário descrito na seção anterior, porém, agora, utilizando criptografia assimétrica (chave pública). Neste, Alice pode enviar $\mathbf{E_B(D_A(P))}$ para Bob, a qual criptografou **P** com a sua chave privada, fazendo $\mathbf{D_A(P)}$ e depois criptografou

$D_A(P)$ usando a chave pública de Bob, fazendo **$E_B(D_A(P))$**. E este, quando recebe **$E_B(D_A(P))$,** descriptografa a mensagem usando a sua chave privada, produzindo **$D_A(P)$**, aplicando posteriormente a chave pública de Alice **E_A** para gerar **P**.

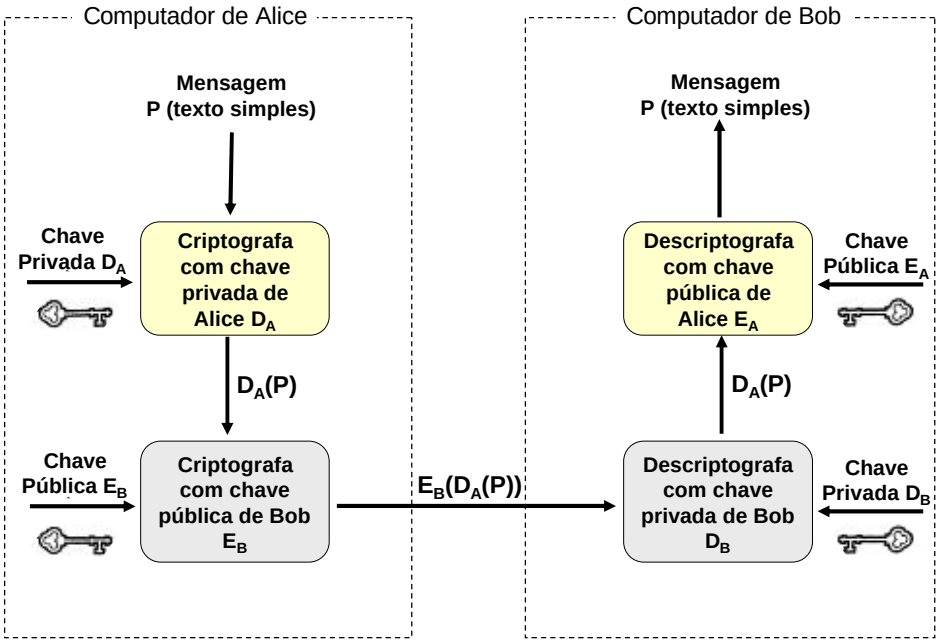


Figura 2.12 - Assinatura Digital com uso de Criptografia de Chave Pública

O problema que ocorre com a Assinatura Digital com o uso de Chave Pública é que a característica de ***não-repúdio*** só ocorre se houver garantias de que as chaves privadas das entidades comunicantes permanecem secretas e estas também não foram alteradas. Por exemplo, Bob só poderá provar que uma mensagem foi enviada por Alice enquanto a chave privada de Alice, chave D_A, permanecer secreta. Quando Alice revelar sua chave secreta, ela pode afirmar que qualquer entidade poderia ter enviado a mensagem.

2.8.3 Sumários de Mensagens (Funções de Hash)

Em geral, os métodos de assinaturas digitais reúnem duas funções distintas: *autenticação* e *sigilo*, sendo esta última função opcional. Normalmente os documentos são enviados em *texto simples* (descriptografado), porém, com assinatura digital. Obviamente, isto dependerá muito dos requerimentos de segurança a serem aplicados às informações e documentos.

O método *Message Digest*, ou Sumário de Mensagem, baseia-se em uma função de *Hash* unidirecional que extrai um texto qualquer do texto simples de uma mensagem original e a partir dele calcula uma *string* de bits de tamanho fixo, chamada de resumo.

Dada uma mensagem original, a função *hash* tem como objetivo produzir uma cadeia de bits, conhecida como ***sumário de mensagem***, ou **resumo**, que representa de forma única esta mensagem. Uma propriedade desta função diz que o caminho inverso deverá ser computacionalmente inviável, ou seja, não poderá ser possível obter uma mensagem original através de um resumo, o que garante a integridade da mesma.

Os algoritmos que implementam a função *hash* têm como objetivo fazer com que o resumo sofra uma grande modificação caso algum caractere do conteúdo da mensagem seja alterado. Dentre os principais, podemos destacar o *Message Digest 5* (MD-5) e o *Secure Hash Algorithm 1* (SHA-1). Ambos processam os dados de entrada em blocos de 512 bits. O MD5 retorna sumários de 128 bits, enquanto o SHA-1 gera sumários de 160 bits. O MD5, por gerar um resumo menor é mais rápido, porém, o sumário gerado pelo SHA-1 é mais seguro por ter 160 bits. Atualmente, já existe uma coleção de algoritmos de sumário denominada *Secure Hash Algorithm* 2 (SHA-2), padronizado pelo *National Institute of Standards and Technology* (NIST), que definiu o SHA-256, o SHA-384 e o SHA-512, tendo como principal característica o retorno de um sumário de 256, 384 ou 512 bits respectivamente. Portanto, mais seguro contra ataques de força bruta que seus antecessores.

É conveniente mencionar que para toda função *Hash* representada por MD, podemos inferir que tal método de Sumário de Mensagem sempre possuirá as seguintes propriedades por definição:

- Se um texto **P** for fornecido, o cálculo de MD(**P**) deverá ser extremamente fácil.

- Dado um texto **P** desconhecido, se MD(**P**) for fornecido, será efetivamente impossível encontrar **P**, ou seja, o caminho inverso deverá ser computacionalmente inviável, não poderá ser possível obter uma mensagem original através de um resumo, o que garante a integridade da mesma.

- Dado um texto **P** qualquer, ninguém pode encontrar um texto **P''** tal que MD(**P''**) = MD (**P**). Para atender a este requisito, a função MD deverá ter pelo menos 128 bits, de preferência mais (TANENBAUM, 2003).

- Caso algum caractere do conteúdo da mensagem seja alterado, o resumo deve sofrer uma grande modificação, isto é, a função *hash* deverá produzir uma saída completamente diferente, caso 1 bit seja alterado.

Cabe ressaltar, que o processo de geração da assinatura utiliza a função *hash* para a obtenção do resumo do documento, e que, em seguida, cifra-o com a chave privada do emissor e envia-o ao receptor. Este utilizará a chave pública do emissor para decifrar a mensagem e a função hash para recalcular o resumo do documento, comparando-o com o resumo recebido. Isto garante a integridade e autenticidade do documento.

A figura 2.13 apresenta a utilização de uma função de sumário de mensagem utilizada para enviar uma mensagem não-secreta e seu resumo assinado de Alice para Bob.

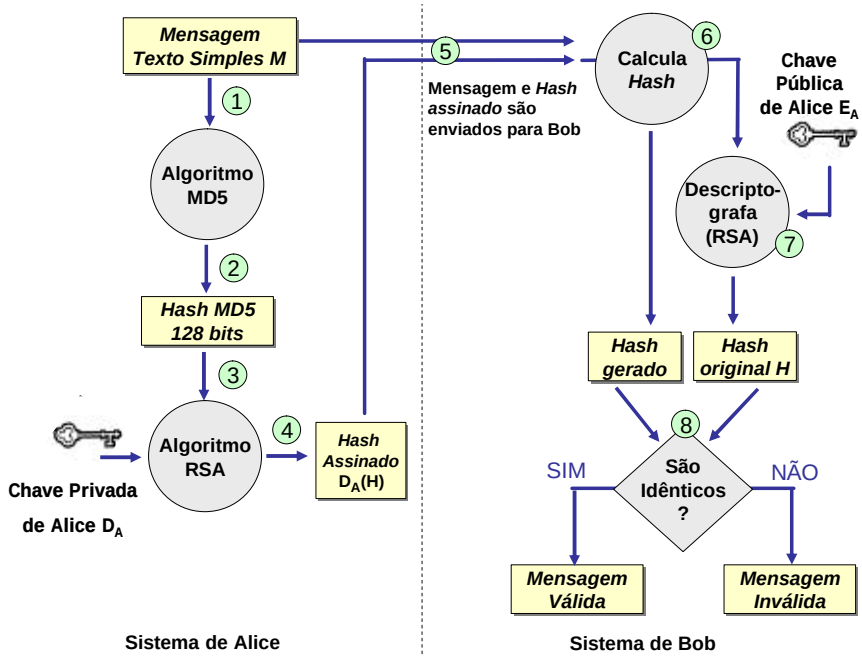


Figura 2.13 - Envio de um sumário de mensagem e do RSA para assinar mensagens não-secretas.

Neste cenário, a mensagem de texto simples é submetida ao algoritmo MD5 para se obter um *hash* de 128 bits do MD5. Na seqüência, após a geração do resumo MD5, Alice assina o *hash* com sua chave privada RSA, enviando o texto simples e o *hash* assinado para

Bob. Posteriormente, Bob calcula o *hash* MD5 da mensagem recebida, descriptografa o *hash* assinado, com a chave pública RSA de Alice, e o compara ao *hash* recebido, H. Se os dois forem idênticos, a mensagem é considerada válida.

Maiores detalhes sobre o MD5 e o SHA-1 podem ser consultados em (RIVEST, 1992) e (NIST, 1993) respectivamente. Para uma análise mais geral, consulte (TANENBAUM, 2003).

2.9 Segurança AAA

A segurança de acesso à rede é baseada em uma arquitetura modular denominada ***Arquitetura AAA*** por possuir três componentes básicos, a saber:

- ***Authentication*** (***Autenticação***) – requer que os usuários provem que são realmente quem dizem ser. As técnicas mais comuns de autenticação utilizam informações sigilosas, como senhas, senhas de único uso (*one-time password* - S/KEY), pergunta/resposta, meios físicos (como cartões magnéticos), verificação de informações biométricas (como impressões digitais) ou através da combinação delas. Um exemplo desta combinação está na utilização de um cartão de crédito, onde se utiliza, além do cartão, uma senha para efetuar a transação.

- ***Authorization*** (***Autorização***) – após a autenticação do usuário, os serviços de autorização decidem quais recursos os usuários podem acessar e quais operações podem realizar. Exemplo: “O usuário *professor* pode acessar o *host LX_Server* através de SSH”.

- ***Accounting*** (***Contabilidade***) – registra o que o usuário realmente fez, o que ele acessou e por quanto tempo, para fins de contabilidade e auditoria, mantendo um registro de como os recursos de rede são utilizados. A contabilidade pode ser utilizada também para controlar acesso à rede e para detectar intrusões. Exemplo: “O usuário *aluno* tentou acessar o *host LX_Server* por meio de Telnet 10 vezes”.

Comumente, muitas referências bibliográficas sobre este assunto referenciam-se aos métodos AAA como sendo métodos de Autenticação, isto porque, praticamente todos os métodos AAA possuem suporte a autenticação, porém, nem sempre um método de autenticação possui métodos de Autorização ou Contabilidade. Seguindo este modelo, discorreremos sobre alguns métodos de autenticação importantes para um melhor entendimento dos processos de autenticação que podem existir em uma infra-estrutura de VPNs.

2.10 Protocolos de Autenticação

Um dos requisitos fundamentais para o estabelecimento de uma conexão VPN é que exista um processo de Autenticação entre suas entidades participantes. Este processo pode garantir que um parceiro na comunicação é quem realmente deve ser e não um intruso ou impostor. Apesar de parecer simples, a técnica de autenticação é difícil e exige protocolos bem complexos e uso de criptografia (TANENBAUM, 2003).

Sem a autenticação, não se pode garantir o sucesso de uma VPN, pois não existiriam garantias de que as identidades dos participantes em uma conexão VPN são de fato verdadeiras. Desta forma, torna-se fundamental um conhecimento mais detalhado dos principais protocolos de autenticação utilizados em redes VPN.

Em redes VPN, a informação de autenticação poderá estar sob o controle de duas ou três entidades. Quando o esquema de autenticação está sob o controle de duas entidades, a entidade que está se autenticando e a entidade autenticadora, o esquema é chamado *Two-Party Authentication*. Se for utilizada uma terceira entidade, que geralmente possui o papel de validar ou certificar a autenticidade das outras entidades, este esquema é chamado de *Trusted Third-Party Authentication* (SILVA, 2003).

A autenticação *Two-Party* ainda se subdivide em dois esquemas: o de uma via (*one-way*) e o de duas vias (*two-ways*). No primeiro esquema, uma entidade, geralmente cliente, se autentica em um servidor, sem que este precise se autenticar no cliente. No esquema de duas vias, todas as entidades devem se autenticar mutuamente entre si. Em ambas situações, a informação, ou parte da mesma, que é compartilhada entre as duas entidades participantes da comunicação, é chamada *Shared Secret*, ou segredo compartilhado. Os principais métodos de

autenticação *Two-Party* conhecidos são: *Shared Secret* (Chave Secreta Compartilhada) e *Challenge/Response*. O *Shared Secret*, pela sua simplicidade, é um dos esquemas de autenticação mais utilizados em redes VPN, ele utiliza uma senha compartilhada entre as entidades das redes. Neste esquema as entidades conhecem previamente a senha ou chave secreta. Enquanto o *Challenge/Response* é um esquema de desafio e resposta, onde o servidor lança um desafio a uma outra entidade, esperando uma resposta, previamente acordada entre eles. Geralmente, este esquema utiliza como desafio e resposta um conjunto de chaves para criptografia simétrica. Os principais protocolos de autenticação que sustentam estes métodos *Two-Party* são: PAP (*Password Authentication Protocol*), o CHAP (*Challenge Handshake Authentication Protocol*), o EAP (Este*nsible Authentication Protocol*), o TACACS+ e o RADIUS. Estes protocolos serão descritos posteriormente.

Como mencionado, a autenticação *Trusted Third-Party* utiliza uma terceira entidade que proverá um conjunto de credenciais ou informações de autenticação das outras entidades envolvidas na comunicação. Este método é particularmente mais adequado à construção de redes VPNs mais robustas e com maior número de usuários estabelecendo redes virtuais. Este esquema permitirá uma maior centralização e controle das informações de autenticação. Os principais métodos de autenticação *Third-Party* conhecidos são: *Kerberos* e a Infra-estrutura de Chave Pública X.509 (PKI), as quais serão detalhadas posteriormente. Este trabalho estudou a aplicação da Infra-estrutura de Chave Pública PKI para a construção de redes VPN.

O Processo de autenticação é geralmente feito no início do estabelecimento de uma seção, podendo ser refeito sistematicamente durante uma conexão, para reduzir as chances de ataques no meio da comunicação. A autenticação também pode ser utilizada para garantir a integridade de pacotes adicionando assinaturas nas mensagens.

2.10.1 Métodos de Autenticação Two-Party

A seguir, iremos discutir alguns métodos de autenticação *two-party*, isto é, aqueles que são baseados entre duas entidades. Na sua maioria, são métodos que utilizam o protocolo PPP para o estabelecimento de conexão. Obviamente, não é objeto deste trabalho descrever detalhadamente todos os métodos, até porque alguns deles são protocolos utilizados para conexão de clientes remotos via linha discada e o escopo deste trabalho restringiu-se a

conexão entre redes corporativas. Porém, é importante sua descrição para um melhor entendimento dos mecanismos de autenticação utilizados em redes VPN.

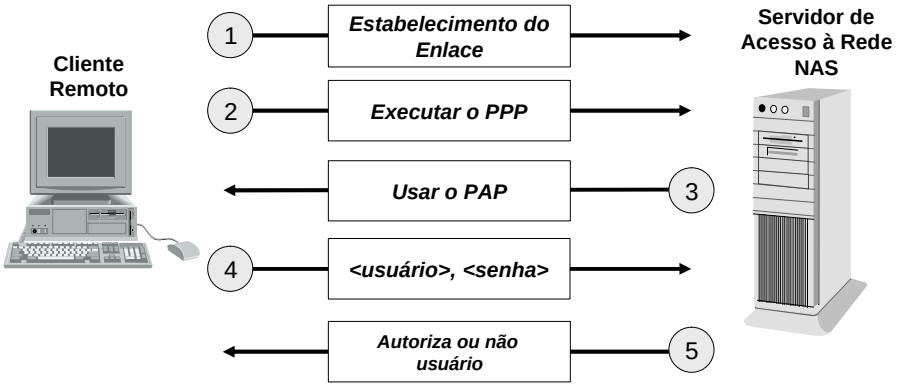
2.10.1.1 Password Authentication Protocol (PAP)

Este foi um dos primeiros protocolos desenvolvidos para realizar autenticação. É um método que utiliza o protocolo PPP (protocolo ponto-a-ponto), geralmente adotado em conexões discadas, para realizar autenticação de cliente. Neste esquema, a senha é enviada a um NAS (*Network Authentication Service* - Servidor de Autenticação da Rede) em forma de *String* (texto simples ou *plain text*), que posteriormente passa por uma validação de informações. Se as informações estiverem corretas, o acesso à rede é liberado, caso contrário, seu acesso é desconectado.

A seguir, encontram-se as mensagens trocadas durante a autenticação PAP (WENSTROM, 2002):

1. O cliente remoto estabelece o enlace discado.
2. O cliente remoto informa ao Servidor de Acesso à Rede que ele está executando o PPP.
3. O Servidor de Acesso à Rede, configurado para usar o PAP, notifica o cliente remoto para usar o PAP na sessão.
4. O cliente remoto envia o nome do usuário e a senha no formato PAP.
5. O Servidor de Acesso à Rede compara o nome do usuário e a senha com os que estão armazenados no banco de dados e aceita ou rejeita o nome do usuário e senha digitados.

A figura 2.14 ilustra este cenário de troca de mensagens no protocolo PAP, de acordo com os passos acima descritos.



Fonte: Adaptado de (WENSTROM, 2002)

Figura 2.14 - Etapas de autenticação do PAP sobre o PPP

Suas principais características são:

- Atua no nível de enlace.
- Não possui criptografia no envio no nome do usuário e da senha para o NAS.
- A autenticação é feita somente no início da conexão.
- Não possui controle sobre o número de tentativas de conexão.
- Não oferece nenhuma proteção contra ataques de reprodução ou tentativas de erros repetidos.

Diante dessas limitações, fica evidente que este protocolo não se adéqua aos requerimentos atuais de segurança para o estabelecimento de *links* confiáveis para a configuração de redes VPN, principalmente pela ausência de criptografia. Desta forma, não será foco deste trabalho o seu maior detalhamento.

2.10.1.2 Challenge Handshake Authentication Protocol (CHAP)

O CHAP (*Challenge Handshake Authentication Protocol*), descrito na RFC 1994 (SIMPSON, 1996), também é um método utilizado pelo protocolo ponto-a-ponto (PPP). Porém, é um método de autenticação mais complexo que o PAP, porque a senha real do

usuário não atravessa o canal de comunicação. Ele é um protocolo de autenticação bastante utilizado nos ambientes Linux atualmente. Em algumas distribuições Linux, este protocolo já está habilitado como padrão (ORTIZ, 2003). Para plataforma Windows existe o MS-CHAP. Entretanto, não é compatível com o CHAP original.

O método de autenticação deste protocolo ocorre em três fases (*handshake* de três vias). O *handshake de três vias* ocorre após as etapas de estabelecimento do *link*, são elas:

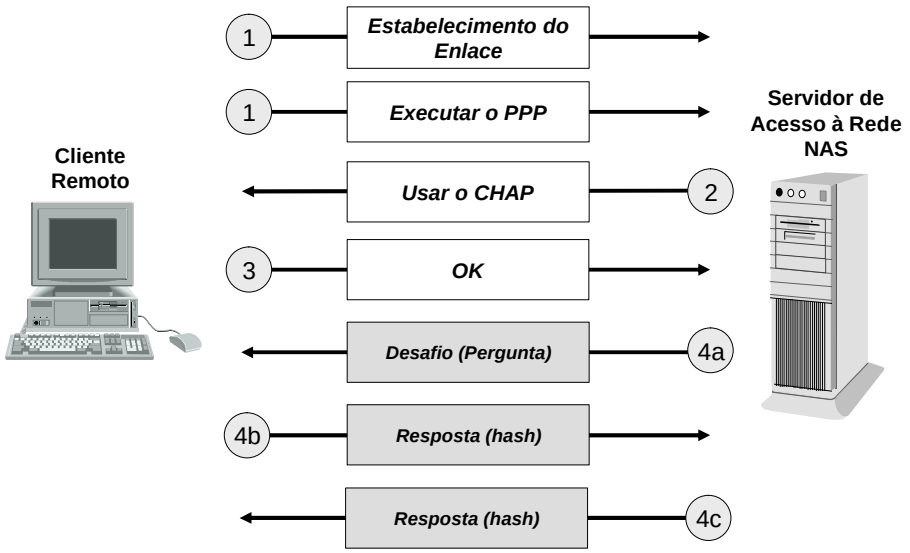
1. O enlace PPP é estabelecido após a discagem. O Servidor de Acesso à Rede (NAS) é configurado para suportar PPP e CHAP.

2. O Servidor de Acesso à Rede, configurado para usar o CHAP, notifica o cliente remoto para usar o CHAP na sessão.

3. O cliente remoto responde OK.

4. O *handshake* de três vias ocorre.

A figura 2.15 ilustra as etapas de autenticação do protocolo CHAP sobre o PPP.



Fonte: Adaptado de (WENSTROM, 2002)

Figura 2.15 – Fases do protocolo de Autenticação por Desafio (CHAP)

A seguir, explicaremos como o *handshake* de três vias ocorre (etapa 4).

A primeira fase do *handshake* (4a), que ocorre após o estabelecimento do *link*, sempre iniciando do NAS, é o envio de um desafio por parte do Autenticador (terminologia usada pela RFC 1994 para denominar o Servidor de Autenticação) para o cliente, selecionando de forma aleatória dentro de um conjunto já pré-estabelecido de desafios e respostas.

Na segunda fase do *handshake* (4b), o cliente utiliza uma função *hash* (utiliza normalmente o MD5) calculado em cima do desafio proposto, para posteriormente devolver a resposta ao servidor.

Na terceira fase do *handshake* (4c), o servidor irá validar a resposta (enviada pelo cliente), verificando-a contra seu próprio cálculo do valor de *hash* esperado. Ele decifra a mensagem enviada através da senha do usuário contida no seu banco, autorizando-o ou não.

É importante ressaltar que, em intervalos randômicos, o Autenticador (Servidor) pode enviar um novo desafio ao cliente, repetindo assim todos as fases mencionadas na etapa 4. E ao contrário do PAP, antes de enviar uma solicitação de conexão ao NAS, ele já faz a criptografia dos dados, utilizando o MD5 (*Message Digest* 5).

Em resumo, o protocolo CHAP possui as seguintes características:

- Depende de um “segredo” conhecido apenas pelo Servidor e pelo Cliente, porém, é importante que se saliente que este “segredo” não é enviado através do *link*.

- É um método “*one-way*”, isto é, apenas um lado da comunicação se autentica. Porém, pode ser facilmente adaptado para uma autenticação mútua utilizando o mesmo “segredo” (SIMPSON, 1996).

- A autenticação pode ser repetida durante a conexão com o envio de diferentes desafios, escolhidos aleatoriamente, o que pode ser interessante para se evitar ataques de *replay*, onde o invasor pode tentar se autenticar utilizando uma resposta de um desafio capturado anteriormente (SILVA, 2003).

- As senhas dos usuários não são armazenadas criptografadas no servidor, pois o servidor necessita da senha em texto simples para descriptografar o *hash*

recebido do cliente. Esta característica representa um ponto bastante falho deste protocolo, pois deixa uma porta de entrada para possíveis invasões.

- Os dados enviados entre cliente e servidor são criptografados.

- Possui a finalidade apenas de permitir ou negar o acesso à rede, ele não permite definir níveis de permissões para determinados usuários. Essa característica representa uma desvantagem para sistemas mais específicos, onde exista a necessidade de se implementar níveis de segurança diferenciados.

2.10.1.3 Extensible Authentication Protocol (EAP)

O EAP (*Extensible Authentication Protocol*), descrito na RFC 2284 (BLUNK; VOLLBRECHT, 1998) é um protocolo geral para autenticação PPP que suporta múltiplos mecanismos de autenticação. Ele funciona como um mecanismo de negociação de protocolos de autenticação. Os protocolos PAP e CHAP, vistos anteriormente, são mecanismos de autenticação simples, escolhidos na fase de *link* (protocolo LCP). No caso do EAP, ele não seleciona um mecanismo específico de autenticação na fase de Controle de Link (LCP), ele posterga a escolha até a fase de autenticação, o que permite que o servidor possa solicitar uma quantidade maior de informações ao cliente antes de determinar o mecanismo de autenticação específico (BLUNK; VOLLBRECHT, 1998).

A RFC 2284 define várias formas de autenticação para o EAP, entre elas, ressaltam-se as seguintes:

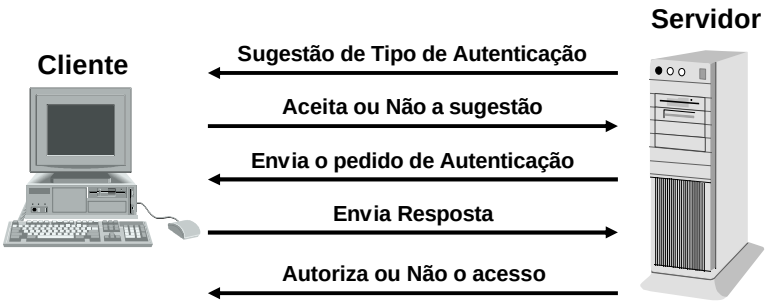
- *MD5 Challenge* – é análogo ao protocolo PPP CHAP, porém, especifica que os desafios e as respostas são construídos através de funções *hash* MD5.

- *One-Time Password* – quando a senha é gerada uma única vez para cada seção.

- *Generic Token Card* – neste tipo é gerado uma combinação numérica aleatória para cada seção.

A figura 2.16 apresenta um esquema de negociação e autenticação em um ambiente que utiliza EAP após a fase de controle de *link*. Nesta, o servidor envia uma mensagem

contendo uma sugestão de método de autenticação, podendo o cliente aceitar ou não a sugestão realizada pelo servidor. Caso o cliente aceite a proposta de autenticação, o servidor processa o pedido de autenticação dependendo do tipo escolhido.



Fonte: Adaptado de (SILVA, 2003, p.100)

Figura 2.16 – Protocolo de Autenticação EAP

Algumas observações quanto ao EAP são pertinentes. A primeira, é que a autenticação pode ser feita *one-way* ou *two-ways*, porém, não existe obrigação do mesmo método de autenticação ser aplicado em ambas as direções, sendo perfeitamente aceitável o uso de diferentes protocolos em cada direção (BLUNK; VOLLBRECHT, 1998). A segunda, é que o EAP é vulnerável a ataques, pois, como é um mecanismo que permite vários métodos de autenticação, sendo bastante flexível, permite que os clientes possam escolher os métodos mais fracos de autenticação entre os disponíveis, característica que pode ser explorada por usuários mal-intencionados na rede. De acordo com (SILVA, 2003), este é o motivo pelo qual cada vez menos este protocolo é utilizado como padrão de autenticação entre duas entidades.

2.10.1.4 RADIUS e TACACS+

Diversos padrões de bancos de dados de segurança foram criados para fornecer controle de acesso uniforme para equipamentos e usuários de rede, entre os principais produtos desenvolvidos para este fim, destacam-se: RADIUS, TACACS+ e o Kerberos[4]. Iremos começar discorrendo sobre o RADIUS.

O RADIUS, ou *Remote Authentication Dial-In User Service*, foi desenvolvido inicialmente pela *Lucent Technologies*, como uma solução para prover autenticação e gerenciamento de Clientes Remotos conectados através de linhas discadas. Em janeiro de

[4] Os Equipamentos da CISCO suportam todos os padrões de banco de dados de segurança citados.

1997, o RADIUS foi padronizado pela IEFT através da RFC 2058. Posteriormente, o protocolo RADIUS teve dezenas de outras RFCs que o alteraram ao longo de sua existência, tornando a sua definição inicial totalmente obsoleta. Atualmente, ele é descrito pela RFC 2865, de junho de 2000 (RIGNEY et al., 2000) e pela RFC 2866 (RIGNEY, 2000) que define o serviço de contabilidade do RADIUS. Possui também algumas RFCs que definem extensões e atualizações ao modelo.

A solução RADIUS baseia-se em uma arquitetura cliente-servidor, a qual adiciona um elemento de rede chamado *Network Access Server* (NAS), ou Servidor de Acesso Remoto, que possui como principais funções: o estabelecimento da conexão remota via linha discada, o gerenciamento dos pedidos de conexão e a liberação ou não dos pedidos. Em vários ambientes, o NAS está pré-configurado para realizar todo o processo de autenticação (desde que seja implementada tal função pelo fornecedor do *hardware* ou *software*), pois pode conter o próprio banco de dados de usuários e senhas, porém, apesar dessa facilidade, esta configuração é um tanto incomum, pois nas corporações, em geral, existem servidores específicos de autenticação que centralizam todos os serviços de autenticação, não somente a autenticação dos clientes via linha discada, mas também dos clientes internos da Intranet. Assim, é mais viável que se utilize um único servidor para autenticação.

Desta forma, o NAS pode, e deve, ser configurado para trabalhar como cliente de um Servidor de Autenticação, no caso, um servidor RADIUS. Com isso, cada pedido de autenticação passa pelo NAS, que passa ao servidor RADIUS, que recebe um retorno do servidor e o encaminha ao cliente, conforme pode ser observado na figura 2.17.

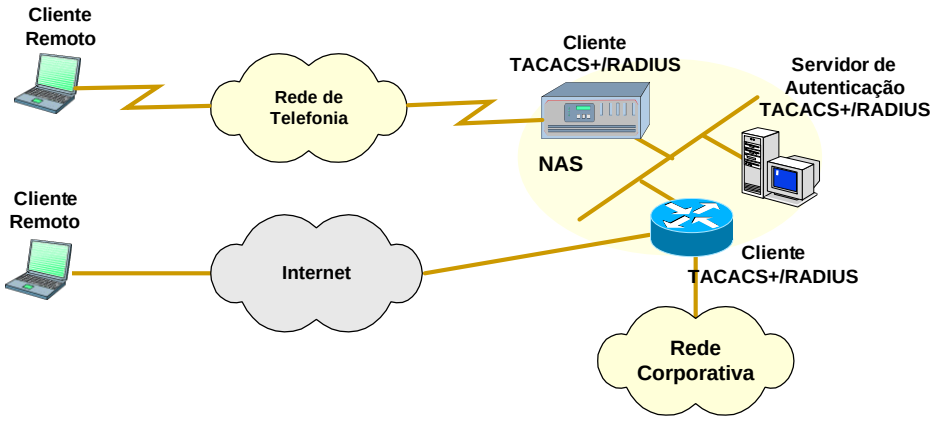


Figura 2.17 – TACACS+ ou RADIUS suportado no NAS, roteador e BD de Segurança

O RADIUS está implementado em várias plataformas, inclusive no Linux. O *Cistron RADIUS Serve*r e o *Merit AAA RADIUS Server* são exemplos de produtos que implementam um Servidor RADIUS em plataforma Linux.

(WENSTROM, 2002) relaciona os recursos de Servidor de Segurança suportados por uma solução RADIUS, os principais são:

- O RADIUS fornece suporte AAA para usuários remotos.

- O RADIUS utiliza o protocolo de transporte UDP (porta 1812 ou 1645) para comunicação entre o NAS e o Servidor de Segurança, simplificando assim, as implementações cliente e servidor do RADIUS.

- O RADIUS possui serviços de autenticação e autorização combinados, possibilitando que o cliente seja autenticado e que receba informações de configuração do Servidor. O serviço de contabilidade é realizado separadamente.

- As senhas de usuários são criptografadas, usando-se o *hash* MD5 para segurança.

- O RADIUS possui suporte a autenticação pergunta/resposta PAP e CHAP.

- Todas as transações entre cliente e o servidor de segurança RADIUS são autenticadas através do emprego de um segredo compartilhado.

O cliente RADIUS e o servidor de segurança RADIUS comunicam-se utilizando os seguintes pacotes: *Access-Request* (solicitação de acesso), *Access-Accept* (acesso aceito), *Access-Reject* (Acesso Rejeitado) e *Access-Challenge* (pergunta de acesso). A seguir descrevemos os principais passos ocorridos quando um cliente tenta realizar login e autenticar-se em um NAS.

As etapas são as seguintes:

1. O usuário solicita autenticação PPP para o NAS e entra com usuário e senha.

2. O NAS então envia um pacote Access-Request contendo o nome do usuário e a sua senha criptografada, além de outros atributos, para o Servidor RADIUS.

3. O Servidor RADIUS valida e autentica o cliente, pesquisa os parâmetros de autorização do usuário e envia o pacote *Acess-Accept* para o cliente, caso o usuário seja autenticado, ou envia o pacote *Access-Reject*, caso o usuário não seja autenticado. O Servidor pode opcionalmente ainda enviar um pacote de desafio *Access-Challenge* para o NAS.

4. A resposta *Access-Accept* ou *Access-Reject* é empacotada com dados adicionais (próprios de cada fabricante) que são usados para a autorização.

5. O Servidor RADIUS pode periodicamente enviar um pacote *Access-Challenge* para o NAS, a fim de solicitar que o usuário digite seu nome e senha novamente, enviar o estado do NAS ou executar outras ações definidas pelo fornecedor RADIUS.

O TACACS+ é um aplicativo de AAA para servidor de segurança e um protocolo que permite o controle central de usuários que tentam obter acesso através de um NAS, roteador ou outro equipamento de rede que suporte TACACS+. A sua especificação de protocolo é de padrão de indústria, inicialmente descrita através da RFC 1492. As suas funções são muito semelhantes às funções do RADIUS, porém, eles apresentam algumas diferenças. A tabela 1 demonstra as principais diferenças existentes entre o RADIUS e o TACACS+.

Tabela 1 – Comparação entre TATACS+ e RADIUS

Fonte: (WENSTROM, 2002)

Funcionalidade	TACACS+	RADIUS
Suporte AAA	Separa os três serviços AAA	Combina autenticação e autorização e separa a contabilidade
Protocolo de Transporte	TCP	UDP
Pergunta/Resposta	Bidirecional	Unidirecional (somente do Servidor RADIUS para o Cliente)
Integridade de Dados	Todo o pacote TACACS+ é criptografado	Somente a senha do usuário é criptografado

2.10.2 Métodos de Autenticação Third-Party

Como anteriormente descrito, todos os métodos de autenticação estudados até o momento são baseados na comunicação entre duas entidades, normalmente cliente e servidor, ou dois servidores, que compartilham algum tipo de informação de autenticação, como senha, por exemplo. Agora, se ampliarmos o escopo de atuação dos serviços de autenticação para possibilitar a conexão entre diferentes redes, de diversas filiais ou parceiros de negócio, podemos perceber que a complexidade e o custo na administração e manutenção das informações de autenticação passam a ser altos, principalmente para a construção de redes VPNs, em que se pressupõem vários usuários estabelecendo redes virtuais (SILVA, 2003).

Desta forma, torna-se essencial a separação dessas tarefas e delegá-las a uma terceira entidade, confiável a toda a rede, que irá centralizar todos os processos de autenticação. E conforme já mencionado, este esquema permitirá uma maior centralização e controle das informações de autenticação, o que é bom para a gestão da segurança. Os principais métodos de autenticação *Third-Party* conhecidos são: Kerberos e a Infra-estrutura de Chave Pública X.509 (PKI). O Kerberos será discutido a seguir. A infra-estrutura de Chave Pública X.509 será descrita em uma seção específica deste trabalho, devido a sua grande relevância para a construção de redes VPN seguras.

2.10.2.1 Kerberos

Kerberos é um protocolo de autenticação desenvolvido pelo *Massachussets Institute of Technology* – MIT. Foi desenvolvido para prover um forte método de autenticação entre aplicações cliente/servidor através do uso de criptografia de chave secreta. Ele utiliza o DES-3 como algoritmo de criptografia. No Kerberos, o cliente pode provar a sua identidade ao servidor, e vice-versa, através de uma conexão insegura. Atualmente, este protocolo está disponível em várias plataformas, inclusive em produtos comerciais (MIT, 2003).

O Kerberos depende de uma terceira entidade validadora, chamada KDC (*Key Distribution Center*), ou Centro de Distribuição de Chaves, para a verificação segura de usuários e serviços. Ele mantém um banco de dados de seus usuários no KDC. O uso principal do Kerberos é para garantir que usuários e os serviços de rede usados sejam realmente quem e o que afirmam ser. Para realizar essa verificação, o KDC emite bilhetes para os usuários. Estes bilhetes são temporizados (possuem uma limitação de tempo) e

armazenados em um *cache* de credencial, podendo ser utilizados em substituição ao mecanismo padrão de autenticação de nome e senha do usuário.

O Kerberos pode ser usado para autenticar sessões PPP, logins em roteadores, logins em Servidores de Acesso à Rede (NAS) e acesso a serviços de Telnet, FTP, entre outros serviços de redes.

As soluções Kerberos consistem de diversos componentes de *software* e de *hardware*. Seus principais componentes são: o **KDC** que contém o banco de dados do Kerberos, contendo as configurações dos usuários, o *software* **Servidor Kerberos**, os *softwares* **clientes Kerberos** que podem ser inclusive implementados em roteadores e **Utilitários Kerberos** que ativam recursos remotos do cliente. A figura 2.18 ilustra uma topologia de rede que apresenta uma infra-estrutura Kerberos com seus principais componentes. Neste cenário, os aplicativos e serviços existentes na rede, que foram modificados para suportar a infra-estrutura de credenciais do Kerberos, são ditos ***componentes kerberizados***.

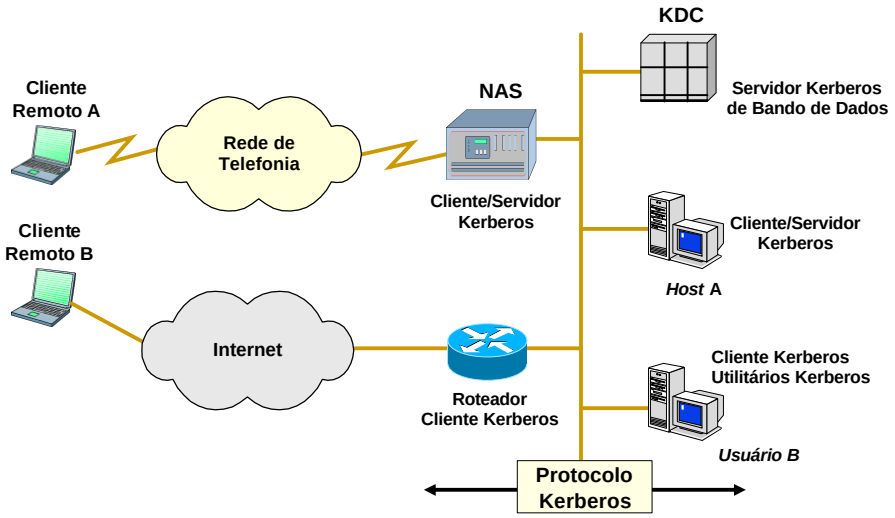


Figura 2.18 – Infra-estrutura kerberizada

Para exemplificar o funcionamento do Kerberos, imaginemos que um usuário B queira acessar um servidor de *Telnet*, instalado no *host* **A**. Os passos a seguir descrevem o processo de autenticação:

1. O usuário **B** realiza o *logon* e é autenticado no KDC utilizando algum programa cliente kerberizado.

2. O KDC fornece uma credencial criptografada para o usuário **B**. E este autentica a Credencial recebida.

3. O usuário **B** tenta acessar o Telnet no *host* **A**, apresentando a credencial recebida no *logon* ao KDC e solicita uma outra credencial de serviço que autorize o acesso ao *Telnet* ao *host* **A**.

4. O KDC fornece uma credencial que autoriza o acesso *Telnet* ao *host* **A**.

5. O sistema do usuário **B** fornece a credencial de serviço ao *host* **A** e obtém acesso por *Telnet*.

6. O sistema do usuário **B** apresenta a credencial de serviço para acesso subseqüente a outros serviços ou sistemas, permitindo o *logon* único.

2.11 Gerenciamento de Chaves Públicas

Como já mencionado neste trabalho, pode-se utilizar a criptografia por chave simétrica, em que a mesma chave serve para criptografar e descriptografar a informação, ou pode-se utilizar os sistemas de criptografia por chave pública, nos quais um par de chaves matematicamente relacionadas é gerado, uma chave privada, a qual ninguém terá acesso, exceto quem gerou, e a chave pública correspondente, podendo ser divulgada e distribuída à vontade. Neste sistema, uma informação criptografada com uma chave só pode ser descriptografada com a chave correspondente.

Aparentemente esse último esquema parece uma solução de criptografia e autenticação bem mais razoável e mais segura. Porém, além de ser um método menos eficiente que o primeiro, um dos seus grandes problemas é como publicar e gerenciar chaves públicas a todos que irão precisar delas para fazer autenticação e criptografia. A infra-estrutura de chave pública (ou *Public Key Infrastructure* - PKI) tem esse como um dos seus objetivos.

A infra-estrutura de chave pública PKI que será descrita, estabelece meios e regras técnicas que têm como objetivo garantir a autenticidade, a integridade e a validade jurídica de documentos em forma eletrônica.

De acordo com (WILSON, 1999) um importante requerimento da infra-estrutura PKI é realizar o gerenciamento de chaves de forma transparente aos usuários finais. Para isso, fazem parte dessa infra-estrutura de gerenciamento de chaves: Certificados Digitais, Autoridades Certificadoras, Autoridades Registradoras, Repositório de Certificados, Sistema de Revogação de Certificados, Lista de Certificados Revogados e Usuários. Todos esses componentes serão discutidos a seguir.

2.11.1 Certificados Digitais

O certificado digital é um arquivo assinado eletronicamente por uma entidade confiável, chamada Autoridade Certificadora (*Certification Authority* ou CA). Um certificado tem como um dos objetivos, associar a chave pública a uma pessoa, Empresa ou outra organização, servindo, assim, como um mecanismo para a divulgação da chave pública. A Autoridade Certificadora verifica a identidade do sujeito (entidade final) e emite o Certificado Digital.

Cabe ressaltar que os Certificados não são secretos ou protegidos, qualquer entidade que conheça a chave pública da CA pode examinar o conteúdo e confirmar a autenticidade de um certificado emitido por esta Autoridade, uma vez que a CA assina os certificados com a sua chave privada.

Dentre os dados de um certificado digital, as seguintes informações estão presentes: chave pública do usuário, número de série do certificado, nome da CA que emitiu o certificado, a assinatura digital da CA, o período de validade do Certificado, entre outras.

A recomendação mais aceita e utilizada para a produção de certificados digitais é a X.509, que está atualmente na versão 3, formulada pela *International Telecommunication Union – Telecommunication Standardization Sector* (ITU-T). Detalharemos este padrão na seção 2.10.3.

Além da distribuição de Certificados, outro papel, não menos importante, de uma CA é a revogação de certificados. Esta ação faz-se necessária sempre que alguma informação do conteúdo de um certificado mudar, ou se o usuário, dono do certificado, sair da Empresa. Cabe a CA incluir os números de série dos certificados revogados em um repositório chamado **Lista de Certificados Revogados** (*Certificate Revocation List*, ou CRL).

2.11.2 Autoridades Certificadoras em VPNs

É possível se utilizar a infra-estrutura de Chave Pública (PKI) nas redes VPN, por meio das configurações do IPSec, padrão este que será detalhado mais adiante nesta dissertação.

Dentro deste contexto, é importante ressaltar que uma Empresa pode possuir a sua própria CA, o que chamamos de ***Autoridade Certificadora Autônoma*** (SILVA, 2003). Esta pode ser uma boa opção para reduzir custos, se as redes VPNs serão exclusivamente utilizadas pela mesma Empresa. Porém, quando se interliga duas ou mais Empresas distintas através de VPN, esta opção pode não ser muito apreciada por parte de uma das Empresas, pois esta terá que confiar plenamente na segurança do servidor que faz o papel de CA na outra Empresa. Neste último caso, talvez a opção mais aconselhável seria a contratação de uma terceira entidade, reconhecida e confiável para prestar este serviço.

Vale ressaltar que o *software* de VPN adotado neste trabalho, o FreeS/WAN, com a aplicação de todos os *patchs*, possui suporte a Certificados Digitais; porém, para a utilização dessa facilidade é necessário que todas as entidades envolvidas em uma conexão VPN (*Gateways* VPNs) estejam configuradas para que utilizem uma CA, podendo esta ser autônoma ou não.

2.11.3 X.509

Conforme mencionado, a ITU-T criou e aprovou um padrão para certificados digitais chamado X.509. Desde sua padronização inicial em 1988, o X.509 passou por três versões. Atualmente, encontra-se na versão 3. A versão da IETF do X.509 v.3 é descrita na RFC 3280, elaborada em abril de 2002. O padrão X.509, descrito na RFC 2459, de Janeiro de 1999, tornou-se obsoleto.

Uma característica do padrão X.509 é a sua proximidade com o modelo OSI, que se contrapõe inclusive aos modelos da IETF como descrito por Tanenbaum:

> O X.509 foi fortemente influenciado pelo mundo OSI, tomando emprestadas algumas de suas piores características (por exemplo, nomenclatura e codificação). Surpreendentemente a IETF aceitou o X.509, embora em quase todas as outras áreas – desde endereços de máquinas até protocolos de transporte e formatos de correio eletrônico – ela tenha ignorado a OSI e tentado fazer tudo da maneira certa (TANEMBAUM, 2003).

Este texto deixa evidente o descontentamento do autor, principalmente em relação a nomenclatura do X.509 que a IETF acatou. Neste padrão, se quiséssemos, por exemplo, endereçar o Professor João, lotado no Departamento de Computação da Universidade Estadual do Amazonas, seu endereço no X.509 seria:

/**C**=BR/O=UEA/**OU**=Computação/**CN**=João

Este padrão é o mesmo adotado no padrão X.500, onde **C** é o país (country), **O** é a organização, **OU** é a Unidade Organizacional e **CN** é o nome comum. Da mesma forma, as Autoridades Certificadoras e outras entidades seriam identificadas.

Um problema com esta nomenclatura é que a mesma gera incompatibilidade com o padrão de nomes Internet. Porém, a versão 3 do X.509 já permite a utilização de nomes DNS em substituição a nomes de padrão X.500. Neste caso, o nome DNS é utilizado dentro do formato X.500, incluindo-o no **CN** (nome comum), como no exemplo seguinte:

/**C**=BR/**OU**=UTecnologia/**O**=Banco do Brasil S.A./**CN**= www2.bancobrasil.com.br

Todos os Certificados X.509 possuem os seguintes campos principais:

- ***Version*** (Versão) - Identifica qual é a versão do padrão X.509 aplicado no Certificado. Três versões são definidas atualmente.

- ***Serial Number*** (Número de Série) - A entidade responsável pela criação do Certificado deve atribuir um número de série para distinguir um Certificado dos demais. Esta informação é usada de diversas maneiras; por exemplo, quando um Certificado é revogado, seu número de série é colocado na Lista de Certificados Revogados (CRL - *Certificate Revocation List*).

- ***Signature Algorithm*** (Assinatura do Algoritmo) - Identifica o algoritmo usado pela Autoridade Certificadora (CA - *Certification Authorities*) para assinar o Certificado.

- ***Issuer*** (Emissor) - É o nome da entidade que assinou o certificado, normalmente uma Autoridade Certificadora (CA). Usar este certificado implica em confiar na entidade que assinou este certificado (Em alguns casos, como o *root* de uma árvore de certificação, uma CA assinará o próprio certificado).

- ***Validity period*** (Período de validade) - Cada certificado tem validade por um determinado período de tempo. Este período é descrito na forma da data e hora de início da validade do certificado até a data e hora que o certificado expira. Tal período pode ser de alguns segundos como pode durar vários anos. A escolha do período de validade depende de uma série de fatores, como o tamanho da chave usada para assinar o certificado ou a quantia que está sendo paga pelo certificado.

- ***Subject name*** (Assunto) - O nome da entidade que a chave pública do certificado identifica. Este nome usa o padrão X.500, sendo um único identificador para cada entidade dentro da Internet. Por exemplo: CN= www2.bancobrasil.com.br, OU= UF Tecnologia, O= Banco do Brasil S.A., C= BR.

- ***Public Key*** (Chave Pública) - Esta é a chave pública da entidade que está sendo chamada, junto com um identificador do algoritmo que especifica que sistema de criptografia a chave pública utiliza e alguns parâmetros da chave associados.

- ***Signature*** (Assinatura Digital) – A assinatura do Certificado assinado pela chave privada da CA através de funções de *hash*. Essa assinatura digital é incluída no Certificado, funcionando como um carimbo de um cartório (Autoridade Certificadora), propiciando autenticidade e integridade ao Certificado.

2.11.4 Processos e Funcionalidades de uma PKI

Os principais processos e funcionalidades de uma infra-estrutura PKI são:

- ***Registro*** – é o processo pelo qual uma entidade torna-se conhecida por uma CA. Este processo pode ser feito por uma entidade específica chamada ***Autoridade Registradora*** (*Registration Authority* ou RA), que estabelece e verifica corretamente a identidade de uma entidade. Caso a Autoridade Registradora não exista, a CA assume este papel. Porém, vale ressaltar, que

uma RA em separado torna mais difícil que um sistema de segurança venha a ser violado.

- ***Certificação*** – é o processo que certifica a associação de uma entidade ou atributo a sua respectiva chave pública, gerando um certificado assinado digitalmente, garantindo assim a autenticidade das informações contidas no Certificado. Este processo é realizado pela Autoridade Certificadora (CA).

- ***Validação*** – é o processo que avalia a autenticidade do certificado, garantindo a sua credibilidade. Este processo verifica a Assinatura Digital do Certificado, utilizando a própria chave pública da CA, checa a existência, ou não, do certificado na Lista de Certificados Revogados (CRL) e verifica o período de validade contido no Certificado.

- ***Revogação de Certificados*** – é o processo que adiciona um Certificado na Lista de Certificados Revogados antes de sua expiração. Este processo é de responsabilidade da CA que distribuiu o Certificado. Esta revogação ocorre sempre que algum atributo contido no Certificado sofre alguma alteração. Geralmente, a CA publica periodicamente os números de série dos Certificados Revogados na CRL, podendo também revogar Certificados a qualquer momento.

- ***Distribuição e Publicação de Certificados*** – é o processo de transferência direta de certificados para seu dono após registro ou a um outro usuário que o requisite para validação de chaves. Uma boa opção é a utilização do LDAP (*light-weight Directory Access Protocol*) como repositório, o qual está descrito na RFC 2251. As CRL's também podem ser depositadas num diretório LDAP ou enviadas periodicamente à comunidade ou serem parte de serviços de consulta on-line.

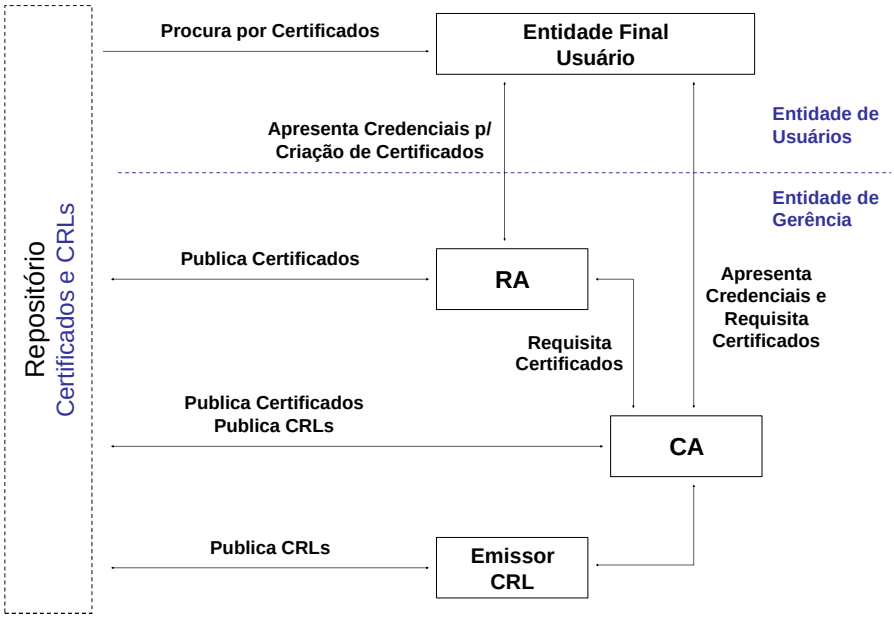
- ***Recuperação de chaves*** – a infra-estrutura PKI possui suporte ao arquivamento e recuperação de chaves para a eventualidade de sua perda.

- ***Geração de chaves*** – processo que gera um par de chaves pública/privada de uma entidade no seu próprio ambiente computacional e posterior envio a CA ou a AR no momento do registro. Pode também ser feita pela CA ou AR.

- ***Atualização de chaves*** - substituição de chaves com validade expirada, perdidas ou reveladas. Também para o caso disso ocorrer com uma CA.

2.11.5 Arquitetura PKI

A figura abaixo apresenta uma visão simplificada do modelo da Arquitetura PKI.



Fonte: RFC 3280

Figura 2.19 - Arquitetura PKI

Pode-se observar que os componentes desse modelo, de acordo com o padrão definido na RFC 3280, são:

- ***Entidade Final***: são os usuários de Certificados PKI ou sistemas de usuários finais sujeitos a Certificados.

- ***CA***: é a *Certification Authority* ou Autoridade Certificadora.

- ***RA***: é *Registration Authority* ou Autoridade Registradora. Sistema Opcional a quem a CA delega certas funções de gerenciamento de chaves.

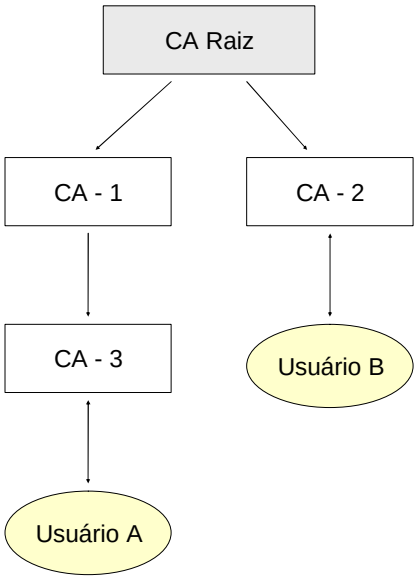
- ***Emissor de CRL***: Sistema opcional para quem a CA delega a função de publicar Certificados Revogados.

- ***Repositório***: Um sistema ou um conjunto de sistemas distribuídos que armazenam Certificados e CRLs.

No modelo apresentado, observa-se que a entidade final realiza requisições às entidades de Gerência, com a finalidade de adquirir ou validar certificado apresentado por uma outra entidade de usuário. Depois que credenciais são apresentadas e verificadas, um certificado é distribuído para o usuário e publicado no repositório.

2.11.6 Estrutura Hierárquica das Autoridades Certificadoras

Para propiciar total disponibilidade e uma maior escalabilidade na infra-estrutura de Chave Pública, as Autoridades Certificadoras são distribuídas e interoperam de forma hierárquica. Desta forma, existe o conceito de Autoridade Raiz que autoriza a atividade das demais CAs. A figura 2.20 apresentada uma visão hierárquica de um cenário hipotético de Autoridades Certificadoras.



Fonte: Adaptado de (SILVA, 2003)

Figura 2.20 – Visão Hierárquica das Autoridades Certificadoras

Neste cenário apresentado, os dois usuários finais, A e B, apresentam certificados emitidos por CA-3 e CA-2 respectivamente. Como ambos os usuários possuem um ponto de confiança em comum na hierarquia de certificações apresentada, que é a CA Raiz, é possível se estabelecer um caminho de certificação de A para B. Vejamos:

> O usuário A tem um certificado assinado pela CA-3, que foi assinado pela CA-1, que foi assinado pela CA Raiz. Agora descendo na hierarquia, tem-se que CA Raiz assinou CA-2, que emitiu o certificado para o usuário B.

Portanto, conclui-se que, neste caso em particular, pode haver autenticação entre os usuários A e B, pois os mesmos possuem um ponto de confiança comum. Diz-se, neste caso, que há um ***Caminho de Certificação*** entre A e B. Em não havendo este caminho entre dois usuários, ou não existindo um ponto de confiança comum na hierarquia de certificações, não poderá ocorrer autenticação entre os mesmos.

Desta forma, os servidores VPN, que desejam estabelecer um túnel VPN, devem possuir um caminho de certificação entre os mesmos para que ambos possam utilizar Certificados Digitais no processo de autenticação.

3 Redes Privadas Virtuais - VPN

Como já mencionado neste trabalho, uma *Virtual Private Network* (VPN) ou Rede Privada Virtual é uma seção de rede protegida formada através de canais desprotegidos, como a Internet. Uma VPN pode permitir que um usuário externo participe da rede interna como se estivesse conectado diretamente a ela utilizando-se de uma rede pública e compartilhada. Outra aplicação de uma VPN, que será a abordagem principal deste trabalho, é o de se interconectar diferentes redes privadas para o tráfego de dados entre elas, utilizando-se como meio, uma rede pública.

Neste trabalho, a rede pública utilizada para a implementação de VPN é a Internet. Desta forma, dizemos que a VPN é baseada na Internet. Porém, deve-se atentar que uma VPN pode também ser implementada através de protocolos no nível de enlace de dados, embora a abordagem de se construir VPNs diretamente sobre a Internet é cada vez mais popular (TANENBAUM, 2003). Uma VPN pode, por exemplo, ser construída utilizando-se uma rede *Frame Relay* ou ATM (SILVA, 2003). Outra denominação comumente encontrada para as redes baseadas na Internet é VPN de nível três, designando assim que se trata de uma VPN implementada na camada de redes. Deste ponto em diante, todas as referências não explicitadas sobre redes VPN deverão ser entendidas como VPN baseadas na Internet.

As organizações de tecnologia da informação em todo o mundo estão buscando cada vez mais atender as demandas de conectividade remota de seus clientes e funcionários, e ao mesmo tempo, lidar da melhor forma possível com aumento de complexidade nas redes e suporte aos usuários, especialmente aquelas organizações que crescem através de novas

aquisições ou fusões entre empresas, as quais necessitam de soluções em conectividade para rapidamente integrar infra-estruturas independentes e incompatíveis. Este pode ser um ponto crítico para o sucesso ou fracasso na relação de negócios. Um requisito emergente na comunidade de usuários é a implementação de *extranets* para dar suporte, a uma antes imprevisível relação com clientes e parceiros de negócio. Não se deve esquecer, porém, os requisitos de gerenciamento e segurança exigidos por estas novas conexões.

As VPNs oferecem soluções de conectividade para estas situações, dando suporte a uma diversidade de meios para fornecer um acesso remoto imediato e redução de custos na conectividade de escritórios, beneficiando-se da infra-estrutura de rede e dos serviços dos provedores de serviços Internet (ISPs) e dos provedores de serviços de rede (NSPs). As VPNs podem oferecer redução de custos, escalabilidade, flexibilidade, gerenciamento e segurança, requisitos essenciais para suportar o crescimento da rede. Outra vantagem, é que as empresas podem obter todos estes benefícios, enquanto mantêm todo o controle central sobre a segurança, gerenciamento e crescimento da rede e conectividade entre escritórios.

Ao invés de depender de aluguel de circuitos dedicados ponto-a-ponto ou da contratação de circuitos virtuais permanentes (*Permanent Virtual Circuits* - PVC)[5], como os implementados pelas redes X.25, *Frame Relay* e ATM, uma rede VPN baseada na Internet utiliza a infra-estrutura aberta e distribuída da Internet para transmitir dados entres sites corporativos. Como a Internet é uma rede pública com uma transmissão aberta de seus dados, uma VPN baseada na Internet deve utilizar mecanismos que possam proteger os dados transmitidos na rede contra acessos não autorizados, preservando a integridade, autenticidade e sigilo das informações trafegadas.

As VPNs baseadas na Internet surgiram então como uma forma mais confiável e segura de transmissão de informações na Internet, pois agregaram a ela características de redes criptografadas, empregando diversos algoritmos de criptografia, autenticação e protocolos de encapsulamento. Associado também ao baixo custo oferecido pela Internet, está a facilidade de implementação e manutenção das redes VPN, o que as tornaram tecnologias acessíveis a qualquer organização que deseje empregar segurança e confiabilidade aos serviços oferecidos através da Internet.

[5] Os PVCs são circuitos cujas conexões são estabelecidas permanentemente.

3.1 Conceitos Gerais de VPN

Existem muitas definições para redes VPN. A seguir, descrevemos algumas:

> Uma VPN é uma conexão que é estabelecida por uma infra-estrutura "pública" ou compartilhada existente, usando tecnologias de criptografia ou autenticação para proteger seu payload. Isso cria um segmento "virtual" entre duas entidades quaisquer que têm acesso (NORTHCUTT et al., 2002).

> Uma VPN é uma rede de comunicação, construída para uso privado de uma Empresa, sobre uma infra-estrutura pública compartilhada (PERLMUTTER, 2001).

> As redes VPN são redes sobrepostas às redes públicas, mas com a maioria das propriedades de redes privadas. Elas são chamadas "virtuais" porque são meramente uma ilusão, da mesma forma que os circuitos virtuais não são circuitos reais e que a memória virtual não é memória real (TANENBAUM, 2003).

> Uma VPN é uma rede corporativa implantada em uma infra-estrutura compartilhada que emprega as mesmas políticas de segurança, gerenciamento e *throughput* aplicadas em uma rede privada. As VPNs são uma infra-estrutura de rede remota (WAN) que pode ser usada para substituir ou aumentar as redes privadas existentes que utilizam linhas privativas (WENSTROM, 2002).

Uma definição simples e menos formal foi dada por Fergson e Huston:

> Uma VPN é uma rede privada construída sobre uma infra-estrutura pública, como a Internet (FERGUSON; HUSTON, 1998).

Diante do exposto, podemos descrever a nossa própria definição de VPN como sendo um ambiente de comunicação com acesso controlado, permitindo conexões seguras para apenas uma determinada comunidade, fazendo uso da infra-estrutura de rede pública já existente, como por exemplo, a Internet.

Como mencionamos neste trabalho, a "infra-estrutura pública compartilhada" é a Internet. O ponto crucial é que, infelizmente, o protocolo de rede da Internet, o IPv4, não possui mecanismos de segurança próprios, tornando a Internet, a princípio, totalmente insegura para tráfego de informações que exijam privacidade, autenticidade e integridade, por exemplo. O próprio protocolo IPv4, em seu cabeçalho, possui apenas uma única opção de segurança, a opção *Security*, no campo *Options* do cabeçalho IP, projetada para especificar 16 níveis de segurança para o datagrama IP (RFC 791), que na prática, não é utilizada por nenhum roteador. Todos o ignoram, pois, de acordo com Tanenbaum (TANENBAUM, 2003), a sua única função prática é ajudar a espiões a descobrirem mais facilmente onde estão as

melhores informações. Desta forma, é necessário adicionar protocolos e procedimentos que possam garantir segurança das informações contidas em um datagrama IP (SILVA, 2003).

A VPN se apresenta como opção de segurança cada vez mais popular para a interconexão de redes corporativas utilizando a Internet. A VPN cria um canal de comunicação com criptografia fim-a-fim, possibilitando uma conexão mais segura entre duas redes distintas, fornecendo privacidade, integridade e autenticidade aos dados transmitidos na rede.

É uma conexão que tem a aparência e várias vantagens de uma conexão dedicada, com a diferença de ser implementada sobre uma rede compartilhada. Através da técnica de "*tunneling*", ou "tunelamento", os pacotes de dados são transmitidos por uma rede pública roteada (como por exemplo, a Internet ou qualquer outra rede disponível comercialmente), em um "túnel" privado que simula uma conexão ponto-a-ponto. Esta abordagem possibilita que o tráfego na rede, gerado por fontes diversas, seja transmitido em uma mesma infra-estrutura, porém, em "túneis" distintos, permitindo que protocolos de rede trafeguem em infra-estruturas incompatíveis. Além disso, pode-se diferenciar o tráfego para ser direcionado a um destino específico e receber diferentes níveis de serviço.

Os *firewalls*, também são instrumentos de segurança que definem um conjunto de regras específicas, cuja ação pode ser bloquear, negar, rejeitar ou aceitar um tráfego específico ou porta que passe por ele. No entanto, um *firewall*, assim como a VPN, é apenas um componente a mais de segurança. A VPN não pode ser vista como uma solução completa de segurança. Neste raciocínio, o risco envolvido em colocar uma rede interna numa rede pública, como a Internet, sem um *firewall* não justifica o esforço. Um importante aspecto de projeto de segurança de rede envolvendo *firewalls* e VPN, que será posteriormente discutido no capítulo 5, será sobre os aspectos da localização ou disposição de um *firewall* dentro de uma rede VPN. Essa análise concluirá que a integração entre *Gateway* VPN e *firewall* é uma das melhores opções para garantia de segurança e uma melhor relação custo e beneficio para interconexão de redes usando uma infra-estrutura pública.

Uma rede VPN implementa criação de túneis de criptografia através da Internet para transmitir informações entre redes privadas. Para os propósitos deste trabalho, podemos definir as VPNs como um serviço de comunicação seguro entre redes privadas de corporações

ou usuários remotos, realizada através de um meio de transmissão inseguro, mais especificamente, a Internet.

Como mencionado, uma VPN tipicamente utiliza a Internet como o meio de transporte para estabelecer conexões seguras com parceiros de negócios, estender comunicações para escritórios regionais e isolados e diminuir significativamente o custo de comunicação para uma comunidade de funcionários crescentemente móvel.

3.2 Funcionamento Básico de uma VPN

Um dos obstáculos a serem superados pelas organizações para se utilizar a Internet como uma rede WAN privada está relacionado à variedade de protocolos existentes e em operação nas redes corporativas, pois, além do protocolo IP, é comum que muitas redes possam operar com protocolos NetBEUI e IPX, por exemplo; e a Internet, em sua camada de rede, trabalha apenas com o protocolo IP. Desta forma, deve ser fornecido um mecanismo para que outros protocolos, além do IP, possam ser transmitidos entre as diversas redes.

Outro problema, já mencionado anteriormente neste trabalho, é que o protocolo de rede da Internet, o IPv4, não possui mecanismos de segurança próprios, tornando a Internet, a princípio, totalmente insegura para tráfego de informações que exijam serviços como integridade, privacidade e autenticidade, pois, a princípio, os pacotes transmitidos pela Internet seguem o formato de texto simples (*plain text*), o que possibilita que um intruso possa monitorar o tráfego e verificar as informações contidas nos pacotes IP, o que representa um grave entrave para aquelas Empresas que anseiam usar a Internet como ferramenta de integração e cooperação com filiais e outras Organizações para troca de informações de negócio consideradas sigilosas e confidenciais.

Felizmente, as VPNs tornaram-se uma alternativa viável para transpor esses dois obstáculos apresentados, através do uso de mecanismos de segurança, baseados no uso de criptografia, e no uso de tecnologia de "*tunneling*", ou tunelamento, a ser descrita posteriormente.

A idéia básica consiste em que ao invés dos pacotes trafegarem na Internet de forma aberta, antes de serem transmitidos, os dados são primeiramente autenticados e criptografados, processo este que utiliza chaves públicas e chaves privadas, e em seguida, os

dados são encapsulados em pacotes IP pelo protocolo do túnel VPN e transmitidos pela Internet. Esta medida possibilitará o sigilo das informações (agora cifrados) mesmo que haja algum tipo de interferência nesse percurso através de uma rede insegura, como é a Internet.

3.3 Modos de Interconexão

As VPNs podem ser implementadas de três maneiras distintas:

- ***VPNs dial-up***, ou VPNs de Acesso Remoto, que provêem conectividade para usuários remotos através de linhas discadas (*dial-up*) ou serviços ISDN. Os maiores benefícios desse método de conexão são: acesso universal, acesso remoto e baixo custo de conexão. Também é chamada de *VPN host-rede*.

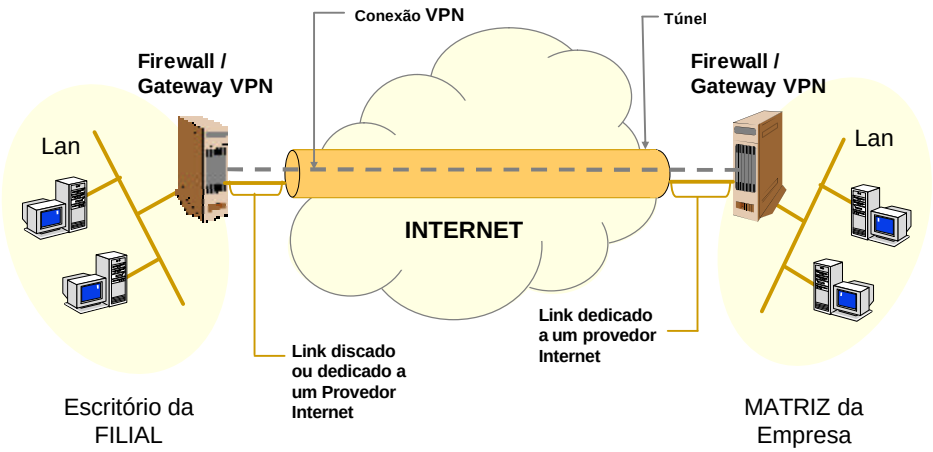
- ***VPNs de Intranet***, também denominadas VPN *Lan-to-Lan*, que propiciam conectividade entre as redes de uma organização, possibilitando a conexão entre filiais, matrizes e outras unidades organizacionais, através de uma infra-estrutura não-confiável, apresentando-se como uma alternativa a contratação de circuitos dedicados e outros *links* WAN.

- ***VPNs de Extranet***, que proporcionam conectividade entre parceiros de negócio, clientes e fornecedores, apresentando-se como uma solução para colaboração, compartilhamento de aplicações e comércio eletrônico entre Empresas diferentes.

Cabe observar que comumente as VPNs de Intranet e Extranet são tratadas indistintamente, são simplesmente referenciadas como sendo VPNs entre redes corporativas, ou ainda VPNs *Lan-to-Lan*. Este trabalho também não fará tal distinção, salvo quando explicitamente mencionado.

Como dissemos, as VPNs implementadas entre redes são utilizadas para interconexão de redes privadas corporativas (*Intranets*), normalmente entre matriz e filiais, ou entre empresas parceiras (*extranets*), ou entre clientes e fornecedores. Desta forma, cada *intranet* ou *extranet* participante de uma conexão VPN necessita de um dispositivo VPN, que pode ser um roteador, um RAS (Servidor de Acesso Remoto) ou um servidor de rede. Na figura abaixo 3.1

é apresentada uma VPN entre duas redes, interligando matriz e filial, onde os dispositivos VPN estão configurados em Servidores *Gateway* VPN, que também possuem a função de *firewall*. Esta configuração cria um segmento virtual entre as duas extremidades de *gateways*, que é chamado de túnel. Vale ressaltar que esta configuração permite que uma rede possa abrir vários túneis VPN, desde que o *software* ou o equipamento responsável pelo tunelamento aceite.



Fonte: Adaptado de (MICROSOFT, 1999)

Figura 3.1 - Conexão VPN conectando dois sites remotos

No cenário apresentado na figura 3.2, exemplificamos uma VPN onde temos a interligação de quatro Empresas parceiras de negócio, utilizando a infra-estrutura da Internet. Esta topologia possibilita que cada *site* corporativo possa abrir um túnel VPN com todas as outras redes corporativas, criando, **virtualmente**, uma rede de topologia totalmente conectada. Desta forma, ao invés de cada Empresa ter que contratar *circuitos* de comunicação dedicados com cada Empresa parceira de negócio, cliente ou fornecedor, cada uma das Empresas deve possuir apenas um *link* com a Internet (o que comumente é fato). Quando a Empresa A precisar trocar dados com o *site* da Empresa C, por exemplo, automaticamente é estabelecido um *túnel* entre elas, e os dados são transmitidos de forma segura. Esta topologia também pode ser utilizada para interconectar *sites* entre filiais e matrizes de uma mesma Empresa (VPN de Intranet). Essa solução permitiria uma comunicação entre as filiais sem que o tráfego tenha que passar obrigatoriamente pela matriz, ou que seja necessário um *link* dedicado adicional entre as filiais, como ocorre em algumas soluções de redes remotas.

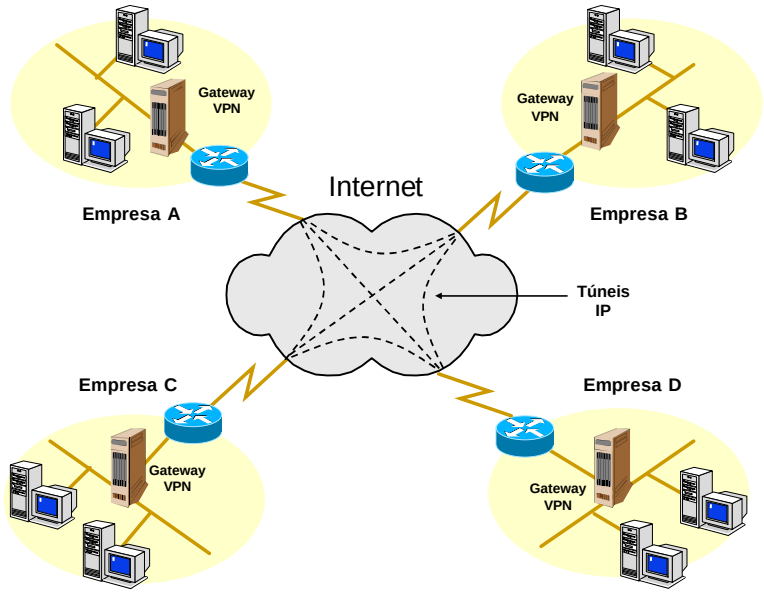
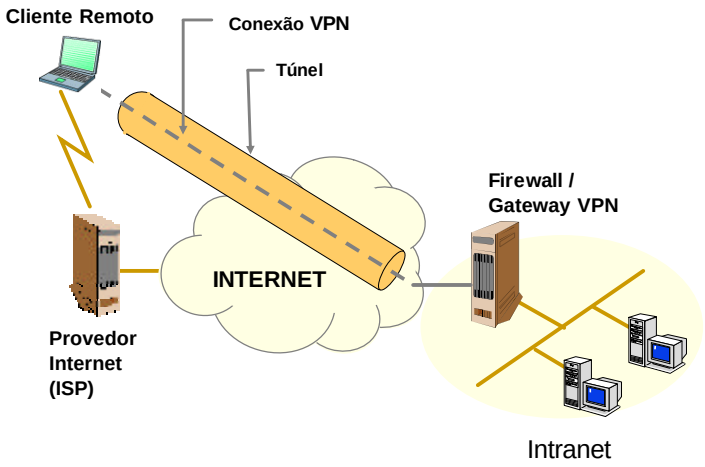


Figura 3.2 – Conexão VPN entre redes corporativas

Como já anteriormente mencionado, outra forma de conexão VPN é aquela entre um cliente remoto e uma rede, VPN de acesso remoto, ou VPN *host-rede*. Este tipo de conexão é bastante comum, possibilita, por exemplo, que funcionários que trabalham em casa (ou que estão viajando e querem economizar interurbanos) e precisam, através de uma conexão discada *local*[6], entrar na rede do escritório de forma segura e utilizar os recursos da mesma como se estivesse ainda em sua mesa de trabalho. A figura 3.3 ilustra este tipo de conexão VPN.



Fonte: Adaptado de (MICROSOFT, 1999)

Figura 3.3 – Conexão VPN de um cliente remoto

[6] Neste contexto, o emprego da palavra *local* significa que a ligação será efetuada a um Provedor de Acesso a Internet, localizado na mesma cidade em que se encontra o cliente remoto.

Neste tipo de solução, o dispositivo VPN (*Gateway* VPN) na rede da Empresa é o mesmo da solução entre redes, enquanto que no usuário remoto o dispositivo VPN pode ser um *software* instalado no PC do usuário ou um serviço prestado pelo próprio Provedor de Acesso a Internet (ISP) ao qual ele se conecta.

3.4 Tunelamento VPN

Tunelamento é o processo de encapsular um tipo de pacote dentro de outro para facilitar algum tipo de vantagem no transporte de uma informação dentro da rede. O tunelamento resolve o problema descrito na seção 3.2, quando mencionava a necessidade de se enviar tráfego pela Internet, diferente do tráfego IP, como por exemplo, pacotes NetBEUI ou IPX, pois, através do tunelamento, estes pacotes podem ser encapsulados por pacotes IP. Desta forma, o tunelamento fornece um mecanismo para que outros protocolos, além do IP, possam ser transmitidos através de uma VPN.

Na figura 3.4, pode-se observar o estabelecimento de um túnel. Os *hosts* não possuem conhecimento do fato de que os pacotes estão sendo criptografados ou que estão sendo enviados por uma rede pública. O tunelamento é totalmente transparente para os *hosts*, nenhum *software* ou configuração especial é exigido para os *hosts*.

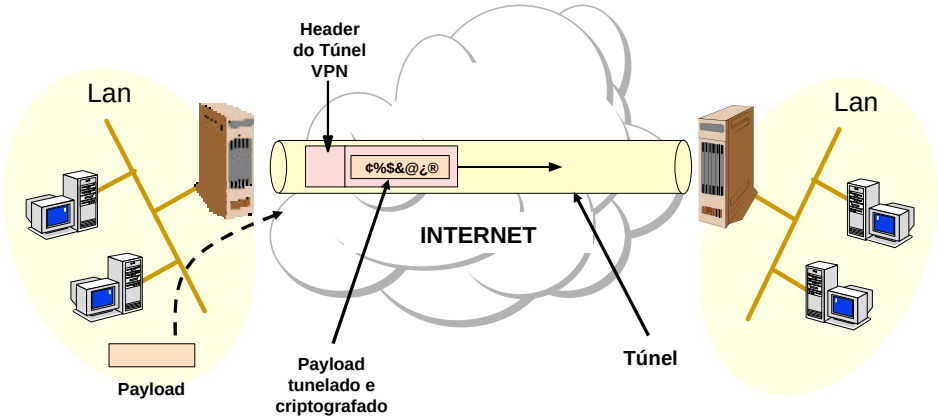


Figura 3.4 – Pacote sendo transmitido via túnel

O processo de envio de uma informação por uma VPN é o seguinte: primeiramente o cliente é autenticado pelo servidor *gateway* VPN ao fazer o pedido para estabelecimento de

uma conexão, também podendo ser autenticado por algum outro servidor de autenticação definido na rede. Posteriormente, a informação é criptografada, possibilitando o envio de formatos de dados não legíveis (textos cifrados), e logo em seguida, estas informações são encapsuladas em pacotes IP. Neste momento, o *Gateway* VPN acrescenta um novo cabeçalho IP, contendo o endereço do *Gateway* VPN origem e o destino, encapsulando o pacote original. A medida em que os pacotes chegam em seu destino, vão sendo reconstituídos e decodificados para um formato legível.

Ao usar tunelamento, embora os endereços dos *hosts* sejam mascarados para o mundo virtual, eles não possuem anonimato completo, pois como os endereços dos *gateways* estão disponíveis nos pacotes, bisbilhoteiros ainda podem determinar quem está se comunicando com quem.

É importante frisar que a criptografia, o encapsulamento e o tunelamento não tornam os pacotes enviados inacessíveis, eles ainda podem ser coletados e analisados , contudo, se for utilizado um algoritmo de criptografia corretamente implementado e adequadamente forte, seu *payload* ainda deverá estar seguro.

Um dos mecanismos mais conhecidos de tunelamento é o GRE (*Generic Routing Encapsulation*) entre um roteador origem e destino. Existem diversos outros protocolos *router-to-router* ou *host-to-host* conhecidos, como o L2TP e o PPTP, os quais serão detalhados posteriormente.

O túnel pode encapsular diferentes protocolos, sendo possível para as VPNs que usam a tecnologia de ‘tunelamento’ simular grande parte das funcionalidades de uma rede privada dedicada.

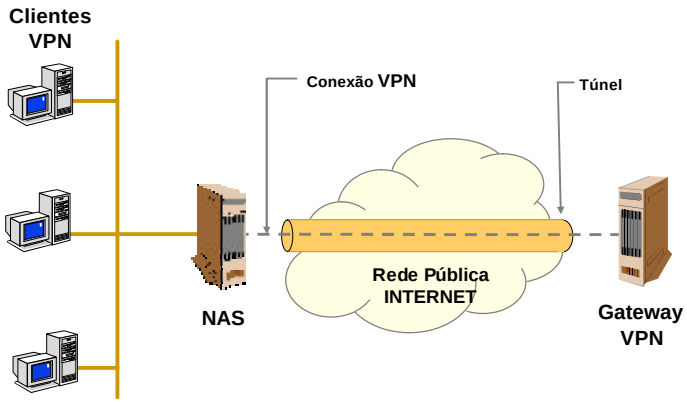
3.5 Tipos de Tunelamento

Diante dos conceitos básicos sobre tunelamento que foram apresentados até o momento, ressaltamos ainda que o tunelamento pode ser considerado de dois tipos:

- ***Tunelamento Voluntário*** – Este tipo de tunelamento ocorre quando a própria estação de trabalho ou servidor de rede utiliza algum *software* cliente de tunelamento (cliente VPN) para estabelecer uma conexão com o Servidor

VPN. Este tipo de tunelamento é comum quando clientes remotos se conectam à Internet, para posteriormente utilizarem o *software* cliente. Neste modo, o túnel VPN termina no cliente. A figura 3.3 ilustra o tunelamento voluntário.

- ***Tunelamento Compulsório*** – Este tipo de tunelamento ocorre quando existe um Servidor de Autenticação para acesso à rede (NAS). Neste caso, o estabelecimento do túnel VPN com o servidor VPN do site remoto e a configuração de autenticação é de responsabilidade dele. Para o cliente, tal tunelamento VPN é transparente, pois os clientes possuem acesso às informações da outras redes através do NAS. Além disso, o cliente não necessita de nenhum *software* cliente VPN para o estabelecimento do túnel. A figura 3.5 ilustra tal esquema.



Fonte: Adaptado de (ORTIZ, 2003)

Figura 3.5 – Tunelamento compulsório

3.6 Algumas Considerações Relevantes sobre as VPNs

Algumas considerações sobre VPN fazem-se necessárias:

A capacidade de processamento dos dispositivos que compõem a VPN é um fator relevante. Essa é questão importante, uma vez que criptografia exige muita capacidade de processamento, e os dispositivos VPN devem criptografar os dados antes de transmiti-los pela Internet, e descriptografá-los antes de enviar os dados recebidos pela Internet para o equipamento da rede privada. As soluções para o problema de necessidade de processamento são a utilização de criptografia por *hardware* dedicado, o que encarece bastante a solução, ou

a utilização de criptografia por *software*. Porém, neste último caso são necessárias CPUs com maior capacidade de processamento.

Se existem limitações críticas de tempo para a transmissão das informações, o uso de VPNs através da Internet pode não ser adequado, pois podem ocorrer problemas de desempenho e atrasos na transmissão sobre os quais a empresa não terá qualquer controle.

Apesar da redução de custos (que pode ou não ocorrer), é preciso muita atenção com a segurança quando se constrói uma WAN utilizando a Internet como meio de transporte. Neste caso, as transmissões da empresa não estarão mais restritas a um *link* dedicado privado, ao invés disso, os dados farão a maior parte do trajeto através de uma teia de roteadores e *hosts* desconhecidos, em território pouco familiar e eventualmente inseguro. Por este motivo, o uso da criptografia nas VPNs é fundamental.

Quando a informação é encriptada no lado emissor, uma chave é necessária para descriptá-la no lado receptor. Os dispositivos que implementam a VPN em cada lado da conexão devem gerenciar esta troca de chaves de forma automática e transparente.

O caso do acesso de usuários remotos à rede através de VPNs é um pouco mais complexo, pois neste caso é necessário algum mecanismo para autenticar o usuário e uma forma qualquer de negociar a troca de chaves. Algoritmos de chaves públicas e assinaturas digitais são utilizados para estas finalidades.

Ressalta-se também que a implementação de uma VPN pode consumir bastante tempo e tornar-se uma grande desvantagem se não houver um planejamento adequado, preocupando-se com a gerência das chaves e a resolução dos problemas encontrados. É importante que se tenha conhecimento de como as redes que se pretende interligar estão funcionando, assim como as suas configurações, pois qualquer imperfeição pode resultar em mais tempo gasto para corrigi-la.

3.7 Tendência: VPNs implementadas pelos ISPs

Nos últimos anos, com o surgimento de tecnologias e produtos que implementam VPNs baseadas na Internet, ou melhor, baseadas no protocolo IP (*network-based* IP VPN), surgiu o interesse cada vez maior das Empresas, principalmente as Concessionárias de Telecomunicações Americanas, em oferecer serviços de VPN para um grande número de clientes sobre os mesmos *backbones*, de maneira escalável e gerenciável (CLERCQ; PARIDAENS, 2002, p. 151). Infelizmente, no Brasil, não existem Empresas que efetivamente possuem este serviço.

A Embratel, por exemplo, possui um serviço de abrangência nacional chamado **IP VPN** que se apresenta como sendo uma VPN implementada em MPLS, a qual possibilita a transmissão de dados, voz e vídeo. A Telemar possui serviço semelhante, também baseado em MPLS. Porém, existe um problema grave nessas soluções: ambas não possuem mecanismo de conexão com a Internet, pois são implementadas através de uma rede fechada MPLS.

Existe uma previsão da indústria e da comunidade em geral, que a tendência dos serviços de conectividade VPN é serem terceirizados por Empresas Concessionárias de Telecomunicações ou por grandes Provedores de Internet. A idéia que está por trás disso é a seguinte: o cliente não terá mais que implementar funções específicas de VPN, como o tunelamento, ou ter servidores de VPN. Os *sites* dos clientes seriam conectados diretamente a roteadores IP pertencentes aos Provedores, e estes manteriam um contexto diferente para cada cliente provido por VPN. Além disso, estes provedores poderiam prover serviços como *Firewall*, Qualidade de Serviços, Detecção de Intrusos, Acesso a Internet, etc.

Produtos e técnicas já existem para a implementação destes Provedores. O que há é uma grande expectativa no mercado de VPN. Espera-se que os grandes provedores de Internet (ISPs) deverão oferecer serviços adicionais proporcionados pelas VPNs. Porém, as redes VPNs só receberão impulso significativo quando os próprios provedores de serviços de telecomunicações e Internet adequarem as suas infra-estruturas. O que falta é investimento e uma avaliação se este tipo de negócio se enquadra às realidades e necessidades das Empresas Brasileiras. Para saber maiores informações sobre este assunto e de como essa idéia pode ser implementada leia (CLERCQ; PARIDAENS, 2002).

Os provedores de Internet que partirem na frente, e com qualidade de serviço, ganharão a preferência do mercado de telecomunicações. Pesquisas da I*nternational Data Group* (IDG) indicam que ocorrerá, nos próximos anos, um crescimento exponencial da terceirização das VPNs através da contratação de provedores de serviço, o que poderá caracterizar no Brasil uma nova tendência na área de Telecomunicações dentro em breve. Resta saber, se os clientes estarão dispostos a pagar por mais esse serviço, ou se os mesmos irão preferir implementar suas próprias soluções de VPN, preferindo ter um controle mais centralizado da sua segurança.

4 PROTOCOLOS PARA VPN

Existem diversos protocolos disponíveis para a construção de redes VPN e que ao mesmo tempo garantem segurança e privacidade da conexão. Desta forma, no processo de implementação de uma VPN é essencial a definição do protocolo para se realizar o tunelamento. A principal questão é: qual protocolo deve ser utilizado? A resposta para este questionamento é: depende de cada caso. Cada situação deve ser analisada. A aplicabilidade de cada protocolo depende dos requisitos e necessidades dos clientes, do problema que está sendo apresentado e da solução que se deseja obter. Depende, também, do controle (quem detém e porque o controle é necessário) e de como é feita cada implementação destes protocolos (túnel voluntário ou compulsório). A segurança nas conexões é garantida por mecanismos de autenticação e controle de acesso usando canais criptografados.

É fato que durante já algum tempo as empresas de telecomunicações têm construído VPNs que aparecem para o cliente como se fossem uma rede privada, mas fisicamente elas compartilham um *backbone* com outros clientes. As redes VPN têm sido construídas pelas concessionárias de telecomunicações sobre os protocolos X.25 (No Brasil, a Rede Nacional de Pacotes – Renpac), Frame Relay e ATM. Porém, atualmente o *Frame Relay* destaca-se como o protocolo mais difundido e utilizado para implementação de redes privadas virtuais e que apresenta uma ótima relação custo/benefício. O X.25 está praticamente extinto, principalmente pelo seu baixo desempenho, limitação de velocidade (abaixo de 64Kbps) e não adequação aos padrões atuais de qualidade de circuitos (foi feito para circuitos de baixa qualidade). O ATM, no Brasil, é pouco utilizado pelas concessionárias de telecomunicações, principalmente devido aos altos custos na aquisição de equipamentos. Cabe observar, que

estas VPNs, baseadas nesses protocolos, dependem de uma infra-estrutura de rede específica montada para cada protocolo pela própria concessionária de telecomunicações. Porém, como já anteriormente mencionado, este trabalho foca a implementação de redes VPN corporativas que possam utilizar a infra-estrutura da Internet já montada, baseada no protocolo IPv4.

Desta forma, com o objetivo de garantir privacidade, autenticidade e integridade em uma infra-estrutura pública, como a Internet, a VPN deve necessariamente utilizar controle de acesso e criptografia para garantir esses requisitos mínimos de segurança. Por conseguinte, os usuários possuem disponível uma diversidade de protocolos que garantem a segurança dos dados. Estes protocolos oferecem a vantagem da transferência de informação através de "túneis". Os principais que serão estudados nesta pesquisa são: PPTP, L2F, L2TP, IPSec. Estes protocolos encontram justificativas de sua existência nos diferentes objetivos para os quais as VPNs foram inicialmente usadas. Para alguns, elas vieram substituir os servidores de acesso remoto, passando as conexões a serem feitas através de um provedor local de serviços Internet (no caso de VPNs de acesso remoto). Para outros, as VPNs estabeleceram os chamados "túneis seguros" para o tráfego entre LANs protegidas. Os protocolos refletem essa dualidade: PPTP, L2F e L2TP são voltados para VPNs de acesso remoto (VPNs *dial-up*), enquanto o IPSec focaliza em soluções *Lan-to-Lan*. A escolha, portanto, deverá se basear no tipo de VPN que se deseja implementar (dial-up, *Lan-to-Lan* ou uma combinação das duas) e também nos aspectos de segurança de cada protocolo que se quer adotar.

Inicialmente abordaremos o protocolo PPP, mencionando suas características, vantagens e desvantagens. Em seguida analisaremos os protocolos de tunelamento PPTP, L2TP e o padrão IPSec.

4.1 PPP (Point-to-Point Protocol)

O Protocolo Ponto-a-Ponto (PPP) é um dos protocolos de enlace de dados mais populares de se interligar *hosts*, através de linha discada, a Provedores de Acesso (ISP). O usuário remoto configura uma conexão PPP entre o *host* remoto e o servidor de acesso remoto (*Remote Access Server* – RAS).

A RFC 1661 (SIMPSON, 1994), que define o protocolo PPP, apresenta três componentes principais definidos para este protocolo, são eles:

1. Um método para encapsular datagramas multi-protocolos.

2. Um protocolo de controle de enlace, o LCP (*Link Control Protocol* – protocolo de controle de enlace), usado para estabelecer, testar, negociar opções e desativar linhas de comunicação PPP.

3. Um protocolo para negociar opções da camada de rede de modo independente do protocolo da camada de rede a ser utilizado. Este protocolo é o NCP (*Network Control Protocol* – Protocolo de Controle de Rede). Ele deve ser um específico para cada camada de rede aceita.

Um usuário doméstico, por exemplo, pode se conectar a um Provedor de Acesso à Internet (ISP) através de uma linha discada e se tornar temporariamente um *host* da Internet. A estação do usuário, usando uma linha discada e um modem, realiza uma conexão física (comutação de circuitos) com o modem do RAS (Servidor de Acesso Remoto) pertencente ao Provedor. Após o estabelecimento dessa conexão física, o PC do usuário envia ao roteador do Provedor uma série de pacotes de negociação LCP no campo da carga útil de um ou mais quadros PPP, e esses pacotes e suas respostas selecionam os parâmetros PPP a serem utilizados. Após a negociação dos parâmetros, pacotes NCP são enviados para configurar a camada de rede, como por exemplo, o recebimento de um endereço de rede IP para o computador do usuário. Depois de configurado os parâmetros da camada de rede, o computador do usuário torna-se um *host* da Internet e pode enviar e receber pacotes IP, da mesma forma que outros *hosts* fisicamente conectados na Internet.

Observa-se que o protocolo PPP requer que a empresa ou Provedor de Acesso à Internet (ISP) possua um banco de modems para acesso discado, sendo este dimensionado para o número de usuários remotos que irão se conectar. Se a conexão for para uma rede específica de uma Empresa, esta deverá possui um RAS com este banco de modems, além de linhas telefônicas próprias. Neste cenário, o usuário ficaria conectado a rede da Empresa. Se a intenção for realizar uma conexão com a Internet, o usuário deverá realizar a ligação para um Servidor de Acesso Remoto de um Provedor, desta forma, após o estabelecimento da conexão PPP, o usuário torna-se um *host* da Internet. Em ambos os casos, o túnel é voluntário.

Maiores detalhes sobre o protocolo LCP podem ser obtidos na própria RFC 1661 que define o PPP. O protocolo NCP é específico para cada protocolo de rede.

A figura abaixo ilustra um possível cenário de conexão PPP. Nesta, tem-se um cliente remoto que estabelece uma conexão PPP através da Rede de Telefonia até o Servidor de Acesso Remoto (RAS) de uma Empresa para que este possa ter acesso à Rede Interna da Empresa, tendo a possibilidade ainda de estabelecer uma comunicação com a Internet através da rede local, ou pode também representar uma conexão a um Provedor de Internet (ISP), porém, possivelmente este cliente teria acesso apenas ao roteador de borda que dá acesso à Internet e a outros poucos serviços, como, por exemplo, serviços de FTP e Web.

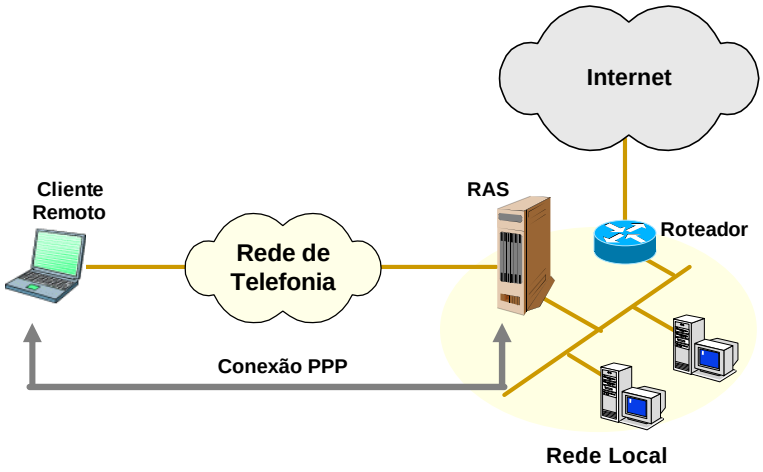


Figura 4.1 – Cenário de Conexão PPP

Em resumo, as principais características do protocolo PPP São:

- Capacidade de encapsular diversos protocolos. Os pacotes dentro do *frame* PPP não precisam ser obrigatoriamente IP. O PPP pode encapsular protocolos como o IPX e o NetBEUI, por exemplo.

- O PPP trata a detecção de erros.

- Permite que endereços IP sejam negociados em tempo de conexão (uso do DHCP), isto é, aceita atribuição dinâmica de endereços IP. Isto é feito através do NCP.

- Permite realizar autenticação de usuários.

Este protocolo não é objeto de maior pesquisa e detalhamento neste trabalho, pois o mesmo é utilizado para conexões de Clientes Remotos, e este trabalho preocupa-se com a interligação entre redes corporativas que já estão inseridas na Internet, porém, é necessário o

conhecimento básico do funcionamento deste protocolo para um melhor entendimento dos protocolos de tunelamento para construção de redes VPN.

4.2 PPTP (Point-to-Point Tunneling Protocol)

O protocolo PPTP, ou *Point-to-Point Tunneling Protocol*, é um protocolo da camada 2 que foi desenvolvido por um consórcio de Empresas de tecnologia da Informação, incluindo a US *Robotics* (parte da 3Com), *Microsoft*, *Ascend Communication* (parte da *Lucent*) e ECI *Telematics*, porém, foi amplamente popularizado através das implementações realizadas pela Microsoft (NORTHCUTT et al., 2002, p. 216) nos seus Sistemas Windows. Ele está documentado na RFC 2637 (HAMZEH et al., 1999).

Este protocolo buscou atender aos interesses de fornecedores de *hardware* que participaram da sua concepção, fornecedoras de Servidores de Acesso Remoto e aos interesses da Microsoft, fornecedora de *software*, para o desenvolvimento de soluções em conectividade através do uso da Internet como sua própria rede privada virtual e segura (uma VPN). Ela foi concebida e integrada ao servidor RAS (*Remote Access Services*) que faz parte do Windows NT Server.

Discorrendo tecnicamente, ele é baseado em uma arquitetura Cliente/Servidor que se propõe a criar um canal seguro de comunicação entre sistemas de rede Microsoft e servidores de acesso remoto, realizando um tunelamento do *Point-to-point Protocol* (PPP) através de uma rede IP (KAEO, 1999). Resumidamente, ele utiliza o PPP para fazer as conexões e, em seguida, encapsula os dados através do protocolo de encapsulamento genérico de roteamento, ou *Generic Routing Encapsulation* – GRE, e os envia à outra extremidade da VPN, um *gateway* PPTP. O uso deste protocolo, dá ao PPTP flexibilidade de lidar com outros protocolos diferentes do IP, como o IPX, da Novell, e o NetBEUI, da Microsoft. Quando os pacotes chegam na outra extremidade, o servidor PPTP destino, eles são desencapsulados, ou seja, são retirados os cabeçalhos GRE, cada pacote segue o seu caminho determinado pelo endereço do cabeçalho original.

Desta forma, ele possibilita que os usuários possam discar para um provedor de Acesso a Internet (ISP) local ou conectar-se diretamente à Internet, e acessar sua rede com a mesma facilidade como se estivessem em suas próprias mesas de trabalho.

O PPTP possui capacidade de PPP para autenticação do usuário usando vários protocolos associados, como o *Password Authentication Protocol* (PAP), o *Challenge Handshake Authentication Protocol* (CHAP), o *Microsoft-Challenge Handshake Authentication Protocol* (MS-CHAP) ou o *Extensible Authentication Protocol* (EAP). Porém, apesar dessa diversidade de opções, o protocolo PPTP apresentou inicialmente várias críticas da comunidade principalmente pela insegurança do seu método de autenticação, que utiliza a autenticação disponível no PPP - o MS-CHAP (NORTHCUTT; ZELTSER; WINTERS; FREDERICK; RITCHEY, 2002) e também por não oferecer serviços de criptografia. O PPTP pega os dados criptografados através do método de criptografia MPPE (*Microsoft Point-to-point Encryption* – Criptografia Ponto-a-Ponto da Microsoft)[7] e faz os encapsulamentos. As chaves de criptografia utilizam a senha do usuário como base, ou seja, se esta senha for fraca, como palavras encontradas em dicionários ou números de telefones, a chave também será.

O protocolo PPTP permite o estabelecimento de túneis ponto-a-ponto individuais a partir de um cliente remoto. Observa-se, entretanto, que não há nenhuma participação do Servidor de Acesso Remoto (RAS) na negociação PPTP e no estabelecimento do túnel. A figura a seguir apresenta um cenário de conexão PPTP. Neste cenário, o cliente estabelece uma ligação *dial-up* com o RAS, usando o protocolo PPP, entretanto, a sessão PPP encerra-se no RAS. A sessão PPTP subseqüente é estabelecida entre o cliente e o servidor PPTP que o cliente deseja acessar.

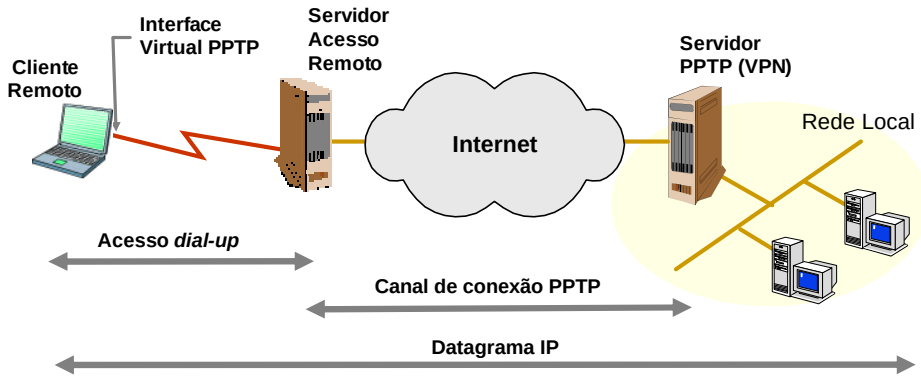


Figura 4.2 – Conexão PPTP

Após o cliente remoto realizar a conexão PPP, uma segunda conexão é realizada sobre a conexão PPP existente, interligando este usuário ao servidor PPTP. Nesta conexão, os pacotes IP são encapsulados pelo PPTP, que por sua vez, são encapsulados pelo PPP. A figura

[7] O protocolo MPPE utiliza chaves de criptografia de 40, 56 e 128 bits

4.3 ilustra os pacotes que são trafegados em uma conexão PPTP, descrevendo passo a passo os pacotes envolvidos.

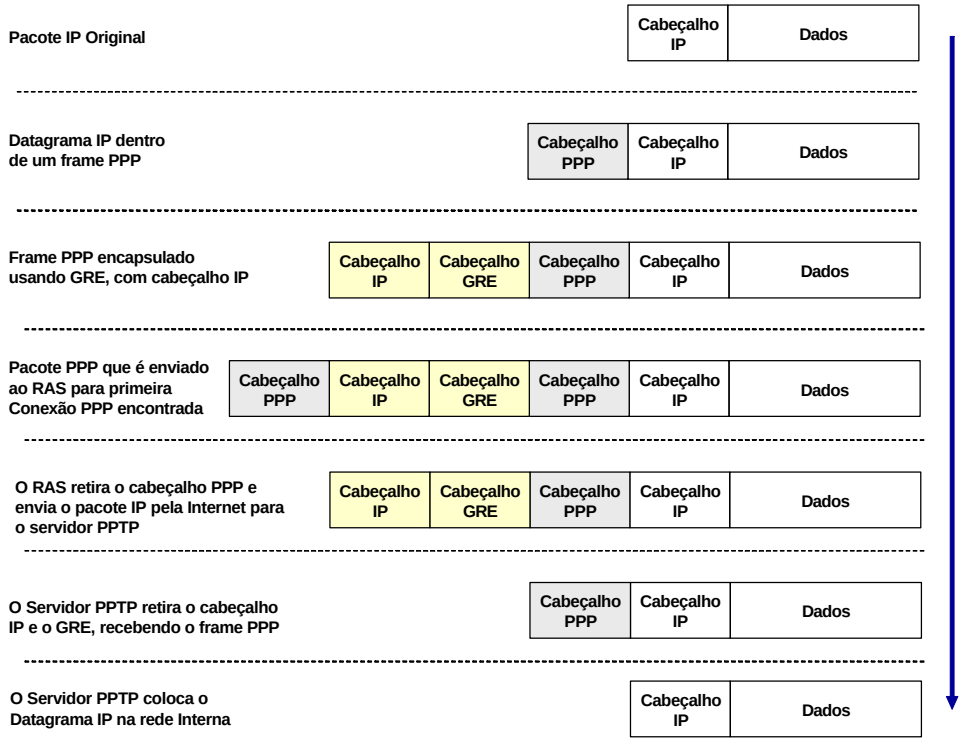


Figura 4.3 – Pacotes envolvidos em uma conexão PPTP

Em resumo, a comunicação PPTP envolve três processos, a saber:

- **Processo de conexão e comunicação PPP** – processo em que o cliente remoto usa PPP para se conectar ao RAS, ou a um Provedor de Internet, utilizando linha telefônica ou algum serviço ISDN de comunicação. Neste, o PPP é utilizado para iniciar e terminar conexões físicas, para autenticar usuários e para criar datagramas PPP contendo pacotes criptografados.

- **Processo de conexão de controle PPTP** – processo que cria um controle de conexão desde o cliente até o servidor PPTP. Essa conexão utiliza TCP e é chamada de túnel PPTP.

- **Processo de tunelamento de dados PPTP** – processo que cria os datagramas IP contendo os pacotes PPP criptografados e os envia através do túnel PPTP

até o servidor PPTP, que finalmente desmonta os pacotes recebidos e descriptografa os pacotes PPP para que sejam enviados à rede corporativa.

Cabe observar que este mecanismo de tunelamento não especifica um esquema de criptografia particular, embora, normalmente utilize o protocolo MPPE (*Microsoft Point-to-Point Encryption*) para este fim, como já havia sido mencionado, o qual adiciona privacidade dos dados ao Serviço de rede Dial-up dos sistemas da Microsoft. Para saber maiores informações sobre os aspectos criptográficos do MPPE em redes Microsoft consulte (PALL; ZORN, 2000) e (GRANADO; VIEIRA; GEUS, 1999). Granado, Vieira e Geus (1999) mencionam inclusive que o protocolo MPPE é **fraco** e **suscetível a ataques** de dicionário, podendo ser feitos ataques contra a cifragem e vários ataques de negação de serviço neste protocolo.

O MPPE criptografa pacotes IP na estação cliente antes de serem transmitidos pelo túnel PPTP. Quando o cliente faz a negociação PPP com o terminador túnel, inicia-se a sessão criptografada. O MPPE utiliza o MS-CHAP para fazer a autenticação do usuário.

Para que tenhamos uma idéia sobre as críticas que a Microsoft recebeu acerca da fragilidade desse protocolo de criptografia, tenhamos como exemplo a seguinte conclusão:

> A implementação do protocolo PPTP pela Microsoft é uma outra mostra de ingenuidade e descuido criptográfico. Infelizmente, não parece haver muita saída nesse caso a não ser refazer o MPPE. De acordo com os últimos *drafts* do MPPE disponíveis na Internet, este não parece ser o caso. Entretanto, a padronização de L2TP e o uso de IPSec poderão trazer segurança a redes Windows NT. Felizmente, a utilização do Kerberos, um sistema de autenticação devidamente escrutinizado, no Windows NT5.0/2000, como um sistema para centralizar a autenticação de todos os serviços e acesso a recursos, e a presença de um Sistema de Arquivos Cifrados são boas tendências na busca de um sistema mais robusto e seguro (GRANADO; VIEIRA; GEUS, 1999).

Uma das grandes vantagens deste mecanismo de tunelamento é a sua facilidade de uso e aplicabilidade para clientes remotos. Eles precisam, apenas, fazer uma conexão local com uma rede pública de dados (Internet) e a partir daí, criar um túnel privado do sistema cliente até o ponto remoto desejado. Outra vantagem desse mecanismo de tunelamento é que o mesmo é totalmente transparente ao Servidor de Acesso Remoto e a toda infra-estrutura Internet, não sendo necessária nenhuma configuração especial no RAS. O RAS ou o Provedor Internet utilizado para conexão simplesmente repassa o tráfego PPTP da mesma maneira que

processa o tráfego IP. O túnel PPTP pode se estender por vários provedores sem a necessidade de configurações explícitas.

Conclui-se, portanto, que este mecanismo de tunelamento também não é adequado para conexões VPN entre redes corporativas, pois a maioria das aplicações da VPN PPTP é destinada aos usuários *roaming*, ou seja, aqueles que se deslocam constantemente, não se aplicando a conexões entre redes. Desta forma, a solução proposta neste trabalho não utilizará este mecanismo de tunelamento.

4.3 L2TP (Layer 2 Tunneling Protocol)

O L2TP (*Layer 2 Tunneling Protocol*), ou Protocolo de Tunelamento da Camada 2, foi desenvolvido pela IETF (*Internet Engineering Task Force*) com o objetivo de se estabelecer um padrão para o encapsulamento de *frames* PPP para a construção de redes VPN de acesso remoto (*dial-up*) como alternativa aos protocolos de tunelamento até então desenvolvidos e propostos para padronização (o PPTP e o L2F). Desta forma, o IETF resolveu convergir neste padrão as melhores características técnicas dos protocolos PPTP e do protocolo de tunelamento que era implementado pela CISCO, o L2F (Layer 2 Forwarding). Porém, os desenvolvedores do PPTP não aceitaram essa imposição e resolveram continuar sozinhos o desenvolvimento e aprimoramento do PPTP, enquanto os desenvolvedores do L2F decidiram parar e assumir o L2TP da IETF como padrão. O L2TP está descrito na RFC 2661 (TOWNSLEY et al., 1999).

Como conseqüência dessa convergência, o L2TP oferece as melhores funções e características destes dois protocolos, além dos benefícios adicionais como o túnel multiponto, o qual permite que um único cliente inicialize várias VPNs. Na prática, um cliente remoto pode criar simultaneamente uma conexão para acessar uma aplicação de banco de dados da corporação e outra para acessar a intranet. Outra característica do L2TP é que ele suporta qualquer protocolo roteado como o IP, IPX, *Appletalk* e qualquer tecnologia e protocolo de *backbone* WAN (ATM, X.25, *Frame Relay*, SONET).

De acordo com a terminologia da RFC 2661, os principais componentes do L2TP são:

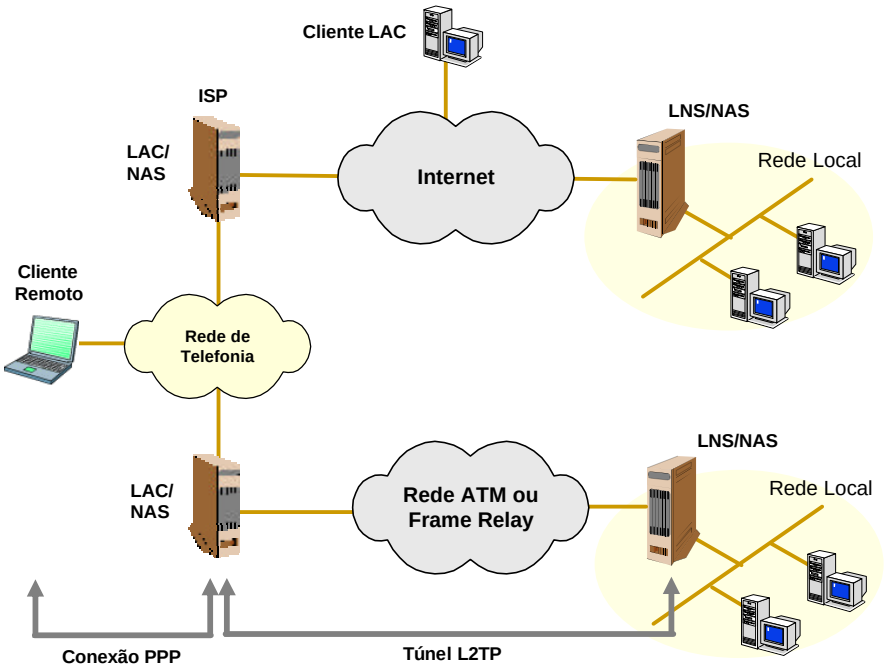
- **Sistema Remoto** – é o sistema do Cliente (sistema final) ou roteador conectado a um servidor de acesso de uma rede, podendo ser tanto a origem como o destino de uma comunicação. Também é denominado Cliente *dial-up* ou simplesmente Cliente Remoto.

- **LAC (**L2TP *Access Concentrator*), ou concentrador de Acesso L2TP – ele concentra o acesso de todas as conexões L2TP. O LAC realiza um tunelamento L2TP com a outra extremidade do túnel, o L2TP *Network Server* (LNS), ou Servidor de Rede L2TP, que na verdade é o Servidor L2TP na rede remota (NAS). O LAC situa-se entre o Cliente e o LNS e realiza o roteamento de pacotes entre estes dois componentes. A conexão entre o LAC e o Cliente pode ser local ou através de *link* PPP.

- **LNS** (L2TP *Network Server*), ou Servidor de Rede L2TP – é o servidor de rede que atua como terminação lógica do túnel L2TP. Atua como um Servidor de Acesso Remoto.

- **NAS** (*Network Access Server*), ou Servidor de Acesso a Rede – é um dispositivo de rede utilizado para acesso pelos clientes remotos a uma rede local. Um Servidor de Acesso de Rede (NAS) pode servir como um LAC, um LNS, ou ambos.

A figura 4.4 descreve um cenário típico do L2TP. Nesse cenário, é apresentada uma topologia típica de como pode ser feito o tunelamento do PPP entre um cliente remoto ou um cliente LAC e o LNS.



Fonte: Adaptada da RFC 2661

Figura 4.4 - Cenário Típico do L2TP

O L2TP funciona da maneira descrita a seguir:

O cliente remoto inicia uma conexão PPP até o LAC. O LAC então realiza o tunelamento do PPP até o LNS através da Internet ou através de outra rede, como, por exemplo, uma rede ATM ou *Frame Relay*. Desta forma, o cliente acessa a rede local e obtém um endereço válido nesta rede através de uma negociação NCP. O processo de autenticação e autorização pode ser provido através de um servidor de domínio dessa rede local (TOWNSLEY; VALENCIA; RUBENS; PALL; ZORN; PALTER, 1999).

É importante descrever as principais características do protocolo L2TP, que são:

- O L2TP foi desenvolvido para suportar os dois modos de tunelamento, voluntário e compulsório. No modo voluntário, o túnel é iniciado pelo cliente remoto. Já no modo compulsório, o túnel é automaticamente criado, exigindo, desta forma, que o NAS do provedor esteja pré-configurado com informações do túnel e de autenticações dos usuários.

- O L2TP herdou os mecanismos de segurança (criptografia e autenticação) do PPP. Portanto, ele não autentica os pacotes que irão sair do cliente remoto,

somente autentica o usuário remoto ao LNS. Ele também não provê mecanismos de gerência de chaves.

- Ao contrário do PPTP, o L2TP utiliza o protocolo UDP para fazer a manutenção de túnel VPN [8].

- Geralmente é utilizado em conjunto com o IPSec com o intuito principalmente de oferecer autenticação de pacotes e suporte a NAT (ORTIZ; FERREIRA, 2003).

Em (TOWNSLEY et al., 1999) e (ORTIZ; FERREIRA, 2003) podem ser encontradas informações mais detalhadas sobre quais mensagens são trocadas entre LAC e LNS para a manutenção dos túneis L2TP, assim como, seus formatos e um maior detalhamento do funcionamento deste protocolo.

4.4 MPLS (Multiprotocol Label Switching)

MPLS (*Multiprotocol Label Switching*), ou comutação de rótulos multiprotocolo, descrito na RFC 3031 (ROSEN; VISWANATHAN; CALLON, 2001) foi desenvolvido inicialmente pelos fabricantes de roteadores que tinham como objetivo principal a busca de métodos de roteamento melhores, que pudessem reduzir ao máximo o processamento necessário para cada roteador redirecionar o pacote. Os trabalhos dos fabricantes de roteadores como a Cisco, a Force10 Networks e a Juniper Networks, concentraram-se na inclusão de um rótulo (*label*) no início de cada pacote e na execução do roteamento baseado no rótulo, e não mais no endereço destino (TANENBAUM, 2003).

A idéia era fazer desse *label* um índice para uma tabela interna, fazendo como se a tarefa de localização da linha de saída de um pacote fosse apenas uma questão de pesquisa em uma tabela, tornando o roteamento muito mais rápido.

Desta maneira, a solução encontrada pelos fabricantes de roteadores citados, aproximou-se bastante dos circuitos virtuais como, por exemplo, os implementados pelos protocolos X.25, ATM e *Frame Relay*, que também trabalham com o uso de rótulos.

[8] No Linux, esta manutenção é feita com UDP através da porta 1701 (ORTIZ; FERREIRA, 2003).

O grande problema enfrentado foi sobre a localização do rótulo, pois os pacotes IP não foram projetados para circuitos virtuais. Não existe nenhum campo disponível para os rótulos dos circuitos virtuais dentro do cabeçalho IP. Desta forma, a solução encontrada foi se adicionar um cabeçalho MPLS antes do cabeçalho IP e se utilizar cabeçalhos PPP para encapsular o MPLS entre os roteadores. A figura 4.5 apresenta o datagrama IP dentro de uma infra-estrutura MPLS.

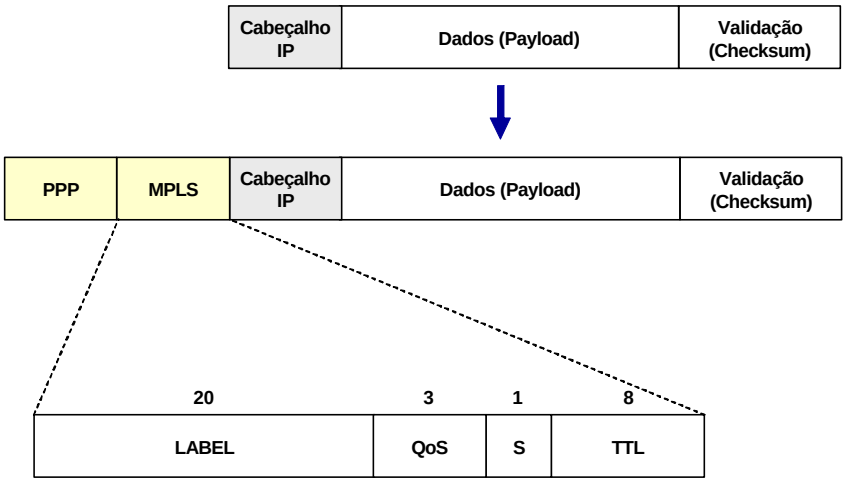


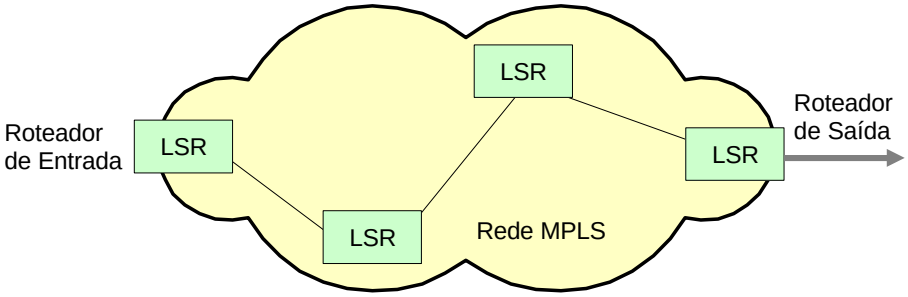
Figura 4.5 - Datagrama IP dentro de uma infra-estrutura MPLS

Como observado na figura 4.5, o cabeçalho MPLS possui quatro campos. Porém, o principal desse cabeçalho que devemos nos ater neste trabalho, é o campo *Label*, o qual contém o rótulo do circuito. Vários cabeçalhos MPLS podem ser inseridos no mesmo pacote (esta situação apresenta-se quando existe a necessidade da existência de diferentes *labels* entre os *end-points*, o *label* **S** indica o empilhamento ou não desses *labels*).

Os roteadores que são capazes de entender os pacotes MPLS são chamados *Label Switch Router* (LSR). O caminho percorrido pelo pacote é denominado *Label Switch Path* (LSP). E o protocolo de comunicação entre os elementos de rede ou roteadores é chamado de *Label Distribution Protocol* (LDP).

Com base nas características apresentadas até o momento sobre o MPLS, facilmente percebemos que o MPLS pode ser utilizado amplamente para a construção de redes VPN, pois garante um isolamento completo do tráfego através da criação de tabelas de rótulos usadas para roteamento exclusivas de cada circuito virtual (SILVA, 2003). O problema dessa solução

para a construção de VPN é que a mesma depende da existência de roteadores LSR na rota entre as entidades comunicantes. A figura 4.6 apresenta um esquema de rede MPLS.



Fonte: Adaptada de (SILVA, 2003)

Figura 4.6 - Componentes de uma Rede MPLS

4.5 IPsec

IPSec, ou *Security Architecture for IP*, é um conjunto de protocolos que define especificações e uma arquitetura para prover serviços de segurança na camada IP, podendo ser aplicado tanto a ambientes IPv4 como a ambientes IPv6 (KENT; ATKINSON, 1998a). Este padrão foi definido em 1998 em meio a intermináveis discussões sobre qual seria a melhor camada para se inserir a criptografia na Internet - em uma *camada fim-a-fim* ou na camada de redes. Muitos especialistas em segurança acreditavam que a melhor abordagem, a qual realmente iria garantir a segurança na Internet, seria aquela que inserisse criptografia e verificações de integridade na camada de aplicação ou na camada de transporte. Entretanto, outro grupo de especialistas, com uma visão totalmente oposta, acreditava que a camada de rede deveria autenticar e codificar pacotes, pois, desta maneira, os mecanismos de segurança a serem implementados na camada de rede seriam totalmente transparentes ao usuário. E seus principais argumentos contra a utilização de criptografia na camada de aplicação foram:

- a implementação de mecanismos de segurança na camada de aplicação exigiria a substituição de todas as aplicações (TANENBAUM, 2003) e

- a codificação na camada de rede não impediria que usuários mais conscientes da segurança a implementassem na camada de aplicação e, até certo ponto, isto poderia ajudar aqueles usuários desprovidos de consciência em segurança (TANENBAUM, 2003).

Diante destes argumentos, esta abordagem foi ganhando cada vez mais apoio, inclusive do IETF (*Internet Engineering Task Force*), culminando nas definições de padrões que compõem o IPSec, descritas inicialmente pelas RFCs 2401, 2402, 2406, 2408 e 2410, entre outras. A RFC 2401 definiu a Arquitetura de Segurança para o protocolo IP (IPSec) e o seu funcionamento. A RFC 2402 descreve o protocolo *Authentication Header* (AH) que integra o IPSec e provê serviços de integridade, autenticação e de não-repúdio nas conexões. A RFC 2406 descreve o protocolo *Encapsulating Security Payload* (ESP), também integrante do IPSec, que provê serviços de confidencialidade (criptografia) e de limite de fluxo de tráfego. A RFC 2408 descreve as Associações de Segurança e o protocolo de gerenciamento de chave, o ISAKMP (*Internet Security Association and Key Management Protocol*). A RFC 2410 descreve um algoritmo de criptografia nulo para ser utilizado no IPSec. Esta solução atende aos usuários que não desejam utilizar a criptografia no IPSec, por esta ser dispendiosa em termos computacionais, permitindo assim o uso de um algoritmo nulo.

Em resumo, o IPSec é um padrão que garante total interoperabilidade, qualidade, segurança baseada em criptografia para os protocolos de rede IPv4 e IPv6 e que define um conjunto de serviços de segurança, incluindo: controle de acesso, integridade, autenticação, proteção contra *replays*, confidencialidade (criptografia) e limite de fluxo de tráfego, oferecendo proteção à camada de rede e às camadas superiores (KENT; ATKINSON, 1998a). A sua especificação define um conjunto de protocolos que dá suporte aos requisitos de segurança exigidos por uma VPN IP. Como é uma função da camada 3, ele não pode fornecer serviços para outros protocolos da mesma camada, como o IPX e SNA. Portanto, o IPSec oferece os meios necessários para garantir a confidencialidade, integridade e autenticidade dos pacotes IPs transmitidos e recebidos. Ele trabalha com uma variedade de esquemas padronizados e processo de negociação de criptografia, bem como, com vários sistemas de segurança (assinatura digital, certificado digital, infra-estrutura de chave pública, etc).

Vale destacar também que:

> IPSec oferece estes serviços independentemente do algoritmo de criptografia usado, ou seja, IPSec possui uma arquitetura aberta no sentido de possibilitar a inclusão de outros algoritmos de autenticação e criptografia (SILVA, 2003).

Essa afirmação evidencia que o IPSec é um padrão que não está vinculado a protocolos de criptografia específicos, portanto, a sua estrutura sobrevive mesmo a violações em algoritmos de criptografia, pois, caso esta situação ocorra, basta alterar o algoritmo

utilizado pelo IPSec. Esta característica do IPSec é também descrita em (TANENBAUM, 2003).

Outro aspecto importante a ser ressaltado sobre o IPSec é que, embora trabalhe na camada de rede IP (não orientada a conexão), o IPSec é orientado a conexão, pois necessita estabelecer uma conexão para o estabelecimento de uma chave que será utilizada durante um certo período de tempo. No contexto do IPSec, uma conexão é chamada de uma **Associação de Segurança**. Os conceitos e funcionamento de uma Associação de Segurança serão descritos na seção 4.5.2.

4.5.1 Conjunto de Transformação

Um conceito importante no IPSec é o chamado ***Conjunto de Transformação*** que é uma lista de protocolos e de algoritmos de criptografia que um dispositivo IPSec pode aceitar, pois o IPSec permite o uso de diferentes algoritmos e protocolos, sendo necessário que um dispositivo IPSec, que pode ser um *host*, um *gateway* ou um roteador, declare e negocie com outro dispositivo IPSec o que este pode suportar. Desta forma, para que dois dispositivos IPSec possam se comunicar, eles necessariamente precisam compartilhar o mesmo conjunto de transformação.

Um conjunto de transformação típico do IPSec contém as seguintes informações:

- **O Protocolo de Segurança IPsec** a ser utilizado (AH ou ESP).
- **O Algoritmo de Criptografia** a ser suportado pelos dois dispositivos IPsec.
- **O Algoritmo de Autenticação** a ser suportado pelos dispositivos.

Como mencionado anteriormente, os serviços de segurança do IPSec são oferecidos por meio de dois protocolos de segurança que fazem parte da arquitetura básica do IPSec. O primeiro é o *Authentication Header* - AH (KENT; ATKINSON, 1998b), ou Cabeçalho de Autenticação, que fornece integridade e autenticidade para os pacotes, mas não disponibiliza privacidade. O outro protocolo possível é o *Encapsulating Security Payload* - ESP (KENT; ATKINSON, 1998c), ou Encapsulamento Seguro de Dados**,** que fornece privacidade, autenticidade e integridade.

Em 1998, ano em que foram definidos esses protocolos, estabeleceu-se que, para melhor garantia de interoperabilidade, todas as implementações do IPSec deveriam suportar obrigatoriamente o DES e o 3-DES como algoritmos de criptografia e os algoritmos HMAC, MD5 e SHA-1 para autenticação (SILVA, 2003). Atualmente, a lista de protocolos de criptografia e autenticação encontra-se bem maior, alguns fornecedores de produtos IPSec já implementam outros algoritmos de criptografia, como por exemplo, o *Advanced Encryption Standard* (AES – Rijndael), *Serpent*, *twofish*, IDEA, entre outros.

Os serviços de segurança que são permitidos no IPSec utilizam o AH ou o ESP, mas não ambos (STALLINGS, 1999a). A tabela a seguir identifica quais os serviços de segurança que são suportados por cada um desses protocolos, sendo que o ESP pode ter ou não serviço de autenticação.

Tabela 2 – Serviços IPSec
Fonte: (STALLINGS, 1999a)

	AH	ESP (somente com criptografia)	ESP (criptografia + autenticação)
Controle de Acesso	V	V	V
Integridade	V		V
Autenticação	V		V
Rejeição de pacotes *replay*	V	V	V
Confidencialidade		V	V
Confidencialidade em fluxo de tráfego limitado		V	V

A seguir serão detalhados cada elemento do IPSec, inclusive os protocolos AH e ESP.

4.5.2 Associações de segurança

O IPSec define o conceito de *Security Associations* – SA (Associação de Segurança). Esta Associação de segurança é basicamente um acordo sobre como as informações serão transmitidas com segurança entre duas entidades na rede, é, no sentido mais amplo, uma negociação de um “contrato de segurança” (KARA, 2001).

Podemos também definir uma Associação de Segurança como sendo uma conexão lógica que protege o fluxo de dados de um dispositivo IPSec a outro usando um *Conjunto de Transformação*.

Como já mencionado anteriormente, uma das características do IPSec é a abertura de seu padrão para dar suporte a vários protocolos e modos de comunicação, assim como, a vários algoritmos de criptografia. Portanto, todos esses detalhes precisam ser previamente negociados, antes que se inicie a troca segura dos dados. Este acordo resultante é uma SA (NORTHCUTT, 2003).

Um fato importante a ser ressaltado é que uma SA é uma conexão *simplex* (unidirecional) entre dois pontos extremos e que gera um identificador de segurança único associado a cada conexão, por conseguinte, havendo necessidade de um tráfego seguro em ambos os sentidos de uma comunicação entre duas entidades, é necessário que se estabeleça duas associações de segurança (TANENBAUM, 2003).

As SAs são identificadas univocamente por três parâmetros:

- ***Security Parameter Index – SPI*** (Índice de Parâmetro de Segurança): é uma *string* de *bits* que é carregada nos *headers* do AH e ESP a fim de possibilitar que o destinatário das mensagens possa identificar qual é a associação de segurança relacionada a cada pacote recebido. O SPI é um campo que surge nos cabeçalhos de segurança IPv6 (AH e ESP), que não é criptografado na transmissão, já que a sua informação é essencial para decifrar a informação transmitida.

- ***Identificador do Protocolo de Segurança***: indica qual o protocolo de segurança que a Associação de Segurança utiliza, se o AH ou o ESP.

- ***Endereço IP destino***: identifica o endereço IP do ponto final (destino) da Associação de Segurança, podendo ser um *host*, um *firewall* ou um roteador.

Quando uma entidade quiser estabelecer uma Associação de Segurança, ela utiliza um SPI, um protocolo de segurança e um endereço destino (pertencente à entidade com que deseja estabelecer comunicação segura) e envia essa informação à entidade com aquela que quer estabelecer o canal seguro. Assim, para cada sessão de comunicação autenticada entre

dois *hosts*, são necessários dois SPI - um para cada sentido, dado que cada Associação de Segurança é unidirecional (*simplex*).

A Associação de Segurança permite negociar protocolos, algoritmos de criptografia e chaves a serem utilizadas e contém informações sobre:

- A identificação do *host* remoto participante do IPSec (um endereço IP ou o nome do *host*).

- O protocolo de segurança (AH ou ESP), o algoritmo de criptografia (se o ESP for utilizado). Estas informações são negociadas quando se estabelece o conjunto de transformação.

- As chaves compartilhadas usadas no algoritmo de criptografia durante a SA.

- A descrição do fluxo de tráfego protegido através da SA. Especifica o endereço IP e os números das portas (*sockets*) protegidas pela SA.

- Um número único que identifica uma SA (o SPI).

- Temporizadores e Contadores que estabelecem o tempo de vida de uma SA.

- Números de Sequenciamento para detectar possíveis ataques (aqueles que reenviam pacotes já utilizados por exemplo - *replay attaks*).

4.5.3 Modos de Transporte e de Túnel

O IPSec define dois modos de uma Associação de Segurança - SA. Define os **modos de transporte** e o de **Túnel**. No ***modo de transporte***, a carga do pacote IP é protegida pelo IPSec. O *Header* (Cabeçalho) do pacote IP original é deixado intacto, sendo adicionado a um *Header* de IPsec após o Cabeçalho do pacote IP original. Isto pode ser visualizado na figura 4.7 a seguir.

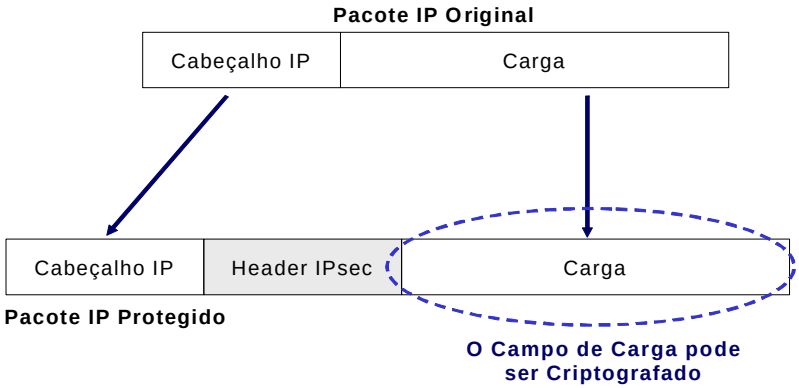


Figura 4.7 - Modo de Transporte SA

No modo de transporte, o tráfego transmitido entre dois *dispositivos* IPSec, uma estação cliente e um servidor por exemplo, pode ser observado, mesmo com o uso de criptografia, pois os endereços origem e destino (contidos no *header*) são deixados intactos. Assim, utilizando-se este modo, um invasor poderia facilmente obter os endereços IP de servidores e clientes de uma comunicação, e com isso, utilizar posteriormente estas informações para atacar servidores de rede, por exemplo.

Perlman e Kaufman, em (PERMAN; KAUFMAN, 2001), destacam que o IPSec é implementado na camada de redes (camada 3), portanto, tudo acima da camada 3 é considerado dados, incluindo o cabeçalho do IP. Desta forma, quando nos referirmos ou exemplificarmos, neste trabalho, aos dados a serem criptografados de um pacote IP, ou simplesmente a Carga, ou *payload*, do pacote IP, estamos incluindo os *headers* e *payloads* dos protocolos das camadas acima da camada de redes.

O ***modo de túnel*** é uma SA entre dois roteadores (ou gateways de segurança como são chamados) ou um *host* e um roteador. Neste modo, todo o pacote IP é protegido e torna-se

uma carga de um novo pacote. Um cabeçalho IPSec é inserido após o cabeçalho de IP do novo pacote. Os endereços de origem e destino no cabeçalho do novo pacote IP são os endereços dos dispositivos que realizam a Associação de Segurança. Na figura 4.8, pode-se visualizar este processo.

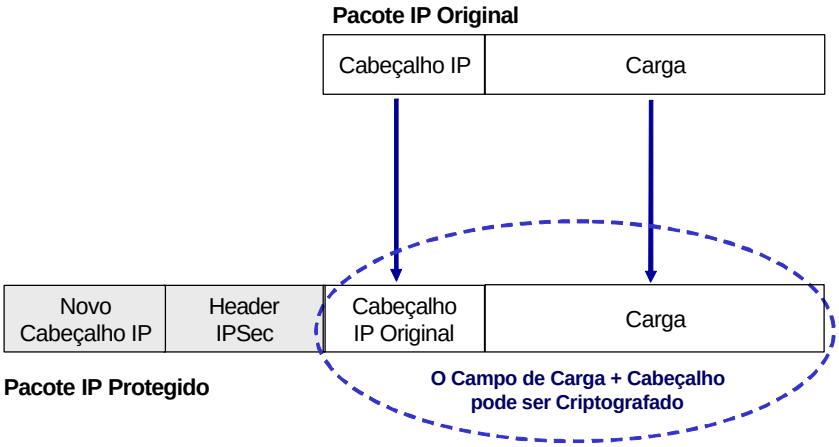


Figura 4.8 - Modo de Túnel SA

O modo de Túnel (ou tunelamento) é a base para implementação das redes VPN dedicadas entre dois roteadores ou a base para conexões VPN entre usuários em *hosts* remotos a um roteador dentro de uma *Intranet*.

Outra questão importante da utilização do modo de Túnel é que neste modo não há necessidade de se equipar todas as estações e servidores com o IPSec; ao invés disso, pode-se ativar o IPSec apenas nos roteadores ou nos *gateways*.

Como será visto posteriormente, tanto o modo de transporte como o modo túnel podem ser encapsulados em cabeçalhos ESP ou AH.

4.5.4 AH (IP Authentication Header)

O Protocolo AH, descrito pela RFC 2402, provê integridade e autenticação aos datagramas IP, e ainda provê proteção contra ataques do tipo *replays*, ou seja, quando uma pessoa mal intencionada captura pacotes válidos e autenticados pertencentes a uma conexão, replica-os e os reenvia, como se fosse a entidade que iniciou a conexão. Este serviço é opcional, que pode ser selecionado pela entidade recebedora dos pacotes quando uma SA é

estabelecida, embora, que por *default* haja um sequenciamento dos pacotes feitos pela entidade que originou os pacotes, este serviço é efetivamente aplicado quando a recebedora checa os números de sequência dos pacotes. O AH também previne ataques do tipo *Spoofing*, ou seja, quando o invasor assume o papel de uma entidade confiável para o destino.

A característica de integridade de dados do AH garante que modificações não detectadas no conteúdo dos pacotes transmitidos não são possíveis. A autenticação permite que um sistema final ou um dispositivo de rede possa autenticar o usuário ou a aplicação, filtrando desta forma o tráfego transmitido.

Porém, vale destacar que:

> Embora a autenticação aconteça no pacote IP, nem todos os campos podem ser autenticados, porque alguns campos do cabeçalho serão alterados no decorrer da transmissão. Esses campos são considerados mutantes, ou variáveis, sendo eles: Tipo do Serviço, Offset, Flags, Tempo de Vida do Pacote e Checksun (SILVA, 2003).

Este processo de integridade e autenticação emprega algoritmos *hashing* que calculam um único valor quando uma mensagem é fornecida como entrada da função *hash*, a qual utiliza a chave estabelecida na Associação de Segurança. Este valor único é chamado ***valor hash*** ou ***código de autenticação***. O mecanismo deve garantir que é praticamente impossível, ou inviável, encontrar uma outra mensagem que gere o mesmo código. Desta forma, quando algum *host* envia uma mensagem, ele associa a esta mensagem o seu código de autenticação, que é o resultado desta função *hash* na origem, e o coloca no cabeçalho do protocolo AH, que será validado na entidade recebedora do pacote através de um novo cálculo do *hash*. Caso o valor do novo *hash* calculado (na entidade recebedora) seja igual ao *hash* enviado no cabeçalho, a autenticação é bem-sucedida, caso contrário, o pacote não é autenticado, sendo descartado.

Os Algoritmos *Hashing* mais populares implementados no IPSec são o **MD5** (*Message Digest 5*) e o ***Secure Hash Algorithm*** (**SHA-1**), desenvolvido pelo **NIST** (*National Institute os Standards and Technology*), **NSA** (*National Security Agency*) e o órgão de defesa oficial dos Estados Unidos. O uso de chave pública com um algoritmo de *hashing* é chamado ***Key-hashing for Message Authentication***, ou simplesmente, **HMAC** (o **H** significa *Hashing* e o **MAC** significa *Message Authentication Code*).

Assim, o AH provê integridade e autenticação, usando algoritmos de *hashing* com chaves públicas como o HMAC MD5 e o HMAC SHA-1, sendo que o AH não provê confidencialidade, isto é, criptografia.

O Cabeçalho de Autenticação consiste de seis campos (figura 4.9):

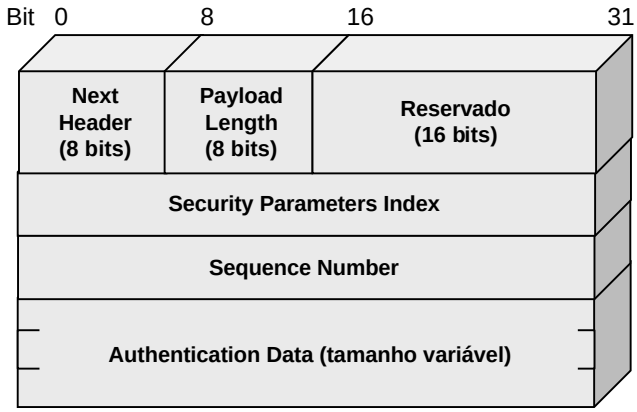


Figura 4.9 – Formato do Protocolo AH

- ***Next Header*** (Próximo Cabeçalho): é um campo de 8 bits que identifica o tipo do próximo *payload* após o *Authentication Header*, ou seja, é o identificador do protocolo do próximo cabeçalho. É o mesmo valor atribuído ao campo Protocolo no cabeçalho IP original.

- ***Payload Length*** (Tamanho do "dado"): é o tamanho do cabeçalho de autenticação em palavras de 32 bits menos 2. Por exemplo, o tamanho *default* do campo *Authentication Data* é 96 bits, ou 3 palavras de 32 bits. Com as 3 palavras de 32 bits da parte fixa deste cabeçalho, o tamanho total é de 6 palavras de 32 bits, desta forma o *Payload Length* será 4 (6 -2).

- ***Reservado***: Um campo de 16 bits reservado para uso futuro.

- ***Security Parameters Index*** (Índice de Parâmetro de Segurança): identifica a Associação de Segurança para um determinado pacote. A utilização deste campo previne ataques do tipo reprodução (*replay*).

- ***Sequence Number*** (Número de Sequência): contador que identifica os pacotes pertencentes a uma determinada SA. A sua utilização ajuda na prevenção de ataques *replay*.

- ***Authentication Data*** (Dados de Autenticação): é um campo de tamanho variável que contém o *Integrity Check Value - ICV* para este pacote, que é calculado seguindo o algoritmo de autenticação usado, definido pela SA.

O Protocolo AH pode operar de duas formas: modo de transporte ou modo túnel. No modo transporte, o AH é inserido após o cabeçalho IP e antes do protocolo de mais alto nível, como por exemplo, o TCP, UDP, ICMP, etc (KENT; ATKINSON, 1998b). Esse modo utiliza o mesmo cabeçalho original do pacote, trocando-se somente o campo Protocolo do cabeçalho IP, que recebe o valor 51 que representa o protocolo AH. O valor original do protocolo é então posto no cabeçalho de autenticação. É evidente que neste modo, como o cabeçalho original é utilizado e, portanto, os endereços IP do *host* origem e do destino são também conhecidos, os *hosts* ou dispositivos de origem e de destino devem obrigatoriamente utilizar o modo transporte do protocolo AH.

No modo túnel, é gerado um novo cabeçalho, contendo o cabeçalho original encapsulado, seguido pelo cabeçalho de autenticação. O pacote IP original fica intacto, sendo encapsulado dentro de um novo pacote. O endereço IP de origem e de destino, do novo pacote IP que encapsula o original, passa a ser o endereço dos dispositivos IPSec, portanto, este modo pode ser usado entre entidades SA ou *gateways* VPN (SILVA, 2003).

A figura 4.10 abaixo ilustra os modos de transporte e túnel para o protocolo AH no contexto do IPv4.

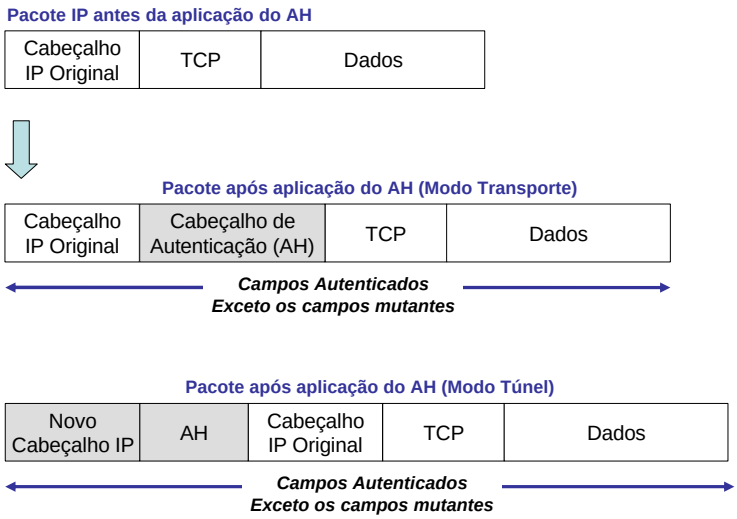


Figura 4.10 - Modo Transporte e Túnel no protocolo AH no IPv4

4.5.5 ESP (IP Encapsulating Security Payload)

O Protocolo de Encapsulamento Seguro de Dados, ou ESP, descrito pela RFC 2406, provê confidencialidade por meio de criptografia, autenticação, integridade, proteção de *replay* e Confidencialidade limitada ao fluxo de tráfego. Assim como no AH, o algoritmo de criptografia utilizado é configurado na SA, na qual os pacotes são enviados. Esse conjunto de serviços depende das opções selecionadas em tempo de estabelecimento de Associação de Segurança. Vale salientar que a confidencialidade pode ser selecionada independentemente de todos os outros serviços, entretanto, o uso de confidencialidade sem a Integridade e Autenticação, por exemplo, fragiliza o tráfego para certos tipos de ataques. Isso é possível se nenhum algoritmo de criptografia for utilizado. Por outro lado, o serviço de *anti-replay*, por exemplo, somente pode ser selecionado se o serviço de autenticação for acionado (KENT; ATKINSON, 1998c).

Cabe observar que como o pacote IP é um datagrama, e portanto não possui garantia de entrega, cada pacote deve conter informações necessárias para estabelecer o sincronismo da Criptografia, permitindo, desta forma, que a descriptografia ocorra no destino sem problemas. Porém, se nenhum algoritmo de criptografia for utilizado, o ESP poderá oferecer somente autenticação (SILVA, 2003).

A figura 4.11 abaixo apresenta o Formato do pacote ESP.

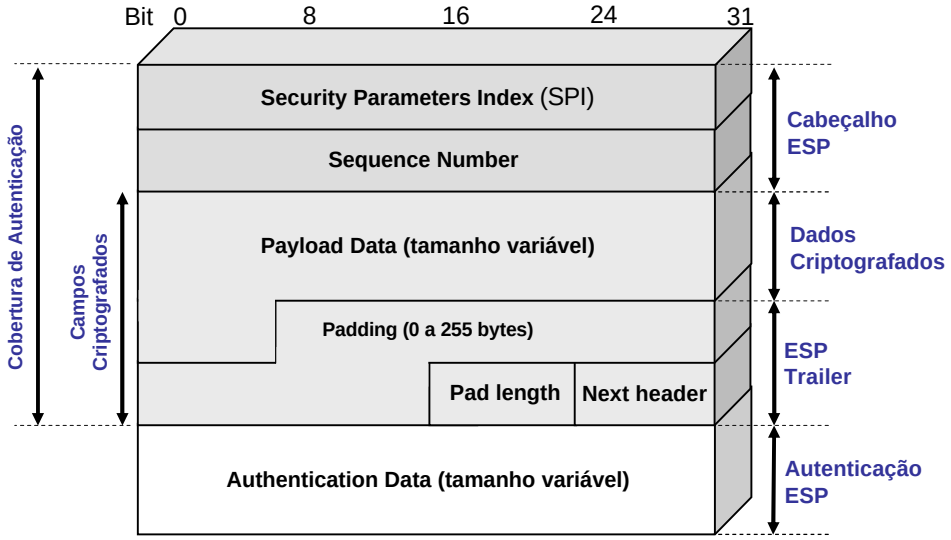


Figura 4.11 – Formato do pacote ESP

- ***Security Parameters Index*** (Índice de Parâmetro de Segurança): campo de 32 bits que identifica a Associação de Segurança.

- ***Sequence Number*** (Número de Sequência): é um contador de 32 bits que identifica os pacotes pertencentes a uma determinada SA. A sua utilização ajuda na prevenção de ataques *replay*.

- ***Payload Data*** ("dados"): este campo corresponde aos dados criptografados. É o segmento do nível de transporte (modo de transporte) ou o pacote IP (modo túnel) que está protegido pela criptografia. Este é um campo variável.

- ***Padding*** (Complemento de 0 a 255 *bytes*): o propósito deste campo é completar o bloco de texto para manter a compatibilidade com o algoritmo de criptografia a ser utilizado (colocando-o múltiplo de um determinado número de *bytes*, geralmente 4 *bytes*) e para suportar confidencialidade limitada ao fluxo de tráfego.

- ***Pad length*** (tamanho do Complemento de 8 bits): determina o número de *bytes* do complemento, ou *Pad*, imediatamente precedentes a este campo.

- ***Next Header*** (8 bits): identifica o tipo dos dados contidos no campo *Payload Data*, indicando o primeiro *header* neste *Payload*.

- ***Authentication Data*** (Dados de Autenticação): é um campo de tamanho variável que contém o ICV (*Integrity Check Value*) para este pacote, que é calculado sobre o pacote ESP menos o campo de Autenticação de Dados.

Como pode ser observado através da figura 4.11, os campos *Payload Data*, *Padding*, *Pad Length* e *Next Header* podem ser criptografados pelo serviço ESP. O campo *Payload Data* contém o ESP, porém, quando é utilizado algoritmo de criptografia, este campo pode conter estruturas necessárias para o sincronismo da criptografia, como é o caso dos Vetores de Inicialização, ou *Initialization Vector*, quando utilizados no algoritmo DES no modo de encadeamento de bloco de cifra. O Vetor de Inicialização corresponde aos 8 *bytes* iniciais da Chave Simétrica, que serve para evitar que duas mensagens iguais sejam mapeadas para o

mesmo bloco criptografado. Este vetor será armazenado no início do Campo *Payload Data* (SILVA, 2003).

O Protocolo ESP também pode operar de duas formas: modo de transporte ou modo túnel. O modo transporte é utilizado para criptografar e opcionalmente autenticar os dados carregados pelo IP (STALLINGS, 1999a). Neste modo, o cabeçalho ESP é inserido entre o cabeçalho IP e os Dados (*Payload*), ou melhor, imediatamente antes do cabeçalho do protocolo da camada de transporte, adicionando o *trailer* ESP. Se a autenticação for selecionada, um campo de Autenticação é adicionado após o trailer ESP. Se o pacote original já tiver um cabeçalho IPSec, este novo cabeçalho é colocado antes do cabeçalho IPSec já construído. A figura 4.2 apresenta a aplicação do ESP no modo transporte.

Já o modo túnel é utilizado para criptografar todo o pacote IP, sendo colocado todo o pacote original dentro de um novo pacote IP. Neste modo, se o túnel for entre dois *hosts*, os endereços de origem e destino do novo cabeçalho serão os mesmos do cabeçalho criptografado. Se o túnel for entre *gateways*, os endereços de origem e destino do novo cabeçalho serão dos *gateways* e os endereços dentro do cabeçalho criptografado serão dos *hosts* ou servidores atrás dos *gateways* (KENT; ATKINSON, 1998c). A figura 4.12 abaixo ilustra estes dois modos para o protocolo ESP.

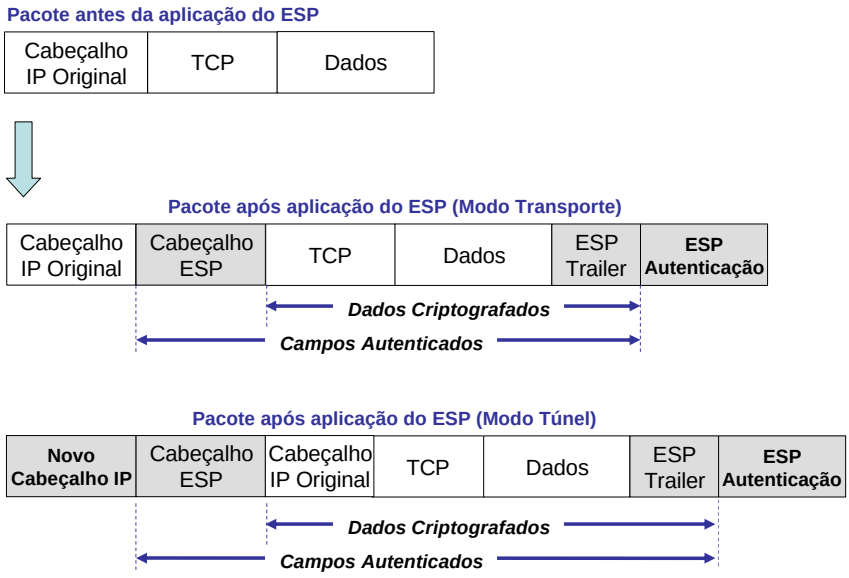


Figura 4.12 - Modo Transporte e Túnel no protocolo ESP

Como pode ser observado na figura 4.12, o modo túnel do protocolo ESP possibilita a criptografia e a autenticação de todo o pacote original, ou seja, garante confidencialidade e autenticidade ao pacote original.

Fica claro que o modo túnel é aplicável em configurações que incluem um *firewall*, ou um outro tipo de *gateway* de segurança, o qual protege uma rede privada (confiável) de redes externas, e neste caso, a criptografia ocorre somente entre um *host* externo e o *gateway* de segurança ou entre dois *gateways* de segurança. Isto simplifica a tarefa de distribuição de chaves, pois há uma redução no número de chaves necessárias para o estabelecimento de conexões seguras (STALLINGS, 1999a).

Como vimos, o ESP provê confidencialidade e, opcionalmente, integridade e autenticação. O ESP suporta Algoritmos como o DES e o triple-DES e também suporta algoritmos de integridade como o HMAC MD5 e o HMAC SHA-1.

Como o ESP pode fazer tudo que o AH faz, pode-se pensar que o mesmo o substitui integralmente, porém, há uma leve diferença, o ESP checa integridade apenas da carga do pacote IPSec e o AH checa integridade do pacote inteiro (carga + cabeçalho IP). Desta forma, para aplicações críticas, que requeiram alto nível de segurança, faz-se necessário utilizar o AH e o ESP em conjunto.

Uma diferença, bastante sutil, entre as funcionalidades do protocolo AH e do ESP, citada por Perlman e Kaufman em (PERLMAN; KAUFMAN, 2001), é que quando utilizamos o protocolo AH, os roteadores e *firewalls* sabem perfeitamente que os pacotes não estão criptografados, pois o AH não utiliza criptografia, e portanto, podem tomar decisões baseadas nos campos do *header* da camada 4, como por exemplo, as portas TCP utilizadas. É importante salientar que isto só é possível quando o tráfego IP não é criptografado.

4.5.6 Bancos de Dados de Segurança

O IPSec utiliza dois bancos de dados, o **banco de dados de Políticas de Segurança** (*Security Policy Database* – **SPD**), onde são definidos o conjunto de políticas de segurança para todos os tipos de tráfego IP de entrada e saída, independentemente da SA, e o **banco de dados de Associação de Segurança** (*Security Association Database* – **SAD**), o qual contém o conjunto de parâmetros associados à SA. Um exemplo típico de política contida no SPD pode

ser a definição da Empresa de qual o algoritmo de criptografia pode ser utilizado para conexões IPSec, mas não a chave de criptografia.

As informações armazenadas no SAD são:

- OS SPIs que identificam cada SA dentro do SAD.
- O Protocolo de Segurança usado na SA (ESP ou AH).
- O modo de tunelamento que a SA está operando (túnel ou transporte).
- Contador seqüencial do pacote IP dentro da SA.
- Endereço IP de Origem e de Destino da SA.
- O Algoritmo e a chave de autenticação.
- O Algoritmo e a chave de criptografia.
- Tempo de vida das chaves de criptografia e autenticação.
- Tempo de vida da SA.

Quando um pacote IP chega em um dispositivo IPSec, a SA é localizada através de três informações: o endereço IP de destino, o tipo de protocolo (AH ou ESP) e o Índice de parâmetro de segurança (SPI), sendo que o endereço IP do destino e o tipo de protocolo são obtidos no cabeçalho IP e o SPI vem no cabeçalho AH ou ESP. Desta forma, caso uma SA seja localizada, o pacote é processado de acordo com o serviço de segurança especificado. Posteriormente, o SPD é processado para avaliar se o pacote atende às Políticas de Segurança especificadas no IPSec (SILVA, 2003).

Quando um pacote IP está saindo, o SPD é processado primeiro; se o pacote atender às políticas de segurança especificadas no IPSec, verifica-se a existência, ou não, de uma SA já estabelecida. Caso a SA já exista, o pacote é processado de acordo com a SAD. Se a SA não for localizada, uma nova SA é negociada para o pacote e as novas informações da SA são armazenadas na SAD.

4.5.7 Avaliação do IPSec - Vantagens e Desvantagens

O protocolo IPSec pode proteger qualquer protocolo que rode sobre o IP e qualquer meio físico onde o IP possa rodar (FREES/WAN, 2003), podendo proteger uma grande variedade de aplicações e protocolos rodando sobre uma infra-estrutura física complexa, e esta é a realidade da Internet. Desta forma, O IPSec é considerado uma solução de serviços de segurança bastante generalista, a qual não gera impactos visíveis aos usuários. Ao contrário do IPSec, por exemplo, para se utilizar o PGP para criptografia e assinaturas digitais nos e-mails, o usuário deve no mínimo: lembrar-se da sua frase geradora da chave (*passphrase*), mantê-la em segurança e seguir os procedimentos para validar as chaves correspondentes (FREES/WAN, 2003).

Desta forma, a grande vantagem desse padrão é o aproveitamento da infra-estrutura de rede IP já existente. O IPSec possui várias vantagens sobre outros protocolos mais tradicionais que também implementam VPN no nível de enlace (nível 2), como é o caso do protocolo PPTP, L2TP e o L2F (Layer 2 *Forwarding*), criado pela *Cisco Systems*, que implementa um túnel que encapsula os pacotes IP trocados entre dois *hosts* no nível de enlace. Em resumo, os maiores benefícios do IPSec são:

- ***Transparente a Subrede***: será necessário implementar os serviços do IPsec somente nos *sites* origem e de destino da conversação, não sendo necessário que a *subrede* implemente o IPSec. Isto difere do L2F, por exemplo, que requer dispositivos de criptografia em todos os nós da *subrede*, e do MPLS que exige que os roteadores da subrede sejam LSRs.

- ***Simplicidade***: com o uso de IPSec, não há necessidade de se configurar as estações ou aplicações para se trabalhar com o IPSec. Com o processo de criptografia no nível de aplicação, cada programa pode possuir a sua própria implementação de segurança.

- ***Flexibilidade***: O IPSec é bastante flexível, com ele pode-se proteger determinado tipo de tráfego e deixar que outro tipo de tráfego circule sem a ação dos serviços de segurança implementada no IPSec.

- ***Fácil manutenção***: a aplicação do IPSec em determinados pontos da rede pode facilitar o esforço e a manutenção de uma política de segurança consistente dentro da Empresa.

- ***Fácil implementação***: uma vez que os serviços do IPSec podem ser implementados nos roteadores ou nos *gateways*, os clientes e outros servidores da rede podem permanecer com o mesmo nível de manutenção que tinham antes da implementação dos mecanismos de segurança através do IPsec.

- ***Gerenciamento Manual e Automático de Chaves***: o IPSec permite o gerenciamento manual e o automático de chaves. Embora a IPSec não integre um mecanismo de gerenciamento automático de chaves, a IETF definiu como norma de gerenciamento de chaves o protocolo híbrido ISAKMP/Oakley (RFC 2408/RFC 2412), também denominado ***Internet Key Exchange* – IKE** (RFC 2409). Este tipo de gerenciamento será posteriormente discutido neste trabalho.

É importante também ressaltar que o IPSec possui algumas limitações. De acordo com (FREES/WAN, 2003), as principais limitações desse padrão são:

- **O IPSec não pode ser seguro se o Sistema não for**. Isto significa que garantir a segurança nas máquinas que implementam os *gateways* IPSec é um requerimento essencial.

- **O IPSec não é fim-a-fim**. Desta forma, as informações no próprio *host*, ou em um determinado segmento interno da rede, podem não estar protegidas (criptografadas), permitindo violações oriundas de um invasor da própria Empresa.

- **O IPsec não pode fazer tudo**. Ele não provê toda a funcionalidade dos sistemas de segurança da camada de aplicação. Entretanto, os ataques a protocolos da camada de aplicação tornam-se mais difíceis. Havendo, por exemplo, uma necessidade posterior de comprovação de autenticidade de um documento através da assinatura digital de um determinado usuário, deverá se

ter obrigatoriamente a assinatura digital e a chave pública desse usuário para essa verificação.

- **O IPsec autentica máquinas, mas não autentica Usuários**. Por exemplo, caso haja necessidade de se controlar quais usuários acessaram um Servidor de Banco de Dados, deve-se utilizar um mecanismo não-IPSEc. Obviamente, o IPSec pode garantir que a comunicação entre as máquinas transcorreu de forma segura e quais máquinas conectaram ao Servidor, mas isto é tudo.

- **O IPsec não previne ataques DoS (*Denial of Service*).** Esses ataques podem causar o travamento de um sistema ou uma sobrecarga, que pode comprometer demasiadamente as operações legais em um sistema, gerando inclusive a negação de serviços a usuários legítimos. Para ter proteção contra este tipo de ataque, é necessário a utilização de um *firewall*, que poderá ser utilizado juntamente com o IPSec.

- **IPsec não evita a análise do tráfego de rede**. Mesmo utilizando IPSec, alguns campos não criptografados dos *Headers*, como por exemplo, os endereços destino e origem dos *gateways* e o tamanho do pacote, podem ser monitorados.

Entretanto, os bons resultados apresentados ao se utilizar esta tecnologia, fazem dela uma grande arma contra vários tipos de ataques e invasões, pois se for utilizada de forma correta, ou seja, combinar bem os recursos de protocolos como o AH e o ESP, ou saber quando tirar os melhores proveitos do modo Transporte e do modo Túnel, ou ainda, de utilizar Certificados Digitais ou o de compartilhar seguramente as chaves secretas através do protocolo IKE, o qual será discutido posteriormente, dificilmente um invasor descobrirá algum ponto falho na conexão.

4.6 Gerenciamento de Chaves

Como os serviços de autenticação, integridade e criptografia do IPSec utilizam chaves secretas, fazem-se necessários mecanismos eficientes para realizar o gerenciamento de chaves, com suporte para distribuição manual ou automática das chaves. Esses mecanismos dizem respeito basicamente à criação, eliminação e alteração das chaves.

Embora a IPSec não possua um mecanismo de gestão de chaves, a IETF definiu como norma de gerenciamento de chaves o protocolo híbrido ***Internet Key Exchange* – IKE**, através da RFC 2409 (HARKINS; CARREL, 1998), o qual realiza a negociação dinâmica de um SA. Ele é um protocolo híbrido porque se originou da associação e implementação de três protocolos, a saber:

- **ISAKMP** (*Internet Security Association and Key Management Protocol*): descrito na RFC 2408 (MAUGHAN et al., 1998), define o método de distribuição de chave, dando suporte à negociação de protocolos de segurança em todos os níveis da pilha de rede.

- **OAKLEY:** descrito na RFC 2412 (ORMAN, 1998), define métodos para autenticar e estabelecer a troca de chaves. Define como as chaves são determinadas.

- **SKEME** (*Secure Key Exchange Mechanism*): que também define métodos para autenticar e estabelecer a troca de chaves.

Vale destacar:

> "O ISAKMP define como duas entidades instituirão um canal de comunicação seguro entre elas, fazendo com que os participantes se autentiquem entre eles, trocando informações de chaves e negociando serviços de segurança. Entretanto, não especifica como a autenticação é feita ou quais as chaves serão geradas, ou seja, é definido um caminho seguro ..." (SILVA, 2003).

Desta forma, torna-se evidente que o protocolo ISAKMP necessita de um mecanismo que defina o processo de autenticação e a troca de chaves. Esses requerimentos são atendidos plenamente pelo protocolo OAKLEY ou o SKEME.

A negociação dinâmica de uma SA é importante porque não se sabe exatamente quando será necessário se negociar uma SA para o estabelecimento de túneis VPN, também porque é aconselhável a troca periódica de uma SA para se fortalecer o canal contra ataques.

Doravante, este trabalho fará referência somente ao protocolo IKE, quando se referir a toda funcionalidade de gerenciamento de chaves definida pelo IETF, englobando tanto o protocolo ISAKMP, como o protocolo OAKLEY.

O IKE provê o IPsec com os seguintes serviços:

- Estabelece Associações de Segurança de forma dinâmica. Sem o IKE, deve-se configurar manualmente todos os requerimentos de SA entre dispositivos IPSec.

- Provê mudança dinâmica das chaves. Com o IKE, chaves expiradas após um determinado período são trocadas por novas chaves de maneira dinâmica.

- Possui proteção contra o ataque *Anti-Replay*.

- Possibilita Autenticação da Origem através de Certificados Digitais e Servidores de Autenticação (CA).

O IKE usa a porta 500 do protocolo UDP e interage com os restantes mecanismos de segurança IPSec através de associações de segurança. Assim, o IKE proporciona a possibilidade de estabelecer associações de segurança para diversos protocolos e aplicações de segurança, tendo assim um método transparente e aberto de associar diferentes mecanismos de segurança, sem envolver as entidades participantes na comunicação.

É importante ressaltar que quando dois dispositivos IPSec necessitam se comunicar usando IPSec, eles primeiramente se autenticam, usando o IKE e assim estabelecem uma Associação de Segurança IKE, também chamada SA IKE (ou SA ISAKMP).

Na figura 4.13, podemos observar, através do modelo de referência, o contexto em que o IPSec e o IKE trabalham.

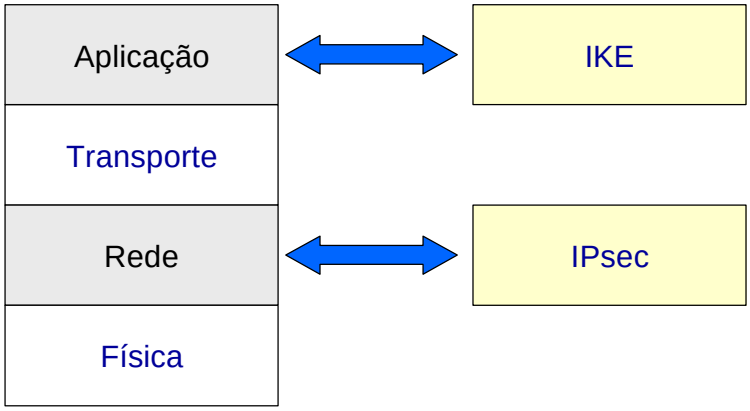


Figura 4.13 – Contexto do IKE e do IPsec

Como o objetivo é o de melhor entendimento desse processo de gerenciamento de chaves, é preciso tomar conhecimento de que o processo de negociação e efetivação da conexão é dividido em duas fases: Fase 1 e Fase 2 (PERLMAN; KAUFMAN, 2001).

A Fase 1 tem o seu início em meio inseguro. A proposta dessa fase é estabelecer um canal seguro para as negociações da Fase 2, isto é, proteger a comunicação futura entre as duas entidades ISAKMP, estabelecendo, desta forma, uma Associação de Segurança do ISAKMP[9], também denominada SA IKE (ou SA ISAKMP). Na Fase 2, é estabelecido uma Associação de Segurança para outros protocolos. Neste caso, é negociada uma SA IPSec, configurando-se, portanto, uma ligação entre as duas entidades que desejam a conexão.

4.6.1 Fase 1 do ISAKMP

Na Fase 1, além de serem negociados o algoritmo de criptografia, o algoritmo de *hash*, o método de autenticação e a informação pertinente ao grupo *Diffie-Hellman* que irá operar, também é definido de que maneira a informação será transmitida. Nesta fase, os métodos possíveis para que se estabeleça um canal são*: Modo Principal (Main Mode) e Modo Agressivo (Agressive* Mode) (PERLMAN; KAUFMAN, 2001). Alguns fornecedores de VPN implementam mais de um modo, chamado híbrido (*Hybrid Mode*), que une particularidades dos dois métodos (SILVA, 2003).

A negociação de chaves do **Modo Principal** ocorre em três etapas que promovem a proteção de identidade. Na primeira, o emissor envia várias mensagens ao receptor contendo

[9] Embora seja chamada de associação de segurança, esta não deve ser confundida com a SA do padrão IPSec. A SA do ISAKMP é bidirecional e não se aplica ao tráfego IPSec (SILVA, 2003).

propostas de SA, o receptor então escolhe uma delas e a envia ao emissor. Na segunda etapa, as entidades trocam os parâmetros de chaves e um valor randômico, chamado *nonces*, que é utilizado para prevenir ataques *replay*. Na terceira, e última etapa, todas as informações da SA IKE são trocadas, autenticadas por um dos seguintes métodos de autenticação: segredo compartilhado, assinatura digital ou criptografia de chave pública (SILVA, 2003). Uma função hash será utilizada nessa chave e seu resultado será trocado entre as duas entidades.

No modo agressivo, todas as três etapas descritas para o modo principal ocorrem em uma única mensagem entre o emissor e o receptor, porém, a informação de autenticação não é criptografada. O que se deve levar em consideração na hora de optar por um modo ou outro é a largura de banda, pois o Modo Principal gera maior *overhead*.

É importante salientar que o algoritmo *Diffie-Hellman* permite aos pares estabelecer chaves secretas compartilhadas em um canal de comunicação sem segurança. O Diffie-Helman é usado pelo IKE para estabelecer uma chave de sessão (WENSTROM, 2002, p. 508).

4.6.2 Fase 2 do ISAKMP

A segunda fase tem por objetivo negociar a SA IPSec, utilizando o canal estabelecido na Fase 1. Como o canal já está estabelecido, esta fase geralmente é mais rápida, desta forma, alguns autores a chamam de *Quick Mode* (Modo Rápido).

É aconselhável que a Fase 2 não utilize diretamente a chave definida na Fase 1, pois um invasor poderá se apoderar da chave utilizada na primeira fase da negociação, o que é extremamente difícil de acontecer segundo (SILVA, 2003), e descriptografar toda a comunicação da fase 2, derivando a chave SA IPSec. Para evitar esta situação, aconselha-se usar na Fase 2 uma nova chave que será derivada por meio do algoritmo *Diffe-Helman*, cuja técnica é chamada *Perfect Forward Secrecy* – PFS (disponível do FreeS/WAN). Obviamente, tal procedimento irá acarretar em queda de desempenho.

Outra característica a ser ressaltada dobre o protocolo ISAKMP é que ele possui um recurso bastante interessante e eficiente para reduzir as consequências dos ataques do tipo DoS, ataques de reprodução. Existe, em seu cabeçalho, dois campos de 8 *bytes* chamados *cookie do emissor* e *cookie do receptor* que realizam este papel. Estes campos são valores

gerados pelas entidades e estão associados aos endereços IP de cada entidade (PERLMAN; KAUFMAN, 2001). Uma entidade A realiza uma solicitação inicial a outra entidade, digamos entidade B, e esta retorna um *cookie* para entidade A. Este *cookie* ficará armazenado em B, associando o Endereço IP da entidade A ao *cookie* fornecido por B. Quando A enviar novamente uma mensagem para B, a entidade B verifica se o *cookie* é válido ou não, baseado no endereço IP de origem. A entidade recebedora só aceitará o pacote se o *cookie* for válido, caso contrário, o pacote é descartado. A figura 4.14 ilustra o formato da mensagem ISAKMP.

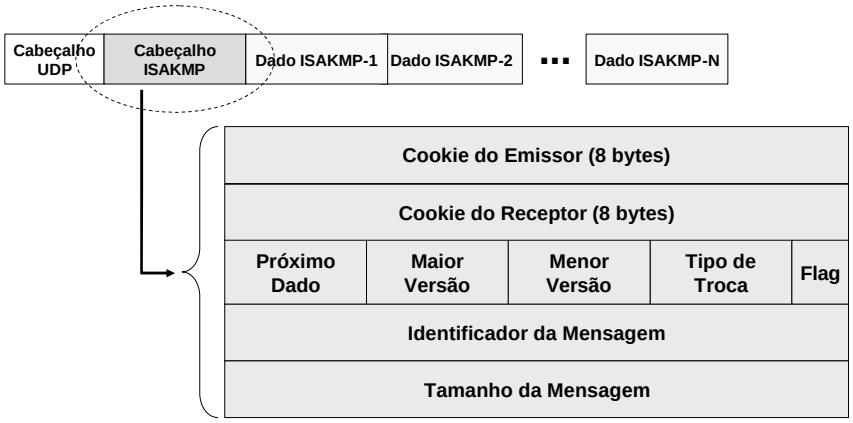


Figura 4.14 – formato da mensagem ISAKMP

5 ANÁLISE E PROPOSTA DE MODELO DE SEGURANÇA PARA REDES VPN

Neste capítulo, discutiremos diversas topologias para interligação de redes corporativas utilizando-se VPN. Iremos analisar cenários e topologias de redes, relacionando vantagens e desvantagens de cada uma. O enfoque principal será o posicionamento de servidores VPN em relação a outros serviços de segurança, como, por exemplo, o *firewall* e servidores PKI. Serão discutidos também aspectos de custos da solução proposta. Realizaremos uma análise comparativa entre os custos de implementação e manutenção de uma VPN com os custos de outras soluções de comunicação existentes, como por exemplo, a contratação de circuitos dedicados.

Obviamente, este trabalho não define uma arquitetura única de VPN que deve ser obrigatoriamente usada para qualquer tipo de corporação, até porque, os requerimentos de segurança e custo de cada cliente são diferentes. Desta forma, este trabalho se propõe a definir um modelo que possa servir como parâmetro ou referência para a implementação de redes VPN nas mais variadas situações.

5.1 Análise da Topologia para VPNs

Um aspecto muito importante quando se trabalha com redes VPN implementadas no nível de rede, especialmente as baseadas na Internet, é se estabelecer uma configuração de um *Gateway* VPN dentro de uma estrutura baseada no uso de *firewalls*, bem como, a localização

do *firewall* dentro de uma rede VPN. Esta preocupação em relação ao posicionamento de um *Gateway* VPN em relação ao *firewall*, foi bastante discutida em (KING, 1999), onde, além deste aspecto, foram analisados outros aspectos de segurança em uma rede VPN baseada na Internet como: autenticação, gerenciamento de chaves, tolerância a falhas, desempenho, confiabilidade, gerenciamento e interoperabilidade. No momento, restringiremos essa discussão somente aos aspectos de posicionamento.

Na análise de posicionamento, a questão então passa a ser a seguinte: a Empresa, após chegar à conclusão que ela deve realmente adotar uma solução VPN, observa suas próprias restrições e requerimentos e se depara com outro problema: onde colocar o *gateway* VPN dentro de sua topologia?

Dentro dessa visão, este trabalho comunga com algumas idéias e regras estabelecidas por Christopher King (1999) a respeito do posicionamento de um *Gateway* VPN em uma topologia de rede. Essas regras, que norteiam este trabalho, são as seguintes:

- O posicionamento de um servidor VPN não deverá comprometer a política de segurança da rede como um todo.

- O Servidor VPN não deverá estar em um ponto de falha.

- O Servidor VPN somente poderá aceitar tráfego de uma rede não-confiável se o tráfego for criptografado. Se a rede for confiável, o Servidor deverá aceitar tanto o tráfego criptografado, como o não-criptografado.

- O *Gateway* VPN deve defender-se de ameaças da Internet.

- Toda arquitetura deve filtrar o tráfego após a descriptografia.

Desta forma, o posicionamento de um *Gateway* VPN em relação ao *Firewall* da rede torna-se importantíssimo e crucial para o sucesso da topologia adotada, uma vez que os *firewalls* não podem controlar o acesso à rede de pacotes criptografados. Além disso, muitas vezes é necessário se lidar com cenários distintos e mais complexos de acesso VPN, por exemplo: conexão entre redes de Empresas distintas (*extranets*), acesso remoto para

funcionários ou para parceiros, onde não se sabe *a priori* qual o endereço IP do Cliente VPN, dificultando assim o processo de filtragem, e conexões de filiais da própria Empresa.

Existem várias formas diferentes de posicionamento de uma VPN em relação ao *firewall*, as principais são: em frente ao *firewall*, atrás do *firewall* ou na mesma máquina do *firewall*. A seguir detalharemos estas opções de posicionamento, mencionando vantagens e desvantagens de cada uma. E posteriormente, iremos sugerir uma topologia de posicionamento do VPN.

5.1.1 VPN em frente ao *firewall*

Um Servidor VPN postado na frente de um *firewall* fere totalmente as regras de posicionamento que foram sugeridas por King (1999), descritas na seção anterior. O *Gateway* VPN se estabelece em um ponto de fragilidade da rede, uma vez que fica sem a proteção adicional do *firewall*. Além disso, este posicionamento requer que as Empresas aceitem o tráfego não-criptografado e criptografado de uma rede não confiável. Desta forma, pode-se abrir uma possibilidade de não se saber se o Servidor VPN foi comprometido ou não.

Na figura 5.1, pode-se visualizar essa topologia. Tem-se um *firewall* após o terminador da VPN. Neste cenário, o gateway VPN cuidará da rede virtual, porém, não aplica as regras da filial aos pacotes do túnel VPN. Mais ainda, os pacotes oriundos da Internet serão analisados somente depois que o pacote entrar na rede interna.

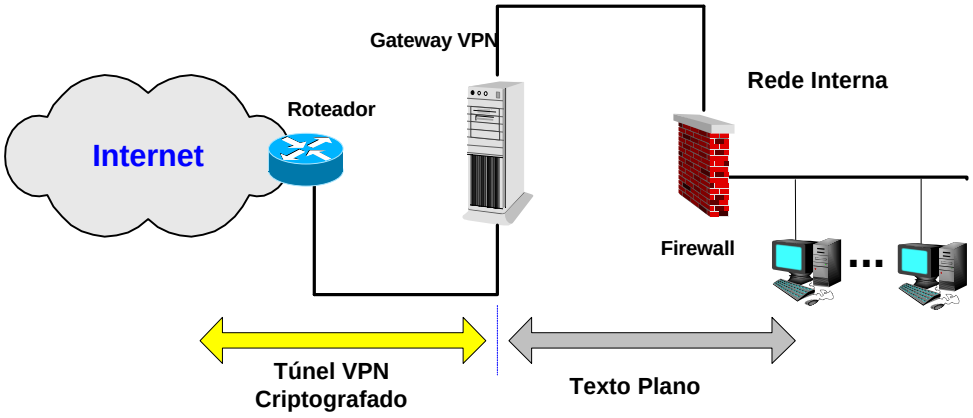


Figura 5.1 – VPN em frente ao Firewall

5.1.2 VPN atrás do *firewall*

A princípio, esta topologia pode parecer adequada e segura, porém, quando analisamos mais profundamente alguns aspectos simples dos protocolos IKE e IPSec, verifica-se também que existe uma grande fragilidade neste posicionamento. Esta topologia requer uma configuração específica, pois para que o túnel VPN funcione é necessário que o *firewall* não analise os pacotes de tipo 50 e 51 (AH e ESP), nem aplique as regras definidas no *firewall*. Os pacotes UDP na porta 500, utilizado pelo IKE, também não podem ser analisados pelo *firewall*, passando diretamente ao *Gateway* VPN. Portanto, cabe ao *firewall* analisar o tráfego que passa fora do túnel VPN, ficando impossibilitado de analisar o tráfego da rede VPN.

Desta forma, esta configuração fragiliza totalmente o *firewall*, uma vez que, este deve necessariamente permitir o tráfego VPN sem a análise do conteúdo das informações.

A figura 5.2 ilustra essa configuração.

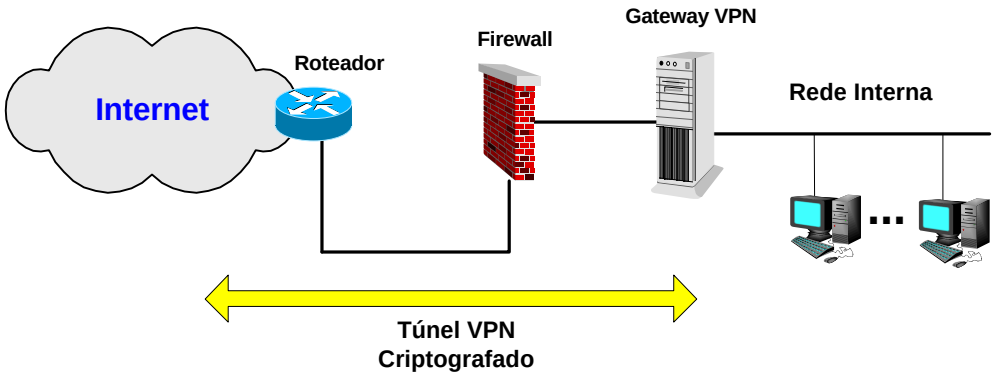


Figura 5.2 – VPN em frente ao Firewall

5.1.3 VPN e *Firewall* integrado

A posição aconselhada e defendida em algumas publicações, inclusive em (KING, 1999) e (SILVA 2003), é a colocação do *Gateway* VPN na mesma máquina que o *firewall*, dado que numa configuração deste tipo todos os pacotes que chegam ao *Gateway* VPN passam antes por um filtro de pacotes ou de estados, o que fornece uma certa proteção contra ataques diretos. Após passarem pelo *Gateway*, terem os cabeçalhos de tunelamento retirados e serem decifrados, os pacotes originais podem passar agora por processo de filtragem, o que não podia ser feito ao entrarem no *firewall* por estarem completamente cifrados.

Esta configuração também possibilita um maior controle e gerência da rede, uma vez que os serviços de VPN e *firewall* estarão concentrados no mesmo equipamento. Por outro lado, esta topologia tem como desvantagem uma possível queda de desempenho no poder de processamento do *firewall* em virtude da própria sobrecarga gerada pela execução dos processos referentes ao *Gateway* VPN e ao *firewall* dentro da mesma máquina.

Cabe observar, que essa desvantagem pode ser reduzida significativamente com a correta e adequada utilização do *software* de filtragem *Iptables*, onde existe a possibilidade de se quebrar a complexidade de regras específicas para a VPN em cadeias de regras menores, otimizando assim a sua interpretação. Além disso, uma outra alternativa que pode vir a melhorar o desempenho desses servidores é a utilização de arquiteturas multiprocessadas com a aplicação de Sistemas Operacionais capazes de suportar o multiprocessamento. O *kernel* 2.6 do sistema *linux* já suporta ambientes multiprocessados de forma bastante eficiente.

5.1.4 *Cenário de um Gateway* VPN com múltiplas DMZs

Outra questão a ser levantada, e não menos importante, refere-se ao posicionamento de um *Gateway* VPN em relação às redes desmilitarizadas (DMZ). Além do posicionamento do *Gateway* VPN em relação ao *firewall*, existem basicamente duas outras configurações possíveis quando se pretende integrar o servidor VPN com outros equipamentos de uma DMZ, a saber: configurar um servidor VPN dentro de uma DMZ separada ou especificar múltiplas DMZs, sendo uma específica para o servidor VPN, separando-se assim, o tráfego VPN do restante da rede.

Na configuração de múltiplos DMZs têm-se um filtro externo e um interno exclusivamente para a VPN. Isto acarreta vantagens significativas como: a simplificação de endereçamentos, a divisão entre o tráfego Internet comum e o tráfego para redes confiáveis via VPN e a possibilidade de uma filtragem exclusiva, das solicitações de conexões originárias da faixa de endereços internos, tanto de máquinas de *extranet*, como de clientes remotos.

A desvantagem dessa configuração é a duplicação de recursos físicos necessários, porém, este problema pode ser minimizado se o *Gateway* VPN for integrado com o filtro externo em um único equipamento. A figura 5.3 apresenta uma topologia com duas redes DMZs.

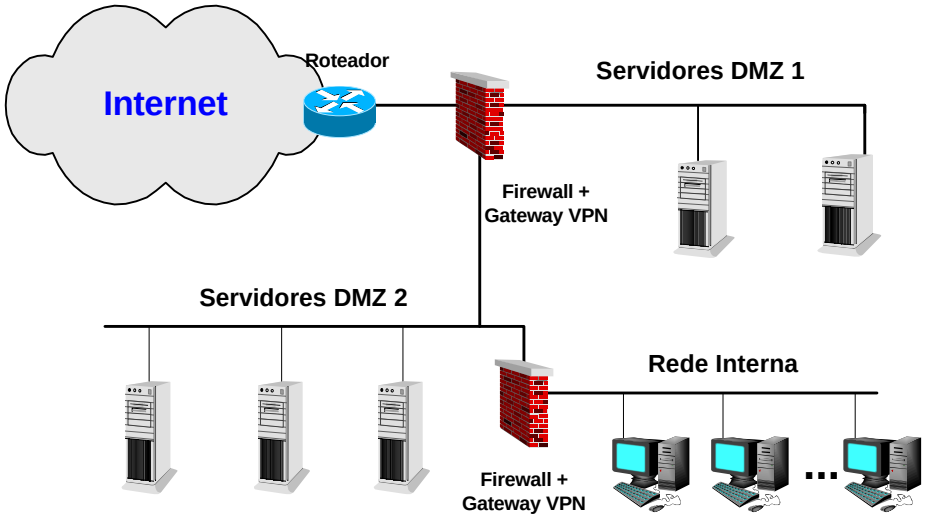


Figura 5.3 – VPN numa configuração de múltiplas DMZs

No cenário de múltiplas DMZs apresentado, o Servidor *Gateway/Firewall* mais externo pode ser bastante útil para conexões de usuários de extranet e usuários remotos que buscam apenas acessar recursos e servidores disponibilizados em uma das DMZs. O Servidor mais interno se predispõe a possibilitar conexões entre redes e usuários da mesma Empresa e os recursos da rede Interna. Esta disposição separa bem os diferentes tráfegos que podem co-existir em uma grande Empresa. Em geral, esta configuração é mais robusta e apresenta um cenário mais adequado para grandes Empresas que desejam estabelecer um modelo mais seguro, em função da própria segmentação do tráfego, e de melhor desempenho.

5.2 Modelo de VPN utilizando infra-estrutura PKI

Como já mencionado neste trabalho, a infra-estrutura de Chave Pública – PKI pode ser utilizada para o estabelecimento de conexões VPN. A idéia neste momento é identificar como o PKI pode ser utilizado e configurado para esse tipo de serviço.

No capítulo sobre IPSec, é descrito o processo IKE (ISAKMP), componente responsável pela troca de chaves para o estabelecimento de conexão VPN. É justamente este processo que pode utilizar a infra-estrutura de Chave Pública para prover o serviço de conexão (estabelecimento de Associação de Segurança), aumentando a segurança na autenticação dos dispositivos VPN. Desta forma, a VPN, utiliza-se do sistema PKI no

momento de se estabelecer uma rede virtual, utilizando os Certificados Digitais para os seguintes serviços: autenticação, gerência de chaves e controle de acesso.

Este trabalho recomenda fortemente a utilização de uma infra-estrutura de Certificados Digitais no processo de autenticação dos servidores VPN IP. Através desta pesquisa, pudemos identificar que certos mecanismos de autenticação, inclusive os disponíveis no FreeS/WAN, fragilizam a segurança de uma infra-estrutura VPN. Desta forma, não recomendamos a utilização dos mecanismos de *Criptografia Oportunista* e de *Chave Secreta*.

A utilização do mecanismo de Criptografia Oportunista fragiliza a solução VPN. A segurança desse método de autenticação depende diretamente da segurança implementada nos servidores DNS. Uma vez que estes servidores geralmente são públicos e alvos de muitos ataques por conterem informações importantes de uma rede, as inclusões das chaves RSA nestes sistemas tornam a segurança de uma infra-estrutura VPN pouco confiável e bastante contestável por muitos especialistas em segurança.

O uso de chaves secretas compartilhadas fragiliza o ambiente das redes corporativas, principalmente nas VPNs extranets e de acesso remoto, uma vez que as redes corporativas devem confiar plenamente na segurança implantada na outra extremidade da conexão (no outro servidor VPN ou no cliente remoto), porém, neste esquema, a Empresa nunca terá garantias reais de privacidade se sua chave secreta. Desta forma, este trabalho não recomenda a utilização de *chaves secretas compartilhadas* para autenticação e criptografia.

5.3 Acesso Remoto

Apesar deste trabalho não possuir interesse em discutir uma solução para o acesso remoto em redes VPN, restringindo-se às pesquisas quanto às conexões entre redes corporativas através de uma VPN, cabe, neste momento, um posicionamento da problemática existente nas conexões remotas.

O grande problema do acesso remoto, através de linha discada, para o estabelecimento de uma VPN, reside no fato que não se sabe, *a priori*, qual é o endereço IP que será atribuído ao equipamento remoto[10]. Este fato gera um problema para a configuração do *gateway* VPN,

[10] Atualmente, praticamente todos os Provedores de Acesso a Internet utilizam serviços de DHCP que realizam atribuições dinâmicas de endereços IP aos clientes remotos.

pois este endereço é necessário para a sua configuração, uma vez que este endereço é necessário para o estabelecimento do túnel VPN. A solução recomendada em geral, a qual é implementada pela maioria dos *softwares* de VPN, estabelece que um *Gateway* VPN considere como a outra ponta do túnel qualquer endereço IP. Este tipo de solução reduz significativamente a segurança da rede corporativa, pois o *firewall* teria que liberar a passagem de pacotes oriundos de qualquer destino endereçados ao *Gateway* VNP.

Várias propostas de solução para o referido problema já foram sugeridas. Em (DENKER et al., 1999), por exemplo, foi proposta uma plataforma chamada MOAT para solucionar parte desse problema[11], uma vez que os provedores forneciam IPs dinâmicos através dos seus servidores DHCP. Essa solução se propôs a atender uma filial da rede principal que se comunica com linha discada, *cable modem* ou similar. A referida plataforma envolvia a modificação dos arquivos de configuração do IPSec (no caso, arquivos de configuração do FreeS/WAN) no momento da conexão, e ambos os lados da conexão deveriam levantar o túnel obedecendo a nova configuração. Essa solução apresentou-se como trabalhosa e complexa demais para o caso de acesso remoto de uma única máquina (FIGUEIREDO, 2003).

Figueiredo (2003, p.6) ressalta que uma solução intermediária é o conhecimento prévio de um conjunto de endereços antecipadamente fornecidos pelo Provedor de Acesso a Internet, endereços que são utilizados pelo acesso remoto, aos seus clientes, para poder configurar o *gateway* VPN, permitindo somente tentativas de conexão a partir de uma determinada faixa de IPs, diminuindo assim a fragilidade do sistema.

5.4 Topologia Proposta

A figura 5.4 apresenta a topologia proposta por este trabalho para Empresas que buscam: interligação entre suas unidades através de VPNs Intranets, estabelecimento de VPNs Extranet (com parceiros de negócio, clientes, fornecedores, etc) e o estabelecimento de VPNs de acesso remoto (usuários remotos), dentro de uma filosofia de defesa em profundidade,

[11] Na época da proposta MOAT, o *software* FreeS/Wan provia somente um mecanismo para a identificação, que era a identificação através do endereço IP de cada extremidade da conexão VPN, descrito no seu arquivo de configuração, não permitindo endereços dinâmicos.

atendendo a requisitos básicos de segurança, e que possuam flexibilidade e escalabilidade. Apresenta também os componentes básicos de segurança para os propósitos descritos.

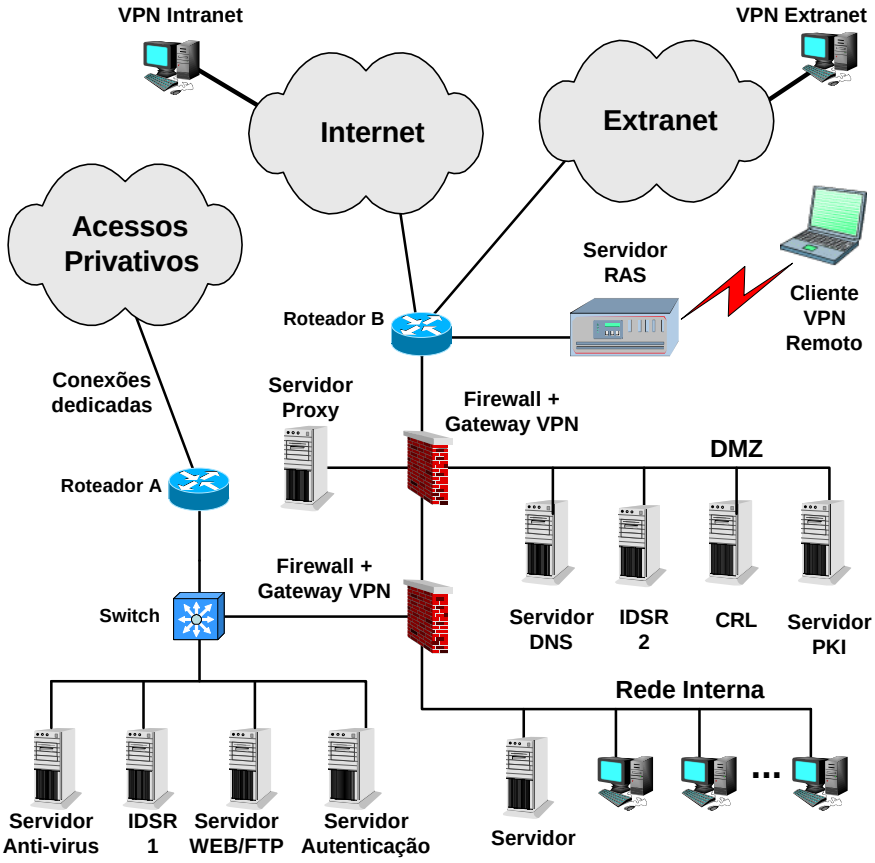


Figura 5.4 - Topologia proposta para Interligação de unidades corporativas através de VPN

Podemos observar que foi proposta a instalação de dois roteadores: o roteador **A,** que tem a finalidade de rotear pacotes entre as unidades organizacionais da própria Empresa (acesso privativo) e o roteador **B**, que realiza o roteamento de pacotes da/para Internet e de clientes remotos através do Servidor de Acesso Remoto, possibilitando acesso a todos os clientes VPN. A vantagem dessa configuração é que ela possibilita uma segmentação do tráfego de rede e permite que as Empresas, que ainda precisam ter canais dedicados de comunicação, tenham uma estrutura mais segura e com tráfego melhor distribuído entre seus componentes de rede.

Para o estabelecimento de uma conexão VPN, cada *gateway* VPN precisa se autenticar com o outro participante da VPN. Como mencionado anteriormente, esta autenticação pode ser feita por meio dos ***Certificados Digitais***. Desta forma, os dispositivos VPN devem implementar um conjunto de funcionalidades capazes de gerar o par de chaves pública e

privada; requisitar e disponibilizar certificados; requisitar e disponibilizar Lista de Certificados Revogados e prover vários métodos de criptografia por chave pública (SILVA, 2003).

Neste cenário, é possível se configurar a Autenticação entre os dispositivos VPNs usando uma Autoridade Certificadora Autônoma. Ele utiliza Certificados Digitais providos através de uma Autoridade Certificadora Autônoma instalada na DMZ, juntamente com a CRL, em uma de suas sub-redes. O posicionamento da CA e da CRL dentro da DMZ possibilitará um tráfego controlado entre os *Gateways* e o Servidor de Autenticação. Se o CA e a CRL estivessem na rede interna da Matriz, somente o *Gateway* VPN da Filial poderia se autenticar na rede da Matriz, uma vez que CA (Servidor PKI) e CRL estão sob a responsabilidade do *Gateway* VPN da Matriz, porém, o inverso não seria aceitável, isto é, não seria possível o *Gateway* da matriz se autenticar na rede da Filial, a menos que configurações específicas fossem feitas entre os *Gateways*.

Foram também postados dois ***Sensores de Rede* SDIR** para monitorar tráfego suspeito em cada segmento de rede desmilitarizado. Há também a utilização de um servidor *web-proxy*, que juntamente com a utilização do protocolo WCCP - *Web Cache Communication Protocol* (protocolo desenvolvido pela CISCO que implementa o *proxy* transparente) deverá centralizar todas as requisições *Web* para ele de forma transparente. Com este *proxy*, é possível se monitorar todo o tráfego de páginas Internet da corporação, além de se ter a possibilidade de restringir o acesso a determinadas páginas *Web* que a Empresa julgue serem não apropriadas ou inseguras. Esta restrição pode ser aplicada a toda rede, a alguns *hosts* específicos (através do IP) ou a subredes. O *software* de *Web-proxy* sugerido por este trabalho é o *Squid*. Maiores informações sobre este *software* podem ser encontradas no seu site oficial, acessando a URL http://www.squid-cache.org/. No *site* da Rede Nacional de Ensino e Pesquisa, em http://www.rnp.br/newsgen/0103/wccp.html, também podemos encontrar uma breve descrição sobre este *software*, o procedimento de instalação e os procedimentos necessários para a configuração e habilitação do WCCP. Desta forma, este trabalho não pretende abordar a sua instalação e configuração por ser bem documentada na Internet.

A figura 5.4 apresenta uma proposta geralmente voltada às exigências de segurança de uma matriz de uma Empresa. Porém, é necessário que se defina alguma proposta mínima de segurança para as redes das outras unidades corporativas (filiais), principalmente àquelas que estão interligadas com a matriz através de VPN. Não é regra, mas estas geralmente requerem

um nível menor de segurança, até porque, essas unidades comumente não apresentam tantos componentes de rede e de Sistemas de Informação quanto a matriz, que justifiquem um investimento em segurança muito pesado. A figura 5.5 ilustra uma proposta menos robusta, direcionada para as redes das filiais, ou pequenas redes corporativas, formadas na sua maioria por *hosts* que rodam sistemas cliente/servidor.

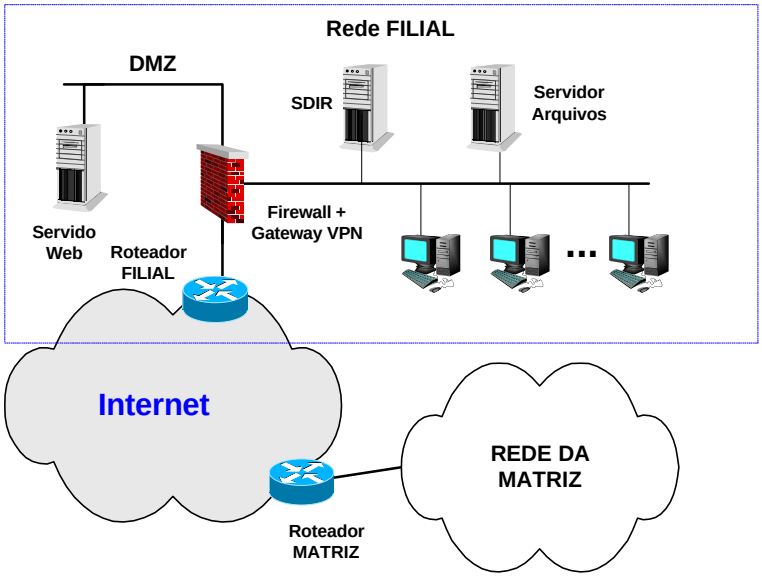


Figura 5.5 - Topologia proposta para as redes das outras unidades corporativas

Cabe observar, que a proposta apresentada pode ser implantada em cada Escritório Estadual da Dataprev (modelo matriz), possibilitando que todas as unidades possam se interligar através de redes VPN baseadas na Internet, e não mais por circuitos de dados interestaduais, o que significaria uma economia bastante considerável, pois o seu atual backbone *Frame Relay* poderia ser substituído por uma infra-estrutura já disponível em todos os Escritórios, a Internet. Porém, como será discutido na próxima seção, ainda não é viável a substituição de todos os circuitos de dados, principalmente os circuitos urbanos e os circuitos para municípios do Interior de alguns Estados.

5.5 Estudo de Caso: proposta do uso de VPNs para as redes da Previdência

A estrutura básica atual da Rede de Telecomunicações da Previdência Social é formada por três camadas distintas, conforme figura 5.6. A primeira é formada pelos nós de

rede dos Centros de Tratamento de Informação do Rio de Janeiro e São Paulo. Esses são equipados com elementos de maior desempenho e redundância, por concentrarem grande parte do fluxo de dados da Rede. Essa camada é responsável pela interligação dos elementos da segunda camada, conexão à Internet e à entidades externas sediadas no Rio ou São Paulo.

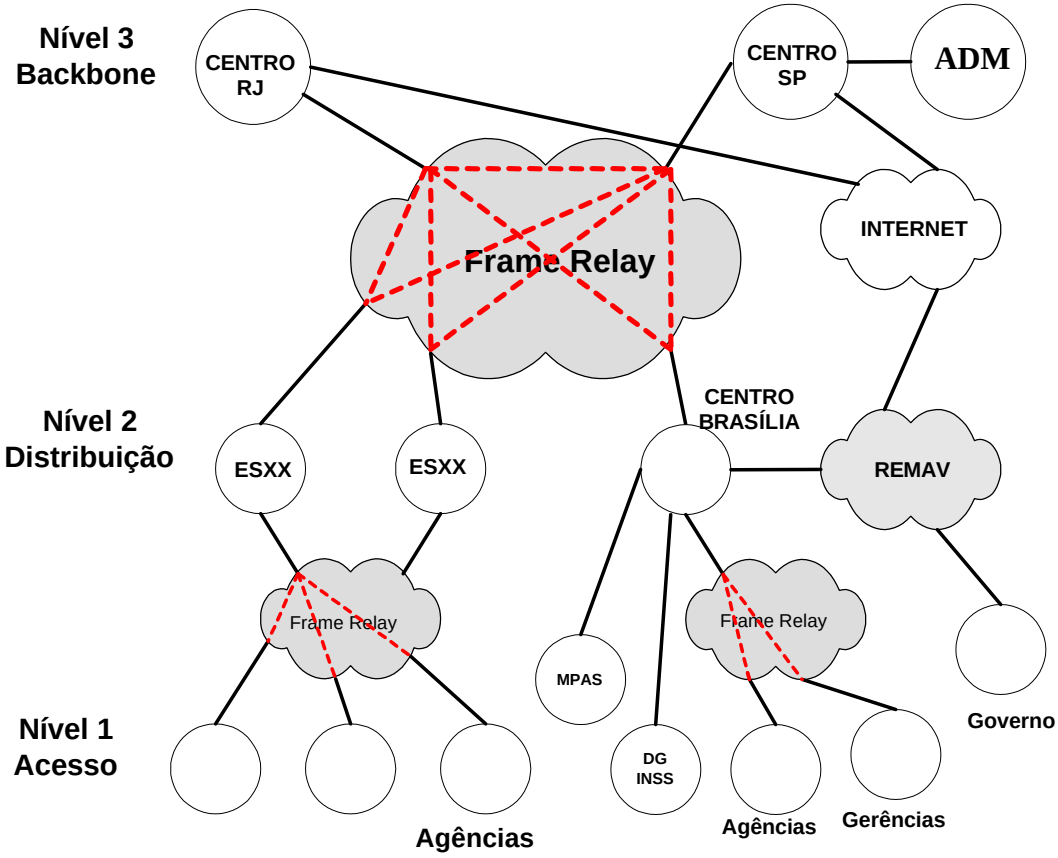


Figura 5.6 - Estrutura básica da Rede da Previdência Social

A segunda camada é formada pelos nós dos Escritórios Estaduais da Dataprev, que interligam todos os pontos de presença da Previdência no Estado. Também existe um acesso Internet para comunicação com a rede interna de Brasília. Cada Escritório Estadual tem uma ligação com o Centro de Informação no Rio e outro em São Paulo.

A terceira camada atende aos postos de atendimento ao público, as Agências da Previdência Social, Gerências Executivas e demais pontos de presença da Previdência Social, com a característica de nós finais de rede.

A nova proposta de topologia para a rede da Previdência Social consiste na interligação dos Escritórios Estaduais aos Centros de Tratamento de Informação do Rio de Janeiro e de São Paulo através de redes VPN.

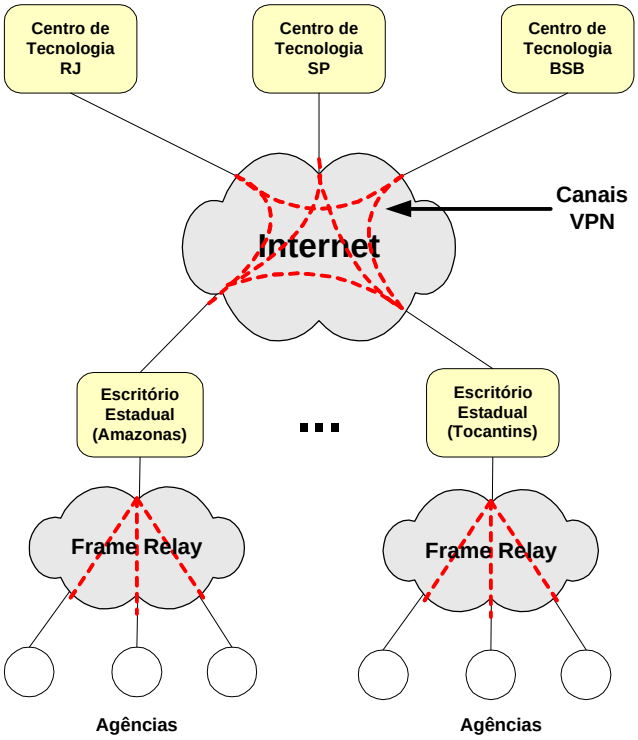


Figura 5.7 - Estrutura proposta para a Rede da Previdência Social usando VPNs

Os *links* dos Escritórios com as Agências permaneceriam estabelecidos em uma rede *Frame Relay* conforme demonstrado na figura 5.7. Os Motivos principais que levaram a permanência desses circuitos *Frame Relay* foram:

- Alto tráfego entre as Agências e os Escritórios, uma vez que os Servidores de Banco de Dados das Agências estão localizados dentro da rede de cada Escritório. Portanto, necessita-se de um canal que propicie segurança e alto desempenho, com velocidade e tempo de respostas adequadas às aplicações de pronto atendimento.

- Os custos de aquisição de *links* de Internet para as Agências, aliado a necessidade de investimento para fortificação do Perímetro Exterior de cada unidade, não compensaria, conforme será analisado na próxima seção.

A estrutura utilizada em cada Escritório seria uma topologia semelhante à mostrada na figura 5.4.

A tabela abaixo apresenta a redução de custos estimada com a substituição dos circuitos *Frame Relay* entre os Escritórios Estaduais e os Centros de Tecnologia da Dataprev

no Rio de Janeiro e em São Paulo. Evidencia-se que as reduções mais significativas são justamente nas regiões mais distantes dos Centros de Tecnologia, Região Norte e Nordeste, pois os circuitos *Frame Relay* nessas localidades são de maior *degrau* e, portanto, mais onerosos.

Tabela 3 – Redução de Custos estimada da Proposta

Fonte: Cotações das Empresas Telemar e Elyte. Mês : Janeiro/2004. (Valores expressos em Real)

	Custo dos Circuitos Frame Relay	***Custo de Acesso a Provedor Local***	***Redução Mensal***	***Redução Anual***
Região Norte	R$ 9.100,00	R$ 3.010,00	R$ 6.090,00	R$ 73.080,00
Região Centro-Oeste	R$ 3.600,00	R$ 1.720,00	R$ 1.880,00	R$ 22.560,00
Região Nordeste	R$ 9.900,00	R$ 3.870,00	R$ 6.030,00	R$ 72.360,00
Região Sudeste	R$ 2.100,00	R$ 1.290,00	R$ 810,00	R$ 9.720,00
Região Sul	R$ 3.150,00	R$ 1.290,00	R$ 1.860,00	R$ 22.320,00
Redução Total de Contratação de Circuitos Ano:				R$ 200.040,00
Custo de Instalações (Provedor Banda Larga):				R$ 37.700,00
Redução de custo de comunicação:				R$ 162.340,00

Vale ressaltar que não estamos considerando os custos referentes aos investimentos necessários na aquisição ou adequação de equipamentos em cada Escritório, a fim de suportar o ambiente de segurança proposto, nem os investimentos em treinamento. Portanto, é necessário ainda que se realize um levantamento específico para tratar esta questão.

5.6 Análise de Custos

A análise de custos realizado neste trabalho tomou por base informações e cotações provenientes de concessionárias de Telecomunicações e Provedores de Internet que operam no Estado do Amazonas. Como estudo de caso, esta análise de custos teve a pretensão de investigar a viabilidade econômica na implementação de VPNs na Previdência Social e discutir questões e problemáticas que influenciam diretamente nesses custos, que nem sempre estão visíveis ou são abordados.

Os custos a que nos referimos nesta seção são os relacionados à contratação de circuitos e serviços para acesso a Internet (para possibilitar a implementação de VPN) em relação ao aluguel de circuitos dedicados.

5.6.1 Situações Favoráveis às VPNs

Neste trabalho, observou-se que nem sempre os custos de uma VPN são menores que as soluções de conectividade feitas através da utilização de circuitos dedicados ou assinaturas de circuitos *Frame Relay*, apesar de todas as bibliografias referentes a VPNs afirmarem categoricamente que as VPNs baseadas em Internet constituem soluções sempre mais econômicas. O Brasil, por exemplo, é um país que possui dimensões continentais e que apresenta uma diversidade muito grande quanto às facilidades de comunicação em suas regiões. Enquanto os Estados da região Sul e Sudeste possuem uma malha de Fibra Ótica cobrindo toda a região, os Estados da região Norte, em especial o Amazonas, apresenta uma malha de telecomunicações ainda muito precária o que influencia diretamente nos custos de circuitos e serviços de Empresas de Telecomunicações e Provedores de Internet, acarretando uma distorção nos valores de circuitos e serviços cobrados em relação ao restante do país. A rede de comunicação do Estado do Amazonas com o resto do mundo ainda é feita via Satélite ou via Rádio. Links de Fibra Ótica só existem dentro de sua Capital. A baixa densidade demográfica no Interior do Amazonas também contribui com a falta de interesse em investimentos em Serviços de rede e telecomunicações nos seus municípios, um exemplo bem claro disso, é a ausência de Provedores de Internet nos municípios do Interior do Estado.

Desta forma, a implementação de redes VPN baseadas na Internet para o interior do Estado do Amazonas, por exemplo, fica totalmente inviável, uma vez que as concessionárias de telecomunicações, como Telemar e Embratel, não possuem pontos de rede IP nesses municípios. Elas fornecem acesso Internet no Interior, porém, a sua rede Internet irá funcionar sobre uma rede determinística ponto-a-ponto até Manaus, o que torna o custo do serviço Internet mais elevado que o custo de manutenção de um circuito ponto-a-ponto, inviabilizando a utilização de VPNs baseadas em Internet. Em alguns municípios, por exemplo, o custo de um circuito *Frame Relay* pode ser mais caro até que um circuito ponto-a-ponto pela ausência de nó *Frame Relay* no município (não faz parte da “nuvem” *Frame Relay*).

Diante desta problemática, conseguimos elaborar um esquema simplificado que descreve as situações em que as VPNs são melhores justificadas, são elas:

1ª Situação

Para redes geograficamente muito distantes, localizadas em Municípios, Estados ou Países distintos, onde o custo de contratação de um circuito *Frame Relay* ou *ponto-a-ponto* ultrapasse significativamente o custo da contratação de um *link* com a Internet e o aluguel dos endereços IPs necessários.

2ª Situação

Para implementação de VPNs extranet. Nesta, as redes que desejam estabelecer uma VPN, que são redes de corporações distintas (Empresas parceiras, Clientes ou Fornecedoras), já devem possuir um *link* com a Internet e endereços IPs alugados.

3ª Situação

Para implementação de VPNs de acesso remoto, onde os usuários remotos já utilizam recursos de Internet disponibilizados por um Provedor de Acesso Remoto (ISP). A Rede corporativa precisa apenas disponibilizar o *link* com a Internet, além dos recursos e serviços que assegurem um nível de segurança satisfatório.

4ª Situação

Para implementação de redes que atendam as situações anteriores e que não tenham missões críticas. Desta forma, não é adequado que se construa uma infra-estrutura de VPN IP para atender redes de alto tráfego e com limitações críticas de tempo de resposta para a transmissão das informações, podendo ocorrer problemas de desempenho, excesso de perda de pacotes e, consequentemente, atrasos na transmissão de dados, sobre os quais a empresa não terá qualquer controle.

Em geral, observou-se que os custos de instalação e manutenção de circuitos dependem de fatores diversos, como por exemplo: localização das redes, quantidade de circuitos e serviços já contratados pelo cliente, disponibilidade de recursos de rede no município, meio físico do *link* pretendido e disponível (fibra ótica, rádio, satélite, par-trançado, etc), larguras de banda, velocidade e quantidade de endereços.

Outra consideração importante relacionada ao custo de construção de uma infra-estrutura VPN, é a necessidade de manutenção e gerência desta rede, o que envolve custo mensal de pessoal de suporte técnico para realizar tais atividades. A análise de custos realizada neste trabalho não contabiliza os custos relacionados na manutenção e na gerência da rede, bem como os custos necessários para a capacitação dos Recursos Humanos e aquisição de novos equipamentos ou módulos de microinformática para suportar o modelo proposto. Existe muito a ser estudado sobre a implantação desse tipo de arquitetura.

A Tabela 4 apresenta uma comparação entre os custos de serviços de comunicação dedicado e de acesso a Internet. Estes foram os menores custos de aluguel de circuitos e serviços de comunicação entre a cidade de Manaus e as cidades de Recife, Brasília, Rio de Janeiro e Porto Alegre (escolhidas por representarem as demais regiões do Brasil). Esse levantamento foi feito junto às concessionárias de Telecomunicações Telemar e Embratel e a alguns Provedores de Acesso a Internet (ISP) via rádio.

Tabela 4 – Comparação entre os custos de serviços de comunicação dedicado e da Internet
Fonte: Cotações das Empresas Telemar e Elyte. Mês : Janeiro/2004. (Valores expressos em Real)

	Tipo de Serviço	***Manaus-Recife***	***Manaus-Brasília***	***Manaus- Rio de Janeiro***	***Manaus-Porto_Alegre***
A	Link Ponto-a-Ponto Serviço TC DATA 128Kbps	R$ 2.595,00	R$ 2.539,00	R$ 2.539,00	R$ 2.634,00
B	Frame Relay 128Kbps/CIR 64Kbps Serviço TC Frame Way	R$ 1.420,00	R$ 1.380,00	R$ 1.380,00	R$ 1.449,00
C	Acesso Internet pelo Serviço TC IP Connect 128Kbps	R$ 1.437,00	R$ 1.437,00	R$ 1.437,00	R$ 1.437,00
D	Acesso Internet por Provedor Banda Larga Local (média), de 128 Kbps	R$ 430,00	R$ 430,00	R$ 430,00	R$ 430,00
E	Redução Mensal Custos (B – D)	R$ 990,00	R$ 950,00	R$ 950,00	R$ 1.019,00
	Redução Anual Média (12 * E)	R$ 11.880,00	R$ 11.400,00	R$ 11.400,00	R$ 12.228,00

A redução apresentada na tabela é referente à diferença do menor custo de circuito dedicado, que foi o serviço *TC Frame Way* da Telemar, que implementa uma rede *Frame Relay*, com o menor custo de *link* com a Internet apresentado, que foi o serviço cotado com a

Empresa Elyte de Manaus, serviço este que possibilita a implementação de VPNs baseadas na Internet.

A tabela seguinte demonstra os custos médios dos mesmos serviços de comunicação para interligação de pontos de rede dentro da capital Manaus, entre um ponto em Manaus e um município do Interior do Amazonas, próximo da Capital, que chamamos de Interior 1 (município cuja distância para Manaus é menor que 500 Km) e entre Manaus e um município do interior mais distante, interior 2 (município cuja distância para Manaus é maior que 500 Km). Ressaltamos que não existe ISP nos municípios do Interior do Amazonas e que o custo de um *Frame Relay* na Capital é menor que o custo de um acesso a Internet. Em função disso, percebemos que a adoção de VPNs para dentro do Estado do Amazonas não se justifica.

Tabela 5 – Comparação entre os custos de serviços na Capital e no Interior do Estado do Amazonas
Fonte: Cotações das Empresas Telemar e Elyte. Mês : Janeiro/2004. (Valores expressos em Real)

Tipo de Serviço	***Ponta A: Manaus Ponta B: Manaus***	***Ponta A: Manaus Ponta B: Interior 1***	***Ponta A: Manaus Ponta B: Interior 2***
Link Ponto-a-Ponto Serviço TC DATA 128Kbps	R$ 690,00	R$ 2.539,00	R$ 1.174,00
Frame Relay 128Kbps/CIR 64Kbps Serviço TC Frame Way	R$ 358,00	R$ 1.174,00	R$ 926,82
Acesso Internet pelo Serviço TC IP Connect 128Kbps	R$ 430,00	R$ 1.437,74	R$ 1.437,74
Acesso Internet por Provedor Banda Larga Local 128Kbps (média)	R$ 430,00	**Não Disponível**	**Não Disponível**

Novamente os menores custos apresentados foram da Empresa Telemar e Elyte.

6 Implementação de uma VPN

Este trabalho desenvolveu um experimento que implementa uma solução VPN utilizando ferramentas e sistemas livres, com o intuito de avaliar e demonstrar o funcionamento e viabilidade técnica nessa plataforma, antes da sua utilização em produção. As seções seguintes descrevem todos os aspectos gerais envolvidos na solução adotada, os quais devem ser observados e seguidos na construção de uma infra-estrutura VPN básica.

6.1 Implementação de uma VPN em GNU/LINUX

Para este trabalho, foram investigadas as características das principais ferramentas existentes no mercado que possibilitam a construção de VPNs, bem como, os principais sistemas operacionais para a solução.

Como o intuito deste trabalho era estabelecer um ambiente de *software* totalmente livre, foi escolhido o sistema operacional GNU/LINUX. Informações mais detalhadas sobre este sistema podem ser obtidas através da leitura do apêndice B.

Resumidamente, os motivos principais pela escolha desse sistema operacional para a implementação de VPNs foram:

- Suporte a ferramentas e aplicações livres para implementação de *Gateways* VPN/*Firewalls*, Servidores *Proxy*, Servidores de Autenticação, Servidores PKI, Servidores SDI e NATs.

- Suporte ao conjunto de protocolos TCP/IP e suporte à interoperabilidade com outros sistemas operacionais como MS Windows, Macintosh, *Novell* e outras variações de UNIX.

- Suporte ao protocolo WCCP (*Web Cache Communication Protocol*) nas ferramentas de *Proxy-cache* existentes no Linux, como por exemplo, o *Squid*, *software* que proporciona transparência na implementação de serviços de *Proxy* através do WCCP e de roteador Cisco. O WCCP é um protocolo que foi desenvolvido pela CISCO que, em conjunto com um servidor *proxy-cache*, constitui em uma solução muito interessante para ***proxy transparente***.

- O baixo custo e facilidade de obtenção, pois o Sistema Operacional, bem como, os produtos de segurança para este ambiente são GPL e estão disponíveis para *download* gratuitamente nas páginas de distribuidores e outros fabricantes. Maiores informações sobre a licença GPL pode ser consultada no Apêndice A deste trabalho.

- O excelente desempenho e estabilidade do sistema que já são conhecidas pela comunidade científica.

- Suporte a várias plataformas como Intel, IBM, Motorola, Alpha, Amiga, Sparc, dentre outros. E com isso, possui maior flexibilidade em uma mudança de *hardware*.

- Possibilidade de se implementar customizações no sistema e de seu *kernel* para o projeto aqui apresentado, pois o seu código-fonte é aberto.

- Por ter sido desenvolvido em uma filosofia de desenvolvimento colaborativo e *Open Source*, possui excelente suporte na rede Internet e constantes atualizações e otimizações no seu código. Assim que é encontrada uma falha de segurança, logo é lançada uma correção para tal, em contraposição ao desenvolvimento de vários *softwares* comerciais fechados e proprietários em que, muitas vezes o fabricante demora a lançar correções de falhas no seu

sistema, sendo este um ponto crucial para escolha deste tipo de sistema para um projeto que exige um alto grau de confiança, estabilidade e escalabilidade.

6.1.1 Software Adotado para Implantação da VPN

Uma VPN pode ser implementada por vários dispositivos, tais como: roteadores, servidores de acesso remoto, equipamentos específicos, ou ainda *software* instalado em servidores ou em micros. Neste trabalho, a VPN será implementada através de *software*, configurando-se assim Servidores *Gateways* VPN. E como este trabalho também possui o intuito de avaliar a implementação de uma VPN em uma plataforma de *software* livre, além de observar fatores como: segurança e baixo custo de implementação, a solução para implementação de uma VPN proposta baseia-se no *software* FreeS/WAN versão 2.03, cuja denominação origina-se do termo *Secure Wide Area Network* (S/WAN), e a palavra *Free* por ser um *software* livre . Além de ser item de avaliação deste trabalho, a escolha deste *software* deveu-se também aos seguintes motivos:

- É um *software* livre e, por conseguinte, utiliza padrões abertos e atende aos objetivos deste trabalho, além de não onerar custos a solução pretendida.

- Possui como base a arquitetura IPSec, padrão amplamente utilizado nos serviços de criptografia na Web. O Capítulo 4 abordou as vantagens desse protocolo, e o porque ele é a melhor alternativa de tunelamento para os propósitos deste trabalho (interligação de redes corporativas).

- Possui suporte ao Gerenciamento de Chaves (IKE – Internet Key Exchange).

- Interoperabilidade: pode ser conectado a qualquer equipamento que implemente o padrão IPSec.

- Aceita conexões fixas e móveis.

- É transparente: não requer modificações em nenhuma das estações de trabalho ou servidores que fazem parte da rede privada (diferentemente do serviço SSH).

- Não possui limitação para número de usuários, podendo, inclusive, serem montados vários túneis com outros *gateways* remotos.

- Pode utilizar Certificados Digitais no estabelecimento de Conexões para Autenticação, Gerência de Chaves e Controle de Acesso.

A tabela abaixo apresenta um comparativo de algumas funcionalidades entre os principais *softwares* de VPN existentes no momento da elaboração desta dissertação. Observa-se nitidamente que o **FreeS/WAN** (com a aplicação dos Patchs) e o **Kame**, dentre os produtos avaliados, são os *softwares* VPN com maior número de funcionalidades para o estabelecimento de Túneis VPN, porém, o FreeS/WAN leva uma certa vantagem, pois aceita conexões remotas, possui suporte a NAT e Criptografia Oportunista, além disso o Kame roda em plataforma FreeBSD, que apesar de ser semelhante ao GNU/LINUX, sendo também uma plataforma livre e com irrefutável qualidade e prestígio na comunidade, o autor deste trabalho ainda não possui experiência nessa plataforma. Desta forma, este autor optou por utilizar um *software* que rodasse em ambiente GNU/LINUX. A utilização do Kame poderá ser avaliada em um outro trabalho de pesquisa.

Tabela 6 – Comparativo de Programas VPN
Fonte: (FREES/WAN, 2003)

	Estabelecimento de Conexão VPN				**Suporte NAT**	**Suporte a Conexão de Usuário Remoto**
	Chave Secreta	**Criptografia Oportunista**	**Chaves RSA**	**Certificado Digital (X.50 9)**		
FreeS/Wan (com Patchs X.509 e NAT)	Sim	Sim	Sim	Sim	Sim	Sim
FreeS/Wan	Sim	Sim	Sim	**NÃO**	**NÃO**	Sim
Kame (FreeBSD)	Sim	**NÃO**	Sim	Sim	**NÃO**	**NÃO**
isakmpd (OpenBSD)	Sim	**NÃO**	**NÃO**	Sim	**NÃO**	**NÃO**
McAfee VPN	**NÃO**	**NÃO**	Sim	Sim	**NÃO**	Sim
MS Windows 2000/XP	**NÃO**	**NÃO**	**NÃO**	Sim	**NÃO**	Sim
SSH Sentinel	**NÃO**	**NÃO**	**NÃO**	Sim	**NÃO**	Sim
Check Point FW-1/ VPN-1	**NÃO**	**NÃO**	**NÃO**	Sim	**NÃO**	Sim

É importante ressaltar que o FreeS/WAN implementa como padrão os algoritmos de chaves assimétrica RSA e simétrica 3DES, além das funções *hash HMAC* MD-5 e HMAC

SHA-1. Entretanto, diversos programadores contribuem, através de *patches*[12], os quais adicionam funções extras não incorporadas ao programa FreeS/WAN original, seja para aumentar a segurança da VPN, seja para permitir a utilização deste em diferentes ambientes. Os principais são: suporte a certificados digitais (X.509); adição de algoritmos de criptografia de chave simétrica, como o AES, *Blowfish* e *Serpent*; passagem pelo *Network Address Translation* (NAT), quando este estiver implementado, entre outros. Os *patchs* mencionados estão disponíveis na página http://www.freeswan.ca/download.php. Existe ainda um programa *espelho* do *FreeS/WAN*, que incorpora os principais *patches* existentes ao programa original, inclusive o suporte a Certificados Digitais e ao NAT, chamando de *SuperFreeS/WAN*, facilitando assim a instalação desta ferramenta.

Como já mencionado, o Sistema Operacional escolhido foi o GNU/Linux, especificamente a distribuição Red Hat 9, com a versão 2.4-20 do Kernel Linux.

6.1.1.1 Componentes do FreeS/WAN

O FreeS/WAN é composto de três partes principais, a saber:

- **KLIPS** (*Kernel* do IPSec) - implementa o protocolo AH e ESP do IPSec e realiza a negociação de pacotes IP dentro do Kernel (FREES/WAN, 2003).

- **PLUTO** (D*aemon IKE)* – Implementa o IKE, realiza a negociação de chaves IKE e trata da negociação de conexões entre entidades VPN (outros sistemas), interagindo com o KLIPS quando necessário. Este serviço pode ser iniciado automaticamente através de scripts de inicialização do FreeS/WAN (FREES/WAN, 2003).

- **Scripts** - que provêem interface de administração do ambiente FreeS/WAN (FREES/WAN, 2003).

6.1.1.2 Criptografia Oportunista no FreeS/WAN

Como exposto anteriormente, o padrão IPSec, padronizado pelo IETF, baseia-se no conceito de Associação de Segurança, que define os parâmetros de segurança a serem

[12] Alteração de um programa, que acrescenta ou modifica apenas uma pequena parte deste.

aplicados entre duas entidades IPSec, como por exemplo, protocolos, algoritmos de criptografia e chaves a serem utilizadas, além de outras informações de segurança. O FreeS/WAN pode ainda utilizar um conceito chamado de ***Criptografia Oportunista*** (*Opportunistics Encryption*), ou simplesmente OE, que significa que qualquer *Gateway* VPN, implementado através do FreeS/WAN, pode criptografar seu tráfego, mesmo sem conhecimento e intervenção dos administradores de redes das entidades (*gateways*) e mesmo sem nenhuma informação pré-configurada sobre a outra entidade (FREES/WAN, 2003).

Vale ressaltar que a Criptografia Oportunista foi submetida a um processo de padronização junto ao IETF. Atualmente, a Criptografia Oportunista é uma *Internet draft* da IETF (RICHARDSON; REDELMEIER; SPENCER, 2003), a qual, possivelmente, pode se tornar um padrão dentro em breve.

Neste mecanismo, ambos os sistemas pegam informações de autenticação e criptografia que necessitam do Servidor DNS (*domain name service*) designado, o qual já dispõe de mecanismo de busca de endereços IPs. Obviamente, para que este mecanismo funcione de forma automática, os administradores de rede devem configurar previamente as informações de autenticação e criptografia no Servidor DNS e habilitar a Criptografia Oportunista nos *Gateways* VPNs (FREES/WAN, 2003). Desta forma, os *Gateways* VPNs podem criptografar o que estiver configurado ou também podem aceitar tráfego não criptografado, caso esta seja a política de segurança implementada pelo administrados da rede. Neste trabalho, mostraremos também como configurar a Criptografia Oportunista.

A seguir será mostrado como a VPN foi implementada, detalhando o cenário dos testes e os aspectos de sua configuração, além dos resultados obtidos.

6.1.2 Cenário de Implementação

A Figura 6.1 ilustra a topologia utilizada durante uma simulação de uma rede VPN utilizando o FreeS/WAN em plataforma GNU/Linux no ambiente da Dataprev. Os testes foram realizados simulando-se duas redes distintas de uma mesma empresa, aqui denominadas de Rede A e Rede B, onde cada rede possui um *Gateway* VPN implementado em FreeS/WAN, na versão 2.03. Com o intuito de simular uma rede pública não confiável, como a Internet, foi postado um computador entre elas, com duas interfaces de rede, com o serviço de Roteamento entre as duas redes.

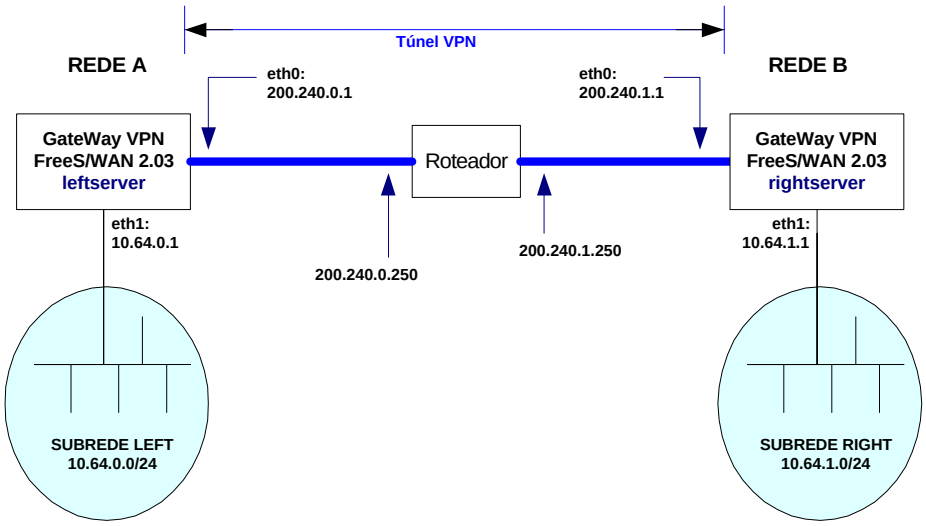


Figura 6.1 – Cenário da implementação de uma VPN com o FreeS/WAN

Nos *Gateways*, foram instalados a distribuição do *Red Hat 9*, com o *kernel* linux 2.4-20 em sua configuração padrão[13]. Nestes, foram instaladas e configuradas duas interfaces de rede, sendo uma voltada para a rede interna da empresa (denominada *eth1* no linux) e a outra para a Internet (denominada *eth0*). Também foram instaladas duas placas de rede no roteador que simulava a Internet, sendo uma voltada para a Rede A (no experimento, interface *eth0*, IP 200.240.0.250) e a outra para a Rede B (interface *eth1*, IP 200.240.1.250)[14], servindo como roteadores das redes.

Posteriormente, foi postado um equipamento Windows XP, rodando o *software* ***Sniffer Pro***, versão 4.70, da *Network Associates*, para monitorar (visualmente) o fluxo da comunicação entre as duas redes, objetivando acompanhar e comprovar o estabelecimento de seção IKE, IPSec e da comunicação criptografada.

6.1.3 Configurações de Rede utilizadas

Por não fazer parte dos objetivos deste trabalho, a instalação do Linux não será abordada. Porém, algumas configurações de roteamento e de DNS (serviço *named* do Linux), específicas para o funcionamento adequado de uma VPN (com Criptografia Oportunista)

[13] O Kernel não foi recompilado, pois foi aplicado os RPMs do FreeS/WAN, que já alteram o Kernel quando da sua instalação.
[14] A máscara de rede adotada em todos os experimentos deste trabalho foi a 255.255.255.0

serão demonstradas. A implementação foi realizada baseando-se na topologia apresentada na seção anterior. A seguir, é apresentada a configuração de rede adotada para a implementação da VPN mencionada.

Tabela 7 – Configuração de Rede Adotada no Experimento

Componentes da REDE A	
- Servidor *GateWay* VPN	
Sistema Operacional:	Red Hat 9, linux 2.4-20
software VPN:	FreeS/WAN 2.03
Hostname:	leftserver
IP da Interface eth0:	200.240.0.1
IP da Interface eth1:	10.64.0.1
Netmask:	255.255.255.0
DNS:	200.240.0.1
- Host (Rede A)	
Sistema operacional:	Windows XP
IP:	10.64.0.50
Gateway:	10.64.0.1

Componentes da REDE B	
- Servidor GateWay VPN	
Sistema Operacional:	Red Hat 9, linux 2.4-20
software VPN:	FreeS/WAN 2.03
Hostname:	rightserver
IP da Interface eth0:	200.240.1.1
IP da Interface eth1:	10.64.1.1
Netmask:	255.255.255.0
DNS Primário:	200.240.1.1
DNS Secundário:	200.240.0.1
- Host (Rede B)	
Sistema operacional:	Windows XP
IP:	10.64.1.50
Gateway:	10.64.1.1

ROTEADOR	
Sistema Operacional:	Conectiva 9, linux 2.4-21
Hostname:	roteador
IP da Interface eth0:	200.240.0.250
Apelido Interface eth0:	gwleft
IP da Interface eth1:	200.240.1.250
Apelido Interface eth1:	gwright
Netmask:	255.255.255.0

Abaixo, é apresentado o roteamento definido no equipamento que implementa o roteador[15]. Pode-se observar claramente que foram definidas duas rotas estáticas nesse

[15] Resultado da execução do comando *route* do linux.

equipamento, uma para a rede 10.64.0.0/24, a qual é roteada através do endereço IP 200.240.0.1, que por sua vez fora definido roteamento através da interface eth0, e a rede 10.64.1.0/24, a qual é roteada através do endereço IP 200.240.1.1, rota esta definida através da interface eth1.

Tabela 8 - Tabela de Roteamento IP do Roteador entre os *Gateways* VPN no experimento

Destino	Roteador	MáscaraGen.	Opções	Métrica	Ref	Uso	Iface
200.240.1.0	*	255.255.255.0	U	0	0	0	eth1
200.240.0.0	*	255.255.255.0	U	0	0	0	eth0
10.64.0.0	200.240.0.1	255.255.255.0	UG	0	0	0	eth0
10.64.1.0	200.240.1.1	255.255.255.0	UG	0	0	0	eth1
127.0.0.0	*	255.0.0.0	U	0	0	0	lo

6.1.4 Instalação e Configuração da VPN

A seguir, será demonstrado todos os passos necessários para a instalação e configuração do FreeS/WAN, bem como, toda a configuração do ambiente VPN, servidor DNS e arquivos de configuração do Linux, de acordo com a topologia apresentada.

6.1.4.1 Preparação do Sistema Operacional Linux

O autor desta dissertação teve a oportunidade de instalar e configurar o FreeS/WAN em diversas distribuições diferentes do Linux, entre elas a **Red Hat**, **Conectiva** e **Debian**, e em todas elas, houve a necessidade de algumas modificações na configuração padrão do Linux. Constatou-se que sem essas modificações os *Gateways* VPN não funcionavam adequadamente.

Desta forma, para que a VPN possa funcionar adequadamente é necessário configurar alguns parâmetros do Linux antes de iniciá-la. Para fazer isso, este trabalho se propôs a documentar tais alterações de configuração que estão relacionadas a seguir:

1. Habilitar o "*packet forwarding*" no Linux. No Red Hat e no Conectiva o arquivo que deve ser modificado é o */etc/sysctl.conf* e a opção que deve ser alterada é a *net.ipv4.ip_forward*, que deverá ficar da seguinte forma:

```
net.ipv4.ip_forward = 1
```

2. Desabilitar a proteção contra o IP *Spoofing* no Linux. Esta proteção deve ser desabilitada nos *Gateways* VPN. Para isso, é recomendado se inserir no final

do arquivo /etc/rc.local, arquivo de *script* que é executado na inicialização do Sistema, os seguintes comandos, considerando que a interface utilizada para o IPSec seja a *eth0*:

```
echo "1" > /proc/sys/net/ipv4/ip_forward
echo "0" > /proc/sys/net/ipv4/conf/eth0/rp_filter
echo "0" > /proc/sys/net/ipv4/conf/ipsec0/rp_filter
```

3. No arquivo /etc/sysconfig/network, alterar o parâmetro FORWARD_IPV4 para:

```
FORWARD_IPV4=yes
```

4. Habilitar as opções de rede do *kernel* listadas a seguir, caso ainda não estejam habilitadas na configuração padrão do Linux. Posteriormente, deve-se recompilar o *kernel*. Ressalta-se que na distribuição do Linux utilizada neste trabalho, todas as opções necessárias foram habilitadas quando da instalação do pacote do *software* FreeS/WAN (em formato RPM).

```
IP Security Protocol
IPSEC: IP-in-IP encapsulation (tunnel mode)
IPSEC: Authentication Header
HMAC-MD5 authentication algorithm
HMAC-SHA1 authentication algorithm
IPSEC: Encapsulating Security Payload
3DES: encryption algorithm
IPSEC: IP Compression
IPSEC: Debugging Option
```

6.1.4.2 Instalação do software FreeS/WAN

Existem três maneiras básicas de instalar o FreeS/WAN:

- Instalando o FreeS/WAN a partir dos pacotes de *software* fornecidos juntamente com distribuição do Linux. A distribuição da Conectiva na sua versão 9, por exemplo, traz o FreeS/WAN 1.99. Vale ressaltar que para este trabalho, esta opção de instalação foi descartada, pois versões mais atualizadas do *software* já estavam disponíveis na Internet.

- Instalando a partir de pacotes *Red Hat Packet Manager* – RPM disponíveis na Internet. Os RPMs são pacotes binários pré-compilados para um *kernel* específico.

- Instalando através da compilação das fontes do FreeS/WAN.

Para este trabalho, a opção escolhida foi a instalação do FreeS/WAN a partir de um RPM para o Linux da Red Hat, *kernel* 2.4.20. Desta forma, obteve-se uma versão mais atual e pré-compilada. Com isso, não houve a necessidade de se recompilar o *kernel* do Linux, pois a própria instalação do FreeS/WAN, via RPM, altera a pilha TCP/IP do *kernel*, incluindo o IPSec na camada de rede. Os arquivos fontes do FreeS/WAN, bem como os RPMs para o Red Hat, foram obtidos em http://www.freeswan.org/download.html. O Arquivo no formato RPM deve ser da forma "freeswan-*XX*.*YY*.rpm", onde *XX* é a versão do arquivo e *YY* é a plataforma. Neste trabalho a plataforma utilizada foi a "i386", apropriada para processadores Intel.

Neste trabalho, foram feitos *downloads* dos seguintes RPMs:

```
freeswan-module-2.03_2.4.20_20.9-0.i386.rpm
freeswan-userland-2.03_2.4.20_20.9-0.i386.rpm
```

Observa-se que em todos as opções de instalação, é necessário que se utilize a conta *root*.

Depois de verificado as assinaturas digitais dos arquivos obtidos, tarefa que pode ser feita com a ferramenta GNU *Privacy Guard – gnupg* (consultar www.gnupg.org para maiores informações sofre este *software*), que é uma implementação do PGP para linux, a instalação pode ser executada através do comando abaixo descrito, executado no diretório onde encontram-se os RPMs:

```
# rpm –ivh freeswan*
```

Após a execução desse comando os arquivos binários e bibliotecas serão instalados e dois arquivos serão criados no diretório **/etc**, os arquivos "**ipsec.conf**" e o arquivo "**ipsec.secrets**".

O arquivo **/etc/ipsec.conf** é o arquivo de configuração das conexões VPN, onde se informa os endereços das redes envolvidas, o tipo de criptografia, transporte e outros parâmetros.

O arquivo **/etc/ipsec.secrets** deverá conter a chave de criptografia.

6.1.4.3 Autenticação no FreeS/WAN

Em uma VPN é necessário que cada participante seja autenticado no momento de se estabelecer uma sessão entre eles. No FreeS/WAN existem duas formas de autenticação a saber (SILVA, 2003):

- **Chaves manuais**. Onde cada participante compartilha uma chave secreta que deve ser distribuída de forma segura. Essa chave é simétrica, servindo para criptografar e descriptografar as mensagens entre as entidades de uma VPN.

- **Chaves automáticas**. Nesta, as duas entidades se autenticam entre si e negociam a chave secreta.

Vale ressaltar que o FreeS/Wan suporta dois tipos de chaves de autenticação, a chave pública RSA (assimétrica) e a chave secreta, e que cada conjunto de chaves entre dois hosts ou *gateways* deve usar o mesmo tipo de chave, RSA ou Secreta, nunca tipos de chaves diferentes.

Como mencionado, o arquivo que conterá a chave é o **/etc/ipsec.secrets**. E este deverá estar presente nos dois *hosts* ou *gateways*. Recomenda-se, por questões de segurança, que somente o usuário *root*, deva ter acesso a leitura e a escrita no arquivo /etc/ipsec.secrets, assim como em outros arquivos de configuração do FreeS/WAN e IPSec. Portanto, a permissão para esse, e para outros arquivos de configuração do FreeS/WAN, deve ser sempre 600 (rw- --- ---), isto é, permissão de leitura e escrita para o dono (*root*) e sem qualquer tipo de acesso para outros usuários e grupos. Para isso, deve-se executar o seguinte comando para garantir essa propriedade:

```
# chmod 600 /etc/ipsec.*
```

Observa-se que na Criptografia Oportunista, a chave pública será colocada no servidor DNS, permitindo que qualquer entidade que queira estabelecer uma conexão possa fazê-lo de uma forma automática mediante consultas no servidor DNS. Ainda neste capítulo, será mostrado como realizar o registro da chave pública no servidor DNS.

Após a instalação do *software* FreeS/WAN, pode-se criar um par de chaves RSA para cada *Gateway* VPN. Conforme já discutido, o FreeS/WAN irá utilizar essas chaves para o processo de autenticação.

Para a criação dessas chaves, foi utilizado o seguinte comando (com usuário *root*):

```
# ipsec newhostkey --output /etc/ipsec.secrets --hostname <nome>
```

Onde <nome> foi substituído por ***leftserver.vpn*** para a criação das chaves no gateway VPN da rede 200.240.0.0 e ***rightserver.vpn*** para a criação das chaves no gateway VPN da rede 200.240.1.0.

O Apêndice C apresenta os arquivos ***/etc/ipsec.secrets*** gerados, no experimento, após a execução do comando ***ipsec newhostkey*** nos servidores VPN.

6.1.4.4 Configuração das conexões VPN no FreeS/WAN

Como mencionado anteriormente, as configurações de conexões VPN no FreeS/WAN são feitas no arquivo **/etc/ipsec.conf**.

Esse arquivo usa a nomenclatura ***left*** e ***right*** para representar as duas entidades *gateways* envolvidas na conexão, bem como parâmetros que começam com essa nomenclatura. Um *gateway* VPN é indicado pela opção *left*, enquanto o outro gateway é designado por *right*. Não é necessário preocupar-se qual entidade será *left* e qual será *right*, é uma questão de escolha do administrador da rede. A expressão *left* e *right* é apenas utilizada para identificar dois servidores *Gateways* VPNs e suas respectivas LANs.

As LANs são identificadas por ***leftsubnet*** e ***rightsubnet***, no formato <IP da Rede/Máscara>. No cenário implementado, o *left* é o *gateway leftserver*, e a *leftsubnet* é 10.64.0.0/24, ou seja, a LAN possui endereço de rede 10.64.0.0 e máscara de rede 255.255.255.0. E o *right* é o servidor *rightserver*. A *rightsubnet* é a 10.64.1.0/24. Observa-se que o servidor *leftserver* protege a subrede 10.64.0.0/24 (*leftsubnet*) e o servidor *rightserver* protege a subrede 10.64.1.0/24 (*rightsubnet*).

O arquivo /etc/ipsec.conf é estruturado em três seções a saber:

- A seção ***config setup***- É também chamada de seção de configuração básica. Nesta seção temos as configurações gerais do IPSec.

- A seção ***conn %default*** – Esta seção indica as configurações que serão padrões para todas as conexões da VPN. Vale relembrar que um *gateway* VPN pode possuir mais de uma conexão VPN.

- **Seções de configuração de conexão VPN**. A palavra reservada ***conn*** indica o início de uma configuração de uma conexão VPN. A opção *conn* é seguida pelo nome da conexão. O arquivo pode conter inúmeras configurações diferentes.

Abaixo segue a listagem do arquivo ***/etc/ipsec.conf*** dos *Gateways* VPN e o comentário sobre os parâmetros mais relevantes. Informações adicionais sobre os parâmetros do */etc/ipsec.conf* podem ser encontrados em (FREES/WAN, 2003).

```
# /etc/ipsec.conf - FreeS/WAN IPsec configuration file

version 2.0       # indica a versão do FreeS/WAN

# configuração básica que se aplica a todas as conexões
config setup
interfaces="ipsec0=eth0"
klipsdebug=all
plutodebug=dns
uniqueids=yes

# parâmetros default para todas as conexões VPN.
conn %default
keyingtries=0
esp=3des-md5-96
authby=rsasig
disablearrivalcheck=no

# Definição da Conexão entre os Gateways left e right
conn gwl-gwr
authby=secret
keyexchange=ike
pfs=no
keylife=8h
left=200.240.0.1
leftnexthop=200.240.0.250
right=200.240.1.1
rightnexthop=200.240.1.250
auto=add
```

```
# Configuração para possibilitar que um host da Rede A possa estabelecer conexao VPN com
# outro host da rede B

conn redea-gwr
authby=secret
keyexchange=ike
pfs=no
left=200.240.0.1
leftnexthop=200.240.0.250
leftsubnet=10.64.0.0/24
right=200.240.1.1
rightnexthop=200.240.1.250
rightsubnet=10.64.1.0/24
auto=add
```

A seguir, apresentamos uma explicação sucinta dos parâmetros utilizados:

- **interfaces= “ipsec0=eth0"** : define as interfaces físicas e virtuais que o IPSec utiliza para alcançar a rede remota. Pode-se utilizar eth0, eth1, ou qualquer outra interface de rede existente. Caso seja utilizada a rota *default* configurada, pode-se configurar a variável “%defaultroute”. Com isso, o programa fará a procura de qual interface utilizará. A opção “ipsec0=eth0” indica que o IPSec utilizará a interface eth0 para alcançar a rede remota. Vale ressaltar que “ipsec0” é uma interface virtual criada pelo FreeS/WAN que representa o IPSec.

- **klipsdebug=all** : este parâmetro habilita o modo *debug* do KLIPS. A opção ***none*** (opção *default*) indica que não haverá log. A opção ***all*** indica que o debug será completo.

- **plutodebug=dns** : este parâmetro habilita o modo *debug* do PLUTO. A opção ***none*** indica que não haverá *log*. A opção ***all*** indica que o debug será completo e a opção ***dns*** indica que o debug será realizado somente na consulta ao DNS.

- **uniqueids=yes** : a opção ***yes*** determina que as conexões antigas são fechadas se uma nova conexão usando o mesmo ID, de um endereço IP diferente surgir.

- **keyingtries=0** : este parâmetro estipula quantas tentativas devem ser feitas para negociar uma conexão. O valor 0 (zero) significa que o protocolo nunca desiste da negociação.

- **esp=3des-md5-96** : este parâmetro determina o algoritmo de criptografia e autenticação que se deseja utilizar. *3des-md5-96* é uma boa opção, a qual é sugerida como padrão. Esta opção, define o algoritmo DES triplo com encadeamento de blocos de cifra com autenticação provida pelo algoritmo MD5. Outra opção é *3des-sha1-96*, que utiliza o algoritmo SHA-1 para autenticação.

- **authby=rsasig** : define como os *gateways* farão autenticação entre eles. Neste caso será utilizada uma autenticação de Assinatura Digital RSA (é a opção *default*). Outras opções possíveis são: ***secret*** usada quando se deseja utilizar compartilhamento de chaves secretas e ***never*** para que a conexão nunca seja aceita (usada para desabilitar conexões). A opção que define assinatura digital é superior a chaves secretas compartilhadas.

- **keyexchange=ike** : define o método de troca de chaves. O único valor aceito é ike, que define o protocolo IKE para a troca de chaves.

- **left=200.240.0.1** : define o IP da interface de rede do *Gateway* VPN esquerdo, a qual está conectada a rede pública (Internet).

- **leftsubnet=10.64.0.0/24** : parâmetro que define o endereço IP da rede privada por trás do *gateway* esquerdo. Deve ser informada também a máscara de rede, no caso **/24** (*netmask* 255.255.255.0).

- **leftnexthop=200.240.0.250** : próximo endereço depois do gateway da esquerda. Observando a topologia utilizada, percebemos que o próximo endereço IP após o gateway esquerdo é a interface do equipamento roteador.

- **right=200.240.1.1** : endereço IP da interface do *gateway* VPN da direita.

- **rightsubnet=10.64.1.0/24** : endereço da rede privada por trás do *gateway* direito.

- **rightnexthop=200.240.1.250** : próximo endereço depois do *gateway* da direita. Observando a topologia utilizada, percebemos que o próximo endereço IP após o gateway direito é a interface do equipamento roteador.

- **Auto=add** : especifica que operação deve ser feita automaticamente quando o IPSec for iniciado. As opções mais comuns são: ***start***, que inicia a conexão e ***add***, que apenas adiciona a conexão não a iniciando, indica que esta conexão pode ser inicializada através de linha de comando do IPSec. Para isso é possível utilizar os comandos seguintes:

```
# ipsec auto --up redea-redeb
# ipsec auto --down redea-redeb
```

  O primeiro comando inicializa a conexão chamada redea-redeb e o segundo comando fecha a conexão.

- **auth=esp** : define o tipo de protocolo IPSec que deve ser utilizado nos pacotes IP, ESP ou AH. Neste caso, define-se o *auth* está habilitando o cabeçalho ESP para os pacotes IP

- **pfs=no** : este parâmetro habilita ou não uma técnica chamada ***Perfect Forward Secrecy***, na qual a chave da Associação de Segurança IPSec será derivada por meio do algoritmo Diffe-Helman. Este parâmetro estando habilitado (pfs=yes) previne que um invasor, mesmo de posse da chave utilizada na primeira fase da negociação ISAKMP que estabelece a Associação de Segurança ISAKMP, não possa derivar a chave da SA IPSec. Porém, de acordo com (SILVA, 2003), algumas versões do FreeS/WAN podem não funcionar bem com o PFS, recomendando-se desabilitar este parâmetro caso algum problema esteja ocorrendo. Posteriormente, quando tudo estiver funcionando, pode-se habilitar novamente o PFS.

6.1.4.5 Configuração da Criptografia Oportunista

Como mencionado anteriormente, o FreeS/WAN pode utilizar um conceito chamado de ***Criptografia Oportunista*** (*Opportunistics Encryption*), onde qualquer *Gateway* VPN, implementado através do FreeS/WAN, pode criptografar seu tráfego, mesmo sem conhecimento e intervenção dos administradores de redes das entidades (*Gateways*) e mesmo sem nenhuma informação pré-configurada sobre a outra entidade (FREES/WAN, 2003). As informações de criptografia e autenticação necessárias ao estabelecimento das SAs IPSec são obtidas do Servidor DNS, o qual já possui mecanismo de busca de endereços IPs.

Portanto, é necessário que as informações de criptografia e autenticação sejam previamente configuradas no DNS. Além disso, os *gateways* VPNs devem habilitar a Criptografia Oportunista. Isso faz com que todo o processo aconteça automaticamente.

A Criptografia Oportunista possibilita uma redução significativa da sobrecarga administrativa em cima das conexões IPSec, uma vez que não são necessárias configurações específicas de túneis, apesar de ainda ser permitido configurá-los para casos específicos. Outra vantagem está no fato dela deixar o tráfego da Internet mais seguro, pois todo o conteúdo transmitido e recebido é criptografado.

Existem dois tipos de Criptografia Oportunista: a parcial e a total. A principal diferença está na forma que a conexão é feita. Na parcial, apenas é possível iniciar conexões, não sendo possível aceitar requisições de conexão, enquanto que na total, permite iniciar e aceitar conexões oportunistas.

Neste trabalho, foram configurados Servidores de DNS nos próprios *gateways* VPN. Foi definido o domínio ***vpn*** em ambos os DNS. Segue a seguir um trecho do arquivo ***/var/named/vpn.zone*** configurado para este experimento, o qual define os registros de recursos da zona vpn criada, demonstrando como foi publicado os registros de chaves públicas do *Gateway* VPN esquerdo (*leftserver*) e do *Gateway* VPN direito (*rightserver*). Os registros são do tipo TXT e estão destacados com o sombreamento.

```
@               IN      SOA     leftserver.  root.localhost (
                                2 ; serial
                                28800 ; refresh
                                7200 ; retry
                                604800 ; expire
                                86400 ; ttl
                )
        IN      NS      leftserver.
leftserver      IN      A       200.240.0.1
leftserver      IN      TXT   "X-IPsec-Server(10)=@leftserver.vpn
AQOLRoyDXEXMWB+m4smekuG1J+nju1qxCaQdeYs/9TaoMCa+8X1+eCR4io+UHzNrYlxQk
e1lkOLagIHe9JZcv/AFVEjeMciAqNNb/ucCpmwk9CFKLMZiJ9AYKEyCRM6CZ7ErDWDg4TO
K7H8wQR9qzjBmKe2OXl9WxT5J4j08Z1+9uw=="

rightserver     IN      A       200.240.1.1
rightserver     IN      TXT    "X-IPsec-Server(10)=@leftserver.vpn
AQPzzql5aXxbO5rVJQZCAeZ+3g1Vh2xlOOc76Jctva5J7pbAoNN3yr8lFzz1Pvbh7pka2OB+Xn
WCWF7zIbTyj+bxmdbWldXWooiCxrx7Q5qbJnmNpGhJl0ziN1oWcUmVgErZ3+kgXogvgrLoxp
c1nrvprybLGyLxTUaFXoVYwTyHzw=="

gwleft          IN      A       200.240.0.250
gwright         IN      A       200.240.1.250
...
```

Foi utilizado o seguinte comando do IPSec para gerar o registro de recurso TXT contendo a chave pública:

```
# ipsec showhostkey --txt @leftserver.vpn
```

Para configurar a Criptografia Oportunista completa foi necessário também incluir os registros de chaves públicas dos *Gateways* VPN no arquivo ***/var/named/0.240.200.in-addr.arpa***, arquivo do DNS Reverso criado para a rede 200.240.0.0/24. Abaixo, apresentamos um trecho onde demonstra a publicação do registro de chave pública do *Gateway* VPN *leftserver*.

```
...
@       IN      NS      leftserver.

1       IN      PTR     leftserver.

        IN      TXT     "X-IPsec-Server(10)=@leftserver.vpn
AQOLRoyDXEXMWB+m4smekuG1J+nju1qxCaQdeYs/9TaoMCa+8X1+eCR4io+UHzNrYlxQk
e1lkOLagIHe9JZcv/AFVEjeMciAqNNb/ucCpmwk9CFKLMZiJ9AYKEyCRM6CZ7ErDWDg4TO
K7H8wQR9qzjBmKe2OXl9WxT5J4j08Z1+9uw=="  leftserver.

250     IN      PTR     gwleft.
...
```

Cabe observar que todas as fontes bibliográficas consultadas para subsidiar a habilitação da Criptografia Oportunista foram bastante incipientes, não detalhando profundamente o assunto. É interessante ressaltar também que nenhum dos livros consultados utilizava a Criptografia Oportunista, o que dificultou demasiadamente a solução dos problemas identificados inicialmente, contribuindo para uma demora inesperada na configuração do ambiente proposto. De qualquer forma, constatou-se que uma das melhores fontes de consulta ainda era (FREES/WAN, 2003), mesmo assim, essa referência não possui informações detalhadas de como os registros de chaves devem ser publicados no DNS.

O Apêndice C apresenta o arquivo ***/etc/ipsec.conf*** utilizado para testar a Criptografia Oportunista.

6.1.4.6 Estabelecendo Conexões VPN

Em função dos problemas enfrentados no experimento citado quando das várias tentativas de conexão VPN realizadas sem sucesso e da falta de detalhamento que caracterizam inúmeras literaturas consultadas que discorrem sobre o assunto, observou-se que havia a necessidade de se estabelecer e documentar um conjunto básico de passos necessários e suficientes para o estabelecimento de conexões VPN (usando o FreeS/Wan) para efeito de futuras avaliações ou verificações dos testes realizados. Portanto, este trabalho definiu os seguintes passos (nem sempre obrigatórios) para o estabelecimento das conexões VPN, a saber:

1. Iniciar o Serviço do IPSec sempre que necessário.

   Para iniciar o serviço do FreeS/WAN deve-se utilizar o comando:

   ```
   # service ipsec start
   ```

2. Verificar a instalação do pacote FreeS/WAN.

   Após a instalação do pacote FreeS/WAN e da configuração de todos os arquivos de configuração já mencionados, inclusive os arquivos de configuração do DNS, é importante que se verifique se todos os componentes do FreeS/WAN estão rodando e funcionando adequadamente. Para isso, deve-

se utilizar o comando abaixo, após o serviço de IPSec estar inicializado (através do comando ***service ipsec start*** ou ***ipsec setup - -start***).

```
# ipsec verify
```

A janela abaixo apresenta a saída deste comando de verificação, o qual foi executado no servidor *Gateway VPN leftserver* do experimento.

```
# ipsec verify
Checking your system to see if IPsec got installed and started correctly:
Version check and ipsec on-path                                 [OK]
Linux FreeS/WAN 2.03
Checking for KLIPS support in kernel                            [OK]
Checking for RSA private key (/etc/ipsec.secrets)               [OK]
Checking that pluto is running                                  [OK]

Opportunistic Encryption DNS checks:
Looking for TXT in forward map: leftserver                      [OK]
Does the machine have at least one non-private address?         [OK]
Looking for TXT in reverse map: 1.0.240.200.in-addr.arpa.       [OK]
```

Como pode ser observado, este comando verifica o funcionamento do KLIPS e do PLUTO, a presença de chave RSA no arquivo /etc/ipsec.secrets e as configurações de DNS para implementação da Criptografia Oportunista.

3. Iniciar uma conexão VPN

Este passo só é necessário se a conexão tiver sido definida com o parâmetro *auto=add*, pois com a opção *start* a conexão já é iniciada automaticamente. O formato do comando abaixo inicia uma conexão:

```
# ipsec auto --config <arq-configuracao> --up <conexão>
```

Caso não seja especificado o arquivo de configuração (opção --config), o arquivo padrão adotado é o ***/etc/ipsec.conf***. Desta forma, para iniciar as conexões criadas em ***/etc/ipsec.conf***, os comandos executados foram:

```
# ipsec auto –-up redea-gwr
```

```
# ipsec auto –-up gwl-gwr
```

A janela abaixo apresenta a saída dos comandos de inicialização das conexões ***redea-gwr*** *e* ***gwl-gwr***, nesta ordem, que foram realizados no experimento.

```
# ipsec auto --up redea-gwr
104 "redea-gwr" #3: STATE_MAIN_I1: initiate
106 "redea-gwr" #3: STATE_MAIN_I2: sent MI2, expecting MR2
108 "redea-gwr" #3: STATE_MAIN_I3: sent MI3, expecting MR3
004 "redea-gwr" #3: STATE_MAIN_I4: ISAKMP SA established
112 "redea-gwr" #4: STATE_QUICK_I1: initiate
004 "redea-gwr" #4: STATE_QUICK_I2: sent QI2, IPsec SA established
{ESP=>0xe33cf35b <0xd24d7e81}

# ipsec auto –up gwl-gwr
112 "gwl-gwr" #5: STATE_QUICK_I1: initiate
004 "gwl-gwr" #5: STATE_QUICK_I2: sent QI2, IPsec SA established
{ESP=>0xe33cf35c <0xd24d7e82}
```

A saída desse comando apresenta claramente que na primeira conexão (***redea-gwr***) ocorreram dois momentos importantes, o primeiro foi o estabelecimento da SA ISAKMP, pois ainda não existia nenhuma Associação de Segurança ISAKMP entre os dois gateways; a segunda conexão foi o estabelecimento da SA IPSec. Neste, a SA ISAKMP já estava estabelecida.

4. Reiniciar o Serviço do IPSec sempre que necessário.

   É necessário que após quaisquer mudanças nos arquivos ***/etc/ipsec.conf*** e ***/etc/ipsec.secrets*** se reinicialize o serviço IPSec. O melhor a fazer é parar a execução do FreeS/WAN e logo em seguida iniciar novamente a execução do mesmo. Para fazer isso, pode-se utilizar os seguintes comandos:

   ```
   # ipsec setup –stop
   # ipsec setup --start
   ```

5. Adicionar as conexões definidas no arquivo ***/etc/ipsec.conf*** para as tabelas de conexões.

Esse passo deve ser sempre executado quando o FreeS/WAN fornece um erro informando que determinada conexão não existe. Geralmente acontece este erro quando se define uma configuração nova ou se muda o nome de uma conexão no arquivo ***/etc/ipsec.conf***, e não se adiciona na tabela a conexão. Para realizar essa tarefa deve-se utilizar o seguinte comando:

```
# ipsec auto --config <arq-configuracao> --add <conexao>
```

Caso não seja especificado o arquivo de configuração (opção --config), o arquivo padrão adotado é o ***/etc/ipsec.conf***. Como o trabalho utilizou duas conexões, a **redea-gwr** e **gwl-gwr** (conforme pode ser observado no arquivo ***/etc/ipsec.conf*** apresentado) foi necessário se adicionar as duas conexões. Esta tarefa foi feita através dos comandos:

```
# ipsec auto --add redea-gwr
# ipsec auto –-add gwl-gwr
```

6.1.5 Testes realizados

Como mencionado anteriormente, foi postada uma estação rodando o *software Sniffer **Pro***, versão 4.70, da *Network Associates*, para monitorar o fluxo de comunicação no entre as duas redes, objetivando acompanhar e comprovar o estabelecimento de seção ISAKMP, IPSec e da comunicação criptografada. Esta estação foi postada entre o roteador e um dos servidores VPN, no caso, o ***leftserver***. Alguns relatórios de confirmação foram exportados e estão presentes no **Apêndice D** deste trabalho. A figura 6.2 apresenta um *print screen* de parte do monitoramento de um estabelecimento de uma SA.

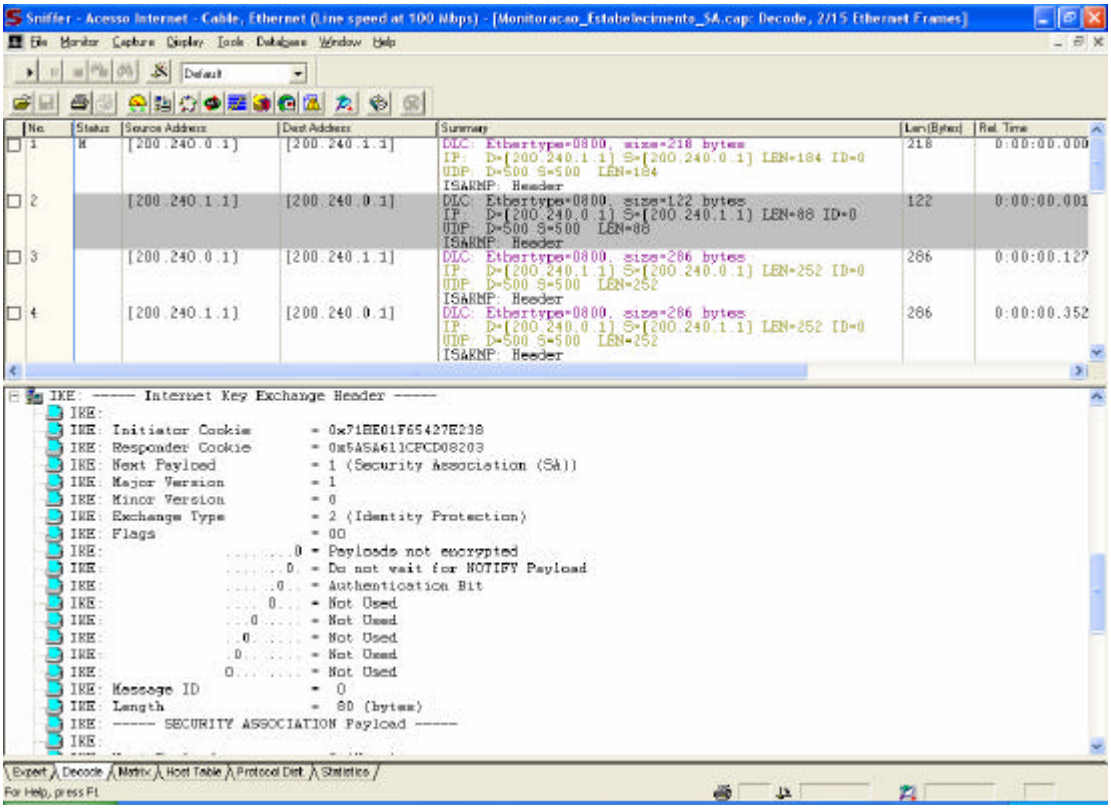


Figura 6.2 – Tela do *Sniffer Pro* no Estabelecimento de SA entre os *gateways* VPN leftserver e rightserver

Outra situação avaliada com o *Sniffer*, que também comprovou a funcionalidade da solução adotada, foi o monitoramento realizado no estabelecimento e utilização de uma conexão *Telnet* entre os dois *gateways* VPN. A figura 6.3 apresenta um *print screen* do início do estabelecimento de uma seção Telnet. Pode-se perceber que o *payload* do pacote selecionado está criptografado, assim como todos os pacotes transmitidos nesta seção, e que os pacotes IP estão encapsulando o protocolo ESP (modo túnel).

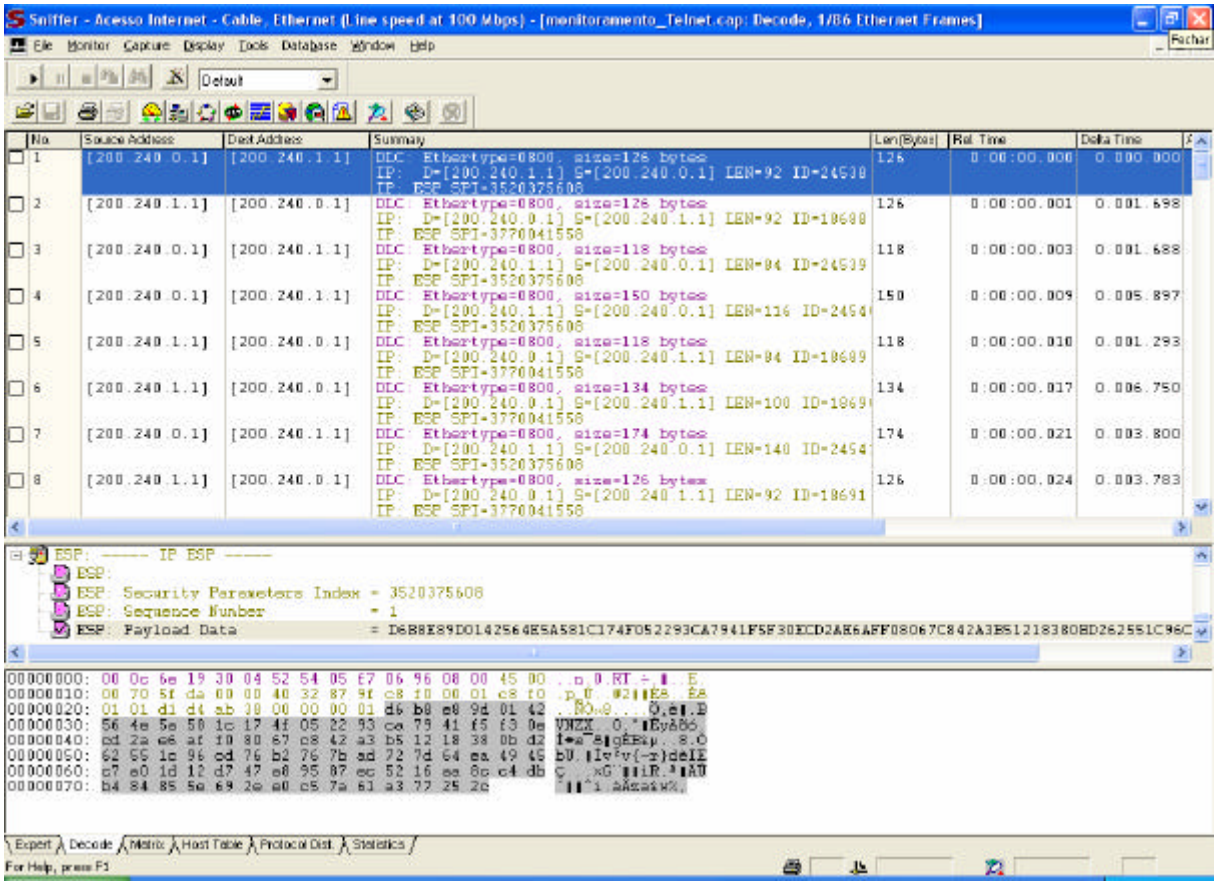


Figura 6.3 - Tela do *Sniffer Pro* durante uma seção *Telnet* entre os servidores VPN leftserver e rightserver

Também foram realizados testes que avaliaram o desempenho da solução. Houve verificação nos tempos de respostas dos pacotes em uma conexão VPN. Constatou-se que os tempos de resposta estavam bem próximos as de um cenário sem conexão VPN (sem criptografia). Porém, cabe se realizar testes mais apurados, com períodos mais longos de avaliação, com pacotes de tamanho maior, cenários mais aleatórios de utilização e com ferramentas mais adequadas para este fim.

A figura 6.4 apresenta um *print screen* de parte do monitoramento de utilização de um *Ping* (com a utilização de pacotes de 150 bytes). A resposta de um ping por si mesma já demonstra que a conexão VPN está funcionando adequadamente. Nesta, podemos perceber, através de uma análise da coluna "*Real time*", que os tempos de resposta, neste caso, estão na faixa de centésimos de segundos, isto é, dentro da normalidade. Obviamente, o cenário adotado não é real, porém, já podemos inferir que existem grandes chances da solução apresentada não apresentar problemas sérios de desempenho.

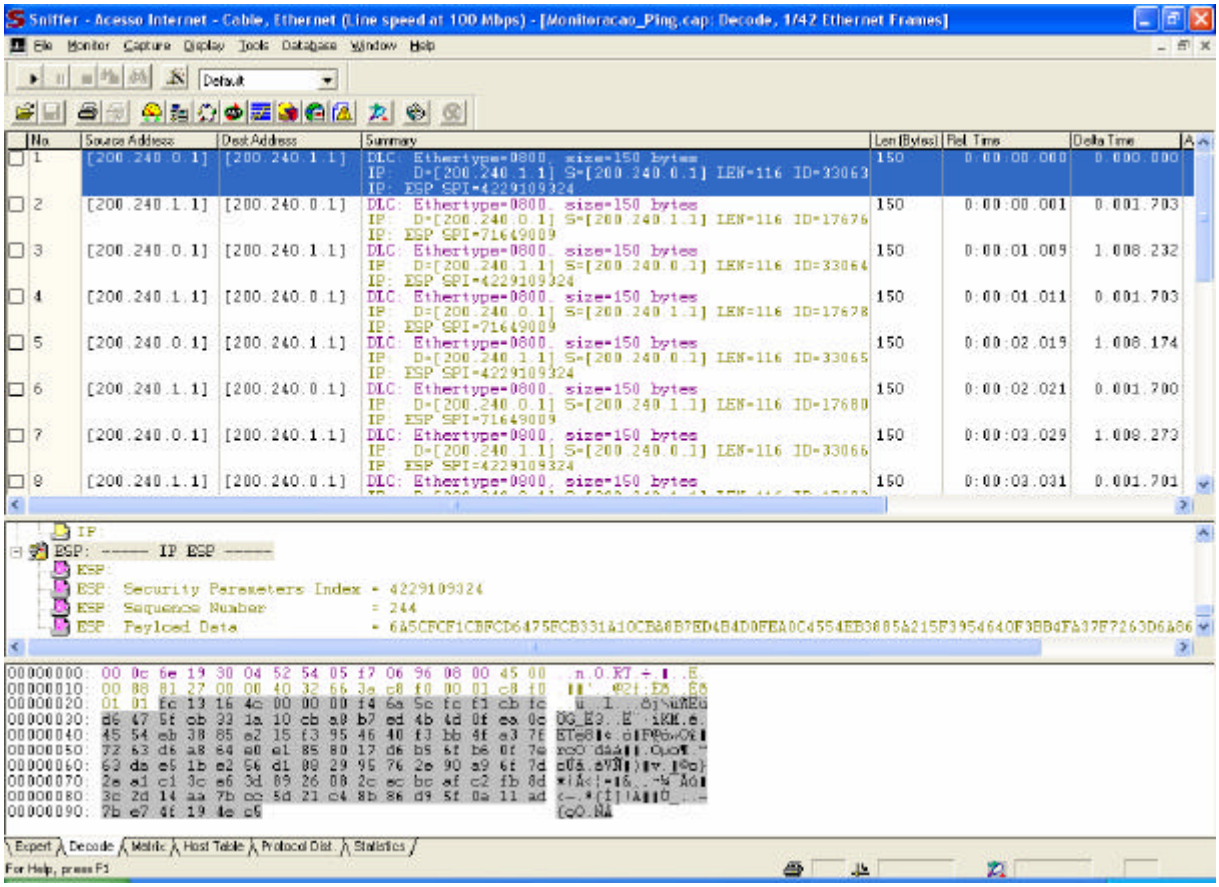


Figura 6.4 - Tela do *Sniffer Pro* durante a utilização do *Ping* entre servidores VPN leftserver e rightserver

No **Apêndice D** foram incluídos alguns relatórios gerados pelo *Sniffer Pro* durante os testes realizados, assim como são apresentados resultados de testes realizados com a ferramenta ***tcpdump*** e análise das saídas de comandos do ***ipsec*** (opção *look*).

6.2 Implementação do Serviço de Firewall

Diante das características e opções de posicionamento de um *Firewall* em relação ao Servidor VPN já abordadas, este trabalho optou por implementar o *firewall* integrado ao *gateway* VPN, principalmente, devido ao atendimento dos pré-requisitos básicos de segurança já discutidos, a sua facilidade de gerenciamento e a indisponibilidade de outros equipamentos para uso dedicado. O experimento realizado neste trabalho não realizou análise de outros posicionamentos do servidor VPN em função ao *firewall*.

A ferramenta proposta e utilizada como *firewall* neste trabalho foi a ***IPTables***, ferramenta de filtro de pacotes existente no *kernel* do Linux 2.4.X[16]. A ferramenta *IPtables* insere e retira regras da tabela de filtragem de pacotes do *kernel*, funcionando por meio de regras estabelecidas na inicialização do sistema operacional. Todos os pacotes que entram no *kernel* do linux são analisados. As *chains* (correntes) são as situações possíveis dentro do *kernel*. Quando um pacote entra no *firewall*, o *kernel* analisa qual o destino do pacote e decide qual *chain* o manipulará. Este processo comumente é chamado de *roteamento interno* (FERREIRA, 2003). Os tipos de *chains* dependerão da tabela do *Iptables* que está sendo utilizada no momento. Verificaremos a seguir, que existem três tabelas possíveis e que o *IPtables* fornece uma interface para que o usuário possa manipular os filtros de pacotes do *kernel*.

As *chains* determinarão se a regra será aplicada quando o pacote entra, sai ou é redirecionado por um NAT. As *chains* previamente configuradas no *kernel* 2.4.X estão descritas abaixo.

Chain	*Descrição*
INPUT	Verifica os pacotes que tentam entrar na rede interna.
OUTPUT	Verifica os pacotes que tentam sair da rede interna.
FORWARD	Verifica todos os pacotes que atravessam a rede.
PREROUTING	Analisa todos os pacotes que entram no *firewall* para sofrerem NAT. Ele realiza ações de NAT com o endereço de destino do pacote (*Destination* NAT).
POSTROUTING	Analisa todos os pacotes que saem do *firewall* para sofrerem NAT. Ele realiza ações de NAT com o endereço de origem do pacote (*Source* NAT).

Observa-se que novas *chains* podem ser criadas e excluídas, com exceção das *chains* apresentadas que não podem ser apagadas.

A figura 6.5, apresenta um esquema onde são representadas as três *chains* consideradas padrão. Quando um pacote atinge uma das *chains*, esta examina o seu cabeçalho e, baseada em sua tabela de regras, decide o que fazer com o pacote. Se a *chain* determina que

[16] Em versões anteriores do kernel do linux era comum a utilização da ferramenta Ipchains, porém, a ferramenta Iptables a substitui com substanciais recursos adicionais.

seja bloqueado, o pacote é descartado neste momento, caso contrário, o pacote segue conforme o diagrama (ACCEPT).

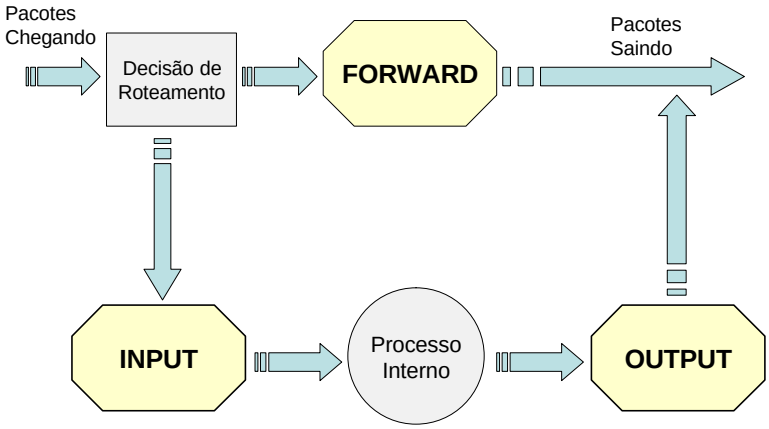


Figura 6.5 – Esquema de fluxo das *chains* padrões

As três tabelas utilizadas para definir quais tipos de *chains* serão utilizadas são:

- Tabela **FILTER**: É a tabela padrão, sendo utilizada quando não houver tabela especificada. Utilizada quando há tráfego normal de dados, sem a utilização de NAT (*Network Address Translation*). Utiliza as *chains* INPUT, OUTPUT e FORWARD, as quais serão descritas posteriormente.

- Tabela **NAT**: É utilizada quando existe NAT. Utiliza as *chains* PREROUTING, POSTROUTING e OUTPUT, as quais também serão descritas posteriormente.

- Tabela **MANGLE**: É utilizada para efetuar alterações especiais em pacotes. Utiliza as chains PREROUTING e OUTPUT.

Para melhor entendimento das regras a serem definidas em nosso ambiente VPN, faz-se necessário apresentar a sintaxe do Iptables, a saber:

```
iptables [-t tabela] <comando> <chains> [opção<parâmetro>] <destino>
```

Sendo que os principais comandos utilizados neste trabalho foram:

Comando	*Ação*
-A	Acrescenta uma nova regra a um *chain*
-D	Apaga uma regra de um *chain*
-F	Apaga todas as regras
-L	Lista o estado do Iptables

As principais opções utilizadas na elaboração das regras do *firewall* foram:

Opção	Descrição
-p	refere-se ao protocolo que deve ser verificado. Pode ser o número do protocolo ou um nome retirado de */etc/protocols*.
-s	refere-se aos dados da rede ou host de origem. Os seus parâmetros podem ser <endereço IP> / <máscara da sub-rede>.
-d	refere-se aos dados da rede ou host de destino. Os seus parâmetros podem ser <endereço IP> / <máscara da sub-rede>.
-i	especifica a interface de rede de entrada.
-o	especifica a interface de rede de saída.
--sport	especifica a porta de origem. Funciona apenas para as opções *–p udp* e *–p tcp*.
--dport	especifica a porta de destino. Também só funciona apenas para as opções *–p udp* e *–p tcp*.
-j	define o destino de uma regra.

Outro parâmetro exigido na sintaxe do Iptables é o <destino>. Este é o campo que definirá qual ação deverá ser tomada com o pacote caso a regra seja satisfeita para o pacote. Os destinos utilizados neste trabalho estão listados abaixo.

Destino	Ação
ACCEPT	Permitirá a passagem do pacote.
REJECT	Não permitirá a passagem do pacote.
REDIRECT	Redireciona uma requisição para uma porta local do *firewall* (utilizada para operação do Proxy).

Outros comandos, opções e destinos do *Iptables* podem ser consultados em (NEMETH; SNYDER; HEIN, 2002), (RUSSEL, 1999) ou (FERREIRA, 2003).

6.2.1 Configuração do Firewall para uso em VPNs

A configuração das regras no Iptables para a utilização de VPNs é relativamente simples. Precisa-se apenas permitir acesso à porta 500, que é a porta utilizada pelo IKE para negociação de chaves.

Uma vez que utilizamos IPSec, necessita-se também regras que aceitem os protocolos ESP (protocolo 50) e o AH (protocolo 51) para que os mesmos também não sejam barrados.

Para tanto, utilizamos o script abaixo descrito.

```
# Para possibilitar a negociação IKE na eth0 e ipsec0, liberando a porta 500 (IKE)
# na chain de entrada e de saída
/sbin/iptables -A INPUT -p udp -i eth0 --sport 500 --dport 500 -j ACCEPT
/sbin/iptables -A OUTPUT -p udp -o eth0 --sport 500 --dport 500 -j ACCEPT
/sbin/iptables -A INPUT -p udp -i ipsec+ --sport 500 --dport 500 -j ACCEPT
/sbin/iptables -A OUTPUT -p udp -o ipsec+ --sport 500 --dport 500 -j ACCEPT

# Possibilita o tráfego de pacotes ESP (protocolo 50) pelas duas interfaces
/sbin/iptables -A INPUT -p 50 -i eth0 -j ACCEPT
/sbin/iptables -A OUTPUT -p 50 -o eth0 -j ACCEPT

# Possibilita o tráfego de pacotes AH (protocolo 51) pelas duas interfaces
/sbin/iptables -A INPUT -p 51 -i ipsec+ -j ACCEPT
/sbin/iptables -A OUTPUT -p 51 -o ipsec+ -j ACCEPT
```

6.3 Gestão de Segurança nos Roteadores

Podemos considerar que o roteador é um dos componentes de rede mais importantes dentro do contexto de segurança e conectividade de uma rede corporativa, bem como, é um dispositivo que participa da Defesa em Profundidade, principalmente quando tratamos de VPNs corporativas baseadas na Internet, pois o roteador é o primeiro elemento de interface com esta rede pública não confiável, isto é, o primeiro elemento passível de ataques. Desta forma, uma das propostas deste trabalho também é de definir algumas normas e procedimentos a serem adotados na configuração dos Roteadores[17] corporativos que fazem parte do Perímetro Externo da rede, assim como no uso de tecnologia de *software* e de *hardware* a fim de minimizar a probabilidade de um ataque bem sucedido, tendo como premissa básica que deve-se utilizar todo e qualquer meio ou mecanismo de segurança para

[17] Como toda a rede da Previdência utiliza roteadores CISCO, a sintaxe dos comandos e os procedimentos adotados/mostrados neste trabalho serão específicos para os equipamentos desta marca.

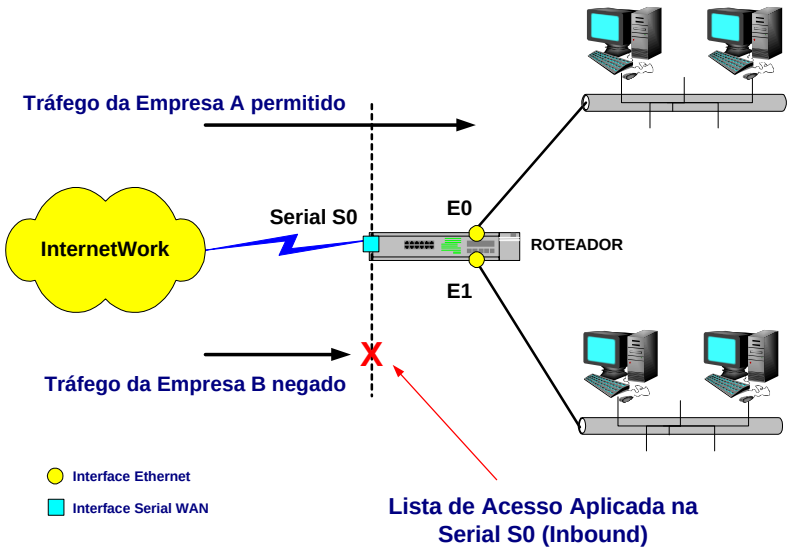
limitar, e se possível impedir, que estes ataques aconteçam, mesmo que estes métodos venham a se sobrepor.

O Roteador utilizado no experimento foi um CISCO 2500, disponível no *site* da Dataprev do Amazonas, o qual já estava configurado para acesso a Internet através de um *link* de 64Kbps com a Embratel (na sua porta WAN). Os testes realizados tiveram o intuito de validar as alterações e configurações de roteamento propostas (descritas nas seções seguintes). Todos os testes foram feitos através de conexão discada a um Servidor de Acesso Remoto (RAS) da 3COM (*Total Control 3Com*) conectado diretamente a esse roteador.

6.3.1 Uso das Access Control Lists

Uma *Access Control Lists* - ACL, ou Lista de Controle de Acesso, é um conjunto de regras de acesso que são utilizadas para o filtro de tráfego (permissão e proibição de pacotes) nas interfaces de Roteador. Nestas listas são definidos os critérios de filtragem de pacotes de uma determinada/apropriada interface. Com isso, o roteador é capaz de inspecionar o fluxo/tráfego na interface e rejeitar os pacotes que não são permitidos pela sua *access list*.

Os pacotes são filtrados tanto no sentido ***Inbound*** (da rede externa para a interna), como no sentido ***Outbound*** (da rede interna para externa). A figura seguinte mostra uma aplicação do uso de ***access list*** que inspeciona o tráfego de entrada (*Inbound*) na interface Serial 0, cuja lista possui propósito de negar o tráfego da Empresa B.



Fonte: Adaptado de (LEE, 1999, p. 184)

Figura 6.6 - Aplicação de Access List no Roteador

O uso de ***access list*** como ferramenta de segurança é importantíssimo, pois define a primeira camada de segurança de uma rede corporativa conectada na Internet. Desta forma, é necessário que se aplique este recurso de segurança a todos os Roteadores das redes conectadas, de acordo com as necessidades de comunicação de cada *Site*. Porém, deve-se adotar um padrão mínimo de segurança, a ser implementado através de ***access list*** em cada roteador da rede. Assim, este trabalho pretende definir um padrão que deve ser adotado em cada roteador a fim de garantir um mínimo de segurança contra ataques. A seguir, definimos os procedimentos de configuração de roteador a serem adotados.

Antes de se propor o conjunto de regras padrão do *access list*, deve-se analisar alguns pontos sobre a definição e criação de listas de acesso que serão úteis para o entendimento desta proposta, a saber:

- Uma *Access List* é aplicada para uma determinada *Interface*, em um determinado sentido (***Inbound*** ou ***Outbound***). Pode-se aplicar a mesma *Access List* para os dois sentidos.

- Pode-se aplicar a mesma *Access list* para múltiplas Interfaces.

- As *Access List* são identificadas por números que devem ser únicos em cada roteador.

- As regras são processadas na ordem seqüencial. O roteador irá comparar o pacote com a lista, iniciando sempre pela primeira regra até que encontre uma regra que estabeleça alguma relação com o pacote (permitir ou negar)

- Após permitir ou negar o pacote, o roteador pára o processamento da *Access List*.

- Caso exista *Access List* na Interface, todo pacote será comparado com no mínimo uma regra, que é a última regra e é a ***default*** : que **NEGA** todos os pacotes. Esta regra é chamada ***deny-any***.

Uma boa literatura para um estudo completo e minucioso sobre ***Acccess Lists***, como por exemplo, sua sintaxe, opções e funcionalidades é o material de Certificação CCNA da Cisco (CISCO, 2000) ou (LEE, 1999), livro escrito por Donal Lee, Engenheiro de Sistemas Sênior da *Cisco Systems*, que discorre aspectos bem interessantes sobre questões de segurança nos roteadores da CISCO.

Observa-se ainda que este trabalho não objetiva explorar todas as funcionalidades das *Access Lists*, somente, aquelas mais relevantes para a construção e utilização de VPNs em Empresas que exigem um nível apropriado de segurança para este serviço, dentro de um contexto de Defesa em Profundidade.

6.3.2 Procedimentos de Segurança para Acesso ao Roteador

Existem duas formas de se acessar o Roteador, uma é através de sua porta CONSOLE, ao qual exige uma conexão direta com a interface serial do Computador com esta porta, através de Cabo ***Cross*** e adaptador RJ45-Serial. Do ponto de vista de segurança, para este tipo de acesso, deve-se preocupar com o acesso físico de pessoas no ambiente onde está instalado o roteador, impedir que pessoas não autorizadas tenham acesso a este equipamento é essencial.

Outra forma de acesso ao roteador é através de uma seção TELNET. Este tipo de acesso além de ser muito comum é bastante preocupante, pois, se não tratado, qualquer usuário conectado na rede Internet, mesmo não fazendo parte da VPN, mas tendo conhecimento do endereço IP do roteador, pode fazer tentativas de abrir uma seção Telnet com o Roteador e, conseqüentemente, tentar romper a segurança da rede. Diante disso, deve-se:

- impedir a abertura de seções **Telnet** para determinados pontos de rede ou pessoas não autorizadas;

- configurar uma senha para proteger o nível do administrador (***enable mode***) do roteador;

- não utilizar senhas óbvias ou fáceis de se descobrir.

A seguir, demonstraremos como realizar o bloqueio do acesso a um roteador através de Telnet.

6.3.2.1 Restringindo o Acesso ao Roteador por TELNET

Para impedir a abertura de seção TELNET nos Roteadores que participam da VPN, propõe-se aplicar uma ***access list*** para todas as cinco interfaces lógicas (linhas de terminais virtuais) que estão disponíveis para ***Telnet*** (em um roteador CISCO), restringindo assim, quem poderá realizar *Telnet*. Para isso, os comandos que devem ser inseridos no roteador são:

```
ROUTER# conf t
ROUTER (config) # access-list N permit  A.B.C.D  X.Y.Z.W
ROUTER (config) # line vty 0 4
ROUTER (config-line) # access-class N in
```

A primeira linha entra na configuração do Router e a segunda define uma regra para a *access-list* de número **N** (que deve estar entre 10 e 99 – listas *defaults*), onde A.B.C.D é o endereço da rede e X.Y.Z.W é a máscara que definirá quais os endereços desta rede terão acesso via Telnet ao Roteador.

A terceira linha diz ao roteador que se deseja configurar simultaneamente todas as linhas disponíveis para ***telnet*** e a quarta linha associa as regras impostas na ***access list*** com as 5 (cinco) linhas de terminais virtuais disponíveis. A expressão **in** na última linha significa que se deseja filtrar o fluxo de entrada.

Por exemplo, para permitir que todos os endereços da rede 10.64.1.0 acessem o roteador via *telnet*, a ***access list***, como sugestão 10, deve ser definida da seguinte forma:

```
ROUTER (config) # access-list 10 permit  10.64.1.0  0.0.0.255
```

E a associação com as linhas de terminais deverá ser:

```
ROUTER (config-line) # access-list 10 in
```

Deve-se lembrar que toda *access-list* contém a regra de negar todo acesso que não seja permitido nas regras explícitas na *access-list* (*deny-any*) como já comentado anteriormente. No exemplo acima, será negado o acesso a qualquer estação que não seja da rede 10.64.1.0.

A definição desta *Access-list* impedirá que um *host* de endereço desconhecido acesse o roteador por Telnet. Isto é importantíssimo quando tratamos de redes VPN baseadas na Internet, cujos roteadores de perímetro exterior estão diretamente conectados na Internet.

6.3.2.2 Restringindo acesso ao Roteador pela Console

Além da Restrição física do local da instalação do Roteador a pessoas não autorizadas, deve-se configurar uma senha para o acesso via Console para desencorajar um *hacker* casual. Para configurar a senha de acesso pela *console*, deve-se proceder com o seguinte comando:

```
ROUTER # conf t
ROUTER (config) # line console 0
ROUTER (config-line) # password <senha>
```

Onde <senha> deve ser substituída pela senha a ser definida por cada Administrador da rede.

6.3.3 Procedimentos para Combate ao IP Spoofing Attack

O *Spoofing Attack* é um dos tipos mais populares de ataques. Ele consiste na troca do IP original por um outro IP que é reconhecido na rede como um *host* confiável. Isto permite com que *hackers* ou *crackers* possam fingir que estão em computadores confiáveis da rede atacada, e desta forma se aproveitarem disso para entrar em máquinas desta rede e no próprio roteador através de TELNET ou RLOGIN.

Assim, para se evitar este tipo de ataque, pretende-se propor alguns ***filtros anti-spoofing*** que podem e devem ser utilizados e configurados nos roteadores que se interligam a Internet, principalmente nas redes corporativas que fazem uso de VPNs.

Para exemplificarmos como deve ser feita esta proteção, através de ***filtros anti-spoofing,*** vamos supor que uma das redes da Dataprev no Amazonas tenha o seguinte endereço: 200.241.114.0. Desta forma, para evitarmos que exista a possibilidade de um ***Spoofing Attack***, temos que **impedir** que qualquer computador em uma rede externa, mesmo que possua um endereço IP considerado confiável, acesse o roteador pela SERIAL. Exceção feita apenas para o endereço IP do Gateway VPN. Desta forma, minimiza-se a probabilidade de ataques de *Spoofing Attack*. Para isso, podemos executar os seguintes procedimentos (nesta ordem):

1. Estando na Configuração da Serial (com o uso do comando **in,** Ex: **in S0**), associe-a a um *access list* que se tenha definido para o tráfego ***Inbound***. Supondo que a *access list* definida seja 101, procede-se ao seguinte comando:

   ROUTER (config-if) # **ip access-group 101 in**

   Obs: Não é necessário este passo se já existe *access list Inbound* para a Serial

2. Uma vez que não é esperado que pacotes oriundos da Internet (vindos através da Interface Serial) cheguem no roteador com endereço de origem sendo um endereço interno (com exceção ao endereço do *Gateway* VPN), deve-se negar o acesso a todos os endereços confiáveis de rede através da Serial, a fim de se evitar o ***Spoofing Attack*** (trapaça de endereço). Para isto, faz-se necessário o seguinte comando:

   ROUTER (config) # ***access-list*** **101 deny ip 200.241.114.0 0.0.0.255 any**

3. Alguns ataques ***Spoofing*** usam endereços de ***loopback*** no campo de endereço origem do pacote, assim os *hackers* não necessitam saber o endereçamento interno da rede. Isto pode ser evitado, criando-se uma regra no *access-list* que impeça que pacotes que iniciem com o octeto 127 - endereço de *loopback* para *hosts* - entrem através da Serial. O comando para criação desta regra está definido abaixo.

   ROUTER (config) # ***access-list*** **101 deny ip 127.0.0.0 0.255.255.255 any**

4. Deve-se liberar acesso ao endereço IP do *Gateway* VPN que fará tunelamento através do roteador. O comando para criação desta regra está descrito abaixo:

   ROUTER (config) # ***access-list*** **101 permit ip <endereço IP do Gateway VPN> 0.0.0.0**

   Onde <endereço IP do *Gateway* VPN> é o endereço do servidor VPN da rede remota que pretende se estabelecer um túnel VPN através do roteador.

5. Como última regra, deve-se liberar o acesso IP para todos aqueles pacotes que não coincidem com as regras impostas anteriormente, pois as regras são checadas de forma seqüencial. Desta forma, os pacotes que não possuem natureza do *Spoofing Attack* (disfarce através do uso de um endereço confiável) ou não são pacotes originários do servidor VPN são liberados. Isto é feito através do comando:

   ROUTER (config) # ***access-list*** **101 permit ip any any**

A figura 6.3 demonstra o mecanismo de *Anti-Spoofing* implementado através do uso de ***Access list*** para o aumento da segurança contra ataques de *hackers* ou *crackers*.

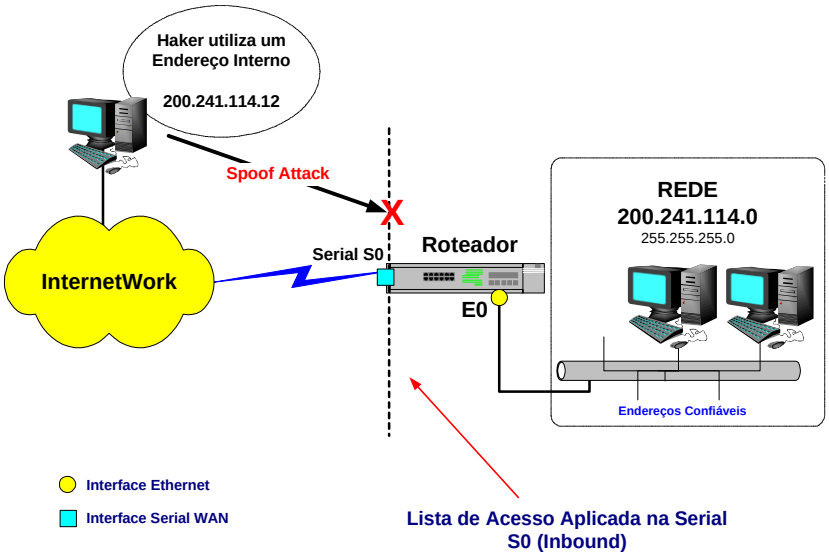


Figura 6.7 - Ação da *Access List* contra o IP *Spoofing Attack*

6.3.4 Desabilitando Alguns Serviços dos Roteadores CISCO para aumento de Segurança

Os procedimentos seguintes possuem como objetivo promover um aumento do nível de segurança de Rede, sendo sugeridos principalmente para serem executados nos Roteadores que estão conectados a redes não confiáveis como é o caso dos Roteadores do Perímetro Externo (Acesso a Internet).

Com o objetivo de propiciar maior segurança às redes que utilizam roteadores Cisco conectados a Internet, alguns serviços TCP/IP devem ser desabilitados para que não sejam utilizados como fontes de informação para ataques. Estes serviços são (LEE, 1999):

- *Desabilitar o ICMP*
- *Diagnósticos UDP e TCP*
- *IP Source Routing*
- *O Cisco Discovery Protocol – CPD em links públicos*
- *Broadcasts Diretos nas Interfaces*

6.3.4.1 Desabilitar o ICMP

É muito importante desabilitar o protocolo ICMP, pois o transbordamento (estouro de *buffer*) por ICMP é muito usado para ataques de DDoS. Um invasor pode criar processos em *background* em várias maquinas na Internet e fazer com que essas máquinas mandem pacotes ICMP para o roteador. Este é um dos ataques mais comuns usados na Internet. Isto, provavelmente, causaria uma perda de disponibilidade no roteador. Para desabilitar o ICMP para redes remotas você deve configurar seu roteador como segue abaixo:

```
ROUTER (config) #  access-list 101 deny icmp any any
```

6.3.4.2 Desabilitar Serviços de Diagnósticos UDP e TCP

Comumente, os roteadores vêm de fábrica com alguns serviços de diagnóstico TCP e UDP (chamados *small servers*) habilitados por *default*. Entretanto, estes serviços podem ser utilizados por *hackers* para obter informações que possibilitem facilidades nas tentativas de invasão e, por conseguinte, a sua ativação torna o sistema mais frágil a invasões. Desta forma, estes serviços devem ser desabilitados. Em roteadores CISCO, isto pode ser feito usando-se os comandos globais de configuração:

```
ROUTER (config) # no service tcp-small-servers
ROUTER (config) # no service udp-small-servers
```

6.3.4.3 Desabilitar o IP Source Routing

Este serviço possibilita que uma máquina em uma rede não confiável (Internet) possa ser um nó intermediário (*hop* intermediário) no caminho de Roteamento entre duas máquinas de uma subnet que tenham endereços confiáveis uma para outra (geralmente da mesma rede). Desta forma, um hacker consegue atacar a máquina de endereço destino do pacote.

Assim, para desabilitar este serviço, que é raro de ser usado (geralmente usado para *troubleshooting*), procede-se com o seguinte comando:

```
ROUTER (config) # no ip source-route
```

6.3.4.4 Desabilitar o CDP

O *Cisco Discovery Protocol* – CDP fornece informações sobre outros Roteadores CISCO que estão na vizinhança de um Roteador CISCO. Tal informação é utilizada para fornecer a topologia da rede e pode ser utilizada também por um *hacker*; desta forma, deve-se impedir que tal serviço esteja disponível para a Interface conectada a uma rede não confiável. O comando para realizar este bloqueio está a seguir e deve ser executado na configuração da serial desejada.

```
ROUTER (config-if) # no cdp enable
```

6.3.4.5 Desabilitar o *Broadcast* Direto nas Interfaces

O ***Broadcast*** direto é um *broadcast* enviado através de um roteamento de rede para uma determinada ***Subnet***. Quando este ***broadcast*** alcança a *subnet*, o roteador realiza um *broadcast* para todos os *hosts* daquela *subnet*. Caso, haja um fluxo de avalanches/rajadas de *broadcasts*, geradas por *hackers*, por exemplo, o roteador poderá sofrer com a sobrecarga gerada. Desta forma, deve-se desabilitar este serviço nos roteadores que fazem conexão com a Internet e com Instituições Externas para se reduzir a possibilidade de um ataque através destes Broadcasts (por sobrecarga). O comando abaixo efetiva esta operação e deve ser executado na Interface ***Ethernet*** das redes Internas.

```
ROUTER (config-if) # no ip directed-broadcast
```

7 CONCLUSÕES

Ao fim deste trabalho conseguimos elaborar uma proposta de um modelo de segurança para interligação de redes corporativas através de VPNs baseadas na Internet, bem como, conseguimos implementar um experimento de uma infra-estrutura de VPN utilizando sistemas e ferramentas livres. Analisamos e discutimos diversas topologias de rede e serviços combinados que podem ser utilizados em uma infra-estrutura VPN. Ao final, chegamos a conclusão que nenhuma proposta, por melhor estruturada que seja, irá satisfazer totalmente aos requisitos de segurança, qualidade e financeiros de uma Empresa. Porém, este trabalho demonstrou que a utilização cuidadosa e combinada de serviços de segurança, como por exemplo, *firewalls*, *proxys*, sistemas de detecção de Intrusos, listas de controles de acesso, entre outros, em uma topologia que favoreça a Defesa em Profundidade, pode proporcionar um aumento sensível de segurança.

A implementação de uma infra-estrutura VPN, realizada neste trabalho, proporcionou respostas aos questionamentos que, a princípio, não foram completamente obtidos mediante simples pesquisa e observação da literatura corrente, mesmo porque, a maior parte das referências bibliográficas disponíveis traziam abordagens específicas e isoladas, não descrevendo resultados do uso de VPNs em conjunto com soluções e produtos de *software* livre para a construção de redes VPN, nem tão pouco, os problemas e as dificuldades encontradas na sua implementação. Portanto, fez-se necessário a implementação e avaliação de uma solução VPN experimental para simular uma VPN baseada na Internet.

Esse processo de investigação, conferido a este trabalho, garantiu respostas aos questionamentos de pesquisa levantados, assim como, a confirmação de suas hipóteses, na medida em que:

- Definiu-se o IPSec como o conjunto de protocolos mais adequado para suportar soluções VPN entre redes corporativas. Descobriu-se que este é um dos protocolos mais bem sucedidos, aceito por grande número de especialistas e com inúmeras aplicações já o suportando. Identificou-se que a tendência de mercado é se estabelecer mecanismos de segurança na camada de rede, mesmo porque, esse conceito não impede que sejam inseridos serviços de segurança em outras camadas de rede.

- Identificou-se que apesar de sabermos que uma solução VPN baseada na Internet aplica princípios de tunelamento e criptografia, os quais podem proporcionar serviços de integridade, privacidade e autenticidade durante uma conexão VPN, é inconcebível se adotar esta solução isoladamente, sem a adoção de outros mecanismos de segurança que possam garantir a integridade da rede corporativa. A ausência desta preocupação pode facilitar ataques e a invasão da rede corporativa. Desta forma, verificou-se a necessidade da aplicação combinada de serviços de segurança como *firewalls*, SDIs, *proxys* e listas de controles de acesso, de forma a estruturar em um ambiente corporativo uma defesa em profundidade.

- Demonstrou-se que em determinadas situações os custos de implementação de uma infra-estrutura VPN podem ser mais elevados que a permanência de circuitos dedicados, o que se contrapõe as afirmações de todas bibliografias consultadas a respeito de VPNs, as quais mencionam que os custos envolvidos na implantação e manutenção de redes VPN são sempre menores quando comparados aos custos de uma rede dedicada.

- Foi proposto um modelo no qual se definiu a disposição ideal de cada componente, discutindo-se as vantagens e deficiências de cada topologia descrita.

- O trabalho revelou que a utilização de sistemas e ferramentas livres, que estão disponíveis largamente na Internet, satisfaz completamente os requisitos de estabelecimento de soluções VPN, bem como, para a implementação de outros componentes de segurança que devem ser combinados.

- Identificou-se que a utilização de *software livre* para construção de VPNs e demais componentes de segurança, torna a solução mais segura, uma solução mais transparente ao Cliente do que *software proprietário*, pois os códigos fontes estão disponíveis a qualquer momento. Além disso, o cliente quando utiliza *softwares* proprietários que não possuem a possibilidade de acessar todo o código fonte, nunca sabe exatamente o que está sendo executado, podendo ser alvo de um *back door*, por exemplo.

Em nosso experimento, demonstramos que a implementação de uma infra-estrutura em *software livre* para interligação de redes corporativas, através de VPNs baseadas na Internet é perfeitamente possível e viável. Porém, a utilização do *software FreeS/WAN* se mostrou ainda um tanto quanto complexa, requerendo uma vasta experiência dos responsáveis pela implantação e manutenção de uma VPN baseada nessa plataforma.

A opção de autenticação através de Certificados Digitais e suporte a NAT não foram implementados, porém, são assuntos e tópicos de interesse para trabalhos futuros.

7.1 Contribuições Adicionais

Além de apresentar uma proposta de um modelo de segurança para VPNs IP corporativas, este trabalho realizou um estudo completo sobre os aspectos e as tecnologias existentes na área de segurança de redes de computadores envolvidos na construção e manutenção dessas redes, bem como, desenvolveu e documentou um experimento que implementa uma solução VPN utilizando ferramentas e sistemas livres, avaliando e demonstrando o funcionamento e viabilidade técnica nessa plataforma.

As fontes bibliográficas que podem ser atualmente encontradas tratam as VPNs em um contexto isolado, sendo um grande problema para o pesquisador realizar um apanhado geral de informações sobre todos os componentes pertencentes a uma solução VPN

corporativa, de forma a inteirar-se do estado da arte das diversas tecnologias e produtos disponíveis para este fim. Todas as publicações consultadas sobre a implementação de VPNs em *software livre* falharam em não detalhar profundamente os passos e cuidados a serem seguidos e tratados para a instalação de uma VPN, tanto é, que todas as tentativas realizadas, seguindo *ipsis litteris* os procedimentos apresentados, falharam. Desta forma, outro recurso bastante utilizado no experimento deste trabalho foi o método da *tentativa e erro*. Aproximadamente três meses foram utilizados somente para a implementação do experimento apresentado neste trabalho.

Portanto, esta dissertação também contribui significativamente na documentação dos procedimentos necessários para se construir uma VPN, baseado no *software FreeS/WAN* e na elaboração de uma proposta de VPN viável para as redes da Previdência Social, bem como para outras redes corporativas, pois a pesquisa não se limitou a uma situação específica, elaborou um modelo de topologia de segurança genérico que pode ser estendido a várias outras redes.

7.2 Trabalhos Futuros

Entende-se que o avanço tecnológico e o desenvolvimento da pesquisa devem-se, principalmente, da continuidade e evolução dos trabalhos científicos e de seus experimentos. Ao longo desta dissertação, apontamos diversos aspectos e serviços de segurança relacionados a VPNs que podem ser aprimorados ou reavaliados. Em especial, temos interesse em realizar trabalhos de pesquisa que dão continuidade e aprimoramento a este trabalho:

- Proposta para resolver a fragilidade de segurança no estabelecimento de túneis VPN com cliente remotos. Esta problemática foi abordada na seção 5.3 desta dissertação. Na ocasião, destacou-se que várias alternativas já foram propostas, mas até então, nenhuma solução sugerida havia sido satisfatória.

- Análise detalhada sobre o desempenho de um *gateway* VPN em função do seu posicionamento em relação ao *firewall*. Nesta, deverá ser avaliado parâmetros e resultados em função do volume e tipo de tráfego, tamanho médio de pacotes, desempenho de servidor, rajadas, entre outros aspectos . A questão de

desempenho em função do posicionamento do *gateway* VPN em função ao *firewall* foi abordada na seção 5.1.

- Uma análise comparativa entre VPNs implementadas em *software* e as implementadas em roteadores, avaliando questões de segurança, desempenho, gerenciamento, escalabilidade, recursos, entre outros aspectos.

- Implementação de uma infra-estrutura de VPN baseada no uso de Certificados Digitais, utilizando *software livre*. Nesta dissertação, abordamos este tema, porém, não chegamos a implementar uma infra-estrutura de PKI usando *software livre*. Desta forma, esta seria uma aprimoração deste trabalho.

- Implementação de uma proposta de uma estrutura de IDSRs na plataforma de *software livre*, utilizando-se o *software Snort*.

- Análise comparativa mais aprofundada entre os modelos de autenticação existentes nas soluções VPN.

- Aplicabilidade e desempenho de VPNs baseadas na Internet para a transmissão de Voz e Imagem, bem como a utilização de VoIP (Voz sobre IP) em uma estrutura de VPN.

- Análise da construção de VPNs em uma infra-estrutura *wireless* (sem fio).

- Análise de escalabilidade, segurança e gerenciamento de VPNs baseadas em Internet implementadas por ISPs. Esta questão foi abordada na seção 3.7. Outra questão em relação a este assunto seria: como deverá ser a infra-estrutura de uma concessionária de telecomunicações para suportar múltiplos clientes VPN de maneira escalável e segura?

- Análise comparativa do funcionamento e da integração dos diversos mecanismos de NAT nas redes VPN, tanto os implementados em *hardware*, como os construídos em cima de uma plataforma de *software*.

REFERÊNCIAS BIBLIOGRÁFICAS

BLUNK, L., VOLLBRECHT, J. **PPP Extensible Authentication Protocol (EAP).** RFC 2284, Março de 1998.

BRIAN, T. **The Moron's Guide to Kerberos.** Disponível em: http://www.isi.edu/~brian/security/kerberos.html, acessado em setembro de 2003.

CAMPOS, A. **Kernel 2.6 – O que muda na nova versão do Linux.** Revista do Linux, Ed. Conectiva S.A., Ano IV, nº 40, p. 35, abril, 2003.

CASWELL, B., BEALE, J., FOSTER, James C., POSLUNS, J. **Snort 2: Sistema de Detecção de Intruso Open Source.** Ed. Alta Books, Rio de Janeiro-RJ, 2003.

CHAKRABARTI, A., MANIMARAN, G., **Internet Infrastructure Security: A Taxonomy.** Revista IEEE Network, vol. 16, número 6, p. 13 – 21, Edição Novembro/Dezembro, 2002.

CHANG, Rocky K. C. **Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial**. IEEE Communications Magazine, vol. 40, número 10, p. 42 – 51, outubro, 2002.

CISCO, **Cisco Networking Academy Program**. CCNA Semester 3, v. 2.1.2, Cisco Systems, 2000.

CLERCQ, Jeremy D., PARIDAENS, Oliver. **Scalability Implications of Virtual Private Networks**. IEEE Communications Magazine, vol. 40, número 5, p. 151 – 157, maio, 2002.

DAEMEN, J. e RIJMEN, V. **The Design of Rijndael / AES - The Advanced Encryption Standard**. Ed. Springer-Verlag, Berlim, 2002.

DENKER, John S., BELLOVIN, Steven M., DANIEL, H., MINTZ, Nancy L., KILLIAN, Tom, PLOTNICK, Mark A. **MOAT: a Virtual Private Network Appliance and Services Platform**. Proceedings of LISA ‘99, Seattle, WA, USA, Novembro, 1999.

DIFFIE, W., HELLMAN, M. E. **New Directions in Cryptography**. IEEE Transactions on Information Theory, vol. IT-22, pg. 644-654, novembro, 1976.

DOWNES, K., FORD, M., LEW, H., SPANIER, S., STEVENSON, T. **Internetworking: manual de tecnologias**. Tradução da 2ª edição original, Ed. Campus, Rio de janeiro-RJ, 2000.

FENZI, K., WRESKI, D. **Linux Security HOWTO**. disponível em: http://www.rau-tu.unicamp.br/nou-rau/*software*livre/document/?code=21, acessado em Outubro de 2003.

FERGUSON, P., HUSTON, G. **What's is a VPN? – Part I**. The Internet Protocol Journal – CISCO, Vol. 1, número 1 - 1998, disponível na Internet em: http://www.cisco.com/warp/public/759/ipj_1-1/ipj_1-1_VPN1.html, acessado em Julho de 2003.

FERGUSON, P., HUSTON, G. **What's is a VPN? – Part II**. The Internet Protocol Journal – CISCO, Vol. 1, número 2 - 1998, disponível na Internet em: http://www.cisco.com/warp/public/759/ipj_1-2/ipj_1-2_vpn.html, acessado em Julho de 2003.

FERREIRA, Rubem E. **Linux – Guia do Administrador do Sistema**. Ed. Novatec, São Paulo-SP, 2003.

FIGUEIREDO, F. **Acesso Remoto em firewalls e topologia para gateways VPN**. Disponível em: http://bastion.las.ic.unicamp.br/paulo/papers/2001-WSeg-francisco.figueiredo-gateway.VPN.pdf, acessado em setembro de 2003.

FreeS/WAN. **Linux Org - Online Documentation**. Disponível em: http://www.freeswan.org/doc.html, acessado em novembro de 2003.

GRANADO, Marcus C., VIEIRA, D., GEUS, P. **Aspectos Criptográficos no Windows NT.** 2º Conferência sobre Redes de Computadores, Universidade de Évora, Portugal, 1999.

HARKINS, D., CARREL, D. **The Internet Key Exchange (IKE)**. IETF RFC 2409, novembro, 1998.

HAMZEH, K., PALL, G., VERTHEIN, W., TAARUD, J., LITTLE, W., ZORN, G., **Point-to-Point Tunneling Protocol (PPTP)**, IETF RFC 2637, Julho, 1999.

LEE, Donald. **Enhanced IP Services for Cisco Networks**. Cisco Press, Indianapolis, USA, 1999.

LINUX JOURNAL, **Howto: Encrypted Tunnels with FreeS/Wan x509 Path**, disponível em: http://www.linuxjournal.com/article.php?sid=7003, acessado em outubro de 2003.

MANIKOPOULOS, C., PAPAVASSILIOU, S. **Network Intrusion and Fault Detection: A Statistical Anomaly Approach**. IEEE Communications Magazine, vol. 40, número 10, p. 76 – 82, outubro, 2002.

MAUGHAN, D., SCHERTLER M., SCHNEIDER, M., TURNER, J. **Internet Security Association and Key Management Protocol (ISAKMP)**. IETF – RFC 2408, Novembro, 1998.

MICROSOFT CORPORATION. **Virtual Private Networking in Windows 2000: an Overview**. White paper, Microsoft Press, Washington, USA, 1999.

MIT. **Kerberos: The Network Authentication Protocol**. Massachussets Institute of Technology – MIT, disponível em: http://web.mit.edu/kerberos/www/, acessado em dezembro de 2003.

NAKAMURA, E. T. **Um Modelo de Segurança de Redes para Ambientes Cooperativos**. Tese de Mestrado, IC – UNICAMP, Campinas, Setembro, 2000.

NAKEN, Ron. **Linux as a VPN Client to FireWall-1**. Check Point *software* Technologies LTD, disponível em: http://support.checkpoint.com/kb/docs/public/firewall1/4_1/pdf/fw-linuxvpn.pdf, acessado em agosto de 2003.

NELSON, Thomas J. **Setting up a Linux FreeS/Wan VPN for Remote Users**. disponível em: http://reguly.net/alvaro/linux/SettingUp_FreeSwan.html, acessado em novembro de 2003.

NEMETH, E., SNYDER, G., HEIN, Trent R. **Linux Administration handbook**. Ed. Prentice Hall, New Jersey, USA, 2002.

NORTHCUTT, S., ZELTSER, L., WINTERS, S., FREDERICK, Karen K., RITCHEY, Ronald W. **Desvendando Segurança em Redes**. Ed. Campus, Rio de Janeiro-RJ, 2002.

NIST, **Secure Hash Algorithm**, U.S. Government Federal Information Processing Standard, 1993.

KAEO, Merike. **Designing Network Security: A pratical guide to creating a secure network infrastructure**, Cisco Press, Indianápolis, USA, 1999.

KARA, Atsushi. **Security Remote Access from Office to Home**. IEEE Communications Magazine, vol. 39, número 10, p. 68 – 72, outubro, 2001.

KENT, S., ATKINSON, R. **Security Architecture for IP**. IETF RFC 2401, Novembro, 1998a.

KENT, S., ATKINSON, R. **IP Authentication Header**. IETF RFC 2402, Novembro, 1998b.

KENT, S., ATKINSON, R. **IP Encapsulating Security Payload (ESP)**. IETF RFC 2406, Novembro, 1998c.

KERCKHOFF, August. **La Cryptographie Militaire**. *Journal des sciences militaires,* vol. IX, p. 5-38, França - janeiro de 1883 e p. 161-191, França - fevereiro de 1983.

KING, Christopher M. **The 8 Hurdles to VPN Deployment**. Information Security Magazine, disponível em: http://infosecuritymag.techtarget.com/articles/1999/vpn.shtml, acessado em dezembro de 2003.

KAUFMAN, C., PERLMAN, R. E., SPENCINER, M. **Network Security**. 2ª edição, Prentice Hall, Englewood Cliffs – NJ, 2002.

OLIVA, F. Bastiglia. **Snort**. Revista Security Magazine, Security Magazine Editora LTDA, Ano III, número 13, p. 5 – 11, 2001.

ORMAN, H. **The OAKLEY Key Determination Protocol**. IETF, RFC 2412, Novembro, 1998.

ORTIZ, Eduardo Bellincanta. **VPN: Implementando soluções com Linux**. Ed. Érica, São Paulo-SP, 2003.

PALL, G.S., ZORN, G. **Microsoft Point-to-Point Encryption (MPPE) Protocol**. The Internet Engineering Task Force, Internet Draft, disponível em: http://www.ietf.org/proceedings/00dec/I-D/draft-ietf-pppext-mppe-05.txt, acessado em dezembro de 2003.

PARKER, Donn B. **Demonstrating the Elements of Information Security with Threats**. Proceedings of the 17th National Computer Security Conference, páginas 421-430, 1994.

PERLMUTTER, Bruce. **Virtual Private Networking – A view from the trenches**. Ed. Prentice Hall, USA, 2000.

PERLMAN, R., KAUFMAN, C. **Analysis of the IPSec Key Exchange Standard**. Tenth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Massachusetts, 20 a 22 de junho, 2001.

PREVELAKIS, V., KEROMYTIS, A. **Designing an Embedded Firewall/VPN Gateway**. Disponível em http://downloads.securityfocus.com/library/EmbeddedVPN.pdf, acessado em novembro de 2003.

RALIO, Paulo R. **VPN no Red Hat Linux**. Artigos Linux in Brazil, disponível em: http://brlinux.linuxsecurity.com.br/tutoriais/2003_05.html, acessado em setembro de 2003.

RAYMOND, Eric S. T**he Cathedral & the Bazaar - Musings on Linux and Open Source by an Accidental Revolutionary**. Ed O'Reilly, USA, 1999.

RENSING, C., KARSTEN, M., STILLER, B. **AAA: A Survey and a Policy-Based Architecture and Framework**. IEEE Network, vol. 16, número 6, p. 22 – 27, Edição de Novembro/Dezembro, 2002.

RICHARDSON M., REDELMEIER H., SPENCER H. **Opportunistic Encryption using The Internet Key Exchange**. Draft – IETF, Junho, 2003.

RIGNEY, C., WILLENS, S., LIVINGSTON, RUBENS, A. **Remote Authentication Dial In User Service (RADIUS)**. IETF RFC 2865, Junho, 2000.

RIGNEY, C. **RADIUS Accounting.** IETF RFC 2866, Junho, 2000.

RIVEST, R. L. **The MD5 Message-Digest Algorithm**. IETF RFC 1320, abril, 1992.

RIVEST, R. L., SHAMIR, A. e ADLEMAN, L. **On a Method for Obtaining Digital Signatures and Public Key Cryptosystems**. Communications of the ACM, vol. 21, p. 120-126, fevereiro, 1978.

ROSEN, E., VISWANATHAN, A., CALLON, R. **Multiprotocol Label Switching Architecture**. IETF RFC 3031, Janeiro, 2001.

RUSSELL, R. **Linux Iptables HOWTO**. Disponível em: http://www.linuxguruz.com/iptables/howto/iptables-HOWTO.html, acessado em dezembro de 2004.

SILVA, Lino Sarlo. **VPN: Aprenda a construir redes privadas virtuais em plataformas Linux e Windows**, Ed. Novatec, São Paulo, SP, Brasil – 2003.

SIMPSON, W. **The Point-to-Point Protocol (PPP)**. RFC 1661, IETF, Julho, 1994.

SIMPSON, W., **PPP Challenge Handshake Authentication Protocol (CHAP)**, RFC 1994, IETF, Agosto de 1996.

STALLINGS, William. **Data & Computer Communications**, Sixth Edittion, Ed. Prentice Hall, USA, 1999a.

STALLINGS, William. **Network Security Essentials – Aplications and Standards**, Ed. Prentice Hall, New Jersey, USA, 1999b.

STALLMAN, Richard. **O Manifesto GNU**. Free *software* Foundation, disponível na Internet em: http://www.gnu.org/gnu/manifesto.pt.html, acessado em julho de 2003.

TANENBAUM, Andrew. **Computer Networks**. Fourth Edition. Prentice Hall. New Jersey, USA, 2003.

TOWNSLEY, W., VALENCIA, A., RUBENS, A., PALL, G., ZORN, G., PALTER, B. **Layer Two Tunneling Protocol "L2TP"**, IETF RFC 2661, Agosto, 1999.

TUCHMAN, W. **Hellman Presents No Shortcut Solutions to DES**. IEEE Spectrum, vol. 16, p. 40-41, julho, 1979.

WENSTROM, Michael. **Managing Cisco Network Security**. Editora Alta Books, Rio de Janeiro, 2002.

WILSON, Matthew D. **VPN NowTo**. Disponível em: http://www.rau-tu.unicamp.br/nou-rau/*software*livre/document/?code=23, acessado em 2003.

WING, P., O'HIGGINS, B. **Using Public-Key Infrastructures for Security and Risk Management**. IEEE Communications Magazine, vol. 37, número 9, p. 71 – 73, setembro, 1999.

Apêndice A

Conceitos sobre Software Livre e GNU

Apêndice A - Conceitos sobre Software Livre e GNU

1. Conceitos sobre Software Livre e GNU

Software livre, antes de tudo, é uma filosofia de desenvolvimento e uso de *software* baseado na idéia de um desenvolvimento aberto e colaborativo, onde vários desenvolvedores podem contribuir para o aperfeiçoamento do *software*. Esta filosofia encontra suas raízes consolidadas na livre troca de conhecimento e de pensamentos que podem tradicionalmente ser encontrada no campo científico. Assim como as idéias, os *softwares* são tangíveis e podem ser copiados sem perda alguma, e dentro do conceito de *software* livre, a sua distribuição é a base de um processo evolutivo que alimenta de forma continuada o desenvolvimento do pensamento.

O grande impulsionador do conceito desse movimento de *software* livre foi Richard Stallman, quando em 1984, criou o projeto GNU que tinha como objetivo o desenvolvimento de um sistema operacional completo e livre similar ao UNIX: o sistema GNU (GNU é um acrônimo recursivo que se pronuncia "guh-NEW" ou "guniw"). Variações do sistema GNU, que utilizam o núcleo Linux, ou Sistema GNU baseado em Linux, são hoje largamente utilizadas; apesar desses sistemas serem normalmente chamados de Linux, eles são mais precisamente chamados Sistemas GNU/Linux.

O conceito de *software* livre definido por Richard Stallman fornece ao usuário a liberdade de executar, copiar, distribuir, estudar, modificar e aperfeiçoar o *software*, o que implica que o código-fonte do mesmo deve estar disponível para alteração. Esta maneira de pensar, descrita por Richard Stallman, consolidou na criação da *Free Software Foundation* e na definição de *software* livre sobre a forma de quatro liberdades, as quais definem uma licença GPL (*General Public License*). São elas:

- **Liberdade 0** - A liberdade de executar o programa, para qualquer propósito.

- **Liberdade 1** - A liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades. Aceso ao código-fonte é um pré-requisito para esta liberdade.

- **Liberdade 2** - A liberdade de redistribuir cópias de modo que você possa cooperar com outros usuários.

- **Liberdade 3** - A liberdade de aperfeiçoar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie. Acesso ao código-fonte é um pré-requisito.

O conceito de *Open Source*, ou código aberto, foi a grande chave para o sucesso desse movimento, pois este permite que programadores possam livremente compartilhar códigos fontes de programas com outros programadores, a fim de que estes possam melhorá-los. A definição deste termo, bem como o de desenvolvimento colaborativo, teve sua origem em um manifesto publicado por Eric S. Raymond, intitulado "*The Cathedral and The Bazaar*" (RAYMOND, 1999), o qual foi divulgado e livremente redistribuído pela Internet, onde foi exposto um novo modelo de desenvolvimento cooperativo, onde milhares de desenvolvedores podem participar do desenvolvimento de um determinado *software* compartilhando código. Eric Raymond faz-se referência a esse modelo de desenvolvimento aberto como sendo o *Bazaar*, (bazar beneficente, em português), em contraposição ao modelo de desenvolvimento fechado e proprietário, que seria a *Cathedral* (catedral, em português). O GNU ficou conhecido com um modelo de desenvolvimento do tipo *Bazaar*.

Outra questão que deve ser esclarecida, ainda bastante mal interpretada pelos próprios profissionais da área de Informática, é quanto a gratuidade, ou não, pela permissão de uso de um *software* livre, como por exemplo, um sistema GNU/Linux. Um *software* livre não significa um *software* gratuito. Observa-se que as liberdades de um *software* livre (GPL) descritos não mencionam que o *software* deve ser gratuito. Desta forma, é perfeitamente possível que Empresas possam fornecer um serviço de distribuição de um *software* livre objetivando o lucro.

Na verdade, o próprio Richard Stallman confessa, que em seu manifesto GNU, descuidou-se na escolha de algumas palavras utilizadas. O trecho do manifesto GNU que ele mesmo “corrige” posteriormente é:

> GNU, que significa Gnu Não é Unix, é o nome para um sistema de *software* completo e compatível com o Unix, que eu estou escrevendo para que possa **fornecê-lo gratuitamente** para todos os que possam utilizá-lo. (STALLMAN, 1985).

E relata em nota de roda pé:

> A intenção era de que ninguém teria que pagar pela **permissão para usar** o sistema GNU. Mas as palavras não deixam isso claro, e as pessoas frequentemente interpretam que isso significa que as cópias do GNU têm sempre que serem distribuídas gratuitamente ou por um valor simbólico (STALLMAN, 1985).

E continua:

> Subsequentemente eu aprendi a distinguir cuidadosamente entre "free" no sentido de liberdade e "free" no sentido de preço. O *software* livre é o *software* que os usuários tem a liberdade de distribuir e modificar. Alguns usuários podem obter cópias sem custo, enquanto que outros podem pagar para receber cópias -- e se a receita ajuda a aperfeiçoar o *software*, melhor ainda (STALLMAN, 1985).

Atualmente, os sistemas de *software* livre em geral são seguramente um novo paradigma da nova era da Computação, influenciando significativamente o comportamento das Organizações, tanto aquelas que utilizam a Tecnologia da Informação como recurso estratégico, e impulsionador dos seus negócios, como aquelas que atuam no desenvolvimento de *software*. Esse crescimento é um fato, e se expande de maneira impressionante, influenciando, inclusive, decisões econômicas e políticas.

Portanto, ente trabalho pretende confirmar também a aplicabilidade dos sistemas livres disponíveis, particularmente no ambiente GNU/Linux, para os propósitos desse estudo de caso, na área de Segurança de Redes. Além disso, é coerente que se pretenda buscar soluções mais econômicas para a implementação de uma VPN, combinada a serviços e protocolos de segurança que definem um modelo de Defesa em Profundidade, através de ferramentas e sistemas livres.

Apêndice B

Sistema Operacional GNU/Linux

Apêndice B - Sistema Operacional GNU/Linux

1. Sistema Operacional GNU/Linux

O *kernel* Linux foi implementado pelo estudante finlandês Linus Torvalds. Linus começou a implementar o sistema a partir de sua vontade de adicionar mais recursos ao sistema operacional Minix, um sistema implementado por Andrew Tanembaum e que é utilizado para fins educacionais, Linus iniciou o desenvolvimento do Linux em um período de inverno na Finlândia onde passou meses em casa construindo o que se tornaria um grande *kernel* de sistema operacional. A partir de uma resposta negativa do autor do Minix sobre a adição de novos recursos ao sistema sobre o pretexto de que este ficaria muito complexo para fins educativos, Linus decidiu começar a implementação do *kernel* Linux "do zero", baseando boa parte do sistema no código do Minix. Linus lançou a primeira versão do *kernel* em 1991 e o postou na Internet gratuitamente; assim, começaram a surgir desenvolvedores para ajudá-lo na implementação do mesmo, o *kernel* foi adicionado ao projeto GNU que possuía um sistema que precisava de um *kernel* mais aperfeiçoado do que o *kernel* que possuía até o momento, com o passar do tempo, centenas de desenvolvedores espalhados pelo mundo se envolveram no projeto e, atualmente, o sistema operacional GNU/Linux pode ser considerado um sistema robusto, com evolução contínua. Esta evolução é demonstrada na Tabela abaixo.

Tabela 9 – Tamanho do código fonte do kernel versus ano de lançamento. Fonte: (CAMPOS, 2003)

Versão	***Data Lançamento***	***Linhas de Código***
0.01	Setembro/1991	7.5 K
1.0	Março/1994	158 K
1.2	Março/1995	277 K
2.0	Julho/1996	649 K
2.2	Janeiro/1999	1536 K
2.6	Versão estável não lançada	+/- 6000 K

Quando o Linux foi escrito, em 1991, o projeto do Sistema Operacional GNU, iniciado em 1984, já estava quase finalizado, possuindo praticamente todos os componentes de um Sistema Operacional completo, com exceção do *kernel*, que ainda estava sendo desenvolvido. Com isso, quando o Linux foi concluído, ele completou a última grande lacuna que faltava. Desta forma, pode-se integrar o Linux junto com o sistema GNU para compor um Sistema Operacional livre completo: um sistema GNU baseado em Linux, ou sistema GNU/Linux, para simplificar.

Apêndice C

Arquivos de Configuração do IPSec

Apêndice C - Arquivos de Configuração do IPSec

Neste apêndice incluímos os scripts de configuração do IPSec no experimento realizado com o *software* FreeS/Wan.

1. Arquivos /etc/ipsec.secrets

Apresentaremos a seguir os arquivos /etc/ipsec.secrets dos servidores VPN do experimento realizado.

Arquivo ***/etc/ipsec.secrets*** gerado no servidor **rightserver**:

```
200.240.0.1 200.240.1.1      "frase secreta"
: RSA {
      # RSA 1024 bits   rightserver   Thu Nov 13 15:24:19 2003
      # for signatures only, UNSAFE FOR ENCRYPTION
      #pubkey=0sAQPzzql5aXxbO5rVJQZCAeZ+3g1Vh2xlOOc76Jctva5J7pbAoNN3yr8lFzz
1Pvbh7pka2OB+XnWCWF7zIbTyj+bxmdbWldXWooiCxrx7Q5qbJnmNpGhJI0ziN1oWcUmVgErZ3+
kgXogvgrLoxpc1nrvprybLGyLxTUaFXoVYwTyHzw==
      Modulus:
0xf3cea979697c5b3b9ad525064201e67ede0d55876c6538e73be8972dbdae49ee96c0a0d37
7cabf25173cf53ef6e1ee991ad8e07e5e7582585ef321b4f28fe6f199d6d695d5d6a28882c6
bc7b439a9b26798da46849234ce2375a16714995804ad9dfe9205e882f82b2e8c697359ebbe
9af26cb1b22f14d46855e8558c13c87cf
      PublicExponent: 0x03
      # everything after this point is secret
      PrivateExponent:
0x28a271943c3f6489ef23862bb5aafbbfcface3969210ded134a6c3dcf4f261a7c3cac5789
3f71fdb83df7e352925a7c42f24256a651395b96528859e286d5127f106b04e93caea6072b3
16662d5971aa082c3ff02efa305ab668ad456338c105499c913f0d4c7885630e9d597fdd871
8d357b361a26b89a06dcfb862ea37a1c3
      Prime1:
0xfc49176dc281b8ad4dc655d4228c1995ebc2a9f225c1a0c5ec0618b3cb1811c30181be140
43340306efa4db915a02a5db1fb89e590d2d6dad61debda1d2dc0a7
      Prime2:
0xf7659d509c936b9884cde04210f5d7945cc17ad5098489fc04dfee1d2b28e8681eb2c3920
a8a1c31f196c8c520cd66f7091d0cebc3993ca91c62472d26c0fc99
      Exponent1:
0xa830ba492c567b1e33d98e8d6c5d6663f281c6a16e8115d948041077dcbab68201012962a
d77802049fc33d0b9157193cbfd06990b3739e7396947e6be1e806f
      Exponent2:
0xa4ee68e0686247bb0333ead6b5f93a62e880fc8e06585bfd58954968c7709af014772d0c0
706bd76a10f30836b3399fa06135df282662870bd96da1e19d5fdbb
      Coefficient:
0xc7e320691f7db85b894e167fa6d43a9ee47e5a26fb4b267e00290af8d3eced755d38ceea2
8a44957a375fb26bee197ebd12a250c7f61fa50db53427dd6fa4046
      }
```

Arquivo ***/etc/ipsec.secrets*** gerado no servidor **leftserver**:

```
200.240.0.1 200.240.1.1      "frase secreta"
: RSA {
      # RSA 1024 bits   leftserver   Thu Nov 13 15:27:06 2003
      # for signatures only, UNSAFE FOR ENCRYPTION
      #pubkey=0sAQOLRoyDXEXMWB+m4smekuG1J+nju1qxCaQdeYs/9TaoMCa+8X1+eCR4io+
UHzNrYlxQke1lkOLagIHe9JZcv/AFVEjeMciAqNNb/ucCpmwk9CFKLMZiJ9AYKEyCRM6CZ7ErDW
Dg4TOK7H8wQR9qzjBmKe2OXl9WxT5J4j08Z1+9uw==
      Modulus:
0x8b468c835c45cc581fa6e2c99e92e1b527e9e3bb5ab109a41d798b3ff536a83026bef17d7
e7824788a8f941f336b625c5091ed6590e2da8081def4965cbff0055448de31c880a8d35bfe
e702a66c24f4214a2cc66227d018284c8244ce8267b12b0d60e0e1338aec7f30411f6ace306
629ed8e5e5f56c53e49e23d3c675fbdbb
      PublicExponent: 0x03
      # everything after this point is secret
      PrivateExponent:
0x17366cc08f60f76405467b219a6dd048dbfc509f39c82c4604e9973553891c080675283f9
51406141717ee05333c9064b818523b9825cf156afa7e190f7552ab4cdaae27bc2b8266eab5
668df7969e736484d6afc75faa8c45d05131373e8340ac65b1752890fa5e9df607fc51962ca
16048111a7d46d1cbf7f155fa14c8377b
      Prime1:
0xf689fc311a907f22df59edfa9b972469762da929783d0cf6fc5ab4dd2c9e1cc8b456aeaa2
25f91e4ffb38ff02ed322cbc8001e1f9b1aba0436840e52c96e782b
      Prime2:
0x909ecd1244eb1b46fc6491b43d4d49d64fff7b7e3dacc3d7890fe640566d37646c548977c
b6e1acfcbb88145527601d2203d099fd493207243b62b0d213ff8b1
      Exponent1:
0xa45bfd7611b5aa173f914951bd0f6d9ba41e70c65028b34f52e7233e1dbebddb22e474716
c3fb698aa77b54ac9e217328555696a676726ad79ad5ee1db9efac7
      Exponent2:
0x6069de0c2df2122f52edb67828de31398aaa5254291dd7e5060a9980399e24ed9d8db0fa8
79ebc8a87d0562e36f9568c157e066a8db76af6d7cec75e162aa5cb
      Coefficient:
0xcc4aae708743d979df980a9ebceaa79e6b035f8e70955700037a0af3052202722cc41f669
584a214930b6fec8b4674171879a120c8fac3de107ec67e4f1c5c17
      }
```

2. Arquivo /etc/ipsec.conf

Apresentamos a seguir o arquivo **/etc/ipsec.conf** que foi utilizado para realizar o teste de Criptografia Oportunista no experimento deste trabalho.

```
# /etc/ipsec.conf - FreeS/WAN IPsec configuration file
# RCSID $Id: ipsec.conf.in,v 1.11 2003/06/13 23:28:41 sam Exp $

# This file:  /usr/local/share/doc/freeswan/ipsec.conf-sample
#
# Manual:     ipsec.conf.5
#
# Help:
# http://www.freeswan.org/freeswan_trees/freeswan-2.03/doc/quickstart.html
# http://www.freeswan.org/freeswan_trees/freeswan-2.03/doc/config.html
# http://www.freeswan.org/freeswan_trees/freeswan-2.03/doc/adv_config.html
#
# Policy groups are enabled by default. See:
# http://www.freeswan.org/freeswan_trees/freeswan-
2.03/doc/policygroups.html
#
# Examples:
# http://www.freeswan.org/freeswan_trees/freeswan-2.03/doc/examples


version     2.0   # define a especificacao da versao do FreeS/Wan

# configuração básica que se aplica a todas as conexões

      config setup
      interfaces="ipsec0=eth0"
      klipsdebug=all
      plutodebug=dns
      uniqueids=yes

# Definicao de Conexões

# Definição os parâmetros default das conexões VPN

conn %default
      keyingtries=0
      esp=3des-md5-96
      authby=secret
      disablearrivalcheck=no
      leftrsasigkey=%dns
      rightrsasigkey=%dns

# Definição da Conexao entre os Gateways

conn gwl-gwr
      type=tunnel
      keyexchange=ike
      pfs=no
      keylife=8h
      left=200.240.0.1
```

```
      leftnexthop=200.240.0.250
      right=200.240.1.1
      rightnexthop=200.240.1.250
      auto=add

# Definição da Conexao para possibilitar um host da Rede A estabelecer
# Conexão VPN

conn redea-gwr
      auth=esp
      authby=rsasig
      pfs=no
      left=200.240.0.1
      leftnexthop=200.240.0.250
      leftsubnet=10.64.0.0/24
      right=200.240.1.1
      rightnexthop=200.240.1.250
      rightsubnet=10.64.1.0/24
      auto=add
```

Apêndice D

Relatórios de Testes Realizados

Apêndice D - Relatórios de Testes Realizados

Neste apêndice incluímos a resultado de alguns comandos de testes de conexão VPN realizados no experimento. As saídas de comando foram capturadas através de redirecionamento da saída padrão do linux (tela) para arquivos txt e coladas neste apêndice. Infelizmente a saída em ***txt*** de alguns comandos possuem um número grande de caracteres por linha, fincando impraticável a apresentação destas linhas por inteiro neste documento.

1. Teste de conexão VPN com a opção *look*

A opção ***look*** do comando ***ipsec*** apresenta informações sobre as conexões VPN estabelecidas no servidor em que se está executando o comando. A seguir, apresentamos a linha de comando que foi executada no servidor ***leftserver*** (200.240.0.1) e seu resultado:

ipsec look

```
leftserver Sat Nov 22 18:50:08 AMT 2003
10.64.0.0/24       -> 10.64.1.0/24       => tun0x1002@200.240.1.1 esp0xe33cf358@200.240.1.1
(195)
200.240.0.1/32     -> 200.240.1.1/32     => tun0x1004@200.240.1.1 esp0xe33cf359@200.240.1.1
(1217)
ipsec0->eth0 mtu=16260(1443)->1500
esp0x4de74089@200.240.0.1 ESP_3DES_HMAC_MD5: dir=in  src=200.240.1.1 iv_bits=64bits
iv=0x2e0e8afdb1b900bd ooowin=64 seq=195 bit=0xffffffffffffffff alen=128 aklen=128 eklen=192
life(c,s,h)=bytes(15600,0,0)addtime(761,0,0)usetime(758,0,0)packets(195,0,0) idle=564
refcount=199 ref=8 reftable=0 refentry=8
esp0x4de7408a@200.240.0.1 ESP_3DES_HMAC_MD5: dir=in  src=200.240.1.1 iv_bits=64bits
iv=0x0eb99b343fe872c4 ooowin=64 seq=542 bit=0xffffffffffffffff alen=128 aklen=128 eklen=192
life(c,s,h)=bytes(56350,0,0)addtime(711,0,0)usetime(704,0,0)packets(542,0,0) idle=0
refcount=546 ref=18 reftable=0 refentry=18
esp0xe33cf358@200.240.1.1 ESP_3DES_HMAC_MD5: dir=out src=200.240.0.1 iv_bits=64bits
iv=0xf61c29f44a327221 ooowin=64 seq=195 alen=128 aklen=128 eklen=192
life(c,s,h)=bytes(21840,0,0)addtime(761,0,0)usetime(758,0,0)packets(195,0,0) idle=564
refcount=4 ref=13 reftable=0 refentry=13
esp0xe33cf359@200.240.1.1 ESP_3DES_HMAC_MD5: dir=out src=200.240.0.1 iv_bits=64bits
iv=0x4c0b944274d139ae ooowin=64 seq=536 alen=128 aklen=128 eklen=192
life(c,s,h)=bytes(72920,0,0)addtime(711,0,0)usetime(704,0,0)packets(536,0,0) idle=0 refcount=4
ref=23 reftable=0 refentry=23
tun0x1001@200.240.0.1 IPIP: dir=in  src=200.240.1.1 policy=10.64.1.0/24->10.64.0.0/24
flags=0x8<> life(c,s,h)=bytes(15600,0,0)addtime(761,0,0)usetime(758,0,0)packets(195,0,0)
idle=564 refcount=4 ref=7 reftable=0 refentry=7
tun0x1002@200.240.1.1 IPIP: dir=out src=200.240.0.1
life(c,s,h)=bytes(15600,0,0)addtime(761,0,0)usetime(758,0,0)packets(195,0,0) idle=564
refcount=4 ref=12 reftable=0 refentry=12
tun0x1003@200.240.0.1 IPIP: dir=in  src=200.240.1.1 policy=200.240.1.1/32->200.240.0.1/32
flags=0x8<> life(c,s,h)=bytes(56350,0,0)addtime(711,0,0)usetime(704,0,0)packets(542,0,0)
idle=0 refcount=4 ref=17 reftable=0 refentry=17
tun0x1004@200.240.1.1 IPIP: dir=out src=200.240.0.1
life(c,s,h)=bytes(55744,0,0)addtime(711,0,0)usetime(704,0,0)packets(536,0,0) idle=0 refcount=4
ref=22 reftable=0 refentry=22
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.64.1.0       200.240.0.250   255.255.255.0   UG        0 0          0 ipsec0
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 eth0
200.240.0.0     0.0.0.0         255.255.255.0   U         0 0          0 eth0
200.240.0.0     0.0.0.0         255.255.255.0   U         0 0          0 ipsec0
200.240.1.0     200.240.0.250   255.255.255.0   UG        0 0          0 eth0
200.240.1.1     200.240.0.250   255.255.255.255 UGH       0 0          0 ipsec0
```

Note que na segunda linha da saída do comando é informado que existe uma conexão da rede 10.64.0.0/24 para a rede 10.64.1.0/24 usando o túnel 200.240.1.1, ou túnel *tun0x1002@200.240.1.1* (conexão ***redea-gwr***). Na terceira linha, percebe-se também que existe uma conexão da rede 200.240.0.1/32 para a rede 200.240.1.1 usando o túnel *tun0x1004@200.240.1.1* (conexão ***gwl-gwr***).

2. Teste de conexão VPN com o utilitário *tcpdump*

O utilitário ***tcpdump*** pode mostrar o conteúdo criptografado que está sendo transportando entre as redes, comprovando inclusive a utilização do protocolo ESP do IPSec. Na verdade, essa ferramenta apresenta os pacotes que estão trafegando em uma determinada interface de rede na tela (saída *default*). Abaixo, apresentamos um trecho da saída do comando ***tcpdump***, que foi executado no servidor ***leftserver*** para realizar a leitura de pacotes que trafegam na interface *eth0*, isto, logo após ter sido estabelecido uma conexão VPN entre os referidos servidores e ser disparado um ping infinito do servidor ***leftserver*** ao servidor ***rightserver.***

tcpdump –i eth0

```
18:51:50.118452 leftserver > 200.240.1.1: ESP(spi=0xe33cf358,seq=0x102)
18:51:50.120205 200.240.1.1 > leftserver: ESP(spi=0x4de74089,seq=0x102)
18:51:51.118527 leftserver > 200.240.1.1: ESP(spi=0xe33cf358,seq=0x103)
18:51:51.120254 200.240.1.1 > leftserver: ESP(spi=0x4de74089,seq=0x103)
18:51:52.118660 leftserver > 200.240.1.1: ESP(spi=0xe33cf358,seq=0x104)
18:51:52.120396 200.240.1.1 > leftserver: ESP(spi=0x4de74089,seq=0x104)
18:51:53.109363 arp who-has gwleft tell leftserver
18:51:53.109608 arp reply gwleft is-at 0:c:6e:19:30:4
18:51:53.118741 leftserver > 200.240.1.1: ESP(spi=0xe33cf358,seq=0x105)
18:51:53.120498 200.240.1.1 > leftserver: ESP(spi=0x4de74089,seq=0x105)
18:51:54.118828 leftserver > 200.240.1.1: ESP(spi=0xe33cf358,seq=0x106)
18:51:54.120560 200.240.1.1 > leftserver: ESP(spi=0x4de74089,seq=0x106)
18:51:55.118920 leftserver > 200.240.1.1: ESP(spi=0xe33cf358,seq=0x107)
18:51:55.120669 200.240.1.1 > leftserver: ESP(spi=0x4de74089,seq=0x107)
18:51:56.119067 leftserver > 200.240.1.1: ESP(spi=0xe33cf358,seq=0x108)
18:51:56.120828 200.240.1.1 > leftserver: ESP(spi=0x4de74089,seq=0x108)
18:51:57.119135 leftserver > 200.240.1.1: ESP(spi=0xe33cf358,seq=0x109)
18:51:57.120889 200.240.1.1 > leftserver: ESP(spi=0x4de74089,seq=0x109)
18:51:58.111386 arp who-has leftserver tell gwleft
18:51:58.111457 arp reply leftserver is-at 52:54:5:f7:6:96
18:51:58.119199 leftserver > 200.240.1.1: ESP(spi=0xe33cf358,seq=0x10a)
18:51:58.120920 200.240.1.1 > leftserver: ESP(spi=0x4de74089,seq=0x10a)
18:51:59.119365 leftserver > 200.240.1.1: ESP(spi=0xe33cf358,seq=0x10b)
18:51:59.121092 200.240.1.1 > leftserver: ESP(spi=0x4de74089,seq=0x10b)
18:52:00.119489 leftserver > 200.240.1.1: ESP(spi=0xe33cf358,seq=0x10c)
18:52:00.121211 200.240.1.1 > leftserver: ESP(spi=0x4de74089,seq=0x10c)
18:52:01.119559 leftserver > 200.240.1.1: ESP(spi=0xe33cf358,seq=0x10d)
18:52:01.121324 200.240.1.1 > leftserver: ESP(spi=0x4de74089,seq=0x10d)
18:52:02.119641 leftserver > 200.240.1.1: ESP(spi=0xe33cf358,seq=0x10e)
...
```

3. Relatórios de testes gerados pelo *Sniffer Pro*

Apresentamos amostras de relatórios da ferramenta *Sniffer Pro*, obtidos a partir do monitoramento em três seções distintas: estabelecimento de uma SA, seção *telnet* e uso de *Ping*, todos realizados entre os *gateways* VPN *leftserver* e *rightserver*. Perceberemos que não há como se verificar o protocolo de aplicação que está sendo utilizado, devido ao tunelamento utilizado.

Quadro obtido durante uma seção Telnet:

```
- - - - - - - - - - - - - - - - - - - - Frame 1 - - - - - - - - - - - - - - - - - - - -
Frame Status Source                      Destination                  Bytes Rel Time
Delta Time   Abs time              Summary
-------------------------------------------------------------------------------------------
1 M      [200.240.0.1]                [200.240.1.1]                  126 0:00:00.000
0.000.000    25/11/2003 14:45:20 DLC        Ethertype=0800, size=126 bytes
                                 IP          D=[200.240.1.1] S=[200.240.0.1] LEN=92 ID=24538
                                 IP         ESP SPI=3520375608

DLC:  ----- DLC Header -----
      DLC:
      DLC:  Frame 1 arrived at  14:45:20.0943; frame size is 126 (007E hex) bytes.
      DLC:  Destination = Station 000C6E193004
      DLC:  Source      = Station 525405F70696
      DLC:  Ethertype   = 0800 (IP)
      DLC:
IP: ----- IP Header -----
      IP:
      IP: Version = 4, header length = 20 bytes
      IP: Type of service = 00
      IP:       000. ....   = routine
      IP:       ...0 .... = normal delay
      IP:       .... 0... = normal throughput
      IP:       .... .0.. = normal reliability
      IP:       .... ..0. = ECT bit - transport protocol will ignore the CE bit
      IP:       .... ...0 = CE bit - no congestion
      IP: Total length    = 112 bytes
      IP: Identification  = 24538
      IP: Flags           = 0X
      IP:       .0.. .... = may fragment
      IP:       ..0. .... = last fragment
      IP: Fragment offset = 0 bytes
      IP: Time to live    = 64 seconds/hops
      IP: Protocol        = 50 (ESP)
      IP: Header checksum = 879F (correct)
      IP: Source address      = [200.240.0.1]
      IP: Destination address = [200.240.1.1]
      IP: No options
      IP:
ESP: ----- IP ESP -----
      ESP:
      ESP: Security Parameters Index = 3520375608
      ESP: Sequence Number           = 1
      ESP: Payload Data              =
D6B8E89D0142564E5A581C174F052293CA7941F5F30ECD2AE6AFF08067C842A3B51218380BD262551C96CD76B27 …
ADDDR  HEX                                                 ASCII
0000: 00 0c 6e 19 30 04 52 54 05 f7 06 96 08 00 45 00 | ..n.0.RT.÷.–..E.
0010: 00 70 5f da 00 00 40 32 87 9f c8 f0 00 01 c8 f0 | .p_Ú..@2‡ŸÈð..Èð
0020: 01 01 d1 d4 ab 38 00 00 00 01 d6 b8 e8 9d 01 42 | ..ÑÔ«8....Ö¸è•.B
0030: 56 4e 5a 58 1c 17 4f 05 22 93 ca 79 41 f5 f3 0e | VNZX..O."“ÊyAõó.
0040: cd 2a e6 af f0 80 67 c8 42 a3 b5 12 18 38 0b d2 | Í*æ¯ð€gÈB£µ..8.Ò
0050: 62 55 1c 96 cd 76 b2 76 7b ad 72 7d 64 ea 49 45 | bU.–Ív²v{r}dêIE
0060: c7 a0 1d 12 d7 47 a8 95 87 ec 52 16 aa 8c c4 db | Ç ..×G¨•‡iR.ªŒÄÛ
0070: b4 84 85 5e 69 2e e0 c5 7a 61 a3 77 25 2c       | ´„…^i.àÅza£w%,
```

Quadro obtido durante uma seção de estabelecimento de uma SA:

```
- - - - - - - - - - - - - - - - - - - - Frame 2 - - - - - - - - - - - - - - - - - - - -
Frame Status Source                    Destination                  Bytes Rel Time
Delta Time   Abs time              Summary
-------------------------------------------------------------------------------------------
2        [200.240.1.1]                [200.240.0.1]                  122 0:00:00.001
0.001.282    25/11/2003 13:46:41 DLC        Ethertype=0800, size=122 bytes
                                 IP         D=[200.240.0.1] S=[200.240.1.1] LEN=88 ID=0
                                 UDP        D=500 S=500  LEN=88
                                 ISAKMP     Header

DLC:  ----- DLC Header -----
      DLC:
      DLC:  Frame 2 arrived at  13:46:41.9272; frame size is 122 (007A hex) bytes.
      DLC:  Destination = Station 525405F70696
      DLC:  Source      = Station 000C6E193004
      DLC:  Ethertype   = 0800 (IP)
      DLC:
IP: ----- IP Header -----
      IP:
      IP: Version = 4, header length = 20 bytes
      IP: Type of service = 00
      IP:       000. ....   = routine
      IP:       ...0 .... = normal delay
      IP:       .... 0... = normal throughput
      IP:       .... .0.. = normal reliability
      IP:       .... ..0. = ECT bit - transport protocol will ignore the CE bit
      IP:       .... ...0 = CE bit - no congestion
      IP: Total length    = 108 bytes
      IP: Identification  = 0
      IP: Flags           = 4X
      IP:       .1.. .... = don't fragment
      IP:       ..0. .... = last fragment
      IP: Fragment offset = 0 bytes
      IP: Time to live    = 63 seconds/hops
      IP: Protocol        = 17 (UDP)
      IP: Header checksum = A89E (correct)
      IP: Source address      = [200.240.1.1]
      IP: Destination address = [200.240.0.1]
      IP: No options
      IP:
UDP: ----- UDP Header -----
      UDP:
      UDP: Source port      =   500 (IKE)
      UDP: Destination port =   500 (IKE)
      UDP: Length           = 88
      UDP: Checksum         = 71F1 (correct)
      UDP: [80 byte(s) of data]
      UDP:
IKE: ----- Internet Key Exchange Header -----
      IKE:
      IKE: Initiator Cookie       = 0x71BE01F65427E238
      IKE: Responder Cookie       = 0x5A5A611CFCD08203
      IKE: Next Payload           = 1 (Security Association (SA))
      IKE: Major Version          = 1
      IKE: Minor Version          = 0
      IKE: Exchange Type          = 2 (Identity Protection)
      IKE: Flags                  = 00
      IKE:              .... ...0 = Payloads not encrypted
      IKE:              .... ..0. = Do not wait for NOTIFY Payload
      IKE:              .... .0.. = Authentication Bit
      IKE:              .... 0... = Not Used
      IKE:              ...0 .... = Not Used
      IKE:              ..0. .... = Not Used
      IKE:              .0.. .... = Not Used
      IKE:              0... .... = Not Used
      IKE: Message ID             =  0
      IKE: Length                 =  80 (bytes)
      IKE: ----- SECURITY ASSOCIATION Payload -----
      IKE:
      IKE: Next Payload           = 0 (None)
      IKE: Reserved               = 0
```

```
      IKE: Payload Length         = 52
      IKE: DOI                    = 0x1(IPSEC DOI)
      IKE: Situation              =  0x1 (SIT_IDENTITY_ONLY)
      IKE: ----- PROPOSAL Payload -----
      IKE:
      IKE: Next Payload           = 0 (This is the last Proposal Payload)
      IKE: Reserved               = 0
      IKE: Payload Length         =  40
      IKE: Proposal #             = 0
      IKE: Protocol ID            = 1 (PROTO_ISAKMP)
      IKE: SPI Size               = 0
      IKE: # of Transforms        = 1
      IKE: SPI Not Present
      IKE: ----- TRANSFORM Payload -----
      IKE:
      IKE: Next Payload           = 0 (This is the last Transform Payload)
      IKE: Reserved               = 0
      IKE: Payload Length         =  32
      IKE: Transform #            = 0
      IKE: Transform ID           = 1 (KEY_IKE)
      IKE: Reserved 2             = 0
      IKE: ***SA Attributes***
      IKE: Flags                   = 80
      IKE:               1... .... = Data Attribute following TV format
      IKE: Attribute Type        = 11 (Reserved)
ADDR  HEX                                                 ASCII
0000: 52 54 05 f7 06 96 00 0c 6e 19 30 04 08 00 45 00 | RT.÷.–..n.0...E.
0010: 00 6c 00 00 40 00 3f 11 a8 9e c8 f0 01 01 c8 f0 | .l..@.?.¨žÈð..Èð
0020: 00 01 01 f4 01 f4 00 58 71 f1 71 be 01 f6 54 27 | ...ô.ô.Xqñq¾.öT'
0030: e2 38 5a 5a 61 1c fc d0 82 03 01 10 02 00 00 00 | â8ZZa.üÐ,.......
0040: 00 00 00 00 00 50 00 00 00 34 00 00 00 01 00 00 | .....P...4......
0050: 00 01 00 00 00 28 00 01 00 01 00 00 00 20 00 01 | .....(....... ..
0060: 00 00 80 0b 00 01 80 0c 0e 10 80 01 00 05 80 02 | ..€...€...€...€.
0070: 00 01 80 03 00 01 80 04 00 05                   | ..€...€...
```

Quadro obtido durante um Ping:

```
- - - - - - - - - - - - - - - - - - - - Frame 1 - - - - - - - - - - - - - - - - - - - -
Frame Status Source                       Destination                  Bytes Rel Time
Delta Time   Abs time               Summary
-------------------------------------------------------------------------------------------
1 M      [200.240.0.1]                [200.240.1.1]                   150 0:00:00.000
0.000.000    25/11/2003 13:33:32 DLC        Ethertype=0800, size=150 bytes
                                 IP          D=[200.240.1.1] S=[200.240.0.1] LEN=116 ID=33063
                                 IP         ESP SPI=4229109324

DLC:  ----- DLC Header -----
      DLC:
      DLC:  Frame 1 arrived at  13:33:32.2629; frame size is 150 (0096 hex) bytes.
      DLC:  Destination = Station 000C6E193004
      DLC:  Source      = Station 525405F70696
      DLC:  Ethertype   = 0800 (IP)
      DLC:
IP: ----- IP Header -----
      IP:
      IP: Version = 4, header length = 20 bytes
      IP: Type of service = 00
      IP:       000. ....   = routine
      IP:       ...0 .... = normal delay
      IP:       .... 0... = normal throughput
      IP:       .... .0.. = normal reliability
      IP:       .... ..0. = ECT bit - transport protocol will ignore the CE bit
      IP:       .... ...0 = CE bit - no congestion
      IP: Total length    = 136 bytes
      IP: Identification  = 33063
      IP: Flags           = 0X
      IP:       .0.. .... = may fragment
      IP:       ..0. .... = last fragment
      IP: Fragment offset = 0 bytes
      IP: Time to live    = 64 seconds/hops
      IP: Protocol        = 50 (ESP)
      IP: Header checksum = 663A (correct)
      IP: Source address      = [200.240.0.1]
      IP: Destination address = [200.240.1.1]
```

```
      IP: No options
      IP:
ESP: ----- IP ESP -----
      ESP:
      ESP: Security Parameters Index = 4229109324
      ESP: Sequence Number           = 244
      ESP: Payload Data              =
6A5CFCF1CBFCD6475FCB331A10CBA8B7ED4B4D0FEA0C4554EB3885A215F3954640F3BB4FA37F7263D6A864E0E18580
17D6...
ADDR  HEX                                                 ASCII
0000: 00 0c 6e 19 30 04 52 54 05 f7 06 96 08 00 45 00 | ..n.0.RT.÷.–..E.
0010: 00 88 81 27 00 00 40 32 66 3a c8 f0 00 01 c8 f0 | .ˆ•'..@2f:Èð..Èð
0020: 01 01 fc 13 16 4c 00 00 00 f4 6a 5c fc f1 cb fc | ..ü..L...ôj\üñËü
0030: d6 47 5f cb 33 1a 10 cb a8 b7 ed 4b 4d 0f ea 0c | ÖG_Ë3..Ë¨·íKM.ê.
0040: 45 54 eb 38 85 a2 15 f3 95 46 40 f3 bb 4f a3 7f | ETë8…¢.ó•F@ó»O£•
0050: 72 63 d6 a8 64 e0 e1 85 80 17 d6 b5 6f b6 0f 7e | rcÖ¨dàá…€.Öµo¶.~
0060: 63 da e5 1b e2 56 d1 88 29 95 76 2e 90 a9 6f 7d | cÚå.âVÑˆ)•v.•©o}
0070: 2a a1 c1 3c a6 3d 89 26 08 2c ac bc af c2 fb 8d | *¡Á<¦=‰&.,¬¼¯Âû•
0080: 3c 2d 14 aa 7b cc 5d 21 c4 8b 86 d9 5f 0a 11 ad | <-.ª{Ì]!Ä‹†Ù_..
0090: 7b e7 4f 19 4e c5                               | {çO.NÅ
```