分类号 TP393 学号 GS06061091

UDC 密级 公开

工程硕士学位论文

基于 IPSec 的 VPN 网关设计与实现

硕士生姓名 杨文武

学科领域 计算机技术

研究方向 计算机网络安全

指导教师 孙志刚 副研究员

国防科学技术大学研究生院

二〇〇八年四月

摘 要

虚拟专用网（Virtual Private Network，VPN）是对内部网的扩展，可以帮助远程用户、分支机构同内部网建立可信的安全连接，并保证数据的安全传输，解决了传统网络中数据传输的安全性问题。VPN 通过对数据进行完整性校验，运用密码算法对数据进行加解密处理来保证其安全性。VPN 网络不同于传统的网络，它能够为逻辑上不能相通的两个局域网络之间建立安全的通讯通道，使得这两个局域网之间的访问能够像同一个局域网内数据互访一样的方便。

本文以 IP 安全协议体系 IPSec 为基础，对虚拟专用网（VPN）这一目前广泛流行的信息安全技术及其实现方案进行深入的研究。结合部队仓库网络的应用环境，设计了基于 IPSec 协议的高效、安全、稳定的 VPN 网关，并对其安全及性能进行了测试。

主要研究内容和成果包括：

1）论文针对仓库部队网络安全需求，对 IPSec 的 AH 协议、ESP 协议、传输模式和隧道模式进行了深入研究。基于网关到网关的 VPN 应用环境，针对 IPSec 协议的特点提出一个 IPSec 协议实现的思路，并对该思路的实现方法进行了研究，设计了一种基于 IPSec 协议的 VPN 网关。为实际应用环境中更好的利用 IPSec 以保证 VPN 的通信安全提供了一种新方法。

2）论文实现了基于 IPSec 协议的 VPN 网关中 IKE 模块、SAB 模块和 SPD 模块、IPSec 处理模块以及策略和 SA 管理模块的详细设计。并对安全策略和安全关联的管理方面设计了图形界面，建立了良好的用户接口。给出了网关实现需用到的具体函数。

论文还设计了一个在安全策略已知情况下，利用 VPN 网关实现的安全隧道的实例。同时还对基于 IPSec 协议硬件加速网关处理以及 IPSec VPN 在实际运用中的多协议问题进行了研究。

主题词：VPN，IPsec，隧道模式，封装安全载荷

ABSTRACT

VPN (Virtual Private Network, VPN) is the extension of the intranet, which can help remote users and branch network establish security connection, and guarantee the security of the data transmission, which sovles the existing security problem during data transferring in the traditional network. Data security is guaranteed by data encryption. Compared with the traditional network, VPN establishes safe communication channel between two LANs that cannot be linked logically, and enables the data access between two LANs as convenient as they are in the same LAN.

This paper aims at further research on Virtual Private Network(VPN), a popular security technique based on the IP Security Protocol architecture(IPSec). Based on the army's warehouse network, a efficient, safe and stable VPN gateway on the IPSec protocol is designed and fully tested in the paper.

The following research content and achievement are included.

1) The paper conducts comprehensive research of AH protocol, ESP protocol, transmitting mode and tunnel mode to IPSec to satisfy the security demands in the army's warehouse network. Based on the fact that the gateway is closely employed to the other gateway in the applicaiton, a VPN gateway based on IPSec protocol is designed, which also considers the features of IPSec protocol. This paper provides a new technique to make better use of IPSec in real application circumstance to ensure the communication security of VPN.

2) The paper implements the detailed design within IKE, SAD and SPD modules in IPSec protocol. Meanwhile IPSec protocol tacticsand SA administration module of the VPN gateway were fully detailed. Additionally, GUI interface for the security strategy and management are well developed. The functions to facilitate gateway implementation are provided as well.

The paper also designed an example of the security tunnel though VPN gateway under certain security strategy. Furthermore, studies were carried on the hardware-accelerated gateway and the multi-protocol application of the IPSec VPN.

Key Words: Virtual Private Network, Internet Protocol security, Tunnel Mode, Security Policy Database

表 目 录

图 目 录

独 创 性 声 明

本人声明所呈交的学位论文是我本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表和撰写过的研究成果，也不包含为获得国防科学技术大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文题目：基于 IPSec 的 VPN 网关设计与实现

学位论文作者签名：杨文武　　日期：2008 年 6 月 13 日

学位论文版权使用授权书

本人完全了解国防科学技术大学有关保留、使用学位论文的规定。本人授权国防科学技术大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档，允许论文被查阅和借阅；可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密学位论文在解密后适用本授权书。）

学位论文题目：基于 IPSec 的 VPN 网关设计与实现

学位论文作者签名：杨文武　　日期：2008 年 6 月 13 日

作者指导教师签名：王小宏刚　　日期：2008 年 6 月 13 日

第一章 绪论

随着计算机网络和通信技术的不断发展，计算机网络在军队中得到了广泛的应用，并渗透到军队工作的各个方面。通过网络连接的计算机用户数量日益增多，网络的开放性和共享性也在不断的扩大。现在计算机已经成为军队工作中不可缺少的重要组成部分。但是，在军队充分享受计算机网络带来的方便和快捷的同时，计算机网络的安全问题也越来越变得重要，成为研究的热点问题。

1.1 课题背景

随着网络技术的发展，网络环境变得越来越复杂，这就产生了各种各样的安全问题。而网络安全问题，直接关系到军事领域的安全和稳定，成为军队关注的焦点。在军队，虽然网络普及的比较晚，但是普及的速度却相当快，目前，无论是总部机关，还是军、师、团及各分队，都已经将网络深深地融入其中，成为必不可少的一部分。在这种情况下，军队的秘密信息，内部核心军事资料，都需要重要的保护。在共享网络资源的同时，安全问题成为军队网络无法回避的现实问题。近几年的军事秘密泄露事件中，有将近三分之一的案件与网络安全有关，而且网络犯罪有发生快、扩散面广、影响面大等特点。而网络所具有的开放性、国际性和自由性，对安全提出了更高的要求。这主要表现在：

开放性的网络，意味着任何一个人或团体都可以获得网络的技术，因而导致网络所面临的破坏和攻击是多方面的。例如：可以是来自物理传输线路上的攻击，也可以是对网络通信协议和实现实施攻击；可以是对软件实施攻击，也可以是对硬件实施攻击。

国际性的网络，意味着网络的攻击不仅仅来自本地网络的用户，还可以来自网络上的任何一台机器，也就是说，网络安全所面临的是一个国际化的挑战。

自由性的网络，意味着网络最初对用户的使用并没有提供任何的技术约束，用户可以自由地访问网络，自由地使用和发布各种类型的信息。

开放的、国际化的、自由的网络的发展给军事领域带来了革命性的变革，利用网络提高办事效率、快速反应能力和竞争力。同时又要面对网络开放所带来的数据安全的新挑战和新危险。保护军队网络安全稳定，已成为影响军队信息化建设健康发展的重要因素之一。

1）军队网络特点

军队网络是实现部队作战指挥网络化、自动化的主要手段。目前主要用于部队的作战指挥、通信、训练和管理等方面。

其特点是：

- 数据安全性要求高；
- 数据存储量大；
- 可用性要求高。

军队网络对数据安全及可用性有很高的要求，既能够在存储大量数据的同时具有较高的保密性、容错性和容灾性能。需要在构建军队网络时，采用一系列相关技术，如网络存储技术、容错技术、容灾技术、网络加密技术等。

2）军队网络数据安全面临的威胁

- 黑客攻击。这里所谓的“黑客”是指那些非组织的、仅出于个人目的或纯粹是出于好奇而非法访问、修改、删除系统或系统敏感数据的人。黑客入侵是目前计算机网络安全的最大威胁之一，对军用计算机网络来说尤为严重。黑客入侵主要采用两种基本方法：社会工程和技术入侵。基于社会工程的入侵方法是黑客通过欺骗手段获得用户口令而轻易地进入网络系统。调查表明，许多黑客就是简单地利用网络管理人员和用户的轻信和马虎而成功进入系统的。基于技术入侵的方法是黑客利用系统设计、配置和管理中的漏洞来入侵系统。任何一种软件系统都或多或少地存在着安全漏洞。而且，在现有条件下发现和修补一个系统的所有的漏洞是十分困难的。一个系统可能存在的安全漏洞主要有：口令漏洞，协议漏洞，缓冲区溢出和拒绝服务。基于技术入侵攻击的基本做法是，通过施放电脑病毒或采用各种远程控制手段攻击对方电脑系统，访问、修改电脑中的文件和数据，安装“逻辑炸弹”，传送虚假命令，或者直接破坏电脑系统使其瘫痪等。

- 网络的缺陷。因特网的共享性和开放性使网上信息安全存在先天不足。因为其赖以生存的 TCP/IP 协议族，缺乏相应的安全机制，因特网与大多数包交换网络一样都是建立在 Internet 协议（IP）基础上的，然而，IP 本身是不安全的。截获传输中的 IP 包比较容易，修改和重放 IP 包而不被目的主机发现也比较容易，IP 使用一个 16 比特的头校验和来验证 IP 数据报的完整性，这个是非常初级的安全机制，因为在修改数据包之后可以重新计算校验和，并把新的校验和重新插入到校验和头域，因此无法保证 IP 包来自它所声称的出处，也无法保证它在从起始地到目的地的传输过程中没有被修改。而且因特网最初的设计考虑是该网不会因局部故障而影响信息的传输，基本没有考虑安全问题，因此它在安全可靠、服务质量、带宽和方便性等方面存在着不适应性。

- 信息战。信息战是一种全新的战争模式，是利用现代信息技术手段，通过夺取信息资源的优势来达到克敌制胜的目的。信息战的内涵非常丰富，包括病毒与防毒杀毒、计算机辐射与屏蔽、侦查与反侦察、信息加密与解密、信息隐藏与提取、远程遥控与入侵探测、信息窃取等。信息战与黑客攻击有许多共同之处，但也存在

许多明显的不同：信息战不但注重对敌方网络通信系统的攻击，而且更注重对本方系统的防卫；信息战比黑客攻击更具有时间上的精确性、目标上的针对性、关系上的敌对性；信息战中使用的攻击手段更隐蔽、多样，除了各种黑客方法以外，还包括窃听、间谍渗透、秘密收集机密载体废弃物、利色引诱甚至暴力胁迫、投放电磁弹等。

- 物理安全的威胁。另外，火灾、水灾、雷击、地震、蓄意攻击等因素也严重威胁军队网络的物理安全。军队网络作为运行大量关键应用的专用网络，以上这些不可预测威胁的存在，使军队网络的安全性和可用性值得特别的关注。

3）军队网络建设中应关注的问题

针对以上所说的各种来自外部和内部的威胁，结合部队计算机网络自身的特点，成功应对目前和将来的各种挑战，军队网络在系统建设和日常管理，特别是网络信息安全方面应重视以下几个方面的工作：

- 建立完善的网络信息安全保障体系，网络必须有足够的安全性。网络安全包括网络设施自身和网络中所存储的数据不因各种自然的和人为的因素的破坏、非法访问而损毁或泄密。当前，自然灾害、恐怖袭击和电子战攻击（包括黑客攻击）是网络，特别是专用网面临的最大威胁。

网络安全不仅是安全设施，而且是个安全过程；不仅是技术防护和一般的内部管理，而且是一个全体人员和设备的综合集成体系；不仅要静态地防护，而且要动态地适应。根据军队中不同系统和应用所担负的作用和其重要程度，分析和研究各个系统和应用所面临的主要威胁，进而明确其信息安全的主要任务，并建立相应的网络信息安全防护体系，完整的安全防护体系应当包括网络安全的全过程，即安全防护、动态检侧和安全反应等环节。安全防护通过防火墙、访问控制和身份认证等技术提供基本的安全保护，安全防护一方面要保证合法用户对网络资源的正常访问，另一方面要保护网络不受非法用户的入侵，它是网络安全的基础；动态检测通过漏洞扫描和入侵检测等技术对网络安全状况进行监控和检测，及时发现攻击者的入侵行为，并适时发出安全警告；安全反应在系统遭受攻击时通过阻断攻击和灾难恢复等技术适时的作出反应，将损失减少到最底限度。以上三个方面应当协同工作，形成一个有机整体。

- 网络必须具备一定的可用性。可用性是建立在安全性、容错性和容灾性之上的、对网络更高层次上的要求。高可用计算技术也是当前网络技术发展的一个重要方面。Internet 上的 WEB 服务、电子政务以及各种军用网络对高可用性都有着特别的需要。

Internet 以 IP 协议为通讯基础，但由于 IP 协议最初的设计是基于一个互信的环境，没有考虑安全问题，从而使协议自身存在安全的脆弱性。TCP/IP 协议安全脆弱

性主要表现在：首先缺乏有效的认证机制，没有验证通讯双方真实性的能力；其次不能保护网络上传输数据的私密性；再次不能提供传输数据的完整性的保护[1][2]。

由于 Internet 是一个开放的网络环境，即一个不安全的网络环境，存在着大量的安全威胁，如：IP 地址欺骗，数据截获，数据篡改[3][4]。为了保护网络上传输的私有数据的安全性，以往企业往往采用的做法是建立企业私有网络，以避免在公共网上传输的私有数据的安全性，但这样的做法要承担高额的通讯费用，而且网络的建设周期长，灵活性差。于是（VPN）虚拟专用网孕育而生。简单地说，VPN 就是利用公共网络构建自己的专用网络。这里，“虚拟”的概念是相对于传统私人专用网络的构建方而言的。对于广域连接，传统的组网方式是通过长途拨号或专线连接来实现，而 VPN 是利用服务提供商所提供的公共网络来实现远程的广域连接。VPN 利用某些机制，使在公共网络上传输的数据也能得到与企业私有网络上同样的安全性和良好的可管理性。在长途通信方面，VPN 利用 Internet 作为传输媒介，比传统租用专线或拨号的方法节省了大量费用。基于公用网构建安全的虚拟专用网（VPN），不仅可以减少重建专用网络的投资，还可以帮助特殊用户建立安全的通信环境，同时为已经构建专用网络的用户提供一条安全的后备通道。因此，建立具有自身特色的 VPN 解决方案，对于计算机网络的安全与保密通信都有很重要的理论与实际运用价值。

1.2 VPN 网络基本原理

对于任何跨区域的单位，都希望得到本单位内部的网络服务，同时合作的单位与客户也希望得到这种方便快捷、更经济、高效、安全的网络服务，专用网就是解决这一问题的途径。

一般来说，VPN 是指利用公共网络，如公共分组交换网、帧中继网、ISDN 或 Internet 等的一部分来发送专用信息，形成逻辑上的专用网络。也可以称 VPN 是专网和公网的折中方案，通过这种技术可以将原来独立的 LAN 通过 WAN 安全地连接在一起。当前构建 VPN 网关主要是依据 IPsec 协议[5][6][7]。

1）VPN 的发展背景

随着企业的收购与合并，再加上企业本身的发展壮大与跨国化，使它的分支机构越来越多，Intranet 覆盖面积也越来越大。在企业网的功能上，不但要能够满足移动用户以及远程用户访问企业网络的需求，同时还需要某种机制来保证企业信息的安全。对通信安全尤其是企业分散子网间通信安全问题传统的解决办法通过租用并独占专用线路或自建线路来连接分布在不同地域的企业子网，由于几乎是独占信道，不管用户是否传输了数据都要全时付费，因此费用比较昂贵；另外一种解决方法就是采用虚拟专用网技术，通过在公用网（主要指 IP 网络）上建立隧道等方式虚拟出

专线来连接分布在各地的企业子网甚至移动用户。由于在 IP 网络上信道主要是统计复用的，用户没有传输数据就不必计费，因此费用相对低廉，这也是推动 VPN 发展的一个动力。

总的来说，VPN 产生和发展的原因有以下几个因素：

- 日益增加的分散的公司部门和移动的成员；
- 客户和提供商的在线互连；
- 商务应用会更多的依靠因特网；
- 新的应用的涌现；
- 运行和维护专用网的费用高。

2）VPN 的定义

虚拟专用网络（Virtual Private Network，VPN）[8][9][10]被定义为通过一个公用网络（通常是 Internet）建立一个临时的安全的连接，是一条穿过公用网络的安全、稳定的隧道。VPN 是对企业内部网的扩展，其可以帮助远程用户、公司分支机构、商业伙伴及供应商同公司的内部网建立可信的安全连接，并保证数据的安全传输。VPN 可用于不断增长的移动用户的全球 Internet 接入，以实现安全连接，可用于实现企业网站之间安全通信的虚拟专用线路。

虚拟专用网络是企业网在 Inernet 等公共网络上的延伸，通过一个私用的通道来创建一个安全的私有连接，虚拟专用网络通过安全的数据通道将远程用户、公司分支机构、公司的业务合作伙伴等与公司的企业网连接起来，构成一个扩展的公司企业网，如图 1.1 所示。Internet 服务提供商（SP）提供高性能、低价位的 Internet 接入（直接通过线路或本电话号码），这样公司就可以摆脱以前使用的昂贵的租用线路。

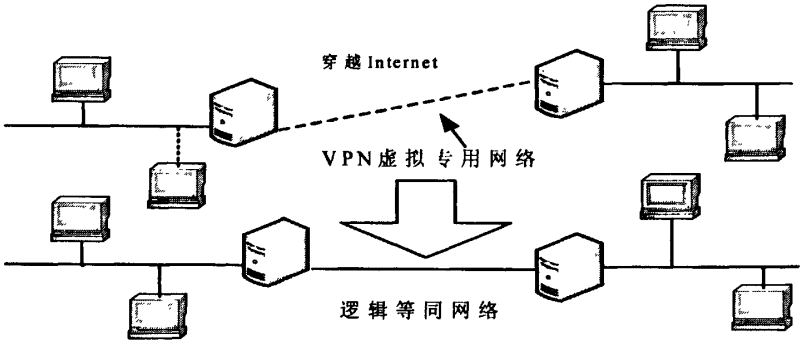

图 1.1　虚拟专用网络

实现虚拟专用网络的技术已经开始标准化，并且 Internet 工程任务小组（IETF）的虚拟专用网络标准已经制订。基于这一标准的产品，将使各种应用场合下的虚拟专用网络有充分的互操作性和可扩展性。

Internet 已经成为一个大众化的、低价格的骨干体系结构。它已经深入当今社会的各个方面，所以有许多公司已经开始设想在 Internet 的基础上建立一个安全的 VPN。设计利用 Internet 等公共网络进行公司内部和公司之间的通信，同时要提供像传统专用网络那样的安全性，就要结合 Internet 和传统专用网络各自的优点。

提高虚拟专用网络效用的问题在于当用户的业务需求发生变化时，用户能很方便快速地调整他的虚拟专用网络以适应变化，并且能方便地升级到将来新的 TCP/IP 技术。而那些提供门类齐全的硬软件虚拟专用网络产品的供应商，则能提供一些灵活的选择以满足用户的要求。现今的虚拟专用网络产品主要是运行在 IPv4 之上的，但是它们具备升级到 IPv6 的能力，同时能保持良好的互操作性。

3）VPN 的关键技术

VPN 技术是指采用隧道技术以及加密、身份认证等方法，在公众网络上构建专用网络的技术，数据通过安全的“加密管道”在公众网络中传播。VPN 技术实现了内部网信息在公众信息网中的传输，就如同在茫茫的广域网中为用户拉出一条专线。对于用户来讲，公众网络起到了“虚拟专用”的效果。通过 VPN，网络对每个使用者也是专用的。也就是说，VPN 根据使用者的身份和权限，直接将使用者接入他所应该接触的信息中。所以 VPN 对于每个用户，也是“专用”的，这一点应该是 VPN 给用户带来的最明显的变化。

关键技术[11]主要包括：

- 安全隧道技术

隧道是构建 VPN 的基础，如图 1.2。隧道代替了传统的 WAN 互联的“专线”。图中以 Internet 为公共传输媒介。通过隧道，可以构建一个 VPN。图中隧道的两边是 Intranet，也可以一边是单个的主机，另一边是 Intranet，甚至两边都是单个的主机。从图中可以看出，隧道需要对进入其中的数据加以处理。这里有两个基本的过程：加密和封装。通过将待传输的原始信息经过加密和协议封装处理后再嵌套装入另一种协议的数据包送入网络中，像普通数据包一样进行传输。经过这样的处理，只有源端和宿端用户对隧道中的嵌套信息进行解释和处理，而对于其它用户而言只是无意义的信息，这里采用的是加密和信息结构变换相结合的方式，而非单纯的加密技术。

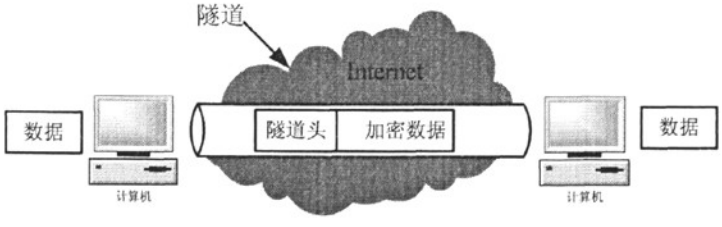

图 1.2 隧道示意图

● 用户认证技术

在正式的隧道连接开始前需要确认用户的身份，以便系统进一步实施资源访问控制或用户授权。用户认证技术是相对比较成熟的一类技术，因此可以考虑对现有技术的集成。

● 访问控制技术

由VPN服务的提供者与最终网络信息资源的提供者共同协商确认特定用户对特定资源的访问权限，以此实现基于用户的细粒度访问控制，实现对信息资源的最大限度的保护。

4）VPN的主要安全协议

在实施信息安全的过程中，为了给通过非信任网络的私有数据提供安全保护，通讯的双方首先进行身份认证，这中间要经过大量的协商，在此基础上，发送方将数据加密后发出，接受端先对数据进行完整性检查，然后解密，使用。这要求双方事先确定要使用的加密和完整性检查算法。由此可见，整个过程必须在双方共同遵守的规范（协议）下进行。

VPN区别于一般网络互联的关键是隧道的建立，数据包经过加密后，按隧道协议进行封装、传送以保证安全性。一般，在数据链路层实现数据封装的协议叫第二层隧道协议，常用的有PPTP、L2TP、L2F等；在网络层实现数据封装的协议叫第三层隧道协议，如IPSec；另外，SOCKSv5协议则在TCP层实现数据安全。

5）VPN发展现状

VPN高性价比及灵活性等优势，使之具有巨大市场潜力。美国通信杂志将IPVPN技术评选为2001年的十大热门技术之一。我国的信息产业部已经将IPVPN业务明确定义为一种电信增值业务，向社会开放。

VPN的发展代表了互联网络今后的发展趋势，它综合了传统数据网络的安全和服务质量，以及共享数据网络结构的简单和低成本，建立安全的数据通道。VPN在降低成本的同时满足了用户对网络带宽、接入和服务不断增加的需求，因此，VPN必将成为未来网络发展的主要方向。尤其是基于IPSec协议[12][13][14][15]的VPN，因为网络安全问题已经成为困扰整个世界的大难题，没有可靠安全保障的网络是个可以随时爆炸的火药桶，这样也就限制了网络的应用和普及。基于IPSec协议的VPN网络很好地解决了安全性问题，提供了多种手段保证了通信的安全，成为VPN的发展方向。

目前大型网络厂商都把VPN作为重要市场目标，诸如3Com、Cisco、NortelNetworks、Lucent和Shiva公司等纷纷出击，提供各具特色的VPN解决方案。然而由于信息安全领域的特殊性，受我国进出口和计算机及信息安全法律限制，国内VPN市场则必将逐渐由国内的厂商来占领。现在随着国家安全部、国家公安部以

及信息产业部等信息安全指导部门相关规定和指导办法的出台，信息安全已被提到了一个更加显著和紧迫的地位上。目前，国内市场对信息安全的需求日益强烈，尤其是规模客户对网络安全的需求越来越紧迫。

1.3 本文研究内容

本文的主要研究内容：

1)针对师仓库部队网络安全需求，深入分析了 VPN 技术和 IPSec 协议，对 IPSec 的复杂性提出了一套改进方案，并对该方案的实现方法进行了研究，设计了一个基于 IPSec 协议的 VPN 网关，为在实际应用环境中更好地利用 IPSec 保证 VPN 的通信安全提供了一种新思路。

2)实现了基于 IPSec 协议的 VPN 网关中 IKE 模块、SAB 模块和 SPD 模块、IPSec 处理模块以及策略和 SA 管理模块的详细设计。并对安全策略和安全关联的管理方面设计了图形界面，建立了良好的用户接口。给出了网关实现需用到的具体函数。

3) 对基于 IPSec 协议硬件加速网关处理以及 IPSec VPN 在实际运用中的多协议问题进行了研究。

1.4 论文的组织结构

论文各章节内容组织如下：

第一章阐述了本课题的研究背景，包括军队网络安全的特点和面临的挑战，VPN 的发展背景、定义、关键技术、安全协议、发展现状，提出了本文研究内容，最后介绍了论文的组织结构。

第二章对 IPSec 安全协议及相关技术进行介绍。分别介绍了 IPSec 协议安全体系结构，IPSec 的传输模式和隧道模式，验证头协议 AH，封状安全载荷协议 ESP，IKE 协议，对其进行系统的分析研究，理解 IPSec 协议各个组成部分的工作原理和过程，并对验证头协议 AH 和封状安全载荷协议 ESP 进行了比较。

第三章针对师仓库部队网络保障系统的使用需求及其网络应用，在通过对各种 VPN 实施方案的优缺点进行对比分析之后，选择了与操作系统集成的 VPN 实施方案。以 IP 安全协议体系 IPSec 为基础，面向网关到网关的 VPN 应用环境，针对 IPSec 协议的特点提出一个 IPSec 协议实现的思路，并对该思路的实现方法进行了研究，设计了一种基于 IPSec 协议的 VPN 网关。并设计了网关的应用层、内核层和 IPSec 进出处理。

第四章实现了基于 IPSec 协议的 VPN 网关中 IKE 模块、SAB 模块和 SPD 模块、IPSec 处理模块以及策略和 SA 管理模块的详细设计。并对安全策略和安全关联的管

理方面设计了图形界面，建立了良好的用户接口。给出了网关实现需用到的具体函数。

第五章，对基于 IPSec 协议硬件加速网关处理以及 IPSec VPN 在实际运用中的多协议问题进行了研究。

第二章 IPSec 技术分析

随着网络的蓬勃发展，安全问题已变得至关重要，在这种形势下，IETF 的 IP 安全协议工作组研究开发了 IP 安全协议套件 IPSec 协议[16][17][18][19]，从而为 IP 无缝地引入了安全特性。

本章介绍了 IPSec 协议安全体系结构、安全关联、安全策略数据库、运行模式、验证头协议、封装安全载荷协议、IKE 协议及验证头协议与封装安全载荷协议在认证、保密和防重放方面的不同。

2.1 IPSec 协议

IPSec 协议是一套在网络层为 IP 协议的各个高层协议（如 TCP、UDP、ICMD、BGP 等）提供安全性保护的协议。它提供了一种标准的、健壮的以及包容广泛的安全机制，可以用它为 IP 及上层协议（如 UDP 和 TCP）提供安全保证。它定义了一套默认的、强制实施的算法，以确保不同的实施方案相互间可以共通。而且如果增加新的算法，其过程也非常直接的，不会对共通性造成破坏。

2.1.1 IPSec 安全体系结构

IPSec 体系结构由一系列 RFC[20]文档定义。除 RFC2401 外，包括 RFC2402 验证头、RFC2406 封装安全载荷、RFC2407 用于 Intemet 安全联结和密钥管理协议 ISAKMP 的 Intemet IP 安全解释域、RFC2408 ISAKMP，RFC2409 Internet 密钥交换 TKE，RFC2411 安全文档指南、RFC2412 OAKLEY 密钥确定协议等。如图 2.1 显示了整个 IPSec 协议族的体系结构、组件及各组件间的相互关系。IPSec 组件包括安全协议验证头 AH[21][28]和封装安全载荷 ESP[22][29]，安全关联 SA.密钥交换 IKE 及加密和验证算法[23][24][25]等。

安全体系结构包含了一般的概念、安全需求、定义和定义 IPSec 的技术机制，通过这些技术机制的共同作用便可以实现 IPSec 的安全功能。

ESP 协议：覆盖了为了包加密（可选身份验证）与 ESP 的使用相关的包格式和常规问题；

AH 协议：包含使用 AH 进行包身份验证相关的包格式和一般问题；

加密算法：描述各种加密算法如何用于 ESP 中；

验证算法：描述各种身份验证算法如何用于 AH 中和 ESP 身份验证选项；

密钥管理：密钥管理的一组方案，其中 IKE 是默认的密钥自动交换协议；

解释域：彼此相关各部分的标识符及运作参数；

策略：决定两个实体之间能否通信及如何进行通信。策略的核心由三部分组成：SA、SAD、SPD[26][27]。SA 表示了策略实施的具体细节，包括源/目的地址、应用协议、Spi（安全策略索引）、所用算法/密钥/长度；SAD 为进入和外出包处理维持一个活动的 SA 列表；SPD 决定了整个 VPN 的安全需求，策略部分是唯一尚未成为标准的组件。

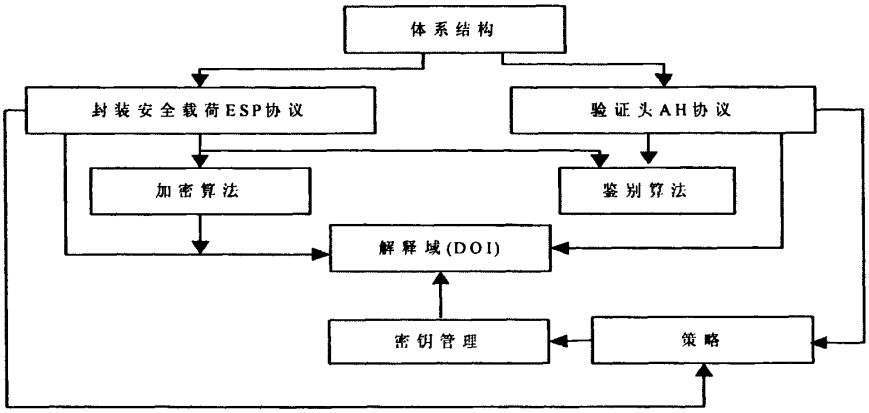

图 2.1　IPSec 体系结构

2.1.2 安全关联

安全关联（Security Association，SA）是构成 IPSec 的基础。要进行安全通信，需要采用身份鉴别和加密服务。所以通信的双方在通信之前需协商好采用哪种安全协议、加密算法以及加密的密钥等问题。所谓的安全关联就是通信双方协商好的安全通信的构建方案，是通信双方共同签署的“协定”。

SA 是两个通信实体经协商建立起来的一种协定。它们决定了用来保护数据包安全的 IPSec 协议、密钥以及密钥的有效存在时间等。任何 IPSec 实施方案会构建一个 SA 数据库（SADB），由它来维护 IPSec 协议为数据包提供安全的 SA 记录。

安全关联是单工的，换言之，它的设计非常简化。即从业务流的发送方到接收方的一个单向逻辑关系。在典型的、双向的点到点连接中，需要提供两个 SA。如果两个主机（比如 A 和 B）正在通过 ESP 进行安全通信，那么主机 A 就需要有一个 SA，即 SA（out），用来处理外发的数据包；另外还需要有一个 SA（in）用来处理进入的数据包。由于 SA 是单向的，所以针对外发和进入处理使用的 SA，需要分别维护一张单独的表。

另外，SA 还是“与协议相关”的。每种协议都有一个 SA。如果主机 A 和 B 同时通过 AH 和 ESP 进行安全通信，那么每个主机都会针对每一种协议来构建一个独立的 SA。一个安全联盟由一个三元组唯一确定：目的 IP 地址、安全协议类型（AH 或 ESP）、安全参数索引（SPI）。它还应该包括验证密钥、加密密钥、生存时间、

抗重播窗口等一系列参数。

安全关联由如下 3 个参数唯一确定：

- 安全参数索引（Security Parameters Index，SPI）：SPI 是一个长度为 32 位的数据，接收方用 AH 和 ESP 报头的 SPI 唯一地确定一个 SA；
- IP 目的地址：即 SA 中接收方的 IP 地址；
- 安全协议标识符：用以标识通信双方采用的是 AH 协议还是 ESP 协议。

除以上 3 个参数外，SA 还包含以下参数：

- 顺序号计数器（Sequence Number Countor）：用来产生 AH 或 ESP 报头中的顺序号（Sequence Number），达到重放攻击目的；
- 顺序号溢出标志（Sequence Number Overflow）：表示顺序号的溢出是否能产生一个可审核的事件并防止这一 SA 上数据报的进一步传送；
- 防重放窗口（Anti-replay Window）：用来判断入站 AH 或 ESP 数据包是否重放。
- AH 信息（AHInformation）：所采用 AH 的身份鉴别算法、密钥、密钥生命周期和其他一些相关参数；
- SA 的生命周期（Life Time of Time SA）：表示一个时间间隔，在该时间以后，此 SA 或者结束或者被一个新的 SA 所替代。同时这一参数中还有一个标识符用来标识此 SA 是被结束还是被替代；
- IPSec 协议模式（IPSec Protocol mode）：IPSec 的协议模式有隧道、传输、通配符模式；
- 路径最大传输单元（Path MTU）：指能传输的最大数据报长度。

以上参数除了 AH 信息和 ESP 信息分别仅为采用 AH 或 ESP 协议时要求以外，其他参数在两种协议中都被要求。

在每一个 IPSec 的执行过程中，都有一个标准的安全关联数据库（Security Association Database，SAD），其中存放了每一个 SA 的相关对数。

SA 的创建分两步进行：先协商 SA 参数，再用 SA 更新安全策略数据库。协商 SA 参数可采用人工协商或 Internet 标准密钥管理协议（比如 IKE）来完成。人工密钥协商是必须支持的，在 IPSec 的早期开发及测试过程中，人工协商是一项非常有用的方式。在人工密钥协商过程中，通信双方都需要离线同意 SA 的各项参数。但人工协商过程非常容易出错，既麻烦、又不安全。因此，在已经有一种稳定、可靠的密钥管理协议的前提下，已经配置好 IPSec 的一个环境中，SA 的建立通过一种 Internet 标准密钥管理协议来完成。如果安全策略要求建立安全、保密的连接，但却找不到相应的 SA，IPSec 的内核便会自动调用 IKE，IKE 会与目标主机协商具体的 SA。

2.1.3 安全策略数据库（SPD）

安全策略决定了为一个包提供的安全服务。IPSec 将安全策略所有可能实施方案，保存在一个数据库中，这个数据库称为安全策略数据库（Security Policy Database，SPD）。根据“选择符”对该数据库进行检索，获取为一个 IP 包提供安全服务的有关信息。

在 IPSec 环境中，SPD 说明了对 IP 数据报提供何种保护，并以何种方式实施保护，是 SA 处理过程的核心部分。

对于一个 IPSec 实施点，无论是正常 IP 数据包还是 IPSec 数据包，无论进入包和外出包都需要参考 SPD，从而决定哪些通信流需要进行 IPSsec 保护。对于进入或外出的每一份数据报，都可能有三种处理：丢弃、绕过或应用 IPSsec。丢弃是指主机（或网关）对数据报不做进一步的处理。绕过是指允许数据报在通过 IPSsec 实施点时不进行 IPSsec 处理。应用是指数据报在通过 IPSsec 实施点时进行 IPSsec 处理。对于这些需要处理的数据报，SPD 应明确地指出应该提供的安全服务、协议类型（AH 或 ESP）、应该使用的算法等等信息。

在实现时，SPD 应包含一个有序的策略项列表。每一个策略可以有一个或多个选择符来指定。每一个项的内容中包含了对相应的通信流应该实施的策略：绕过、丢弃、处理。如果需要实施处理策略，则从 SPD 项中提取 SA 的相关描述：目的地址、SPI，IPSsec 协议，从而选择一个 SA 或 SA 束。

一条安全策略可以要求一个或多个 SA 应用于一个指定的通信流上。因此 SPD 中的策略必须在必要时保存这些 SA 的顺序要求。在实现时必须可以为输入输出包定义一个 SA 处理序列。对于外出处理，SPD 中的每一项可以由单个的 SA 组成，也可以由一个有序的 SA 束组成。对于进入处理，被应用了多个 SA 的 IPSsec 包，将根据三元组：源数据包的目的地址；协议；SPI 来指定唯一的耽搁 SA。从而检验进入策略实施的正确性。

同时，SPD 应提供管理接口，便于用户或系统管理员对 SPD 进行维护。该接口允许用户或管理员对每一个进入或外出包都应作出相应的策略说明，并支持通过选择符对 SPD 添加有序的策略项。

IPSec 要求在所有通信流处理的过程中都必须查询 SPD，不管通信流是输入还是输出。SPD 中包含一个策略条目的有序列表。通过使用一个或者多个选择符来确定每一个条目。IPSee 目前允许的选择符有：

- 目的 IP 地址：目的 IP 地址是一个 32 位的 IPv4 或者 128 位的 IPv6 地址。该地址可以是一个主机 IP 地址，广播地址，单播地址，任意播地址，多播 地址，地址范围，地址加子网掩码或者通配地址。目的 IP 地址从 AH，ESP 或者 IP 头（如果没有对数据包应用 IPSec 的话）的目的 IP 地址字段中得到。
- 源 IP 址：同目的 IP 地址一样，源 IP 地址可以是一个 32 位的 1Pv4 或者 128

位的 IPv6 地址。同样，该地址可以是一个主机 IP 地址，广播地址，单播地址，任意播地址，多播地址，地址范围，地址加掩码或者通配地址。源 IP 地址从 AH，ESP 或者源 IP 地址字段中得到。

- 系统名：系统名可以是完整的 DNS 名或者 E-mail 地址。
- 用户 ID：操作系统使用的用户识别符。此选择值不在 IP 或 IP 上层协议字段中，但如果 IPSec 与用户处在相同的操作系统层次上，就可以用用户的 ID 作为选择值。

2.2 IPSec 的两种运行模式

IPSec 协议（包括 AH 和 ESP）既可用来保护一个完整的 IP 载荷，亦可用来保护某个 IP 载荷的上层协议。这两方面的保护分别是由 IPSec 两种不同的模式来提供的，如图 2.2 所示。其中，传送模式用来保护上层协议；而隧道模式用来保护整个 IP 数据报。

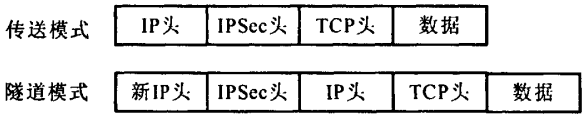

图 2.2 IPSec 的传送模式和隧道模式

2.2.1 传输模式

传输模式用来保护 IP 载荷；IP 头与上层协议头之间需插入一个特殊的 IPSec 头；其通信的终点必须是一个加密的终点，用于保障端到端的通信安全。如果没有启动安全保护时，传送层数据包会流入网络层，IP 协议为数据包增加 IP 头，然后转到链路层。如果启动了安全保护，传送层的包会流入 IPSec 组件。IPSec 组件作为网络层的一部分来实现（当与 OS 集成在一起时），为数据包增加 AH 或 ESP 或两个头都增加。随后，调用网络层的一部分，并增加网络层的头。

传送模式中同时使用 AH 和 ESP 时，应首先使用 ESP，再用 AH 重新保护一遍数据，数据的完整性就能同时应用于 ESP 载荷。

在传输模式中，IPSec 先对上层协议进行封装，增加一 IPSec 头，对上层协议的数据进行保护，然后才由 IP 协议对封装的数据进行处理，增加 IP 头；而在隧道模式中，IPSec 对 IP 协议处理后的数据进行封装，增加一 IPSec 头，对 IP 数据报进行保护，然后再由 IP 协议对封装的数据进行处理，增加新 IP 头。

传输运行模式的实施如图 2.3 所示。

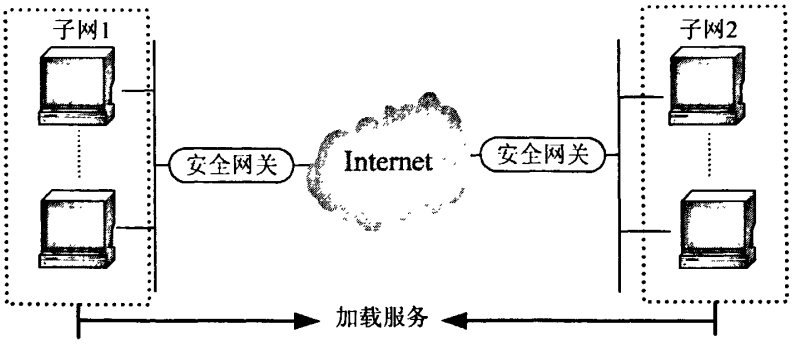

图 2.3 传输模式的实施

在传输模式下，IPSec 模块运行于通信的两个主机。在此种模式下的 IPSec 有如下优点：

即使位于同一子网内其他用户，也不能非法修改通信双方的数据内容。分担了安全网关的处理负荷。

但同时也具有以下缺点：

每个需要实现传输模式的主机都必须安装并实现 IPSec 模块，因此端用户无法得到透明的安全服务，并且端用户为获得 AH 服务必须付出内存、处理时间等方面的代价。不能使用私有的 IP 地址，必须使用公有地址资源。

2.2.2 隧道模式

隧道模式用来保护整个 IP 数据包，要保护的整个 IP 包都封装到另一个 IP 包里，同时在外部和内部之间插入一个 IPSec 头，而隧道模式在被网关使用时，可用来保护与其连接的网络实体，虚拟专用网络中采用的便是隧道模式。在隧道模式中，通信的终点是由受保护的内部头指定的地点，而加密终点则是那些由外部 IP 头指定的地点。IPSec 处理结束后，安全网关会剥离出内部 IP 包，再将那个 IP 包转发到它最终目的地。

在隧道模式中，数据包的内部头是由主机创建，外部头是由提供安全服务那个设备（即可是主机，也可是路由器）添加的。IPSec 支持嵌套隧道，所谓嵌套隧道就是对一个已经隧道化了的数据包再进行一次隧道化的处理，从而支持多级网络安全保护。例如:一家公司有一个安全网关 ，为防止被竞争者和黑客的侵犯，在该公司网络内部另有一个安全网关，防止某些内部员工进入敏感子网。此时，若对一个保护网络内部的保护子网进行访问就要用到嵌套式隧道。但嵌套式隧道难以构建和维护，目前，多级的嵌套隧道应用不广。

隧道运行模式的实施如图 2.4 所示。

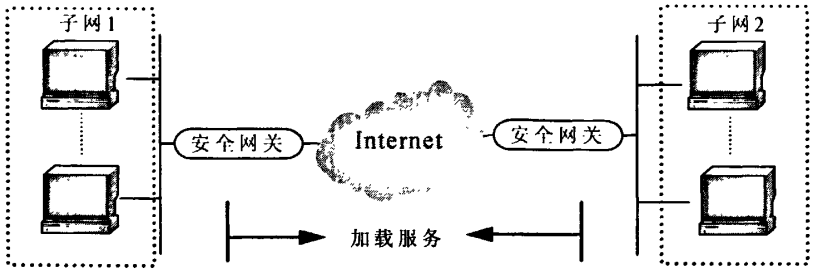

图 2.4 隧道运行模式实施

采用隧道模式，IPSec 模块运行于安全网关或主机，具有以下优点：

- 子网内部的各主凭借安全网关的 IPSec 处理透明地得到安全服务。
- 可以在子网内部使用私有 IP 地址，无需占用公有地址资源。

但同时 IPSec 也具有以下缺点：

- 增加了安全网关的处理负载。
- 无法控制来自网内部的攻击者。

2.3 验证头协议 AH

1）验证头协议规范

验证头协议 AH（Authentication Header）是一种 IPSec 协议，用于为 IP 提供无连接完整性、数据原始身份验证和一些可选的、有限的抗重播攻击服务，能保护通信免受篡改，但不能防止窃听，适合用于传输非机密数据，它的定义在 RFC2402 中。

AH 可用来保护一个上层协议（传输模式）或一个完整的 IP 数据包（隧道模式），即对数据采用认证算法，使终端能确定通过验证的数据是没有受到篡改攻击的。AH 与 ESP 最大的区别就是它不提供机密性服务。

2）报文格式

AH 的协议号是 51[28]。即紧靠在 AH 头之前的协议头的下一个头字段就会被设成 51。AH 头格式如图 2.5 所示。

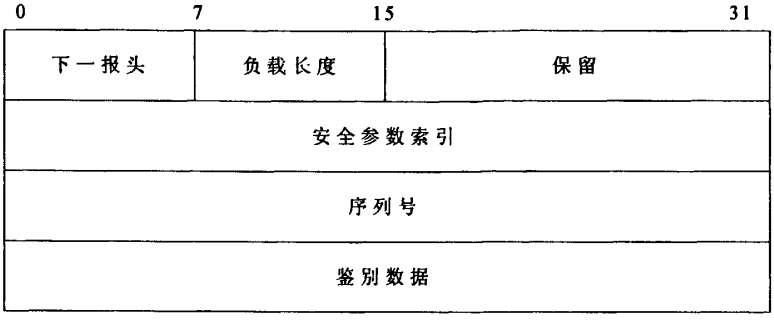

图 2.5 AH 报头格式

其中的字段意义：

- 下一报头（Next Header）：为 8 位的字段，指明 AH 头之后的载荷类型。例如值为 4 表明下一个载荷是 IPv4，41 表示 IPv6，6 则指明是 TCP。
- 负载长度（Payload Length）：这是一个 8 位的字段，其值是 32 位字的 AH 报头长度减 2。例如鉴别字段的默认长度是 96 位，即 3 个 32 位字，而 AH 报头要占 3 个 32 位字的固定长度，此时 AH 报头长度为 6 个字，负载长度字段的值为 4。
- 保留（Reserved）：这是个 16 位的字段被保留为将来使用，必须将它设成 0。
- 安全参数索引（Security Parameters Index, SPI）：SPI 是一个任意的 32 位值，被接受者用来识别对进入包进行身份验证的安全联盟 SA。对一个单播的 SA 来说，SPI 本身就可确定一个 SA，或与 IPSec 协议类型一起确定 SA。这是因为 SPI 的值是由接收端产生的。这时 SPI 是必需的字段。对多播的 SA 束来说。SPI 和可选的 IPSec 协议号加上目的地址一起来指定一个 SA。因为多播的 SA 束是由一个多播控制器产生，而不是 IPSec 接受者产生。其中 1~255 之间的 SPI 值被 IANA 保留将来使用。值 0 为内部保留值，在实际传输过程，IP 数据包的 SPI 不能取 0 值。如果一个新的 SA 尚未建立好，即它的密钥还在通信双方协商之时，这时，该 SA 内部的 SPI 值要取为 0。
- 序列号：是一个 32 位的单向递增的计算器。对一个单播 SA 或一个单发送者的多播 SA 来说，发送者每传送一个包都必须增加这个字段的值;在多发送者间要共享 SA 以避免灾难。这个字段是强制的，必须一直存在，即使接收者不选择抗重播攻击服务。当 SA 建立时，发送者和接受者的 SA 被初始化为 0，而且计数器的值不允许循环，因此，当一个 SA 传送了 2^{32} 个包时，计数器必需被重置。为了支持高速的 IPSec 实现，一个 64 位的扩展序列号（ESN）可被使用。这个序列号只有低 32 位用于每个包的 AH 头中被传送，高 32 位被包括在 ICV 值中，不被传送。
- 鉴别数据（Authentication Data）：其长度可变，但必须是 32 位字的整数倍，其中包含完整性检查值（Integrity Check Value，ICV）。

3）AH 协议首部处理

由于 IPSec 存在传输模式和隧道模式两种模式，AH 验证头的位置也不相同。

AH 用于传输模式时，保护的是端到端的通信。AH 头紧跟在 IP 头之后和上层协议之前，对这个数据包进行保护。在隧道模式下，内层的 IP 头含有该 IP 数据包的最终目的地址和最初源地址，外层的 IP 头可能含有与内层不同的地址，如隧道网关的地址。在隧道模式下，AH 协议保护了整个内层 IP 数据包。与传输模式类似，AH 头的位置也是紧接在最外面的 IP 头之后。一个 IP 数据包在经 AH 模式处理后的形式如图 2.6 所示。

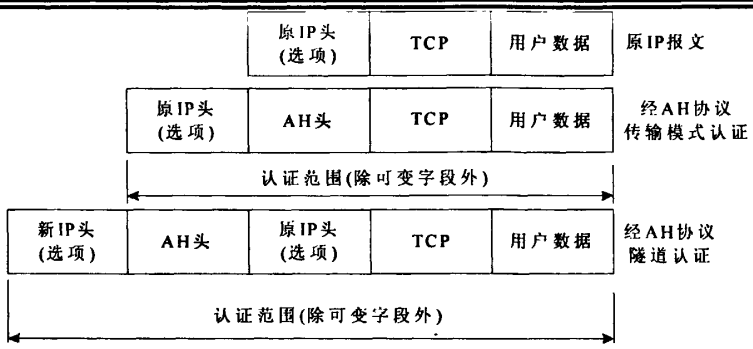

图 2.6 AH 报头格式

2.4 封装安全载荷协议 ESP

1）封装安全载荷规范

封装安全载荷协议 ESP（Encapsulating Security Payload）是 IPSec 安全协议的一个重要组成部分，它是插入 IP 数据包内的一个协议头，以便为 IP 提供机密性、数据源验证、抗重播、数据完整性以及有限的流量控制等安全服务。它可以单独使用，也可以与 AH 协议一起使用。

一个 IP 数据包所用的具体 ESP 服务由相应的安全联盟规定，保密服务是 ESP 的主要功能，数据源认证和完整性认证（统称认证）作为一个整体，是 ESP 的可选服务。防重播功能仅在有 ESP 认证时生效，并且具体处理取决于报文的接受方。流量保密需要使用 ESP 隧道模式，一般在安全网关处实施，这样可隐藏报文的实际收发地址。

2）报文格式

ESP 的协议号是 50[29]，ESP 头包的格式如图 2.7 所示：

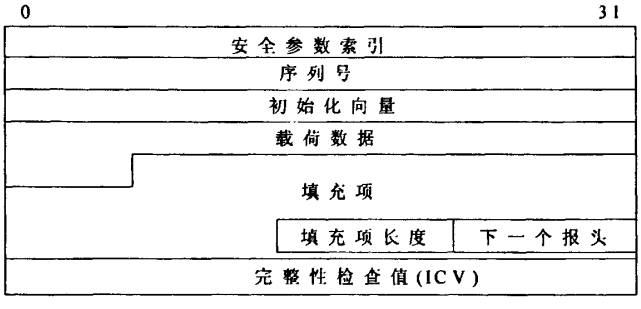

图 2.7 ESP 报头格式

相关字段的说明如下：

- 安全索引值（Security Parameters Index–SPI）：为数据包识别安全关联。是一个 32 为任意值，与目的地址及 ESP 中的安全协议字段相同，唯一地为数据报指定一个安全关联（SA），指定通信中使用的安全参数。

● 序列号（Sequence Number）：与 AH 协议一样，这一序号也是一个递增计数器的值，用来防止重播攻击。

● 初始向量（Initializated Vector）：是一个可选的 32 位字段。ESP 载荷中，依据 IPSec DOI 和所选用的加密算法，会出现以下三种不同的情况：

加密算法不需要初始向量（IV）；

加密算法需要隐式的（Implicit）IV；

加密算法需要显示的（Explicit）IV。

对于前两种情况，载荷数据中不会含有 IV 数据。但如果加密算法需要隐式的初始化向量，则在相应的 DOI 中必须给出由一个 ESP 封装载荷计算初始化向量的方式。而最后则必须以 IV 数据开始，接下来的数据因 ESP 封装的模式而异。对传输模式，为上层协议数据部分；对隧道模式，为整个 1P 分组。如果所选的加密算法需要 64 位初始向量，则相应 DOI 应说明由 32 位 IV 计算 64 位 IV 的规则。但一般都采用逐位复制的方式。

● 载荷数据（Payload Data）：用来存放经 ESP 协议处理过的数据，这些数据所属的类型由“下一个头”字段定义。如果 ESP 算法在加密时需要算法同步数据(例如初始向量)，那么，这些同步数据也包含在载荷数据中。

● 填充数据（Padding Data）：由下面原因 ESP 需要填充字段。

加密算法需要明文长度为分组块长度的整数倍；

填充字段还能保证上层协议字段的右边界以 4 字节对齐；

另外，通过使用填充字段，ESP 协议能有效地隐藏实际载荷的长度，从而提供一定的流量保密性。但是，这种填充方式会浪费线路的有效带宽，因此在使用时一定要权衡利弊。填充字段不超过 255 字节。

如果 ESP 需要填充字段，但是加密算法没有规定填充的内容，那么就要依照 ESP 协议的缺省规定执行：填充字段为无序列号单字节整形串，第一个字节值取 1 开始，以后依次增加，形成序列（1，2，3....）如果采用这种填充方式，在完成解密后，收方应该检查填充字段的内容，从而在一定程度上防止“剪贴”攻击。

● 下一个头（Next Header）：是一个 8 位字段，指出 ESP 封装的模式，该字段必须出现。ESP 载荷总是 IP 分组的最后一个载荷。这里下一载荷表示 ESP 封装的协议数据类型，为一个 IP 头、IP 扩展头，或者是上层协议头。

● 完整性检查值（Integrity Check Value）：是对整个 ESP 载荷的一个单向散列输出。但这个字段是可选的。如果选择了加密服务，按照选用的 ESP 模式，整个分组或上层协议被加密;如果选用了鉴别服务，便能验证 ESP 载荷是否被篡改、数据源实体的身份是否真实。

3）报文处理

协议头位置：对于 IP 分组，按照不同的 ESP 封装模式，分别对应两种封装格式，如图 2.8 所示。

在传输模式中，ESP 协议将上层协议数据作为 ESP 封装的载荷数据，而原 IP 报头仍作为封装后的 IP 分组的报头。在隧道模式中，原 IP 分组被作为载荷数据封装入 ESP。

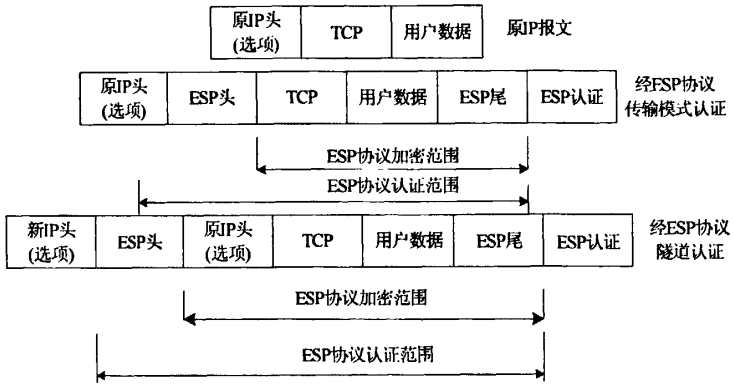

图 2.8 ESP 头位置

算法一致性要求：所谓一致性要求，即为了保证不同的策略配置、环境配置的 IPSec 能够互通，对 ESP 的实现方面的要求。

ESP 协议规定，所有 ESP 实施都必须支持：

- DES-CBC 加密算法
- HMAC-MD5 鉴别算法
- HMAC-SHA-1 鉴别算法
- 空鉴别算法
- 空加密算法

如果不同的 ESP 配置都遵从以上的一致性要求，主机间总是可以互通的（即安全联盟协调总是会成功的）。但 ESP 协议规定，任何 ESP 协议实施作用都不能使用“空加密算法/空鉴别算法”的组合。

2.5 AH 协议和 ESP 协议的比较

AH 协议和 ESP 协议都是互联层的安全协议，但二者侧重面不同。下面就认证、保密和防重放三个方面加以对比。

1）认证服务方面。AH 协议和 ESP 协议都提供认证服务功能。AH 协议是专门用以提供认证保护；ESP 协议的认证服务是它的选项。ESP 提供的认证服务范围要比 AH 的窄，即 ESP 头以前的 IP 报文部分不会被保护；而 AH 协议认证了几乎所有的 IP 报文字段。

2）保密服务方面。AH 协议不提供保密服务，当不需要保密或者法律不允许保密的情况下，AH 协议是一种恰当的安全协议，ESP 主要用于数据保密。当使用隧道模式时，位于两个网关之间的使用 ESP 保护的安全关联还可以提供一定的流量保密性。使用隧道方式时，由于内层的 IP 包被加密，所以隐藏了报文的实际源头和终点。更进一步的是，ESP 使用的填充字节隐藏了报文的实际尺寸，从而更进一步地隐藏了这个报文的外在特性。当网络中的移动用户使用动态地址，使用隧道模式的安全关联穿过安全网关时，也有类似的通信流的保密性。

3）防重放方面。AH 协议和 ESP 协议都有防止重放的功能。只要收方使能，AH 协议的防重放功能就可以可靠地工作。而 ESP 协议必须要有认证机制的配合，此项功能才能起作用。

2.6 IKE 协议

IKE 协议[30]属于 IPSec VPN 体系的动态密钥协商部分，它是可供选择的动态密钥交换机制之一，目前已经成为事实上的工业标准。IKE 协议用来进行虚拟专用网 VPN 的认证与 SA 会话密钥的协商，它沿用了 Internet 安全关联和密钥交换协议（ISAKMP）[31]的基础、Oakley 密钥确定协议[32]的模式以及 SKEME 协议的共享和密钥更新技术。IKE 协议可以动态地建立安全关联，为通信双方提供 IPSec 安全通信所需的相关信息，例如加密算法、会话密钥、通信双方身份认证等。IKE 协议为自动密钥交换奠定了框架，其高强度与可靠性是 IPSec VPN 数据安全传输的先决条件和保证。IKE 是个非常复杂的协议，整个 IKE 协议规范分为三个部分：SAKMP，IKE，DOI（Domain of Interpretation）。另外，在整个 IKE 协议规范中，ISAKMP 和 IKE 的区分是比较模糊的。

IKE 机制协商的目的是产生一个通过验证的密钥和提供双方同意的安全服务，即最终提供 IPSec 安全关联（IPSec SA）使 IPSec VPN 之间能够建立安全的数据通信隧道。IKE 通过两个阶段的协商过程来建立 IPSec 安全关联（IPSec SA）。第一阶段建立 ISAKMP SA，第二阶段利用第一阶段得到的 ISAKMP SA 建立 IPSec SA。ISAKMP SA 和 IPSec SA 的区别在于前者是双向的。IKE 把动态协商过程定义成两个阶段的原因是为了提高 IKE 的协商效率。因为第一阶段协商的结果可以应用于多个第二阶段协商过程，而第二阶段协商过程可以同时进行多个，这样就能减少传输往返和幂运算，从而大大提高了协商的效率。对于协商过程的第一阶段，IKE 存在两种模式：主模式和积极模式。主模式是一种身份保护交换模式，而积极模式基于 ISAKMP 的野蛮交换法。在第二阶段，IKE 提供了快速交换模式，它的作用是为 IKE 之外的其它协议协商安全服务。对于参与密钥交换的双方，如果建立了 ISAKMP SA，那么不管谁是发起者，任何一方都可以主动发起第二阶段的交换。第一阶段中的主

模式提供了身份保护，当身份保护不必要时，可以使用积极模式进一步减少传输往返。在 IKE 整个协商过程中，ISAKMP 消息（或者称为 IKE 消息）被用来进行安全关联的协商交互。

第三章 基于 IPSec 的 VPN 网关总体方案

部队仓库驻地较分散，机关位于市区，各仓库则分散在偏远山区、农村，加之军队专用网络还没有铺设。各单位之间没有互通互连的内部网络，致使数据和信息交流的不通畅。因此，依托公网构建一个安全、便捷、低成本的专用网，使各单位之间数据互通、资源共享，是解决这一问题的有效途径。

本章针对师仓库部队网络安全需求，以 IP 安全协议体系 IPSec 为基础，面向网关到网关的 VPN 应用环境，针对 IPSec 协议的特点提出一个 IPSec 协议实现的思路，并对该思路的实现方法进行了研究，设计了一种基于 IPSec 协议的 VPN 网关。

3.1 VPN 网关设计需求

传统上，部队采用单独的组建网络方式，该网络与其它网络（包括部队内部其它网络、相关其它事业单位的网络、互连网）实行物理隔离，主要实现方式是直接租借地方包括光纤在内的各种物理线路或租界二层专线（如：DDN、X.25 等）组建自己的网络。从部队角度来看，物理线路和二层专线是透明的、是独占的，可以认为实现了物理上的隔离。但租用该专线的费用相当昂贵，而且又缺乏灵活性。本文采用 IPSec VPN 技术借助公共网络（如 Internet）实现仓库部队组网方案，不仅集灵活性、经济性以及扩展性与一身，而且在安全性方面可与专线具有相同的效果，可充分满足远程用户的需要。

随着师信息化管理的进展，以及人性化的要求，师仓库部队与师直属机关之间信息交互越来越多。师仓库部队的生活保障、仓库物质的远程维护等信息需要通过有效手段与师直属机关进行交互，以保障师仓库官兵的生活娱乐、文书传递、邮件收发以及仓库物质的远程维护等。这些信息为非战术性数据且信息量大、占用带宽多，需借助公网进行传递。

部队机关位于市区，而部队仓库分散在偏远山区、农村，军队专用网络还没有铺设。造成了仓库基层部队与直属机关之间的网络没有连接成为互通互连的内部网络，导致了数据和信息交流的不通畅。将仓库部队与师直属机关的局域网连接起来，为仓库保障系统搭建起一个统一完整的信息化管理平台。使仓库部队与师直属机关处于统一的信息化管理之下，仓库部队与师直属机关的用户享有相同的信息化服务。构建一个安全、高效、低成本的专网，形成部队各仓库部队与机关之间数据互通、资源共享，成为偏远仓库部队电子政务发展的当务之急。

本文通过运用 IPSec VPN 的技术，设计了适应于偏远仓库基层部队网络实现方案，改变目前偏远仓库基层部队基于固定物理地点的专线连接网络的状况。保证了

偏远仓库基层部队能够利用公共网络（Internet）传递私有数据来组建自己的安全可靠的网络。从而使部队机关及仓库基层部队构成了一个严密网络整体，保证了信息数据的安全传输。如图 3.1 所示。

两个安全网关分别保护位于后面的子网。在从安全网关流向 Internet 网络的数据都是经过安全网关加密和认证的，这样 Internet 上的用户非法获取这些数据将对源主机或网络不会造成威胁。同样，从 Internet 网络流入子网的数据也将先经过安全网关的处理，只有经过安全网关确认有效的数据包才能流入子网。两个子网之间传输的数据是经过两个安全网关协商后处理过的加密和认证的数据。

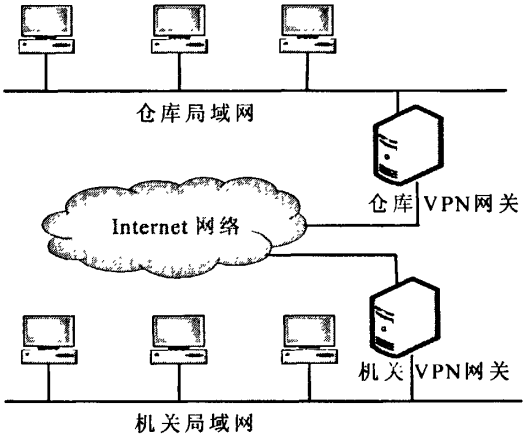

图 3.1　仓库虚拟专用网结构图

设计 VPN 网关，首先要考虑的就是网关的操作系统的选型问题。由于 IPSec 协议是在 IP 层实现的，需要对 IP 数据包进行操作。因此，操作系统最好能够支持 TCP/IP 协议栈源码公开。这样做的好处在于 IPSec 能够与 IP 层紧密集成在一起，更有利于诸如分片、PMTU 等操作的实施，另外，可以大大减少开发所需要的工作量。

基于上述说明，因此在设计 VPN 网关时，选择开源的 Linux 操作系统。选择 Linux 操作系统的另一个好处是：Linux 操作系统能够经过裁减掉不必要的服务和设备后装入比较小的硬盘或者闪盘，这样，VPN 网关的生产成本和使用成本上都会降低。

3.2 IPSec 协议实施方案

IPSec 协议是 IETF 安全工作组制定的一套可以用 IPv4 和 IPv6 上的，具有互操作性的，基于密码学的安全协议。它可以提供（无连接的）完整性、数据源头的认证、防重放功能、数据保密和一定的流量保密。IPSec 协议产生的最初衷是解决 Internet 上 IP 传输的安全性，它包括从 RFC2401 到 RFC2412 的一系列 RFC，它定义了一套默认的、强制实施的算法，以保证不同的实施方案可以互通。

IPSec 协议是目前基于密码学的安全协议中最完善、安全性最高、适应范围最广

的一套协议，可以为上层协议提供透明的安全保证，它既可以保护端系统到端系统的安全性，还可以保证网关到网关的安全性。

IPSec 包括两个协议：认证头协议（AH，协议号 51），和封装安全载荷协议（ESP，协议号 50）。

认证头协议：不包括加密过程，只提供完整性、数据源认证和防重放功能。

封装安全载荷协议（ESP）：提供数据保密性、完整性、抗重播、有限的传输流量的保密。

这两种协议的重要区别在于 AH 没有加密措施，ESP 有加密措施；同时 AH 的验证范围包括整个报文，而 ESP 不包括外部 IP 头。

根据实施的系统和保护的通信的不同，IPSec 协议定义了两种模式：隧道模式和传输模式，隧道模式用于隧道的终点和通信的终点不一致的时候，如主机和网关及网关和网关之间，用于保护主机到子网或子网之间的所有通信；传输模式保护的是隧道的终点和通信的终点一致的情况，用来保护端到端的安全。对端到端系统，IPSec 通过传输模式，对上层协议（TCP、UDP）数据进行保护。

IPSec 提供的具体服务内容是由系统的安全策略决定。策略是 IPSec 协议体系的重要组成部分，如果没有安全策略就不会有真正的安全性。同时安全策略的不匹配将造成通信的不可通。IPSec 协议体系中包括一个安全策略数据库（SPD），对安全策略应包括的一些属性进行了概念性的描述，但并没有规定安全策略的具体字段，如何表达等，在不同的实施方案中，安全策略往往是造成彼此不能通信的主要原因。

根据不同的安全策略，IPSec 规定的对网络层提供的处理包括三种：接受 IPSec 服务、丢弃 IP 数据报、或绕过 IPSec 服务。

IPSec 提供的安全性主要依靠于加密和验证算法，而且是基于对称密钥的算法，因此如何分发密钥成为 IPSec 协议的一个重要环节。IPSec 有人工和自动两种密钥分发办法。不过具体实施 IPSec 协议时，也可利用其它现成方法：如基于密钥分发中心（KDC）的系统或其它基于公开密钥机制的系统（如 SKIP）。

IPSec 可以在终端主机、网关/路由器、防火墙中进行实施和配置。根据不同的实施方式，IPSec 的实现方式也有很多种。

RFC2401 中规定了三种标准的 IPSec 实现方式：

1）与操作系统集成

主机和路由器均适合采用。特征是 IPSec 处理与 IP 层融为一体，同属内置功能。IPSec 完全嵌入到原有的 IP 层，将其作为网络层的一部分来实现，这需要涉及 IP 源码，IPSec 层需要 IP 层的服务来构建 IP 头，这个模型与其它网络层协议（如 ICMP）的实施等同。这类实现的优点是性能稳定、兼容性好。但关键问题是必须修改内核的源代码，鉴于 Linux 操作系统的源码开放性，这种实施方案最合适在基于 Linux

的系统平台上实施。

2）堆栈中的块“bump-in-the-stack”（BITS）

多在主机中采用。其主要的想法是在 IP 协议栈和本地的网络底层之间实现 IPSec，简单的说，就是在 IP 层和底层之间“嵌入”一层 IPSec 的实现，这种方法比较适用于在某台主机上实现 IPSec（注：必须指出的是，在这里“嵌入”这个词并不严格，因为虽然上层协议如 TCP 是直接和 IP 层打交道，它看不到 IPSec 层，但经由 IPSec 处理过的 IP 数据包，仍然以 IP 数据包的形式给 IP 的下层。）这类实现不需要访问 IP 层的源代码，而是采用软件封装/替换的思想截获所有进出 IP 层的数据流并施加 IPSec 处理。通用操作系统平台上的大多数防火墙产品都采用这种方式实现 IPSec，也有嵌入到路由器操作系统中的。这种方式存在功能的重复，从而造成复杂的局面。

3）线缆中的块“bump-in-the-wire”（BITW）

多在路由器中采用。特征是专门为 IPSec 构建一个独立的包含物理层、数据链路层和 IP 层的协议栈，采用专用软硬件实现。可以把这种实现看成是一种连在网络上的硬件或软件的黑盒子，未加密的数据从一个网络接口（trusted）流入，加密的数据从另一个网络接口（untrusted）流出，解密的过程则相反。这种实现方法既可以用于一台主机，也可以在一个局域网的网关处。前者和 BITS 相似，后者则充当安全网关或安全路由器的角色，可以有自己的 IP 地址，也常在商业性质的产品中和防火墙捆绑在一起，或充当一个 VPN 整体解决方案的一部分。这种方式是在一个设备中实现 IPSec，这个设备直接接入路由器的物理接口。缺点是实现技术难度大，成本高。

4）实施方案的比较

与操作系统集成效率最高，也最为有效。由于 IPSec 与网络层紧密集成在一起，因此它更有利于诸如分段、PMTU 之类的网络服务，这使得实施方案更为有效。另外，由于 IPSec 实现于操作系统内核，具有内核级别的调度优先级，因此效率是很高的。该方案的最大缺点是其实施必须依赖于操作系统源码的可得性。

BITS 方案的实施无需改动操作系统的源代码，无法获得系统源码时，这是一种比较理想的解决方案。这种方案的最大问题就是功能的重复。它要求实现大部分网络层特性，比如分段和路由表。功能的重复会使局面变得令人难以接受地复杂。很难解决像分段、PMTU 和路由之类的问题。

BITW 方案的实施较之 BITS 更为复杂，所以很少采用。

表 3.1　VPN 实现方案比较

方案比较	独立于操作系统	实现难易度	可扩展性	重复实现 IP 层功能	效率
与操作系统集成	否	易，难调试	差	不必	高
BITS	是	易	好	必须	低
BITW	是	难	好	不必	高

各种实施方式的比较分析如表 3.1 所示，经过分析本文采用与操作系统集成的方

式创建自己的 VPN 网关。Linux 是一个自由的源代码完全开放的操作系统，它不但提供了强大高效的多任务调度功能，而且用户完全可以按照自己的意愿改造定制适合自己本系统应用的内核，可以保证内核的小巧、健壮和高效率运作。从这一点看来，Linux 作为本系统所采用的与操作系统集成的方案实施 IPSec 协议组件所需的源代码开放的操作系统是再合适不过了。通过对 IPSec 协议的分析，和对 Linux 内核的理解掌握，对其网络部分进行了相应的改造，将 IPSec 协议紧密整合到 Linux 内核网络部分。

修改 Linux 内核将 IPSec 与 IP 层融合成为 IP+IPSec 层。修改内核的标准 IP 处理程序，将 IPSec 处理程序与内核 IP 处理程序捆绑，形成新的、具有 IPSec 处理功能的 IP 层处理。这种方案处理效率高，将 IPSec 处理与 IP 处理融合到一起，提高了 IPSec 网关的处理效率，极大地提高了 IP 包在 IPSec 网关上的传输速度，使 IPSec 网关成为瓶颈的可能大大降低。该方案的不足，也是难点即实现较为困难，而操作系统本身对其也有较大的限制，内核的修改涉及到整个系统的稳定性。

在 Linux 内核 2.4 版本之前，内核采用 ipchains 实现包过滤和防火墙功能，在这种模式下通常采用通过注册 Linux 的 netdevice 设备进行入口操作。自 2.4.0 版本之后，Linux 内核采用了 Netfilter 网络底层开发结构，网络新特性的扩展可以十分简单的通过插入新模块来实现。考虑即要获得高效的功能实现，又要避免大量的修改内核 IP 处理程序，采用 Linux 最新的 Netfilter 网络架构实现 IPSec 处理入口函数，与内核模块进行无缝链接，这样其代码实现方便，结构清晰，运行高效且系统稳定性好。经过改造的 Linux 安全网关能够对所有经过它的 IP 包进行 IPSec 处理。

3.3 网关的总体框架设计

3.3.1 IPSec 的实现思路

在对 IPSec 进行深入的研究和分析后，发现复杂性是 IPSec 最突出的缺陷之一。安全性最大的敌人是复杂性。复杂系统比简单系统有更多的缺陷，并且是复杂的缺陷。纠正这些缺陷是困难的，并且在你了解这些系统之前已变得不可管理。美国卡内基·梅隆大学的一项研究发现，编程人员编写的程序平均每增加 1000 行代码就会增加 100 至 150 个错误。在开发项目中使用程序体系上的最小原理（PTL），因为一个系统越复杂，系统的漏洞就会越多，系统的建立和维护，特别是安全性维护工作就会变得更复杂。

IPSec 的复杂性主要指包含的太多的可选项和灵活性，常常有好几种方法做相同或相似的事情，这是没有必要的。系统在面临多种选择时必然要花费很多的系统时间与资源。分析在网关上实现 VPN 时提到，除了考虑安全性这个重要因素外，由于

网关的特殊地位，高效性也是一个重要考虑因素。因此在设计具体的基于 IPSec 的 VPN 时，针对具体的应用环境，对 IPSec 的复杂性进行改进，并将这种理论上的建议进行实践，设计一种高效同时又不削弱系统安全性的 VPN 模型。

由第二章知道，IPSec 有两种封装模式：隧道模式和传输模式。有两种协议可选用：AH 协议和 ESP 协议。这样就有四种不同的组合：隧道/AH、隧道/ESP、传输/AH、传输/ESP。同时在 ESP 中加密和认证都是可选的，因此单就认证方面来说，上面四个组合在功能和性能上的差别都很小。首先实现是在网关上，由于在网关上实现隧道时，隧道口的地址和真正通信的主机地址往往是不同的，因此需要添加一个外部 IP 头，外部 IP 头中的地址为网关的 IP 地址。基于上述原因，系统必须采用隧道模式，这在一定程度上减少了复杂性。这样上面四种组合在系统中只剩下了两种可选，即隧道/AH、隧道/ESP。

从前面的 2.5 节 AH 和 ESP 的比较中发现，在隧道模式下，在认证方面 ESP 与 AH 认证的唯一区别就是对外部 IP 头有无认证，AH 对其进行了认证，而 ESP 没有。但是经过分析发现，实现 VPN 时最重要的是保护 VPN 网内部主机的通信数据。外部 IP 头的设立主要是为了穿过公网时能正确地被转发到隧道口。对外部 IP 头的修改无非产生两个结果：

1）数据不能到达隧道口。产生这个结果的原因有很多，如攻击者修改了目的地址、减小了 TTL 字段使数据还没到达就失去生命期等。这对于 VPN 系统不会造成什么伤害，系统可以超时重发，而攻击者得到的数据因为不能解密而无用。

2）数据可以到达隧道口。有可能攻击者修改了源地址等不影响正常转发的数据字段。这样的数据不能通过主机网络层和互联层的本身的认证，否则外部头 IP 的作用已经完成就会被剥离。不影响 IPSec 模块对其正确的处理。对于外部 IP 头中协议字段的修改问题，无非是将本该交给 IPSec 处理的数据交给了互联层，但互联层检测出错误后会将其简单丢弃，数据就会超时重发。

所以针对外部 IP 的攻击是没有什么价值的，也是很少的。那么看来隧道/AH、隧道/ESP 两种组合在认证方面保护力度就是一样的了。而且隧道/AH 不提供保密服务。所以在系统的实现中只采用隧道/ESP 这一种方式。这样在消除传输模式的两种组合方式基础上又消除了隧道/AH 方式，再一次减少了系统的复杂性。但在实现中应当注意，没有了 AH 协议以后，ESP 的认证不再是可选的，而只有加密是可选的，这样做同时也避免了系统管理员的误配置。

综上所述，只选用隧道模式的 ESP 协议，同时 ESP 的加密和认证不是可选的，而是必须实施的。

3.3.2 网关系统实现结构

选择在网关上实施安全措施，这些安全措施由 IPSec 的相关技术所提供的。因此重点是设计和实施 IPSec，实现一个安全、高效的 VPN 网关，保障 VPN 内部通信的安全。在对 IPSec 的相关技术和协议进行了深入的研究以后提出了下面这个网关系统结构，如图 3.2 所示：

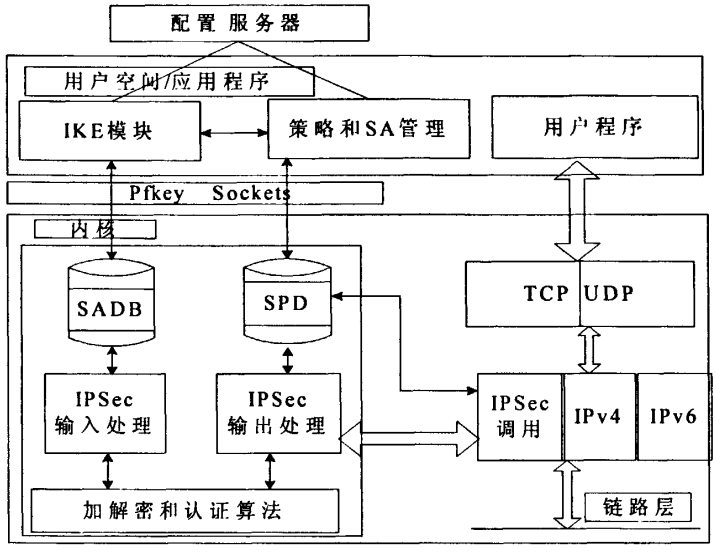

图 3.2 VPN 安全网关系统结构

系统拥有两种工作层：用户层和核心层。一般应用程序工作在用户层，而内核模块和最基本的操作系统核心都工作在核心层。在用户层下的进程只能访问自己的进程空间中的数据结构，而在核心层下执行的进程还可以访问各种系统核心数据[33]。由于 IPSec 的处理需要访问系统的协议堆栈，因此，本系统需要在用户层和核心层两个工作层下操作。

在实现安全网关时，将 IPSec 分成图 3.2 中几个模块实现：IKE 模块、SADB 和 SPD 数据库、IPSec 处理模块、策略和 SA 管理模块、加密和认证算法模块组成，其主要功能见表 3.2。

表 3.2 模块功能表

模块名	主要功能
IKE 模块	主要负责动态的协商和管理 SADB。
SAD 模块	存放安全联盟 SADB，决定了具体采取什么处理。
SPD 模块	用来存放完整的安全策略，并决定了数据包是否要采用 IPSec 处理。
IPSec 处理模块	主要实现 ESP 的功能，负责按照查找到的安全策略对数据包进行处理。
策略和 SA 管理模块	主要是在手工方式下对 SPD，SADB 进行添加、删除、排序等数据库管理操作。
加密和验证算法模块	用来存放系统需要的加密和验证算法，以供 IPSec 处理时使用，加密算法包含 DES、3DES 等常用加密算法，验证算法包含 HAMC-SHAI[34]、HAMC-MD5[35] 等常用验证算法。

各模块功能的实现见第四章。

3.3.3 应用层组件设计

VPN 网关的应用层组件由于处于操作系统上层，都是应用程序，因此其设计和实现相对于内核层组件来说比较简单。但是，由于应用层组件表现为最终系统的用户视图，所以，在视觉效果和功能性的设计方面比较重要。关于视觉效果和功能性设计的叙述，将不会在本文中论述。

应用层组件包括策略数据库配置接口和 ISAKMP/IKE 协议。对策略数据库配置接口而言，主要是能够实现对策略数据库项目的增加、删除、修改等操作。对 ISAKMP/IKE 协议而言，主要是能够实现 IKE 通讯，经过密钥协商，最后产生 IPSec SA。

1）策略数据库概述

策略是 IPSec 协议族中一个比较重要的组件，IETF 有专门的工作组负责制定关于策略的标准和草案，目前已经制定的标准有两个，并且还有其他相关标准正在形成中。

策略是一个非常重要的问题，因为它决定了为一个数据包提供什么样的安全服务，以及两个实体之间是否能够通信。

IP 数据包的外出和进入处理都要以安全策略为准。在进入和外出包处理过程中，需查阅 SPDB，以判断为这个包提供的安全服务有哪些。为了提供对非对称策略的支持（亦即在两个主机之间，分别为进入和外出的数据包提供不同的安全服务），可为进入与外出的数据包分别维持不同的 SPDB。然而，密钥管理协议总是协商的双向 SA。在实际应用中，通道和嵌套处理大多数都是对称的。

安全策略要求策略管理应用能够增添、删除和修改策略。由于 SPDB 保存在内核，所以对一个具体的 IPSec 实施方案而言，应提供一个恰当的接口，以便对 SPDB 进行操纵。至于 SPDB 的具体管理方式，则要由实施方案来决定，而且并未为此专门定义一套统一的标准。

由于 SPDB 保存在操作系统内核层，所以，需要在操作系统应用层对 SPDB 进行相关操作如增加、删除和修改等。也就是说，在应用层实现的 SPDB 是一个面向用户的可视的策略数据库的管理接口。

2）策略数据库管理接口

策略数据库管理接口主要是为了用户能够方便快捷的操作安全策略数据库。在 VPN 网关策略数据库管理接口的设计中，将策略数据库中的策略看作是一条条策略条目组成的集合，而每一条策略条目的内容包括本地 VPN 网关地址、本地被保护网络地址、对方 VPN 网关地址、对方被保护网络地址以及其他相关信息。

为了对 VPN 网关策略数据库进行很好的配置操作，结合 Linux 和 Linux 下配置文件格式，通过设计并使用 XML 格式的配置文件对 VPN 网关策略数据库进行配置。

使用 XML 格式配置文件的好处在于：

- XML 配置文件可以跨平台，有很好的移植性。
- XML 可以设计简单有效的程序，大大提高对配置文件的操作效率。

3.3.4 内核层组件设计

内核层组件运行在 Linux 操作系统的内核中，与操作系统各组件共享同一个上下文空间。内核层组件的运行必须依赖于内核中的其他组件的配合，由于内核层没有进程空间等概念，所以内核层组件的设计和实现必须足够小心，否则容易导致整个操作系统内核的崩溃。

Linux 操作系统既不是一个纯粹的微内核操作系统也不是一个单一的（monolithic）操作系统，Linux 同时具有了单一操作系统和微内核操作系统的优势，是两种操作系统某种意义上的折中。

单一操作系统是目前大多数操作系统所使用的模式。所有的操作系统相关组件都是在一个统一的内核环境中运行。而微内核操作系统一直是操作系统理论界比较推崇的一种操作系统模式。微内核操作系统由各个功能模块组成，如文件系统模块、进程模块、设备驱动模块等。各个模块之间相对独立而又互相依赖。这样的好处是便于移植，管理简单，并且有更好的内存利用效率。缺点就是各个功能模块之间要传递消息，性能上相对于单一操作系统会稍有降低。

Linux 通过使用“模块”的概念来实现微内核的设计理念。但是各个模块又是在单一操作系统环境中运行。Linux 内核在需要用到某个模块的时候会将该模块调入到操作系统中，在不使用某个模块的时候将其调出。

在 VPN 网关内核层组件的设计中，可以遵循 Linux 操作系统的内核设计理念，考虑将各个组件设计成模块，在需要的时候调入，不需要的时候调出。

VPN 网关内核层组件包括了安全关联数据库、安全策略数据库和 IPSec 协议实现。其中，安全策略数据库和安全关联数据库在第二章中已经有详细的叙述，在此不再赘述。

安全策略数据库和安全关联数据库的设计遵循相应的数据结构，以链表的形式进行组织。并在此链表上定义相应的操作如增加、修改和删除等。而 IPSec 协议在内核中的实现是 VPN 网关的至关重要的部分。

1）内核层 IPSec 协议需求

经过对 IPSec 协议族的分析，基于安全性的考虑，在 IPSec 协议族的内核实现中，只对 ESP 协议的隧道模式进行设计和实现。

- ESP 能够提供高强度的机密性并且能够提供完整性校验。
- ESP 协议隧道模式能够穿越 NAT（Network Address Translation）。

2）内核层 IPSec 协议设计

由于 IPSec 协议中的 ESP 协议需要对原始的 IP 数据报进行改造，因此，内核层 IPSec 协议的设计需要与 TCP/IP 协议找紧密结合，这也就是要采用 Linux 操作系统作为 VPN 网关操作系统的最主要的原因。

Linux 操作系统是开放源代码的操作系统。根据 Linux 操作系统的特点，在 Linux 中设计并实现 IPSec 协议，需要将 IPSec 协议模块设计成为设备驱动的方式。以设备驱动的方式被 Linux 操作系统加载在操作系统内核中，在数据报发送和接收的过程中对符合条件的 IP 数据报进行处理，完成 IPSec 数据报的封装和拆装的工作。

在内核层 IPSec 协议设计中，把 IPSec 协议模块设计成一个网络驱动程序，能够异步的接收和发送数据报。

3.4 IPSec 的进出处理

传统实现方案采用插入 IPSec 处理模块方式，本文采用的方案是 IPSec 融人 IP 层。修改内核的标准 IP 处理程序，将 IPSec 处理程序与内核 IP 处理程序捆绑，形成新的、具有 IPSec 处理功能的 IP 层处理。

3.4.1 进入数据包处理

1）传统实现方案对于接收包处理

当数据从网卡到达链路层数据时，首先交给 IPSec 层，IPSec 层要求实现大部分的网络层的特性，如分段和路由表。此时调用 IPSec 进入处理模块。经过 IPSec 进入处理后，重新组装 IP 包，发到内核的标准 IP 处理入口。内核的标准 IP 层处理程序进行 IP 分片、重组等再一次 IP 处理后，IP 层把数据包传给传输层。而对于外地包，则需要转发。转发时对于大数据包需要进行分片处理。在发送到网卡前，调用 IPSec 外出处理模块。经过 IPSec 外出处理后，重组外出包，并重新路由、分片处理，然后发送到物理接口。

2）本文采用的方案对于接收包处理

当来自外网的数据包到达本地安全网关时，安全网关进行接收包的 IPsec 处理。处理过程中，进来的数据包在 IP 层首先进行必要的重组（Reassemble），然后流入 IPSec 处理模块，检查 IP 数据包标记，判断该 IP 包是否是一个 IPSec 包。若包内不含 IPSec 头，由其它层进行相应的策略检查和处理。如果是，则将进行 IPSec 进入处理。

IPSec 进入处理首先将从 IP 包中提取目的地址、下一协议、安全参数索引（SPI）信息，并根据这些信息查找安全关联库（SAD），找到对应的唯一的 SA。然后根据找到的 SA 进行 SA 状态处理、SA 生存期处理、重播窗口处理、模式处理、相关 IPSec

协议处理（改进的 ESP）。当处理完一个 SA 后，必须要标记处理过的 SA（用于后期的进入策略匹配），并检查是否存在 SA 串（即多个 SA）。如果有，则循环处理 SA 串，直到遇到传输层协议头或者非本机的 IP 头（表明 SA 串上的所有 SA 都处理完毕）。

最后根据处理过的 IP 包的相关信息查找安全策略库（SPD），找到该 IP 包所对应的进入策略。然后检查该策略所指定的 SA 串是否与处理过并标记的 SA 串相同，从而判断是否进行了所要求的 IPSec 处理。如果不相同，则丢弃该 IP 包。其过程流程图如图 3.3 所示。

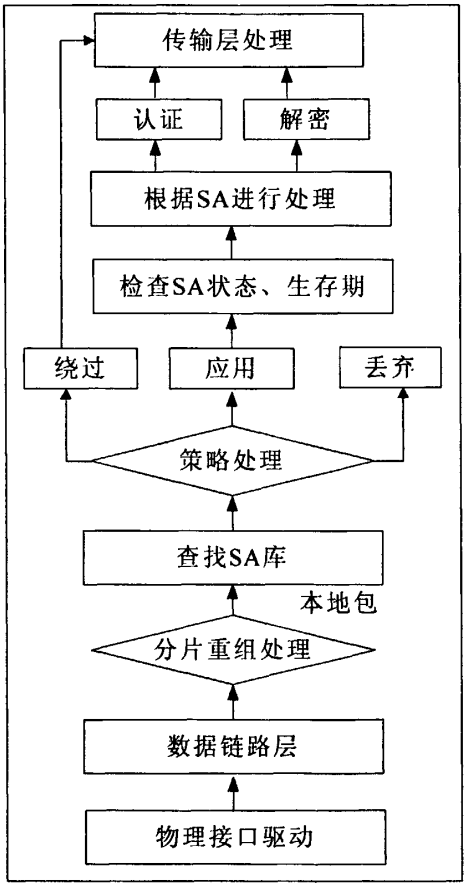

图 3.3　进入数据包处理流程图

3.4.2 外出数据包处理

1）传统实现方案对于外出数据包处理

对于发送包处理，当传输层数据到达时，首先交给内核的标准 IP 处理程序。在进行完 IP 的分片等 IP 层外出包处理及路由后，准备发送到链路层。此时调用 IPSec

外出处理模块。经过 IPSec 外出处理后，重新发送到内核的标准 IP 层外出处理入口。然后重新进行 IP 分片及其它 IP 层外出处理，并重新路由。最后将包发送到链路层，进入网卡。

2）本文采用的方案对于外出数据包处理

对于传输层来的包，首先根据选择符（如协议类型、源地址、目标地址）检索安全策略数据库（SPD），判断是否为这个包提供安全服务，是否需要进行 IPSec 处理。如果检索的结果为丢弃，则中断对该包的发送操作。如果是不应用安全服务，则将该包直接发送到 IP 层。如果需要应用安全服务，则要求将 IPsec 操作应用于这个数据包。在这种情况下，首先，检查安全联盟数据库（SADB），找到对应的 SA。如果没有对应的 SA，则利用 IKE 协议，与远端对等实体进行协商，建立一个新的 SA。为安全起见，在 SA 没有建立之前，IP 包不会被传送出去，IPsec 一直处于等待状态。有了 SA 之后，IPsec 根据 SA 为 IP 包添加合适的 ESP 头，进行相应的安全处理。最后，对重新构造的包进行分片等其它 IP 外出处理和路由，然后将包发送到链路层处理。如图 3.4 所示。

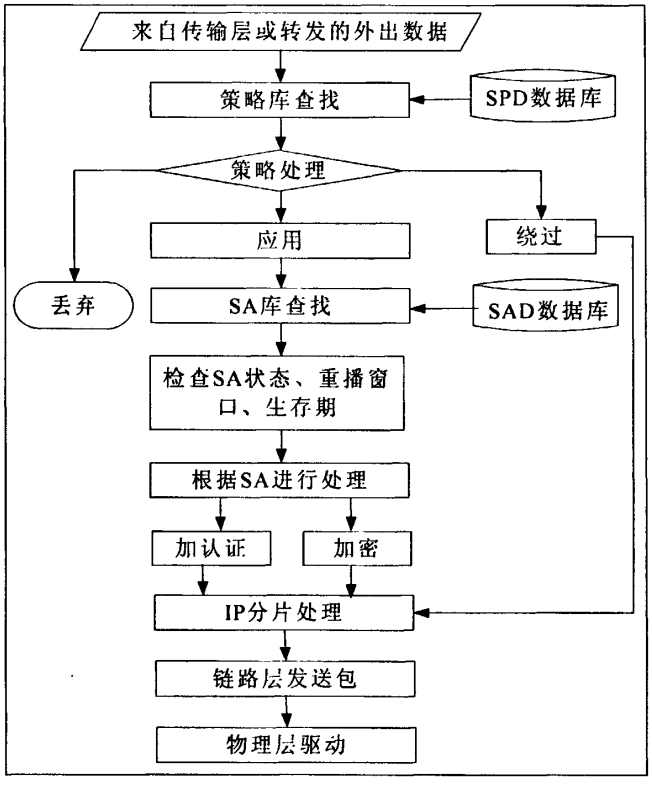

图 3.4　外出数据包处理流程图

3.4.3 优缺点

1）传统实现方案对于数据包处理

该方案最大的优点在于 IPSec 处理模块独立于内核，IPSec 处理程序几乎对内核不进行任何修改。然而缺点也是显而易见的，该方案多次重复地处理 IP 包，大大地增加了 IP 包在 IP 层的处理时间，极大地影响了 IP 包处理的效率。这对于业务繁忙的网关，将影响正常的通讯，成为通讯的瓶颈。严重时将造成大量的丢包，使业务受阻。

2）本文采用的方案对于数据包处理

该方案最大的优点是处理效率高。该方案将 IPSec 处理与 IP 处理融合到一起，提高了 IPSec 网关的处理效率，极大地提高了 IP 包在 IPSec 网关上的传送速度，使 IPSec 网关成为瓶颈的可能大大降低。该方案的不足是实现较为困难。因为要修改内核代码，因此要熟知内核的网络模块代码，甚至要全面的了解操作系统的代码，这样将大大增加实现的难度。利用 Linux 的开放源码机制优势，选用 Linux 操作系统，通过修改操作系统内核中的 TCP/IP 协议源码实现。将 IPSec 集成到 IP 源码中可以直接利用 IP 模块的某些功能（如分片和重组、路由选择等），避免了相同功能的重复构造。

第四章　网关的详细设计与实现

方案中各功能模块：IKE 模块、SAB 模块和 SPD 模块、IPSec 处理模块、策略和 SA 管理模块是 VPN 重要组成部分，是实现网关的坚实的基础，其设计的详细直接影响到网关的安全与性能。

本章介绍了网关的上述关键模块的实现，由于加密和认证算法模块采用的是现有的算法，如加密算法 DES，3DES 等，验证算法 HAMC-SHAI、HAMC-MD5 等，这里不再涉及。给出了网关实现所需用到的关键技术和具体函数，并给出了一个在安全策略已知情况下，利用 VPN 网关实现的安全隧道的实例。

4.1 IKE 模块的实现

4.1.1 IKE 模块框架结构

IKE 是作为一个守护进程运行的，负责处理用户的管理配置命令、同协商实体的交互、IKE 数据报的处理以及同内核的 SAD 的交互。为了各个模块能对协商的数据进行共享，设计了 IKE 状态库模块，能够提供统一的接口实现查询，更新、删除、添加等操作。在 IKE 的实现过程中，IPSec 通信的过程中 IKE 协议协商的整个处理流程如图 4.1 所示。通信双方通过 IKE 协商，建立起安全关联索引，并在安全关联索引的保护下，建立使用 IPSec 协议保护的通信服务。

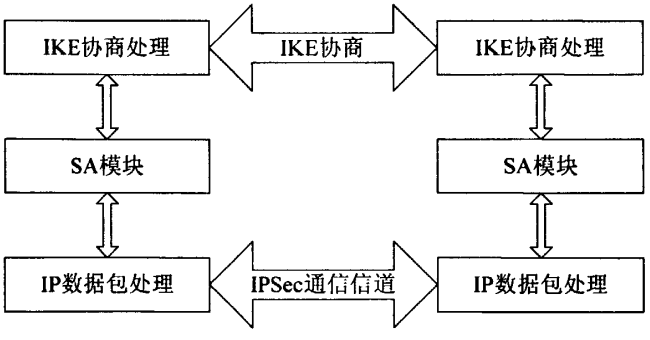

图 4.1　IKE 协议协商的整个处理流程

为完成 IKE 的协商过程，本文设计了 IKE 的实现框架，如图 4.2 所示。

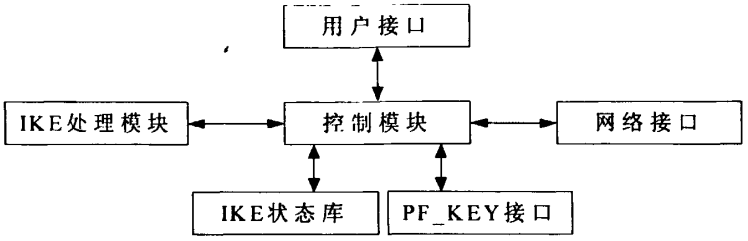

图 4.2 IKE 实现框架图

IKE 存在于用户空间，是作为一个守护进程运行的，负责处理用户的管理配置命令、同协商实体的交互、IKE 数据包的处理以及同内核的 SADB 的交互。整个系统按照功能划分成三个模块：IKE 控制模块、IKE 处理模块以及 IKE 的状态库，并且提供了三个接口，即网络接口、用户接口和 PF_key 接口。IKE 控制模块是一个守护进程，负责整个 IKE 的控制管理等。IKE 处理模块的功能是处理 IKE 的协商数据包，它负责验证 IKE 协议的载荷数据，并构造响应或请求数据包内容。为了各个模块能对协商的数据进行共享，使用 IKE 状态库模块来进行管理。IKE 状态库记录 IKE 运行期间需要的协商信息和当前的 SA 信息，能够提供统一的接口实现查询，更新、删除、添加等操作。

IKE 作为一个应用层协议在应用层中实现，由于需要同内核 SADB 进行 SA 消息的传递，所以要提供了一个接口，使用 PF_KEY 接口作为内核和 IKE 守护进程的接口。

为了方便用户的管理、配置、监视，系统还提供了用户管理接口。在通信双方进行协商过程中，需要使用 UDP 协议的 500 端口为 IKE 守护进程提供网络通信的网络接口。

4.1.2 控制模块与处理模块

1）控制模块负责整个 IKE 协商过程中的控制和管理，并包括 IKE 系统中三个主要的接口，在系统运行期间，这些接口有许多事件和消息要进行处理，其中包括用户接口可能要发送管理消息，内核接口发出的 SA 请求、更新消息，以及网络接口传来的协商消息等。所以控制模块设计成服务器方式，负责监听这些消息和事件。控制模块使用三个消息队列，对每种消息类型分别设置一条消息队列，模块轮询这几条消息队列，如果有消息就调用对应的消息处理函数来处理，如图 4.3 所示。其中 pfkey_event 函数处理内核接口的 SA 协商请求或其他内核事件；comm_handle 函数处理网络接口上的 IKE 协商消息；whack_handle 函数处理用户接口发来的请求。此外，控制模块还要负责时间事件等的处理过程。

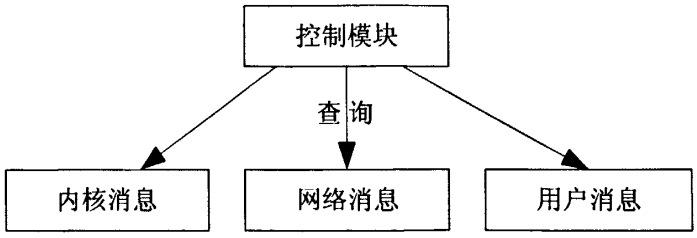

图 4.3 IKE 控制模块

内核接口模块监听到有消息到达时，取出消息。根据消息类型调用对应的消息分析处理函数，检查它的完整性和合法性。然后根据 SA 类型调用相应响应函数，可以向内核的 SADB 增加一个 SA，删除一个 SA，或读取一个 SA。

根据 RFC2367 文档，通过 PF_KEY 协议来作为内核和 IKE 守护进程的接口。PF_KEY 是一个 socket 协议簇，用于可信任的密钥管理程序和操作系统内核内部的密钥管理进行通信及服务器程序和客户程序之间的通信。PF_KEY 协议簇 socket 的建立以及读写等操作同其他类型的 socket 操作类似，如下是面向连接的套接字的使用流程。

- 使用函数 socket (int family, int type, int protocol)来定义套接口。
  sockfd = socket (PF_KEY, SOCK_RAW, PF_KEY_v2)。
- 用函数 bind(int sockfd, const struct sockaddr* servaddr, socklen_taddrlen)将服务器地址和生成的套接字捆绑在一起。即
  bind(sockfd, (struct sockaddr*) & serv_addr, sizeof(serv_addr))。
- 服务器调用函数 listen(int sockfd, int backlog)来进行监听。
- 服务器用函数 accept（int sockfd, const struct sockaddr*cli_addr, socklen_taddrlen）等待连接请求。
- 客户端调用函数 connect（int sockfd, const struct sockaddr*sevaddr, socklen_taddrlen）来向服务器请求建立一个连接。

在 IKE 启动时调用初始化函数 init_kernel()初始化 pf_key 接口，创建一个 PF_KEY 的套接字 pfkeyfd，IKE 守护进程通过使用这个 socket 接口来与内核通信，发送和接收信息。使用 pfkey_iq_head 的静态链表存放内核请求消息，其节点的主要组成部分的结构如下：

```
struct sadb msg{
  uint8_t sadb_msg_versicn;    /*PF_KEY 消息的版本号*/
  uint8_t sadb_msg_type;       /*标识消息的类型*/
  uint8_t sadb_msg_errno;      /*返回错误号*/
  uint8_t sadb_msg_satype;     /*标识安全关联的类型*/
```

```
  uint16_t sadb_msg_len;        /*消息的整个长度*/
  uint16_t sadb_msg_reserved;   /*保留值*/
  uint32_t sadb_msg_seq;        /*消息的序列号*/
  uint32_t sadb_msg_pid;        /*进程号去*/
}
```

网络服务接口子模块的功能是把本地的 IKE 载荷数据封装成 UDP 数据报，把接收到的 UDP/500 数据包剥离出 IKE 载荷。系统启动的时候，启动一个网络服务的子进程。网络消息处理模块既充当服务器又充当客户端角色。当监听 UDP 的 500 端口时，它是服务器，作为协商的响应者。当它发出协商请求时，它作为协商的发起者充当客户端。

网络消息处理模块分析并提取数据，组织成 msg_digest 结构数据，然后把此数据作为参数调用 IKE 验证模块进行处理。IKE 验证模块返回后处理标志，把构建的响应载荷或失败代码添加到 msg_digest 结构中。当消息验证成功的时候，将在 IKE 的状态库中添加新的连接。结构体 msg_digest 如下：

```
Struct msg_digest{
  chunk_t raw_packet;           /*原始数据包*/
  const struct iface *iface;    /*消息到达的接口*/
  ip_address sender;            /*发送者 ip 地址*/
  u_intl6_t sender_port;        /*发送者端 M 号*/
  pb_stream packet_pbs, message_pbs; /*IKE 载荷*/
  struct isakmp_hdr hdr;        /*消息头*/
  hool encrypted;               /*加密标志*/
  enum state_kind from_state;   /*包对应的协商对象的状态*/
  struct state *st;             /*当前状态对象*/
  pb_stream reply, rbody;       /*响应报文载荷*/
  notification_t note;          /*验证失败原因*/
  #define PAYLIMIT 20
  struct payload_digest digest[PAYLIMIT], digest_roof, chain[ISAKMP_
NEXT_ROOF];
}
```

2）IKE 处理模块负责验证 IKE 协议的载荷数据，并构造响应或请求数据包内容。在 IKE 协议的各种模式下，每种协商过程都有其固定的消息条数且每条消息的内容都有明确的规定。因此可以按照协商过程中接收到的消息来确定协商状态，即当前所处的阶段，例如主模式的发起者的第一个状态定义为 STATE_MAIN_Il。对每种状

态定义对应的状态处理函数，把所有的状态处理函数的入口地址统一放在状态处理函数表中，并使用索引表来进行查找。静态索引表 ike_microcode_index 中的元素 state_microcode 的结构如下：

```
struct state_microcode{
  enumstate_kind state;              /*当前状态，是处理函数表的入口项*/
  lset_tflags;                       /*标志，指示此函数能处理哪种消息*/
  lset_t req_payloads;               /*加消息中必需的载荷类型*/
  lset_topt_payloads;                /*消息中可选的载荷类型*/
  u_int8_tfirst_out_payload;         /*构建相应消息中的第一个载荷类型*/
  enumevent_typetimeout_event;       /*函数要注册的超时事件*/
  state_transition_fn *processor;    /*状态处理函数地址指针*/
}
```

这样当需要对某条消息进行处理时，以此时协商的状态做索引找到的状态函数的入口地址，然后根据认证方式找到相应的状态处理函数进行处理。在协商过程中接收一条消息时，控制模块的网络子模块的处理过程 comm_handle 就调用此时状态所对应的处理函数进行分析处理，验证完成后当前的状态就变为下一个状态。在处理函数中按照协议规定对载荷数据进行验证，验证通过要构建响应载荷。如果验证失败，根据安全性级别决定是否构建通知载荷发送给协商的另一个实体。状态处理函数中如果 SA 已经建立起来，则要通过 pf_key 接口向内核安装该 SA。

4.1.3 IKE 状态库

IKE 在开始协商时需要知道一些必要的协商策略、密钥信息、协商网关的身份等数据。在协商期间还需要记录协商对象的状态，协商对象要存放协商 SA 的所有协商信息，而且 IKE 协商时也要查询 SA 的状态。为了方便这些数据统一组织管理，提供统一接口便于各模块共享，设计了 IKE 的状态库。

实现的数据结构是通过指针将数据组织成链表形式。查找、更新、删除、添加在对应的链表上操作。系统共有两条全局变量链表，分别是 connections 和 states。

1）connections：记录的是一条协商通道的信息，它的集合构成 IKE 的策略库。

- 名字：标识一条 connections。
- 策略：策略中既包含 ISAKMP 认证策略，又包括 IPSec 验证加密策略以及无 SA 时的策略，以掩码位指示。
- 时间参数：包括 ISAKMP SA 存活期，IPSec SA 存活期，更新密钥时间间隔 rekeymargin，随机更新参数 refuzz，更新密钥重试次数。当 SA 的剩余时间小于 rekeymargin*（1-rnd%）时进行更新，其中 rnd 是 0 到 refuzz 内的数。防止协商双方

同时进行密钥更新产生冗余的 SA 协商。

- 协商两网关主机的信息：包括身份（IPaddr，FQDN，USER@FQDN），主机地址，下一个路由地址，子网地址，端口号等等，接口设备，协商的 SA 的序列号。
- 连接类型：CK_TEMPLATE，通配的连接即 peer 的地址为 anyaddr。用于监听移动用户和内核的协商请求：CK_PERMANENT，一般连接；CK_INSTANCE，CK_TEMPLATE 连接的实例化；CK_GOING_AWAY，正在被删除的连接。

2）states：记录协商的 SA 的状态，属性值，它的集合可以看作是 IKE 的 SA 库。

- 序列号：依附的 connections；
- 加密/验证信息：包括算法类型，加密器/验证器，oakley 组，AH/ESP/IPCOMP 信息，包括具体的协议属性，spi，密钥；
- oakley 组：DOI，Situation，策略，消息序列号，已使用的消息序列号，DH 值，各种密钥值，状态，对应的超时事件。

所有的 connections 结构被连接到 connections 链上，查找、更新、删除、添加在此链上操作。

4.2 SAD 模块的实现

IPSec 对通信数据的保护依赖于用来保护数据包安全的 IPSec 协议（AH 或 ESP）、模式、算法及密钥、生存期、抗重播窗口、计数器等安全参数的选取。安全关联 SA 决定了以上的各项安全参数，是通信实体间通过协商而建立起来的一种协定，是 IPSec 实施的基础。同时，SA 的产生也依赖于安全策略的正确制定。

SAD 维护了 IPSec 协议用来保障数据包安全的 SA 记录。每个 SA 都在 SAD 中有一条记录相对应。对于外出处理，在 SPD 中查找指向 SAD 中 SA 的指针，如 SA 未建立，则应激活 IKE 建立 SA，并同 SPD 和 SAD 的记录关联起来。对于进入处理，SAD 的记录用目的 IP 地址、IPSec 协议类型和 SPI 标识。SAD 实现的具体方式是使用哈希表，使用<SPI，目的地址，安全协议>三元组（SAID）作为关键码，借助哈希表的链接方式来解决冲突。同时使用链接方式组成 SA 束，以描述复杂的报文处理过程。查找时，通过对 SAID 的散列找到 SA 头，然后再进行详细匹配找到相应的 SA。

当对一数据包进行处理时，首先根据 SPD 中相对应表项，取得 SAID。然后，通过 SAID 在 SPD 中查找得到同组 SA 链的链首，依次按该 SA 链中的 SA 对数据包进行加密、认证和封装等处理。

4.2.1 SA 的操作

1）创建一个 SA

根据用户提供的 SA 相关参数构建 SA 结构，然后提取 SAID 值，并对 SAID 进行散列，将 SA 结构放入散列链头。

2）删除一个 SA

根据用户参数，提取 SAID。根据 SAID 查找 SAD，找到后将 SA 结构从链中删除。

3）查找一个 SA

根据用户参数，提取 SAID。对 SAID 散列后，在 SAD 散列表中中找到 SA 链头，再进行详细 SAID 匹配找到唯一的 SA。

4.2.2 SAD 模块中的关键数据结构

SA 的信息存放在 SAD 数据结构里，其数据结构描述如下：

```
struct sa{
  atomic_t   ips_usecount;        /*计数信号量*/
  struct   sa * ips_hnext;        /*指向 sa 链中下一个 sa*/
  struct   sa* ips_onext;         /*双向链表中，指向前一个 SA 的指针*/
  struct   sa* ips_inext;         /*双向链表中，指向下一个 SA 的指针*/
  struct   sa_id ips_ said;       /*SAID*/
  struct   ifnet ips_rcvif;       /*相关 rcv encap 接口*/
  _u8    ips_ authalg;            /*此 SA 中的认证算法*/
  _u8    ips_encalg;              /*此 SA 中的加密算法*/
  _u32   ips_seq;                 /*协商了该 SA 的那条消息的序列号*/
  _u32   ips _pid;                /*协商了该 SA 的那个进程的进程号*/
  struct   stats   ips_ errs;
  _u32   ips _replaywin_lastseq;  /*最后一个报的序列号值*/
  _u64   ips _replaywin_hitmap;   /*接收到的 pkts 位图*/
  _u32   ips _replaywin_maxdiff;  /*最大 pkt 序列号差异值*/
  _u32   ips _f lags;             /*生成旗标*/
  _u8    ips_ replaywin;          /*重发窗口大小*/
  _u8    ips_ state;              /*SA 的状态*/
  struct   lifetimes   ips_life;  /*生存时间记录*/
  caddr_t   ips_key_a;            /*认证密钥*/
  caddr_t   ips_ key_e;           /*加密密钥*/
  caddr_t   ips_iv;               /*初始化向量*/
  struct   ident ips_ident_s;     /*识别源地址*/
```

```
    struct    ident ips_ident_d;          /*识别目的地址*/
}
```

4.3 SPD 模块的实现

将安全策略库的配置设置在安全网关上，各个安全网关独立配置策略库，但要保证策略的一致性。安全策略将决定应用于数据包的安全服务以及如何对数据包进行安全处理。因此，安全策略的正确制定、安全策略到 SA 的正确转换以及安全关联的正确实施是 IPSec 实现的核心问题。

安全策略库（SPD）说明了对 IP 数据报提供何种保护，并以何种方式实施保护。SPD 中策略项的建立和维护应通过协商，而且对于进入和外出处理都应该有自己的策略库。对于进入或外出的每一份数据报，都可能有三种处理：丢弃、绕过或应用 IPSec。SPD 提供了便于用户或系统管理员进行维护的管理接口。可允许主机中的应用程序选择 IPSec 安全处理。SPD 中的策略项记录对 SA（SA 束）进行了规定，其字段包含了 IPSec 协议、模式、算法和嵌套等要求。SPD 还控制密钥管理（如 ISAKMP）的数据包，即对 ISAKMP 数据包的处理明确说明。

SPD 是利用 radix 树型结构来构造，每一个结点就是一个策略项。策略项中包含一个 SAID 数据结构，它是 SPD 与 SAD 之间的接口。可以由它来查找 SAD，从而指定相关的 SA（或 SA 串）。这样使得策略项可以对应相关的一个 SA 或者多个 SA（SA 串）。

SPD 中策略项的查找是通过选择符来进行的。SA 或 SA 束的粒度决定于选择符。通过选择符，可以找到外出或进入的 IP 包应该实行的策略项。两个策略项的选择符可以相同。选用匹配项，保证 SPD 始终以同样的顺序进行查找，这样就保证了匹配策略项的一致性选择。

4.3.1 安全策略库的实现过程

本系统构建了基于 Radix 树的策略表，存放在内核中。radix 是一种二叉树，它在查找速度上有很大的优势，并且在构造上简单、灵活。将各策略项组成一个由目的地址和掩码作为关键字的策略表。查找策略表的目标就是为了找到一个最能匹配给定目标的特定地址。称这个给定的目标为查找键（search key）。所谓最能匹配的地址，也就是说，一个能够匹配的主机地址要优于一个能够匹配的网络地址，而一个能够匹配的网络地址要优于默认地址。

如果策略表项是针对某个具体的主机的，则掩码默认为全 1 的比特，即 Oxffffffff；如果策略表项是针对某个局域网的，则掩码即为网络掩码。策略表项的查找方法为：使用查找键和策略表项的掩码进行逻辑与运算，根据得到的结果进行比较，如果结

果与该路由表项的目的地址相同，则称该策略表项是匹配的。对于某个给定的查找键，可能会从路由表中找到多条这样的匹配路由，根据最优匹配的原则，就保证了匹配的一致性和可预测性。

4.3.2 SPD 模块中的关键数据结构

SPD 中策略项的查找是通过选择符（目的地 IP 地址、源 IP 地址、数据保密等级、源端口和目的端口）来进行的。通过 SPD 可以在树中找到对应的节点，从而找到策略项。SPD 的数据结构如下：

```
struct SPD{
   struct     rjtentry er_rjt;     /*进入 Radix 树的入口*/
   struct     sa_id er_said;       /*通过该 id 号，查找 SAD 库，可获得相应 SA*/
   uint32_t   er_pid;              /*进程号*/
   uint32_t   er_count;            /*使用次数*/
   struct     sockaddr_encap er_eaddr;      /*地址信息*/
   truct      sockaddr_encap er_emask;      /*源地址掩码信息*/
}
```

4.4 IPSec 处理模块的实现

IPSec 处理模块实现 ESP 协议处理，在实现 IPSec 时应满足以下几个几则：

1）IPSec 实施不应该干扰传输层与网络层、网络接口层与网络层之间的接口。也就是说，不应该让传输层和网络接口层知道网络层传来的包是由 IP 组件还是由 IPSec 组件传输的。对于由传输层和网络接口层往网络层传送的数据包，也同样不应要求区分应该交给 IP 组件还是交给 IPSec 组件。

2）IPSec 模块中尽量避免分段重组这些功能，尽量避免做重复性工作，尽可能利用 IP 层本身的功能。

3）尽量避免不需要 IPSec 处理的数据包进入 IPSec 模块。在 TCP/IP 协议中传输层的数据传给 IP 层，IP 层预处理完后，通过寻路，找到下一步将要进行的相应处理模块的网络设备接口，然后将该 IP 包放到该设备的发送队列中等待处理。在实现中，通过向 Linux 系统注册一个虚拟的网络接口，然后添加路由将需要 IPSec 处理的输出接口指向该虚拟设备，就可以在该虚拟设备的发送函数中进行相应的 IPSec 处理。并且在该虚拟设备对象内部构造一个指向真实网络设备的指针。在 IPSec 处理完后，调用 ip_send()，将数据通过此接口发送给 IP 协议相应的处理模块。

虚拟网络接口定义如下：

```
Struct device dev_ipsec(){
```

```
  ipsec(),        /*虚拟设备接口名称*/
  ipsec_init,     /*虚拟设备初始化*/
}
```

在 Linux 系统初始化时，当执行到 net/IPv4/af_inet.c 时调用了 IPSec 初始化函数 ipsec_init()，进行 IPSec 的初始化。

```
Static inet_init(void){
  #if defined( CONFIG_IPSEC)
  extem ipsec_init();
  #end if
}
```

在注册 ipsec()虚拟接口时，将进行一系列初始化工作，使 ipssec()的数据结构与 IPSec 处理理模块绑定，初始化工作描述如下：

```
ipsec_init(void){
   dev->open=ipsec_open;                  /*加载模块*/
   dev->do_ioctl=ipsec_ioctl;             /*虚拟接口与实际接口挂接模块*/
   inet_add_protocol(&esp_protocol);      /*注册 ESP 协议模块*/
   init_pfkeyfd(void);                    /*创建 PF_KEY 型的套接口*/
}
```

在隧道模式中实现需要的 ESP 协议头定义如下：

```
struct esphdr{
   ipsec_spi_t      esp_spi;      /*安全参数索引*/
   _u32             esp_rpl;      /*抗重播序列号*/
   _u8              esp_iv[8];    /*初始化向量*/
}
```

其中 ESP 的协议结构定义如下：

```
struct inet_protocol esp_protocol ={
   recv_ipsec ,      /*定义此协议处理函数*/
   NULL;             /*错误处理函数*/
   0;                /下一个头
   50;               /*协议的编号*/
   0;                /*协议共享字段*/
   NULL;             /*私有数据*/
   "ESP"             /*协议名称*/
}
```

4.5 策略和 SA 管理模块的实现

任何实现都必须提供一个管理接口，用于配置策略、故障诊断以及状态查询或是手工进行启用和停用隧道。策略和 SA 管理模块是对策略和 SA 进行管理的应用，在用户级实施。用户和这一模块交互，该模块与内核进行交互，以便手动更新内核的 SPD 和 SA。该模块设计了两个用户接口：对 SPD 的增添、查找和删除；对 SADB 的手工添加、查找和删除。

在 Linux 下为了和在 Windows 下一样对用户提供良好的用户接口采用了 Gtk/Gnome 构件编程，并利用 Linux 下的图形界面生成工具。Glade 制作图形界面的用户管理程序，使管理员能够方便地对 SA 进行手工配置。应用程序的图形界面如图 4.4 和图 4.5 所示：

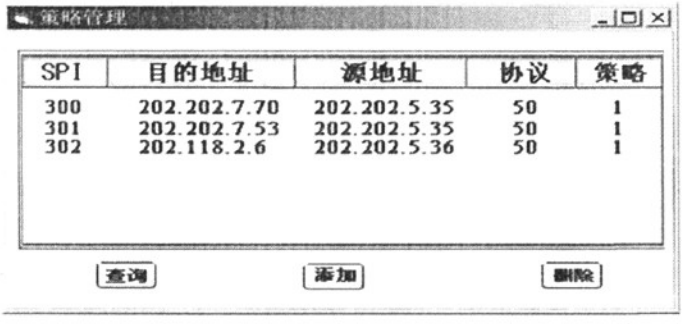

图 4.4 策略管理的程序界面

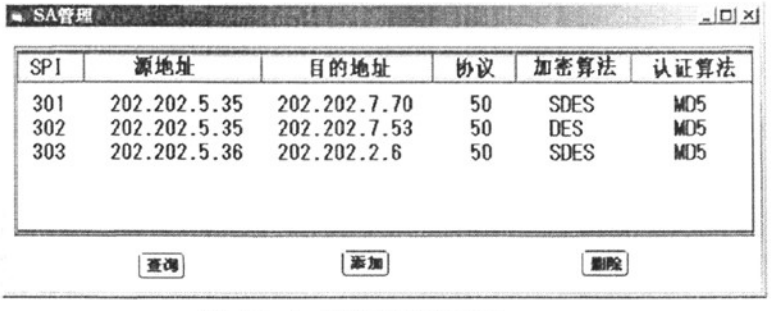

图 4.5 SA 管理的程序界面

4.6 实验分析

4.6.1 实验环境

为了分析网关系统的性能，在师仓库网络中搭建如图 4.6 所示的实验环境。

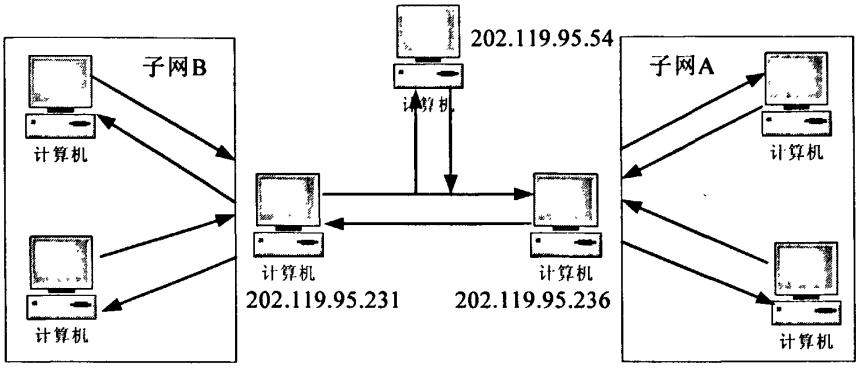

图 4.6 实验环境

其中 202.119.95.231 和 202.119.95.236 为配置的两个 VPN 网关，这两台用做 VPN 网关的机器的操作系统都是 Linux7.2，它的内核是 2.4.2 版。为了真实的模拟实际的网络环境，让子网 A 和子网 B 分别由这两个 VPN 网关控制。

这两个子网是实际的师仓库网络的一个模拟，根据实验的环境中，采用隧道方式的 ESP 协议实施，在配置 VPN 网关的时候，必须保证这两个 VPN 网关采用完全相同安全策略，否则，就无法进行相互的通信。在本系统的实验中采用的是 DES 加密算法和 MD5 认证算法。

4.6.2 安全性测试

安全性是 VPN 网络的主要性能，VPN 网络的提出就是为了解决网络的安全性问题，所以首先要验证 VPN 网关的安全性。在实验中把子网 A 中的一台主机配置成 FTP 服务器，让 202.119.95.54 这台机器充当黑客，这样当子网 B 中的某台主机和它通信的时，就利用 Ethereal 这个免费的网络协议检测程序截获双方通信的数据包。在本实验中，在 FTP 服务器上放置一个内容为 HACKRE 的文本文件。

开始实验的时候不加载 IPSec 模块，这样原来的 VPN 网关现在就变成了普通意义上的网关。实验时从子网 B 中的某台机器上到子网 A 中的 FTP 服务器上下载这个文本文件，在下载的过程中，在充当黑客的那台机器上使用 Ethereal 这个工具软件来截获数据包，实验结果显示文本文件的内容是 HACKER，双方通信使用的 IP 地址和协议也可以看到，实验表明网络很不安全。

为了做比较，进行了第二个实验，在这个实验中把 IPSec 模块加载上，进行与上面同样的操作。在这个实验中，通过 Ethereal 这个工具软件截获数据包，但文件内容显示为乱码。在对截获的 IP 包进行详细的分析后，也只能获得是 202.119.95.236 和 202.119.95.231 这两个网关在进行通信这样的信息。同时在子网 B 中的主机下载的文件的内容就是 HACKER。

通过实验表明 VPN 网关的确是安全的，是能够保护网络中网关之间的通信的。

4.6.3 VPN 网关的性能测试

对于性能的测试，在本实验中采用常见的 Ping 命令。Ping 命令程序通过发送一份 ICMP 回显请求报文给主机，并等待返回 ICMP 回显应答来测试另一台主机是否可达。它不用经过传输层（TCP/UDP）。由于 ICMP 是定义在第三层的，所以通过计算连续发送 10 个大小不一样的数据包的平均时间来对普通网关和 VPN 网关的包处理速度进行比较。

表 4.1 是在实验过程中得到的数据，发现 VPN 网关的处理速度比普通的网关要慢些，并且发现 CPU 的速度越高，相应的 VPN 网关的包处理能力也明显加强。

表 4.1 实验数据

源主机	目的主机	模式	包大小(bytes)				性能比
			50	1000	2000	5000	
PIII600	C366	普通	1.71ms	2.34ms	4.24ms	9.27ms	40.7%~55.4%
		IPSce	4.24ms	5.18ms	7.50ms	16.65ms	
PIII500	PIII500	普通	0.430ms	1.893ms	3.895ms	8.559ms	74.7%~86.7%
		IPSec	0.581ms	2.327ms	4.598ms	9.873ms	

VPN 网关处理过程中要做加密和解密的工作，消耗不少的时间，本系统采用的是软件加密和解密的，这样就会比普通网关的处理速度下降不少。解决的方法之一是采用专门的硬件来进行加密和解密。

第五章 下一步工作

在实际应用中 IPSec VPN 仍存在一些问题：在处理过程中要做加密和解密的工作，消耗不少的时间；只支持 IP 协议；对于较复杂的网络其稳定性不够；没有成熟的统一的 QoS 管理策略等。限制了其在实际中的应用和发展。本章主要对进一步提高 IPSec VPN 性能以及多协议问题在实际中的应用做出一些研究。

5.1 基于硬件加速提高性能

IPSec 协议涉及大量的安全计算，这些安全计算包含了对称密码算法（DES，AES）、公钥密码算法（RSA，DSA）和哈希函数（MD5，SHA）等。随着计算机网络传输速率的不断提高和 IPSec 协议中加/解密、消息认证等各种安全计算任务的增加，采用软件实现 IPSec 协议所要求的安全计算使得网络设备负载明显提高和吞吐量的显著下降，导致服务器、网关、路由器和交换机等关键网络设备的处理性能大大降低。在基于 IPSec 的 VPN 中，大量经过 IPSec 网关的数据流造成网络资源的紧张。并且 IPSec 提供的是逐包加密，因此网络资源紧张程度更大，为了防止网络性能降低，网关应该被设计达到尽可能与线速度接近的速度。选择硬件实施方案，将计算量大的加密、解密操作在硬件中完成，可以提高整个系统的性能。

在 VPN 网关中采用 Hifn7854[36]安全加密芯片，代替软件来实现一些安全通讯协议，如 IPSec、PPTP、SSL、IKE 等，来提高系统对 IPSec 处理的性能，简化 IPSec 系统设计和实现的复杂性。

1）硬件设计目标

目前，所有的加密产品都是特定的硬件形式，网络加密产品也不例外。采用硬件有许多好处。首先是速度，加密算法含有很多对明文的复杂运算，常常是高强度的计算任务，计算机微处理器对此效率不高。将加解密移到芯片上会使整个系统速度加快，这对数据在网络中高效地传输是非常重要的。另外，对运行在没有物理保护的一般计算机上的某个加密算法，可以用各种跟踪工具秘密修改算法而使任何人都不知道。而硬件加密设备可以安全地将加密算法封装起来，能避免这类事件的发生。对于硬件加速来说，主要考虑两点问题：安全性和速度。由于算法被固化在硬件中，也就是芯片，算法被篡改的可能性很小。硬件加速器和微处理器比较如表 5.1。

表 5.1 硬件加速器和微处理器比较

硬件加速器与微处理器比较	硬件加速器	作为硬件集成电路，具有高速处理能力，具有优化的体系结构和指令集，比 CPU 有着更高的处理性能，能够满足网络高速发展的需求。
	微处理器	作为通用处理器，由于考虑各种应用的需要，具有一般化的通用体系结构和指令集。处理速度一般相对较慢，可扩展性差，很难满足网络安全高速发展的需求。

采用专用芯片安全处理器 Hifn7854 完成 IPSec 数据通道的加密、解密、IPSec 安全协议封装等处理。安全处理器通过 PCI 总线与主机间进行通信。芯片拥有自己的 SDRAM，主机将要处理的数据以及相关的命令参数通过 PCI 总线传送到 SDRAM 中，芯片再根据命令对传入数据进行相关的操作，对于数据包的加解密，MAC 计算，以及压缩解压等操作会在包处理引擎中完成。同时，安全芯片还可以根据用户需要通过随机数生成器产生随机值。最后，将输出再通过 PCI 送给主机。

Hifn7854 外接的 MIPS CPU 负责执行安全处理器的初始化、SA 的建立以及例外处理等功能。与 IPSec 协议处理相关的各种数据结构存储在 SRAM 存储器中。关于 Hifn 7854 处理器的详细说明参见[36]。

Hifn 7854 处理器最大的优点是用硬件加速处理过程，支持 OC-12 的端口速率，因此可以满足网络主机对 IPSec 的处理要求；同时，Hifn7854 处理器采用的 PCI 总线接口可以与目前大多数网络处理器（如 IBM 的 4GS3、Intel 的 IXP2000 等）无缝结合，在网关的接口上实现通道模式的 IPSec。

2）VPN 网关硬件设计方案

基于 Hifn 7854 处理器设计如图 5.1 所示硬件加速安全处理器。下面以在网关上的应用为例，简要介绍对接收到的 IPSec 报文的处理过程。

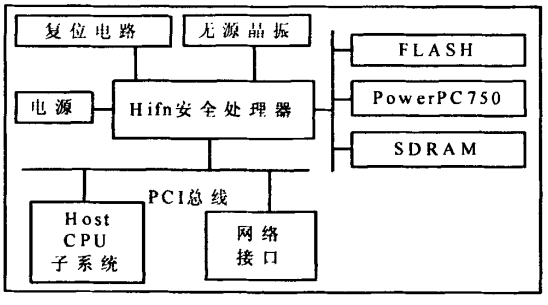

图 5.1 硬件实现框图

- Host CPU 从网络接口接收到 IPSec 协议报文，将其存放到系统内存中；
- Host CPU 根据 IPSec 头中的 SPI 域查找 SA，若找不到，丢弃该报文，处理结束；
- 根据查找到的 SA，Host 在内存中生成报文处理的命令字，并通知 Hifn 7854；
- Hifn 7854 通过 DMA 方式将该报文及处理命令字从系统内存拷贝到本地

SRAM 中排队；

- 安全压缩核读取报文的处理命令字，并根据 SA 查找处理该报文控制信息，包括使用的加密算法，算法执行的上下文信息，DPU 开始执行的指令地址等；
- DPU 根据上述信息控制安全压缩处理引擎对报文进行处理，并将处理后的报文 和状态信息存到本地 SRAM 存储器中缓存；
- Hifn 7854 通过 DMA 方式将报文送回系统内存，并通知 Host CPU；
- Host CPU 对报文进行其他处理（如送高层协议栈）或继续转发该报文。

5.2 IPSec VPN 在实际运用中的多协议问题

在实际应用中，当企业向 VPN 过渡时，如果原来有使用非 IP 协议的应用，那么这些应用就难以在基于 IPSec 的 VPN 中继续使用，包括常用的路由协议如 OSPF（Open Shortest Path First）等也不能够通过 IPSec 隧道，这样对于企业想继续利用原有的应用就显得比较被动，一些中大型、复杂的网络管理就变得比较困难，不利于网络的扩展。目前常用的隧道协议比较如表 5.2。

表 5.2 隧道协议的比较

隧道协议	对应 OSI 层	数据安全	多协议支持	QoS 能力
PPTP	数据链路层	弱	支持	不支持
L2F	数据链路层	弱	支持	不支持
L2TF	数据链路层	一般	支持	弱
IPSec	网络层	最强	不支持	不支持
GRE	网络层	一般	支持	不支持
MPLS	数据链路层网络层	强	支持	较强
ATM	ATM 层	强	支持	强

通过比较，IPSec 的安全性最强，但不能够支持多协议；GRE 安全性较弱，但支持多协议；而 MPLS 和 ATM 对数据安全有一定的保障，也支持多协议，但其网络的构建比较复杂，对核心设备要求较高，需要耗费大量的资源。经过研究和分析，可以利用 GRE 支持多协议的特点，采用 GRE Over IPSec 的方法来改善基于 IPSec 的 VPN 网络。这种方法简单、投入少、灵活方便。

利用 GRE 的优点，把需要通过 IPSec 隧道的非 IP 数据包先进行 GRE 封装，然后再进行 IPSec 封装，最后通过 IPSec 隧道送到目的网络。在目的网关进行相反的操作，先去除 IPSec 封装，然后再去除 GRE 封装，最后将还原的非 IP 数据包送到目标主机上。

致 谢

在课题和硕士论文完成之际，谨向在我攻读硕士学位过程中曾经指导过我的老师，关心过我的朋友和所有帮助过我的人们致以崇高的敬意和深深的感谢！

衷心感谢我的导师孙志刚老师。孙老师尽管工作繁忙，但还是抽出时间跟我们讨论学术方面的问题，使我们对学术研究有了更具体的认识和更多的兴趣。老师时刻关心着我的成长，从课题的开展到论文的完成一直严格要求，悉心指导。每当我遇到问题时，孙老师总是及时帮助解决，他对我的信任和鼓励是我努力的动力和源泉。另外孙老师高深的学术造诣、渊博的知识、严谨的治学作风以及努力的工作精神都给我留下深刻印象，这也将是我一生的奋斗目标。

感谢在百忙之中抽出时间评阅论文，参加答辩并进行指导的各位专家和教授。

感谢和我一起学习生活过的同学和朋友：路云志、刘志强、张之文、苏栋林、林海龙、赵学明等等。感谢他们在学习上给予我的帮助，在生活上给予我的照顾，从他们身上我学到了很多东西。

感谢继教一队的全体同学，他们和我一起走过了这段美好的时光，寒窗生活因为他们而变得温暖。

感谢继教一队给我们提供的良好生活环境，以及所有关心和帮助过我的人们。

最后把我所有的感激都送给我的父母和爱人。你们的爱让我的生活变得更加充实，你们的鼓励让我的意志变得更加坚强，你们的期望让我的自信变得更加十足。你们对我的关心、理解和支持是我顺利完成课题研究取之不尽的动力！

参考文献

[1] Steven M. Bellovin. Security Problem in the TCP/IP protocol suite[J]. Computer Communications Review, April 1989 :20~21.

[2] 翁晶，邓元庆. 网络安全中认证协议的分析和比较[J]. 计算机安全，2004(5): 26~32.

[3] 南湘浩. 与认证系统有关的几个问题[J]. 计算机安全，2003(11): 11~15.

[4] Ray Hunt . Internet firewall security: policy, architecture and transaction services[J]. Computer Communication, 1998: 904~908.

[5] 袁珏，王能，曹晓梅. IPSec 协议在 VPN 中的应用[J]. 计算机应用研究，2002(5): 64~67.

[6] 刘上伟，周安民. 基于 IPSEC 的 VPN 实现及安全性分析[J]. 网络安全技术与应用，2005(7): 37~39.

[7] S. Kent, R. Atkinson. Securuty Architecture for the Internet Protocol[J]. RFC 2401, Nov 1998: 55~57.

[8] Bryang Juhan. A Framework for IP Based Virtual Private Networks[J]. Internet Draft IETF, September 1998: 1~56.

[9] D uszenko. IP VPN networks[J]. Studio Information, 2003(12): 307~324.

[10] Nora Boukrai Aliajine. Security and Auditing of VPN[J]. IDG Books Worldwide Inc, 1998(4): 1~370.

[11] 陈性元，宋国文. IP-VPN 及关键技术[J]. 电信科学, 2000(5): 38~41.

[12] R. Glenn, S. Kent. The NULL Encryption Algorithm and Its Use With IPsec[J]. RFC2410, November 1998: 1~6.

[13] 戴宗乾，唐三平. VPN 与网络安全[M]. 北京:电子工业出版社, 2002.

[14] S. Kent, R. Atkinson. Security Architecture for the Internet Protocol[J]. RFC2401, 1998:105~120.

[15] 冯登国. 网络安全原理及技术[M]. 北京:科学出版社, 2003.

[16] Agarwal, Mittal. IPSec(Internet Protocol Security): Redefining Internet in the Context of Pervasive Computing[J]. Proceedings of ICCC 2002-15th International Conference on Computer Communication, 2002. 2(2): 543~603.

[17] Ruixi Yuan, W. Timothy Strayer. Virtual Private Network Technologies and Solutions[D]. China Electric Power Press, 2003.

[18] 田春歧，王立明，蔡勉. IPSec VPN 的研究和分析[J]. 计算机工程与应用，2004(4):163~166.

[19] N Ferguson, B Schneior. A Cryptographic Evaluation of IPSec[M]. Counterpane Internet Security. Inc.

[20] [美]尤戴尔，马远，李钟. 实用 Internet 群件[M]. 北京:中国电力出版社，2003.

[21] C. Madson, R. Glenn. The Use of HMAC-SHA-1-96 within ESP and AH[J]. RFC2404, November 1998: 1~7.

[22] C. Madson, N. Doraswamy. The ESP DES-CBC Cipher Algorithm With Explicit IV[J]. RFC2405, November 1998: 1~10.

[23] D. Harkins, D. Carrel. The Internet Key Exchange(IKE)[J]. RFC2409, 1998:403~438.

[24] Charlie Kaufman. Internet Key Exchange(IKE) Protocol[J]. Draft-ietf-ipsec_ikev2-17 Txt, 2004:88~104.

[25] Haddad. H, Berenjkoub. M, Gazor. S. Aproposed protocol for Internet key exchange(IKE)[J]. Canadian Conference on Electrical and Computer Engineering, 2004. 4(4):217~237.

[26] Steven Brown. 构建虚拟专用网[M]. 北京:人民邮电出版社，2000.

[27] 李涛. 网络安全概论[M]. 北京:电子工业出版社，2004.

[28] S. Kent, R. Atkinson. IP Autentication Header[J]. RFC2402, 1998:144~169.

[29] S. Kent, R. .Atkinson. IP Encapsulating Security Payload(ESP)[J]. RFC2406, 1998:34~48.

[30] RFC2409 [S]. The Internet Key Exchange:79~92.

[31] D. Piper. The Internet IP Security Domain of Interpretation for ISAKMP[J]. RFC2407, 1998:236~264.

[32] Orman. H. Oakeley Key Determination Protocol[J]. RFC2412, 1998:78~81.

[33] David A, Solomon, Mark E. Russsinovich. Inside Microsoft Windows 2000(Third Edition)[M]. Washington: Microsoft Press, 2000.

[34] C. Madsen, R. Glenn. The Use of HMAC-SHA-1-96 within ESP and AH. The Network Working Group[J]. RFC 2404, November 1998:211~235.

[35] C. Madson, R. Glenn. The Use of HMAC-MDS-96 within ESP and AH[J]. The Network Working Group. RFC2403, November 1998:24~68.

[36] 7854 Network Security Processor Device Specification [R]. http://www.hifn.com

作者在学期间取得的学术成果

[1] 杨文武，陆庭元．基于 IPSec 的 VPN 网关研究．科技信息，2008(9):206~207.

基于IPSec的VPN网关设计与实现

作者： 杨文武
学位授予单位： 国防科学技术大学

相似文献(10条)

1. 学位论文 邱静 基于IXP425的VPN网关的设计与实现 2006

虚拟专用网是指采用隧道技术以及加密、身份认证等方法，在公共网络（如Internet）上构建专用网络的技术，数据通过安全的“加密通道”在公众网络中传输。随着企业信息化程度的发展，对于虚拟专用网的需求也在增加。对于中小型企业以及家庭SOHO用户来说，最重要的是设计并实现一种安全、方便并且价格低廉的VPN解决方案。而目前市场上的VPN网关要么是大型服务器，价格昂贵；要么是在x86架构上实现VPN，性能较低。本论文作者在广泛深入学习IPSec VPN、网络处理器的相关知识基础上，提出了一种基于IXP425网络处理器的IPSec VPN网关设计方案，既具有优越的性能，又具有低廉的价格，可以满足中小型企业用户和SOHO用户对VPN网关需要。该方案参考开源IPSec VPN网关的实现方案，在嵌入式Linux上实现了VPN网关的基本功能，即提供数据完整性、数据安全性以及抗重放攻击等服务；同时利用IXP425网络处理器在网络加密方面的优势，将影响VPN网关性能的最大瓶颈，即各种加密/解密/认证工作，交给网络处理器引擎来完成，而不占用Xscale内核的处理时间，从而解决了传统VPN网关方案下存在的系统性能瓶颈问题，大大提高了VPN网关的性能。通过对嵌入式系统进行裁减，去掉不必要的功能和模块，减少了系统占用的空间，提高了系统运行的效率。最后，笔者通过在IXP425开发板上实现并运行本方案设计的VPN网关，得到测试数据，充分证明了该方案的可行性和优越性。

2. 会议论文 黄邦强.庞宏冰 嵌入式VPN网关的设计与实现 2003

本论文首先讨论开放的公用网络系统中存在的安全威胁,接下来对虚拟专用网技术进行了简单介绍,论述在公用网上组建虚拟专用网的基本要求和关键技术.在分析现有基于IPSec协议的VPN系统基础上,提出了结合VPN功能的防火墙,以及在RTEMS嵌入式系统上如何实现具有VPN功能的防火墙,构建安全的VPN网关.

3. 学位论文 温翔 基于嵌入式系统RTEMS的VPN网关的研究与实现 2002

网络的开放性、网络协议自身的缺陷以及黑客的攻击是造成网络通信不安全的主要原因.随着Internet的迅速发展,网络安全问题变得越来越重要.当前防火墙是最常用的保护内部网安全的设备.但是防火墙无法保证网络通信的安全.虚拟专用网技术是保护网络信息传输的安全的技术.使用虚拟专用网代替租用线路或专线连接的私人专用网是现在的发展趋势.该论文首先讨论开放的公用网络系统中存在的安全威胁.然后对虚拟专用网技术进行介绍,论述在公用网上组建虚拟专用网的基本要求和关键技术.网络隧道技术是虚拟专用网的关键技术,该文分析比较几种现有的网络隧道技术:PPTP、L2TP和IPSec协议.最后在分析现有基于IPSec协议的VPN系统基础上,提出结合VPN功能的防火墙.并在RTEMS嵌入式系统上实现具有VPN功能的防火墙,构建安全的VPN网关.

4. 学位论文 袁刚 应用于中小型企业的VPN网关的设计与实现 2004

随着Internet应用的普及，由于虚拟专用网(VPN)以Internet媒介，能为企业提供一种安全、经济、灵活的组网方式受到越来越多的关注。VPN大致可分为两种类型：VDPN(VirtualPrivateDialNetwork)和IntranetVPN(即VPN网关)。VDPN是远程用户或移动雇员和公司内部网之间的VPN，它为远程用户或移动雇员和公司内部网之间提供一条安全、经济的数据通道；VPN网关是公司远程分支机构的LAN和公司总部LAN之间的VPN，可使企业中的异地局域网通过Internet实现安全的互连。

目前实现VPN网关主要是基于IPSec协议，基于IPSec协议的VPN网关能为上层应用提供透明的支持，所以是企业构建跨地区信息化管理平台的首选。但由于基于IPSec协议的VPN网关实现复杂(需要更改或重构操作系统的TCP/IP协议栈)，高开发成本导致VPN网关产品昂贵，不能很好满足国内中小型企业的需要(性价比)。

为简化实现难度，降低开发成本，满足国内中小型企业的需要，本文在分析了当前典型的基于IPSec协议VPN网关解决方案的基础上，参照IPSec协议，结合国内中小企业的应用环境(Windows平台)和企业应用(Web服务、FTP服务、邮件服务、ERP系统、OA系统)，运用Winpcap函数库、原始套接字技术和CryptoAPI，提出了一种在应用层加密待转发IP数据包的新设计方案并将之实现。

该VPN网关作为一个模块已经成功用于某无线办公网的控制系统中，对上层的企业应用提供了良好的透明支持；对异地局域网之间传输的IP包提供了私密性保护和完整性、真实性验证；实现了会话密钥的自动分发和更新。在实际应用中，取得了良好的效果。

5. 会议论文 何正安.艾明晶 基于IPSec的VPN网关的研究与实现 2003

讨论了开放的公用网络系统中存在的安全威胁,对虚拟专用网技术进行了介绍,论述了在公用网上组建虚拟专用网的基本要求和关键技术.基于现有IPSec协议的VPN系统,提出并实现了具有VPN功能的防火墙,构建安全的VPN网关.

6. 期刊论文 虞金华.YU Jinhua 构建VPN网关,实现一卡通网络数据安全传输 -常熟理工学院学报2005,19(6)

利用Linux系统内含的IPSec功能来构建VPN网关,从而有效解决一卡通数据在校园网上安全传输的问题.

7. 会议论文 陆骞 利用FreeS/WAN构建基于Linuz的IPSec VPN网关 2006

VPN虚拟专用网(Virtual Private Network)是一种通过采用隧道、加密和身份认证技术来构建专用网络的技术。IPSec作为新一代网络安全协议，能够构建安全VPN，实现数据的安全通讯。本文在研究IPSec体系结构及实现方法的基础上，对Linux平台下IPsec协议的实现软件—FreeS/WAN的系统结构进行了深入剖析，给出了其配置、测试步骤和源码框架，并为Linux下的VPN构建提供了一个较为实用的示例。

8. 学位论文 吴凌俊 高性能VPN网关的研究与实现 2004

随着网络带宽的不断提高，研究和实现1000兆高速、高可用的VPN网关成为当前网络安全的前沿研究热点。本文通过在Linux2.4.18-3核心TCP／IP协议栈的多个关键点上挂接HOOK函数，完成IPSec协议的封装、加密和认证的功能；实现基于隧道模式的VPN网关。并加载硬件加密卡以提高吞吐量等技术来实现上述目的。

在百兆环境下利用核心线程驱动加密卡。核心线程特有的调度时机决定了报文处理时的不可剥夺，减少了进程切换。CPU的大部分时间用于网络报文的处理，确保了系统吞吐量。没有时间片概念的SCHED_FIFO调度策略，在系统中存在众多进程时可以被优先调度。

在千兆环境下利用中断底半部BH驱动加密卡。网卡软中断NET_RX_SOFTIRQ与加密卡中断BH在Linux系统中的相同优先处理级别，避免了采用核心线程方案时核心线程调度执行时间过少导致的低吞吐量。NET_RX_SOFTIRQ与BH对CPU的竞争使得加密之前和之后的两部分处理都不会成为系统的瓶颈。在网络流量不是很大时，让加密卡优先处理IKE数据，保证IKE及时的协商，并将协商的密钥交给IPSec。

在IPSec模块初始化时，按照设置的最大队列长度预先分配缓冲队列，并在运行期绕过系统的内存管理。增加系统重组队列的内存容量，减少过多报文重组造成的报文丢弃。利用ICMP向源主机发送由于要分片造成的目的不可达消息，减少报文分片。本机外发报文则根据加密规则相应减小MTU。

采用多个加密卡并行处理，高速缓存的增加提高了加密卡的并行度。报文按顺序入队列，并且先来先处理，保证报文传送的有序性。

通过安全性分析和系统吞吐量测试，证明了这两种方案都是可行的。在两种工作环境下VPN网关都可以达到较高的吞吐量。

9. 期刊论文 张仁.徐敬东.尹乐.吴功宜.ZHANG Ren.XU Jing-dong.YIN Yue.WU Gong-yi 基于Web浏览器的SSL VPN网关系统的设计和实现 -计算机工程与设计2007,28(4)

在分析IPSec VPN与SSLVPN各自优劣势的基础上,提出了一种基于Web浏览器的SSL VPN网关系统的体系结构并给出了具体的实现方法.系统主要由HTTP协议代理Applet、VPN网关上的Web服务器和VPN反向代理服务器组成.系统提供了两种安全访问内网的模式,即隧道模式和替换模式.用户直接通过浏览器从VPN网关上下载Applet或点击用户界面中提供的资源列表就可以安全地访问内部网络中的资源.