



中国科学技术大学

本科毕业论文

题目                      SSL VPN 加密算法研究

英文                      Research of encryption algorithms in SSL VPN

院系                      电子科学与技术系

姓名                      徐飞虎                      学号                      PB05210151

导师                      司虎

日期                      2009 年 6 月

## 致谢

首先要感谢我的指导老师司虎老师，从论文的选题到论文的修改，都是在司老师的悉心指导下完成的，司老师丰富的知识，严谨的治学态度和敬业的精神都让我受益匪浅。在论文的写作期间，他时常关注我的进展情况，并对我遇到的难题，提出了很好的指导。

在此还要感谢管乐和吴晓伟两位同学，他们在我毕业设计的过程中，提供了很多很有价值的参考意见，并且在我遇到难题时，提出了很好的建议，给了我很大的帮助。

感谢四年来的所有任课老师，他们不仅在学习上让我们完成了基础知识的学习，还向我们展示了他们在科研工作中的严谨。还要感谢每一个和我一起走过大学四年的0523的同学，是他们使我的大学生活更加丰富多彩，成为人生永恒的回忆。

感谢四年来在科大给我莫大帮助的朋友们，是他们在困难时帮我排忧解难，在我高兴时与我分享快乐；是他们让我的大学生活充满了欢声笑语；是他们让我的四年科大生活永远难忘。

最后，我要感谢我的爸爸妈妈、我的女友，是他们始终在背后支持我的一切。

再次向所有给我支持和帮助的老师 and 同学表示深深的感谢！

# 目录

致谢.....	1
目录.....	2
中文内容摘要.....	4
Abstract .....	5
第 1 章    绪论.....	6
1.1 背景.....	6
1.2 SSL VPN 发展趋势 .....	7
1.3 本文章节安排.....	9
第 2 章    SSL VPN 基础技术 .....	10
2.1 虚拟专用网介绍.....	10
2.1.1 VPN 的定义和工作原理 .....	10
2.1.2 VPN 的类型和应用 .....	11
2.2 SSL 协议介绍.....	13
2.2.1 SSL 概述 .....	13
2.2.2 SSL 体系结构.....	14
2.2.3 SSL 协议 .....	14
2.3 SSL VPN 和传统 VPN 的比较 .....	16
2.3.1 IPSec 技术介绍.....	16
2.3.2 IPSec VPN 与 SSL VPN 比较 .....	16
第 3 章    SSL VPN 系统结构与关键技术 .....	19
3.1 SSL VPN 概念 .....	19
3.2 SSL VPN 系统结构 .....	20
3.3 SSL VPN 中的关键技术 .....	20
3.3.1 代理和转发技术.....	20
3.3.2 访问控制.....	21
3.1.3 身份认证.....	22
3.1.4 审计日志.....	22
3.4 SSL VPN 的解决方案 .....	23
第 4 章    SSL VPN 的实现 .....	25
4.1 主流 VPN 协议.....	25

4.1.1 PPTP 协议.....	25
4.1.2 L2F 协议和 L2TP 协议 .....	26
4.1.3 IPSec VPN 和 MPLS VPN .....	26
4.2 基于 OpenVPN 的实现 .....	27
4.2.1 Router (tun) 的实现 .....	27
4.2.2 Bridged (TAP) 的实现 .....	31
4.2.3 两种实现方法比较 .....	35
第 5 章 加密算法分析 .....	36
5.1 密码学简介 .....	36
5.1.1 对称密钥和非对称密钥学 .....	36
5.1.2 数字摘要 .....	37
5.1.3 数字证书相关技术 .....	37
5.1.4 公钥基础设施 PKI .....	39
5.2 SSL 算法的选择 .....	40
5.3 OpenVPN 协议和安全性分析 .....	40
5.4 基于 ECC 加密算法的 SSL VPN .....	42
5.4.1 ECC 加密体制 .....	42
5.4.2 ECC 加密体制在 SSL 安全均衡握手方面的优势 .....	43
5.4.3 ECC 公钥算法的优化 .....	44
5.4.4 基于优化 ECC 加密算法的 SSL 安全均衡握手过程 .....	45
5.5 量子密码在 VPN 中的应用 .....	46
5.5.1 量子密码 .....	46
5.5.2 基于 QKD 和 IPSec 的 VPN 体系结构 .....	47
5.5.3 QKD 在 IPSec VPN 中的应用过程 .....	49
5.5.4 QKD 在 VPN 中的具体实施 .....	52
5.5.5 QKD 应用于 VPN 的总结 .....	54
5.6 全文展望 .....	55
参考文献 .....	56

## 中文内容摘要

随着网络技术和普及，Virtual Private Network（VPN，虚拟专用网络）得到了越来越多的应用，而基于Secure Sockets Layer（SSL，安全套接层）技术的VPN系统是目前一种安全易用且低成本的远程访问方案。

SSL VPN是一种新型的VPN技术。随着网络应用的多样性，对远程访问的安全需求日益增加，目前主流IPSec VPN无法满足应用多样性的需求。SSL VPN因其配置方便、与操作系统无关、支持设备广泛等优势，弥补了IPSec VPN的不足，成为VPN领域的研究热点。

本文对SSL VPN的概念、基本原理、关键技术进行了研究，通过OpenVPN软件设计并实现了一个SSL VPN系统，并着重分析了其中的加密算法与安全性，希望提出方法和加密算法来对SSL VPN系统加以提高与完善，其主要内容如下：

1. 详细分析了SSL技术和VPN技术，并对SSL VPN的技术特点进行了总结。
2. 对SSL VPN的体系结构以及其关键技术进行了深入研究。包括代理技术、用户验证技术、访问控制技术和审计日志技术等，并给出SSL VPN的两个解决方案。
3. 归纳总结了一些VPN技术的实现机理，包括PPTP协议、L2F协议和L2TP协议、IPSec VPN和MPLS VPN；在这个基础上，着重对基于SSL的VPN实现方法的代表---OpenVPN做了深入研究，并通过虚拟机(VMware)成功完成两种不同方式实现的实验平台和演示。
4. 针对VPN系统中的研究热点——安全性，对SSL VPN中各种加密算法进行研究，着重分析了ECC加密算法在SSL VPN中的特点与优势；最后将目前发展迅速的量子密码技术引入到VPN系统中，详细分析了QKD(Quantum Key Distribution)技术在VPN中的体系结构、应用方式以及具体实施情况。

关键词：SSL，VPN，SSLVPN，SSL控制协议，OpenVPN，加密算法

## Abstract

With the development of networking technology, Virtual Private Network(VPN) plays a more and more important role today. Secure Sockets Layer(SSL) VPN is a secure and low-cost Remote Access system.

Based on the diversity of network applications, the demand of remote access has a significant growth; IPSec VPN can not satisfy the need of various applications. But SSL VPN which has the priority of easy configuration, no associated with the operating system and extensive support equipment, can make up the shortage of IPSec VPN. So, SSL VPN has become a research hotspot.

This paper studies the characteristics of SSL VPN, Such as concept, basic principles, key technologies, and so on. We also realize a system via the software of OpenVPN and fully analyze the encryption algorithms in SSL VPN, trying to propose methods and encryption algorithms to improve VPN system. The main contents include:

1. We detailed research and analyze SSL technology and VPN technology, summarize the technical features of SSL VPN.
2. SSL VPN architecture and its key technology are deeply analyzed, such as Agent technology, User authentication, Access control, Audit log and so forth. Two solutions of SSL VPN are also proposed.
3. The realization mechanisms of VPN are summarized, such as PPTP protocol、L2F protocol和L2TP protocol、IPSec VPN and MPLS VPN; What we focus on is its representative-OpenVPN; Through VMware, we successfully use two methods to complete the realization demon of SSL VPN.
4. Based on the research topic in VPN system-security, we investigate various encryption algorithm in SSL VPN and detailed analyze ECC encryption algorithm. Finally, with the rapid development of Quantum Cryptography(QC), we propose QC can be better used in VPN system, and deeply analyze architecture and application of Quantum Key Distribution(QKD) is VPN system.

Keyword: SSL, VPN, SSL VPN, OpenVPN, Encryption algorithm, ECC, QKD.

## 第1章 绪论

### 1.1 背景

随着网络经济的发展,企业规模日益扩大客户分布日益广泛,合作伙伴日益增多,各企业开始依靠网络来维持和加强与其生意伙伴,供应商的信息交流。传统的租用专线虽然在安全性方面有足够的保证,但是不能从根本上解决企业用户的困难。因为现代企业分支机构越来越多,企业每年的连网费用巨大而且相互间的网络基础设施互不兼容也更为普遍,同时这样的连网方式不但复杂而且日后网络发展的灵活性扩展性都难以适应现代企业的发展需求,所以企业对自身网络的灵活性、安全性、经济性、扩展性等方面提出了更高的要求,在这种需求下虚拟专用网(VPN)技术应运而生<sup>[1][2]</sup>。

VPN是指依靠Internet服务提供商Internet Supply Provider (ISP)和其它网络服务提供商Network Supply Provider(NSP)在公用网络中建立专用数据通信网络的技术<sup>[3]</sup>。据估计VPN技术可以使企业的远程访问和分支机构连接成本降低50%以上;而它所带来的网络战略则是难以用金钱来衡量的,事实上基于标准的VPN技术近来已成为网络界的新热点,据Infonetics预测,VPN市场每年至少增长一倍<sup>[4]</sup>。目前国内大多数因特网服务提供商都能够提供VPN服务,随着企业现代化程度的不断增加,企业对网络技术的要求也越来越高。传统的VPN逐渐显现出了一些缺点,比如安装维护费用高、配置复杂、可扩展性差等。而基于SSL协议的VPN(SSL VPN)可以弥补这些缺点,提供更为完善的远程访问服务。SSL VPN为远程访问提供了一种新的解决方案,SSL VPN被定义为使用SSL协议和代理技术向终端用户提供远程内部网络资源的授权安全访问的VPN技术。

SSL VPN主要支持对HTTPS,以SSL为基础的HTTP。应用的访问也可支持一些基于SSL的网络、应用程序如电子邮件客户端程序等。SSL VPN经常被称之为无客户端,因为目前大多数计算机上时都已经安装了支持HTTP和HTTPS的Web浏览器<sup>[1]</sup>。

SSL VPN的主要功能是提供网络连接,远程用户可以使用SSL VPN对企业网络进行远程接入,类似于现在的IPSec和PPTP提供的客户端接入业务。然而SSL VPN只需要标准Web浏览器将其内置的SSL/HTTPS功能用作高度安全的传输机制,无需在最终用户的电脑上安装配置客户端。

SSL VPN提供基于Web的应用的接入,如Microsoft Outlook Web Access OWA基于标准的电子邮件,以及文件共享和远程登录的应用等。在使用了代理技术后SSL

VPN也可以支持一般的网络应用。SSL VPN还可以提供更安全的管理功能；传统的网络层VPN对企业的网络层接入并未加以限制，易遭受来自远程设备的特洛伊木马和病毒的威胁。与之不同的是SSL VPN对网络中应用进行控制接入而不是完整的网络层接入，从而进一步提高了企业安全性。

SSL VPN具有安全性高、易管理和访问便捷等优点，SSL VPN的主要优点是不用安装客户端程序，远程用户基本上不需要IT部门的支持就可以随时随地从任何支持SSL协议的浏览器上安全地访问企业内部资源。从而最大限度的减少了分发和管理客户端软件的麻烦，降低了系统部署成本和IT部门日常性的管理支持工作费用。而且SSL VPN适用性强，广泛支持B/S和C/S应用以及手持设备等。因此，SSL VPN成为了远程访问市场上的热点，SSL VPN系统的研究开发具有广阔的市场前景和实用意义。

## 1.2 SSL VPN 的发展趋势

目前SSL VPN技术飞速发展，不同的产品除了基本的远程功能之外，在易用性和安全性方面各有特色，如Content aware intrusion prevention技术可以过滤HTTP数据、阻塞特定的HTTP流量，还可以防止蠕虫病毒和URL漏洞攻击，并且可以支持大多数C/S的网络程序、支持直接IP地址访问，NetScaler的SSL VPN还有应用层优化功能，可以压缩TCP的头从而加快用户访问的速度。Whale公司产品注重客户端安全性，它的产品可以检查客户浏览器的设置，如果不符合安全要求，会拒绝用文件下载请求，用户退出的时候，还可以删除保存在Internet临时文件夹中的内容，删除地址自动完成，密码自动完成的记录。Menlo logic的产品支持单点登录Single Sign On，在访问不同资源的时候，不用多次进行用户名密码的验证。Motivus的产品可以提供对知识产权保护功能，可以限制用户的访问权限，比如文档只读等。Nokia的SSL VPN可以通过客户端完整性检查，检查开放的端口和文件，授予用户不同的安全级别根据安全级别，授予用户访问权限。F5的Fire Pass产品可以实现缓存和临时文件的自动删除，还可以进行客户端完整性检查，并对文件下载进行控制。这些新技术不断发展并得到了用户的认可，可见SSL VPN的发展方向有这么几个：

### 1. 强有力的安全保障

SSL VPN是建立在企业移动人员和公司总部之间的一条专用通道，在这条通道中传输的数据是企业内部数据，是不公开的。因此必须要在安全的前提下进行远程连接。其安全性包含三层含义：一是客户端接入的安全性；二是数据传输的安全性；三是内



部资源访问的安全性。对于远程移动用户，用户身份验证是安全的第一个环节，方便可靠的验证方法是未来发展的趋势。第二，在合法用户通过验证连接到公司内部网后客户端设备的安全性就成为影响整个局域网的核心。虽然内部网络建设非常坚固，但是由于移动人员可以使用笔记本电脑、PDA或者网吧中的公用电脑登录公司内部系统，客户端设备是否安装了个人防火墙防病毒软件就成为远程访问系统安全的关键点。客户端检测功能将成为SSL VPN系统的必备功能，它可以扫描出客户端安装的防火墙及防病毒程序，并确定它们的安全级别，从而判定此设备是否满足接入条件，确保整个系统的安全性。第三，在移动用户完成远程访问后，黑客或不法分子可以通过拷贝，复制驻留在客户端缓冲区内数据盗取企业机密。为此，在用户离线后需要自动清除用户缓冲区的内容。

## 2. 全面支持网络应用

最早推出的SSL VPN产品只支持Web应用的远程连接。由于大多数企业应用状况非常复杂，企业往往不仅仅应用基于B/S结构的应用程序，还要应用传统的C/S应用和其它非TCP应用，如UDP，这在一定程度上制约了SSL VPN的发展。随着产品研发和升级的推进，SSL VPN必将全面支持各种网络应用，包括基于TCP协议的B/S和C/S应用，UDP应用，如WebDav SMB文件共享访问、标准email协议、Lotus Note、Telnet服务、远程终端、Citrix等

## 3. 易于管理和维护使用操作性强

SSL VPN的突出优势就是移动性强，易用性强，但这些特性往往会增加管理难度。所以以后的SSL VPN产品需要界面简单、使用方便、可以灵活、细致地设置访问权限，基于用户/组/角色的认证机制，对每个文件，网址或应用都可进行单独的访问权限设置，使访问控制更易于管理。

## 4. 提高运行效率

由于是集中系统，SSL加速决定整个网络的吞吐量。如果SSL加速跟不上，远程接入就会比实际的Internet接入带宽低很多。可以采用专门的SSL加速硬件，提高VPN的响应速度。另外，还可以使用数据压缩技术，对所传输的数据进行压缩后再进行传输，这样就提高了整个网络的运行效率和实用性。

## 5. 服务质量保证(QoS)

传统VPN可以为企企业数据提供不同等级的服务质量保证，SSL VPN作为新的远程访问解决方案，对QoS的支持也是必需的。不同的用户和业务对服务质量保证的要求差别较大。如移动办公用户，为其提供广泛的连接和覆盖是保证VPN服务的一个主要

因素；而其它应用(如视频等)则对网络时延及误码率等提出了更明确的要求，所有上网应用均要求网络根据需要提供不同等级的服务质量。

### 1.3 本文章节安排

本文一共分为五章：

第一章，对SSL VPN技术及其研究现状进行总结，归纳其发展趋势和研究动向。

第二章，详细分析了SSL VPN的基础技术：VPN技术和SSL协议安全技术，先说明VPN的定义，工作原理和类型；然后介绍SSL安全协议的体系结构和协议SSL协议族的构成，并研究如何提高SSL协议的性能。最后对SSL VPN和传统VPN在安全性工作层面，系统差异等方面做一下综合比较。

第三章，归纳总结了SSL VPN的体系结构，并详细讨论SSL VPN中的关键技术，包括代理技术、用户验证技术、访问控制技术和审计日志技术等；并给出SSL VPN的两个解决方案。

第四章，首先对目前主流的一些VPN技术的实现机理进行调研，包括PPTP协议、L2F协议和L2TP协议、IPSec VPN和MPLS VPN，在这个基础上，对这些主流技术做一下小结，着重对基于SSL的VPN实现方法的代表---OpenVPN做一下研究，并在VMware中成功通过两种不同实现方式搭建其实现的实验平台和演示。

第五章，对SSL VPN中应用的各种加密算法进行了研究，着重分析了ECC加密算法在SSL VPN中的特点；将量子密码技术引入到VPN系统中，详细分析了QKD技术在VPN中的体系结构、应用方式以及具体实施情况；最后对全文进行了总结与展望。

## 第 2 章 SSL VPN 基本概念

SSL VPN将SSL安全协议使用在传统的VPN技术中，VPN技术和SSL安全协议是SSL VPN的基础，VPN是目前广泛应用的网络技术，也是广域网建设的最佳解决方案。SSL安全协议为两台机器上的应用层协议提供安全通道，它使用数据加密、服务器认证、消息完整性、以及可选的客户端认证为应用层提供通信安全。

### 2.1 虚拟专用网介绍

#### 2.1.1 VPN 的定义和工作原理

##### 1. 定义

VPN即虚拟专用网络，指的是依靠ISP(Internet服务提供商)和其它NSP(网络服务提供商)在公共网上为一组用户或一些点实现的一种专用通讯网络<sup>[5]</sup>。VPN具有三层含义：

1. 它是虚拟的网，即没有固定的物理连接网络只有用户需要时才建立。
2. 它是利用公众网络设施构成的专用网，用于构建VPN的公共网络包括因特网、帧中继、ATM等。
3. 在公共网络上组建的VPN，利用VPN的隧道技术、认证技术和加密技术，能够在一种不可信、不安全的网络上的两个单独实体之间建立一条安全的私有的专用信道。像企业现有的私有网络一样，提供安全性、QoS、可靠性和可管理性等，使得企业网络几乎可以无限延伸到地球的每一个角落，从而以安全、低廉的网络互联模式为应用服务。

##### 2. 工作原理

VPN系统使分布在不同地方的专用网络在不可信任的公共网络(如：因特网)上安全地通信。它采用复杂的算法来加密传输的信息，使得需要受保护的数据不会被窃取。一般来说其工作流程大致如下<sup>[11]</sup>：

1. 要保护的主机发送不加密信息到连接公共网络的VPN设备。
2. VPN设备根据网络管理员设置的规则确认是否需要对数据进行加密或让数据直接通过。
3. 对需要加密的数据，VPN对整个数据包，包括要传送的数据发送端和接收端的IP地址进行加密，并附上数字签名以便接收方验证发送方的真实身份。

4. VPN设备加上新的数据包头,其中包括目的地VPN设备需要的安全信息和一些初始化参数。

5. VPN设备对加密后的数据、验证数据、源IP地址、目标VPN设备IP地址进行重新封装,重新封装后数据包通过虚拟隧道在公共网络上传输。

6. 当数据包到达目标VPN设备时,数字签名被核对无误即通过身份验证后,数据包被解密发送到目的地。

## 2.1.2 VPN 的类型和应用

### 1. VPN的类型

互联网时代,为解决企事业单位通过公共通信网络将地域分散的分支机构互连提出了内联网Intranet概念,为解决企事业单位通过公共通信网络与合作伙伴实现互连互通和信息交换而提出了外联网Extranet 概念,以及为解决企事业单位职员出差或旅游在外,通过公共通信网络共享内联网资源而提出了远程接入Remote access概念<sup>[2]</sup>。按这些应用领域可以将VPN分为三类:即Access VPN(接入VPN), Intranet VPN(内联网VPN)及Extranet VPN(外联网VPN)。

#### (1) 远程接入VPN

VPN的一个基本的应用就是满足漫游用户访问企业网络资源的要求。由于工作的需求,公司的员工经常会到异地出差,而在公司外部,出于安全性考虑,一般不允许员工访问公司内部服务器,同时企业网的地址空间规划与因特网也完全不同,因此无法使用许多企业高层应用软件,比如企业资源规划系统, Notes服务器等等,这会大大限制移动用户对企业网络资源的使用。为了解决这个问题,以往的做法基本是采用企业设置网络接入服务器, 移动用户采用长途电话拨号的方式拨号到公司远程接入端口实现。采用这种方式主要的缺点就是成本昂贵。通过接入VPN提供的移动用户接入企业网的业务充分利用了因特网资源,通过共享的IP网络承载用户业务,使业务提供成本大大降低。但是,共享的使用方式也带来了安全问题,用户的数据流在共享的IP网络中传递,容易被别人监听和拦截,也容易被黑客伪造攻击,因此在漫游用户的使用过程中,可选择使用多种验证,授权,加密机制,以最大限度地防止受到安全攻击。比如L2TP协议和IPSec协议的嵌套使用,这样既可以提供验证功能,又保证了数据流传输过程中的安全性。当然,这些安全措施的使用增加了对处理设备性能的要求,带来了额外的费用,但总的业务提供成本仍远小于以前的长途拨号接入模式的费用。

#### (2) 内联网VPN

在VPN技术出现以前，如果公司两个异地机构的局域网想要互联，一般都会采用租用专线的方式。但租用DDN电路昂贵的成本一般只有大型企业才负担得起，一般的中小公司都会望而却步。如果通过基于IP的VPN业务提供企业网之间的互联，则可以方便而低廉地为企业提供的互联业务，使网络更广泛地为用户所用。内联网VPN业务用于连接公司内部各办事处，可以建立企业总部及分支机构间的安全连接，为企业现有的专线网络增加或建立新的带宽。这样，只有企业分支机构和服务供应商之间的线路需要按月付费，不再需要从企业总部到企业分支机构的专线连接，因此可以节省大量的专线费用。对于国际性的连接，这种费用的节省就更为明显，因为国际专线的费用非常昂贵。另外，IP VPN可以增强企业的地域覆盖，使得企业在没有直接网络连接的情况下快速地满足增加企业的分支机构所带来的需求。由于企业只需要租用从办公地点到运营商网络边缘的专线，而运营商的物理网络资源由多个企业共享使用，这在降低单个企业的业务使用成本的同时，大大提高了运营商网络的使用效率，也就可以为运营商获取了更大的收益。

### (3) 外联网VPN

外联网VPN业务用于将公司与外部供应商，客户及其它利益相关群体相连接。外联网使公司与其供应商，销售商和客户之间能进行电子商务等活动，其最主要的好处是改善提供商务的速度和效率。当然，由于这些连接可能是各私有网络之间的远程和专用的连接，或者是一个私有网络和互联网间的连接，因此必须要求在这些连接之间具有必要的接入控制和验证机制，以便向各用户群（包括企业伙伴和客户）提供对公司各业务和数据的动态访问权限。

## 2. VPN的应用

VPN的本质在于利用互连网、帧中继或者ATM作为广域网WAN的中枢来补充或替换专用网上昂贵的长距离专线或拨号上网，通过这种新的公共网络发送私人信息。由于可以依靠公共服务而不必为各自的专用网的运行承担很大的责任，除了可以节省资金，在很多方面都将使企业受益匪浅。VPN能够连接一个公司内部的办公组织、远距离办公、在外地出差的员工、甚至其世界范围内的顾客和产品供应商。由于互连网的存在，任何地方的使用者都可以通过本地交换机或者其它服务方式相互联系。同时，VPN还能通过互连网使用户更容易使用，并通过网络的无所不在性和便捷的访问特性来发挥更大的灵活性。因此，我们有理由相信，对于企业网建设，VPN是一个不容质疑的选择，VPN将会为众多企业网络的需求提供更具魅力的解决方法。总结起来有三种应用场合特别适合使用VPN<sup>[12]</sup>：

### (1) 分支机构连接网络

分支机构连接指的是在一个组织内部安全地连接两个相互信任的内部网。在这种情况下,需要做的不仅是要防范外部入侵者对公司内部网的攻击,还要保护在公共因特网上传送的公司数据。

### (2) 业务合作伙伴/供应商网络

今后的工业界,只有那些能方便实惠且安全地同其业务合作伙伴、附属机构以及供应商进行通信的公司才有能居于主导地位。为了实现上述目标,许多公司选择了帧中继方案或是使用租用线,但这些方法通常费用比较昂贵,而且受到地域的限制。VPN技术为用户提供了另外一种选择,利用它可以以高的成本效率实现扩展的公司私有网络,由于借助于因特网或其它公用网络,它还不受地域限制。

### (3) 远程访问网络

无论是在家中还是在旅途中,一个远程用户都希望能够安全、高效地访问自己公司的内部网。可能现在很多用户使用的仍是昂贵的长途拨号方式,但如果利用因特网实现远程访问,其费用肯定会大大降低。举个例子,如果你在家里或在路上,却想访问公司内部网上的一个保密文件,这时候你可以通过拨号到某个ISP来连入因特网,然后你就可以同内部网中的文件服务器通信,取到需要的文件。

## 2.2 SSL 协议介绍

### 2.2.1 SSL概述

SSL最初是由Netscape Communication公司设计开发的,又叫安全套接层协议,主要用于提高应用程序之间的数据的安全系数。SSL协议的整个概念可以被总结为:一个保证任何安装了安全套接字的客户和服务端间事务安全的协议,它涉及所有TCP/IP应用程序<sup>[6]</sup>。SSL安全协议主要提供三方面的服务:

#### 1. 用户和服务器的合法性认证

认证用户和服务器的合法性,使得它们能够确信数据将被发送到正确的客户机和服务器上。客户机和服务器都是有各自的识别号,这些识别号由公开密钥进行编号,为了验证用户是否合法,SSL协议要求在握手交换数据进行数字认证,以此来确保用户的合法性。

#### 2. 加密数据以隐藏被传送的数据

SSL协议所采用的加密技术,既有对称密钥技术也有公开密钥技术。在客户机与服务器进行数据交换之前,交换SSL初始握手信息,在SSL握手信息中采用了各种加

密技术对其加密，以保证其机密性和数据的完整性，并且用数字证书进行鉴别。这样就可以防止非法用户进行破译。

### 3. 保护数据的完整性

SSL协议采用Hash函数和机密共享的方法来提供信息的完整性服务，建立客户机与服务器之间的安全通道，使所有经过安全套接层协议处理的业务在传输过程中能全部完整准确无误地到达目的地。

#### 2.2.2 SSL体系结构

SSL被设计成使用TCP来提供一种可靠的端到端的安全服务，SSL不是单个协议而是两层协议，如图2-1 所示：

SSL握手协议	SSL 修改密文协议	SSL警报协议	HTTP
SSL 记录协议			
TCP			
IP			

图2-1.SSL两层协议模型

可见SSL握手协议、SSL修改密文协议、SSL告警协议构成了较高层的协议，SSL记录协议为这些较高层协议提供了基本的安全服务，并与客户服务器上的HTTP交互作用。

SSL中两个重要的概念是SSL会话和SSL连接：

- (1) 连接是提供恰当类型服务的传输(在OSI分层模型中定义), 对于SSL, 连接是点对点的关系, 连接是短暂的, 每个连接与一个会话相联系.
- (2) SSL的会话是客户和服务端之间的关联, 会话通过握手协议来创建, 会话定义了加密安全参数的一个集合, 该集合可以被多个连接所共享, 会话可以用来避免为每个连接进行昂贵的新安全参数的协商.

实际上每个会话存在一组状态, 一旦建立了会话就有当前的操作状态用于读和写(即接收和发送), 另外, 在握手协议期间, 创建了挂起读和写状态, 一旦握手协议达成成功的结果, 挂起状态就变成当前的状态.

#### 2.2.3 SSL协议

##### 1. SSL记录协议

SSL协议为SSL连接提供了两种服务，机密性和报文完整性；

- (1) 机密性：握手协议定义了共享的，可用于对SSL有效载荷进行常规加密的密钥；

(2) 报文完整性：握手协议还定义了共享的，可以用来形成报文的鉴别码MAC的密钥。

SSL记录协议的完整操作：记录协议接受传输的应用报文，将数据分片成可管理的块，可选的压缩数据，再应用MAC，加密，增加首部，在TCP 报文段中传输结果单元，被接受的数据被解密、验证、解压缩和重新装配，然后交付给更高层的协议。

SSL将被发送的数据分为可供处理的数据段，它没有必要去解释这些数据并且这些数据可以是任意长度的非空数据块。接着对这些数据段进行压缩，加密，然后把密文交给下一层网络传输协议处理。对收到的数据，处理过程与上相反，即解密、验证、解压缩、拼装然后发送到更高层的协议。

## 2. 修改密文规约协议

修改密文规约协议是二个较高级SSL协议之一，也是最简单的。这个协议由单个报文组成，该报文协议的唯一目的就是使得挂起的状态被复制到当前状态，改变了这个连接将要使用的密文簇。改变加密约定协议的存在是为了使密码策略能得到及时的通知，该协议只有一个消息（是一个字节的数值）。客户方和服务器方都会发出改变加密约定消息，通知接收方后面发送的记录将使用刚刚协商的加密约定来保护。客户方在发送握手密钥交换和证书检验消息（如果需要）后发送改变加密约定消息；服务器方则在成功处理从客户方接收到的密钥交换消息后。发送一个意外的改变加密约定消息将导致一个Unexpected Message警报。当恢复之前的会话时，改变加密约定消息将在问候消息后发送。

## 3. 警报协议

警报协议是SSL记录层支持的协议之一。警报消息传送该消息的严重程度和该警报的描述。与其它消息一样警报消息也经过加密和压缩，使用当前连接状态的约定。警报协议包括关闭警报和错误警报，关闭警报是为了防止截断攻击Truncation Attack，在关闭警报之后收到的数据都会被忽略。错误警报用于SSL协议中的错误处理，当检测到错误时，检测的这一方就发送一个消息给另一方，传输或接收到一个致命警报消息，双方都马上关闭连接，要求服务器方和客户方都清除会话标识，密钥以及与失败连接有关的信息。

## 4. 握手协议

SSL握手协议是对服务器进行认证并确立用于保护数据传输的加密密钥，建立使发送和接收受保护数据成为可能所需要的共享状态，它有四个目的；

(1) 客户端与服务器就用于保护数据的加密算法与压缩算法达成一致。



- (2) 客户端对服务器进行身份认证。
- (3) 它们需要确立一组由那些算法所使用的加密密钥。
- (4) 握手协议还可以选择对客户端进行身份认证。

握手协议是SSL中最复杂的协议，该协议使得服务器和客户能够相互鉴别对方的身份，协商加密和MAC算法以及用来保护在SSL记录中发送的数据加密密钥。在传输任何应用数据前，使用握手协议。握手协议由一系列在客户和服务器之间交换的报文组成，握手协议可以分为四个阶段，分别是建立安全能力这个阶段用于开始逻辑连接并建立和这个连接关联的安全能力；服务器鉴别和密钥交换；客户鉴别和密钥交换；阶段完成安全连接的建立。

## 2.3 SSL VPN 和传统 VPN 的比较

VPN的发展过程中产生过多种VPN技术，实现VPN的方法也有很多种。但就计算机网络来说其实现VPN所选用的层次，可以有网络层VPN、数据链路层VPN等。网络层VPN多采用IP协议，而数据链路层VPN则由ATM或帧中继虚电路来实现，随着技术的发展，通过IP层实现VPN已成为目前业界主流。

### 2.3.1 IPSec技术简介

IPSec协议是由IETF的IPSec工作组提出的将安全机制引入TCP/IP网络的一系列标准，包括安全协议（验证头AH和封装安全净荷ESP）、安全联盟、密钥管理和安全算法等，它定义了IP数据包格式和相关基础结构，以便为网络通信提供端对端加强的身份验证、完整性、反重播和保密性等。使用IETF定义的Internet 密钥交换(IKE)，还提供按需要安全协商和自动密钥管理服务。IPSec可保障主机之间，安全网关之间（如路由器或防火墙）或主机与安全网关之间的数据报的安全。IPSec协议可以实现各种方式的VPN。

### 2.3.2 IPSec VPN和SSL VPN比较

#### 1. 安全通道Secure Tunnel

IPSec和SSL这两种安全协议，都有采用对称式和非对称式的加密算法来执行加密作业。在安全的通道比较上，并没有优劣之分，仅在于应用上的不同。

#### 2. 认证和权限控管

IPSec采取Internet Key Exchange (IKE)方式使用数字证书Digital Certificate或是一组Secret Key来做认证，而SSL仅能使用数字证书。如果都是采取数字证书来认证，两者在认证的安全等级上就没有太大的差别。SSL的认证，大多数的厂商都会建置硬件的

token。来提升认证的安全性对于使用权限的控管，IPSec可以支持Selectors，让网络封包过滤阻隔某些特定的主机或应用系统。但是实际作业上，大多数人都是开放整个网段Subset以避免太多的设定所造成的麻烦。SSL可以设定不同的使用者，执行不同的应用系统，它在管理和设定上比IPSec简单方便许多。SSL VPN的认证方式比较单一，而且一般是单向认证。支持其它认证方式往往要进行长时间的二次开发。IPSec VPN认证方式更为灵活。

### 3. 安全测试

IPSec VPN已经有多年的发展，有许多的学术和非营利实验室，提供各种的测试准则和服务，其中以ICSA Labs 是最常见的认证实验室，大多数的防火墙，VPN 厂商，都会以通过它的测试及认证为重要的基准。但SSL VPN在这方面，则尚未有一个公正的测试准则，因此它的安全性和健壮性还有待检验。

### 4. 应用系统的攻击

远程用户以IPSec VPN的方式与公司内部网络建立连接，之后内部网络所连接的应用系统都是可以侦测得到，这就给黑客提供了攻击的机会。若是采取SSL VPN 来联机，因为是直接开启应用系统并没在网络层上连接，黑客不易侦测出应用系统内部网络结构所受到的威胁，仅是所联机的这个应用系统攻击机会相对就减少。

### 5. 病毒入侵

一般企业在Internet联机入口，都是采取适当的防毒侦测措施。不论是IPSec VPN 或SSL VPN联机，对于入口的病毒侦测效果是相同的，但是比较从远程客户端入侵的可能性，就会有所差别。采用IPSec联机，若是客户端电脑遭到病毒感染，这个病毒就有机会感染到内部网络所连接的每台电脑。相对于SSL VPN的联机，所感染的可能性，会局限于这台主机，而且这个病毒必须是针对应用系统的类型，不同类型的病毒是不会感染到这台主机的。

### 6. 防火墙上的通讯端口Port

在TCP/IP的网络架构上，各式各样的应用系统会采取不同的通讯协议，并且通过不同的通讯端口来作为服务器和客户端之间的数据传输通道。以Internet Email 系统来说，发信和收信一般都是采取SMTP和POP3通讯协议，而且两种通讯端口是采用25 端口，若是从远程电脑来联机Email 服务器就必须在防火墙上开放25 端口，否则远程电脑是无法与SMTP和POP3主机沟通的。IPSec VPN联机就会有这个困扰和安全顾虑。在防火墙上，每开启一个通讯端口，就多一个黑客攻击机会。反观之，SSL VPN就没有这方面的困扰。因为在远程主机与SSLVPN Gateway 之间，采用SSL通讯端口443端

口来作为传输通道，这个通讯端口，一般是作为Web Server对外的数据传输通道，因此，不需在防火墙上做任何修改，也不会因为不同应用系统的需求，而来修改防火墙上的设定，减少IT管理者的困扰。

## 7. 工作的层面不同

IPSec VPN工作在OS的TCP/IP协议栈中，SSL VPN工作于应用层。这造成了两者在功能上存在极大的差异，IPSec VPN对TCP数据全部进行了处理；而SSL VPN 只对应用层数据进行处理，如基于HTTP的Web应用、基于TCP的Telnet应用等。基于协议栈的处理需要安装特定的客户端软件，这些软件修改了操作系统的网络系统。SSL VPN是基于应用层的VPN，这就意味着在安全性上已经不仅局限在可以让数据安全过来，而且还关注这过来的数据究竟是什么内容。对于企业而言，采用SSL VPN，不仅可以让外地员工对Web化的ERP应用进行访问，而且可以知道访问ERP应用的数据连接究竟是由哪个员工发起，他究竟有哪些权限来进行操作。SSL VPN的另一个主要局限在于用户主要访问基于Web服务器的应用，而IPSec VPN却几乎可以为所有的应用提供访问，包括客户端/服务器模式和某些传统的应用。

## 8. 平台差异

IPSec VPN平台目前在Windows和Linux系统应用普遍；相反，SSL VPN几乎在所有的平台中都存在，如Linux，SCO UNIX，Solaris等。在平台差异上，国内和国外有不同之处。在国外，比较广泛使用Windows 提供的IPSec VPN，而在国内，电子政务普遍使用精简的Linux系统提供的IPSec VPN。另外，由于IPSec VPN 与操作系统密切相关，因此一般把IPSec VPN作为网关设备，如果作为客户端设备，则成本会相应增加。

## 第3章 SSL VPN 系统结构与关键技术

SSL VPN是一种基于SSL安全协议的应用层VPN，它和传统远程接入VPN的系统结构和访问方法都不一样，本章提出了SSL VPN的一种系统结构，并且详细分析了SSL VPN的关键技术。

### 3.1 SSL VPN 概念

SSL VPN的定义：SSL VPN使用SSL和代理技术向终端用户提供对HTTP，客户机/服务器，以及文件共享资源的授权和安全访问。以HTTPS为基础的SSL VPN，也包括可支持SSL的应用程序，例如电子邮件客户端程序，如Microsoft Outlook 或Eudora。SSL VPN经常被称之为“无客户端”。因为目前大多数计算机上时都已经安装了支持HTTP 和HTTPS（以SSL为基础的HTTP）的Web浏览器。SSL VPN的定义包含了其主要功能特性，分别是代理访问和协议转发，无客户端访问，面向远程访问，细粒度访问控制。

#### 1. 代理访问和协议转发

SSL VPN产品使用代理转发技术支持不同协议，基于不同协议的各种网络应用程序可以通过SSL网关访问内部网服务器，远程用户自由使用内部网的各种资源就像位于同一局域网内一样。SSL VPN能够代理大多数传输层和应用层网络协议，如传输层的TCP UDP协议，应用层的HTTP 协议，SMB/NFS/FTP文件服务协议，SMTP POP IMAP 邮件服务协议等等。

#### 2. 无客户端访问

SSL VPN的主要特点是无客户端访问。只要用户主机上安装了主流浏览器，不需要预先安装任何客户端软件即可访问远程服务。目前的大多数产品为了提供端口转发之类的扩展功能需要安装浏览器内的客户端，比如Java applet ActiveX插件，Netscape 插件，不过这些插件都可以自动安装配置，不需要用户干预。

#### 3. 面向远程访问

VPN的实施有三种方式，远程访问虚拟网(Access VPN)，企业内部虚拟(Intranet VPN)，企业扩展虚拟网(Extranet VPN)。SSL VPN被设计使用在远程访问的环境中，终端用户通过它连接到企业局域网和内部资源。SSL VPN的设计考虑到终端用户的使用经验，包括用户登录屏幕、资源访问界面等等，不需要特定软件支持，方便各种环境下的远程访问。

#### 4. 细粒度访问控制

SSL VPN具有细致具体的访问控制能力，尤其是与大多数IPSec VPN比较。网络管理员能够进行全部范围内的AAA(authentication authorization accounting)管理。SSL VPN提供了精确的访问控制：用户或用户组在什么时候可以访问哪些资源，不可以访问哪些资源，都可以设置。

### 3.2 SSL VPN 系统结构

**SSL VPN系统结构：**SSL VPN网关安装于企业网络的边缘处，为用户访问企业内部的应用提供接口，终端用户和SSL网关之间使用SSL协议建立跨越Internet 的安全隧道。远程用户在任何时间、任何地点、通过任何浏览器远程访问企业内部网的SSL VPN网关设备，并经由该设备访问内部网的认证系统、文件服务器、电子邮件服务器、应用程序等网络资源。

远程访问用户访问SSL VPN时，首先在浏览器中以HTTPS方式输入SSL VPN 网关的Web地址，随后SSL VPN系统通过SSL协议的握手建立SSL安全隧道，SSL VPN网关系统根据请求，首先返回用户登录页面，以此要求进行用户的身份认证。用户输入用户名和密码并提交，这些信息将通过SSL安全隧道传送给Web 服务器的身份认证引擎进行处理。身份认证引擎取出用户信息，根据配置文件确定的认证方式调用相应的认证模块。

如果认证能够通过，身份认证引擎将把用户信息传递给访问策略引擎进行处理。访问策略引擎根据策略数据库定义的对应该用户的策略来赋予该用户相应权限。之后，可访问资源显示模块根据用户权限以及配置文件等资源信息将用户可以访问的资源以html文档的形式传递给Web服务器，Web服务器再将其传递给客户端显示出来供用户选择。当用户选择访问其中某种资源时，请求被浏览器传递给Web服务器的资源访问接口。资源访问接口提供一个统一的访问接口，根据资源类型将请求分派给相应的资源访问客户端。

### 3.3 SSL VPN 中的关键技术

从整体考虑，SSL VPN就是使用基于SSL协议的代理技术，转发终端用户的访问请求，提供对企业内部网络应用的授权安全访问的一种VPN技术。同时为了保证用户会话期间数据的安全，还集成了身份认证，访问控制和审计日志等功能。

#### 3.3.1 代理和转发技术

通过SSL VPN, 用户可以使用浏览器访问私有网络中的各种网络服务, 它的核心功能是代理和转发外部网络中终端用户的访问请求, 使之能够进入内部网络。代理转发的通常实现是在企业的防火墙后面放置一个SSL VPN网关, SSL VPN网关位于企业网的边缘介于企业服务器与远程用户之间, 控制二者的通信, 如果用户希望安全地连接到公司网络上, 那么当用户在浏览器上输入一个URL 后, 连接将被SSL VPN网关取得并验证该用户的身份, 然后SSL VPN网关将提供远程用户与不同的应用服务器之间的连接。这种方式优于传统的IPSec VPN的地方是不需要安装任何客户端软件, 客户端只需要拥有一个支持SSL的浏览器就可以了。

使用代理转发技术, 用户可以访问公司内部的各种网络资源, 但是网络应用不同所需要的代理技术也不一样, SSL VPN中使用的代理技术可以分为三大类: 应用层代理, 协议转发(端口转发)代理, 网络层扩展技术。应用层代理是最基本的一种代理技术, 也是比较简单的一种, 因为它依赖于已经使用了SSL功能的Web应用, 好处是不需要客户端插件, 基本可以使用在各种平台和浏览器上。

应用层代理主要支持基于Web的应用, 对于非Web的应用, 需要将应用Web 化, 也就是为非Web应用提供Web页面。用户提交访问请求后, 相应的应用代理模块与企业网内部的应用服务器连接, 这时候应用服务器返回的数据并不能直接回传给用户, 需要代理模块对数据处理, 将这些服务器对客户端的响应转化为HTTPS协议和HTML格式发往客户端, 终端用户感觉这些服务器就是一些基于Web的应用。这种代理一般用于文件服务, 如微软的CIFS服务或FTP服务。

而有一些应用, 如微软Outlook或MSN它们的外观会在转化为基于Web界面的过程中丢失。此时要用到端口转发代理技术。端口转发用于端口定义明确的应用, 它需要在终端系统上运行一个非常小的Java或ActiveX程序作为端口转发代理, 监听某个端口上的连接。当数据包进入这个端口时, 它们通过SSL连接中的隧道被传送到SSL VPN网关, SSL VPN网关解开封装的数据包, 将它们转发给目的应用服务器, 然后回送服务器响应的数据。

一些SSL VPN网关还可以帮助企业实现网络扩展。这是一种网络层的代理技术, 它可以将终端用户系统连接到企业网上, 提供网络层的连接, 根据网络层信息(如目的IP 地址和端口号)进行访问控制。虽然牺牲了高级别的安全性, 却也换来了复杂拓扑结构下网络管理简单的好处。

### 3.3.2 访问控制

作为安全设备, SSL VPN可以使用基于组的访问控制。有的产品允许网管把Web

应用定义为一系列的URL。一旦定义了应用，用户和组就允许或禁止访问该应用。有的产品可以提供细粒度的控制，不仅做到允许或禁止，还包括所能访问的是什么资源，以及能对这些资源做些什么。

SSL VPN具有细致具体的访问控制能力，网络管理员能够进行全部范围内的AAA(authentication authorization accounting)管理包括每个用户的授权方式，每个用户和组的访问权限等等。由于SSL VPN工作在应用层，网络管理员可以对它进行细粒度的访问控制策略设置，可以基于应用，TCP/IP端口以及用户本身的特性，对SSL VPN进行详细的访问控制和规则设置。

SSL VPN允许IT管理人员具备更高的灵活性，从而为不同的用户群定义丰富的身份验证和授权策略。同时，更高精确性的参数亦能支持动态访问的部署，因为管理员能依据各种因素随时定义不同的访问权限和规定不同的会话角色。这些因素包括：用户身份、网络信任级别、设备类型（PC或无线）、会话参数（实际时间登录时间）、以及安全级别（如双重验证或证书授权）、等等。通过结合动态验证策略、精确角色定义和映射规则、以及基于资源的授权策略中的各项新的变量、网络管理员能根据企业的需要或目的进行有效的访问部署，使其更加符合公司的安全策略和商业要求。

### 3.3.3 身份验证

SSL VPN首先是一种VPN，对于VPN系统，用户身份认证是一项重要的功能，因为业界存在多种不同的认证技术，不同的企业具体使用的用户验证方法也不同，所以SSL VPN系统必须支持多种认证，只有这样才能保证在不同网络环境中可以方便安装部署。SSL VPN的认证方式并没有统一的标准，目前市场上的SSL VPN产品普遍支持的认证方法是RADIUS(Remote Access Dial-In User Service)验证，LDAP(Lightweight Directory Access Protocol)验证，数字证书验证和Active Directory验证，一些产品还提供了像Secure ID卡认证这样的一次性认证方法。SSL VPN的身份验证功能一般使用模块化结构实现，这种设计模式优点在于提高了认证系统的灵活性和可扩展性，便于开发者添加新的认证模块；缺点是由于没有统一的标准，开发者必须为针对每一种认证协议开发自己的认证模块，并自行定义用户验证流程和用户认证方法，而且各个产品自行设计的用户验证流程可能存在安全问题。

### 3.3.4 审计日志

审计日志不仅可以记录用户访问系统的情况，提供系统错误信息，还可以用作系统性能分析的数据来源和发生安全事故后的事实依据，所以，作为安全设备，SSL网关需要强大的审计、日志和报告功能。审计日志需要记录每次配置的修改，每个会话

数据，以显示用户何时登录、何时退出的，以及用户消耗了多少资源。还需要具备日志分析报告功能，统计每日交易数据，以显示正常用户登录情况，非法用户登录情况，网络数据总额等，并用易于理解的形式提交给系统管理员。

有的产品提供了扩展的审计和日志处理功能，不仅能显示谁登录了，还能显示系统本身是如何运行的。可以显示多个图表，网管能够清楚地知道CPU、内存和I/O 负载情况。它们除了拥有需要的所有记录之外，还可以使用FTP、SMTP或者安全拷贝自动地把其记录上传至服务器的某个地方。可以选择某些特殊的用户和应用，可以选择日志记录水平。不论用户是出于调试目的，还仅仅是为了更密切地观察系统的某个部分，这都是一项很好的企业级性能。

### 3.4 SSL VPN 的解决方案

作为一种新的VPN解决方案，SSL VPN提出无客户端的自由、安全接入的新概念，主要是针对流动性较大、无法固定IP 地址的远端用户或是合作伙伴，公司客户等方便、快捷、安全的接入VPN 环境。

SSL VPN的应用有：针对网站信息的保护方案，如使用微软SharePoint Server 搭建的企业内部网，针对Web 应用程序的保护方案，如PeopleSoft 8或基于Web的Onyx CRM等等。下面就给出SSL VPN的两个典型解决方案：

#### (1) Web应用的保护和外部访问

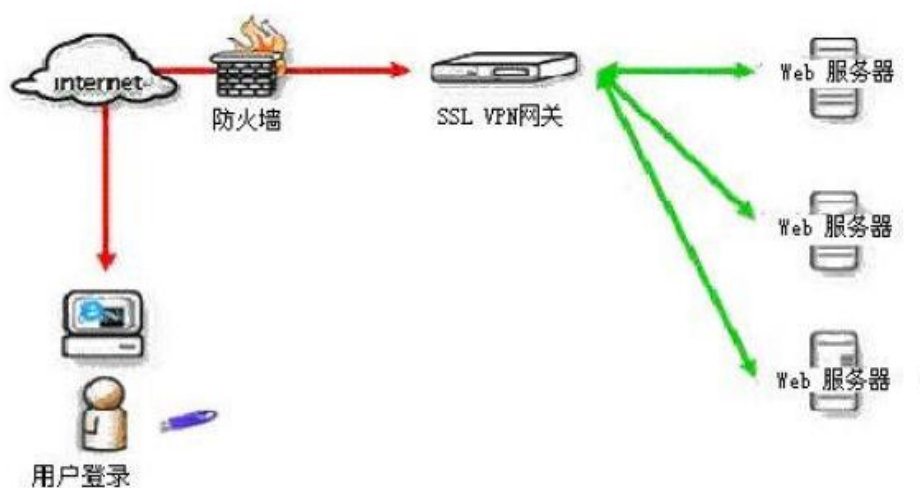


图3-1 针对Web服务的SSL VPN 解决方案

如图3-1所示，此解决方案主要针对企业内部网或需要限制访问的收费网站等



Web应用，步骤如下：

- a) 通过超级终端或Web 界面对SSL VPN 进行网络配置为其分配一个内部IP 地址；
- b) 指定要保护站点的内部IP 地址，并且设定一个与之对应的虚拟IP(VIP)；
- c) 使用管理员界面添加用户为其指定可访问的资源。

经过以上几个简单的步骤，已经对站点资源应用进行了完善的保护，所有对站点的访问都必须通过SSL VPN进行验证，信息的传输过程也完全使用SSL协议进行加密。

SSL VPN对网站资源的保护甚至可以细化到文件，即针对每个组、角色和用户，可以定义不同的访问权限。而且，如果有多个需要保护的站点资源，可以为他们建立一个公用的入口界面，在其中添加多个链接，分别指向不同的站点。

## (2) 网络应用程序的外部访问

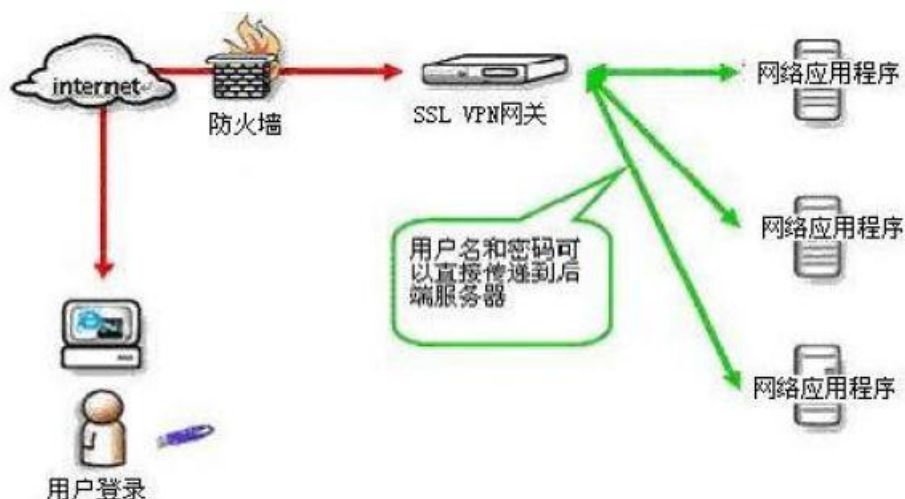


图3-2 针对网络应用程序的SSL VPN解决方案

如图3-2所示，此方案主要针对网络应用程序，应用保护之前的程序客户端和服务端之间通过标准网络协议通信，使用SSL VPN后，程序客户端和服务端之间通过SSL VPN 网关转发数据进行通信。与保护 Web站点相同，可以为多个应用程序建立公用的入口界面，也可以给不同用户分配不同的可用资源，在登录后的入口界面显示不同的链接。

## 第 4 章 SSL VPN 的实现

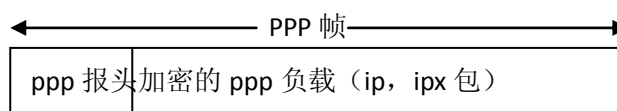
首先对目前主流的一些 VPN 实现技术进行了分析与总结,着重对 SSL VPN 实现方法的代表---OpenVPN 做了研究,并在 VMware 中搭建了其实现的实验平台和演示,结果显示,这种实现方法确实是一种透明的、非面向特定应用、配置简单的 VPN 解决方案。

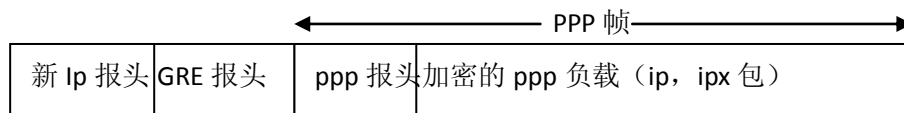
### 4.1 主流 VPN 协议

目前的主流 VPN 技术主要有:第二层的 PPTP(point to point tunneling protocol), L2TP(layer 2 tunneling protocol), L2F(layer 2 forwarding)等,第三层的 IPSec VPN,介于二三层之间的 MPLS 以及 SSL VPN。从应用的广泛程度讲,由于 Windows 的集成, PPTP 用的比较多,新版本的 Windows 也安装更有效更安全的 L2TP 协议, L2TP 的应用也多了起来。由于 IPSec 工作在内核区,且配置比较复杂,尽管它得到了很多权威人士及企业的投资,但其应用确实没有想象中的多。SSL VPN 是一种新的 VPN 技术,它调用的 SSL 协议和其附属的很多加密库,依托 SSL 协议的健壮性,可以在用户空间运行,且配置非常方便,对 NAT(Network Address Translation),动态地址分配等新技术有良好的支持,其缺点是协议栈比较深,可能在抵抗 DoS(Denial of Service)攻击是显得比较脆弱,但瑕不掩瑜,可以预见,在不远的将来,这种轻量级的用户区实现方法,很可能会主导 VPN 的市场。

#### 4.1.1 PPTP 协议:

支持远程访问(remote access, client-LAN)和内外网互联(LAN-LAN)两种模式,主要是基于 PPP 协议的,不同的是增强了 PPP 的认证,压缩和加密功能。数据通路仍是之前的 PPP 通过拨号建立的,而 PPTP 建立了隧道。其接入服务器称为前端处理器 FEP。PPTP 控制包和数据包分开,控制包用 TCP 控制,用于严格的状态查询以及信令信息,数据包先封装在 PPP 头中,在加一层 GRE(generic routing encapsulation, 通用路由封装)头,这样就可以用 IP 头来包装任何形式包,支持多协议。其安全性依赖于 PPP 的机制,包括 PAP, CHAP, MS-CHAP 身份验证机制以及 MPPE(Microsoft point to point encrypt)。其数据包格式如下:



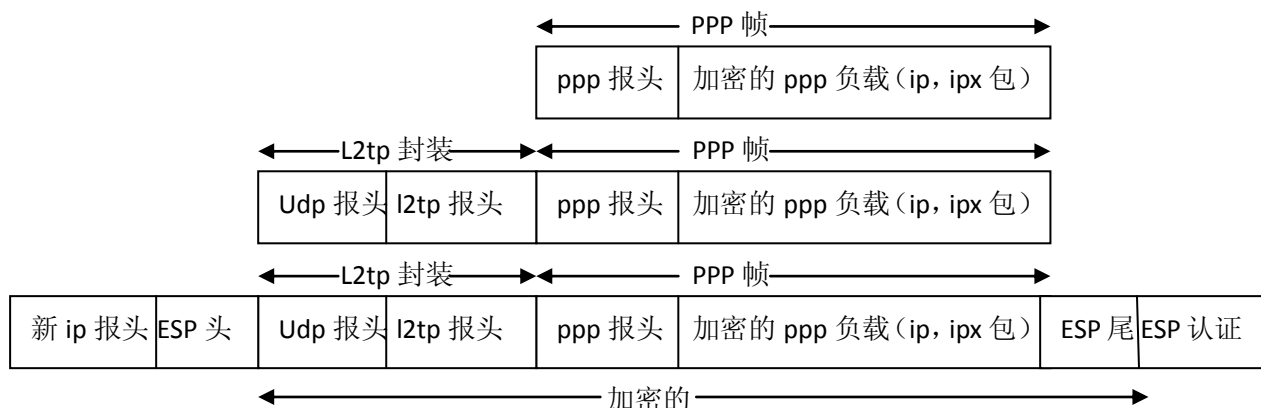


#### 4.1.2 L2F 协议和 L2TP 协议

**L2F 协议:** 1996 年 cisco 提出, 支持多协议, 仅支持 LAN-LAN 模式, 不支持 client-LAN。其基本思想和 PPTP 差不多。

**L2TP 协议:** 综合了 PPTP 和 L2F 的优点, 由 IETF 管理, 于 1999 年 8 月供不其标准 RFC2661. 支持 client-LAN 和 LAN-LAN。支持多协议。与 PPTP 最大的改进是允许成员在物理上连接到不同的 NAS 的 PPP 链路, 在逻辑上的终点是为同一个物理设备。硬件上包括两部分: LAC (l2tp access concentrator) 和 LNS (L2tp network server)。LAC 支持客户端的 L2TP, 发起呼叫, 接受呼叫和建立隧道; LNS 是所有隧道的终点。

L2TP 与 PPTP 最大不同在于 L2TP 把控制包和数据包合二为一, 且运行在 UDP 上, 所以速度上有所提高。L2TP 协议本身不提供加密功能, 而是依赖 IPSec 来实现。其一般数据封装格式如下:



#### 4.1.3 IPSec VPN 和 MPLS VPN

**IPSec VPN:** 第三层的实现。IPSec 本身不是为了实现 VPN 的, 但是, 从它支持的机制, 很容易组建一个基于 IPSec 的 VPN。实际中的实现主要是 LAN 间的隧道模式的 VPN。密钥协商等过程依赖 IKE, 不同于大部分第二层实现方法加密和认证依赖 PPP 协议。

**MPLS VPN:** 根据 PE (服务提供者边缘路由器) 的配置可以实现第 2、3 层的 VPN, 速度比较高, 支持多协议, 但硬件配置较麻烦。

## 4.2 基于 OpenVPN 的实现

实际上, OpenVPN 是一种 VPN 在第二 (tap) 或第三层 (tun) 的实现, 其实现方法如下: 用户程序从/dev/tunX 读入/写入分组, 内核从 tunX interface 写入/读入分组。同样, 若在第二层实现, 则为/dev/tapX 和/dev/tapX interface。运用 SSL 握手协议协商密钥及算法、初始向量等。生成 4 个 (或 2 个) 会话密钥用于加密和认证。加密算法推荐使用 bf-cbc, 即密文块链模式下的 Blowfish 加密。由于 SSL 工作于 TCP 之上, OpenVPN 用 UDP 封装原始 IP 报文, 避免了双重的面向连接服务, 而且实现了隧道服务, 提供点到点的通讯。而用 UDP 封装本来也符合 IP 不是面向连接的性质<sup>[7-9]</sup>。

我们在 Windows XP 上装了 OpenVPN 作为服务器, 在 VMware (虚拟机) 上安装了 Linux, 并编译安装了 OpenVPN 作为客户端, Linux 桥接在 windows 上。我们实现了两种配置, 即 tun 和 tap。都可以实现 server multi-client, 即可以实现多个客户端远程访问服务器, 具体的工作模式是 TSL 认证, tun 隧道, UDP 封装。

首先, 我们在 Windows 上通过 OpenSSL 的 easyrsa 库生成了 CA 的证书 ca.crt, 密钥 ca.key, 并用 ca.crt 签名了两个 client, 生成 client1.car, client1.key, client2.crt, client2.key。并且给证书生成了 dh 参数 dh1024.pem, 用于 dh 密钥交换。这时, 需要用到的密钥都已生成。

### 4.2.1 Router (tun) 的实现:

```
在 server 端, 修改配置文件,
server.ovpn:
port 1194 #在 1194 端口监听
proto udp #用 udp 封装
dev tun #封装 ip 包
ca "C:\\Program Files\\OpenVPN\\easy-rsa\\keys\\ca.crt" #CA 公钥路径
cert "C:\\Program Files\\OpenVPN\\easy-rsa\\keys\\server.crt" #服务器证书
key "C:\\Program Files\\OpenVPN\\easy-rsa\\keys\\server.key" #服务器私钥
dh "C:\\Program Files\\OpenVPN\\easy-rsa\\keys\\dh1024.pem" #dh 参数
server 10.8.0.0 255.255.255.0 #私网的子网, 服务器为 10.8.0.1, 给客户从网段分配地址
ifconfig-pool-persist ipp.txt #用于捆绑之前连过的客户, 给其分配固定的 ip 地址
keepalive 10 120 #每 10s ping 一次, 120s 如无应答, 认为连接已断
comp-lzo #用 lzo 压缩
persist-key #保持密钥参数
persist-tun #保持隧道参数
status openvpn-status.log #连接状态记录在此日志文件中
```

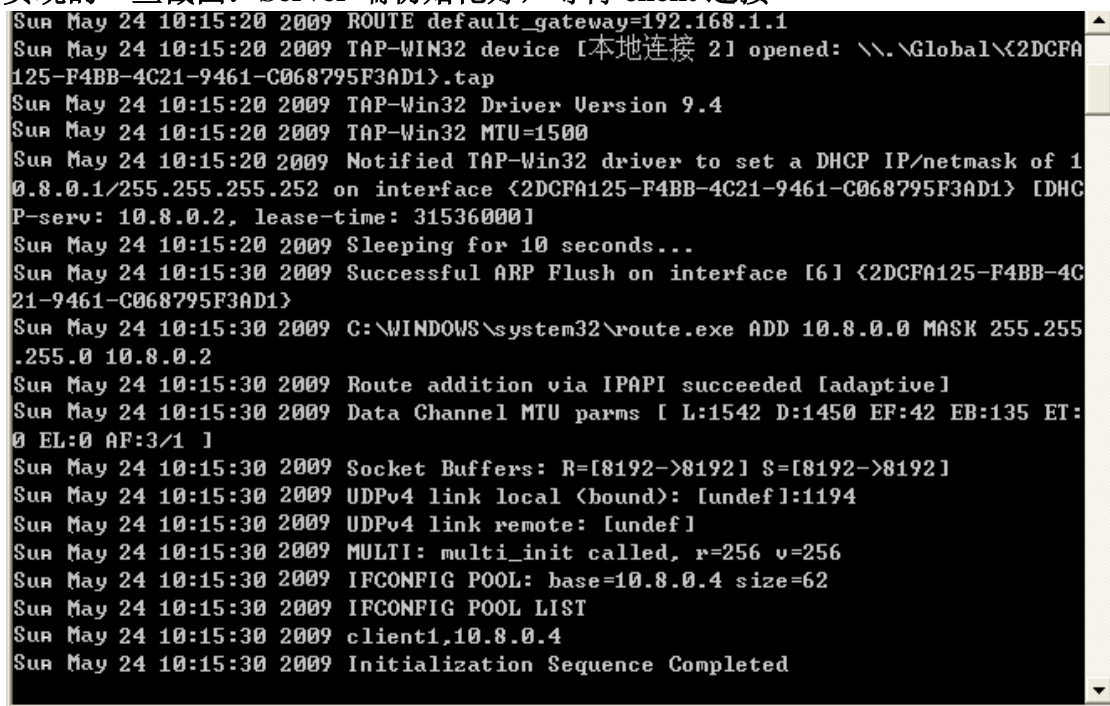
```
verb 3 #日志记录冗余程度为 3, 9 最冗余
management guanle 7505 #使能管理连接, 允许客户端远程管理连接
#end
```

在虚拟机上, 拷贝之前生成的 ca.crt, client1.crt, client1.key, 放在同一个目录下, 修改配置文件.

```
client.cnf:
client #说明是客户端
proto udp #udp 封装
dev tun #封装 ip 包
remote 192.168.1.102 1194 #连接的服务器实际 ip 地址即端口
resolv-retry infinite #如连接不上服务器, 则不断解析主机地址
nobind #连接的端口不固定, 随机的
user nobody #ssl 握手后, 降低程序的级别
group nobody #同上
persist-key #同 server 端
persist-tun #同 server 端
ca ca.crt #CA 公钥
cert client1.crt #自己的证书
key client1.key #自己的私钥
ns-cert-type server #检查服务器证书, 保证 nsCertType 域是 server 端的
verb 3 #同服务端
comp-lzo #启用 lzo 压缩
#end
```

这样, 先启动 server, initialization sequence completed 后, 启动虚拟机上的 client, 连接便建立起来。

实现的一些截图: Server 端初始化好, 等待 client 连接



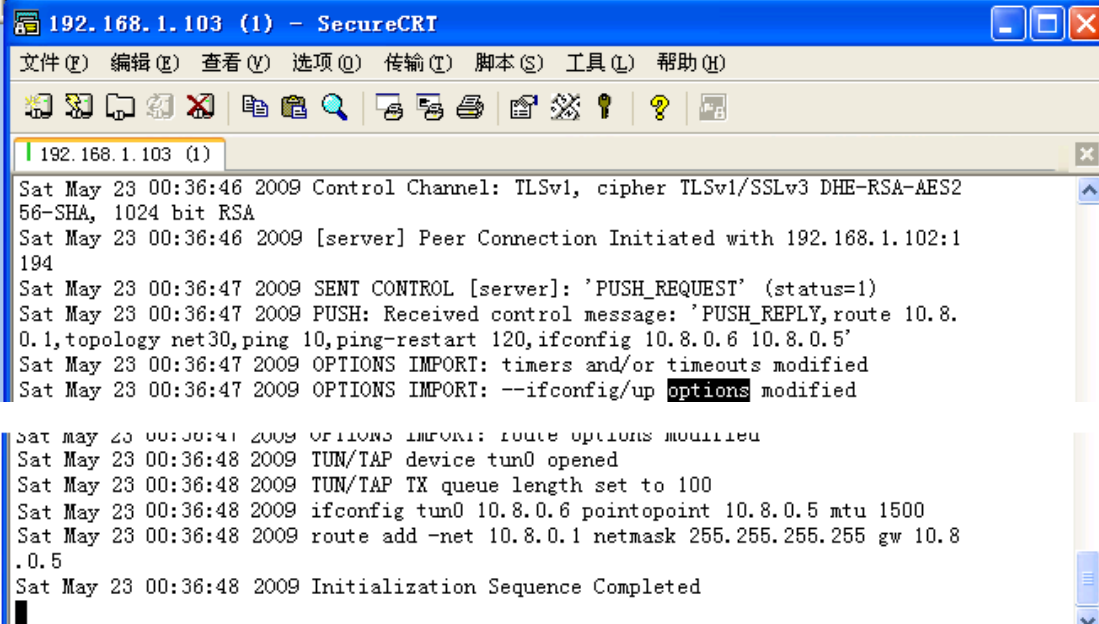
```
Sun May 24 10:15:20 2009 ROUTE default_gateway=192.168.1.1
Sun May 24 10:15:20 2009 TAP-WIN32 device [本地连接 2] opened: \\.\Global\{2DCFA125-F4BB-4C21-9461-C068795F3AD1}.tap
Sun May 24 10:15:20 2009 TAP-Win32 Driver Version 9.4
Sun May 24 10:15:20 2009 TAP-Win32 MTU=1500
Sun May 24 10:15:20 2009 Notified TAP-Win32 driver to set a DHCP IP/netmask of 10.8.0.1/255.255.255.252 on interface {2DCFA125-F4BB-4C21-9461-C068795F3AD1} [DHCP-serv: 10.8.0.2, lease-time: 31536000]
Sun May 24 10:15:20 2009 Sleeping for 10 seconds...
Sun May 24 10:15:30 2009 Successful ARP Flush on interface [6] {2DCFA125-F4BB-4C21-9461-C068795F3AD1}
Sun May 24 10:15:30 2009 C:\WINDOWS\system32\route.exe ADD 10.8.0.0 MASK 255.255.255.0 10.8.0.2
Sun May 24 10:15:30 2009 Route addition via IPAPI succeeded [adaptive]
Sun May 24 10:15:30 2009 Data Channel MTU parms [ L:1542 D:1450 EF:42 EB:135 ET:0 EL:0 AF:3/1 ]
Sun May 24 10:15:30 2009 Socket Buffers: R=[8192->8192] S=[8192->8192]
Sun May 24 10:15:30 2009 UDPv4 link local (bound): [undef]:1194
Sun May 24 10:15:30 2009 UDPv4 link remote: [undef]
Sun May 24 10:15:30 2009 MULTI: multi_init called, r=256 v=256
Sun May 24 10:15:30 2009 IFCONFIG POOL: base=10.8.0.4 size=62
Sun May 24 10:15:30 2009 IFCONFIG POOL LIST
Sun May 24 10:15:30 2009 client1,10.8.0.4
Sun May 24 10:15:30 2009 Initialization Sequence Completed
```

Client 启动后, Damon 的状态提示——上面是 windows 中的 server, 下面是虚拟机 Linux 的 client (用 secureCRT 做 ssh 登陆):

```

message hash 'SHA1' for HMAC authentication
Sat May 23 15:36:46 2009 192.168.1.103:39755 Data Channel Decrypt: Cipher 'BF-CB
C' initialized with 128 bit key
Sat May 23 15:36:46 2009 192.168.1.103:39755 Data Channel Decrypt: Using 160 bit
message hash 'SHA1' for HMAC authentication
Sat May 23 15:36:46 2009 192.168.1.103:39755 Control Channel: TLSv1, cipher TLSv
1/SSLv3 DHE-RSA-AES256-SHA, 1024 bit RSA
Sat May 23 15:36:46 2009 192.168.1.103:39755 [client1] Peer Connection Initiated
with 192.168.1.103:39755
Sat May 23 15:36:46 2009 MULTI: new connection by client 'client1' will cause pr
evious active sessions by this client to be dropped. Remember to use the --dupl
icate-cn option if you want multiple clients using the same certificate or usern
ame to concurrently connect.
Sat May 23 15:36:46 2009 MULTI: Learn: 10.8.0.6 -> client1/192.168.1.103:39755
Sat May 23 15:36:46 2009 MULTI: primary virtual IP for client1/192.168.1.103:397
55: 10.8.0.6
Sat May 23 15:36:47 2009 client1/192.168.1.103:39755 PUSH: Received control mess
age: 'PUSH_REQUEST'
Sat May 23 15:36:47 2009 client1/192.168.1.103:39755 SENT CONTROL [client1]: 'PU
SH_REPLY,route 10.8.0.1,topology net30,ping 10,ping-restart 120,ifconfig 10.8.0.
6 10.8.0.5' (status=1)

```



```

192.168.1.103 (1) - SecureCRT
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(L) 帮助(H)
192.168.1.103 (1)
Sat May 23 00:36:46 2009 Control Channel: TLSv1, cipher TLSv1/SSLv3 DHE-RSA-AES2
56-SHA, 1024 bit RSA
Sat May 23 00:36:46 2009 [server] Peer Connection Initiated with 192.168.1.102:1
194
Sat May 23 00:36:47 2009 SENT CONTROL [server]: 'PUSH_REQUEST' (status=1)
Sat May 23 00:36:47 2009 PUSH: Received control message: 'PUSH_REPLY,route 10.8.
0.1,topology net30,ping 10,ping-restart 120,ifconfig 10.8.0.6 10.8.0.5'
Sat May 23 00:36:47 2009 OPTIONS IMPORT: timers and/or timeouts modified
Sat May 23 00:36:47 2009 OPTIONS IMPORT: --ifconfig/up options modified
Sat May 23 00:36:47 2009 OPTIONS IMPORT: route options modified
Sat May 23 00:36:48 2009 TUN/TAP device tun0 opened
Sat May 23 00:36:48 2009 TUN/TAP TX queue length set to 100
Sat May 23 00:36:48 2009 ifconfig tun0 10.8.0.6 pointopoint 10.8.0.5 mtu 1500
Sat May 23 00:36:48 2009 route add -net 10.8.0.1 netmask 255.255.255 gw 10.8
.0.5
Sat May 23 00:36:48 2009 Initialization Sequence Completed

```

在服务端的日志中有连接状态:

openvpn-status.log:

OpenVPN CLIENT LIST

Updated, Sat May 23 15:57:53 2009

Common Name,Real Address,Bytes Received,Bytes Sent,Connected Since

client1,192.168.1.103:39755,11161,11475, Sat May 23 15:36:45 2009

ROUTING TABLE

Virtual Address,Common Name,Real Address,Last Ref

10.8.0.6,client1,192.168.1.103:39755, Sat May 23 15:40:18 2009

GLOBAL STATS

Max bcst/mcast queue length,0

END

可见: client1 在 192.168.1.103 的 39755 端口接收了 11161 字节, 发送了 11475 字节。client1 的虚拟地址是 10.8.0.6。  
管理界面:

```

vious active sessions by this client to be dropped. Remember to use the --dupl
icate-cn option if you want multiple clients using the same certificate or usern
ame to concurrently connect.
Sat May 23 15:36:46 2009 MULTI: Learn: 10.8.0.6 -> client1/192.168.1.103:39755
Sat May 23 15:36:46 2009 MULTI: primary virtual IP for client1/192.168.1.103:397
55: 10.8.0.6
Sat May 23 15:36:47 2009 client1/192.168.1.103:39755 PUSH: Received control mess
age: 'PUSH_REQUEST'
Sat May 23 15:36:47 2009 client1/192.168.1.103:39755 SENT CONTROL [client1]: 'PU
SH_REPLY,route 10.8.0.1,topology net30,ping 10,ping-restart 120,ifconfig 10.8.0.
6 10.8.0.5' (status=1)
Sat May 23 16:06:27 2009 MANAGEMENT: Client connected from 192.168.1.102:7505
Sat May 23 16:07:22 2009 MANAGEMENT: Client disconnected
Sat May 23 16:07:29 2009 MANAGEMENT: Client connected from 192.168.1.102:7505

194
Sat May 23 00:36:47 2009 SENT CONTROL [server]: 'PUSH_REQUEST' (status=1)
Sat May 23 00:36:47 2009 PUSH: Received control message: 'PUSH_REPLY,route 10.8.
0.1,topology net30,ping 10,ping-restart 120,ifconfig 10.8.0.6 10.8.0.5'
Sat May 23 00:36:47 2009 OPTIONS IMPORT: timers and/or timeouts modified
Sat May 23 00:36:47 2009 OPTIONS IMPORT: --ifconfig/up options modified

```

192.168.1.103 (2) - SecureCRT

文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(L) 帮助(H)

192.168.1.103 (2) 公钥助手

```

guanle@guanle-desktop:~$ telnet localhost 7505
Trying 127.0.0.1...
telnet: Unable to connect to remote host: Connection refused
guanle@guanle-desktop:~$ telnet 192.168.1.102 7505
Trying 192.168.1.102...
Connected to 192.168.1.102.
Escape character is '^]'.
>INFO:OpenVPN Management Interface Version 1 -- type 'help' for more info

```

在 client 通过 telnet 连接到服务器, 可以远程管理连接状态。

在客户端 ifconfig 的结果:

guanle@guanle-desktop:~\$ ifconfig

```

eth0      Link encap:Ethernet  HWaddr 00:0c:29:77:4d:54
          inet addr:192.168.1.103  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe77:4d54/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1498 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1286 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:150039 (146.5 KB)  TX bytes:155333 (151.6 KB)
          Interrupt:16 Base address:0x2024

```

```

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1

```



```

RX packets:486 errors:0 dropped:0 overruns:0 frame:0
TX packets:486 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:24300 (23.7 KB) TX bytes:24300 (23.7 KB)

```

```

tun0    Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
        inet addr:10.8.0.6  P-t-P:10.8.0.5  Mask:255.255.255.255
        UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
        RX packets:7 errors:0 dropped:0 overruns:0 frame:0
        TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:100
        RX bytes:588 (588.0 B) TX bytes:588 (588.0 B)

```

可见，除了跟 Windows 桥接的以太网口 eth0 和本地回环 lo 外，增加了虚拟的点  
到点接口 tun0。这便是虚拟的 VPN 网内部的网口，其 IP 地址为：10.8.0.6。

#### 从客户端 ping 服务器 10.8.0.1:

```

guanle@guanle-desktop:~$ ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_seq=1 ttl=128 time=25.5 ms
64 bytes from 10.8.0.1: icmp_seq=2 ttl=128 time=1.50 ms
64 bytes from 10.8.0.1: icmp_seq=3 ttl=128 time=1.80 ms
64 bytes from 10.8.0.1: icmp_seq=4 ttl=128 time=1.47 ms
64 bytes from 10.8.0.1: icmp_seq=5 ttl=128 time=1.19 ms
64 bytes from 10.8.0.1: icmp_seq=6 ttl=128 time=1.29 ms
64 bytes from 10.8.0.1: icmp_seq=7 ttl=128 time=0.997 ms
--- 10.8.0.1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6000ms
rtt min/avg/max/mdev = 0.997/4.827/25.514/8.448 ms

```

可以 ping 通，说明网络已经正确的建立了。这样建立了多客户到服务器的连接，  
如果要连接到服务器所在的子网（设为 192.168.1.0/24），即访问服务器所在子网的  
任意一台主机，服务端加选项 push "route 192.168.1.0 255.255.255.0"即可。

#### 4.2.2 Bridged (TAP) 的实现:

首先，将本地连接（192.168.1.101）和 OpenVPN 虚拟的 tap 桥接在一起，网桥的  
IP 地址为 192.168.1.104。

##### 网络桥





**Ethernet adapter 网络桥:**

```

Connection-specific DNS Suffix  . : domain
IP Address. . . . . : 192.168.1.104
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.1.1

```

虚拟机的 IP 为 192.168.1.105

```

guanle@guanle-desktop:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:77:4d:54
          inet addr:192.168.1.105  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe77:4d54/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:27 errors:0 dropped:0 overruns:0 frame:0
          TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4237 (4.1 KB)  TX bytes:6844 (6.6 KB)
          Interrupt:16 Base address:0x2024

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1278 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1278 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:63900 (62.4 KB)  TX bytes:63900 (62.4 KB)

```

**Server 端配置:**

Server-bri.ovpn:

port 1194 #在 1194 端口监听

proto udp #用 udp 封装

dev tap #封装以太帧

dev-node tap-bridge

ca "C:\\Program Files\\OpenVPN\\easy-rsa\\keys\\ca.crt" #CA 公钥路径

cert "C:\\Program Files\\OpenVPN\\easy-rsa\\keys\\server.crt" #服务器证书

key "C:\\Program Files\\OpenVPN\\easy-rsa\\keys\\server.key" #服务器私钥

dh "C:\\Program Files\\OpenVPN\\easy-rsa\\keys\\dh1024.pem" #dh 参数

server-bridge 192.168.1.104 255.255.255.0 192.168.1.128 192.168.1.254 #私网的子网, 服务器为 192.168.104, 给客户从网段分配地址的 192.168.1.12 到 192.168.1.254

ifconfig-pool-persist ip.txt #用于捆绑之前连过的客户, 给其分配固定的 ip 地址

keepalive 10 120 #每 10s ping 一次, 120s 如无应答, 认为连接已断

comp-lzo #用 lzo 压缩

persist-key #保持密钥参数

persist-tun #保持隧道参数

status openvpn-status.log #连接状态记录在此日志文件中

verb 3 #日志记录冗余程度为 3, 9 最冗余

management guanle 7505 #使能管理连接, 允许客户端远程管理连接

#end

**Client 端配置:**

只要把 dev tun 改为 dev tap 即可。

启动 server 端和 client 端，下图为双方握手的状态(上为服务器，下为客户端):

```

Sun May 24 16:10:58 2009 Data Channel MTU parms [ L:1574 D:1450 EF:42 EB:135 ET:
32 EL:0 AF:3/1 ]
Sun May 24 16:10:58 2009 Socket Buffers: R=[8192->8192] S=[8192->8192]
Sun May 24 16:10:58 2009 UDPv4 link local (bound): [undef]:1194
Sun May 24 16:10:58 2009 UDPv4 link remote: [undef]
Sun May 24 16:10:58 2009 MULTI: multi_init called. r=256 v=256
Sun May 24 16:10:58 2009 IFCONFIG POOL: base=192.168.1.128 size=127
Sun May 24 16:10:58 2009 IFCONFIG POOL LIST
Sun May 24 16:10:58 2009 Initialization Sequence Completed
Sun May 24 16:18:41 2009 MULTI: multi_create_instance called
Sun May 24 16:18:41 2009 192.168.1.105:41494 Re-using SSL/TLS context
Sun May 24 16:18:41 2009 192.168.1.105:41494 LZO compression initialized
Sun May 24 16:18:41 2009 192.168.1.105:41494 Control Channel MTU parms [ L:1574
D:138 EF:38 EB:0 ET:0 EL:0 ]
Sun May 24 16:18:41 2008 192.168.1.105:41494 Data Channel MTU parms [ L:1574 D:1
450 EF:42 EB:135 ET:32 EL:0 AF:3/1 ]
Sun May 24 16:18:41 2009 192.168.1.105:41494 Local Options hash (VER=04): 'f7df5
6b8'
Sun May 24 16:18:41 2009 192.168.1.105:41494 Expected Remote Options hash (VER=0
4): 'd79ca330'
Sun May 24 16:18:41 2009 192.168.1.105:41494 TLS: Initial packet from 192.168.1.
105:41494, sid=cc8d8e86 56be68a9
Sun May 24 16:18:41 2009 192.168.1.105:41494 VERIFY OK: depth=1, /C=cn/ST=AnHui/
L=HeFei/O=USTC/OU=infosec/CN=openvpn-ca/emailAddress=guanlelennon@gmail.com
Sun May 24 16:18:41 2009 192.168.1.105:41494 VERIFY OK: depth=0, /C=cn/ST=AnHui/
O=USTC/OU=infosec/CN=client2/emailAddress=guanlelennon@gmail.com
Sun May 24 16:18:41 2009 192.168.1.105:41494 Data Channel Encrypt: Cipher 'BF-CB
C' initialized with 128 bit key
Sun May 24 16:18:41 2009 192.168.1.105:41494 Data Channel Encrypt: Using 160 bit
message hash 'SHA1' for HMAC authentication
Sun May 24 16:18:41 2009 192.168.1.105:41494 Data Channel Decrypt: Cipher 'BF-CB
C' initialized with 128 bit key
Sun May 24 16:18:41 2009 192.168.1.105:41494 Data Channel Decrypt: Using 160 bit
message hash 'SHA1' for HMAC authentication
Sun May 24 16:18:41 2009 192.168.1.105:41494 Control Channel: TLSv1, cipher TLSv
1/SSLv3 DHE-RSA-AES256-SHA, 1024 bit RSA
Sun May 24 01:18:41 2009 Data Channel Decrypt: Cipher 'BF-CBC' initialized with
128 bit key
Sun May 24 01:18:41 2009 Data Channel Decrypt: Using 160 bit message hash 'SHA1'
for HMAC authentication
Sun May 24 01:18:41 2009 Control Channel: TLSv1, cipher TLSv1/SSLv3 DHE-RSA-AES2
56-SHA, 1024 bit RSA
Sun May 24 01:18:41 2009 [server] Peer Connection Initiated with 192.168.1.104:1
194
Sun May 24 01:18:42 2009 SENT CONTROL [server]: 'PUSH_REQUEST' (status=1)
Sun May 24 01:18:42 2009 PUSH: Received control message: 'PUSH_REPLY,route-gatew
ay 192.168.1.104,ping 10,ping-restart 120,ifconfig 192.168.1.128 255.255.255.0'
Sun May 24 01:18:42 2009 OPTIONS IMPORT: timers and/or timeouts modified
Sun May 24 01:18:42 2009 OPTIONS IMPORT: --ifconfig/up options modified
Sun May 24 01:18:42 2009 OPTIONS IMPORT: route-related options modified
Sun May 24 01:18:42 2009 WARNING: --remote address [192.168.1.104] conflicts wit
h --ifconfig subnet [192.168.1.128, 255.255.255.0] -- local and remote addresses
cannot be inside of the --ifconfig subnet. (silence this warning with --ifconfi
g-nowarn)
Sun May 24 01:18:43 2009 TUN/TAP device tap0 opened
Sun May 24 01:18:43 2009 TUN/TAP TX queue length set to 100
Sun May 24 01:18:43 2009 ifconfig tap0 192.168.1.128 netmask 255.255.255.0 mtu 1
500 broadcast 192.168.1.255
Sun May 24 01:18:43 2009 Initialization Sequence Completed

```

这时，查看虚拟机的网络接口：

```

guanle@guanle-desktop:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:77:4d:54
          inet addr:192.168.1.105  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe77:4d54/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:124 errors:0 dropped:0 overruns:0 frame:0
          TX packets:141 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17591 (17.1 KB)  TX bytes:20445 (19.9 KB)
          Interrupt:16 Base address:0x2024

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1278 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1278 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:63900 (62.4 KB)  TX bytes:63900 (62.4 KB)

tap0      Link encap:Ethernet  HWaddr 00:ff:0d:0e:06:28
          inet addr:192.168.1.128  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::2ff:dff:fe0e:628/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:15 errors:0 dropped:0 overruns:0 frame:0
          TX packets:31 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100

```

可见，比之前多了一个 tap0，这个接口便是虚拟的桥接到 server 端网桥的那个接口。其 IP 为 192.168.1.128。正是 server 端配置文件中指定的 192.168.1.128 到 192.168.1.254 的第一个。

查看 **openvpn-status.log**：

```

OpenVPN CLIENT LIST
Updated, Sun May 24 16:43:58 2009
Common Name,Real Address,Bytes Received,Bytes Sent,Connected Since
client2,192.168.1.105:41494,18095,81434, Sun May 24 16:18:41 2009
ROUTING TABLE
Virtual Address,Common Name,Real Address,Last Ref
00:ff:0d:0e:06:28,client2,192.168.1.105:41494, Sun May 24 16:18:42 2009
GLOBAL STATS
Max bcast/mcast queue length,1
END

```

可见，客户端实际 IP 为 192.168.1.105，与实际一致。

从客户端远程管理 VPN 连接：

```

guanle@guanle-desktop:~$ telnet 192.168.1.104 7505
Trying 192.168.1.104...
Connected to 192.168.1.104.
Escape character is '^]'.
>INFO:OpenVPN Management Interface Version 1 -- type 'help' for more info
help
Management Interface for OpenVPN 2.1_rc12 i686-pc-mingw32 [SSL] [LZ02] [PKCS1
built on May 23 2009
Commands:
auth-retry t           : Auth failure retry mode (none,interact,nointeract).
bytecount n           : Show bytes in/out, update every n secs (0=off).
echo [on|off] [N|all]  : Like log, but only show messages in echo buffer.
exit|quit              : Close management session.
forget-passwords       : Forget passwords entered so far.
help                  : Print this message.
hold [on|off|release]  : Set/show hold flag to on/off state, or
                        release current hold and start tunnel.
kill cn                : Kill the client instance(s) having common name cn.
kill IP:port           : Kill the client instance connecting from IP:port.
log [on|off] [N|all]   : Turn on/off realtime log display
                        + show last N lines or 'all' for entire history.
mute [n]               : Set log mute level to n, or show level if n is absen
needok type action     : Enter confirmation for NEED-OK request of 'type',
                        where action = 'ok' or 'cancel'.
needstr type action    : Enter confirmation for NEED-STR request of 'type',
                        where action is reply string.
net                    : (Windows only) Show network info and routing table.
password type p        : Enter password p for a queried OpenVPN password.
pkcs11-id-count        : Get number of available PKCS#11 identities.
pkcs11-id-get index    : Get PKCS#11 identity at index.
client-auth CID KID    : Authenticate client-id/key-id CID/KID (MULTILINE)

```

由于是桥接法实现的，所有的网都在一个子网中，客户端肯定可以连接到服务器所在子网的任何机器。

### 4.2.3 两种实现方法比较

我们用两种方法实现了 VPN, Bridging 和 Routing。配置上 Routing 应该更简单一点，它提供了点到点的 tunnel，然后通过路由选项，可以实现客户访问服务器所在子网。而 bridging 只要把本地物理连接和虚拟 tap 用网桥连接后，以后的客户连接到 tap 时，他们就已经在一个子网了。他们的优缺点如下：

**Bridging 优点：** 1: 由于在同一子网，不用路由，支持广播

2: 封装以太网帧，故支持的高层协议更广泛

**Bridging 缺点：** 1: 由于在同一子网，网络规模受限

2: 效率相对 routing 较低

**Routing 优点：** 效率更高

**Routing 缺点：** 1: 必须为每个连到其上的子网添加路由选项

2: 不支持广播

3: 只支持 ipv4

## 第 5 章 加密算法分析

### 5.1 密码学简介

#### 5.1.1 对称密钥和非对称密钥学

现代的密码系统都是基于数学和信息理论的。一个简单的密码系统可以用这样一个模型来描述：当发送者 Alice 要向接收者 Bob 发送消息时，Alice 先用加密算法和加密密钥对明文进行加密得到密文。通过公开信道传送给 Bob。Bob 在接收到密文后用相应的解密算法和解密密钥恢复出明文。从数学的角度看，加密和解密操作就是各调用一个数学函数：加密函数的参量对应加密密钥和明文，输出为密文；解密函数的参量对应解密密钥和密文，输出为明文。加密算法、加密密钥、解密算法和解密密钥构成了密码系统。如果有窃听者 Eve 在公开信道窃取密文，虽然他可能知道加密算法和解密算法，但是因为不知道加密密钥或者解密密钥的任意其中一个（或者都不知道），那么他就没法知道最终的明文。因此，密码系统的安全性是依赖于密钥的安全性的。

根据加密密钥和解密密钥是否相同，密码系统可以分为对称密钥系统和非对称密钥系统。加密和解密使用同一个密钥的系统，称为对称密钥系统。常用的对称密钥算法包括 DES, 3DES, RC2, RC4, RC6, AES 等。加密和解密使用的是不同的密钥，称为非对称密钥系统，公钥加密系统即属于非对称密钥系统。对于对称加密而言，需要着重保护的是对称密钥，对于公钥加密而言，需要着重保护的是私钥，因为公钥是公开的，任何人都可能知道，只有接收者的私钥才是保密的。

我们再来看看公钥加密算法。从数学的角度看，设计公钥就是寻找一个陷门单向函数。单向函数  $y=f(x)$  具有这样的特点，对于定义域上任意的  $x$ ，计算  $f(x)$  是容易的；但是反过来，对于值域上的任意  $y$ ，要计算  $f^{-1}(y)$  则是及其困难甚至不可能的。如果给出某些辅助信息，计算  $f^{-1}(y)$  是容易的，则这样的单向函数被称为陷门单向函数，这些辅助信息就是陷门信息。如果把自变量  $x$  看作是明文，陷门单向函数  $f(x)$  看做公开密钥（公钥），那么函数值  $y$  就是密文，陷门信息则是秘密密钥（私钥）。当然，一个实际的公钥加密算法不仅仅是构造一个陷门单向函数这么简单，但陷门单向函数是整个公钥密码系统的基础。公钥加密算法，以及衍生出的数字签名、数字证书技术，都广泛应用于 Internet 通讯和各种现代商业加密系统中。

从上面的分析我们可以看到， $f(x)$  逆运算的计算复杂度越高，密码系统的安

全性就越高。最理想的情况是用一个在不知道陷门信息的条件下无法进行逆运算的陷门函数构建密码系统。但是到目前为止，这样的函数还没有被发现。陷门函数主要有两大类，一类是基于大数质因数分解问题，其中最典型的应用代表就是 RSA 加密算法；另一类是基于离散对数问题，比如椭圆曲线离散对数问题等。这些问题属于“NP 难解问题”但并不是不可解问题。在电子计算机上，求解 NP 问题所进行的计算步数通常与问题规模成指数关系，导致计算量巨大、时间漫长。

### 5.1.2 数字摘要

数字摘要就是通过散列函数对数据作用产生一个值，这个值与原来数据的各个位都相关。散列函数是多对多的映射，但是被映射区域比映射区域一般大得多。通过散列函数，就可以为文件、报文或者其他数据产生指纹，散列函数有以下性质：

- 1) 散列函数可以用于任意大小的数据分组；
- 2) 散列函数的输出是定长的；
- 3) 给定数据  $m$ ，很容易计算指纹  $h$ ；
- 4) 给定指纹  $h$ ，根据  $H(m)=h$  计算数据  $M$  很难；
- 5) 给定  $m$  要找到另一个消息  $m'$  并满足  $H(m)=H(m')$ ，很难

散列算法有很多，如：MD2, MD5, SHA-1 等。MD2 是 RSA 公司的一个算法，产生 128 位的摘要，并且针对低端 8 位微处理器作了优化。MD5 也产生 128 位的摘要，并且是针对 32 位处理器的。SHA-1 是针对高端处理器的，产生 160 位摘要。SSL 中主要应用 SHA-1 算法。

数字摘要的主要用途就是计算数字签名和信息认证码(MAC)。MAC 类似于摘要算法，但是它在计算的时候还要采用一个密钥，因此 MAC 同时依赖于使用的密钥及要计算其 MAC 的信息。SSL 中使用的是一种 HMAC 的变种，而真正的 HMAC 在 TLS 中使用。HMAC 使用嵌套的密钥控制摘要。也就是说，先计算出输入的密钥和数据摘要，然后再使用该摘要值作为另一个密钥控制摘要的输入。

### 5.1.3 数字证书相关技术

#### 1. 数字签名技术

对于重要的文件，为了防止出现对文件的否认、伪造、篡改等问题，传统的方法是在文件上手写签名。但是在计算机系统中无法使用手写签名，取而代之的是数字签名机制。数字签名应该能实现手写签名的作用，其本质特征就是利用签名者的私有信



息产生签名。因此,当被验证时,能通过信任的第三方在任何时候证明只有私有信息的唯一掌握者才能产生此签名。

数字签名可以用私有密钥来实现,也可用公开密钥来实现。采用对称密钥是建立在有信任的中间仲裁机构的基础上,它的完整性的基础是这个中间仲裁机构。这种的安全性不高,而且步骤繁琐。而采用非对称密钥加密法进行数字签名则不受此限制,收发两方之间不需要任何可信赖机构。它的完整性的基础是每个通信者所拥有的私钥。在当前广泛应用的 PKI 中所用的数字签名就是采用非对称密钥加密法。而其中 RSA 结合 MD5 的签名算法最好。它的过程如下:先用 hash 算法将原文压缩为数据摘要,然后用公开密钥算法对摘要进行加密和解密。散列函数的特性决定原文任何变化都会使数据摘要改变。在使用发送者的私钥对这个散列值进行加密,形成签名,附在原文后。发送者所具有的私钥的特殊性决定这个签名是来自于这个发送者,其他人不能假冒:接收者在接到这个附有签名的文件后,使用发送者的公钥进行解密,得到发送者所形成的散列值;然后接收者在用相同的方法对原文计算散列值:比较这两个散列值,如果相同,就表明原文在传送过程中没有被篡改。不相同,则已经被篡改。

## 2. 数字证书技术

数字证书又称为数字标识。它提供了一种在 Internet 上身份验证的方式如果用来标志和证明网络通信双方身份的数字信息文件时,交易双方需要使用数字证书来表明自己的身份,的交易操作。是在网上进行电子商务活动并使用数字证书来进行有关从原理上来讲,就是一个可信的第三方实体对另一个实体的一系列信息进行签名得到一个数字文档,证书用户可以通过这个可信第三方来证明另一实体的身份。它由三部分组成:实体的一系列信息,签名加密算法和一个数字签名。其中实体的信息主要包括三方面的内容:证书所有者的信息,有者的公开密钥和证书颁发机构的信息。

证书中通常不仅包括证书所有者的名字和它的公开密码,还可以包括其它很多信息,这样的证书称为扩展证书,使用扩展证书,用户不仅能得到其他人的公开密钥,可以得到其他的有关信息,如电子邮件地址等。一旦用户拥有了一个证书,就可以通过证书的交换,证实自己的身份。

用户 A 想拥有自己的数字证书时,需要向证书管理中心 CA 发送一个证书请求,其中包括用户的基本信息以及它的公开密钥。证书管理中心 CA 利用 A 发送的请求生成一个特殊的报文,并用它自己的私有密钥对该报文进行签名,然后将报文和签名返回给用户 A,这两部分就构成了 A 的证书。

当 B 想使用 A 的公开密钥时, A 将自己的证书传递给 B, B 用证书管理中心 CA

的公开密钥验证证书中的签名, 如果签名通过了验证, 则 B 接受 A 的公开密钥。标准的证书格式是由国际电信电报联盟制定的 X.509V 3, 该标准已经成为当今网络安全应用如 PKI, SET, SSL, PGP 等的基础。对于证书的管理, 有可能在发生一些特殊的情况时需要撤销证书, 这些特殊情况包括: 证书持有者的工作变动, 怀疑密钥泄漏等。一般通过周期性的发布证书撤销列表(CRL)的方法来公布被撤销的证书。

#### 5.1.4 公钥基础设施 (PKI)

PKI体系结构采用证书管理公钥, 通过第三方的可信机构CA, 把用户的公钥和用户的其他标识信息(如名称、e-mail、身份证号等)捆绑在一起, 在Internet网上验证用户的身份, PKI体系结构把公钥密码和对称密码结合起来, 在Internet网上实现密钥的自动管理, 保证网上数据的机密性、完整性。完整的PKI系统必须具有权威认证机构(CA), 数字证书库、密钥备份及恢复系统、证书作废系统、应用接口(API)等基本构成部分, 构建PKI也将围绕着这五大系统来着手构建。

认证机构 (CA): 即数字证书的申请及签发机关, CA必须具备权威性的特征;

数字证书库: 用于存储已签发的数字证书及公钥, 用户可由此获得所需的其他用户的证书及公钥

密钥备份及恢复系统: 如果用户丢失了用于解密数据的密钥, 则数据将无法被解密, 这将造成合法数据丢失。为避免这种情况, PKI提供备份与恢复密钥的机制。但须注意, 密钥的备份与恢复必须由可信的机构来完成, 并且密钥备份与恢复只能针对解密密钥, 签名私钥为确保其唯一性而不能够作备份。

证书作废系统: 证书作废处理系统是PKI的一个必备的组件。与日常生活中的各种身份证件一样, 证书有效期以内也可能需要作废, 原因可能是密钥介质丢失或用户身份变更等。为实现这一点PKI必须提供作废证书的一系列机制。

应用接口(API): PKI的价值在于使用户能够方便地使用加密、数字签名等安全服务, 因此一个完整的PKI必须提供良好的应用接口系统, 使得各种各样的应用能够以安全、一致、可信的方式与PKI交互, 确保安全网络环境的完整性和易用性。作为一种网络基础设施, PKI以证书为手段保证公开密钥的可靠分发, 但如何使用公开密钥进行安全通信, 如交换会话密钥等, 并不是PKI所能提供的服务, 需要另外的协议机制进行基于公开密钥的安全交互, SSL协议中即包含这样一种以PKI为基础进行安全的会话密钥交换的机制。



## 5.2 SSL 算法的选择

在 SSL 及其后继的 TLS1.0 的规范中,对于加密算法的选择主要集中在 3DES, DES 和 RC4 这三种。在这之中, RC4 的速度最快,而 3DES 慢得多,但是从安全角度来看, 3DES 更安全。在 SSL 应用中用得最多的是 RC4, RC4 是 RSA 数据安全公司开发的一种密码算法。它是一种密钥长度可变的算法,其密钥长度可以在 8-2048 位之间。不管密钥有多长,密钥都被扩展为一张固定尺寸的内部状态表,所以无论使用什么长度的密钥,该算法都运行得一样快。SSL 和 TLS 总是使用密钥长度为 128 位的 RC4, RC4 的速度非常快,一台奔腾 11/400 的机器可获得 45MB/S 的速度。RC4 本质上是一个伪随机数生成器,并且生成算法的输出与数据流进行异或运算。因此,非常重要的一点是,绝对不应当对两个不同的数据流采用同一个 RC4 密钥进行加密。

在 SSL 的加密套件中,所需要的非对称加密算法主要以 RSA 为主,且密钥长度为 1024 位。这样可以保证在可以预见的将来,即使机器的运算速度按照摩尔定律递增,也是很难攻破。在部署 SSL 系统时,根据 SSL 的基本性能法则,非对称算法选择使用 RSA 随着密钥尺寸的增大,公用密钥算法的性能急剧下降。1024 位的 RSA 比 512 位的 RSA 大约慢 4 倍。

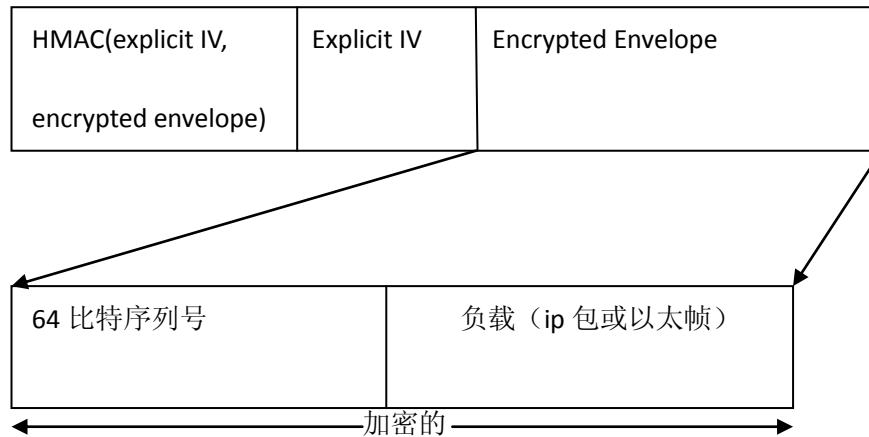
因此选择私用密钥的长度要在安全与握手性能上加以权衡。一般情况下,768 位的密钥对于大多数应用来说已经足够并且要比 1024 位密钥快很多。对称算法上,使用 RC4 来获得最佳性能,使用 3DES 可以获得最好的保密性。摘要算法中 MD5 相对于 SHA-1 只提升了 40% 的性能,大多数情况下,要保证安全的话选用 SHA-1。另外,客户端应当使用会话恢复,服务器应当只在客户端 5 到 10 分钟内就重新连接时使用会话恢复。

## 5.3 OpenVPN 协议安全性分析:

OpenVPN 支持两种认证方式,静态密钥和基于 TSL 的证书认证和密钥交换方式。OpenVPN 对传统 SSL 协议进行了一定改进,传统 SSL 曾受到过缓存区溢出攻击,OpenVPN 支持 TSL-auth 选项,在这种模式下,双方的 TSL 握手过程受到之前共享密钥的保护,通过这个密钥,双方生成 HMAC 的密钥,对握手过程进行认证,从而保护了 TSL 协商的过程<sup>[10]</sup>。

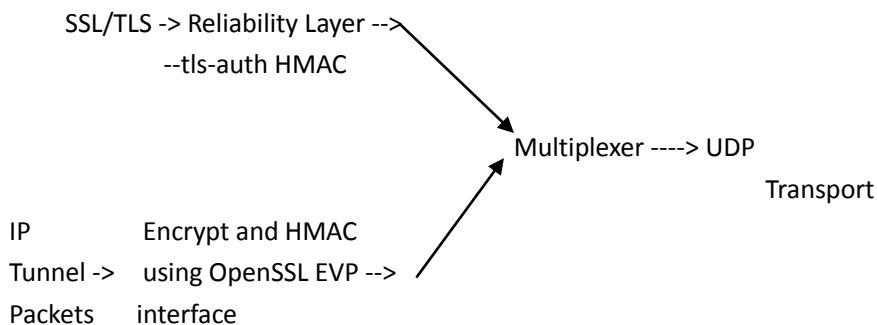
因为 TSL 工作在可靠的传输层之上,所以 OpenVPN 在 UDP 协议上也能提供可靠的传输层。

加密的包形式如下:



和SSL一样，OpenVPN握手期间可以由双方协商加解密算法、密钥长度、签名算法等，而默认情况下，用BlowFish的CBC模式作为加密算法，SHA1作为签名算法。

实际的OpenVPN对SSL会话的数据流和实际加密的隧道数据流通过一个多路选择器。他们都被封装到UDP里面。所以所有加密的IP包都在一个不可靠的隧道里传输。这样提高了系统的效率。对于TSL会话，由于它本身是设计在TCP之上，所以是可靠的。而对于加密的数据流，是在不可靠的传输层传输的。当然，如果加密的数据本身是依赖于可靠的TCP连结的话，TCP本身就会提供可靠性保证。用UDP封装除了效率上的提高，还避免了所谓的可靠曾碰撞问题，举个简单的例子，有两个TCP层，内层的等待时间比外层的短，那么系统处理到内层TCP时，很有可能就会由于超时将包丢掉。这样造成系统性能的严重下降。当然，OpenVPN也提供用TCP封装，只要用选项—Protocol TCP。



这样，SSL层看到的将是一个可靠的传输层，IP包转发看到的是一个不可靠的传输层。可靠性保证的层和认证层是完全独立的，即序列号被加密，在认证层不会被看到。OpenVPN的安全模型可以如下总结：它很像IPSec ESP协议，提供隧道，但摒弃了IPSec中的IKE过程，转而用TSL协议提供会话和协商密钥等。这样，就形成了一个轻量级，移植性好的一种VPN实现。它提供和IPSec一样能力，却省去了复杂臃肿的

IKE过程。SSL相比IPSec VPN也有缺点，比如：IPSec在协议栈的底层分析和处理包，而SSL VPN工作在较上层，所以要解析到这一层需要相对较长的时间，所以，如果认证失败，系统丢包会浪费很多时间。这样，对于DOS攻击，SSL VPN显得比较薄弱，在高负载下，SSL VPN的性能也会下降。

IPSec在操作系统内核实现，而SSL VPN在用户空间实现。现代操作系统遵从环模型，即越外的环上的程序特权越底，IPSec VPN却出现在第0环，即内核空间，违反了这一条原则，SSL VPN却完全工作在用户程序区，属于第3环上。这样，从安全性上讲，SSL VPN胜过IPSec VPN。OpenVPN调用SSL实现RSA或DH握手，实现了IPSec中IKE的功能，之后调用SSL加密库，通过对称加密技术保障隧道的安全，这跟IPSec的实现也是类似的。所以，从功能上讲，OpenVPN的隧道可以让任何数据流通过，就像IPSec一样，没有限制。然而，IPSec VPN的有些模式不能实现NAT穿越，比如若IP包中有AH头，由于NAT的地址转换，接收网关认证IP包失败，从而丢弃包。故只能用ESP头在隧道模式中实现的VPN才能穿越NAT。而SSL VPN不在源地址上做验证，所以不存在此问题。

## 5.4 基于 ECC 加密算法的 SSL VPN

在SSL会话中，常用证书来验证双方身份、协商加密算法、生成密钥等，用基于RSA的非对称加密算法交换密钥，保证密钥的安全性。然而，密码运算，特别是基于RSA的非对称密码算法的运算占用了过多的CPU时间。虽然SSL协议中的会话重用，即会话高速缓存机制，可以减少需要从头建立的连接数，但是对于新的会话的连接，依旧比较耗费系统的时间和资源。因此，在这部分针对利用RSA算法进行加密的VPN系统存在的不足，提出将ECC（Elliptic Curve Cryptography，椭圆曲线密码）应用于SSL安全握手过程。这样，可以大大提高运算效率，并在相同的安全强度下减少密钥的长度，其运算量较小、复杂度也随之降低。

### 5.4.1 ECC 加密体制

#### 1. ECC 简介

ECC体制的安全性基于有限域上椭圆曲线离散对数问题(ECDLP)的难解性。ECDLP难解性是指：对于曲线上给定的离散点P和Q，难以找到整数I，使得 $IP=a$ 。设P为公钥，Q为私钥，其安全性就表现为知道P无法推导Q。对于有限群上的a和b，若存在正整数n，使得 $a^n = b$ ，求解 $n = \log_a b$ 的问题称为有限群上离散对数问题(LDP)；

而对椭圆曲线上离散点 $P$ 和 $Q$ ，求解，使得 $IP=Q$ 称为椭圆曲线离散对数问题，ECDLP 优于LDP。

ECC是一种能适应未来通信技术和信息安全技术发展的新型密码体制，在运算速度和存储空间方面占有很大的优势，具有安全性高、密钥量小、灵活性好的特点，具有较好的抗攻击性，由于服务器端的CPU处理能力和存储空间都有限，所以ECC相对于以往基于RSA加密算法的握手过程具有绝对的优势， 目前已成为公钥密码体制中的研究热点。

## 2. 椭圆曲线公钥密码体制的加解密方案：

公钥加密方案可用来提供机密性，主要应用是在少量数据的加密上，如信用卡号码。大批量的数据加密使用运算速度更快的对称密钥算法来完成。下面给出了加解密方案：

加密过程，当用户A发送信息 $M$ 给用户B时，用户A执行下列步骤：

- (1)查找B的公钥 $Q$ ；
- (2)将数据 $M$  表示成一个 $E(F_q)$ 上的一个点 $m$ ；
- (3)在区间 $[1, n-1]$ 内选取一个随机整数 $k$ ；
- (4)计算点  $(X_1, Y_1)=kP$ ；
- (5)计算点 $(X_2, Y_2)=k*Q$ ，如是 $X_2=0$ ，则返回到第(3)步；
- (6)计算 $c=mX_2$ ；
- (7)传送加密数据 $(X_1, Y_1, c)$ 给B.

解密过程，当用户B解密从用户A收到的密文 $(X_1, Y_1, c)$ 时，执行下列步骤：

- (1)使用他的私钥 $d$ ，计算点 $(X_2, Y_2)=d(X_1, Y_1)$ ；
- (2)通过计算 $m=-cX_2^{-1}$ ，恢复出数据 $m$ ，

在上述过程中， $Q=d*P$ 是公开的，如果除A、B外的第三者能解椭圆曲线上的离散对数问题，就能从 $d*P$ 中求出 $d$ ，从而解密信息。

### 5.4.2 ECC 加密体制在 SSL 安全均衡握手方面的优势

ECC算法的计算量小并且处理速度快，ECC算法的存储空间占用小，ECC的密钥尺寸和系统参数与RSA相比要小得多，160 bit ECC与1024bitRSA具有相同的安全强度，210 bit ECC则与2048 bit RSA具有相同的安全强度，意味着它所占的存储空间要小得多，同时ECC算法的带宽要求低。此外，ECC算法的灵活性要高于RSA算法，可以通过改变参数设置获得不同的曲线，具有丰富的群结构和多选择性。有下面的分析可见，

采用加速点积运算优化后的ECC算法技术运用到了SSLVPN的安全握手技术中，使SSL服务器的CPU可以提高将近60%来处理应用程序逻辑。克服了以往用户将大部分时间浪费在等待CPU处理加解密过程中的弊端，提高了工作效率。

### 5.4.3 ECC 公钥算法的优化

针对点积运算的优化。公钥产生和加密解密算法中需要大量的点积运算，即计算  $nP=P+P+\dots+P(n\text{个}P)$

在此采用的点积优化算法如下：

(1) 将 $n$ 表示成二进制数形式，即 $n=(n_k n_{k-1} \dots n_m \dots n_1)$

式中： $n_m = 0$ 或 $1$ ； $k = \lfloor \log_2 n \rfloor + 1$

(2) 去掉 $(n_k n_{k-1} \dots n_m \dots n_1)$ 的最高位 $n_k$ ，得 $(n_{k-1} \dots n_m \dots n_1)$ 。

(3) 按照 $(n_k n_{k-1} \dots n_m \dots n_1)$ 从高位到低位次序，当 $n_m=0$ ，计算 $2P$ ；当 $n_m=1$ ，计算 $2P+P$ ，并将结果作为下次计算的初值，即

$2P \Rightarrow P$ 或 $2P+P \Rightarrow P$

采用常规方法，需进行 $n$ 次点加运算；在本算法中，平均只须 $3/2 \lfloor \log_2 n \rfloor$ 次运算，最多需要 $2 \lfloor \log_2 n \rfloor$ 次运算。在明文到椭圆曲线映射过程中，需要判别一个数是否为模 $p$ 下的平方剩余，即平方剩余判定。目前现有的平方剩余判定算法仅仅简单地根据平方剩余定义来判别，涉及大数的平方运算和取模运算，算法效率非常低。针对这种情况，本文采用了一种快速平方剩余判定算法。

设明文分段 $m$ 映射到点 $P_m(x, y)$ 上，使其满足

$$\begin{cases} 256m \leq x \leq 256(m+1) \\ P_m(x, y) \in F_p \end{cases}$$

下面要解决的问题是在 $256$ 和 $256(256+1)$ 之间给定一个 $X$ ，判定 $A=x^3+ax+b$ 是否是模 $P$ 下的平方剩余。即判定 $(A/q)$ 是否为 $1$ 。改进的快速平方剩余判定算法如下：

(1) 设 $J$ 为平方剩余判定变量，初始时 $J=1$ 。

(2) 如果 $A$ 为偶数，则根据定理， $(A/q)$ 可分解为： $(A/q)=(2/p)((A/2)/p)$  根据定理，计算 $(2/p)$ ，然后执行 $J(2/p) \Rightarrow J$ ， $A/2 \Rightarrow A$

如果 $A$ 为奇素数，则根据定理得

$$(A/P)(P/A)=(A/P)((P \bmod A)/P)=(-1)^{\left(\frac{A+1}{2}\right)((P-1)/2)}$$

$$(A/P)=(-1)^{\left(\frac{A+1}{2}\right)((P-1)/2)}((P \bmod A)/A)$$

这样，对 $(A/P)$ 的判定等价于对 $((P \bmod A)/A)$ 的判定，即执行如下操作：

$$J(-1)^{\left(\frac{A+1}{2}\right)((P-1)/2)} \Rightarrow J$$

$$A \Rightarrow q$$

$$P \bmod A \Rightarrow A$$

$$q \Rightarrow p$$

如果A为奇数但不为素数，可将A分解为 $\prod A_i$ ；其中 $A_i$ 为奇素数，根据定理得：

$$(A/P) = (A_1/P) (A_2/P) \dots (A_i/P) \dots (A_n/P)。再分别计算(A_i/P)。$$

(3) 当 $A \neq 1$ 时，返回(2)；否则，算法结束。此时根据J值判定 $x^3 + ax + b$ 是否是模P下的平方剩余；若 $J=1$ ，则为平方剩余，若 $J \neq -1$ ，则为非平方剩余。

#### 5.4.4 基于优化 ECC 加密算法的 SSL 安全均衡握手过程<sup>[13]</sup>

首先设计适合于VPN系统的安全协议必须考虑的问题：服务器端的低计算能力和计算资源。

本例选用适用于VPN系统，简化的X509标准的证书形式，用户A的证书：

$CertA = \{IDA, pubkeyA, periodA, others, signcA, [Hash(IDA, pubkeyA, periodA, others)]\}$

首先CA 选择定义在有限域上的椭圆曲线 $E(F_m)$

$$x^3y^2 = x^3 + ax + b(a, b \in F_m, m \text{ 是一大素数})$$

$$\Delta = 27b_2 + 4a_3 \neq 0$$

$E$ 上的点构成循环群 $P \in E(F_m)$  是CA选择的公开的基点， $L = \text{ord}(p)$  是公开基点的阶， $L$ 至少为160位的椭圆曲线的阶至少是有40位以上的大素数因子，这样选择的椭圆曲线能保证其上的离散对数难解性。

CA选择一个数 $S_{ca} \in \{1, 2, \dots, L-1\}$ 作为其私钥， $(S_{ca}, L-1) = L$ ，计算 $P_{ca} = S_{ca}$ ，并将 $P_{ca}$  CA作为公钥并公开，这样就可以对服务器端和客户端的证书进行签名。

客户选择一个随机数 $S_{vn} \in \{1, 2, \dots, L-1\}$ 作为其私钥， $(S_{vn}, 1) = 1$ ，计算 $P_{vn} = S_{vn}$  作为其公钥，并提交给CA为其颁发证书：

$$Cert_{vn} = \{ID_{vn}, P_{vn}, period_{vn}, others, SIGN_{vn}\}$$

其中 $SIGN_{vn}$ 是CA的签名。用户入网时，为其生成私钥 $S_{ms}$ 和公钥 $P_{ms}$ ，同时根据用户提交的有效信息生成证书：

$$Cert_{ms} = \{ID_{ms}, P_{ms}, period_{ms}, others, SIGN_{ms}\}$$

客户接收到服务器端证书后，验证证书的合法性。利用证书中的信息 $ID_{vn}, P_{vn}, period_{vn}$ 和 $others$ 生成：

$$m_{vn} = \text{Hash}(ID_{vn}, P_{vn}, period_{vn}, others)$$

然后验证:  $m_{vn}=Y_{vn}R_{vn}+r_{vn}XP_{ca}$  是否成立, 如果成立, 则此签名有效, 证书合法, 在认证之前, 服务器端可以先进行预计算产生随机数:  $k, k \in (1,2,...,l)$

计算  $kP, k, P_{vn}$ , 以及:  $Q=(S_{ms}+k) P_{vn}=(qMSY)$

$qMSx$ 和 $qMSY$ 分别是 $Q$  点的坐标。保存 $k, kP, P_{vn}, qMSX$ . 在这里一定要注意, 每次进行身份认证的时候都必须重新产生随机数 $k$ .

## 5.5 量子密码在 VPN 中的应用

随着量子密码的发展, 凭借其无条件安全性的特点, 近年来得到了广泛发展。美国、欧洲、日本近年来兴起的量子通信领域研究, 其中一个方向就是, 如何把具有无条件安全的量子密钥分配协议融入现实的各种网络中, 其中一些组网方案已经在高等院校和国防部门实施, 并已经证明了量子密钥分配协议确实可以提供接近于保证传送信息无条件安全的服务。

### 5.5.1 量子密码

量子密码学的鼻祖是美国人Wiesner。1976年, 美国哥伦比亚大学的Wiesner最早提出将量子力学与密码术相结合, 并撰写了一篇“共轭编码”的论文。也许是这种思想在当时看来过于离奇, 甚至没有一家科学杂志愿意发表他的研究成果, 直到1983年, 他才有机会将论文发表在一家刊物上。幸运的是, IBM公司的Bennett和加拿大蒙特利尔大学的Brassard两人对此进行了深入研究, 于1984年在一次IEEE会议中提出了第一个量子密钥分配协议, 阐述了如何在一个不安全的公开信道上利用量子态在通讯双方之间安全地交换密钥, 即著名的“BB84”协议<sup>[14]</sup>, 这是一个广泛应用于各种量子密码系统的协议, 其安全性得到了理论证明。用来传输密钥的载体通常是单个光子, 可以用其偏振状态(极化方向)、相位或者频率等物理量来携带信息。除了一个量子信道用于量子信号的传输外, 还需要一个经典信道做相关的经典通信, 经典信道可以不用保密, 也就是说通信信息是公开的。而中途的窃听者能很方便的窃听经典信道, 但是不能随意对数据进行篡改(这个假定是合理的, 可以通过消息认证等手段来保证)。

量子密码的安全性是由量子力学的几个基本规律决定的。第一是海森堡不确定性原理, 也叫做测不准原理<sup>[15]</sup>, 它告诉我们两个非对易的物理量是不可能同时被精确测量的。在经典物理中粒子的坐标和动量是可以同时取确定值的, 但在量子力学里面它们就不行, 当其中一个完全确定时, 另外一个就完全不确定。另外还有时间和能量的不确定性等。不确定关系是粒子波动性的必然结果, 是微观粒子的固有性质, 与测

量仪器精度无关。第二是测量塌缩原理<sup>[16]</sup>，即对量子态进行测量会不可避免的使该量子态塌缩到某一个本征态上。除非被测量的量子态正好是某个本征态，否则测量前后的量子态是不同的，这意味着对量子态进行测量都会留下痕迹。测量即破坏的道理在经典物理里面也是很奇怪的事情，但在微观世界却是很普遍。第三就是量子不可克隆定理<sup>[17]</sup>，即一个未知的量子态是无法被精确克隆的。这个定理虽然颠覆了我们平时数据可以任意拷贝的直观印象，但是却真实的存在于量子世界。我们可以这样来理解它，假设有一个未知的量子态，我们能够完全的拷贝它，这就意味着能得到足够多的完全拷贝，可以任意精确的测出它的任何两个不对易力学量（不断的重复测量），但这个与海森堡不确定关系矛盾，所以精确的拷贝未知量子态是不可能的，量子不可克隆定理也是量子力学推导的必然结果。后面我们将详细介绍BB84协议是如何利用这三个原理来实现量子密钥分配安全性的。

量子密码术（Quantum Cryptography）解决的也是对称密钥系统中密钥分配的难题，因此叫做量子密钥分配（Quantum Key Distribution, QKD）可能更准确一点。当然，该技术也可以用于传递其他密钥系统的密钥甚至直接传递明文，后面我们将可以看到，因为通信双方原始得到的是一串事先并不知道随机数，而且有可能其中部分信息被窃听者得到（尽管可以保证检测出并剔除窃听者的信息），因此用来传递并不含任何信息的密钥更合适<sup>[19]</sup>。

### 5.5.2 基于 QKD 和 IPSec 的 VPN 体系结构

如第二章所述，SSL VPN 与 IPSec VPN 加密的实现方式以及加密算法大体是相同的，加密方式本质上没区别，并且 IPSec VPN 技术发展的比较成熟；因此，在这一部分，将重点分析 QKD 在 IPSec VPN 中的应用，QKD 在 SSL VPN 中的应用可以类比 QKD 在 IPSec VPN 中的应用，只需在实施上稍作修改即可。

图5-1是基于QKD和IPSec的新型TCP/IP 体系结构，图中第1列是传统的TCP/IP体系结构，第2列和第3列则细化了体系结构中的网络层和网络接入层。由于以太网技术是当今最重要的局域网组网技术，因此把以太网技术作为研究对象，在细化和实用TCP/IP体系结构中的网络层和网络接入层时，采用了IEEE标准802.2逻辑链路控制和IEEE标准802.3带有冲突检测的载波侦听多路存取CSMA作为TCP/IP体系结构的底层协议，而QKD与IEEE标准802.2逻辑链路控制同处网络接入层，它为上层提供具有无条件安全的加密服务。IPSec安全协议所提供的服务是在网络层上提供的，它为上层协议提供诸如网络单元的访问控制、数据源认证和有限的数据流保密。同时我们对



QKD再进行子层的划分，编码层主要是产生原始密钥流；由于噪声的影响、窃听的存在以及实际系统中各种因素的影响，接收方可能没有收到，或者收到一些不符合要求的量子密钥比特，因此有必要通过筛选层筛选掉这些没有收到或者不符合要求的量子密钥比特；同时为了保证通信双方各自保存的量子密钥比特的完整性和一致性，我们通过检错和纠错层对筛选后的量子密钥比特进行检错和纠错；实际系统中可能存在着窃听者拥有通信双方一部分量子密钥比特的情况，为了使窃听者所拥有的这部分量子密钥比特无效，增加了保密放大这一层；最后为了防止中间人的攻击，再增加了认证这一层。

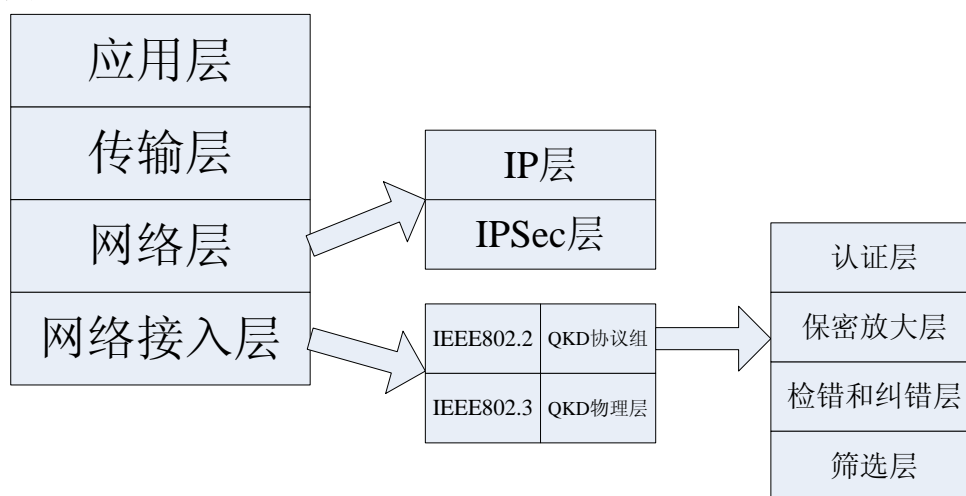


图5-1 基于量子密钥分配协议和IPSec安全协议的新型TCP/IP体系结构

应该指出，增加的4个子层都提供了一些可以完成相似功能的技术，而且这些技术已经实用化或者在实验室里得到了很好的仿真应用。对于这些待选的技术，如何选择和操作是一个难题，为了解决这个难题，提出了QKD安全关联的概念，目的是让虚拟专用网络的通信双方在数据传输之前建立具体的通信策略，协商具体的通信技术，以达到组建一个可控安全服务等级的虚拟专用网络要求。同时为了保证全球互联网上的互操作性，相应规定了一组默认的技术，这样做是为了建立全球无条件安全的互联网量子网络。

当数据流进入IPSec安全协议层时，通过IPSec安全协议的安全策略来决定该数据流是否接受下层的量子密钥分配协议的保护，如果不接受的话，那就进行常规的IPSec处理，如果IPSec安全协议的安全策略决定数据流要接受下层的量子密钥分配协议的保护，那就进入了量子密钥分配协议的处理，而这种做法很容易实现的，只需要在IPSec安全协议中的安全关联数据库中增加一个域，称之为可选的QKD域，就可以让VPN的通信双方决定是否对数据流采用QKD处理了。采用安全策略来决定是否使用

量子密钥分配协议，目的是使VPN的通信双方有一种基于数据流安全等级区分的概念，如果用户认为自己现在所传输的数据流需要较高安全等级的服务，我们就采用量子密钥分配协议来对该数据流进行保护，如果用户认为自己现在所传的数据流不需要较高等级的安全服务，那就仅仅采用IPSec安全协议来对数据流进行保护。

当通信方一旦确定要对数据流进行QKD的处理，就进入到我们所提出的QKD安全关联的概念中，在这个安全关联中建立了一个安全关联数据库，库中含有很多安全条目，每个条目由以下几个域所组成：

- (1) 筛选域：该域供VPN的通信双方协商具体的筛选技术和策略；
- (2) 检错和纠错域：该域供VPN的通信双方协商具体的检错和纠错技术；
- (3) 保密增强域：该域供VPN的通信双方协商具体的保密增强技术和策略；

(4) 安全关联生存期域该域：中包含了一个时间间隔，外加一个当该安全关联过期时是否被替代还是被终止的标志。安全关联的生存期用两种参数形式来表示，一种是时间间隔的形式，另一种是所生成的最终用于加密的密钥数。如果这两种参数都使用了，则以先过期的为准，即最先过期的参数优先。

同时在条目中我们预留了一些域，以适应将来量QKD的发展。当有一个数据包要进行QKD处理时，通过从上层数据包头域中解析出来的域信息来搜索QKD安全关联数据库，如果找到一个匹配的条目，就处理该数据包，如果未找到匹配的条目，则丢弃该数据包。具体选择哪一QKD组是依赖于具体通信环境的；然而有时候为了保证QKD网络的互操作性，规定了一组默认的QKD组。

### 5.5.3 QKD 在 IPSec VPN 中的应用过程

采用QKD协议的IPSec工作流程如下图所示。VPN作为网关服务器，内部的通信为安全通信，外部的通信一般认为是不安全的，需要进行保密通信，因此采用了IPSec协议。OPC为QKD协议处理计算机，负责QKD的协商和密钥生成，负责给运行IPSec协议的VPN服务器提供密钥。

IPSec协议包括了三个部分：

1. **AH**（Authentication Header），认证头标<sup>[RFC 2402]</sup>，规定了对IP分组进行认证的协议框架，以防止在传输过程中IP分组内的数据被篡改。此外它还提供防御重发攻击（Replay Attack）的功能。具体认证协议并不由AH规定，但最低要求安装HMAC-MD5（Hash for Message Authentication Code-Message Digest 5）或HMAC-SHA（HMAC-Secure Hash Algorithm）认证算法。

λ **ESP** (Encapsulating Security Payload), 安全封装数据<sup>[RFC 2406]</sup>, 规定了IP分组加密协议的框架。同AH一样, 它也没有规定具体加密算法, 但最低要求DES-CBC (Data Encryption Standard-Cipher Block Chaining)。

λ **密钥管理协议**, 规定了通信双方的密钥交换和管理, 包括ISAKMP (Internet Security Association and Key Management Protocol: Internet 安全关联与密钥管理协议<sup>[RFC 2408]</sup>)、IKE (Internet Key Exchange: Internet 密钥交换<sup>[RFC 2409]</sup>)及Oakley<sup>[RFC 2412]</sup>。

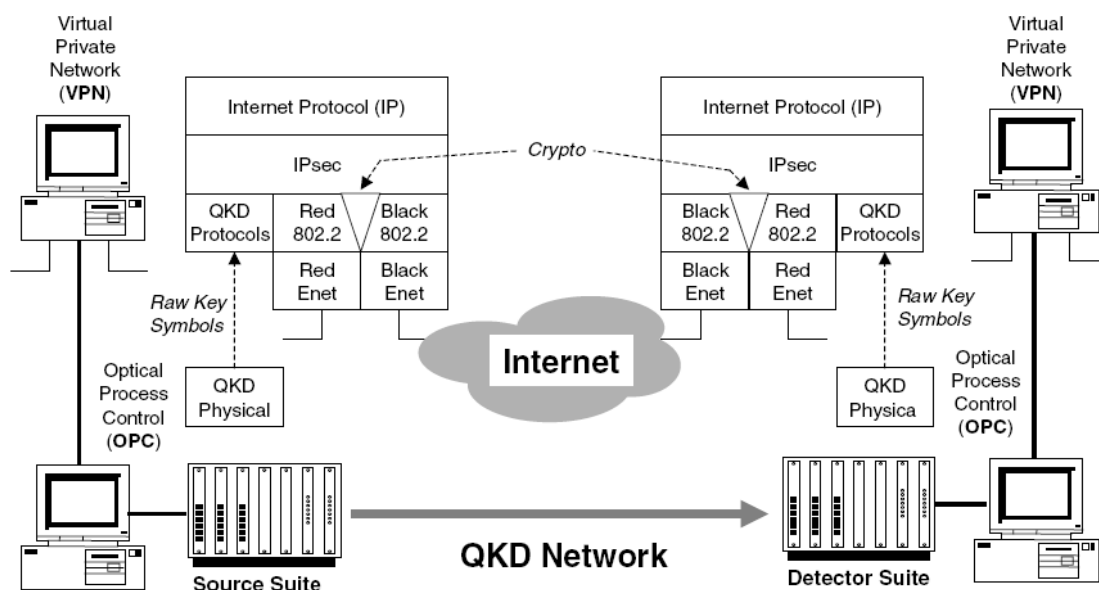


图5-2 基于QKD 技术的IPSec, 图片摘自文献<sup>[18]</sup>

当利用IPSec进行通信时, 采用哪种认证算法、加密算法或是采用哪个密钥都要事先决定, 并达成一致, 这个一致的安全策略叫做安全关联SA (Security Association)。需要注意的是一个SA只针对单向通信, 因此如果要双工通信必须建立两个SA。而SA的建立过程就由ISAKMP来负责, ISAKMP是一个建立和管理安全关联SA的总体框架。它定义了默认的交换类型、通用的载荷格式、通信实体间的身份鉴别机制以及安全关联的管理等内容。它不要求使用哪个具体的密钥生成方案, 也不要求使用哪一个具体的DOI, 但ISAKMP 给出了通用的几种密钥生成方案, 以及使用DOI的建议。

IKE通过两阶段的协商来完成SA的建立。在第一阶段, 由IKE交换的发起方发起一个主模式交换(Main Mode), 交换的结果是建立一个名为ISAKMP SA的安全关联。这个安全关联的作用是保护为安全协议协商SA的后续通信。主模式将SA的建立和对端身份的鉴别以及密钥协商结合起来, 能够抵抗中间人攻击MITM。为了给ISAKMP SA协商提供一个更快捷的方式, IKE还提供了另一种模式: 积极模式 (Aggressive Mode), 这种模式使得协商更为快捷, 但抵抗攻击的能力较差, 也不能提供身份保

护。第二阶段可由通信的任何一方发起一个快速模式(Quick Mode)的消息交换序列,完成用于保护通信数据的IPSec SA的协商。

发起者Alice 响应者Bob: 1) HDR, SA; => 2) HDR, SA; => 3) HDR, KE, Ni; => 4) HDR, KE, Nr; => 5) HDR\*, IDi, CERT, SIG-I; => 6) HDR\*, IDir, CERT, SIG-R.

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
~          ISAKMP Header with XCHG of Main Mode,          ~
~          and Next Payload of ISA_SA                      ~
+-----+-----+-----+-----+-----+-----+-----+-----+
!          0          !   RESERVED   !          Payload Length   !
+-----+-----+-----+-----+-----+-----+-----+-----+
!          Domain of Interpretation          !
+-----+-----+-----+-----+-----+-----+-----+-----+
!          Situation          !
+-----+-----+-----+-----+-----+-----+-----+-----+
!          0          !   RESERVED   !          Payload Length   !
+-----+-----+-----+-----+-----+-----+-----+-----+
! Proposal #1 ! PROTO_ISAKMP ! SPI size = 0 | # Transforms !
+-----+-----+-----+-----+-----+-----+-----+-----+
!   ISA_TRANS   !   RESERVED   !          Payload Length   !
+-----+-----+-----+-----+-----+-----+-----+-----+
! Transform #1 ! KEY_OAKLEY   |          RESERVED2          !
+-----+-----+-----+-----+-----+-----+-----+-----+
~          preferred SA attributes          ~
+-----+-----+-----+-----+-----+-----+-----+-----+
!          0          !   RESERVED   !          Payload Length   !
+-----+-----+-----+-----+-----+-----+-----+-----+
! Transform #2 ! KEY_OAKLEY   |          RESERVED2          !
+-----+-----+-----+-----+-----+-----+-----+-----+
~          alternate SA attributes          ~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

图5-3 ISAKMP可能的数据格式, 摘自RFC

两阶段协商有如下的好处:

- (1) 一个ISAKMP SA可以用于保护多个第二阶段的协商。并且, 通过HDR头中MID字段的使用, 可以同时并发地协商多个IPSec SA。在大规模的网络通信环境中, 这显然有利于提高协商效率。
- (2) 两阶段协商的方式降低了某些敏感信息的使用频率。

关于ISAKMP SA的建立, 我们这里只介绍主模式。在IKE主模式中, 完成了六个消息的交换。这六个消息的载荷组成因IKE使用的鉴别方法的不同而有所差别。ISAKMP 规定了四种身份鉴别方式, 分别为共享密钥、数字签名、公钥加密以及增强的公钥加密身份鉴别方案。HDR是ISAKMP 的通用头, 每个ISAKMP 消息都以它开头。第一、第二消息协商安全属性, 包括: 加密算法; 散列算法; 鉴别方式; Oakley群。第三、第四两个消息完成Diffie-Hellman交换, 为通信的双方生成一个共识的秘密。第五、第六两个消息主要用于对已交换的IKE 消息进行一致性检查, 以及对对端进行身份验证。到此, 一个ISAKMP SA已成功地建立了。在整个建立过程中, 依据双方的安全策略、环境配置, 形成了用于保护通信

双方后续IPSec SA协商的各个安全参数：鉴别方式、加密算法、散列算法、加密密钥、鉴别密钥、Oakley群、ISAKMP SA生存期、ISAKMP SA的标识符等等。

整个协商过程受到了完整性保护，身份等敏感信息受到了机密性保护。

**IPSec SA的建立：**通过快速模式建立IPSec SA。该模式通过交换三个消息完成IPSec SA的协商。其消息序列如图所示。在这三个消息中，发起者通过第一个消息，按本地的策略要求，用SA载荷，提议一种或多种保护数据信息的安全协议（如ESP或AH），并给出其相应的变换（即安全协议的安全参数）。应答者依据本地策略，从提议的一种或多种安全协议中选择一种，并从而为选中的安全协议给出的一种或多种保护套件（即变换）中选一种，形成选择后的SA，应答给发起者。这个选择后得到的SA，将用于保护数据通信。

到此，一个IPSec SA建立完成。整个建立过程受到ISAKMP SA的机密性、完整性保护。而且，通过HDR中的cookie字段以及瞬时载荷，使整个建立过程能一定程度上地抵抗重放和拒绝服务攻击。

IKE中基于Diffie-Hellman交换生成密钥，它是基于计算安全性的，并不是一个绝对安全的算法。我们用QKD技术代替DH密钥交换过程，提高了密钥分配的安全性，同时也减少了对主机计算量的需求。更新后的过程如下所示：

发起者Alice 响应者Bob

- 1) HDR, SA
- 2) HDR, SA
- 3) HDR, QKD1
- 4) HDR, QKD1(QKD process)
- 5) HDR\*, IDii, CERT, SIG-I**
- 6) HDR\*, IDir, CERT, SIG-R

第一、第二个消息中添加QKD密钥交换的选项。其中第三、第四个消息为QKD参数的协商。

至此可见，QKD技术可以很好的应用到IPSec VPN中，并保证其绝对安全性。对于QKD在SSL VPN中的应用，可以以同样的层次机构，同样的层次管理方法；并与IPSec一样，以同样的管理、协商、实现方式将QKD应用到SSL VPN。

#### 5.5.4 QKD 在 VPN 中的具体实施

##### 1. QKD在组建内联网虚拟专用网络中的实施

内联网VPN是一个组织机构的总部或中心网络与跨地域的分支机构以及各分支机构在公共通信基础设施上采用VPN技术而构成的组织机构内部的VPN，图5-4是典型的系统结构，我们画出了一个组织机构的总部和该组织机构的一个分支机构采用QKD协议和IPSec安全协议的VPN网络结构，其中QKD协议服务器不仅完成物理层由硬件完成的量子编码的功能，还完成软硬件皆可实现的筛选、检错和纠错、保密增强和认证等功能。

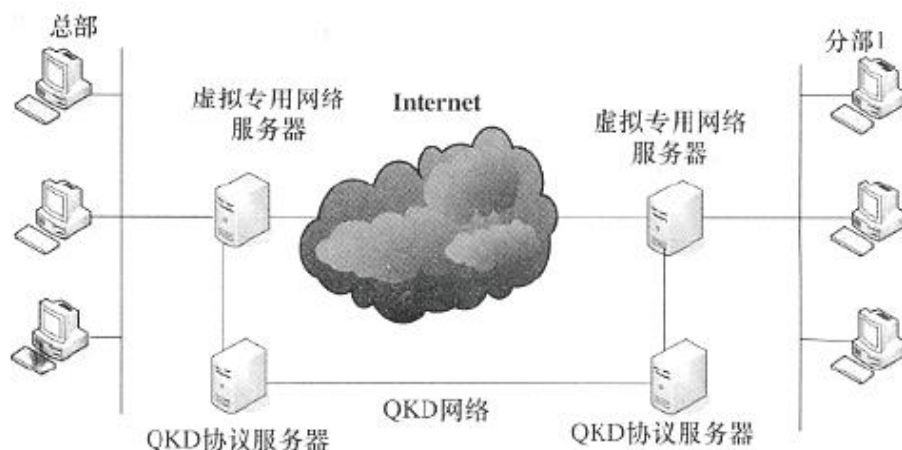


图5-4 基于量QKD的内联网VPN

在确定使用QKD协议的条件下其主要的处理过程如下：

- (1) 发送端生成原始密钥且经过QKD协议4 层子层处理后，经QKD网络传输到接收端，经双方协商，生成最终用于加密的密钥；
- (2) 这些密钥在发送端作为IPSec 安全协议中的加密算法AES或者3DES 的密钥输入，对虚拟专用网通信双方的数据流进行加密；
- (3) 加密后的数据流在公共基础设施网上传输，到达接收端；
- (4) 接收端采用协商的密钥解密这些数据流，生成原始数据流。

## 2. QKD在组建外联网VPN中的实施

外联网VPN是不同组织或企业在公共通信基础网络上通过VPN技术而构成的一个VPN，图5-5是其典型的网络结构图，与内联网VPN的主要区别在于：在安全网关之后增加了认证服务器这一模块，把公司1 的安全网关配置成能够接收从公司2发送过来的数据包。同时基于全球互操作性的需要，在QKD安全关联中不再进行具体的通信策略的协商，而是直接利用该安全关联所规定的一组默认的QKD协议组来对通信流进行处理，其余处理过程同上。

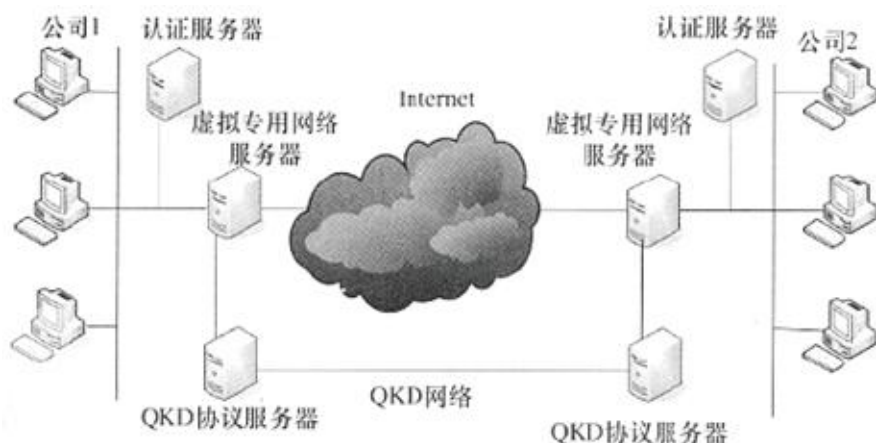


图5-5 基于量子密钥分配协议和IPSec安全协议的外联网虚拟专用网络

### 3. QKD在组建远程访问VPN中的实施

图5-6是组建远程访问VPN结构，与组建内联和外联VPN最重要的区别在于远程访问客户端一侧没有安全网关，其IP地址是动态的，在这种情况下远程客户端的认证就不同于外联网的认证方式了，一般是把安全网关配置为只接收持有合法数字证书的用户发送过来的数据包，而我们组建基于QKD协议的远程访问VPN的真正难题在于为每个远程客户端配置QKD协议的非现实性，因此我们提出把QKD协议的实现转交给ISP提供商解决，让ISP提供商提供远程用户的远程访问和QKD协议服务，同样基于互操作性的需要，在QKD协议安全关联中也不再进行具体的通信策略的协商，而是直接利用该安全关联所规定的一组默认的QKD协议组来对数据流进行处理。

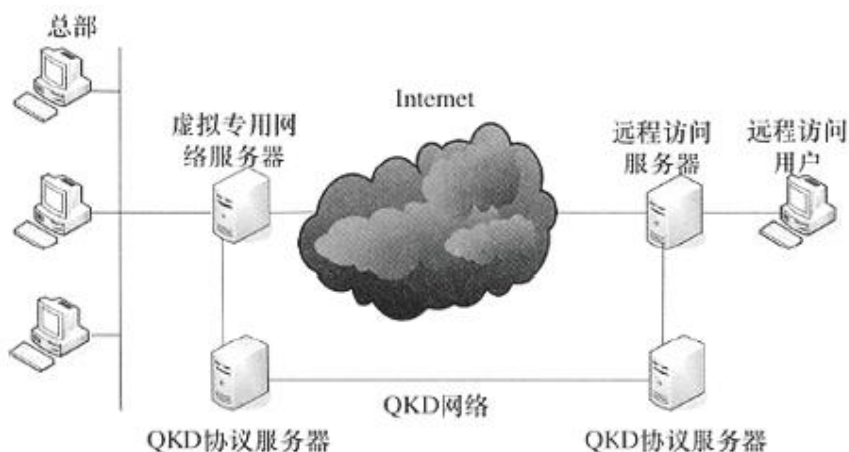


图5-6 基于量子密钥分配协议和IPSec安全协议的远程访问虚拟专用网络

#### 5.5.5 QKD 应用于 VPN 的总结

充分利用量子密码的无条件安全性组建一个安全性相对较高的VPN，真正解决VPN的安全问题或者说真正解决IP的通信安全问题，是各国网络安全专家密码学专

家始终研究的热点,美国、欧洲、日本近年来兴起的量子通信领域研究其中一个方向就是如何把具有无条件安全的量子密钥分配协议融入现实的各种网络中,取得了很好的成果,其中一些组网方案已经在高等院校和国防部门实施,并已经证明了 QKD 确实可以提供接近于保证传送信息无条件安全的服务。本部分正是基于这种思想描绘了融入 QKD 和 IPSec 的 TCP/IP 体系结构模型,并类比将其应用于 SSL VPN 中,同时也将这个结构模型应用在目前倍受关注的 VPN 中。从分析的过程和当前国内外 QKD 技术的发展来看,提出的基于 QKD 的 TCP/IP 体系结构是可以在 VPN 中实施发挥其良好的无条件安全作用的。

## 5.6 全文展望

在前面五章里,我们从SSL和VPN基本技术出发,对SSL VPN的技术特点,SSL VPN系统的实现,以及其中的加密算法进行了全面的总结。SSL VPN研究的理论和实践正处于一个活跃的时期,本文对这一领域的探索和研究还局限于一小部分,还有大量的内容有待进一步发掘。作者认为对以下内容的进一步研究是有意义的:

1. 提供更强有力的安全保障。对于SSL VPN系统,还可以通过使用专用身份识别系统来提高系统的安全性,如IC卡身份识别、指纹识别等等;通过身份识别系统与认证及访问控制机制的集成,可大幅降低远程用户接入的风险性。现有系统只能尽力提供安全的数据通道而不能证明连接的安全性,而且没有专用的客户端安全检测工具。因此,应进一步研究如何能提高系统的安全性。
2. 系统效率,由于是集中系统,SSL 网关决定整个网络的吞吐量。如果SSL 网关效率跟不上,远程接入就会比实际的Internet 接入带宽低很多。对于SSL 网关性能的测试改进需要进一步的研究。
3. 随着量子密码的发展,凭借其无条件安全性的特点,近年来得到了广泛发展。美国、欧洲、日本近年来兴起的量子通信领域研究,其中一个方向就是,如何把具有无条件安全的QKD协议融入现实的各种网络中,其中一些组网方案已经在高等院校和国防部门实施,并已经证明了量子密钥分配协议确实可以提供接近于保证传送信息无条件安全的服务。因此,如何将QKD技术更好的应用到VPN当中去,并与SSL协议结合起来组建一个安全性相对较高的VPN,真正解决VPN的安全问题或者说真正解决IP的通信安全问题,是一个非常值得研究的方向。



## 参考文献

- [1] 王达, 等著. 虚拟专用网精解. 北京:清华大学出版社. 2004 3~30
- [2] 贾晶, 陈元 王丽娜著. 信息系统的安全与保密. 第1 版. 北京:清华大学出版社. 1999,1.141-142
- [3] 戴宗坤, 唐三平著. VPN 与网络安全. 北京:电子工业出版社, 2002.1 10
- [4] (美)Casey Wilson (美)Peter Doak 著, 锺鸣魏允韬等译. 虚拟专用网的创建与实现. 北京:机械工业出版社, 2000
- [5] William R.Cheswick, Steven M.Bellovin 著. 防火墙与因特网安全. 戴宗坤,罗万伯译. 北京: 机械工业出版社, 2002. 2 8
- [6] Eric Rescorla 著. SSL 与TLS.崔凯译. 北京:中国电力出版社, 2002.35 38
- [7] <http://openvpn.net/howto.html> last visit on 2009.5.20.
- [8] <http://vtun.sourceforge.net/> last visit on 2009.5.20.
- [9] <http://openvpn.net/faq.html> last visit on 2009.5.20.
- [10] Charlie Hosner. OpenVPN and the SSL VPN Revolution .SANS Institute 2004
- [11] 雍建明. 虚拟专用网VPN 及其数据封装技术. 数据通信,1999,2:22 24
- [12] Naganand Doraswamy Dan Harkins 著京京工作室译. IPSec 新一代因特网安全标准. 机械工业出版社2000 年1 月
- [13] M Aydos. T Yanik. CK Koc. “High-speed Implementation of an ECC—based wireless Authentication Protocol On all ARM 1Vficroprocessor[J] ” . IEEE Proceedings-Communications, 2001, 5: 25—29.
- [14] C. H. Bennett, and G. Brassard, “Quantum cryptograghy:public key distribution and coin tossing”, Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing, Bangalore, India, pp. 175-179, 1984.
- [15] 喀兴林, 《高等量子力学》, 高等教育出版社, 2001。
- [16] 曾谨言, 《量子力学》卷I, 科学出版社, 2000。
- [17] W. K. Wootters, and W. H. Zurek, “A single quantum cannot be cloned” , Nature, vol. 299, no. 5586, pp. 802-803, 1982.
- [18] 李津生、洪佩琳, 《下一代Internet 的网络技术》, 人民邮电出版社, 2001。
- [19] Gisin N. Preprint Quant-ph/0101098. [http://www. Arxiv.org](http://www.Arxiv.org), 2001