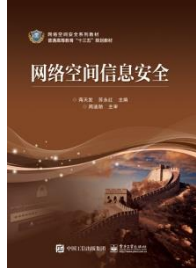
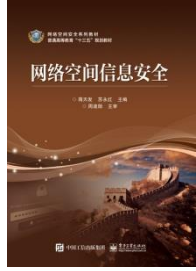


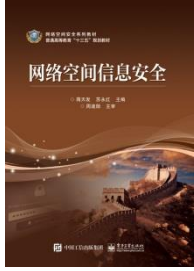
网络空间信息安全

第6章 网络安全协议



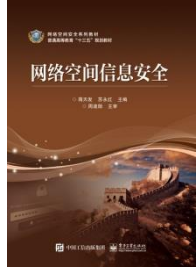
本章主要内容

- 6.1 概述
- 6.2 网络安全协议的类型
- 6.3 网络层安全协议IPSec
- 6.4 传输层安全协议SSL/TSL
- 6.5 应用层安全协议
- 6.6 EPC的密码机制和安全协议



6.1 概述

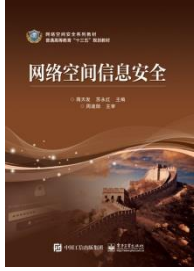
- 网络安全协议就是在协议中采用了若干密码算法协议——加密技术、认证技术、以保证信息安全交换的网络协议。
- 安全协议具有以下三种特点：
- 1、保密性：即通信的内容不向他人泄漏。为了维护个人权利，必须确保通信内容发给所指定的人，同时还必须防止某些怀有特殊目的的人的“窃听”。



6.1 概述

2、完整性：把通信的内容按照某种算法加密，生成密码文件即密文进行传输。在接收端对通信内容进行破译，必须保证破译后的内容与发出前的内容完全一致。

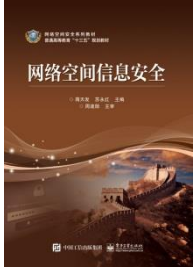
3、认证性：防止非法的通信者进入。进行通信时，必须先确认通信双方的真实身份。甲乙双方进行通信，必须确认甲乙是真正的通信双方，防止除甲乙以外的人冒充甲或乙的身份进行通信。



6.1 概述

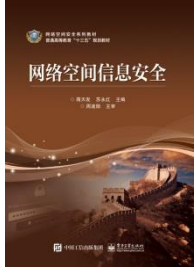
网络安全协议按照其完成的功能可以分为:

- （1）密钥建立协议 :一般情况下是在参与协议的两个或者多个实体之间建立共享的秘密，通常用于建立在一次通信中所使用的会话密钥。已有密钥建立协议，如Diffie-Hellman协议，Blom协议，MQV协议，端—端协议、MTI协议等。
- （2）认证协议:认证协议中包括实体认证（身份认证）协议、消息认证协议、数据源认证和数据目的认证协议等，用来防止假冒、篡改、否认等攻击。



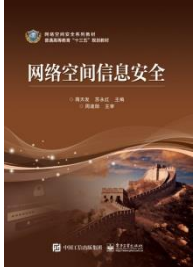
6.1 概述

- 最具代表性的身份认证协议有两类：一类是1984年Shamir提出的基于身份的身份认证协议；另一类是1986年Fiat等人提出的零知识身份认证协议。随后，人们在这两类协议的基础上又提出了新的使用的身份认证协议：Schnorr协议、Okamoto协议、Guillou-Quisquater协议和Feige-Fiat-Shamir协议等。
- 数字签名协议主要有两类：一类是普通数字签名协议，通常称为数字签名算法，如RSA数字签名算法、DSA等；另一类是特殊数字签名协议，如不可否认的数字签名协议、Fail-Stop数字签名协议、群数字签名协议等。



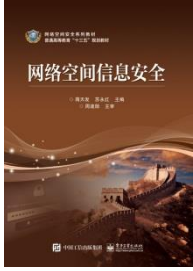
6.1 概述

- （3）认证和密钥交换协议 ：将认证和密钥交换协议结合在一起，是网络通信中最普遍应用的安全协议。该类协议首先对通信实体的身份进行认证，如果认证成功，进一步进行密钥交换，以建立通信中的工作密钥，也叫密钥确认协议。
- 常见的认证密钥交换协议有：互联网密钥交换（IKE） 协议、分布式认证安全服务（DASS）协议、Kerberos协议、X.509协议。



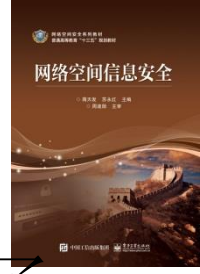
6.1 概述

- (4)电子商务协议 这类协议用于电子商务系统中以确保电子支付和电子交易的安全性、可靠性、公平性。常见的协议有：SET协议、iKP协议和电子现金等。
- (5)安全通信协议 这类协议用于计算机通信网络中保证信息的安全交换。常见的协议有：PPTP/L2PP 协议、IPSEC协议、SSL/TLS协议、PGP协议、SIME协议、S-HTTP协议等。

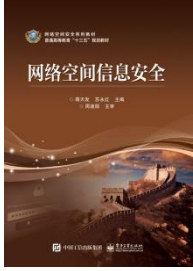


6.2 网络安全协议的类型-IPSec

- IPSec是IP Security的缩写。为了加强Internet的安全性，Internet安全协议工程任务组研究制定了IPSec协议用于保护IP层通信的安全协议。
- IPsec协议工作在OSI 模型的第三层，与传输层或更高层的协议相比，IPsec协议必须处理可靠性和分片的问题，这同时也增加了它的复杂性和处理开销。相对而言，SSL/TLS依靠更高层的TCP(OSI的第四层)来管理可靠性和分片。IPSec 基于端对端的安全模式，在源 IP 和目标 IP 地址之间建立信任和安全性。只有发送和接收的计算机需要知道通讯是安全的。



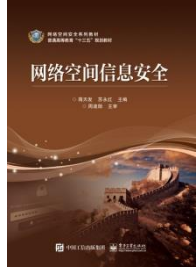
- IPSec 协议不是一个单独的协议，它给出了应用于IP层上网络数据安全的一整套体系结构，包括网络认证协议 Authentication Header(AH)、封装安全载荷协议。
- Encapsulating Security Payload(ESP)、密钥管理协议Internet Key Exchange (IKE)和用于网络认证及加密的一些算法等。这些协议用于提供数据认证、数据完整性和加密性三种保护形式。AH和ESP都可以提供认证服务，但AH提供的认证服务要强于ESP。而IKE主要是对密钥进行交换管理，对算法、协议和密钥3个方面进行协商。



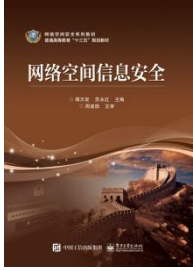
6.2 网络安全协议的类型-SSL

- SSL(Secure Sockets Layer 安全套接层)协议由两层组成，底层是建立在可靠的传输协议（例如：TCP）上的是SSL的记录层，用来封装高层的协议。SSL握手协议准许服务器端与客户端在开始传输数据前，能够通过特定的加密算法相互鉴别。SSL的先进之处在于它是一个独立的应用协议，其它更高层协议能够建立在SSL协议上。

  目前大部分的Web Server及Browser大多支持SSL的资料加密传输协定。因此，可以利用这个功能，将部分具有机密性质的网页设定在加密的传输模式，如此即可避免资料在网络上传送时被其他人窃听。 它利用公开密钥的加密技术（RSA）来作为用户端与主机端在传送机密资料时的加密通讯协定。

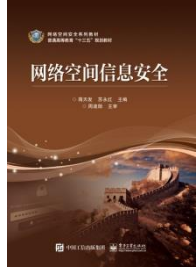


- 2014年10月15日，Google发布了一份关于SSLv3（SSL3.0）漏洞的简要分析报告。该报告认为SSLv3漏洞贯穿于所有的SSLv3版本中，利用该漏洞，黑客可以通过中间人攻击等类似的方式(只要劫持到的数据加密两端均使用SSL3.0)，便可以成功获取到传输数据(例如cookies)。
- BEAST既是此种攻击，攻击者可获取SSL通信中的部分信息的明文，对明文内容的完全控制。而另一种POODLE攻击是针对SSLv3中CBC模式加密算法，和BEAST不同的是，它不需要对明文内容的完全控制。问题的原因出在SSL设计上:SSL先进行认证之后再加密，SSL的加密和认证过程搞反了。
- 在SSLv3之后，TLS 1.0开始出现，TLS(Transport Layer Security)安全传输层协议是SSLv3发展的新阶段，TLS 1.0就等于SSLv3.1。

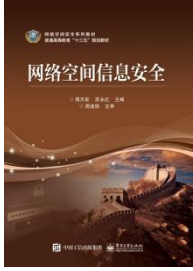


6.2 网络安全协议的类型-SET

- SET是Secure Electronic Transaction的缩写，即安全电子交易，它可以保证消费者信用卡数据不会被泄露或窃取。
- SET协议中，支付环境的信息保密性是通过公钥加密法RSA和私钥加密法DES相结合的算法来加密支付信息而获得的。消息首先以56位的DES密钥加密，然后装入使用1024位RSA公钥加密的数字信封在交易双方传输。
- SET协议是通过数字签名（双重签名）方案来保证消息的完整性和进行消息源的认证的，数字签名通过RSA加密算法结合生成信息摘要(消息通过HASH函数处理后得到的唯一对应于该消息的数值)，信息摘要的特征保证了信息的完整性。



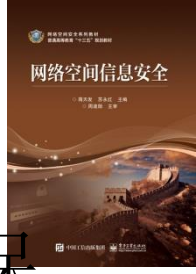
- 网上银行使用已经存在的程序和设备通过确认信用卡、清算客户银行户头完成交易，SET协议则通过隐藏信用卡号来保证整个支付过程的安全。因此，SET必须保证信用卡持有者与银行在现存系统和网络上能够保持持续的联系。
- SET是由Electronic Wallet（电子钱包）、Merchant Server（商店端服务机）、Payment Gateway（付款转接站）和Certification Authority（认证中心）组成的，它们构成了Internet 上符合SET标准的信用卡授权交易。



6.2 网络安全协议的类型-S-HTTP

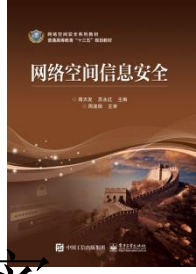
- 安全超文本传输协议（Secure HyperText Transfer Protocol，S-HTTP）是EIT公司结合HTTP而设计的一种消息安全通信协议。
- S-HTTP协议处于应用层，它是HTTP协议的扩展，它仅适用于HTTP联结上，S-HTTP可提供通信保密、身份识别、可信赖的信息传输服务及数字签名等。S-HTTP提供了完整且灵活的加密算法及相关参数。选项协商用来确定客户机和服务器在安全事务处理模式、加密算法（如用于签名的非对称算法 RSA 和 DSA等、用于对称加解密的 DES 和 RC2 等）及证书选择等方面达成一致。

  S-HTTP 支持端对端安全传输，客户机可能“首先”启动安全传输（使用报头的信息），如，它可以用来支持加密技术。S-HTTP是通过在S-HTTP所交换包的特殊头标志来建立安全通讯的。当使用S-HTTP时，敏感的数据信息不会在网络上明文传输。



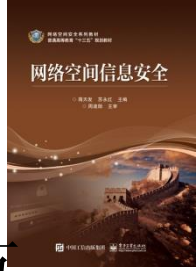
6.2 网络安全协议的类型-PGP

- PGP是英文Pretty Good Privacy(更好地保护隐私)的简称，是一个基于RSA公钥&私钥及AES等加密算法的加密软件系统，常用的版本是PGP Desktop Professional(PGP专业桌面版)。
- PGP是用一个128位的二进制数作为“邮件文摘（对一封邮件用某种算法算出一个最能体现这封邮件特征的数来，一旦邮件有任何改变这个数都会变化，那么这个数加上作者的名字（实际上在作者的密匙里）还有日期等等，就可以作为一个签名了。）”的，用来产生它的算法叫MD5(message digest 5)。 MD5是一种单向散列算法，很难找到一份替代的邮件与原件具有同样的MD5特征值。



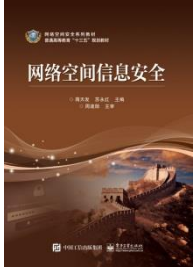
6.2 网络安全协议的类型-PGP

- 甲用自己的私匙将上述的128位的特征值加密并附加在邮件后，然后用乙的公匙将整个邮件加密。乙收到密文以后，用自己的私匙将邮件解密，从而得到甲的原文和签名，乙的PGP也从原文计算出一个128位的特征值来和用甲的公匙解密签名所得到的数比较，如果符合就说明这份邮件确实是甲寄来的。这样两个安全性要求都得到了满足。



6.2 网络安全协议的类型-PEM

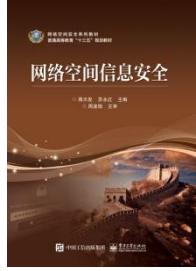
- PEM是增强电子邮件隐秘性的标准草案，它在互联网电子邮件的标准格式上增加了加密、鉴别和密钥管理的功能，允许使用公开密钥和专用密钥的加密方式，并能够支持多种加密工具。
- PEM提供以下四种安全服务：
  - 数据隐蔽:使数据免遭非授权的泄露,防止有人半路截取和窃听。
  - 数据完整性:提供通信期间数据的完整性可用于侦查和防止数据的伪造和篡改。
  - 对发送方的鉴别:用来证明发送方的身份防止有人冒名顶替。
  - 防发送方否认:结合上述功能,防止发送方事后不承认发送过此文件。



6.2 网络安全协议的类型-PEM

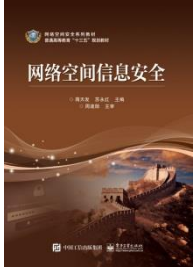
- PEM安全功能使用了多种密码工具,包括非对称加密算法RSA，，对称加密算法DES以及报文完整性.对RSA来说，通信双方均需2个密钥，DES要求通信双方共享一个密钥。DES的优点是软件实现比较快（比RSA快100倍），缺点是不能用作鉴别。然而RSA 有数字签名，管理相对简单，但缺点是实现需要占用较多的CPU时间。PEM的加密过程通常包括四个步骤：报文生成:一般使用用户所常用的格式。规范化:转换成SMTP的内部表示形式。加密:执行选用的密码算法。编码:对加密后的报文进行编码以便传输。

6.2 网络安全协议的类型-S/MIE



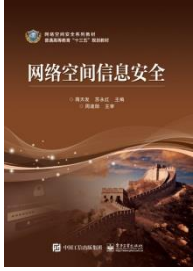
- S/MIME，多用途网际邮件扩充协议（Secure Multipurpose Internet Mail Extensions. RFC 2311）。Internet电子邮件由一个邮件头部和一个可选的邮件主体组成，其中邮件头部含有邮件的发送方和接收方的有关信息。
- 用户可以使用MIME增加非文本对象，比如把图像、音频、格式化的文本或微软的Word文件加到邮件主体中去。MIME中的数据类型一般是复合型的，也称为复合数据。
- S/MIME是多功能电子邮件扩充报文基础上添加数字签名和加密技术的一种协议，是在MIME上定义安全服务措施的实施方式。目前，S/MIME已成为产业界广泛认可的协议。
-

6.2 网络安全协议的类型-S/MIE

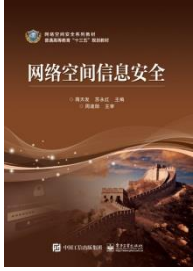


- S/MIME只保护邮件的邮件主体，对头部信息则不进行加密。S/MIME增加了新的MIME数据类型如“应用 /pkcs7-MIME”（application/pkcs7-MIME）、“复合/已签名”（multipart/signed）和“应用 /pkcs7-签名”（application/pkcs7-signature）等，这些复合数据用于提供数据保密、完整性保护、认证和鉴定服务等功能。邮件如果包含了上述MIME复合数据，则有关的MIME 附件也会在邮件中存在。接收者在客户端阅读邮件之前，S/MIME应用处理这些附件。它是MIME 的安全版本。

6.2 网络安全协议的类型-SSH



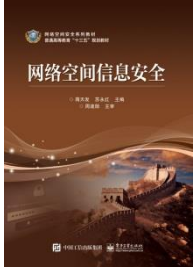
- SSH（ Secure Shell，安全外壳协议），IETF 的网络小组（Network Working Group）所制定，专为远程登录会话和其他网络服务提供安全性的协议。
- 传统的网络服务程序，如：ftp、pop和telnet在本质上都是不安全的，因为它们在网络上用明文传送口令和数据，别有用心的人非常容易就可以截获这些口令和数据。这些别有用心的人就是“中间人”(man-in-the-middle)，他们冒充真正的服务器接收你发送的数据，还冒充你发送数据给服务器。SSH使得这种“中间人”攻击不复存在，它对所传输的数据加密，还可以防止DNS和IP欺骗。而且它传输的数据速度很快，因为这些数据都是经过压缩的。



6.2 网络安全协议的类型-SSH

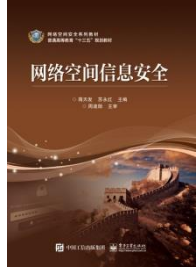
- SSH提供两种级别的验证——
  - 基于口令的认证
  - （需传送口令，可能受到“中间人”攻击）
  - 基于密匙的验证（不需要传送口令，但是你必须知道自己的密匙，登录过程需要10秒）。
- SSH共分为三层
  - 传输层协议（SSH-TRANS），提供强力的加密技术、密码主机认证及完整性保护。
  - 用户认证协议 [SSH-USERAUTH]，用于向服务器提供客户端用户鉴别功能。
  - 连接协议 [SSH-CONNECT]将多个加密隧道分成逻辑通道，它提供了交互式登录话路、远程命令执行、转发 TCP/IP 连接和转发 X11 连接。

6.2 网络安全协议的类型-Kerberos

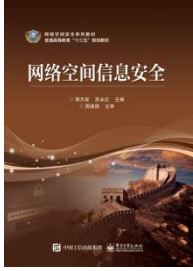


- kerberos：（Network Authentication Protocol，网络认证协议）。苹果的Mac OS X，Red Hat Enterprise Linux4 和后续的操作系统，windows2000和后续的操作系统都默认Kerberos为其默认认证方法。
- 它采用客户端/服务器结构与DES加密技术，客户端和服务器端能够相互认证，可用于防止窃听、防止replay攻击、保护数据完整性等场合，是一种应用对称密钥体制进行密钥管理的系统。

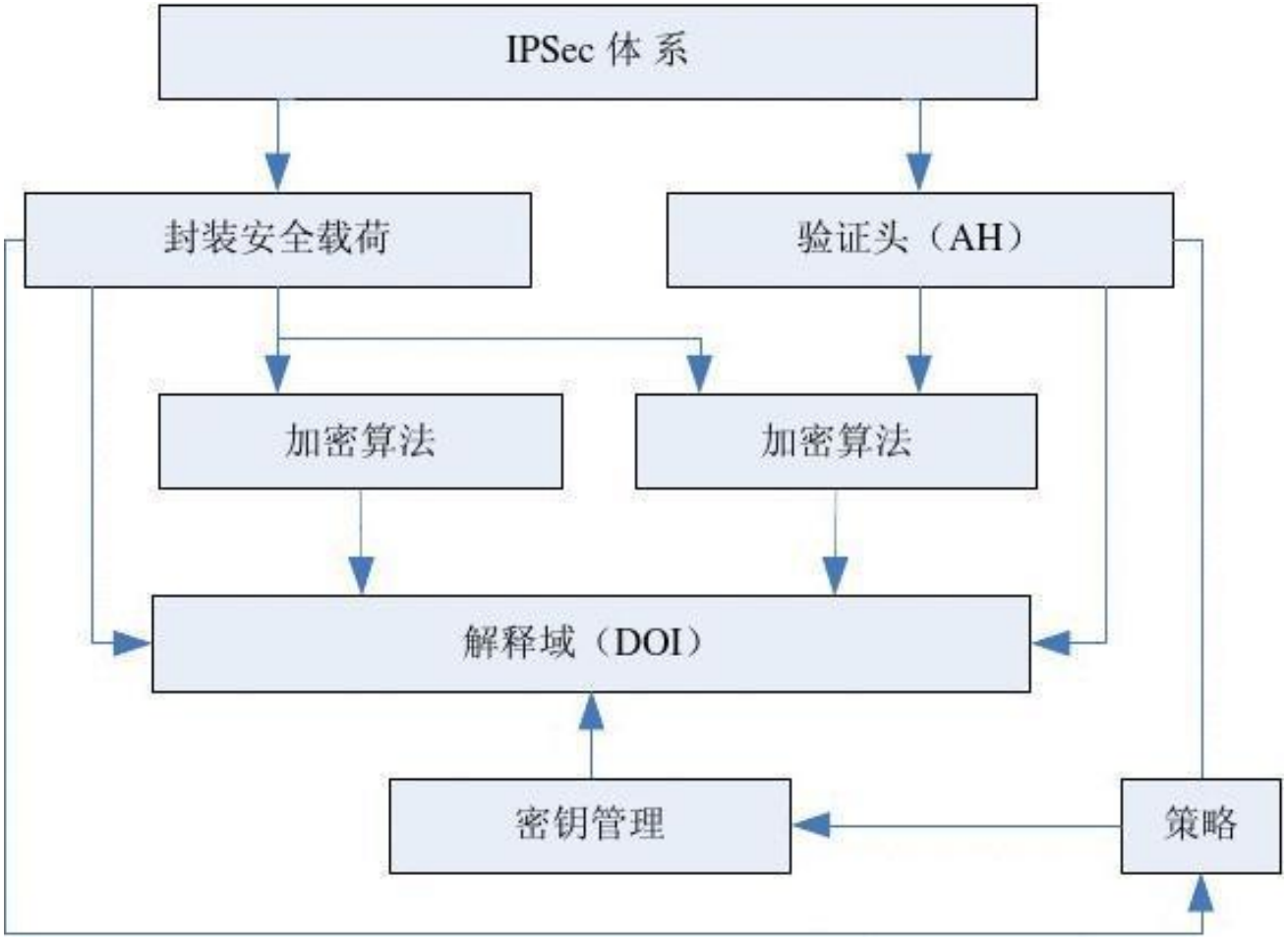
6.2 网络安全协议的类型-Kerberos

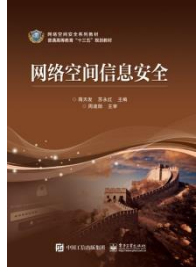


- Kerberos的认证过程：客户机向认证服务器（AS）发送请求，要求得到某服务器的证书，然后AS的响应包含这些用客户端密钥加密的证书。
- 证书的构成为：
  1）服务器 “ticket” ；
  2）一个临时加密密钥（又称为会话密钥 “session key”） 。
- 客户机将 ticket （包括用服务器密钥加密的客户机身份和一份会话密钥的拷贝）传送到服务器上。



6.3 网络层安全协议IPSec



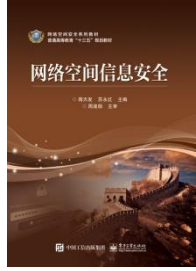


6.3 网络层安全协议IPSec

IPSec协议由四个主要部分组成：

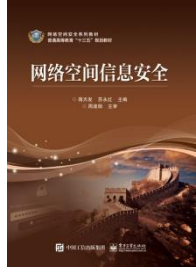
- 安全载荷协议
  ESP(IP Encapsulating Security Payload)、
- 认证头协议AH (IP Authentication Header)、
- 安全关联SA（Security Associations）
- 密钥管理IKE
  （Internet Key Exchange Protocol）。

其中，AH和ESP都可以提供认证服务，但AH提供的认证服务要强于ESP。而IKE主要是对密钥进行交换管理，对算法、协议和密钥3个方面进行协商。



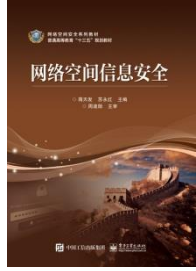
6.3 网络层安全协议IPSec

- IPsec协议工作在OSI 模型的第三层，使其在单独使用时适于保护基于TCP或UDP的协议（如 安全套接子层（SSL）就不能保护UDP层的通信流）。
- 与传输层或更高层的协议相比，IPsec协议必须处理可靠性和分片的问题，这同时也增加了它的复杂性和处理开销。而SSL/TLS依靠更高层的TCP（OSI的第四层）来管理可靠性和分片。



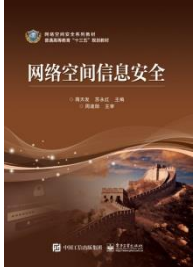
6.3 网络层安全协议IPSec

- IPSec提供了两种安全协议：认证头AH(Authentication Header)和封装安全有效载荷ESP(Encapsulating Security Payload)，这两个协议以IP扩展头的方式增加到IP包中，可以对IP数据包或上层协议数据包进行安全保护，增加了对IP数据项的安全性。
- AH只提供了数据完整性认证机制，用来证明数据源端点，保证数据完整性，防止数据篡改和重播。ESP同时提供数据完整性认证和数据加密传输机制。



6.3 网络层安全协议IPSec

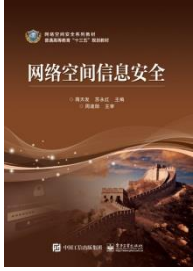
- IPSec提供了两种安全协议：
(1)认证头传输模式：为上层协议数据和选择的IP头字段提供认证保护，且仅适用于主机实现。在这种模式中，AH和ESP会拦截从传输层到网络层的数据包，并根据具体的配置提供安全保护。
- (2)隧道模式：对整个IP数据项提供认证保护，除了用于主机还可用于安全网关。如果安全性是由一个设备来提供的，而该设备并非数据包的始发点，或者数据包需要保密传输到与实际目的地不同的另一个目的地，则需要采用隧道模式。



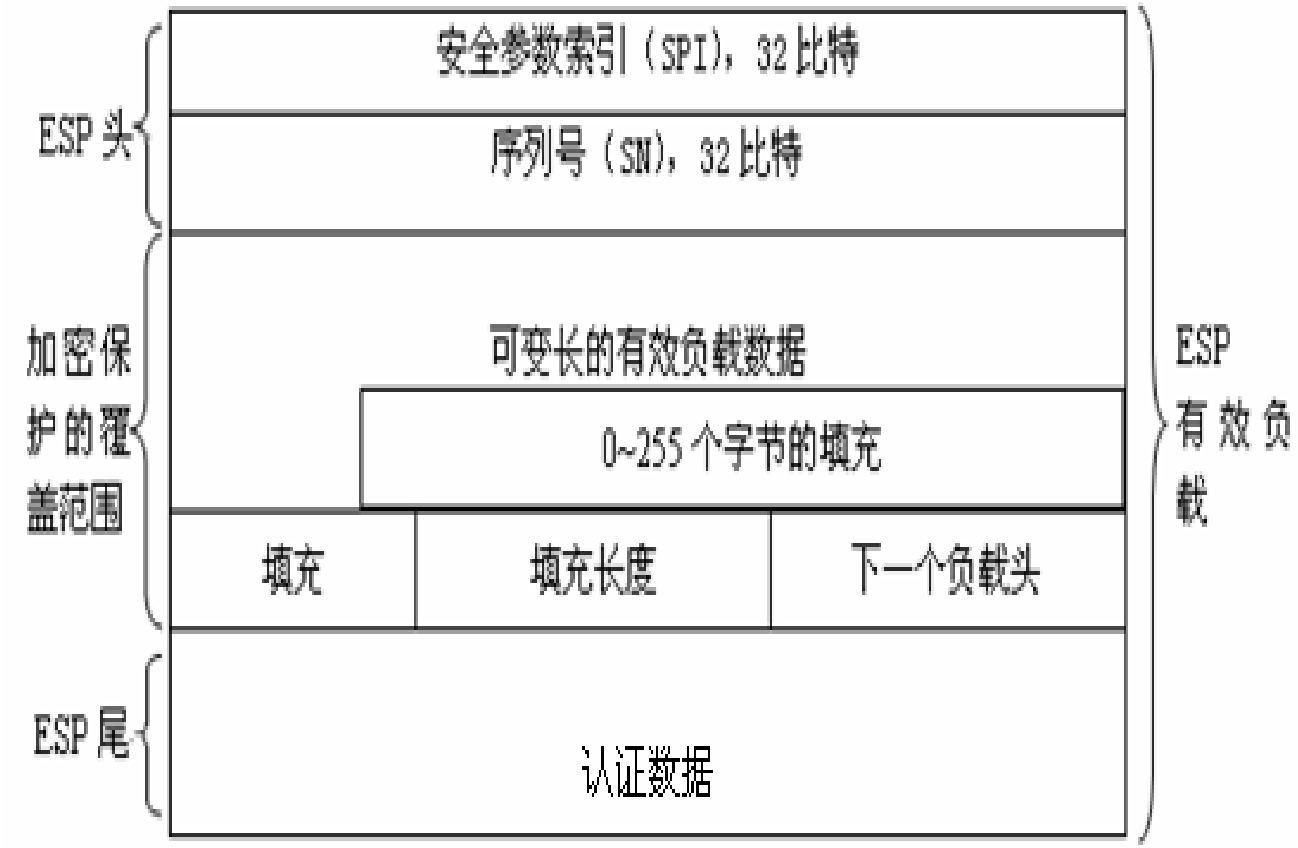
6.3 网络层安全协议IPSec

AH和ESP可以分别单独使用，也可以联合使用。每个协议都支持两种应用模式，即传输模式和隧道模式。这两种模式的区别是其所保护的内容不相同：一个是IP包，一个是IP载荷。

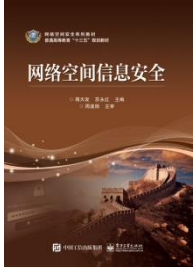
Mode / Protocol	Transport	Tunnel
AH	IP AH Data	IP AH IP Data
ESP	IP ESP Data ESP-T	IP ESP IP Data ESP-T
AH-ESP	IP AH ESP Data ESP-T	IP AH ESP IP Data ESP-T


网络空间信息安全

ESP 格式


安全参数索引（SPI），32比特
序列号（SN），32比特
可变长的有效负载数据
0~255个字节的填充
填充
填充长度
下一个负载头
认证数据
ESP头
加密保护的覆盖范围
ESP尾
ESP有效负载

- ESP头紧跟在IP头后。在IPv4中，这个IP头的协议字段是50，以表明IP头之后是一个ESP头。
- 安全参数索引(SPI)：它是一个32位的随机数。通过目的地址和安全协议(ESP)，来标识这个数据所属的安全关联。接收方通过这个字段就对收到的IP数据包进行相应的处理。通常，在密钥交换过程中由目标主机来选定SPI。SPI是经过验证的，但并没有加密，因为SPI是一种状态标识，由它来指定所采用的加密算法及密钥，以及对数据包进行解密。

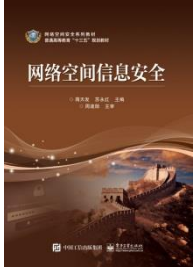


- 序列号(SN)：它是一个单向递增的32位无符号整数。使用序列号可以使ESP具有抗重播攻击的能力，因为通过它，可以区分使用同一组加密策略处理不同数据包。加密数据部分包含原IP数据包的有效负载和填充域。
- 有效负载数据：被ESP保护的数据包包含在载荷数据字段中，其字段长度由数据长度来决定。因此是可变长的数据。可通过下一负载头来指明其数据类型。
- 填充：0～255个字节，填充内容可以由密码算法来指定。

- 填充长度：该字段为8位，指出添加多少填充字段的长度，接收端利用它恢复载荷数据的实际长度。该字段是必须存在，因此，即使没有填充项时，其值也必须表示出来（为0）。
- 下一负载头：该字段为8位，用来指出有效负载所使用的类型。在隧道模式下使用ESP，则该值为4。如果在传输模式下使用，这个值表示它上一级协议的类型，如TCP对应的值为6。
- 认证数据：ESP数据的完整性校验值，该字段是可变长的。通常是由认证算法对ESP数据包进行密钥处理的散列函数。该字段是可选的，只有对ESP数据包进行处理的SA提供了完整性认证服务，才会有该字段。

- ESP协议有两种工作模式（以应用于IPv4的数据包为例）：传输模式(Transport Mode)和隧道模式(Tunnel Mode)

Mode / Protocol	Transport	Tunnel
AH	IP | AH | Data	IP | AH | IP | Data
ESP	IP | ESP | Data | ESP-T	IP | ESP | IP | Data | ESP-T
AH-ESP	IP | AH | ESP | Data | ESP-T	IP | AH | ESP | IP | Data | ESP-T



- AH头位于IPv6头和一些上层协议头之间，如果存在扩展包头，则AH必须位于逐跳选项头、选路扩展头和分段扩展头之后。AH的验证范围与ESP有所区别，包括了整个IPv6数据包。
- AH可以工作在传输模式及隧道模式下，下图分别显示了在传输模式和隧道模式中增加AH时IPv6数据包的变化。

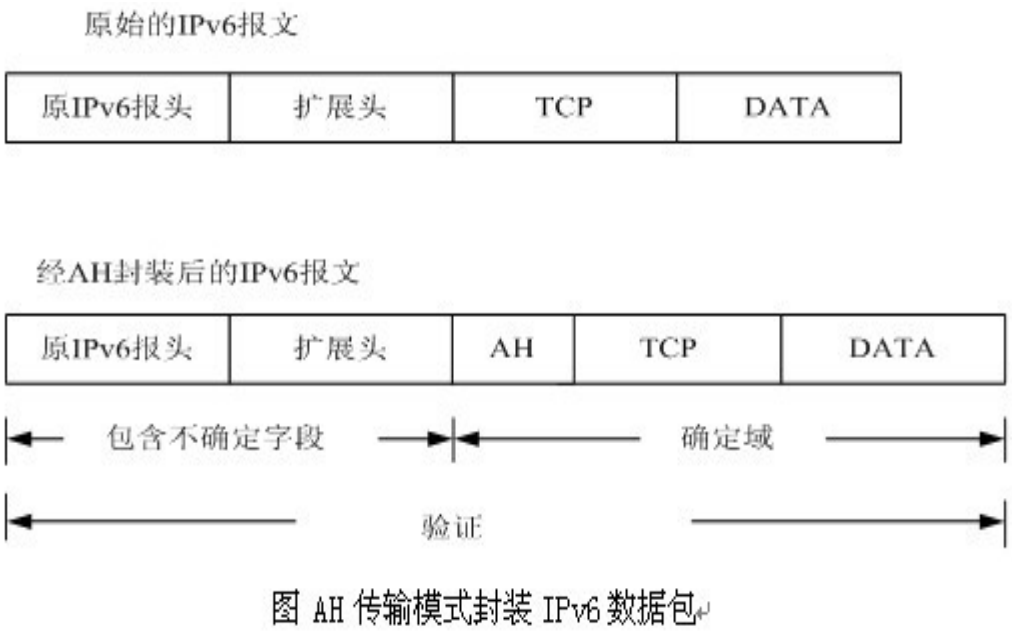
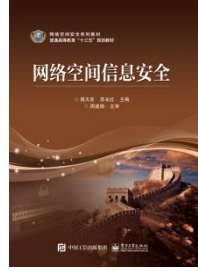
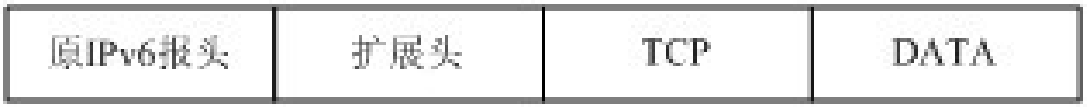


图 AH 传输模式封装 IPv6 数据包

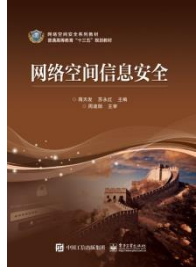


原始的IPv6报文



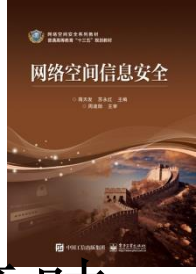
经AH封装后的IPv6报文

图 AH 隧道模式封装 IPv6 数据包



6.3 网络层安全协议IPSec

- IPsec安全关联（security association, SA）指定由通信主机识别的安全属性。单个 SA 保护单一方向的数据，此保护针对单个主机或一组（多播）地址。由于多数通信为对等通信或客户机/服务器通信，因此，必须存在两个SA来保证两个方向的通信安全。
  - 以下三个元素唯一地标识 IPsec SA：
  - 安全协议（AH 或 ESP）
  - 目标 IP 地址
  - security parameter index, SPI（安全参数索引）（任意 32 位的值，与 AH 或 ESP 包一起传输。）



6.3 网络层安全协议IPSec

- IPSec进行加密可以有两种工作模式，意味着SA也有两种工作模式即传输模式和隧道模式。

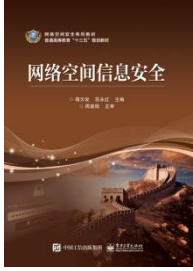
1. 传输模式的SA

在该模式下，经过IPSec处理的IP数据报格式如图：

IP 头	安全协议头（AH/ESP）	高层协议头	数据

图 6.10 经过传输模式 SA 处理的 IP 数据包格式

如果安全协议为ESP，则SA只为高层协议提供安全服务；如果选择了AH，则可将安全服务范围扩展到IP头的某些在传输过程中不变的字段。

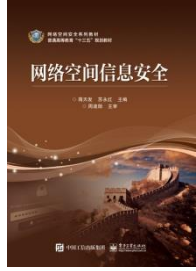


- 隧道模式的SA将在两安全网关之间或者主机与安全网关之间建立一个IP隧道。在这种模式下，IP数据包有两个IP头。一个是用来指明IPSec数据包源地址的外部IP头，；另一个是用于指明IP数据包的目的地址的内部IP头。

外部 IP 头	安全协议头（AH/ESP）	内部 IP 头	高层协议头	数据

图 6.11　经过隧道模式 SA 处理的 IP 数据报格式

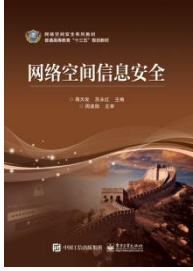
- 如果选择的安全协议为ESP，则SA只为内部IP头、高层协议头和数据提供安全服务；如果选择使用AH，则可将安全范围扩大到外部IP头中某些在传输过程中不变的字段。



6.3 网络层安全协议IPSec

- 密钥管理（IKE）并非IPSec专有，其他协议也可以用IKE进行具体的安全服务。在IPSec模型中，使用IPSec保护一个数据报之前，必须先建立一个SA，SA可以手工创建，也可以自动建立。在自动建立SA时，要使用IKE协议。IKE代表IPSec与SA进行协商，并将协商好的SA填入SAD中。

6.3 网络层安全协议IPSec



- IKE协议主要是对密钥交换进行管理，它主要包括四个功能：

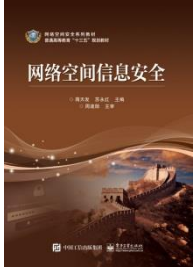
  1、协商服务：对使用的协议、加密算法和密钥进行协商。

  2、身份认证服务：对参与协商的身份进行认证，确保身份的合法性。

  3、密钥的管理：对协商的结果进行管理。

  4、安全交换：产生和交换所有密钥的信息。

6.3 网络层安全协议IPSec



- IKE是一种混合型协议，它建立在以下三个协议基础上：

  1、ISAKMP：它是一种密钥交换框架，独立于具体的密钥交换协议。在这个框架上，可以支持不同的密钥交换协议。

  2、OAKLEY：描述了一系列的密钥交换模式，以及每种模式所提供服务的细节，如身份保护和认证等。

  3、SKEME：描述了一种通用的密钥交换技术。这种技术提供了基于公钥的身份认证和快速密钥刷新。

6.3 网络层安全协议IPSec

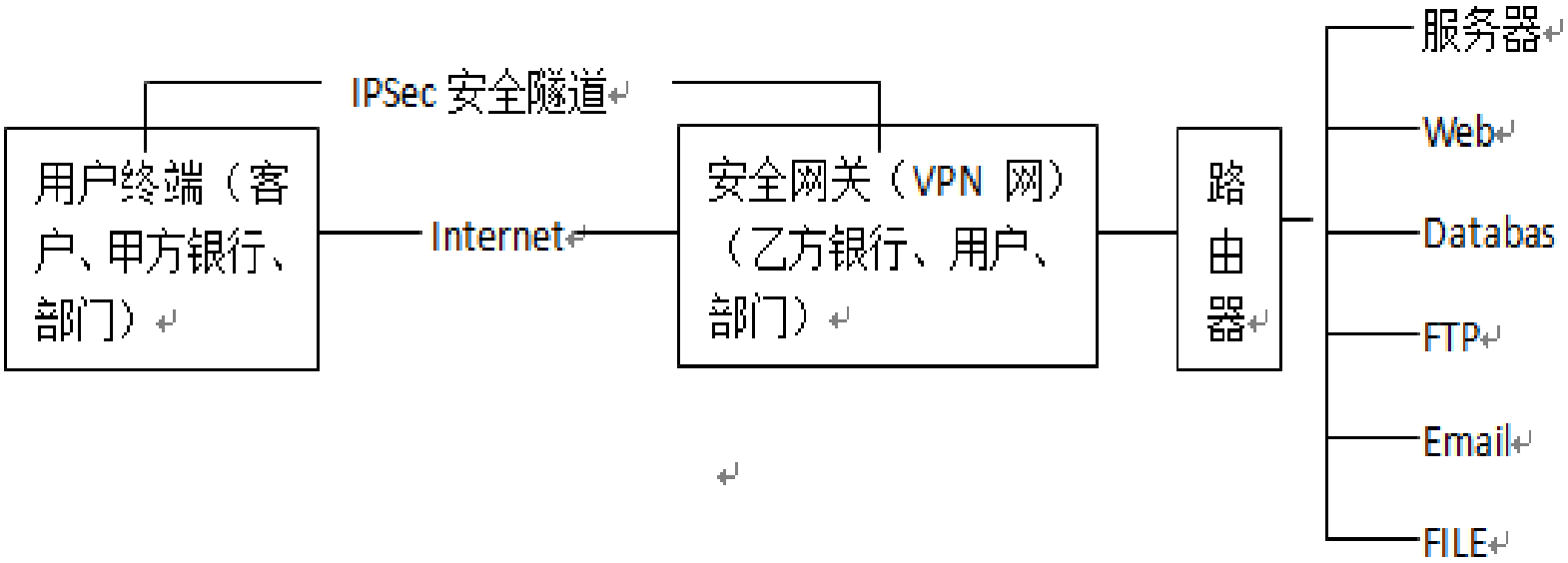
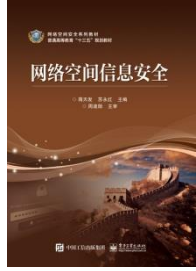


图 6.12 VPN-IPSec 拓扑结构

VPN-IPSec拓扑结构

- 用户在使用图6.12所示的VPN-IPSec拓扑结构中的IPSec安全隧道时，应按以下步骤进行：
- （1）用户终端与VPN虚拟专用网上的安全网关进行物理联接；
- （2）用户终端通过DHCP动态主机分配方式，动态获取一个专用的IP地址。注意：此IP地址在通过安全隧道通信之前要做动态变动。此IP地址在未变动前，先作为用户外出包的源地址和进入用户的进入包的目的地址；
- 3）进入IKE的初始化阶段，一个部门和输入数据的信息，首先通过IPSec的安全性参考SPI索引，自动选择AH协议或者ESP协议，选择加密算法、密钥和密钥的相应生存周期，进行了以上初始化定义后。

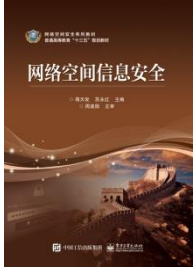


6.3 网络层安全协议IPSec

（4）进行安全关联SA处理。在IKE第一阶段，终端与网关之间只有一对交互数据包，在IKE的第二阶段会生成满足双方请求相应的四对交互数据包；

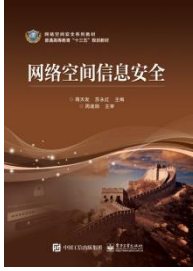
（5）经过IKE的两个阶段后，就建成动态的专用安全隧道，此后进入和输出的数据包都要通过不同的SPI进行标识；

（6）安全隧道建成后，安全网关又通过DHCP为用户终端分配一个防窃取的IP地址，供用户及对方使用。



6.4 传输层安全协议SSL/TSL

- SSL(Secure Sockets Layer 安全套接层),及其继任者传输层安全（Transport Layer Security，TLS）是为网络通信提供安全及数据完整性的一种安全协议。TLS与SSL在传输层对网络连接进行加密。SSL协议建立在运输层和应用层之间，是提供客户和服务器双方网络应用安全通信的开放式协议。SSL协议是一个分层协议，由两层组成：SSL握手协议和SSL记录协议。其中记录协议在握手协议的下端。



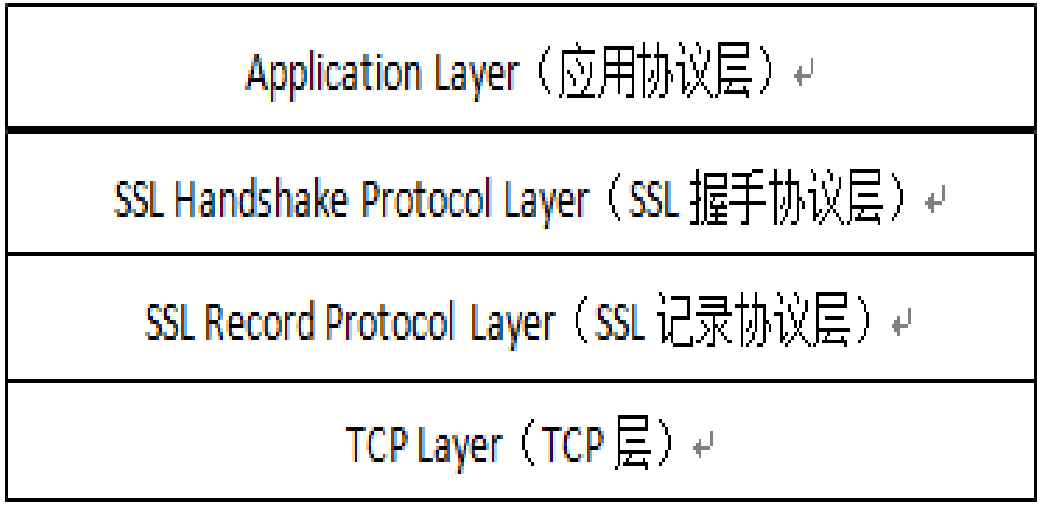
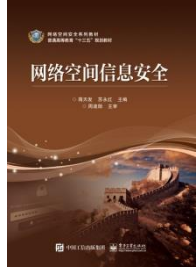


图 6.13　SSL 协议的基本结构

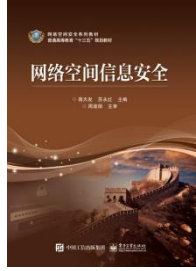
其中Handshake Protocol用来协商密钥，协议的大部分内容就是通信双方如何利用它来安全的协商出一份密钥。 Record Protocol则定义了传输格式。

6.4 传输层安全协议SSL/TSL



SSL/TLS可提供3中基本的安全功能服务：

- 信息加密。加密技术既有对称加密技术DES、IDEA，也有非对称加密技术RSA。加密数据可防止数据中途被窃取。
- 身份认证。认证的算法包括RSA（数字签名技术）、DSA（数字签名算法）、ECDSA（椭圆曲线数字签名算法）。SSL协议要求在握手交换数据前进行验证，为的是确保用户的合法性。认证用户和服务器从而确保数据发送到正确的客户机和服务器。
- 信息完整性校验。发送方通过散列函数产生的消息验证码（MAC），接收方验证MAC来保证信息的完整性。维护数据的完整性，确保数据在传输过程中不被改变。

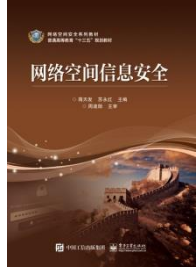


SSL握手协议是位于SSL记录协议之上的主要子协议，包含两个阶段。

第一阶段用于交换密钥等信息，通信双方通过相互发送HELLO消息进行初始化。通过HELLO消息，就有足够的信息确定是否需要一个新的密钥。

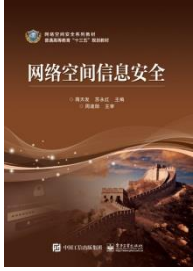
如果本次会话是建立在一个已有的连接上，双方则进入握手协议的第二阶段；如果本次会话是一个新会话，则需要产生新的密钥，双方需要进入密钥交换过程。此时服务器方的SERVER—HELLO消息将包含足够的信息使客户方产生一个新的密钥。

第二阶段用于用户身份认证。通常服务器方向客户方发出认证请求消息，客户方在收到该请求后，发出自己的证书，并等待服务器的应答。服务器如果收到客户的证书，则返回成功的消息，否则返回错误的消息。至此，握手协议结束。



6.4 传输层安全协议SSL/TSL

- 在SSL协议中，所有要传输的数据都被封装在记录中，记录协议规定了数据传输格式，它包括应用程序提供的信息的压缩、数据认证等。记录由记录头和长度不为0的记录数据组成。所有的SSL通信，包括握手消息、安全记录和应用数据，他们都要通过SSL记录层传送。
- 在SSL记录层，主要提供机密性和报文完整性服务。每个上层应用数据被分成2^{14}字节或更小的数据块，封装在一个SSL记录中。多个同种类型的客户信息可能被连接成一个单一的SSL明文记录。



信息类型	次要版本	主要版本	压缩长度	数据段

图 6.14　SSL 记录格式

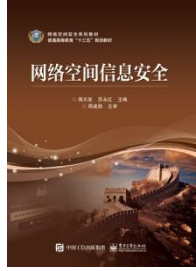
(1)信息类型：该字段为8位。指示封装在数据段中的信息类型。

(2)主要版本：该字段为8位。表明所使用SSL协议的主要版本号。

(3)次要版本：该字段为8位。表明所使用SSL协议的次要版本号。对于SSL v3.0，值为0。

(4)压缩长度：该字段为16位。以字节为单位表示数据段的长度。

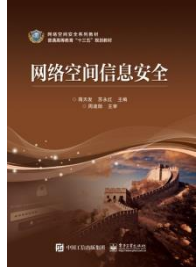
(5)数据段：上层协议处理的数据。



发送方记录层的工作过程如下(接收方过程反之)：

- 分段：把从高层接收到的数据进行分段，使其长度不超过2^{14}个字节。
- 压缩：该操作是可选的。SSL记录协议不指定压缩算法，但压缩必须是无损的。实际情况中，有时使用压缩算法后，使得数据扩大了。因此必须保证不能增加2^{10}字节以上的长度。对压缩后的数据计算SSL记录验证码即MAC。记录可使用专用公式进行计算。
- 用当前握手协议协商的一套加密参数中指定的MAC算法计算压缩后数据的MAC；用加密算法加密压缩数据和记录验证的MAC，形成密文结构。而且加密不能增加2^{10}字节以上的内容长度。

6.4 传输层安全协议SSL/TSL



- SSL最初的几个版本（SSL 1.0、SSL2.0、SSL 3.0）由网景公司设计和维护，从3.1版本开始，SSL协议由因特网工程任务小组（IETF）正式接管，并更名为TLS（Transport Layer Security），发展至今已有TLS 1.0、TLS1.1、TLS1.2这几个版本。新版本的TLS（Transport Layer Security，传输层安全协议）是IETF（Internet Engineering Task Force，Internet工程任务组）制定的一种新的协议，它建立在SSL 3.0协议规范之上，是SSL 3.0的后续版本。目前市面上所有的Https都是用的是TLS，而不是SSL。

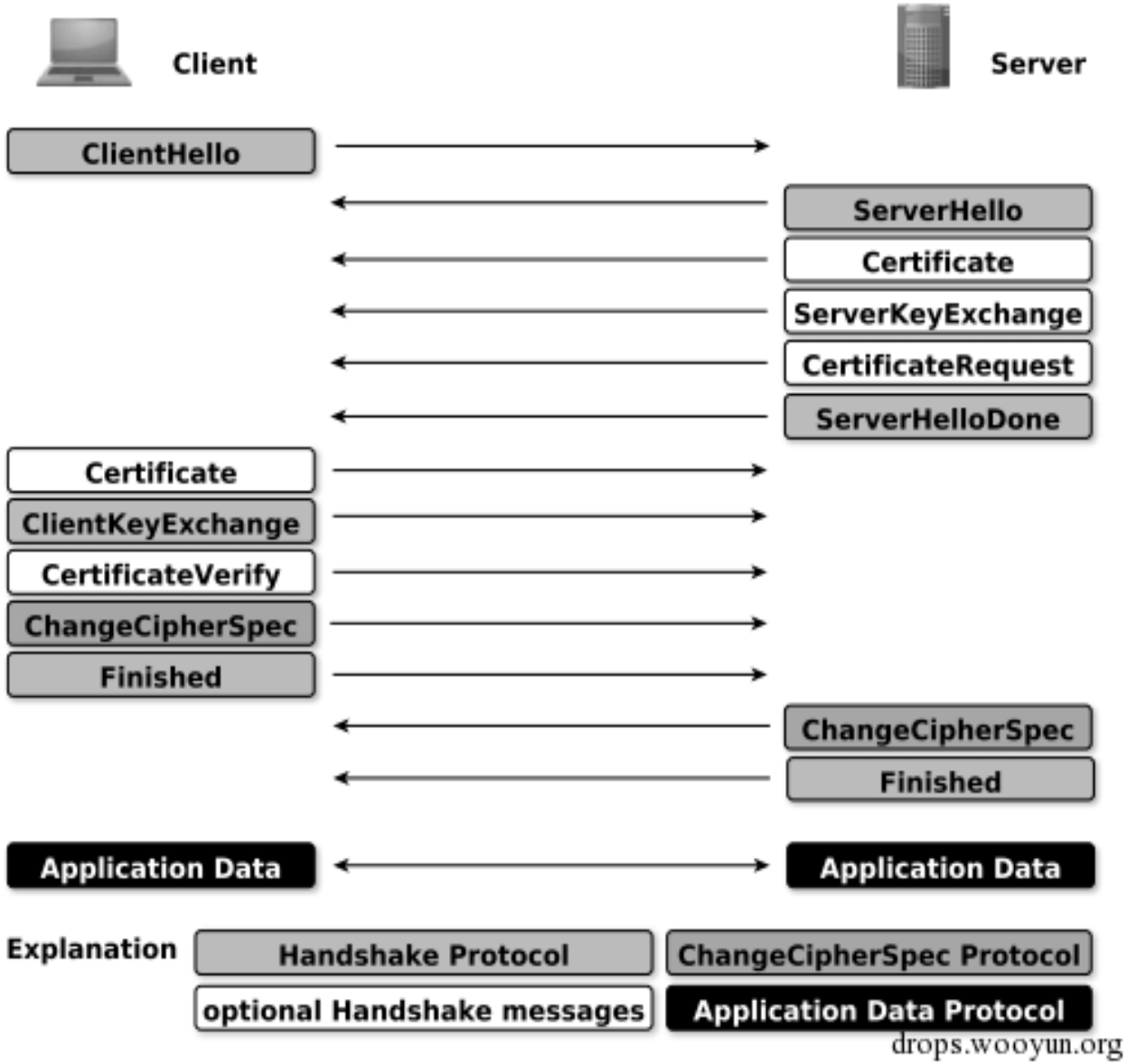
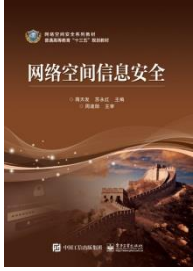


图6.16 典型的TLS 1.0协议交互流程



TLS与SSL的差异

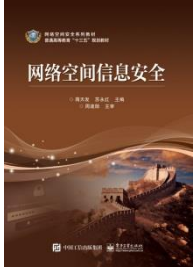
- 1）版本号：TLS记录格式与SSL记录格式相同，但版本号的值不同，TLS的版本1.0使用的版本号为SSLv3.1。

2）报文鉴别码：SSLv3.0和TLS的MAC算法及MAC计算的范围不同。TLS使用了RFC-2104定义的HMAC算法。SSLv3.0使用了相似的算法，两者差别在于SSLv3.0中，填充字节与密钥之间采用的是连接运算，而HMAC算法采用的是异或运算。但是两者的安全程度是相同的。

3）伪随机函数：TLS使用了称为PRF的伪随机函数来将密钥扩展成数据块，是更安全的方式。

4）报警代码：TLS支持几乎所有的SSLv3.0报警代码，而且TLS还补充定义了很多报警代码，如解密失败（decryption_failed）、记录溢出（record_overflow）、未知CA（unknown_ca）、拒绝访问（access_denied）等。

TLS与SSL的差异

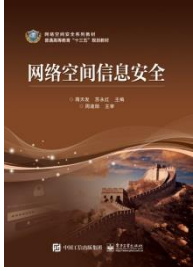


- 5）密文族和客户证书：SSLv3.0和TLS存在少量差别，即TLS不支持Fortezza密钥交换、加密算法和客户证书。

6）certificate_verify和finished消息：SSLv3.0和TLS在用certificate_verify和finished消息计算MD5和SHA-1散列码时，计算的输入有少许差别，但安全性相当。

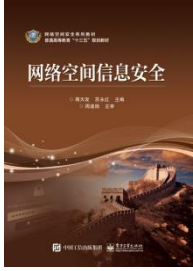
7）加密计算：TLS与SSLv3.0在计算主密值（master secret）时采用的方式不同。对称加密用以数据加密（DES 、RC4 等）。当然，现在很多人准备放弃RC4数据加密算法。对称加密所产生的密钥对每个连接都是唯一的，且此密钥基于另一个协议（如握手协议）协商。

TLS与SSL的差异



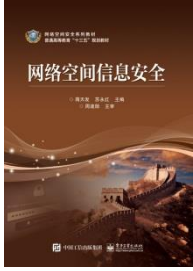
- 8）填充：用户数据加密之前需要增加的填充字节。在SSL中，填充后的数据长度要达到密文块长度的最小整数倍。而在TLS中，填充后的数据长度可以是密文块长度的任意整数倍（但填充的最大长度为255字节），这种方式可以防止基于对报文长度进行分析的攻击。

6.4 传输层安全协议SSL/TSL



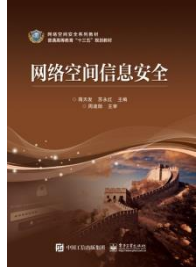
- TLS 的最大优势就在于：TLS 是独立于应用协议。高层协议可以透明地分布在 TLS 协议上面。然而， TLS 标准并没有规定应用程序如何在 TLS 上增加安全性；它把如何启动 TLS 握手协议以及如何解释交换的认证证书的决定权留给协议的设计者和实施者来判断。

在安全性的改进方面，TLS有以下优点：

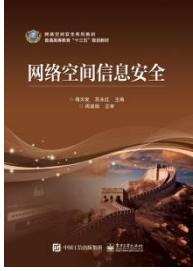


- 1）对于消息认证使用密钥散列法HMAC：当记录在开放的网络上传送时，该代码确保记录不会被变更。SSLv3.0还提供键控消息认证，但HMAC比SSLv3.0使用的（消息认证代码）MAC 功能更安全。
- 2）增强的伪随机功能（PRF）：PRF生成密钥数据。HMAC定义PRF，PRF使用两种散列算法保证其安全性。若任一算法暴露，只要第二种算法未暴露，则数据仍然是安全的。
-

在安全性的改进方面，TLS有以下优点：

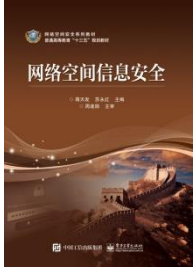


- 3）改进的已完成消息验证： TLS将已完成消息基于PRF和HMAC值之上，比SSLv3.0更安全。
- 4）一致证书处理：TLS试图指定必须在TLS之间实现交换的证书类型。
- 5）特定警报消息：TLS提供更多的特定和附加警，还对何时应该发送某些警报进行记录。



6.5 应用层安全协议

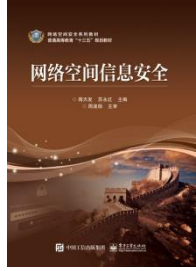
- 安全电子交易SET主要用于保障Internet上信用卡交易的安全性，利用SET给出的整套安全电子交易的过程规范，保护消费者信用卡不暴露给商家，它是公认的信用卡/借记卡的网上交易的国际标准，解决了电子商务的安全难题。
- 1997年5月31日正式推出SET1.0版，Secure Electronic Transaction，简称SET。它是VISA与MasterCard两大国际信用卡组织研发的。其主要目标是保证银行卡电子支付的安全。其数字证书验证及业务流程，也可以为其它电子支付方式所采用。



6.5.1 SET安全协议

SET协议要达到以下主要目标：

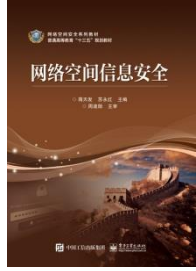
- 机密性。保护有关支付信息在Internet的安全传输，数据不被黑客窃取。
- 保护隐私。消费者给商家的订单中包含支付信用卡帐号、密码及隐私信息，但收到订单的商家只能看到订货信息，看不到帐户信息密码及隐私信息，银行只能看到支付信息，看不到订货信息。
- 完整性。采用密钥加密算法和Hash函数及数字信封技术，保证传输信息的完整性。



6.5.1 SET安全协议

SET协议要达到以下主要目标：

- 多方认证性。参与交易的交易者通过第三方权威机构进行身份认证。第三方权威机构还可提供在线交易方的信用担保。
- 标准性。为保证在线交易各方的不同操作平台和操作软件的相互兼容，SET要求各方遵循统一的协议和报文格式，包括加密算法、数字证书信息及其对象格式、订货信息及其对象格式、认可信息及其对象格式及资金划账信息及其对象格式。



6.5.1 SET安全协议

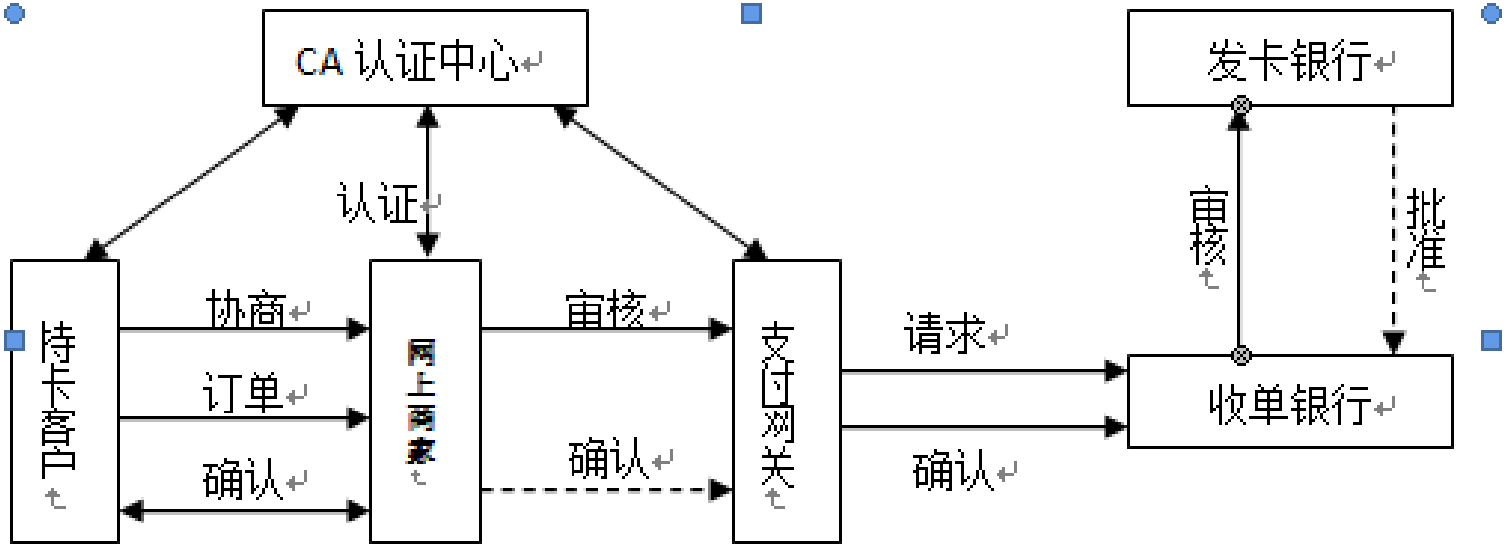
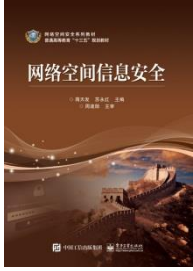
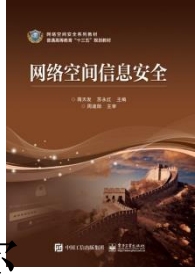


图 6.17 SET 协议的应用框架



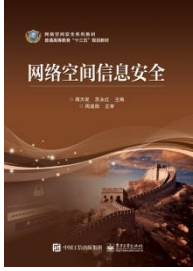
6.5.1 SET安全协议

- SET交易分成三个阶段进行：
  - 购买请求阶段：客户选择商品；客户发送初始请求；商家产生初始应答；客户对商家进行验证；客户提出购物请求。
  - 支付的认定阶段：商家产生支付请求；支付网关验证双方信息；银行产生支付应答。
  - 收款阶段：商家发出售物应答；验证商家证书；完成支付；商家收款。



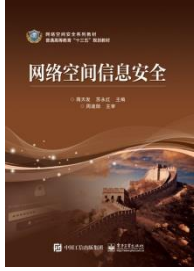
6.5.1 SET安全协议

- 从加密机制来看，SET中采用的公钥加密算法是RSA，私钥加密算法是DES。
- SET工作原理：持卡人将消息摘要用私钥加密得到数字签名。随机产生一对称密钥，用它对消息摘要、教字签名与证书(含客户的公钥)进行加密，组成加密信息，接着将这个对称密钥用商家的公钥加密得到数字信封；当商家收到客户传来的加密信息与数字信封后，用他的私钥解密数字信封得到对称密钥，再用它对加密信息解密，接着验证数字签名：用客户的公钥对数字签名解密，得到消息摘要，再与消息摘要对照；认证完毕，商家与客户即可用对称密钥对信息加密传送。



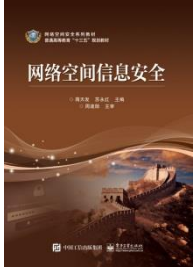
6.5.2 电子邮件安全协议

- 2015年，PGP 发明人 Phillip Zimmermann 和加密邮件服务Lavabit 的创始人 Ladar Levison 等人宣布了替代SMTP协议的Dark Internet Mail Environment(DIME)，通过有多层密钥管理和多层信息加密的核心模式使得元数据信息被限制泄露，邮件处理代理仅能访问它们需要看到的信息。
- 目前比较常用的是PGP技术，还有已经成为Internet标准的PEM技术。前者虽然被广泛使用，但还不是Internet的正式标准。



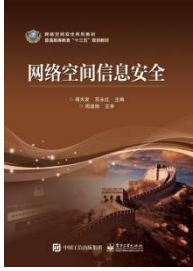
6.5.2 电子邮件安全协议

- PGP（Pretty Good Privacy）是Zimmermann于1995年开发出来的。它是一个完整的电子邮件安全软件包，包括加密、鉴别、电子签名和压缩等技术。PGP并没有使用新的概念，它只是将现有的一些算法如MD5、RSA以及IDEA等综合在一起。由于包括源程序的整个软件包可从Internet免费下载，因此PGP在MS DOS/Windows以及UNIX等平台上得到了广泛的应用。
- PGP安全电子邮件系统做为安全通信的加密标准,具有很高的安全性。一但加密後，讯息看起来是一堆无意义的乱码（Random Characters）。PGP提供了极强的保护功能，即使是最先进的解码分析技术也无法解 读（Decrypt）加密後的文字。



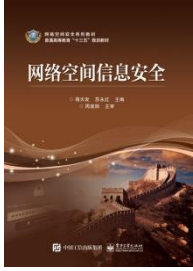
6.5.2 电子邮件安全协议

- PGP是一个基于RSA公钥加密体系的邮件加密软件。我们可以用它对邮件保密以防止非授权者阅读，它还能对用户的邮件加上数字签名，从而使收信人可以确信发信人的身份。它让用户可以安全地和从未见过的人们通信，事先并不需要任何保密措施的来传递密钥，因为它采用了非对称的“公钥”和“私钥”加密体系。



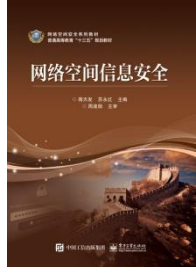
6.5.2 电子邮件安全协议

- PGP把RSA公钥体系的方便和传统加密体系高度结合起来。PGP不是一种完全的非对称加密体系，它是个混合加密算法，它是由一个对称加密算法(IDEA)、一个非对称加密算法(RSA)、一个单向散列算法(MD5)以及一个随机数产生器(从用户击键频率产生伪随机数序列的种子)组成的，每种算法都是PGP不可分割的组成部分，PGP集中了算法的优点。并且在数字签名和密钥认证管理机制上有巧妙的设计。



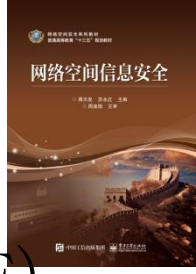
6.5.2 电子邮件安全协议

- 传统加密方法就如DES是用一个密钥加密明文，然后用同样的密钥解密。而PGP是以一个随机生成的密钥，用IDEA算法对明文加密，然后用RSA算法对该密钥加密。这样收件人同样是用RSA解出这个随机密钥，再用IDEA解密邮件本身。这样的链式加密就做到了既有RSA体系的保密性，又有IDEA算法的快捷性。创始人Philip Zimmermann找到了公钥和对称加密算法的均衡点。



6.5.2 电子邮件安全协议

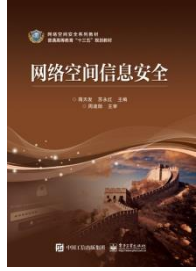
- PGP给邮件加密和签名的过程是这样的:首先发送者用自己的私钥将128位值（二进制数，由MD5算法产生）加密，附加在邮件后，再用接收者的公钥将整个邮件加密。这份密文被接收者收到以后，接收者用自己的私钥将邮件解密，得到发送者的原文和签名，接收者的PGP也从原文计算出一个128位的特征值来和用发送者的公钥解密签名所得到的数进行比较，如果符合就说明这份邮件确实是发送者寄来的。这样两个安全性要求都得到了满足。



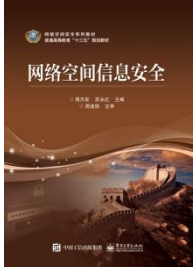
6.5.2 电子邮件安全协议

- 到1993年初，Internet工程特别工作组(IETF)以及Internet研究特别工作组(IRTF)已提出四份RFC(Requests For Comments)作为建议的标准，其编号为1421～1424。这些RFC定义了PEM的保密功能以及相关的管理问题。PEM的基本原理如下：
- 各用户的用户代理(User Agent-CA)配有PEM软件。CA提出PEM用户证件的注册申请(按照X.509协议)。用户的证件被存储在一个可公开访问的数据库之中，该数据库提供一种基于X.500的目录服务。密钥等机密信息则存储在用户的个人环境(Personal SecureEnvironment—PSE)中。用户使用本地PEM软件以及PSE环境信息生成PEM邮件，然后通过基于SMTP的报文传递代理(MTA)发给对方。接收方在自身的PSE中将报文解密，并通过目录检索其证件，查阅证件注销表以核实证件的有效性。

6.5.2 电子邮件安全协议



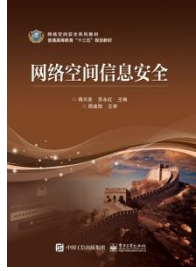
- PEM 提供以下四种安全服务：
  - 数据隐蔽:，
  - 数据完整性:，
  - 对发送方的鉴别，
  - 防发送方否认。
- 以使用非对称密码为例，安全服务的具体实现如下：



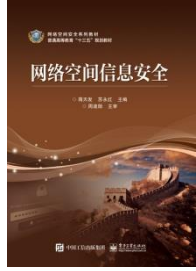
6.5.2 电子邮件安全协议

- 数据隐蔽:首先随机生成一个DES密钥,然后用接收方的公开钥对DES密钥采用非对称加密算法加密后,存放在PEM报文的头部。接收方收到此报文后,用其秘密钥对DES密钥解密,接着再用此DES密钥对报文解密即可。
- 数据完整性和对发送方的鉴别:用数字签名完成。首先,对准备传送的报文用MD2或MD5算法生成一个MIC码。然后，MIC码可以用发送者的密钥"解密"。然后存放在PEM邮件的头部。接收方可用发送方的公开密钥译出报文的MIC。最后，用此MIC与接收方用收到的报文实时生成的MIC比较后,即可断定报文的完整性,并完成对发送方的鉴别。

6.5.2 电子邮件安全协议



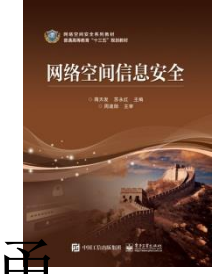
- 防发送方否认:此项功能亦在上述过程中自动实现。只有用发送方的密钥加过密的MIC,才能经接收方解密后与当时生成的MIC相匹配。发送方发送此报文是不容抵赖的。



6.5.2 电子邮件安全协议

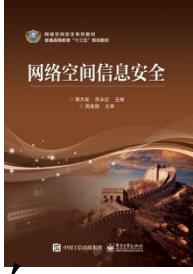
PEM的加密过程通常包括四个步骤：

- 报文生成：一般使用用户所常用的格式。
- 规范化：转换成SMTP的内部表示形式。
- 加密：执行选用的密码算法。
- 编码：对加密后的报文进行编码以便传输。



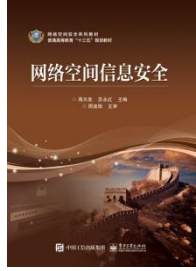
6.5.2 电子邮件安全协议

- 用户的证件是用户在网上使用PEM的通行证。每个证件除包含公开钥外,还含有用户的唯一名、证件的有效期、证件编号以及证件管理机构的签名等。证件的管理由证明机构CA (Certification Authcrity)完成。证件的结构和管理均在X.509"The Directory-Authentication Framework"中定义。



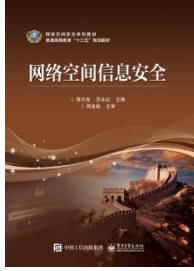
6.5.2 电子邮件安全协议

- 网上用户想使用PEM,要先行注册。用户应向本地CA发"证明申请",填写证件内容并鉴上名。本地CA审查同意后赋予证件有效期和流水编号,同时用CA的秘密钥签名,之后证件生效。
- 存放证件的数据库,其分布式结构由X.500协议定义。其他网上用户可从中取用发送方的公开钥以及核实邮件发送证件的有效期。而注销后的证件存放在证件注销表(Certificate Revocation List—CRL)中,供查对用。核实工作由PEM软件自动进行,其结果通知接收方。



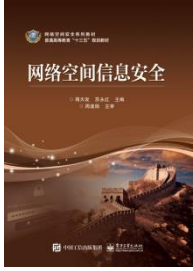
6.5.2 电子邮件安全协议

- S/MIME (Secure/Multipurpose Internet Mail Extensions) 设计用来支持邮件的加密。基于 MIME 标准， S/MIME 为电子消息应用程序提供如下加密安全服务：认证、完整性保护、鉴定及数据保密等。传统的邮件用户代理（MUA）可以使用S/MIME来加密发送邮件及解密接收邮件。

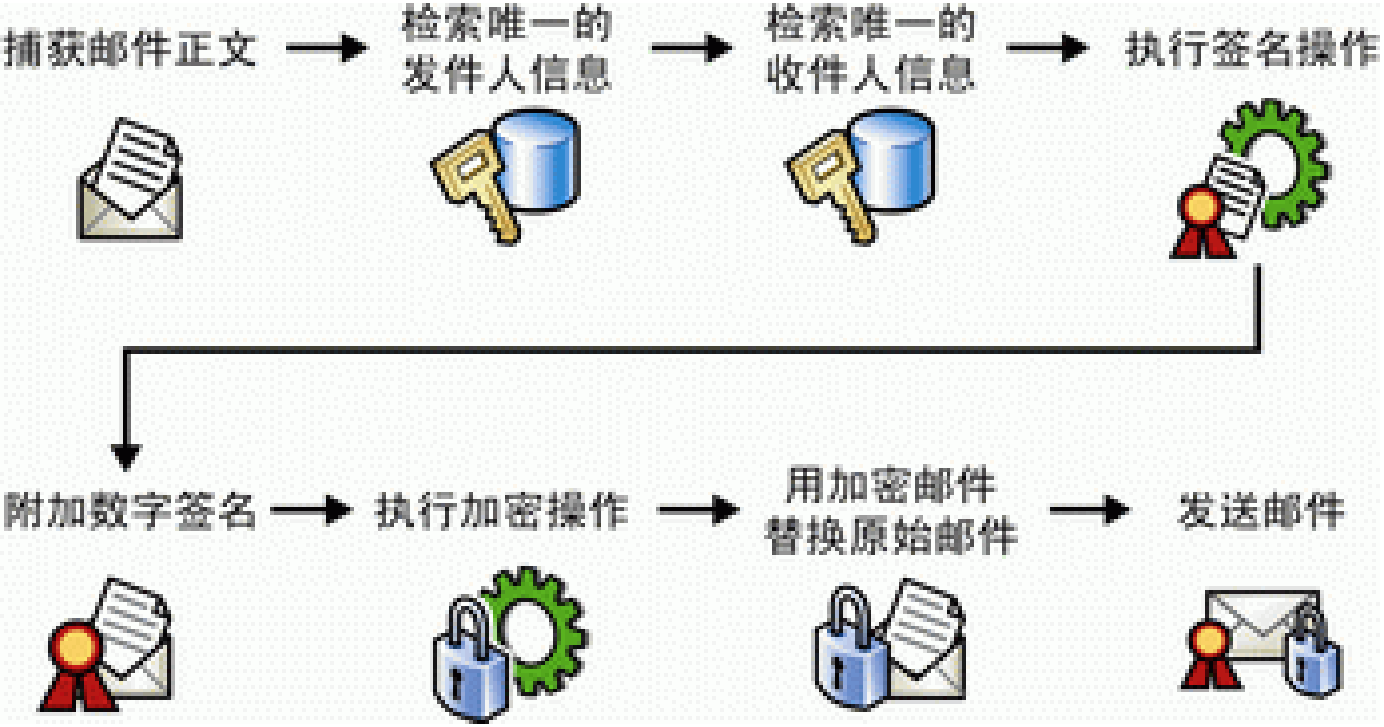


6.5.2 电子邮件安全协议

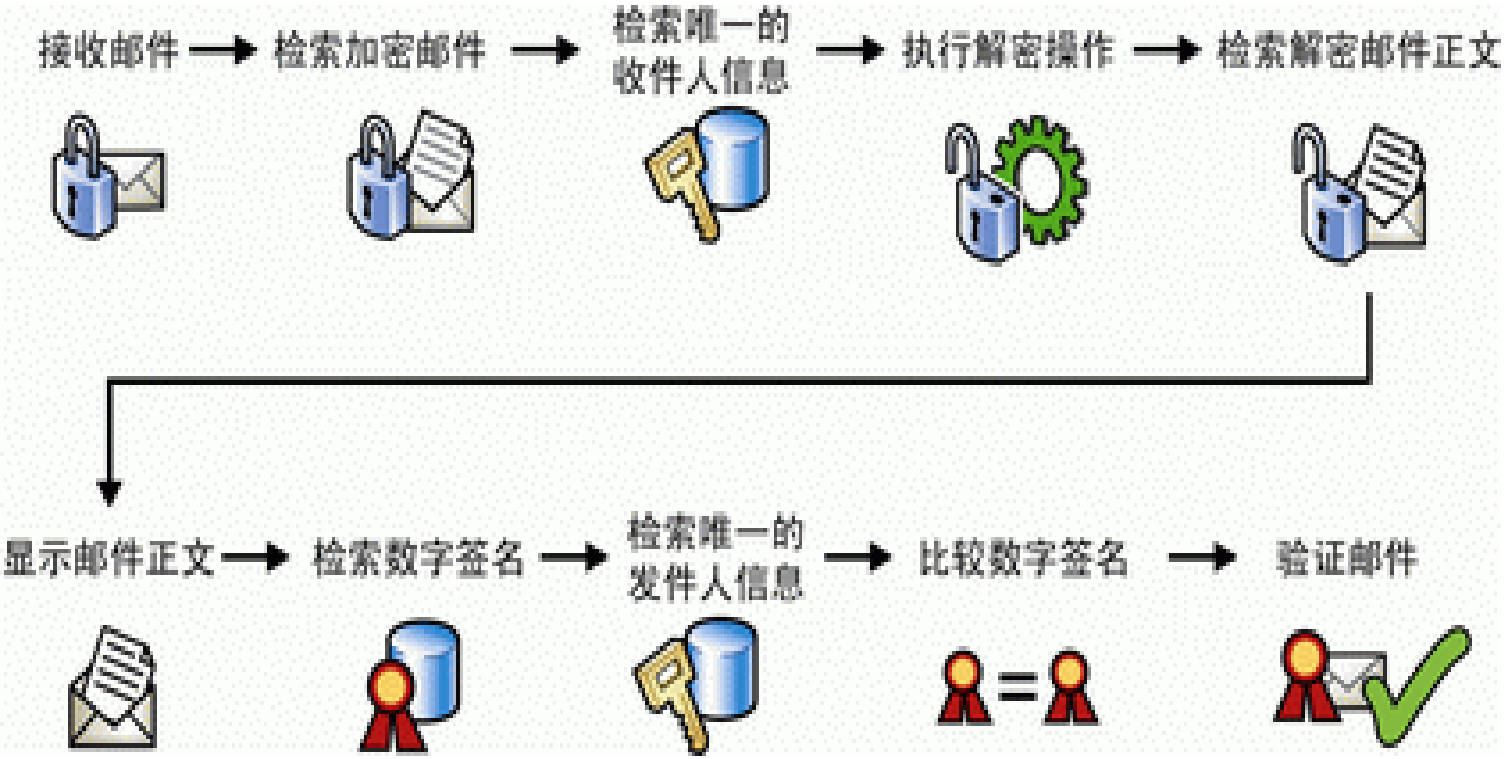
- S/MIME 提供两种安全服务：数字签名和邮件加密。数字签名和邮件加密并不是相互排斥的服务。数字签名解决身份验证和认可问题，而邮件加密则解决保密性问题。邮件安全策略通常同时需要这两个服务。这两个服务被设计为一起使用，因为它们分别针对发件人和收件人关系的某一方。

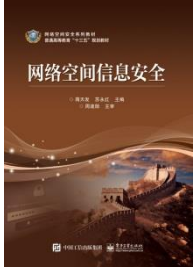


对电子邮件进行签名和加密的顺序



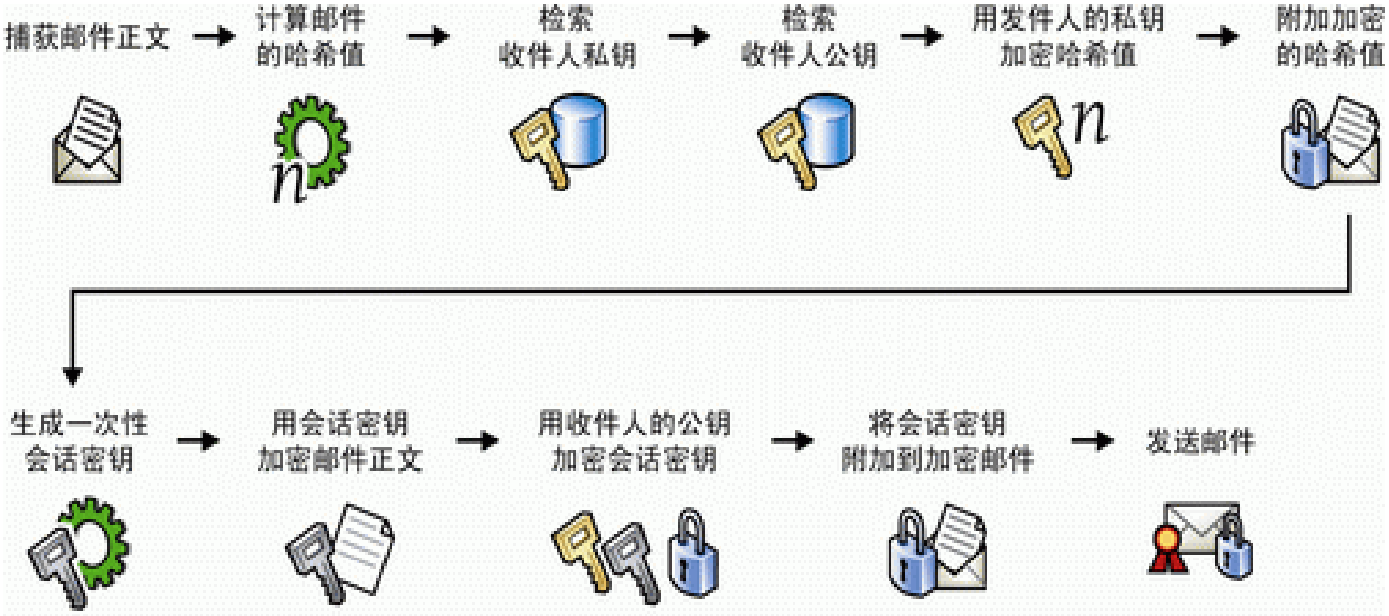
对电子邮件进行解密和验证的顺序

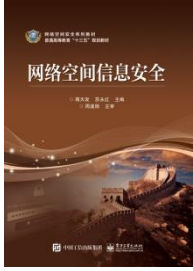




6.5.2 电子邮件安全协议

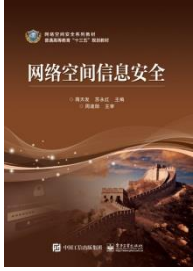
- 数字证书和邮件加密是 S/MIME 的核心功能。在邮件安全领域，最重要的支持性概念是公钥加密。





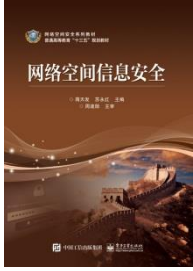
6.5.3 安全外壳协议

- SSH为SecureShell（安全套接层）的缩写，由IETF的网络工作小组（NetworkWorkingGroup）所制定，是专为远程登录会话和其他网络服务提供安全性的协议。利用SSH协议可以有效防止远程管理过程中的信息泄露问题。SSH最初是UNIX系统上的一个程序，后来又迅速扩展到其他操作平台：几乎所有UNIX平台—包括HP-UX、Linux、AIX、Solaris、DigitalUNIX、Irix，以及其他平台，都可运行SSH。



6.5.3 安全外壳协议

- Ftp、 pop和 telnet在本质上都是不安全的，容易受到“ 中间人”（man-in-the-middle）这种方式的攻击。SSH可以把所有传输的数据进行加密，防止”中间人”攻击方，而且也能够防止 DNS欺骗和 IP欺骗。SSH还有一个额外的好处就是传输的数据是经过压缩的，传输的速度可以加快。



6.5.3 安全外壳协议

- SSH 主要由三部分组成：

–传输层协议[SSH-TRANS]，该协议提供了服务器认证，保密性及完整性，有时还提供压缩功能。 SSH-TRANS 通常运行在 TCP/IP连接上，也可能用于其它可靠数据流上。该协议中的认证基于主机，不执行用户认证。

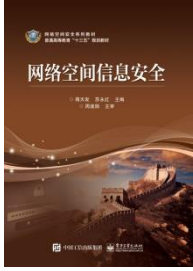
–用户认证协议 [SSH-USERAUTH]

运行在 传输层协议上面，用于向服务器提供客户端用户鉴别功能。用户认证协议需要知道低层协议是否提供保密性保护，当用户认证协议开始后，它从低层协议那里接收会话标识符。

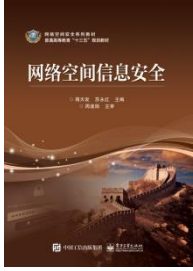
–连接协议 [SSH-CONNECT]

将多个加密隧道分成逻辑通道。它运行在用户认证协议上。它提供了交互式登录话路、远程 命令执行、转发 TCP/IP 连接和转发 X11 连接。

6.5.4 安全超文本转换协议

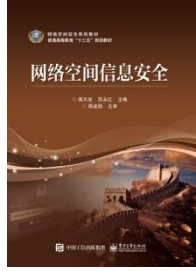


- SHTTP由EIT公司开发，主要目的保证商业贸易的传输安全。版本：1.0，1.2，1.3，1.4. 是一种面向安全信息通信的协议，它可以和 HTTP 结合起来使用。S-HTTP 能与 HTTP 信息模型共存并易于与 HTTP 应用程序相整合。
- S-HTTP还为客户机和服务器提供了对称能力（及时处理请求和恢复，及两者的参数选择），维持 HTTP 的通信模型和实施特征。S-HTTP 客户机和服务器是与某些加密消息格式标准相结合的。



6.5.4 安全超文本转换协议

- S-HTTP支持多种兼容方案并且与 HTTP 相兼容。有 S-HTTP 性能的客户机能够与没有 S-HTTP 的服务器连接，但是这样的通讯明显地不会利用 S-HTTP安全特征。
- S-HTTP 不需要客户端公用密钥认证（或公用密钥），但它支持对称密钥的操作模式。这意味着即使没有要求用户拥有公用密钥，私人交易也会发生。S-HTTP 支持端对端安全事务通信。客户机可能“首先”启动安全传输（使用报头的信息），它可以用来支持已填表单的加密。使用 S-HTTP，敏感的数据信息不会以明文形式在网络上发送。



6.5.4 安全超文本转换协议

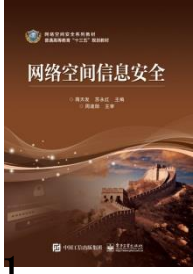
S-HTTP 提供了完整且灵活的加密算法、模态及相关参数。通过以下四个方面来提供安全服务：

–1）签名。应用了数字签名后，消息附上适当的认证信息，接受者能进行认证。签名使用CMS中的signeddata类型

–2）加密和密钥交换。一种是公钥封装（CMS和MOSS）。一种是预先准备好的密钥（密钥加密是以CMS信封的方式实现；预先准备好的密钥使用CMS的encrypterdata数据类型。）。

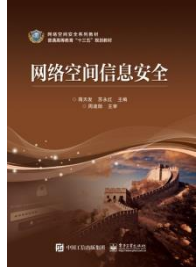
–3）消息完整性和发送者的身份认证。通过计算消息码来验证。

–4）实时性、简单的竞争响应机制，保证事务的实时性。



6.5.5 网络验证协议

- Kerberos是一种网络认证协议（Network Authentication Protocol），这一名词来源于希腊神话“三个头的狗——地狱之门守护者”。其设计目标是通过密钥系统DES为客户机/服务器应用程序提供强大的认证服务。该认证过程的实现不依赖于主机操作系统的认证，无需基于主机地址的信任，不要求网络上所有主机的物理安全，并假定网络上传送的数据包可以被任意地读取、修改和插入数据。可以用于防止窃听、防止replay攻击、保护数据完整性等场合，是一种应用对称密钥体制进行密钥管理的系统。

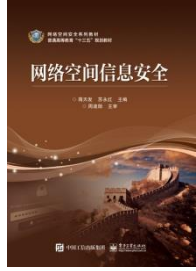


6.5.5 网络验证协议

- Kerberos认证过程具体如下：客户机向认证服务器（AS）发送请求，要求得到某服务器的证书，然后 AS 的响应包含这些用客户端密钥加密的证书。证书的构成为：

  1）服务器 “ticket” ；

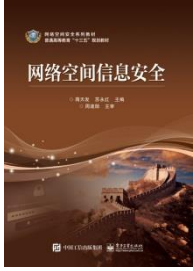
  2）一个临时加密密钥 （又称为会话密钥 “session key”）



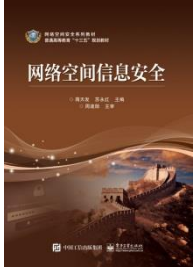
6.5.5 网络验证协议

- Windows2000和后续的操作系统都默认Kerberos为其默认认证方法。RFC 3244记录整理了微软的一些对Kerberos协议软件包的添加。RFC4757"微软Windows2000Kerberos修改密码并设定密码协议"记录整理了微软用RC4密码的使用。苹果的Mac OS X也使用了Kerberos的客户和服务器版本。Red Hat Enterprise Linux4 和后续的操作系统使用了Kerberos的客户和服务器版本。
- Kerberos由以下部分组成：

6.5.5 网络验证协议



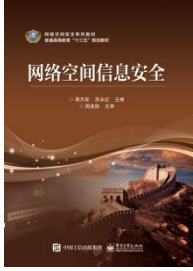
- Kerberos应用程序库：应用程序接口。包括创建和读取认证请求、创建safe message 和private message的子程序。
- 加密/解密库：DES等。
- Kerberos数据库：记载了每个Kerberos 用户的名字、私有密钥、截止信息(记录的有效时间，通常为几年)等信息。
- 数据库管理程序：管理Kerberos数据库。



6.5.5 网络验证协议

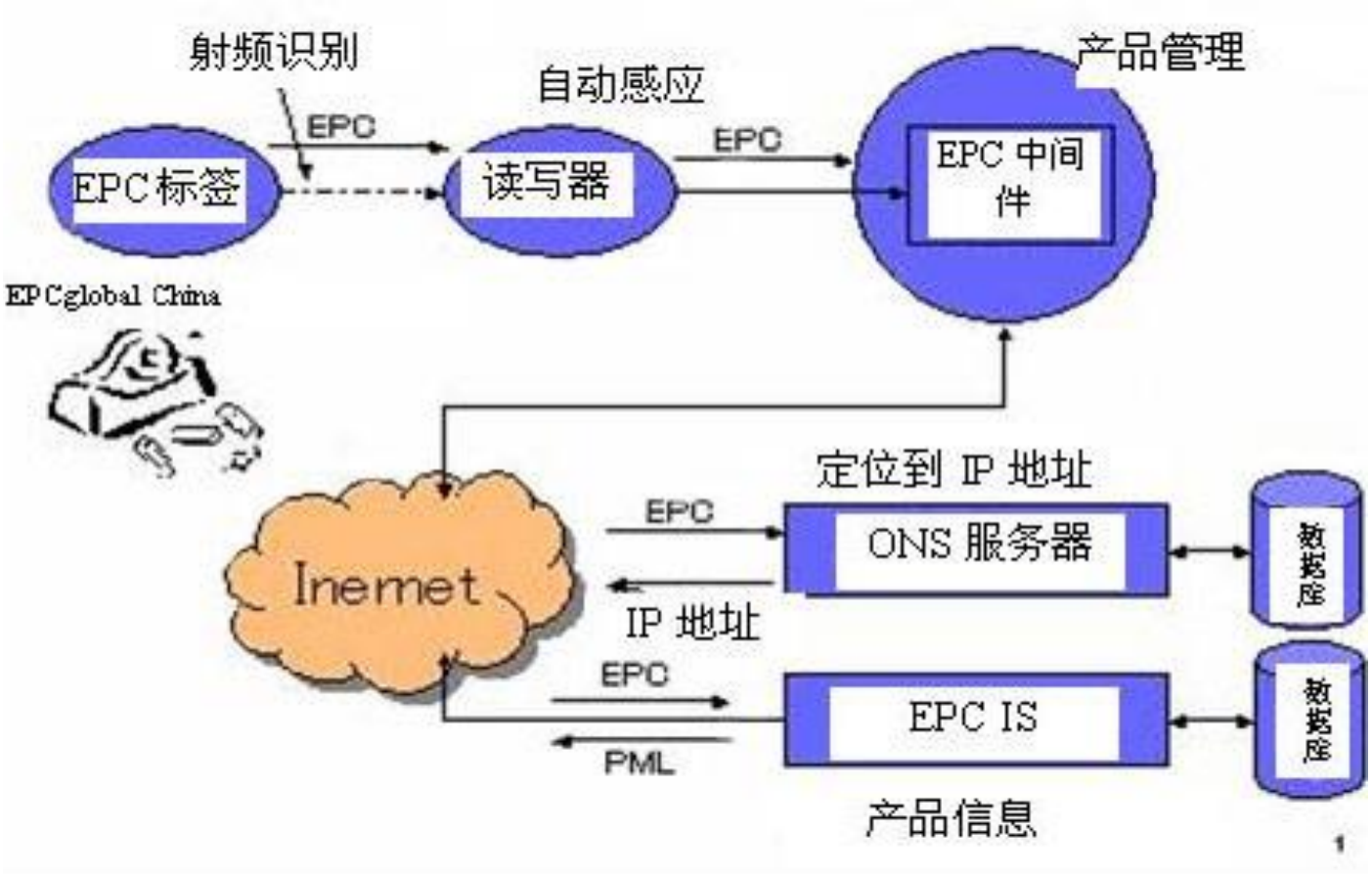
- KDBM服务器(数据库管理服务器)：接受客户端的请求对数据库进行操作。
- 认证服务器(AS)：存放一个Kerberos数据库的只读的副本，用来完成principle的认证，并生成会话密钥.
- 数据库复制软件：管理从KDBM服务所在的机器，到认证服务器所在的机器的数据库复制工作，为了保持数据库的一致性，每隔一段时间就需要进行复制工作.
- 用户程序：登录Kerberos，改变Kerberos密码，显示和破坏Kerberos标签（ticket）等工作。

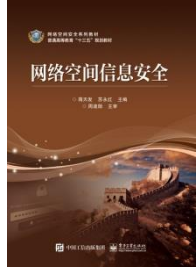
6.6 EPC的密码机制和安全协议



- 产品电子代码（EPC编码）是国际条码组织推出的新一代产品编码体系。原来的产品条码仅是对产品分类的编码，EPC码是对每个单品都赋予一个全球唯一编码，EPC编码96位（二进制）方式的编码体系。96位的EPC码，可以为2.68亿公司赋码，每个公司可以有1600万产品分类，每类产品有680亿的独立产品编码，形象的说可以为地球上的每一粒大米赋一个唯一的编码。
- EPC的载体是RFID电子标签，并借助互联网来实现信息的传递。EPC旨在为每一件单品建立全球的、开放的标识标准，实现全球范围内对单件产品的跟踪与追溯，从而有效提高供应链管理水平、降低物流成本。EPC是一个完整的、复杂的、综合的系统。

6.6 EPC的密码机制和安全协议

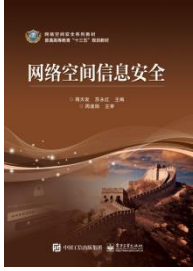




6.6 EPC的密码机制和安全协议

- 读写器读出的EPC只是一个信息参考（指针），由这个信息参考从INTERNET找到IP地址并获取该地址中存放的相关的物品信息，并采用分布式的EPC中间件处理由读写器读取的一连串EPC信息。由于在标签上只有一个EPC代码，计算机需要知道与该EPC匹配的其它信息，这就需要ONS来提供一种自动化的网络数据库服务，EPC中间件将EPC代码传给ONS，ONS指示EPC中间件到一个保存着产品文件的服务器（EPC IS）查找，该文件可由EPC中间件复制，因而文件中的产品信息就能传到供应链上

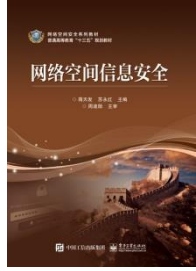
6.6 EPC的密码机制和安全协议



- EPC系统的信息网络系统是在全球互联网的基础上，通过EPC中间件、对象命名称解析服务（ONS）和EPC信息服务（EPC IS）来实现全球“实物互联”。
- 1）EPC中间件

  EPC中间件是加工和处理来自读写器的所有信息和事件流的软件，是连接读写器和企业应用程序的纽带，主要任务是在将数据送往企业应用程序之前进行标签数据校对、读写器协调、数据传送、数据存储和任务管理。

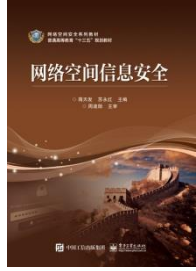
6.6 EPC的密码机制和安全协议



- 2）对象名称解析服务(ONS)

对象名称解析服务(ONS) 是一个自动的网络服务系统，类似于域名解析服务(DNS)， ONS给EPC中间件指明了存储产品相关信息的服务器。

ONS服务是联系EPC中间件和EPC信息服务的网络枢纽，并且ONS设计与架构都以因特网域名解析服务DNS为基础，因此，可以使整个EPC网络以因特网为依托，迅速架构并顺利延伸到世界各地。



6.6 EPC的密码机制和安全协议

- 3）EPC信息服务(EPC IS)

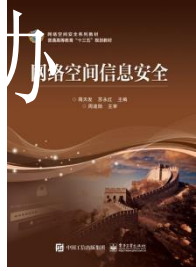
EPC IS提供了一个模块化、可扩展的数据和服务的接口，使得EPC的相关数据可以在企业内部或者企业之间共享。它处理与EPC相关的各种信息，例如：

– EPC的观测值：What / When / Where / Why，通俗的说，就是观测对象、时间、地点以及原因，这里的原因是一个比较泛的说法，它应该是EPC IS步骤与商业流程步骤之间的一个关联，例如订单号、制造商编号等商业交易信息。

– 包装状态：例如：物品是在托盘上的包装箱内。

– 信息源：例如：位于Z仓库的Y通道的X识读器。

6.6 EPC的密码机制和安全协议



- EPC标签作为EPC系统的一部分，包含了很多重要的信息，这些隐私容易受到威胁：

  1）行为威胁

  容易根据一组标签的行踪而获取一个人的行为。

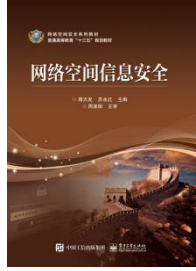
  2）关联威胁

  在购买携带EPC标签的物品时，可将用户的身份和该物品的电子序列号相关联，这类关联可能是秘密的，也可能是无意的。

  3）位置威胁

  携带标签的位置易于未经授权的被暴露，携带标签的个人行踪可能被监控。

6.6 EPC的密码机制和安全协议



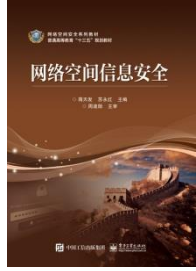
4）喜好威胁

利用EPC网络，物品上的标签可以唯一的识别生产者、产品类型和物品的唯一身份。竞争者可以以非常低廉的成本获得宝贵的用户喜好信息。

5）事务威胁

当携带标签的对象从一个星座转移到另一个星座时，在与这些星座关联的个人之间可以很容易的推导出正在发生的事务

6.6 EPC的密码机制和安全协议



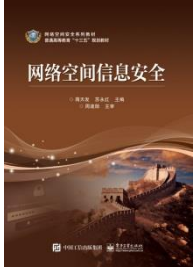
6）星座威胁

多个标签可在一个人的周围形成一个唯一的星座，对手可使用这个特殊的星座实施跟踪。

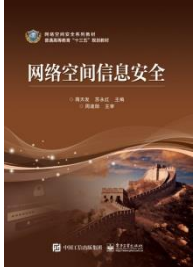
7）面包屑威胁

从个人收集携带标签的物品，然后，在公司信息系统中建立一个与他们的身份关联的物品数据库。丢弃标签不会丢失这种关联，适用这些丢失的“面包屑”可实施犯罪或某些恶意行为。

6.6 EPC的密码机制和安全协议

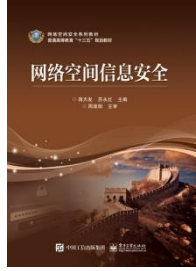


- 基于密码技术的软件安全机制受到人们更多的青睐:其主要研究内容是利用各种成熟的密码方案和机制来设计和实现RFID安全需求的密码协议。这已经成为当前RFID研究的热点。目前，已经提出了多种RFID安全协议



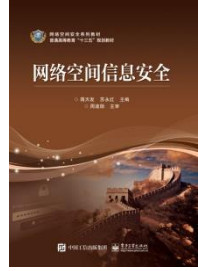
本章小结

- 本章阐述了网络安全协议的概念、分类和类型：网络层的安全协议IPSec，传输层的安全协议SSL/TLS，应用层的安全协议SHTTP（Web安全协议）、PGP（电子邮件安全协议），S/MIME（电子邮件安全协议）、PEM（电子邮件安全协议）、SSH（远程登录安全协议），Kerberos（网络验证协议）等。
- 本章重点论述了IPSec、SSL/TLS和应用层的相关安全协议，并介绍了面向用户的IPSec安全隧道的构建技术、电子交易中SET协议的应用框架和SET电子支付流程。安全电子交易SET协议为信用卡交易提供了安全，可以实现电子商务交易中的机密性、验证性、数据完整性等安全功能。



思考题

- 6.1　网络安全协议的应用领域是什么？
- 6.2　安全协议一般使用哪些基础技术？
- 6.3　阐述目前网络上使用了哪几种网络安全协议？
- 6.4　SSL协议有什么优缺点？
- 6.5　SET协议可否取代SSL协议？为什么？
- 6.6　IPSec协议在IPv4和IPv6使用中有什么不同？
- 6.7　IPSec的主要组成组件有哪几部分？各有什么功能？
- 6.8　叙述IPSec安全隧道的构建步骤。
- 6.9　SET交易涉及哪6个实体？各起什么作用？
- 6.10　电子邮件安全协议有哪些？
- 6.11　SSH协议由哪几部分组成？各个部分具有什么功能？



- 谢谢！