网络安全——

虚拟专用网

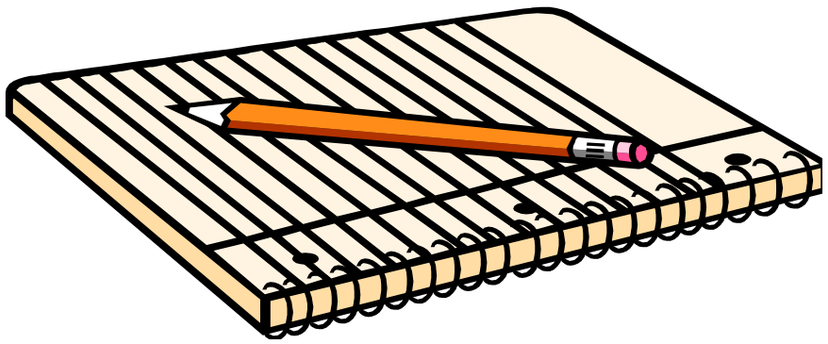
北京邮电大学

郑康锋

zkfbupt@163.com

本次课程内容（VPN）

- **VPN概念**
- **链路层安全**
- **网络层安全**
- **传输层安全**



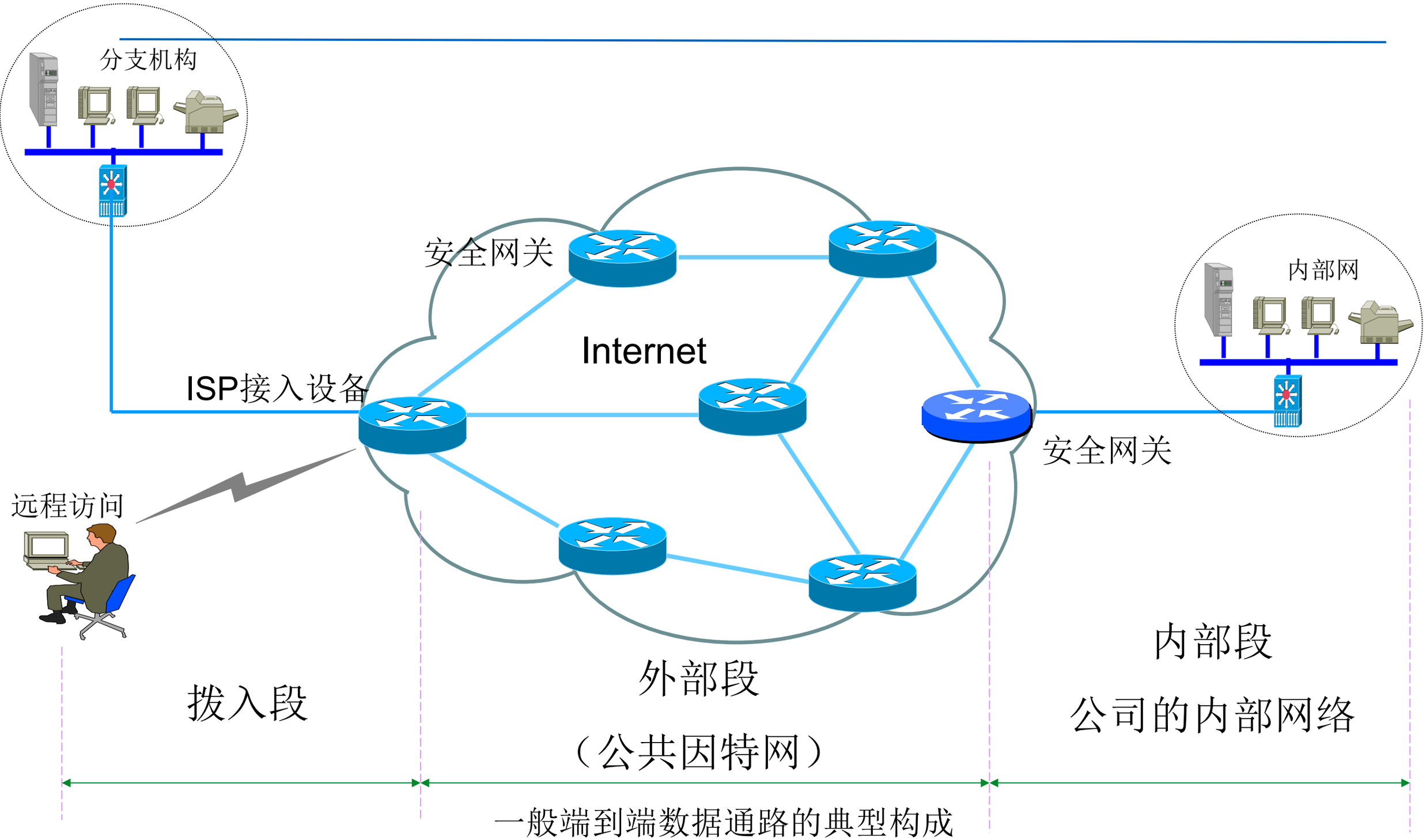
网络通信安全

- TCP/IP不同层次的安全机制
  - 链路层：PPTP、L2F、L2TP
  - 网络层：IPSec
  - 传输层：SSL
  - 应用层：SHTTP、S/MIME、SET
- 上述协议能够提供以下安全服务
  - 身份认证
  - 通信数据的保密性保护
  - 通信数据的完整性保护

虚拟专用网——

VPN概念

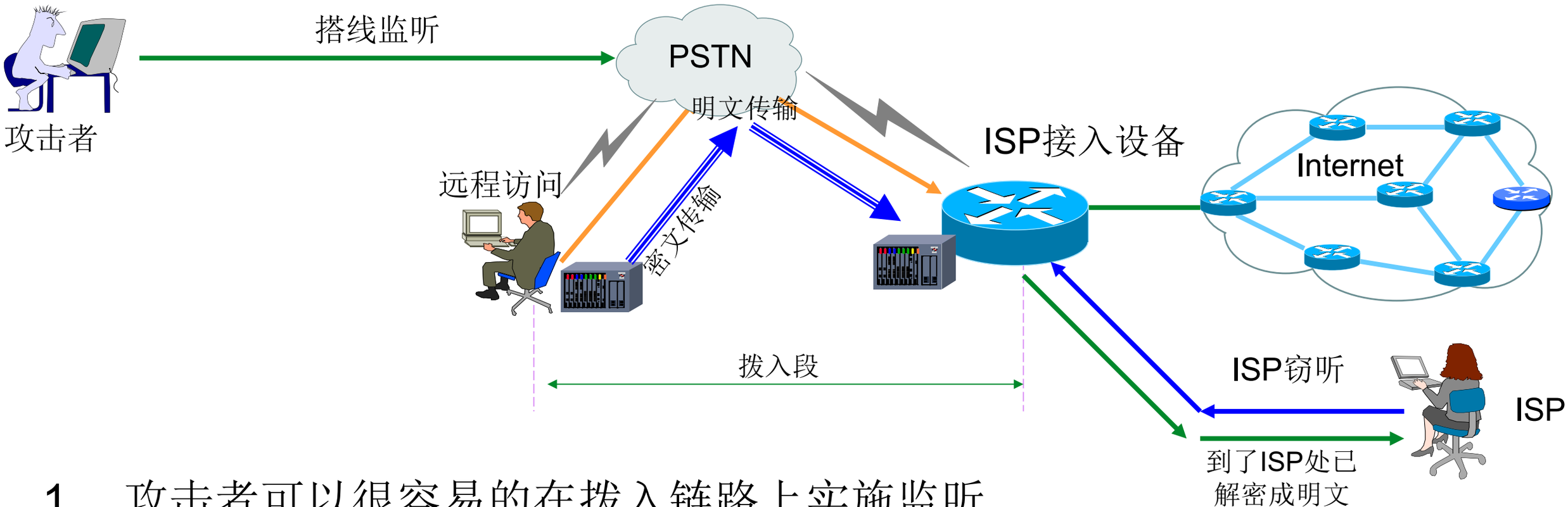
VPN提出----端到端数据安全性



一般端到端数据通路的典型构成

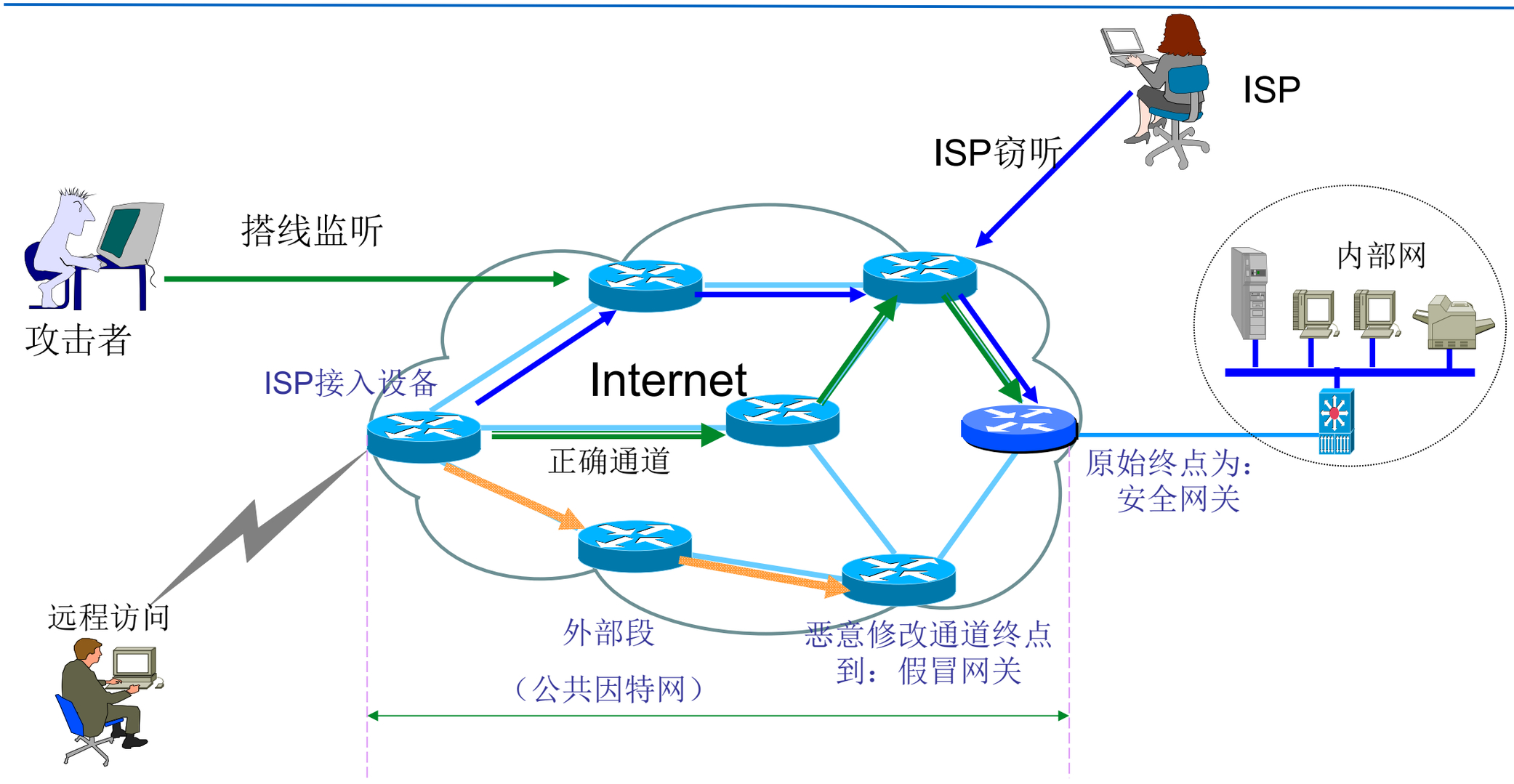
端到端数据通路安全风险(1)

拨入段用户数据以明文方式直接传递到ISP：



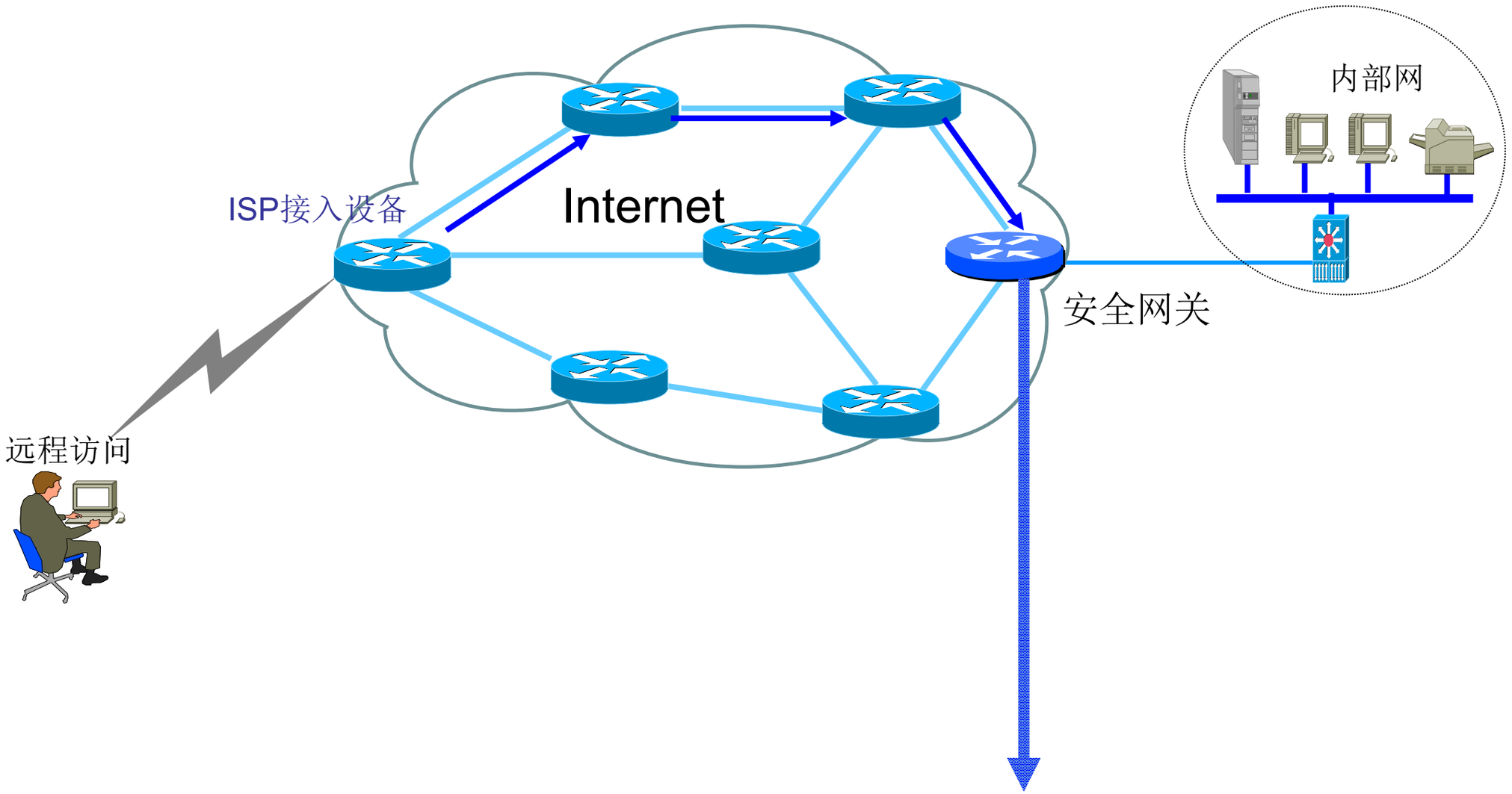
1. 攻击者可以很容易的在拨入链路上实施监听
2. ISP很容易检查用户的数据
3. 可以通过链路加密来防止被动的监听，但无法防范恶意窃取数据的ISP。

端到端数据安全性(2)



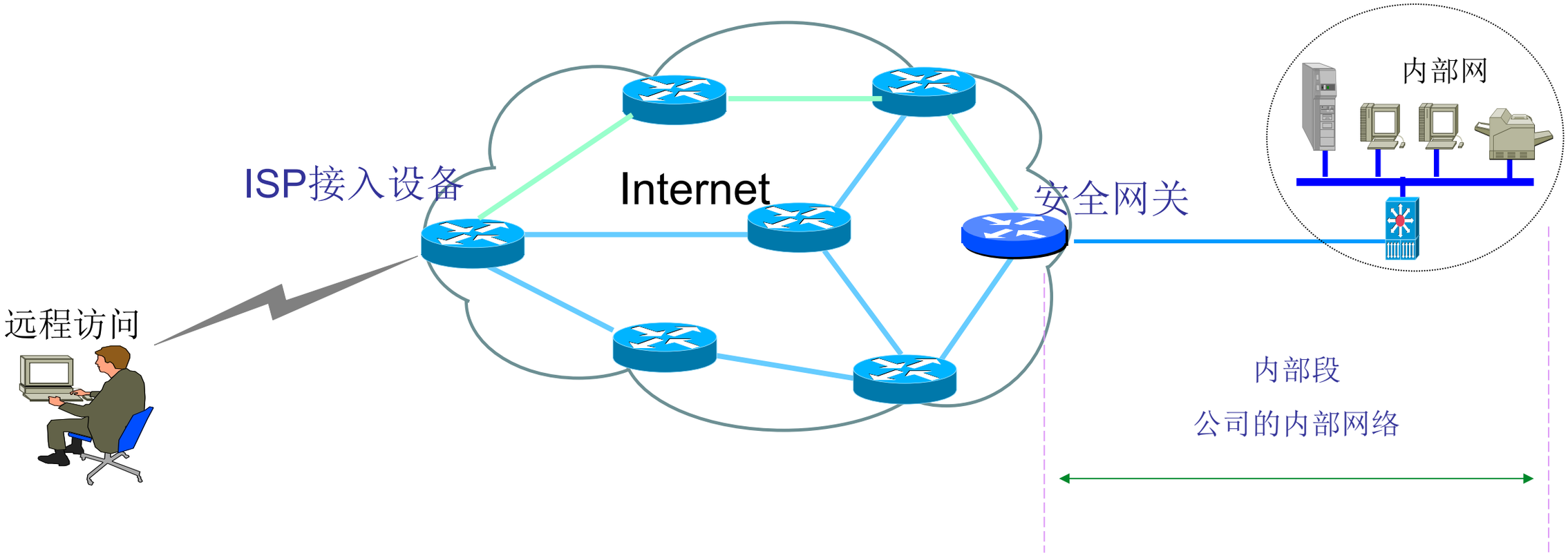
1. 数据在到达终点之前要经过许多路由器，明文传输的报文很容易在路由器上被查看和修改
2. 监听者可以在其中任一段链路上监听数据
3. 逐段加密不能防范在路由器上查看报文，因为路由器需要解密报文选择路由信息，然后再重新加密发送
4. 恶意的ISP可以修改通道的终点到一台假冒的网关

端到端数据安全性(3)



1. 数据在安全网关中是明文的，因而网关管理员可以直接查看机密数据
2. 网关本身可能会受到攻击，一旦被攻破，流经安全网关的数据将面临风险

端到端数据安全性(4)



1. 内部网中可能存在不信任的主机、路由器等
2. 内部员工可以监听、篡改、重定向企业内部网的数据报文
3. 来自企业网内部员工的其他攻击方式

VPN要解决的问题

- 在端到端的数据通路上随处都有可能发生数据的泄漏，包括
  - 拨入段链路上
  - ISP接入设备上
  - 在因特网上
  - 在安全网关上
  - 在企业内部网上。

能否提供一个综合一致的解决方案，它不仅能提供端到端的数据保护，同时也能提供逐段的数据保护呢？

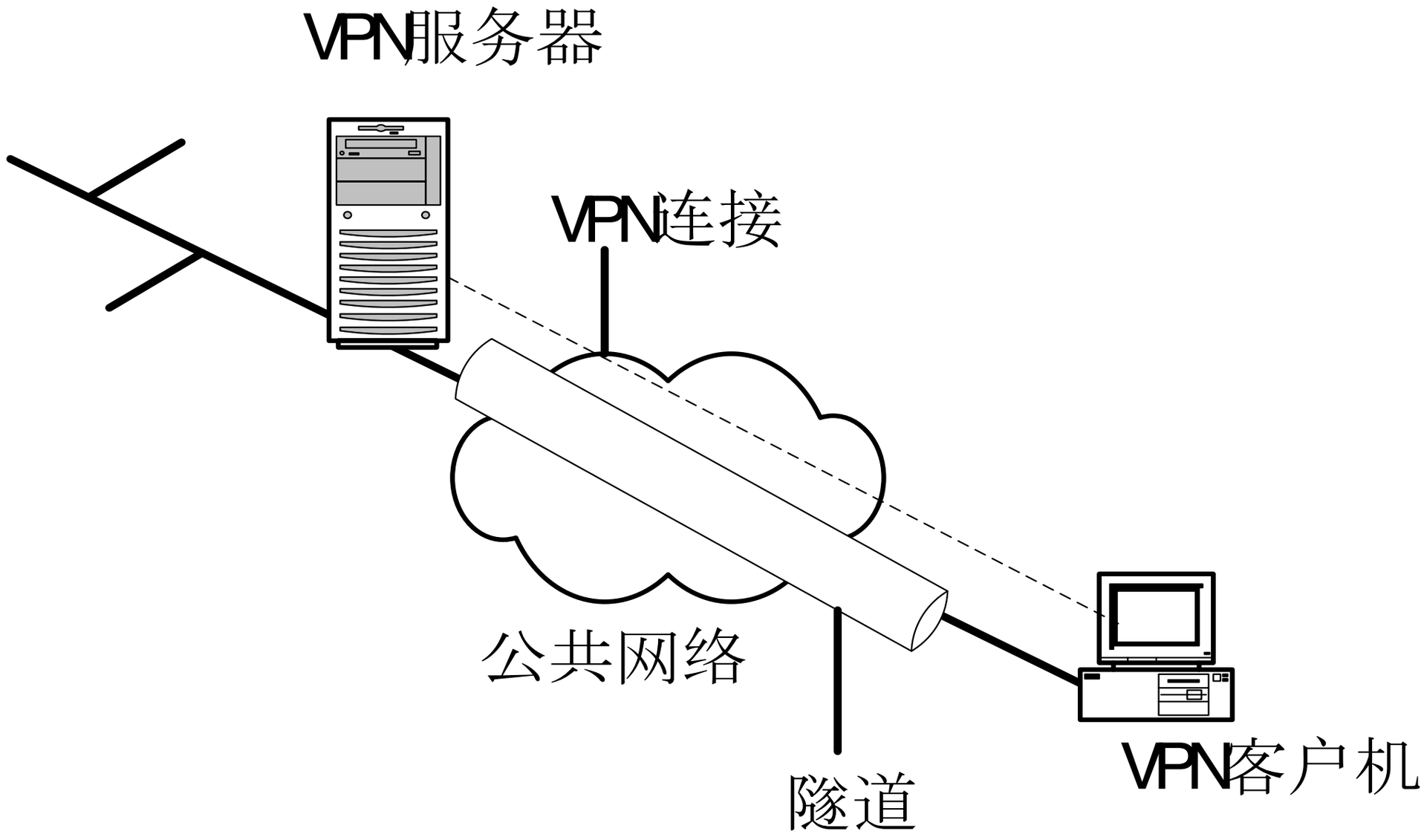
VPN

- **VPN：virtual private network，虚拟专用网**

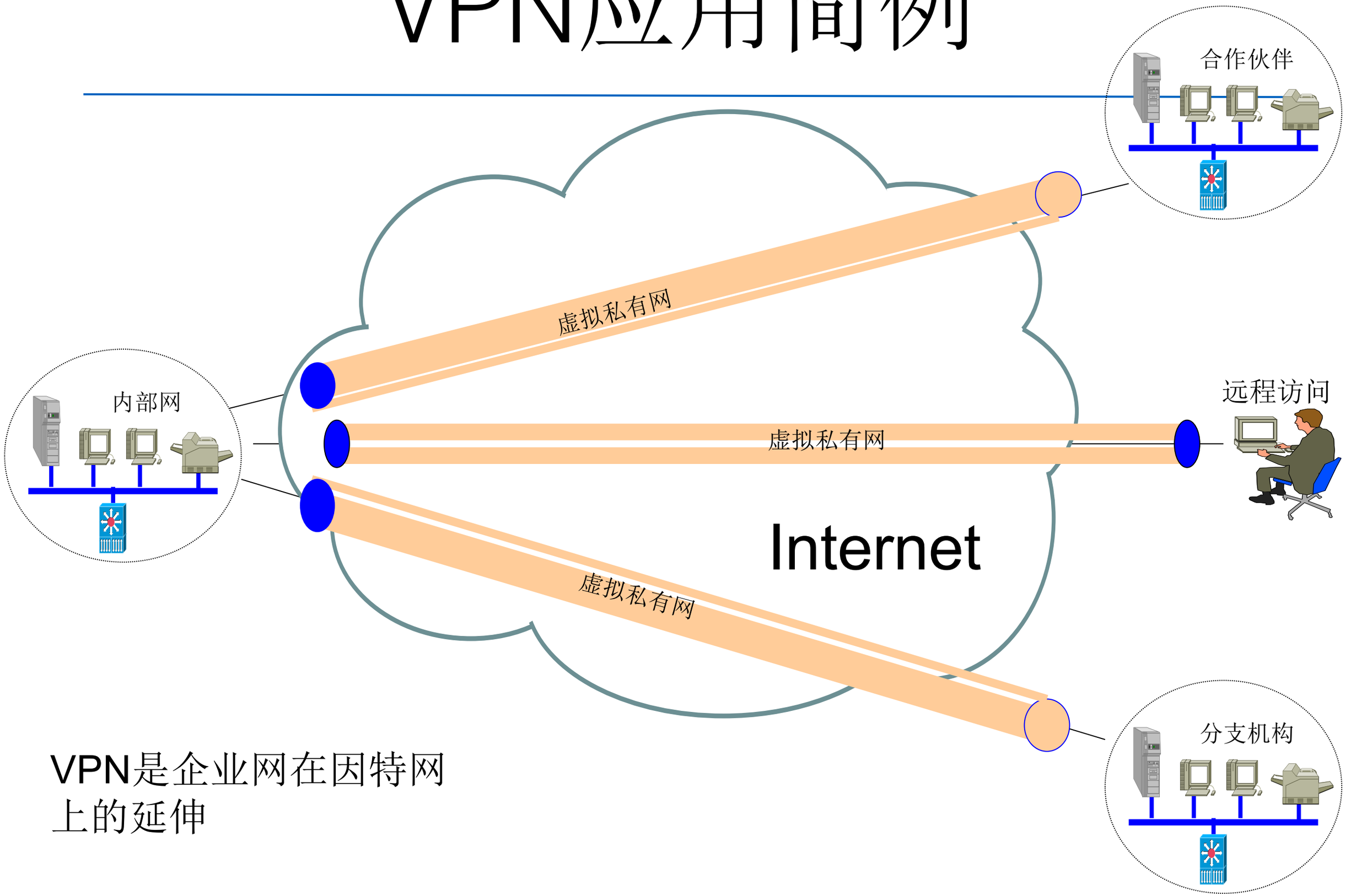
- VPN的定义：是指依靠ISP或其他NSP在公用网络基础设施之上构建的专用的数据通信网络，这里所指的公用网络有多种，包括IP网络、帧中继网络和ATM网络。
  - 虚拟
  - 专用网：封闭的用户群、安全性高、服务质量保证

- IETF对基于IP的VPN定义：使用IP机制仿真出一个私有的广域网

VPN的构成



VPN应用简例



VPN是企业网在因特网上的延伸

VPN的特点

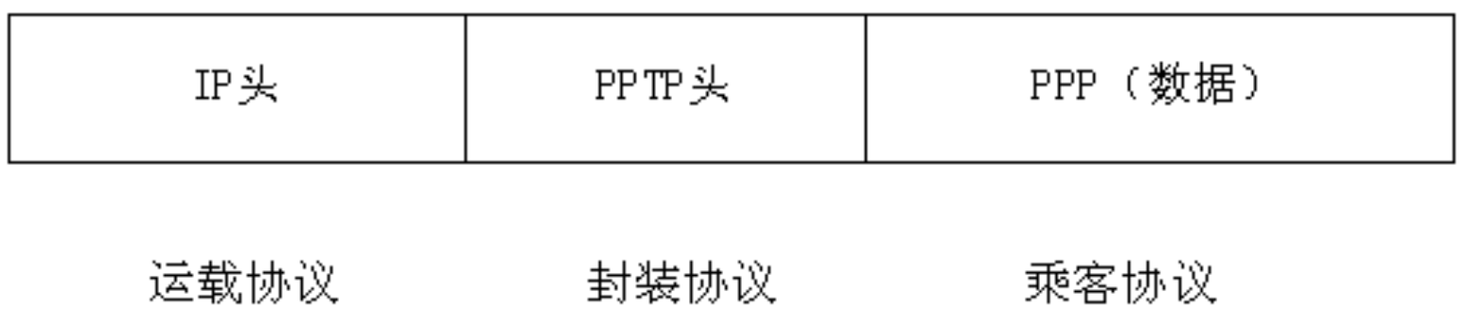
- 专用网的特点：
  - 封闭的用户群
  - 安全性高
  - 服务质量保证
- VPN的实现要求
  - 支持数据分组的透明传输
  - 支持安全功能
  - 提供服务质量保证

VPN技术

- 隧道技术
  - 隧道是在公共通信网络上构建的一条数据路径，可以提供与专用通信线路等同的连接特性。
  - 隧道使用隧道协议来封装数据。一种协议X的数据报被封装在协议Y中，可以实现协议X在公共网络的透明传输。这里协议X称作被封装协议，协议Y称为封装协议。隧道的一般封装格式为(协议Y(隧道头(协议X)))。
- 隧道协议
  - 第二层隧道：以PPTP，L2TP为代表
  - 第三层隧道：IPSec

隧道的相关知识

- 隧道的定义：实质上是一种封装，将一种协议（协议X）封装在另一种协议（协议Y）中传输，从而实现协议X对公用传输网络(采用协议Y)的透明性
- 隧道协议内包括以下三种协议
  - 乘客协议（Passenger Protocol）
  - 封装协议（Encapsulating Protocol）
  - 运载协议（Carrier Protocol）
- 隧道协议例子



VPN技术

- 密钥管理
  - VPN技术的开放性预示着必须采用各种公开密码算法，这样算法的安全强度不能仅依赖于算法本身，只能依靠密钥的机密性。大规模部署VPN，也离不开自动密钥管理协议的支持。
  - VPN系统中常用的几种密钥管理协议包括：IKE协议、SKIP协议、Kerberos协议。

VPN分类

- 按VPN业务类型划分：
  - Intranet VPN（内部公文流转）
  - Access VPN（远程拨号VPN）
  - Extranet VPN（各分支机构互联）
- 按VPN发起主体划分：
  - 客户发起，也称基于客户的VPN
  - 服务器发起，也称客户透明方式或基于网络的VPN

VPN分类

- 按隧道协议层次划分：
  - 二层隧道协议：L2F/L2TP 、PPTP
  - 三层隧道协议：GRE （通用路由封装协议） 、IPSec
  - 介于二、三层间的隧道协议：MPLS
  - 基于SOCKS V5的VPN
- 此外，根据VPN实现方式不同，还可进一步分为软件实现和硬件实现等。

VPN应用类型

根据网络类型的差异，一般可以把VPN分为Client-LAN和LAN-LAN两种类型。

（1）、Client-LAN类型的VPN也称为Acess VPN，即远程访问方式的VPN。它提供了一种安全的远程访问手段，例如，出差在外的员工、有远程办公需要的分支机构，都可以利用这种类型的VPN，实现安全的对企业内部网络资源进行远程访问。它又分为基于internet远程访问的VPN，和基于intranet远程访问的VPN。

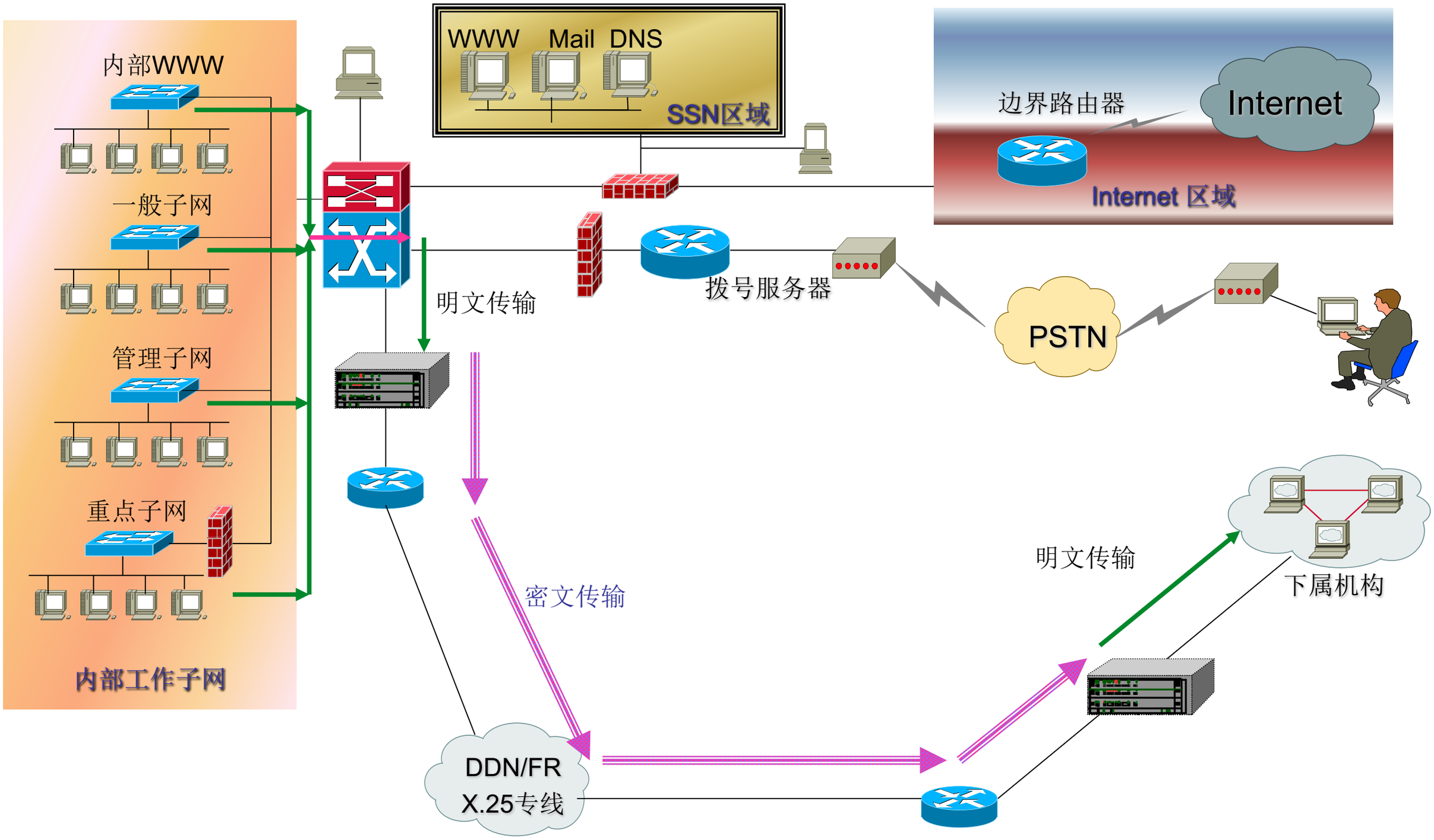
VPN应用类型

(2)、采用LAN-LAN类型的VPN，可以利用基本的internet和intranet网络建立起全球范围内物理的连接，再利用VPN的隧道协议实现安全保密需要，就可以满足公司总部与分支机构以及合作企业间的安全网络连接。这种类型的VPN通常采用IPSec协议建立加密传输数据隧道。LAN-LAN类型的VPN，当用来构建内联网时称为Intranet VPN，用于企业和合作企业进行网络互联时称为Extranet VPN。

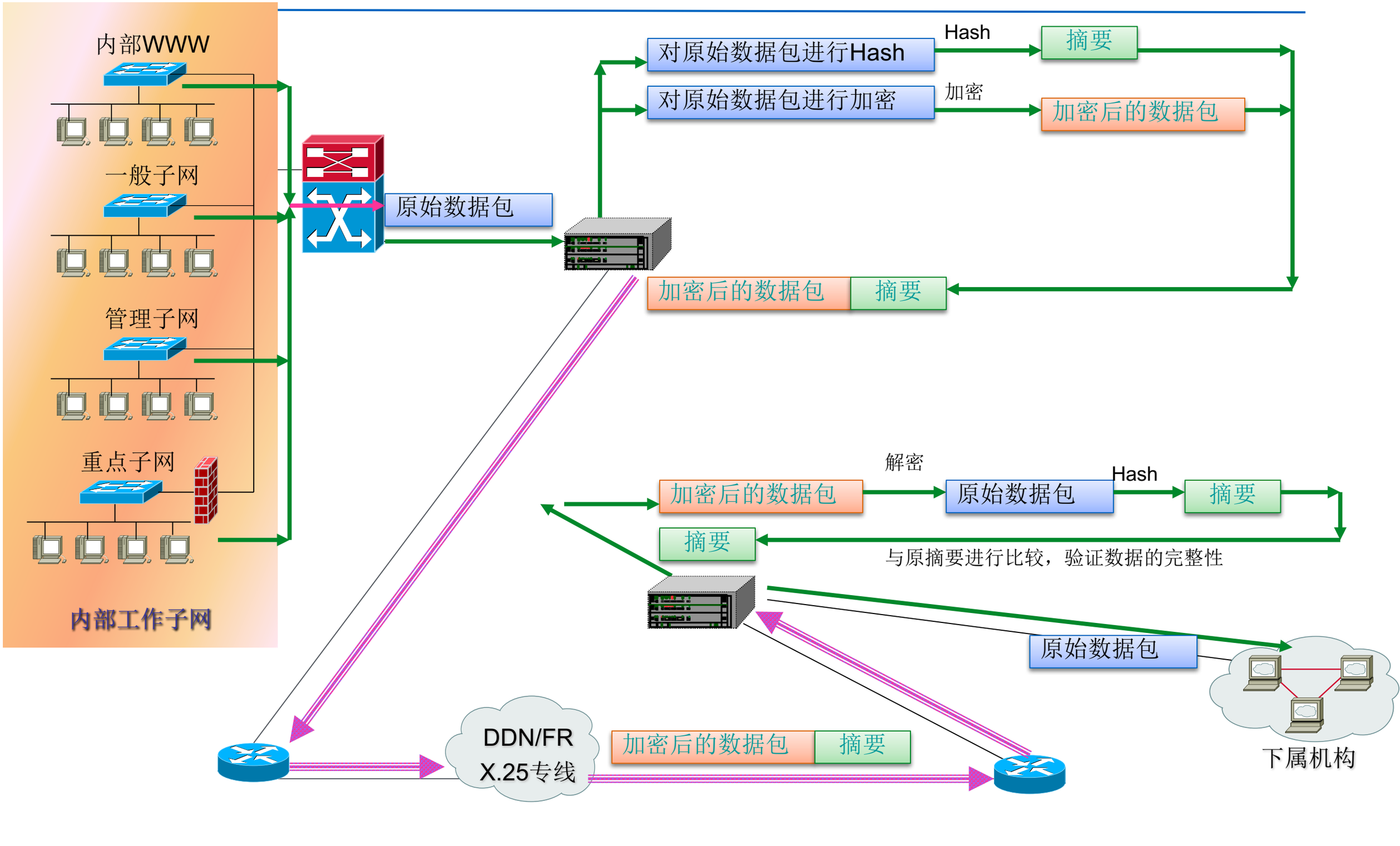
VPN功能

- 数据机密性保护
- 数据完整性保护
- 数据源身份认证
- 重放攻击保护

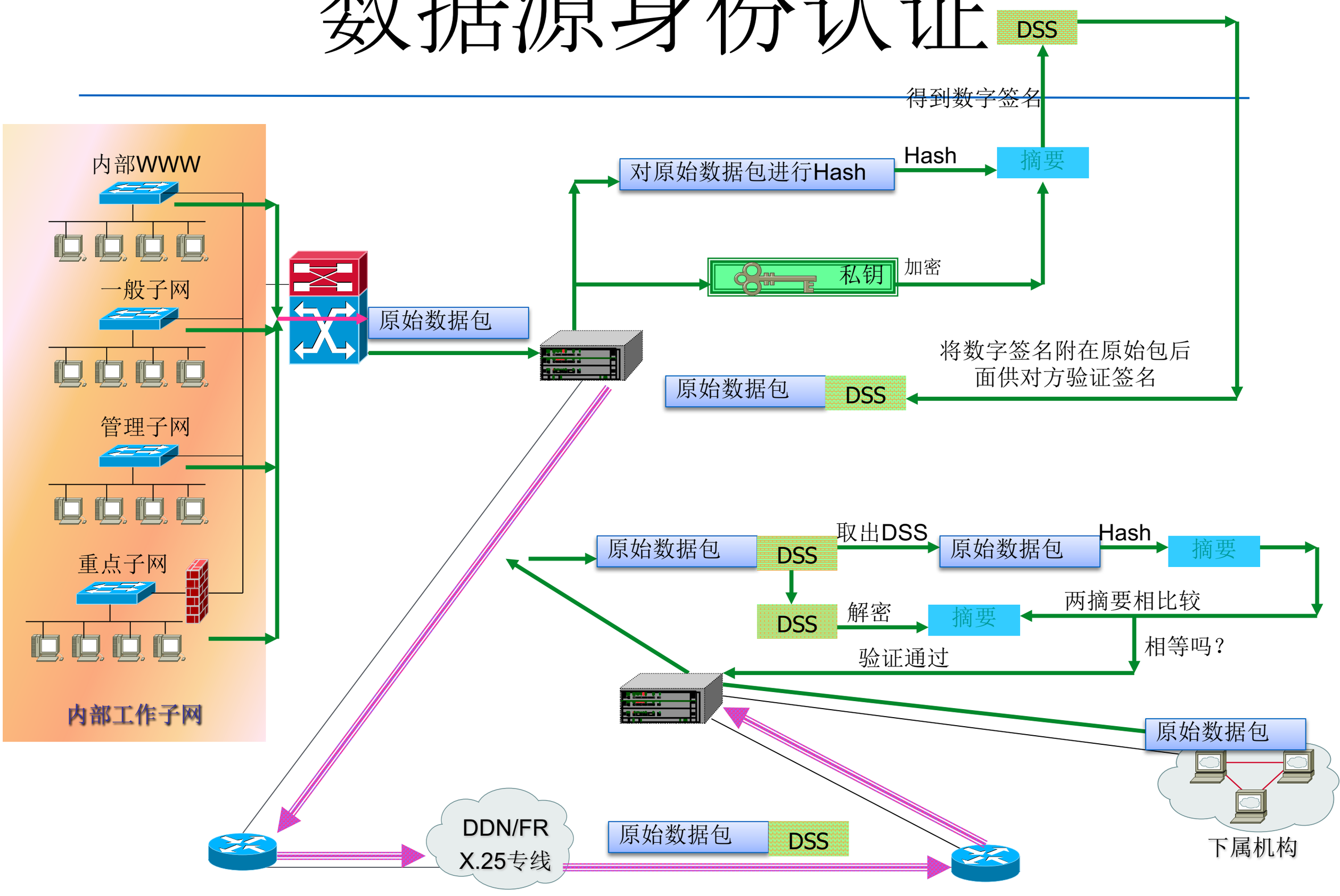
数据机密性保护


内部WWW
一般子网
管理子网
重点子网
内部工作子网
WWW
Mail
DNS
SSN区域
边界路由器
Internet
Internet 区域
明文传输
拨号服务器
PSTN
密文传输
明文传输
下属机构
DDN/FR
X.25专线

数据完整性保护

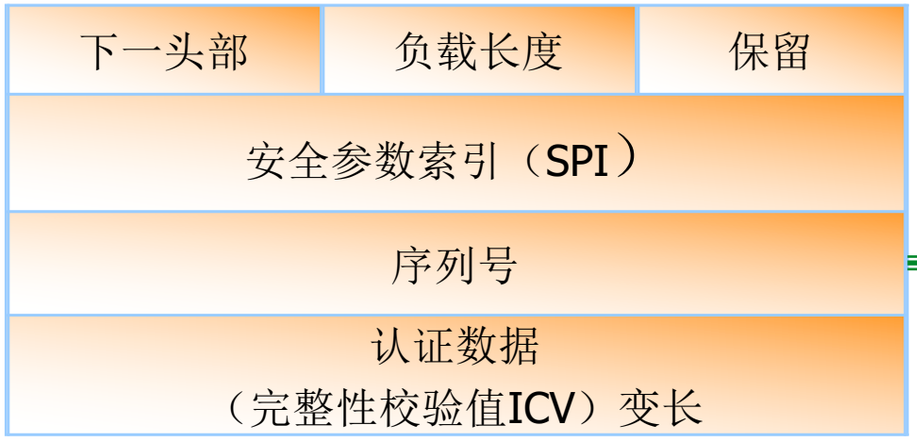

内部WWW
一般子网
管理子网
重点子网
内部工作子网
原始数据包
对原始数据包进行Hash
Hash
摘要
对原始数据包进行加密
加密
加密后的数据包
加密后的数据包
摘要
DDN/FR
X.25专线
加密后的数据包
摘要
解密
加密后的数据包
原始数据包
Hash
摘要
摘要
与原摘要进行比较，验证数据的完整性
原始数据包
下属机构

数据源身份认证

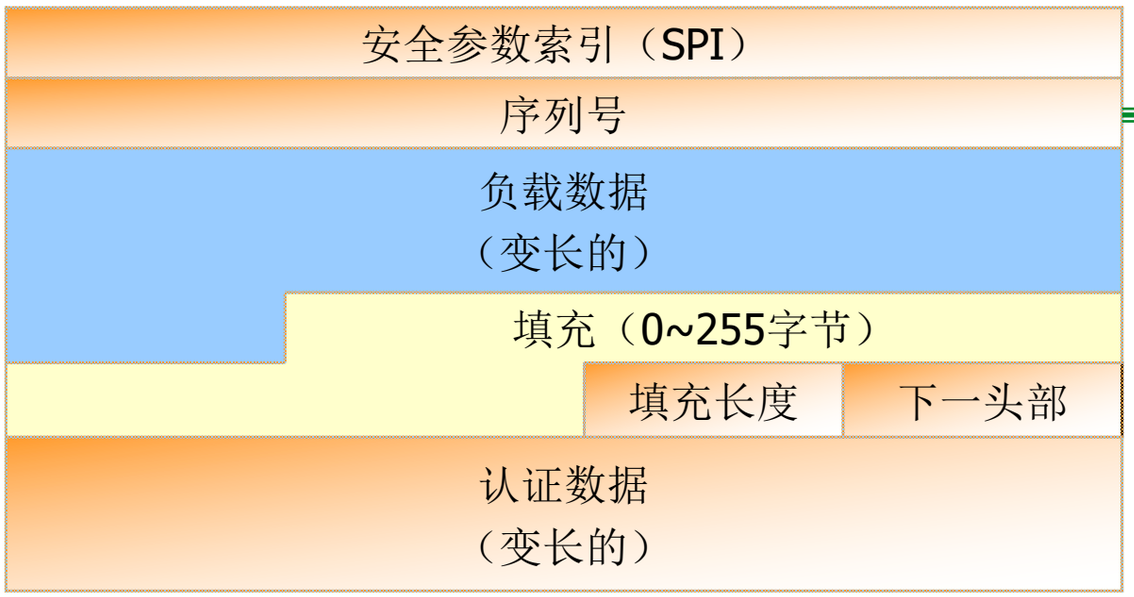

内部WWW
一般子网
管理子网
重点子网
内部工作子网
原始数据包
对原始数据包进行Hash
Hash
摘要
私钥
加密
DSS
得到数字签名
将数字签名附在原始包后
面供对方验证签名
原始数据包
DSS
取出DSS
原始数据包
Hash
摘要
两摘要相比较
DSS
解密
摘要
相等吗？
验证通过
原始数据包
DDN/FR
X.25专线
原始数据包
DSS
下属机构

重放攻击保护

AH协议头



ESP协议头



SA建立之初，序列号初始化为0，使用该SA传递的第一个数据包序列号为1，序列号不允许重复，因此每个SA所能传递的最大IP报文数为2^{32}—1，当序列号达到最大时，就需要建立一个新的SA，使用新的密钥。

虚拟专用网——

链路层安全

链路层安全

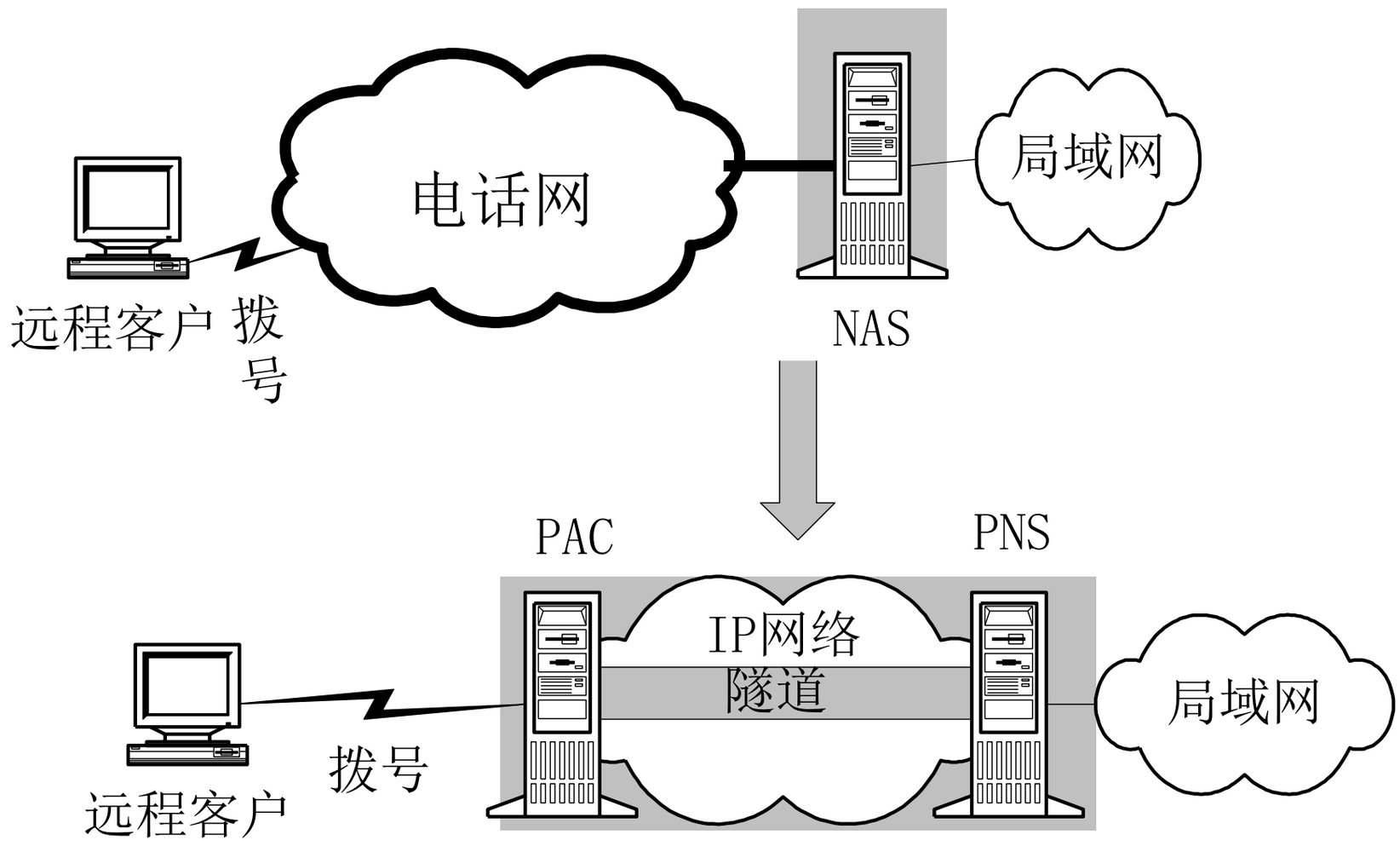
- PPTP
- L2TP
- L2F

PPTP

- PPTP（Point to Point TunnelingProtocol，点对点通道协议）
  - PPTP 提供PPTP 客户机和PPTP 服务器之间的加密通信。
    - PPTP 客户机是指运行了该协议的PC机，如启动该协议的Windows95/98；
    - PPTP 服务器是指运行该协议的服务器，如启动该协议的WindowsNT 服务器。
  - PPTP 可看作是PPP 协议的一种扩展。
    - 提供了一种在Internet 上建立多协议的安全虚拟专用网（VPN）的通信方式。远端用户能够透过任何支持PPTP的ISP 访问公司的专用网络。

PPTP

PPTP由微软公司设计，用于将PPP分组通过IP网络封装传输



PPTP

PPTP的数据封装：

数据链路层报头	IP报头	GRE报头	PPP报头	加密PPP有效载荷	数据链路层报尾

PPTP客户机或PPTP服务器在接收到PPTP数据包后，将做如下处理：

- 处理并去除数据链路层报头和报尾；
- 处理并去除IP报头；
- 处理并去除GRE和PPP报头；
- 如果需要的话，对PPP有效载荷即传输数据进行解密或解压缩；
- 对传输数据进行接收或转发处理。

PPTP

- 通过PPTP，客户可采用拨号方式接入公共IP网络Internet
  - 客户按常规方式拨号到ISP接入服务器（NAS），建立PPP连接；
  - 然后，客户进行二次拨号建立到PPTP服务器的连接，该连接称为PPTP 隧道， 实质上是基于IP 协议上的另一个PPP 连接，其中的IP包可以封装多种协议数据，包括TCP/IP、IPX和NetBEUI。
  - 对于直接连到Internet 上的客户则不需要第一重PPP的拨号连接，可以直接与PPTP 服务器建立虚拟通道。
  - PPTP 只支持IP 作为传输协议。
  - PPTP 采用了基于RSA 公司RC4 的数据加密方法， 保证了虚拟连接通道的安全性。

L2F

- L2F（Layer 2 Forwarding，二层转发协议）
  - L2F 是由Cisco 公司提出的可以在多种介质如ATM、帧中继、IP 网上建立多协议的安全虚拟专用网（VPN） 的通信方式。
  - 远端用户能够透过任何拨号方式接入公共IP 网络
    - 首先按常规方式拨号到ISP 的接入服务器（NAS），建立PPP 连接；
    - NAS 根据用户名等信息， 发起第二重连接， 通向HGW （家庭网关）服务器。
    - 在这种情况下隧道的配置和建立对用户是完全透明的。

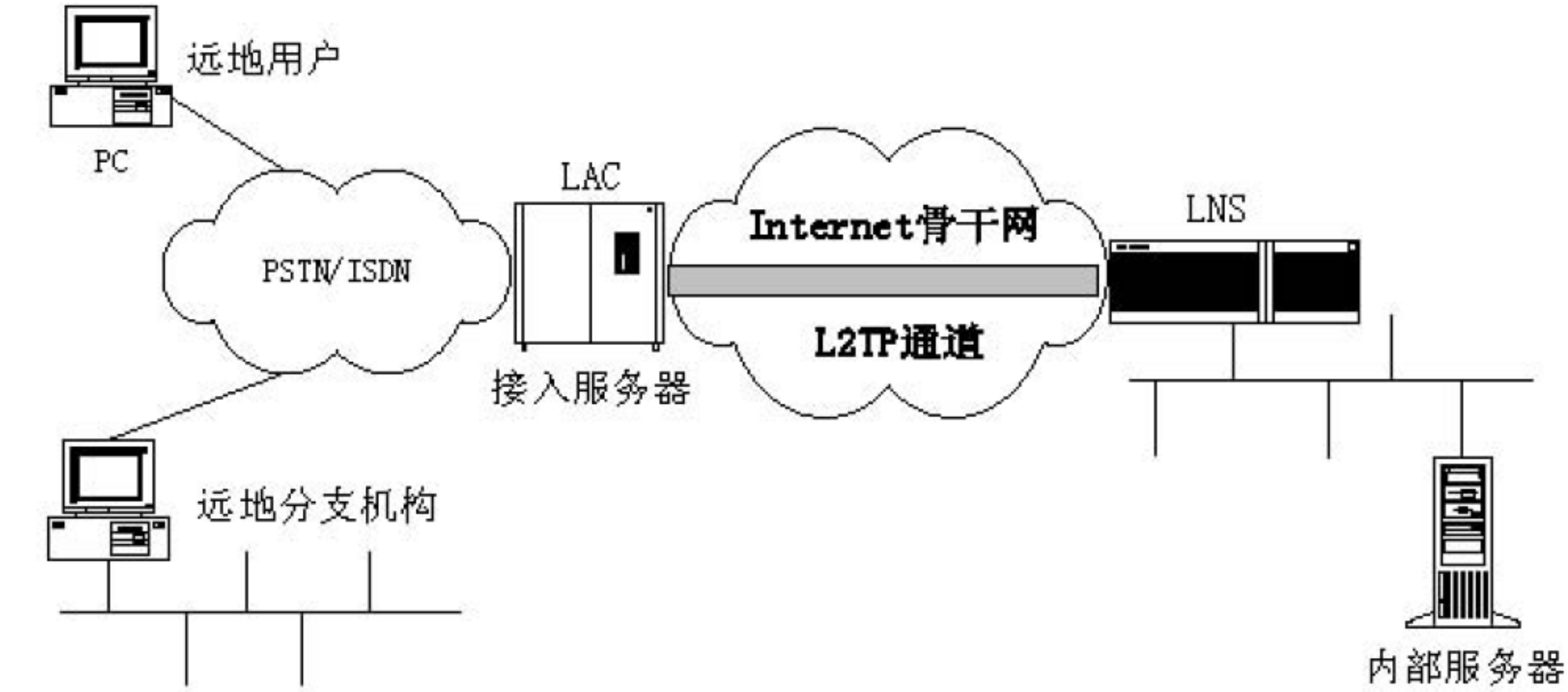
L2TP

- L2TP（Layer 2 Tunneling Protocol，二层通道协议）
  - L2TP 结合了L2F 和PPTP 的优点， 可以让用户从客户端或访问服务器端发起VPN 连接。L2TP 是把链路层PPP 帧封装在公共网络设施如IP、ATM、帧中继中进行隧道传输的封装协议。
  - L2TP主要由LAC(L2TP Access Concentrator)和LNS(L2TPNetworkServer)构成，LAC(L2TP访问集中器)支持客户端的L2TP，他用于发起呼叫，接收呼叫和建立隧道；LNS(L2TP网络服务器)是所有隧道的终点。在传统的PPP连接中，用户拨号连接的终点是LAC，L2TP使得PPP协议的终点延伸到LNS。
  - Cisco、Ascend、Microsoft 和RedBack 公司的专家们在修改了十几个版本后， 终于在1999 年8 月公布了L2TP 的标准RFC2661。

L2TP

- L2TP支持多种协议
  - L2TP 支持多个PPP链路的捆绑问题
    - PPP链路捆绑要求其成员均指向同一个NAS，L2TP可以使物理上连接到不同NAS的PPP链路，在逻辑上的终结点为同一个物理设备。
  - L2TP扩展了PPP连接
    - 在传统方式中用户通过模拟电话线或ISDN/ADSL与网络访问服务器(NAS)建立一个第2层的连接，并在其上运行PPP，第2层连接的终结点和PPP会话的终结点在同一个设备上(如NAS)。L2TP作为PPP 扩展提供更强大的功能，包括第2层连接的终结点和PPP会话的终结点可以是不同的设备。

L2TP协议结构



- LAC表示L2TP访问集中器（L2TP Access Concentrator），是附属在交换网络上的具有PPP端系统和L2TP协议处理能力的设备，LAC一般是一个网络接入服务器NAS，主要用于通过PSTN/ISDN网络为用户提供接入服务。
- LNS表示L2TP网络服务器（L2TP Network Server），是PPP端系统上用于处理L2TP协议服务器端部分的设备。

L2TP

- 数据封装格式：

IP头	UDP头	L2TP头	PPP数据

- 特点：
  - 它综合了第二层转发协议（L2F）和PPTP两种协议各自的优点
  - 协议的额外开销较少

L2TP与PPTP不同点

- PPTP和L2TP都使用PPP协议对数据进行封装，然后添加附加包头用于数据在互联网络上的传输。尽管两个协议非常相似，但是仍存在以下几方面的不同：
  - 1.PPTP要求互联网络为IP网络。L2TP只要求隧道媒介提供面向数据包的点对点的连接。L2TP可以在IP（使用UDP），桢中继永久虚拟电路（PVCs),X.25虚拟电路（VCs）或ATM VCs网络上使用。
  - 2.PPTP只能在两端点间建立单一隧道。L2TP支持在两端点间使用多隧道。使用L2TP，用户可以针对不同的服务质量创建不同的隧道。
  - 3.L2TP可以提供包头压缩。当压缩包头时，系统开销（overhead）占用4个字节，而PPTP协议下要占用6个字节。
  - 4.L2TP可以提供隧道验证，而PPTP则不支持隧道验证。但是当L2TP或PPTP与IPSEC共同使用时，可以由IPSEC提供隧道验证，不需要在第2层协议上验证隧道

虚拟专用网——

网络层安全

网络层安全

- IPSec
  - IPSec体系
  - AH和ESP
  - IPSec模式
  - SADB
  - IPSec流程
- IKE
  - IKE基本情况
  - IKE交换格式
  - 第一阶段交换
  - 第二阶段交换

基于IPSec的VPN解决方案

在通信协议分层中，网络层是可能实现端到端安全通信的最低层，它为所有应用层数据提供透明的安全保护，用户无需修改应用层协议。

- 该方案能解决的问题：
  - ➔ 数据源身份认证：证实数据报文是所声称的发送者发出的。
  - ➔ 数据完整性：证实数据报文的内容在传输过程中没被修改过，无论是被故意改动或是由于发生了随机的传输错误。
  - ➔ 数据保密：隐藏明文的消息，通常靠加密来实现。
  - ➔ 重放攻击保护：保证攻击者不能截获数据报文，且稍后某个时间再发放数据报文，而不会被检测到。
  - ➔ 自动的密钥管理和安全关联管理：保证只需少量或根本不需要手工配置，就可以在扩展的网络上方便精确地实现公司的虚拟使用网络方针

IPSec

- IPSec
  - 即IP层安全协议，是由Internet组织IETF的IPSec工作组制定的IP网络层安全标准。
  - 它通过对IP报文的封装以实现TCP/IP网络上数据的安全传送。

IPSec

- IPSec体系
- AH和ESP
- IPSec模式
- SADB
- IPSec流程

IPSec体系结构

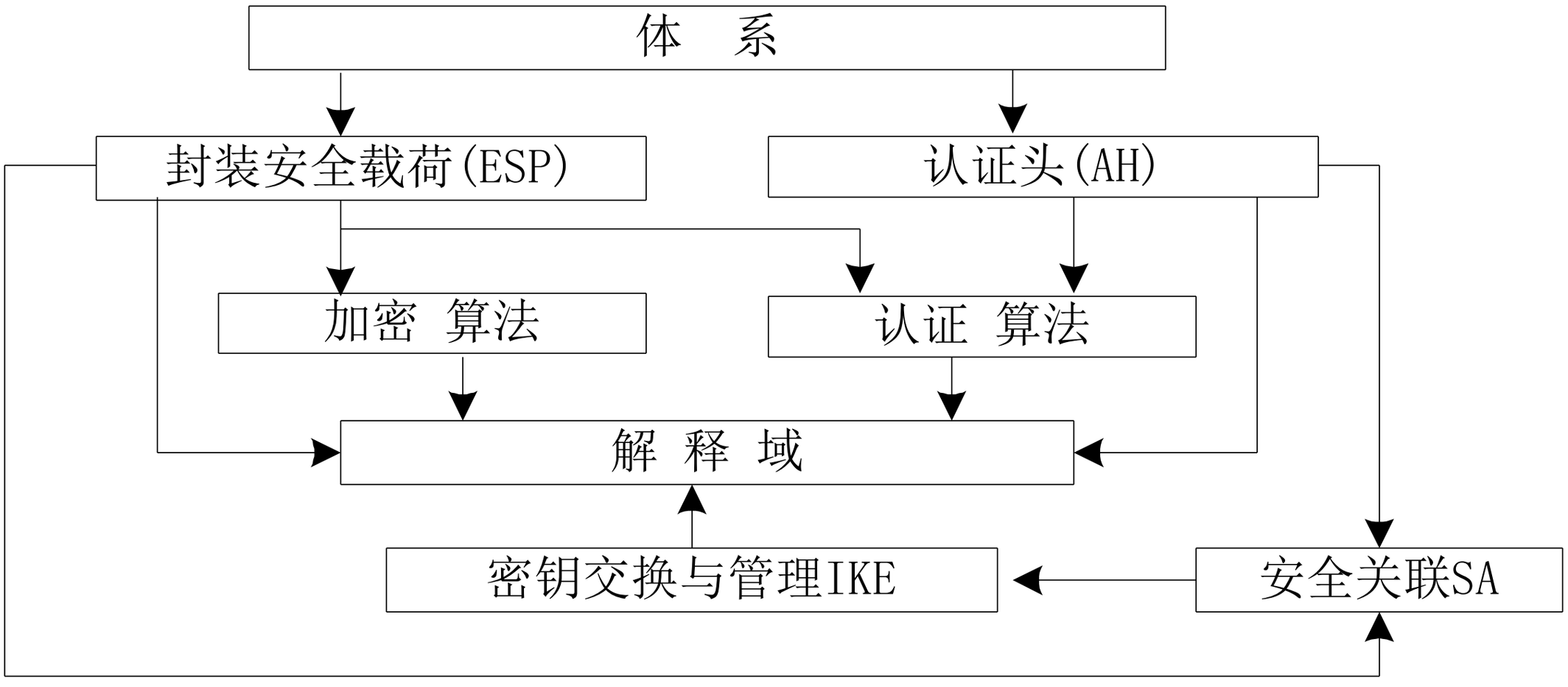


图 IPSec安全体系结构

IPSec协议框架（1）

- 综合了密码技术和协议安全机制，IPSec协议的设计目标是在IPV4和IPV6环境中为网络层流量提供灵活的安全服务。

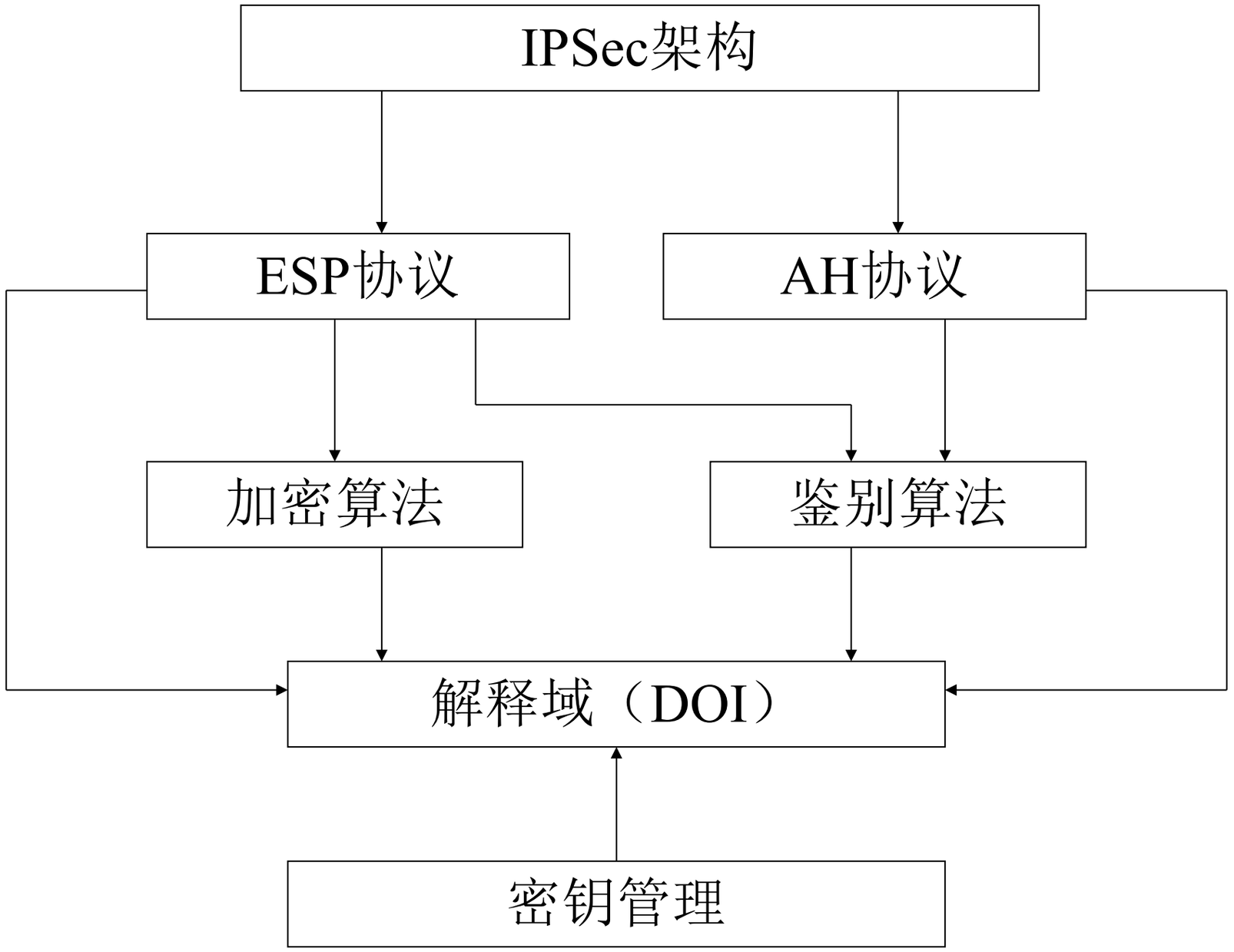
- IPSec协议提供的安全服务包括：访问控制、无连接完整性、数据源鉴别、重传攻击保护、机密性、有限的流量保密等。

- IPSec协议主要内容包括：
  - 协议框架－RFC2401
  - 安全协议：AH协议－RFC2402、ESP协议－RFC2406

IPSec协议框架（2）

- 密钥管理协议：IKE － RFC2409 、ISAKMP－RFC2408、 OAKLEY协议－RFC2412。
- 密码算法：HMAC－RFC2104/2404、CAST－RFC2144、ESP加密算法－RFC2405/2451等。
- 其他：解释域DOI－RFC2407、IPComp－RFC2393、Roadmap－RFC2411。

IPSec协议框架（3）



IPSec协议文件框架图

AH协议

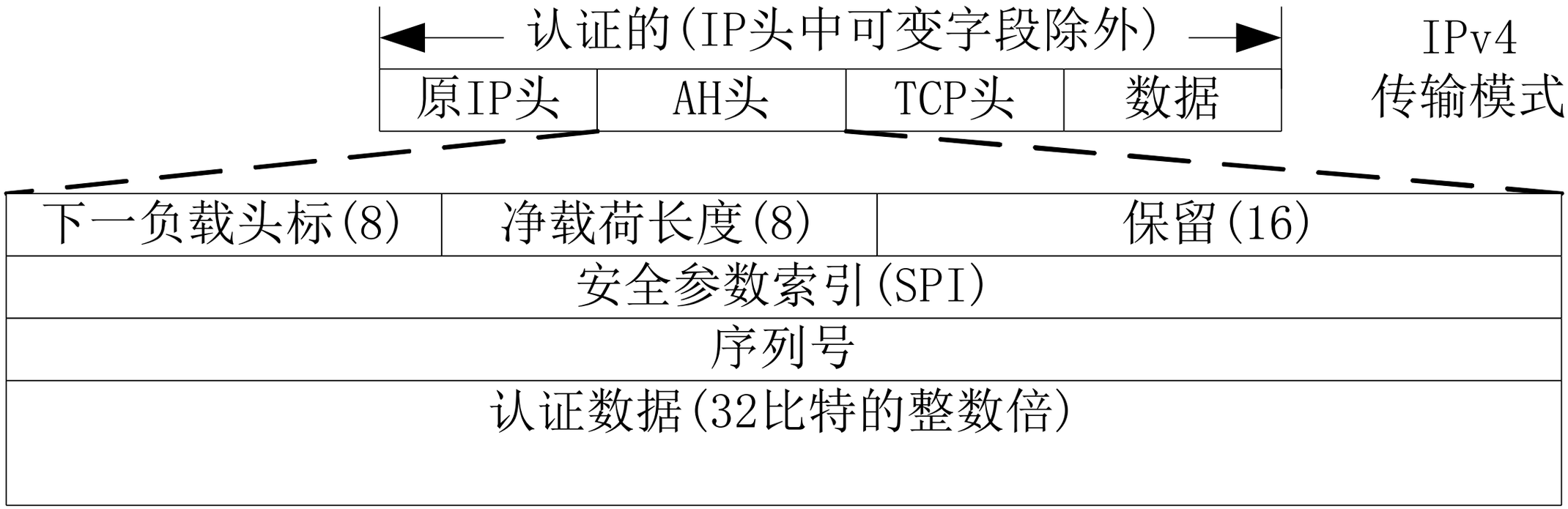
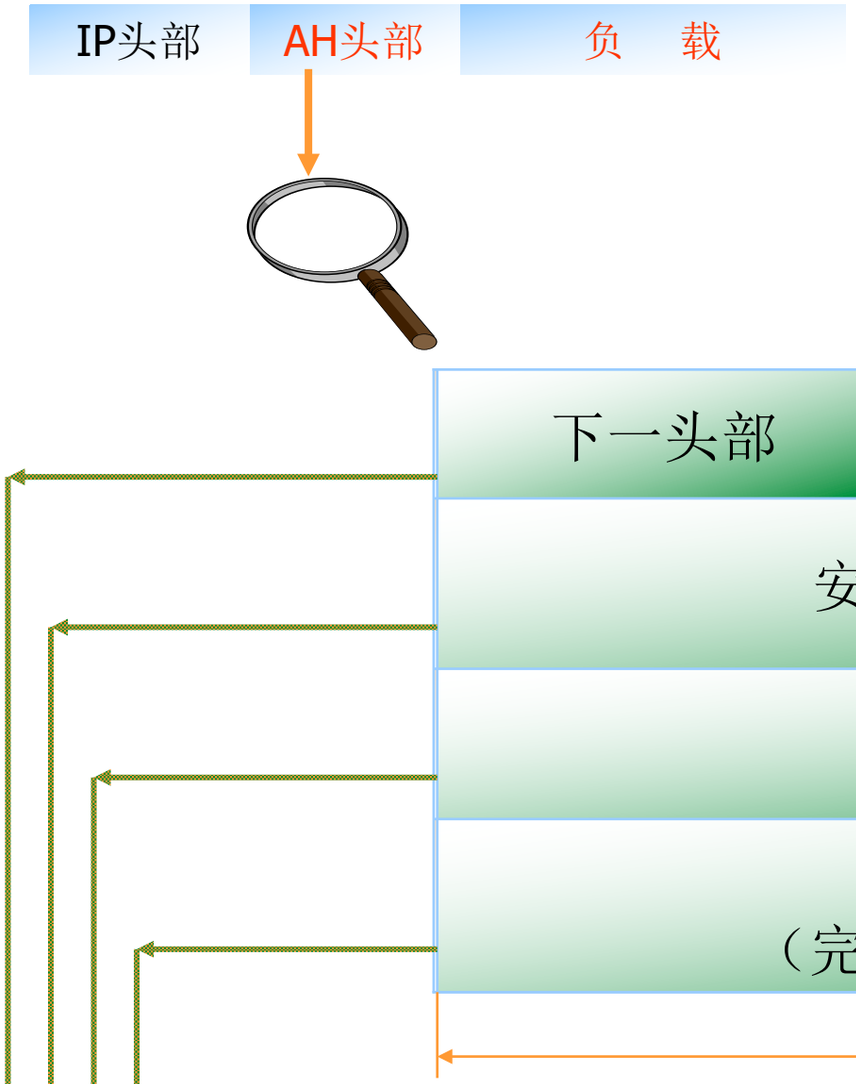

认证的(IP头中可变字段除外)
原IP头
AH头
TCP头
数据
IPv4
传输模式
下一负载头标(8)
净载荷长度(8)
保留(16)
安全参数索引(SPI)
序列号
认证数据(32比特的整数倍)

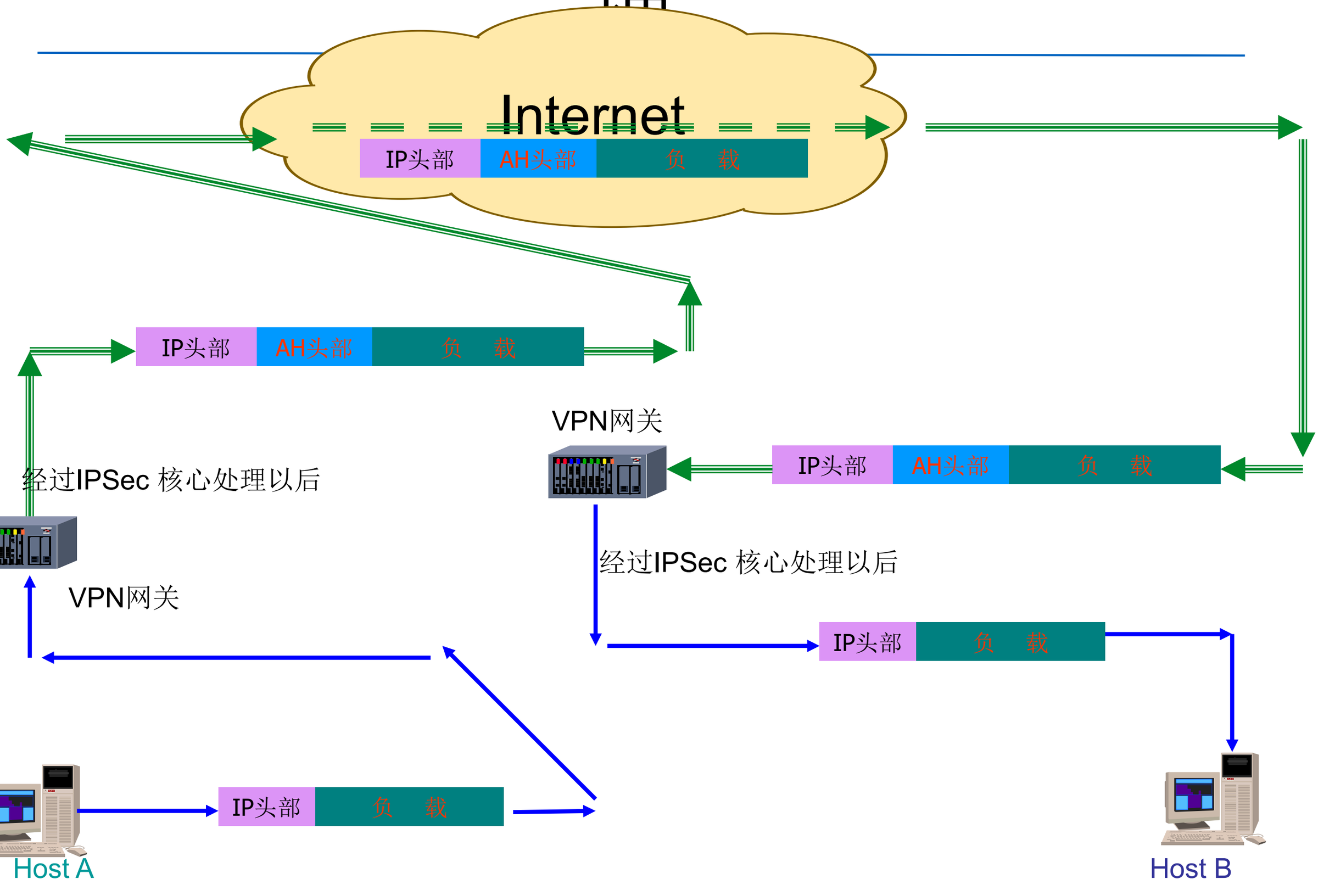
图 AH的格式

认证头部(AH)

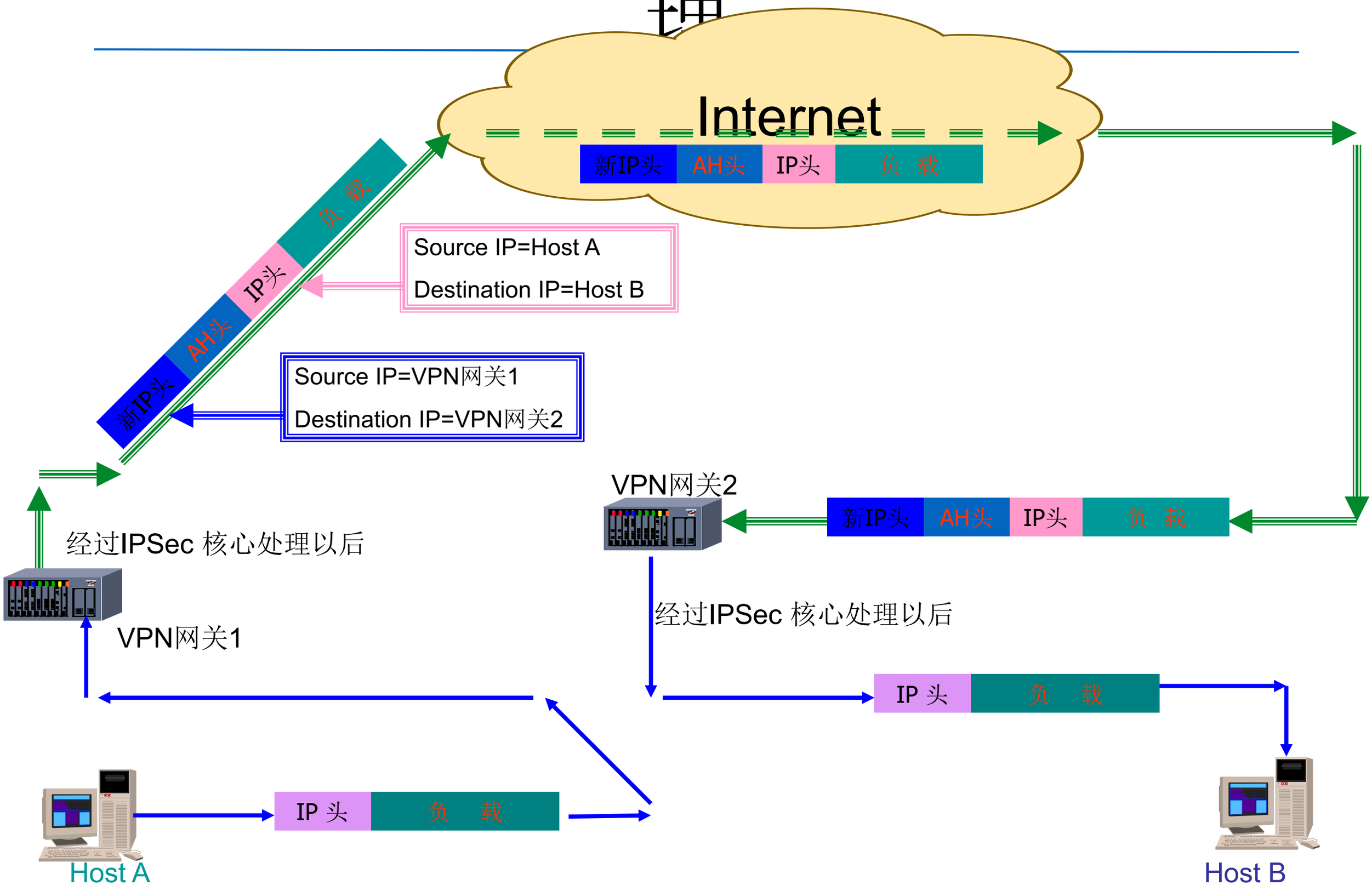


- ❖认证数据：一个变长字段，也叫Integrity Check Value，由SA初始化时指定的算法来计算。长度=整数倍32位比特
- ❖序列号：32比特，一个单项递增的计数器，用于防止重放攻击，SA建立之初初始化为0，序列号不允许重复
- ❖SPI：32比特，用于标识有相同IP地址和相同安全协议的不同SA。由SA的创建者定义，只有逻辑意义
- ❖ 下一头部：8比特，标识认证头后面的下一个负载类型
- ❖保留字段：16比特，保留将来使用，Default=0
- ❖ 负载长度：8比特，表示以32比特为单位的AH头部长度减2，Default=4

传输模式下的AH认证工作原理



隧道模式下的AH认证工作原理



ESP协议

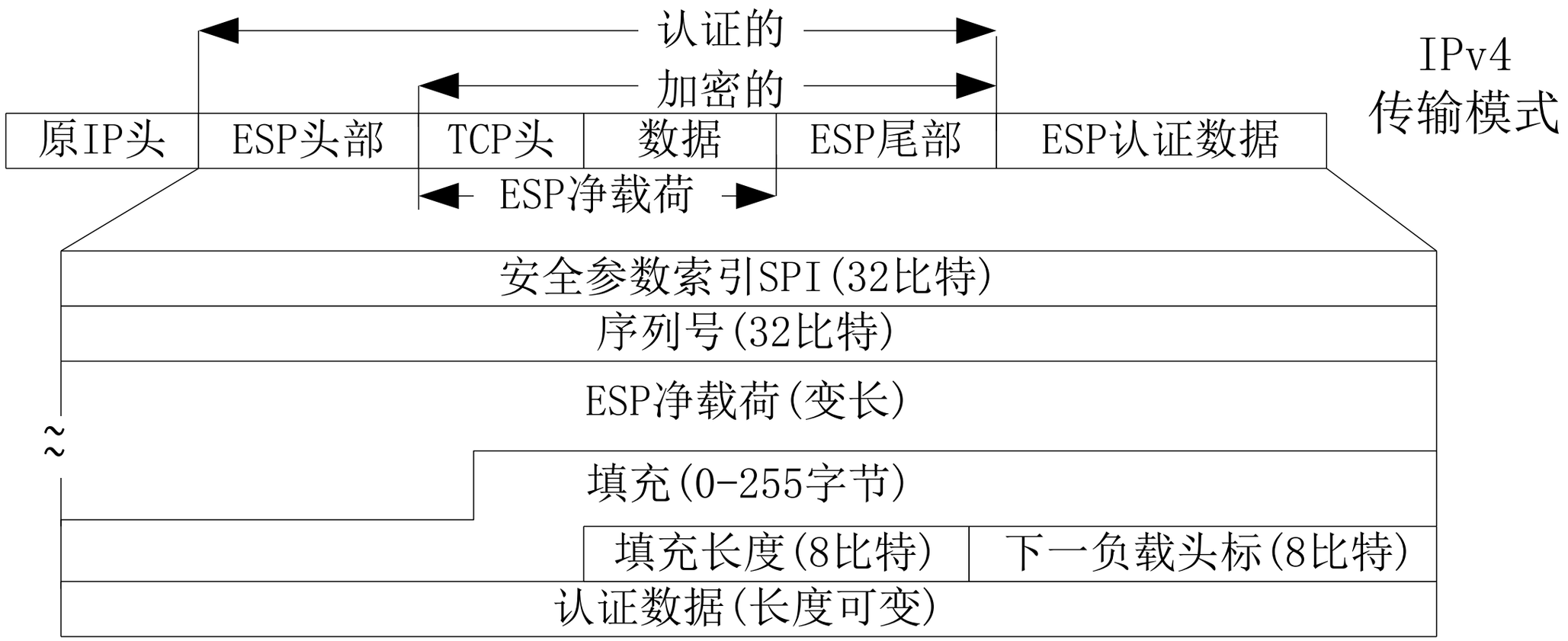
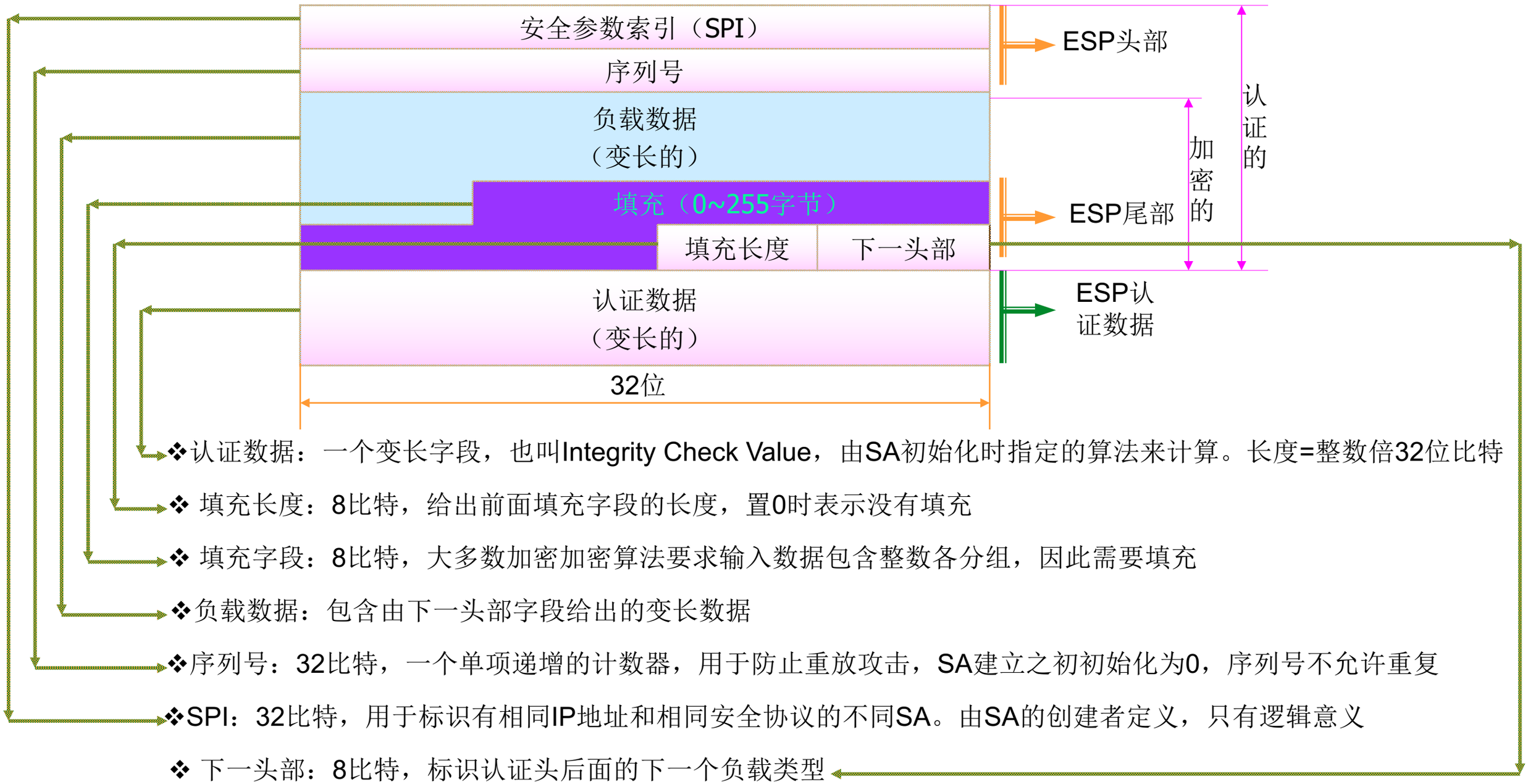


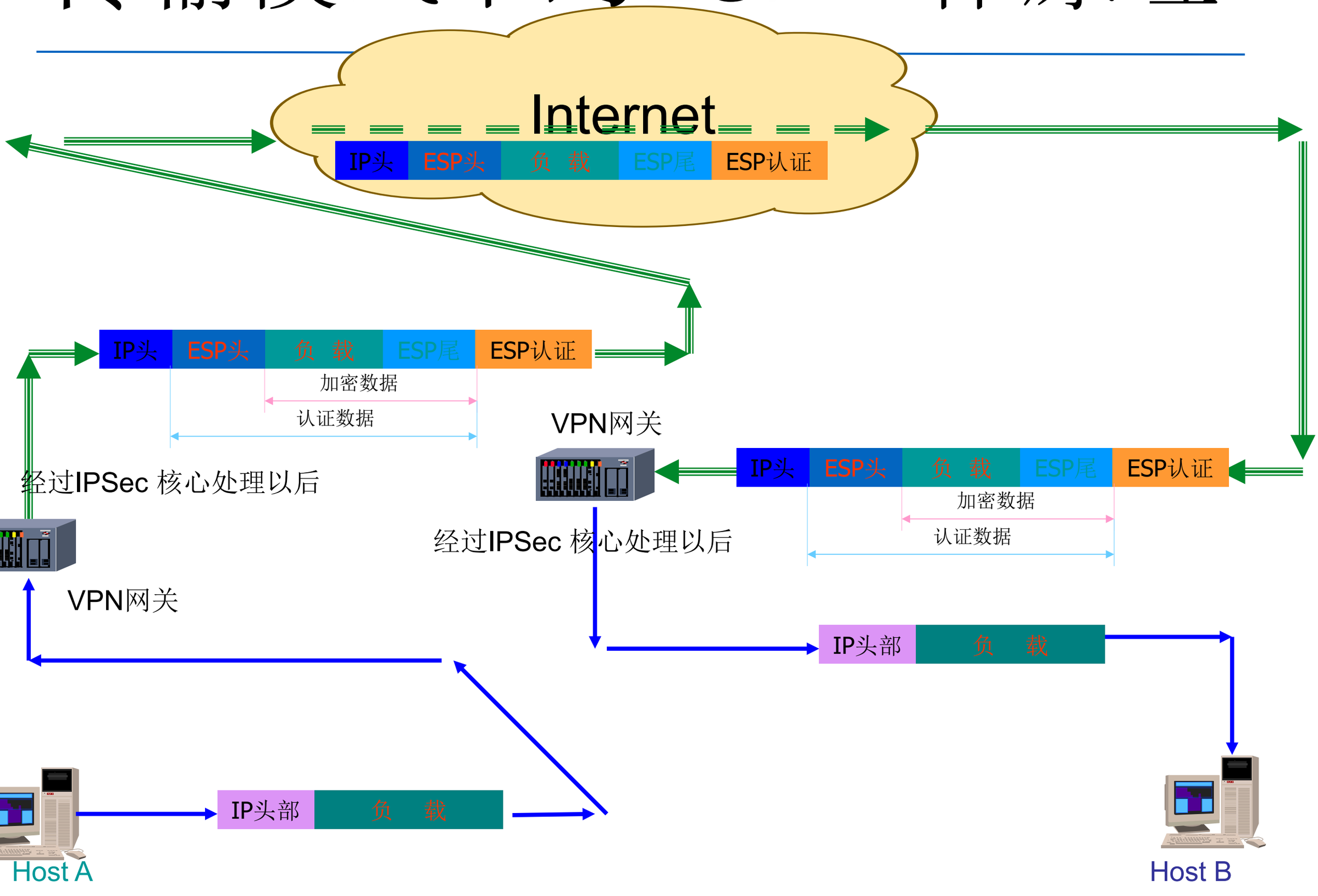
图 ESP格式

负载安全封装(ESP)

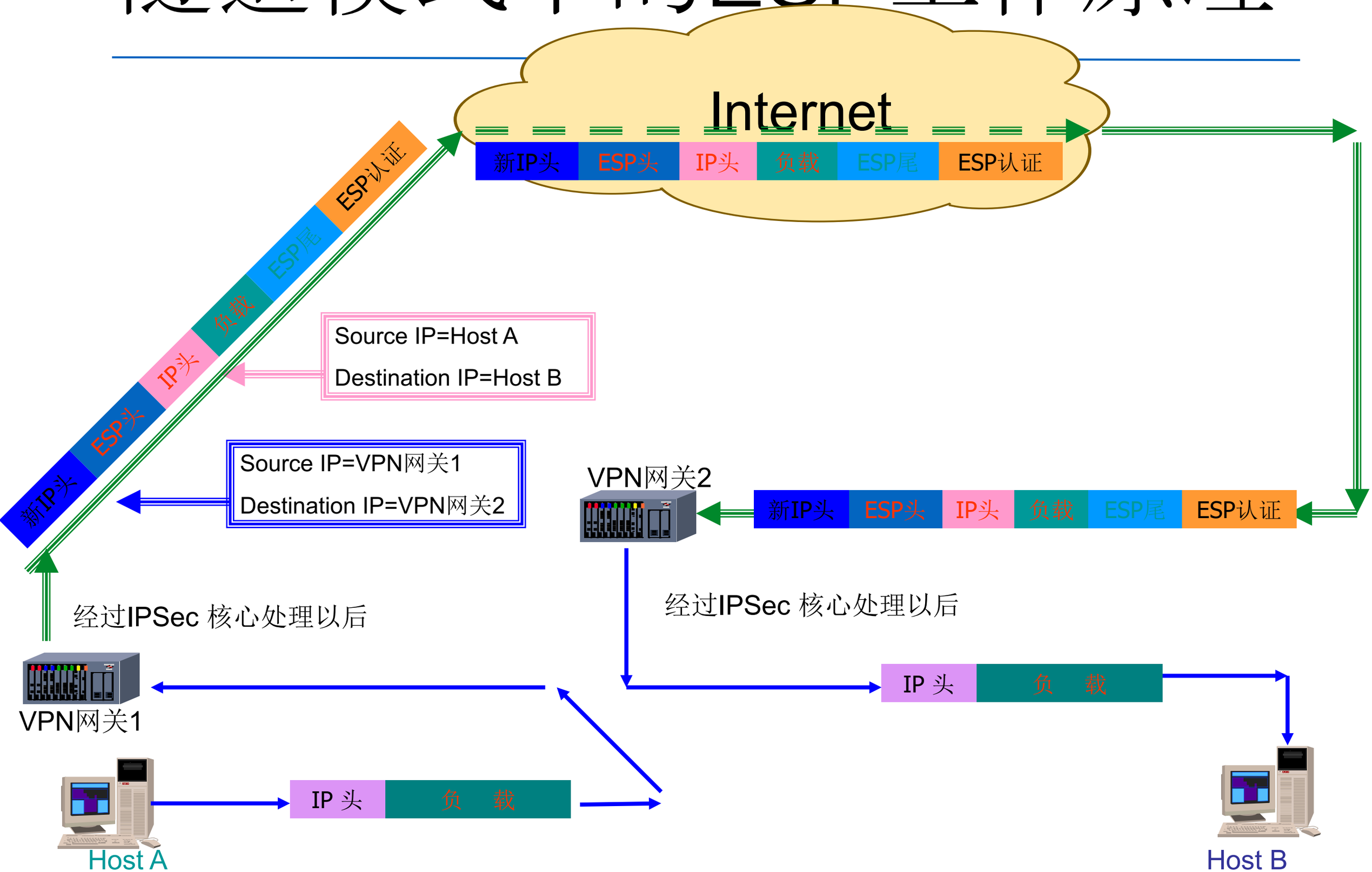


- ❖认证数据：一个变长字段，也叫Integrity Check Value，由SA初始化时指定的算法来计算。长度=整数倍32位比特
- ❖ 填充长度：8比特，给出前面填充字段的长度，置0时表示没有填充
- ❖ 填充字段：8比特，大多数加密加密算法要求输入数据包含整数各分组，因此需要填充
- ❖负载数据：包含由下一头部字段给出的变长数据
- ❖序列号：32比特，一个单项递增的计数器，用于防止重放攻击，SA建立之初初始化为0，序列号不允许重复
- ❖SPI：32比特，用于标识有相同IP地址和相同安全协议的不同SA。由SA的创建者定义，只有逻辑意义
- ❖ 下一头部：8比特，标识认证头后面的下一个负载类型

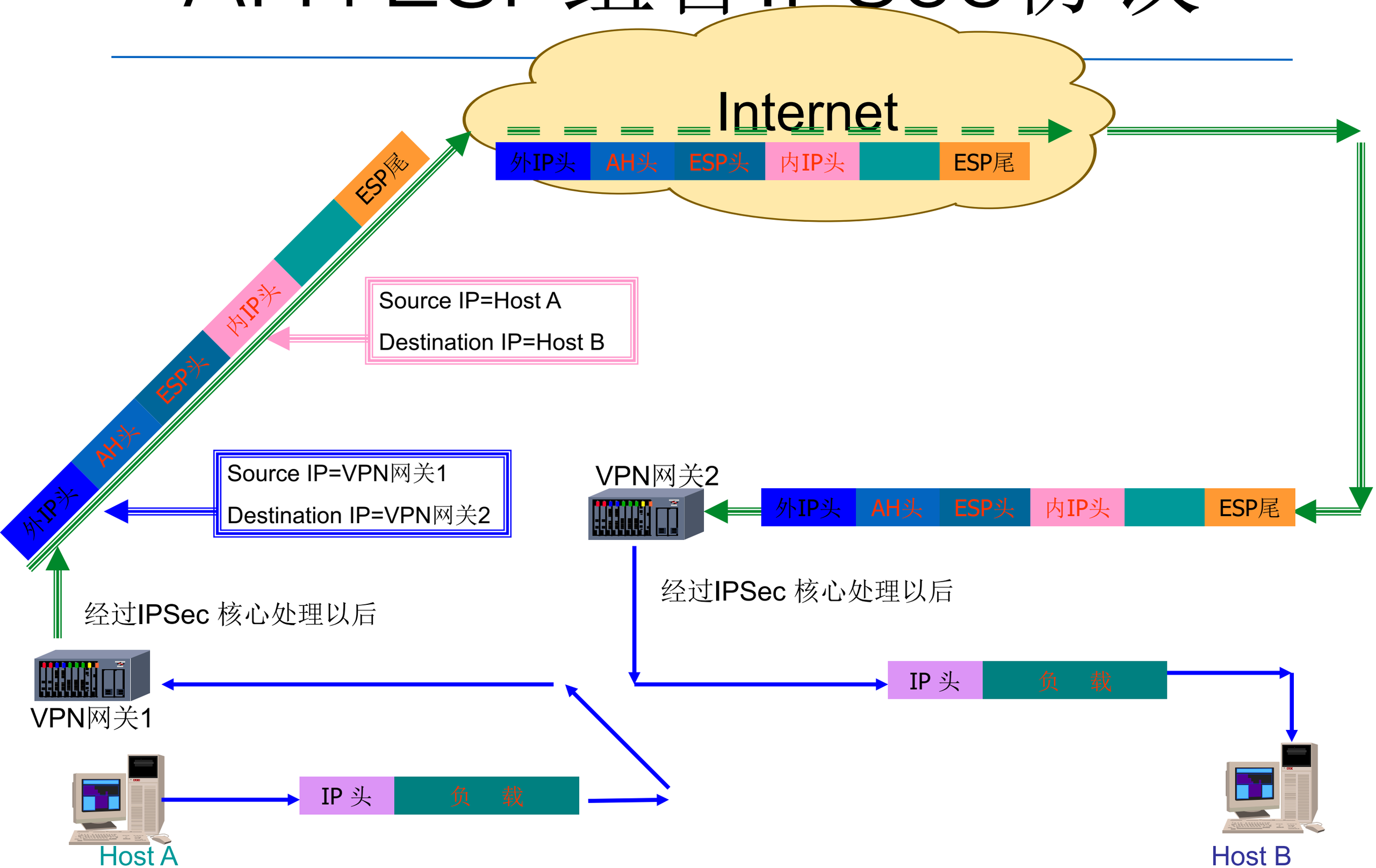
传输模式下的ESP工作原理



隧道模式下的ESP工作原理



AH+ESP组合IPSec协议



AH\ESP协议分工

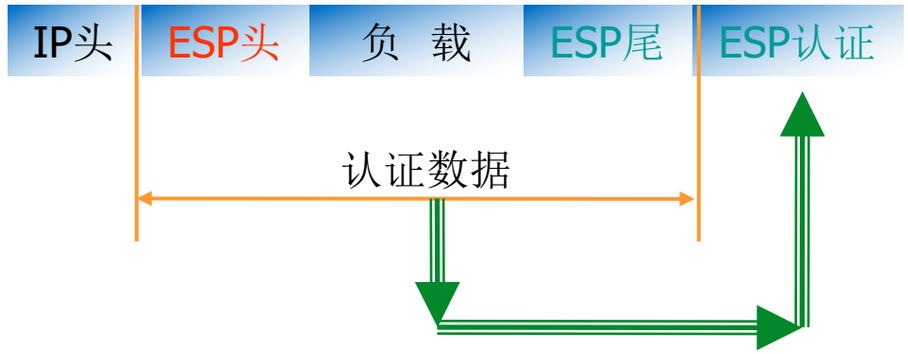
AH协议

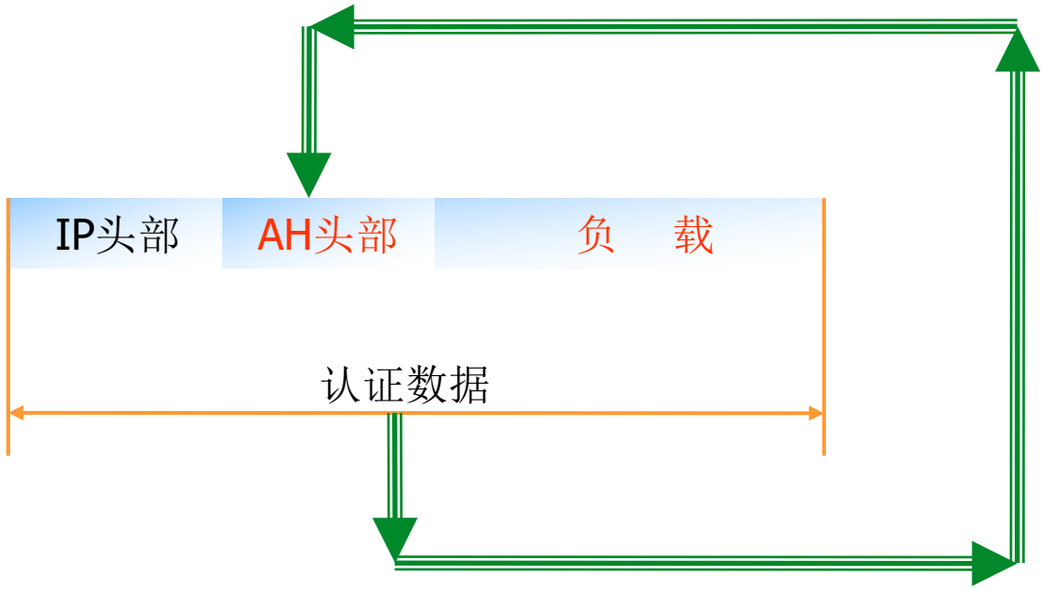
- 身份认证
- 数据完整性校验
- 重放攻击保护

ESP可以取代AH吗？

ESP协议

- 身份认证
- 数据加密
- 数据完整性校验
- 重放攻击保护





IPSec传输模式

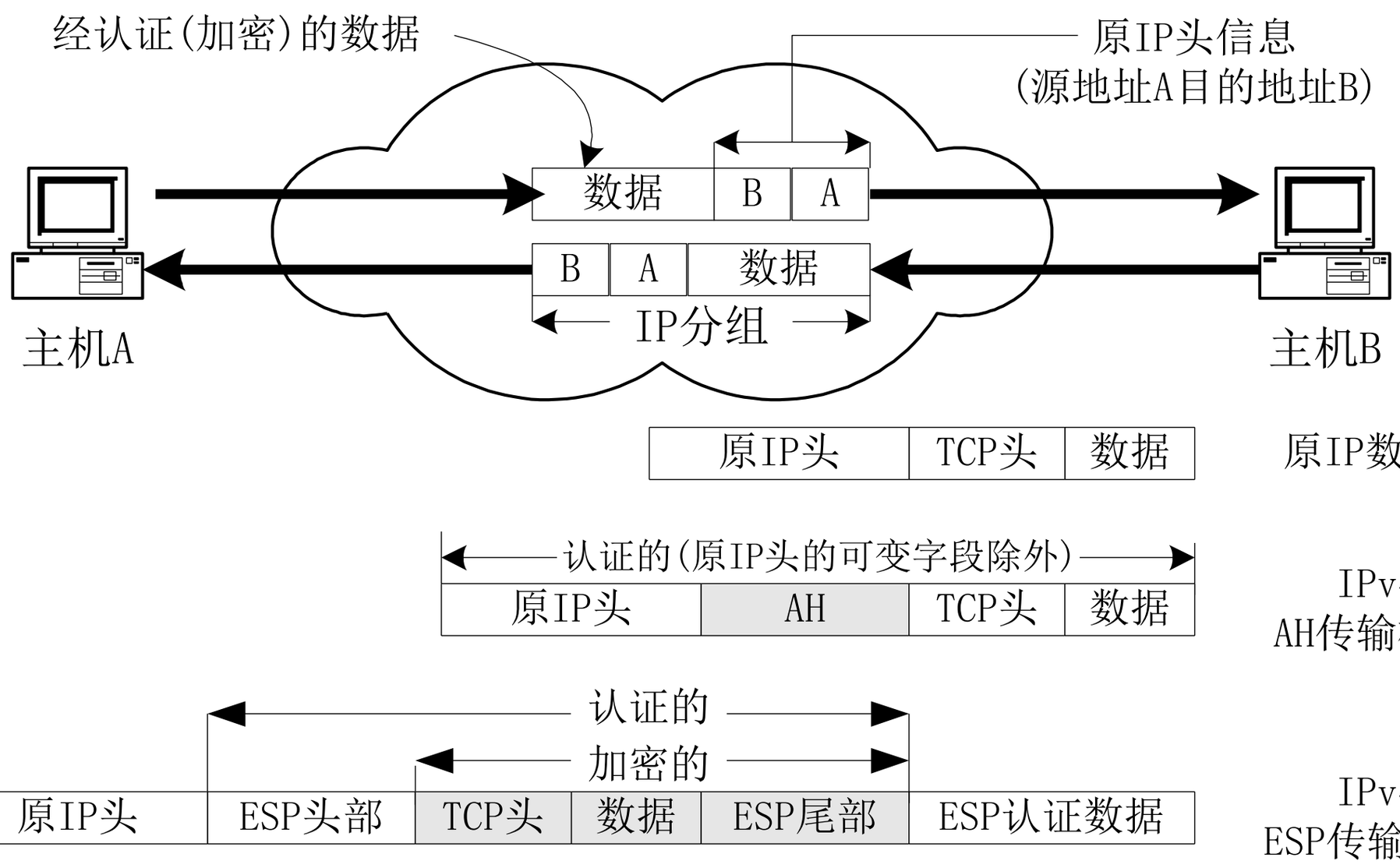
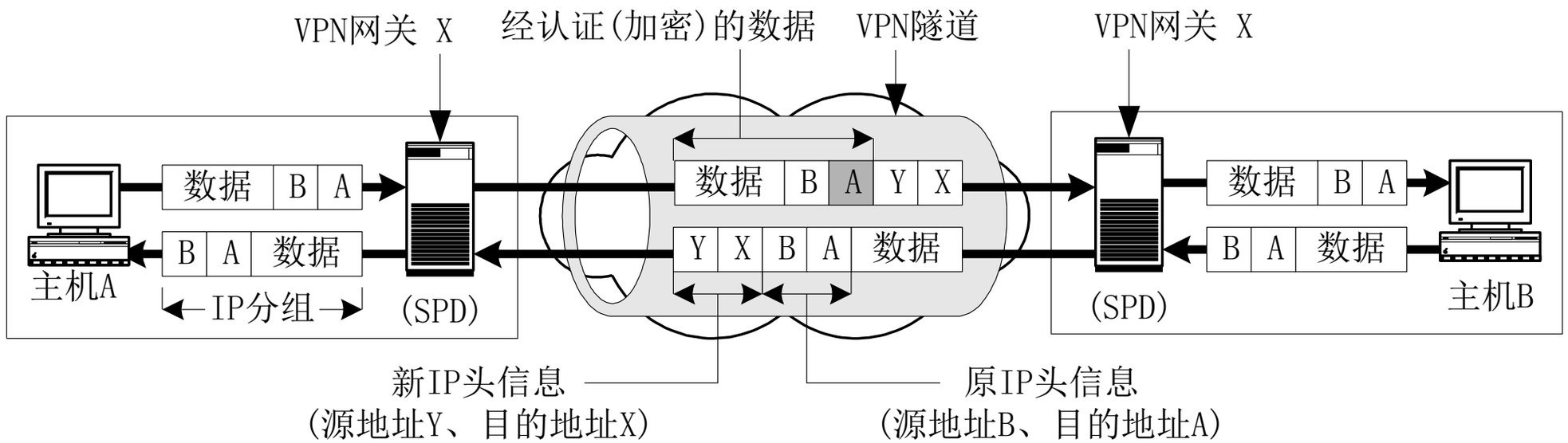


图 IPSec传输模式下的AH、ESP数据封装格式

IPSec隧道模式



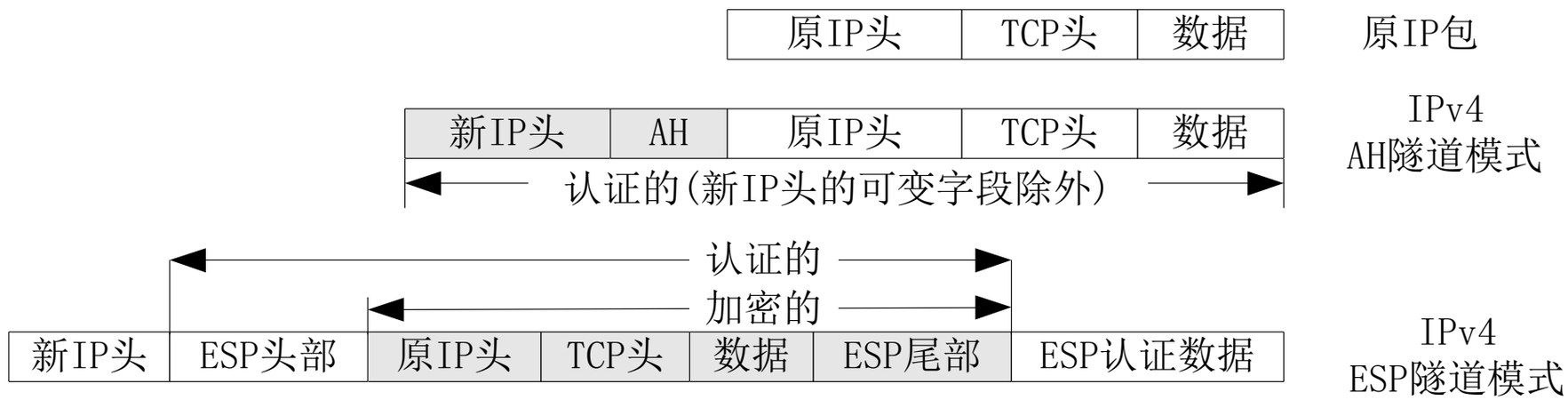


图 IPSec隧道模式下的AH、ESP的数据封装格式

安全联盟数据库(SADB)

- SA(Security Association)是两个IPSec通信实体之间经协商建立起来的一种共同协定，它规定了通信双方使用哪种IPSec协议保护数据安全、应用的算法标识、加密和验证的密钥取值以及密钥的生存周期等等安全属性值。

安全联盟数据库(SADB)（2）

- 安全联盟常用参数：
  - 加密及验证密钥。
  - 密码算法在系统中的标识。
  - 序列号，32位的字段，在处理外出的数据包时，一个SA被应用一次，它的序列号号字段就递增一，并被填充到数据包的IPSec头中，接收方可以利用此字段进行抗重播攻击。
  - 抗重播窗口。接收方使用滑动窗口算法来进行对恶意主机重复发出的数据包进行检测。
  - 生存周期。规定了该SA的有效使用周期，可以按照建立至今的时间或者处理的流量来计算。
  - 实施模式。即通道模式还是传输模式。
  - IPSec隧道目的地址。
  - 安全参数索引(SPI)。参与唯一标识某SA。

安全策略数据库(SPD)（1）

- SP是一个描述规则，定义了对什么样的数据流实施什么样的安全处理，至于安全处理需要的参数在SP指向的一个结构SA(安全联盟)中存储。

- SP描述：对本地子网和远程网关后的子网间的Telnet通信流，实施ESP通道保护，采用3DES加密算法和HMAC-SHA1验证算法。

安全策略数据库(SPD)（2）

- 系统中的安全策略组成了SPD,每个记录就是一条SP,定义类似上例中的描述规则，一般分为应用IPSec处理、绕过、丢弃。

- 从通信数据包中，可以提取关键信息填充到一个称为“选择符”的结构中去，包括目标IP、源IP、传输层协议、源和目标端口等等。然后利用选择符去搜索SPD，找到描述了该通信流的SP。

IPSec流程——数据包输出处理

- 数据包被从网络设备发送出去之前，截取到IP包，然后从中提取选择符信息，依据之搜索SPD，产生如下可能结果：
  - SP决定丢弃此包，于是直接丢弃，或者还可 以向源主机发送ICMP信息；
  - SP决定通过此包，直接将数据包投放到网络设备的发送队列；
  - SP决定应用IPSec，此时SP要么指向一个SA,可以根据它进行安全处理，要么需要的SA不存在，则触发IKE模块协商建立SA，协商周期内数据包进入等待队列等待协商完成，若协商超时，也会丢弃该包。

IPSec流程——数据包输入处理

- 系统收到IP包后，判断如果是IPSec包，则从头部取到<src_ip,protocol,SPI>，搜索SADB。
  - 若找不到SA，丢弃包；
  - 若找到，根据其进行解封装，得到去通道化后的原始IP包，再从原始IP包中提取选择符，搜索到SPD中某一条目，检查收到包的安全处理是否符合描述规则，不符合则丢弃包，符合则转入系统IP协议栈进行后继处理。

IKE

- IKE： Internet Key Exchange, 互联网密钥交换
  - IKE基本情况
  - IKE交换格式
  - 第一阶段交换
  - 第二阶段交换

IKE基本情况

- 功能
  - 用IPSec保护数据包，必须首先建立一个IPSec的安全联盟，这个安全联盟可以手工建立，也可以动态由特定进程来创建。这个特定的进程就是Internet Key Exchange, 即IKE。IKE的用途就是在IPSec通信双方之间通过协商建立起共享安全参数及验证过的密钥，也就是建立安全联盟。
  - IKE协议是Oakley和SKEME协议的混合，在由ISAKMP规定的一个框架内运作，可以为多种需要安全服务的协议进行策略磋商和密钥建立，比如SNMPv3,OSPFv2,IPSec等。

密钥交换包格式(ISAKMP)

8	12	16	24	32
发起方Cookie				
应答方Cookie				
下一个载荷	主版本	次版本	交换类型	标志
消息ID				
报文长度				

- 发起方Cookie：启动SA建立、SA通知或SA删除的实体Cookie
- 应答方Cookie：响应SA建立、SA通知或SA删除的实体Cookie
- 下一个载荷：信息中的Next Payload字段类型
- 主、次版本：使用的ISAKMP协议的主、次要版本
- 交换类型：正在使用的交换类型
- 标志：为ISAKMP交换设置的各类选项
- 消息ID：唯一的消息标识符，用来识别第二阶段的协议状态
- 报文长度：全部信息（头+有效载荷）长（八位）

密钥交换包格式(ISAKMP)

- 载荷类型
  - 安全联盟载荷，
  - 转码载荷表示协商时供对方选择的一组安全属性字段的取值，比如算法，安全联盟的存活期，密钥长度等等。
  - 密钥交换载荷，表示了实施密钥交换必需的信息。 散列载荷，是一个散列函数的运算结果值。
  - nonce载荷，是一串伪随机值，用以衍生加密材料。
  - 证书载荷，在身份验证时向对方提供证书。
  - 证书请求载荷。

密钥交换的两个阶段

- 阶段一交换(phase1 exchange)：在“阶段一”周期里，两个IKE实体建立一个安全的，经验证的信道进行后续通信，要建立这样的安全信道，双方会建立一对ISAKMP安全联盟。阶段一交换可以用**身份保护模式(也叫主模式)或野蛮模式**来实现，而这两种模式也仅用于阶段一中。

- 阶段二交换(phase2 exchange)：“阶段二”周期里，IKE实体会在阶段一建立起来的安全信道中，为某种进程协商和产生需要的密钥材料和安全参数，在VPN实现中，就是**建立IPSec安全联盟**。快速模式交换可用来实现阶段二交换并且仅用于此阶段中。

IKE阶段一

- 1. 主模式交换
  - 主模式交换提供了身份保护机制，经过三个步骤，共交换了六条消息。三个步骤分别是策略协商交换、Diffie Hellman共享值、nonce交换以及身份验证交换。

- 2. 野蛮模式交换
  - 野蛮模式交换也分为三个步骤，但只交换三条消息：头两条消息协商策略，交换Diffie Hellman公开值必需的辅助数据以及身份信息；第二条消息认证响应方；第三条消息认证发起方，并为发起方提供在场的证据

主模式交换和野蛮模式交换

- 第一阶段的主要任务是建立IKE SA，为后面的交换提供一个安全通信信道。使用主模式交换和野蛮模式交换。这两种模式都可以建立SA，两者的区别在于野蛮模式只用到主模式一半的消息，因此野蛮模式的协商能力受到限制的，而且它不提供身份保护。

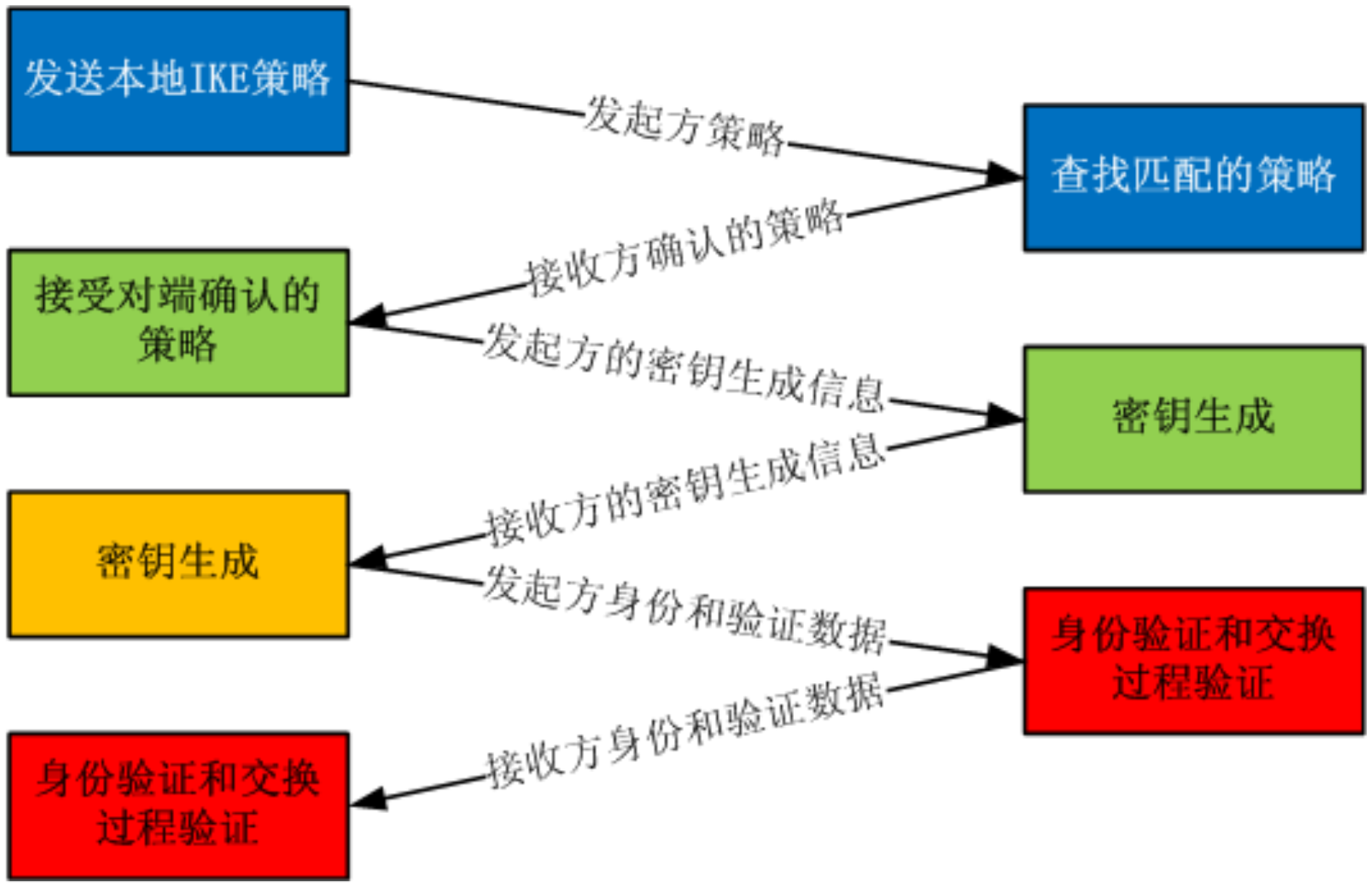
- 但是野蛮模式可以有一些特殊用途，比如远程访问等。另外如果发起者已经知道响应者的策略，利用野蛮模式可以快速的建立IKE SA。主模式和野蛮模式都允许4中不同的验证方法：（1）预共享密钥（2）DSS数字签名、（3）RSA数字签名（4）交换加密。

IKE阶段一协商流程简图

SA交换

密钥交换

ID及验证交换



交换流程（1）
(阶段一身份保护模式)

Intiator	Direction	IR ⇐	Note
(1)HDR;SA	=>		发起协商
(2)	<=	HDR;SA	对 SA 达成一致
(3)HDR;KE;NONCE	=>		交换 DH 公开值
(4)	<=	HDR;KE;NONCE	交换 DH 公开值
(5)HDR*;IDi;AUTH DATA	=>		发送验证身份数据，加密传送
(6)	<=	HDR*;IDr;AUTH DATA	发送验证身份数据，加密传送

交换流程（2）阶段一说明

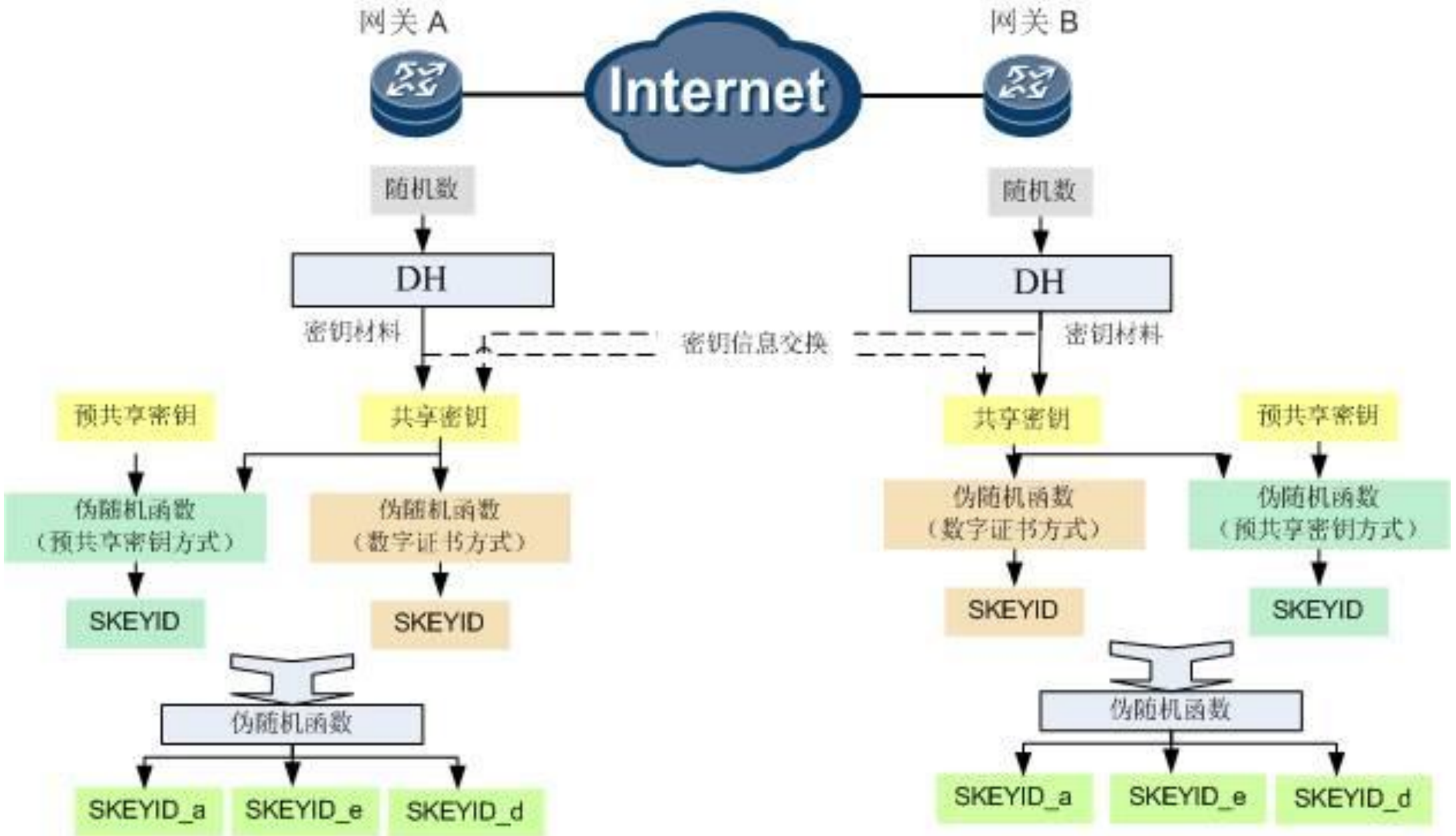
- 在消息(1)中，发起者生成他认为适当的安全提案列表，提交给响应方。消息(2)中，响应者与本地策略进行匹配和选择之后，将最终决定的安全联盟内容同样用相应载荷回送发起者。

- 在消息(3)、(4)中，发起者和响应者交换DH公开值，和随机信息串nonce，在第四步完成时，双方已经可以经计算得出共享的DH公共值，以及各自计算出SKEYID和相关衍生密钥。

- 消息(5)和消息(6)中，双方使用前两步得出的加密、验证算法和密钥保护传输的数据。

- 当采用数字签名的身份验证方法时，消息(5)和(6)可以包含证书载荷，将自己的公钥证书发给对方，验证数据AUTH DATA就是数字签名的运算结果，在这里数字证书也可以是从有效的远程有效的认证中心通过LDAP、DNSSEC等协议获得。

交换流程（2）阶段一说明



DH交换及密钥生成

Diffie-Hellman密钥交换

D-H交换的安全性源于在有限域上计算离散对数比计算指数更为困难。

DH 交换的原理简述如下。

通信双方为 Alice 和 Bob,双方约定好一个参数组，其中指定了运算中使用的质数 p 和底数 g,Alice 和 Bob 分别选择一个随机的私人数字 a 和 b,然后两人分别计算:

Alice: $A = g^a \bmod p$ Bob: $B = g^b \bmod p$

通过开放信道，两人交换 A 和 B,然后再次进行乘幂运算，使用收到的数字作底数，生成共享的一个公共值:

$$B^a \bmod p = g^{ab} \bmod p = A^b \bmod p$$

在交换运算过程中，只有私人数字 a 和 b 需要保密，其他数字 A, B, g, p 都不必保密。交换双方生成共享密钥后，就可以用之来保护后续的通信，这样，原来不安全的信道就变得安全了。

阶段一野蛮模式

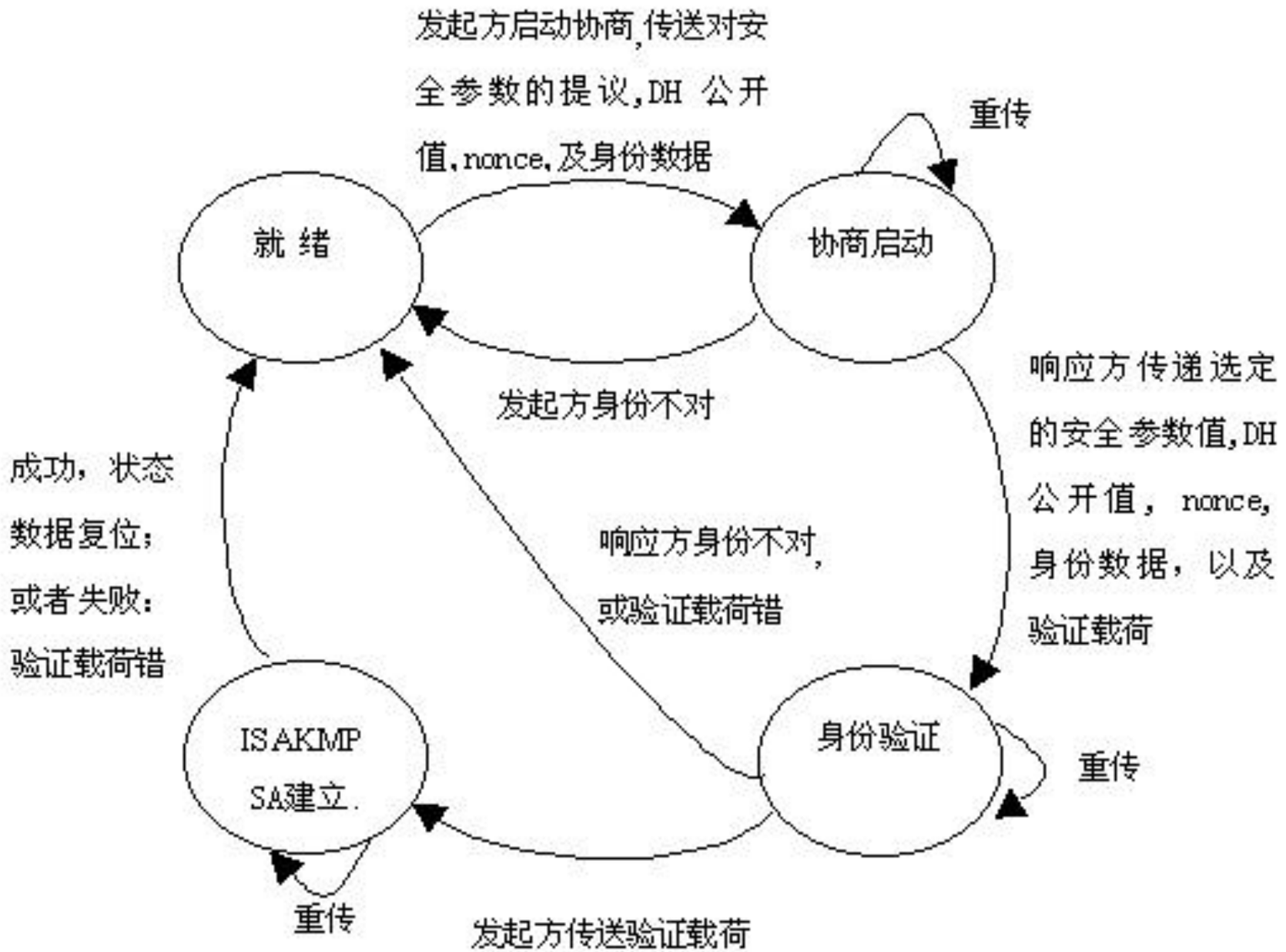


图 3. 野蛮模式交换状态转换图

野蛮模式

野蛮模式时IKEv1阶段1的协商过程：

- 网关A发送ISAKMP消息，携带建立IKE SA所使用的参数、与密钥生成相关的信息和身份验证信息。
- 网关B对收到的第一个数据包进行确认，查找并返回匹配的参数、密钥生成信息和身份验证信息。
- 网关A回应验证结果，并建立IKE SA。

与主模式相比，野蛮模式的优点是建立IKE SA的速度较快。但是由于密钥交换与身份认证一起进行，野蛮模式无法提供身份保护。

交换流程（3）
(阶段二快速模式)

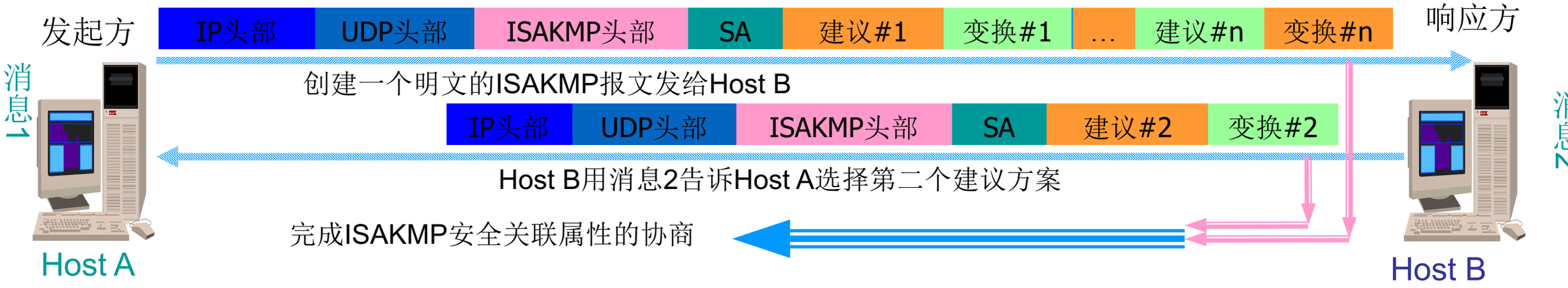
Initiator	Director	R e s
(1)HDR*;HASH(1);SA;Ni [;KE][;IDci;Idcr]	=>	
(2)	<=	(2)HDR*;HASH(2);SA ;Nr [;KE][;IDci;Idcr]
(3)HDR*;HASH(3)	=>	

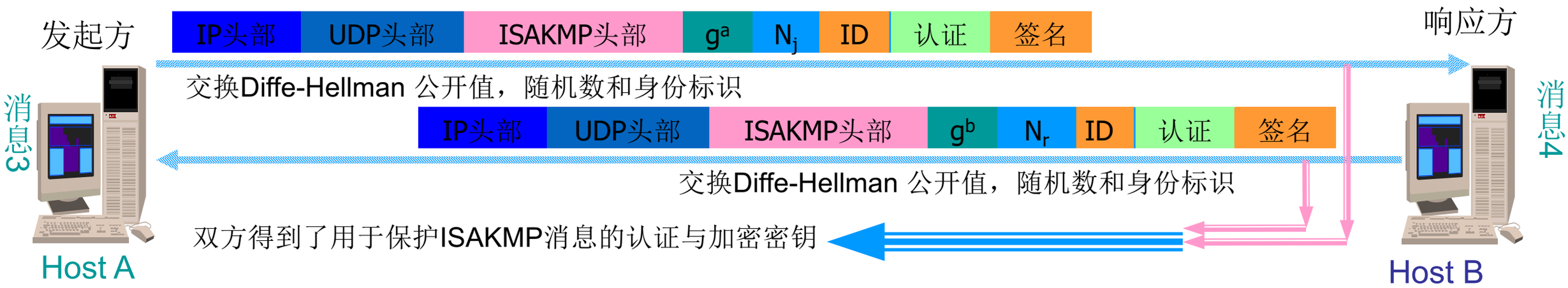
交换流程（4）阶段二说明

所有消息从ISAKMP头之后都是加密传输的，并且在消息头之后紧跟了散列值进行验证。如果使用了完美向前加密(PFS)，则消息交换中还包含一次DH交换的公开值载荷KE,身份载荷表示的是要保护的通信流的源和目的，通常是子网内的主机或主机的集合。

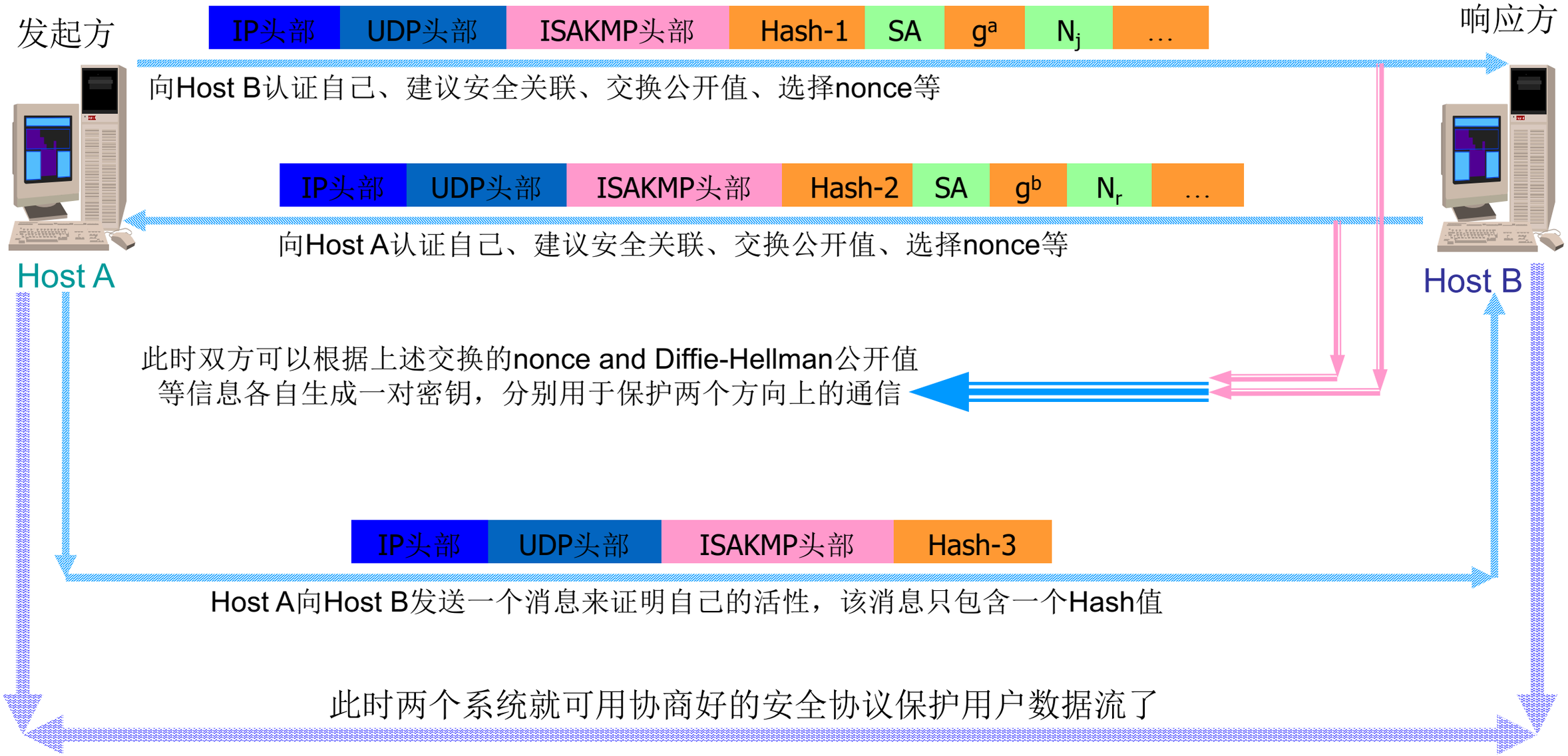
在前两个消息交换完成后，双方可以计算出共享的密钥材料，这将是最终提供给IPSec模块的密钥信息。

ISAKMP/Oakley 阶段一工作原理





ISAKMP/Oakley 阶段二工作原理



虚拟专用网——

传输层安全

SSL

- SSL
  - SSL基本情况
  - SSL协议体系
  - SSL记录层协议
  - SSL高层协议

SSL基本情况

- SSL： Secure Socket Layer 安全套接字层。1994年Netscape开发，专门用于保护Web通讯

- 版本和历史

  - 1.0，不成熟
  - 2.0，基本上解决了Web通讯的安全问题
  - 同时，Microsoft公司发布了PCT(Private Communication Technology)，并在IE中支持
  - 3.0，1996年发布，增加了一些算法，修改了一些缺陷
  - TLS 1.0(Transport Layer Security, 也被称为SSL 3.1)，1997年IETF发布了Draft，同时，Microsoft宣布放弃PCT，与Netscape一起支持TLS 1.0
  - 1999年，发布RFC 2246(The TLS Protocol v1.0)

SSL基本情况

- 协议的设计目标
  - 为两个通讯个体之间提供保密性和完整性(身份认证)
  - 互操作性、可扩展性、相对效率
- 为上层协议提的供安全性
  - 保密性
  - 身份认证和数据完整性

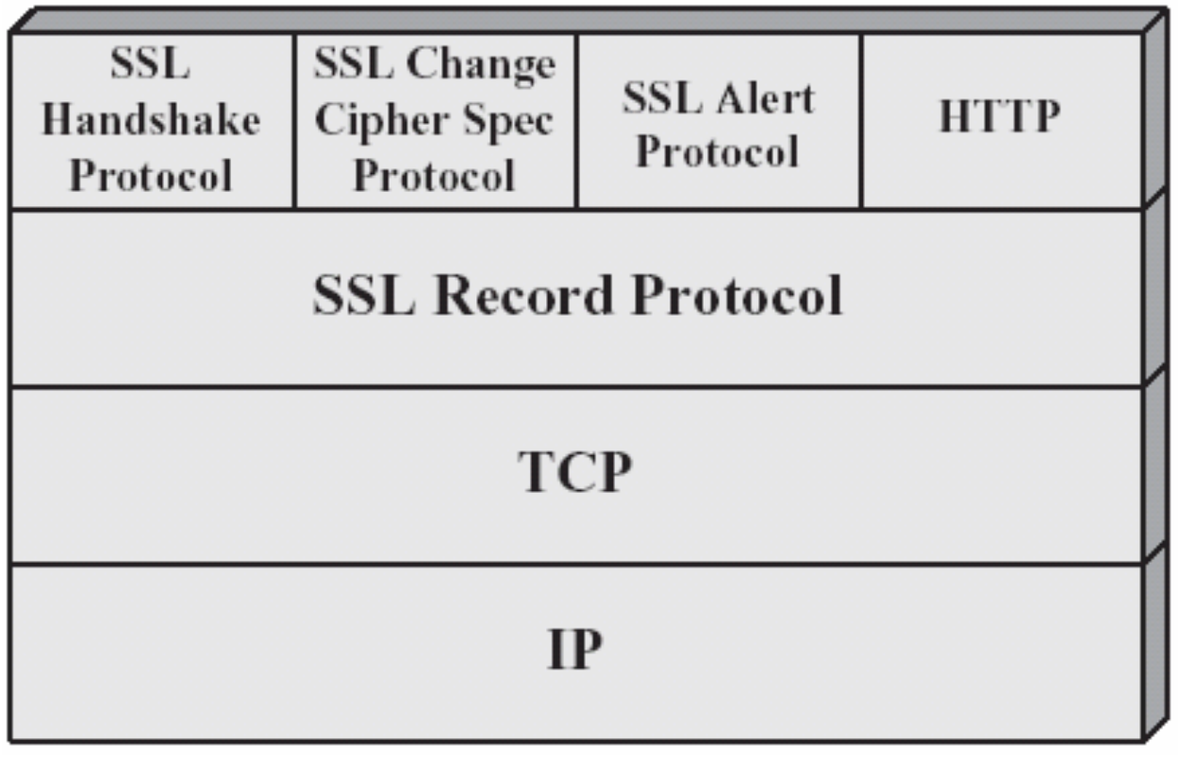
SSL基本情况

- SSL实现
  - OpenSSL, 实现了SSL(2,3), TLS(1.0)
    - Openssl —— a command line tool.
    - ssl(3) —— the OpenSSL SSL/TLS library.
    - crypto(3)—— the OpenSSL Crypto library.
    - **URL: http://www.openssl.org**
  - SSLeay
    - http://www2.psy.uq.edu.au/~ftp/Crypto/
  - Microsoft Win2k SSL implementation

SSL协议体系

- SSL被设计用来使用TCP提供一个可靠的端到端安全服务。

- 协议分为两层
  - 底层：SSL记录协议
  - 上层：SSL握手协议、SSL密码变化协议、SSL警告协议



SSL协议体系

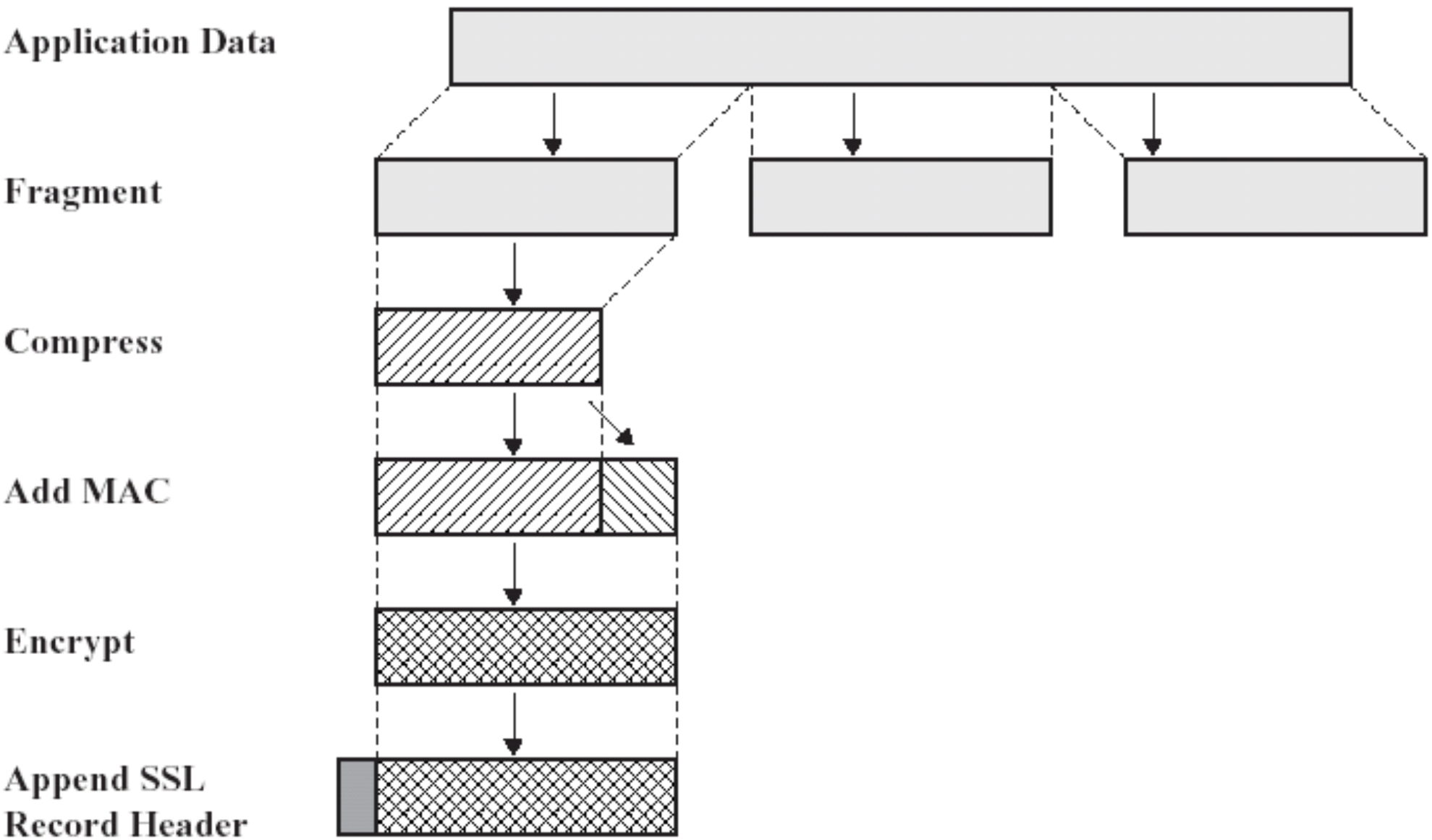
- SSL记录协议
  - 建立在可靠的传输协议(如TCP)之上，为更高层提供基本安全服务。特别是HTTP，它提供了Web的client/server交互的传输服务，可以构造在SSL之上。
  - 它提供连接安全性，有两个特点
    - 保密性，使用了对称加密算法
    - 完整性，使用HMAC算法
  - 用来封装高层的协议
- SSL Handshake Protocol, SSLChange Cipher Spec Protocol, SSL Alert Protocol是SSL的高层协议，用于管理SSL交换。

两个重要概念

- SSL连接（connection)
  - 一个连接是一个提供一种合适类型服务的传输（OSI分层的定义）
  - SSL的连接是点对点的关系
  - 连接是暂时的，每一个连接和一个会话关联
- SSL会话（session）
  - 一个SSL会话是在客户与服务器之间的一个关联。会话由Handshake Protocol创建。会话定义了一组可供多个连接共享的加密安全参数。
  - 会话用以避免为每一个连接提供新的安全参数所需昂贵的谈判代价。

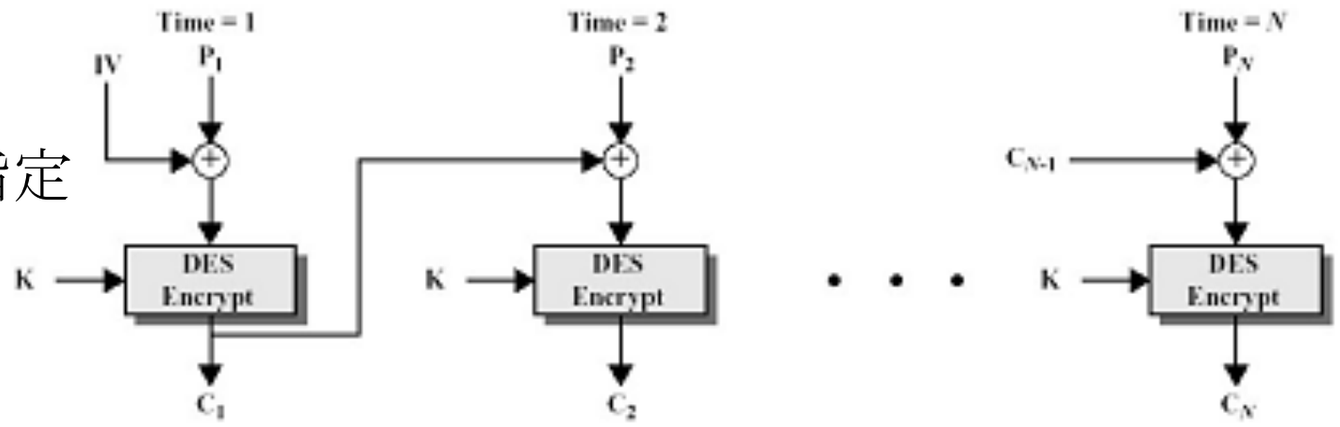
SSL记录层协议

- 记录层数据封装过程



SSL记录层协议

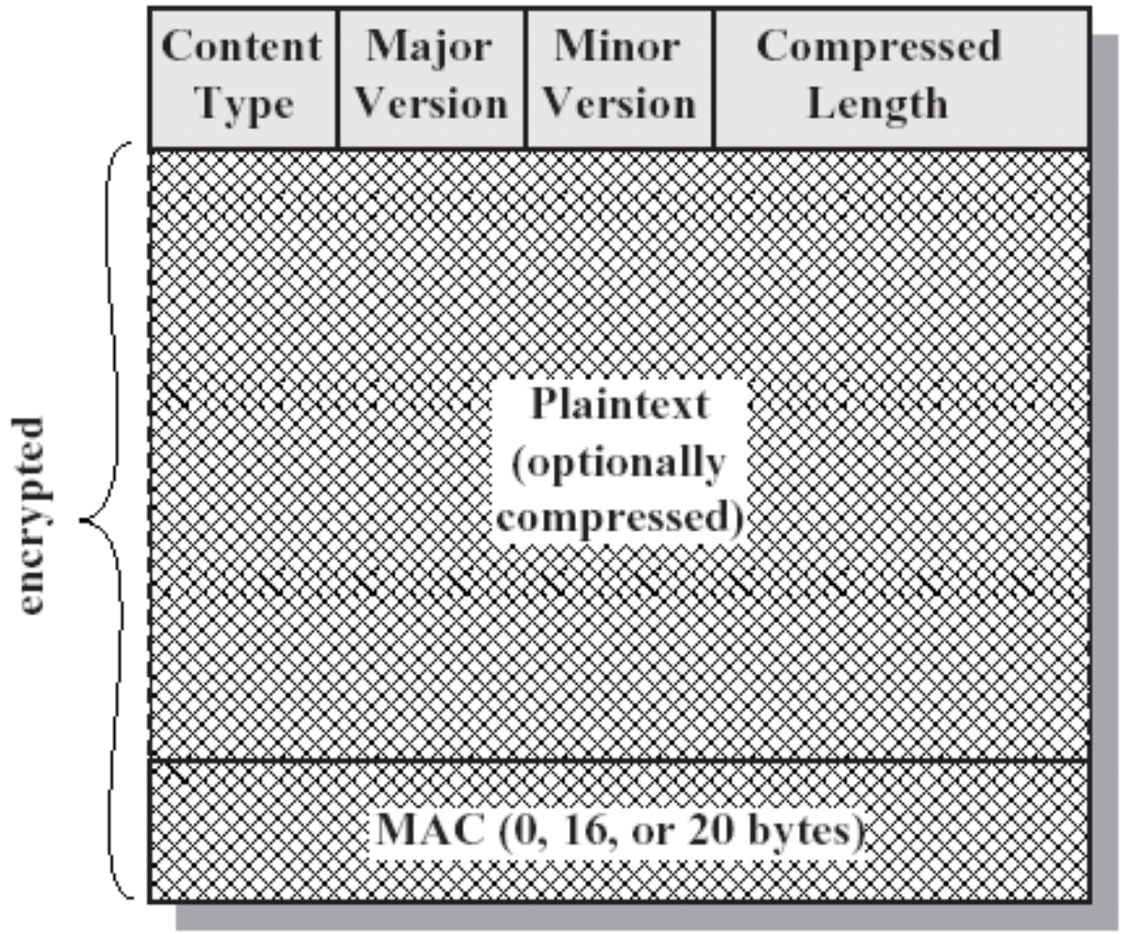
- 第一步，fragmentation
  - 上层消息的数据被分片成214字节大小的块，或者更小
- 第二步，compression(可选)
  - 必须是无损压缩，如果数据增加的话，则增加部分的长度不超过1024字节
- 第三步，计算消息认证码(MAC)
  - 计算公式：HMAC_hash(MAC_write_secret, seq_num || TLSCompressed.type ||TLSCompressed.version ||TLSCompressed.length || TLSCompressed.fragment)
- 第四步，encryption
  - 采用CBC，算法由cipher spec指定
  - 数据长度不超过2^{14}+2048字节



SSL记录层协议

- 结果：

```
struct {
        ContentType type;
            // 8位，上层协议类型
        ProtocolVersion version;
            //16位，主次版本
    uint16 length;
        //加密后数据的长度，不超过
 2^14+2048字节
        EncryptedData fragment;
        //密文数据
} TLSCiphertext;
```



SSL高层协议

- 握手协议（见后面）
- 密码变化协议(Change Cipher Spec Protocol)
  - 它位于TLS记录协议之上,它用到了TLS记录协议的处理过程
  - ContentType = 20
  - 协议只包含一条消息，一个字节 1
  - 用途：切换状态；把密码参数设置为当前状态；在握手协议中，当安全参数；协商一致后，发送此消息
  - 这条消息使得接收方改变当前状态读参数，使得发送方改变当前状态写参数

SSL高层协议

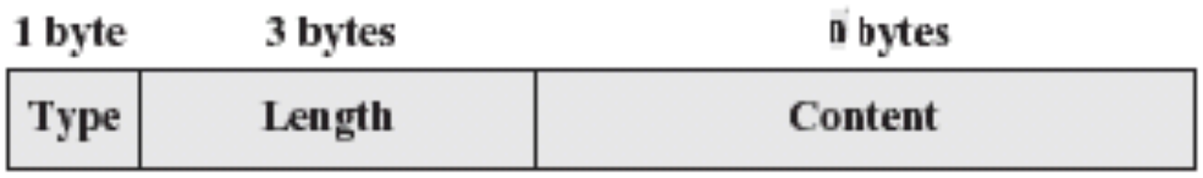
- 警告协议(Alert Protocol)
  - 位于TLS记录协议之上,也用到了TLS记录协议的处理过程
  - ContentType = 21
  - 协议数据包含两个字节：第一个字节为level，分别为warning(1)和fatal(2)两种情况；第二个字节为情况说明
  - Fatal类型的alert消息导致连接立即终止，此时，对应该会话的其他连接可以继续，但是会话标识符无效，以免
    利用此失败的连接来建立新的连接

SSL密钥交换

- 协议整体情况
- 各阶段交互
- 简化交互过程
- 会话密钥生成

SSL密钥交换——协议整体情况

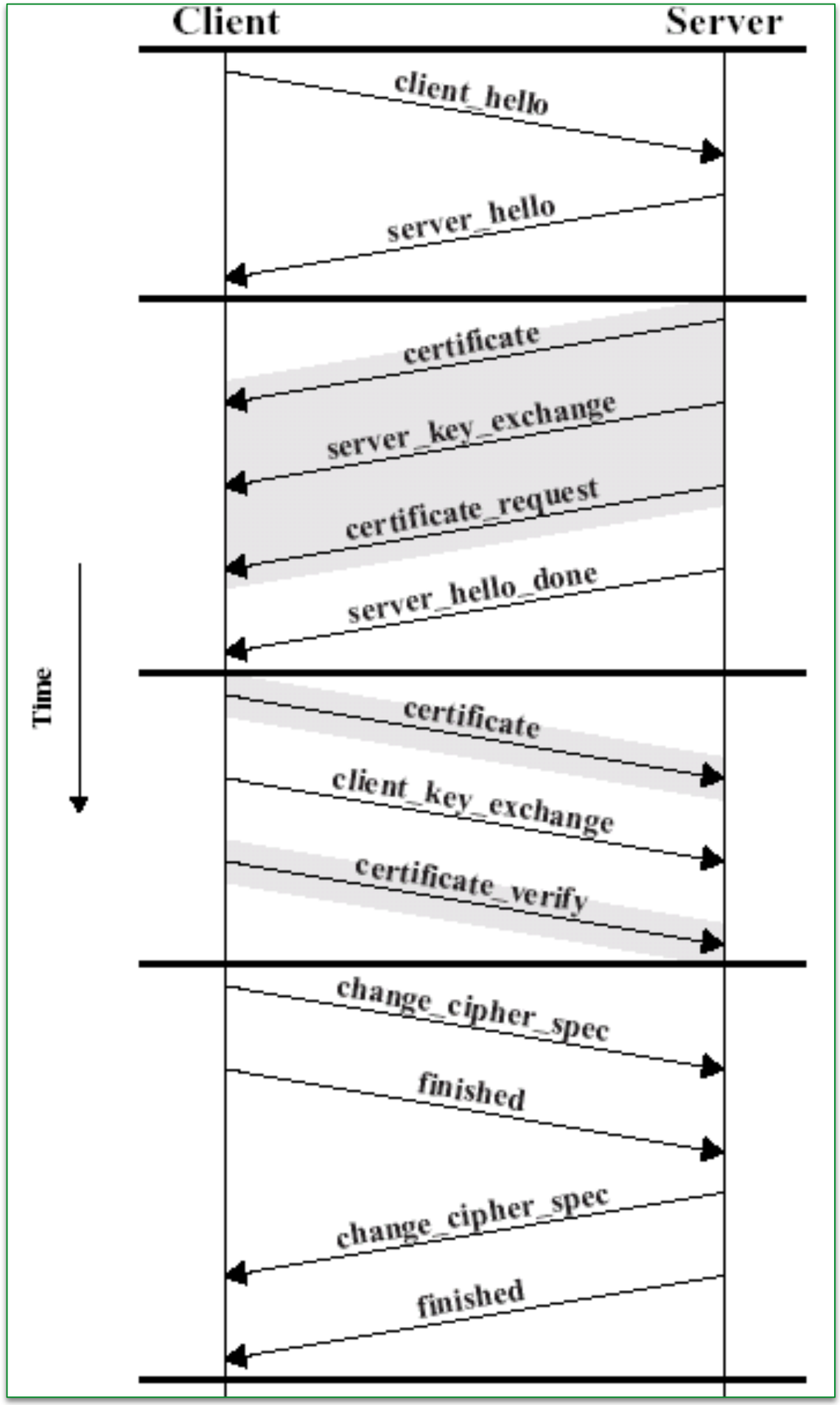
- 功能
  - 客户和服务器之间相互认证
  - 协商加密算法和密钥
  - 它提供连接安全性，有三个特点
    - 身份认证，至少对一方实现认证，也可以是双向认证
    - 协商得到的共享密钥是安全的，中间人不能够知道
    - 协商过程是可靠的
- 规范说明
  - 位于TLS记录协议之上，也用到了TLS记录协议的处理过程
  - ContentType = 22
  - 协议格式如图：



Handshake Protocol

整体流程

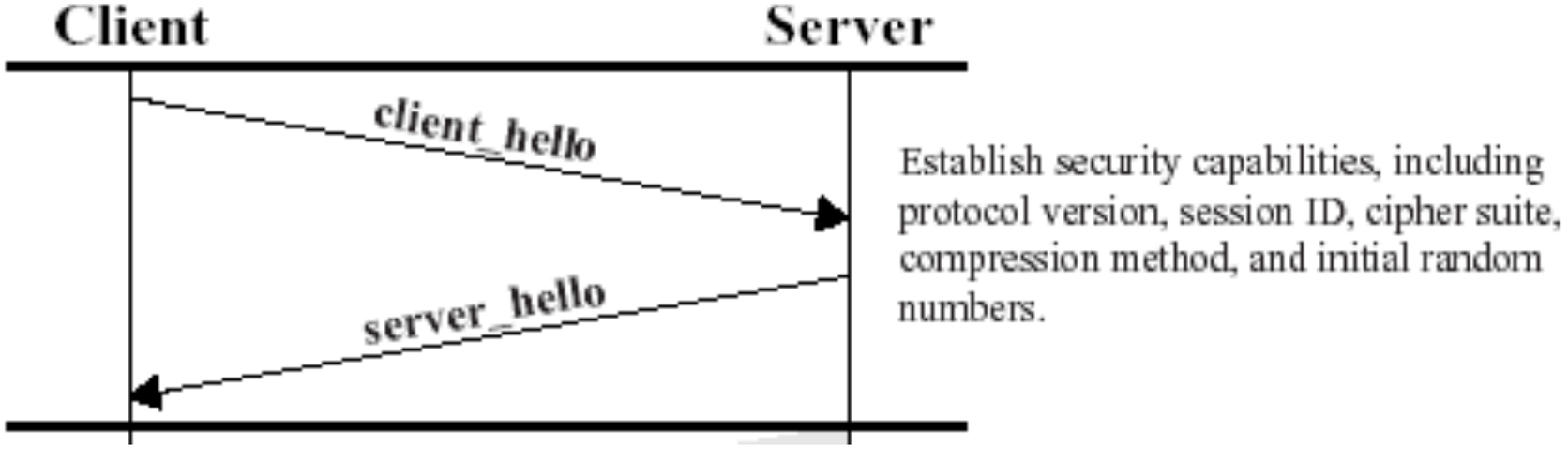
- (1)、交换Hello消息，对于算法、交换随机值等协商一致
- (2)、交换必要的密码参数，以便双方得到统一的premaster secret
- (3)、交换证书和相应的密码信息，以便进行身份认证
- (4)、产生master secret
- (5)、把安全参数提供给TLS记录层
- (6)、检验双方是否已经获得同样的安全参数



整体流程

流程中消息	相关参数
hello_request	Null
client_hello	版本，随机数，会话id，密码参数，压缩方法
server_hello	
certificate	X.509 v3证书链
server_key_exchange	参数，签名
certificate_request	类型，CAs
server_done	Null
certificate_verify	签名
client_key_exchange	参数，签名
finished	Hash值

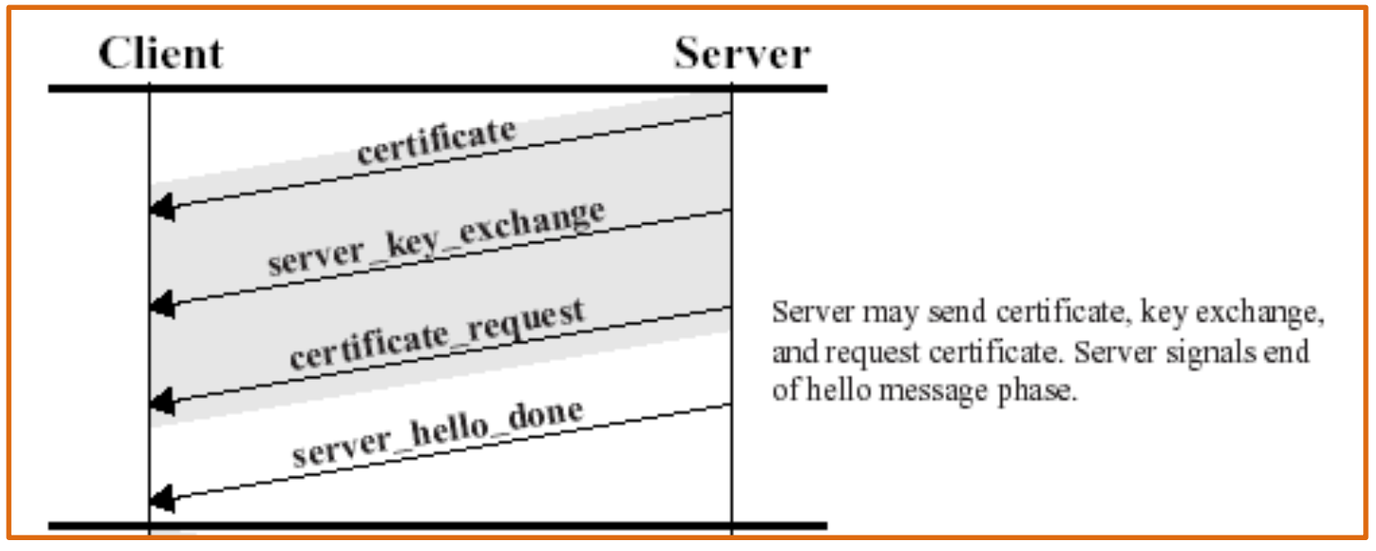
各阶段交互



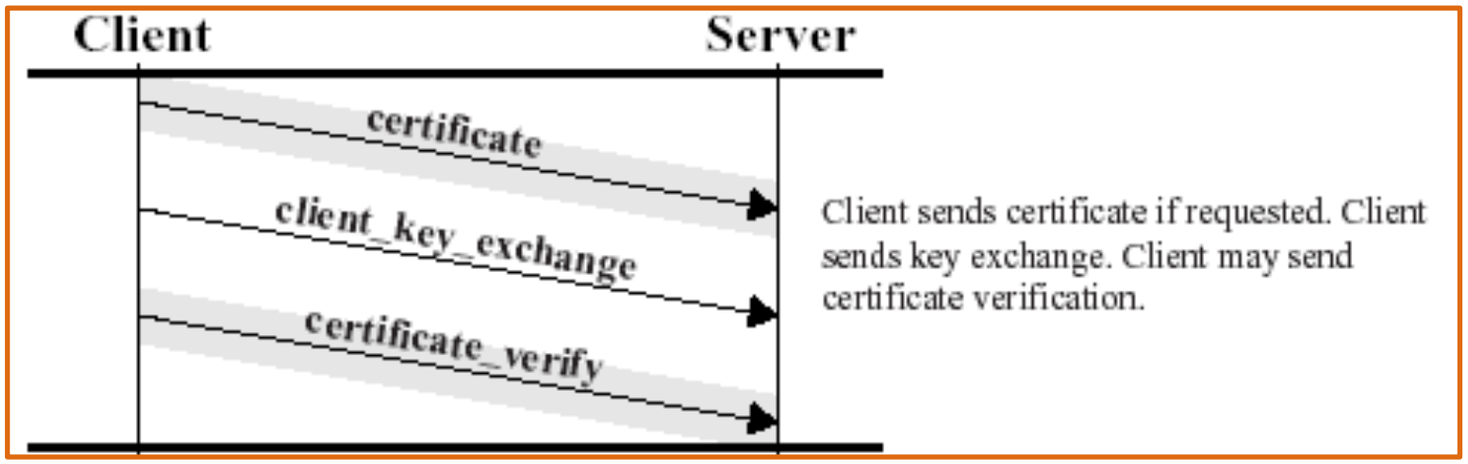
- 第一阶段交互
  - 客户发送一个client_hello消息，包括以下参数：
    版本、随机数(32位时间戳+28字节随机序列)、会话ID、客户支持的密码算法列表(CipherSuite)、客户支持的压缩方法列表
  - 然后，客户等待服务器的server_hello消息
  - 服务器发送server_hello消息，参数：
    客户建议的低版本以及服务器支持的最高版本、服务器产生的随机数、会话ID、服务器从客户建议的密码算法中挑出一套、服务器从客户建议的压缩方法中挑出一个

各阶段交互

- 第一阶段进一步说明
  - 关于会话ID
    - 客户方：若会话ID不等于0，希望基于这个会话来更新已有连接的安全参数，或者创建一个新的连接；否则，表示客户希望在一个新的会话上建立一个新的连接
    - 服务器：或者同意客户指定的会话ID，需要检查cache中的会话状态；或者返回一个新的会话ID
  - 关于CipherSuite
    - 第一个元素指定了密钥交换的方法，TLS支持以下一些方法：RSA、DH、EDH(Ephemeral Diffie-Hellman)、匿名的DH
    - 然后，指定以下信息：加密算法和类型(流还是分组密码算法)、HMAC算法（MD5还是SHA-1）、是否可出口、HashSize、Key Material、IV Size



- 第二阶段交互
  - 服务器发送自己的证书，消息包含一个X.509证书，或者一条证书链
    - 除了匿名DH之外的密钥交换方法都需要
  - 服务器发送server_key_exchange消息
    - 可选的，有些情况下可以不需要。只有当服务器的证书没有包含必需的数据的时候才发送此消息
    - 消息包含签名，被签名的内容包括两个随机数以及服务器参数
  - 服务器发送certificate_request消息
    - 非匿名server可以向客户请求一个证书
    - 包含证书类型和CAs
  - 服务器发送server_hello_done，然后等待应答



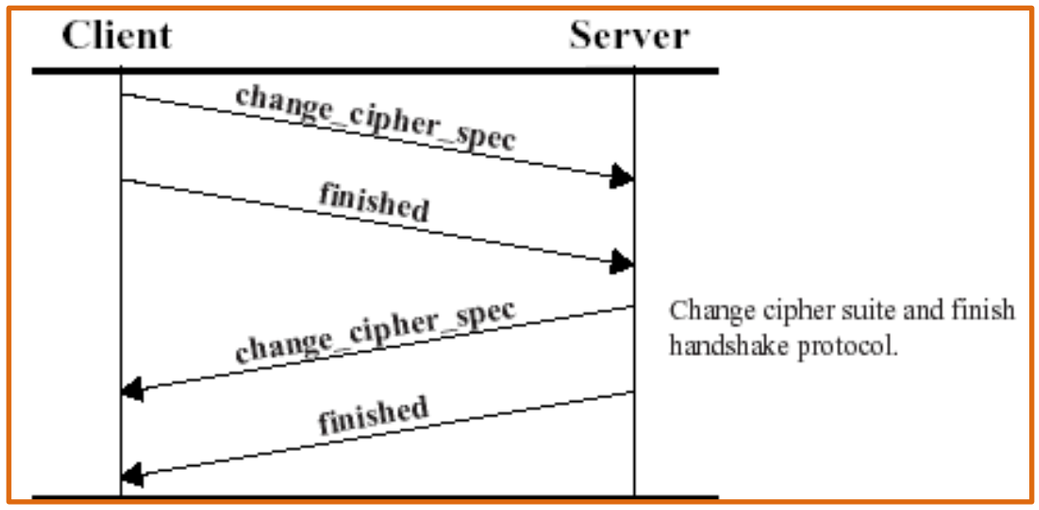
- 第三阶段交互

  - 客户收到server_done消息后，它根据需要检查服务器提供 的证书，并判断server_hello的参数是否可以接受，如果都没有问题的话，发送一个或多个消息给服务器

  - 如果服务器请求证书的话，则客户首先发送一个certificate消息，若客户没有证书，则发送一个no_certificate警告

  - 然后客户发送client_key_exchange消息，消息的内容取决于密钥交换的类型

  - 最后，客户发送一个certificate_verify消息，其中包含一个签名，对从第一条消息以来的所有握手消息的HMAC值(用master_secret)进行签名

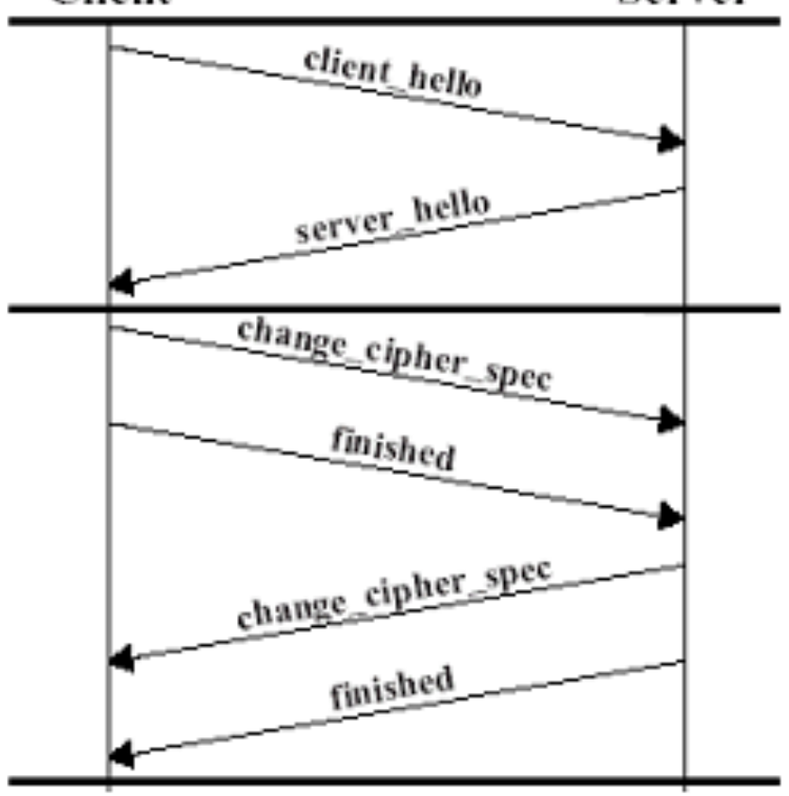


- 第四阶段交互－建立起一个安全的连接
  - 客户发送一个change_cipher_spec消息，并且把协商得到的CipherSuite拷贝到当前连接的状态之中
  - 然后，客户用新的算法、密钥参数发送一个finished消息，这条消息可以检查密钥交换和认证过程是否已经成功。其中包括一个校验值，对所有以来的消息进行校验。
  - 服务器同样发送change_cipher_spec消息和finished消息。
  - 握手过程完成，客户和服务器可以交换应用层数据

简化交互过程

- SSL可通过重用一个TLS会话ID使用简化的交互过程

  - 客户和服务器在交换hello消息中，客户要求重用已有的TLS会话，服务器同意使用cache中的会话
    * session id
  - 跳过第二第三阶段，直接把TLS会话中的参数传递给TLS记录层



会话密钥生成

- TLS记录协议需要：CipherSuite, master secret, and the client and server random values
- 在hello消息中，交换随机数以及各种算法
- 对于各种密钥交换算法，从pre_master_secret计算得到master_secret，然后从内存中删除
- Master_secret总是48字节长，而pre_master_secret长度不定，取决于密钥交换算法
- 两类密钥交换算法：
  - RSA，客户产生一个48字节的pre_master_secret，然后通过服务器的公钥传递给服务器
  - Diffie-Hellman，双方协商得到的密钥被用作pre_master_secret

问题和讨论