网络安全——

# 2025-2026第一学期

北京邮电大学

郑康锋

kfzheng@bupt.edu.cn

网络安全——

# 网络安全概述

# 信息保障阶段

- 信息保障技术框架IATF：由美国国家安全局制定，提出“纵深防御策略”DiD（Defense-in-Depth Strategy）。

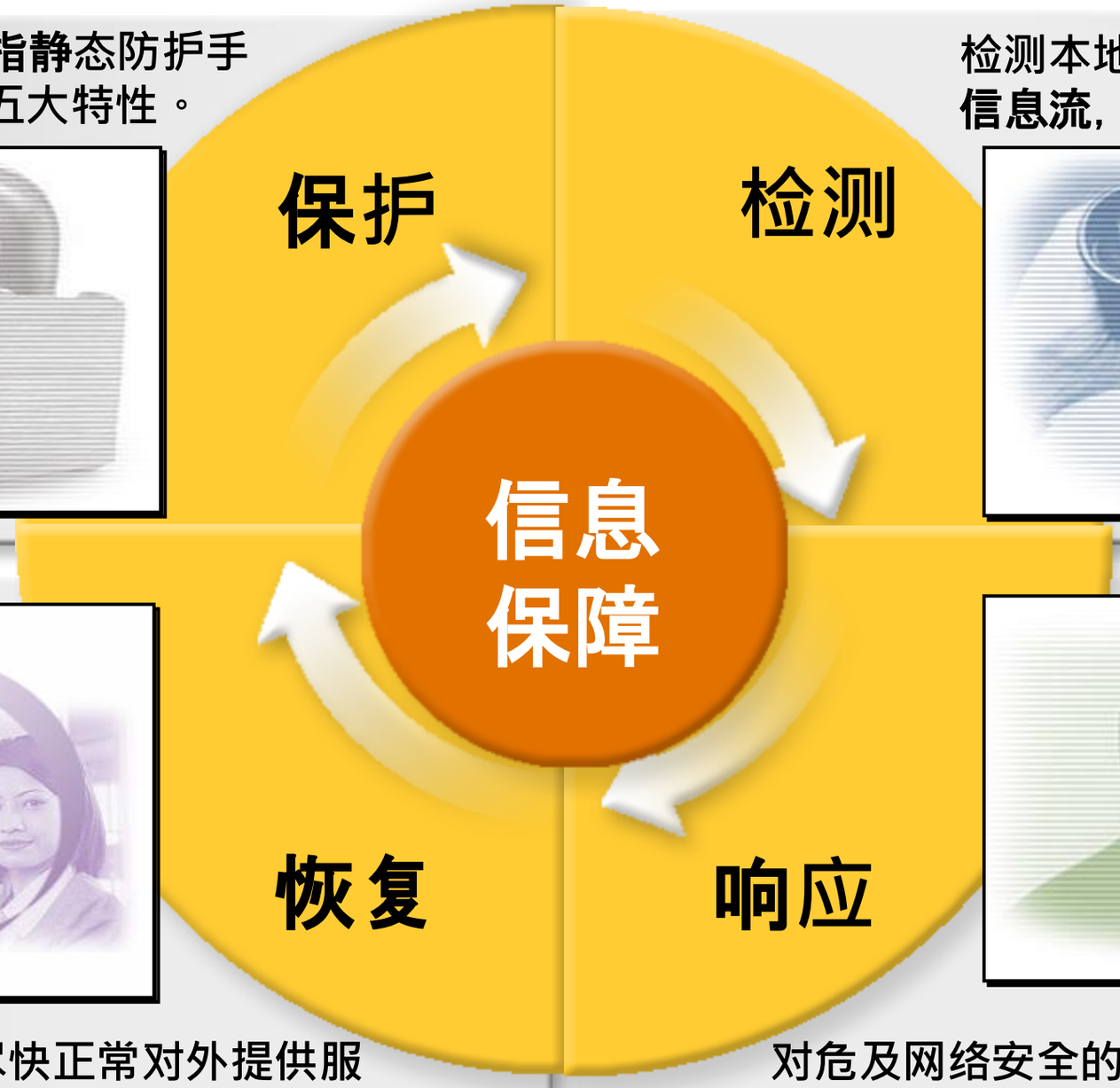
- 在信息保障的概念下，信息安全保障的PDRR模型的内涵已经超出了传统的信息安全保密，而是保护（Protection）、检测（Detection）、响应（Reaction）和恢复（Restore）的有机结合。

- 信息保障阶段不仅包含安全防护的概念，更重要的是增加了主动和积极的防御观念。

# PDRR安全模型

采用一切手段（主要指静态防护手段）保护信息系统的五大特性。

检测本地网络的安全漏洞和存在的非法信息流，从而有效阻止网络攻击

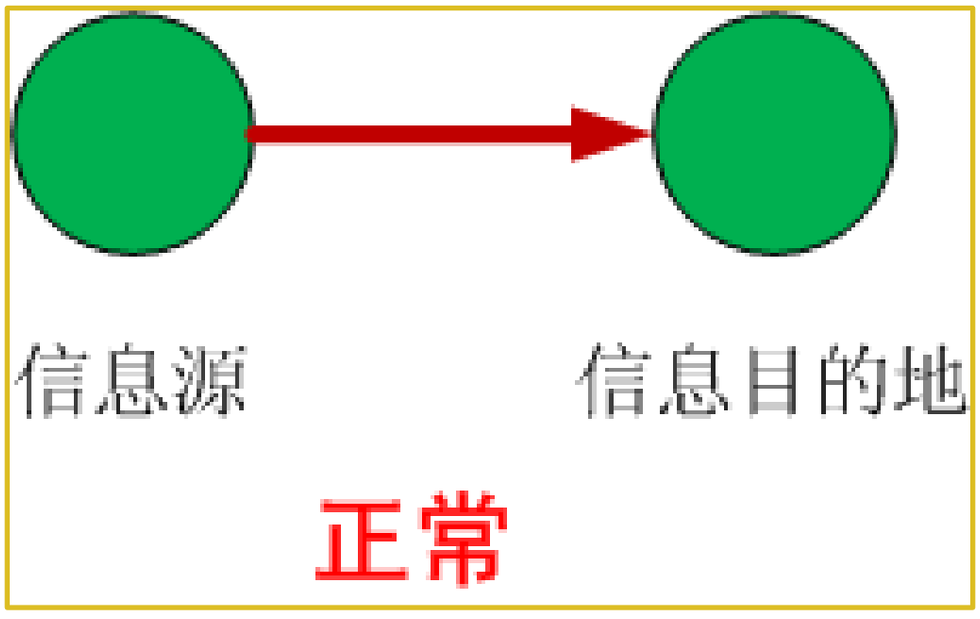



及时恢复系统，使其尽快正常对外提供服务，是降低网络攻击造成损失的有效途径

对危及网络安全的事件和行为做出反应，阻止对信息系统的进一步破坏并使损失降到最低

# 攻击类型

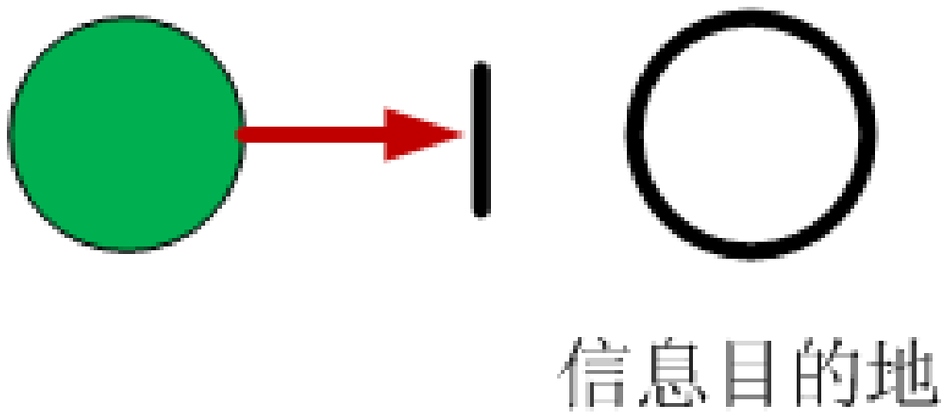
攻击的类型。从安全属性来看，攻击类型可分为4类：阻断攻击、截取攻击、篡改攻击、伪造攻击；




上图是从源站到目的站的正常信息流。

# 攻击类型

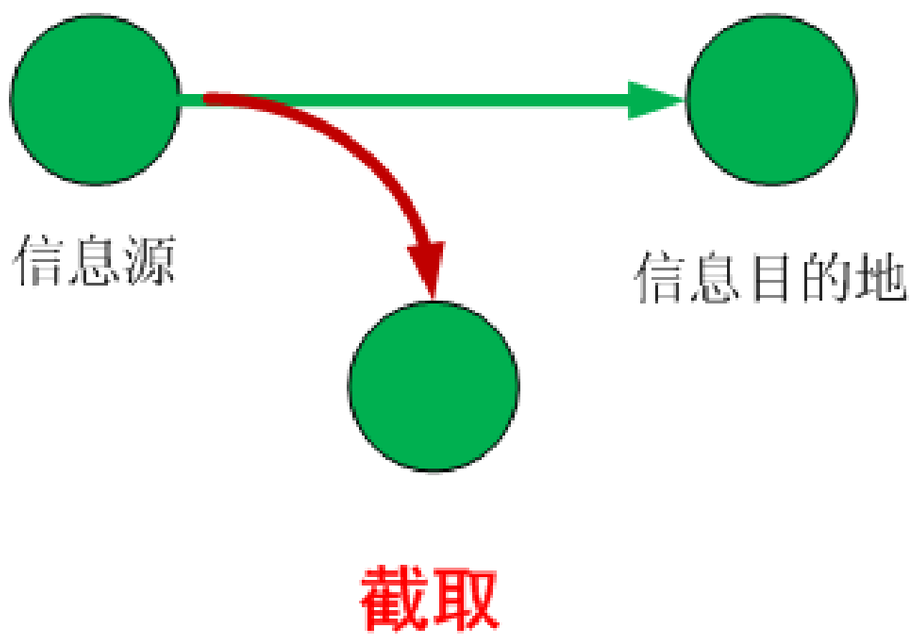
- (1)阻断攻击
- 阻断攻击使系统的资产被破坏，无法提供用户使用，这是一种针对可用性的攻击。例如，破坏硬盘之类的硬件，切断通信线路，使文件管理系统失效等。




阻断

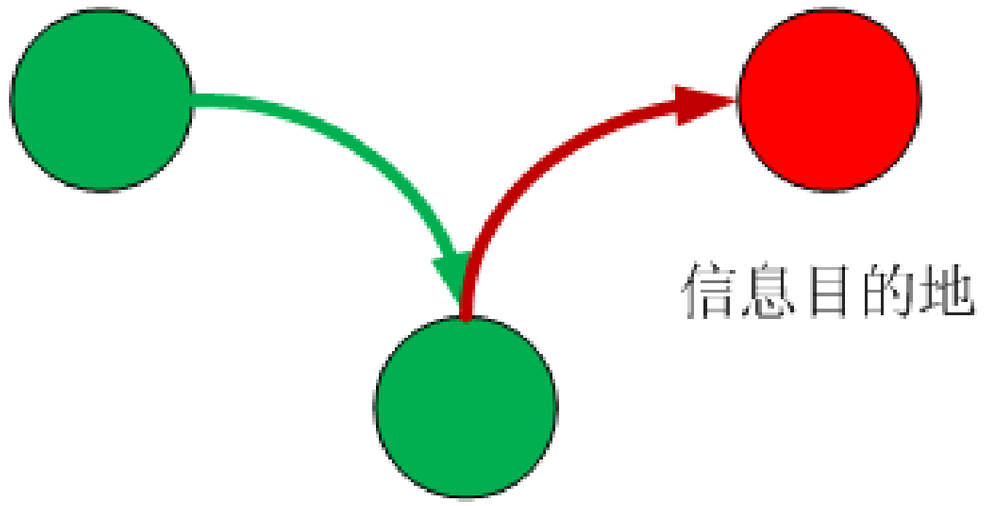
# 攻击类型

- (2)**截取攻击**
- 截取攻击可使非授权者得到资产的访问，这是一种针对机密性的攻击。非授权者可以是一个人、一个程序或一台计算机，例如，通过窃听获取网上数据以及非授权的复制文件和程序。



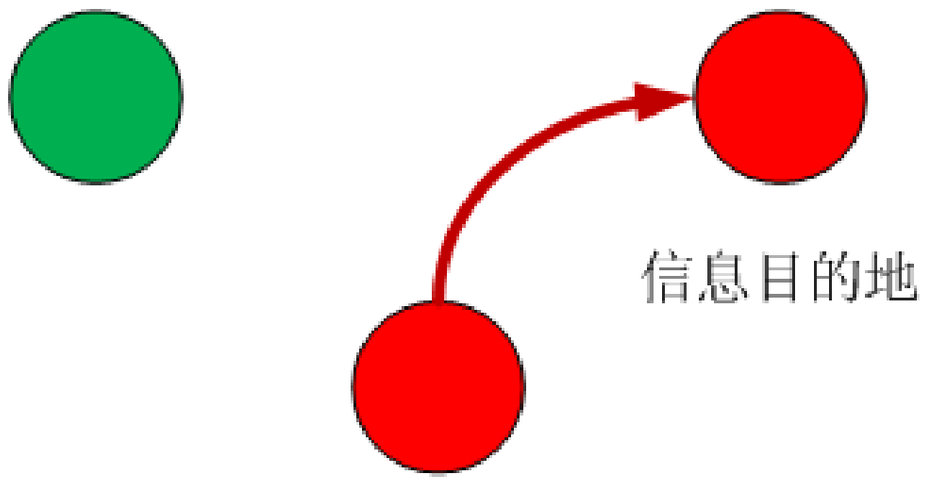
# 攻击类型

- (3)篡改攻击
- 篡改攻击是非授权者不仅访问资产，而且能修改信息，这是一种针对完整性的攻击。例如，改变数据文件的值，修改程序以及在网上正在传送的报文内容。



# 攻击类型

- (4)**伪造攻击**
- 伪造攻击是非授权者在系统中插入伪造的信息，这是一种针对真实性的攻击。例如：在网上插入伪造的报文，或在文件中加入一些记录。




**伪造**

# 网络信息安全服务

- **机密性服务**：提供信息的保密。
- **完整性服务**：提供信息的正确性。
- **可用性服务**：提供的信息是可用的。
- **可审性服务**：本身不针对攻击提供保护，需与其它服务结合。

网络安全——

# 网络攻击

网络攻击——

# 恶意代码

# 木马的定义

- 病毒、蠕虫和木马的比较

| 特性 | 病毒 | 蠕虫 | 木马 |
|---|---|---|---|
| **传染性** | 强 | 很少 | 很少 |
| **传播能力** | 强 | 极强 | 一般 |
| **感染对象** | 文件 | 进程 | 进程 |
| **主要传播方式** | 文件 | 网络 | 网络 |
| **破坏性** | 强 | 强 | 很少 |
| **隐蔽性** | 强 | 强 | 极强 |
| **顽固性** | 较强 | 较强 | 极强 |
| **欺骗性** | 一般 | 一般 | 强 |
| **主要攻击目的** | 破坏数据和信息 | 耗尽计算机资源 | 窃取信息、提供后门 |

# 利用CreateRemoteThread()函数注入

- 代码注入的方式很多，最著名的当属CreateRemoteThread()这个API函数，它可以在其它进程的地址空间中开启远程线程。它的原型如下：

```
HANDLE CreateRemoteThread(
  HANDLE hProcess,
  LPSECURITY_ATTRIBUTES lpThreadAttributes,
  SIZE_T dwStackSize,
  LPTHREAD_START_ROUTINE lpStartAddress,
  LPVOID lpParameter,
  DWORD dwCreationFlags,
  LPDWORD lpThreadId
);
```

- 它与CreateThread()函数很相似，唯一的区别是多了一个参数hProcess，即目标进程的句柄。

# 利用CreateRemoteThread()函数注入

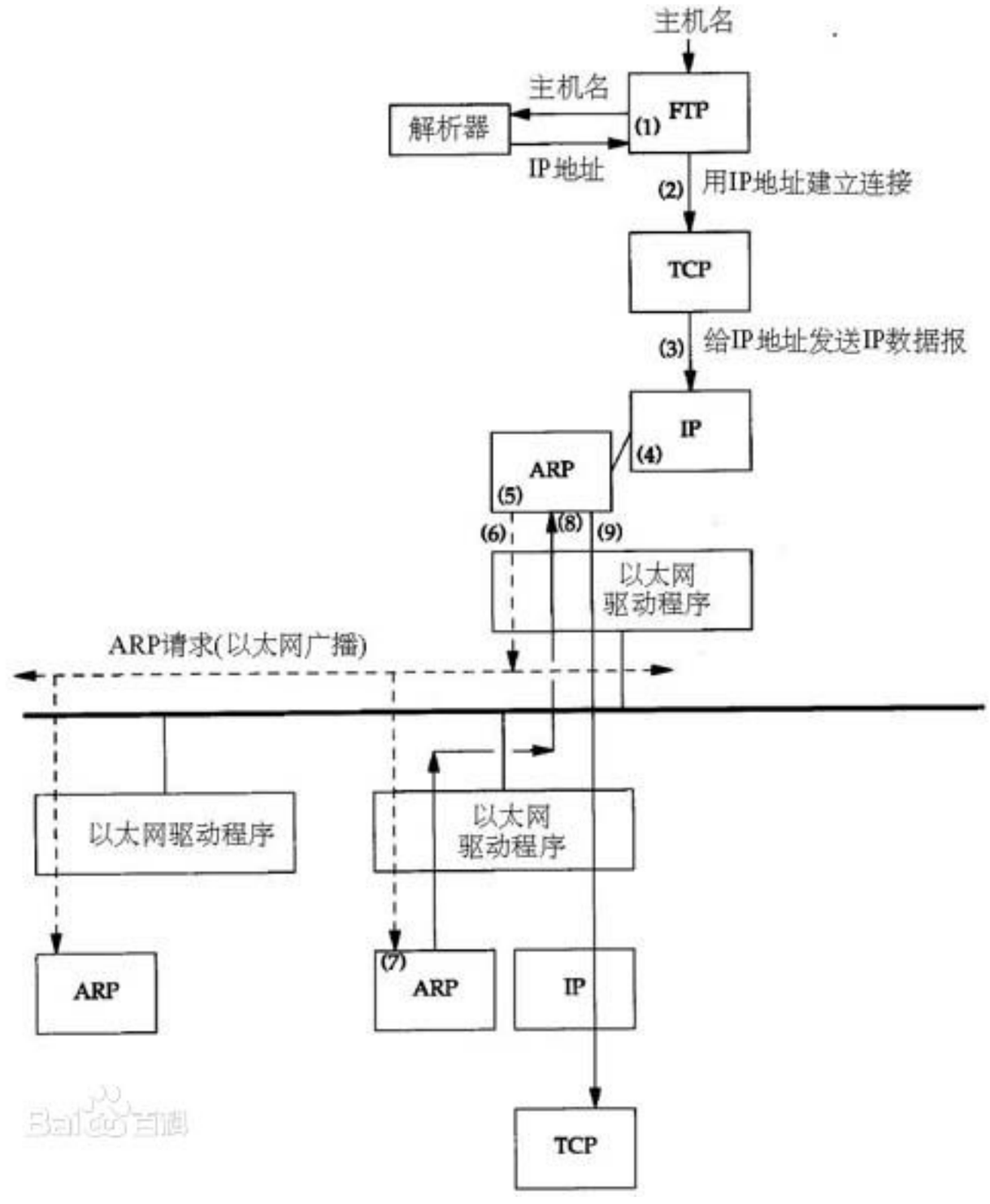
- 远线程注入DLL的流程：
  - **提升自身进程的权限为SeDebugPrivilege，（如果不注入系统进程，此步骤可省略）。**
  - **OpenProcess()获取目标进程的句柄。**
  - **VirtualAllocEx()在目标进程中分配一块内存。**
  - **WriteProcessMemory()将要注入的DLL路径写入步骤3分配的内存。**
  - **GetProcAddress()获得LoadLibraryA ()函数的地址。**
  - **CreateRemoteThread()创建远程线程，线程起始地址设为LoadLibraryA ()函数的地址，线程的参数为步骤4中DLL路径在目标进程中的地址。**
  - 等待远程线程结束后退出。

网络攻击——

# ARP攻击

# ARP协议

- **地址解析**协议，即ARP（Address Resolution Protocol），**是根据IP地址**获取物理**地址的一个**TCP/IP协议。




注：资料及图片来自于百度百科。

# ARP攻击技术

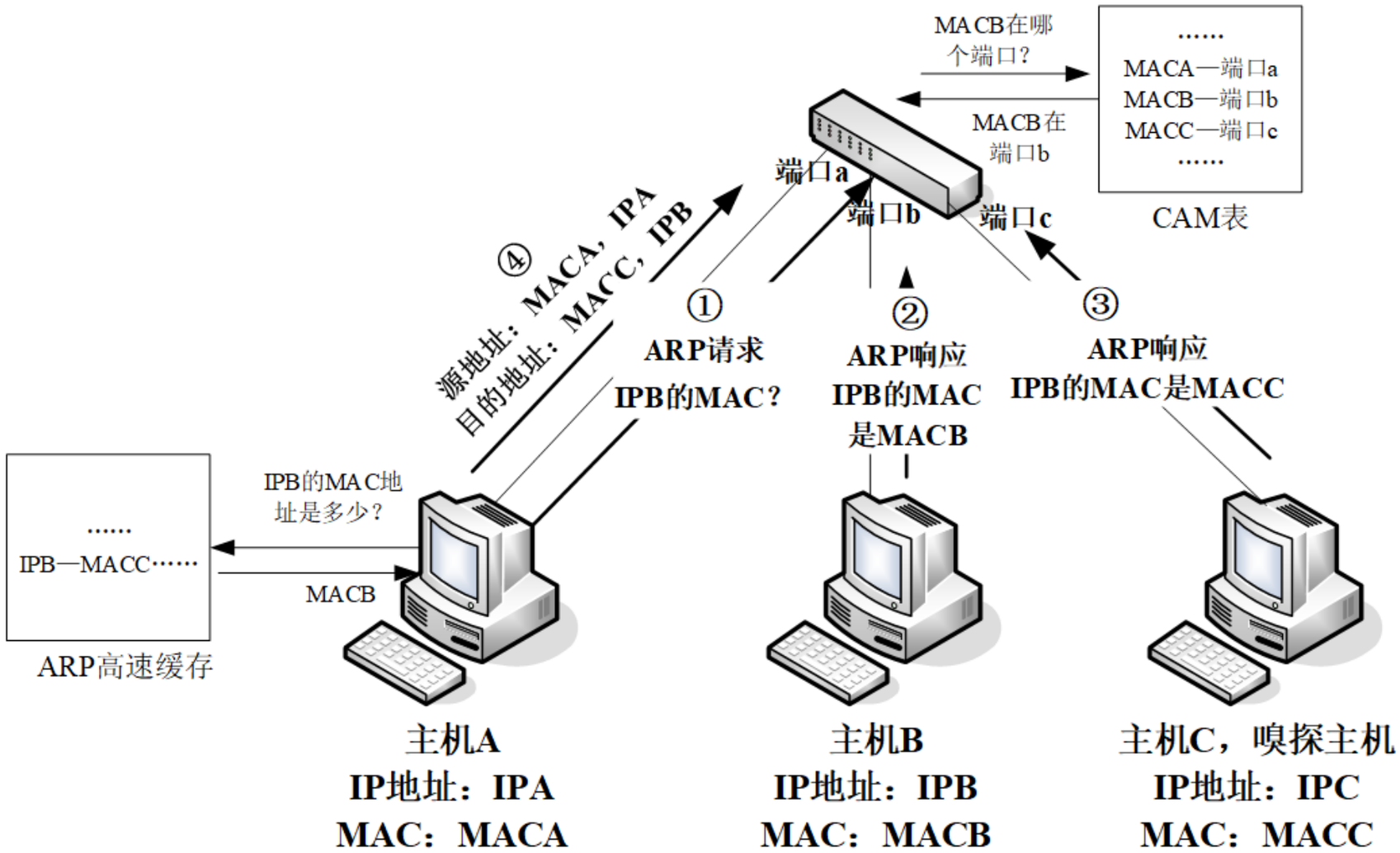
- **ARP欺骗**

  - ARP欺骗利用修改主机ARP缓存表的方法达到嗅探的目的，是一种中间人攻击。主机C为了达到嗅探的目的，会向主机A和主机B分别发送ARP应答包，告诉它们IP地址为IPB的主机MAC地址为MACC，IP地址为IPA的主机MAC地址为MACC。

  - 这样，主机A和主机B的ARP缓存中就会有IPB—MACC和IPA—MACC的记录。这样，主机A和主机B的通信数据都流向了主机C，主机C只要再发送到其真正的目的地就可以了。当然ARP缓存表项是动态更新的（一般为两分钟），如果没有更新信息，ARP映射项会自动删除。所以，主机C在监听过程中，还要不断地向主机A和主机B发送伪造的ARP应答包。

# ARP攻击技术

- ARP欺骗



网络攻击——

# 拒绝服务攻击

# 典型的拒绝服务攻击

- Ping of Death
- Teardrop
- Land
- Syn Flood
- Smurf
- HTTP Flood
- CC

# 典型DoS—Ping of Death

- A ping of death is a type of attack on a computer system that involves sending a malformed or otherwise malicious ping to a computer.。
- 早期的操作系统处理ICMP分组时，只开辟64KB缓冲区用来存放收到的数据包。
- 攻击者故意在ICMP Echo数据包(Ping包)之后附加非常多的冗余信息，使数据包的尺寸超过65535个字节的上限。
- 接收方对这种数据包进行处理时就会出现内存分配错误，导致TCP/IP堆栈溢出，从而引起系统崩溃，挂起或重启。

```
C:\Users\bupt>ping -l 66000 192.168.0.201
选项 -l 的值有错误，有效范围从 0 到 65500。
```

# 典型DoS——Teardrop

- A teardrop attack involves sending mangled IP fragments with overlapping, oversized payloads to the target machine.
- This can crash various operating systems because of a bug in their TCP/IP fragmentation re-assembly code.[54] Windows 3.1x, Windows 95 and Windows NT operating systems, as well as versions of Linux prior to versions 2.0.32 and 2.1.63 are vulnerable to this attack.
- One of the fields in an IP header is the “fragment offset” field, indicating the starting position, or offset, of the data contained in a fragmented packet relative to the data in the original packet. If the sum of the offset and size of one fragmented packet differs from that of the next fragmented packet, the packets overlap.

From Wikipedia

# 典型DoS——Teardrop

## Fragmentation and reassembly

| Fragment | Total bytes | Header bytes | Data bytes | "More fragments" flag | Fragment offset (8-byte blocks) |
|---|---|---|---|---|---|
| 1 | 2500 | 20 | 2480 | 1 | 0 |
| 2 | 2040 | 20 | 2020 | 0 | 310 |

an MTU of 1,500 bytes, For example, consider a Transport layer segment with size of 4,500 bytes.

| Fragment | Total bytes | Header bytes | Data bytes | "More fragments" flag | Fragment offset (8-byte blocks) |
|---|---|---|---|---|---|
| 1 | 1500 | 20 | 1480 | 1 | 0 |
| 2 | 1020 | 20 | 1000 | 1 | 185 |
| 3 | 1500 | 20 | 1480 | 1 | 310 |
| 4 | 560 | 20 | 540 | 0 | 495 |

We can use the last offset and last data size to calculate the total data size: 495*8 + 540 = 3960 + 540 = 4500.

https://en.wikipedia.org/wiki/IPv4#Fragmentation_and_reassembly

# 典型DoS——Teardrop

- MTU（maximum transfer unit，最大传送单位）限制传输数据的包大小，大数据包需要分段。Teardrop攻击就是利用这种分割重组间的漏洞而产生的攻击方式。

- Teardrop指的是向目标机器发送损坏的IP包，诸如重叠的包或过大的包载荷，该攻击通过TCP/IP协议栈中分片重组代码中的bug来瘫痪各种不同的操作系统。

- Teardrop攻击使接收数据方重组数据包时，出现数据包长度超大（如负值），导致溢出。(假设数据包中第二片IP包的偏移量小于第一片结束的位移，而且算上第二片IP包的Data，也未超过第一片的尾部，这就是重叠现象。)

# 典型DoS——Teardrop

第一片：Fragment offset=0；数据包长度（length）=ip.total-length-ip.headerlength=36。

| IP 首部(20 字节) | UDP 首部(8 字节) | 数据 1(28 字节) |
|---|---|---|

第二片：Fragmentoffset=32；数据包长度（length）=ip.total-length-ip.headerlength=3。

| IP 首部(20 字节) | 数据 2 |
|---|---|

若两片重组，则第二片必然嵌于第一片内，

| 1 | | 8 | 9 | 32 | 35 | 36 |
|---|---|---|---|---|---|---|
| IP 首部(20 字节) | UDP 首部(8 字节) | | 数据 1 | 数据 2 | | |

这时第一片的 end=36，offset=0；第二片的 end=32+3=35，而正常情况下第二片的段偏移 offset=36，这样造成 fp->len=end-offset=35-36=-1。此时调用 memcpy ((ptr + fp->offset), fp->ptr, fp->len)时，由于 fp->len 为负数，会引起堆栈溢出。

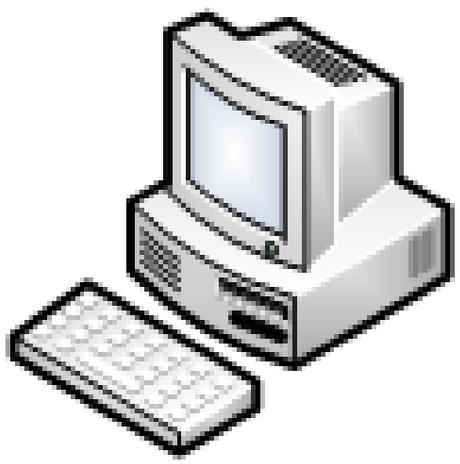
来自：DoS攻击原理与对策的进一步研究 陈波 于泠

# 典型DoS——Land攻击

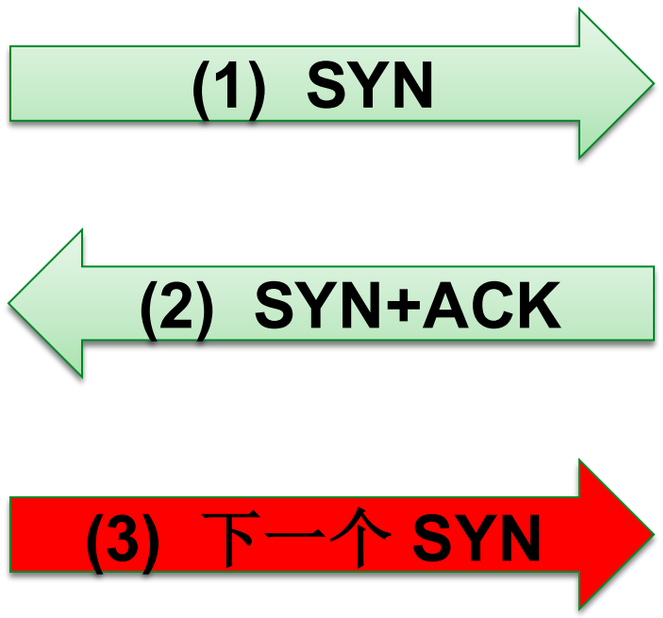
- Land攻击：利用特殊的TCP封包传送至目标主机，使其因无法判别而当机或被迫重新启动。
- 攻击原理是：用一个特别打造的SYN包，它的源地址和目标地址都被设置成某一个服务器地址。此举将导致接受服务器向它自己的地址发送SYN－ACK消息，结果这个地址又发回ACK消息并创建一个空连接。被攻击的服务器每接收一个这样的连接都将保留，直到超时。

# 典型DoS—SYN洪水

- **SYN洪水**
  - **原理：**
    - **每个机器都需要为半开连接分配一定的资源**
    - **这种半开连接的数量是有限制**
    - **共计方利用TCP连接三次握手过程，打开大量的半开TCP连接**
    - **目标机器不能进一步接受TCP连接。机器就不再接受进来的连接请求。**



攻击机



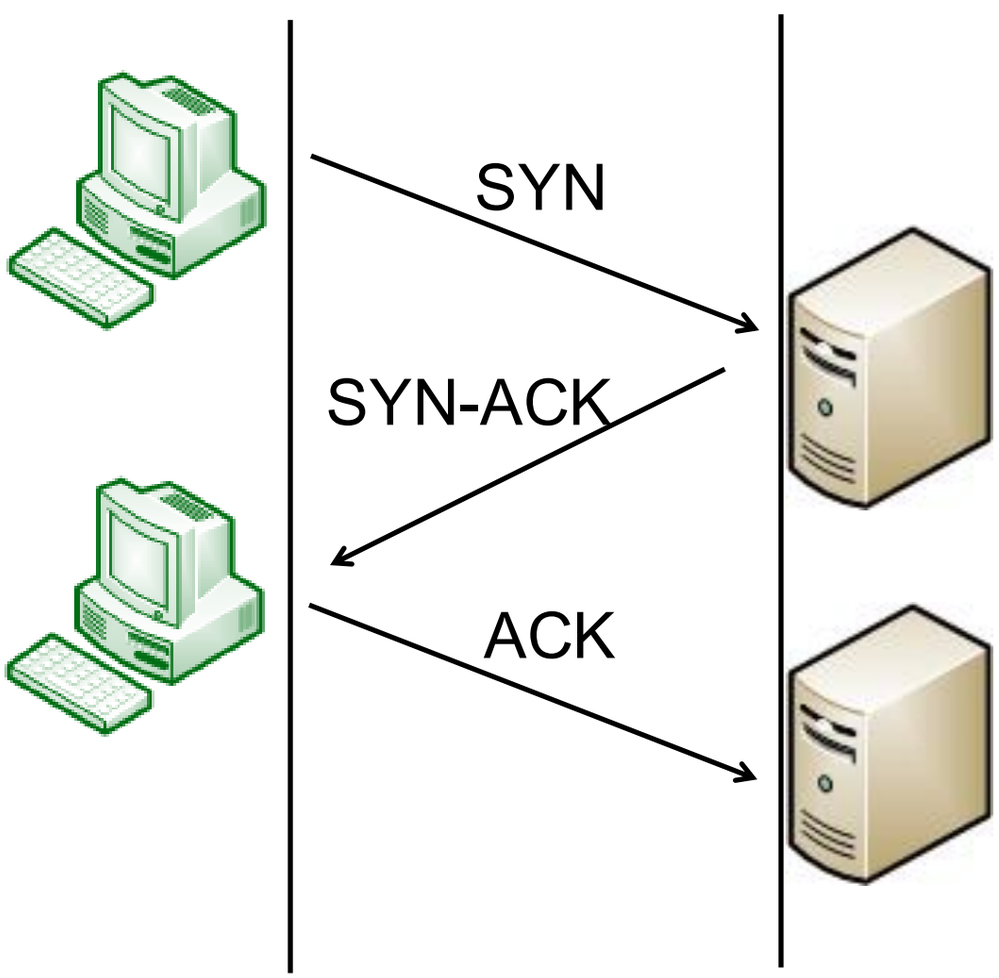

服务器

# 典型DoS—SYN洪水

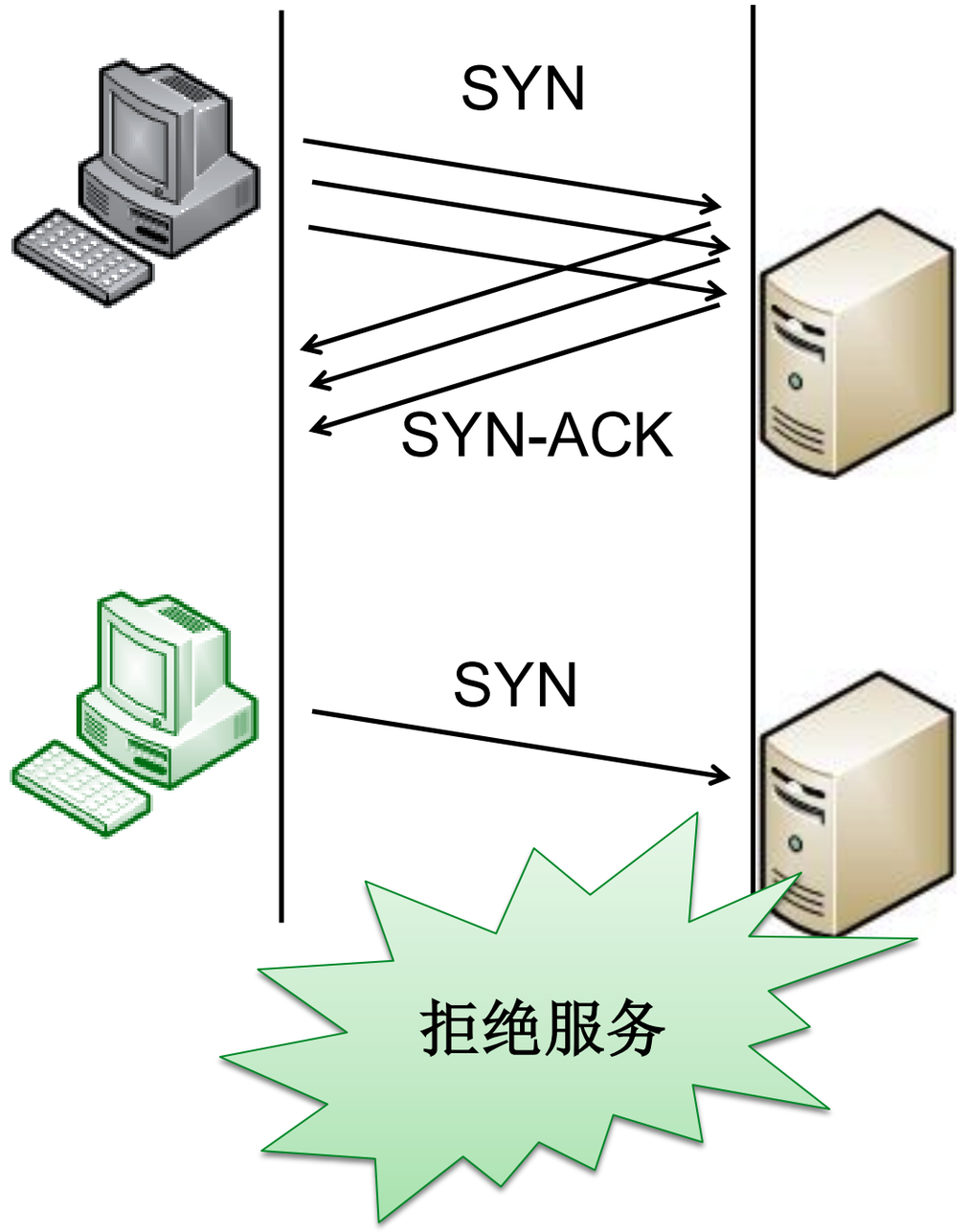
- SYN洪水
  - 受影响的系统：大多数操作系统
  - 攻击细节
    - 连接请求是正常的，但是，源IP地址往往是伪造的，并且是一台不可达的机器的IP地址，否则，被伪造地址的机器会重置这些半开连接
    - 一般，半开连接超时之后，会自动被清除，所以，攻击者的系统发出SYN包的速度要比目标机器清除半开连接的速度要快
    - 任何连接到Internet上并提供基于TCP的网络服务，都有可能成为攻击的目标
    - 这样的攻击很难跟踪，因为源地址往往不可信，而且不在线

# 典型DoS—SYN洪水

正常的SYN连接：




SYN/ACK Flood Attack：

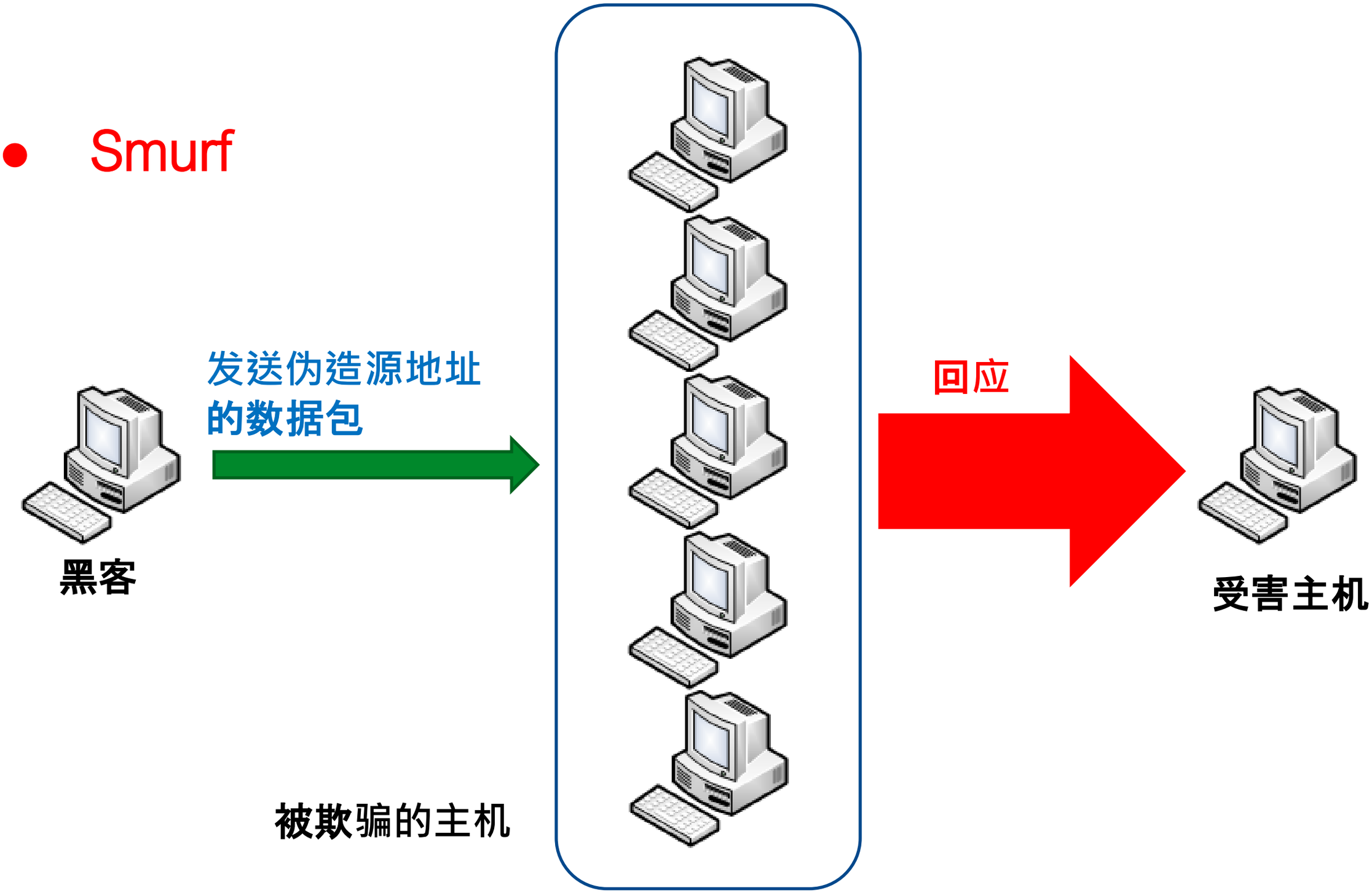


# 典型DoS—Smurf

- Smurf
  - **原理：**
    - **攻击者向一个广播地址发送ICMP Echo请求，并且用受害者的IP地址作为源地址**
    - **广播地址网络上的每台机器响应这些Echo请求，同时向受害者主机发送ICMP Echo-Reply应答**
    - **受害者主机会被这些大量的应答包淹没**
  - **受影响的系统：大多数操作系统和路由器**

# 典型DoS—Smurf

- Smurf



# 典型DoS—HTTP洪水

- HTTP洪水
- 这类攻击会占用大量的HTTP进程，从而耗费大量的系统资源。最终，会导致系统因不堪重负而崩溃掉。
- 以最典型的HTTP GET FLOOD攻击为例。
  - HTTP GET FLOOD是针对应用服务器上的某个文件，对其进行快速的反复的重复读取操作，从而造成服务器的资源减少直至崩溃。
  - HTTP GET FLOOD针对的不仅仅是WEB服务器，还有数据库服务器。大量的HTTP请求产生了大量的数据库查询，可以在几秒之内使数据库停止响应，系统负载升高，最终导致服务器宕机。

网络安全——

# 网络攻击—web安全

# XSS

XSS（Cross Site Scripting）攻击，全称跨站脚本攻击。

- 跨站脚本攻击是指通过存在安全漏洞的Web网站注册用户的浏览器内运行非法的HTML标签或JavaScript进行的一种攻击。
- 跨站脚本攻击有可能造成以下影响:
  - 利用虚假输入表单骗取用户个人信息。
  - 利用脚本窃取用户的Cookie值，被害者在不知情的情况下，帮助攻击者发送恶意请求。
  - 显示伪造的文章或图片。
- XSS的原理是恶意攻击者往 Web 页面里插入恶意可执行网页脚本代码，当用户浏览该页之时，嵌入其中 Web 里面的脚本代码会被执行，从而可以达到攻击者盗取用户信息或其他侵犯用户安全隐私的目的。

*参考文献 https://www.cnblogs.com/fundebug/p/details-about-6-web-security.html*
*https://zoumiaojiang.com/article/common-web-security/*

# XSS

## 非持久型 XSS（反射型 XSS ）

- 反射型XSS，也叫非持久型XSS，是指发生请求时，XSS代码出现在请求URL中


- **具体流程：**

1、Alice给Bob发送一个恶意构造了Web的URL。

2、Bob点击并查看了这个URL。

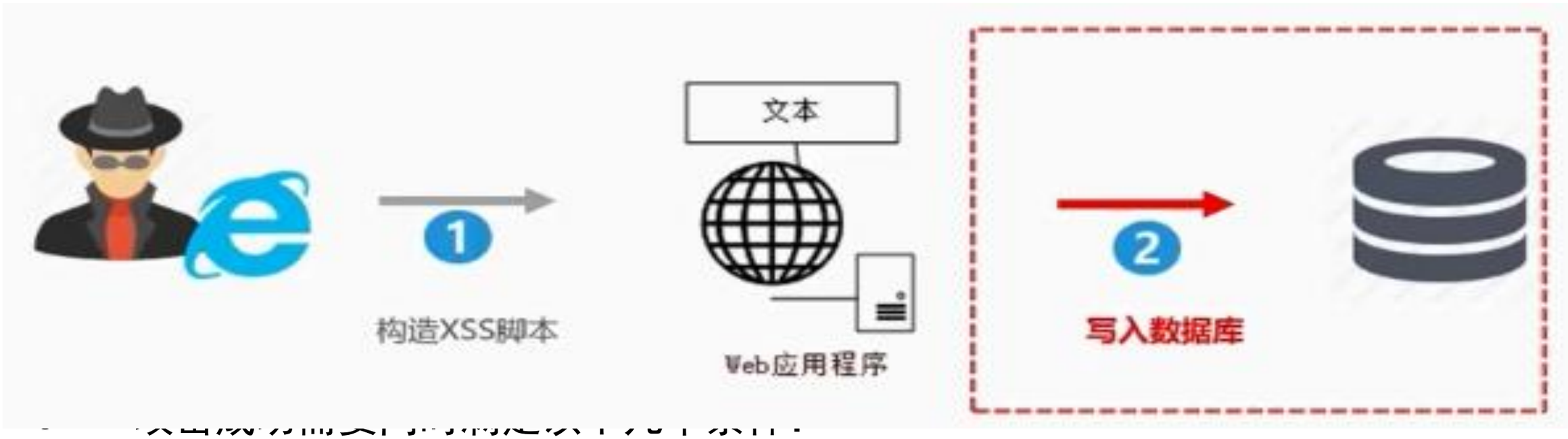
3、恶意页面中的JavaScript打开一个具有漏洞的HTML页面并将其安装在Bob电脑上。

4、具有漏洞的HTML页面包含了在Bob电脑本地域执行的JavaScript。

5、Alice的恶意脚本可以在Bob的电脑上执行Bob所持有的权限下的命令。

# XSS

存储型XSS，也叫持久型XSS




- 攻击成功需要同时满足以下几个条件：
  - POST 请求提交表单后端没做转义直接入库；后端从数据库中取出数据没做转义直接输出给前端；前端拿到后端数据没做转义直接渲染成 DOM（文档对象模型（Document Object Model））。
- 持久型 XSS 有以下几个特点：
  - 持久性，植入在数据库中
  - 危害面广，甚至可以让用户机器变成 DDoS 攻击的肉鸡。
  - 盗取用户敏感私密信息。

# XSS

## 基于字符集的 XSS

- 浏览器以及各种开源的库都专门针对了 XSS 进行转义处理，尽量默认抵御绝大多数 XSS 攻击，但是还是有很多方式可以绕过转义规则，让人防不胜防。比如「基于字符集的 XSS 攻击」就是绕过这些转义处理的一种攻击方式，比如有些 Web 页面字符集不固定，用户输入非期望字符集的字符，有时会绕过转义过滤规则。

## 未经验证的跳转 XSS

- 有一些场景是后端需要对一个传进来的待跳转的 URL 参数进行一个 302 跳转，可能其中会带有一些用户的敏感（cookie）信息。如果服务器端做302跳转，跳转的地址来自用户的输入，攻击者可以输入一个恶意的跳转地址来执行脚本。

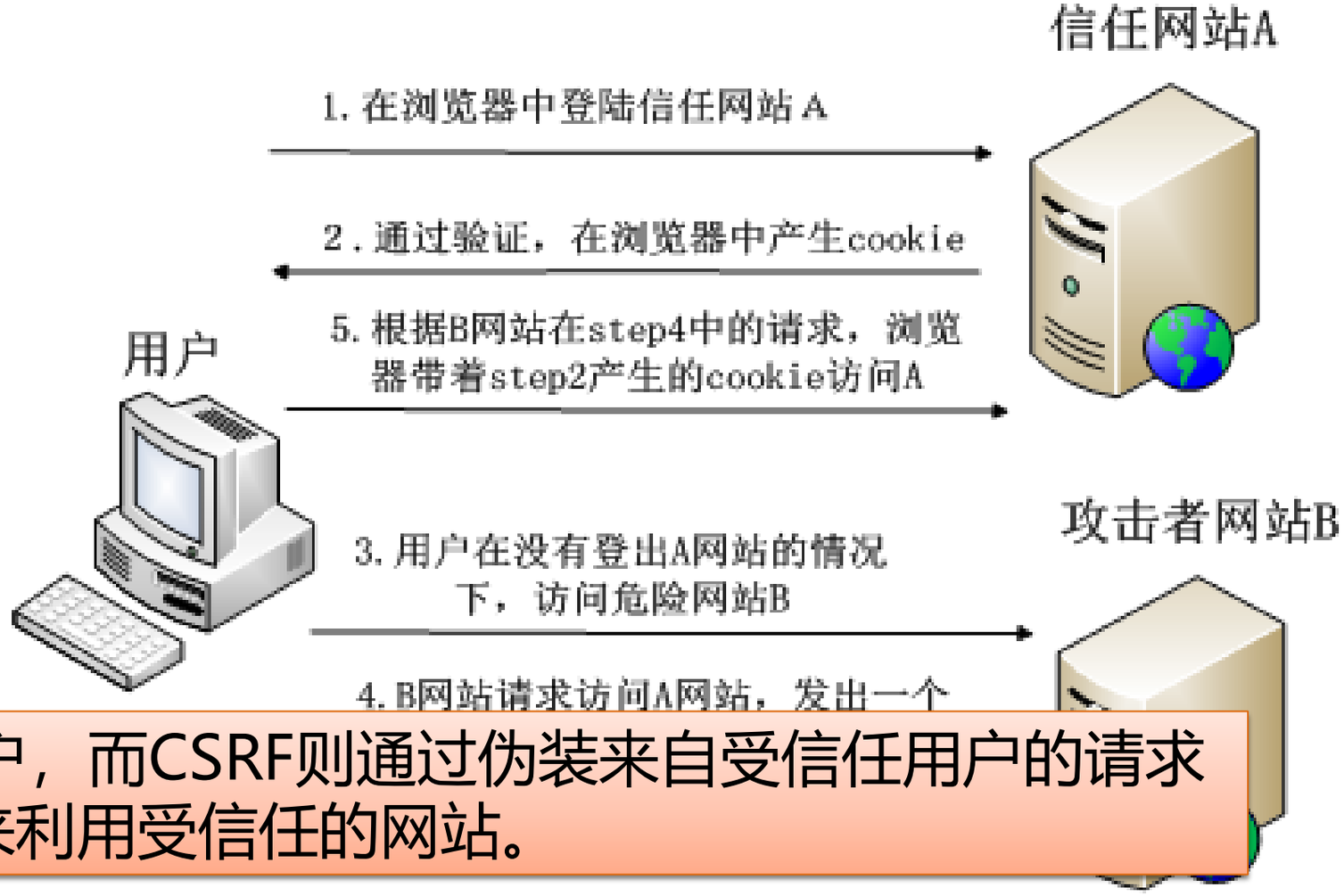
# CSRF

CSRF（Cross-Site Request Forgery），跨站请求伪造攻击

- CSRF是一种常见的Web攻击，它利用用户已登录的身份，在用户毫不知情的情况下，以用户的名义完成非法操作。

完成 CSRF 攻击必须要有三个条件：

- 用户已经登录了站点 A，并在本地记录了 cookie
- 在用户没有登出站点 A 的情况下（也就是 cookie 生效的情况下），访问了恶意攻击者提供的引诱危险站点 B (B 站点要求访问站点A)
-




XSS利用站点内的信任用户，而CSRF则通过伪装来自受信任用户的请求来利用受信任的网站。

# CSRF

## CSRF防御

一般的 CSRF 防御也都在服务端进行，主要从以下两个方面入手：

正确使用 GET，POST 请求和 cookie

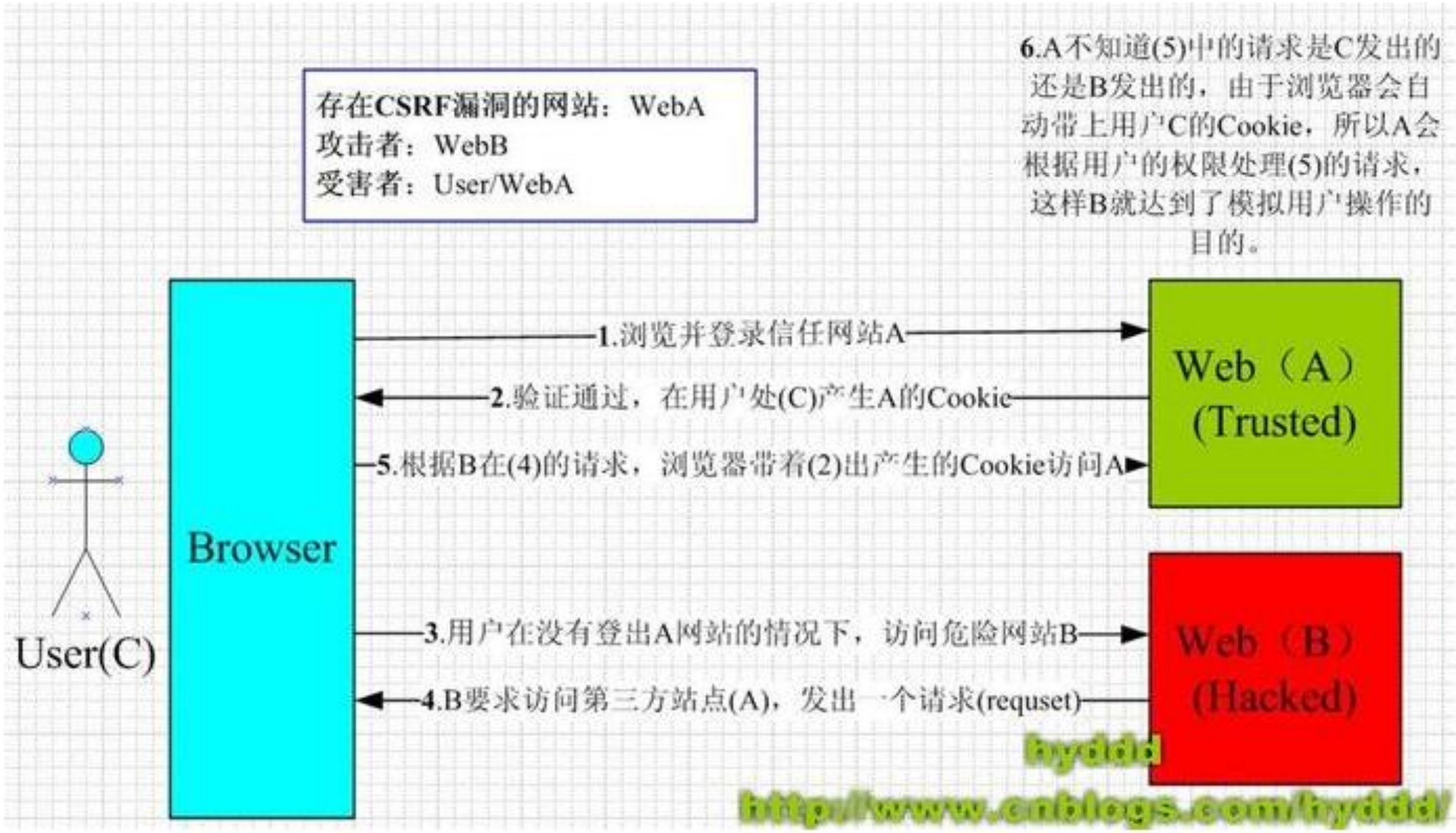
- GET 请求常用在查看，列举，展示等不需要改变资源属性的时候（数据库 query 查询的时候）
- POST 请求常用在 From 表单提交，改变一个资源的属性或者做其他一些事情的时候（数据库有 insert、update、delete 的时候）

在非 GET 请求中增加 token

- **为每个用户生成一个唯一的 cookie token**，所有表单都包含同一个伪随机值，但是由于用户的 cookie 很容易由于网站的 XSS 漏洞而被盗取，所以这个方案必须要在没有 XSS 的情况下才安全。
- **每个 POST 请求使用验证码**，这个方案算是比较完美的，但是需要用户多次输入验证码，用户体验比较差，所以不适合在业务中大量运用。
- **渲染表单的时候，为每一个表单包含一个 csrfToken**，提交表单的时候，带上 csrfToken，然后在后端做 csrfToken 验证。

# CSRF

- CSRF（Cross Site Request Forgery, 跨站域请求伪造）




XSS利用站点内的信任用户，而CSRF则通过伪装来自受信任用户的请求来利用受信任的网站。

# SQL与SQL注入的基本概念

- **SQL注入**

所谓SQL注入，就是通过**把SQL命令插入到Web表单递交或输入域名或页面请求的查询字符串**，最终达到欺骗服务器执行恶意的SQL命令，比如先前的很多影视网站泄露VIP会员密码大多就是通过WEB表单递交查询字符暴出的，这类表单特别容易受到SQL注入式攻击。

# SQL注入的由来

- SQL第一次为公众所知

1998年，著名的黑客杂志《Phrack》第54期，一位名字为rfp （rain forest puppy）的黑客发表了一篇题为“NT Web Technology Vulnerabilities”的文章。

第一次向公众介绍了SQL注入这种新型的攻击技术。

http://www.phrack.org

# ASP/SQL注入

- 一个最简单的登陆查询:

```
SELECT * FROM users
        WHERE username= ‘sqlin‘ AND password = ‘aaa'
```

- ASP/MS SQL中查询的语法会成为如下：

```
var sql = "SELECT * FROM users
        WHERE username= '" + formusername +
        "' AND password = '" + formpassword + "'";
```

# ASP/SQL注入

- 通过SQL语句从以下的users表中可以查询用户表单提供的用户名和密码是否匹配正确，以判断登陆是否成功。

| userID | username | password |
|---|---|---|
| 1 | admin | helloworld |
| 2 | guest | iamguest |

# ASP/SQL注入

- 当我们构造如下的用户名密码：

```
formusername = ‘ or 1=1 – –

formpassword = anything
```

- 提交表单之后，SQL语句则会变成如下：

```
SELECT * FROM users

        WHERE username = ' ' or 1=1

        – – AND password = ‘anything‘
```

# PHP/MySQL

- 上面的那个例子是针对MS SQL构造的SQL注入语句。
- 在PHP/MySQL则会变成如下语句:

```
$sql = "SELECT * FROM users
        WHERE userID= $formuserID
        AND password = $formpassword";
```

# PHP/MySQL的例子

- 构造的注入语句：

```
$formuserID = 1 or 1=1 #

$formpassword = anything
```

- 提交表单之后，SQL语句则会变成如下：

```
SELECT * FROM users

        WHERE  userID = 1 or 1=1

        #AND password = 'anything'
```

# SQL注入的基本流程

寻找注入点

信息采集

权限判断

攻击系统

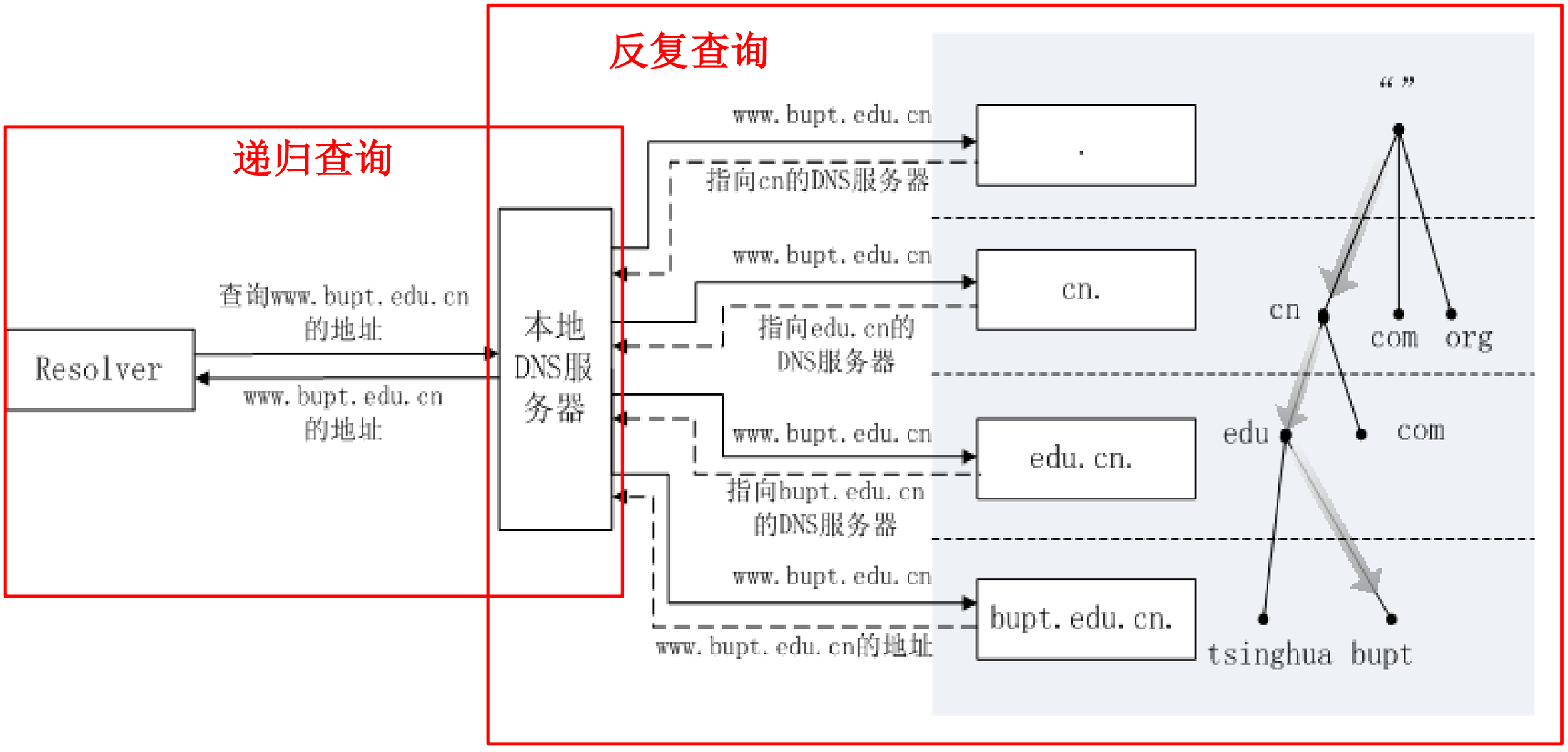
网络攻击——

# DNS安全

# DNS工作流程

- 查询方式
  - **递归查询：一般客**户机和服务器之间是递归查询，DNS**服**务器如**果未能在本地找到相**应的信息，就代替用户向其它服务器进行查询，这时它是代替用户扮演了解析器（resolver）**的角色，直到最后把**结果找到，也可能根本没有结果，那就返回错误，并返回给**用**户为止。
  - **反复查询（迭代查询）：一般服**务器之间属于反复查询。DNS**服**务器返回的要么是本地存在的结果信息，要么是一个错误码，告诉查询者你要的信息这里没有，然后再返回一个可能会有查询结**果的**DNS**服**务器地址，让查询者到那里去查一查。

# DNS工作流程


递归查询
反复查询
Resolver
查询www.bupt.edu.cn
的地址
www.bupt.edu.cn
的地址
本地
DNS服
务器
www.bupt.edu.cn
指向cn的DNS服务器
www.bupt.edu.cn
cn.
指向edu.cn的
DNS服务器
www.bupt.edu.cn
edu.cn.
指向bupt.edu.cn
的DNS服务器
www.bupt.edu.cn
bupt.edu.cn.
www.bupt.edu.cn的地址
“ ”
cn
com
org
edu
com
tsinghua
bupt

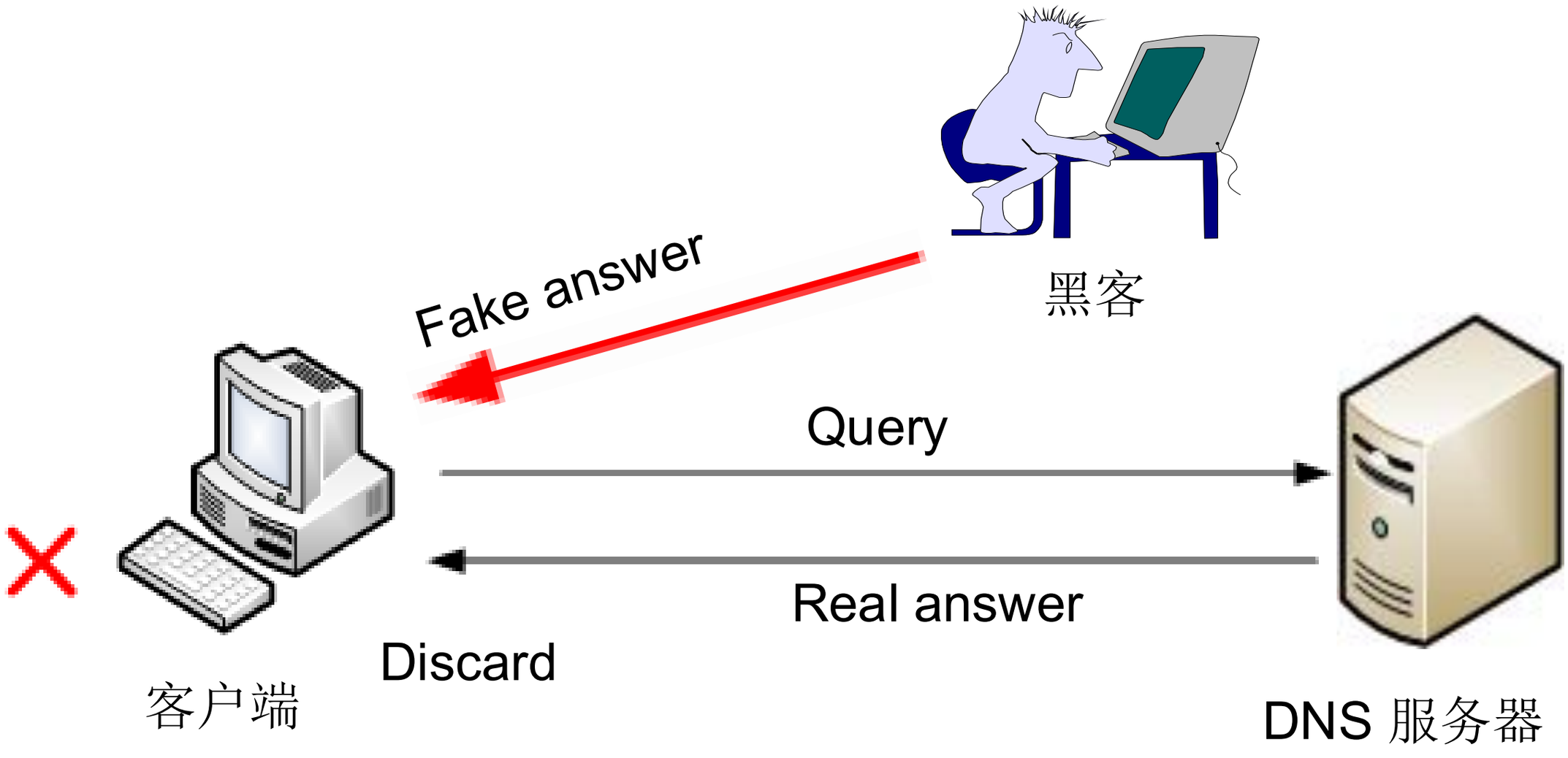
# DNS安全威胁

- DNS应答包被客户端接受需要满足以下五个条件
  - 1、应答包question**域和**请求包question**域的域名信息一致**。
  - 2、应答包的Transaction ID**和**请求包中的Transaction ID**一致**。
  - 3、应答包的源IP**地址与**请求包的目的IP**地址一致**。
  - 4、应答包的目的IP**地址和端口与**请求包的源IP**地址和端口一致**。
  - 5、**第一个到达的符合以上四个条件的**应答包。

- **从以上五个条件可以看出，最初**设计DNS时没有考虑它的安全问题，这导致DNS协议存在很多漏洞，这使得DNS**很容易受到攻**击。

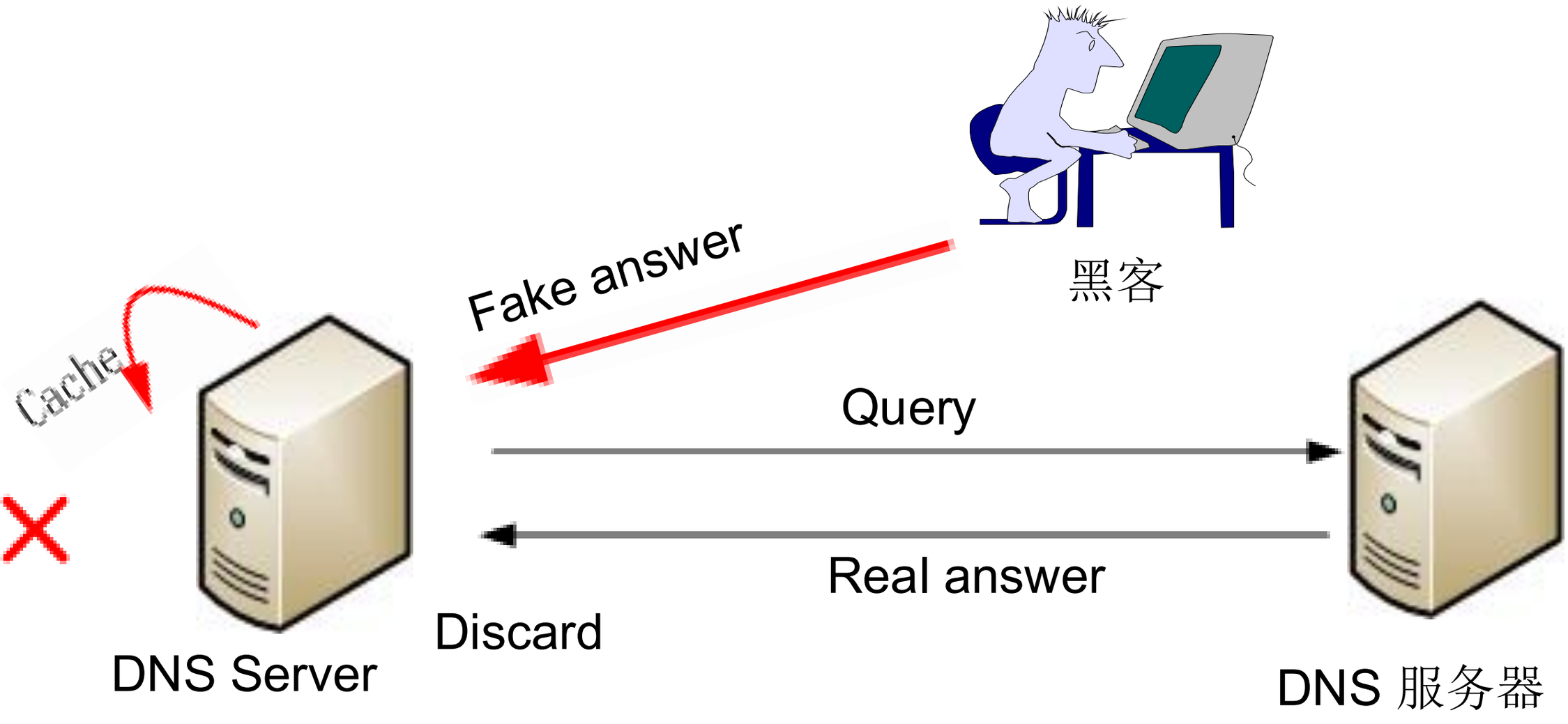
# DNS安全威胁

- DNS欺骗攻击流程



# DNS安全威胁

- DNS下毒攻击流程



网络安全——

# 网络安全扫描

# 端口扫描技术

- 端口扫描技术是一项自动探测本地和远程系统端口开放情况的策略及方法，它使系统用户了解系统目前向外界提供了哪些服务，从而为系统用户管理网络提供了一种手段。
- 原理：向目标主机的TCP/IP服务端口发送探测数据包，并记录目标主机的响应，通过分析响应来判断服务端口是打开还是关闭，即可得知端口提供的服务或信息。也可以通过捕获本地主机或服务器流入流出IP数据包来监视本地主机的运行情况，通过对接收到的数据进行分析，帮助我们发现目标主机的某些内在的弱点。
- 分类：全连接扫描、半连接扫描、秘密扫描

# 端口扫描技术–全连接扫描

- 全连接扫描技术是TCP端口扫描的基础，包括：
  - TCP connect()扫描

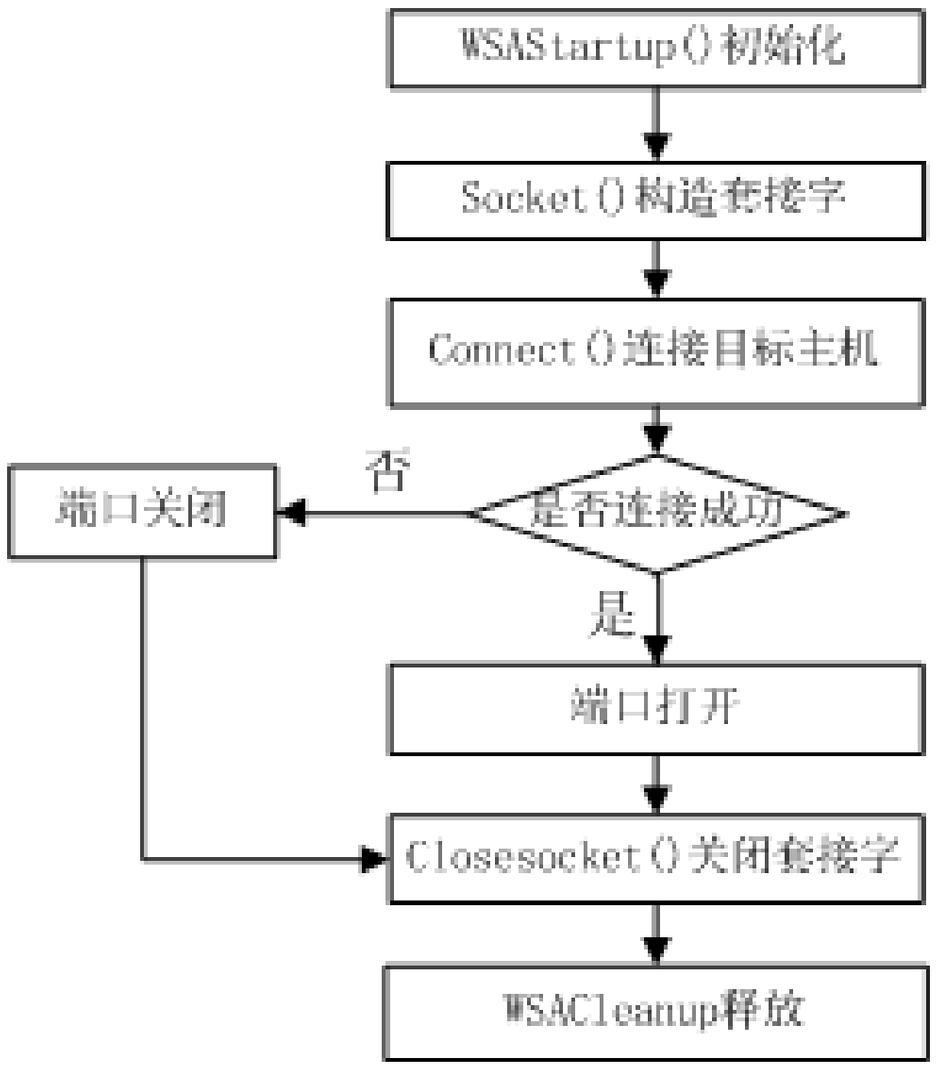
利用操作系统提供的connect()系统调用，与每一个感兴趣的目标计算机的端口进行连接。如果端口处于侦听状态，那么connect()就能成功；否则，该端口是不能用的，即没有提供服务。

  - TCP反向ident扫描

ident协议允许（RFC1413）看到通过TCP连接的任何进程的拥有者的用户名，即使这个连接不是由这个进程开始的。

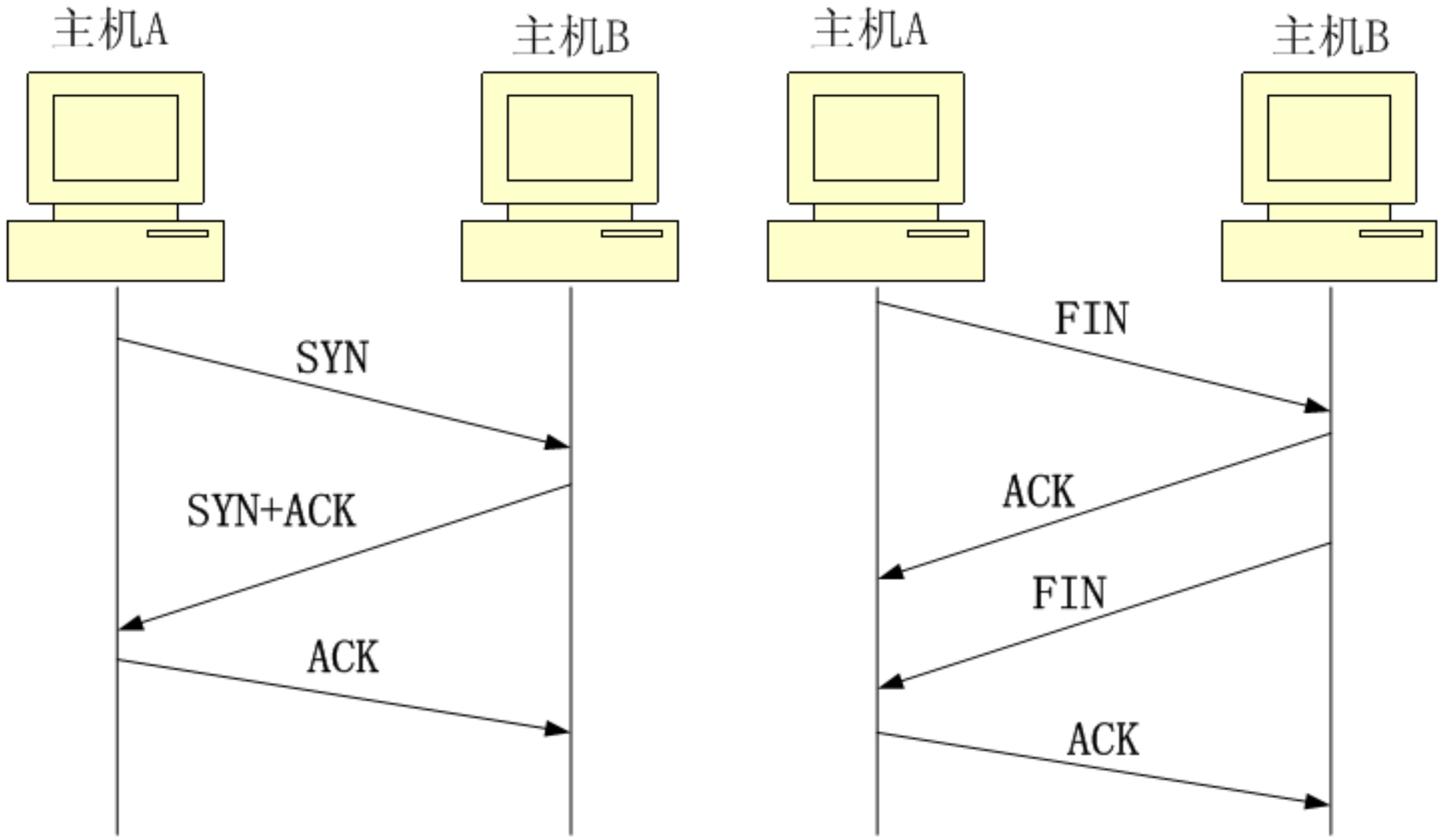
# TCP connect()扫描

- TCP连接扫描利用了TCP协议的正常连接过程，其流程为：



# TCP connect()扫描

- 正常的TCP连接与终止连接过程为：



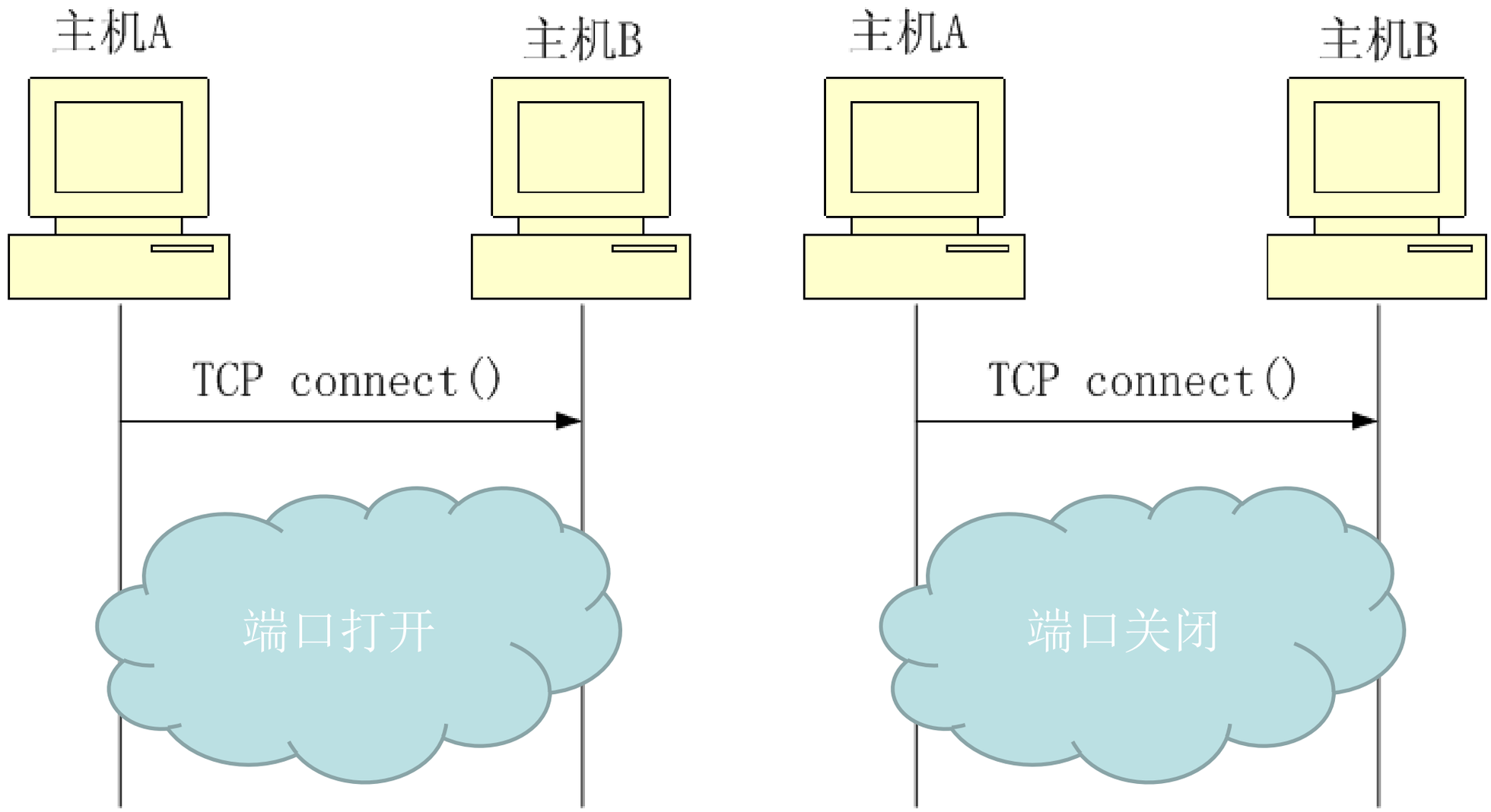

TCP正常连接过程




TCP正常终止连接过程

# TCP connect()扫描

- 基于TCP连接的扫描过程为：

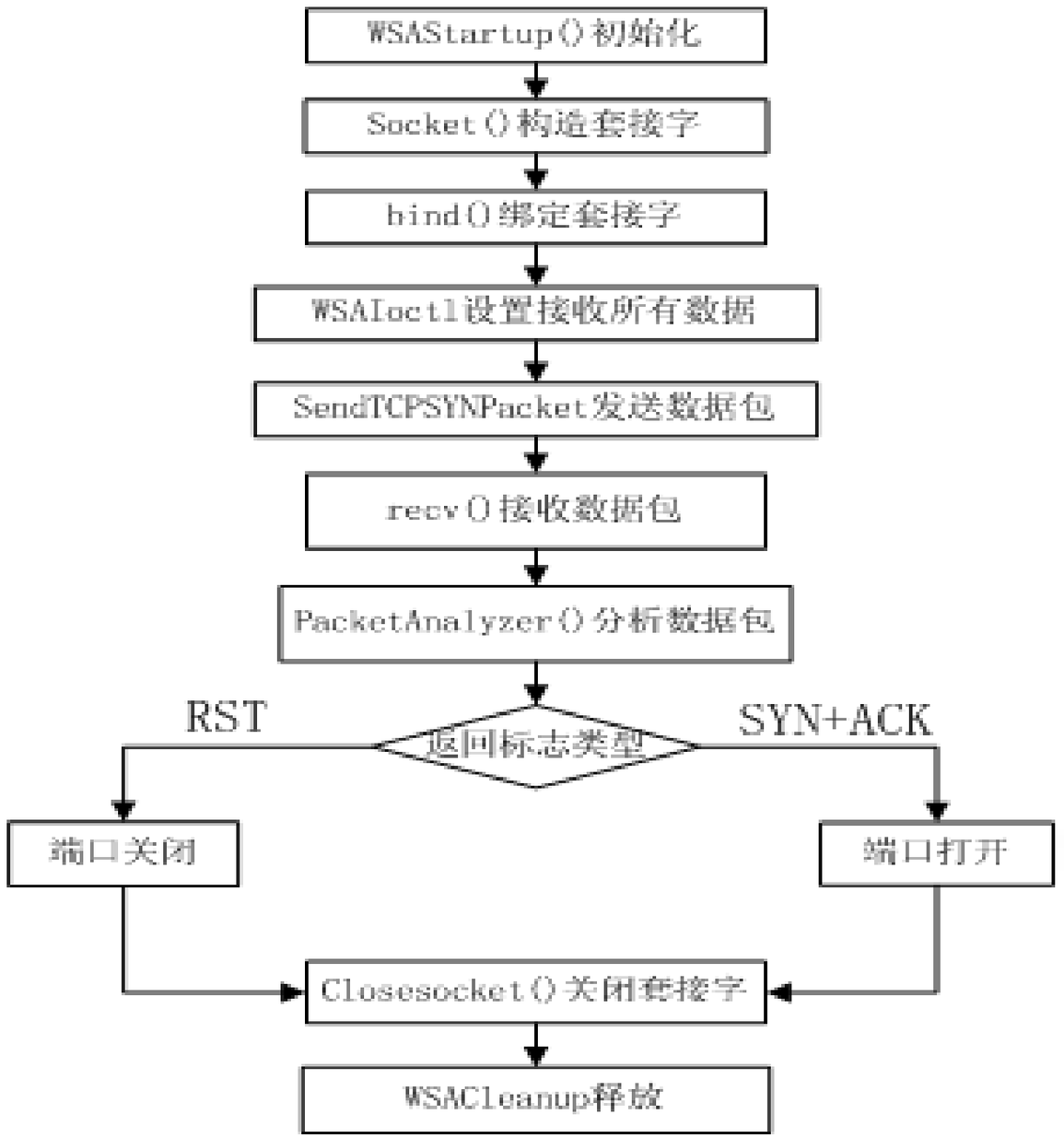


# TCP connect()扫描

- 优点
  - 扫描迅速
  - 准确而且不需要任何权限，系统中的任何用户可以使用这个调用
  - 可以同时打开多个套接字，加速扫描，使用非阻塞I/O还允许设置一个低的时间用尽周期，同时观察多个套接字
- 缺点
  - 扫描方式不隐蔽，服务器日志会记录下大量密集的连接和错误记录
  - 易被目标主机防火墙发觉而被过滤掉

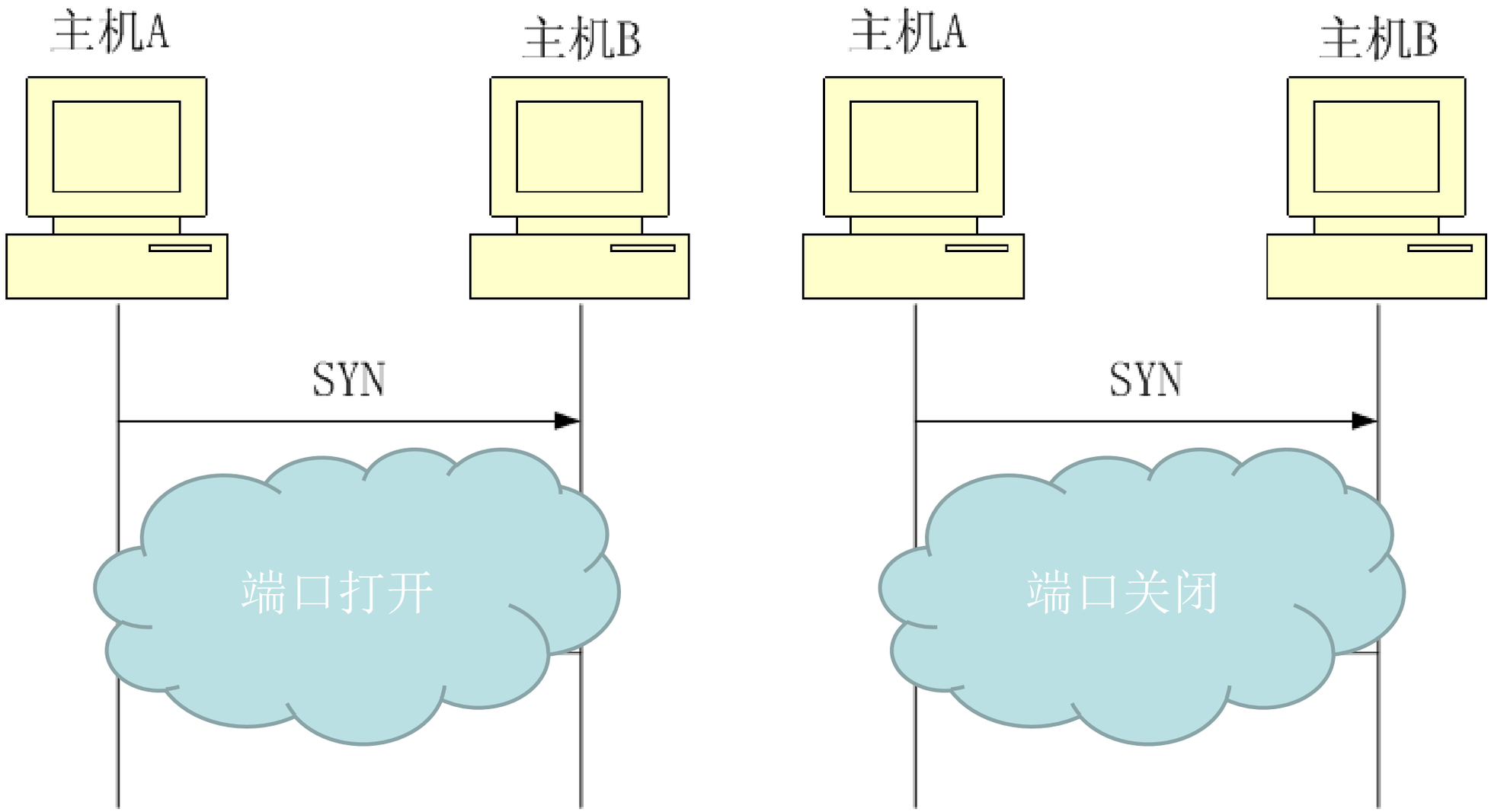
# TCP SYN扫描

- TCP SYN扫描应用非常广泛，其流程为：



# TCP SYN扫描

- 基于TCP SYN的扫描过程为：



# TCP SYN扫描

- 优点
  - 扫描迅速快，效率高
  - 一般不会在目标计算机上留下记录，比较隐蔽
- 缺点
  - 在大部分操作系统下，扫描主机需要构造适合于这种扫描的包，而通常情况下，必须要有root权限才能建立自己的SYN数据包

网络安全——

# 防火墙

# 什么是访问控制？

- 访问控制包括三个要素：主体、客体和控制策略。

  - （1）**主体S（Subject）。是指提出**访问资源具体请求。是某一操**作**动作的发起者，但不一定是动作的执行者，可能是某一用户，**也可以是用**户启动的进程、服务和设备等。

  - （2）**客体O（Object）。是指被**访问资源的实体。所有可以被操**作的信息、**资源、对象都可以是客体。客体可以是信息、文件、记录等集合体，也可以是网络上硬件设施、无限通信中的终端，**甚至可以包含另外一个客体。**

  - （3）**控制策略A（Attribution）。是主体**对客体的相关访问规则**集合，即属性集合。**访问策略体现了一种授权行为，也是客体对**主体某些操作行**为的默认。

# 访问控制策略

- **安全策略的**实施原则：安全策略的制定实施也是围绕主体、客体**和安全控制**规则集三者之间的关系展开的。

  （1） **最小特权原则**：**最小特**权原则是指主体执行操作时，按照主**体所需**权利的最小化原则分配给主体权力。最小特权原则的优点**是最大限度地限制了主体**实施授权行为，可以避免来自突发事件、错误和未授权用主体的危险。也就是说，为了达到一定目的，主**体必**须执行一定操作，但他只能做他所被允许做的，其它除外。

  （2） **最小泄漏原则**：**最小泄漏原**则是指主体执行任务时，按照**主体所需要知道的信息最小化的原**则分配给主体权力。

  （3） **多级安全策略**：**多**级安全策略是指主体和客体间的数据流向**和**权限控制按照安全级别的绝密（TS）、**秘密**（S）、**机密**（C）、**限制**（RS）**和无**级别（U）**五**级来划分。多级安全策略的优点是**避免敏感信息的**扩散。具有安全级别的信息资源，只有安全级别**比他高的主体才能**够访问。

# 自主访问控制模型DAC

- 自主访问控制模型（DAC Model，Discretionary Access Control Model）是根据自主访问控制策略建立的一种模型，允许合法用户以用户或用户组的身份访问策略规定的客体，同时阻止非授权用户访问客体，某些用户还可以自主地把自己所拥有的客体的访问权限授予其它用户。

- 自主访问控制又称为任意访问控制。LINUX，UNIX、WindowsNT或是SERVER版本的操作系统都提供自主访问控制的功能。

- 在实现上，首先要对用户的身份进行鉴别，然后就可以按照访问控制列表所赋予用户的权限允许和限制用户使用客体的资源。主体控制权限的修改通常由特权用户或是特权用户（管理员）组实现。

# 强制访问控制模型（MAC）

- **强制访问控制模型**（MAC Model：Mandatory Access Control Model）**一种多级访问控制策略**。

- **系**统对访问主体和受控对象实行强制访问控制，系统事先给访问主体和受控对象分配不同的安全级别属性，实施访问控制时，系统先对访问主体和受控对象的安全级别属性进行比较，再决定访问主体能否访问该受控对象。

- MAC对访问主体和受控对象标识两个安全标记：一个是具**有偏序关系的安全等**级标记；另一个是非等级分类标记。**当主体s的安全**类别为TS，**而客体o的安全**类别为S时，用**偏序关系可以表述**为SC(s)≥SC(o)。

# 强制访问控制模型（MAC）

- 考虑到偏序关系，主体对客体的访问主要有四种方式：

（1）**向下读**（rd，read down）：**主体安全**级别高于客体**信息**资源的安全级别时允许查阅的读操作；

（2）**向上读**（ru，read up）：**主体安全**级别低于客体信息资源的安全级别时允许的读操作；

（3）**向下写**（wd，write down）：**主体安全**级别高于客体**信息**资源的安全级别时允许执行的动作或是写操作；

（4）**向上写**（wu，write up）：**主体安全**级别低于客体信息资源的安全级别时允许执行的动作或是写操作。

# 强制访问控制模型（MAC）

- Bell-LaPadula模型

  BLP[Bell and LaPadula，1976]模型是典型的信息保密性多级安全模型，主要应用于军事系统。

- **无上读、无下写**

- Bell-LaPadula模型通常是处理多级安全信息系统的设计基础，客体在处理绝密级数据和秘密级数据时，要防止处理绝密级数据的程序把信息泄露给处理秘密级数据的程序。

- BLP模型的出发点是维护系统的保密性，有效地防止信息泄露，忽略了完整性指标，使非法、越权篡改成为可能。

# 强制访问控制模型（MAC）

- **Biba模型**

  Biba模型[Biba,1977]在研究BLP模型的特性时发现，BLP模型只解决了信息的保密问题，其在完整性定义存在方面有一定缺陷。

- **禁止向上写，没有向下读**

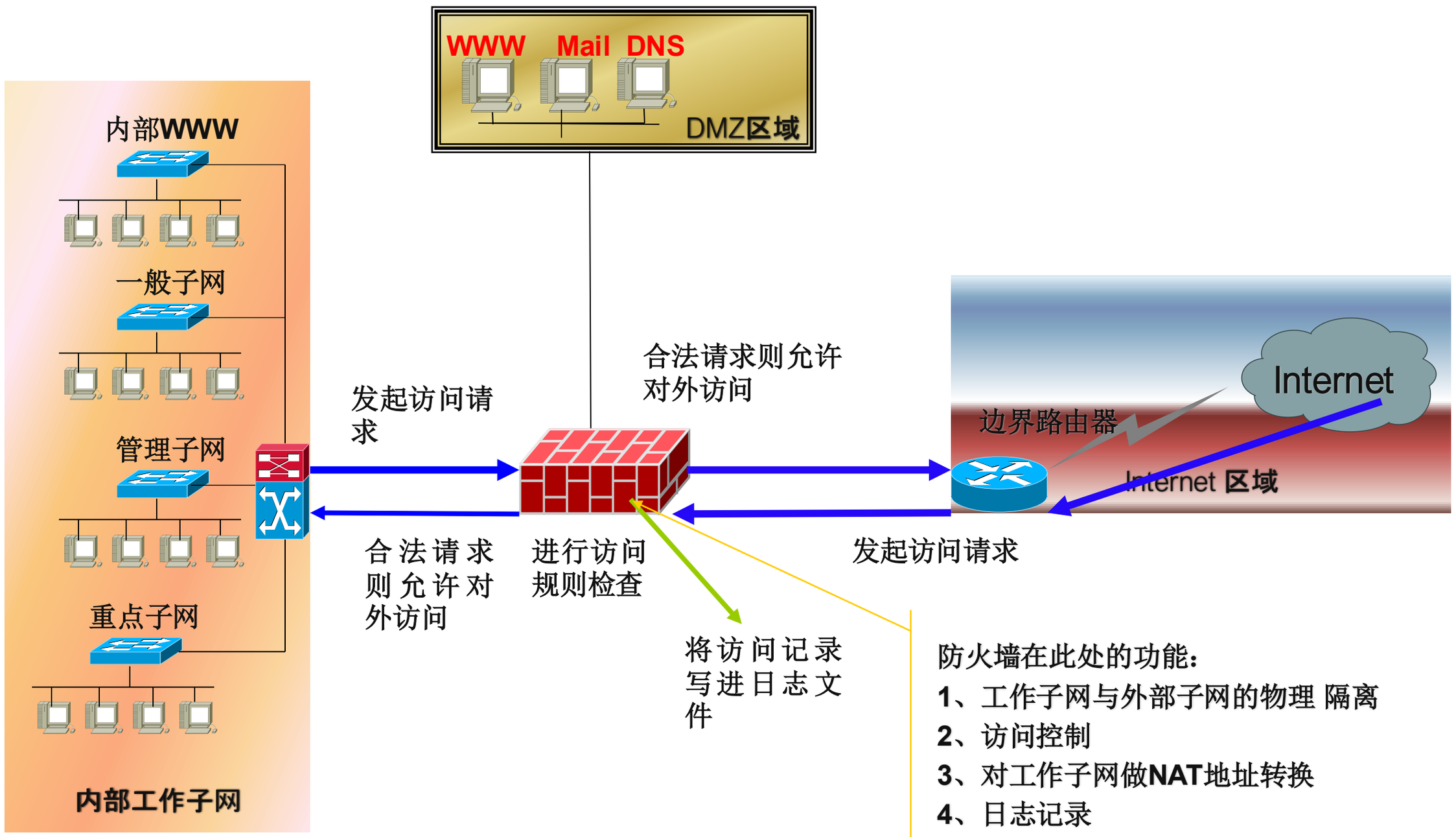
- Biba模型模仿BLP模型的信息保密性级别，定义了信息完整性级别，在信息流向的定义方面不允许从级别低的进程到级别高的进程，也就是说用户只能向比自己安全级别低的客体写入信息，从而防止非法用户创建安全级别高的客体信息，避免越权、篡改等行为的产生。

- Biba模型可同时针对有层次的安全级别和无层次的安全种类。

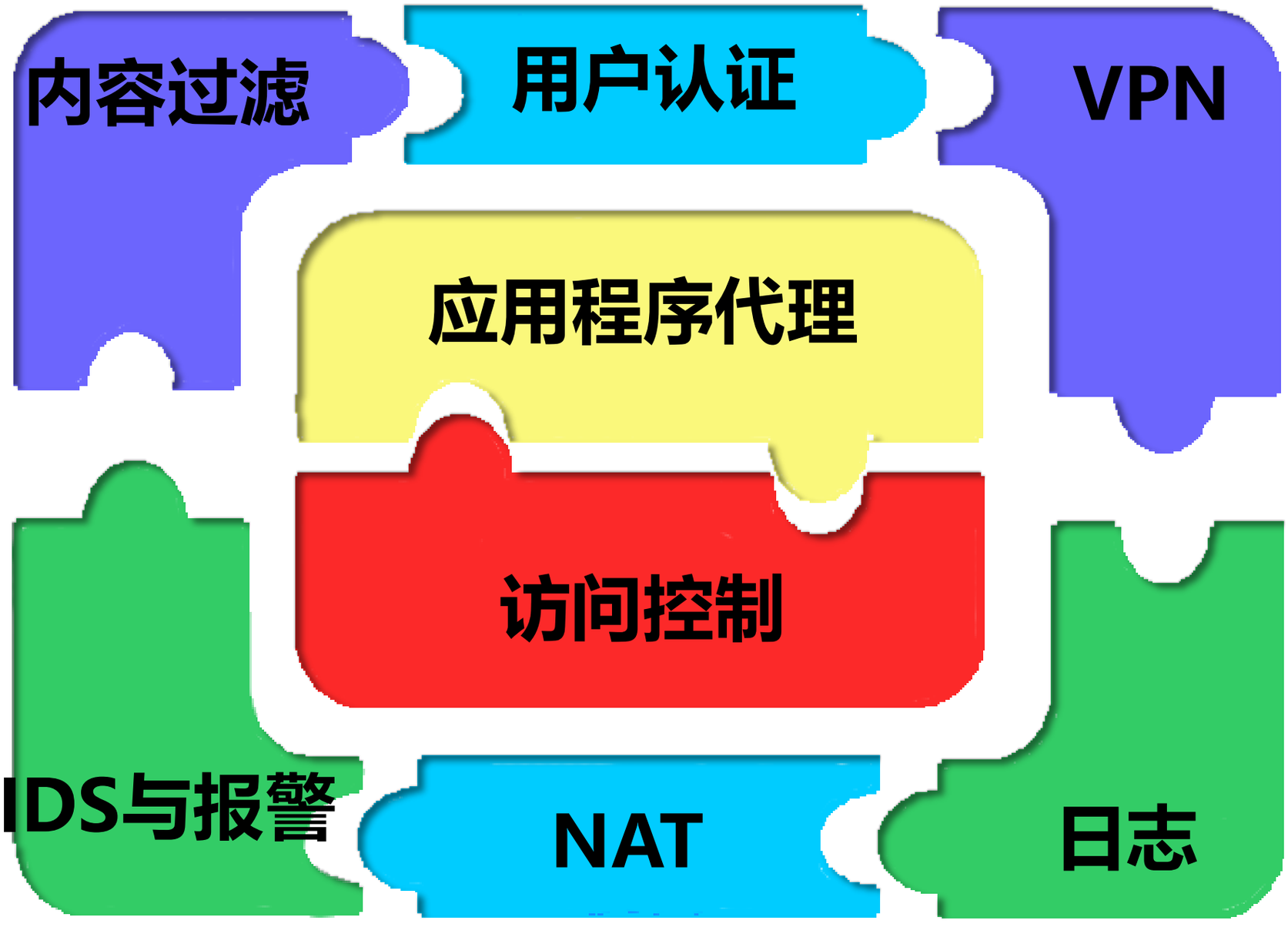
# 防火墙定义

- **防火墙是位于两个(或多个)网络间实施网间访问控制的一组组件的集合。**
- 它满足以下条件
  - **所有进出被保护网络的通信必须通过防火墙**
  - **所有通过防火墙的通信必须经过安全策略的过滤或者防火墙的授权**
  - **防火墙自身应对渗透(peneration)免疫**

# 一个典型的防火墙使用形态



# 防火墙的功能



# 防火墙访问控制方法

- **服务控制**：确定可以访问的服务类型；
- **方向控制**：确定特定的服务请求可以发起并允许通过防火墙；
- **用户控制**：不同的用户具有不同服务访问的权限；
- **行为控制**：控制怎样使用特定服务。如过滤垃圾邮件

# 防火墙分类

**从形**态上分类

软件防火墙

**硬件防火**墙

**从**实现技术分类

**包**过滤防火墙

应用网关防火墙

**代理防火**墙

**状**态检测防火墙

电路级网关

**从部署位置分**类

**主机防火**墙

网络防火墙

# 包过滤防火墙

- 包过滤防火墙对所接收的每个数据包做允许、拒绝的决定。
- 防火墙审查每个数据报以便确定其是否与某一条包过滤规则匹配。
- 过滤规则基于可以提供给IP转发过程的包头信息。
- 分为静态包过滤与动态包过滤两类。
  - 动态包过滤对外出数据包的身份做一个标记，对相同连接的进入的数据包也被允许通过，也就是说，它捕获了一个“连接”，而不是单个数据包头中的信息。

# 包过滤防火墙

静态包过滤
动态包过滤

- 对所接收的每个数据包做允许、拒绝的决定。
- 审查每个数据报以便确定其是否与某一条包过滤规则匹配。

## 判断依据

- 基本信息
  - 地址信息：源、目的IP地址
  - 协议信息：数据包协议类型TCP、 UDP、 ICMP 、IGMP等
  - 源、目的端口FTP、 HTTP 、DNS等
- 协议具体信息
  - IP选项源路由、记录路由等
  - TCP选项SYN 、ACK、 FIN 、RST等
  - 其它协议选项ICMP、 ECHO、ICMP、 ECHO 、REPLY等
- 流向及接口信息
  - 数据包流向in或out
  - 数据包流经网络接口eth0 eth1

# 包过滤防火墙

## 优点

- 逻辑简单，价格便宜，成本低；
- 对网络性能的影响较小，有较强的透明性。
- 易于匹配绝大多数网络层、传输层数据包，定制策略灵活。
- 并且它的工作与应用层无关，无须改动任何客户机和主机上的应用程序，易于安装和使用。

## 缺点

- 需要对IP、TCP、UDP、ICMP等协议有深入了解，否则容易出现因配置不当带来的问题；
- 据以过滤判别的只有网络层和传输层的有限信息，因而各种安全要求不能得到充分满足；
- 由于数据包的地址及端口号都在数据包的头部，不能彻底防止IP地址欺骗；
- 允许外部客户和内部主机的直接连接；
- 不提供用户的鉴别机制；
- 仅工作在网络层，提供较低水平的安全性。

# 包过滤防火墙原理

| Source | Destination | Permit | Protocol |
|---|---|---|---|
| Host A | Host C | Pass | TCP |
| Host B | Host C | Block | UDP |

控制策略

查找对应的控制策略

根据策略决定如何处理该数据包

拆开数据包

数据包

数据包

数据包

数据包

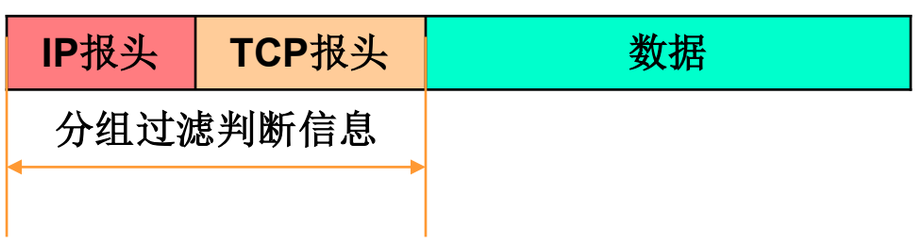
安全网域

Host CHost D

数据包

数据包

过滤依据主要是TCP/IP报头里面的信息，不能对应用层数据进行处理

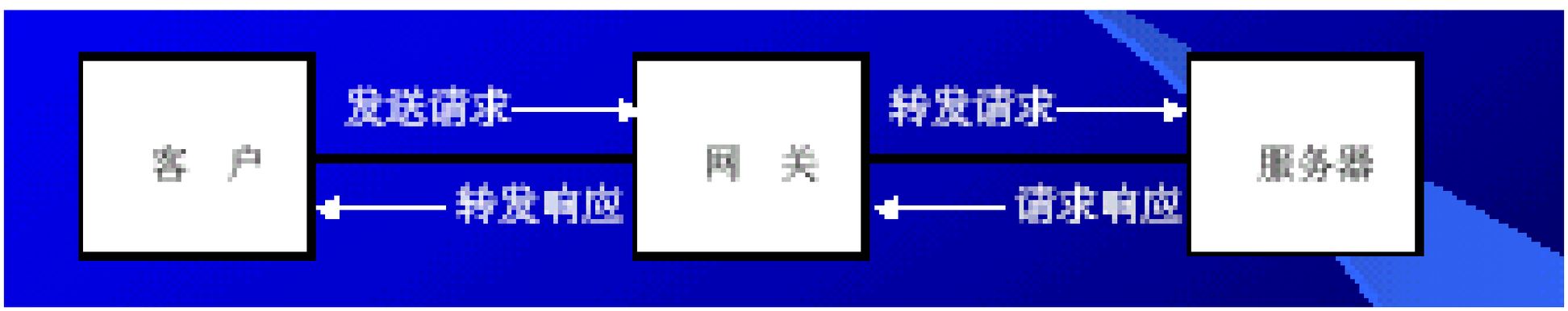


# 应用网关

- 应用网关防火墙（AGF,Application Gateway Firewall），**又称代理防火**墙或简称应用网关。
- 应用网关在应用层处理信息
- AGF**可以支持多个**应用，如E-mail，Web，DNS，Telnet，FTP**等**

# 应用网关

- 应用网关代理服务器的工作过程为：
  - **首先，它**对该用户的身份进行验证。
  - **若**为合法用户，则把请求转发给真正的某个内部网络的**主机，同**时监控用户的操作，拒绝不合法的访问（访问权限）。
  - **当内部网**络向外部网络申请服务时，代理服务器的工作过程刚好相反。（认证输入、输出两个方向的连接）



# 应用网关

## 优点

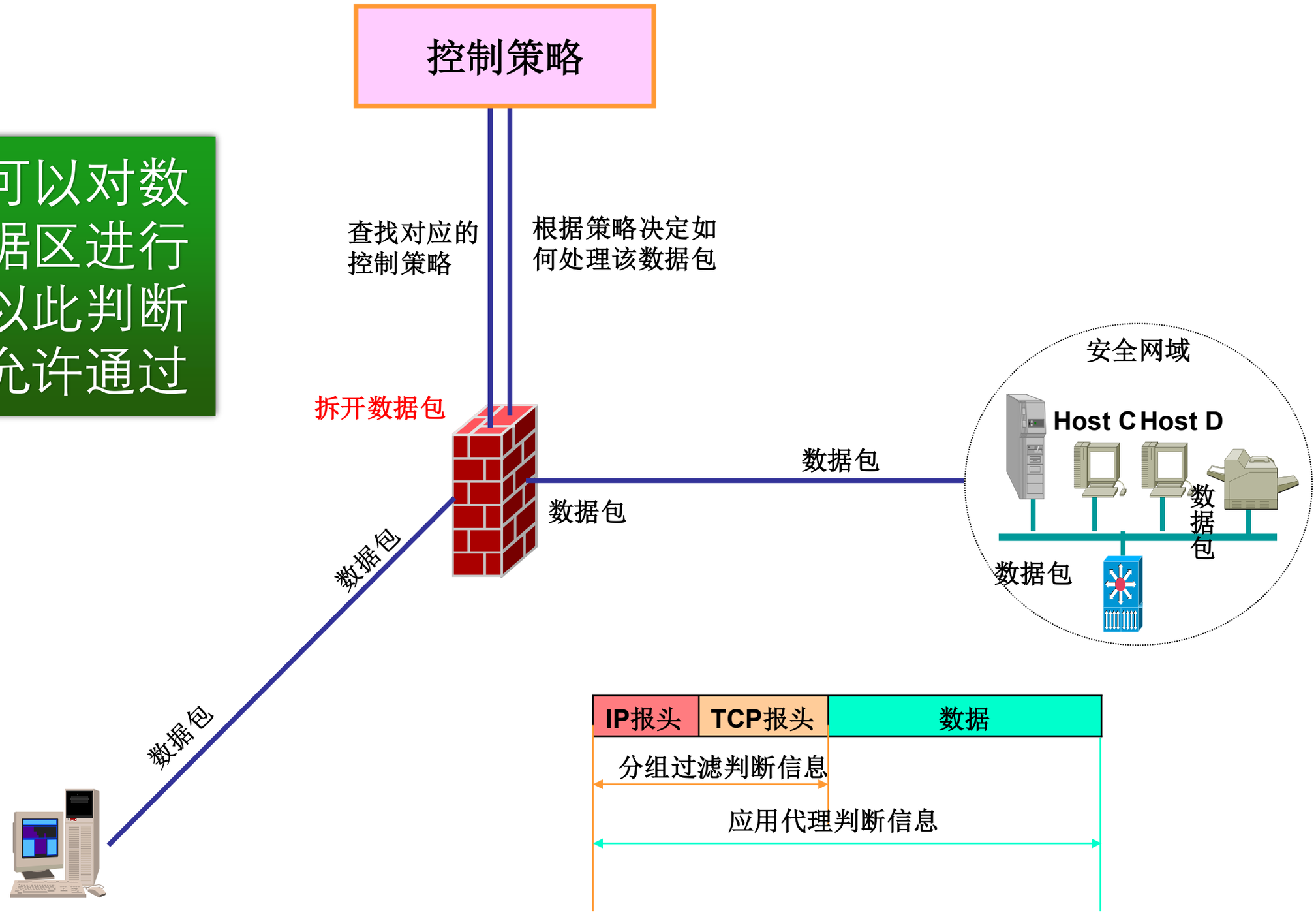
- 不允许内外网主机的直接连接；
- 可以提供比包过滤更详细的日志记录（应用层信息）；
- 可以隐藏内部IP地址；
- 认证用户而非设备；
- 可以为用户提供透明的加密机制；
- 可以与认证、授权等安全手段方便的集成；
- 监控、过滤应用层信息；

## 缺点

- 代理速度比包过滤慢；
- 代理对用户不透明，给用户的使用带来不便，灵活性不够；
- 这种代理技术需要针对每种协议设置一个不同的代理服务器；
- 有时要求特定的客户端软件。

# 应用网关

应用代理可以对数据包的数据区进行分析，并以此判断数据是否允许通过



# 状态检测防火墙

- 状态检测防火墙是在动态包过滤防火墙基础上，增加了状态检测机制而形成的。
- 具有连接的跟踪能力。

## 示例

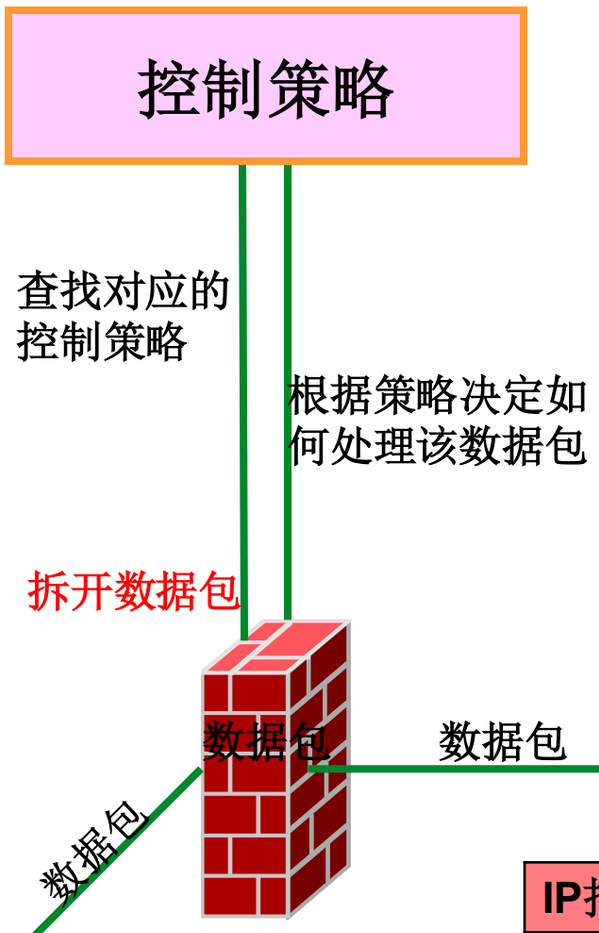
- 以TCP协议为例：所谓的状态检测机制关注的主要问题不再仅是SYN和ACK标志位，或者是来源端口和目标端口，还包括了序号、窗口大小等其它TCP协议信息。

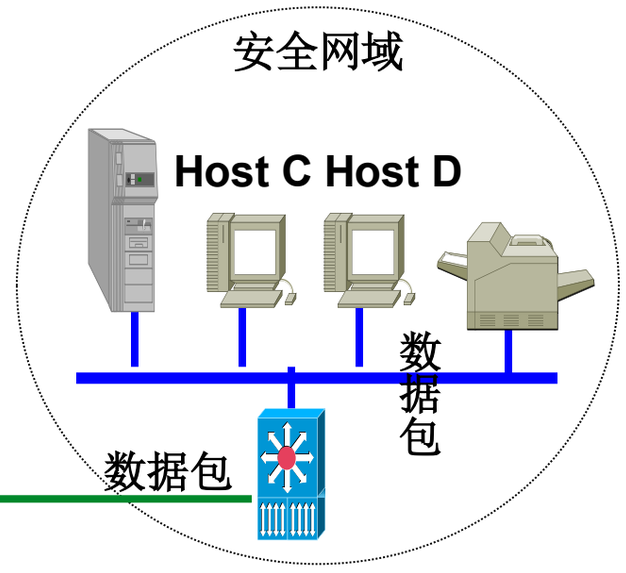
## 优缺点

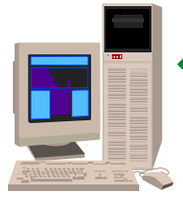
- 优点：
  - 具备动态包过滤的所有优点，同时具有更高的安全性。
- 缺点：
  - 检测的层次仅限于网络层与传输层，无法对应用层内容进行检测，从而无法抵抗应用层的攻击；
  - 性能比动态包过滤稍差：因为检测更多的内容。

# 状态检测原理

状态检测可以结合前后数据包里的数据信息进行综合分析决定是否允许该包通过

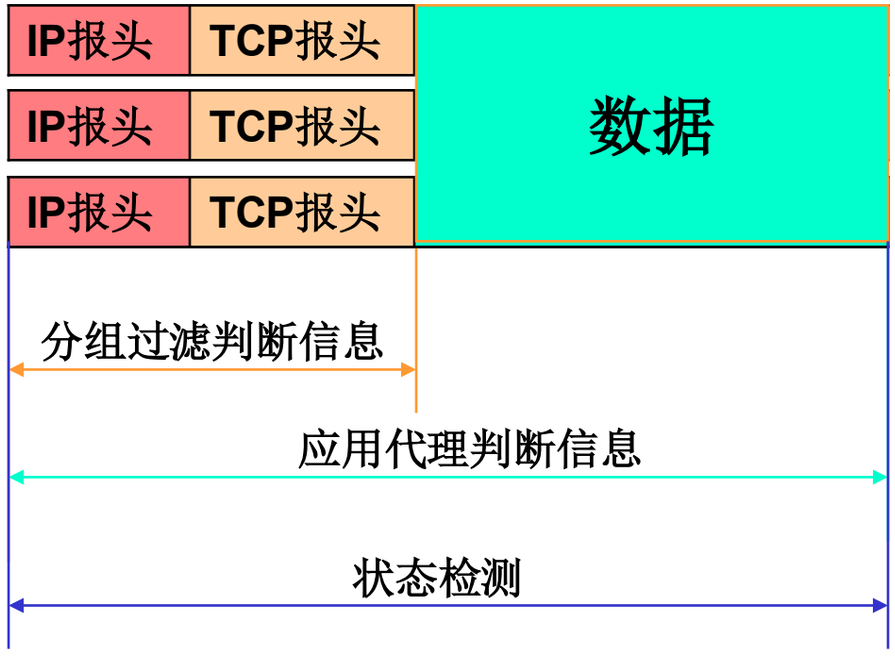


数据包

数据包



# 网络地址翻译(NAT)

- 目的
  - 解决IP地址空间不足问题
  - 向外界隐藏内部网结构
- 方式
  - M-1 多个内部网地址翻译到1个IP地址
  - 1-1 简单的地址翻译
  - M-N 多个内部网地址翻译到N个IP地址池

# NAT技术

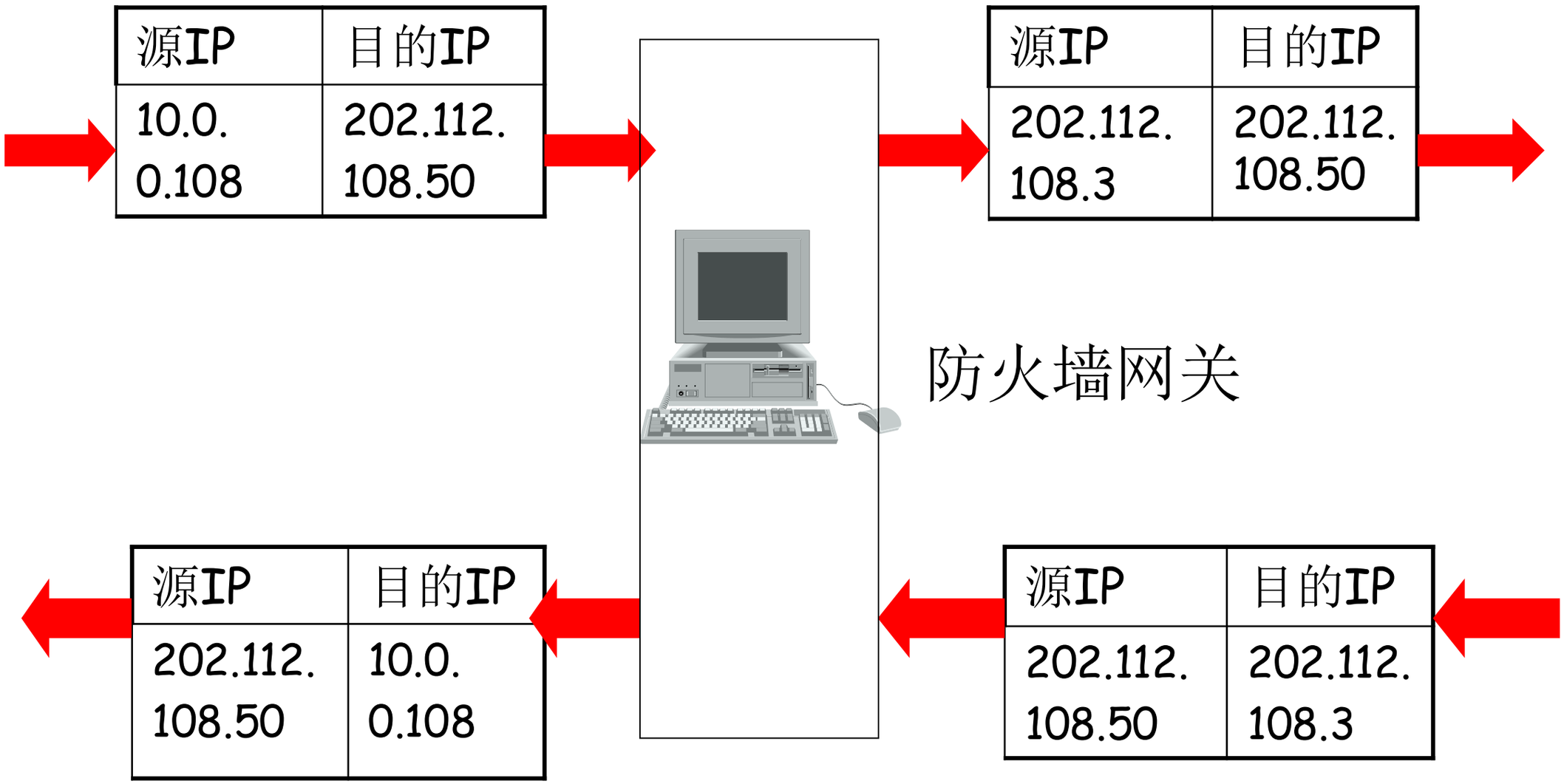
- 地址翻译NAT（Network Address Translation）就是将一个IP地址用另一个IP地址代替。地址翻译主要用在两个方面：
  - 网络管理员希望隐藏内部网络的IP地址。这样互联网上的主机无法判断内部网络的情况。
  - 内部网络的IP地址是无效的IP地址。 这种情况主要是因为现在的IP地址不够用， 要申请到足够多的合法IP地址很难办到，因此需要翻译IP地址。

# NAT技术

- NAT的三种类型
  - 静态NAT
    - 内部网络每个主机都永久映射成外部合法的地址
  - NAT池
    - 在外部网络中定义了一系列合法地址，采用动态分配的方法映射到内部网络
  - 端口NAT（PNAT）
    - 把内部地址映射到外部网络的一个IP地址的不同端口上

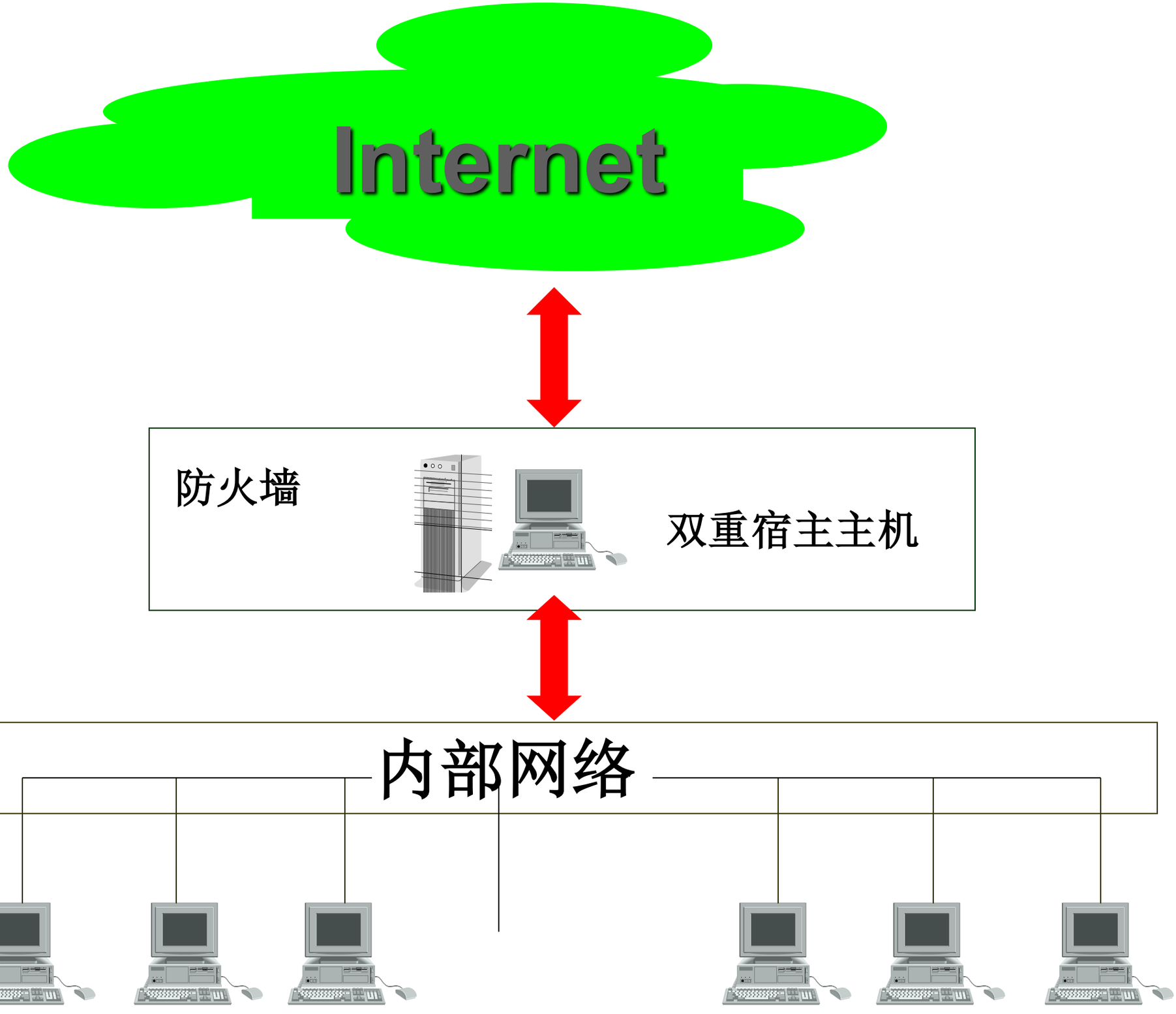
# NAT技术

- **基本原理**


源IP
目的IP
10.0.0.108
202.112.108.50
源IP
目的IP
202.112.108.3
202.112.108.50
防火墙网关
源IP
目的IP
202.112.108.50
10.0.0.108
源IP
目的IP
202.112.108.50
202.112.108.3

# 双重宿主主机体系结构

- 双重宿主主机体系结构是围绕双重宿主主机构筑的。
- 双重宿主主机至少有两个网络接口
  - 它位于内部网络和外部网络之间，这样的主机可以充当与这些接口相连的网络之间的路由器，它能从一个网络接收IP数据包并将之发往另一网络。
  - 双重宿主主机的防火墙体系结构禁止这种发送功能，完全阻止了内外网络之间的IP通信。
- 两个网络之间的通信可通过应用层数据共享和应用层代理服务的方法实现。一般情况下采用代理服务的方法。



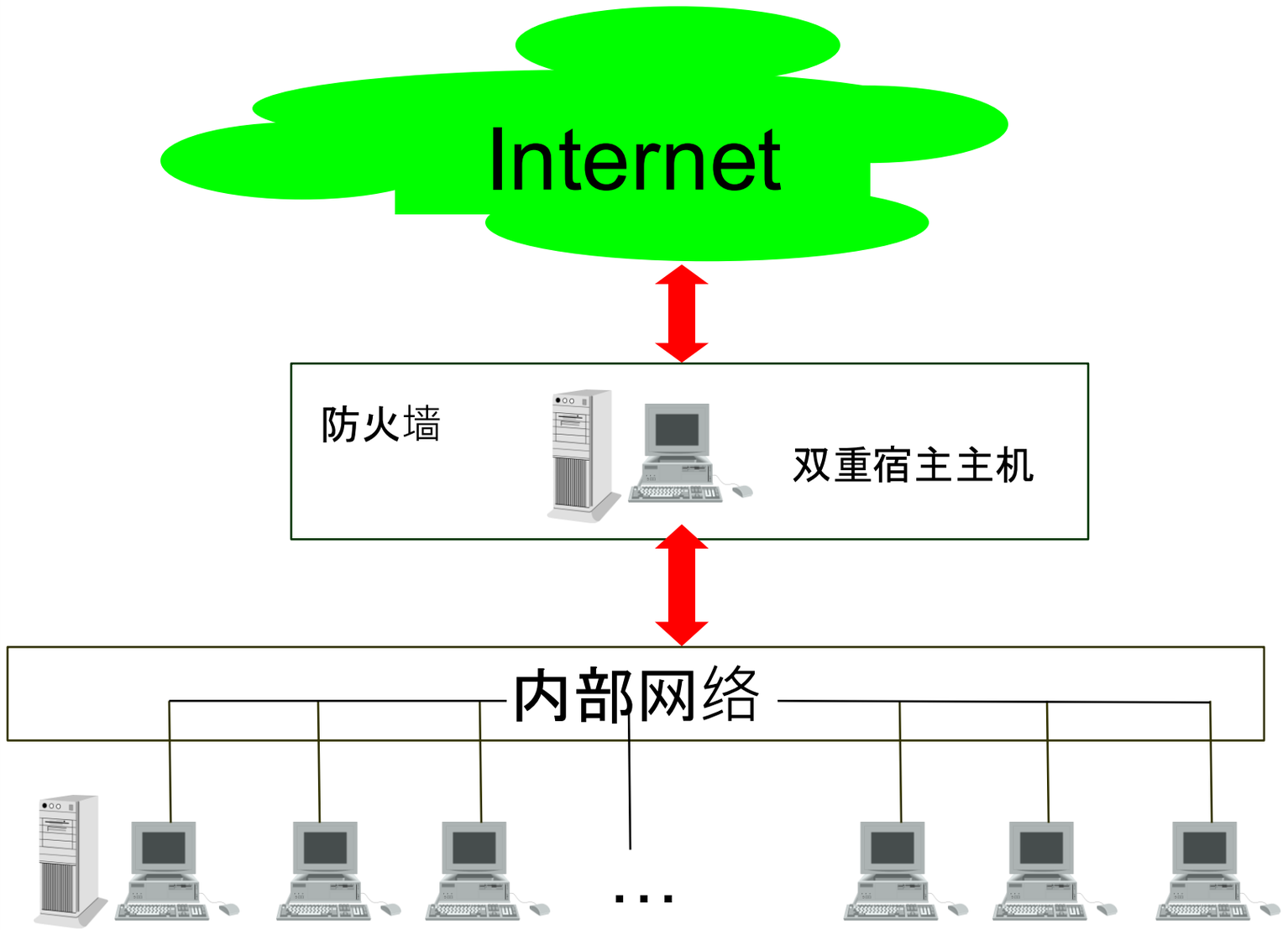

双重宿主主机体系结构

# 双重宿主主机体系结构

- 双重宿主主机的特性：
  - 安全至关重要（唯一通道），其用户口令控制安全是关键
  - 必须支持很多用户的访问（中转站），其性能非常重要。
- 缺点：双重宿主主机是隔开内外网络的唯一屏障，一旦它被入侵，内部网络便向入侵者敞开大门。

# 双重宿主主机体系结构

- 双重宿主主机的特性：
  - 安全至关重要（唯一通道），其用户口令控制安全是关键
  - 必须支持很多用户的访问（中转站），其性能非常重要。
- 缺点：双重宿主主机是隔开内外网络的唯一屏障，一旦它被入侵，内部网络便向入侵者敞开大门。



# 屏蔽主机体系结构

- 屏蔽主机体系结构由防火墙和内部网络的堡垒主机承担安全责任。一般这种防火墙较简单，可能就是简单的路由器。

- 典型构成：**包过滤路由器**＋**堡垒主机**。
  - 包过滤路由器配置在内部网和外部网之间，保证外部系统对内部网络的操作只能经过堡垒主机。
  - 堡垒主机配置在内部网络上，是外部网络主机连接到内部网络主机的桥梁，它需要拥有高等级的安全。

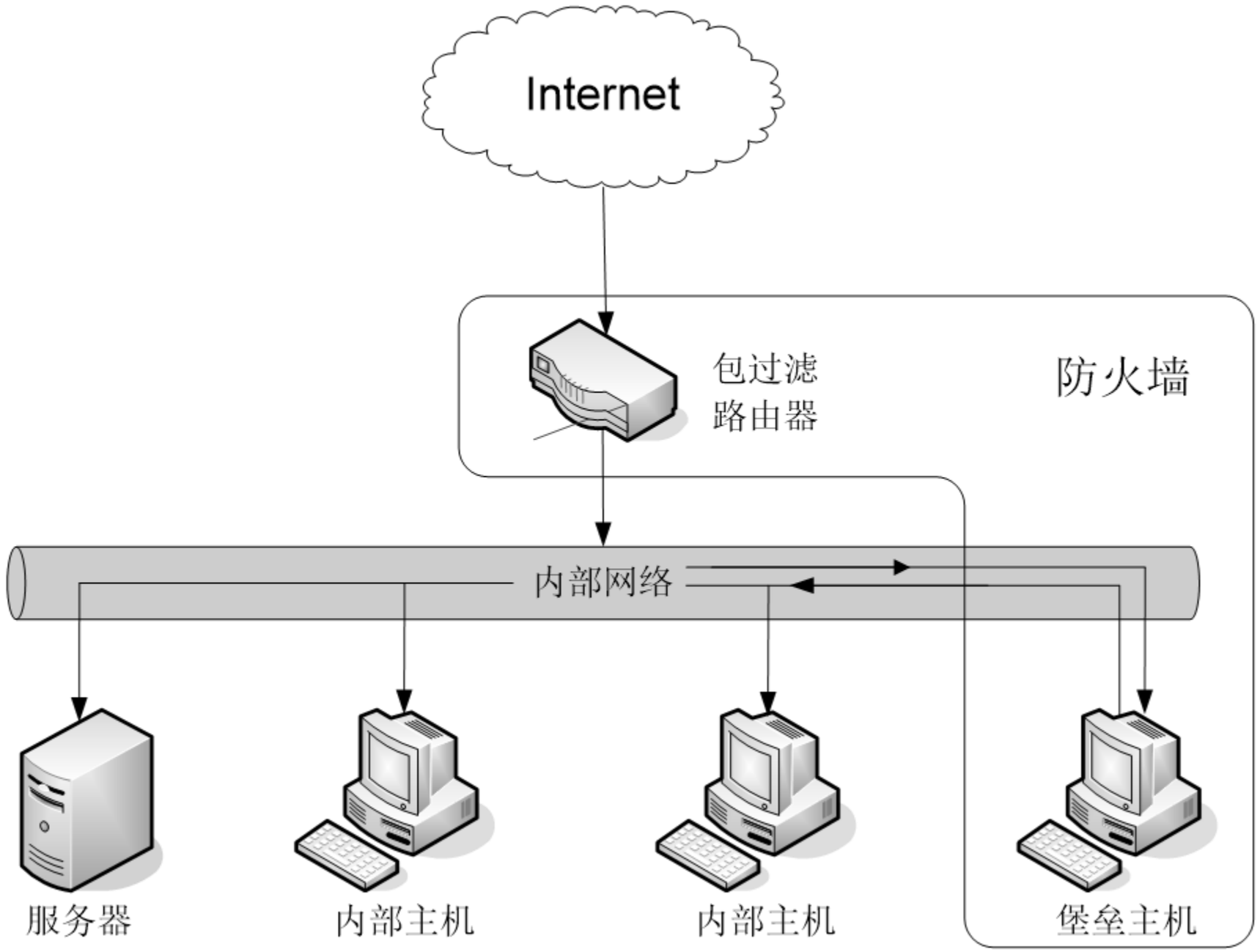
# 屏蔽主机体系结构

- 屏蔽路由器可按如下规则之一进行配置：
  - 允许内部主机为了某些服务请求与外部网上的主机建立直接连接（即允许那些经过过滤的服务）。
  - 不允许所有来自外部主机的直接连接。

- 安全性更高，双重保护：实现了网络层安全（包过滤）和应用层安全（代理服务）。

- 缺点：过滤路由器能否正确配置是安全与否的关键。如果路由器被损害，堡垒主机将被穿过，整个网络对侵袭者是开放的。

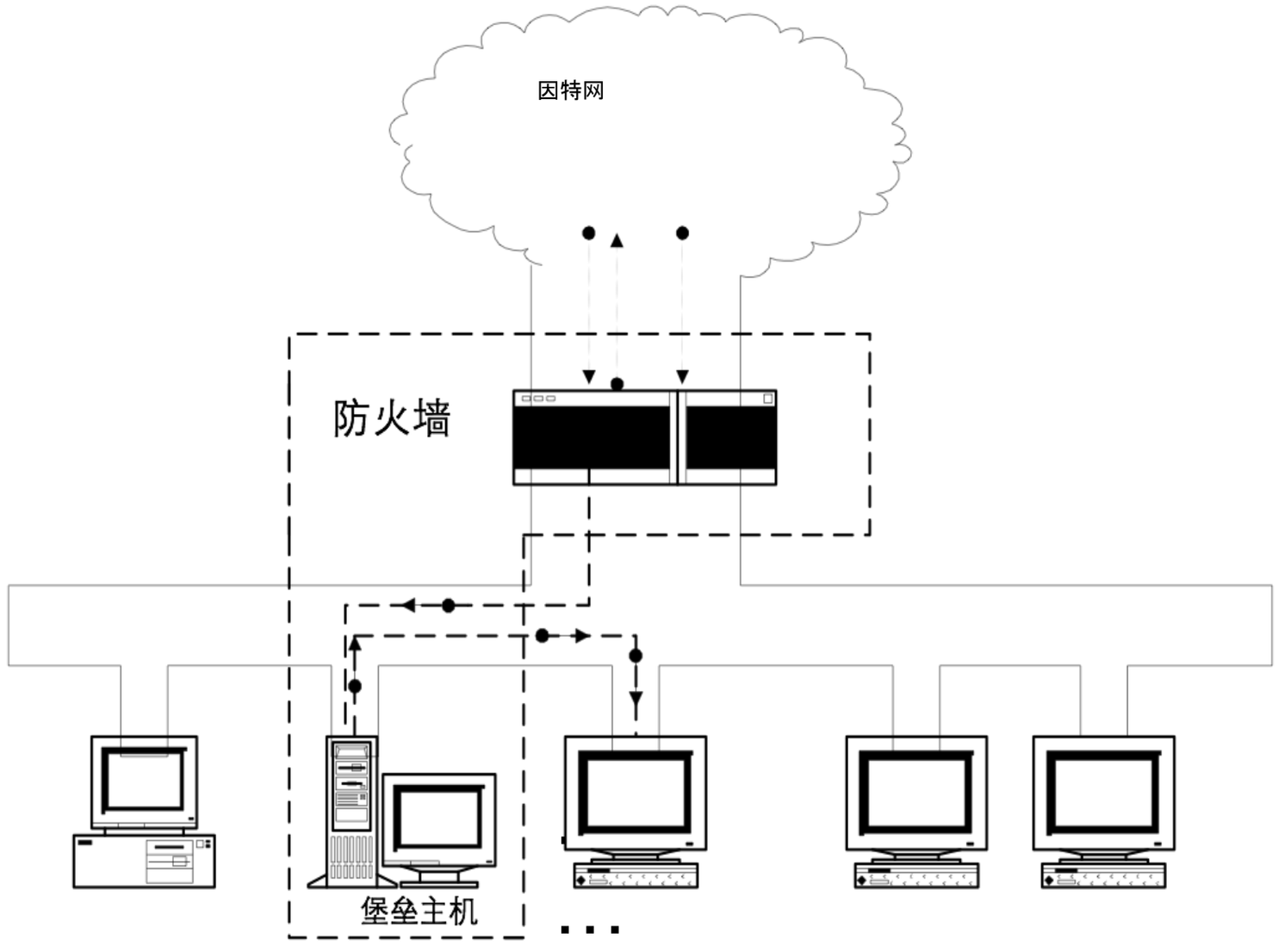
# 屏蔽主机体系结构



# 屏蔽主机体系结构

- 典型构成：包过滤路由器 + 堡垒主机。
  - 包过滤路由器保证外部系统对内部网络的操作只能经过堡垒主机
  - 堡垒主机配置在内部网络上，是外部网络主机连接到内部网络主机的桥梁。

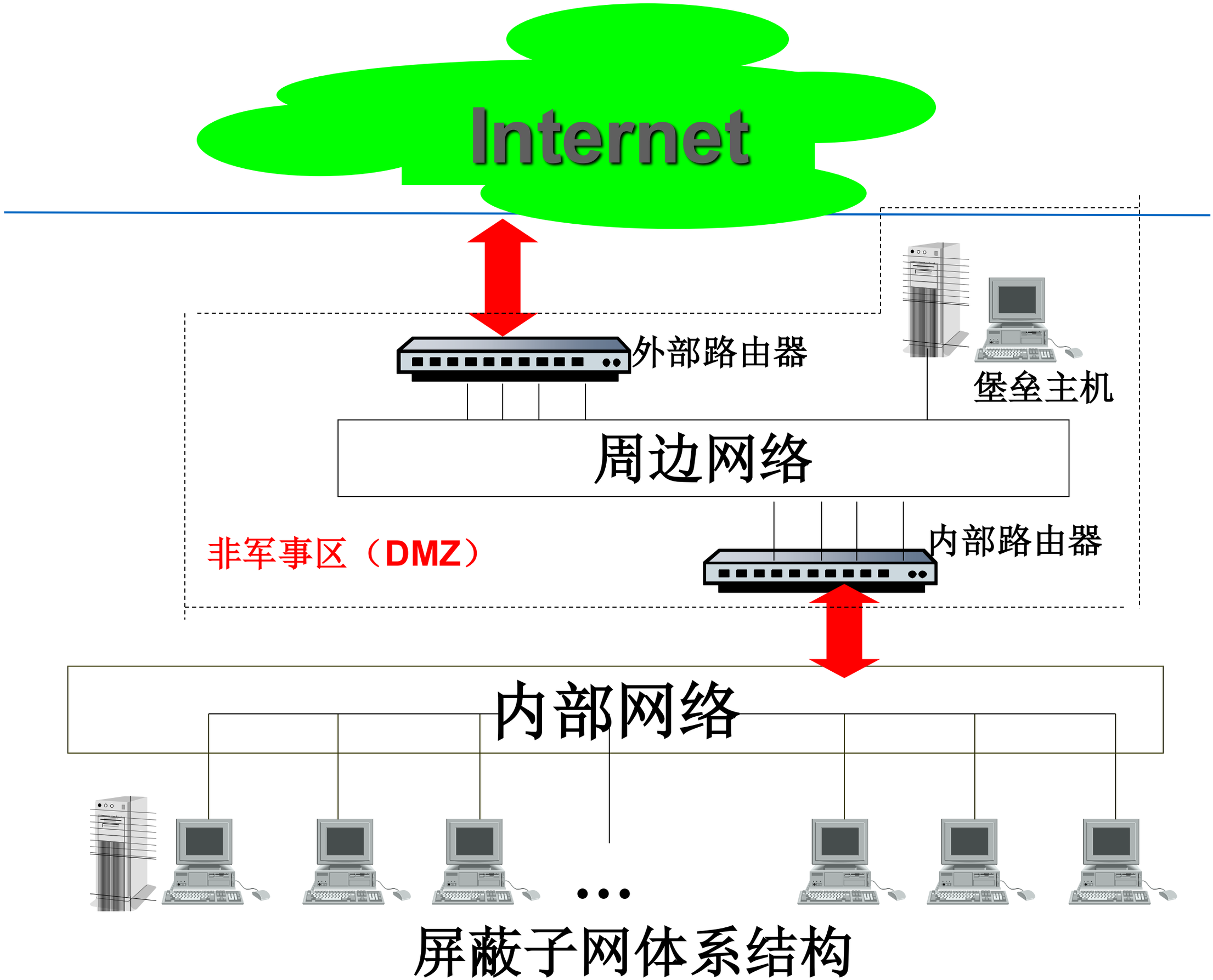
- 安全性更高，双重保护：实现了网络层安全（包过滤）和应用层安全
- 缺点：过滤路由器能否正确配置是安全与否的关键。如果路由器被损害，堡垒主机将被穿过。



# 屏蔽子网体系结构

- **屏蔽子网体系**结构在本质上与屏蔽主机体系结构一样，但添加了额外的一层保护体系——**周**边网络。**堡**垒主机位于周边网络上，周边网络和内部网络被**内部路由器分开**。

- **原因**：**堡**垒主机是用户网络上最容易受侵袭的机器。**通**过在周边网络上隔离堡垒主机，能减少在堡垒主**机被侵入的影响**。




屏蔽子网体系结构

# 屏蔽子网体系结构

- 周边网络是一个防护层，在其上可放置一些信息服务器，它们是牺牲主机，可能会受到攻击，因此又被称为非军事区（DMZ）。

- 周边网络的作用：即使堡垒主机被入侵者控制，它仍可消除对内部网的侦听。例: netxray等的工作原理。

# 屏蔽子网体系结构

- 堡垒主机
  - 堡垒主机位于周边网络，是整个防御体系的核心。
  - 堡垒主机可被认为是应用层网关，可以运行各种代理服务程序。
  - 对于出站服务不一定要求所有的服务经过堡垒主机代理，但对于入站服务应要求所有服务都通过堡垒主机。

# 屏蔽子网体系结构

- **外部路由器**（访问路由器）
  - **作用**：保护周边网络和内部网络不受外部网络的侵犯。
    - **它把入站的数据包路由到堡垒主机。**
    - **防止部分IP欺骗，它可分辨出数据包是否真正来自周边网络，而内部路由器不可。**

- **内部路由器（阻塞路由器）**
  - **作用**：保护内部网络不受外部网络和周边网络的侵害，它执行大部分过滤工作。
  - **外部路由器一般与内部路由器**应用相同的规则。

# 屏蔽子网体系结构优点

- 入侵者需突破3个不同的设备才能入侵内部网络
- 只对外通告DMZ区的网络，保证内部网络不可见
- 内部网络用户通过堡垒主机或代理服务器访问外部网络

# 网络安全——

# IDS

# IDS的历史

- IDS：intrusion detection system，入侵检测系统：
  - 概念的诞生：1980年4月，James P。Anderson 为美国空军做了题为《计算机安全威胁监控与监视》的技术报告，第一次详细阐述了入侵检测的概念,将威胁分为外部渗透、内部渗透和不法行为三种；
  - 模型的发展：1984-1986年，Dorothy Denning 和 Peter Neumann研究出了一个实时入侵检测系统，取名为：IDES（入侵检测专家系统）；
  - 里程碑的产生：1990年，加州大学的L.T.Heberlein等人开发出了NSM（network security monitor）。该系统第一次直接将网络流作为审计数据来源；从此入侵检测系统发展史形成了两大阵营：基于网络的IDS和基于主机的IDS。

# 入侵检测概念

- 入侵检测是**从计算机网络或计算机系统中的若干关键点搜集信息并对其进行分析，从中发现网络或系统中是否有违反安全策略的行为和遭到袭击的迹象的一种机制**。

- IDS，Intrusion Detection System，**入侵**检测系统

# 入侵检测概念

- 一种主动保护自己的网络和系统免遭非法攻击的网络安全技术。

- **它从**计算机系统或者网络中收集、分析信息，检测**任何企**图破坏计算机资源的完整性、机密性和可用**性的行**为，即查看是否有违反安全策略的行为和遭**到攻**击的迹象，并做出相应的反应。

- **安全**审计中的核心技术之一

# 入侵检测的分类（1）

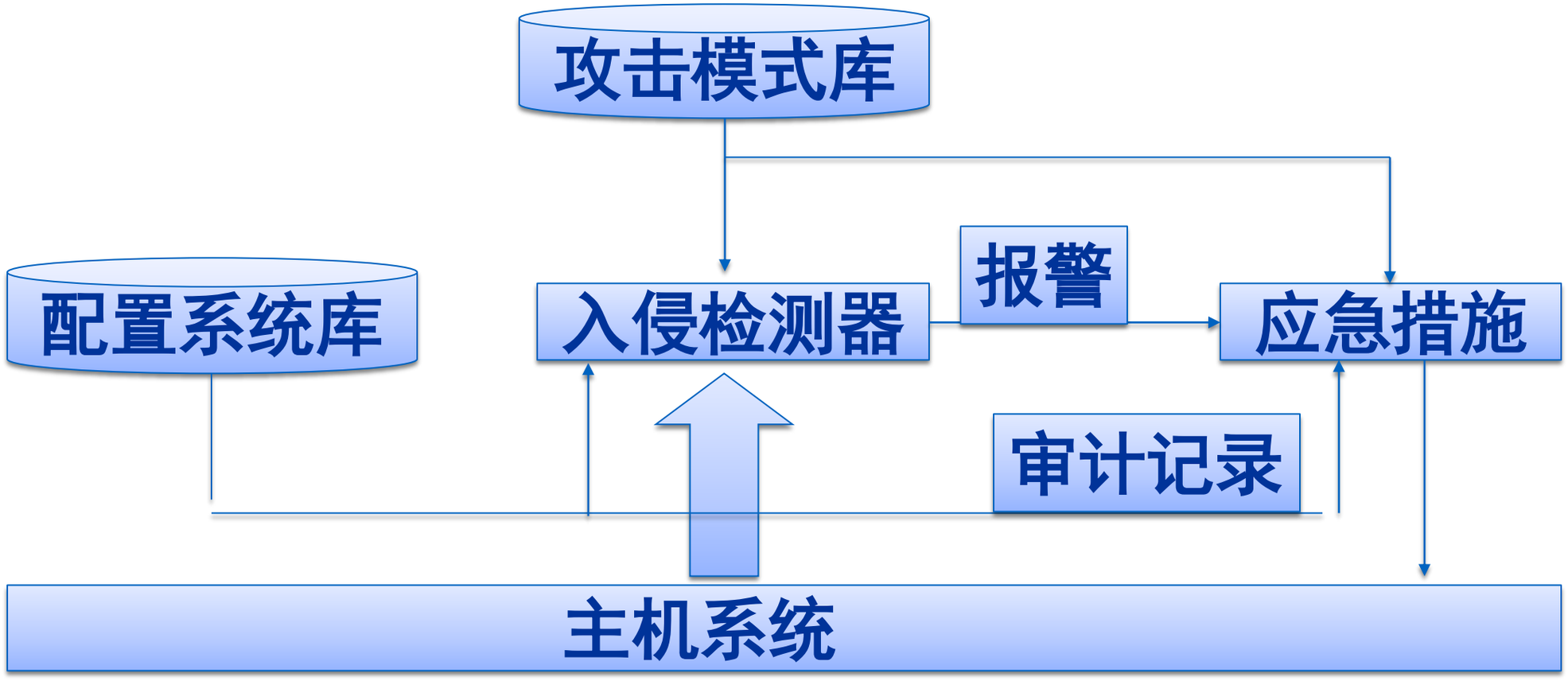
- **根据原始数据的来源**
  - **基于主机的入侵检测系统**：监控粒度更细、配置灵活、可用于加密的以及交换的环境
  - **基于网络的入侵检测系统**：视野更宽、隐蔽性好、攻击者不易转移证据
- **根据检测原理**
  - **异常入侵检测**：根据异常行为和使用计算机资源的情况检测出来的入侵。
  - **误用入侵检测**：利用已知系统和应用软件的弱点攻击模式来检测入侵。

# 入侵检测的分类（2）

- **根据体系结构**
  - **集中式**：多个分布于不同主机上的审计程序，一个入侵检测服务器
  - **等级式**：定义了若干个分等级的监控区域，每个IDS负责一个区域，然后将当地的分析结果传送给上一级IDS
  - **协作式**：将中央检测服务器的任务分配给多个基于主机的IDS，这些IDS不分等级，各司其职，负责监控当地主机的某些活动。
- **根据工作方式分类**
  - **离线检测**：非实时工作系统，在事件发生后分析审计事件，从中检查入侵事件。
  - **在线检测**：对网络数据包或主机的审计事件进行实时分析，可以快速反应，保护系统的安全；但系统规模较大时，难以保证实时性。

# 基于主机系统结构

- HIDS：Host-Based IDS
- 检测的目标主要是主机系统和系统本地用户。检测原理**是根据主机的**审计数据和系统的日志发现可疑事件，检测系统可以运行在被检测的主机或单独的主机上。

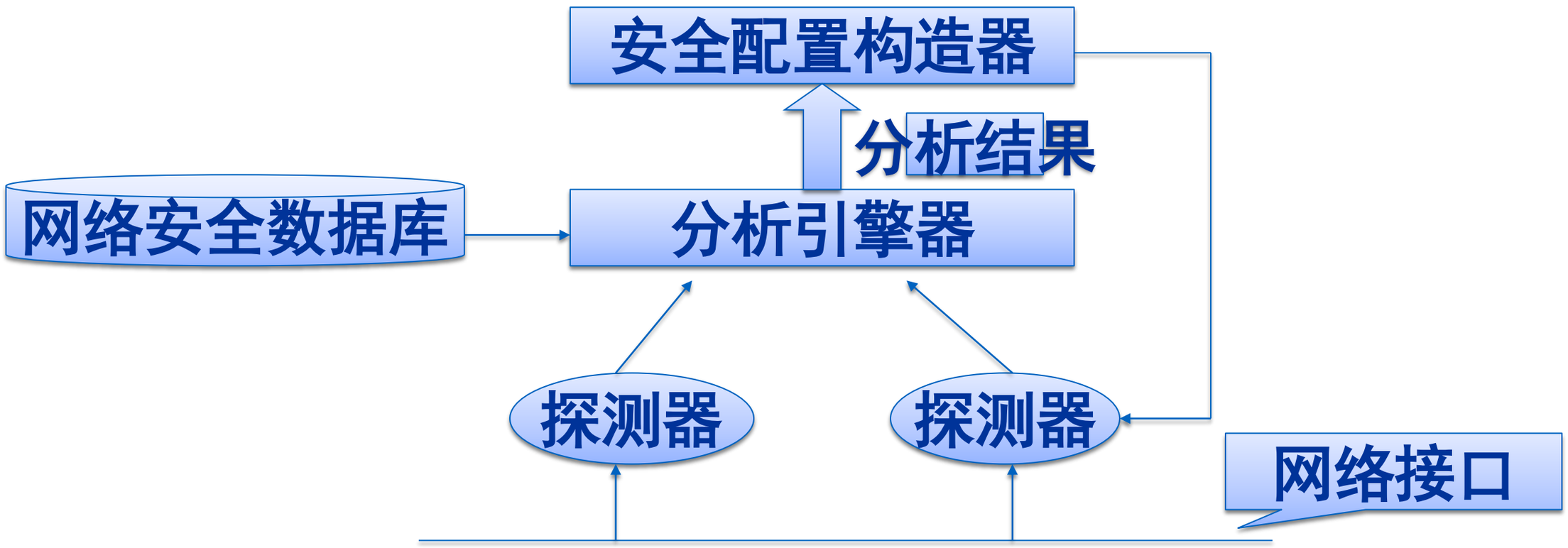


# HIDS

- HIDS优点
  - 性能价格比高
  - 细腻性，审计内容全面
  - 视野集中
  - 适用于加密及交换环境
- HIDS缺点
  - IDS的运行影响服务器的性能
  - HIDS依赖性强，依赖于审计数据或系统日志准确性和完整性，以及安全事件的定义。
  - 如果主机数目多，代价过大
  - 不能监控网络上的情况

# 基于网络系统结构

- NIDS：Network-Based IDS
- **根据网**络流量、协议分析、单台或多台主机的审计**数据**检测入侵。



# 基于网络系统结构

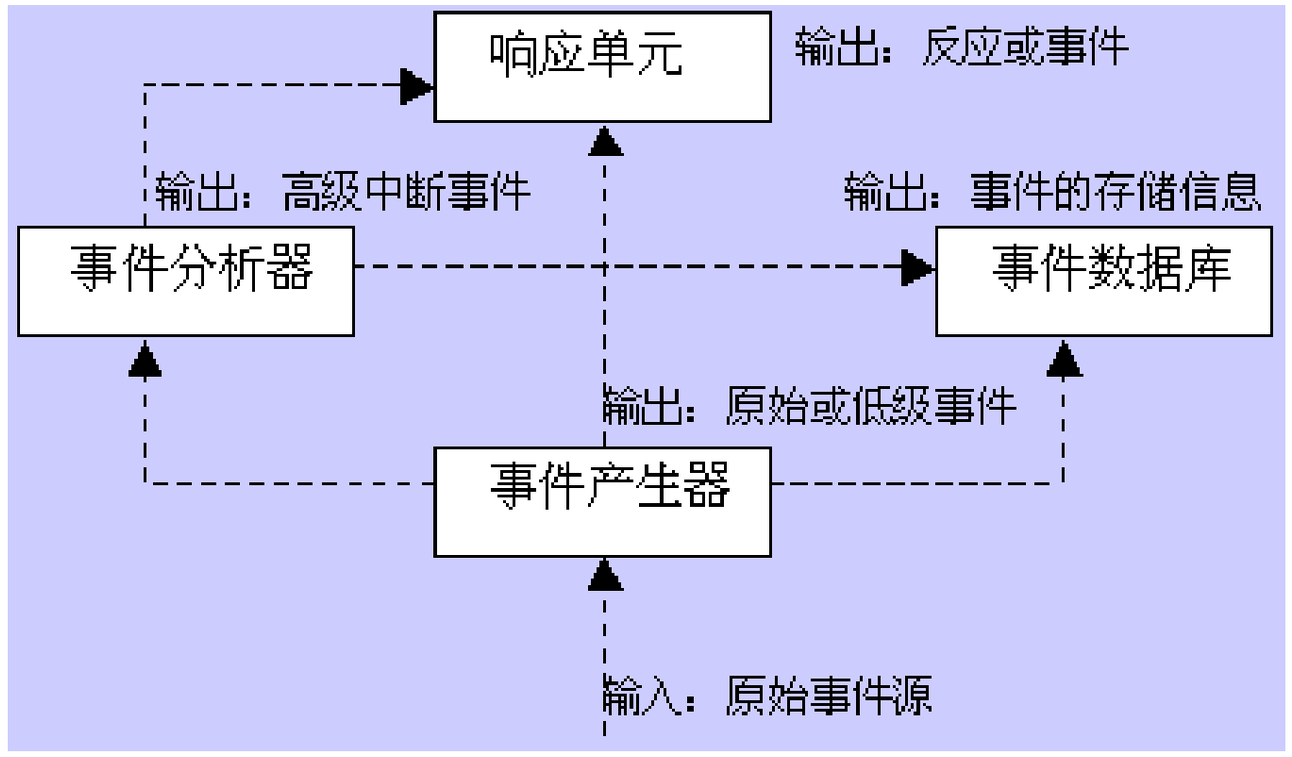
- NIDS优点：
  - **服**务器平台独立性：监视通信流量而不影响服务器的平台的变化和更新；
  - **配置**简单：只需要一个普通的网络访问接口即可；
  - **众多的攻**击标识：探测器可以监视多种多样的攻击包括协议攻击和特定环境的攻击。
- NIDS**缺点**
  - **不能**检测不同网段的网络包
  - **很**难检测复杂的需要大量计算的攻击
  - 协同工作能力弱
  - 难以处理加密的会话

# 入侵检测结构

- CIDF

- Common Intrusion Detection Framework

- 由DARPA于1997年3月开始着手制定




- 事件产生器(Event Generators)

- 事件分析器(Event analyzers)

- 响应单元(Response units)

- 事件数据库(Event databases)

# CIDF

- 事件产生器
  - 事件产生器的目的是从整个计算环境中获得事件，并向系统的其他部分提供此事件。
  - 入侵检测的第一步
  - 采集内容包括系统日志、应用程序日志、系统调用、网络数据、用户行为、其他IDS的信息
- 事件分析器
  - 事件分析器分析得到的数据，并产生分析结果。
  - 分析是核心，效率高低直接决定整个IDS性能

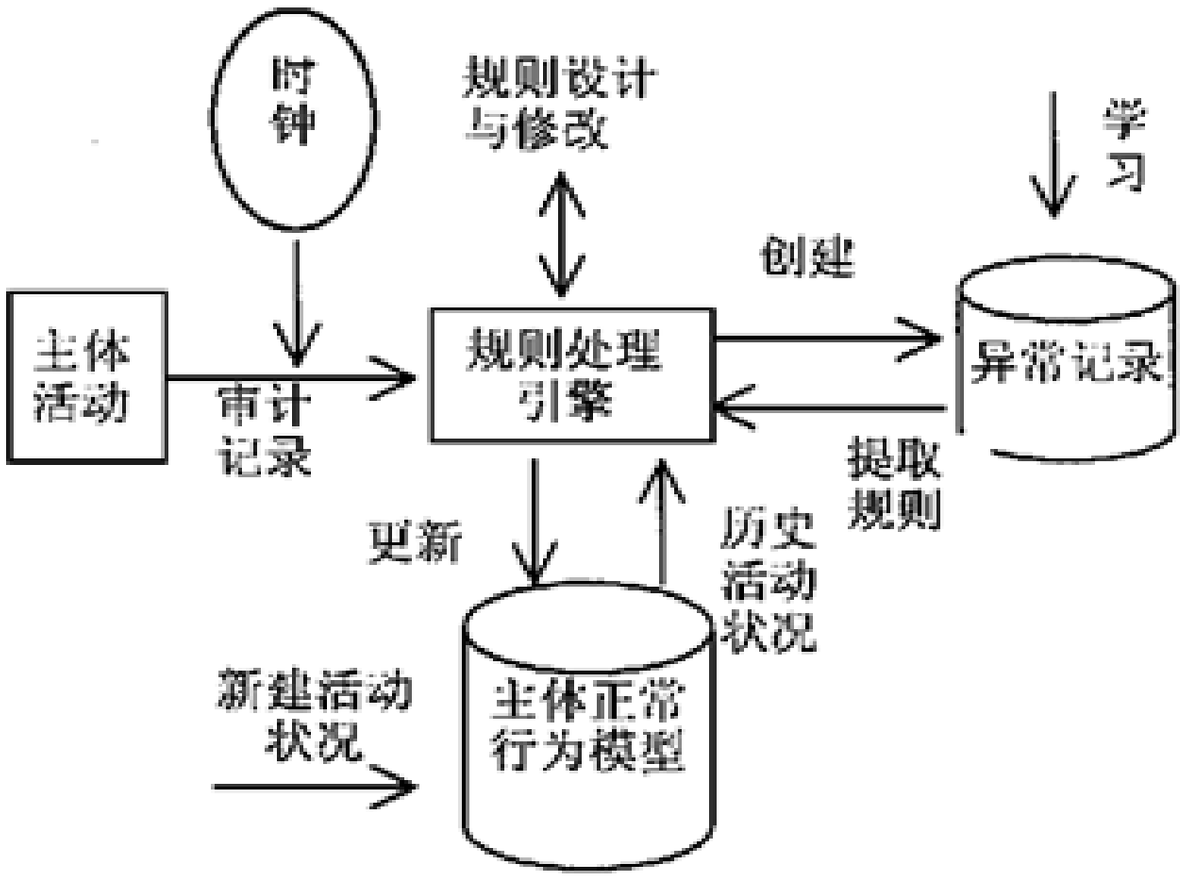
# CIDF

- 响应单元
  - 响应单元则是对分析结果作出作出反应的功能单元，功能包括：
    - 告警和事件报告
    - 终止进程，强制用户退出
    - 切断网络连接，修改防火墙设置
    - 灾难评估，自动恢复
    - 查找定位攻击者
- 事件数据库
  - 事件数据库是存放各种中间和最终数据的地方的统称，它可以是复杂的数据库，也可以是简单的文本文件。

# 入侵检测结构

- Denning模型
- 1987年提出的一个通用入侵检测模型

- 主体(Subjects)：在目标系统上活动的实体，如用户。
- 对象(Objects)：指系统资源，如文件、设备、命令等。
- 审计记录(Audit records)：由主体、活动(主体对目标的操作)、异常条件(系统对主体的该活动的异常情况的报告)、资源使用状况(系统的资源消耗情况)和时间戳(Time-Stamp)等组成。
- 活动档案(Active Profile)：即系统正常行为模型，保存系统正常活动的有关信息。
- 异常记录(Anomaly Record)：由事件、时间戳和审计记录组成，表示异常事件的发生情况。
- 活动规则(Active Rule)：判断是否为入侵的准则及相应要采取的行动。。



# 入侵检测技术

异常检测技术　　误用检测技术

# 异常检测技术

- **思想：任何正常人的行**为有一定的规律，而入侵会引起用户**或系**统行为的异常
- **需要考**虑的问题：
  - 选择哪些数据来表现用户的行为
  - **通**过以上数据如何有效地表示用户的行为，主要在于学习和检测方法的不同
  - **考**虑学习过程的时间长短、用户行为的时效性等问题
- **典型算法**：统计分析 ( 均值、偏差、Markov过程)

# 异常检测技术

- 异常检测模型（Anomaly Detection）：
  - 首先总结正常操作应该具有的特征（用户轮廓），当用户活动与正常行为有重大偏离时即被认为是入侵
- 检测原理
  - 正常行为的特征轮廓
  - 检查系统的运行情况
  - 是否偏离预设的门限？
- 举例
  - 多次错误登录、午夜登录

# 异常检测技术

- 优点
  - **可以**检测到未知的入侵
  - **可以**检测冒用他人帐号的行为
  - **具有自适**应，自学习功能
  - **不需要系**统先验知识
- **缺点**
  - **漏**报相对低、误报率相对高
  - 统计算法的计算量庞大，效率很低
  - 统计点的选取和参考库的建立比较困难

# 异常检测技术

- **统计分析：**

  - **依据系**统中特征变量的历史数据建立统计模型，并运用该模型对特征变量未来的取值进行预测和检测偏离

  - **典型的系**统主体特征有：系统中的特征变量有用户登录失败**次数、**CPU**和**I/O**利用率、文件**访问数及访问出错率、网络连**接数、**击键频率、事件间的时间间隔等。

  - **缺点**：

    - **未考**虑事件的发生顺序，所以对利用事件顺序关系的攻击难以检测；
    - **利用**统计轮廓的动态自适应性，通过缓慢改变其行为来训练正常特**征**轮廓，最终使检测系统将其异常活动判为正常；
    - 难以确定门限值 。

# 异常检测技术

- **统计分析：**

  - (a)均值与标准偏差模型。以单个特征变量为检测对象，假定特征变量满足正态分布，根据该特征变量的历史数据统计出分布参数(均值、标准偏差)，并依此设定信任区间。在检测过程中，若特征变量的取值超出信任区间，则认为发生异常。

  - (b)多元模型。以多个特征变量为检测对象，分析多个特征变量间的相关性，是均值与标准偏差模型的扩展，不仅能检测到单个特征变量值的偏离，还能检测到特征变量间关系的偏离。

  - (c) Markov过程模型。将每种类型的事件定义为系统的一个状态，用状态转换矩阵来表示状态的变化，若对应于所发生事件的状态转移概率较小，则该事件可能为异常事件。

  - (d) 时间序列模型。将事件计数与资源消耗根据时间排列成序列，如果某一新事件在相应时间发生的概率较低，则该事件可能为入侵。

# 误用检测技术

- **思想**：**主要是通**过某种方式预先定义入侵行为，然后监视系统，从中找出符合预先定义规则的入侵行为
- 误用信号需要对入侵的特征、环境、次序以及完成入侵的事**件相互**间的关系进行描述
- 重要问题
  - **如何全面的描述攻**击的特征
  - **如何排除干**扰，减小误报
  - **解决**问题的方式
- **典型算法**：专家系统、模型推理、完整性分析

# 误用检测技术

- 优点：
  - **算法**简单
  - **系**统开销小
  - **准确率高**
  - 效率高
- **缺点**：
  - **被**动：只能检测出已知攻击 **、新**类型的攻击会对系统造成很大的**威**
  - **模式**库的建立和维护难：模式库要不断更新，知识依赖于硬件平**台、操作系**统和系统中运行的应用程序等

# 误用检测技术

- **专家系统**

  - **将有关入侵的知**识转化为if-then结构的规则，即将构成入侵**所要求的条件**转化为if**部分**，**将**发现入侵后采取的相应**措施**转化成then**部分**。

  - 专家系统主要面临：

    - 1）**全面性**问题，即难以科学地从各种入侵手段中抽象出全面**地**规则化知识；

    - 2）**效率**问题，即所需处理的数据量过大，而且在大型系统上，**如何**获得实时连续的审计数据也是个问题。

# 误用检测技术

- **模型推理:**
  - **模型推理是指**结合攻击脚本推理出入侵行为是否出现。
  - **其中有关攻**击者行为的知识被描述为：攻击者目的，攻击者达到此目的的可能行为步骤，以及对系统的特殊使用等。
  - **根据**这些知识建立攻击脚本库，每一脚本都由一系列攻击行为组成。
  - **减少了需要**处理的数据量，如何有效地翻译攻击脚本是问题

# 检测方法实例一：snort

- **典型的**误用检测技术
- **需要准确理解攻**击模式并撰写检测规则
- **无法**应对未知攻击，造成漏报
- 检测效率高、准确率高
- **攻**击规则更新要求高，好在攻击方式变化不剧烈

**方法：基于攻击特征匹配的原理，准确度高，误报率低，无法检测未知攻击，典型的误用检测技术。**

# 检测方法实例一：snort

## EXAMPLE

| | |
|---|---|
| **Rule Header** | `alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any` |
| **Message** | `msg: "BROWSER-IE Microsoft Internet Explorer CacheSize exploit attempt";` |
| **Flow** | `flow: to_client,established;` |
| **Detection** | `file_data; content:"recordset"; offset:14; depth:9; content:".CacheSize"; distance:0; within:100; pcre:"/CacheSize\s*=\s*/"; byte_test:10,>,0x3ffffffe,0,relative,string;` |
| **Metadata** | `policy max-detect-ips drop, service http;` |
| **References** | `reference:cve,2016-8077;` |
| **Classification** | `classtype: attempted-user;` |
| **Signature ID** | `sid:65535;rev:1;` |

# 检测方法实例一：snort

## DoS检测——Land

- 攻击目的：land 攻击是一种使用相同的源和目的主机发送数据包到某台机器的攻击，结果通常使存在漏洞的机器崩溃。
- 攻击原理：一个特别打造的SYN包中的源地址和目标地址都被设置成某一个服务器地址，这时将导致接受服务器向它自己的地址发送SYN一ACK消息，结果这个地址又发回ACK消息并创建一个空连接，每一个这样的连接都将保留直到超时掉。
- 检测方法：**判断网络数据包的源地址和目标地址是否相同**。

**注：实例规则为示意性规则，sid非snort官方编号。**

# Snort规则实例一

alert ip $HOME_NET any -> $HOME_NET any
(msg:“DoS Land attack”; flags:S; flow：stateless
classtype:attempted-dos; sid:6001; rev:1;sameip)

- 过滤筛选ip数据包，来自本地地址的任意端口，发往本地地址的任意端口，在报警和包日志中打印“DoS Land attack”；数据包中flags字段的值是SYN。
- 规则类别标识是attempted-dos；snort规则id号为6001；rev是用来识别规则修改的次数，为1。
- sameip关键字允许规则检测源IP和目的IP是否相等。

# Snort规则实例二

## 扫描检测——TCP NULL扫描

- 攻击目的：**可以判断目标主机操作系统是windows还是类Unix系统**。
- 攻击原理：根据RFC 793，类Unix系统接收到没有设置任何标志位的数据包，端口关闭的情况下舍弃掉该数据包并且发送一个RST数据包，端口开放的话不会响应；而Windows系统不满足RFC 793，当收到该数据包的情况下，无论端口开放或者关闭，都会响应一个RST数据包给发送方。
- 检测方法：验证数据包中所有标志位是否都为0。

# Snort规则实例二

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:“SCAN NULL”;flags:0; flow : from_client ; classtype:attempted-recon; sid:6002; rev:1;)

- 过滤筛选tcp协议的数据包，来自外部网络的任意端口，发往**本地的任意端口，并且tcp包标志位全部为0，在**报警和包日**志中打印**“SCAN NULL”
- Flow**表示**规则只应用到来自客户端的TCP**数据包**；规则类别标识是attempted-recon（Attempted Information Leak）；
- 规则id为6002，规则修改次数为1。

# 检测方法实例一：snort

## 漏洞利用检测——SQL注入漏洞

- 攻击目的：恶意用户通过构造特殊字符串从而从服务器获得敏感信息。

- 攻击原理：#可以注释掉SQL语句后面的一行SQL代码，相当于去掉了后面的where条件，而且前面的1=1永远都是成立的，即where子句总是为真，例如select * from users where username='' or 1=1 #' and password=md5('')

- 检测方法：验证数据包中的内容中是否存在“’ or 1=1 #”

# Snort规则实例三

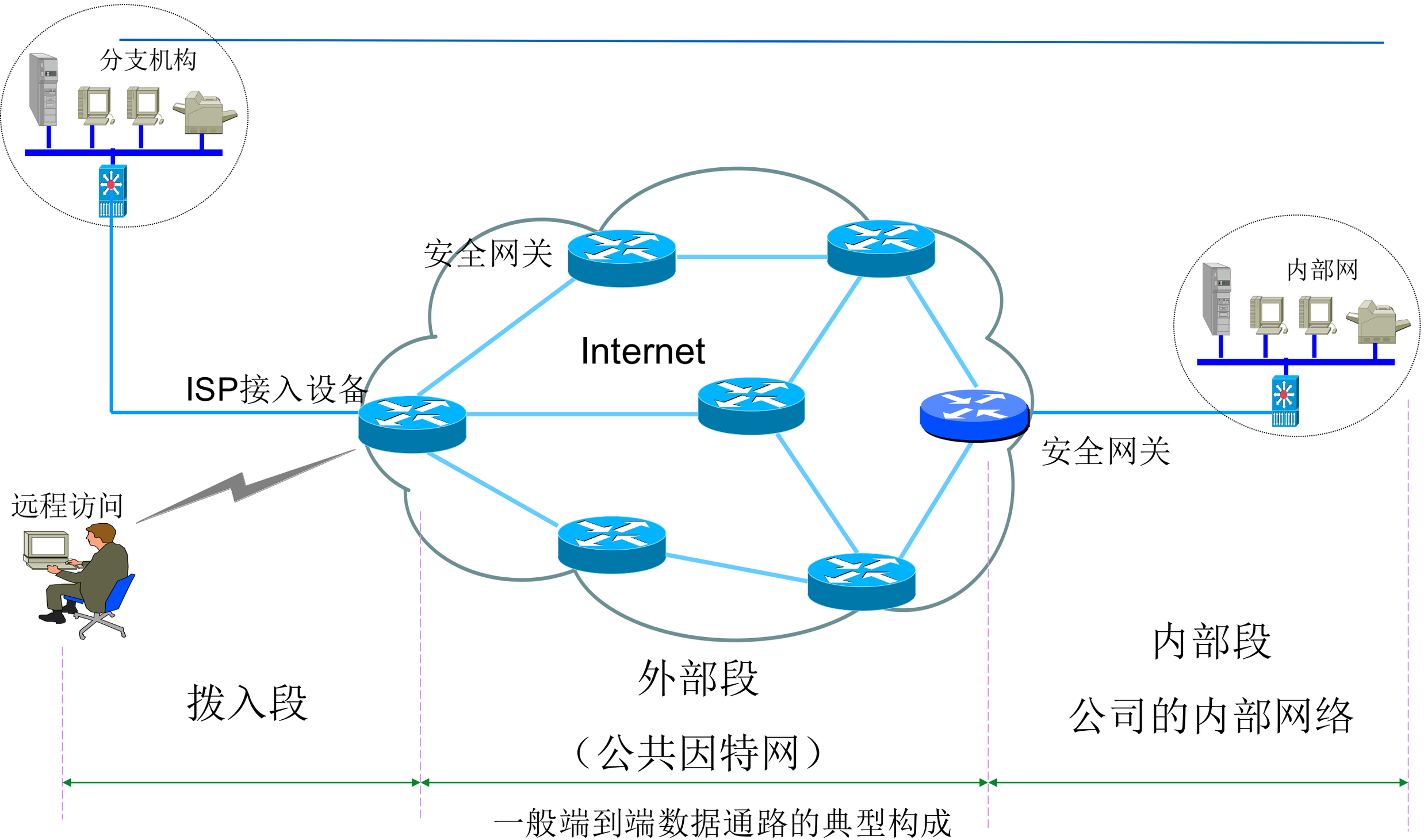
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:“SQL Injection found”; flow:from_client, established; content:“’%20and%201=1#”; classtype:web-application-attack; sid:6003; rev:1;)

- 过滤筛选来自客户端的tcp**数据包**，**来自外部网**络的任意端口，发往本**地的任意端口**，**并且包的**净荷中含有“’%20and%201=1#”，**在**报警和包**日志中打印**“SQL Injection found”
- Flow**表示**规则只应用到已经建立的来自客户端的TCP连接；
- 规则类别标识是web-application-attack；规则id为6003，规则修改次数为1

网络安全——

# VPN

# 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

# VPN要解决的问题

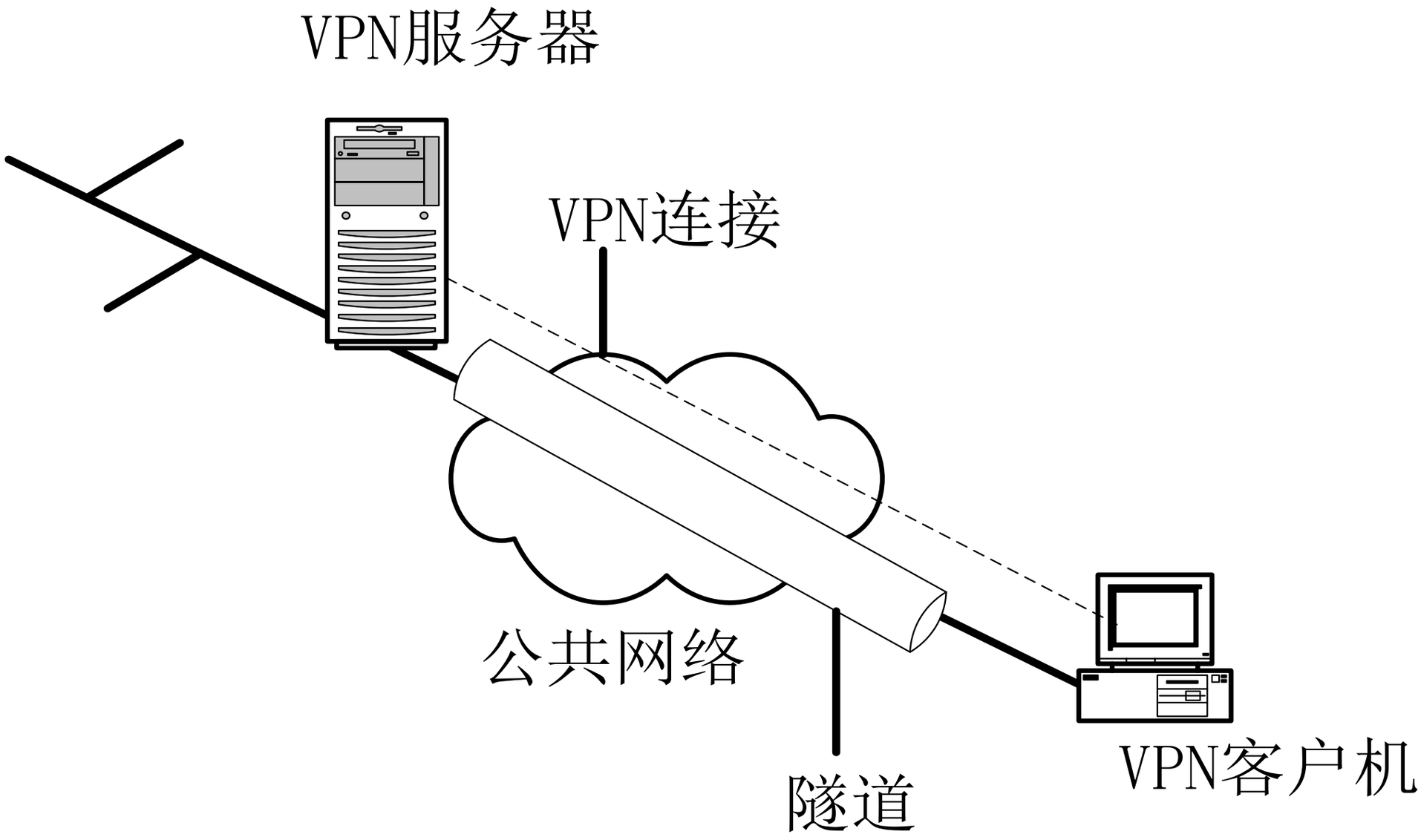
- 在端到端的数据通路上随处都有可能发生数据的泄漏，包括
  - 拨入段链路上
  - ISP接入设备上
  - 在因特网上
  - 在安全网关上
  - 在企业内部网上。

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

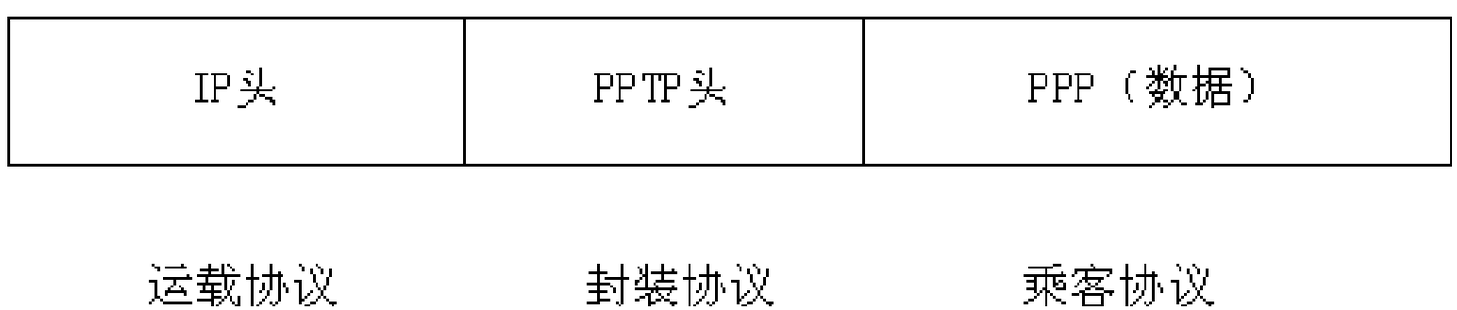
- 专用网的特点：
  - 封闭的用户群
  - 安全性高
  - 服务质量保证
- VPN的实现要求
  - 支持数据分组的透明传输
  - 支持安全功能
  - 提供服务质量保证

# 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

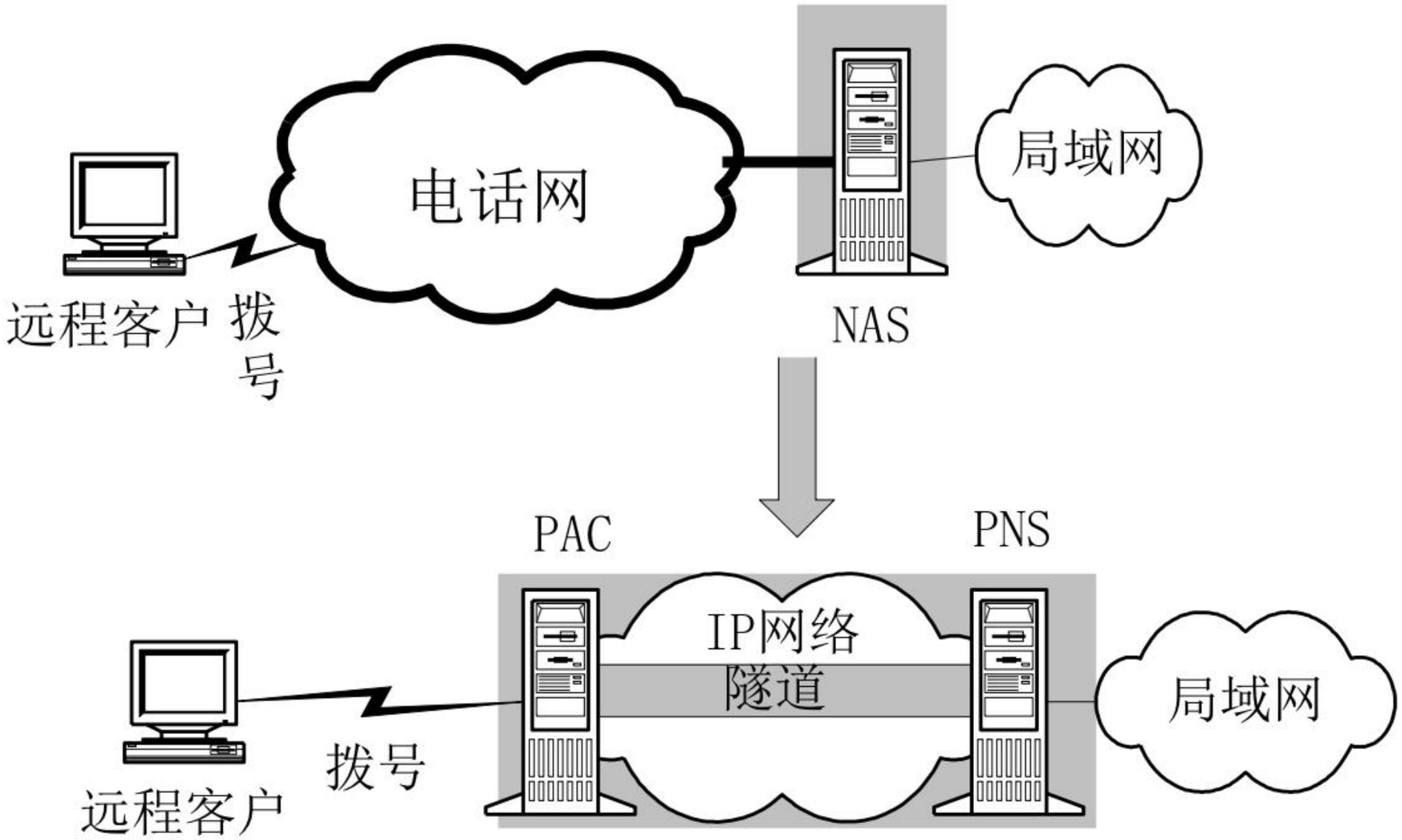
- **数据机密性保护**
- **数据完整性保护**
- **数据源身份认证**
- **重放攻击保护**

# 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。

# 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。

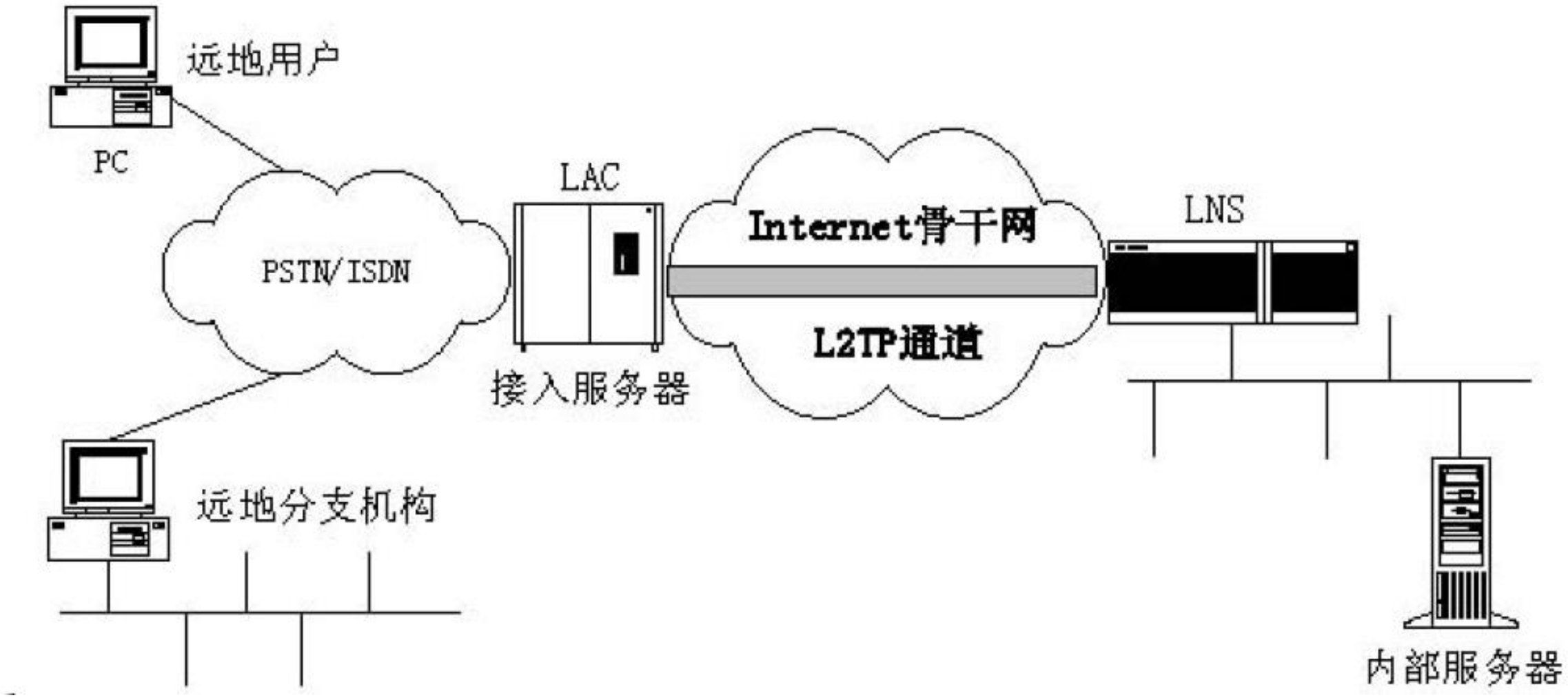
  - L2TP主要由LAC(L2TP Access Concentrator)和LNS(L2TPNetworkServer)构成，LAC(L2TP访问集中器)支持客户端的L2TP，他用于发起呼叫，接收呼叫和建立隧道；LNS(L2TP网络服务器)是所有隧道的终点。在传统的PPP连接中，用户拨号连接的终点是LAC，L2TP使得PPP协议的终点延伸到LNS。

  - 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。

# 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体系结构

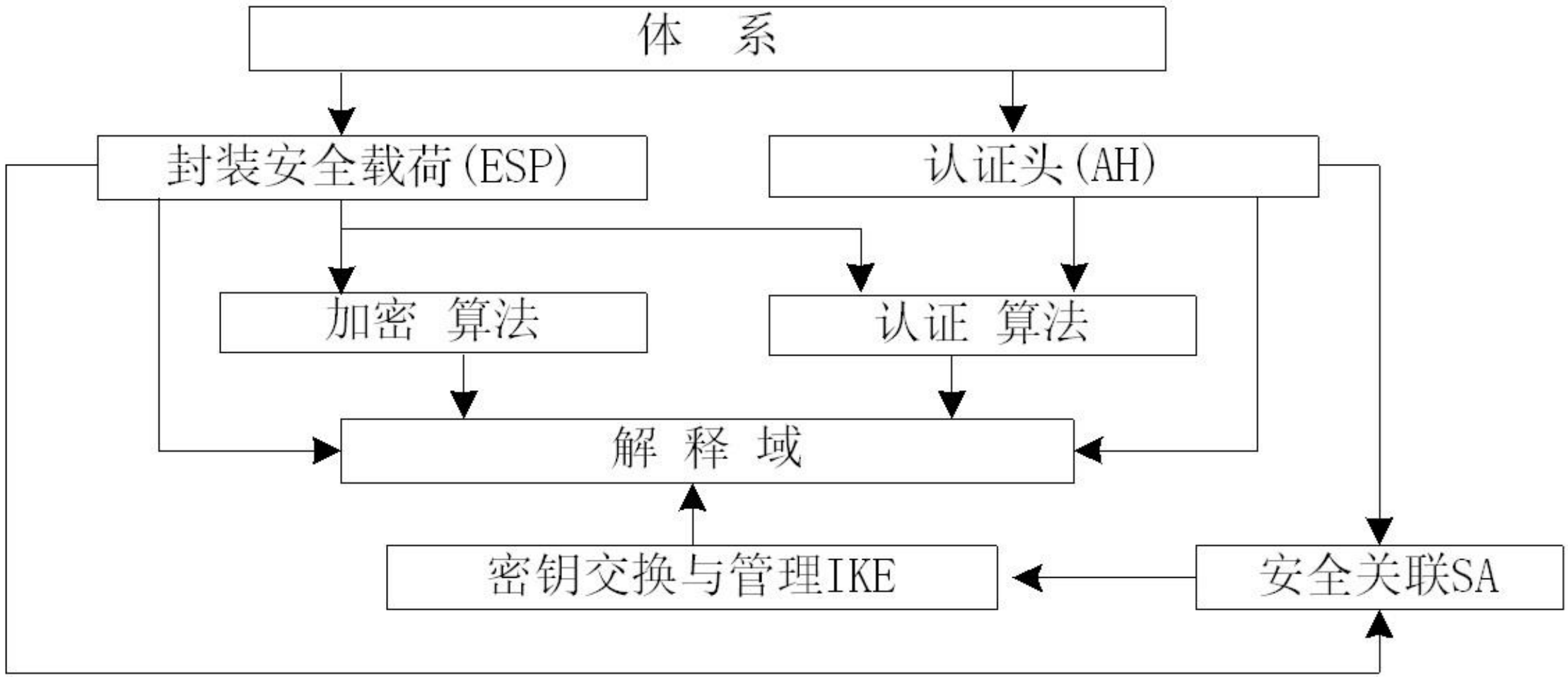



图 IPSec安全体系结构

# AH协议

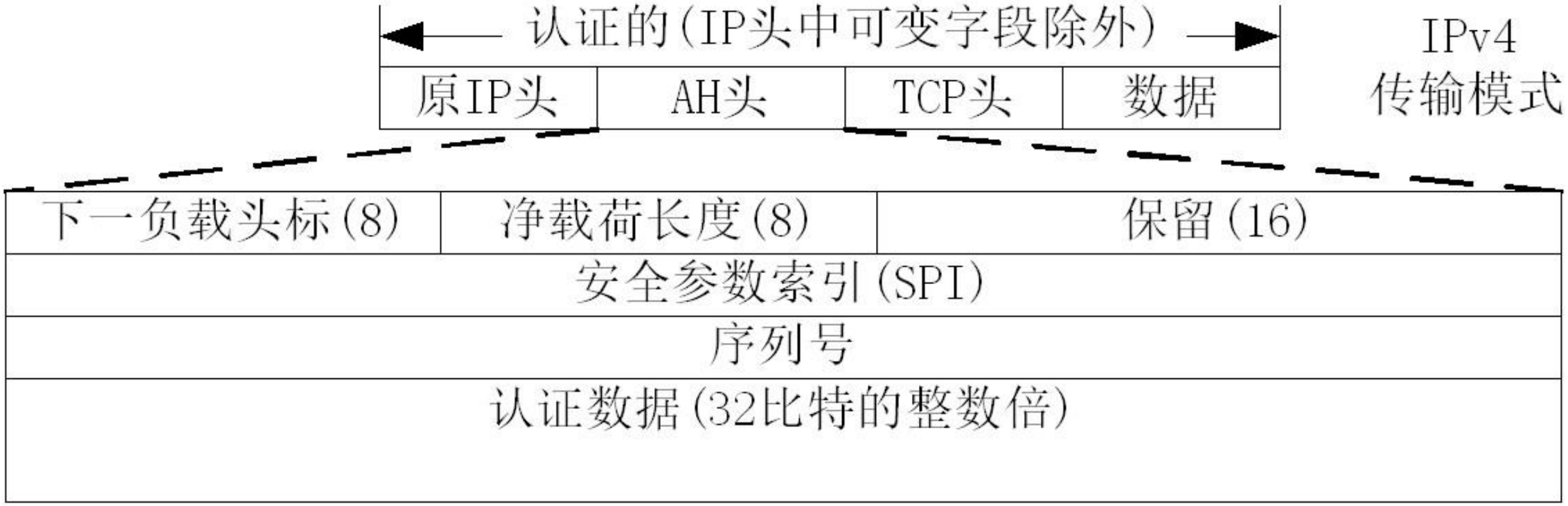
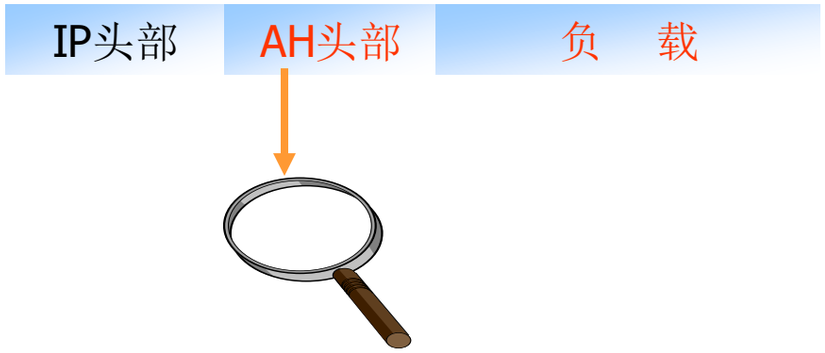
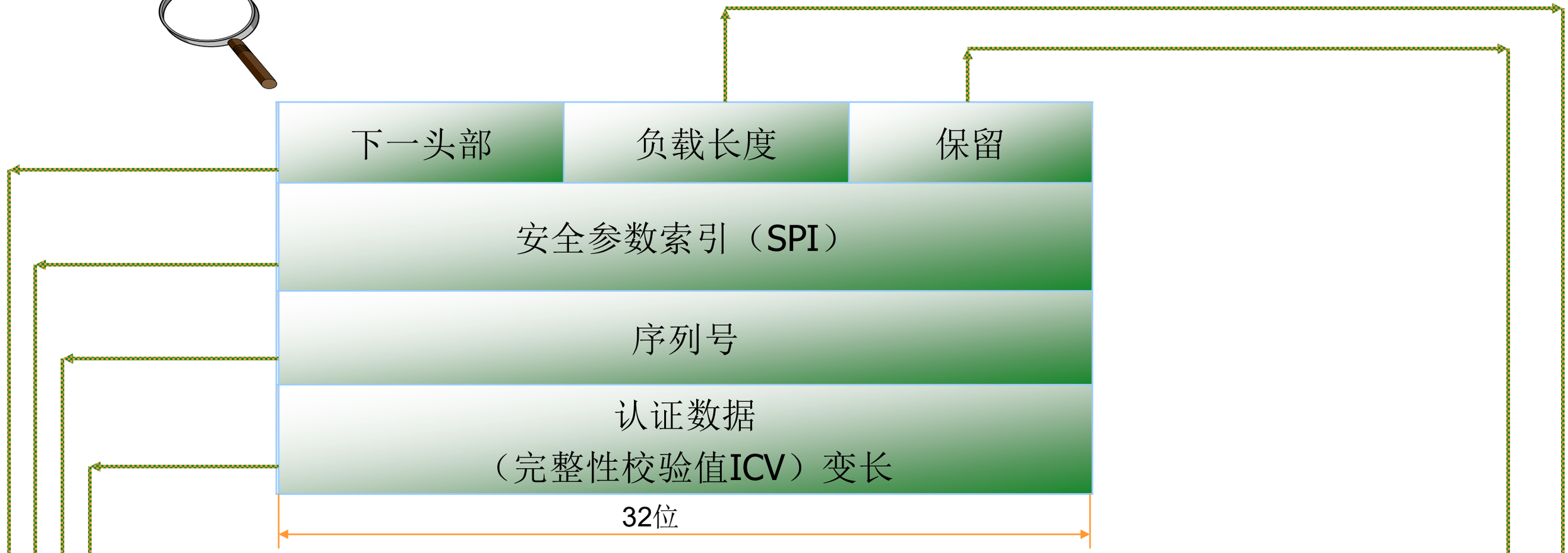

认证的(IP头中可变字段除外)
原IP头
AH头
TCP头
数据
IPv4
传输模式
下一负载头标(8)
净载荷长度(8)
保留(16)
安全参数索引(SPI)
序列号
认证数据(32比特的整数倍)

图 AH的格式

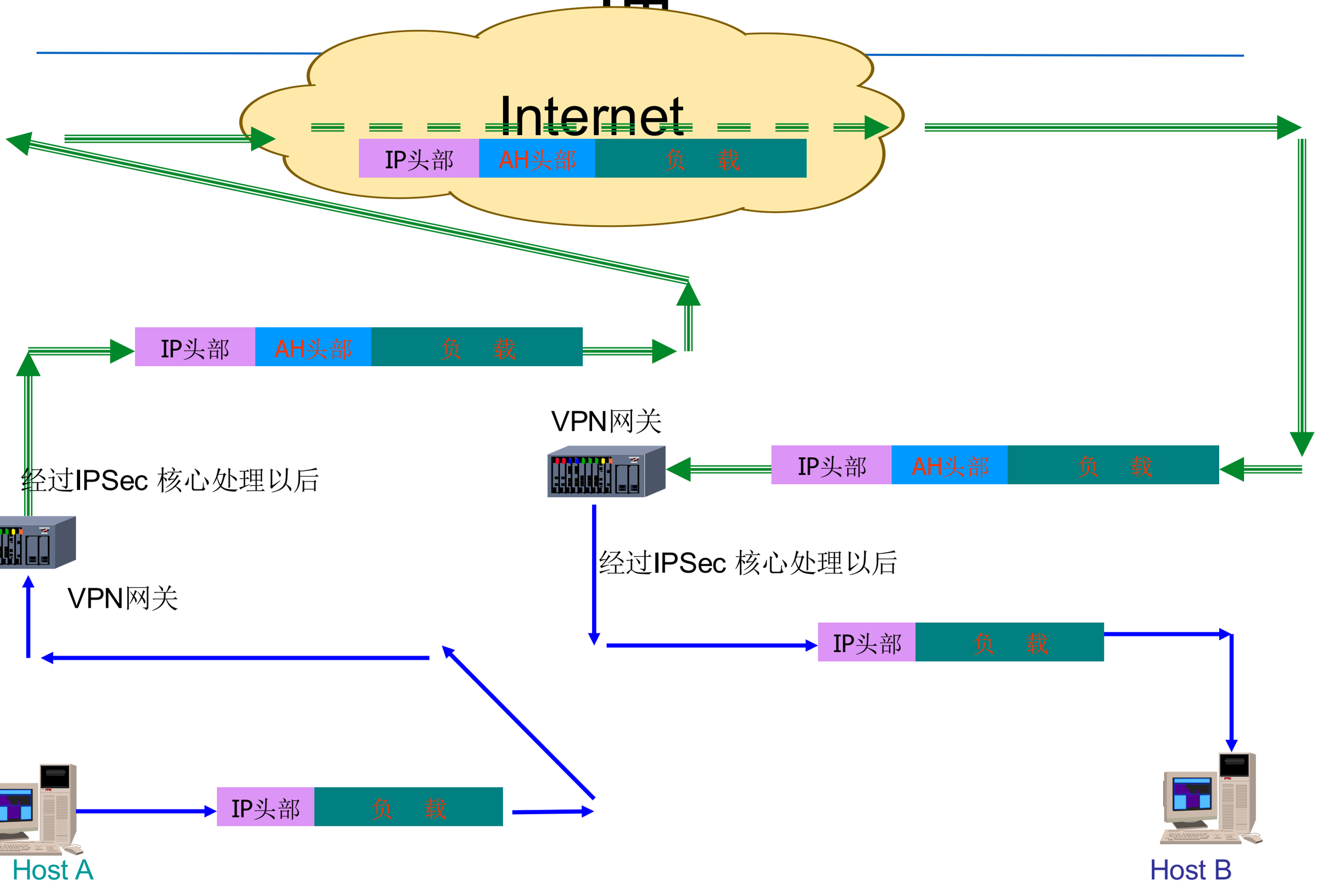
# 认证头部(AH)



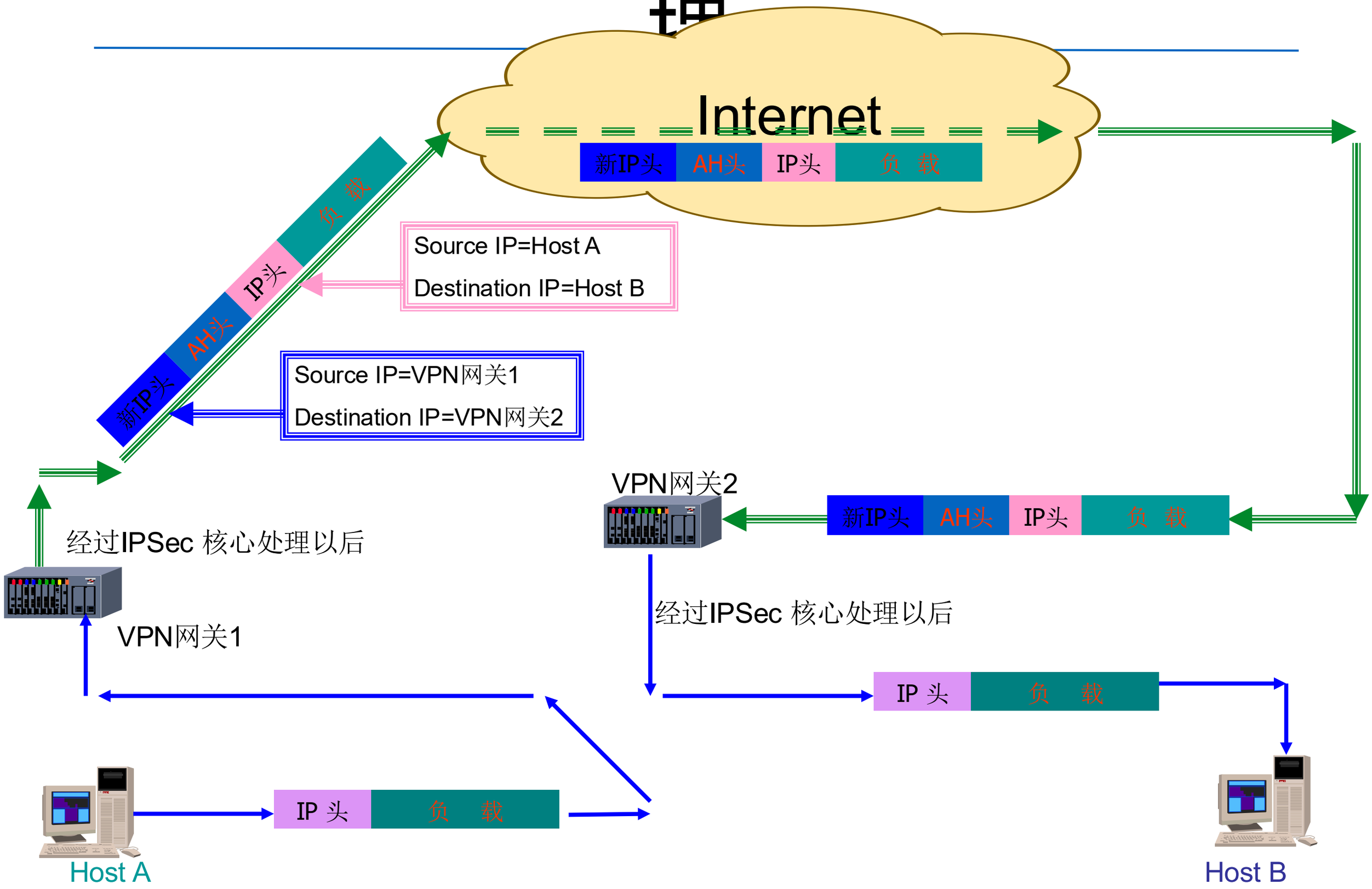




- ❖认证数据：一个变长字段，也叫Integrity Check Value，由SA初始化时指定的算法来计算。长度=整数倍32位比特
- ❖序列号：32比特，一个单项递增的计数器，用于防止重放攻击，SA建立之初初始化为0，序列号不允许重复
- ❖SPI：32比特，用于标识有相同IP地址和相同安全协议的不同SA。由SA的创建者定义，只有逻辑意义
- ❖ 下一头部：8比特，标识认证头后面的下一个负载类型
- ❖保留字段：16比特，保留将来使用，Default=0
- ❖ 负载长度：8比特，表示以32比特为单位的AH头部长度减2，Default=4

# 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

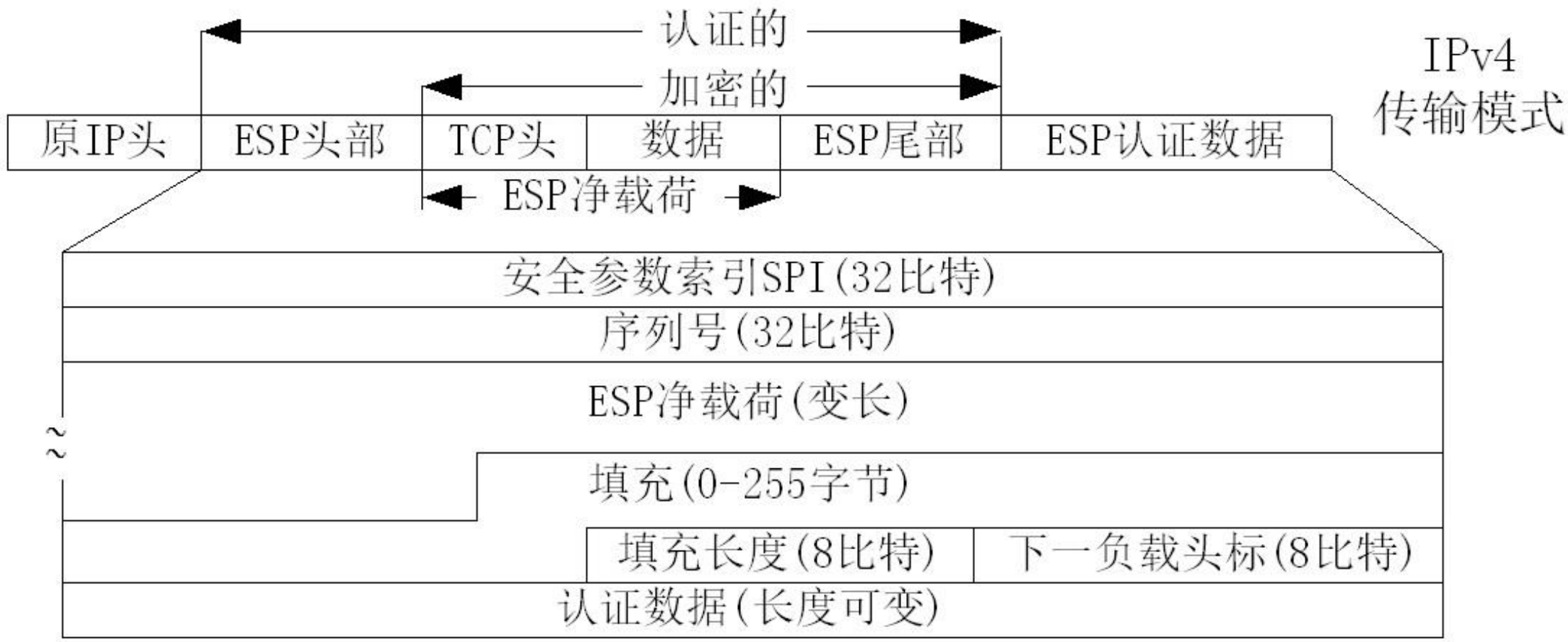
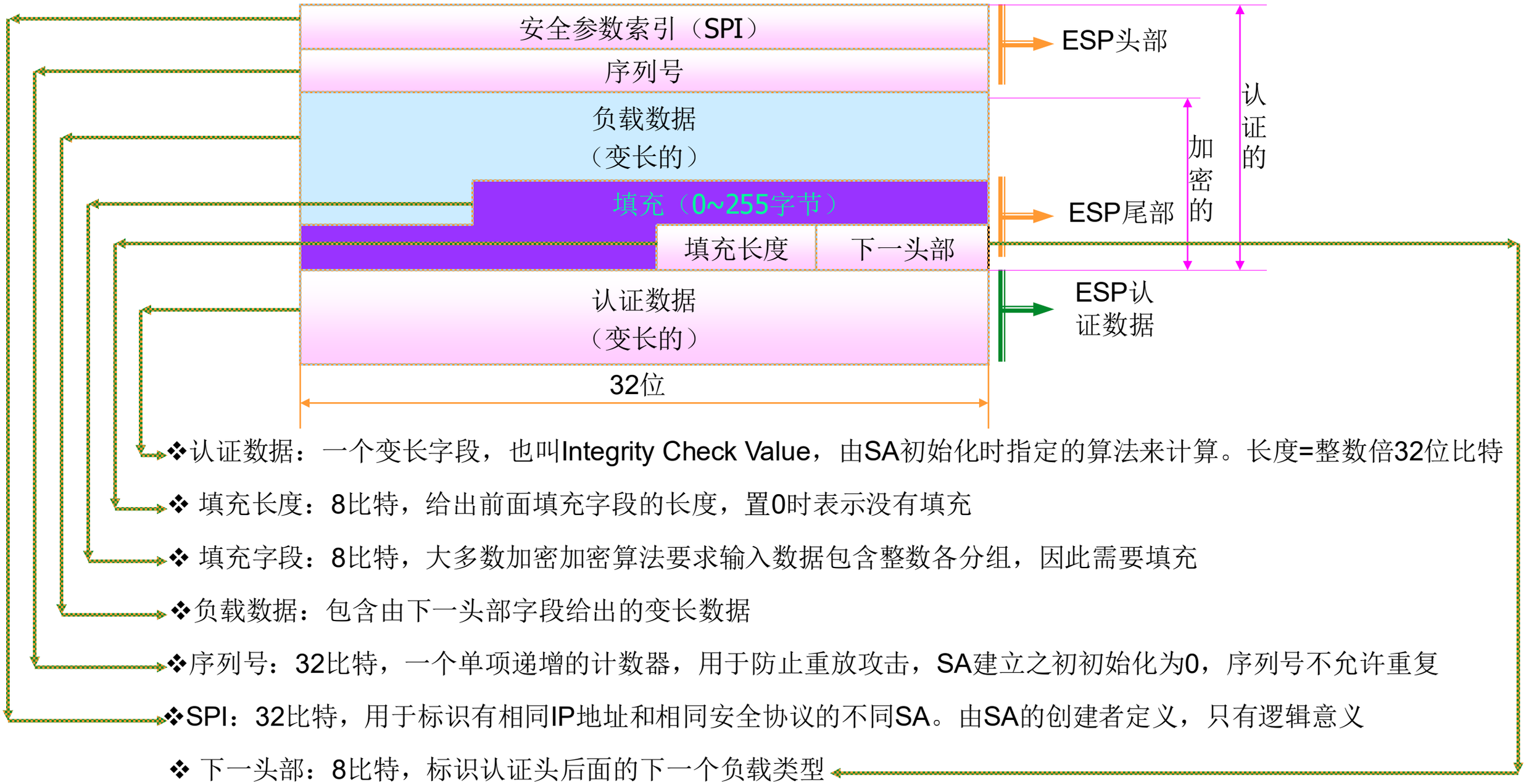


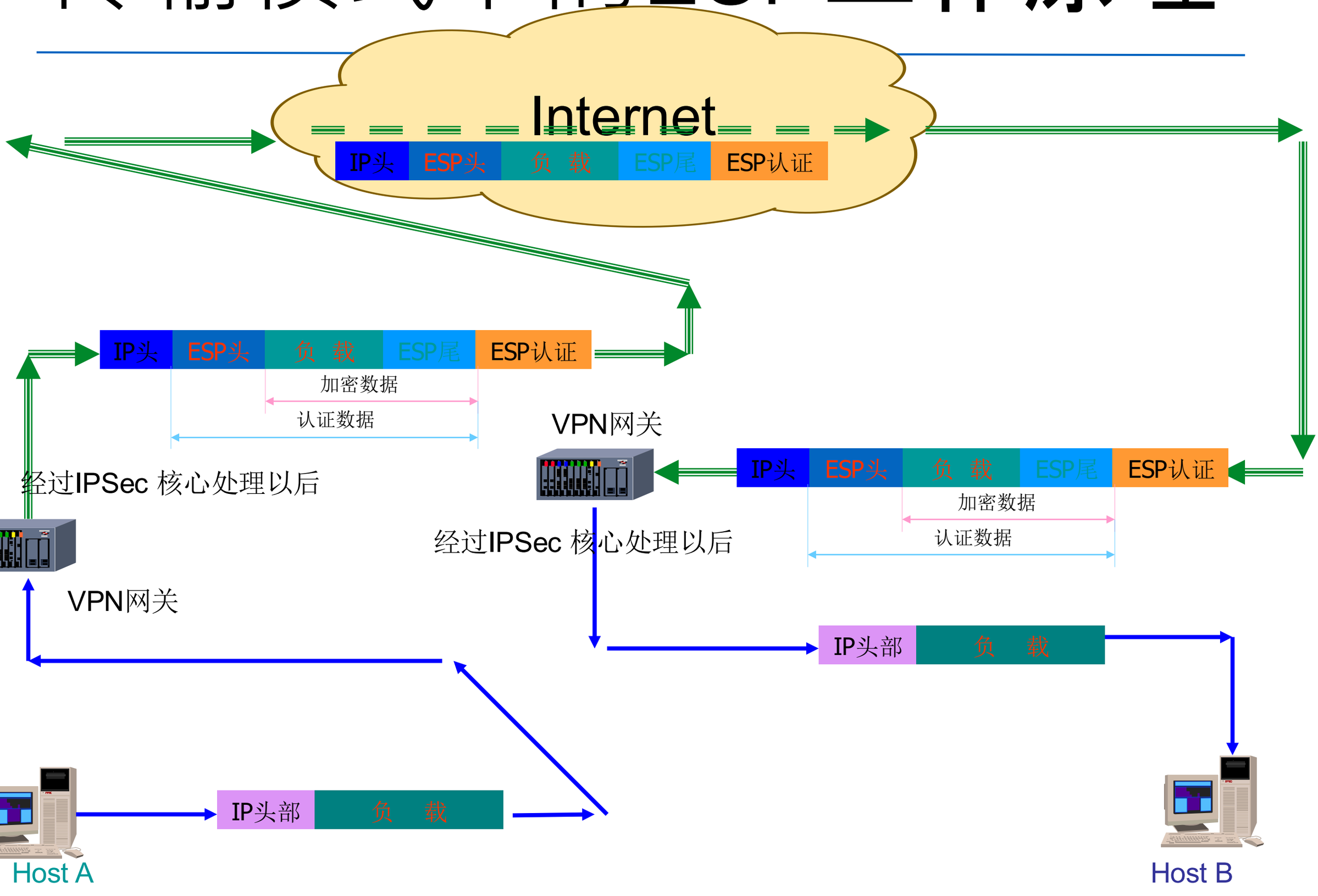

图 ESP格式

# 负载安全封装(ESP)

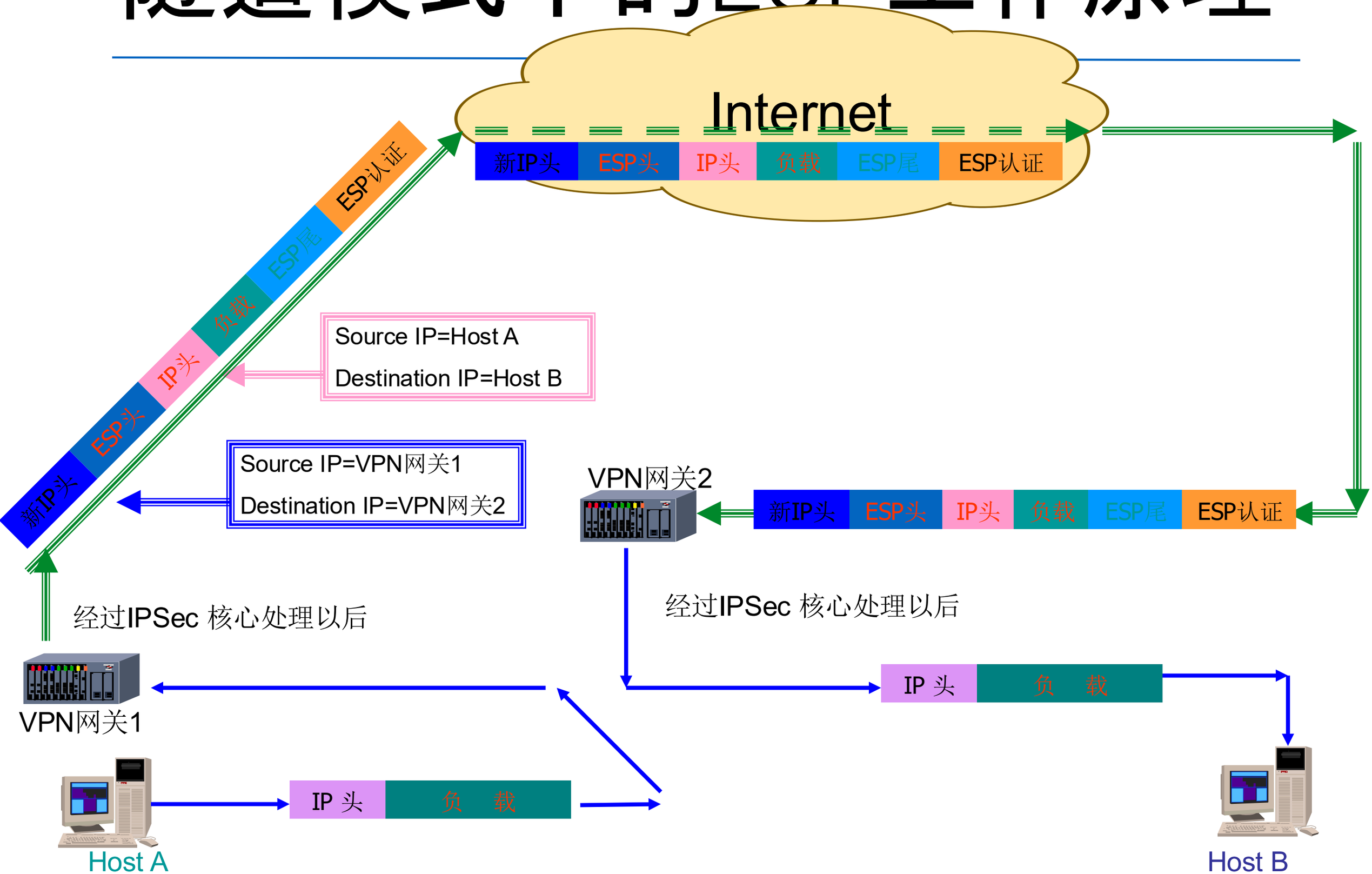




- ❖认证数据：一个变长字段，也叫Integrity Check Value，由SA初始化时指定的算法来计算。长度=整数倍32位比特
- ❖ 填充长度：8比特，给出前面填充字段的长度，置0时表示没有填充
- ❖ 填充字段：8比特，大多数加密加密算法要求输入数据包含整数各分组，因此需要填充
- ❖负载数据：包含由下一头部字段给出的变长数据
- ❖序列号：32比特，一个单项递增的计数器，用于防止重放攻击，SA建立之初初始化为0，序列号不允许重复
- ❖SPI：32比特，用于标识有相同IP地址和相同安全协议的不同SA。由SA的创建者定义，只有逻辑意义
- ❖ 下一头部：8比特，标识认证头后面的下一个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

# 传输模式下的ESP工作原理



# 隧道模式下的ESP工作原理



# IPSec传输模式

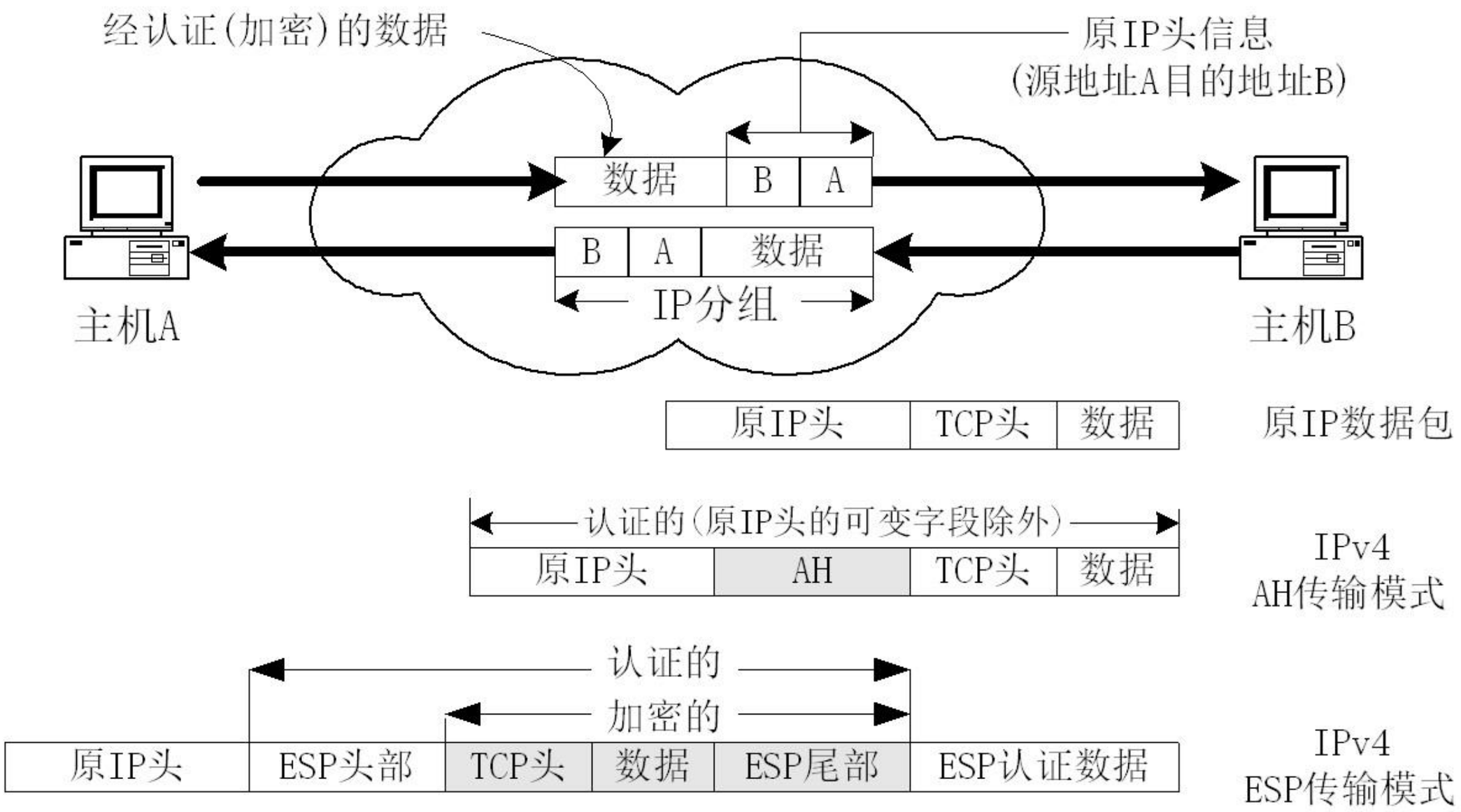



图 IPSec传输模式下的AH、ESP数据封装格式

传输模式适合于保护主机之间连接。

# IPSec隧道模式

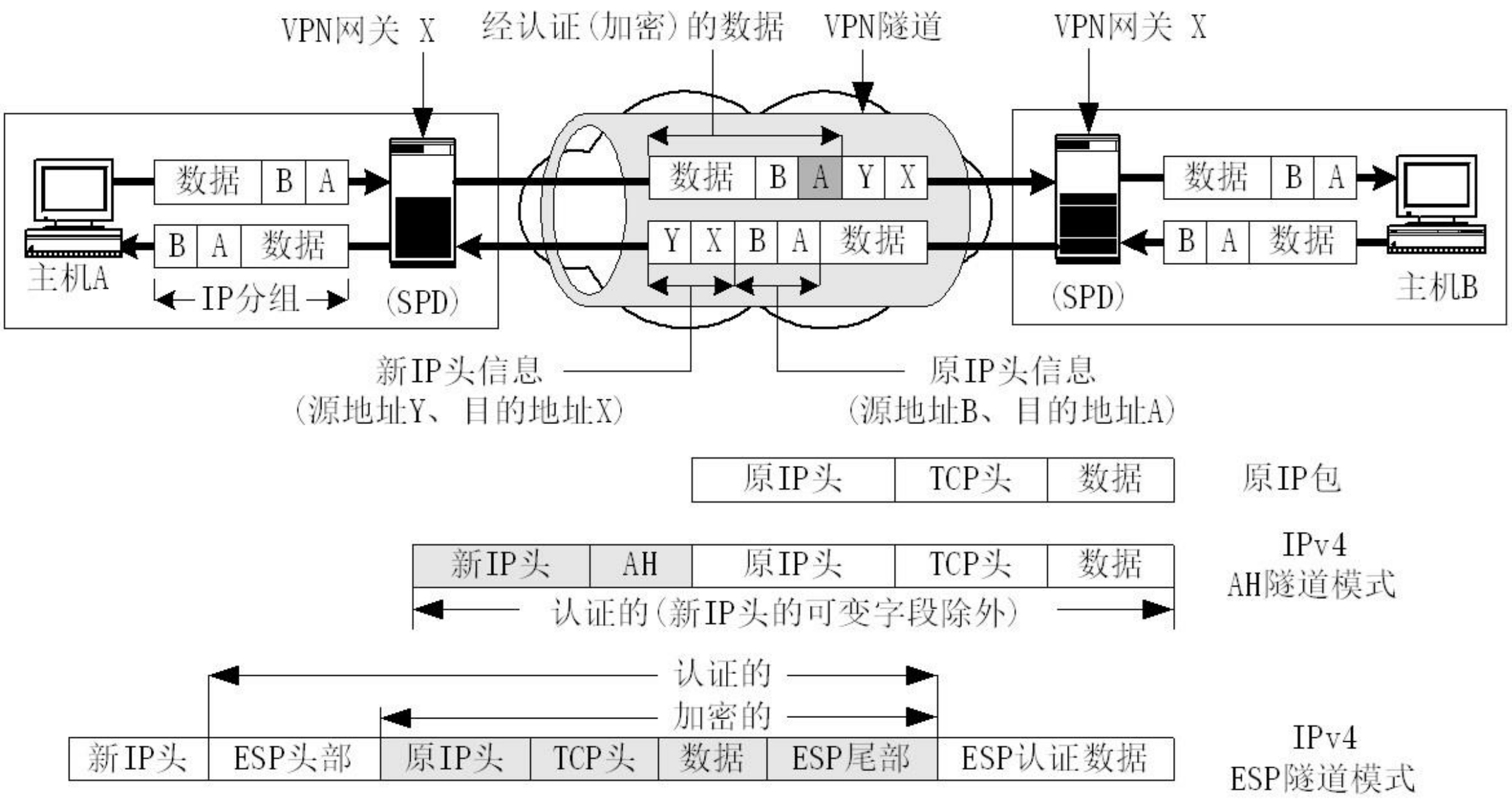



图 IPSec隧道模式下的AH、ESP的数据封装格式

隧道模式适合于保护网络之间连接。

# 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。

# 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<dst_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。

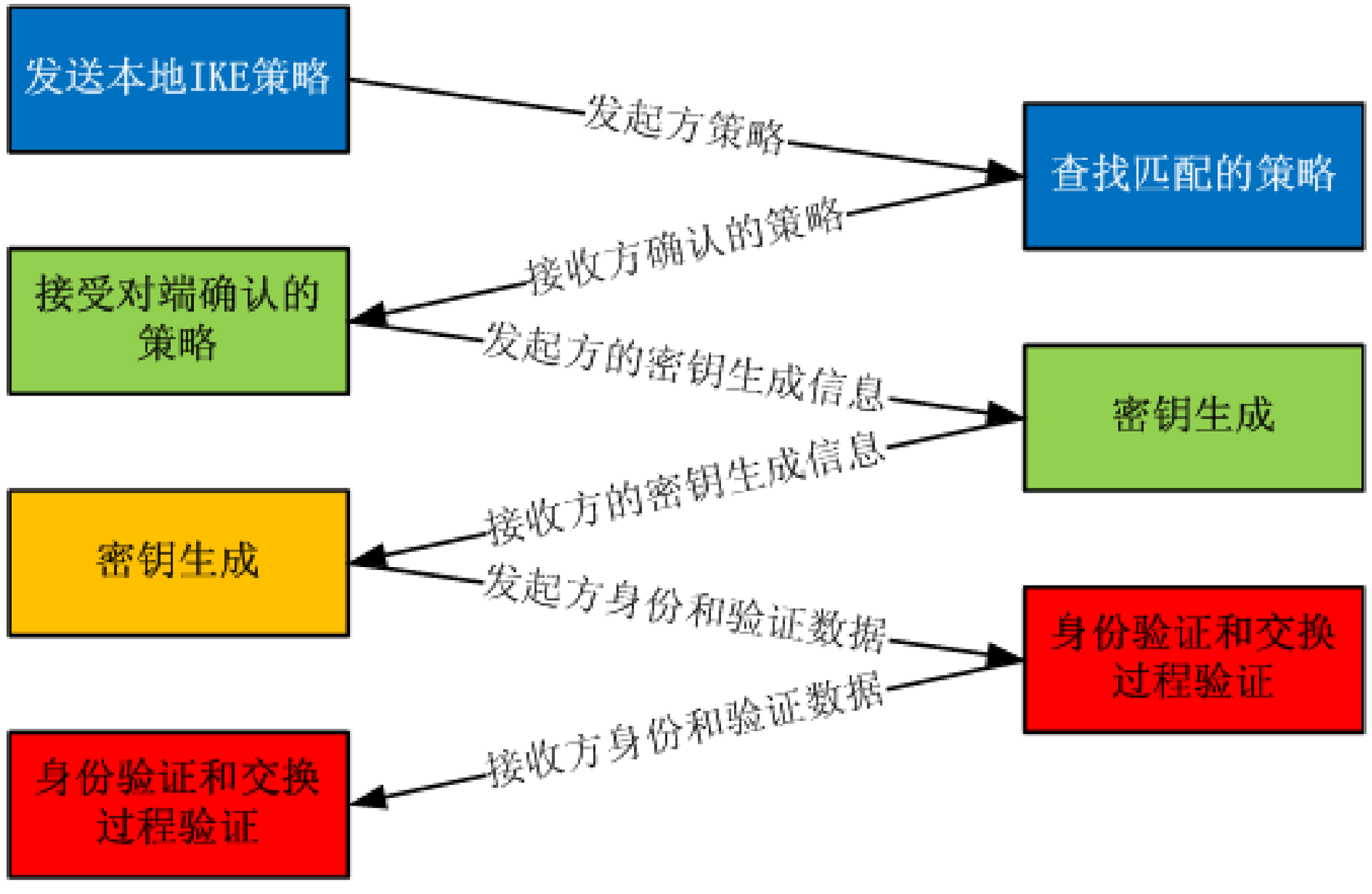
- 2．野蛮模式交换
  - 野蛮模式交换也分为三个步骤，但只交换三条消息：头两条消息协商策略，交换Diffie Hellman公开值必需的辅助数据以及身份信息；第二条消息认证响应方；第三条消息认证发起方，并为发起方提供在场的证据

# IKE阶段一协商流程简图

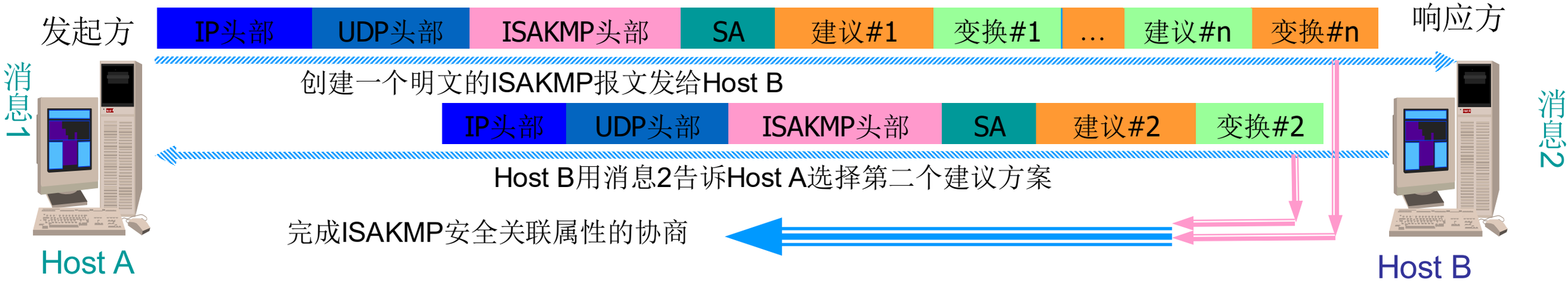
SA交换

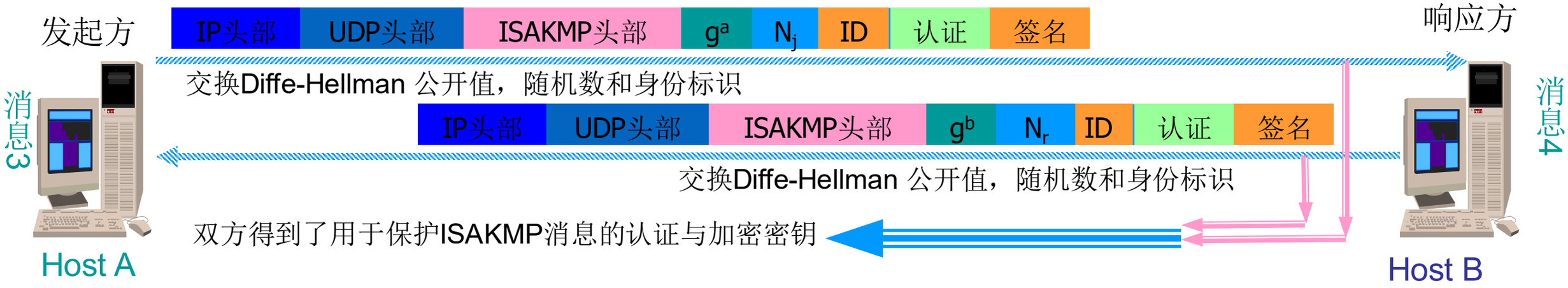
密钥交换

ID及验证交换



# ISAKMP/Oakley 阶段一工作原理



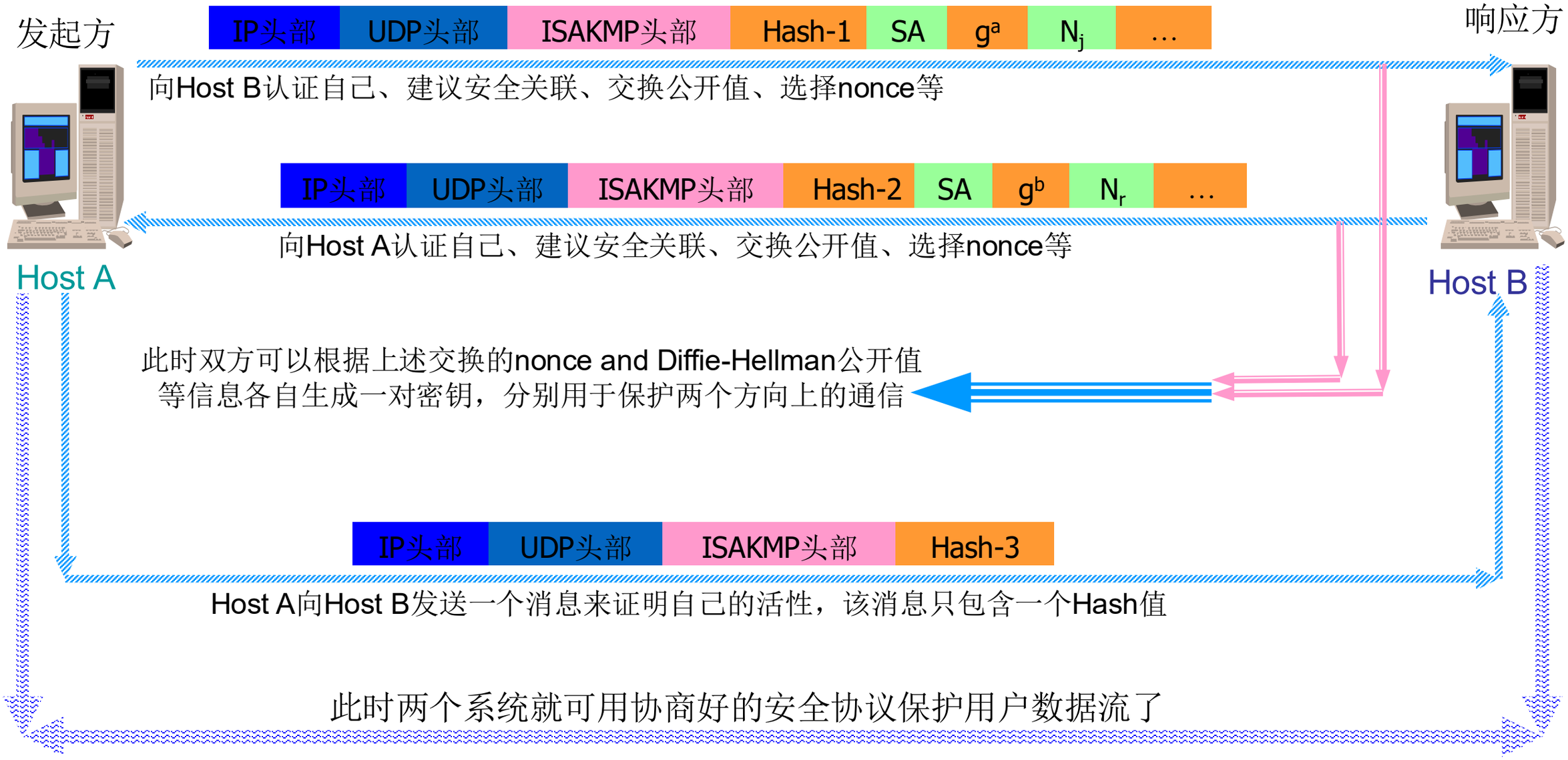

# 交换流程（3）
# （阶段二快速模式）

| Initiator | Director | R 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

- 阶段二分为三个步骤，交换三条消息。
- 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 阶段二工作原理



# 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)

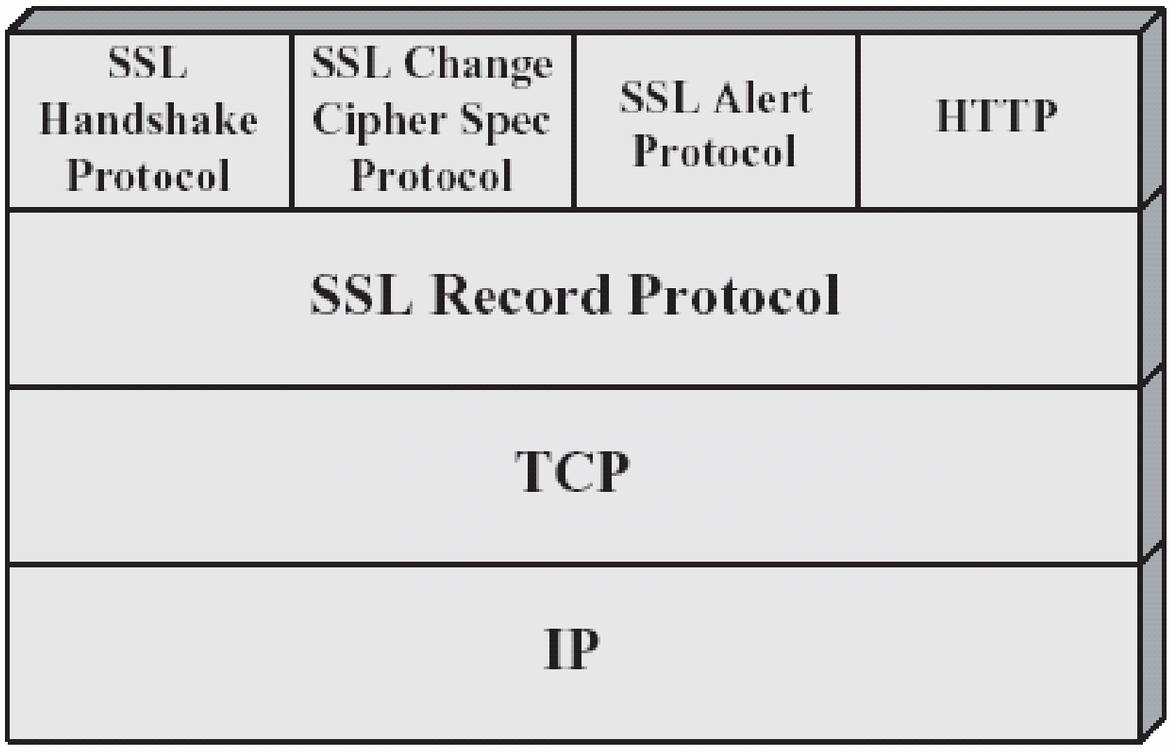
# SSL基本情况

- 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

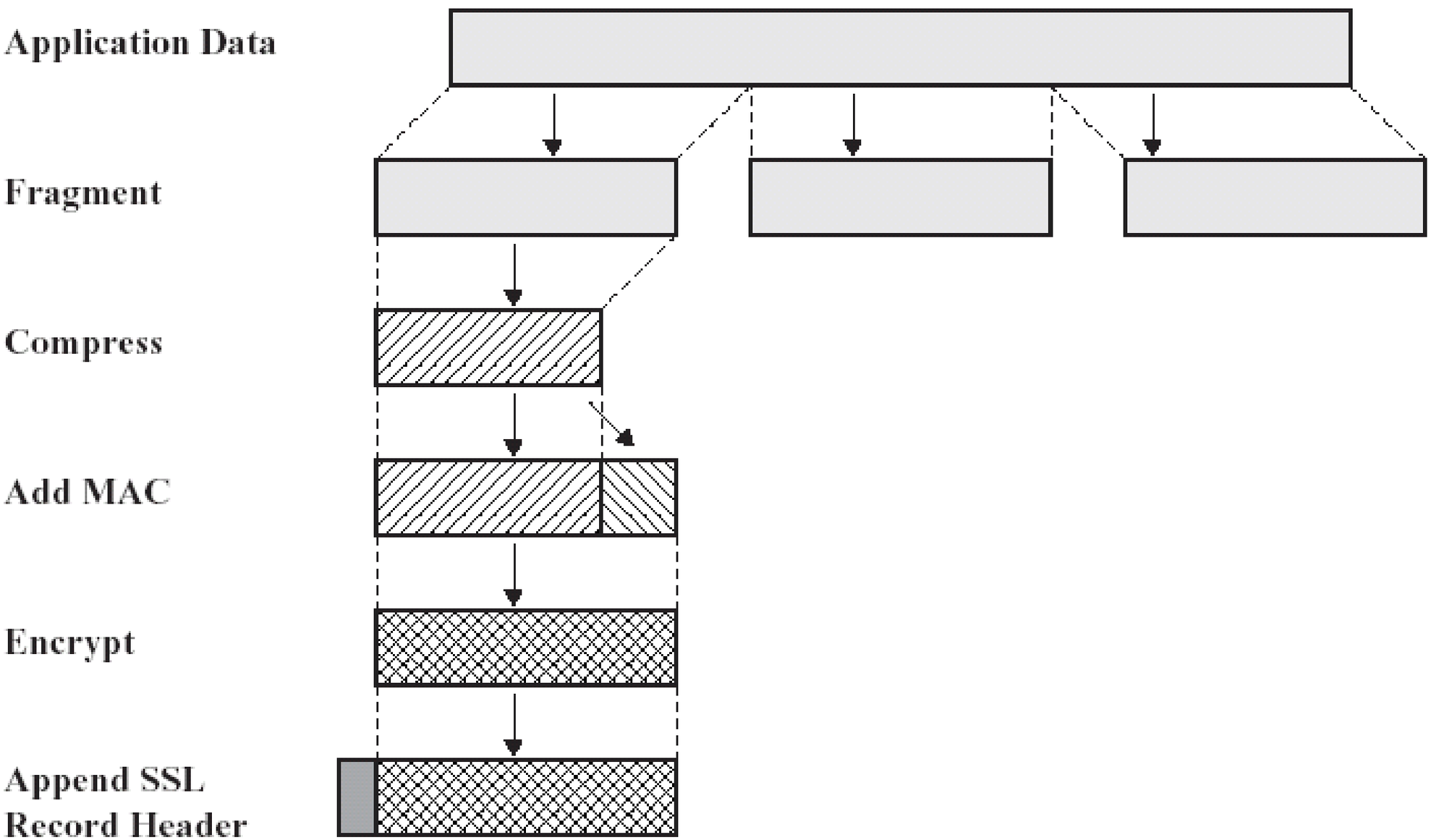
- SSL记录协议
  - **建立在可靠的**传输协议(**如**TCP)**之上**，为更高层提供基本安全服务。特别**是**HTTP，**它提供了**Web**的**client/server**交互的**传输服务，可以构造在SSL**之上**。
  - **它提供**连接安全性，有两个特点
    - **保密性，使用了**对称加密算法
    - **完整性，使用**HMAC**算法**
  - **用来封装高**层的协议
- SSL Handshake Protocol, SSLChange Cipher Spec Protocol, SSL Alert Protocol**是**SSL**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

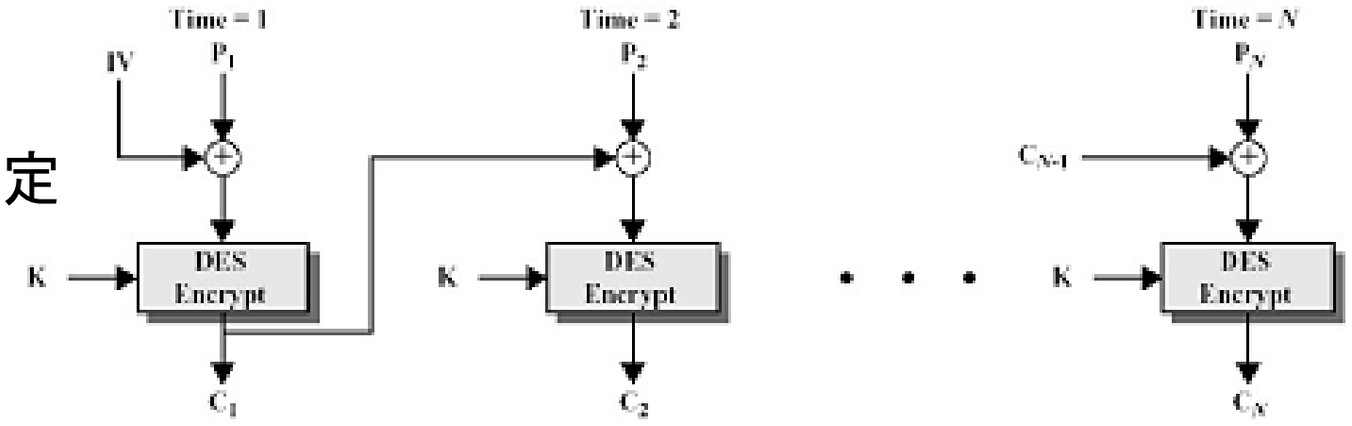
# SSL记录层协议

- 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

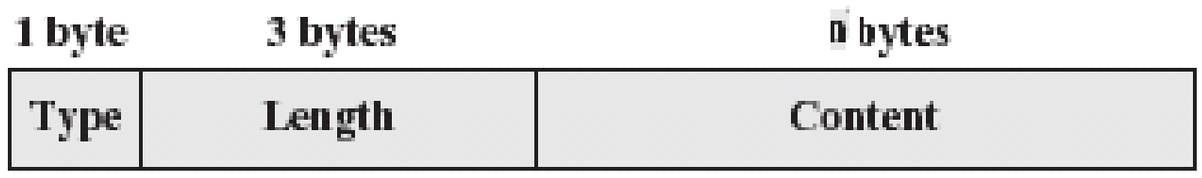
- **握手**协议（见后面）
- **密**码变化协议(Change Cipher Spec Protocol)
  - **它位于**TLS记录协议之上,**它用到了**TLS记录协议的处理过程
  - ContentType = 20
  - 协议只包含一条消息，一个字节 1
  - **用途**：**切**换状态；把密码参数设置为当前状态；在握手协议**中**，**当安全参数**；协商一致后，发送此消息
  - 这条消息使得接收方改变当前状态读参数，使得发送方改变**当前状**态写参数

# 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

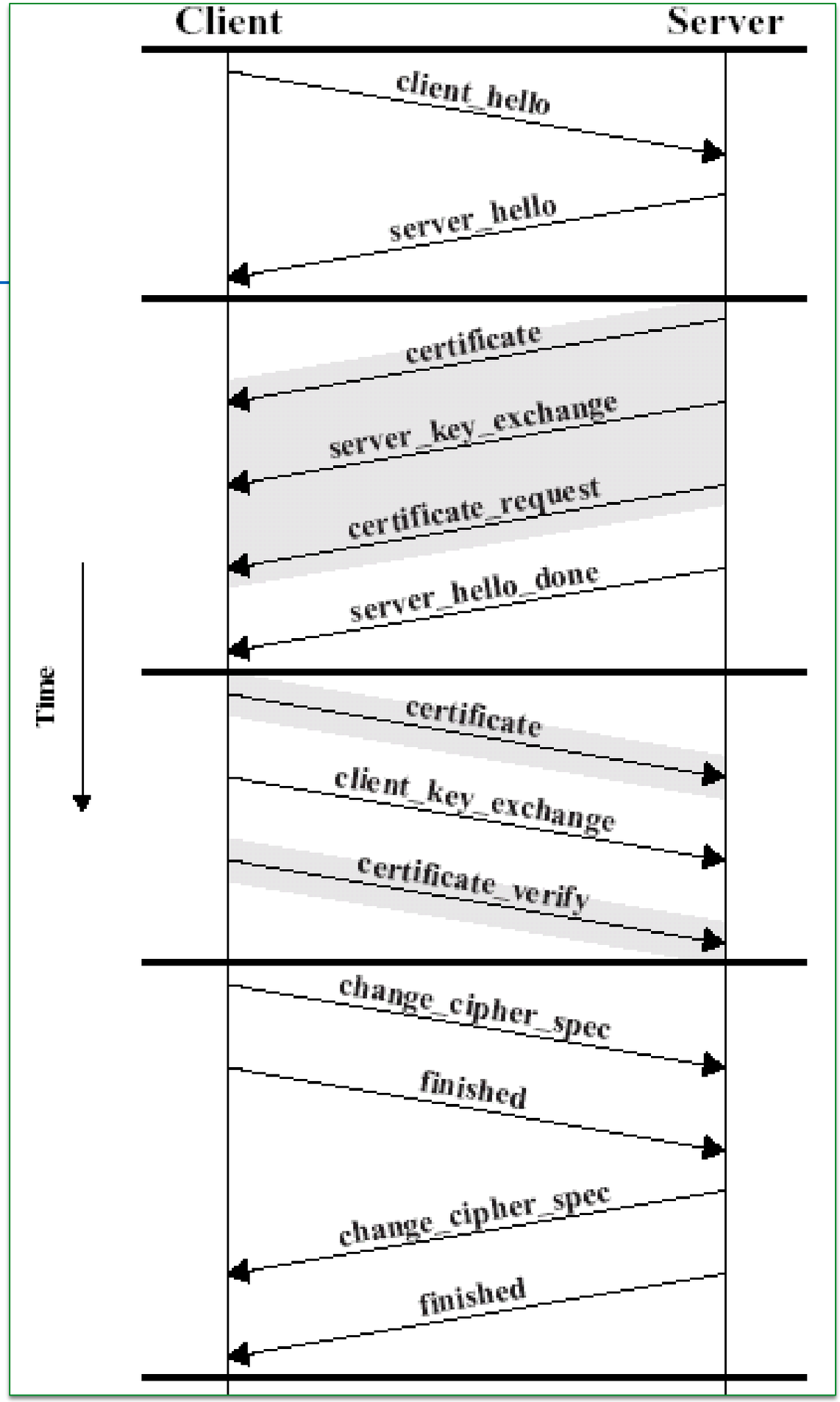
- **功能**
  - **客**户和服务器之间相互认证
  - 协商加密算法和密钥
  - **它提供**连接安全性，有三个特点
    - **身份**认证，至少对一方实现认证，也可以是双向认证
    - 协商得到的共享密钥是安全的，中间人不能够知道
    - 协商过程是可靠的
- 规范说明
  - **位于**TLS记录协议之上，也用到了TLS记录协议的处理过程
  - ContentType = 22
  - 协议格式如图：




**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**



网络安全——

# PGP

# 安全电子邮件系统
# PGP (Pretty Good Privacy)

- 由个人发展起来——
  - Phil Zimmermann(齐默尔曼)
- PGP为电子邮件和文件存储应用提供了认证和保密性服务
  - 选择理想的密码算法
  - 把算法很好地集成到通用应用中，独立于操作系统和微处理器
  - 自由发放，包括文档、源代码等
  - 与商业公司(Network Associates)合作，提供一个全面兼容的、低价位的商业版本PGP
- 历史
  - 1991年推出1.0版，1994年推出2.6版，现在9.6版等
  - 算法的专利之争。困扰了3年多
  - 与美国出口管理限制之争，长达5年的调查

# PGP(Pretty Good Privacy)

- Philip R. Zimmerman的主要工作
  - 选择了最好的加密算法作为基础构件
  - **集成加密算法，形成通用的**应用程序
  - **制作**软件包和文档，包括源码，免费提供
  - **提供完全兼容的低价格的商用版本**
- PGP快速发展和流行的原因
  - **免**费获得，运行不同平台的多个版本
  - **建立在普遍**认为非常安全的算法的基础上，**算法的安全性已**经得到**了充分的**论证，如公钥加密包括RSA、DSS、Diffie-Hellman，单钥**加密包括**CAST-128、IDEA、3DES、AES，**以及**SHA-1**散列算法**
  - 应用范围广泛，**适用性**强
  - **不受任何**组织和政府控制

# PGP功能列表

| 服　务 | 采用算法 | 说　明 |
| --- | --- | --- |
| 数字签名 | DSS/SHA或RSA/SHA | 用SHA-1创建散列码，用发送者的私钥和DSS或RSA加密消息摘要 |
| 消息加密 | CAST或IDEA或3DES、AES及RSA或D-F | 消息用一次性会话密钥加密，会话密钥用接收方的公钥加密 |
| 压缩 | ZIP | 消息用ZIP算法压缩 |
| 邮件兼容性 | Radix 64 | 邮件应用完全透明，加密后的消息用Radix 64转换 |
| 数据分段 | | 为了适应邮件的大小限制，PGP支持分段和重组 |

# PGP所使用的符号

Ks：常规加密中的会话密钥
KRa：公开密钥系统中用户A的私有密钥
KUa：公开密钥系统中用户A的公开密钥
EP：公开密钥加密
DP：公开密钥解密
EC：常规加密
DC：常规解密
H:　散列函数
||：　串接操作(并置)
Z：　使用ZIP算法进行压缩
R64: 基数为64的ASCII格式转换

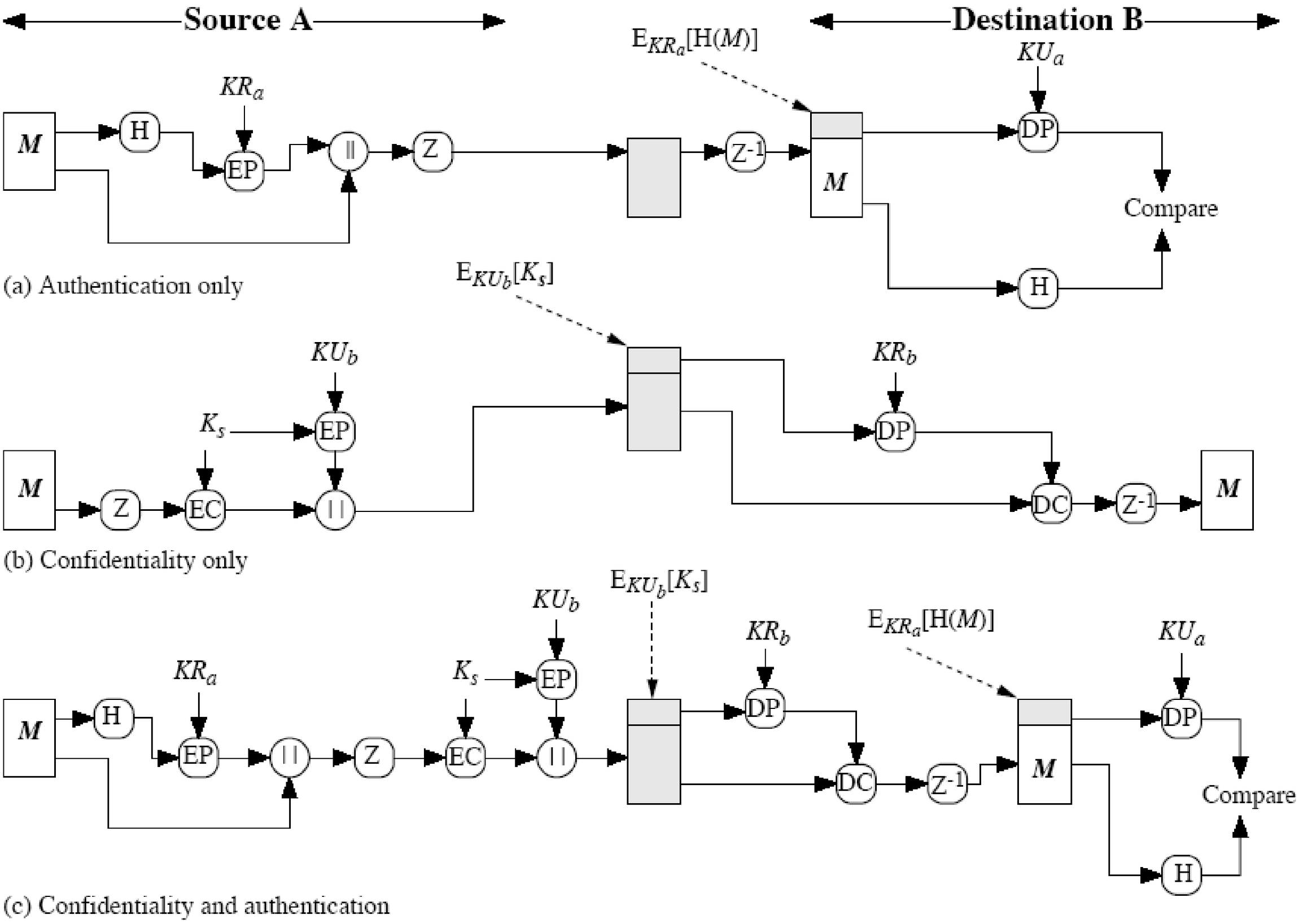



**Figure 15.1 PGP Cryptographic Functions**

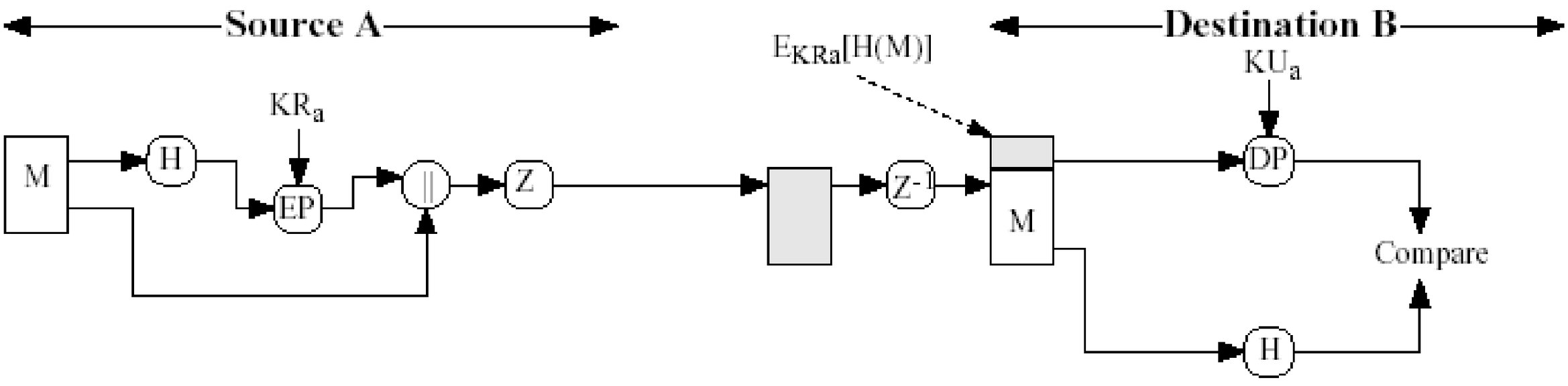
# PGP操作描述

- 数字签名与认证
  - 发送者创建报文
  - 使用SHA-1生成报文的160位散列码
  - 使用发送者的私有密钥，用RSA算法对散列码加密(签名)，并置在报文前面
  - 接收者使用发送者的公开密钥，用RSA解密和恢复散列码
  - 接收者计算报文的散列码，与解密得到的进行比较，如果两者匹配，则报文通过鉴别
- 签名也可以使用DSS/SHA-1来生成
- PGP也支持分离的数字签名

# 功能：身份认证

- 发送方
  - 产生消息M
  - 用SHA-1对M生成一个160位的散列码H
  - 用发送者的私钥对H加密，并与M连接

- 接收方
  - 用发送者的公钥解密并恢复散列码H
  - 对消息M生成一个新的散列码，与H比较。如果一致，则消息M被认证。



# 身份认证说明

- 说明

  1. RSA的强度保证了发送方的身份
  2. SHA-1的强度保证了签名的有效性
  3. DSS/SHA-1可选替代方案。

- 签名与消息**可以分离**
  - 对消息进行单独的日志记录
  - **可**执行程序的签名记录，检查病毒
  - **文档多方**签名，可以避免嵌套签名

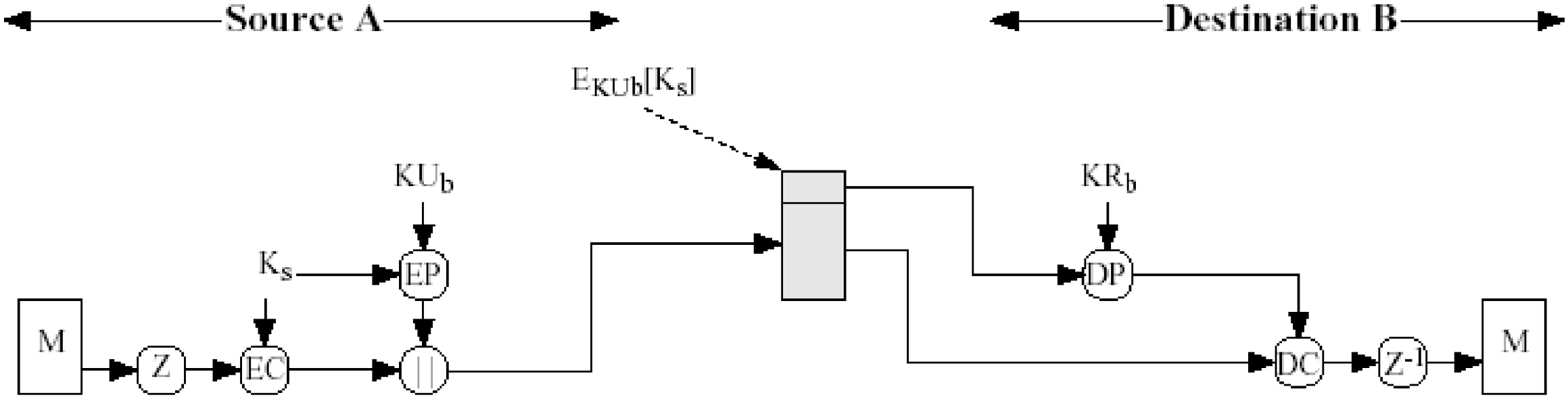
# PGP操作描述

- 保密性
  - 发送者生成报文和128位会话密钥随机数
  - 采用CAST-128(或IDEA或3DES)对报文加密
  - 采用RSA，使用接收者的公开密钥加密会话密钥，并置到报文前面
  - 接收者采用RSA，解密和恢复会话密钥
  - 接收者使用会话密钥解密报文
  - 可以使用Diffie-Hellman算法生成会话密钥
- 保密性与认证
  - 同时实现保密和认证

# 保密性

- 发送方
  - **生成消息**M**并**为该消息生**成一个随机数作**为会话密钥。
  - **用会**话密钥加密M
  - **用接收者的公**钥加密会话**密**钥并与消息M结合

- 接收方
  - **用自己的私**钥解密恢复会话密钥
  - **用会**话密钥解密恢复消息M

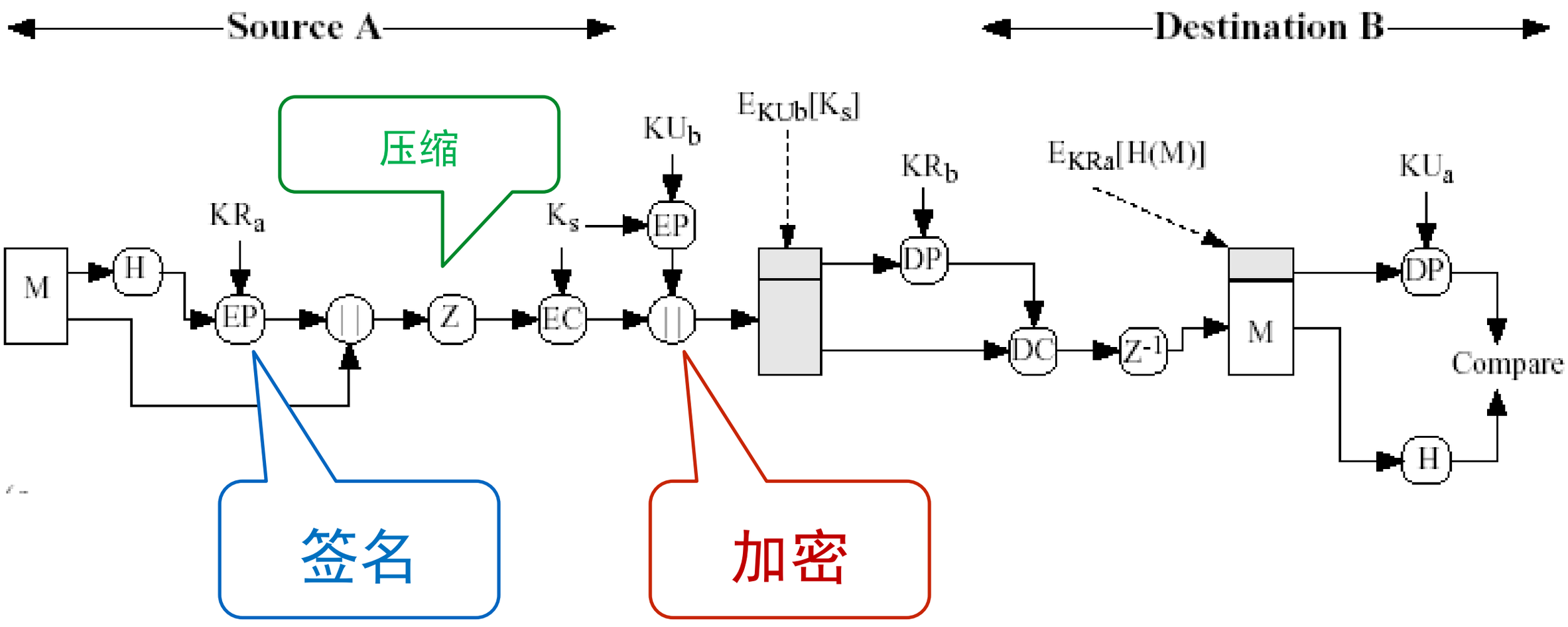


# 保密性说明

- 对称加密算法和公钥加密算法的结合可以缩短加密时间
- 用公钥算法解决了会话密钥的单向分发问题
  - 不需要专门的会话密钥交换协议
  - 由于邮件系统的存储-转发的特性，用握手方式交换密钥不太可能
- 每个消息都有自己的一次性密钥，进一步增强了保密强度。所以，每个密钥只加密很小部分的明文内容

# 保密与认证的结合

- **两种服**务都需要时，发送者先用自己的私钥签名，然后用**会**话密钥加密消息，再用接收者的公钥加密会话密钥。



# 加密密钥和密钥环

- PGP使用四种类型的密钥
  - 一次性会话传统密钥
  - 公钥
  - 私钥
  - 基于口令短语的传统密钥
- PGP对密钥的需求
  - 会话密钥：需要一种生成不可预知的会话密钥的方法，PGP使用了一种复杂的随机密钥生成算法(一定的真随机性)
  - 公钥和私钥
    - 需要某种手段来标识具体的密钥
    - 一个用户拥有多个公钥/私钥对
    - 密钥更新管理
  - 私钥如何保存

# 密钥标识符和钥匙环

- 一个用户有多个公钥/私钥对时，接收者如何知道发送者是用哪个公钥来加密会话密钥的？
  - 将公钥与消息一起传送。
  - 将一个标识符与一个公钥关联，对一个用户来说唯一。即用户ID和密钥ID标识一个密钥
- 定义KeyID 包括64个有效位(PGP采用公钥的低64位作为KeyID)
- 对于PGP数字签名，KeyID也很必需。用哪个公钥来验证签名？
- 钥匙环
  - KeyID对于PGP非常关键。
    - PGP消息中包括两个keyID，分别提供保密与认证功能。
    - 需要一种系统化的方法存储和组织这些密钥以保证有效使用这些密钥
- PGP密钥管理方案
  - 用户机器(节点)上有一对数据结构：
    私钥环：存储本节点拥有的公钥/私钥对
    公钥环：存储本节点所知道的其他用户的公钥

# PGP私钥环

- 信息
  - 时间戳、KeyID、公钥、私钥、UserID
- UserID
  - 通常是用户的邮件地址。也可以是一个名字，可以重名
- 私钥如何保存
  - 用户选择一个口令短语用于加密私钥
  - 当系统用RSA生成一个新的公钥/私钥对时，要求用户输入口令短语。对该短语使用SHA-1生成一个160位的散列码后，销毁该短语
  - 系统用其中128位作为密钥用CAST-128加密私钥，然后销毁这个散列码，并将加密后的私钥存储到私钥环中
  - 当用户要访问私钥环中的私钥时，必须提供口令短语。PGP将取出加密后的私钥，生成散列码，解密私钥

# 邮件数据处理

- 顺序：签名 —— 压缩 —— 加密
- 压缩对邮件传输或存储都有节省空间的好处
- 签名后压缩的原因
  - 不需要为检验签名而保留压缩版本的消息
  - 为了检验而再做压缩不能保证一致性，压缩算法的不同实现版本可能会产生不同的结果
- 压缩之后再做加密的原因
  - 压缩后的消息其冗余小，增加密码分析的难度
  - 若先加密，则压缩难以见效
- E-mail兼容性
  - PGP处理后的消息，部分或者全部是加密后的消息流，为任意的8位字节。某些邮件系统只允许ASC字符，所以PGP提供了转换到ASC格式的功能。采用了Radix-64转换方案

# PGP 发送方处理消息的过程

- 签名
  - **从私**钥环中得到私钥，利用userid**作**为索引
  - PGP**提示**输入口令短语，恢复私钥
  - **构造**签名部分
- **加密**
  - PGP产生一个会话密钥，并加密消息
  - PGP**用接收者**userid**从公**钥环中获取其公钥
  - **构造消息的会**话密钥部分

# PGP接收方处理消息的过程

- **解密消息**
  - PGP**用消息的会**话密钥部分中的KeyID**作**为索引，从私钥环中获取**私**钥
  - PGP**提示**输入口令短语，恢复私钥
  - PGP**恢复会**话密钥，并解密消息
- 验证消息
  - PGP**用消息的**签名部分中的KeyID**作**为索引，从公钥环中获取发送**者的公**钥
  - PGP**恢复被**传输过来的消息摘要
  - PGP对于接收到的消息作摘要，并与上一步的结果作比较

# 答疑