南京大学 软件学院
NANJING UNIVERSITY · SOFTWARE INSTITUTE

Network Security

CISCO SYSTEMS
EMPOWERING THE INTERNET GENERATION℠

南京大学 NANJING UNIVERSITY

Network Security

- 网络安全问题概述
- 一般的数据加密模型
- 对称密钥和公钥密码体制
- 数字签名
- 防火墙
  - 访问控制列表ACL

网络安全问题概述

- 计算机网络上的通信面临以下的四种威胁：

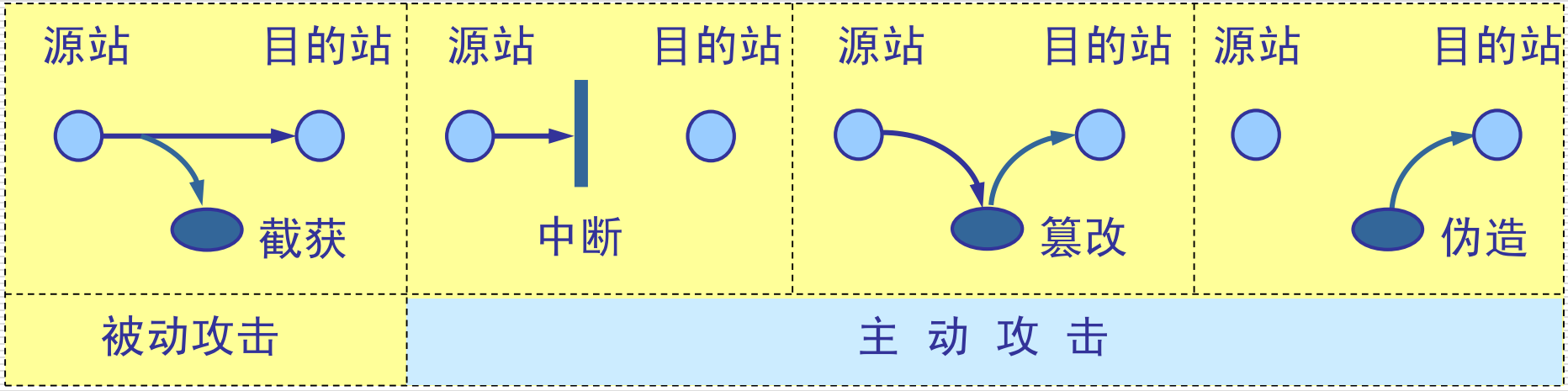
(1) 截获——从网络上窃听他人的通信内容。

(2) 中断——有意中断他人在网络上的通信。

(3) 篡改——故意篡改网络上传送的报文。

(4) 伪造——伪造信息在网络上传送。

被动攻击和主动攻击



- 截获信息的攻击称为被动攻击
- 更改信息和拒绝用户使用资源的攻击称为主动攻击。

被动攻击和主动攻击

- 在被动攻击中，攻击者只是观察和分析某一个协议数据单元 PDU 而不干扰信息流。
- 主动攻击是指攻击者对某个连接中通过的 PDU 进行各种处理。
  - 更改报文流
  - 拒绝报文服务
  - 伪造连接初始化

计算机网络通信安全的目标

- 防止析出报文内容
- 防止通信量分析
- 检测更改报文流
- 检测拒绝报文服务
- 检测伪造初始化连接

恶意程序(malicious program)

- 计算机病毒——会“传染”其他程序的程序，“传染”通过修改其他程序来把自身或其变种复制进去而完成。
- 计算机蠕虫——通过网络的通信功能将自身从一个结点发送到另一个结点并启动运行的程序。
- 特洛伊木马——一种程序，它执行的功能超出所声称的功能。
- 逻辑炸弹——一种当运行环境满足某种特定条件时执行其他特殊功能的程序。

计算机网络安全的内容

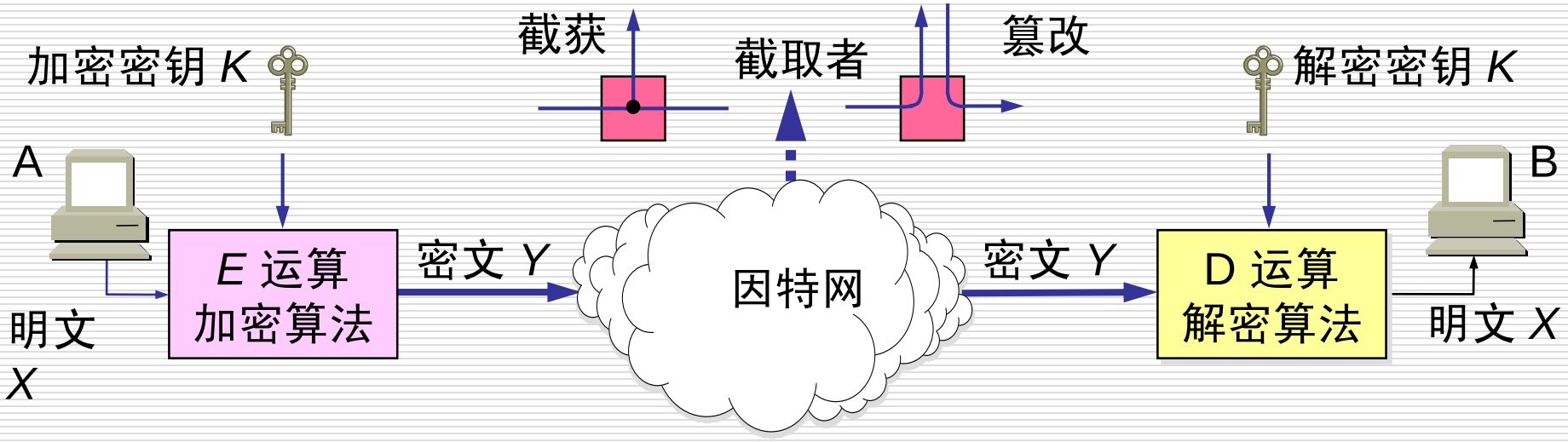
- 保密性
- 安全协议的设计
- 访问控制

南京大学 NANJING UNIVERSITY

Network Security

- 网络安全问题概述
- 一般的数据加密模型
- 对称密钥和公钥密码体制
- 数字签名
- 防火墙
  - 访问控制列表ACL

一般的数据加密模型



一些重要概念

- 密码编码学(cryptography)是密码体制的设计学，而密码分析学(cryptanalysis)则是在未知密钥的情况下从密文推演出明文或密钥的技术。密码编码学与密码分析学合起来即为密码学(cryptology)。
- 如果不论截取者获得了多少密文，但在密文中都没有足够的信息来唯一地确定出对应的明文，则这一密码体制称为无条件安全的，或称为理论上是不可破的。
- 如果密码体制中的密码不能被可使用的计算资源破译，则这一密码体制称为在计算上是安全的。

南京大学 NANJING UNIVERSITY

Network Security

- 网络安全问题概述
- 一般的数据加密模型
- 对称密钥和公钥密码体制
- 数字签名
- 防火墙
  - 访问控制列表ACL

对称密钥密码体制

- 所谓常规密钥密码体制，即加密密钥与解密密钥是相同的密码体制。
- 这种加密系统又称为对称密钥系统。

数据加密标准 DES

- 数据加密标准 DES 属于常规密钥密码体制，是一种分组密码。
- 在加密前，先对整个明文进行分组。每一个组长为 64 位。
- 然后对每一个 64 位 二进制数据进行加密处理，产生一组 64 位密文数据。
- 最后将各组密文串接起来，即得出整个的密文。
- 使用的密钥为 64 位（实际密钥长度为 56 位，有 8 位用于奇偶校验)。

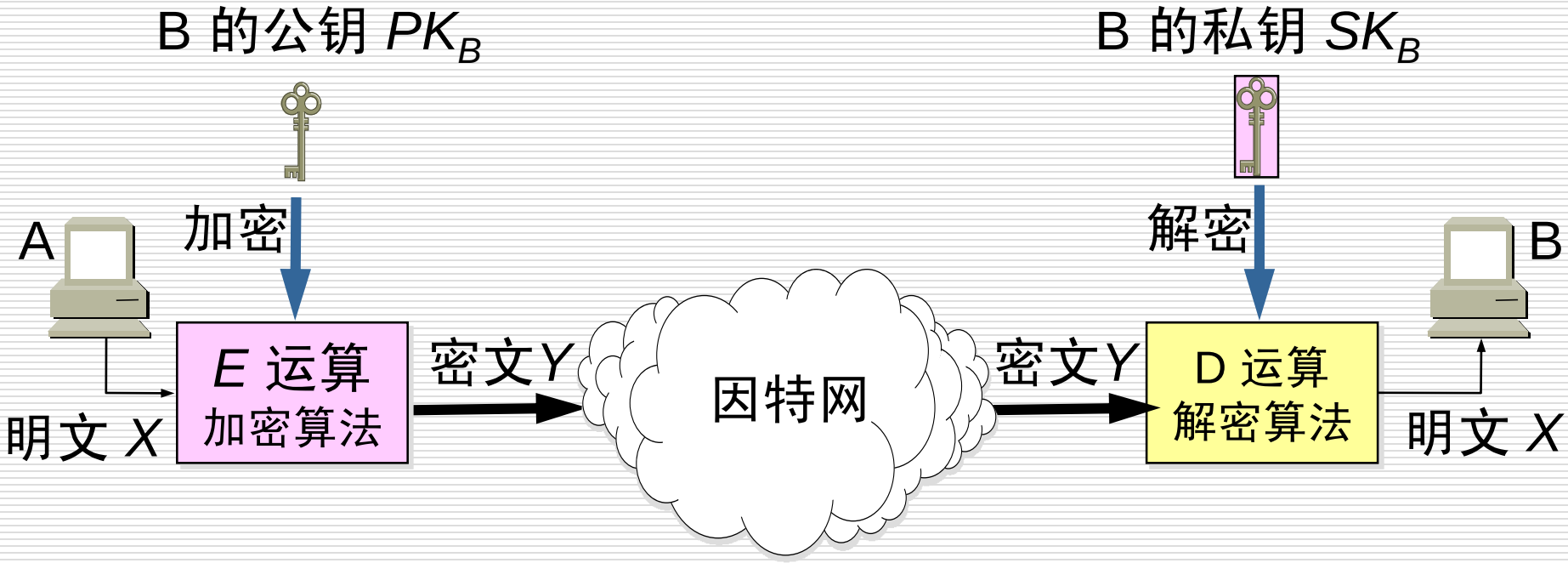
DES 的保密性

- DES 的保密性仅取决于对密钥的保密，而算法是公开的。尽管人们在破译 DES 方面取得了许多进展，但至今仍未能找到比穷举搜索密钥更有效的方法。
- DES 是世界上第一个公认的实用密码算法标准，它曾对密码学的发展做出了重大贡献。
- 目前较为严重的问题是 DES 的密钥的长度。
- 现在已经设计出来搜索 DES 密钥的专用芯片。

公钥密码体制

- 公钥密码体制使用不同的加密密钥与解密密钥，是一种“由已知加密密钥推导出解密密钥在计算上是不可行的”密码体制。
- 公钥密码体制的产生主要是因为两个方面的原因，一是由于常规密钥密码体制的密钥分配问题，另一是由于对数字签名的需求。
- 现有最著名的公钥密码体制是RSA 体制，它基于数论中大数分解问题的体制，由美国三位科学家 Rivest, Shamir 和 Adleman 于 1976 年提出并在 1978 年正式发表。

公钥密码体制



- 在公钥密码体制中，加密密钥(即公钥) *PK* 是公开信息，而解密密钥(即私钥或秘钥) *SK* 是需要保密的
- 加密算法 *E* 和解密算法 *D* 也都是公开的
- 虽然*SK* 是由*PK* 决定的，但却不能根据 *PK* 计算出 *SK*

公钥算法的特点

- 发送者 A 用 B 的公钥 PK_B 对明文 X 加密（E 运算）后，在接收者 B 用自己的私钥 SK_B 解密（D 运算），即可恢复出明文：

$$D_{SK_B}(Y) = D_{SK_B}(E_{PK_B}(X)) = X$$

- 解密密钥是接收者专用的秘钥，对其他人都保密。
- 加密密钥是公开的，但不能用它来解密，即

$$D_{PK_B}(E_{PK_B}(X)) \neq X$$

公钥算法的特点（续）

- 加密和解密的运算可以对调，即

$$E_{PK_B}(D_{SK_B}(X)) = D_{SK_B}(E_{PK_B}(X)) = X$$

- 在计算机上可容易地产生成对的 *PK* 和 *SK*。
- 从已知的 *PK* 实际上不可能推导出 *SK*，即从 *PK* 到 *SK* 是“计算上不可能的”。
- 加密和解密算法都是公开的。

应当注意

- 任何加密方法的安全性取决于密钥的长度，以及攻破密文所需的计算量
- 在这方面，公钥密码体制并不比传统加密体制更加优越
- 由于目前公钥加密算法的开销较大，在可见的将来还不会放弃传统的加密方法
- 公钥需要密钥分配协议，具体的分配过程并不比采用传统加密方法时更简单

南京大学 NANJING UNIVERSITY

Network Security

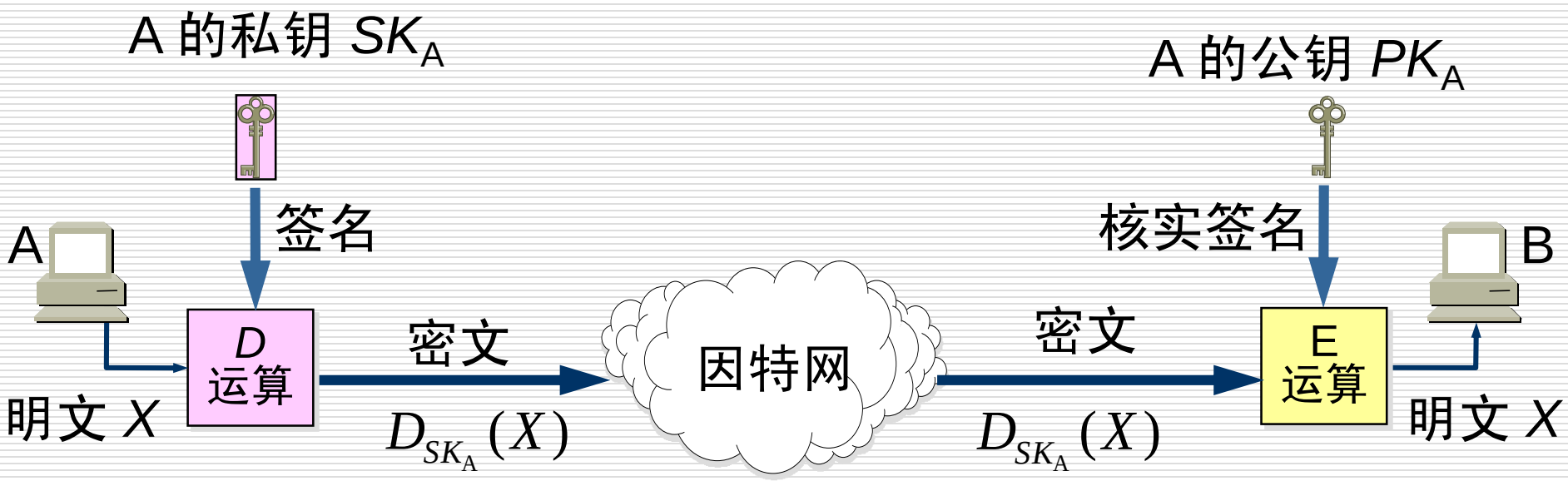
- 网络安全问题概述
- 一般的数据加密模型
- 对称密钥和公钥密码体制
- 数字签名
- 防火墙
  - 访问控制列表ACL

数字签名

- 数字签名必须保证以下三点：
  - 报文鉴别——接收者能够核实发送者对报文的签名
  - 报文的完整性——发送者事后不能抵赖对报文的签名
  - 不可否认——接收者不能伪造对报文的签名
- 现在已有多种实现各种数字签名的方法。但采用公钥算法更容易实现

南京大学 NANJING UNIVERSITY

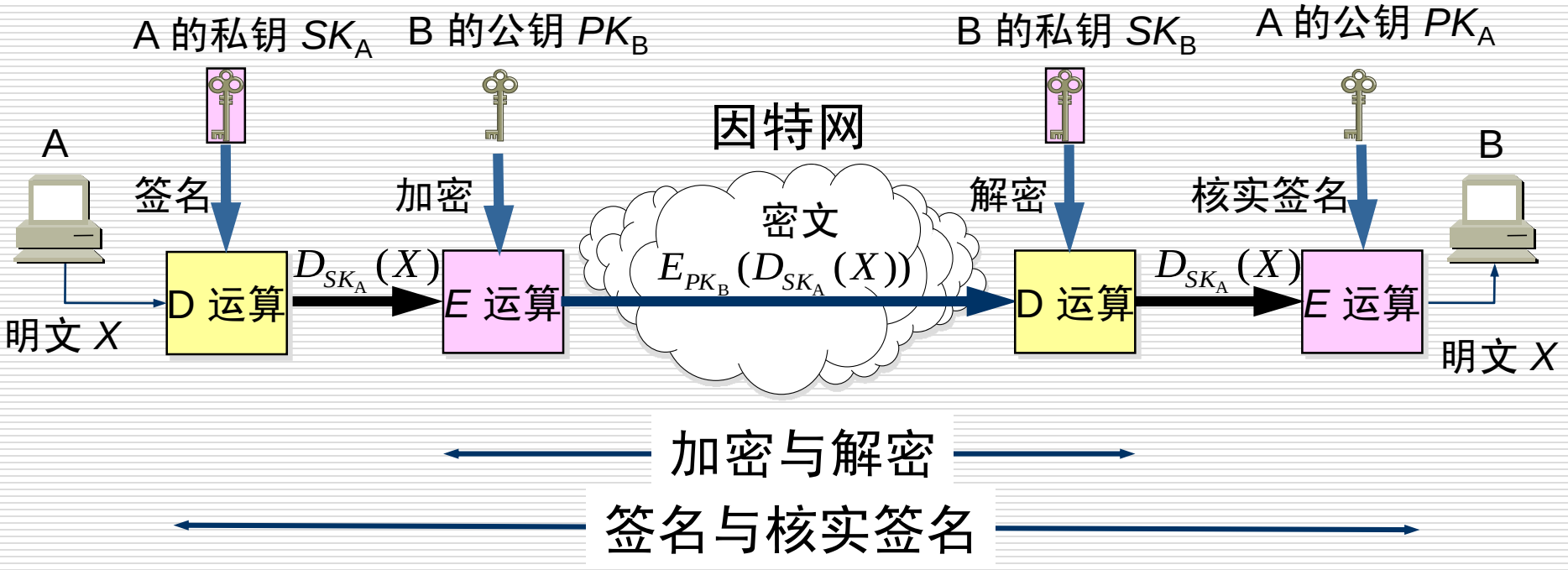
数字签名的实现



数字签名的实现

- 因为除 A 外没有别人能具有 A 的私钥，所以除 A 外没有别人能产生这个密文。因此 B 相信报文 *X* 是 A 签名发送的。
- 若 A 要抵赖曾发送报文给 B，B 可将明文和对应的密文出示给第三者。第三者很容易用 A 的公钥去证实 A 确实发送 *X* 给 B。
- 反之，若 B 将 *X* 伪造成 *X*‘，则 B 不能在第三者前出示对应的密文。这样就证明了 B 伪造了报文。

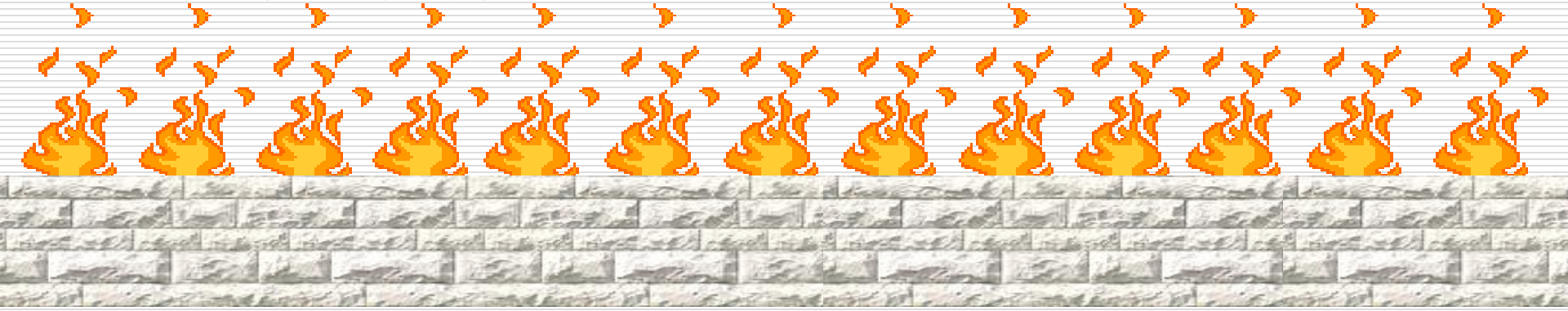
具有保密性的数字签名


A 的私钥 SK_A
B 的公钥 PK_B
B 的私钥 SK_B
A 的公钥 PK_A
A
签名
加密
因特网
解密
核实签名
B
D 运算
$D_{SK_A}(X)$
E 运算
密文
$E_{PK_B}(D_{SK_A}(X))$
D 运算
$D_{SK_A}(X)$
E 运算
明文 X
明文 X
加密与解密
签名与核实签名

南京大学
NANJING UNIVERSITY

Network Security

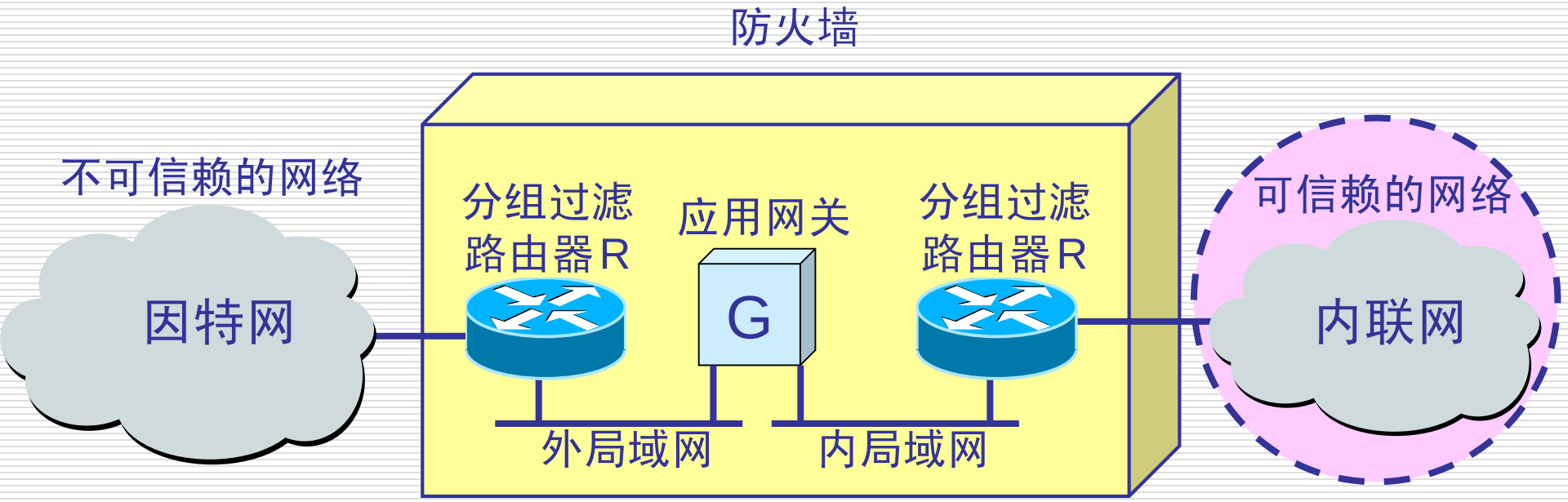
- 网络安全问题概述
- 一般的数据加密模型
- 对称密钥和公钥密码体制
- 数字签名
- **防火墙**
  - **访问控制列表ACL**



防火墙(firewall)

- 防火墙是由软件、硬件构成的系统，是一种特殊编程的路由器，用来在两个网络之间实施接入控制策略。接入控制策略是由使用防火墙的单位自行制订的，为的是可以最适合本单位的需要。
- 防火墙内的网络称为“可信赖的网络”(trusted network)，而将外部的因特网称为“不可信赖的网络”(untrusted network)。
- 防火墙可用来解决内联网和外联网的安全问题。

防火墙在互连网络中的位置



防火墙的功能

- 防火墙的功能有两个：阻止和允许。
- “阻止”就是阻止某种类型的通信量通过防火墙（从外部网络到内部网络，或反过来）。
- “允许”的功能与“阻止”恰好相反。
- 防火墙必须能够识别通信量的各种类型。不过在大多数情况下防火墙的主要功能是“阻止”。

防火墙技术一般分为两类

- 网络级防火墙——用来防止整个网络出现外来非法的入侵。属于这类的有**分组过滤**和**授权服务器**
  - 前者检查所有流入本网络的信息，然后拒绝不符合事先制订好的一套准则的数据
  - 后者则检查用户的登录是否合法
- 应用级防火墙——从应用程序来进行接入控制。通常使用应用网关或代理服务器来区分各种应用
  - 例如，可以只允许通过访问万维网的应用，而阻止 FTP 应用通过

What Are ACLs?

- An ACL is a list of instructions that tells a router what type of packets to permit or deny.
  - You must configure an ACL if you want a router to deny some packets. Otherwise, the router will accept and forward all packets as long as the link is up.
  - You can permit or deny packets based upon such thing as:
    - Source address
    - Destination address
    - Upper Layer protocols (e.g. TCP & UDP ports)

Testing Packets with ACLs

- To determine whether a packet is to be permitted or denied, it is tested against the ACL statements in sequential order.
  - When a statement "matches," **no more statements are evaluated**.
  - The packet is either permitted or denied.
- There is an implicit "deny any" statement at the end of the ACL
  - If a packet does not match any of the statements in the ACL, it is dropped.

Testing Packets with ACLs

- Example:
  - If we have an ACL list described as below:
    - 1. Permit packets from 192.168.100.1 to pass
    - 2. Permit packets from 192.168.100.2 to pass
    - 3. Deny packets from 192.168.100.3
  - Then:
    - Packets from 192.168.100.1 will be forwarded
    - Packets from 192.168.100.3 will be denied
    - But how does the router process the packets from 192.168.100.4?

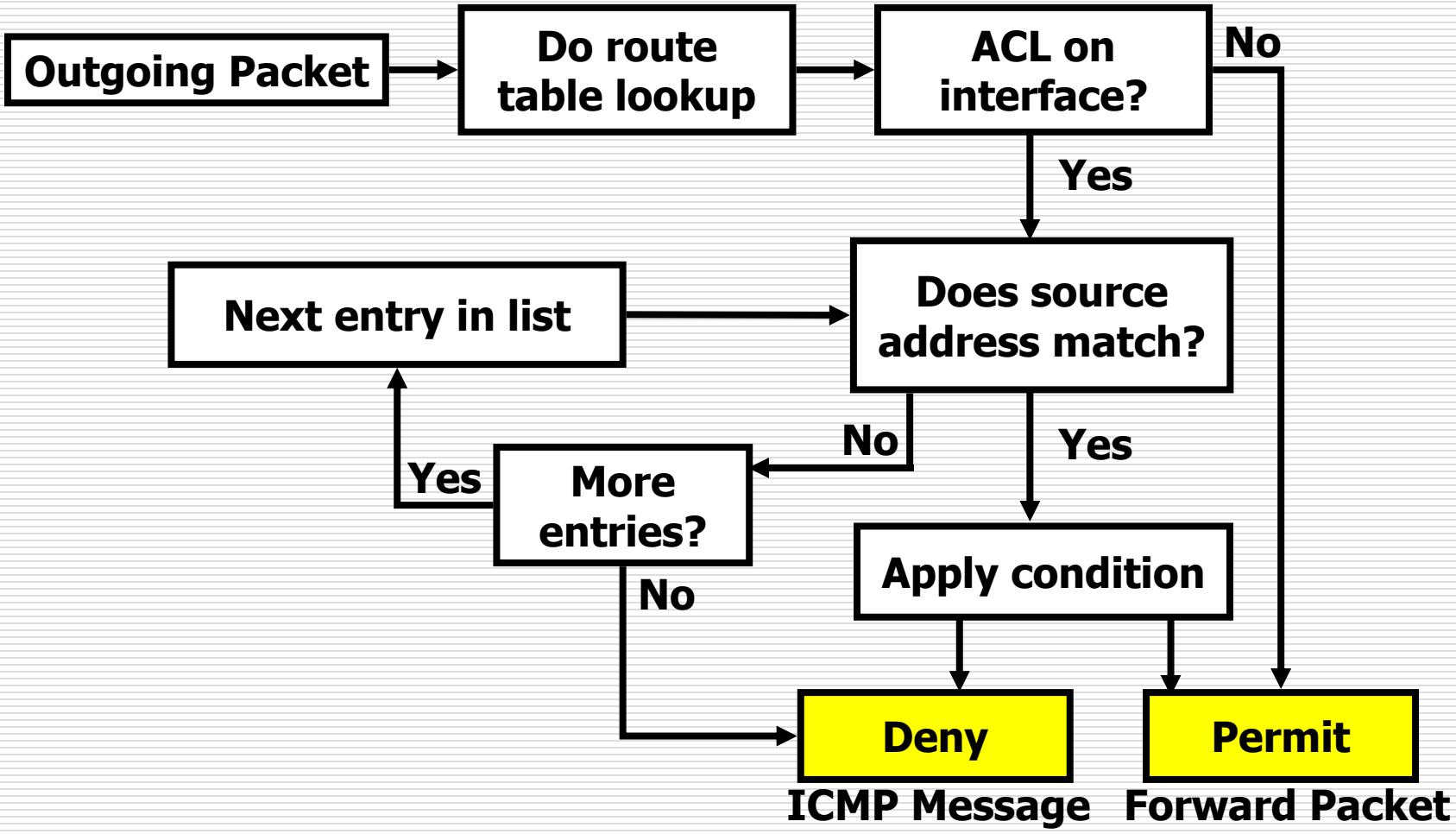
Packets from 192.168.100.4 will be *denied*

How a Router Uses an ACL (outbound)

- Check to see if packet is routable. If so, look up route in routing table
- Check for an ACL for the outbound interface
  - If no ACL, switch the packet out the destination interface
  - If an ACL, check the packet against the ACL statements sequentially--denying or permitting based on a matched condition.
- If no statement matches, what happens?

Deny all these packets!

Outbound Standard ACL Process



Two Basic Tasks (Standard ACL)

- Write the ACL statements sequentially in global configuration mode.

  Router(config)#**access-list** *access-list-number* **{permit/deny}** *{test-conditions}*

  Lab-D(config)#access-list 1 deny 192.5.5.10 0.0.0.0

- Group the ACL to one or more interfaces in interface configuration mode.

  Router(config-if)#{protocol} **access-group** *access-list-number* **{in/out}**

  Lab-D(config-if)#ip access-group 1 out

The ***access-list-number*** parameter

- ACLs come in many types. The ***access-list-number*** specifies what types.
- The table below shows common access list types.

ACL Type	ACL Number
IP Standard	1 to 99
IP Extended	100 to 199
AppleTalk	600 to 699
IPX Standard	800 to 899
IPX Extended	900 to 999
IPX SAP	1000 to 1099

Router(config)#**access-list *access-list-number* {permit/deny}***{test-conditions}***

The **`permit/deny`** parameter

- After you've typed **`access-list`** and chosen the correct ***`access-list-number`***, you type either **`permit`** or **`deny`** depending on the action you wish to take.

Permit
Forward Packet

Deny
ICMP Message

`Router(config)#`**`access-list`** *`access-list-number`* **`{permit/deny}`** *`{test-conditions}`*

南京大學 NANJING UNIVERSITY

The *{test-conditions}* parameter

- In the {test conditions} portion of the ACL, common to most access lists is the source address' ip mask and wildcard mask.
- The source address can be a subnet, a range of addresses, or a single host. It is also referred to as the ip mask because the wildcard mask uses the source address to check bits.
- The wildcard mask tells the router what bits to check.

Router(config)#**access-list** *access-list-number* **{permit/deny}** ***{test-conditions}***

The Wildcard Mask

- A wildcard mask is written to tell the router what bits in the address to match and what bits to ignore.
  - A “0” bit means check this bit position
  - A “1” means ignore this bit position
- Our previous example of 192.5.5.10 0.0.0.0 can be rewritten in binary as:

  11000000.00000101.00000101.00001010
  (Source address)

  00000000.00000000.00000000.00000000
  (Wildcard mask)

Masking Practice

- Write an ip mask and wildcard mask to check for all hosts on the network: 192.5.5.0 255.255.255.0
- Answer: 192.5.5.0 0.0.0.255
  - Notice that this wildcard mask is a mirror image of the default subnet mask for a Class C address.
  - WARNING: This is a helpful rule only when looking at whole networks or subnets.

Masking Practice

- Write an ip mask and wildcard mask to check for all hosts in the subnet: 192.5.5.32 255.255.255.224
  - If you answered 192.5.5.32 0.0.0.31 YOU'RE RIGHT!!
  - 0.0.0.31 is the mirror image of 255.255.255.224
  - Let's look at both in binary:
    - 11111111.11111111.11111111.11100000 (255.255.255.224)
    - 00000000.00000000.00000000.00011111 (0.0.0.31)

Time Savers: the **any** command

- Since ACLs have an implicit "deny any" statement at the end, you must write statements to permit others through.
- Using our previous example, if the students are denied access and all others are allowed, you would write two statements:
  - **Lab-A(config)#access-list 1 deny 192.5.5.0 0.0.0.127**
  - **Lab-A(config)#access-list 1 permit 0.0.0.0 255.255.255.255**
- Since the last statement is commonly used to override the "deny any," Cisco gives you an option--the **any** command:
  - **Lab-A(config)#access-list 1 permit any**

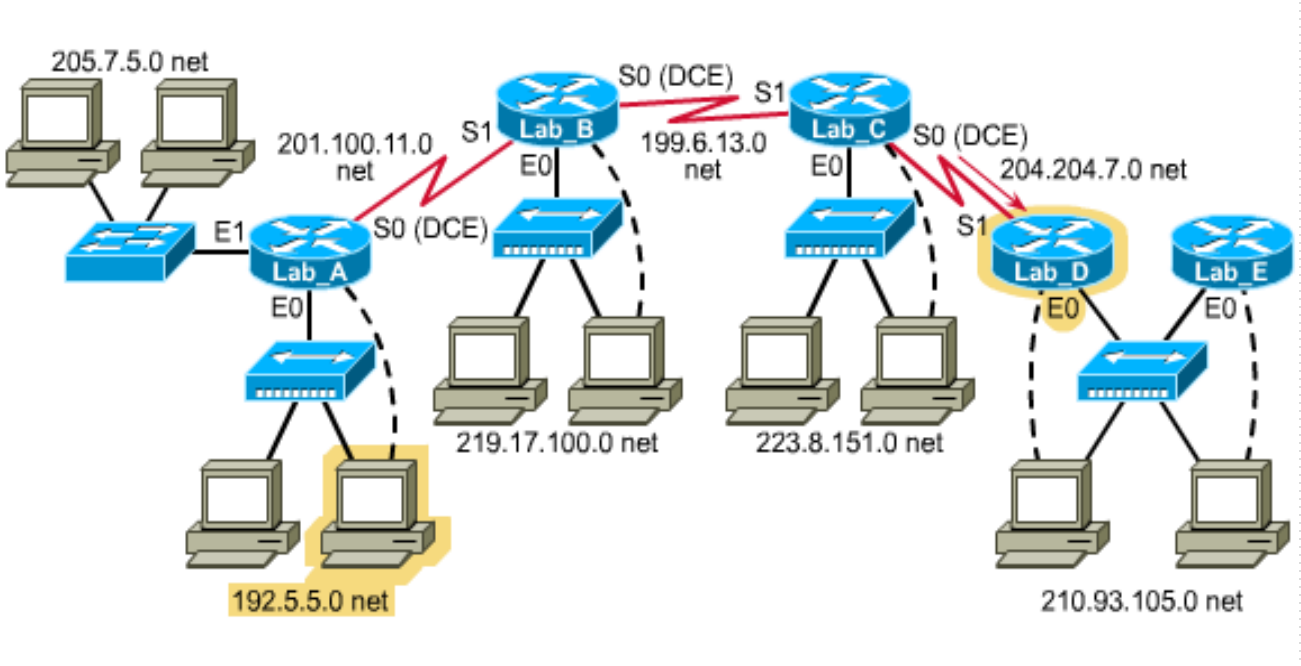
Time Savers: the `host` command

- Many times, a network administrator will need to write an ACL to permit a particular host (or deny a host). The statement can be written in two ways. Either...
  - **`Lab-A(config)#access-list 1 permit 192.5.5.10 0.0.0.0`**
- or...
  - **`Lab-A(config)#access-list 1 permit host 192.5.5.10`**

南京大学 NANJING UNIVERSITY

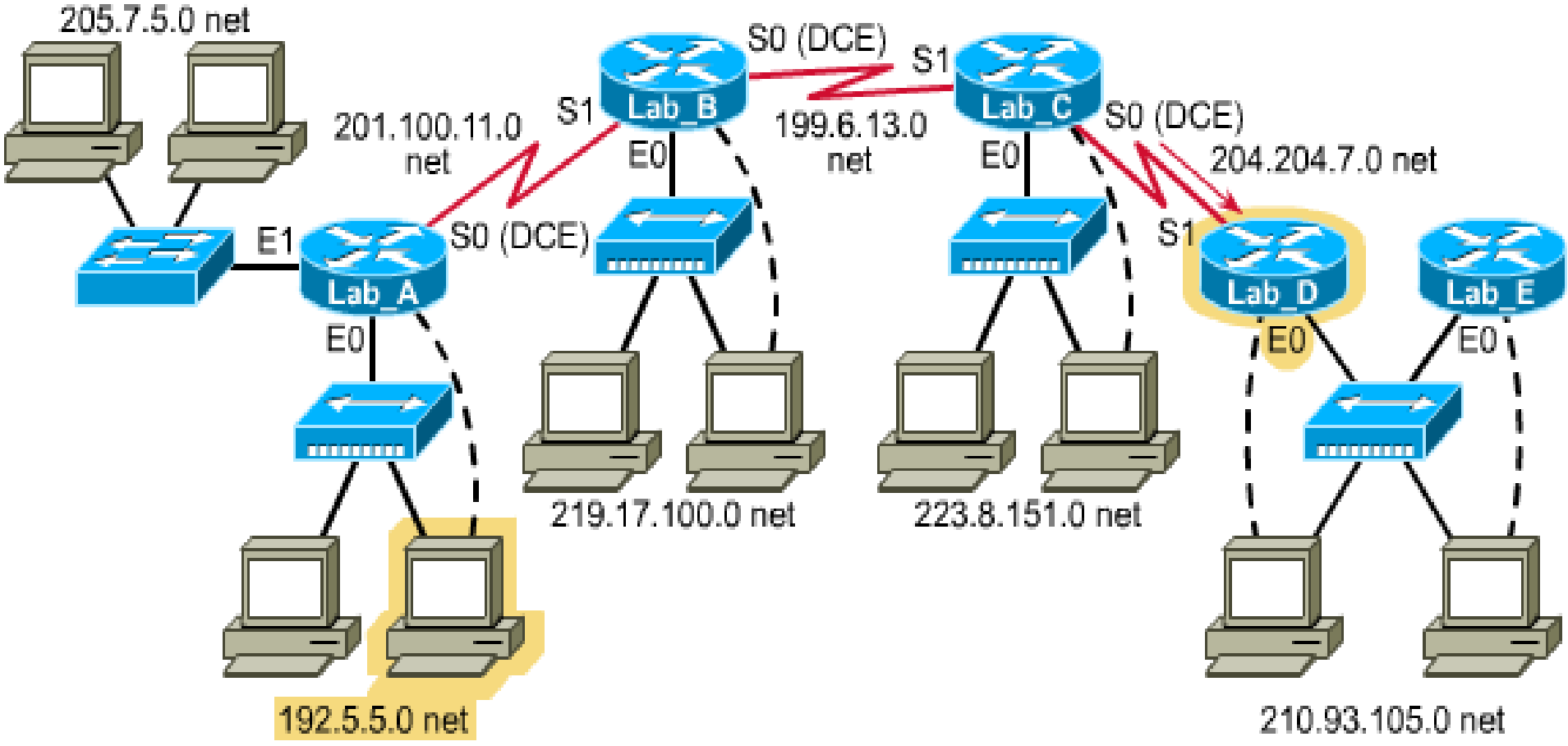
Correct Placement of Standard ACLs

- Standard ACLs do not have a destination parameter. Therefore, you place standard ACLs as close to the destination as possible.
- To see why, ask yourself what would happen to all ip traffic if you placed a "deny 192.5.5.0 0.0.0.255" statement on Lab-A's E0?



南京大学
NANJING UNIVERSITY

Correct Placement of Standard ACLs



Extended ACL Overview

- Extended ACLs are numbered from 100 - 199 and “extend” the capabilities of the standard ACL.
- Extensions include the ability to filter traffic based on...
  - **destination address**
  - **portions of the ip protocol**
    - You can write statements to deny only protocols such as “icmp” or routing protocols like “rip” and “igrp”
  - **upper layers of the TCP/IP protocol suite**
    - You can write statements to deny only protocols such as “tftp” or “http”
    - You can use an operand like eq, gt, lt, and neq (equal to, greater than, less than, and not equal to) to specify how to handle a particular protocol.
    - For example, if you wanted an access list to permit all traffic except http access, you would use **`permit ip any any neq 80`**

Two Basic Tasks (Extended ACL)

- Write the ACL statements sequentially in global configuration mode.

```
Router(config)# access-list access-list-number
{permit|deny} {protocol|protocol-keyword}{source
source-wildcard} {destination destination-wildcard}
[protocol-specific options] [log]
Lab-A(config)#access-list 101 deny tcp 192.5.5.0
0.0.0.255 210.93.105.0 0.0.0.255 eq telnet log
```

- Group the ACL to one or more interfaces in interface configuration mode

```
Router(config-if)#{protocol} access-group
access-list-number {in/out}
Lab-A(config-if)#ip access-group 101 out
```

The Extended Parameters

- ***access-list-number***
  - choose from the range 100 to 199
- {***protocol*** | ***protocol-number***}
  - For the CCNA, you only need to know **ip** and **tcp**--many more are available
- {***source source-wildcard***}
  - same as in standard
- {***destination destination-wildcard***}
  - formatted like the standard, but specifies the destination
- [***protocol-specific options***]
  - This parameter is used to specify particular parts of a protocol that needs filtering.

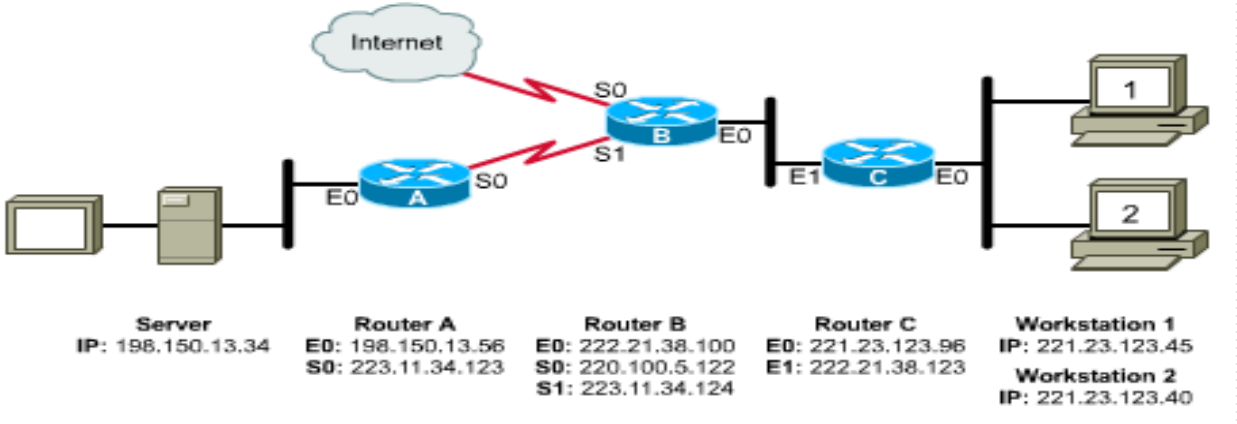
Port Numbers

- Review the various port numbers for the tcp and udp protocols and know the most common ones below.
- You can also simply type the name (**`telnet`**) instead of the number (**`23`**) in the **`{`*`protocol-specific options`*`}`**

Port Number	Description
21	FTP
23	Telnet
25	SMTP
53	DNS
69	TFTP

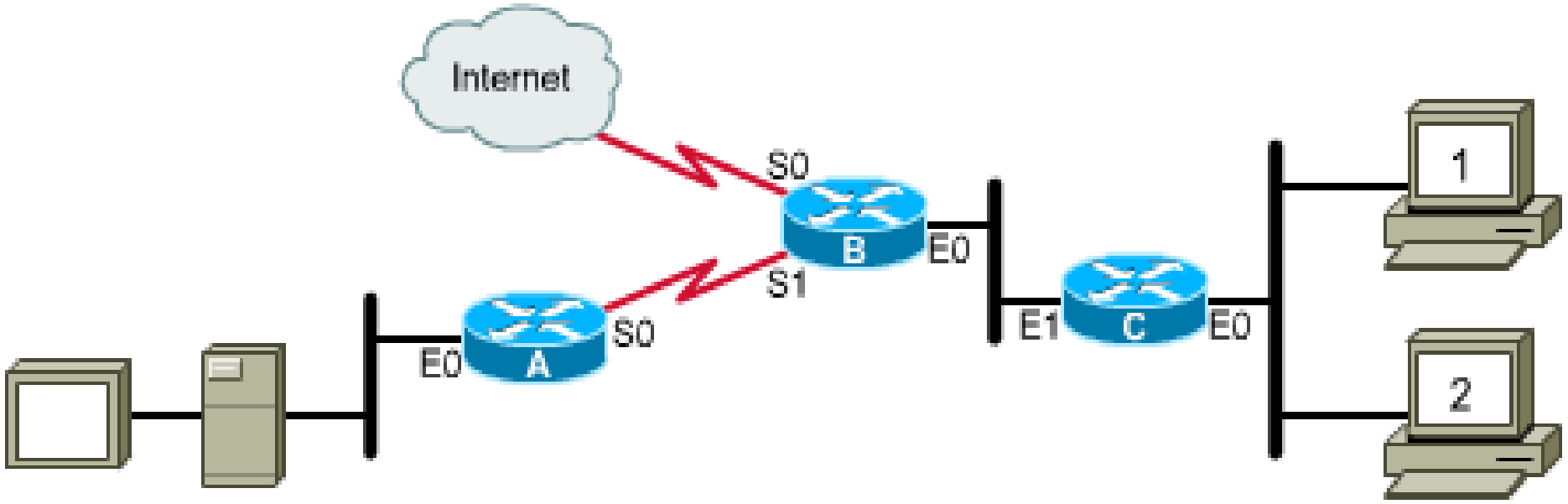
Correct Placement of Extended ACLs

- In the graphic below, we want to deny network 221.23.123.0 from accessing the server 198.150.13.34.
- What router and interface should the access list be applied to?
  - **Write the access list on Router C, apply it to the E0, and specify in**
  - **This will keep the network free of traffic from 221.23.123.0 destined for 198.150.13.34 but still allow 221.23.123.0 access to the Internet**



Correct Placement of Extended ACLs

Since extended ACLs have destination information, you want to place it as close to the source as possible.



Server
IP: 198.150.13.34

Router A
E0: 198.150.13.56
S0: 223.11.34.123

Router B
E0: 222.21.38.100
S0: 220.100.5.122
S1: 223.11.34.124

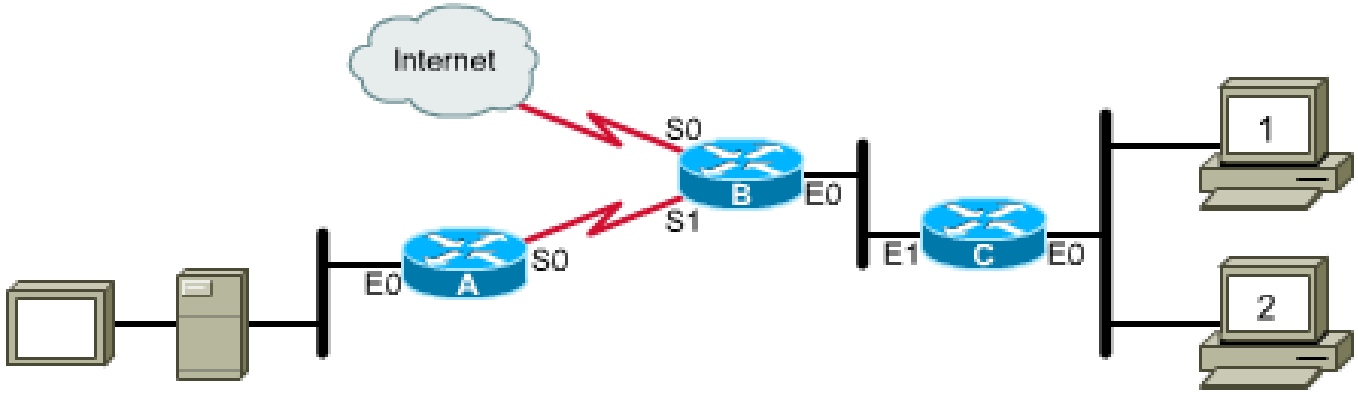
Router C
E0: 221.23.123.96
E1: 222.21.38.123

Workstation 1
IP: 221.23.123.45

Workstation 2
IP: 221.23.123.40

Writing & Applying the ACL

```
Router-C(config)#access-list 100 deny ip 221.23.123.0
  0.0.0.255 198.150.13.34 0.0.0.0
Router-C(config)#access-list 100 permit ip any any
Router-C(config)#int e0
Router-C(config-if)#ip access-group 100 in
```



Server
IP: 198.150.13.34

Router A
E0: 198.150.13.56
S0: 223.11.34.123

Router B
E0: 222.21.38.100
S0: 220.100.5.122
S1: 223.11.34.124

Router C
E0: 221.23.123.96
E1: 222.21.38.123

Workstation 1
IP: 221.23.123.45

Workstation 2
IP: 221.23.123.40

Naming ACLs

- One nice feature in the Cisco IOS is the ability to name ACLs. This is especially helpful if you need more than 99 standard ACLs on the same router.
- Once you name an ACL, the prompt changes and you no longer have to enter the **access-list** and ***access-list-number*** parameters.
- In the example below, the ACL is named **over_and** as a hint to how it should be placed on the interface--**out**

```
Lab-A(config)# ip access-list standard over_and
Lab-A(config-std-nacl)#deny host 192.5.5.10
.........
Lab-A(config-if)#ip access-group over_and out
```

Verifying ACLs

- Show commands:
  - **`show access-lists`**
    - shows all access-lists configured on the router
  - **`show access-lists {`*`name`*` | `*`number`*`}`**
    - shows the identified access list
  - **`show ip interface`**
    - shows the access-lists applied to the interface--both inbound and outbound.
  - **`show running-config`**
    - shows all access lists and what interfaces they are applied on

谢 谢！