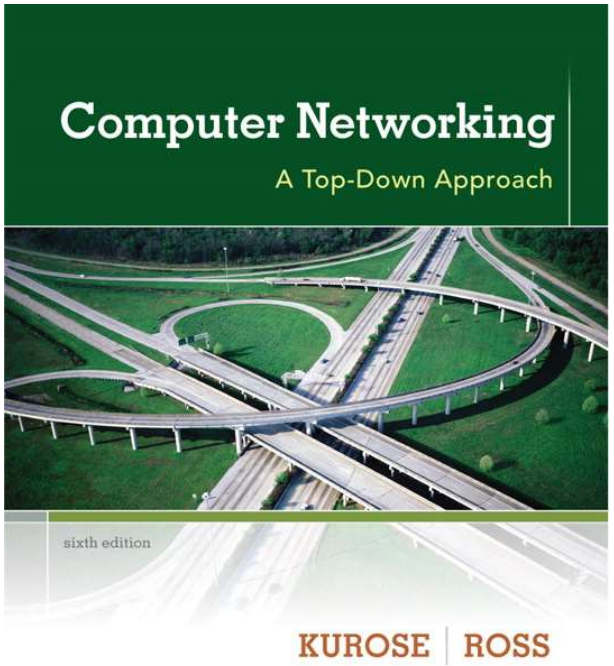
第 8 章
网络安全



中国科学技术大学
自动化系 郑烃
改编自Jim kurose, Keith Ross

Computer Networking: A Top Down Approach
6th edition
Jim Kurose, Keith Ross
Addison-Wesley
March 2012

第8章：网络安全

本章目标：

- 网络安全原理:
  - 加密，不仅仅用于机密性
  - 认证
  - 报文完整性
  - 密钥分发
- 安全实践:
  - 防火墙
  - 各个层次的安全性：应用层，传输层，网络层和链路层

提纲

8.1 什么是网络安全？

8.2 加密原理

8.3 认证

8.4 报文完整性

8.5 密钥分发和证书

8.6 访问控制：防火墙

8.7 攻击和对策

8.8 各个层次的安全性

什么是网络安全？

机密性：只有发送方和预订的接收方能否理解传输的报文内容

- 发送方加密报文
- 接收方解密报文

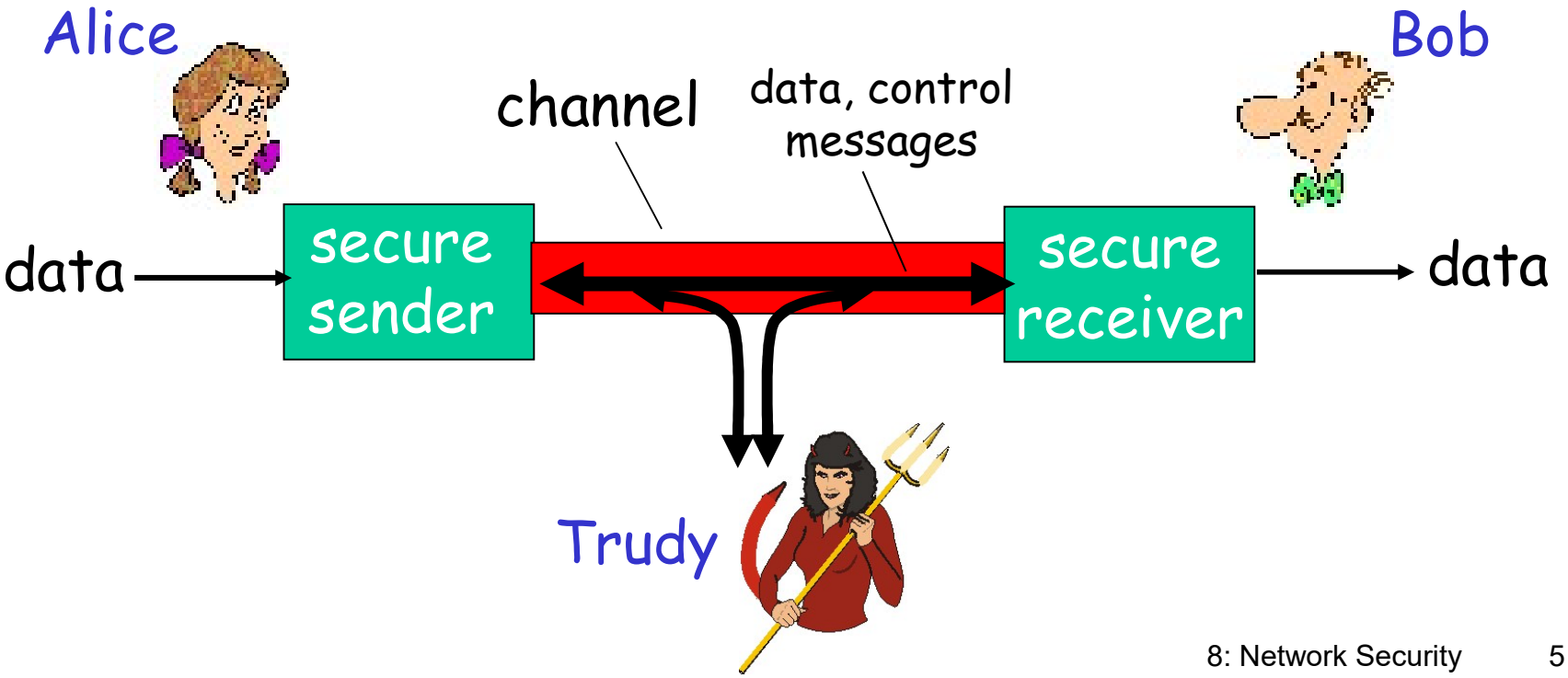
认证：发送方和接收方需要确认对方的身份

报文完整性：发送方、接受方需要确认报文在传输的过程中或者事后没有被改变

访问控制和服务的可用性：服务可以接入以及对用户而言是可用的

朋友和敌人: Alice, Bob, Trudy

- 网络安全世界比较著名的模型
- Bob, Alice (lovers!) 需要安全的通信
- Trudy (intruder) 可以截获，删除和增加报文



谁有可能是Bob, Alice?

- … 现实世界中的Bobs和Alices!
- 电子交易中的Web browser/server (e.g.,在线购买)
- 在线银行的client/server
- DNS servers
- 路由信息的交换
- 其它例子?

网络中的坏蛋

Q: "bad guy"可以干什么?

A: 很多!

- *窃听*: 截获报文
- *插入*: 在连接上插入报文
- *伪装*: 可以在分组的源地址写上伪装的地址
- 劫持: 将发送方或者接收方踢出，接管连接
- *拒绝服务*: 阻止服务被其他正常用户使用 (e.g.,通过对资源的过载使用)

more on this later ……

提纲

8.1 什么是网络安全？

8.2 加密原理

8.3 认证

8.4 报文完整性

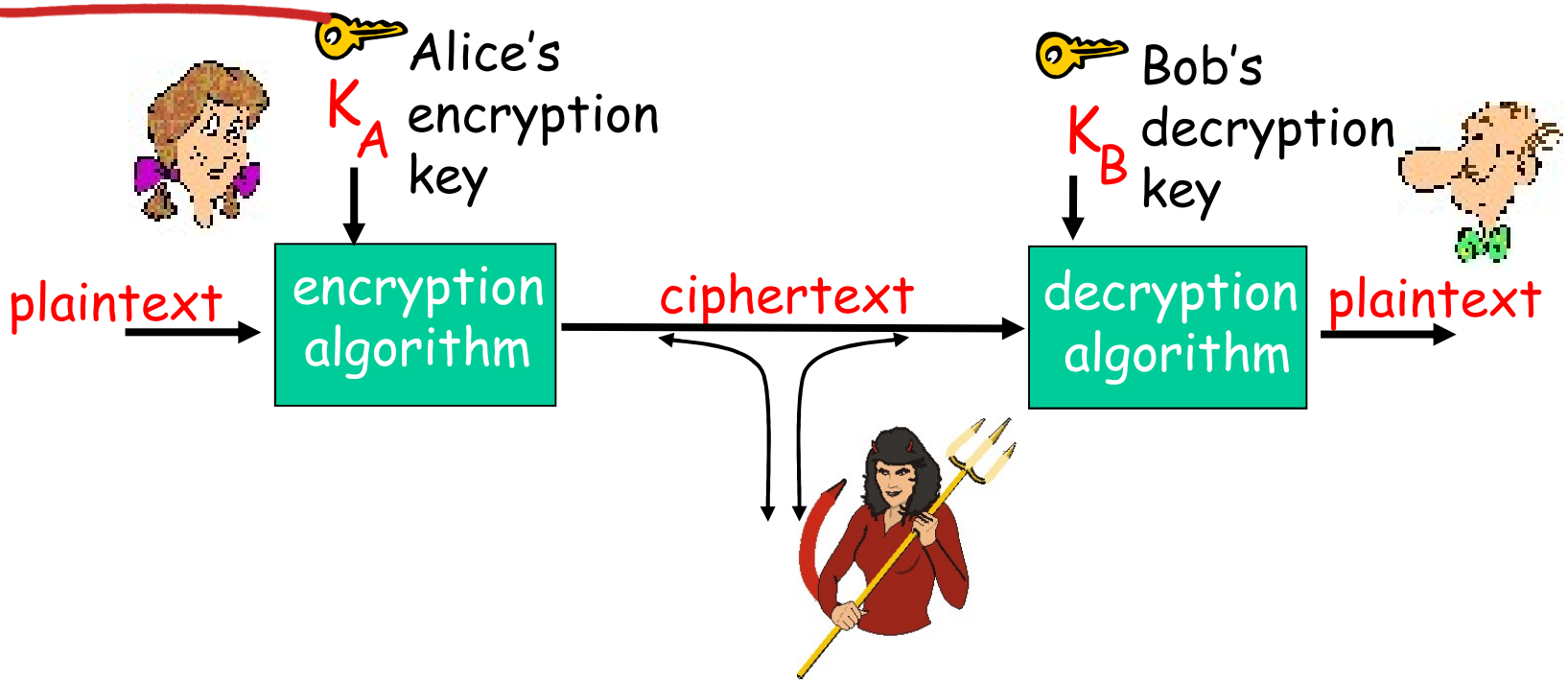
8.5 密钥分发和证书

8.6 访问控制：防火墙

8.7 攻击和对策

8.8 各个层次的安全性

加密语言



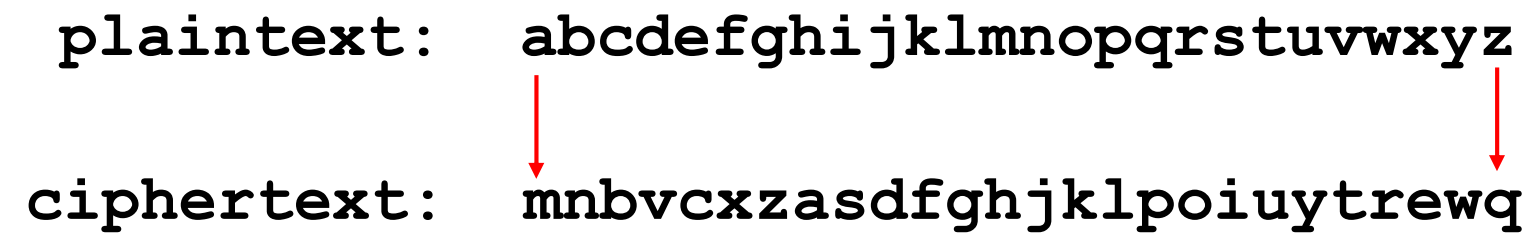
对称密钥密码学：发送方和接收方的密钥相同

公开密钥密码学：发送方使用接收方的公钥进行加密，接收方使用自己的私钥进行解密

对称密钥加密

替换密码: 将一个事情换成另外一个事情

- 单码替换密码: 将一个字母替换成另外一个字母



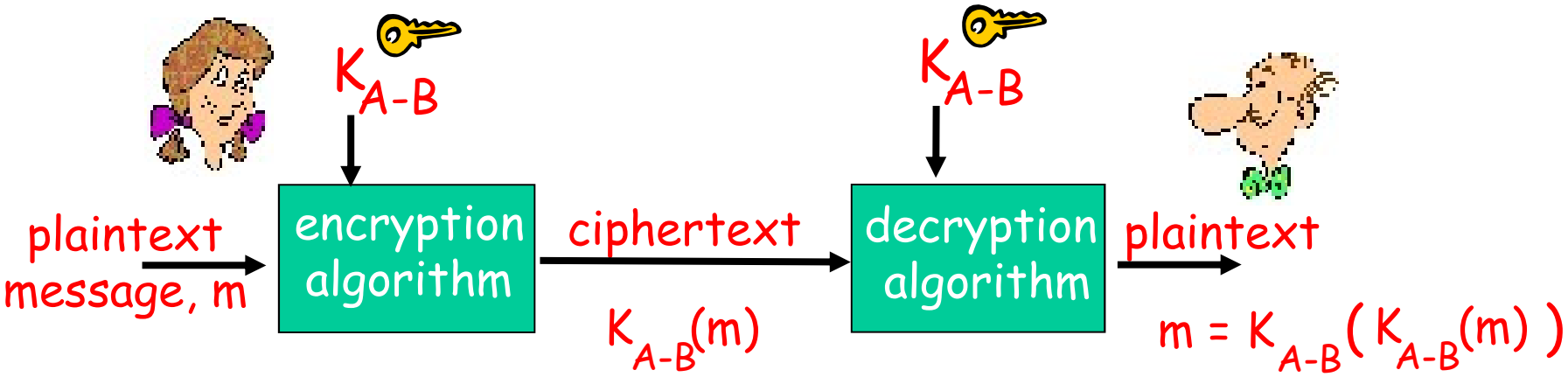
E.g.: Plaintext: bob. i love you. alice

ciphertext: nkn. s gktc wky. mgsbc

Q: 破解这个密码的强度?

- brute force (how hard?)
- other?

对称密钥加密



对称密钥密码: Bob和Alice共享一个对称式的密钥: K_{A-B}

- e.g., 密钥在单码替换加密方法中是替换模式
- Q: 但是Bob和Alice如何就这个密钥达成一致呢?

对称密钥加密学: DES

DES: Data Encryption Standard

- US 加密标准[NIST 1993]
- 56-bit 对称密钥, 64-bit明文输入
- DES有多安全?
  - DES挑战: 56-bit密钥加密的短语 ("Strong cryptography makes the world a safer place") 被解密，用了4个月的时间
  - 可能有后门
- 使DES更安全:
  - 使用3个key， 3重DES 运算
  - 密文分组成串技术

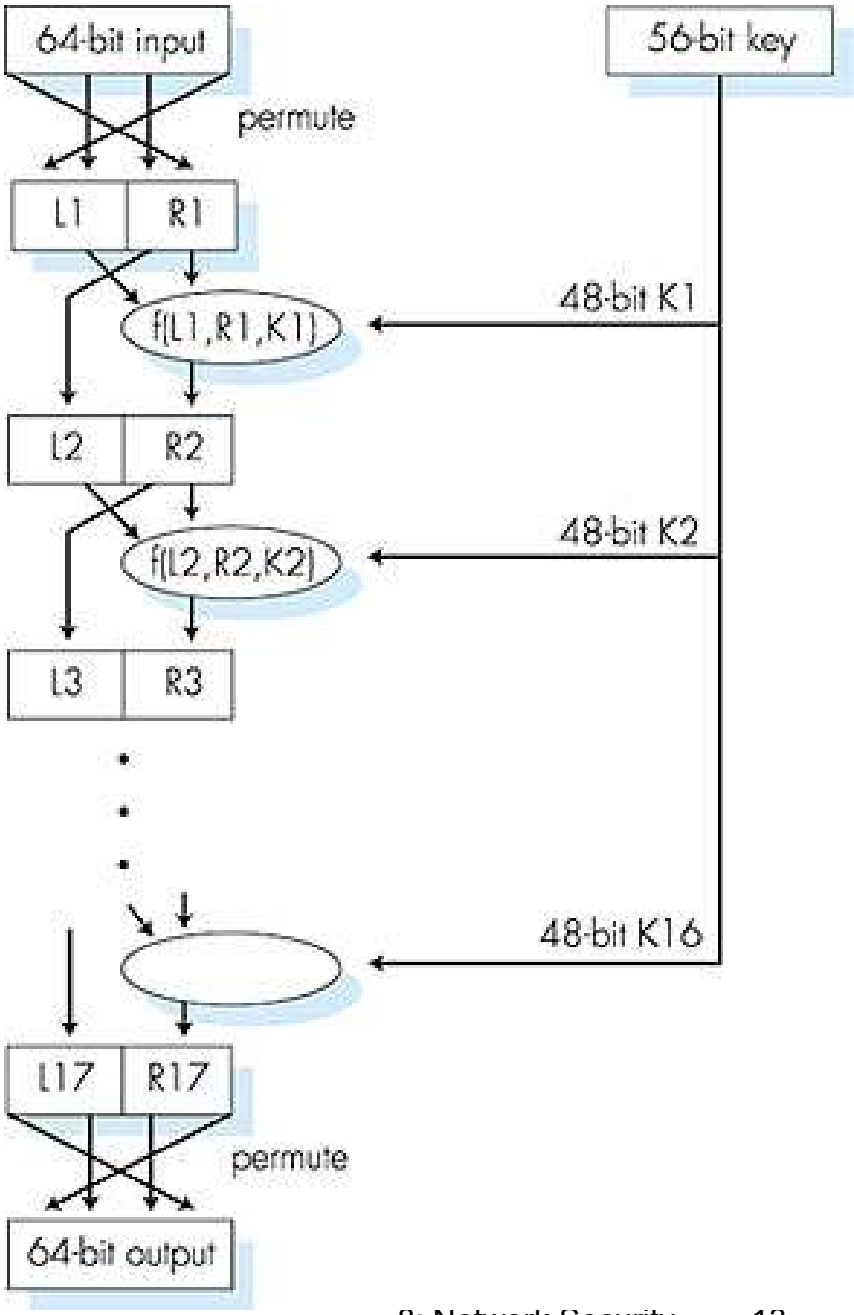
对称密钥加密学:DES

DES operation

初始替换

16 轮一样的函数应用，每一轮使用的不同的48bit密钥

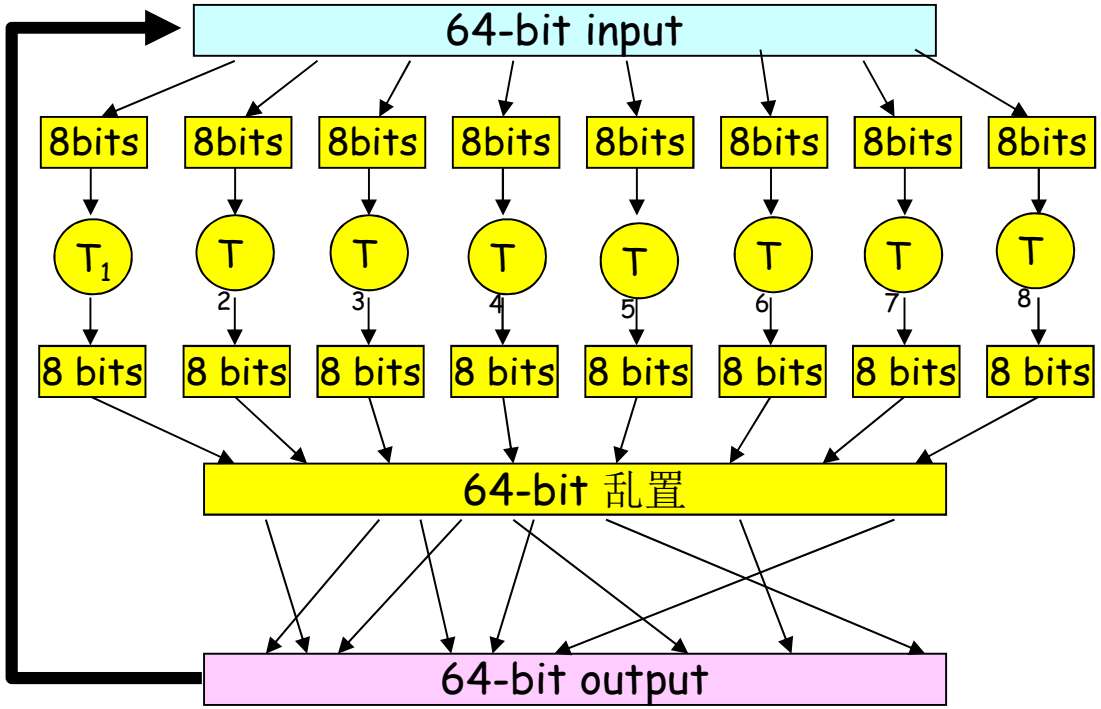
最终替换



AES: Advanced Encryption Standard

- 新的对称 密钥NIST标准(Nov. 2001) 用于替换 DES
- 数据128bit成组加密
- 128, 192, or 256 bit keys
- 穷尽法解密如果使用1秒钟破解 DES, 需要花149万亿年破解AES

块密码

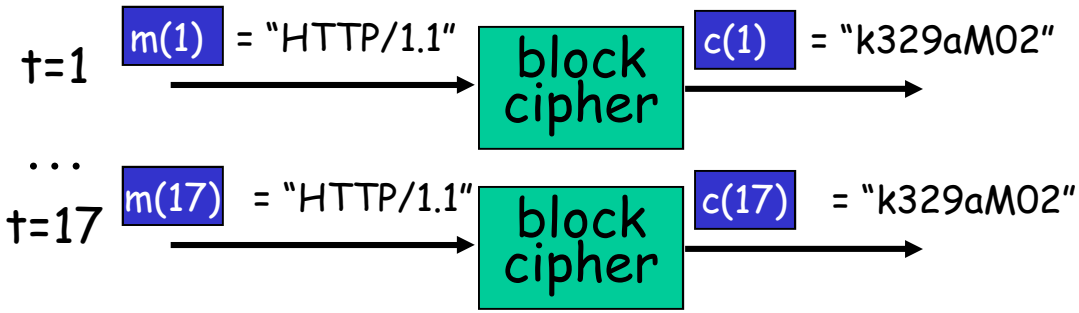


- 一个循环：一个输入bit影响8个输出bit

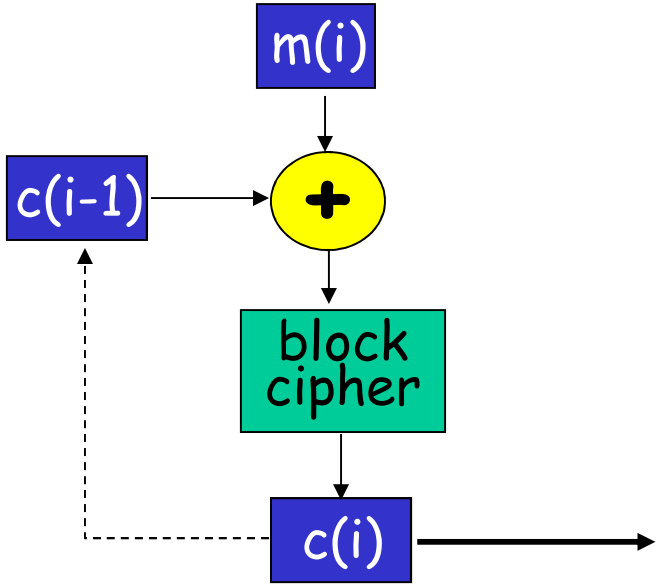
- 多重循环: 每个输入比特影响所有的输出bit
- 块密码：DES, 3DES, AES

密码块链

- 密码块：如果输入块重复，将会得到相同的密文块



- *密码块链：* 异或第i轮输入 m(i), 与前一轮的密文, c(i-1)
  - c(0) 明文传输到接收端
  - what happens in "HTTP/1.1" scenario from above?



公开密钥密码学

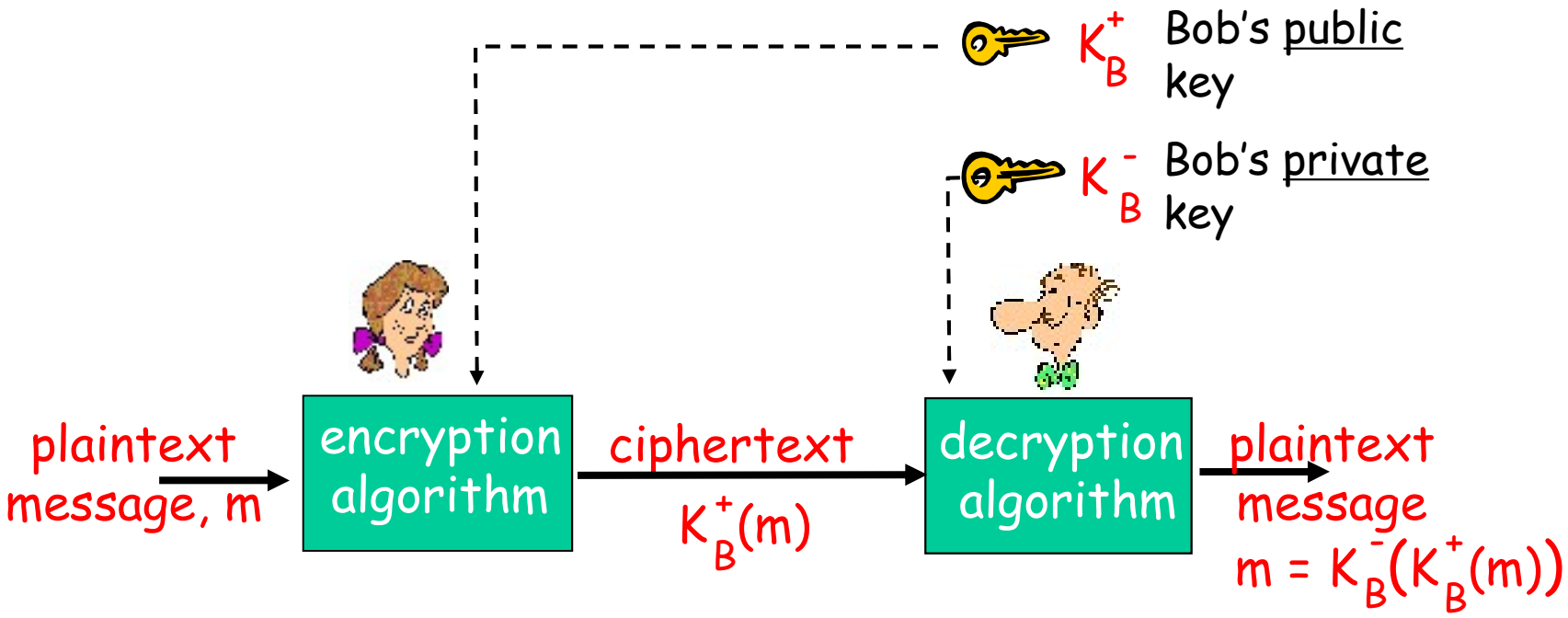
对称密钥密码学

- 需要发送方和接收方对共享式对称密钥达成一致
- Q: 但是他们如何第一次达成一致 (特别是他们永远不可能见面的情况下)?

公开密钥密码学

- 完全不同的方法 [Diffie-Hellman76, RSA78]
- 发送方和接收方无需共享密钥
- *一个实体的公钥公诸于众*
- *私钥*只有他自己知道

公开密钥密码学


K_B^+ Bob's public key
K_B^- Bob's private key
plaintext message, m
encryption algorithm
ciphertext
$K_B^+(m)$
decryption algorithm
plaintext message
$m = K_B^-(K_B^+(m))$

公开密钥加密算法

要求:

① 需要 $K_B^+(\cdot)$ 和 $K_B^-(\cdot)$，满足

$$K_B^-(K_B^+(m)) = m$$

② 给定一个公钥K_B^+推出私钥K_B^-计算上不可行

RSA: Rivest, Shamir, Adelson algorithm

RSA: 选择密钥

1. 选择2个很大的质数 *p*, *q*.
   (e.g., 1024 bits each)

2. 计算 *n* = *pq*, *z* = (*p*-1)(*q*-1)

3. 选择一个*e* (要求 *e*<*n*) 和z 没有一个公共因子，互素
   ("relatively prime").

4. 选择 *d* 使得*ed*-1 正好能够被*z*整除.
   (也就是： *ed* mod *z* = 1).

5. 公钥$\underbrace{(n,e)}_{K_B^+}$. 私钥 $\underbrace{(n,d)}_{K_B^-}$.

RSA: 加密,解密

0. 给定按照上述算法得到的 (n,e) 和 (n,d)

1. 加密一个bit模式, m, 如此计算：

   $c = m^e \bmod n$ (i.e., m^e除以n的余数)

2. 对接收到的密文c解密，如此计算

   $m = c^d \bmod n$ (i.e., c^e除以n的余数)

Magic happens!

$$m = \underbrace{(m^e \bmod n)}_{c}{}^d \bmod n$$

RSA 例子:

Bob 选择 $p=5$, $q=7$. 因此 $n=35$, $z=24$.

$e=5$ (so e, z 互素).

$d=29$ (so $ed-1$ 能够被 z整除).

encrypt:

letter	m	m^e	$c = m^e \bmod n$
l	12	1524832	17

decrypt:

c	c^d	$m = c^d \bmod n$	letter
17	48196857210675091509141182522307169 7	12	l

RSA: 为什么

$$m = (m^e \bmod n)^d \bmod n$$

一个有用的数论定理: 如果 p,q 都是素数，$n = pq$, *那么*:

$$x^y \bmod n = x^{y \bmod (p-1)(q-1)} \bmod n$$

$$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n$$

$$= m^{ed \bmod (p-1)(q-1)} \bmod n$$

(使用上述定理)

$$= m^1 \bmod n$$

(因为我们选择ed 使得正好被z 除余1)

$$= m$$

RSA: 另外一个重要的特性

下面的特性将在后面非常有用

$$\underbrace{K_B^-(K_B^+(m))}_{\text{先用公钥，然后用私钥}} = m = \underbrace{K_B^+(K_B^-(m))}_{\text{先用私钥，然后用公钥}}$$

结果一致!

解密的几种类型

- 加密算法已知，求密钥
- 加密算法和密钥均不知道

- 唯密文攻击
- 已知明文攻击
  - 已经知道部分密文和明文的对应关系
- 选择明文攻击
  - 攻击者能够选择一段明文，并得到密文

提纲

8.1 什么是网络安全？
8.2 加密原理
8.3 认证
8.4 报文完整性
8.5 密钥分发和证书
8.6 访问控制：防火墙
8.7 攻击和对策
8.8 各个层次的安全性

认证

目标: Bob需要Alice证明她的身份

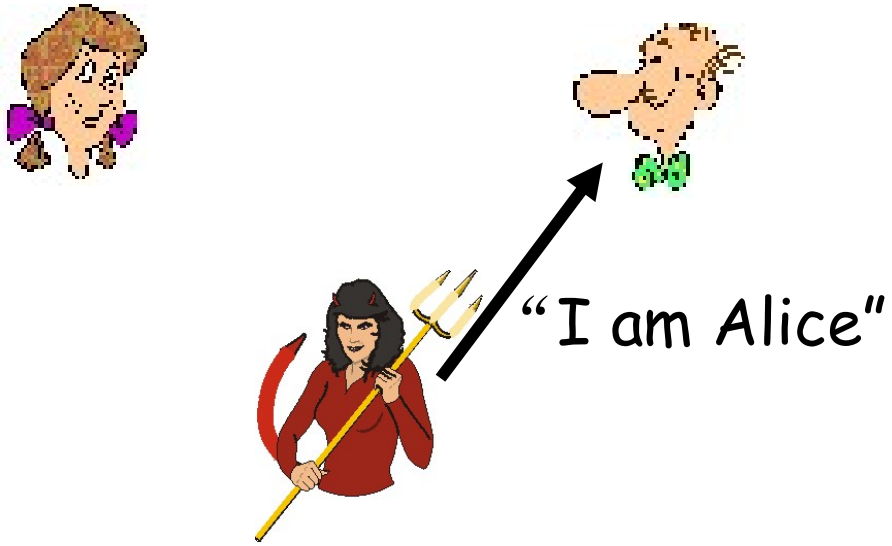
Protocol ap1.0: Alice说 “I am Alice”

失败的情景？

认证

目标: Bob需要Alice证明她的身份

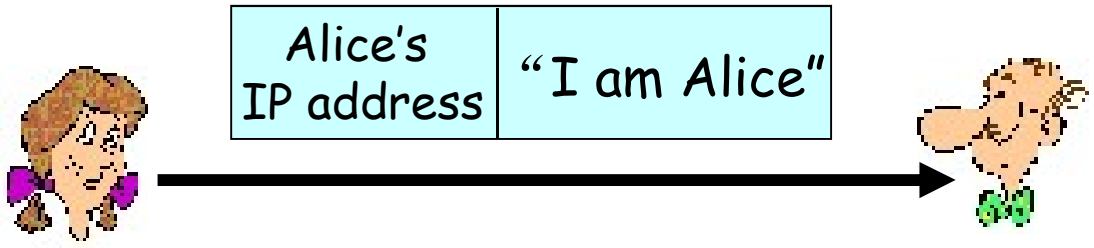
Protocol ap1.0: Alice 说 “I am Alice”



在网络上Bob看不到Alice, 因此Trudy可以简单地声称她是 Alice

认证:重新尝试

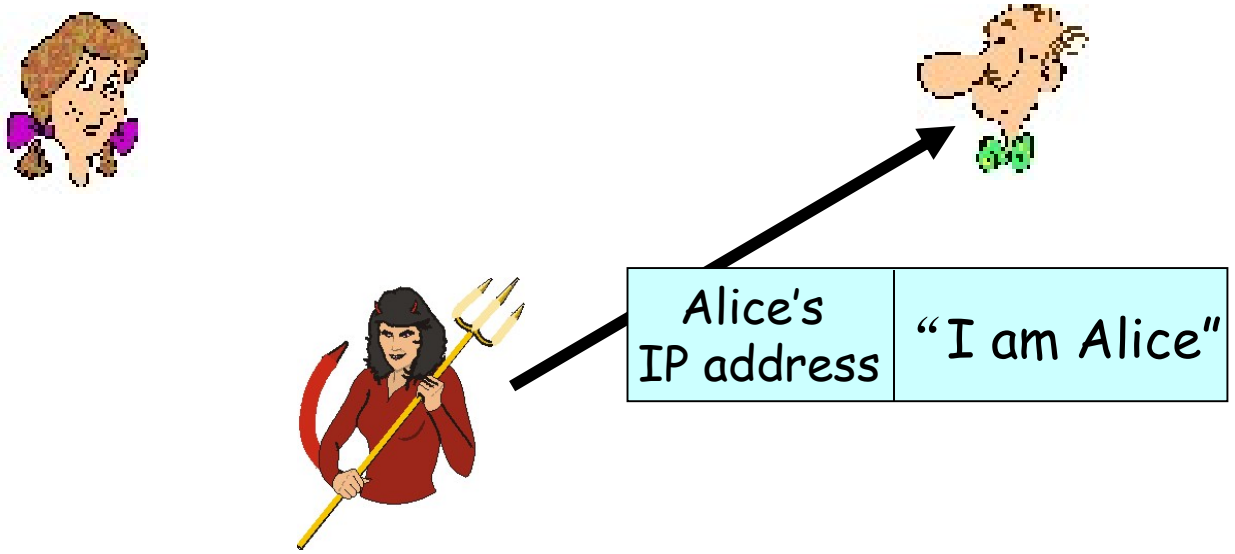
Protocol ap2.0: Alice 说 “I am Alice”，在她发送的IP数据包中包括了她的IP地址



失败的情景？

认证:重新尝试

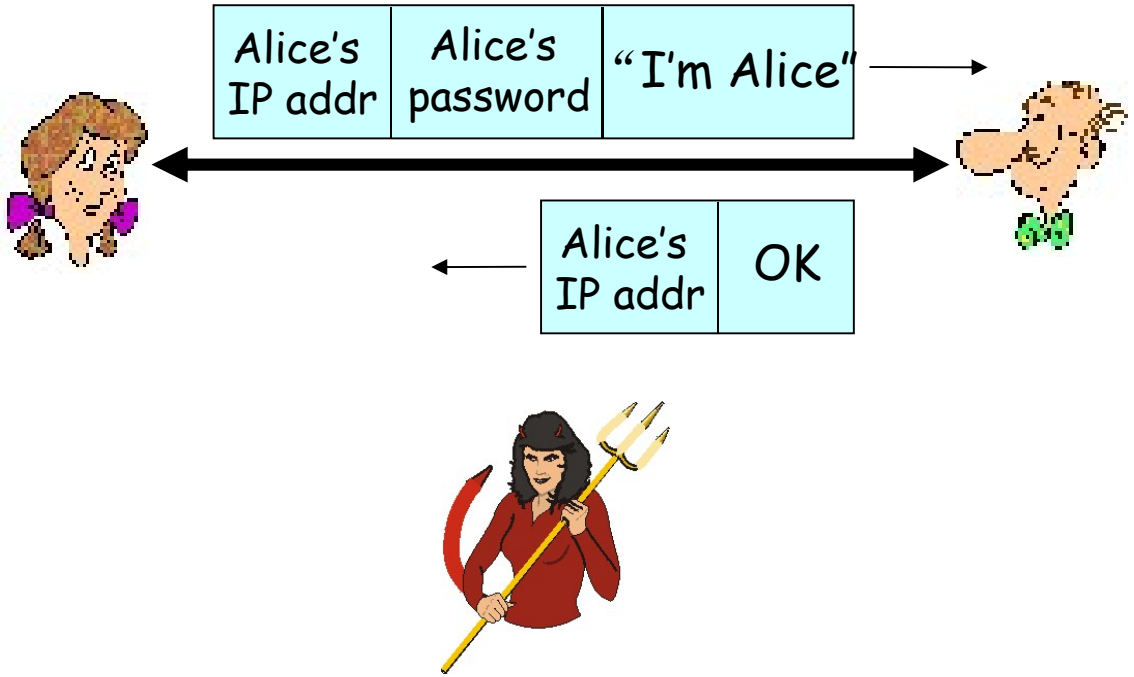
Protocol ap2.0: Alice 说 “I am Alice”，在她发送的IP数据包中包括了她的IP地址



Trudy可以生成一个分组，包括伪造的Alice的地址

认证:重新尝试

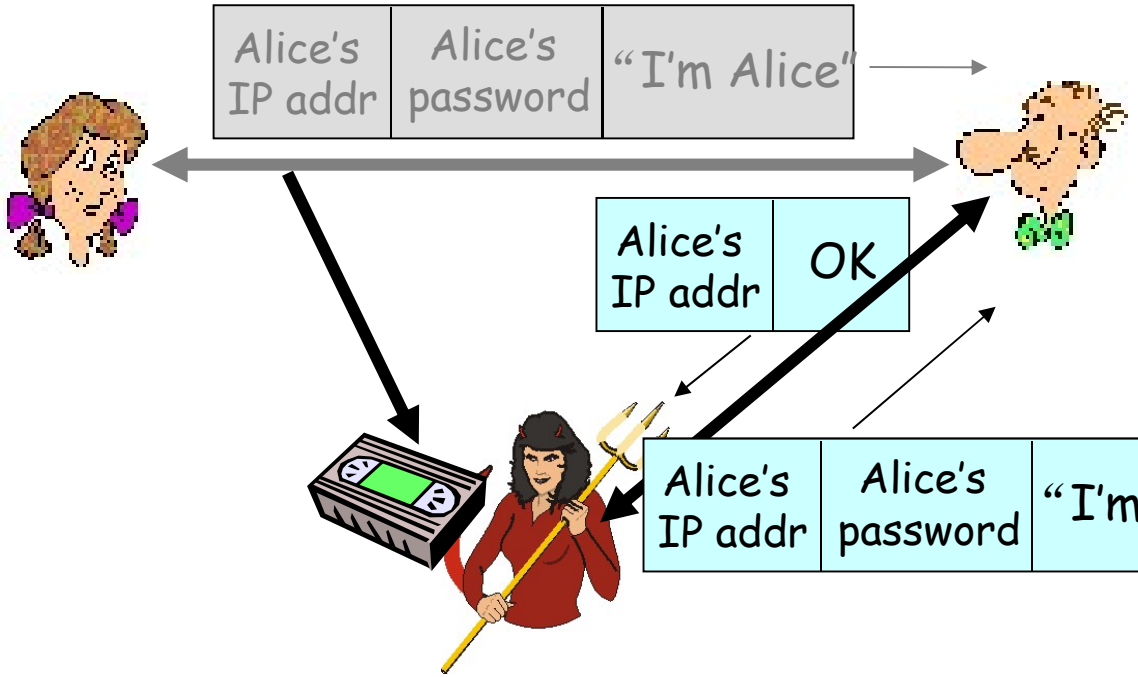
Protocol ap3.0: Alice 说 “I am Alice”， 而且传送她的密码来证明.



Failure scenario??

认证:重新尝试

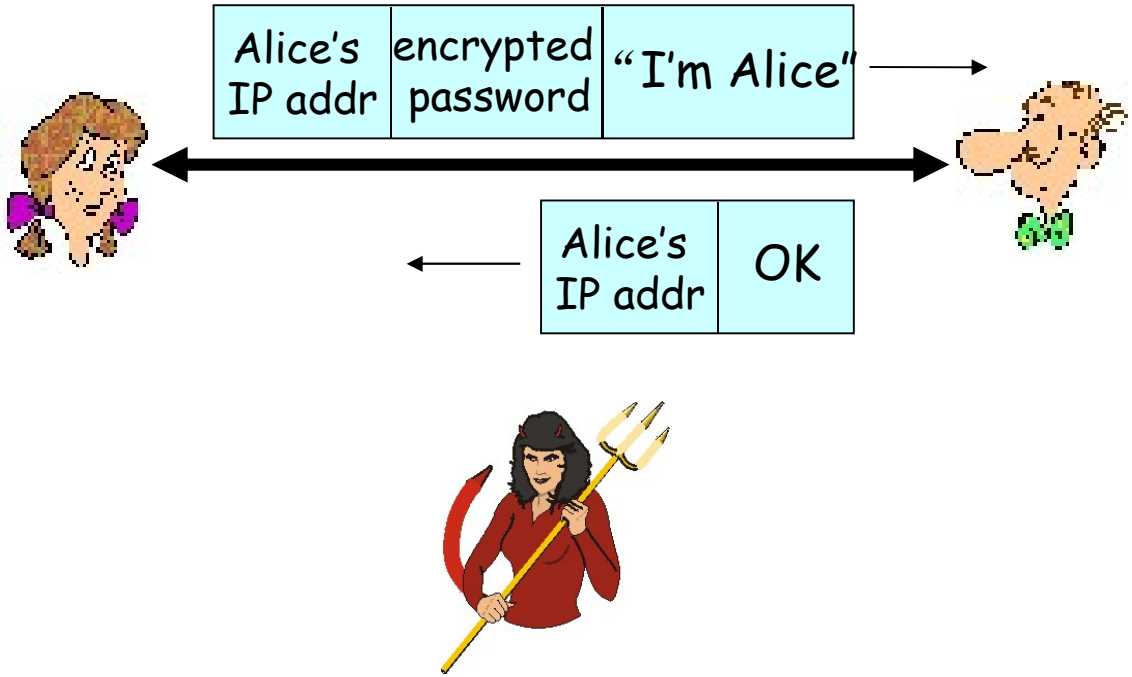
Protocol ap3.0: Alice 说 “I am Alice”， 而且传送她的密码来证明



重放攻击playback attack: Trudy记录 Alice 的 分组，事后向Bob重放

认证:重新尝试

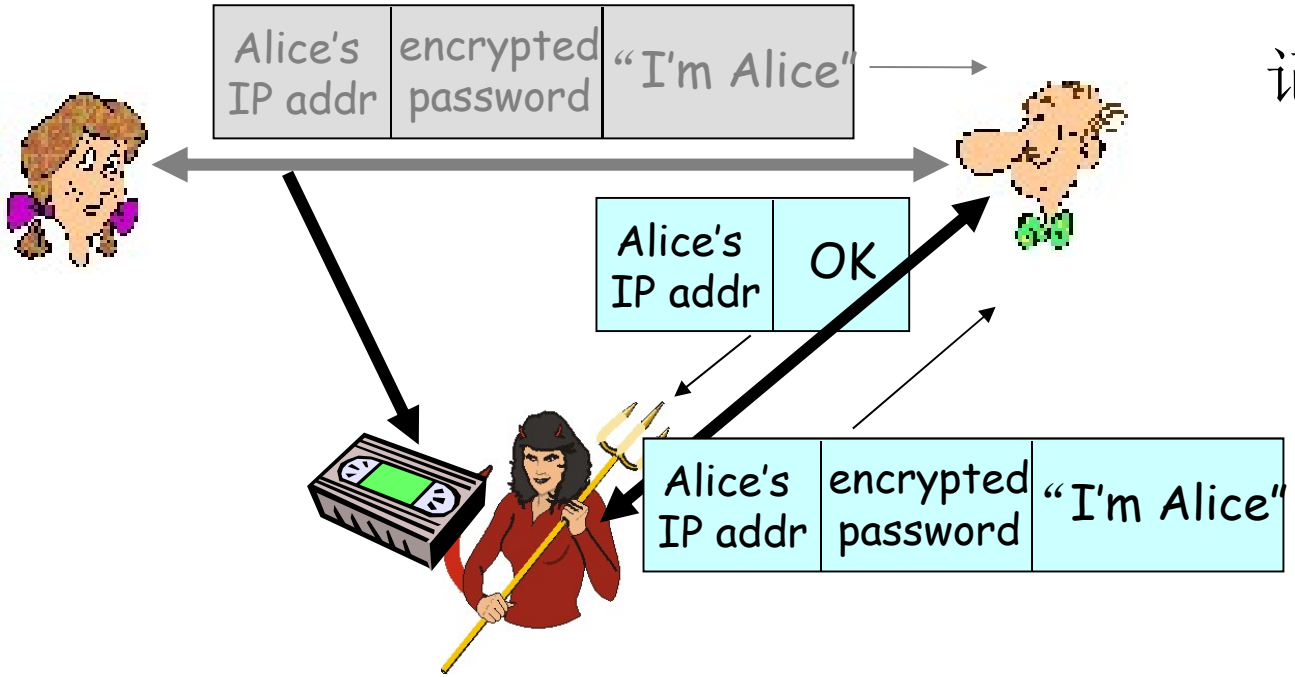
Protocol ap3.1: Alice 说 “I am Alice”， 而且传送她的加密之后的密码来证明



失败的情景？

认证:重新尝试

Protocol ap3.1: Alice 说 “I am Alice”， 而且传送她的加密之后的密码来证明.



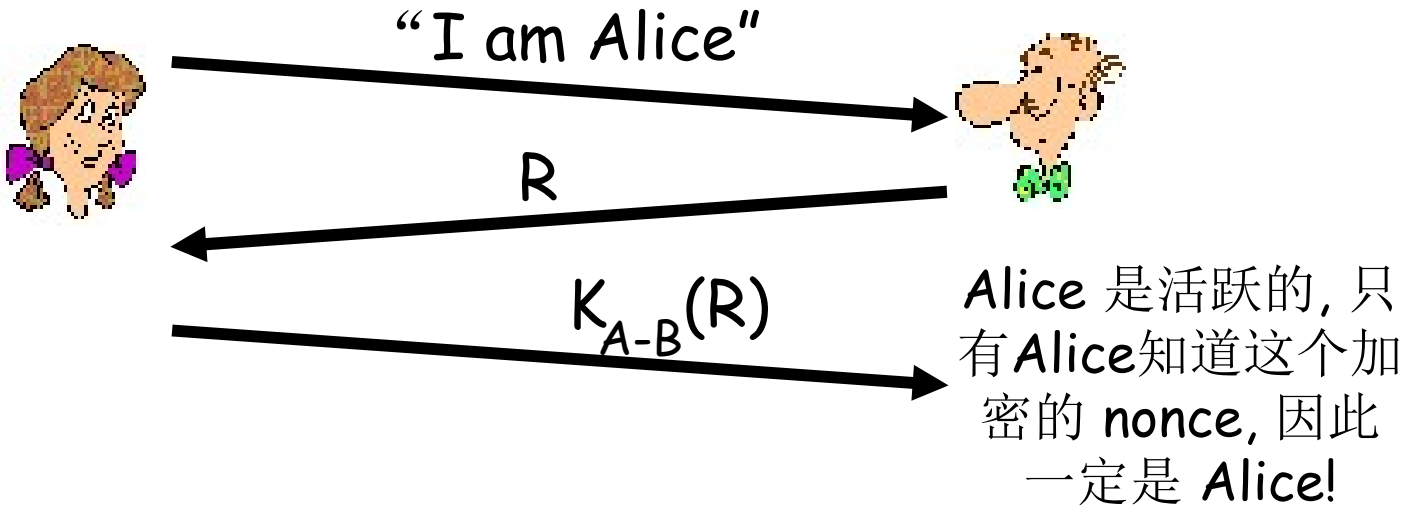
记录，重放仍然有效

认证:重新尝试

目标: 避免重放攻击

Nonce: 一生只用一次的整数 (R)

ap4.0: 为了证明Alice的活跃性, Bob发送给Alice一个nonce, R. Alice 必须返回加密之后的R，使用双方约定好的key



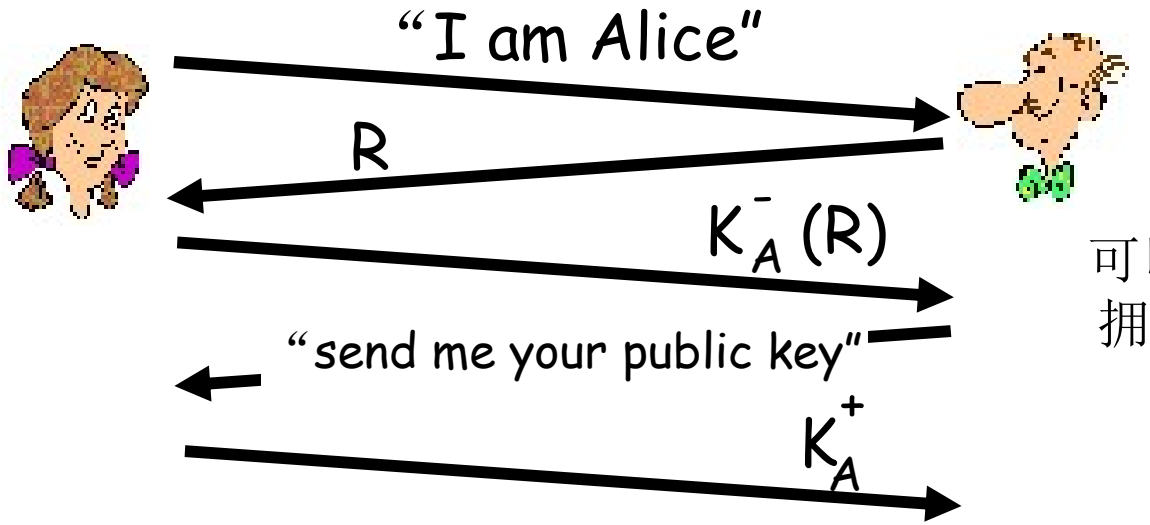
会失败吗，有问题吗?

认证: ap5.0

ap4.0 需要双方共享一个对称式的密钥

- 是否可以通过公开密钥技术进行认证呢?

ap5.0: 使用nonce,公开密钥加密技术



Bob 计算

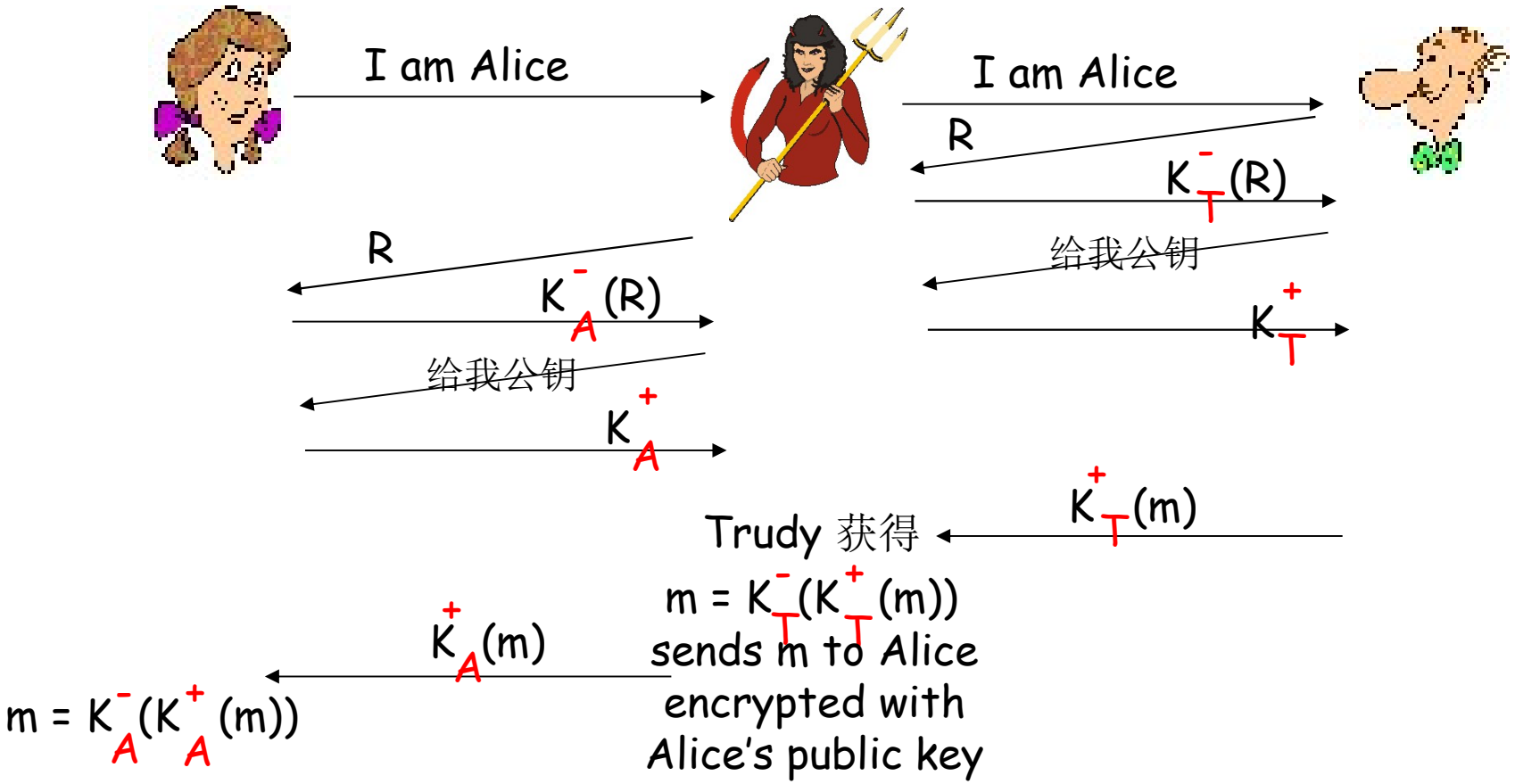
$K_A^+(K_A^-(R)) = R$

可以知道只有 Alice 能够拥有这个私钥, 因此才能够得到

$K_A^+(K_A^-(R)) = R$

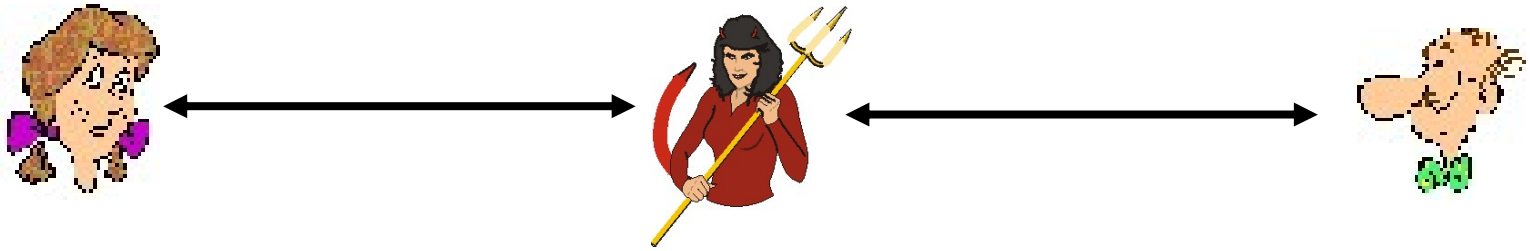
ap5.0: 安全漏洞

中间攻击: Trudy 在 Alice (to Bob)和 Bob之间 (to Alice)



ap5.0: 安全漏洞

中间攻击: Trudy 在 Alice (to Bob)和 Bob之间 (to Alice)



难以检测:

- Bob收到了Alice发送的所有报文, 反之亦然. (e.g., so Bob, Alice一个星期以后相见，回忆起以前的会话)
- 问题时Trudy也接收到了所有的报文!

提纲

8.1 什么是网络安全？

8.2 加密原理

8.3 认证

8.4 报文完整性

8.5 密钥分发和证书

8.6 访问控制：防火墙

8.7 攻击和对策

8.8 各个层次的安全性

数字签名

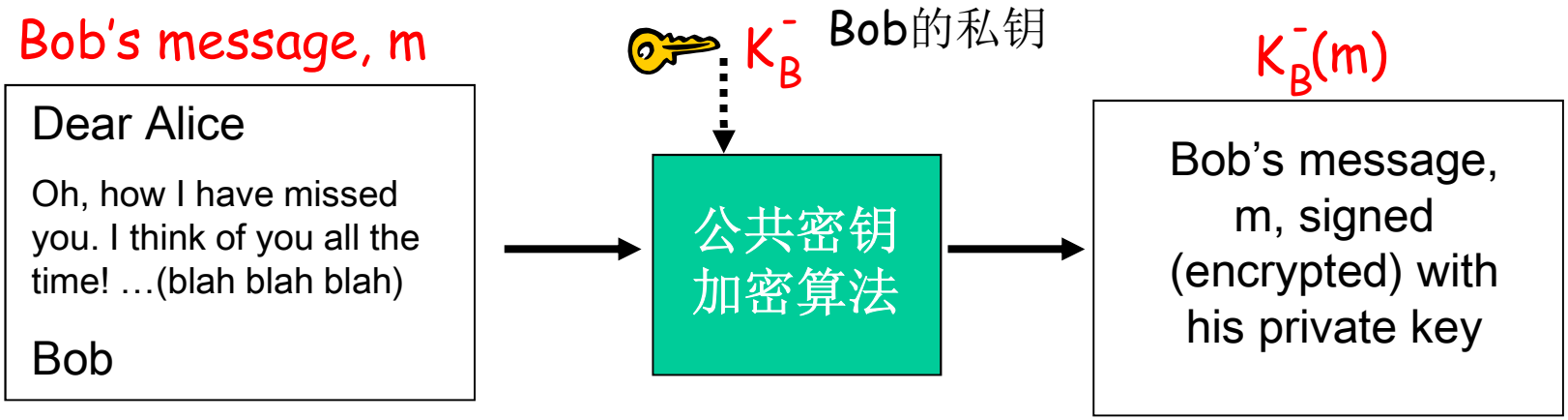
数字签名类比于手写签名

- 发送方 (Bob) 数字签署了文件，前提是他(她)是文件的拥有者/创建者.
- 可验证性，不可伪造性，不可抵赖性
  - **谁签署：**接收方 (Alice)可以向他人证明 是 Bob, 而不是其他人签署了这个文件 (包括Alice)
  - **签署了什么：**这份文件，而不是其它文件

数字签名

简单的对m的数字签名：

- Bob使用他自己的私钥对m进行了签署，创建数字签名 $K_B^-(m)$



数字签名（续）

- 假设Alice收到报文m, 以及数字签名$K_B^-(m)$
- Alice 使用Bob的公钥K_B^+对$K_B^-(m)$进行验证， 判断$K_B^+(K_B^-(m)) = m$是否成立.
- 如 $K_B^+(K_B(m)) = m$成立, 那么签署这个文件的人一定拥有Bob的私钥.

Alice 可以验证:

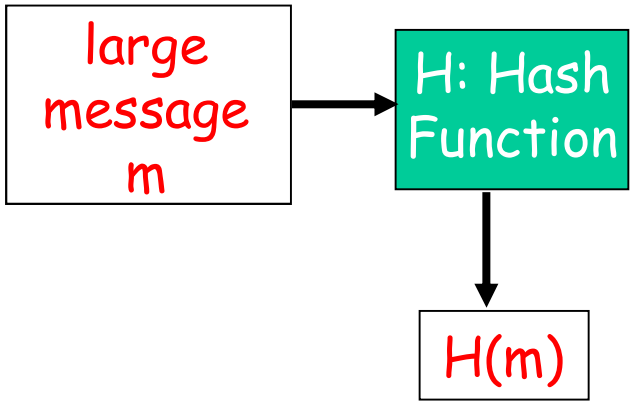
- ✓ Bob 签署了m.
- ✓ 不是其他人签署了m.
- ✓ Bob签署了m 而不是m'.

不可抵赖性:

- ✓ Alice可以拿着m,以及数字签名$K_B^-(m)$到法庭上，来证明是Bob签署了这个文件 m.

报文摘要

对长报文进行公开密钥加密算法的实施需要耗费大量的时间



Goal: 固定长度，容易计算的 "fingerprint"

- 对m使用散列函数H，获得固定长度的 报文摘要 H(m).

散列函数的特性：

- 多对1
- 结果固定长度
- 给定一个报文摘要x, 反向计算出原报文在计算上是不可行的x = H(m)

Internet校验和：弱的散列函数

Internet 校验和拥有一些散列函数的特性:

- ✓ 产生报文m的固定长度的摘要 (16-bit sum)
- ✓ 多对1的

但是给定一个散列值，很容易计算出另外一个报文具有同样的散列值:

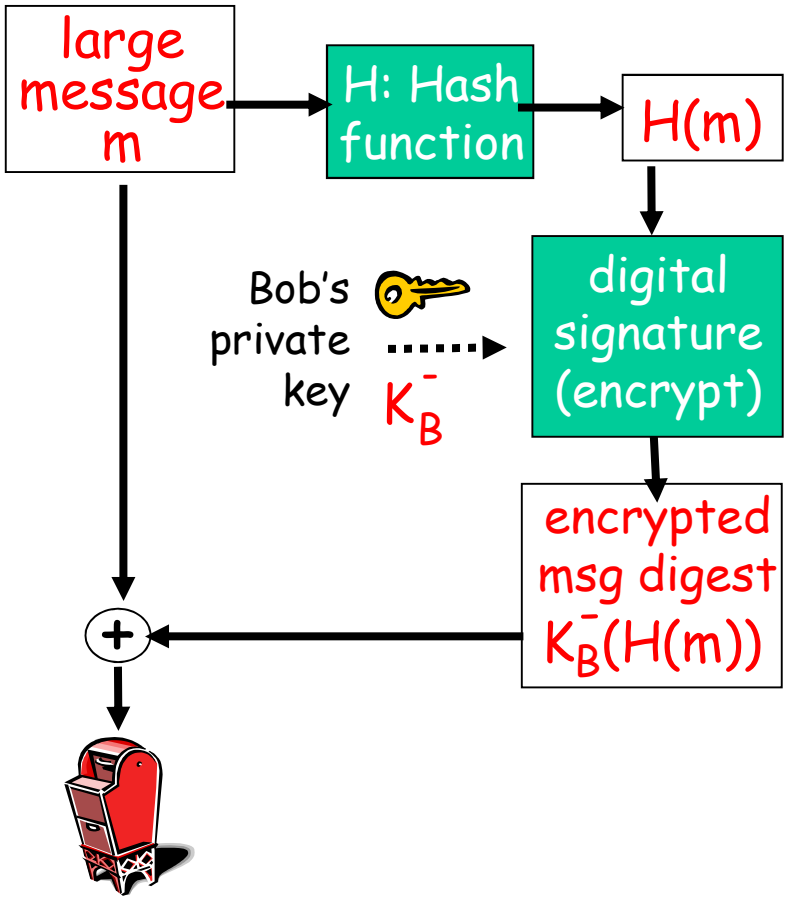
message	ASCII format
I O U 1	49 4F 55 31
0 0 . 9	30 30 2E 39
9 B O B	39 42 D2 42
	B2 C1 D2 AC

message	ASCII format
I O U 9	49 4F 55 39
0 0 . 1	30 30 2E 31
9 B O B	39 42 D2 42
	B2 C1 D2 AC

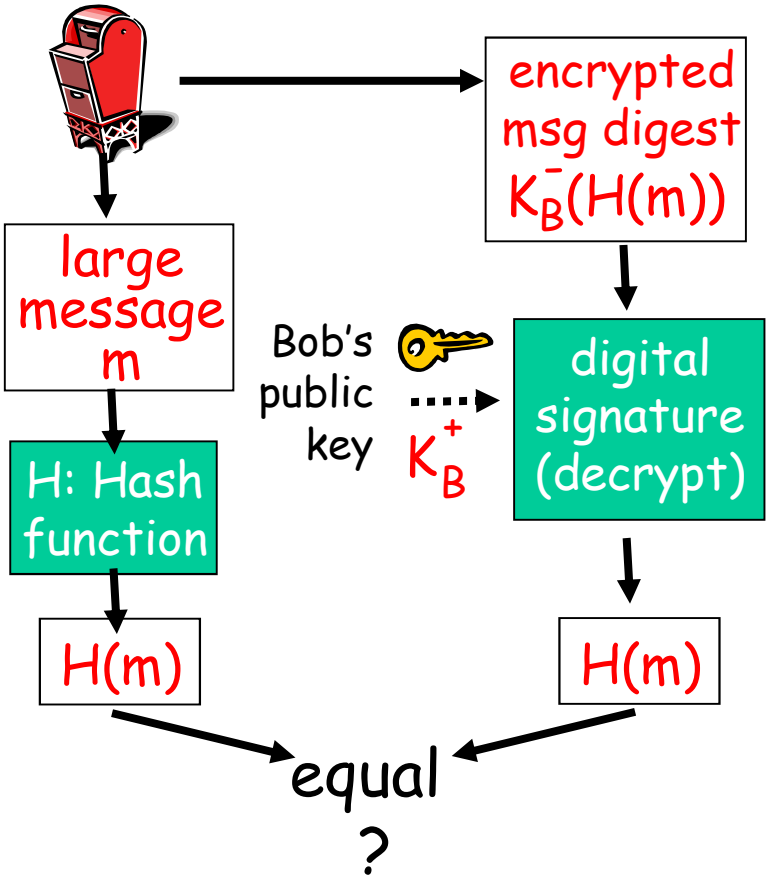
不同的报文
但是相同的校验和!

数字签名 = 对报文摘要进行数字签署

Bob 发送数字签名的报文:



Alice校验签名和报文完整性:



散列函数算法

- MD5散列函数(RFC 1321)被广泛地应用
  - 4个步骤计算出128-bit的报文摘要
  - 给定一个任意的128-bit串x, 很难构造出一个报文m具有相同的摘要x.
- SHA-1也被使用.
  - US标准 [NIST, FIPS PUB 180-1]
  - 160-bit报文摘要

提纲

8.1 什么是网络安全?
8.2 加密原理
8.3 认证
8.4 报文完整性
8.5 密钥分发和证书
8.6 访问控制：防火墙
8.7 攻击和对策
8.8 各个层次的安全性

可信赖中介

对称密钥问题

- 相互通信的实体如何分享对称式的密钥?

解决办法:

- trusted key distribution center (KDC) 在实体之间扮演可信赖中介的角色

公共密钥问题

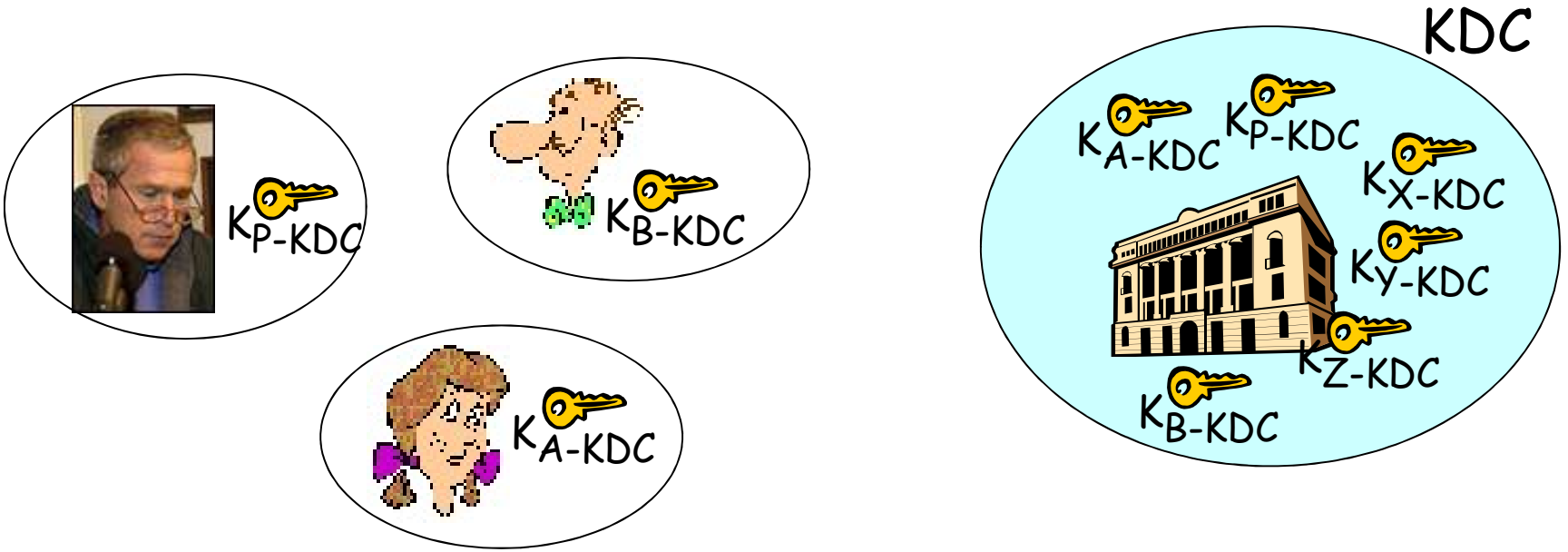
- 当Alice获得Bob的公钥 (from web site, e-mail, diskette), 她如何知道就是Bob的public key, 而不是Trudy的?

解决办法:

- 可信赖的certification authority (CA)

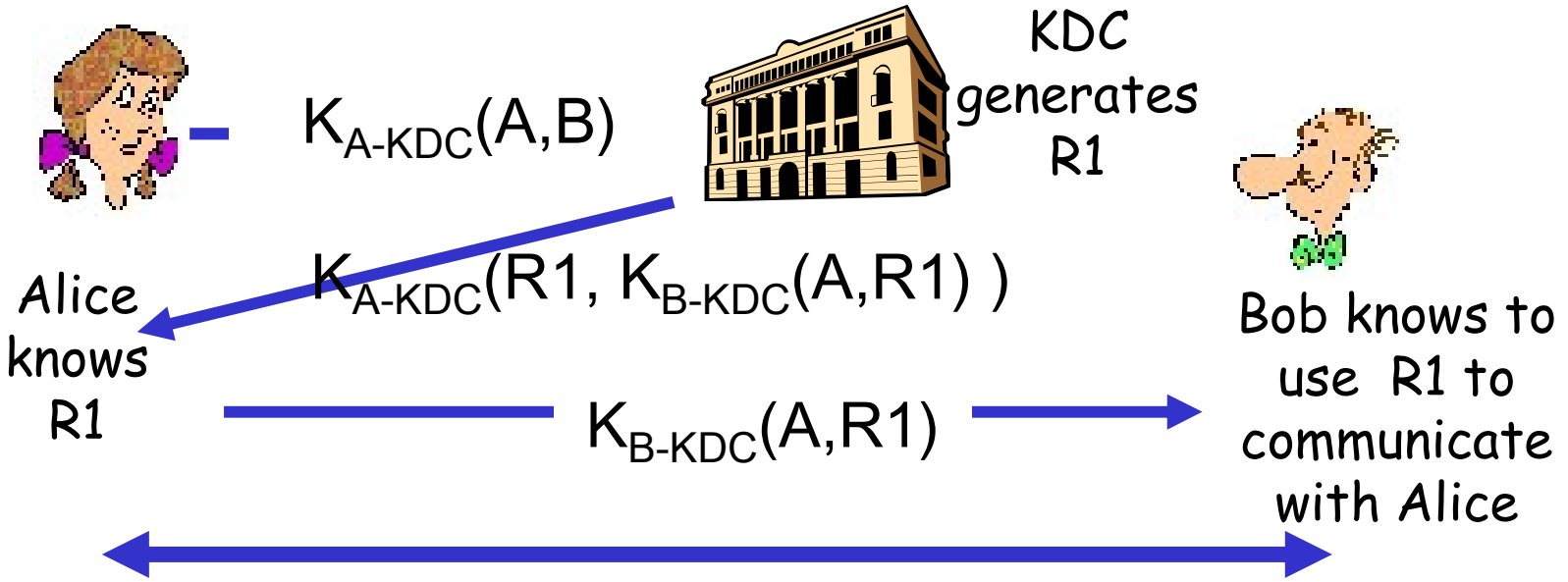
Key Distribution Center (KDC)

- Alice, Bob 需要分享对称式密钥.
- KDC: 服务器和每一个注册用户都分享一个对称式的密钥 (many users)
- Alice, Bob在和KDC通信的时候，知道他们自己的对称式密钥 $K_{A\text{-}KDC}$ $K_{B\text{-}KDC}$.



Key Distribution Center (KDC)

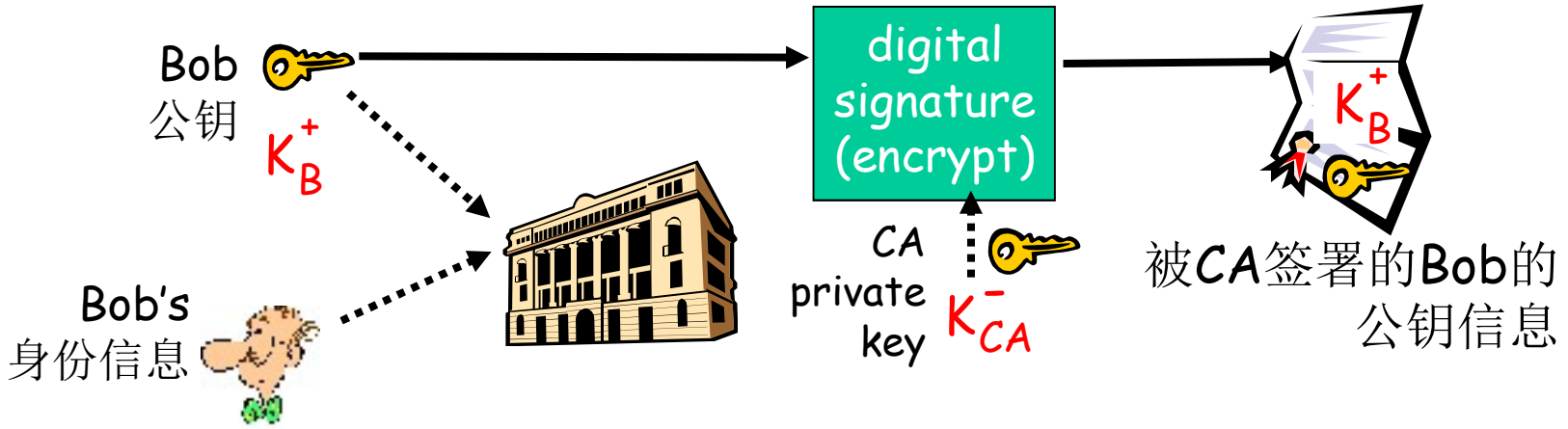
Q: KDC如何使得 Bob和Alice在和对方通信前，就对称式会话密钥达成一致?



Alice 与Bob通信: 使用R1作为对称式的会话密钥

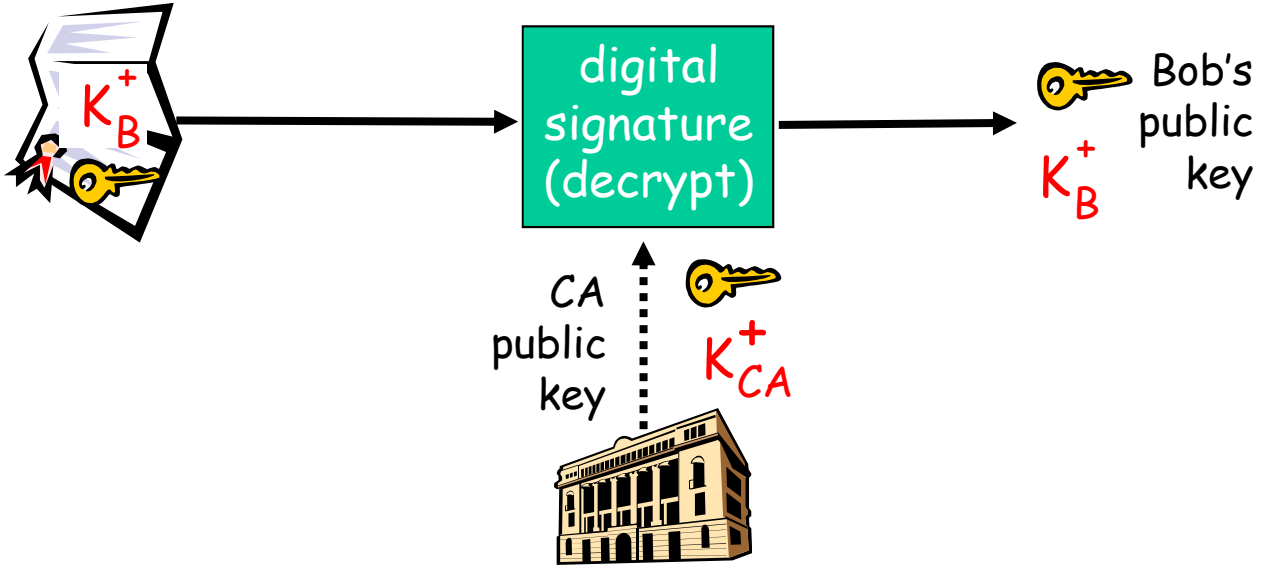
Certification Authorities

- Certification authority (CA): 将每一个注册实体E和他的公钥捆绑.
- E (person, router) 到CA那里注册他的公钥.
  - E 提供给CA，自己身份的证据 "proof of identity"
  - CA创建一个证书，捆绑了 实体信息和他的公钥.
  - Certificate包括了E的公钥，而且是被CA签署的（被CA用自己的私钥加了密的） - CA说 "this is E's public key"



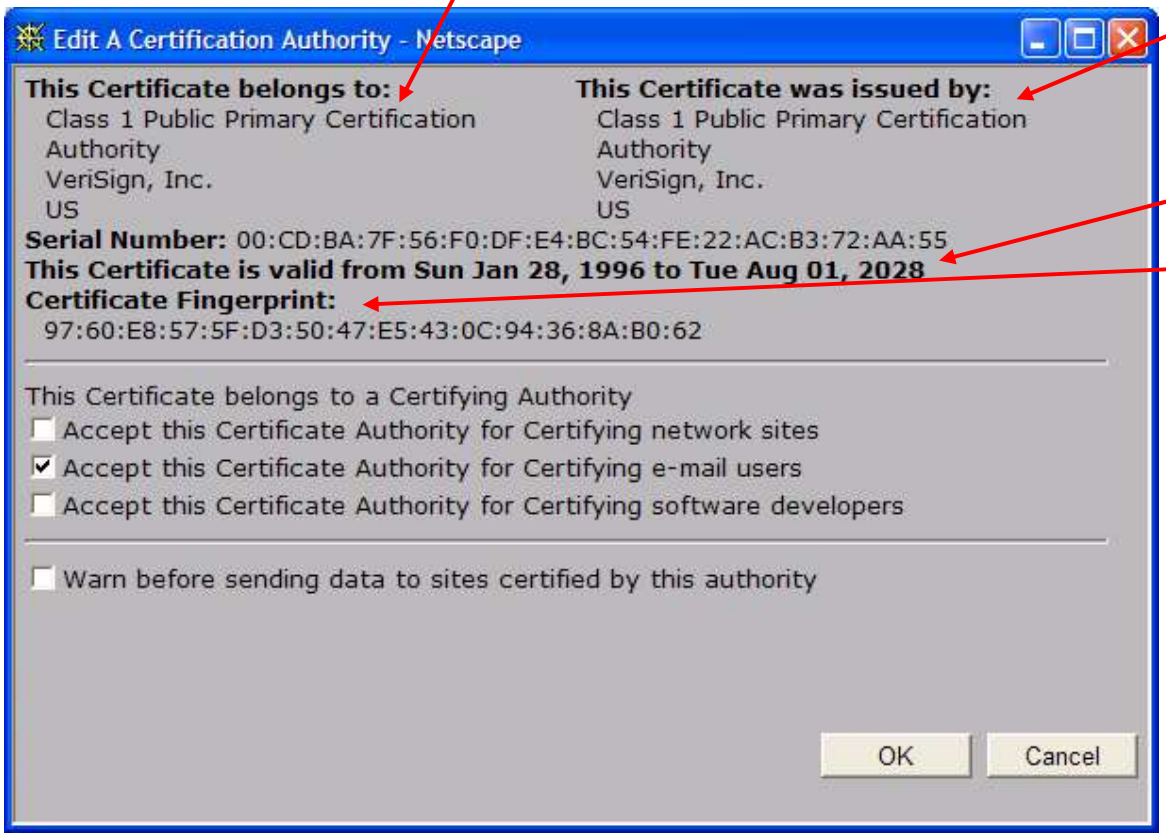
Certification Authorities

- 当Alice需要拿到Bob公钥
  - 获得Bob的证书certificate (从Bob或者其他地方).
  - 对Bob的证书，使用CA的公钥来验证



证书包括:

- 串号 (证书发行者唯一)
- 证书拥有者信息，包括算法和密钥值本身 (不显示出来)



- 证书发行者信息
- 有效日期
- 颁发者签名

信任树

- 根证书：根证书是未被签名的公钥证书或自签名的证书
  - 拿到一些CA的公钥
  - 渠道：安装OS自带的数字证书；从网上下载，你信任的数字证书
- 信任树：
  - 信任根证书CA颁发的证书，拿到了根CA的公钥
    - 信任了根
  - 由根CA签署的给一些机构的数字证书，包含了这些机构的数字证书
  - 由于你信任了根，从而能够可靠地拿到根CA签发的证书，可靠地拿到这些机构的公钥

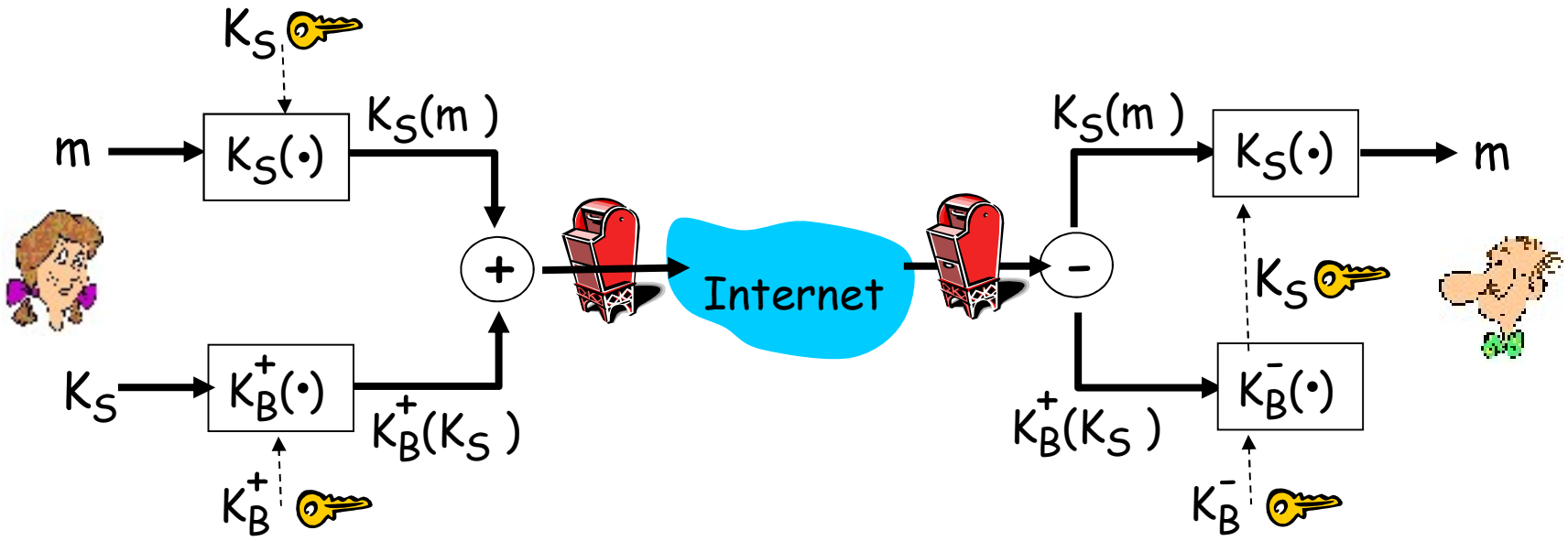
提纲

8.1 什么是网络安全？
8.2 加密原理
8.3 认证
8.4 报文完整性
8.5 密钥分发和证书
8.6 各个层次的安全性
- 8.8.1. 安全电子邮件
- 8.8.2. 安全套接字
- 8.8.3. IPsec
- 8.8.4. 802.11中的安全性

8.6 访问控制：防火墙
8.7 攻击和对策

安全电子邮件

- Alice 需要发送机密的报文 m 给Bob.

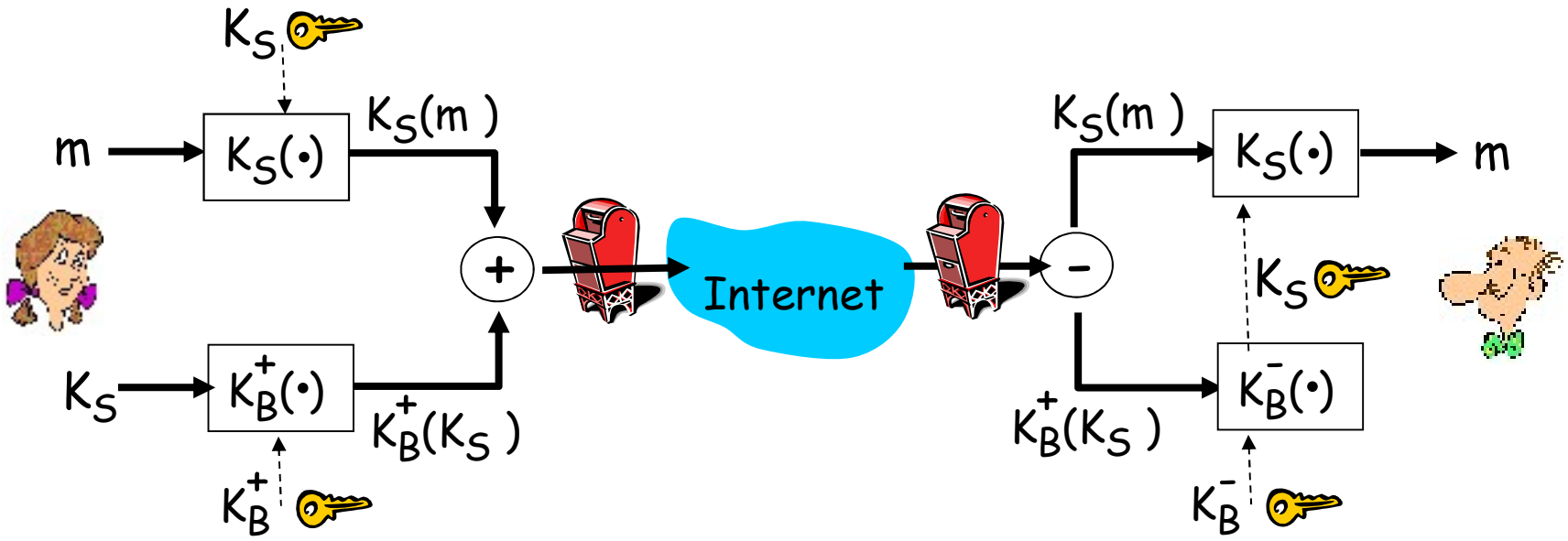


Alice:

- 产生随机的对称密钥, K_S.
- 使用K_S对报文加密(为了效率)
- 对 K_S 使用 Bob的公钥进行加密.
- 发送$K_S(m)$ 和$K_B(K_S)$ 给 Bob.

安全电子邮件

- Alice 需要发送机密的报文 m 给Bob.

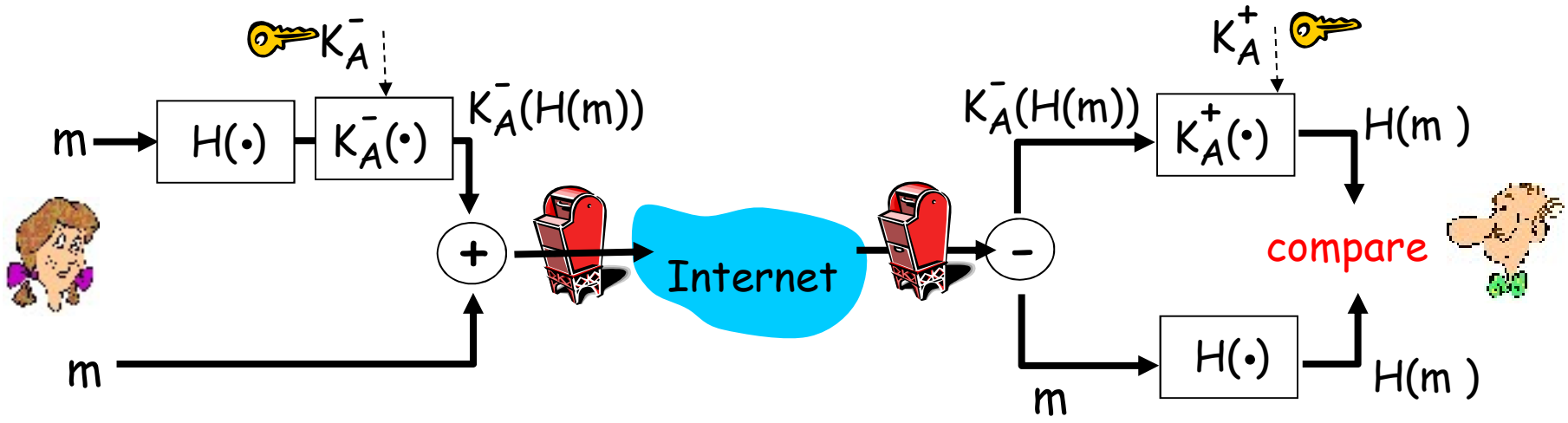


Bob:

- 使用自己的私钥解密 K_S
- 使用 K_S 解密 $K_S(m)$ 得到报文

安全电子邮件（续）

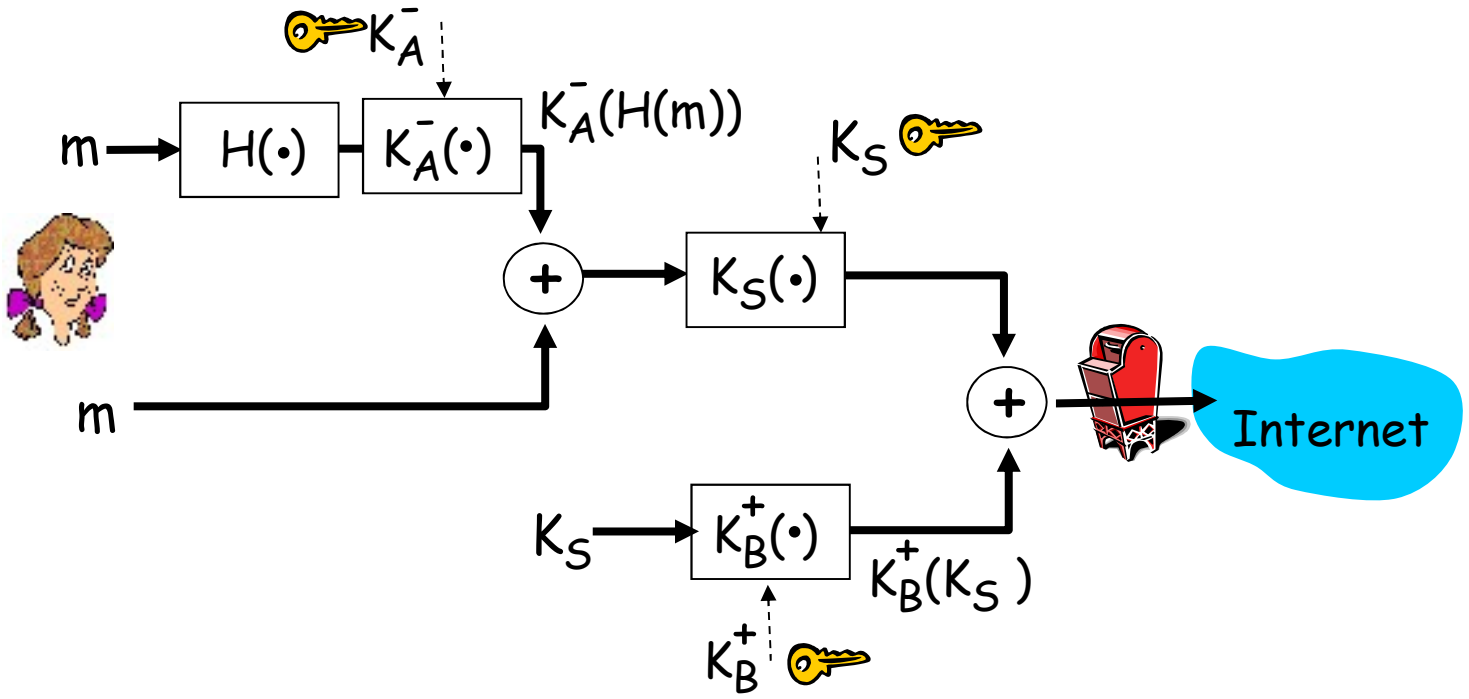
- Alice 需要提供源端的可认证性和报文完整性



- Alice 数字签署文件.
- 发送报文（明文）和 数字签名

安全电子邮件（续）

- Alice 需要提供机密性，源端可认证性和报文的完整性



Alice 使用了3个keys: 自己的私钥，Bob的公钥, 新产生出的对称式密钥

Pretty good privacy (PGP)

- Internet e-mail加密方案，事实上的标准.
- 使用前面讲述的：对称密钥加密，公开密钥加密，散列函数和数字签名.
- 能够提供机密性，源端的可认证性和报文完整性.
- 发明者, Phil Zimmerman, 是 3 年的犯罪调查的目标

A PGP signed message:

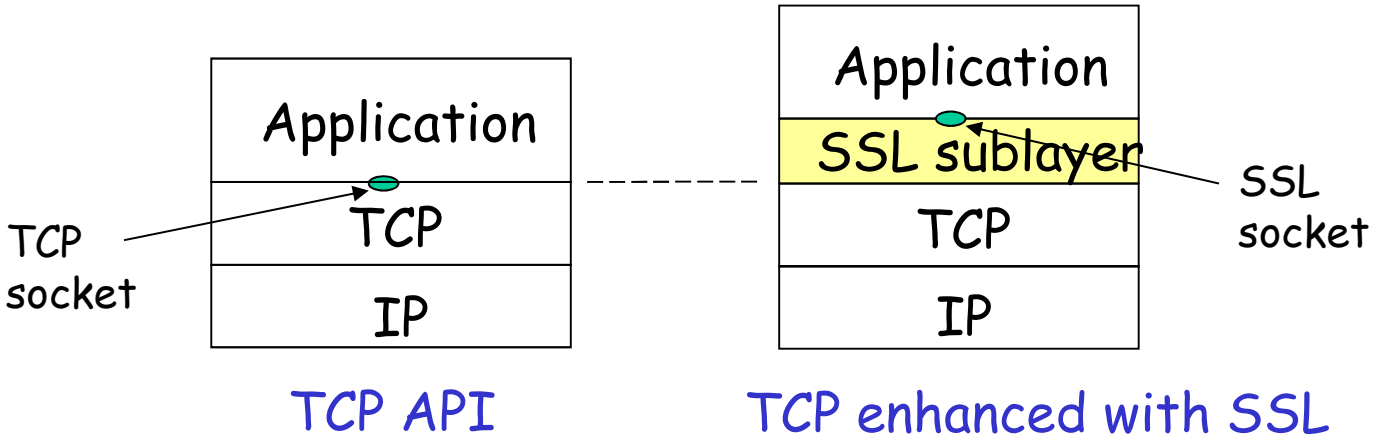
```
---BEGIN PGP SIGNED MESSAGE---
   -
Hash: SHA1

Bob:My husband is out of town
   tonight.Passionately
   yours, Alice

---BEGIN PGP SIGNATURE---
Version: PGP 5.0
Charset: noconv
yhHJRHhGJGhgg/12EpJ+lo8gE4vB3
   mqJhFEvZP9t6n7G6m5Gw2
---END PGP SIGNATURE---
```

Secure sockets layer (SSL)

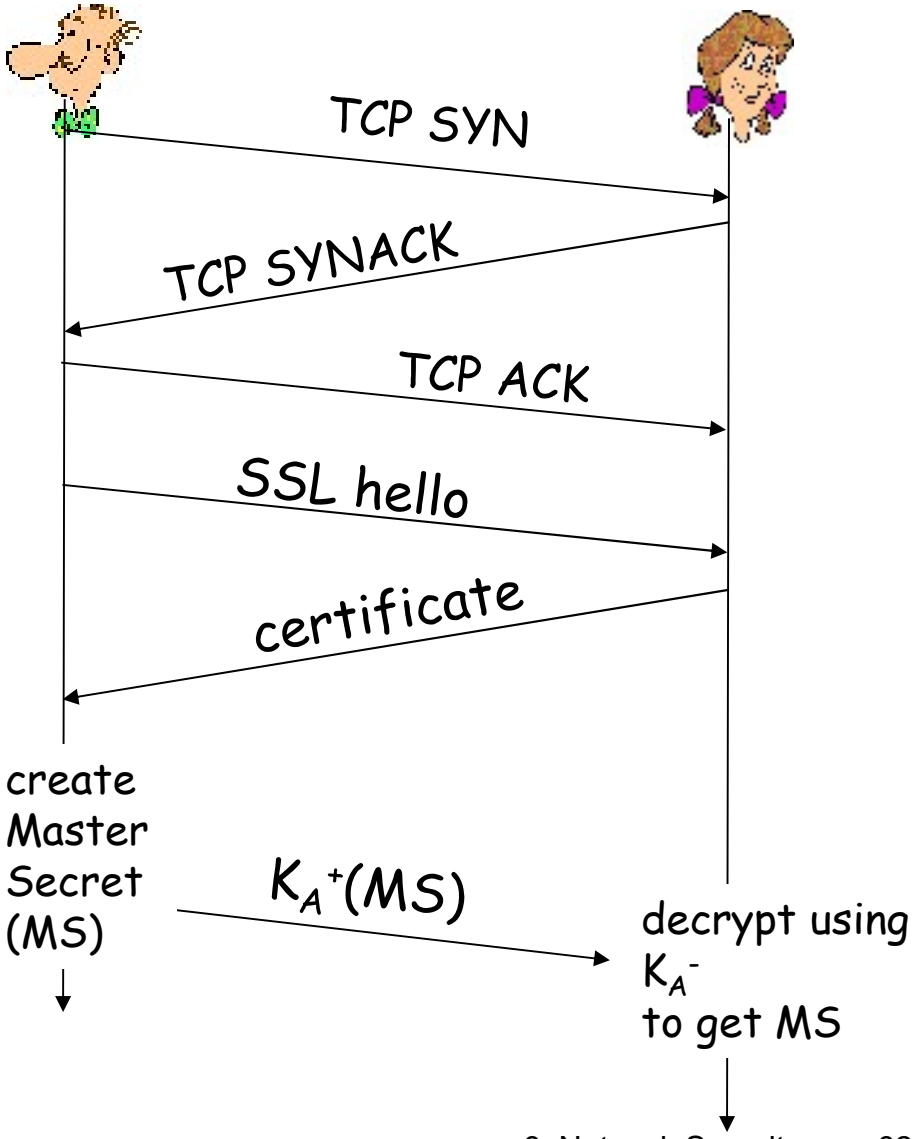
- 为使用SSL服务的、基于TCP的应用提供传输层次的安全性
  - e.g., 在WEB的浏览器和服务器之间进行电子商务的交易 (shttp)
- 所提供的安全服务:
  - 服务器的可认证性，数据加密，客户端的可认证性 (可选)



SSL: 3阶段

1. 握手:

- Bob 和Alice 建立TCP 连接
- 通过CA签署的证书认证 Alice的身份
- 创建，加密 (采用Alice的公钥), 传输主密钥给Alice
  - 不重数交换没有显示



SSL: 3阶段

2. 密钥导出:

- Alice, Bob采用共享的MS产生4个keys:
  - E_B: Bob->Alice 数据加密key
  - E_A: Alice->Bob数据加密key
  - M_B: Bob->Alice MAC（报文鉴别编码）key
  - M_A: Alice->Bob MAC key
- 加密和MAC算法在Bob, Alice之间协商
- 为什么要4个keys?
  - 更安全

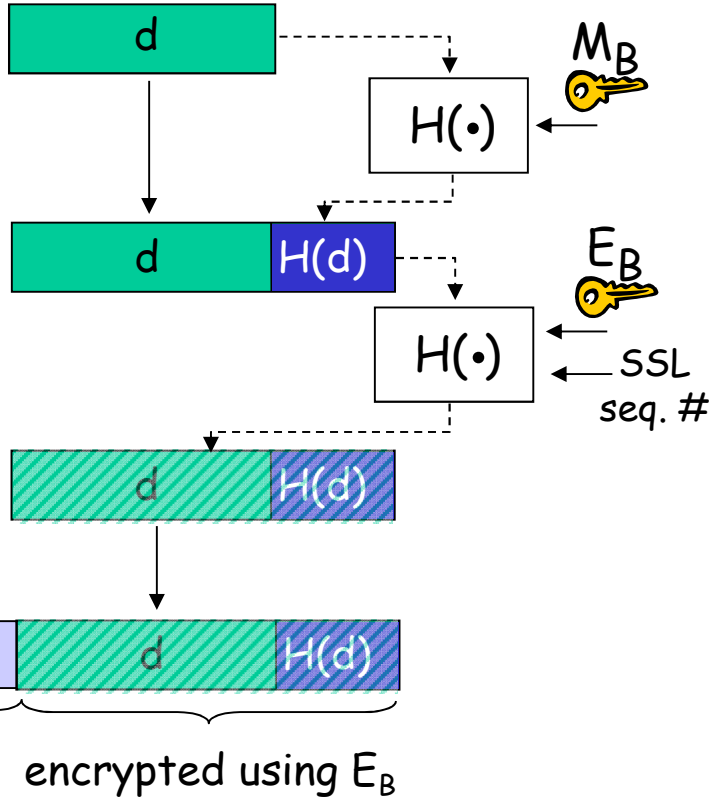
SSL: 3阶段

3. 数据传输

TCP 字节流

$b_1b_2b_3$ … b_n

n字节成一个块



IPsec: 网络层次的安全性

- 网络层次的机密性:
  - 发送端主机对IP数据报中的数据进行加密
  - 数据：TCP或者UDP的段; ICMP和SNMP 报文.
- 网络层次的可认证性
  - 目标主机可以认证源主机的IP地址
- 2个主要协议:
  - 认证头部 (AH)协议
  - 封装安全载荷 encapsulation security payload (ESP) 协议
- 不管AH 还是ESP, 源和目标在通信之前要握手:
  - 创建一个网络层次的逻辑通道：安全关联 security association (SA)
- 每一个SA 都是单向
- 由以下元组唯一确定:
  - 安全协议 (AH or ESP)
  - 源 IP地址
  - 32-bit连接ID

Authentication Header (AH) 协议

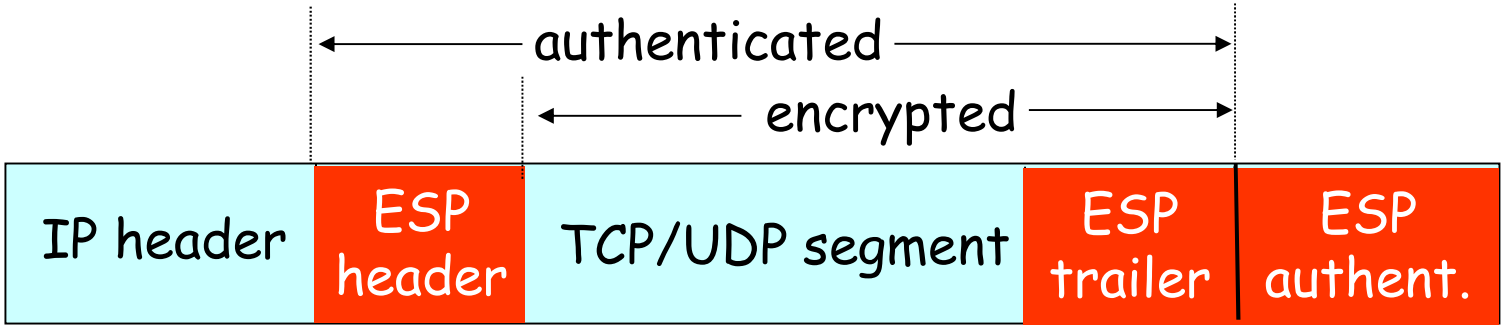
- 提供源端的可认证性，数据完整性，但是不提供机密性
- 在IP头部和数据字段之间插入AH的头部
- 协议字段: 51
- 中间的路由器按照常规处理这个数据报

AH 头部包括:

- 连接ID
- 认证数据: 对原始数据计算报文摘要，使用源端的私钥进行数字签名.
- 下一个字段: 定义了数据的类型 (e.g., TCP, UDP, ICMP)

ESP 协议

- 提供机密性，主机的可认证性，数据的完整性.
- 数据和ESP尾部部分被加密
- next header字段在ESP尾部
- ESP 认证的头部与AH类似
- 协议号 = 50.



IEEE 802.11 security

- *War-driving:* drive around Bay area, see what 802.11 networks available?
  - More than 9000 accessible from public roadways
  - 85% use no encryption/authentication
  - packet-sniffing and various attacks easy!
- Securing 802.11
  - encryption, authentication
  - first attempt at 802.11 security: Wired Equivalent Privacy (WEP): a failure
  - current attempt: 802.11i

Wired Equivalent Privacy (WEP):

- authentication as in protocol *ap4.0*
  - host requests authentication from access point
  - access point sends 128 bit nonce
  - host encrypts nonce using shared symmetric key
  - access point decrypts nonce, authenticates host
- no key distribution mechanism
- authentication: knowing the shared key is enough

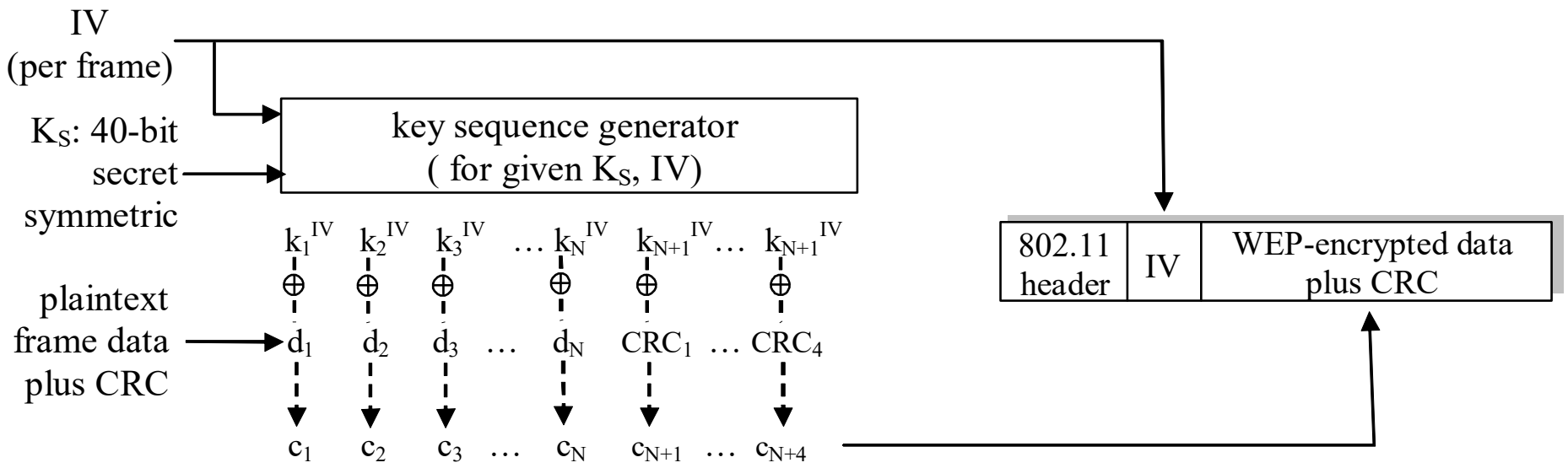
WEP data encryption

- Host/AP share 40 bit symmetric key (semi-permanent)
- Host appends 24-bit initialization vector (IV) to create 64-bit key
- 64 bit key used to generate stream of keys, k_i^{IV}
- k_i^{IV} used to encrypt ith byte, d_i, in frame:

$$c_i = d_i \text{ XOR } k_i^{IV}$$

- IV and encrypted bytes, c_i sent in frame

802.11 WEP encryption



Sender-side WEP encryption

Breaking 802.11 WEP encryption

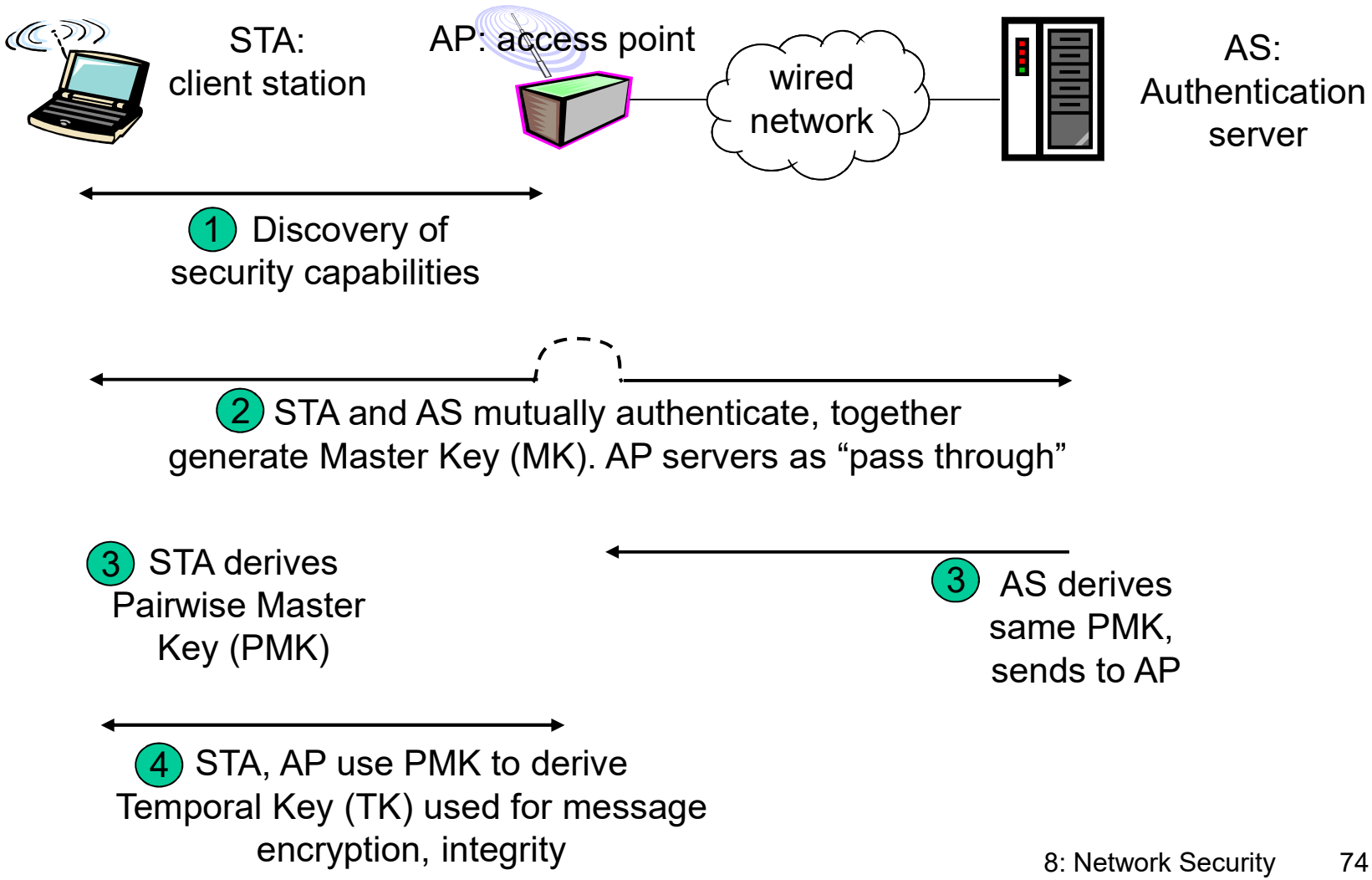
Security hole:

- 24-bit IV, one IV per frame, -> IV's eventually reused
- IV transmitted in plaintext -> IV reuse detected
- **Attack:**
  - Trudy causes Alice to encrypt known plaintext d_1 d_2 d_3 d_4 …
  - Trudy sees: $c_i = d_i$ XOR k_i^{IV}
  - Trudy knows c_i d_i, so can compute k_i^{IV}
  - Trudy knows encrypting key sequence k_1^{IV} k_2^{IV} k_3^{IV} …
  - Next time IV is used, Trudy can decrypt!

802.11i: improved security

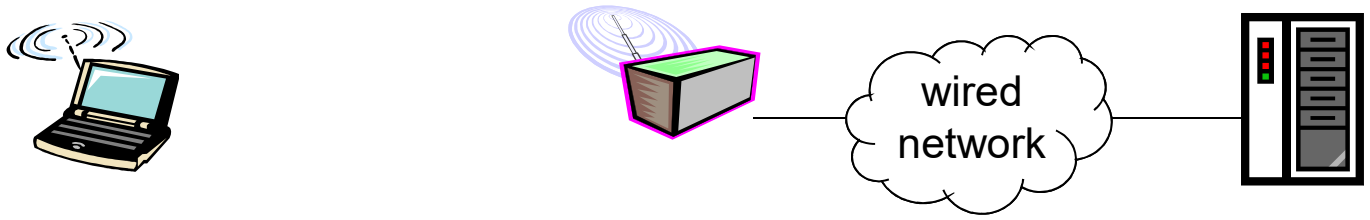
- numerous (stronger) forms of encryption possible
- provides key distribution
- uses authentication server separate from access point

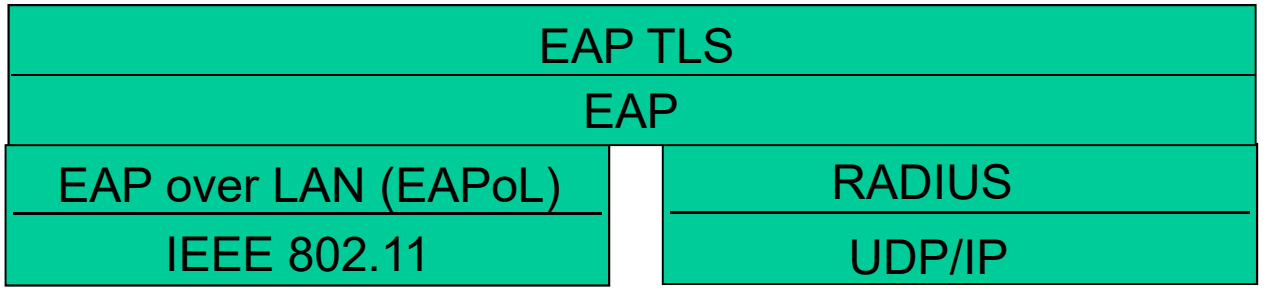
802.11i: four phases of operation



EAP: extensible authentication protocol

- EAP: end-end client (mobile) to authentication server protocol
- EAP sent over separate "links"
  - mobile-to-AP (EAP over LAN)
  - AP to authentication server (RADIUS over UDP)





提纲

8.1 什么是网络安全?

8.2 加密原理

8.3 认证

8.4 报文完整性

8.5 密钥分发和证书

8.6 访问控制：防火墙

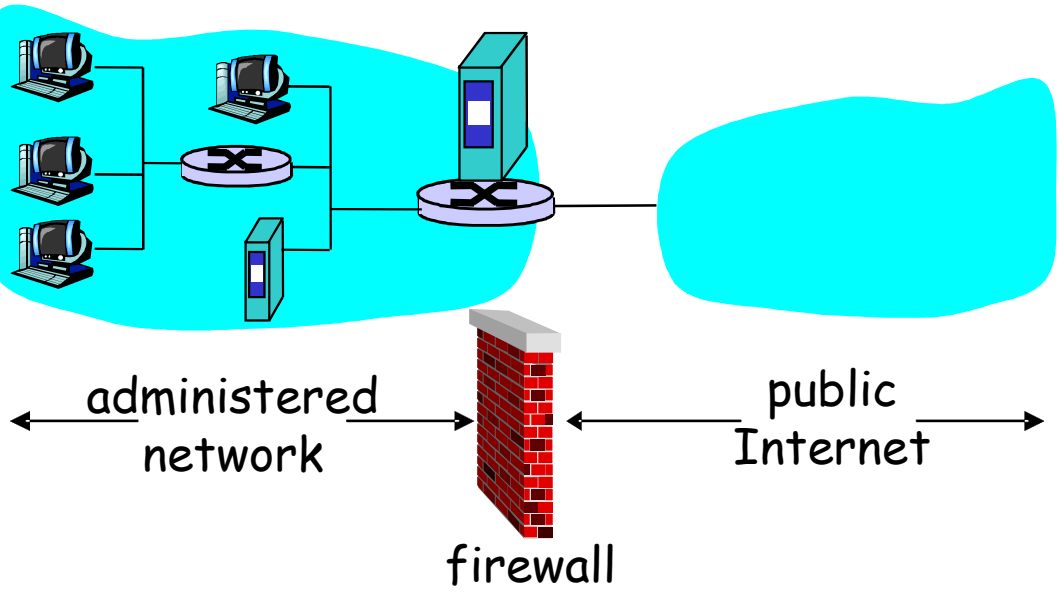
8.7 攻击和对策

8.8 各个层次的安全性

防火墙

firewall

将组织内部网络和互联网络隔离开来，按照规则允许某些分组通过（进出），或者阻塞掉某些分组



防火墙: 为什么需要

阻止拒绝服务攻击:

SYN flooding: 攻击者建立很多伪造TCP链接，对于真正用户而言已经没有资源留下了

阻止非法的修改/对非授权内容的访问

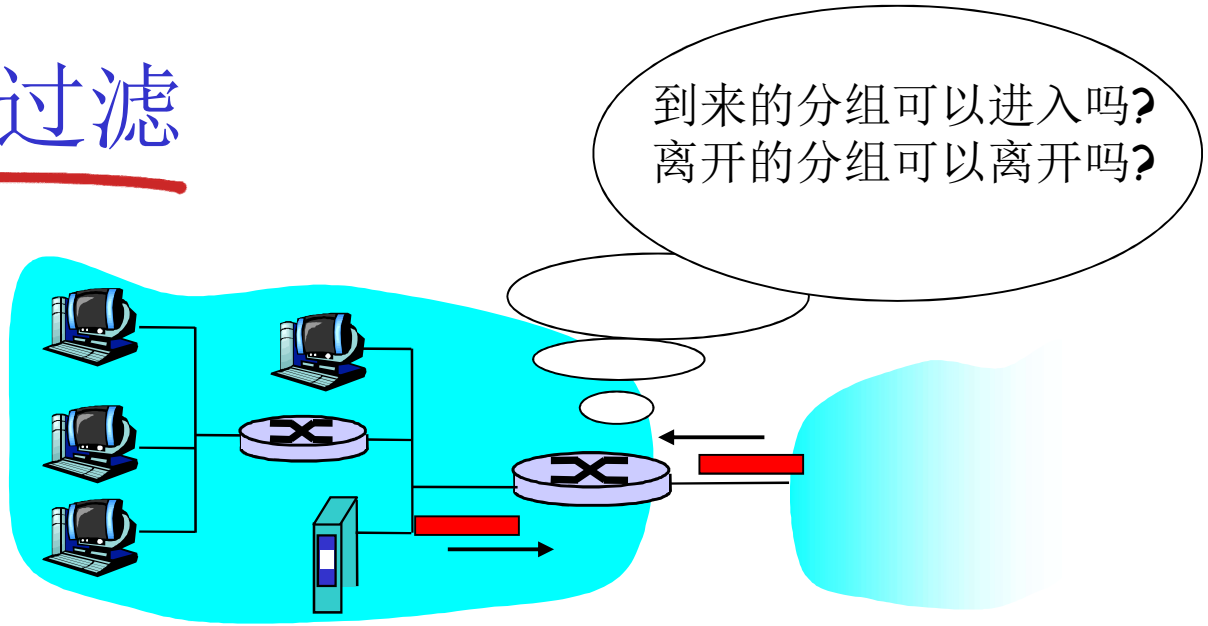
- e.g., 攻击者替换掉CIA的主页

只允许认证的用户能否访问内部网络资源 (经过认证的用户/主机集合)

2种类型的防火墙:

- 网络级别：分组过滤器
  - 有状态，无状态
- 应用级别：应用程序网关

分组过滤



- 内部网络通过配置防火墙的路由器连接到互联网上
- 路由器对分组逐个过滤，根据以下规则来决定转发还是丢弃:
  - 源IP地址,目标IP地址
  - TCP/UDP源和目标端口
  - ICMP报文类别
  - TCP SYN 和ACK bits

分组过滤-无状态

- 例1:阻塞进出的数据报：只要拥有IP协议字段 = 17，而且 源/目标端口号 = 23.
  - 所有的进出UDP流 以及telnet 连接的数据报都被阻塞掉
- 例2: 阻塞进入内网的TCP段：它的ACK=0.
  - 阻止外部客户端和内部网络的主机建立TCP连接
  - 但允许内部网络的客户端和外部服务器建立TCP连接

无状态分组过滤器: 例子

策略	防火墙设置
不允许外部的web进行访问	阻塞掉所有外出具有目标端口80的IP分组
不允许来自外面的TCP连接，除非是机构公共WEB服务器的连接	阻塞掉所有进来的TCP SYN分组，除非130.207.244.203, port 80
阻止Web无线电占用可用带宽.	阻塞所有进来的UDP分组 - 除非 DNS 和路由器广播
阻止你的网络被smurf DoS所利用	阻塞掉所有具有广播地址的ICMP分组 (eg 130.207.255.255).
阻止内部网络被tracerout，从而得到你的网络拓扑	阻塞掉所有外出的 ICMP TTL过期的流量

Access Control Lists

- *ACL:* 规则的表格，top - bottom应用到输入的分组: (action, condition) 对

action	source address	dest address	protocol	source port	dest port	flag bit
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----
deny	all	all	all	all	all	all

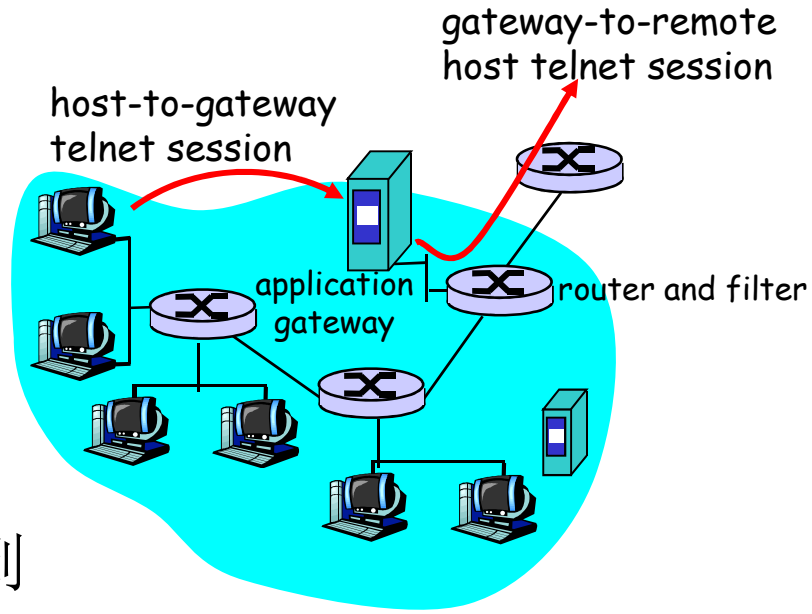
有状态分组过滤

- 无状态分组过滤根据每个分组独立地检查和行动
- 有状态的分组过滤联合分组状态表检查和行动
- ACL增强：在允许分组之前需要检查连接状态表

action	source address	dest address	proto	source port	dest port	flag bit	check conxion
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any	
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK	X
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---	
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----	X
deny	all	all	all	all	all	all	

应用程序网关

- 根据应用数据的内容来过滤进出的数据报，就像根据IP/TCP/UDP字段来过滤一样
  - 检查的级别：应用层数据



- Example: 允许内部用户登录到外部服务器，但不是直接登录

1. 需要所有的telnet用户通过网关来telnet
2. 对于认证的用户而言，网关建立和目标主机的telnet connection，网关在2个连接上进行中继
3. 路由器过滤器对所有不是来自网关的telnet的分组全部过滤掉

防火墙和应用程序网关的局限性

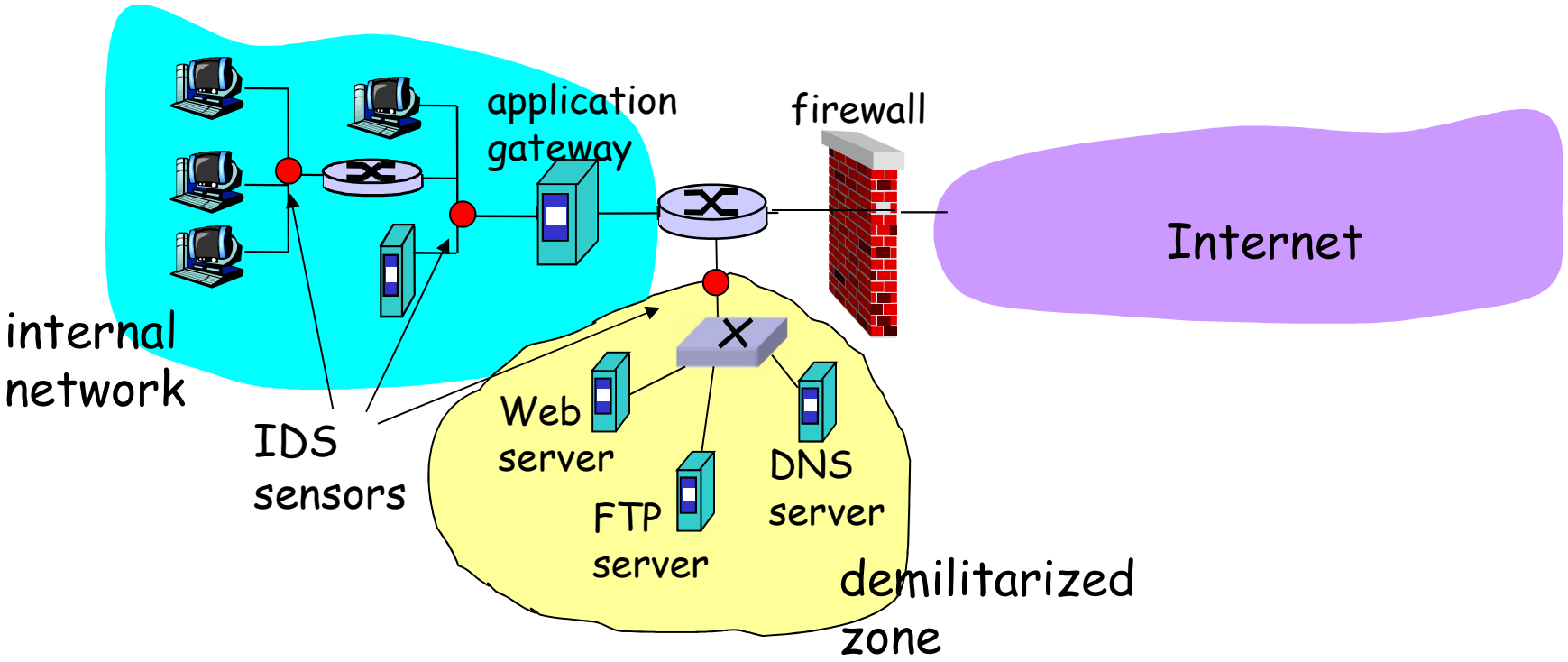
- IP spoofing: 路由器不知道数据报是否真的来自于声称的源地址
- 如果有多个应用需要控制，就需要有多个应用程序网关
- 客户端软件需要知道如何连接到这个应用程序
  - e.g., 必须在Web browser中配置网络代理的Ip地址
- 过滤器对UDP段所在的报文，或者全过或者全都不过
- 折中：与外部通信的自由度，安全的级别
- 很多高度保护的站点仍然受到攻击的困扰

IDS：入侵检测系统

- 分组过滤：
  - 对TCP/IP头部进行检查
  - 不检查会话间的相关性
- *IDS: intrusion detection system*
  - *深入分组检查*：检查分组的内容 (e.g., 检查分组中的特征串 已知攻击数据库的病毒和攻击串
  - 检查分组间的相关性，判断是否是有害的分组
    - 端口扫描
    - 网络映射
    - DoS 攻击

IDS：入侵检测系统

- multiple IDSs：在不同的地点进行不同类型的检查



Internet 安全威胁

映射:

- 在攻击之前：“踩点” - 发现在网络上实现了哪些服务
- 使用ping来判断哪些主机在网络上有地址
- 端口扫描：试图顺序地在每一个端口上建立TCP连接 (看看发生了什么)
- nmap (http://www.insecure.org/nmap/) mapper: “network exploration and security auditing”

对策?

Internet 安全威胁

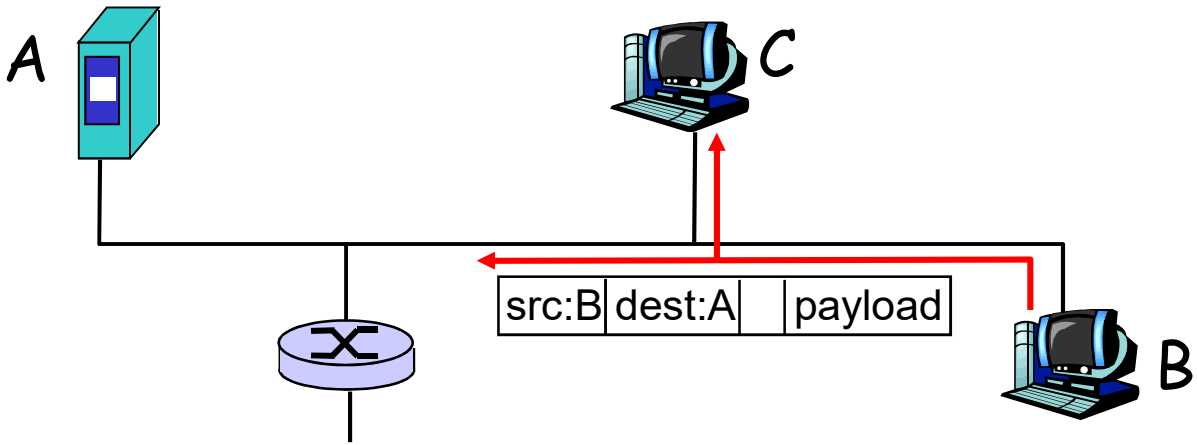
映射: 对策

- 记录进入到网络中的通信流量
- 发现可疑的行为 (IP addresses, 端口被依次扫描)

Internet 安全威胁

分组嗅探:

- 广播式介质
- 混杂模式的NIC获取所有的信道上的分组
- 可获取所有未加密的数据 (e.g. passwords)
- e.g.: C 嗅探B的分组

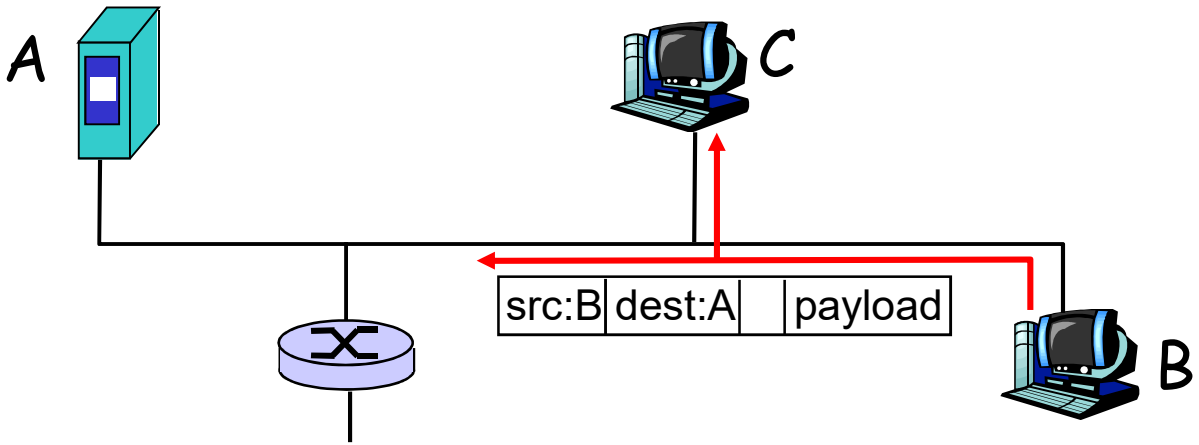


对策?

Internet 安全威胁

分组嗅探: 对策

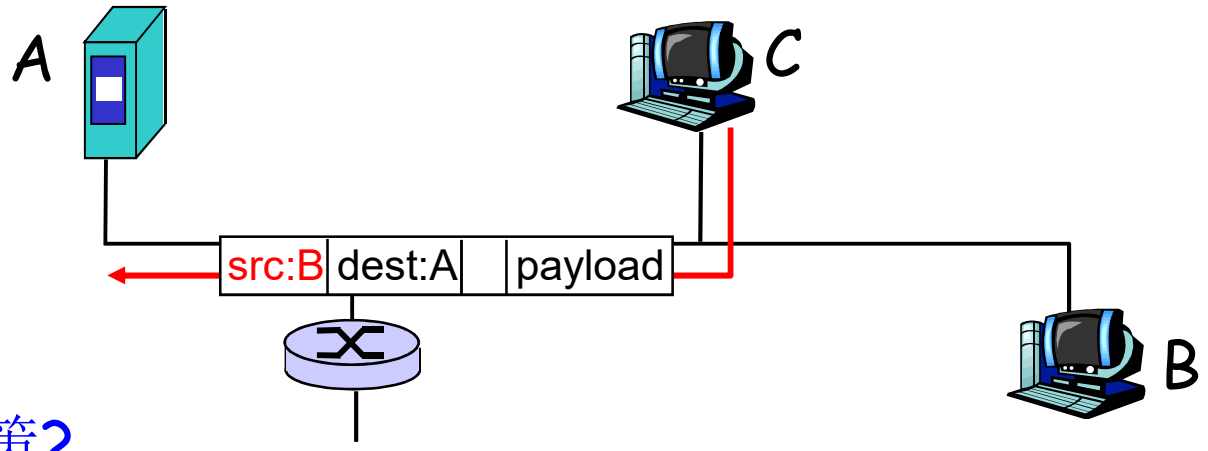
- 机构中的所有主机都运行能够监测软件，周期性地检查是否有 网卡运行于混杂模式
- 每一个主机一个独立的网段 (交换式以太网而不是使用集线器)



Internet 安全威胁

IP Spoofing欺骗:

- 可以有应用进程直接产生 "raw" IP分组, 而且可以在IP源地址部分直接放置任何地址
- 接收端无法判断源地址是不是具有欺骗性的
- e.g. C 伪装成B

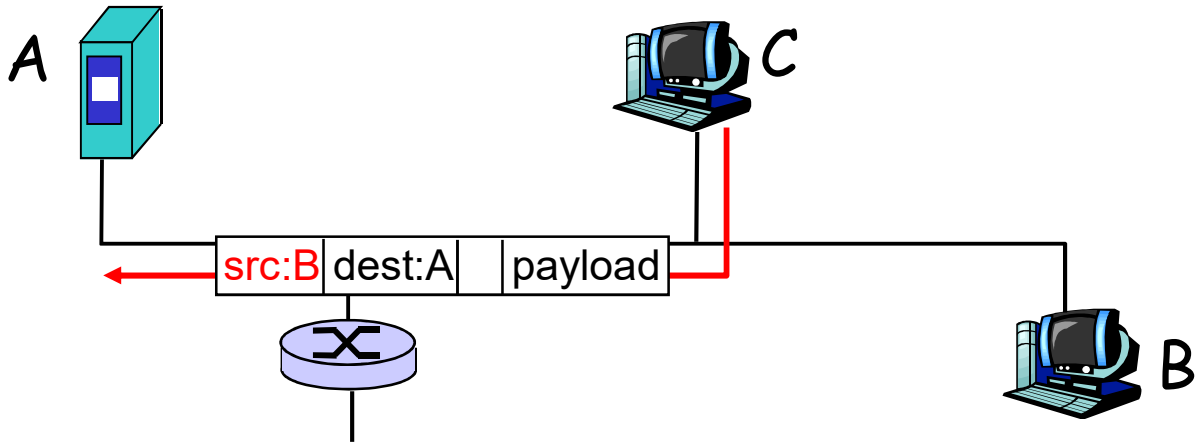


对策?

Internet 安全威胁

IP Spoofing：入口过滤

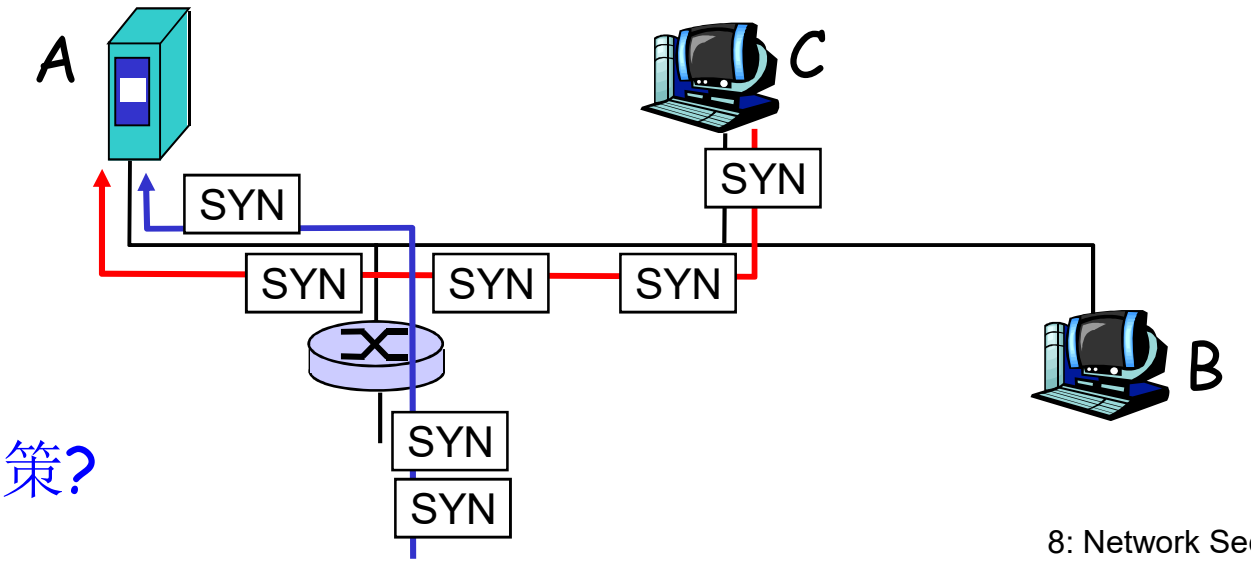
- 路由器对那些具有非法源地址的分组不进行转发 (e.g., 数据报的源地址不是路由器所在的网络地址)
- 很好，但是入口过滤不能够在全网范围内安装



Internet 安全威胁

Denial of service (DOS):

- 产生的大量分组淹没了接收端
- Distributed DOS (DDOS): 多个相互协作的源站淹没了接收端
- e.g., C 以及远程的主机SYN-attack A

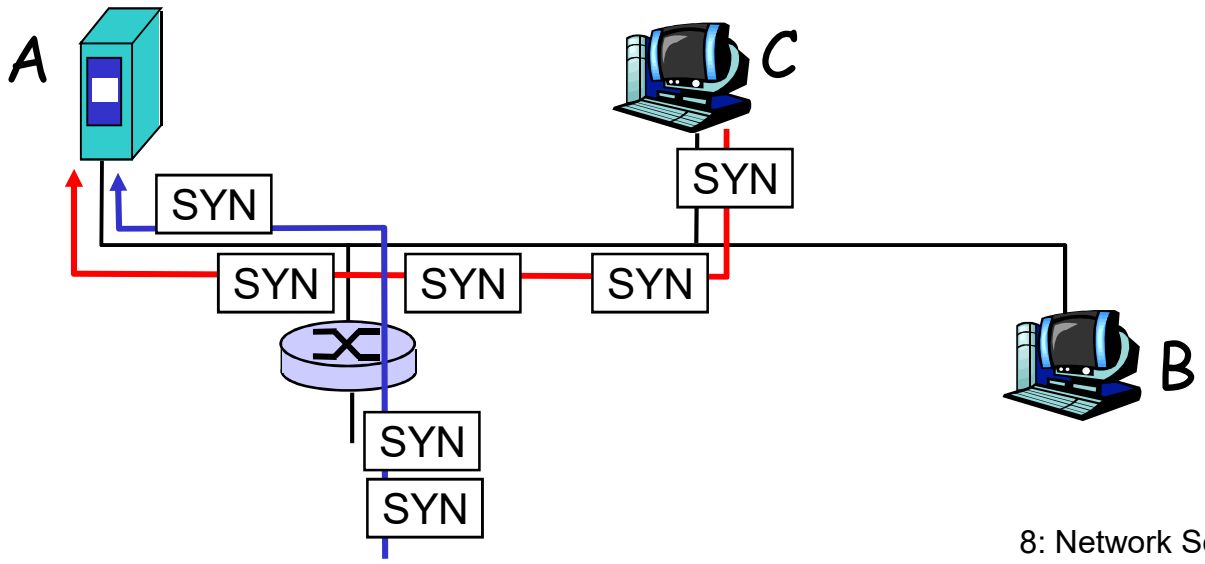


对策?

Internet 安全威胁

Denial of service (DOS): 对策

- 在到达主机之前过滤掉这些泛洪的分组 (e.g., SYN) : throw out good with bad
- 回溯到源主机(most likely an innocent, compromised machine)



网络安全 (总结)

基本原理

- 加密 (对称和公开)
- 报文完整性
- 端节点的认证（鉴别）

在多种安全场景中使用

- 安全电子邮件
- 安全传输层 (SSL)
- IP sec
- 802.11

运行中的安全性: firewalls and IDS

作业

- 第十二周 第1次
  - 复习题：1，3，5，9
  - 习题：5，6
- 第十二周 第2次
  - 复习题：10，11，13
  - 习题：10，11，13