



# 计算机网络概论 ——网络安全概述

---

刘志敏

liuzm@pku.edu.cn



# 提纲

---

- 信息传输安全的概念
- 加密技术
- 数字签名
- 访问控制：认证
- 安全的传输协议：
  - 网络层IPSec
  - 传输层SSL
  - WLAN 安全协议
- 应用层安全



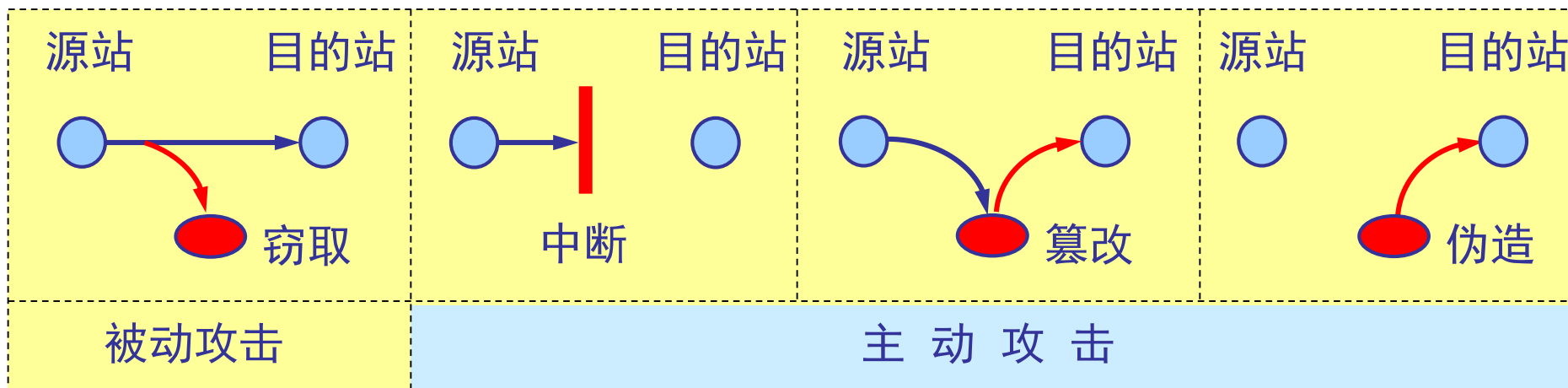
# 信息传输安全的概念

---

- **信息传输安全：**通过各种计算机、网络和通信技术，保证在网络中传输信息的机密性、认证性、完整性和不可否认性
  - **机密性：**保证信息为授权者使用，不会泄漏给未经授权者
  - **认证性：**在接收敏感信息或进行商业交易时，确认通信的对方是谁；保证信息从真实的发送者传送到真实的接收者
  - **完整性：**信息传递过程没有被他人添加、删除、替换
  - **不可否认性：**发送者都应对其发布的信息负责，不能否认曾对信息进行的生成、签发、接受等行为

# 信息传输面临的安全问题

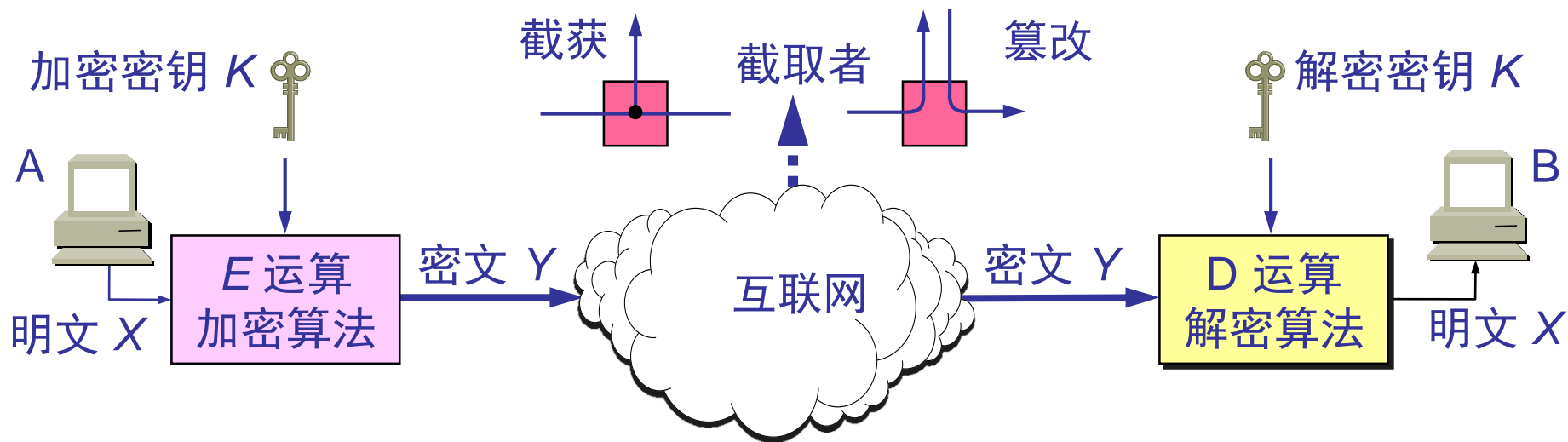
- (1) 窃取——从网络上截获他人的通信内容
- (2) 中断——有意中断他人在网络上的通信
- (3) 篡改——故意篡改网络上传送的报文
- (4) 伪造——伪造信息在网络上传送



应对被动攻击的手段——数据加密

应对主动攻击的手段——数字签名，认证

# 数据加密模型



$$D_K(Y=E_K(X)) = X$$

- 用加密函数E及参数K（密钥）加密明文X得到密文Y，用解密函数D及参数K（密钥）解密密文Y得到明文X
- 原则：所有的算法是公开的，而只有密钥是保密的
  - 如何实现密钥的保密性？信息传输与密钥传输不在同一个网络上，定期更换密钥，其他？



# 密码体制及算法



## ■ 常规密码体制

- 加密密钥与解密密钥是**相同的**，这种加密系统又称为**对称密钥系统**
- 对称密钥算法
  - 置换密码
  - 替代密码
  - 数据加密标准 DES ( Data Encryption Standard )

## ■ 公钥密码体制

- 加密密钥与解密密钥是不同的

- 





# 加密技术——置换密码

明文    **abcdefghijklmnopqrstuvwxyz**

密文    **QWERTYUIOPASDFGHJKLZXCVBNM**

**attack —————> QZZQEA**

- 单字母置换：密钥为对应的26个字母，密钥的可能性为 $26! = 4 \times 10^{26}$
- 如何破解？密码分析！
  - 破解单字母密码：利用自然语言的统计特性，如e最常见，其次是t、o、a、n、i，两字母组合th、in、er、an，三字母组合the、ing、and、ion；统计密文中出现的频率，试探地分配给字母e和t；找到常见的tXe，则X可能是h；逐个字母地构造试探性报文
  - 猜测一个可能的单词或短语，利用单词本身的特点，例如，某一会计事务所的密文，financial为关键词，i之间间隔4个字母，a重复出现……，分析密文中有此规律的，猜测i的密钥。





# 破解密码练习

- 试破解下面的密文诗。加密采用置换密码（将26个字母用其它字母替代），密文中无标点符号，空格未加密
- kfd ktbd fzm eubd kfd pzyiom mztz ku kzyg ur bzha  
kfthcm ur mftnm zhx mftnm zhx mdzythc pzq ur  
ezsszcdm zhx gthcm zhx pfa kfd mdz tm sutythc fuk zhx  
pfdkfdi ntcn fzld pthcm sok pztz z stk kfd uamkdim eitdx  
sdruid pd fzld uoi efzk rui mubd ur om zid uok ur sidzkd  
zhx zyy ur om zid rzk hu foia mztz kfd ezindhkdi kfda  
kfzhgdx ftb boef rui kfzk
- 提示：1) 频度统计：单字母，双字母、三字母  
2) 利用频度信息，做字母猜测及替换，建立“替换表”，具有自动学习及更新功能

# 加密技术——替代密码

<u>M</u>	<u>E</u>	<u>G</u>	<u>A</u>	<u>B</u>	<u>U</u>	<u>C</u>	<u>K</u>
<u>7</u>	<u>4</u>	<u>5</u>	<u>1</u>	<u>2</u>	<u>8</u>	<u>3</u>	<u>6</u>
p	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	i	o	n	
d	o	l	l	a	r	s	t
o	m	y	s	w	i	s	s
b	a	n	k	a	c	c	o
u	n	t	s	i	x	t	w
o	t	w	o	a	b	c	d

明文

pleasetransferonemilliondollarsto  
myswissbankaccountsixtwo

密文

AFLLSKSOSELAWAIATOOSSCTCLNMOMANT  
ESILYNTWRNNTSOWDPAEDOBUEOERIRICXB

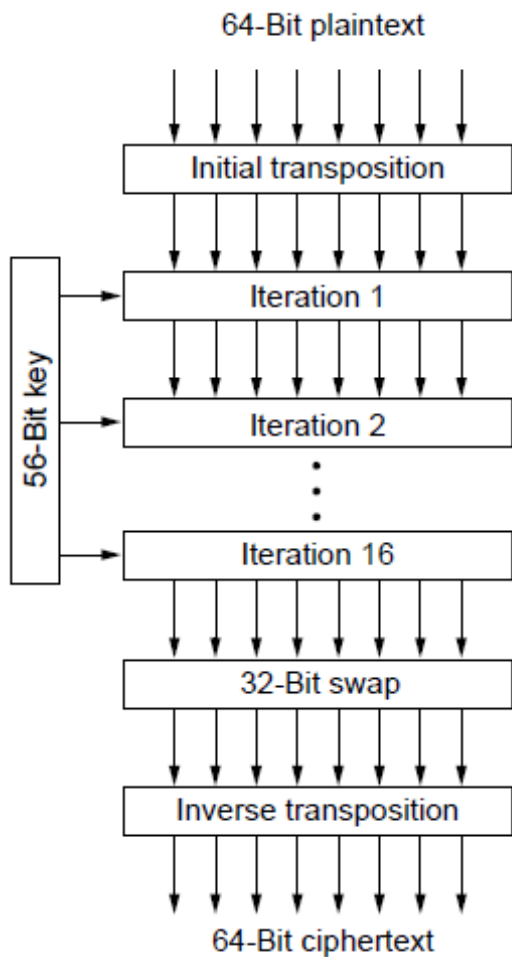
- 选择单词或短语作密钥，如MEGABUCK，要求不含重复字母
- 用密钥对字母顺序重新排列：明文按水平方向写，7个（为密钥长度）一组，从密钥字母最小的那列开始则按列读出密文
- 如何破解？
  - 统计字母频度，判断是否为替代密码
  - 再猜密钥长度：根据两字母（th、in、er、an）的间距
  - 再判定列的顺序



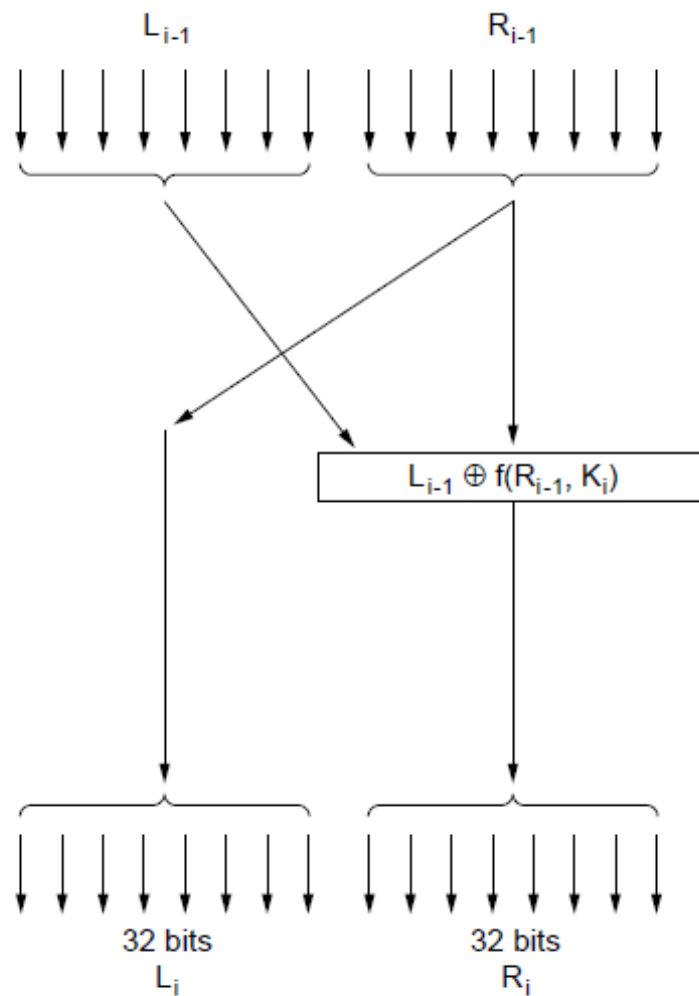
# 数据加密标准 DES

- 1972年美国IBM公司研制的的数据加密标准DES，一种对称密码体制，是一种分组密码；
- 在加密前，先对明文分组，每组长64位
- 然后对每个64位进行加密，产生一组64位密文
  - 密钥为64位（实际密钥长56位，8位为奇偶校验）
- 最后将各组密文串接起来得到密文
- 共19个步运算：每组64分为 $L_0$ 、 $R_0$ 两部分；在密钥控制下经过16次迭代运算后得到 $L_{16}$ 、 $R_{16}$ ；将此作为输入进行逆置换，即得到密文

# 数据加密标准(1)



(a)



(b)

(a) 一般结构. (b) 每次迭代的细节  $\oplus$  表示异或



# DES的保密性

- DES 的保密性仅取决于对密钥的保密，因算法是公开的
- DES是世界上第一个公认的实用密码算法标准
- 在破译DES方面取得了许多进展，但至今更有效的方法还是穷举搜索密钥
  - 密码破译竞赛：在一定时间内解密一段密文；由RSA公司提供奖金1万美元
  - 破解密文的时间：四个月（1997年）、41天（1998年）、22小时（1999年1月）
- 问题：DES的密钥长度有限，已经设计出搜索DES密钥的专用芯片

# 公钥密码体制

- 因简单加密的加密算法公开，安全性取决于密钥的安全
- 1976年，美国斯坦福大学的迪菲（Diffie）和赫尔曼（Hellman）提出公开密钥密码的新思想（论文“New Direction in Cryptography”）：加密算法本身公开，加密用的密钥也可以公开

- 迪菲（Diffie）、  
赫尔曼（Hellman）：  
2015年图灵奖获得者



- 并非降低保密程度，因为加密密钥和解密密钥不同，只需对解密密钥保密。这就是**公钥密码**，也称为**非对称密码**

# RSA算法(1)

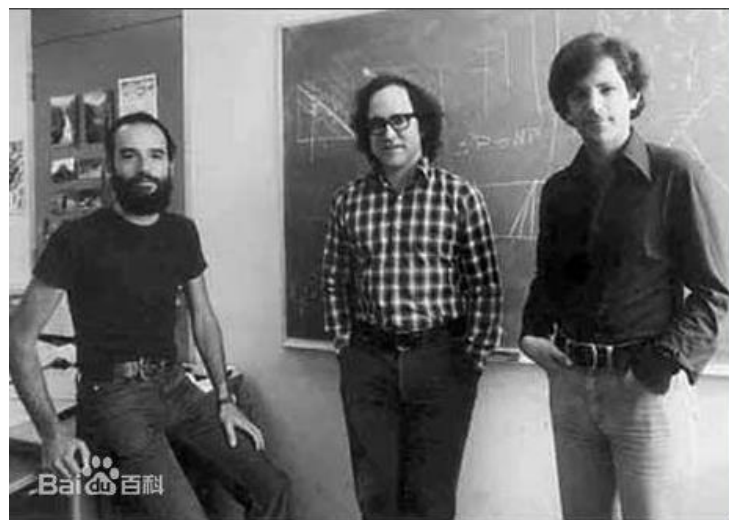
- RSA算法：（Rivest, Shamir, Adleman MIT的三位学者，1977年提出，1987年公布）

1. 选择两个大素数 $p$ 和 $q$
2. 计算  $n = p \times q$  和  $z = (p-1) \times (q-1)$
3. 选择一个与  $z$  互素的数  $d$
4. 找到  $e$  使其满足  $e \times d = 1 \bmod(z)$

将明文分块，使每个消息 $p$ 满足  $0 \leq p < n$ ;

加密明文信息，只要计算  $C = p^e \bmod(n)$

解密 $C$ ，只要计算  $P = c^d \bmod(n)$



# RSA算法(2)

## RSA 算法举例

选择  $p=3$ ,  $q=11$ ,

则  $n = p \times q = 33$ ;  $z = (p-1) \times (q-1) = 20$

取  $d = 7$  (与  $z$  互素); 则  $e = 3$

Plaintext (P)		Ciphertext (C)			After decryption	
Symbolic	Numeric	$P^3$	$P^3 \pmod{33}$	$C^7$	$C^7 \pmod{33}$	Symbolic
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	01	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	05	E
Sender's computation				Receiver's computation		

- 明文状态数  $p$  小于 33, 只能实现单字母置换
- 加密密钥 3, 33; 解密密钥 7, 33



# 公开密钥算法的特点

(1) 发送者用加密密钥PK对明文X加密，接收者用私有密钥SK解密，即可恢复明文，写为：

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

- SK是接收者专用的秘密密钥，对其他人保密
- 加密和解密运算可以对调，即

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

(2) 加密密钥是公开的，但不能用于解密，即：

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

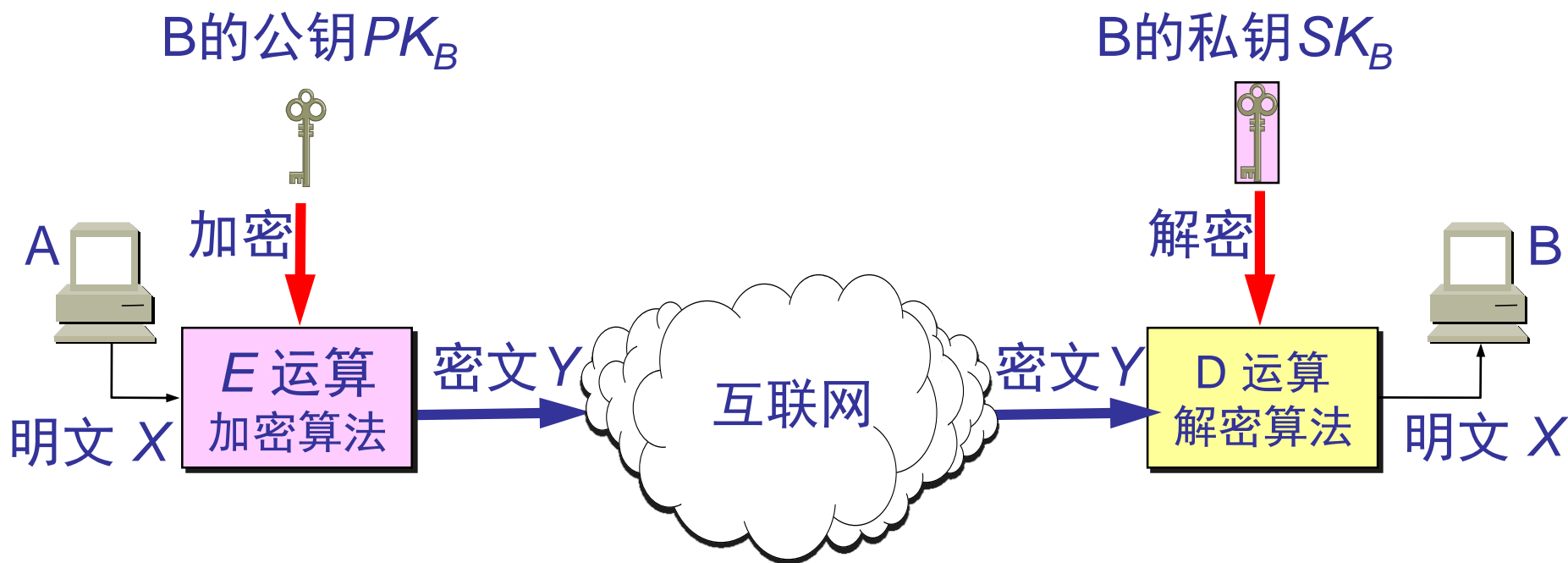
(3) 在计算机上可容易地产生一对PK和SK

(4) 从已知的PK不能推出SK

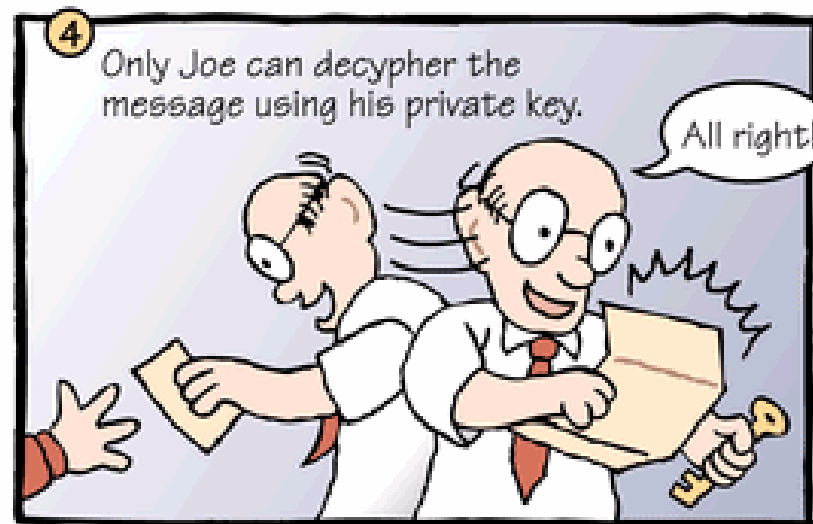
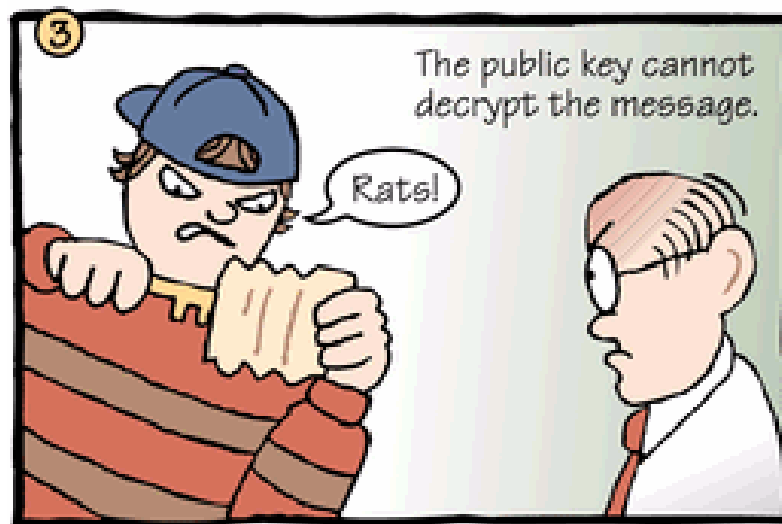
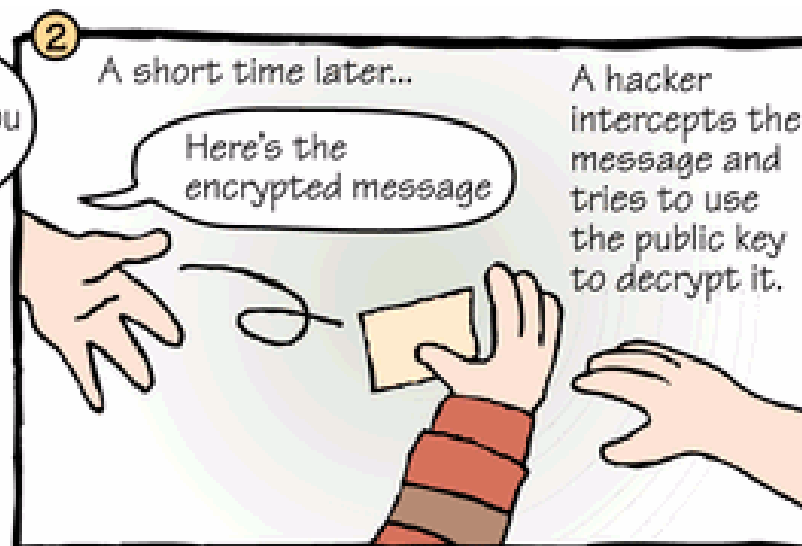
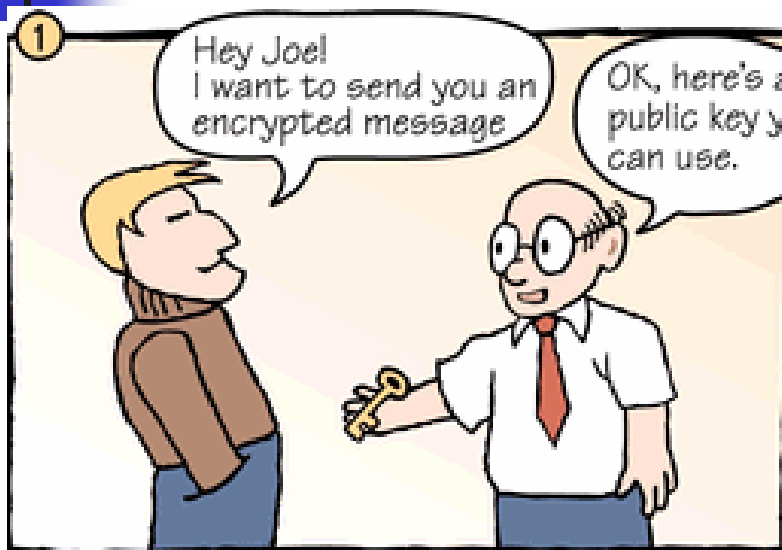
(5) 加密算法和解密算法都是公开的

# 公钥密码体制

- 只有拥有私钥的才能恢复X，实现信息保密传输



# 保证信息安全的网络





# 提纲

---

- 信息传输安全的概念
- 加密技术
- 数字签名
- 访问控制：认证
- 安全的传输协议：
  - 网络层IPSec
  - 传输层SSL
  - WLAN 安全协议
- 应用层安全



# 数字签名

- 法律、金融、证明材料等文档的真实性，需要手写签名，数字签名要求：

- (1) 接收者能够核实发送者的身份；
- (2) 发送者事后不能抵赖报文的内容；
- (3) 接收者不能伪造报文。

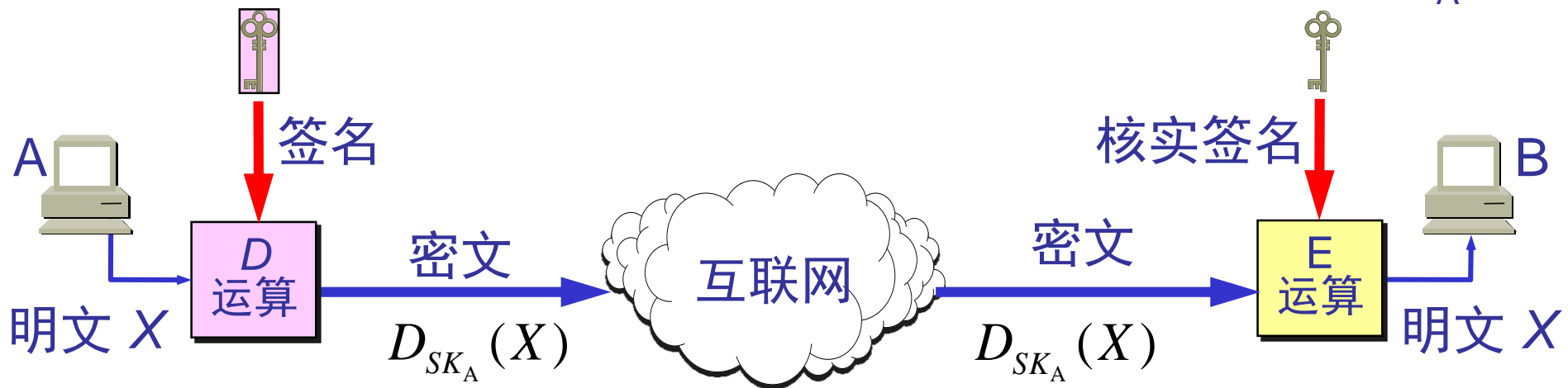
- 数字签名的方法

- 对称密钥签名
- 公开密钥签名
- 报文摘要

# 数字签名的实现

A 的私钥  $SK_A$

A 的公钥  $PK_A$



- 因为A的私钥只能A拥有，因此B相信报文X是A签发的
- 若A要抵赖曾发送报文给B，则B可将明文和对应的密文出示给第三方，由第三方用A的公钥证实A发送X给B
- 反之，若B将X伪造为 $X'$ ，则B不能在第三方前出示对应的密文，因此证明 $X'$ 是B伪造的报文。



# 报文完整性

- 数字签名可实现数据加密与认证
- 数字签名可以认证报文（来源、内容完整性），但许多报文并不需要加密，因加密增加计算量
- 当传送不需要加密的报文时，使接收者用简单的方法认证报文的真伪——**报文**摘要 MD(Message Digest)
- 报文摘要使接收方验证收到报文的真伪：发送者和报文内容、发送时间、序列等

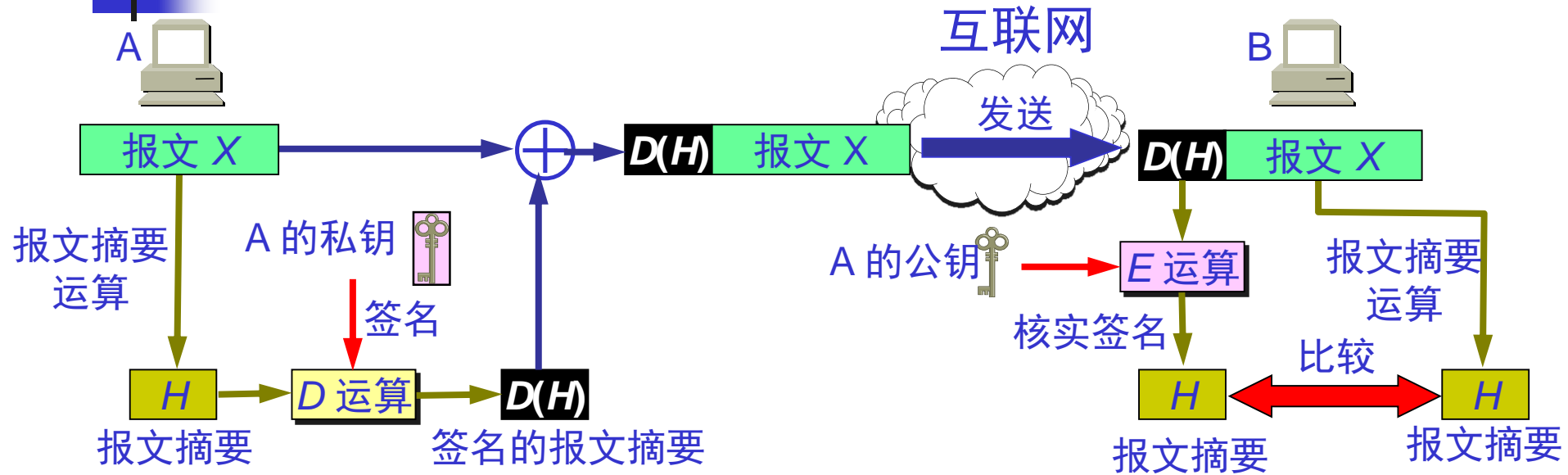


# 报文摘要 MD(Message Digest)

- 散列 (Hash) 函数：对于输入 $X$ ，经过运算 $F$ （如校验和、CRC等）得到固定长度的输出；且若 $X \neq Y$ ，则 $F(X) \neq F(Y)$
- A将报文 $X$ 经散列算法后得到报文摘要 $H$ 。然后用私钥对 $H$ 数字签名，得到 $D(H)$ 后，将其追加在报文 $X$ 后发送给B
- B收到报文后：
  - 用A的公钥对 $D(H)$ 进行 $E$ 运算，得出报文摘要 $H$
  - 对报文 $X$ 进行报文摘要运算，输出 $H'$ 。若 $H'$ 与 $H$ 相同，则断定收到的报文是A产生的；否则就不是



# 报文摘要的实现



- 对报文摘要 $H$ 数字签名比对整个报文数字签名，计算量少
- 但对认证报文 $X$ 来说，效果是一样的。报文 $X$ 和已签名的报文摘要 $D(H)$ 合在一起是**不可伪造的**，是**可检验的**和**不可否认的**



# 提纲

---

- 信息传输安全的概念
- 加密技术
- 数字签名
- 访问控制：认证
- 安全的传输协议：
  - 网络层IPSec
  - 传输层SSL
  - WLAN 安全协议
- 应用层安全



# 认证(authentication)

---

日常生活中的认证：

见面时识别面容、电话中识别声音、通过海关时海关官员识别护照上的照片

认证：验证通信对端是期望的实体而不是假冒者

认证是必要的：在用户与真实网站之间，截获重要的信息比较难，但假冒某一用户或网站则相对容易，例如钓鱼网站冒充银行网站或用低价商品吸引用户注册并访问，要求用户注册（账户、密码、银行卡号、身份证号、电话号码）；钓鱼网站获得了重要信息；

# 认证

目标: Bob希望Alice向其“证明”身份

Protocol ap1.0: Alice says "I am Alice"



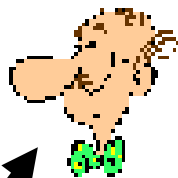
Failure scenario?



# 认证

目标: Bob希望Alice向其“证明”身份

Protocol ap1.0: Alice says "I am Alice"

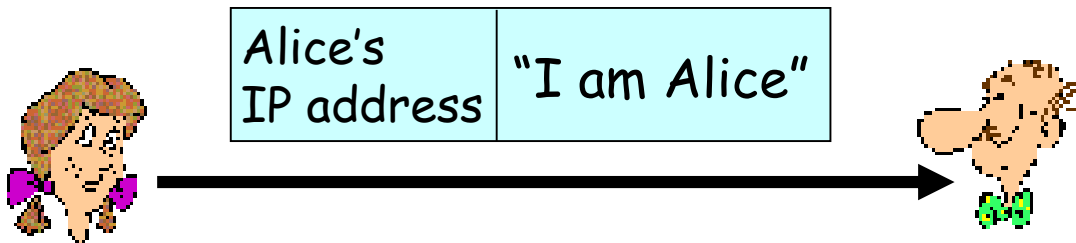


"I am Alice"

在网络中, Bob 无法看到Alice, 所以Trudy声称他自己就是Alice。  
如何识别是真的Alice?

# 认证: 再试

- Protocol ap2.0: Alice 在一个 IP 分组中说 “I am Alice”，并含有其源 IP 地址

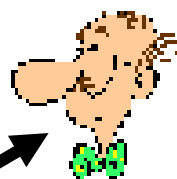
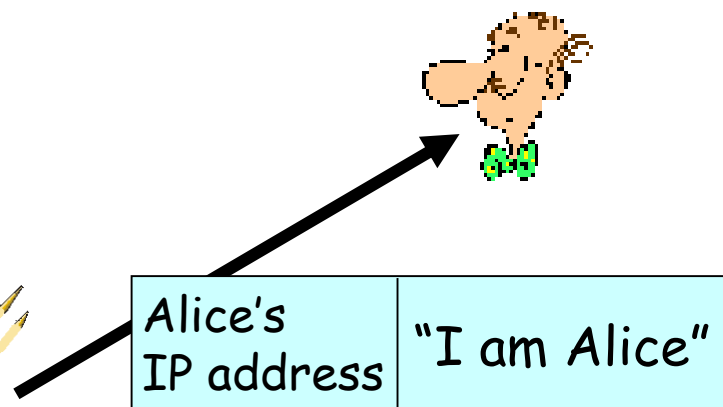


Failure scenario?



# 认证: 再试

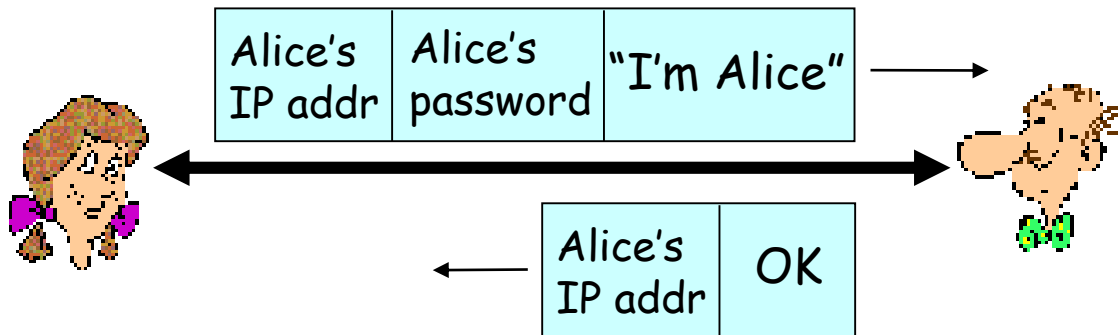
- Protocol ap2.0: Alice 在一个 IP 分组中说 “I am Alice”，并含有其源 IP 地址



Trudy 产生一个假冒 Alice 地址的分组 (spoofing)  
如何识别是真的 Alice?

# 认证: 再试

- Protocol ap2.0: Alice 在一个 IP 分组中说 “I am Alice”，并含有其源 IP 地址



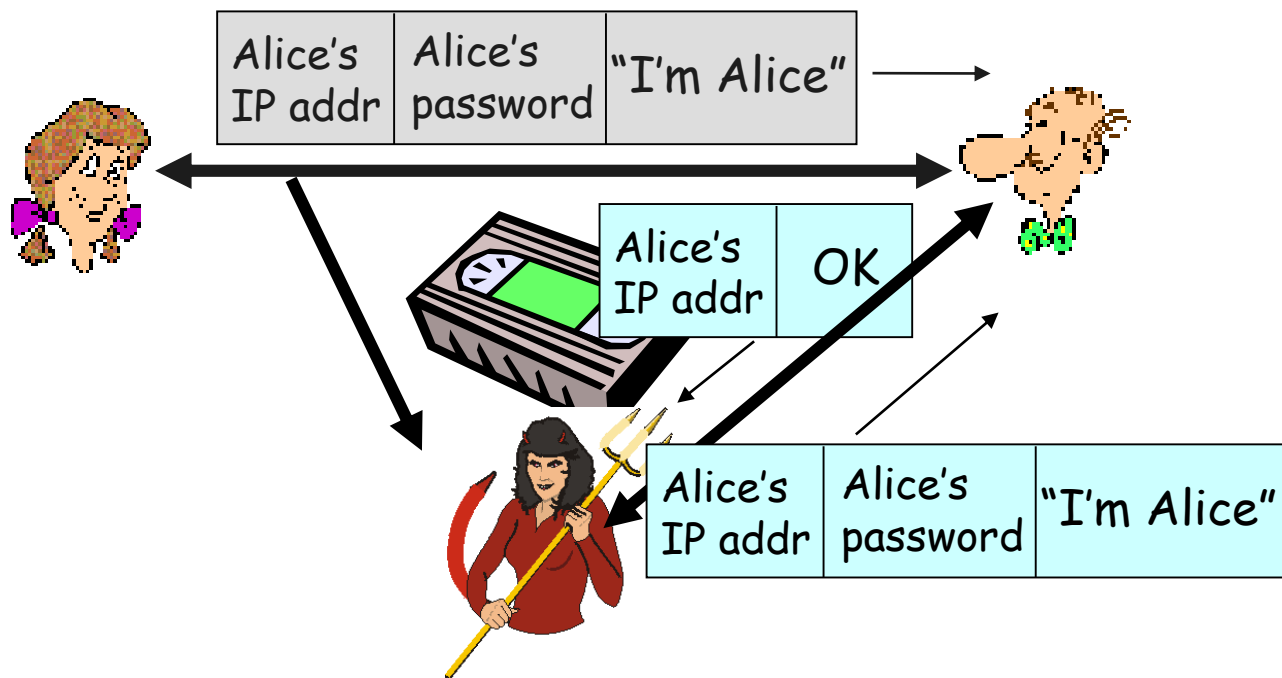
Failure scenario?





# 认证: 再试

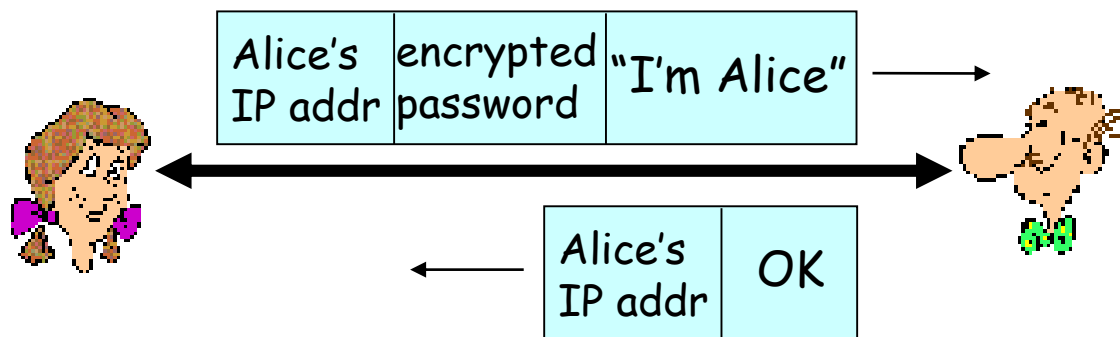
Protocol ap3.0: Alice说 “I am Alice”, 并发送她自己的密码口令以 “证明”。



重放攻击 *playback attack*: Trudy 记录 Alice 的分组, 之后再发给 Bob

# 认证: 再试

- Protocol ap3.1: Alice说 “I am Alice”, 并发送她的已用对称密钥加密的秘密口令以 “证明”

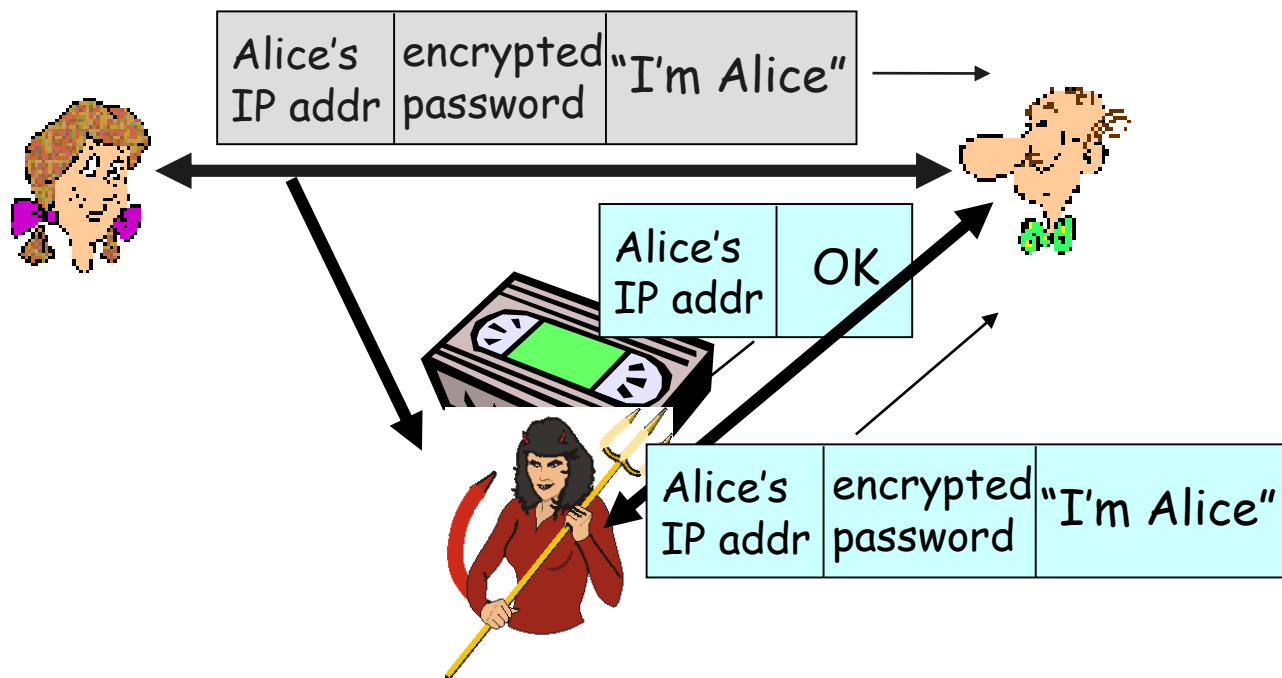


Failure scenario?



# 认证: 再试

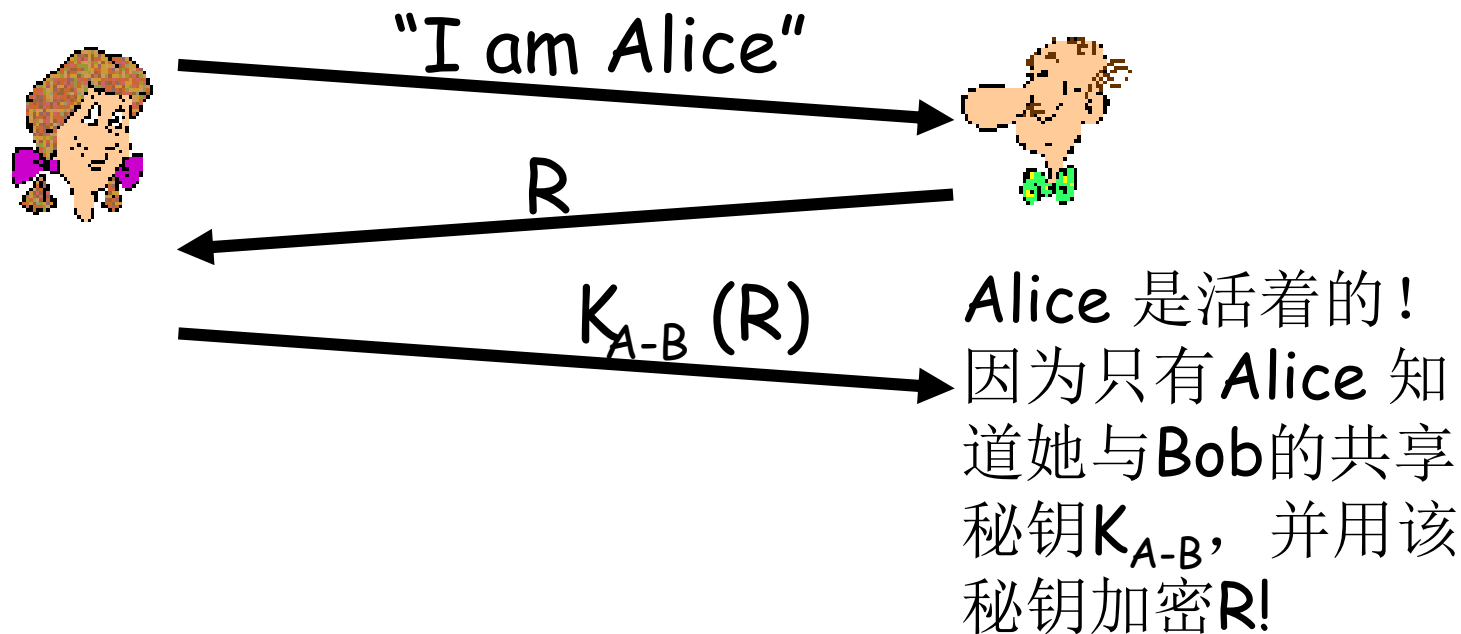
- Protocol ap3.1: Alice说 “I am Alice”, 并发送她的已用对称密钥加密的秘密口令以 “证明”



记录并重放  
仍能工作!

# 认证: 再试

- 目的: 避免重放攻击, 使用不重数 (Nonce)  $R$ , 之后在相当长时间内不再使用该数
- ap4.0: 为证明Alice “活着”, Bob发给Alice不重数 $R$

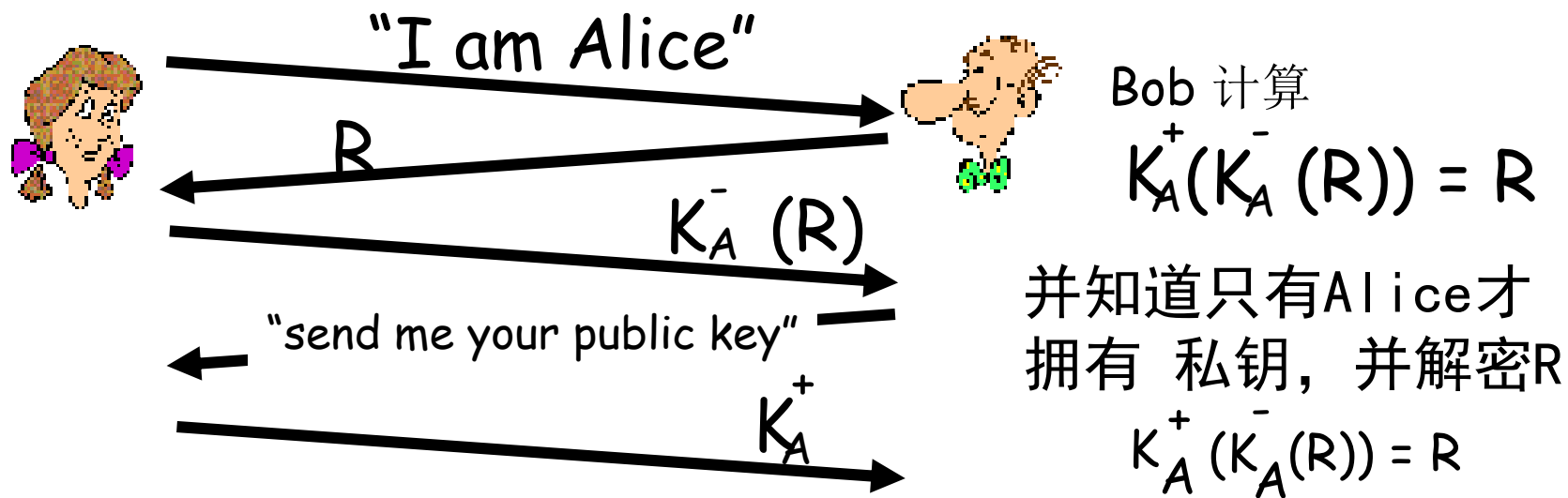


# 认证: ap5.0

ap4.0 需要共享对称密钥

- 可以使用公钥加密技术吗?

ap5.0: 使用不重数, 并采用公钥加密



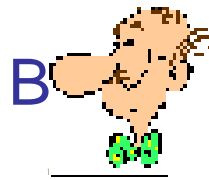
Failure scenario?

# 中间人攻击

中间人 C

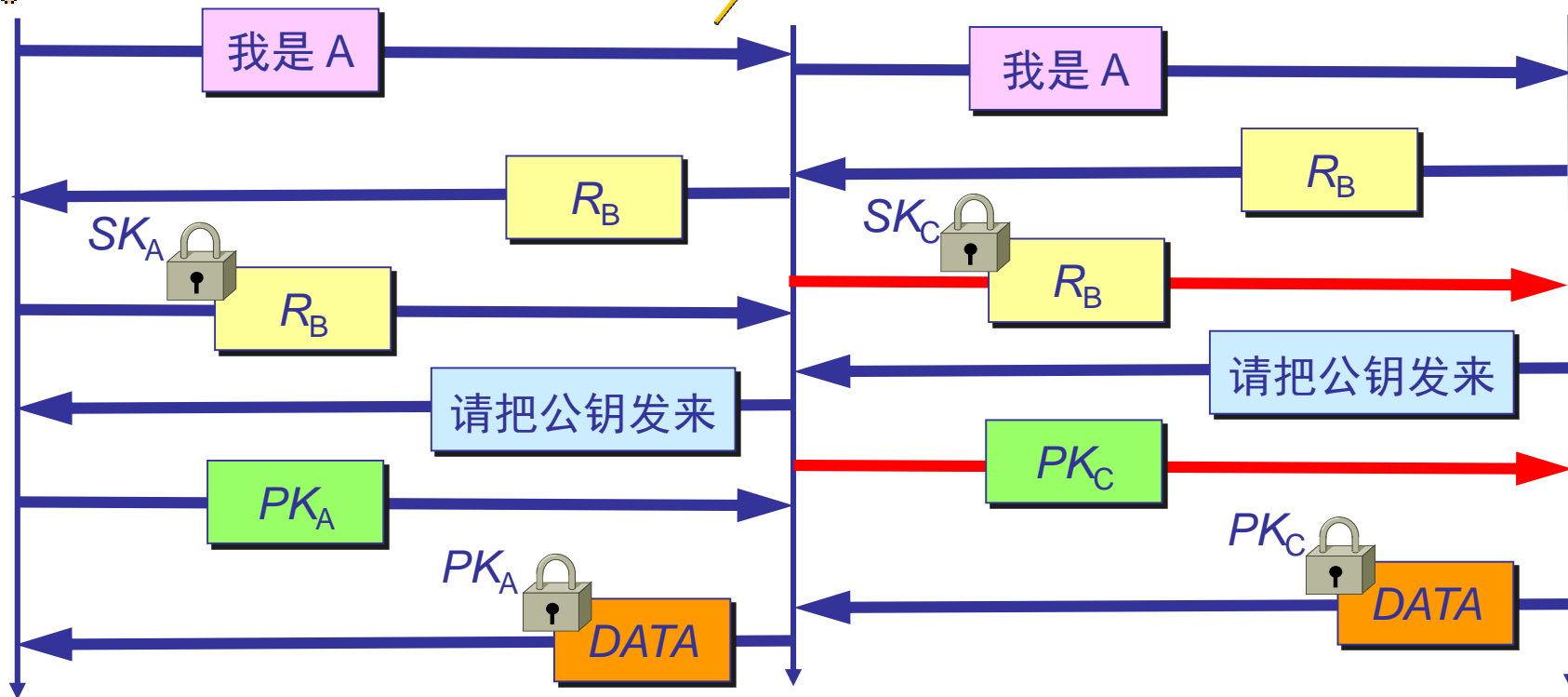


A



B

时间





# 中间人攻击说明

- A向B发送“我是A”的报文，并给出其身份。“中间人”C截获报文并转发给B。B选择一个不重数 $R_B$ 发送给A，但同样被C截获后也照样转发给A
- C用私钥 $SK_C$ 对 $R_B$ 加密后发回给B，使B误以为是A发的；A收到 $R_B$ 后也用其私钥 $SK_A$ 对 $R_B$ 加密后发回给B，中途被C截获并丢弃；B向A索取其公钥，此报文被C截获后转发给A
- C把其公钥 $PK_C$ 冒充是A的发给B，而C也截获到A发给B的公钥 $PK_A$
- B用收到的公钥 $PK_C$ （以为是A的）对数据加密发给A。C截获后用其私钥 $SK_C$ 解密，复制一份留下，再用A的公钥 $PK_A$ 对数据加密后发送给A。A收到数据后，用其私钥 $SK_A$ 解密，以为和B进行了保密通信。其实，B发送给A的加密数据已被C截获并解密了一份，但A和B却都不知道



# 认证

---

认证：验证通信对端是期望的实体而不是假冒者

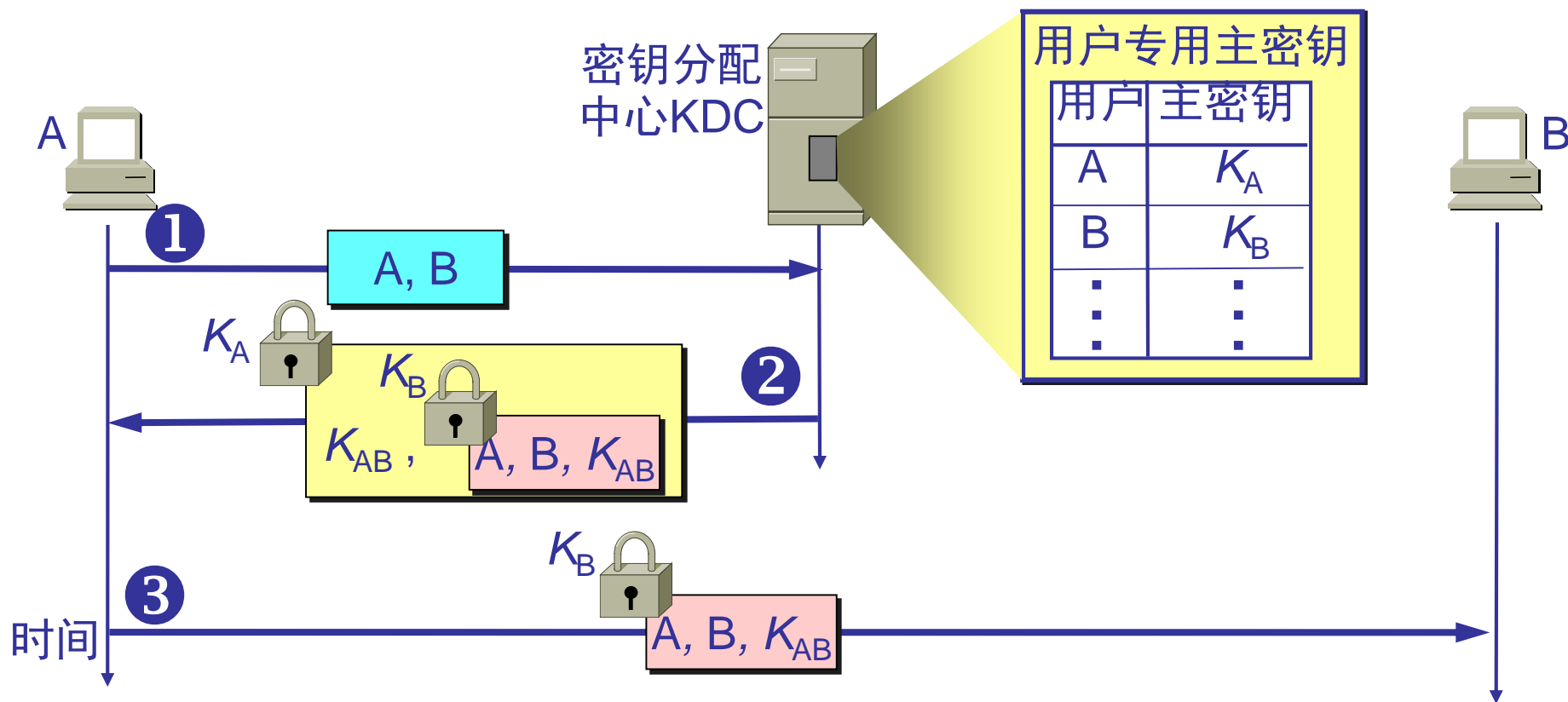
认证方法：

- 基于共享密钥的认证
- 基于公开密钥的认证
- 使用密钥分发中心的认证
- 使用Kerberos（网络认证协议，客户和服务端可相互均认证对方的）的认证
- 认证中心及证书



# 认证：用KDC分配对称密钥

- 设立KDC(Key Distribution Center), 应对中间人攻击
- 用户A和B是KDC用户, 并已经在KDC上安装与KDC通信的**主密钥** $K_A$ 和 $K_B$ , 简称为“密钥”
- 由KDC给需要秘密通信的用户分配一个临时密钥 $K_{AB}$



# Kerberos: 认证协议

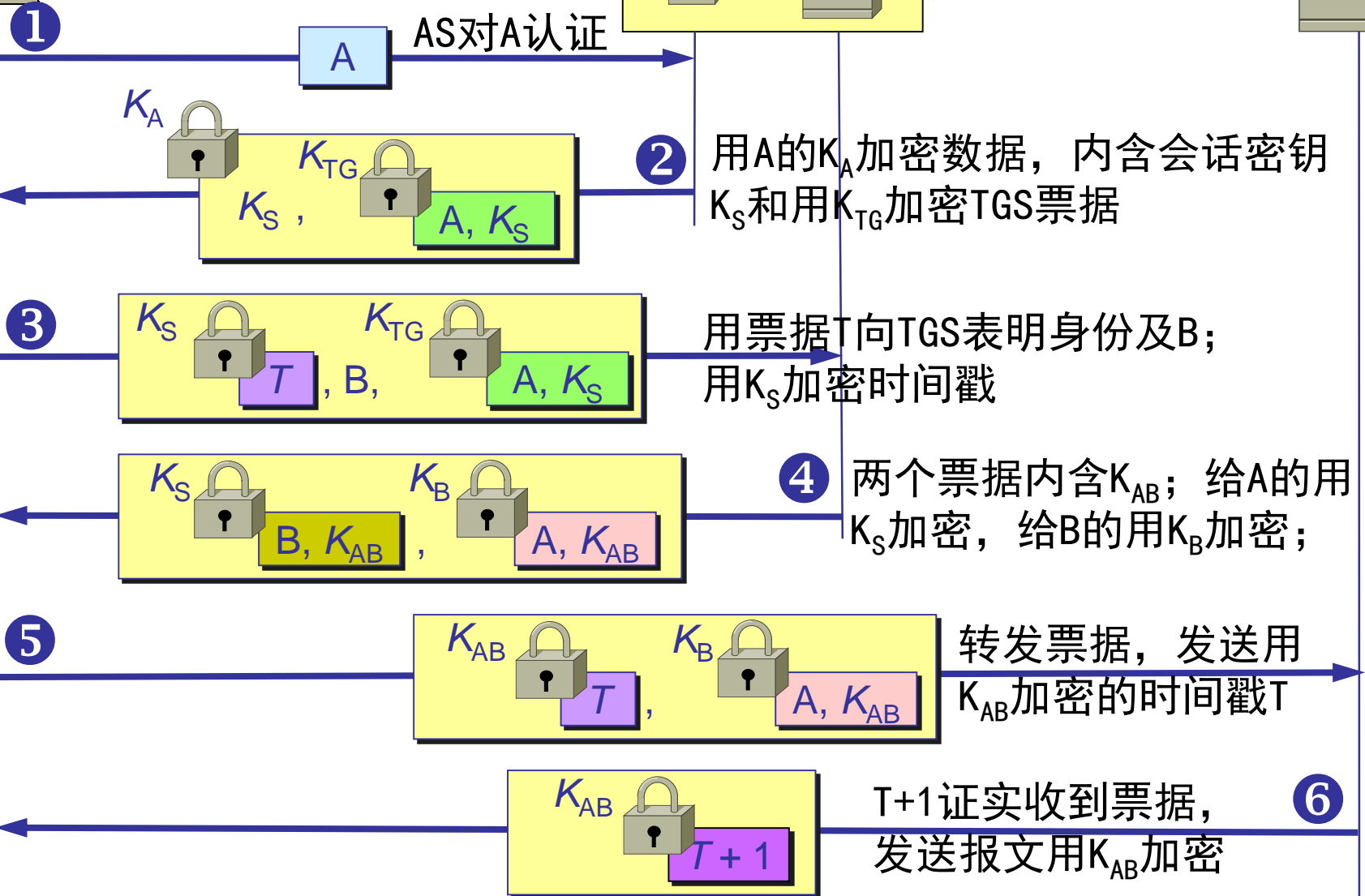
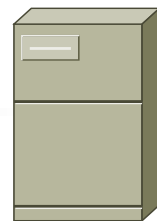


AS

TGS

AS: 认证服务器  
TGS: 票据授予服务器

B





# 认证中心与证书

- **认证中心CA** (Certification Authority)将实体（人或机器）的公钥与实体进行**绑定**
- 每个实体都有CA签发的**证书**，内含公钥及其拥有者的标识，此证书经CA进行数字签名
- 证书的标准：X.509（实体名称、签名算法、公钥、有效期、CA对证书的签名等）
- 用户从可信的地方获得CA的公钥，以及某个实体的证书
- 若用户要伪造用户B的证书，因证书未被CA签名而无法通过认证
- CA不必总是在线，避免KDC成为通信的瓶颈



# 提纲

---

- 信息传输安全的概念
- 加密技术
- 数字签名
- 访问控制：认证
- 安全的传输协议：
  - 网络层IPSec
  - 传输层SSL
  - WLAN 安全协议
- 应用层安全



# 网络层安全协议

---

**IPsec:** 在IP数据报中的数据都是加密的，包括两部分

- **认证首部AH**(Authentication Header): 使接收节点根据AH认证源节点并检查数据完整性，但不提供数据保密性
- **封装安全有效载荷ESP**(Encapsulation Security Payload): 使接收节点认证源点、检查数据完整性，并提供数据保密性

# 认证首部协议 AH

- 增加AH首部，IP首部中的协议字段为51
- 路由器不检查AH；目的主机处理AH字段，用以认证源点和检查数据报的完整性
- AH首部
  - (1) 下一个首部，标志本首部的下一个首部的类型（如TCP或UDP）
  - (2) 安全参数索引SPI：标志安全关联
  - (3) 序号：占32位，每个数据报的序号唯一，创建SA时为0，AH用该序号避免重放攻击；若序号回绕，则重建SA
  - (4) 认证数据(可变)：包含经数字签名的报文摘要，用来认证源主机和检查IP数据报的完整性

IP 首部

AH 首部

TCP/UDP 报文段

协议 = 51

# 封装安全有效载荷ESP

- IP首部协议字段为50，增加ESP尾部和ESP数据
- 在ESP首部中，有安全参数索引SPI和序号
- 在ESP尾部中有下一个首部（作用与AH的相同），数据及ESP尾部一起被加密，因此攻击者无法得知传输层协议
- ESP认证：与AH中的认证数据相同
- ESP：既认证源站和检查数据报完整性，又提供了保密





# 安全关联 SA(Security Association)

- 在使用AH或ESP之前，先从源主机到目的主机建立一条安全关联SA，并有一个共享密钥
  - IPsec把无连接的网络层转换为有逻辑连接的网络层
  - 对于一个SA，每个IPsec数据报都有一个SPI字段，通过此SA的数据报使用同样的SPI
- SA是一个单向连接，由一个三元组唯一确定，包括：
  - (1) 安全协议（使用AH或ESP）标识符
  - (2) 连接的源IP地址
  - (3) 一个连接标识符，称为安全参数索引SPI (Security Parameter Index)
- 如何获得密钥，如何协商加密算法？
  - 方法1：由系统管理员人工配置主机的密钥
  - 方法2：源主机和目的主机利用互联网密钥交换自动获得密码算法和密钥

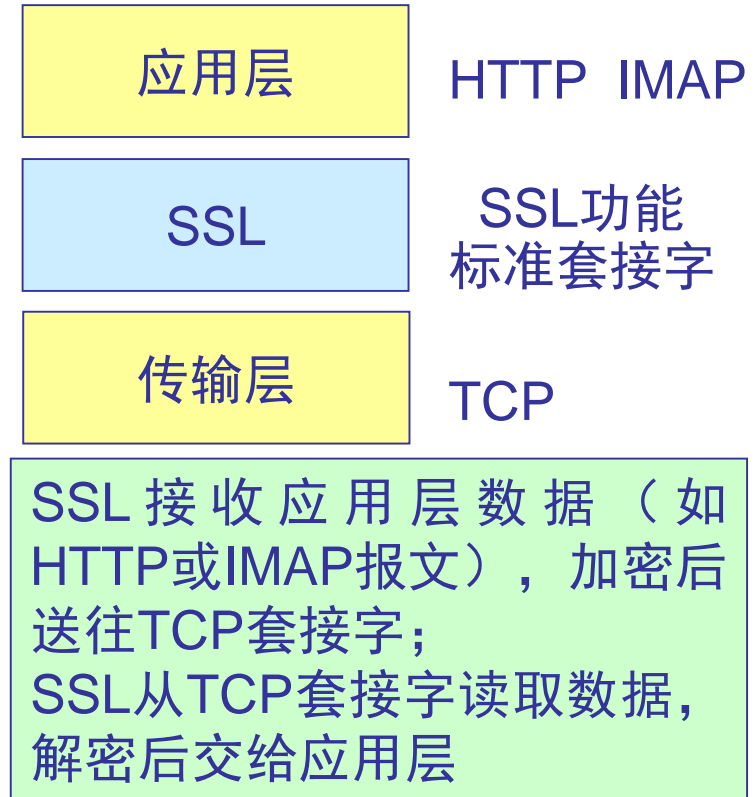


# 传输层安全协议

## 安全套接层SSL

(Secure Socket Layer)

- 对Web客户与服务器之间传送的数据进行加密和认证
- 在连接建立阶段协商加密算法和密钥，以及客户与服务器的认证
- 传送的数据使用商定的会话密钥
- 浏览器和Web服务器支持SSL
  - HTTPS（HTTP over SSL），是在HTTP下加入SSL层





# 传输层安全协议

## SSL提供的三个功能

- (1)认证服务器：用户证实服务器的身份。支持SSL的浏览器维持一个表，保存可信赖的认证中心CA和服务器的公钥
- (2)加密会话数据：客户和服务器的交互数据，在发送方加密，在接收方解密
- (3)认证客户：服务器证实客户的身份，利用CA的证书

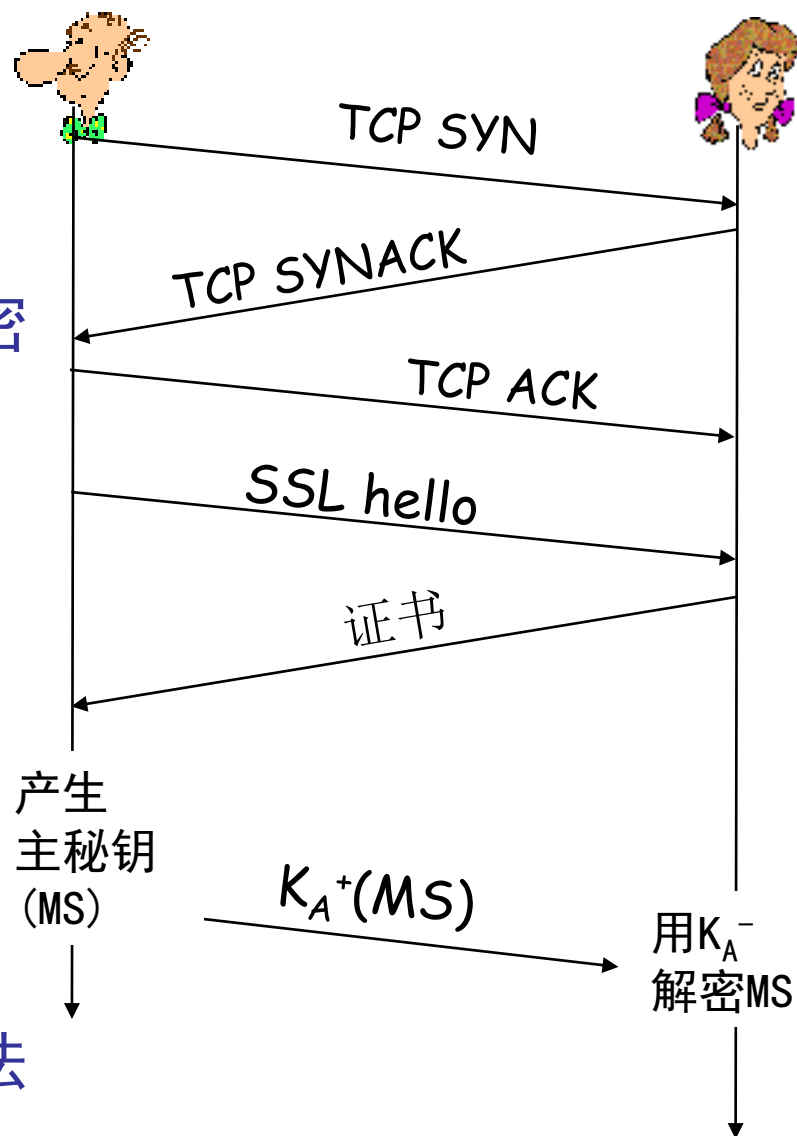
# SSL: 三次握手

## 1. 握手

- Bob与Alice建立TCP连接
- 通过CA签发的证书认证Alice
- 生成密钥MS, 用Alice的公钥加密MS并发送给 Alice
  - 改变不重数, 未显示

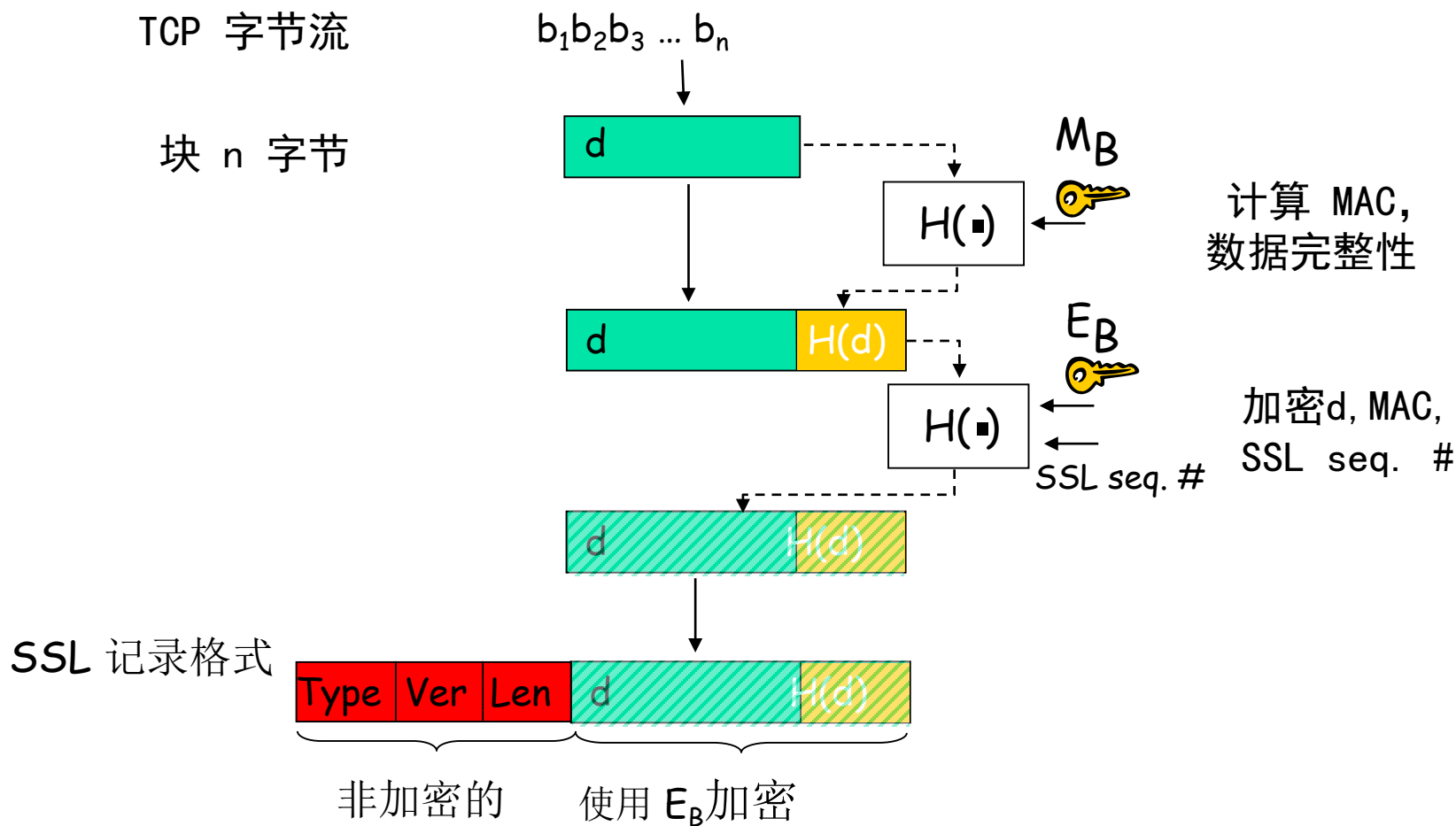
## 2. 密钥导出:

- Alice, Bob用MS产生4个密钥
  - $E_B$ : Bob→Alice 数据加密密钥
  - $E_A$ : Alice→Bob 数据加密密钥
  - $M_B$ : Bob→Alice MAC密钥
  - $M_A$ : Alice→Bob MAC密钥
- 两者之间协商加密算法和MAC算法
- 为何需要四个密钥?



# SSL: 三次握手

## 3. 数据传输





# IEEE 802.11 安全性

---

- 通过WLAN接入互联网很普遍，实例
  - 校园网、公共场所
  - 有WIFI吗？免费蹭网
- 面临安全传输问题
  - 无线信号在空间传输捕获无线帧
  - 接入不安全的AP！
- 安全的802.11
  - 加密，认证
  - 802.11 安全性：有线等效保密（WEP: Wired Equivalent Privacy）
  - 802.11i



# 有线等效保密(WEP: Wired Equivalent Privacy)

- 在主机和AP之间提供认证和数据加密
- 认证过程
  - 无线主机通过AP请求认证
  - AP回应一个128字节的不重数
  - 无线主机用一个与AP共享密钥加密该不重数
  - AP解密该不重数
- 若解密的不重数与之前的不重数相同，则通过认证
- 没有密钥分配机制
- 知道共享密钥即可以通过认证
- 问题：
  - 每个主机的共享密钥是否不同？
  - 与用户密码有关系吗？

# WEP 数据加密

- 主机/AP 共享40b对称密钥 (半永久性的, 很少变化)
- 主机附加24b初始向量 (IV) 产生64b密钥
- 用64b密钥生成密钥流  $k_i^{IV}$
- 用  $k_i^{IV}$  加密第  $i$  字节  $d_i$ :  $c_i = d_i \text{ XOR } k_i^{IV}$
- 在发送帧中有 IV 和加密的字节  $c_i$

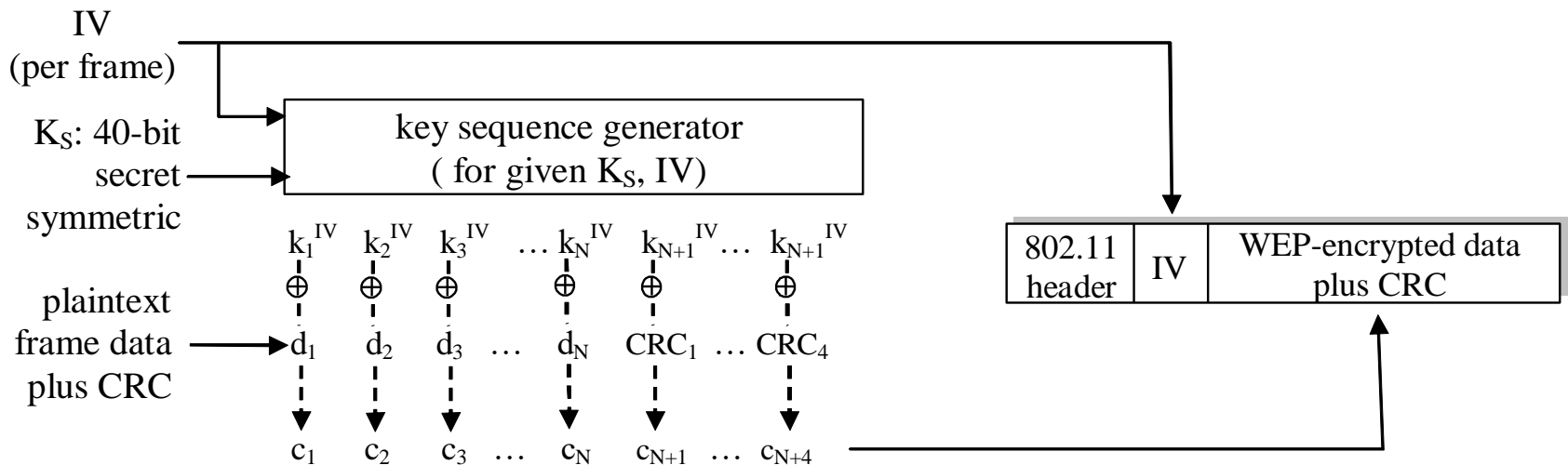


Figure 7.8-new1: 802.11 WEP protocol



# 破解 802.11 WEP 加密

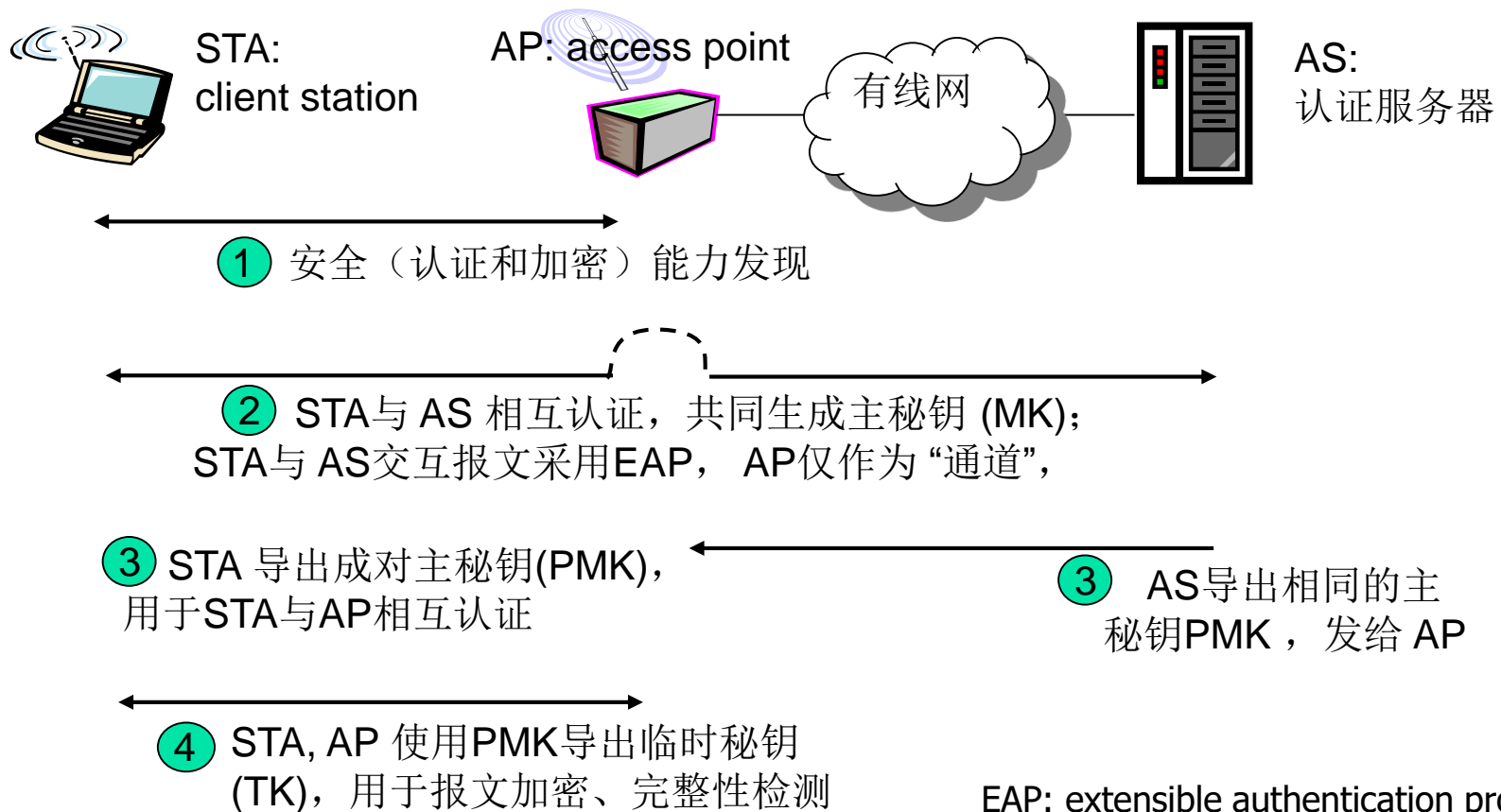
## 安全漏洞:

- 帧中的24b IV，每帧改变一次，导致IV经过一段时间后重复使用
- 在帧中用明文传输IV，导致IV重用检测
- 攻击:
  - Trudy 让 Alice 发送已知明文  $d_1 d_2 d_3 d_4 \dots$
  - Trudy 看到密文:  $c_i = d_i \text{ XOR } k_i^{\text{IV}}$
  - Trudy 已知  $c_i d_i$ ，所以计算  $k_i^{\text{IV}}$
  - Trudy 知道加密密钥序列  $k_1^{\text{IV}} k_2^{\text{IV}} k_3^{\text{IV}} \dots$
  - 一段时间之后，IV被重用，则Trudy可以解密!



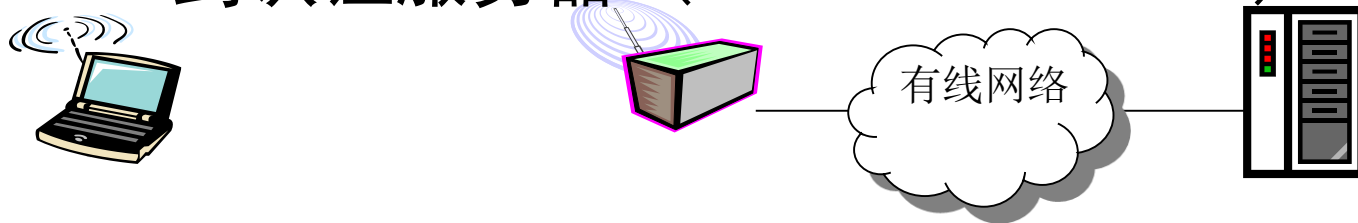
# 802.11i: 改进安全性

- 使用认证服务器AS: AS与AP分离, 一个AS支持多个AP
- 提供密钥分发, 更强的加密机制



# EAP: extensible authentication protocol

- EAP: 可扩展的认证协议, 为STA到认证服务器的协议
- TLS: Transport Layer Security
- EAP经过不同的链路传输
  - STA到AP (EAP over WLAN)
  - AP 到认证服务器 (RADIUS over UDP)



EAP TLS	
EAP	
EAP over LAN (EAPoL)	RADIUS
IEEE 802.11	UDP/IP



# 提纲

---

- 信息传输安全的概念
- 加密技术
- 数字签名
- 访问控制：认证
- 安全的传输协议：
  - 网络层IPSec
  - 传输层SSL
  - WLAN 安全协议
- 应用层安全



# 应用层安全问题

---

- 网络层安全传输无法保证服务的安全性
- 在应用层上，更容易实现某些安全技术
- 例如：
  - CA及证书、KDC、公钥分配协议是在应用层；
  - 安全的邮件服务
  - Web安全性问题及DNS欺骗



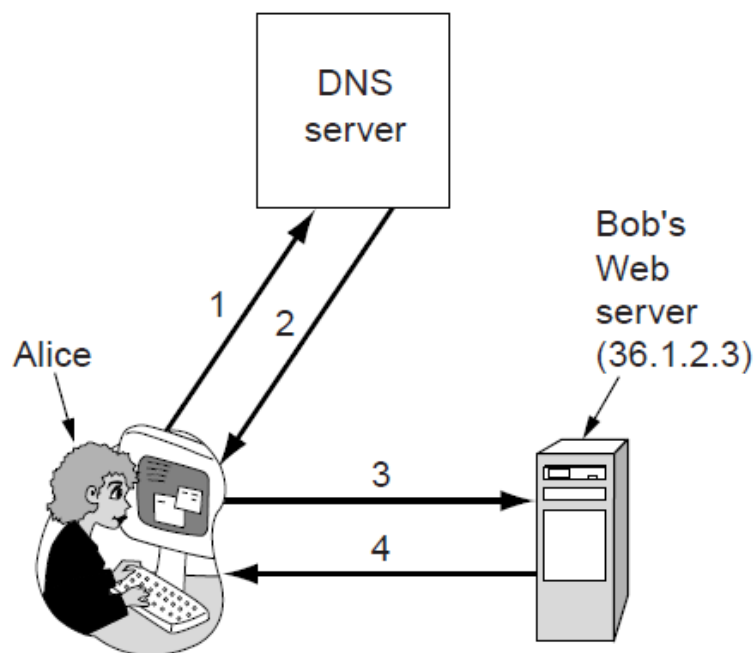
# 电子邮件安全

---

- 安全性：认证发送方，提供内容保密性及完整性
- 电子邮件证书：证明电子邮件身份的标识，分别使用证书中的公钥和私钥对邮件加密和签名
- **邮件加密**：发件人用接收者证书中的公钥对邮件内容及附件加密；只有接收者才能阅读，其他人截获该邮件看到的只是乱码
- **邮件签名**：发送者使用其证书的私钥对邮件签名；接收者验证邮件签名以及签名者的证书来验证邮件是否被篡改，并判断发送者的身份；确保电子邮件的真实性和完整性。
- PGP (Pretty Good Privacy)：由Phil Zimmermann于1991年创立的电子邮件安全软件包，包括加密、认证、电子签名和压缩等技术；综合了MD5，RSA以及IDEA等算法；已被广泛使用，成为事实的标准。

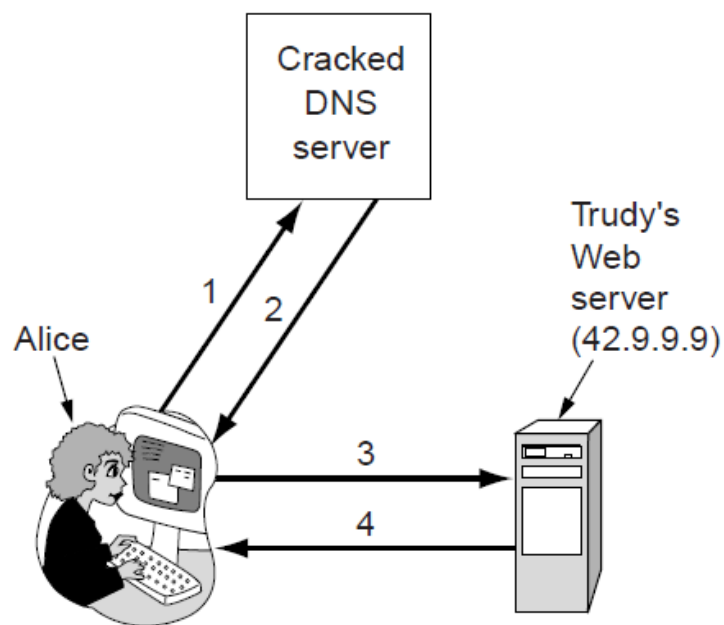
# Web安全性问题

## 正常情况



1. Give me Bob's IP address
2. 36.1.2.3 (Bob's IP address)
3. GET index.html
4. Bob's home page

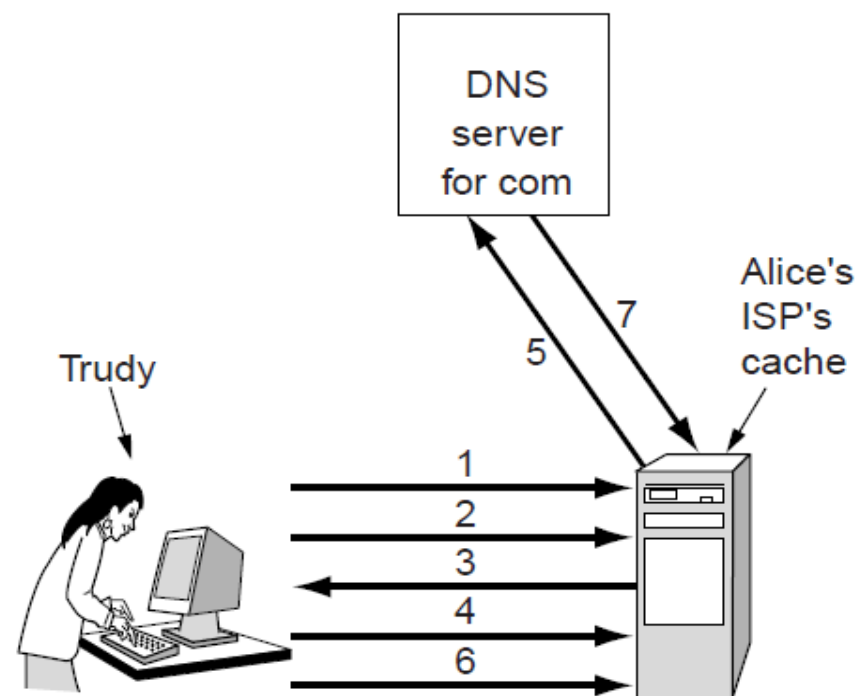
## 被攻陷的DNS服务器 修改了Bob记录



1. Give me Bob's IP address
2. 42.9.9.9 (Trudy's IP address)
3. GET index.html
4. Trudy's fake of Bob's home page

# 利用域名解析假冒对方

- Trudy 如何欺骗Alice的ISP?
- 假设Alice的ISP无Bob.com DNS项或已过期，Trudy假冒对Bob.com的IP地址请求并冒充顶级DNS回答——**DNS欺骗，染毒缓存**
- Trudy先注册一个DNS.Trudy-the-intruder.com域，IP为42.9.9.9



1. 查询foobar.Trudy-the-intruder.com (迫使它进入ISP缓存)
2. 查询[www.Trudy-the-intruder.com](http://www.Trudy-the-intruder.com) (目的是为获得ISP的下一个序号)
3. 请求[www.Trudy-the-intruder.com](http://www.Trudy-the-intruder.com) (携带ISP的下一个序号，n)
4. 快速地查询Bob.com (迫使ISP在第5步查询.com服务器)
5. 用序号n+1合法地查询Bob.com
6. Trudy伪装.com顶级DNS回答Bob.com是42.9.9.9，发送多个序号n+1递增报文
7. 真实回答太晚了，被拒绝



# DNSsec

---

- 安全的DNS DNS sec
  - 序号ID随机，不要递增
  - DNS发送的信息用私钥签名，接收方验证其真实性





# 知识点

---

- 信息安全的含义是什么？
- 什么是数据加密？
- 什么是数字签名？
- 什么是认证？
- 常见的网络安全传输协议



# 练习题

- 假设一台机器位于NAT盒子之后，试问：IPSec还能使用传输模式AH吗？为何？
- 主机A与B之间使用APSec发送分组流，要为每个分组创建一个SA吗？
- 假设在IPSec之上运行TCP，若重传一个TCP报文，则这两个分组的AH首部序号相同吗？
- A与B通过SSL通信，假设一个没有共享密钥的攻击者在会话流中插入一个伪造的TCP报文段。问这个报文段在接收端会被传递给有效载荷吗？为什么？
- 假设certifier.com为foo.com生成一份证书，通常该证书将用certifier.com的公钥还是私钥加密？为何？
- 在WLAN链路上采用WEP加密。帧长为1KB，链路速率为11Mbps。问大约经过多长时间会出现重复的IV。



# 通知

---

- 上机实习题目提交截止时间：5月31日
- 6月3日（周一）：复习及答疑
- 6月10日（周一上午）：期末考试

