1 散列函数Hash Functions

散列函数定义

- 散列函数H是一个公开函数，将任意长的消息映射为较短的、固定长度的一个值H(M)。
- H(M)称为散列值、消息摘要，是消息中所有比特的函数，提供了错误检测的能力。
- 散列函数的目的是为文件、报文或其它的分组数据产生“数字指纹”。

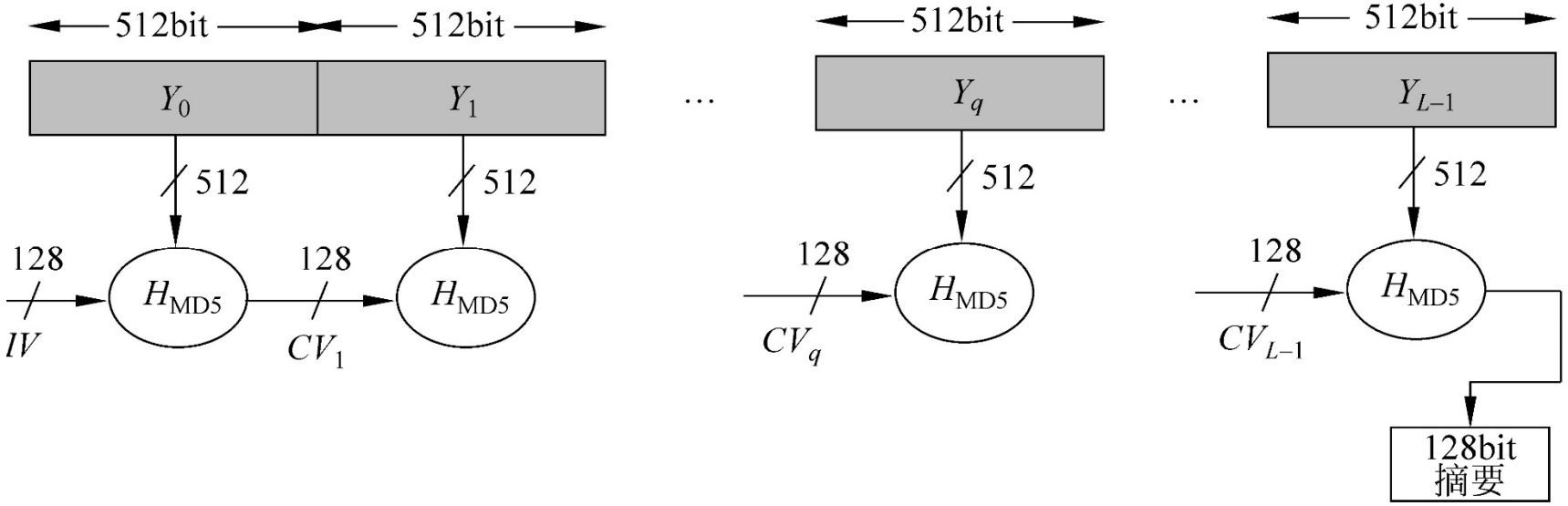
1 散列函数Hash Functions

- MD5是由国际著名密码学家、“图灵奖”获得者兼公钥加密算法RSA的创始人、麻省理工大学的Ronald Rivest教授于1991年设计的。
- SHA-1是由美国国家标准技术研究院（NIST）与美国国家安全局（NSA）设计，1993年成为联邦信息处理标准(FIPS PUB 180)。
- 两大算法是目前国际电子签名及许多其它密码应用领域的关键技术，广泛应用于金融、证券等电子商务领域。

1 散列函数Hash Functions

输入:任意长的消息
分组:512比特
输出:128比特

- MD5散列函数



- 分组填充
- 缓冲初始化　$CV_0 = IV$（使用ABCD4个32比特长寄存器）
- MD5运算　$CV_{q+1} = \mathrm{MD5}[Y_q, CV_q]$
- 输出　$CV = CV_L$（最终的散列值）

1 散列函数Hash Functions

MD5 算法主要流程：

1.对于输入的字符串，按位填补一个1以及若干个0使得填补后的位数为 *N**512+448, *N*>=0，再添加一个64位的数字为原始长度，使得最后的字符串变为(*N*+1)*512位。

3.初始化四个32位的数，A=0x67452301; B=0xefcdab89; C=0x98badcfe; D=0x10325476;

4.将字符串分成*N*+1块，每块512位，循环*N*+1次，对于每次循环:

1）令*a*=A; *b*=B; *c*=C; *d*=D;

2）将当前的512位转为16个32位的数*M*[0~15]。

3）进行64轮操作。

4）A+=*a*;B+=*b*;C+=*c*;D+=*d*;

5.输出ABCD的级联。

1 散列函数Hash Functions

处理位操作函数

F(X,Y,Z)=(X&Y)|((~X)&Z)

G(X,Y,Z)=(X&Z)|(Y&(~Z))

H(X,Y,Z)=X^Y^Z

I(X,Y,Z)=Y^(X|(~Z))

(&是与,|是或,~是非,^是异或)

1 散列函数Hash Functions

主要变换操作：

设M_j表示消息的第j个子分组（从0到15），<<s表示循环左移s位，则四种操作为：

$FF(a,b,c,d,M_j,s,t_i)$表示

$$a=b+((a+F(b,c,d)+M_j+t_i)<<s)$$

$GG(a,b,c,d,M_j,s,t_i)$表示

$$a=b+((a+G(b,c,d)+M_j+t_i)<<s)$$

$HH(a,b,c,d,M_j,s,t_i)$表示

$$a=b+((a+H(b,c,d)+M_j+t_i)<<s)$$

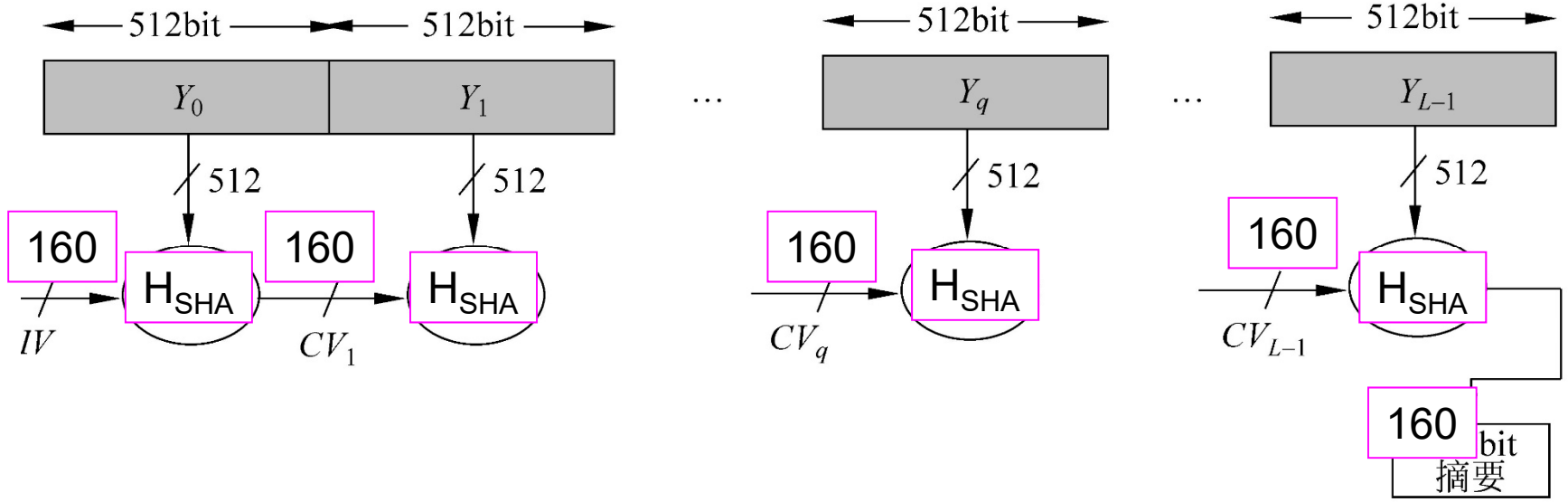
$II(a,b,c,d,M_j,s,t_i)$表示

$$a=b+((a+I(b,c,d)+M_j+t_i)<<s)$$

1 散列函数Hash Functions

输入:任意长的消息
分组:512比特
输出:160比特

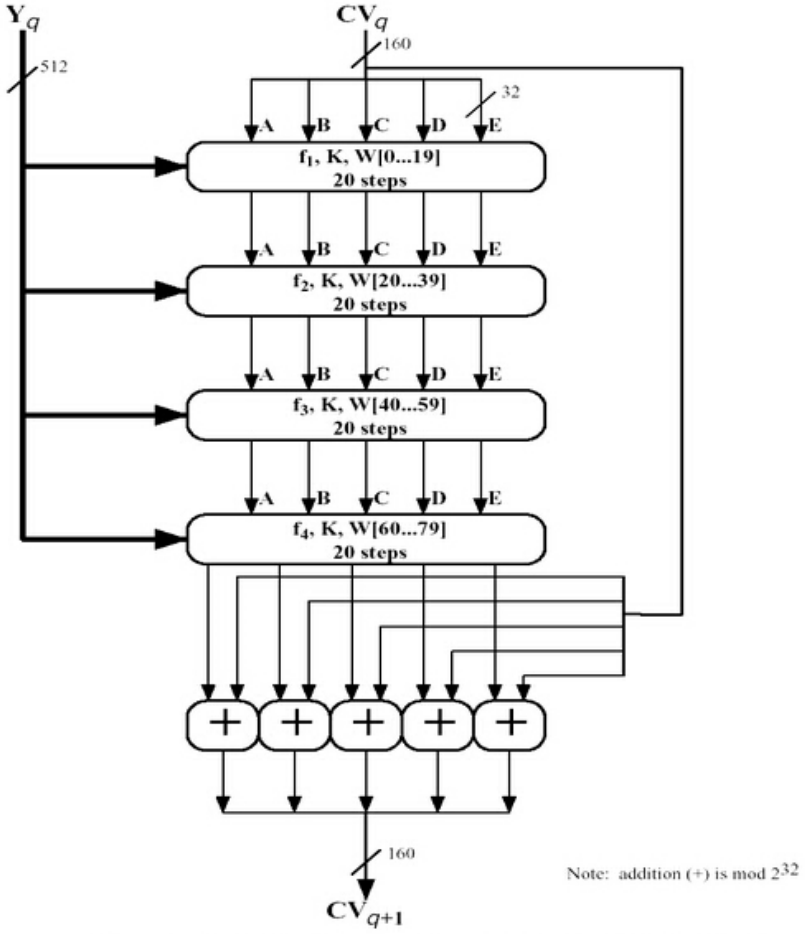
- SHA: Secure Hash Algorithm



- 分组填充
- 缓冲初始化　　$CV_0=IV$（使用ABCDE5个32比特长寄存器）
- SHA运算　　$CV_{q+1}=SHA[Y_q, CV_q]$
- 输出　　$CV=CV_L$（最终的散列值）

1 散列函数Hash Functions

- SHA: Secure Hash Algorithm



1 散列函数Hash Functions

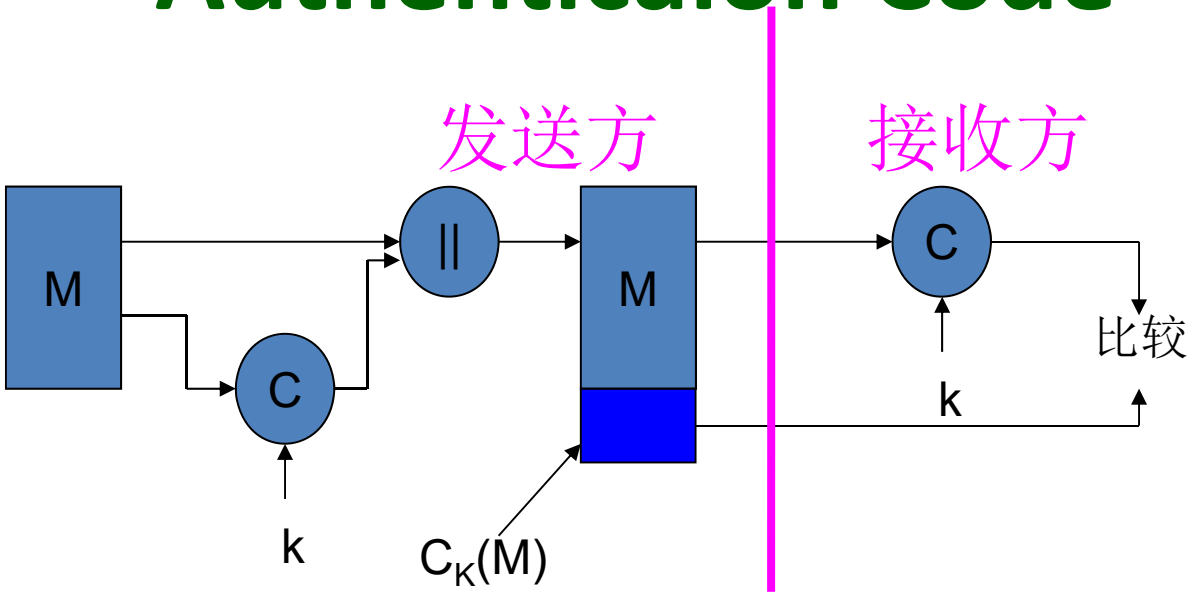
MD5与SHA比较：

- 抗穷搜索能力
  - 寻找指定散列值： SHA：$O(2^{160})$，MD5：$O(2^{128})$
  - 生日攻击：SHA：$O(2^{80})$，MD5：$O(2^{64})$
- 抗密码分析攻击的强度：SHA似乎高于MD5
- 速度：SHA较MD5慢
- 简捷与紧致性：描述都比较简单，都不需要大的程序和代换表

2 消息认证码Message Authenticaion Code

- 消息认证码：消息M被一密钥k控制的公开函数C作用后产生的、用于认证的、固定长度的数值$C_k(M)$，也称为密码校验和。
- 通信双方共享密钥k。
- 可理解为：带密钥的散列函数。

2 消息认证码Message Authenticaion Code



其中M是可变长的报文，k是共享密钥，$C_k(M)$是定长的消息认证码。

- 完整性：接收方相信发送方发来的消息未受篡改。
- 可用性：接收方相信发送方不是冒充的。

3 数字签名

消息认证码可以保护通信双方以防第三方的攻击，然而却不能防止双方中一方的欺骗或伪造。

欺骗场景①：

A

共享密钥k

M

$C_k(M)$

B

共享密钥k

B做：对消息M生成消息认证码$C_k(M)$

B说：该消息认证码是A生成的

3 数字签名

欺骗场景②：

A做：对消息M生成消息认证码$C_k(M)$并发送给B

A说：该消息认证码是B生成的，因为B也有共享密钥k

3 数字签名

- 类似于手书签名，数字签名应具有以下性质：

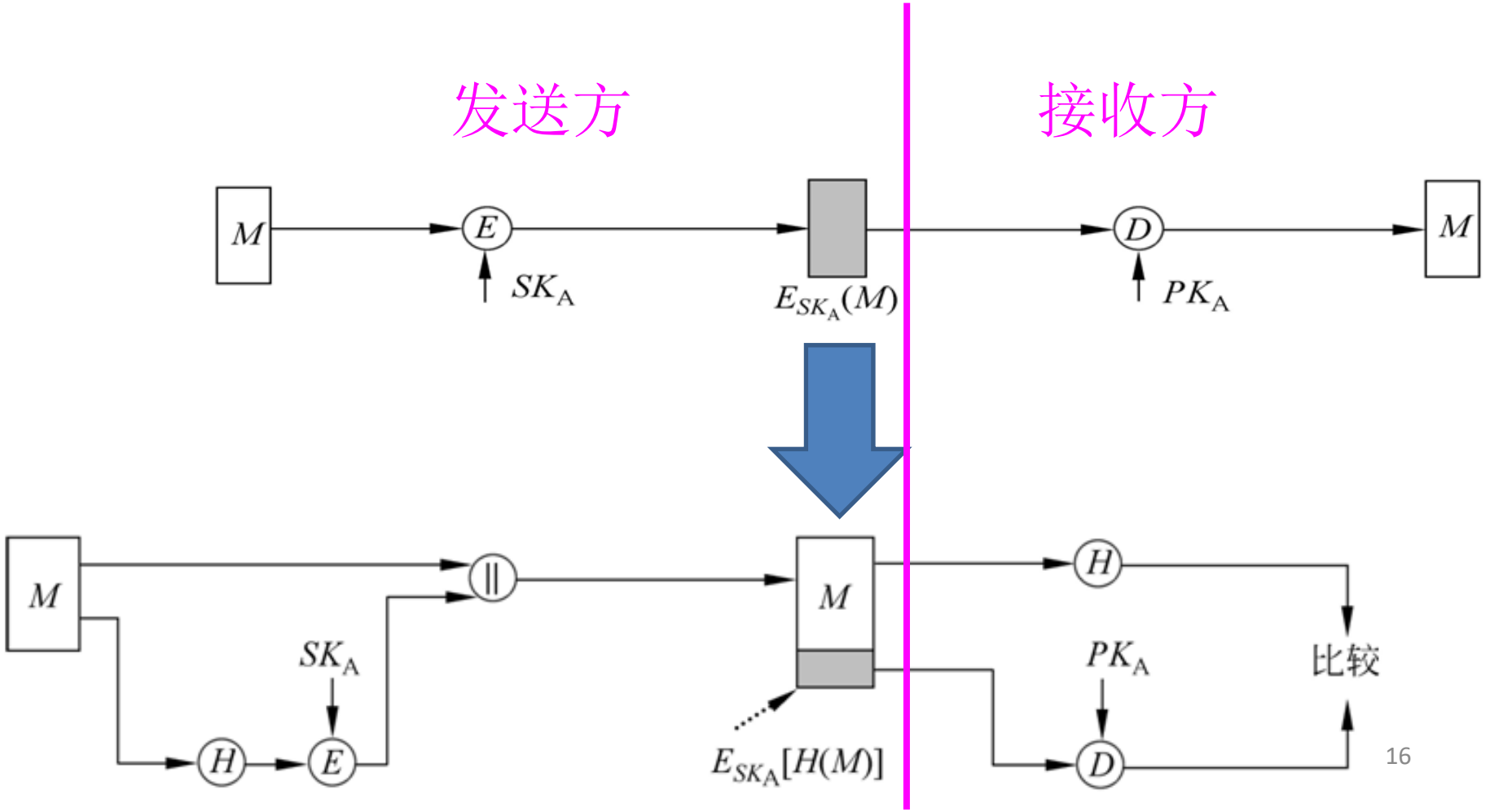
① 不可否认性：能够验证签名产生者的身份，接收方相信发送方不是冒充的，且发送方不能否认。

② 完整性：能用于证实被签消息的内容，接收方相信发送方发来的消息未受篡改。

③ 公开验证性：数字签名可由第三方验证，从而能够解决通信双方的争议。

3 数字签名

思考：利用加密算法，私钥签名，公钥验证。



3 数字签名

DSA是在ElGamal和Schnorr两个签名方案的基础上设计的，其安全性基于离散对数难题。生成签名长度320 bit，算法描述如下：

- 秘钥产生算法Gen： p：满足$2^{L-1}<p<2^{L}$的大素数，其中$512\le L\le 1024$且L是64的倍数。q：p-1的素因子，满足$2^{159}<q<2^{160}$，即q长为160比特。g：$g=h^{(p-1)/q} \bmod p$，h是满足$1<h<p-1$且使得$h^{(p-1)/q} \bmod p>1$的任一整数。用户私钥x是满足$0<x<q$的随机数。用户公钥$y\equiv g^{x} \bmod p$。

3 数字签名

- 签名算法Enc：用户为待签消息选取随机数k满足$0<k<q$。用户对消息M的签名为(r, s)，其中$r\equiv(g^k \bmod p) \bmod q$，$s\equiv[k^{-1}(\mathrm{H(M)}+xr)] \bmod q$，H(M)是由SHA求出的散列值。
- 验证算法Verify：设接收方收到的消息为M′，签名为(r′, s′)。计算$w\equiv(s')^{-1} \bmod q$, $u_1\equiv[\mathrm{H(M')}w] \bmod q$，$u_2\equiv r'w \bmod q$, $v\equiv[(g^{u1}y^{u2}) \bmod p] \bmod q$。检查$v=r'$是否成立，若相等，则认为签名有效。

3 数字签名

正确性验证：

若(M′, r′, s′)=(M, r, s)，则：

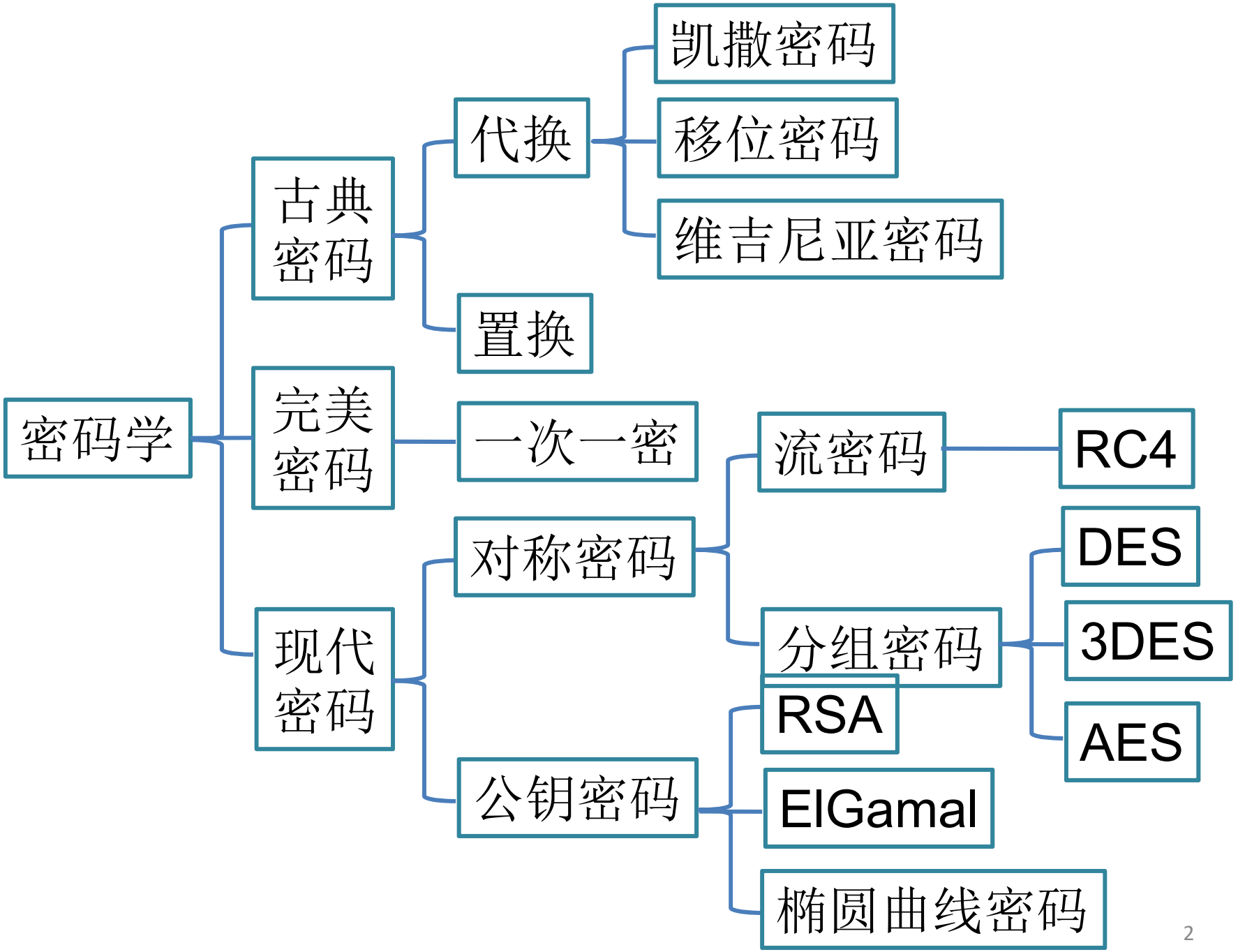
$$v \equiv [g^{H(M)w} g^{xrw} \bmod p] \bmod q$$

$$\equiv [g^{(H(M)+xr)s^{-1}} \bmod p] \bmod q$$

$$\equiv [g^{k} \bmod p] \bmod q$$

$$\equiv r$$

安全性：由于离散对数的困难性，敌手从r恢复k或从s恢复x都是不可行的。


密码学
古典密码
代换
凯撒密码
移位密码
维吉尼亚密码
置换
完美密码
一次一密
现代密码
对称密码
流密码
RC4
分组密码
DES
3DES
AES
公钥密码
RSA
ElGamal
椭圆曲线密码

目　录

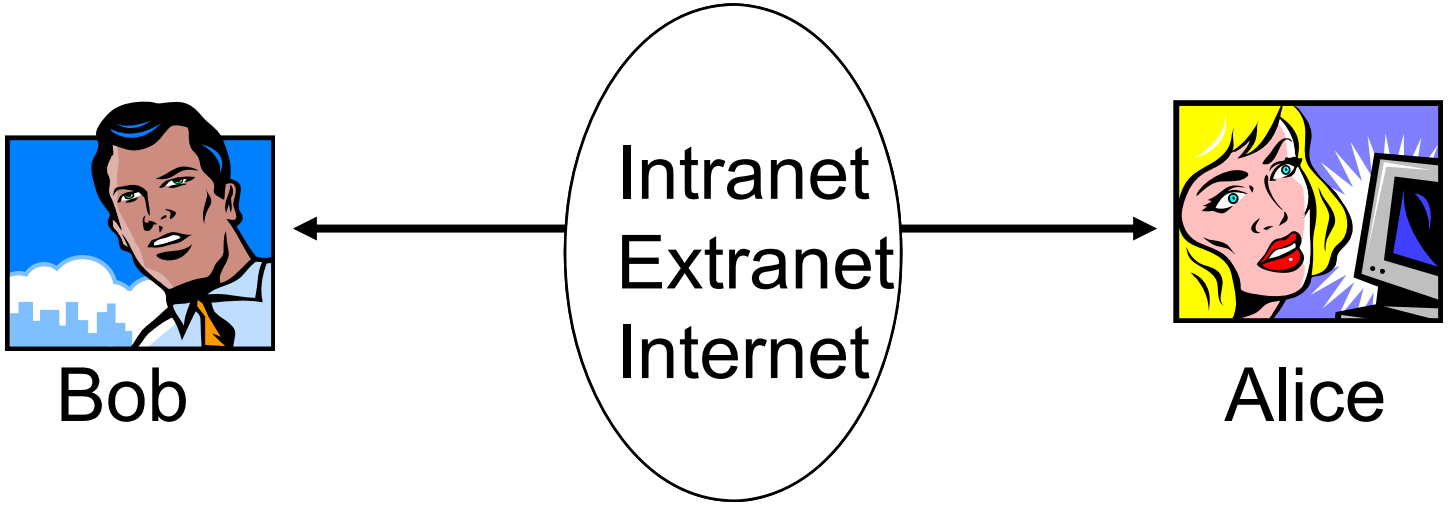
1. 网络与信息安全概述

2. 密钥管理和分发

3. PKI概述

4. 用户认证

1. 网络与信息安全概述



Bob and Alice want to exchange data in a digital world.

There are Confidence and Trust Issues …

1. 网络与信息安全概述

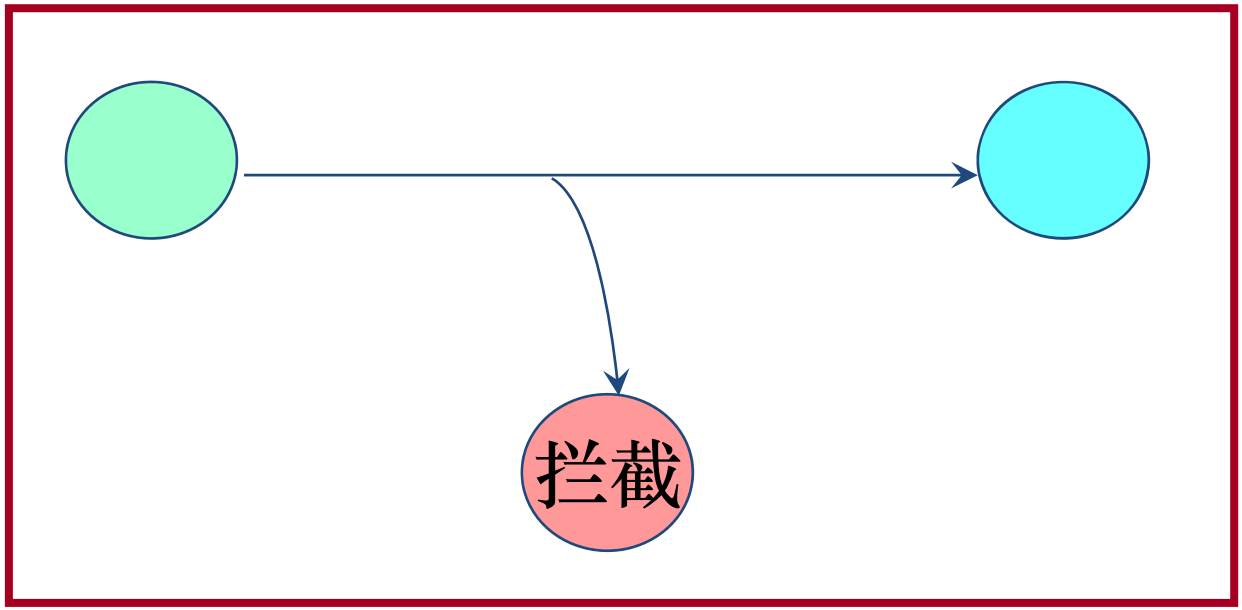
如何保证网络上的通信信息安全?

- 使用**LAN/Internet . . .**
  - 发送邮件
  - 下载软件
  - 发送敏感的或私有的数据
  - 访问应用系统
- 但人们担心的是 . . .
  - 如何确认某人的身份?
  - 如何知道我连接的是一个可信的站点?
  - 怎样才能保证我的通讯安全?
  - 怎样确定电子信息是否被篡改?
  - 如何证明某人确实给我发过电子邮件?

1. 网络与信息安全概述

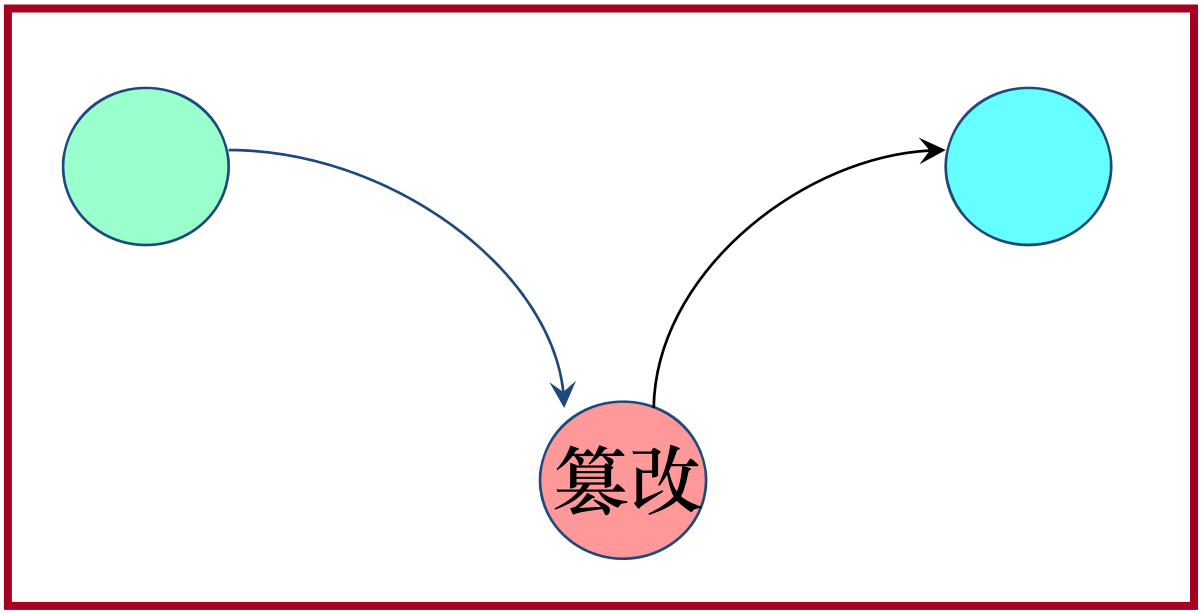
Internet上没人知道你是一只狗

1. 网络与信息安全概述



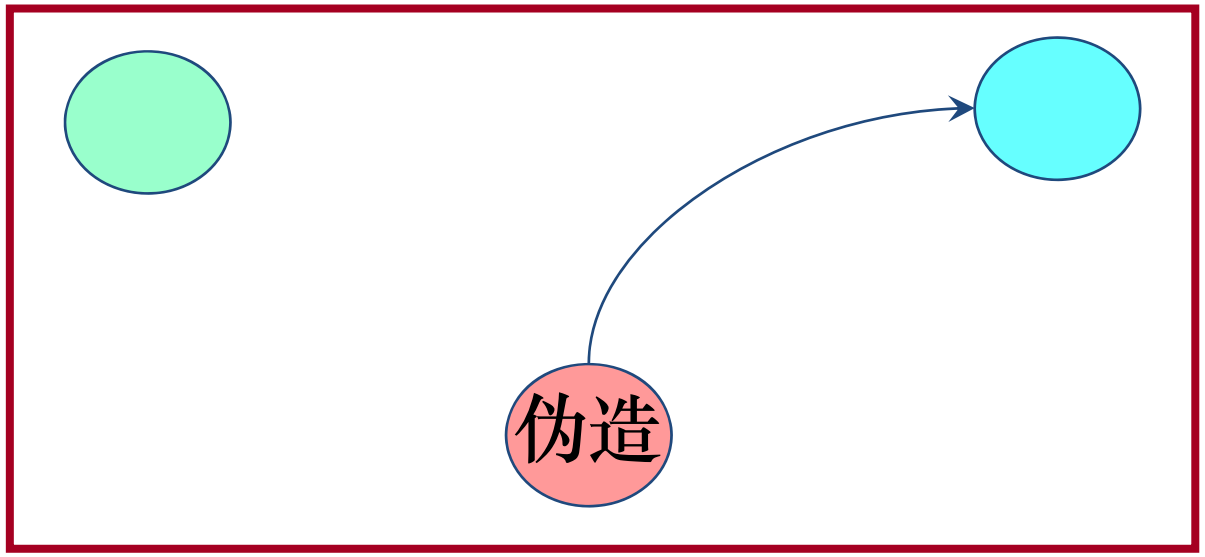
机密性：通讯信息是否被泄露?

1. 网络与信息安全概述



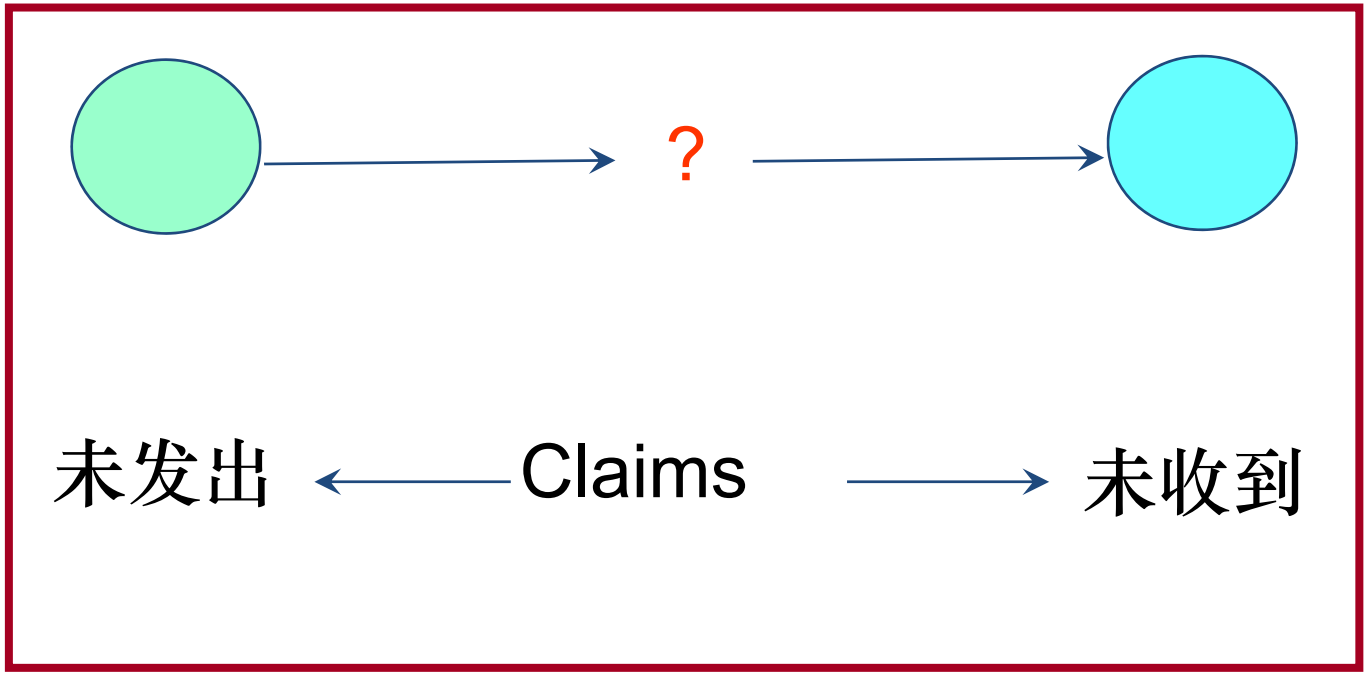
完整性：通信信息是否被篡改?

1. 网络与信息安全概述



可认证性（可用性）：我在与谁通讯?能否确认对方的身份。

1. 网络与信息安全概述



不可否认性：是否发出/收到信息?

1. 网络与信息安全概述

信息安全最基本的目标：

机密性CONFIDENTIALITY

That the information will be kept Private

完整性INTEGRITY

That information cannot be Manipulated

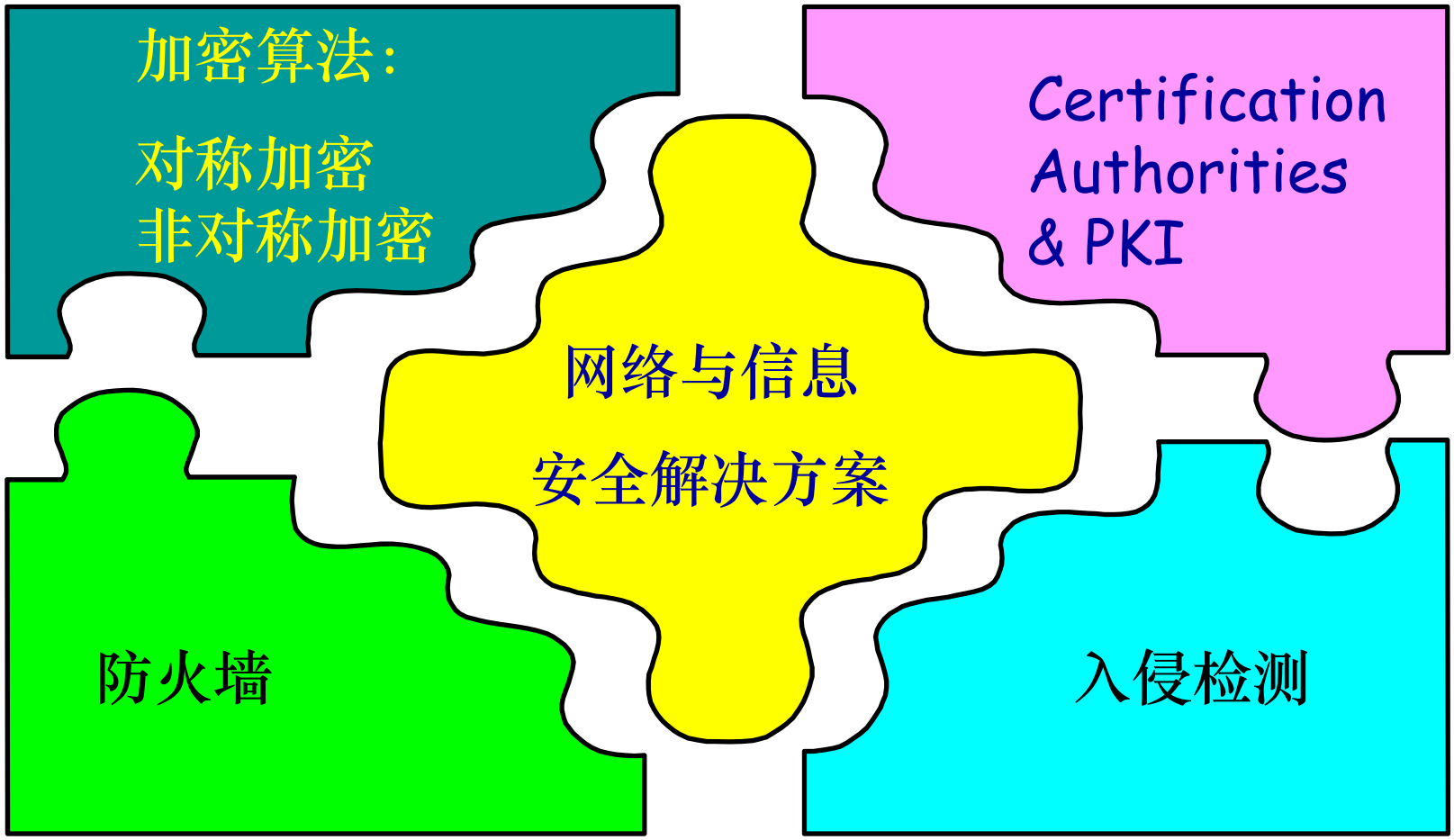
可认证性/可用性AUTHENTICATION

In the Identity of an Individual or Application

不可否认性NON-REPUDIATION

That information cannot be Disowned

1. 网络与信息安全概述


加密算法：
对称加密
非对称加密
Certification
Authorities
& PKI
网络与信息
安全解决方案
防火墙
入侵检测

1. 网络与信息安全概述

所有的密码系统均基于以下三类算法:

消息摘要 *(MD2-4-5, SHA, SHA-1, …)*

将任意长的消息映射为较短的、固定长度的一个值。
不使用密钥，计算不可逆即不能从消息摘要恢复出消息。

对称密码算法 *(DES, IDEA, RC2-4-5, Triple-DES, MAC…)*

加解密使用相同的密钥。

公钥密码算法 *(DSA, RSA, …)*

加解密使用两个不同的密钥：一对公钥和私钥。

2. 密钥管理和分发

- 已知密钥，可加密与解密。然而加密者与解密者处于不同的地理位置，密钥如何分发？
- 任何密码系统的强度都与密钥分发方法有关。
- 密钥分发方法是指将密钥发送给希望交换数据的双方而不让别人知道的方法。

2. 密钥管理和分发

2.1 对称加密的对称密钥分发

1）A选择密钥，并亲自交给B。

2）第三方选择密钥，并亲自交给A和B。

Solutions that are based on private-key cryptography are not sufficient to deal with the problem of secure communication in open systems where parties cannot physically meet and/or have transient interactions.

2. 密钥管理和分发

2.1 对称加密的对称密钥分发

3）A和B有秘密渠道。

- 密钥分发问题：网络中有N个人，则每人需存储N-1个密钥。
- 密钥管理问题：密钥越多，存储空间越大，泄露的可能性越大。

(You can hide a needle in a haystack but it’s hard to hide thousands of needles in a haystack.)

- 不适用于开放环境

2. 密钥管理和分发

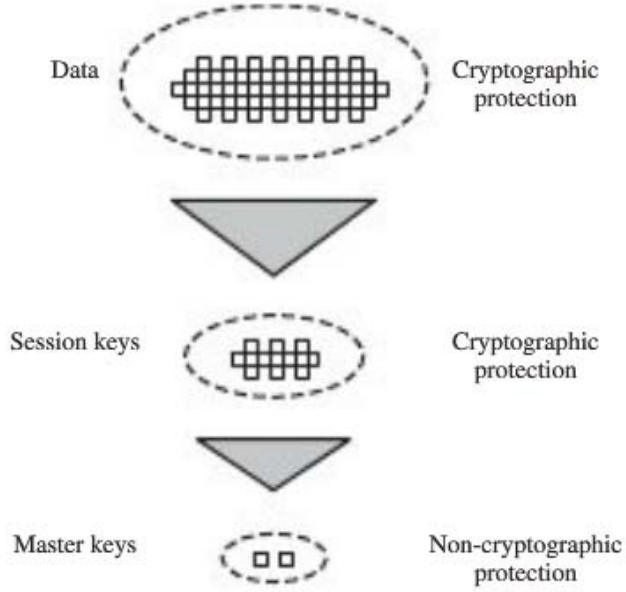
2.1 对称加密的对称密钥分发

4）A和B与第三方有秘密渠道。

- 密钥分发中心(key distribution center， KDC)
- 所有用户与KDC共享唯一的密钥

两个终端系统之间的通信使用会话密钥。

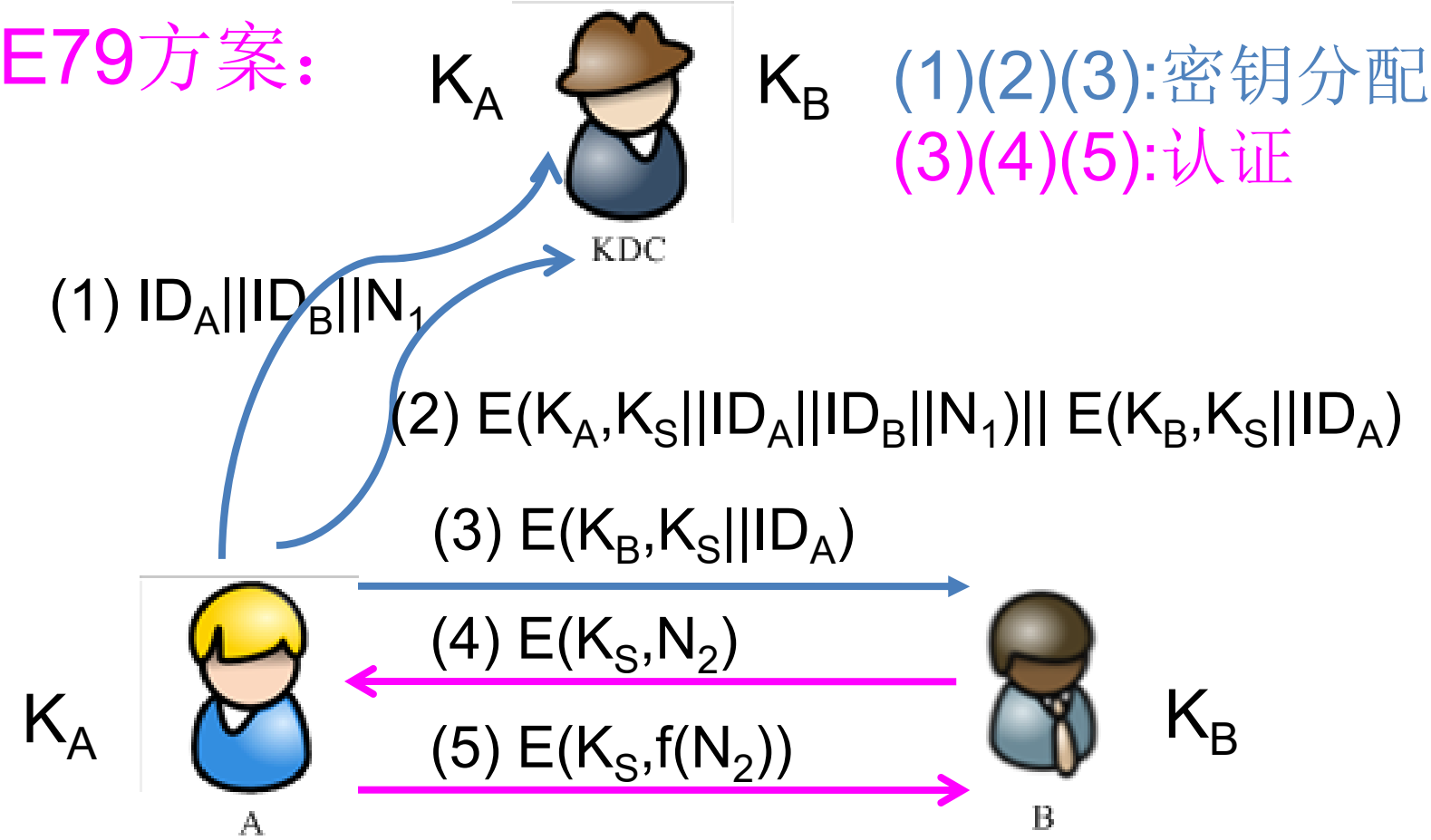
每一个终端系统和密钥分发中心共用唯一的主密钥。



2. 密钥管理和分发

2.1 对称加密的对称密钥分发

POPE79方案：



(1)(2)(3):密钥分配

(3)(4)(5):认证

- 1、A以明文形式向KDC发送会话密钥请求包。包括通话双方A、B的身份以及该次传输的唯一标识N1，称为临时交互号(nonce)。
- 临时交互号可以选择时间戳、随机数或者计数器等。KDC可根据临时交互号设计防重放机制。
- 2、KDC返回的信息包括两部分。
- 第一部分是A想获取的信息，用A的主密钥K_A加密，包括通话密钥K_s和KDC收到的请求包内容用以验证消息到达KDC前是否被修改或者重放过。
- 第二部分是B想获取的信息，用B的主密钥K_B加密，包括通话密钥K_s和A的身份。A收到后这部分消息便原样发给B。
- 3、为保证A发给B的会话密钥信息未被重放攻击，A、B使用会话密钥进行最后的验证。
- B使用新的会话密钥K_s加密临时交互号N_2并发给A。A对N_2进行一个函数变换后，用会话密钥发给B验证。

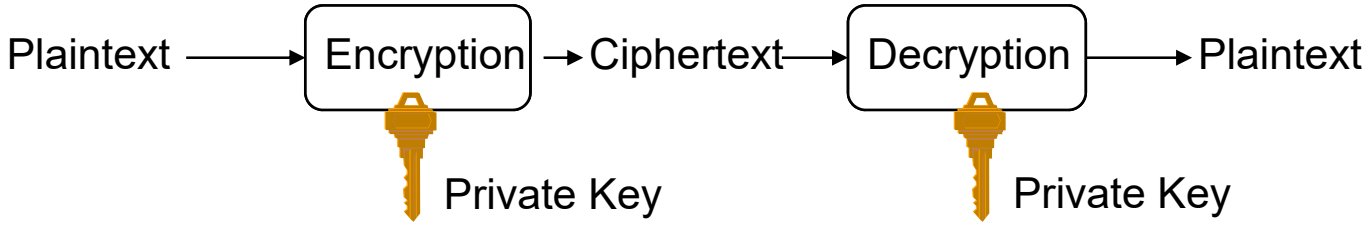
2. 密钥管理和分发

2.1 对称加密的对称密钥分发

缺点：

- KDC要可信
- KDC易成为攻击点
- 系统单点失效

2. 密钥管理和分发



对称密码模型

Pros：

- 计算效率高
- 模型简单
- 可提供机密性、完整性保障

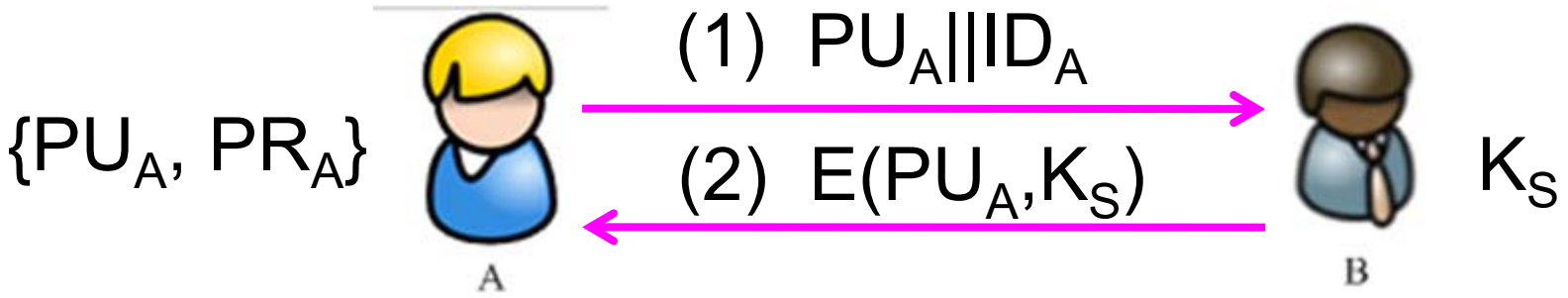
Cons：

- 数据交换前必须共享相同的密钥
- 扩展性差
- 存在密钥分发与管理难题
- 不提供可用性、不可否认性保障

2. 密钥管理和分发

2.2 公钥加密的对称密钥分发

MERK79方案：



(3) A计算$D(PR_A, E(PU_A,K_S))$，恢复K_S

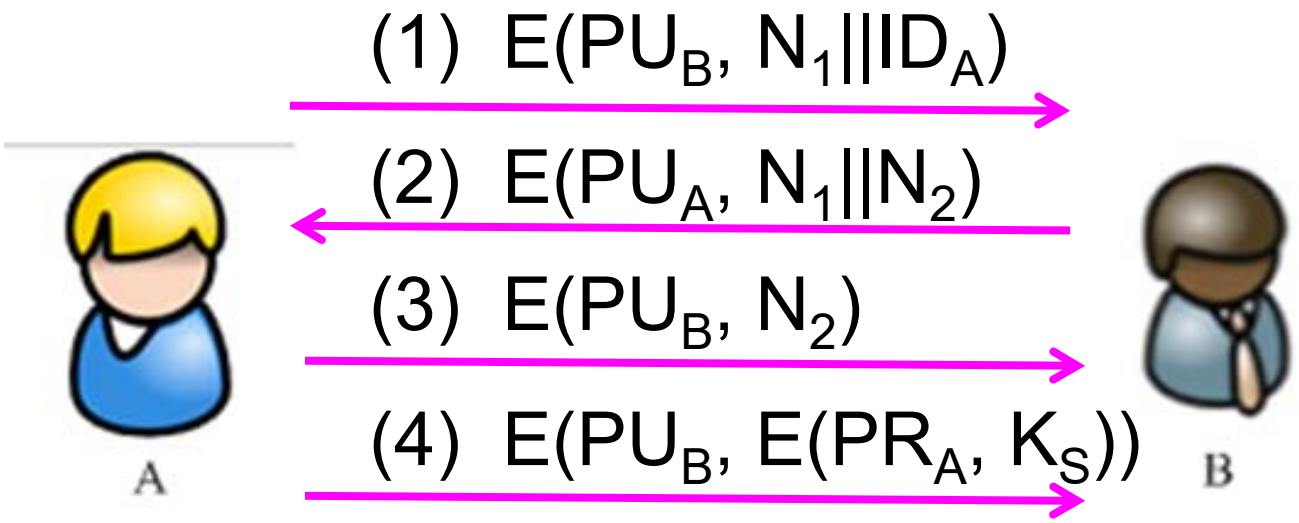
(4) A 丢弃$\{PU_A, PR_A\}$，B丢弃PU_A

缺陷：该协议存在中间人攻击的威胁

2. 密钥管理和分发

2.2 公钥加密的对称密钥分发

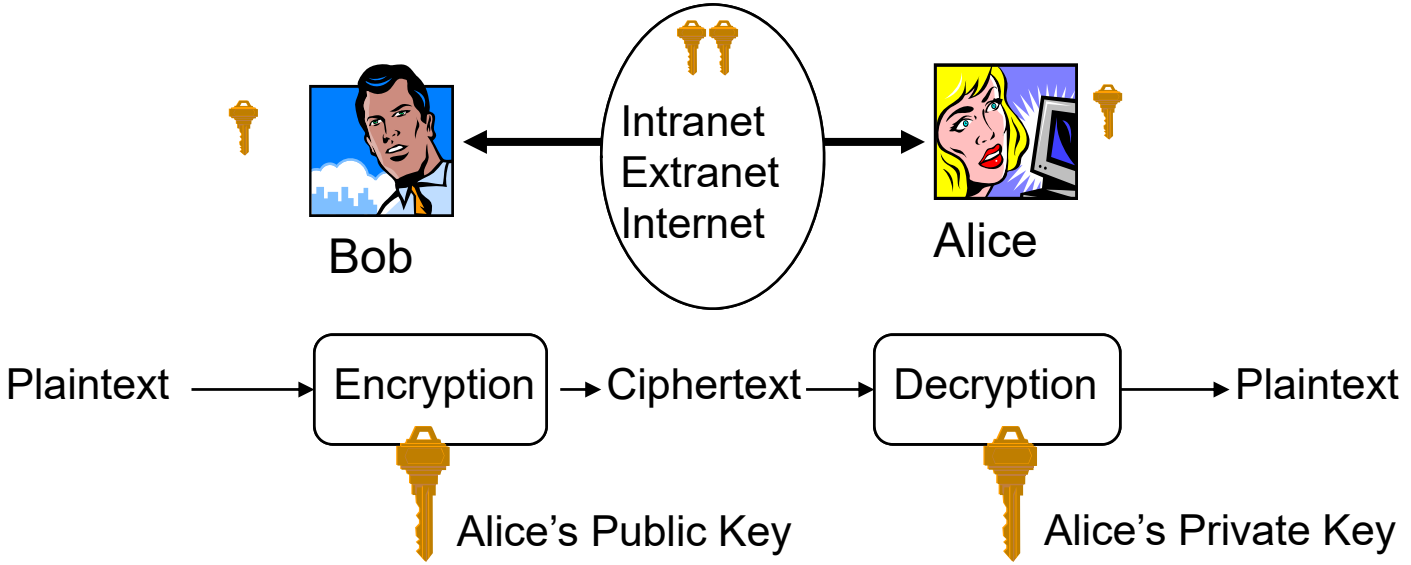
NEED78方案：



前提：A与B交换公钥

优点：机密性与身份认证

2. 密钥管理和分发

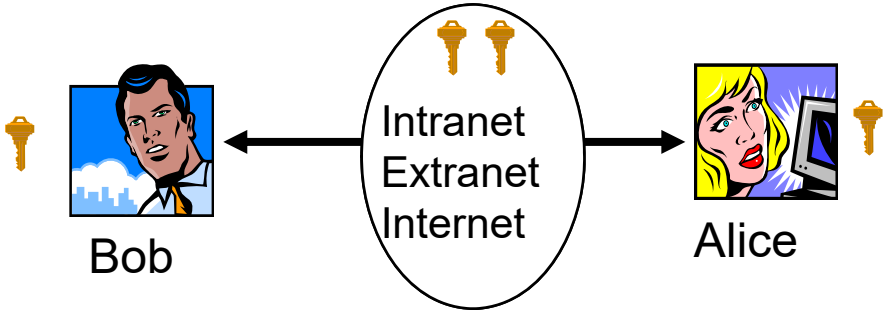


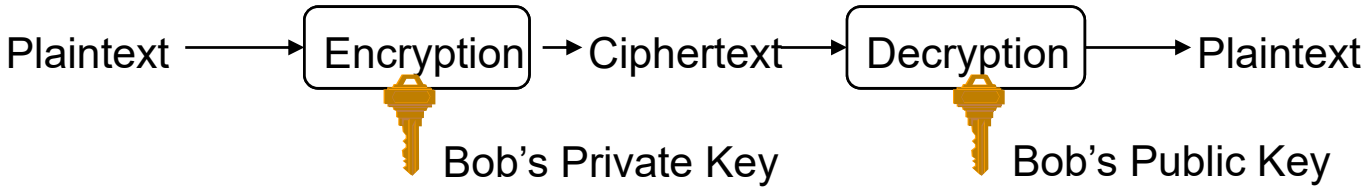
公钥密码模型 1

Pros：

- 私钥仅自己知道，降低了密钥泄露的风险
- 公钥公开，系统易扩展，易密钥分发与管理
- 可提供机密性保障

2. 密钥管理和分发





公钥密码模型 2

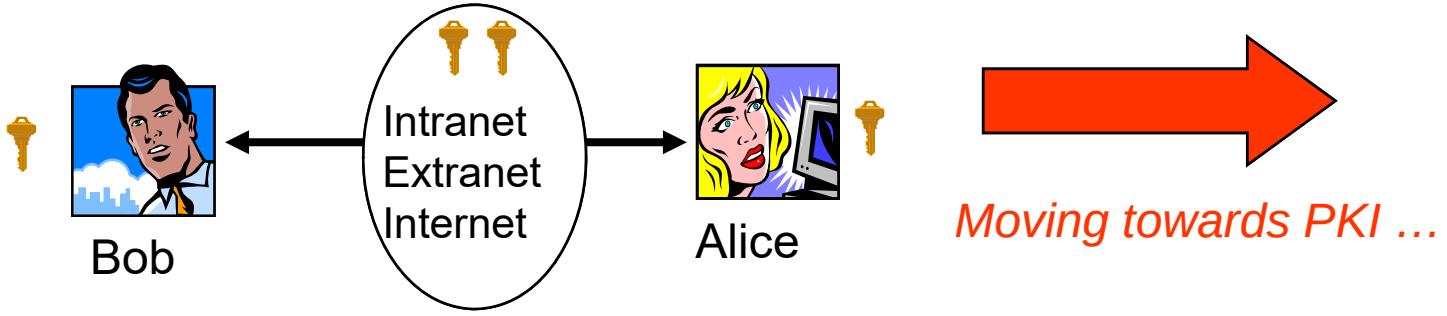
Pros：

- 私钥仅自己知道，降低了密钥泄露的风险
- 公钥公开，系统易扩展，易密钥分发与管理
- 可提供可认证性、不可否认性、完整性保障

2. 密钥管理和分发

Cons：

- 公钥密码算法运行速度比对称密码算法慢100 – 1000 倍。
- 公钥如何告之他人？
- 如何确保这一对密钥的主人就是Alice本人？



2. 密钥管理和分发

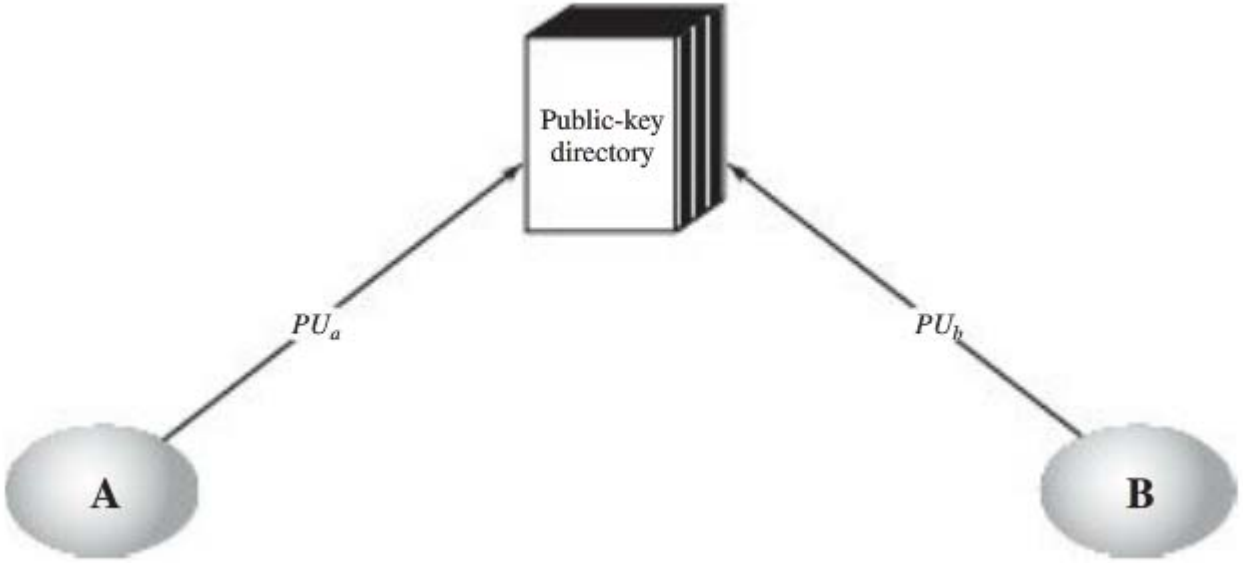
2.3 公钥发布

（1）公开发布

2. 密钥管理和分发

2.3 公钥发布

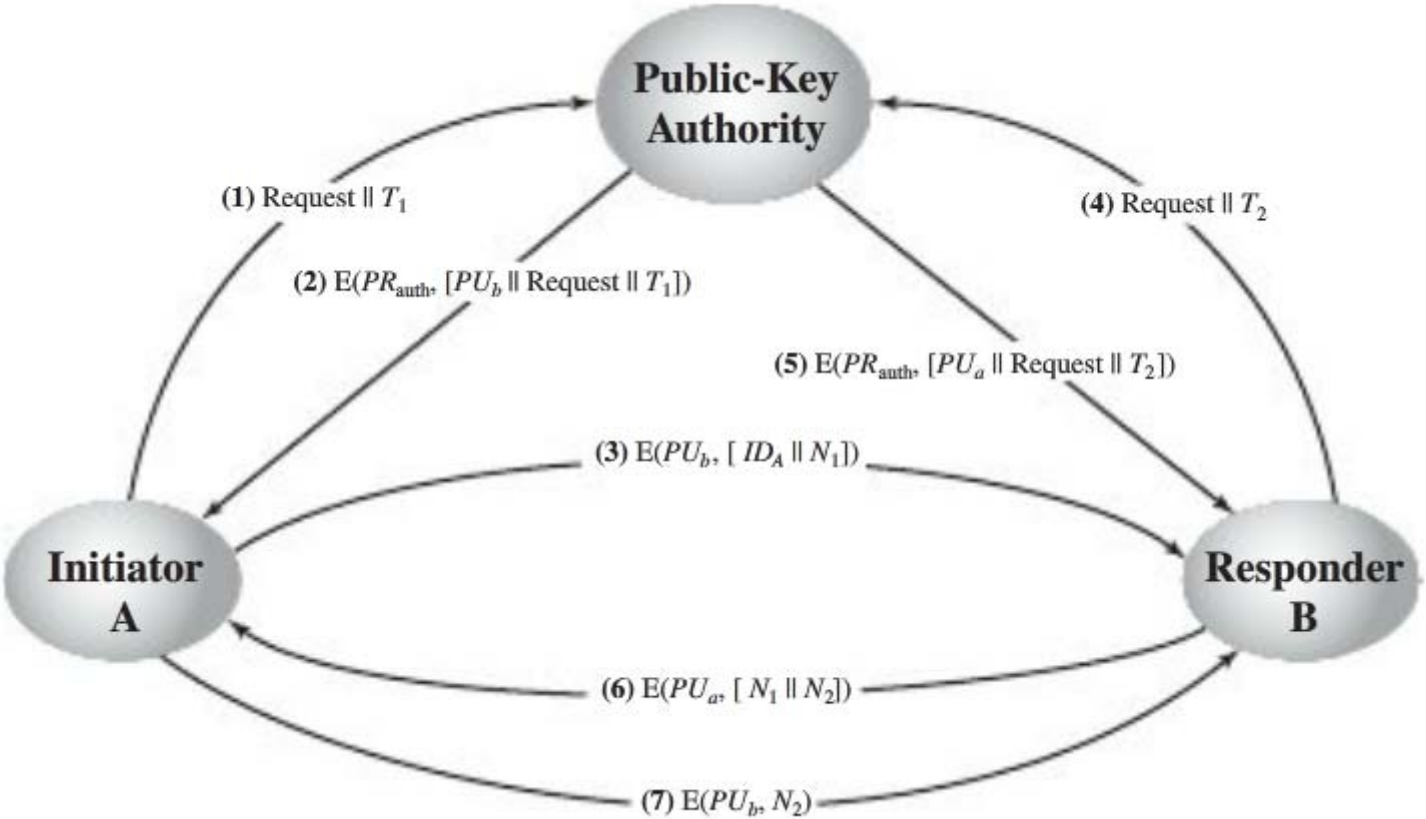
（2）公开可访问的目录



2. 密钥管理和分发

2.3 公钥发布

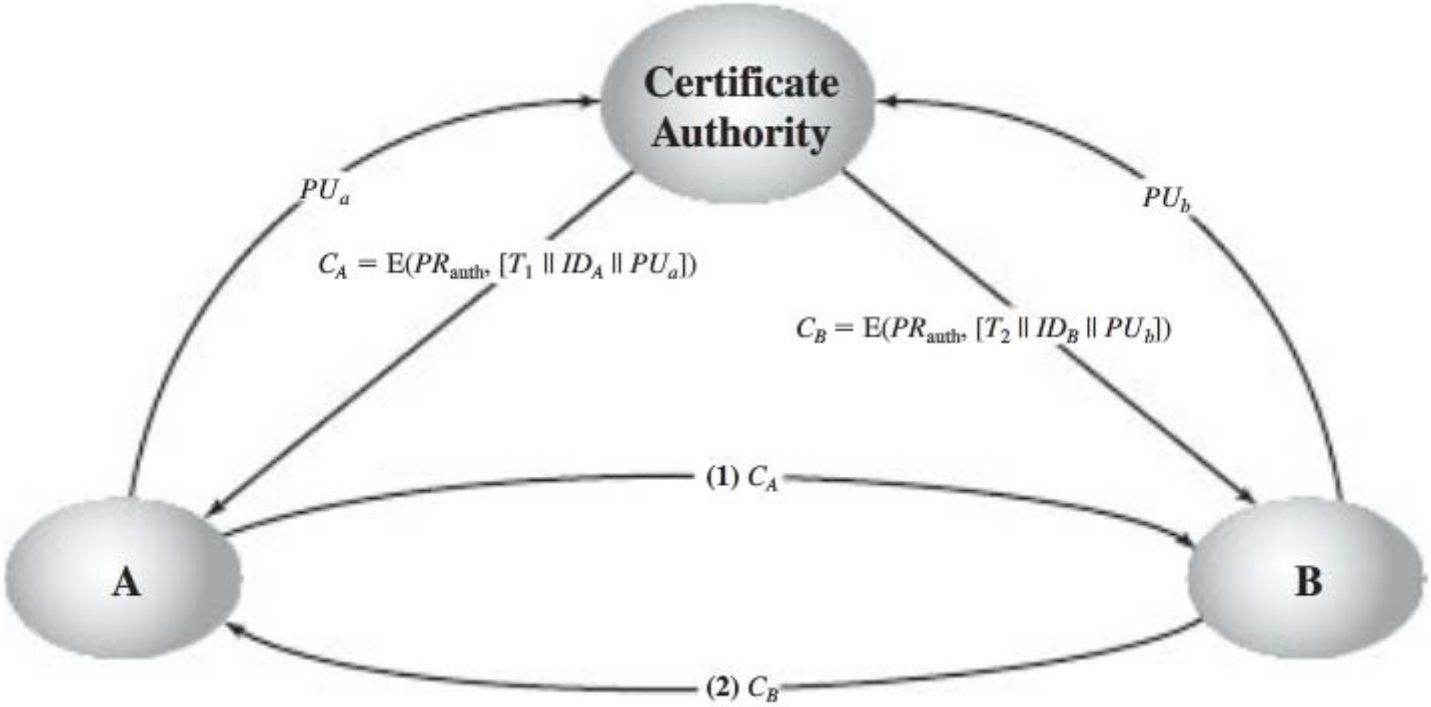
（3）公钥授权



2. 密钥管理和分发

2.3 公钥发布

（4）公钥证书



3. PKI概述

- PKI是“Public Key Infrastructure”的缩写，意为“公钥基础设施”，是一个用密码算法原理和技术实现的、具有通用性的安全基础设施。
- PKI是硬件、软件、人员、策略和操作规程的总和，它们要完成创建、管理、保存、发放和废止证书的功能。
- PKI是提供公钥加密和数字签名服务的系统，目的是为了自动管理密钥和证书，保证网上数字信息传输的机密性、完整性、可用性和不可否认性。

3. PKI概述

要使PKI得到广泛应用，必须解决标准化问题，这是建立互操作性的基础。目前，PKI主要有两个标准：

- RSA公司的公钥加密标准PKCS
- 国际电信联盟ITU 的数字证书标准X.509

3. PKI概述

PKCS#1：定义RSA公开密钥算法加密和签名机制，主要用于组织PKCS#7中所描述的数字签名和数字信封。

PKCS#3：定义Diffie-Hellman密钥交换协议。

PKCS#5：描述一种利用从口令派生出来的安全密钥加密字符串的方法。使用MD2或MD5 从口令中派生密钥，并采用DES-CBC模式加密。主要用于加密从一个计算机传送到另一个计算机的私人密钥，不能用于加密消息。

PKCS#6：描述了公钥证书的标准语法，主要描述X.509证书的扩展格式。

PKCS#7：定义一种通用的消息语法，包括数字签名和加密等用于增强的加密机制，PKCS#7与PEM兼容，所以不需其他密码操作，就可以将加密的消息转换成PEM消息。

3. PKI概述

PKCS#8：描述私有密钥信息格式，该信息包括公开密钥算法的私有密钥以及可选的属性集等。

PKCS#9：定义一些用于PKCS#6证书扩展、PKCS#7数字签名和PKCS#8私钥加密信息的属性类型。

PKCS#10：描述证书请求语法。

PKCS#11：称为Cyptoki，定义了一套独立于技术的程序设计接口，用于智能卡和PCMCIA卡之类的加密设备。

PKCS#12：描述个人信息交换语法标准。描述了将用户公钥、私钥、证书和其他相关信息打包的语法。

PKCS#13：椭圆曲线密码体制标准。

PKCS#14：伪随机数生成标准。

PKCS#15：密码令牌信息格式标准。

3. PKI概述

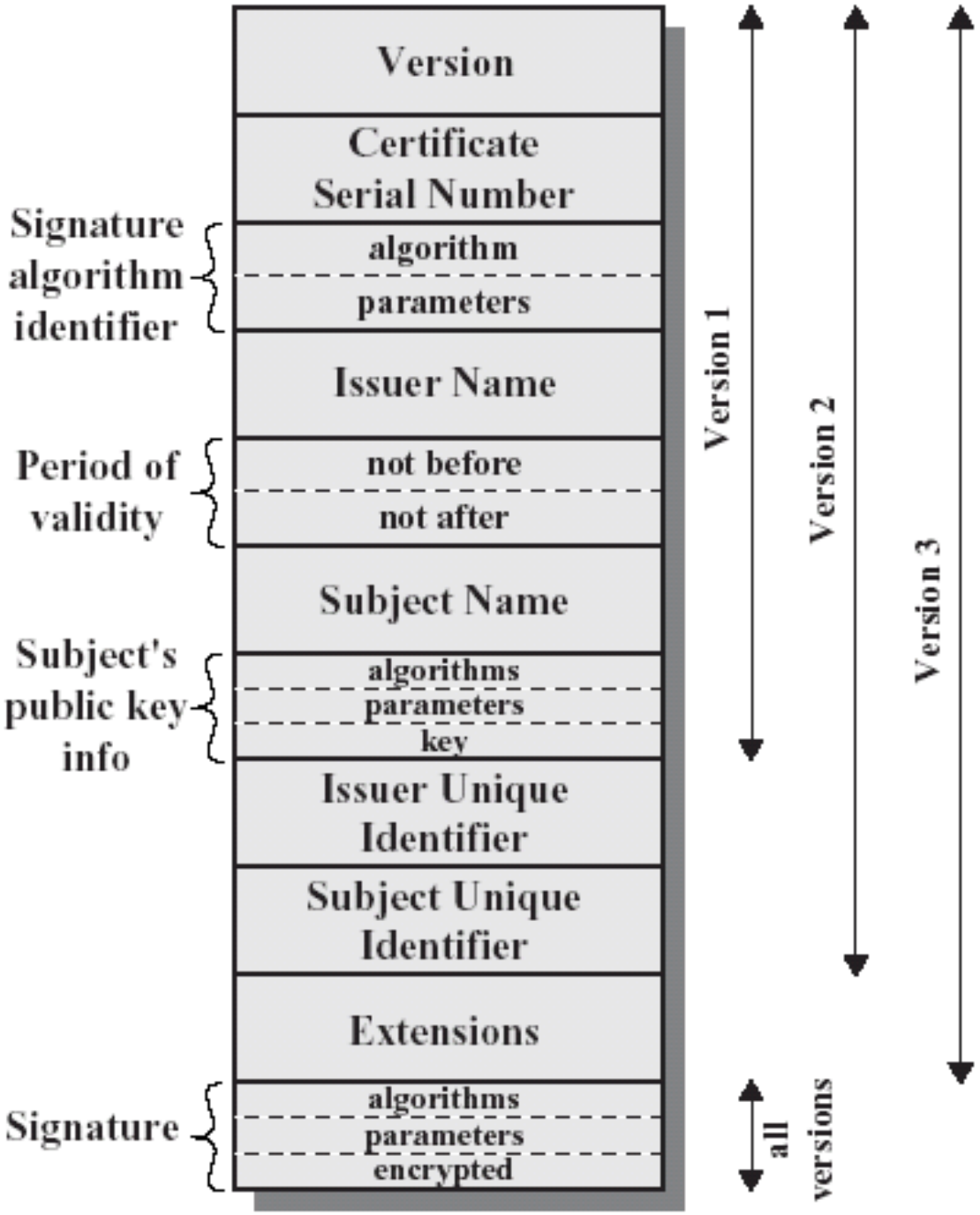
- X.509协议是国际标准组织CCITT建议使用的公钥密码技术的认证标准。 X.509协议是PKI技术体系中应用最广泛、也是最基础的一个国际标准。

- X.509协议的主要目的是定义一个规范的数字证书格式，而并非定义一个完整的、可互操作的PKI认证体系。

3. PKI概述

- X.509规定CA颁发的公钥证书中应包含如下信息：

Version
Certificate Serial Number
Signature Algorithm Identifier
Issuer Name
Validity (Not Before / Not After)
Subject Name
Subject Public Key Information
Issuer Unique Identifier
Subject Unique Identifier
Extensions
Certification Authority's Digital Signature


Version
Certificate Serial Number
Signature algorithm identifier
algorithm
parameters
Issuer Name
Period of validity
not before
not after
Subject Name
Subject's public key info
algorithms
parameters
key
Issuer Unique Identifier
Subject Unique Identifier
Extensions
Signature
algorithms
parameters
encrypted
Version 1
Version 2
Version 3
all versions

3. PKI概述

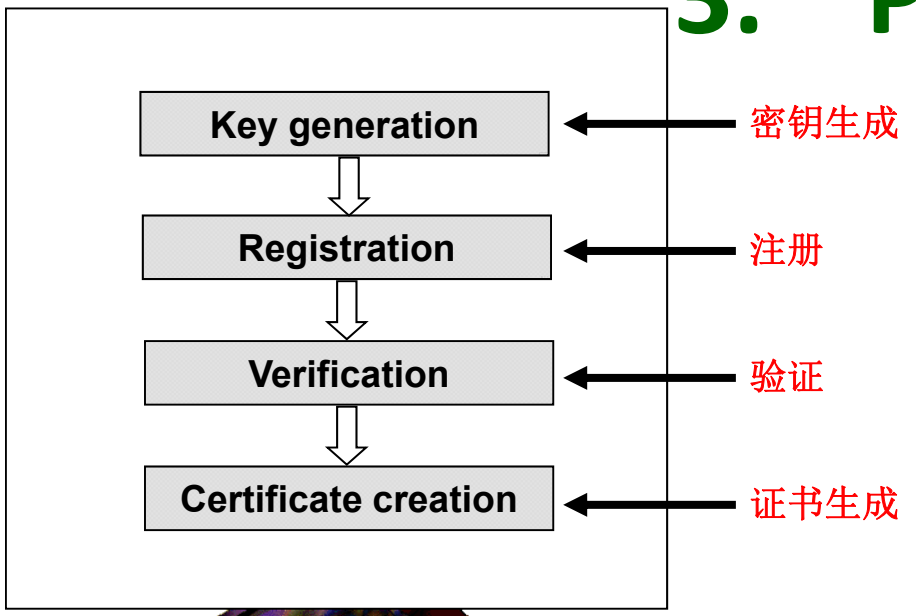
- PKI系统的组成：
  - 认证中心CA
  - 注册机构RA
  - 证书库CR
  - 证书申请者
  - 证书信任方

前三部分是PKI的核心，证书申请者和证书信任方则是利用PKI进行网上交易的参与者。

3. PKI概述

- CA是公钥基础设施中受信任的第三方实体
- CA向主体颁发证书
- CA是信任的起点

3. PKI概述



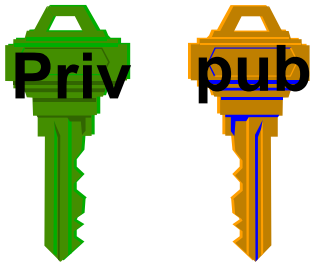
密钥生成

注册

验证

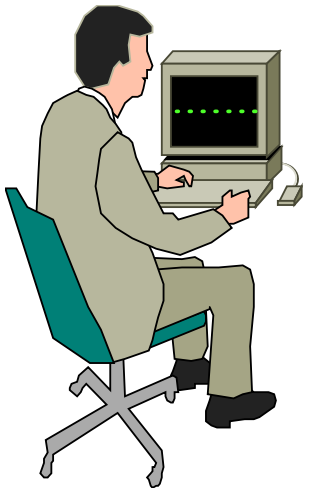
证书生成

生产密钥对



Certificate Authority

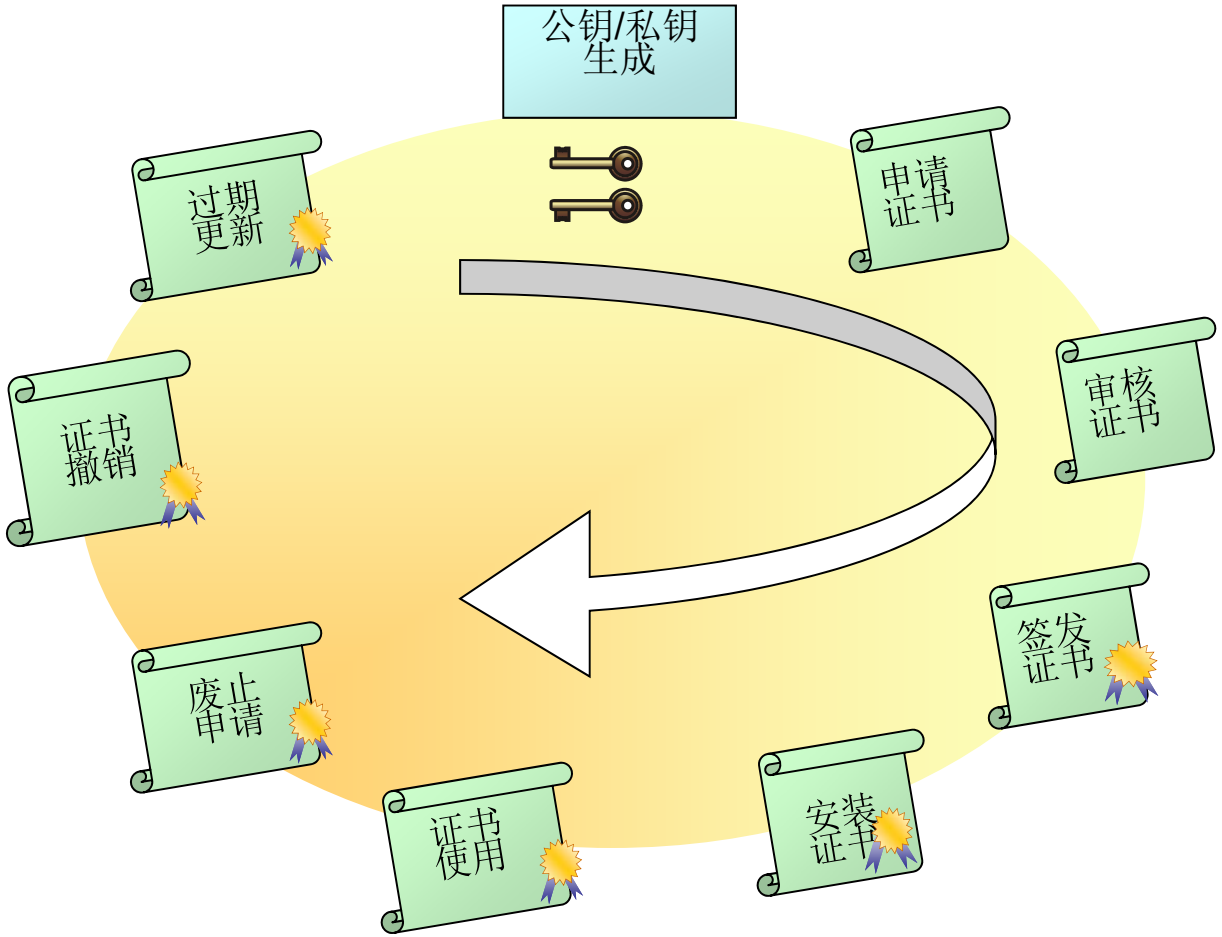
请求生成证书



核实申请者的身份

并生成数字证书

3. PKI概述



证书的生命周期

3. PKI概述

http://www.trustis.com/secure-email-encryption-demo.htm

4. 用户认证

认证一个用户的身份大致有四个常用工具：

- 知道什么：如口令、个人身份号或问题答案
- 拥有什么：如加密密钥、智能卡、U盾
- 静态生物特征：如指纹、虹膜、人脸
- 动态生物特征：声音、手写特征、打字节奏

4. 用户认证

4.1 基于对称加密的远程用户认证

NEED78认证协议：

1. A → KDC: $ID_A \| ID_B \| N_1$
2. KDC → A: $E(K_a, [K_s \| ID_B \| N_1 \| E(K_b, [K_s \| ID_A])])$
3. A → B: $E(K_b, [K_s \| ID_A])$
4. B → A: $E(K_s, N_2)$
5. A → B: $E(K_s, f(N_2))$

当A与B间有多个会话密码，存在重放攻击可能。

4. 用户认证

4.1 基于对称加密的远程用户认证

DENN82认证协议：

1. $A \rightarrow KDC: \quad ID_A \| ID_B$
2. $KDC \rightarrow A: \quad E(K_a, [K_s \| ID_B \| T \| E(K_b, [K_s \| ID_A \| T])])$
3. $A \rightarrow B: \quad E(K_b, [K_s \| ID_A \| T])$
4. $B \rightarrow A: \quad E(K_s, N_1)$
5. $A \rightarrow B: \quad E(K_s, f(N_1))$

要求A与B时钟同步，当发送者时钟快过接收者时钟，存在压制重放攻击可能。

4. 用户认证

4.1 基于对称加密的远程用户认证

KEHN92认证协议：

1. $A \rightarrow B$: $ID_A \| N_a$
2. $B \rightarrow KDC$: $ID_B \| N_b \| E(K_b, [ID_A \| N_a \| T_b])$
3. $KDC \rightarrow A$: $E(K_a, [ID_B \| N_a \| K_s \| T_b]) \| E(K_b, [ID_A \| K_s \| T_b]) \| N_b$
4. $A \rightarrow B$: $E(K_b, [ID_A \| K_s \| T_b]) \| E(K_s, N_b)$

4. 用户认证

4.2基于非对称加密的远程用户认证

DENN81认证协议：

1. $A \rightarrow AS: \quad ID_A \| ID_B$
2. $AS \rightarrow A: \quad E(PR_{as}, [ID_A \| PU_a \| T]) \| E(PR_{as}, [ID_B \| PU_b \| T])$
3. $A \rightarrow B: \quad E(PR_{as}, [ID_A \| PU_a \| T]) \| E(PR_{as}, [ID_B \| PU_b \| T]) \| E(PU_b, E(PR_a, [K_s \| T]))$

要求A与B时钟同步。

4. 用户认证

4.2 基于非对称加密的远程用户认证

WOO92a认证协议：

1. $A \rightarrow KDC: \quad ID_A \| ID_B$
2. $KDC \rightarrow A: \quad E(PR_{auth}, [ID_B \| PU_b])$
3. $A \rightarrow B: \quad E(PU_b, [N_a \| ID_A])$
4. $B \rightarrow KDC: \quad ID_A \| ID_B \| E(PU_{auth}, N_a)$
5. $KDC \rightarrow B: \quad E(PR_{auth}, [ID_A \| PU_a]) \| E(PU_b, E(PR_{auth}, [N_a \| K_s \| ID_B]))$
6. $B \rightarrow A: \quad E(PU_a, [E(PR_{auth}, [(N_a \| K_s \| ID_B)]) \| N_b])$
7. $A \rightarrow B: \quad E(K_s, N_b)$

4. 用户认证

4.2 基于非对称加密的远程用户认证

WOO92b认证协议：

1. $A \rightarrow KDC: \quad ID_A \| ID_B$
2. $KDC \rightarrow A: \quad E(PR_{auth}, [ID_B \| PU_b])$
3. $A \rightarrow B: \quad E(PU_b, [N_a \| ID_A])$
4. $B \rightarrow KDC: \quad ID_A \| ID_B \| E(PU_{auth}, N_a)$
5. $KDC \rightarrow B: \quad E(PR_{auth}, [ID_A \| PU_a]) \| E(PU_b, E(PR_{auth}, [N_a \| K_s \| ID_A \| ID_B]))$
6. $B \rightarrow A: \quad E(PU_a, [E(PR_{auth}, [(N_a \| K_s \| ID_A \| ID_B) \| N_b]))$
7. $A \rightarrow B: \quad E(K_s, N_b)$

习 题

15.8 Consider a one-way authentication technique based on asymmetric encryption:

$$\begin{aligned} \text{A} \rightarrow \text{B:} \quad & ID_A \\ \text{B} \rightarrow \text{A:} \quad & R_1 \\ \text{A} \rightarrow \text{B:} \quad & \text{E}(PR_a, R_1) \end{aligned}$$

a. Explain the protocol.
b. What type of attack is this protocol susceptible to?