密码学

第八章 认证

网络空间安全学院
朱丹 戚明平
zhudan/mpqi@nwpu.edu.cn

知识回顾-利用公钥密码实现数字签名

- **利用RSA密码实现数字签名：**
  - 设M为明文，$K_{e_A} =< e,n >$是A的公钥
  - $K_{d_A} =< p,q,\varphi(n),d >$是A的私钥
- 则A对M的签名过程是：
  - $S_A = D(M, K_{d_A}) = M^d \ mod \ n$
  - S_A便是A对M的签名
- 验证签名的过程是：
  - $E(S_A, K_{e_A}) = \left(M^d\right)^e mod \ n = M$

知识回顾-利用公钥密码实现数字签名

- **利用ElGamal密码实现数字签名：**
  - 密钥选择：选p是一个大素数，$p-1$有大素因子，a是模p的一个本原元，将p和a公开作为密码基础参数；用户随机地选择一个整数x（$1 \le x \le p-1$）作为私有的解密密钥；计算$y = a^x \ mod \ p$，取y作为公开的加密密钥。
  - 产生签名：设明文为M，$0 \le M \le p-1$，签名过程如下
    ① 用户A随机地选择一个整数$k, 1 < k < p-1$，且$(k, p-1) = 1$
    ② 计算$r = a^k \ mod \ p$
    ③ 计算$s = (M - xr)k^{-1} \ mod \ p-1$
    ④ 取(r, s)作为M的签名，并以$< M, r, s >$的形式发给用户B

知识回顾-利用公钥密码实现数字签名

- **利用ElGamal密码实现数字签名：**
  - 用户B接收$< M, r, s >$，用A的公钥验证

$$a^M = y^r r^s \ mod \ p$$

是否成立，若成立则签名为真，否则签名为假。

知识回顾-利用公钥密码实现数字签名

美国数字签名标准的签名算法DSA

密钥选择：

① 选取一个素数p，要求$2^{L-1} < p < 2^L$，其中$L = 512 + 64j, j = 0,1,2,\ldots,8$；

② 选取一个素数q，它是$p-1$的因子，$2^{159} < q < 2^{160}$；

③ 计算$g = h^{(p-1)/q} \bmod p$，h满足$1 < h < p-1$，且满足使$h^{(p-1)/q} \bmod p > 1$；

④ 随机选择一个数x作为用户的私钥，$0 < x < q$；

⑤ 计算$y = g^x \bmod p$作为用户的公钥。

参数p、 q和g可以公开。

知识回顾-利用公钥密码实现数字签名

美国数字签名标准的签名算法DSA

- 产生签名：设明文为M，签名过程如下

① 用户A随机地选择一个整数$k, 0 < k < q$；

② 计算$r = \left(g^k \ mod \ p\right) mod \ q$；

③ 计算$\mathrm{s} = k^{-1}(\mathrm{SHA}(M) + xr) \ mod \ q$；

④ 检验r和s是否为0，若其中一个为0，则重新选取k，重新计算r和s；

⑤ 取(r, s)作为M的签名，并以$< M, r, s >$的形式发给用户B

知识回顾-利用公钥密码实现数字签名

美国数字签名标准的签名算法DSA

用户B接收到$< M, r, s >$，根据$< p, q, g >$，验证过程如下

① 检验$0 < r < q$及$0 < s < q$是否成立，若其中之一不成立，签名为假；

② 计算

$$w = s^{-1} \ mod \ q$$

$$u_1 = \mathrm{SHA}(M) w \ mod \ q$$

$$u_2 = rw \ mod \ q$$

$$v = (g^{u_1} y^{u_2} \ mod \ p) \ mod \ q$$

若$v = r$，则签名为真；否则签名为假或数据被篡改。

知识回顾-利用公钥密码实现数字签名

利用椭圆曲线实现数字签名（ECDSA）

一个椭圆曲线密码由下面的六元组描述：

$$T = < p, a, b, G, n, h >$$

其中，p为大于3素数，p确定了有限域$GF(p)$；元素$a, b \in GF(p)$，a和b确定了椭圆曲线；G为循环子群E_1的生成元，n为素数且为生成元G的阶，G和n确定了循环子群E_1。

$$y^2 = x^3 + ax + b \ mod \ p$$

知识回顾-利用公钥密码实现数字签名

利用椭圆曲线实现数字签名（ECDSA）

- 密钥选择：
  - ① 随机选择一个数$d \in \{1,2,\dots,n-1\}$作为用户的私钥；
  - ② 计算$Q = dG$作为用户的公钥。
- 生成签名：设明文为M，签名过程如下
  - ① 随机选择一个数$k \in \{1,2,\dots,n-1\}$；
  - ② 计算$R(x_R, y_R) = kG$，并记$r = x_R \bmod n$；
  - ③ 计算$s = (\mathrm{HASH}(M) + rd)k^{-1} \bmod n$；
  - ④ 取(r,s)作为M的签名，并以$< M,r,s >$的形式传输或存储

知识回顾-利用公钥密码实现数字签名

利用椭圆曲线实现数字签名（ECDSA）

接收到消息$< M, r, s >$，验证过程如下

① 计算

$$e = \mathrm{HASH}(M)$$

$$u = es^{-1}\ mod\ n$$

$$v = rs^{-1}\ mod\ n$$

② 利用公钥Q计算$R(x_R, y_R) = uG + vQ$；

③ 如果$r = x_R\ mod\ n$，则签名为真；否则签名为假或数据被篡改。

知识回顾-利用公钥密码实现数字签名

利用SM2算法实现数字签名

密钥推荐使用256位素域$GF(p)$上的椭圆曲线 $y^2 = x^3 + ax + b$，参数如下：

$p = FFFFFFFE\ FFFFFFFF\ FFFFFFFF\ FFFFFFFF\ FFFFFFFF\ 00000000\ FFFFFFFF\ FFFFFFFF$

$a = FFFFFFFE\ FFFFFFFF\ FFFFFFFF\ FFFFFFFF\ FFFFFFFF\ 00000000\ FFFFFFFF\ FFFFFFFC$

$b = 28E9FA9E\ 9D9F5E34\ 4D5A9E4B\ CF6509A7\ F39789F5\ 15AB8F92\ DDBCBD41\ 4D940E93$

$n = FFFFFFFE\ FFFFFFFF\ FFFFFFFF\ FFFFFFFF\ 7203DF6B\ 21C6052B\ 53BBF409\ 39D54123$

$x_G = 32C4AE2C\ 1F198119\ 5F990446\ 6A39C994\ 8FE30BBF\ F2660BE1\ 715A4589\ 334C74C7$

$y_G = BC3736A2\ F4F6779C\ 59BDCEE3\ 6B692153\ D0A9877C\ C62A4740\ 02DF32E5\ 2139F0A0$

密钥选择：

① 私钥是随机数$d, d \in [1, n-2]$

② 公钥$P = dG = d(x_G, y_G) = (x_P, y_P)$

知识回顾-利用公钥密码实现数字签名

利用SM2算法实现数字签名

在利用SM2算法实现数字签名时，签名者和验证者都需要用密码学杂凑函数求得用户A的杂凑值

$$Z_A = H_{256}(ENTL_A \| ID_A \| a \| b \| x_G \| y_G \| x_P \| y_P)$$

其中ID_A是用户的标识，$ENTL_A$是由标识长度转换而成的两个字节；

$H_{256}()$选用的SM3；

SM2 密码算法使用规范（GM/T 0009-2012）规定，在无特殊约定的情况下，用户身份标识ID默认值从左至右依次为：

0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38。

知识回顾-利用公钥密码实现数字签名

利用SM2算法实现数字签名

- 生成签名：设明文为M，签名过程如下

① 置$\bar{M} = Z_A \,||M$;

② 计算$e = H_v(\bar{M})$并将e的数据表示为整数;

③ 用随机数发生器产生随机数$k \in [1,\ n-1]$；

④ 计算椭圆曲线点$kG = (x_1,\ y_1)$，并将x_1的数据表示为整数;

⑤ 计算$r = e + x_1 \ mod\ n$，若$r = 0$或$r + k = n$，则返回③;

⑥ 计算$s = (1+d)^{-1}(k - rd)\ mod\ n$，若$s = 0$则返回③;

⑦ 取(r, s)作为M的签名，并以$< M, r, s >$的形式传输或存储。

知识回顾-利用公钥密码实现数字签名

利用SM2算法实现数字签名

接收到消息$< M', r', s' >$，验证过程如下

① 检验$r' \in [1, n-1]$是否成立，若不成立则验证不通过；

② 检验$s' \in [1, n-1]$是否成立，若不成立则验证不通过；

③ 置$\overline{M'} = Z_A \,||M'$；

④ 计算$e' = H_v(\overline{M'})$并将e'的数据表示为整数；

⑤ 将r', s'的数据表示为整数，计算$t = (r' + s') \, mod \, n$，若$t = 0$，则验证不通过；

⑥ 计算椭圆曲线点$(x_1', y_1') = s'G + tP$；

⑦ 计算$R = e' + x_1' \, mod \, n$，检验$R = r'$是否成立，若成立，则验证通过；否则，验证不通过。

章节安排

Outline

认证的概念

身份认证

站点认证

消息认证

章节安排

Outline

认证的概念

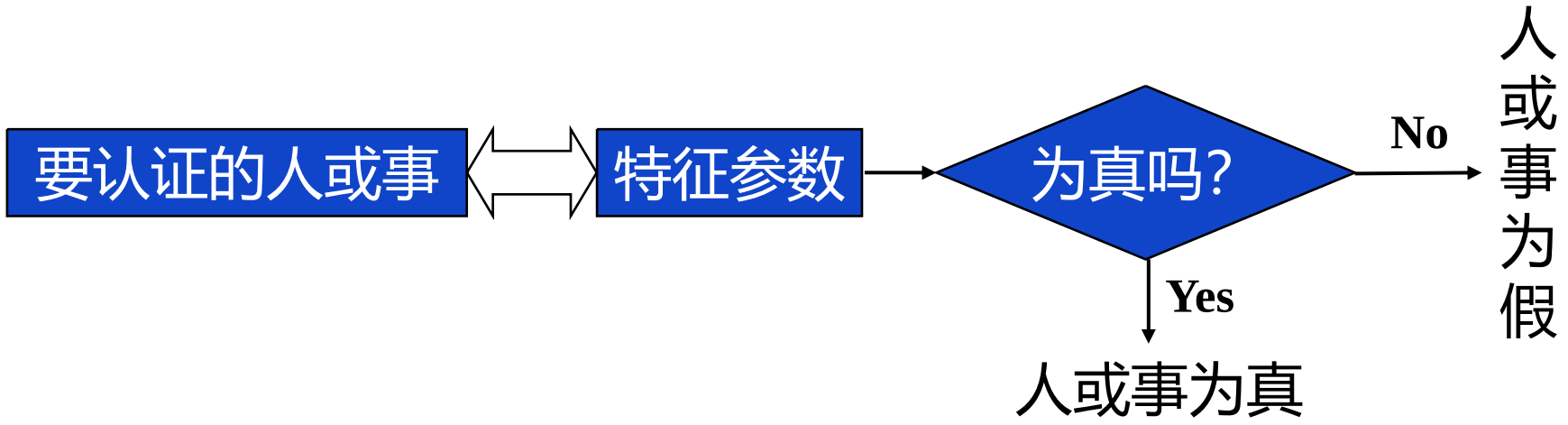
身份认证

站点认证

消息认证

8.1 认证的概念

- 认证（Authentication）又称鉴别、确认，它是证实某人某事是否名符其实或是否有效的一个过程
- 认证往往是许多应用系统中安全保护的第一道设防，因而极为重要



认证模型

8.1 认证的概念

- 认证参数有口令、标识符、密钥、信物、智能卡、USB-Key 、指纹、人脸、视网膜纹（虹膜）等
- 一般说来，利用人的生物特征参数进行认证的安全性高，但技术要求也高。指纹识别和人脸识别应用逐渐推广
- **认证和加密的区别**
  - 加密用以确保数据的机密性
  - 认证用以确保当事人身份的真实性以及数据的完整性

认证和数字签名的区别

- 认证总是基于某种收发双方共享的保密数据来认证对象的真实性，而数字签名中用于验证签名的数据是公开的
- 认证允许收发双方互相验证其真实性，不准许第三方验证，数字签名允许收发双方和第三方都能验证
- 数字签名具有发送方不能抵赖、接收方不能伪造和能够公开验证解决纠纷的能力，认证则不一定具备

章节安排

Outline

认证的概念

身份认证

站点认证

消息认证

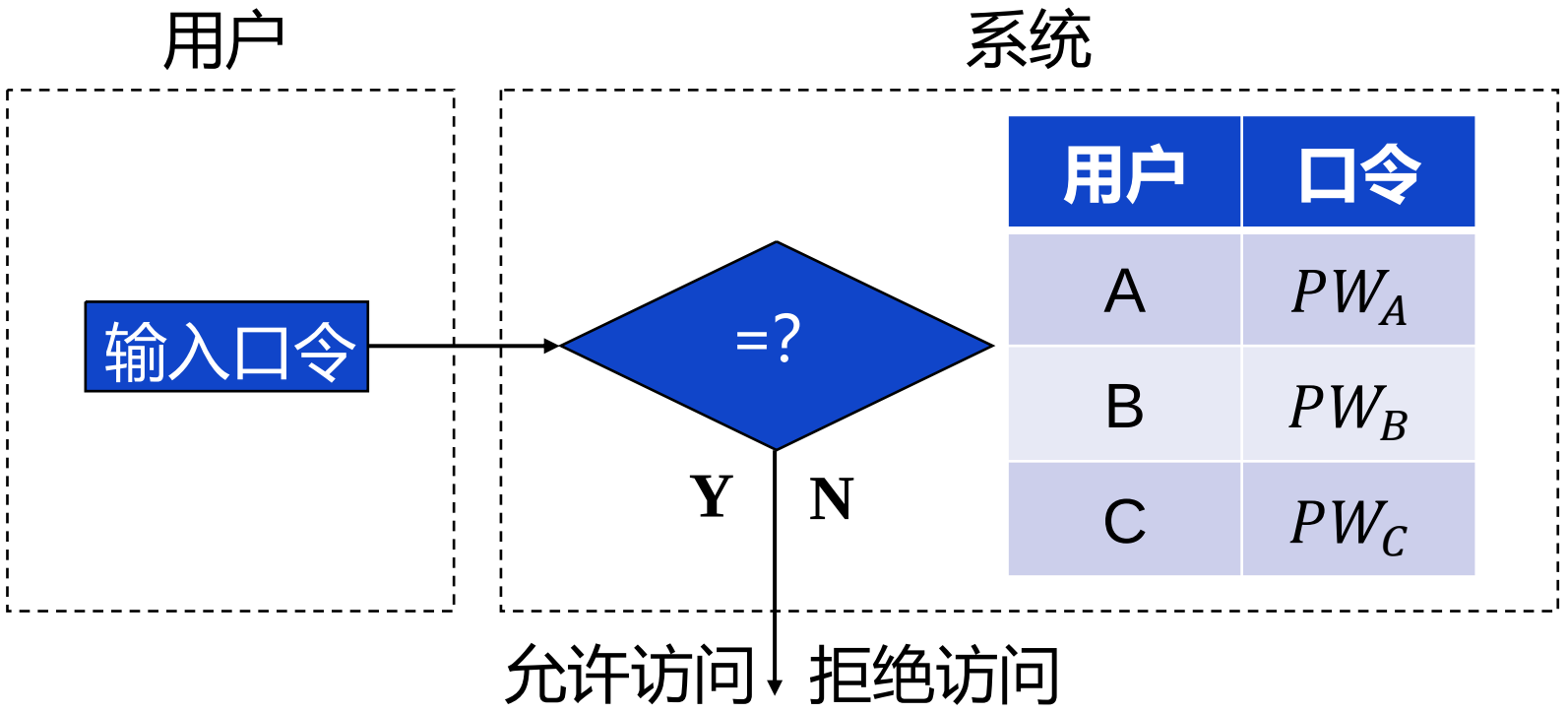
8.2 身份认证

- 用户的身份认证是许多应用系统的第一道防线，其目的在于识别用户的合法性，从而阻止非法用户访问系统
- 身份认证对确保系统的信息安全是极其重要的
- 一般，可以通过以下验证方式，来认证用户的身份
  - 用户知道什么（所知）
  - 用户拥有什么（所有）
  - 用户的生物特征（所是）

身份认证—口令

- 口令是双方预先约定的秘密数据，口令认证属于验证用户知道什么
- 口令验证的安全性虽然不如其他几种方法，但是口令验证简单易行，因此口令验证仍是目前应用最为广泛的身份认证方法
- 在一些简单的系统中，用户的口令以口令表的形式存储。当用户要访问系统时，系统要求用户提供口令，系统将用户提供的口令与口令表中存储的口令进行比较，若相等则确认用户身份有效，否则确认用户身份无效，拒绝访问。

身份认证—口令



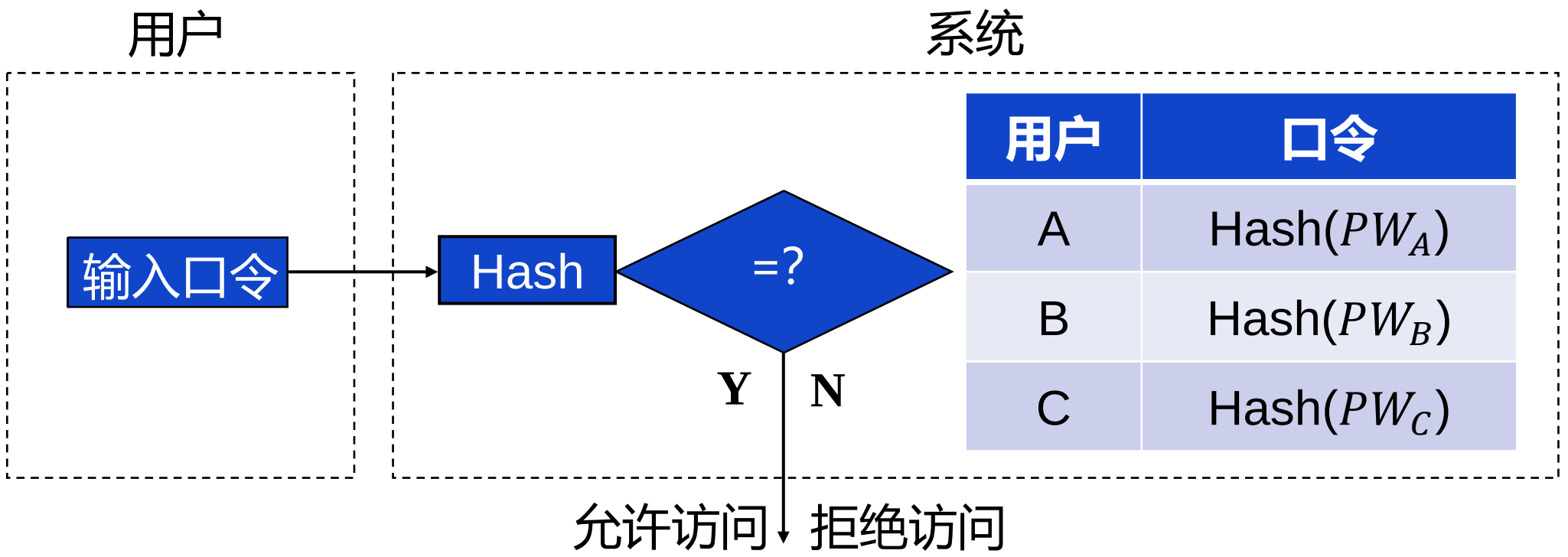
身份认证—口令

- 这样的简单口令系统存在以下问题
  - 因为用户的口令以明文形式存储在系统中，系统管理员可以获得所有口令，攻击者也可利用系统的漏洞来获得他人的口令
  - 因为用户的口令在用户终端到系统的线路上以明文形式传输，所以攻击者可在传输线路上截获用户口令
  - 只有系统验证用户的身份，用户不能验证系统的身份

身份认证—口令

- 利用单向函数加密口令
  - 用户的口令在系统中以密文的形式存储
  - 口令一旦加密，将不可解密
  - 用户访问系统时提供其口令，系统对该口令用单向函数加密，并与存储的密文相比较。若相等，则确认用户身份有效，否则确认用户身份无效
  - 可选用强的HASH函数作为单向函数

身份认证—口令



用户	口令
A	Hash(PW_A)
B	Hash(PW_B)
C	Hash(PW_C)

身份认证—口令

- 利用数字签名的方法验证口令

① 用户i将其公钥提交给系统，作为验证口令的数据，系统为每个用户建立一个已访问次数标志T_i

② 用户访问系统时将其签名信息提供给系统，

$$ID_i||D\big((ID_i,N_i),K_{d_i}\big)$$

其中N_i表示本次访问是第N_i次访问。

③ 系统根据明文形式的标识符ID_i查出K_{e_i}，并计算

$$E\big(D\big((ID_i,N_i),K_{d_i}\big),K_{e_i}\big)=<ID_i^*,N_i^*>$$

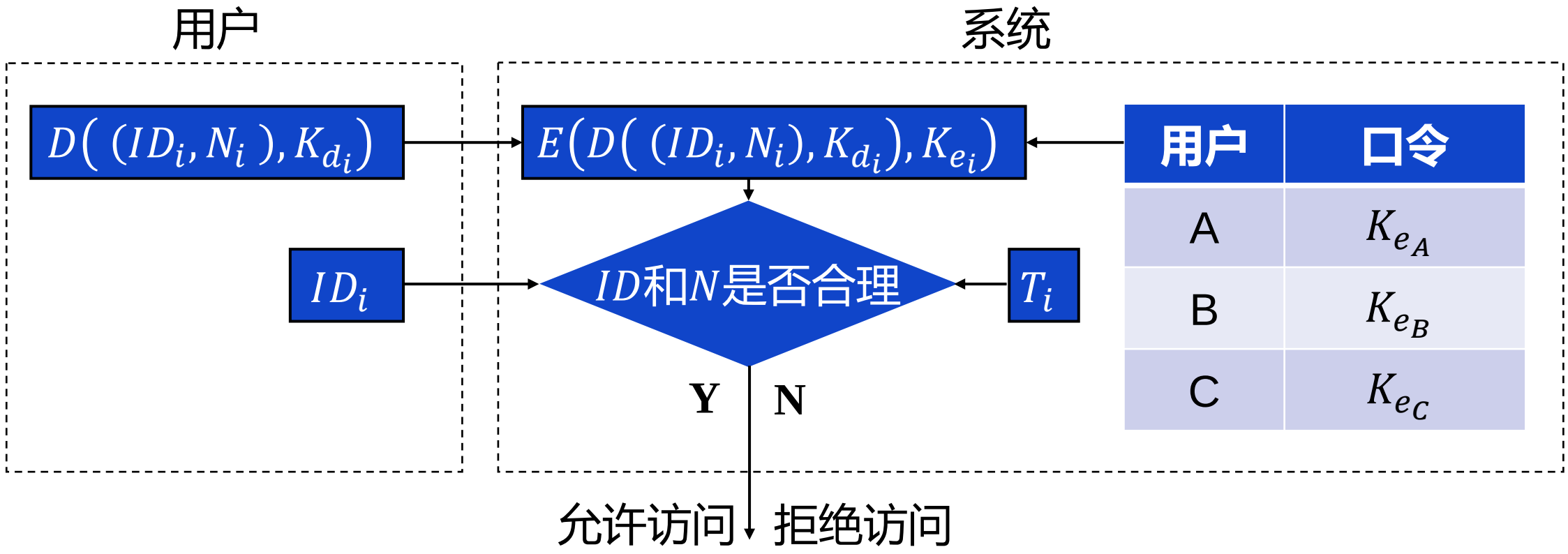
身份认证—口令

- 利用数字签名的方法验证口令

④ 当且仅当$ID_i = ID_i^*, N_i^* = T_i + 1$时系统才确认用户身份有效

⑤ 安全性分析：口令是用户的保密的解密密钥K_{d_i}，它不存储于系统中，所以任何人都不可能得到；虽然K_{e_i}存储于系统中，但是由K_{e_i}不能推出K_{d_i}；由于从终端到系统的通道上传输的是签名数据而不是K_{d_i}本身，所以攻击者也不能通过截取获得K_{d_i}；由于系统为每用户设置了已访问次数标志T_i，且仅当$N_i^* = T_i + 1$时才接收访问，所以可以抗重放攻击。但必须对T_i实施保护。

身份认证—口令


用户
系统
$D((ID_i, N_i), K_{d_i})$
$E(D((ID_i, N_i), K_{d_i}), K_{e_i})$
ID_i
ID和N是否合理
T_i
Y
N
允许访问
拒绝访问
用户
口令
A
K_{e_A}
B
K_{e_B}
C
K_{e_C}

身份认证—口令

- 利用零知识证明的方法验证口令

P要向V证明自己拥有某个房间的钥匙，假设该房间只能用钥匙打开锁，而其他任何方法都打不开。这时有2个方法：

① P把钥匙出示给V，V用这把钥匙打开该房间的锁，从而证明P拥有该房间的正确的钥匙；

② V确定该房间内有某一物体，P用自己拥有的钥匙打开该房间的门，然后把物体拿出来出示给V，从而证明自己确实拥有该房间的钥匙。

方法②属于零知识证明。它的好处在于，在整个证明的过程中，B始终不能看到钥匙的样子，从而避免了钥匙的泄露。

身份认证—口令

- 利用零知识证明的方法验证口令

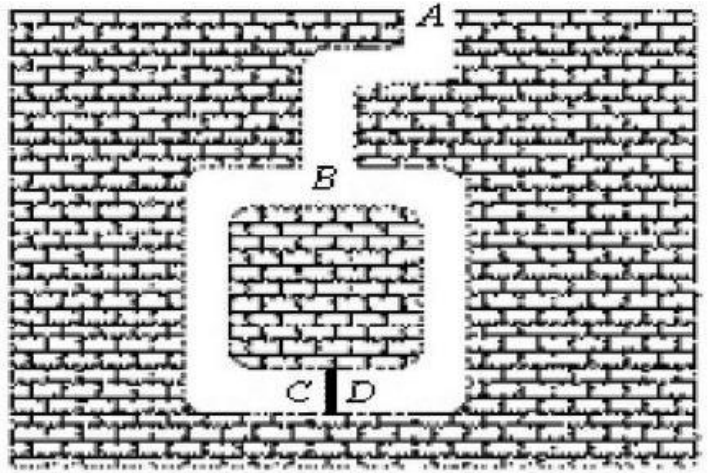
C和D之间存在一道密门，并且只有知道咒语的人才能打开。P知道咒语并想向V证明，但证明过程中不想泄露咒语。他该怎么办呢?

① 首先，V走到A，P走到C或者D点;

② 然后，V走到B，让P从洞穴的一边或者另一边出来;

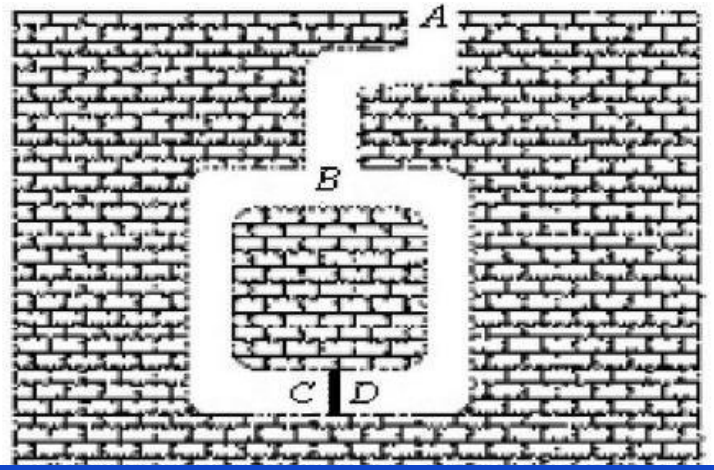
③ 如果P知道咒语，就能正确地从V要求的那一边出来;

④ V重复上述过程很多次，直到他相信P确实知道打开密门的咒语为止。



身份认证—口令

- 利用零知识证明的方法验证口令
  - 在这里，P是证明方，V是验证方。P通过上述方法证明了自己确实知道咒语，但是没有像V透露任何咒语的相关信息，这一过程也就是零知识证明；
  - 经过n轮上述过程，如果P不知道咒语，那么P成功欺骗V的概率为$\frac{1}{2^n}$（当n=16时，P成功的概率仅有1/65536）



身份认证—口令

- 利用零知识证明的方法验证口令
  - 基于离散对数问题的零知识证明
  - P向V证明他知道$y = g^x$对应的私钥x，其中$x \in F_p^*$但是不能泄露私钥x
    ① P随机选择$r \in F_p^*$, 计算$h = g^r \ mod \ p$, 将h发送给V
    ② V发送一个随机比特位$b \in \{0, \ 1\}$给P
    ③ P计算$s = r + bx \ mod \ p - 1$，并发送s给V
    ④ V验证$g^s ? = hy^b \ mod \ p$
    ⑤ 重复(1)至(4) t 次

身份认证—口令

- 利用零知识证明的方法验证口令
  - 基于离散对数问题的零知识证明
  - 如果P知道私钥x，则算法的验证一定成功，这时P没有欺骗V；
  - 如果P不知道私钥x，则每次P欺骗V成功的概率为1/2，连续t次P欺诈成功的概率为$\frac{1}{2^t}$。所以，当t足够大时V能够识别出P是否在欺骗他，能够满足实际应用需要。

身份认证—口令

- 利用零知识证明的方法验证口令
  - Schnorr 身份认证方案：选取两个素数p和q且$q|p-1$，选取$g \in F_p^*$且$g^q \equiv 1 \, mod \, p$，$g \neq 1$。P生成自己的公私钥对：$(y = g^{-x}, x \in [1, \, q-1])$。P向V证明他拥有私钥$x$但不能泄露私钥$x$的过程：
    ① P随机选择$r \in [1, q-1]$, 计算$h = g^r \, mod \, p$, 将h发送给V
    ② V发送一个随机选取的$e \in \{1, 2^t\}$给P，其中$2^t < q$
    ③ P计算$s = r + ex \, mod \, q$，并发送s给V
    ④ V验证$h ? = g^s y^e \, mod \, p$
  - 如果P知道私钥x，则算法的验证一定成功，这时P没有欺骗V；
  - 如果P不知道私钥x，则每次P欺骗V成功的概率为$\frac{1}{2^t}$。所以，当t足够大时V能够识别出P是否在欺骗他，能够满足实际应用需要。

身份认证—口令

- 口令的双向验证
  - 仅有一方能验证另一方的身份，而另一方不能验证对方的身份，是不全面的，也是不平等的
- 设A的口令为P_A，B的口令为P_B
  - 当A要求与B通信时，B必须验证A的身份，因此A应当首先向B出示表示自己身份的数据
  - 但此时A尚未对B的身份进行验证，所以A不能直接将自己的口令发给B
  - 如果B要求与A通信也存在同样的问题

身份认证—口令

口令的双向验证

- 设f是单向函数，R_A是A的随机数，R_B是B的随机数。P_A是A的口令，P_B是B的口令，A和B共享口令

① A→B： R_A

② B→A： $f(P_B \| R_A) \| R_B$，A利用f对自己的R_A和共享的P_B进行加密，并与接收到的$f(P_B \| R_A)$进行比较。若两者相等，则A确认B的身份是真实的

③ A→B： $f(P_A \| R_B)$，B利用f对自己的R_B和共享的P_A进行加密，并与接收到的$f(P_A \| R_B)$进行比较。若两者相等，则B确认A的身份是真实的

身份认证—口令

- 口令的双向验证
  - 由于f是单向函数，攻击者即使知道$f(P_A \| R_A)$和R_A也不能计算出P_A；同理，即使知道$f(P_B \| R_B)$和R_B也不能计算出P_B
  - 在上述口令验证中，即使有一方是假冒者，他也不能骗得对方的口令
  - 为了阻止重播攻击，可在$f(P_B \| R_A)$和$f(P_A \| R_B)$中加入时间性参量（时间戳）

只有共享口令的双方能正常完成认证

身份认证—口令

- 一次性口令：为了安全，口令应当经常更换，最好是一个口令只使用一次
- 利用单向函数可实现一次性口令
  - 设A和B要进行通信，A选择随机数x，并计算$y_0 = f^n(x)$
  - A将y_0发送给B作为验证口令的数据。因为f是单向函数，所以无需对y_0保密
  - A以$y_i = f^{n-i}(x)\ (1 \leq i \leq n-1)$作为其第$i$次通信的口令发送给B
  - B计算并验证：$f(y_i) ?= y_{i-1}$。若相等，则确认A的身份是真实的，否则可知A的身份是不真实的

身份认证—口令

- 使用口令认证注意事项：
  - 应使用多种字符。在口令中同时使用字母、数字、特殊字符等；
  - 足够的长度。口令一般应选择6~10个字符为宜；
  - 应尽量随机。避免使用相关的人名、生日、电话号码等；
  - 定期更换。

身份认证—磁卡、智能卡

- 磁卡是目前已广泛应用的一种个人身份持证物，在银行界得到广泛地应用。磁卡使用方便、成本低。但磁卡仅有有限的数据存储能力，无数据处理能力，安全性低。
- 智能卡是一种镶嵌有单片机芯片的集成电路卡
- 卡上有CPU、RAM、EEPROM或FLASH、ROM和I/O接口。因此智能卡被誉为最小的个人计算机。芯片操作系统COS（Chip Operating System）管理资源。安全性高
- 可集成TPM

身份认证—USB-Key

- USB-Key是一种具有USB接口，具有加解密、数字签名等多种安全保密功能的便携式安全设备。从技术上看，USB-Key就是一个具有USB接口智能卡

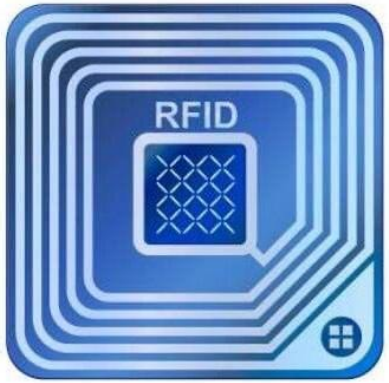
如果仅仅只靠磁卡、智能卡和USB-key这种物理持物来作为用户的身份凭证进行身份认证，尚有不足。因为**它们会丢失，则捡到的人就可假冒真正的用户**。为此，还需要一种磁卡、智能卡和USB-key上不具有的身份信息。这种身份信息通常采用个人识别号PIN(Personal Identification Number)

身份认证—生物特征

- 人的指纹、掌纹、面孔、发音、视网膜、DNA等都具有唯一性和稳定性的特征，即每个人的这些特征都与别人不同且终生不变，因此可以据此进行身份识别
- 基于这些特征，人们发展了指纹识别、视网膜识别、语音识别、人脸识别等多种生物识别技术，其中指纹识别和人脸识别技术比较成熟，得到广泛应用

身份认证—RFID技术

- 射频识别（RFID, Radio Frequency Identification）技术是一种利用射频信号和电感耦合实现目标识别的技术，利用电感耦合实现基站和标签之间的数据传输
- 射频标签是非接触的、读写速度快、功耗低、标签可读写、可存储大量数据
- 结合密码技术可实现高安全性的应用



章节安排

Outline

认证的概念

身份认证

站点认证

消息认证

8.3 站点认证

- 为了确保通信安全，在正式传送消息之前，应首先认证通信是否在意定的站点之间进行，这一过程称为站点认证；站点认证是通过验证加密的数据能否正确地在两个站点间进行传送来实现的
- 设A、B是意定的两个站点，A是发送方，B是接收方。利用传统密码体制，则A和B相互认证的过程如下（假定A、B共享保密的会话密钥K_S）

1. A产生随机数R_A		1. B产生随机数R_B
2. A→B：$E(R_A, K_S)$		2. B接收$E(R_A, K_S)$
3. A接收$E(R_A\|\|R_B, K_S)$		3. B→A：$E(R_A\|\|R_B, K_S)$
并解密判断 $R_A = R_A$? 若相等则A认为B是合法站点		
4. A→B：$E(R_B, K_S)$		4. B接收$E(R_B, K_S)$
		5. B判断 $R_B = R_B$?
		若相等则B认为A是合法站点

- 安全性：上述协议成功，则表明A和B都拥有K_S， K_S是保密的，因此A和B是合法的

8.3 站点认证

- 利用公钥密码，则A和B相互认证的过程如下:

1. A产生随机数R_A		1. B产生随机数R_B
2. A→B：R_A		2. B接收R_A
3. A接收$D(R_A\|\|R_B, Kd_B)$		3. B→A：$D(R_A\|\|R_B, Kd_B)$
并验证B的签名，如正确则A认为B是合法站点。		
4. A→B：$D(R_B, Kd_A)$		4. B接收$D(R_B, Kd_A)$
	并验证A的签名，如正确则B认为A是合法站点	

- 安全性：上述认证本质上是验证数字签名，数字签名具有确保真实性的能力

章节安排

Outline

认证的概念

身份认证

站点认证

消息认证

消息认证

- 消息认证必须使通信方能够验证每份消息的发送方、接收方、内容和时间性的真实性和完整性。能够确定：
  - 消息是由意定的发送方发的
  - 消息传送给意定的接收方
  - 消息内容有无篡改或发生错误
  - 消息按确定的次序接收

消息认证

- 信源的认证（采用传统密码）
- 设A为发送方，B为接收方。A和B共享保密的密钥K_S。A的标识为ID_A，消息为M，在消息中增加标识ID_A，那么B认证A的过程如下：

$$A \rightarrow B: < ID_A, E(ID_A||M, K_S) >$$

- B收到消息后用K_S解密，若解密所得的发送方标识与ID_A相同，则B认为消息是A发来的

消息认证

- 信源的认证（采用公钥密码）
- 只要发送方对每一消息进行数字签名，接收方验证签名即可：

$$A \rightarrow B: < D(ID_A||M, K_{d_A}) >$$

$$B: E(D(ID_A||M, K_{d_A}), K_{e_A})$$

若收方验证签名正确，则认为发方为真

- 此方案不能保密，若还需要保密，则应先签名后加密

消息认证

- 信宿的认证：只要将消息源的认证方法稍加修改便可实现消息宿的认证
- 采用传统密码：在每份消息中加入接收方标识符ID_B，并加密：

$$A \rightarrow B: < E(ID_B||M, K_S) >$$

- 若采用公开密钥密码：对每份消息加入接收方标识符ID_B，并用B的公钥进行加密：

$$A \rightarrow B: E(ID_B||M, K_{e_B})$$

此方案不能保真，因为K_{e_B}是公开的，任何人都可以冒充A，发送$E(ID_B||M, K_{e_B})$

消息认证

- 消息内容的认证
- 消息内容认证使接收方能够确认消息内容的真实性和完整性，这可以通过验证消息认证码 的正确性来实现
- 消息认证码MAC（Message Authentication Code）是消息内容和密钥的公开函数，其输出是长度固定的短数据块：

$$MAC = C(M, K)$$

消息认证

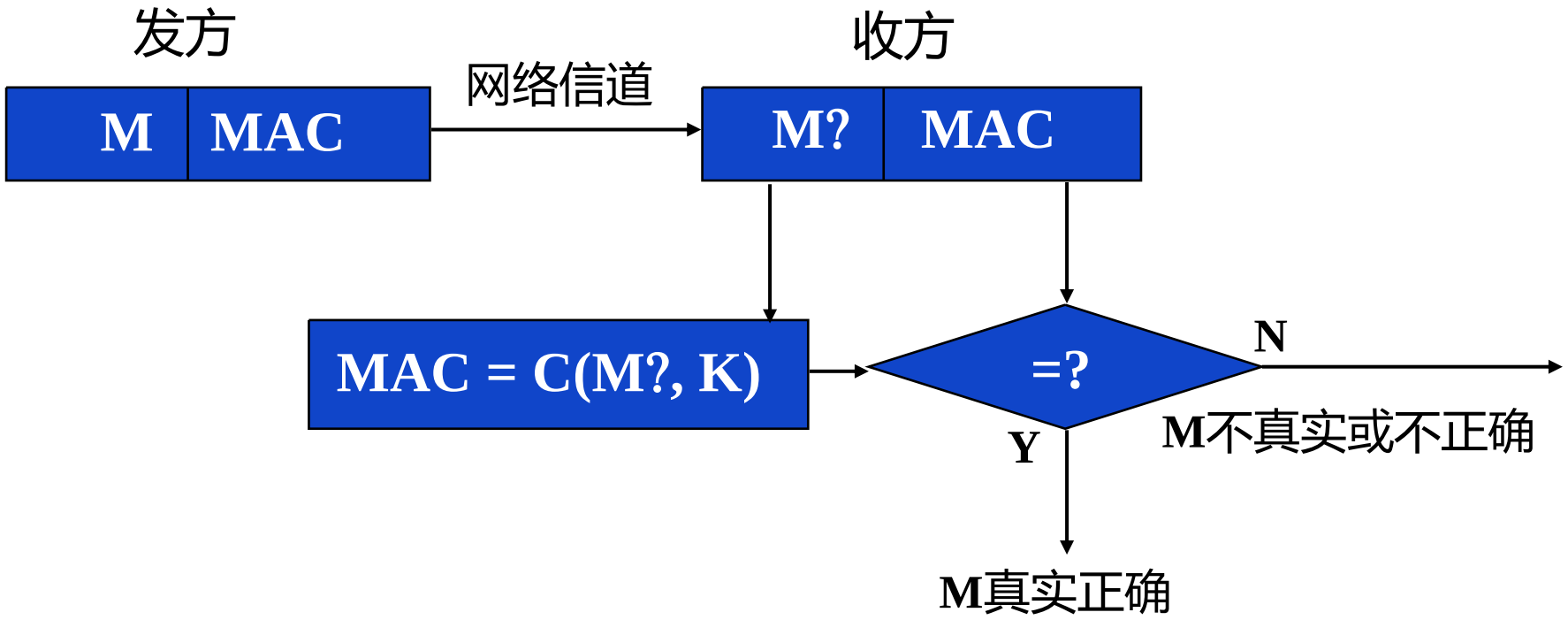
- 消息内容的认证
- 通信双方共享秘密钥K，A计算MAC并将消息M和MAC发送给接收方：

$$A \rightarrow B: < M||MAC >$$

- 接收方收到消息M后用相同的秘密密钥K重新计算得出新的MAC，并将其与接收到的MAC进行比较，若二者相等，则认为消息是真实的完整的

消息认证

- 消息内容的认证



消息认证

- 消息内容的认证
- 在上述方法中，消息是以明文形式传送的，所以该方法可以提供认证，但不能提供保密性
- 若要获得保密可在MAC算法之后对消息加密： $A \rightarrow B: E(M||MAC, K_2)$，其中 $MAC = C(M, K_1)$
- 安全性分析：因为只有A和B共享K_1，所以可提供认证；因为只有A和B共享K_2，所以可提供保密
- 另一种方案： $A \rightarrow B: E(M, K_2)||C(E(M, K_2), K_1)$


感谢聆听!
THANK YOU FOR YOUR ATTENTION!