第一部分 加解密

03. 公钥密钥

厚 德 健 行

- 双钥
  - 发送者使用接收者的**公钥**加密数据
  - 接收者使用其自身的**私钥**解密数据
- 基于"**单向陷门函数**"
  - 一个函数正向计算很容易，但是反向计算则是非常困难的
  - 如: 给定两个素数p和q,计算乘积N=pq是很容易的,但是给定N,分解出它的因子p和q则是很困难的
  - "陷门"的目的是用于生产密钥

- 加密
  - 假设我们使用Bob的公钥加密消息M
  - 那么只有Bob的私钥可以对消息M解密
- 数字签名
  - Bob还能够使用他的私钥进行"加密"得到消息M的签名
  - 任何人可以用Bob的公钥来验证他的数字签名
  - 但只有私钥拥有者才能进行签名
  - 数字签名与手工签名很类似

RSA

- RSA的思想最早由GCHQ的Cocks提出的，数年后Rivest, Shamir和Adleman才重新提出了它
  - RSA算法是公共密钥加密的黄金标准
- 令 p和q为两个大素数
- 计算乘积 $N = pq$
- 然后选择与$(p-1)(q-1)$互素的整数
- 求解模$(p-1)(q-1)$的e的乘法逆d，并且使其满足$ed = 1 \bmod (p-1)(q-1)$
- 公钥：(N,e)
- 私钥： d

- 将信息M当成成是一串数字
- 将消息M以加密指数e做乘方，并取模
  - $C = M^e \bmod N$
- 将密文C以解密指数d做乘方，并取模N
  - $M = C^d \bmod N$
- 这里e和N是公开的
- 因为$ed = 1 \bmod (p-1)(q-1)$，如果攻击者能分解N，她就能使用公钥e轻松地找出私钥d
- 只要能分解模数N就能攻破RSA
  - 因子分解是唯一一种攻破RSA的方法吗？

RSA如何正确工作?

- 给定 $C = M^e \bmod N$ ， 我们必须验证
  - $M = C^d \bmod N = M^{ed} \bmod N$
- **Euler定理**
  - 如果x与n互素，那么$x^{\varphi(n)} = 1 \bmod n$
- 已知:
  - $ed = 1 \bmod (p-1)(q-1)$
  - 以及$\varphi(N) = (p-1)(q-1)$
  - 对于某个整数K，有
  - $ed - 1 = k(p-1)(q-1) = k\varphi(N)$
- $M^{ed} = M^{(ed-1)+1} = M \cdot M^{ed-1} = M \cdot M^{k\varphi(N)}$
  $= M \cdot (M^{\varphi(N)})^k \bmod N = M \cdot 1^k \bmod N = M \bmod N$

- RSA例子
  - 选择两个"大"素数$p = 11$和 $q = 3$
  - 由此得出 $N = pq = 33$ 且 $(p-1)(q-1) = 20$
  - 选择加密指数 $e = 3$
  - 因为$ed = 1 \bmod 20$，我们计算得出相应的解密指数$d = 7$
- **公钥:** $(N, e) = (33, 3)$
- **私钥:** $d = 7$

RSA示例

- **公钥：** $(N, e) = (33, 3)$
- **私钥：** $d = 7$
- 假定消息$M = 8$
- 计算密文*C*：
  - $C = M^e \bmod N = 8^3 = 512 = 17 \bmod 33$
- 从密文*C*恢复出原始的明文*M*：
  - $M = C^d \bmod N = 17^7 = 410{,}338{,}673$
    $= 12{,}434{,}505 * 33 + 8 = 8 \bmod 33$

浙江工业大学 ZHE JIANG UNIVERSITY OF TECHNOLOGY

- 指数乘方取模例子
  - $5^{20} = 95367431640625 = 25 \bmod 35$
- 反复平方乘
  - $20 = (10100)_2$
  - $(1, 10, 101, 1010, 10100) = (1, 2, 5, 10, 20)$
  - 因此 $1=0 \cdot 2 + 1, 2 = 1 \cdot 2, 5 = 2 \cdot 2 + 1, 10 = 5 \cdot 2, 20 = 10 \cdot 2$
  - $5^1 = 5 \bmod 35$
  - $5^2 = (5^1)^2 = 5^2 = 25 \bmod 35$
  - $5^5 = (5^2)^2 \cdot 5^1 = 25^2 \cdot 5 = 3125 = 10 \bmod 35$
  - $5^{10} = (5^5)^2 = 10^2 = 100 = 30 \bmod 35$
  - $5^{20} = (5^{10})^2 = 30^2 = 900 = 25 \bmod 35$
- 没有大数的大指数乘方取模运算，因此是高效的!

高效的RSA (2)

- 对于普通加密指数选择 $e = 3$
  - 每次公钥加密运算只需要进行两次乘法
  - 私钥解密操作仍然需要大量运算
  - 如果$M < N^{1/3}$，那么$C = M^e = M^3$，模N操作无效，可能遭受立方根攻击
  - 如果同样的消息M被三个不同的用户加密,分别得出密文C_1, C_2, C_3，那么就可以使用中国余弦定理(Chinese Remainder Theorem) 把消息恢复出来
- 在实践中通过每次加密时对消息M随机填充数据或对消息M添加用户身份信息，可以避免此类攻击
- 另外一个常用的加密指数是 $e = 2^{16} + 1$

Diffie-Hellman算法

- 由GCHQ的Williamson提出，随后不久Diffie和Hellman也独立提出了该算法
- "DH"是 "密钥交换"算法
  - 用于建立共享对称密钥
- DH不是用于加密或签名
- DH的安全性建立在离散对数的计算困难性之上。
  - 给定: g, p, 和$g^k \bmod p$
  - 求解k的问题

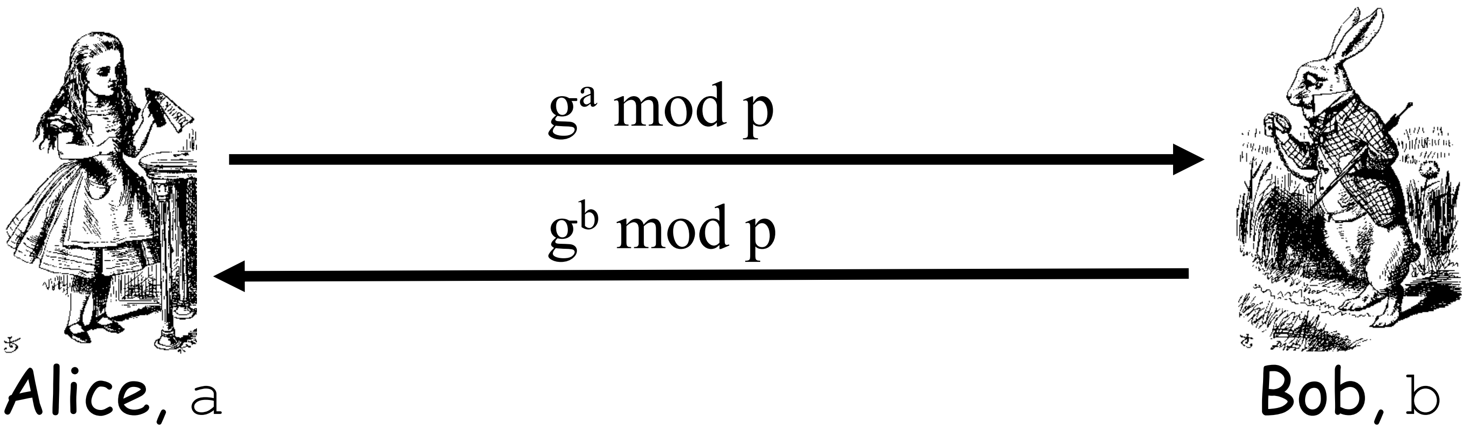
Diffie-Hellman

- 令 p 为素数，g 为生成元
  - 对任意 $x \in \{1,2,\ldots,p-1\}$，能够找到指数n使得$x = g^n \bmod p$
- Alice选择秘密指数为a
- Bob 选择秘密指数为b
- Alice发送$g^a \bmod p$给Bob
- Bob发送$g^b \bmod p$给Alice
- Both双方都计算出共享秘密$g^{ab} \bmod p$
- 共享的秘密通常用作对称密钥

- 假设Bob和Alice使用g^{ab} mod p作为对称密钥
- 攻击者Trudy能看到g^{a} mod p和g^{b} mod p
  - 因为$g^{a}\, g^{b} \bmod p = g^{a+b} \bmod p \neq g^{ab} \bmod p$
- 但如果Trudy能够找出a或b,密码系统将会被攻破
- 如果Trudy能求解离散对数问题，她就能找出a或b

DH密钥交换过程

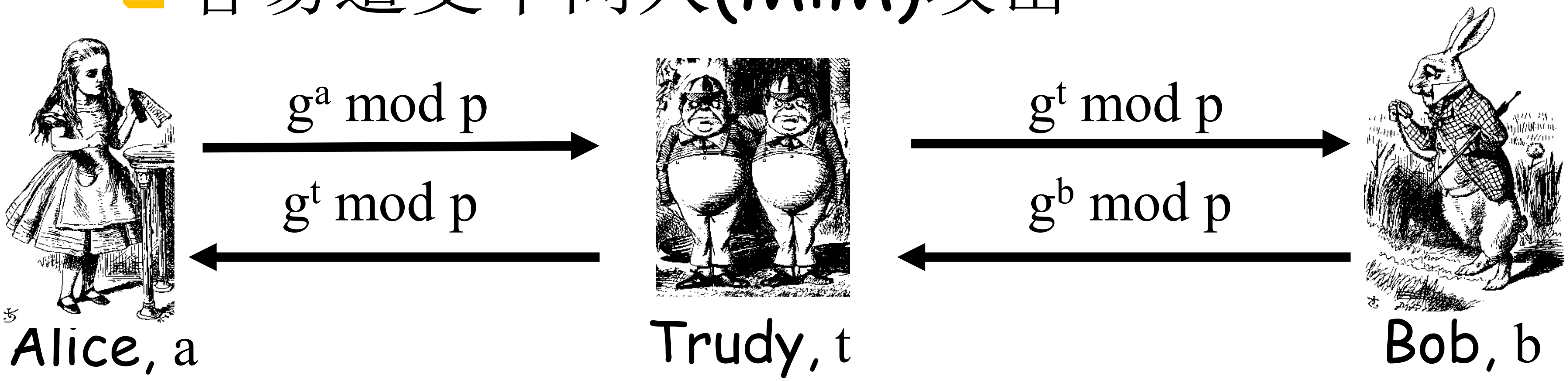
- 公开：g 和 p
- 秘密：Alice生成的秘密指数a, Bob生成的秘密指数b



- Alice计算$(g^b)^a = g^{ba} = g^{ab} \bmod p$
- Bob计算$(g^a)^b = g^{ab} \bmod p$
- 通常使用$K = g^{ab} \bmod p$作为对称密钥

对DH的中间人攻击

- 容易遭受中间人(MiM)攻击



- Trudy首先与Alice之间建立一个共享秘密g^{at} mod p
- Trudy并与Bob之间建立另外一个共享秘密g^{bt} mod p
- Alice和Bob都没有意识到Trudy的存在!

防止MiM攻击

- 如何防止MiM攻击呢?
  - 使用共享的对称密钥加密DH交换
  - 使用公钥加密 DH交换
  - 使用私钥签名DH交换
  - 其他?
- 在这一点上，DH看起来毫无意义
  - 但其实不是的（协议）
- 无论如何，你必须意识到DH算法容易遭受MiM的攻击

椭圆曲线密码

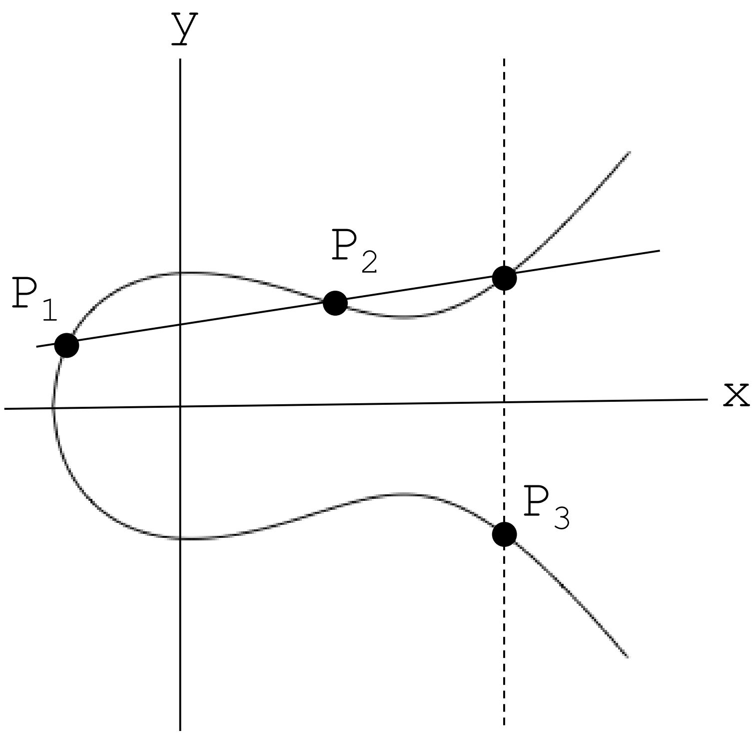
椭圆曲线密码 (ECC)

- "椭圆曲线" 并不是一个特定的密码系统
- 椭圆曲线只是为公钥密码需要的复杂数学运算提供另外一种方案
- 椭圆曲线有DH, RSA等版本.
- 椭圆曲线与其他非椭圆曲线密码相比
  - 相同安全强度所需要的参数更小
  - 但操作更为复杂

何为椭圆曲线?

- 椭圆曲线E是具有如下形式的函数的图

  E: $y^2 = x^3 + ax + b$
- 其中还包括一个定义的特殊的无穷远点：∞
- 椭圆曲线是怎样的形状?
- 如下页左图!



- 考虑椭圆曲线

$$E:\ y^2 = x^3 - x + 1$$

- 若P_1 和 P_2 位于E上,我们可以定义:

$$P_3 = P_1 + P_2$$

如图所示

- 考虑 $y^2 = x^3 + 2x + 3 \pmod 5$ 上点

$x = 0 \Rightarrow y^2 = 3 \Rightarrow$ 无解 $\pmod 5$

$x = 1 \Rightarrow y^2 = 6 = 1 \Rightarrow y = 1,4 \pmod 5$

$x = 2 \Rightarrow y^2 = 15 = 0 \Rightarrow y = 0 \pmod 5$

$x = 3 \Rightarrow y^2 = 36 = 1 \Rightarrow y = 1,4 \pmod 5$

$x = 4 \Rightarrow y^2 = 75 = 0 \Rightarrow y = 0 \pmod 5$

- 那么椭圆曲线上的点为

(1,1) (1,4) (2,0) (3,1) (3,4) (4,0) 和∞

- 给定: 曲线E: $y^2 = x^3 + ax + b \pmod p$
  - E上的点$P_1=(x_1,y_1)$, $P_2=(x_2,y_2)$
- 求解: $P_1 + P_2 = P_3 = (x_3, y_3)$

  算法:

  $x_3 = m^2 - x_1 - x_2 \pmod p$

  $y_3 = m(x_1 - x_3) - y_1 \pmod p$

  这里 $m = (y_2-y_1)*(x_2-x_1)^{-1} \bmod p, P_1 \neq P_2$

  $m = (3x_1^2+a)*(2y_1)^{-1} \bmod p, P_1 = P_2$

  特例1: 若 m 为∞, 则$P_3 = \infty$,

  特例2: 对于所有的P,有$\infty + P = P$

- 给定曲线E: $y^2 = x^3 + 2x + 3 \pmod 5$.
  - E上的点 (1,1) (1,4) (2,0) (3,1) (3,4) (4,0) 和 ∞
- $(1,4) + (3,1) = P_3 = (x_3, y_3)$?

$$m = (1-4)*(3-1)^{-1} = -3*2^{-1}$$

$$= 2(3) = 6 = 1 \pmod 5$$

$$x_3 = 1 - 1 - 3 = 2 \pmod 5$$

$$y_3 = 1(1-2) - 4 = 0 \pmod 5$$

- 于是有, $(1,4) + (3,1) = (2,0)$

ECC Diffie-Hellman算法

- 公开：椭圆曲线及其上点 (x,y)
- 秘密：Alice选择自己的秘密乘数为A， Bob选择自己的秘密乘数为B

A(x,y) →

← B(x,y)

Alice, A

Bob, B

- Alice 计算 A(B(x,y))
- Bob 计算 B(A(x,y))
- 因为A(B(x,y))与B(A(x,y))相同，所以 AB = BA

- 公开：曲线 $y^2 = x^3 + 7x + b \pmod{37}$ 及点(2,5) $\Rightarrow b = 3$
- Alice的秘密乘数：A = 4
- Bob的秘密乘数：B = 7
- Alice向Bob发送：4(2,5) = (7,32)
- Bob向Alice发送：7(2,5) = (18,35)
- Alice将计算：4(18,35) = (22,1)
- Bob将计算：7(7,32) = (22,1)

公钥密码的应用

公钥密码的应用

- 机密性
  - 在不安全的通道中传输数据
  - 在不安全的媒介中存储数据
- 认证
- 数字签名可以用来保护数据的完整性和不可否认性
  - 公钥密码来实现不可否认性

不可否认

- 假设Alice从Bob那订购了100 股股票
- Alice使用共享对称密钥计算**MAC**
- 股票暴跌后, Alice宣称他没有下过订单，否认了此交易
- 那么Bob能否证实Alice曾经下过订单?
- **他不能!** 因为Bob也知道对称密钥，他可以伪造Alice在订单上放置的消息
- **问题:** 尽管Bob知道Alice确实下了订单，但是却不能证明这一点

不可否认

- 假设Alice从Bob那订购了100 股股票
- Alice使用其私钥对订单进行了签名
- 股票暴跌后，Alice宣称他没有下过订单，否认了此交易
- 那么Bob能否证实Alice曾经下过订单?
- **能!** 因为只有Alice才拥有她自己的私钥
- 这是有个假设条件：Alice的私钥没有被偷

公钥密码符号

浙江工业大学 ZHE JIANG UNIVERSITY OF TECHNOLOGY

- 使用Alice的**私钥**对消息M进行解密(**签名**)：$[M]_{Alice}$
- 使用Alice的**公钥加密**消息：$\{M\}_{Alice}$
- 因此

$$\{[M]_{Alice}\}_{Alice} = M$$

$$[\{M\}_{Alice}]_{Alice} = M$$

签名并加密
VS
加密并签名

- 假设我们希望确保秘密性和不可否认性
- 公钥密码可以达到此目的吗?
- Alice向Bob发送消息
  - 签名并加密： $\{[M]_{Alice}\}_{Bob}$
  - 加密并签名： $[\{M\}_{Bob}]_{Alice}$
- 操作顺序对安全性有无影响?

签名并加密

- 消息M = "I love you"

$\{[M]_{Alice}\}_{Bob}$ →

$\{[M]_{Alice}\}_{Charlie}$ →

Alice　　　　Bob　　　　Charlie

- **Q:** 存在什么问题?
- **A:** Charlie将产生误解!

加密并签名

- 消息M = "My theory, which is mine...."

Alice

$[\{M\}_{Bob}]_{Alice} \longrightarrow$

Charlie

$[\{M\}_{Bob}]_{Charlie} \longrightarrow$

Bob

- 注意：Charlie没有对M进行解密
- Q: 存在什么问题?
- A: Bob将产生误解!

公钥基础设施

公钥证书

- 一个保护用户名和用户公钥的证书
- 证书必须是由签证机构（CA）签发
  - M = (Alice, Alice’s public key), $S = [M]_{CA}$
  - Alice’s Certificate = (M, S)
- 证书上的签名是通过拥有者的公钥验证的
  - 证明 $M = \{S\}_{CA}$

签证机构

- 签证机构 (CA) 为可信的第三方 (TTP)，可以进行证书的签发和认证
- 确认该证书对应的用户正是相应的私钥的拥有者y
  - 对于签名的验证过程并不是对证书源进行验证!
  - 证书是公开的!
- 如果CA出现错误，那么后果是非常严重的(如：假设VeriSign一旦将微软签发的证书错发给其他人!)
- 证书的通用格式标准是X.509

- 公钥基础设施(PKI)是安全地使用公钥密码所需要的所有事物的统称。包括：
  - 密钥生成和管理
  - 签证机构
  - 证书撤销列表
- 没有标准的PKI模型
- 可以使用数种可行的"信任模型"

- 垄断模型
  - 对于CA构建一个通用的信任组织
  - 该方法是当时最大的商业CA所采用
  - 最大缺点就是形成一个巨大的目标
  - 并且如果你不相信垄断CA，那就无法使用这个系统!

- 寡头模式
  - 多个可信CA
  - 目前网页浏览器采用的是此模式
  - 一个网页浏览器需要配置80个以上的CA证书!
  - 用户可以自由决定哪个CA寡头可以信任

PKI 信任模型

- 无政府模式
  - 任何人都可以作为CA!
  - 用户将决定哪些"CA"可以信任
  - 该方法用于PGP，在其中称为网页信任
- 为何称其为"无政府"模式?
  - 假设一个证书由Frank签署，但你不认识Frank。不过你信任Bob，Bob说Alice是值得信任的，Alice为Frank证明，那么，你应该接受的证书吗？
- 还有许多其他PKI信任模型

现实世界中的秘密性

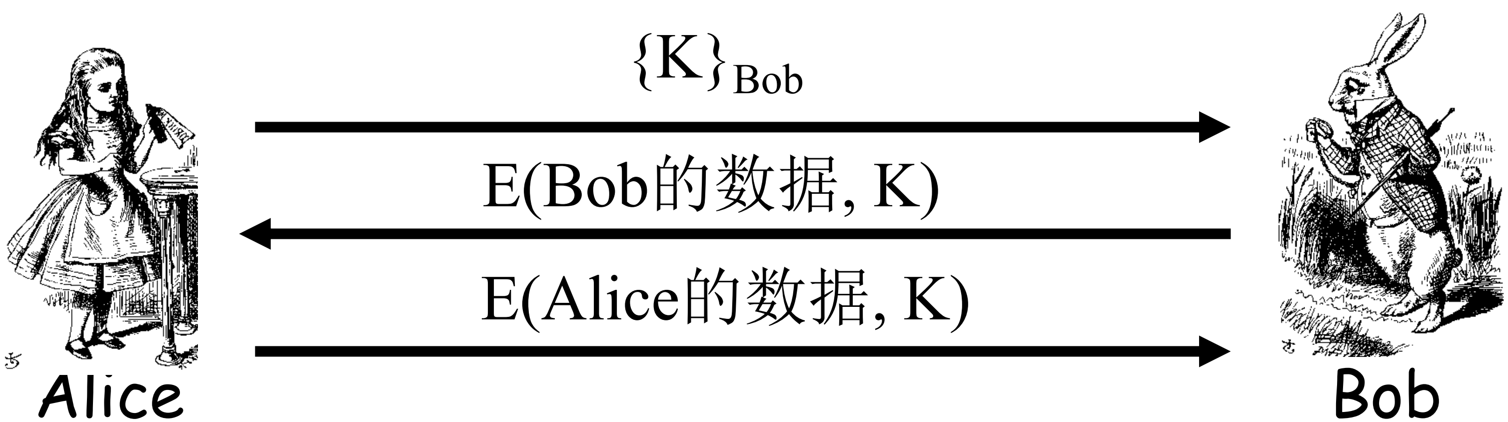
对称密钥 vs 公钥

- 对称密钥密码的优点
  - 高效
  - 不需要PKI
- 公钥密码的优点
  - 签名 (不可否认)
  - 不需要共享密钥

- 公钥概念
  - 使用Alice的私钥对消息M签名
    - $[M]_{Alice}$
  - 使用Alice的公钥对消息M加密
    - $\{M\}_{Alice}$
- 对称密钥概念
  - 使用对称密钥K对明文P加密
    - $C = E(P,K)$
  - 使用对称密钥K对密文C解密
    - $P = D(C,K)$

混合密码体制

- 混合密码体制
  - 公钥密码建立对称密钥
  - 对称密钥用于加密数据
  - 如下图所示



- Bob能判断他是否在于Alice通信?

1.假设Alice的RSA公钥是(e,N),她的私钥是d。Alice想要对消息M实施签名,也就是说,她想要计算 $[M]_{Alice}$。请列出她会用到的数学表达式。

$[M]_{Alice} = M^d \bmod N$

真的好吗？

$[h(M)]_{Alice} = h(M)^d \bmod N$

练习

2.假设Bob收到了Alice的数字证书,其发送方声称自己就是Alice。请思考下面问题。

a.请写出Alice的数字证书的表达式。在Bob验证该证书上的签名之前，他对于该证书发送方的身份能够知道多少呢？

Alice's Certificate = (M, S)

M = ("Alice" , Alice's public key), S = $[M]_{CA}$

不知道任何信息。

b. Bob如何验证该证书上的签名呢？通过验证签名，Bob能够获得什么有用的信息呢？

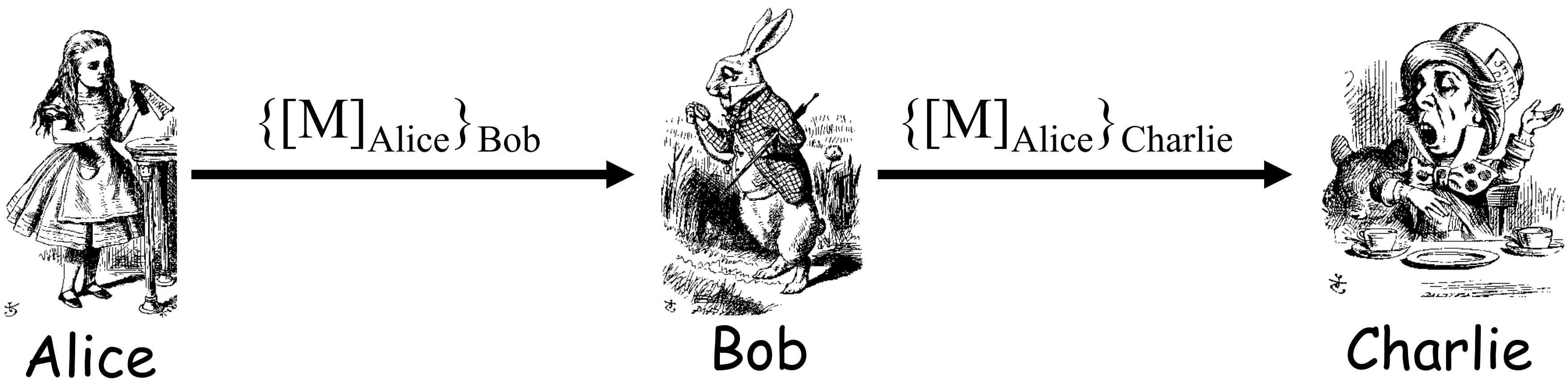
判断$\{S\}_{CA}$与M是否相同，即通过CA的公钥解密验证。可以知道证书信息是否被篡改。

c. 在Bob验证了该证书上的签名之后，他对于该证书发送方的身份又能够知道些什么呢？

不确定。只有Bob信任CA，才能确定证书中公钥对应的私钥在Alice手中。

练习

3.针对Alice签名并加密“I love you”中出现的问题，Alice是否可以使用对称秘钥加密技术防止此类攻击。



可以，Bob不知道Alice与Charlie的密钥

浙江工业大学
ZHE JIANG UNIVERSITY OF TECHNOLOGY

关注我，下节内容更精彩：
04：哈希函数