



# RSA 公钥密码体制简介

熊 虎

信息与软件工程学院

[xionghu.uestc@gmail.com](mailto:xionghu.uestc@gmail.com)

---



# RSA公钥密码体制历史



**RSA公钥密码体制历史**



**RSA公钥加密体制原理**



**RSA公钥加密体制安全吗？**



**数字签名体制介绍**



**RSA公钥密码体制现状**



# RSA公钥密码体制历史



麻省理工学院Ron Rivest、Adi Shamir和Leonard Adleman于1978年一起提出RSA加密算法，并受到广泛关注。



Published in: Communications of the ACM

## A Method for Obtaining Digital Signatures and Public-Key Cryptosystems

R.L. Rivest, A. Shamir, and L. Adleman\*

### Abstract

An encryption method is presented with the novel property that publicly revealing an encryption key does not thereby reveal the corresponding decryption key. This has two important consequences:

1. Couriers or other secure means are not needed to transmit keys, since a message can be enciphered using an encryption key publicly revealed by the intended recipient. Only he can decipher the message, since only he knows the corresponding decryption key.
2. A message can be “signed” using a privately held decryption key. Anyone can verify this signature using the corresponding publicly revealed encryption key. Signatures cannot be forged, and a signer cannot later deny the validity of his signature. This has obvious applications in “electronic mail” and “electronic funds transfer” systems.

A message is encrypted by representing it as a number  $M$ , raising  $M$  to a publicly specified power  $e$ , and then taking the remainder when the result is divided by the publicly specified product,  $n$ , of two large secret prime numbers  $p$  and  $q$ . Decryption is similar; only a different, secret, power  $d$  is used, where  $e \cdot d \equiv 1 \pmod{(p-1) \cdot (q-1)}$ . The security of the system rests in part on the difficulty of factoring the published divisor,  $n$ .

# RSA公钥密码体制历史



为奖励Ron Rivest、Adi Shamir和Leonard Adleman发明RSA公钥算法，2002年度美国计算机协会(ACM)为三位学者颁发图灵奖Turing Award。



# RSA公钥密码体制历史

---



**RSA**目前被广泛应用及部署到不同的场景，比如**HTTPS**（全称：**Hyper Text Transfer Protocol over Secure Socket Layer**，是以安全为目标的**HTTP**通道，简单讲是**HTTP**的安全版）



# RSA公钥密码体制原理



**RSA公钥密码体制历史**



**RSA公钥加密体制原理**



**RSA公钥加密体制安全吗？**



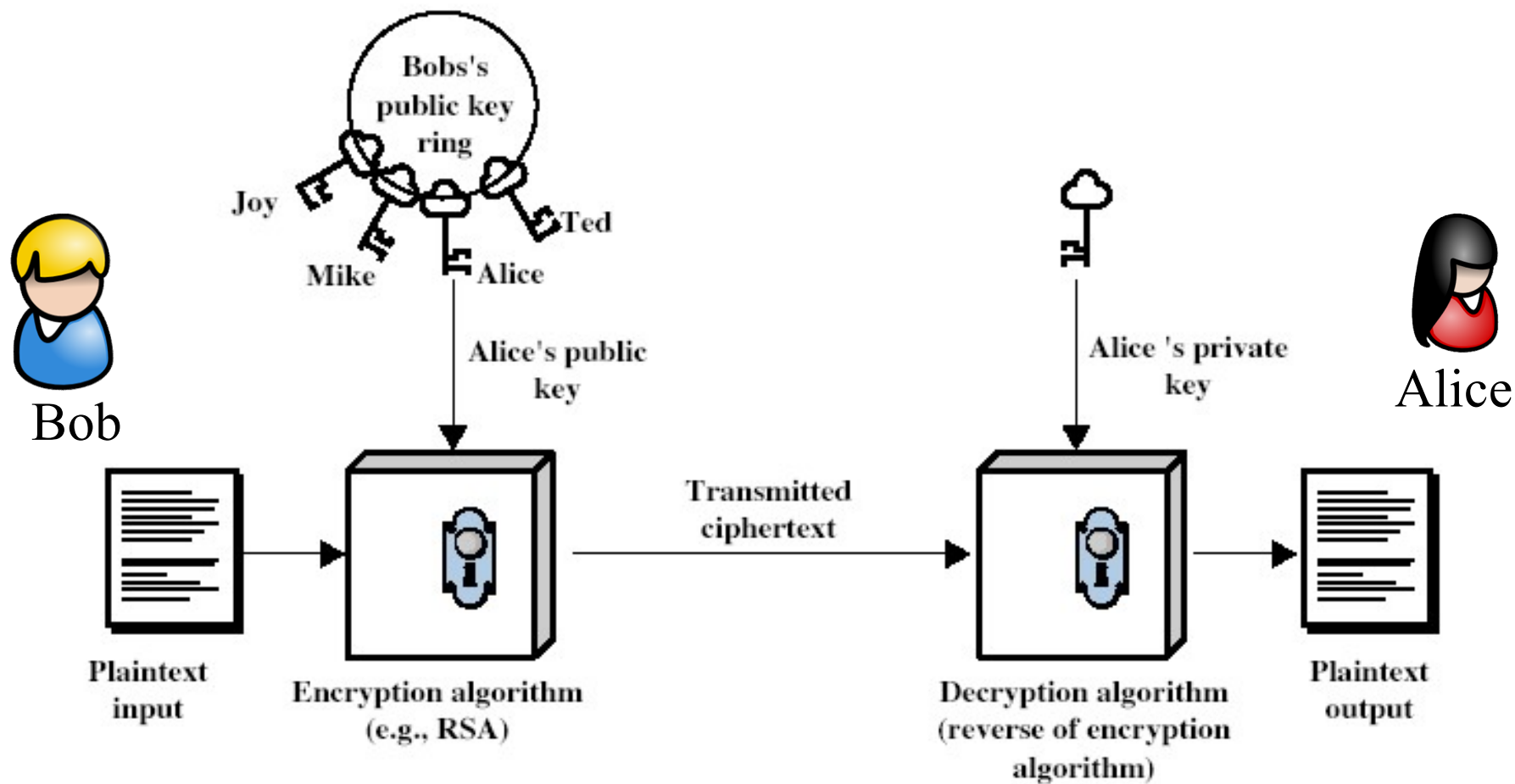
**数字签名体制介绍**



**RSA公钥密码体制现状**



# RSA公钥加密体制原理







# RSA公钥加密体制原理



密钥生成：

1. 选择两个大素数  $p, q$ 。（例如：每个**1024**位）
2. 计算  $n = pq$ ， $z = (p - 1)(q - 1)$ 。
3. 随机选取  $e$ （其中  $e < n$ ）， $e$  与  $z$  没有公因数。（ $e, z$  “互为质数”）
4. 选取  $d$  使得  $ed - 1$  能够被  $z$  完全整除。  
（换言之： $ed \bmod z = 1$ ）
5. 公钥是  $\underbrace{(n, e)}_{K_B^+}$ 。私钥是  $\underbrace{(n, d)}_{K_B^-}$ 。



# RSA公钥加密体制原理



加密/解密算法：

如上所述给出  $(n, e)$  和  $(n, d)$ 。

加密：由  $c = m^e \bmod n$  将明文  $m$  转变为密文  $c$ （即：当  $m^e$  除以  $n$  所得的余数）。

注意： $m < n$ （如果需要，则分块）

解密： $m = c^d \bmod n$ （即： $c^d$  除以  $n$  所得的余数）。

核心思想：

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



# RSA公钥加密体制原理



由欧拉定理得出：

当  $\gcd(a, N) = 1$  时，  $a^{\phi(N)} \bmod N = 1$  。

在**RSA**中有：

1.  $N = p \cdot q$
2.  $\phi(N) = (p - 1)(q - 1)$
3. 选择整数  $e$  和  $d$ ，  $d$  为  $e$  关于模  $\phi(N)$  的模反元素
4.  $e \cdot d = 1 + k \cdot \phi(N) \ (k > 0, k \in \mathbb{Z})$

于是有：

$$\begin{aligned} C^d &= (M^e)^d = M^{1+k \cdot \phi(N)} = M^1 \cdot (M^{\phi(N)})^k \\ &= M^1 \cdot (1)^k = M^1 = M \bmod N \end{aligned}$$



# RSA公钥加密体制原理



**Bob**选择  $p = 5, q = 7$  , 则  $n = 35, z = 24$  。  $e = 5$  (所以  $e, z$  互为质数)  $d = 29$  (所以  $ed - 1$  能完全被  $z$  整除) 。

加密:      letter                  m                  m<sup>e</sup>                  c = m<sup>e</sup> mod n

                 1                      12                  1524832                  17

解密:                  c                      c<sup>d</sup>                  m = c<sup>d</sup> mod n      letter

                 17      481968572106750915091411825223071697                  12                  1



# RSA公钥加密体制原理



1. 选取质数:  $p = 17$  和  $q = 11$  ;
2. 计算  $n = pq = 17 \times 11 = 187$  ;
3. 计算  $\phi(n) = (p - 1)(q - 1) = 16 \times 10 = 160$  ;
4. 选取  $e$ :  $\gcd(e, 160) = 1$  ; 选择  $e = 7$  ;
5. 确定  $d$ :  $de = 1 \pmod{160}$  且  $d < 160$  从而确定  $d$  的值  $d = 23$  , 因为  $23 \times 7 = 161 = 10 \times 160 + 1$  ;
6. 公开公钥  $KU = \{7, 187\}$  ;
7. 保留私钥  $KR = \{23, 17, 11\}$  。

# RSA公钥加密体制原理



示例**RSA**加密/解密如下:

1. 给定消息  $M = 88$  (nb.  $88 < 187$  )

2. 加密:

$$C = 88^7 \bmod 187 = 11$$

3. 解密:

$$M = 11^{23} \bmod 187 = 88$$



# RSA公钥密码体制安全吗？



**RSA公钥密码体制历史**



**RSA公钥加密体制原理**



**RSA公钥加密体制安全吗？**



**数字签名体制介绍**



**RSA公钥密码体制现状**



# RSA公钥加密体制安全吗？



## 教科书式的RSA加密体制

### 算法 8.1 RSA 密码体制

#### 密钥建立

为了生成用户的基本参数,用户 Alice 执行以下步骤:

1. 随机选择两个素数  $p$  和  $q$ , 满足  $|p| \approx |q|$ ; (\* 应用 Monte-Carlo 找素数的算法, 即算法 4.7 \*)
2. 计算  $N = pq$ ;
3. 计算  $\phi(N) = (p-1)(q-1)$ ;
4. 随机选择整数  $e < \phi(N)$ , 满足  $\gcd(e, \phi(N)) = 1$ , 并计算整数  $d$  满足

$$ed \equiv 1 \pmod{\phi(N)}$$

(\* 由于  $\gcd(e, \phi(N)) = 1$ , 这个同余式的确有一个解  $d$ , 可以应用扩展的欧几里得算法求得(算法 4.2) \*)

5. 公开她的公钥  $(N, e)$ , 安全地销毁  $p, q$  和  $\phi(N)$ , 并保留  $d$  作为她的私钥。

#### 加密

为了秘密地将  $m < N$  发送给 Alice, 发送者 Bob 生成密文  $c$  如下

$$c \leftarrow m^e \pmod{N}$$

(\* 虽然实际上明文空间是  $\mathbb{Z}_N^*$ , 在 Bob 看来, 明文空间仍然是小于  $N$  的所有正整数集合。\*)

#### 解密

为了解密密文  $c$ , Alice 计算

$$c \leftarrow m^d \pmod{N}$$





# RSA公钥加密体制安全吗？



## 参数的选取

经过比较长时间的使用和学者们的研究，从算法和计算角度看是安全的，只是随着人类计算能力的提高，**RSA**算法中的选取的参数（ $p, q$ ）越来越大，现在普遍认为， $n = pq$ 的取值为**2048**比特是安全的，这相当于**600**位的十进制整数。



# RSA公钥加密体制安全吗？



- 共模攻击
  - 扩展欧几里得算法
- 直接分解模数 $n$ 攻击
  - 欧几里得算法
- 时间攻击
- 低指数攻击
  - 中国剩余定理



# 数字签名体制介绍



**RSA**公钥密码体制历史



**RSA**公钥加密体制原理



**RSA**公钥加密体制安全吗？



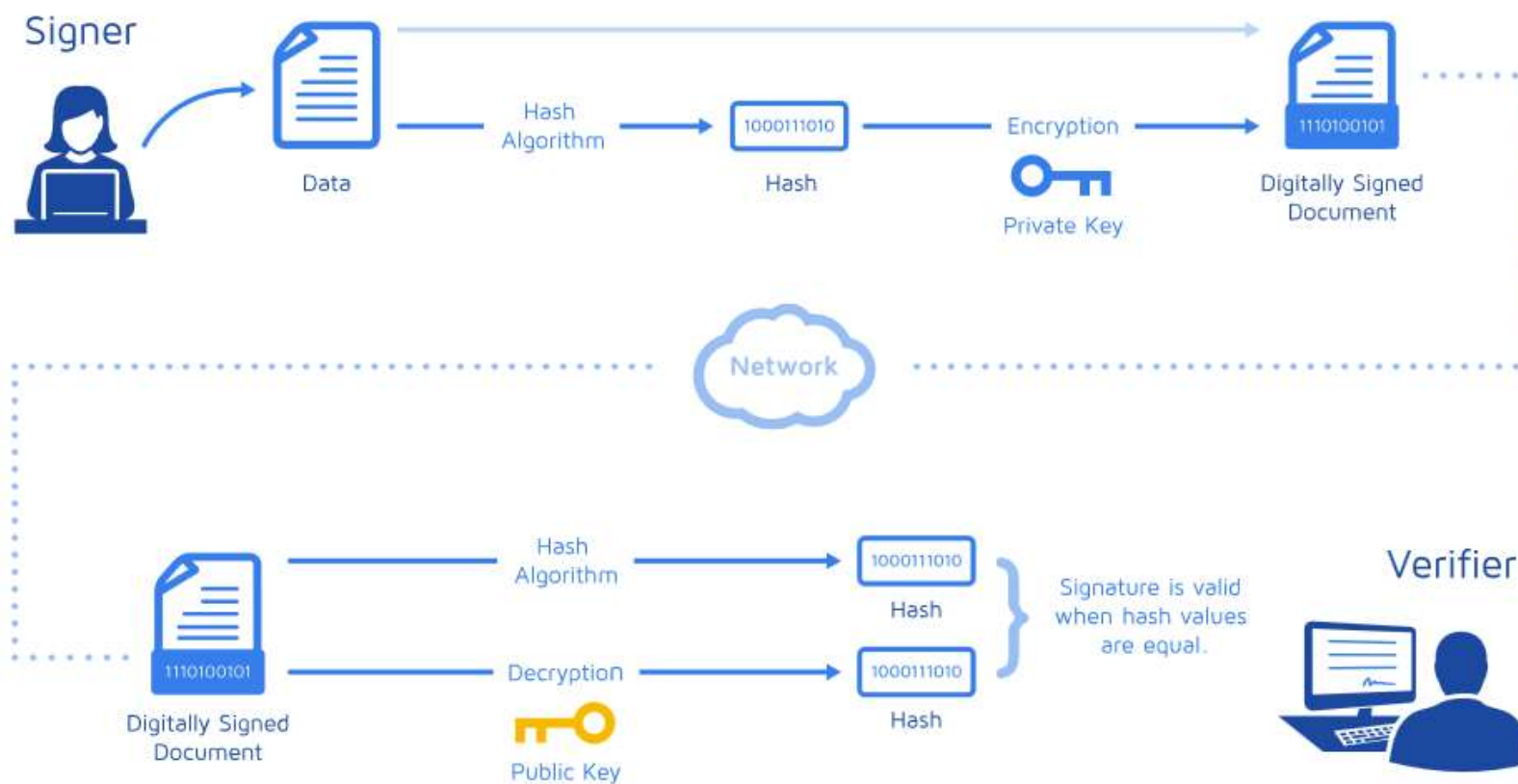
数字签名体制介绍



**RSA**公钥密码体制现状



# 数字签名体制





# RSA公钥密码体制现状



**RSA公钥密码体制历史**



**RSA公钥加密体制原理**



**RSA公钥加密体制安全吗？**



**数字签名体制介绍**



**RSA公钥密码体制现状**

# RSA 公钥密码体制现状

---



性能提升（中国剩余定理？）

可证明安全？

功能扩展（代理重加密、聚合签名...）

**PKI?**

数学困难问题？