《密码学理论与实践》

第 9 章：公钥密码学与 RSA

本章要点

- 非对称密码是一种密码体制，其加密算法和解密算法使用不同的密钥，一个是公钥，一个是私钥。非对称密码也称为公钥密码。
- 非对称密码用两个密钥中的一个以及加密算法将明文转换为密文，用另一个密钥以及解密算法从密文恢复出明文。
- 非对称密码可以用来保密、认证或者两者兼而有之。
- 应用最广泛的公钥密码体制是 RSA，破解 RSA 的困难，是基于分解大合数素因子的困难。

主要内容

1. 公钥密码学的基本原理
2. RSA 非对称加密算法
3. RSA 安全性

主要内容

1. 公钥密码学的基本原理
2. RSA 非对称加密算法
3. RSA 安全性

对称密码体制的问题

- 加密能力与解密能力是捆绑在一起的。
- 密钥更换、传递和交换需要可靠信道，密钥分发困难。
- 例如有 n 个用户，则需要 $\binom{n}{2}$ 个密钥，$n = 1000$ 时，$\binom{1000}{2} \approx 500000$, 密钥管理困难。
- 无法满足不相识的人之间通信的保密要求。
- 不能实现数字签名。

Diffie 和 Hellman 提出的设想

- 1976 年，Whitfield Diffie 和 Martin Hellman 提出了一种新设想：
- 每个用户 A 有一个加密密钥 k_a，一个解密密钥 k'_a；
- 解密密钥 k'_a 需要保密，而加密密钥 k_a 可以公开，要求 k_a 的公开不影响 k'_a 的安全；
- 若用户 B 要向用户 A 秘密发送明文 m，可查寻 A 的公开密钥 k_a，加密后得到密文 $C = E_{k_a}(m)$；
- 用户 A 收到密文 C 后，用只有用户 A 才拥有的解密密钥 k'_a 对 C 进行解密得到明文 $m = D_{k'_a}(C)$。
- 实现方案的发展依赖于单向陷门函数。

Diffie 和 Hellman

New Directions in Cryptography

Invited Paper

WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE

***Abstract*—Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing.**

The best known cryptographic problem is that of privacy: preventing the unauthorized extraction of information from communications over an insecure channel. In order to use cryptography to insure privacy, however, it is currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such as private courier or registered mail. A private conversation

Stanford | News

Home Find Stories For Journalists Contact

Stanford Report, March 1, 2016

Stanford cryptography pioneers Whitfield Diffie and Martin Hellman win ACM 2015 A.M. Turing Award

The groundbreaking algorithm from Whitfield Diffie and Martin Hellman enabled a secure Internet and sparked a clash with the NSA that foreshadowed current privacy battles between government agencies and Silicon Valley companies.

非对称密码体制的基本特点

- 加密能力与解密能力是分开的。
- 密钥分发简单，需要保存的密钥量大大减少，n 个用户只需要 n 个密钥。
- 可满足不相识的人之间保密通信。
- 可以实现数字签名。

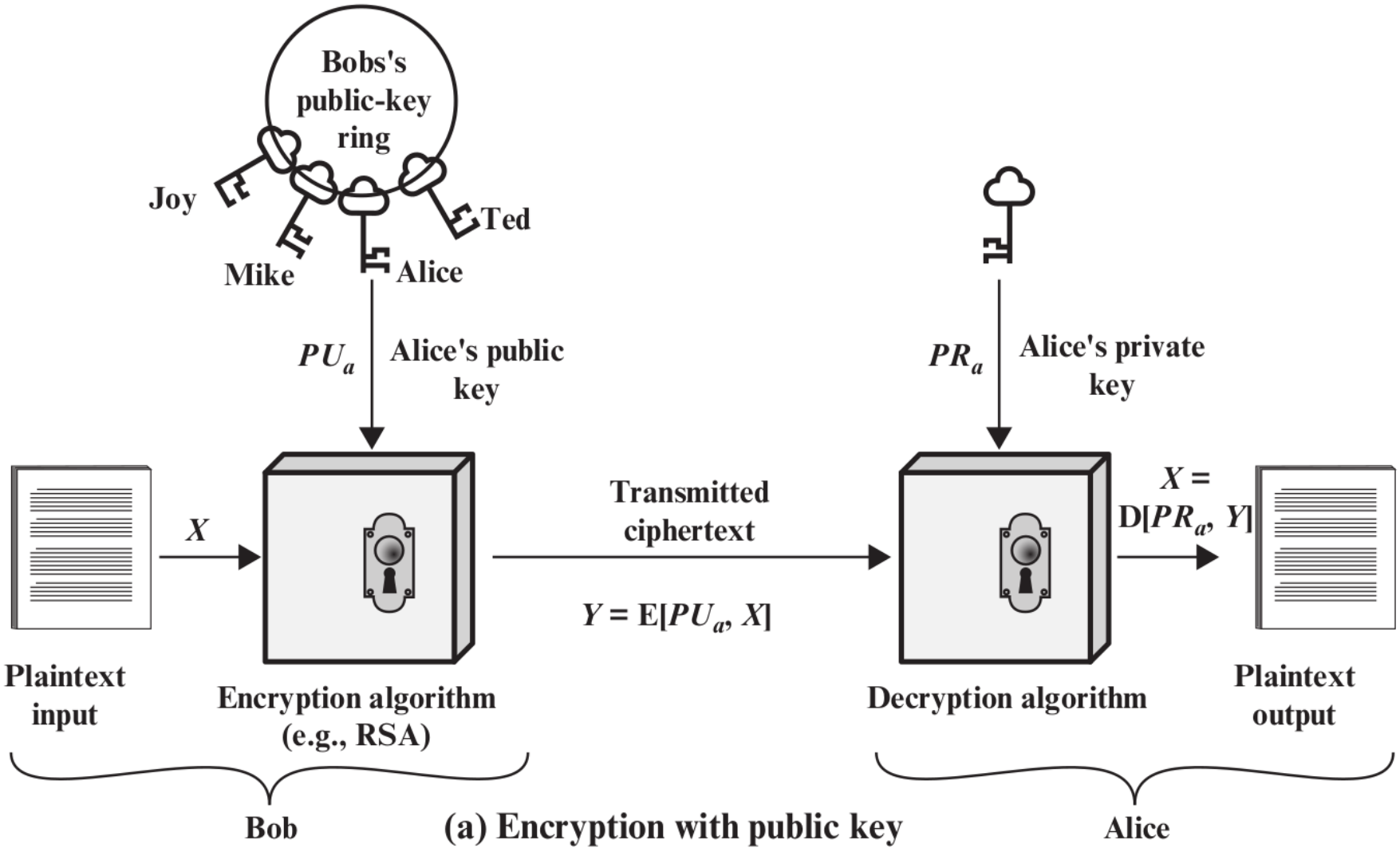
公钥密码体制

- 公钥算法依赖于一个加密密钥和一个与之相关的不同的解密密钥。算法有如下特点：
  - 仅根据密码算法和加密密钥来确定解密密钥在计算上是不可行的；
  - 两个密钥的任何一个都可用来加密，另一个用来解密。
- 公钥密码体制的组成：
  - 明文：算法的输入，可读信息或数据；
  - 加密算法：对明文进行各种转换；
  - 公钥和私钥：算法的输入，分别用于加密和解密；
  - 密文：算法的输出，依赖于明文和密钥；
  - 解密算法：根据密文和密钥，还原明文。

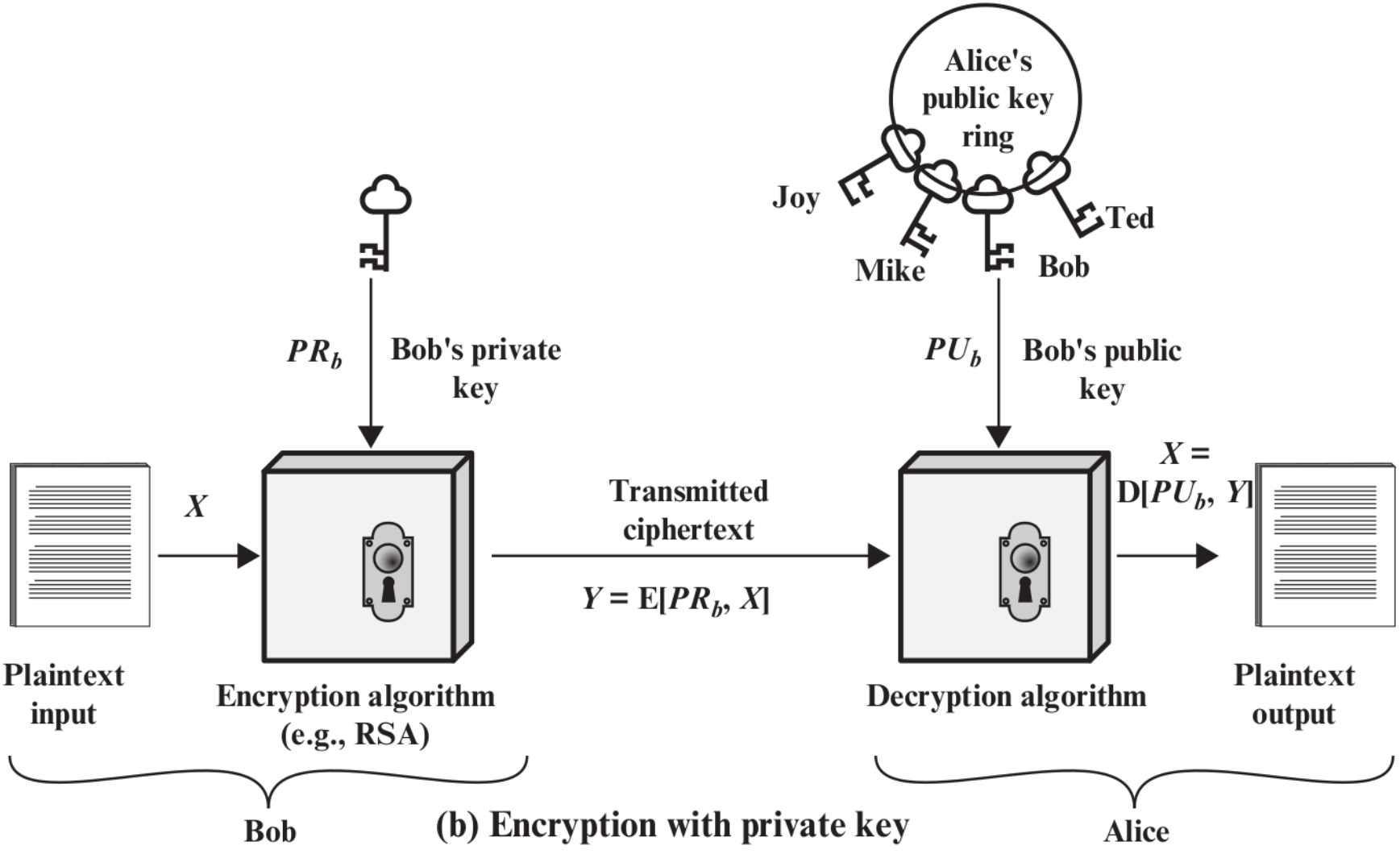
算法的主要步骤

- 每个用户产生密钥，用来加密和解密消息；
- 每个用户将其中一个密钥（公钥）存于公开的寄存器或其他可访问的文件中，另一密钥私有，每个用户可以拥有若干其他用户的公钥；
- 若 Bob 要发消息给 Alice，则用 Alice 的公钥加密；
- Alice 收到密文后，用其私钥解密，由于只有 Alice 知道私钥，所以其他接收者不能解密；
- 需要认证时示证方用自己的私钥加密消息（签名）；
- 验证方用示证方的公钥解密消息（验证），如果结果证实公钥与示证方的私钥相吻合，则可以确认示证方确为合法的用户（认证）；
- 加密和认证可以结合起来，同时实现保密性和认证。

加密过程


Bobs's public-key ring
Joy
Ted
Mike
Alice
PU_a
Alice's public key
PR_a
Alice's private key
X
Transmitted ciphertext
$Y = E[PU_a, X]$
$X = D[PR_a, Y]$
Plaintext input
Encryption algorithm (e.g., RSA)
Decryption algorithm
Plaintext output
Bob
(a) Encryption with public key
Alice

认证过程

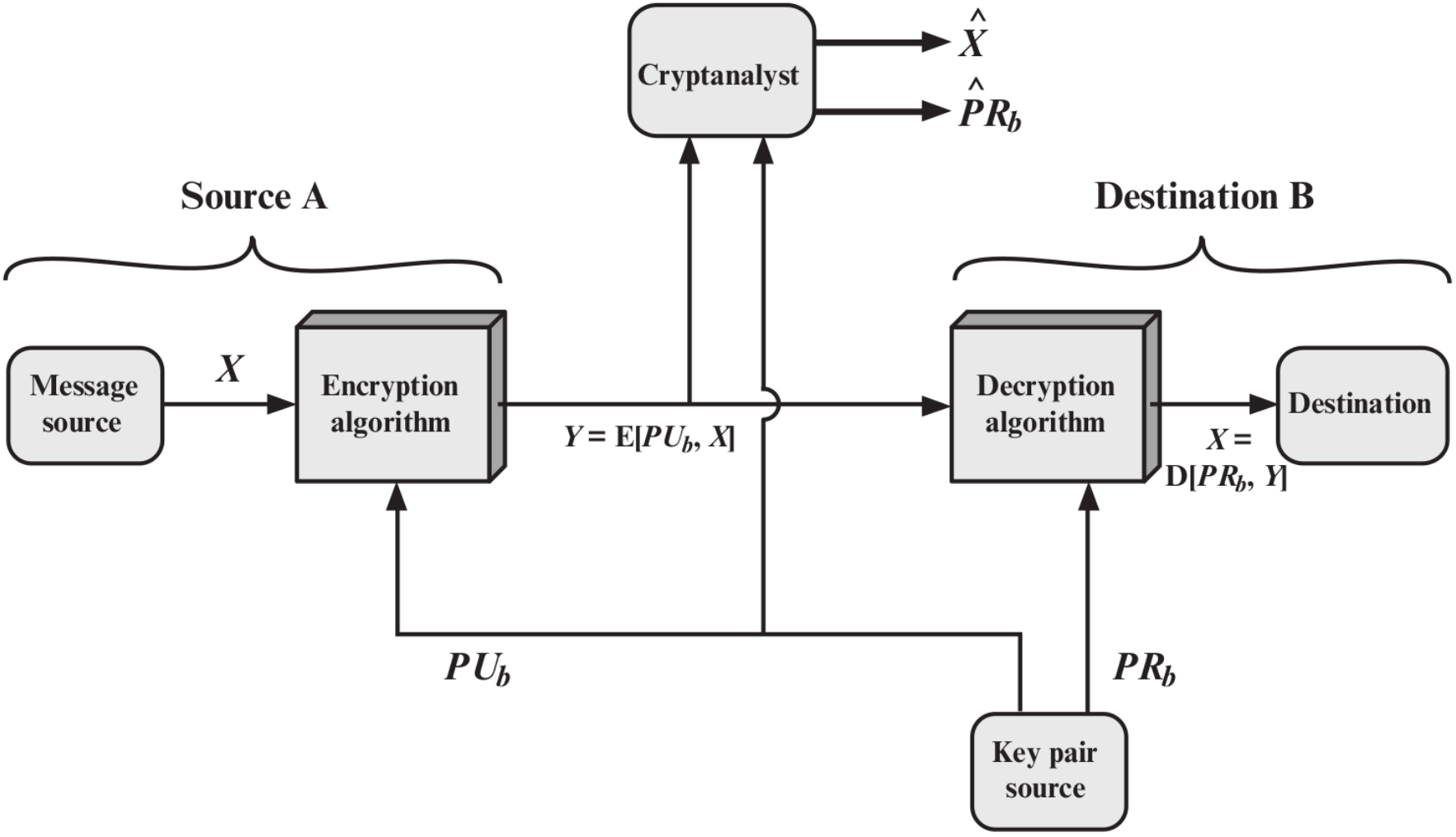


(b) Encryption with private key

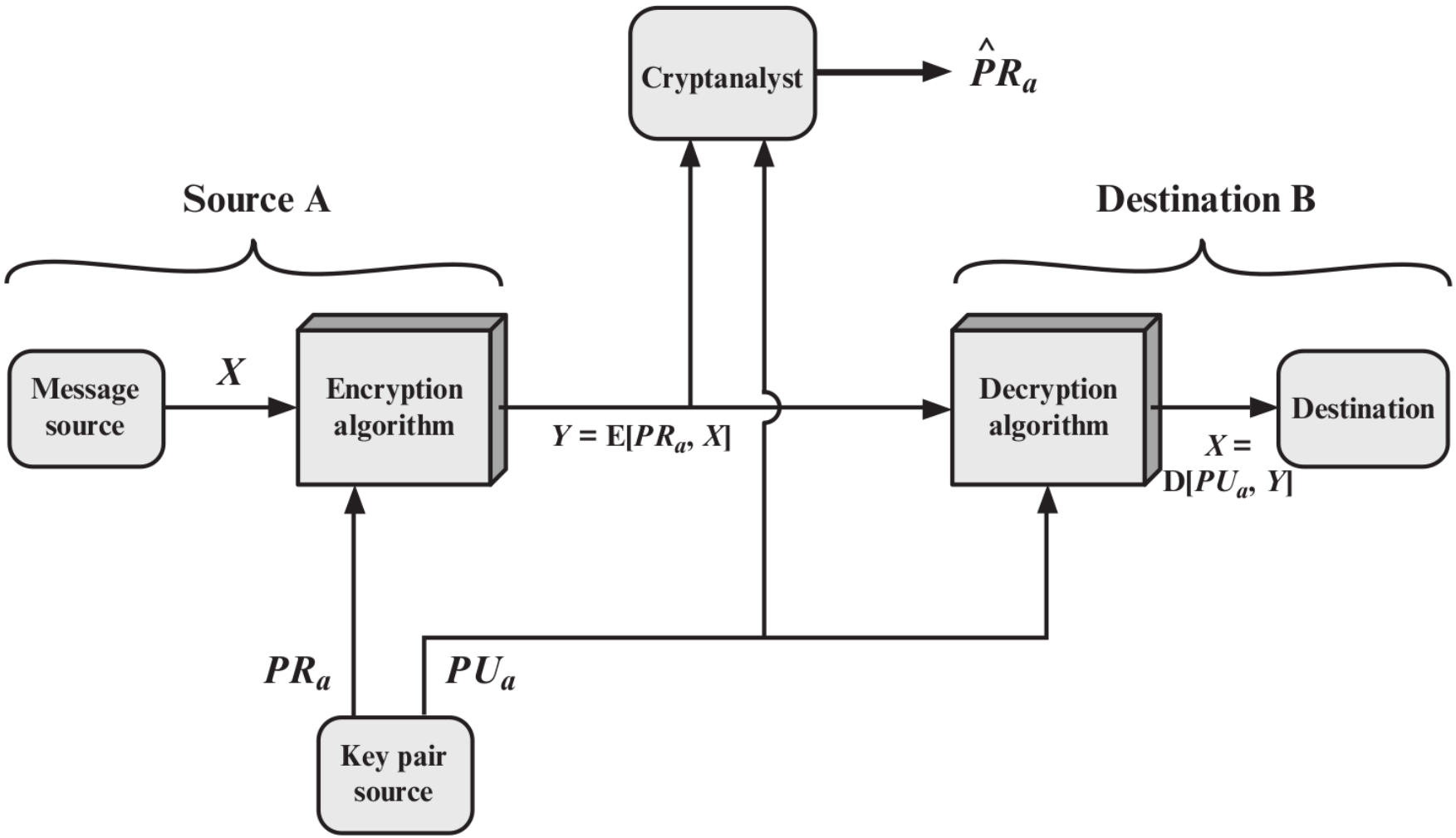
常规加密 vs 公钥加密

Conventional Encryption	Public-Key Encryption
Needed to Work: 1. The same algorithm with the same key is used for encryption and decryption. 2. The sender and receiver must share the algorithm and the key. *Needed for Security:* 1. The key must be kept secret. 2. It must be impossible or at least impractical to decipher a message if the key is kept secret. 3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	*Needed to Work:* 1. One algorithm is used for encryption and a related algorithm for decryption with a pair of keys, one for encryption and one for decryption. 2. The sender and receiver must each have one of the matched pair of keys (not the same one). *Needed for Security:* 1. One of the two keys must be kept secret. 2. It must be impossible or at least impractical to decipher a message if one of the keys is kept secret. 3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

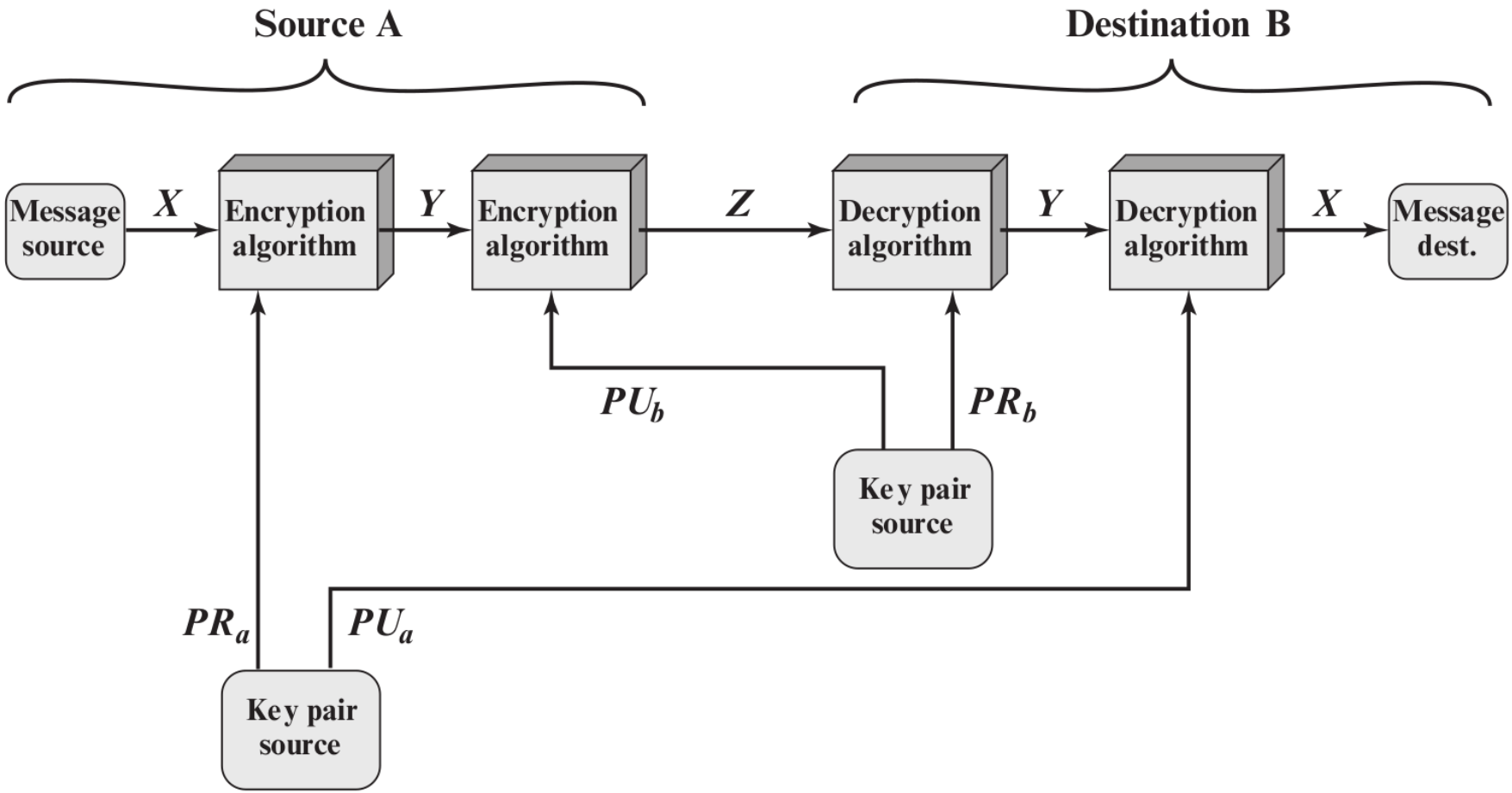
公钥密码：实现保密


Cryptanalyst
$\hat{X}$
$\hat{PR}_b$
Source A
Destination B
Message source
X
Encryption algorithm
$Y = E[PU_b, X]$
Decryption algorithm
$X = D[PR_b, Y]$
Destination
PU_b
PR_b
Key pair source

公钥密码：实现认证


Cryptanalyst
$\hat{PR}_a$
Source A
Destination B
Message source
X
Encryption algorithm
$Y = E[PR_a, X]$
Decryption algorithm
Destination
$X = D[PU_a, Y]$
PR_a
PU_a
Key pair source

公钥密码：同时实现保密与认证



公钥密码体制的应用

- 加密/解密：发送方用接收方的公钥对消息加密
- 数字签名：发送方用其私钥对消息签名，可以对整体消息签名或对消息的摘要签名
- 密钥交换：通信双方交换会话密钥

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie–Hellman	No	No	Yes
DSS	No	Yes	No

公钥密码体制的要求

- 容易产生一对密钥（公钥 PU 和私钥 PR）；
- 不难计算 $c = E(\mathrm{PU}, m)$ 和 $m = D(\mathrm{PR}, c)$；
- 知道 PU，计算 PR 不可行；
- 不知道 PR，即使知道 PU, E, D 及 c，计算 m 不可行；
- 对明文 $m, E(\mathrm{PU}, m)$ 有定义，且 $D(\mathrm{PR}, E(\mathrm{PU}, m)) = m$；
- 对密文 $c, D(\mathrm{PR}, c)$ 有定义，且 $E(\mathrm{PU}, D(\mathrm{PR}, c)) = c$；
- 两个密钥可以交换顺序，即 $D[\mathrm{PU}, E(\mathrm{PR}, m)] = D[\mathrm{PR}, E(\mathrm{PU}, m)]$。

公钥密码体制的分析

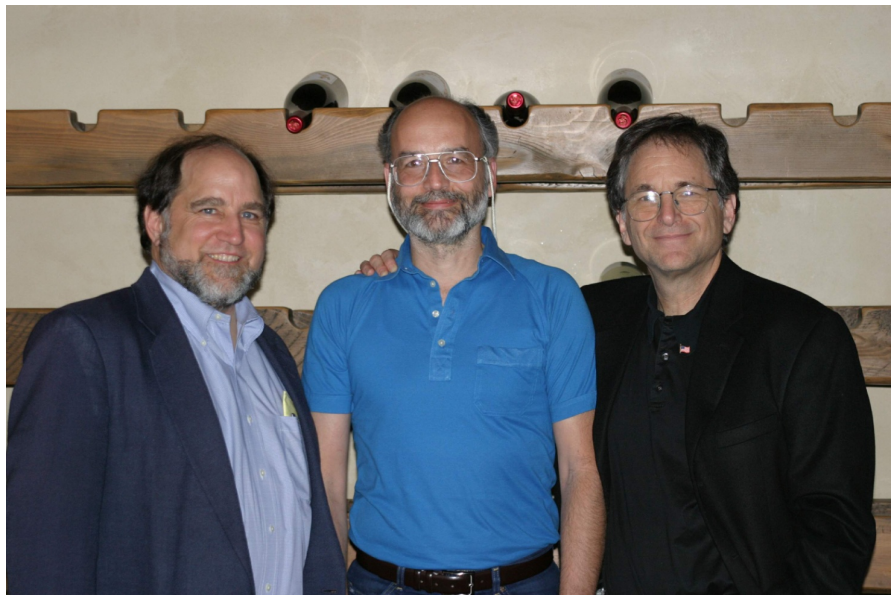
- 穷举攻击：公钥密码易受穷举攻击，解决方法是使用长密钥；同时为了便于实现加密和解密，又希望密钥足够短。目前公钥密码仅限于密钥管理和签名。
- 从给定的公钥计算出私钥：尚未在数学上证明对一特定公钥算法这种攻击是不可行的。因此包括 RSA 在内的任何算法都是值得怀疑的。
- 穷举消息攻击：攻击者用公钥对所有可能的消息加密，并与传送的密文匹配，从而解密任何消息。抵抗的方法是在要发送的消息后附加随机数。

主要内容

1. 公钥密码学的基本原理
2. RSA 非对称加密算法
3. RSA 安全性

RSA 非对称加密算法

- 1977 年，Rivest、Shamir、Adleman 提出了非对称加密算法 RSA。RSA 是基于大合数的素因子分解问题的困难性。
- 1994 年 4 月一个小组通过 Internet 合作，8 个月时间成功分解 129 位的数，大约 428 比特；1999 年分解 155 位合数，最新的记录是 2005 年 5 月分解 200 位十进制数。
- RSA 专利于 2000 年 9 月 20 日到期。



(Left to Right: Ron Rivest, Adi Shamir, Len Adleman)

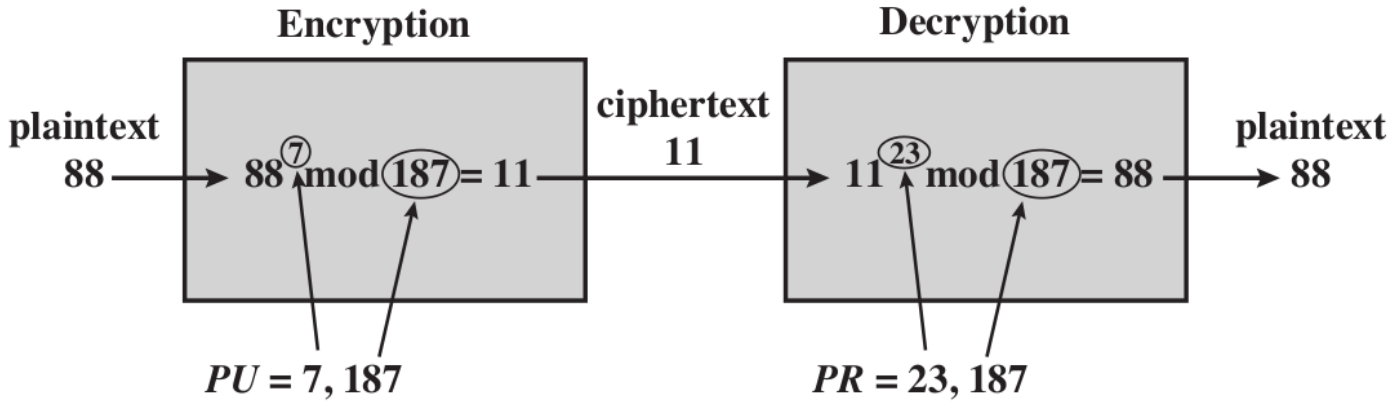
RSA 密码体制算法流程

- 随机选择两个秘密大素数 p 和 q；
- 计算公开模数 $n = pq$；
- 计算秘密的欧拉函数 $\phi(n) = (p-1)(q-1)$；
- 选择一个与 $\phi(n)$ 互素的数，作为 e 或 d；
- 用扩展 Euclid 算法计算模 $\phi(n)$ 的乘法逆元素，即根据 $ed \bmod \phi(n) = 1$, 求 d 或 e；
- 加密：$C = M^e \bmod n$，e 为公钥
- 解密：$M = C^d \bmod n$，d 为私钥

RSA 解密的计算过程

$$
\begin{aligned}
C^d \bmod n &= (M^e \bmod n)^d \bmod n \\
&= M^{ed} \bmod n && (a \bmod n)(b \bmod n) = ab \bmod n \\
&= M^{k\phi(n)+1} \bmod n && \text{因为 } ed \bmod \phi(n) = 1 \\
&= M^{(k-1)\phi(n)+\phi(n)+1} \bmod n \\
&= M^{(k-1)\phi(n)+1} \bmod n && \text{欧拉定理：} a^{\phi(n)+1} \equiv a (\bmod n) \\
&= M^{(k-2)\phi(n)+1} \bmod n \\
&\dots \\
&= M
\end{aligned}
$$

RSA 算法举例



- 选择 $p = 17, q = 11$，则 $n = pq = 187$，$\phi(n) = (p-1)(q-1) = 160$；
- 选择 $e = 7$ 满足 $\gcd(7, 160) = 1$。因为 $23 \times 7 = 161$，所以 $d = 23$；
- 公钥 $PU = 7$, 私钥 $PR = 23$；
- 明文 $M = 88$；
- 加密计算 $C = 88^7 \bmod 187 = 11$；
- 解密计算 $M = 11^{23} \bmod 187 = 88$。

RSA 密码体制基本原理

RSA 算法满足公开密钥加密的要求, 必须符合下列条件：

- 有可能找到 e, d, n 的值, 使得对所有 $M < n$ 有 $M^{ed} \bmod n = M$
- 对于所有 $M < n$ 的值, 要计算 M^e 和 C^d 是相对容易的
- 在给定 e 和 n 时, 计算 d 是不可行的

主要内容

1. 公钥密码学的基本原理
2. RSA 非对称加密算法
3. RSA 安全性

对 RSA 算法的攻击方法

- 穷举攻击：尝试所有可能的密钥
- 数学攻击：对两个素数乘积的因子分解（FAC 问题）
- 计时攻击：依赖于解密算法的运行时间
- 选择密文攻击：利用了 RSA 算法的性质

数学攻击

- 1977 年，RSA 的三位发明者在《科学美国人》杂志上发布一段密文让读者解密，解得明文者可获得 100 美元，他们预言需要 4×10^{16} 年才能解得明文。
- 这里 n 为 129 位十进制位，或 428 位二进制位。
- 但是，一个在互联网上工作的小团体只用了 8 个月的时间，于 1994 年 4 月正确解密。
- RSA 实验室也发布了使用不同 n 长度加密的密文，让公众解密。

数学攻击

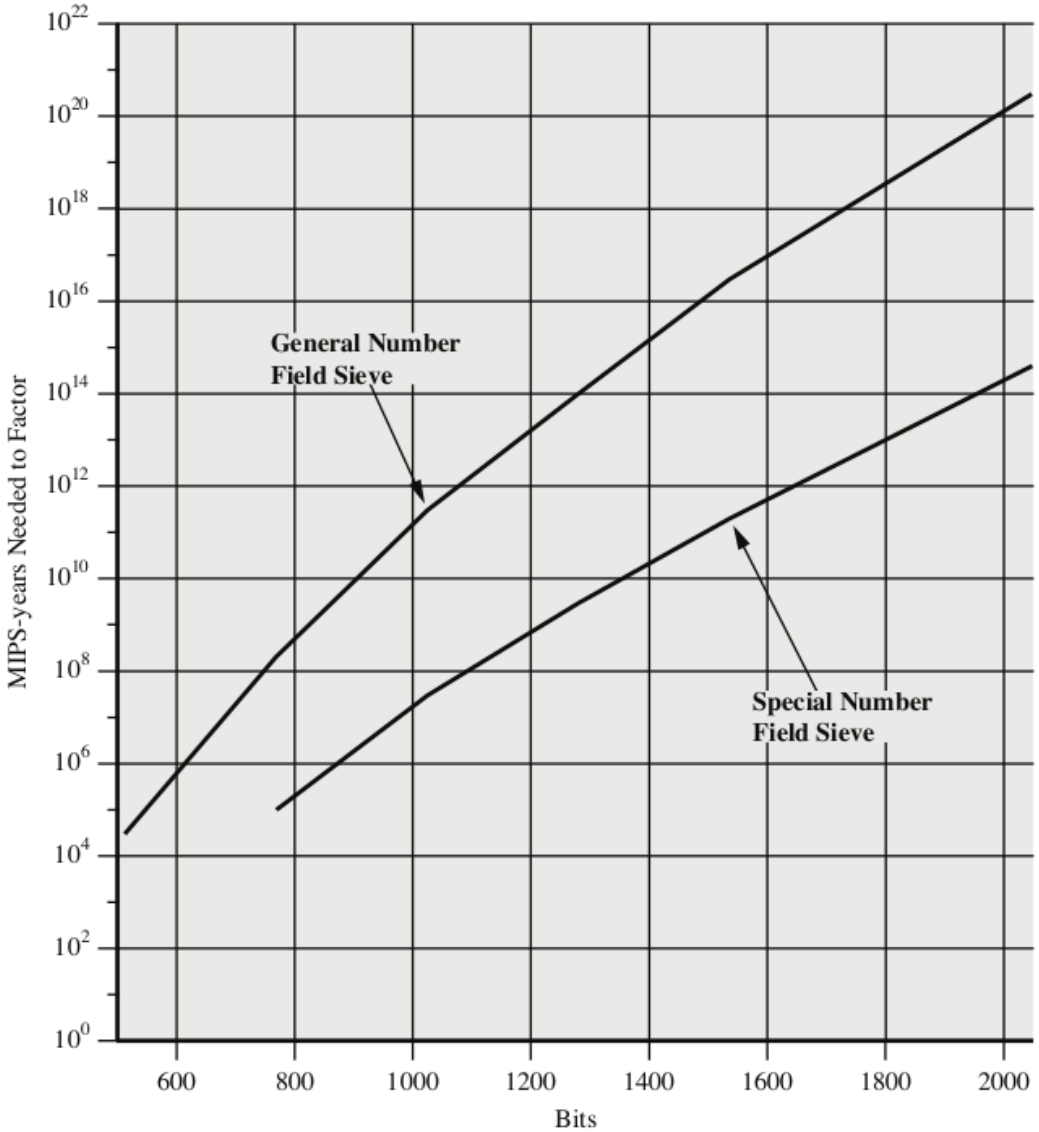
Table 9.5 Progress in RSA Factorization

Number of Decimal Digits	Number of Bits	Date Achieved
100	332	April 1991
110	365	April 1992
120	398	June 1993
129	428	April 1994
130	431	April 1996
140	465	February 1999
155	512	August 1999
160	530	April 2003
174	576	December 2003
200	663	May 2005
193	640	November 2005
232	768	December 2009

数学攻击

- n 的位数应取 1024 到 2048 位；
- p 和 q 应满足下列约束条件：
  - p 和 q 的长度应仅相差几位，p 和 q 都应约在 10^{75} 到 10^{100} 之间；
  - $(p-1)$ 和 $(q-1)$ 都应有一个大的素因子；
  - $\gcd(p-1, q-1)$ 应该较小。
- 另外，已经证明，若 $e < n$ 且 $d < n^{1/4}$，则 d 很容易确定。

MIPS-years Needed to Factor



- MIPS: million instructions per second
- MIPS-year: the number of instructions executed during one year of computing at one MIPS.
- GNFS 和 SNFS 是两种大数分解算法

计时攻击

- 类似于通过观察他人转动保险柜拨号盘的时间长短来猜测密码。
- 可能的解决办法：
  - 不变的幂运行时间，可能会降低性能
  - 在求幂运算中加入随机延时
  - 隐蔽：在执行幂运算之前先将密文乘上一个随机数

RSA 数据安全算法

- 产生 0 到 $n-1$ 之间的秘密随机数 r；
- 计算 $C' = Cr^e \bmod n$, e 是公钥；
- 计算 $M' = (C')^d \bmod n$；
- 计算 $M = M'r^{-1} \bmod n$, 其中 r^{-1} 是 r 模 n 的乘法逆元，根据 $r^{ed} \bmod n = r$，可以证明结论是正确的。

选择密文攻击（Chosen-Ciphertext Attack, CCA）

- 选择密文攻击指攻击者可以选择一些密文，并获得相应的明文；
- 基本的 RSA 算法容易受选择密文攻击；
- 攻击者可以利用 RSA 的性质，选择数据块使得当用目标对象的私钥处理时，产生密码分析所需要的信息。例如攻击者欺骗目标对象对一段看似没意义的消息做数字签名，从而利用数字签名的结果解密密文。
- 利用 CCA 攻击 RSA 利用了 RSA 如下的性质：$E(PU, M_1) \times E(PU, M_2) = E(PU, [M_1 \times M_2])$，即乘法同态特性。

选择密文攻击步骤

- 攻击者的目标是在不知道目标对象私钥的情况下解密密文 $C = M^e \bmod n$；
- 攻击者选择一个随机数 r 并计算 $X = (Cr^e) \bmod n$；
- 攻击者将 X 发送给目标对象，欺骗目标对象对 X 签名，得到 $Y = X^d \bmod n$；
- 注意到 $X = (rM)^e \bmod n$，因此攻击者收到的 $Y = rM \bmod n$，从而得到 $M = Yr^{-1} \bmod n$，其中 $r^{-1} \bmod n$ 为 r 模 n 的乘法逆元。
- 为防止 CCA 攻击，需要让 RSA 在加密之前对明文进行随机填充，破坏 RSA 的乘法同态性。

小结

1. 公钥密码学的基本原理
2. RSA 非对称加密算法
3. RSA 安全性