南京邮电大学
Nanjing University of Posts & Telecommunications

算法设计与分析

第13章 密码算法

学习要点：

- 了解信息安全的基本知识和现代密码体制
- 掌握同余、按模计算、欧拉定理、求逆等数论基础知识
- 掌握RSA算法的加密/解密原理和过程

第13章 密码算法

- 章节内容:

13.1 信息安全和密码学

13.2 数论初步

13.4 RSA算法

13.1 信息安全和密码学

信息安全的目标：

保护信息的**机密性**、**完整性**，并具有**抗否认性**和**可用性**。

信息安全的目标：

保护信息的机密性、完整性，并具有抗否认性和可用性。

机密性（confidentiality）是指非授权用户不能知晓信息内容。

一方面，可以进行访问控制（access control），阻止非授权用户获得机密信息；

另一方面，通过加密变换使非授权用户即使得到机密信息（密文形式），也无法知晓信息内容（明文）。

信息安全的目标：

保护信息的**机密性**、**完整性**，并具有**抗否认性**和**可用性**。

完整性（integrity）是指维护信息的一致性，即信息在生成、传输、存储和使用过程中不发生非授权的篡改。

一方面，可以通过**访问控制**来阻止篡改行为；

另一方面，可通过**消息认证（message authentication）**来检验信息是否已经被篡改。

信息安全的目标：

保护信息的**机密性**、**完整性**，并具有**抗否认性**和**可用性**。

抗否认性（non-repudiation）是指确保通信双方**无法事后否认**曾经对信息进行的生成、签发和接收等行为。

信息安全的目标：

保护信息的**机密性**、**完整性**，并具有**抗否认性**和**可用性**。

可用性（availability）是指保证授权用户能方便的访问所需信息。

信息安全的目标：

保护信息的**机密性**、**完整性**，并具有**抗否认性**和**可用性**。

信息安全的**机密性**、**完整性**和**抗否认性**都依赖于**密码算法**：

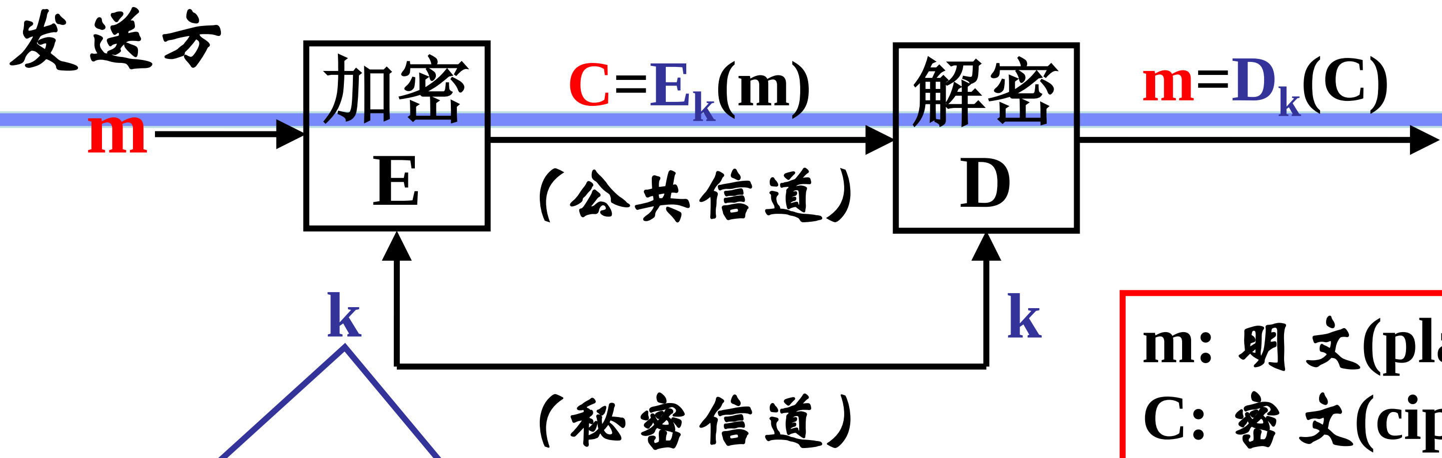
- 通过**加密**可以保护信息的**机密性**；
- 通过**信息摘要**可以检测信息的**完整性**；
- 使用**数字签名**可达到**抗否认性**的目的。

密码技术是实现信息安全的**核心技术**，是信息安全的基础。

密码学发展大致经历三个阶段：古代加密（手工阶段）、古典密码（机械阶段）和现代密码（计算机阶段）。

拆字法(青鹅= 我自与)：公元683年唐中宗即位，被武则天废除，立第四 但朝政大事均由她专断。裴炎等人不满，欲聚兵十 裴被捕，未发出的、只有"青 惑不解，武破解了秘密。
藏头
斯巴达棒：公元前405年，雅典和斯巴达之间的战争中，斯巴达军队截获了一条写满杂乱无章的希腊字母的腰带，斯巴达百思不得其解，胡乱将腰带缠到宝剑上，从而发现隐藏的军机。
凯撒密码：A->D，B->E......

恩尼格玛密码机：
德国人二战期间使用的。



m: 明文(plaintext)
C: 密文(ciphertext)
k: 密钥(key)

传统密码体制中，加密和解密所用的密钥是相同的，所以称为**对称密码（symmetric encryption）**。
这种密码体制下，通信双方使用的密钥必须通过秘密信道传递，因而**分发密钥成为薄弱环节**。

含参数k的变换
为**加密算法**。

文后，进行逆变
换m=D_k(C)，恢复明文m的过程。D_k称为**解密算法**。

用于加密和解密的数学函数称为**密码算法(cipher)**。

13.1.3 密码体制

一个密码系统包括可能的明文、密文、密钥、加密算法和解密算法。

密码系统的安全性是基于密钥的，而不是加密和解密算法自身。

因此算法往往可以作为标准公布。

密码体制从原理上分成两类：对称密码体制和非对称密码体制。

对称密码体制（symmetric encryption）

- 加密和解密使用**相同**的密钥；
- 从加密模式上分为**序列密码**和**分组密码（代表:DES）**两类。
- **序列密码**的工作方式：将明文逐位转换成密文。
- **分组密码**的工作方式：将明文分成固定长度的**组**（如：64位/组），用同一密钥和算法对每一块加密，输出也是固定长度的密文。

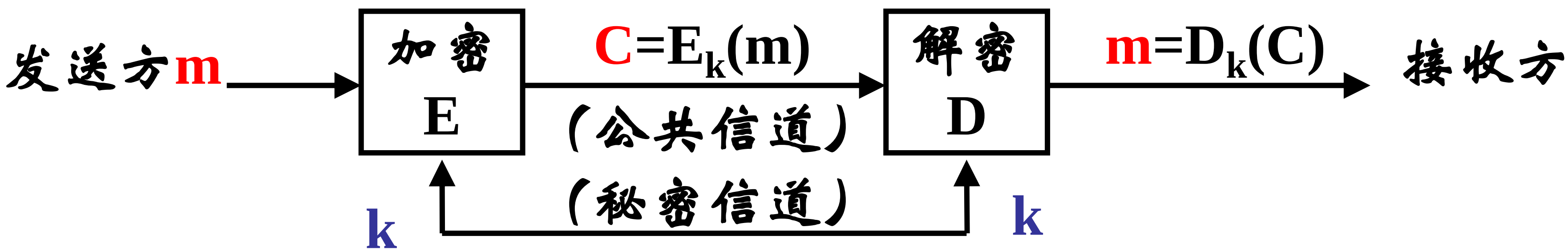


图13-1 传统保密通信机制

对称密码体制（symmetric encryption）

- 加密和解密使用**相同**的密钥；
- 从加密模式上分为**序列密码**和**分组密码（代表:DES）**两类。
- **序列密码**的工作方式：将明文逐位转换成密文。
- **分组密码**的工作方式：将明文分成固定长度的**组**（如：64位/组），用同一密钥和算法对每一块加密，输出也是固定长度的密文。

主要问题：

- 任意一对用户间均需有各自的密钥，**n个用户共需C(n,2)个密钥**，这些密钥必需在发送/接收数据之前**经秘密信道完成分发**。
- 每个用户必须心记**与其通信的n-1个用户的密钥**。

对称密码体制（symmetric encryption）

- 加密和解密使用相同的密钥；
- 从加密模式上分为序列密码和分组密码（代表:DES）两类。
- 序列密码的工作方式：将明文逐位转换成密文。
- 分组密码的工作方式：将明文分成固定长度的组（如：64位/组），用同一密钥和算法对每一块加密，输出也是固定长度的密文。

优点：

- 加、解密速度快。
- 安全强度高。

Diffie–Hellman密钥交换协议

为解决密钥的发放与管理，Whitfield Diffie和Martin Hellman于1976发布Diffie–Hellman密钥交换协议：

- 提出了公开密钥思想。
- 是离散对数问题的应用（双方各自持有a、b两个秘密整数，原根g和模p公开。g^{ab} mod p=g^{ba} mod p即是双方商定的密钥。）
- 它可以让双方在完全没有对方任何预先信息的条件下通过公共信道建立起一个密钥。
- 这个密钥可以在后续的通讯中作为对称密钥来加密通讯内容。

- 公开:大素数p 和 p的原根g(g<p)
- 秘密: Alice的指数 a, Bob的指数b (a,b<p)

原根g： g mod p， g^2 mod p……g^{p-1} mod p构成1~p-1的一个排列，如p=11,g=2）

Alice的公钥 $Y_A=g^a \bmod p$ →

← Bob的公钥 $Y_B=g^b \bmod p$

Alice，a　　　　Bob，b

- Alice 计算 $K = (Y_B)^a \bmod p= (g^b)^a = g^{ba} \bmod p$
- Bob 计算 $K = (Y_A)^b \bmod p=(g^a)^b = g^{ab} \bmod p$
- 他们可以使用 $K = g^{ab} \bmod p$ 做为对称密钥

Diffie–Hellman举例

- 选定p=97,g=5
- Alice选择自己的私有密钥 a=36,则她的公钥是

  $Ya = 5^{36} \bmod 97 = 50$
- Bob选择自己的私有密钥 b=58,则他的公钥是

  $Yb = 5^{58} \bmod 97 = 44$
- 用户Alice和Bob互换公钥后，都可以计算出

  $k = Yb^{a} \bmod 97 = 44^{36} \bmod 97 = 75$

  $k = Ya^{b} \bmod 97 = 50^{58} \bmod 97 = 75$

非对称密码体制（asymmetric encryption）

- 使用两个密钥：一个是公有密钥k_1（任何人都可以用），一个是私有密钥k_2（只有解密人可以使用）。
- 在不知道陷门信息的情况下，加密密钥和解密密钥是不能相互算出的。
- 也称公开密钥密码体制或双密钥密码体制。
- 基础是NP难度问题，如大整数分解问题。
- 最著名的代表：RSA加密算法（安全性基于大整数分解的难度）。

任何人都可用公有密钥k_1加密消息并发送给持有相应私有密钥k_2的人，只有持有k_2的人才能解密。

——实现公共网络的保密通信。

非对称密码体制（asymmetric encryption）

- 使用两个密钥：一个是公有密钥k_1（任何人都可以用），一个是私有密钥k_2（只有解密人可以使用）。
- 在不知道陷门信息的情况下，加密密钥和解密密钥是不能相互算出的。
- 也称公开密钥密码体制或双密钥密码体制。
- 基础是NP难度问题，如大整数分解问题。
- 最著名的代表：RSA加密算法（安全性基于大整数分解的难度）。

用私有密钥k_2加密的消息，任何人都可用公有密钥k_1解密，并由此证明消息来自持有k_2的人。

——实现对消息的数字签名。

13.2 数论初步

定理11-2 费马小定理（Fermat little）
若n是素数，则对所有整数0<a<n，应有$a^{n-1}\equiv 1 \pmod n$

n=5, a=3, 则$a^{n-1}=3^{5-1}=81 \bmod 5\equiv 1$
实际上对a=2、3、4，上式均恒等于1。

如果n ……。

Carmichael数虽然满足费马小定理，却是合数而不是素数。定理11-3有助于检测Carmichael数的合数性。

定理11-3 二次探测定理
如果n是一个素数，且0<x<n，则方程$x^2\equiv 1 \pmod n$有且仅有两个解为x=1和x=n-1。

定义13-1 同余

设n是一个自然数，若a-b是n的倍数，则称a与b关于模n同余，记作$a \equiv b \pmod{n}$，称b是a对模n的余数。反之，a也是b对模n的余数。

$a \equiv b \pmod{n}$等价于$a \pmod{n} = b \pmod{n}$。

$a \pmod{n} = b$意味着$a = kn + b$，k是整数。

定理13-1 按模计算原理

设a和b是整数，θ代表二元算术运算+、-或×，则

$$(a\ \theta\ b) \bmod n = [(a \bmod n)\ \theta\ (b \bmod n)] \bmod n$$

推论：$e^t \bmod n = (\prod_{i=1}^{t}(e \bmod n)) \bmod n$

按模计算的好处是：限制了中间结果的范围，使得可以对大数执行$a^t \bmod n$，而不会产生很大的中间结果。

公开密钥密码算法大量使用幂的取模运算。

定义13-2 欧拉(Euler)函数

设n是自然数，数列1,2,...,n-1中与n互素的数的个数称为n的Euler函数，记作$\Phi(n)$。

性质13-2 若p是素数，则$\Phi(p)= p-1$。

定理13-2 设p和q是素数，对n=pq，有

$$\Phi(n)= \Phi(p)\,\Phi(q)= (p-1)(q-1)$$

定理13-3 欧拉(Euler)定理

对任意整数a和n互素，则 $a^{\Phi(n)}=1 \pmod n$

当n为素数时，$\Phi(n)= n-1$，有 $a^{n-1}=1 \bmod n$。

——费马小定理

推论13-2 若 $0\le m<n$，$\gcd(m,n)=1$，有

$$m^{k\Phi(n)} \equiv 1 \pmod n$$

$$m^{k\Phi(n)+1} \equiv m \pmod n$$

定义13-3

设a是整数，若存在x使得$ax \equiv 1 \pmod{n}$，则称a与x**互逆**，**x是a关于模n的乘法逆元(inverse)**，记做$x=a^{-1}$。

由欧拉定理：若a和n互素，则$a^{\Phi(n)} \equiv 1 \pmod{n}$。

①因此$ax \equiv a^{\Phi(n)} \pmod{n}$，则$x=a^{\Phi(n)-1} \pmod{n}$。

②若n是素数，$\Phi(n)=n-1$，则$x=a^{n-2} \pmod{n}$。

定理13-4 求逆

若$\gcd(a,n)=1$，则一元同余方程$ax \equiv 1 \pmod{n}$有唯一解为：

$$x=a^{\Phi(n)-1} \pmod{n}$$

若n是素数，则进一步简化为：

$$x=a^{n-2} \pmod{n}$$

13.4 RSA算法

- 第一个较完善的公开密钥算法。
- 既能用于**加密**，也能用于**数字签名**。
- 安全性基于**大整数分解的难度**。
- 1977年由三位科学家在MIT发明，1978年公布。

产生一对密钥的过程

（1）选择两个大素数（如200位十进制数）p和q，p≠q；

（2）计算乘积n=pq，得到欧拉函数Φ(n)=(p-1)(q-1)；

（3）选择随机整数e，使得gcd(e,Φ(n))=1，且0<e<Φ(n)；

（4）计算$d=e^{-1} \bmod \Phi(n)$。d为e的关于模Φ(n)的乘法逆元，满足$ed=1 \pmod{\Phi(n)}$；

（5）则公开密钥为{e, n}，私人密钥为{d, n}。

注意：

- p和q在之后的加、解密过程中不再需要，但必须保密；
- 欧拉函数Φ(n)仅用于生成私人密钥d，之后也不再用到。

加/解密过程

（若消息很长，则将消息分成小于n的明文分组。）

若M是一个明文分组，C是M对应的密文：

◆加密公式：$C=M^e \bmod n$；

◆解密公式：$M'=C^d \bmod n$;

证明：

$M'= C^d \bmod n= (M^e \bmod n)^d \bmod n$

$=M^{ed} \bmod n$

$=M^{1+k\cdot \Phi(n)} \bmod n$

$=M \bmod n$

n为两个大素数p和q的乘积，M恰为p或q的可能性很小，因此认为M和n互质。

由推论13-2有$M^{k\Phi(n)}\equiv 1 \pmod n$

因此$M'=M$，从密文分组C能够解密恢复得到明文M。

问题：
如何从公开密钥e求得e关于模Φ(n)的逆元——私人密钥d？

用扩展Euclid算法。
如：求17关于模24的逆元

$$\begin{cases} 24*\ 1\ +17*0\ =24 \\ 24*\ 0\ +17*1\ =17 \end{cases}$$

取余

$$24*\ 1\ +17*(-1)=7$$

问题：
如何从公开密钥e求得e关于模Φ(n)的逆元——私人密钥d?

用扩展Euclid算法。

如：求17关于模24的逆元

$$\begin{cases} 24*\ 1\ +17*0\ =24 \\ 24*\ 0\ +17*1\ =17 \end{cases}$$

$$24*\ 1\ +17*(-1)=7$$

$$24*(-2)+17*3\ =3$$

取余

问题：
如何从公开密钥e求得e关于模Φ(n)的逆元——私人密钥d?

用扩展Euclid算法。

如：求17关于模24的逆元

$$\begin{cases} 24*\ 1\ +17*0\ =24 \\ 24*\ 0\ +17*1\ =17 \end{cases}$$

$$24*\ 1\ +17*(-1)=7$$

$$24*(-2)+17*3\ =3$$

$$24*\ 5\ +17*(-7)=1 \quad \textbf{mod 24}$$

取余

因此-7是17关于模24的逆元，将其正化操作得

-7+ Φ(n)=-7+24=17 是17关于模24的逆元。

例13-6 用RSA机制进行保密通信

（1）王先生产生密钥、分发公开密钥；

（2）李先生使用王先生公布的公开密钥加密消息，并发送给王先生；

（3）王先生接收李先生发送的密文，使用自己的私人密钥进行解密，恢复明文。

具体过程如下：

（1）王先生选择p=101，q=113，计算n=pq=11413，$\Phi(n)=(p-1)(q-1)=100*112=11200$；

选择加密密钥e，使得gcd(e,11200)=1。因为$11200=2^6*5^2*7$，选择e=3533。

计算解密密钥$d=e^{-1}$ (mod 11200) =6597。

王先生在网络上公布公开密钥(e, n)=(3533,11413)

具体过程如下：

（1）王先生选择p=101，q=113，计算n=pq=11413，$\Phi(n)=(p-1)(q-1)=100*112=11200$；

选择加密密钥e，使得gcd(e,11200)=1。因为$11200=2^6*5^2*7$，选择e=3533。

计算解密密钥$d=e^{-1}$ (mod 11200) =6597。

王先生在网络上公布公开密钥(e, n)=(3533,11413)

（2）李先生使用王先生的公开密钥e和n对消息M=9726加密，得到$C=9726^{3533}$ (mod 11413)=5761，并在公开信道发送密文5761。

每次乘后取模，因此中间计算结果并不大。

具体过程如下：

（1）王先生选择p=101，q=113，计算n=pq=11413，$\Phi(n)=(p-1)(q-1)=100*112=11200$；

选择加密密钥e，使得gcd(e,11200)=1。因为$11200=2^6*5^2*7$，选择e=3533。

计算解密密钥$d=e^{-1}$ (mod 11200) =6597。

王先生在网络上公布公开密钥(e, n)=(3533,11413)

（2）李先生使用王先生的公开密钥e和n对消息M=9726加密，得到$C=9726^{3533}$ (mod 11413)=5761，并在公开信道发送密文5761。

（3）王先生使用自己的私人密钥d=6597，对李先生发来的密文C进行解密，恢复明文$M=5761^{6597}$ (mod 11413)=9726。

文本加密

公钥（n，e）＝（33，3）私钥（n，d）＝（33，7）

- 令26个字母对应0-25
- 设明文为 M＝public
  - E(p)=15^3 = 9 （mod 33）
  - E(u)=20^3 = 14 （mod 33）
  - E(b)=1^3 = 1 （mod 33）
  - E(l)=11^3 = 11 (mod 33)
  - E(i)=8^3 = 17 (mod 33)
  - E(c)=2^3 = 8 (mod 33)
- 则c ＝ E（M）＝ 09 14 01 11 17 28＝ joblri

文本解密

- 收到密文后用**d=7，n=33**进行解密
- D(j)= 09^7 = 15 mod 33, 即明文 **p**
- D(o)= 14^7 = 20 mod 33, 即明文 **u**
- D(b)= 01^7 = 1 mod 33, 即明文 **b**
- D(l)= 11^7 = 11 mod 33, 即明文 **l**
- D(r)= 17^7 = 8 mod 33, 即明文 **i**
- D(i)= 08^7 = 2 mod 33, 即明文 **c**

RSA的安全性

RSA的安全性依赖于大整数分解的难度：

- 公开n=pq和e，但对p和q进行保密。
- 要从公开密钥n和e得到私人密钥d，只能通过分解n，先求得$\Phi(n)=(p-1)(q-1)$的值，然后才能得到e关于$\Phi(n)$的逆元$d=e^{-1} \bmod \Phi(n)$。
- 但通过分解大整数n得到p和q是一个困难问题。

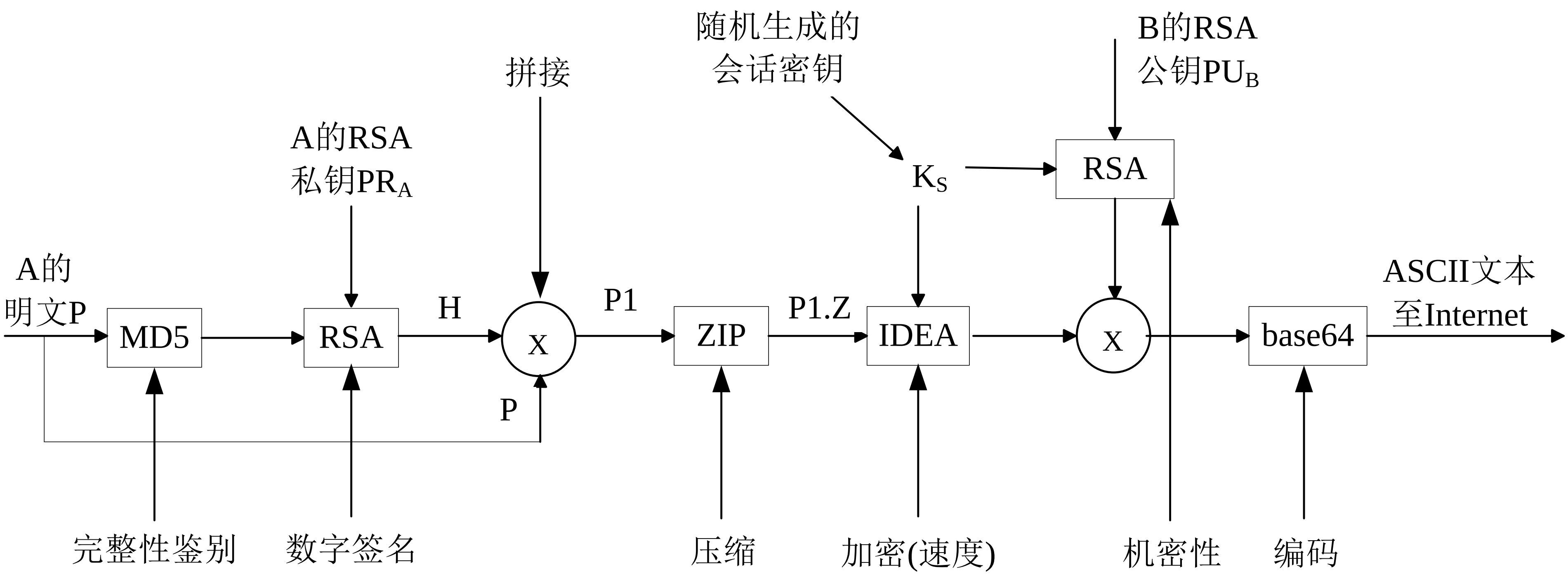
针对RSA可能存在的攻击方式

- **大整数分解**:129位十进制数已经通过分布式计算解开了。所以n应该不止129位。
- **时间攻击**：如果对硬件有充分的了解，就有可能根据加密的运行时间反推出私钥的内容。
- **针对密钥分配的攻击**:对RSA来说，分配公钥的过程非常重要，必须能够抵挡**中间人**（从中取代的）**攻击**。

假设A交给B一个公钥，并使其相信这是C的公钥，并且A可以截下C和B间的所有信息传递。那么A可以将自己的公钥给B，B以为这是C的公钥。A拦截所有B传递给C的消息，将该消息用自己的密钥解密，读取消息内容后，再将该消息用C的公钥加密后传给C。理论上说，C和B都不会发现A在偷听他们的消息。

今天人们一般用数字认证——证书授权（Certificate Authority）来防止这样的攻击。

PGP



数字指纹、数字身份证、数字签名、数字信封、数字证书

机密性、完整性、不可否认、发送方鉴别