1.5.11 RSA密码系统

- RSA是一种分组密码，加密密钥(或称公共密钥)为(n, e)，其中$n = pq$是一个由两个大素数，比如各有300位数字的p和q的乘积构成的模数，e是与$(p-1)(q-1)$互素的指数.
- 要生成这样的加密密钥，需要找到两个大素数．比如$n = pq$大约有600位数字，目前不可能在合理时间内被因子分解，因此没有单独的解密密钥不可能迅速解密.

1.5.11 RSA加密

- RSA中, 用公共密钥(n, e)**加密过程**：
  - 首先, 将明文消息M翻译成整数序列. 其中, 每个明文字母翻译成两位数, 如A翻译成00, B翻译成01,....J翻译成09.
  - 然后, 将这些两位数连接起来构成数字串.
  - 将这个数字串分为$2N$位数字等长的分组, 这里$2N$是一个大偶数使得$2N$位数字的整数2525...25不超过n. 其中, 必要时在明文消息最后填充无意义的X使得最后一组的大小和其他分组一样.
  - 到此, 明文消息M翻译成了一个整数序列$m_1, m_2,..., m_k$, 其中k为整数.
  - 然后, 将每个分组m_i转换成密文分组c_i, 其转换由函数$C = M^e \bmod n$实现.
  - 加密以后的消息依然是数的分组形式, 并发送给接受者.

1.5.11 RSA加密

- 例: 用RSA密码系统以及密钥(2537,13)加密消息“STOP”. $2537 = 43 \cdot 59$, p =43, q =59 是素数，并且 $\gcd(e,(p-1)(q-1)) = \gcd(13,42 \cdot 58) = 1$.
- 解:
  - 首先, 消息中每个字母翻译成两位数字:18 19 14 15
  - 因为2525 < 2537 < 252525, 所以划分为2N=4位等长的分组:1819 1415.
  - 对每个分组采用$C = M^{13}$ **mod** 2537来进行加密.
  - 因此 1819^{13} **mod** 2537= 2081, 1415^{13} **mod** 2537= 2182
  - 综上, 加密后的消息为2081 2182.

1.5.11 RSA解密

- RSA中, 用解密密钥d**解密**：
  - 因为$\gcd(e,(p-1)(q-1))=1$, 所以e模$(p-1)(q-1)$的逆d一定存在.
  - 当已知解密密钥d, RSA能够快速完成解密. 通过对每个分组用解密函数$M = C^d \bmod pq$, 这儿C是加密后的消息, p和q是$n=pq$中的两个大素数.

- RSA和一般的公钥密码系统一样, 只有知道解密密钥d才能解密消息. 当前不知道d的情况下, 因为两个大素数是无法在短时间内破译加密的消息.
- 例如按照现有的能力分解一个400位的整数需要上亿年的时间. 因此当p和q是200位的素数时, 就目前的水平而言, RSA是安全的.
- 随着因子分解能力的提高, 可能需要使用更大的素数.

1.5.11 RSA解密

- 例: 由上个例子中的密钥加密后的消息为0981 0461，求解密后的消息是多少?
- 解:
  - 加密密钥中n = 43· 59=2537, e=13. 那么d=937是13模42∗58的逆. 我们用937作为解密指数.
  - 要解密每个分组C, 需要计算$M = C^{937} \ \boldsymbol{mod}\ 2537$.
  - 因此, $0981^{937} \ \boldsymbol{mod}\ 2537 = 0704$, $0461^{937} \ \boldsymbol{mod}\ 2537 = 1115$.
  - 那么, 解密后的消息为0704 1115.
  - 对这个解密后的消息每两位翻译为一个字母:HELP.
  - 综上, 最终解密后的消息是HELP.

1.5.11 RSA解密

- 例:构造RSA公钥密码体系的密钥，令N=77
  - (1) 以d = 13为解密私钥, 求对应的加密公钥e.
  - (2) 求明文25对应的密文.
  - (3) 求密文15对应的明文.
- 解:
  - (1)N=77=7*11. 因此(p−1)(q −1)=60. e模(p−1)(q −1)的逆为d. 那么$de \equiv 1(mod\ 60)$, 可以得e=37.
  - (2) 要加密, $C = 25^{37}\ \boldsymbol{mod}\ 77$. 计算可得$C = 53$.
  - (3) 要解密, $M = 15^{13}\ \boldsymbol{mod}\ 77$. 计算可得$M = 64$.

1.5.12 密码协议: 密钥交换

- **密码协议**，两方或者多方为了达到一个特定的安全目标而进行的消息交换.

- **密钥交换**是一种在双方以往没有共享过任何信息的情况下可以用来在不安全的通信信道上交换密钥的协议. 其中**迪菲-赫尔曼密钥协商协议**(或Diff-Helmen key交换协议，DH协议，DHKE)，它是在1976年由惠特菲尔德.迪菲和马丁.赫尔曼两个人的名字命名的协议.
  - 其实，早在1974年，为英国GCHQ秘密工作的马尔科姆.威廉姆森已经发明了该协议，但因为保密的原因，直到1997年他的发现才公之于众.

1.5.12 密码协议: 密钥交换

- 接下来将DH协议描述为如下例子:
  - 假设A和B想要共享一个公共密钥. 两人同意使用素数p和p的一个原根a.
  - A选择一个秘密整数k_1, 然后将$a^{k_1} \ \boldsymbol{mod}\ p$ 发送给B.
  - B选择一个秘密整数k_2, 然后将$a^{k_2} \ \boldsymbol{mod}\ p$ 发送给A.
  - A计算$(a^{k_2})^{k_1} \ \boldsymbol{mod}\ p$.
  - B计算$(a^{k_1})^{k_2} \ \boldsymbol{mod}\ p$.
- 在协议的最后，A和B已经计算了共享的密钥，即 $(a^{k_2})^{k_1} \ \boldsymbol{mod}\ p = (a^{k_1})^{k_2} \ \boldsymbol{mod}\ p$.
- 想要找到密钥，那么需从p，a, $a^{k_1} \ \boldsymbol{mod}\ p$ 和$a^{k_2} \ \boldsymbol{mod}\ p$中计算出$k_1$, k_2. 这是求解离散对数的实例. 然而, 如果p和a足够大, 目前计算能力无法破解.

1.5.12 密码协议: 数字签名

- 密码学除了能够确保消息的保密性，还可以用来使得消息的接受方知道消息来自哪个该来自的人. 例如利用RSA密码系统对消息施加**数字签名**可以达到以上要求.

1.5.12 密码协议: 数字签名

- 假设A的RSA公钥是(n, e)，她的私钥是d．A用加密函数$E_{(n,e)}\ (x) = x^e\ \boldsymbol{mod}\ n$来加密明文消息$x$．A用解密函数$D_{(n,e)}\ (y) = y^d\ \boldsymbol{mod}\ n$来解密密文消息$y$.
- A想要发送消息M, 使得每个收到该消息的人都知道来自她.
  - 首先像RSA加密一样，将字母翻译成对应的数值，并将所得的串分割成分组m_1，m_2,…，m_k使得每个分组具有相同的大小，并且其大小满足$0 \leq m_i \leq n,\ i = 1,2, \dots, k$.
  - 然后，针对每个分组应用解密函数$D_{(n,e)}$将得到$D_{(n,e)}(m_i)$，$i = 1,2, \dots, k$. 将这个结果发送给所有预期的消息接收者.
  - 最后，消息接收者对每个分组应用A的加密函数$E_{(n,e)}$将得到$E_{(n,e)}(D_{(n,e)}(x)) = x$, 即结果为原始的明文信息M.

1.5.12 密码协议: 数字签名

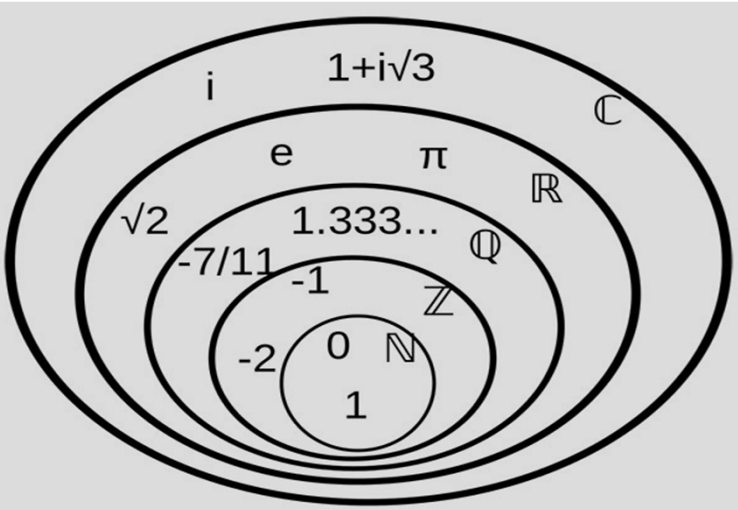
- 例: Alice的RSA公钥是(2537,13), $2537 = 43 \cdot 59$, 她的解密密钥是d=937. 如果她想要发送消息“MEET AT NOON”给她的朋友使得朋友们能够确信消息来自她, 她该如何发送?
- 解:
  - 首先, 将消息翻译成数字分组为1204 0419 0019 1314 1413.
  - 然后, 对每个分组应用解密函数$D_{(2537,13)}(x) = x^{937} \ \boldsymbol{mod}\ 2537$. 可得结果分别为 $1204^{937} \ \boldsymbol{mod}\ 2537 = 817$, $419^{937} \ \boldsymbol{mod}\ 2537 = 555$, $19^{937} \ \boldsymbol{mod}\ 2537 = 1310$, $1314^{937} \ \boldsymbol{mod}\ 2537 = 2173$, $1413^{937} \ \boldsymbol{mod}\ 2537 = 1026$.
  - 因此, 她将发送0817 0555 1310 2173 1026.
  - 当她的朋友收到该消息时, 针对每个分组应用她的加密函数$E_{(2537,13)}$就能获得原始消息的数字分组, 然后通过翻译可以得到最初的英文字母消息.

1.5.13 同态加密

- **同态加密**是一种特殊的加密方法. 允许对密文进行处理得到仍然是加密的结果. 即对密文直接进行处理, 跟对明文进行处理后再对处理结果加密得到的结果相同. 从抽象代数的角度讲, 保持了同态性.
  - 备注:有兴趣的同学自学同态加密.

第1.5节 密码学小结

- 欧拉函数$\Phi(n)$, 欧拉定理$a^{\Phi(n)} \equiv 1 \ (mod \ n)$
- 古典密码学: 移位密码; 仿射密码; 换位密码; 维吉利亚密码
- 私钥加密, 加密和解密密钥都需要保密的加密
- 公钥加密, 加密密钥公开, 解密密钥保密的加密
- RSA加密密钥$k = (n, e)$, 解密密钥d是e模$(p-1)(q-1)$的逆
- 密码协议: 迪菲-赫尔曼密钥协商协议; 数字签名



第1.6节 同余应用

Section 1.6: Applications of Congruences

知识要点

1. 散列函数
2. 伪随机数
3. 校验码

散列函数

- 定义: 一个**散列函数**h将内存地址$h(k)$分配给以k为键值的.
  - 最常用的散列函数是$h(k) = k \bmod m$, 其中m是可供使用的内存地址的数目.
  - 散列函数是满射的, 这样所有内存地址均可用.

- 例: 令$h(k) = k \bmod 111$. 找出分配给社会保障号分别为064212848, 037149212, 107405723的客户记录的内存地址.
- 解:
  - h(064212848) = 064212848 **mod** 111 = 14,
  - h(037149212) = 037149212 **mod** 111 = 65,
  - h(107405723) = 107405723 **mod** 111 = 14

散列函数

- 解(续):
  - 在以上结果中, 注意因为散列函数不是一对一的, 所以有可能多个记录被分配到同一个内存地址, 例如上面的064212848和107405723, 这时叫做**冲突**.
  - 消解冲突的一个办法就是使用散列函数分配已被占用地址后面的第一个未占用的地址, 107405723的分配的内存地址则变为15.

伪随机数

- 在计算机中随机选择的数经常会被用到. 目前已经设计了许多方法来产生具有随机选择性质的数, 但因为由系统方法产生的数并不真正是随机的, 所以被称为**伪随机数**.

- 我们可以利用线性同余法来产生伪随机数.
- 我们选择4个整数:模数m, 倍数a, 增量c, 种子x_0, 满足 $2 \leq a < m$, $0 \leq c < m$, $0 \leq x_0 < m$. 通过连续地应用下面的递归函数, 来产生随机数序列$\{x_n\}$, 满足对于所有的n, $0 \leq x_n < m$:

$$x_{n+1} = (ax_n + c) \bmod m$$

伪随机数

- 例: 利用线性同余法来生成伪随机数序列, 其中模m=9, 倍数 a=7, 增量c=4, 种子x_0=3.
- 解: 连续应用递归定义的函数x_{n+1} = (7x_n + 4) **mod** 9, 其中 x_0=3.
  - x_1 = 7x_0 + 4 **mod** 9 = 7·3 + 4 **mod** 9 = 25 **mod** 9 = 7,
  - x_2 = 7x_1 + 4 **mod** 9 = 7·7 + 4 **mod** 9 = 53 **mod** 9 = 8,
  - x_3 = 7x_2 + 4 **mod** 9 = 7·8 + 4 **mod** 9 = 60 **mod** 9 = 6,
  - x_4 = 7x_3 + 4 **mod** 9 = 7·6 + 4 **mod** 9 = 46 **mod** 9 = 1,
  - x_5 = 7x_4 + 4 **mod** 9 = 7·1 + 4 **mod** 9 = 11 **mod** 9 = 2,
  - x_6 = 7x_5 + 4 **mod** 9 = 7·2 + 4 **mod** 9 = 18 **mod** 9 = 0,
  - x_7 = 7x_6 + 4 **mod** 9 = 7·0 + 4 **mod** 9 = 4 **mod** 9 = 4,
  - x_8 = 7x_7 + 4 **mod** 9 = 7·4 + 4 **mod** 9 = 32 **mod** 9 = 5,

伪随机数

- 解(续):
  - $x_9 = 7x_8 + 4 \bmod 9 = 7 \cdot 5 + 4 \bmod 9 = 39 \bmod 9 = 3$.
  - 由于$x_9 = x_0$,而且每一项都只依赖于其前面的一项,所以产生序列3,7,8,6,1,2,0,4,5,3,7,8,6,1,2,0,4,5,3,...
  - 这个序列中包含9个不同的数, 然后重复.
- 以上伪随机数重复太快，因此大部分计算机生成伪随机数时使用增量$c = 0$. 这种的生成器称为**纯倍式生成器**. 例如$2^{31} - 1$为模, 7^5为倍数的纯倍式生成器就被广泛使用，它在重复之前会产生的$2^{31} - 1$个数.

校验码

- 同余可用于检查数字串中的错误.
- **奇偶校验位**：数字信息一般用位串表示，并划分为指定大小的块．每个块在存储之前，为末尾额外添加一位，称为奇偶校验位．位串$x_1x_2x_3\ldots x_n$的奇偶校验位x_{n+1}定义为$x_{n+1}=x_1+x_2+x_3+\ldots+x_n \mod 2$.
- 由此得出如果在这n位中有偶数个1位，则x_{n+1}=0；如果有奇数个1位，则x_{n+1}=1.
- 在检查阶段，如果奇偶校验位错了那我们就知道位串中有错误．但它也有局限性，可以检测到前面n位中奇数个错误，不能检查到偶数个错误.

校验码

- 例:接受到位串01100101和11010110，其中每个位串都以一个奇偶校验位结束, 这些位串是正确的吗?
- 解:首先将这些位串假定为正确的. 我们来检测它的奇偶校验位.
  - 第一串(01100101)的奇偶校验位应该为0+1+1+0+0+1+0 **mod** 2=1, 而传输过来的奇偶校验位也是1, 所以奇偶校验位是正确的.
  - 第二串(11010110)的奇偶校验位应该为1+1+0+1+0+1+1 **mod** 2=1, 而传输过来的奇偶校验位是0, 所以奇偶校验位不是正确的.
  - 因此我们可以得出结论，第一串可能是正确的(说可能是因为它有可能包含偶数个错误, 奇偶校验位检测不出来), 但第二串一定是错误的.

校验码:UPC

- 零售产品通常由其**通用产品代码标识**(Universal Product Code,UPC).
- UPC最常用的形式是12位十进制数字$x_1x_2x_3\ldots x_{11}x_{12}$，第一位数字标识产品种类($x_1$)，接着五位标识制造商($x_2x_3x_4x_5x_6$)，再五位标识特定产品($x_7x_8x_9x_{10}x_{11}$)，最后一位是校验码($x_{12}$). 校验码由同余式决定:

$$3x_1+x_2+3x_3+x_4+3x_5+x_6+3x_7+x_8+3x_9+x_{10}+3x_{11}+x_{12}\equiv 0 (mod 10)$$

校验码:UPC

- 例:假设UPC的前11位为79357343104, 求最后一位校验码是多少?
- 解:将校验码x_{12}代入上面的同余式$3\cdot7 + 9 + 3\cdot3 + 5 + 3\cdot7 + 3 + 3\cdot4 + 3 + 3\cdot1 + 0 + 3\cdot4 + x_{12} \equiv 0 \pmod{10}$, 简化后得$98 + x_{12} \equiv 0 \pmod{10}$. 由此$x_{12} \equiv 2 \pmod{10}$, 校验位为2.

- 例:04133102164 1是否是一个合法的UPC?
- 解:将这些数字代入同余式, 得$3\cdot0 + 4 + 3\cdot1 + 3 + 3\cdot3 + 1 + 3\cdot0 + 2 + 3\cdot1 + 6 + 3\cdot4 + 1 \equiv 0 \pmod{10}$, 但是$0 + 4 + 3 + 3 + 9 + 1 + 0 + 2 + 3 + 6 + 12 + 1 = 44 \equiv 4 \not\equiv 0 \pmod{10}$, 所以它不是一个合法的UPC.

校验码:ISBN

- 所有图书都由一个**国际标准书号**(International Standard Book Number, ISBN-10)标识.
- 一个由出版商指定的10位数代码$x_1x_2x_3\ldots x_{10}$ (备注，最新已经有ISBN-13, 这儿不讲这种更大量标识的国际标准书号)，它包含不同分组来表示语言、出版商、出版公司赋予图书的编号、最后一位校验码(可以是数字也可以是字母X,其中字母X表示整数10). 其中校验码需要满足:

$$x_{10} = \sum_{i=1}^{9} ix_i \ (\mathbf{mod}\ 11)$$

或者等价于$\sum_{i=1}^{10} ix_i \equiv 0\ (mod\ 11)$

校验码:ISBN

- 例:某书的ISBN-10的前9位为 007288008. 求校验码是多少?
- 解:将前9位代入同余式可得x_{10} ≡ 1·0 + 2·0 + 3·7 + 4·2 + 5·8 + 6·8 + 7· 0 + 8·0 + 9·8 (mod 11). 这意味着x_{10} ≡ 189 ≡ 2 (mod 11). 故校验码x_{10}= 2.

- 例:084930149X是否是合法的ISBN-10?
- 解:将10位一起代入第二个同余式可得1·0 + 2·8 + 3·4 + 4·9 + 5·3 + 6·0 + 7· 1 + 8·4 + 9·9 + 10·10 =0 + 16 + 12 + 36 + 15 + 0 + 7 + 32 + 81 + 100 = 299 ≡ 2 ≠ 0 (mod 11), 故它不是一个合法的 ISBN-10.

本章知识点回顾

- **a|b**:存在整数c使得$b = ac$
- **a和b模m同余**:$m|a - b$
- **模算术**:以一个整数m（m大于2）为模数所做的计算
- **素数**:大于1且恰只有1和它自身两个正因子的整数
- **合数**:大于1又不是素数的整数
- **gcd(a,b)**:能整除a和b的最大整数
- **互素整数**:满足$\gcd(a, b) = 1$的整数a和b
- **lcm(a,b)**:能被a和b整除的最小正整数
- $\boldsymbol{n = (a_k a_{k-1 \ldots} a_1 a_0)_b}$:$n$的$b$进制表示

本章知识点回顾

- **二进制表示**:整数以2为基数的表示
- **八进制表示**:整数以8为基数的表示
- **十六进制表示**:整数以16为基数的表示
- **贝祖系数**:$sa + tb = \gcd(a, b)$成立的整数s和t
- **a模m的逆**:$\bar{a}a \equiv 1(mod\ m)$成立的整数$\bar{a}$
- **线性同余方程**:$ax \equiv b(mod\ m)$,其中x为整数变量
- **以b为基数的伪素数**:$b^{n-1} \equiv 1(mod\ n)$成立的合数n
- **卡米切尔数**:合数n使得对所有满足$\gcd(b, n) = 1$的正整数b，n是以b为基数的伪素数

本章知识点回顾

- **素数p的原根**:Z_p中的整数r使得每个不能被p整除的整数模p同余r的一个幂次
- **以r为底a模p的离散对数**:满足0≤ e ≤ $p-1$,$r^e \equiv a \ (mod\ p)$的整数e
- **移位密码**:将明文p加密成（$p+k$）**mod** m的密码，其中k为整数
- **仿射密码**:将明文p加密成（$ap+k$）**mod** m的密码，其中a和k是整数, 且$\gcd(a, 26) = 1$
- **字符密码**:逐个字符地加密的密码
- **分组密码**:按等长字符分组加密的密码

本章知识点回顾

- **密码系统**:一个五元组(p, c, k, e, d), p是明文消息的集合, c是密文消息的集合, k是密钥的集合, e是加密函数的集合, d是解密函数的集合
- **私钥加密**:加密和解密密钥都需要保密的加密
- **公钥加密**:加密密钥公开, 解密密钥保密的加密
- **RSA**:密码系统,加密密钥$k = (n, e)$, 其中$n = pq$,其中p和q是大素数, e是正整数. 解密密钥d是e模$(p-1)(q-1)$的逆. $E_k(p) = p^e \bmod n$, $D_k(c) = c^d \bmod n$.