RSA密码体制

公钥密码学简介

目前为止我们研究的密码学模型中，加密采用的密钥与解密采用的密钥是相同的。我们称这类密码体制为对称密钥密码体制。

对称密码体制的缺点是加密方和解密方必须在传输密文前使用一个安全信道交换密钥，这在实际中很难实现。

为此引入公钥密码体制，公钥密码体制中加密密钥和解密密钥不同，并且已知加密密钥无法计算得到解密密钥。

因此实体可以对外公布自己的加密密钥，其他实体就可以利用该加密密钥加密消息。由于其他实体没有解密密钥，因此无法解密密文，因此可以保证消息的私密性。

公钥密码学简介

公钥密码学简介

1970年，James Ellis在一篇题为“非秘密加密的可能性”的论文中提出了公钥密码学的思想，但是这篇论文没有在公开文献中发表。

1973年，Clifford Cocks在“关于非秘密加密的注释”论文中描述了一个本质上与RSA密码体制相同的公钥密码体制，这篇论文也没有在公开文献中发表。

1976年，Diffie和Hellman在一篇题为“密码学的新方向”中公开提出公钥密码体制的思想。

1977年，Rivest,Shamir和Adleman发明了著名的RSA密码体制。

公钥密码学简介

Merkle_hellman_diffie

所需的加密方案

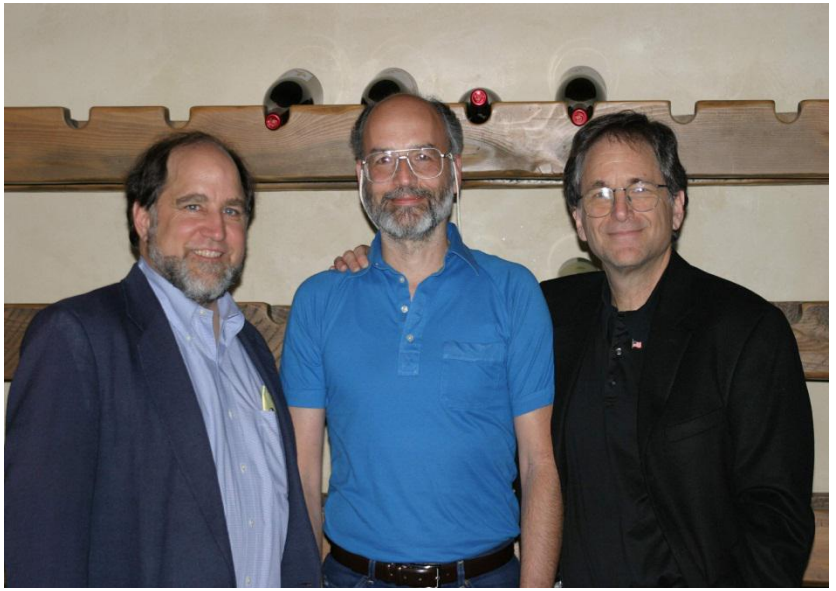
公钥密码体制可以抽象为一种陷门单向函数。

已知明文x和加密密钥k很容易计算出密文y，而且已知密文y和公钥k很难计算出明文x，这与单向函数的定义类似。

而且已知密文y和解密密钥d，可以很容易算出明文x，解密密钥d是单向函数的陷门。

公钥密码学简介

Shamir, Rivest, Adleman



Rivest, Shamir, Adleman in 2003

思路

单向函数的一个例子：
假设n为两个大素数p和q的乘积，b为一个正整数。

$$f : Z_n \rightarrow Z_n$$

$$f(x) = x^b \bmod n$$

思路

$$f : Z_n \rightarrow Z_n$$

$$f(x) = x^b \bmod n$$

$b = 8$

$n = 11$

$f(x) = 3$

$x = ?$

$b = 9$

$n = 13$

$f(x) = 5$

$x = ?$

- RSA加密方案
  - 数论知识
    - 数的逆
    - 求逆
    - 群

数论知识

欧几里得算法 Euclidean Algorithm(a, b)

$r_0 \leftarrow a$

$r_1 \leftarrow b$

$m \leftarrow 1$

$while\ r_m \neq 0$

$$do\begin{cases} q_m \leftarrow \left\lfloor \dfrac{r_{m-1}}{r_m} \right\rfloor \\ r_{m+1} \leftarrow r_{m-1} - q_m r_m \\ m \leftarrow m+1 \end{cases}$$

$m \leftarrow m-1$

$return(q_1, \ldots, q_m; r_m)$

$comment: r_m = \gcd(a, b)$

在该算法中

$$\gcd(r_0, r_1) = \gcd(r_1, r_2) = \ldots = \gcd(r_{m-1}, r_m) = r_m$$

数论知识

假定按下面构造定义了两个数列$s_0, s_1, \ldots, s_m$和$t_0, t_1, \ldots, t_m$

$$s_j = \begin{cases} 1 & j = 0 \\ 0 & j = 1 \\ s_{j-2} - q_{j-1}s_{j-1} & j \geq 2 \end{cases} \qquad t_j = \begin{cases} 0 & j = 0 \\ 1 & j = 1 \\ t_{j-2} - q_{j-1}s_{j-1} & j \geq 2 \end{cases}$$

定理5.1

对于$0 \leq j \leq m$，有$r_j = s_j r_0 + t_j r_1$，其中r_j按欧几里得算法定义，s_j, t_j按上述定义

数论知识

利用数学归纳法进行证明：
对于 $j = 0$ 和 $j = 1$，命题显然成立。
假设命题对于 $j = i - 1$ 和 $j = i - 2$ 成立，其中 $i \geq 2$

由归纳假定，则有：
$r_{i-2} = s_{i-2}r_0 + t_{i-2}r_1$ 和 $r_{i-1} = s_{i-1}r_0 + t_{i-1}r_1$

此时 $r_i = r_{i-2} - q_{i-1}r_{i-1}$

$$= s_{i-2}r_0 + t_{i-2}r_1 - q_{i-1}\left(s_{i-1}r_0 + t_{i-1}r_1\right)$$

$$= \left(s_{i-2} - q_{i-1}s_{i-1}\right)r_0 + \left(t_{i-2} - q_{i-1}t_{i-1}\right)r_1$$

$$= s_i r_0 + t_i r_1$$

数论知识

扩展欧几里得算法 Extended Euclidean Algorithm(a, b)

$$
\begin{aligned}
&a_0 \leftarrow a \\
&b_0 \leftarrow b \\
&t_0 \leftarrow 0 \\
&t \leftarrow 1 \\
&s_0 \leftarrow 1 \\
&s \leftarrow 0 \\
&q \leftarrow \left\lfloor \frac{a_0}{b_0} \right\rfloor \\
&r \leftarrow a_0 - qb_0
\end{aligned}
$$

$$
\begin{aligned}
&while\ r > 0 \\
&do \begin{cases}
temp \leftarrow t_0 - qt \\
t_0 \leftarrow t \\
t \leftarrow temp \\
temp \leftarrow s_0 - qs \\
s_0 \leftarrow s \\
s \leftarrow temp \\
a_0 \leftarrow b_0 \\
b_0 \leftarrow r
\end{cases}
\end{aligned}
$$

$$
\begin{cases}
q \leftarrow \left\lfloor \frac{a_0}{b_0} \right\rfloor \\
r \leftarrow a_0 - qb_0
\end{cases}
$$

$$
\begin{aligned}
&r \leftarrow b_0 \\
&return(r, s, t)
\end{aligned}
$$

$$
\begin{aligned}
&comment: \\
&r = \gcd(a, b) \\
&sa + tb = r
\end{aligned}
$$

数论知识

推论5.2 假定$\gcd(r_0, r_1) = 1$ 那么$r_1^{-1} \bmod r_0 = t_m \bmod r_0$

证明：由定理5.1，有

$$1 = \gcd(r_0, r_1) = s_m r_0 + t_m r_1$$

两边模 r_0 约化等式，得 $t_m r_1 \equiv 1 (\bmod\ r_0)$

例：计算 $28^{-1} \bmod 75$

数论知识

例：计算 $28^{-1} \bmod 75$

i	r_j	q_j	s_j	t_j
0	75		1	0
1	28	2	0	1
2	19	1	1	-2
3	9	2	-1	3
4	1		3	-8

数论知识

例：计算 $28^{-1} \bmod 75$

因此，我们发现 $3*75-8*28=1$

应用推论5.2，可得到

$$28^{-1} \bmod 75 = -8 \bmod 75 = 67$$

群

定义 设S是一个非空集合，函数f：$S^n \to S$称为S上的一个n元运算，n称为运算f的阶数。

定义 一个非空集合S连同若干个定义在S上的运算f_1、f_2、…、f_m组成的系统称为代数系统或代数结构，记作$<S, f_1, f_2, \ldots, f_m>$。

由定义可知，一个代数系统需满足以下3个条件：

(1)有一个非空集合S。

(2)有建立在S上的一些运算。

(3)这些运算在S上是封闭的。

群

例 设$Z_m=\{[0]，[1]，\ldots，[m-1]\}$是模$m$同余关系所有剩余类组成的集合，在$Z_m$上定义运算$+_m$和$\times_m$为：对任意的$[a]$、$[b]\in Z_m$，$[a]+_m[b]=[a+b]$，$[a]\times_m[b]=[a\times b]$，则$+_m$和$\times_m$是$Z_m$上的二元运算。

群

代数系统的性质

1.交换律

定义 设<S，*>是一个代数系统，若对任意的x、y∈S有x*y＝y*x，则称二元运算*是可交换的，或说*满足交换律。

2.结合律

定义 设<S，*>是一个代数系统，若对任意的x、y、z∈S有(x*y)*z＝x*(y*z)，则称二元运算*是可结合的，或说*满足结合律。

代数系统的性质

3. 分配律

设代数系统$<S，*，\odot>$，对任意x、y、$z \in S$，若$x*(y\odot z)=(x*y)\odot(x*z)$，则称*对$\odot$满足左分配律；若$(y\odot z)*x=(y*x)\odot(z*x)$，则称*对$\odot$满足右分配律；若两者都满足，则称*对$\odot$满足分配律。

群

定义 设代数系统<S，*>，且存在e_l、e_r、$e \in S$，对任意$x \in S$，若$e_l * x = x$，则称e_l是S中关于*的一个左单位元；若$x * e_r = x$，则称e_r是S中关于*的一个右单位元。若e关于*既是左单位元又是右单位元，则称e为S中关于*的单位元。

例：在<Z_m，$+_m$，$\times_m$>中，运算$+_m$的单位元是[0]，运算$\times_m$的单位元是[1]。

群

设<S，*>是一个代数系统，e是S中关于*的**单位**元。对于$x \in S$，若存在$y_l \in S$使得$y_l * x = e$，则称y_l是x的左逆元；对于$x \in S$，若存在$y_r \in S$使得$x * y_r = e$，则称y_r是x的右逆元。对于$x \in S$，若存在$y \in S$既是x的左逆元又是x的右逆元，则称y为x的逆元，通常记为x^{-1}。

群

定义 设代数系统<S，*>，且存在θ_l、θ_r、$\theta \in S$，对任意的$x \in S$，若$\theta_l * x = \theta_l$，则称θ_l是S中关于*的一个左零元；若$x * \theta_r = \theta_r$，则称θ_r是S中关于*的一个右零元。若θ关于*既是左零元又是右零元，则称θ为S中关于*的零元。

例 在<Z_m，$\times_m$>中，[0]是零元，因为对任意的[a]$\in Z_m$，有[a]$\times_m$[0]=[0]$\times_m$[a]=[0]。

群

定义 群〈G，$*$〉是一代数系统, 其中二元运算*满足以下3条

(1) 对所有的$a, b, c \in G$

$$a * (b * c) = (a * b) * c$$

(2) 存在一个元素e, 对任意元素$a \in G$, 有

$$a * e = e * a = a$$

(3) 对每一$a \in G$, 存在一个元素a^{-1}, 使

$$a^{-1} * a = a * a^{-1} = e$$

简单地说, 群是具有一个可结合运算, 存在单位元, 每个元素存在逆元的代数系统。

群

如果G是有限集合, 则称〈G, *〉是有限群; 如果G是无限集合, 则称〈G, *〉是无限群。有限群G的基数$|G|$称为群的**阶数**。

群中的运算 * 一般称为乘法。 如果 * 是一个可交换运算, 那么群〈G , * 〉就称为**可交换群**, 或称**阿贝尔群**。在可交换群中, 若运算符*改用+, 则称为**加法群**, 此时逆元a^{-1}写成$-a$。

群

定理 如果〈G，$*$〉是一个群, 则对于任何a、$b \in G$,

(a) 存在一个唯一的元素x, 使得$a * x=b$。 ■

(b) 存在一个唯一的元素y, 使得$y * a=b$。 ■

证 (a) 至少有一个x满足$a * x=b$, 即$x=a^{-1} * b$, 因为

$$a * (a^{-1} * b)=(a * a^{-1}) * b= e * b=b$$

如果x是G中满足$a * x=b$的任意元素, 则

$$x=e * x=(a^{-1} * a) * x = a^{-1} * (a * x) = a^{-1} * b$$

所以, $x=a^{-1} * b$是满足$a * x=b$的唯一元素。

群

定义群〈$G, *$〉的任意元素a的幂。 如果$n \in N$, 则

$$a^0 = e$$

$$a^{n+1} = a^n * a$$

$$a^{-n} = (a^{-1})^n$$

对任意m、$k \in I$，a^m，a^k都是有意义的，另外群中结合律成立，不难证明以下指数定律成立。

$$a^m * a^k = a^{m+k} \qquad (m、k \in I)$$

$$(a^m)^k = a^{mk} \qquad (m、k \in I)$$

群

定义 设〈G，*〉是一个群, 且$a \in G$, 如果存在正整数n使$a^n=e$, 则称元素的阶是有限的, 最小的正整数n称为元素a的阶。如果不存在这样的正整数n, 则称元素a具有无限阶。 ■

显然, 群的单位元e的阶是1。

定理 群中的任一元素和它的逆元具有同样的阶。

证 设$a \in G$具有有限阶n, 即$a^n=e$, 因此♡♡

$$(a^{-1})^n = a^{-1 \cdot n} = (a^n)^{-1} = e^{-1} = e$$

如果(a^{-1})的阶是m, 则$m \le n$。 另一方面♡♡

$$a^m = \left[(a^{-1})^m\right]^{-1} = e^{-1} = e$$

因而$n \le m$, 故$m=n$。

群

定理 在有限群〈G, $*$〉中, 每一个元素具有一有限阶, 且阶数至多是$|G|$。 ■

证 设a是〈G, $*$〉中任一元素。在序列$a, a^2, a^3, ..., a^{|G|+1}$中至少有两元素是相等的。不妨设$a^r=a^s$, 这里$1 \le s < r \le |G|+1$。 因为

$$e=a^0=a^{r-r} = a^r * a^{-r} = a^r * a^{-s} = a^{r-s}$$ ■

所以, a的阶数至多是$r-s \le |G|$。 证毕。

群

定义 设〈G, *〉是一个群, I是整数集合。如果存在一个元素$g \in G$, 对于每一个元素$a \in G$都有一个相应的$i \in I$, 能把a表示成g^i形式, 则称〈G , *〉是一个循环群。或说循环群是由g生成的, g是〈G, *〉的**生成元**。

群

定理 设〈G, *〉是由$g \in G$生成的有限循环群, 如果$|G|=n$, 则$g^n=e$,♡

$$G = \{g , g^2, g^3, ..., g^n = e\}$$

且n是使$g^n=e$的最小正整数。 ■

证 ■

(1) 假定有正整数$m<n$使$g^m=e$, 则对G中任一元素g^k, 设$k=mq+r$,

$0 \le r < m$, 于是♡♡

$$g^k=g^{mq+r} = (g^m)^q * g^r = g^r$$

这意味着G中每一元素都可写成g^r形式, 但$r<m$, 所以G中至多有m个不同元素, 这与$|G|=n$矛盾, 所以$g^m=e$而$m<n$是不可能的。

(2) $\{g, g^2, g^3, \ldots, g^n\}$中的元素全不相同。若不然有$g^i=g^j$, 不妨设$i<j$, 于是$g^{j-i}=e$。但$j-i<n$。 所以这是不可能的。 ■

由于〈G , *〉是群, 其中必有单位元, 由(2)得$G=\{g, g^2, g^3, \ldots, g^n\}$, 因此, 由(1)得$g^n=e$。 证毕。

群

$Z_n^* = \{[1],[p_1],...[p_t]\}$，其中$\gcd(p_i, n) = 1, 1 \le i \le t$

其中t为Z_n^*的阶，记为$\Phi(n)$。

若p为素数，则$Z_p^* = \{1, 2, ..., p-1\}$

$Z_{13}^* = \{[1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12]\}$

$(Z_n^*, \times_n)$是阶为$\Phi(n)$的循环群，简写为Z_n^*。

[2]是生成元。

[4]不是生成元。

群

$2^1 \bmod 13=2$ $2^5 \bmod 13=6$ $2^9 \bmod 13=5$

$2^2 \bmod 13=4$ $2^6 \bmod 13=12$ $2^{10} \bmod 13=10$

$2^3 \bmod 13=8$ $2^7 \bmod 13=11$ $2^{10} \bmod 13=7$

$2^4 \bmod 13=3$ $2^8 \bmod 13=9$ $2^{10} \bmod 13=1$

群

例子 假定 $p = 13$ ，可以验证2是一个模13的生成元

$$2^{12} \bmod 13 = 1 \qquad 2^{6} \bmod 13 = 12$$

$$2^{1} \bmod 13 = 2 \qquad 2^{7} \bmod 13 = 11$$

$$2^{2} \bmod 13 = 4 \qquad 2^{8} \bmod 13 = 9$$

$$2^{3} \bmod 13 = 8 \qquad 2^{9} \bmod 13 = 5$$

$$2^{4} \bmod 13 = 3 \qquad 2^{10} \bmod 13 = 10$$

$$2^{5} \bmod 13 = 6 \qquad 2^{11} \bmod 13 = 7$$

群

定理　　如果$p > 2$是一个素数，且$\alpha \in Z_p^*$，那么α是一个模 p 的生成元，当且仅当 $\alpha^{(p-1)/q} \pmod p$ 恒不等于 1 对于所有满足$q \mid (p-1)$ 的素数 q 都成立。

证明：如果α 是一个模p的本原元素，那么对于所有的 $1 \le i \le p-2$，都有α^i恒不等于$1 \pmod p$，所以结果成立。

群

反过来，假定 $\alpha \in Z_p^*$ 不是模 p 的生成元。则令 d 为 α 的阶，那么根据Lagrange定理，有 $d \mid (p-1)$。因为 α 不是本原的，所以 $d < p-1$。那么 $(p-1)/d$ 是一个大于1的整数令 q 为 $(p-1)/d$ 的素因子，那么 d 是 $(p-1)/q$ 的一个因子。由于 $\alpha^d \equiv 1 (\mathrm{mod}\ q)$ 且 $d \mid (p-1)/q$，于是有

$$\alpha^{(p-1)/q} \equiv 1 (\mathrm{mod}\ p)$$

群

12的因式分解为 $12 = 2^2 * 3$，因此，在前面的例子中，我们可以通过验证 $2^6 \neq 1(\mathrm{mod}13)$ 以及 $2^4 \neq 1(\mathrm{mod}13)$ 来验证2是一个模13的生成元。

群

$2^1 \bmod 10=2$　$2^4 \bmod 10=6$

$2^2 \bmod 10=4$　$2^5 \bmod 10=2$

$2^3 \bmod 10=8$

练习

- 元素？
- 生成元？

$$Z_{10}^{*} \quad Z_{13}^{*}$$

群

定理（Lagrange）假定 G 是一个阶为$\varphi(n)$的乘法群，且 $g \in G$，那么g的阶整除$\varphi(n)$。

推论 如果$b \in Z_n^*$，那么 $b^{\phi(n)} \equiv 1 \pmod{n}$

推论 (Fermat)假定 p 是一个素数，且$b \in Z_p$，那么 $b^p \equiv b \pmod{p}$

RSA密码体制

RSA密码体制描述如下：

设$N = p * q$，其中 p 和 q 为素数。设 $P = C = Z_n$

且定义 $K = \{(n, p, q, a, b) : ab \equiv 1(\mathrm{mod}\ \phi(n))\}$

对于 $K = (n, p, q, a, b)$，定义

$e_K(x) = x^b \bmod n$ 和 $d_K(y) = y^a \bmod n$

$(x, y \in Z_n)$

公钥 n, b；私钥 p, q, a

RSA密码体制

- 1. RSA是否正确？
- 2. RSA是否能够在多项式时间加密和解密？
- 3. RSA是否安全？

RSA密码体制

由于 $ab \equiv 1(\bmod \phi(n))$，所以有 $ab \equiv t\phi(n)+1$

对于某个整数 $t \geq 1$，假定 $x \in Z_n^*$，那么就有

$$\left(x^{b}\right)^{a} \equiv x^{t\phi(n)+1}\left(\bmod \quad n\right)$$

$$\equiv \left(x^{\phi(n)}\right)^{t} x\left(\bmod \quad n\right)$$

$$\equiv 1^{t}\, x\left(\bmod \quad n\right)$$

$$\equiv x\left(\bmod \quad n\right)$$

RSA密码体制

$x \in Z_n / Z_n^*$也满足 $\left(x^{b}\right)^{a} \equiv x$

$$x = p^{e_p} \cdot q^{e_q} \cdot \prod x_i^{e_i}, \left(x_i \in Z_N^*\right)$$

$$\left(x^a\right)^b$$

$$= \left(p^{e_p} \cdot q^{e_q} \cdot \prod x_i^{e_i}\right)^{ab}$$

$$= \left(p^{ab}\right)^{e_p} \left(q^{ab}\right)^{e_q} \prod \left(x_i^{ab}\right)^{e_i}$$

$$= \left(p^{ab}\right)^{e_p} \left(q^{ab}\right)^{e_q} \prod x_i^{e_i}$$

$$p^{ab} = p \bmod N ? \quad q^{ab} = q \bmod N ?$$

$$p^{ab} = p \bmod N ?$$

$$ab = k\varphi(n) + 1 = k(p-1)(q-1) + 1$$

$$\left(p^b\right)^a \equiv p^{k(p-1)(q-1)+1} \left(\bmod p\right) \equiv 0 \left(\bmod p\right)$$

$$\left(p^b\right)^a \equiv \left(p^{q-1}\right)^{k(p-1)} \left(\bmod q\right) \cdot p \left(\bmod q\right)$$

$$\equiv (1)^{k(p-1)} \left(\bmod q\right) \cdot p \left(\bmod q\right) \equiv p \left(\bmod q\right)$$

$$\left(p^b\right)^a \equiv 0 \left(\bmod p\right) \quad \left(p^b\right)^a \equiv p \left(\bmod q\right)$$

$$\left(p^b\right)^a \equiv p \left(\bmod N\right), k \in \{0,1,2,\ldots\}$$

RSA密码体制

下面描述一个RSA密码体制的小例子

假定Bob选取 p = 101， q = 113, 那么 n = 11413，

$$\phi(n) = 100 * 112 = 11200$$

由于 $11200 = 2^6 * 5^2 * 7$，所以可以选择一个整数b, 当且仅

当b不能被2，5或7整除。假定Bob选取 b = 3533。那么

$$b^{-1} \bmod 11200 = 6597$$

Bob在一个目录中发布 n = 11413， b = 3533

RSA密码体制

假定Alice想加密明文9726并发送给Bob。她将计算

$$9726^{3533} \bmod 11413 = 5761$$

然后把密文5761通过信道发出。Bob在收到密文5761后，计算

$$5761^{6597} \bmod 11413 = 9726$$

RSA密码体制的安全性是基于相信加密函数 $e_K(x) = x^b \bmod n$ 是一个单向函数这一事实，所以，对于一个敌手来说试图解密密文将是计算上不可行的。

RSA密码体制

RSA参数生成算法

1. 生成两个大素数，p 和 q，$p \neq q$

2. $n \leftarrow pq$，且 $\phi(n) \leftarrow (p-1)(q-1)$

3. 选择一个随机数 $b(1 < b < \phi(n))$，使得

$$\gcd(b, \phi(n)) = 1$$

4. $a \leftarrow b^{-1} \bmod \phi(n)$

5. 公钥为 (n, b)；私钥为 (p, q, a)

RSA密码体制

对RSA密码体制的一个明显攻击就是密码分析者试图分解 n

如果敌手可以做到这点，那么就可以很简单的计算出

$$\phi(n) = (p-1)(q-1)$$

然后敌手就可以和Bob一样地利用 b 计算出解密密钥

$$a = b^{-1} \bmod \phi(n)$$

如果RSA密码体制要成为安全的，那么要求 n=pq 必须足够大，

使得分解它是计算上不可行的。

RSA密码体制

RSA加密的效率问题。

假定 x 和 y 分别是 k 位和 l 位二进制表示的正整数；即

$$k = \lfloor lbx \rfloor + 1 \qquad l = \lfloor lby \rfloor + 1$$

假定 $k \geq l$ ，容易看到对 x 和 y 执行各种运算所需的

时间的上界估计：

计算 $x + y$ 的时间复杂度为 $O(k)$

计算 $x - y$ 的时间复杂度为 $O(k)$

RSA密码体制

计算 $x \cdot y$ 的时间复杂度为 $O(kl)$

计算 $\gcd(x, y)$ 的时间复杂度为 $O(k^3)$

Euclidean算法的迭代次数为 $O(k)$

每次迭代执行一次除需要时间 $O(k^2)$

RSA密码体制

计算形如 $x^c \bmod n$ 的函数：

在RSA密码体制中，加密和解密显然都是这类模指数运算。

计算 $x^c \bmod n$ 可以通过 $c-1$ 次模乘来实现，但是相对于 k 这是指数阶大的。

下面介绍平方—乘算法，该算法在计算上述模指数运算时可以运行在 $\mathrm{O}(lk^2)$ 时间。

RSA密码体制

计算形如 $x^c \bmod n$ 的函数：

首先将指数 c 用二进制表示，即 $c = \sum_{i=0}^{l-1} c_i 2^i, c_i = 0/1$

平方-乘算法 Square-and-Multiply(x,c,n)：

$z \leftarrow 1$

$for\ i \leftarrow l-1 \quad downto \quad 0$

$$do \begin{cases} z \leftarrow z^2 \bmod n \\ if \quad c_i = 1 \\ then \quad z \leftarrow (z \times x) \bmod n \end{cases}$$

$return(z)$

在该算法中，模乘的次数等于 c 的二进制中 1 的次数

因此模乘的执行次数至少为 l，最多为 $2l$

RSA密码体制

例：令 $n = 11413$，公开加密指数 $b = 3533$。Alice利用平方-乘算法，通过计算 $9726^{3533} \bmod 11413$ 来加密明文9726过程如下：

i	b_i	z
11	1	$1^2 \times 9726 = 9726$
10	1	$9726^2 \times 9726 = 2659$
9	0	$2659^2 = 5634$
8	1	$5634^2 \times 9726 = 9167$
7	1	$9167^2 \times 9726 = 4958$

RSA密码体制

例：通过计算 $9726^{3533} \bmod 11413$ 来加密明文 9726

i	b_i	z
6	1	$4958^2 \times 9726 = 7783$
5	0	$7783^2 = 6298$
4	0	$6298^2 = 4629$
3	1	$4629^2 \times 9726 = 10185$
2	1	$10185^2 \times 9726 = 105$
1	0	$105^2 = 11025$
0	1	$11025^2 \times 9726 = 5761$

因此，密文是5761

RSA密码体制

到目前为止，我们已经讨论了RSA的加密和解密运算。

其中RSA参数生成算法中的第一步，构造素数 p 和 q 的方法将在下一节讨论；

第二步是直接的，可以在时间 $O(\mathrm{k}^3)$ 内完成；

第三步和第四步，时间复杂度为 $O(\mathrm{k}^3)$