CN43-1258/TP
ISSN 1007-130X
计算机工程与科学
COMPUTER ENGINEERING & SCIENCE
2003 年第 25 卷 第 5 期
Vol.25,No.5,2003

文章编号:1007-130X(2003)05-0013-04

REESSE 1 公开密钥密码体制*
The REESSE 1 Public Key Cryptosystem

苏盛辉
SU Sheng-hui
(北京石油化工学院信息管理系,北京 102617)
(Department of Information Management, Institute of Petrochemical Technology, Beijing 102617, China)

摘 要:本文给出了互素序列的定义和杠杆函数的概念,介绍了 REESSE 1 公开密钥密码体制及其密钥生成、加密、解密、数字签名和身分验证五个算法。文章对加密和解密算法进行了有关推导和证明,对 REESSE 1 公钥体制的安全性进行了初步分析。另外,作者还给出了一个用于公钥密码体制中求模逆元的新递归算法。

Abstract: The paper gives the definition of relatively prime sequence and the concept of lever function, and expounds the REESSE 1 public key cryptosystem that includes five algorithms for key, encryption, decryption, digital signature and identity validation. The author makes derivations and demonstrations of the encryption and decryption algorithms, and makes a primary analysis of the security of the cryptosystem. Furthermore, a new recursive algorithm is presented, which is used to get a modulus inverse in a public key cryptosystem.

关键词:互素序列;杠杆函数;公开密钥;加密;签名

Key words: relatively prime sequence; lever function; public key; encryption; signature

中图分类号:TP309 **文献标识码**:A

1 引言

公开密钥密码体制(简称为公钥密码体制或公钥体制)是信息安全和公钥基础设施(PKI)的核心技术所在,因此对它的研究显得非常重要和必要。本文要介绍的 REESSE 1 公钥密码体制不同于美国的 RSA、ElGamal 或 MH 背包等体制[1],而是一种建立在互素序列基础上的新体制,可以用来加密、解密、数字签名和身分验证。REESSE 1 公钥体制虽然用到了陷门函数,但主要是引入了具有不确定性的杠杆函数来保障其安全性,并且在运算速度方面也显示出了优越性。此外,作者设计的用于求模逆元或解一元线性同余方程的递归算法展示了递归算法在密码学领域的应用。

2 REESSE 1 的数学基础

2.1 互素序列的定义与性质

定义 如果 $A_1, A_2, \cdots, A_n$ 为 n 个互不相同、两两互素(即最大公约数等于 1)、大于 1 的正整数,则称这样的正整数序列为互素序列,记为 $\{A_1, A_2, \cdots, A_n\}$,简记为 $\{A_i\}$。

性质 对于任意的正整数 $m(1 \leqslant m \leqslant n)$,从互素序列 $\{A_i\}$ 中任选 m 个项组成子序列 $\{A_{x_1}, A_{x_2}, \cdots, A_{x_m}\}$,则子序列的连乘积 $G_P = A_{x_1} * A_{x_2} * \cdots * A_{x_m}$ 是唯一确定的,即 G_P 与子序列 $\{A_{x_1}, A_{x_2},$

* 收稿日期:2003-01-18;修订日期:2003-05-09
作者简介:苏盛辉(1964-),男,湖南冷水江人,硕士,副教授,研究方向为信息安全、数据库系统和电子商务。
通讯地址:100037 北京市海淀区甘家口 24 号楼 1508 室;Tel:(010)69244752-245;E-mail:zhuohui@publica.bj.cninfo.net
Address: Room 1508, Building 24, Ganjiakou, Haidian District, Beijing 100037, P.R.China

$\cdots, A_{x_m}\}$一一对应("*"代表乘号，G_P 称为互素序列乘积)。

下面用反证法证明上述结论。由于序列 $\{A_1, A_2, \cdots, A_n\}$ 两两互素，因此任取 $A_j、A_k \in \{A_1, A_2, \cdots, A_n\}$，必有最大公约数 $gcd(A_j, A_k) = 1$，即 A_j 与 A_k 没有相同的素因子。这说明，互素序列的每一项所包含的素因子均不属于其它项。

假设 G_P 由两个不同子序列 $\{A_{x_1}, A_{x_2}, \cdots, A_{x_m}\}$ 和 $\{A_{y_1}, A_{y_2}, \cdots, A_{y_k}\}$ 得到，即

$$G_P = A_{x_1} * A_{x_2} * \cdots * A_{x_m} = A_{y_1} * A_{y_2} * \cdots * A_{y_k}$$

由于两个子序列不相等，因此必存在某项 A_q 不同时属于两个子序列。

不妨令 A_q 属于 $\{A_{x_1}, A_{x_2}, \cdots, A_{x_m}\}$，而不属于 $\{A_{y_1}, A_{y_2}, \cdots, A_{y_k}\}$。

根据算术基本定理[2]，必存在一素数 p 是 A_q 的因子。

因为前面已经阐明互素序列的每一项所包含的素因子均不属于其它项，所以素数 p 必是 $A_{x_1} * A_{x_2} * \cdots * A_{x_m}$ 的因子，而不是 $A_{y_1} * A_{y_2} * \cdots * A_{y_k}$ 的因子。这就说明，整数 G_P 可以有两种不同的素数表示法，与算术基本定理矛盾。

故 G_P 与 $\{A_{x_1}, A_{x_2}, \cdots, A_{x_m}\}$ 一一对应。

2.2 互素序列乘积与二进制数之间的可逆转换

(1)从二进制数得到互素序列乘积。

设 $\{A_i\}$ 为 n 项互素序列，$b_1 b_2 \cdots b_n$ 为 n 位二进制数，因此，对应的互素序列乘积为 $G_P = \prod A_i^{b_i}$。其中，i 从 1 至 n，b_i 表示幂次，则 G_P 唯一代表了二进制数 $b_1 b_2 \cdots b_n$。

(2)从互素序列乘积得到二进制数。

①令 $b_1 b_2 \cdots b_n$ 各位皆为 0，$i = 1$；

②如果 G_P / A_i 为整数，则 $b_i = 1$ 且 $G_P = G_P / A_i$；

③$i = i + 1$，如果 $i \leqslant n$ 且 $G_P \neq 1$，则转至②，否则，结束。

最后，得到原二进制数 $b_1 b_2 \cdots b_n$，由互素序列的性质知，这种转换是正确的。

2.3 互素序列与非互素序列之间的可逆转换

找到一个正素数 M，使得 $M > \prod A_i$，且 $M - 1$ 含 $n + 4$ 范围内的所有素数因子。选取一个正整数 W，使 $W < M$。

(1)从互素序列得到非互素序列。

对于互素序列 $\{A_i\}$，令 $C_i = (A_i * W^{f(i)}) \bmod M, i = 1, 2, \cdots, n$。其中，$f(i)$ 表示幂次，为随机正整数，且 $5 \leqslant f(i) \leqslant n + 4$，两两不同。请注意，$f(i)$ 的值域与置换函数不一样，mod 代表模运算。从近世代数知，$(Z_M^*, *)$ 构成交换群[3]。

$f(i)$ 的本质是：从公开密钥破译私有密钥时，需考虑 $f(i)$ 的全排列数 $n!$。因此，当 n 足够大时，$f(i)$ 的排列在有效时间内不可穷举。但是，从私有密钥解开密文时只需考虑 $f(i)$ 的累加和，时间复杂度与 n 多项式相关，故解密是可行的。

若把密文当作支点，则 $f(i)$ 是一边计算量大，另一边计算量小，故称 $f(i)$ 为杠杆函数。

可以验证，$\{C_1, C_2, \cdots, C_n\}$ 各项之间并非两两互素，所以称 $\{C_i\}$ 为非互素序列。

(2)从非互素序列得到互素序列。

令 $W_i = W^{f(i)} \bmod M$。由于 Z_M^* 是群，满足 $(W_i^{-1} * W_i) \bmod M = 1$ 的 W_i 之逆元 W_i^{-1} 必定存在，故有 $A_i = (C_i * W_i^{-1}) \bmod M, i = 1, 2, \cdots, n$。

2.4 从非互素序列乘积得到互素序列乘积

2.4.1 非互素序列乘积的计算

设 $\{C_1, C_2, \cdots, C_n\}$ 为 n 项非互素序列，从中任选 $m(1 \leqslant m \leqslant n)$ 个项，则这 m 个项的连乘积被称为非互素序列乘积。以下用 G_N 代表非互素序列乘积。

与 n 位二进制数 $b_1 b_2 \cdots b_n$ 相对应的非互素序列乘积为 $G_N = \prod C_i^{b_i} \bmod M$。其中，$i$ 从 1 至 n，b_i 表示幂次。如果 $b_1 b_2 \cdots b_n$ 是明文分组，则 G_N 是相应的密文。

2.4.2 互素序列乘积的导出

设 W^{-1} 为 W 的逆元，满足 $(W * W^{-1}) \bmod M = 1$。由于 Z_M^* 是交换群，所以对于任意正整数 k $(1 \leqslant k < M)$，有 $(W^k * (W^{-1})^k) \bmod M = (W^k * W^{-k}) \bmod M = 1$ 一定成立。

下面证明 $(G_N * (W^{-1})^k) \bmod M = G_P$。

设 $\{A_i\}$ 为互素序列，$f(i)$ 为杠杆函数，$b_1 b_2 \cdots b_n$ 为 n 位二进制数，$k = \sum (f(i) * b_i)$。

由于 $C_i = (A_i * W^{f(i)}) \bmod M$，$G_N = \prod C_i^{b_i} \bmod M$，因此有

$$(G_N * (W^{-1})^k) \bmod M =$$
$$(\prod C_i^{b_i} * (W^{-1})^k) \bmod M =$$
$$(\prod (A_i * W^{f(i)})^{b_i} * (W^{-1})^k) \bmod M =$$
$$(\prod A_i^{b_i} * (W^{\sum (f(i) * b_i)}) * (W^{-1})^k) \bmod M =$$
$$(\prod A_i^{b_i} * (W^k) * (W^{-1})^k) \bmod M =$$
$$\prod A_i^{b_i} \bmod M = \prod A_i^{b_i} = G_P$$

上述证明过程从理论上给出了求 G_P 的方

法。不过在实际求解中，明文 $b_1b_2\cdots b_n$ 是事先不可知的，因而就无法计算 k。但是，由于 k 的取值范围极为有限（$1 \leqslant k \leqslant \sum f(i)$），所以可以搜索 k，并根据 G_P 的值能否被 $\{A_i\}$ 的某些项整除为 1 来验证 G_P。从 2.2 节知，在 G_P 得到验证的同时，也就求出了原二进制明文。

2.4.3 解的唯一性问题

由于 $\{C_i\}$ 是非互素序列，其子序列的连乘积不唯一，所以，采用边搜索边验证的方法来求取 G_P 时，符合条件的 G_P 值可能不止一个，从而会导致 $b_1b_2\cdots b_n$ 的不唯一性。

设密文 G_N 可由两个不同子序列的连乘积得到，因此

$$G_N \equiv C_{x_1} * C_{x_2} * \cdots * C_{x_m} \equiv (C_{y_1} * C_{y_2} * \cdots * C_{y_h}) (\mathrm{mod}\ M)$$

即 $(A_{x_1} * A_{x_2} * \cdots * A_{x_m}) * W^{k1} \equiv ((A_{y_1} * A_{y_2} * \cdots * A_{y_h}) * W^{k2}) \mathrm{mod}\ M)$

其中，$k1 = f(x_1) + f(x_2) + \cdots + f(x_m)$，$k2 = f(y_1) + f(y_2) + \cdots + f(y_h)$

不妨令 $k_1 \geqslant k_2$，由于 Z_M^* 是交换群，有

$$W^{k1-k2} \equiv ((A_{y1} * A_{y2} * \cdots * A_{yh}) * (A_{x1} * A_{x2} * \cdots * A_{x_m})^{-1}) \mathrm{mod}\ M$$

简记为

$$W^{k1-k2} \equiv (\prod A_{y_i} * \prod A_{x_j}^{-1}) \mathrm{mod}\ M$$

上式表明，当 G_P 的值不唯一时，必有 W 的值与 $\prod A_{y_i} * \prod A_{x_j}^{-1}$ 有关。其逆否含义是，如果 W 的值与 $\prod A_{y_i} * \prod A_{x_j}^{-1}$ 无关，则 G_P 的值将会唯一。因此，我们需要确定 W 取 $\prod A_{y_i} * \prod A_{x_j}^{-1}$ 相关值的可能性有多大。

由于 2^{80} 约为 120 892 581 961 463 $\times 10^{10}$，若窃密者用简单穷举法来攻击 80 比特的密文分组，且其计算机以每秒验证 1 万亿个值的速度不停地工作，那么，要验证完所有可能的取值将需要 38 334 年。故当项数 $n \geqslant 80$ 时，窃密者在有效时间内不能穷尽 2^{80} 个值。

注意，当项数 $n = 80$ 时，$\prod A_{y_i} * \prod A_{x_j}^{-1}$ 型值的个数最多为 2^{160} 个。另一方面，前 80 个素数显然可以组成一个连乘积最小的互素序列。这时，模数 M 约为 2^{552}，故在 $n \geqslant 80$ 时，任意 W 取得 $\prod A_{y_i} * \prod A_{x_j}^{-1}$ 相关值的可能性小于 $2^{160}/2^{552}$，即 $1/2^{392}$。显然，它几乎为零。并且，当 W 取素数时，这种可能性将进一步降低。

这说明，G_P 取值不唯一的可能性也几乎为零，故 2.4.2 节的试探求解法是可行的。

3 求模逆元的 REESSE 递归算法

作者受到大衍求一术思想的启发[4]，设计了一个求模逆元的递归算法。该算法的优点在于它不仅可以用来求模逆元，而且也可以用来求一元线性同余方程的一般解。两种情况下，时间复杂度不超过 $O(\log_2 M)$。

设线性同余方程为 $a * x \equiv b \mathrm{mod}\ M$，$a$、$M$ 互素，欲求 x。若 $b = 1$，则 x 为 a 的逆元。显然，同余方程可以表示为：$a * x - y * M = b$。

设子程序入口参数为：a、M、b、x、y。其中，a、M、b 传递值，为已知量；x、y 传递名，用来存放返回值。算法如下：

(1) 定义局部变量：$q1$、$q2$、$r1$、$r2$、t；
(2) 如果 a 等于 1，则 $x = b$，$y = 0$，并转至(7)，否则，执行下一步；
(3) 求 $q1$、$r1$，使得 $M = q1 * a + r1$，求 $q2$、$r2$，使得 $b = q2 * a + r2$；
(4) $M = a$，$a = r1$，$b = -r2$，$t = y$，$y = x - q1 * y - q2$，$x = t$；
(5) 以 a、M、b、x、y 为参数，调用子程序自身；
(6) $t = x$，$x = y + q1 * x + q2$，$y = t$；
(7) 返回上一层子程序之(6)或返回主程序。

注意，在主程序中，应判断 x、y 是否为负数。如果 x 为负数，则把 $x + M$ 赋予 x；如果 y 为负数，则把 $y + a$ 赋予 y。此算法已被程序实现。

4 REESSE 1 公钥体制的算法描述

公钥密码体制和对称密钥密码体制互有优势。因此，在实际应用中，一般是采用混合密码体制，即用对称算法来加密明文，再用公钥算法来加密对称算法的密钥。

4.1 密钥的生成

(1) 随机产生项数为 n 的互素序列 $\{A_1, A_2, \cdots, A_n\}$；
(2) 找到一个正素数 M，使得 $M > \prod A_i$，且 $M - 1$ 含 $n + 4$ 范围内所有素数因子；
(3) 选取一个正素数 W，有 $W < M$，根据 REESSE 递归算法求出 W^{-1}；
(4) 随机产生杠杆函数值 $f(1), f(2), \cdots, f(n)$，$5 \leqslant f(i) \leqslant n + 4$，且两两不同；
(5) 对于 $\{A_i\}$，$C_i = (A_i * W^{f(i)}) \mathrm{mod}\ M$，$i = 1, 2, \cdots, n$。

最后，得到公开密钥（$\{C_i\}$、M）和私有密钥（$\{A_i\}$、W^{-1}、$f(i)$、M）。

4.2 加密（用 $\{C_i\}$、M 作为公开密钥）

设 $b_1b_2\cdots b_n$ 为 n 位二进制明文分组。

(1) 令 $G_N = 1$，$i = 1$；
(2) 如果 $b_i = 1$，则 $G_N = (G_N * C_i) \mathrm{mod}\ M$；

(3) $i=i+1$,如果 $i\leqslant n$,则转至(2),否则,结束。
最后,G_N 即为所求的密文。

4.3 解密(用$\{A_i\}$、W^{-1}、$f(i)$、M 作为私有密钥)

注意,解密算法中并没有用到 $f(i)$的值,G_N 为已知的密文。

(1) $G_N=(G_N * W^{-1}) \bmod M$;
(2) 令 $b_1b_2\cdots b_n$ 各位皆为 0,$G_P=G_N$,$i=1$;
(3) 如果 G_P/A_i 为整数,则 $b_i=1$,且 $G_P=G_P/A_i$;
(4) $i=i+1$,如果 $i\leqslant n$ 并且 $G_P\neq 1$,则转至(3);
(5) 如果 $G_P\neq 1$,则转至(1),否则,结束。
最后,$b_1b_2\cdots b_n$ 即为原二进制明文分组。

4.4 数字签名(用$\{A_i\}$、W^{-1}、$f(i)$、M 作为私有密钥)

在本算法和下一个算法中,常量 S、$T\geqslant 5$,为 $M-1$ 之互异真因子,且 $S*T<M$。

(1) 利用 HASH 函数,得到签名文件的 n 位比特摘要 $H=b_1b_2\cdots b_n$;
(2) 产生小于 M 的随机正整数 R;
(3) 令 $K=0$,$U=R$,$V=R$,$i=1$;
(4) $V=(V * A_i) \bmod M$,如果 $b_i=0$,则 $U=(U * A_i) \bmod M$,否则 $K=K+f(i)$;
(5) $i=i+1$,如果 $i\leqslant n$,则转至(4);
(6) $Q=(W^{-1})^K \bmod M$,$U=(U * Q * H)^S \bmod M$,$V=(V)^T \bmod M$。
最后,得到数字签名码 U、V,可将其附在签名文件的后面发送给接收方。

4.5 身分验证(用$\{C_i\}$、M 作为公开密钥)

(1) 利用 HASH 函数,得到签名文件的 n 位比特摘要 $H=b_1b_2\cdots b_n$;
(2) 令 $Q=1$,$i=1$;
(3) 如果 $b_i=1$,则 $Q=(Q * C_i) \bmod M$;
(4) $i=i+1$,如果 $i\leqslant n$,则转至(3);
(5) $X=(Q^S * U)^T \bmod M$,$Y=(H^T * V)^S \bmod M$;
(6) 如果 X、Y 相等,则说明签名者身分正确,且签名文件没有被修改;如果 X、Y 不等,则说明签名者身分不正确,或者签名文件已被修改。

数字签名与身分验证算法配合使用,可以达到鉴别身分、抗修改和抗抵赖的目的。

5 REESSE 1 安全性初步分析

5.1 从密文与公开密钥破译出明文是一个难题

非互素序列乘积(即密文)$G_N=\prod C_i^{b_i} \bmod M$ 可以表示为 $G_N=\prod C_i^{b_i}-L*M$。其中 L 为一正整数,移项可得 $\prod C_i^{b_i}=G_N+L*M$。窃密者如果想获知某个 b_i 是否为 1,则必须知道对应的 C_i 能否整除 G_N+L*M,进而须测定 L 的值。但是,根据密钥生成算法中 M 和 $\{C_i\}$ 的定义,L 的上限肯定大于 2^n。由 2.4.3 节知,L 无法有效穷举。

5.2 从公开密钥推导出私有密钥是一个难题

(1)$\{A_i\}$排列顺序的不可见性。为使模数 M 不至于太大,互素序列 $\{A_i\}$ 的取值不会太大。一般地,单独一个小数值很难隐藏在大数值中。但是,在 REESSE 1 体制的公钥中隐藏的不仅是 $\{A_i\}$ 每个项的值,而且也是 $\{A_i\}$ 这个序列的排列顺序,因而使得 A_i 的大小并不影响体制的安全性。

(2)杠杆函数 $f(i)$的不确定性。首先,如果窃密者企图猜测 $f(i)$的排列,则当 $n\geqslant 80$ 时,它的全排列数将达 P_{80}^{80},远远大于 2^{80}。由 2.4.3 节知,$f(i)$的排列不可有效穷举。其次,令 $W_i=W^{f(i)} \bmod M$,当 W_i 确定时,由数论的知识知,在 $f(i)$与 $M-1$ 不互素时,W 与 $f(i)$之间没有一一对应关系[2]。当 W 确定时,由群论知[3],W_i 与 $f(i)$之间也没有一一对应关系,进而 C_i 与 $f(i)$之间没有一一对应关系。这些充分说明,在某个具体的公钥序列中,$f(i)$的排列是不确定的。

(3)密钥映射的不充分性。在 REESSE 1 体制中,私有密钥虽然包括 $\{A_i\}$、$f(i)$、W 三部分,但只有 $\{A_i\}$ 到 $\{C_i\}$ 的直接映射。由于 $C_i=(A_i * W^{f(i)}) \bmod M$ 实际包含了 n 个非线性等式和 $2n+1$ 个未知变量,所以求解 $\{A_i\}$ 和 W 是困难的。

5.3 数字签名算法的安全性

(1)从数字签名码得到私有密钥是个难题。数字签名码 $U=(U * Q * H)^S \bmod M$,$V=(V)^T \bmod M$。首先,根据 Gauss 的结论,当 S、T 分别为 $M-1$ 的真因子时,求解 U 的 S 次方根、V 的 T 次方根是极为困难的;其次,随机数 R 对私有密钥 $\{A_i\}$ 有掩护作用。故从数字签名码得到私有密钥是个难题。

(2)从公开密钥伪造数字签名码是个难题。从 4.5 节知,如果正整数 U、V 满足等式 $(Q^S * U)^T\equiv(H^T * V)^S \bmod M$,则 U、V 是数字签名码。如果窃密者假设一个 U 值,并通过等式的右边来求出 V,则 U、V 是一对伪造的数字签名码。但是,根据 Gauss 的结论,当 S 为 $M-1$ 的真因子时,求解 $(Q^S * U)^T$ 的 S 次方根是极为困难的。所以,从公开密钥伪造数字签名码是个难题。

6 结束语

公钥密码体制可以用于电子商务、电子政务、国防信息系统等方面。一个公钥体制能不能加以利用,关键要看它的安全性。本文试图从上述三个方面来回答这个问题,但毕竟还是浅显的,因此,作者期待同仁对 REESSE 1 的安全性作出更深入的分析。

作者已在 Windows 98 下开发 (下转第 30 页)

所有跨线程调用，应使用上述方法封送。Invoke 方法用于同步调用委托；BeginInvoke 和 EndInvoke 方法用于异步调用委托。Invoke 方法采用对委托的引用，通常此委托是 MethodInvoker 委托的一个实例：Invoke public Object Invoke(Delegate method)。

在拥有此控件的基础窗口句柄的线程上调用给定的委托，并且在委托的方法返回委托方法的结果。但是，Invoke 方法并不能在控件所属的同一个线程上调用，而必须使用 BeginInvoke 和 EndInvoke 方法。上述 DataGrid1 是主线程创建的控件，在异步调用线程对其直接绑定是无效的。因此，需要定义一个委托，在该委托的方法中将 DataSet 绑定到 dataGrid1：

```
private delegate void otherThdCallBack();
private otherThdCallBack callBack;
callBack = new otherThdCallBack(this.otherThdCallBackMethod);
private void otherThdCallBackMethod()
{   this.dataGrid1.DataSource = ds.Tables[0];}
```

然后，在异步调用线程中执行 dataGrid1 的 Invoke 方法：

```
ds = srv.EndgetData(ira);this.dataGrid1.Invoke(this.callBack)
```

至此获得了预期的异步调用结果。

5 结束语

本文开发的网上失业保险费申报系统作为一种电子政务的网络服务，具有较多的用户交互和异步操作与处理过程，采用.NET 框架技术实现了简化系统结构、缩短开发过程和提高系统运行效率等目标，希望能对开发相似 Web 服务系统的读者提供一定的参考。

参考文献：

[1] Louis Rosenfeld. How Information Architecture Can Help[Z]. Michigan State University, 2002.

[2] Simon Robinson.杨浩，杨铁男译.C#高级编程[M].北京：清华大学出版社，2002.

[3] Fabio Claudio Ferracciati, Jay Glynn.毛尧飞译. .NET 数据服务 C#高级编程[M].北京：清华大学出版社，2002.

(上接第 12 页)

为一种安全的网上支付协议，也日益得到了国际 IT 界的认同。而且，有必要在不损害其安全性的前提下，提高其便捷性，使之不断完善，从而得到更广泛的应用。本文就 SET 协议的便捷性提出了几点改进方法，由于降低了加密要求，用户可以不需要去验证一系列证书来找到商家的公钥。这些都可以使用户的效率大为提高。

参考文献：

[1] Visa and MasterCard. Secure Electronic Transaction: Business Description[S].1997.

[2] Albert Levi, Cetin Kaya Koc. CONSEPP: CONvenient and Secure Electronic Payment Protocol Based on X959[S]. 1997.

[3] 汤志华，林浒，马跃. SET 协议安全性分析[J]. 小型微型计算机系统，1999，20(9)：645－649.

[4] 张大陆，李恩民. SET 的支付流程分析和改进措施[J]. 微型电脑应用，2001，17(8)：23－28.

(上接第 16 页)

出 REESSE 1 公钥体制的演示程序，包括密钥生成、加密、解密、数字签名和身分验证等五个部分。在程序中，作者对解密算法进行了优化，使其时间复杂度降至约 $O(n^2)$的水平，并且最大整数(即模数 M)可以控制在 2^{384}范围之内。

参考文献：

[1] Bruce Schneier. 吴世忠译. Applied Cryptography[M]. 北京：机械工业出版社，2000.

[2] 潘承洞，潘承彪. 初等数论[M]. 北京：北京大学出版社，1992.

[3] 李超，谢端强. 代数学基础[M]. 长沙：国防科学技术大学出版社，2000.

[4] 万哲先. 孙子定理和大衍求一术[M]. 北京：高等教育出版社，1989.

(上接第 22 页)

本文提出的用于网络系统服务器信息流切换的模糊控制器是有效的。

参考文献：

[1] S A Lippman. Semi-Markov Decision Processes with Unbounded Rewards[J]. Management Science, 1973, 19(7):717－731.

[2] R R Weber. On the Optimal Assignment of Customers to Parallel Servers[J]. Journal Applied Probability, 1988, 15: 104－109.

[3] S Xu, R Righter, J G Shanthikumar. Optimal Dynamic Assignment of Customers to Heterogeneous Servers in Parallel[J]. Operation Research, 1992, 40:1126－1138.

[4] M Hlynka, D A Stanford, W H Poon, et al. Observing Queues Before Joining[J]. Operation Research, 1994, 42:365－371.

[5] Runtong Zhang, Yannis A Phillis. Fuzzy Control of Queuing Systems with Heterogeneous Servers[J]. IEEE Trans on Fuzzy systems, 1999, 7(1):17－26.

[6] 杨洪勇，宗广灯，武玉强. 多输入多输出网络系统的拥塞控制方法[J]. 计算机工程与应用，2002，38(15)：27－30.