AES 与 RSA 算法优化及其混合加密体制*

肖振久[1,2]，胡 驰[1]，姜正涛[2]，陈 虹[1]
(1. 辽宁工程技术大学 软件学院，辽宁 葫芦岛 125105；2. 中国传媒大学 计算机学院，北京 100024)

摘 要：针对 AES 密钥扩展过程存在的安全隐患，提出一种改进的密钥扩展方法。针对轮函数中列混合和逆列混合运算耗时相差较大的问题，找出有限域 $GF(2^8)$ 上最简形式的列混合和逆列混合运算，使其在加解密过程中消耗同样的运算资源；RSA 方面，针对其运算效率的缺陷，将传统双素数变为四素数，在签名(解密)过程中，采用中国剩余定理结合蒙哥马利模乘来优化大数的模幂运算。在此基础上，结合两种算法的优点，并采用消息摘要、数字签名、数字信封等技术构建了一个既方便密钥管理又确保加密解密效率的混合加密体制。实验结果表明优化后的算法在运算速度上有一定优势并且有较高的可行性。

关键词：AES 密码算法；RSA 密码算法；中国剩余定理；消息摘要；数字签名；数字信封
中图分类号：TP309.7 **文献标志码**：A **文章编号**：1001-3695(2014)04-1189-06
doi:10.3969/j.issn.1001-3695.2014.04.056

Optimization of AES and RSA algorithm and its mixed encryption system

XIAO Zhen-jiu[1,2], HU Chi[1], JIANG Zheng-tao[2], CHEN Hong[1]
(1. *College of Software, Liaoning Technical University, Huludao Liaoning* 125105, *China*; 2. *School of Computer, Communication University of China, Beijing* 100024, *China*)

Abstract: In order to overcome AES key expansion's safety problem, this paper proposed a new key expansion method and used the simplest MixColumn and inverse MixColumn operation on $GF(2^8)$ to replace the former. In aspect of RSA, this paper suggested a four prime RSA with Chinese remainder theorem and Montgomery algorithm to improve the computing efficiency. Then it proposed a mixed encryption system and created a high security communicating model. Experiments show that advanced algorithm is faster than the original and mixed system has higher practicability.

Key words: AES algorithm; RSA algorithm; Chinese remainder theorem(CRT); message digest; digital signature; digital envelope

0 引言

随着网络和通信技术的发展，信息安全问题受到人们广泛的关注，以数据加密为核心的信息安全技术得到了飞速发展。数据加密技术根据密钥类型可分为对称加密和非对称加密。AES 是目前主流的对称加密算法，它是继 DES 之后新一代的数据加密标准。1997 年美国国家标准技术研究所(NIST)发布公告征集新的加密标准，2000 年由比利时的 Daemen 等人[1]提交的 Rijndael 算法被不加修改地宣布为 AES 算法。该算法经验证是所有候选算法中安全性最高、运行速度最快的迭代分组密码算法。RSA[2]是应用最广泛的非对称加密算法，它的特点是安全性高，易于实现，既可用来加密数据，又能用于身份认证。本文分析了这两种算法的研究现状，对还存在的问题加以改进，然后将其结合起来，再采用消息摘要、数字签名、数字信封等安全技术构建了一个混合加密体制，使其既能安全高效地传输数据，又能有效验证数据来源，防止一方抵赖的情况发生。

1 AES 算法研究现状

AES 作为新一代数据加密标准有高安全性、高性能和灵活易用等优点，然而，随着网络的发展，不安全因素越来越多，对运行效率要求也越来越高。文献[3]指出 AES 密钥扩展算法虽然有直接高效的特点，但也有通过其中任何一轮子密钥即可破解全部密钥的缺陷。文献[4]就此缺陷提出了三种改进算法，前两种算法运算效率会降低，而且对第三轮及之前的子密钥攻击防御较弱，第三种算法虽然使得对密钥扩展运行两轮之后的攻击强度与暴力破解相同，但是算法结构过于复杂，不易实现。文献[5]指出 AES 中的 S 盒存在仿射变换对周期小、迭代输出周期短和代数表达式只有 9 项的缺陷。文献[6]通过对字节作一次仿射变换后再求逆元，然后再作一次仿射变换的方式构造了新 S 盒，使之克服了原 S 盒的缺陷，具有更好的代数性质。文献[7,8]通过改变 AES 算法结构，使得加密和解密存在可重复利用的单元，提出一种可重构的设计方式，非常适合硬件实现。文献[9]提出将 AES 轮变换中的四种变换利用查表的方式简化成一步来提高 AES 算法运行效率，但是该文

收稿日期：2013-05-16；**修回日期**：2013-06-29 **基金项目**：北京市自然科学基金资助项目(4112052)；国家自然科学基金资助项目(61103199)

作者简介：肖振久(1968-)，男，内蒙古宁城人，副教授，主要研究方向为网络与信息安全、数字版权管理(hcshrhh001@163.com)；胡驰(1988-)，男，湖北武汉人，硕士研究生，主要研究方向为数据加密、数字签名；姜正涛(1976-)，男，山东青岛人，教授，主要研究方向为密码箱理论与分析、可信计算与抗毁生存等；陈虹(1967-)，女，辽宁阜新人，副教授，主要研究方向为网络安全.

献没有解决密钥扩展中存在的安全隐患和解密相对于加密存在时延的问题。

2 AES 算法优化

本文在文献[9]查表法的基础上,改进了密钥扩展方法并解决了加密解密耗时不对等的问题,使其在保证安全性的前提下进一步提高了运算效率。

2.1 AES 密钥扩展分析与改进

AES 密钥扩展过程如图 1 所示(本文均以 10 轮 AES 为例)。初始密钥为 128 bit,4 个字 w_0、w_1、w_2 和 w_3。然后由 w_0 和 w_3 生成 w_4,由 w_1 和 w_4 生成 w_5,由 w_2 和 w_5 生成 w_6,由 w_3 和 w_6 生成 w_7。w_4、w_5、w_6 和 w_7 称为第一轮扩展密钥,再以此为基础依次下推,直到生成全部密钥。其中每轮密钥的第一个字要进行一次复杂的运算 f,它由一个 S 盒函数 SubWord、一个移位运算 RotWord 和一个轮常量异或运算组成。每一轮的运算只依赖于上一轮,依次下推可得到所需的任意轮密钥。这种密钥扩展方法具有高效性和即时性的优点,但是如果攻击者得到其中一轮密钥就可以破获全部密钥。因为每个字 w_i 都与 w_{i-1} 和 w_{i-4} 相关,也就是说知道其中任意两个就能推出第三个。假设攻击者知道了 AES 某一轮密钥 w_i、w_{i+1}、w_{i+2} 和 w_{i+3},那么可以通过 w_{i+2} 和 w_{i+3} 推得 w_{i-1},通过 w_{i+1} 和 w_{i+2} 推得 w_{i-2},通过 w_i 和 w_{i+1} 推得 w_{i-3},再通过 w_{i-1} 和 w_i 推得 w_{i-4},这样便得到该轮密钥前一轮全部子密钥,也可以通过类似的方法得到后一轮密钥,从而得到全部密钥。

针对以上的安全隐患,本文从抗攻击强度并兼顾程序执行时间方面考虑,提出一种改进的密钥扩展算法:初始密钥不变,第一轮扩展密钥用一套与初始密钥无关的新密钥进行填充,在新密钥的基础上,再用 AES 固有的算法进行密钥扩展,直到生成全部子密钥。该方法的原理如图 2 所示。这样改动之后,由于初始密钥和扩展密钥之间不存在任何关系,所以攻击者无法通过某一轮密钥来推出全部密钥。如果采用穷尽密钥攻击,设种子密钥长为 k bit,穷尽密钥攻击的最好情况是 1,最坏情况是 2^k,由于每种情况的概率都相等,所以平均复杂度为

$$\sum_{i=1}^{2^k}\frac{1}{2^k}\times i=\frac{1}{2^k}\times\sum_{i=1}^{2^k} i=$$

$$\frac{1}{2^k}\times\frac{(1+2^k)\times 2^k}{2}=\frac{1}{2}+2^{k-1}\approx 2^{k-1}$$

对于 10 轮 AES 算法来说,攻击者平均需尝试 2^{127} 种可能的密钥,而本文的密钥扩展算法使得攻击者平均需尝试 2^{255} 种可能的密钥。以目前的计算能力而言,完成这种穷尽搜索至少需要数亿年时间。因此,改进的密钥扩展方法只是在原有的方法上作了一小部分改动,既克服了原来的安全隐患又能保证程序运行效率。

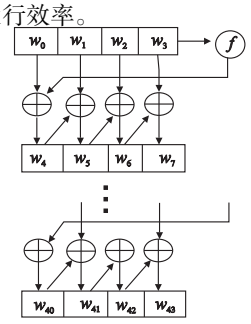

图1 AES密钥扩展过程

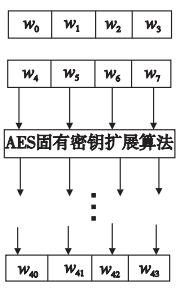

图2 改进的密钥扩展方法

2.2 列混合与逆列混合变换分析与优化

AES 算法加密运算与解密运算耗时不对等,分析其原因是由于列混合与逆列混合变换复杂度不同所致。在列混合变换中,把状态的每一列看做 $GF(2^8)$ 域上的多项式,并与一个固定的多项式 $C(x)=\{03\}x^3+\{01\}x^2+\{01\}x+\{02\}$ 相乘,然后模多项式 x^4+1 取余。用矩阵表示如下:

$$\begin{pmatrix}02&03&01&01\\01&02&03&01\\01&01&02&03\\03&01&01&02\end{pmatrix}\begin{pmatrix}a_{00}&a_{01}&a_{02}&a_{03}\\a_{10}&a_{11}&a_{12}&a_{13}\\a_{20}&a_{21}&a_{22}&a_{23}\\a_{30}&a_{31}&a_{32}&a_{33}\end{pmatrix}=$$

$$\begin{pmatrix}b_{00}&b_{01}&b_{02}&b_{03}\\b_{10}&b_{11}&b_{12}&b_{13}\\b_{20}&b_{21}&b_{22}&b_{23}\\b_{30}&b_{31}&b_{32}&b_{33}\end{pmatrix}$$

解密过程中逆列混合变换同样可以表示为矩阵的乘法:

$$\begin{pmatrix}0E&0B&0D&09\\09&0E&0B&0D\\0D&09&0E&0B\\0B&0D&09&0E\end{pmatrix}\begin{pmatrix}b_{00}&b_{01}&b_{02}&b_{03}\\b_{10}&b_{11}&b_{12}&b_{13}\\b_{20}&b_{21}&b_{22}&b_{23}\\b_{30}&b_{31}&b_{32}&b_{33}\end{pmatrix}=$$

$$\begin{pmatrix}a_{00}&a_{01}&a_{02}&a_{03}\\a_{10}&a_{11}&a_{12}&a_{13}\\a_{20}&a_{21}&a_{22}&a_{23}\\a_{30}&a_{31}&a_{32}&a_{33}\end{pmatrix}$$

不难看出,解密过程要比加密过程复杂得多。加密过程中,列混合变换需要执行 4 次 xor 加法运算和 2 次 xtime 乘法运算,而解密过程中的逆列混合变换需要执行 9 次 xor 加法运算和 12 次 xtime 乘法运算[10]。由于乘法运算消耗较多的时间和空间资源,从而导致解密过程相对于加密过程存在一定时延,这在实际应用过程中,往往让用户难以接受。

针对以上问题,文献[7]通过将解密矩阵进行分解,使其变为两个较简单矩阵的乘积,以降低倍乘次数从而提高解密速度。但是,这种方法具体实现起来较复杂,效率提高也不明显。本文利用定理 1 找出了具有最简形式的列混合运算和逆列混合运算。

定理 1 在有限域 $GF(2^8)$ 中,若一个线性矩阵 $\boldsymbol{A}$,其形式为 $\boldsymbol{A}=\begin{pmatrix}a&b&c&d\\d&a&b&c\\c&d&a&b\\b&c&d&a\end{pmatrix}$,其中 $a,b,c,d\in GF(2^8)/\{0\}$,若 $\boldsymbol{A}^{-1}=\boldsymbol{A}$,则 $\boldsymbol{A}=\begin{pmatrix}a&b&c&b\\b&a&b&c\\c&b&a&b\\b&c&b&a\end{pmatrix}$,其中 $a^2+c^2=1$。

证明 设 g 是有限域 $GF(2^8)$ 中的一个生成元,α、β、γ、ρ 分别是元素 a、b、c、d 的阶,由 $\boldsymbol{A}\boldsymbol{A}^{-1}=1$ 可构造如下方程:

$$\begin{pmatrix}a&b&c&d\\d&a&b&c\\c&d&a&b\\b&c&d&a\end{pmatrix}\begin{pmatrix}a&b&c&d\\d&a&b&c\\c&d&a&b\\b&c&d&a\end{pmatrix}=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}$$

可得 $\begin{cases}a^2+c^2=1\\b^2+d^2=0\end{cases}$,即 $\begin{cases}g^{2\alpha}+g^{2\gamma}=1\\g^{2\beta}+g^{2\rho}=0\end{cases}$,由于 $g^{2\beta}+g^{2\rho}=0$,$1\leqslant\beta,\rho$

$\leqslant 255$,于是有 $2\beta = 2\rho$ 或 $2\beta = 2\rho \pm 255$,但由于 2β 是偶数,所以 $2\beta = 2\rho \pm 255$ 不成立,因而 $2\beta = 2\rho$,即 $\beta = \rho$,推得 $b = d$。综上可得 $\boldsymbol{A} = \begin{pmatrix} a & b & c & b \\ b & a & b & c \\ c & b & a & b \\ b & c & b & a \end{pmatrix}$,其中 $a^2 + c^2 = 1$。

根据上述定理,令

$\boldsymbol{M} = \begin{pmatrix} 2 & 1 & 3 & 1 \\ 1 & 2 & 1 & 3 \\ 3 & 1 & 2 & 1 \\ 1 & 3 & 1 & 2 \end{pmatrix} = \boldsymbol{M}^{-1}$,用此矩阵替换原算法中的列混合和逆列混合矩阵,使得列混合和逆列混合运算消耗相同的运算资源,解决了解密相对于加密存在的时延问题,具有很高的实用价值。

3　RSA 算法改进及其在数字签名中的应用

RSA 算法中大数的模幂运算比较费时,这一直是制约 RSA 发展的瓶颈。早期,人们建议使用较小的加密指数或解密指数以加快加解密等基本运算,但是,1990 年 Wiener[11] 提出当私钥 d 小于模数 $N^{1/4}$ 时,RSA 密码系统是不安全的,其分析方法的本质是利用了连分数中的 Legendre 定理。随后,1999 年 Boneh 等人[12] 把弱密钥 d 的上界提高到 $d < N^{0.292}$。因此,出于安全性考虑,目前 RSA 密码系统的模数为 1024 ~ 2048 bit,如此庞大的模数,其运算效率必然受到影响。针对这一问题,很多学者提出了不同的优化算法,如基于乘同余对称特性和指数 2^k 次方组合优化算法、加法链方法、滑动窗口法和模重复平方算法等[13],这些都是从运算操作的角度去优化,还有一些从结构上进行改进的算法,如 Batch RSA、Mprime RSA、Mpower RSA 和 Rebalanced RSA[14]。这些方法都在一定程度上提高了运算效率,其中效果比较明显的是利用中国剩余定理(CRT)进行解密或签名(本文把融入中国剩余定理的 RSA 算法叫做 CRT-RSA 算法)。已证实,若考虑中国剩余定理的计算代价,双素数 CRT-RSA 在运算速度上分别是原算法的 3.32 倍(模为 1024 bit)和 3.47 倍(模为 2048 bit)[15]。这个速度虽令人满意,却存在安全隐患,文献[16]指出传统双素数 CRT-RSA 签名算法容易遭受出错攻击,攻击者能够利用错误的签名来分解模数 N。

本文从算法安全性、运算效率和易操作性方面考虑,将原双素数 RSA 算法变为四素数,再将其运用到数字签名中,具体思路如下。

3.1　四素数 RSA 算法基本原理

在传统双素数 RSA 密码算法基础上,把素数个数取为 4,算法依然成立,其描述如下:

a)随机选取四个不同的大素数 p、q、r 和 s,计算

$$n = pqrs, \varphi(n) = (p-1)(q-1)(r-1)(s-1)$$

b)取加密密钥,满足一定条件的 e,计算出私钥 d,满足 $de \equiv 1 \bmod \varphi(n)$。

c)加解密过程与传统算法一样,仍为

$$\text{加密算法:} c = E(m) \equiv m^e \bmod n$$

$$\text{解密算法:} m = D(c) \equiv c^d \bmod n$$

下面,本文从数论的角度来证明算法的正确性。

证明　假设明文为 m,密文 c,密钥(d,n),公钥(e,n)。由加解密过程有 $D(c) \equiv c^d \bmod n \equiv (m^e)^d \bmod n \equiv m^{ed} \bmod n$,又因为 $de \equiv 1 \bmod \varphi(n)$,所以 $de \equiv 1 + k\varphi(n)$,其中 k 为正整数,带入上式得 $D(c) \equiv m^{1+\varphi(n)} \bmod n$。若 $\gcd(m,n) = 1$,根据欧拉定理有 $m^{\varphi(n)} \equiv 1 \bmod n$,故有 $m^{1+\varphi(n)} \equiv m \bmod n$;若 $\gcd(m,n) \neq 1$,由于 $n = pqrs$,故 (m,n) 中必含 p、q、r、s 之一,或者 pq、pr、ps、qr 之一,或者 pqr、prs、qrs 之一。由于这几种情况类似,这里只给出 $\gcd(m,n)$ 包含 p,q,r,s 之一的证明过程。若 $\gcd(m,n) = p$、$m = cp$、$1 \leqslant c < qrs$,由欧拉定理得:$m^{\varphi(q)} \equiv 1 \bmod q$,$m^{\varphi(r)} \equiv 1 \bmod r$,$m^{\varphi(s)} \equiv 1 \bmod s$。因此,对任意 k 恒有 $m^{k(q-1)} \equiv 1 \bmod q$,$m^{k(q-1)(r-1)(s-1)(p-1)} \equiv 1 \bmod q$,就有 $m^{k(q-1)(r-1)(s-1)(p-1)} \equiv 1 + hq$,得出 $m^{k\varphi(n)} \equiv 1 + hq$,因为 $m = cp$,所以得出 $m^{k\varphi(n)+1} \equiv m + cphq$。已知 p,q,r,s 都是素数,故 $\gcd(p,qrs) = 1$,$\gcd(m,qrs) = 1$,推出 $m^{\varphi(qrs)} \equiv 1 \bmod qrs$,对任意 k,总有 $m^{k(q-1)(r-1)(s-1)} \equiv 1 \bmod qrs$,所以 $m^{k(q-1)(r-1)(s-1)(p-1)} \equiv 1 \bmod qrs$,即 $m^{k\varphi(n)} \equiv 1 + \text{hqrs}$;又因为 $m = cp$,所以 $m^{k\varphi(n)+1} \equiv m + chpqrs$,这就证明了 $m^{k\varphi(n)+1} \equiv m \bmod n$。其他同理可证均成立。

3.2　四素数 RSA 算法在数字签名中的应用

首先简单介绍一下杂凑函数,它是指将任意长度的消息映射成某一固定长度消息的一种函数,一般用于消息完整性检测和认证。它是一种单项散列函数,也就是说给定一个消息 M,用杂凑函数 H 生成消息摘要 $D = H(M)$,这个计算很容易,但反推 M 却很难。把杂凑函数运用到数字签名中,一方面是为了加快签名速度,因为,如果直接对消息签名,当消息很长时需要对其分组,再对每组消息用 RSA 签名,这样签名效率相当低;另一方面,可以不泄露要签名的消息,这适用于一些特殊的签名场景。利用杂凑函数再结合四素数 RSA 算法,其签名过程如下:

a)用户 A 将要发送的消息 M 通过杂凑函数 H,产生消息摘要 $D = H(M)$。

b)用户 A 用私钥 d 对消息摘要进行签名 $S = D^d \bmod n$。

c)用户 A 将消息 M 和签名 S 一同发送给用户 B。

d)用户 B 接收到消息和签名后用 A 的公钥解密签名 S 得到 D,再用杂凑函数计算一次消息摘要 D',判断 D' 是否等于 D。若相同,则说明消息确实来自于 A,并且在传输途中没有被窜改;否则,很有可能消息不是来自于 A。

运用以上的签名方法,如果用户 A 想否认曾经发送消息 M 给用户 B,用户 B 只需把 A 的公钥和签名 S 一并出示给公证方,通过正确的计算方法就能证实用户 A 确实发送了消息 M;如果用户 B 伪造了一个消息 M',由于他不知道用户 A 的私钥,也就无法出示正确的签名给公证方。这样一来,通信双方都必须真实地反映通信情况,有效防止了一方抵赖的情况发生。

3.3　运用 Montgomery 模指数算法和中国剩余定理对签名过程优化

3.3.1　Montgomery 模指数算法

Montgomery 模指数算法是 Montgomery 模乘结合指数算法来求大数模幂运算的一种方法。Montgomery 模乘是一种将除法运算改为移位运算从而简化两数模乘的算法。这里,本文运用文献[17]中一种改进的 Montgomery 模乘——CIOS(coarsely integrated operand scanning)模乘法。设整数 $m = (m_{n-1} \cdots m_1 m_0)_b$($b$ 为进制数),$x = (x_{n-1} \cdots x_1 x_0)_b$,$y = (y_{n-1} \cdots y_1 y_0)_b$,$\gcd(m,b) = 1$,$R = b^n$,$m' = -m^{-1} \bmod b$($m^{-1}$ 为 m 模 b 的乘法

逆元)。其算法描述如下:

a) $A=0$。

b) 对 i 从 0 到 $n-1$ 执行:$A=A+x_i y, u_i=a_0 m' \bmod b, A=(A+u_i m)/b$。

c) 如果 $A \geqslant m$,则 $A=A-m$。

d) 返回 $A=xyR^{-1} \bmod m$。

再定义函数 mont(u,v)为 $uvR^{-1} \bmod m$,用上述 CIOS 算法进行计算,然后再结合指数算法,便可得到 $x^e \bmod m$ 的计算方法,其中 x 为整数,$1 \leqslant x < m, e=(e_t \cdots e_1 e_0)_2, e_t=1$。算法描述如下:

a) $x'=\text{mont}(x, R^2 \bmod m), A=R \bmod m$。

b) 对 i 从 t 到 0 执行:$A=\text{mont}(A,A)$,若 $e_i=1$ 则 $A=\text{mont}(A,x')$。

c) $A=\text{mont}(A,1)$。

d) 返回 $A=x^e \bmod m$。

经分析知,如果仅仅是求模乘运算,Montgomery 模乘并不能有速度上的提高,因为算法在预处理时还是要用到一般的模运算和模逆运算,但如果是模幂运算,其中含有 $(3\log_{10}e)/2$ 次模乘运算,这些预处理只需一次就能进行多次没有除法的 Montgomery 模乘运算,这将大幅度提高模幂运算的速度。

3.3.2 中国剩余定理

中国剩余定理又叫孙子定理,是中国古代求解一次同余式组的方法,也是数论中一个重要的定理。令 r 个整数 $m_1, m_2, \cdots, m_r$ 两两互素,$a_1, a_2, \cdots, a_r$ 是任意 r 个整数,则同余方程组 $x \equiv a_i \bmod m_i (1 \leqslant i \leqslant r)$ 的模 $M=m_1 m_2 \cdots m_r$ 有唯一解,其表达式为

$$x=\sum_{i=1}^{r} a_i M_i y_i (\bmod M)$$

其中:$M_i=M/m_i, y_i M_i \equiv 1 \bmod m_i, 1 \leqslant i \leqslant r$。

可见,中国剩余定理能够把高位宽大数的模幂运算转换为低位宽相对较小的数的模幂运算。

3.3.3 中国剩余定理的应用

运用中国剩余定理,对消息摘要 D 的数字签名可转换为以下运算过程:a) 计算 $m_p=D \bmod p, m_q=D \bmod q, m_r=D \bmod r, m_s=D \bmod s$;b) 计算 $d_p=d \bmod (p-1), d_q=d \bmod (q-1), d_r=d \bmod (r-1), d_s=d \bmod (s-1)$;c) 计算 $M_1=m_p^{d_p} \bmod p, M_2=m_q^{d_q} \bmod q, M_3=m_r^{d_r} \bmod r, M_4=m_s^{d_s} \bmod s$;d) 计算 $S=(M_1(qrs)^{p-1}+M_2(prs)^{q-1}+M_3(pqs)^{r-1}+M_4(pqr)^{s-1}) \bmod n$,即得出签名 S。

在上述计算过程中,先把传统的签名算法 $S=D^d \bmod n$ 转换为求解四个同余式:$S \equiv D^d \bmod p, S \equiv D^d \bmod q, S \equiv D^d \bmod r$ 和 $S \equiv D^d \bmod s$,再利用中国剩余定理进行求解。在求乘法逆元时,本文没有用扩展欧几里德算法,而是运用了费马小定理:对任何不被素数 p 整除的整数 A,恒有 $A^{p-1} \equiv 1 \bmod p$,可得 $A^{-1} \equiv A^{p-2} \bmod p$,巧妙地通过一个多项式运算来代替其中一个逆元的求解,进一步提高了运算效率。

4 签名过程中安全性分析

4.1 选择密文攻击

选择密文攻击是攻击者对 RSA 等公钥算法最常用的攻击,其攻击方法可分为明文破译、骗取仲裁签名和伪造合法签名。本文重点讨论后两者。

4.1.1 骗取仲裁签名

在有仲裁情况下,若攻击者有一非法消息 x 想让仲裁 T 签名,则它可任选一数 N,计算 $y \equiv N^e \bmod n$(e 是 T 的公钥),然后计算 $M=yx$ 发送给 T,T 将签名的结果 $M^d \bmod n$ 发送给攻击者,则有:$(M^d \bmod n)N^{-1} \equiv (yx)^d N^{-1} \equiv (x^d y^d N^{-1}) \equiv x^d N N^{-1} \equiv x^d \bmod n$,攻击者成功骗取 T 对 x 的签名。

4.1.2 伪造合法签名

攻击者可利用自己伪造的两个合法消息 x_1 和 x_2 来拼凑出所需要的 $x_3 \equiv (x_1 x_2) \bmod n$。攻击者如果得到用户 U 对 x_1 和 x_2 的签名 $x_1^d \bmod n$ 和 $x_2^d \bmod n$ 就可以计算出 x_3 的签名:$x_3^d \equiv ((x_1^d \bmod n)(x_2^d \bmod n)) \bmod n$。

由于本文的签名算法并不是对消息直接签名,而是运用杂凑函数先对消息进行压缩处理,再对消息摘要进行签名。由杂凑函数的不可逆性和随机性知,当给定一个 D,很难找到一个消息 M,使得 $H(M)=D$,从这个意义上来说,给攻击者造成了极大的难度,以上攻击方法趋于不可行。在实际应用过程中,为了安全起见,应该拒绝对随机的陌生文档签名。

4.2 出错攻击

文献[16]指出,用传统双素数 RSA 算法对明文 M 进行签名时,容易遭受出错攻击。签名方用自己的私钥计算 $C=M^d \bmod n$,使用中国剩余定理,C 可通过 $C_p=M^d \bmod p, C_q=M^d \bmod q$ 有效计算出来。假设错误发生在 C_p 时,也就是说计算了一个错误的值 $C_p' \neq C_p$,而 C_q 被正确计算出来,C_p' 和 C_q 就联合产生了一个错误的签名 C',这样一来有如下性质:$q=\gcd(C'^e - M \bmod n, n)$。

证明 根据中国剩余定理有

$$C'=C_p'(q^{-1} \bmod p)q+C_q(p^{-1} \bmod q)p(\bmod n),$$

所以

$$\begin{aligned}(C'^e - M \bmod n) \bmod q &= ((C_p'(q^{-1} \bmod p)q)^e + \cdots + \\ &(C_q(p^{-1} \bmod q)p)^e - M \bmod n) \bmod q = \\ ((C_q(p^{-1} \bmod q))^e - M \bmod n) \bmod q &= (C^e - M) \bmod q = 0\end{aligned}$$

而 C' 是一个错误的签名,所以 $(C'^e - M \bmod n) \bmod p \neq 0$,因而性质成立。

由以上证明过程可知,在四素数 CRT-RSA 签名过程中,若 C_p 出错,$\gcd(C'^e - M \bmod n, n)$ 求得的是另外三个素数 q、r 和 s 的乘积,并不能求得某一个素因子,也就无法分解模数 n,因此,四素数 CRT-RSA 签名能抵抗出错攻击。

4.3 连分数攻击

一个连分数可以表示成

$$q_0+\cfrac{a_1}{q_1+\cfrac{a_2}{q_2+\cfrac{a_3}{\cdots \cfrac{}{q_{m-1}+\cfrac{a_m}{q_m}}}}}$$

它经常被用于有理数或者无理数的逼近,数论中将 $a_1=1$ 的情况称为狭义连分数。在文献[11]中,作者提出当私钥 d 小于模数 $N^{1/4}$ 时,RSA 密码系统是不安全的,他的分析方法本质是利用了连分数中 Legendre 定理[18],通过展开一个已知的正有理数 f' 的第 i 级连分数,逼近所求的分数 f 的值。在文章的末尾,作者提出了两个开放性问题,其中之一是是否存在对小解密指数 CRT-RSA 的攻击。分析这个问题之前,来看两个定理[18]。

定理 2 设 f' 是小于 f 的一个近似估算值,有 $f'=f(1-\delta)$,

$\delta \geqslant 0$,当 m 是偶数且展开式 $\langle q_0, q_1, \cdots, q_m - 1\rangle < f' \leqslant \langle q_0, q_1, \cdots, q_m\rangle$;当 m 是奇数且 $\langle q_0, q_1, \cdots, q_m + 1\rangle < f' \leqslant \langle q_0, q_1, \cdots, q_m\rangle$,那么连分数算法是有效的。

也就是说,连分数算法是否成功依赖于 f' 和 f 之间的差值百分比 δ 的大小。

定理 3　当 $\delta < \dfrac{1}{\frac{3}{2} n_m d_m}$ 时,用连分数算法可以找到 f,其中 n_m 和 d_m 分别是 f 的分子和分母。

由改进的 RSA 算法列知,$ed \equiv 1 \bmod \varphi(n)$ 可得 $ed_p \equiv 1 \bmod (p-1)$,那么 $ed_p = k(p-1)+1$,其中 k 为正整数。两边同时除以 $pqrsd_p$,得到 $\dfrac{e}{pqrs} = \dfrac{k}{d_p}\left(\dfrac{p-1}{pqrs} + \dfrac{1}{kpqrs}\right)$。根据定理 2,令 $\dfrac{e}{pqrs} = \dfrac{k}{d_p}(1-\delta)$,而 $n = pqrs$,得到 $\delta = 1 - \dfrac{p-1}{n} - \dfrac{1}{kn}$。根据定理 3,要通过连分数算法找出 k 和 d_p 需要满足 $\delta < \dfrac{1}{\frac{3}{2} n_m d_m}$,即 $kd_p < \dfrac{2n}{3\left(n - p + 1 - \frac{1}{k}\right)}$。由于 n 和 p 都是大素数,所以分母中的 $1 - \dfrac{1}{k}$ 可忽略,再同时除以 n 可得:$kd_p < \dfrac{2}{3} \cdot \dfrac{1}{1 - \frac{1}{qrs}} \approx \dfrac{2}{3}$,那么 k 必须小于 1,这与 k 为正整数相矛盾,所以无法找到满足 $\delta < \dfrac{1}{\frac{3}{2} n_m d_m}$ 的情况,攻击无效。

5　混合加密体制

5.1　两种算法比较

1)密钥管理方面　AES 算法不及 RSA 算法。由于 RSA 是非对称密码体制,采用公开的形式分配加密密钥,所以对加密密钥的更新很容易,并且对不同的通信对象,只需要保密自己的解密密钥即可;AES 算法属于对称密码体制,要求通信前对密钥进行秘密分配,密钥的更换困难,而且对不同的通信对象,AES 需产生和保管不同的密钥。

2)加解密运算速度方面　AES 算法优于 RSA 算法。因为 AES 算法基于扩散和混淆原理,大多数的操作是字节的移位和替换,而且密钥最大是 256 位,可以利用软件或硬件进行高速实现;而 RSA 算法基于大整数分解的困难性进行设计,加解密的过程中需要进行大整数的模幂运算,这是相当耗时的操作,因此,RSA 算法不适合大批量的数据加密。

3)签名和认证方面　由于 AES 属于对称密码体制,加密和解密使用同一个密钥,所以,从原理上不能由 AES 实现数字签名;而 RSA 属于公钥密码体制,可以非常容易地实现数字签名和身份认证。

4)安全性方面　两者的安全性都很好,目前还未出现能够完全破译 AES 和 RSA 的有效方法。

5.2　具有保密性和认证性的密码体制

在本文优化算法的基础上,再结合两种算法的优势,提出一种既能高效安全传输数据又能进行身份认证的混合加密体制。信息传输流程如图 3 所示,具体步骤如下:

a)发送方 A 将原文信息 M 进行杂凑函数运算,得到一个消息摘要 D。

b)发送方 A 用自己的私钥 PVA,采用 RSA 算法对消息摘要 D 进行签名,得到 S。

c)发送方 A 用 AES 算法对原文信息 M 和数字签名 S 进行加密,得到加密信息 E。

d)发送方 A 用接收方 B 的公钥 PBB,采用 RSA 算法对 AES 的密钥 SK 加密,形成数字信封 DE,就好像将对称密钥 SK 装到了一个用收方公钥加密的信封里。

e)发送方 A 将加密信息 E 和数字信封 DE 一起发送给接收方 B。

f)接收方 B 收到数字信封 DE 后,首先用自己的私钥 PVB 解密数字信封,取出对称密钥 SK。

g)接收方 B 用对称密钥 SK 同 AES 算法解密加密信息 E,得到原文信息 M 和数字签名 S。

h)接收方 B 对数字签名进行验证,先用发送方 A 的公钥 PBA 解密数字签名 S,得到消息摘要 D。

i)接收方 B 同时将原文信息用同样的杂凑函数算出一个新的消息摘要 D'。

j)接收方 B 将两个消息摘要进行比较,如果两者相等,说明数据没有被窜改,是保密传输的,签名是真实的;否则拒绝该签名。

这样一来就可以保证信息在传输过程中的保密性、完整性以及不可否认性,同时也保证了对称密钥的安全传输。

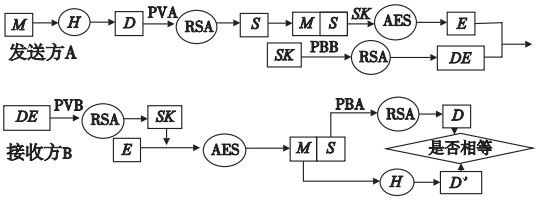

图3 AES与RSA混合加密体制

6　实验结果及比较分析

为了充分发挥 AES 效率优势,本文选用 10 轮 AES 算法,RSA 方面考虑到未来一段时间内的安全性,采用模数为 2 048 bit 的 RSA 算法。杂凑函数方面,由于 MD5、SHA1 已经被王小云等人破解,本文采用的是 SHA512[19]。本次实验的硬件环境为 i3-2350M CPU、3 GB 内存、360 GB 硬盘;操作系统为 Windows 7;开发工具为 Visual C++ 6.0。

6.1　AES 扩散混淆性测试

扩散和混淆是 Shannon 提出的设计密码体制的两种基本方法,其目的是为了抵抗敌手对密码体制的统计分析。所谓扩散就是让明文中的每一位影响密文中的许多位,这样可以隐蔽明文的统计特性;混淆就是将密文与密钥之间的统计关系变得尽可能复杂,使得敌手即使获取了关于密文的一些统计特性,也无法推出密钥。在本文的算法中,加密和解密使用了同一个矩阵,这是否会影响原算法的扩散和混淆特性,本文通过实验来说明。

首先测试其扩散性,对一个长 128 bit 的字符串进行 AES 加密,保证密钥不变,记录明文每变化一位密文变化的位数(由于篇幅原因,这里只给出明文变化三位的测试结果)。当明文改变一位时,所引起密文的改变如图 4 所示,原算法密文改变位数的范围是 65 ± 5,改进算法密文改变位数的范围是 64 ± 7;当明文改变两位时,所引起密文的改变如图 5 所示,原算法密文改变位数的范围是 63 ± 7,改进算法密文改变位数的

范围是 63 ± 5 ;当明文改变三位时,所引起密文的改变如图 6 所示,原算法密文改变位数的范围是 63 ± 7,改进算法密文改变位数的范围是 64 ± 5。

然后,测试其混淆性,保证明文不变,记录密钥每改变一位对密文的影响。当密钥改变一位时,所引起密文的改变如图 7 所示,原算法密文改变位数的范围是 63 ± 7,改进算法密文改变位数的范围是 63 ± 5;当密钥改变两位时,所引起密文的改变如图 8 所示,原算法密文改变位数的范围是 64 ± 7,改进算法密文改变位数的范围是 64 ± 7;当密钥改变三位时,所引起密文的改变如图 9 所示,原算法密文改变位数的范围是 64 ± 5,改进算法密文改变位数的范围是 63 ± 7。本文一共测试了 30 位明文和密钥变化的情况,密文改变的位数都在 64 位左右,这说明改进算法并没有影响原算法的扩散混淆特性。

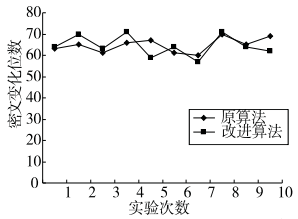

图3 明文改变一位时密文变化

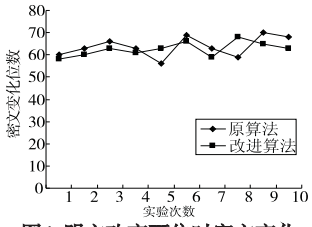

图4 明文改变两位时密文变化

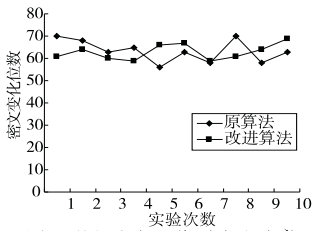

图5 明文改变三位时密文变化

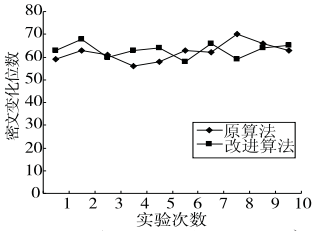

图6 密钥改变一位时密文变化

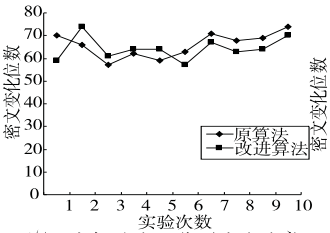

图7 密钥改变两位时密文变化

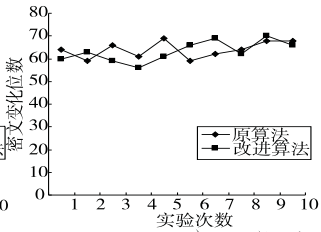

图8 密钥改变三位时密文变化

6.2 AES 加解密速率测试

为了使实验效果更明显,本文对 5 组长 128 bit 的字符串加解密 10 000 次,记录其耗时情况,如表 1 所示。从表中可以看出,改进后的 AES 算法,其加速度较原算法可提高 22%,解密速度较原算法可提高 26%,而且加密解密耗时相当。这说明改进的 AES 算法在运算效率上较原算法有一定优势,而且解决了原算法中解密相对于加密存在的时延问题。

表 1 加解密耗时

时间 /ms	原 AES 算法		改进 AES 算法	
	加密	解密	加密	解密
第一组	470	491	348	351
第二组	450	501	340	341
第三组	461	490	351	353
第四组	471	500	358	361
第五组	460	481	349	351
平均	462.7	486.3	355.2	358.4

6.3 RSA 签名速率测试

本文基于 C ++ 构建了一个较为完善的大数库,采用无符号长整型数组从低位到高位存储大数,也就是说,把一个大数用 2^{32} 进制表示,这样可以最大程度减少基本运算中循环的次数,提高运算效率。然后用文献[20]中一种改进的 Miller-Rabin 算法随机生成两个 1 024 位素数和四个 512 位素数,分别用传统双素数 RSA 算法、融入 CRT 的双素数 RSA 算法和文中的四素数 CRT-RSA 算法对三组 512 位消息摘要进行签名并记录其耗时情况。实验结果如表 2 所示。

表 2 三种签名算法耗时情况 /ms

时间 /ms	传统双素数 RSA 签名	双素数 CRT-RSA 签名	四素数 CRT-RSA 签名
第一组	3 432	996	311
第二组	3 621	1 107	338
第三组	3 568	1 059	329
平均	3 541	1 054	326

从表中可以看出,用 CRT 进行优化的双素数 RSA 签名效率是传统签名算法的 3.36 倍,与理论值 3.47 相接近。而采用 CRT 和 Montgomery 模指数算法进行优化的四素数 RSA 签名算法,其效率约是传统算法的 10.86 倍,是双素数 CRT-RSA 的 3.23 倍,这足以说明改进的 RSA 签名可以大幅度提高签名速度。

7 结束语

本文主要研究了目前较流行的对称加密算法 AES 和非对称加密算法 RSA,分析了其研究现状,并对不足之处加以改进,并且通过实验证实了改进的算法确实有一定优势。然后将其结合起来发挥各自的长处,构建了一个混合加密体制,下一阶段将会研究这一混合加密体制在具体实例中的应用。

参考文献:

[1] DAEMEN J, RIJMEN V. AES proposal: Rijndael[EB/OL]. [1999-10-05]. http://www.east.kuleuven.ac.be/~rijmen/rijndael.

[2] RIVEST R L, SHAMIR A, ADLEMAN L. A method for obtaining digital signatures and public key cryptosystems[J]. Communications of the Association for Computer Machinery, 1978, 21(2): 120-126.

[3] 吴文玲,冯登国. 分组密码工作模式的研究现状[J]. 计算机学报, 2006, 29(1): 22-25.

[4] 胡亮,袁巍,于孟涛,等. 单向性策略与 AES 密钥生成算法的改进[J]. 吉林大学学报:工学版, 2009, 39(1): 138-141.

[5] MOZAFFARI-KERMANI M, REYHANI-MASOLEH A. Fault detection structures of the S-boxes and inverse S-boxes for the advanced encryption standard[J]. Electronic Testing, 2009, 25(4): 225-245.

[6] 王兴刚. 基于 RSA 和 AES 混合算法的加密芯片设计[D]. 济南:济南大学, 2011.

[7] 高娜娜,宋丽华. AES 算法中密钥扩展和列混合单元的可重构设计[J]. 北京信息科技大学学报, 2012, 27(4): 51-55.

[8] RACHH R R, MOHAN A, ANAMI B S. Efficient implementations for AES encryption and decryption[J]. Circuits, Systems, and Signal Processing, 2012, 31(5): 1765-1785.

[9] 代大勇. AES 算法及其 DSP 实现[D]. 哈尔滨:哈尔滨工业大学, 2008.

[10] 程桂花,齐学梅,罗勇龙. AES 算法中的多项式模运算及其性能分析[J]. 计算机技术与发展, 2010, 20(9): 115-118.

[11] WIENER M J. Cryptanalysis of short RSA secret exponents[J]. IEEE Information Theory Society, 1990, 36(3): 553-558.

[12] BONEH D, DURFEE G. Cryptanalysis of RSA with private key d less than $N^{0.292}$[J]. IEEE Information Theory Society, 2000, 46(4): 1339-1349.

(下转第 1198 页)

签密的验证等式为 $H_2\{m',(w_a u_a y^{h'_1} g^{h'})^{s'},\mathrm{ID}_a,u_a,r_2\}=h'$。为构造 σ' 满足验证等式,设 $r_1=A_2$,$h'=H_2\{A_1,g^{A_2},\mathrm{ID}_a,A_3,A_4\}$,$s'=A_5$,$w_a=A_6$,$c'=A_7$,其中 A_i 各项均为伪造参数。

只要构造 A_i 满足 $H_2\{m',(A_6A_3y^{H_1\{\mathrm{ID}_a,A_6\}}g^{h'})^{A_5},\mathrm{ID}_a,A_3,A_4\}=h'$等式,即可在不解 DL 难题下,实现对签名的伪造攻击。通过分析 h'的参数构成,只要同时满足 $(A_6A_3y^{H_1\{\mathrm{ID}_a,A_6\}}g^{h'})^{A_5}=g^{A_2}$和 $A_7\oplus H_3((w_au_ay^{h'_1}g^{h'})^{A_5(z_b+t_b)})=A_1$ 即可实现签名的伪造攻击。

首先对 $E((w_au_ay^{h'_1}g^{h'})^{A_5(z_b+t_b)},A_7)=A_1$ 进行分析,$A_7=D((A_6A_3y^{H_1\{\mathrm{ID}_a,A_6\}}g^{h'})^{A_5(z_b+t_b)},A_1)$,在定义 3、4 所述的安全模型下,虽然 (z_a+t_a)不能查询,但 (z_b+t_b)可以在攻击的准备阶段完成查询。所以 A_7 可构造,即 c'可构造。

然后对 $(A_6A_3y^{H_1\{\mathrm{ID}_a,A_6\}}g^{h'})^{A_5}=g^{A_2}$ 进行分析,$(A_6A_3y^{H_1\{\mathrm{ID}_a,A_6\}}g^{H_2\{A_1,g^{A_2},\mathrm{ID}_a,A_3,A_4\}})^{A_5}=g^{A_2}$。由于 A_5 为伪造参数,攻击者 At 可以任意选择 $A_5\in Z_q^*$,确定 A_5 后,可对上式进行化简,得到 $y^{H_1\{\mathrm{ID}_a,A_6\}}g^{H_2\{A_1,g^{A_2},\mathrm{ID}_a,A_3,A_4\}}=A_6^{-1}A_3^{-1}g^{-A_5A_2}$。对化简后的等式分析可见,$A_6$、$A_3$ 参数均出现在指数位置,且分别在 H_1、H_2 内部,A_6、A_3 参数不独立,无法分离变量;A_5 已经确定,不能改变;A_2 参数出现在指数位置,且出现在 H_2 函数的第 2 项参数的指数位置,在不解 CDH 问题和 H_2 函数强碰撞的前提下,无法分离变量;A_1、A_4 参数出现在指数位置的 H_2 函数第 1、5 项,同样,在不解 CDH 问题和 H_2 函数强碰撞的前提下,无法分离变量。所以,该等式的 $A_i(i=1,2,\cdots,6)$变量的分离都必须解决 CDH 问题,其中 A_1、A_3、A_4、A_6 的分离还要额外解决 H_1、H_2 的强碰撞问题,哈希碰撞问题的目前最新进展为王小云教授所做的工作。但只要选择安全的哈希函数,目前没有有效的破解方案,即使解决了 H_1、H_2 的强碰撞问题,仍然面临 DL 或 CDH 难题。

通过上述分析可知,本文所提出的签密方案在签名方面是安全可靠的,本文签密方案满足选择消息攻击下的不可伪造性。

5 结束语

本文提出了一个新的无双线性对的无证书签密方案,基于离散对数难题,避免双线性配对计算,相比采用双线性对的签密方案提高了计算效率。在安全分析及证明方面,避免采用 ROM 的证明方法,通过数学构造分析的方式,分析了在不解决 DL 难题的前提下对签密方案的攻击,并证明了破解本文所提出的签密方案的必要条件是解决 DL 难题,不存在不解决 DL 难题攻击本文签密方案的多项式时间算法,本文签密方案比现有的签密方案更加安全可靠。

参考文献:

[1] ZHENG Yu-liang. Digital signcryption or how to achieve cost (signature and encryption) < cost(signature) + cost(encryption)[C]// Proc of CRYPTO. Berlin:Springer, 1997: 165-179.

[2] AL-RIYAMI S S, PATERSON K G. Certificateless public key cryptography[C]//Proc of ASIACRPT. Berlin: Springer, 2003: 452-473.

[3] ARANHA D, CASTRO R, LOPEZ J, *et al*. Efficient certificateless signcryption[EB/OL]. (2010-05-12)[2013-03-15]. http://sbseg2008.inf.ufrgs.br/proceedings/data/pdf/st03_01_resumo.pdf.

[4] WU Chen-huang, CHEN Zhi-xiong. A new efficient certificateless signcryption scheme[C]//Proc of International Symposium on Information Science and Engineering. 2008: 661-664.

[5] BARRETO P L, DEUSAJUTE A M, CRUZ E C, *et al*. Toward efficient certifi-cateless signcryption from (and without) bilinear pairings[EB/OL]. (2010-05-12)[2013-03-15]. http://www.redes.unb.br/ceseg/anais/2008/data/pdf/st03_03_artigo.pdf.

[6] SHARMILA D S, VIVEK S S, PANDU R C. On the security of certificateless signcryption schemes[EB/OL]. (2010-05-12)[2013-03-15]. http://eprint.iacr.org/2009/298.pdf.

[7] LI Fa-gen, MASAAKI S, TSUYOSHI T. Certificateless hybrid signcryption[C]// Proc of International Conference on Information Security Practice and Experience. Berlin: Springer, 2009: 112-123.

[8] 朱辉, 李晖, 王育民. 不使用双线性对的无证书签密方案[J]. 计算机研究与发展, 2010, 47(9): 1587-1594.

[9] 许春根, 张傲红, 韩牟, 等. 一种基于离散对数问题的无证书代理签名方案[J]. 南京理工大学学报: 自然科学版, 2010, 34(6): 733-737.

[10] 刘文浩, 许春香. 无双线性配对的无证书签密方案[J]. 软件学报, 2011, 22(8): 1918-1926.

[11] YU Yong, MU Yi, WANG Gui-lin, *et al*. Improved certificateless signature scheme provably secure in the standard model[J]. IET Information Security, 2012, 6(2): 102-110.

[12] JING Xiao-fei. Provably secure certificateless signcryption scheme without pairing[C]//Proc of International Conference on Electronic and Mechanical Engineering and Information Technology. 2011: 4753-4756.

[13] SWAPNA G, REDDY V. An efficient id-based public verifiable signcryption sheme[J]. International Journal of Cryptography and Security, 2013, 3(1): 41-46.

[14] 何俊杰, 焦淑云, 祁传达. 一个基于身份的签密方案的分析与改进[J]. 计算机应用研究, 2013, 30(3): 913-916.

[15] SHARMILA S S D, VIVEK S S, PEADHAN V K. Efficient certificateless online/offline signature with tight security[J]. Journal of Internet Services and Information Security, 2013, 1(1/2): 115-137.

(上接第 1194 页)

[13] 贺克英. 改进的 RSA 算法实现研究[D]. 成都: 电子科技大学, 2010.

[14] PAIXAO C A M. An efficient variant of the RSA cryptosystem[EB/OL]. [2010-10-02]. http:// www.ime.usp.br/~capaixao/paper.pdf.

[15] 杨波. 现代密码学[M]. 北京: 清华大学出版社, 2007.

[16] 贵晓飞, 胡捍英. CRT-RSA 算法安全性分析[J]. 微计算机信息, 2009, 25(3): 36-38.

[17] 王安. RSA 公钥密码算法的快速实现[D]. 济南: 山东大学, 2008.

[18] 陈恭亮. 信息安全数学基础[M]. 北京: 清华大学出版社, 2011.

[19] STALLINGS W. Cryprography and network security principles and practice[M]. 5th ed. Beijing: Publishing House of Eletronics Industry, 2011.

[20] COUVEIGNES J M, EZOME T, LERCIER R. A faster pseudo-primality test[J]. Rendiconti del Circolo Matematico di Palermo, 2012, 61(2): 261-278.