RSA 密码协处理器的实现

李树国,周润德,冯建华,孙义和
(清华大学微电子学研究所,北京 100084)

摘 要: 密码协处理器的面积过大和速度较慢制约了公钥密码体制 RSA 在智能卡中的应用. 文中对 Montgomery 模乘算法进行了分析和改进,提出了一种新的适合于智能卡应用的高基模乘器结构. 由于密码协处理器采用两个 32 位乘法器的并行流水结构,这与心动阵列结构相比它有效地降低了芯片的面积和模乘的时钟数,从而可在智能卡中实现 RSA 的数字签名与认证. 实验表明:在基于 0.35μm TSMC 标准单元库工艺下,密码协处理器执行一次 1024 位模乘需 1216 个时钟周期,芯片设计面积为 38k 门. 在 5MHz 的时钟频率下,加密 1024 位的明文平均仅需 374ms. 该设计与同类设计相比具有最小的模乘运算时钟周期数,并使芯片的面积降低了 1/3. 这个指标优于当今电子商务的密码协处理器,适合于智能卡应用.

关键词: 模乘器; 智能卡; 公钥; 模乘
中图分类号: TP332.2 **文献标识码**: A **文章编号**: 0372-2112 (2001) 11-1441-04

Implementation for RSA Cryptography Coprocessor

LI Shu-guo, ZHOU Run-de, FENG Jian-hua, SUN Yi-he
(Institute of Microelectronics, Tsinghua University, Beijing 100084, China)

Abstract: The area and speed of cryptography coprocessor impede the application of public-key cryptography RSA for smart card. A new VLSI architecture of high-radix modular multiplier to compute RSA public-key cryptosystem using our modified Montgomery algorithm is proposed. With TSMC 0.35μm CMOS technology models, a 1024-bit RSA cryptography coprocessor based on our proposed VLSI architecture have been implemented. Its simulation results show that the time to calculate 1024-bit modular multiplication is about 1216 clock cycles and the gate count of the coprocessor is about 38k. At a clock rate of 5MHz it will take about 374ms to encrypt 1024-bit message on average. Compared with previous works our proposed architecture can achieve good performance in chip area and speed, therefore it is well suited to smart cards.

Key words: modular multiplier; smart card; public-key; modular multiplication

1 引言

随着智能卡的日益普及,智能卡交易中的数据安全变得越来越重要. 由于公钥密码体制 RSA[1] (Rivest, Shamir, Adleman) 解决了数字签名、信息验证和身份认证,因此智能卡采用公钥密码体制的 RSA 实施数据加密越来越必要. 但是,智能卡采用公钥密码体制 RSA 进行加密目前存在两个主要问题: (1) RSA 密码协处理器的 VLSI(Very Large Scale Integration) 实现面积过大 (2) RSA 密码协处理器的模幂乘运算速度较低.

大数模乘运算 $AB \bmod N$ 是公钥密码体制 RSA 的核心运算. 大数模乘运算所带来了极其高的硬件复杂性和低效率性,已成为制约公钥密码体制发展和应用的瓶颈. Montgomery 算法[2]是一种免除法的快速模乘运算方法,文献[3～5]基于心动阵列结构实现了 Montgomery 的模乘算法,但心动阵列结构的 VLSI 实现面积较大,不能用于面积较小的智能卡. 本文分析和改进大数模乘的 Montgomery 算法,提出了一种新的高基模乘器结构. 这种结构与心动阵列结构相比不仅降低了芯片面积,而且还减少了模幂乘运算的时钟周期数,适合于智能卡应用.

2 改进的 Montgomery 算法

2.1 基本的模乘 AB mod N 运算

RSA 加密算法[1]是目前在理论和实际应用中较为成功的一种公钥密码体制,它的安全性是基于数论中大整数分解为素数因子的困难性上. 它有一对密钥,即公钥或加密密钥 (e, N) 和私钥或解密密钥 (d, N).

对明文 m,其加密过程: $c \equiv E(m) = m^e \bmod N$,式中 c 表示密文

而解密过程: $m \equiv D(c) = c^d \bmod N$, m 表示明文

由 Euler 定理[1]可以证明加/解密过程的一致性. RSA 算法加密/解密过程实质上就是一个计算模幂 $m^e \bmod N$ 或 $c^d \bmod N$ 的过程. 但是由于 m, e, c, d, N 等操作数大于 1024 比

收稿日期:2000-12-05;修回日期:2001-05-16
基金项目:国家自然科学基金重大项目(No. 59995550-1)

特,直接的模幂运算已不可能,必须先将其分解为基本的大数模乘运算 $AB \bmod N$. Montgomery 算法[2]正是为解决大数模乘运算 $AB \bmod N$ 而提出的.

2.2 原始的 Montgomery 模乘算法

设 N 为模数且 $N>1$, R 是与 N 互素的一个基,通常, $R=2^u$, u 是 N 的位数; R^{-1} 和 N 满足 $0<R^{-1}<N$, $0<N'<R$, $RR^{-1}-NN'=1$,即 $RR^{-1}\equiv 1 \pmod N$ 或 $NN'\equiv -1 \pmod N$; 对给定大整数 T,且 $0 \le T<RN$

Montgomery 算法如下[2]:

function REDC (T)

$m \leftarrow (T \bmod R) N' \bmod R$

$t \leftarrow (T+mN)/R$

if $t \ge N$ then $t-N$ else return t

上述算法从表面上看仅有两次大数乘法 TN 和 mN,但由于模乘运算时 $T=AB$, $0 \le A<N$, $0 \le B<N$,所以算法共进行三次大数乘法运算. 当 A, B 和 N 都为 1024 位以上的大整数时,大数相乘给硬件实现带来了困难,因此必须对大数进行分解. 另外,由于算法的返回结果是 Montgomery 积 $ABR^{-1} \bmod N$,而不是模乘积 $AB \bmod N$,所以使用时还应消除 Montgomery 积的常数项 R^{-1} 而变为模乘积.

2.3 改进的 Montgomery 模乘 VLSI 实现算法

Koç[6]给出了一种改进的 Montgomery 的软件实现算法 FIPS[6],但由于 Koç 的 FIPS 算法是基于单处理器的,用于 VLSI 实现时其并行执行度较低,因此将其改进为适合于 VLSI 实现的高并行度算法. 改进后算法的并行度比原来提高了近 1 倍,改进后的算法如下:

设 A, B 分别为 s 位 r 进制整数,即 $A=(a_{s-1}\ a_{s-2}\cdots a_1\ a_0)$, $B=(b_{s-1}\ b_{s-2}\cdots b_1\ b_0)$,同时,也设模 N 为 s 位 r 进制整数, $N=(n_{s-1}\ n_{s-2}\cdots n_1\ n_0)$,且 $R=r^s$ 则有 $N<R$, $n_0\ n_0' \bmod r=-1$,并使 $A<N$, $B<N$,改进后的算法 MonPro(A, B) 如下:

$n'[0] := -n[0]^{-1} \bmod r$ // 求 n_0 的模逆

$S := 0$

// Part A:计算中间结果 $m[i]$

$A.1$ for $i=0$ to $s-1$

$A.2$ for $j=0$ to $i-1$

$A.2.1$ $S := S+a[j]b[i-j]+m[j]n[i-j]$

$A.3$ $S := S+a[i]b[0]$

$A.4$ $m[i] := S\ n'[0] \bmod r$

$A.5$ $S := S+m[i]n[0]$

$A.6$ $S := S/r$ // 右移一个 r 进制位

// Part B:计算最终结果并存于 $m[i]$中

$B.1$ for $i=s$ to $2s-1$

$B.2$ for $j=i-s+1$ to $s-1$

$B.2.1$ $S := S+a[j]b[i-j]+m[j]n[i-j]$

$B.3$ $m[i-s] := S \bmod r$

$B.4$ $S := S/r$ // 右移一个 r 进制位

// Part C:调整 Montgomery 积到 $[0, N)$

$C.1$ $t0 := S \bmod r$ // $t0$ 是一个 r 进制位

$C.2$ $Cy=1$

$C.3$ for $j=0$ to $s-1$

$C.3.1$ $(Cy, b[j]) := m[j]+\text{not}(n[j])+Cy$

// Cy 为进位位,随进位而变

$t0 := t0+\text{not}[0]+Cy$

$C.4$ if $t0=0$

then return $(b[s-1]b[s-2]\cdots b[1]b[0])$

else return $(m[s-1]m[s-2]\cdots m[1]m[0])$

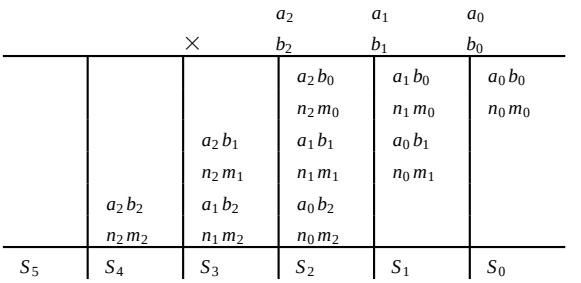

图 1 FIPS 模乘的方法

图 1 为 $s=3$ 时 FIPS 方法的一个实例,该实例可对上述算法进行说明. 该算法分 Part A、Part B 和 Part C 三部分, Part A 对应于图 1 中点划线右侧的计算,即计算乘积结果的低位 s 个字; Part B 对应于点划线左侧的计算,即计算乘积结果的高位 s 个字. 为节省存储空间高位 s 个字的存储借用了存储变量 m,最后 Montgomery 积存储在 $(m[s-1]m[s-2]\cdots m[1]m[0])$. 由于 Montgomery 积只能保证在 $[0, 2N)$ 的范围,所以还应将其调整到 $[0, N)$ 的范围内. Part C 正是完成该调整功能的. 上述算法的计算瓶颈是乘法的次数. Part A 需要进行 s^2+2s 次乘法, Part B 需要进行 s^2-s 次乘法,共进行 $2s^2+s$ 次乘法. Part C 需要进行 s 次加法以调整模乘积由 $[0, 2N)$ 到 $[0, N)$. 算法的实质是把原始的 Montgomery 算法的 3 次大数乘分解为 $2s^2+s$ 次小整数乘,以利于 VLSI 实现.

3 模乘器的结构

3.1 k 值的确定

模乘器是 RSA 密码协处理器核心运算部件. 模乘运算 $AB \bmod N$ 速度取决于模乘运算的时钟周期数,所以模乘器设计目标应在规定的面积下尽可能降低模乘运算的时钟周期数. 在 2.3 的 VLSI 实现算法中,由于 A, B, N 都是 r 进制整数,因此称 r 为基,而通常取 $r=2^k$. 若 $r=2^k$ 且 $k \ge 16$,则称 r 为高基. 基于高基的模乘器就为高基模乘器. 在本设计中,大数 A, B, N 各为 u 个二进制位,从数据的安全考虑[7],确定取 $u=1024$ 比特. 这样 A, B, N 就可以表示成由 $s=u/k$ 个字组成的多精度数, $A=(a_{s-1}, a_{s-2}, \cdots a_i \cdots a_1\ a_0)_r$,而 $a_i=(\bar{a}_{k-1}, \bar{a}_{k-2}, \cdots, \bar{a}_1\ \bar{a}_0)$. 即每个 $a_i\ (0 \le i<s)$ 可表示 k 个二进制位. k 值越大,硬件的 VLSI 实现规模也就越大.

在 2.3 的 VLSI 实现算法中,当 $s=u/k$ 时,总乘法次数 $2s^2+s$ 就变为 $2(u/k)^2+u/k$. 当 u 固定时,乘法次数 $2(u/k)^2+u/k$ 将随着 k 的增大而减少,相应的运算时间也就越少,这是我们所希望的. 但是,由于 k 值与 VLSI 的硬件实现规模成正比, k 值过大会导致 VLSI 的实现面积和时延较大. 因

此,k 的取值应在面积的约束下尽可能的降低运算的时钟数.由于采用 TSMC 的 0.35μm 工艺进行流片,RSA 密码协处理器的设计面积应限制在 40k 门以内(1k = 1000 门),才能满足智能卡芯片面积的要求.

如果选取 $k = \sqrt{u}$,那么 $2(u/k)^2 + u/k$ 就变为 $2u + \sqrt{u}$.取 u 的平方根的理由是:在忽略时 $\sqrt{u}$(当 $u \geq 1024$ 时,与 u 相比很小),乘法次数就从非线性的 u^2 变为线性的 u,这种变化对降低运算时钟数很有利.当 $k = \sqrt{u}$ 时,基于 TSMC 的 0.35μm 的标准单元库进行综合,结果表明密码协处理器硬件面积约为 38k 门,不超过密码协处理器面积 40k 门的上限.若再增加 k 的取值,在相同的实验条件下进行综合,密码协处理器模乘器硬件面积超过 40k 门.因此,设计中我们确定 $k = \sqrt{u}$.

3.2　模乘器的结构

由于确定了 $u = 1024$ 比特,$k = \sqrt{u} = 32$,那么基 $r = 2^k = 2^{32}$,所以用 32 位的乘法器来实现 1024 位的模乘运算.在 2.3 的 VLSI 实现算法中,Part A 和 Part B 各含有共同的乘积项 $a[j]b[i-j]$ 和 $m[j]n[i-j]$,由于这两个乘积项无数据相关,因此,可用两个 32 位乘法器同时并行地进行乘法运算如图 2 所示,于是在一个时钟周期内可完成两次乘法运算.

在 2.3 的 VLSI 实现算法 Part A.2.1 中,由于 $a[j]b[i-j]$ 和 $m[j]n[i-j]$ 两项可并行执行,这样,完成 $a[j]b[i-j]$ 和 $m[j]n[i-j]$ 的 $s^2 - s$ 次乘仅需 $(s^2 - s)/2$ 个时钟周期.而其它三个乘积项 $a[i]b[0]$、$Sn[0]$ 和 $m[i]n[0]$ 之间存在两次数据相关,即 $a[i]b[0]$ 相关 $Sn[0]$ 和 $Sn[0]$ 相关 $m[i]n[0]$,依据图 2 的三级流水结构,每次相关需要等待 3 个时钟周期,故两次相关共需 6 个时钟周期.又由于 $a[i]b[0]$、$Sn'[0]$ 和 $m[i]n[0]$ 需要循环 s 次,所以完成这三个乘积项的累加需要 $6s$ 个时钟周期.简言之,Part A 的乘加运算需要 $6s + (s^2 - s)/2$ 个时钟周期,即 $(s^2 + 11s)/2$ 个时钟周期.

在 2.3 的 VLSI 实现算法 Part B.2.1 中,仅存在可并行执行的乘积项 $a[j]b[i-j]$ 和 $m[j]n[i-j]$,所以,$(s^2 - s)$ 次乘仅需 $(s^2 - s)/2$ 个时钟周期.而在 Part C 中,将模乘积调整到 $[0, N)$ 应进行 s 次加法,还需 s 个时钟周期.因此,2.3 算法的 Part A, B, C 三项所耗的时钟数之和为 $s^2 + 6s$ 或者 $u + 6\sqrt{u}$ 个时钟周期.(将 $s = u/k$, $k = \sqrt{u}$ 代入式 $s^2 + 6s$ 得 $u + 6\sqrt{u}$)

在 2.3 的 VLSI 实现算法 Part A.4 中,由于 $Sn[0]$ 这 s 次乘积并没有计入累加和 S 中,累加和应为 $2s^2 + s - s = 2s^2$ 次乘积之和,因此,用作累加的加法器位宽至少应大于 $\log_2(2s^2 2^{64})$,而 $s = u/k = \sqrt{u} = 32$,所以,$\log_2(2s^2 2^{64}) = 75$,于是选择用于累加的加法器位宽为 76 位.(见图 2)

模乘器的数据通路采用三级流水结构,以增强模乘器的并行性.即 mul32 = > adder64 = > adder76,第一级为两个 32 乘法器并行执行,第二级一个 64 位的加法器累加两个 64 位的积并产生一位进位 Cy,第三级一个 76 位的加法器求总的累加和.模乘器的控制通路采用状态机模型控制循环叠代以及模乘器与存储器之间的数据交换.总之,模乘器完成一次模乘运算需要 $u + 6\sqrt{u}$ 个时钟周期.当 $u = 1024$ 比特时,一次模乘运算需要 1216 个时钟周期.

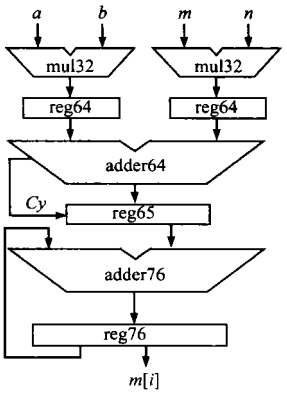

图 2　RSA 模乘器 Monpro

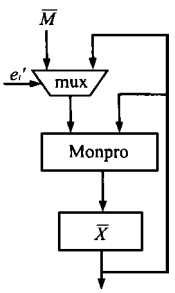

图 3　RSA 加密处理器

4　RSA 密码协处理器

模幂 $M^e \bmod N$ 硬件实现算法如下:$R = r^s = 2^{ks}$

```
function MonExp(M, e, N, R)        /* N 是奇数 */
1: M̄ := M·R mod N
2: x̄ := 1·R mod N
3: for i = u - 1 downto 0
4:     x̄ := MonPro(x̄, x̄)
5:     if (e_i = 1) then x̄ := MonPro(M̄, x̄)
6:  x := MonPro(x̄, 1)
7:  return x
```

模幂算法从左到右扫描 $e = (e_{u-1} \cdots e_i \cdots e_0)$ 来调用图 2 中的 RSA 模乘器 MonPro.由于 Montgomery 积不是模乘积,因此步骤 1、2、6 用来消去 Montgomery 积中的 R^{-1} 乘积项使之变为模乘积.模幂算法的 VLSI 实现就是 RSA 密码协处理器,如图 3 所示.模幂算法中的 e_i 和图 3 中的 e_i 关系是:当 $e_i = 0$ 时,$e_i' = 0$,即只进行一次模乘运算,当 $e_i = 1$ 时,$e_i' = 01$,进行两次模乘运算.

在平均的情况下,对任意的 i,$e_i = 1$ 或 $e_i = 0$ 的概率各半,所以平均需进行 1.5 次模乘运算,则完成模幂运算所需的时钟周期数:$1.5u(s^2 + 6s) = 1.5u^2 + 9u\sqrt{u}$

在最坏的情况下,对任意的 i,所有的 $e_i = 1$,全都进行 2 次模乘运算,则完成模幂运算所需的时钟周期数:$2u(s^2 + 6s) = 2u^2 + 12u\sqrt{u}$ $(s = u/k, k = \sqrt{u})$.

基于 5MHz 的工作时钟,加密 $u = 1024$ 位,平均执行时间为:$1.5 \times 1024 \times (s^2 + 6s)/(5 \times 10^6) = 1.5 \times 1024 \times (u + 6\sqrt{u})/(5 \times 10^6) = 374\text{ms}$

最坏执行时间为:$2 \times 1024 \times (s^2 + 6s)/(5 \times 10^6) = 2 \times 1024 \times (u + 6\sqrt{u})/(5 \times 10^6) = 498\text{ms}$

5　实验结果

用 Verilog HDL 实现了 1024 位 RSA 密码协处理器,用 Cadence 工具 Verilog-XL 进行仿真,验证了加/解密 $M \equiv M^{ed} \bmod N$ 的一致性和正确性.基于 0.35μm TSMC 标准单元库,用 Synopsys 工具进行综合,实验结果表明:RSA 密码协处理器共用 38k 门,它完成一次 1024 位模乘运算需要 1216 个时钟周期.它的最大时延为 32 位乘法器的组合逻辑时延,其值为 28ns,

所以 RSA 密码协处理器最高可允许 35MHz ,满足智能卡最高 20MHz 的工作频率. 在基于外部 5MHz 的工作时钟下 ,RSA 密码协处理器加密 1024 位的明文平均需要 374ms ,这与德国西门子智能卡 simens SLE66CX160S 加密器[8]相比面积减少了 1/3 ,而速度提高 1 倍 ,见表 1.

表 1 与西门子智能卡 siemens SLE66CX160S 性能之比

作者	年	门数	每 1k 位时钟数	加密位数	加密 1k 位
siemens[8]	2000	55k	2700 个	1024	880ms
本文	2001	38k	1216 个	1024	374ms

6 结论

本文改进了 Montgomery 实现算法 ,提出了一种新的的高基模乘器结构 ,实现了公钥密码体制 RSA 的 1024 位密码协处理器. 它的设计面积为 38k 门 ,时钟频率可达 35M. 在 5MHz 的时钟下 , RSA 密码协处理器加密 1024 位的明文平均需要 374ms ,这些指标优于当今电子商务的密码协处理器 ,适合于智能卡应用. 同时 ,这也表明我国的智能卡交易可采用自主产权 RSA 密码协处理器 ,这对我国的数据安全来说具有十分重要的意义.

参考文献:

[1] R L Rivest ,A Shamir ,L Adleman. A method for obtaining digital signatures and public-key cryptosystem [J]. Communications of ACM ,1978 , 21(2) :120 - 126.

[2] P L Montgomery. Modular multiplication without trial division [J]. Mathematics of computation ,1985 ,44(170) :519 - 521.

[3] P S chen ,S A Hwang ,C W Wu. A systolic RSA public key cryptosystem [A]. IEEE international Symposium on Circuits and Systems [C] , Atlanta ,1996 :4-408 - 4-411.

[4] Jyh Huei Guo ,Chin liang Wang ,Hung-chin Hu. Design and implementation of an RSA public-key cryptosystem [A]. Proceedings of the IEEE International Symposium on Circuits and Systems [C] ,Orlando , 1999 :Ⅰ-504 - Ⅰ-507.

[5] Ching chao Yang ,Tiam-sheuan ,Chein-wei Jen. A new RSA Cryptosystem hardware design based on montgomery 's algorithm [J]. IEEE transactions on circuits and systems- Ⅱ. ,1998 ,45(7) :908 - 913.

[6] C K KoC ,T Acar. Analyzing and comparing Montgomery multiplication algorithms [J]. IEEE Micro ,1996 ,16(3) :26 - 33.

[7] C K KoC ,C Y Huang. Bit-level systolic array for modular multiplication [J]. Journal of VLSI Signal Processing ,1991 ,3 :215 - 223.

[8] H Handschuh ,P Paillier. Smart card Crypto-Coprocessor for public-key cryptography [A]. Smart card research and applications [C]. Berlin : Springer ,2000 :372 - 379.

作者简介:

李树国 男. 1963 年出生于山东临沂. 现清华大学微电子所博士后 ,主要从事 ASIC 设计、智能 IC 卡、加密算法及计算机体系结构研究. 1999 年于西北工业大学计算机科学系获博士学位.

周润德 男. 1945 年出生于上海市. 清华大学微电子所教授 ,博导 ,主要从事计算机 RISC 体系结构、高速低电压低功耗 CMOS 电路设计、加密算法及 VLSI 设计方法学的研究. 1986 年在美国波士顿大学电机系获电子工程博士学位.

冯建华 男. 1963 年出生于山东省郓城. 清华大学微电子所博士后 ,主要从事 ASIC 设计及可测试性设计等研究.

孙义和 男. 1945 年出生于安徽省合肥市. 清华大学微电子所教授 ,博导 ,主要从事 ASIC 设计及可测试性研究.