支持国产密码算法的高速PCIe密码卡的设计与实现

赵 军[①②] 曾学文*[①②③] 郭志川[①②③]

[①](中国科学院声学研究所国家网络新媒体工程技术研究中心 北京 100190)

[②](中国科学院大学电子电气与通信工程学院 北京 100190)

[③](北京中科视云科技有限公司 北京 100190)

摘 要：密码卡在信息安全领域发挥着重要作用，但当前密码卡存在性能不足的问题，难以满足高速网络安全服务的需要。该文提出一种基于MIPS64多核处理器的高速PCIe密码卡的设计与系统实现方法，支持SM2/3/4国产密码(GM)算法以及RSA, SHA, AES等国际密码算法，系统包括硬件模块，密码算法模块，主机驱动模块和接口调用模块；对SM3的实现提出一种优化方案，性能提升了19%；支持主机以Non-Blocking方式发送请求，单进程应用即可获得密码卡满载性能。该卡在10核CPU下SM2签名和验证速度分别为18000次/s和4200次/s, SM3杂凑速度2200 Mbps, SM4加/解密速度8/10 Gbps，多项指标达到较高水平；采用1300 MHz主频16核CPU时，SM2/3的性能指标提高1倍，采用48核CPU时SM2签名速度可达到10^5次/s。

关键词：密码卡；PCIe总线；国产密码算法；非阻塞

中图分类号：TP393.08 **文献标识码**：A **文章编号**：1009-5896(2019)10-2402-07

DOI: 10.11999/JEIT190003

Design and Implementation of High Speed PCIe Cipher Card Supporting GM Algorithms

ZHAO Jun[①②] ZENG Xuewen[①②③] GUO Zhichuan[①②③]

[①](*National Network New Media Engineering Research Center*, *Institute of Acoustics*, *Chinese Academy of Sciences*, *Beijing* 100190, *China*)

[②](*School of Electronic*, *Electrical and Communication Engineering*, *University of Chinese Academy of Sciences*, *Beijing* 100190, *China*)

[③](*Beijing Zhongke Vision Cloud Technology Co.*, *Ltd.*, *Beijing* 100190, *China*)

Abstract: Cipher cards play an important role in the field of information security. However, the performance of cipher cards are insufficient, and it is difficult to meet the needs of high-speed network security services. A design and system implementation method of high-speed PCIe cipher card based on MIPS64 multi-core processor is proposed, which supports the GM algorithm SM2/3/4 and international cryptographic algorithms, such as RSA, SHA and AES. The implemented system includes module of hardware, cryptographic algorithm, host driver and interface calling. An optimization scheme for the implementation of SM3 is proposed, the performance is improved by 19%. And the host to send requests in Non-Blocking mode is supported, so a single-process application can get the cipher card's full load performance. Under 10-core CPU, the speed of SM2 signature and verification are 18000 and 4200 times/s, SM3 hash speed is 2200 Mbps, SM4 encryption/decryption speed is 8/10 Gbps, multiple indicators achieve higher level; When using 16-core CPU @1300 MHz, SM2/3 performance can be improved by more than 100%, and the speed of SM2 signature could achieve 10^5 times/s with 48-core CPU.

Key words: Cipher card; PCIe bus; GM algorithm; Non-Blocking

收稿日期：2019-01-03；网络出版：2019-04-25

*通信作者： 曾学文 zengxw@dsp.ac.cn

基金项目：中国科学院战略性科技先导专项课题(XDC02010701)

Foundation Item: Strategic Priority Research Program of the Chinese Academy of Sciences(XDC02010701)

1　引言

随着通迅网络的飞速发展，信息安全问题成为关注的焦点，对信息进行加密处理是保证信息安全的重要手段，因此密码运算的安全性及性能变得越来越重要。根据文献[1]，在ARM Cortex-A53平台上实现的256 bit 椭圆曲线密码算法(Elliptic Curves Cryptography, ECC)签名运算速度只有1000次/s左右，验证200次/s左右，性能低，无法满足高速网络安全服务的需求。文献[2]在Intel Xeon E3上实现的256 bit ECC签名速度达到2.9×10^4次/s，验证速度达到1.2×10^4次/s，算法性能较高，但安全性低，导致CPU负载过重，系统响应缓慢。文献[3]在基于统一计算设备架构(Compute Unified Device Architecture, CUDA)的GeForce GTX 780 Ti平台上，采用众核多线程技术实现的256 bit ECC签名达8.71×10^6次/s，验证达9.29×10^5次/s，性能优良，但功耗达420 W以上，发热严重。在Altera Cyclone Ⅱ系的现场可编程门阵列(Field Programmable Gate Array, FPGA)平台上实现的有限域GF(2^{192})上的标量乘运算速度为9000次/s左右[4]，依然无法满足当下的使用需求。本文基于外设部件互连标准总线(Peripheral Component Interconnect express, PCIe)的密码卡具有以下特点[5]：采用内置密码协处理器的可编程多核CPU，可完成高性能密码运算，降低主机CPU负荷，提高安全性；多核运算架构易于扩展提高密码运算性能。PCIe3.0总线[6]单通道速度达1 GB/s, 8通道速度高达8 GB/s。为避免输入/输出(Input/Output, I/O)限制密码运算速度，本文基于PCIe总线提出密码卡的设计与实现方法，本系统采用直接内存存取(Direct Memory Access, DMA)引擎，支持非阻塞(Non-Blocking)方式的请求；同时利用MIPS64多核芯片的大数运算支持指令和cache/register对SM2椭圆密码算法中的大数模乘和SM3杂凑算法中的迭代压缩运算进行优化；采用分层设计的思想，高效实现各种算法，便于上层扩展国产密码标准接口之外的相应功能，具有安全性高、性能优良可伸缩、扩展性强的特点。

2　密码卡硬件设计

PCIe密码卡采用10核 OCTEON 系统级芯片(System On a Chip, SOC)CN6645[7]，每个核均带有密码协处理器(CO-Processor, COP)并支持大数乘加运算加速指令。该芯片内置物理噪声源，可生成真随机数，还包含安全密钥存储区，可提高密码运算的安全性；配备了6个DMA引擎，大大提高了数据传输速度。密码卡硬件设计如图1所示。

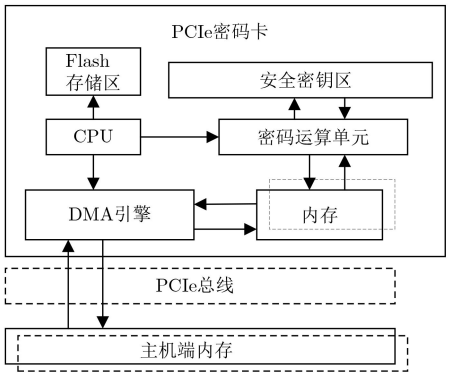

图 1 硬件设计框图

在图1中，Flash为32 MB，为固件程序、验证信息及密钥密文的存储提供了条件；安全密钥区为8 kB，可用于存储私钥，确保安全性。CPU首先从Flash读取固件程序及密钥信息，并把私钥安装到安全密钥区，同时完成DMA的初始化。在收到运算请求后，直接交由DMA调度，省去了CPU的包调度及中断响应开销，可提高数据传输效率。密码运算单元从内存中读取运算请求，并从安全密钥区获得密钥信息完成运算，之后把运算结果暂存到内存中，由DMA根据PCIe总线的工作状态把结果发送到总线上。DMA支持内/外部(Internal/External)模式，可直接完成密码卡及主机内存的数据移动，节省CPU时间，提高了工作效率。

3　密码卡软件设计

密码卡软件包括固化在卡中的固件程序，符合文献[8]的主机调用高速密码运算功能接口软件，支持主机非阻塞(Non-Blocking)批量请求调用密码运算的标准扩展接口及支持非阻塞批量请求的驱动程序。下面对密码卡的固件程序，主机调用接口使用及支持批量请求的驱动程序的主要设计流程展开说明。

3.1　固件程序设计及SM2/3算法优化

固件为密码运算模块，用于完成各种密码运算，其中的国产密码算法SM2/3/4参考国产密码算法标准和OpenSSL框架自主研发并进行优化。

对于SM4对称密码算法[9]、SHA及AES等运算直接由COP完成，CPU及DMA只负责密钥的设置和数据输入/输出调度。

为获得高I/O性能，设置DMA引擎使其工作在Scatter-Gather传输模式，该模式支持同时分配多个I/O缓冲区，并使用一个链表描述物理上不连续的缓冲区，然后把链表首地址返回到DMA控制器。DMA控制器在传输完一块缓冲区的数据后，不用发起中断，可根据链表直接传输下一缓冲区的数据，直到传输完毕后再发起一次中断。这避免了

单个Block传输模式下的频繁中断，提高了数据在PCIe总线上的传输速率。

RSA, SM2[10]算法中耗时最多的操作是大数模乘、模除及SM2中的标量乘运算。本文采用文献[11–15]提出的优化算法，结合CPU的双指令并行特性及流水线架构，通过合理设计指令顺序，使得每个周期尽量执行两条指令。对SM2从点乘运算、点加和倍点运算及素数域大数运算3个层次进行优化，以减少运行耗时。首先，对于SM2中的点乘运算，实现时使用预计算的方法，在最开始时预先计算出与椭圆曲线基点G有关的点并存储在cache中，之后使用时直接通过查表的方式读取相应结果，省去每次计算的时间开销，提高标量乘的运算速度。其次，对于SM2中的点加和倍点运算，把曲线上的点从仿射坐标系映射到Jacobe投影坐标系，之后用映射后的坐标进行运算，可大幅降低运算复杂度。对于有限域的大数运算，利用多核芯片硬件支持的192×64 bit及64×64 bit两种大数乘加指令，用汇编语言实现Montgomery模乘运算，把SM2算法中的256 bit大数模乘运算切分为5个192×64 bit及1个64×64 bit的模乘运算，这样共需要进行6次模乘运算，再把各中间结果错位相加得到最后结果。该算法同时使用多个寄存器共享操作数，每个操作数只需被加载1次，也减少了内存存取操作的数量，显著提高了大数运算的性能。

SM3[16]算法对任意长度消息杂凑输出256 bit消息摘要，文献[17]在Xilinx V5平台上实现的SM3算法吞吐率达到1.5 Gbps，性能较低。而文献[18]在Intel Core i3平台上实现的SM3性能更是仅为1 Gbps，无法满足使用需求。在65 nm工艺的专用集成电路(Application Specific Integrated Circuit, ASIC)平台上实现的SM3性能较高，可达到3.4 Gbps[19]。本密码卡的SM3算法由多核CPU编程实现，结合算法流程及多核CPU特性对其进行优化。SM3标准定义的算法过程如下：

(1) 定义如式(1)和式(2)的两个布尔函数：

$$\mathrm{FF}_j(x,y,z)=\begin{cases}x\oplus y\oplus z, & 0\le j\le 15\\ (x\wedge y)\vee(x\wedge z)\vee(y\wedge z), & 16\le j\le 63\end{cases} \tag{1}$$

$$\mathrm{GG}_j(x,y,z)=\begin{cases}x\oplus y\oplus z, & 0\le j\le 15\\ (x\wedge y)\vee(\widetilde{x}\wedge z), & 16\le j\le 63\end{cases} \tag{2}$$

(2) 定义向量**IV** 及常量T_j并初始化：

$$\mathbf{IV}=7380166f\ 4914b2b9\ 172442d7\ da8a0600\ a96f30bc\ 163138aa\ e38dee4d\ b0fb0e4e$$

$$T_j=\begin{cases}79cc4519, & 0\le j\le 15\\ 7a879d8a, & 16\le j\le 63\end{cases} \tag{3}$$

(3) 定义转换函数P_0, P_1

$$P_0(x)=x\oplus(x\ll 9)\oplus(x\ll 17) \tag{4}$$

$$P_1(x)=x\oplus(x\ll 15)\oplus(x\ll 23) \tag{5}$$

(4) 定义所需的32 bit字寄存器及中间变量

$W_0'\sim W_{63}'$, $W_0\sim W_{67}$, A, B, C, D, E, F, G, H, SS1, SS2, TT1, TT2

(5) 定义压缩函数$V^{(j)}$

$$\left.\begin{aligned}&\mathrm{SS1}\leftarrow((A\ll 12)+E+(T_j\ll j))\ll 7\\ &\mathrm{SS2}\leftarrow \mathrm{SS1}\oplus(A\ll 12)\\ &\mathrm{TT1}\leftarrow \mathrm{FF}_j(A,B,C)+D+\mathrm{SS2}+W_j'\\ &\mathrm{TT2}\leftarrow \mathrm{GG}_j(E,F,G)+H+\mathrm{SS1}+W_j\\ &D\leftarrow C\\ &C\leftarrow B\ll 9\\ &B\leftarrow A\\ &A\leftarrow \mathrm{TT1}\\ &H\leftarrow G\\ &G\leftarrow F\ll 19\\ &F\leftarrow E\\ &E\leftarrow P_0(\mathrm{TT2})\end{aligned}\right\} \tag{6}$$

将填充后的消息分组扩展成132个字$W_0\sim W_{67}$, $W_0'\sim W_{63}'$：

(a) 将消息分组划分为16个字$W_0\sim W_{15}$；

(b) 对于$16\le j\le 67$

$$W_j\leftarrow P_1(W_{j-16}\oplus W_{j-9}\oplus(W_{j-3}\ll 15))\oplus(W_{j-13}\ll 7)\oplus W_{j-6} \tag{7}$$

(c) 对于$0\le j\le 63$

$$W_j'=W_j\oplus W_{j+4} \tag{8}$$

(d) 压缩计算过程如式(9)所示

$$ABCDEFGH\leftarrow V^{(i)} \tag{9}$$

对于$0\le j\le 63$

$$V^{(i+1)}\leftarrow ABCDEFGH\oplus V^{(i)} \tag{10}$$

$$ABCDEFGH\leftarrow V^{(n)} \tag{11}$$

当所有的分组都处理完后，输出256 bit的杂凑值$ABCDEFGH$。

针对上述算法流程，本文提出算法流程和CPU编程实现两个层次的优化方案。

3.1.1 算法流程层次

(1) 根据上述算法中循环变量的值，将FF和GG两个函数拆分成FF_0, FF_1和GG_0, GG_1 4个函数

$$\mathrm{FF}_0(x,y,z)=x\oplus y\oplus z \tag{12}$$

$$\mathrm{FF}_1(x,y,z)=(x\wedge y)\vee(x\wedge z)\vee(y\wedge z) \tag{13}$$

$$\mathrm{GG}_0(x,y,z)=x\oplus y\oplus z \tag{14}$$

$$\mathrm{GG}_1(x,y,z)=(x\wedge y)\vee(\widetilde{x}\wedge z) \tag{15}$$

(2) 将V中的64次循环拆分为0～15, 16～63两组，分别对应FF_0, GG_0和FF_1, GG_1，以减少不必要的分支判断；

(3) 将T的两个取值$0x79cc4519$, $0x7a879d8a$直接代入V的64次循环中，减少赋值操作，可节省指令的执行时间，提高运算速度。

3.1.2 CPU编程实现层次

(1) 将FF_0, FF_1, GG_0, GG_1, rotate_left, P_0, P_1定义为静态内联函数，降低跳转和参数进出栈的开销；

(2) 将MIPS64的标准汇编指令(异或、循环右移)用宏汇编封装，并替换上述函数中的相应运算，利用硬件加速平台提升处理速度；

(3) 将频繁读写的数组与Cache-line Block (128 B)对齐并预放入Cache中，避免从内存中读写数据，同时提高了数据读取的效率。SM3优化前后单核性能对比如表1所示。

表 1　优化前后对比

输入长度(Byte)	运算速度(Mbps)		性能提升(%)
	优化前	优化后	
64	96	115	19.8
256	156	186	19.2
1 k	185	220	18.9
4 k	194	231	19.1
16 k	196	233	18.9

由表1可知，优化后SM3的处理速度提升19%左右，大包处理性能单核达到233 Mbps，可见上述优化方案具有良好的效果。

3.2 主机接口软件及调用示例设计

主机接口软件按文献[8]的应用接口规范进行开发，用于发送和接收命令与数据，采用分层实现的策略，各层设计如下：

应用层示例：用户编写的程序，首先调用接口SDF_OpenDevice, SDF_OpenSession打开设备及会话，返回句柄DevHandle后，该句柄传入到其它符合文献[8]的API中发送各种运算请求。

API层：具体实现文献[8]规定的标准API及本文实现的标准外扩展接口，把输入数据及运算指令按照标准格式进行封装并向下层传送，同时把返回的运算结果写入到相应的输出区域中。

PCI驱动层：该层主要实现数据的填充、相关请求格式的转换及主机端非阻塞类型请求的处理。该层以驱动文件的形式存在，在使用时需先将其安装到linux主机系统内核中。

3.3 驱动软件设计

驱动软件分为密码卡端和主机端两部分，为支持非阻塞方式的请求，驱动程序对请求采取如图2所示的处理机制。

由图2可知，主机请求由运算指令和运算数两部分组成。针对非阻塞方式的请求，主机端驱动首先在内存中创建输入/输出队列(Input/Output Queue)及请求标识队列(Request ID Queue)，并从请求中提取运算指令及输入数据地址等信息加入到输入队列，同时把输出缓存区地址存储到输出队列，之后发送输入队列到密码卡。密码卡收到请求后，为每个请求分配对应的请求标识码(req_id)并将其反馈到主机端，主机驱动根据req_id查询请求是否处理完成，完成后主机驱动便可去输出队列中的地址区域读取相应的运算结果。

卡端驱动初始化DMA引擎并在内存中分配输入/输出缓存区(Input/Output Buffer)及工作队列入口(Work Queue Entry, WQE)用于暂存操作数据、运算结果以及请求指令，WQE可存储1024个请求。在收到运算请求后，卡端驱动把请求指令加入到WQE中，并通过DMA的Inbound模式把各请求的运算数据读入密码卡并加入到输入缓存区中。同时为WQE中存储的每个请求分配req_id并立即

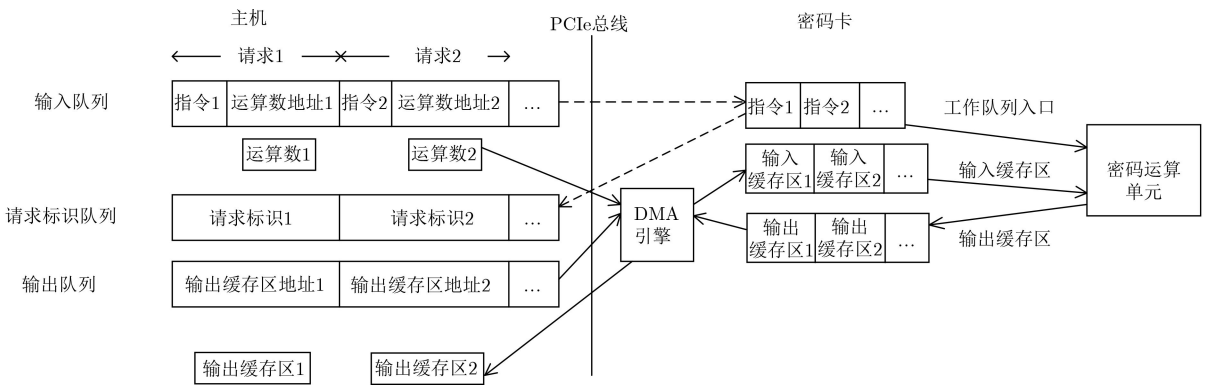

图 2 非阻塞请求处理机制

向主机驱动返回该req_id，但不会立即进行运算。卡端驱动根据密码运算单元(Crypto Unit)的工作状态把WQE中的请求指令及输入缓存区中的运算数据取出并发送到相应的Crypto Unit完成运算，运算结果暂存到输出缓存区中。DMA通过Inbound模式获取主机端输出队列并通过Outbound模式把输出缓存区中的数据发送到主机端。

这种方式可以对请求进行批量处理，省去每个请求独立发送的等待响应时间，大大提高了单进程的性能。阻塞方式和非阻塞方式的单进程性能对比如表2所示。

表 2 阻塞、非阻塞单进程对比

请求运算类型	运算速度(次/s)		性能提升(%)
	阻塞	非阻塞	
SM2 签名	1710	17523	900
SM2 验签	418	4240	900
RSA(2048) 签名	219	2200	900
RSA(2048) 验签	2018	20232	900

由表2可知以非阻塞方式发送请求的性能为阻塞方式请求的10倍，相当于密码卡中10个核同时工作。另外，由于发起的每个请求均携带请求组(Req_Group)的信息，该信息为数值0～9中的任意值，因此可通过设置每个MIPS64核可处理的请求组号，如设置0～9号核每核可处理的请求组为0～9，可实现各个核间任务的均衡调度，使每个核都工作在最佳状态，从而达到了最佳性能。

主机端以阻塞方式发送请求时，为达到类似非阻塞发送请求方式的性能，需主机端CPU支持多核多进程编程，通过并行发送密码运算请求达到高性能。

4 密码卡安全机制

作为密码运算设备，密码卡自身的安全性及使用过程中产生的安全问题同样受到人们的关注。基于此，在密码卡的设计及使用中设计了相应的安全机制。

4.1 密码卡自身安全机制

密钥的存储及固件的完整性对安全设备而言至关重要，针对该问题，设置如下安全机制：设置密钥安全存储区。该密码卡设置的密钥安全存储区域无法通过技术手段从外部探测密钥信息，同时不对用户提供私钥导出接口，实现私钥不出密码卡，从而达到安全存储密钥的目的。验证密码卡内固件完整性。为保证固件完整性，预先对卡内固件做SM3杂凑处理，对杂凑结果做SM2签名处理，在运行前进行验证操作，只有在验证通过时才会提供服务，从而保证了固件完整性及设备安全。

4.2 用户安全机制

密码卡可同时向多用户提供服务，因此涉及到各用户的密钥信息及数据安全问题。为此，该密码卡采取了如下各安全机制[20]：

设置安全口令。不同用户分配不同密钥存储区，同时每个用户掌握自己的私钥使用口令及管理口令，保证密钥的安全性。

设置口令试探次数。对用户的管理口令设置尝试次数的限制，在多次输入错误后该用户的密钥区会被清零，防止用户密钥被非法使用。

支持密钥加密导入。密码卡支持导入加密后的密钥，保证密钥的传输安全。

设置不同角色的用户。角色不同的用户拥有不同的权限，管理员用户权限最高，可管理普通用户及设备的密钥及口令，在密码卡出现故障时可由管理员进行恢复，可降低安全风险。

5 密码卡性能测试及对比

为提高密码运算性能，该密码卡软件部分使用C语言进行编程(部分使用汇编语言)。性能测试环境：CPU为Intel(R) Xeon(R) E5-2640 v2，主频2.00 GHz；内存16 G；系统为64 bit Centos6.5，测试时主机端分别调用文献[8]规定的各类密码运算接口API，其中SM2, RSA性能测试采用16或64 Byte小包，SM3, SM4, AES及SHA的测试采用16 kB的大包，各API调用循环2000次，再统计各密码运算的性能。当前主流的密码卡有卫士通的SJK1572系列[21]、渔翁信息的SJK1120系列[22]及西电捷通的SJK1337系列[23]，性能测试结果对比如表3所示。

由表3的测试结果可知：基于多核[24]SOC芯片的密码卡相较于其它同类产品，对国产密码SM2/3/4算法具有优良的支持性能，同时RSA, AES及SHA性能表现同样出色，尤其是SM4及SHA1/256，在业内处于国内领先地位。在采用主频1300 MHz的16核MIPS64 CPU时，性能可大幅提高，SM2的签名速度可达38000次/s，验证速度达9300次/s, SM3杂凑速度达5400 Mbps, SM4加/解密速度达12/13 Gps，采用48核MIPS64 CPU时SM2签名可达10^5次/s。相较于文献[1–4]，该加密卡的性能远高于文献[1,4]的签名1000次/s，验证200次/s及9000次/s的标量乘的性能指标，同时具有比文献[2,3]更高的安全性及性能功耗比，可实现高强度，高性能的密码运算。

6 结束语

本文提出了一种基于PCIe的高速密码卡的设

表 3　密码卡性能测试结果对比

密码卡种类	SM2 (次/s)		SM3 (Mbps)	SM4 (Gbps)	RSA2048 (次/s)		AES128 (Gbps)	SHA1 (Gbps)	SHA256 (Gbps)
	签名	验证			签名	验证			
SJK1572	14000	4000	1300	1.3	–	–	–	–	–
SJK1120	1800	1300	1	1.2	30	350	1.2	–	–
SJK1337	31000	19000	1700	2.2	–	–	–	–	0.8
本密码卡	18000	4100	2200	8.1	2200	20232	9.0	13.0	13.0

计与系统实现方法，并从算法流程及编程实现两个层面对SM3杂凑算法进行了优化，优化后性能大幅提升。同时提出了非阻塞模式的请求发送方式，使得应用在单进程下即可达到最大性能，大幅提高了密码运算速度。对于大数模乘运算，本文也提出了一种利用大数乘加指令进行优化的方式，降低了乘法次数及用时。此外，本密码卡支持按需配置启动核数，可实现性能的动态配置，并且对密码卡的核间任务调度提出了一种负载均衡方法，使各核可同时达到满载或重载状态。相比于国内其它国产密码算法密码卡，本文的密码卡多项运算指标均具有较高的性能，可完成高强度的密码运算，满足高速网络应用的需求。

参 考 文 献

[1] ABBASINEZHAD-MOOD D and NIKOOGHADAM M. An anonymous ECC-based self-certified key distribution scheme for the smart grid[J]. *IEEE Transactions on Industrial Electronics*, 2018, 65(10): 7996–8004. doi: 10.1109/TIE.2018.2807383.

[2] ADALIER M. Efficient and secure elliptic curve cryptography implementation of curve P-256[EB/OL]. http://csrc.nist.gov/groups/ST/ecc-workshop-2015/papers/session6-adalier-mehmet.pdf.

[3] PAN Wuqiong, ZHENG Fangyu, ZHAO Yuan, *et al*. An efficient elliptic curve cryptography signature server with GPU acceleration[J]. *IEEE Transactions on Information Forensics and Security*, 2017, 12(1): 111–122. doi: 10.1109/TIFS.2016.2603974.

[4] 程明智, 周由胜, 辛阳, 等. GF(2^{192})域上ECC加密的FPGA实现[J]. 华中科技大学学报 (自然科学版), 2009, 37(10): 9–12. doi: 10.13245/j.hust.2009.10.023.
CHENG Mingzhi, ZHOU Yousheng, XIN Yang, *et al*. FPGA realization of ECC encryption algorithm in GF(2^{192})[J]. *Journal of Huazhong University of Science and Technology* (*Natural Science Edition*), 2009, 37(10): 9–12. doi: 10.13245/j.hust.2009.10.023.

[5] ROTA L, CASELLE M, CHILINGARYAN S, *et al*. A PCIe DMA architecture for multi-gigabyte per second data transmission[J]. *IEEE Transactions on Nuclear Science*, 2015, 62(3): 972–976. doi: 10.1109/TNS.2015.2426877.

[6] PCI express base specification revision 3.0[EB/OL]. https://doc.mbalib.com/view/e99fb1d0aab4982329ffd43f1a0dbf3b.html, 2010.

[7] CAVIUM. OCTEON Ⅱ CN66XX multi-core MIPS64 Proce-ssors[J/OL]. http://www.cavium.com/OCTEON Ⅱ CN66XX.html. 2011.

[8] 国家密码管理局. GM/T 0018–2012 密码设备应用接口规范[S]. 北京: 中国标准出版社, 2012.
State Cryptography Administration Office of Security Commercial Code Administration. GM/T 0018–2012 Interface specifications of cryptography device application[S]. Beijing: China Standard Press, 2012.

[9] 国家密码管理局. GM/T 0002–2012 SM4分组密码算法[S]. 北京: 中国标准出版社, 2012.
State Cryptography Administration Office of Security Commercial Code Administration. GM/T 0002–2012 SM4 block cipher algorithm[S]. Beijing: China Standard Press, 2012.

[10] 国家密码管理局. GM/T 0003–2012 SM2椭圆曲线公钥密码算法[S]. 北京: 中国标准出版社, 2012.
State Cryptography Administration Office of Security Commercial Code Administration. GM/T 0003–2012 Public key cryptographic algorithm SM2 based on elliptic curves[S]. Beijing: China Standard Press, 2012.

[11] LI Yang, WANG Jinlin, ZENG Xuewen, *et al*. Fast Montgomery modular multiplication and squaring on embedded processors[J]. *IEICE Transactions on Communications*, 2017, E110.B(5): 680–690. doi: 10.1587/transcom.2016EBP3189.

[12] MONTGOMERY P L. Modular multiplication without trial division[J]. *Mathematics of Computation*, 1985, 44(170): 519–521. doi: 10.1090/S0025-5718-1985-0777282-X.

[13] MÖLLER B. Improved techniques for fast exponentiation[C]. The 5th International Conference on Information Security and Cryptology-ICISC 2002, Seoul, Korea, 2002: 298–312. doi: 10.1007/3-540-36552-4_21.

[14] ZHANG Dan and BAI Guoqiang. High-performance implementation of SM2 based on FPGA[C]. The 8th IEEE International Conference on Communication Software and Networks, Beijing, China, 2016: 718–722. doi:

10.1109/ICCSN.2016.7586618.

[15] ZHOU Xin and TANG Xiaofei. Research and implementation of RSA algorithm for encryption and decryption[C]. The 6th International Forum on Strategic Technology, Harbin, China, 2011, (2): 1118–1121. doi: 10.1109/IFOST.2011.6021216.

[16] 国家密码管理局. GM/T 0004–2012 SM3密码杂凑算法[S]. 北京: 中国标准出版社, 2012.
State Cryptography Administration Office of Security Commercial Code Administration. GM/T 0004–2012 SM3 cryptographic hash algorithm[S]. Beijing: China Standard Press, 2012.

[17] 朱宁龙, 戴紫彬, 张立朝, 等. SM3及SHA-2系列算法硬件可重构设计与实现[J]. 微电子学, 2015, 45(6): 777–780. doi: 10.13911/j.cnki.1004-3365.2015.06.021.
ZHU Ninglong, DAI Zibin, ZHANG Lichao, *et al*. Design and implementation of hardware reconfiguration for SM3 and SHA-2 hash function[J]. *Microelectronics*, 2015, 45(6): 777–780. doi: 10.13911/j.cnki.1004-3365.2015.06.021.

[18] 杨先伟, 康红娟. SM3杂凑算法的软件快速实现研究[J]. 智能系统学报, 2015, 10(6): 954–959. doi: 10.11992/tis.201507036.
YANG Xianwei and KANG Hongjuan. Fast software implementation of SM3 hash algorithm[J]. *CAAI Transactions on Intelligent Systems*, 2015, 10(6): 954–959. doi: 10.11992/tis.201507036.

[19] 于永鹏, 严迎建, 李伟. SM3算法高速ASIC设计及实现[J]. 微电子学与计算机, 2016, 33(4): 21–26. doi: 10.19304/j.cnki.issn1000-7180.2016.04.005.
YU Yongpeng, YAN Yingjian, and LI Wei. High speed ASIC design and implementation of SM3 algorithm[J]. *Microelectronics & Computer*, 2016, 33(4): 21–26. doi: 10.19304/j.cnki.issn1000-7180.2016.04.005.

[20] JUANG W S. Efficient multi-server password authenticated key agreement using smart cards[J]. *IEEE Transactions on Consumer Electronics*, 2004, 50(1): 251–255. doi: 10.1109/TCE.2004.1277870.

[21] 卫士通. 商用PCI-E密码卡[EB/OL]. http://www.westone.com.cn/index.php?m=content&cindex&ashow&catid17&id1, 2018.
WESTONE. Commercial PCI-E cipher card[EB/OL]. http://www.westone.com.cn/index.php?m=content&cindex&ashow&catid17&id1, 2018.

[22] 渔翁信息. 如何选择商密加密卡[EB/OL]. http://www.fisec.com.cn/page118?article_id=30, 2017.
FISEC. How to Choose a commercial encryption card [EB/OL]. http://www.fisec.com.cn/page118?article_id=30, 2017.

[23] 西电捷通. 高速通用密码卡之西电捷通综合性测试分析[EB/OL]. http://www.sohu.com/a/124421829_446726, 2017.
IWNCOMM. Comprehensive test analysis of IWNCOMM with high-speed universal cipher card[EB/OL]. http://www.sohu.com/a/124421829_446726, 2017.

[24] 李军, 陈君, 倪宏, 等. 基于多核协作的流媒体内容缓存算法[J]. 网络新媒体技术, 2014, 3(4): 12–18. doi: 10.3969/j.issn.2095-347X.2014.04.003.
LI Jun, CHEN Jun, NI Hong, *et al*. Multi-core platform based multimedia collaboration caching algorithm[J]. *Journal of Network New Media*, 2014, 3(4): 12–18. doi: 10.3969/j.issn.2095-347X.2014.04.003.

赵 军：男，1991年生，博士生，研究方向为信息安全.
曾学文：男，1968年生，研究员，研究方向为网络新媒体技术.
郭志川：男，1975年生，研究员，研究方向为FPGA硬件加速技术.