(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 111917548 B
(45)授权公告日 2021.06.04

(21)申请号 201910375545.3

(22)申请日 2019.05.07

(65)同一申请的已公布的文献号
申请公布号 CN 111917548 A

(43)申请公布日 2020.11.10

(73)专利权人 北京大学
地址 100871 北京市海淀区颐和园路5号

(72)发明人 郁莲 王晓天

(74)专利代理机构 北京万象新悦知识产权代理有限公司 11360
代理人 黄凤茹

(51)Int.Cl.
H04L 9/32(2006.01)
G06F 9/50(2006.01)

(56)对比文件
CN 103475469 A,2013.12.25
CN 103546288 A,2014.01.29
US 2013246797 A1,2013.09.19
Sokjoon Lee.Hybrid approach of parallel implementation on CPU-GPU for high-speed ECDSA verification.《The Journal of Supercomputing》.2019,全文.

审查员 孙铭君

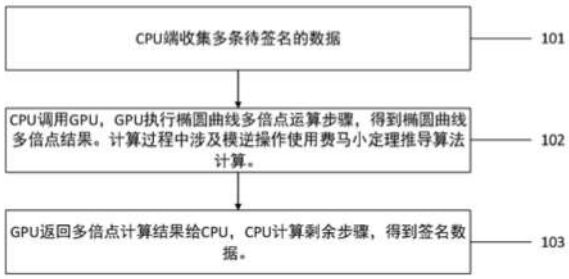
权利要求书2页 说明书5页 附图2页

(54)发明名称

基于GPU与CPU异构结构的椭圆曲线数字签名方法

(57)摘要

本发明公布了一种椭圆曲线数字签名方法，采用基于CPU和GPU异构结构的分工协作方式进行ECDSA数字签名生成算法的计算，对费马小定理进行变形，使用该变形方法优化椭圆曲线数字签名的模逆运算在GPU中的实现步骤，批量进行椭圆曲线数字签名中椭圆曲线的多倍点计算，再根据GPU计算得到的结果，由CPU将待签名消息根据椭圆曲线数字签名计算步骤顺序进行计算，充分利用GPU架构的并行性，提高ECDSA数字签名生成算法的计算效率，从而节省待签名消息进行数字签名的计算时间。



CN 111917548 B

1.一种椭圆曲线数字签名方法，采用基于CPU和GPU异构结构的分工协作方式进行ECDSA数字签名生成算法的计算，对费马小定理进行变形，使用该变形方法优化椭圆曲线数字签名的模逆在GPU中的实现步骤，批量进行椭圆曲线数字签名中椭圆曲线的多倍点计算，再根据GPU计算得到的结果，由CPU将待签名消息根据椭圆曲线数字签名计算步骤顺序进行计算，从而节省待签名消息进行数字签名的计算时间；

椭圆曲线数字签名的输入为：参数组D＝(q,F_q,a,b,P,n)，其中F_q为素域，其阶为q；a,b为椭圆曲线方程$y^2=x^3+ax+b$的参数，点P为基点，是域中的两个元素组成的椭圆曲线上的已知点，P＝(x_p,y_p)，P点有素数阶n；输出为签名(r,s)；

基于CPU和GPU异构结构的椭圆曲线数字签名方法包括如下步骤：

1)CPU每次等待一段时间，汇总这段时间内收到的签名请求，即获得i条待签名消息m，作为待计算数据；

2)CPU将i条数据传送给GPU；

3)通过GPU并行执行ECDSA算法中的计算椭圆曲线多倍点运算，对于第j条数据，j∈[1,i]，选择随机数k_j∈[0,n-1]，计算(x_j,y_j)＝k_jP；(x_j,y_j)即为多倍点结果；具体执行操作：

将椭圆曲线多倍点的计算步骤拆解为有限域运算层基础操作，包括：点乘、点加、倍点；点乘再分解为点加和倍点；点加和倍点再分解为更低层次的有限域运算，包括：模乘；模平方；模加；模逆；

在GPU平台使用费马小定理进行模逆算法的计算；具体为：

根据费马小定理：假如z是质数，t是整数，且t和z互质，那么$t^{z-1}\equiv 1 \bmod z$；

推导得到：$t^{z-2}\equiv t^{-1} \bmod z$，即$t^{-1}\equiv t^{z-2} \bmod z$；

在多倍点计算过程中，有质数p和f，f∈[1,p-1]，求f^{-1}；即输入为：质数p，整数f，其中f∈[1,p-1]；输出为：$f^{-1}\equiv f^{p-2} \bmod p$，mod表示求模；f对于质数p的模逆值$f^{-1}$，$f^{-1}$即为所求结果；

由此得到i个椭圆曲线点乘结果；

4)将GPU计算得到的多个椭圆曲线点乘结果(x_1,y_1)…(x_i,y_i)返回给CPU；

5)CPU获得GPU计算得到的结果，继续进行步骤51)-54)的计算，得到每条数据对应的签名消息，每个签名消息对应一组签名(r_j,s_j)，j∈[1,i]；

对每一条数据，执行如下计算步骤：

步骤51)计算$r_j=x_j \bmod n$，若r_j＝0，则将对应的原始数据返回到步骤1)；

步骤52)计算e_j＝H(m_j)；H为密码哈希函数；

步骤53)计算$s_j = k_j^{-1}(e_j + dr_j) \bmod n$，d为私钥；若$s_j$＝0，则将对应的原始数据返回步骤1)；

步骤54)获得签名(r_j,s_j)；

通过上述步骤，实现基于CPU和GPU异构结构的分工协作方式进行ECDSA数字签名生成算法的计算，得到ECDSA数字签名。

2.如权利要求1所述的椭圆曲线数字签名方法，其特征是，步骤1)中，CPU每次等待时间为1毫秒～1秒；获得待计算数据条数为1～2048条。

3.如权利要求1所述的椭圆曲线数字签名方法，其特征是，CPU和GPU异构结构中，CPU计算单元采用Intel公司的X86指令集通用计算机。

4.如权利要求1所述的椭圆曲线数字签名方法，其特征是，CPU和GPU异构结构中，GPU计算单元包括Nvidia公司支持CUDA运算的任意图形显示卡。

基于GPU与CPU异构结构的椭圆曲线数字签名方法

技术领域

[0001] 本发明涉及信息安全中的公钥密码技术，特别涉及一种基于中央处理器(CPU)和图形处理器(GPU，Graphics Processing Unit)的椭圆曲线数字签名算法(ECDSA)的实现方法。

背景技术

[0002] 采用多核并行计算是提升处理器性能的重要方式，因此出现了包括海量并行结构运算单元的GPU(图形处理器，Graphics Processing Unit)，GPU已经发展成为了并行度高、多线程、计算快捷及内存带宽大的高性能通用处理器。GPU体系结构在组成上分为三层：第一层由若干个线程处理器簇(TPC，Thread Preocessing Cluster)组成，第二层由多个流多处理器(SM，Streaming Multiprocessor)组成，第三层为构成SM的流处理器(SP，Stream Processor)，也可以称为线程处理器。SM作为GPU的一个任务执行和调度单元，主要负责执行GPU分发的线程指令，而SP是GPU中最基本的指令执行单元，其执行的操作由所属的SM控制。

[0003] 2006年NVIDIA公司推出了计算机一体设备结构(CUDA，Compute Unified Device Architecture)可编程平台，可以实现GPU线程的调度。在CUDA可编程平台架构下，GPU执行的最小单位是线程(thread)，数个线程(thread)可以组成一个线程块(block)。一个block中的thread可以存取同一共享内存且同步。执行相同程序的thread，组成栅格(grid)，不同的grid可以执行不同的程序。

[0004] 相对于中央处理器(CPU)，GPU具有强大的数据处理能力，在浮点运算及并行计算等方面，提高几十倍甚至数百倍于CPU的性能。GPU具有几千个核，有很高的并行性，但是每个SM相比CPU处理能力比较弱。GPU计算能力强，但是核数比较少。因此，采用何种CPU和GPU协作方式进行构架，完成各种计算，以提高系统整体计算能力，是一个亟待解决的问题。

[0005] 椭圆曲线数字签名算法(ECDSA)明确规定了ECDSA算法的数字签名算法。其中，ECDSA算法的数字签名算法包括生成算法和验证算法，应用在消息传输过程中可靠性的消息传输及使用消息的合法者验证。在ECDSA的数字签名算法中，包括一一对应的公钥和私钥，其中，私钥用于待签名消息生成数字签名，公钥用于对数字签名进行验证。签名者采用私钥进行待签名消息M的数字签名生成算法计算，得到待签名消息M的数字签名；验证者采用公钥对接收的待签名消息M进行数字签名验证计算，验证待签名消息M的数字签名是否匹配，如果匹配，确认接收的待签名消息M是正确的。

[0006] 椭圆曲线数字签名算法(ECDSA)是数字签名算法的椭圆曲线版本，其安全性是以椭圆曲线密码体制的安全性为保证的，而椭圆曲线密码体制的安全性是基于椭圆曲线离散对数问题之上的。最广泛标准化的基于椭圆曲线的签名方案包括ANSI X9.62、FIPS 186-2、IEEE1363-2000等。

[0007] 目前ECDSA的计算方式是使用CPU完成全部计算。在实现ECDSA数字签名生成算法时，是由CPU按照上述步骤顺序计算，最终得到待签名消息m的数字签名。由于CPU计算核心

数量少，而ECDSA的计算过程中计算量比较大，会占用CPU的大部分资源，导致ECDSA计算速度不高，比较费时，得到数字签名效率比较低。

发明内容

[0008] 为了克服上述现有技术的不足，本发明提供一种基于GPU与CPU异构结构的椭圆曲线数字签名方法，提高数字签名生成算法(ECDSA)的计算效率。

[0009] 本发明的原理是：

[0010] ECDSA是一种具有标准协议的，国际通用的签名算法。现有方法由CPU按照ECDSA数字签名生成算法计算步骤顺序执行，其中的多倍点计算复杂性比较高，比较耗时，因此，现有的使用CPU计算的方法在实现ECDSA数字签名生成算法时计算效率不高，无法满足越来越多计算需求。但由于针对ECDSA模逆算法并不适合GPU计算，现有技术难以实现针对ECDSA的GPU加速算法。本发明依托新型的GPU计算平台对该算法进行计算加速，本专利对费马小定理进行变形，使用该变形方法优化椭圆曲线数字签名的模逆在GPU中的实现步骤，获得GPU端的性能提升。具体地，本发明通过将一些重要的计算过程移植到GPU平台进行计算，依托于GPU平台，本发明首次提出了采用CPU和GPU协作方式进行ECDSA数字签名生成算法计算，将ECDSA数字签名生成算法中的计算椭圆曲线多倍点步骤批量在GPU中进行，充分利用GPU架构的并行性，然后由CPU根据ECDSA数字签名生成算法计算步骤对待签名消息顺序计算时，调用GPU计算得到的结果，GPU可以进行并行计算，即同时计算多组数据(CPU只能串行计算，此差异可以使GPU针对多数据时，获得比CPU快很多的速度)，从而节省了对待签名消息进行数字签名的计算时间，提升计算速度，提高ECDSA数字签名生成算法的计算效率。

[0011] 本发明提供的技术方案是：

[0012] 一种基于GPU与CPU异构结构的椭圆曲线数字签名方法，采用CPU和GPU异构结构(如图2所示，CPU和GPU各负责一部分数据计算)和协作方式，进行ECDSA数字签名生成算法计算，充分利用GPU架构的并行性，将计算步骤分为两个部分：在GPU中批量进行ECDSA中椭圆曲线多倍点的计算，再由CPU将待签名消息根据计算步骤顺序进行计算时，使用GPU计算得到的结果，从而节省了待签名消息进行数字签名的计算时间；使用GPU加速ECDSA计算时，由于GPU架构对程序分支敏感，而运算过程中，会包含大量的分支片段，导致运行效率低下。本发明提出了通过对费马小定理进行变形，使用该变形方法对关键的模逆步骤进行优化，使得利用GPU加速ECDSA成为可能。

[0013] 具体地，ECDSA签名算法计算步骤如下，设待签名的消息为m，为了取得消息m的数字签名(r，s)，作为签名者的用户应实现以下运算步骤：

[0014] 输入：椭圆曲线算法定义的参数组D＝(q，F_q，a，b，P，n)，其中F_q为素域，其阶为q；a、b为椭圆曲线方程$y^2=x^3+ax+b$的参数且a，b∈F_q，点P为素域中的两个元素组成的椭圆曲线上的点，P＝(x_p，y_p)，P∈F_q，是椭圆曲线上的一个固定点，被称为基点，具体值是已知的，P点有素数阶n。以上参数是椭圆曲线密码学中的运算参数，在椭圆曲线密码学中具有确定的含义；在ECDSA算法中，素域F_q的阶q为算法定义的固定值，ECDSA计算宽度为256比特(bits)时，q＝0xFFFFFFFFFFFFFFFF 00000000FFFFFFFF 0000000000000000FFFFFFFF00000001。素数阶n为算法定义的固定值，ECDSA计算宽度为256比特(bits)时，n＝0xF3B9CAC2FC632551 BCE6FAADA7179E84FFFFFFFFFFFFFFFF FFFFFFFF00000000。

[0015] H为一个密码哈希函数(SHA,国际通用哈希算法,可以输出0～512比特(bits)哈希值),d(0＜d＜n)为私钥,m为待签名消息;以上数据均以计算机可以识别的数据格式进行存储,以待计算。下述过程中的mod表示数学计算中的求模运算。

[0016] 输出:签名结果(r,s),r、s表示经ECDSA签名算法计算输出的两个数值;

[0017] 本发明方法包括如下步骤:

[0018] 1)CPU每次等待一定时间(如1毫秒～1秒),汇总这段时间内收到的签名请求,获得i(1～2048)条待计算数据;

[0019] 2)CPU将i条待计算数据传送给GPU,通过GPU对多倍点运算进行并行计算;

[0020] 3)通过GPU并行执行ECDSA算法中的计算椭圆曲线多倍点运算,得到i个椭圆曲线点乘结果;具体执行操作:对于第j(j∈[1,i])条数据,选择随机数k_j∈[0,n-1];计算(x_j,y_j)＝k_jP;(x_j,y_j)即为多倍点结果。

[0021] 4)将GPU计算得到的多个椭圆曲线点乘结果(x_1,y_1)…(x_i,y_i)返回给CPU。

[0022] 5)CPU获得GPU计算得到的结果,继续进行下述计算,最终得到每条数据对应的签名消息(每个签名消息对应一组(r_j,s_j),j∈[1,i]);对每一条数据,执行如下计算步骤:

[0023] 步骤51)计算r_j＝x_j mod n,若r_j＝0,则将对应的原始数据返回到步骤1);

[0024] 步骤52)计算e_j＝H(m_j),此处H()为哈希函数,采用的算法是国际通用哈希算法SHA256;

[0025] 步骤53)计算s_j＝k_j^{-1}(e_j+dr_j)mod n,若s_j＝0,则将对应的原始数据返回步骤1);

[0026] 步骤54)获得签名(y_j,s_j)。

[0027] GPU进行椭圆曲线多倍点计算时,会将上述椭圆曲线多倍点的计算步骤拆解为4个有限域运算层基础操作,如图3所示,ECDSA计算可以分解为点乘,点加,倍点三类操作,点乘可分解为点加和倍点,点加和倍点又可以分解为更低层次的有限域运算,包括:1)模乘;2)模平方;3)模加;4)模逆。其中,模逆算法的选择不当会导致GPU计算耗时巨大,进而整个系统性能受到冲击。当前常用的求模逆的算法是欧几里得算法,该算法程序分支过多,不利于在GPU上实现。针对模逆算法操作,本发明首次提出了在GPU平台使用费马小定理获得高效的模逆计算速度。

[0028] 在CUDA环境下,GPU内执行的程序是以32个线程合为1组来运行的。如果在程序中遇到分支语句,并且组内的某个或某几个线程与其余线程分支不同,则其余的线程通道被屏蔽,迫使它们等待分支线程完成其工作,然后才能恢复执行,增加了程序完成的时间。如果所有线程采用相同的执行路径,则不会受到屏蔽影响。

[0029] 费马小定理是数论中的一个重要定理,在1636年提出,其内容为:假如z是质数,t是整数,且t和z互质,那么t^{z-1}≡1mod z;

[0030] 由费马小定理推导可得:t^{z-2}≡t^{-1}mod z,即t^{-1}≡t^{z-2}mod z

[0031] 在多倍点计算过程中,有质数p和f(f∈[1,p-1]),求f^{-1}:

[0032] 输入:质数p,整数f(f∈[1,p-1])

[0033] 输出:f^{-1}≡f^{p-2} mod p

[0034] 注:mod是数学计算中求模的含义,由上式可以得到f对于质数p的模逆值f^{-1},f^{-1}即为所求结果。

[0035] 与现有技术相比,本发明的有益效果是:

[0036] 现有技术由CPU按照ECDSA数字签名生成算法计算步骤顺序执行，其中的多倍点计算复杂性比较高，比较耗时。本发明采用CPU和GPU协作方式进行ECDSA数字签名生成算法计算，充分利用GPU架构的并行性，通过采用批量计算有效利用GPU计算资源，从而节省了对待签名消息进行数字签名的计算时间，提高ECDSA数字签名生成算法的计算效率。具体地，本发明具有包括以下几方面的技术优势：

[0037] (一)运算速度快，相对于CPU单独计算有近10倍提升。

[0038] (二)解决了GPU实现模逆算法的架构原因导致的运算效率低效问题。

[0039] (三)随机数由GPU生成，有效节约CPU计算资源。

[0040] (四)批量计算签名，可以最大限度利用GPU的计算资源，提升算效率。

附图说明

[0041] 图1为本发明实施例提供的ECDS数字签名生成算法实现方法的流程框图。

[0042] 图2是本发明实施例采用CPU和GPU异构结构实现数字签名生成系统的模块数据流示意图。

[0043] 图3是本发明实施例中GPU进行椭圆曲线多倍点计算时，将计算步骤拆解为4个有限域运算层基础操作计算过程的示意图。

具体实施方式

[0044] 下面结合附图，通过实施例进一步描述本发明，但不以任何方式限制本发明的范围。

[0045] 图1为本发明实施例提供的一种ECDSA算法中的数字签名生成算法实现方法的流程框图，其具体步骤为：

[0046] 步骤101、CPU收集待签名的数据；

[0047] 步骤102、将ECDSA数字签名生成算法中的计算椭圆曲线点步骤由GPU计算得到；当计算模逆操作时，使用费马小定理进行计算。

[0048] 在该步骤中，结果为椭圆曲线多倍点计算结果；

[0049] 步骤103、CPU根据ECDSA数字签名生成算法计算步骤对待签名消息顺序计算时，调用GPU返回的多倍点计算结果，完成对待签名消息的数字签名。

[0050] 在图1中，步骤102由GPU执行。

[0051] 在图1中，步骤101和步骤103都是由CPU线程完成的，可以是同一CPU线程，也可以是不同的CPU线程。

[0052] 在图1中，GPU执行ECDSA数字签名生成算法中的计算椭圆曲线多倍点步骤时，可以由一个GPU线程完成，也可以由多个GPU线程针对不同的数据同时计算完成。

[0053] 图2所示是上述实施例采用的模块数据流，采用CPU和GPU异构结构和协作方式，CPU和GPU各负责一部分数据计算。其中，CPU计算单元采用Intel公司的X86指令集通用计算机；GPU计算单元包括Nvidia公司支持CUDA运算的任意图形显示卡。数据流处理过程如下：

[0054] 步骤201：由CPU在一定时间内收集多条待签名信息；

[0055] 步骤202：CPU将每条签名过程中的多倍点计算信息合并发给GPU侧；

[0056] 步骤203：GPU将每条签名的多倍点计算完成，将结果一同返回给CPU；

[0057] 步骤204:CPU完成每条签名计算的剩余步骤,将结果返回给签名请求的来源方。

[0058] 步骤202中,GPU进行椭圆曲线多倍点计算时,将计算步骤拆解为4个有限域运算层基础操作,如图3所示,ECDSA运算层表示椭圆曲线数字签名算法,算法主要计算步骤是多倍点操作,即图示中的点乘运算,点乘运算是由点加和倍点运算组成。点加和倍点运算又是由四类有限域运算构成:模乘,模平方,模加,模逆。这四类有限域运算是构成椭圆曲线数字签名算法计算的基础运算步骤。

[0059] 其中,模逆算法的选择不当会导致GPU计算耗时巨大,进而整个系统性能受到冲击。当前常用的求模逆的算法是欧几里得算法,该算法程序分支过多,不利于在GPU上实现。针对模逆算法操作,本发明首次提出了在GPU平台使用费马小定理获得高效的模逆计算速度。

[0060] 具体实现伪代码如下所示:

Algorithm Inv

Input: 256 bits prime p,**oprater** $f \in [0, p-1]$

Output: $\mathbf{f}^{-1} \bmod p$

1: **initialize:** $v = p-2; y = f; g = 1$
2: **for** $l = 0, 1, ..., 255$ **do**
3: **if** $v_l = 1$ **then**
4: $g = (g * y) \mod p$
5: **end if**
6: $y = y^2$
7: **end for**
8: return g

[0061]

[0062] 其中,第1行是变量初始化;第2～7行是一个循环体,根据v每个比特的值进行循环累乘求模运算;第8行得到最终结果,g的值即是所求的$f^{-1} \bmod p$。对于GPU实现来说,计算模逆运算最重要方面是所有线程执行相同的指令路径。算法中唯一的条件语句在第3行,v_1表示v的第1比特,它的值取决于素数p,在ECDSA算法中p的值就是椭圆曲线数字签名算法输入参数组中的q,是固定值。因此素数p在所有线程中都是相同的,所以所有线程计算过程中的指令执行路径完全相同。

[0063] 需要注意的是,公布实施例的目的在于帮助进一步理解本发明,但是本领域的技术人员可以理解:在不脱离本发明及所附权利要求的精神和范围内,各种替换和修改都是可能的。因此,本发明不应局限于实施例所公开的内容,本发明要求保护的范围以权利要求书界定的范围为准。

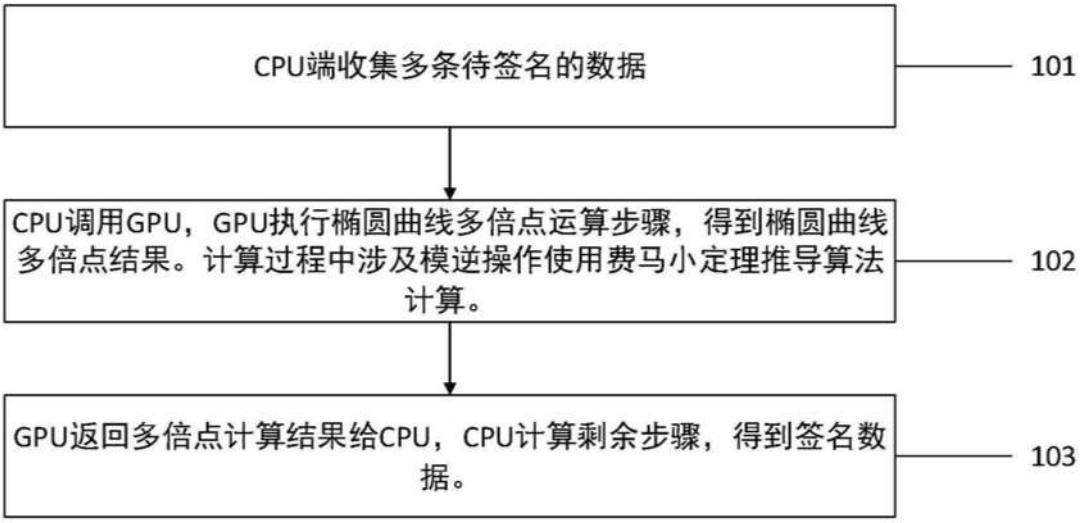

CPU端收集多条待签名的数据
101
CPU调用GPU，GPU执行椭圆曲线多倍点运算步骤，得到椭圆曲线多倍点结果。计算过程中涉及模逆操作使用费马小定理推导算法计算。
102
GPU返回多倍点计算结果给CPU，CPU计算剩余步骤，得到签名数据。
103

图1

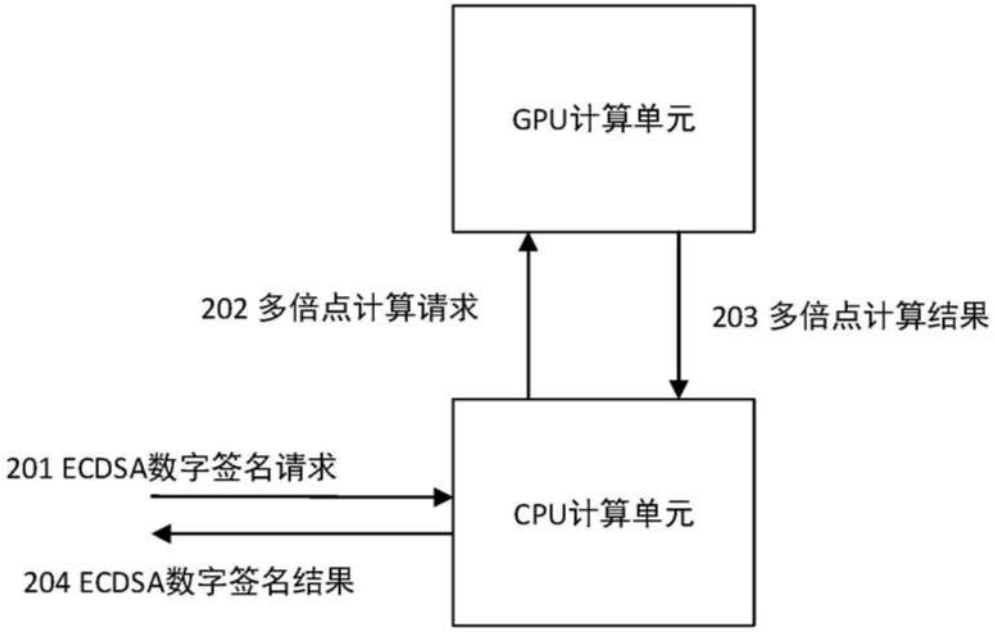

GPU计算单元
202 多倍点计算请求
203 多倍点计算结果
201 ECDSA数字签名请求
CPU计算单元
204 ECDSA数字签名结果

图2

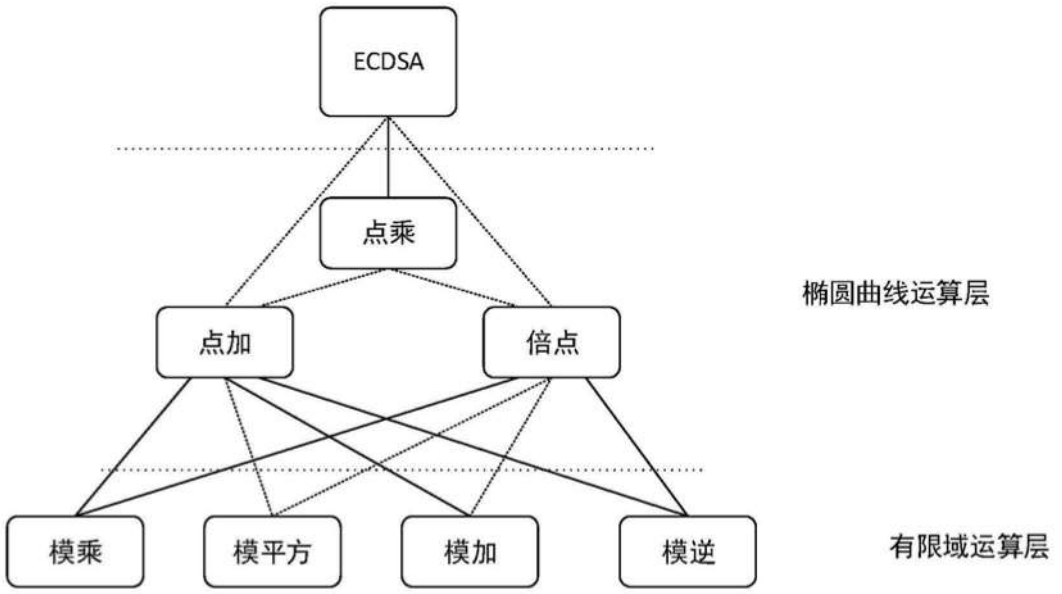


图3