(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 113221193 B
(45)授权公告日 2022.07.29

(21)申请号 202110613751.0
(22)申请日 2021.06.02
(65)同一申请的已公布的文献号
申请公布号 CN 113221193 A
(43)申请公布日 2021.08.06
(73)专利权人 上海交通大学
地址 200240 上海市闵行区东川路800号
(72)发明人 邱卫东 张崴城 王杨德 田昊 郭捷 唐鹏
(74)专利代理机构 上海交达专利事务所 31201
专利代理师 王毓理 王锡麟
(51)Int.Cl.
G06F 21/64(2013.01)

(56)对比文件
CN 112887081 A,2021.06.01
CN 107147488 A,2017.09.08
CN 110365481 A,2019.10.22
CN 109600233 A,2019.04.09
CN 112187469 A,2021.01.05
WO 2019174402 A1,2019.09.19

审查员 刘金鑫

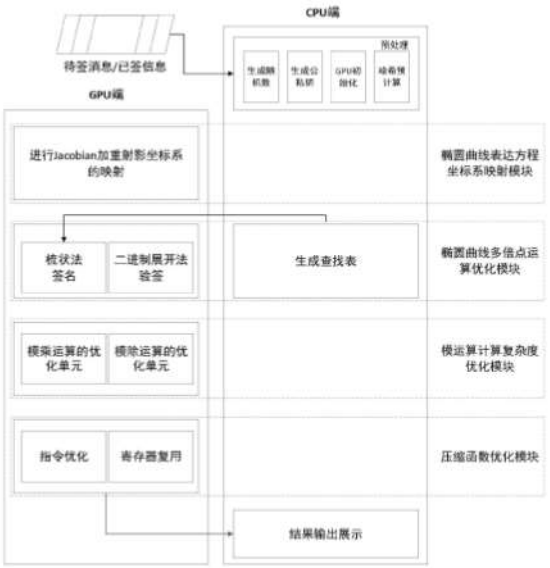
权利要求书2页 说明书5页 附图2页

(54)发明名称

基于GPU的SM2数字签名与验签快速实现方法及系统

(57)摘要

一种基于GPU的SM2数字签名与验签快速实现方法及系统，通过在CPU端对待签名信息或待验签信息进行预处理，得到包含公钥私钥、随机数、压缩函数SM3预计算与GPU初始化以及查找表的预处理结果，然后在GPU端对预处理得到的结果进行Jacobian加重射影坐标系的映射后，进一步进行模运算优化处理和压缩函数优化的签名处理或验签处理。本发明实施简单，性能稳定，其运算吞吐率可达$9.1*10^5$ops，极大提高了SM2签名及验签算法的计算效率。



CN 113221193 B

1.一种基于GPU的SM2数字签名与验签快速实现方法，其特征在于，通过在CPU端对待签名信息或待验签信息进行预处理，得到包含公钥私钥、随机数、压缩函数SM3预计算与GPU初始化以及查找表的预处理结果，然后在GPU端对预处理得到的结果进行Jacobian加重射影坐标系的映射后，进一步进行模运算优化处理和压缩函数优化的签名处理或验签处理；

所述的模运算优化处理是指：通过对签名/验签任务进行的大量大整数模运算中消耗大量计算资源的模乘运算和模除运算进行优化，降低计算复杂度，其中模乘运算优化采用蒙哥马利化简算法与蒙哥马利乘法作为模乘运算的替代优化算法；模除运算优化采用采用费马小定理与扩展欧几里得算法相结合的方式作为模除运算的替代优化算法；

所述的压缩函数优化的签名处理或验签处理是指：采用GPU结构优化技术对SM2算法中的杂凑算法SM3进行优化，具体包括：指令优化与寄存器复用，其中指令优化使用OpenCL内置的bitselect函数和rotate函数对SM2算法中采用的SM3杂凑算法运算的逻辑运算和循环移位运算进行优化；寄存器复用在SM3杂凑算法运算进行64步消息扩展阶段，采用16个字的寄存器空间复用64个字；

所述的签名处理由CPU端预生成的查找表配合在GPU端进行梳状签名法运算以得到签名结果，验签处理在GPU端通过二进制展开法运算以得到验签结果。

2.根据权利要求1所述的基于GPU的SM2数字签名与验签快速实现方法，其特征是，所述的预处理是指：在CPU端对①SM2数字签名任务或②验签任务进行GPU初始化以及公钥私钥、随机数、压缩函数SM3的预计算，以方便后续GPU端的运算；对③签名任务，进一步进行[k]倍点查找表的预计算以提高后续GPU签名处理的效率。

3.根据权利要求1或2所述的基于GPU的SM2数字签名与验签快速实现方法的系统，其特征在于，包括：用于降低整体运算复杂度的椭圆曲线表达方程坐标系映射模块、用于获取模乘运算、模除运算的最佳运行效率的模运算计算复杂度优化模块、用于实现消息杂凑算法的性能优化的压缩函数优化模块和用于提高签名与验签运算效率的椭圆曲线多倍点运算优化模块，其中：椭圆曲线表达方程坐标系映射模块接收CPU端预处理的数据信息进行行椭圆曲线的坐标轴映射处理并输出映射后的椭圆曲线至后续签名/验签任务的计算，模运算计算复杂度优化模块负责对椭圆曲线多倍点运算中包含的大量大整数模运算进行优化处理并高速输出大整数模运算的结果至椭圆曲线多倍点运算优化模块，压缩函数优化模块负责对SM2签名/验签任务中的SM3压缩函数运算进行优化处理并输出SM3压缩运算结果至椭圆曲线多倍点运算优化模块，椭圆曲线多倍点运算优化模块接收映射后的椭圆曲线信息进行SM2签名/验签任务中多倍点运算的优化处理，结合大整数模运算结果与SM3杂凑运算结果输出最终SM2签名/验签结果。

4.根据权利要求3所述的基于GPU的SM2数字签名与验签快速实现方法的系统，其特征是，所述的椭圆曲线表达方程坐标系映射模块在进行椭圆曲线的坐标轴映射过程中，通过Jacobian加重射影坐标系进行群倍点运算，从而避开仿射坐标系下大量出现的模逆运算，其中Jacobian加重射影坐标系是指：F_p上的椭圆曲线方程在标准射影坐标系下表示简化为$y^2=x^3+axz^4+bz^6$，其中$a,b\in F_p$，且$4a^3+27b^2\neq 0 \mod p$；椭圆曲线上的点集记为：$E(F_p)=\{(x,y,z)|x,y,z\in F_p\}$，且满足曲线方程$y^2=x^3+axz^4+bz^6$，对于$(x_1,y_1,z_1)$和$(x_2,y_2,z_2)$，当存在某个$u\in F_p$且$u\neq 0$，使得：$x_1=u^2x_2$，$y_1=u^3y_2$，$z_1=uz_2$，则称这两个三元组等价，表示同一点；椭圆曲线通常用仿射坐标系表示，在仿射坐标系下椭圆曲线上的点集$E(E_p)=\{(X,Y)|X,Y\in F_p$

且满足曲线方程$Y^2=X^3+aX+b$,其中$(4a^2+27b^2)\neq 0 (\mathrm{mod}\ p)\}$,令$x=X\cdot z^2$,$y=Y\cdot z^3$,则可将椭圆曲线从仿射坐标系转化为Jacobian加重射影坐标系表示;反之当$z\neq 0$,记$X=x/z^2$,$Y=y/z^3$,则可从Jacobian加重射影坐标系转化为仿射坐标系表示:$Y^2=X^3+aX+b$;同时当$z=0$,(1,1,0)对应的仿射坐标系下的点即为无穷远点0。

5.根据权利要求3所述的基于GPU的SM2数字签名与验签快速实现方法的系统,其特征是,所述的模运算计算复杂度优化模块包括:优化模乘运算单元和优化模除运算单元,其中:优化模乘运算单元采用蒙哥马利化简算法与蒙哥马利乘法进行模乘运算的优化运算得到大整数高速模乘运算结果;优化模除运算单元采用费马小定理与扩展欧几里得算法相结合的方式,当并行签名/验签数量小于2048时,在CPU端使用费马小定理进行模除优化运算;当并行签名/验签数量大于或等于2048时,在GPU端使用扩展欧几里得算法进行模除优化运算,得到大整数高速模除运算结果。

6.根据权利要求3所述的基于GPU的SM2数字签名与验签快速实现方法的系统,其特征是,所述的压缩函数优化模块采用GPU结构优化技术,对SM2算法中的杂凑算法SM3进行优化实现,该压缩函数优化模块包括:指令优化单元和寄存器复用单元,其中:指令优化单元使用OpenCL内置的bitselect函数和rotate函数对SM3压缩函数运算的逻辑运算和循环移位运算进行优化;寄存器复用单元对SM3压缩函数运算中的64步消息扩展阶段,采用16个字的寄存器空间复用64个字,优化SM3运算过程中的空间开销。

7.根据权利要求3所述的基于GPU的SM2数字签名与验签快速实现方法的系统,其特征是,所述的椭圆曲线多倍点运算优化模块包括:用于数字签名的椭圆曲线固定点乘法运算单元和用于验签的椭圆曲线未知点乘法运算单元,其中:椭圆曲线固定点乘法运算单元针对椭圆曲线群运算中对基点G的[k]倍点问题,采用梳状法进行计算得到签名过程中固定点乘法结果,椭圆曲线未知点乘法运算单元针对不同公钥所处椭圆曲线中互不相同的基点G的[k]倍点问题,采用二进制展开法得到验签过程中未知点乘法结果。

8.根据权利要求3所述的基于GPU的SM2数字签名与验签快速实现方法的系统,其特征是,所述的梳状法具体是指:首先计算256个固定点乘,0*P,1*P,...,255*P,设标量k长为L-bit,然后将标量值k划分为字节串,即$k=k_7||k_6||k_5||k_4||k_3||k_2||k_1||k_0$,则每段字长为L/8,在计算时,仅需计算$k_i$在[k]G查找表中的预存值$[k_i]G$,然后进行累加即可,直到遍历完字节串k。

9.根据权利要求3所述的基于GPU的SM2数字签名与验签快速实现方法的系统,其特征是,所述的二进制展开法具体是指:将基点G的[k]倍点问题中的k转换为二进制字节串$k=(k_{t-1},\cdots k_1,k_0)_2$,设=G,Q=∞,然后从$k_0$到$k_{t-1}$开始遍历二进制字节串,每次遍历中先判断当前的$k_i$值,若$k_i=1$则Q←Q+P,若$k_i=0$则Q←Q,再计算P←2P并遍历到下一个$k_{i+1}$值,遍历完字节串后即有Q=[k]G。

基于GPU的SM2数字签名与验签快速实现方法及系统

技术领域

[0001] 本发明涉及的是一种信息安全领域的技术，具体是一种基于GPU的SM2数字签名与验签快速实现方法及系统。

背景技术

[0002] 在通用计算硬件上实现椭圆曲线加密(Elliptic Curve Cryptography,ECC)算法的关键是实现有限域内大整数模乘和模除运算，以及椭圆曲线群运算中的点加和倍点运算。一个常用的做法是利用Intel提供的的SSE指令集。然而，受限于CPU硬件架构，性能并不十分理想。近年来，随着GPU在通用计算方面性能不断提高，针对非对称密码算法的优化和快速实现技术也在不断发展，Fangyu Zheng等提出了充分利用GPU浮点计算能力的方法，分别对GPU平台下有限域大整数模乘、模除、模幂、模逆运算实现与算法优化进行了探讨；Wuqiong Pan,Fangyu Zheng等基于GPU，实现了一个高速ECC签名验证服务器，并对椭圆曲线群运算中点加和倍点运算进行实现与优化。但这些现有技术大多仅针对RSA和ECDSA这两种椭圆曲线公钥算法在GPU上的高速实现，目前尚未发现SM2椭圆曲线公钥算法在GPU上的高速实现方法与系统。

发明内容

[0003] 本发明针对现有技术存在的上述不足，提出一种基于GPU的SM2数字签名与验签快速实现方法及系统，实施简单，性能稳定，其运算吞吐率可达$9.1*10^5$ops，极大提高了SM2签名及验签算法的计算效率。

[0004] 本发明是通过以下技术方案实现的：

[0005] 本发明涉及一种基于GPU的SM2数字签名与验签快速实现方法，通过在CPU端对待签名信息或待验签信息进行预处理，得到包含公钥私钥、随机数、压缩函数SM3预计算与GPU初始化以及查找表的预处理结果，然后在GPU端对预处理得到的结果进行Jacobian加重射影坐标系的映射后，进一步进行模运算优化处理和压缩函数优化的签名处理或验签处理。

[0006] 所述的预处理是指：在CPU端对①SM2数字签名任务或②验签任务进行GPU初始化以及公钥私钥、随机数、压缩函数SM3的预计算，以方便后续GPU端的运算；对③签名任务，进一步进行[k]倍点查找表的预计算以提高后续GPU签名处理的效率。

[0007] 所述的签名处理由CPU端预生成的查找表配合在GPU端进行梳状签名法运算以得到签名结果，验签处理在GPU端通过二进制展开法运算以得到验签结果。

[0008] 所述的模运算优化处理是指：通过对签名/验签任务进行的大量大整数模运算中消耗大量计算资源的模乘运算和模除运算进行优化，降低计算复杂度，其中模乘运算优化采用蒙哥马利化简算法(Montgomery Reduce Algorithm)与蒙哥马利乘法(Montgomery Multiple Algorithm)作为模乘运算的替代优化算法；模除运算优化采用采用费马小定理(Fermat's Little Theorem)与扩展欧几里得算法(Extended Euclidean Inverse Algorithm)相结合的方式作为模除运算的替代优化算法。

[0009] 所述的压缩函数优化的签名处理或验签处理是指:采用GPU结构优化技术对SM2算法中的杂凑算法SM3进行优化,具体包括:指令优化与寄存器复用,其中指令优化使用OpenCL内置的bitselect函数和rotate函数对SM2算法中采用的SM3杂凑算法运算的逻辑运算和循环移位运算进行优化;寄存器复用在SM3杂凑算法运算进行64步消息扩展阶段,采用16个字的寄存器空间复用64个字。

[0010] 本发明涉及一种实现上述方法的系统,包括:用于降低整体运算复杂度的椭圆曲线表达方程坐标系映射模块、用于获取模乘运算、模除运算的最佳运行效率的模运算计算复杂度优化模块、用于实现消息杂凑算法的性能优化的压缩函数优化模块和用于提高签名与验签运算效率的椭圆曲线多倍点运算优化模块,其中:椭圆曲线表达方程坐标系映射模块接收CPU端预处理的数据信息进行行椭圆曲线的坐标轴映射处理并输出映射后的椭圆曲线至后续签名/验签任务的计算,模运算计算复杂度优化模块负责对椭圆曲线多倍点运算中包含的大量大整数模运算进行优化处理并高速输出大整数模运算的结果至椭圆曲线多倍点运算优化模块,压缩函数优化模块负责对SM2签名/验签任务中的SM3压缩函数运算进行优化处理并输出SM3压缩运算结果至椭圆曲线多倍点运算优化模块,椭圆曲线多倍点运算优化模块接收映射后的椭圆曲线信息进行SM2签名/验签任务中多倍点运算的优化处理,结合大整数模运算结果与SM3杂凑运算结果输出最终SM2签名/验签结果。

[0011] 所述的椭圆曲线表达方程坐标系映射模块在进行椭圆曲线的坐标轴映射过程中,通过Jacobian加重射影坐标系进行群倍点运算,从而避开仿射坐标系下大量出现的模逆运算。

[0012] 所述的Jacobian加重射影坐标系是指:F_p上的椭圆曲线方程在标准射影坐标系下表示简化为$y^2=x^3+axz^4+bz^6$,其中$a,b\in F_p$,且$4a^3+27b^2\neq 0 \bmod p$;椭圆曲线上的点集记为:$E(F_p)=\{(x,y,z)\mid x,y,z\in F_p$且满足曲线方程$y^2=x^3+axz^4+bz^6\}$,对于$(x_1,y_1,z_1)$和$(x_2,y_2,z_2)$,当存在某个$u\in F_p$且$u\neq 0$,使得:$x_1=u^2x_2,y_1=u^3y_2,z_1=uz_2$,则称这两个三元组等价,表示同一点。椭圆曲线通常用仿射坐标系表示,在仿射坐标系下椭圆曲线上的点集$E(E_p)=\{(X,Y)\mid X,Y\in F_p$且满足曲线方程$Y^2=X^3+aX+b$,其中$(4a^2+27b^2)\neq 0 \pmod p\}$,令$x=X\cdot z^2$,$y=Y\cdot z^3$,则可将椭圆曲线从仿射坐标系转化为Jacobian加重射影坐标系表示。反之当$z\neq 0$,记$X=x/z^2$,$Y=y/z^3$,则可从Jacobian加重射影坐标系转化为仿射坐标系表示:$Y^2=X^3+aX+b$。同时当$z=0$,$(1,1,0)$对应的仿射坐标系下的点即为无穷远点O。

[0013] 所述的模运算计算复杂度优化模块包括:优化模乘运算单元和优化模除运算单元,其中:优化模乘运算单元采用蒙哥马利化简算法(Montgomery Reduce Algorithm)与蒙哥马利乘法(Montgomery Multiple Algorithm)进行模乘运算的优化运算得到大整数高速模乘运算结果;优化模除运算单元采用费马小定理(Fermat's Little Theorem)与扩展欧几里得算法(Extended Euclidean Inverse Algorithm)相结合的方式,当并行签名/验签数量小于2048时,在CPU端使用费马小定理进行模除优化运算;当并行签名/验签数量大于或等于2048时,在GPU端使用扩展欧几里得算法进行模除优化运算,得到大整数高速模除运算结果。

[0014] 所述的压缩函数优化模块采用GPU结构优化技术,对SM2算法中的杂凑算法SM3进行优化实现,该压缩函数优化模块包括:指令优化单元和寄存器复用单元,其中:指令优化单元使用OpenCL内置的bitselect函数和rotate函数对SM3压缩函数运算的逻辑运算和循

环移位运算进行优化；寄存器复用单元对SM3压缩函数运算中的64步消息扩展阶段，采用16个字的寄存器空间复用64个字，优化SM3运算过程中的空间开销。

[0015] 所述的椭圆曲线多倍点运算优化模块包括：用于数字签名的椭圆曲线固定点乘法运算单元和用于验签的椭圆曲线未知点乘法运算单元，其中：椭圆曲线固定点乘法运算单元针对椭圆曲线群运算中对基点G的[k]倍点问题，采用梳状法进行计算得到签名过程中固定点乘法结果，椭圆曲线未知点乘法运算单元针对不同公钥所处椭圆曲线中互不相同的基点G的[k]倍点问题，采用二进制展开法得到验签过程中未知点乘法结果。

[0016] 所述的梳状法具体是指：首先计算256个固定点乘，0*P,1*P,...,255*P，设标量k长为L-bit，然后将标量值k划分为字节串，即$k=k_7||k_6||k_5||k_4||k_3||k_2||k_1||k_0$，则每段字长为L/8。在计算时，仅需计算$k_i$在[k]G查找表中的预存值$[k_i]G$，然后进行累加即可，直到遍历完字节串k。

[0017] 所述的二进制展开法具体是指：将基点G的[k]倍点问题中的k转换为二进制字节串$k=(k_{t-1},...k_1,k_0)_2$，设P=G，Q=∞，然后从k_0到k_{t-1}开始遍历二进制字节串，每次遍历中先判断当前的k_i值，若$k_i=1$则Q←Q+P，若$k_i=0$则Q←Q，再计算P←2P并遍历到下一个k_{i+1}值。遍历完字节串后即有Q=[k]G。

[0018] 技术效果

[0019] 与现有技术相比，本发明对SM2椭圆曲线公钥算法进行GPU平台下的详细分析和优化，并通过实验方式，验证了优化的积极作用，最终在GPU平台下的签名/验签吞吐量大致为$9.1*10^5$ops，与目前常用的FPGA平台吞吐量$3*10^3$ops相比，具备巨大的提升效果，这也就意味着单位时间内，本方法处理更多的SM2数字签名/验签请求。

附图说明

[0020] 图1为本发明系统结构示意图；

[0021] 图2为本发明系统模块签名/验签流程图；

[0022] 图3为查找表结构示意图。

具体实施方式

[0023] 如图1所示，为本实施例涉及一种使用GPU平台实现针对SM2椭圆曲线数字签名与验签算法的快速实现系统，包括：椭圆曲线多倍点运算优化模块、模运算计算复杂度优化模块、椭圆曲线表达方程坐标系映射模块和压缩函数优化模块。

[0024] 如图2所示，上述系统通过以下方式进行SM2椭圆曲线数字签名与验签优化算法的快速实现：

[0025] 步骤1)首先进行OpenCL平台初始化：通过OpenCL应用程序编程接口(API)选择OpenCL平台和设备，创建设备上下文，创建Kernel和初始化存储空间。

[0026] 步骤2)在CPU端进行签名/验签任务的预计算，包括公钥私钥、随机数、压缩函数SM3的预计算。对于签名任务，还需额外进行[k]倍点查找表的计算，具体为：计算256个固定点乘，0*P,1*P,...,255*P，并将数据整理为如图3所示的数据结构。

[0027] 步骤3)使用OpenCL接口函数clEnqueueReadBuffer()进行内存与GPU显存之间的数据传输，使用clFlush()和clFinish()函数进行CPU与GPU之间运算的同步。

[0028] 步骤4)使用上述系统中包含的椭圆曲线表达方程坐标系映射模块，在进行椭圆曲线的坐标轴映射过程中，选择Jacobian加重坐标系进行群倍点运算，从而避开仿射坐标系下大量出现的模逆运算。Jacobian加重坐标系的数学表示为：在F_p上的椭圆曲线方程在标准射影坐标系下表示简化为$y^2=x^3+axz^4+bz^6$，其中$a,b\in F_p$，且$4a^3+27b^2\neq 0 \bmod p$。椭圆曲线上的点集记为：$E(F_p)=\{(x,y,z)\mid x,y,z\in F_p$且满足曲线方程$y^2=x^3+axz^4+bz^6\}$。对于$(x_1,y_1,z_1)$和$(x_2,y_2,z_2)$，当存在某个$u\in F_p$且$u\neq 0$，使得：$x_1=u^2x_2$，$y_1=u^3y_2$，$z_1=uz_2$，则称这两个三元组等价，表示同一点。椭圆曲线通常用仿射坐标系表示，在仿射坐标系下椭圆曲线上的点集$E(E_p)=\{(X,Y)\mid X,Y\in F_p$且满足曲线方程$Y^2=X^3+aX+b$，其中$(4a^2+27b^2)\neq 0 \pmod p\}$，令$x=X\cdot z^2$，$y=Y\cdot z^3$，则可将椭圆曲线从仿射坐标系转化为Jacobian加重射影坐标系表示。反之当$z\neq 0$，记$X=x/z^2$，$Y=y/z^3$，则可从Jacobian加重射影坐标系转化为仿射坐标系表示：$Y^2=X^3+aX+b$。同时当$z=0$，(1,1,0)对应的仿射坐标系下的点即为无穷远点O。

[0029] 步骤5)对于签名任务，主要面临固定点乘法问题，即椭圆曲线群运算中对基点G的[k]倍点问题，采用梳状法进行计算，具体算法为：设标量k长为L-bit，然后将标量值k划分为字节串，即$k=k_7||k_6||k_5||k_4||k_3||k_2||k_1||k_0$，则每段字长为L/8。在计算时，仅需计算$k_i$在[k]G查找表中的预存值$[k_i]G$，然后进行累加即可，直到遍历完字节串k。

[0030] 步骤6)对于验签任务，主要面临未知点乘法问题，由于不同公钥所处椭圆曲线的基点G均不相同，因此并行计算时无法进行预计算处理。此时采用二进制展开法，以时间代价换取空间效率，即将基点G的[k]倍点问题中的k转换为二进制字节串$k=(k_{t-1},\ldots k_1,k_0)_2$，设$P=G$，$Q=\infty$，然后从$k_0$到$k_{t-1}$开始遍历二进制字节串，每次遍历中先判断当前的$k_i$值，若$k_i=1$则$Q\leftarrow Q+P$，若$k_i=0$则$Q\leftarrow Q$，再计算$P\leftarrow 2P$并遍历到下一个值$k_{i+1}$。遍历完字节串后即有$Q=[k]G$。

[0031] 在进行已知点乘法和未知点乘法过程中，需要进行大量的大整数模运算，其中模乘运算、模除运算均需要消耗大量的计算资源。在上述系统中包含的模运算计算复杂度优化模块主要对这两项运算进行优化，具体为：

[0032] 1)对于大整数模乘运算，采用蒙哥马利化简算法(Montgomery Reduce Algorithm)与蒙哥马利乘法(Montgomery Multiple Algorithm)进行模乘运算的替代算法，避开计算复杂度极大的模幂运算环节。

[0033] 2)对于大整数模除运算，采用费马小定理(Fermat's Little Theorem)与扩展欧几里得算法(Extended Euclidean Inverse Algorithm)相结合的方式，当并行签名/验签数量小于2048时，在CPU端使用费马小定理进行模除运算；当并行签名/验签数量大于或等于2048时，在GPU端使用扩展欧几里得算法进行模除运算。

[0034] 在进行具体操作GPU的OpenCL编码时，采用压缩函数优化模块对SM2算法中包含的SM3杂凑函数进行优化，具体方式包括：

[0035] 1)指令优化：使用OpenCL内置的bitselect函数和rotate函数对SM2算法中采用的SM3的逻辑运算和循环移位运算进行优化；

[0036] 2)寄存器复用：SM3算法中每轮16步的压缩函数仅和该轮的16个寄存器取值有关。则在进行64步消息扩展阶段，采用16个字的寄存器空间复用64个字。

[0037] 经过以上步骤，可完成SM2算法的签名/验签快速运算，运算结果使用OpenCL接口函数clEnqueueWriteBuffer()/clEnqueueReadBuffer()，即可读回内存并进行结果显示。

[0038] 综上所述，本实施例在实施过程中，使用椭圆曲线多倍点运算优化模块，提高签名与验签运算效率；使用模运算计算复杂度优化模块，获取模乘运算、模除运算的最佳运行效率；使用椭圆曲线表达方程坐标系映射模块，降低整体运算复杂度；使用压缩函数优化模块，实现消息杂凑算法的性能优化。

[0039] 本实施例在AMD R9 290系列显卡上进行实现，具体平台设置如下表所示：

[0040]

参数名称	GPU
GPU型号	AMD Radeon R9 290
架构	GCN1.1
显存容量	4GB
默认主频频率	947MHz
计算单元数目(CU)	40
流处理器数目	2560
显存带宽	320GB/s

[0041] 在此环境下，对比在CPU平台，使用Intel Xeon E5620系列的实现，在2048线程并发下，SM2签名加速比可达74.2，SM2验签加速比可达9.4；在4096线程并发下，SM2签名加速比可达144.3，SM2验签加速比可达18.2；在8192线程并发下，SM2签名加速比可达288.1，SM2验签加速比可达37.2，充分说明了基于GPU平台的SM2优化签名/验签算法的有效性。

[0042] 在此环境下，对比FPGA平台，使用XilinxVirtex-6系列芯片的实现，在8192线程并发下，SM2签名算法吞吐量加速比可达45.3，验签吞吐量加速比可达5.2，均优于FPGA平台的实现。

[0043] 上述具体实施可由本领域技术人员在不背离本发明原理和宗旨的前提下以不同的方式对其进行局部调整，本发明的保护范围以权利要求书为准且不由上述具体实施所限，在其范围内的各个实现方案均受本发明之约束。

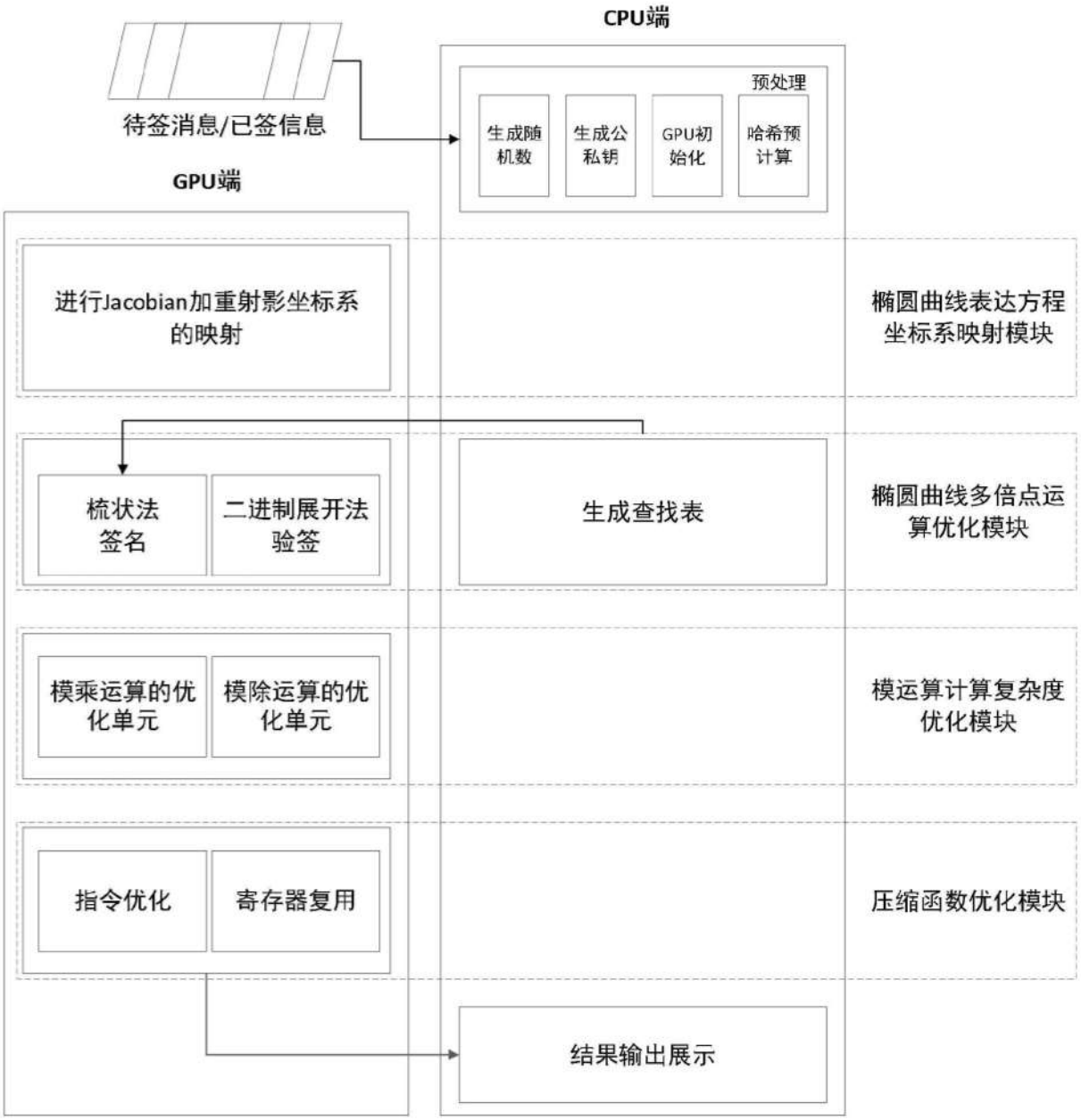

CPU端
预处理
生成随机数
生成公私钥
GPU初始化
哈希预计算
待签消息/已签信息
GPU端
进行Jacobian加重射影坐标系的映射
椭圆曲线表达方程坐标系映射模块
梳状法签名
二进制展开法验签
生成查找表
椭圆曲线多倍点运算优化模块
模乘运算的优化单元
模除运算的优化单元
模运算计算复杂度优化模块
指令优化
寄存器复用
压缩函数优化模块
结果输出展示

图1

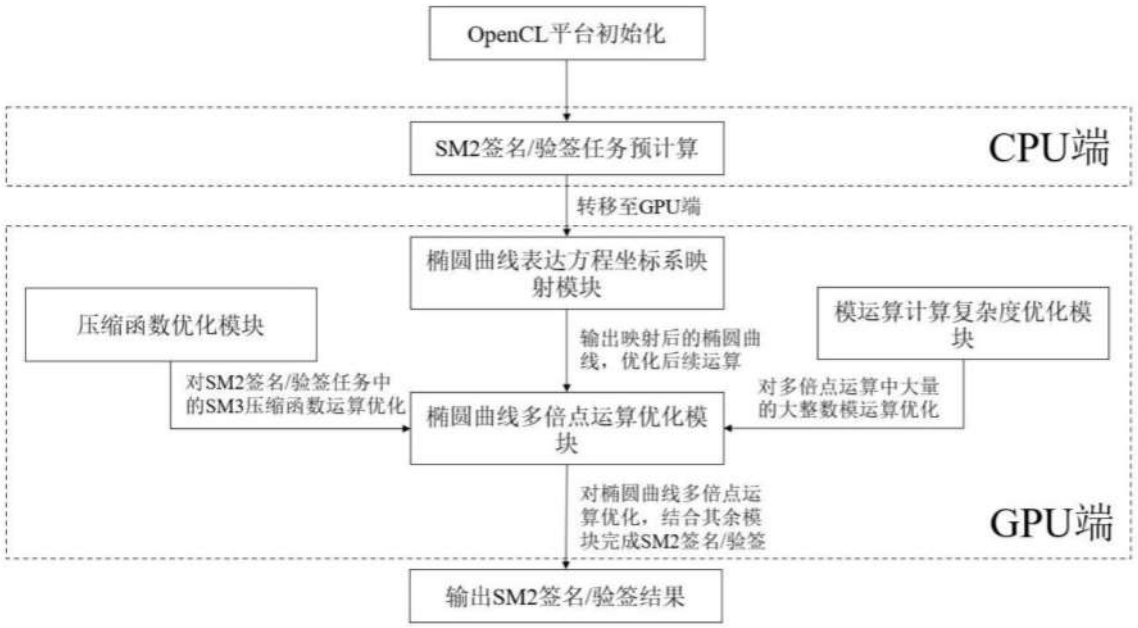

OpenCL平台初始化
SM2签名/验签任务预计算
CPU端
转移至GPU端
椭圆曲线表达方程坐标系映射模块
压缩函数优化模块
模运算计算复杂度优化模块
输出映射后的椭圆曲线，优化后续运算
对SM2签名/验签任务中的SM3压缩函数运算优化
对多倍点运算中大量的大整数模运算优化
椭圆曲线多倍点运算优化模块
对椭圆曲线多倍点运算优化，结合其余模块完成SM2签名/验签
GPU端
输出SM2签名/验签结果

图2

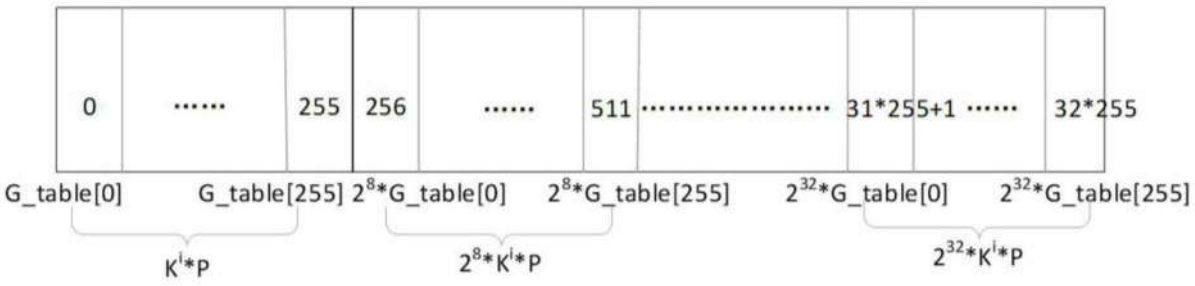

0
……
255
256
……
511
……
31*255+1
……
32*255
G_table[0]
G_table[255]
2^8*G_table[0]
2^8*G_table[255]
2^32*G_table[0]
2^32*G_table[255]
K^i*P
2^8*K^i*P
2^32*K^i*P

图3