(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 114172651 A
(43)申请公布日 2022.03.11

(21)申请号 202111347783.7

(22)申请日 2021.11.15

(71)申请人 武汉大学
地址 430072 湖北省武汉市武昌区珞珈山武汉大学

(72)发明人 彭聪 刘洋 罗敏 何德彪 崔晓晖 黄欣沂

(74)专利代理机构 武汉科皓知识产权代理事务所(特殊普通合伙) 42222
代理人 罗飞

(51)Int.Cl.
H04L 9/30(2006.01)

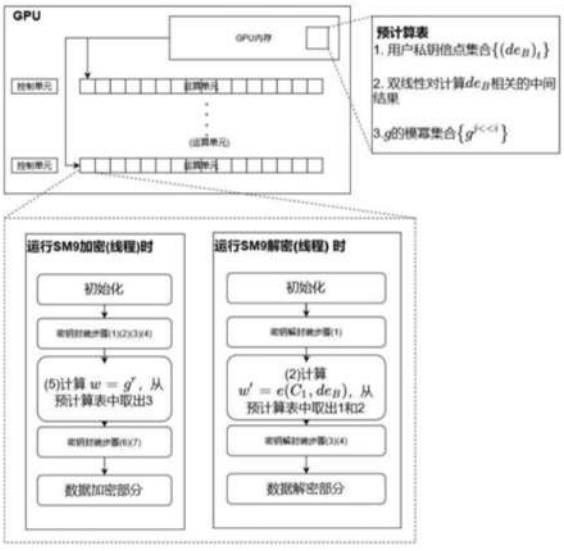
权利要求书1页 说明书5页 附图1页

(54)发明名称

一种SM9公钥加密算法、解密算法的GPU加速实现方法

(57)摘要

本发明提供了一种SM9公钥加密算法、解密算法的GPU加速实现方法，针对解密算法过程的双线性对运算中存在固定点即用户私钥的特点，通过生成包含该固定点的倍点坐标以及户私钥在双线性对运算中的中间结果的预计算表，将缓存到GPU中的预计算表与双线性对的动态输入值相结合，用于加速解密算法中双线性对计算性能，从而实现SM9解密算法的高速并行化。针对加密算法过程中的存在固定底数的特点，通过生成该固定底数的模幂运算的预计算表，将缓存到GPU中的预计算表与模幂运算的动态输入相结合，用于加速加密运算中模幂运算的计算性能，从而实现SM9加密算法的高速并行化。发明在保证计算出SM9正确运算结果的同时，实现加解密运算的高速并行实现。



CN 114172651 A

1.一种SM9公钥加密算法的GPU加速实现方法，其特征在于，包括：

S1：密文发送者根据密文接收者公钥对应的底数g的值，生成G_T群元素g模幂运算对应的预计算表，并生成的预计算表缓存到GPU，G_T为阶为素数N的乘法循环群；

S2：密文发送者根据动态输入r的值，结合缓存到GPU中的预计算表，加速加密运算。

2.如权利要求1所述的SM9公钥加密算法的GPU加速实现方法，其特征在于，步骤S1包括：

S1.1：密文发送者选择小整数m，n，使得$\log_2(m\times n)$相等或略大于$\log_2 N$，其中，N为G_1群的阶；

S1.2：密文发送者遍历i从0到n-1，计算g^{0+i*m}，g^{1+i*m}，g^{2+i*m}，...，$g^{2^m-1+i*m}$，并依次存储到预计算表T_i中；

S1.3：密文发送者将预计算表T_0，T_1，...，T_{n-1}载入到GPU内存。

3.如权利要求2所述的SM9公钥加密算法的GPU加速实现方法，其特征在于，步骤S2包括：

S2.1：每个线程根据$w=g^r$中的动态输入r的值，将r对应的比特串rb划分为n′段，使得$rb=r_{n'-1}||r_{n-2}||...||r_1||r_0$；

S2.2：根据r_i的数值，密文发送者从对应的预计算表T_i取出对应的值g^{r_i+i*m}，并通过$w=\prod_{i=0}^{n-1} g^{r_i+i*m}$实现加密算法的计算加速。

4.一种SM9公钥解密算法的GPU加速实现方法，其特征在于，包括：

S1：密文接收者根据G_2上固定点的坐标，生成G_2群上点倍运算和双线性对运算的预计算表，并生成的预计算表缓存到GPU，G_2为阶为素数N的加法循环群；

S2：密文接收者根据动态输入G_1上点的坐标，结合缓存到GPU中的预计算表，加速解密运算，G_1为阶为素数N的加法循环群。

5.如权利要4所述的SM9公钥解密算法的GPU加速实现方法，其特征在于，步骤S1包括：

S1.1：密文接收者计算双线性对$e(C,de_B)$，将每一步与de_B相关的中间结果存储到预计算表T_1，其中，C为封装密文，de_B为密文接收者的私钥；

S1.2：密文接收者将预计算表T_1载入到GPU内存。

6.如权利要5所述的SM9公钥解密算法的GPU加速实现方法，其特征在于，步骤S2包括：

每个线程根据$e(C,de_B)$的动态输入C的坐标，从预计算表T_1取出对应的中间结果，代入C的坐标进行计算，以加速双线性对计算。

一种SM9公钥加密算法、解密算法的GPU加速实现方法

技术领域

[0001] 本发明涉及密码技术领域，尤其涉及一种SM9公钥加密算法、解密算法的GPU加速实现方法。

背景技术

[0002] 在国密SM9公钥密码算法的解密算法中，需要执行解封装算法，在国密SM9公钥密码算法的加密算法中，需要执行密钥封装算法。为了保证数据的机密性，SM9公钥加密算法包括多种重量级的运算，比如椭圆曲线点乘和双线性对运算。然而在服务器和特定用户进行频繁稳定交互的场景下，用户和服务器在实际加密消息时会重复计算大量的重量级运算，这样的重复运算会浪费双方计算资源，严重降低双方通信的效率。

发明内容

[0003] 本发明提出一种SM9公钥加密算法、解密算法的GPU加速实现方法，用以解决或者至少部分解决现有技术中存在的计算效率不高的技术问题。

[0004] 为了解决上述技术问题，本发明第一方面提供了一种SM9公钥加密算法的GPU加速实现方法，包括：

[0005] S1：密文发送者根据密文接收者公钥对应的底数g的值，生成G_T群元素g模幂运算对应的预计算表，并生成的预计算表缓存到GPU，G_T为阶为素数N的乘法循环群；

[0006] S2：密文发送者根据动态输入r的值，结合缓存到GPU中的预计算表，加速加密运算。

[0007] 在一种实施方式中，步骤S1包括：

[0008] S1.1：密文发送者选择小整数m,n，使得$\log_2(m\times n)$相等或略大于$\log_2 N$，其中，N为G_1群的阶；

[0009] S1.2：密文发送者遍历i从0到n-1，计算g^{0+i*m}，g^{1+i*m}，g^{2+i*m}，…，$g^{2^m-1+i*m}$，并依次存储到预计算表T_i中；

[0010] S1.3：密文发送者将预计算表T_0，T_1，…，T_{n-1}载入到GPU内存。

[0011] 在一种实施方式中，步骤S2包括：

[0012] S2.1：每个线程根据$w=g^r$中的动态输入r的值，将r对应的比特串rb划分为n'段，使得$rb=r_{n'-1}||r_{n-2}||\cdots||r_1||r_0$；

[0013] S2.2：根据r_i的数值，密文发送者从对应的预计算表T_i取出对应的值g^{r_i+i*m}，并通过$w=\prod_{i=0}^{n-1} g^{r_i+i*m}$实现加密算法的计算加速。

[0014] 基于同样的发明构思，本发明第二方面提供了一种SM9公钥解密算法的GPU加速实现方法，包括：

[0015] S1：密文接收者根据G_2上固定点的坐标，生成G_2群上点倍运算和双线性对运算的预计算表，并生成的预计算表缓存到GPU，G_2为阶为素数N的加法循环群；

[0016] S2：密文接收者根据动态输入G_1上点的坐标，结合缓存到GPU中的预计算表，加速解密运算，G_1为阶为素数N的加法循环群。

[0017] 在一种实施方式中，步骤S1包括：

[0018] S1.1：密文接收者计算双线性对$e(C,1de_B)$，将每一步与de_B相关的中间结果存储到预计算表T_1，其中，C为封装密文，de_B为密文接收者的私钥；

[0019] S1.2：密文接收者将预计算表T_1载入到GPU内存。

[0020] 在一种实施方式中，步骤S2包括：

[0021] 每个线程根据$e(C,1de_B)$的动态输入C的坐标，从预计算表T_1取出对应的中间结果，代入C的坐标进行计算，以加速双线性对计算。

[0022] 本申请实施例中的上述一个或多个技术方案，至少具有如下一种或多种技术效果：

[0023] 本发明在不改变SM9公钥加密算法、解密算法整体架构的基础上，针对服务器和特定用户进行频繁稳定交互场景中双方公钥保持不变的特点，对数据加密过程中密钥封装算法以及数据解密过程中解封装算法进行优化，两种算法均以预计算的方式生成预计算表的方式实现优化。在优化后的密钥封装算法以及优化后的解封装算法执行过程中，通过查询预计算表，减少运算数量，通过GPU进行并行计算，实现了数据的快速加密，有效提高了加密效率。

附图说明

[0024] 为了更清楚地说明本发明实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图是本发明的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

[0025] 图1为本发明具体实施例中SM9公钥加密算法、解密算法的GPU加速实现方法的流程图。

具体实施方式

[0026] 本发明提供了一种SM9公钥加密算法、解密算法的GPU加速实现方法，在保证计算出SM9正确运算结果的同时，实现加解密运算的高速并行实现。

[0027] 为了达到上述技术效果，本发明的主要发明构思如下：

[0028] 针对加密算法过程中的存在固定底数的特点，通过生成该固定底数的模幂运算的预计算表，将缓存到GPU中的预计算表与模幂运算的动态输入相结合，用于加速加密运算中模幂运算的计算性能，从而实现SM9加密算法的高速并行化。

[0029] 针对解密算法过程的双线性对运算中存在固定点即用户私钥的特点，通过生成包含该固定点的倍点坐标以及户私钥在双线性对运算中的中间结果的预计算表，将缓存到GPU中的预计算表与双线性对的动态输入值相结合，用于加速解密算法中双线性对计算性能，从而实现SM9解密算法的高速并行化。

[0030] 为使本发明实施例的目的、技术方案和优点更加清楚，下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是

本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0031] 本发明实施例提供了一种SM9公钥加密算法的GPU加速实现方法，包括：

[0032] S1：密文发送者根据密文接收者公钥对应的底数g的值，生成G_T群元素g模幂运算对应的预计算表，并生成的预计算表缓存到GPU，G_T为阶为素数N的乘法循环群；

[0033] S2：密文发送者根据动态输入r的值，结合缓存到GPU中的预计算表，加速加密运算。

[0034] 在国密SM9公钥密码算法的解密算法中，需要执行解封装算法。例如当用户B收到封装密文C后，为了对比特长度为klen的密钥解封装，需要执行如下运算步骤：

[0035] (1)按GM/T 0044.1-2016的4.5给出的细节验证$C \in G_1$是否成立，若不成立则报错退出；

[0036] (2)计算群G_T中的元素$\omega' = e(C, 1de_B)$，按GM/T 0044.1-2016的6.2.6和6.2.5给出的细节将ω'的数据类型转换为比特串；

[0037] (3)按GM/T 0044.1-2016的6.2.6和6.2.5给出的细节将C的数据类型转换为比特串，计算封装的密钥$K' = KDF(C||\omega'||ID_B, klen)$，若K'为全0比特串，则报错并退出；

[0038] (4)输出密钥K'。

[0039] 在国密SM9公钥密码算法的加密算法中，需要执行密钥封装算法。例如为了封装比特长度为klen的密钥给用户B，作为封装者的用户A，需要执行如下运算步骤：

[0040] (1)计算群G_1中的元素$Q_B = [H_1(ID_B||hid, N)]P_1 + P_{pub-e}$；

[0041] (2)产生随机数$r \in [1, N-1]$；

[0042] (3)计算群G_1中的元素$C = [r]Q_B$，按照按GM/T 0044.1-2016的6.2.8和6.2.5给出的细节将C的数据类型转换为比特串；

[0043] (4)计算群G_T中的元素$g = e(P_{pub-e}, P_2)$；

[0044] (5)计算群G_T中的元素$\omega = g^r$按GM/T 0044.1-2016的6.2.6和6.2.5给出的细节将ω的数据类型转换为比特串；

[0045] (6)计算$K = KDF(C||\omega||ID_B, klen)$，若K为全0比特串，则返回(2)。

[0046] (7)输出(K,C)，其中K是被封装的秘钥，C是封装密文。

[0047] 为保证通用性，本申请的参数选取与SM9算法标准参数保持一致。具体符号描述如下：

[0048] A：使用SM9公钥密码系统的用户，B：使用SM9公钥密码系统的用户，P_{pub-e}：系统加密主公钥，de_B：用户B的私钥。e：从$G_1 \times G_2$到G_T的双线性对，G_T：阶为素数N的乘法循环群，G_1：阶为素数N的加法循环群。G_2：阶为素数N的加法循环群，P_1：群G_1的生成元，P_2：群G_2的生成元，ID_B：用户B的标识。KDF()：密钥派生函数，N：循环群G_1、G_2和G_T的阶，为大于2^{191}的素数，hid：用一个字节表示的私钥生成函数识别符，由KGC选择并公开。x||y：x与y的拼接，x和y是比特串或字节串。

[0049] 本发明的技术方案主要应用于国密SM9公钥密码算法软硬件实现领域。本发明实施例中，密文发送者为用户A，密文接收者为用户B。

[0050] 在一种实施方式中，步骤S1包括：

[0051] S1.1：密文发送者选择小整数m,n，使得$\log_2(m \times n)$相等或略大于$\log_2 N$，其中，N为

G_1群的阶；

[0052] S1.2：密文发送者遍历i从0到n-1，计算g^{0+i*m}，g^{1+i*m}，g^{2+i*m}，…，$g^{2^m-1+i*m}$，并依次存储到预计算表T_i中；

[0053] S1.3：密文发送者将预计算表T_0，T_1，…，T_{n-1}载入到GPU内存。

[0054] 在一种实施方式中，步骤S2包括：

[0055] S2.1：每个线程根据$w=g^r$中的动态输入r的值，将r对应的比特串rb划分为n'段，使得$rb=r_{n'-1}||r_{n-2}||\cdots||r_1||r_0$；

[0056] S2.2：根据r_i的数值，密文发送者从对应的预计算表T_i取出对应的值g^{r_i+i*m}，并通过$w=\prod_{i=0}^{n-1} g^{r_i+i*m}$实现加密算法的计算加速。

[0057] 在服务端和特定用户的加密传输过程中，用户需要对传输的信息进行SM9加密处理。用户(密文发送者)在加密过程中需要计算w，由于用户的通信对象未发生变化，w的计算过程存在固定值g，因此针对w的计算方式有优化，针对w计算过程中的固定值g进行预计算加速，具体实现包括以下步骤：

[0058] (1)用户计算固定值g的不同幂次值，生成多个预计算表；

[0059] (2)用户将这些预计算表载入到GPU内存；

[0060] (3)在SM9算法加密时，每个GPU线程根据w的动态输入r的值，从对应预计算表取出对应g的幂次值。由此用户将所有取出的值进行相乘，实现加密算法的计算加速。

[0061] 实施例二

[0062] 本发明实施例提供了一种SM9公钥解密算法的GPU加速实现方法，包括：

[0063] S1：密文接收者根据G_2上固定点的坐标，生成G_2群上点倍运算和双线性对运算的预计算表，并生成的预计算表缓存到GPU，G_2为阶为素数N的加法循环群；

[0064] S2：密文接收者根据动态输入G_1上点的坐标，结合缓存到GPU中的预计算表，加速解密运算，G_1为阶为素数N的加法循环群。

[0065] 在一种实施方式中，步骤S1包括：

[0066] S1.1：密文接收者计算双线性对$e(C,de_B)$，将每一步与de_B相关的中间结果存储到预计算表T_1，其中，C为封装密文，de_B为密文接收者的私钥；

[0067] S1.2：密文接收者将预计算表T_1载入到GPU内存。

[0068] 在一种实施方式中，步骤S2包括：

[0069] 每个线程根据$e(C,de_B)$的动态输入C的坐标，从预计算表T_1取出对应的中间结果，代入C的坐标进行计算，以加速双线性对计算。

[0070] 在服务端和特定用户的加密传输过程中，服务端对用户传输的SM9密文C，需要进行解密操作。服务端在解密过程中需要计算ω'，其中需要服务端私钥参与运算。当利用GPU进行解密时，需要对不同的密文C_i计算ω'_i，注意到每次解密过程中均使用相同的服务端私钥参与运算，因此针对双线性对的计算方式有优化，针对ω'计算过程中的服务端私钥参与部分进行预计算加速，具体实现包括以下步骤：

[0071] (1)服务端计算双线性对结果ω'时，将每一步与服务端私钥相关的中间结果存储到预计算表T_1；

[0072] (2)服务端将预计算表T_1载入到GPU内存；

[0073] (3)在SM9算法解密时，每个GPU线程根据每次接收到的封装密文C，从预计算表T_1取出对应的中间结果，代入C的坐标进行计算。由此加速双线性对计算。

[0074] 请参见图1，为本发明提供的技术方案的运算流程示意图，其中，一种SM9公钥加密算法的GPU加速实现方法中，在运行SM9加密线程时(步骤S2)，首先进行初始化操作，然后执行密钥封装算法步骤的(1)(2)(3)(4)，接着计算$w=g^r$，并从对应预计算表中取出3，然后执行密钥封装算法步骤的(6)和(7)，最终得到加密部分的结果，即输出密文。

[0075] 一种SM9公钥解密算法的GPU加速实现方法中，在运行SM9解密线程时(步骤S2)，首先进行初始化操作，然后执行密钥解封装算法步骤的(1)，接着计算$w'=e(C_1,de_B)$，并从对应预计算表中取出1和2，然后执行密钥解封装算法步骤的(3)和(4)，最终得到解密部分的结果，即输出明文。

[0076] 本发明在不改变SM9公钥算法整体架构的基础上，针对服务器和特定用户进行频繁稳定交互场景中双方公钥保持不变的特点，对数据加密过程中密钥封装算法以及数据解密过程中解封装算法进行优化，两种算法均以预计算的方式生成预计算表的方式实现优化。在优化后的密钥封装算法以及优化后的解封装算法执行过程中，通过查询预计算表，减少运算数量，通过GPU进行并行计算，实现数据的快速加密，有效提高了加密效率。

[0077] 以上实施例仅用以说明本发明的技术方案，而非对其限制；尽管参照前述实施例对本发明进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本发明各实施例技术方案的精神和范围。

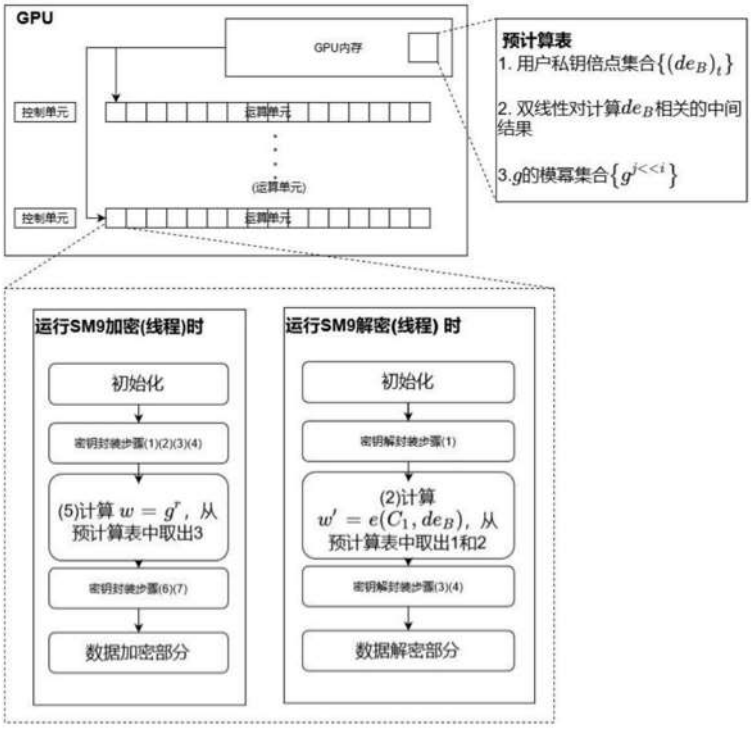

GPU
GPU内存
控制单元
运算单元
(运算单元)
控制单元
运算单元
预计算表
1. 用户私钥倍点集合$\{(de_B)_t\}$
2. 双线性对计算de_B相关的中间结果
3.g的模幂集合$\{g^{j<<i}\}$
运行SM9加密(线程)时
初始化
密钥封装步骤(1)(2)(3)(4)
(5)计算 $w = g^r$，从预计算表中取出3
密钥封装步骤(6)(7)
数据加密部分
运行SM9解密(线程) 时
初始化
密钥解封装步骤(1)
(2)计算 $w' = e(C_1, de_B)$，从预计算表中取出1和2
密钥解封装步骤(3)(4)
数据解密部分

图1