(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 119998784 A
(43)申请公布日 2025.05.13

(21)申请号 202380070581.3
(22)申请日 2023.08.23
(30)优先权数据
22192312.1 2022.08.26 EP
(85)PCT国际申请进入国家阶段日
2025.04.01
(86)PCT国际申请的申请数据
PCT/EP2023/073111 2023.08.23
(87)PCT国际申请的公布数据
WO2024/042117 EN 2024.02.29
(71)申请人 鲁汶天主教大学
地址 比利时鲁汶
(72)发明人 扬-彼得·德安弗斯
米希尔·范贝伦多克
(74)专利代理机构 成都超凡明远知识产权代理有限公司 51258
专利代理师 卫兰英
(51)Int.Cl.
G06F 7/483(2006.01)
G06F 7/499(2006.01)
G06F 7/523(2006.01)

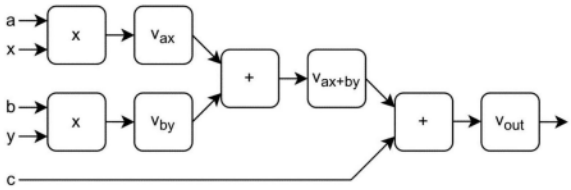
权利要求书2页 说明书12页 附图3页

(54)发明名称

用于在加密应用中执行运算的方法

(57)摘要

一种用于在加密应用中执行运算的方法,本发明涉及一种用于推导出对在加密应用中对输入操作数执行运算时使用的变量的表示进行确定的一个或更多个参数的方法,其中,执行运算所得的输出满足一个或更多个预定义约束,所述运算包括一个或更多个子任务,其中,所述运算的子任务将所述变量输出作为用于后续子任务的输入或作为运算的输出,并且其中子任务中的至少一个子任务接收所述输入操作数。该方法包括:-基于源自所述子任务中的一个子任务的噪声源的模型并且/或者基于所述一个子任务的输入的至少一个属性,确定由所述一个子任务输出的所述变量的一个或更多个属性,所述变量的所述一个或更多个属性由一个或更多个参数表征,-通过将参数中的至少一个参数固定为一值使得所述一个或更多个属性满足一个或更多个预定义约束,来确定所述变量的表示,-将对变量的表示应用在对加密应用中的运算的实现中。



CN 119998784 A

1.一种借助于包括处理装置的设备来推导出对在加密应用中对输入操作数执行运算时使用的变量的表示进行确定的一个或更多个参数的方法,其中,执行所述运算所得的输出满足一个或更多个预定义约束,所述运算包括一个或更多个子任务,其中,所述运算的子任务将所述变量输出作为用于后续子任务的输入或作为所述运算的输出,并且其中,所述子任务中的至少一个子任务接收所述输入操作数,所述方法包括:

-基于源自所述子任务中的一个子任务的噪声源的模型并且/或者基于所述一个子任务的输入的至少一个属性,确定由所述一个子任务输出的所述变量的一个或更多个属性,所述变量的所述一个或更多个属性由所述一个或更多个参数表征,

-通过将所述参数中的至少一个参数固定为一值使得所述一个或更多个属性满足所述一个或更多个预定义约束,来确定所述变量的所述表示,

-将对所述变量的所述表示应用在对所述加密应用中的所述运算的实现中。

2.根据权利要求1所述的用于确定的方法,其中,所述变量借助于包括{位宽,动态范围,整数部分的大小,小数部分的大小,小数点的位置}的一组参数中的一个或更多个参数来表征。

3.根据权利要求1或2所述的用于确定的方法,其中,一个或更多个所述子任务包括包含{乘法,加法}的子任务组中的一个或更多个。

4.根据权利要求1至3中任一项所述的用于确定的方法,所述方法包括确定由所述运算中包括的一个或更多个所述子任务引入的总误差。

5.根据前述权利要求中任一项所述的用于确定的方法,其中,一个噪声源源自移除位于所述一个子任务的输入的最低有效位侧处的位。

6.根据前述权利要求中任一项所述的用于确定的方法,其中,一个噪声源源自丢弃位于所述一个子任务的输入的最高有效位侧处的位。

7.根据前述权利要求中任一项所述的用于确定的方法,其中,所述运算是借助于快速傅里叶变换执行的乘法或乘积-累加运算,并且其中,所述输入操作数是一组多项式。

8.根据前述权利要求中任一项所述的用于确定的方法,其中,确定由所述子任务中的一个子任务输出的所述变量的所述表示包括查找由所述变量所表示的最大值。

9.根据前述权利要求中任一项所述的用于确定的方法,其中,所述参数中的一些参数被归为一组参数。

10.根据前述权利要求中任一项所述的用于确定的方法,所述方法应用在全同态加密方案中。

11.根据前述权利要求中任一项所述的用于确定的方法,其中,所述运算在可重构硬件设备上执行。

12.一种能够在可编程设备上执行的程序,所述程序包含指令,当执行所述指令时,执行根据权利要求1至11中任一项所述的方法。

13.一种设备,所述设备被布置成用于在加密应用中对输入操作数执行运算,其中,执行所述运算所得的输出满足一个或更多个预定义约束,所述运算包括一个或更多个子任务,其中,所述运算的子任务输出变量作为用于后续子任务的输入或者作为所述运算的输出,并且其中,所述子任务中的至少一个子任务接收所述输入操作数,所述设备包括处理装置,所述处理装置用于:基于源自所述子任务中的一个子任务的噪声源的模型并且/或者基

于所述一个子任务的输入的至少一个属性来确定由所述一个子任务输出的所述变量的一个或更多个属性,所述变量的所述一个或更多个属性由所述一个或更多个参数表征;以及

通过将所述参数中的至少一个参数固定为一值使得所述一个或更多个属性满足所述一个或更多个预定义约束,来确定所述变量的表示;以及在所述加密应用中执行所述运算时应用所述变量的所述表示。

14.根据权利要求13所述的设备,所述设备被实现为硬件加速计算系统。

用于在加密应用中执行运算的方法

技术领域

[0001] 本发明总体上涉及加密的领域。更具体地,本发明涉及一种用于在加密应用中执行运算的方法和设备。

背景技术

[0002] 在例如云计算等的背景下,用户将原始数据上传至不受信任的云服务器时会面临一定的风险。因此,需要提供足够的安全来保护用户的数据。在数据安全的领域中出现了一种有前途的新技术,即全同态加密(FHE),它允许人们在不了解有关所加密的数据(密文)的另外的信息的情况下对该数据执行同态计算。换言之,不需要首先对数据进行解密。多年来,执行FHE的方法已经得到改进,直至已经可以进行实际应用。

[0003] FHE算法通常在云计算服务器上执行。然而,计算速度较慢。在FHE方案中执行计算的密文数据是来自某个与方案相关的多项式环的大型多项式(长度为N)。这些多项式的典型运算包括加法和乘法。虽然加法与多项式的长度成线性关系(O(N)运算),但当使用通用的直接技术(也称为教科书式乘法)时,乘法具有二次成本(O(N^2)运算)。

[0004] FHE在实际应用中面临的主要挑战之一是其计算开销。由于FHE方案中成本最高的运算之一是多项式乘法,因此提高乘法运算速度可能会非常有助于降低计算开销。这可以通过利用多项式的特定属性来实现。各种经过充分研究的算法可以用于加速这种乘法,包括数论变换(NTT)、图姆-库克乘法(Toom-Cook multiplication)、或Karatsuba乘法。使用NTT的乘法一般是这些算法中最有效的,但它也对所使用的多项式环施加了最严格的条件。因此,它不能用于每个FHE方案。

[0005] 当底层环结构允许NTT时,FHE方案典型地使用NTT进行快速多项式乘法。不应用NTT的两个值得注意的例外是在论文“FHEW:Bootstrapping homomorphic encryption in less than a second(FHEW:在不到一秒内引导同态加密)”(L.Ducas等人,Eurocrypt,第617-640页,2015年)中公开的FHEW方案和在“TFHE:Fast Fully Homomorphic Encryption Over the Torus(TFHE:环面上的快速全同态加密)”(I.Chillotti等人,J.Cryptol.33,34-91,2020年)中描述的TFHE方案。它们两者的环结构都禁止使用NTT。相反,这些方案可以使用Toom-Cook乘法、Karatsuba乘法、或快速傅里叶变换(FFT)进行快速多项式乘法,其中,典型的FFT是最快的选项。FFT变换类似于NTT变换,但对多项式环对FFT变换的条件要求没有那么严格。FHEW和TFHE两者都支持使用同态布尔代数,例如,NAND逻辑门、XOR逻辑门、和XNOR逻辑门。

[0006] 每个FHE运算都会将一定量的噪声引入到安全所需的密文中。只要不超过某个阈值噪声水平,FHE就可以容忍这种噪声。因此,FHE方案必须定期调用引导运算来降低密文中的噪声量,使得噪声量保持在阈值噪声水平以下。TFHE和FHEW与之前的FHE方案不同,因为它们在每个同态布尔门之后调用引导运算。在TFHE方案中,优选地,引导运算保持较短。

[0007] 为了在计算机架构中表示数字,必须选择一种表示类型(例如,定点/浮点/整数)以及一组参数(例如,尾数大小、指数大小、……)。精度和动态范围是对数字可以被表示的

多么准确的度量。精度限定了使用多少位来表示数字。动态范围限定了数字可以表示的最小值或最大值。

[0008] 实数可以以多种方式用有限精度表示。在CPU上,典型的方法是使用单精度浮点数或双精度浮点数。精度由尾数的大小限定,并且动态范围由指数的大小限定。由于在CPU中对浮点单元(FPU)的集成,因此这种方法是有效的,并且因此它对于软件设计人员来说是典型的选定的表示方式。上面提及的TFHE和FHEW方案的实现已经仅限于双精度浮点FFT,因为发现单精度FFT会引入太多噪声。已经发现双精度浮点FFT将引入的噪声量保持在足够小的水平。定点表示由该表示中的位数和缩放因子决定。在定点表示中,尾数具有固定的位数。

[0009] 论文“MATCHA:A Fast and Energy-Efficient Accelerator for Fully Homomorphic Encryption over the Torus(MATCHA:一种用于环面上全同态加密的快速节能的加速器)”(L.Jiang等人,第59届年度设计自动化会议,2022年7月,2022年2月17日预发布在互联网上)也涉及上面所提及的TFHE方案。呈现了一种用于对TFHE门进行处理的硬件加速器,在效率方面,它优于频繁调用成本昂贵的双精度浮点FFT和IFFT内核的加速器。为了充分利用对TFHE的容错能力,通过使用仅需要加法和二进制移位的近似无乘法整数FFT和IFFT来加速多项式乘法。虽然近似的FFT和IFFT会在每个密文中引入误差,但由于误差可以在解密期间与噪声一起四舍五入,因此仍然可以对密文进行正确地解密。整数表示可以看作是定点表示的经过缩放以移除小数点的版本。

[0010] 当确定由FFT和IFFT中的数字表示的有限准确度引起的误差时,对于给定的数字表示(例如,浮点、定点)或给定的精度(例如,32位)会发现此误差。如果位宽或任何其他加密参数发生变化,则需要重新执行所有误差计算。因此,拥有一种可以预先计算变化的参数对所得的误差的影响的技术将是有益的。

[0011] 因此,需要一种方法,其中,每次在加密参数发生变化之后进行新的误差计算时,可以对数据的表示以灵活的方式进行调整。这不仅适用于基于FFT的乘法,而且通常适用于在加密应用中对操作数(operant)执行的任何运算。

发明内容

[0012] 本发明的实施方式的目的是提供一种方法,该方法用于确定在加密应用中对输入操作数执行运算时的变量的参数值,以及对所述运算的输出施加一个或更多个约束。

[0013] 上面的目的是通过根据本发明的解决方案来实现的。

[0014] 在第一方面中,本发明涉及一种借助于包括处理装置的设备来推导出对在加密应用中对输入操作数执行运算时使用的变量的表示进行确定的一个或更多个参数的方法,其中,执行所述运算所得的输出满足一个或更多个预定义约束。该运算包括一个或更多个子任务,其中,该运算的子任务将变量输出作为用于后续子任务的输入或作为运算的输出,并且其中,该子任务中的至少一个子任务接收所述输入操作数,该方法包括:

[0015] -基于源自所述子任务中的一个子任务的噪声源的模型并且/或者基于所述一个子任务的输入的至少一个属性,确定由所述一个子任务输出的变量的一个或更多个属性,所述变量的所述一个或更多个属性由所述一个或更多个参数表征,

[0016] -通过将参数中的至少一个参数固定为一值使得所述一个或更多个属性满足一个或更多个预定义约束,来确定所述变量的所述表示,

[0017] -将对变量的所述表示应用在对加密应用中的运算的实现中。

[0018] 所提出的方法呈现了一种程序,该程序用于预先导出用于表示执行运算时使用的变量的一个或更多个参数值。所提出的解决方案确实允许在给定对运算的输出所施加的约束的条件下,获得具有最佳或接近最佳的值的变量的表示。变量是作为运算的某个子任务的结果而获得的中间变量,并且该变量用作用于后续子任务的输入,或者该变量是作为运算的输出的变量。在优选实施方式中,针对在所考虑的运算中发挥作用的所有变量重复该方法步骤,如下面将描述的。

[0019] 在该方法的第一步骤中,确定在子任务输出处获得的变量的一个或更多个属性,即表征变量的属性的相关参数。一个或更多个属性与评估预定义约束相关。在一些实施方式中,这基于子任务的输入处的(一个或更多个)变量的至少一个属性来完成。在其他实施方式中,对属性的确定是基于源自所考虑的子任务的噪声源的模型的。在又一其他实施方式中,将两个选项组合在一起,即噪声模型和子任务的输入的一个或更多个属性都用于查找针对所考虑的变量的一个或更多个相关属性(参数)。在下一步骤中,通过固定参数值使得满足对运算的输出所施加的约束,来确定表示变量的一个或更多个参数。在优选实施方式中,这通过解决优化问题来实现。然后,如将在前面的方法步骤中固定的,将具有一个或更多个参数的变量的表示应用在计算机实现的加密应用中的运算的实际实现中。

[0020] 与现有技术解决方案相比,本发明提供了几个优势。当确定满足一个或更多个所施加的约束的变量的合适表示时,它允许以比传统解决方案快得多的方式扫描可能参数值的搜索空间。实际上,在现有技术中需要迭代方法,其中在每次迭代中对另一参数值进行测试。相反,在本发明中,采用一次性方法来获得针对参数的值。所提出的方法还允许更快地查找针对参数的最佳值或至少该最佳值的良好近似值。本发明的方法适用于快速原型设计。

[0021] 在优选实施方式中,所述变量借助于包括{位宽,动态范围,整数部分的大小,小数部分的大小,小数点的位置,即整数部分与小数部分之间的间隔}的一组参数中的一个或更多个参数来表征。

[0022] 优选地,所述运算的一个或更多个子任务是乘法和/或加法。

[0023] 有利地,该方法包括确定由一个或更多个子任务引入的总误差的步骤。

[0024] 在一些实施方式中,一个噪声源源自移除位于一个子任务的输入的最低有效位侧处的位。在其他实施方式中,一个噪声源可能源自丢弃位于子任务的输入的最高有效位侧处的位。注意的是,子任务的输入也可以看作是前一子任务的输出。

[0025] 有利地,所述运算是借助于快速傅里叶变换执行的乘法或乘积-累加运算,并且输入操作数是一组多项式。

[0026] 在另一实施方式中,确定由所述子任务中的一个子任务输出的变量的表示可以包括查找由该变量所表示的最大值。

[0027] 在一个实施方式中,所述参数中的一些参数被归为一组参数。这在有大量参数时尤其地有利。然后,分组可能有助于减少模型中的参数的数目。

[0028] 在优选实施方式中,如所描述的方法应用在全同态加密方案中。

[0029] 在另一优选实施方式中,该运算在可重构硬件设备上执行。

[0030] 本发明的另一方面涉及一种可在可编程设备上执行的程序,该程序包含指令,当

执行该指令时,执行如前所描述的方法。

[0031] 本发明的又一方面涉及一种设备,该设备被布置成用于在加密应用中对输入操作数执行运算,其中,执行所述运算所得的输出满足一个或更多个预定义约束,所述运算包括一个或更多个子任务,其中,所述运算的子任务输出变量作为用于后续子任务的输入或者作为所述运算的输出,并且其中,子任务中的至少一个子任务接收所述输入操作数,所述设备包括处理装置,所述处理装置用于:基于源自所述子任务中的一个子任务的噪声源的模型并且/或者基于所述一个子任务的输入的至少一个属性来确定由所述一个子任务输出的所述变量的一个或更多个属性,所述变量的所述一个或更多个属性由所述一个或更多个参数表征;以及通过将所述参数中的至少一个参数固定为一值使得所述一个或更多个属性满足所述一个或更多个预定义约束,来确定所述变量的所述表示;以及在所述加密应用中执行所述线性运算时应用所述变量的所述表示。

[0032] 在优选实施方式中,该设备被实现为硬件加速计算系统。

[0033] 为了概括本发明及其相对于现有技术的优势,上面对本发明的某些目的和优势进行了描述。当然,应理解的是,并非所有这样的目的或优势都可以根据本发明的任何特定实施方式实现。因此,例如,本领域技术人员将认识到,本发明可以以实现或优化如本文所教导的一个优势或一组优势的方式实施或执行,而不必实现本文所教导或建议的其他目的或优势。

[0034] 参照下文所描述的一个或更多个实施方式,本发明的上面和其他方面将变得明显并得以阐明。

附图说明

[0035] 现在将通过示例参照附图对本发明进行进一步描述,其中在各个图中相同的附图标记指代相同的元件。

[0036] 图1示出了用于执行线性运算ax+by+c的框图。

[0037] 图2示出了图1的方案中的中间变量和输出变量。

[0038] 图3示出了将附加的构建块引入图2的方案中。

[0039] 图4示出了示例2中用于节点处计算的分解方案。

[0040] 图5示出了FFT实现中常用的蝶形结构。

[0041] 图6示出了在第三示例中分解成子任务以及针对中间变量的附加的构建块。

[0042] 图7示出了针对使用FFT的多项式乘法的方案。

具体实施方式

[0043] 本发明将针对特定实施方式并参照特定附图进行描述,但本发明并不限于此,而仅受权利要求的限制。

[0044] 此外,说明书和权利要求书中的术语“第一”、“第二”等用于区分类似元素,而不一定用于描述时间上、空间上、排序上、或任何其他方式的顺序。应理解的是,在适当情况下,如此使用的术语可以互换,并且本文中所描述的发明的实施方式能够以除本文中所描述的或所示的顺序以外的顺序进行运算。

[0045] 将注意的是,权利要求中使用的术语“包括”不应被解释为仅限于其后所列出的装

置;它不排除其他元件或步骤。因此,应将其解释为指定所提及的所述特征、整体、步骤、或组件的存在,但并不排除一个或更多个其他特征、整体、步骤或组件或其组的存在或添加。因此,“一种包括装置A和装置B的设备”这一表述的范围不应限于仅包括组件A和组件B的设备。这意味着,就本发明而言,设备的相关组件仅是A和B。

[0046] 在整个说明书中,对“一个实施方式”或“实施方式”的引用意指结合该实施方式描述的特定特征、结构、或特性被包括在本发明的至少一个实施方式中。因此,在整个说明书中,短语“在一个实施方式中”或“在实施方式中”出现在各处并不一定都指代同一实施方式,但可以指同一实施方式。此外,如本领域的普通技术人员从本公开中明显理解的那样,特定特征、结构、或特性可以以任何合适的方式组合在一个或更多个实施方式中。

[0047] 类似地,应理解的是,在本发明的示例性实施方式的说明中,为了精简本公开内容并帮助理解一个或更多个不同的发明方面,本发明的各种特征有时会在单个实施方式、附图、或其描述中被组合在一起。然而,这种公开的方法不应被解释为反映所要求保护的发明需要比每个权利要求中明确叙述的更多的特征的意图。相反,如所附权利要求所反映的,发明方面在于比前述所公开的单个实施方式的所有特征少。因此,详细说明所附的权利要求在此明确地并入本详细说明中,其中每个权利要求其自身都作为本发明的单独实施方式。

[0048] 此外,尽管本文中所描述的一些实施方式包括了一些其他实施方式中未包括的其他特征,但不同的实施方式的特征的组合旨在本发明的范围内,并且形成不同的实施方式,如本领域技术人员所理解的。例如,在所附权利要求中,任何要求保护的实施方式可以以任何组合使用。

[0049] 应当注意的是,在描述本发明的某些特征或方面时使用特定术语不应被理解为暗示在本文中该术语被重新限定为仅限于包括与该术语相关联的本发明的特征或方面的任何特定特性。

[0050] 在本文中提供的说明中,阐述了许多具体细节。然而,可以理解的是,本发明的实施方式可以在没有这些具体细节的情况下实践。在其他情况下,为了不模糊对本说明的理解,未详细示出公知的方法、结构和技术。

[0051] 本发明公开了一种新颖的方法,该方法用于推导出对如何在加密应用(例如,全同态加密方案)中执行运算(例如,FFT)时表示变量进行确定的参数,或者用于推导出对如何表示作为该运算的结果获得的输出变量进行确定的参数。

[0052] 在寻找在加密应用中有效实现FFT运算的方式时,本发明的发明人做出了以下观察:

[0053] -某些类别的加密应用需要包含确定性数学的“噪声”的计算。因此,这些应用也可以容忍包含确定性算法噪声的近似计算。

[0054] -加密使用均匀分布的随机值工作,这限制了计算所基于的系数的动态范围。这使得对表示系数所需的必要位表示范围(即字长)进行预测变得更容易。

[0055] -加密应用中FFT所容忍的噪声取决于(自由选择的)加密参数集,而不是取决于应用约束。可以将加密参数集选择成能够容忍更多或更少的噪声。

[0056] 这些见解提供了以这样的方式执行FFT运算的机会,使得执行该运算时应用的中间变量具有优化的表示。换言之,它们形成了如此处所呈现的发明的触发因素。

[0057] 更具体地,本发明呈现了一种用于建立一个或更多个参数的方法,该一个或更多

个参数确定如何表示在加密应用中对输入操作数执行运算时使用的变量。虽然对执行这样的运算的算法的数学描述典型地假设变量的无限精度,但硬件实现和软件实现需要决定变量的具体表示类型,诸如定点、浮点、块浮点,以及所选表示的精度。这种选择会对实现的成本和准确性产生影响。

[0058] 一旦做出了针对每个变量的表示类型的选择,仍然必须决定该表示的确切参数。例如,针对定点表示,需要决定所表示的最低有效位(LSB)和最高有效位(MSB)。浮点表示由尾数和指数两者的最小可能值和最大可能值(因此,位大小(字长))参数化。例如,MSB值以是否可能发生溢出为特征。在本发明中,呈现了一种方法来确定用于表示的这些参数中的一个或更多个的有效值,以用于加密应用的实现。

[0059] 通过示例,MSB和LSB的参数化如下所示。假设值a具有8个整数位和16个小数位,以及值b具有4个整数位和8个小数位。全精度结果c=a×b则有8+4=12个整数位和16+8=24个小数位。在这种情况下,可以说c的MSB位于位置12处并且LSB位于位置-24处。在实现方式中,针对变量c执行参数化:MSBc和LSBc。通过截断MSB处的位,即执行重新缩放,(例如,选择MSBc=11),允许一定的溢出概率$P_{overflow}$。该值取决于值a和值b的分布,并且实际上所有MSB都设置成使得,例如,$P_{overflow}<2^{-64}$。通过截断LSB侧处的位,会引入由于量化而产生的一些噪声,这会增加输出噪声。注意的是,可以考虑使用任何形式的舍入来代替截断。

[0060] 重要地,变量的表示会对实现的成本产生影响,因为可能的值的范围越大使得计算成本就越高,但同时也会致使实现具有越高的准确性。针对典型的实现,将对输出变量的准确性有设计约束。这些约束是预先已知的,并且例如,可以采用所引入的最大噪声方差或具有变量溢出的最大概率(例如2^{-20}或2^{-40})的形式,但不限于此。目标是确定允许有效实现的参数,同时满足所述设计约束。

[0061] 现在提供所提出方法的一般概述,该方法应用在要实现的特定的算法上(例如,FFT),该特定的算法对加密应用中的输入操作数执行运算。输入操作数包括一个或更多个变量,取决于正在考虑的运算。例如,当子任务对被馈送至子任务的两个变量执行乘法时,这两个变量可以形成输入操作数。对运算的所得的输出的某些属性施加一个或更多个约束(例如,所引入的噪声的方差的最大值)。目标是确定对输出变量和实现方式中的中间变量的表示的有效参数。在高水平上讲,该方法包括三个主要步骤。首先,根据要确定的(一个或更多个)表示的参数来构建对关于所述输出的属性的一个或更多个约束造成影响的一个或更多个噪声源的模型。在下一步骤中,确定针对所述表示的每个参数的特定值,使得满足所有约束。优选地,针对这些参数的值以最优的方式进行确定,即,根据适当的优化函数进行确定。然后使用这些参数值在可编程(硬件)设备上实现运算。

[0062] 在一些实施方式中,该运算是线性运算,即,下述运算f,满足对于所有x和y,都有f(x+y)=f(x)+f(y)且f(cx)=cf(x),其中c为常数。

[0063] 执行运算涉及执行运算的一个或更多个子任务(例如,加法、乘法),其中子任务中的至少一个子任务在其输入处接收该运算的输入操作数。子任务中的一个子任务输出变量,该变量是该运算的所得的输出。任何不生成所考虑运算的输出的子任务都会产生一个变量,该变量在本描述中称为中间变量。任何中间变量接下来都用作一个或更多个后续子任务的输入。

[0064] 构建模型可以按如下方式执行。执行该运算的算法被分解成各种子任务,并且对

中间变量进行识别。针对每个子任务,确定属性相对于约束的输入-输出行为(例如,所引入的噪声、输入至输出的缩放因子,……)。针对这些变量中的每一个变量,选择表示的类型(即,定点、浮点,……)。此表示的参数(例如,如最高可表示值、最低可表示值,……)仍未确定,并且最初保留为符号变量,即,尚无特定值的变量。

[0065] 为了对算法的执行进行分析,人们利用了额外的构建块,该构建块被引入到运算的方案中每个中间变量的位置处。该构建块表示对变量的表示的有限精度的影响,但对算法本身没有影响。为了实现这一点,额外的构建块具有将符号参数与所施加的约束联系起来的输入-输出行为。

[0066] 然后逐步从输入到输出进行算法,并且针对每个约束构建模型。为此,确定子任务的输入处或输入操作数的输入处的一个或更多个相关属性(例如,噪声方差、输入方差、最大可能值,……),并且然后将这些属性传播至输出。针对每个子任务,使用输入-输出行为将输入属性转换成对应的输出属性。结果,针对每个约束,都将获得属性的模型,在该模型上根据符号参数施加约束。

[0067] 在该方法的下一步骤中,确定针对符号参数中的每个符号参数的值,使得满足约束。优选地,选择这些值使得实现成本尽可能低。实现这一点的一种方式是为每个符号参数选择一个成本函数,该函数为该参数的给定值建模实现成本。然后可以通过解决优化问题来查找解决方案,其中确定一值,该值在遵守所有约束的同时降低了总成本函数并且优选地使总成本函数最小化。

[0068] 最后,应用针对参数所选择的值来对用于在加密应用中执行线性运算的算法的设计进行实例化。

[0069] 第一示例

[0070] 在第一示例中,考虑一种简单的算法,该算法针对包括x值和y值的输入操作数计算表达式ax+by+c的值,其中,中间变量被表示为定点数,并且a、b、和c表示常数。图1中给出了针对此线性运算的子任务的确切序列。目标是为每个中间变量的最低有效位(LSB)的位置选择一值。约束为针对输出变量的最大噪声方差$\sigma^2_{maxnoise}$。为简单起见,在本示例中假设输入方差具有给定方差和均值为零的分布,并且假设当变量的精度降低时,使用舍入运算执行此降低。

[0071] 如上面所提及的,首先开发一个模型,允许对约束进行评估。如图1中所示的,将计算分解成子任务,并且对中间变量进行识别。针对每个子任务,需要确定相对于与一个或更多个所施加的约束相关的属性的输入-输出行为。在这个示例中,噪声和信号的最低有效位(LSB)和方差被视为相关属性。针对子任务,f(x,y)=x+y是具有LSB_{in1}和$\sigma^2_{noise,in1}$的第一输入x与具有LSB_{in2}和$\sigma^2_{noise,in2}$的第二输入y的加法,其具有输入-输出行为

[0072] $LSB_{out}=\min(LSB_{in1},LSB_{in2})$

[0073] 和噪声

[0074] $\sigma^2_{noise,out}=\sigma^2_{noise,in1}+\sigma^2_{noise,in2}$。

[0075] 针对两个输入的乘法f(x,y)=x*y,其具有输入-输出行为

[0076] $LSB_{out}=LSB_{in1}++LSB_{in2}$

[0077] 和

[0078] $\sigma^2_{noise,out}=\sigma^2_{noise,in1}*\sigma^2_{in2}+\sigma^2_{in1}*\sigma^2_{noise,in2}+\sigma^2x_{oise,in1}*\sigma^2_{noise,in2}$

[0079] (假设输入具有零均值的情况下,为简单起见,此处仅作此假设)。针对每个中间变量v(见图2,其中,中间变量已被赋予下标,表示它们由何种运算产生),其需要引入符号参数,即最低有效位LSB_v。符号参数以粗体显示,以将这些符号参数区别于其他参数或属性。

[0080] 接下来,将中间变量作为附加的构建块引入方案中,如图3中所描绘的。这些块对由于有限的表示的范围而导致的不准确性进行建模。构建一个模型,该模型是由于截断以均匀分布的最低有效位——即,被截断的LSB位是独立均匀分布的——而引入的噪声。针对每个附加的构建块,都具有以下的输入-输出行为:

[0081] $LSB_{out}=LSB_v$和

[0082] $\sigma^2_{noise,out}=\sigma^2_{noiSe,in}+ramp(2^{2LSBv}-2^{2LSBin})/12$,

[0083] 其中输入具有最低有效位LSB_{in}和噪声方差$\sigma^2_{noise,in}$,并且其中噪声方差$\sigma^2_{noise,in}$表示由于精度降低而增加的噪声水平。ramp()函数针对负的输入值返回0,而针对正的输入值返回输入值。之所以有这个函数,是因为只有当相关位被丢弃时才会出现附加的噪声,如果$LSB_v<LSB_{in}$就会发生这种情况。

[0084] 一旦对输入-输出行为进行了描述,就可以从开始到结束对模型进行检查,以确定与根据符号参数的约束相关的属性。然后可以写下各个节点处的中间计算。例如,在将a和x相乘之后,可以得出

[0085] $LSB=LSB_a+LSB_\times$

[0086] $\sigma^2=\sigma^2_a+\sigma^2_x$

[0087] $\sigma^2_{noise}=0$

[0088] 考虑到附加的构建块和对中间变量^{v}ax的表示的不准确性,可以得出

[0089] $LSB=LSB_{ax}$

[0090] $\sigma^2=\sigma^2_a+\sigma^2_\times$

[0091] $\sigma^2_{noise}=ramp(2^{2LSBax}-2^{2(LSBa+LSBx)}/12$

[0092] 针对另一分支,可以写出类似的表达式,其中将b和y相乘。将b.y添加至a.x之后,可以写出

[0093] $LSB=min(LSB_{ax},LSB_{by})$

[0094] $\sigma^2=\sigma^2_a*\sigma^2_\times+\sigma^2_b*\sigma^2_y$

[0095] $\sigma^2_{noise}=ramp(2^{2LSBax}-2^{2(LSBa+LSBx)})/12+ramp(2^{2LSBby}-2^{2(LSBb+LSBy)})/12$以及考虑到对中间变量^{v}ax+by的不准确的表示,可以获得

[0096] $LSB=LSB_{ax+by}$

[0097] $\sigma^2=\sigma^2_a*\sigma^2_x+\sigma^2_b*\sigma^2_y$

[0098] $\sigma^2_{noise}=ramp(2^{2LsBax}-2^{2(LSBa+LSBx)})/12+ramp(2^{2LSBby}-2^{2(LSBb+LSBy)})/12+ramp(2^{2LSBax+by}-2^{2min(LSBax,LSBby)})/12$该处理最终产生根据符号参数$LSB_{ax}$、$LSB_{by}$、$LSB_{ax+by}$、和$LSB_{out}$的$\sigma^2_{noise,out}$的值。请记住,该值被限制为低于或等于输出处的最大噪声方差$\sigma^2_{maxnoise}$。

[0099] 在该方法的下一步骤中,确定符号参数的值。在前一步骤中推导出约束函数$\sigma^2_{noise,out}(LSB_{ax},LSB_{by},LSB_{ax+by},LSB_{out})\leq\sigma^2_{maxnoise}$。现在参数都应当各自被固定为一值,使得满足该约束。实现此目的的一种方法是构建成本函数,例如,所有参数都根据其位宽被同等地计算成本的函数,这将导致成本函数:$LSB_{ax}+LSB_{by}+LSB_{ax+by}+LSB_{out}$。然后可以使用在给定约束下对成本函数进行优化的优化器来查找有效的参数值。在该方法的一些实施方式

中,当然可以将成本函数更改为更接近表示实现成本的函数。

[0100] 人们可能寻找的不同的成本函数例如是在FPGA上的实现,其中乘法是在专用DSP块中执行的。在这种情况下,每个DSP块具有可用的18位的乘法,其以使用多个这些DSP块为代价,可以被扩展至更多位。如果位宽(MSB-LSB)在1至18之间,则乘法的成本为1,并且如果位宽在19至36之间,则乘法的成本为3。

[0101] 在存在噪声源的数目爆炸风险的实施方式中,可以通过将具有(近似)相似属性的噪声源加在一起来将它们捆绑在一起。这使得要考虑的变量的数目更少。

[0102] 第二示例

[0103] 考虑与第一示例中相同的算法来计算ax+by+c,但现在添加了另一约束,即没有变量溢出,并且添加了最高有效位NSB_v作为需要确定的符号参数。下面讨论了该方法中考虑该额外的约束所需的额外的步骤。

[0104] 添加了与溢出约束相关的输入-输出行为。对以下的两个额外的属性保持跟踪:变量的最大绝对值(MAV)和所有先前溢出条件的列表。为每个中间变量制定一个这样的列表,并且该列表表示为避免溢出需要满足的关于该变量的条件。然后,关于这些属性的输入-输出行为如下。针对加法,$MAV_{out}=MAV_{op1}+MAV_{op2}$,并且输入的溢出条件列表被组合成新的溢出条件组合列表。针对乘法,$MAV_{out}=MAV_{op1}*MAV_{op2}$,并且与之前一样,输入的溢出条件的列表被组合成具有溢出条件的新的列表。

[0105] 接下来必须添加中间变量构建块的输入-输出行为。此处有$MAV_{out}=MAV_{in}$并且条件$MSB_v \geq \log_2(MAV_{in})$被添加至溢出条件的列表。

[0106] 图4示出了一些属性的计算。a和x相乘之后,可以得出

[0107] $MAV=MAV_a*MAV_x$

[0108] List=[]

[0109] 包括附加的构建块V_{ax}之后,就变成了:

[0110] $MAV=MAV_a*MAV_x$

[0111] $List=[MSB_{ax} \geq \log_2(MAV_a*MAV_x)]$

[0112] 同样,针对执行b和y相乘的分支,可以写出类似的表达式。

[0113] 一旦已经求出和a.x+b.y,就会得到以下表达式

[0114] $MAV=MAV_a*MAV_x+MAV_b*MAV_y$

[0115] $List=[MSB_{ax} \geq \log_2(MAV_a*MAV_x), MSB_{by} \geq \log_2(MAV_b*MAV_y)]$

[0116] 针对Vax+by的附加块之后:

[0117] $MAV=MAV_a*MAV_x+MAV_b*MAV_y$

[0118] $List=[MSB_{ax} \geq \log_2(MAV_a*MAV_x), MSB_{by} \geq \log_2(MAV_b*MAV_y)]$

[0119] $LSB_{ax+by} \geq \log_2(MAV_a*MAV_x+MAV_b*MAV_y)]$。

[0120] 接下来,同样确定针对符号参数的值。这可以通过构建成本函数来完成,如上面已经讨论的。

[0121] 上面所描述的方法可以推广至其他约束、输入分布、或情况。例如,可以考虑其他约束,诸如每个变量处的最大溢出概率$P_{overflow,max}$。针对此约束,然后对整个算法中的变量的分布保持跟踪,并且在每个中间变量v处添加条件$P[v>2^{MSBv}]<P_{overflow,max}$。此条件的简化版本可以仅对变量的方差进行跟踪(并且可能对变量的均值进行跟踪)以及对其分布做出

假设,其中典型地由于中心极限定理假设高斯分布。

[0122] 第三示例

[0123] 本发明的方法的一个值得注意的应用领域是借助于快速傅里叶变换(FFT)执行多项式乘法。采用FFT方案允许以加速的方式执行多项式乘法。基于FFT的乘法本身是一种众所周知的技术,并且依赖于卷积定理:

[0124] $c=a\times b=FFT^{-1}(FFT(a).FFT(b))$

[0125] 其中c、a、和b表示多项式。基于FFT的乘法最近也正以其方式被用于加密应用,例如,被用于全同态加密(FHE)中。基于FFT的乘法通过使用FFT将输入多项式转换成另一表示来工作。在这个域中,乘法运算可以逐点执行(N次运算)。之后,需要使用逆FFT(IFFT)将结果转换回初始表示。FFT和IFFT转换运算典型地是基于FFT的乘法中最昂贵的运算,需要O(N.log(N))次运算,其中N是多项式中的系数的数目。系数的数目确定了FFT的深度和宽度以及其他参数。

[0126] 基于FFT的乘法对复数进行运算,其中实部和虚部两者都是实数,而其他乘法算法使用整数。当使用有限精度来表示实数时,乘法的计算并不总是精确的并且可能存在噪声,即,可能引入小的误差δ:

[0127] $FFT^{-1}(FFT(a).FFT(b))=c+\delta$

[0128] 由于FHE的特殊性质,可以容忍由FFT引入的一定程度的噪声。在FHE中,公式中已经存在安全所需的(数学)噪声。FHE本质上可以容忍一定程度的这种数学噪声,并且由于使用FFT而产生的附加的噪声也可以容忍直至一定程度。这意味着需要非常仔细地考虑噪声δ的大小。FHE的实现对所引入的噪声施加了严格的限制。如果由FFT在多项式乘法中引入了过多的噪声,计算就会失败并返回不正确的结果。

[0129] NTT是FFT的精确变型,其中$\delta=0$,但是,如已经提及的,对加密方案的参数施加了限制。

[0130] FFT变换的重要部分是蝶形运算,如图5的图示中所描绘的。具有其两个输入和两个输出的蝶形运算在FFT算法的实现中是众所周知的,并且将复合大小$n=rm$的离散傅里叶变换递归地分解成大小为m的r个较小变换,其中r是变换的基数。然后经由大小为r的蝴蝶将这些较小的DFT组合起来,蝴蝶本身是大小为r的DFT(在子变换的对应输出上执行m次),预乘以称为旋转因子的单位根。

[0131] 快速傅里叶变换中的变量是复数,并且因此蝶形运算中的变量也是复数。然而,典型地可以假设实部和虚部的分布和属性相同。在这种情况下,人们在分析中可以只关注实部的属性。

[0132] 在此处考虑的示例中,对蝶形结构施加了最大噪声方差约束(正好与第一示例一样)。同样,首先执行将算法划分成子任务。接下来,将针对中间变量v_{at}、v_c、v_d的附加的构建块添加至方案,如图6中所描绘的。

[0133] 乘法子任务可以通过利用旋转输入属性的知识来简化。任何具有实部t_{real}和虚部t_{imag}的旋转因子t的一个感兴趣属性是$t_{real}^2+t_{imag}^2=1$。给定一个实部和虚部的方差相同的数x,将x与旋转因子t相乘不会改变方差,即$var(x.t)=var(x)$。这可以很容易地推导出如下:

[0134]
$$\begin{aligned}&\operatorname{var}\left(\operatorname{real}\left(\left(x_{real}+i\cdot x_{imag}\right)\cdot\left(t_{real}+i\cdot t_{imag}\right)\right)\right)\\&=\operatorname{var}\left(x_{real}\cdot t_{real}-x_{imag}\cdot t_{imag}\right)\\&=t_{real}^{2}\cdot\operatorname{var}(x_{real})+t_{imag}^{2}\cdot\operatorname{var}(x_{imag})\\&=\operatorname{var}(x_{real}),\end{aligned}$$

[0135] 对于乘法,输入-输出行为因此可以被描述为$LSB_{at}=LSB_a+LSB_t$和$\sigma^2_{noise,out}=\sigma^2_{noise,x}+\sigma^2_x\sigma^2_{noise,t}$(同样假设输入具有零均值的情况下)。请注意,变量in1和in2相加的加法块的输入-输出行为与之前保持不变,即:

[0136] $LSB_{out}(\min(LSB_{in1},LSB_{in2})$和$\sigma^2_{noise,out}=\sigma^2_{noise,in1}+\sigma^2_{noise,in2}$。

[0137] 然后可以从旋转输入开始对模型进行计算,由于有限的表示,其准确性降低(旋转1和-1除外)。

[0138] $LSB=LSB_t$

[0139] $\sigma^2=1$

[0140] $\sigma^2_{noise}=2^{2LSBt}/12$

[0141] 在针对中间变量v_{at}的附加的构建块之后,从而获得乘积的实部:

[0142] $LSB=LSB_{at}$

[0143] $\sigma^2=\sigma^2_a$

[0144] $\sigma^2_{noise}=\sigma^2_{noise,a}+2^{2LSBt}/12*\sigma^2_a+\text{ramp}(2^{2LSBat}-2^{2(LSBa)})/12$,对于虚部,表达式结果类似。

[0145] 在针对中间变量v_c的附加的构建块之后,可以写出:

[0146] $LSB_{out}=LSB_c$

[0147] $\sigma^2_c=\sigma^2_a+\sigma^2_b$

[0148] $\sigma^2_{noise,out}=\sigma^2_{noise,c}=\sigma^2_{noise,a}+2^{2LSBt}/12*\sigma^2_a+\sigma^2_{noise,b}+\text{ramp}(2^{2LSBat}-2^{2(LSBa)})/12+\text{ramp}(2^{2LSBc}-2^{2\min(LSBat,LSBb)})/12$。

[0149] 第四示例

[0150] 在第四示例中,考虑使用FFT的多项式乘法,具体用例为全同态加密方案。此运算的目标是为了对输入(多项式的向量)与引导密钥(也是多项式的向量)之间的内积进行计算。为了对这种乘法进行有效地处理,可以使用三个步骤的程序:FFT、系数乘积和累加、以及逆FFT。图7中描绘了这种程序。注意的是,与典型的基于FFT的乘法相反,图中的第二乘法项(即,引导密钥)没有经过显式FFT运算。这是因为输入是预先已知的,并且因此可以以非常高的准确性预先计算FFT,这暗示着在该方法中不需要考虑这个特定的FFT。

[0151] 首先考虑乘积-累加运算。该运算以系数方式执行,并且因此可以使用与示例1和示例2中讨论的类似的乘法运算和加法运算来建模。FFT运算和逆FFT运算主要包括多层蝶形运算。因此,将对示例3的分析应用在这些蝶形运算上产生FFT运算和IFFT运算的模型。在多项式乘法的一些实施方式中,实现可以使用不同类型的蝶形运算(基数2、基数4、……),但对这些蝶形运算的分析可以以与示例3中的分析类似的方式进行。通过组合先前所讨论的构建块,可以构建基于全FFT的多项式乘法的噪声模型。

[0152] 一个挑战是需要确定大量的运算,并且因此需要确定大量的参数。为了减少模型中的参数的数目,可以将相似的参数组合在一起。在这个示例中,存在高度的并行性和结

构,可以用来实现这一点。例如,可以将FFT的同一“层”中的变量(即,经历了相同数目的蝶形运算的变量)的相似的参数或乘积-累加中的乘法运算之后的变量组合在一起。这将使参数的数目从大约O((V+1) N/2log2(N/2))(其中V是向量长度,N是多项式中的系数的数目)减少至约O(2log2(N/2)),因为这大约是所提出的算法中的层的数目。

[0153] 此外,将注意的是,像FHE这样的典型的加密应用需要执行负循环卷积,而不是传统的循环卷积。在循环卷积(具有N个系数)中,超出界限的系数(在位置i>N处)被循环回(cycled around)至第一系数(在位置i-N处)。相反,在负循环卷积中,这些系数不仅被循环回,而且还被取反。为了实现这一点,许多加密算法的实现在算法的开始和结束时执行所谓的扭曲(twist)和折叠步骤,这解释了负循环行为。这个扭曲和折叠步骤包括对输入的额外打包和与复数的乘法。打包采用两个整数a、b并将它们组合成复数a+bi。此操作典型地不会产生任何噪声。可以使用示例1和示例2的方法对附加的乘法运算进行建模。

[0154] 如已经提及的,必须选择针对变量的表示的类型。在例如用于执行多项式乘法的FFT方案中,有利地选择定点表示。然后可以应用上面所设置的方法来确定当执行运算时出现的变量的定点表示的参数,同时满足所施加的约束。

[0155] 给定使用根据本发明的方法获得的最佳参数,可以使用针对这些参数的定点算术构建硬件电路。实际上,具有一个参数化硬件电路实现的库可能是有利的,其中定点位宽是通用参数。可以选择与输入类型相匹配的电路,并且在“电路合成时间”处设置这些参数以与所需的输出噪声δ相匹配。

[0156] 给定最佳参数,可以使用该参数集来对硬件电路进行模拟。对输出噪声δ进行测量,并且可以将输出噪声与浮点参考实现进行比较。验证输出噪声是否满足上面确定的噪声界限(例如,标准偏差2)。可以为具有上面提出的方法中确定的最佳定点参数集的电路创建FPGA比特流。FPGA比特流允许加速FHE引导程序,该FHE引导程序涉及多项式向量乘法的许多(数千)次迭代。

[0157] 虽然本发明已在附图和前述说明中进行了详细地图示和描述,但这种图示和说明应被视为说明性或示例性的,而非限制性的。前述说明详述了本发明的某些实施方式。然而,将理解的是,无论前述文本中显示的多么详细,本发明都可以通过多种方式得以实践。本发明不限于所公开的实施方式。

[0158] 本领域技术人员通过根据对附图、公开内容、和所附权利要求的研究,在实践所要求保护的发明时,可以理解并实现对所公开的实施方式的其他变型。在权利要求书中,“包括”一词并不排除其他元件或步骤,且不定冠词“一”或“一个”不排除多个。单个处理器或其他单元可以实现权利要求中记载的多个项的功能。在相互不同的从属权利要求中记载特定措施的事实本身并不表明这些措施的组合不可被有利地利用。计算机程序可以存储/分布在合适的介质上,诸如与其他硬件一起提供的或作为其他硬件的部分提供的光学存储介质或固态介质,但也可以以其他形式分布,诸如经由互联网或其他有线或无线电信系统。权利要求书中的任何附图标记不应被解释为限制该范围。

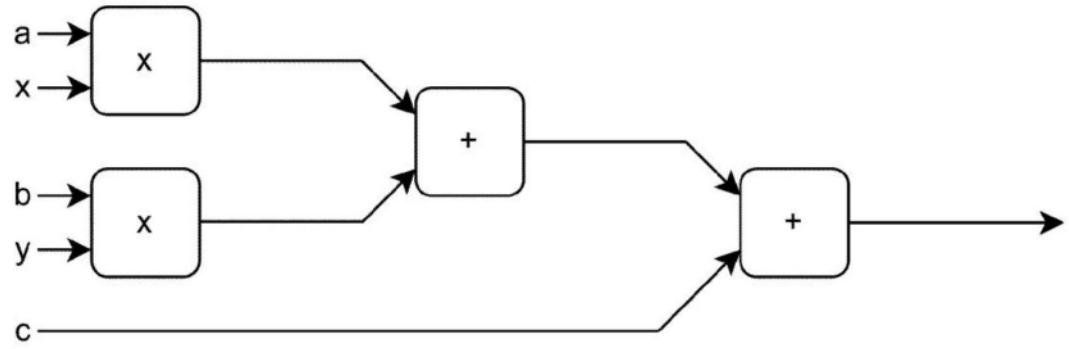

a
x
x
b
y
x
+
+
c

图1

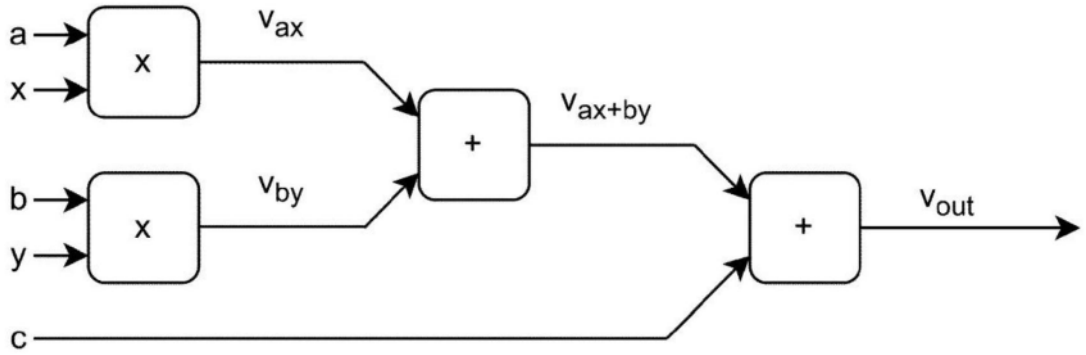

a
x
x
b
y
x
v_{ax}
v_{by}
+
v_{ax+by}
+
v_{out}
c

图2

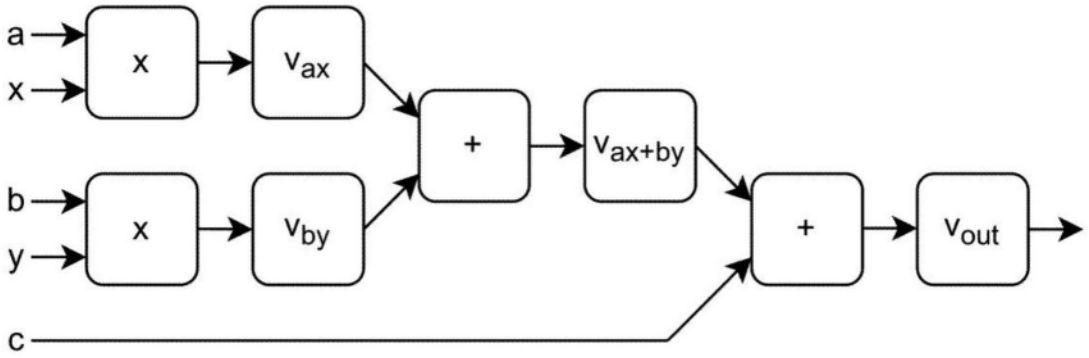

a
x
x
b
y
x
v_{ax}
v_{by}
+
v_{ax+by}
+
v_{out}
c

图3

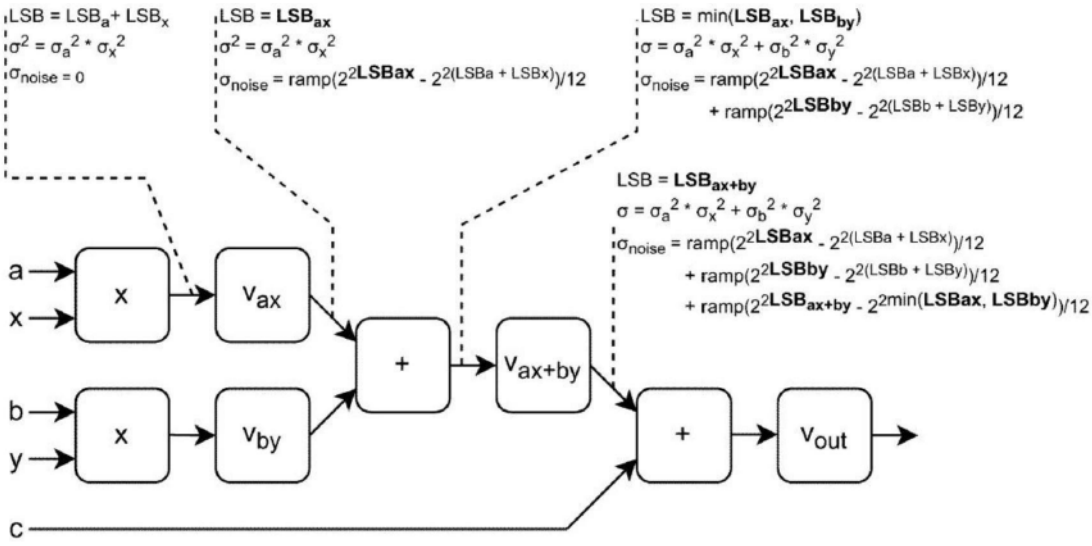

LSB = LSBa+ LSBx
σ2 = σa2 * σx2
σnoise = 0
LSB = LSBax
σ2 = σa2 * σx2
σnoise = ramp(2^2LSBax - 2^2(LSBa + LSBx))/12
LSB = min(LSBax, LSBby)
σ = σa2 * σx2 + σb2 * σy2
σnoise = ramp(2^2LSBax - 2^2(LSBa + LSBx))/12
+ ramp(2^2LSBby - 2^2(LSBb + LSBy))/12
LSB = LSBax+by
σ = σa2 * σx2 + σb2 * σy2
σnoise = ramp(2^2LSBax - 2^2(LSBa + LSBx))/12
+ ramp(2^2LSBby - 2^2(LSBb + LSBy))/12
+ ramp(2^2LSBax+by - 2^2min(LSBax, LSBby))/12
a
x
x
vax
b
y
x
vby
+
vax+by
+
vout
c

图4

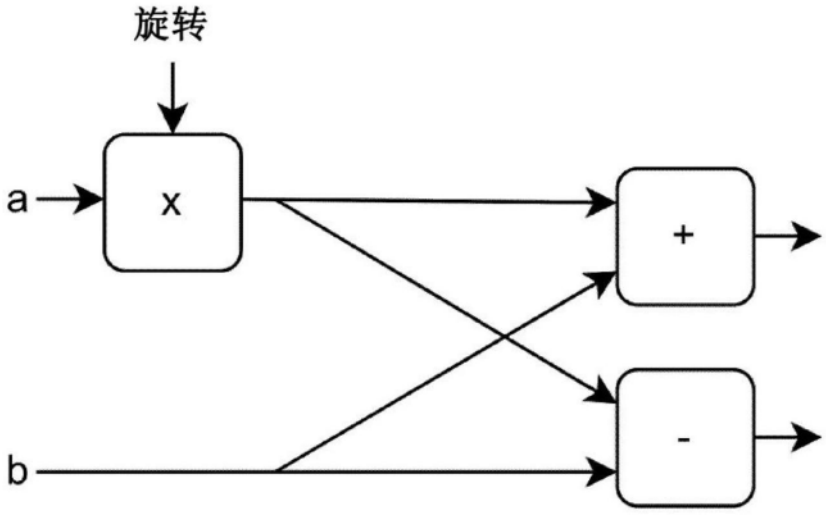

旋转
a
x
b
x
+
-

图5

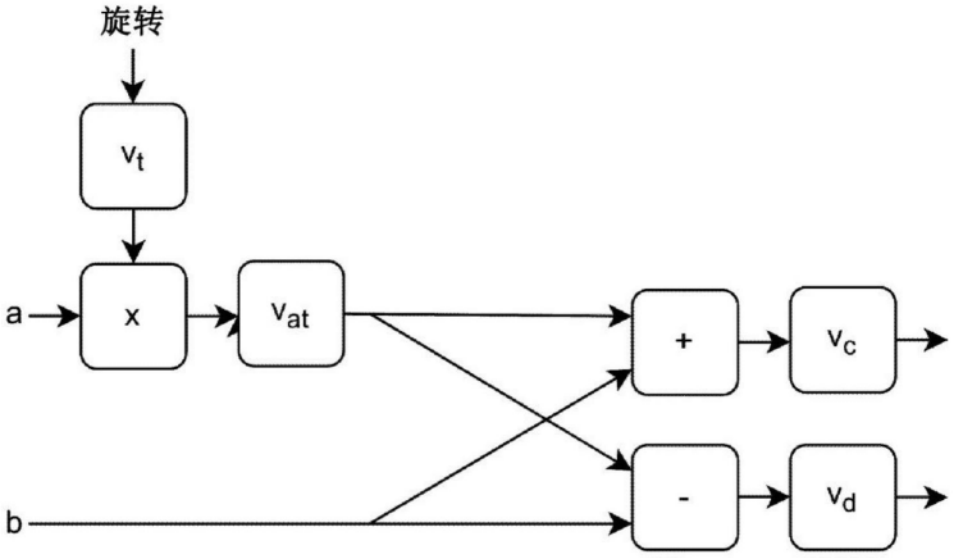


图6

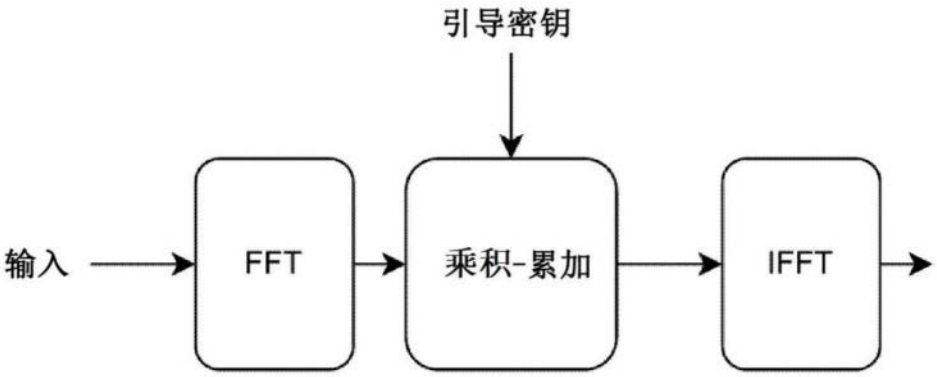


图7