(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 111480140 B
(45)授权公告日 2024.05.28

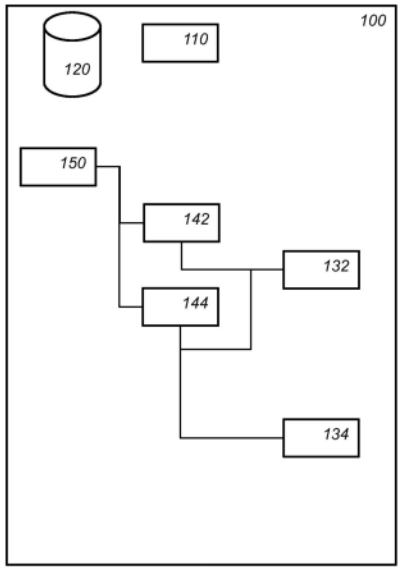
(21)申请号 201880079830.4
(22)申请日 2018.10.29
(65)同一申请的已公布的文献号
申请公布号 CN 111480140 A
(43)申请公布日 2020.07.31
(30)优先权数据
17201106.6 2017.11.10 EP
(85)PCT国际申请进入国家阶段日
2020.06.10
(86)PCT国际申请的申请数据
PCT/EP2018/079537 2018.10.29
(87)PCT国际申请的公布数据
WO2019/091809 EN 2019.05.16
(73)专利权人 皇家飞利浦有限公司
地址 荷兰艾恩德霍芬
(72)发明人 R·里特曼 S·J·A·德雷赫
(74)专利代理机构 永新专利商标代理有限公司 72002
专利代理师 刘兆君
(51)Int.Cl.
G06F 7/72(2006.01)
G06F 7/76(2006.01)
H04L 9/08(2006.01)
H04L 9/30(2006.01)
(56)对比文件
WO 2015004286 A1,2015.01.15
WO 2017063986 A1,2017.04.20
US 2016328543 A1,2016.11.10
US 2016328541 A1,2016.11.10
审查员 付苗
权利要求书2页 说明书16页 附图4页

(54)发明名称

计算设备和方法

(57)摘要

一些实施例针对一种被布置用于对乘法的混淆的运行的电子计算设备(100)。所述设备包括存储设备(120),所述存储设备被布置用于存储在算术运算的运行中使用的多个变量,所述多个变量中的变量(x;y;2)被表示为多个乘法共享($X=(x_0,x_1,\ldots,x_{m-1})$;$Y=(y_0,y_1,\ldots,y_{m-1})$;20),所述乘法共享在所述存储设备中被表示为多个加法共享($X_i=(x_{i,0},x_{i,1},\ldots,x_{i,n-1})$;$Y_i=(y_{i,0},y_{i,1},\ldots,y_{i,n-1})$;210,220)。



CN 111480140 B

1.一种被布置用于对乘法的混淆的运行的电子计算设备(100),所述设备包括:

存储设备(120),其被布置用于存储在算术运算的运行中使用的多个变量,所述多个变量中的变量(2)被表示为一个或多个乘法份额(20),所述乘法份额在所述存储设备中被表示为多个加法份额(210,220);

处理器电路,其被配置为将所述存储设备中的第一变量与所述存储设备中的第二变量相乘以获得乘法结果,所述相乘包括:

针对所述第一变量的每个乘法份额,

计算表示所述第一变量的所述乘法份额的加法份额与表示所述第二变量的对应的乘法份额的加法份额的卷积,

将得到的多个加法份额存储在所述存储设备中,作为以所述乘法结果的乘法份额的加法份额形式的表示。

2.根据权利要求1所述的电子计算设备,包括通信接口,所述通信接口被布置为获得所述第一变量和所述第二变量中的至少一个。

3.根据权利要求1或2所述的电子计算设备,其中,所述计算设备被布置用于对加密运算的所述混淆的运行,所述加密运算至少包括所述算术运算。

4.根据权利要求1所述的电子计算设备,其中,所述算术运算包括至少一个乘法和至少一个条件赋值,所述处理器电路被配置为根据二元条件将所述存储设备中的第一变量赋值给第三变量或者将所述存储设备中的第二变量赋值给所述第三变量,所述处理器电路被布置为:

根据所述二元条件来获得第一选择数字和第二选择数字,所述第一选择数字和所述第二选择数字被表示为多个加法份额,

针对所述第一变量的每个乘法份额,

计算第一选择数字与表示所述第一变量的所述乘法份额的所述加法份额的第一卷积,

计算第二选择数字与表示所述第二变量的所述对应的乘法份额的所述加法份额的第二卷积,

将所述第一卷积和所述第二卷积的结果相加,并且将所述得到的多个加法份额存储在所述存储设备中,作为以赋值结果的乘法份额的加法份额形式的表示。

5.根据权利要求4所述的电子计算设备,其中,所述第一选择数字和所述第二选择数字分别是0和1,或,分别是1和0。

6.根据权利要求4和5中的任一项所述的电子计算设备,其中,所述算术运算包括条件赋值,所述条件赋值之后是与所述条件赋值的结果的乘法。

7.根据权利要求1或2所述的电子计算设备,其中,所述算术运算包括幂运算,所述幂运算包括重复的混淆的乘法。

8.根据权利要求6所述的电子计算设备,其中,所述算术运算包括幂运算,所述幂运算包括重复的乘法,所述重复的乘法取决于指数中的位,条件赋值是根据所述指数中的所述位来运行的,所述条件赋值之后是所述重复的乘法中的乘法。

9.根据权利要求1或2所述的电子计算设备,其中,所述算术运算包括幂运算,所述幂运算是根据蒙哥马利阶梯来运行的,在所述蒙哥马利阶梯中的乘法和条件赋值被混淆。

10.根据权利要求9所述的电子计算设备,其中,所述蒙哥马利阶梯是根据以下内容来

实施的：

s←1

t←h

for i＝λ-1至0,进行

u←(1-d_i)s+d_it mod N (I)

s←su mod N (II)

t←tu mod N (II)

End for

其中,h表示所述幂运算的底数,λ表示常数,N表示模数并且位d_i表示指数的位,所述条件赋值(I)和所述乘法(II)被混淆。

11.根据权利要求7所述的电子计算设备,其中,所述处理器电路被布置为通过获得第一指数和第二指数来执行幂运算,第一指数具有比所述第二指数更少的位,由所述第一指数和所述第二指数进行的所述后续的幂运算等于通过所述指数进行的幂运算,其中,具有所述第一指数的所述幂运算包括混淆的乘法和/或条件赋值。

12.根据权利要求1或2所述的电子计算设备,其中,表示乘法份额的所述加法份额以编码形式被存储。

13.根据权利要求1或2所述的电子计算设备,其中,

乘法份额的数量是一,或者

乘法份额的数量是一个以上。

14.一种用于对乘法的混淆的运行的计算方法(400),所述方法包括：

存储(410)在算术运算的运行中使用的多个变量,所述多个变量中的变量(2)被表示为一个或多个乘法份额(20),所述乘法份额在所述存储设备中被表示为多个加法份额(210,220)；

将所述存储设备中的第一变量与所述存储设备中的第二变量相乘(420)以获得乘法结果,所述相乘包括：

针对所述第一变量的每个乘法份额,

计算(430)表示所述第一变量的所述乘法份额的加法份额与表示所述第二变量的对应的乘法份额的加法份额的卷积,

将得到的多个加法份额存储(440)在所述存储设备中,作为以所述乘法结果的乘法份额的加法份额形式的表示。

15.一种计算机可读介质(1000),其包括表示指令的瞬态或非瞬态数据(1020),所述指令使处理器系统执行根据权利要求14所述的方法。

计算设备和方法

技术领域

[0001] 本发明涉及计算设备、计算方法以及计算机可读介质。

背景技术

[0002] 运行某种加密运算的加密设备会受到攻击者在这些设备上的各种类型的威胁。在一个级别上,攻击者可能尝试攻击加密运算本身。例如,攻击者可能试图收集成对的输入和输出(例如,加密或解密的成对的输入和输出或签名运算的成对的输入和输出)并尝试从中导出秘密信息(例如,秘密密钥,例如,秘密对称密钥或秘密非对称密钥(例如,所谓的私钥))。这种类型的攻击有时被称为黑盒攻击,因为攻击者无需知晓关于密码设备中的运算的实施方式的任何信息。

[0003] 在下一个级别上,攻击者可能试图通过在设备上运行加密运算并观察该过程来获得额外的信息。特别地,攻击者可能会收集所谓的边信道信息。典型的边信道包括功率使用或设备在其运行运算时产生的电磁辐射。如果在运算设计时未考虑边信道,则设备通常容易受到边信道的攻击。已知针对多种类型的算法的某种类型的边信道攻击。这包括对称算法(例如,分块加密)和非对称算法。基于使用数学原理的算术的算法(例如,用于解密或签名的基于RSA的算法或Diffie-Hellman密钥协商算法)也容易受到边信道的攻击。防御边信道的已知对策是减小所执行的运算与边信道之间的相关性。基于边信道的攻击有时被称为灰箱攻击,因为攻击者具有某种能力来窥视执行运算的设备。边信道攻击的复杂性正在不断提高。例如,不仅个体测量值应当与秘密信息不相关,优选地,测量值的组合也应当与秘密信息不相关等。

[0004] 另外,白盒攻击是已知的规模最大的攻击,它也会激发攻击者在灰盒设置中进行更复杂的攻击。在白盒攻击中,攻击者可以完全访问设备的内部运行情况。但是,灰盒攻击与白盒攻击之间有明显的界限。例如,在某些情况下,白盒模型中最著名的攻击实际上是从边信道分析中获取的。

[0005] 保护基于算术的算法存在困难。通常,这样的算法所作用的案例数量非常大。例如,RSA算法常规地应用于2048位或4096位数字。一种用于保护这样的算法的方法是隐藏或掩蔽数字。例如,代替利用实变量x来进行存储和计算,可以替代地利用盲数字x′来进行存储和计算。在基于算术的加密算法的背景中,已经探索了两种类型的掩蔽:乘法掩蔽和加法掩蔽。在乘法掩蔽中,对于某个盲化数据c,取x′=cx;在加法掩蔽中,取x′=x+c。通常,如果要执行乘法,则乘法掩蔽更为方便,而如果需要加法,则加法掩蔽更为方便。如果需要两种类型的运算,则必须重新编码为另一种类型的掩蔽或者在运算之后执行校正计算以考虑由掩蔽引入的错误。请注意,在变量的值为0的情况下,乘法掩蔽效果不佳。这通常不是问题并且可以通过避免值0来解决。

[0006] 为此,通常使用乘法盲化来保护依赖于许多乘法的算法(例如,RSA或Diffie-Hellman中的幂运算)。但是,已经尝试在基于乘法的加密运算中使用加法掩蔽。

[0007] 在US2009/0086961中给出了第一示例。在这种方法中,通过将变量A替换为A+kn来

掩蔽变量A,其中,k是随机数,并且n是乘法的模数。由于模数(例如通过模数缩减)最终将下降,因此掩蔽对加法或乘法均无影响。在这种情况下,可以通过限制可允许的掩蔽数据来进行加法掩蔽。仅允许加上多个公共模数以进行掩蔽。这限制了能够引入的随机性。

[0008] HeeSeok Kim在“Thwarting side-channel analysis against RSA cryptosystems with additive blinding”中给出了第二示例。在该示例中,加法盲化用于保护RSA加密系统免受功率分析。在这种类型的盲化中,两个输入值A和B分别被盲化$A+M_A$为和$B+M_B$。接下来,本文将执行数学分析以计算掩蔽条件,使得乘法$(A+M_A)(B+M_B)$接收到正确且经掩蔽的结果。施加的条件是输入要具有恒定的比率,即,$\frac{B}{A}=c$,其中,c是常数。事实证明:如果唯一需要的计算是幂运算,那么能够满足该限制。因此,在该示例中,获得了完全加法盲化,但是不利之处在于仅有限数量的乘法是可能的。

[0009] Claude Carlet等人的文章“Higher-Order Masking Schemes for S-Boxes”(Lecture Notes in Computer Science,2012年)示出了一种用于将两个被表示为加法份额的变量相乘的方法,该方法适合用于特征域2,即,F2^n,它能够用于通过将对s盒的处理表示为F_2^n仿射函数的序列来对没有特定数学结构的随机s盒进行高阶掩蔽,它本身包括在F2^n上的加法、平方和标量乘法。

发明内容

[0010] 提出了一种解决这些关心问题和其他关心问题的计算设备,该计算设备在权利要求中得到定义。该计算设备能够对变量执行各种算术运算,例如,乘法和条件赋值。这些运算又能够被组合为其他安全运算,例如,安全幂运算。

[0011] 发明人洞察到了若干内容,这些内容改善了对变量的掩蔽(特别是用于混淆的算术),以例如提高对边信道攻击的抵抗力。首先,发明人创建了即使对输入进行加法掩蔽也能够正确计算乘法结果的算法。可以用值的集合来代替变量,而不是用盲化变量来代替变量。可以随机选择集合的个体成员,但是集合的加和等于表示的值。这是将数字表示为多个加法份额(additive shares)。即使被表示为加法份额的集合,该算法也能够对值进行乘法,并且产生其本身被掩蔽为加法份额的数字作为输出。该集合中的加法份额的数量能够为2,但是也能够大于2。例如,元素的数量可以是4或更大,8或更大等。实际上,更大数量的加法份额意味着边信道与实际计算之间的关系更加脆弱。因此,即使组合使用,大多数计算也可以对随机数据进行运算。只有所有加法份额的组合才与实际表示的变量有关系。

[0012] 然而,发明人发现,更复杂的表示能够与算术运算相兼容。例如,在实施例中,计算设备在两个阶段中对变量进行编码。首先,将变量编码为一个或多个乘法份额(multiplicative shares)。然后将该乘法份额独立编码为多个加法份额。仅后者需要存在于设备的存储器中。虽然能够根据表示它们的加法份额的集合来计算乘法份额,但是通常并不需要这样。例如,后者可以在解码步骤期间完成,但是对于计算的大部分(例如对于乘法、条件赋值或幂运算),并不需要计算乘法份额。

[0013] 乘法份额的数量至少为一。如果仅使用一个乘法份额,那么该乘法份额等于该数字本身。这样的实施例具有以下优点:能够容易地将数字进行相加和相乘。例如,为了将两个数字相加,可以将两个加数分量的加法份额进行相加。为了将两个数字相乘,可以使用本

文公开的算法。仅使用一个乘法份额的缺点是份额与所表示的数字之间的关系是线性的。此外,使用多个乘法份额具有以下优点:攻击者必须同时监视更多值。

[0014] 因此,在实施例中,乘法份额的数量为一。这样的实施例的示例是一种被布置用于对乘法的混淆的运行的电子计算设备。所述设备包括:

[0015] 存储设备,其被布置用于存储在算术运算的运行中使用的多个变量,所述多个变量中的变量在所述存储设备中被表示为多个加法份额,

[0016] 处理器电路,其被配置为将所述存储设备中的第一变量与所述存储设备中的第二变量相乘以获得乘法结果,所述相乘包括:

[0017] 计算表示所述第一变量的加法份额与表示所述第二变量的加法份额的卷积,

[0018] 将得到的多个加法份额存储在所述存储设备中,作为以所述乘法结果的加法份额形式的表示。

[0019] 在实施例中,乘法份额的数量大于一。使用一个以上的乘法份额的优点是份额与所表示的数字之间的关系是非线性的。这意味着试图在秘密信息与多个测量结果之间建立关联的边信道攻击更加困难,因为测量结果与秘密数据之间的关系不是线性的。例如,乘法份额的数量可以是2或更多,4或更多,8或更多等。份额的总数是乘法份额的数量与加法份额的数量的乘积。在实施例中,份额的总数至少为3。

[0020] 下面,我们通常假定使用了一个以上的乘法份额,但是这样的实施例很容易被调整到只使用一个乘法份额的情况。发明人洞察到:在这样的表示上,能够有效地计算乘法和条件赋值这两者。这样的计算既不需要重新编码成不同的编码格式,也不需要校正来校正对编码变量进行的计算结果。优选对份额的集合进行运算,以防止受到边信道攻击,因为任何并未完全知晓的份额都不会泄漏关于正在编码的信息的信息。这种属性使得该表示特别适合用于对秘密信息进行运算,例如可以在加密运算中进行运算。具有乘法和条件赋值后,能够执行安全幂运算。后者能够用于取决于具有秘密指数的幂运算的安全加密运算,例如,RSA解密、RSA签名和有限字段中的(例如具有素数元素的)Diffie-Hellman密钥协商。

[0021] 计算设备是电子设备。例如,计算设备可以是移动电子设备,例如,移动电话。计算设备可以是机顶盒、智能卡、计算机等。本文描述的计算设备和方法可以在广泛的实际应用中得到应用。这样的实际应用包括使用加密运算保护机密性或真实性的通信。

[0022] 根据本发明的方法可以在计算机上被实施为计算机实施的方法,或者可以被实施在专用硬件中,或者以这两者的组合来实施。用于根据本发明的方法的可执行代码可以被存储在计算机程序产品上。计算机程序产品的示例包括存储器设备、光学存储设备、集成电路、服务器、在线软件等。优选地,计算机程序产品包括在计算机可读介质上存储的非瞬态程序代码,该非瞬态程序代码用于在所述程序产品在计算机上被运行时执行根据本发明的方法。

[0023] 在优选实施例中,计算机程序包括计算机程序代码,该计算机程序代码适于当计算机程序在计算机上运行时执行根据本发明的方法的所有步骤。优选地,计算机程序被实施在计算机可读介质上。

[0024] 本发明的另一方面提供了一种使计算机程序可用于下载的方法。当将该计算机程序上载到例如苹果公司的App Store、谷歌公司的Play Store或微软公司的Windows Store时和当可从这样的商店下载该计算机程序时,将使用这方面的内容。

附图说明

[0025] 将仅通过举例的方式,参考附图来描述本发明的其他细节、方面和实施例。为了简单和清楚地图示了附图中的元件,并不一定按比例绘制这些元件。在附图中,与已经描述的元件相对应的元件可以具有相同的附图标记。在附图中:

[0026] 图1示意性地示出了计算设备的实施例的示例,

[0027] 图2示意性地示出了变量的表示的实施例的示例,

[0028] 图3a示意性地示出了乘法的实施例的示例,

[0029] 图3b示意性地示出了条件赋值的实施例的示例,

[0030] 图4示意性地示出了计算方法的实施例的示例,

[0031] 图5a示意性地示出了根据实施例的具有包括计算机程序的可写部分的计算机可读介质,

[0032] 图5b示意性地示出了根据实施例的处理器系统的表示。

[0033] 图1-3b、5a-5b中的附图标记列表

[0034]	100	计算设备
[0035]	110	通信接口
[0036]	120	存储设备
[0037]	132	卷积单元
[0038]	134	加法单元
[0039]	142	乘法单元
[0040]	144	条件赋值单元
[0041]	150	幂运算单元
[0042]	2	数字
[0043]	20	多个乘法份额
[0044]	21-22	乘法份额
[0045]	210、220	多个加法份额
[0046]	211-213、221-223	加法份额
[0047]	30、40	多个乘法份额
[0048]	31-33、41-43	加法份额的集合
[0049]	51、52	被表示为多个加法份额的选择数字
[0050]	350、351、352	卷积单元
[0051]	353	加法单元
[0052]	1000	计算机可读介质
[0053]	1010	可写部分
[0054]	1020	计算机程序
[0055]	1110	(一个或多个)集成电路
[0056]	1120	处理单元
[0057]	1122	存储器
[0058]	1124	专用集成电路
[0059]	1126	通信元件

[0060] 1130 互连

[0061] 1140 处理器系统

具体实施方式

[0062] 虽然本发明可以有许多不同形式的实施例,但是在附图中示出且将在本文中详细描述一个或多个特定实施例,应当理解,本公开内容被认为是本发明原理的示例,并非旨在将本发明限于所示和所述的特定实施例。

[0063] 在下文中,为了便于理解,在操作中描述了实施例的元件。然而,很明显,各个元件被布置为执行由它们执行的所述功能。

[0064] 另外,本发明不限于这些实施例,并且本发明在于本文描述的或在互不相同的从属权利要求中记载的每个新颖特征或特征组合。

[0065] 图1示意性地示出了计算设备100的实施例的示例。该计算设备被布置用于乘法的混淆的运行。使用多个份额对表示运行乘法,使得攻击者很难分析乘法。特别地,难以从边信道分析乘法。例如,该计算设备特别适合用于对应当受到保护而免受攻击的秘密信息进行运算的运算。例如,该设备适合用于加密运算,特别是使用诸如秘密密钥之类的秘密信息的加密运算。这样的加密运算的特定示例是RSA解密运算、RSA签名运算以及Diffie-Hellman密钥协商。这些运算包括使用秘密输入(在这种情况下为秘密指数)的幂运算。

[0066] 发明人已经认识到:在乘法中涉及的变量的特定表示是特别有利的。该表示组合了两种不同类型的混淆的表示以实现额外的优点。特别地,该表示允许乘法和条件赋值的有效计算。更具体地,该表示不需要将变量从一个表示重新编码为另一表示,或者不需要计算补偿数据以校正对混淆的表示而不是对常规表示的运算效果。

[0067] 通常,本文所指示的计算将是以某个模数为模,这例如取决于以某个数字为模的变量(例如这是在RSA幂运算中所要求的)。例如,可以在合适的点处插入模运算以避免(例如在每次乘法之后)数字变得太大等。不必总是将每个整数保持在模数以下。例如,实施例可以采用所谓的伪残基,该伪残基仅偶尔以模数为模而降低。该模数可以例如是素数p,或者可以是合成数,例如,两个素数的乘积pq。

[0068] 计算设备100包括存储设备120,存储设备120包括变量,计算设备100以编码形式对变量进行运算。数字x(例如,整数,例如,以模数为模的整数)最初被表示为多个乘法份额$X=(x_0,x_1,\ldots,x_{m-1})$。也就是说,数字$x_i$的乘积等于被表示的数字x。我们将用大写字母表示多个数字的序列(例如,元组)并且用小写字母表示特定的数字。我们将数字x_i称为乘法份额。注意,乘法份额通常将不被存储在存储设备120中,并且实际上通常根本不存在于存储设备120中——可能有一些暂时的例外情况,例如在将变量编码或解码到本文定义的特殊表示或者将变量从本文定义的特殊表示进行编码或解码期间的情况。换句话说,在实施例中,攻击者对设备100执行例如存储器抓取攻击,其中,存储器被梳理以找到与运算有关的变量,这将不会发现乘法份额本身。取而代之的是,乘法份额本身被表示为加法份额。例如,乘法份额x_i可以在存储设备120中(例如在设备100的存储器中)被表示为多个加法份额,例如被表示为元组$X_i=(x_{i0},x_{i,1},\ldots,x_{i,n-1})$。假定使用m个乘法份额来表示数字,每个乘法份额均被表示为n个加法份额,因此可以将该数字表示为nm个份额。表示为份额是期望的结果,因为即使完全知晓除了所有份额份以外的任何数量的份额,也不会透露关于基础数字

的任何信息。

[0069] 图2示意性地示出了变量2的表示的实施例的示例。例如,假定数字2是以素数113为模的数字94。该数字首先被表示为多个乘法份额20。图2示出了乘法份额21和22,但是可能还有更多乘法份额。例如,乘法份额可以是数字:55、40和70,因为它们以113为模数的乘积为94。生成整数的乘法表示能够通过以下操作来完成:生成m-1个随机的(例如以模数为模的)非零数字,并且计算最终的数字而使得所有乘法份额的乘积等于所表示的数字(例如,数字2)。注意,在存储设备120中不存在乘法份额20。还使用加法份额来表示乘法份额中的每个。图2示出利用多个加法份额210来表示乘法份额21,例如将乘法份额21表示为至少三个加法份额:211、212和213;并且将乘法份额22表示为多个加法份额210,例如表示为至少三个加法份额:221、222和223。可能有2个或3个以上的加法份额。通常,使用相等数量的乘法份额和相等数量的加法份额来表示要彼此相乘的数字。但是,每个乘法份额的加法份额的数量可能有所不同。例如,在实施例中,第一乘法份额是利用第一数量的加法份额表示的,并且第二乘法份额是利用第二数量的加法份额表示的。第一数量和第二数量优选大于1。通常,第一数字和第二数字是相等的,但这并不是必需的,因为它们也可能是不相等的。在实施例中,加法份额(例如,加法份额210和220)可以被存储在存储设备120中。

[0070] 例如,数字:55、40和70可以被表示为元组(78、105、98),(34、40、79),(12、81、90),因为它们以113为模的加和分别为55、40和70。在实施例中,以113为模的数字94可以因此被表示为序列78、105、98、34、40、79、12、81、90。进一步的混淆技术可以进一步应用于该序列。例如,不需要以该特定顺序存储该序列,而是可以在整个存储设备120中进行排列或分散该序列。例如,可以对加法份额进行编码。

[0071] 返回图1。计算设备100可以包括通信接口110。例如,通信接口110可以被布置为获得存储设备120中的变量中的至少一个或至少其初始值。例如,通信接口110可以被布置为接收加密的消息(例如,利用RSA加密的消息)或者用于接收用于签名的消息(例如,利用RSA签名的消息)。例如,可以使用如在RFC 3447中所定义的RSA解密或签名。例如,通信接口110可以被布置为以(如在RFC 2631中所定义的)Diffie-Hellman协议接收用于我们的公共密钥。在这三个示例的每个中,利用(至少部分为)秘密的指数来执行幂运算。

[0072] 通信接口110可以被布置为通过计算机网络与其他设备通信。计算机网络可以是互联网、内联网、LAN、WLAN等。计算机网络可以是互联网。计算机网络可以是全部或部分有线的和/或全部或部分无线的。例如,计算机网络可以包括以太网连接。例如,计算机网络可以包括无线连接,例如,Wi-Fi、ZigBee等。连接接口可以被布置为根据需要与其他设备通信。例如,连接接口可以包括连接器,例如,有线连接器(例如,以太网连接器)或无线连接器(例如,天线,例如,Wi-Fi、4G或5G天线)。通信接口110可以用于接收交易以对其进行运算,或者用于接收秘密信息,例如,秘密密钥。消息可以是例如以电子形式接收的数字消息。

[0073] 计算设备100可以包括各种单元,例如,卷积单元132、加法单元134、乘法单元142、条件赋值单元144以及幂运算单元150中的一个或多个。如果例如不需要幂运算,则可以省去幂运算单元150等。如果不需要条件赋值,则可以省去加法单元134和条件赋值单元144等。

[0074] 可以在一个或多个处理器电路中实施例如计算方法的实施例的计算设备的运行,在本文中示出了其示例。图1示出了功能单元,其可以是处理器电路的功能单元。例如,图1

可以用作处理器电路的可能的功能组织的蓝图。处理器电路未被示为与图1中的单元分开。例如,图1所示的功能单元可以全部或部分地以在设备100处(例如,在设备100的电子存储器中)存储的计算机指令来实施,并且能够由设备100的微处理器来运行。在混合式实施例中,功能单元部分地以硬件(例如,协处理器,例如,算术协处理器)来实施,并且部分地以在设备100上存储和运行的软件来实施。

[0075] 发明人洞察到:能够使用卷积来乘以如上所示的表示的数字。为此,一个或多个卷积单元132被布置为对多个加法份额的两个集合进行卷积,以例如一起表示一个乘法份额。

[0076] 乘法单元142可以被布置为将存储设备中的第一变量与存储设备中的第二变量相乘。例如,第一变量可以是被表示为多个乘法份额$X=(x_0,x_1,\ldots,x_{m-1})$的数字x,该乘法份额在存储设备中又被表示为多个加法份额$X_i=(x_{i,0},x_{i,1},\ldots,x_{i,n-1})$,其中,$0\leq i<m$。例如,第二变量可以是被表示为多个乘法份额$Y=(y_0,y_1,\ldots,y_{m-1})$的数字y,所述乘法份额在存储设备中被表示为多个加法份额$Y_i=(y_{i,0},y_{i,1},\ldots,y_{i,n-1})$,其中,$0\leq i<m$。

[0077] 为了获得如上面所表示的乘法结果$z=xy$,乘法单元142被配置为:

[0078] 针对第一变量的每个乘法份额,

[0079] 计算表示第一变量的所述乘法份额的加法份额(X_i)与表示第二变量的对应的乘法份额的加法份额(Y_i)的卷积($Z_i=X_i*Y_i$),

[0080] 将得到的多个加法份额(Z_i)存储在存储设备中,作为以乘法结果(z)的乘法份额的加法份额形式的表示。换句话说,每个多个加法份额X_i对应于多个加法份额Y_i。对对应的多个的加法份额进行卷积以获得乘法结果的表示。换句话说,可以将变量z视为被表示为多个乘法份额$Z=(z_0,z_1,\ldots,z_{m-1})$,多个乘法份额$Z=(z_0,z_1,\ldots,z_{m-1})$在存储设备中又被表示为多个加法份额$Z_i=(z_{i,0},z_{i,1},\ldots,z_{i,n-1})$,其中,$0\leq i<m$。

[0081] 下面给出了通过卷积相乘的更多细节。

[0082] 图3a示意性地示出了乘法的实施例的示例。图3a中示出了两个数字。第一数字由多个乘法份额30表示。第二数字由多个乘法份额40表示。每个乘法份额由加法份额的集合表示。在图3a中,附图标记31-33和41-43中的每个指代加法份额的集合。在这种情况下,示出了3个乘法份额,但是也可以是2个或3个以上的乘法份额。没有示出加法集合中的份额的数量,但是这可以是2个或更多个。

[0083] 图3a还示出了卷积单元350。在图3a所示的时刻,卷积单元350执行第一数字的加法集合(例如,集合31)与第二数字的加法集合(例如,集合41)之间的卷积。在该卷积之后,卷积单元350可以执行集合32与42之间的卷积。加法份额的第一集合中的份额并不需要与加法份额的第二集合中的具有相同索引的份额相对应,只要对应关系在份额之间形成双射即可。例如,卷积单元350还可以将31与42、32与43以及33与41等进行卷积。卷积能够很好地并行化,例如,在将32与42进行卷积的同时,可以将31与41进行卷积等。

[0084] 返回图1。在实施例中,计算设备100可以包括条件赋值单元144。除了卷积单元132以外,在该实施例中的条件赋值单元144还使用加法单元134。加法单元134被配置为例如通过彼此加上对应元素来加上加法份额的两个集合。通过加上两个加法集合,可以获得加和的加法表示。在仅使用一个乘法份额的情况下,加法单元134通常可以用于将两个变量相加。如果使用了一个以上的乘法份额,则需要格外小心。然而,如果需要,可以使用所谓的电路来加上具有多个乘法份额的数字。

[0085] 条件赋值单元144获取输入的两个变量(即,第一变量x和第二变量y)以及二元条件d。对于d,可以取值为0或1的常规二元变量。可以将根据d进行的运算与计算条件d的计算机程序代码集成在一起。此外,变量d可以被混淆(例如被编码)。条件赋值单元144根据条件d将第一变量赋值给其输出(例如赋值给第三变量z)或者将第二变量赋值给其输出。在实施例中,条件赋值单元144使用两个选择数字(例如,R和R′)来做到这一点。选择数字可以是常数,并且可以被存储在存储设备120中,或者可以被硬编码在单元144中等。选择数字被表示为加法份额的集合。对于选择数字,可以是使用乘法份额和加法份额的表示,但这不是必需的。然而,对于选择数字,使用多个乘法份额可能需要加法电路,优选避免这种情况。条件赋值单元被配置为:

[0086] 针对第一变量的每个乘法份额,

[0087] 计算第一选择数字R与表示第一变量的乘法份额的加法份额X_i的第一卷积$R*X_i$,

[0088] 计算第二选择数字R′与表示第二变量的对应的乘法份额的加法份额Y_i的第二卷积$R'*Y_i$,

[0089] 将第一卷积和第二卷积的结果相加$Z_i=R*X_i+R'*Y_i$,并且将得到的多个加法份额Z_i存储在存储设备中,作为以赋值结果z的乘法份额的加法份额形式的表示。

[0090] 使用哪个选择数字取决于条件d。例如,如果存储设备120存储两个选择数字R_1和R_2,则单元144可以设置:如果d为真(例如,d=1),则$R=R_1$,$R'=R_2$,并且如果d为假(例如,d=0),则$R=R_2$,$R'=R_1$。在实施例中,选择数字是0或1的加法表示。例如,选择数字中的一个选择数字可以是0或1的加法表示,而另一个选择数字则表示0和1的另一个。

[0091] 图3b示意性地示出了条件赋值单元144的实施例的示例。图3b示出了两个数字,这两个数字被表示为多个乘法份额30和40。每个乘法份额由加法份额的集合表示。在图3b中,附图标记31-33、41-43中的每个表示加法份额的集合。图3b示出了两个选择数字,这两个选择数字由加法份额集合51和52表示。第一卷积单元351被示为具有选择数字51的卷积集合31。第二卷积单元352被示为具有选择数字52的卷积集合41。加法单元353将单元351和352的结果相加。该结果是赋值结果的一个元素。在该迭代之后,将集合32与51,集合42与52进行卷积并将结果进行相加。

[0092] 不需要在每次迭代时都使用相同的选择数字,而是也可以使用不同的数字。特别地,可以在每次迭代中使用表示相同值(例如,值0)的不同表示,但是每次或者某些次被表示得像利用不同表示一样。也不需要按照本文所指示的顺序来执行计算。乘法和赋值都是高度并行的。在实施例中,可以通过例如根据随机变量排列计算的顺序(作为另外的混淆步骤)来利用这一点。随机化有助于防止平均攻击。注意,选择数字可以在运算之前,赋值之前或甚至在每次迭代或某些次迭代之前随机生成。

[0093] 通过加上伪运算,可以进一步混淆的乘法和赋值。例如,合适的伪运算是进行与数字1的表示的乘法。例如,一种合适的伪运算是将数字0的加法表示加到一个或多个加法份额集合。包括一个或多个伪运算将使迭代之间和/或运算的不同调用之间的时序随机化。在根据随机过程(例如,对随机数生成器的调用)来插入伪运算的情况下特别如此。随机时序阻碍平均攻击,此外,如果伪运算类似于上述常规运算,则攻击者很难进行重新同步。

[0094] 注意,例如使用乘法单元142的混淆的乘法和例如使用赋值单元144的混淆的赋值能够对相同表示中的数字进行运算。不需要重新编码以从单元142移动到单元144,反之亦

然。这意味着能够轻松地将这些运算串在一起。例如,在实施例中,诸如加密运算之类的运算包括条件赋值,该条件赋值之后是与条件赋值的结果的乘法。例如,在实施例中,诸如加密运算之类的运算包括条件赋值,该条件赋值之后是与赋值的结果的乘法。在实施例中,幂运算包括重复的混淆的乘法。

[0095] 也能够通过使用混淆的赋值来获得更好的混淆的幂运算。例如,能够通过使用乘法和条件赋值来有效地执行幂运算。例如,幂运算可以包括重复的乘法,其中,该乘法取决于指数中的位。条件赋值是根据指数中的位来运行的,该条件赋值之后是乘法。例如,为了有效地执行幂运算,可以使用所谓的蒙哥马利阶梯。

[0096] 可以根据以下内容来实施蒙哥马利阶梯:

[0097] s←1

[0098] t←h

[0099] for i=λ-1至0,进行

[0100] $u \leftarrow (1-d_i)s+d_i t \bmod N$ (I)

[0101] $s \leftarrow su \bmod N$ (II)

[0102] $t \leftarrow tu \bmod N$ (II)

[0103] End for

[0104] 其中,h表示幂运算的底数,并且位d_i表示指数的位,如在实施例中那样,条件赋值(I)和乘法(II)被混淆。注意,使用混淆的赋值并在每次迭代中包括相等数量的乘法的幂运算算法对边信道攻击的抵抗力有所增强,因为从赋值和乘法中都很难确定指数。上面的蒙哥马利阶梯具有该属性,因为赋值和乘法都受到份额保护。此外,乘法与秘密指数位无关。

[0105] 如果乘法份额的数量和/或加法份额的数量很大,那么幂运算(其已经是很昂贵的运算)的确能够变得非常昂贵。能够通过利用较小的受保护的指数执行受保护的幂运算并利用未受保护的指数执行常规幂运算来避免这种情况。能够选择受保护的指数和未受保护的指数,使得它们的组合结果等于具有预期指数(在RSA的情况下为具有秘密指数,例如,秘密密钥)的幂运算。例如,在实施例中,幂运算单元150可以被配置为通过获得第一指数和第二指数来执行幂运算,第一指数具有比第二指数更少的位,由第一指数和第二指数进行的所述后续的幂运算等于由该指数进行的幂运算,其中,具有第一指数的幂运算包括混淆的乘法和/或条件赋值。例如,可以在密钥生成期间选择第一指数和第二指数,并且将第一指数和第二指数存储在设备100中(例如存储在存储设备120中)。

[0106] 在设备100的各种实施例中,可以从各种替代方案中选择通信接口。例如,该接口可以是去往局域网或广域网(例如,互联网)的网络接口、去往内部或外部数据存储设备的存储设备接口、键盘、应用程序接口(API)等。

[0107] 设备100可以具有用户接口,该用户接口可以包括众所周知的元件,例如,一个或多个按钮、键盘、显示器、触摸屏等。用户接口可以被布置用于容纳用于执行计算(例如,加密运算,例如,解密运算或签名运算)的用户交互。

[0108] 存储设备120可以被实施为电子存储器(例如,闪速存储器)或磁性存储器(例如,硬盘)等。存储设备120可以包括一起构成存储设备110的多个离散存储器。存储设备120也可以是临时存储器(例如,RAM)。在临时存储设备120的情况下,存储设备120包含一些在使用之前获得数据(例如通过在任选的网络连接(未示出)上获得数据)的单元。

[0109] 典型地,设备100包括微处理器(未单独示出),该微处理器运行在设备100处存储的合适的软件;例如,该软件可能已经被下载并且/或者被存储在对应的存储器(例如,诸如RAM之类的易失性存储器或诸如Flash之类的非易失性存储器(未单独示出))中。替代地,设备100可以全部或部分地以可编程逻辑单元来实施,例如被实施为现场可编程门阵列(FPGA)。设备100可以全部或部分地被实施为所谓的专用集成电路(ASIC),即,针对其特定用途而定制的集成电路(IC)。例如,可以例如使用诸如Verilog、VHDL等的硬件描述语言在CMOS中实施电路。

[0110] 在实施例中,计算设备包括以下中的一个或多个或全部:卷积电路、加法电路、乘法电路、条件赋值电路、幂运算电路、通信接口电路。该电路实施本文描述的对应单元。这些电路可以是处理器电路和存储设备电路,该处理器电路运行在存储设备电路中以电子方式表示的指令。

[0111] 处理器电路可以以分布式方式来实施,例如被实施为多个子处理器电路。存储设备可以被分布在多个分布式子存储设备上。存储器的部分或全部可以是电子存储器、磁性存储器等。例如,存储设备可以具有易失性部分和非易失性部分。存储设备中的部分存储设备可能是只读的。

[0112] 因此,公开了用于利用秘密指数来执行模幂运算的设备和方法,该秘密指数对于通过边信道泄漏指数具有增强的抵抗力。实施例组合了例如蒙哥马利阶梯、乘法份额和加法份额。应用包括RSA算法和Z_p^*中的Diffie-Hellman算法。在RSA应用中,秘密指数通常很大,例如为4096位中的2048位。优化被示为减小所使用的指数的大小,这大大加快了计算速度。在下文中,以更为数学的语言讨论了这些实施期和其他实施例。

[0113] 1术语和符号

[0114] 如果数字$x = \sum_{i=0}^{n-1} x_i$,则n个数字x_0、x_1、K、x_{n-1}是x的加法n份额表示。符号:x=A(x_0,x_1,K,x_{n-1})。如果数字$x = \prod_{i=0}^{m-1} x_i$,则m个数字$x_0$、$x_1$、K、$x_{m-1}$是x的乘法m份额表示。符号:

[0115] x=M(x_0,x_1,K,x_{m-1})

[0116] 如果x=A(x_0,K,x_{n-1})并且y=A(y_0,y_1,K,y_{n-1}),则每份额加法给出加和x+y的加法n份额表示:A(x_0+y_0,x_1+y_1,K,x_{n-1}+y_{n-1})=x+y。如果x=M(x_0,K,x_{m-1})并且y=M(y_0,y_1,K,y_{m-1}),则每份额乘法给出乘积xy的乘法m份额表示:M(x_0y_0,x_1y_1,K,$x_{n-1}y_{n-1}$)=xy。

[0117] 令X=(x_0,K,x_{n-1})和Y=(y_0,K,y_{n-1})为两个数字n元组,则X与Y的卷积(被表示为X*Y)被定位为n元组Z=(z_0,K,z_{n-1}),其中,$z_i = \sum_{j=0}^{n-1} x_j\, y_{(i-j) \bmod n}$,其中,0≤i≤n-1。针对n=3的示例:如果X=(x_0,x_1,x_2)并且Y=(y_0,y_1,y_2),则X*Y=(x_0y_0+x_1y_2+x_2y_1,x_0y_1+x_1y_0+x_2y_2,x_0y_2+x_1y_1+x_2y_0)。

[0118] 它适用于满足A(X*Y)=A(X)A(Y)的所有X和Y。换句话说:两个加法份额表示的卷积会得到乘积的加法份额表示。令x=M(x_0,K,x_{m-1})并且y=M(y_0,K,y_{m-1})。

[0119] 如果存在对乘法份额表示具有类似作用的线性运算,那将是有利的:两个乘法份额表示的这样的假定运算将得到加和的乘法份额表示,即,针对所有X和Y,被表示为$X \circ Y$的操作将满足$M(X \circ Y) = M(X) + M(Y)$。事实证明:能够通过数学论证来证明这样的假定运算通常无法存在。尽管如此,即使不可能通过采用线性组合来根据x和y的乘法份额

来创建任意加权和(1-α)x+αy的乘法份额,也能够在α=0或α=1时有:$z_i=(1-\alpha)x_i+\alpha y_i$(其中,0≤i≤m-1)满足:

[0120] 当α∈{0,1}时,$M(z_0,K,z_{m-1})=(1-\alpha)M(x_0,K,x_{m-1})+\alpha M(y_0,K,y_{m-1})$

[0121] 注意,x和y的每个份额都乘以零或一。例如,能够通过以下方式对此进行伪装。

[0122] 1.构造表示0的加法份额集合$0_0,0_1,K,0_{K-1}$的数字(K≥1),即,$0_i=(o_{i,0},K,o_{i,n-1})$,其中,$A(o_{i,0},K,o_{i,n-1})=0$,其中,0≤i≤K-1

[0123] 2.构造表示1的加法份额集合J_0,J_1,K,J_{K-1}的数字(K≥1),即,$J_i=(j_{i,0},K,j_{i,n-1})$,其中,$A(j_{i,0},K,j_{i,n-1})=1$,其中,0≤i≤K-1

[0124] 3.通过加法份额集合$X_i=(x_{i,0},K,x_{i,n-1})$来表示x的每个乘法份额$x_i$,使得

[0125] $x_i=A(x_{i,0},K,x_{i,n-1})$

[0126] 4.通过加法份额集合$Y_i=(y_{i,0},K,y_{i,n-1})$来表示y的每个乘法份额$y_i$,使得

[0127] $y_i=A(y_{i,0},K,y_{i,n-1})$

[0128] 5.如果α=0,则将z的乘法份额z_i的加法份额集合Z_i计算为:

[0129] $$Z_i = J_{k_i} * X_i + O_{l_i} * Y_i$$

[0130] 其中,k_i和l_i是任意选择的满足$0\leq k_i,l_i\leq K-1$的整数。

[0131] 如果α=1,则将Z_i计算为:

[0132] $$Z_i = O_{k_i} * X_i + J_{l_i} * Y_i$$

[0133] 能够用一个公式来概括这两种情况:

[0134] $$Z_i = R(1-d_i)_{k_i} * X_i + R(d_i)_{l_i} * Y_i$$

[0135] 其中,R(0)代表0并且R(1)代表J。

[0136] 2使用蒙哥马利阶梯进行幂运算

[0137] 在RSA和Diffie-Hellman协议中,通常需要执行以公共模数N为模的“公共”数字h和“秘密”指数的模幂运算:

[0138] $s=h^d \bmod N$

[0139] 例如,在RSA签名中,我们可能有:h是填充的消息摘要,d是私钥,并且s是得到的签名。令λ表示N(和d)的位长并将d的二元扩展编写为$d=\sum_{i=0}^{\lambda-1} d_i 2^i$,其中,$d_i\in\{0,1\}$。一种用于计算s的简单算法如下:

[0140] 简单直接的幂运算

[0141] s←1

[0142] for i=λ-1至0,进行

[0143] $s\leftarrow s^2 \bmod N$

[0144] if $d_i=1$,则

[0145] $s\leftarrow sh \bmod N$

[0146] end if

[0147] end for

[0148] 该算法对边信道敏感,因为平方和乘法的模式揭示了指数d。当针对给定的d展开if语句时,也是如此。

[0149] 蒙哥马利阶梯以如下方式计算s:

[0150] 蒙哥马利阶梯

[0151] s←1

[0152] t←h

[0153] for i＝λ-1至0,进行

[0154] if d_i＝0,则

[0155] a←s^2mod N

[0156] b←st mod N

[0157] else

[0158] a←st mod N

[0159] b←t^2mod N

[0160] end if

[0161] s←a

[0162] t←b

[0163] end for

[0164] 蒙哥马利阶梯提供了一些防止边信道攻击的保护措施,因为在每个步骤中均会出现平方和乘法。然而,任何允许攻击者在每个步骤中观察到s或t是否被平方的边信道仍会泄露密钥。

[0165] 利用蒙哥马利阶梯的以下变体来获得相同的结果:

[0166] 蒙哥马利阶梯,变体

[0167] s←1

[0168] t←h

[0169] for i＝λ-1至0,进行

[0170] u←$(1-d_i)s+d_i t$ mod N

[0171] s←su mod N

[0172] t←tu mod N

[0173] end for

[0174] 蒙哥马利阶梯的该变体使用乘法和加法,但不使用平方。允许攻击者在每个步骤中观察到s是乘以零还是乘以一或者等效地观察到是u＝s还是u＝t的边信道会泄露密钥。

[0175] 为了使攻击者更难以从边信道攻击中获得密钥,我们可以使用第一章节中的混淆技术来使攻击者更难看到是u＝s还是u＝t或者某数字是否乘以零还是乘以一。

[0176] 实施者选择数字m≥1(表示蒙哥马利阶梯变量的乘法份额的数量)和n≥2(乘法份额的加法份额的数量)。因此,蒙哥马利阶梯变量由mn个份额表示。优选m和n两者都很大。

[0177] 实施者选择数字集合$\{A_{\mu,v}\}_{\mu=0,v=0}^{m-1,n-1}$和$\{B_{\mu,v}\}_{\mu=0,v=0}^{m-1,n-1}$,使得

[0178] $$\prod_{\mu=0}^{m-1}\left(\sum_{v=0}^{n-1}A_{\mu,v}\right) mod\, N = 1$$

[0179] 并且

[0180] $$\prod_{\mu=0}^{m-1}\left(\sum_{v=0}^{n-1}B_{\mu,v}\right)\bmod N=h$$

[0181] 来初始化阶梯。梯形利用mn个数字$S_{\mu,v}$、$T_{\mu,v}$和$U_{\mu,nu}$来进行工作，n个份额($S_{\mu,0}$,K,$S_{\mu,n-1}$)被表示为S_{μ}，并且对于T个份额和U个份额具有类似情况。数字$k_{i,\mu}$和$l_{i,\mu}$(其中，0≤i≤λ-1并且0≤μ≤m-1)可以是从[0,K)中任意选择的整数。

[0182] 具有乘法份额和加法份额的受保护的蒙哥马利阶梯

[0183] forμ＝0至m-1，进行

[0184] for v＝0至n-1，进行

[0185] $S_{\mu,v}\leftarrow A_{\mu,v}$

[0186] $T_{\mu,v}\leftarrow B_{\mu,v}$

[0187] end for

[0188] end for

[0189] for i＝λ-1至0，进行

[0190] forμ＝0至m-1，进行

[0191] $$U_{\mu}\leftarrow R(1-d_i)_{k_{i,\mu}}*S_{\mu}+R(d_i)_{l_{i,\mu}}*T_{\mu}\bmod N$$

[0192] end for

[0193] forμ＝0至m-1，进行

[0194] $S_{\mu}\leftarrow S_{\mu}*U_{\mu}\bmod N$

[0195] $T_{\mu}\leftarrow T_{\mu}*U_{\mu}\bmod N$

[0196] end for

[0197] end for

[0198] $$s\leftarrow\prod_{\mu=0}^{m-1}\left(\sum_{v=0}^{n-1}S_{\mu,v}\right)\bmod N$$

[0199] 在实施例中，可以对保留基础值不变的份额执行伪变换，例如，他可以选择应用n个加法份额的随机排列或m个(由加法份额集合表示的)乘法份额的随机排列。优选地，在运行时间期间而不是在编译时间期间选择这些变换。例如，可以根据输入h来选择这些变换。例如，该输入h可以用于为随机数生成器设定种子。

[0200] 在一些应用中，例如签名主要是针对幂运算中的秘密的指数，但是算法的输入和输出都不是机密的。在这种情况下，由于中间结果仅依赖于密钥的部分，因此中间结果可能仍然很敏感，因此可能会允许强力得到密钥的部分。

[0201] 为了将输入变量h掩盖为多个份额，可以首先计算乘法份额，然后将乘法份额中的每个表示为加法份额。例如，在实施例中，变量h用作随机数生成器的种子，该种子继而生成除了一个以外的所有乘法份额或加法份额，在此之后计算最终份额。替代地，可以在编译时选择随机的乘法函数的集合f_i或加法函数的集合g_i。这些函数满足$\Pi f_i(x)=x$和$\Sigma g_i(x)=x$，在这两种情况下均以公共模数为模。因此，这些函数的乘积或加和与输入无关，至少以模数为模。函数的数量等于乘法份额的数量或每个乘法份额的加法份额的数量。如果只有一个乘法份额，则可以省去乘法函数。为了将任何变量(例如，变量h)表示为份额，可以首先计算$(f_0(h),f_1(h),\ldots,f_{m-1}(h))$。为了表示乘法份额s，可以计算$(g_0(s),\ldots,g_{n-1}(s))$。

[0202] 也可以部分使用该方法。例如,可以在运行时间动态地计算乘法份额,例如使用随机数生成器,使用逆向算法计算最终份额来动态地计算乘法份额。然后,使用加法函数的集合将乘法份额中的每个映射到多个加法份额的集合。注意,针对加法份额的集合中的一些或每个,加法函数可能不同。例如,在编译时,可以选择除了一个以外的所有加法函数作为随机多项式,在此之后能够计算最终多项式。

[0203] 例如,为了生成份额,可以进行以下操作。假定我们希望在输入I上表示f(I),然后令$r_i(I)$为在输入I上提供随机份额的函数(其中,i=1..n-1)并且令$r_n(I)$被定义为$f(I)-r_1(I)...-r_{n-1}(I)$。现在,$x_i=r_i(I)$给出f(I)的加法n份额。类似地,令$R_i(I)$为在输入I上提供随机份额的函数(其中,i=1..n-1)并且令$R_n(I)$被定义$f(I)/(R_0(I)*...*R_{n-1}(I))$,然后,$X_i=R_i(I)$给出f(I)的n乘法份额。这些函数r和R可以是伪随机函数。

[0204] 具有mn个乘法份额和加法份额的蒙哥马利阶梯花费大致mn次,这与没有份额的蒙哥马利阶梯的运算次数一样。这可能是针对额外的边信道安全性所付出的高昂代价。以下方法可以在不影响安全性的情况下显著减少运算次数。考虑以下内容:

[0205] 1.能够分解N的攻击者无需付出太多额外努力即可找到私钥d。当前已知的分解N的最好方法使用数域筛选来渐近地运算:

[0206] $$exp\left(\left(\sqrt[3]{\frac{64}{9}}+o(1)\right)(\ln(N))^{1/3}(\ln(\ln(N)))^{2/3}\right)$$

[0207] 针对$N\approx 2^{2048}$,这转变为大致112位的安全性;针对$N\approx 2^{4096}$,这转变为大约150位的安全性。

[0208] 2.从s中找到d,h和N等于解决离散对数问题,针对该问题,最佳求解方法渐近地采用$O(\sqrt{N})$运算。这比分解N昂贵得多。

[0209] 3.如果知道离散对数问题中的指数很小(例如位),则强力攻击进行$O(2^t)$次运算。只要t大于RSA的安全级别,对N的因式分解就是对d最好攻击。

[0210] 秘密指数d可以分为秘密部分和公开部分。例如,选择一个随机t位数字d_s,d为"秘密"部分,并且计算$d_p=d_s^{-1}d \bmod \phi(N)$,d为λ位"公开"部分。这使用了模数N的欧拉函数φ(N),欧拉函数φ(N)例如是从N的因式分解获得的。这里,φ是欧拉函数。

[0211] 幂运算分为两个步骤:

[0212] $$s=h^d \bmod N=h^{d_s d_p} \bmod N=(h^{d_s} \bmod N)^{d_p} \bmod N$$

[0213] 使用上面的受保护的蒙哥马利阶梯,使用mn个乘法份额和加法份额来完成具有指数d_s的内部模幂运算,使用安全性较低的蒙哥马利阶梯来完成具有指数d_p的外部模幂运算。指数d_p很可能会通过边信道泄漏,但这并不重要,只要d_s不泄漏即可。注意,因为

[0214] $$(h^{d_s} \bmod N)^{d_p} \bmod N\ (h^{d_p} \bmod N)^{d_s} \bmod N$$

[0215] 能够互换"内部"模幂运算与"外部"模幂运算。

[0216] 与针对上面的受保护方法的λmn相比,针对该方法的总工作量为λ+tmn:当mn>1时会有显著改善,因为t比λ要小得多。例如,在实施例中,t小于λ。例如,10t小于λ。例如,t可以小于256,而λ可以大于2048。

[0217] 能够在完成密钥生成的同时将秘密指数分成秘密部分和公开部分。指数的这些部分能够被存储在存储设备(例如,计算设备的存储器)中。

[0218] 图4示意性地示出了计算方法400的实施例的示例。计算方法400被布置用于对乘法的混淆的运行,方法400包括:

[0219] 存储410在算术运算的运行中使用的多个变量,多个变量中的变量(x;y;2)被表示为多个乘法份额($X=(x_0,x_1,\ldots,x_{m-1})$;$Y=(y_0,y_1,\ldots,y_{m-1})$;20),所述乘法份额在存储设备中被表示为多个加法份额($X_i=(x_{i,0},x_{i,1},\ldots,x_{i,n-1})$;$Y_i=(y_{i,0},y_{i,1},\ldots,y_{i,n-1})$;210,220),并且

[0220] 将存储设备中的第一变量与存储设备中的第二变量相乘(420)以获得乘法结果(z=xy)。

[0221] 所述相乘包括:

[0222] 针对所述第一变量的每个乘法份额,

[0223] 计算430表示第一变量的所述乘法份额的加法份额(X_i)与表示第二变量的对应的乘法份额的加法份额(Y_i)的卷积($Z_i=X_i*Y_i$),

[0224] 将得到的多个加法份额(Z_i)存储440在存储设备中,作为以乘法结果(z)的乘法份额的加法份额形式的表示。因此,根据需要而频繁地重复(例如迭代)运算430和440,例如在存储设备中的数字的表示中频繁地重复(例如迭代)运算430和440。

[0225] 本领域技术人员能够容易地想到运行该方法的许多不同方式。例如,能够改变步骤的顺序,或者可以并行运行一些步骤。此外,在步骤之间可以插入其他方法步骤。插入的步骤可以表示对本文所述的方法的细化,也可以与该方法无关。例如,运算430和440可以至少部分地并行运行。此外,在开始下一步骤之前,给定步骤可能尚未完全完成。

[0226] 可以使用软件来运行根据本发明的方法,该软件包括用于使处理器系统执行方法400的指令。软件可能只包括系统的特定子实体所采取的那些步骤。该软件可以被存储在合适的存储介质(例如,硬盘、软盘、存储器、光盘等)中。可以将该软件作为信号而沿着有线、无线或使用数据网络(例如,互联网)进行发送。可以使该软件可用于从服务器上下载和/或在服务器上远程使用。可以使用位流来运行根据本发明的方法,该位流被布置为配置可编程逻辑单元(例如,现场可编程门阵列(FPGA))以执行该方法。

[0227] 应当理解,本发明还扩展到适于使本发明付诸实践的计算机程序,特别是在载体上或载体中的计算机程序。该程序可以是源代码、目标代码、代码中间源和目标代码的形式(例如部分编译的形式),或者是适合用于实施根据本发明的方法的任何其他形式。与计算机程序产品有关的实施例包括与所阐述的方法中的至少一种的处理步骤中的每个相对应的计算机可执行指令。这些指令可以被细分成子例程并且/或者被存储在一个或多个可以静态或动态链接的文件中。与计算机程序产品有关的另一实施例包括与所阐述的系统和/或产品中的至少一个的单元中的每个相对应的计算机可执行指令。

[0228] 图5a示出了根据实施例的具有包括计算机程序1020的可写部分1010的计算机可读介质1000,计算机程序1020包括用于使处理器系统执行计算方法的指令。计算机程序1020可以作为物理标记或借助于对计算机可读介质1000的磁化而被实施在计算机可读介质1000上。然而,也可以想到任何其他合适的实施例。此外,应当理解,虽然计算机可读介质1000在这里被示为光盘,但是计算机可读介质1000也可以是任何合适的计算机可读介质

(例如,硬盘、固态存储器、闪速存储器等),并且可以是不可记录的或可记录的。计算机程序1020包括用于使处理器系统执行所述计算方法的指令。

[0229] 图5b以示意性表示示出了根据计算设备的实施例的处理器系统1140。该处理器系统包括一个或多个集成电路1110。在图5b中示意性地示出了一个或多个集成电路1110的架构。电路1110包括处理单元1120(例如,CPU),处理单元1120用于运行计算机程序部件以运行根据实施例的方法并且/或者实施其模块或单元。电路1110包括用于存储编程代码、数据等的存储器1122。存储器1122的部分可以是只读的。电路1110可以包括通信元件1126(例如,天线、连接器或这两者)等。电路1110可以包括专用集成电路1124,专用集成电路1124用于执行该方法中定义的部分或全部处理。处理器1120、存储器1122、专用IC 1124和通信元件1126可以经由互连1130(例如,总线)彼此连接。处理器系统1110可以被布置用于分别使用天线和/或连接器进行接触和/或无接触通信。

[0230] 例如,在实施例中,计算设备可以包括处理器电路和存储器电路,处理器被布置为运行在存储器电路中存储的软件。例如,处理器电路可以是英特尔酷睿i7处理器,ARM Cortex-R8等。在实施例中,处理器电路可以是ARM Cortex M0。存储器电路可以是ROM电路,或者可以是非易失性存储器(例如,闪速存储器)。存储器电路可以是易失性存储设备(例如,SRAM存储器)。在后一种情况下,设备可以包括被布置用于提供软件的非易失性软件接口(例如,硬盘驱动器、网络接口等)。

[0231] 应当注意,以上提及的实施例说明而非限制本发明,并且本领域技术人员将能够设计许多替代实施例。

[0232] 在权利要求中,置于括号内的任何附图标记均不应被解读为对权利要求的限制。动词“包括”及其词性变化的使用不排除权利要求中记载的那些以外的其他元件或步骤的存在。元件前的词语“一”或“一个”不排除多个这样的元件的存在。本发明可以借助于包括若干不同元件的硬件,以及借助于被适当编程的计算机来实施。在列举了若干单元的装置型权利要求中,这些单元中的若干可以由同一项硬件来实施。某些措施被记载在互不相同的从属权利要求中的事实并不指示不能有利地使用这些措施的组合。

[0233] 在权利要求中,括号内的附图标记指代示例性实施例的附图中的附图标记或实施例的公式,由此提高了权利要求的可理解性。这些附图标记不应被解释为限制权利要求。

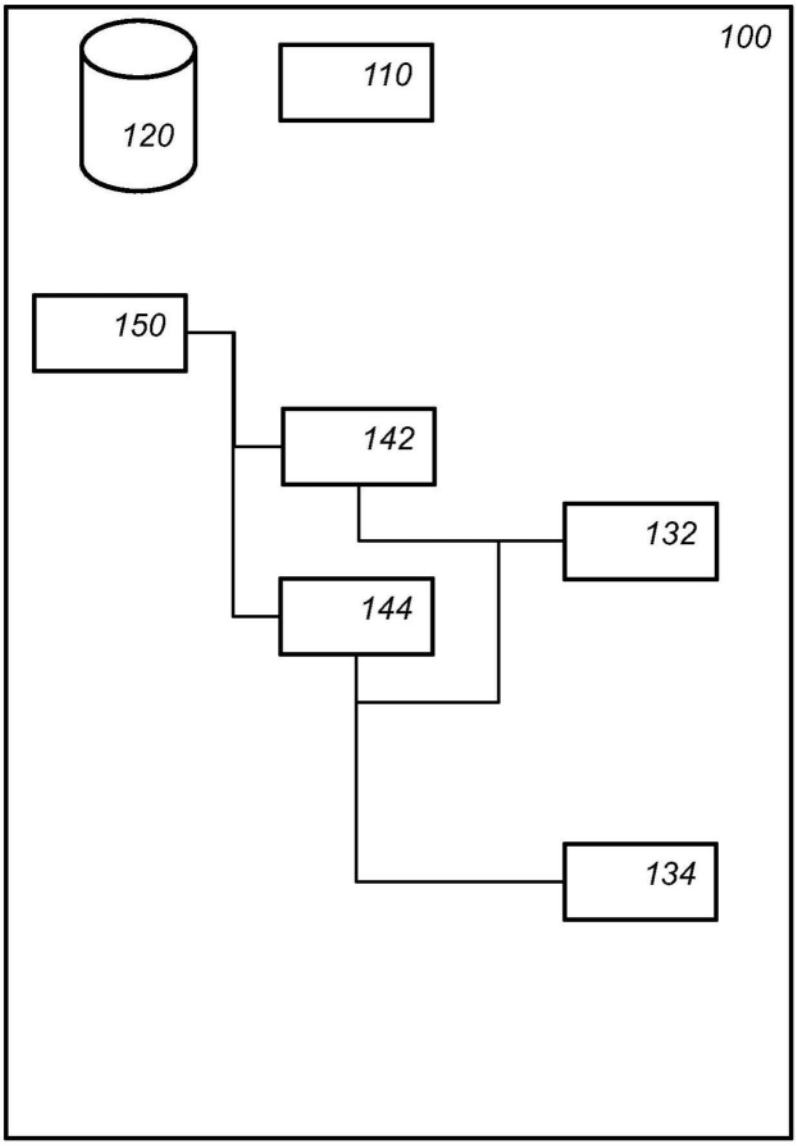


图1

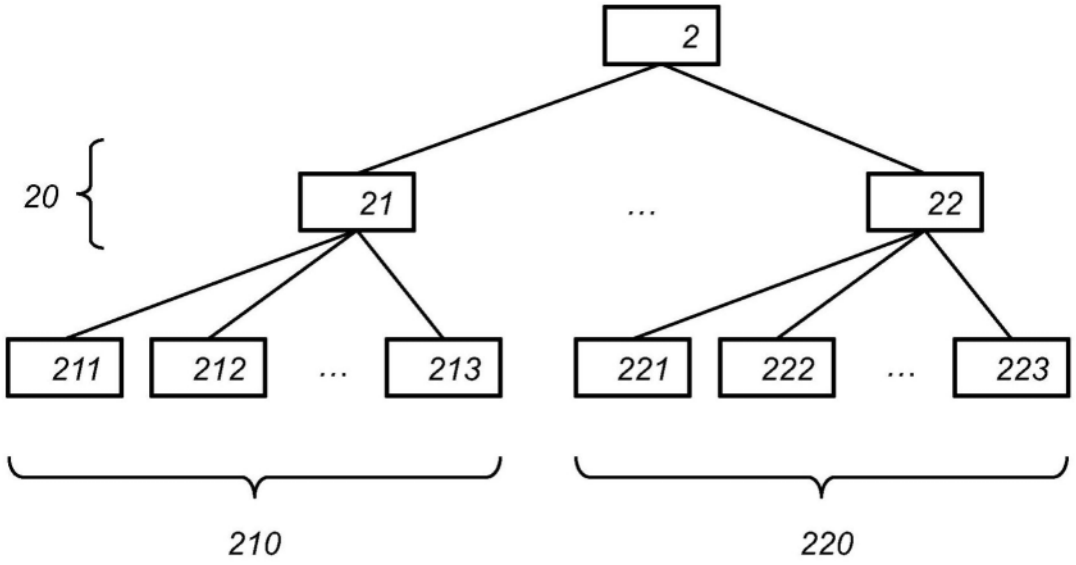

2
20
21
...
22
211
212
...
213
221
222
...
223
210
220

图2

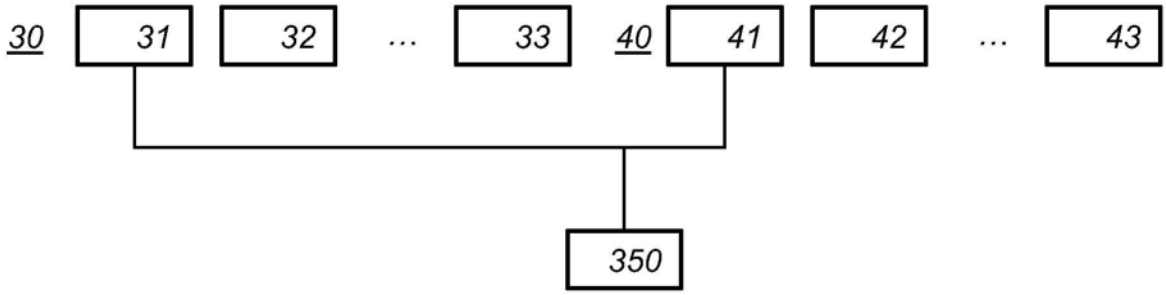

30
31
32
...
33
40
41
42
...
43
350

图3a

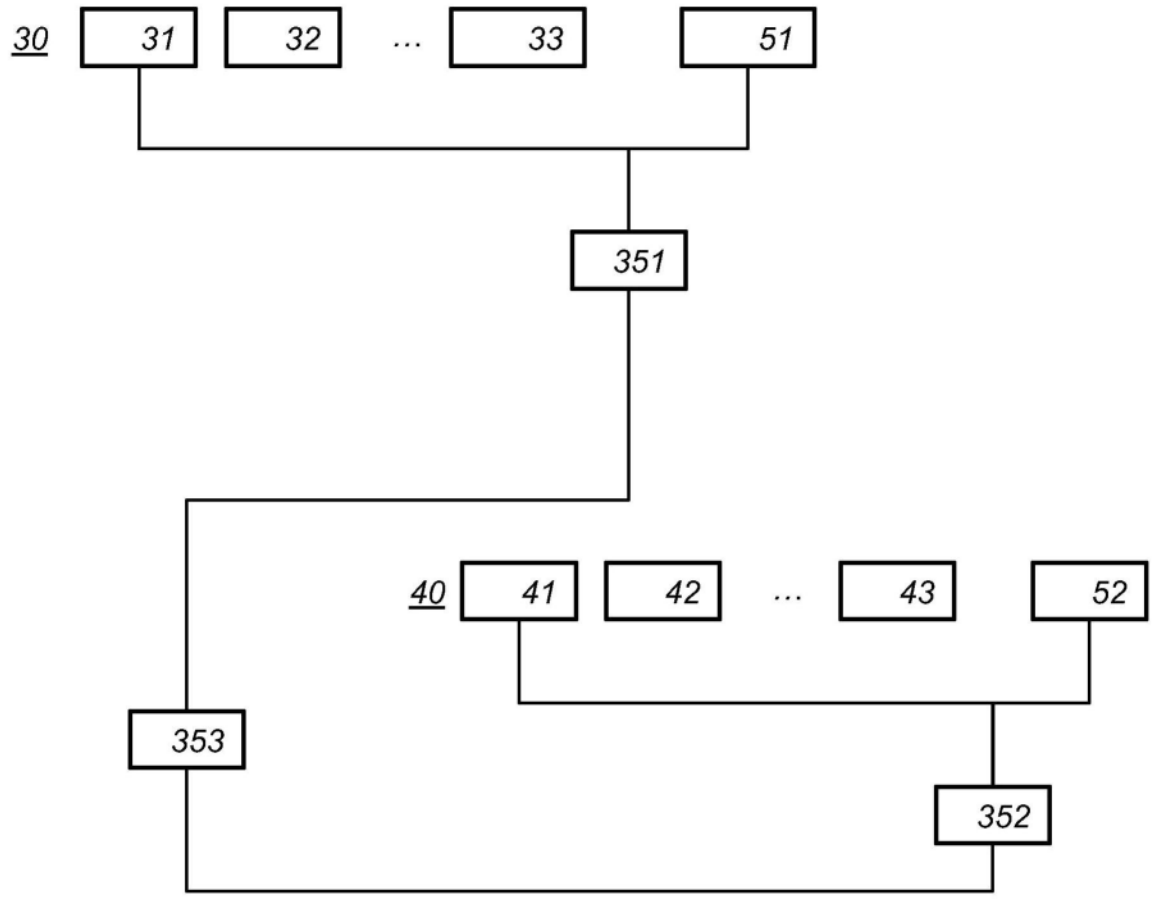


图3b

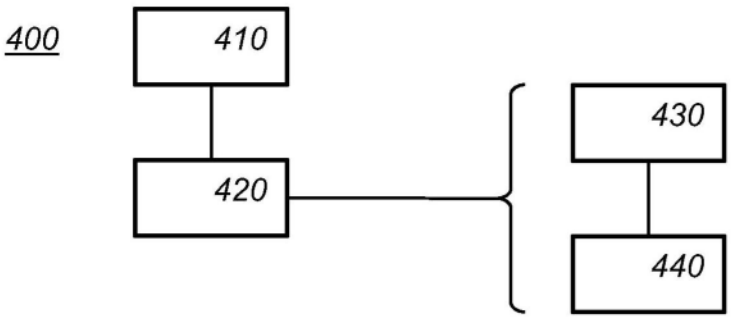


图4

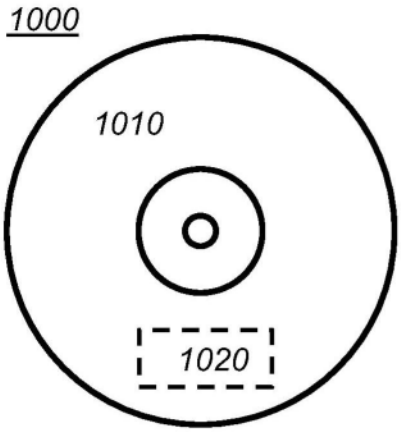


图5a

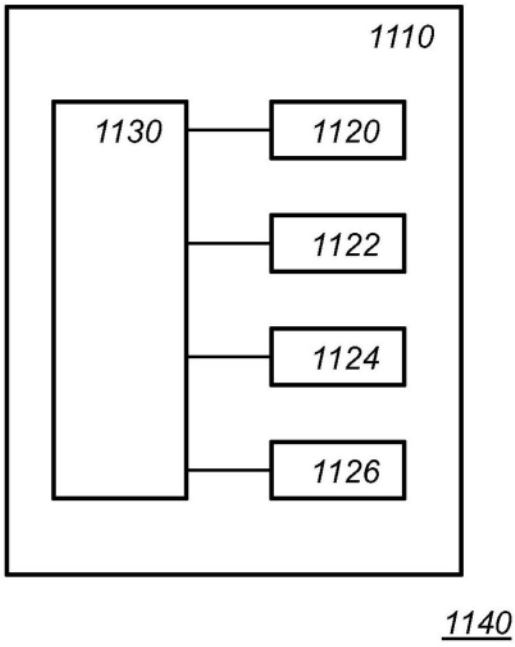


图5b