(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 106031079 B
(45)授权公告日 2019.10.11

(21)申请号 201480076008.4

(22)申请日 2014.12.17

(65)同一申请的已公布的文献号
申请公布号 CN 106031079 A

(43)申请公布日 2016.10.12

(30)优先权数据
13198943.6 2013.12.20 EP

(85)PCT国际申请进入国家阶段日
2016.08.19

(86)PCT国际申请的申请数据
PCT/EP2014/078107 2014.12.17

(87)PCT国际申请的公布数据
WO2015/091583 EN 2015.06.25

(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 P.M.H.M.A.戈里斯森
L.M.G.M.托胡伊泽恩

(74)专利代理机构 中国专利代理(香港)有限公司 72001
代理人 张同庆 景军平

(51)Int.Cl.
H04L 9/00(2006.01)

(56)对比文件
WO 2006058561 A1,2006.06.08,
CN 102460404 A,2012.05.16,
CN 101242275 A,2008.08.13,
CN 1926800 A,2007.03.07,
CN 1890914 A,2007.01.03,
CN 101578813 A,2009.11.11,
CN 101093627 A,2007.12.26,
CN 101969374 A,2011.02.09,
WO 2006012638 A2,2006.02.02,
Claude Castelluccia, Einar Mykletun, Gene Tsudik.efficient aggregation of encrypted data in wireless sensor networks.《The second annual international conference on Mobile and Ubiquitous System:Networking and Services》.2005,

审查员 陈丽锋

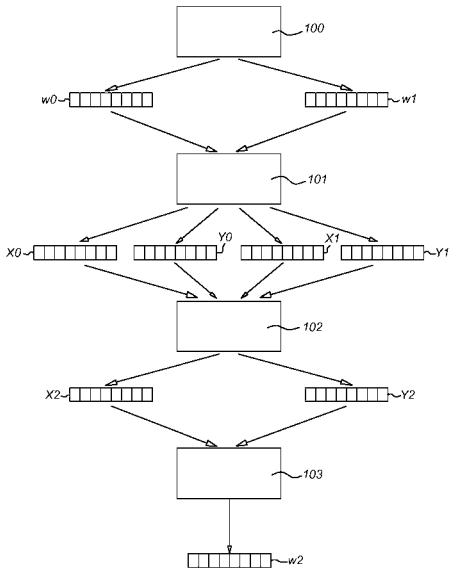
权利要求书2页 说明书10页 附图2页

(54)发明名称

加密算法中的运算符提升

(57)摘要

公开了一种用于使用数据的混淆表示在数据上执行运算的系统。获取构件被配置成获取第一数据值的第一混淆表示并且获取第二数据值的第二混淆表示。确定构件102被配置成通过在第一数据值的混淆表示和第二数据值的混淆表示上执行对应运算来确定第三数据值的混淆表示。混淆构件101可以配置成基于第一数据值生成第一混淆表示和并且基于第二数据值生成第二混淆表示。去混淆构件103可以配置成对第三数据值的混淆表示去混淆,以便使用方程组来获取第三数据值。



CN 106031079 B

1.一种用于使用数据的混淆表示在数据之上执行运算的系统,包括:

获取构件,其配置成获取第一数据值w_0的第一混淆表示(X_0,Y_0)并且获取第二数据值w_1的第二混淆表示(X_1,Y_1),其中以下关系成立:

$$X_0 = A_0(w_0) \oplus B_0(\sigma_0)$$
$$Y_0 = A_1(w_0) \oplus B_1(\sigma_0)$$
$$X_1 = A_0(w_1) \oplus B_0(\sigma_1)$$
$$Y_1 = A_1(w_1) \oplus B_1(\sigma_1)$$

其中

$\oplus$是运算符,

A_0和A_1是取决于数据值(w_0,w_1)的线性运算符,且B_0和B_1是取决于状态变量(σ_0,σ_1)的线性运算符,并且将(u,v)映射到$(A_0(u) \oplus B_0(v),A_1(u) \oplus B_1(v))$的运算符E关于u可逆,并且

σ_0和σ_1是向混淆表示提供冗余的状态变量;以及

确定构件(102),其配置成通过在第一数据值w_0的混淆表示(X_0,Y_0)和第二数据值w_1的混淆表示(X_1,Y_1)上执行以下运算来确定第三数据值w_2的混淆表示(X_2,Y_2):

$X_2=X_0 \oplus X_1$

$Y_2=Y_0 \oplus Y_1$,

其中$w_2=w_0 \otimes w_1$,其中$\otimes$为运算符。

2.权利要求1所述的系统,其中获取构件包括混淆构件(101),其配置成生成基于第一数据值w_0的第一混淆表示(X_0,Y_0)和基于第二数据值w_1的第二混淆表示(X_1,Y_1)。

3.权利要求1或2所述的系统,进一步包括:

去混淆构件(103),其配置成对第三数据值w_2的混淆表示(X_2,Y_2)去混淆,以便使用以下方程组获取第三数据值w_2:

$$X_2 = A_0(w_2) \oplus B_0(\sigma_2)$$
$$Y_2 = A_1(w_2) \oplus B_1(\sigma_2),$$

其中σ_2是用于向第三数据值w_2的混淆表示(X_2,Y_2)提供冗余的状态变量。

4.权利要求2所述的系统,其中混淆构件(101)包括状态生成器以用于随机地或伪随机地生成状态变量σ_0的值和/或状态变量σ_1的值,并且其中混淆构件(101)配置成基于第一数据值w_0和状态变量σ_0来生成第一混淆表示(X_0,Y_0)并且基于第二数据值w_1和状态变量σ_1来生成第二混淆表示(X_1,Y_1)。

5.权利要求2所述的系统,其中混淆构件(101)被配置成在查找表中查找第一混淆表示(X_0,Y_0)和第二混淆表示(X_1,Y_1),并且/或者去混淆构件(103)被配置成在查找表中查找第三数据值w_2。

6.权利要求3所述的系统,其中混淆构件(101)和去混淆构件(103)是第一设备的部分并且确定构件是第二不同设备的部分,其中第一设备进一步包括传送构件和接收构件,其中第二设备进一步包括传送构件和接收构件,其中第一设备的传送构件配置成向第二设备的接收构件传送第一混淆表示(X_0,Y_0)和第二混淆表示(X_1,Y_1),并且其中第二设备的传送

构件配置成向第一设备的接收构件传送混淆表示(X_2,Y_2)。

7.权利要求1所述的系统,其中确定构件(102)配置成以明文执行从X_0和X_1计算X_2以及从Y_0和Y_1计算Y_2中的至少一个。

8.权利要求1所述的系统,其中w_0、w_1、w_2、σ_0、σ_1、σ_2、X_0、X_1、X_2、Y_0、Y_1和Y_2是具有相同位数的值。

9.权利要求1所述的系统,其中运算符A_0、B_0、A_1和B_1中的至少一个是可逆运算符。

10.权利要求9所述的系统,其中运算符A_0、B_0、A_1和B_1的每一个是可逆运算符。

11.权利要求1所述的系统,其中运算符⊕是按位XOR运算并且运算符⊗是按位XOR运算符。

12.权利要求11所述的系统,其中按位XOR运算通过至少一个XOR机器指令来执行。

13.一种用于使用数据的混淆表示在数据上执行运算的方法,包括以下步骤:

获取(201)第一数据值w_0的第一混淆表示(X_0,Y_0)并且获取第二数据值w_1的第二混淆表示(X_1,Y_1),其中以下关系成立:

$$X_0 = A_0(w_0) \oplus B_0(\sigma_0)$$
$$Y_0 = A_1(w_0) \oplus B_1(\sigma_0)$$
$$X_1 = A_0(w_1) \oplus B_0(\sigma_1)$$
$$Y_1 = A_1(w_1) \oplus B_1(\sigma_1),$$

其中

⊕是运算符,

A_0和A_1是取决于数据值(w_0,w_1)的线性运算符,且B_0和B_1是取决于状态变量(σ_0,σ_1)的线性运算符,并且将(u,v)映射到(A_0(u) ⊕ B_0(v),A_1(u) ⊕ B_1(v))的运算符E关于u可逆,并且

σ_0和σ_1是用于向混淆表示提供冗余的状态变量;以及

通过在第一数据值w_0的混淆表示(X_0,Y_0)和第二数据值w_1的混淆表示(X_1,Y_1)上执行以下运算来确定(202)第三数据值w_2的混淆表示(X_2,Y_2):

$X_2=X_0 \oplus X_1$

$Y_2=Y_0 \oplus Y_1$,

其中$w_2=w_0 \otimes w_1$,其中⊗为运算符。

14.一种计算机可读存储介质,其上存储有计算机程序,其特征在于,该程序在被处理器执行时实现权利要求13所述的方法。

加密算法中的运算符提升

技术领域

[0001] 本发明涉及使用操作数的混淆(obfuscated)表示执行运算。

背景技术

[0002] 当今,大量数据经由网络、移动电话、蓝牙设备、银行自动取款机等转移。为了保护信息免受非期望的访问,非常频繁地使用加密。在密码学中,加密是以使得第三方不能读取它而仅授权方可以读取它的这种方式对消息进行编码的过程。在加密方案中,使用加密算法来对被称为明文的消息加密,从而将它转变成不可读的密文。这通常使用加密密钥完成,加密密钥指定消息如何被编码。可以看到密文的任何对手应当不能确定关于原始消息的任何事物。然而,授权方能够使用解密算法对密文解码,这通常需要对手不能访问的秘密的解密密钥。

[0003] 加密也可以应用于保护存储的数据,诸如在计算机和存储设备中的文件。

[0004] 在云计算中,执行网络之上的分布式计算,其通常涉及在实时网络之上连接的大量计算机。那些计算中所涉及的数据需要受保护,因为其存储在第三方可能容易访问的网络中。

[0005] 在Craig Gentry的"Computing Arbitrary Functions of Encrypted Data" Communications of the ACM, Vol. 53, No3, 第97-105页, 2010年3月中,公开了一种保持数据私有但允许执行运算的加密方案。然而,该加密方案在计算上是昂贵的。

[0006] Castelluccia C等人的"Efficient Aggregation Of Encrypted Data In Wireless Sensor Networks", Mobile and Ubiquitous Systems: Networking and Services, 2005. MOBIQUIT OUS 2005, 2005年7月17日, 第109-117页, XP010853989, ISBN: 978-0-7695-2375-0公开了一种加法同态流密文。

[0007] WO 2006/058561 A1公开了一种在用户身份模块(SIM)上实现的密码学功能。随机掩模用于针对要执行的密码学功能掩蔽输入数据。特别地,掩蔽功能有利地为群运算。

发明内容

[0008] 将有利的是具有一种允许使用数据值的加密表示来执行运算的系统。为了更好地解决这一关切,本发明的第一方面提供一种用于使用数据的混淆表示在数据之上执行运算的系统,包括:

[0009] 获取构件,其配置成获取第一数据值w_0的第一混淆表示(X_0,Y_0)并且获取第二数据值w_1的第二混淆表示(X_1,Y_1),其中以下关系成立:

[0010] $$X_0 = A_0(w_0) \oplus B_0(\sigma_0)$$

[0011] $$Y_0 = A_1(w_0) \oplus B_1(\sigma_0)$$

[0012] $$X_1 = A_0(w_1) \oplus B_0(\sigma_1)$$

[0013] $Y_1 = A_1(w_1) \oplus B_1(\sigma_1)$

[0014] 其中

[0015] ⊕是运算符，

[0016] A_0、B_0、A_1和B_1是线性运算符，并且将(u,v)映射到($A_0(u) \oplus B_0(v)$，$A_1(u) \oplus B_1(v)$)的运算符E关于u可逆，并且

[0017] σ_0和σ_1是用于向混淆表示提供冗余的状态变量；以及

[0018] 确定构件，其配置成通过在第一数据值w_0的混淆表示(X_0,Y_0)和第二数据值w_1的混淆表示(X_1,Y_1)上执行以下运算来确定第三数据值w_2的混淆表示(X_2,Y_2)：

[0019] $X_2=X_0 \oplus X_1$

[0020] $Y_2=Y_0 \oplus Y_1$，

[0021] 其中$w_2=w_0 \otimes w_1$，其中⊗为运算符。

[0022] 该系统具有以下优点：两个输入数据值w_0和w_1之间的运算⊗可以使用输入数据值w_0的混淆表示(X_0,Y_0)和输入数据值w_1的混淆表示(X_1,Y_1)来执行，而无需对混淆表示解码。此外，该运算的计算复杂度类似于运算⊕的计算复杂度。因而，该运算可以高效地执行。因此，不必对w_0和w_1的混淆表示去混淆以用于执行它们之间的运算，以这种方式在不增加太多复杂度的情况下改进系统的安全性。

[0023] 例如，可以存在域W、Σ和Z，其限定成使得X_0、Y_0、X_1和Y_1是Z的元素，并且w_0和w_1是W的元素，并且σ_0和σ_1是Σ的元素，并且$A_0: W \times W \rightarrow Z$，$A_1: W \times W \rightarrow Z$，$B_0: \Sigma \times \Sigma \rightarrow Z$，$B_1: \Sigma \times \Sigma \rightarrow Z$。运算符⊕可以在Z上限定，运算符⊗可以在W上限定，并且运算符Δ可以在Σ上限定。运算⊕是可交换的(也就是说，对于所有$z_1, z_2 \in Z$，$z_1 \oplus z_2 = z_2 \oplus z_1$)和可结合的，也就是说，对于所有$z_1, z_2, z_3 \in Z$，$(z_1 \oplus z_2) \oplus z_3 = z_1 \oplus (z_2 \oplus z_3)$。从W到Z的映射$A_0$、$A_1$可以使得对于所有$w_0, w_1 \in W$以及$i = 0,1, A_i(w_0 \otimes w_1) = A_i(w_0) \oplus A_i(w_1)$。这可以通过说$A_0$和$A_1$为线性的来表达。从Σ到Z的映射$B_0$、$B_1$可以使得对于所有$\sigma_0, \sigma_1 \in \Sigma$以及$i = 0,1, B_i(\sigma_0 \, \Delta \, \sigma_1) = B_i(\sigma_0) \oplus B_i(\sigma_1)$。我们将通过说$B_0$和$B_1$为线性的来表达这一点。此外，$A_0$、$B_0$、$A_1$和$B_1$被选择成使得有可能从$A_0(w) \oplus B_0(\sigma)$和$A_1(w) \oplus B_1(\sigma)$的组合唯一地确定$w \in W$。也就是说，如果$w, w' \in W$和$\sigma, \sigma' \in \Sigma$使得对于i=1,2，$A_i(w) \oplus B_i(\sigma) = A_i(w') \oplus B_i(\sigma')$，则$w = w'$。

[0024] 系统可以进一步包括混淆构件，其配置成生成基于第一数据值w_0的第一混淆表示(X_0,Y_0)和基于第二数据值w_1的第二混淆表示(X_1,Y_1)。

[0025] 系统可以进一步包括去混淆构件，其配置成对第三数据值w_2的混淆表示(X_2,Y_2)去混淆，以便通过从以下方程组获取第三数据值w_2：

[0026] $$X_2 = A_0(w_2) \oplus B_0(\sigma_2)$$

[0027] $$Y_2 = A_1(w_2) \oplus B_1(\sigma_2),$$

[0028] 其中σ_2是用于向第三数据值w_2的混淆表示(X_2,Y_2)提供冗余的状态变量。

[0029] 系统可以进一步包括状态生成器以用于随机地或伪随机地生成状态变量σ_0的值和/或状态变量σ_1的值，并且其中混淆构件配置成基于第一数据值w_0和状态变量σ_0来生成第一混淆表示(X_0,Y_0)并且基于第二数据值w_1和状态变量σ_1来生成第二混淆表示(X_1,Y_1)。这允许通过控制由状态变量σ_0和/或σ_1施加的增加的冗余而创建强混淆。

[0030] 混淆构件可以配置成在查找表中查找第一混淆表示(X_0,Y_0)和第二混淆表示(X_1,Y_1)。此外或者可替换地，去混淆构件可以配置成在查找表中查找第三数据值w_2。这是实现混淆的高效方式。利用查找表的实现方式还使得攻击者更加难以破坏混淆。

[0031] 混淆构件和去混淆构件可以是第一设备的部分，其中确定构件是第二不同设备的部分。第一设备可以进一步包括传送构件和接收构件，并且第二设备可以进一步包括传送构件和接收构件。第一设备的传送构件可以配置成向第二设备的接收构件传送第一混淆表示(X_0,Y_0)和第二混淆表示(X_1,Y_1)。第二设备的传送构件可以配置成向第一设备的接收构件传送混淆表示(X_2,Y_2)。该配置允许将⊗运算委派给第二设备，而没有准许第二设备对非混淆(或明文)数据值w_0，w_1和w_2的访问。

[0032] 确定构件可以配置成以明文执行从X_0和X_1计算X_2以及从Y_0和Y_1计算Y_2中的至少一个。这允许X_2和Y_2的高效计算，而无需单独地混淆计算，但是仍旧不会向攻击者暴露原始数据值。

[0033] w_0、w_1、w_2、σ_0、σ_1、σ_2、X_0、X_1、X_2、Y_0、Y_1和Y_2的值可以是具有相同位数的值。这促进了实现方式。

[0034] 运算符A_0、B_0、A_1和B_1可以是可逆运算符。这使得更容易设计系统参数。

[0035] 运算符⊕可以是按位XOR运算。这是特别适用于该应用的运算。按位XOR运算可以借助于至少一个XOR机器指令来执行。这是计算XOR运算的高效方式，并且不会向攻击者暴露原始数据值。

[0036] 在本发明的另一方面中，提供了一种用于使用数据的混淆表示在数据上执行运算的方法。该方法包括以下步骤：

[0037] 获取第一数据值w_0的第一混淆表示(X_0,Y_0)并且获取第二数据值w_1的第二混淆表示(X_1,Y_1)，其中以下关系成立：

[0038] $$X_0 = A_0(w_0) \oplus B_0(\sigma_0)$$

[0039] $$Y_0 = A_1(w_0) \oplus B_1(\sigma_0)$$

[0040] $$X_1 = A_0(w_1) \oplus B_0(\sigma_1)$$

[0041] $$Y_1 = A_1(w_1) \oplus B_1(\sigma_1),$$

[0042] 其中

[0043] ⊕是运算符，

[0044] A_0、B_0、A_1和B_1是线性运算符，并且将(u,v)映射到$(A_0(u) \oplus B_0(v), A_1(u) \oplus B_1(v))$的

运算符E关于u可逆，并且

[0045] σ_0和σ_1是用于向混淆表示提供冗余的状态变量；以及

[0046] 通过在第一数据值w_0的混淆表示(X_0,Y_0)和第二数据值w_1的混淆表示(X_1,Y_1)上执行以下运算来确定第三数据值w_2的混淆表示(X_2,Y_2)：

[0047] $X_2=X_0 \oplus X_1$

[0048] $Y_2=Y_0 \oplus Y_1$，

[0049] 其中$w_2=w_0 \otimes w_1$，其中$\otimes$为运算符。

[0050] 在另一方面中，提供了一种计算机程序产品，包括用于使处理器系统执行所阐述的方法的指令。

[0051] 本领域技术人员将领会到，以上提及的本发明的实施例、实现方式和/或方面中的两个或更多可以以被认为有用的任何方式组合。

[0052] 对应于系统的所述修改和变形的图像采集装置、工作站、系统、方法和/或计算机程序产品的修改和变形可以由本领域技术人员基于当前描述而实施。

附图说明

[0053] 本发明的这些和其它方面从以下描述的实施例显而易见并且将参照以下描述的实施例进行阐述。在附图中：

[0054] 图1是用于使用输入数据值的混淆表示安全地执行运算的系统的框图。

[0055] 图2是图示了使用输入数据值的混淆表示安全地执行运算的方法的图。

[0056] 图3是图示了在使用输入数据值的混淆表示安全地执行运算之后去混淆数据的方法的图。

具体实施方式

[0057] 在许多应用中，必要的是以安全方式向第一输入数据值w_0和第二输入数据值w_1应用运算，其中第一输入数据值w_0的第一混淆表示Z_0和第二输入数据值w_1的第二混淆表示Z_1是可获得的。将期望的是，针对恶意用户隐藏第一输入数据值w_0和第二输入数据值w_1，即便恶意用户能够完全访问设备，包括访问工作存储器，或者即便是恶意用户具有实用调试工具分析应用的能力。

[0058] 因此，代替于计算值w_0和w_1以及执行运算，可以使用第一输入数据值w_0的第一混淆表示Z_0和第二输入数据值w_1的第二混淆表示Z_1来执行运算。

[0059] 要指出的是，Z_0和Z_1可以分成两个分量，使得$Z_0=(X_0,Y_0)$并且$Z_1=(X_1,Y_1)$。

[0060] 图1图示了用于执行安全运算的系统的实施例。在图示中，已经通过矩形标示若干处理构件。

[0061] 此外，数据元素已经通过它们的可变符号以及概略阵列来指示，该阵列用符号表征给定长度的位序列。然而，每一个数据元素的位序列的实际长度可以变化。附图没有指示数据元素的实际长度。系统可以实现在单个处理设备上，诸如适当编程的计算机、智能电话或智能卡。系统也可以分布在若干不同处理设备之上。

[0062] 系统包括获取构件以用于获取第一数据值w_0的第一混淆表示(X_0,Y_0)和第二数据值w_1的第二混淆表示(X_1,Y_1)，其中以下方程成立：

[0063] $X_0 = A_0(w_0) \oplus B_0(\sigma_0)$

[0064] $Y_0 = A_1(w_0) \oplus B_1(\sigma_0)$

[0065] $X_1 = A_0(w_1) \oplus B_0(\sigma_1)$

[0066] $Y_1 = A_1(w_1) \oplus B_1(\sigma_1),$

[0067] 其中⊕是运算符，A_0、B_0、A_1和B_1是线性运算符，将(u,v)映射到(A_0(u) ⊕ B_0(v)，A_1(u) ⊕ B_1(v))的运算符E关于u可逆，并且σ_0和σ_1是用于向混淆表示提供冗余的状态变量。运算符⊕和⊗可以是按位XOR运算。可替换地，运算符可以是在给定域上限定的算术加法。

[0068] 要指出的是，可以存在域W、Σ和Z，其定义成使得X_0、Y_0、X_1和Y_1是Z的元素；w_0和w_1是W的元素，以及σ_0和σ_1是Σ的元素，并且$A_0: W \times W \longrightarrow Z$，$A_1: W \times W \longrightarrow Z$，$B_0: \Sigma \times \Sigma \longrightarrow Z$，$B_1: \Sigma \times \Sigma \longrightarrow Z$。运算符⊕可以在Z上限定，运算符⊗可以在W上限定，并且运算符Δ可以在Σ上限定。运算符A_0、B_0、A_1和B_1是线性运算符。这意味着，例如，对于W中的所有w_0和w_1，$A_0(w_0 \otimes w_1) = A_0(w_0) \oplus A_0(w_1)$；对于W中的所有$w_0$和$w_1$，$A\ (w_0 \otimes w_1) = A_1(w_0) \oplus A_1(w_1)$；$B_0(\sigma_0\ \Delta\ \sigma_1) = B_0(\sigma_0) \oplus B_0(\sigma_1)$；并且$B_1(\sigma_0\ \Delta\ \sigma_1) = B_1(\sigma_0) \oplus B_1(\sigma_1)$。

[0069] 运算符⊕是可交换的(也就是说，对于所有$z_1, z_2 \in Z$，$z_1 \oplus z_2 = z_2 \oplus z_1$)和可结合的，也就是说，对于所有$z_1, z_2, z_3 \in Z$，$(z_1 \oplus z_2) \oplus z_3 = z_1 \oplus (z_2 \oplus z_3)$。从W到Z的映射$A_0$、$A_1$使得对于所有$w_0, w_1 \in W$以及$i = 0,1, A_i(w_0 \otimes w_1) = A_i(w_0) \oplus A_i(w_1)$。从Σ到Z的映射$B_0$、$B_1$使得对于所有$\sigma_0, \sigma_1 \in \Sigma$以及$i = 0,1, B_i(\sigma_0\ \Delta\ \sigma_1) = B_i(\sigma_0) \oplus B_i(\sigma_1)$。最后，应当可行的是从$A_0(w) \oplus B_0(\sigma)$和$A_1(w) \oplus B_1(\sigma)$确定$w \in W$。也就是说，如果$w' w' \in W$和$\sigma, \sigma' \in \Sigma$使得对于i=1,2，$A_i(w) \oplus B_i(\sigma) = A_i(w') \oplus B_i(\sigma')$，则$w = w'$。例如，其中$E: (w, \sigma) \mapsto \left(A_0(w) \oplus B_0(\sigma), A_1(w) \oplus B_1(\sigma)\right)$的映射$E: W \times \Sigma \longrightarrow Z \times Z$是可逆的。一般地，从给定$X, Y \in Z$和$\sigma \in \Sigma$，应当有可能获取w。

[0070] 现在将讨论具体示例以说明该原理。要注意，所选集合和运算可以不同地且以更复杂的方式来选择以更好地混淆数据值。在该示例中，$W=\{0,1\}^3$，$\Sigma = \{0,1\}^2$并且$Z=\{0,1\}^2$。换言之，W是所有三位值的集合，Σ是所有二位值的集合，并且Z是所有二位值的集合。运算符⊕、⊗和Δ是在其相应域上的按位XOR运算符。该示例的线性运算符在其相应域上定义如下：

[0071] $A_0(w_1, w_2, w_3) = (w_1, w_3)$

[0072] $B_0(\sigma_1, \sigma_2) = (0, \sigma_1)$

[0073] $A_1(w_1,w_2,w_3)=(0,w_2)$

[0074] $B_1(\sigma_1,\sigma_2)=(\sigma_1,0)$。

[0075] 具有状态参数$\sigma=(\sigma_1,\sigma_2)$的值$w=(w_1,w_2,w_3)$的混淆表示$(X,Y)=((x_1,x_2),(y_1,y_2))$然后可以计算如下：

[0076] $X=(x_1,x_2)=A_0(w_1,w_2,w_3)+B_0(\sigma_1,\sigma_2)=(w_1,w_3)+(0,\sigma_1)=(w_1+0,w_3+\sigma_1)=(w_1,w_3+\sigma_1)$；

[0077] $Y=(y_1,y_2)=A_1(w_1,w_2,w_3)+B_1(\sigma_1,\sigma_2)=(0,w_2)+(\sigma_1,0)=(0+\sigma_1,w_2+0)=(\sigma_1,w_2)$。

[0078] 要注意，由于需要去混淆数据，所以$((x_1,x_2),(y_1,y_2))$的每一个值唯一地限定(w_1,w_2,w_3)的值，因为从任何给定$((x_1,x_2),(y_1,y_2))$和(σ_1,σ_2)有可能唯一地确定(w_1,w_2,w_3)，因为$A_1(w_1,w_2,w_3)+B_1(\sigma_1,\sigma_2)=(\sigma_1,x_2)$并且$A_0(x_1,x_2,x_3)+B_0(\sigma_1,\sigma_2)=(x_1,\sigma_1+x_2)$。

[0079] 在该具体示例中，(σ_1,σ_2)的值由$((x_1,x_2),(y_1,y_2))$的值唯一地限定。然而，不必能够恢复(σ_1,σ_2)的值，因为感兴趣的数据由(w_1,w_2,w_3)体现。

[0080] 另一简化示例在下文中呈现。在该情况下，W、Σ、Z等于正实数的集合。运算符Δ和⊕是实数乘法，并且运算符⊗是实数加法。此外，线性运算符被选择如下：$A_0(w)=w,A_1(w)=w^2,B_0(\sigma)=B_1(\sigma)=e^{\sigma}$。而且在该情况下，w可以从给定的(X,Y)和σ恢复。实际上，我们可以通过执行除法而从$A_0(w)\oplus B_0(\sigma)=we^{\sigma}$和$A_1(w)\oplus B_1(\sigma)=w^2e^{\sigma}$获取w。

[0081] 在下文中，运算符由所有域W、Σ和Z上的⊕指示。然而，应当谨记的是，原则上，W、Σ和Z上的运算符可以全部是不同的运算符。可替换地，例如如果W=Σ=Z，则可以在每一个域上使用相同运算符。

[0082] 在具体示例中，w_0、σ_0、X_0、Y_0、w_1、σ_1、X_1和Y_1全部是具有相同位数的数据值。例如，w_0、σ_0、X_0、Y_0、w_1、σ_1、X_1和Y_1可以具有8位，或者可以具有作为2的倍数的位数，以便以更高效的方式实现系统。

[0083] 在具体示例中，A_0、B_0、A_1和B_1中的至少一个是可逆线性运算符。在更具体的示例中，A_0、B_0、A_1和B_1中的每一个是可逆线性运算符。

[0084] 系统可以包括用于确定第一输入数据值w_0和第二输入数据值w_1的数据输入单元100。例如，输入单元100被配置成经由设备的通信子系统接收第一输入数据值w_0和第二输入数据值w_1。可替换地，输入单元100可以配置成从存储器接收输入数据值，存储器可以是内部存储器或外部存储器。

[0085] 例如，获取构件可以包括混淆构件101，其配置成从数据输入单元100接收作为输入值的第一输入数据值w_0和第二输入数据值w_1，并且基于第一输入数据值w_0而生成第一混淆表示(X_0,Y_0)以及基于第二输入数据值w_1而生成第二混淆表示(X_1,Y_1)。例如，混淆表示与数据值之间的关系可以预计算并且存储在查找表中。可选地，混淆构件101包括用于生成状态变量σ_0的值和/或状态变量σ_1的值的状态生成器。这些值可以例如随机地或者伪随机地生成。例如，这些值可以分别取决于w_0和w_1。混淆构件101可以配置成基于第一数据值w_0和状态

变量σ_0来生成第一混淆表示(X_0,Y_0)，并且基于第二数据值w_1和状态变量σ_1来生成第二混淆表示(X_1,Y_1)。在该情况下，例如，混淆表示与数据值和状态值对之间的关系可以预计算并且存储在查找表中。

[0086] 可替换地，获取构件被配置成以不同方式获取第一混淆表示(X_0,Y_0)和第二混淆表示(X_1,Y_1)。例如，这些值可以从外部源接收，或者可以是关于其它数据的混淆表示的计算的结果。

[0087] 系统进一步包括确定构件102。确定构件102被配置成确定数据值w_2的混淆表示(X_2,Y_2)，其中$w_2=w_0 \oplus w_1$。更具体地，确定构件102计算：

[0088] $X_2=X_0 \oplus X_1$

[0089] $Y_2=Y_0 \oplus Y_1$。

[0090] 在特定示例中，这些运算$\oplus$以明文计算。例如，在$\oplus$是XOR运算的情况下，该运算可以使用在其上实现系统的设备的处理器的对应XOR机器指令来执行运算。

[0091] 由于运算符$\oplus$的可交换和可结合的性质以及若干运算符的线性，下式成立：

[0092]

$$X_2 = X_0 \oplus X_1 = A_0(w_0) \oplus B_0(\sigma_0) \oplus A_0(w_1) \oplus B_0(\sigma_1) = A_0(w_0) \oplus A_0(w_1) \oplus B_0(\sigma_0) \oplus B_0(\sigma_1) = A_0(w_0 \oplus w_1) \oplus B_0(\sigma_0 \oplus \sigma_1)$$

$$Y_2 = Y_0 \oplus Y_1 = A_1(w_0) \oplus B_1(\sigma_0) \oplus A_1(w_1) \oplus B_1(\sigma_1) = A_1(w_0) \oplus A_1(w_1) \oplus B_1(\sigma_0) \oplus B_1(\sigma_1) = A_1(w_0 \oplus w_1) \oplus B_1(\sigma_0 \oplus \sigma_1)$$

。

[0093] 鉴于此，(X_2,Y_2)是$(w_0 \oplus w_1, \sigma_0 \oplus \sigma_1)$的混淆表示。如之前所限定的，$w_2=w_0 \oplus w_1$。当限定为$\sigma_2=\sigma_0 \oplus \sigma_1$，我们有$(X_2,Y_2)$是$w_2$的混淆表示，其中$\sigma_2$作为状态变量。

[0094] 要注意，混淆构件101可以借助于查找表实现。例如，混淆构件101可以通过单个查找表实现。可选地，这些查找表可以进一步通过使用从例如Chow等人得知的技术对查找表的输入和输出进行编码而被混淆。

[0095] 在去混淆之前，混淆值(X_2,Y_2)可以可选地经受另外的混淆处理，例如通过执行附加的$\oplus$运算或者其它类型的运算。当是时候恢复由任何所获取的混淆值表示的数据值时，混淆值可以被提供给去混淆构件以用于去混淆。相应地，系统可以进一步包括去混淆构件103。去混淆构件103可以接收数据值w_2的混淆表示(X_2,Y_2)并且可以对数据值w_2的混淆表示(X_2,Y_2)去混淆，以便通过求解上文提及的下述方程组来获取w_2：

[0096] $$X_2 = A_0(w_2) \oplus B_0(\sigma_2)$$

[0097] $$Y_2 = A_1(w_2) \oplus B_1(\sigma_2),$$

[0098] 其中σ_2是向混淆表示(X_2,Y_2)提供冗余的状态变量。

[0099] 系统可以进一步包括输出单元，其被配置成从去混淆构件103接收所计算的w_2的值并且将w_2的值转发给系统的其它组件(未示出)，和/或将w_2的值存储在存储器中。例如，输出单元可以配置成在显示设备上显示数据w_2的可视化和/或在音频设备上再现数据。

[0100] 输入构件100和/或混淆构件101可以是第一设备的部分，并且确定构件102可以是第二设备的部分，其中第一设备是与第二设备不同的设备。例如，输入构件100可以从存储器或者从外部源接收第一输入数据值w_0和第二输入数据值w_1，并且将它们提供给混淆构件

101，混淆构件101计算第一输入数据值w_0的第一混淆表示(X_0,Y_0)和第二输入数据值w_1的第二混淆表示(X_1,Y_1)。第一设备可以包括传送器构件。该传送器构件可以将第一输入数据值w_0的第一混淆表示(X_0,Y_0)和第二输入数据值w_1的第二混淆表示(X_1,Y_1)传送给第二设备。第二设备可以包括接收构件。该接收构件可以从第一设备接收第一输入数据值w_0的第一混淆表示(X_0,Y_0)和第二输入数据值w_1的第二混淆表示(X_1,Y_1)，并且将它们提供给确定构件102。确定构件102可以以上文所阐述的方式确定数据值w_2的混淆表示(X_2,Y_2)，其中$w_2=w_0 \oplus w_1$。去混淆模块103(以及可选的输出单元)可以是第一设备的部分，或者它们可以是第二设备的部分，或者它们可以是另外的第三设备的部分。相应地，第二设备可以包括传送器，其被配置成将混淆表示(X_2,Y_2)传送给第一或第三设备。

[0101] 图2图示了使用输入数据值的混淆表示安全地执行运算的方法。

[0102] 该方法包括步骤201：混淆第一输入数据值w_0和第二输入数据值w_1以生成第一输入数据值w_0的第一混淆表示(X_0,Y_0)和第二输入数据值w_1的第二混淆表示(X_1,Y_1)。第一输入数据值w_0的第一混淆表示(X_0,Y_0)和/或第二输入数据值w_1的第二混淆表示(X_1,Y_1)可以通过计算以下方程而生成：

[0103] $$X_0 = A_0(w_0) \oplus B_0(\sigma_0)$$

[0104] $$Y_0 = A_1(w_0) \oplus B_1(\sigma_0)$$

[0105] $$X_1 = A_0(w_1) \oplus B_0(\sigma_1)$$

[0106] $$Y_1 = A_1(w_1) \oplus B_1(\sigma_1)。$$

[0107] 第一输入数据值w_0的第一混淆表示(X_0,Y_0)和/或第二输入数据值w_1的第二混淆表示(X_1,Y_1)可以通过在查找表中查找而生成。查找表可以限定数据值w_3的混淆表示(X_3,Y_3)与第一输入数据值w_0的混淆表示(X_0,Y_0)之间的关系。

[0108] 该方法可以进一步包括步骤202：确定第三数据w_2的混淆表示(X_2,Y_2)，其中$w_2=w_0 \oplus w_1$。第三数据w_2的混淆表示(X_2,Y_2)可以通过执行以下运算来确定：

[0109] $X_2=X_0 \oplus X_1$

[0110] $Y_2=Y_0 \oplus Y_1$

[0111] 其中(X_0,Y_0)可以是第一输入数据值w_0的第一混淆表示并且(X_1,Y_1)可以是第二输入数据值w_1的第二混淆表示。

[0112] 该方法可以进一步包括步骤203：发送所确定的第三数据w_2的混淆表示(X_2,Y_2)以用于进一步的处理(例如用于执行新的运算)或者用于存储在查找表中，其中查找表可以随后用于生成混淆表示。

[0113] 图3图示了在使用输入数据值的混淆表示执行运算之后去混淆混淆数据的方法。

[0114] 该方法可以包括步骤301：接收第一输入数据值w_0的第一混淆表示(X_0,Y_0)和二输入数据值w_1的第二混淆表示(X_1,Y_1)。第一输入数据值w_0的第一混淆表示(X_0,Y_0)和/或二输入数据值w_1的第二混淆表示(X_1,Y_1)可能已经通过计算以下方程而生成：

[0115] $$X_0 = A_0(w_0) \oplus B_0(\sigma_0)$$

[0116] $$Y_0 = A_1(w_0) \oplus B_1(\sigma_0)$$

[0117] $X_1 = A_0(w_1) \oplus B_0(\sigma_1)$

[0118] $Y_1 = A_1(w_1) \oplus B_1(\sigma_1)$。

[0119] 第一输入数据值w_0的第一混淆表示(X_0,Y_0)和/或二输入数据值w_1的第二混淆表示(X_1,Y_1)可能已经使用查找表而生成。查找表可以限定数据值w_3的混淆表示(X_3,Y_3)与第一输入数据值w_0的混淆表示(X_0,Y_0)之间的关系。

[0120] 该方法可以进一步包括步骤302：确定第三数据w_2的混淆表示(X_2,Y_2)，其中$w_2=w_0 \oplus w_1$。第三数据w_2的混淆表示(X_2,Y_2)可以通过执行以下运算来确定：

[0121] $X_2=X_0 \oplus X_1$

[0122] $Y_2=Y_0 \oplus Y_1$

[0123] 其中(X_0,Y_0)可以是第一输入数据值w_0的第一混淆表示并且(X_1,Y_1)可以是第二输入数据值w_1的第二混淆表示。

[0124] 该方法可以进一步包括步骤303：去混淆所确定的第三数据w_2的混淆表示(X_2,Y_2)以便获取w_2。去混淆可以通过求解下述方程组来执行：

[0125] $X_2 = A_0(w_2) \oplus B_0(\sigma_2)$

[0126] $Y_2 = A_1(w_2) \oplus B_1(\sigma_2)$,

[0127] 其中$\oplus$是运算符，A_0、B_0、A_1和B_1是关于运算符$\oplus$为线性的运算符，并且将(u,v)映射到$(A_0(u) \oplus B_0(v), A_1(u) \oplus B_1(v))$的运算符E关于u可逆，并且$\sigma_2$是用于向混淆表示提供冗余的状态变量。

[0128] 去混淆值w_2可以被发送给另一单元以用于进一步的处理(例如用于执行新的运算或者用于显示目的)或者用于存储在查找表中，其中查找表可以随后用于去混淆混淆表示。

[0129] 将领会到的是，本发明还适用于计算机程序，特别地适于将本发明付诸实践的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，诸如部分编译的形式，或者适合用在根据本发明的方法的实现方式中的任何其它形式。还将领会到的是，这样的程序可以具有许多不同的架构设计。例如，实现根据本发明的方法或系统的功能性的程序代码可以细分成一个或多个子例程。在这些子例程之中分配功能性的许多不同方式对于本领域技术人员将是显然的。子例程可以一起存储在一个可执行文件中以形成自包含式程序。这样的可执行文件可以包括计算机可执行指令，例如处理器指令和/或解译器指令(例如Java解译器指令)。可替换地，一个或多个或者全部的子例程可以存储在至少一个外部库文件中并且静态地或者动态地与主程序链接，例如在运行时间。主程序包含对至少一个子例程的至少一个调用。子例程可以进一步包括对彼此的调用。涉及计算机程序产品的一个实施例包括对应于本文阐述的至少一个方法的每一个处理步骤的计算机可执行指令。这些指令可以细分成子例程和/或存储在可以静态地或动态地链接的一个或多个文件中。涉及计算机程序产品的另一实施例包括对应于本文阐述的系统和/或产品中的至少一个的每一个构件的计算机可执行指令。这些指令可以细分成子例程和/或存储在可以静态地或动态地链接的一个或多个文件中。

[0130] 计算机程序的载体可以是能够承载程序的任何实体或设备。例如，载体可以包括存储介质，诸如ROM，例如CD ROM或者半导体ROM，或者磁性记录介质，例如闪速驱动器或硬

盘。此外，载体可以是可以经由电缆或光缆或者通过无线电或其它措施运送的可传送载体，诸如电信号或光信号。当程序体现在这样的信号中时，载体可以通过这样的线缆或其它设备或构件而构成。可替换地，载体可以是程序嵌入其中的集成电路，该集成电路适于执行相关方法或者用在相关方法的执行中。

[0131] 应当指出的是，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可替换实施例而不脱离随附权利要求的范围。在权利要求中，放置在括号之间的任何参考标记不应当解释为限制权利要求。动词“包括”及其词形变化的使用不排除除权利要求中陈述的那些之外的元件或步骤的存在。元件之前的冠词“一”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分立元件的硬件而实现，以及借助于适当编程的计算机而实现。在枚举若干构件的设备权利要求中，这些构件中的若干个可以由同一个硬件项体现。在相互不同的从属权利要求中陈述某些措施的仅有事实并不指示这些措施的组合不能用于获益。

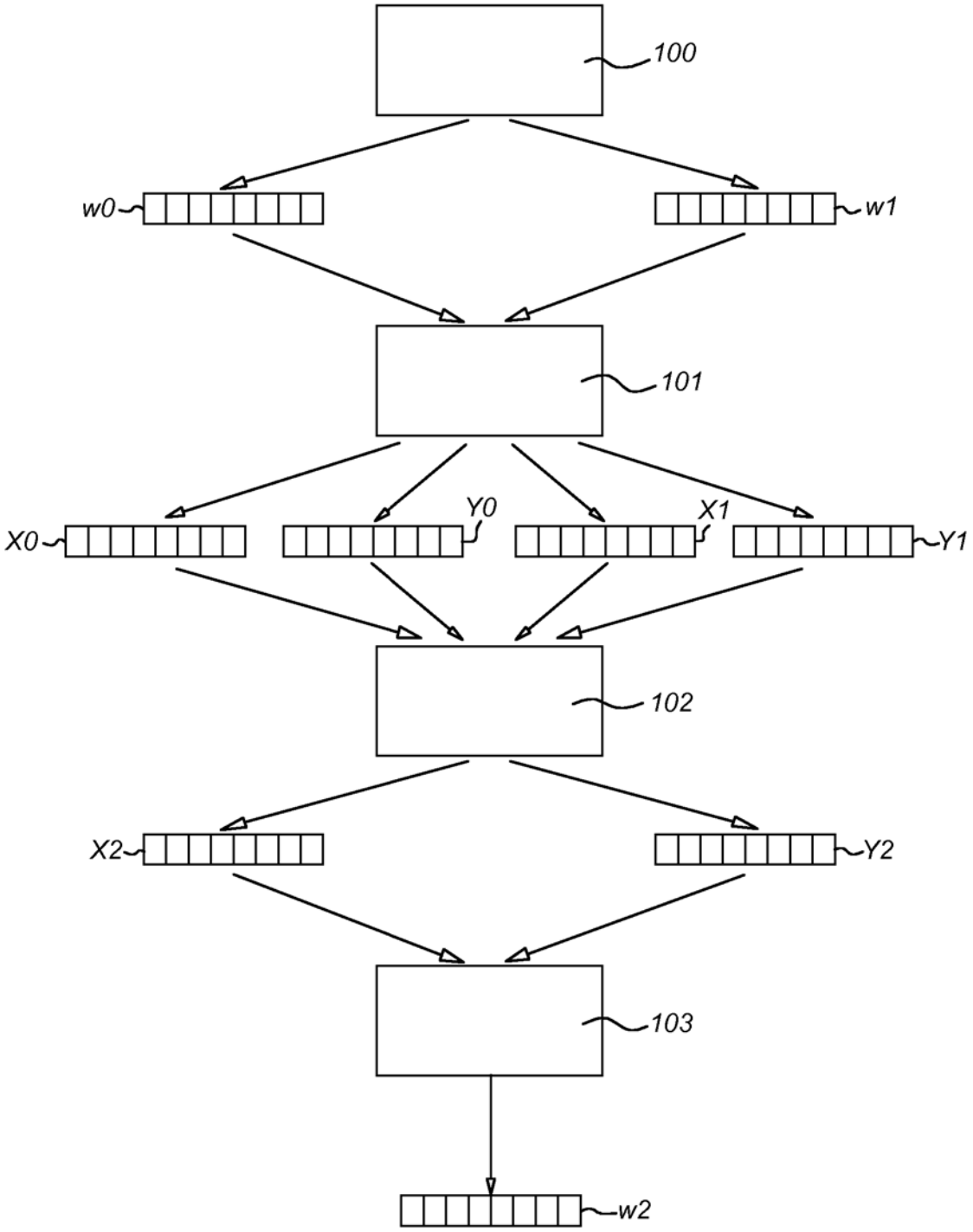


图 1

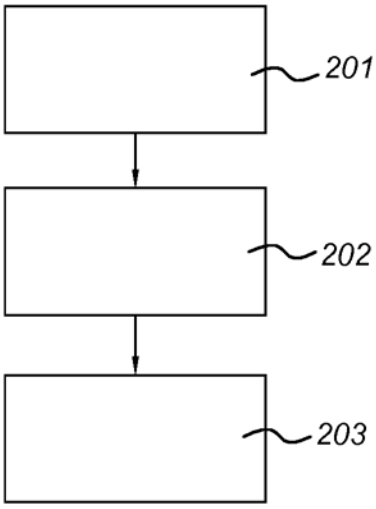


图 2

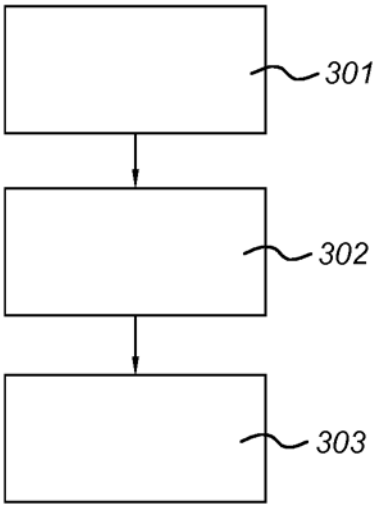


图 3