(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 110383275 B
(45)授权公告日 2023.08.22

(21)申请号 201780088267.2
(22)申请日 2017.12.13
(65)同一申请的已公布的文献号
申请公布号 CN 110383275 A
(43)申请公布日 2019.10.25
(30)优先权数据
102017204020.3 2017.03.10 DE
(85)PCT国际申请进入国家阶段日
2019.09.10
(86)PCT国际申请的申请数据
PCT/EP2017/082508 2017.12.13
(87)PCT国际申请的公布数据
WO2018/162107 DE 2018.09.13
(73)专利权人 西门子股份公司
地址 德国慕尼黑
(72)发明人 J.兹万茨格
(74)专利代理机构 中国专利代理(香港)有限公司 72001
专利代理师 孙云汉 刘春元
(51)Int.Cl.
G06F 21/14(2013.01)
H04L 9/00(2022.01)
(56)对比文件
US 2014101458 A1,2014.04.10
US 5892899 A,1999.04.06
EP 2937803 A1,2015.10.28
WO 2009010338 A1,2009.01.22
审查员 段玥

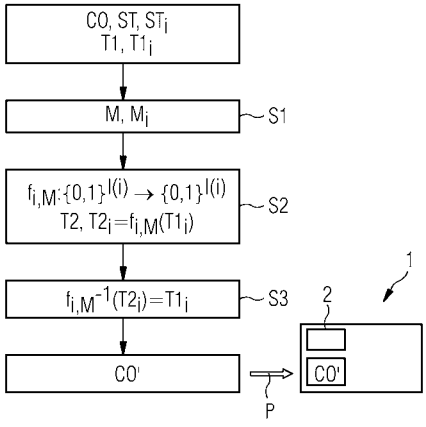
权利要求书2页 说明书5页 附图1页

(54)发明名称

用于程序代码的计算机辅助的代码混淆的方法

(57)摘要

本发明涉及一种用于程序代码(CO)的计算机辅助的代码混淆的方法,其中在所述程序代码(CO)中实施多个计算步骤(ST),其中所述多个计算步骤(ST)中的预先确定的计算步骤在实施所述程序代码(CO)时以预先确定的顺序被调用,而且所述预先确定的计算步骤中的至少有些计算步骤是预先规定的计算步骤(ST_i),在所述预先规定的计算步骤中,分别访问寄存在所述程序代码(CO)中的由多个第一数字表格值($T1_i$)构成的第一表格(T1),以便从所述第一表格(T1)中读出对于相应的预先规定的计算步骤(ST_i)来说所需的第一表格值($T1_i$)。在程序代码的代码混淆的范围内,使用由多个数字掩码值(M_i)构成的动态掩码(M),其中针对相应的预先规定的计算步骤使用另一掩码值(M_i),用于通过第二表格值($T2_i$)来替代所述第一表格(T1)的第一表格值($T1_i$)。此外,待进行代码混淆的程序代码(CO)被适配为使得在其运行期间在相应的预先规定的计算步骤(ST_i)中将所述第二表格值($T2_i$)计算回到原来的第一表格值($T1_i$)。按照本发明的方法能够实现对程序代码中的表格式寄存的值得保护的信息的高效的代码混淆。在此,对表格式信息的去掩码化在程序代码运行期间分布在整个程序代码,由此这些信息的重构对于未经授权的攻击者来说变得困难。



CN 110383275 B

1.一种用于程序代码(CO)的计算机辅助的代码混淆的方法,其中在所述程序代码(CO)中实施多个计算步骤(ST),其中所述多个计算步骤(ST)中的预先确定的计算步骤在执行所述程序代码(CO)时以预先确定的顺序被调用,而且所述预先确定的计算步骤中的至少有些计算步骤是预先规定的计算步骤(ST_i),在所述预先规定的计算步骤中,分别访问寄存在所述程序代码(CO)中的由多个第一数字表格值($T1_i$)构成的第一表格(T1),以便从所述第一表格(T1)中读出对于相应的预先规定的计算步骤(ST_i)来说所需的第一表格值($T1_i$),其中为了改变所述程序代码而实施如下步骤:

-生成动态掩码(M),所述动态掩码包括多个数字掩码值(M_i),其中所述动态掩码(M)中的至少一部分动态掩码的掩码值(M_i)彼此不同而且相应的掩码值(M_i)对于相应的预先规定的计算步骤(ST_i)来说有效;

-所述第一表格(T1)的每个第一表格值($T1_i$)都借助于在相应的预先规定的计算步骤(ST_i)中读入相应的第一表格值($T1_i$)时有效的掩码值(M_i)来被转换成第二数字表格值($T2_i$),由此得到由第二表格值($T2_i$)构成的第二表格(T2),所述第二表格替代所述第一表格(T1)被寄存在所述程序代码(CO)中;

-针对每个预先规定的计算步骤(ST_i)在所述程序代码(CO)中实施附加的计算步骤($f_{i,M}^{-1}$),所述附加的计算步骤将所读出的第二表格值($T2_i$)回算到对于相应的预先规定的计算步骤(ST_i)来说所需的那个第一表格值($T1_i$),

其中为了生成所述动态掩码(M),规定初始掩码值和更新步骤,其中通过将一个或多个连续的更新步骤应用到当前有效的掩码值(M_i)来计算对于下一个预先规定的计算步骤(ST_i)来说有效的掩码值(M_i),其中所述更新步骤也在所述程序代码(CO)中实施,使得在相应的预先规定的计算步骤(ST_i)中存在所述当前有效的掩码值(M_i),其中所述附加的计算步骤($f_{i,M}^{-1}$)在相应的预先规定的计算步骤中取决于所述当前有效的掩码值(M_i),其中所述更新步骤针对其所应用于的掩码值(M_i)中的至少一部分掩码值以彼此不一样的方式来被规定。

2.根据权利要求1所述的方法,其特征在于,在所述预先规定的计算步骤(ST_i)中的每个计算步骤中都实施更新步骤,其中此外也在不是预先规定的计算步骤(ST_i)的预先确定的计算步骤中的至少一部分计算步骤中分别实施更新步骤。

3.根据权利要求1或2所述的方法,其特征在于,所述初始掩码值(M_i)和/或所述更新步骤借助于随机数发生器来被规定。

4.根据权利要求1或2所述的方法,其特征在于,所述第一表格值($T1_i$)和所述第二表格值($T2_i$)以及所述掩码值(M_i)分别是位序列。

5.根据权利要求4所述的方法,其特征在于,通过在相应的第一表格值($T1_i$)的位序列与当前有效的掩码值(M_i)的位序列之间应用逻辑运算来将相应的第一表格值($T1_i$)转换成第二表格值($T2_i$),其中对所述逻辑运算的应用提供所述第二表格值($T2_i$)。

6.根据权利要求5所述的方法,其特征在于,所述逻辑运算逐位地被应用于相应的第一表格值($T1_i$)和所述当前有效的掩码值(M_i)的位序列的彼此对应的位,其中所述逻辑运算包括一个或多个OR和/或XOR和/或NOR和/或XNOR和/或AND和/或NAND运算。

7.根据权利要求6所述的方法,其特征在于,在所述当前有效的掩码值(M_i)的位序列比相应的第一表格值($T1_i$)的位序列短的情况下,所述当前有效的掩码值(M_i)的初始位序列通

过多次使用所述初始位序列的位来被延长，使得针对相应的第一表格值($T1_i$)的位序列的每个位都存在所述当前有效的掩码值(M_i)的位序列的对应的位。

8.根据权利要求7所述的方法，其特征在于，在所述当前有效的掩码值(M_i)的位序列比相应的第一表格值($T1_i$)的位序列短的情况下，所述当前有效的掩码值(M_i)的初始位序列通过重复所述初始位序列一次或多次来被延长，使得针对相应的第一表格值($T1_i$)的位序列的每个位都存在所述当前有效的掩码值(M_i)的位序列的对应的位。

9.一种用于执行利用根据权利要求1至8之一所述的方法来被代码混淆的程序代码(CO')的方法，其中在调用所述程序代码(CO')的相应的预先规定的计算步骤(ST_i)时，从第二表格(T2)中读出第二表格值($T2_i$)并且针对相应的预先规定的计算步骤(ST_i)来实施附加的计算步骤($f_{i,M}^{-1}$)，所述附加的计算步骤将所读出的第二表格值($T2_i$)回算到对于相应的预先规定的计算步骤(ST_i)来说所需的那个第一表格值($T1_i$)。

10.一种用于执行利用根据权利要求1至8之一所述的方法来被代码混淆的程序代码(CO')的技术设备，其特征在于，所述技术设备(1)包括计算装置(2)，所述计算装置被设立用于根据权利要求9所述的方法来执行所述程序代码(CO')。

11.根据权利要求10所述的技术设备，其特征在于，所述技术设备(1)是自动化设备或者自动化设备的组件或者电能生成和/或电能分配系统或者电能生成和/或电能分配系统的组件或者医疗设备。

12.一种机器可读载体，在其上存储有计算机程序，所述计算机程序具有程序代码部分，当所述程序代码部分在计算机上被执行时，所述程序代码部分用于执行根据权利要求1至8之一所述的方法。

用于程序代码的计算机辅助的代码混淆的方法

技术领域

[0001] 本发明涉及一种用于程序代码的计算机辅助的代码混淆的方法以及一种用于执行这种经代码混淆的程序代码的方法。本发明还包括一种技术系统、一种计算机程序产品和一种计算机程序。

背景技术

[0002] 常常存在保护程序代码中的信息以防第三方的未经授权的访问的需求。在此，常见的方案在于：利用适当的加密函数对信息进行加密或者借助于代码打包程序(Code-Packer)来对程序代码进行打包。在此，不利的是：在程序代码的运行期间，信息重新被解密或代码重新被解包，使得值得保护的信息在程序执行时完全以明文存在于程序存储器中，而且因此存在攻击者利用适当的技术从存储器中读出这些信息的危险。

发明内容

[0003] 本发明的任务是提供一种计算机辅助的方法，利用该方法来非常好地保护程序代码中的值得保护的信息以防在程序代码的运行期间第三方的未经授权的访问。

[0004] 该任务通过按照本发明的方法来解决。本发明的扩展方案在下文中限定。

[0005] 按照本发明的方法用于程序代码的计算机辅助的代码混淆(也就是说模糊处理)。在待模糊处理或代码混淆的程序代码中实施多个计算步骤，其中所述多个计算步骤中的预先确定的计算步骤在执行程序代码时以预先确定的顺序被调用。计算步骤以及也包括更下面提到的更新步骤的概念在此应宽泛地来理解。一个计算步骤不一定必须只包含一个单独的运算，而是该计算步骤可以由多个运算、必要时具有条件、循环、嵌套以及诸如此类的运算来组成。因此，通过计算步骤或更新步骤，可以将运算序列封装。

[0006] 待模糊处理的程序代码的预先确定的计算步骤包含预先规定的计算步骤。这些预先确定的计算步骤例如可以只包括预先规定的计算步骤，然而除了这些预先规定的计算步骤之外，必要时也可以设置其它计算步骤。这些预先规定的计算步骤的特点在于：在这些步骤中分别访问寄存在程序代码中的由多个第一数字表格值构成的第一表格，以便从第一表格中读出对于相应的预先规定的计算步骤来说所需的第一表格值。对于所有不是预先规定的计算步骤(只要存在的话)的预先确定的计算步骤来说，虽然还规定了对这些预先确定的计算步骤的调用的预先确定的顺序，但是这些计算步骤没有访问第一表格。

[0007] 表格的概念应宽泛地来理解。表格是数字值的集合，通过程序代码利用相对应的指令可以有针对性地访问该集合。此外，表格值的概念也应宽泛地来理解。一个表格值尤其可以由多个子值组成，这些子值在相对应地被调用的计算步骤中的不同的位置被处理。

[0008] 在按照本发明的对程序代码的代码混淆的范围内，生成动态掩码，该动态掩码包括多个数字掩码值，其中该动态掩码中的至少一部分动态掩码的掩码值以及优选地该掩码的所有掩码值都彼此不同而且相应的掩码值对于相应的预先规定的计算步骤来说有效。

[0009] 在按照本发明的方法的范围内，第一表格的每个第一表格值都借助于在相应的预

先规定的计算步骤中读入相应的第一表格值时有效的掩码值来被转换成第二数字表格值，由此得到由这些第二表格值构成的第二表格，该第二表格替代第一表格被寄存在程序代码中。换言之，在执行经代码混淆的程序代码时，不再访问第一表格值，而是访问相对应的第二表格值。在下文，在相应的预先规定的计算步骤中读入相应的第一表格值时有效的掩码值也被称作当前有效的掩码值。

[0010] 为了经代码混淆的程序代码提供与未经代码混淆的程序代码相同的结果，在代码混淆的范围内，还针对每个预先规定的计算步骤在程序代码中实施附加的计算步骤，在实施相对应的预先规定的计算步骤时，该附加的计算步骤将所读出的第二表格值回算到对于相应的预先规定的计算步骤来说所需的那个第一表格值。

[0011] 按照本发明的方法具有如下优点：借助于动态掩码来对表格式寄存的值得保护的信息非常好地进行模糊处理。在此，去掩码化或去模糊化的运算被分配到整个程序代码中的多个计算步骤。因此，在经代码混淆的程序代码的运行期间对经模糊处理的信息的重构变得非常困难。

[0012] 在一个特别优选的实施方式中，为了生成动态掩码，规定初始掩码值和更新步骤，其中通过将一个或多个连续的更新步骤应用到当前有效的掩码值来计算对于下一个预先规定的计算步骤来说有效的掩码值。初始掩码值和更新步骤也在程序代码中实施，使得在相应的预先规定的计算步骤中存在当前有效的掩码值，其中该附加的计算步骤在相应的预先规定的计算步骤中取决于当前有效的掩码值。由于在程序代码中实施更新步骤，攻击者为了重构在预先规定的计算步骤中的第一表格值而需要关于初始掩码值和之前的更新步骤的知识。因为该知识分布在程序代码中，所以实现了对程序代码中的信息的非常好的保护。

[0013] 为了使回算到原来的第一表格值变得困难，在刚刚描述的实施方式的一个优选的变型方案中，更新步骤针对其所应用于的掩码值中的至少一部分掩码值而且尤其是针对其所应用于的所有掩码值以彼此不一样的方式来被规定。在另一优选的变型方案中，在这些预先规定的计算步骤中的每个计算步骤中都设置更新步骤，其中此外优选地也在不是预先规定的计算步骤(只要存在的话)的预先确定的计算步骤中的至少一部分计算步骤中(尤其是在所有预先确定的计算步骤中)分别设置更新步骤。

[0014] 在另一优选的变型方案中，初始掩码值和更新步骤借助于随机数发生器来被规定。以这种方式实现了对这些值或步骤的非常任意的规定并且因此进一步改善了对程序代码的模糊处理。

[0015] 在另一优选的实施方式中，第一表格值和第二表格值以及也包括掩码值分别是位序列。优选地，在此，通过在第一表格值的位序列与当前有效的掩码值的位序列之间应用逻辑运算来将相应的第一表格值转换成第二表格值。在此，对逻辑运算的应用提供第二表格值。

[0016] 优选地，上文的逻辑运算逐位地被应用于相应的第一表格值和当前有效的掩码值的位序列的彼此对应的位，其中这些逻辑运算优选地包括一个或多个OR(或)和/或XOR(异或)和/或NOR(或非)和/或XNOR(同或)和/或AND(与)和/或NAND(与非)运算。因此，在当前有效的掩码值的位序列比第一表格值的位序列长的情况下，并不是将当前有效的掩码值的所有位都用于修改第一表格值。

[0017] 在当前有效的掩码值的位序列比相应的第一表格值的位序列短的情况下，在按照本发明的方法的一个优选的变型方案中，当前有效的掩码值的初始位序列通过多次使用初始位序列的位来延长，使得针对相应的第一表格值的位序列的每个位都存在当前有效的掩码值的位序列的对应的位。优选地，将位序列延长为使得初始位序列被重复一次或多次。以这种方式，即使掩码值的位长度比相对应的(第一)表格值的那个位长度短，也实现了对相对应的表格值的良好的模糊处理。

[0018] 除了用于程序代码的代码混淆的方法之外，本发明也涉及一种用于执行经代码混淆的程序代码的方法。在此，在调用(经代码混淆的)程序代码的相应的预先规定的计算步骤时，从第二表格中读出第二表格值并且针对相应的预先规定的计算步骤来实施附加的计算步骤，该附加的计算步骤将所读出的第二表格值回算到对于相应的预先规定的计算步骤来说所需的那个第一表格值。

[0019] 在已经利用了使用更新步骤来确定当前有效的掩码值的实施方式来生成经代码混淆的程序代码的情况下，在执行经代码混淆的程序代码的范围内，也实施在该经代码混淆的程序代码中实现的更新步骤。

[0020] 本发明还涉及一种技术系统，该技术系统包括计算装置，该计算装置被设立用于按照刚刚描述的方法来执行经代码混淆的程序代码。在此，技术系统的术语应宽泛地来理解，而且该技术系统也可以是单独的技术设备。在此，经代码混淆的程序代码可以寄存在不同的技术系统中。该技术系统例如可以是自动化设备或者自动化设备的组件或者电能生成和/或电能分配系统或者电能生成和/或电能分配系统的组件或者医疗设备。

[0021] 本发明还涉及一种计算机程序产品，该计算机程序产品具有被存储在机器可读载体上的程序代码部分，当所述程序代码部分被执行到计算机上时，所述程序代码部分用于执行按照本发明的用于程序代码的计算机辅助的代码混淆的方法或用于执行按照本发明的用于执行经代码混淆的程序代码的方法或用于执行这些方法的优选的变型方案。

[0022] 本发明还涉及一种计算机程序，该计算机程序具有程序代码部分，当所述程序代码部分在计算机上被执行时，所述程序代码部分用于执行按照本发明的用于程序代码的计算机辅助的代码混淆的方法或用于执行按照本发明的用于执行经代码混淆的程序代码的方法或用于执行这些方法的优选的变型方案。

[0023] 在上面的计算机程序产品或计算机程序用于实施用于程序代码的代码混淆的方法的情况下，利用相对应的程序代码部分来引起该代码混淆。因此，这些程序代码部分并不是待进行代码混淆的程序代码。

[0024] 在该计算机程序产品或该计算机程序用于执行经代码混淆的程序代码的情况下，这些程序代码部分对应于经代码混淆的程序代码。

附图说明

[0025] 随后，本发明的实施例依据随附的图1详细地予以描述。该附图示出了按照本发明的用于程序代码的代码混淆的方法的实施方式的流程图。

具体实施方式

[0026] 按照图1，该方法的起点是应在本发明的范围内被模糊处理的程序代码CO。该程序

代码包含多个计算步骤，所述多个计算步骤在图1中用ST来表示。在此，这些计算步骤ST_i(i＝1,...,n)中的一部分访问第一表格T1，其中相应的计算步骤从该表格中读出相对应的录入项$T1_i$。通过下标i来规定预先确定的顺序，计算步骤ST_i以该顺序依次被实施。在此，这些计算步骤ST_i对应于预先规定的计算步骤。在这些计算步骤ST_i之间也还可以实施程序代码CO的其它计算步骤，其中然而这些其它计算步骤没有访问表格T1。

[0027] 因此，按照程序代码CO，只在这些计算步骤ST_i中访问第一表格T1的确定的录入项$T1_i$。在此，相应的表格录入项是在相对应的计算步骤中被处理的位序列。在此，各个表格录入项的位序列具有长度l(i)。所述长度对于不同的表格录入项来说可能不一样大。

[0028] 在运行期间，程序代码CO根据已知的输入通过应用计算步骤ST来生成确定的输出。原则上，这些计算步骤ST在此可以以任意的顺序来经历，然而特殊的计算步骤ST_i彼此间总是以相同的顺序并且与输入无关地始终被经历正好一次。相应的计算步骤ST_i总是只访问一个单独的表格录入项$T1_i$，其中然而该计算步骤在其实施时允许任意频繁地访问该表格录入项。

[0029] 具有相对应的表格录入项$T1_i$的未经模糊处理的表格T1包含值得保护的信息，而且随后描述的实施方式的目标是对程序代码CO进行代码混淆，使得未经授权的第三方不能根据程序代码来重构这些表格录入项或者只能花费很高地根据程序代码来重构这些表格录入项。不同于将加密函数一次性地应用到整个表格T1，模糊处理分布在整个程序代码，由此使回算到原来的表格值变得非常困难。

[0030] 按照图1的步骤S1，为了程序代码CO的代码混淆，首先确定具有掩码录入项M_i(i＝1,...,n)的动态掩码M。因此，对于每个预先规定的计算步骤ST_i来说都存在相对应的掩码值M_i。该掩码值M_i只对于计算步骤ST_i有效。通过对这些掩码值的索引，规定了与计算步骤ST_i的实施相对应的顺序。各个掩码值M_i分别是具有确定的长度的位序列，其中该长度在这里所描述的实施方式中关于掩码值M_i保持相同，然而情况不一定必须如此。

[0031] 各个掩码值M_i通过更新步骤的序列来确定，这些更新步骤从任意的初始掩码值出发连续地被应用于最后被更新的掩码值。在此，更新步骤的概念应宽泛地来理解。特别是，一个更细步骤不仅可包含单独的运算，而且必要时可包含多个运算。在这里所描述的实施方式中，这些更新步骤是任意地被规定的并且因此彼此间至少部分不一样。虽然如此，各个更新步骤固定地预先规定并且包含在其中的运算以固定的顺序来实施。因此，按照步骤S1，生成具有任意的掩码分配或掩码值M_i的掩码M。优选地，在此使用随机数发生器来规定掩码的初始值以及规定更新步骤。

[0032] 接着，在步骤S2中，在相应的计算步骤ST_i中被读出的表格录入项$T1_i$借助于函数$f_{i,M}$来改变，该函数取决于在相对应的计算步骤ST_i中有效的掩码值M_i。在此，该函数$f_{i,M}$可以任意地来规定，重要的仅仅是：该函数取决于相应的掩码值M_i而且将表格录入项$T1_i$双射映射到第二表格T2的新的表格录入项$T2_i$。如从图1中可见，该函数$f_{i,M}$在该实施例中是如下映射，该映射将表格录入项$T1_i$的位序列映射到长度相同的另一位序列，其中该另一位序列对应于新的表格录入项$T2_i$。

[0033] 在针对所有计算步骤ST_i都生成了经修改的表格录入项$T2_i$之后，原来的表格录入项$T1_i$被这些新的表格录入项$T2_i$替代，使得对于攻击者来说不可能轻易地从程序代码CO中读出原来的表格录入项。由于对表格录入项的修改，程序代码CO也还必须按如下地被改变：

在相对应的计算步骤ST_i中，将所读出的表格录入项$T2_i$换算成原来的表格录入项$T1_i$。这按照图1在步骤S3中实现。在该步骤中，初始掩码值在程序代码的开头被寄存而上文的更新步骤也在程序代码中被实施，其中在这里所描述的实施方式中，在调用相应的计算步骤ST_i时，执行相对应的更新步骤来确定当前有效的掩码值。此外，在每个计算步骤中都实施附加步骤，该附加步骤借助于反函数$f_{i,M}^{-1}$将所读出的表格录入项$T2_i$换算成原来的表格录入项$T1_i$，该反函数取决于当前有效的掩码值M_i。

[0034] 因此，作为步骤S1至S3的结果，得到经代码混淆的程序代码CO'，在该经代码混淆的程序代码中，原来的表格T1被模糊处理。然而，通过回算到原来的表格值的回算步骤，实现了与在未经代码混淆的程序代码CO的情况下相同的计算结果。紧接着，经代码混淆的程序代码CO'可以被寄存在任意的技术设备中并且被执行。这在图1中通过箭头P来表示。在此，经代码混淆的程序代码CO'被存储在技术设备1上，该技术设备拥有计算装置2，利用该计算装置来执行经代码混淆的程序代码CO'。

[0035] 先前所描述的对程序代码的代码混淆可以被用在不同的技术领域。特别是，在此可以改变在SCADA系统、PLC(PLC＝ProgrammableLogicController(可编程逻辑控制器))、运动控制系统、自动化软件、计算机断层扫描设备、智能电网(SmartGrid)等等中的程序代码。在此，可以对任意的算法进行代码混淆，例如在PC(PC＝PersonalComputer(个人计算机))上的程序或者在设备上的固件。这些程序可以承担任意的任务，例如可以涉及控制和/或调节算法或者基于神经元网络的算法。

[0036] 通常，利用按照本发明的代码混淆实现了：在软件中的表格式存储并且值得保护的数据被模糊处理。值得保护的数据例如可以是加密信息或者是对于许可证检查来说重要的信息。

[0037] 先前所描述的按照本发明的方法的实施方式具有一系列优点。尤其是，由于表格的掩码化，攻击者丝毫不再能从该表格中提取出信息。更确切地说，为了可以合理地解释表格的经掩码化的形式，需要额外地了解掩码的初始值以及针对该掩码的更新步骤的整个序列。

[0038] 不同于一次性的解密或解包运算，对表格的去掩码化的步骤不是在程序代码中的逐点的位置上进行，而是该去掩码化分布在整个程序代码。因此，攻击者必须分析整个程序代码，以便反推出该表格的原来的值。与之相反，在对表格的一次性的加密的情况下，攻击者可以在了解解密的情况下直接访问表格录入项的明文。

[0039] 此外，在按照本发明的代码混淆中，在执行程序代码时丝毫不能依据存储器状态、也就是说尤其是当前的掩码值来轻易地推断出超过在当前的步骤中使用的表格录入项。在对掩码值的下一次更新时已经不再可能反推出之前的表格录入项。

[0040] 按照本发明的代码混淆方法还可以非常好地与逆向工程(ReverseEngineering)的其它技术、诸如反调试(Anti-Debug)措施或者自修改代码的使用相结合。以这种方式，重构去掩码化运算的整个序列从而提取出原来的表格值对于攻击者来说难上加难。

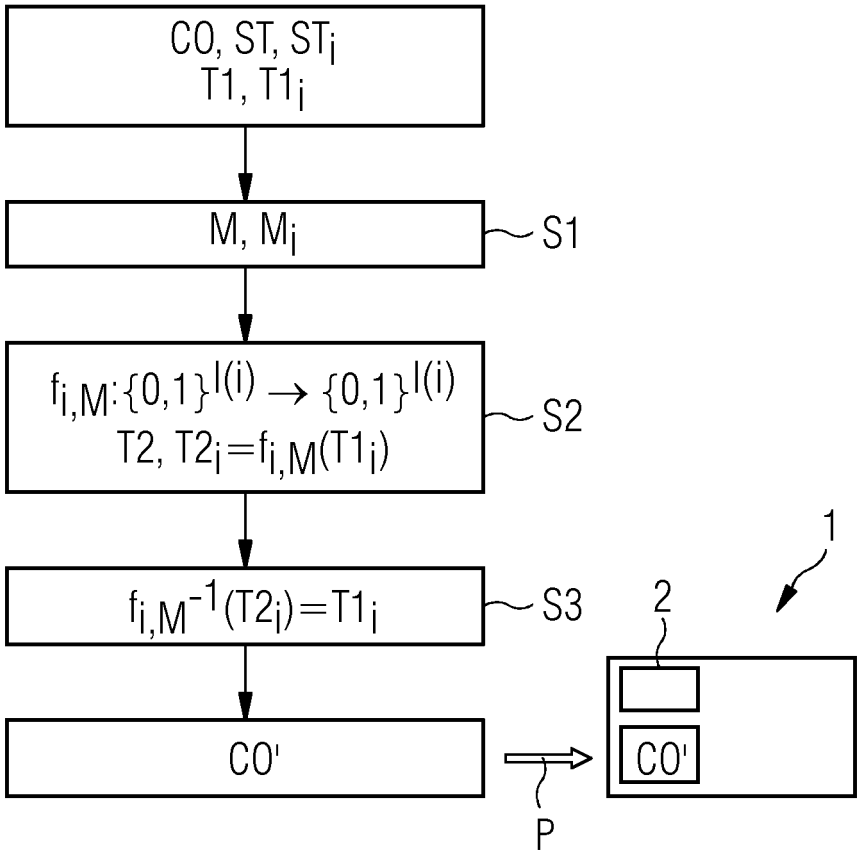

CO, ST, STi
T1, T1i
M, Mi
S1
fi,M:{0,1}l(i) → {0,1}l(i)
T2, T2i=fi,M(T1i)
S2
fi,M-1(T2i)=T1i
S3
CO'
CO'
1
2
P

图 1