[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 9/30 (2006.01)

[12] 发明专利说明书

专利号 ZL 02803824.X

[45] 授权公告日 2006 年 1 月 18 日

[11] 授权公告号 CN 1237442C

[22] 申请日 2002.1.16 [21] 申请号 02803824.X
[30] 优先权
[32] 2001.1.17 [33] DE [31] 10101956.4
[86] 国际申请 PCT/DE2002/000110 2002.1.16
[87] 国际公布 WO2002/057905 德 2002.7.25
[85] 进入国家阶段日期 2003.7.17
[71] 专利权人 因芬尼昂技术股份公司
地址 德国慕尼黑
[72] 发明人 H·哈特利布 H·塞德拉克
F·克鲁格
审查员 刘宇儒

[74] 专利代理机构 中国专利代理(香港)有限公司
代理人 吴立明 张志醒

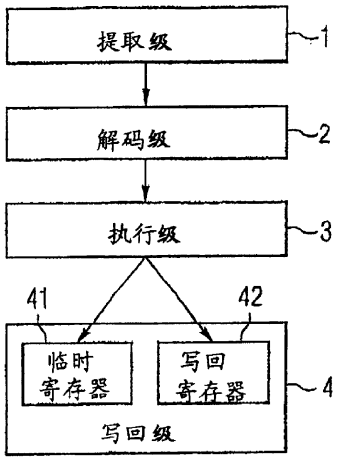
权利要求书 1 页 说明书 3 页 附图 1 页

[54] 发明名称

CPU 安全的提升方法

[57] 摘要

在本发明公开的方法中，使用包括提取级(1)、解码级(2)、执行级(3)及写回级(4)的流水线。写回级包含至少一个寄存器(41)，其使用不会导致 CPU 的任何状态的改变，以及包含至少一个寄存器(42)，其使用会导致 CPU 的状态改变。根据本发明，至少一个随机选择码序列会于解码级中插入，以作为保持位置码序列或虚设码序列，而使得藉由 DPA 的攻击变得较为困难。



1．一种提升 CPU 安全的方法，其中，一条 流水线具有至少一个解码级(2)以及一个写回级(4)，该写回级(4)包含至少一第一寄存器(41)以及包含至少一个第二寄存器(42)，所述第一寄存器(41)的使用不会导致该 CPU 状态的改变，所述第二寄存器(42)的使用会导致该 CPU 的状态改变，

该方法的特征为：

对于每次执行的一个特定程序时，选择至少一不会使该 CPU 的状态改变的随机选择的码序列，作为保持位置码序列或虚设码序列；并且利用该解码级(2)将该至少一个随机选择的码序列插入该特定程序的代码序列中使得获取程序的执行时间不同于先前的执行时间。

2．如权利要求 1 的方法，其中

利用一个或者多个随机确定的存储器地址从一个存储器读取该一个或者多个随机选择的码序列。

3．如权利要求 2 的方法，其中

一个只读存储器被用作存储器。

CPU 安全的提升方法

技术领域

本发明涉及提升 CPU 安全的方法。

背景技术

对于 CPU 的安全而言，差动功率分析(Differential power analysis，简称 DPA)为熟知的攻击状况，在这样的攻击中，CPU 中的一连串的程序命令及其效能，利用由功率消耗的特征的统计分析来决定。关于所执行的程序的详细的结论，可自这些分析中获得。

叙述于 DE 199 36 939 A1 及 WO 00/50977 的方法会使 DPA 变得较为困难，特别是关于智能卡的应用，只为了欺骗的目的，而藉由执行定义的处理器运作或程序步骤，以植入于以随机选择为基础的执行的程序中。

发明内容

本发明的目的就是定义提升 CPU 安全的方法。

在所述的方法中，一条流水线具有至少一个解码级以及一个写回级，该写回级包含至少一个第一寄存器以及包含至少一个第二寄存器，所述第一寄存器的使用不会导致该 CPU 状态的改变，所述第二寄存器的使用会导致该 CPU 的状态改变，该方法的特征为:对于每次执行的一个特定程序时，选择至少一个不会使该 CPU 的状态改变的随机选择的码序列，作为保持位置码序列或虚设码序列；并且利用该解码级将该至少一个随机选择的码序列插入该特定程序的代码序列中使得获取程序的执行时间不同于先前的执行时间。

在根据本发明的方法中，CPU 的结构使用流水线，具有至少一个解码级及一个写回级，并且通常包括提取级、解码级、执行级及写回级。写回级包含至少一个寄存器，其使用不会导致 CPU 的任何状态的改变，以及包含至少一个寄存器，其使用会导致 CPU 的状态改变。根据本发明，至少一个随机选择码序列会于解码级中插入，当作保持位置码序列或虚设码序列。理论上，这种方法可用于任何的流水线，特别而言，除了经由例子所指定的级外，其可具有另外的级，并且会配合附图做更详细地解释。

附图说明

图 1 绘示的是所叙述的流水线图。

图 2 绘示的是插入编码序列过程的概略图。

具体实施方式

图 1 绘示的是显示如一个例子的流水线的程序执行的流程图，从提取级 1，经由解码级 2，到执行级 3，并且从该处到写回级 4。这里的写回级 4 包含至少一个当作临时寄存器的第一寄存器 41，以及一个当作写回寄存器的第二寄存器 42。临时寄存器是一种使用时不会导致 CPU 任何状态改变的寄存器，而写回寄存器的使用会导致 CPU 的状态改变。为了提升 CPU 的安全，在流水线中转移的程序代码中，可藉由解码级 2，植入编码序列，实际的理论上可为任意的编码序列，也可以在程序代码的许多点，插入特定额外的编码序列，当作保持位置码序列或虚设码序列。这概略绘示于图 2 中。

图 2 概略地绘示任意程序的编码序列 5。在此编码序列 5 中，随机选择码序列 6(虚设序列)会插入于各种不同定义的位置，或者也会插入于随机选择的位置，而产生扩充码序列 50。例如，插入的编码序列可自内存读取，特别是自只读存储器(ROM)。

可产生插入编码序列的个别命令，例如，藉由呼叫随机数(random-number)产生器所产生的地址，插入的编码序列可自内存读取，并且会以随机的长度及次序转移到解码器。解码器会将这些虚设码序列的码植入于执行的程序代码(编码流(stream))中。甚至在程序代码中所植入的随机选择码的地址，可使用此技术中所熟知的随机方法来决定。

CPU 的状态不会因为随机插入的编码序列而改变，也不会因为随机选择及插入的复数个编码序列而改变，其只会当作保持位置序列或虚设码序列。此方法的主要优点是，每次执行相同程序的实际程序代码的执行时间，可以依照有关先前执行的需要而改变，因而使企图以统计分析(如在序言中所提及的 DPA)为主的攻击变得相当困难。

参考清单

1 提取级

2 解码级

3 执行级

4 写回级

5 码序列

6 随机选择码序列

41 第一寄存器(临时寄存器)

42 第二寄存器(写回寄存器)

50 扩充码序列

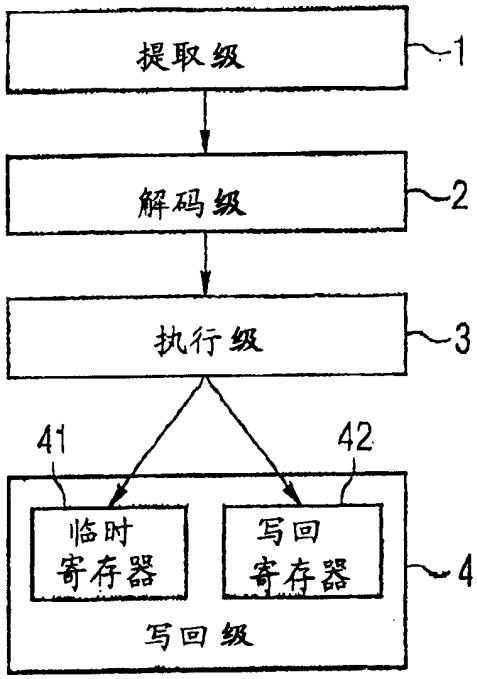


图 1

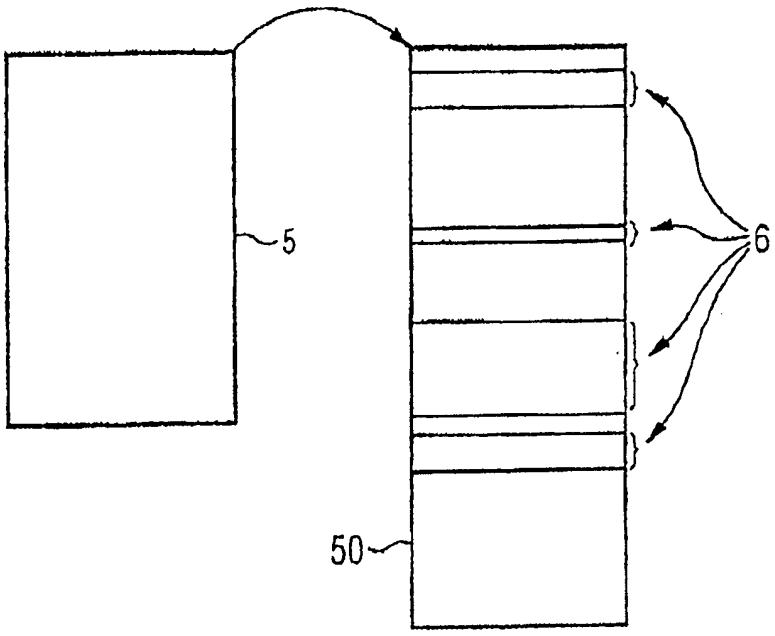


图 2