(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102063586 A
(43)申请公布日 2011.05.18

(21)申请号 201010552883.9

(22)申请日 2010.11.17

(30)优先权数据
0958141 2009.11.18 FR

(71)申请人 意法半导体(鲁塞)公司
地址 法国鲁塞

(72)发明人 雅尼克·特戈利亚

(74)专利代理机构 北京同达信恒知识产权代理有限公司 11291
代理人 黄志华 钟锦舜

(51)Int.Cl.
G06F 21/00(2006.01)
G06K 19/07(2006.01)
G06K 7/00(2006.01)

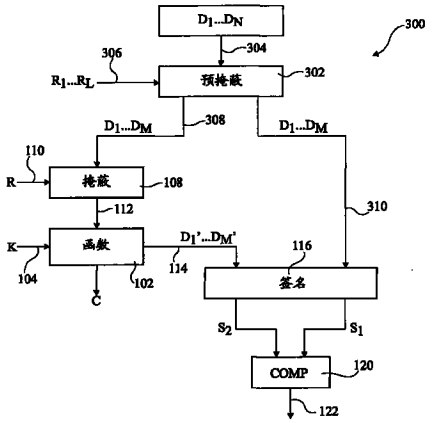
权利要求书 2 页 说明书 6 页 附图 4 页

(54)发明名称

用于检测故障攻击的方法和装置

(57)摘要

本发明涉及一种检测故障攻击的方法，该方法包括：提供多个掩蔽值(R_1至R_L)；生成第一数据元素集合($D_1...D_M$)，该第一数据元素集合包括第一组数据元素($D_1...D_N$)和通过在第一组中的至少一个数据元素与所述掩蔽值中的至少一个掩蔽值之间进行异或而生成的至少一个附加数据元素；生成与在第一数据元素集合的每一个数据元素与所述多个掩蔽值中的所选掩蔽值之间的异或对应的第二数据元素集合($D_1'...D_M'$)；通过在第一数据元素集合的各数据元素之间进行交换运算来生成第一签名(S_1)；通过在第二数据元素集合的各数据元素之间进行所述交换运算来生成第二签名；以及比较第一签名与第二签名以检测故障攻击。



CN 102063586 A

1. 一种检测故障攻击的方法，包括：

提供多个掩蔽值（R_1 至 R_L）；

生成第一数据元素集合（$D_1 \ldots D_M$），所述第一数据元素集合包括第一组数据元素（D_1 至 D_N）和至少一个附加数据元素，所述至少一个附加数据元素通过在所述第一组数据元素中的至少一个数据元素与所述掩蔽值中的至少一个掩蔽值之间进行异或而生成；

生成与在所述第一数据元素集合的每一个数据元素与所述多个掩蔽值中的所选掩蔽值（R）之间的异或对应的第二数据元素集合（$D_1' \ldots D_M'$）；

通过在所述第一数据元素集合的各数据元素之间进行交换运算来生成第一签名（S_1）；

通过在所述第二数据元素集合的各数据元素之间进行所述交换运算来生成第二签名（S_2）；以及

比较所述第一签名与第二签名以检测故障攻击。

2. 根据权利要求 1 所述的方法，其中，生成所述至少一个附加数据元素包括：通过在所述第一组数据元素（A'）的每一个数据元素与所述多个掩蔽值中的每一个掩蔽值之间进行异或来生成附加的多组数据元素；并且选择至少一个附加组的至少一个数据值以形成所述至少一个附加数据元素。

3. 根据权利要求 2 所述的方法，其中，生成所述附加的至少一组数据元素还包括在所述第一组数据元素的每一个数据元素与所述掩蔽值中的两个掩蔽值的每个组合之间进行异或。

4. 根据权利要求 1 所述的方法，其中，通过在所述第一数据元素集合的每个数据元素与所述多个掩蔽值中的所选掩蔽值（R）之间进行异或来掩蔽所述第一数据元素集合的每个数据元素，从而生成所述第二数据元素集合（$D_1' \ldots D_M'$）。

5. 根据权利要求 1 所述的方法，其中，通过在所述第一组的每个数据元素与所述多个掩蔽值中的所选掩蔽值（R）之间进行异或来掩蔽所述第一组数据元素（$D_1 \ldots D_N$）以生成被掩蔽的数据元素组（$D_1' \ldots D_N'$），并且在所述被掩蔽的数据元素组的至少一个数据元素与所述掩蔽值中的所述至少一个掩蔽值之间进行异或，从而生成所述第二数据元素集合（$D_1' \ldots D_M'$）。

6. 根据权利要求 1 所述的方法，其中，对于每一被掩蔽的组中的每一个数据值，选择所述至少一个附加数据元素包括：

确定所述数据值是否已经存在于所述第二数据元素集合中；以及

如果所述数据值未存在于所述第二数据元素集合中，则将所述数据值添加到所述第二数据元素集合中。

7. 根据权利要求 6 所述的方法，其中，选择所述至少一个附加数据元素还包括：

将添加到所述第二数据元素集合中的各数据值标记为不可用；以及

如果所述数据值已经存在于所述第二数据元素集合中，则确定所述数据值是否被标记为不可用，若否，则将所述数据值标记为不可用，而若是，则将所述数据值添加到所述第二数据元素集合中。

8. 根据权利要求 1 所述的方法，其中，生成所述第一签名和第二签名包括在对应的数据元素集合的各数据元素之间进行异或。

9. 根据权利要求 1 所述的方法，其中，生成所述第一签名和第二签名包括将对应的数

据元素集合的各数据元素全部相加。

10. 根据权利要求 1 所述的方法，其中，生成所述第一签名和第二签名包括将对应的数据元素集合的各数据元素全部相乘。

11. 一种用于检测故障攻击的电路，包括：

存储器，存储多个掩蔽值（R_1 至 R_L）；

计算块（302，108，102），被布置成：生成第一数据元素集合（$D_1 \ldots D_M$），所述第一数据元素集合包括第一组数据元素（D_1 至 D_N）和至少一个附加数据元素，所述至少一个附加数据元素通过在所述第一组中的至少一个数据元素与所述掩蔽值中的至少一个掩蔽值之间进行异或而生成；并且生成与在所述第一数据元素集合的每一个数据元素与所述多个掩蔽值中的所选掩蔽值（R）之间的异或对应的第二数据元素集合（$D_1' \ldots D_M'$）；

签名块（116），被布置成通过在所述第一数据元素集合的各数据元素之间进行交换运算来生成第一签名（S_1）并且通过在所述第二数据元素集合的各数据元素之间进行所述交换运算来生成第二签名（S_2）；以及

比较器（120），被布置成比较所述第一签名与第二签名以检测故障攻击。

12. 根据权利要求 11 所述的电路，还包括：函数单元（102），被布置成基于所述第一数据元素集合的至少一个数据元素来进行运算，其中，所述函数单元被布置成基于密钥（K）进行加密或者解密运算。

13. 一种集成电路，包括根据权利要求 11 所述的电路。

14. 一种集成电路卡，包括根据权利要求 13 所述的集成电路。

15. 一种集成电路读卡器，包括根据权利要求 13 所述的集成电路。

用于检测故障攻击的方法和装置

技术领域

[0001] 本发明涉及一种用于检测故障攻击的方法和装置，并且特别涉及一种用于基于签名来提供检测的方法和装置。

背景技术

[0002] 集成电路可以包括鉴于所操纵的诸如认证密钥、签名等数据的安全性或者鉴于所使用的算法如加密算法或解密算法而被视为敏感的电路。这样的信息需要保密，这意味着该信息不应被传达给第三方或未授权电路，或者可由第三方或未授权电路检测出。

[0003] 用于剽窃由集成电路操纵的信息的常用过程包括检测该电路中在处理该信息期间使用的分区。为此，在功能环境中激活该电路或者将电路置于功能环境中，并且在输入端引入待编码的数据包。当进行处理数据时，由激光扫射集成电路的表面以在电路的运行中注入故障。通过并行分析电路的输出，这使得能够确定电路中处理该数据的分区。将这些分区局部化后，剽窃者能够对这些分区集中攻击，以确定正被处理的秘密数据。

[0004] 签名提供一种保护电路免受故障攻击的方式。签名基于将由算法使用的一个或者多个数据值来生成。然后在相同的数据值经过算法使用之后，对这些数据值生成签名。这两个签名的差异表明出现攻击。检测电路一旦检测到这样的攻击，可以触发对策，比如重置电路，和 / 或将计数器递增，这使得在检测到一定数量的故障时使集成电路持久地不工作。

[0005] 旁路攻击是包括例如测量电路功耗的一种不同类型的攻击。掩蔽提供一种保护电路免受旁路攻击的方式。掩蔽包括以无损方式更改使用伪随机变量的算法的输入。

[0006] 希望提供一种电路，其中，同组数据值可以用来生成签名以检测故障攻击并且可以被掩蔽以使旁路攻击更为困难。然而，难以实现结合这些功能的有效的电路。

发明内容

[0007] 本发明的一个目的在于至少部分地解决现有技术中的一个或者多个问题。

[0008] 根据本发明的一个方面，提供一种检测故障攻击的方法，该方法包括：提供多个掩蔽值；生成第一数据元素集合，该第一数据元素集合包括第一组数据元素和通过在第一组中的至少一个数据元素与所述掩蔽值中的至少一个掩蔽值之间进行异或而生成的至少一个附加数据元素；生成与在第一数据元素集合的每一个数据元素与所述多个掩蔽值中的所选掩蔽值之间的异或对应的第二数据元素集合；通过在第一数据元素集合的各数据元素之间进行交换运算来生成第一签名；通过在第二数据元素集合的各数据元素之间进行所述交换运算来生成第二签名；以及比较第一签名与第二签名以检测故障攻击。

[0009] 根据一个实施例，生成所述至少一个附加数据元素包括：通过在第一组数据元素的每一个数据元素与所述多个掩蔽值中的每一个掩蔽值之间进行异或来生成附加的多组数据元素；并且选择至少一个附加的组的至少一个数据值以形成至少一个附加数据元素。

[0010] 根据另一实施例，生成附加的至少一组数据元素还包括在第一组数据元素的每一个数据元素与所述掩蔽值中的两个掩蔽值的每个组合之间进行异或。

[0011] 根据另一实施例，通过在第一数据元素集合的每一个数据元素与多个掩蔽值中的所选掩蔽值之间进行异或来掩蔽所述第一数据元素集合的各数据元素从而生成第二数据元素集合。

[0012] 根据另一实施例，通过在第一组的每一个数据元素与多个掩蔽值中的所选掩蔽值之间进行异或来掩蔽所述第一组数据值以生成被掩蔽的数据元素组并且在被掩蔽的数据元素组的至少一个数据元素与所述掩蔽值中的至少一个掩蔽值之间进行异或从而生成第二数据元素集合。

[0013] 根据另一实施例，对于每一被掩蔽的组中的每个数据值，选择至少一个附加数据元素包括：确定所述数据值是否已经存在于第二数据元素集合中；以及如果所述数据值未存在于第二数据元素集合中，则将所述数据值添加到第二数据元素集合中。

[0014] 根据另一实施例，选择至少一个附加数据元素还包括：将添加到第二数据元素集合中的各数据值标记为不可用；以及如果所述数据值已经存在于第二数据元素集合中，则确定所述数据值是否标记为不可用，若否，则将所述数据值标记为不可用，而若是，则将所述数据值添加到第二数据元素集合中。

[0015] 根据另一实施例，生成所述第一签名和第二签名包括在对应的数据元素集合的各数据元素之间进行异或。

[0016] 根据另一实施例，生成所述第一签名和第二签名包括将对应的数据元素集合的各数据元素全部相加。

[0017] 根据另一实施例，生成所述第一签名和第二签名包括将对应的数据元素集合的各数据元素全部相乘。

[0018] 根据本发明的另一方面，提供一种用于检测故障攻击的电路，该电路包括：存储器，存储多个掩蔽值；计算块，被布置成生成第一数据元素集合，该第一数据元素集合包括第一组数据元素和通过在第一组中的至少一个数据元素与所述掩蔽值中的至少一个掩蔽值之间进行异或而生成的至少一个附加数据元素，并且生成与在第一数据元素集合的每个数据元素与所述多个掩蔽值中的所选掩蔽值之间的异或对应的第二数据元素集合；签名块，被布置成通过在第一数据元素集合的各数据元素之间进行交换运算来生成第一签名并且通过在第二数据元素集合的各数据元素之间进行交换运算来生成第二签名；以及比较器，被布置成比较第一签名与第二签名以检测故障攻击。

[0019] 根据一个实施例，该电路还包括：函数单元，被布置成基于第一数据元素集合的至少一个数据值来进行运算，其中，所述函数块被布置成基于密钥来进行加密或者解密运算。

[0020] 根据本发明的另一方面，提供一种包括上述电路的集成电路、IC（集成电路）卡或者 IC 读卡器。

附图说明

[0021] 根据以下参照附图通过图示方式而非限制方式给出的对实施例的详细描述，容易理解本发明的前述和其它目的、特征、方面及优点，附图中：

[0022] 图 1 图示了根据一个实施例的用于检测故障攻击并且防范旁路攻击的电路；

[0023] 图 2 图示了图 1 的电路所使用的数据值；

[0024] 图 3 图示了根据本发明的实施例的用于检测故障攻击并且防范旁路攻击的电路；

[0025] 图 4 图示了图 3 的电路所使用的数据的示例；

[0026] 图 5 图示了根据本发明的另一实施例的用于检测故障攻击并且防范旁路攻击的电路；

[0027] 图 6 图示了图 5 的电路所使用的数据的示例；

[0028] 图 7 图示了根据本发明的实施例的用于生成数据集合的方法中的步骤；以及

[0029] 图 8 图示了根据本发明的实施例的包括用于检测故障攻击的电路的电子设备。

具体实施方式

[0030] 为求清楚，只有那些对理解本发明有用的步骤和元件在图中示出并且在下面详细描述。特别地，未详细描述用于在检测到一个或者多个故障注入时将集成电路重置或使其不工作的电路，本发明适用于任何这样的电路。另外，未详细描述受保护的集成电路的主要功能，本发明可与实现任何敏感功能（比如加密或者解密、或者涉及到敏感数据的其它功能）的集成电路兼容。

[0031] 图 1 图示了包括函数单元 (FUNCTION，函数）102 的电路 100，该函数单元 102 例如实施涉及诸如加密密钥等敏感数据的算法。在本示例中，该单元 102 通过输入线 104 接收密钥 K。

[0032] 通过线路 106 向掩蔽块 (BLINDING，掩蔽）108 提供一组数据值 D_1 至 D_N，该掩蔽块 108 基于通过输入线 110 向掩蔽块 108 提供的掩蔽值 R 对这些数据值应用掩蔽算法。掩蔽值 R 例如是伪随机值。然后通过线路 112 向函数单元 102 提供被掩蔽的数据值 D_1' 至 D_N'。函数单元 102 实施使用被掩蔽的数据值 D_1' 至 D_N' 的算法并且输出结果 C，该结果 C 可以是加密的或解密的数据块或者其它值。

[0033] 在由函数单元 102 进行使用时，被掩蔽的数据值 D_1' 至 D_N' 例如存储于寄存器（图 1 中未示出）中。在由函数块 102 执行算法期间的各个阶段和 / 或在这一执行结束时，通过线路 114 向签名块 (SIG)116 输出被掩蔽的数据值 D_1' 至 D_N'。通过线路 118 还向签名块 116 提供原数据值 D_1 至 D_N。

[0034] 签名块 116 基于原数据值 D_1 至 D_N 生成签名 S_1 并基于被掩蔽的数据值 D_1' 至 D_N' 生成签名 S_2。然后比较器 (COMP)120 比较签名 S_1 与 S_2，该比较器 120 通过输出线 122 提供表明签名是否匹配的输出。

[0035] 块 108 所采用的掩蔽算法例如是 XOR 函数，在掩蔽值 R 与所述数据值 D_1 至 D_N 中的每一个之间采用。然而，问题在于：选择待应用于这两组数据值的签名函数以在不存在故障攻击时得到相同的结果。例如，图 2 图示了这一问题的示例。

[0036] 图 2 在第一行 200 中针对 N 等于 9 的情况图示了数据值 D_1 至 D_N 的示例，并且值 D_1 至 D_9 分别等于 12、1、0、128、245、0、1、2 和 8。假设用于确定签名的函数是将所有数据值相加，则 D_1 至 D_9 的签名等于 397。图 2 中的第二行 202 图示了在这一示例中通过在数据值 D_1 至 D_9 中的每一个与值 01 之间应用 XOR 运算而获得的被掩蔽的数据值 D_1' 至 D_9'。因此，值 D_1' 至 D_9' 分别等于 13、2、1、129、244、1、0、3 和 9。然而，假设无故障，数据值 D_1' 至 D_9' 之和等于 402，因而签名 S_2 等于 402。因此，尽管无故障，但是签名方案由于签名 S_1 和 S_2 的不匹配而无效。

[0037] 图 3 图示了用于检测故障攻击和用于防范旁路攻击的电路 300。电路 300 包括多

个与图 1 的电路相同的元件，并且不再详细描述这些元件。特别地，这一电路包括函数单元 102、掩蔽块 108、签名块 116 和比较器 120。

[0038] 电路 300 还包括预掩蔽块 (PRE-BLINDING，预掩蔽)302，该预掩蔽块 302 通过输入线 304 接收数据值 D_1 至 D_N 并通过输入线 306 接收多个掩蔽值 R_1 至 R_L。预掩蔽块 302 生成数据值 D_1 至 D_M 的数据集合。该集合 D_1 至 D_M 包括数据值 D_1 至 D_N，以及一个附加数据值 D_M 或者多个附加数据值 D_{N+1} 至 D_M。数据值 D_1 至 D_M 通过线路 308 提供给掩蔽块 108 且通过线路 310 提供给签名块 116。

[0039] 生成集合 D_1 至 D_M，使得由块 108 来应用 XOR 掩蔽函数时，结果是数据值 D_1 至 D_M 进行排列而不引入新值，因此签名块 116 能够使用任何交换函数来生成有效签名。这例如按照下文中参照图 4 更详细描述的那样来实现。

[0040] 图 4 图示了包含数据集合 D_1 至 D_M 的数据值的示例的表。该表的第一行示出数据值 D_1 至 D_N，这些数据值在这一情况下包括分别等于十进制值 12、1、0、128、245、0、1、2 和 8 的等效二进制值的九个值 D_1 至 D_9。这些数据值称为组 A。在这一示例中，假设掩蔽值 R_1 至 R_L 包括两个分别等于十进制值 01 和 02 的等效二进制值的值 R_1 和 R_2。

[0041] 预掩蔽块 302 生成新的一组数据值“A+1”(符号“+”在这里表示函数 XOR)，其等于数据组 A 的每一个值与第一掩蔽值 01 的 XOR。这在图 4 中的表的第二行中示出，并且这些值等于 13、0、1、129、244、1、0、3 和 9。预掩蔽块还生成新的一组数据值“A+2”，其等于数据组 A 的每一个值与第二掩蔽值 02 的 XOR。这在图 4 中的表的第三行中示出，并且这些值为 14、3、2、130、247、2、3、0 和 10。

[0042] 在图 4 的示例中，还生成另一组数据值“(A+1)+2”。这等效于“A+3”并且等于数据组 A 的每一个值与值 03 的 XOR。在图 4 的第四行中示出的值为 15、2、3、131、246、3、2、1 和 11。该组并非总是会生成，因为在一些实施例中仅生成与各掩蔽值 R_1 至 R_L 对应的数据值集合。

[0043] 例如，还生成与掩蔽值 R_1 至 R_L 中的两个掩蔽值的每一可能组合对应的行，除非该组合等效于已经存在的行的掩蔽值。在图 4 的示例中，(A+1)+2 ＝ A+3 不与先前生成的任一行等效。然而，如果还使用附加掩蔽值 R_2 ＝ 03，则因为 (A+1)+3 等效于 A+2、(A+1)+2 等效于 A+3，而 (A+2)+3 等效于 A+1，所以仅生成行 A+1、A+2 和 A+3。

[0044] 图 4 的表因此表示分别通过线路 308 和 310 向掩蔽块 108 和签名块 116 提供的值 D_1 至 D_M。当对该表应用掩蔽时，结果仅为对行进行排列而不添加任何新值。特别地，如果例如从值 R_1 和 R_2 中伪随机选择的掩蔽值 R 等于 01，则第一行会变成 A+1，而第二行变成等于 A 的 (A+1)+1。类似地，第三行变成等于 A+3 的 (A+2)+1，而第四行变成等于 A+2 的 (A+3)+1。

[0045] 函数单元 102 使用数据值 D_1 至 D_N 的被掩蔽的版本。在图 4 的示例中，取决于掩蔽值是 01 还是 02，这些被掩蔽的值是第二行或者第三行的被掩蔽的值。

[0046] 图 5 图示了用于检测故障攻击和用于防范旁路攻击的电路 500。

[0047] 电路 500 包括多个与图 1 的电路相同的元件，并且不再详细描述这些元件。相对于图 1 的实施例，图 5 的实施例还包括用于向附加预签名块 504 提供数据值 D_1 至 D_N 的线路 502，以及位于函数单元 102 与签名块 116 之间用于在签名生成之前处理被掩蔽的数据值的附加预签名块 506。现在参照图 6 描述图 5 的电路的操作。

[0048] 图 6 图示了在应用于图 5 的电路时与图 4 相同的数值示例。预签名块 504 生成与

图 4 中生成的表相同的值 D_1 至 D_M 的表（不再图示），这不再详细描述。另一方面，预签名块 506 基于被掩蔽的值 D_1' 至 D_N' 生成新的表。该表的行以与图 4 的表相同的方式生成，即通过系统地应用各个掩蔽值来生成。在图 6 的示例中，假设通过线路 110 接收的掩蔽值为 02，因此值 D_1' 至 D_N' 对应于所述值 D_1 至 D_N 中的每一个与值 02 的 XOR。因此，图 6 的表中被标记为 A' 的第一行等于 A+2，其与图 4 中的表的第三行相同。图 6 中的表的第二行等于与 A+3（即图 4 的表的第四行）等效的 A'+1。图 6 的表的第三行等于与 A（即图 4 的表的第一行）等效的 A'+2。最后，图 6 的表的第四行等于与 A+1（即图 4 中的表的第二行）等效的 (A'+1)+2。

[0049] 因此，已经指出，图 6 的表的值为图 4 的表的值的排列，因此签名块 116 可以通过对两个数据值集合应用交换签名算法来有效地检测故障。

[0050] 与图 3 的实施例相比时，图 5 的实施例的一个优点在于它使得多故障攻击的使用更加困难。特别地，尽管以图 3 为目标的多故障攻击可能攻击预掩蔽块 302 和函数块 102，但是在图 5 中，这样的攻击除了将函数块 102 作为目标之外还必须将预签名块 504 和 506 二者作为目标。

[0051] 图 3 或者图 5 的块 116 所采用的签名为交换函数，使得数据值的顺序不会影响结果。这样的函数的示例为在各个值 D_1' 至 D_M' 之间应用的 XOR 函数、或者所有值 D_1' 至 D_M' 之和、或者所有值 D_1' 至 D_M' 之积。可替选地，可以使用这些函数中的一个或者多个函数的组合或者变型。

[0052] 可以通过去除重复的值来减少集合 D_1 至 D_M 中存在的其它值 D_{N+1} 至 D_M 的数量。例如，在图 4 的示例中，可以去除第二行中的值“0”和“1”而完全不会降低签名比较的有效性。现在参照图 7 的流程图描述用于确定哪些数据值可以被添加到集合 D_1 至 D_M 中的初始值 D_1 至 D_N 的技术的示例。

[0053] 图 7 示出一种用于生成数据值 D_{N+1} 至 D_M 的方法。在该示例中，D_n 表示组 D_1 至 D_N 中的第 n 个值，而 R_p 表示值 R_1 至 R_p 的集合的第 p 个值，该集合中前 L 个值 R_1 至 R_L 是从中选出 R 的掩蔽值集合，而值 R_{L+1} 至 R_P 是附加组合，其等于以下 XOR 计算的结果：$R_1+R_{(L+1)}$，$R_1+R_{(L+2)}$，... R_1+R_p，$R_2+R_{(L+1)}$，$R_2+R_{(L+2)}$，... R_2+R_p，... $R_{(L-1)}+R_L$。可以去除集合 R_1 至 R_p 中的任何重复值，使得每个值仅出现一次，由此允许减少处理时间。字母 S 用来表示值 D_1 至 D_M 的集合，该集合起初仅包括值 D_1 至 D_N。

[0054] 在该方法的第一步骤 S1 中，将变量 n 和 p 均设置为等于 1。

[0055] 接着，在步骤 S2 中，将变量 Q 设置为等于 D_n XOR R_p。起初，这将等于 D_1 与第一掩蔽值 R_1 的 XOR。

[0056] 在下一步骤 S3 中，确定 Q 是否为集合 S 的元素，即，该值是否存在于值 D_1 至 D_N 之中。若否，则下一步骤为 S4，在步骤为 S4 中，将 Q 添加到集合 S 中，并且将标签与值 Q 关联从而表明该值不可用。该标签可以例如是存储器中与该值关联的标记或者表明该值不可用的任何其它手段。

[0057] 如果在步骤 S3 中确定 Q 为 S 的元素，则在步骤 S5 中检验 S 中的该数据值是否可用。换而言之，检验该值是否已被标记为不可用。如果该值不可用，则下一步骤为步骤 S4，如上文所述，在步骤 S4 中，将 Q 添加到集合 S 中并且被标记为不可用。然而，如果在步骤 S5 中确定 S 中的该数据值尚未被标记为不可用，则下一步骤为 S6，在步骤 S6 中，不向集合 S 添

加数据值 Q，而是将 S 中的该值标记为不可用。

[0058] 在步骤 S4 和 S6 之后，下一步骤为 S7，在步骤 S7 中，确定 p 是否等于 P。若否，则在步骤 S8 中递增 p，并且该方法返回到步骤 S2。然而，如果在步骤 S7 中确定 p 等于 P，则下一步骤为 S9。

[0059] 在步骤 S9 中确定 n 是否等于 N。若否，则在步骤 S10 中递增 n，并且该方法返回到步骤 S2。然而，如果 n 等于 N，则该过程结束。

[0060] 图 8 图示了电子设备 800，该电子设备 800 包括微处理器 802、存储器块 804 和向微处理器 802 提供输入值的输入线 806。微处理器 802 通过输出线 808 提供输出值。另外，设置有故障检测电路 810，与微处理器的输出耦合，该故障检测电路 810 例如包括如本文所述的预掩蔽块 302 或者预签名块 504 和 506、签名块 116 和比较器 120。如果由于签名之间的不匹配而检测到故障攻击，则该电路 810 通过环回至微处理器 802 的输出线 812 提供报警信号。该报警信号例如触发微处理器 802 重置和 / 或使计数器（图 8 中未示出）递增，一旦达到某个计数值，该计数器使微处理器持久地停用。

[0061] 电子设备 800 例如是 IC（集成电路）卡（比如智能卡）、IC 读卡器（比如信用卡支付终端）、或者机顶盒、用于 PC 或者膝上型计算机的硬盘驱动、PC 或者膝上型计算机、自动售货机或者其他处理敏感信息的设备。

[0062] 这里描述的实施例的一个优点在于：通过生成如这里描述的数据值 D_1 至 D_M 的集合用于生成签名，使用 XOR 运算来掩蔽这些值的结果将获得这些值的排列。这造成可以广泛选择用来产生用于检测故障攻击的可比较的签名的签名算法。

[0063] 尽管已经描述了本发明的多个具体实施例，但是本领域技术人员清楚可以应用许多变型和改变。

[0064] 例如，本领域技术人员清楚这里描述的实施例可以应用于其中使用签名变化来检测故障的各种电路。

[0065] 另外，本领域技术人员应当清楚，尽管已经描述了一些实施例，其中放大的集合 D_1'...D_M' 包括许多附加值 D_{N+1}' 至 D_M'，但是在一些情况下仅提供一个或者少数几个附加值。

[0066] 另外，可以用软件、硬件或者其组合来实现这里描述的实施例。此外，可以在替选实施例中以任何组合来组合关于各个实施例描述的特征。

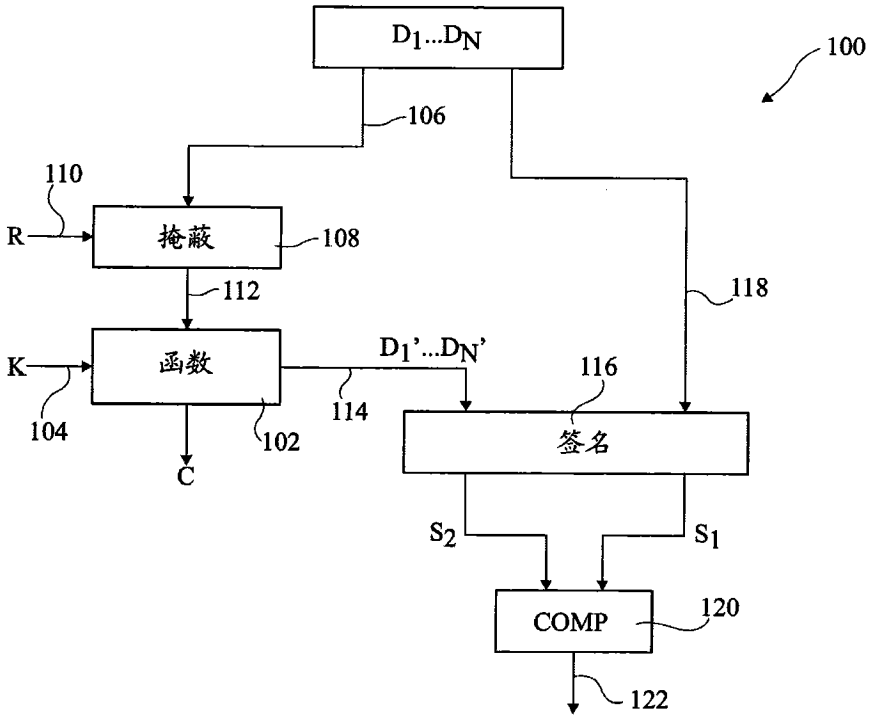

D1...DN
100
106
110
R
掩蔽
108
112
118
K
函数
D1'...DN'
116
104
114
102
C
签名
S2
S1
COMP
120
122

图 1

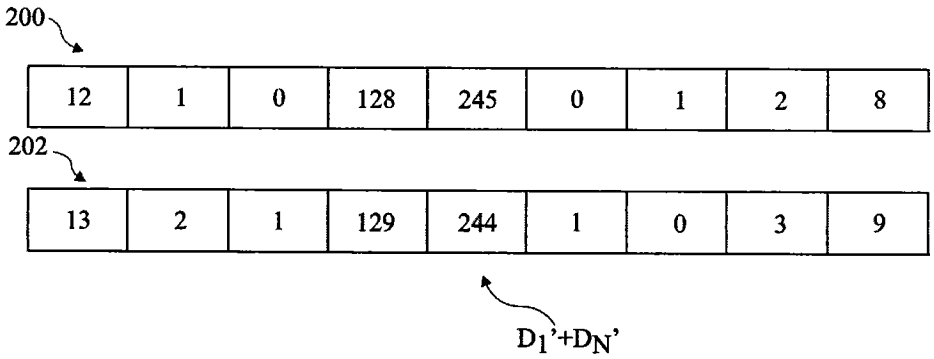

200
12
1
0
128
245
0
1
2
8
202
13
2
1
129
244
1
0
3
9
D1'+DN'

图 2

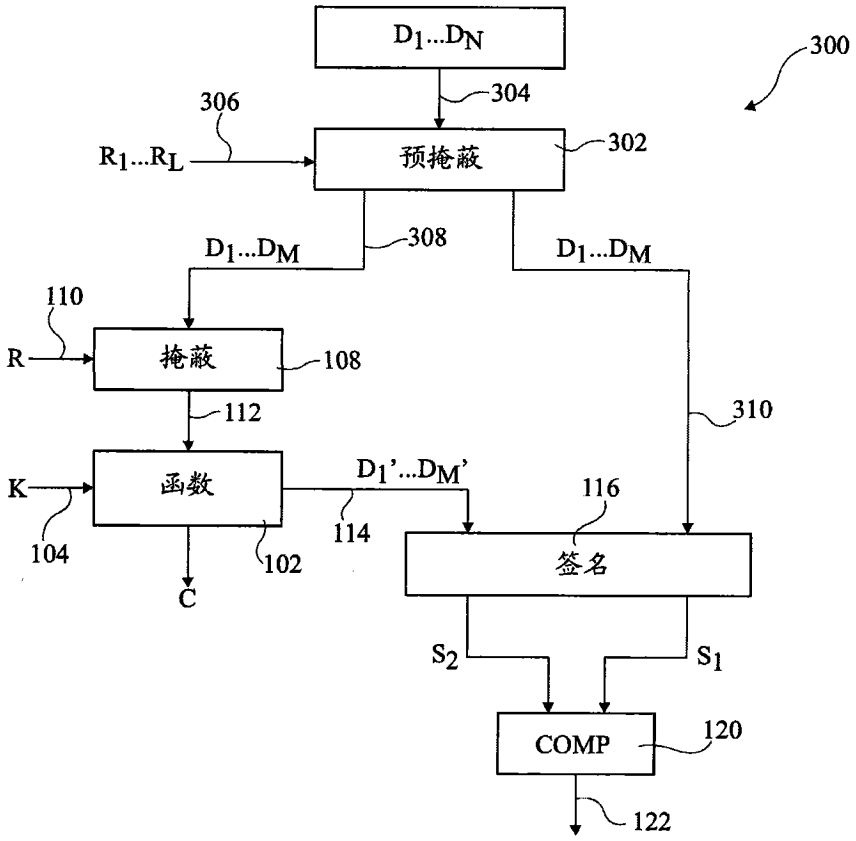


图 3

D1 至 DM

A	12	1	0	128	245	0	1	2	8
A+1	13	0	1	129	244	1	0	3	9
A+2	14	3	2	130	247	2	3	0	10
(A+1)+2=A+3	15	2	3	131	246	3	2	1	11

图 4

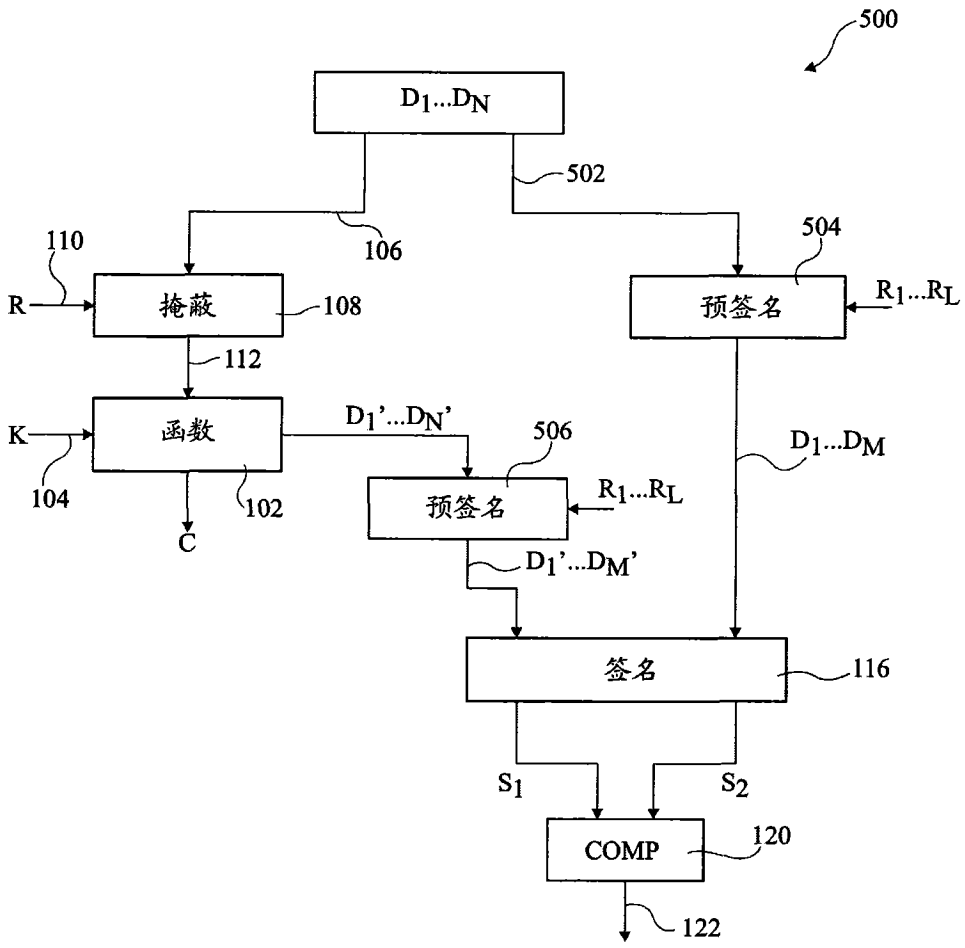


图 5

A'=A+2	14	3	2	130	247	2	3	0	10
A'+1=A+3	15	2	3	131	246	3	2	1	11
A'+2=A	12	1	0	128	245	0	1	2	8
(A'+1)+2=A+1	13	0	1	129	244	1	0	3	9

图 6

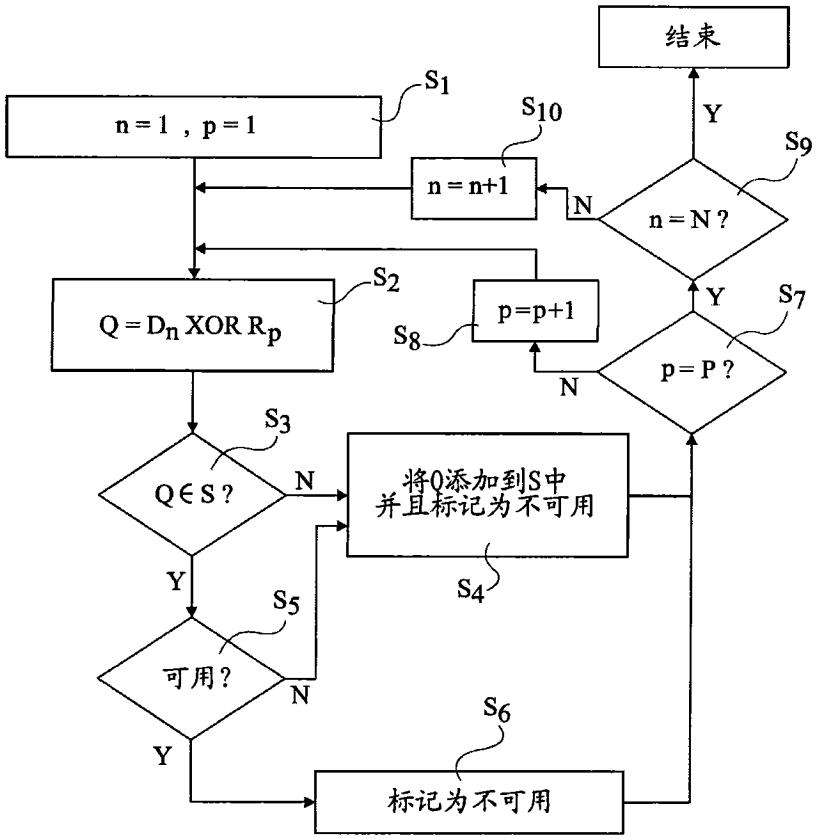

结束
S1
n = 1 , p = 1
S10
n = n+1
N
S9
n = N ?
Y
S2
Q = Dn XOR Rp
S8
p=p+1
N
Y
S7
p = P ?
S3
Q∈S ?
N
将0添加到S中
并且标记为不可用
S4
Y
S5
可用?
N
Y
S6
标记为不可用

图 7

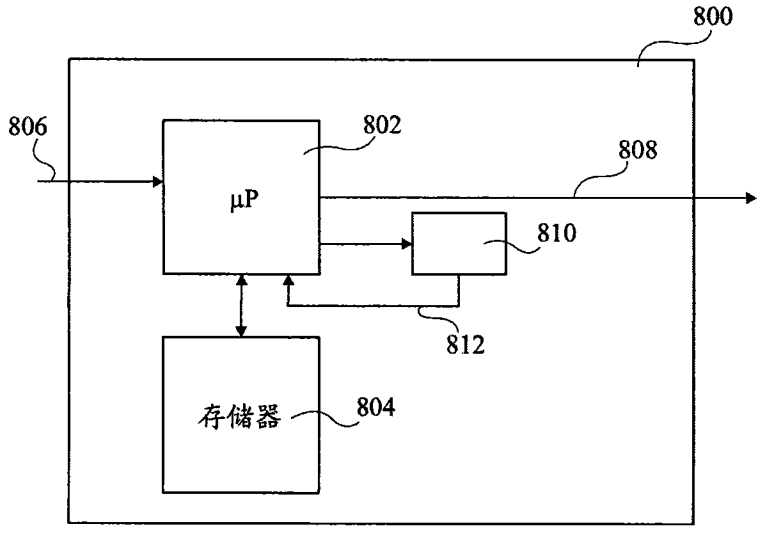

800
806
802
μP
808
810
812
存储器
804

图 8