(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 102725737 B
(45)授权公告日 2016.04.20

(21)申请号 201080060319.3

(22)申请日 2010.12.02

(30)优先权数据
61/266,948 2009.12.04 US
12/958,570 2010.12.02 US

(85)PCT国际申请进入国家阶段日
2012.07.04

(86)PCT国际申请的申请数据
PCT/US2010/058768 2010.12.02

(87)PCT国际申请的公布数据
WO2011/068996 EN 2011.06.09

(73)专利权人 密码研究公司
地址 美国加利福尼亚州

(72)发明人 P·C·科彻 P·罗哈吉
J·M·雅菲

(74)专利代理机构 北京市金杜律师事务所 11256
代理人 王茂华

(51)Int.Cl.
G06F 11/00(2006.01)

(56)对比文件
US 2009/0070583 A1,2009.03.12,说明书第[0009]-[0010],[0102],[0290],[0215],[0490]-[0496]段.
US 2008/0263363 A1,2008.10.23,参见说明书第[0020],[0039][0054]-[0056],[0069],[0091],[0112],[0143]-[0148],[0154],[0164],[0173],[0185],[0222]-[0290]段.

审查员 张晓芳

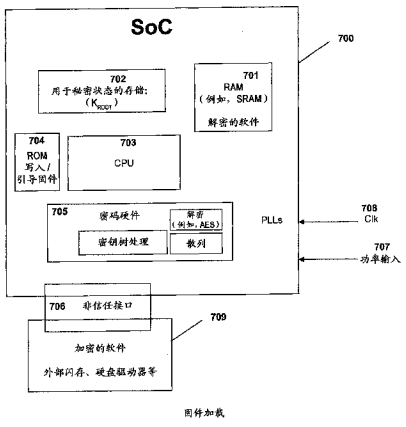
权利要求书5页 说明书20页 附图14页

(54)发明名称

可验证防泄漏的加密和解密

(57)摘要

公开了用于以提供安全性以防外部监视攻击的方式加密和解密敏感数据的方法和设备。加密设备具有对解密设备也已知的基本秘密密码值(密钥)的访问权。将敏感数据分解成段,并且用根据基本密钥和消息标识符导出的单独加密密钥加密每段,以创建一组加密段。加密设备使用基本密码值以创建如下验证器,这些验证器证实用于这一消息标识符的加密段由具有对基本密钥的访问权的设备创建。解密设备响应于接收加密段使用验证器以验证未修改消息标识符和加密段。



CN 102725737 B

1. 一种用于由设备用内部秘密状态加密明文数据而又限制重用密码密钥的方法，包括：

(a) 通过计算从所述内部秘密状态的至少部分开始并且促成消息密钥的多个相继中间密钥来根据所述内部秘密状态和消息标识符导出所述消息密钥，其中至少基于所述消息标识符的部分和先前密钥来导出每个相继中间密钥；

(b) 使用至少基于所述消息密钥的所述一个或者多个密码密钥以加密所述明文数据的一个或者多个段以产生一个或者多个加密的数据段；

(c) 使用秘密密钥以计算至少基于所述加密的数据段中的一个或者多个加密的数据段并且可用于验证所述加密的数据段中的一个或者多个加密的数据段的密码验证值；以及

(d) 输出所述一个或者多个加密的数据段和所述密码验证值；

其中所述步骤 (c) 包括：(i) 根据至少一个所述加密的数据段计算密码散列；以及 (ii) 根据至少秘密值和所述密码散列导出验证器，其中所述导出包括计算始于所述秘密值的多个相继中间值，其中每个相继中间值至少基于所述相继中间值中的先前值和所述密码散列的部分。

2. 根据权利要求 1 所述的方法，其中在 (a) 中所述导出每个所述相继中间密钥包括选择熵分布操作，然后将所述熵分布操作应用于所述先前密钥。

3. 根据权利要求 2 所述的方法，其中将所述消息标识符分解成多个部分，每个所述部分确定待应用的特定熵分布操作。

4. 根据权利要求 3 所述的方法，其中所述熵分布操作是至少依赖于以下各项的密码散列操作：(i) 所述先前密钥；以及 (ii) 对应于所述消息标识符的至少部分的值。

5. 根据权利要求 1 所述的方法，其中使用密钥链接根据所述消息密钥导出 (b) 中的所述密码密钥。

6. 根据权利要求 1 所述的方法，其中通过遵循经过密钥树的路径来计算 (a) 中的所述消息密钥，所述路径包括所述消息标识符。

7. 根据权利要求 6 所述的方法，其中通过遵循经过密钥树的路径来计算所述验证器，所述路径包括基于所述加密的数据段的散列。

8. 根据权利要求 1 所述的方法，其中 (b) 包括通过执行多个熵分布操作根据所述消息密钥导出多个所述密码密钥；并且进一步：(i) 通过将熵分布函数应用于所述消息密钥来计算所述密码密钥之一；以及 (ii) 通过将熵分布函数应用于多个相继密码密钥中的先前密码密钥来计算所述密码密钥中的每个密码密钥。

9. 根据权利要求 1 所述的方法，其中：(i) 所述明文数据包括多个段；(ii) 在加密仅一个明文段时使用每个所述密码密钥，从而产生加密的数据段；并且 (iii) 计算基于所有所述加密的数据段的密码散列。

10. 根据权利要求 9 所述的方法，其中使用散列链接来计算 (iii) 中的所述密码散列。

11. 根据权利要求 10 所述的方法，其中在所述散列链接中，以与输出所述加密的数据段的顺序相反的顺序散列化所述加密的数据段。

12. 根据权利要求 1 所述的方法，其中在 (b) 中：(i) 所述明文数据包括多个段，并且每段包括多个子段；并且 (ii) 对于每个明文子段，所述密码密钥中的新密码密钥用来加密所述子段。

13. 根据权利要求 12 所述的方法，还包括使用散列链接以基于所有所述加密的数据段计算密码散列。

14. 根据权利要求 1 所述的方法，其中在 (b) 中 ：i) 所述明文数据包括多个段 ；(ii) 密码密钥更新在段内基础上出现 ；并且 (iii) 通过使用所述更新的密码密钥中的至少一个密码密钥来产生每个所述加密的数据段。

15. 根据权利要求 1 所述的方法，其中每个所述相继中间值是密码散列操作的结果，所述密码散列操作的输入仅通过以下确定 ：(1) 所述相继中间值的直接父值 ；以及 (2) 所述加密的数据段散列的一位或者两位。

16. 根据权利要求 1 所述的方法，其中存在仅一个明文数据段和仅一个加密的数据段，并且其中所述密码散列计算还包括数据段长度。

17. 根据权利要求 1 所述的方法，其中随机生成所述消息标识符。

18. 根据权利要求 1 所述的方法，其中所述消息标识符为计数器。

19. 根据权利要求 1 所述的方法，其中所述设备是向解密设备传输数据分组的联网通信设备。

20. 根据权利要求 1 所述的方法，还包括在向解密设备发送加密的数据之前认证所述解密设备。

21. 根据权利要求 1 所述的方法，其中所述密码验证值是认证所述加密的数据段中的至少一个加密的数据段的数字签名。

22. 一种用于由设备用内部秘密状态解密数据而又限制重用密码密钥的方法，包括 ：

(a) 接收一个或者多个加密的数据段和密码验证值并且获得与之对应的消息标识符 ；

(b) 验证所述密码验证值以确定是否已经修改了所述消息标识符或者所述加密的数据段中的至少一个加密的数据段 ；

(c) 通过计算从所述内部秘密状态的至少部分开始并且促成消息密钥的多个相继中间密钥来根据所述内部秘密状态和所述消息标识符导出所述消息密钥，其中至少基于所述消息标识符的部分和先前密钥来导出每个相继中间密钥 ；以及

(d) 使用至少基于所述消息密钥的所述一个或者多个密码密钥以解密所述加密的数据的一个或者多个经验证段以产生一个或者多个明文数据段 ；

其中所述步骤 (b) 包括 ：(i) 根据至少一个所述加密的数据段计算密码散列 ；(ii) 根据至少秘密值和所述密码散列导出预期验证器，其中所述导出包括计算始于所述秘密值的多个相继中间值，其中每个相继中间值至少基于所述相继中间值中的先前值和所述密码散列的部分 ；以及 (iii) 比较所述导出的预期候选验证器与所述接收的密码验证值。

23. 根据权利要求 22 所述的方法，其中在 (c) 中所述导出每个所述相继中间密钥包括选择熵分布操作，然后将所述熵分布操作应用于所述先前密钥。

24. 根据权利要求 23 所述的方法，其中将所述消息标识符分解成多个部分，每个所述部分确定待应用的特定熵分布操作。

25. 根据权利要求 24 所述的方法，其中所述熵分布操作是至少依赖于以下各项的密码散列操作 ：(i) 所述先前密钥 ；以及 (ii) 对应于所述消息标识符的至少部分的值。

26. 根据权利要求 22 所述的方法，其中使用密钥链接根据所述消息密钥导出 (d) 中的所述密码密钥。

27. 根据权利要求 22 所述的方法，其中通过遵循经过密钥树的路径来计算 (c) 中的所述消息密钥，所述路径包括所述消息标识符，并且其中通过遵循经过密钥树的路径来计算所述验证器，所述路径包括基于所述加密的数据段的散列。

28. 根据权利要求 22 所述的方法，其中 (d) 包括通过执行多个熵分布操作根据所述消息密钥导出多个所述密码密钥。

29. 根据权利要求 28 所述的方法，其中 (i) 通过将熵分布函数应用于所述消息密钥来计算所述密码密钥之一；并且 (ii) 通过将熵分布函数应用于多个相继密码密钥中的先前密码密钥来计算所述密码密钥中的每个密码密钥。

30. 根据权利要求 22 所述的方法，其中：(i) 所述加密的数据包括多个段；(ii) 除了最后的加密的数据段之外的每个加密的数据段包括下一加密的数据段的密码散列的表示；(iii) 在解密第一加密的数据段之前计算和验证它的密码散列；(iv) 在解密每个后续段之前计算它的密码散列并且与在先前的加密的数据段中表示的所述散列比较；并且 (v) 在解密仅一个加密的数据段时使用每个所述密码密钥。

31. 根据权利要求 22 所述的方法，其中在 (d) 中：(i) 所述加密的数据包括多个段，并且每段包括多个子段；并且 (ii) 对于每个加密的子段，所述密码密钥中的新密码密钥用来解密所述子段。

32. 根据权利要求 22 所述的方法，其中所述散列依赖于所有所述加密的数据段。

33. 根据权利要求 22 所述的方法，其中在 (b) 中：所述加密的数据包括多个段；(ii) 密码密钥更新在段内基础上出现；并且 (iii) 通过使用所述更新的密码密钥中的至少一个密码密钥来产生每个所述明文数据段。

34. 根据权利要求 22 所述的方法，其中每个所述相继中间值是密码散列操作的结果，所述密码散列操作的输入包括：(A) 父值；以及 (B) 它的对应于所述加密的数据段散列的至少部分的值。

35. 根据权利要求 34 所述的方法，其中向所述密码散列操作的所述输入仅通过以下确定：(1) 所述相继中间值的直接父值；以及 (2) 所述加密的数据段散列的一位或者两位。

36. 根据权利要求 22 所述的方法，其中存在仅一个明文数据段和仅一个加密的数据段，并且其中所述密码散列计算还包括数据段长度。

37. 根据权利要求 22 所述的方法，其中所述预期验证器验证散列，其中：(i) 所述散列依赖于第一加密的数据段，所述第一加密的数据段并入第二加密的数据段的散列；并且 (ii) 所述第二加密的数据段和后续加密的数据段各自并入下一加密的数据段的散列。

38. 根据权利要求 22 所述的方法，其中所述消息标识符为计数器。

39. 根据权利要求 22 所述的方法，其中所述消息标识符是所述明文数据的至少部分的密码散列。

40. 根据权利要求 22 所述的方法，其中所述明文数据包括 FPGA 位流。

41. 根据权利要求 22 所述的方法，包括使用处理器以执行所述明文数据的至少部分的附加步骤。

42. 根据权利要求 22 所述的方法，用来使由包含处理器的芯片从外部存储器向片上高速缓存中加载的数据安全。

43. 根据权利要求 22 所述的方法，其中所述设备是从解密设备接收数据分组的联网通

信设备。

44. 根据权利要求 22 所述的方法，还包括防回滚保护。

45. 根据权利要求 22 所述的方法，其中所述密码验证值是认证所述加密的数据段中的至少一个加密的数据段的数字签名。

46. 一种用于加密明文数据而又限制重用密码密钥的设备，所述设备包括 ：

装置 (a)，用于通过计算从内部秘密状态的至少部分开始并且促成消息密钥的多个相继中间密钥来根据所述内部秘密状态和消息标识符导出所述消息密钥，将基于所述消息标识符的至少部分和先前密钥来导出每个相继中间密钥 ；

装置 (b)，用于使用至少基于所述消息密钥的所述一个或者多个密码密钥以加密所述明文数据的一个或者多个段以产生一个或者多个加密的数据段 ；

装置 (c)，用于使用秘密密钥以计算至少基于所述加密的数据段中的一个或者多个加密的数据段并且可用于验证所述加密的数据段中的一个或者多个加密的数据段的密码验证值 ；以及

装置 (d)，用于输出所述一个或者多个加密的数据段和所述密码验证值 ；

其中所述装置 (c) 包括 ：(i) 用于根据至少一个所述加密的数据段计算密码散列的装置 ；以及 (ii) 用于根据至少秘密值和所述密码散列导出验证器的装置，其中所述导出包括计算始于所述秘密值的多个相继中间值，每个相继中间值将至少基于所述相继中间值中的先前值和所述密码散列的部分。

47. 根据权利要求 46 所述的设备，其中装置 (a) 包括用于包括选择熵分布操作，然后将所述熵分布操作应用于所述先前密钥的装置。

48. 根据权利要求 47 所述的设备，其中所述熵分布操作是至少依赖于以下各项的密码散列操作 ：(i) 所述先前密钥 ；以及 (ii) 对应于所述消息标识符的至少部分的值。

49. 根据权利要求 46 所述的设备，其中通过遵循经过密钥树的路径来计算 (a) 中的所述消息密钥，所述路径包括所述消息标识符。

50. 根据权利要求 46 所述的设备，其中装置 (b) 包括用于通过执行多个熵分布操作根据所述消息密钥导出多个所述密码密钥的装置 ；并且所述设备包括 ：(i) 用于通过将熵分布函数应用于所述消息密钥来计算所述密码密钥之一的装置 ；以及 (ii) 用于通过将熵分布函数应用于多个相继密码密钥中的先前密码密钥来计算所述密码密钥中的每个密码密钥的装置。

51. 根据权利要求 46 所述的设备，其中每个所述相继中间值是密码散列操作的结果，所述密码散列操作的输入仅通过以下确定 ：(1) 所述相继中间值的直接父值 ；以及 (2) 所述加密的数据段散列的一位或者两位。

52. 根据权利要求 46 所述的设备，其中所述密码验证值是认证所述加密的数据段中的至少一个加密的数据段的数字签名。

53. 一种用于解密数据而又限制重用密码密钥的设备，所述设备包括 ：

装置 (a)，用于接收一个或者多个加密的数据段和密码验证值并且获得与之对应的消息标识符 ；

装置 (b)，用于验证所述密码验证值以确定是否已经修改了所述消息标识符或者所述加密的数据段中的至少一个加密的数据段 ；

装置 (c)，用于通过计算从内部秘密状态的至少部分开始并且促成消息密钥的多个相继中间密钥来根据所述内部秘密状态和所述消息标识符导出所述消息密钥，其中至少基于所述消息标识符的部分和先前密钥来导出每个相继中间密钥；以及

装置 (d)，用于使用至少基于所述消息密钥的所述一个或者多个密码密钥以解密所述加密的数据的一个或者多个经验证段以产生一个或者多个明文数据段；

其中所述装置 (b) 包括：(i) 用于根据至少一个所述加密的数据段计算密码散列的装置；(ii) 用于根据至少秘密值和所述密码散列导出预期验证器的装置，其中所述导出包括计算始于所述秘密值的多个相继中间值，其中每个相继中间值至少基于所述相继中间值中的先前值和所述密码散列的部分；以及 (iii) 用于比较所述导出的预期候选验证器与所述接收的密码验证值的装置。

54. 根据权利要求 53 所述的设备，其中装置 (c) 包括用于包括选择熵分布操作，然后将所述熵分布操作应用于所述先前密钥的装置。

55. 根据权利要求 54 所述的设备，其中所述熵分布操作是至少依赖于以下各项的密码散列操作：(i) 所述先前密钥；以及 (ii) 对应于所述消息标识符的至少部分的值。

56. 根据权利要求 53 所述的设备，其中通过遵循经过密钥树的路径来计算 (c) 中的所述消息密钥，所述路径包括所述消息标识符。

57. 根据权利要求 53 所述的设备，其中装置 (d) 包括用于通过执行多个熵分布操作根据所述消息密钥导出多个所述密码密钥的装置；并且所述设备包括：(i) 用于通过将熵分布函数应用于所述消息密钥来计算所述密码密钥之一的装置；以及 (ii) 用于通过将熵分布函数应用于多个相继密码密钥中的先前密码密钥来计算所述密码密钥中的每个密码密钥的装置。

58. 根据权利要求 53 所述的设备，其中每个所述相继中间值是密码散列操作的结果，所述密码散列操作的输入仅通过以下确定：(1) 所述相继中间值的直接父值；以及 (2) 所述加密的数据段散列的一位或者两位。

59. 根据权利要求 53 所述的设备，其中存在仅一个明文数据段和仅一个加密的数据段，并且用于所述密码散列计算还包括数据段长度。

60. 根据权利要求 53 所述的设备，其中所述预期验证器验证散列，并且其中：(i) 所述散列依赖于第一加密的数据段，所述第一加密的数据段并入第二加密的数据段的散列；并且 (ii) 所述第二加密的数据段和后续加密的数据段各自被配置成并入下一加密的数据段的散列。

61. 根据权利要求 53 所述的设备，其中所述设备为 FPGA，并且其中所述设备被配置用于所述明文数据包括 FPGA 位流。

62. 根据权利要求 53 所述的设备，还包括用于使用所述设备中包含的处理器以执行所述明文数据的至少部分的装置。

63. 根据权利要求 53 所述的设备，其中所述设备是包含处理器的芯片，并且其中所述设备包括用于使由所述芯片从外部存储器向片上高速缓存中加载的数据安全的装置，并且其中所述数据将如由所述芯片先前加密的那样被加载。

64. 根据权利要求 53 所述的设备，其中所述密码验证值是认证所述加密的数据段中的至少一个加密的数据段的数字签名。

可验证防泄漏的加密和解密

[0001] 相关申请的交叉引用

[0002] 本申请要求对通过引用而整体结合于此的、于 2010 年 12 月 02 日提交的第 12/958,570 号美国专利申请和于 2009 年 12 月 4 日提交的第 61/266,948 号美国临时专利申请的优先权。

技术领域

[0003] 本专利涉及用于处理加密的数据输入的技术，并且更具体地涉及保护这样的系统和数据以防外部监视攻击。

背景技术

[0004] 对敏感数据进行操作的系统需要防范攻击者对这样的数据的未授权访问或者公开或者变更。获得对密码密钥和其它秘密的访问的攻击者可以窃取或者篡改敏感数据，从而导致严重后果（比如通过引入未授权命令和暴露机密的或者专有的信息来颠覆系统的关键操作）。一个被危及的元件也可能用来增加更多攻击，从而危害系统的其它元件。更具体地，先前的研究已经说明了攻击者可以监视设备的外部特性（比如操作时序、功率消耗和 / 或电磁辐射）并且使用这一附加信息以提取正在设备内使用的秘密密钥。例如，如 Kocher 等人描述的那样（见 P.Kocher、J.Jaffe、B.Jun 的“Differential Power Analysis”，Advances in Cryptology-Crypto 99Proceedings，Lecture Notes In Computer Science Vol.1666，Springer-Verlag，1999），在本领域中公知的是对将相同一组密钥与不同数据一起用来执行密码操作序列的设备的外部监视可能造成密钥泄漏。

[0005] 由于外部监视攻击通常是被动的和非入侵的，所以基于阻碍物理访问或者检测不恰当使用的传统防篡改防范对于提供保护以防这样的攻击而言是不足或者不切实际的。例如，在背景技术中已知用于使用物理上安全、屏蔽良好的空间来管理秘密密钥的方法。然而，在许多应用中，要求密码系统保持于物理上隔离的设施中在给定了它们预期操作于其中的环境时是不可行的。此外，这样的设施建造和运营起来昂贵并且可能在它们防止将少量信息泄漏给对手的能力上仍有缺点。

[0006] 当然，在背景技术中已知可以减轻监视攻击所致的信息泄漏问题而不必依赖于物理屏蔽的其它方法。这些包括如下方法，该方法用于减少从事务泄漏信息的数量（或者速率）、修改密码算法实现方式以对计算中介进行随机化和 / 或在功率消耗和操作时序方面引入噪声。

[0007] 例如，标题为“Leak-Resistant Cryptographic Indexed Key Update”的美国专利 6539092 提供了用于将共享的主密钥和索引值（例如，计数器）转换成事务密钥的方法，其中保护导出以防外部监视攻击。这些方法在如下应用中工作良好，在这些应用中，为了防范外部监视攻击而保护的设备可以对事务密钥的导出起作用。例如，第 092 号专利描述了智能卡如何可以维护随着每个事务而递增的索引计数器，然后在密钥导出中使用索引计数器。

[0008] 然而，存在如下应用，在这些应用中，应当保护协议中的参与者以防外部监视攻击，但是缺乏如第092号专利中描述的那样的存储序列计数器和更新的密钥的能力。例如，考虑如下情况，在该情况中设备需要有规律地处理相同输入数据（比如如下设备，该设备包含固定并且不变的嵌入式密钥，该密钥反复用来以任意顺序解密密文）。固件加密是这样的应用的示例；可以制造在熔断器中具有嵌入式密钥的微处理器，并且在每次重新引导时，微处理器需要重新解密它的从非信任外部闪存加载的固件映像。可以偶然更新固件映像，但是也可以反复解密相同的密文。因此，应用要求和物理制造限制（比如由于使用一次性可编程熔断器以保持密钥而不能修改存储的密钥）两者可能使设备限制解密密钥将被使用的次数不切实际。固件发行者可以每当发布新的加密的固件时将在'092专利中描述的方法与新索引值一起使用，但是解密设备无法在每次重新引导时使用不同索引值，因为将索引值改变成除了由加密设备使用的值之外的值将造成不正确解密。因此，攻击者可以潜在地向解密设备供应篡改的数据集，继而在设备处理（例如，解密等）这些密文时尝试通过监视外部特性来恢复秘密密钥。统计旁信道攻击（比如差分功率分析(DPA)）可以根据在设备反复使用相同密钥以对不同输入值（比如之前示例中的不同固件密文或者相同固件密文的篡改的版本）操作时收集的一组测量推断秘密密钥。即使未篡改密文消息，来自单个长消息（例如，包括多个块密码输入）或者合法消息的汇集（比如多个固件版本）的测量也可以提供用于旁信道攻击的充分数据。

[0009] 当然，在其中设备将相同密钥用于每个事务的一些情形中，设备在理论上可以实施封锁（例如，通过如果超过事务或者失败阈值则自毁）以限制对手可以观测的事务数目。然而，封锁机制引入了诸多实际问题，比如与存储失败计数器相关联的可靠性顾虑和困难（例如，许多半导体制造工艺缺乏安全的片上非易失性存储并且难以使片外存储安全）。

[0010] 鉴于所有前述，一种提供用于设备传达和交换数据的而防范外部监视攻击的可验证安全方式以及设备拒绝非真实数据的能力的方法将是有利的。

发明内容

[0011] 本专利描述用于使利用秘密密码密钥的设备安全免受外部监视攻击以及提供改善的安全性以防收集与设备的内部操作相关的信息的常规密码分析和其它攻击（比如DPA和其它形式的外部监视攻击）的方式。在说明书中公开了用于加密敏感数据的各种示例性实施例。

[0012] 尽管这些各种实施例可以在它们的细节上明显变化，但是它们如关于在说明书中描述的各种实施例可以容易验证的那样都涵盖于以下通用技术内：关于加密，待加密的每个数据集与消息标识符（比如事务/消息计数器、明文的散列、随机值或者另一唯一或者半唯一值）相关联。加密设备使用消息标识符和与解密设备共享的初始秘密内部状态来导出消息密钥。通过从共享的秘密内部状态的至少部分开始并且促成消息密钥的一连串一个或者多个中间密钥以迭代方式执行这一导出，其中在每次迭代中，下一密钥依赖于至少一个先前密钥和消息标识符的至少部分。可以将明文分解成一段或者多段。用可以包括消息密钥或者根据消息密钥进一步导出的密钥的一个或者多个秘密密钥加密每个明文段以创建对应的加密段。通常，不同密钥（或者不同的一组密钥）用于每段。

[0013] 加密设备继而使用与解密设备共享的秘密密钥（比如消息密钥、秘密内部密钥、

不同密钥、根据前述各项导出的密钥等）以计算至少一个验证器。可以使用与用来产生消息密钥的迭代过程相似的迭代过程来执行导出验证器，借此将变换的序列应用于秘密密钥以产生相继值（例如，其中生成每个中间密钥包括散列化它的父值）。

[0014] 加密设备输出一个或者多个加密段和一个或者多个验证器。也可以如需要的那样输出附加信息以使接收方能够确定消息标识符。

[0015] 在对应的解密过程期间，解密设备接收一个或者多个加密段、一个或者多个验证器和对应于加密段的消息标识符。它继而使用一个或者多个验证器以验证至少尚未修改待解密的第一加密段。验证器的验证可以包括计算从与加密设备共享的秘密开始的相继中间值序列，并且其中每个中间值是它的父代的散列（并且具体散列操作依赖于所述加密段的散列的部分）。通常，如果验证未修改加密段则才允许用于该段的解密过程继续。如果验证成功，则解密设备通过遵循与加密设备遵循的相同迭代密钥导出过程（即从共享的秘密内部状态的至少部分开始、经过中间密钥序列、促成最终消息密钥，其中在每个步骤，下一密钥依赖于消息标识符的至少部分和至少一个先前密钥）、使用它与加密设备共享的秘密内部状态来计算消息密钥（如果尚未导出）。用根据消息密钥导出的一个或者多个对应的秘密密钥解密每个加密段（如果确定为未修改）以恢复对应的明文段。

附图说明

[0016] 图 1 示出了用于使用密钥和密文散列链接的可验证、防泄漏加密的整个过程的示例性实施例。

[0017] 图 2 示出了始于共享的密码秘密 K_{START} 并且经过路径 $P_1 \ldots P_Q$ 继续的防泄漏基于密钥树的密钥导出过程的示例性实施例。图 2 的密钥导出过程可与图 1 和图 3 的第一示例性加密过程以及图 4 的第一示例性解密过程结合使用。它也可与图 5、图 11 和图 13 的其它示例性加密过程以及图 6、图 12 和图 14 的其它示例性解密过程结合使用。

[0018] 图 3 示出了用于加密的防泄漏密钥和密文散列链接过程（例如，包括图 1 中所示的整个加密过程的部分）的示例性实施例。

[0019] 图 4 示出了对应于图 1（和图 3）的加密过程的使用密钥和密文散列链接的可验证、防泄漏解密过程的示例性实施例。

[0020] 图 5 示出了用于使用密钥和明文散列链接的可验证、防泄漏加密的过程的示例性实施例。

[0021] 图 6 示出了对应于图 5 的加密过程的用于使用密钥和明文散列链接的可验证、防泄漏解密的过程的示例性实施例。

[0022] 图 7 示出了如下环境，在该环境中，可验证、防泄漏密码操作用于向片上系统上加载固件。

[0023] 图 8 示出了如下环境，在该环境中，可验证、防泄漏密码操作使用于安全 CPU 芯片内，其中不信任外部存储器（比如闪存和 / 或者 RAM）。

[0024] 图 9 示出了如下环境，在该环境中，可验证、防泄漏密码操作用于向现场可编程门阵列上加载位流映像。

[0025] 图 10 示出了如下环境，在该环境中，可验证、防泄漏密码操作使用于基于分组的网络通信设备中。

[0026] 图 11 示出了可以用在图 10 中描述的环境以及在其它实施例中使用的用于可验证分组级防泄漏加密的过程的示例性实施例。

[0027] 图 12 示出了对应于图 11 中描述的加密过程的用于可验证分组级防泄漏解密的过程的示例性实施例。

[0028] 图 13 示出了使用具有段内密钥改变的密码块链接 (CBC) 的示例性 ENC() 操作的示例性实施例。

[0029] 图 14 示出了对应于图 13 的加密操作的使用具有段内密钥改变的密码块链接 (CBC) 的示例性 DEC() 操作的示例性实施例。

具体实施方式

[0030] 在本专利中描述的技术使多方能够传达密码保护的敏感数据并具有增加的安全性以防外部监视攻击。虽然描述了涉及到两方（通常称为“加密设备”和“解密设备”）的示例性实施例，但是术语“设备”是为了便利而加以选择并且无需必然直接对应于系统设计中的任何特定角色。设备可以但是并不必须利用不同形式的因素或者实现方式。例如，加密设备和解密设备可以均为便携硬件设备。备选地，加密设备可以是在操作于设施中的服务器上运行的软件应用，而解密设备可以是便携式硬件设备（或者反之亦然）。另外，虽然多数密码操作涉及到两方，但是本专利的技术当然可以应用于仅涉及到一方的环境中（比如安全存储器或者存储系统（其中两个角色例如在图 8 中所示的示例性环境中在单方和 / 或单个设备的控制之下）中）或者应用于涉及到多于两方和 / 或两个设备的环境中（比如图 10 中所示的示例性实施例中）。

[0031] 熵重新分布操作

[0032] 如这里所用，“熵重新分布操作”（或者“熵分布操作”）是如下操作，该操作混合它的输入，从而使得关于输入位的未知信息在输出位之间重新分布。例如，假设用熵重新分布操作 f 反复处理 x 位密码密钥 K_0，从而使得对于每个 $i > 1$ 而言密钥 $K_i = f(K_{i-1})$。接着，假设对手获得关于 n 个不同密钥 K_i 中的每个密钥的 y 位信息（例如，作为尝试的外部监视攻击的部分而获得），从而提供用于求解密钥 K_0 的绰绰有余的信息（例如 $y*n > x$）。使用熵分布操作 f 可以使这样的解决方案在计算上不可行。密码散列函数 H 是可以作为熵重新分布操作而使用的操作的示例。例如，考虑产生 256 位结果的强散列函数 H。在随机 256 位初始密钥 K_0 给定时，对于每个 $i > 1$ 而言令 $K_i = H(K_{i-1})$。知道（例如）每个 $K_0 \ldots K_{999,999}$ 的最低有效位的对手具有与 K_0 有关的 1,000,000 位数据。具有无限计算能力的假想对手可以通过测试用于 K_0 的所有可能的 2^{256} 个值以标识与最低有效位的已知序列一致的值来发现 K_0。然而，实际对手具有有限计算能力可用，并且熵重新分布操作阻碍其成为一种用于在给定通过尝试的外部监视攻击而泄漏的信息时求解 K_0（或者任何其它 K_i）的在计算上实际的方式。

[0033] 可以而不限于使用密码散列函数、使用块密码（比如 AES）构造的操作、伪随机变换、伪随机排列、其它密码操作或者其组合来实施熵重新分布操作。为求便利，关于散列描述某些示例性实施例，但是本领域技术人员将理解：按照前文，附加地或者备选地也可以使用其它熵重新分布函数。

[0034] 也可以根据基本操作构造多个熵重新分布操作。举例而言，如果需要两个 256 位

熵重新分布操作 $f_0()$ 和 $f_1()$，则 $f_0()$ 可以包括将 SHA-256 密码散列函数应用于与向 $f_0()$ 的输入级联的操作标识符串“f0”，而 $f_1()$ 可以包括将 SHA-256 应用于与和 $f_1()$ 的输入级联的操作标识符串“f1”。可以使用公知的 AES 块密码来解析熵重新分布操作。例如，为了实施 $f_0()...f_{b-1}()$，每个 $f_i()$ 可以使用它的输入作为 AES-256 密钥以加密对于 i 在 0...b-1 内的选择而言唯一的一对 128 位输入块，从而产生 256 位的输出。在背景技术中也知道并且也可以运用广泛多种基于块密码的散列函数和 MAC 构造。

[0035] 共享的密码值和操作

[0036] 本节描述由加密设备和它的对应的解密设备两者共享用来执行如在本专利中描述的可验证防泄漏密码操作的某些密码值和 / 或操作。

[0037] 设置加密设备和解密设备，从而使得每个设备具有对基本共享秘密密码状态值（比如表示为 K_{ROOT} 的秘密密钥）的访问权。这一秘密状态可以例如存储于防篡改芯片上的 EEPROM、闪存、熔断器或者其它存储中的一项或者多项中，并且可以完全或者部分根据其它值或者过程来导出，或者可以从外部获得。这些设备中的每个设备获得 K_{ROOT} 的方法可以包括而不限于每个设备与 K_{ROOT} 一起制造、设备通过经由物理键控接口接收 K_{ROOT}、随机生成 K_{ROOT}（例如，如果加密设备和解密设备相同）等来直接或者经由第三方相互协商 K_{ROOT}（例如，使用如下协议，这些协议利用 RSA、Diffie-Hellman 或者其它公共密钥密码技术或者对称技术）。

[0038] 此外，加密设备和解密设备也均能够计算一组非线性密码熵重新分布操作 $f_0()$，$f_1()$，...，$f_{b-1}()$，其中 $b > 1$ 为正整数。可以在树结构中配置这些 b 熵重新分布函数。例如，可以通过使用 b 个不同熵分布函数 $f_0()...f_{b-1}()$ 以代表这一 b 进制树在树的每个节点（每个节点代表可能的导出的密钥）的 b 个可能的分支来创建高度为 Q 的简单 b 进制树结构。在这样的树中，始于根密码密钥 K_{START}（该密钥在 0 级），可以在 1 级计算 b 个可能的导出的密钥：用于最左分支的 $f_0(K_{START})$；用于下一分支的 $f_1(K_{START})$；并且继续直至用于最右分支的 $f_{b-1}(K_{START})$。在 2 级，可以导出 b^2 个可能的密钥，因为 $f_0()...f_{b-1}()$ 中的每个函数可以应用于 b 个可能的 1 级密钥中的每个密钥。当然，计算具体 2 级节点仅需两次（而不是 b^2 次）计算（即不计算不在路径上的节点）。树针对相继 1 级至 Q 级继续，其中可以通过轮流应用 $f_0()...f_{b-1}()$ 来处理先前级的每个可能的密钥（即不同节点）以导出 b 个附加的可能的导出的密钥。整个密钥树具有从在 0 级的单个节点开始、以在 i 级的 b^i 个节点继续并且以在 Q 级的 b^Q 个节点为结束的 Q+1 级。因此，存在从 0 级的根节点到 Q 级的 b^Q 个最终节点的 b^Q 个可能的路径。对应于在不同级应用的唯一函数序列的每个这样的可能路径可以表示为 Q 个整数的序列（每个整数选自于 (0...b-1)）。

[0039] 例如，在示例性实施例中 $b = 2$。因此，使用（并且可以例如如上文描述的那样根据基本操作构造）两个熵重新分布操作 $f_0()$ 和 $f_1()$。如果 $Q = 128$（即高度为 128），则 2^{128} 个路径是可能的并且需要 128 次熵重新分布函数计算以根据 0 级节点（即起始密钥）导出 Q 级密钥。

[0040] 作为变体，实施例可以涉及到 b 的更多选择多样性（比如在级之间变化 b 的值和 / 或基于去往特定级而取道的路由来变化 b）。类似地，也可以比如通过使熵重新分布操作 $f_i()$ 在不同级不同或者使这些操作依赖于对特定级采取的序列来变化熵重新分布操作。

[0041] 加密设备和解密设备也能够执行可以（但是未必）与函数 $f_i()$ 相异的密码非线

性密钥链接操作 g()。例如，在一个实施例中，g() 由密码散列操作构成。不同实施例可以将不同函数用于 g() 的不同应用（包括根据基本函数构造的变体（例如，通过用计数器或者代表 g() 的应用的另一个值散列化输入数据））。

[0042] 加密设备和解密设备也具有可以（但是未必）与操作 $f_i()$ 和 g() 相异的密码防冲突单向散列函数 h()（例如，用作段散列函数）。

[0043] 在示例性实施例中，通过将操作 $f_i()$、g() 和 h() 中的每个操作计算为操作标识符和输入数据的密码散列来根据共同密码散列函数构造每个操作。操作标识符可以例如是由“f＃”、“g”或者“h”构成的零结尾串，其中＃是用于给定的 $f_i()$ 的 i 的值，从而使得用于 $f_0()$ 的操作标识符将为“f0”。使用输入作为密钥的操作标识符的 HMAC 也可以用来实施这些操作。可与本专利的技术一起使用的散列函数包括而不限于 MD5、SHA-1、SHA-256、SHA-512、任何 SHA3 候选操作以及前述各项与使用前述各项的构造（比如 HMAC）的组合。如这里所用，函数 BLAKE、Blue Midnight Wish、CubeHash、ECHO、Fugue、Grostl、Hamsi、JH、Keccak、LANE、Luffa、Shabal、SHAvite-3、SIMD 和 Skein 中的每个函数是“SHA3 候选操作”。在其它实施例中，使用将块密码（比如 AES、DES 或者其它密码）转换成散列函数的其它公知构造（比如而不限于 Matyas-Meyer-Oseas、Davies-Meyer、Miyaguchi-Preneel、Merke-Damgard 等）来导出散列函数。非防冲突的变换（比如 MD5、散列变换的精简轮回变体或者其它混合操作）也可以重新分布输入中存在的熵、但是对于用作单向函数 h() 而言吸引力将更小。

[0044] 更多其它实施例可以在实施熵重新分布操作 $f_{0...b1}()$ 时利用流密码（潜在地包括轻量级并且潜在地在密码上为弱的流密码）。例如，可以运用流密码 RC4，其中熵重新分布操作输入用作 RC4 密钥并且 RC4 输出字节用作（或者用来形成）熵重新分布操作输出。

[0045] 加密设备和解密设备具有秘密密钥加密函数（或者函数集）ENC()，其具有对应的解密函数 DEC()。在一些实施例（比如固定长度消息的实施例）中，ENC() 和 DEC() 可以比如在 ECB 或者 CBC 模式中利用常规密码构造（比如 AES）。后文分别关于图 13 和图 14 描述用于其它实施例的 ENC() 和 DEC() 构造。

[0046] 图 1 和图 2 中的示例性实施例

[0047] 本节描述用于可验证防泄漏加密和解密的通用技术的示例性实施例。这一第一示例性实施例使用密钥链接和密文散列链接。加密

[0048] 为求便利，按照密码学中的传统命名法，使用术语“明文”以指代待加密的数据。如本领域技术人员将理解的那样，这未必意味着输入数据为人类可读的，并且实际上并未排除在利用本专利的技术保护这样的数据之前对其本身进行压缩、编码乃至加密。类似地，本领域技术人员将理解术语“数据”涵盖被处理的任何量并且可以包括而不限于内容、数据、软件、代码和任何其它类型的信息。

[0049] 在给定待保护的敏感明文数据消息 D 时，并且在知道共享的基本秘密密码值 K_{ROOT} 时，加密设备执行如图 1 中概括的以下步骤。首先，它将敏感明文数据 D 分解成 L 段 $D_1,...,D_L$ 的序列（步骤 100），其中 $(L \geq 1)$，每段小到足以相配到接收器中的用于传入段的存储器中。此外，这些段中的每段的大小应当充分小以满足应用和实现方式的泄漏要求。段可以但是未必是相同大小。此外，其它变体也可以如下文将关于图 13 和图 14 示出的那样通过改变密钥（例如，在 ENC() 和 DEC() 内）来支持无限制大小的段。

[0050] 加密设备也生成（步骤 101）临时数 N，该临时数（如下文将示出的那样）可以用

作用于与对 D 的加密结合使用的消息标识符（或者其前身）。例如，可以使用真随机数生成器、伪随机数生成器、真随机数生成器与伪随机数生成器的一些组合、计数器值或者其它（优选为唯一或者很少重复的）参数或者通过根据可用于加密设备的密钥和 / 或数据（包括而不限于 D（例如，通过将 N 设置成 D 的部分或者全部的散列））导出 N 来生成该临时数。在图 1 中，对于给定的 K_{ROOT}，用来加密特定消息的 N 的值优选地未用来加密任何其它消息（或者如果用来加密，则任何重用应当是有限的、不太可能的和 / 或不频繁的）。

[0051] 在以下示例性实施例中，使用临时数 N 来形成消息标识符 H_1。在最简单直接的实现方式（其中 N 充当消息标识符）中，H_1 可以简单地等于 N。作为另一示例（其中 N 充当消息标识符的前身），加密设备可以将 H_1 计算（步骤 102）为使用函数 h() 的 N 的散列。散列化在如下情形中有用，在这些情形中，希望产生固定大小的消息标识符以例如在为求计算效率而对更短数量操作时允许并入更长数据值（比如文本串），或者为求计算简化而将可变长度的数据值转换成统一长度的消息标识符，或者减少对手可能具有的对 H_1 的选择有影响的任何能力。当然，散列化仅为一种用于产生消息标识符的方式，并且本领域技术人员将理解可以运用除了 h 之外的其它函数以产生 H_1。

[0052] 在计算 H_1 之后，加密设备使用共享的基本秘密密码值 K_{ROOT} 和 H_1（103）作为向防泄漏的、基于密钥树的密钥导出过程的输入来计算消息密钥 $K_{MESSAGE}$。为了便于讨论，这里在（例如，由加密设备执行的）加密的上下文中并且更具体地在图 1 的第一示例性加密过程的上下文中呈现密钥导出过程。然而，相同密钥导出过程也将使用于图 4 的第一示例性解密过程中，在该情况下，它将由解密设备执行。类似地，密钥导出过程也将与其它过程（包括图 5、图 11 和图 13 的示例性加密过程以及图 6、图 12 和图 14 的示例性解密过程）结合使用。

[0053] 在图 2 中图解表示示例性密钥导出过程。该过程始于树的起点（表示为 K_{START}（201））和路径 $P_1 \ldots P_Q$（202）。例如，在图 1 的上述步骤 103 中，K_{START} 是共享的秘密密钥 K_{ROOT} 的值，并且路径 $P_1 \ldots P_Q$（202）根据 H_1 来确定。（下文讨论将 H_1 转换成 $P_1 \ldots P_Q$。）路径指定将应用于 K_{START} 的一连串熵重新分布操作。

[0054] 在示例性实现方式中，将消息标识符 H_1 分解成 Q 个部分 $P_1, P_2, \ldots, P_Q$。在示例性分解中，每个部分 P_i 是来自 0 至 (b-1) 的整数（例如，如果 b = 4，则每个 P_i 是两位值（0、1、2 或者 3））。类似地，如果 b = 2，则每个 P_i 是单个位（0 或者 1）。因此，路径部分 $P_1 \ldots P_Q$ 可以用来通过应用函数 $f_0(), f_1(), \ldots, f_{b-1}()$ 以产生促成 $K_{START,PATH}$ 的多个中间密钥来如下指定从 K_{START} 到 $K_{START,PATH}$ 的具体路径。首先，将函数 f_{P_1} 应用于 K_{START}（203）以产生中间密钥 K_{START,P_1}，继而对 K_{START,P_1} 应用 f_{P_2} 以产生中间密钥 K_{START,P_1,P_2}（204），并且以此类推直至对中间密钥 $K_{START,P_1,P_2,\ldots,P_{Q-1}}$（205）应用 f_{P_Q} 以产生最后导出的密钥 $K_{START,P_1,P_2,\ldots,P_Q}$（206）。注意，每个中间密钥的导出依赖于至少一个前代密钥（例如，在图 2 的情况下为它的直接父代）和消息标识符的相关部分。为求便利，应当用符号表示 $K_{START,PATH}$（指示通过始于密钥 K_{START} 并且遵循 PATH 而达到的密钥）表示这一最后导出的密钥。类似地，在图 1 的步骤 103 的情况下，最后导出的密钥（向 $K_{MESSAGE}$ 分配的消息密钥）表示为 K_{ROOT,H_1}，因为起始密钥事实上为 K_{ROOT} 并且路径事实上为 $P_1, P_2, \ldots, P_Q$，这简单地是 H_1 的分解。（在备选实施例中，可以例如通过散列化 K_{ROOT,H_1} 来根据 K_{ROOT,H_1} 导出 $K_{MESSAGE}$。在任一方式中，$K_{MESSAGE}$

基于K_{ROOT,H_1}。）

[0055] 在步骤 104，基于所述消息密钥 $K_{MESSAGE}$使用至少一个密码密钥来加密数据段从而根据输入段 $D=D_1,\ldots,D_L$产生密文 $E=E_1,\ldots,E_L$。在图 3 中示出了用于步骤 104 的示例性实施例，该图描绘了在计算加密段 $E_1,\ldots,E_L$时涉及到的步骤和状态。

[0056] 图 3 的过程使用 $K_{MESSAGE}$以计算 L 个个别段加密密钥 K_i（i ＝ 1 至 L），每个密钥用来加密秘密消息数据 D 的对应段 D_i（i ＝ 1 至 L）。首先，将函数 g() 应用于 $K_{MESSAGE}$以产生 K_1(302)（将用于第一段的加密密钥）。继而，将函数 g() 应用于密钥 K_1以产生 K_2（用于第二段的加密密钥 (303)），并且以此类推。最后，将函数 g() 应用于密钥 K_{L-1}以产生 K_L（用于最后段的加密密钥 (305)）。将这一类型的过程称为密钥链接，因为加密密钥相互链接。

[0057] 在已经确定用于加密 L 段的 L 个密钥 $K_1,\ldots,K_L$之后，段的加密如下进行。首先，处理最后（第 L 个）段，其中向 ENC() 函数的明文输入 (306) 是与通过在密码上散列化整个明文 $D_1\ldots D_L$而计算的消息完整性值级联的第 L 个数据段 D_L。（包括 $D_1\ldots D_L$的散列是可选的；实施例可以省略这一点或者级联其它数据（比如‘0’字节序列或者某一其它形式的填充））。这一第 L 个明文段由密钥 K_L加密以产生加密段 E_L(307)。

[0058] 接着，在 (308) 通过将散列函数 h() 应用于 E_L、向数据段 D_{L-1}附加这一散列值并且使用结果作为向第 L-1 段的加密输入来处理第 L-1 段。在 (309)，继而使用密钥 K_{L-1}来加密第 L-1 个明文段以产生加密段 E_{L-1}。针对其它段重复这一过程。例如，对应于第二明文段的加密输入 (310) 由后接 $h(E_3)$（第三加密段的散列）的第二数据段 D2 组成，并且继而使用密钥 K_2来加密输入 (310) 以产生加密段 E_2(311)。最后，对应于第一明文段的加密输入 (312) 由后接 $h(E_2)$（第二加密段 (311) 的散列）的第一数据段 D1 组成，继而使用密钥 K_1来加密输入 (311) 以产生加密段 E_1(313)。（作为前述的变体，无需加密后续段散列（例如，可以通过加密 D_i、继而级联加密结果与 E_{i+1}的散列来形成 E_i）。）

[0059] 加密段 $E_1\ldots E_L$形成密文 E。继而完成图 1 中的步骤 104。在计算 E_i时使用每个 E_{i+1}的散列有效地将加密值链接在一起，这适于使解密设备能够在解密缺陷段之前检测修改（或者缺陷）的密文段。将这称为“密文散列链接”。在上文所示的示例中，每个密文段 E_i(1 ＜ i ＜ L) 依赖于下一密文段的散列（例如，验证器 V 用来认证第一密文段 (E_1) 的散列，继而 E_1（如果必要则在解密成 D_1之后）产生 E_2，的预期散列。类似地，E_2（如果必要则在解密之后）产生段 E_3的散列），并且以此类推。

[0060] 注意，当所有数据在一段中（即 L ＝ 1）时（例如，由于输入消息小或者运用加密过程 ENC()（比如图 13 中所示的过程））仍然可以执行图 3 的过程。对于 L ＝ 1 的情况，仅需 K_1并且 $K_1=g(K_{MESSAGE})$。备选地，$K_{MESSAGE}$可以直接用作 K_1，在该情况下可以完全省略操作 g()。如上文描述的那样，包括 $D_1..D_L$（在该情况下将仅为 D_1，因为 L ＝ 1）的散列是可选的。该过程的结果是 $E=E_1$，因为这是仅有的段。

[0061] 回顾图 1，在已经计算数据段 D_i之后，计算如下验证器 V，该验证器将使加密消息的已授权接收方能够在解密之前认证密文。首先，将值 H_2计算 (105) 为第一加密段 E_1的散列。回顾第一段 E_1并入所有其它段的散列。因此，E_1的散列实际上反映所有段（包括段 E_1）的内容并且可以用来验证段都尚未改变。（可选地，除了 E_1之外，向产生 H_2的散列的输入也可以包括关于消息的附加信息（比如长度、版本号、发送方身份、N 的值等））。

[0062] 接着，解密设备使用秘密密钥以计算 V(106)，该 V 是消息标识符和明文段 E_i的验

证器。使用至少一个密文段的散列（例如，散列 $H_2 = h(E_1)$ 和初始秘密（例如，$K_{MESSAGE}$或者如在以下段落中描述的其它值））来计算验证器 V。可以使用图 2 中所描述的防泄漏的、基于密钥树的密钥导出过程以起始密钥 K_{START}为 $K_{MESSAGE}$和使用 H_2确定路径来执行 V 的计算(106)。因此，V 的导出包括计算促成 V 的多个相继中间值，其中每个值依赖于至少一个前代（例如，在图 2 的情况下为它的父值）和散列（例如，H_2）的相关部分。注意，函数 $f_i()$、值 b 等可以（但是并非必须）与在 (103) 中使用的函数 $f_i()$、值 b 相同。这一过程造成导出密钥$\mathbf{K_{MESSAGE,H_2}}$，该密钥是（或者被进一步处理以形成）验证器 V。

[0063] 前文描述在导出验证器时始于 $K_{MESSAGE}$，但是备选实施例可以始于不同值。例如，在步骤 104 的密钥 $K_{MESSAGE}$和在步骤 106 的密钥 $K_{MESSAGE}$可以互不相同，但是二者均根据$\mathbf{K_{ROOT,H_1}}$来导出。类似地，可以根据在步骤 104 使用的 $K_{MESSAGE}$导出在步骤 106 使用的密钥或者反之亦然，或者可以运用不同基本密钥（除了 K_{ROOT}之外）作为 K_{START}。当然，K_{ROOT}本身可以甚至用作 K_{START}（例如，如果 H_2是 N 和 / 或 H_1和一个或者多个密文段的散列）。

[0064] 验证器如在本专利中利用的那样是如下可验证密码证据：某一推定密文是对与特定消息标识符相关联的一些明文消息数据的加密的未修改版本，并且由具有对秘密密码值的访问权的实体产生。在步骤 106 构造的验证器可以便利地由接收方（比如解密设备）以如下方式验证，该方式避免易受差分功率分析和有关外部监视攻击。此外，验证器创建过程（即执行步骤 106）也使加密设备能够避免易受差分功率分析和有关外部监视攻击。

[0065] 在计算验证器之后，加密过程完成。在步骤 107，输出结果。输出数据由为了使接收方能够导出消息标识符而需要的信息（如果存在（例如，临时数 N））、验证器 V 和加密的结果 E（包括加密段 $E_1 \ldots E_N$）构成。通过组合密钥链接与密文散列连接，这一类型的加密过程能够产生具有消息认证的在密码上为强的输出，而又避免以如下方式重用位于加密设备中的秘密密钥，这些方式将有助于对加密设备不利的差分功率分析和有关攻击。以如下形式创建加密结果，该形式使解密设备能够执行解密而不以如下方式重用秘密密钥，这些方式将有助于对解密设备不利的差分功率分析和有关攻击。密钥树过程限制在形成 $K_{MESSAGE}$和验证器 V 时重用密钥，而密文散列链接方法限制使用在数据加密中使用的密钥。

[0066] 下一节说明输出数据如何可以随后由解密设备解密。

[0067] 解密

[0068] 图 4 示出了对应于图 1 和图 3 的示例性加密过程的示例性解密过程。如更早所言，这要求解密设备和加密设备二者具有导出相同消息标识符（例如，因为每个设备知道临时数 N，所以它可以计算 H_1）、基本秘密密码值 K_{ROOT}、密码函数 f()、g() 和 h() 的能力。示例性解密过程将使用图 2 中描绘的相同密钥导出过程（和密钥链接）。

[0069] 示例性解密过程在步骤 400 始于获得（例如，通过非信任数字接口）对加密的推定结果（即消息标识符（例如，临时数 N）、验证器 V 和包括段 $E_1, \ldots, E_N$的加密的结果 E）。在步骤 401，设备接着通过散列化接收的临时数 N 来计算值 H_1。注意，除非不正确地接收临时数，否则导出的 H_1将等于在加密过程中使用的 H_1。在步骤 402，解密设备通过散列化段 E_1（并且如果先前在加密期间使用则散列化向 H_2的导出中并入的关于消息的其它信息）来计算值 H_2。在步骤 403，设备尝试使用图 2 中描述的防泄漏的、基于密钥树的密钥导出过程以 $K_{START} = K_{ROOT}$并且 $PATH = H_1$来计算消息密钥 $K_{MESSAGE}$。在步骤 404，设备通过使用与加密设备相同的防泄漏的、基于密钥树的密钥导出过程（例如，使用密钥 $K_{START} = K_{MESSAGE}$并且 PATH

$= H_2$的图 2 中的过程）来计算预期验证器 V'。在步骤 405，将计算出的值 V' 与接收的验证器 V 进行比较。如果预期验证器 V' 未与提供的验证器 V 匹配，则该过程以错误终结（步骤 406），因为可能已经破坏或者恶意修改了提供的数据或者某一其它错误已经出现。

[0070] 如果在步骤 405 的校验成功，则该过程前进到步骤 407，其中将计数器 i 初始化成值 1、将密钥寄存器 K 初始化成计算 $g(K_{MESSAGE})$ 的结果（该结果是用于解密第一加密段 E1 的密钥（即在图 3 中标注为 302 的 K_1的值））。同样在步骤 407，将变量 H 初始化成 H_2。继而在如图 4 中所示的循环中执行以下操作。首先计算待解密的下一密文段的散列（即 $h(E_i)$）并且比较该散列与预期散列 H（步骤 408）。如果比较失败，则已经变更了加密段，因而该过程以错误终结 (409) 并且不执行进一步解密。如果比较在步骤 408 成功，则在步骤 410 利用密钥 K 使用解密函数 DEC() 来解密段 E_i以产生解密段，该段被解释为包含明文 D_i、继而为下一密文段的推定散列。将 H 设置成这一推定散列值。接着，在步骤 411，执行校验以查看是否已经解密了所有 L 段（即计数器 i 是否等于 L）。如果计数器尚未达到 L，则在步骤 412 递增计数器 i 并且通过计算 $K = g(K)$ 将寄存器 K 更新成用于下一段的解密密钥，并且从步骤 408 向前重复该过程。如果步骤 411 确定 i 已经达到 L，则在步骤 413 执行校验以查看 H 是否等于预期填充数据（例如，$D_1 \ldots D_L$的散列）。如果这一校验失败，则解密以失败状况结束 (414)。如果校验成功，则解密过程成功并且在步骤 415 返回恢复的解密的输出 $D = D_1 \ldots D_L$。

[0071] 注意，在这一实施例中，解密过程可以用流方式（即解密设备可以初始获得 N、V 和 E_1、继而一次一个地接收剩余段 $E_2, \ldots, E_L$）来完成，并且仍然能够实现上文概括的步骤。例如，如果解密设备缺乏用于保持整个消息的充分存储器，或者如果解密的数据的初始部分需要在已经接收和解密所有数据之前可用，则流操作是有用的。

[0072] 第二示例性实施例

[0073] 本节描述用于可验证防泄漏加密和解密的通用技术的第二示例性实施例。与使用密文散列链接的第一示例性实施例对照，第二示例性实施例使用明文散列链接。然而，在这两种情况下，在加密设备和解密设备二者控制密钥重用以防止差分功率分析和有关攻击。

[0074] 加密

[0075] 在为求简洁而描绘为组合的过程图和状态图的图 5 中示出了由加密设备进行的加密的第二示例性实施例。加密设备创建或者获得待加密的消息 D 和消息标识符 N（该标识符可以是计数器、随机生成的值、明文散列等）。

[0076] 将输入消息 D 划分成段 $D_1, \ldots, D_L$的序列（但是允许 $L = 1$），并且这些段用来创建明文段 $B_1, \ldots, B_L$，如下所述。首先，通过级联消息段 D_1与任何所需消息数据（表示为 X，该数据可以包括诸如长度 L、消息标识符 N、事务标识符或者计数器等元素）的散列来形成段 B_1(501)。接着，通过级联 D2 与 $h(B_1)$（即 B_1的散列）来形成 B_2(502)。继而，通过级联 D_i 与 B_{i-1}的散列来形成每个后续 B_i直至 B_{L-1}。最后，通过级联 D_L与 $h(B_{L-1})$ 来形成最后的明文段 B_L(504)。

[0077] 该过程的接下来的步骤 (505-508) 使用密钥链接过程来生成用于每个明文段的加密密钥，从而使得与第一示例性实施例相似，每个加密密钥直接或者间接地基于消息密钥。在第二示例性实施例中，将第一加密密钥 K_1简单地设置成通过使用如图 2 中描述的防泄漏的、基于密钥树的密钥导出过程以 $K_{START} = K_{ROOT}$和 PATH = h(N) 来计算 h(N) 并且继

而 $K_1 = K_{MESSAGE} = K_{ROOT,h(N)}$ 来导出 (505) 的消息密钥 $K_{MESSAGE}$ 的值。因此，第二密钥 K_2 是计算 $g(K_1)$ 的结果 (506)。重复这一过程，从而使得将第 L-1 个密钥 (K_{L-1}) 计算为 $g(K_{L-2})$ (507)，并且将最终段密钥 K_L 计算为 $g(K_{L-1})$ (508)。因此，每个密钥 K_i 基于消息密钥 $K_{MESSAGE}$（例如，等于 $K_{MESSAGE}$ 或者使用 $K_{MESSAGE}$ 来导出）。

[0078] 该过程中的下一步骤是用对应的密钥 $K_1, \ldots K_L$ 加密每个明文段 $B_1, \ldots, B_L$ 以产生加密段 $E_1, \ldots, E_L$。例如，通过用 K_1 加密 B_1 来创建加密段 E_1 (509)，通过用 K_2 加密 B_2 来创建 E_2 (510)，并且以此推论，以通过用 K_{L-1} 加密 B_{L-1} 来创建 E_{L-1} (511)，并且通过用 K_L 加密 B_L 来创建 E_L (512)。加密的结果 E 由段 $E_1, \ldots, E_L$ 构成。

[0079] 该过程中的下一步骤是计算用于加密的验证器 V(513)。首先，散列函数 h() 用来计算 $h(N||E_1||\ldots||E_L)||h(B_L))$，其中“||”表示级联。接着，计算 $Z = h(N||E_1||\ldots||E_L||h(B_L))$，继而使用防泄漏的、基于密钥树的密钥导出过程（例如，如图 2 中所描述的，以 $K_{START} = K_{ROOT}$ 和 PATH = Z) 来计算 $K_{ROOT,Z}$。继而，将验证器 V 计算为密钥树结果的散列（即 $h(K_{ROOT,Z})$）。最后，提供加密过程的结果，该结果包括 N、$h(B_L)$、E 和验证器 V(514)。

[0080] 可以在其中输入数据 D 通过流来到达或者其中出于其它原因而不能一次全部处理 D（例如，由于存储器限制）的系统中运用上述加密过程。在这一情况下，加密设备通过获得 N、h(X) 和 K_1 来开始。此外，用 N 初始化运行散列计算。

[0081] 1. 创建或者获得 N

[0082] 2. 初始化运行散列计算

[0083] 3. 令 H = h(X)

[0084] 4. 令 $K = K_{ROOT,h(N)}$

[0085] 5. 用 N 更新运行散列计算

[0086] 6. 令 i = 1

[0087] 7. 接收输入数据 Di（例如，流输入）

[0088] 8. 创建 $B_i = D_i$ 与 H 的级联

[0089] 9. 令 $H = h(B_i)$

[0090] 10. 创建 $E_i = ENC(K, D_i)$

[0091] 11. 用 E_i 更新运行散列计算

[0092] 12. 输出 E_i

[0093] 13. 递增 i

[0094] 14. 如果存在更多输入数据，则去往步骤 7

[0095] 15. 用 H 更新运行散列计算

[0096] 16. 完成运行散列计算并且在 Z 中存储

[0097] 17. 计算 $V = h(K_{ROOT,Z})$

[0098] 18. 输出 H（其等于 $h(B_L)$、N、V)

[0099] 解密

[0100] 在图 6 中图示了解密的过程。在步骤 600，解密设备接收（通常从非信任接口）接收加密过程的推定结果（即 E、$h(B_L)$、临时数 N 和验证器 V)。解密设备将 E 划分成 $E_1, \ldots, E_L$、将计数器 i 初始化成 1，并且将寄存器 H 设置成接收的值散列 $h(B_L)$。还接收或者确定消

息 L 的长度（例如，如果 1 千字节的段大小用于除了可以少于 1 千字节最后分段之外的所有分段，则 L 是消息以千字节为单位的向上取整的长度）。在步骤 605，解密设备计算 Z = h(N||E_1||...||E_L||H)，其中“||”表示级联。在步骤 (610)，解密设备使用图 2 中描述的防泄漏的、基于密钥树的密钥导出过程以根为 K_{START} = K_{ROOT} 和 PATH = Z 来计算 $K_{ROOT,Z}$ 的值，并且继而散列化结果以产生 h($K_{ROOT,Z}$)。在步骤 620，它比较计算的 h($K_{ROOT,Z}$) 与接收的验证器 V。如果结果不等于 V，则存在数据破坏并且该过程停止于 611 而不执行任何解密。如果校验成功，则在步骤 620，解密设备计算 h(N)，继而用如下计算的结果初始化密钥寄存器 K 并且将计数器 i 设置成 1，该计算使用图 2 中描述的防泄漏的、基于密钥树的密钥导出过程以 K_{START} = K_{ROOT} 和 PATH = h(N) 来计算 $K_{ROOT,h(N)}$。

[0101] 接着，在循环中执行以下操作：在步骤 630，用密钥寄存器 K 中的密钥解密段 E_i 以产生由数据段 $_i$ 和散列值构成的明文段 B_i。在步骤 640，校验来自解密的当前段的散列。对于第一段（即 i = 1），比较散列与 h(X)，其中 X 由与在加密期间的 X 相同的字段构成。对于在第一段之后的段（即 i > 1），比较来自 B_i 的散列与先前段的散列（即 h(B_{i-1})）。如果比较失败，则解密过程在步骤 641 失败。否则，在步骤 650，向输出缓冲器（例如，在 RAM 中）添加 B_i 的消息部分（即 D_i），并且通过计算 g(K)、继而在密钥寄存器 K 中存储结果来向下一段密钥推进 K。还将计数器 i 递增 1。在步骤 660，比较 i 的值与 L，并且如果 i 的值未超过 L，则解密过程循环回到步骤 630。否则，解密过程完成，并且在步骤 670，其中比较最后的明文段的散列（即 h(B_L)）与接收的散列 H。如果在步骤 670 的比较失败（即值不相等），则错误已经出现并且解密失败（步骤 671）。否则，在步骤 680 中输出结果数据 D_1,...,D_L。

[0102] 在这一实施例中，链接明文的散列而明文段 B_i 包含明文 B_{i-1} 的散列。这一链接尽管对于防泄漏而言并非严格必需，但是提供如下附加属性：可以检测在解密过程期间出现的任何故障，因为明文被验证为与加密的明文相同的明文。因此，这一实施例有利于在其中有可能破坏解密过程的环境中使用。

[0103] 系统、应用和变体

[0104] 到目前为止，本专利已经将一种用于防泄漏的加密和解密的通用技术与该技术的一些示例性实施例一起描述。这一节将描述其中可以利用前述内容的一些示例性系统和 / 或应用以及上文描述的示例性实施例的方面的附加变体。

[0105] 安全防火墙加载

[0106] 图 7 示出了将可验证、防泄漏密码术应用于在中央处理单元 (CPU) 上安全地加载敏感固件作为例如所谓的片上系统 (SoC) 的部分。为求便利，根据上下文，参考号可以指代过程中的步骤和 / 或由这样的过程步骤使用（或者产生）的量。在这一实施例中，SoC 由包含 CPU(703) 和各种类型的存储器的单个集成电路 (700) 构成。存储器可以包括而不限于随机存取存储器 (RAM)(701)（可以从该 RAM 执行代码）、只读存储器 (ROM)(704)（该 ROM 包含信任的自引导代码）和秘密状态存储存储器 (702)（该存储器保持共享的密码秘密 K_{ROOT}）。可以使用多种技术，比如而不限于熔断器 / 防熔断器、后备电池 RAM 和 EEPROM，来实施密钥存储储存器。SoC 可以具有可以从（例如，潜在地在对手的控制和 / 或观测之下的）非信任源接收功率的外部功率输入 (707)。也可以接收外部供应的时钟 (708)（并且该时钟可以与 PLL 一起用来形成附加时钟）。SoC 具有密码硬件部件 (705)，该部件具有用于数据加密和解密的 AES 引擎、散列函数引擎（比如而不限于 SHA-1 或者 SHA-256 或者基于 AES 的散

列函数引擎）和基于图 2 的防泄漏的、基于密钥树的密钥导出过程的实现方式，而使用散列函数和 / 或 AES 功能或者它们的变体来实施函数 $f_0()$,...,$f_{b-1}()$。本领域技术人员应当清楚，在其它实施例中，密码硬件部件 (705) 的全部功能或者其某一子集可以用软件（例如，由 CPU) 来执行。

[0107] 响应于从 ROM 中的信任的自引导代码自引导，SoC 从在这一实施例中为闪存 (709) 的外部非信任存储设备通过非信任接口 (706) 加载它的敏感软件 / 数据。为了保护敏感软件 / 数据以防公开或者未授权修改，由使用共享的秘密密码值 K_{ROOT} 的设备制造商或者其它代码签发者使用可验证的防泄漏技术（例如，如图 1 或者图 5 中所示）来加密它。加密结果存储于闪存 (709) 中。SoC 首先从闪存 (709) 向它的内部 RAM(701) 加载加密的代码 / 数据。它继而执行防泄漏解密（例如，如图 4 中所示），其中在存储于 ROM(704) 中的信任自引导代码中在密码硬件部件 (705) 中实施并且使用来自密钥库 (702) 的共享秘密密钥 K_{ROOT} 来执行该过程。如果成功，则这一过程在 RAM 存储器 (701) 内创建继而可以执行的验证和解密的敏感代码 / 数据映像。在解密过程失败的情况下，清除 RAM 中的加密的代码（数据）（和任何部分解密的代码 / 数据）并且操作在需要时从开始重启。

[0108] 在对这一实施例的可选增强中，通过在设备的将向其上加载软件的熔断器、后备电池存储器或者其它本地存储中存储最小可接收软件版本号来补充安全性。将向设备上加载的所有软件将携带版本号，并且设备将仅接受版本号大于最小值的软件。此外，一些软件版本可以具体指令 SoC 以更新最小可接受软件版本号，由此防止软件恶意回滚至视为不可接受的先前版本。可以独立于可验证的防泄漏操作（例如，作为其附件）实施前述防回滚方法。备选地，防回滚方法可以实施为消息标识符的部分、验证器或者在可验证的防泄漏操作中使用的其它安全量。

[0109] 本领域普通技术人员将容易认识到 SoC 应用并不限于这里呈现的具体架构，并且可以保护如下 SoC 或者其它设备，这些 SoC 或者其它设备具有不同内部架构和 / 或来自图 7 中呈现的实施例的部件。

[0110] 例如，图 8 示出了将可验证的防泄漏密码术应用于安全处理器架构 (800)。为求便利，根据上下文，参考号可以指代过程中的步骤和 / 或由这样的过程步骤使用（或者产生）的量。在这一设置中，设备包含 CPU、密钥库（该密钥库保持内部秘密状态（包括基本秘密密码密钥 K_{ROOT}））。可以运用非易失性存储（比如而不限于熔断器 (801)）以用于存储内部密码状态。密码硬件子部件 (804) 加密和 / 或完整性保护和 / 或回放保护从片上数据 / 指令高速缓存 (803) 向外部不安全 RAM 存储器 (806) 移出的所有数据，并且解密和 / 或完整性校验和 / 或回放校验从外部不安全 RAM 存储器取得的所有数据。此外，所有代码以加密和完整性保护的形式存储于不安全闪存 (805) 中并且在带入片上数据 / 指令高速缓存 (803) 中时被解密和完整性校验。背景技术的示例性处理器架构，其可以通过添加可验证的防泄漏密码术来提高其安全性，这些密码术包括而不限于来自 IBM 的 Secure Blue 设计（在 2006 年 4 月 6 日、标题为“IBM Extends Enhanced Data Security to Consumer Electronics Products”的 IBM 新闻发布中通告）和来自 MIT 的 AEGIS 设计（在第 17 届 Annual International Conference on Supercomputing 的 AEGIS:Architecture for Tamper-evident and Tamper-resistant Processing,Proceedings 的第 160-171 页 (2003) 中描述）。

[0111] 使用可验证的防泄漏密码术通过提供保护以防监视攻击来明显改善现有处理器设计的安全性。具体而言，这一实施例增强密码硬件子部件 (804) 以包括散列功能和密钥树处理能力，该能力重用现有安全处理器设计的（例如，AES) 加密能力并且实施第一示例性实施例的步骤和方法以创建安全防泄漏安全处理器。具体而言，使用防泄漏加密过程（例如，如图 1 中所示）和从非信任闪存 (805) 读取的任何代码来加密从高速缓存 (803) 向 RAM 存储器 (806) 写入的任何数据，并且使用图 4 中概括的防泄漏解密过程来解密非信任 RAM。当向特定段写入数据时，递增对应于段的计数器，并且在用于段的加密和 / 或完整性校验创建过程中并入计数器值、由此实现检测涉及到替换旧数据的攻击。

[0112] FPGA 位流加载

[0113] 将向现场可编程门阵列 (FPGA) 中加载的逻辑经常包含需要被保护以防公开或者复制的高度敏感商业秘密、密码秘密和 / 或其它敏感信息。通常从外部源（比如而不限于闪存设备或者 CPU 或者某一其它来源 (907)) 向 FPGA 供应这一加载的逻辑或者升级的逻辑作为位流。一些 FPGA 包含用于存储配置数据的非易失性存储器，而其它 FPGA 每当将芯片上电时必须重载。现有 FPGA 具有通常使用后备电池存储器中保持的或者本地存储（比如使用片上闪存、EEPROM 或者熔断器）的密钥来解密位流的能力。FPGA 在向存在于 FPGA 内的可编程分片中安装供应的加密的位流之前（或者之时）解密它。可以对位流解密过程尝试差分功率分析攻击和有关的外部监视攻击，从而引起严重的安全风险，因为成功攻击可以造成公开位流解密密钥和 / 或位流本身。

[0114] 参照图 9，可验证的防泄漏密码术可以用来在 FPGA 上创建安全位流解密能力。在解密之前，（使用软件、硬件或者其某一组合的）外部设备使用防泄漏加密过程（例如，如在第一示例性实施例中所描述的）来加密敏感位流从而产生加密位流。加密位流可以位于 (907) 非信任存储器（比如外部闪存或者硬盘驱动器）中或者从非信任源（比如 CPU 等）被取回。

[0115] 在 FPGA 内，用于防泄漏解密的密码秘密 K_{ROOT} 保持于密钥库 (902) 中，该密钥库存储内部秘密状态并且可以使用比如而不限于熔断器、后备电池 RAM(902，903)、EEPROM、闪存等技术来实施。FPGA(900) 通过接口 (906) 接收加密位流。例如，可能已经使用了第一实施例或者第二示例性实施例（对应于图 1 和图 5) 中的任一实施例来加密这一位流。

[0116] 如果图 1 的实施例用于加密，则 FPGA 首先接收临时数 N、验证器 V、长度 L 和初始段 E_1。E_1 存储于加密段缓冲器 (905) 中。使用如上文描述的防泄漏的解密过程（例如，见图 4) 来计算 E_1 的散列，并且用 K_{ROOT}、L 和散列来验证验证器 V，从而产生（如果成功）$K_{MESSAGE}$ 或者致命错误（在该情况下，该过程暂停）。如果成功，则 FPGA 使用段解密处理部件 (904) 以对 E_1 执行防泄漏解密过程。E_1 的解密产生被加载、验证和解密的段 E2 的散列。该过程一次继续一段，直至最后段被解密和验证。如果错误出现，则该过程暂停并且擦去所有部分 FPGA 解密数据。（响应于失败，该过程可以从开始再次重启。）一个或者多个状态寄存器 910 用于追踪位流加载过程的状态（例如，追踪过程是否是在进行中、失败或者完成）。也可以导出状态以用于诊断目的和用于由外部部件使用。一旦已经成功加载所有段，则现在配置并且可以使用 FPGA（例如，FPGA 现在可以允许 I/O、钟控等应用于加载的位流映像）。可以阻止 FPGA 操作直至完全加载位流（例如，以避免展现关于不完整的 FPGA 映像的信息并且以避免整个电路的由于不正确 FPGA 配置而发生的不可预测行为）。

[0117] 如果图 5 的第二实施例用于加密，则 FPGA 首先接收 E、V、N 和 $h(B_L)$，并且在缓冲器中存储 E。FPGA 的段解密处理部件 904 继而使用图 6 中描述的方法以验证和解密所提供的加密段。状态寄存器 (910) 用来跟踪位流加载、验证和解密过程的状态，并且任何严重错误造成暂停该过程和擦去任何部分解密的数据。

[0118] 网络通信和其它基于分组的应用

[0119] 图 10 示出了将可验证的防泄漏密码术应用于保护网络通信免受外部监视攻击。在这一实施例中，诸如设备 A(1000)、设备 B(1030) 和设备 C、D、E 等 (1040) 多个网络设备通过网络 (1020) 相互通信。这些通信中的一些或者所有通信可以包含敏感信息，从而使它对于加密和认证数据有用。另外，要求这些设备中的一些设备（比如在这一实施例中为设备 A) 保护它们的密码计算和密钥免受外部监视攻击。

[0120] 设备 A 具有密钥库 (1001)，该密钥库用于存储与它需要与之通信的其它设备的共享的密码根密钥的表。可能先前已经存储或者可以协商（例如，使用公共密钥密码术）这些密钥。用于使用公共密钥密码系统以协商密钥的方法在背景技术中公知并且在协议（比如 SSL 和 IPSEC) 中被利用。这一实施例可以容易地集成到这些或者其它协议中。

[0121] 待加密的出站分组或者数据段源于应用、操作系统、驱动器或者其它部件 (1002) 并且进入明文分组缓冲器 (1003)。继而使用段加密 / 解密处理部件 (1004) 来处理每个分组，其中使用可验证的防泄漏加密算法（例如，如图 1 中所描述）来加密它。用于这一加密的根密钥是从密钥库 (1001) 获得的在设备 A 与目的地设备之间的共享密钥。对于这一处理，消息标识符临时数 N 可以是包括计数器的任何（优选为）唯一值。例如，临时数可以等于分组标识符、TCP 序列号（该序列号有可能并入附加最高有效位以防止溢出）、值的散列、随机值等。对于每个分组，防泄漏加密操作产生加密段和验证器 V。临时数可以被传输或者可以是隐式的（例如，基于先前接收的分组数目）。将加密段、V 和任何其它所需数据组装成传出分组并且移向网络接口部件 (1006)，并且继而移向网络 (1020) 以用于向适当目的地设备寻路由。

[0122] 对于入站加密分组，假设发送设备已经执行了如上文描述的加密。这些分组由网络接口部件 (1006) 从网络 (1020) 接收，并且继而移向密文分组缓冲器 (1005)。每个分组继而由段加密 / 解密处理部件 (1004) 处理，其中执行防泄漏解密过程（例如，如图 4 中所描述的）。对于这一解密过程，(i) 从密钥库 (1001) 获得在接收与发送设备之间的共享密钥（例如，K_{ROOT} 或者用来导出 K_{ROOT} 的前身），(ii) 从分组恢复或者以别的方式确定临时数 N，(iii) 按照 N 和加密分组验证验证器，并且 (iv) 如果验证器正确则解密分组数据。在设备 A 与发送设备之间的共享密码密钥可以用作 K_{ROOT}。如果解密或者验证失败，则丢弃分组。否则，响应于成功解密，可以向应用、操作系统、驱动器等提供解密结果。

[0123] 在图 11 和图 12 中概括这一过程。图 11 图示了可验证分组级防泄漏加密过程，并且图 12 图示了对应的解密过程。可验证分组级防泄漏加密过程如下 :在给定输入分组数据 D(1100) 而源和目的地共享基本密码值 K_{ROOT} 时，在步骤 1101 中生成消息标识符 N（例如，使用随机源和 / 或存在于分组 D 中的信息和 / 或某一分组标识符（比如与通信协议相关联的序列号））。对于 TCP/IP 通信，可以根据会话标识符、序列号（可选地有附加最高有效位以防止翻转）、源端口、目的地端口和 / 或其它值。接着，在步骤 1102 中，计算 N 的散列。（可选地，可以省略这一步骤并且可以在导出 $K_{MESSAGE}$ 时使用 N 而不是 h(N)。）随后，在步骤 1103

中，使用图 2 中描述的防泄漏的基于密钥树的密钥导出过程以 $K_{START} = K_{ROOT}$ 和 PATH = h(N) 来计算消息密钥 $K_{MESSAGE} = K_{ROOT, h(N)}$。用密钥 $K_{MESSAGE}$ 加密输入分组数据 D 以产生加密的结果 E(1104)。

[0124] 接着，计算 E 的散列 (1105)（例如，使用 SHA-256）。继而，使用图 2 中概括的防泄漏的基于密钥树的密钥导出过程以 $K_{START} = K_{MESSAGE}$ 和 PATH = h(E) 将用于加密的验证器 V 计算为 $K_{MESSAGE, h(E)}$ (1106)。最后，形成输出分组以包括 V、E 和 N（或者为了使接收方能够恢复 N 而需要的任何其它信息（如果存在））(1107)。继而，在分组中向远程设备（比如通过因特网向远程计算机）传送输出数据 E。

[0125] 作为可选优化，如果加密设备具有为了发送而缓冲的多个分组，则它可以同时加密多个分组，从而使得仅需单个验证器以用于所有分组。例如，可以如图 3 中所示执行加密过程，其中每段 D_i 为分组。以这一方式组合分组减少发送方和接收方二者需要的密钥树操作的次数。

[0126] 在图 12 中图示了对应的可验证分组级防泄露解密过程。在给定包括 V、E、N（或者足以恢复 N 的数据（例如，序列号））的加密的分组和共享的密码秘密 K_{ROOT} (1200) 时，解密过程如下进行：首先，计算 h(N) 的值 (1201)（或者如果加密设备直接使用 N，则省略这一步骤）。继而，计算 E 的散列 (1202)。接着，在步骤 1203 使用图 2 中图解表示的防泄漏的基于密钥树的方式以 $K_{START} = K_{ROOT}$ 和 PATH = h(N) 来计算 $K_{MESSAGE} = K_{ROOT, h(N)}$。接着，使用图 2 中概括的防泄漏的基于密钥树的过程以 $K_{MESSAGE} = K_{ROOT}$ 和 PATH = h(E) 来计算 $V' = K_{MESSAGE, h(E)}$ (1204)。随后，解密设备校验是否 V' = V(1205)。如果它们不相等，则针对这一分组停止处理并且丢弃分组 (1206)。如果校验成功，则用 $K_{MESSAGE}$ 解密 E 以产生 D（明文分组）(1207)（例如，使用图 14 中所示的 DEC() 过程）。

[0127] 智能卡应用

[0128] 可验证的防泄漏加密和解密可以实施于智能卡中（例如，结合如下协议，在这些协议中要求智能卡以安全地免受差分功率分析和有关的外部监视攻击的方式执行加密和 / 或解密）。这样的系统和协议的示例包括而不限于导出用于解密付费电视信号、付款（包括离线付款）、身份验证 / 网络登入、移动电话 SIM 卡和过境护照的密钥（控制字）。在本专利中公开的示例性密码技术可以用来保证在执行这样的协议时保护智能卡内的秘密密钥免受外部监视攻击。比如，如果智能卡实施图 3 的基于密钥树的密钥导出过程，从而使得 K_{START} 从未需要离开智能卡，则智能卡（或者其它安全芯片）也可以用来实施在更大系统中利用的防泄漏加密或者解密过程中的部分或者所有过程。

[0129] 相互认证应用

[0130] 在许多应用中，两个或者更多设备需要相互认证和 / 或在它们之间交换敏感信息。这样的协议的示例应用包括而不限于：(i) 在打印机与墨盒之间认证以保证两个设备真实而非伪造；(ii) 在机顶盒与智能卡之间认证以保证部件真实（例如，以防止引入窃取的视频解密密钥）；(iii) 在车库门与打开者之间认证；(iv) 无密钥进入系统（比如可以在汽车中使用的系统），该系统认证密钥（例如，在将车门解锁或者启动引擎之前）；(v) 由被频繁窃取的物品（比如汽车无线电、GPS 单元、蜂窝电话等）执行认证协议以防止窃取或者操作篡改的设备；以及 (vi) 进入系统（比如在安全大楼中存在的在允许进入之前认证密钥 / 口令的系统）。在这些应用中，在设备之间的挑战响应协议已经在传统上用于相互认证以

及设置用于交换敏感信息的共享的秘密密钥。可以通过使用本专利的方法以执行任何所需加密操作或者解密操作来构造用于在防范 DPA 时执行这些认证的简单协议。例如，设备可以通过它的供应有效验证器和 / 或解密消息的能力、使用在本专利中公开的技术来论证它的真实性。

[0131] 具有段内密钥改变的段加密和解密

[0132] 这一节描述可以在实施示例性实施例（例如，如在图 3 的步骤 320、图 4 的步骤 410、图 5 的步骤 509、图 6 的步骤 630、图 11 的步骤 1104 和图 12 的步骤 1207 所示）时替代常规加密过程（比如在 ECB 或者 CBC 模式中的 AES）而使用的 ENC() 操作和 DEC() 操作的示例性变体。在图 13 和图 14 中分别所示的 ENC() 变体和 DEC() 变体中，为了甚至更大安全性而频繁改变密码密钥。具体而言，附加密码密钥更新在将数据段 D_i 加密成 E_i（或者反之亦然）时出现。因而，将这些变体称为实施段内密钥改变。

[0133] 除了对 ENC() 和 DEC() 的改变之外，可以如先前描述的那样实施第一示例性实施例和第二示例性实施例中的其余操作。例如而非限制，无需改变涉及到初始消息密钥 KMESSAGE、验证器 V 等的操作。

[0134] 图 13 示出了用于加密数据段的 ENC() 操作的示例性实施例。图 14 示出了 DEC() 操作的对应的示例性实施例。在这一实施例中，在密码块链接 (CBC) 模式中使用块密码 AES 来构建这些操作，但是本领域技术人员应当清楚也可以使用其它块密码或者加密 / 解密原语或者加密模式。

[0135] 向用于段 i 的加密过程的输入是段密钥 K_i(1301) 和数据段 D_i(1310)。将输入数据段 D_i(1310) 划分成子段 $D_{i,1}$(1311)、$D_{i,2}$(1312) 等。图 13 和图 14 示出了将数据段 D 划分成 3 个 AES 块的子段，但是也可以使用其它大小并且当然也可以运用除了 AES 之外的算法。（更小子段增加计算开销，而更大子段使密钥使用于更多操作中从而增加信息泄漏的可能性。）用散列操作 m() 变换段密钥 K_i 从而产生 $K_{i,1}$(1302)，该密钥是用于第一子段 $D_{i,1}$ 的密钥。如果将使用初始化矢量 (IV)(1314)，则将它与 $D_{i,1}$ 的第一 AES 块 XOR。（如果将不使用 IV，则可以省略这一 XOR 步骤。如果使用 IV，则可以例如通过向验证器计算中并入它或者通过根据经验证值（比如消息标识符）导出 IV 来认证它。）使用段密钥 $K_{i,1}$(1302) 用 AES(1315) 加密 (D_iXORIV) 的前多个位，从而形成密文子段 $E_{i,1}$(1320) 的第一部分。也将这一密文部分与子段 $D_{i,1}$(1311) 的接下来多个位 XOR，从而产生另一 AES 输入，随后使用段密钥 $K_{i,1}$(1302) 来加密该另一 AES 输入以产生子段 $D_{i,1}$(1311) 的下一部分。执行相似密码块连接操作以形成向也用密钥 $K_{i,1}$ 执行的第三 AES 加密的输入。三次 AES 操作的结果是密文子段 $E_{i,1}$(1320)。对下一数据子段 $D_{i,2}$(1312) 的第一块执行第四 AES 操作，并且使用通过将 m() 应用于 $K_{i,1}$(1302) 而导出的新密钥（特别地为 $K_{i,2}$(1303)）。来自处理 $D_{i,1}$ 的最后密文变成用于 $D_{i,2}$(1312) 的第一部分的 IV(1317)。加密过程继续，直至已经加密所有 s 个数据子段的所有块，从而最终产生加密子段 $E_{i,2}$(1321)，...，$E_{i,s}$(1322)，并且其中将 m() 用于每个子段来导出新密钥。最后，组装密文子段以形成最后的密文段 E_i(1330)。

[0136] 参照图 14，解密过程 DEC() 是 ENC() 过程的反过程。经由与用于上述加密相同的过程使用 m() 根据段密钥 K_i(1401) 导出子密钥 $K_{i,1}$(1402)、$K_{i,2}$(1403) 等。将加密段 Ei 划分成用子密钥解密的子段，每个子段包括一个或者多个 AES 输入。在每次解密操作之后，将适当 IV（如果存在）或者先前密文与数据 XOR。组装最后的数据以形成子段 (1420、1421、

1432 等），转而组装这些子段以形成 D_i(1430)。

[0137] 上述 ENC() 和 DEC() 过程是涉及到迅速密钥改变以便提供更大泄漏容忍度的示例。可以使用其它段加密和解密方法（包括在 ECB、CBC 或者计数器（例如，Galois 计数器）模式中应用流密码和 / 或块密码（比如 RC4、SEAL、AES、DES、三元 DES 等））。对于这样的操作（其中相同密钥应用于段中的所有数据），在加密之前限制每段的大小以便限制用每个密钥执行的操作的次数由此减少对手可以观测的用每个密钥执行的操作次数可以是有利的。

[0138] 通信信道

[0139] 可以用广泛的可能方式实现这里描述的数据交换。例如而不限于，常规总线 / 接口（比如 I2C、JTAG、PCI、串行 I/O（包括 USB）、PCI Express、以太网等）、无线协议（比如 802.11 系列、蓝牙、蜂窝电话协议、ISO14443 等）和芯片内连接（比如 APB、与其它触发电路的直接连接等）都可以被使用。对于每个前述方式，发送设备和接收设备将具有可以发送、接收或者发送和接收（视情况而定）的适当接口（例如，前述类型的接口）。

[0140] 在解密之前的数据验证的备选形式

[0141] 迄今为止呈现的示例性实施例已经利用防泄漏的基于密钥树的导出过程（例如，如图 2 中所示）以计算明文的可以在解密之前安全验证的验证器。尽管这一过程很好地适合于广泛应用，但是用于创建值的其它技术可以发挥相似作用，并且可以在某些设置中是足够的。例如，在一些实施例中，无需加密过程来防范外部监视（但是解密过程需要这样的防范）和 / 或可以存在用于公共密钥数字签署过程（比如在美国专利 6,304,658 中描述的过程）的算法级对策。对于这些系统，数字签署（数字签名）操作可以用来构造如下值，可以在解密时验证该值以保证未修改密文。例如，数字签名可以认证消息标识符和至少一个加密段。公共密钥数字签署算法的示例包括而不限于 RSA、DSA 和椭圆曲线 DSA 变体（包括而不限于 EC-DSA）。数字签名的验证无需任何敏感性信息并且因而可以在解密之前被执行。然而，这一灵活性换来的代价为需要加密设备内的公共密钥签署逻辑和解密设备内的公共密钥验证逻辑。也有可能让验证器（或者验证器替代物）包括多个对称验证器、公共密钥签名或者其它元素。

[0142] 非依序段密钥导出

[0143] 无需依序导出密钥段（例如，图 3 中的 $K_1, K_2, \ldots K_L$）和后续段密钥（图 13 中的 $K_{i,1}, K_{i,2}$ 等）。例如，可以在分级树模式中导出密钥，或者更一般地，每个密钥可以是任何先前密钥的函数或者可以使用密钥树构造根据 K_{ROOT} 来独立导出，或者可以使用其它密钥与密钥树构造的某一组合来导出密钥。

[0144] 数据传输和计算的重新排序

[0145] 可以变更数据传输和操作的排序。例如，图 1、图 3 和图 4 中描述的第一示例性实施例示出了从最后段 D_L 向第一段 D_1 进行的加密过程，其中每个段 D_i 包含第 i+1 段的加密结果 E_{i+1} 的散列。针对第一加密段 E_1 计算单独验证器（例如，见步骤 106）。这一方式可以如图 4 中所示有利于解密设备，因为它无需在解密之前缓冲整个加密结果，而加密设备必须这样做。

[0146] 备选地，加密设备可以加密始于 D_1 而结束于 D_L 的段，并且每个段 D_{i+1} 包含对前一段的加密 E_i 的散列。在这一示例中，段 D_1（例如）由大小与散列函数的输出长度相等的 0

的串扩展以指示它是第一段。继而使用 PATH = h(E_L) 来计算使用密钥树创建的验证器。对于这一变体，解密过程与图 4 相似、但是在从最后加密段到第一加密段的反方向上进行。因此，加密设备不再需要缓冲数据段，但是解密设备现在必须这样做。

[0147] 附加验证器替代散列

[0148] 虽然一些示例在数据段中示出了认证后续加密段的散列，但是后续段可以备选地携带它们自己的独立验证器。例如，图 3 示出了第一数据段 (312)，该数据段携带用于验证未改变段 E_2 的散列 h(E_2)。然而，并非总是需要并且在一些情况下可以省略这样的散列（例如，如果下一段代之以携带验证器）。这有些简化加密、但是增加计算时间，因为需要计算和校验更多验证器。在流应用中或者如果存储 / 存储器有限，考虑到避免需要让后续数据可用和缓冲这样的益处，附加计算工作变得合理。

[0149] 散列化中的变化

[0150] 在一些图中，多次应用单个操作（比如图 3 中的 h()）和 / 或将单个操作用于不同用途。一般并不要求这些都是相同函数。例如，不同步骤可以运用不同散列函数。

[0151] 散列函数的输出可以被截短、与其它散列函数输出组合或者以别的方式通过后处理来修改。例如，SHA-2 产生 256 位输出散列，但是可能希望更短消息标识符（比如 160、128、80 或者 64 位）。函数 h() 可以内部使用 SHA-2 并且仅返回它的结果的一些位。

[0152] 操作顺序的变化

[0153] 一些示例性实施例指明数据元被级联或者组合的具体顺序。例如，在图 3 的步骤 303-312 中，级联数据 D_i 与散列 h(E_{i+1})。其中在散列化之前依次级联数据段的其它示例包括图 5 的要素 501-504 和 513、在图 3 的步骤 306 中。这些具体排序仅为可能的排序的一个示例，并且可以在备选实施例中利用多个其它数据排序。

[0154] 基于树的密钥导出的变化

[0155] 如果操作（比如 f_i）可逆，则有可能使用除了树的顶部之外的值作为起始值。类似地，可以高速缓存计算的值（例如，如果消息标识符为计数器，则初始操作将通常不从一个消息到下一消息改变并且因此无需被重新计算）。

[0156] 错误检测和 / 或校正

[0157] 在本领域中公知的是由于在密码设备的操作中注入故障而产生的不正确输出可以产生关于敏感数据和密钥的信息。在实际时，可以校验密码操作以帮助防止释放可能危及秘密的不正确计算。例如，一种简单而有效的技术是理想地将两个（或者更多）独立硬件处理器和实现方式与比较器一起使用来两次执行密码操作以验证二者（或者全部）产生相同结果。如果由单元产生的结果不匹配，则比较器将防止任一结果被使用和 / 或触发其它错误条件。在个别密码操作（比如散列化步骤）内，也可以运用错误检测和 / 或错误校正逻辑以帮助防止或者检测其中不正确地执行密码操作的情形。

[0158] 在本专利中公开的技术还可以附加地针对某些类型的对加密过程和解密过程的故障注入攻击提供一些固有防范。在加密过程期间，在基于密钥树的密钥导出过程期间引入的有限或者部分错误将由于在这一过程内使用熵重新分布函数而产生随机的不可预测结果。具体而言，破坏的中介将通常由后续熵重新分布函数混合，这将限制对手发动如下攻击的能力，这些攻击利用缺陷结果。

[0159] 类似地，在解密期间，在密文或者消息标识符处理中引入的故障或者错误将一般

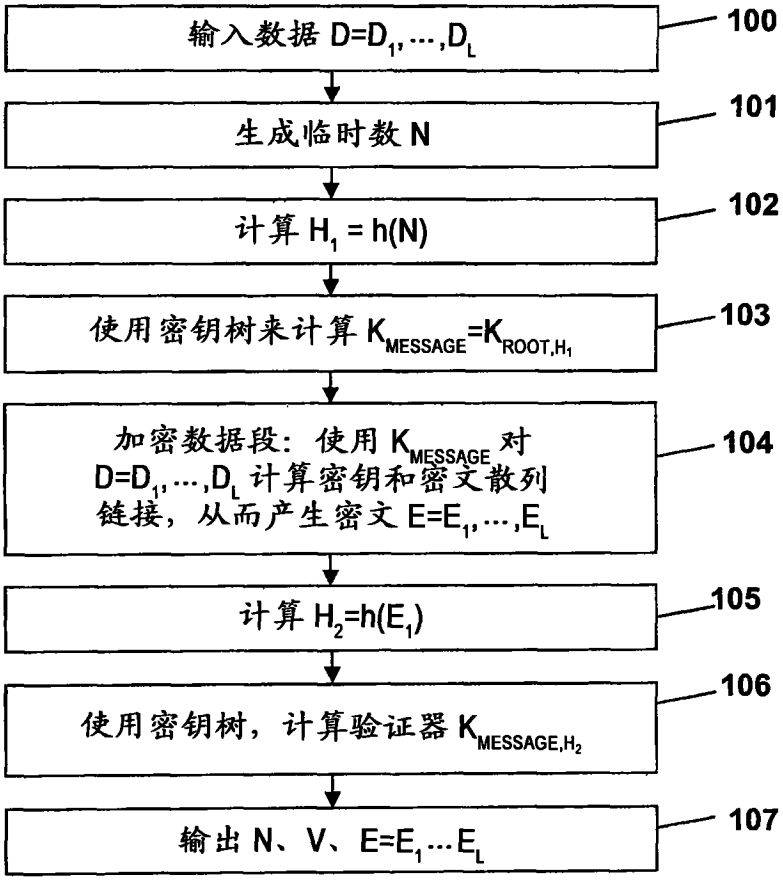
造成拒绝验证器。第二实施例用明文散列链接提供进一步防范，因为在输出之前独立认证明文段以求正确。当然，还可以附加地利用操作的校验和其它公知故障检测技术。

[0160] 也可以并入自诊断功能（比如 POST（上电自测）和随机数测试）以验证尚未毁坏密码功能和随机数生成能力。

[0161] 附加主机环境和形式因素

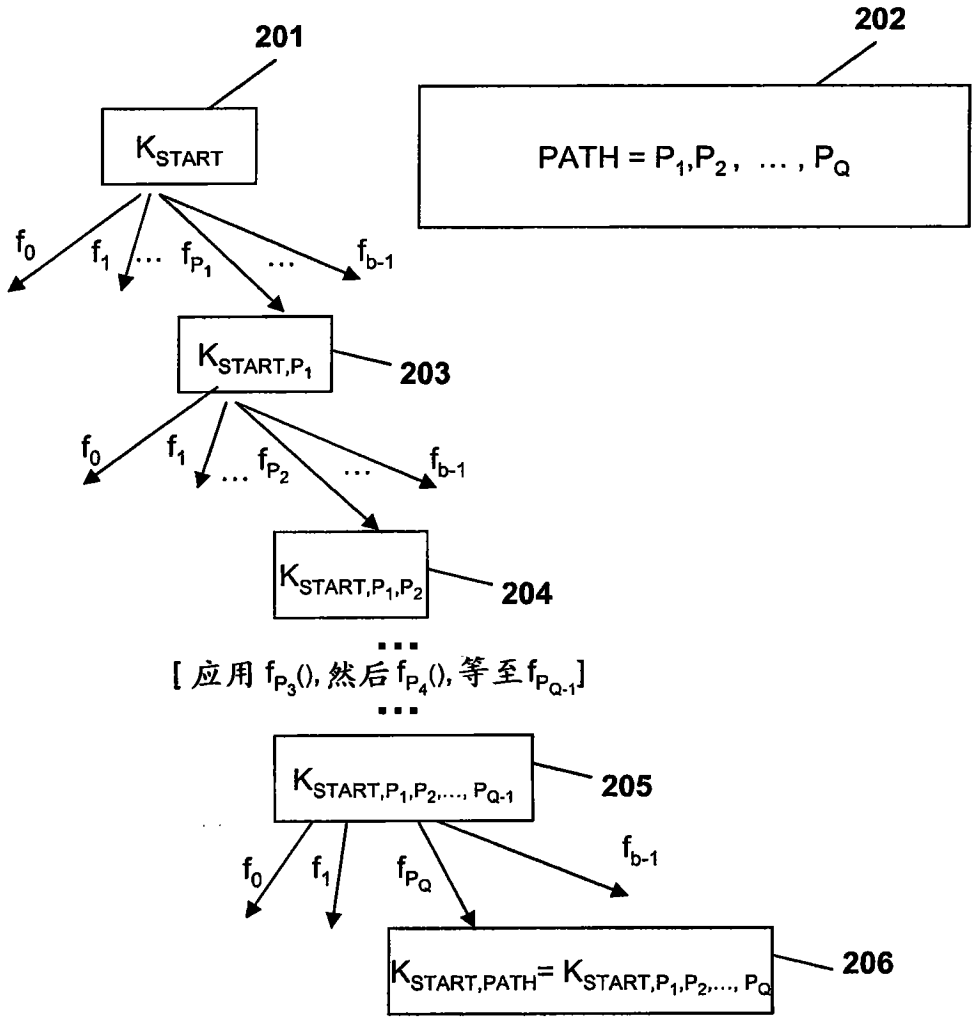
[0162] 上文描述了用于利用可验证的防泄漏密码术的若干示例性系统和应用。然而，如本领域技术人员将理解的那样，上文描述的技术并不限于特定主机环境或者形式因素。实际上，它们可以使用于广泛多种应用中，这些应用包括而不限于：专用集成电路 (ASIC)、现场可编程门阵列 (FPGA)、片上系统 (SoC)、微处理器、安全处理器、安全网络设备、所有种类的密码智能卡（包括而不限于基本上符合 ISO7816-1、ISO7816-2 和 ISO7816-3 的智能卡（“ISO7816 顺应智能卡”））；无接触和基于邻近度的智能卡和密码令牌（包括而不限于基本上符合 ISO14443 的智能卡）；存储值卡和系统；密码安全保护的信用卡和借记卡；客户忠诚卡和系统；密码认证的信用卡；密码加速器；赌博和下赌系统；安全密码芯片；防篡改微处理器；软件程序（包括而不限于用于在个人计算机、服务器等上使用的程序和可以向密码设备上加载或者在密码设备中嵌入的程序）；密钥管理设备；银行密钥管理系统；安全 web 服务器；防御系统；电子支付系统；微支付系统和计量器；预付费电话卡；密码标识卡和其它身份验证系统；用于电子资金转账的系统；自动柜员机；销售点终端；证书签发系统；电子证章；门户进入系统；使用密码钥匙的所有种类的物理锁；用于解密电视信号（包括而不限于广播电视、卫星电视和有线电视）的系统；用于解密加密的音乐和其它音频内容（包括通过计算机网络分发的音乐）的系统；用于保护所有种类的视频信号的系统；内容保护和复制保护系统（比如用来防止未授权复制或者使用电影、音频内容、计算机程序、视频游戏、图像、文本、数据库等的系统）；蜂窝电话加扰和认证系统（包括电话认证智能卡）；安全电话（包括用于这样的电话的密钥存储设备）；密码 PCMCIA 卡；便携密码令牌；以及密码数据审核系统。

[0163] 所有前述内容例示了可验证的防泄漏密码术的示例性实施例和应用，根据这些示例性实施例和应用，有关变化、增强和修改将在本公开内容的精神实质和范围的上下文内是明显的。因此，由本专利保护的发明不应限于前文公开内容，而是按照所附权利要求来解释。



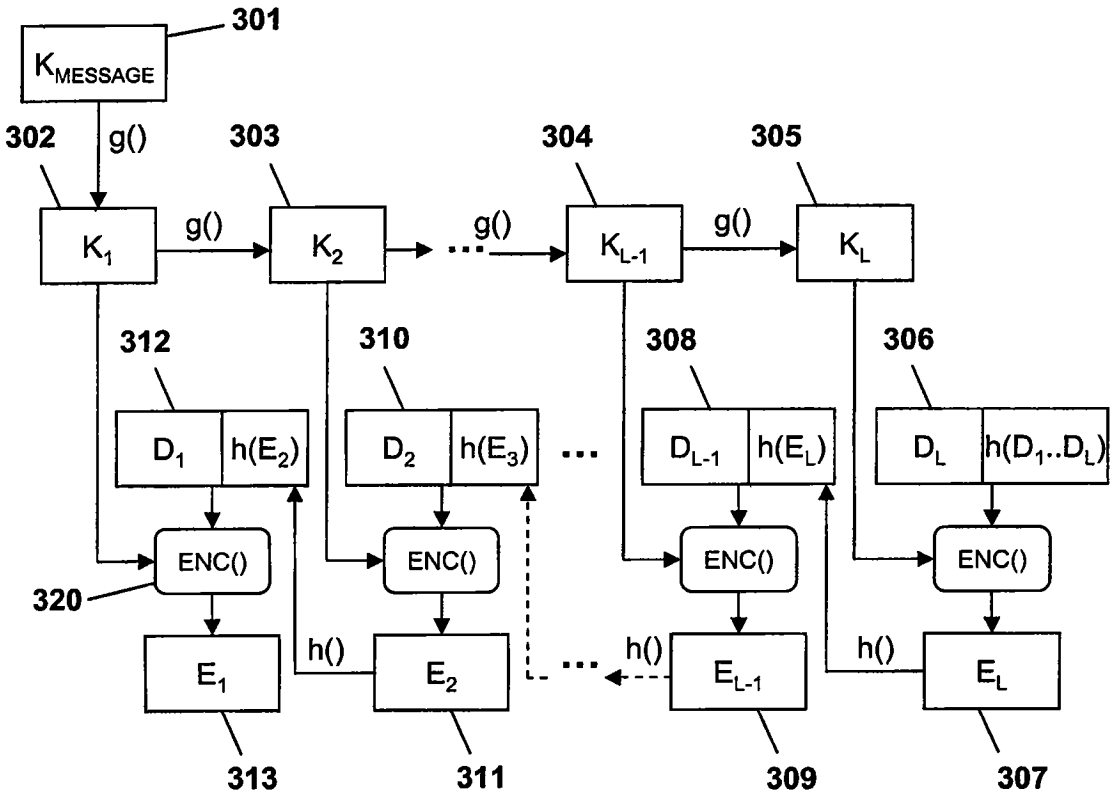
使用密文散列链接的可验证防泄漏加密

图 1



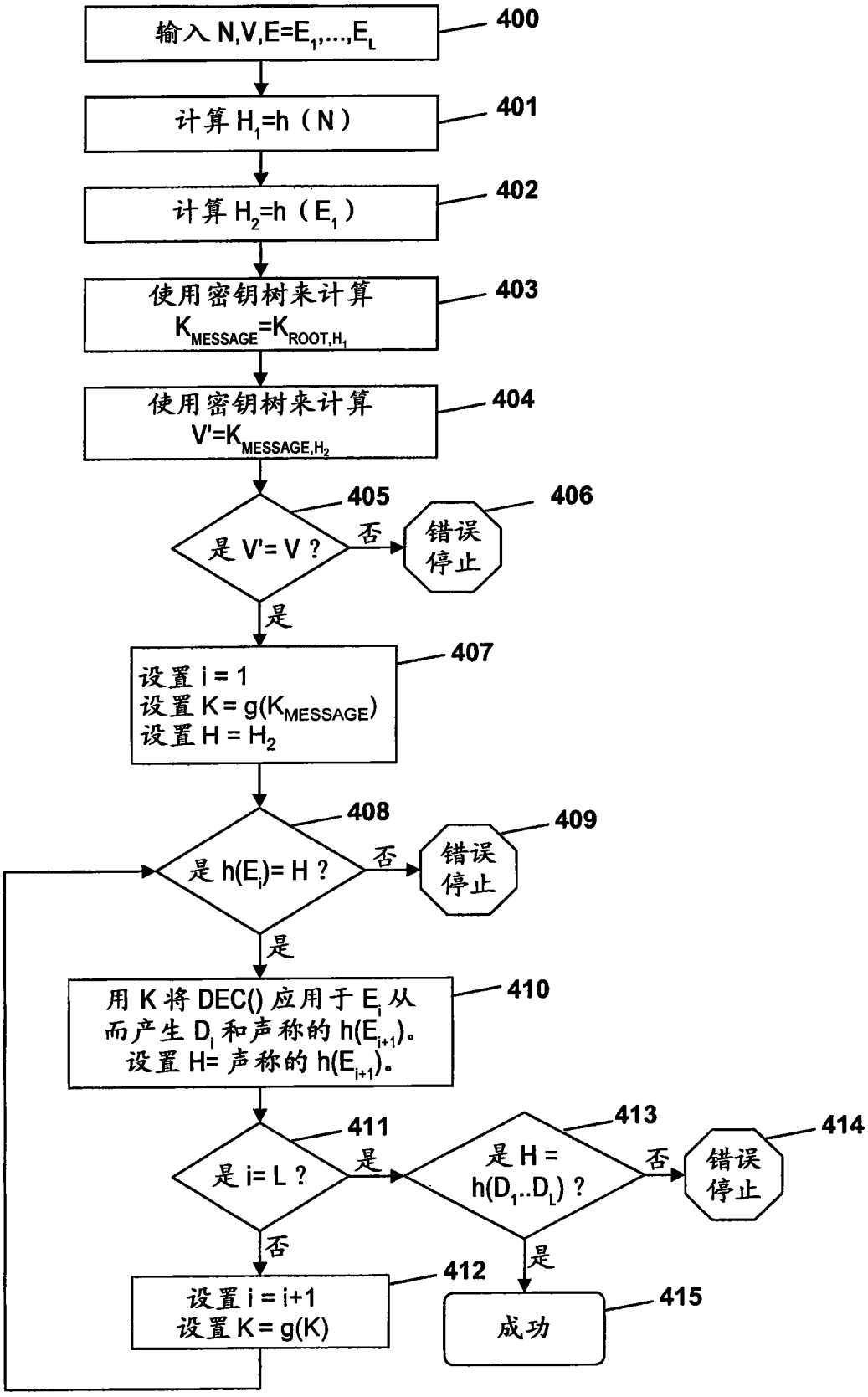
从密钥 K_{START} 起的防泄漏的基于密钥树的密钥导出过程

图 2



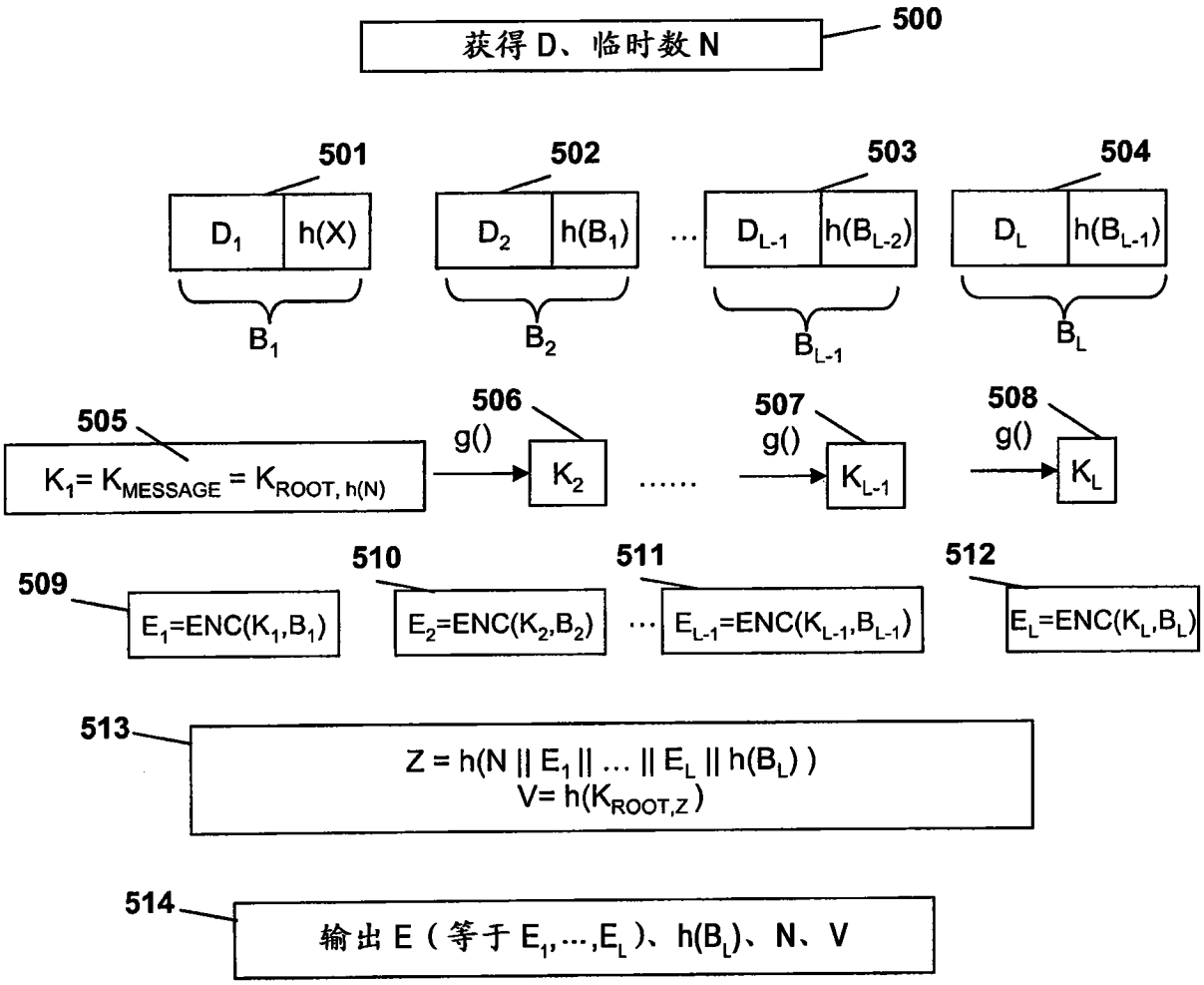
用于加密的防泄漏、密钥和密文散列链接过程

图 3



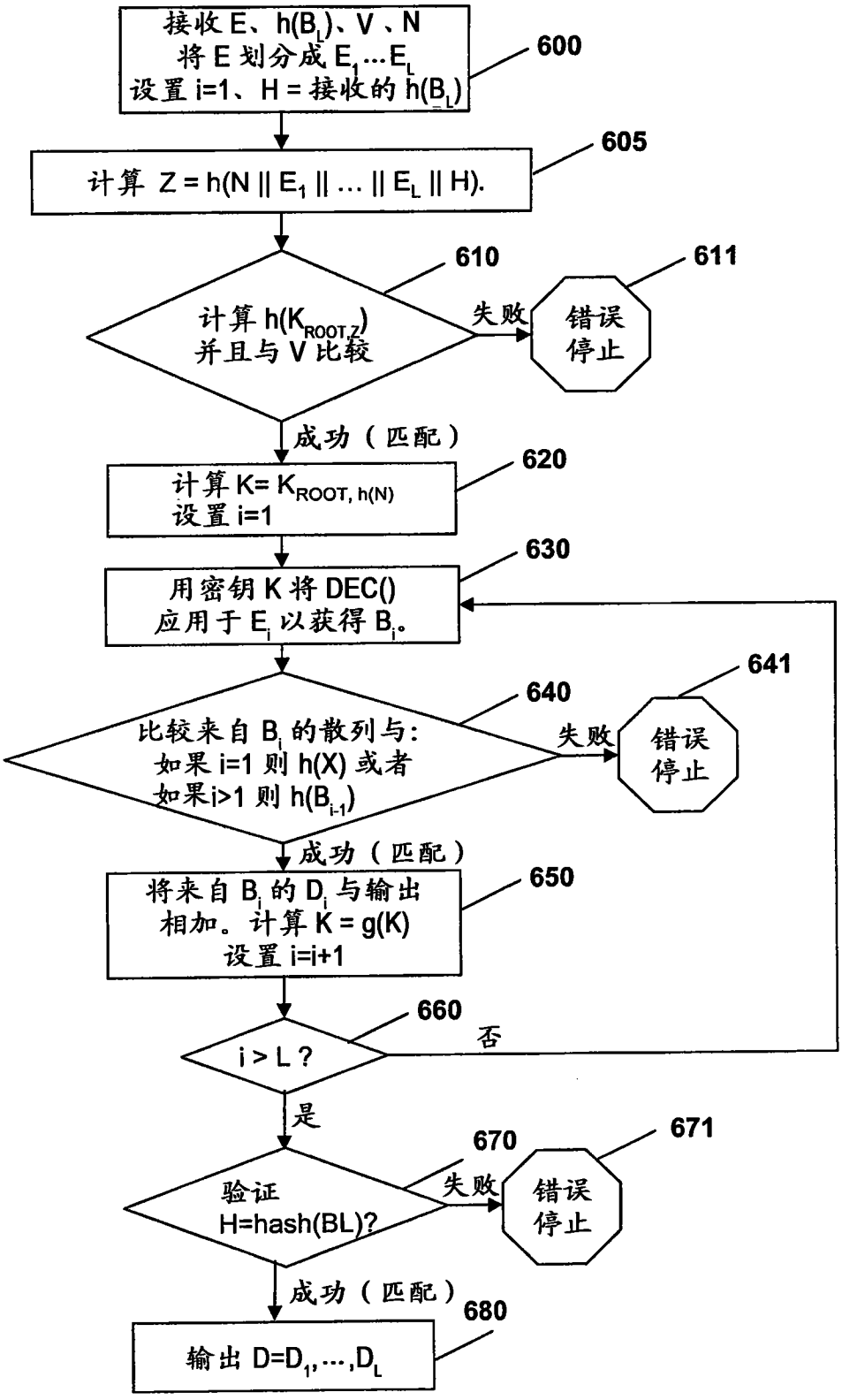
使用密文散列链接的可验证防泄漏解密

图 4



使用明文散列链接的可验证防泄漏加密

图 5



使用明文散列链接的可验证防泄漏解密

图 6

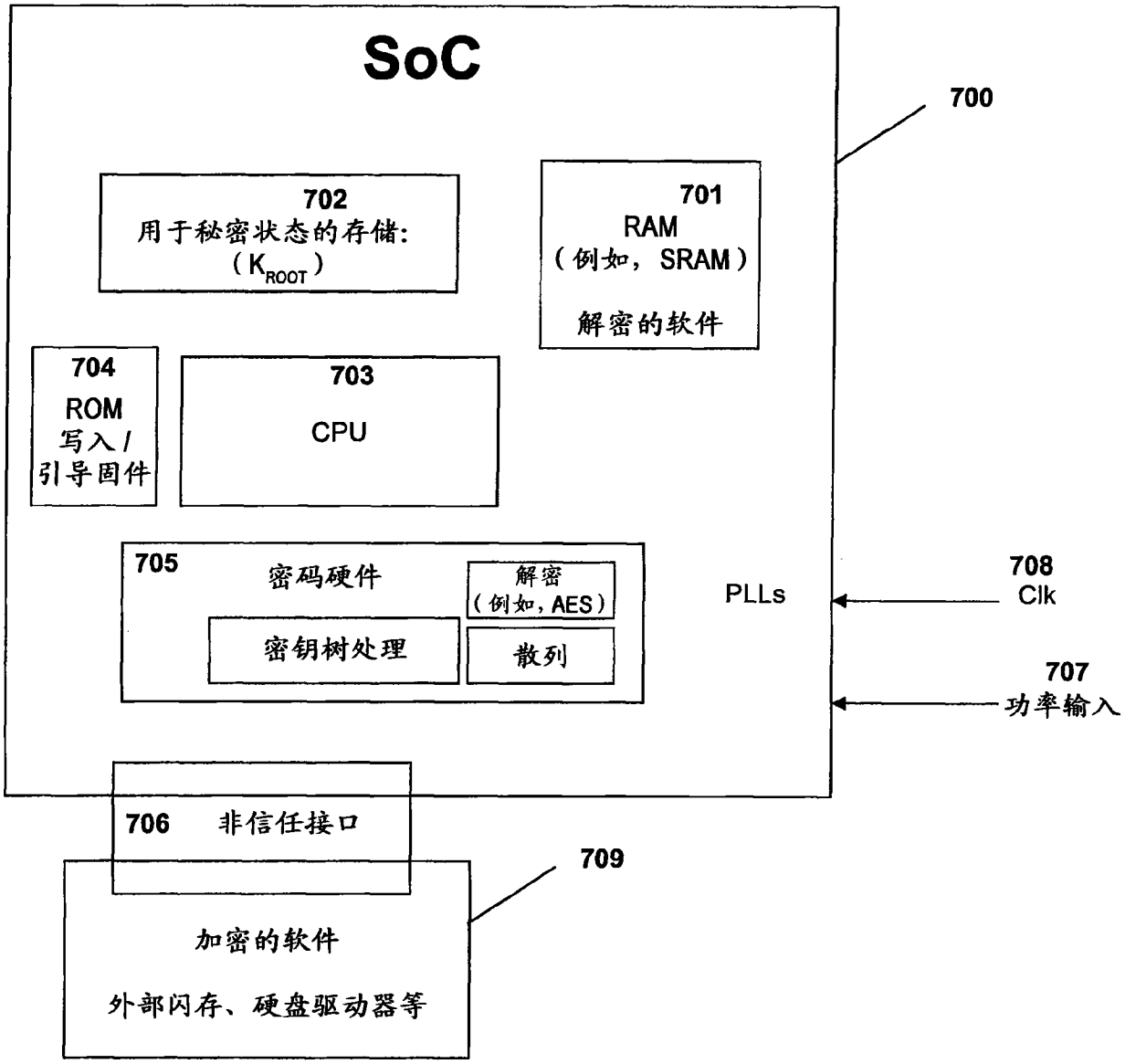

SoC
700
702
用于秘密状态的存储:
(K_{ROOT})
701
RAM
(例如，SRAM)
解密的软件
704
ROM
写入/
引导固件
703
CPU
705
密码硬件
解密
(例如，AES)
密钥树处理
散列
PLLs
708
Clk
707
功率输入
706
非信任接口
709
加密的软件
外部闪存、硬盘驱动器等
固件加载

图 7

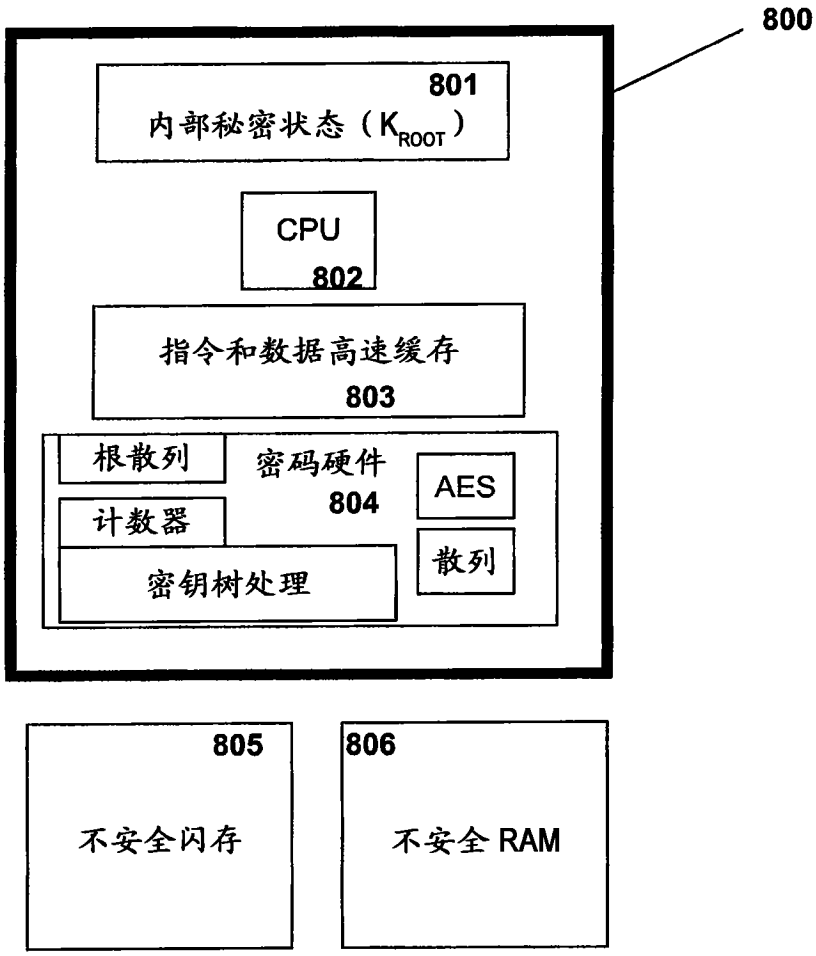

800
801
内部秘密状态（K_{ROOT}）
CPU
802
指令和数据高速缓存
803
根散列
密码硬件
804
AES
计数器
散列
密钥树处理
805
不安全闪存
806
不安全 RAM
安全 CPU

图 8

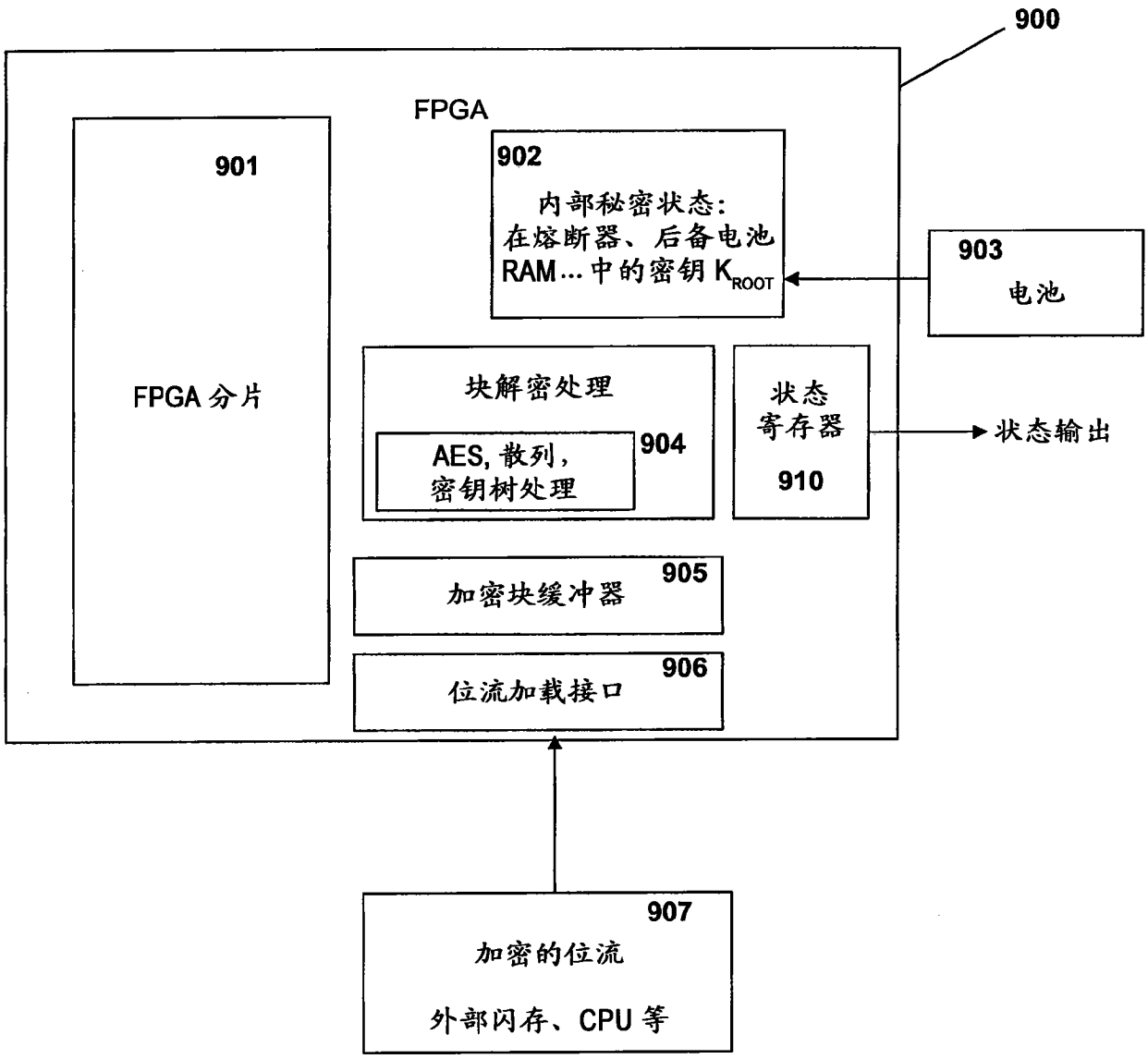

900
FPGA
901
FPGA 分片
902
内部秘密状态:
在熔断器、后备电池
RAM…中的密钥 K_{ROOT}
903
电池
块解密处理
AES, 散列,
密钥树处理
904
状态
寄存器
910
状态输出
905
加密块缓冲器
906
位流加载接口
907
加密的位流
外部闪存、CPU 等
FPGA 位流加载

图 9

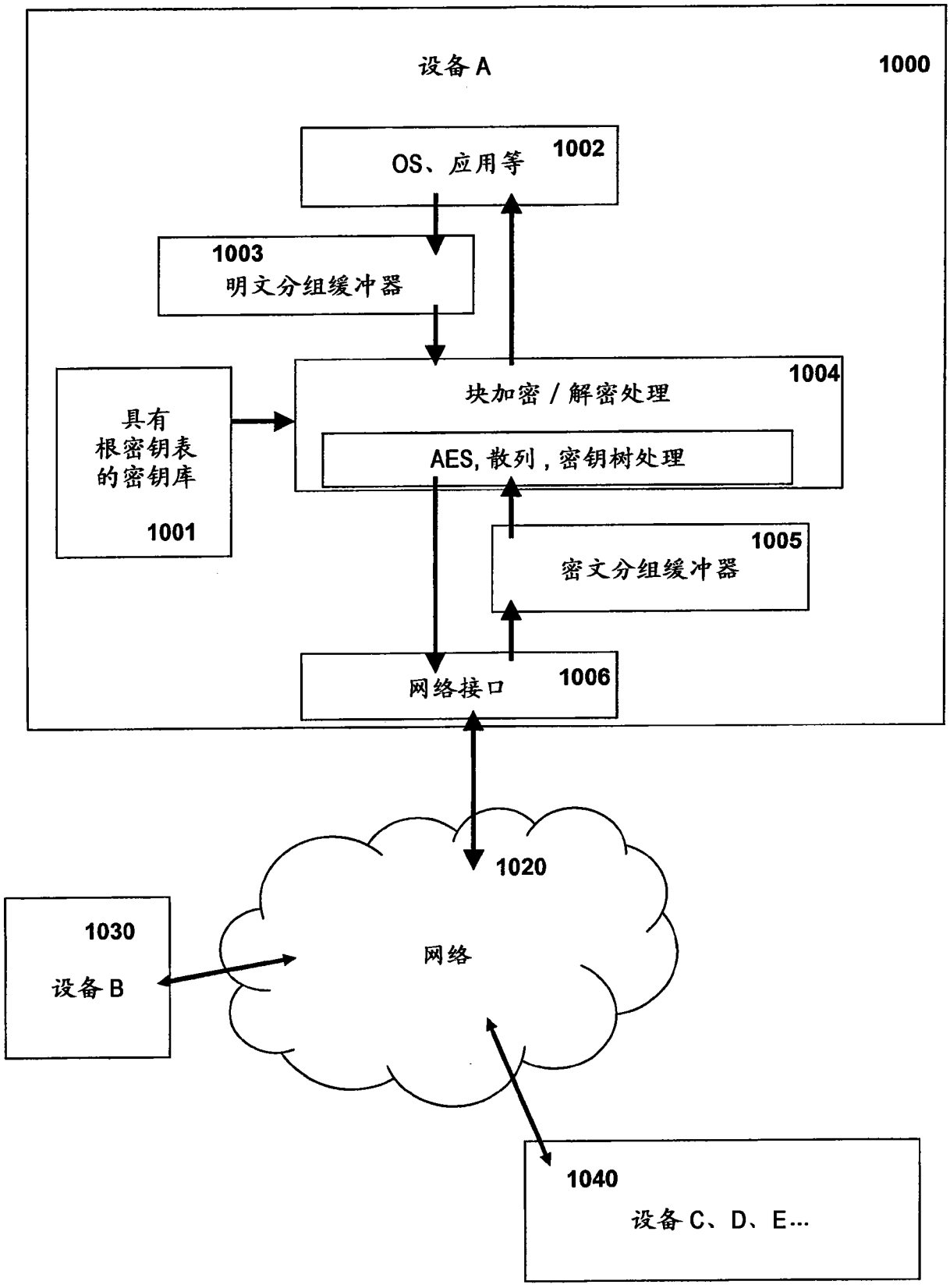
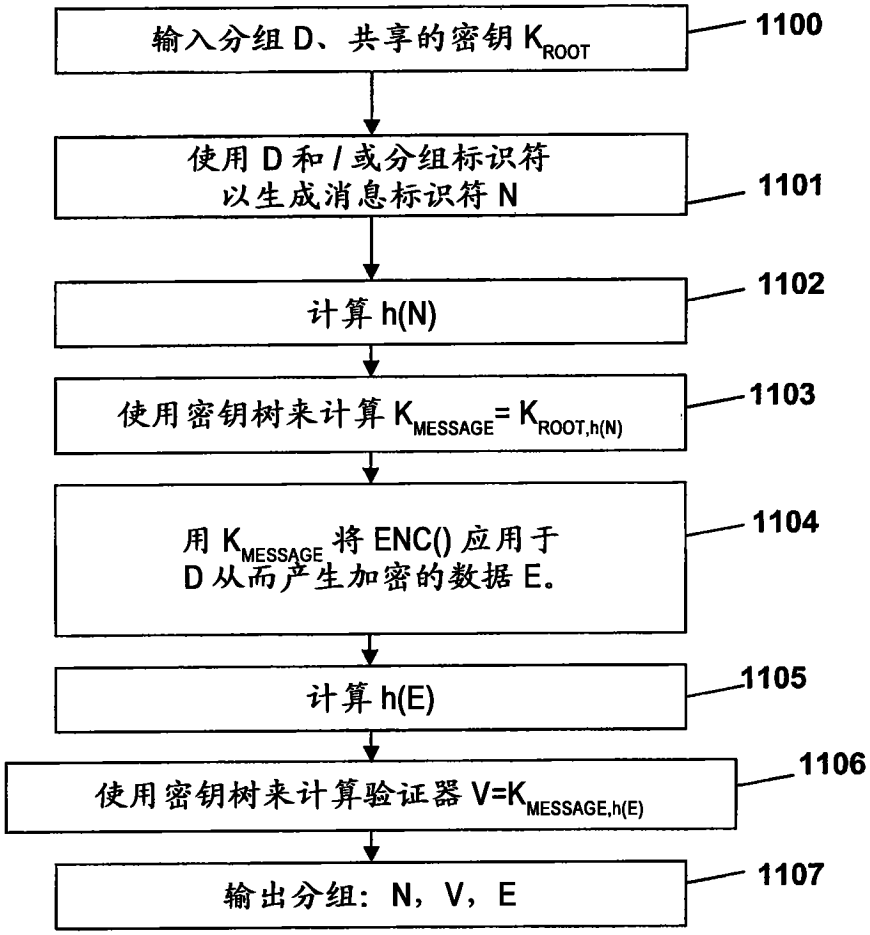
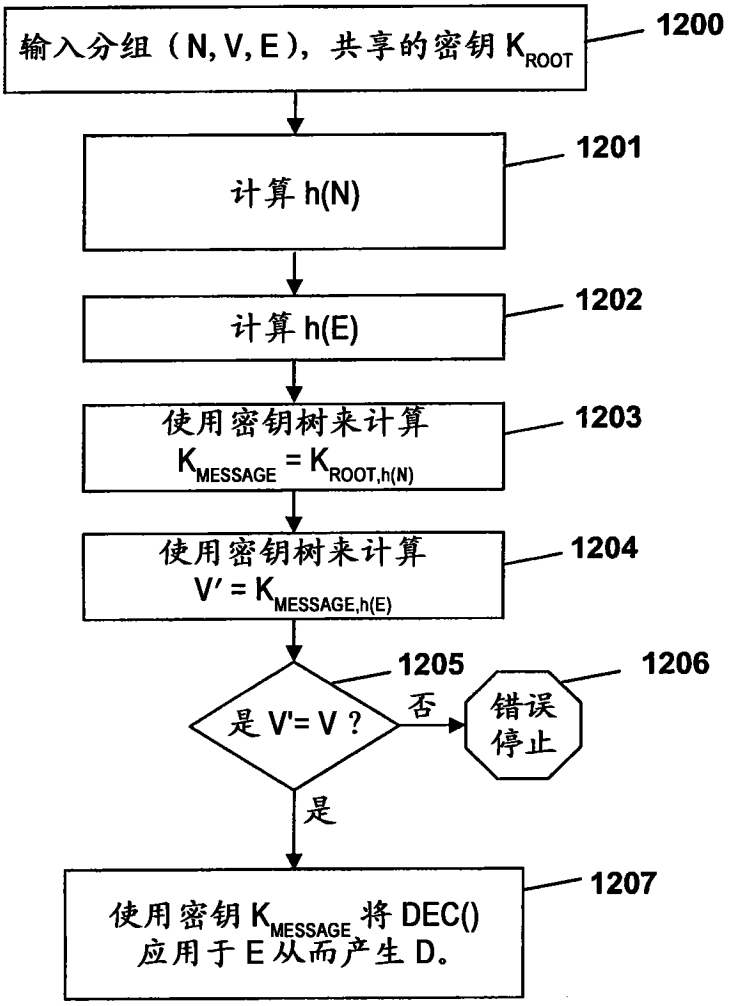

设备 A
1000
OS、应用等
1002
1003
明文分组缓冲器
具有
根密钥表
的密钥库
1001
1004
块加密 / 解密处理
AES, 散列 , 密钥树处理
1005
密文分组缓冲器
网络接口
1006
1020
网络
1030
设备 B
1040
设备 C、D、E…
网络通信设备

图 10



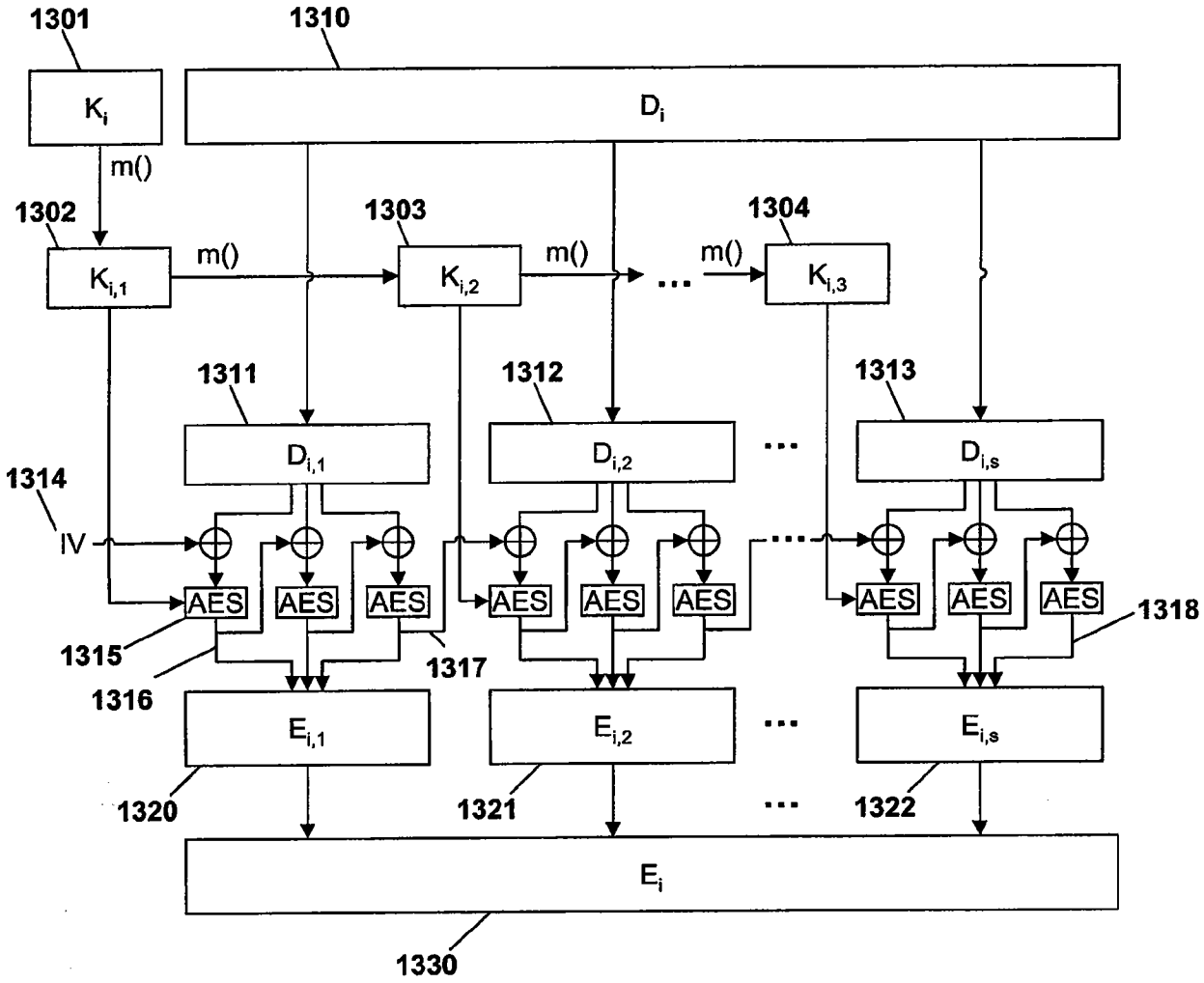
可验证分组级防泄漏加密

图 11



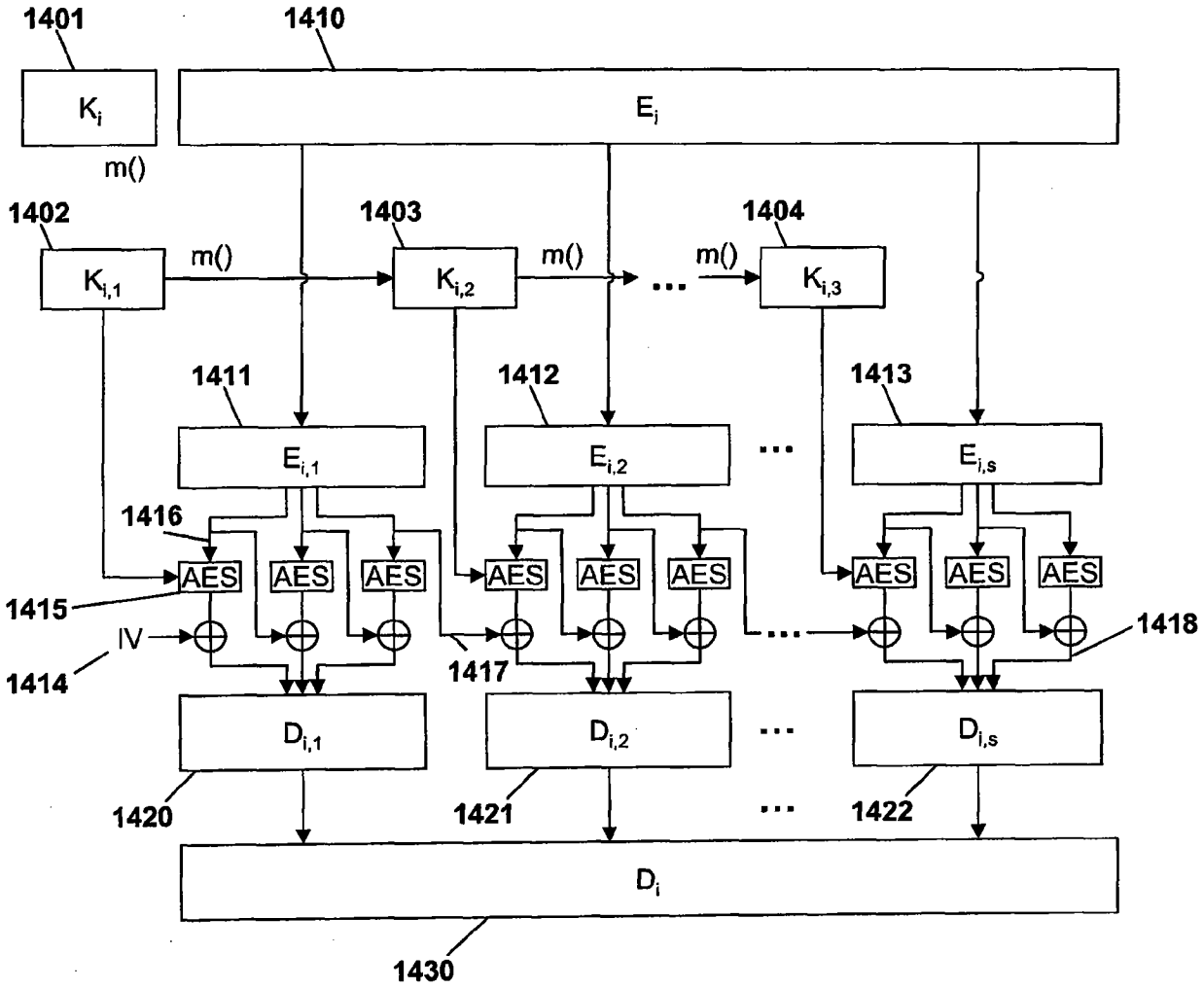
可验证分组级防泄漏解密

图 12



具有段内密钥改变的示例性 ENC() CBC 方法

图 13



具有段内密钥改变的示例性 DEC() CBC 方法

图 14