(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 110892673 B
(45)授权公告日 2023.10.27

(21)申请号 201880047043.1
(22)申请日 2018.08.13
(65)同一申请的已公布的文献号
申请公布号 CN 110892673 A
(43)申请公布日 2020.03.17
(30)优先权数据
62/547,669 2017.08.18 US
62/579,775 2017.10.31 US
62/622,515 2018.01.26 US
62/656,852 2018.04.12 US
62/669,906 2018.05.10 US
16/101,400 2018.08.10 US
(85)PCT国际申请进入国家阶段日
2020.01.14
(86)PCT国际申请的申请数据
PCT/US2018/046475 2018.08.13
(87)PCT国际申请的公布数据
WO2019/036356 EN 2019.02.21

(73)专利权人 乔纳蒂克斯公司
地址 美国加利福尼亚州
(72)发明人 吴英芳 R·J·内森
H·L·特雷丹尼克
(74)专利代理机构 北京市金杜律师事务所 11256
专利代理师 王茂华

(51)Int.Cl.
H04L 9/32(2006.01)
H04L 9/40(2022.01)
(56)对比文件
US 2015178143 A1,2015.06.25
US 9501664 B1,2016.11.22
CN 105279441 A,2016.01.27
US 2016110130 A1,2016.04.21
审查员 杨丽鲜

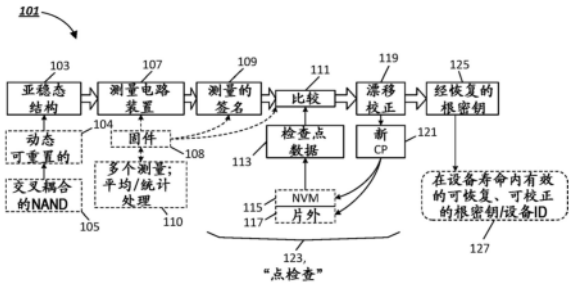
权利要求书3页 说明书52页 附图14页

(54)发明名称

安全硬件签名以及相关方法和应用

(57)摘要

本公开提供了一种用于从电路功能的测量中恢复根密钥的技术。在一些实施例中,使用检查点特征来周期性地标记该功能的测量,并且由此在数字设备的整个寿命中追踪根密钥的值的漂移;点检查特征以抵消递增漂移的方式允许功能的任何测量的回滚,并且允许在设备(例如,IC电路或嵌入了IC的产品)的寿命内恢复根密钥。本公开还提供新颖的PUF设计和应用。



CN 110892673 B

1.一种集成电路,包括:

电路处理器;

用于测量的电路装置,用于测量的所述电路装置在相应的时间测量电路阵列的性能状况,并且产生表示相应的所述测量的比特的相应集合;

用于创建检查点的电路装置,所述检查点表示在给定时间的所述集合中的一个集合中的所述比特的状态,并且用于创建检查点的所述电路装置将表示所述检查点的数据以加密形式存储在所述集成电路的外部存储装置中;以及

用于恢复密钥的电路装置,用于恢复所述密钥的所述电路装置被耦合到用于测量的所述电路装置,从其接收表示所述性能状况的新测量的比特的集合;

其中用于恢复所述密钥的所述电路装置从所述外部存储装置接收所述数据,并且迭代地修改表示所述新测量的所述比特的集合,直到与表示所述检查点的所述数据的比较指示经修改的所述比特的集合对应于在所述给定时间的所述比特的所述状态。

2.根据权利要求1所述的集成电路,其中所述集成电路还包括:用于在所述密钥被恢复时根据所述密钥对操作数执行加密和解密的电路装置。

3.根据权利要求1所述的集成电路,其中所述电路处理器是第一处理器,其中所述集成电路包括多个处理器,所述多个处理器包括所述第一处理器,并且其中所述密钥用于所述第一处理器的使用以排除所述多个处理器中的每个另外的处理器。

4.根据权利要求1所述的集成电路,其中:

所述相应集合中的每个相应集合中的所述比特的数目为至少512;

所述电路阵列包括与所述数目相等的量的亚稳态电路;

所述性能状况包括物理不可克隆函数;以及

用于测量的所述电路装置响应于由所述电路处理器选择性发出的信号,在所述集成电路的正常操作模式期间,动态地测量所述物理不可克隆函数。

5.根据权利要求4所述的集成电路,其中所述亚稳态电路中的每个亚稳态电路包括一对交叉耦合的NAND门,并且不被时钟控制。

6.根据权利要求1所述的集成电路,其中:

所述检查点是第一检查点;

用于创建所述检查点的所述电路装置创建多个检查点,所述多个检查点包括所述第一检查点和根密钥检查点,所述多个检查点中的每个检查点表示在所述给定时间的所述集合中的一个集合的所述比特的状态;

用于创建的所述电路装置将表示所述多个检查点中的每个检查点的相应数据以加密形式存储在所述集成电路的外部存储装置中;以及

用于恢复所述密钥的所述电路装置在迭代的基础上,从所述外部存储装置接收表示所述多个检查点中的每个检查点的所述数据,迭代地修改表示所述新测量的所述比特的集合,直到与表示所述多个检查点中的每个检查点的所述数据的比较连续地指示经修改的所述比特的集合对应于表示所述根密钥检查点的所述比特的所述状态。

7.根据权利要求6所述的集成电路,其中所述集成电路还包括用于根据所述密钥对操作数执行加密和解密的电路装置。

8.根据权利要求7所述的集成电路,其中所述密钥是根密钥,其中所述操作数是秘密密

码密钥，并且其中对所述操作数执行加密和解密的所述电路装置使用所述根密钥对用于外部存储装置的所述秘密密码密钥进行加密，从外部存储装置中取回所述秘密密码密钥的加密版本，并且使用所述根密钥对所述秘密密码密钥的所述加密版本进行解密。

9.根据权利要求6所述的集成电路，其中用于创建所述多个检查点的所述电路装置根据预定频率创建检查点。

10.根据权利要求6所述的集成电路，其中用于创建所述多个检查点的所述电路装置响应于经检测的状况来创建检查点。

11.根据权利要求1所述的集成电路，其中所述加密形式对应于证明曲线。

12.根据权利要求1所述的集成电路，其中所述加密形式是所述密钥的哈希。

13.根据权利要求1所述的集成电路，其中用于创建所述检查点的所述电路装置将在所述给定时间的所述集合中的一个集合中的所述比特分类为第一集合和第二集合，所述检查点表示在所述给定时间的所述集合中的一个集合中的所述比特的所述状态，并且其中用于创建所述检查点的所述电路装置通过使用所述第一集合中的所述比特作为加密密钥来对所述第二集合中的所述比特进行加密，以生成所述加密形式。

14.根据权利要求13所述的集成电路，其中用于恢复的所述电路装置从所述外部存储装置接收所述加密形式的所述数据，将表示所述新测量的所述比特的集合中的所述比特分类为第三集合和第四集合，并且通过使用所述第三集合的所述比特作为解密密钥对所述第二集合的所述比特进行解密，并且将经解密的所述比特与所述第二集合中的所述比特进行比较，来执行所述比较。

15.根据权利要求13所述的集成电路，其中：

用于创建所述检查点的所述电路装置使用多个不同的加密过程来生成所述加密形式的所述数据，所述检查点表示在所述给定时间的所述集合中的一个集合中的所述比特的所述状态，所述多个不同的加密过程包括：

第一过程，在所述第一过程中，用于创建所述检查点的所述电路装置将所述集合中的所述一个集合中的所述比特分类为所述第一集合和所述第二集合，其中用于创建的所述电路装置应用第一分类顺序将所述集合中的所述一个集合中的所述比特分类为所述第一集合和所述第二集合，以及

第二过程，在所述第二过程中，用于创建所述检查点的所述电路装置将所述集合中的所述一个集合中的所述比特分类为第三集合和第四集合，其中用于创建的所述电路装置应用第二分类顺序将所述集合中的所述一个集合中的所述比特分类为所述第三集合和所述第四集合；并且

用于创建所述检查点的所述电路装置使用所述第三集合中的所述比特作为加密密钥来对所述第四集合中的所述比特进行加密。

16.根据权利要求15所述的集成电路，其中用于创建所述检查点的所述电路装置屏蔽经加密的数据的至少一个比特，以创建被屏蔽的数据，并且其中所述集成电路将所述被屏蔽的数据存储在所述集成电路外部的非易失性存储器中。

17.根据权利要求13所述的集成电路，其中用于创建所述检查点的所述电路装置通过使用所述第一集合中的所述比特生成转置向量、比特翻转向量和非线性置换表中的至少一者，以生成所述加密形式，并且通过向所述第二集合中的所述比特应用所述转置向量、所述

比特翻转向量和所述非线性置换表中的所述至少一者来修改所述第二集合中的所述比特。

18.根据权利要求13所述的集成电路，其中用于创建所述检查点的所述电路装置通过使用所述第一集合中的所述比特生成转置向量、比特翻转向量和非线性置换表中的每一者，以生成所述加密形式，并且通过向所述第二集合中的所述比特应用所述转置向量、所述比特翻转向量和所述非线性置换表中的每一者来修改所述第二集合中的所述比特。

19.一种集成电路，包括：

电路处理器；

用于测量的电路装置，用于测量的所述电路装置在相应的时间测量电路阵列的性能状况、并且产生表示相应的所述测量的比特的相应集合；

用于创建检查点的电路装置，每个检查点表示在对应的所述相应时间的所述集合中的一个集合中的所述比特的状态，并且用于创建检查点的所述电路装置将表示所述检查点的数据以加密形式存储在所述集成电路的外部存储装置中；以及

用于恢复根密钥的电路装置，用于恢复所述根密钥的所述电路装置被耦合到用于测量的所述电路装置，从其接收表示所述性能状况的新测量的比特的集合；

其中所述检查点中的第一检查点表示标识所述根密钥的检查点；并且

其中用于恢复所述根密钥的所述电路装置针对检查点中的每个检查点从所述外部存储装置中接收所述数据，并且迭代地修改表示所述新测量的所述比特的集合，直到与表示所述检查点中的所述一个检查点的所述数据的比较指示经修改的所述比特的集合对应于所述检查点中的所述一个检查点，并且依次针对所述检查点中的相应检查点，重复所述迭代修改，直到所述比较指示经修改的所述比特的集合对应于所述根密钥。

安全硬件签名以及相关方法和应用

[0001] 相关申请的交叉引用

[0002] 本专利申请要求以下专利申请的优先权:以第一发明人Paul Ying-Fung Wu为代表的于2017年8月18日提交的针对“Secure Transactions With Chip Level Identity”的美国临时专利申请号62/547669;以第一发明人Paul Ying-Fung Wu为代表的于2017年10月31日提交的针对“Techniques For Secure Key Storage In Nonvolatile Memory”的美国临时专利申请号62/579775;以第一发明人Paul Ying-Fung Wu为代表的于2018年1月26日提交的针对“Real-Time CPU-Knowledge-Only Keyed Encryption”的美国临时专利申请号62/622515;以第一发明人Paul Ying-Fung为代表的于2018年4月12日提交的针对“Secure Embedded Shadow NVRAM&Volatile RAM,Encryption By Zero-Error PUF”的美国临时专利申请号62/656852;以第一发明人Paul Ying-Fung为代表的于2018年5月10日提交的针对“Nanosecond-Latency,Shallow Datapath Secured Embedded Shadow Memory With Zero-Error PUF(Physically Unclonable Function)”的美国临时申请号62/669906;和以第一发明人Paul Ying-Fung为代表的于2018年8月11日提交的针对“Secure Hardware Signature And Related Methods And Applications”的美国实用专利申请号16/101400。前述的实用和临时专利申请中的每一个均通过引用并入于此。该专利申请还通过引用并入以下文件:于2017年4月25日公开的针对“Encryption And Decryption Techniques Using Shuffle Function”的美国专利号9635011;以及以第一发明人Paul Ying-Fung为代表于2016年3月7日提交的、被公开为_____的、针对“Secure Communications Using Loop-Based Authentication Flow”的美国专利申请号15/063487。

背景技术

[0003] 存在许多应用,其中期望具有不受破坏的、唯一的、秘密的设备标识(ID)。作为其示例,例如,拷贝保护方案通常依赖于唯一的设备ID来提供对软件和内容的拷贝保护和追踪(例如,其中每个拷贝被单独地追踪和/或被单独地授权),并且唯一的数字ID通常被用于交换数字支付的加密技术中(例如,其中使用数字ID对令牌进行加密,然后令牌被交换为现金的替代物)。唯一的设备ID还可以用于为信用卡、蜂窝电话、计算机、启用Web的设备以及许多其他应用提供安全的过程和/或标识。

[0004] 通常,设备ID依赖于某种类型的安全秘密密钥。该密钥通常是复杂的(例如,长度为128比特或更长),并且用于生成经加密的序列,该经加密的序列可以被用于唯一地认证所讨论的设备;在大多数情况下,密钥不被外部地共享,即使与同一设备上的其他芯片(集成电路或IC)也不共享。即,为了防止密钥破解,密钥被嵌入单个IC内并且在单个IC内被使用是常见的,其中涉及该密钥的所有处理都被片上执行;这种架构有助于提供可以唯一地认证数字设备、甚至可以将其与属于同一用户的其他设备区分开的方案。常规地,密钥被存储在芯片上的某种类型的内部非易失性存储器(“NVM”)中,使得它不能被容易地侵入或破解。

[0005] 然而,随着技术的进步,出现了许多问题。首先,黑客变得越来越老练,并且越来越

多的攻击试图读取或发现所存储的密钥。其次，随着半导体行业继续受益于工艺尺寸的减小，实现更小并且更有能力的处理器(并且因此，智能设备)，创新的步伐倾向于超过存储器的能力，这使得难以在片上存储安全ID。作为示例，现在领先的CMOS技术可以实现7纳米的部件特征尺寸(“7nm工艺”技术)，而例如，NVM(诸如闪速存储器)目前仅使用大的多的65nm或55nm工艺技术是切合实际的(和可负担的)。这种差异抑制了较新工艺技术的能力；例如，作为实际的问题，不能使用7nm工艺技术来提供片上嵌入式非易失性存储器，而且这又意味着处理空间(例如，用于处理操作数、执行加密等目的的存储器)通常必须被片外(off-chip)地存储为明文(即未加密的)。因此，秘密密钥处于来自外部攻击的风险中，并且对于较新的工艺技术，该问题由于需要使用用于处理(包括用于密码操作)的外部存储器而加重。对于较新工艺技术的某些应用(例如FPGA和其他处理器)，对片外存储的需求意味着编程比特流也会经受拦截和利用，即，因为非易失性解密密钥实际上不能被存储在片上，所以无法在片上执行对经加密的程序流的解密。

[0006] 在解决这些问题上显示出希望的技术包括使用物理不可克隆函数(“PUF”)。秘密密钥通常不被存储在片上，而是该技术通常依赖于并且测量唯一的硬件工艺角(即，芯片与芯片间变化的、并且不能被容易地片外测量和检测的小的设计变化)；秘密密钥是易失性的，并且必须内部地重新测试或重新测量，以获得唯一的硬件“签名”，然后该“签名”至少在理论上可以用来唯一地标识或指纹识别设备。作为这方面的一个示例，许多电子电路设计(包括存储器单元设计)依赖于在上电后具有不可预测的状态的某种类型的双稳态电路，例如，通过电路装置“竞争条件”；当大量的这些电路被并行地观看作时，尽管由它们的一般设计引起的初始状态的结果看似不可预测，但是事实上，个体电路将倾向于可重复地假定可以被测量以提供唯一的设备签名的一致的输出状态(即，如果双稳态电路的数目足够大，并且状态测量过程足够鲁棒)。作为示例，对于具有数千个存储器单元的易失性存储器阵列，其中每个存储器单元都基于双稳态电路进行预测，该易失性存储器阵列理论上应该在上电时针对其每个存储器单元具有不可预测状态；然而，在实践中，由于不容易被外部检测到的芯片与芯片间的工艺变化，因此在上电时整个存储器阵列可以测量其存储器单元的状态，并将其用作唯一硬件签名的基础。因为该签名只能在芯片固件指定的时间被读取(即，诸如在上电时由系统BIOS读取)，并且从电路的外部检查中是无法预测的，因此攻击者不能容易地发现该签名。不幸的是，尽管PUF的使用是有希望的，尤其是针对较新的FPGA和其他处理器，但是这些设计也易于随时间被损坏(例如，个体双稳态单元操作可能会由于设备老化、或由于其他因素而随着时间慢慢改变它们的可预测性)，而导致设备签名的漂移；如果将该签名应用于加密和认证的目的，这将会是问题，即用于加密的根密钥的损坏意味着设备不能再被认证，而且加密/解密过程将不再工作。附加地，PUF函数通常不被在动态基础上使用(即，在大部分设计中，PUF只能在上电时被测量，并且因此，难以进行频繁的、重复的或动态的测量)。此外，常规的PUF通常依赖于差错校正码(ECC)(诸如降低差错概率的BCH码或Reed-Solomon码)，但是它们决不能确保差错的完全消除，这使得它们不适合用于生成和维持永久秘密密钥或安全设备ID的目的。

[0007] 因此，需要克服上述问题的技术。更特别地，需要用于获得安全密钥的技术，该密钥可以被用于以上提及的认证和/或加密函数；理想地，这些技术将提供“防损坏”方案，该方案始终允许恢复相同的原始硬件密钥(“根密钥”)，而与设备的老化、温度漂移和/或其他

因素无关。依然理想地，这种技术和相关联的密码操作理想地将与减小的特征尺寸(例如7nm和更小的工艺技术)相兼容，并且将在单个芯片内是可使用的，并且在动态的基础上是可使用的(例如，在任何时间)。最后，这种技术理想地将促进由使用较新的、较小的工艺技术构建的处理器可用的对编程或数据比特流的有效加密和解密，并且提供有效的、安全的设备ID管理。本文中呈现的技术解决了上述需求，并且提供另外的相关优点。

附图说明

[0008] 图1A是产生对损坏(例如，通过漂移和/或设备老化)鲁棒的唯一根密钥的设备的一个实施例的框图。

[0009] 图1B是描绘用于尽管在漂移(或其他类型的差错)的情况下恢复根密钥的技术的一个实施例的流程图。

[0010] 图1C是引入了“证明曲线”的使用的流程图，该“证明曲线”允许以递增或大批地移除漂移的方式对测量的硬件签名进行追踪漂移(或其他类型差错)和回滚，直到原始硬件签名(即，“根密钥”)被恢复。

[0011] 图1D是依赖于物理不可克隆函数(“PUF”)的测量以获得表示根密钥的硬件签名的集成电路(“IC”)的框图。

[0012] 图2A是示出了可以被用作PUF测量的基础的双稳态电路设计的电路图。

[0013] 图2B是示出了可以在同一时间被测量(例如，作为整行或阵列)的一组或多组双稳态电路的电路图。

[0014] 图2C是示出了具有提供动态可测量的PUF的双稳态阵列的IC的图示。

[0015] 图3A示出了具有实施本文所讨论的一些技术的处理器的IC。

[0016] 图3B是示出了用于测量和处理PUF以获得设备硬件签名的技术的一个实施例的流程图。

[0017] 图3C是示出了用于为所测量的硬件签名创建检查点的技术的一个实施例的流程图。

[0018] 图3D是示出了用于将硬件签名与一个或多个先前检查点进行比较的技术的一个实施例的流程图。

[0019] 图3E示出了使用示例数据可以解释过程的两个文本框。

[0020] 图4A是用于解释证明曲线的图示。

[0021] 图4B是用于解释混叠以及如何使用多个证明曲线来安全地标识先前检查点的图示。

[0022] 图4C是说明性的图示，用于解释硬件签名漂移以及如何使用证明曲线和检查点恢复来渐进地追踪以及允许消除漂移。

[0023] 图5是用于解释针对检查点管理的不同选项的流程图。

[0024] 图6是用于解释针对私密密钥(标识)存储、恢复和应用的一个或多个实施例的硬件图。

[0025] 图7是提供基于硬件的随机数生成的IC的框图。

[0026] 图8A是提供经加密的数据在外部非易失性存储器(“NVM”)上的存储的IC的框图。

[0027] 图8B是提供经加密的数据在外部易失性存储器(“NVM”)上的存储的IC的框图。

[0028] 图9A是示出了选择性地提供加密和解密的电路的一个实施例的说明性图示。

[0029] 图9B是示出基于非线性置换表和逻辑门的使用的加密/解密电路装置的图示。

[0030] 通过参考下面的详细描述，可以更好地理解由所枚举的权利要求限定的主题，应当结合附图来阅读该详细描述。以下阐述的一个或多个特定实施例的描述使人员能够构建和使用由权利要求阐述的技术的各种实施方式，但并不旨在限制所枚举的权利要求，而是例示其应用。在不限制前述内容的情况下，本公开提供了用于获得、管理和/或使用硬件签名的技术的数个不同示例。在一些实施例中，这些技术可以被实施为集成电路或其他装置、被实施为相关的方法、以存储在非暂态机器可读介质上的并且被适配为控制电路装置的软件(例如，固件)的形式被实施、和/或被实施为存储在非暂态机器可读介质上以用于制造特定设计的芯片的指令，其中该设计使得芯片可以执行如本文公开的方法。尽管呈现了特定示例，但是本文描述的原理还可以被应用于其他方法、设备和系统。

具体实施方式

[0031] 本公开提供用于以在设备寿命期间对漂移或其他形式的差错有弹性的方式来恢复易失性秘密密钥的技术、以及用于基于这种密钥生成安全设备身份并使用基于这种密钥的加密/解密过程的技术。

[0032] 在第一组可选实施例中，数字设备提供秘密密钥(在本文中被称为“根密钥”)，该密钥被存储在集成电路(“IC”)中或在集成电路(“IC”)内被测量。但是，该根密钥的读取可能会经受随时间的差错，包括通过非限制性示例的方式的漂移。附加地，由于底层电路的固有不稳定性和/或噪声，动态误差可能会在这种读取过程的任何时刻发生。为了尽管在有可能的差错的情况下提供根密钥恢复，该密钥的该值被读取。该读数在本文中被称为“硬件签名”，表示它可以具有或不具有与根密钥相同的值。然后硬件签名以如下方式被处理，以便于总是恢复到相同的原始根密钥。可选地，每个硬件签名(和原始根密钥)可以是测量值，例如，对物理不可克隆函数(或“PUF”)的测量响应，尽管这对于所有这些实施例是不需要的。相关联的签名(例如，以及与设备相关联的根密钥)是可选地大的，例如512比特、1024比特、2048比特，或者实际上，任何期望的长度(无论是否对应于2的幂)；在以下讨论的一个可选实施例中，使用的密钥长度为768比特。在一些实施例中，期望始终能够恢复原始根密钥，而不具有根密钥可以从其中被拦截或恢复的任何外部数据。为了获得这些目的，可以使用“点检查”过程来周期性地生成用于所测量的硬件签名的数据，这些数据稍后可用于验证新的签名测量，并且允许将新测量的硬件签名回滚到较早的状态。在一些实施例中，当存在漂移或其他差错时，使用允许电路逻辑通过处理当前硬件签名来“猜测”先前测量的硬件签名(以及最终，根密钥)的正确状态的过程，直到验证过程标识与较早的检查点硬件签名(或根密钥)的对应；该过程可以以最终将当前硬件签名迁移回到与原始根密钥相同的方式，在递增的基础上被逐个检查点地执行。注意，即使这也不是所有实施方式都需要的，例如，在一个实施例中，点检查过程的性质是这样的，使得可以选择性地执行回滚以直接恢复原始根密钥；但是，逐个检查点的过程可以更快，并且因此可选地在一些实施方式中是优选的。

[0033] 在一个实施方式中，点检查简单地是差错校正过程。但是，在其他实施方式中，由点检查过程存储的数据由加密函数产生，使得检查点数据本身不能被攻击者使用以恢复任何所测量的硬件签名或原始根密钥；在一个版本中，检查点数据表示根密钥本身的加密，而

在另一版本中，检查点数据包括值和随机混淆函数的加密(例如，将该值链路到根密钥)。这些加密过程允许检查点数据可选地被存储在相对于与根密钥相关联的设备或集成电路(“IC”)的外部。作为这些过程的实用性的简单的非限制性示例，IC可以具有提供嵌入式根密钥的电路装置；该密钥不在该IC的外部被外部地共享。该IC可选地是处理器IC。检查点数据(在一个可选实施例中被称为“证明曲线”数据)被存储在外部的非易失性存储器(或“NVM”)中；可选地这可以是在共同系统(诸如计算机、智能电话或其他数字设备)中的第二不同的IC。第一IC可选地使用较新的工艺技术(诸如7nm工艺技术或更小)来制造，或可以具有这样的性质：当其断电时，所有内部数据丢失。当它被重新供电时，为了确保获得适当的原始根密钥，检查点数据从第二IC中被读取。因为检查点数据本身不允许恢复任何先前检查点硬件签名或原始根密钥，因此秘密密钥(即根密钥)不受攻击者的攻击。处理器IC将取回的检查点数据和与新读取或测量的硬件签名相对应的新检查点数据进行比较，并且经由对新读取或测量的硬件签名的迭代修改以及对每个迭代修改的检查点数据进行的重新计算，得出与紧接地先前检查点对应的硬件签名。该过程持续直到恰好第一个检查点(与原始根密钥对应的)被确认。注意这是重复的，逐个检查点的过程是可选的；在备选实施例中，通过由“检查点密钥”直接加密所述原始根密钥的方式，经由总累积检查点的任何子集，可以从所有累积检查点中的任何检查点直接恢复原始根密钥。在其他可选实施例中，然后，根密钥被用作秘密密钥以允许解密外部存储的、但经加密的私密密钥或其他秘密密钥，该密钥将被直接用于加密/解密过程(诸如通过非限制性示例的方式，基于设备ID的涉及数字设备的外部事务)。注意，这不是所有实施例都需要的。在一些实施方式中，根密钥还可以被用于对其他外部存储的数据进行加密/解密，诸如(通过非限制性示例的方式)编程比特流或软件/固件版本升级或用于双向相互认证；作为示例，根密钥可以被用于处理根密钥的设备之间的加密/认证。在可选实施例中，点检查过程(以及检查点数据的关联存储，诸如证明曲线数据)根据日历或特定地(ad hoc)执行，但足够频繁使得相邻检查点之间的较差情况的漂移为九比特或更少；在其他实施例中，该值可以低得多(诸如五比特或更少、三比特或更少、一比特或更少等)。

[0034] 在第二组可选实施例中，一组电路结构被用于促进可测量的物理不可克隆函数(“PUF”)。这些电路结构可以基于亚稳态电路设计，该亚稳态电路可以在集成电路(“IC”)的操作期间被动态地重置，而无需将功率循环到整个芯片或电路装置的阵列(例如，诸如整个存储器阵列)。在一个非限制性示例中，这种亚稳态电路可以被实现为一对交叉耦合的NAND门，其中对于每个NAND门，第一输入被耦合到另一NAND门的输出，并且其中针对两个NAND门中的任何一个NAND门的第二输入被连结在一起，并且用于动态重置(即“激励”)电路；在备选实施例中，还可以使用一对交叉耦合的NOR门。这种设计创建双稳态电路，其中看似随机的状态可以在任何期望的时刻被测量、动态重置以及重新测量。交叉耦合的NAND门在同一设备上的多个激励实例上，甚至在相同制造的晶片上的多个设备实例上，在输出状态中都呈现看似随机的特性，由于IC制造工艺自然赋予电路的变化的寄生和内在行为(例如晶体管驱动强度和增益、寄生电阻和电容、阈值电压、线路的电阻性/电容性贡献、和来自相邻电路的噪声耦合。此外，在设备的操作寿命期间，操作漂移(诸如负偏置温度不稳定性(“NBTI”)和正偏置温度不稳定性(“PBTI”))可能会驱动该电路的个体化“老化”行为进一步发散。测量任务可以在硬件或固件的控制下被执行，而无需去除电路功率，可选地在每个亚

稳态单元或一组亚稳态单元的个体基础上。对于许多实施例，该特定亚稳态电路设计是可选的，并且在其他实施例中，而是可以使用其他亚稳态电路设计(例如，基于交叉耦合的锁存器、反相器和/或其他电路元件)。在特定实施例组中，该组电路结构被可选地配置为阵列，使得阵列中电路结构的子集可以个体地被即时读取。可选地，这些第二实施例中的一组电路结构可以被用于提供如上所述的硬件签名;硬件签名可以根据需要被重复地测量，所有这些都不需要循环功率，其中得到的测量结果被平均化，或者经受增强相关联的硬件签名的可靠性和/或安全性的统计处理。在另外的(可选的)实施例中，这种一组电路结构可以用于随机数生成，和/或提供根密钥。在更详细的实施方式实施例中，可选地，这些结构可以与上面讨论的第一组实施例组合，并且作为根密钥和相关联的硬件签名测量以及检查点恢复过程的基础(如较早介绍的)。

[0035] 在第三组可选实施例中，唯一的设备ID可以基于硬件签名和/或相关联的根密钥(即，无论是否使用或基于以上讨论的与前两组实施例相关的其他选出的特征)被生成和应用。例如，根密钥(可选地，如上所述)可以被用于生成设备ID或其他数据，这些数据可以被共享或在外部存储，但是不能从这些数据中推断出硬件签名或密钥。在一个实施例中，设备ID可以是秘密密钥或私密密钥的形式(即，对应于相关联的公共密钥，其中后者可以是设备ID的可选发布的部分)。设备ID可以可选地被用于软件过程，例如，内容拷贝追踪和保护、令牌生成、认证和加密。根密钥可以被用于生成和/或加密私密密钥，然后其被外部存储在NVM中(本地或远程);如果由半导体设计支持，它还可以被嵌入在设备(即，IC)中，并且以一次性可编程或多次可编程的非易失性存储器的形式可用。与允许动态取回或动态测量正确根密钥的过程(如刚刚例示的)相结合，外部存储的、经加密的秘密密钥或私密密钥始终可以被安全地由硬件存储、恢复和解密，并且在需要时被应用;硬件可以以免于拦截的方式外部地存储私密密钥，并且始终可以使用不能被外部地看到的内部秘密密钥(例如，根密钥)来解密经远程存储的密钥。唯一的设备ID的可选应用还可以包括促进安全的外部存储器(包括但不限于影子RAM或影子NVRAM和/或对经加密的处理器操作数、数据和编程比特流的外部存储，以及根据需求的现场软件/固件更新)。换言之，一些常规过程(包括但不限于密码操作)有时要求处理器具有板载RAM或NVRAM，其被用于动态存储计算中使用的数据，但是期望将该数据维持在安全的、受保护的存储器中，其中该数据不能在处理器/IC的外部被发现(或者甚至是那些驻留在处理器/IC内部的存储器，该处理器/IC由多个同时运行的虚拟机或过程共享，这些虚拟机或过程可能潜在地从彼此之间窃取数据);如前所述的，针对较小尺寸的工艺技术，这样做是非常困难，尤其是对于NVM，因为可以被用于提供片上受保护的存储器的技术有时与特定处理器或其他IC的工艺技术不兼容。与第三组实施例相关联的原理可以通过提供对受保护数据的片上加密和解密来用于解决该难题，然后该受保护数据可以被安全地片外存储在不受保护的存储器中(但以加密的形式)。阐述利用本文呈现的新颖技术的较新的工艺技术，而不是将数据存储在受保护的、安全的、片上存储器中，可以可选地使用安全的、唯一的密钥(例如，基于本文所描述的一个或多个所选检查点硬件签名测量)来加密这种片上数据(例如，在处理器IC中)，并且然后将数据片外存储(通过非限制性示例的方式，常规地，65nm工艺NVM);该过程可以被可选地应用于允许对编程比特流(例如，其中固件或其他指令存储在外部)或动态存储器内容(例如，创建内部存储器的“影子”，然后其被存储在片外，根据需要对其进行取回和解密)的外部加密存储。

[0036] 其他组实施例将在下文中进一步介绍，或者从以下描述显而易见。这些各种实施例及其各种特征可以可选地一起地、个体地或以任何期望的排列或组合被采用。例如，在一个可选的实施方式中，本文描述的各种实施例可以被组合以提供安全的标识处理器芯片，其中该芯片包括用于恢复安全的、易失性或非易失性密钥并且提供安全处理功能的宽阵列的电路装置。

[0037] 特定设想的实施方式可以包括“硬件逻辑”、“电路”或“电路装置”(各自表示一个或多个电子电路)。一般来说，这些术语可以包括模拟和/或数字电路装置，并且本质上可以是专用目的或通用目的。例如，如本文所使用的，用于执行特定功能的术语“电路装置”可以包括一个或多个电子电路，这些电子电路被“硬连线”的(或“专用”的)以执行所述功能(即，在一些情况下没有指令逻辑的辅助)，并且相反该术语可以包括微控制器、微处理器、FPGA或其他形式的处理器，该其他形式的处理器为通用的设计，但是运行使通用电路装置或将通用电路配置为(例如，配置或引导电路处理器)以执行所述功能的软件或固件(例如，指令逻辑)。注意如该定义所表示的，用于一个目的的“电路”或“电路装置”不一定与用于另一目的“电路”或“电路装置”相互排斥，例如，这种术语指示：一个或多个电路被配置为执行功能，并且一个、两个、甚至所有的电路都可以与“电路装置”共享以执行另一功能(实际上，在“电路装置”包括处理器的情况下通常如此)。正如上面所表示的，“逻辑”可以包括硬件逻辑、指令逻辑，或者两者兼有。指令逻辑可以是具有某种结构(架构特征)的方式被编写或设计的代码，使得当代码最终被执行时，该代码使一个或多个通用机器(例如处理器、计算机或其他机器)中的每一个机器表现为专用机器，该专用机器具有如下结构：该结构必须根据代码对输入操作数执行所描述的任务以采取特定动作或产生特定输出。贯穿本公开，将描述各种过程，其中任何过程通常都可以根据实施例或特定设计而被实现为指令逻辑(例如，作为存储在非暂态机器可读介质或其他软件逻辑上的指令)、作为硬件逻辑、或作为这些事物的组合。本文使用的“非暂态”机器可读或处理器可访问“介质”或“存储装置”意味着任何有形的(即，物理的)存储介质，而与用于在该介质上存储数据的技术或者数据存储的格式无关，例如，包括但不限于，随机存取存储器、硬盘存储器、光学存储器、软盘、CD、固态驱动器(SSD)，服务器存储装置、易失性存储器、非易失性存储器和其他有形机构，其中指令可以通过机器随后被取回。介质或存储装置可以是独立的形式(例如，程序磁盘或固态设备)，或被实施为更大机构的一部分，例如，膝上型计算机、便携式设备、服务器、网络、打印机或其他一组一个或多个设备的一部分的驻存存储器；例如，这种介质可以包括网络可访问的设备、或者选择性地连接到计算设备然后被读取的设备。指令可以以不同的格式被实现，例如，作为元数据(其在被调用时有效调用某个动作)、作为Java代码或web脚本、作为以特定编程语言写入的代码(例如，C++代码)、作为处理器专用指令集、或以一些其他的形式；指令还可以根据实施例由相同的处理器、不同的处理器或处理器核、FPGA或其他可配置电路执行。贯穿本公开，将描述各种过程，其中任何过程通常都可以被实现为在非暂态机器可读介质上存储的指令。指令还可以根据实施方式由单个计算机执行，并且在其他情况下，指令可以在分布式的基础上(例如，使用一个或多个服务器、web客户端或专用设备)被存储和/或被执行。

[0038] 参考本文中的各种附图所提到的每个功能还可以被实现为组合程序的一部分或作为独立模块，被一起存储在单个介质表达(例如单个软盘)上，或存储在多个单独的存储

设备上。本文所使用的“模块”是指专用于特定功能的结构;例如,当执行第一特定功能的“第一模块”和执行第二特定功能的“第二模块”在指令的上下文中被使用时(例如,计算机代码),“第一模块”和“第二模块”指的是相互排斥的代码集。当在机械或机电结构的上下文中使用时(例如,“加密模块”),术语“模块”指的是可以包括硬件和/或软件的专用部件集合。在所有情况下,术语“模块”被用于指代用于执行功能或操作的特定结构,该功能或操作可以被本领域的普通技术人员理解,所述主题对于本领域的普通技术人员属于特定领域的常规结构(例如,软件模块或硬件模块),而不作为用于执行所记载的功能的“任何结构”(例如,“一组oxen”)的通用占位符、“nonce”或“means”。这里使用的“硬连接”是指作为固有硬件设计的一部分的功能的实现,即,其可以“被构建在架构中”;该术语包括这样的情况:其中专用电路必须被设计为以某种方式操作,而不是接受某种类型的可变配置。本文使用的“哈希”是指任何单向函数,即不论这种函数是否符合任何常规密码操作。本文使用的“多稳态”是指具有两个或更多稳定状态的对象(例如,电路);本文使用的“双稳态”是指具有两个稳定状态的对象(例如,电路),即,双稳态电路是多稳态电路的一种。本文使用的“亚稳态”是指电路或条件在一段时间内不稳定,然后解析为数个稳定状态中的一个稳定状态。本文所描述的“多稳态”和“双稳态”电路也是“亚稳态电路”。一般来说,这些电路将具有不稳定的状态或条件,其中该状态或条件在某一点处衰减,并且在不确定性的一段时段之后必须假定其稳定状态中的(至少在理论上)其不可预测的一个稳定状态,其中“稳定”衰减或阻尼振荡发生;一般来说,在一些(但不一定是所有的)情况下,这些电路涉及某种类型的竞争条件,其输出难以基于电路的一般设计的知识来被预测或被复制。例如,本文公开的双稳态(亚稳态)电路可以具有输出(“Q”),在理论上,当电路被激励时,该输出是不可预测的,但它将假定“0”或“1”的逻辑状态。理论上,在激励之后,这种电路应该有时假定逻辑“1”输出,并且在其他时候假定逻辑“0”输出,但是在实践中,由于制造工艺角,这种双稳态电路的特定实例可能会倾向于比不产生逻辑“1”输出更多地产生逻辑“1”输出,或者相反,比不产生逻辑“0”输出更多地产生逻辑“0”输出。注意,虽然在以下各种实施例中双稳态电路被讨论为多稳态电路的特殊情况,但是设想本文中的技术通常适用于具有多于两个的稳定输出状态(例如,三、四、五、或者实际上是任意数目的稳定输出状态,在给定总体的电路单元设计的情况下,只要激励条件产生理论上不可预测的输出)的多稳态电路。关于本文中的各种实施例,术语“设备”是指具有电路装置和可能驻留的软件或固件的电子产品(例如,基于芯片、系统或板);术语集成电路(“IC”)通常是指封装或以其他方式的裸片;IC还可以是设备。本文中使用的“硬件签名”是指所测量的或导出的值(即,可表示为比特数目),该值表面上表示一块硬件;通常(根据本文中讨论的可选技术)硬件签名从设备被读取,或者以其他方式产生作为对设备的测量结果,并且被渐进地修改以恢复“根密钥”。“根密钥”指的是密钥,通常是秘密密钥,它是从硬件中获得或导出的,并且旨在作为锚点(即,其旨在在设备的使用寿命期间随时间被固定)。在一些但不是所有实施例中,硬件签名表示根密钥的测量,即,其中测量可以具有或不具有相对于根密钥的差错。“设备ID”或如本文使用的安全数字身份是指将直接用于标识或用作特定设备的代理的数据;通常(但不总是),设备ID是秘密密钥或不对称的私密密钥/公共密钥对的组件。在一些实施例中,设备ID可以从硬件根密钥中被生成(或依赖于硬件根密钥被得出)(即,设备ID基于根密钥被生成,但是根密钥本身并不直接用作设备ID或由设备ID可导出)。在其他实施例中,硬件根密钥被用于加密秘密密钥或私密

密钥(或设备ID的其他部分),使得它能够被安全地存储,即设备ID的部分可以被加密并被片外存储(换言之,以断电后仍能持续的方式);当设备被重新供电时,设备恢复其硬件根密钥而无需暴露片外的该根密钥,并且然后使用根密钥来取回和解密私密密钥,从而增强唯一的设备ID在IC(例如,使用新工艺技术制造的IC)上的使用。

[0039] 将本公开粗略地如下组织:图1A至图1D将用于讨论与上面介绍的第一组实施例相关的一些可选的一般原理(即,涉及根密钥的生成和点检查)。图2A至图2C将用于讨论与上面介绍的第二组实施例相关的一些可选的一般原理(即,涉及亚稳态电路和物理不克隆函数的测量)。图3A至图3E、图4A至图4C和图5至图6将用于讨论更详细的实施例,这些实施例例示本文所述的特定证明曲线过程,并且混合和匹配来自前两组实施例的可选特征。最后,图7、图8A至图9B和图9A至图9B将用于讨论一些更具体的应用。

[0040] 图1A是产生可恢复根密钥的设备101(例如,包括硬件或具有固件的硬件)的一个实施例的说明性图示。设备101基于一组亚稳态结构的使用。这些结构可选地可以单独地(逐个电路)或在子集中被重置为一组(由附图标记104表示),并且可选地可以基于交叉耦合的NAND门设计105(如图2A-2C中所示),或以其他方式被实现为亚稳态电路。

[0041] 在一个实施例中,所描绘的亚稳态结构103促进测量PUF,即,测量不可克隆的而只能从电路的内部被测量的操作或结果。与许多常规结构不同,所描绘的阵列是其中PUF可以以期望的间隔(无论是在系统、芯片或阵列上电或其他)被测量的阵列。例如,每个亚稳态结构是具有初始不稳定条件的电路,其中该不稳定条件得到解析,并且电路以基于其一般设计不可预测的或固有的方式表现。例如,对于图2A中所看到的交叉耦合NAND门,在初始上电时或在施加所描绘的“激励”信号时,“Q”值应该是不稳定或不确定的;当“激励”信号被移除(或在竞争条件被解析后),“Q”值假定为两个稳定输出状态中的特定一个。,虽然该值一般来说在理论上是不可预测的,但是对于IC上的这种多稳态电路的大型阵列,由于IC-IC工艺的变化,这些多稳态电路的一些子集将倾向于在重复的基础上可预测地产生相同的行为,尽管有该假定的不可预测性(例如,对于一组多稳态结构,一些亚稳态电路“单元”一旦被激励将倾向于在一致的基础上产生逻辑“1”,一些亚稳态电路“单元”当被激励时将倾向于在一致的基础上产生逻辑“0”,并且一些亚稳态电路“单元”将是真的不可预测的,有时假定一个状态或另一状态具有混合概率,即,尽管一般的单元设计使得生成的结果状态应该总是不可预测的);在每个生产的IC独立的基础上,对于每个多稳态电路“单元”以唯一的且不可预测的方式,这些倾向或概率可以由细微的电路变化、设计或制造所引起,这会引起一个路径比另一路径大,或者影响电路操作的某种类型的施加偏置(诸如噪声、电容或其他寄生或固有参数因素)。注意的是,IC制造商通常试图设计IC,使得每个亚稳态单元完全相同或不可预测,并且这是非预期的(例如,基于制造的)工艺角,其导致变化和唯一性,该唯一性是随机的并且在理论上提供唯一的签名(如果电路单元的数目足够大)。虽然这种倾向的倾向性不能基于将电路观察为实际问题而被物理地检测(即,在外部测量路径差异或这些偏置是极其困难的),但是该电路操作当退出时的值和/或统计倾向(即,物理不可克隆函数或PUF)可以将结构组作为整体进行测量;在一些实施例中,这被称为“性能特性”,例如,个体电路相对于工艺角的性能是通常被测量的。为此,在适当的测量时间,在合适的硬件或指令逻辑(例如,固件108)的控制下,测量电路107读取多稳态结构组(例如亚稳态电路的整个阵列)的输出,以导出测量的硬件签名(参考附图标记109)。对于其中亚稳态结构动态可重置

的实施例(换言之,它们可以在任何时间被动态地测量),该硬件/指令逻辑可以可选地进行多个测量,并对这些测量进行平均,或以其他方式对它们进行统计处理,以导出这种测量的硬件签名的稳定版本(参考附图标记110)。下面将呈现出示例性的可选实施方式。

[0042] 如前所述,电路可能会随时间老化,导致特性改变以及设备硬件签名的潜在的漂移和不可标识性。为了减轻这种情况,本文公开的一些实施例提供了在这种漂移的情况下恢复并继续使用原始标记或测量的原始硬件签名(即,根密钥)的容错机制。注意,在通常的实施例中如潜在的其他因素,时间推移和电路老化可以是漂移的原因,但还要注意,操作温度和频率通常不是亚稳态结构和产生的PUF的电路操作变化的显著来源;对于许多硬件实施方式而言,这通常是有益且期望的品质,即,这表示尽管操作条件发生短期变化,基于PUF测量的硬件签名对于IC将是稳定的,并且IC操作将在宽的温度范围内(例如0℃至70℃,-40℃至85℃或其他某个范围)、以及宽的操作频率范围(例如几千赫兹至几千兆赫)是稳定的。如在图1A中参考附图标记111,硬件和/或指令逻辑将测量的硬件签名与检查点数据"CP"113的集合的数据(在一些实施例中被配置为证明曲线或"AC"数据)进行比较,以确定测量的硬件签名是否对应于先前的经点检查的硬件签名;如果是,则测量的硬件签名被视为与较早的检查点签名(或原始根密钥)相同。如果比较结果不匹配,则推断出测量的硬件签名中存在某种类型的差错(例如,漂移或瞬时随机噪声),该差错应该被校正/去除。在一些实施方式中,检查点数据(例如,证明曲线数据)可以被片上存储在同一芯片或IC上(例如,在与处理器并置的存储器中),尽管在其他实施方式中可以将该数据存储在片外117。无论是片上还是片外,这些代码通常被存储在某种类型的NVM115中。在一个实施例中,检查点数据在后进先出("LIFO")的基础上被取回和被用于比较,其目的将在下面被进一步说明,但是同样,这不必是针对所有实施例的情况。注意,如参考附图标记121,新测量的硬件签名(即,无论是否存在任何的表示的漂移)可以被用于生成新的检查点数据,然后将该数据添加到针对先前检查点的已经存储的数据(即,新的检查点数据可以用于未来的签名测量和检查点回滚过程)。如前所述,在一个实施例中,检查点数据和存储过程可以表示如下架构:其中(a)任何当前或先前测量的硬件签名或(b)原始根密钥都不可以单独(无论是单一地或共同地)从检查点数据中导出。换言之,在一个实施例中,检查点数据可选地是(或可以认为是)单向哈希函数。在另一实施例中,检查点数据可以被认为对应于与在许多现代密码过程中使用的椭圆曲线的概念类似的"证明曲线",并且其可选地呈现可逆过程,其中不同的曲线可以被定义,每个曲线都表示不相关联的(即,看起来不相关的)检查点数据。下面将进一步讨论这些选项和能力。在发生差错的情况下,电路装置119"回滚"或修改/校正当前测量的硬件签名,直到比较电路装置111以确认当前硬件签名已经被回滚(即,修改)的方式检测与先前存储的检查点的匹配,以便于匹配先前的经点检查的状态。如所指示,该过程提供了总差错校正的形式,其中漂移被移除并且原始的(经恢复的)根密钥125作为部分的迭代过程最终被获得。其结果是可恢复的、可校正的根密钥/设备ID,该根密钥/设备ID在数字设备的寿命期间始终可以被恢复(参考附图标记数字127)。通过2^{-1280}(大约等于10^{-350})的统计导出的差错率目标,这种密钥恢复过程在其无差错行为中表现出明显的高度"置信度"。

[0043] 注意利用所描述的点检查过程123,在理论上可以简单地将最近测量的硬件签名和与原始根密钥相对应的检查点进行比较,并且经由修改最近测量的硬件签名的单个过程,简单地直接恢复原始根密钥,直到过程验证当前(例如,如经修改的)签名匹配原始根密

钥。在下面讨论的其他实施例中，虽然这针对一些实施方式被设想出，但是呈现了迭代回滚过程，该迭代回滚过程以潜在地需要多次迭代的方式，将每个测量的硬件签名迁移回紧接在前面的检查点状态，然后迁移至其紧接在前面的检查点状态，等等，直到遇到最早的检查点状态(即，表示原始根密钥的检查点状态)。在一些实施例中使用这种迭代方法的原因是基于这样的假定：点检查被执行得足够频繁，使得在不同检查点之间的漂移最轻微，例如根本没有漂移，或者只有0至3比特的偏移等。在典型实施方式中，由迭代修改过程所消耗的时间对于轻微的偏移(例如微秒)非常短，并且在遇到更多的迭代偏移并且必须同时被解析或被猜测的情况下，所消耗的时间可能会大得多。更简单地说，对于这些实施例，与通过与仅单个检查点的比较来试图同时恢复20比特的漂移相比，迭代地校正20比特的漂移(一次一个比特作为相应检查点的一部分)使得20个检查点被用来去除例如每次1比特的漂移通常是更快的。在一个实施方式中，对于给定的芯片设计(例如，特定的FPGA设计或其他处理器设计)，制造商在产品发布之前反复测试设计，以标识最坏情况下的漂移，并且理想地实现点检查过程以使点检查被执行地足够频繁，使得永远不会遇到最差情况下的边缘漂移场景。例如，作为与本文中特定实施例相关的被执行的仿真或表征的结果，相信对于大多数硬件设计，将不断地遇到硬件签名的漂移每年不超过2至3比特；注意，这些参数可以根据产品设计、制造工艺可靠性和质量等其他因素变化。特定的硬件制造商可以测试其预期最坏情况漂移的特定设计，并且可以实施点检查，使得对于任何点检查迭代(即，相对于先前的检查点)不再遇到超过1至5比特(或更少)的漂移。例如，如果每年会有2至3比特的漂移被预期作为给定设计的最坏，那么制造商可能会例如每3个月或更频繁地调度点检查(以及新的证明代码的存储)，结果逐个检查点的漂移将不超过2至3比特(并且通常是0至1比特)。如从该讨论中可以推断出的，点检查和特定点检查算法的频率在很大程度上将是实施方式的选择。例如，在一些实施例中，检查点数据管理可以采用滚动窗口方法，由此达预定数目的检查点被保留，诸如最近的30个检查点(以及对应于原始根密钥的“epoch”检查点)，而较旧的CP被清除；这种实施例可以有助于包含检查点管理所需的存储空间。当然，其他的修改也是可能的；例如，另一个方案可以保留最近的30个检查点，同时保留每年的一个检查点作为清除较旧检查点的结果。

[0044] 图1B提供了描绘尽管在漂移(或其他类型的差错)的情况下用于恢复根密钥的技术的另一实施例的流程图。通常使用附图标记131来指定图1B表示的技术。如附图标记133所示，片上硬件阵列使得能够测量PUF以恢复易失性或非易失性的密钥；如通过可选的特征块134所表示的，这种硬件阵列可选地可以由个体亚稳态或双稳态单元组成，例如，其可以包括诸如图2A中所示出的交叉耦合的NAND，并且可以被动态地测量(例如，被激励以触发PUF)，而无需对整个芯片或阵列进行循环通电和关断(或者以其他方式转储芯片或设备内部所需的存储器内容)。参考附图标记136，一旦PUF被触发，则测量电路装置随后测量多稳态阵列中的各种单元的生成的逻辑状态，并且基于这些状态输出测量的硬件签名。该测量的硬件签名可以是亚稳态阵列的单个测量的结果、或多个测量的结果(例如，平均测量、或来自或使用统计过程的测量-作为非限制性的示例，并且如下面进一步描述的，例如可以多次测量每个单元，并“丢弃”被认为表示统计偏离的多个测量，并且然后对剩余测量进行平均)。在一些实施例中，许多测量在由正在输出的许多测量所表示的主导状态下进行。这些过程和算法的许多变化将被那些本领域普通技术人员所想到。

[0045] 如附图标记138所示，给定时间的推移和硬件设备的老化，通常假定一个或多个差错源将损坏测量的硬件签名(相对于原始根密钥)。在一些实施例中，该差错被假定为在随机瞬时误差之上的随机的、渐进的漂移，其显示为亚稳态电路在被激励时采用一个状态对另一状态的可测量行为(倾向)的随时间的变化。以简化的示例为例，如果测量的签名的特定比特(“比特4”)(对应于特定的亚稳态单元，例如，阵列的“第四”)倾向于在PUF被测量时产生逻辑“1”，则随着时间的推移，漂移可能会慢慢将该倾向改变为逻辑“0”；预期这种倾向将会持续某段时间(即，进一步的测量会倾向于产生针对该比特的逻辑“0”)。如更进一步地随着时间的推移，漂移可能会损坏测量的签名的另一随机比特(例如，比特“7”对应于阵列的第7个单元)和/或它可能会再次进一步地损坏“比特4”，使得针对该比特的测量的签名再次趋向于逻辑“0”。因此，测量的硬件签名可以表示相对于前面测量的硬件签名的一些未知漂移量；该漂移可以随着时间改变，通常是几个月或几年，来影响签名的不同比特。

[0046] 因此，所描绘的技术131的集合试图确定测量的硬件签名是否与先前的“检查点”硬件签名和所讨论的硬件设备的原始根密钥相同或不同。参考附图标记139，针对新测量的硬件签名计算检查点数据。例如，可选地该数据可以是单向或双向加密函数，该函数产生数据集合，该数据集合将为刚刚测量的硬件签名提供未来的检查点。参考附图标记140，在可选的实施方式中，该检查点数据是证明曲线(“AC”)数据(下面将进一步讨论)。在一个实施例中，产生多个数据集合，每个数据集合表示不同的加密，其中将至少一个数据集合拆分为小数据集合，以促进对(大的)测量的签名是否对应于先前检查点值进行碎片化分析。例如，在一些实施例中(下面讨论的)签名可以包括许多比特(如256、512、768、798、1024、2048或其他的比特数)，并且对于检查点数据的一个集合，验证/比较过程可以被拆分，使得相对大的签名长度(例如，768比特)被拆分成较小的分区或“块”(如24、32、42、64或不同的比特数，作为非限制性的示例)，并且只在逐块的基础上执行漂移减缓过程。例如，在使用768比特签名的情况和使用42比特块以定位漂移的情况下，针对每个检查点，18个代码或加密产品可以被产生并且被存储(即，18*42＝756～768)。当新测量的签名(例如，当前的硬件签名)与检查点进行比较时，新的签名被分割为类似的块(例如，42比特子集)，并且检查点数据被逐块地使用，以隔离可能表示漂移的块以及针对正确的、先前的、经点检查的签名可能存在的候选解。注意，此处给出的特定值是说明性的，例如，可选地不同的密钥长度/签名长度/块大小和/或检查点代码的数目可以被用于其他实施例中；例如，可以使用24个块，每个块表示32比特签名子集(例如，32*24＝768)。该变型是实施方式的决策，即，所使用的比特(块大小)越少，假阳性(即，候选)的数目就越大，但是每个块的运行处理时间就越快。许多假阳性可能需要较长的下游运行时间处理，以清除假阳性，并且合适的块大小的选择通常需要在个体块的运行时间处理和与修剪、假阳性相关的运行时间处理之间进行平衡。如前所述，检查点数据的集合可以包括提供提供多个检查的数据，其中这些检查中的至少一个检查针对块特定比较而被优化，并且这些检查中的至少第二个检查表示未分段的硬件签名。换言之，基于块的过程表面上可以增加混叠的可能性(例如，多个不同的签名值匹配基于块的检查点数据)；提供附加的全长度检查有助于避免该问题，并且提供区分(即，修剪)从基于分区的过程产生的假阳性的方法。此外，全长度检查的渐进增加可以在数学上将混叠概率调整为任何期望目标，例如10^{-300}，这是模拟-6000dB“单点通过”数字滤波器的极小数字(概念上类似于“带通”数字滤波器，由此，只有窄带的数据点可以保持存在，而其他所有的数据点都

被修剪)。例如,在一个实施例中,检查点的第一加密数据允许逐块的漂移分析以标识先前检查点签名的一个或多个“候选”(例如,先前检查点签名的可能状态),而该检查点的第二加密数据可以被用于区分备选的候选解,即,以标识真实签名检查点。

[0047] 在一个实施例中,标记检查点的数据以不允许导出产生该数据的测量但有助于确定稍后的、类似处理的签名测量是否相同的方式,被哈希处理或者被压缩或被加密,。更具体地,该检查点数据可以可选地是“证明曲线”数据;如本文使用的,“证明曲线”或“AC”是函数,其中比特集合被划分为第一比特子集和第二比特子集,并且第一子集用于对第二子集进行可逆加密。这种形式的检查点与真实哈希略有不同,因为如果某些假设正确(例如,具有对加密函数和第一比特子集的了解,其有效地被用于提供加密密钥),则原始数据可以从加密数据中被恢复。比特的两个子集可以是所期望的任何大小,例如,在一个实施方式中,第一个子集比第二个子集大,在不同的实施方式中,正好相反,在第三实施方式中,这些子集的大小相同。在更详细的实施例中,针对每个检查点计算多个AC,其中这些AC中的至少一个AC特定于基于块的比较,并且至少一个AC是全长度AC;针对每个AC,第一比特子集和第二比特子集被不同地选择,因此使得每个AC有效地呈现出完全不同的非多项式曲线,该曲线通过与检查点签名对应的点。一个或多个AC的分区促进任何检测到的漂移的定位和隔离,从而允许硬件(或硬件/固件)执行运行时间优化,并且实际上来说,实时校正漂移并且回滚给定的(当前)签名以匹配先前检查点。针对该划分可以产生多个可能的解的实施例,每个全长度(未分区的)AC提供对候选进行区分的方法,并且提供标识针对来自候选的先前检查点签名的真实解的方法。在一个特定实施例中,AC生成过程基于函数c＝H(x,y),其中x和y表示硬件签名的比特的不同子集,并且其中c被视为特定曲线实例的常数。下面将进一步讨论这种实施例的示例。在该实施方式的更具体的版本中,比特的“x”子集使用格式保留加密(“FPE”)过程提供用于加密比特的“y”子集的加密密钥,使得输出(c)也具有与“y”比特相同的比特数目并且被加密,使得在没有更多信息的情况下签名的原始xy比特无法从经加密的输出(c)中被恢复。因为每个AC表示不同的x比特子集,每个曲线表示不同的加密过程,使得这些曲线中只有一个唯一的交点对应于经点检查的签名的校正值。在AC过程的具体示例上进行扩展,如果对签名的逐块分析将呈现出18个块,并且用所存储的检查点AC数据分析那些块,产生逐块解候选集合(1,1,2,1,1,1,3,1,1,1,1,2,1,1,1,1,1,1),这将指示针对签名的块1的该漂移分析已经产生所分析的经点检查的签名的相应比特的一个匹配,然而,块3已经产生针对所分析的经点检查的签名的相应比特的三个候选解。进一步地,考虑所有块,存在针对完整签名的可能被标识的12个解(即,1×1×2×1×1×1×3×1×1×1×1×2×1×1×1×1×1×1×＝12)。如果这些可能的解(候选)的比特以与其他不同的证明曲线(即,AC2至AC 5)对应的方式被加扰,则12个可能的解中只有一个解将与先前检查点的所有四个对应曲线产生精确匹配,并且该特定候选将必须与先前经点检查的签名匹配。当然,该示例性的过程仅表示一个可选的点检查算法,但是这里它被引用为下面将进一步讨论的特定实施例的介绍性示例。如所述,每个AC表示唯一的数学函数,并且多个曲线的使用生成强大的统计可能性,即所有曲线都将相交于针对硬件签名的一个点,该硬件签名精确地匹配于先前检查点使用的签名。

[0048] 因此,在一个实施例中,针对新测量的硬件签名计算新的(对应的)检查点数据,并且该检查点数据与针对先前检查点签名的以前存储的数据143进行比较-该先前检查点的

多个数据集合(诸如针对相同检查点的多个证明曲线的数据144)可以被用于唯一地标记先前检查点签名。如果所分析当前硬件签名的新计算的检查点数据与先前检查点签名的检查点数据匹配(例如,针对该检查点的检查点数据的所有经不同加密的集合),那么这两个签名是相同的。如果不匹配,则假定相对于所分析的先前检查点存在某种类型的附加的、未经补偿的差错(例如某种类型的漂移);对当前硬件签名进行附加的逐比特修改或调整,并重新检查该签名,其中该过程一直持续到标识出正确的先前检查点签名。结果是,根据需要有效地修改所分析的当前硬件签名,以便有效地将其回滚,以匹配作为先前检查点基础的签名。在一个实施例中,按照附图标记141,该回滚过程可以被迭代地、逐检查点地执行,直到原始根密钥被恢复。在备选实施方式中,新测量的硬件签名可以与原始根密钥的证明代码(或其他一些更早的检查点,即,而不是紧接地前面的那个)直接比较,通过进行更少迭代的算法来还原当前签名以匹配原始根密钥(但潜在地增加每次迭代的运行时间,即,增加在任何给定的迭代中所需的处理时间以补偿多于数个比特的漂移)。在另一可选的实施方式中,任何检查点可以被用于加密(并且用于外部地存储经加密的版本的)根密钥,并且因此,在这样的实施例中,它可以足以简单地还原到一个先前检查点,并且然后简单地取回和解密外部存储的(经加密的密钥)。再一次,本领域技术人员将想到许多变型,这些变型由本公开所设想和包含。注意,逐检查点回滚过程提供了一些优势,例如,漂移/差错可以被拆分为多个检查点恢复会话(潜在地最小化运行时间),并且所有经外部加密的"线索"都不提供根密钥或签名的完整版本(例如,经加密或以其他方式)。

[0049] 此时,说明这些特定的可选技术的示例性假设比较过程将是有用的:将7月1日测量的768比特的硬件签名与6月13日经点检查的先前硬件签名进行比较,并被认为是精确匹配的。然后将当前硬件签名与6月2日的先前检查点进行比较,并且检测到不匹配;然后在逐块的基础上进行比较,使得不匹配可以被隔离以存在于小得多的比特组中(例如,在该示例中,在表示特定签名块的42比特的某个子集中)。经由测试这些42比特的不同调整或修改的过程,标识出正确的42比特的候选解,并且针对所存储的先前检查点数据进行测试;最后,针对这些比特推导正确值,从而产生比较匹配,并且从而整个768比特的硬件签名被对应地校正(即,它被"回滚"以匹配先前的签名)。然后将修改后的该当前签名(使用相同的底层点检查/加密技术)与5月16日的更先前的检查点进行比较,并且在该假设中再次检测到不匹配。因此,当前签名将再次被修改,这次将进一步地检测并校正三个另外的不匹配的比特,从而将当前签名"回滚"以匹配5月16日检查点的签名。然后,将当前再次修改的签名随后与表示原始根密钥的第一个检查点进行比较,并且该比较过程指示当前签名与原始(易失性)根密钥相同;从而,原始根密钥已经被恢复,并且当前签名的两次修改版本可以被立即应用于密码应用(即,因为点检查过程已经验证它与原始根密钥匹配)。

[0050] 注意,可以根据检查点数据的形式,以各种方式执行比较过程。例如,如果使用证明曲线,则每个候选解可以被划分为对应的如上所述的x和y的子集——x比特可以被用作反向加密密钥,以从证明曲线的检查点数据("c")还原(解密)到比特y',以确定他们是否匹配于真实解的y比特。对于42比特的签名块,例如,其中比特被划分为14个x比特和28个y比特,28比特的经解密的集合("y'"或"y-素数")可以与候选解的28比特进行比较,以确定它们是否完全一致或紧密对应。注意,这并不是所有的解都需要的,例如,其中点检查仅对候选进行哈希处理的情况下,每个新的候选都可以被进行类似的哈希处理,并且将其与存储

的检查点的对应哈希进行比较，以确定它们是否匹配(即，以标识候选解)；可以通过使用多个不同的加密过程并且验证每个不同加密过程的完整候选签名哈希来类似地解析在候选解中的混叠。

[0051] 如前所述，可选地，点检查被足够频繁地执行使得逐检查点的漂移是小的，从而便于进行快速的片上处理。因此，按照附图标记145，针对新测量的硬件签名(即，标识硬件签名的预修改版本，可选地直接基于PUF测量和/或基于对PUF的处理或修改)，检查点数据被存储以提供该签名测量的检查点，供将来使用；该检查点数据可以与表示先前检查点的数据(按照附图标记146)一起累积地被存储。然后，尽管在硬件签名的测量中存在漂移，该存储为以后发生的PUF(硬件签名)的任何测量提供了始终回滚到该新的检查点并随后回滚到原始根密钥的机制。按照附图标记148，这提供了在硬件设备的整个寿命中可恢复的抗损坏的根密钥；一旦根密钥被恢复，经恢复的根密钥可以被立即应用于加密/解密和其他应用程序中(按照附图标记149)。在一个实施例中，然后该能力允许对外部存储装置的私密密钥(或另一密钥或其他数据)进行进一步的解密150，然后其被用于密码操作(即，同时安全防护根密钥以免受字典攻击)。在其他实施例中，按照附图标记150，这允许对操作数、指令或其他数据进行加密的、安全的外部存储(例如，对编程比特流或其他外部存储器内容进行加密)。下面将提供这些技术的各种示例。

[0052] 图1C是流程图，其示出了用于使用证明曲线以允许漂移(或其他类型的差错)的追踪和允许以逐步移除递增漂移的方式回滚测量的硬件直至恢复原始根密钥的技术的实施例。通常使用附图标记155来指定这些技术。在该实施例的情况下，技术依赖于从物理不可克隆函数(PUF)159获得的片上硬件签名的某些源。通常地，假定给定的硬件设计(例如，给定的集成电路(IC))将包括亚稳态电路阵列(如交叉耦合的反相器、锁存器、NOR门、NAND门和复用器，以及实现竞争条件或不可预测但多稳态输出的任何无数其他结构)。IC制造商通常将阵列设计构建为使得每个亚稳态电路对应于将被复制多次的“单元”；尽管理论上在IC上实现的每个单元旨在在功能上相同地操作，但是阵列中的每个亚稳态电路反映了不同的制造工艺角(例如，包括单元位置、根据晶片和根据工艺批次的变化)，当亚稳态电路被在一起测量时，这将引起唯一的硬件指纹。一般来说，为了提供足够的唯一性，亚稳态单元的数目可以被选择为至少256-2048个，或者更多数目。对于下面参考的一些实施例，如前面介绍的，768个单元的阵列被用于提供768比特的硬件签名。IC设计还有利地支持上面介绍的一些或全部的电路元件(即，无论是专用硬件，还是将由固件配置和/或控制的硬件)以提供未来的片上硬件签名测量和检查点恢复，如前面所参考的。

[0053] 根据实施方式，IC系统制造商或下游系统集成者测量PUF以标识和检查点根密钥160。如图所示，对于所表示的实施例，这是通过标识硬件签名以及计算和存储与该签名对应的证明曲线数据来完成的。在一个实施例中，这可以由测量根密钥并存储用于经由web下载的证明数据的IC制造商来完成(即，使得证明数据提供仅由拥有IC的一方可恢复的已公布基准)；在一个实施例中，固件可以在第一次上电时或在配置步骤期间(例如，假设网络连接)自动地执行该功能。在本实施例中，IC制造商或系统集成者(或者甚至是用户)还可以建立唯一的设备ID，例如，通过生成对设备唯一的私密密钥和公共密钥对并且使用根密钥对私密密钥进行加密(即按照附图标记161)。如附图标记163所示，可选地该经加密的私密密钥可以被片外存储，诸如在与PUF阵列分离的IC中的NVM芯片中。在一个可能的实施方式中，

IC制造商销售第一IC(例如,具有至少一个处理器和嵌入式PUF阵列),并且系统集成者将该IC安装在具有其他电路的板或模块上,如在单独的芯片中实现的NVM,然后对根密钥进行测量和点检查(例如,其中检查点数据被存储在NVM芯片中,或者例如以web可访问的方式再次被远程地存储。但在另一实施例中,IC制造商可以选择通过将初始检查点数据的某种形式的检查和(或哈希签名)记录到IC中的嵌入式的OTP存储器元件中(诸如将128比特或64比特哈希签名存储到IC嵌入式的eFuseOTP中),以将在装运之前测量或得到的根密钥牢固地提交到其第一客户,以便于将它牢牢锁定而不允许在现场或通过OEM(或其他任何人)对根密钥进行任何的进一步修改。在另一变型中,多个不同的根密钥可以由相应的过程被建立,例如,一个特定于IC制造商,另一个特定于系统集成者。许多这种示例将被本领域普通技术人员所想到。为了提供这些选项所提供的能力的一个非限制性示例,IC制造商可以是在IC资格期间建立并检查根密钥的实体,并且使相关联的证明数据通过web下载可获得;在系统组装期间,然后系统集成者提供固件,该固件使用该证明数据恢复根密钥,并且在作为新系统的一部分的局部NVM中的生成和存储新检查点。这种系统集成者还可以生成私密密钥,并且使用经恢复的根密钥加密私密密钥,然后该私密密钥也被存储在NVM中。然后,当系统被售出给消费者(或其他用户)并且被首次运行时,根密钥和私密密钥则准备好立即使用和/或应用。这些技术的许多进一步的示例和变型将被本领域普通技术人员所想到。无论使用该过程的哪个版本,在一些实施例,当设备ID被建立并且与芯片和/或产品分布相关联时,对应的公共ID可以被发布(165)以便于提供所讨论的芯片和/或产品的随后的认证和/或标识(例如,使用PKI处理和相关联的证书)。

[0054] 线157标明在首次配置和使用之前例如由制造商或芯片供应商执行的离线过程(即,线157以上);该线以下的步骤表示通过设备后分布执行的根密钥取回和抗损坏过程。线157以下的功能可以在IC寿命期间的任何点执行,通常是在需要标识根密钥时(例如,在设备断电之后,以及需要将其重新打开时——注意,这可以根据实施例而变化,例如,在其他实施例中,即使产品没有被关闭,过程也可以被动态地重新执行)。如前所述,每个分布式IC(及其亚稳态电路和PUF的相关联阵列)在设计上看起来是相同的,并且甚至对这些电路内部工作的检查需要对所讨论的特定电路进行解封和破坏。在防篡改或显窃启领域中的技术可以被选择性地应用,以保护原始的仅内部的硬件签名测量和信号不被有效地探测或发现。针对任何给定的IC,硬件签名不能被读取为实际问题,直到IC阵列内部的PUF阵列被“激励”,随后“竞争条件”结算接着被测量,以标识阵列中亚稳态单元的倾向性。如前所述,针对本文中的某些实施例,可以进行阵列设计使得PUF是动态可读取的,即它可以在任何时间被测量,并且不需要清空有效使用的存储器阵列或对存储器阵列或整个IC进行断电和通电。结合图1C的讨论的其余部分,假定期望测量特定集成电路的硬件签名;根据设计,由附图标记167表示的该测量可以由硬件电路装置或指令逻辑控制硬件电路装置触发。使用任何期望的算法对测量进行处理,该算法生成签名值“h-sign”。数据块168将该签名标识为“根密钥”(根密钥“素数”),以指示它与根密钥相对应,但可以与根密钥相同也可以不相同,即,可以存在一些漂移。

[0055] 如前所述,处理逻辑(硬件和/或指令逻辑)试图将测量的硬件签名与标记先前检查点的数据相匹配(直到,并且包括,在工厂标识的原始根密钥,或当IC被首次配置、安装或集成时)。为此,电路装置从片上或片外存储器访问(取回)一个或多个检查点数据集合171;

在图中，先前检查点数据的特定集合用附图标记k表示，即，假设存在n个先前检查点，其中n可以是从“0”(即，先前检查点标记根密钥)至较大数字中的任何数。如前所述，在一个实施例中，检查点数据本身不允许任何硬件签名或根密钥的导出，而是提供有关稍后新测量的硬件签名(或其修改版本)是否正确的线索。请注意，该根密钥取回操作只能由IC来执行，因为IC是唯一对测量的硬件签名有访问权的一方，而IC外部的任何攻击者由于不具备这种知识而非常不利，并且只能对存储的检查点数据进行蛮力的全枚举攻击。同样如附图标记169所示，新测量的硬件签名可以被自身进行点检查，其中经加密的数据(例如，证明曲线数据)以所描述的方式被生成和存储——这种数据将提供可以在未来被使用的新检查点(“k+1”)。

[0056] 然后，密钥匹配逻辑执行具有一对多迭代(即n…0)的循环过程，如块171所示，针对每个先前检查点的一次迭代被用于恢复根密钥。针对每次迭代，逻辑采用所分析的硬件签名(“当前”硬件签名)，并且将比特分类到x和y子集中，就像创建较早检查点(AC集0，1…n)所完成的那样；该过程可以使用用于创建先前检查点数据的共同选择标准来执行。例如，如果所分析的检查点针对证明曲线“AC1”将比特号为8、344、67、……等的比特分类到x子集，该x子集用作用于加密y比特的密钥(从而形成证明曲线“AC1”)，然后相同的分类被用于当前硬件签名。注意，分类标准与检查点数据一起被存储为明文的形式(在一些实施例中，该标准被表达为转置向量“T”，如下面将讨论的，一个转置向量特定于每个所使用的证明曲线，例如，T1针对AC1，T2针对AC2，等)。在第一迭代中，用于匹配/比较过程的检查点数据(和分类标准)可以可选地是最近的检查点(AC_n)，并且因为附加的回滚迭代被处理以恢复先前检查点，最终将到达与原始根密钥对应的最后检查点(AC_0)。如决策块173所示，然后密钥匹配逻辑尝试确定与先前检查点签名的对应比特的加密相对应的比特的适当的x和y子集；如前所述，在一些实施例中，这通过生成反向加密信息并且重新创建生成证明曲线数据的y比特来执行，即，通过使用当前硬件签名的x比特来计算函数$y'=H^{-1}(x,c)$(即，通过使用这些x比特和先前经加密的输出c进行解密以取回y')。回想每个证明曲线的x比特被用于形成对应曲线的加密密钥；针对所分析的曲线，如果所选择的x比特与先前签名的对应x比特不匹配，则由于比特扩散/雪崩效应被用作加密过程的一部分，结果通常会有很大的不同(即，将会存在许多不匹配的比特)。相反，如果结果精确匹配，则所分析的当前硬件签名的比特将被视为先前检查点的对应比特的匹配候选(对于x和y比特都映射回到当前硬件签名中它们的位置。最后，如果只有微小的变化(例如，在解密之后，不到三分之一的比特是不同的)，则可以假定漂移存在于操作数中(即，在y比特子集中)。当然，同时出现两种类型的差错(x比特差错和y比特差错)是可能的，并且所描述的方法论首先解析任何的x比特不匹配，然后标识y比特中剩余的任何差错。附图标记174表示虚线框，详细说明了在这种详细描述的实施例中执行的一些过程，即，与AC1对应的证明数据的第一(逐块)集合可以首先被逐个块地检查，如附图标记175；应该回想起，每个块表示比特的分区，诸如24、32、42、64，或者原始硬件签名的比特的一些其他子集。块最初简单地通过以下来被选择：对硬件签名的比特进行分类(例如，使用转置向量)，并且然后采取经分类的比特的分段(例如，针对每个相应块，比特1至42、比特43-84、比特85至126等)。通过执行逐块分析，逻辑能够隔离对特定(多个)块的任何漂移，以有助于最小化运行时间处理。例如，如果假设在当前硬件签名中恰好存在1比特漂移，相对于所分析的检查点，要标识(例如，签名的768比特中的)哪个比特对应于漂移，

这在计算上可能是令人生畏的。通过将签名划分为分区(例如,每个分区32个比特的24个分区,或每个分区42个比特的18个分区,等等),这允许密钥匹配逻辑聚焦在用于分析的小比特集合,并且从根本上最小化运行时间处理要求。在预期的变型中,这些比特集合的分区可以在比特上重叠,尽管这不是所有实施例都需要的(例如,它们可以被相互排斥地分区,如这里例示的)。正如过程块176所指示的,密钥匹配逻辑适当地调整当前硬件签名,直到匹配与比特相对应的加密数据,以标识一个或多个候选解。按照块177,它然后通过查找相同检查点的其他数据(例如,其他的全长度证明曲线)来标识正确的候选,因为从统计上讲,只有一个候选解应该满足所有的这些功能。允许混叠点通过这些全长度证明曲线提供的检查的可能性是低的,并且通过划分器=MIN($2^{|x|}$,$2^{|y|}$)由每个递增AC测试来降低该可能性,其中|x|是x的比特大小,并且|y|是y的比特大小。例如,如果|x|为256,并且|y|为512,然后划分器是2^{256},这是约等于10^{70};每个附加的证明曲线有助于混叠可能性的进一步累积降低;针对768比特大小的硬件签名,在|x|=256和|y|=512的情况下,由5个附加的全尺寸AC的累积检查总计达10^{-350}的净混叠概率,这是极小的数。然后正确的解将被采取为当前的硬件签名,并且被视为与考虑下的先前检查点(k)相等;按照附图标记179,确定它是否对应于根密钥(即,是否k=0)。按照存在于决策块179的标记为"是"的路径,如果k=0,则根密钥确实被视为已恢复(即按照数据块180),并且该方法进行基于根密钥的应用(按照附图标记181);如前面所介绍的,可选地,经解密的根密钥可以用于取回和解密(182)外部存储的经加密的私密密钥(或其他秘密密钥)。如果当前检查点不是根密钥(按照存在决策块179中标记为"否"的路径),则逻辑递减k,并且循环回下一检查点迭代,如附图标记183和185所表示的。该进程的结果是,逻辑将在一个或多个迭代中"回滚"测量的硬件签名,以获得原始根密钥。注意,在一个实施例中,所描述的方法对每个检查点按顺序逐检查点地进行操作;但是,这是可选的,而且方法可以跳过检查点、或者首先检查根密钥,或者使用其他步骤来到达原始根密钥也是可能的。例如,如附图标记184所示,并且如前所述,在一个实施例中,经点检查的硬件签名可以被自身用作加密密钥,并且用于外部存储根密钥的加密版本;因此,在这种实施例中,可以潜在地简单地恢复先前检查点,并且然后使用对应的签名来取回和解密外部存储的根密钥(或向量值,该向量值在与现有签名结合或用于修改现有签名时,以允许对根密钥进行立即恢复)。

[0057] 回顾刚刚讨论的一些原理,点检查过程可以是证明曲线过程。检查点计算过程和匹配过程可以以允许对漂移的智能的、确定性的导出及其缓解的方式被构建;例如,一些实施例采用雪崩效应来生成证明曲线数据,使得一个比特的差(例如,在加密过程的输入中)级联到输出中的多个比特的差。所描述的证明曲线生成过程可以被构建为,使得每个分段的高阶比特中的单个比特或少量比特差(例如,用作加密密钥的"x"比特)影响在对低阶比特的加密和解密中的大量比特差(例如,在作为对加密操作数"y"进行加密的结果而被输出的"c"比特中,或在作为对解密操作数"c"进行解密的结果而被输出的"y"比特中);这种结构允许密钥匹配逻辑进一步隔离和确定地分析当前硬件签名和检查点之间的不匹配。这种雪崩效应有利地引起块分析(例如在AC1中)以产生非常少的假阳性,从而减轻了随后运行全尺寸AC的工作负载和运行时间。它还有助于增强全尺寸AC的混叠拒绝能力,以接近"混叠可能性划分器"的理想统计模型(即,如以上讨论的,MIN($2^{|x|}$,$2^{|y|}$))。这种结构允许对漂移的快速导出(例如,微秒级)和对当前硬件签名的快速修改,直到它与检查点数据匹配(即,

证明曲线数据)。

[0058] 注意,本领域普通技术人员将想到对以上描述的可选的回滚和点检查过程的许多变型。尽管在本公开中没有被明确地阐述,但是本公开设想了这种变型。作为一个非限制性示例,预示下面将进一步讨论的实施例,在一个实施方式中,证明曲线数据的“精确匹配”不是必需的,并且所描述的技术对于标识产生证明曲线数据的候选是有效的,该证明曲线数据在距先前检查点(k)的证明曲线数据的预定汉明(Hamming)距离之内。这种解可以被容忍,因为最终,给定先前检查点的证明曲线的数目(例如,5至6个或更多),以及硬件签名的复杂性(例如,至少512比特,优选地更多),比较过程实际上只能产生与所有数据集合匹配的一个解。事实上,下面进一步讨论一个可选的实施例,可以利用该能力进一步提高所存储的检查点数据的安全性和对攻击的抵抗力;新的检查点数据可以有意增加随机噪声(例如,在值中被屏蔽和/或翻转的特定比特)达点检查算法的汉明距离公差。例如,在下面讨论的一个特定实施例中,针对正被存储的检查点的每个备份的未分区的证明曲线(即,AC2至AC5),高达36比特的差错可以被有意地注入到证明曲线数据中,因为所描述的回滚过程将仍然恢复正确的先前检查点(即,将仍然仅存在一个解落入在先前检查点的所有五个证明曲线的汉明距离内);如前所述,差错的可能性仍然很低,以致于实际上它不会在实践中发生(例如,即使通过这种故意的差错,多于一个解的可能性仍然非常低)。例如,上述的对混叠概率的递增划分器可以适度地稍稍从2^{256}减少到2^{220}。注意,注入的差错并不被存储在任何地方,即,所存储的检查点数据“c”(即“c-素数”,其中c’≈c)封装一些差错,但是通过密钥x被不可标识地加密。以这种方式注入差错将使攻击者(例如,拥有运作的量子计算机的攻击者,其承载可以解决为唯一解的q比特阵列)更难以标识正确的先前检查点签名,甚至即使完全了解表示先前检查点的所有外部存储证明曲线数据,也是如此。在可选实施方式中,这种有意注入的差错可以被随机改变,例如,经由将0至36比特的差错随机添加到加密操作数或加密输出中的随机比特位置的函数(例如,fn{y}或fn{c}→c’)。对于其中将这种差错注入经扰乱的“y”操作数(c)中的实施例,如果解密过程仅依赖于转置和选择性的比特翻转(例如,在进行解密时,差错比特的数目仍将在对应的全签名长度证明曲线的预定义汉明距离内),则比特的数目将承载到c’(y”)的解密中。

[0059] 图1D是集成电路(“IC”)187的框图,它提供了易失性根密钥或硬件导出密钥。IC包括板载阵列或其他电路188,其支持物理不可克隆函数(PUF)。当需要测量PUF时,板载电路装置189执行该测量以获得硬件签名。如附图标记191所指示的,IC还具有提供板载点检查功能的电路装置,该功能提供原始根密钥的恢复,尽管在读取硬件密钥时假定存在差错。该点检查功能可以可选地依赖片外存储器192来存储差错恢复信息,诸如通过非限制性的示例,如前所例示的差错代码、哈希和/或证明曲线数据。测量/校正过程的结果是经恢复的根密钥和/或唯一的设备ID 193,然后其可以被应用于各种应用195。在一个实施例中,这些应用可以可选地包括密码学(例如,入站(inbound)或输出信息199的加密和/或解密)。在另一实施例中,恢复的根密钥可以用于解密外部存储的经加密的秘密或私密密钥197,如前所述。可选地,这些应用可以包括加密将存储在外部存储器198中的数据或解密来自外部存储器198的数据。在另一非限制性的示例中,应用195可以包括硬件随机数生成,例如,PUF还可以被用来提供随机数生成种子,如下面将进一步描述的。注意,下面将描述每个IC、处理器或核使用两个PUF阵列的实施例;例如,一个PUF可以被用于根密钥导出,而另一PUF可以专

用于随机数生成。该表述表示,IC 187可以包括单个处理器或多核处理器或另一形式的处理器,诸如(但不限于)FPGA或GPU(图形处理单元),并且IC作为整体或每个这种处理器或核可以具有相关联的根密钥和检查点生成和恢复逻辑(例如,以专用于每个核的方式)。IC可以是封装的或未封装的独立的裸片、安装在印刷电路板上的裸片,或是与其他裸片组合(堆叠)的裸片的形式,并且可选地它可以包括本文中描述的任何其他技术(例如,特定的点检查过程和/或基于NAND的双稳态单元)的任意排列或组合。

[0060] 图2A-2C被用于描述示例性的实施例,该实施例允许动态测量例如植于处理器中或集成电路(IC)上的电路元件阵列中的PUF。注意,在这些实施例中,上述较早提及的硬件签名、根密钥和其他过程的使用是可选的,即,参照图2A至图2C描述的技术中的电路装置和相关结构提供了电路和相关技术的唯一集合,该唯一集合可以由其自身或者可选地与本文所描述的一个或多个其他特征一起实践。

[0061] 图2A是示出了可以被用作动态PUF测量的基础的双稳态电路设计201的图示。该设计指的是可以被复制多次的单个单元(以虚线框来指示),其中每个单元提供一个比特的信息。所描绘的设计依赖于两个NAND门203和205,它们相应的输出207和209交叉耦合。换言之,第一NAND门203的输出207形成第二NAND门205的输入207',并且第二NAND门205的输出209形成第一NAND门203的输入209'。一个NAND门(例如,203)具有可以被看作是对测量单元的"设置"输入的第二输入,而另一NAND门(例如,205)具有可以被看作是对测量单元的"重置"输入的第二输入。注意NAND门本质上是负逻辑设备,当"设置"输入被激活至逻辑"0"并且"重置"输入在逻辑"1"处被去激活时,测量单元的主输出(Q)假定逻辑"1"的状态,并且当"重置"输入被激活至逻辑"0"且"设置"输入被保持为高时,Q输出假定逻辑"0"的状态。而"设置"和"重置"信号可以是单独地并且根据需要可以被独立地使用(即,根据设计),在一个实施例中,这两个输入被接合在一起(即,不是被单独地使用),并且作为单个"激励"输入211被一起使用。当"激励"输入从浮点或高z状态或逻辑"1"中被取出时,即,保持逻辑"0",它使测量单元跳转到"非法"状态,即,输出Q和Q-bar都假定逻辑"1"的状态,并且当该信号被释放时(诸如再次向左浮动,或保持在逻辑"1"处),就会出现竞争条件;然后,每个输出在倒逆基础上假定为逻辑"0"或逻辑"1"。这发生在"激励"信号被释放后的很短的时间段,例如大约纳秒之内或更少。在两个输出均达到稳定状态之前的随后的时间段内,稳定衰减或某种形式的阻尼振荡可能会发生。如前所述,虽然由于制造工艺角、位点变化、设计偏差或其他不可预测性的来源,由输出Q/Q-bar假定的倒逆状态例如理论上应该是不可预测的,但是当"激励"输入被去激活时(即,保持逻辑"1"或高z),每个测量单元的输出将倾向于采取具有混合概率的相同逻辑状态(逻辑"0"或"1")。于是这提供了可以为每个测量单元提供潜在地唯一结果(即,逻辑状态和/或频率或采用相同状态的其他统计倾向)的PUF。注意,与其他设计不同(例如,诸如基于内置SRAM阵列,该阵列仅在上电时被感测,然后被用于存储数据),图2A中描绘的设计可以被动态地激励和测量。换言之,当需要测量多稳态单元201时,对IC和/或PUF阵列(或存储器阵列)上电,并且"激励"信号简单地被脉冲(例如,个体地针对给定的测量单元,或者同时针对一组单元,诸如行、列、其他组,甚至是整个阵列;为引用示例,在阵列配置中,当任何测量单元被"激励"并且其状态或倾向被重新测量时,行访问电路装置、采样锁存器和其他部件可以保持通电。图2A本身的测量单元中也没有时钟控制元件,并且所描绘的电路非常快,即,为每个测量迭代提供纳秒级响应。为了提供动态PUF功能,制

造简单地将具有所需功能特性的一组多稳态单元构建到所讨论的的相关IC设计中，并且，制造商还提供电路装置，以用于在“激励”信号的去激活之后足以使输出状态稳定的一段时间后对测量单元的响应进行采样和/或处理。虽然许多设计是可能的，但是图2B介绍了具有双稳态PUF单元的行和列的特定的阵列设计(参见图2B)，其中这些单元的行可以同时被测量(即，同时)。如前所提及的，其他亚稳态电路设计也可以被用于在所描绘的亚稳态电路中创建竞争条件，该亚稳态电路包括例如NOR门、锁存器、反相器和/或其他元件或电路。

[0062] 图2B是示出了一组或多组亚稳态电路的电路图，这些亚稳态电路可以被同时测量，例如，根据设计，可以作为整行或整列被同时测量。图2B示出了具有双稳态单元的第一行243和双稳态单元的第二行253的阵列241。双稳态单元可以可选地与图2A所介绍的设计相同(例如，交叉耦合的NAND门设计)。单元的第一行243被示出为具有多个单元245、246、257……248、249(例如，16、32或其他一些数目)，并且单元的第二行253被示出为具有相同数目的双稳态单元245’、246’、247’……248’、249’等。图中所描绘的亚稳态单元中的每一个都具有如上所介绍的“Q”输出，并且这些单元中的每一个都被耦合至特定行共同的“激励”信号，即，行243共享的一个激励信号(即“激励-1”)和行253共享的一个激励信号(即“激励-n”)；每个激励信号可以(通常是)在离散的基础上被单独驱动(即在不同的时间)，以测量对应单元行的输出。同样，虽然图中的每一行只描绘了五个单元，但省略号250和250’表示作为设计选择可以包括的任何数目的单元。在一个实施例中，例如，其中768个亚稳态单元被用于阵列中，在行中，每行可以具有32个单元，但这又将根据设计实现而变化。每个双稳态单元以针对每行都是唯一的但被跨行连接的方式被连接到复用比特线；因此，单元245和245’被有效地复用到比特线255上，单元246和246’被有效地复用到比特线256上，单元247和247’被有效地复用到比特线257上，单元248和248’被有效地复用到比特线258上，并且单元249和249’被有效地复用到比特线259上。在每行具有32个亚稳态单元的情况下，通常将会存在32个这种复用比特线连接(注意，可以存在比32个多的复用比特线连接，例如，在其中同时测量Q和Q-bar输出例如以提供差分读数或其他方式的设计中，可以存在64个复用比特线连接，针对阵列中单元的每一列，Q和Q-bar各有一个复用比特线)。另外，虽然为了简化讨论，只描述了两行，但是可以存在任意数目的行，如由省略号254表示的。因此，在一个可选实施例中，可以存在24个这种行(并且因此存在24个不同的“激励”信号线，每行一个“激励”信号线)。针对所描绘的阵列，每个“激励”信号线在不同的时间被脉冲化，并且接着与双稳态单元相关的输出被门控到对应的比特线上，然后所有的比特线同时被读取至整行的读取比特(即，32比特)，并且然后当读取下一行时，第一行的“激励”信号保持非活动(并且第二行之后的所有其他行保持非活动)，并且然后第二行的“激励”信号被脉冲化，等等。以这种方式读取所有行的结果将是读取所描绘阵列的所有比特，以获取所描绘阵列的硬件签名的读数。再次注意，这样的阵列可以被配置为具有任何期望的尺寸或配置，例如，可以具有单个非常长的单元行(即，整个阵列使用单个共同的“激励”信号来感测)，或具有被单独感测的单元(即，每个单元具有不同的专用“激励”信号，或它与其他行共享“激励”信号，但是具有单独控制的输入，例如，该输入使用晶体管被选择性地切换)、或前述的任何组合或排列。作为一个非限制性的示例，具有64行并且每行具有64个单元的阵列将产生4096个比特，这些比特可以被测量，并且通常被用于导出具有相同比特数目的硬件签名。注意，虽然该表述表示，在典型设计中，硬件签名的每个比特都仅是阵列的对应单个亚稳态单元的

一个或多个输出或输出的测量的函数，但是这不是所有实施例都需要的，例如，在理论上可以具有结构化的签名，使得其中比特依赖于两个或更多单元的输出，否则或者是以两个或更多单元的函数。

[0063] 在描绘的阵列中，在每个“激励”信号恰好被脉冲化后(针对任何给定的行)，寄存器261采样并保持每个比特线的状态(例如，如果每行存在32个双稳态单元，则保存32个或64个比特线)，并且然后经采样并保持的输出被提供给列计数器263的集合。如标记为“φ”的信号所指示的，寄存器和列计数器两者可选地可以在时钟控制的基础上被操作(备选地，该时钟“φ”可以被分解为所描绘的针对这些电路中的每个电路的锁存信号或使能信号，或者它可以根据设计完全被省略)。列计数器被用于聚合(aggregate)多个PUF测量，诸如，例如通过对每个比特的Q输出值为逻辑“1”时的发生数目进行计数，其中每个Q输出在给定行中对单元的对应“激励”信号的每次独立脉冲之后已经被合理地稳定，并且如果需要，计数值可以然后针对每个单元进行平均化或进行其他统计处理。换言之，对于任何给定的亚稳态单元，无论是由于电压或电容效应或其他因素，都不能保证每次激励对应的行时都会产生相同的输出(逻辑“1”或逻辑“0”)。在一个实施例中，每个单元被测量若干次(例如，20次或更多次)，使得每一行的测量的输出将是数目的(例如，0到20)对应阵列；在一个设想的设计中(下面将进一步讨论)，该数目是120或更多。在实践中，通常观察到的是，工艺角(和其他变化的来源)的效果有足够的影响，使得大多数亚稳态单元将倾向于产生“零”(例如，20个测量的计数接近于“零”)或“一”(20个测量的计数接近于“20”)，而双稳态单元的小的子集将产生中间值(例如，20个测量的计数在这两个值中间(例如，接近“十”))。这些计数可以从一个组测量变化为另一组测量，例如，给定的单元可能一次产生“12”的计数，另一次产生“14”的计数。注意，在所描绘的设计中，单元或它们的一些子集还可以被用作随机数生成器的基础，如下面所描述。所描绘的列寄存器261和列计数器263被示出为还具有“锁存”(列寄存器的操作被驱动以便于采样各种比特线)、“计数”(选通该信号使得列计数器263加载寄存器261的输出并且将其逐列地添加到现有的经计数的内容)、“重置”(每列的计数器被归零)和“读取”(由列计数器保持的总数目的输出被使能)的输入和输出；所描绘的列计数器263还馈送数据输出以为每个列提供聚合计数，即根据设计，数据可以被并行地或顺序地输出。

[0064] 因此，在一个实施例中，所描述的阵列电路装置按顺序测量亚稳态单元的每一行，但一次测量一行，以便同时共同地测量该行中所有单元的状态。例如，针对行243，行控制电路装置(图2B未描绘出)重复地脉冲该行的相关联的“激励”信号。在每个脉冲之后，但在下一脉冲之前，寄存器261的锁存输入被使能或被脉冲化，使得每个比特线的锁存器或其他采样和保持电路采用对应比特线的状态(即，由行243的对应的亚稳态单元输出的比特)，并且列计数器263被控制以便将所保持的值(逻辑“0”或逻辑“1”)添加到该比特线的累积计数。在一组测量之后(例如，在以上的示例中为20，并且作为参考在一个示例中为120)，单元的每列的列计数器263聚合该列的逻辑状态的计数，经聚合的计数被读取和存储，列计数器263被重置，并且阵列控制电路装置继续进行到下一行(例如，行控制电路装置被控制，使得下一行和下一“激励”信号被使用以类似地获取下一行的一组测量等)。

[0065] 下面将进一步讨论使用以上讨论的示例性方案的数据输出(即，768个亚稳态单元中的每一个亚稳态单元的多个、聚合的测量)以及如何从每个单元的聚合输出计数导出硬

件签名值的示例过程。相对于图2B,应该注意,一个实施例提供了支持可以动态地(即,在任何期望的时间,可选地包括上电时)进行测量的PUF的硬件阵列,这种测量可以在无需对所描绘阵列断电的情况下进行,并且可以根据设计对每个单元进行单个测量或多个测量。阵列中的每个单元都是如本文一般所讨论的多稳态单元,并且可选地是基于交叉耦合的NAND设计的双稳态单元,即,如图2A中所图示的。

[0066] 图2C示出了具有这种阵列273的集成电路(IC)271的实施例。该IC通常是如前面所介绍的封装的或未封装的裸片,并且还具有板载的一个或多个其他电路结构,例如,一个或多个处理器或处理器核275、加密/解密电路装置277、嵌入式存储器(例如易失性或非易失性)279、以及用于经由一个或多个导电信号线外部地通信的接口电路装置280(例如,可选地分离的地址和数据总线、或一个或多个串行链路)。还示出了多个电源输入,均被标记为"V-in",例如,总体地向裸片供电的一个或多个外部引脚281、向亚稳态阵列273(和相关联的阵列控制电路装置和PUF测量电路装置)分发功率的一个或多个路径282、向一个或多个处理器或处理器核分发功率的一个或多个路径295、向加密/解密电路装置分发功率的一个或多个路径296、向嵌入式存储器279(例如,在一个实施例中,SRAM)分发功率的一个或多个路径297、以及向接口电路装置280分发功率的一个或多个路径298。在一个实施例中,IC 271上的板载电路装置可以单独地关断对IC电路装置的子集的供电,例如,选择性地从接口电路装置或加密/解密电路装置中去除供电。注意,无论如何,由于亚稳态阵列的设计,相关联的PUF可以在任何时间被动态地测量,而无需对IC或亚稳态阵列或其控制电路装置的任何集合进行循环供电;即,例如,当多稳态阵列273被激励和测量时,向可选的嵌入式存储器279供应的功率可以为"开启"状态,并且同样地,当多稳态阵列273的一个或多个单元被测量时,多稳态阵列273的支持电路装置可以被供电。相反地,当关断对其他部件(诸如可选的接口280或嵌入式存储器279)的供电时,多稳态阵列可以被选择性地读取。作为其一个示例,即使当相关联的处理器或IC处于非活动、待机或其他功率节省状态时,也可以读取PUF。所描绘的IC也无需为了执行PUF测量而转储或移动嵌入式存储器279的内容。

[0067] 为了在所描绘的设计中执行PUF测量,处理器或处理器核275中的一个处理器或处理器核向亚稳态阵列的阵列控制电路装置发出"测量"信号287。然后,排序电路装置285编排各种测量函数的定时,包括亚稳态阵列的每行的顺序激活(即"激励"或脉冲)以及该信号和随后的测量的相关联的定时。针对所期望的测量数目,行控制电路装置286响应地被控制为以期望的方式脉冲每个特定行的"激励"信号,并且读取电路装置288被控制以便聚合每个亚稳态单元的测量数目(例如,如以上所例示的,1次、20次或120次或根据需要每个单元的某个其他数目的测量)。这些读数被提供给硬件签名导出电路装置289,该硬件签名导出电路装置289基于PUF读数计算硬件签名(如前所述,在一些实施例中,统计过程被用于计算签名,例如,基于指定的一组或一系列测量,诸如对于每个亚稳态单元采取120个或更多读数/测量以及随后的统计处理的多于一个迭代)。然后测量的硬件签名被输出以用在各种应用291中(诸如根密钥恢复292、随机数生成293、使用加密/解密电路装置277的加密应用等)。

[0068] 图2A至图2C已被用于描述可以被应用于PUF测量的电路装置。如参考图1A-图1D中所介绍的,所描绘的电路装置可选地可以与根密钥计算和硬件签名回滚结合使用;然而,这些电路还可以被应用于其他应用。类似地,相对于图1A至图1D介绍的根密钥计算和回滚过

程可选地可以与图2A至图2C所介绍的电路一起使用,但是可选地可以取而代之使用其他电路/硬件。

[0069] 再次注意,PUF阵列(诸如以上在图2A至图2C中所描述的PUF阵列)可以被用于硬件随机数的生成以及硬件签名的测量。在一个实施例中,共同PUF阵列(例如,基于相对较大数目的单元,诸如前面所介绍的768-元件阵列)还可以被用于随机数生成和易失性(或非易失性)密钥恢复(即,硬件签名测量和输出)。在一些设计中,第一PUF专用于硬件签名测量,而单独的(即,独立的)PUF阵列(例如,具有类似数目的单元或其他一些数目,诸如256个)被用于随机数生成。下面将参考图7进一步讨论随机数生成的详情。

[0070] 图3A至图3E将被用于讨论与使用PUF以提供硬件签名以及与使用证明曲线以允许在设备的整个寿命中进行根密钥的恢复相关联的特定实施例。

[0071] 图3A示出了具有测量PUF和恢复易失性(或非易失性)根密钥的内置电路装置的集成电路("IC")301。IC包括标记为"分子签名堡垒(SMB768)"的电路装置块303,其是768个单元的PUF阵列(例如,相对于图2A至图2C所介绍的基于交叉耦合的NAND门和支持电路装置)。首字母缩写"SMB"表示"智能存储器块"。在所描述的实施例中,该电路装置块具有24行单元,其中每行有32个亚稳态单元(如以上所例示的)。自然地,还可以根据实施例使用其他配置通过非限制性地示例,可以使用每行其他某个数目的单元和/或可以使用不同的列组织(例如,16个48行的单元,64个12行的单元等),和/或可以使用不同于768的聚合数目的单元。768比特硬件签名被使用在一些实施例中来使能256比特安全强度,这提供对蛮力的全枚举攻击的抵抗。如果需要,安全强度可以增长超过256比特,例如,在亚稳态单元阵列的实现中,以与总比特计数成比例的方式。如以上各种示例所指示的,PUF阵列的构造和组织的详情将主要是设计选择的问题,并且通常将依赖于如下因素:诸如所需的签名比特数目、底层IC架构、制造工艺技术、所需的安全级别和其他因素。在所描绘的实施例中,电路装置303提供768比特的硬件签名值,该值是多次测量的结果,其中每个单元的输出经过处理以标识提供逻辑"1"或逻辑"0"的倾向。如前所述,这些单元的倾向以及因此与每个单元相关联的签名的比特可以随时间而改变。

[0072] 所描绘的IC 301还包括标记为"分子随机数生成器"的电路装置块305,其中是另一PUF阵列,在该情况下具有256个单元。在所描绘的实施例中的电路装置块305还基于以上在图2A中所介绍的上述交叉耦合NAND单元设计。该电路装置305产生256比特值(或者256字的阵列或字符串,例如,每个字具有在0至120的范围内不可预测地变化的内容),该256比特值独立于由电路装置303产生的768比特签名。然而,如下面将进一步讨论的,电路装置块305还包括处理电路装置,它从该电路装置中产生期望长度的随机数。例如,该电路装置可以以产生雪崩或比特扩散效应的方式向测量的256比特或256字值哈希、加密、或以其他方式应用函数(例如,由电路装置输出的随机数随着许多比特改变状态而从根本上改变,即使256比特签名的仅单个比特在测量之间变化)。例如,该电路装置可以应用函数,其中值(例如,所需的长度的值,例如,一个或多个256比特的签名测量被组合和/或串接和/或链路在一起)可选地与一个或多个其他前端或后端填充值一起,并且然后经受CRC32划分、AES或SHA-256过程以产生期望长度的输出;作为其仅一个非限制性的示例,256比特的测量可以被链路(例如,以形成512比特长度值),然后其以产生256比特余数的方式经受CRC32划分处理。一般地,参见在共同拥有的美国专利号9635011中描述的与雪崩效应生成相关联的技

术,其通过引用被并入。CRC32划分、AES和/或SHA-256过程创建的输出随着由不同的输入字符串提供的轻微的熵而根本上改变-因此,即使假定256比特的PUF阵列倾向于产生抽象相似的输出字符串256,但足够的比特将改变状态或具有变化的输出,这些哈希过程将从相应测量生成看似完全独立的数目。在一些实施例中,电路装置305被用于产生字值(例如,0-120),该字值提供高阶熵、或不可预测性;例如,即使比特单元在大多数时间在统计上输出逻辑“1”,并且因此当测量120次时倾向于产生“120”计数,但是该聚合测量可能会随着每次的测量而变化(例如,有时119次,其他时间120次)。本领域普通技术人员将想到随机数处理和产生的许多变型,例如,在一些实施例中,还可以使用其他类型的随机数生成器,并且用于随机数生成的PUF阵列(或第二PUF阵列)的使用是可选的。作为一个非限制性的示例,如在本文其他位置所描述的,如果这种双重使用的实践不妨碍电路装置块303的标称功能,则块305所需的PUF阵列功能可以由还被用于电路装置块303的768比特的PUF阵列所提供。在所描绘的IC的情况下,如下面将进一步描述的,由电路装置305产生的随机数可以被用于生成一系列向量,该向量被用于证明曲线的生成、用于IC的处理和/或事务需求的私密密钥或秘密密钥的生成、和/或对某些存储器操作数的加密。

[0073] 所描绘的IC还包括标记为“精确密钥取回(ACC512)”的电路装置块307。该电路装置307为多个证明曲线中的每个证明曲线生成证明曲线加密数据(例如,用于768比特签名的512比特数据,其中x＝256,和y＝512,以及用于分区的加密数据),并且然后它还使用取回检查点和相关联的证明曲线数据来执行与IC301链路的原始硬件根密钥的精确恢复。然后,经由路径310(例如,内部总线)向标记为“密码服务模块(CSM)”的电路装置块309提供经恢复的根密钥或其导出后代(descendant)中的一个后代、或先前由根秘钥加密的经解密的秘密密钥,该电路装置块309执行加密和解密服务,其可选地加密用于外部存储的私密密钥或秘密密钥,并且使用经恢复的根密钥解密私密密钥或秘密密钥;经解密的私密/秘密密钥可以用于IC301的加密操作中。备选地或此外,在一些实施例中,经恢复的根密钥可以被直接用于加密操作以及处理操作数和/或指令和/或将由IC301在外部交换的其他数据;注意,在这种实施例中,可以期望以某个确定性或非确定性的方式混淆或修改根密钥,以便使根秘钥不可识别(例如,使得看起来好像不同的密钥用于加密将被外部传输的数据)-在这种实施例中,为了该目的,“屏蔽”密钥或“混淆”密钥(例如,图6中的元件621)可以被加密并且被外部地存储以允许密钥恢复。如通信箭头311所指示的,密码服务模块可以根据需要命令根密钥恢复,例如,在检测到密码差错的事件中,在IC上电时,或以其他方式;在一个实施例中,密码服务模块还负责触发点检查功能。作为非限制性的示例,在一个可能的设计中,当IC301被开启时,可以测量和直接应用768比特的硬件签名,其中只有在检测到密码差错的情况时(比如,经解密的私密密钥、秘密密钥或编程比特流发起差错),回滚/检查点恢复才会触发;备选地或此外,例如,基于特定基础,这种差错可以被用来触发新检查点的标识和存储,即,差错被检测到,其被用于推断至少一个比特的漂移,并且这使得新检查点被存储。被本领域的普通技术人员将想到许多示例和变化。还在一些实施例中,密码服务模块309可以基于随机数生成器305的输出生成新的私密密钥或秘密密钥;例如,如果IC上的应用需要临时的秘密密钥,那么为了该目的,特定长度的随机数可以被产生,并且使用设备ID或私密密钥进行加密和交换(例如,使用Diffie-Hellman密钥交换技术,无论使用经典版本还是基于椭圆曲线的变型)。在一个实施例中,加密服务模块可以生成和维持任意数目的私密密

钥/秘密密钥或相关联的设备ID，例如，生成它们、从外部接收它们和/或加密用于外部非易失性存储装置的私密密钥或秘密密钥元件。

[0074] 标记为“暂存缓冲器”的电路装置312被用于提供易失性存储器处理/存储空间，以“缓存”任意量的受保护的易失性或非易失性存储器/密钥，以及密钥恢复、点检查和/或密码功能中的每个功能。在一个实施例中，该电路装置可以包括SRAM单元的阵列，其为操作数和值提供内部易失性存储器存储。在一个实施例中，当根密钥被恢复时，它被存储在该电路装置中以供密码服务模块309使用。当将功率从IC 301或电路装置312移除时，电路装置312的内容丢失。在恢复功率之后，根密钥可以被恢复并且被存储在该电路装置309中以用于支持密码操作。

[0075] 图3A还示出了标记为“接口控制单元(ICU)”的电路装置块313。该电路装置编排用于IC内的一些或全部电路子块的操作，并且向IC 301外部的目的地发送和接收通信。在一个可选实施例中，一个或多个导电路径被用于直接将这些通信与这些目的地耦合。一个或多个导电路径可选地可以包括分离的命令/地址总线315和数据总线317，或者这些功能可以备选地被组合成单个总线(例如，315)，诸如串行通信链路，其发送和接收分组化通信的串行通信链路。例如，在各种实施例中，所描绘的总线315和317中的一者或两者可以与通信标准(诸如ATA、SATA、USB、SPI、显示端口、PCIE和/或其他发布的标准及其各种版本(例如SATA、USB、PCIE版本2.0、3.0等))兼容。

[0076] 最后，注意，如本文中经常引用的，所描绘的IC还可以可选地包括使电路装置执行所描述的功能的一些或全部功能或它们中的任何部分的指令。换句话说，在一个实施例中，IC 301的各种部件以及它们的相关联功能仅仅由硬件逻辑组成；在其他实施例中，所描述的电路和/或相关联功能中的一者或多者可以由通用电路装置部分或全部地提供，该通用电路装置经由指令逻辑(诸如固件)进行配置或控制，以便执行所描述的功能、操作或过程。将指令逻辑与通用目的或可配置电路装置组合使用的选项在图3A中由“软盘”图标319来象征性地表示，即，表示使用指令(即，并非强制任何特定格式的非暂态介质)。

[0077] 图3B是图示了用于例如基于PUF的测量导出硬件签名的功能的一个实施例的流程图。图3B所表示的技术总体上使用附图标记321来指定。指令或硬件逻辑发出“测量!”命令323以导致读取亚稳态单元的阵列。正如过程块325和326所指示的，该读取可以被逐行执行，每行的单元数目由阵列配置来指定，例如，同时测量32个双稳态单元(如上面讨论的实施例所表示的)。测量可以被执行多次，例如，每个单元被重复地执行120次，由附图标记327表示。如文本框329所指示的，聚合的测量可以被表示为每个单元的聚合计数，例如在00(十六进制)和78(十六进制)之间的值。例如，文本框329示出了24行(0001:至0024:)中的每行的清单，其中每行具有32个双字符值。例如，针对第一行(0001:)，可以看到第一单元具有十六进制的假设计数78(在第一虚线椭圆330内突出显示)，其对应于聚合计数值120；该计数指示，针对120次测量，第一行(0001:)的第一单元每次都产生逻辑“1”输出。相比之下，可以看出第三行(0003:)的第一单元的聚合计数值为0(在第二虚线椭圆331内突出显示)；该计数指示，针对120次测量，第三行的第一单元(0003:)每次都产生逻辑“0”输出。不是所有的双稳态单元以如此可预测的方式表现，例如，在附图中，第三和第四虚线椭圆(332/333)分别突出显示其他值，由椭圆332标记的值“01”(十六进制)表示，针对120次测量，第一行的第11单元几乎总是产生逻辑“0”输出，但产生一次逻辑“1”输出，并且由椭圆333标记的十六进

制的值“14”(十进制值为20)表示,针对120次测量,第二行的第16单元在大多数情况下产生逻辑“0”输出,但是在120次测量中的20次测量中产生逻辑“1”输出。这种结果是典型的,因为许多单元将倾向于产生“全为1”或“全为0”的结果,而一些单元子集将产生其他值。如附图标记335所指示的,如果需要,每个单元的120次测量的每个这种“帧”都可以被重复,以产生附加的签名值,或者用于统计处理函数以测量硬件特性。在一个实施例中,由附图标记227表示的签名在22帧中被重新测量,每次对每个单元进行120次测量,因而每次将对该阵列的每个单元进行2640次测量。应该注意的是,测量的数目、以及测量是否在多个“帧”中被执行,是实现的选择。一般来说,执行多次测量的一个原因是为了统计地平均倾向,但是,无论基于直接平均或其他方式,任何类型的所期望的统计处理都可以被执行。例如,利用每个单元的多次测量,每个单元的统计特性(诸如均值、平均数、加权平均数、标准偏差、离群值群体、其他扩展度量(例如,2σ、3σ等)等)可以被计算和被用于增强签名导出。在一个实施例中,针对每一帧,利用如框340中图形化说明的假设结果,可选地产生直方图(按照附图标记339)。每个个体单元的测量根据产生的值被归仓(例如,从00至78的十六进制的值可以被划分为12个范围,以产生粗略的概率测量,其中基于仓的内容计算加权均值)。例如,可以看出一个突出显示的单元(由圆形虚线341包围)具有在最右仓中的与逻辑“1”相关联的所有值。可以看出第二突出显示的单元(由圆形虚线342包围)具有中间结果,其值被归仓在逻辑“1”和逻辑“0”之间稍微偏向逻辑“1”。在一个实施例中,对22帧中的每一帧计算直方图,并且计算并使用扩展度量(或“离群值”度量),以丢弃在低端和高端中的每一个上的22个测量中最差的3个,然后计算并使用剩余帧的均值(直接值或仓出现计数),以确定签名值更倾向于逻辑“1”或逻辑“0”。在一个实施例中,即使对于明显处于中间的结果,也做出关于对应亚稳态单元倾向于产生逻辑“1”或逻辑“0”的决策,即,在此基础上标识和解析签名比特,如附图标记343所指示的。当然,许多不同的算法可以被使用,并且可以基于平均值、滤波、样本丢弃、加权概率、标准偏差和/或其他统计处理344。这种类型的处理有助于最小化随机统计偏离,该随机统计偏离可能不表示根密钥的漂移或在测量的硬件签名中普遍存在的差错。所说明的处理结果是具有若干比特(在本实施例中为768比特)的测量的硬件签名,表面上为所讨论的特定IC提供指纹。

[0078] 图3C示出了可以被用于生成检查点和相关联的证明曲线以标记硬件签名的过程的一个实施例。由图3表示的技术一般地由附图标记351表示。

[0079] 更特别地,当逻辑确定是时候创建检查点以标记特定硬件签名的状态时(由附图中间的框345表示),则由图3C表示的过程被触发。结合图5讨论了逻辑可以做出或基于该确定所依据的各种标准;为了图3C的目的,应当假定检查点基于测量的硬件签名被生成。这种签名可以是刚刚讨论的签名测量过程的产物,例如,单元的重复、动态测量被执行若干次,并且统计处理被执行以标识表面上与根密钥对应的比特集合;测量的输出由框345来表示。

[0080] 注意前面提到的,在一个实施例中,公开的技术可以针对每个检查点生成5个证明曲线,每个曲线表示不同的加密参数(例如,每各曲线有效地表示不同的加密过程,即,具有不同的曲线参数)。在一个实施例中,检查点生成逻辑(例如,硬件和/或指令逻辑)的第一任务是生成五个对应的转置(或分类)向量,即,为每个证明曲线提供用于对测量的768比特签名的比特进行分类的不同的标准;这些中的每个将被用于将硬件签名的比特混洗成专用于证明曲线的分类顺序,从而提供五种不同的混洗排列,这有效地创建了五个不同的非多项

式曲线作为下面描述的加密过程的函数。然后硬件签名的经重新排序的比特被处理以获得每个对应曲线的加密数据。对每条曲线的这种个体化的比特进行重新分类(或转置)提供除简单地允许创建多条不同的曲线之外的优点,即,1)它主动地将比特从其原始位置分散使得聚类的差错将不会集中在块分割曲线的单个块中;2)它去除了试图分析和关联多个检查点曲线数据的相关攻击的可能性,例如,通过试图定位相似点和偏差。

[0081] 每个转置向量具有与图3C的框359所表示的示例性转置向量类似的通用形式(即,它具有对应于操作数的比特或位置的一些条目,这些操作数将被排列或转置)。例如,该假设的转置向量指示,通过如下而获得一个证明曲线的数据:将测量的硬件签名的第8比特映射到证明曲线输入的第1比特(即,“比特0”),将测量的硬件签名的第344比特映射到下一比特(即,“比特1”),将硬件签名的第67比特映射到下一比特(即,“比特2”)等。预示已经以这种方式被重新排序的768比特的加密过程和相关联的曲线生成过程,这些比特中的最高1/3比特形成值x(由SMB-1内容表示,图中的元件362),并且这些比特中的最低2/3比特形成值y(由SMB-2的内容表示,图中的元件362)。x比特形成密钥值,该密钥值被用于根据证明曲线过程对y比特进行加密。通过针对每个检查点对多个曲线(例如,AC2、AC3、AC4、AC5等)进行这种操作,每个曲线的数据和加密过程都表示唯一的非多项式曲线,其中所有曲线都相交于一个共同点,即,由测量的硬件签名所定义的点。这将在图4A-图4C中被进一步讨论。注意,该表述并不表示这些AC曲线的子集不会相交于另一个或多个点。

[0082] 为了生成每个转置向量,768元组被首先获得。768元组是某个限定大小的、具有768个整数或字的向量。针对具有不同长度签名的其他实施例,即n比特,可以获得n元组。在一个实施例中,通过从PUF阵列接收测量、从该测量中生成随机数、然后使用某个函数对随机数进行处理以生成大的随机整数(例如,具有96次迭代的串接)来获得这些整数;然后采用有限域中的模运算以生成可变转置,并且对值进行分类。例如,在如前所述的一个实施例中,第二PUF阵列可以生成256字的PUF测量值,由数据输入353表示。该256字的值然后被提供给过程355,该过程根据需要提供比特扩散(即,雪崩)以及密钥/值扩展;即使重复的256字的PUF测量将与其比特值中的多个比特相关联,但仍将存在由状态改变或不可预测的单元引起的足够的熵,使得来自256字的PUF测量的这种“哈希”值将提供足够的熵以获得真正随机的数值输出。可选地,这种哈希值的状态可以被保留用于下一调用,以用作与新的256字的测量相串接的前缀,以生成新的哈希值。这种操纵提供了在继承历史熵的同时补充新熵的优势。例如,如框355所指示的,256字的测量可以被提供给提供256比特输出的哈希处理(例如CRC划分、AES或SHA-256处理),该哈希处理将256字的测量所表示的熵扩散到整个256比特的哈希输出。在一些实施例中,来自PUF阵列的输出与其自身串接或被填充,例如,以提供512比特(或512字)或更大的值,然后利用CRC32/AES/SHA-256处理对该值进行操作并且生成余数,该余数为随机数。在该示例中,应该假定期望获得768个相对大的整数。这可以通过获得来自CRC32/AES/SHA-256随机数生成过程的多个(如96个)不同的256比特的随机数来获得,即,在该示例中,每个随机数提供32字节的信息,并且因此96个随机数提供共3072字节、或者为768个值中的每个值提供4字节的整数(即,每个整数的值为0至2^{32}-1)。按照过程357,提前选取的、并且具有比特中近似长度大小的签名或者更大长度的签名的素数接着被用作有限域运算符,以应用模运算并计算每个整数值的余数,然后对对应余数的分类被用于确定转置向量。例如,如果测量的签名具有768比特的位置,那么可以选择优选大

于500至1000的素数。这里的示例可以是有帮助的:如果假设所选择的素数(和有限域运算符)是2423,并且4字节的768个整数中的前三个整数是3633287885、1504121945和1072682973(即,以十进制),则这些数字的余数分别为1808、1081和2135。以这种方式获得的768个余数(即,来自随机数的产生、串接、和模运算)然后以数值升序被分类以产生转置顺序。再次使用刚刚给出的示例(1808、1081和2135),通过应用生成的分类向量,可以根据这些值的升序(例如,第二、第一和第三)对签名的前三个数字进行分类。所有768个余数被分类,并且被用于生成转置向量,使得所需的转置通常比仅具有三个数字的该简单示例复杂得多,例如,使用该基本过程生成的采样转置向量359指示,测量的签名的第八比特应该成为转置输出的第一比特,测量的签名的第344比特应该成为转置输出的第二比特,测量的签名的第67比特应该成为第3比特,等等。这种过程每次都基于随机数生成器的使用创建真正随机的转置向量。注意,对生成转置向量的上述描述只是示例性的,而且可以使用许多处理来生成转置向量。作为示例,发起硬件签名的相同的PUF阵列可以被用于生成转置向量,该选项由图3C中的数据输入329来表示,例如,768个数字(例如,如按照图3B的示例,来自单个测量帧的每个数字在0至120)可以被直接输入至CRC32/AES/SHA256处理,其中生成768个整数。备选地,可以从另一源(如由数据输入352所表示的)获得种子。在其他实施例中,如上所述,可以引入可选的随机化器(354)、或前面生成并保留的随机数,并且将其用于填充CRC32/AES/SHA-256处理的输入,即,进一步增加输入的熵并且产生不同的结果;例如,固件开发人员可以选择提供任意的、固定的前端填充(例如,时间戳)以追加来自PUF的输入,接着利用CRC32/AES/SHA-256过程从该输入生成余数。本领域普通技术人员将想到许多这种变型。一般地,参见通过引用并入本文的美国专利号9635011,尤其是其中从第10栏开始的关于生成转置向量的讨论。再次,为了在哈希/CRC划分之前,从相同的输入创建多个(非常不同的)转置向量,对于待生成的每个转置向量(例如,“转置向量1”、“转置向量2”、“转置向量3”、“转置向量4”和“转置向量5”的ASCII版本可以被用于该目的,代替在前述专利的图4中所描绘的次级密钥或辅助密钥),768元组可以被追加不同的随机化器值、或被追加前面步骤保留的随机数或字符串。由于过程355所提供的比特扩散以及从CRC划分/AES/SHA处理中产生余数的事实,即使针对辅助密钥中的这种变型所表示的细微的输入变化,其结果也将是从根本上不同的转置向量。

[0083] 针对每个转置向量,因此来自过程357的768值输出的值提供将硬件签名的每个比特混洗到经分类或经转置的输出中的不同比特位置的映射(即,如框361所示)。注意,按照附图标记377,每个转置向量和与证明曲线对应的经加密的数据一起都将以明文形式被存储。因此,为了生成检查点,生成五个唯一的转置向量,并且根据这些向量中相应的每一个向量生成五个证明曲线的数据。在加密过程中,每个经分类的数据集合(即,每个证明曲线的数据)一次使用一个,以创建证明曲线数据。

[0084] 针对每个证明曲线,所生成的经转置的数据首先将被划分为高阶比特组和低阶比特组。较高阶组的经转置的比特按照比特序被放置在第一智能存储器块362中(即,第一缓冲器SMB1),而较低阶组的转置比特则被放置在第二智能存储器块363中(即,第二缓冲器SMB2)。如可选的过程框365和366所示,几乎任何期望的划分方法都可以被用作实现选择的问题;例如,给出假定的768比特的签名长度,这些比特中前三分之一的比特(即,经转置的768比特中的前256比特)可以被放置在块362中,而这些比特中后三分之二的比特(即,经转

置的768比特中的后512比特)可以被放置在块363中;该分配由框365表示。根据需要,可以使用不同的划分,例如,两者各半(例如,在块362和363的每一个块中的384个经转置的比特),如框366所指示的。其他的划分也是可能的,包括其中SMB1比SMB2具有更多比特的划分。关于图3C的加密过程,SMB1(362)中的比特将形成加密密钥(按照附图标记369),而SMB2(363)中的比特将形成加密操作数(即,如按照附图标记371的)。如附图标记371所指示的,该实施例中的加密过程是格式保留加密(FPE)过程,该过程使用与美国专利号9635011所讨论的与图3C相关的步骤625、629、631和633类似的过程;换言之,来自SMB1(362)的比特被用作主密钥来生成另一转置向量,然后该转置向量被用于加密(混洗)SMB2的内容,以获得也具有与SMB2中的比特数目相同的比特数目(如512)的输出。根据需要,转置向量可以通过以下而被生成:重复或串接主密钥,并且然后再次执行CRC32划分和/或AES和/或SHA-256操作,以及使用密钥扩展技术来生成整数集合(例如,512),并且然后在有限域中执行模运算,正如上面所描述的。因此,例如,在该实施例中,如果SMB2(363)具有512比特,则512个值的转置向量被生成,以唯一地混洗SMB2中的比特。在该FPE过程的后续步骤接着创建随机比特翻转向量,该向量可以由主密钥导出、或(根据实现)独立于主密钥。例如,在一个实施例中,刚刚提到的512个整数被转换为二进制向量,该二进制向量针对512个整数中的作为奇数的每个整数具有逻辑“1”的比特值,并且针对512个整数中的作为偶数的每个整数具有逻辑“0”的比特值;当被应用于来自SMB2的经转置的512个比特时,在与512比特翻转向量的对应比特位置的逻辑“1”相匹配的位置处的每个经转置的比特接着被状态反转,以产生进一步修改的数据。最后,在FPE过程的第三步骤中,非线性置换可以被执行以将比特子集置换为该修改后的数据;来自比特翻转操作的进一步修改的比特按集合被提供给查找表,该查找表输出类似数目的比特(但潜在地具有极其不同的值),只要查找表将唯一的比特集合映射到唯一的输入,这种操作产生经加密的输出,原始的SMB2内容可以从该经加密的输出中被恢复。例如,512值的转置向量可以被用于创建九比特的非线性置换表,该置换表以1比1的对应(例如,分类位置被用于将二进制输入映射到二进制输出)将九比特输入的每个排列映射到潜在地非常不同的九比特输出,使得例如如果512个整数中的前两个整数分别是第289个最大值和第11个最大值,那么非线性置换可以将二进制输入000000000和000000001(即,512个二进制数中的前2个数)映射到二进制值100100001和000001011(分别为289和11)。FPE的非线性置换步骤的实用性之一是阻止差分密码分析,该差分密码分析有时被用于攻击本质上至少是部分线性的密码过程。FPE过程(例如,转置和/或选择性的比特翻转和/或非线性变换操作)的结果是对SMB2内容的加密版本,其具有与SMB2中的比特数目相同的比特数目,如可选过程块373所指示的。仅知道加密的输出不允许攻击者获得完整的硬件签名(或硬件签名的部分),即,768比特的原始硬件签名(由附图标记345表示)以及甚至SMB2的内容被转置,以至于是无法识别的,但是如果SMB1的内容已知,则经加密的输出可以被解密以恢复SMB2的内容。在一个实施例中,所描述的加密函数的该特性被用来隔离和校正漂移,即,用来创建曲线的转置向量(例如,以明文形式存储的768值的转置向量)被用于检查点恢复以创建新测量的硬件签名的相同转置,并且如果类似分类的x-比特(SMB1)匹配,则解密过程可以被标识,该解密过程将是反向加密过程,但是新测量的硬件签名和SMB1的内容仅被保存在与易失性(或非易失性)根密钥相关联的芯片中,并且因此只有该芯片可以被用于从外部存储的加密证明数据中恢复易失性根密钥。从密码学的角度来说,芯片相对于芯片

外部的攻击者具有不公平的优势，因为该芯片是了解由新的硬件测量提供的SMB1和SMB2的近似副本的唯一方。在某种意义上，经加密的输出(例如，曲线的512个比特)与哈希类似，因为它表示对完整的768比特硬件签名的压缩且经加密的导出；然而，不同于典型的哈希，考虑到已知SMB1的比特、转置向量(即，用于填充SMB1/SMB2)和用于生成比特翻转和非线性置换操作的任何唯一的信息，所描述的加密操作是完全可逆的过程。还要注意，如下面将进一步描述的，该函数是通用类型，其中c＝H(x,y)，其中x是SMB1(362)的内容，并且y是SMB2(363)的内容，并且c是加密输出(例如，存储在NVM中)；刚刚描述的处理产生c＝Fn{x,y}形式的输出，其中x是SMB1的比特，y是SMB2的比特，并且c是证明曲线数据372(并且可以被外部地存储)。如前所指示的，该“经加密”的数据(例如，AC2至AC5的512个比特)与具有任何相关报头的相关曲线(即，按照块377)的明文转置向量一起被存储，例如，以便标识特定的加密详情，并且以便允许根密钥恢复过程来将特定的检查点和特定的证明曲线与候选匹配，以为了稍后的检查点匹配。

[0085] 如过程块375所表示的，一个实施例中，针对多个(例如，5个)不同的证明曲线来重复刚刚描述的基本过程，以标记单个检查点，并且这些曲线一起可以被称为单个证明曲线集合。然而，在所描绘的方案中，证明曲线中的一条曲线被略有不同地处理，即将被用于逐块比较的该一条曲线(AC1)。换言之，如过程框364所表示的，在一个实施例中，在转置之后(例如，按照块361)，签名被划分为块或分区，并且然后每个块或分区被个体地进行加密。例如，如果块大小是42比特，则前14比特可以被用作主密钥，以完全按照刚刚描述的方式生成28值的转置向量、比特翻转向量和非线性置换表，在这种方式中，这些向量针对每个块被单独地标识。该尺寸(即，42比特)表示设计选择，其被选择用于数个与运行时间相关的优势；进一步注意，该过程不必包含所测量的硬件签名的所有比特，例如，在当前示例中的所有768个比特。例如，尽管第一证明曲线AC1使用专用转置向量，以对该曲线唯一的方式排列测量的硬件签名的768个比特，在一个实现中，只有随后的转置值中的前756个比特被用于生成曲线数据；该方法将这些756个比特划分为18块，每块42比特，例如，块1＝比特1至比特42、块2＝比特42至比特84、块3＝比特85至比特126、等等。这些块中的每个块表示测量的硬件签名的任何可能的子集，即，继续前面介绍的假设的子集，第一块可以具有测量的硬件签名的第8比特、第344比特和第67比特作为其前三比特，即，如通过转置向量分类的，如按照以上示例。然后，该方法以上述一般方式针对每个块进行，例如，前三分之一的比特(14比特)可以被加载到SMB1(362)，而该块的后28比特可以被加载到SMB2(363)；该划分由可选的过程块364表示。再次，该划分并非对于所有实施例都是需要的，并且它表示设计选择，例如，相反可以将768比特划分成24块，每块32比特(24*32＝768)，或者其他一些分配。该过程的输出是每块具有28比特的经加密的证明数据，稍后将在逐块的比较中将其用作硬件签名回滚的一部分，如上所述。

[0086] 虚线框374指示针对某些曲线(如AC2至AC5)的SMB2输入(在某些实施例中，和/或输出)可以被选择性地并且随机地屏蔽。换言之，在一个实施例中，在应用加密函数之前，SMB2中的多个比特可以被选择性地屏蔽和/或翻转(值被改变)。如前所述，在一个实施例中，在(后来)比较过程中，即在匹配检查点的过程中，可以容忍许多比特差错，例如，可以使用“模糊比较”过程，其中如果p个比特或更少的比特可以被进一步翻转(改变)来产生与SMB2内容相同的值，该“混淆比较”过程认为任何证明曲线都匹配；如下所述，即使考虑到该

差错和多个候选，给定适度长的硬件签名长度(如512比特或更多)，针对给定的检查点(即，即使具有这种有意注入的差错)，两个解将匹配所有五个证明曲线的这种概率也是极少发生的(即，作为实际问题是不可能)。因此，在一个实施例中，多达p个比特的随机差错被有意地注入到加密操作数中。注意，这种随机差错也可以被潜在地注入到输出中(例如，对于不使用非线性置换、但仅依赖于比特翻转和转置的过程，这种有意注入的m个比特的差错将同样地移载到稍后的解密过程中的恰好m个比特)。无论使用哪个过程，数据输出(即，证明曲线数据372)可以被外部地存储。在实验用例中，已经发现针对512比特证明数据和5个曲线，p可以达到36；换言之，利用5个证明曲线，即使仍然将36个比特的差错作为噪声添加到每个曲线上，多于一个的候选解将适合所有五个曲线在计算上仍然是不可能的(即，实际上是不可能的)。换言之，非正式的统计建模指示，示例性的证明曲线的混叠抑制能力提供了不经过刻意差错注入的2^{256}个混叠可能性减小划分器，以及针对高达36比特的差错注入提供不小于2^{220}个的减小划分器。该过程有助于进一步屏蔽来自潜在攻击者的证明数据，即，实际上使得无法读取(测量)PUF的攻击者可以反转证明数据来标识根密钥成为不可能。

[0087] 当该过程完成时，在存储装置中通过完整的证明曲线的集合来标记检查点，在由图3C表示的实施例的情况下，证明曲线为5个(AC1-AC5)。框378示出了针对这些曲线的存储的示例性格式，例如，每个曲线可以包括报头，该报头标识检查点数(或另一检查点标志符，例如，日期)、证明曲线集合的标志符(其中多组曲线集合被用于为给定的检查点提供冗余)、以及特定曲线(例如，AC1至AC5)的标识符；每条曲线还伴随有与转置向量(即，T1至T5)对应的明文版本，该转置向量被用于对原始硬件签名进行比特混洗；最后，每个曲线伴随有经加密的输出，即，使用上述的FPE技术产生的SMB2的该版本。仅作为参考，可以可选地针对单个检查点产生这些证明曲线的多组集合(即，每组集合5个曲线)，例如，以提供冗余或增加可以被有意注入的随机差错的比特数目(例如，以增加模糊比较过程可以容忍的差错)。

[0088] 注意，与上述例示的技术相关的许多变型被设想出，这些变型将容易被本领域普通技术人员所想到。例如，根据实施例，任何数目的证明曲线都可以被用于形成集合。在一个实现中，真实哈希被用于测量硬件签名的不同排列，而不是用于证明曲线。在另一实施例中，使用不同的签名长度和/或使用不同的加密过程。在另一实施例中，再次使用证明曲线，但使用不同的机制来对测量的硬件签名的版本进行不同的排列，或创建该签名的子集或其他产品，以用作过程操作数、或以执行加密。在另一变型中，如图3C中的虚线(可选)过程所示，所描述的加密操作的子集或排列被执行，例如，仅执行非线性置换、或仅执行比特排列或仅执行随机比特翻转或这些操作的某些组合。

[0089] 每个检查点提供了一种机制，以回滚稍后预期测量的硬件签名，使其具有与检查点相同(或类似)的状态。检查点数据可以被存储在片内、系统内、在NVM中(例如，在同一板上或同一系统中的不同芯片上)，还可以以远程可访问的方式(例如，通过局域网“LAN”或广域网“WAN”)存储。针对片内解决方案，一次性可编程(“OTP”)存储器元件可以被有利地用于存储一些或全部的早期检查点数据的简洁验证或哈希签名，作为以后的真实性验证的基础，使得后续用户或操作人员不能清除已建立的和“锁定”的检查点数据，或为了任何未经授权或非法的目的尝试替换检查点数据。例如，如前所述，在一个实施例中，IC制造商以经由英特网可访问的方式(即，下载)来测量硬件签名和检查点数据。当系统的用户或集成者运行配置过程时，IC通过经由WAN(例如，经由无线网络)下载和使用发布的证明数据，并且

使用取回的数据以恢复根密钥,从而恢复其根密钥。然后更多的检查点可以被建立并被本地存储在系统上(例如,在NVM芯片中),或它们还可以被远程地存储(例如,通过LAN或WAN以远程可访问的方式)。再次,本领域的普通技术人员将想到包括具有两个或更多个根密钥和相关联的检查点的系统(如前面所介绍的)的许多示例实现和功能。

[0090] 图3D被用于描述用于回滚硬件签名以匹配当前检查点的技术的一个实施例。技术由参考附图标记381一般地表示。如前所述,通常期望能够恢复无法在本地存储的根密钥,并且在从设备或IC上断电时,根密钥可能会丢失,并且接着根密钥必须被恢复。恢复过程可以是被自动启动(即,作为上电配置步骤的一部分),或响应于无法正常工作的测量的硬件签名的应用,即,某种类型的差错被检测到。一般来说,硬件签名首先被读取或测量(例如,PUF阵列被测量)以生成值。例如,这可以是参考图3A的过程,并且该结果值可以是768比特的值,如参考附图标记345所表示的。为了执行回滚,将被用于比较的先前检查点被标识,并且然后该检查点的检查点数据由IC从外部(例如,从同一系统中的NVM芯片或经由LAN或WAN从远程位置)被取回。在该情况下,取回的数据376被假定为在图3C中描述的过程结束时存储的检查点数据,并且包括五个曲线的证明数据和明文转置向量T1至T5,即,每个转置向量专用于每个曲线,如由附图标记所表示的。

[0091] 新测量的硬件签名(345)首先根据针对用于比较的检查点的取回的转置向量中的每一个进行转置比特,以创建五个单独的输出,一个进输出对应于先前检查点的证明曲线中的每个证明曲线。这由图3D中的功能块382表示。从理论上讲,对于相应的曲线,这些转置中的每个转置应该产生与图3C完全相同的SMB1和SMB2的内容,但是可以由新测量的硬件签名相对于所考虑的检查点反映的任何漂移。每个转置向量提供的分类、以及转置向量集提供的不同分类应该有效地分散和减轻影响硬件签名的几个相邻比特的任何集群差错,即,测量的签名的相关比特将以不同的方式排列,并通常在签名的转置版本中重新分配到可能非常不同的位置。

[0092] 然后,该方法将个体块(表示当前硬件签名比特的子集)与用于取回的先前检查点的对应曲线(AC1)的对应证明数据块进行比较,以确定是否存在匹配。再次假定先前检查点和新测量的硬件签名两者都使用相同的转置向量,新测量的硬件签名的每个块应该以与被用于为考虑中的检查点创建对应的证明曲线的顺序相同的比特顺序表示完全相同的比特子集;例如,使用图3C的框359中的示例转置向量,新测量的签名的第一块将填充新测量的签名的第8比特、第344比特、第67比特……等等。再次假定块大小为42比特,并且将这些比特的前三分之一比特分配给SMB1,则42个经转置的比特的前14个比特将被加载到SMB1,并且经转置的比特的后28个比特将被加载到SMB2。然而,在由图3C所调用的操作中,然后这些14个比特被用于导出图3C的加密算法的反向,例如,反向非线性置换表、反向比特翻转向量和反向转置向量,并且来自检查点曲线(AC1)的对应块的28个经加密的比特通过这些过程以反向顺序进行馈送。由于图3C的加密过程是可逆的,如果转置向量相同并且硬件签名相同,则该过程将产生与SMB2的内容完全匹配的经解密的28个比特输出。如果这两个值(即y匹配$y'=H^{-1}(x,c)$,其中y是SMB2中的数据,c是经加密数据,并且y'是经解密的数据),则当前签名和经点检查的签名的对应块的比特值被标识为匹配,并且过程继续比较下一对的对应块,如共同地由图3D的附图标记384和385来表示。如果该比较显示y和$H^{-1}(x,c)$不相同,则接着该方法以确定的方式进行,以试图标识导致不匹配的SMB1和/或SMB2的比特,一般按照

附图标记386至388表示的过程进行。换言之，按照附图标记386，该方法查询在被比较的28比特值之间是否存在超过j个不匹配的比特。应该回想起，转置/加密过程使用SMB1的内容作为对SMB2的操作数起作用的加密密钥，该加密密钥通过使用CRC32划分、AES或SHA-256处理被处理以创建比特扩散(和生成的雪崩效应)以及生成的转置向量(和/或比特翻转算法和/或非线性置换表)。该结构被用于试图标识差错在哪里，即，如果两个28比特值之间存在许多的差异比特，则该方法首先假定误差可能是1至2比特或SMB1中的其他少量比特(即，x比特被用作加密密钥)，其中比特扩散是导致被比较的两个值中的差异更大的原因。然后该方法在迭代的基础上将SMB1的比特一次一个切换，每次计算新的加密密钥和/或比特翻转密钥和/或非线性置换表，并且然后每次比较$H^{-1}(x,c)$与SMB2的内容，并且确定该值是否匹配(或接近匹配)。如果没有找到适当的解(即仍然存在大于j个不匹配的比特)，然后，该方法切换两个比特的不同组合，类似地尝试每个解、接着尝试三个比特的不同组合，等等。这个过程一般由函数块387表示。相反地，如果只有少量的28个比特值不匹配，则该方法假定差错在SMB2中，并且如功能块388所指示的，该方法可以接受y’作为当前块的可行候选点(x,y’)的低阶比特，该候选点被保留在记录中以用于进一步的全长度AC审查。针对以这种方式进行比较的每个块，该方法将仅标识一个候选、或少量候选，这些候选可以潜在地提供正确且完整的解决方案。例如，虚线框389示出了18元组值{1,1,2,1,1,3,1,1,1,1,1,2,2,1,1,1,1,1,1,1}，该18元组值表示，在假设的比较中，几乎所有18个块的按块比较产生了精确的匹配，但是块#3导致两个候选，块#6导致三个候选、并且块#11和块#12中的每个块都导致两个候选。因为逻辑具有用于创建AC1的排序向量，因此如果系统检测多于一个的候选(即，过程390)，则它可以将每个候选依次反向映射回到当前的(例如，新测量的)所分析的硬件签名，并使用该反向映射来构建完整版本的修改后的签名(例如，包括生成候选所需的任何更改后的比特；例如，如果块1包括签名的第344比特，并且该比特被修改以产生所讨论的候选，则当前硬件签名的第344比特将被对应地修改(即，在图中可以看出由包含“h-sign’”(h-sign.素数)的虚线框所表示的)。针对AC2至AC5的该签名的排列将基于完整(候选)签名来被计算(按照块391)，并且这些768比特值然后将与存储的检查点的类似曲线进行类似的比较，即，由函数块392所表示的。换言之，这些值将被划分为SMB1和SMB2，并且SMB1将被用于对检查点对应曲线的经加密的数据进行解密，并且再次，y(即，SMB2的内容)将与针对对应的曲线(如AC2、AC3、AC4和AC5)的$H^{-1}(x,y')$进行比较。由于块彼此“正交”，由于在该实施例中它们在硬件签名的相互互斥的比特集合上进行操作，因此如果对于块存在超过1个选择，则来自块的所有候选选择必须与所有其他块的候选集合“组合地”参与，以还原到用于最终验收的待检查的全长度签名候选。针对刚刚提供的18元组示例，这意味着所有2*3*2*2＝24个可能的组合都必须可以被完全枚举，每个组合潜在地被用于重构全长度签名候选，以便进行进一步的AC检查。该混叠-区分过程驱使对AC1的设计选择(诸如SMB1和SMB2的比特大小)的需求，以名义上地生成每块极少的可行候选，优选是每个块仅生成一个候选，使得在进行全长度证明时只需枚举非常少量的组合。如前所述，由于根据不同转置向量生成的AC2至AC5曲线的曲线参数不同，并且由于签名大小相对大(768比特)，因此在计算上不太可能(即，几乎不可能)存在跨所有5条曲线进行验证的多于一个解，当确定精确匹配时，就可以将其作为正确的先前的经点检查的签名(即，如由过程块394所指示的)。

[0093] 前面提到，在可选实施例中，例如，从图3C中可选过程块374看出，在证明曲线检查

点数据被存储之前，可以使少量随机比特屏蔽或反转。在这种实施例中，可以预期，一个或多个AC2至AC5曲线在被比较时不会产生精确匹配(即，由于有意引起的差错)。但是，如果比较过程导致的差错比特数少于预先确定的比特数d(其中，值d为实现选择)，则该过程可选地被设计为检测匹配。该值d表示可编程地汉明距离。即使针对相对大的d值(例如，36比特)，考虑到签名大小，在统计上仍然不太可能(即，不可能)存在两个或更多个结合地匹配横跨所有五个证明曲线的给定检查点的候选解决方案，上面描述的具体技术的设计也足够鲁棒。为了处理候选，该逻辑将候选从最可能到最不可能进行排序，并尝试匹配所有5个证明曲线，以标识针对个体曲线的经解密的数据是否在SMB2中的汉明距离值d之内(即，按照附图标记393)。由于只可以存在一个解，如果遇到任何的不匹配，系统继续到下一候选；如果通过落入在所有五个曲线的汉明距离内结果确实匹配(即，按照附图标记394)，则正确的先前经点检查的签名被找到，并且然后系统可以继续到下一检查点(即，如过程块395所指示的，它递减k并且循环，如图所指示的)。如果考虑中的当前检查点是第一检查点(即，对应于根密钥)，则根密钥已经被恢复(如过程终止块396所表示的)。该值接着被返回，使得它可以被应用于和/或以其他方式被应用于前面提到的密码操作中。在另一实施例中，如由块395所表示的，经点检查签名可以被用于直接加密(针对外部存储)原始根密钥，使得在经点检查签名被发现时，它可以被用于取回和直接解密该根密钥(例如，仅使用单个检查点从安全的NVM存储中)；如前所述，在一些实现中，可以期望不在外部存储任何版本的根密钥(经加密的或其他方式)，而是依赖于512比特的曲线数据(类似于768比特签名的哈希表示)来提供“线索”，该“线索”允许芯片硬件经由迭代回滚过程逐检查点地猜测根密钥。

[0094] 注意，以上存在的许多变型将容易地被本领域普通技术人员所想到。为了对此提供一个限制性示例，在一个实施例中，模糊比较步骤可以被省略，即，将汉明距离d编程为0，并且在存储检查点时不向证明曲线数据引入随机差错。在另一实施例中，经测量的硬件签名的所有比特作为基于块的第一比较的一部分被进行比较，例如，在使用768比特签名的情况下，可以将该签名划分为20个32比特块、32个24比特块等等(即，每个合计占全部768比特)；还可以使用上面这种设计，但是要处理19个42比特块(例如，针对最后一个块，可以填充签名的最后12比特)。在另一实施例中，来自块的比特可以彼此重叠和/或被组合。再次，本领域普通技术人员将想到许多变型，并且刚刚被讨论的技术应该被看作只是示例性的。

[0095] 图3E示出了使用示例数据可以更好地解释这些过程的两个文本框。该数据一般由附图标记397来表示。图左侧的文本框示出了假设的当前硬件签名的证明曲线数据，每行呈现对应于不同块的数据，并且示出了相关的测量和试图将该数据与存储的检查点数据逐块进行匹配的比较。注意针对第4行，其中列出了值DE749BD6 121D2410 82B2A3E5＞1＞18。如该文本框底部所指示的，在该示例中第一字段DE749BD6对应于SMB1中的比特(例如，被用作生成转置和/或比特翻转向量和/或非线性置换表的密钥)，而第二字段121D2410对应于SMB2。第三字段82B2A3E5对应于值$H^{-1}(x,c)$(即，将在下一段中解释)，而第四字段(即，＞＞18)指示存在18个差错比特(即，$\Sigma[y=SMB2 \oplus H^{-1}(x,c)]$)。

[0096] 该图右侧的文本框示出了与同一假设签名的先前检查点对应的正确的证明曲线数据以及针对每个块所产生的所存储的证明数据(c)和用于比较目的的值$H^{-1}(x,c)$。正如该文本框的第4块(第4行)所指示的，SMB1和SMB2的正确数据应该是DE749AD6 12D2410(十六进制)，其中经加密的输出c(或H(x,y))为BA7272E7。注意，在该行的右侧文本块中图示的信

息中，只有后者的值(即，c，使用参考附图标记398来指定)将被存储在外部存储器中，即，BA727E7。第四列的值(即，121D2410)与第二列完全匹配，如果从第二列中减去第四列(或与第二列异或的)，将产生零结果。

[0097] 在对新测量的硬件签名的第四块(AC1)执行证明曲线比较时，在左边的文本框中可以看到，该方法从外部存储器(即，BA7272E7)取回(加载)对应的证明曲线值，并且它执行反向转置(和/或比特翻转和/或非线性置换操作)，并且将该结果值与SMB2中的量进行异或处理。但是，反向转置y'与该值不匹配，并且可以看出，在两者之间存在18比特的差异。换言之，值DE749BA6被用作密钥，并且导致差错的反向转置向量，从而产生与SMB2的内容不匹配的值(398')，并且因此导致非零的异或结果。系统标志该差错；请注意，实际上，如通过比较附图标记399和399'参考的数据观察到的，实际上仅存在单个比特差错(即，对应于漂移)，但是系统尚未标识出该比特差错。检查点回滚逻辑确定性地进行，即，它标识出18比特的差异是很大的差异，考虑到总体为32比特，并且因此它假定差错在SMB1中的某处。因此，逻辑迭代地继续试图猜测差错是什么，再次继续猜测可能产生正确结果的值，每次进行一个修改。它采取SMB1(DE749BD6)的当前内容作为其起始值，起初每次修改一个比特并分析该结果。例如，如在下一文本框的后续行中所表示的，逻辑尝试DE749BD7的可能值(即，修改最低有效比特)，以确定这是否是差错的来源，反向转置的结果为58486668，这仍然与SMB2(121D2410)的内容不匹配，即限制存在13比特的差错。因此，该逻辑尝试第二个可能的值，这一次将修改第二最低有效比特，并且这导致反向转置结果78FDCD32，其中产生14个差错比特。然后逻辑继续迭代，修改第三最低有效比特(SMB1＝DE749BD2)，并且确定这将导致23个差错比特。如通过左侧文本框的第23行可以看到的，在某个时刻，系统尝试正确的SMB1值DE749AD6，并且这将产生与当前SMB2内容匹配的结果(即，结果为121D2410)，并且异或运算的结果为零(即，零差错比特)。因此，系统将SMB1 SMB2标识为与相关联数据块的正确比特相对应，并且将正确比特置换回原始签名(即，产生候选)并且然后，继续进行分析下一个数据块。

[0098] 关于该讨论，应当注意几点。首先，考虑到在该实施例中每个块中只有64个比特(|SMB1|＝32并且|SMB2|＝32)被分析，混叠的可能性增加(即，多于一个的正确解)。其次，为了最小化运行时间，检查点回滚逻辑可以被设计为当每个候选被标识时跳到下一块(即，针对AC1)。因此，如果当针对每个已标识的候选比较每个次级证明曲线(即，AC2至AC5)时，甚至存在单个不匹配，则不能找到正确的签名，并且然后逻辑必须返回以评估其他可能的签名候选(例如，如果这还未被完成，它必须尝试确定对SMB1和/或SMB2的其他按块修改是否会导致其他候选解)。这相当于“懒惰而乐观”的候选集合“搜索并提名”策略，该策略推迟引入新的候选以用于最终证明，直到在当前候选集合中的最坏情况的不匹配所要求为止，该策略在名义上可以执行得非常快，因为它只需要在漂移的例外情况下深入。注意，该策略是可选的，即，这不是所有实施例都需要的。

[0099] 对以上讨论的与图3A至图3E相关的操作进行回顾。表示存储的检查点的证明曲线数据可以被取回并且被用于修改当前硬件签名，直到它与对应于所存储的检查点的硬件签名匹配为止。所存储的证明数据本身不能被拦截并被用于标识硬件签名，但是它可以被所讨论的IC使用以标识或猜测正确的先前硬件签名状态。所描述的原理可以在板载IC上被执行，速度快，并且即使在(达某个量的)随机噪声(即，被有意引入的比特差错)被注入以屏蔽

来自攻击者的正确解的情况下，也能产生正确解(即，从而潜在地向量子计算攻击提供弹性)。所描述的过程允许迭代地或以其它方式回滚任何经测量的硬件签名和修改，直到经测量的硬件签名与原始根密钥(例如，易失性或非易失性密钥)匹配。注意，本领域技术人员将想到在这些描述的过程中的许多变型，给定具体应用或实现，这些变型可以提供效率。

[0100] 图4A-图4C提供了用于图示与证明曲线的使用相关联的一些原理的概念图。

[0101] 特别地，图4A示出了证明曲线403、汉明邻域405和证明曲线上的点407的图形401。如前所述，由上面讨论的证明曲线过程提供的加密类似于非多项式曲线加密过程。例如，在有限域F上，椭圆曲线加密过程的形式一般为$y^2=x^3+ax+b$，其中a和b是常数。然而，在证明曲线的情况下，证明曲线过程的曲线形式不是利用椭圆曲线由多项式的形式和诸如a和b的常数来定义的，而是通过将签名值的比特具体排列到SMB1并且通过使用在有限域中可以实现的SMB1的比特对SMB2进行的加密有效地定义。然而，在图4A的情况下，点407可以被认为是硬件签名的非排列比特，并且通过SMB1的不同可能排列和将它作为加密密钥的使用来有效地呈现出各种曲线排列。以上定义的证明曲线加密函数一般对应于c＝H(x,y)的形式，其中c被视为常数，并且其中加密是可逆的，使得$y=H^{-1}(x,c)$。一般来说，期望提供这样的过程，在该过程中，对汉明邻域中的某个点的近似正确的导出允许精确地导出由点407表示的正确密钥(即，硬件签名和相关联的x、y坐标)。

[0102] 图4B提供了另一图形403，这次示出了两个证明曲线403和413。在这种情况下，两个曲线403和413可以被类比于前面提到的曲线AC1至AC5中两个曲线；在所描绘的坐标空间中，如果使用相同的硬件签名生成两个曲线，则两个曲线应当相交于共同点407。换言之，应当只有一个唯一解满足两个曲线的约束(注意，除了前面提到的点407，还存在两个曲线具有其他交点的概率，如前所述的，但是这种可能性在统计上是极小的；并且对于围绕点407的紧邻汉明邻域，该可能性更小)。例如，参考前面对多个候选评估的讨论，如果存在针对一个曲线的多个候选解，诸如第一曲线403上的点417和第二曲线413上的不同点417'，则它们不共同地存在于所有曲线上，并且不表示用于形成两个曲线的唯一解，在这种情况下为点407。再次注意，从理论上讲，曲线可以具有多于一个交点，使用多于两个证明曲线有助于抑制这种可能性，即，三个、四个、五个或更多曲线都相交于同一点(尤其假定点被长的签名长度(例如，768比特或更多)解析)的可能性是如此微小的，以至于在实践中永远不会发生。如前所述，在一个实施例中，四个或更多曲线被用于解决这种可能性(考虑到至少一个曲线主要用于如上所述的分区，使用五个曲线)。事实上，因为更多这种曲线被用来解析唯一的交点，所以概率使得对于相应曲线允许更大的可接受的有意诱导的比特差错(例如，汉明距离差错，405/415')变得越来越可行，并且仍然准确地恢复正确的交点。不知道加密参数(例如，SMB1中的值)的攻击者将无法精确地标识任何曲线，并且在经加密的证明数据中使用少量有意诱导的随机比特差错意味着，即使证明数据(单独)本身并不暗示或要求正确的共同交点，也可以通过检查点恢复过程标识曲线的共同交点。也许另有说明，在所描述的检查点恢复过程中，其中SMB1的比特未存储在外部，攻击者不能标识任何曲线，并且外部存储的每个经加密的证明代码不提供关于它所表示或在由图表示的坐标空间中的任何具体的点的曲线的信息。如由图4B所示的，可以定义均相交于点407的许多证明曲线，即，作为上述签名定义过程的函数。使用相同的硬件签名(SMB1SMB2＝该签名比特的某个函数)定义的任何曲线都将有效地通过点407，并且该特性可以被用于提供鲁棒性并且精确地标记任何硬件签

名检查点和/或根密钥值。

[0103] 图4C图示了由参考附图标记421所指示的另一示图。在该图中，添加了由新点427引出的第三曲线423，新点427具有相关联的汉明邻域425。所描绘的点被用于表示相对于先前签名值的漂移(即，偏离点407的漂移由点427表示)。因为漂移导致不同的签名，所以在理论上应该与点407匹配的点427在所描绘的空间中的某个不同的点。这些点之间的差异被表示为漂移向量429。通过标识和追踪检查点之间的该向量以及通过标识和追踪多个这种向量，该过程有效地允许将后来的经点检查的证明曲线(它标记唯一的点，诸如点427)和迭代回滚链路，例如所描述的诸如从点427到点407，尽管是渐进的漂移。该回滚可以在链路的基础上被递归地重复，以追踪任何经测量的硬件签名返回至原始根密钥，尽管在合计漂移量的情况下。通过用多个证明曲线(例如，AC1至AC5如以上所例示的，以确保唯一解)标记诸如点427的新点，并且通过足够频繁的点检查以便提供用于隔离和解决每个时刻的边缘漂移的快速运行时间的解决方案，其可以提供有效的、快速的解决方案，以解决任何可以想象的漂移量，甚至潜在地多达并且包括所有根密钥比特的破坏。

[0104] 鉴于此，图5被用于讨论与检查点存储和生成以及关联技术相关联的一些选项501。如附图标记503所表示的，根据实施例和应用，期望集成电路(“IC”)制造商、BIOS、软件应用或用户中的至少一者应该能够有效地标识何时(when)(例如，在什么时候，或多频繁地)应该对新硬件签名测量进行点检查，以便提供快速运行时间的解决方案。如前所述的，在一些实施例中，通常期望足够频繁地生成检查点，使得固件(在硬件逻辑的辅助下操作)应该能够处理表示检查点之间漂移较小(例如，小于或等于大约3比特)的检查点，从而有助于提供微秒级的解决方案。注意，即使没有该目的，本文中所描述的技术将仍然可以运行，尽管可能需要更长运行时间的解决方案。如该讨论所表示的，根据实现、应用、快速透明处理的需求和漂移率，点检查可以在不同的时间被执行。例如，在一个应用中，可以由设计人员确定在每次上电时应该建立新的检查点，如附图标记505所表示的。在另一实施例中，由附图标记507表示，可以在调度的基础上执行新检查点的生成和相关联的数据(例如，证明曲线数据)的存储，其中调度是无法推算的。例如，按照附图标记511，针对IC上板载的逻辑，可以检测预定数目的事件的发生，例如，电源开/关的预定循环次数的通过，或特定的软件或硬件事件(例如，固件/软件升级、性能维护、由存储器磨损所指示的设备老化、系统使用的持续时间，等等)。在如附图标记513所表示的另一个实施例中，设计人员可以静态地或动态地编程用于新的检查点生成的设置时间表(例如，每年的1月1日、或在其他一些基础上，根据需要，并且根据需要这可以以编程的方式被改变)。按照附图标记515，在一个实施例中，特定IC设计的制造商可以测试或鉴定其产品，以检测(即，测量)预期的漂移率以及响应性地配置点检查(例如，给定以经验为主测量的漂移率，通过对新检查点的频率进行编程，使得在检查点之间发生不超过几个比特的漂移)；再次引用前面介绍的示例，特定的制造商确定在经测量的硬件签名中预期每年多达3比特漂移，可能提供固件(或硬件逻辑)，该固件(或硬件逻辑)根据该速率导致新检查点的生成，例如，使得每季度执行的点检查可以被预期遇到0至2比特漂移的最坏情况。注意，漂移行为的这种收集或统计特性不应向制造商显示任何特定的硬件签名值，即，理想情况下，这种统计特性仅针对给定的一般设计被执行，而针对给定的设计，芯片间的密钥详情被保密。如一般地由该过程块515所涵盖的，还可以被设置或以编程的方式定义其他详情，例如，在任意集合中的AC曲线的数目(AC曲线的集合

的数目被用于提供鲁棒性)、具体的加密算法(例如,SMB1/SMB2的大小)、等等。作为进一步的示例,购买IC以用于组装其系统的特定系统集成者可以确定只需要三个证明曲线(而不是五个),或者确定在给定预期的应用和期望的安全性的情况下,32比特块的按块处理提供比42比特块更快的运行时间恢复。作为进一步的示例,通过将"冻结"检查点数据集合的简洁校验和验证签名编写到在IC上的OTP存储器中(例如,诸如使用电子熔丝),制造商可以确定一旦IC被装运到第一客户就提交和锁定根密钥,以防止进一步重新生成。这种校验和的大小可以是小的,诸如128比特或64比特。校验和是不可更改的,并牢牢地被锁定在IC的内部,使得在尝试取回根密钥之前,可以针对为证明而取回的初始检查点数据组对其进行验证。备选地,制造商可以确定将这种锁定延迟到OEM,或者甚至将其延迟到最终客户。这些参数和其他参数都可以由IC制造商、系统集成者、OS或应用提供商或用户根据实现进行测量和可编程地定义。在其他备选的实现中,点检查可以被动态地进行或特定地进行,即,如由过程块509所表示的。举几个示例,如果在试图将恢复的硬件签名应用到外部操作数的解密中时发现差错,则处理器IC可以被配置为动态触发新检查点的生成(即,按照函数块517)。在另一实现中,新检查点(按照函数519)可以被软件操作系统、软件应用或用户命令所触发;在另一实现(521)中,可以在某些操作数发生改变的任何时候(例如,新的私密密钥被安装、或者PKI证书过期)调用新的检查点。自然地,这些各种技术可以以任何期望的组合被混合和匹配;例如,可以定义点检查,使得其发生的频率不低于每个季度以及任何时候遇到漂移差错的频率。本领域普通技术人员将想到实现中的许多示例和变型,并且在很大程度上将是考虑设计质量和安全性要求的实现决策。注意,在典型的应用中,这些参数可以被硬连线到设计中(例如,使得例如密钥恢复逻辑的逻辑被固有地设计为应用它们)。在其他实施例中,包括那些支持板载NVM的参数,这些参数可以被存储在内部(诸如寄存器中)、或者它们可以以其他方式被加密和/或片外存储。

[0105] 图6示出了系统601的另一实施例,这次示出了处理器IC 603和NVM IC 605;图6被用于提供基于上述根密钥(易失性或非易失性密钥)恢复过程的应用的介绍性示例。所描绘的IC 603和605都可选地被安装到共同板或模块(例如,CPU模块)或以其他方式构成同一系统的一部分。如果以电子板的形式实现,电子板将具有外部电气连接、或用于利用被用于接收和发射信号的接口643来输入/输出连接的引脚或焊盘(如附图标记644所表示的)。如前所述,这些引脚或焊盘可以使用并行或串行通信(例如,与ATA、SATA、USB、SPI、显示端口、PCIE和/或其他发布的标准和它们的不同版本兼容)连接到一个或多个外部导电路径(可选地被配置为分离的命令/地址总线和数据总线317)。所描绘的系统601还具有内部导电路径607,内部导电路径607用于处理器IC 603和存储器IC 605之间的通信,并且导电路径607可以是与这些标准兼容的并行或串行路径。在该系统是分布式系统(诸如由局域网或广域网(LAN或WAN)表示)的情况下,这些路径可以被组合以用于分组通信,并且可以包括无线链路。

[0106] 图6被用来叙述如下过程:恢复硬件根密钥,以及接着使用该硬件根密钥对用于安全的外部存储装置的私密密钥或秘密密钥进行加密,例如,尽管在一个实施例中,处理器IC使用较新的制造工艺技术被制造,并且可能无法在非易失性的基础上内部地保留私密密钥。在这方面,处理器IC可以是CPU、FPGA、微控制器、图形控制器、蜂窝电话处理器、多核设备、多处理器芯片或其他形式的处理器。如附图标记609和611所指示的,处理器或处理器核

使用测量电路装置测量或读取硬件签名。经测量的硬件签名609可能会受到漂移或差错的妨碍，并且为了减轻该漂移或差错，处理器取回外部存储的检查点数据613。这允许对经测量的硬件签名进行差错校正，如前面描述的直接回滚或迭代回滚，以恢复固定在初始设备配置上的原始根密钥615。通过非限制性的示例，硬件签名可以可选地通过测量PUF来获得，并且回滚可以可选地使用证明曲线过程(如刚刚描述的一个过程)以及其相关联的益处被执行。根密钥永远不会在处理器或处理器核的外部被共享，并且存储在处理器或处理器核外部的该数据有利地不允许根密钥的导出或与根密钥(诸如硬件签名)密切相关的任何事物。如前所示，在一个实施例中，检查点数据(或检查点数据的至少部分)可以通过IC制造商或系统集成者被远程地存储，使得在随后的附加配置时间，发布的检查点数据可以被取回(例如，经由WAN)，该检查点数据基于设备ID被用于安全地标识所制造的设备，并且被用于在首次使用或软件升级时配置设备。

[0107] 在图6的实施例的上下文中，应当假定作为初始配置的一部分，经销商或用户将针对密码操作配置处理器IC 603和/或系统601(或其一部分)，并且将安装秘密密钥(或公共密钥-私密密钥对中的私密密钥组件)以供IC 603上的处理器核或处理器使用。尽管处理器或处理器核的潜在的易失性性质，所描述的技术允许根密钥总是被恢复，并且然后这允许对秘密密钥加密以及外部存储经加密的秘密密钥。因此，一个或多个密钥可以被安全地存储在外部的NVM中，并且可以根据需要被取回、解密和管理。尽管IC 603上的处理器可以不具有内部的易失性存储装置，但是它可以外部地加密私密密钥或秘密密钥，并且当它被重新供电时，它可以恢复其易失性(或非易失性)的根密钥，使用该根密钥来解密外部存储的加密工具，并且接着使用RAM内部的那些加密工具。图6示出了至少两个这种密钥(617/619)和省略号618的存在，省略号618表示可以存在任何期望的数目的这种密钥。这些密钥中的每个密钥都可以使用根密钥进行加密，并且然后被外部地存储，并且可以根据需要(例如，当向系统或处理器IC供电时)被取回和解密。注意，在可选的实施例中，将描述快速加密和解密引擎，该引擎仅使用硬件提供这些密钥(以及任何期望的数据，例如，被选择性加密的数据)的几乎瞬时的加密和解密。在发生基于系统的密码通信的字典攻击的情况下(例如，基于私密密钥1至N的使用，对应于附图标记617至619)，这样不显示在该应用中的根密钥，因为根密钥仅仅被用于加密非易失性存储装置的私密密钥，并且因为(多个)次级密钥(例如，密钥617至619)被用于外部通信和潜在的更广泛的加密(例如，存储在存储器中的可能的大数据集合)。换言之，在发生破解的情况下，被破解的私密密钥或秘密密钥可以被撤销或被丢弃，并且新的密钥和/或凭证被安装在系统601上，并且然后通过基于根密钥的加密再次受到保护。为此，在上电或稍后的动态时间，处理器或处理器核从NVM IC 605取回它需要的任何密钥，并且然后使用解密电路装置625对这些密钥进行解密。一旦解密，获得的密码密钥(627)可以通过以下方式内部地保留在易失性存储器中(例如SRAM、628)：这些密钥不能以未经加密的形式在处理器或处理器核的外部被共享或使用。此外，为了阻止对受根密钥保护的高值秘密资产的字典攻击，可以采用不确定的加密技术，从而在密文大小中分配小的开销空间来容纳嵌入的随机填充比特。例如，如果假定256比特(32字节)大小的秘密密钥被用于外部存储装置的根密钥进行加密，则IC可以分配288比特(36字节)的扩大的密文大小，从而允许额外的32比特(4字节)空间用于填充开销。在加密期间，32比特的随机填充(例如，来自片上随机数生成器)被串接或预先连接到256比特的明文有效载荷，以组成

288比特的操作数，然后使用根密钥作为加密密钥对其进行加密，产生288比特的密文(FPE)输出。这种嵌入式填充方案允许不确定性加密来提供结果，使得即使针对相同的有效载荷值，也很少在新的加密中重复输出密文(如果有的话)，从而使字典攻击变得不切实际。

[0108] 如附图标记621所示，用于加密/解密的其他数据和/或信息也可以被外部地存储在NVM IC 605中。例如，在一个实施例中，经加密的屏蔽密钥或混淆密钥621也被存储在该存储器中。该可选的屏蔽密钥被用于在加密之前或之后翻转一个或多个私密密钥603/605的比特，并且相反地再次在解密之前或之后翻转从存储器中取回的数据(例如，私密密钥)。例如，在一个实施例中，用于比特翻转和/或转置的密钥和/或屏蔽密钥从经测量的硬件签名或从单独的PUF阵列中被导出。作为示例，被用于256比特随机数生成(即，以及如前所述的相关联的密钥扩展、哈希和处理)的单独的PUF可以被用于导出作为秘密密钥应用的第一数和向量，该向量确定该秘密密钥的(或从该秘密密钥导出的)比特是否要被翻转。恢复后的根密钥可以被应用于对用于外部存储装置的私密密钥或秘密密钥进行加密，而次级PUF(或随机数生成)接着提供独立密钥，该独立密钥可以被用于进一步伪装值并且提供另一层加密。这个的示例，用于比特重新排序和/或随机比特翻转的一个或多个转置向量可以使用与上述相同的基本过程来被生成，即，通过获得256比特的随机数、将其与自身串接并且使用密钥扩展技术和比特扩散(例如，经由CRC划分、AES或SHA-256处理)通过应用模运算来获得之前描述的整数；在向量被用于比特翻转的情况下，代替对结果的整数值进行排序，而是这些值可以被转换为偶数或奇数(即，逻辑“1”或逻辑“0”)，并且被用来确定经加密或未经加密的私密密钥/秘密密钥的对应符号位置是否应该被翻转。另外，根密钥可以使用如美国专利号9635011中描述的FPE过程来加密私密密钥，该过程参照前面讨论的与SMB2内容的加密(例如，可选地结合上面提到的嵌入式随机填充方案)相关的转置、选择性比特翻转和非线性置换。当结合可选地还使用随机数生成的能力来生成在处理器或处理器核上板载的秘密密钥时，即针对临时或长期使用，这些工具提供了现成的手段来支持各种密码过程，包括根据需要的新密钥的生成和对那些密钥的安全存储。在一个实施例中，该屏蔽密钥或混淆密钥还可以被应用于根密钥本身。例如，附图标记621表示可以将屏蔽密钥/混淆密钥应用到原始根密钥，以便获得与原始根密钥无关的秘密密钥/私密密钥(除作为由屏蔽/混淆密钥提供的加密函数之外)。因此，当根密钥被恢复时，其可以被用于取回和解密屏蔽/混淆密钥，并且从而恢复从根密钥导出的私密密钥；由于导出可以包括(例如，如下面所述)比特置换、翻转和非线性置换，这种过程可以使私密密钥与根密钥完全不可识别。确信由这种AC组成的检查点数据集合可以使芯片恢复硬件根密钥，同时，原始根密钥的经加密的版本(即，如点检查的)以及根密钥的混淆数据(或如经加密的，经混淆的根密钥)可以与检查点数据集合存储一起被存储。然后，在随后恢复经点检查的根密钥的成功尝试中，简单地通过取回存储的经加密的原始秘密并且利用取回的检查点根密钥对其进行解密，就可以有效地恢复原始根密钥和/或其经混淆的版本。所有这些都可以由图6中的屏蔽/混淆密钥621的存在来表示，如根据需要使用多个这种密钥。

[0109] 如附图标记623所表示的，其他信息(除屏蔽/混淆密钥或比特翻转密钥之外)可以被存储在NVM IC 605中，以辅助处理器IC 603进行密码操作。作为这个的非限制性的示例，用户可以将其他凭证(例如文件、网站、应用等的密码)以加密的形式存储在NVM芯片605中，其中处理器IC 603根据需要自动取回、解密和应用经解密的凭证。上面所讨论的排列、比特

翻转和/或非线性置换原理还可以被可选地用于保护该数据，并且再次，根密钥仅被用于(在该示例中)本地或系统内加密，而经解密的数据(在该示例中与根密钥不相关)被用于外部密码和其他安全交互。

[0110] 处理器可以使用SRAM 630中的经解密的私密密钥作为主加密解密电路629的输入，以允许对内部有效负载631(即，将被外部地传输的数据，例如，被存储在NVM IC 605中或经由接口643和WAN 645外部地传输的数据)进行加密，或备选地，以允许经由NVM IC 605或接口643/WAN 645接收经加密的数据供内部使用/消耗。为了提供这方面的一些非限制性示例，在一个实施例中，处理器IC 603是FPGA，它从NVM芯片605接收经加密的编程比特流643，编程比特流643被解密并且被用于配置专有操作的FPGA。处理器或处理器核还可以通过使用解密的秘密密钥对这种信息进行加密来以加密的形式外部地存储其操作数(例如，操作参数、状态、维护信息和其他操作数)，以这种方式，即使将易失性处理器IC 603断电，这些操作数仍然以受保护免于攻击者拦截和解密的方式继续存留在外部NVM中。其他数据639也可以以这种方式被存储。例如，可以以安全、加密的形式维护NVM中的所有系统文件，其中处理器IC使用取回的、经解密的私密密钥/秘密密钥来保护这些文件。在另一应用中，处理器IC 603可以使用本地生成的有效载荷(例如，由用户输入到系统来获得的，例如，经由可选的用户接口646获得的)使用外部网络(WAN或LAN 645)参与事务和其他通信，例如，使用处理块641来加密信用卡号、执行安全事务等。

[0111] 再次，本领域普通技术人员将想到各种修改、置换和选项，并且针对图6的实施例，前面描述的具体特征(包括PUF和/或证明曲线的使用)将被看作是可选的。

[0112] 图7被用于提供关于硬件随机数生成的附加细节。更特别地，图7示出了集成电路(“IC”)701，它具有亚稳态电路的阵列705。如上所述，随机数生成可以被用于创建用于如上所述的转置生成和比特翻转向量的秘密加密密钥，以及用于生成被用于创建非线性置换表和/或用于处理器或处理器核703的其他目的的信息。如前所指示的，这种阵列可以起到双重作用，可选地被用于硬件签名生成和随机数生成；然而，在一些实施例中，阵列705可以可选地专用于随机数生成。如图所示，所描绘的阵列提供了物理不可克隆函数，这次具有256个单元；亚稳态单元的设计可选地可以基于图2A中示出的结构。如参考附图标记707和709所表示的，在该实施例中，排序和行控制电路装置的使用是可选的；换言之，在一个实施例中，所有的256个亚稳态单元使用共同的激励信号同时被感测，这些单元的状态被锁存电路装置711感测和保持。类似地，针对该实施例，硬件签名计算电路装置713也是可选的，即，在一种可选的情况下，此处还使用了前面描述的基于帧的重复测量和直方图的使用；但是，在另一种情况下，只使用一次测量，产生逻辑“1”和逻辑“0”的256比特字符串。备选地，如前所述，可以使用256字的字符串，其中每个字的内容从“计数”电路中获得，在0至120的范围之间。然后将该测量用作随机数种子。因此，每当处理器或其子组件需要新种子时，就会向阵列705提供测量信号，阵列705向电路装置717提供新鲜种子。

[0113] 注意，如前所述，尽管阵列705的每个单元被设计得尽可能相同，但由于工艺角，通常会有许多单元重复地生成相同的值，并且因此阵列705的多个值一般是关联的。因为在输出值中的这种关联，从该阵列中产生的256比特值(或256字值)不被直接用作随机数，但它与其他信息714和/或721(如将在下面描述的)串接，并且然后该串接信息被提供给提供比特扩散的电路装置717，例如，使用CRC32划分、或AES或SHA256加密处理。尽管只有亚稳态单

元的子集产生真正不可预测的状态，但这些过程将提供足够多的熵，使得从电路装置717产生的将是随机数。该输出被提供给寄存器719，其中它被保持以用于由相关的处理器或处理器核读取。注意，在一个实施例中，电路装置717的输出可选地基于其自身被反馈到求和结点，其中它被串接或被用于确定性地修改来自阵列705的下一个种子。此处的示例将再次是有帮助的。在某个时刻，例如在上电时或在动态的或事件驱动的基础上(例如，“每小时”)，IC 701向阵列705发出测量信号以产生新鲜种子。256比特(或256字)的输出值被提供给求和结点715，在该求和结点715之前连接某个nonce信息714(例如，从随机池中起草的，与明文值“随机密钥1、……随机密钥9”等同的ASCII，时间戳或由软件供应商或系统设计人员提供的一些事物)，并且与经由路径221到达的反馈信息结合(例如，串接或异或处理)。然后，电路装置717应用加密过程，该加密过程将该输入中的熵在电路装置的输出的所有256比特上传播。然后，电路装置717将该输出721反馈给求和结点，使得它的输入不断地改变，并且因此，使得由电路装置输出的256比特以高熵不断地改变。还注意电路装置也被时钟控制(即，由时钟输入φ的描绘所指示的)，即，随机的数字输出是不断产生并不断改变。因此，来自处理器或处理器核或其他电路装置的读取信号可以被用于在读取信号被发出的任何时间触发新随机数的输出。注意，寄存器719也被时钟控制，使得它不断地被加载新鲜数；一旦种子被生成，在本实施例中所描绘的电路装置非常快，使得不同的随机数基于需求不断地可用。

[0114] 如前所述，生成的随机数可以被用于各种目的。上述讨论的并且由附图标记725所表示的一个目的是随机转置向量的生成。换言之，如前所述，随机数可以被用于生成整数值(例如，如前所述的，768元组)，并且然后电路装置可以对每个整数值排序，以导出如前面描述的转置或排序向量。在图7的上下文中，框725表示生成这种向量的在IC 701上板载的(或由非暂态存储介质提供的)硬件和/或指令逻辑。第二，前面已经示出的，一些实施例生成并且使用比特翻转向量，例如，768比特的值可以被用于翻转硬件签名的对应比特。框727表示逻辑，再次地硬件和/或指令逻辑，其导致这种情况发生，例如，通过接收生成的768值的转置向量、将每个值转变为奇数或偶数(通过简单的模数2、或通过模数7或6运算(例如，接着是“>3”或“>＝3”分支测试，例如)，并且然后执行比特翻转。第三，生成的随机数还可以被用于非线性置换表(“NLS表”)的生成。换言之，前面已经描述的，为了加密SMB2(参见图3A至图3E，如上所述)的内容，加密过程可以在可逆的基础上生成这种表。生成的转置向量或另一随机数可以被用于填充这种表。举出非限制性的示例来说明这是如何做到的，转置向量可以被生成并且被用于分配二进制数的排序；在以12比特为示例的情况下，表可以被根据12比特索引的4096个不同的二进制值填充，其中每个表条目依赖于4096值的转置向量进行填充(例如，转置向量将二进制值“9”映射到位置“143”，并且因此使得输入比特“0000 0001 0001”(即，“9”)经由该表被映射到输出“00001000 1111”(143)，其中通过内容寻址输出或备选地通过分配另一表实例以存储反向映射来实现反向转换(使得，例如，输入比特“00001000 1111”(143)被反向映射到输出“0000 0001 0001”(即，9))。针对转置向量生成逻辑、比特翻转逻辑和NLS表生成逻辑中的每一个，生成的值可以由硬连接、存储或编程的参数733来指示。例如，在一个实施例中，可以期望产生12比特的NLS表，而在另一实施例中，可以使用特定的有限域运算来应用某种模运算。本领域普通技术人员将想到许多示例，并且根据设计，生成的随机数可以被用于各种方式。在一个实施例中，逻辑725、727和729中的

每个逻辑使用共同转置向量来播种加密/解密(例如,用于证明数据),但这不是所有实施例都需要的。如附图标记731所表示的,随机数还可以被应用于其他目的。

[0115] 较早注意到的,在一个可选的实施例中,经恢复的根密钥不被直接用于IC的密码操作(除了私密密钥/秘密密钥的安全存储)。相反,在这种实施例中,一个或多个秘密或私密密钥可以使用根密钥被加密,经加密的(多个)密钥从NVM被取回和被解密(即,在该根密钥被恢复以后并且接着被用于恢复这些私密/秘密密钥)。进一步地,在一个实施例中,这种其他的私密密钥/秘密密钥可以被片上生成,并且在其他实施例中,它们可以被外部地提供(例如,经由Diffie-Hellman密钥交换过程)。依赖于这些独立过程的架构的动机可以是,所讨论的IC没有内部的非易失性存储器,并且因此,当断电时,所有内部内容丢失。因此,板载存储的数据可能需要被移动至片外,这可能会是有问题的,其中安全是个问题。尽管相信许多设计将具有一些用于内部处理的受限板载易失性存储,但是事实上,还可以期望总是为经加密的RAM内容的外部存储提供资源。一个充分示例是许多现代处理器使用的以高速缓存为中心的存储器管理系统("MMS"),由此小容量的芯片驻留的L1高速缓存可以通过将随时处于活跃状态的存储器部分及时地从外部DRAM存储装置或甚至硬盘驱动器调换进来或调换出去,来为多万亿字节大小的"虚拟"存储器空间提供服务。图8A和图8B被用于描述辅助这些功能的实施例。换言之,图8A描述了其中内容可以在NVM中被加密并被片外存储的实施例,以及图8B描述了其中内容可以在RAM中被加密并被片外存储的实施例。在一个实施例中,图8A示出了用于实现阴影非易失性存储器的策略,并且图8B示出了用于阴影易失性存储器的架构。图8A和图8B所描绘的机制还可选地使能在阴影非易失性存储器和阴影易失性存储器之间的受控双向内容交换。

[0116] 图8A示出了IC 801,其选择性地加密用于外部存储在外部非易失性存储器(NVM)809中的数据。NVM 809可以驻留在共同板或系统中的第二集成电路中,或者它可以远程地位于例如位于在LAN或WAN的另一端。一般来说,有线的(或"导电的")链路(例如由附图标记815、817和818所指示的)被用于至少部分地与NVM 809通信,但是在一些实施例中,连接这些元件的路径还可以包括无线链路组件。如以前的,连接链路可以兼容于诸如ATA、SATA、USB、SPI、显示端口、PCIE和/或其他已发布的标准及其各种版本(例如SATA、USB、PCIE的版本2.0、3.0等)的通信标准。

[0117] 假定IC 801具有至少一个处理器或处理器核802,并且该处理器或处理器核将从事要求以NVM的某种形式存储参数、设置或操作的操作;这由图8A中的函数块803表示。在常规设备中,这种NVM可以在板载芯片上,但是在图8A的实施例的情况下,假定该存储器是片外的(例如,如上所述,IC 801可以只具有易失性存储)。因此,期望具有本地存储器(影子NVRAM)805,其将处理器或处理器核视为非易失性存储器,尽管它实际上是易失性存储器。当断电或以其他方式期望或需要收回(或更新以向NVM提交以在整个电源周期内保持数据)数据时,写入信号811被提供给影子NVRAM以使其将数据从内部本地存储器写入到位于片外的NVM809。为了协助该任务,处理器或处理器核将数据引导至加密/解密电路813,在该实施例中,该加密/解密电路813被视为可选地在处理器或处理器核的外部(但是在IC 801上)。在下面描述的一个实施例中,在IC 801上可以存在许多处理器核,其中这些处理器核时间复用地使用加密/解密电路813。在一种实现中,加密/解密电路813是快速电路,其被实现为提供纳秒级加密/解密响应的非时钟控制的逻辑门和查找表的集合;下面将结合图9A至图

9B来描述这种电路的示例。

[0118] 然而,注意,针对处理器或处理器核802的所有应用和需求可能可以不都需要存储经加密的数据。换言之,在一些实施例中,可以存在处理器或处理器核期望加密的一些数据(例如,如上面所讨论的私密密钥/秘密密钥和关于密码和其他凭证的其他数据,例如,信用卡号、密码等)(这由数据流动路径815表示)以及未被加密(或进一步加密)的其他数据(由流动路径817和818表示)。无需(进一步)加密的数据示例包括:证明数据和转置向量T1至T5(由流动路径818表示,即,其已使用特定处理器被加密,或以明文形式被存储);以及被认为不敏感或不保密的数据(即,由流动路径817表示)或本身已被通信协议加密的用于外部传输的数据。为了满足这些变化的需求,加密/解密电路813可以实现复用能力,其中它使用加密电路装置819执行可逆加密函数,或者将数据直接或经由旁路电路装置820传递到片外的NVM 809或任何可能的目的地。

[0119] 如前所述的,加密性能可以可选地基于经恢复的易失性(或非易失性)根密钥825。在IC 801上板载的电路装置测量PUF阵列823,并且使用用于执行检查点(“CP”)的生成来标记硬件签名中的渐进漂移的电路装置,并使用用于将经测量的硬件签名还原到根密钥825的状态的回滚电路装置,来处理经测量的硬件签名。如前所述,随机数生成器(“RNG”)827可以被用于辅助这些操作,并且生成转置向量、比特翻转向量和非线性置换表。这种RNG电路可以可选地基于上面讨论的与图7相关的设计,并且,如前所述,RNG还可以基于需要提供随机数,该随机数将被用作秘密/私钥密钥828(例如,用于加密出站数据/解密入站数据的目的、用于待共享对称会话密钥或非对称密钥对的私密密钥部分的基于Diffie-Hellman交换的目的、或用于其他目的)。如前所述,不管生成的随机数是否被用作秘密密钥,都可以期望加密该密钥以用于外部存储,例如,如果电源被移除,则可以需要该秘密密钥来恢复NVM存储器内容。因此,为了该目的,根密钥825被选择性地用作加密密钥,并且因此被视为加密/解密电路813的一个输入。该根密钥有效地被用于加密秘密密钥,其被选择性地加载到本地存储器805中;经解密的秘密密钥然后可以被用作其他数据的加密/解密密钥,并且为了该目的可以被类似地用作加密/解密电路813的输入。如下面将进一步描述的,存储在NVM中的数据可以具有添加的明文比特或少量信息,以指示特定的数据是以明文的形式被存储还是以加密的形式被存储。因此,当加密数据从NVM 809被取回时(例如,经由路径815),它可以被提供给加密电路装置819,并且被转换为经解密的数据829,然后该经解密的数据829被提供给本地存储器805。相反地,当未经加密的数据从NVM 809中被取回时(例如,经由路径817),它可以被路由到旁路电路装置820中,并且被直接存储在本地存储器805中(即,由路径831所表示的)。根据需要,如路径818所示,类似的、附加的比特可以可选地被用于区分证明数据,使得它还可以被直接提供给旁路电路装置820,在这种情况下,其被直接提供给CP生成和回滚逻辑821。

[0120] 关于图8A,应当注意几点。首先,虽然路径815、817和818由单独的线表示,但是在一些实施例中,它们可以被实现为一个共同连接,例如作为单个通信链路或总线。第二,尽管图中单独地示出了用于检查点(CP)生成和回滚的逻辑元件,但是在某些情况下,这些元件可以与处理器或处理器核802集成(例如,指令逻辑可以将所有这种参数(包括经测量签名、证明值、和其他操作数、数据或参数)加载到缓冲器或存储器805并且使用通用电路装置对这些值进行操作。最后,尽管图9A描绘了IC,其中给定的处理器核具有其自己专用电路元

件,例如,用于根密钥生成和点检查、用于本地存储器和用于其他功能的硬件和/或指令逻辑,但是在其他实施例中,这些电路/功能可以横跨处理器或处理器核被共享,或者被IC上存在的这些子集共享,或者由所有板载IC上的处理器或处理器核共享。如参考附图标记833所示,这些功能中的一些或全部可以被用于支持密码操作,并且为此目的被馈送给CRC32划分、SHA256或AES加密电路装置;这是由卷积块834所表示的,或由附图标记835表示在处理器/处理器核外。

[0121] 图8B示出了另一框图,在该情况下,示出了IC 851,其选择性地对外部存储在外部易失性存储器857中的数据加密。通过非限制性的示例,这种易失性存储器可选地可以是动态随机存取存储器(“DRAM”)芯片或模块。可以有许多原因用于对外部DRAM存储装置的数据进行加密,通过非限制性的示例,包括,IC 851的易失性存储器不足和/或有必要将数据从本地影子RAM 855收回到IC 851的通用高速缓存或外部的DRAM存储装置。再次,假定处理器或处理器核使用存储器(这次是易失性存储器)来存储各种参数、设置或操作数,如所有这些由附图标记853表示的。在由图8B所表示的架构中,为了相同或相似的效果和目的,还使用许多与上述元件相同的元件;这种元件通常用与图8A相同的附图标记表示。例如,还存在加密/解密电路813,其具有加密和旁路电路装置819和820;再一次,该电路装置可以提供对证明曲线恢复数据832的旁路,以用来导出恢复的根密钥825。

[0122] 然而,图8B也表示了一些差异。首先注意的是,通常没有必要对共享秘密密钥进行加密并且将共享秘密密钥存储在外部的RAM中,即,由于存储在RAM中的数据是易失性的,因此在发生断电的情况下,预期相关的处理器或处理器核852将重新生成相关的参数、设定和操作数。第二,注意附图中还描绘了高速缓存和存储器管理单元(“MMU”)电路装置859。换言之,在所描绘的实施例中,该电路装置实现可选的中间高速缓存,其可以由多个处理器和处理器核或多个虚拟机或运行过程共享,并且可以期望一个处理器/处理器核或虚拟机/过程无法访问另一处理器/处理器核或虚拟机/过程的经加密的操作数,为此效果,电路装置859可以通过使用常规高速缓存和高速缓存收回协议来以与在片外使用相同的格式(换言之,以加密的形式)提供数据的高速缓存存储。如下面将讨论的,针对给定的处理器或处理器核,如果并且当数据被移动至影子RAM 855时,在加密/解密电路813中的电路装置可以针对相关的锁定和权限提供访问控制,例如,只有正确的(所有者)处理器/处理器核和/或虚拟机/过程被提供对其解密数据的访问。电路装置859还执行常规存储器管理功能,包括虚拟到物理地址的转换、高速缓存管理(即,一致性控制和收回过程)、以及在由影子RAM 855使用的那些、由高速缓存/MMU电路装置859使用的那些和在存储器857中片外使用的那些之间的页和记录大小的转变。由信号861所表示的,从影子RAM 855收回数据的决策通常来自处理器/处理器核852或高速缓存/MMU电路装置,当指示影子RAM 855中存在的空间不足并且某些比特数据必须被收回以为所需要的操作数腾出空间的状况发生时,该决策被触发。与前面的讨论一致,这种被收回的数据选择性地被加密(即,根据数据是否敏感和是否需要加密,或数据源/目的地的数据不能加密/解密的遗留因素),并且然后这种数据被存储在高速缓存/MMU电路装置859中或是它被完全地片外收回并且被存储在外部RAM 857中。再次与上面的情况一样,针对要被加密的数据,可以由例如从随机数生成器(827)或外部或其他来源获得的一个或多个私密/秘密密钥执行。在可选的实施例中,根密钥本身可以被用于某种加密,尽管如前所述,在许多实施例中,根密钥仅被用于对外部存储的经加密的密码密钥和ID

的恢复。

[0123] 对图8A和图8B所呈现的结构进行回顾，芯片可能缺乏非易失性存储器(或足够的易失性存储器)，例如，这种芯片可能是根据与嵌入式非易失性存储器技术要求不兼容的较新的制造工艺技术制造出的。易失性(或非易失性)根密钥使用片上电路装置被恢复，从而促进对解密参数的恢复。作为其非限制性示例，如果密码密钥被存储在板上，当断电时其也可能会丢失，其可以使用根密钥进行加密，并且以安全的方式被存储在外部以免于受到攻击者的拦截。由图8A和图8B所呈现的结构允许以类似的安全方式将几乎所有片外数据存储在易失性或非易失性存储器中，其可以根据易失性(或非易失性)根密钥恢复再次直接或间接地被恢复。在一些实施例中，电路装置可选地可以被选择设计为容纳多个处理器或处理器核，使得数据甚至相对于在同一IC上操作的其他处理器或处理器核被加密。本领域普通技术人员将想到所描述的原理的其他特征、益处、应用以及扩展。

[0124] 先前提到的，一个实施例使用快速电路装置来提供对所选择数据的加密/解密。图9A和图9B图示了这种快速电路装置如何被设计/实现的一个实施例。

[0125] 更特别地，图9A示出了参考使用附图标记901的这种电路的一种可选设计的说明。该电路基于一个或多个虚拟机、处理器或处理器核903。这些各种设备产生数据907，其伴随着相应的命令和地址信息909。后者信息通常包括对拥有的虚拟机、处理器或处理器核的标识，由图9A中的附图标记910和910’表示。通常地，管理程序分配线程ID(或虚拟机ID)，该线程ID被用于链路与在虚拟机(“VM”)上运行的过程相关的数据和特定机器。在一些实施例中，该信息被用于防止第二VM访问属于第一VM的信息。为此，所描绘的电路901包括电路装置913，该电路装置913选择性地允许或阻止读取数据到达不“拥有”该数据的处理器或机器，如通过利用读取数据取回的VMID与请求机器的VMID之间的比较来揭示的。在一些实施例中，该过程还与写入命令关联使用，例如，只有如果与写入对应的VMID对应于将被重写的信息的VMID，则写入就可以通过读取-修改-写入操作来实现，其中写入在电路装置913中完成。为此，当命令从第一IC发出时，电路901将这些命令存储在队列911中；当从存储器中取回相关联的数据(例如，读取数据或将通过写入修改的读取数据)时，决策块912针对取回数据的VMID，对排队的正在进行事务的VMID进行检查，以确保在操作继续之前匹配。在一些实施例中，该特征/检查针对所有数据(无论是否被加密)被执行，并且在其他实施例中，该功能/检查仅针对数据的子集(诸如仅针对经加密的数据)执行。可选地，针对流线型的写入保护的操作，假如操作系统已经具有覆写保护机制(诸如由分段故障事件提供)，写入VM/过程的VMID可以被简单地“填充”到明文操作数，以在实际写出之前被加密为密文，而无需检查特定的VMID是否可以对虚拟存储器空间中的RAM地址正确地执行这种写入操作。

[0126] 如前所述，并非所有数据都需要被加密，并且给定的VM/处理器/处理器核可以存储经加密的数据(例如，敏感数据，诸如信用卡信息)和未经加密的数据两者。在所描绘的实施例中，因此所有数据都伴随有明文ID 905，该明文ID 905指示是否要对数据进行加密。在某些情况下，单个比特可以被用于该目的，例如，逻辑“1”意味着数据将被加密，并且逻辑“0”意味着数据将不被加密。在典型的实施例中，该明文ID以明文格式被存储，即，使得当任何给定量的数据可以被取回时，快速加密/解密电路装置917就可以立即识别数据是否需要解密。如前所述，并且在图中由附图标记915表示，旁路电路装置(即，被标记为“旁路mux”)拦截读取和写入数据两者，并且在需要加密/解密时调用加密处理硬件917。因此，如图所

示，当从存储器中用明文ID值“1”读取经加密的数据907’时，旁路mux电路装置915将经加密的数据907’引导到加密处理硬件917，加密处理硬件917对数据(包括存储的操作数和VMID)进行解密。如果读取数据的VMID与请求者的VMID不对应，则操作将停止(即，数据可以被丢弃)。如果VMID确实与请求者对应，则请求的操作可以被允许。在一个实施例中，为了防止选择明文攻击(“CPA”)的潜在易损性，该电路可以可选地被设计为防止将明文数据改变为加密形式的过程(或反之亦然)。换言之，可以将电路设计为防止模式切换，除非自从最后一次入站负载发生后有效载荷内容已经被修改。

[0127] 图9B提供了关于图9A的加密处理硬件917的一个实施例的附加细节。更具体地，图9B示出了包括如下的部件：第一组查找表931-1、931-2、931-3和931-4；第二组查找表932-1、932-2、932-3和932-4；非线性置换表生成器(“NLSubs.Tbl.Gen”)933；以及第一组和第二组异或(“XOR”)门943和944。非线性置换表生成器可选地使用固件来辅助非线性置换表的生成，但是在所描绘的实施例中，附图中的所有其他元件都被实例化为非时钟控制的硬件，所描绘的该结构因此可选地为加密和解密功能提供纳秒响应。非线性置换表生成器接收主密钥935、辅助密钥936和转置向量937中的一者或多者；在一些实施例中，所有这三个参数都被用于构建一个或多个非线性置换表。注意，与用于恢复检查点的生成(即，用于证明数据的解密)的非线性置换表的生成不同，这通常基于外部存储的明文信息(例如，一个或多个转置向量)，因为在易失性(或非易失性)根密钥被恢复之前，对外部存储数据的解密不能发生。相比之下，针对图9B所描绘的电路，假定这种电路装置形成用于对文本、参数、秘密密钥和其他信息进行加密的加密/解密电路的一部分(例如，作为来自图8A中加密/解密电路813的一部分)；因此，根密钥可能已经被解密，并且用于构建非线性置换表的信息可以可选地基于独立密钥、RNG生成的密钥或其他信息。无论输入935、936和/或937是否基于根密钥还是其他方式，非线性置换表生成器都使用相关的输入值来建立在输入比特集合和输出比特集合之间的非线性映射，并且它单独地填充每个表(即，如箭头934所指示的)。举非常简单的示例，两比特输入(00、01、10和11)可以随机地被映射到这些比特的排列(例如，分别映射到10、11、00和01)；例如，当输入为“00”时，在该简化示例中的置换表输出比特“10”，并且相反地，当从外部读取值“10”时，该表使用内容可寻址映射，或使用反向表输出倒逆值“00”。在所描绘的实施例中，非线性置换表生成器根据输入信息935、936和/或937填充许多表，即，931-1、931-2、931-3、931-4、932-1、932-2、932-2、932-4等；该表的结构被有利地设计成确定性的，即，使得基于对相同的输入935、936和937的重新应用生成相同的表。以这种方式，即使断电，在恢复操作后，电路装置917重新构建相同的置换表，并且能够使用适当的解密参数对存储在外部NVM中的任何经加密的信息进行完全且快速的解码。如果电路被用于保护易失性RAM，那么可以假设在加密和解密循环期间内随时间的流逝未断电。再次注意，在经加密的数据表示影子RAM的内容的情况下，不需要该确定和对加密参数的恢复，例如，在影子RAM的情况下，随机密钥或其他随机信息(例如，nonce)可以被用于生成非线性置换表。

[0128] 所描绘的电路装置917接收明文ID值905(来自图9A)，在该情况下，该明文ID值被假定为逻辑“1”(即，因为假设伴随数据907被路由到加密电路装置中)。伴随数据907通常以并行的形式被提供，并且包括相关的VMID 910(或者相反，如果不使用VMID，则是逻辑“0”或其他随机填充)。再次注意，随机填充可以可选地被使用以提供不确定性加密来阻止字典攻

击，并且VMID可以被部署以通过在VM之间间的交叉读取来阻挡存储器的泄漏。在所描绘的图的情况下，假设128比特数据有效载荷以及7比特的串接(VMID和/或随机填充比特)将被接收，虽然其他任何数目的比特或相对数目的比特可以被接收(例如，144比特，具有132比特的数据和12比特的VMID/填充，等等)。可以看出非线性置换表931-1、931-2、931-3、931-4、932-1、932-2、932-3、932-4中的每个非线性置换表都以一种形式或另一种形式接收这些比特，并且这些表中的每一个都将被填充，以根据该特定表如何被填充，来用成对的比特替换集合代替一些输入比特。在一个示例实现中，针对设计的两层中的每一层，存在三个提供12比特到12比特映射的非线性置换表(即，n＝12)和九个提供11比特到11比特的映射的非线性置换表，总共135比特。由省略号939所表示的，任意数目的表都可以被用作与设计相关，并且由每个表提供的映射可以是相同或不同的，并且可以针对任意数目的比特而被执行为与设计相关，即，这些参数表示实现选择。例如，表931-1接收组合的VMID和/或填充/数据的135比特中的最高12比特，并且生成12比特的替换集合。这些比特经由个体比特线941被输出到表932-1至932-4中相应的一个表(注意，在该层932中将存在12个表，并且因此，这些表中的每个表从非线性置换表931-1接收一比特的输入)。第二层的每个表(见附图的底部)类似地用比特的替换集合来代替其输入，并且将这些比特提供为n比特或n-1比特，这些比特共同形成了135比特的经加密的数据(即数据和对应的VMID/填充都以全卷积方式一起加密)。再次，因为在该设计中没有时钟控制元件，所以加密速度非常快，达到纳秒级，主要由存储器查找延迟的两倍所确定。生成的135比特值与明文ID比特与向存储器提供的地址信息(该信息表示主机的逻辑地址，在图9B中没有示出，并且被直接传递给存储器)相关联地一起被存储在如关于图8A和8B所指示的外部NVM或DRAM中。当期望读取数据时，执行反向变换，即，最高的经加密的比特n(例如，12)被传递给第一非线性置换表932-1(在图中底部描绘的第二层932)，并且从该表读取对应比特(即，再次，使用内容查找来触发地址的输出，或者相反地，使用反向表的单独实例)。然后，从表发出的每个比特线被传递到附图顶部附近的第一层的表931。例如，表932-2描绘了个体比特线942，这些比特线被看作指向第一层的不同表。类似的内容可寻址或反向查找置换操作在那里被执行，产生原始VMID的恢复，该原始VMID与所讨论的数据、填充比特和数据负载一起被存储。

[0129] 注意，图中还包括两组不同的异或(“XOR”)门943和945。在这方面，在存在如该示例所示出的135比特的信息的信息时，并不是第一层的每个表都可以向第二层的每个表传递比特线。例如，在所描绘的图中，非线性置换表931-3和931-4只输出11比特，然而在第二层存在12个表，该12个表将从第一层的输出中被馈送至少1比特的输入。类似地，非线性置换表932-3和932-4只向上输出11比特，然而更高一层存在12个表。为了解决该问题并且在加密过程中提供全卷积熵，在加密方向上该示例中的第一层(931)n-1个表中的每个表的一个比特行与来自第一层(931)的n个表中的一个表中的一个比特行进行异或(即，经由电路943)，以产生输出944；相反地，在解密方向上，来自n个表中的每个表的一个比特线(例如，来自表932-2的信号942’)被作为输入馈送到较高层的n-1个表中的一个表(例如，到表931-4)，而相同的比特线也被异或(例如，经由电路945，与信号线948进行异或)以提供输出947。该配置允许将表的第一层(931)上的被编码的所有替换比特传播到第二层(932)上的不同表，并且在解密期间提供原始值的完全恢复。基本上，使用所描绘的XOR电路对比特线进行合并(应用在从第一层到第二层的正向方向和从第二层到第一层的反向方向)，允许调节在

产生的不同数目的比特线与跨n宽和(n-1)宽的查找表所消耗的比特线之间的差异,以便组成许多不同的所期望的数据路径的总比特宽,诸如135或143。通过这种调整,第1层中的每个查找表向第2层中的每个查找表至少提供一个比特行,反之亦然。再次,如上所述,由于所描绘的表提供了简单的查找,并且不是时钟控制元件,因此使用图9B的电路进行加密和解密的速度非常快,达到纳秒级。再次注意,n可以是任何期望的值,并且因此所描绘的结构提供了快速的高熵机制以用于加密和解密任何比特集合。设计中的外部考虑因素是查找存储器阵列的最终大小。例如,针对n＝12,使用每个字大小为12比特的4096字阵列,总计49152比特(6144字节)。针对n-1为11的情况,比特大小总计为2048*11＝22528比特＝2816字节。所有查找阵列的总字节数将占用存储器的大约87千字节。如果反向查找表被部署为单独的实例阵列,那么总存储器的大小将增加一倍,达到175千字节。通过将第二层对应的每个查找表作为其第1层对等方的反向阵列,可以将巧妙的双重用途方案部署为只需占用存储器使用的总计87千字节。考虑在存储器使用、数据路径宽度和与层深度成比例的延迟/时延之间权衡,组成三层或四层结构化的解决方案也是可行的。

[0130] 再次回顾与刚刚描述的结构相关联的一些原理,IC可以提供一些嵌入式加密和/或密码功能。具有一个或多个处理器或处理器核的IC可以使用任何期望的工艺技术来被制造,其中PUF被用于标记可完全恢复的根密钥,该根密钥在相关联的设备或产品的寿命内是可恢复的。即使密码参数不能被内部地存储,所描述的机制通过重新测量PUF以及通过使用点检查特征以实际上来说不可能被攻击者破译的方式来恢复根密钥,从而允许安全地恢复根密钥。因此,根密钥在任何时候都是完全安全的,只有通过所讨论的IC,根密钥才是可被解码和可被理解的。可选地,每个处理器或处理器核可以具有专用于这些任务的电路(例如,每个核或处理器具有单独的根密钥和易失/非易失密码密钥)。板载随机数生成可以被用于辅助根密钥加密,并且提供可以被安全存储的密码密钥(例如,这些次级密钥可以被根密钥加密并且被相同的根密钥解密,即根据需要,在断电后被恢复)。所讨论的IC提供快速加密/解密机制,其允许用于易失性或非易失性存储器的内容被立即加密并且被外部地存储,并且相反地,当被取回时被立即解密,可选地在针对每个虚拟机/过程或处理器是安全的基础上,并且可选地在不可被任何其他虚拟机/过程、处理器或处理器核辨认的基础上。所描绘的架构还提供了新颖的PUF设计以及影子RAM和影子NVM的安全实例化。因此,从前述中应当理解,在促进利用任何工艺技术可用的优秀安全性方面,以及在提供容易在任何设备(例如,包括较新的智能设备和智能电话)中实现的密码过程的方面,所描述的技术和结构为IC架构并且特别是处理器架构提供了各种改进。鉴于上面的描述,本领域普通技术人员将想到许多应用和实现的变型。

[0131] 在一组实施例中,如前所述,点检查功能允许易失性或非易失性根密钥的恢复,同时通过一个或多个存储的检查点回滚经测量的硬件签名,直到该经测量的硬件签名被确认为与原始根密钥匹配为止。在一种可选的情况下,可以使用证明曲线过程。这些实施例可选地可以与本申请所描述的其他功能和结构一起使用或不一起使用。

[0132] 另一组实施例提供了新颖的亚稳态电路阵列,其可以在任何时间被动态测量,无需从例如存储器单元的集合中移除功率或内容。这种阵列可以可选地基于交叉耦合的“NAND”门(如所说明的)或者基于其他电路(诸如通过非限制性的示例,交叉耦合的“NOR”门、锁存器等)。再次,这些实施例可以可选地与本申请所描述的其他功能和结构一起使用

或不一起使用。

[0133] 另一组实施例提供点检查功能，其中不同的加密过程被用于标记特定点(例如，根密钥或硬件签名)。不管是否使用证明曲线密码，使用不同的功能标记同一点都允许标识所有过程所命令的唯一交叉点，并且因而可以标识具体的签名或密钥。在一个实施例中，例如，哈希数据可以以一种方式被外部地存储，该方式允许设备或程序恢复较早检查点以猜测期望的结果，并且经由次级加密过程确认其是否具有正确结果，其中只有一个解匹配所有不同加密过程。在一个非限制性实施例中，可以基于相同的签名数据通过不同的证明曲线来提供不同的加密过程，尽管这不是所有实施例都需要的；针对这些实施例，证明曲线的使用特别地应当被看作是可选的。再次，这些实施例可以可选地与本申请所描述的其他功能和结构一起使用或不一起使用。

[0134] 另一组实施例提供了利用证明曲线的通用密码功能，例如，其中数据被划分为第一集合和第二集合，并且其中第一集合被用作加密第二集合的密钥，并且其中经加密的第二集合被外部地存储并且被用于认证和/或加密。无论它们是否与PUF测量或易失性密钥恢复有关，这种技术可选地被应用于根密钥恢复，即，具体地设想出它们还可以被应用于其他操作数，即，它们通常可以被应用于密码操作。如例如可能地，被应用于零知识证明系统(Zero Knowledge Proof system)。再次，PUF、随机数生成以及其他所描绘的电路和应用应当被视为对这些证明曲线方法是可选的。

[0135] 另一实施例提供了操作IC或器件以追踪漂移的方法。更特别地，IC制造商可选地对IC或器件进行资格评定，以测量最坏情况或其他预期的漂移，而根本无需向其自身透露任何硬件签名。然后，点检查过程基于该资格评定或测量被配置为定期或间歇地点检查IC或设备的状态。以使得从检查点到检查点中遇到的漂移量很小的方式设置检查点生成的频率或定时。然后以允许基于检查点数据的回滚状态的方式存储或标记表示检查点的数据。检查点之间的低漂移有助于最小化处理时间，并且有助于确保确定性地将漂移回滚到原始设备状态或检查点。在一个实施例中，所讨论的经回滚的状态被用作设备根密钥，尽管这不是所有实施例都需要的。此处同样，在本公开的其余部分中阐述的各种实施例、技术和特征可以可选地被使用，但并不是必需的。此外，在实施例中呈现了用于执行根密钥的一步恢复而无需沿检查点谱系向后追踪的技术。

[0136] 在另一可选的实现中，使用多个加密过程标记根密钥的检查点。根密钥可以具有定义的长度，例如，768比特长(或另一长度)。第一加密过程被分解为多个块，使得第一加密过程允许快速隔离差错，例如，而不需要尝试寻找可能被反映在签名的768比特中的任何一个未知比特的差错，而按块加密过程允许将差错定位到一个或多个特定块或分区。然后该差错可以被处理和校正。一个或多个第二加密过程是基于待恢复的整个根密钥(或其表示形式，诸如硬件签名)。冗余签名过程提供了消除混叠的方法(例如，从按块分析产生的可能的多个解)，再次允许原始根密钥的恢复。原始的根密钥可以可选地从PUF中被测量，并且上面描述的其他技术和特性可以可选地与该实现进一步结合或集成。

[0137] 基于对本公开的回顾，其他实施例将立即变得明显，包括：使用非线性置换表提供快速加密和解密，提供特定于VM的隔离加密和解密的电路结构，以及允许对影子RAM或影子NVM进行有效、安全的实现的结构。与本文其他地方所描述的那些实施例一样，这些实施例可以可选地与本文描述的任何结构和技术混合和匹配，并且针对任何实现或实施例，它们

都不被认为是“必不可少的”。

[0138] 可以使用制造集成电路的自动化系统来进一步构造上述电路和技术，并且可以将其描述为在适于控制这种集成电路的制造的非暂态介质上的指令。例如，所描述的组件和系统可以基于设计控制指令被设计为一个或多个集成电路或集成电路的(多个)部分，该设计控制指令使用控制集成电路块的制造的电路形成装置来这样做。指令可以是以存储在例如计算机可读介质(诸如磁带、光盘或磁盘或其他如前所述的非暂态介质)中的数据的形式。这种设计控制指令通常对数据结构或描述电路装置的其他信息或方法进行编码，这些电路装置可以被物理地创建为集成电路的块。尽管可以使用任何适当的格式进行这种编码，这种数据结构通常以Caltech中间格式(CIF)、Calma GDS II流格式(GDSII)、或电子设计交换格式(EDIF)、以及高级描述语言(诸如VHDL或Verilog或另一形式的寄存器传送语言(“RTL”)描述)的形式进行编写。集成电路设计领域的技术人员可以根据上述类型的示意图和对应的描述来开发这种数据结构，并且在计算机可读介质上对数据结构进行编码。集成电路制造领域的技术人员然后可以使用这种经编码的数据来制造包括本文所述的一个或多个电路的集成电路。

[0139] 在上述的描述和附图中，阐述了特定的术语和附图符号，以提供对现有技术的透彻理解。在某些实例中，术语和符号可以表示实践该技术不需要的特定细节。例如，尽管这里使用了术语“第一”和“第二”，但除非另有说明，该语言并不旨在提供任何指定的顺序，而仅用于辅助解释技术的要素。在某些实例中，术语和符号可以表示实践那些实施例不需要的特定细节。术语“示例性的”和“实施例”被用于标识示例，而不是偏好或要求。此外，尽管参考特定实施例描述了本文中技术，但是应当理解，这些实施例仅是该技术的原理和应用的说明。因此，应当理解，可以在不脱离技术的精神和范围的情况下，对说明性实施例进行各种修改，并且可以设计其他布置。

[0140] 在不脱离本公开的更广泛的精神和范围的情况下，可以对本文呈现的实施例进行各种修改和改变。至少在可行的情况下，任何实施例的特征或方面可以与任何其他实施例结合或者代替其对应的特征或方面来被应用。因此，各种实施例的特征不旨在相对于彼此是排他性的，并且说明书和附图应被认为是说明性的而非限制性的。

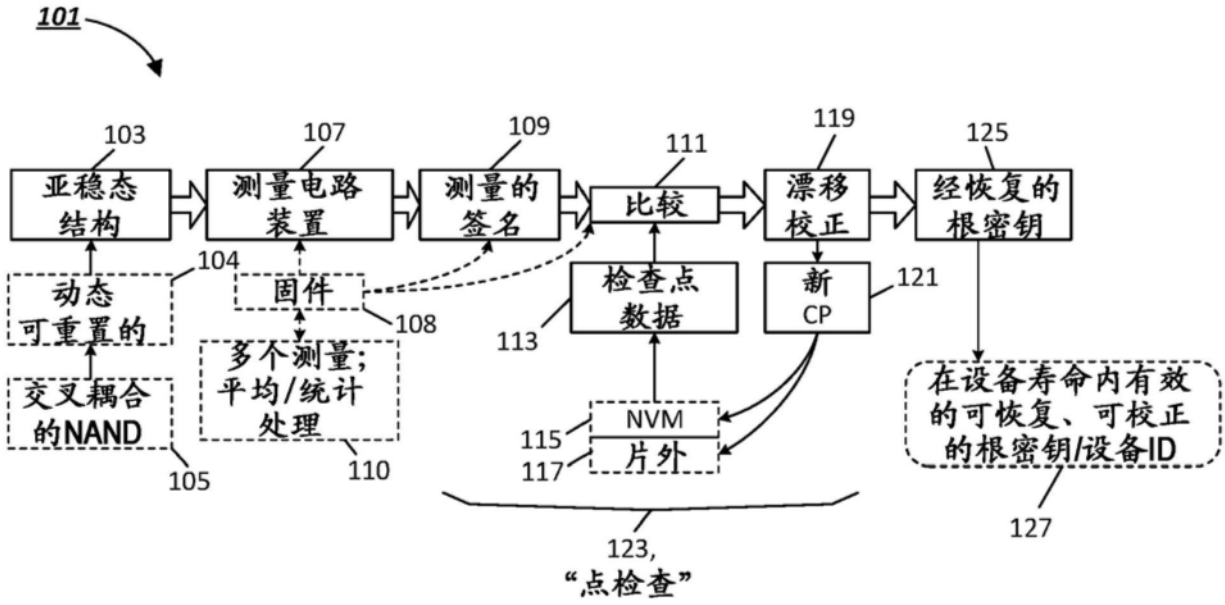

101
103
亚稳态结构
107
测量电路装置
109
测量的签名
111
比较
119
漂移校正
125
经恢复的根密钥
104
动态可重置的
固件
108
检查点数据
113
新CP
121
交叉耦合的NAND
105
多个测量;平均/统计处理
110
NVM
115
片外
117
在设备寿命内有效的可恢复、可校正的根密钥/设备ID
127
123,
“点检查”

图1A

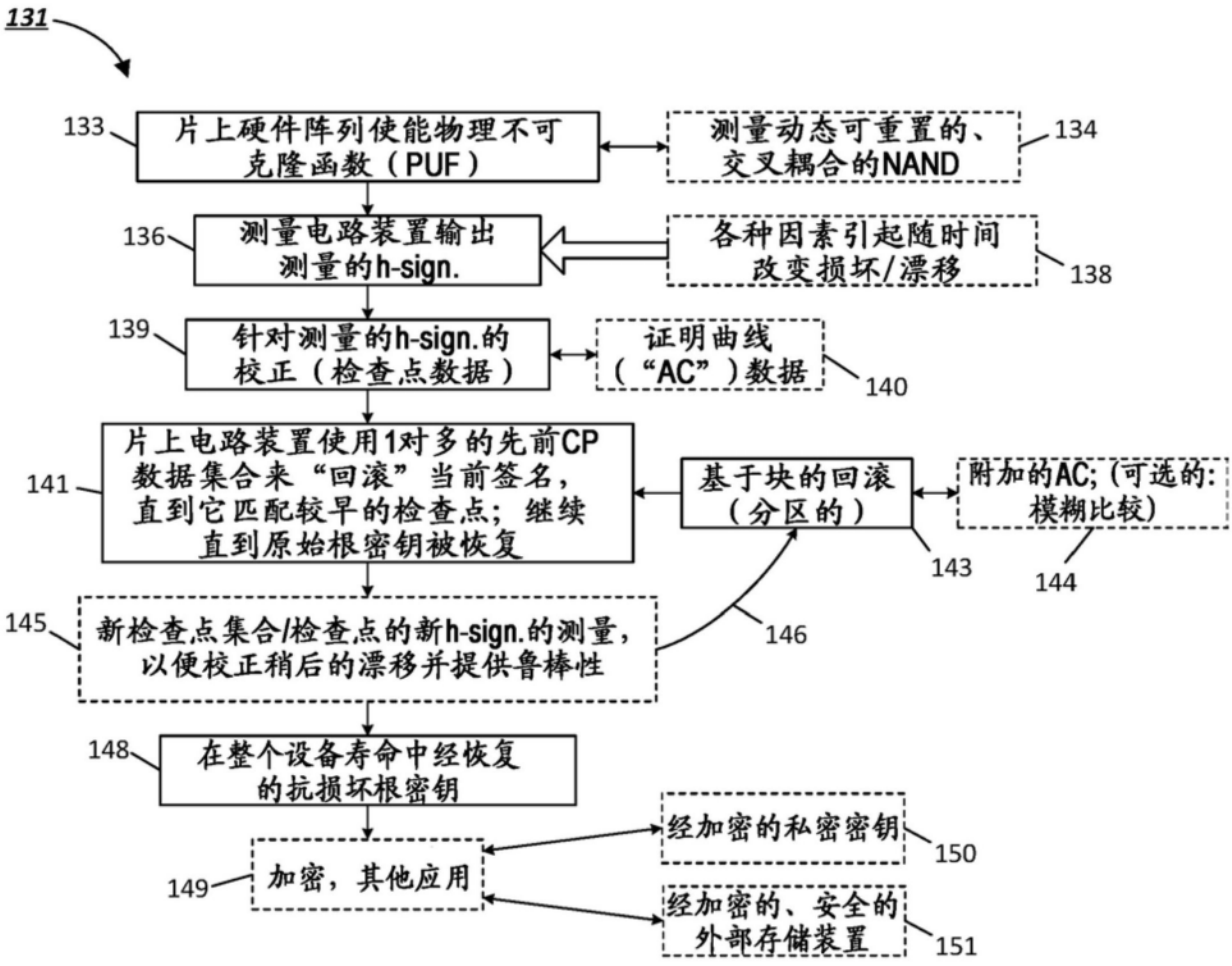

131
133
片上硬件阵列使能物理不可克隆函数（PUF）
测量动态可重置的、交叉耦合的NAND
134
136
测量电路装置输出测量的h-sign.
各种因素引起随时间改变损坏/漂移
138
139
针对测量的h-sign.的校正（检查点数据）
证明曲线（“AC”）数据
140
141
片上电路装置使用1对多的先前CP数据集合来“回滚”当前签名，直到它匹配较早的检查点；继续直到原始根密钥被恢复
基于块的回滚（分区的）
附加的AC;(可选的:模糊比较)
143
144
145
新检查点集合/检查点的新h-sign.的测量，以便校正稍后的漂移并提供鲁棒性
146
148
在整个设备寿命中经恢复的抗损坏根密钥
经加密的私密密钥
150
149
加密，其他应用
经加密的、安全的外部存储装置
151

图1B

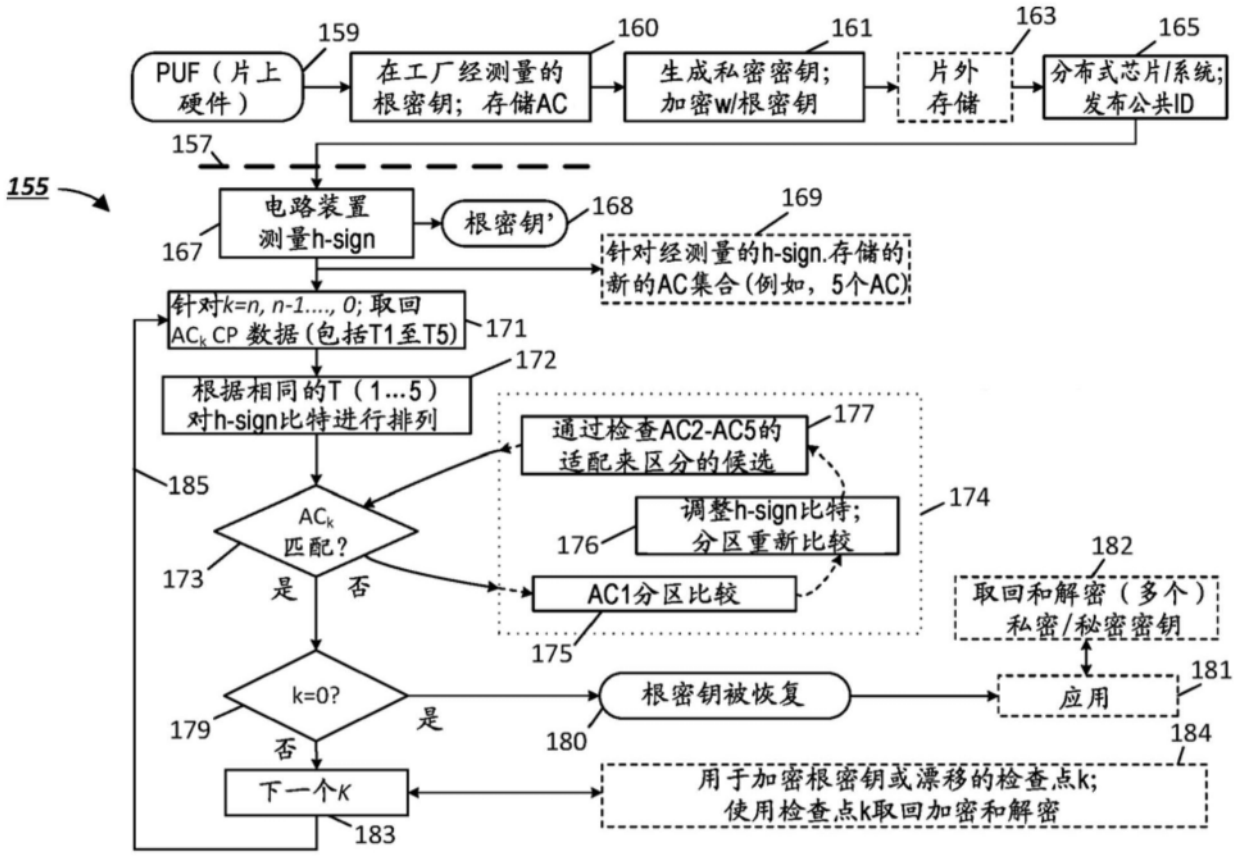

155
PUF（片上硬件）
159
在工厂经测量的根密钥；存储AC
160
生成私密密钥；加密w/根密钥
161
片外存储
163
分布式芯片/系统；发布公共ID
165
157
电路装置测量h-sign
167
根密钥'
168
169
针对经测量的h-sign.存储的新的AC集合（例如，5个AC）
针对k=n, n-1..., 0; 取回 AC_k CP 数据（包括T1至T5）
171
根据相同的T（1...5）对h-sign比特进行排列
172
通过检查AC2-AC5的适配来区分的候选
177
调整h-sign比特；分区重新比较
176
174
185
AC_k 匹配？
173
是
否
AC1分区比较
175
182
取回和解密（多个）私密/秘密密钥
k=0?
179
是
否
根密钥被恢复
180
应用
181
184
下一个K
用于加密根密钥或漂移的检查点k；使用检查点k取回加密和解密
183

图1C

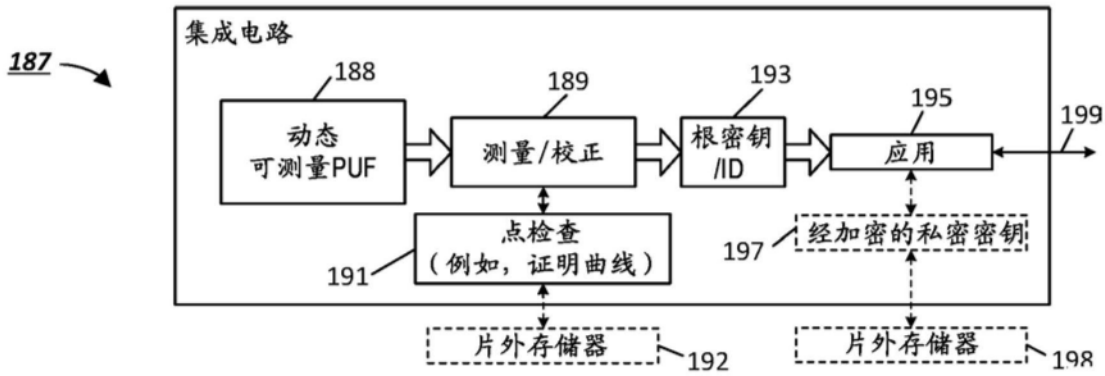

187
集成电路
188
动态可测量PUF
189
测量/校正
193
根密钥/ID
195
应用
199
点检查（例如，证明曲线）
191
197
经加密的私密密钥
片外存储器
192
片外存储器
198

图1D

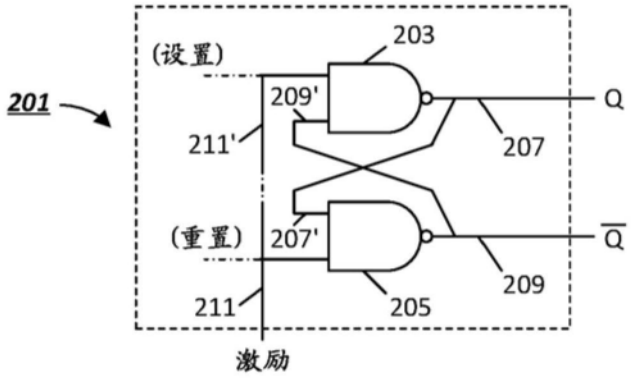

201
(设置)
203
209'
Q
211'
207
(重置)
207'
$\overline{Q}$
209
211
205
激励

图2A

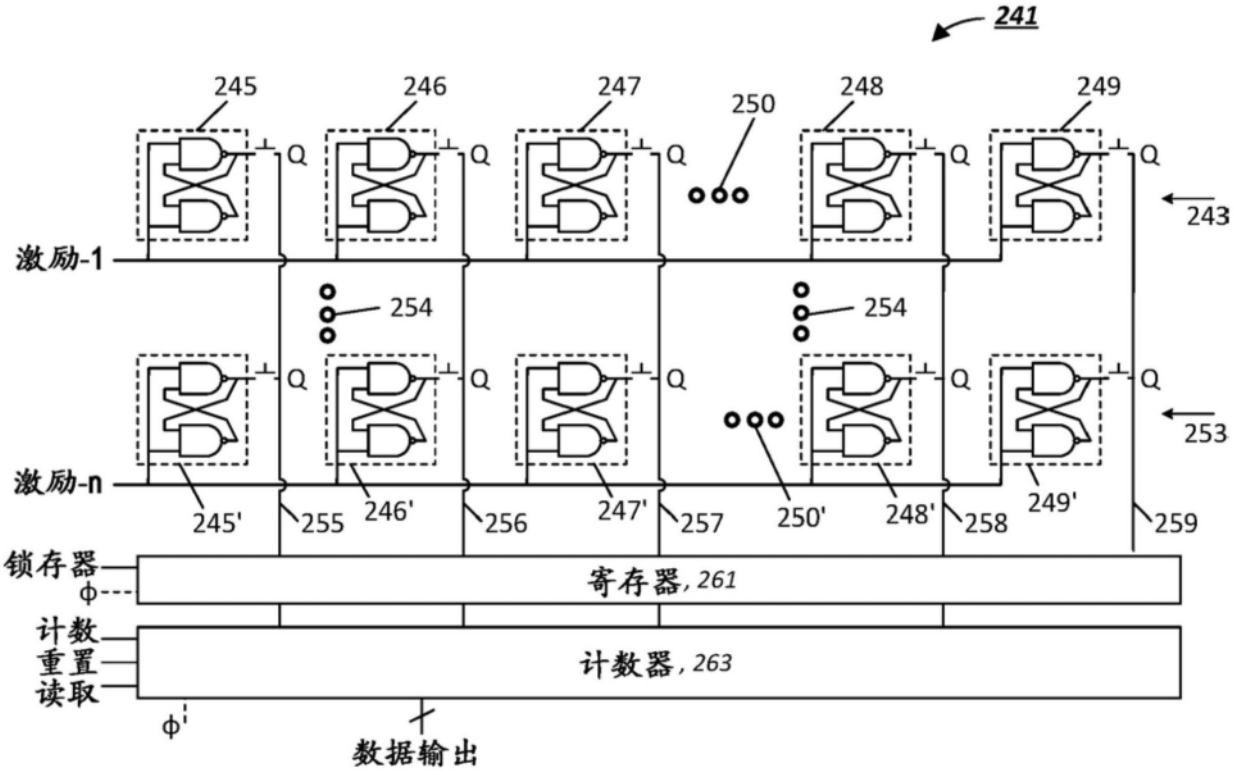

241
245
246
247
250
248
249
243
激励-1
254
254
253
激励-n
245'
255
246'
256
247'
257
250'
248'
258
249'
259
锁存器
ϕ
寄存器, 261
计数
重置
读取
计数器, 263
ϕ'
数据输出

图2B

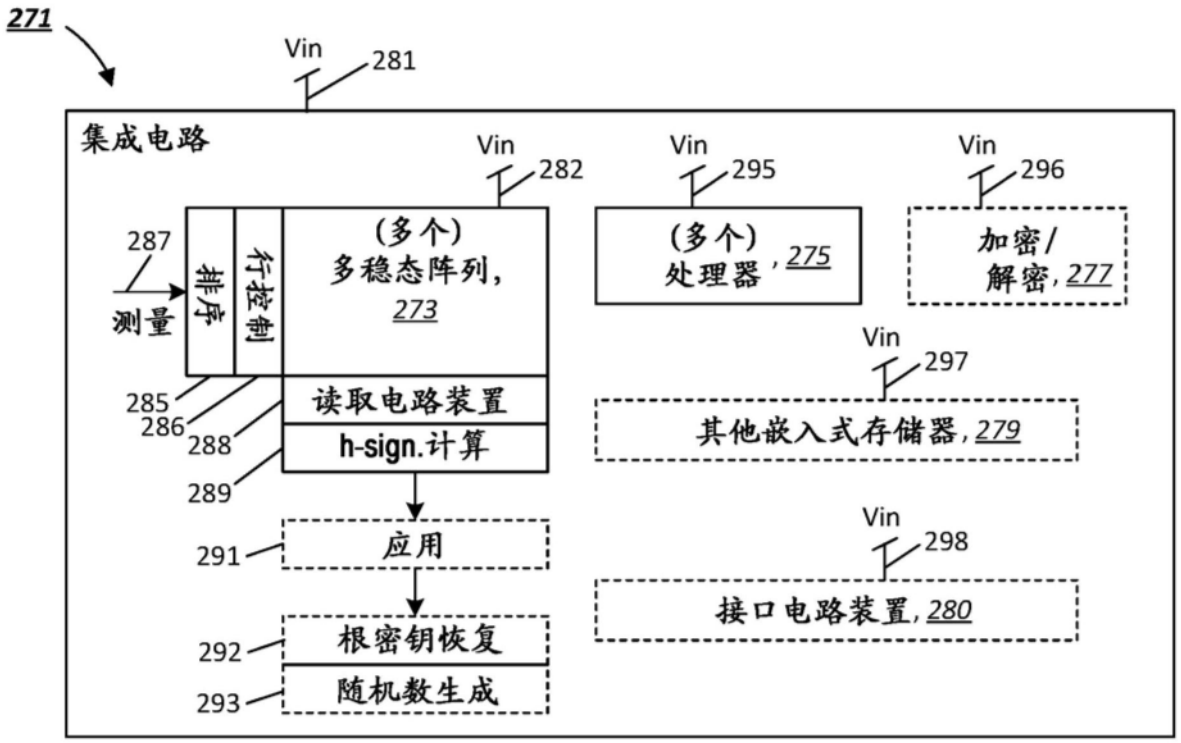

271
Vin
281
集成电路
Vin
282
Vin
295
Vin
296
287
测量
排序
行控制
(多个)
多稳态阵列,
273
(多个)
处理器,275
加密/
解密,277
Vin
297
285
286
288
289
读取电路装置
h-sign.计算
其他嵌入式存储器,279
Vin
298
291
应用
接口电路装置,280
292
根密钥恢复
293
随机数生成

图2C

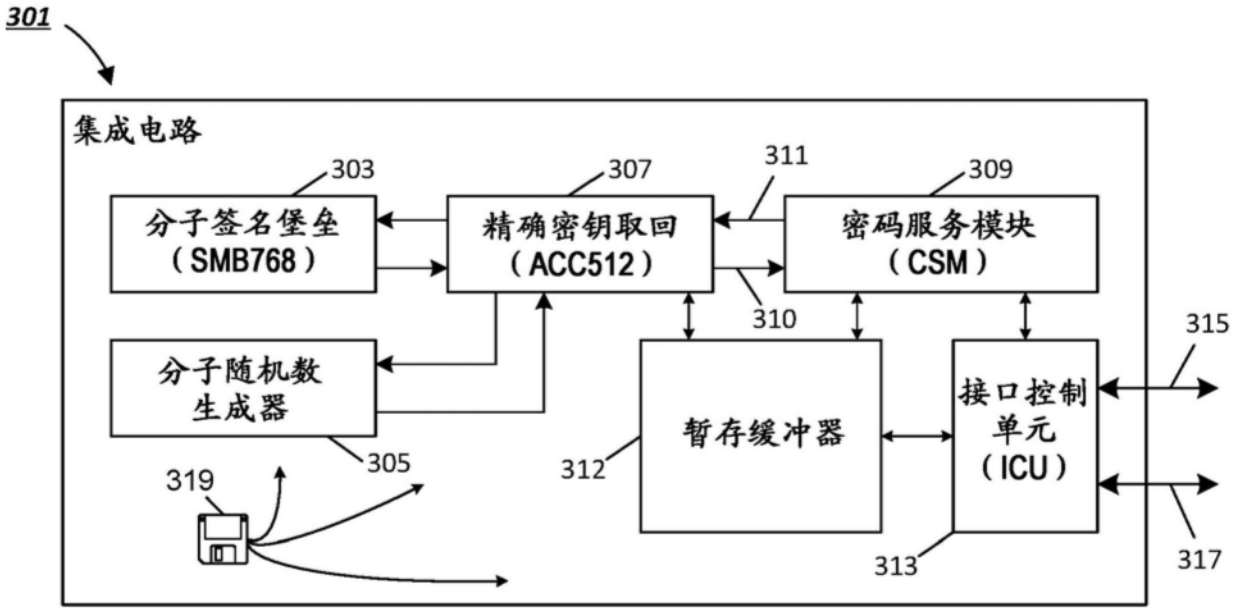

301
集成电路
303
307
311
309
分子签名堡垒
(SMB768)
精确密钥取回
(ACC512)
密码服务模块
(CSM)
310
315
分子随机数
生成器
305
暂存缓冲器
接口控制
单元
(ICU)
312
319
313
317

图3A

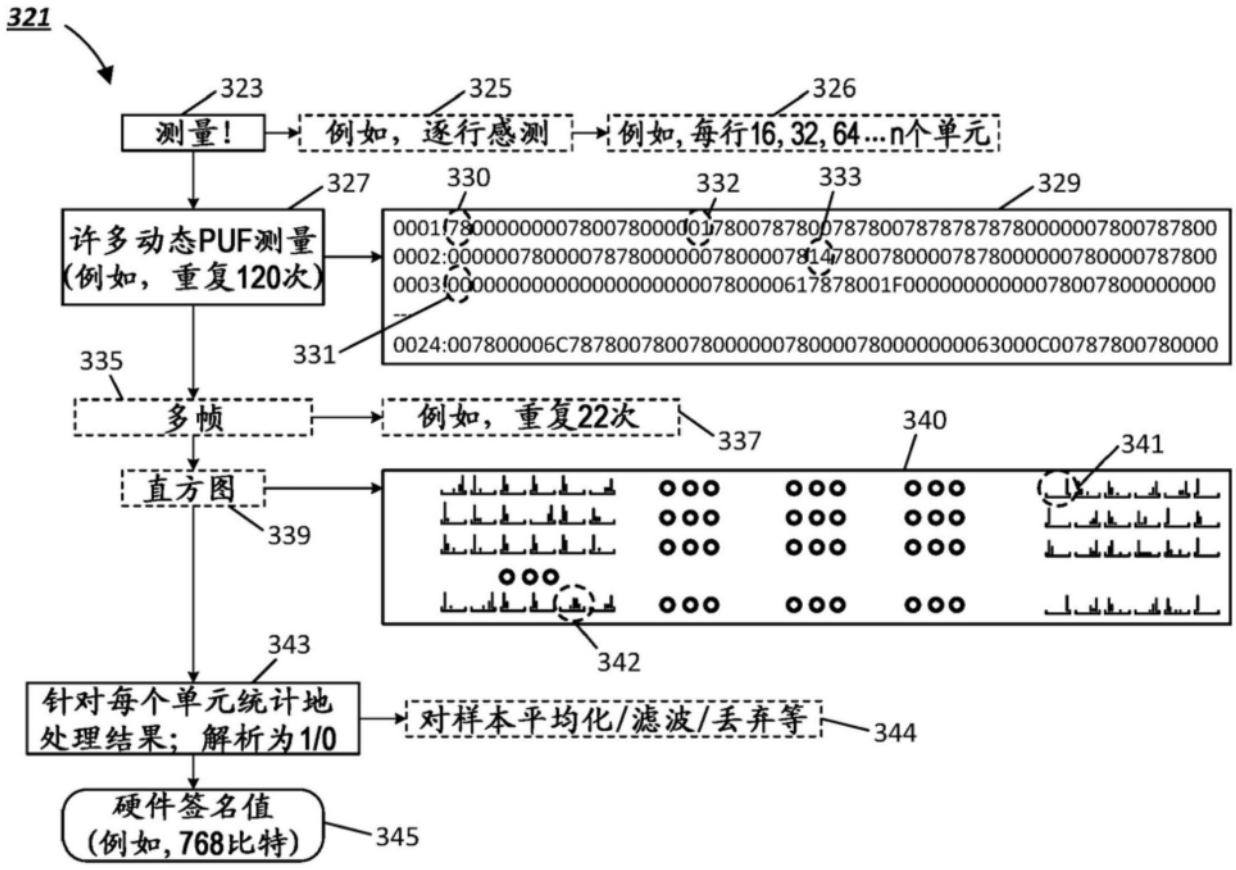

321
323
测量!
325
例如，逐行感测
326
例如,每行16,32,64...n个单元
327
许多动态PUF测量
(例如，重复120次)
330
332
333
329
0001:
0002:
0003:
0024:007800006C787800780078000000780000780000000063000C00787800780000
331
335
多帧
例如，重复22次
337
340
341
直方图
339
342
343
针对每个单元统计地
处理结果；解析为1/0
对样本平均化/滤波/丢弃等
344
硬件签名值
(例如,768比特)
345

图3B

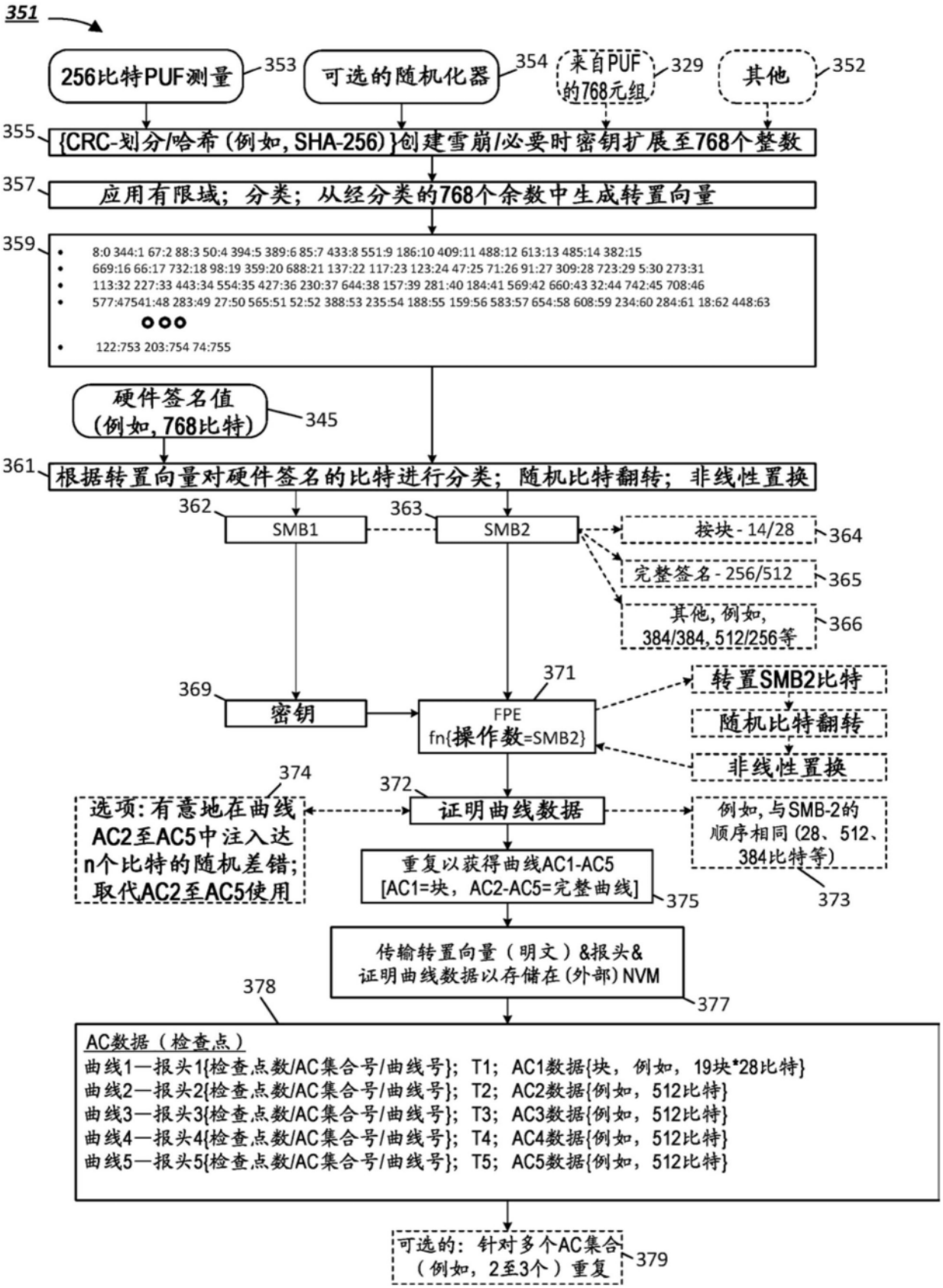

351
256比特PUF测量
353
可选的随机化器
354
来自PUF的768元组
329
其他
352
355
{CRC-划分/哈希(例如,SHA-256)}创建雪崩/必要时密钥扩展至768个整数
357
应用有限域；分类；从经分类的768个余数中生成转置向量
359
8:0 344:1 67:2 88:3 50:4 394:5 389:6 85:7 433:8 551:9 186:10 409:11 488:12 613:13 485:14 382:15
669:16 66:17 732:18 98:19 359:20 688:21 137:22 117:23 123:24 47:25 71:26 91:27 309:28 723:29 5:30 273:31
113:32 227:33 443:34 554:35 427:36 230:37 644:38 157:39 281:40 184:41 569:42 660:43 32:44 742:45 708:46
577:47541:48 283:49 27:50 565:51 52:52 388:53 235:54 188:55 159:56 583:57 654:58 608:59 234:60 284:61 18:62 448:63
122:753 203:754 74:755
硬件签名值(例如,768比特)
345
361
根据转置向量对硬件签名的比特进行分类；随机比特翻转；非线性置换
362
SMB1
363
SMB2
按块-14/28
364
完整签名-256/512
365
其他,例如,384/384,512/256等
366
369
密钥
371
FPE
fn{操作数=SMB2}
转置SMB2比特
随机比特翻转
非线性置换
374
选项:有意地在曲线AC2至AC5中注入达n个比特的随机差错;取代AC2至AC5使用
372
证明曲线数据
例如,与SMB-2的顺序相同(28、512、384比特等)
373
重复以获得曲线AC1-AC5
[AC1=块，AC2-AC5=完整曲线]
375
传输转置向量（明文）&报头&
证明曲线数据以存储在(外部)NVM
377
378
AC数据（检查点）
曲线1—报头1{检查点数/AC集合号/曲线号}；T1；AC1数据{块，例如，19块*28比特}
曲线2—报头2{检查点数/AC集合号/曲线号}；T2；AC2数据{例如，512比特}
曲线3—报头3{检查点数/AC集合号/曲线号}；T3；AC3数据{例如，512比特}
曲线4—报头4{检查点数/AC集合号/曲线号}；T4；AC4数据{例如，512比特}
曲线5—报头5{检查点数/AC集合号/曲线号}；T5；AC5数据{例如，512比特}
可选的：针对多个AC集合（例如，2至3个）重复
379

图3C

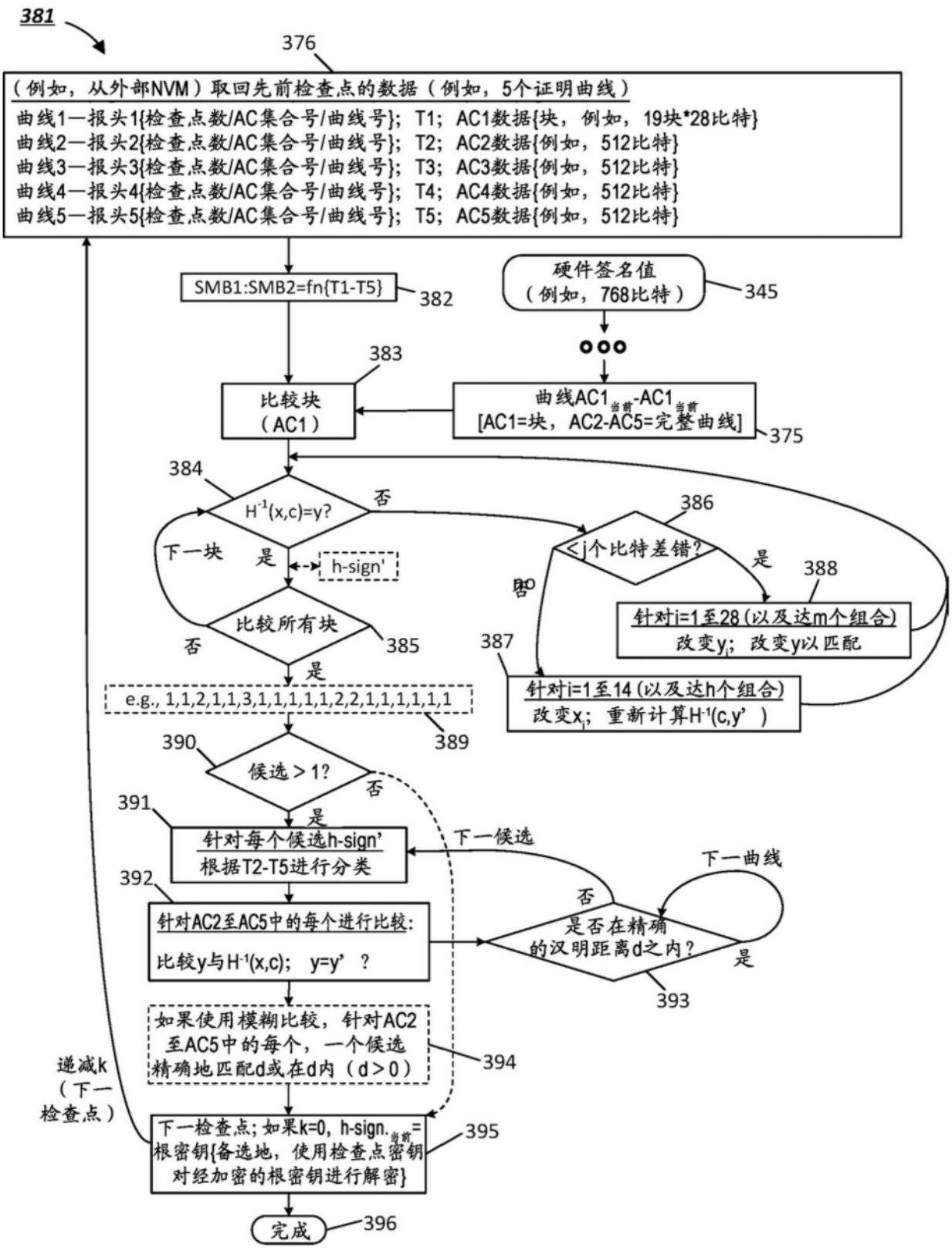

381
376
(例如，从外部NVM）取回先前检查点的数据（例如，5个证明曲线）
曲线1—报头1{检查点数/AC集合号/曲线号}; T1; AC1数据{块，例如，19块*28比特}
曲线2—报头2{检查点数/AC集合号/曲线号}; T2; AC2数据{例如，512比特}
曲线3—报头3{检查点数/AC集合号/曲线号}; T3; AC3数据{例如，512比特}
曲线4—报头4{检查点数/AC集合号/曲线号}; T4; AC4数据{例如，512比特}
曲线5—报头5{检查点数/AC集合号/曲线号}; T5; AC5数据{例如，512比特}
SMB1:SMB2=fn{T1-T5}
382
硬件签名值
（例如，768比特）
345
383
比较块
（AC1）
曲线AC1当前-AC1当前
[AC1=块，AC2-AC5=完整曲线]
375
384
H-1(x,c)=y?
否
386
<j个比特差错?
是
388
下一块
是
h-sign'
否
针对i=1至28（以及达m个组合）
改变yi；改变y以匹配
比较所有块
385
否
387
针对i=1至14（以及达h个组合）
改变xi；重新计算H-1(c,y')
是
e.g., 1,1,2,1,1,3,1,1,1,1,1,2,2,1,1,1,1,1,1
389
390
候选>1?
否
是
391
针对每个候选h-sign'
根据T2-T5进行分类
下一候选
下一曲线
392
针对AC2至AC5中的每个进行比较:
比较y与H-1(x,c); y=y' ?
否
是否在精确
的汉明距离d之内?
是
393
如果使用模糊比较，针对AC2
至AC5中的每个，一个候选
精确地匹配d或在d内（d>0）
394
递减k
（下一
检查点）
下一检查点；如果k=0，h-sign.当前=
根密钥{备选地，使用检查点密钥
对经加密的根密钥进行解密}
395
完成
396

图3D

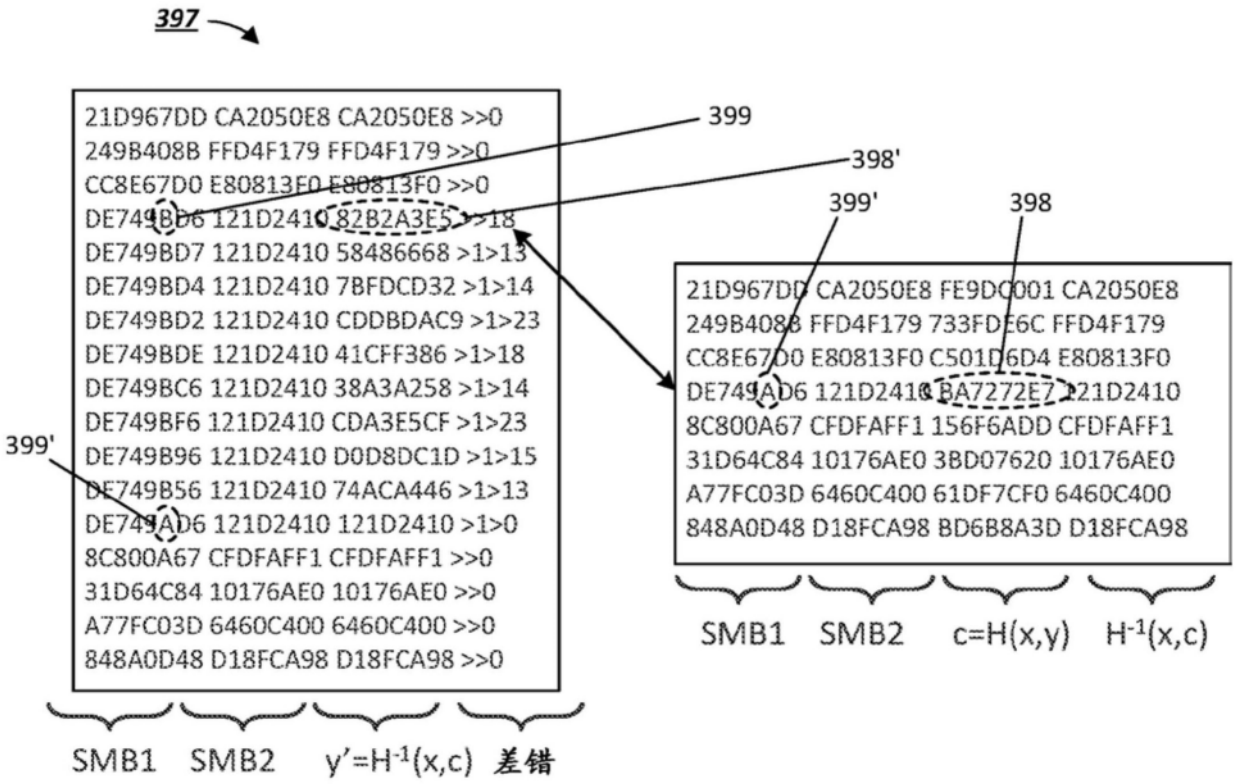

397
21D967DD CA2050E8 CA2050E8 >>0
249B408B FFD4F179 FFD4F179 >>0
CC8E67D0 E80813F0 E80813F0 >>0
DE749BD6 121D2410 82B2A3E5 >>18
DE749BD7 121D2410 58486668 >1>13
DE749BD4 121D2410 7BFDCD32 >1>14
DE749BD2 121D2410 CDDBDAC9 >1>23
DE749BDE 121D2410 41CFF386 >1>18
DE749BC6 121D2410 38A3A258 >1>14
DE749BF6 121D2410 CDA3E5CF >1>23
DE749B96 121D2410 D0D8DC1D >1>15
DE749B56 121D2410 74ACA446 >1>13
DE749AD6 121D2410 121D2410 >1>0
8C800A67 CFDFAFF1 CFDFAFF1 >>0
31D64C84 10176AE0 10176AE0 >>0
A77FC03D 6460C400 6460C400 >>0
848A0D48 D18FCA98 D18FCA98 >>0
SMB1
SMB2
y'=H-1(x,c)
差错
399
398'
399'
398
399'
21D967DD CA2050E8 FE9DC001 CA2050E8
249B408B FFD4F179 733FDE6C FFD4F179
CC8E67D0 E80813F0 C501D6D4 E80813F0
DE749AD6 121D2410 BA7272E7 121D2410
8C800A67 CFDFAFF1 156F6ADD CFDFAFF1
31D64C84 10176AE0 3BD07620 10176AE0
A77FC03D 6460C400 61DF7CF0 6460C400
848A0D48 D18FCA98 BD6B8A3D D18FCA98
SMB1
SMB2
c=H(x,y)
H-1(x,c)

图3E

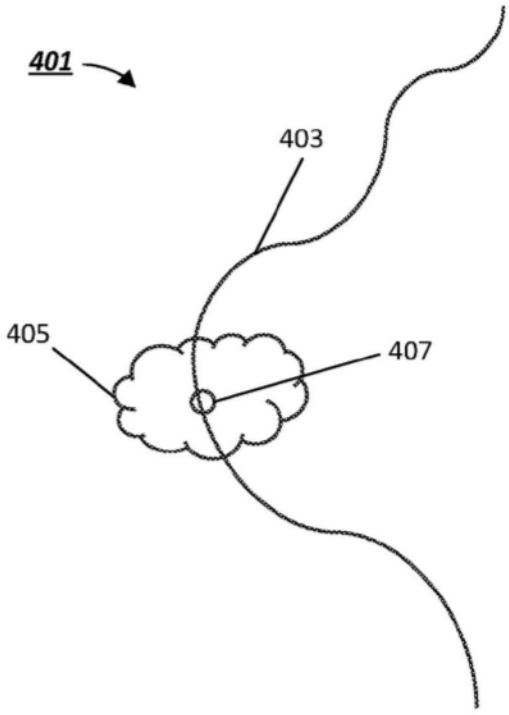

401
403
405
407

图4A

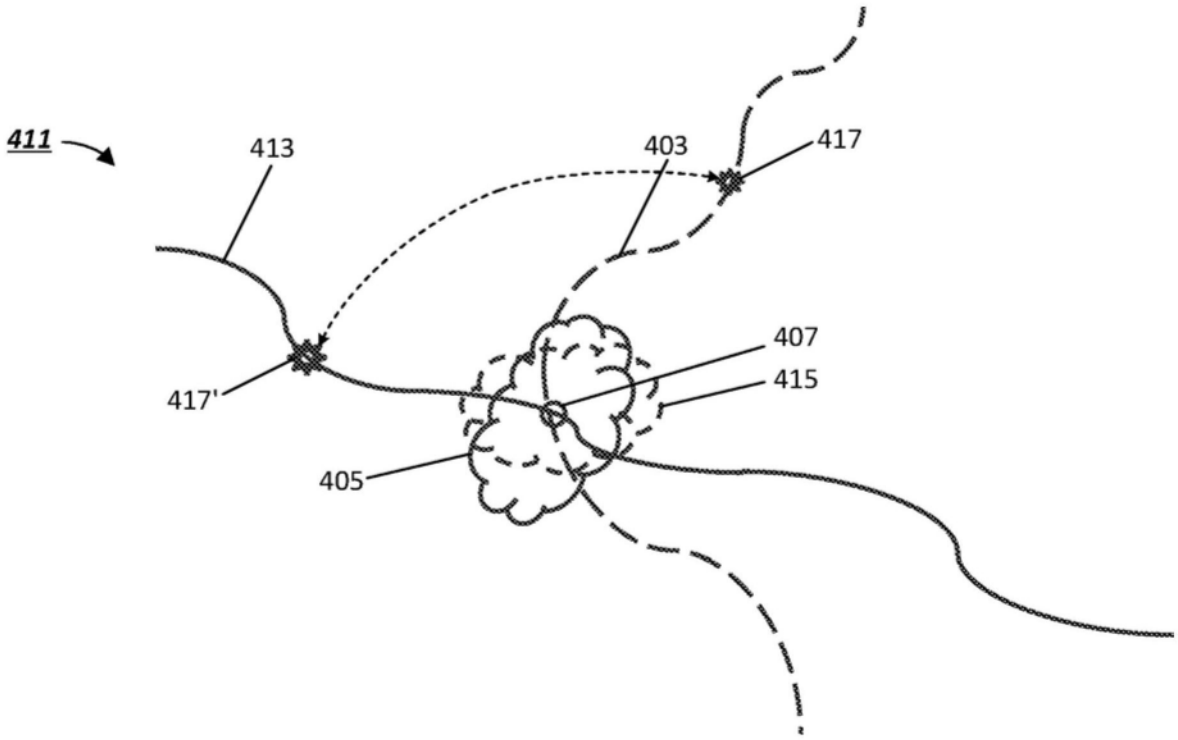

411
413
403
417
417'
407
415
405

图4B

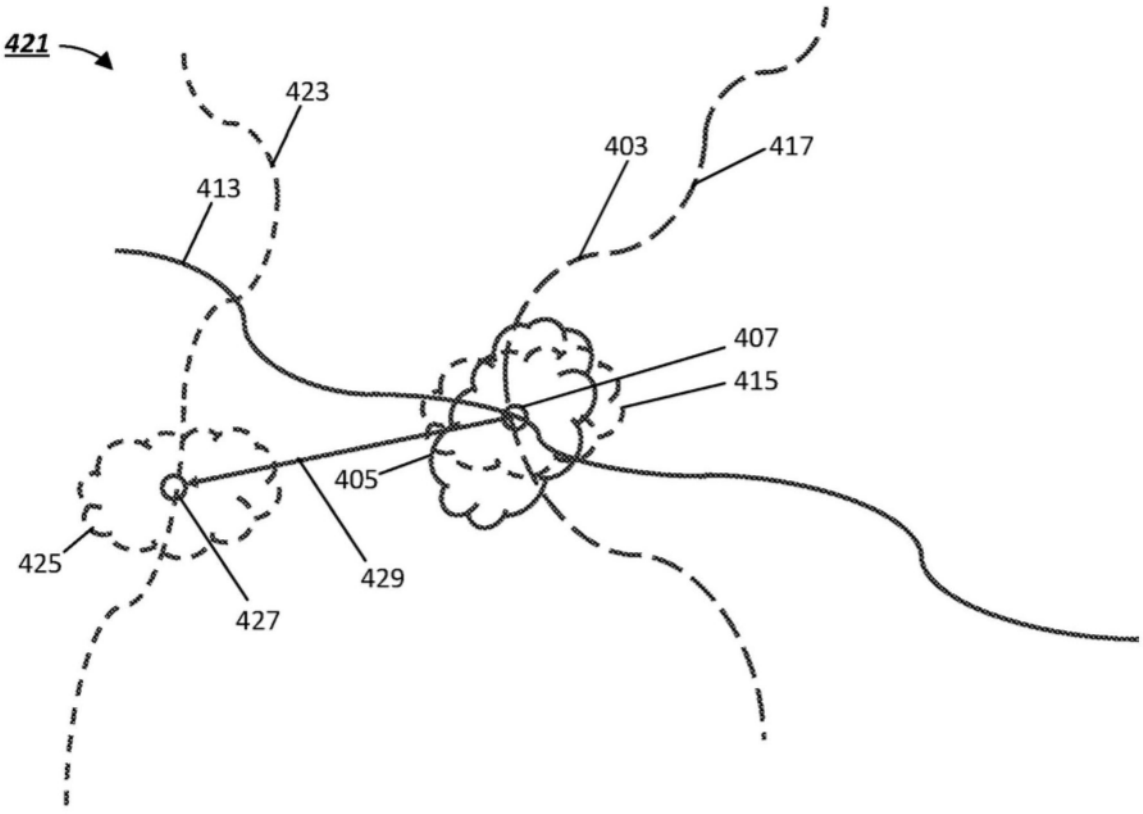

421
423
403
417
413
407
415
405
425
429
427

图4C

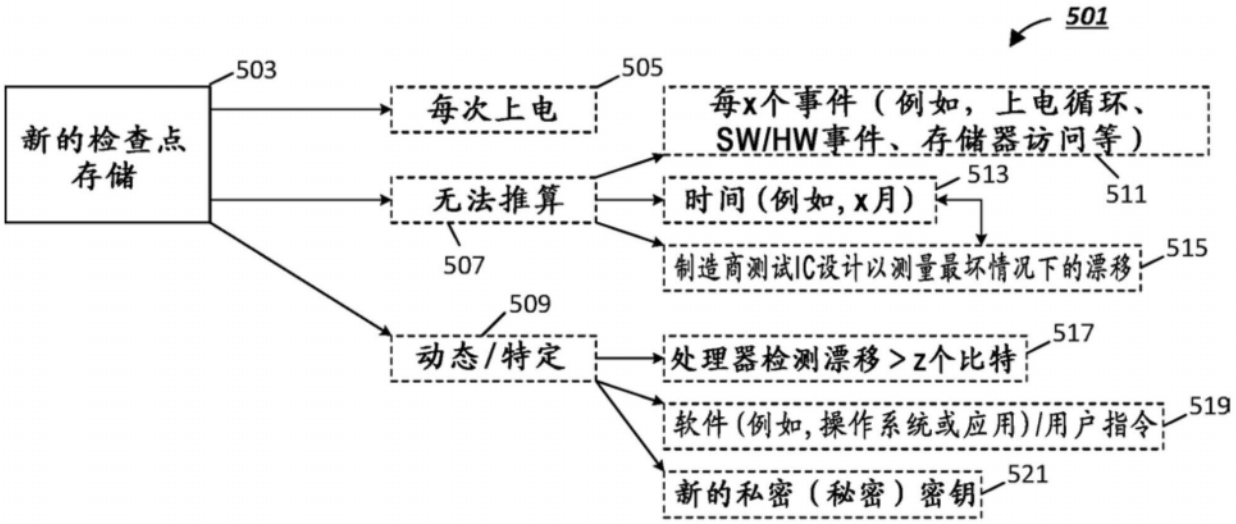

501
503
新的检查点存储
505
每次上电
每x个事件（例如，上电循环、SW/HW事件、存储器访问等）
511
无法推算
513
时间（例如，x月）
507
制造商测试IC设计以测量最坏情况下的漂移
515
509
动态/特定
517
处理器检测漂移>z个比特
519
软件（例如，操作系统或应用）/用户指令
521
新的私密（秘密）密钥

图5

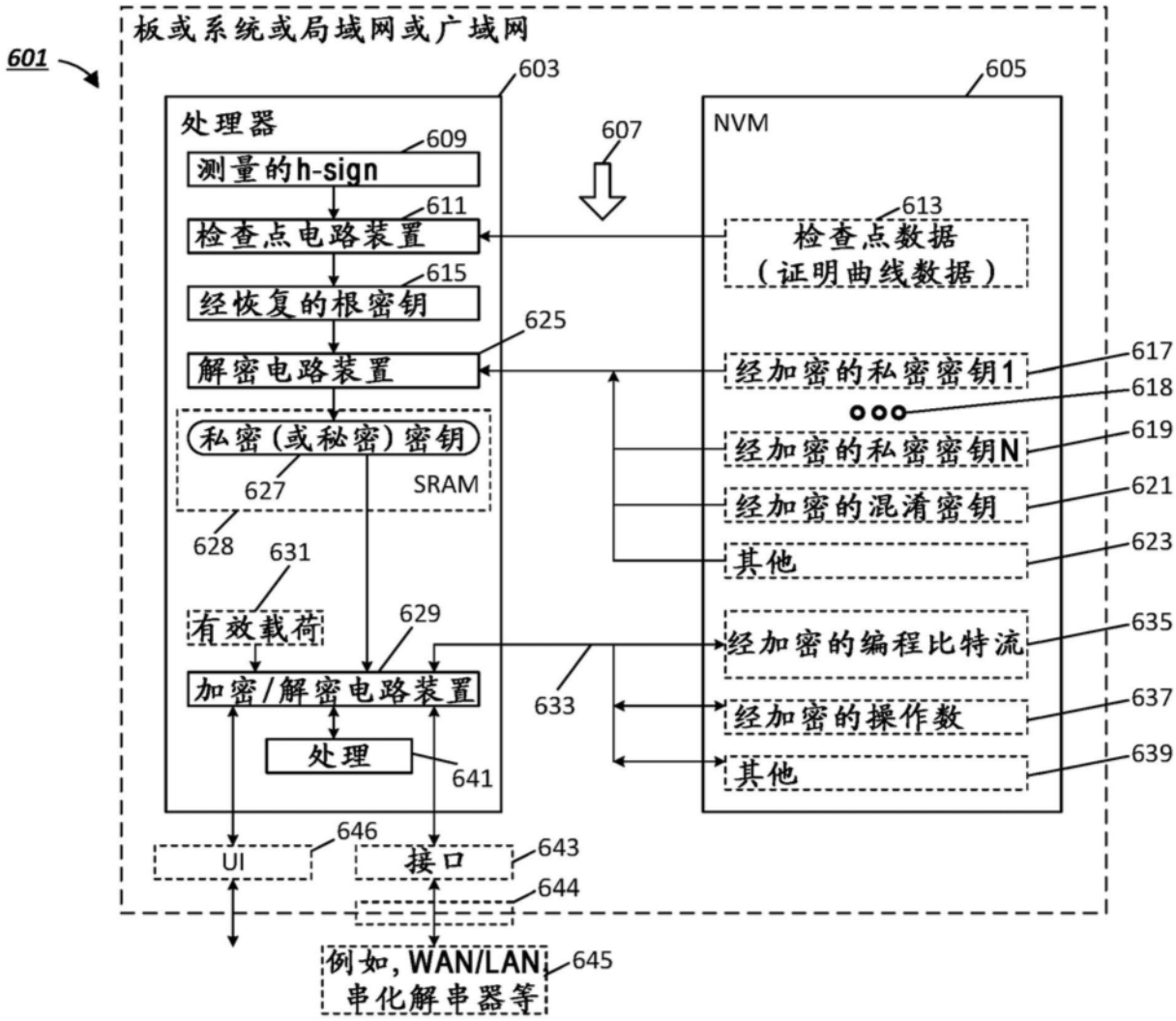

601
板或系统或局域网或广域网
603
处理器
609
测量的h-sign
611
检查点电路装置
615
经恢复的根密钥
625
解密电路装置
私密（或秘密）密钥
627
SRAM
628
631
有效载荷
629
加密/解密电路装置
处理
641
607
605
NVM
613
检查点数据（证明曲线数据）
617
经加密的私密密钥1
618
619
经加密的私密密钥N
621
经加密的混淆密钥
623
其他
635
经加密的编程比特流
633
637
经加密的操作数
639
其他
646
UI
643
接口
644
例如，WAN/LAN串化解串器等
645

图6

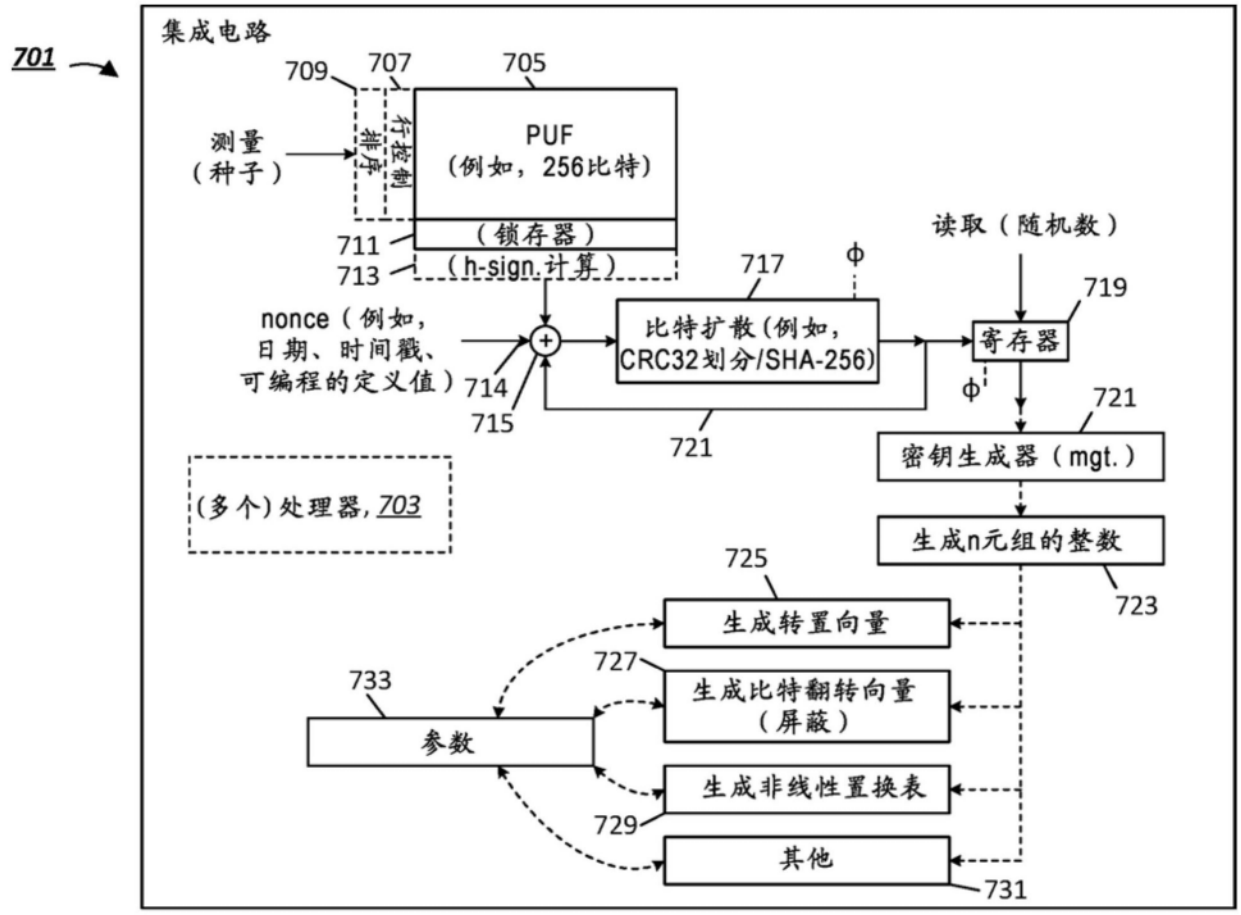

701
集成电路
709
707
705
测量
（种子）
行控制
排序
PUF
（例如，256比特）
（锁存器）
711
713
（h-sign.计算）
读取（随机数）
717
Φ
719
nonce（例如，
日期、时间戳、
可编程的定义值）
714
715
比特扩散(例如，
CRC32划分/SHA-256)
寄存器
Φ
721
721
密钥生成器（mgt.）
(多个)处理器，703
生成n元组的整数
723
725
生成转置向量
727
733
生成比特翻转向量
（屏蔽）
参数
生成非线性置换表
729
其他
731

图7

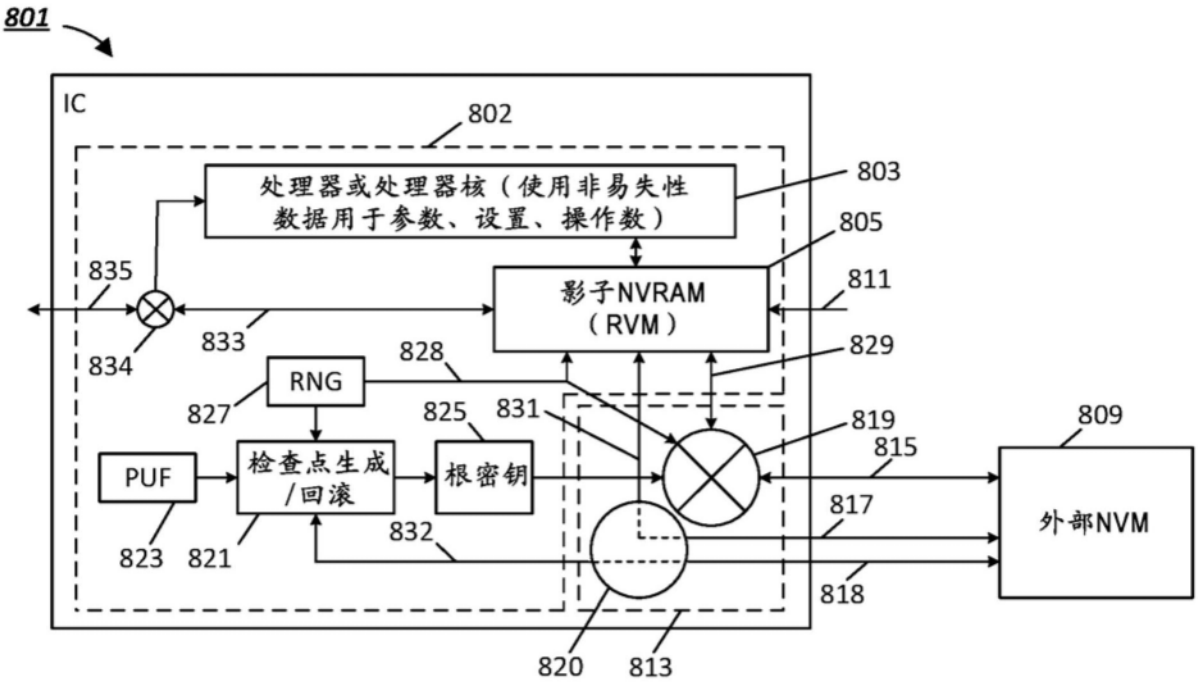

801
IC
802
处理器或处理器核（使用非易失性数据用于参数、设置、操作数）
803
805
影子NVRAM（RVM）
811
835
833
834
828
RNG
827
825
831
829
819
809
815
PUF
检查点生成/回滚
根密钥
817
外部NVM
832
823
821
818
820
813

图8A

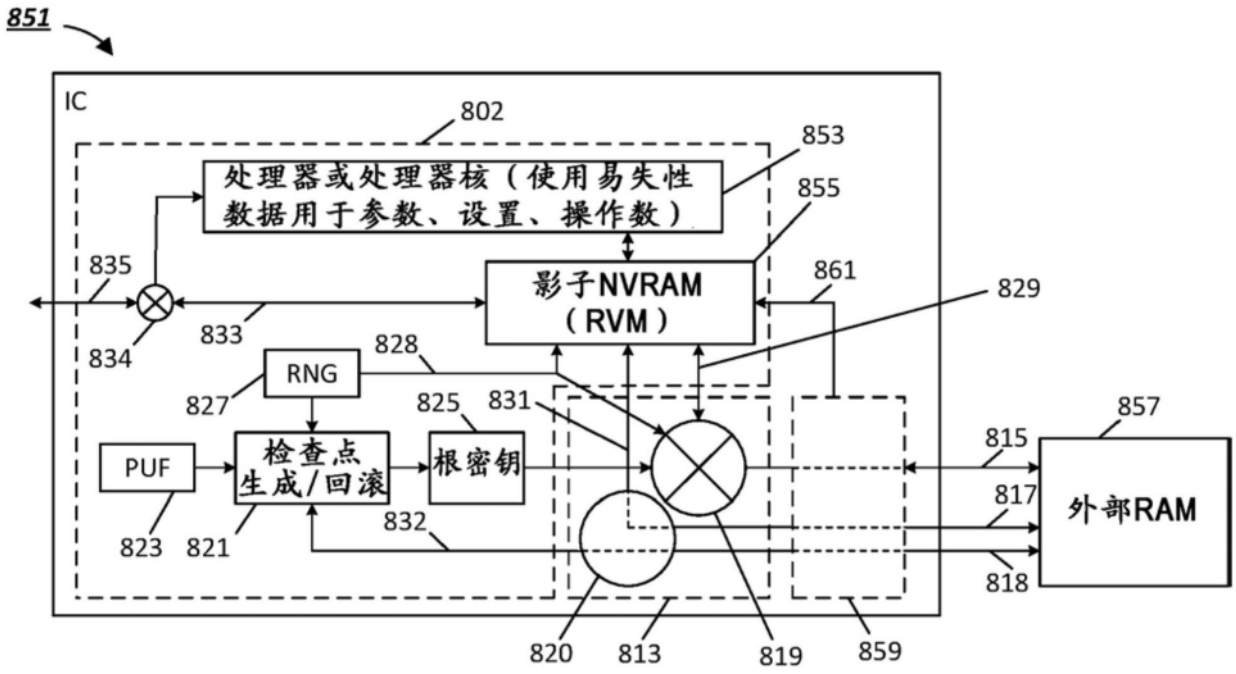

851
IC
802
853
处理器或处理器核（使用易失性数据用于参数、设置、操作数）
855
835
影子NVRAM（RVM）
861
829
833
834
828
RNG
827
825
831
857
815
PUF
检查点生成/回滚
根密钥
817
外部RAM
832
823
821
818
820
813
819
859

图8B

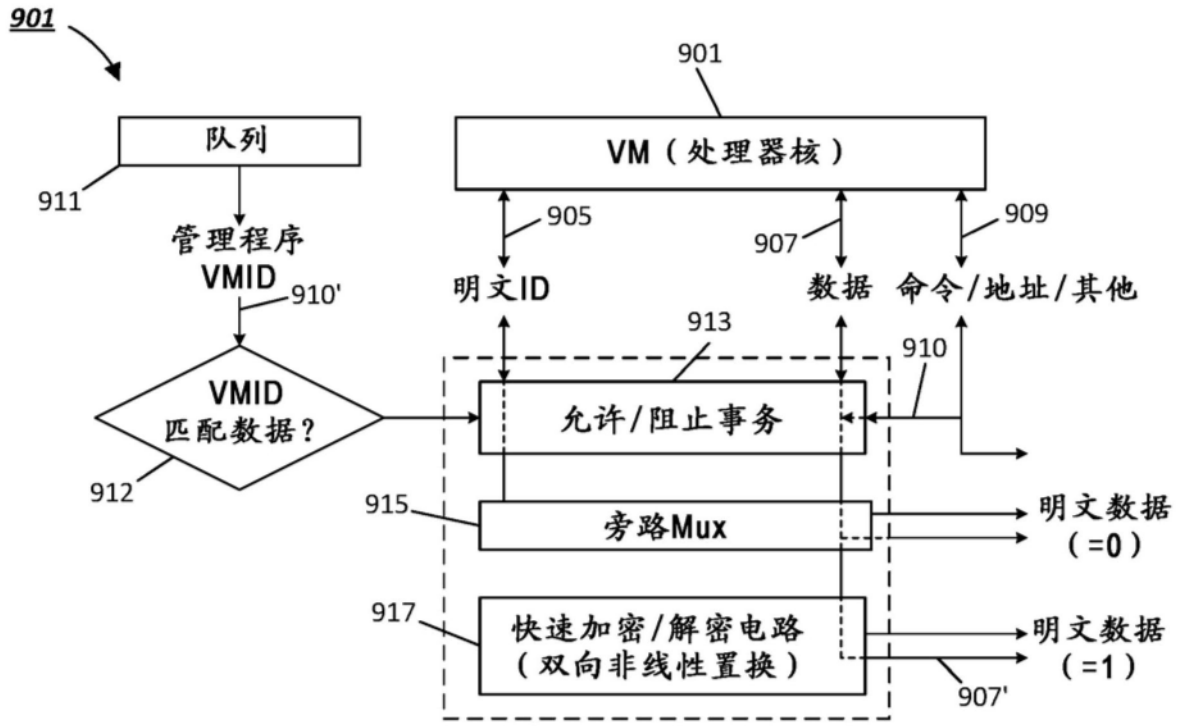

901
901
队列
911
VM（处理器核）
905
907
909
管理程序
VMID
910'
明文ID
数据
命令/地址/其他
913
910
VMID
匹配数据？
912
允许/阻止事务
915
旁路Mux
明文数据
（=0）
917
快速加密/解密电路
（双向非线性置换）
明文数据
（=1）
907'

图9A

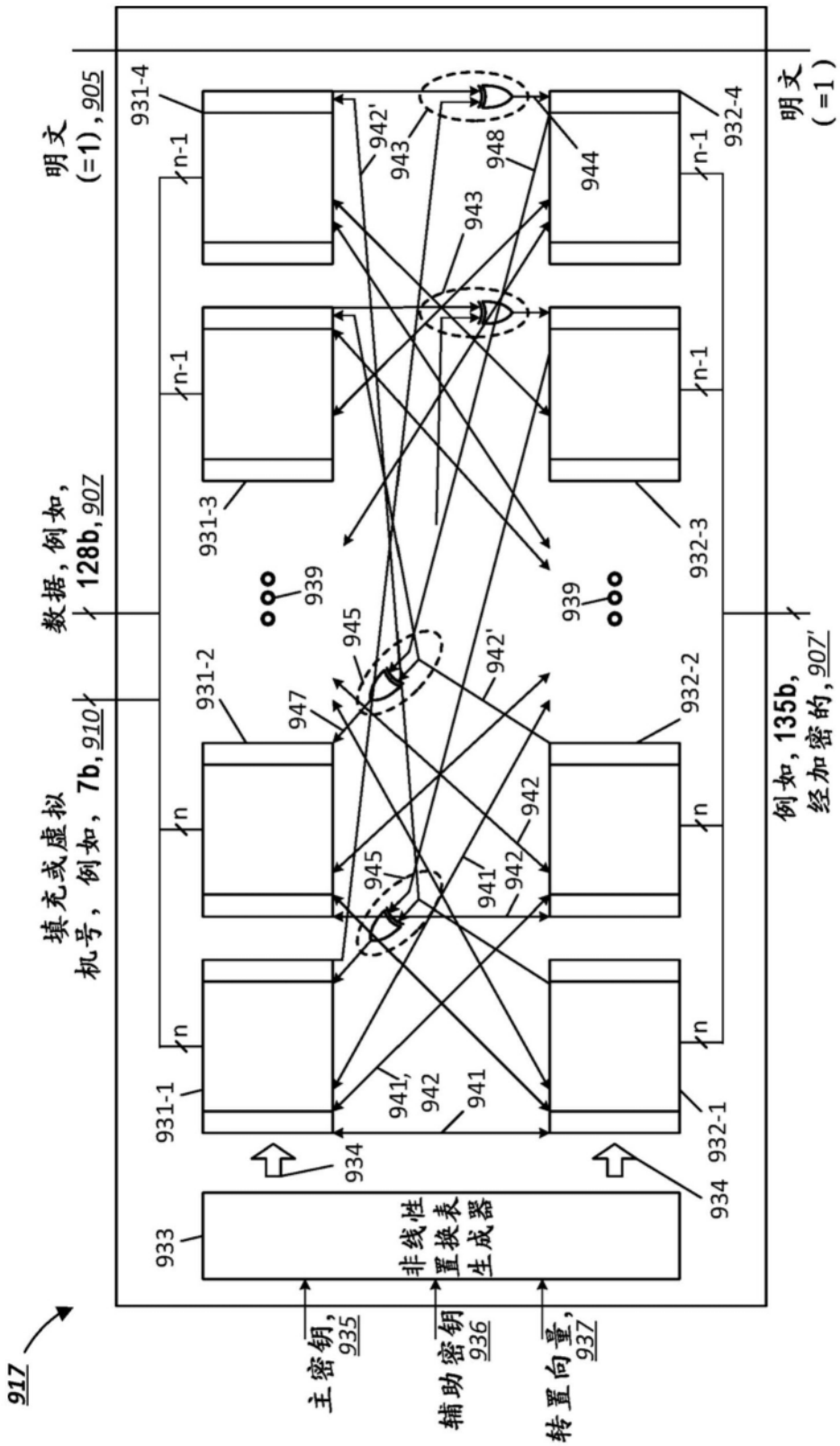

明文 (=1), 905
数据，例如，128b, 907
填充或虚拟机号，例如，7b, 910
明文 (=1)
例如，135b, 经加密的, 907'
931-4
931-3
931-2
931-1
932-4
932-3
932-2
932-1
n-1
n
942'
943
948
944
939
945
947
942
941
934
933
非线性置换表生成器
917
主密钥, 935
辅助密钥 936
转置向量, 937

图9B