[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 9/00 (2006.01)
G06F 9/445 (2006.01)

[12] 发明专利说明书

专利号 ZL 03133052.5

[45] 授权公告日 2007 年 6 月 27 日

[11] 授权公告号 CN 1323350C

[22] 申请日 2003.7.23 [21] 申请号 03133052.5

[30] 优先权

[32] 2002.7.24 [33] JP [31] 2002-215096

[73] 专利权人 松下电器产业株式会社

地址 日本大阪府

[72] 发明人 藤原睦 根本祐辅 安井纯一 前田卓治 伊藤孝幸 山田泰司 井上信治

[56] 参考文献

CN1302399A 2001.7.4
CN1307284A 2001.8.8
CN1342007A 2002.3.27
US5982887A 1999.11.9
US6418225B2 2002.7.9
CN1279861A 2001.1.10
CN131992A 1996.9.25

审查员 刘宇儒

[74] 专利代理机构 中科专利商标代理有限责任公司

代理人 汪惠民

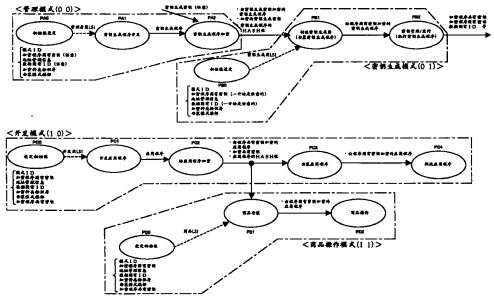
权利要求书 2 页 说明书 15 页 附图 22 页

[54] 发明名称

确保 LSI 中程序安全的方法、程序开发支持装置及其方法

[57] 摘要

本发明公开了一种程序开发方法、程序开发支援装置及程序安装方法。 对密钥安装系统提供一安全水平很高的开发环境。 用一个其结构和拥有包括机密存储器的 LSI 系统中的 LSI 一样的 LSI，将它设定在与商品操作模式不同的开发模式上，来进行该拥有包括机密存储器的 LSI 系统的程序的开发；还将它设定为管理模式而进行密钥生成程序的开发和加密；也将它设定为密钥生成模式，执行已加密的密钥生成程序而生成各种密钥。



1、一种确保 LSI 中程序安全的方法，包括如下步骤:

在所述 LSI 中设置 CPU 和机密存储器，和在预先规定的多个模式中的任何一个上设定所述 LSI 的模式定序器，

该机密存储器具有不可改写区域；

所述多个模式具有开发模式和商品操作模式，所述开发模式使所述 CPU 能够执行明文程序地设定 LSI，所述商品操作模式使 CPU 不能执行明文程序地设定 LSI，

将所述不可改写区域中的模式 ID 设定在所述模式定序器内的规定的寄存器中的步骤，

根据设定在所述规定的寄存器中的所述模式 ID，所述模式定序器把其结构和所述 LSI 一样的 LSI 作开发用 LSI，并将它设定在所述开发模式的步骤，

通过使用设定在所述开发模式中的开发用 LSI，开发所述程序的步骤。

2、根据权利要求 1 所述的确保 LSI 中程序安全的方法，其中包括:

在所述开发用 LSI 上，对在所述程序开发步骤中所开发的程序加密的加密步骤。

3、根据权利要求 1 所述的确保 LSI 中程序安全的方法，其中:

限制所述 LSI 的操作，做到: 当将它设定为开发模式时，不会生成用以将明文程序加密的密钥。

4、根据权利要求 1 所述的确保 LSI 中程序安全的方法，其中包括:

以其结构与所述 LSI 一样的 LSI 作密钥生成用 LSI，将它设定为与开发模式及商品操作模式不同的密钥生成模式的步骤；

将加密后的密钥生成程序安装到所述密钥生成用 LSI 上并通过执行该密钥生成程序而生成密钥的步骤。

5、根据权利要求 4 所述的确保 LSI 中程序安全的方法，其中:

限制所述 LSI 的操作，做到: 在将它设定为密钥生成模式的时候，

不能执行明文程序。

6、根据权利要求 4 所述的确保 LSI 中程序安全的方法，其中包括:

以其结构与所述 LSI 一样的 LSI 作管理者用 LSI，将它设定为与开发模式、商品操作模式及密钥生成模式不同的管理模式的步骤；及

在上述管理者用 LSI 上开发所述密钥生成程序并用任意密钥加密的步骤。

7、一种程序开发支持装置，它支持加密程序的开发，其中:

包括: 其结构与作为上述加密程序运作的商品的 LSI 一样的 LSI 和将明文程序存储起来的外部存储器；

所述 LSI 包括：

存储与未加密的共有密钥有关的共有密钥密钥信息的机密存储器；

在包含开发模式的多个模式中的任何一个上设定所述 LSI 的模式定序器；

通过所述模式定序器，当设定在所述开发模式时，从存储在所述机密存储器中的共有密钥密钥信息得到未加密的共有密钥的同时，用所述未加密的共有密钥将从所述外部存储器输入的明文程序加密的秘密密钥运算处理部分。

8、根据权利要求 7 所述的支持装置，其中:

所述 LSI，还包括存储引导程序的引导 ROM；且:

所述支持装置，通过执行存储在所述引导 ROM 中的引导程序，而执行从存储在所述机密存储器中的共有密钥密钥信息得到未加密的共有密钥，及用所述未加密的共有密钥对从所述外部存储器输入的明文程序加密。

9、根据权利要求 7 所述的支持装置，其中:

所述共有密钥密钥信息，包括: 用未加密的第 1 中间密钥对未加密的共有密钥加密后而得到的加密共有密钥、和用未加密的第 2 中间密钥对所述未加密的第 1 中间密钥加密后而得到的加密第 1 中间密钥；

在执行从存储在所述机密存储器中的共有密钥密钥信息得到未加密的共有密钥时，利用所述加密共有密钥、加密第 1 中间密钥及程序加密种子对所述未加密的共有密钥解密。

确保 LSI 中程序安全的方法、程序开发支持装置及其方法

技术领域

本发明涉及安装了密钥的系统、用于该系统的 LSI 的程序开发以及程序的安装这一技术。

对密钥的机密性和隐匿性很高的密钥安装系统而言，最大课题是，在如何维持它的程序开发和程序安装的安全性。

发明内容

本发明的目的，在于：对上述密钥安装系统提供安全水平很高的程序开发方法、程序开发环境及程序写入方法。

具体而言，本发明提供一种确保 LSI 中程序安全的方法，包括如下步骤：在所述 LSI 中设置 CPU 和机密存储器，和在预先决定的多个模式中的任何一个上设定所述 LSI 的模式定序器，该机密存储器具有不可改写区域；所述多个模式具有开发模式和商品操作模式，所述开发模式使所述 CPU 能够执行明文程序地设定 LSI，所述商品操作模式使 CPU 不能执行明文程序地设定 LSI，将所述不可改写区域中的模式 ID 设定在所述模式定序器内的规定的寄存器中的步骤，根据设定在所述规定的寄存器中的所述模式 ID，所述模式定序器把其结构和所述 LSI 一样的 LSI 作开发用 LSI，并将它设定在所述开发模式的步骤，通过使用设定在所述开发模式中的开发用 LSI，开发所述程序的步骤。

根据本发明，安装到具有包括机密存储器的 LSI 的系统中的程序的开发，是在其结构和该 LSI 一样且被设定为和程序安装及产品工作时的商品操作模式不同的开发模式的开发用 LSI 中进行的。换句话说，因为可将拥有含不可改写区域的机密存储器且隐匿性高的 LSI 的操作模式从安装模式转换为开发模式，而将该 LSI1 作程序开发环境用，故能使程序开发环境

的安全性比现有技术下的安全性高。

最好是，在所述本发明所涉及的程序开发方法中，限制 LSI 的操作，做到：当将它设定为开发模式时能执行明文程序，同时当将它设定为商品操作模式时则不能执行明文程序。

所述本发明所涉及的确保程序安全方法，最好是，包括：在所述开发用 LSI 上，对在所述程序开发步骤中开发的程序加密的加密步骤。

在所述本发明所涉及的确保程序安全方法中，最好是，限制 LSI 的操作，以做到：当将它设定为开发模式时，不会生成用以将明文程序加密的密钥。

所述本发明所涉及的确保程序安全方法，最好是，包括：以其结构与所述 LSI 一样的 LSI 作密钥生成用 LSI，将它设定为与开发模式及商品操作模式不同的密钥生成模式的步骤；将加密后的密钥生成程序安装到所述密钥生成用 LSI 上并通过执行该密钥生成程序而生成密钥的步骤。而且，最好是，限定所述 LSI 的操作，做到：在将它设定为密钥生成模式的时候，不能执行明文程序。或者是，最好是，包括：以其结构与所述 LSI 一样的 LSI 作管理者用 LSI，将它设定为与开发模式、商品操作模式及密钥生成模式不同的管理模式的步骤；在上述管理者用 LSI 上开发所述密钥生成程序并用任意密钥加密的步骤。

本发明提供一种支持加密程序的开发的程序开发支持装置， 一种程序开发支持装置，它支持加密程序的开发，其中:包括: 其结构与作为上述加密程序运作的商品的 LSI 一样的 LSI 和将明文程序存储起来的外部存储器；所述 LSI 包括：存储与未加密的共有密钥有关的共有密钥密钥信息的机密存储器； 在包含开发模式的多个模式中的任何一个上设定所述 LSI 的模式定序器；通过所述模式定序器，当设定在所述开发模式时，从存储在所述机密存储器中的共有密钥密钥信息得到未加密的共有密钥的同时，用所述未加密的共有密钥将从所述外部存储器输入的明文程序加密的秘密密钥运算处理部分。

根据本发明，给出了其结构与成为开发对象的加密程序运作的 LSI 的结构一样的 LSI 作开发环境。在该 LSI 上，从存储在机密存储器中的共有

密钥密钥信息得到原有的共有密钥，同时使用该原有的共有密钥对从外部存储器输入的明文程序加密。换句话说，可执行原有的共有密钥的解密和利用了该原有的共有密钥对明文程序的加密。因此，故可在原有的共有密钥不被程序开发者不得知的情况下，执行明文程序的加密。

根据本发明，通过在 LSI 上执行引导程序，从存储在机密存储器中的共有密钥密钥信息得到原有的共有密钥，同时使用该原有的共有密钥将从外部存储器输入的明文程序加密。换句话说，是通过引导程序执行原有的共有密钥的解密和利用了该原有的共有密钥的明文程序的加密，而不是利用来自外部的指示来执行原有的共有密钥的解密和利用了该原有的共有密钥的明文程序的加密。因此，故既确能防止原有的共有密钥被程序开发者得知，又能执行对明文程序的加密。

在所述本发明所涉及的程序开发支援装置中，最好是，所述共有密钥密钥信息包括：用原有的第 1 中间密钥对原有的共有密钥加密后而得到的加密共有密钥、和用原有的第 2 中间密钥将所述原有的第 1 中间密钥加密后而得到的加密第 1 中间密钥。所述第 1 步骤，最好是利用所述加密共有密钥、加密的第 1 中间密钥及程序加密种子对所述原有的共有密钥解密。

本发明提供一种将加密的程序写入到包括拥有机密存储器的 LSI 及外部存储器的密钥安装系统中的方法，该方法包括：一种将加密的程序写入到具有 LSI 及外部存储器的密钥安装系统中的方法，所述 LSI 具有存储引导程序的引导 ROM 和机密存储器，包括如下步骤：将与未加密的共有密钥有关的共有密钥密钥信息和与未加密的固有密钥有关的固有密钥密钥信息存储在所述机密存储器中的初始值设定处理，在所述 LSI 上，从存储在所述机密存储器中的共有密钥密钥信息得到未加密的共有密钥的第 1 步骤，在所述 LSI 上，用在所述第 1 步骤得到的未加密的共有密钥对从所述外部存储器施来的共有密钥加密程序解密的第 2 步骤，在所述 LSI 上，从存储在所述机密存储器中的固有密钥密钥信息得到未加密的固有密钥的第 3 步骤，在所述 LSI 上，用在所述第 3 步骤得到的未加密的固有密钥对在所述第 2 步骤得到的明文程序加密的第 4 步骤，以及仅在所述 LSI 第一次起动时，执行上述第 1～第 4 步骤，将在所述

第 4 步骤中得到的固有密钥加密程序写入到所述外部存储器中。根据本发明，用从存储在机密存储器中的共有密钥密钥信息得到的原有的共有密钥对施加给 LSI 的共有密钥加密程序解密。用从存储在机密存储器中的共有密钥密钥信息得到的原有的固有密钥对已解密的明文程序加密。换句话说，共有密钥加密程序是通过将加密的密钥从共有密钥变化为固有密钥而安装到系统中的。因此，在不同用户所持有的不同产品中，装上了由相互不同的固有密钥加密的程序，隐匿性提高。即使万一密码被破，受害产品也是有限的，故和现有技术下的安全性相比，这时的安全性提高了。

在所述本发明所涉及的程序安装方法中，最好是，LSI 包括：存储引导程序的引导 ROM，通过在所述 LSI 上执行存储在所述引导 ROM 中的引导程序来执行上述第 1～第 4 步骤。

在所述本发明所涉及的程序安装方法中，最好是，将固有密钥密钥信息存储到所述机密存储器的不可改写区域。

在所述本发明所涉及的程序安装方法中，最好是，共有密钥密钥信息包括：用原有的第 1 中间密钥对原有的共有密钥加密后而得到的加密共有密钥、和用原有的第 2 中间密钥对所述原有的第 1 中间密钥加密后而得到的加密第 1 中间密钥。所述第 1 步骤，最好是，利用所述加密共有密钥、加密第 1 中间密钥和程序加密种子对所述原有的共有密钥解密。

在所述本发明所涉及的程序安装方法中，最好是，固有密钥密钥信息包括：用原有的第 1 中间密钥对原有的固有密钥加密后而得到的加密固有密钥、和用原有的第 2 中间密钥对所述原有的第 1 中间密钥加密后而得到的加密第 1 中间密钥。所述第 3 步骤，最好是，利用所述加密固有密钥、加密的第 1 中间密钥和程序加密种子对所述原有的固有密钥解密。

在所述本发明所涉及的程序安装方法中，最好是，固有密钥密钥信息为该 LSI 所固有的固有 ID。

附图说明

图 1 为显示本发明的实施例所涉及的机密 LSI 的结构的方框图。

图 2 为显示使用了图 1 的机密 LSI 的开发及产品化的整个流程的图。

图 3 为显示引导程序的整个处理流程的流程图。

图 4 为前一个处理 SZ2 的数据流。

图 5 为密钥生成密钥的加密的数据流。

图 6 为程序加密处理 SA2 的流程图。

图 7 为程序加密处理 SA2 的数据流。

图 8 为密钥产生模式中的密钥生成器制造处理 SB1 的流程图。

图 9 和图 10 为密钥生成器制造处理 SB1 的数据流。

图 11 为密钥产生模式中密钥管理、发行处理 SB2 的流程图。

图 12 和图 13 为密钥管理、发行处理 SB2 的数据流。

图 14 为开发模式中的程序加密处理 SC1 的流程图。

图 15 为程序加密处理 SC1 的数据流。

图 16 为商品操作模式中的程序安装处理 SD1 的流程图。

图 17 和图 18 为程序安装处理 SD1 的数据流。

图 19 为商品操作模式中的通常引导处理 SD2 的流程图。

图 20 和图 21 为通常引导处理 SD2 的数据流。

图 22 为初始值设定处理 SZ1 的数据流。

具体实施方式

下面，参考附图，说明本发明的实施例。需提一下，在以下的说明中，用 Enc（X，Y）表示用密钥 Y 将 X（密钥或者程序）加密后而得到的加密密钥或者程序。

图 1 为表示本实施例所涉及的机密 LSI 的内部结构的方框图。图 1 中的结构是这样的，即机密 LSI1 可通过外部总线 120 和外部存储器（闪烁存储器）100、外部工具 110 等连接。而且，可通过加上模式 ID 来设定操作模式。

对本实施例所涉及的主要的结构要素进行简单的说明。

首先，机密 LSI1 包括: 含不可改写区域 11 的机密存储器（保密 Flash）10。该不可改写区域 11 中设有不可改写区域写入旗标 12。一旦模式 ID 写到机密存储器 10 中，不可改写区域写入旗标 12 的旗标值就会从“可写入”变成“已经写完”，之后就不能向不可改写区域中写入了。需提一下，在本实施例中，机密存储器 10 及外部存储器 100 是由闪烁存储器构成的，但并不限于此，只要是非易失性存储器就行了。

还有，秘密密钥运算处理部分 20，拥有: 存储各种密钥及程序加密种子的寄存器，由它进行加密处理。密钥生成 / 更新定序器 30，拥有模式 ID 存储寄存器 31，该密钥生成 / 更新定序器 30 根据存储在该模式 ID 存储寄存器 31 中的模式 ID，控制秘密密钥运算处理部分 20 的操作，换句话说，控制是否生成各种密钥。该密钥生成 / 更新定序器 30，拥有: 存储表示密钥或者程序由什么算法、密钥长加密的加密种类种类标识符的加密种类标识符存储寄存器 32。还拥有程序加密种子 33。

模式定序器 40 也拥有模式 ID 存储寄存器 41。该模式定序器 40 根据存储在模式 ID 存储寄存器 41 中的模式 ID 和跳线 43 的值控制外部主接口（I / F）50 的操作，换句话说，是控制通过哪一个主接口来将存储在外部存储器 100 中的程序、数据读进来。由此可控制是否可执行存储在外部存储器 100 中的明文程序。模式定序器 40 还拥有: 存储了表示用什么方法将密钥加密的加密种类标识符的加密种类标识符存储寄存器 42。

外部主接口 50，在模式定序器 40 的控制下，通过程序处理部分 51

所拥有的通过部分 52、延迟部分 53 及程序解密用加密引擎 54、数据处理部分 55 所拥有的通过部分 56 及内容加密 / 解密用加密引擎 57 中之任一个，在外部存储器 100、外部工具 110 之间进行程序、数据的输出入。

这里，除了后述的管理模式以外，通过通过部分 52 输入的程序不会在机密 LSI1 内部执行。换句话说，通过部分 52，是一在明文程序的加密、或者是用其他的密钥对已经加密的程序再次加密时有效的部分。机密 LSI1 的结构是这样的，除了后述的管理模式以外，操作不会进入通过通过部分 52 输入的程序。因此，即使例如已经成为商品的机密 LSI1 通过通过部分 52 取入了明文程序，也不能执行该明文程序。需提一下，在执行明文程序的时候，机密 LSI1 通过延迟部分 53 将程序输到它的内部。

引导 ROM60 存储控制机密 LSI1 的启动操作的引导程序。HASH 运算部分 70，为验证读到机密 LSI1 的程序的正当性而计算 HASH 值。

还有，外部存储器 100 中存储了程序、内容。外部工具 110 中存储了一开始启动机密 LSI1 时存储在机密存储器 10 中的各种初始值。该初始值的种类随着所设定的操作模式的不同而不同。

图 2 为显示使用了图 1 中的机密 LSI1 的开发及产品化的整个流程的图。如图 2 所示，机密 LSI1 在管理模式（模式 ID: 00）、密钥生成模式（模式 ID: 01）、开发模式（模式 ID: 10）及商品操作模式（模式 ID: 11）这 4 种操作模式下操作。

首先，被设定为管理模式的机密 LSI1 作为管理者用 LSI 操作。在管理者用 LSI 中，开发密钥生成程序，而且使用任意的密钥生成密钥对该密钥生成程序加密。

被设定为密钥生成模式的机密 LSI1 作为密钥生成用 LSI 操作，在密钥生成用 LSI 中，安装了在管理者用 LSI 中生成、加密的密钥生成程序。执行该密钥生成程序以后，就生成了各种密钥。

被设定为开发模式的机密 LSI1 作为开发用 LSI 操作，在开发用 LSI 中，开发在实际的产品中执行的应用程序。而且，使用程序共有密钥对该应用程序加密。

被设定为商品操作模式的机密 LSI1 作为实际的商品 LSI 操作。在商品 LSI 中，安装了在开发用 LSI 中生成的由程序共有密钥加密的应用程

序，在其内部，用程序固有密钥将所安装的应用程序变换成加密后的应用程序。需提一下，在开发用 LSI 中也可为调试应用程序来执行该变换处理。

下面，参考流程图及数据流，对每一个模式下的机密 LSI1 的操作进行详细的说明。机密 LSI1，通过执行存储在引导 ROM60 中的引导程序而进行以下操作。

图 3 为显示引导程序的整个处理过程的流程图。一给机密 LSI1 通上电以后，就由 CPU65 来执行引导 ROM60 中所存储的引导程序。如图 3 所示，首先，将每一个硬件初始化（SZ0）。然后，从外部工具 110 读入各种各样的初始值，设定在机密存储器 10 中（SZ1）。

图 22 为初始值设定处理 SZ1 的流程图。首先，在跳线 44，判断机密存储器 10 是否安装在 LSI 内。接着，判断不可改写区域写入旗标 12 是否为“已写完”，因为当为“已写完”时，初始值就已经设定在机密存储器 10 中，故让处理 SZ1 结束。当不可改写区域写入旗标 12 为“可写入”时，就将初始值写到机密存储器 10 中。不仅将模式 ID 写到机密存储器 10 的不可改写区域 11 中，还将加密的程序固有密钥、地址管理信息、数据固有密钥写到机密存储器 10 的不可改写区域 11 中。需提一下，在最开始的判断结果是，机密存储器 10 在 LSI 的外部的时候，就将模式 ID 写在表示商品操作模式的值上。这样以来，机密存储器 10 在 LSI 包外那样的欺诈产品，只可在商品操作模式下工作。

接着，将不可改写区域写入旗标 12 设定为“已写完”。这样以来，以后的不可改写区域 11 就不能改写了。而且，还将加密种类标识符及安装模式旗标写到通常区域 13、14 中。而且，当模式 ID 显示管理模式以外的模式的时候，除了将加密种类标识符及安装模式旗标写到通常区域 13、14 中以外，还将已加密的共有密钥 / 密钥生成密钥写到通常区域 13、14 中。

之后，执行前处理 SZ2。图 4 为前处理 SZ2 的数据流。这里，设定在机密存储器 10 的不可改写区域 11 中的模式 ID，被设定在密钥生成 / 更新定序器 30 的模式 ID 存储寄存器 31 及模式定序器 40 的模式 ID 存储寄存器 41 中；设定在机密存储器 10 的第 1 通常区域 13 中的加密种类标识

符被设定在密钥生成／更新定序器 30 的加密种类标识符存储寄存器 32 及模式定序器 40 的加密种类标识符存储寄存器 42 中；机密存储器 10 的不可改写区域 11 中所存储的地址管理信息被设定在 MEMC80 的密码地址区分存储寄存器 81 中。到这里为止的操作，和图 2 中的初始值设定阶段 PA0、PB0、PC0、PD0 相对应。

之后，根据模式 ID 的值来进行每一个模式下的操作（SZ3）。

（管理模式）

当模式 ID 为“00”时，机密 LSI1 成为管理模式，根据跳线 43 的值（SA0）执行明文程序执行处理 SA1 或者是程序加密处理 SA2。

在密钥生成程序开发阶段 PA1，进行明文程序执行处理 SA1，在这里生成密钥生成程序。该密钥生成程序存储在外部存储器 100 中。

在密钥生成程序加密阶段 PA2，首先，如图 5 的数据流所示，通过执行密钥生成程序而对施来的任意密钥生成密钥加密。换句话说，在外部主接口 50 中，程序处理部分 51 的通过部分 52 在模式定序器 40 的作用下有效。外部存储器 100 中所存储的密钥生成程序通过通过部分 52 加到 CPU65 中而得以执行。执行了该密钥生成程序以后，外部存储器 100 中所存储的密钥生成密钥就由秘密密钥运算处理部分 20 利用装在密钥生成／更新定序器 30 中的程序加密种子加密。

需提一下，在本实施例中，密钥的加密是利用第 1 中间密钥和第 2 中间密钥来进行的。换句话说，加密的结果是，得到了明文密钥（这里为密钥生成密钥）由第 1 中间密钥（这里为 MK1）加密后而得到的加密密钥（这里为 Enc（密钥生成密钥，MK1））、和由第 2 中间密钥（这里为 CK）将第 1 中间密钥加密后而得到的加密第 1 中间密钥（这里为 Enc（MK1，CK）。当然，本发明并不限于这样的密钥加密方法。

之后，执行程序加密处理 SA2。图 6 为该程序加密处理 SA2 的流程图，图 7 为数据流。首先，通过外部主接口 50 的通过部分 52 将存储在外部存储器 100 中的已加密的密钥生成密钥 Enc（密钥生成密钥，MK1）、Enc（MK1，CK）设定在秘密密钥运算处理部分 20 中（SA21）。再用安装在密钥生成／更新定序器 30 中的程序加密种子将该加密后的密钥生成密钥解密，得到密钥生成密钥（SA22）。之后，取入存储在外部存储器 100

中的明文的密钥生成程序，再用已在 SA22 解密的密钥生成密钥对明文的密钥生成程序加密，写到外部存储器 100 中（SA23）。由 HASH 运算部分 70 对外部存储器 100 中的明文的密钥生成程序进行 HASH 运算，并将计算出的 HASH 值写到外部存储器 100 中（SA24）。

经过了这样的操作以后，在管理模式，生成由密钥生成密钥加密的密钥生成程序 Enc（密钥生成程序，密钥生成密钥）、已加密的密钥生成密钥 Enc（密钥生成密钥，MK1）、Enc（MK1，CK）、密钥生成程序的 HASH 值。

－密钥生成模式－

当模式 ID 为“01”时，机密 LSI1 成为密钥生成模式，根据安装模式旗标的值（SB0）来执行密钥生成器制造处理 SB1 或者是密钥管理／发行处理 SB2。

在密钥生成器制造阶段 PB1，执行密钥生成器制造处理 SB1，图 8 为该处理 SB1 的流程图，图 9 及图 10 为数据流。这里，根据模式 ID 和安装模式旗标的值，在外部主接口 50 所拥有的程序处理部分 51 中通过部分 52 被设定为有效。

首先，将存储在机密 LSI1 的不可改写区域 11 中加密的程序固有密钥 Enc（程序固有密钥，MK0）、Enc（MK0，CK）设定在秘密密钥运算处理部分 20 的加密密钥存储寄存器中（SB11）。再用安装在密钥生成／更新定序器 30 中的程序加密种子对已加密的程序固有密钥解密，得到程序固有密钥（SB12）。接着，将在初始值设定阶段 PB0 所设定的、被加密的密钥生成密钥 Enc（密钥生成密钥，MK1）、Enc（MK1，CK）设定在秘密密钥运算处理部分 20 的加密密钥存储寄存器中（SB13），用安装在密钥生成／更新定序器 30 中的程序加密种子对该加密的密钥生成密钥解密，得到密钥生成密钥（SB14）。

之后，通过外部主接口 50 所拥有的程序处理部分 51 的通过部分 52，将存储在外部存储器 100 中由密钥生成密钥加密的密钥生成程序 Enc（密钥生成程序，密钥生成密钥）取到秘密密钥运算处理部分 20 中（SB15）。接着，用密钥生成密钥将所取入的加密密钥生成程序解密以后，再用程序固有密钥加密，得到加密的密钥生成程序 Enc（密钥生成程序，程序

固有密钥）（SB16）。写到外部存储器 100 中（SB17）。接着，再通过通过部分 52 将存储在外部存储器 100 中的 HASH 值设定在机密存储器 10 的通常区域 13 中（SB18）。

由 CPU65 将存储在机密存储器 10 的通常区域 13 中的安装模式旗标的值设定为“OFF”（SB19）。接着，消除存储在机密存储器 10 中的通常区域 13 中加密的密钥生成密钥 Enc（密钥生成密钥，MK1）、Enc（MK1，CK）（SB1A），同时消除存储在外部存储器 100 中加密的密钥生成程序 Enc（密钥生成密钥程序，密钥生成密钥）及 HASH 值（SB1B）。

在密钥管理 / 发行阶段 PB2，执行密钥管理 / 发行程序 SB2。图 11 为该处理 SB2 的流程图，图 12 及图 13 为数据流。这里，根据模式 ID 和安装模式旗标的值，将外部主接口 50 所拥有的程序解密用加密引擎 54 设定为有效。

首先，将存储在机密存储器 10 的不可改写区域 11 中且加密了的程序固有密钥 Enc（程序固有密钥，MK0）、Enc（MK0，CK）设定在秘密密钥运算处理部分 20 的加密密钥存储寄存器中（SB21）。接着，用安装在密钥生成 / 更新定序器 30 中的程序加密种子对已加密了的程序固有密钥解密，得到程序固有密钥（SB22）。所得到的程序固有密钥被设定在外部主接口 50 的程序解密用加密引擎 54 的程序固有密钥存储寄存器中（SB23）。

之后，通过外部主接口 50 所拥有的程序处理部分 51 的程序解密用加密引擎 54，对存储在外部存储器 100 中由程序固有密钥加密了的密钥生成程序 Enc（密钥生成程序，程序固有密钥）解密并将它取到 HASH 运算部分 70 中，计算 HASH 值（SB24）。接着，对计算出的 HASH 值和存储在机密存储器 10 的通常区域 13 中的 HASH 值进行比较，检查密钥生成程序是否被窜改了（SB25）。当 HASH 值一致时（在 SB26 为 No），处理将移到存储在外部存储器 100 的密钥生成程序 Enc（密钥生成程序，程序固有密钥）中，执行密钥的生成（SB27）。另一方面，当 HASH 值不一致时（在 SB26 为 Yes），就推测是有欺骗行为，而执行欺骗访问控制处理（SB28）。

在商品操作模式下，仅使通过部分 52 有效而输入程序，或者使程序

解密用加密引擎 54 有效而将已加密的程序解密并将它输入，故机密 LSI1 的操作受到限制，结果是不能执行明文程序。

－开发模式－

当模式 ID 为“10”时，机密 LSI1 成为开发模式，根据跳线 43 的值（SC0）来执行程序加密处理 SC1、明文程序执行处理 SC2、程序安装处理 SC3 或者是加密程序执行处理 SC4。

在应用程序开发阶段 PC1，设延迟部分 53 有效，执行明文程序执行处理 SC2，开发出应用程序。所开发的应用程序存储在外部存储器 100 中。

在应用程序加密阶段 PC2，执行程序加密处理 SC1。图 14 为该程序加密处理 SC1 的流程图，图 15 为数据流。首先，将存储在机密存储器 10 的通常区域 14 中的作为共有密钥密钥信息的加密的程序共有密钥 Enc(程序共有密钥，MK2)、Enc（MK2，CK）设定在秘密密钥运算处理部分 20 中（SC11）。接着，用安装在密钥生成 / 更新定序器 30 中的程序加密种子对已加密的程序共有密钥解密，得到程序共有密钥（SC12）。之后，将存储在外部存储器 100 中的明文应用程序取进来，用在 SC12 解密的程序共有密钥对它加密，并写到外部存储器 100 中(SC13)。接着，再由 HASH 运算部分 70 对外部存储器 100 的明文应用程序进行 HASH 运算，将计算出的 HASH 值写入外部存储器 100 中（SC14）。

经过了这样的操作以后，生成了由程序共有密钥加密的应用程序 Enc（应用程序，程序共有密钥）、应用程序的 HASH 值。

其次，在应用程序安装阶段 PC3，执行程序安装处理 SC3；在应用程序调试阶段 PC4，执行加密程序执行处理 SC4。因为这些处理和商品操作模式中的各个处理 SD1，SD2 一样，故详情省略不述。

就这样，因为可将拥有含不可改写区域 11 的机密存储器 10 且具有高隐匿性的 LSI1 的操作模式从安装模式转换为开发模式，而让该 LSI1 作程序开发环境用，故能使程序开发环境中的安全性比现在的安全性高。

因原有的共有密钥由存储在机密存储器 10 中的作为共有密钥密钥信息的已加密的共有密钥（共有密钥信息）解密，并用该原有的共有密钥对明文程序加密，故可在程序开发者不知道原有的共有密钥的情况下，

对明文程序加密。

因为原有的共有密钥的解密、利用了该原有的共有密钥的明文程序的加密，是通过引导程序执行的，而不是通过接收来自外部的指示而执行的，故既确可防止程序开发者知道原有的共有密钥，又能执行明文程序的加密。

－商品操作模式－

当模式 ID 为“11”时，机密 LSI1 成为商品操作模式，根据安装模式旗标的值（SD0）来执行程序安装处理 SD1 或者通常引导处理 SD2。

在商品安装阶段 PD1，执行程序安装处理 SD1。图 16 为该处理 SD1 的流程图，图 17 及图 18 为数据流。这里，根据模式 ID 和安装模式旗标的值，在外部主接口 50 所拥有的程序处理部分 51 中通过部分 52 被设定为有效。

首先，将存储在机密存储器 10 的不可改写区域 11 中作为固有密钥密钥信息的已加密的程序固有密钥（程序固有密钥，MK0）、Enc（MK0，CK）设定在秘密密钥运算处理部分 20 的加密密钥存储寄存器中（SD11）。接着，用安装在密钥生成 / 更新定序器 30 中的程序加密种子对已加密的程序固有密钥解密，得到程序固有密钥（SD12）。接着，将在初始值设定阶段 PD0 所设定的作为共有密钥密钥信息的已加密的程序共有密钥 Enc（程序共有密钥，MK2）、Enc（MK2，CK）设定在秘密密钥运算处理部分 20 的加密密钥存储寄存器中（SD13），再使用安装在密钥生成 / 更新定序器 30 中的程序加密种子对该已加密的程序共有密钥解密，而得到程序共有密钥（SD14）。

之后，通过外部主接口 50 所拥有的程序处理部分 51 的通过部分 52，将存储在外部存储器 100 中且由程序共有密钥加密的应用程序 Enc(应用程序，程序共有密钥)取到秘密密钥运算处理部分 20 中（SD15）。接着，用程序共有密钥将所取入的加密了的应用程序解密以后，再用程序固有密钥对它加密，得到加密后的应用程序 Enc（应用程序，程序固有密钥）（SD16），并写到外部存储器 100 中（SD17）。接着，再通过通过部分 52 将存储在外部存储器 100 中的 HASH 值设定在机密存储器 10 的通常区域 13 中（SD18）。

由 CPU65 将存储在机密存储器 10 的通常区域 13 中的安装模式旗标的值设定为“OFF”（SD19）。消除存储在机密存储器 10 的通常区域 13 中且已加密的程序共有密钥 Enc（程序共有密钥，MK1）、Enc（MK1，CK）（SD1A），同时消除存储在外部存储器 100 中且已加密的应用程序 Enc（应用程序，程序共有密钥）及 HASH 值（SD1B）。

换句话说，共有密钥加密程序，是通过让加密的密钥从共有密钥变换为固有密钥而安装到系统上的。因此，用户所持有的每一个产品就安装了分别由不同的固有密钥加密后而得到的程序，隐匿性提高。万一密码子遭到破坏，受害产品的数量也是有一定限度的，故和现有的技术相比，安全性得到了提高。

需提一下，可以以固有 ID 为本生成固有密钥。换句话说，对每一个机密 LSI1 而言，在它的机密存储器 10 中安装一个自己的固有 ID 作固有密钥密钥信息，在该产品安装阶段 PD1，由引导程序从所安装的固有 ID 生成固有密钥。

在商品操作阶段 PD2，执行通常引导处理 SD2。图 19 为该处理 SD2 的流程图；图 20 及图 21 为数据流。这里，根据模式 ID 及安装模式旗标的值将外部主接口 50 所拥有的程序解密用加密引擎 54 设定为有效。

首先，将存储在机密存储器 10 的不可改写区域 11 中加密了的程序固有密钥 Enc（程序固有密钥，MK0）、Enc（MK0，CK）设定在秘密密钥运算处理部分 20 的加密密钥存储寄存器中（SD21）。用安装在密钥生成 / 更新定序器 30 中的程序加密种子对该已加密的程序固有密钥解密，得到程序固有密钥（SD22）。所得到的程序固有密钥设定在外部主接口 50 的程序解密用加密引擎 54 的程序固有密钥存储寄存器中（SD23）。

之后，将存储在机密存储器 10 的不可改写区域 11 中的数据固有 ID 设定在秘密密钥运算处理部分 20 的固有 ID 存储寄存器中（SD24）。由 CPU65 产生随机数，并将它设定在秘密密钥运算处理部分 20 的随机数存储寄存器中（SD25）。由秘密密钥运算处理部分 20 从数据固有 ID 和随机数生成数据固有密钥（SD26）。

之后，通过外部主接口 50 所拥有的程序处理部分 51 的程序解密用加密引擎 54，对存储在外部存储器 100 中且由程序固有密钥加密的应用程

序 Enc（应用程序，程序固有密钥）解密并将它取到 HASH 运算部分 70 中，计算 HASH 值（SD27）。接着，对该计算出的 HASH 值和存储在机密存储器 10 的通常区域 13 中的 HASH 值进行比较，检查应用程序是否被窜改（SD28）。当 HASH 值一致时（在 SD29 为 No），处理将移到存储在外部存储器 100 的应用程序 Enc（应用程序，程序固有密钥），执行应用（SD2A）。另一方面，当 HASH 值不一致时（在 SD29 为 Yes），就推测是有不正行为，而执行不正访问控制处理（SD2B）。

在商品操作模式下，仅使通过部分 52 有效输入程序，或者仅使程序解密用加密引擎 54 有效而将已加密的程序解密并将它输入，故机密 LSI1 的操作受到了限制，结果是不能执行明文程序。

需提一下，在开发模式和商品操作模式下，即使从外部利用秘密密钥运算处理部分 20 执行生成密钥这样的处理，也能由密钥生成 / 更新定序器 30 判断出这一情况而不得执行产生密钥这样的处理。换句话说，密钥生成 / 更新定序器 30，在开发模式及商品操作模式下，操作受限而只在启动时使用程序加密种子，除此以外的其他时候都不能使用程序加密种子。故不能执行生成密钥的处理。

需提一下，在本实施例中，程序、数据存储在外部存储器 100 中，安装在机密存储器 10 中的初始值存储在外部工具 110 中，将程序、数据及初始值安装在哪里都行。例如，可从外部工具 110 中读入程序、数据，再对它加密，是没有任何问题的。

需提一下，在本实施例中。由引导程序执行各种处理，本发明并不限于此，由其他手段来执行处理的一部分或者全部也是可以的。只不过是，由引导程序来执行处理时的安全性会比由来自外部的指示执行处理的安全性更高。

综上所述，根据本发明，因为可将拥有含不可改写区域的机密存储器且隐匿性高的 LSI 的操作模式从安装模式转换为开发模式，而让该 LSI 作程序开发环境用，故能使程序开发环境中的安全性比现在的安全性高。

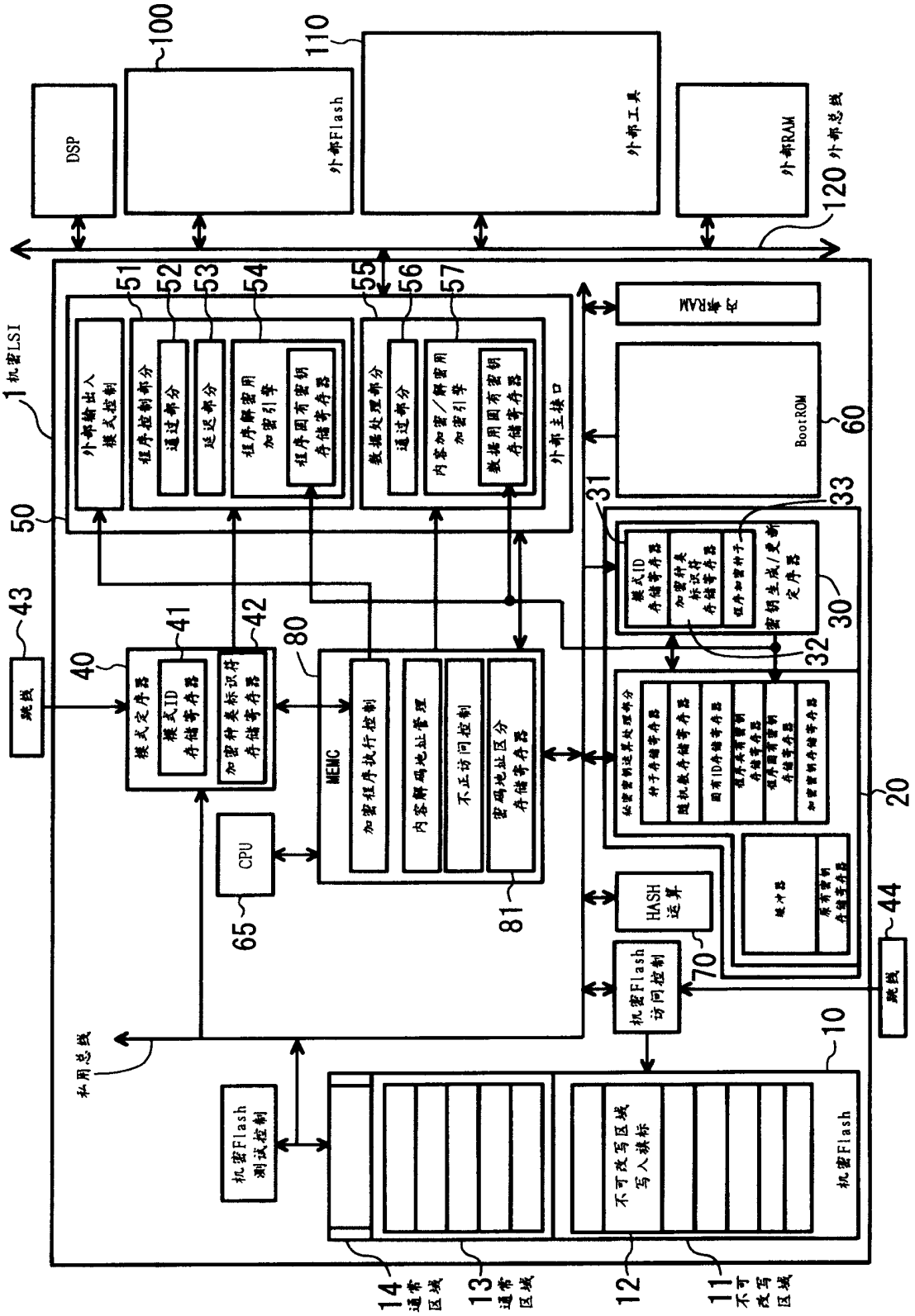

DSP
外部Flash
外部工具
外部RAM
外部总线
机密LSI
外部输出入模式控制
程序控制部分
通过部分
延迟部分
程序解密用加密引擎
程序固有密钥存储寄存器
数据处理部分
通过部分
内容加密/解密用加密引擎
数据用固有密钥存储寄存器
外部主接口
内部RAM
BootROM
模式ID存储寄存器
加密种类标识存储寄存器
程序加密种子
密钥生成/更新定序器
跳线
模式定序器
模式ID存储寄存器
加密种类标识存储寄存器
MEMC
加密程序执行控制
内容解码地址管理
不正访问控制
密码地址区分存储寄存器
CPU
HASH运算
机密Flash访问控制
私用总线
机密Flash测试控制
不可改写区域写入缺省
机密Flash
通常区域
通常区域
不可改写区域

图 1

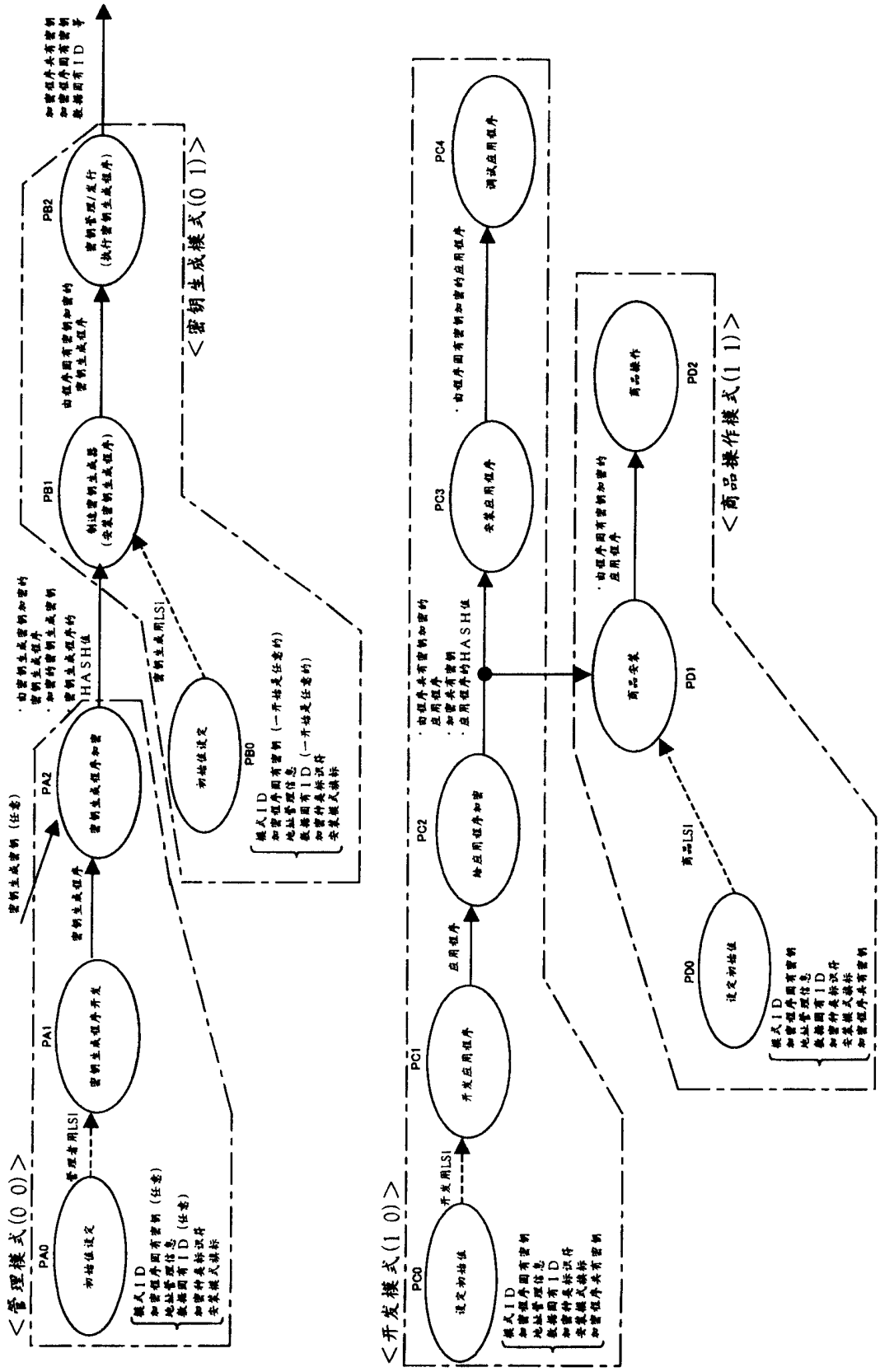

<管理模式(0 0)>
<密钥生成模式(0 1)>
<开发模式(1 0)>
<商品操作模式(1 1)>
PA0
PA1
PA2
PB0
PB1
PB2
PC0
PC1
PC2
PC3
PC4
PD0
PD1
PD2
初始值设定
密钥生成程序开发
密钥生成程序加密
密钥生成/发行
(执行密钥生成程序)
开发应用程序
地应用程序加密
安装应用程序
调试应用程序
商品安装
商品操作
开发用LSI
商品LSI
密钥生成用LSI

图 2

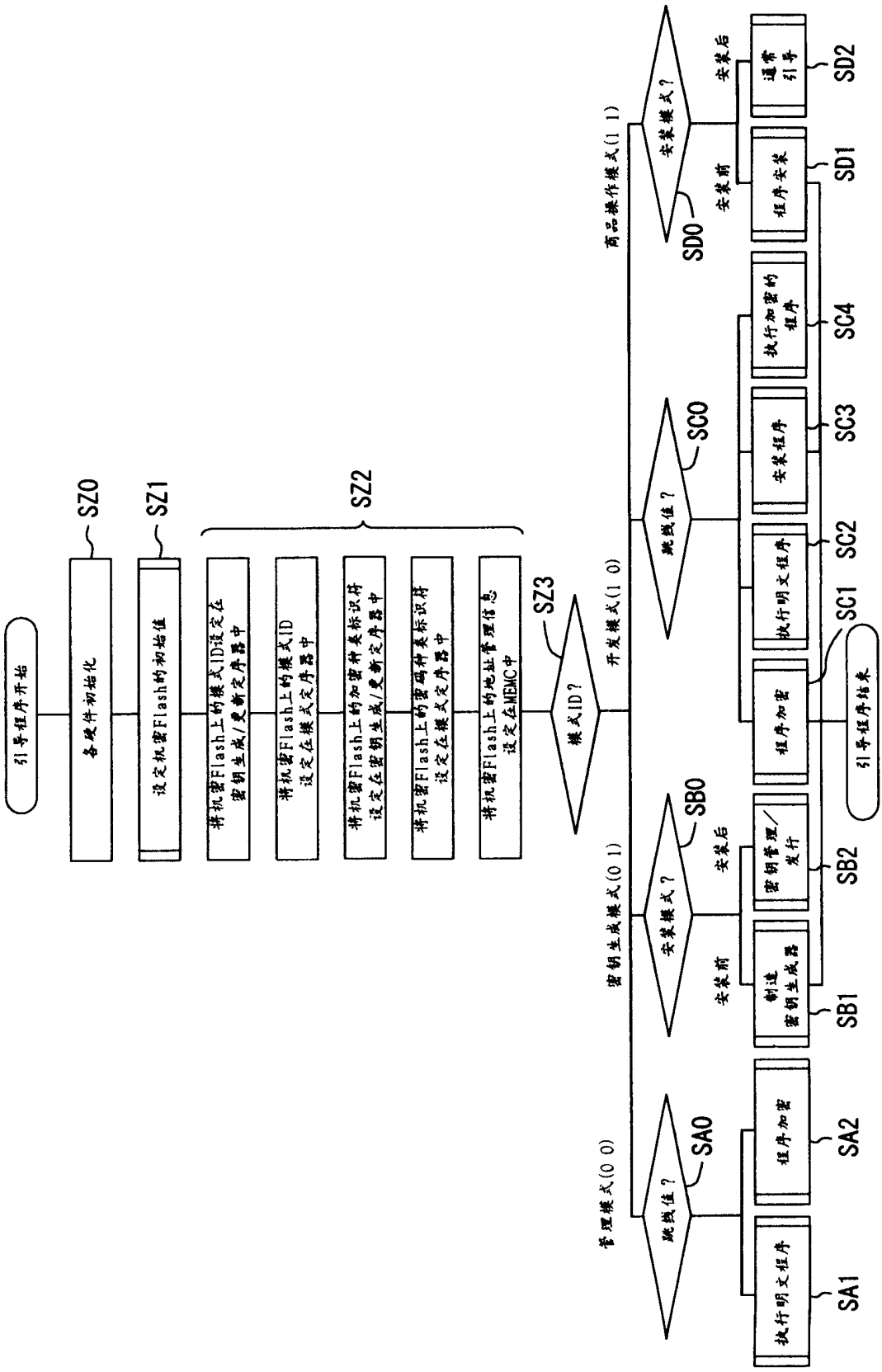

引导程序开始
各硬件初始化
SZ0
设定机密Flash的初始值
SZ1
将机密Flash上的模式ID设定在密钥生成/更新定序器中
将机密Flash上的模式ID设定在模式定序器中
将机密Flash上的加密种类标识符设定在密钥生成/更新定序器中
将机密Flash上的密码种类标识符设定在模式定序器中
将机密Flash上的地址管理信息设定在MEMC中
SZ2
SZ3
模式ID?
开发模式(1 0)
管理模式(0 0)
密钥生成模式(0 1)
商品操作模式(1 1)
跳线值?
SA0
执行明文程序
SA1
程序加密
SA2
安装前
制造密钥生成器
SB1
安装后
密钥管理/发行
SB2
安装模式?
SB0
程序加密
SC1
执行明文程序
SC2
安装程序
SC3
执行加密的程序
SC4
跳线值?
SC0
安装前
程序安装
SD1
安装后
通常引导
SD2
安装模式?
SD0
引导程序结束

图 3

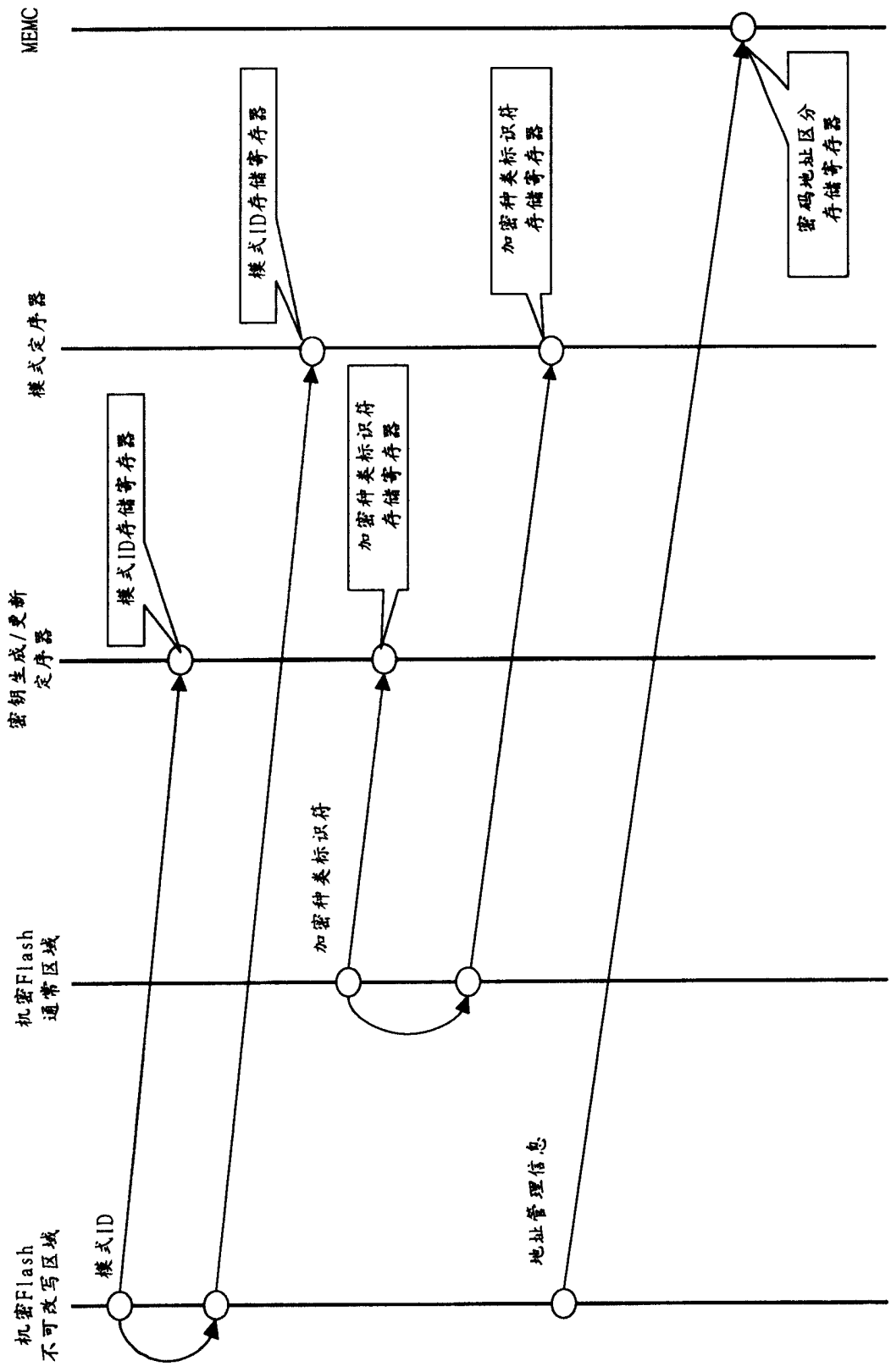

MEMC
模式定序器
密钥生成/更新
定序器
机密Flash
通常区域
机密Flash
不可改写区域
模式ID存储寄存器
加密种类标识符
存储寄存器
密码地址区分
存储寄存器
模式ID
加密种类标识符
地址管理信息

图 4

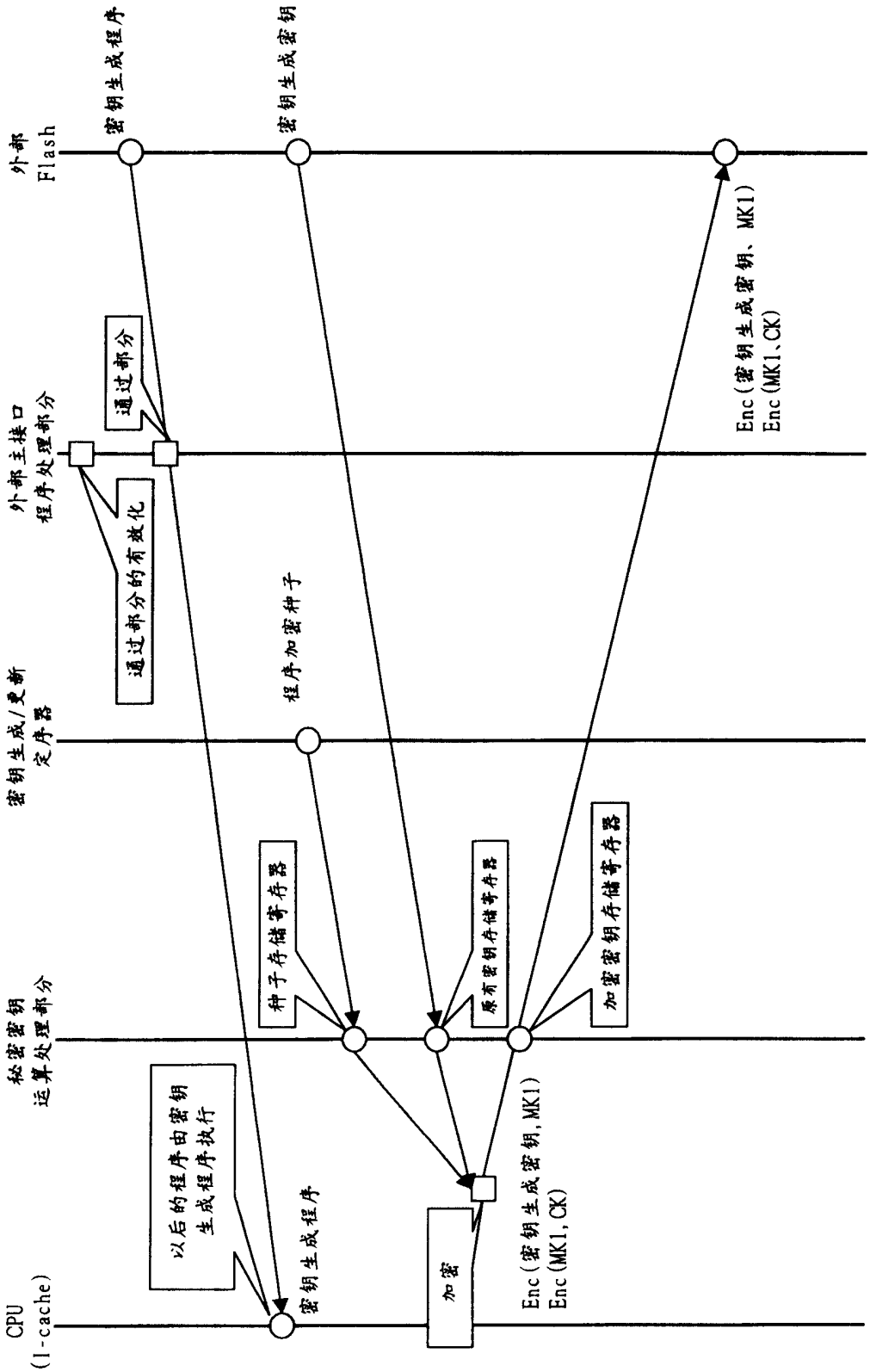

外部
Flash
密钥生成程序
密钥生成密钥
外部主接口
程序处理部分
通过部分
通过部分的有效化
Enc(密钥生成密钥、MK1)
Enc(MK1、CK)
密钥生成/更新
定序器
程序加密种子
秘密密钥
运算处理部分
种子存储寄存器
原有密钥存储寄存器
加密密钥存储寄存器
以后的程序由密钥
生成程序执行
CPU
(I-cache)
密钥生成程序
加密
Enc(密钥生成密钥,MK1)
Enc(MK1,CK)

图 5

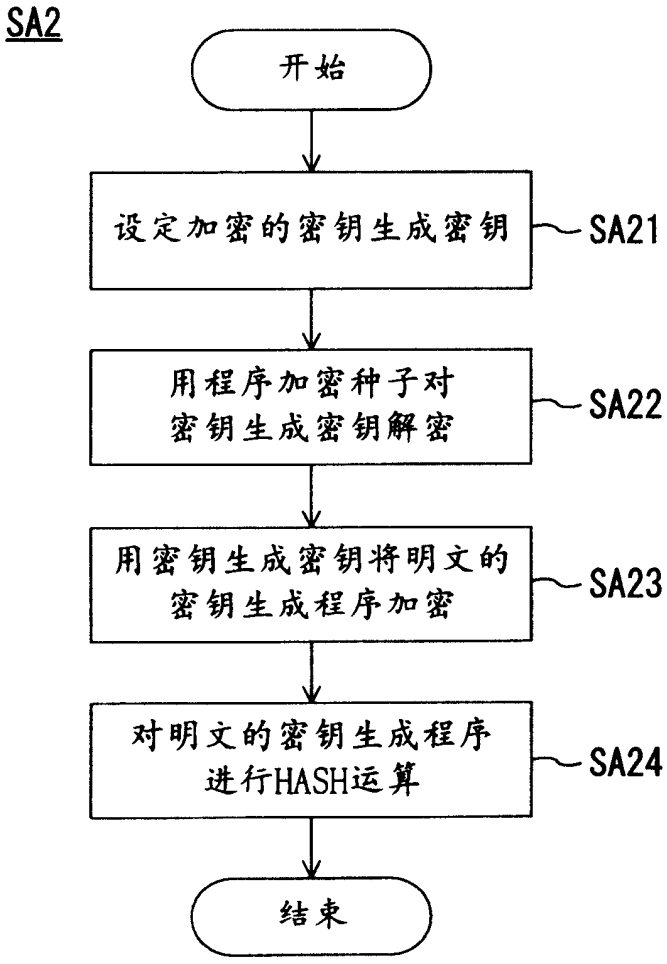

SA2
开始
设定加密的密钥生成密钥
SA21
用程序加密种子对
密钥生成密钥解密
SA22
用密钥生成密钥将明文的
密钥生成程序加密
SA23
对明文的密钥生成程序
进行HASH运算
SA24
结束

图 6

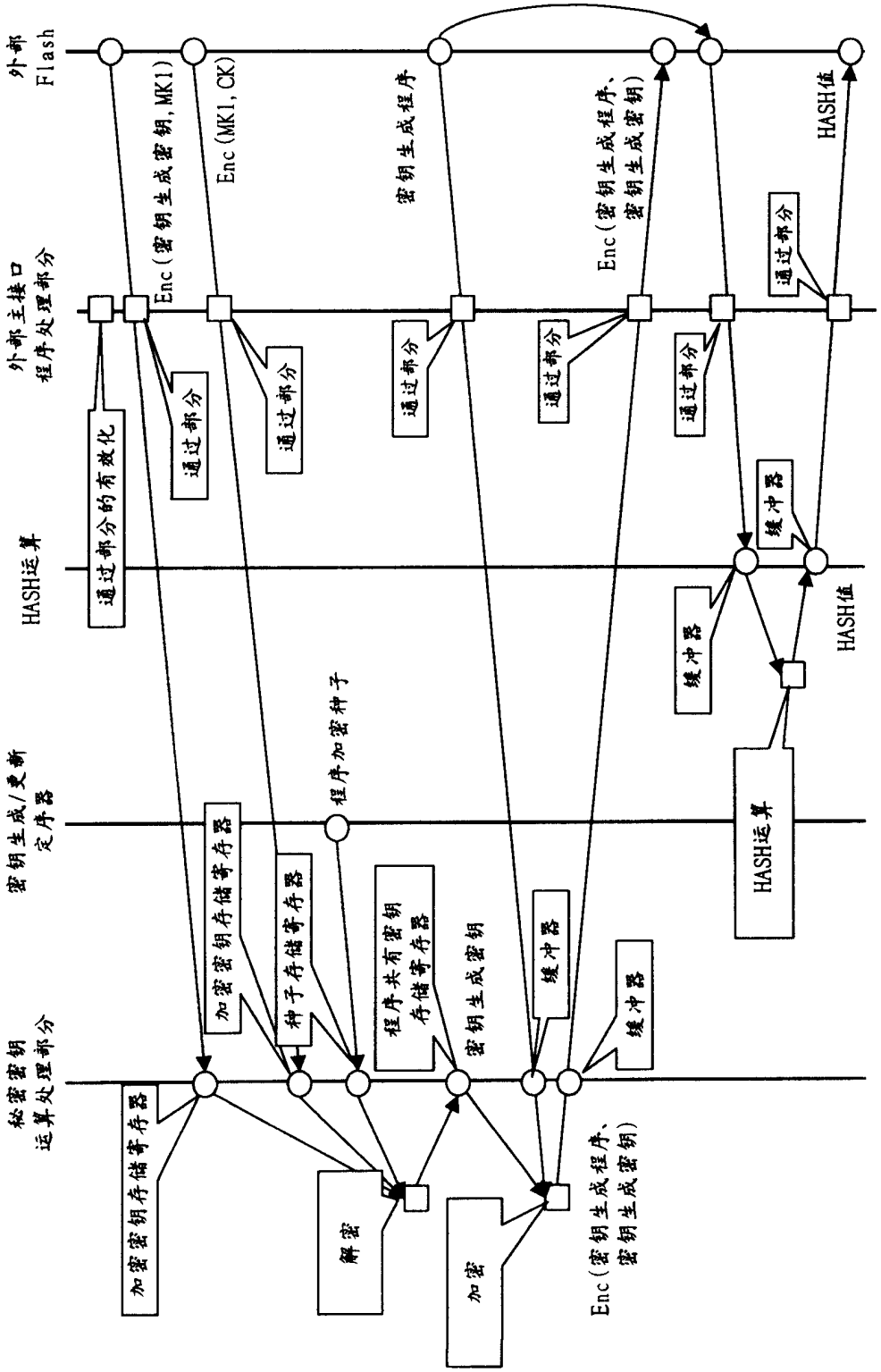

外部 Flash
外部主接口 程序处理部分
HASH运算
密钥生成/更新 定序器
秘密密钥 运算处理部分
Enc(密钥生成密钥,MK1)
Enc(MK1,CK)
密钥生成程序
Enc(密钥生成程序、密钥生成密钥)
HASH值
通过部分的有效化
通过部分
通过部分
通过部分
通过部分
通过部分
缓冲器
缓冲器
HASH值
加密密钥存储寄存器
种子存储寄存器
程序加密种子
程序共有密钥 存储寄存器
密钥生成密钥
缓冲器
缓冲器
HASH运算
加密密钥存储寄存器
解密
加密
Enc(密钥生成程序、密钥生成密钥)

图 7

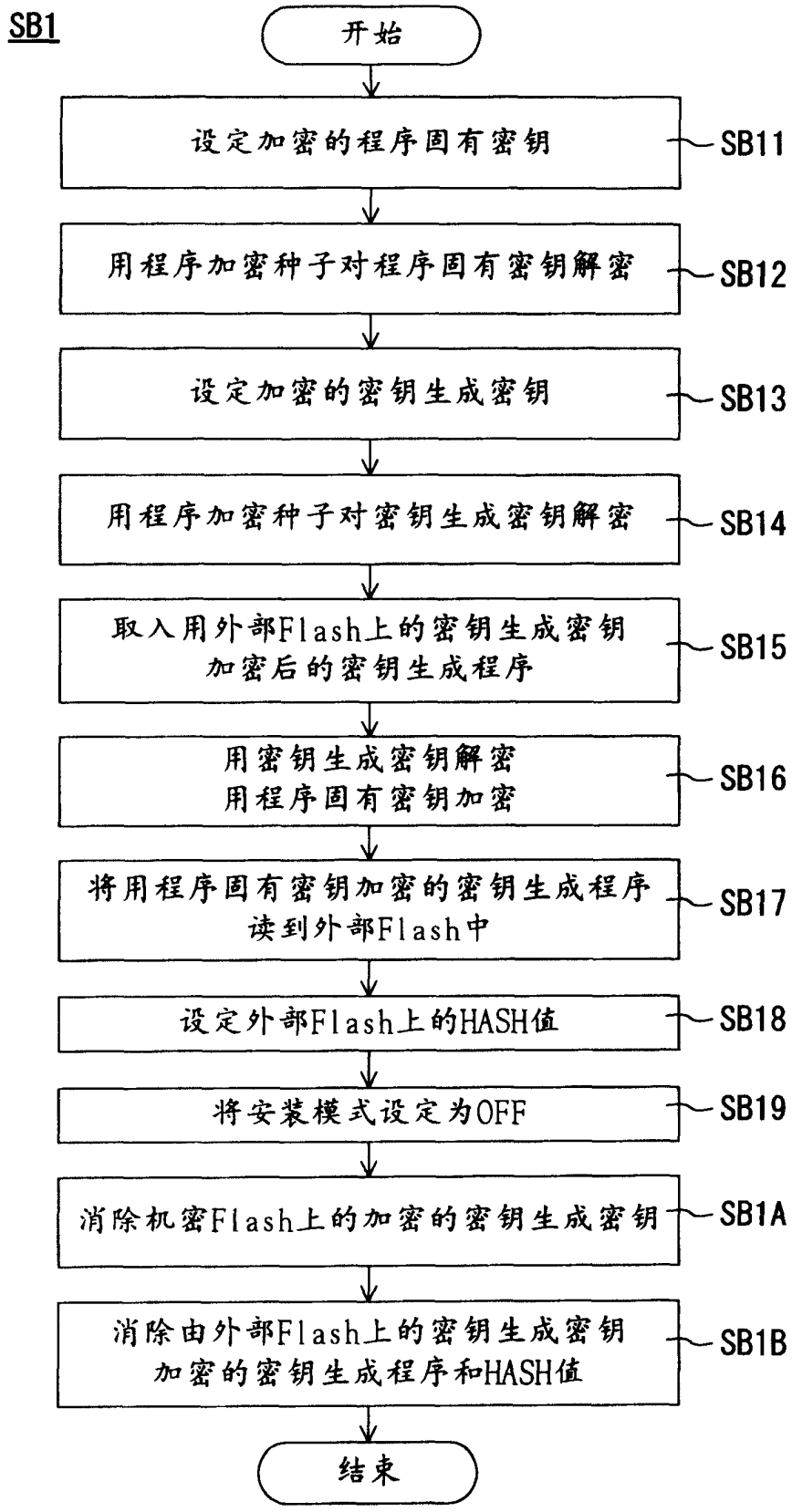

SB1
开始
设定加密的程序固有密钥
SB11
用程序加密种子对程序固有密钥解密
SB12
设定加密的密钥生成密钥
SB13
用程序加密种子对密钥生成密钥解密
SB14
取入用外部Flash上的密钥生成密钥
加密后的密钥生成程序
SB15
用密钥生成密钥解密
用程序固有密钥加密
SB16
将用程序固有密钥加密的密钥生成程序
读到外部Flash中
SB17
设定外部Flash上的HASH值
SB18
将安装模式设定为OFF
SB19
消除机密Flash上的加密的密钥生成密钥
SB1A
消除由外部Flash上的密钥生成密钥
加密的密钥生成程序和HASH值
SB1B
结束

图 8

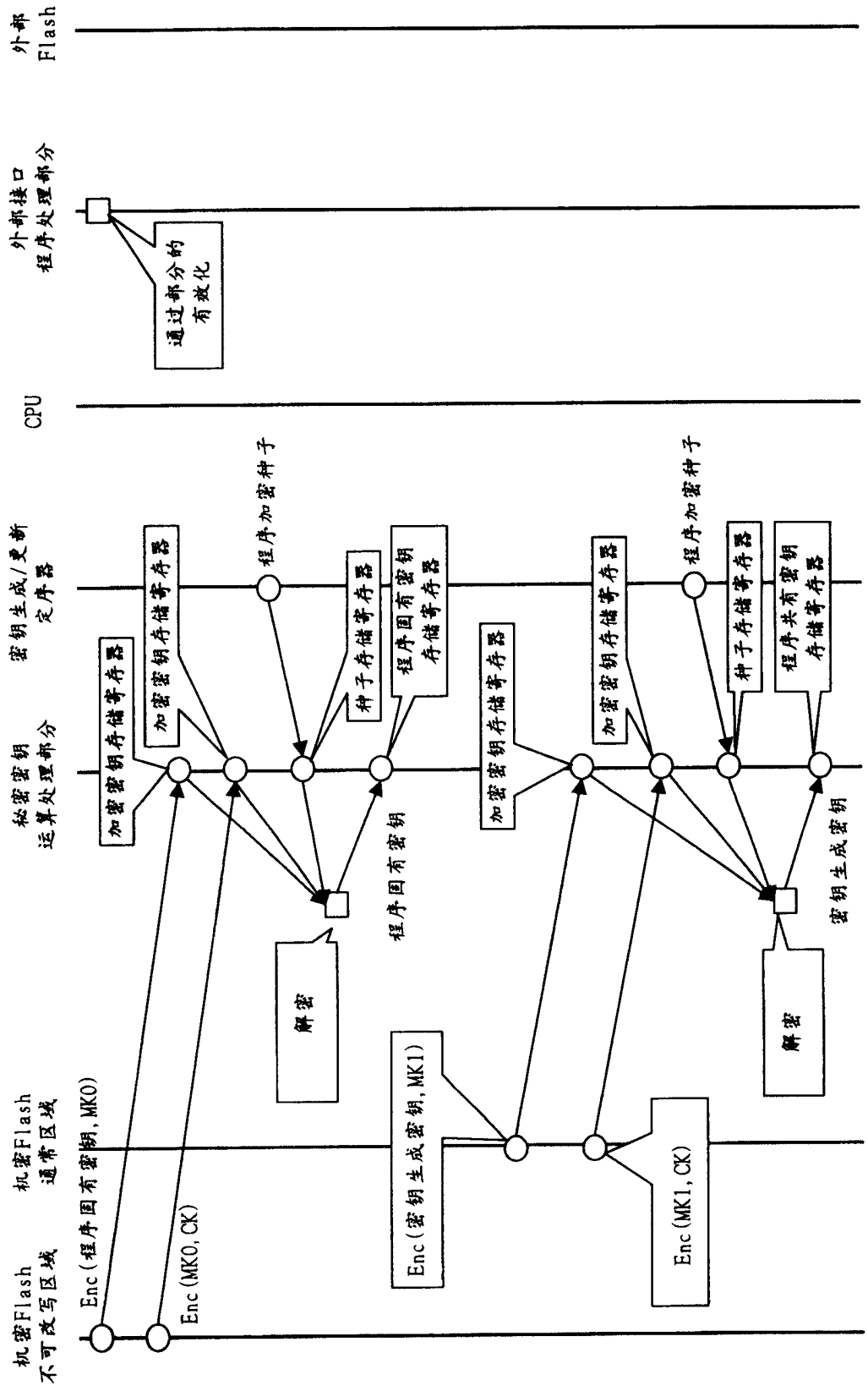

外部Flash
外部接口程序处理部分
CPU
密钥生成/更新定序器
秘密密钥运算处理部分
机密Flash通常区域
机密Flash不可改写区域
通过部分的有效化
加密密钥存储寄存器
加密密钥存储寄存器
程序加密种子
种子存储寄存器
程序固有密钥存储寄存器
加密密钥存储寄存器
加密密钥存储寄存器
程序加密种子
种子存储寄存器
程序共有密钥存储寄存器
Enc(程序固有密钥,MK0)
Enc(MK0,CK)
解密
程序固有密钥
Enc(密钥生成密钥,MK1)
Enc(MK1,CK)
解密
密钥生成密钥

图 9

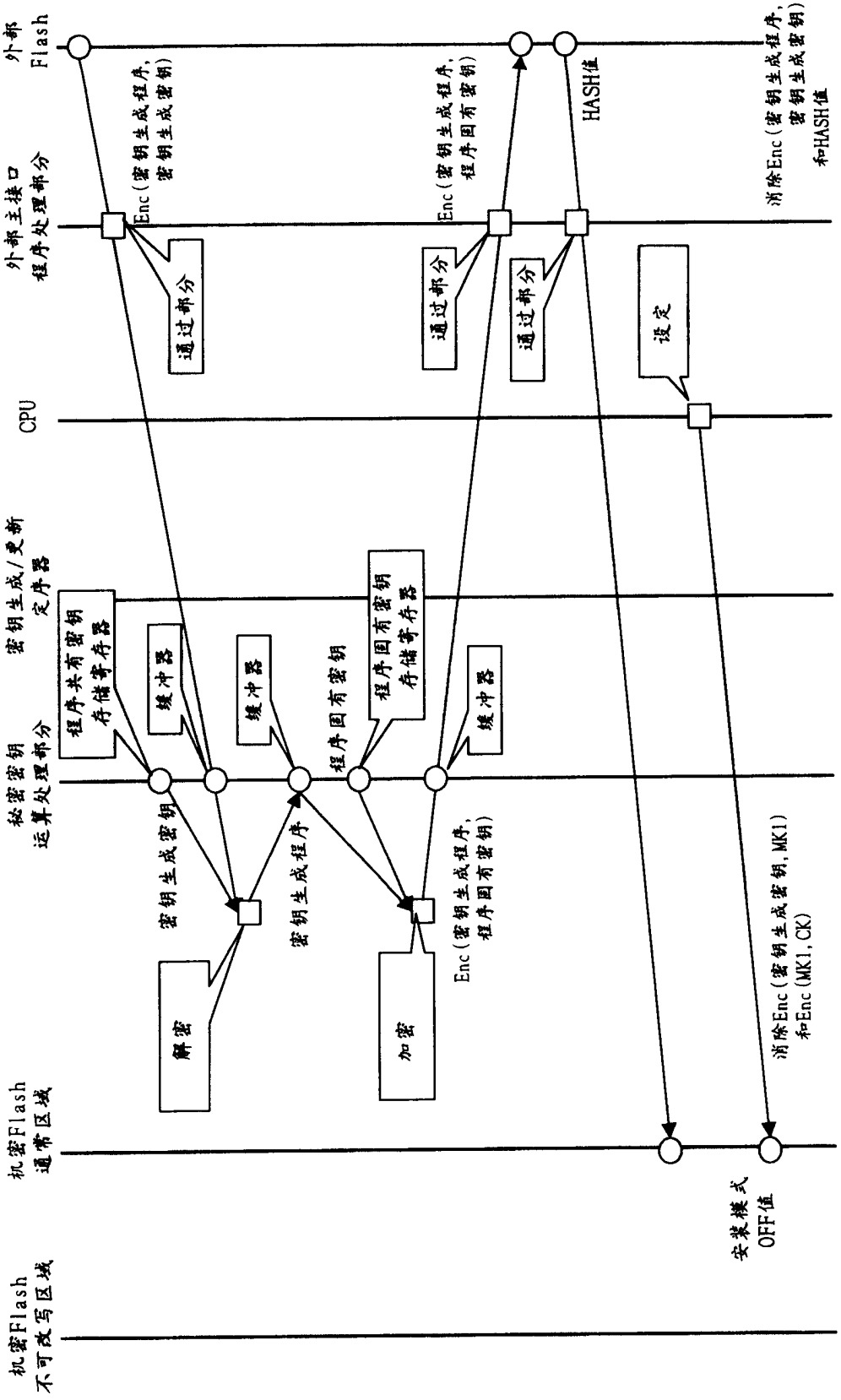

外部Flash
外部主接口程序处理部分
CPU
密钥生成/更新定序器
秘密密钥运算处理部分
机密Flash通常区域
机密Flash不可改写区域
Enc(密钥生成程序,密钥生成密钥)
通过部分
程序共有密钥存储寄存器
缓冲器
缓冲器
程序固有密钥
程序固有密钥存储寄存器
缓冲器
Enc(密钥生成程序,程序固有密钥)
通过部分
通过部分
HASH值
设定
消除Enc(密钥生成程序,密钥生成密钥)和HASH值
密钥生成密钥
密钥生成程序
解密
加密
Enc(密钥生成程序,程序固有密钥)
消除Enc(密钥生成密钥,MK1)和Enc(MK1,CK)
安装模式OFF值

图 10

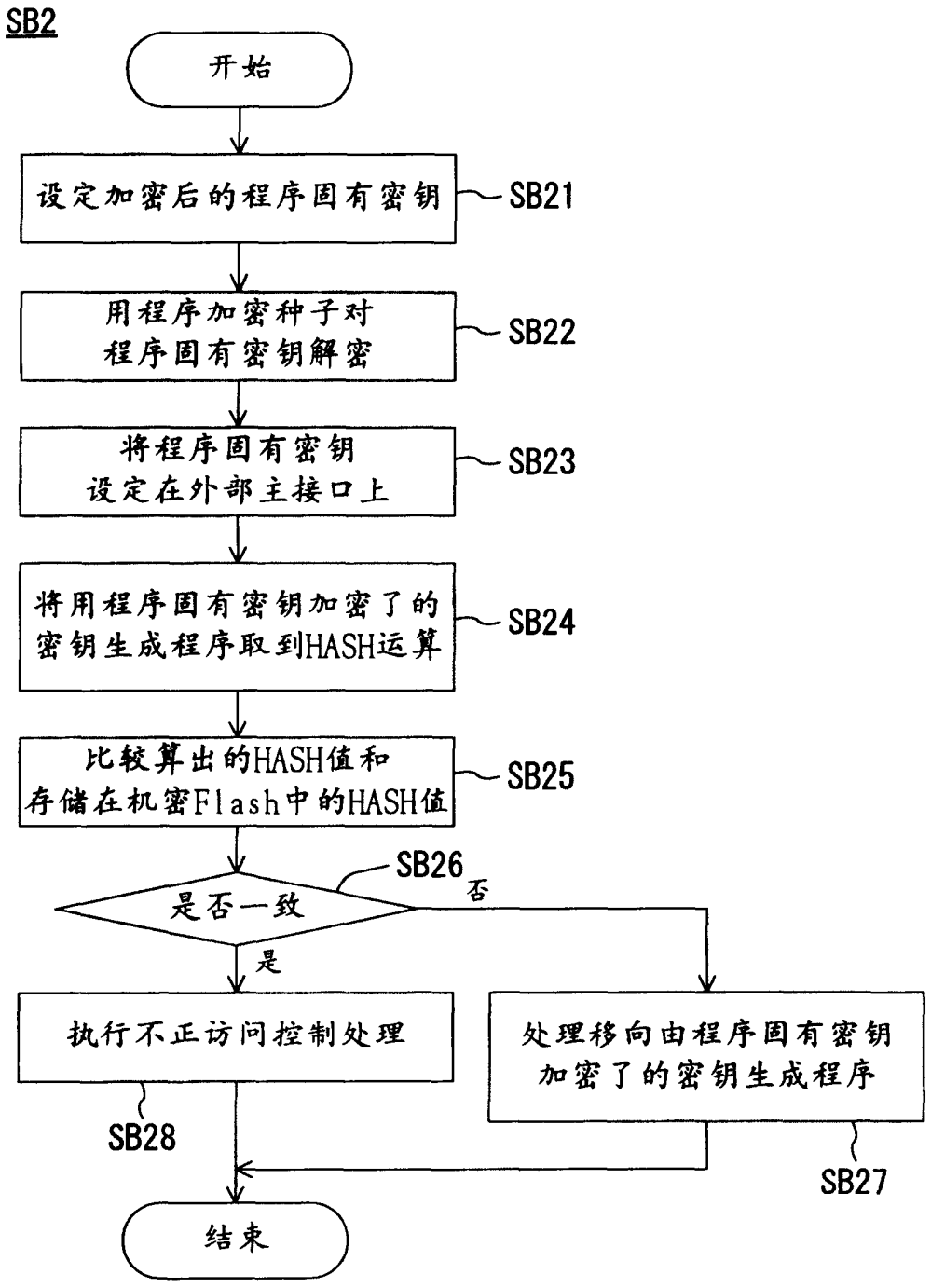

SB2
开始
设定加密后的程序固有密钥
SB21
用程序加密种子对
程序固有密钥解密
SB22
将程序固有密钥
设定在外部主接口上
SB23
将用程序固有密钥加密了的
密钥生成程序取到HASH运算
SB24
比较算出的HASH值和
存储在机密Flash中的HASH值
SB25
SB26
是否一致
否
是
执行不正访问控制处理
SB28
处理移向由程序固有密钥
加密了的密钥生成程序
SB27
结束

图 11

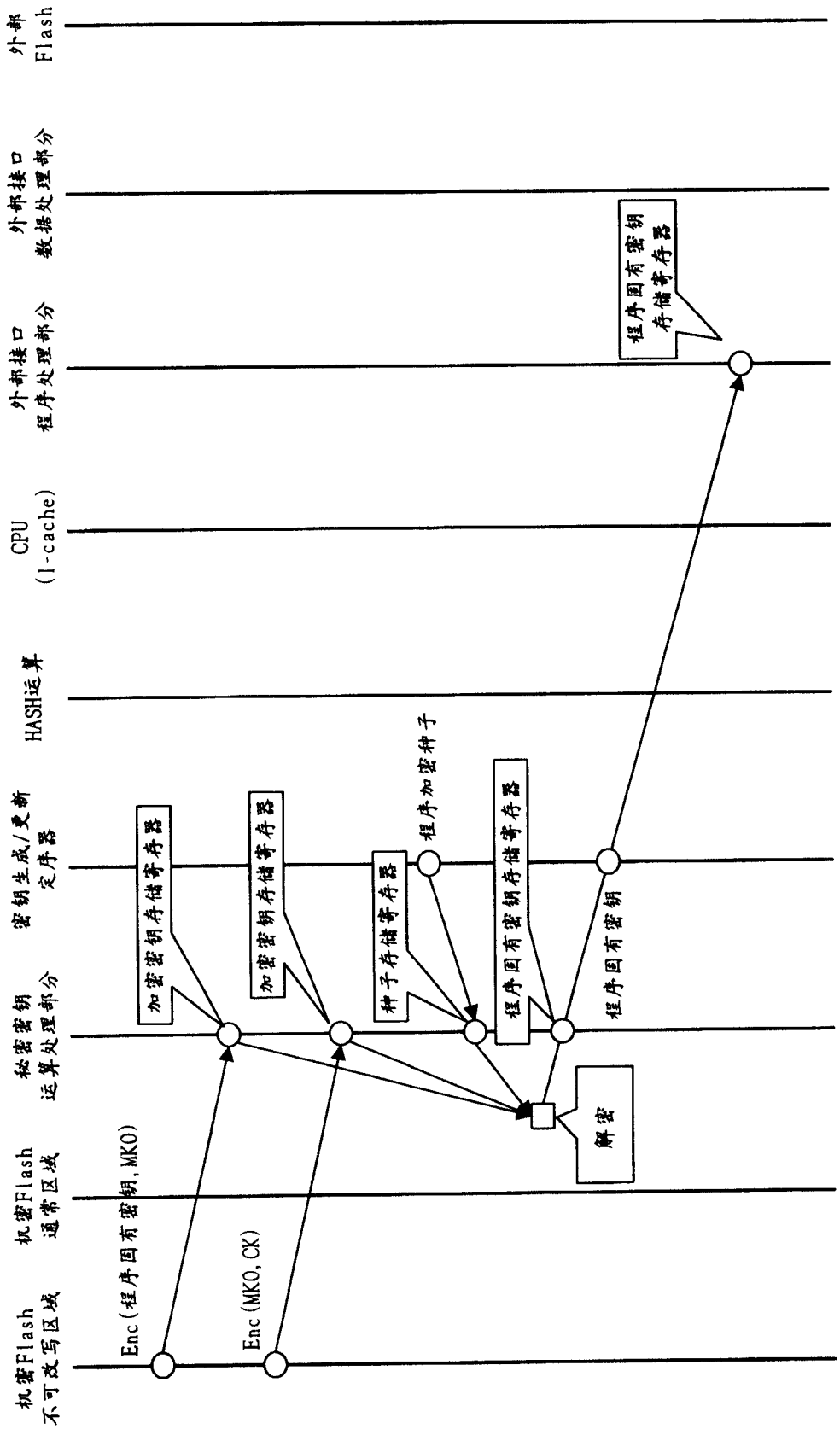

机密Flash
不可改写区域
机密Flash
通常区域
秘密密钥
运算处理部分
密钥生成/更新
定序器
HASH运算
CPU
(I-cache)
外部接口
程序处理部分
外部接口
数据处理部分
外部
Flash
Enc(程序固有密钥,MKO)
Enc(MKO,CK)
加密密钥存储寄存器
加密密钥存储寄存器
种子存储寄存器
程序加密种子
程序固有密钥存储寄存器
程序固有密钥
解密
程序固有密钥
存储寄存器

图 12

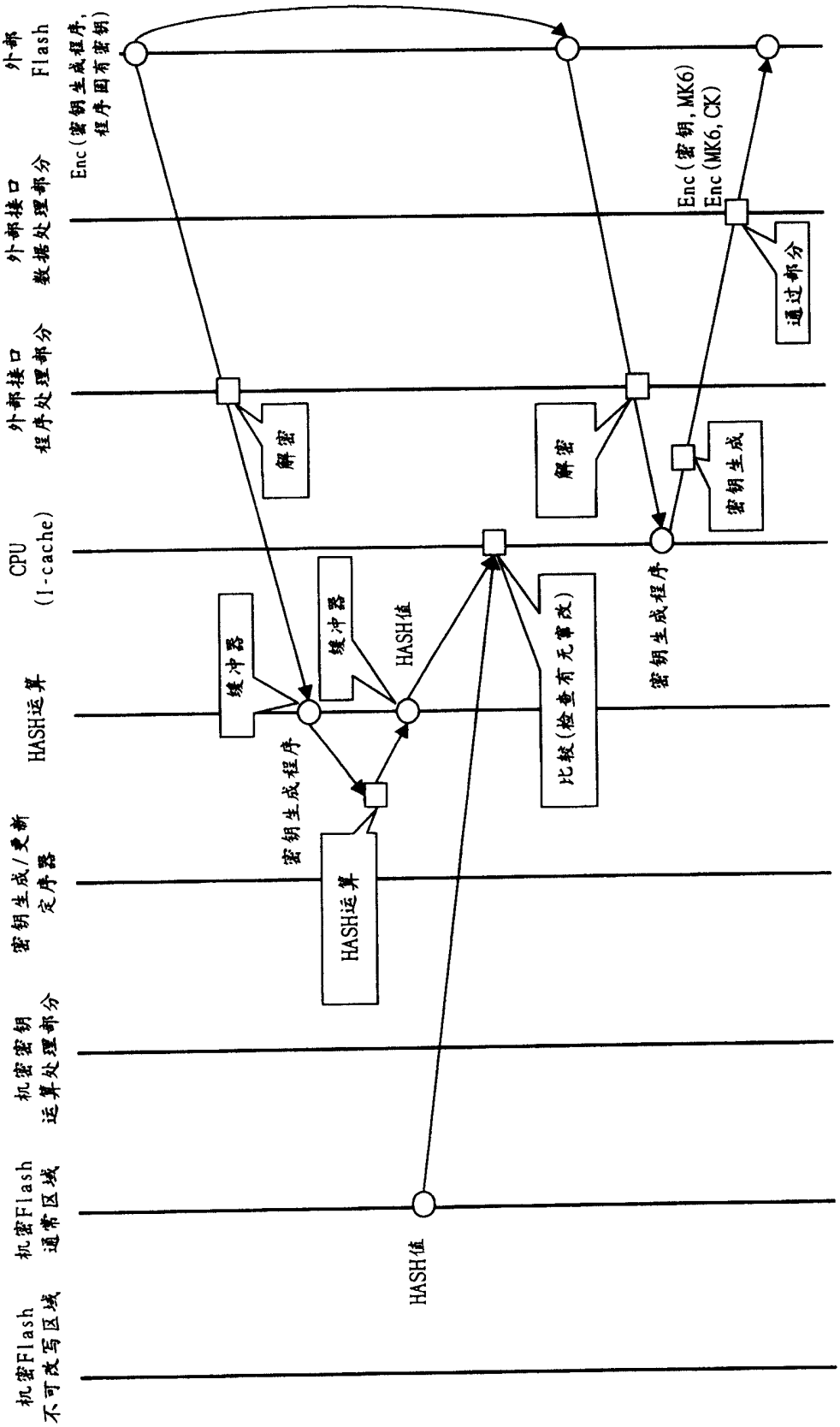

机密Flash
不可改写区域
机密Flash
通常区域
机密密钥
运算处理部分
密钥生成/更新
定序器
HASH运算
CPU
(I-cache)
外部接口
程序处理部分
外部接口
数据处理部分
外部
Flash
Enc(密钥生成程序,
程序固有密钥)
缓冲器
缓冲器
HASH值
密钥生成程序
HASH运算
HASH值
比较(检查有无窜改)
解密
解密
密钥生成程序
密钥生成
Enc(密钥,MK6)
Enc(MK6,CK)
通过部分

图 13

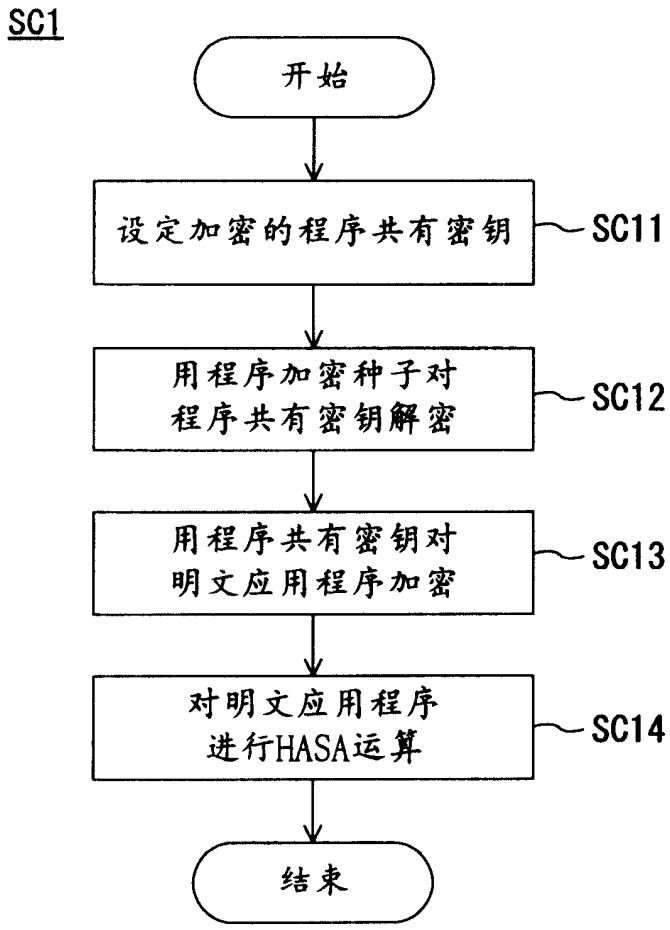

SC1
开始
设定加密的程序共有密钥
SC11
用程序加密种子对
程序共有密钥解密
SC12
用程序共有密钥对
明文应用程序加密
SC13
对明文应用程序
进行HASA运算
SC14
结束

图 14

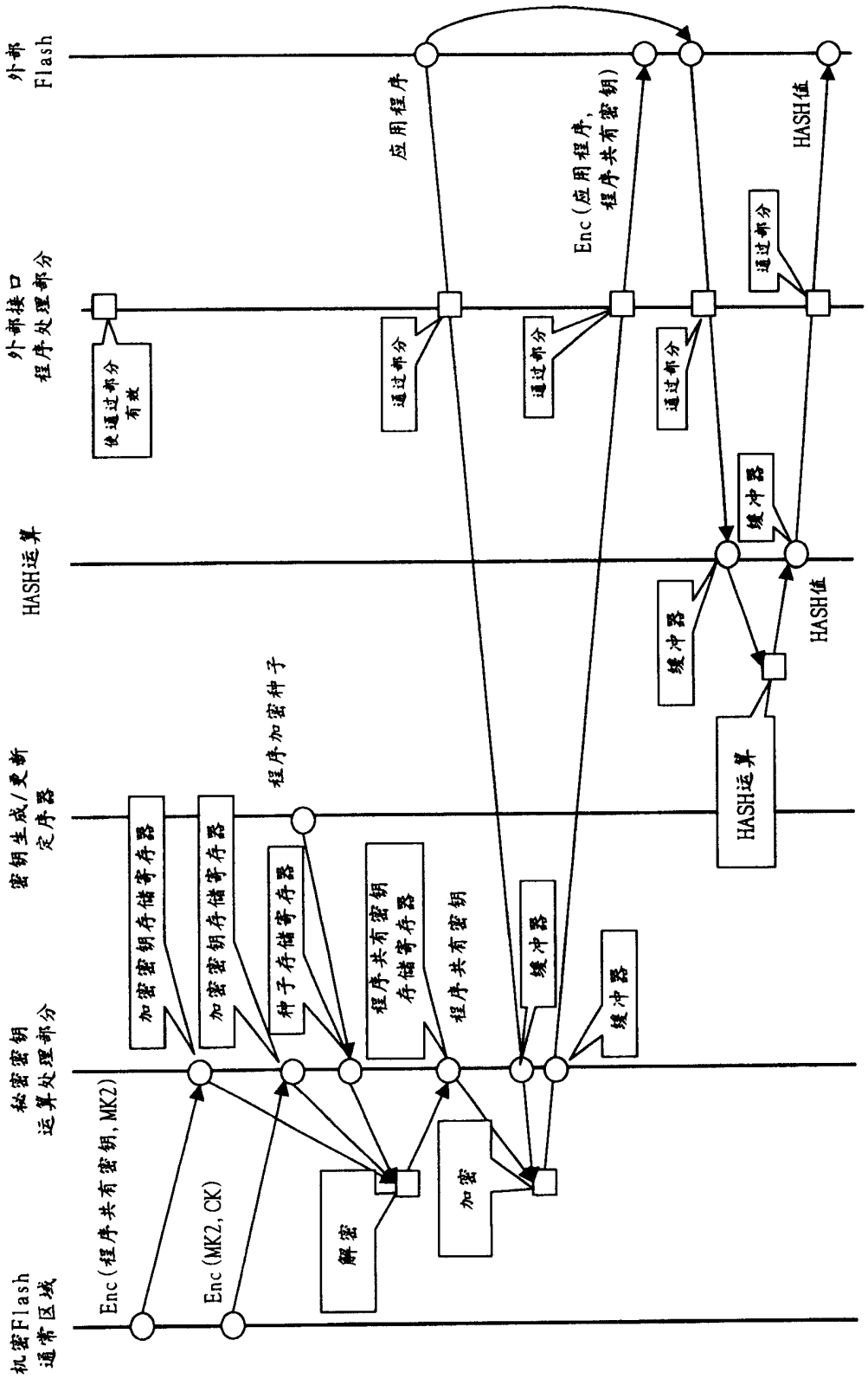

机密Flash
通常区域
秘密密钥
运算处理部分
密钥生成/更新
定序器
HASH运算
外部接口
程序处理部分
外部
Flash
Enc(程序共有密钥,MK2)
Enc(MK2,CK)
解密
加密
加密密钥存储寄存器
加密密钥存储寄存器
种子存储寄存器
程序加密种子
程序共有密钥
存储寄存器
程序共有密钥
缓冲器
缓冲器
HASH运算
缓冲器
HASH值
仅通过部分
有效
通过部分
通过部分
通过部分
通过部分
应用程序
Enc(应用程序,
程序共有密钥)
HASH值

图 15

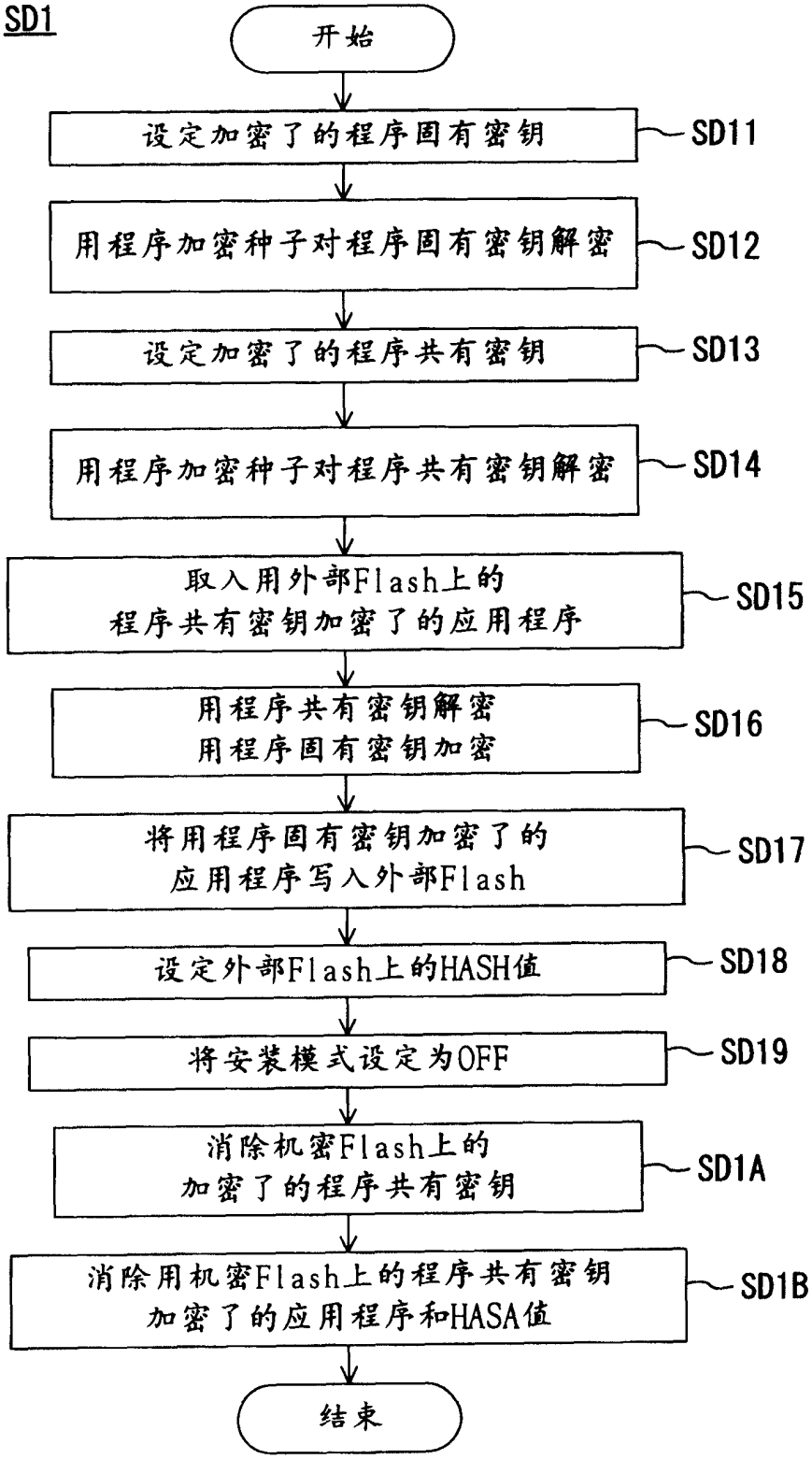

SD1
开始
设定加密了的程序固有密钥
SD11
用程序加密种子对程序固有密钥解密
SD12
设定加密了的程序共有密钥
SD13
用程序加密种子对程序共有密钥解密
SD14
取入用外部Flash上的
程序共有密钥加密了的应用程序
SD15
用程序共有密钥解密
用程序固有密钥加密
SD16
将用程序固有密钥加密了的
应用程序写入外部Flash
SD17
设定外部Flash上的HASH值
SD18
将安装模式设定为OFF
SD19
消除机密Flash上的
加密了的程序共有密钥
SD1A
消除用机密Flash上的程序共有密钥
加密了的应用程序和HASA值
SD1B
结束

图 16

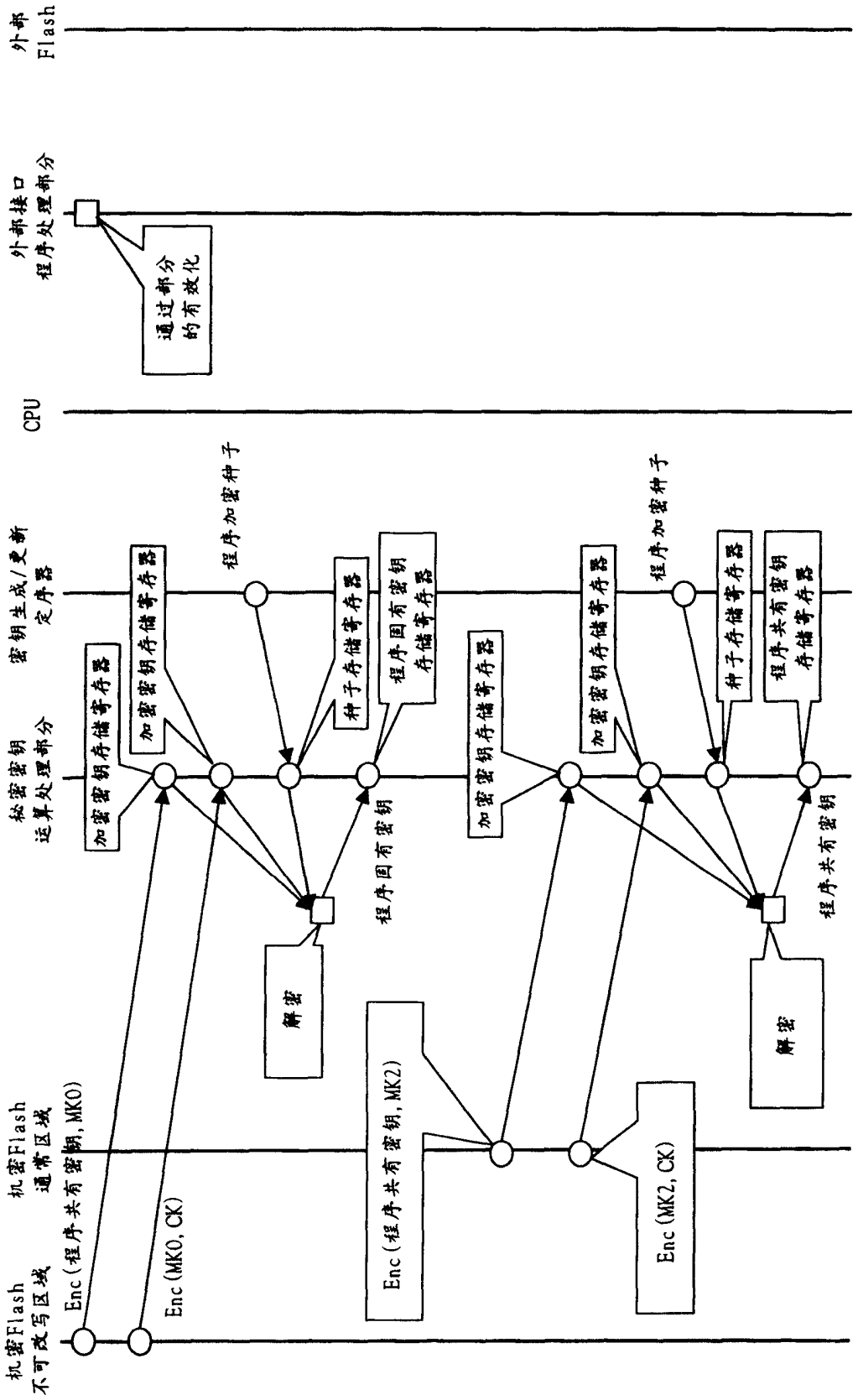

外部Flash
外部接口程序处理部分
CPU
密钥生成/更新定序器
秘密密钥运算处理部分
机密Flash通常区域
机密Flash不可改写区域
通过部分的有效化
加密密钥存储寄存器
加密密钥存储寄存器
程序加密种子
种子存储寄存器
程序固有密钥存储寄存器
加密密钥存储寄存器
加密密钥存储寄存器
程序加密种子
种子存储寄存器
程序共有密钥存储寄存器
解密
程序固有密钥
加密密钥存储寄存器
解密
程序共有密钥
Enc(程序共有密钥,MK0)
Enc(MK0,CK)
Enc(程序共有密钥,MK2)
Enc(MK2,CK)

图 17

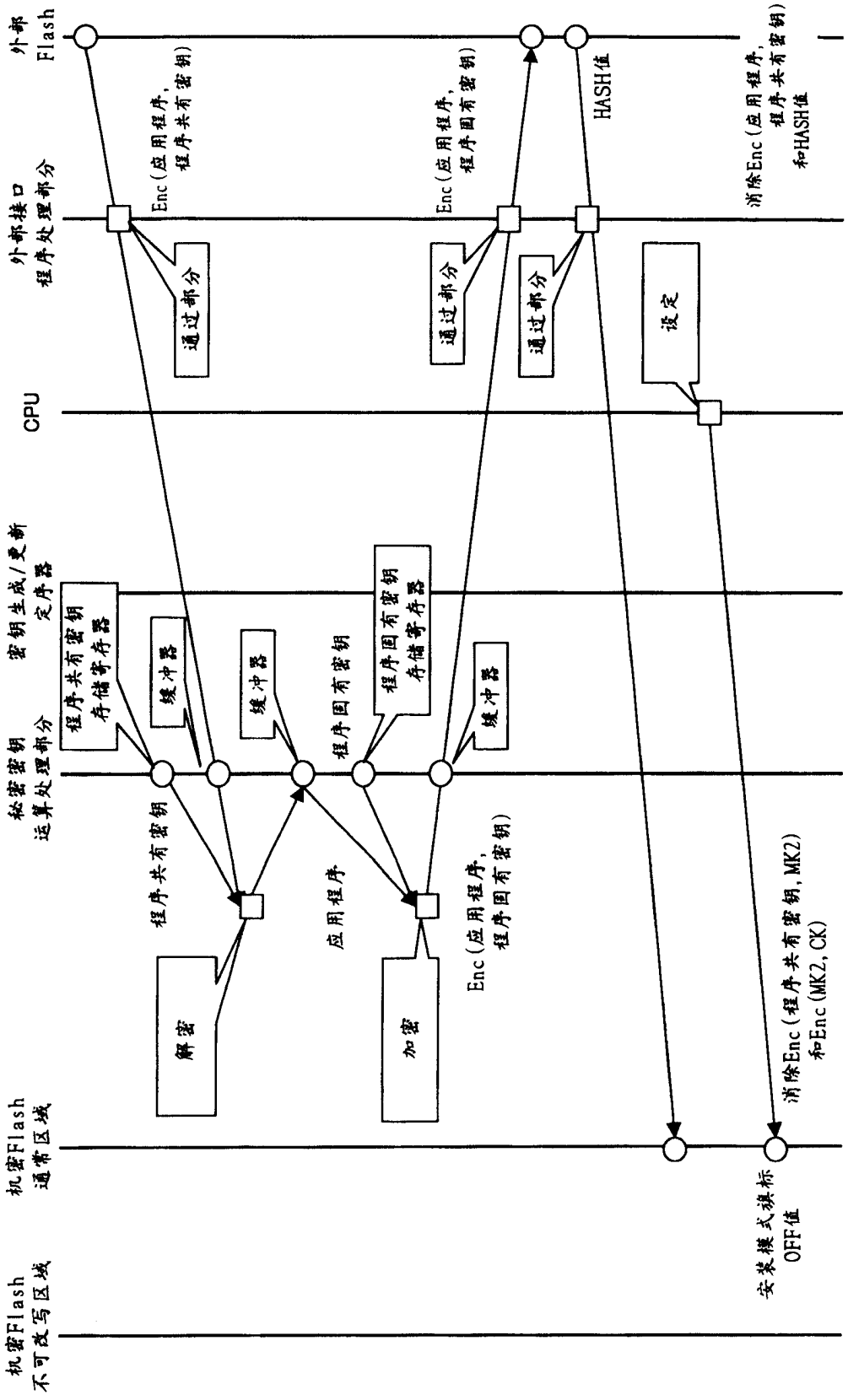

机密Flash
不可改写区域
机密Flash
通常区域
秘密密钥
运算处理部分
密钥生成/更新
定序器
CPU
外部接口
程序处理部分
外部
Flash
安装模式旗标
OFF值
程序共有密钥
存储寄存器
缓冲器
缓冲器
程序固有密钥
程序固有密钥
存储寄存器
缓冲器
程序共有密钥
解密
应用程序
加密
Enc(应用程序,
程序固有密钥)
消除Enc(程序共有密钥,MK2)
和Enc(MK2,CK)
Enc(应用程序,
程序共有密钥)
通过部分
Enc(应用程序,
程序固有密钥)
通过部分
HASH值
设定
消除Enc(应用程序,
程序共有密钥)
和HASH值

图 18

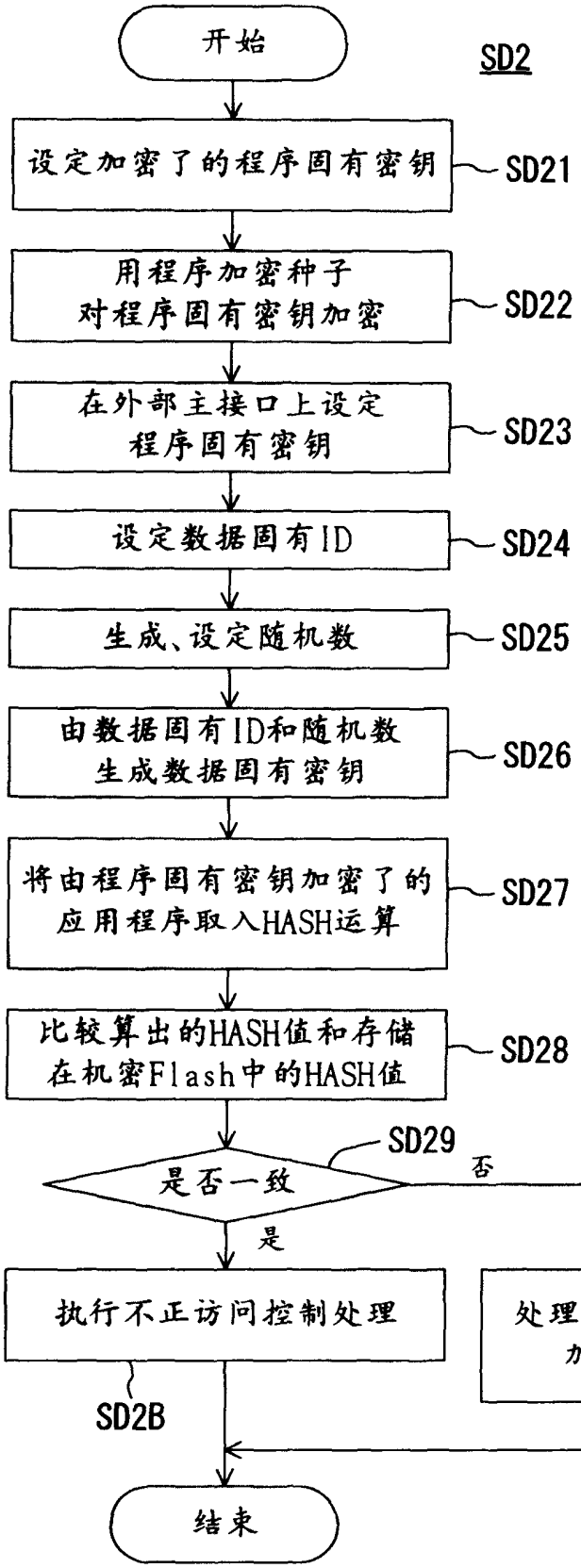

开始
SD2
设定加密了的程序固有密钥
SD21
用程序加密种子
对程序固有密钥加密
SD22
在外部主接口上设定
程序固有密钥
SD23
设定数据固有ID
SD24
生成、设定随机数
SD25
由数据固有ID和随机数
生成数据固有密钥
SD26
将由程序固有密钥加密了的
应用程序取入HASH运算
SD27
比较算出的HASH值和存储
在机密Flash中的HASH值
SD28
SD29
是否一致
否
是
执行不正访问控制处理
SD2B
处理移向由程序固有密钥
加密了的应用程序
SD2A
结束

图 19

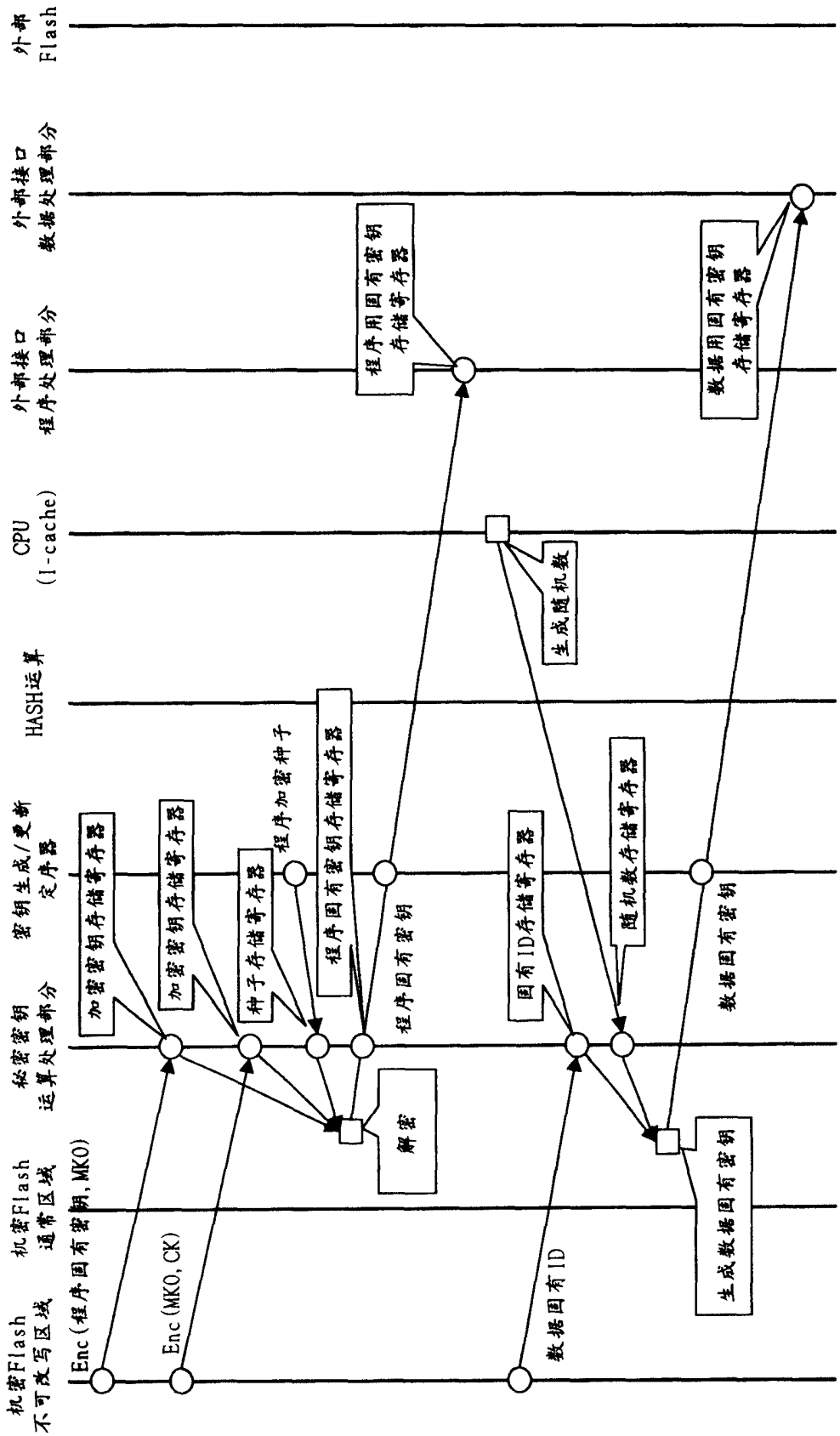

机密Flash
不可改写区域
机密Flash
通常区域
秘密密钥
运算处理部分
密钥生成/更新
定序器
HASH运算
CPU
(I-cache)
外部接口
程序处理部分
外部接口
数据处理部分
外部
Flash
Enc(程序固有密钥,MK0)
Enc(MK0,CK)
加密密钥存储寄存器
加密密钥存储寄存器
种子存储寄存器
程序加密种子
程序固有密钥存储寄存器
程序固有密钥
解密
固有ID存储寄存器
随机数存储寄存器
生成随机数
数据固有ID
生成数据固有密钥
数据固有密钥
程序用固有密钥
存储寄存器
数据用固有密钥
存储寄存器

图 20

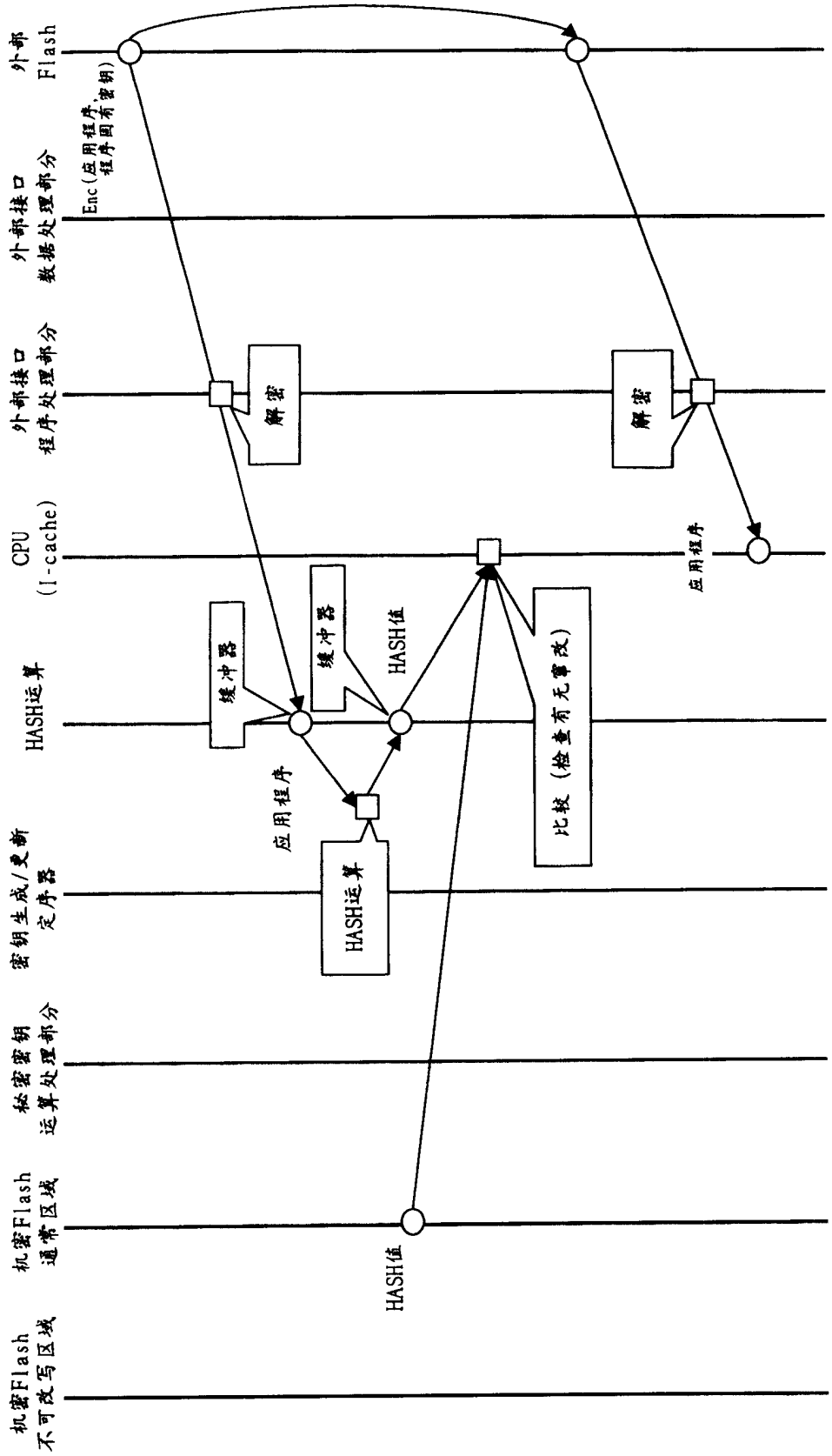

外部 Flash
外部接口 数据处理部分
外部接口 程序处理部分
CPU (I-cache)
HASH运算
密钥生成/更新 定序器
秘密密钥 运算处理部分
机密Flash 通常区域
机密Flash 不可改写区域
Enc(应用程序,程序固有密钥)
解密
解密
缓冲器
缓冲器
HASH值
应用程序
应用程序
HASH运算
比较（检查有无窜改）
HASH值

图 21

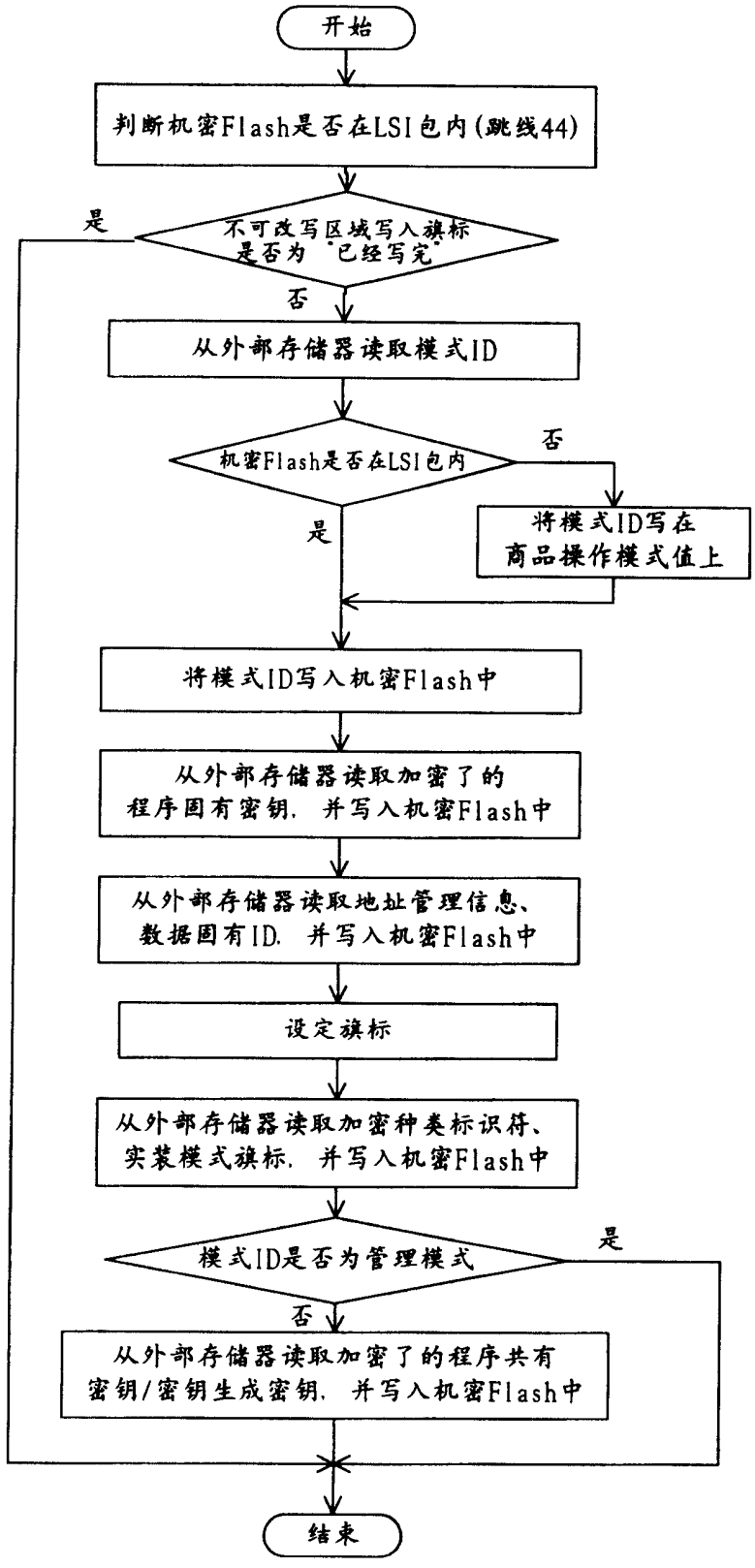

SZ1
开始
判断机密Flash是否在LSI包内(跳线44)
是
不可改写区域写入旗标
是否为"已经写完"
否
从外部存储器读取模式ID
机密Flash是否在LSI包内
否
是
将模式ID写在
商品操作模式值上
将模式ID写入机密Flash中
从外部存储器读取加密了的
程序固有密钥，并写入机密Flash中
从外部存储器读取地址管理信息、
数据固有ID，并写入机密Flash中
设定旗标
从外部存储器读取加密种类标识符、
实装模式旗标，并写入机密Flash中
模式ID是否为管理模式
是
否
从外部存储器读取加密了的程序共有
密钥/密钥生成密钥，并写入机密Flash中
结束

图 22