[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 12/14 (2006.01)
G06F 21/00 (2006.01)
G09C 1/00 (2006.01)

[12] 发明专利说明书

专利号 ZL 02801690.4

[45] 授权公告日 2009 年 1 月 7 日

[11] 授权公告号 CN 100449507C

[22] 申请日 2002.3.7 [21] 申请号 02801690.4

[30] 优先权

[32] 2001. 3.15 [33] JP [31] 73352/01

[86] 国际申请 PCT/JP2002/002112 2002.3.7

[87] 国际公布 WO2002/076012 日 2002.9.26

[85] 进入国家阶段日期 2003.1.15

[73] 专利权人 索尼公司

地址 日本东京都

[72] 发明人 吉野贤治 石桥义人 白井太三 高田昌幸

[56] 参考文献

JP2000 – 151583A 2000.5.30

JP6 – 222980A 1994.8.12

JP7 – 84959A 1995.3.31

JP2000 – 215165A 2000.8.4

JP6 – 289782A 1994.10.18

审查员 王 冉

[74] 专利代理机构 中国专利代理(香港)有限公司

代理人 刘宗杰 叶恺东

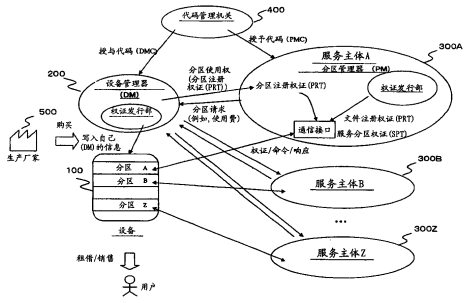
权利要求书 9 页 说明书 163 页 附图 99 页

[54] 发明名称

使用存取控制权证的存储器存取控制系统及管理方法

[57] 摘要

提供一种存储器存取控制系统，可独立地管理设备内创建的分割存储区域——分区。 在各设备或分区管理器的管理下发行各种存取控制权证以存取分割为多个分区的存储区域，根据各权证上记述的规则在搭载有存储器的设备中执行处理。 存储部具有作为由分区管理器管理的存储区域的分区区域、由设备管理器管理的设备管理器管理区域，可根据公开密钥、对称密钥中的某一种指定方式来执行分区鉴别、设备鉴别。



1、一种存储器存取控制系统，用于具有保存数据文件的存储部的搭载有存储器的设备，包括：

设备管理器，设置用来对所述搭载有存储器的设备执行设备管理；

分区管理器，设置用来对所述搭载有存储器的设备执行分区管理；和

所述搭载有存储器的设备，包括：

存储部，具有1个以上的分区区域，保存所述数据文件，作为由分区管理器管理的存储区域，以及还具有设备管理器管理区域，由作为该搭载有存储器的设备的管理者的设备管理器管理；

接收部，用于从存取机器接收所述设备管理器管理的存取控制权证、或所述分区管理器管理的存取控制权证作为对所述存储部的存取控制权证；和

控制部，用于按照由接收部接收的所述所接收权证的记述来执行处理。

2、如权利要求1所述的存储器存取控制系统，其特征在于，

上述存取控制权证

包含相互鉴别指定数据，该相互鉴别指定数据指定上述搭载有存储器的设备和输出权证的存取机器间应执行的相互鉴别形态；

上述搭载有存储器的设备

按照该存取控制权证的相互鉴别指定数据来执行相互鉴别，以鉴别成立为条件按照所接收权证的记录来执行处理。

3、如权利要求1所述的存储器存取控制系统，其特征在于，

上述存取控制权证

包含权证验证指定数据，该权证验证指定数据指定上述搭载有存储器的设备接收到的存取控制权证的验证形态；

上述搭载有存储器的设备

按照该存取控制权证的权证验证指定数据来执行权证验证处理，以验证成立为条件按照所接收权证的记录来执行处理。

4、如权利要求1所述的存储器存取控制系统，其特征在于，

上述存取控制权证

包含该存取控制权证的发行部件的范畴或标识符；

上述搭载有存储器的设备

根据从存取机器接收到的存取控制权证上记述的该存取控制权证的发行部件的范畴或标识符，来确认权证是合法的发行部件发行的权证，以该确认为条件按照所接收权证的记录来执行处理。

5、如权利要求 1 所述的存储器存取控制系统，其特征在于，

上述存取控制权证

包含该存取控制权证的利用部件--存取机器的范畴或标识符；

上述搭载有存储器的设备

根据从存取机器接收到的存取控制权证上记述的该存取控制权证的利用部件---存取机器的范畴或标识符，来确认权证是合法的利用部件提供的权证，以该确认为条件按照所接收权证的记录来执行处理。

6、如权利要求 1 所述的存储器存取控制系统，其特征在于，

在上述设备管理器管理的存取控制权证中，

包含容许对上述搭载有存储器的设备的存储部执行分区创建处理或删除处理的分区注册权证(PRT)；

上述搭载有存储器的设备

在从上述存取机器接收到分区注册权证(PRT)的情况下，

根据所接收分区注册权证(PRT)的记录来执行分区创建处理或删除处理。

7、如权利要求 6 所述的存储器存取控制系统，其特征在于，

上述分区注册权证(PRT)是从上述设备管理器管理的权证发行部件向上述分区管理器管理的作为权证用户的存取机器发行的。

8、如权利要求 1 所述的存储器存取控制系统，其特征在于，

在上述分区管理器管理的存取控制权证中，

包含容许在上述搭载有存储器的设备的存储部内创建的分区内执行数据文件创建处理或删除处理的文件注册权证(FRT)；

上述搭载有存储器的设备

在从上述存取机器接收到文件注册权证(FRT)的情况下，

根据所接收文件注册权证(FRT)的记录来执行文件创建处理或删除处理。

9、如权利要求 8 所述的存储器存取控制系统，其特征在于，

上述文件注册权证(FRT)是从上述分区管理器管理的权证发行部

件向上述分区管理器管理的作为权证用户的存取机器发行的。

10、如权利要求 1 所述的存储器存取控制系统，其特征在于，

在上述分区管理器管理的存取控制权证中，

包含容许对上述搭载有存储器的设备的存储部内的分区内的数据文件执行存取的服务许可权证(SPT)；

上述搭载有存储器的设备

在从上述存取机器接收到服务许可权证(SPT)的情况下，

根据所接收服务许可权证(SPT)的记录对数据文件执行存取处理。

11、如权利要求 10 所述的存储器存取控制系统，其特征在于，

上述服务许可权证(SPT)是从上述分区管理器管理的权证发行部件向上述分区管理器管理的作为权证用户的存取机器发行的。

12、如权利要求 1 所述的存储器存取控制系统，其特征在于，

在上述设备管理器或上述分区管理器管理的存取控制权证中，

包含容许对上述搭载有存储器的设备的存储部内保存的数据执行更新处理的数据更新权证(DUT)；

上述搭载有存储器的设备

在从上述存取机器接收到数据更新权证(DUT)的情况下，

根据所接收数据更新权证(DUT)的记录来执行数据更新处理。

13、如权利要求 12 所述的存储器存取控制系统，其特征在于，

用于更新上述设备管理器管理的设备管理器管理区域的数据的数据更新权证(DUT)是从上述设备管理器管理的权证发行部件向上述设备管理器管理的作为权证用户的存取机器发行的；

用于更新上述分区管理器管理的分区区域的数据的数据更新权证(DUT)是从上述分区管理器管理的权证发行部件向上述分区管理器管理的作为权证用户的存取机器发行的。

14、一种设备管理装置，对搭载有存储器的设备执行设备管理，包括用于发行分区注册权证(PRT)的发行部件，其中：

该搭载有存储器的设备具有：1 个以上的分区区域，保存所述数据文件，作为由分区管理器管理的存储区域，和设备管理器管理区域，由设备管理器管理，以及所述发行部件包括：

存储部，用于存储发行的所述分区注册权证(PRT)的内容和不同

信息;

控制部，用于控制发行所述分区注册权证(PRT)，所述分区注册权证(PRT)作为容许对上述搭载有存储器的设备的存储部执行分区创建处理或删除处理的存储器存取控制权证。

15、如权利要求 14 所述的设备管理装置，其特征在于，

上述设备管理装置

具有注册机构，该注册机构对上述搭载有存储器的设备执行公开密钥证书发行管理。

16、如权利要求 14 所述的设备管理装置，其特征在于，

上述分区注册权证(PRT)

包含相互鉴别指定数据，该相互鉴别指定数据指定上述搭载有存储器的设备和输出权证的存取机器间应执行的相互鉴别形态。

17、如权利要求 14 所述的设备管理装置，其特征在于，

上述分区注册权证(PRT)

包含权证验证指定数据，该权证验证指定数据指定上述搭载有存储器的设备接收到的存取控制权证的验证形态。

18、如权利要求 14 所述的设备管理装置，其特征在于，

上述分区注册权证(PRT)

包含该存取控制权证的发行部件的范畴或标识符。

19、如权利要求 14 所述的设备管理装置，其特征在于，

上述分区注册权证(PRT)

包含该存取控制权证的利用部件---存取机器的范畴或标识符。

20、一种分区管理装置，对搭载有存储器的设备执行分区管理，包括用于发行存取控制证的发行装置；其中：

所述搭载有存储器的设备具有：1 个以上的分区区域，保存上述数据文件，作为由分区管理器管理的存储区域，和设备管理器管理区域，由设备管理器管理；所述发行装置包括：

存储部，用于存储发行的所述存取控制证的内容和不同信息；

控制部，用于控制发行存取控制权证，该存取控制权证容许对上述搭载有存储器的设备的存储部内创建的分区执行存取。

21、如权利要求 20 所述的分区管理装置，其特征在于，

上述存取控制权证

是容许在上述搭载有存储器的设备的存储部内创建的分区内执行数据文件创建处理或删除处理的文件注册权证(FRT)。

22、如权利要求 20 所述的分区管理装置，其特征在于，

上述存取控制权证

是容许对上述搭载有存储器的设备的存储部内的分区内的数据文件执行存取的服务许可权证(SPT)。

23、如权利要求 20 所述的分区管理装置，其特征在于，

上述分区管理装置具有注册机构，该注册机构对上述搭载有存储器的设备执行公开密钥证书发行管理。

24、如权利要求 20 所述的分区管理装置，其特征在于，

上述存取控制权证

包含相互鉴别指定数据，该相互鉴别指定数据指定上述搭载有存储器的设备和输出权证的存取机器间应执行的相互鉴别形态。

25、如权利要求 20 所述的分区管理装置，其特征在于，

上述存取控制权证

包含权证验证指定数据，该权证验证指定数据指定上述搭载有存储器的设备接收到的存取控制权证的验证形态。

26、如权利要求 20 所述的分区管理装置，其特征在于，

上述存取控制权证

包含该存取控制权证的发行部件的范畴或标识符。

27、如权利要求 20 所述的分区管理装置，其特征在于，

上述存取控制权证

包含该存取控制权证的利用部件---存取机器的范畴或标识符。

28、一种搭载有存储器的设备，具有可保存数据的存储部，包括:

存储部，包括 1 个以上的分区区域，作为由分区管理器管理的存储区域，以及设备管理器管理区域，由作为该搭载有存储器的设备的管理者的设备管理器管理;

接收部，用于接收从存取机器接收上述设备管理器管理的存取控制权证、或上述分区管理器管理的存取控制权证作为对上述存储部的存取控制权证; 和

控制部，用于按照所接收权证的记述来执行处理。

29、如权利要求 28 所述的搭载有存储器的设备，其特征在于，

上述存取控制权证

包含相互鉴别指定数据，该相互鉴别指定数据指定上述搭载有存储器的设备和输出权证的存取机器间应执行的相互鉴别形态；

上述控制部件

按照该存取控制权证的相互鉴别指定数据来执行相互鉴别，以鉴别成立为条件按照所接收权证的记录来执行处理。

30、如权利要求 28 所述的搭载有存储器的设备，其特征在于，

上述存取控制权证

包含权证验证指定数据，该权证验证指定数据指定上述搭载有存储器的设备接收到的存取控制权证的验证形态；

上述控制部件

按照该存取控制权证的权证验证指定数据来执行权证验证处理，以验证成立为条件按照所接收权证的记录来执行处理。

31、如权利要求 28 所述的搭载有存储器的设备，其特征在于，

上述存取控制权证

包含该存取控制权证的发行部件的范畴或标识符；

上述控制部件

根据从存取机器接收到的存取控制权证上记述的该存取控制权证的发行部件的范畴或标识符，来确认权证是合法的发行部件发行的权证，以该确认为条件按照所接收权证的记录来执行处理。

32、如权利要求 28 所述的搭载有存储器的设备，其特征在于，

上述存取控制权证

包含该存取控制权证的利用部件---存取机器的范畴或标识符；

上述控制部件

根据从存取机器接收到的存取控制权证上记述的该存取控制权证的利用部件---存取机器的范畴或标识符，来确认权证是合法的利用部件提供的权证，以该确认为条件按照所接收权证的记录来执行处理。

33、一种存储器存取控制方法，用于具有保存数据文件的存储部的搭载有存储器的设备，其中，上述搭载有存储器的设备的存储部具有：1 个以上的分区区域，保存上述数据文件，作为由分区管理器管理的存储区域；以及设备管理器管理区域，由作为该搭载有存储器的

设备的管理者的设备管理器管理；所述存储器存取控制方法包括以下步骤：

从存取机器接收上述设备管理器管理的存取控制权证、或上述分区管理器管理的存取控制权证作为对上述存储部的存取控制权证；和

按照所接收权证的记述来执行处理。

34、如权利要求 33 所述的存储器存取控制方法，其中，上述存取控制权证包含相互鉴别指定数据，该相互鉴别指定数据指定上述搭载有存储器的设备和输出权证的存取机器间应执行的相互鉴别形态；所述存储器存取控制方法还包括以下步骤：

按照该存取控制权证的相互鉴别指定数据来执行相互鉴别，和

以鉴别成立为条件按照所接收权证的记录来执行处理。

35、如权利要求 33 所述的存储器存取控制方法，其中，所述存取控制权证包含权证验证指定数据，该权证验证指定数据指定上述搭载有存储器的设备接收到的存取控制权证的验证形态；所述存储器存取控制方法还包括以下步骤：

按照该存取控制权证的权证验证指定数据来执行权证验证处理，

以验证成立为条件按照所接收权证的记录来执行处理。

36、如权利要求 33 所述的存储器存取控制方法，其中，上述存取控制权证包含该存取控制权证的发行部件的范畴或标识符；所述存储器存取控制方法还包括以下步骤：

根据从存取机器接收到的存取控制权证上记述的该存取控制权证的发行部件的范畴或标识符，来确认权证是合法的发行部件发行的权证；

以该确认为条件按照所接收权证的记录来执行处理。

37、如权利要求 33 所述的存储器存取控制方法，其中，所述存取控制权证包含该存取控制权证的利用部件---存取机器的范畴或标识符；所述存储器存取控制方法还包括以下步骤：

根据从存取机器接收到的存取控制权证上记述的该存取控制权证的利用部件---存取机器的范畴或标识符，来确认权证是合法的利用部件提供的权证；

以该确认为条件按照所接收权证的记录来执行处理。

38、如权利要求 33 所述的存储器存取控制方法，其中，在上述设备管理器管理的存取控制权证中，包含容许对上述搭载有存储器的设备的存储部执行分区创建处理或删除处理的分区注册权证(PRT)；所述存储器存取控制方法还包括以下步骤：

当从上述存取机器接收到分区注册权证(PRT)时，根据所接收分区注册权证(PRT)的记录来执行分区创建处理或删除处理。

39、如权利要求 38 所述的存储器存取控制方法，其特征在于，

上述分区注册权证(PRT)是从上述设备管理器管理的权证发行部件向上述分区管理器管理的作为权证用户的存取机器发行的。

40、如权利要求 33 所述的存储器存取控制方法，其中，在上述分区管理器管理的存取控制权证中，包含容许在上述搭载有存储器的设备的存储部内创建的分区内执行数据文件创建处理或删除处理的文件注册权证(FRT)；所述存储器存取控制方法还包括以下步骤：

当从上述存取机器接收到文件注册权证(FRT)时，根据所接收文件注册权证(FRT)的记录来执行文件创建处理或删除处理。

41、如权利要求 40 所述的存储器存取控制方法，其特征在于，

上述文件注册权证(FRT)是从上述分区管理器管理的权证发行部件向上述分区管理器管理的作为权证用户的存取机器发行的。

42、如权利要求 33 所述的存储器存取控制方法，其中，在上述分区管理器管理的存取控制权证中，包含容许对上述搭载有存储器的设备的存储部内的分区内的数据文件执行存取的服务许可权证(SPT)；所述存储器存取控制方法还包括以下步骤：

当从上述存取机器接收到服务许可权证(SPT)时，根据所接收服务许可权证(SPT)的记录对数据文件执行存取处理。

43、如权利要求 42 所述的存储器存取控制方法，其特征在于，

上述服务许可权证(SPT)是从上述分区管理器管理的权证发行部件向上述分区管理器管理的作为权证用户的存取机器发行的。

44、如权利要求 33 所述的存储器存取控制方法，其中，在上述设备管理器或上述分区管理器管理的存取控制权证中，包含容许对上述搭载有存储器的设备的存储部内保存的数据执行更新处理的数据更新权证(DUT)；所述存储器存取控制方法还包括以下步骤：

当在从上述存取机器接收到数据更新权证(DUT)时，根据所接收

数据更新权证(DUT)的记录来执行数据更新处理。

45、如权利要求 33 所述的存储器存取控制方法，其特征在于，

用于更新上述设备管理器管理的设备管理器管理区域的数据的数据更新权证(DUT)是从上述设备管理器管理的权证发行部件向上述设备管理器管理的作为权证用户的存取机器发行的；

用于更新上述分区管理器管理的分区区域的数据的数据更新权证(DUT)是从上述分区管理器管理的权证发行部件向上述分区管理器管理的作为权证用户的存取机器发行的。

使用存取控制权证的
存储器存取控制系统及管理方法

技术领域

本发明涉及存储器存取控制系统、设备管理装置、分区管理装置、搭载有存储器的设备、及存储器存取控制方法、以及程序存储媒体。更详细地说，涉及下述存储器存取控制系统、设备管理装置、分区管理装置、搭载有存储器的设备、及存储器存取控制方法、以及程序存储媒体：将1个存储器划分为多个区域(分区)，在各分区内保存服务提供者或相关实体管理的数据，用户可用1个搭载有存储器的设备来提供各种服务。

背景技术

以往，作为具有存储器的设备，使用着磁带媒体、软盘、硬盘、光盘、半导体媒体等。其中，作为能够安全地管理设备内的存储器的设备，半导体媒体一直引人注目。其理由是因为，半导体存储器容易实现从外部难以存取的结构、即防窜改结构。

防窜改结构例如通过下述结构来实现：设备由单个半导体芯片构成，在该芯片上包括控制部、存储器控制器、非易失性存储器、电压检测部件、频率检测部件等，为了使非易失性存储器从外部难以读写，由铝层等哑(ダミー)层夹着。

用图96“现有存储器结构”来说明这种安全设备的现有存储器结构。图96的存储器示出例如可用作电子货币的存储器结构。如图96所示，存储区域大体分为3个。即，数据区域、存储器管理区域、系统区域。

在数据区域中根据各数据内的头部保存的“数据结构”而保存有数据，在本例中，保存有用户名、地址、电话号码、金额、备注、日志各数据。在存储器管理区域中，保存有用于存取数据区域的各数据的保存地址、存取方法、存取鉴别密钥等。例如，数据区域的数据1(用户名)的存取可利用存取鉴别密钥(0123……)来只读(Read)。此外，在系统区域中，保存有设备标识符(ID)、用于在数据区域中保留存储区域的鉴别密钥---存储器管理密钥等。

图96所示的存储设备的数据区域可分割为多个，这些分割数据区域可以由不同的服务主体来管理，例如如果是电子货币，则可以分

别由不同的电子货币服务提供主体(例如，不同的银行)来管理。各分割区域的数据除了由各个服务提供主体之外，还由用户、例如利用电子货币进行商品销售的商店具备的作为设备存取机器的读写器(专用读写器或PC等)来执行数据的读出、写入(例如，余额数据的更新)。

具有图 96 所示的多个分割数据区域的安全设备的管理者和用户的关系示于图 97“存储器管理者、用户”。如图 97 所示，存储器管理者是安全设备的发行主体，而存储器用户请该存储器管理者分配存储区域，来利用该分配的存储器。作为存储器用户，有数据 1 用户～数据 6 用户。例如根据前述电子货币的例子，存储器用户是银行或商店等。

存储器管理者知道用于保留存储区域的存取控制用的存储器管理密钥，利用该存储器管理密钥来分配各个存储器用户的存储器(分割数据区域)。而存储器用户知道用于存取各数据区域的数据的存取鉴别密钥，能够利用该存取鉴别密钥来分别存取所分配的数据区域内的存储器。存取的形态各种各样，有数据的读出(Read)、写入(Write)、余额的减少(Decrement)等，可以按照各个处理形态来个别设定存取鉴别密钥，设定可否进行个别处理。

例如图 96 所示的存储器中的数据 4 是金额数据，如图 97 所示，数据 4 的用户可以对数据 4 进行减少(Decrement)处理和读写(Read/Write)处理。如图 96 右下的表所示，在减少(Decrement)处理和读写(Read/Write)处理中，存取密钥不同，必须使用与各处理对应的存取密钥来存取存储器。

图 98 示出存储器管理者向存储器用户分配存储设备内的某个数据区域的存储器保留处理的说明图。如图 98 的“存储器保留方式”所示，存储器管理者用图左侧所示的存储器保留读写器(R/W: Reader/Writer)对图右侧所示的存储设备执行数据区域保留处理。在存储器保留读写器(R/W: Reader/Writer)中，包括用于保存存储器管理密钥的安全 NVRAM (Non-Volatile RAM，非易失性随机存取存储器)。其中，存储器保留 R/W 可以是安全设备专用的读写 R/W; 而在安全设备是具有 USB、PCMCIA 等 I/F(接口)的设备的情况下，也可以是可经这些 I/F 进行读写的装置、例如 PC。

为了用 R/W 来保留存储器，首先从安全设备中读出设备 ID。接着在 R/W 内，用存储器管理密钥和设备 ID 来生成鉴别密钥，用生成的

鉴别密钥与安全设备执行相互鉴别。相互鉴别处理例如根据基于对称密钥体制的相互鉴别(例如，ISO/IEC9798-2)来执行。

在相互鉴别成功后，R/W 用会话密钥对数据结构、数据长度、存取方法、存取鉴别密钥进行加密，在必要时附加数据验证用的 MAC (Message Authentication Code，报文鉴别码)值，向安全设备发送命令。接收到命令的安全设备对接收数据进行解密，在必要时通过 MAC 验证处理来验证数据完整性，其后，按照接收数据内的数据长度在存储器的数据区域中保留存储区域，向保留的区域中写入数据结构，并且向存储器管理区域中写入保留的存储器的地址、存取方法、存取鉴别密钥。

这样，在存储设备中设定了多个分割数据区域。接着，根据图 99 的“存储器存取方法”，来说明具有多个分割数据区域的存储设备的存储器存取方法。图 99 左侧的读写器是存储器用户具有的存储器存取读写器(R/W)，与上述存储器保留 R/W 同样，由专用 R/W 或 PC 等构成。在存储器存取读写器(R/W)中，包括用于保存存取鉴别密钥的安全 NVRAM。为了用 R/W 来存取安全设备的数据区域，首先从安全设备中读出设备 ID。接着在 R/W 内，用存取鉴别密钥和设备 ID 来生成鉴别密钥，用生成的鉴别密钥与安全设备执行相互鉴别。在相互鉴别成功后，R/W 对与存取鉴别密钥对应的数据区域的数据进行规定的存取。

此时在存储器管理区域中规定了存取方法，所以例如如图 99 的“存储器存取方法”所示，在数据 4(金额数据)的减少用的存取鉴别成功的情况下，可进行数据 4 的数据的减少，但是不能进行增加、或自由的改写处理。通过这样按照各个存取形态来不同设定鉴别处理所用的存取鉴别密钥，能够提高各数据的安全性。例如即使在减少处理 R/W 被盗、被盗的减少处理 R/W 内的 NVRAM 被破译的情况下，也能够降低对图 99 的安全设备内的数据 4(金额数据)进行非法增加处理的可能性。

一般，存款终端与 ATM 同样，能够提高安全性，但是取款终端多在商店等中被用作交付商品时的收银机，设置场所也各种各样，终端被盗的风险也高，难以提高安全性。因此，对数据存取采用不同的存取鉴别密钥的结构很有效。

在上述现有具有分割数据区域的存储设备的利用形态中，在存储

器的数据区域保留处理、各数据区域的存取处理中，通过分别用存储器管理密钥执行鉴别处理，或用存取鉴别密钥执行鉴别处理来执行各个处理，但是它们具体应用例如基于 DES 加密算法的对称密钥，未考虑基于公开密钥体制的鉴别、或基于公开密钥体制的验证。

如上所述，在存储器管理密钥、存取鉴别密钥应用对称密钥的结构中，优点是能用一次处理来执行鉴别及存取允许，而缺点是在鉴别密钥泄漏时，可用泄漏密钥来进行存储器存取，在安全性上成问题。

此外，为了降低对存储设备执行存取的读写器(R/W)的成本，也可以考虑在读写器(R/W)中不安装加密算法的结构，但是如果采用这种结构，则不能与设备执行鉴别、通信数据的加密等处理，不适合作为与保存用户的金额数据、用户的其他私人信息等的设备对应的读写器。

发明内容

本发明就是鉴于上述现有技术的现状而提出的，其目的在于，通过在各设备或分区管理实体的管理下对分割为多个分区的存储区域的存取发行各种存取控制权证，在搭载有存储器的设备中根据各权证上记述的规则执行处理，来实现各分区内数据的独立管理结构。

此外，其目的在于提供存储器存取控制系统、设备管理装置、分区管理装置、搭载有存储器的设备、及存储器存取控制方法、以及程序存储媒体，可根据公开密钥、对称密钥中的某一种指定方式来执行分区鉴别、设备鉴别，在各种环境下都可执行安全的数据通信。

本发明的第 1 侧面是一种存储器存取控制系统，用于具有保存数据文件的存储部的搭载有存储器的设备，包括:

设备管理器，设置用来对所述搭载有存储器的设备执行设备管理;

分区管理器，设置用来对所述搭载有存储器的设备执行分区管理; 和

所述搭载有存储器的设备，包括:

存储部，具有 1 个以上的分区区域，保存所述数据文件，作为由分区管理器管理的存储区域，以及还具有设备管理器管理区域，由作为该搭载有存储器的设备的管理者的设备管理器管理;

接收部，用于从存取机器接收所述设备管理器管理的存取控制权证、或所述分区管理器管理的存取控制权证作为对所述存储部的存取

控制权证；和

控制部，用于按照由接收部接收的所述所接收权证的记述来执行处理。

再者，在本发明的存储器存取控制系统的一实施形态中，其特征在于，上述存取控制权证包含相互鉴别指定数据，该相互鉴别指定数据指定上述搭载有存储器的设备和输出权证的存取机器间应执行的相互鉴别形态；上述搭载有存储器的设备按照该存取控制权证的相互鉴别指定数据来执行相互鉴别，以鉴别成立为条件按照所接收权证的记录来执行处理。

再者，在本发明的存储器存取控制系统的一实施形态中，其特征在于，上述存取控制权证包含权证验证指定数据，该权证验证指定数据指定上述搭载有存储器的设备接收到的存取控制权证的验证形态；上述搭载有存储器的设备按照该存取控制权证的权证验证指定数据来执行权证验证处理，以验证成立为条件按照所接收权证的记录来执行处理。

再者，在本发明的存储器存取控制系统的一实施形态中，其特征在于，上述存取控制权证包含该存取控制权证的发行部件的范畴或标识符；上述搭载有存储器的设备根据从存取机器接收到的存取控制权证上记述的该存取控制权证的发行部件的范畴或标识符，来确认权证是合法的发行部件发行的权证，以该确认为条件按照所接收权证的记录来执行处理。

再者，在本发明的存储器存取控制系统的一实施形态中，其特征在于，上述存取控制权证包含该存取控制权证的利用部件---存取机器的范畴或标识符；上述搭载有存储器的设备根据从存取机器接收到的存取控制权证上记述的该存取控制权证的利用部件---存取机器的范畴或标识符，来确认权证是合法的利用部件提供的权证，以该确认为条件按照所接收权证的记录来执行处理。

再者，在本发明的存储器存取控制系统的一实施形态中，其特征在于，在上述设备管理器管理的存取控制权证中，包含容许对上述搭载有存储器的设备的存储部执行分区创建处理或删除处理的分区注册权证(PRT)；上述搭载有存储器的设备在从上述存取机器接收到分区注册权证(PRT)的情况下，根据所接收分区注册权证(PRT)的记录来执行分区创建处理或删除处理。

再者，在本发明的存储器存取控制系统的一实施形态中，其特征在于，上述分区注册权证(PRT)是从上述设备管理器管理的权证发行部件向上述分区管理器管理的作为权证用户的存取机器发行的。

再者，在本发明的存储器存取控制系统的一实施形态中，其特征在于，在上述分区管理器管理的存取控制权证中，包含容许在上述搭载有存储器的设备的存储部内创建的分区内执行数据文件创建处理或删除处理的文件注册权证(FRT)；上述搭载有存储器的设备在从上述存取机器接收到文件注册权证(FRT)的情况下，根据所接收文件注册权证(FRT)的记录来执行文件创建处理或删除处理。

再者，在本发明的存储器存取控制系统的一实施形态中，其特征在于，上述文件注册权证(FRT)是从上述分区管理器管理的权证发行部件向上述分区管理器管理的作为权证用户的存取机器发行的。

再者，在本发明的存储器存取控制系统的一实施形态中，其特征在于，在上述分区管理器管理的存取控制权证中，包含容许对上述搭载有存储器的设备的存储部内的分区内的数据文件执行存取的服务许可权证(SPT)；上述搭载有存储器的设备在从上述存取机器接收到服务许可权证(SPT)的情况下，根据所接收服务许可权证(SPT)的记录对数据文件执行存取处理。

再者，在本发明的存储器存取控制系统的一实施形态中，其特征在于，上述服务许可权证(SPT)是从上述分区管理器管理的权证发行部件向上述分区管理器管理的作为权证用户的存取机器发行的。

再者，在本发明的存储器存取控制系统的一实施形态中，其特征在于，在上述设备管理器或上述分区管理器管理的存取控制权证中，包含容许对上述搭载有存储器的设备的存储部内保存的数据执行更新处理的数据更新权证(DUT)；上述搭载有存储器的设备在从上述存取机器接收到数据更新权证(DUT)的情况下，根据所接收数据更新权证(DUT)的记录来执行数据更新处理。

再者，在本发明的存储器存取控制系统的一实施形态中，其特征在于，用于更新上述设备管理器管理的设备管理器管理区域的数据的数据更新权证(DUT)是从上述设备管理器管理的权证发行部件向上述设备管理器管理的作为权证用户的存取机器发行的；用于更新上述分区管理器管理的分区区域的数据的数据更新权证(DUT)是从上述分区管理器管理的权证发行部件向上述分区管理器管理的作为权证用户

的存取机器发行的。

再者，本发明的第 2 侧面是一种设备管理装置，对搭载有存储器的设备执行设备管理，包括用于发行分区注册权证(PRT)的发行部件，其中：

该搭载有存储器的设备具有：1 个以上的分区区域，保存所述数据文件，作为由分区管理器管理的存储区域，和设备管理器管理区域，由设备管理器管理，以及所述发行部件包括：

存储部，用于存储发行的所述分区注册权证(PRT)的内容和不同信息；

控制部，用于控制发行所述分区注册权证(PRT)，所述分区注册权证(PRT)作为容许对上述搭载有存储器的设备的存储部执行分区创建处理或删除处理的存储器存取控制权证。

再者，在本发明的设备管理装置的一实施形态中，其特征在于，上述设备管理装置具有注册机构，该注册机构对上述搭载有存储器的设备执行公开密钥证书发行管理。

再者，在本发明的设备管理装置的一实施形态中，其特征在于，上述分区注册权证(PRT)包含相互鉴别指定数据，该相互鉴别指定数据指定上述搭载有存储器的设备和输出权证的存取机器间应执行的相互鉴别形态。

再者，在本发明的设备管理装置的一实施形态中，其特征在于，上述分区注册权证(PRT)包含权证验证指定数据，该权证验证指定数据指定上述搭载有存储器的设备接收到的存取控制权证的验证形态。

再者，在本发明的设备管理装置的一实施形态中，其特征在于，上述分区注册权证(PRT)包含该存取控制权证的发行部件的范畴或标识符。

再者，在本发明的设备管理装置的一实施形态中，其特征在于，上述分区注册权证(PRT)包含该存取控制权证的利用部件---存取机器的范畴或标识符。

再者，本发明的第 3 侧面是一种分区管理装置，对搭载有存储器的设备执行分区管理，包括用于发行存取控制证的发行装置；其中：所述搭载有存储器的设备具有：1 个以上的分区区域，保存上述数据文件，作为由分区管理器管理的存储区域，和设备管理器管理区域，

由设备管理器管理；所述发行装置包括：

存储部，用于存储发行的所述存取控制证的内容和不同信息；

控制部，用于控制发行存取控制权证，该存取控制权证容许对上述搭载有存储器的设备的存储部内创建的分区执行存取。

再者，在本发明的分区管理装置的一实施形态中，其特征在于，上述存取控制权证是容许在上述搭载有存储器的设备的存储部内创建的分区内执行数据文件创建处理或删除处理的文件注册权证(FRT)。

再者，在本发明的分区管理装置的一实施形态中，其特征在于，上述存取控制权证是容许对上述搭载有存储器的设备的存储部内的分区内的数据文件执行存取的服务许可权证(SPT)。

再者，在本发明的分区管理装置的一实施形态中，其特征在于，上述分区管理装置具有注册机构，该注册机构对上述搭载有存储器的设备执行公开密钥证书发行管理。

再者，在本发明的分区管理装置的一实施形态中，其特征在于，上述存取控制权证包含相互鉴别指定数据，该相互鉴别指定数据指定上述搭载有存储器的设备和输出权证的存取机器间应执行的相互鉴别形态。

再者，在本发明的分区管理装置的一实施形态中，其特征在于，上述存取控制权证包含权证验证指定数据，该权证验证指定数据指定上述搭载有存储器的设备接收到的存取控制权证的验证形态。

再者，在本发明的分区管理装置的一实施形态中，其特征在于，上述存取控制权证包含该存取控制权证的发行部件的范畴或标识符。

再者，在本发明的分区管理装置的一实施形态中，其特征在于，上述存取控制权证包含该存取控制权证的利用部件---存取机器的范畴或标识符。

再者，本发明的第 4 侧面是一种搭载有存储器的设备，具有可保存数据的存储部，包括：

存储部，包括 1 个以上的分区区域，作为由分区管理器管理的存储区域，以及设备管理器管理区域，由作为该搭载有存储器的设备的管理者的设备管理器管理；

接收部，用于接收从存取机器接收上述设备管理器管理的存取控

制权证、或上述分区管理器管理的存取控制权证作为对上述存储部的存取控制权证；和

控制部，用于按照所接收权证的记述来执行处理。

再者，在本发明的搭载有存储器的设备的一实施形态中，其特征在于，上述存取控制权证包含相互鉴别指定数据，该相互鉴别指定数据指定上述搭载有存储器的设备和输出权证的存取机器间应执行的相互鉴别形态；上述控制部件按照该存取控制权证的相互鉴别指定数据来执行相互鉴别，以鉴别成立为条件按照所接收权证的记录来执行处理。

再者，在本发明的搭载有存储器的设备的一实施形态中，其特征在于，上述存取控制权证包含权证验证指定数据，该权证验证指定数据指定上述搭载有存储器的设备接收到的存取控制权证的验证形态；上述控制部件按照该存取控制权证的权证验证指定数据来执行权证验证处理，以验证成立为条件按照所接收权证的记录来执行处理。

再者，在本发明的搭载有存储器的设备的一实施形态中，其特征在于，上述存取控制权证包含该存取控制权证的发行部件的范畴或标识符；上述控制部件根据从存取机器接收到的存取控制权证上记述的该存取控制权证的发行部件的范畴或标识符，来确认权证是合法的发行部件发行的权证，以该确认为条件按照所接收权证的记录来执行处理。

再者，在本发明的搭载有存储器的设备的一实施形态中，其特征在于，上述存取控制权证包含该存取控制权证的利用部件---存取机器的范畴或标识符；上述控制部件根据从存取机器接收到的存取控制权证上记述的该存取控制权证的利用部件---存取机器的范畴或标识符，来确认权证是合法的利用部件提供的权证，以该确认为条件按照所接收权证的记录来执行处理。

再者，本发明的第 5 侧面是一种存储器存取控制方法，用于具有保存数据文件的存储部的搭载有存储器的设备，其中，上述搭载有存储器的设备的存储部具有：1 个以上的分区区域，保存上述数据文件，作为由分区管理器管理的存储区域；以及设备管理器管理区域，由作为该搭载有存储器的设备的管理者的设备管理器管理；所述存储器存取控制方法包括以下步骤：

从存取机器接收上述设备管理器管理的存取控制权证、或上述分

区管理器管理的存取控制权证作为对上述存储部的存取控制权证；和

按照所接收权证的记述来执行处理。

再者，在本发明的存储器存取控制方法的一实施形态中，其特征在于，上述存取控制权证包含相互鉴别指定数据，该相互鉴别指定数据指定上述搭载有存储器的设备和输出权证的存取机器间应执行的相互鉴别形态；上述搭载有存储器的设备按照该存取控制权证的相互鉴别指定数据来执行相互鉴别，以鉴别成立为条件按照所接收权证的记录来执行处理。

再者，在本发明的存储器存取控制方法的一实施形态中，其特征在于，上述存取控制权证包含权证验证指定数据，该权证验证指定数据指定上述搭载有存储器的设备接收到的存取控制权证的验证形态；上述搭载有存储器的设备按照该存取控制权证的权证验证指定数据来执行权证验证处理，以验证成立为条件按照所接收权证的记录来执行处理。

再者，在本发明的存储器存取控制方法的一实施形态中，其特征在于，上述存取控制权证包含该存取控制权证的发行部件的范畴或标识符；上述搭载有存储器的设备根据从存取机器接收到的存取控制权证上记述的该存取控制权证的发行部件的范畴或标识符，来确认权证是合法的发行部件发行的权证，以该确认为条件按照所接收权证的记录来执行处理。

再者，在本发明的存储器存取控制方法的一实施形态中，其特征在于，上述存取控制权证包含该存取控制权证的利用部件---存取机器的范畴或标识符；上述搭载有存储器的设备根据从存取机器接收到的存取控制权证上记述的该存取控制权证的利用部件---存取机器的范畴或标识符，来确认权证是合法的利用部件提供的权证，以该确认为条件按照所接收权证的记录来执行处理。

再者，在本发明的存储器存取控制方法的一实施形态中，其特征在于，在上述设备管理器管理的存取控制权证中，包含容许对上述搭载有存储器的设备的存储部执行分区创建处理或删除处理的分区注册权证(PRT)；上述搭载有存储器的设备在从上述存取机器接收到分区注册权证(PRT)的情况下，根据所接收分区注册权证(PRT)的记录来执行分区创建处理或删除处理。

再者，在本发明的存储器存取控制方法的一实施形态中，其特征在于，上述分区注册权证(PRT)是从上述设备管理器管理的权证发行部件向上述分区管理器管理的作为权证用户的存取机器发行的。

再者，在本发明的存储器存取控制方法的一实施形态中，其特征在于，在上述分区管理器管理的存取控制权证中，包含容许在上述搭载有存储器的设备的存储部内创建的分区内执行数据文件创建处理或删除处理的文件注册权证(FRT)；上述搭载有存储器的设备在从上述存取机器接收到文件注册权证(FRT)的情况下，根据所接收文件注册权证(FRT)的记录来执行文件创建处理或删除处理。

再者，在本发明的存储器存取控制方法的一实施形态中，其特征在于，上述文件注册权证(FRT)是从上述分区管理器管理的权证发行部件向上述分区管理器管理的作为权证用户的存取机器发行的。

再者，在本发明的存储器存取控制方法的一实施形态中，其特征在于，在上述分区管理器管理的存取控制权证中，包含容许对上述搭载有存储器的设备的存储部内的分区内的数据文件执行存取的服务许可权证(SPT)；上述搭载有存储器的设备在从上述存取机器接收到服务许可权证(SPT)的情况下，根据所接收服务许可权证(SPT)的记录对数据文件执行存取处理。

再者，在本发明的存储器存取控制方法的一实施形态中，其特征在于，上述服务许可权证(SPT)是从上述分区管理器管理的权证发行部件向上述分区管理器管理的作为权证用户的存取机器发行的。

再者，在本发明的存储器存取控制方法的一实施形态中，其特征在于，在上述设备管理器或上述分区管理器管理的存取控制权证中，包含容许对上述搭载有存储器的设备的存储部内保存的数据执行更新处理的数据更新权证(DUT)；上述搭载有存储器的设备在从上述存取机器接收到数据更新权证(DUT)的情况下，根据所接收数据更新权证(DUT)的记录来执行数据更新处理。

再者，在本发明的存储器存取控制方法的一实施形态中，其特征在于，用于更新上述设备管理器管理的设备管理器管理区域的数据的数据更新权证(DUT)是从上述设备管理器管理的权证发行部件向上述设备管理器管理的作为权证用户的存取机器发行的；用于更新上述分区管理器管理的分区区域的数据的数据更新权证(DUT)是从上述分区

管理器管理的权证发行部件向上述分区管理器管理的作为权证用户的存取机器发行的。

再者，本发明的第 6 侧面是一种程序存储媒体，提供计算机程序以在计算机系统上对搭载有存储器的设备执行存储器存取控制处理，该搭载有存储器的设备具有的存储部具有：1 个以上的分区区域，保存数据文件，作为由分区管理器管理的存储区域；以及设备管理器管理区域，由作为该搭载有存储器的设备的管理者的设备管理器管理，其特征在于，上述计算机程序具有下述步骤：

从存取机器接收上述设备管理器管理的存取控制权证、或上述分区管理器管理的存取控制权证作为对上述存储部的存取控制权证；

与存取机器执行相互鉴别；

按照所接收权证的记述来执行权证验证处理；以及

按照所接收权证的记述来执行处理。

其中，本发明的程序存储媒体例如是以计算机可读的形式向可执行各种程序代码的通用计算机系统提供的媒体。媒体的形态没有特别的限定，例如可以是 CD 或 FD、MO 等记录媒体、可通信媒体等。

这种程序存储媒体定义了用于在计算机系统上实行规定的计算机程序的功能的、计算机程序和存储媒体之间的结构上或功能上的协同关系。换言之，通过经该存储媒体将计算机程序安装到计算机系统上，能够在计算机系统上发挥协同作用，得到与本发明的其他侧面同样的作用效果。

通过根据后述的本发明的实施例和附图进行的更详细的说明，本发明的其他目的、特征和优点将会变得更加清楚。其中，在本说明书中，系统是指多个装置的逻辑集合结构，各结构的装置不一定位于同一壳体内。

附图说明

图 1 是概要说明本发明的系统结构的系统结构示意图(其 1)。

图 2 是概要说明本发明的系统结构的系统结构示意图(其 2)。

图 3 是说明本发明的系统结构的具体例的系统结构示意图(其 3)。

图 4 是本发明的系统中的存取控制权证的发行部件及利用部件的

关系说明图。

图 5 是本发明的系统中的具有存储部的设备的结构图。

图 6 是本发明的设备的存储格式图。

图 7 是本发明的系统中的设备管理器的结构图。

图 8 是本发明的系统中的设备管理器的控制部件的结构图。

图 9 是本发明的系统中的分区管理器的结构图。

图 10 是本发明的系统中的读写器(R/W)的结构图。

图 11 是本发明的系统中可利用的公开密钥证书的格式说明图。

图 12 是本发明的系统中可利用的公开密钥体制的签名生成处理的流程图。

图 13 是本发明的系统中可利用的公开密钥体制的签名验证处理的流程图。

图 14 是本发明的设备中的存储部中保存的数据中的生产信息块的数据结构图。

图 15 是本发明的设备中的存储部中保存的数据中的设备管理信息块的数据结构图。

图 16 是本发明的设备中的存储部中保存的数据中的公开密钥类设备密钥定义块的数据结构图。

图 17 是本发明的设备中的存储部中保存的数据中的对称密钥类设备密钥定义块的数据结构图。

图 18 是本发明的设备中的存储部中保存的数据中的设备密钥区域的数据结构图。

图 19 是本发明的设备中的存储部中保存的数据中的分区定义块的数据结构图。

图 20 是本发明的设备中的存储部中保存的数据中的分区管理信息块的数据结构图。

图 21 是本发明的设备中的存储部中保存的数据中的公开密钥类分区密钥定义块的数据结构图。

图 22 是本发明的设备中的存储部中保存的数据中的对称密钥类分区密钥定义块的数据结构图。

图 23 是本发明的设备中的存储部中保存的数据中的分区密钥区域的数据结构图。

图 24 是本发明的设备中的存储部中保存的数据中的文件定义块的数据结构图。

图 25 是本发明的设备中的存储部中保存的数据中的文件的结构类型的说明图。

图 26 是本发明的系统中应用的作为存取控制权证的分区注册权证(PRT)的格式图。

图 27 是本发明的系统中应用的作为存取控制权证的文件注册权证(FRT)的格式图。

图 28 是本发明的系统中应用的作为存取控制权证的服务许可权证(SPT)的格式图(例 1)。

图 29 是利用本发明的系统中应用的作为存取控制权证的服务许可权证(SPT)的文件存取模式的类别的说明图。

图 30 是利用本发明的系统中应用的作为存取控制权证的服务许可权证(SPT)来存取的文件结构的说明图。

图 31 是本发明的系统中应用的作为存取控制权证的服务许可权证(SPT)的格式图(例 2)。

图 32 是本发明的系统中应用的作为存取控制权证的数据更新权证(DUT)的格式图。

图 33 是利用本发明的系统中应用的作为存取控制权证的数据更新权证(DUT)来更新的数据的说明图。

图 34 是本发明的系统中利用设备之前的处理的概略说明图。

图 35 是本发明的系统中的设备生产实体执行的设备初始注册处理的流程图。

图 36 是本发明的系统中的设备管理者执行的设备初始注册处理的流程图(其 1)。

图 37 是本发明的系统中的设备管理者执行的设备初始注册处理的流程图(其 2)。

图 38 是本发明的系统中的设备管理者执行的设备初始注册处理的流程图(其 3)。

图 39 是本发明的系统中的设备管理者执行的设备初始注册处理的流程图(其 4)。

图 40 是本发明的系统中的设备管理者执行的设备初始注册处理的

流程图(其 5)。

图 41 是本发明的系统中的设备管理器进行设备初始注册处理后的设备保存数据的说明图。

图 42 是本发明的系统中的设备管理器执行的公开密钥证书发行处理的流程图(其 1)。

图 43 是本发明的系统中的设备管理器执行的公开密钥证书发行处理的流程图(其 2)。

图 44 是本发明的系统中的设备管理器执行的公开密钥证书发行处理的说明图。

图 45 是本发明的系统中的设备管理器执行的公开密钥证书发行处理的说明图。

图 46 是本发明的系统中的设备管理器进行公开密钥证书发行处理后的设备保存数据的说明图。

图 47 是本发明的系统中对设备执行的分区创建、删除处理的流程图。

图 48 是说明本发明的系统中与设备执行的相互鉴别处理的流程(其 1)。

图 49 是说明本发明的系统中与设备执行的相互鉴别处理(设备鉴别)的流程(其 2)。

图 50 是本发明的系统中与设备执行的公开密钥体制的相互鉴别处理的说明图。

图 51 是本发明的系统中与设备进行相互鉴别处理后设备生成的鉴别表的结构说明图。

图 52 是本发明的系统中与设备进行相互鉴别处理后读写器生成的鉴别表的结构说明图。

图 53 是本发明的系统中与设备执行的对称密钥体制的相互鉴别处理的说明图。

图 54 是本发明的系统中与设备执行的对称密钥体制的相互鉴别处理的说明图。

图 55 是说明本发明的系统中与设备执行的相互鉴别处理(分区鉴别)的流程(其 3)。

图 56 是说明本发明的系统中与设备执行的相互鉴别处理(分区鉴

别)的流程(其4)。

图57是说明本发明的系统中的权证完整性、用户检查处理的流程(其1)。

图58是说明本发明的系统中的权证完整性、用户检查处理的流程(其2)。

图59是说明本发明的系统中的权证完整性可应用的MAC生成方式的流程(其1)。

图60是说明本发明的系统中的分区创建、删除操作的流程(其1)。

图61是说明本发明的系统中的分区创建、删除操作的流程(其2)。

图62是说明本发明的系统中的分区初始注册处理的流程(其1)。

图63是说明本发明的系统中的分区初始注册处理的流程(其2)。

图64是说明本发明的系统中的分区初始注册处理的流程(其3)。

图65是本发明的系统中的分区初始注册处理后的设备保存数据的说明图。

图66是本发明的系统中的分区管理器执行的公开密钥证书发行处理的说明图(其1)。

图67是本发明的系统中的分区管理器执行的公开密钥证书发行处理的说明图(其2)。

图68是本发明的系统中的分区管理器执行的分区创建处理中执行公开密钥体制鉴别、公开密钥体制权证验证的情况下的处理的说明图。

图69是本发明的系统中的分区管理器执行的分区创建处理中执行公开密钥体制鉴别、对称密钥体制权证验证的情况下的处理的说明图。

图70是本发明的系统中的分区管理器执行的分区创建处理中执行对称密钥体制鉴别、对称密钥体制权证验证的情况下的处理的说明图。

图71是本发明的系统中的分区管理器执行的分区创建处理中执行对称密钥体制鉴别、公开密钥体制权证验证的情况下的处理的说明图。

图 72 是说明本发明的系统中应用文件注册权证(FRT)的文件创建删除处理的流程图。

图 73 是说明本发明的系统中应用文件注册权证(FRT)的文件创建删除处理的流程图。

图 74 是说明本发明的系统中应用文件注册权证(FRT)来创建文件后的设备保存数据的说明图。

图 75 是本发明的系统中用文件注册权证(FRT)执行的文件创建处理中执行公开密钥体制鉴别、公开密钥体制权证验证的情况下的处理的说明图。

图 76 是本发明的系统中用文件注册权证(FRT)执行的文件创建处理中执行公开密钥体制鉴别、对称密钥体制权证验证的情况下的处理的说明图。

图 77 是本发明的系统中用文件注册权证(FRT)执行的文件创建处理中执行对称密钥体制鉴别、对称密钥体制权证验证的情况下的处理的说明图。

图 78 是本发明的系统中用文件注册权证(FRT)执行的文件创建处理中执行对称密钥体制鉴别、公开密钥体制权证验证的情况下的处理的说明图。

图 79 是本发明的系统中应用服务许可权证(SPT)的文件存取处理的流程图。

图 80 是本发明的系统中应用服务许可权证(SPT)的文件打开操作的流程图。

图 81 是本发明的系统中通过应用服务许可权证(SPT)的文件打开操作而生成的文件打开表的结构说明图(例 1)。

图 82 是本发明的系统中通过应用服务许可权证(SPT)的文件打开操作而生成的文件打开表的结构说明图(例 2)。

图 83 是本发明的系统中应用服务许可权证(SPT)的文件存取处理例的说明图(例 1)。

图 84 是本发明的系统中应用服务许可权证(SPT)的文件存取处理例的说明图(例 2)。

图 85 是本发明的系统中通过鉴别而生成的会话密钥的处理的说明图。

图 86 是说明本发明的系统中应用服务许可权证(SPT)的文件存取处理例的流程图(例 1)。

图 87 是说明本发明的系统中应用服务许可权证(SPT)的文件存取处理例的流程图(例 2)。

图 88 是本发明的系统中应用服务许可权证(SPT)的复合文件存取处理例的说明图。

图 89 是本发明的系统中用服务许可权证(SPT)执行的文件存取处理中执行公开密钥体制鉴别、公开密钥体制权证验证的情况下的处理的说明图。

图 90 是本发明的系统中用服务许可权证(SPT)执行的文件存取处理中执行公开密钥体制鉴别、对称密钥体制权证验证的情况下的处理的说明图。

图 91 是本发明的系统中用服务许可权证(SPT)执行的文件存取处理中执行对称密钥体制鉴别、对称密钥体制权证验证的情况下的处理的说明图。

图 92 是本发明的系统中用服务许可权证(SPT)执行的文件存取处理中执行对称密钥体制鉴别、公开密钥体制权证验证的情况下的处理的说明图。

图 93 是本发明的系统中用数据更新权证(DUT)执行的数据更新处理的流程图。

图 94 是本发明的系统中用数据更新权证(DUT)执行的数据更新操作的流程图。

图 95 是本发明的系统中用数据更新权证(DUT)执行的数据更新处理例的说明图。

图 96 是现有存储器结构图。

图 97 是现有存储器管理者、用户的关系的说明图。

图 98 是现有存储区域保留处理的说明图。

图 99 是现有存储器存取方式的说明图。

具体实施方式

以下，参照附图来详细说明本发明的实施形态。

其中，说明根据以下的项目来进行。

A. 关于利用设备的数据处理系统的构成实体及权证的说明

A1. 利用搭载有存储器的设备的数据管理系统概述

A2. 设备的结构

A3. 设备管理器的结构

A4. 分区管理器的结构

A5. 权证用户(作为设备存取机器的读写器)的结构

A6. 公开密钥证书

A7. 设备的存储部中的保存数据

A7. 1. 设备唯一信息及设备内分区信息区域

A7. 2. 分区区域

A8. 各权证的数据格式

A8. 1. 分区注册权证(PRT)

A8. 2. 文件注册权证(FRT)

A8. 3. 服务许可权证(SPT)

A8. 4. 数据更新权证(DUT)

B. 对用户发放设备、对设备进行各种设定、设备利用处理详述

B1. 从设备初始注册到利用的流程

B2. 设备生产实体执行的初始注册处理

B3. 设备管理器的管辖处理

B3. 1. 设备管理器执行的设备注册处理

B3. 2. 设备管理器管理下的公开密钥证书发行处理

B4. 分区管理器的管辖处理

B4. 1. 分区管理器管理下的利用分区注册权证(PRT)的分区设定注册、删除处理

B4. 2. 分区管理器管理下的公开密钥证书发行处理

B4. 3. 分区创建处理各方式中的处理过程

B4. 4. 利用文件注册权证(FRT)的文件创建、删除处理

B4. 5. 文件创建处理各方式中的处理过程

B4. 6. 利用服务许可权证(SPT)的服务(文件存取)处理

B4. 7. 利用服务许可权证(SPT)的存取处理各方式中的处理过程

B5. 利用数据更新权证(DUT)的设备数据更新处理

[A1. 利用搭载有存储器的设备的数据管理系统概述]

图 1 示出本发明的数据管理系统的概要说明图。搭载有存储器的设备(以下称为设备)100 由设备生产实体(manufacturer)500 生产，在作为设备管理实体的设备管理器(DM: Device Manager)200 的管理下提供给用户来利用。向用户提供设备的形态可以是租借或销售(包含转让)等中的任一形态。

设备 100 的存储区域被分割为多个作为数据保存区域的分区，各个分区(Partition A、B…Z)在作为各种服务主体(A、B、…Z)300A～300Z 的分区管理器的管理下被用于各种服务。

在对设备 100 执行的分区设定注册处理、设备中设定的分区内的文件设定注册处理、以及对注册的各文件执行的存取处理中，分别需要合法的权证发行部件(Ticket Issuer)发行的与设备对应的存取控制权证。

在对设备 100 执行的分区设定注册处理中，需要合法的权证发行部件(Ticket Issuer)发行的分区注册权证(PRT: Partition Registration Ticket)；在设备中设定的分区内执行的文件设定注册处理中，需要合法的权证发行部件(Ticket Issuer)发行的文件注册权证(FRT: File Registration Ticket)；而在对各文件执行的存取中，需要合法的权证发行部件(Ticket Issuer)发行的服务许可权证(SPT: Service Permission Ticket)。

在各权证上，保存有存取设备 100 的规则，除了例如关于对设备执行读写等各种处理的读写器和设备间的相互鉴别处理的规则之外，例如如果是分区注册权证(PRT)，则保存有能够设定的分区长度；如果是文件注册权证(FRT)，则保存有能够设定的文件长度；如果是服务许可权证(SPT)，则保存有可执行的存取形态(例如，数据读出、写入等)等，还保存有关于权证发行者、权证用户的信息和其他信息。此外，可记录检查这些权证保存数据有无窜改的 ICV (Integrity Check Value，完整性检查值)，以没有窜改权证为条件来执行权证上记录的范围内的处理。这些权证将在后面详述。

在图 1 所示的例子中，发行分区注册权证(PRT)的权证发行部件(Ticket Issuer)被设定在设备管理器(DM)200 内，在作为分区管理器的服务主体 A 300A 内设定发行文件注册权证(FRT)、及服务许可权证(SPT)的权证发行部件(Ticket Issuer)。其中在图 1 的结构中，服务

主体 B…Z 300B～300Z 也基本上具有与服务主体 A 同样的结构，在各服务主体中设定发行文件注册权证(FRT)、及服务许可权证(SPT)的权证发行部件(Ticket Issuer)。

其中，在图 1 中，示出服务主体和分区管理器(PM)为同一实体，但是这些实体并不一定要为同一实体，管理设备中设定的作为存储区域的分区的分区管理器、和在规定合同下从分区管理器借来分区管理器管理的存储区域---分区并在借来的分区内保存各种文件来提供服务的服务主体也可以作为不同的实体来存在。在以下说明中，为了简化说明，说明服务主体具有分区管理器的功能的结构例。

作为各服务主体 300A～300Z 的分区管理器(PM)例如在支付相应的价钱等规定的合同下，向设备管理器(DM)200进行分区注册权证(PRT)发行请求，在设备管理器(DM)的允许下，设备管理器(DM)内的权证发行部件(Ticket Issuer)向作为各服务主体的分区管理器(PM)发行分区注册权证(PRT)。

各服务主体(分区管理器(PM))300 经通信接口(I/F)对用户拥有的设备 100 执行存取，根据从设备管理器(DM)200 接收到的分区注册权证(PRT)上记录的规则来执行鉴别、验证等处理，而且执行分区注册权证(PRT)上记录的许可范围内的分区设定注册处理。该处理将在后面详细说明。

通信 I/F 不管是有线、无线，只要是可与外部设备进行数据通信的接口即可，例如，在设备具有 USB 连接结构的情况下由 USB I/F 构成；如果是 IC 卡型的则由 IC 卡读写器构成；而如果是公用线路、通信线路、因特网等具有各种通信功能的设备、或可连接在这些通信装置上的设备，则由符合各通信体制的数据通信 I/F 构成。

此外，在设备 100 中设定服务主体 300 的分区后，各服务主体 300 经通信接口(I/F)存取用户拥有的设备 100，根据各服务主体 300 的权证发行部件(Ticket Issuer)发行的文件注册权证(FRT)上记录的规则来执行鉴别、验证等处理，而且执行文件注册权证(FRT)上记录的许可范围内的文件设定注册处理。该处理将在后面详细说明。

再者，服务主体 300 经通信接口(I/F)存取用户拥有的设备 100，根据各服务主体的权证发行部件(Ticket Issuer)发行的服务许可权证(SPT)上记录的规则来执行鉴别、验证等处理，而且执行服务许可权

证(SPT)上记录的许可范围内的存取(例如，数据的读取、写入等)处理。该处理将在后面详细说明。

此外，如图 1 所示，在设备管理器 200、分区管理器 300 的上级设定代码管理机关 400，向各个设备管理器、分区管理器分配作为各实体的标识信息的代码。将授予这些管理器的代码作为前述分区注册权证(PRT)、文件注册权证(FRT)等存取控制权证的保存数据。

在设备 100 被提供给(例如，租借、销售给)用户、用户利用以前，设定管理所提供设备的设备管理器(DM)200，向该所提供设备内除了写入设备管理器代码外，还写入设备管理器的管理信息。这些数据将在后面详述。

用图 2 来说明本发明的利用存储设备的数据管理系统中的公开密钥证书发行处理和各实体的关系。

图 2 示出:设备管理器,作为设备管理实体;2 个分区管理器 300A、300B，作为设备中设定的各分区的管理实体；代码管理机关 400，向设备管理器 200 授予标识代码。还存在：设备管理器认证机构(CA(DEV)：Certificate Authority)610，按照来自设备管理器 200 管辖的注册机构 210 的公开密钥证书发行请求，发行与设备管理器 200、设备管理器管辖的各机器(分区注册权证(PRT)发行部件(PRT Issuer)210)、或设备 100 对应的设备公开密钥证书(CERT-DEV)；以及分区管理器认证机构(CA(PAR)：Certificate Authority)620、630,发行与分区管理器 300A、300B 管辖的各机器(文件注册权证(FRT)发行部件(FRT Issuer)310、服务许可权证(SPT)发行部件 320、权证用户---作为设备存取机器的读写器 711～714)、或设备 100 的分区对应的分区公开密钥证书(CERT-PAR)。

其中，图 2 示出认证机构分为设备管理器认证机构“DM 用 CA(Certificate Authority)(或 CA(DEV))”610、和分区管理器认证机构“PAR 用 CA (或 CA(PAR))”620、630 的结构，但是其结构是自由的，可以设置具有两种功能的单个认证机构，也可以分别设置与多个分区管理器对应的通用认证机构和设备管理器认证机构。

设备管理器 200、分区管理器 300A、300B 具有注册机构(RA:Registration Authority)220、230，该注册机构 220、230 受理设备管理器 200、分区管理器 300A、300B 的公开密钥证书、各管理器管理

的各机器(权证发行部件、权证用户)的公开密钥证书、或设备 100 的公开密钥证书发行请求，验证受理的发行请求，在验证后，将证书发行请求传送到认证机构，并且管理已发行的公开密钥证书。

经这些注册机构(RA)220、330 从各认证机构(CA)610、620、630 发行的公开密钥证书被保存到设备 100 中，用于对设备 100 的处理例如分区设定处理、或对分区的处理例如文件设定处理、以及对文件的存取处理等时的相互鉴别处理、或前述各权证的完整性验证处理。这些公开密钥证书的发行处理、使用公开密钥证书的各处理将在后面详述。

在图 2 中，作为分区，设备 100 具有分区管理器 1 300A 的管理分区“PM1 区域”、分区管理器 2 300B 的管理分区“PM2 区域”，还具有设备管理器 200 的管理区域“DM 区域”。

设备管理器 200 具有分区注册权证发行部件(PRT Issuer)210，而分区管理器 300 具有文件注册权证发行部件(FRT Issuer)310、及服务许可权证发行部件(SPT Issuer)320，分别发行各权证。

分区管理器 1 300A 具有因 PRT、FRT、SPT 各权证而异的专用读写器(对设备进行数据读出写入用的接口)711～713，而分区管理器 2 300B 具有各权证通用的读写器 714。读写器可这样采用各种各样的结构。

进而用图 3 来说明实体的具体例。在图 3 中，示出下述利用设备的结构例：作为利用设备中设定的分区来提供服务的服务主体的分区管理器有东西铁道株式会社及南北铁道株式会社这 2 个服务主体，对这些分区管理器进行分区设定注册的设备管理器是日本铁道集团这一组织。

东西铁道株式会社在用户的设备中设定的自身管理的分区“PM1”内设定注册多个文件。即，月票文件、预付文件、其他文件。作为各服务主体的分区管理器可以在按照自己提供的服务而设定的、由设备管理器分配的分区内注册各种文件。而在文件设定注册中需要文件注册权证(FRT)。

东西铁道株式会社具有管理设备的 1 个分区“PM1 区域”的分区管理器的功能。分区“PM1 区域”由作为设备管理器的日本铁道集团根据日本铁道集团的 PRT Issuer 发行的分区注册权证(PRT)上记录的规则

来执行鉴别、验证等处理，而且通过分区注册权证(PRT)上记录的许可范围内的分区设定注册处理来设定，授予东西铁道株式会社。

东西铁道株式会社在授予的分区“PM1 区域”上按照自身提供的服务来设定各种文件。例如是月票文件、预付文件，在月票文件内的数据保存区域中例如记录月票用户名、使用期限、利用区间等作为月票管理数据而必须的各种数据。此外，在预付文件中，记录用户名、预付金额、余额数据等。在该文件设定处理中，根据东西铁道的 FRT Issuer 发行的文件注册权证(FRT)上记录的规则来执行鉴别、验证等处理，而且通过文件注册权证(FRT)上记录的许可范围内的文件设定注册处理来设定。

这样设定了各种文件的设备由用户使用。例如，用户可使用设备，将设备放入包括作为设备存取机器的读写器的剪票机来利用。例如剪票机包括的合法的读写器存取月票文件，读取利用区间。此外，存取预付文件，执行预付文件内的余额数据的更新处理。

存取设备内的哪一个文件并执行哪种处理(读取、写入、减少等)，记录在东西铁道的服务许可权证(SPT)发行部件(SPT Issuer)发行的服务许可权证(SPT)中。例如在剪票机包括的作为合法的设备存取机器的读写器中保存有这些权证，根据权证上记录的规则来执行与设备间的鉴别处理、权证验证等处理。在作为设备存取机器的读写器及设备相互是合法的机器、使用权证合法的情况下，执行服务许可权证(SPT)上记录的许可范围内的处理(例如，文件内的数据读取、写入、减少等)。

发行分区注册权证(PRT)、文件注册权证(FRT)、及服务许可权证(SPT)各种权证的权证发行部件(Ticket Issuer)和利用权证发行部件发行的权证的权证用户(Ticket User)的一般对应关系示于图 4。

如图 1 等所述，权证发行部件(Ticket Issuer)处于设备管理器、或分区管理器的管理下，发行与对设备执行的处理对应的分区注册权证(PRT)、文件注册权证(FRT)、及服务许可权证(SPT)各种权证。权证用户(Ticket User)是利用权证发行部件发行的权证的机器、部件，具体地说，例如作为对设备执行数据写入、读取等处理的设备存取机器的读写器等机器即相当于此。

如图 4 所示，权证用户可保存并使用多个权证。此外，也可以只

保存单一权证、例如用图 3 说明过的只许可读取月票文件的区间数据的服务许可权证(SPT)，只读取区间数据。

例如，在某个服务主体(分区管理器)---铁道会社的月票只读剪票机中，设定作为设备存取机器的读写器，该读写器只保存只许可读取上述月票文件的区间数据的服务许可权证(SPT)，从用户拥有的设备中读取区间数据。例如，在执行月票、预付票两种处理的剪票机的作为设备存取机器的读写器中，也可一并保存只许可读取月票文件的区间数据的服务许可权证(SPT)、及许可预付文件的余额数据减少处理的服务许可权证(SPT)，可对月票文件、及预付文件两种文件执行处理。

此外，也可以构成下述权证用户(例如，读写器)，该权证用户保存分区注册权证(PRT)、文件注册权证(FRT)、及服务许可权证(SPT)，可执行分区注册、文件注册、文件内的数据存取等所有处理。这样，权证用户可执行的处理由权证用户可应用的权证来决定。

[A2. 设备的结构]

接着，说明具有将数据保存区域分割为上述多个分区的存储器的设备。在图 5 中示出设备的结构图。

如图 5 所示，设备 100 具有：CPU (Central Processing Unit，中央处理单元)101，具有程序执行功能、运算处理功能；通信接口 102，具有与作为设备存取机器的读写器等外部设备进行通信处理接口的功能；ROM (Read Only Memory，只读存储器)103，存储 CPU 101 执行的各种程序、例如加密处理程序等；RAM (Random Access Memory，随机存取存储器)104，用作执行程序的加载区域、和各程序处理中的工作区域；加密处理部 105，执行与外部设备的鉴别处理、电子签名的生成、验证处理、保存数据的加密、解密处理等加密处理；以及存储部 106，设定保存前述分区、文件，并且保存包含各种密钥数据的设备的唯一信息，例如由 EEPROM (Electrical1 Erasable Programmable ROM，电可擦可编程只读存储器)构成。存储部 106(例如，EEPROM)106 中保存的信息将在后面详述。

存储部 106 的数据保存结构示于图 6。存储部例如是称为 EEPROM (Electrical1 Erasable Programmable ROM)的电可改写非易失性存储器的一种形态---闪速存储器。

如图 6 所示，在本实施例中，具有每 1 块为 32 字节、块数为 0xFFFF

的数据保存区域，主要区域有分区区域、未使用区域、设备唯一信息及设备内分区信息区域。

在分区区域中，设定注册有作为前述分区管理器的管理区域的分区。其中，示出图 6 所示的存储器已经设定了分区的例子，而在新生产出来的设备中，未设定分区，不存在分区区域。如前所述，分区由作为各服务主体的分区管理器按照设备管理器管理的分区注册权证(PRT)发行部件(PRT Issuer)发行的 PRT 权证根据规定的过程、即分区注册权证(PRT)中设定的规则设定在设备内的存储器中。

在设备唯一信息及设备内分区信息区域中，保存有设备生产实体的信息、关于设备管理器的信息、设定分区信息、存取设备来执行分区设定注册处理时所需的密钥信息等。这些保存信息将在后面详述。其中，设备唯一信息区域的保存数据可用作与后述相互鉴别时应用的作为设备唯一值的 IDm 对应的数据。

此外，如图所示，分区区域还具有 1 个以上的文件区域、未使用区域、分区唯一信息及分区内文件区域。文件区域是作为分区管理器的服务主体例如保存前述对月票、预付票等每种服务设定的文件的区域。分区唯一信息及分区内文件信息区域例如保存关于分区内的文件的信息、文件存取处理所需的密钥信息等。这些保存信息将在后面详述。

[A3. 设备管理器的结构]

接着，用图 7 来说明设备管理器的结构。设备管理器是向用户提供(销售或租借)的设备的管理实体。

设备管理器 200 具有发行分区注册权证(PRT)的分区注册权证(PRT)发行部件(PRT Issuer)210，该分区注册权证(PRT)使得可按照来自分区管理器的请求对设备进行分区设定，该分区管理器是利用设备内的存储部的作为分割区域而设定的分区来提供服务的服务主体。

设备管理器 200 还发行与设备对应的设备公开密钥证书(CERT-DEV)。设备管理器 200 具有注册机构(RA: Registration Authority)220 的功能：受理来自设备的公开密钥证书发行请求，验证受理的发行请求，在验证后，将证书发行请求传送到认证机构(CA(DEV): Certificate Authority)610，并且管理已发行的公开密钥证书。

如图 7 所示，设备管理器 200 的分区注册权证(PRT)发行部件(PRT Issuer)210 具有控制部件 211 和数据库 212;作为分区注册权证(PRT)的发行管理数据，数据库 212 保存权证发行管理数据，例如将接受权证发行的分区管理器的标识符、权证标识符、权证用户(例如，读写器、PC 等)标识符等相对应的数据。

此外，注册机构(RA: Registration Authority)220 具有控制部 221、公开密钥证书发行管理数据库 222; 作为公开密钥证书发行管理数据，例如保存将发行了公开密钥证书的设备的标识符、公开密钥证书的标识符(序列号)等相对应的数据。

设备管理器 200 的分区注册权证(PRT)发行部件(PRT Issuer)210 的控制部件 211 通过与分区管理器的数据通信，来执行分区注册权证(PRT)的发行处理。而注册机构(RA: Registration Authority)220 的控制部件 221 对设备执行公开密钥证书发行处理，此时，执行与设备的通信、与设备管理器认证机构(CA(DEV))610 的通信。这些处理将在后面详述。这里，用图 8 来说明控制部件 211、221 的结构。

控制部件 211、221 都例如通过与作为数据处理部件的 PC 同样的结构来实现，具体地说，例如具有图 8 所示的结构。下面说明控制部件的结构。控制部 2111 由执行各种处理程序的中央处理单元(CPU: Central Processing Unit)构成。ROM (Read Only Memory)2112 是存储有加密处理程序等执行处理程序的存储器。RAM (Random Access Memory)2113 被用作控制部 2111 执行的程序、例如数据库管理程序、加密处理程序、通信程序等执行程序的保存区域、和这些程序处理中的工作区域。

显示部 2114 具有液晶显示器、CRT 等显示部件，在控制部 2111 的控制下，显示执行各种程序时的数据、例如待处理的数据内容等。输入部 2115 具有键盘、或例如鼠标等点击设备，将来自这些输入设备的命令、输入数据输出到控制部 2111。HDD (Hard Disk Drive，硬盘驱动器)2116 保存数据库管理程序、加密处理程序、通信程序等程序、以及各种数据。

驱动器 2117 具有控制对下述等各种记录媒体的存取的功能：例如 HD (Hard Disk，硬盘)、FD (Floppy Disk，软盘)等磁盘; CD-ROM (Compact Disk ROM，光盘只读存储器)等光盘; 小盘(ミニディスク)

等磁光盘；ROM或闪速存储器等半导体存储器。磁盘等各种记录媒体存储程序、数据等。通信接口2118用作经网络、电缆连接、电话线路等进行有线、无线通信的接口，用作与用户的设备、分区管理器、认证机构等各实体进行的通信接口。

[A4. 分区管理器的结构]

接着，用图9来说明分区管理器的结构。分区管理器是向用户提供(销售或租借)的设备中设定的分区的管理实体。

分区管理器300使用设备管理器授予的分区注册权证(PRT)，根据授予的PRT上记录的规则，在用户的设备内的存储部中设定分区作为分割区域，利用设定的分区来提供服务。

在设定的分区中可设定与服务、数据对应的文件。其中，文件设定处理需要取得文件注册权证(FRT)；而文件内的数据的读出、写入等数据存取需要取得服务许可权证(SPT)。文件设定、数据存取处理由权证用户、即具体地说例如作为专用的设备存取机器的读写器等使用权证来执行。

分区管理器300具有作为向这种权证用户发行权证的处理部件的文件注册权证(FRT)发行部件(FRT Issuer)310、及服务许可权证(SPT)发行部件(SPT Issuer)320。

分区管理器300还发行与设备的各分区对应的分区公开密钥证书(CERT-PAR)。分区管理器300具有注册机构(RA: Registration Authority)330的功能：受理来自设备的公开密钥证书发行请求，验证受理的发行请求，在验证后，将证书发行请求传送到认证机构(CA(PAR): Certificate Authority)620，并且管理已发行的公开密钥证书。

如图9所示，分区管理器300的文件注册权证(FRT)发行部件(FRT Issuer)310具有控制部件311和数据库312，作为文件注册权证(FRT)的发行管理数据，数据库312保存权证发行管理数据、例如将接受权证发行的权证用户(例如，读写器、PC等)的标识符、权证标识符等相对应的数据。

再者，分区管理器300的服务许可权证(SPT)发行部件(SPT Issuer)320具有控制部件321、和数据库322，作为服务许可权证(SPT)的发行管理数据，数据库322保存权证发行管理数据、例如将接受权

证发行的权证用户(例如，作为设备存取机器的读写器、PC 等)的标识符、权证标识符等相对应的数据。

此外，注册机构(RA: Registration Authority)330 具有公开密钥证书发行管理数据库 332，作为公开密钥证书发行管理数据，例如保存将发行了公开密钥证书的设备的标识符、分区标识符、公开密钥证书的标识符(序列号)等相对应的数据。

设备管理器 300 的文件注册权证(FRT)发行部件(FRT Issuer)310 的控制部件 311 通过与权证用户(例如，作为设备存取机器的读写器、PC 等)的数据通信，来执行文件注册权证(FRT)发行处理，而服务许可权证(SPT)发行部件(Ticket Issuer)320 的控制部件 321 通过与权证用户(例如，作为设备存取机器的读写器、PC 等)的数据通信，来执行服务许可权证(SPT)发行处理。此外，注册机构(RA: Registration Authority)330 的控制部件 331 对设备执行公开密钥证书发行处理，此时，执行与设备的通信、与分区管理器认证机构(CA(PAR))620 的通信。这些处理将在后面详述。

其中，分区管理器 300 的控制部件 311、321、331 的结构是与用图 8 说明过的前述设备管理器中的控制部件同样的结构，所以省略其说明。

[A5. 权证用户(作为设备存取机器的读写器)的结构]

作为设备存取机器的读写器是对设备执行分区的设定、文件的设定、数据的读取、写入、金额数据的减少、增加等各种处理的机器。对设备的处理根据处理时应用的分区注册权证(PRT)、文件注册权证(FRT)、或服务许可权证(SPT)上记录的规则来进行。即，对设备执行的所有处理由这些应用的权证来限制。

作为设备存取机器的读写器的结构例示于图 10。如图 10 所示，读写器 700 具有：CPU (Central Processing Unit)701，具有程序执行功能、运算处理功能;通信接口 702,具有与设备、权证发行部件(Ticket Issuer)等外部设备进行通信处理接口的功能；ROM (Read Only Memory)703，存储 CPU 701 执行的各种程序、例如加密处理程序等；RAM (Random Access Memory)704，用作执行程序的加载区域、和各程序处理中的工作区域；加密处理部 705，执行与外部设备的鉴别处理、电子签名的生成、验证处理、保存数据的加密、解密处理等加密

处理；以及存储部 706，保存鉴别处理、加密、解密处理用的各种密钥数据、及读写器的唯一信息，例如由 EEPROM (Electricall Erasable Programmable ROM)构成。

[A6. 公开密钥证书]

在本发明的具有分区分割存储区域的设备的利用中，在各实体、权证发行部件、权证用户、设备等相互间的数据通信中，在确认数据发送端和数据接收端相互是合法的数据发送接收对象后，传送所需的数据，作为这种数据传送时实现安全性结构的手法，应用传送数据加密处理、对数据执行的签名生成、验证处理。

加密数据可以通过基于规定过程的解密处理来还原为可利用的解密数据(平文)。在这种信息的加密处理中使用加密密钥、在解密处理中使用解密密钥的数据加密、解密方法以往已为人熟知。

使用加密密钥和解密密钥的数据加密-解密方法的形态各种各样，作为其中的一例，有所谓的公开密钥加密体制。公开密钥加密体制使发送者和接收者的密钥不同，一个密钥采用可由不特定的用户使用的公开密钥，另一个采用保密的私有密钥。例如，数据加密密钥采用公开密钥，解密密钥采用私有密钥。或者，以下述等形态来使用：鉴别符生成密钥采用私有密钥，鉴别符验证密钥采用公开密钥。

与加密、解密使用同一密钥的所谓的对称密钥加密体制不同，在公开密钥加密体制中，需要保密的私有密钥由特定的 1 个用户持有即可，所以在密钥的管理中很有利。但是，公开密钥加密体制与对称密钥加密体制相比，数据处理速度慢，多用于私有密钥的配送、数字签名等数据量少的对象。公开密钥加密体制的代表性的体制有 RSA (Rivest-Shamir-Adleman)加密。它使用 2 个非常大的素数(例如 150 位)之积，利用了 2 个大素数(例如 150 位)之积难以进行素因子分解处理的特点。

在公开密钥加密体制中，使得不特定多数用户可使用公开密钥，使用证明发放的公开密钥是否合法的证书、即所谓的公开密钥证书的方法用得较多。例如是下述系统。用户 A 生成公开密钥、私有密钥对，将生成的公开密钥送至认证机构，从认证机构得到公开密钥证书。用户 A 将公开密钥证书向公众公开。不特定的用户由公开密钥证书经规定的过程得到公开密钥来对文档等进行加密并送至用户 A。用户 A 用私

有密钥对加密文档等进行解密等。此外，用户 A 用私有密钥在文档等上附加签名，不特定的用户由公开密钥证书经规定的过程得到公开密钥，来进行该签名的验证。

公开密钥证书是公开密钥加密体制中认证机构(CA: Certificate Authority)发行的证书，是如下来创建的证书：用户将自己的 ID、公开密钥等提交给认证机构，认证机构一侧附加认证机构的 ID 或有效期等信息，还附加认证机构的签名。

图 11 概略示出公开密钥证书的格式。下面概略说明各数据。

证书版本(version)号表示公开密钥证书格式的版本。

证书序列号(SN: Serial Number)是公开密钥证书发行机构(认证机构：CA)设定的公开密钥证书的序列号。

签名算法标识符字段(Signature algorithm Identifier)的签名算法(algorithm)、算法参数(parameters)是记录公开密钥证书的签名算法和其参数的字段。其中，签名算法有椭圆曲线加密及 RSA，在应用椭圆曲线加密的情况下记录参数及密钥长度，而在应用 RSA 的情况下记录密钥长度。

发行机构(认证机构：CA)名是以可识别的形式(Distinguished Name)来记录公开密钥证书的发行者、即公开密钥证书发行机构(CA)的名称(Issuer)的字段。

证书有效期(validity)记录证书的有效期---开始日期时间、结束日期时间。

公开密钥证书用户名(Subject)记录用户---待鉴别者的标识数据。具体地说，例如记录用户机器的 ID、或服务提供主体的 ID 等标识符或范畴。

用户公开密钥字段(subject Public Key Info)的密钥算法(algorithm)和密钥(subject Public key)是保存作为用户的公开密钥信息的密钥算法、密钥信息本身的字段。

在选项区域中，记录用户的属性数据、其他伴随公开密钥证书的发行、利用的任选数据。作为属性数据，记录作为用户所属组信息的设备管理器代码(DMC)、分区管理器代码(PMC)。这里，用户是公开密钥证书的用户，例如是设备管理器、分区管理器、权证用户、权证发行部件、设备等。

在选项区域中，作为范畴信息，还记录权证用户、权证发行部件、设备、设备管理器、分区管理器等表示实体、机器类别的范畴。

其中，在设备管理器兼作分区注册权证发行部件(PRT Issuer)的情况下，可将后述分区注册权证发行部件代码(PRTIC: PRT Issuer Code)设定为设备管理器代码(DMC)；而在分区管理器兼作文件注册权证发行部件、服务许可权证发行部件的情况下，可将文件注册权证发行部件代码(FRTIC: FRT Issuer Code)、服务许可权证发行部件代码(SPTIC: SPT Issuer Code)设定为分区管理器代码(PMC)。其中，这些代码也被记录在后述各权证(PRT、FRT、SPT等)中。

此外，也可以向各权证发行部件分配与设备管理器代码(DMC)、分区管理器代码(PMC)不同的独自的代码。此情况下的代码授予由前述代码管理机关执行。

发行机构签名是公开密钥证书发行机构(CA)用私有密钥对公开密钥证书的数据执行的电子签名，公开密钥证书的用户用公开密钥证书发行机构(CA)的公开密钥来进行验证，可检查公开密钥证书有无窜改。

用图12来说明使用公开密钥加密体制的电子签名的生成方法。图12所示的处理是使用EC-DSA ((Elliptic Curve Digital Signature Algorithm，椭圆曲线数字签名算法)，IEEE P1363/D3)的电子签名数据的生成处理流程。其中，这里，说明公开密钥加密采用椭圆曲线加密(Elliptic Curve Cryptography(以下，称为ECC))的例子。其中，在本发明的数据处理装置中，除了椭圆曲线加密以外，也可以采用同样的公开密钥加密体制的例如RSA加密((Rivest、Shamir、Adleman)等(ANSI X9.31))。

下面说明图12的各步骤。在步骤S1中，设p为特征值，a、b为椭圆曲线的系数(椭圆曲线：$y^2=x^3+ax+b$，$4a^3+27b^2\neq 0 \pmod p$)，G为椭圆曲线上的基点，r为G的序号，Ks为私有密钥($0<Ks<r$)。在步骤S2中，计算报文M的散列值，设f=Hash(M)。

这里，说明用散列函数来求散列值的方法。散列函数是下述函数：以报文作为输入，将其压缩为规定比特长的数据，作为散列值来输出。散列函数具有下述特征：难以由散列值(输出)来预测输入，在输入到散列函数中的数据的1个比特变化时，散列值的多个比特变化，并且

难以找出具有同一散列值的不同的输入数据。散列函数有时采用 MD4、MD5、SHA-1 等，也有时采用 DES-CBC。在此情况下，作为最终输出值的 MAC(相当于检查值“ICV”)成为散列值。

接着，在步骤 S3 中，生成随机数 u (0<u<r)，在步骤 S4 中计算基点的 u 倍的坐标 V (X_V, Y_V)。其中，椭圆曲线上的加法、乘 2 如下定义。

设 P=(Xa, Ya)，Q=(Xb, Yb)，R=(Xc, Yc)=P+Q，则

在 P≠Q 时(加法)，

$Xc=\lambda^2-Xa-Xb$

$Yc=\lambda\times(Xa-Xc)-Ya$

$\lambda=(Yb-Ya)/(Xb-Xa)$

在 P=Q 时(乘 2)，

$Xc=\lambda^2-2Xa$

$Yc=\lambda\times(Xa-Xc)-Ya$

$\lambda=(3(Xa)^2+a)/(2Ya)$

用它们来计算点 G 的 u 倍(速度虽慢、但最易理解的运算方法如下所述进行。计算 G、2×G、4×G···，将 u 展开为二进制数，将与值为 1 的位对应的 $2^i\times G$(对 G 进行 i 次乘 2 所得的值(i 是从 u 的 LSB 数时的比特位置))相加。

在步骤 S5 中，计算 $c=X_V \bmod r$，在步骤 S6 中，判定该值是否为 0，如果不为 0，则在步骤 S7 中计算 $d=[(f+cKs)/u] \bmod r$，在步骤 S8 中判定 d 是否为 0，如果 d 不为 0，则在步骤 S9 中将 c 及 d 作为电子签名数据来输出。假定 r 的长度为 160 比特，则电子签名数据的长度为 320 比特。

在步骤 S6 中，在 c 为 0 的情况下，返回到步骤 S3 来重新生成新的随机数。同样，在步骤 S8 中 d 为 0 的情况下，也返回到步骤 S8 来重新生成随机数。

接着，用图 13 来说明使用公开密钥加密体制的电子签名的验证方法。在步骤 S11 中，设 M 为报文，p 为特征值，a、b 为椭圆曲线的系数(椭圆曲线：$y^2=x^3+ax+b$，$4a^3+27b^2\neq 0 \pmod p$)，G 为椭圆曲线上的基点，r 为 G 的序号，G 及 Ks×G 为公开密钥(0<Ks<r)。在步骤 12 中验证电子签名数据 c 及 d 是否满足 0<c<r、0<d<r。在满足它的条件下，

在步骤S13中，计算报文的散列值，设f=Hash(M)。接着，在步骤S14中计算h=1/d mod r，在步骤S15中计算h1=fh mod r、h2=ch mod r。

在步骤S16中，用已经计算出的h1及h2，来计算点P=(X_P, Y_P)=h1×G+h2·Ks×G。电子签名验证者知道基点G及Ks×G，所以可以与图12的步骤S4同样来计算椭圆曲线上的点的标量倍。然后，在步骤S17中判定点P是否为无穷远点，如果不是无穷远点则进至步骤S18(实际上，在步骤S16中就能进行无穷远点的判定。即，将P=(X, Y)、Q=(X, -Y)相加，则判明不能计算λ，P+Q为无穷远点。)在步骤S18中计算X_P mod r，与电子签名数据c进行比较。最后，在该值一致的情况下，进至步骤S19，判定为电子签名正确。

在判定为电子签名正确的情况下，知道数据未被窜改，持有与公开密钥对应的私有密钥的人生成了电子签名。

在步骤S12中，在电子签名数据c或d不满足0<c<r、0<d<r的情况下，进至步骤S20。此外，在步骤S17中，在点P为无穷远点的情况下也进至步骤S20。再者，在步骤S18中，在X_P mod r的值与电子签名数据c不一致的情况下也进至步骤S20。

在步骤S20中，在判定为电子签名不正确的情况下，知道数据已被窜改，不是持有与公开密钥对应的私有密钥的人生成了电子签名。

本发明的系统中的设备将经设备管理器的管理注册机构对设备发行的设备公开密钥证书(CERT-DEV)保存到设备中，还将经分区管理器的管理注册机构对设备的分区发行的分区公开密钥证书(CERT-PAR)保存到设备的各分区中。在对设备的处理、即应用分区注册权证(PRT)的分区注册设定处理、应用文件注册权证(FRT)的文件注册设定处理、以及应用服务许可权证(SPT)的数据处理中，这些公开密钥证书被应用于权证用户(例如，作为设备存取机器的读写器)和设备间的相互鉴别、签名生成、验证处理等。后面将用流程来说明这些处理的具体例。

[A7. 设备的存储部中的保存数据]

接着，说明本发明的具有分割为分区的存储区域的设备的保存数据。如前面用图6说明过的那样，设备具有例如由EEPROM构成的存储部，主要区域有分区区域、未使用区域、设备唯一信息及设备内分区信息区域。以下，参照附图来依次说明这些区域的保存数据。

[A7. 1. 设备唯一信息及设备内分区信息区域]

首先，说明设备唯一信息及设备内分区信息区域的各数据。在设备唯一信息及设备内分区信息区域中，保存设备生产实体的信息、关于设备管理器的信息、设定分区信息、存取设备来执行分区设定注册处理时所需的密钥信息等。

图 14 示出生产信息块(Manufacture Information Block)的数据结构。各区域的数值表示字节数。如用图 6 说明过的那样，在本实施例的结构中，每 1 块为 32 字节。其中，图中灰色部分可以是加密过的数据，也可以是未加密过的数据。

在生产信息块(Manufacture Information Block)中，保存以下数据。

* Writable Flag: 是否许可写入到块中的判别标志(例如，0xffff: 许可写入，其他: 不可写入)

* Manufacture Code: 卡生产厂家标识号

* Manufacture Equipment Code: 卡生产线号

* Manufacture Date: 卡生产日期。例如，设 2001 年 1 月 1 日为 0x0000

* Manufacture Serial Number: 卡生产序列号

* Device Vender Code: IC 芯片供应公司号

* Device Code: IC 芯片的型号

* Device Parameter: 其他参数

这些块中写入的信息是唯一的，所以根据这些信息将设备(Device)唯一标识符定义为 IDm。其中，设备(Device)唯一标识符可以由生产信息块(Manufacture Information Block)中写入的全部信息、写入的部分信息、或根据写入的信息而取得的运算数据来取得。

图 15 示出设备管理信息块(Device Management Information Block)的数据结构。在设备管理信息块(Device Management Information Block)中保存以下数据。

* Writable Flag: 是否许可写入到块中的判别标志(例如，0xffff: 许可写入，其他: 不可写入)

* DMC (Device Manager Code): 设备管理器(DM: Device Manager)的标识号

* DMC Version: 设备管理器代码(DMC)的版本。例如，被用作更

新 DMC 时的比较条件。

* Total Block Number in Device: 设备(Device)内的总块数

* Free Block Number in Device: 设备(Device)内的空闲块数

* Partition Number: 当前已注册的分区(Partition)数

* Pointer of Free Area: 空闲区域的指针

图 16 示出公开密钥类设备密钥定义块(Device Key Definition Block(PUB))的数据结构。在公开密钥类设备密钥定义块(Device Key Definition Block(PUB))中保存以下数据。

* PUB_CA(DEV) Pointer: 指向保存有设备管理器认证机构(CA(DEV))的公开密钥的块的指针，该设备管理器认证机构(CA(DEV))经设备管理器管辖的注册机构来发行公开密钥证书

* PUB_CA(DEV) Size: 认证机构 CA(DEV)的公开密钥的长度

* PRI_DEV Pointer: 指向保存有设备(Device)的私有密钥的块的指针

* PRI_DEV Size: 设备(Device)的私有密钥的长度

* PARAM_DEV Pointer: 指向保存有设备(Device)的公开密钥参数的块的指针

* PARAM_DEV Size: 设备(Device)的公开密钥参数的长度

* CERT_DEV Pointer: 指向保存有认证机构 CA(DEV)发行的设备(Device)的公开密钥证书的块的指针

* CERT_DEV Size: 认证机构 CA(DEV)发行的设备(Device)的公开密钥证书的长度

* CRL_DEV Pointer: 指向保存有设备(Device)的作废表(Revocation List)的块的指针

* CRL_DEV Size: 设备(Device)的作废表(Revocation List)的长度

* PRTIC (PRT Issuer Category): 分区注册权证(PRT)发行者范畴

* PRTIC Version: 分区注册权证(PRT)发行者范畴(PRTIC)的版本

* DUTIC_DEV (DUT Issuer Category): 数据更新权证(DUT: Data Update Ticket)发行者范畴

* DUTIC_DEV Version: 数据更新权证(DUT: Data Update Ticket)发行者的版本

其中，上述数据中的作废表是非法设备的列表，例如是设备流通系统的管理者发行的设备作废表，是非法设备的标识数据的列表化数据。在作为设备存取机器的读写器中装入的设备是作废表中记载的设备的情况下，采取停止处理等措施。

例如经常向对设备执行处理的所有作为设备存取机器的读写器发放最新的作废表，在对设备执行处理时参照列表来判定执行还是停止处理。或者通过作为设备存取机器的读写器的通信功能经网络浏览最新的作废表来取得列表中记载的非法设备信息，判定执行还是停止处理。后面将用流程来说明利用作废表的具体处理。

此外，上述数据中的数据更新权证(DUT: Data Update Ticket)是用于在执行设备中保存的各种数据的更新处理时许可、限制更新处理的存取限制权证，与前述 PRT、FRT、SPT 等各权证同样，是记录有存取设备的规则的权证。后面将更详细地说明该数据更新权证(DUT: Data Update Ticket)。

图 17 示出对称密钥类设备密钥定义块(Device Key Definition Block (Common))的数据结构。在对称密钥类设备密钥定义块(Device Key Definition Block (Common))中保存以下数据。

* Mkauth_DEV_A Pointer: 双向个别密钥鉴别主密钥(MKauth_DEV_A)的指针

* Mkauth_DEV_A Size: 双向个别密钥鉴别主密钥(MKauth_DEV_A)的长度

* Kauth_DEV_B Pointer: 双向个别密钥鉴别密钥(Kauth_DEV_B)的指针

* Kauth_DEV_B Size: 双向个别密钥鉴别密钥(Kauth_DEV_B)的长度

* Kprt Pointer: 指向保存有分区注册权证(PRT)的 MAC 验证密钥(Kprt)的块的指针

* Kprt Size: 分区注册权证(PRT)的 MAC 验证密钥(Kprt)的长度

* Kdut_DEV1-4 Pointer: 指向保存有数据更新权证(DUT)的 MAC 验证密钥(Kdut)的块的指针

* Kdut_DEV1-4 Size: 数据更新权证(DUT)的MAC验证密钥(Kdut)的长度

* IRL_DEV Pointer: 指向作为设备(Device)的作废表(Revocation List)、保存有非法设备的设备ID (Device ID)的块的指针

* IRL_DEV Size: 设备(Device)的作废表(Revocation List)的长度

后面将详细说明上述数据中所示的双向个别密钥鉴别的方法、MAC(Message Authenticate Code)验证处理.

此外, Kdut_DEV 有 4 种, 成对使用(Kdut_DEV1, Kdut_DEV2)、(Kdut_DEV3, Kdut_DEV4). 例如, Kdut_DEV1、3 用于生成 MAC, Kdut_DEV2、4 用于加密.

图 18 示出设备密钥区域(Device Key Area)的数据结构. 在设备密钥区域(Device Key Area)中保存以下数据. 其中, 在设备密钥区域(Device Key Area)的各保存密钥中, 一并保存版本信息. 在更新密钥时, 也一并更新版本.

* IRL_DEV: 注册有作废设备(Device)、作废机器(作为设备存取机器的读写器、PC 等权证用户、权证发行部件)的标识符(ID)的作废表(Revocation List (Device ID))

* CRL_DEV: 注册有作废设备(Device)、作废机器(作为设备存取机器的读写器、PC 等权证用户、权证发行部件)的公开密钥证书标识符(例如, 序列号: SN)的作废表(Revocation List (Certificate))

* Kdut_DEV1: 数据更新权证(DUT)的MAC验证密钥

* Kdut_DEV2: 数据更新加密密钥

* Kdut_DEV3: 数据更新权证(DUT)的MAC验证密钥

* Kdut_DEV4: 数据更新加密密钥

* Kprt: 分区注册权证(PRT)的MAC验证密钥

* CERT_DEV: 发行设备管理器公开密钥的认证机构 CA(DEV)发行的设备(Device)的公开密钥证书

* PRI_DEV: 设备(Device)的私有密钥

* PARAM_DEV: 设备(Device)的公开密钥参数

* PUB_CA(DEV): 发行设备管理器公开密钥的认证机构 CA(DEV)

的公开密钥

* Kauth_DEV_B: 双向个别密钥鉴别对称密钥

* MKauth_DEV_A: 双向个别密钥鉴别主密钥

其中，在图示的设备密钥区域(Device Key Area)中保存有Kauth_DEV_A“双向个别密钥鉴别对称密钥”、MKauth_DEV_B“双向个别密钥鉴别主密钥”，但是在设备没有进行对称密钥鉴别处理的请求的情况下，也可以不保存这些密钥。此外，在设备不执行权证验证处理的情况下，也可以不保存 Kprt“分区注册权证(PRT)的 MAC 验证密钥”。

此外，在不存在作废的设备的情况下，或者在使用其他来源来取得作废表的情况下，也可以不保存 IRL_DEV“注册有作废设备(Device)的设备标识符(ID)的作废表(Revocation List(Device ID))”、CRL_DEV“注册有作废设备(Device)的公开密钥证书标识符(例如，序列号: SN)的作废表(Revocation List(Certificate))”。

图 19 示出分区定义块(Partition Definition Block)的数据结构。在分区定义块(Partition Definition Block)中保存以下数据。

* PMC (Partition Manager Code): 分配给分区管理器(Partition Manager)的代码(PMC)。例如号码。

* PMC Version: 分区管理器代码(PMC)的版本

* Partition Start Position: 分区(Partition)保存起始地址

* Partition Size: 分区(Partition)的长度

以上是设备的存储部的设备唯一信息及设备内分区信息区域的各数据。

[A7. 2. 分区区域]

接着，说明分区区域的各数据。分区区域是分区管理器的管理区域。如前所述，由作为各服务主体的分区管理器按照设备管理器管理的分区注册权证(PRT)发行部件(PRT Issuer)发行的 PRT 权证根据规定的过程、即分区注册权证(PRT)中设定的规则设定在设备内的存储器中。以下，说明分区区域的数据结构。

图 20 示出分区管理信息块(Partition Management Information Block)的数据结构。在分区管理信息块(Partition Management Information Block)中保存以下数据。

* PMC (Partition Manager Code): 分区(Partition)拥有者的号码

* PMC Version: 分区管理器代码(PMC)的版本

* Total Block Number in Partition: 分区(Partition)内的总块数

* Free Block Number in Partition: 分区(Partition)内的空闲块数

* Pointer of Free Area: 分区(Partition)内的未使用区域的指针

* File Number: 分区中当前注册的文件(File)数

图 21 示出公开密钥类分区密钥信息块(Partition Key Definition Block(PUB))的数据结构。在公开密钥类分区密钥信息块(Partition Key Definition Block(PUB))中保存以下数据。

* PUB_CA(PAR) Pointer: 指向保存有认证机构 CA(PAR)的公开密钥的块的指针，该认证机构 CA(PAR)经分区管理器管辖的注册机构来发行公开密钥证书

* PUB_CA(PAR) Size: 认证机构CA(PAR)的公开密钥的长度

* PRI_PAR Pointer: 指向保存有分区(Partition)的私有密钥的块的指针

* PRI_PAR Size: 分区(Partition)的私有密钥的长度

* PARAM_PAR Pointer: 指向保存有分区(Partition)的公开密钥参数的块的指针

* PARAM_PAR Size: 分区(Partition)的公开密钥参数的长度

* CERT_PAR Pointer: 指向保存有认证机构 CA(PAR)发行的分区(Partition)的公开密钥证书的块的指针

* CERT_PAR Size: 认证机构CA(PAR)发行的分区(Partition)的公开密钥证书的长度

* CRL_PAR Pointer: 指向保存有分区(Partition)的作废表(Revocation List)的块的指针

* CRL_PAR Size: 分区(Partition)的作废表(Revocation List)的长度

* FRTIC (FRT Issuer Category): 文件注册权证(FRT)发行者范

畴

* FRTIC Version: 文件注册权证(FRT)发行者范畴(FRTIC)的版本

* DUTIC_PAR (DUT Issuer Category): 数据更新权证(DUT)发行者范畴

* DUTIC_PAR Version: 数据更新权证(DUT)发行者范畴(DUTIC)的版本

图 22 示出对称密钥类分区密钥信息块(Partition Key Definition Block (Common))的数据结构. 在对称密钥类分区密钥信息块(Partition Key Definition Block (Common))中保存以下数据.

* Mkauth_PAR_A Pointer: 双向个别密钥鉴别主密钥(MKauth_PAR_A)的指针

* Mkauth_PAR_A Size: 双向个别密钥鉴别主密钥(MKauth_PAR_A)的长度

* Kauth_PAR_B Pointer: 双向个别密钥鉴别密钥(Kauth_PAR_B)的指针

* Kauth_PAR_B Size: 双向个别密钥鉴别密钥(Kauth_PAR_B)的长度

* Kfrt Pointer: 指向保存有文件注册权证(FRT)的 MAC 验证密钥(Kfrt)的块的指针

* Kfrt Size: 文件注册权证(FRT)的 MAC 验证密钥(Kfrt)的长度

* Kdut_PAR1-4 Pointer: 指向保存有数据更新权证(DUT)的 MAC 验证密钥(Kdut)的块的指针

* Kdut_PAR1-4 Size: 数据更新权证(DUT)的 MAC 验证密钥(Kdut)的长度

* IRL_PAR Pointer: 指向保存有作废表(Revocation List-Device ID)的块的指针, 该作废表(Revocation List-Device ID)保存分区(Partition)的作废设备的 ID

* IRL_PAR Size: 分区(Partition)的作废表(Revocation List-Device ID)的长度

后面将详细说明上述数据中所示的双向个别密钥鉴别的方法、MAC (Message Authenticate Code)验证处理.

此外，Kdut_PAR 有 4 种，成对使用(Kdut_PAR1，Kdut_PAR2)、(Kdut_PAR3，Kdut_PAR4)。例如，Kdut_PAR1、3 用于生成 MAC，Kdut_PAR2、4 用于加密。

图 23 示出分区密钥区域(Partition Key Area)的数据结构。在分区密钥区域(Partition Key Area)中保存以下数据。其中，在分区密钥区域(Partition Key Area)的各保存密钥中，一并保存版本信息。在更新密钥时，也一并更新版本。

* IRL_PAR: 注册有分区存取作废设备(Device)、作废机器(作为设备存取机器的读写器、PC 等权证用户、权证发行部件)的标识符(ID)的作废表(Revocation List (Device ID))

* CRL_PAR: 注册有分区存取作废设备(Device)、作废机器(作为设备存取机器的读写器、PC 等权证用户、权证发行部件)的公开密钥证书标识符(例如，序列号: SN)的作废表(Revocation List (Certificate))

* Kdut_PAR1: 数据更新权证(DUT)的 MAC 验证密钥

* Kdut_PAR2: 数据更新加密密钥

* Kdut_PAR3: 数据更新权证(DUT)的 MAC 验证密钥

* Kdut_PAR4: 数据更新加密密钥

* Kfrt: 文件注册权证(FRT)的 MAC 验证密钥

* CERT_PAR: 认证机构 CA(PAR)发行的分区(Partition)的公开密钥证书

* PRI_PAR: 分区(Partition)的私有密钥

* PARAM_PAR: 分区(Partition)的公开密钥参数

* PUB_CA(PAR): 认证机构 CA(PAR)的公开密钥

* MKauth_PAR_A: 双向个别密钥鉴别主密钥

* Kauth_PAR_B: 双向个别密钥鉴别对称密钥

图 24 示出文件定义块(FDB: File Definition Block)的数据结构。在文件定义块(FDB: File Definition Block)中保存以下数据。

* File ID: 文件(File)标识名

* File Start Position: 文件(File)起始地址

* File Size: 文件(File)长度

* SPTIC (SPT Issuer Category): 服务许可权证(SPT)发行者范

畴

* SPTIC Version: 服务许可权证(SPT)发行者范畴(SPTIC)的版本

* File Structuer Type Code: 文件结构类型(File Structuer Type)的代码

* Acceptable Authentication Type: 表示容许鉴别类型。对各文件结构类型(File Structuer Type)定义的存取模式和该字段的各比特(在本例中最大为16个)对应。下面将详细说明。

* Acceptable Verification Type: 表示容许验证类型。对各文件结构类型(File Structuer Type)定义的存取模式和该字段的各比特(在本例中最大为16个)对应。下面将详细说明。

* Kspt: 服务许可权证(SPT)的MAC验证密钥(Kspt)

上述容许鉴别类型(Acceptable Authentication Type)是使对各文件结构类型(File Structuer Type)定义的存取模式和该字段的各比特(在本例中最大为16个)对应而设定的容许鉴别类型，例如在执行某个存取模式时，在与该模式对应的比特为 1 的情况下，如果公开密钥鉴别未完毕则不执行该存取模式。由此，在执行重要性更高的命令(例如存款处理等)时，能够强制执行公开密钥鉴别，确保安全性。通过使用权证也可以进行同样的控制，但是容许鉴别类型(Acceptable Authentication Type)与权证不同，作为文件定义块(FDB: File Definition Block)的一部分保存在设备中，所以该信息在创建文件后不会变更。因此，通过在想施加绝对不许变更容许鉴别类型的强烈制约时利用，能够提供安全性的最低限度的保证。

此外，上述容许验证类型(Acceptable Verification Type)是使对各文件结构类型(File Structuer Type)定义的存取模式和该字段的各比特(在本例中最大为16个)对应而设定的容许验证类型，例如在执行某个存取模式时，在与该模式对应的比特为 1 的情况下，如果基于公开密钥体制的权证鉴别未完毕则不执行该存取模式。在本例中，各字段为2个字节，所以最大只能与16个存取模式相对应，但是通过按照需要来增大字段长度，能够与更多的命令相对应。

此外，在本实施例结构中，将容许鉴别类型(Acceptable Authentication Type)、容许验证类型(Acceptable Verification

Type)设定为在设定为“1”时需要进行公开密钥体制的鉴别或验证，但是也可以使这些字段以 2 个比特为单位来构成，进行下述等细化的设定：在值为“11”的情况下容许公开密钥体制，在值为“01”的情况下容许对称密钥体制，在值为“00”“10”的情况下容许公开密钥体制、对称密钥体制中的任一个。

上述数据中的文件结构类型(File Structure Type)是表示分区内创建的文件的结构的代码。文件结构和代码的对应关系的一例示于图 25。

在文件结构中，有图 25 所示的各种结构(File Structure)，分别被分配以代码 0001～0007。各结构的意义如下所示。

* Random：具有本文件结构的数据是可随机进行所有读写的文件。

* Purse：具有本文件结构的数据是金额信息数据，是可进行减少(Sub)、增加(Add)等金额的价值变更处理的数据文件。

* Cyclic：具有本文件结构的数据是可进行循环型(Cyclic)数据写入的文件结构。

* Log：具有本文件结构的数据是日志数据文件，是各数据处理信息的记录信息文件。

* Key：表示具有本文件结构的数据是密钥信息数据。

* 复合文件：是具有上述各种文件结构的复合结构(例如，Purse 和 Log)的文件。向复合文件分配因其组合方式而异的代码(在图中，0006：复合文件 1，0007：复合文件 2)。

以上，说明了设备的存储部中保存的数据。后面将说明使用这些数据的具体处理。

[A8. 各权证的数据格式]

如前所述，在对设备执行的分区设定注册处理中，需要合法的权证发行部件(Ticket Issuer)发行的分区注册权证(PRT：Partition Registration Ticket)；在设备中设定的分区内执行的文件设定注册处理中，需要合法的权证发行部件(Ticket Issuer)发行的文件注册权证(FRT：File Registration Ticket)；而在对各文件执行的存取中，需要合法的权证发行部件(Ticket Issuer)发行的服务许可权证(SPT：Service Permission Ticket)。此外，如前述设备的存储部的数据说明一栏中简单说明过的那样，在设备保存数据更新处理中，需

要数据更新权证(DUT)。

这些权证由将存取设备的规则变为二进制数据而记述的数据串构成。按照对设备执行的处理，将权证从权证用户---例如作为设备存取机器的读写器发送到设备。接收到权证的设备执行权证完整性验证处理，在完整性验证成功的情况下，根据权证上记录的规则来执行各种处理(例如，分区创建、文件创建、数据存取)。以下，说明这些权证的数据格式。

[A8. 1. 分区注册权证(PRT)]

分区注册权证(PRT: Partition Registration Ticket)是对设备进行分区设定注册处理时应用的存取控制权证。使用合法的设备管理器管辖下的权证发行部件(Ticket Issuer)发行的PRT，根据PRT上记录的过程，通过分区管理器管辖下的权证用户(例如，作为设备存取机器的读写器)来存取设备，从而能够在PRT上记录的限制内设定分区。

图26示出分区注册权证(PRT: Partition Registration Ticket)的数据格式。在分区注册权证(PRT: Partition Registration Ticket)中保存以下说明的数据。

* Ticket Type: 权证(Ticket)的类型
* Format Version: 权证(Ticket)的格式版本
* Ticket Issuer: 设备管理器的标识符(=DMC)
* Serial Number: 权证(Ticket)的序列号
* Size of Ticket: 权证(Ticket)的长度
* Authentication Flag: 表示在权证(Ticket)利用处理中是否需要与设备(Device)进行相互鉴别的标志
* Ticket User 所属组(Group): 权证(Ticket)用户所属组
* Authentication Type: 设备(Device)的相互鉴别类型(公开密钥鉴别、对称密钥鉴别、或任一种皆可(Any))
* Ticket User 的标识符: 判别权证(Ticket)用户的标识数据(范畴或标识符)

本字段为与[Authentication Type]关联的数据，在[Authentication Type]为公开密钥鉴别的情况下，保存识别名(DN: Distinguished Name)、范畴(Category)或序列号(SN)；而在[Authentication Type]为对称密钥鉴别的情况下，保存鉴别ID。在

无需鉴别的情况下，不必保存。

* PMC：作为分区管理器代码(Partition Manager Code)在分区定义块(Partition Definition Block)中记述的代码

* PMC Version：分区管理器代码(PMC)的版本

* Operation Type：指定分区(Partition)创建或删除(创建(Generate)/删除(Delete))

* Partition Size：分区(Partition)的长度

* Integrity Check Type：权证(Ticket)的完整性验证值的类型(公开密钥体制(Public)/对称密钥体制(Common))

* Integrity Check Value：权证(Ticket)的完整性验证值(公开密钥体制：签名(Signature)，对称密钥体制：MAC)

其中，在将分区注册权证(PRT)发行部件(PRT Issuer)发行的权证(Ticket)向权证用户发送时，在公开密钥体制的情况下，分区注册权证(PRT)发行部件(PRT Issuer)的公开密钥证书(CERT_PRTI)也一起发送。PRT发行部件的公开密钥证书(CERT_PRTI)的属性(Attribute)与PRT发行部件(PRT Issuer)的标识符(PRTIC)一致。

在记录有设备(Device)的相互鉴别类型(公开密钥鉴别、对称密钥鉴别、或任一种皆可(Any))的[Authentication Type]中，记录作为使用权证的相互鉴别应执行的鉴别类型。具体地说，记录下述信息：指定执行设备鉴别、分区鉴别中的某一种、还是两种鉴别，执行公开密钥体制、对称密钥体制中的某一种、还是任一种鉴别皆可，这将在后面详细说明。

在记录权证(Ticket)的完整性验证值(公开密钥体制：签名(Signature)，对称密钥体制：MAC)的[Integrity Check Value]字段中，如果是公开密钥体制，则根据分区注册权证发行部件(PRT Issuer)的私有密钥来生成签名(参照图 12)并保存。在设备管理器本身兼作分区注册权证发行部件(PRT Issuer)的情况下，用设备管理器的私有密钥来生成签名。在签名验证处理(参照图 13)时，使用对应的CA(DEV)的公开密钥。因此，执行权证验证的设备需要在接收权证时、或提前取得分区注册权证发行部件(PRT Issuer)(例如，设备管理器)的公开密钥(公开密钥证书)。

在验证分区注册权证(PRT)发行部件(PRT Issuer)的公开密钥证

书(CERT_PRTI)后，可通过从公开密钥证书(CERT_PRTI)中取出的分区注册权证(PRT)发行部件(PRT Issuer)的公开密钥来进行 ICV(Integrity Check Value)的签名验证。后面将用流程来说明这些处理。

[A8. 2. 文件注册权证(FRT)]

文件注册权证(FRT: File Registration Ticket)是在对设备设定的分区中设定注册文件时应用的存取控制权证。使用合法的分区管理器管辖下的权证发行部件(Ticket Issuer)发行的 FRT，根据 FRT 上记录的过程通过权证用户(例如，作为设备存取机器的读写器)来存取设备，从而能够在 FRT 上记录的限制内设定文件。

图 27 示出文件注册权证(FRT: File Registration Ticket)的数据格式。在文件注册权证(FRT: File Registration Ticket)中保存以下说明的数据。

* Ticket Type: 权证(Ticket)的类型

* Format Version: 权证(Ticket)的格式版本

* Ticket Issuer: 分区管理器的标识符(=PMC)

* Serial Number: 权证(Ticket)的序列号

* Size of Ticket: 权证(Ticket)的长度

* Authentication Flag: 表示在权证(Ticket)利用处理中是否需要与设备(Device)进行相互鉴别的标志

* Ticket User 所属(Group): 权证(Ticket)用户所属组

* Authentication Type: 设备(Device)的相互鉴别类型(公开密钥鉴别、对称密钥鉴别、或任一种皆可(Any))

* Ticket User 的标识符: 判别权证(Ticket)用户的标识数据(范畴或标识符)

本字段为与[Authentication Type]关联的数据，在[Authentication Type]为公开密钥鉴别的情况下，保存识别名(DN: Distinguished Name)、范畴(Category)或序列号(SN)；而在[Authentication Type]为对称密钥鉴别的情况下，保存鉴别 ID。在无需鉴别的情况下，不必保存。

* SPTIC: 服务许可权证发行部件的代码

* SPTIC Ver: 服务许可权证发行部件的代码(SPTIC)的版本

* File ID: 分区内创建的文件(File)的标识符(ID)

* Operation Type: 指定文件的创建或删除(创建(Generate)/删除(Delete))

* File Size: 创建的文件(File)的长度

* File Structure: 创建的文件(File)的文件结构(Structure)

* Acceptable Authentication Type: 表示对用该权证定义的文件执行存取模式所需的相互鉴别的种类(公开密钥体制、公开密钥、对称密钥中任一种皆可)的比特串

* Acceptable Verification Type: 表示对用该权证定义的文件执行存取模式所需的服务许可权证(SPT)验证的种类(公开密钥体制、公开密钥、对称密钥中任一种皆可)的比特串

* Kspt_Encrypted: 用该分区的文件注册权证的 MAC 验证密钥 Kfrt 对文件定义块(File Definition Block)中记载的服务许可权证(SPT)的 MAC 验证密钥 Kspt 进行加密所得的数据 Kfrt (Kspt)。

* Integrity Check Type: 权证(Ticket)的完整性验证值的类型(公开密钥体制(Public)/对称密钥体制(Common))

* Integrity Check Value: 权证(Ticket)的完整性验证值(公开密钥体制: 签名(Signature), 对称密钥体制: MAC)

其中，在将文件注册权证(FRT)发行部件(FRT Issuer)发行的权证(Ticket)向权证用户发送时，在公开密钥体制的情况下，文件注册权证(FRT)发行部件(FRT Issuer)的公开密钥证书(CERT_FRTI)也一起发送。FRT 发行部件的公开密钥证书(CERT_FRTI)的属性(Attribute)与文件注册权证(FRT)发行部件(FRT Issuer)的标识符(FRTIC)一致。

在记录有设备(Device)的相互鉴别类型(公开密钥鉴别、对称密钥鉴别、或任一种皆可(Any))的[Authentication Type]中，记录作为使用权证的相互鉴别应执行的鉴别类型。具体地说，记录下述信息：指定执行设备鉴别、分区鉴别中的某一种、还是两种鉴别，执行公开密钥体制、对称密钥体制中的某一种、还是任一种鉴别皆可，这将在后面详细说明。

在记录权证(Ticket)的完整性验证值(公开密钥体制: 签名(Signature), 对称密钥体制: MAC)的[Integrity Check Value]字段中，如果是公开密钥体制，则根据文件注册权证发行部件(FRT

Issuer)的私有密钥来生成签名(参照图 12)并保存。在分区管理器本身兼作文件注册权证发行部件(FRT Issuer)的情况下，用分区管理器的私有密钥来生成签名。在签名验证处理(参照图 13)时，使用文件注册权证发行部件的公开密钥。因此，执行权证验证的设备需要在接收权证时、或提前取得文件注册权证发行部件(FRT Issuer)(例如，分区管理器)的公开密钥(公开密钥证书)。

在验证文件注册权证(FRT)发行部件(FRT Issuer)的公开密钥证书(CERT_FRTI)后，可通过从公开密钥证书(CERT_FRTI)中取出的文件注册权证(FRT)发行部件(FRT Issuer)的公开密钥来进行 ICV (Integrity Check Value)的签名验证。后面将用流程来说明这些处理。

[A8. 3. 服务许可权证(SPT)]

服务许可权证(SPT: Service Permission Ticket)是存取对设备设定的分区内的各数据来执行数据读出、数据写入、金额数据的减少、增加等处理时应用的存取控制权证。使用合法的分区管理器管辖下的权证发行部件(Ticket Issuer)发行的 SPT，根据 SPT 上记录的过程通过权证用户(例如，作为设备存取机器的读写器)来存取设备，从而能够在 SPT 上记录的限制内设定文件。

其中，服务许可权证(SPT: Service Permission Ticket)有只许可从分区中设定的文件中存取单个文件的形式、和许可存取多个文件的形式，下面说明各个形式。

图 28 示出只许可从分区中设定的文件中存取单个文件的形式的服务许可权证(SPT: Service Permission Ticket)的数据格式。在服务许可权证(SPT: Service Permission Ticket)中保存以下说明的数据。

* Ticket Type: 权证(Ticket)的类型
* Format Version: 权证(Ticket)的格式版本
* Ticket Issuer: 分区管理器的标识符(=PMC)
* Serial Number: 权证(Ticket)的序列号
* Size of Ticket: 权证(Ticket)的长度
* Authentication Flag: 表示在权证(Ticket)利用处理中是否需要与设备(Device)进行相互鉴别的标志

* Ticket User 所属组(Group): 权证(Ticket)用户所属组

* Authentication Type: 设备(Device)的相互鉴别类型(公开密钥鉴别、对称密钥鉴别、或任一种皆可(Any))

* Ticket User 的标识符: 判别权证(Ticket)用户的标识数据(范畴或标识符)

本字段为与[Authentication Type]关联的数据，在[Authentication Type]为公开密钥鉴别的情况下，保存识别名(DN: Distinguished Name)、范畴(Category)或序列号(SN)；而在[Authentication Type]为对称密钥鉴别的情况下，保存鉴别 ID。在无需鉴别的情况下，不必保存。

* File ID: 分区内创建的文件(File)的标识符(ID)

* File Access Mode: 对允许存取的文件(File)的存取模式(Access Mode)

* Integrity Check Type: 权证(Ticket)的完整性验证值的类型(公开密钥体制(Public)/对称密钥体制(Common))

* Integrity Check Value: 权证(Ticket)的完整性验证值(公开密钥体制: 签名(Signature)，对称密钥体制: MAC)

其中，在将服务许可权证(SPT)发行部件(SPT Issuer)发行的权证(Ticket)向权证用户发送时，在公开密钥体制的情况下，服务许可权证(SPT)发行部件(SPT Issuer)的公开密钥证书(CERT_SPTI)也一起发送。SPT 发行部件的公开密钥证书(CERT_SPTI)的属性(Attribute)与(SPT)发行部件(SPT Issuer)的标识符(SPTIC)一致。

在分区管理器兼作服务许可权证(SPT)发行部件(SPT Issuer)的结构中，服务许可权证(SPT)发行部件(SPT Issuer)的代码可设定为分区管理器代码(PMC)。

在记录有设备(Device)的相互鉴别类型(公开密钥鉴别、对称密钥鉴别、或任一种皆可(Any))的[Authentication Type]中，记录作为使用权证的相互鉴别应执行的鉴别类型。具体地说，记录下述信息: 指定执行设备鉴别、分区鉴别中的某一种、还是两种鉴别，执行公开密钥体制、对称密钥体制中的某一种、还是任一种鉴别皆可，这将在后面详细说明。

在记录权证(Ticket)的完整性验证值(公开密钥体制: 签名

(Signature)，对称密钥体制：MAC)的[Integrity Check Value]字段中，如果是公开密钥体制，则根据服务许可权证发行部件(SPT Issuer)的私有密钥来生成签名(参照图 12)并保存。在分区管理器本身兼作服务许可权证发行部件(SPT Issuer)的情况下，用分区管理器的私有密钥来生成签名。在签名验证处理(参照图 13)时，使用服务许可权证(SPT)发行部件(SPT Issuer)的公开密钥。因此，执行权证验证的设备需要在接收权证时、或提前取得服务许可权证发行部件(SPT Issuer)(例如，分区管理器)的公开密钥(公开密钥证书)。

在验证服务许可权证(SPT)发行部件(SPT Issuer)的公开密钥证书(CERT_SPTI)后，可通过从公开密钥证书(CERT_SPTI)中取出的服务许可权证(SPT)发行部件(SPT Issuer)的公开密钥来进行 ICV (Integrity Check Value)的签名验证。后面将用流程来说明这些处理。

用图 29 来说明上述权证格式的说明中的 File Access Mode“对允许存取的文件(File)的存取模式(Access Mode)”中记录的模式和存取形态。

作为文件而生成的数据各种各样，有用户的标识数据、金额数据、加密处理密钥数据、日志数据、或复合文件数据等，对存取数据执行与各数据对应的存取处理、即数据读取、写入、删除、增加、减少、加密、解密….

在服务许可权证(SPT)的 File Access Mode 中，定义了许可该各种各样的存取形态中的哪一种存取模式。存取模式的一览示于图 29。图 29 所示的存取模式只是一例，也可以设定其他与设备中保存的数据对应的存取模式。

可以对服务许可权证(SPT)中设定的[File ID：分区内的存取文件(File)的标识符(ID)]所示的文件执行文件存取模式[File Access Mode]定义的处理。如果服务许可权证(SPT)中设定的文件存取模式是读出(Read)，则可以读出文件内的数据。如果服务许可权证(SPT)中设定的文件存取模式是写入(Write)，则可以向文件内写入数据。

这种存取模式由前述文件结构(File Structure)(参照图 25)限制其可设定的模式。例如如果文件结构是 purse 则是金额数据，可以设定增加(Add)、减少(Sub)等存取模式。这种文件结构、可设定的存取

模式、以及从作为设备存取机器的读写器向设备发送的命令的例子示于图 30。

图 30 示出文件结构为 Random 的情况下、和为复合文件的情况下可设定的存取模式、及命令的例子。

例如在文件结构为 Random 的情况下，在存取模式为读出(Read)的情况下，设备可容许的命令只有[Read]。而同样在文件结构为 Random 的情况下，在存取模式为加密读出(Read)的情况下，设备可容许的命令只有加密读出[EncRead]。

此外，在文件结构为包含 Purse 及 Log 的复合文件的情况下，在存取模式为存款类的情况下，设备可容许的命令只有存入[Deposit]。而同样在文件结构为包含 Purse 及 Log 的复合文件的情况下，在存取模式为取款类的情况下，设备可容许的命令的为取出[Withdrawal]、收据生成[Make Receipt]、收据读出[Read Receipt]。

作为与文件存取模式(参照图 29)的存款类对应的容许命令(参照图 30)，定义上述存入命令(Deposit Command)，在存取许可权证的文件存取模式(File Access Mode)中设定[存款类]，作为文件 ID (File ID)，生成指定了构成电子货币的复合文件的存取许可权证(SPT)，与存入命令(Deposit Command)一起发送存入金额数据，从而可例如将复合文件中的文件[Purse]加上 X 日元，执行向复合文件中的文件[Log]中写入处理记录等的顺序处理。这些处理将在后面详述。

除了图 30 所示的之外，可以组合其他各种存取模式、命令，按照实际的设备利用形态来设定。

设备将存储部中保存的各文件容许的命令的定义数据保存为图 30 所示的表，只在从上述存取机器输入的命令是上述定义数据定义的命令的情况下才执行命令。在复合文件容许的命令的定义数据中，包含可如上所述与复合文件中包含的多个文件分别对应来执行的多个命令构成的序列命令。

将待处理的特定的文件设定在服务许可权证(SPT)的文件 ID (File ID)一栏中，将规定的存取模式设定在服务许可权证(SPT)的文件存取模式(File Access Mode)中，从而可控制利用该服务许可权证(SPT)的文件存取。后面将用流程来说明具体处理。

接着，图 31 示出许可存取分区中设定的文件中的多个文件的形式

的服务许可权证(SPT: Service Permission Ticket)的数据格式。在服务许可权证(SPT: Service Permission Ticket)中保存以下说明的数据。

* Ticket Type: 权证(Ticket)的类型

* Format Version: 权证(Ticket)的格式版本

* Ticket Issuer: 分区管理器的标识符(=PMC)

* Serial Number: 权证(Ticket)的序列号

* Size of Ticket: 权证(Ticket)的长度

* Authentication Flag: 表示在权证(Ticket)利用处理中是否需要与设备(Device)进行相互鉴别的标志

* Ticket User 所属组(Group): 权证(Ticket)用户所属组

* Authentication Type: 设备(Device)的相互鉴别类型(公开密钥鉴别、对称密钥鉴别、或任一种皆可(Any))

* Ticket User 的标识符: 判别权证(Ticket)用户的标识数据(范畴或标识符)

本字段为与[Authentication Type]关联的数据，在[Authentication Type]为公开密钥鉴别的情况下，保存识别名(DN: Distinguished Name)或范畴(Category)；而在[Authentication Type]为对称密钥鉴别的情况下，保存鉴别 ID。在无需鉴别的情况下，不必保存。

* File ID: 分区内的存取文件(File)的标识符(ID)

* File Access Mode: 对允许存取的文件(File)的存取模式(Access Mode)

* Group of Target File: 允许存取的文件(File)的组(Group)

* Target File ID: 允许存取的文件(File)的标识符(ID)

* Read/Write Permission: 对允许存取的文件(File)(目标文件(Target File))的处理形态(读出(Read)、写入(Write))的许可

* Integrity Check Type: 权证(Ticket)的完整性验证值的类型(公开密钥体制(Public)/对称密钥体制(Common))

* Integrity Check Value: 权证(Ticket)的完整性验证值(公开密钥体制: 签名(Signature)，对称密钥体制: MAC)

这样，通过定义 Group of Target File “允许存取的文件(File)

的组(Group)”并记录在权证上，可用单个服务许可权证(SPT)来许可对分区内的多个文件的存取。

其中，在将上述服务许可权证(SPT)发行部件(SPT Issuer)发行的权证(Ticket)向权证用户发送时，在公开密钥体制的情况下，服务许可权证(SPT)发行部件(SPT Issuer)的公开密钥证书(CERT_SPTI)也一起发送。SPT 发行部件的公开密钥证书(CERT_SPTI)的属性(Attribute)与“服务许可权证”(SPT)发行部件(SPT Issuer)的标识符(SPTIC)一致。

在记录有设备(Device)的相互鉴别类型(公开密钥鉴别、对称密钥鉴别、或任一种皆可(Any))的[Authentication Type]中，记录作为使用权证的相互鉴别应执行的鉴别类型。具体地说，记录下述信息：指定执行设备鉴别、分区鉴别中的某一种、还是两种鉴别，执行公开密钥体制、对称密钥体制中的某一种、还是任一种鉴别皆可，这将在后面详细说明。

在验证服务许可权证(SPT)发行部件(SPT Issuer)的公开密钥证书(CERT_SPTI)后，可通过从公开密钥证书(CERT_SPTI)中取出的服务许可权证(SPT)发行部件(SPT Issuer)的公开密钥来进行 ICV(Integrity Check Value)的签名验证。后面将用流程来说明这些处理。

[A8. 4. 数据更新权证(DUT)]

数据更新权证(DUT: Data Update Ticket)是存取设备中保存的各种数据来执行数据更新处理时应用的存取控制权证。使用合法的数据更新权证(DUT)发行部件(Ticket Issuer)发行的 DUT，根据 DUT 上记录的过程通过权证用户(例如，作为设备存取机器的读写器)来存取设备，从而能够在 DUT 上记录的限制内执行数据处理。

其中，数据更新权证(DUT: Data Update Ticket)有为了执行设备管理器管理的数据项目的更新处理而应用的权证 DUT(DEV)、和为了执行分区管理器管理的分区内的数据项目的更新处理而应用的权证 DUT(PAR)。权证 DUT(DEV)发行部件处于设备管理器的管理下，而权证 DUT(PAR)发行部件处于分区管理器的管理下。

图 32 示出 2 种数据更新权证(DUT: Data Update Ticket) DUT(DEV)、DUT(PAR)的数据格式。在数据更新权证(DUT: Data Update

Ticket)中保存以下说明的数据。

* Ticket Type: 权证(Ticket)的类型(DUT(DEV)/DUT(PAR))

* Format Version: 权证(Ticket)的格式版本

* Ticket Issuer: 设备/分区管理器的标识符。如果权证(Ticket)的类型(Ticket Type)为 DUT(DEV)则为 DMC，而如果权证(Ticket)的类型(Ticket Type)为 DUT(PAR)则为 PMC

* Serial Number: 权证(Ticket)的序列号

* Size of Ticket: 权证(Ticket)的长度

* Ticket User 所属组(Group): 权证(Ticket)用户所属组

* Ticket User 的标识符: 判别权证(Ticket)用户的标识数据(范畴或标识符)

本字段为与[Authentication Type]关联的数据，在[Authentication Type]为公开密钥鉴别的情况下，保存识别名(DN: Distinguished Name)或范畴(Category)；而在[Authentication Type]为对称密钥鉴别的情况下，保存鉴别 ID。在无需鉴别的情况下，不必保存。

* Authentication Type: 设备(Device)的相互鉴别类型(公开密钥鉴别、对称密钥鉴别、或任一种皆可(Any))

* Encrypted Flag: 被更新的数据是否加密过(加密过: Encrypted/未加密: none)

* Old Data Code: 被更新的旧数据的代码(Code)

* Data Version Rule: 进行数据更新时的版本条件

* Data Version Condition: 进行数据更新时的版本值

* Size of New Data: 要更新的新数据的长度

* New Data: 要更新的新数据(有时被加密)

* New Data Version: 要更新的数据的版本

* Integrity Check Type: 权证(Ticket)的完整性验证值的类型(公开密钥体制(Public)/对称密钥体制(Common))

* Integrity Check Value: 权证(Ticket)的完整性验证值(公开密钥体制: 签名(Signature)，对称密钥体制: MAC)

在应用数据更新权证(DUT: Data Update Ticket)来进行数据更新时，通过[Data Version Rule: 进行数据更新时的版本条件]、和

[Data Version Condition: 进行数据更新时的版本值]这 2 个字段的组合来表现条件.

进行数据更新时的版本条件[Data Version Rule]有 3 种: Any、Exact、Older.

Any 可与版本(Version)条件无关地进行数据更新;

Exact 在与后续[Data Version Condition]指定的值相同的情况下可进行数据更新;

Older 只在 New Data Version 更加新的情况下才可进行数据更新. 其中, 在版本条件[Data Version Rule]为 Any 或 Older 的情况下, 不使用或忽略[Data Version Condition].

在记录有设备(Device)的相互鉴别类型(公开密钥鉴别、对称密钥鉴别、或任一种皆可(Any))的[Authentication Type]中, 记录作为使用权证的相互鉴别应执行的鉴别类型. 具体地说, 记录下述信息: 指定执行设备鉴别、分区鉴别中的某一种、还是两种鉴别, 执行公开密钥体制、对称密钥体制中的某一种、还是任一种鉴别皆可, 这将在后面详细说明.

在设备管理器兼作数据更新权证-DUT(DEV)发行部件(DUT Issuer)的结构中, 可将数据更新权证-DUT(DEV)发行部件(DUT Issuer)的代码(权证用户(Ticket User))设定为设备管理器代码(DMC). 而在分区管理器兼作数据更新权证-DUT(PAR)发行部件(DUT Issuer)的结构中, 可将数据更新权证-DUT(PAR)发行部件(DUT Issuer)的代码设定为设备管理器代码(PMC).

在记录有设备(Device)的相互鉴别类型(公开密钥鉴别、对称密钥鉴别、或任一种皆可(Any))的[Authentication Type]中, 记录作为使用权证的相互鉴别应执行的鉴别类型. 具体地说, 记录下述信息: 指定执行设备鉴别、分区鉴别中的某一种、还是两种鉴别, 执行公开密钥体制、对称密钥体制中的某一种、还是任一种鉴别皆可, 这将在后面详细说明.

在记录权证(Ticket)的完整性验证值(公开密钥体制: 签名(Signature), 对称密钥体制: MAC)的[Integrity Check Value]字段中, 如果是公开密钥体制, 则根据设备更新权证发行部件(DUT Issuer)的私有密钥来生成签名(参照图 12)并保存. 在设备管理器本

身兼作设备更新注册权证发行部件(DUT Issuer)的情况下，用设备管理器的私有密钥来生成签名。此外，在分区管理器自身兼作设备更新注册权证发行部件(DUT Issuer)的情况下，用分区管理器的私有密钥来生成签名。在此情况下，在签名验证处理(参照图 13)时，使用设备管理器或分区管理器的公开密钥。因此，执行权证验证的设备需要在接收权证时、或提前取得设备更新权证发行部件(DUT Issuer)(例如，设备管理器或分区管理器)的公开密钥(公开密钥证书)。

在验证设备更新权证(DUT)发行部件(DUT Issuer)的公开密钥证书(CERT_DUTI)后，可通过从公开密钥证书(CERT_DUTI)中取出的数据更新权证(DUT)发行部件(DUT Issuer)的公开密钥来进行 ICV(Integrity Check Value)的签名验证。

应用数据更新权证(DUT: Data Update Ticket)来更新的数据例示于图 33。

如图 33 所示，在待更新数据中，包含设备管理器代码、设备管理器代码版本、分区管理器代码、分区管理器代码版本、各权证发行部件代码、各权证的 MAC 生成密钥及版本、作废表等。这些待更新的各数据应用数据更新权证(DUT: Data Update Ticket)，根据 DUT 上记录的规则被更新。后面将用流程来说明更新处理的具体过程。其中，设备管理器代码版本、分区管理器代码版本等版本信息在附加有各版本的数据的更新处理时一并被更新。这些版本信息被保存到数据更新权证(DUT: Data Update Ticket)中。

[B. 对用户发放设备、对设备进行各种设定、设备利用处理详述]

接着，参照附图来详细说明利用上述具有进行过分区分割的存储区域的设备之前的处理、以及设备利用处理。说明的过程根据以下项目来进行。

B1. 从设备初始注册到利用的流程

B2. 设备生产实体执行的初始注册处理

B3. 设备管理器的管辖处理

B3. 1. 设备管理器执行的设备注册处理

B3. 2. 设备管理器管理下的公开密钥证书发行处理

B4. 分区管理器的管辖处理

B4. 1. 分区管理器管理下的利用分区注册权证(PRT)的分区设定

注册、删除处理

B4. 2. 分区管理器管理下的公开密钥证书发行处理

B4. 3. 分区创建处理各方式中的处理过程

B4. 4. 利用文件注册权证(FRT)的文件创建、删除处理

B4. 5. 文件创建处理各方式中的处理过程

B4. 6. 利用服务许可权证(SPT)的服务(文件存取)处理

B4. 7. 利用服务许可权证(SPT)的存取处理各方式中的处理过程

B5. 利用数据更新权证(DUT)的设备数据更新处理

[B1. 从设备初始注册到利用的流程]

具有EEPROM(闪速存储器)的设备由设备生产实体(manufacturer)生产，由设备管理器执行初始数据的写入，提供(例如，销售、租借)给用户来利用。为了使用户从各种服务主体利用设备来接受服务，需要由分区管理器在设备的存储部中设定分区，在设定的分区内设定保存服务提供用的数据的文件。

此外，在对设备执行的各种处理、即利用分区注册权证(PRT)的分区设定、利用文件注册权证(FRT)的文件设定、以及利用服务许可权证(SPT)的数据存取等各种处理时，在设备和对设备执行处理的权证用户(例如，作为设备存取机器的读写器)之间执行各种过程。例如确认双方是合法的机器、设备的相互鉴别处理；或用于保证并确认传送数据的完整性的签名生成、验证处理；以及数据加密、解密处理等。在本发明的结构中，提出了在进行这些处理时使用公开密钥证书的结构。因此，在通过设备来利用服务前对设备执行公开密钥证书发行处理、设备保存处理。

例如在设备和对设备执行处理的权证用户(例如，作为设备存取机器的读写器)之间用公开密钥证书来执行相互鉴别处理，以确认了双方的完整性为条件来执行利用分区注册权证(PRT)的分区设定、利用文件注册权证(FRT)的文件设定、以及利用服务许可权证(SPT)的数据存取等各种处理。此外，在必要时在相互传送的数据上附加电子签名，执行验证。此外，在必要时也执行传送数据的加密、解密处理。

图34是从设备的生产直至利用之前的流程的示意图。后面将参照流程来详细说明这些处理，但是为了理解整体性的处理，简单说明图34所示的各阶段。

1. 首先，设备由设备生产实体(manufacturer)生产。在生产设备时，作为各设备的标识数据(ID)的标识代码被附加在各设备上。在生产阶段中，设备代码、生产代码等各种生产信息(Manufacture Information Block(参照图 14))被写入并保存到设备的存储器中。

2. 接着，在设备管理器向用户提供设备前，将自己的 ID、认证机构的公开密钥(PUB CA(DEV))等设备管理信息(Device Management Information(参照图 15))、设备密钥(Device Key(参照图 18))等信息保存到存储器中。

3. 由设备管理器写入了管理信息的设备被提供给用户。

4. 接着，用户执行设备公开密钥证书取得处理，将取得的设备公开密钥证书(CERT DEV)保存到设备的设备密钥区域(参照图 18)中。

5. 在设备的存储部中设定分区，想要提供服务的服务主体(分区管理器)向设备管理器请求设定分区，接受允许并且接收分区注册权证(PRT)。此外，指定与设备的通信处理中使用的认证机构的公开密钥(PUB CA(PAR))。

6. 设备与分区管理器管理的权证用户(例如，作为设备存取机器的读写器)执行通信，应用分区注册权证(PRT)来进行分区注册处理，并且将认证机构的公开密钥(PUB CA(PAR))保存到分区密钥区域(参照图 23)中。

7. 设定了分区的设备将分区公开密钥证书发行请求发送到分区管理器，将取得的分区公开密钥证书(CERT PAR)保存到分区密钥区域(参照图 23)中。

对想要设定分区并提供服务的各个分区管理器执行上述 5～7 的分区设定等处理，将多个分区注册到设备中。

8. 接着，分区管理器在设备中设定的分区内，例如应用文件注册权证(FRT)来执行与服务对应的文件设定注册处理。

9. 10. 通过在设定的分区内注册文件，例如可执行电子货币、月票等由文件内数据定义的各种服务。在文件内的数据读取、数据写入等处理中，应用服务许可权证(SPT)。即只在应用合法的权证发行部件发行的服务许可权证(SPT)的情况下，才根据 SPT 上记录的规则来执行数据的读取、写入等。

此外，虽然未图示，但是在必要时使用数据更新权证(DUT)来执行

设备的保存数据中的待更新处理数据(例如，设备管理器代码、设备管理器代码版本、分区管理器代码、分区管理器代码版本、各权证发行部件代码、各权证的MAC生成密钥及版本、作废表等)的更新处理。其中，设备管理器代码版本、分区管理器代码版本等版本信息在附加有各版本的数据的更新处理时一并被更新。这些版本信息被保存到数据更新权证(DUT: Data Update Ticket)中。

以下，参照流程和其他图来详细说明各处理。

[B2. 设备生产实体执行的初始注册处理]

首先，用图 35 来说明设备生产实体执行的初始注册处理。图 35 左侧示出设备生产实体(Manufacture)的注册装置的处理，右侧示出设备(参照图 5)的处理。其中，设备生产实体(Manufacture)的注册装置由读写器(参照图 10)构成，该读写器是可对设备进行数据读取写入处理的专用的设备存取机器。

首先，在步骤 S101 中，注册装置向设备发送生产信息块(MIB: Manufacture Information Block(参照图 14))写入标志(Writable Flag)读出命令。设备接收到命令(S121)后，将设备的存储部的生产信息块(MIB)内的写入(Writable)标志发送到注册装置(S122)。

接收到生产信息块(MIB)内的写入(Writable)标志(S102)的注册装置判别写入标志(Writable Flag)是否已被设定为可写入(0xffff)(S103)。在写入标志(Writable Flag)未被设定为可写入(0xffff)的情况下，不能执行以下的生产信息块(MIB: Manufacture Information Block)写入处理，作为出错而结束。

在写入标志(Writable Flag)已被设定为可写入(0xffff)的情况下，生成设备的生产信息块(MIB: Manufacture Information Block(参照图 14))(S104)，与 MIB 写入命令一起，将 MIB 数据发送到设备(S105)。

接收到 MIB 写入命令及 MIB 数据的设备验证 MIB 写入标志(Writable Flag)(S124)，在写入标志(Writable Flag)未被设定为可写入(0xffff)的情况下，不能执行以下的生产信息块(MIB: Manufacture Information Block)写入处理，作为出错而结束。在写入标志(Writable Flag)已被设定为可写入(0xffff)的情况下，将接收到的MIB数据写入到MIB区域中(S125)。

MIB 数据写入处理结束后，将写入结束通知发送到注册装置(S126)。接收到写入结束通知(S106)的注册装置将初始注册完成命令发送到设备(S107)，接收到初始注册完成命令(S127)的设备将生产信息块(MIB: Manufacture Information Block)的写入标志(Writable Flag)设定为不可写入(0x0000)(S128)，将写入结束通知发送到注册装置(S129)。

接收到写入结束通知(S108)的注册装置向设备发送生产信息块(MIB: Manufacture Information Block(参照图 14))写入标志(Writable Flag)读出命令(S109)。设备接收到命令(S130)后，将设备的存储部的生产信息块(MIB)内的写入标志(Writable Flag)发送到注册装置(S131)。

接收到生产信息块(MIB)内的写入标志(Writable Flag)(S110)的注册装置判别写入标志(Writable Flag)是否已被设定为不可写入(0x0000)(S111)。在写入标志(Writable Flag)未被设定为不可写入(0x0000)的情况下，表示正常的 MIB 数据写入处理未结束，作为出错而结束处理。在写入标志(Writable Flag)已被设定为不可写入(0x0000)的情况下，认为正常的 MIB 数据写入处理已结束，结束处理。

[B3. 设备管理器的管辖处理]

接着，说明设备管理器的管辖处理。这里，说明开始使用设备以前执行的处理。开始使用设备以前执行的设备管理器的处理有向设备的存储部的设备管理信息块(DMIB: Device Management Information Block)、公开密钥类设备密钥定义块(DKDB: Device Key Definition Block(PUB))、对称密钥类设备密钥定义块(DKDB: Device Key Definition Block(Common))、设备密钥区域(Device Key Area)中写入数据的设备注册处理、以及对设备发行设备公开密钥证书(CERT DEV)的处理。以下，详细说明这些处理。

[B3. 1. 设备管理器执行的设备注册处理]

用图 36 以下的流程，来说明设备管理器对设备执行的附带设备管理信息等的保存处理的初始注册处理。在图 36 以下的流程中，左侧示出设备管理器(DM)的初始注册装置的处理，右侧示出设备(参照图 5)的处理。其中，设备管理器(DM)的初始注册装置是可对设备进行数据读取写入处理的装置(例如，作为设备存取机器的读写器、PC)，具有

与图 10 的作为设备存取机器的读写器相当的结构.

首先，在步骤 S201 中，将设备标识符 IDm 读出(Read)命令输出到设备。设备接收命令(S211)，将设备的标识符 IDm 发送到注册装置(S212)。

接收到设备的标识符 IDm (S202)的注册装置在步骤 S203 中，向设备发送设备管理信息块(DMIB: Device Management Information Block(参照图 15))写入标志(Writable Flag)读出命令。设备接收到命令(S213)后，将设备的存储部的生产信息块(MIB)内的写入(Writable)标志发送到注册装置(S214)。

接收到设备管理信息块(DMIB)内的写入(Writable)标志(S204)的注册装置判别写入标志(Writable Flag)是否已被设定为可写入(0xffff)(S205)。在写入标志(Writable Flag)未被设定为可写入(0xffff)的情况下，不能执行以下的设备管理信息块(DMIB: Device Management Information Block)写入处理，作为出错而结束.

在写入标志(Writable Flag)已被设定为可写入(0xffff)的情况下，将设备管理器代码(DMC)及 DMC 版本写入(DMC Write)命令发送到设备(S206)。该命令是代码管理机关(参照图 1～图 3)向设备管理器预先分配的数据。

接收到 DMC Write 命令(S215)的设备验证 DMIB 写入标志(Writable Flag)(S216)，在写入标志(Writable Flag)未被设定为可写入(0xffff)的情况下，不能执行以下的设备管理信息块(DMIB: Device Management Information Block)写入处理，作为出错而结束。在写入标志(Writable Flag)已被设定为可写入(0xffff)的情况下，将接收到的设备管理器代码(DMC)及 DMC 版本写入到 DMIB 区域中(S217)。

设备管理器代码(DMC)及 DMC 写入处理结束后，将写入结束通知发送到注册装置(S218)。接收到写入结束通知(S207)的注册装置接着将设备总块数(Device Total Block Number)写入命令发送到设备(S208)。

接收到设备总块数(Device Total Block Number)写入命令(S219)的设备验证 DMIB 写入标志(Writable Flag)(S220)，在写入标志(Writable Flag)未被设定为可写入(0xffff)的情况下，不能执行以

下的设备管理信息块(DMIB: Device Management Information Block)写入处理，作为出错而结束。在写入标志(Writable Flag)已被设定为可写入(0xffff)的情况下，将接收到的设备总块数(Device Total Block Number)写入到 DMIB 区域中(S221)。进而，设备向 DMIB 区域的设备空闲块数信息区域(Free Block Number in Device)中写入 TB-4(S222)。TB 表示设备总块数(Device Total Block Number)。其中，TB-4 的 4 块表示生产信息块(MIB: Manufacture Information Block)、设备管理信息块(DMIB: Device Management Information Block)、公开密钥类设备密钥定义块(DKDB: Device Key Definition Block(PUB))、对称密钥类设备密钥定义块(DKDB: Device Key Definition Block(Common))。

接着,设备向设备管理信息块(DMIB)的分区数(Partition Number)区域中写入 0(S223)。因为此时在设备中未设定分区。进而，向 DMIB 的空闲区域的指针(Pointer of Free Area)中写入 0(S224)，将写入处理完成发送到注册装置(S225)。

从设备接收到写入处理完成通知(S209)的注册装置接着判定在设备鉴别中是否使用对称密钥(S231)。对于鉴别处理，后面将详细说明，可执行公开密钥鉴别方式、对称密钥鉴别方式中的某一种，设备管理器可设定设备所需的鉴别方式。如果设备是执行对称密钥鉴别的设备，则设备管理器将对称密钥鉴别所需的信息(例如，鉴别密钥生成用的主密钥等)设置到设备中，而如果设备是不执行对称密钥鉴别的设备，则不将这些信息保存到设备中。设备管理器按照设备采用的鉴别方式将可执行对称密钥鉴别、公开密钥鉴别中的某一种、或两种方式的数据设定到设备中。

如图 37 所示，在设备鉴别中使用对称密钥的情况下，执行步骤 S232～S233、S241～S245；而在设备鉴别中不使用对称密钥的情况下，省略这些步骤。

在设备鉴别中使用对称密钥的情况下，在步骤 S232 中，作为对称密钥鉴别数据写入命令，注册装置将 MKauth_DEV_A“双向个别密钥鉴别主密钥”、Kauth_DEV_B“双向个别密钥鉴别对称密钥”、IRL_DEV“注册有作废设备(Device)的设备标识符(ID)的作废表(Revocation List(Device ID))”、及其版本信息发送到设备。

在步骤 S241 中，设备接收上述写入命令，在步骤 S242 中，确认 DMIB 的写入标志(Writable Flag)是可写入，将所接收数据写入到设备密钥区域(参照图 18)中(S243)。接着，执行通过数据写入而产生的指针、长度、设备内的空闲块数的调整(S244)，将写入结束通知发送到注册装置(S245)。

接收到写入结束通知(S233)的注册装置在步骤 S234 中判定在设备鉴别中是否使用公开密钥。如图 37 所示，在设备鉴别中使用公开密钥的情况下，执行步骤 S235～S239、S246～S254；而在设备鉴别中不使用公开密钥的情况下，省略这些步骤。

在设备鉴别中使用公开密钥的情况下，在步骤 S235 中，作为公开密钥鉴别数据写入命令，注册装置将 PUB_CA(DEV) “发行设备管理器公开密钥的认证机构 CA(DEV)的公开密钥”、PARAM_DEV“设备(Device)的公开密钥参数”、CRL_DEV “注册有作废设备(Device)的公开密钥证书标识符(例如，序列号：SN)的作废表(Revocation List(Certificate))、及其版本信息发送到设备。

在步骤 S246 中，设备接收上述写入命令，在步骤 S247 中，确认 DMIB 的写入标志(Writable Flag)是可写入，将所接收数据写入到设备密钥区域(参照图 18)中(S248)。接着，执行通过数据写入而产生的指针、长度、设备内的空闲块数的调整(S249)，将写入结束通知发送到注册装置(S250)。

接收到写入结束通知(S236)的注册装置将公开密钥和私有密钥的密钥对生成命令发送到设备(S237)。其中，在本实施例中，密钥对的生成由设备执行，但是例如也可以由注册装置执行并提供给设备。

接收到密钥对生成命令(S251)的设备在设备内的加密处理部(参照图 5)中生成公开密钥(PUB DEV)和私有密钥(PRI DEV)的密钥对，将生成的密钥写入到设备密钥区域(参照图 18)中(S252)。其中，将公开密钥(PUB DEV)暂时保存到设备密钥区域的 CERT · DEV 区域中，其后，在接收到保存有公开密钥(PUB DEV)的公开密钥证书时置换为公开密钥证书(CERT)。接着执行通过数据写入而产生的指针、长度、设备内的空闲块数的调整(S253)，将生成保存的公开密钥发送到注册装置(S254)。

注册装置从设备接收公开密钥(PUB DEV)，与先前从设备接收到的

设备的标识符 IDm 一起，保存到设备管理器内的数据库(DB(DEV))(参照图 7)中。

接着，设备管理器的注册装置判定在分区注册权证(PRT: Partition Registration Ticket)验证处理中是否使用对称密钥(S261)。对于权证鉴别，后面将详细说明，可应用基于 MAC 值验证等的对称密钥体制、和用前述图 12、图 13 说明过的进行基于私有密钥的签名生成、基于公开密钥的签名验证的公开密钥体制中的某一种，设备管理器可以设定设备采用的验证处理方式。设备管理器按照设备采用的 PRT 权证验证方式将可执行对称密钥、公开密钥中的某一种、或两种方式的数据设定到设备中。

如果设备是执行对称密钥鉴别的设备，则设备管理器将对称密钥体制的 PRT 验证所需的信息(例如，PRT 验证对称密钥)设置到设备中，而如果设备是不执行对称密钥鉴别的设备，则不将这些信息保存到设备中。

如图 38 所示，在 PRT 验证中使用对称密钥体制的情况下，执行步骤 S262～S263、S271～S275；而在 PRT 验证中不使用对称密钥的情况下，省略这些步骤。

在 PRT 验证中使用对称密钥的情况下，在步骤 S262 中，作为 PRT 验证对称密钥写入命令，注册装置将 Kprt“分区注册权证(PRT)的 MAC 验证密钥”、及版本信息发送到设备。

在步骤 S271 中，设备接收上述写入命令，在步骤 S272 中，确认 DMIB 的写入标志(Writable Flag)是可写入，将所接收数据写入到设备密钥区域(参照图 18)中(S273)。接着，执行通过数据写入而产生的指针、长度、设备内的空闲块数的调整(S274)，将写入结束通知发送到注册装置(S275)。

接收到写入结束通知(S263)的注册装置在步骤 S264 中判定在 PRT 验证中是否使用公开密钥。如图 38 所示，在 PRT 验证中使用公开密钥的情况下，执行步骤 S265～S266、S276～S282；而在 PRT 验证中不使用公开密钥的情况下，省略这些步骤。

在 PRT 验证中使用公开密钥的情况下，在步骤 S265 中，作为 PRT 验证数据写入命令，注册装置将 PRTIC (PRT Issuer Category)“分区注册权证(PRT)发行者范畴”、PUB_CA(DEV)“发行设备管理器公开

密钥的认证机构 CA(DEV)的公开密钥”、PARAM_DEV“设备(Device)的公开密钥参数”、CRL_DEV“注册有作废设备(Device)的公开密钥证书标识符(例如，序列号：SN)的作废表(Revocation List(Certificate))”、及其版本信息发送到设备。

在步骤 S276 中，设备接收上述写入命令，在步骤 S277 中，确认 DMIB 的写入标志(Writable Flag)是可写入，在步骤 S278 中，将所接收数据中的 PRTIC (PRT Issuer Category)“分区注册权证(PRT)发行者范畴”写入到公开密钥类设备密钥定义块(DKDB: Device Key Definition Block (PUB)(参照图 16))中，将版本信息写入到同一块的版本区域中。

接着，设备在步骤 S279 中判定 PUB_CA(DEV)“发行设备管理器公开密钥的认证机构 CA(DEV)的公开密钥数据”是否已写入，在未写入的情况下，在步骤 S280 中，将 PUB_CA(DEV)、PARAM_DEV、CRL_DEV 写入到设备密钥区域(参照图 18)中。接着，执行通过数据写入而产生的指针、长度、设备内的空闲块数的调整(S281)，将写入结束通知发送到注册装置(S282)。

接收到写入结束通知(S266)的注册装置接着在步骤 S291 中判定设备是否支持对称密钥数据的更新。设备中保存的某些数据可作为待更新数据用前述数据更新权证(DUT: Data Update Ticket)(参照图 32)来更新。待更新数据如先前用图 33 所述。在使用该数据更新权证(DUT: Data Update Ticket)的更新处理中，也可采用对称密钥体制、或公开密钥体制中的某一种体制，设备管理器按照设备将可执行某一种体制或两种体制的数据设定到设备中。

如果设备是执行基于对称密钥体制的数据更新的设备，则设备管理器将对称密钥体制的数据更新处理所需的信息(例如，数据更新权证(DUT)的 MAC 验证密钥等)设置到设备中，而如果设备是不执行对称密钥鉴别的设备，则不将这些信息保存到设备中。

如图 39 所示，在使用数据更新权证(DUT: Data Update Ticket)的数据更新处理中使用对称密钥体制的情况下，执行步骤 S292～S293、S301～S305；而在数据更新中不使用对称密钥体制的情况下，省略这些步骤。

在数据更新中使用对称密钥的情况下，在步骤 S292 中，作为数据

更新权证(DUT: Data Update Ticket)验证对称密钥写入命令，注册装置将Kdut_DEV1“数据更新权证(DUT)的MAC验证密钥”、Kdut_DEV2“数据更新加密密钥”、Kdut_DEV3“数据更新权证(DUT)的MAC验证密钥”、Kdut_DEV4“数据更新加密密钥”及它们的版本信息发送到设备。

在步骤S301中，设备接收上述写入命令，在步骤S302中，确认DMIB的写入标志(Writable Flag)是可写入，将所接收数据写入到设备密钥区域(参照图18)中(S303)。接着，执行通过数据写入而产生的指针、长度、设备内的空闲块数的调整(S304)，将写入结束通知发送到注册装置(S305)。

接收到写入结束通知(S293)的注册装置在步骤S294中判定设备是否支持使用基于公开密钥体制的数据更新权证(DUT: Data Update Ticket)的数据更新处理。如图39所示，在支持公开密钥体制的情况下，执行步骤S295～S296、S306～S310；而在不支持公开密钥体制的情况下，省略这些步骤。

在支持公开密钥体制的情况下，在步骤S295中，作为数据更新权证(DUT: Data Update Ticket)发行者代码写入命令，注册装置将DUTIC_DEV (DUT Issuer Category)“数据更新权证(DUT: Data Update Ticket)发行者范畴”、及版本信息发送到设备。

在步骤S306中，设备接收上述写入命令，在步骤S307中，确认DMIB的写入标志(Writable Flag)是可写入，在步骤S308中，将所接收数据写入到公开密钥类设备密钥定义块(DKDB(PUB): Device Key Definition Block (PUB))中(S308)。接着，执行通过数据写入而产生的指针、长度、设备内的空闲块数的调整(S309)，将写入结束通知发送到注册装置(S310)。

接收到写入结束通知(S296)的注册装置接着在步骤S321中，将设备管理器(DM)初始注册完成命令发送到设备。接收到命令(S331)的设备在步骤S332中，判定分别对相互鉴别、分区注册权证(PRT)的验证、以及数据更新权证(DUT)的验证是否已设定了至少可执行公开密钥体制、对称密钥体制中的某一种处理的数据。在这些数据不足的情况下，不能执行任一种处理，判定设备管理器执行的初始注册出错，结束处理。

在步骤 S332 中，在判定分别对相互鉴别、分区注册权证(PRT)的验证、以及数据更新权证(DUT)的验证已设定了至少可执行公开密钥体制、对称密钥体制中的某一种处理的数据的情况下，在步骤 S333 中，设备将设备管理信息(DMIB: Device Management Information Block)的写入 (Writable) 标志设定为不可写入(0x0000)，将写入结束通知发送到注册装置(S334)。

接收到写入结束通知(S332)的注册装置向设备发送设备管理信息(DMIB: Device Management Information Block)(参照图 15)写入标志(Writable Flag)读出命令(S323)。设备接收到命令(S335)后，将设备的存储部的设备管理信息块(DMIB)内的写入标志(Writable Flag)发送到注册装置(S336)。

接收到设备管理信息块(DMIB)内的写入标志(Writable Flag)(S324)的注册装置判别写入标志(Writable Flag)是否已被设定为不可写入(0x0000)。在写入标志(Writable Flag)未被设定为不可写入(0x0000)的情况下，表示正常的 DMIB 数据写入处理未结束，作为出错而结束处理。在写入标志(Writable Flag)已被设定为不可写入(0x0000)的情况下，认为正常的 DMIB 数据写入处理已结束，结束处理。

设备生产实体(Manufacture)的注册装置执行的初始注册(图 35 的处理流程)、及设备管理器执行的初始注册处理(图 36～图 40 的处理流程)完成的状态下设备的存储器内保存数据结构例示于图 41。图 41 示出用图 6、图 14 至图 18 说明过的生产信息块(Manufacture Information Block)、设备管理信息块(Device Management Information Block)、公开密钥类设备密钥定义块(Device Key Definition Block (PUB))、对称密钥类设备密钥定义块(Device Key Definition Block (Common))、设备密钥区域(Device Key Area)。此时，在存储器中未形成分区。

如用图 14 说明过的那样，在生产信息块(Manufacture Information Block)中，写入作为设备唯一信息的设备代码等。写入到该生产信息块(Manufacture Information Block)中的信息、写入的部分信息、或根据写入的信息取得的运算数据相当于设备的标识符(IDm)。

其中，在图示的设备密钥区域(Device Key Area)中保存有Kauth_DEV_B“双向个别密钥鉴别对称密钥”、MKauth_DEV_A“双向个别密钥鉴别主密钥”，但是在设备没有进行对称密钥鉴别处理的请求的情况下，也可以不保存这些密钥，此外，在设备不执行基于对称密钥的权证验证处理的情况下，也可以不保存Kprt“分区注册权证(PRT)的MAC验证密钥”。

此外，在发行设备时不存在作废了的设备的情况下，或者在使用其他来源来取得作废表的情况下，也可以不保存 IRL_DEV“注册有作废设备(Device)的设备标识符(ID)的作废表(Revocation List(Device ID))”、CRL_DEV“注册有作废设备(Device)的公开密钥证书标识符(例如，序列号：SN)的作废表(Revocation List(Certificate))”。

[B3. 2. 设备管理器管理下的公开密钥证书发行处理]

接着用图42以下来说明设备管理器执行的设备公开密钥证书发行处理。在设备中，可保存整个设备的鉴别、以设备为单位的处理中可应用的设备公开密钥证书(CERT DEV)、以及对设备内的特定的分区进行处理时的鉴别及其他验证处理等可应用的分区公开密钥证书(CERT PAR)。可对设备中设定的每个分区设定保存分区公开密钥证书(CERT PAR)。

设备公开密钥证书(CERT DEV)被保存到设备管理器管辖的存储区域---设备密钥区域(Device Key Area)(参照图18)中，而分区公开密钥证书(CERT PAR)被保存到各分区管理器管辖的存储区域---分区密钥区域(Partition Key Area)(参照图23)中。

设备公开密钥证书(CERT DEV)通过经设备管理器管辖的注册机构将认证机构(CA for DM)(参照图2、图3)发行的公开密钥证书授予设备的过程来发行，对设备管理器管辖的注册机构发行的公开密钥证书(CERT DEV)执行管理(数据库222(参照图7))。

而分区公开密钥证书(CERT PAR)通过经分区管理器管辖的注册机构将认证机构(CA for PM)(参照图2、图3)发行的公开密钥证书授予设备的过程来发行，对分区管理器管辖的注册机构发行的公开密钥证书(CERT PAR)执行管理(数据库332(参照图9))。

根据图42及图43，来说明设备管理器管辖的注册机构对设备执行

的设备公开密钥证书(CERT DEV)发行处理的过程。其中，只取出设备管理器的注册机构(RA)结构的发行装置(DM)、认证机构(CA)、用户设备的关系示于图 44。如图 44 所示，控制部件 221 具有加密处理部件。其中，加密处理通过在控制部(CPU(图 8 的 2111))的控制下执行与加密处理有关的程序来进行。

在图 42、图 43 中，左侧是设备管理器管辖的注册机构的 CERT(公开密钥证书)发行装置、具体地说是图 7 所示的设备管理器的结构图中的控制部件 221 的处理，右侧是设备的处理。

首先在步骤 S351 中，CERT 发行装置取得作为设备公开密钥证书(CERT DEV)的发行对象的设备的用户信息，进行证书发行的许可(判定)，确保与作为发行对象的设备的信道。作为设备公开密钥证书(CERT DEV)的发行对象的设备的用户信息例如可从设备的初始注册时生成的数据中取得。此外，也可以另外经别的路径来取得用户名或地址、电话号码、e-mail 地址等。其中，用户信息也可以在设定与设备的信道后，从设备取得。信道不管有线、无线，只要作为可发送接收数据的信道来确保即可。

接着，CERT 发行装置在步骤 S352 中，将包含随机数的鉴别数据生成命令发送到设备。接收到鉴别数据生成命令(S361)的设备对所接收随机数 R、和设备标识符(IDm)的结合数据应用设备私有密钥(PRI DEV)来执行数字签名(S)生成处理(参照图 12)(S362)。设备将设备的设备数据(IDm)和签名(S)发送到 CERT 发行装置。

从设备接收到设备数据(IDm)和签名(S)(S353)的 CERT 发行装置以接收到的设备标识数据(IDm)为搜索关键字，从数据库 DB(DEV)222 中取得已保存的设备公开密钥(PUB DEV)。进而，应用取得的设备公开密钥(PUB DEV)来执行签名(S)的验证处理(参照图 13)(S355)。在验证未成功的情况下，判定为来自设备的发送数据是非法数据，结束处理。

在验证成功的情况下，请求认证机构(CA for DM)610 进行设备公开密钥证书(CERT DEV)发行处理(S357)。设备管理器接收认证机构 610 发行的设备公开密钥证书(CERT DEV)(S358)，发送到设备(S359)。

从设备管理器(注册机构)接收到设备公开密钥证书(CERT DEV)的设备用预先已保存在设备密钥区域中的认证机构的公开密钥(PUB CA(DEV))来执行接收到的设备公开密钥证书(CERT DEV)的签名验证。

即在公开密钥证书中有用认证机构的私有密钥执行的签名(参照图11)，进行该签名验证(S366)。

在签名验证失败的情况下，判定为不是合法的公开密钥证书，将出错通知发送到 CERT 发行装置(S385)。

在签名验证成功的情况下，比较设备公开密钥证书(CERT DEV)中保存的设备公开密钥(PUB DEV)和本设备中保管的设备公开密钥(PUB DEV)(S382)，在不一致的情况下执行出错通知；而在一致的情况下，将接收到的设备公开密钥证书(CERT DEV)保存到设备密钥区域(参照图 18)中(S383)。其中，在发行设备公开密钥证书(CERT DEV)以前，在该区域中保存本设备生成的公开密钥(PUB DEV)，在发行了合法的设备公开密钥证书(CERT DEV)时，由设备公开密钥证书(CERT DEV)来盖写。

在设备公开密钥证书(CERT DEV)的保存结束后，将保存处理结束通知发送到 CERT 发行装置(S384)。CERT 发行装置接收保存处理结束通知(S371)，确认保存成功(S372)，结束处理。在未得到保存成功的确认的情况下，作为出错而结束处理。

图 45 示出设备公开密钥证书(CERT DEV)发行处理中设备管理器 200、设备 100、认证机构(CA)610 各实体间的数据发送接收处理的说明图。

按图 45 中的 No. 1～14 的顺序来执行处理。其中，处理 No. 1 的设备管理器 200 从设备 100 取得设备标识符(IDm)、设备公开密钥(PUB DEV)的处理、及处理 No. 2 的设备标识符注册处理是由设备管理器在初始注册中执行的处理。

设备公开密钥证书(CERT DEV)的发行过程从处理 No. 3 开始，3. 设备管理器从设备取得客户信息；4. 注册客户信息(在已注册的情况下无需)；5. 从设备取得设备标识符(IDm)；6. 根据取得的设备标识符(IDm)来执行数据库搜索，取得对应的公开密钥(PUB DEV)；7. 设备和设备管理器间的鉴别处理，该处理在图 42、图 43 的处理中被省略了，但是在图 42、图 43 中，在从设备取得设备标识符(IDm)时执行签名验证，通过确认通信对方的发送数据，而省略了鉴别。最好执行图 42、图 43 中的签名验证、图 45 的鉴别中的至少某一个，或者都执行。其中，在后面的 4. 分区管理器的管辖处理的项目中将详细说明鉴别处

理。

8. 从设备管理器向认证机构发出设备公开密钥发行请求；9. 生成设备公开密钥证书(CERT DEV)；10. 在认证机构中注册生成公开密钥证书的数据；11. 从认证机构(CA)610 向设备管理器 200 发放公开密钥证书；12. 设备管理器更新数据库(注册公开密钥证书发行信息)；13. 从设备管理器向设备发送设备公开密钥证书(CERT DEV)；14. 在设备中保存设备公开密钥证书(CERT DEV)，如前所述，保存是写入保存到设备密钥区域中。

通过以上处理，设备取得设备公开密钥证书(CERT DEV)，保存到存储部中。将该设备公开密钥证书(CERT DEV)保存到存储器的设备密钥保存区域中后存储器的各块的数据保存结构示于图 46。图 46 示出先前用图 6、图 14 至图 18 说明过的生产信息块(Manufacture Information Block)、设备管理信息块(Device Management Information Block)、公开密钥类设备密钥定义块(Device Key Definition Block (PUB))、对称密钥类设备密钥定义块(Device Key Definition Block (Common))、设备密钥区域(Device Key Area)。此时，在存储器中未形成分区。

在图 46 所示的设备密钥区域(Device Key Area)中保存设备公开密钥证书(CERT DEV)。在发行设备公开密钥证书(CERT DEV)前，在该区域中保存设备生成的公开密钥(PUB DEV)，接收到设备公开密钥证书(CERT DEV)后，用设备公开密钥证书(CERT DEV)来进行盖写处理。其中，在有指针、长度、管理数据的情况下，通过该盖写处理来执行必要的变更处理。

[B4. 分区管理器的管辖处理]

接着，说明分区管理器的管辖处理。这里，说明开始使用设备以前执行的处理。作为开始使用设备以前执行的分区管理器的处理，有在设备的存储部中设定分区的处理、和对设备发行分区公开密钥证书(CERT PAR)的处理。以下，详细说明这些处理。在设定分区的处理中，包含设备和分区管理器间的相互鉴别处理(设备鉴别或分区鉴别)、分区注册权证(PRT: Partition Registration Ticket)完整性验证处理。其中，分区删除处理也基本上可根据与分区创建同样的过程来执行，所以一并说明。

[B4. 1. 分区管理器管理下的利用分区注册权证(PRT)的分区设定注册、删除处理]

首先，说明利用分区注册权证(PRT)(参照图 26)的分区设定注册、删除处理。参照图 47 以下的流程等附图来进行说明。其中，如上所述，在分区设定处理中，包含设备和分区管理器间的相互鉴别处理(设备鉴别或分区鉴别)、分区注册权证(PRT:Partition Registration Ticket)完整性验证处理，也说明这些处理。

下面说明图 47 所示的分区设定注册、删除处理流程。在图 47 中，左侧示出分区管理器的分区创建/删除装置的处理，右侧示出设备(参照图 5)的处理。其中，分区管理器的分区创建/删除装置是可对设备进行数据读取写入处理的装置(例如，作为设备存取机器的读写器、PC)，相当于图 10 的作为设备存取机器的读写器。首先，用图 47 来概要说明分区创建、删除处理，其后，用图 48 以下的流程来依次详细说明该处理中包含的各处理。

首先，在图 47 的步骤 S401 和 S410 中，执行分区创建/删除装置和设备间的相互鉴别处理。在执行数据发送接收的 2 个部件间，相互确认对方是否是合法的数据通信者，其后进行必要的数据传送。确认对方是否是合法的数据通信者的处理是相互鉴别处理。在相互鉴别处理时执行会话密钥的生成，将生成的会话密钥作为共享密钥来执行加密处理，进行数据发送，这种结构是一种最好的数据传送方式。

在本发明的系统中的相互鉴别处理中，执行设备鉴别或分区鉴别中的某一种。此外，对它们分别应用对称密钥体制鉴别、或公开密钥体制鉴别处理中的某一种。这些待后述。

其中，作为相互鉴别处理应执行的处理由应用的分区注册权证(PRT)(参照图 26)的

* Authentication Flag: 表示在权证(Ticket)利用处理中是否需要与设备(Device)进行相互鉴别的标志

* Authentication Type: 设备(Device)的相互鉴别类型(公开密钥鉴别、对称密钥鉴别、或任一种皆可(Any))

来决定。

在鉴别处理失败的情况下(S402、S411 中为“否(No)”)，表示不能确认相互是合法的机器、设备，不执行以下的处理，作为出错而结

束处理。

如果鉴别处理成功，则分区创建/删除装置向设备发送分区注册权证(PRT: Partition Registration Ticket)。分区注册权证(PRT)是根据分区管理器的请求、由设备管理器管理的分区注册权证(PRT)发行部件(PRT Issuer)向分区管理器发行的权证。分区注册权证(PRT)是对设备的存取控制权证，是具有先前说明过的图 26 的数据格式结构的权证。

其中，在将分区注册权证(PRT)向权证用户发送时，在公开密钥体制的情况下，分区注册权证(PRT)发行部件(PRT Issuer)的公开密钥证书(CERT_PRTI)也一起发送。PRT 发行部件的公开密钥证书(CERT_PRTI)的属性(Attribute)与分区注册权证(PRT)发行部件(PRT Issuer)的标识符(PRTIC)一致。

接收到分区注册权证(PRT)(S412)的设备执行接收到的权证(PRT)的完整性和用户检查处理(S413)。权证完整性验证处理应用基于对称密钥体制的 MAC 验证、或基于公开密钥体制的签名验证处理中的某一种来执行。用户检查是检查发送来权证的机器(权证用户)的完整性的处理，在相互鉴别已成立时，作为验证对方鉴别者的标识数据、和权证中记录的权证用户标识符(参照图 26)是否一致等的处理来执行。后面将详述这些处理。

在设备中，在所接收权证(PRT)的完整性和用户检查处理的结果是未能确认权证及用户合法的情况下(S414 中为“否”)，将分区注册权证(PRT)受理出错通知给分区创建/删除装置(S418)。在得以确认权证及用户合法的情况下(S414 中为“是(Yes)”)，根据接收到的分区注册权证(PRT)上记述的规则来执行设备内的存储部中的分区创建、或删除处理。后面还将用别的流程来详述该处理。

根据分区注册权证(PRT)的记述，分区的创建或删除处理成功(S416 中为“是”)后，将 PRT 受理成功通知给分区创建/删除装置(S417)。而在分区的创建或删除处理失败(S416 中为“否”)的情况下，将 PRT 受理出错通知给分区创建/删除装置(S418)。

分区创建/删除装置接收 PRT 受理结果(S404)，判定 PRT 处理结果，在 PRT 受理结果是出错的情况下(S405 中为“否”)，作为出错而结束处理；而在 PRT 受理结果是成功(S405 中为“是”)、处理是分区

创建的情况下，执行分区初始数据写入处理(S406、S419)。后面将用别的流程来详述初始数据写入处理。在分区初始数据写入处理结束的情况下、及PRT受理结果是成功(S405中为“是”)、处理是分区删除的情况下，执行会话清除命令的发送接收(S407、S420)，抛弃设备一侧生成的鉴别表(S421)，结束处理。鉴别表是在步骤S401、S410的相互鉴别处理中生成的表，后面将详述它。

这样利用分区注册权证(PRT)，在设备内执行新的分区的创建、或已创建的分区的删除。以下，详述该处理中包含的相互鉴别处理(S401、S410)、权证完整性和用户检查(S413)、分区创建、删除处理(S415)、分区初始数据写入处理(S406、S419)各处理。

(相互鉴别处理)

下面说明图47的步骤S401、S410中执行的相互鉴别处理。其中，以下说明的相互鉴别处理是在使用其他权证、即文件注册权证(FRT: File Registration Ticket)、服务许可权证(SPT: Service Permission Ticket)、数据更新权证(DUT: Data Update Ticket)的设备存取处理中也可在必要时适当进行的处理。

相互鉴别处理的处理流程示于图48。在图48中，左侧示出分区管理器的鉴别装置的处理，右侧示出设备(参照图5)的处理。其中，分区管理器的鉴别装置是可对设备进行数据读取写入处理的装置(例如，作为设备存取机器的读写器、PC)，具有与图10的读写器相当的结构。

在图48的步骤S431中，鉴别装置从权证中读出并决定利用分区注册权证(PRT)所需的鉴别方式。其中，鉴别形态不限于权证记载的鉴别方式，例如也可以根据存取机器(例如，读写器)指定的方式来决定设备鉴别、分区鉴别。

设决定的鉴别方式为A(1)～A(n)。在应用分区注册权证(PRT)的分区设定注册、或删除处理中，设定各种鉴别处理形态。例如分区设定注册需要以设备为对象的设备鉴别；而在分区删除的情况下，需要设备鉴别、和待删除分区鉴别两者等。这样，能够按照处理将需要某一种鉴别、或两种鉴别的设定记述在分区注册权证(PRT)上。PRT利用处理中需要的鉴别方式被记述在分区注册权证(PRT)的[Authentication Type]中，鉴别装置从权证中读出并决定利用分区注册权证(PRT)所需的鉴别方式，将鉴别处理过程定义为A(i):A(1)～

A(n)。

在步骤S432中，读出第一个鉴别处理方式A(1)，判别A(1)的鉴别方式是设备鉴别、还是分区鉴别(S433)，如果是设备鉴别则执行设备鉴别(S434、S441)，而如果是分区鉴别则执行分区鉴别(S435、S442)。在与设备的鉴别结果是鉴别未成功的情况下，作为出错而结束处理。在鉴别成功的情况下，在步骤S437中递增i并返回到步骤S433，判别下一鉴别方式，根据方式来执行鉴别。对A(1)～A(n)执行这些处理，在所有鉴别都成功的情况下进至下一步骤。

根据图49的流程来说明设备鉴别处理。在图49中，左侧示出分区管理器的设备鉴别装置的处理，右侧示出设备(参照图5)的处理。其中，分区管理器的鉴别装置是可对设备进行数据读取写入处理的装置(例如，作为设备存取机器的读写器、PC)，具有与图10的作为设备存取机器的读写器相当的结构。

设备鉴别装置在步骤S451中，根据分区注册权证(PRT)来判定在设备鉴别处理中是否应用使用公开密钥的公开密钥鉴别方式。设备鉴别以公开密钥体制或对称密钥体制中的某一种来执行，在权证中记述了对该执行体制的指定。

在指定了对称密钥体制的情况下，不执行图49的步骤S452～S455、S461～S465的处理，进至步骤S456。在指定了公开密钥体制的情况下，设备鉴别装置在步骤S452中将公开密钥设备鉴别开始命令发送到设备。设备接收到命令(S461)后，参照设备的存储部的公开密钥类设备密钥定义块(参照图16)，来验证是否保存有设备公开密钥证书(CERT DEV)(S462)。在未保存设备公开密钥证书(CERT DEV)的情况下，不能执行公开密钥体制的相互鉴别，判定为出错而结束处理。

如果确认为保存有设备公开密钥证书(CERT DEV)，则在步骤S453、S463中，用设备管理器认证机构(CA(DEV))发行的公开密钥证书来执行相互鉴别及密钥共享处理。

用图50来说明基于公开密钥体制的相互鉴别及密钥共享处理。图50示出作为公开密钥加密体制的使用160比特长的椭圆曲线加密(ECC)的相互鉴别序列。在图50中，公开密钥加密体制采用ECC，但是也可以应用公开密钥体制的其他体制。此外，密钥长度也可以不是160比特。在图50中，首先，B生成64比特的随机数Rb，发送到A。接收到

它的 A 新生成 64 比特的随机数 Ra 及比特征值 p 小的随机数 Ak。然后，求基点 G 的 Ak 倍的点 Av=Ak×G，对 Ra、Rb、Av(X 坐标和 Y 坐标)生成电子签名 A. Sig，与 A 的公开密钥证书一起发回到 B。这里，Ra 及 Rb 分别为 64 比特，Av 的 X 坐标和 Y 坐标分别为 160 比特，所以对合计 448 比特生成电子签名。

在利用公开密钥证书时，用户使用自己保存的公开密钥证书发行机构(CA)的公开密钥，来验证该公开密钥证书的电子签名，在电子签名的验证成功后，从公开密钥证书中取出公开密钥，利用该公开密钥。因此，利用公开密钥证书的所有用户都需要保存通用的公开密钥证书发行机构(CA)的公开密钥。其中，电子签名的验证方法已用图 13 说明过了，所以省略其细节。

接收到 A 的公开密钥证书、Ra、Rb、Av、电子签名 A. Sig 的 B 验证 A 发送来的 Rb 与 B 生成的是否一致。在其结果是一致的情况下，用认证机构的公开密钥来验证 A 的公开密钥证书内的电子签名，取出 A 的公开密钥。然后，用取出的 A 的公开密钥来验证电子签名 A. Sig。在电子签名的验证成功后，B 将 A 鉴别为合法的。

接着，B 生成比特征值 p 小的随机数 Bk。然后，求基点 G 的 Bk 倍的点 Bv=Bk×G，对 Rb、Ra、Bv(X 坐标和 Y 坐标)生成电子签名 B. Sig，与 B 的公开密钥证书一起发回到 A。这里，RB 及 Rb 分别为 64 比特，Bv 的 X 坐标和 Y 坐标分别为 160 比特，所以对合计 448 比特生成电子签名。

接收到 B 的公开密钥证书、Rb、Ra、Bv、电子签名 B. Sig 的 A 验证 B 发送来的 Ra 与 A 生成的是否一致。在其结果是一致的情况下，用认证机构的公开密钥来验证 B 的公开密钥证书内的电子签名，取出 B 的公开密钥。然后，用取出的 B 的公开密钥来验证电子签名 B. Sig。在电子签名的验证成功后，A 将 B 鉴别为合法的。

在两者的鉴别都成功的情况下，B 计算 Bk×Av(Bk 是随机数，而 Av 是椭圆曲线上的点，所以需要计算椭圆曲线上的点的标量倍)，A 计算 Ak×Bv，将这些点的 X 坐标的低 64 比特作为会话密钥用于以后的通信(在对称密钥加密为 64 比特密钥长度的对称密钥加密的情况下)。当然，也可以由 Y 坐标来生成会话密钥，也可以不是低 64 比特。其中，在相互鉴别后的秘密通信中，发送数据有时不仅用会话密钥来加密，

而且还附加电子签名。

在验证电子签名或验证接收数据时，在发现非法、不一致的情况下，认为相互鉴别失败，中断处理。

在这种相互鉴别处理中，用生成的会话密钥对发送数据进行加密，相互执行数据通信。

返回到图 49 来继续说明流程。在步骤 S453、S463 中如果上述相互鉴别、密钥共享处理成功，则设备在步骤 S464 中参照设备的存储部的设备密钥区域(参照图 18)中保存的 CRL_DEV“注册有作废设备(Device)、作废机器(作为设备存取机器的读写器、PC 等权证用户、权证发行部件)的公开密钥证书标识符(例如，序列号：SN)的作废表(Revocation List (Certificate))”，来验证作为通信对方的设备鉴别装置是否未作废。在已作废的情况下，不能许可分区创建处理，所以作为出错而结束处理。

在未作废的情况下，在步骤 S465 中，将相互鉴别及密钥共享处理中生成的会话密钥 Kses、和通信对方(构成设备鉴别装置的作为设备存取机器的读写器、PC 等)的公开密钥证书中的识别名(DN：Distinguished Name)、序列号、范畴保存到以设备管理器代码(DMC)为关键字来相对应的鉴别表中。

另一方面，设备鉴别装置也在步骤 S454 中参照 CRL_DEV“注册有作废设备(Device)、作废机器(作为设备存取机器的读写器、PC 等权证用户、权证发行部件)的公开密钥证书标识符(例如，序列号：SN)的作废表(Revocation List (Certificate))”，来判定设备是否未作废。设备鉴别装置可从注册机构(RA(PAR))取得作废表(CRL_DEV)。在已作废的情况下，不能许可分区创建处理，所以作为出错而结束处理。

在未作废的情况下，在步骤 S455 中，将相互鉴别及密钥共享处理中生成的会话密钥 Kses、和通信对方(设备)的公开密钥证书中的识别名(DN：Distinguished Name)、序列号、范畴保存到以设备管理器代码(DMC)为关键字来相对应的鉴别表中。

图 51 示出设备内生成的鉴别表的例子，而图 52 示出作为鉴别装置的作为设备存取机器的读写器(也可以是 PC)生成的鉴别表的例子。

图 51 是设备鉴别、及作为后面将说明的分区鉴别的分区 1、2 的鉴别结束时设备内生成的鉴别表的例子。作为组(Group)，在设备鉴别

的情况下记录设备管理器代码(DMC)，而在分区鉴别的情况下记录分区管理器代码(PMC)，根据执行的各鉴别方式来保存各个数据。在公开密钥鉴别方式的情况下，如前所述，将会话密钥Kses、和通信对方(作为设备存取机器的读写器)的公开密钥证书中的识别名(DN: Distinguished Name)、序列号、范畴保存到表中；而在对称密钥鉴别的情况下，保存会话密钥Kses、和通信对方(作为设备存取机器的读写器)的标识符(ID RW)。

鉴别表在清除会话时被抛弃。设备能够通过参照表内的信息来确认设备及各分区的鉴别状态，可确认应使用的会话密钥。

另一方面，图52示出作为设备存取机器的读写器一侧的鉴别表。该例也是设备鉴别、及作为分区鉴别的分区1、2的鉴别结束时的鉴别表的例子。基本结构与设备内的鉴别表相同，作为组(Group)，在设备鉴别的情况下记录设备管理器代码(DMC)，而在分区鉴别的情况下记录分区管理器代码(PMC)，根据执行的各鉴别方式来保存各个数据。在公开密钥鉴别方式的情况下，如前所述，将会话密钥Kses、和通信对方(设备)的公开密钥证书中的识别名(DN: Distinguished Name)、序列号、范畴保存到表中；而在对称密钥鉴别的情况下，保存会话密钥Kses、和通信对方(设备)的标识符(ID RW)。读写器一侧的鉴别表也在清除会话时被抛弃。在作为设备存取机器的读写器一侧，也可通过参照表内的信息来判定设备及各分区的鉴别状态的有无，可确认应使用的会话密钥。

返回到图49，继续说明设备鉴别处理。设备鉴别装置在步骤S451中判定为设备鉴别方式不是公开密钥体制后，设备鉴别装置在步骤S456中，将对称密钥设备鉴别命令输出到设备。设备接收到命令(S466)后，参照设备的存储部的对称密钥类设备密钥定义块(参照图16)来验证是否保存有对称密钥鉴别中使用的双向个别密钥鉴别主密钥(MKauth_DEV)(S467)。在未保存双向个别密钥鉴别主密钥(MKauth_DEV)的情况下，不能执行对称密钥体制的相互鉴别，判定为出错，结束处理。

如果确认为保存有双向个别密钥鉴别主密钥(MKauth_DEV)，则在步骤S457、S468中，使用主密钥来执行相互鉴别及密钥共享处理。

用图53来说明使用主密钥的基于对称密钥体制的相互鉴别及密钥

共享处理。在图 53 中，A 及 B 是使用主密钥来执行对称密钥体制的鉴别的实体，A 具有自己的标识符 IDa、双向个别密钥鉴别对称密钥 Ka、双向个别密钥鉴别主密钥 MKb，而 B 具有自己的标识符 IDb、双向个别密钥鉴别对称密钥 Kb、双向个别密钥鉴别主密钥 MKa。其中，在图 53 的例子中，对称密钥加密体制采用 DES 算法(例如，DES、三重 DES)，但是只要是同样的对称密钥加密体制即可，也可应用其他加密体制。

首先，B 生成 64 比特的随机数 Rb，将 Rb 及自己的 ID---IDb 发送到 A。接收到它的 A 新生成 64 比特的随机数 Ra，通过用双向个别密钥鉴别主密钥 MKb 对 IDb 进行 DES 加密处理来取得双向个别密钥鉴别对称密钥 Kb。进而，用密钥 Ka、Kb，按 Ra、Rb、IDa、IDb 的顺序，用 DES 的 CBC 模式对数据进行加密，与自己的标识符 IDa 一起发回到 B。

接收到它的 B 首先通过用双向个别密钥鉴别主密钥 MKa 对 IDa 进行 DES 加密处理来取得双向个别密钥鉴别对称密钥 Ka。进而，用密钥 Ka、Kb 对接收数据进行解密。在解密得到的 Ra、Rb、IDa、IDb 内，验证 Rb 及 IDb 与 B 发送的是否一致。在通过该验证的情况下，B 将 A 鉴别为合法的。

接着，B 生成用作会话密钥的 64 比特的随机数 Kses，用密钥 Kb、Ka，按 Rb、Ra、IDb、IDa、Kses 的顺序，用 DES 的 CBC 模式对数据进行加密，发回到 A。

接收到它的 A 用密钥 Ka、Kb 对接收数据进行解密。验证解密得到的 Ra、Rb、IDa、IDb 与 A 发送的是否一致。在通过该验证的情况下，A 将 B 鉴别为合法的。在相互鉴别了对方后，会话密钥 Kses 被用作用于鉴别后的秘密通信的对称密钥。

其中，在验证接收数据时，在发现非法、不一致的情况下，认为相互鉴别失败，中断处理。

对于本发明的系统的使用与保存数据相对应的主密钥的对称密钥鉴别，数据的流程的说明图示于图 54。如图 54 所示，作为设备存取机器的读写器(R/W)生成 64 比特的随机数 Rb,将 Rb 及自己的 ID---IDrw 发送到设备(Device)。接收到它的设备(Device)新生成 64 比特的随机数 Ra，通过用双向个别密钥鉴别主密钥 MKauth_DEV_A 对 IDrw 进行 DES 加密处理来取得双向个别密钥鉴别对称密钥 Kauth_DEV_A。进而，用密钥 Kauth_DEV_A、Kauth_DEV_B，按 Ra、Rb、IDrw 的顺序，作为

加密算法，例如用 DES-CBC 模式对数据进行加密，与自己的标识符 IDm 一起发回到作为设备存取机器的读写器(R/W)。

接收到它的读写器(R/W)首先通过用双向个别密钥鉴别主密钥 MKauth_DEV_B 对 IDm 进行 DES 加密处理来取得双向个别密钥鉴别对称密钥 Kauth_DEV_B。进而，用密钥 Kauth_DEV_A、Kauth_DEV_B 对接收数据进行解密。验证解密得到的 Rb 及 IDrw 与作为设备存取机器的读写器(R/W)发送的是否一致。在通过该验证的情况下，读写器(R/W)将设备(Device)鉴别为合法的。

接着，读写器(R/W)生成用作会话密钥的 64 比特的随机数 Kses，用双向个别密钥鉴别对称密钥 Kauth_DEV_A、Kauth_DEV_B，按 Rb、Ra、Kses 的顺序，用作为 DES 算法的例如三重 DES 模式对数据进行加密，发回到设备(Device)。

接收到它的设备用双向个别密钥鉴别对称密钥 Kauth_DEV_A、Kauth_DEV_B 对接收数据进行解密。验证解密得到的 Ra、Rb、IDrw 与设备(Device)发送的是否一致。在通过该验证的情况下，设备(Device)将读写器(R/W)鉴别为合法的，在鉴别后，将会话密钥 Kses 用作用于秘密通信的对称密钥。

其中，作为设备唯一值的 IDm 可以如先前使用图 6 的设备存储格式所说明的那样，应用基于设备管理器管理区域的保存数据的值。

如上所述，根据通过使用主密钥的对称密钥体制来执行的相互鉴别及密钥共享处理，将执行基于通信对方的主密钥的处理而生成的通信对方的个别密钥、和自己的个别密钥这 2 个密钥设定为对称密钥，用设定的 2 个密钥来执行基于对称密钥体制的相互鉴别，所以与在设备或存取装置中预先保存对称密钥的现有对称密钥鉴别结构相比，能实现更安全的鉴别系统及方法。

返回到图 49 来继续说明流程。在步骤 S457、S468 中如果上述相互鉴别、密钥共享处理成功，则设备在步骤 S469 中参照设备的存储部的设备密钥区域(参照图 18)中保存的 IRL_DEV“注册有作废设备(Device)、作废机器(作为设备存取机器的读写器、PC 等权证用户、权证发行部件)的标识符(ID)的作废表(Revocation List (ID))”，来验证作为通信对方的设备鉴别装置是否未作废。在已作废的情况下，不能许可分区创建处理，所以作为出错而结束处理。

在未作废的情况下，在步骤S470中，将相互鉴别及密钥共享处理中生成的会话密钥Kses、和通信对方(构成设备鉴别装置的作为设备存取机器的读写器、PC 等)的标识信息(IDrw)保存到以设备管理器代码(DMC)为关键字来相对应的鉴别表(参照图 51)中。

另一方面，设备鉴别装置也在步骤S458中参照IRL_DEV“注册有作废设备(Device)、作废机器(作为设备存取机器的读写器、PC 等权证用户、权证发行部件)的标识符(ID)的作废表(Revocation List (ID))”,来判定设备是否未作废。设备鉴别装置可从注册机构(RA(PAR))取得作废表(IRL_DEV)。在已作废的情况下，不能许可分区创建处理，所以作为出错而结束处理。

在未作废的情况下，在步骤S459中，将相互鉴别及密钥共享处理中生成的会话密钥 Kses、和通信对方(设备)的标识信息(IDm)保存到以设备管理器代码(DMC)为关键字来相对应的鉴别表中(参照图 52)。

以上处理是在分区管理器管辖的作为设备存取机器的读写器和设备间执行的设备鉴别处理。

接着，用图 55、图 56 来说明图 48 的步骤S435、S442中执行的分区鉴别处理。在应用分区注册权证的分区设定注册、或删除处理中，如先前说明过的那样，需要按照处理来进行设备鉴别、分区鉴别中的某一种、或两种鉴别。这些设定被记述在分区注册权证(PRT)上。在分区注册权证(PRT)上有执行分区鉴别的记述的情况下，执行分区鉴别。

下面说明图 55、图 56 的处理流程的各步骤。在图 55 中，左侧示出分区管理器的分区鉴别装置的处理，右侧示出设备(参照图 5)的处理。其中，分区管理器的分区鉴别装置是可对设备进行数据读取写入处理的装置(例如，作为设备存取机器的读写器、PC)，具有与图 10 的作为设备存取机器的读写器相当的结构。

分区鉴别装置在步骤S471中，输出分区 A 存在检查命令来对设备执行待鉴别分区 A 的存在确认。接收到命令(S481)的设备检查在设备的存储部内是否存在分区 A(S482)。这里，分区的标识符 A 例如使用分区管理器代码(PMC)，设备可以例如根据分区定义块(PDB: Partition Definition Block)保存的 PMC 来判定有无分区。设备判定出有无分区后，将检查结果发送到分区鉴别装置。

接收到(S472)检查结果(S472)的分区鉴别装置验证检查结果

(S473)，在接收到不存在分区的结果的情况下，不可鉴别，所以出错结束。在检查结果表示存在分区的情况下，分区鉴别装置在步骤 S474 中根据分区注册权证(PRT)来判定在分区鉴别处理中是否应用使用公开密钥的公开密钥鉴别方式。分区鉴别与前述设备鉴别同样，以公开密钥体制或对称密钥体制中的某一种来执行，在权证中记述了对该执行体制的指定。

在指定了对称密钥体制的情况下，不执行图 55 的步骤 S475～S478、S484～S488 的处理，进至步骤 S491。在指定了公开密钥体制的情况下，分区鉴别装置在步骤 S475 中将公开密钥分区 A 鉴别开始命令发送到设备。设备接收到命令(S484)后，参照设备的存储部的公开密钥类分区密钥定义块(参照图 21)，来验证是否保存有分区公开密钥证书(CERT PAR)(S485)。在未保存分区公开密钥证书(CERT PAR)的情况下，不能执行公开密钥体制的相互鉴别，判定为出错而结束处理。

如果确认为保存有分区公开密钥证书(CERT PAR)，则在步骤 S476、S486 中，用分区管理器认证机构(CA(PAR))发行的公开密钥证书来执行相互鉴别及密钥共享处理。

基于公开密钥体制的相互鉴别及密钥共享处理与先前在设备鉴别处理中说明过的图 50 所示的序列相同，所以省略其说明。只是，分区鉴别中利用的公开密钥证书是分区管理器认证机构(CA(PAR))发行的公开密钥证书，在该公开密钥证书的签名验证中，使用分区管理器认证机构(CA(PAR))的公开密钥(PUB CA(PAR))。公开密钥(PUB CA(PAR))被保存在分区密钥区域(参照图 23)中。在这种相互鉴别处理中，用生成的会话密钥对发送数据进行加密，相互执行数据通信。

在步骤 S476、S486 中，如果根据图 50 所示的序列执行的相互鉴别、密钥共享处理成功，则设备在步骤 S487 中参照设备的存储部的分区密钥区域(参照图 23)中保存的 CRL_PAR“注册有作废设备(Device)、作废机器(作为设备存取机器的读写器、PC 等权证用户、权证发行部件)的公开密钥证书标识符(例如，序列号：SN)的作废表(Revocation List (Certificate))”，来验证作为通信对方的分区鉴别装置是否未作废。在已作废的情况下，不能许可分区创建处理或删除处理，所以作为出错而结束处理。

在未作废的情况下，在步骤 S488 中，将相互鉴别及密钥共享处理

中生成的会话密钥Kses、和通信对方(构成分区鉴别装置的作为设备存取机器的读写器、PC等)的公开密钥证书中的识别名(DN: Distinguished Name)、序列号、范畴保存到以分区管理器代码(PMC)为关键字来相对应的鉴别表中。

另一方面，分区鉴别装置也在步骤S477中参照CRL_PAR“注册有作废设备(Device)、作废机器(作为设备存取机器的读写器、PC等权证用户、权证发行部件)的公开密钥证书标识符(例如，序列号：SN)的作废表(Revocation List (Certificate))”，来判定设备是否未作废。设备鉴别装置可从注册机构(RA(PAR))取得作废表(CRL_PAR)。在已作废的情况下，不能许可分区创建处理或删除处理，所以作为出错而结束处理。

在未作废的情况下，在步骤S478中，将相互鉴别及密钥共享处理中生成的会话密钥Kses、和通信对方(设备)的公开密钥证书中的识别名(DN: Distinguished Name)、序列号、范畴保存到以分区管理器代码(PMC)为关键字来相对应的鉴别表中。其结果是，例如在设备内生成图51所示的鉴别表，而在作为分区鉴别装置的作为设备存取机器的读写器(PC也可)中生成图52所示的鉴别表。图51、图52是设备鉴别、及作为分区鉴别的分区1、2的鉴别结束时设备内及作为设备存取机器的读写器内生成的鉴别表的例子。

在分区鉴别的情况下记录分区管理器代码(PMC)，根据执行的各鉴别方式来保存各个数据。在公开密钥鉴别方式的情况下，如前所述，将会话密钥Kses、和通信对方的公开密钥证书中的识别名(DN: Distinguished Name)、序列号、范畴保存到表中；而在对称密钥鉴别的情况下，保存会话密钥Kses、和通信对方的标识符。鉴别表在清除会话时被抛弃。设备及作为设备存取机器的读写器能够通过参照表内的信息来确认设备及各分区的鉴别状态，可确认应使用的会话密钥。

用图55、图56的流程来继续说明分区鉴别处理。分区鉴别装置在步骤S474中判定为分区鉴别方式不是公开密钥体制后，分区鉴别装置在步骤S491中将对称密钥分区A鉴别命令输出到设备。设备接收到命令(S501)后，参照设备的存储部的对称密钥类分区密钥定义块(参照图22)来验证是否保存有对称密钥鉴别中使用的双向个别密钥鉴别主密

钥(MKauth_PAR_A)(S502)。在未保存双向个别密钥鉴别主密钥(MKauth_PAR_A)的情况下，不能执行对称密钥体制的相互鉴别，判定为出错，结束处理。

如果确认为保存有双向个别密钥鉴别主密钥(MKauth_PAR_A)，则在步骤 S492、S503 中，使用主密钥来执行相互鉴别及密钥共享处理。基于对称密钥体制的相互鉴别及密钥共享处理与先前的设备鉴别中用图 53、图 54 说明过的序列同样，所以省略其说明。只是，分区鉴别的情况下应用的密钥在分区密钥定义块(参照图 22)中被定义，是分区密钥区域(参照图 23)中保存的双向个别密钥鉴别对称密钥(Kauth_PAR_B)、及双向个别密钥鉴别主密钥(MKauth_PAR_A)。

在步骤 S492、S503 中如果对称密钥体制的相互鉴别、密钥共享处理成功，则设备在步骤 S504 中参照设备的存储部的分区密钥区域(参照图 23)中保存的 IRL_PAR“注册有作废设备(Device)、作废机器(作为设备存取机器的读写器、PC等权证用户、权证发行部件)的标识符(ID)的作废表(Revocation List (ID))”，来验证作为通信对方的分区鉴别装置是否未作废。在已作废的情况下，不能许可分区创建处理或删除处理，所以作为出错而结束处理。

在未作废的情况下，在步骤 S505 中，将相互鉴别及密钥共享处理中生成的会话密钥 Kses、和通信对方(构成设备鉴别装置的作为设备存取机器的读写器、PC 等)的标识信息(IDrw)保存到以分区管理器代码(PMC)为关键字来相对应的鉴别表(参照图 51)中。

另一方面，分区鉴别装置也在步骤 S493 中参照 IRL_PAR“注册有作废设备(Device)、作废机器(作为设备存取机器的读写器、PC 等权证用户、权证发行部件)的标识符(ID)的作废表(Revocation List (ID))”,来判定设备是否未作废.分区鉴别装置可从注册机构(RA(PAR))取得作废表(IRL_PAR)。在已作废的情况下，不能许可分区创建处理或删除处理，所以作为出错而结束处理。

在未作废的情况下，在步骤 S494 中，将相互鉴别及密钥共享处理中生成的会话密钥 Kses、和通信对方(设备)的标识信息(IDm)保存到以分区管理器代码(DMC)为关键字来相对应的鉴别表中(参照图 52)。

以上处理是在分区管理器管辖的作为设备存取机器的读写器和设备间执行的分区鉴别处理。通过这种相互鉴别，设备或分区和作为设

备存取机器的读写器间的鉴别成立，实现了会话密钥的共享，可用会话密钥对通信数据进行加密来进行通信。

其中，上述设备鉴别处理、分区鉴别处理是在使用其他权证、即文件注册权证(FRT: File Registration Ticket)、服务许可权证(SPT:Service Permission Ticket)、数据更新权证(DUT:Data Update Ticket)的设备存取时也可在必要时适当进行的处理。这些将在后面利用各权证的处理的说明中描述。

(权证完整性和用户检查)

接着，用图 57、图 58 的流程来详细说明图 47 的分区创建、删除处理流程中的步骤 S413 的设备中的权证完整性和用户检查处理。其中，以下说明的权证完整性和用户检查处理是在使用其他权证、即文件注册权证(FRT: File Registration Ticket)、服务许可权证(SPT: Service Permission Ticket)、数据更新权证(DUT: Data Update Ticket)的设备存取处理中也可在必要时适当进行的处理，图 57、图 58 的流程是各权证通用的处理流程。

权证完整性和用户检查处理是设备根据从与设备执行通信的权证用户(例如，作为设备存取机器的读写器、PC 等)接收到的权证(参照图 5)来执行的处理。设备在权证完整性和用户检查处理中确认了权证及作为权证用户(例如，作为设备存取机器的读写器、PC 等)的用户的完整性后，许可权证上记述的限制范围内的处理。

用图 57、图 58 来详细说明权证完整性和用户检查处理。从权证用户(例如，作为设备存取机器的读写器、PC 等)接收到权证的设备在图 57 的步骤 S511 中判验证权证类型，判定权证是否是分区注册权证(PRT: Partition Registration Ticket)。权证类型被记录在各权证上(参照图 26、图 27、图 28、图 31、图 32)。

在权证类型是分区注册权证(PRT: Partition Registration Ticket)的情况下，执行步骤 S512～S514；而在权证类型不是分区注册权证(PRT: Partition Registration Ticket)的情况下，进至步骤 S515。

在权证类型是分区注册权证(PRT: Partition Registration Ticket)的情况下，在步骤 S512 中，判定权证上记述的 Integrity Check Type(权证(Ticket)的完整性验证值的类型(公开密钥体制

(Public)/对称密钥体制(Common))的设定是否是公开密钥体制(Public)。

在完整性验证值的类型(Integrity Check Type)是公开密钥体制(Public)的情况下，进至步骤 S513，执行各种处理。步骤 S513 中执行的处理首先是使用设备管理器认证机构(CA(DEV))的公开密钥 PUB CA(DEV)的权证发行者(Ticket Issuer)公开密钥证书(CERT)验证处理。

如前所述，在将分区注册权证(PRT)发行部件(PRT Issuer)发行的权证(Ticket)向权证用户发送时，在公开密钥体制的情况下，分区注册权证(PRT)发行部件(PRT Issuer)的公开密钥证书(CERT_PRTI)也一起被发送到设备。其中，PRT发行部件的公开密钥证书(CERT_PRTI)的属性(Attribute)与分区注册权证(PRT)发行部件(PRT Issuer)的标识符(PRTIC)一致。

在公开密钥证书(参照图 11)上附加有用设备管理器认证机构(CA(DEV))的私有密钥执行的签名，用设备管理器认证机构(CA(DEV))的公开密钥 PUB CA(DEV)来验证该签名。签名生成、验证例如作为根据先前说明过的图 12、图 13 的流程执行的处理来执行。通过该签名验证，来判定权证发行者的公开密钥证书(CERT)是否是未窜改过的合法的公开密钥证书(CERT)。

进而，在步骤 S513 中，判定通过签名验证确认了完整性的权证发行部件的公开密钥证书(CERT)的选项区域中记录的作为用户的范畴的代码与设备内的 DKDB (Device Key Definition Block)(PUB)中记录的权证发行部件代码(PRTIC: PRT Issuer Code)是否一致。

如图 11 的公开密钥证书的说明一栏中记述的那样，在公开密钥证书中，记录有作为各权证(PRT、FRT、SPT 等)的发行部件的权证发行部件(Ticket Issuer)的所属组代码，在此情况下，为 PRTIC (PRT Issuer Code)。通过确认该选项区域的代码和设备内的 DKDB (Device Key Definition Block)(PUB)中记录的权证发行部件代码(PRTIC: PRT Issuer Code)一致，来确认所接收权证(PRT)是合法的权证发行部件发行的权证。

进而，设备参照设备的存储部内的设备密钥区域(参照图 18)中保存的 CRL_DEV“注册有作废设备(Device)、作废机器(作为设备存取机

器的读写器、PC等权证用户、权证发行部件)的公开密钥证书标识符(例如，序列号：SN)的作废表(Revocation List (Certificate))”，来验证权证发行部件(Ticket Issuer)是否未作废。

进而，执行所接收权证---分区注册权证(PRT)(参照图26)上记录的签名、即Integrity Check Value(权证(Ticket)的完整性验证值(公开密钥体制：签名(Signature)))的验证，确认权证是否未被窜改。签名验证与先前的公开密钥证书的签名验证同样，例如根据与图13的流程同样的序列来执行。

以上，(1)权证发行者(Ticket Issuer)的公开密钥证书(CERT)是未窜改过的合法的公开密钥证书(CERT)；(2)权证发行者(Ticket Issuer)的公开密钥证书(CERT)的选项区域中记录的代码、和设备内的DKDB (Device Key Definition Block)(PUB)中记录的权证发行部件代码(PRTIC：PRT Issuer Code)一致；(3)权证发行部件(Ticket Issuer)未作废；(4)通过验证所接收权证(PRT)的签名(Signature)而确认权证没有窜改。以以上全部被确认为条件而认为权证完整性验证成功。在上述(1)～(4)中某一个未被确认的情况下，判定为不能得到权证完整性的确认，中止利用分区注册权证(PRT：Partition Registration Ticket)的处理。

此外，在步骤S512中，在判定为权证上记述的Integrity Check Type(权证(Ticket)的完整性验证值的类型(公开密钥体制(Public)/对称密钥体制(Common)))的设定是对称密钥体制(Common)的情况下，进至步骤S514，执行MAC (Message Authentication Code)验证。设备使用设备的设备密钥区域(参照图18)中保存的分区注册权证(PRT)的MAC验证密钥“Kprt”来执行权证的MAC验证处理。

图59示出使用DES加密处理结构的MAC值生成例。如图59的结构所示，以8个字节为单位对作为对象的报文进行分割，(以下，设分割所得的报文为M1、M2、…、MN)，首先，对初值(Initial Value(以下，记作IV))和M1进行逻辑“异或”(设其结果为I1)。接着，将I1输入到DES加密部中，用MAC验证密钥“Kprt”来进行加密(设输出为E1)。接着，对E1及M2进行逻辑“异或”，将其输出I2输入到DES加密部中，用密钥Kprt进行加密(输出E2)。以下，重复这种处理，对所有报文实施加密处理。最后出来的EN为报文鉴别码(MAC (Message

Authentication Code))。其中，作为报文，可使用构成待验证数据的部分数据。

对已保证没有窜改的例如数据发送端在生成数据时生成的 ICV (Integrity Check Value)、和数据接收端根据接收数据而生成的 ICV 进行比较，如果得到同一 ICV，则保证数据没有窜改，而如果 ICV 不同，则判定为有窜改。已保证没有窜改的例如数据发送端在生成数据时生成的 ICV 如与图 26 的分区注册权证(PRT)的格式有关的记述中说明过的那样，被保存在 PRT 的 ICV (Integrity Check Value)字段中。对设备生成的 ICV 和所接收权证(PRT)中保存的 ICV 进行比较，如果一致，则判定为权证有完整性；而在不一致的情况下，判定为权证有窜改，中止利用权证的处理。

通过上述处理来完成权证上记述的 Integrity Check Type 是对称密钥体制的情况下的权证验证处理。

返回到图 57 的流程，继续说明权证完整性和用户检查处理。在步骤 S511 中，在判定为权证类型不是分区注册权证(PRT: Partition Registration Ticket)的情况下，在步骤 S515 中，验证权证类型，判定权证是否是文件注册权证(FRT: File Registration Ticket)。

在权证类型是文件注册权证(FRT: File Registration Ticket)的情况下，执行步骤 S516～S518；而在权证类型不是文件注册权证(FRT: File Registration Ticket)的情况下，进至步骤 S519。

在权证类型是文件注册权证(FRT: File Registration Ticket)的情况下，在步骤 S516 中，判定权证上记述的 Integrity Check Type(权证(Ticket)的完整性验证值的类型(公开密钥体制(Public)/对称密钥体制(Common)))的设定是否是公开密钥体制(Public)。

在完整性验证值的类型(Integrity Check Type)是公开密钥体制(Public)的情况下，进至步骤 S517，执行各种处理。步骤 S517 中执行的处理首先是使用分区管理器认证机构(CA(PAR))的公开密钥 PUB CA(PAR)的权证发行者(Ticket Issuer)公开密钥证书(CERT)验证处理。

在将文件注册权证(FRT)发行部件(FRT Issuer)发行的权证(Ticket)向权证用户发送时，在公开密钥体制的情况下，文件注册权证(FRT)发行部件(FRT Issuer)的公开密钥证书(CERT_FRTI)也一起被

发送到设备。其中，FRT发行部件的公开密钥证书(CERT_FRTI)的属性(Attribute)与文件注册权证(FRT)发行部件(FRT Issuer)的标识符(FRTIC)一致。

在公开密钥证书(参照图 11)上附加有用分区管理器认证机构(CA(PAR))的私有密钥执行的签名，用分区管理器认证机构(CA(PAR))的公开密钥 PUB CA(PAR)来验证该签名。签名生成、验证例如作为根据先前说明过的图 12、图 13 的流程执行的处理来执行。通过该签名验证，来判定权证发行者的公开密钥证书(CERT)是否是未窜改过的合法的公开密钥证书(CERT)。

进而，在步骤 S517 中，判定通过签名验证确认了完整性的权证发行部件的公开密钥证书(CERT)的选项区域中记录的用户所属组代码、和设备内的 PKDB (Partition Key Definition Block)(PUB)中记录的权证发行部件代码(FRTIC: FRT Issuer Code)是否一致。

如图 11 的公开密钥证书的说明一栏中记述的那样，在公开密钥证书中，记录有作为各权证(PRT、FRT、SPT 等)的发行部件的权证发行部件(Ticket Issuer)的所属组代码，在此情况下，为 FRTIC (FRT Issuer Code)。通过确认该选项区域的代码和设备内的 PKDB (Partition Key Definition Block)(PUB)中记录的权证发行部件代码(FRTIC: FRT Issuer Code)一致，来确认所接收权证(FRT)是合法的权证发行部件发行的权证。

进而，设备参照设备的存储部内的分区密钥区域(参照图 23)中保存的 CRL_PAR(注册有作废设备(Device)、作废机器(作为设备存取机器的读写器、PC 等权证用户、权证发行部件)的公开密钥证书标识符(例如，序列号: SN)的作废表(Revocation List (Certificate)))，来验证权证发行部件(Ticket Issuer)是否未作废。

进而，执行所接收权证文件注册权证(FRT)(参照图 27)上记录的签名、即 Integrity Check Value(权证(Ticket)的完整性验证值(公开密钥体制: 签名(Signature)))的验证，确认权证是否未被窜改。签名验证与先前的公开密钥证书的签名验证同样，例如根据与图 13 的流程同样的序列来执行。

以上，(1)权证发行者(Ticket Issuer)的公开密钥证书(CERT)是未窜改过的合法的公开密钥证书(CERT)；(2)权证发行者(Ticket

Issuer)的公开密钥证书(CERT)的选项区域中记录的代码、和设备内的PKDB (Partition Key Definition Block)(PUB)中记录的权证发行部件代码(FRTIC: FRT Issuer Code)一致; (3)权证发行部件(Ticket Issuer)未作废; (4)通过验证所接收权证(FRT)的签名(Signature)而确认权证没有窜改. 以以上全部被确认为条件而认为文件注册权证(FRT)的完整性验证成功. 在上述(1)~(4)中某一个未被确认的情况下, 判定为不能得到文件注册权证(FRT)的完整性的确认, 中止利用文件注册权证(FRT)的处理.

此外, 在步骤S516中, 在判定为权证上记述的Integrity Check Type(权证(Ticket)的完整性验证值的类型(公开密钥体制(Public)/对称密钥体制(Common))的设定是对称密钥体制(Common)的情况下, 进至步骤S518, 执行MAC (Message Authentication Code)验证. 设备使用设备的分区密钥区域(参照图23)中保存的文件注册权证(FRT)的MAC验证密钥"Kfrt"来执行权证的MAC验证处理. MAC验证处理根据先前说明过的使用图59的DES加密处理结构的MAC值生成处理来执行.

对已保证没有窜改的例如数据发送端在生成数据时生成的ICV (Integrity Check Value)、和数据接收端根据接收数据而生成的ICV进行比较, 如果得到同一ICV, 则保证数据没有窜改, 而如果ICV不同, 则判定为有窜改. 已保证没有窜改的例如数据发送端在生成数据时生成的ICV如与图27的文件注册权证(FRT)的格式有关的记述中说明过的那样, 被保存在FRT的ICV (Integrity Check Value)字段中. 对设备生成的ICV和所接收权证(FRT)中保存的ICV进行比较, 如果一致, 则判定为权证有完整性; 而在不一致的情况下, 判定为权证有窜改, 中止利用权证的处理.

通过上述处理来完成权证上记述的Integrity Check Type是对称密钥体制的情况下的文件注册权证(FRT)验证处理.

在步骤S515中, 在判定为权证类型不是文件注册权证(FRT: File Registration Ticket)的情况下, 在步骤S519中, 验证权证类型, 判定权证是否是服务许可权证(SPT: Service Permission Ticket).

在权证类型是服务许可权证(SPT: Service Permission Ticket)的情况下, 执行步骤S520~S522; 而在权证类型不是服务许可权证

(SPT: Service Permission Ticket)的情况下，进至步骤S523.

在权证类型是服务许可权证(SPT: Service Permission Ticket)的情况下，在步骤 S520 中，判定权证上记述的 Integrity Check Type(权证(Ticket)的完整性验证值的类型(公开密钥体制(Public)/对称密钥体制(Common)))的设定是否是公开密钥体制(Public).

在完整性验证值的类型(Integrity Check Type)是公开密钥体制(Public)的情况下，进至步骤 S521，执行各种处理. 步骤 S521 中执行的处理首先是使用分区管理器认证机构(CA(PAR))的公开密钥 PUB CA(PAR)的权证发行者(Ticket Issuer)公开密钥证书(CERT)验证处理.

在将服务许可权证(SPT)发行部件(SPT Issuer)发行的权证(Ticket)向权证用户发送时，在公开密钥体制的情况下，服务许可权证(SPT)发行部件(SPT Issuer)的公开密钥证书(CERT_SPTI)也一起被发送到设备. 其中，SPT 发行部件的公开密钥证书(CERT_SPTI)的属性(Attribute)与服务许可权证(SPT)发行部件(SPT Issuer)的标识符(SPTIC)一致.

在公开密钥证书(参照图 11)上附加有用分区管理器认证机构(CA(PAR))的私有密钥执行的签名，用分区管理器认证机构(CA(PAR))的公开密钥 PUB CA(PAR)来验证该签名. 签名生成、验证例如作为根据先前说明过的图 12、图 13 的流程执行的处理来执行. 通过该签名验证，来判定权证发行者的公开密钥证书(CERT)是否是未窜改过的合法的公开密钥证书(CERT).

进而，在步骤 S521 中，判定通过签名验证确认了完整性的权证发行部件的公开密钥证书(CERT)的选项区域中记录的用户所属组代码、和设备内的文件定义块 (FDB: File Definition Block)中记录的权证发行部件代码(SPTIC: SPT Issuer Code)是否一致.

如图 11 的公开密钥证书的说明一栏中记述的那样，在公开密钥证书中，记录有作为各权证(PRT、FRT、SPT 等)的发行部件的权证发行部件(Ticket Issuer)的所属组代码，在此情况下，为 SPTIC (SPT Issuer Code). 通过确认该选项区域的代码和设备内的 FDB (File Definition Block)中记录的权证发行部件代码(SPTIC: SPT Issuer Code)一致，来确认所接收权证(SPT)是合法的权证发行部件发行的权

证。

进而，设备参照设备的存储部内的分区密钥区域(参照图 23)中保存的 CRL_PAR(注册有作废设备(Device)、作废机器(作为设备存取机器的读写器、PC 等权证用户、权证发行部件)的公开密钥证书标识符(例如，序列号：SN)的作废表(Revocation List (Certificate)))，来验证权证发行部件(Ticket Issuer)是否未作废。

进而，执行所接收权证---服务许可权证(SPT)(参照图 28、图 31)上记录的签名、即 Integrity Check Value(权证(Ticket)的完整性验证值(公开密钥体制：签名(Signature)))的验证，确认权证是否未被窜改。签名验证与先前的公开密钥证书的签名验证同样，例如根据与图 13 的流程同样的序列来执行。

以上，(1)权证发行者(Ticket Issuer)的公开密钥证书(CERT)是未窜改过的合法的公开密钥证书(CERT)；(2)权证发行者(Ticket Issuer)的公开密钥证书(CERT)的选项区域中记录的代码、和设备内的 FDB (File Definition Block)中记录的权证发行部件代码(SPTIC：SPT Issuer Code)一致；(3)权证发行部件(Ticket Issuer)未作废；(4)通过验证所接收权证(SPT)的签名(Signature)而确认权证没有窜改。以以上全部被确认为条件而认为服务许可权证(SPT)的完整性验证成功。在上述(1)～(4)中某一个未被确认的情况下，判定为不能得到服务许可权证(SPT)的完整性的确认，中止利用服务许可权证(SPT)的处理。

此外，在步骤 S520 中，在判定为权证上记述的 Integrity Check Type(权证(Ticket)的完整性验证值的类型(公开密钥体制(Public)/对称密钥体制(Common)))的设定是对称密钥体制(Common)的情况下，进至步骤 S522，执行 MAC (Message Authentication Code)验证。设备使用设备的文件定义块(参照图 24)中保存的服务许可权证(SPT)的 MAC 验证密钥“Kspt”来执行权证的 MAC 验证处理。MAC 验证处理根据先前说明过的使用图 59 的 DES 加密处理结构的 MAC 值生成处理来执行。

对已保证没有窜改的例如数据发送端在生成数据时生成的 ICV (Integrity Check Value)、和数据接收端根据接收数据而生成的 ICV 进行比较，如果得到同一 ICV，则保证数据没有窜改，而如果 ICV 不同，

则判定为有窜改。已保证没有窜改的例如数据发送端在生成数据时生成的 ICV 如与图 28、图 31 的服务许可权证(SPT)的格式有关的记述中说明过的那样，被保存在 SPT 的 ICV (Integrity Check Value)字段中。对设备生成的 ICV 和所接收权证(SPT)中保存的 ICV 进行比较，如果一致，则判定为权证有完整性；而在不一致的情况下，判定为权证有窜改，中止利用服务许可权证(SPT)的处理。

通过上述处理来完成服务许可权证(SPT)上记述的 Integrity Check Type 是对称密钥体制的情况下的服务许可权证(SPT)验证处理。

在步骤 S515 中，在判定为权证类型不是服务许可权证(SPT: Service Permission Ticket)的情况下，在步骤 S523 中，验证权证类型，判定权证是否是数据更新权证-DEV (DUT: Data Update Ticket (DEV))(参照图 32)。如前所述，数据更新权证(DUT: Data Update Ticket)是执行设备的存储部中保存的各种数据的更新处理时的存取控制权证，有更新设备管理器的管理数据的处理中应用的数据更新权证-DEV (DUT(DEV))和更新分区管理器的管理数据的处理中应用的数据更新权证-PAR (DUT(PAR))。

在权证类型是数据更新权证-DEV (DUT(DEV))的情况下，执行步骤 S524～S528；而在权证类型不是数据更新权证(DEV) (DUT: Data Update Ticket (DEV))的情况下，进至步骤 S529。

在权证类型是数据更新权证-DEV (DUT(DEV))的情况下，在步骤 S524 中，判定权证上记述的 Integrity Check Type(权证(Ticket)的完整性验证值的类型(公开密钥体制(Public)/对称密钥体制(Common)))的设定是否是公开密钥体制(Public)。

在完整性验证值的类型(Integrity Check Type)是公开密钥体制(Public)的情况下，进至步骤 S525，执行各种处理。步骤 S525 中执行的处理首先是使用设备管理器认证机构(CA(DEV))的公开密钥 PUBCA(DEV)的权证发行者(Ticket Issuer)公开密钥证书(CERT)验证处理。

在将数据更新权证-DEV (DUT(DEV))发行部件(DUT Issuer)发行的权证(Ticket)向权证用户发送时，在公开密钥体制的情况下，数据更新权证(DUT)发行部件(DUT Issuer)的公开密钥证书(CERT_DUTI)

也一起被发送到设备。其中，DUT发行部件的公开密钥证书(CERT_DUTI)的属性(Attribute)与设备内的 DKDB(PUB) (Device Key Definition Block) (PUB) 中记录的权证发行部件代码(DUTIC_DEV)的标识符(DUTIC)一致。

在公开密钥证书(参照图 11)上附加有用设备管理器认证机构(CA(DEV))的私有密钥执行的签名，用设备管理器认证机构(CA(DEV))的公开密钥 PUB CA(DEV)来验证该签名。签名生成、验证例如作为根据先前说明过的图 12、图 13 的流程执行的处理来执行。通过该签名验证，来判定权证发行者的公开密钥证书(CERT)是否是未窜改过的合法的公开密钥证书(CERT)。

进而，在步骤 S525 中，判定通过签名验证确认了完整性的权证发行部件的公开密钥证书(CERT)的选项区域中记录的用户所属组代码、和设备内的 DKDB (Device Key Definition Block) (PUB)中记录的权证发行部件代码(DUTIC_DEV: DUT Issuer Category for Device)是否一致。

如图 11 的公开密钥证书的说明一栏中记述的那样，在公开密钥证书中，记录有作为各权证(PRT、FRT、SPT、DUT)的发行部件的权证发行部件(Ticket Issuer)的所属组代码，在此情况下，为 DUTIC (DUT Issuer Code)。通过确认该选项区域的代码和设备内的 DKDB (Device Key Definition Block) (PUB) 中记录的权证发行部件代码(DUTIC_DEV: DUT Issuer Category for Device) (参照图 16)一致，来确认所接收权证(DUT)是合法的权证发行部件发行的权证。

进而，设备参照设备的存储部内的设备密钥区域(参照图 18)中保存的 CRL_DEV(注册有作废设备(Device)、作废机器(作为设备存取机器的读写器、PC 等权证用户、权证发行部件)的公开密钥证书标识符(例如，序列号: SN)的作废表(Revocation List (Certificate)))，来验证权证发行部件(Ticket Issuer)是否未作废。

进而，执行所接收权证数据更新权证-DEV(DUT(DEV)) (参照图 32)上记录的签名、即 Integrity Check Value(权证(Ticket)的完整性验证值(公开密钥体制: 签名(Signature)))的验证，确认权证是否未被窜改。签名验证与先前的公开密钥证书的签名验证同样，例如根据与图 13 的流程同样的序列来执行。

以上，(1)权证发行者(Ticket Issuer)的公开密钥证书(CERT)是未窜改过的合法的公开密钥证书(CERT)；(2)权证发行者(Ticket Issuer)的公开密钥证书(CERT)的选项区域中记录的代码、和设备内的DKDB(PUB) (Device Key Definition Block)(PUB)中记录的权证发行部件代码(DUTIC_DEV: DUT Issuer Category for Device)一致；(3)权证发行部件(Ticket Issuer)未作废；(4)通过验证所接收权证(DUT)的签名(Signature)而确认权证没有窜改。以以上全部被确认为条件而认为数据更新权证-DEV(DUT(DEV))的完整性验证成功。在上述(1)～(4)中某一个未被确认的情况下，判定为不能得到数据更新权证-DEV(DUT(DEV))的完整性的确认，中止利用数据更新权证-DEV(DUT(DEV))的处理。

此外，在步骤S524中，在判定为权证上记述的Integrity Check Type(权证(Ticket)的完整性验证值的类型(公开密钥体制(Public)/对称密钥体制(Common)))的设定是对称密钥体制(Common)的情况下，在步骤S526中，判定数据更新权证-DEV(DUT(DEV))上记述的Old Data Code所示的数据是否是设备密钥区域(参照图18)中保存的Kdut_DEV1(数据更新权证(DUT)的MAC验证密钥)或Kdut_DEV2(数据更新加密密钥)。

在数据更新权证-DEV(DUT(DEV))上记述的Old Data Code(被更新的旧数据的代码)所示的数据是设备密钥区域(参照图18)中保存的Kdut_DEV1(数据更新权证(DUT)的MAC验证密钥)或Kdut_DEV2(数据更新加密密钥)的情况下，在步骤S528中，用设备密钥区域(参照图18)中保存的Kdut_DEV3(数据更新权证(DUT)的MAC验证密钥)来执行MAC验证处理；而在数据更新权证-DEV(DUT(DEV))上记述的Old Data Code(被更新的旧数据的代码)所示的数据不是设备密钥区域(参照图18)中保存的Kdut_DEV1(数据更新权证(DUT)的MAC验证密钥)或Kdut_DEV2(数据更新加密密钥)的情况下，在步骤S527中，用设备密钥区域(参照图18)中保存的Kdut_DEV1(数据更新权证(DUT)的MAC验证密钥)来执行MAC验证处理。

如上所述执行MAC验证密钥的区分使用是因为，在待更新的数据是Kdut_DEV1(数据更新权证(DUT)的MAC验证密钥)或Kdut_DEV2(数据更新加密密钥)的情况下，这些密钥数据是由于某些理由、例如密钥

信息的泄漏等而预定要停止使用的信息，所以避免用这些待更新数据来进行MAC验证。MAC验证处理根据先前说明过的使用图59的DES加密处理结构的MAC值生成处理来执行。

其中，在设备的设备密钥区域(参照图18)中新保存Kdut_DEV1(数据更新权证(DUT)的MAC验证密钥)的情况下，与以前已保存的Kdut_DEV3(数据更新权证(DUT)的MAC验证密钥)进行交换处理。再者，在新保存Kdut_DEV2(数据更新加密密钥)的情况下，也与以前已保存的Kdut_DEV4(数据更新加密密钥)进行交换处理。

通过该交换Kdut_DEV1和Kdut_DEV3、及交换Kdut_DEV2和Kdut_DEV4，来始终维持Kdut_DEV3(数据更新权证(DUT)的MAC验证密钥)、Kdut_DEV4(数据更新加密密钥)这一对儿比Kdut_DEV1(数据更新权证(DUT)的MAC验证密钥)、Kdut_DEV2(数据更新加密密钥)这一对儿的版本新。即，密钥Kdut_DEV1和Kdut_DEV2是经常使用的密钥，Kdut_DEV3和Kdut_DEV4具有下述作为备份密钥的作用：在非常时更新Kdut_DEV1和Kdut_DEV2，并且被置换为当前正在使用的密钥Kdut_DEV1和Kdut_DEV2。其中，在后面的使用数据更新权证(DUT)的数据更新处理的说明中还将说明这些处理。

对已保证没有窜改的例如数据发送端在生成数据时生成的ICV(Integrity Check Value)、和数据接收端根据接收数据而生成的ICV进行比较，如果得到同一ICV，则保证数据没有窜改，而如果ICV不同，则判定为有窜改。已保证没有窜改的例如数据发送端在生成数据时生成的ICV如与图32的数据更新权证(DUT)的格式有关的记述中说明过的那样，被保存在数据更新权证(DUT)的ICV (Integrity Check Value)字段中。

对设备生成的ICV和所接收权证---数据更新权证-DEV(DUT(DEV))中保存的ICV进行比较，如果一致，则判定为权证有完整性；而在不一致的情况下，判定为权证有窜改，中止利用数据更新权证-DEV(DUT(DEV))的处理。

通过上述处理来完成数据更新权证-DEV(DUT(DEV))上记述的Integrity Check Type是对称密钥体制的情况下的数据更新权证-DEV(DUT(DEV))验证处理。

在步骤S523中，在判定为权证类型不是数据更新权证-

DEV(DUT(DEV))的情况下，判定为权证是数据更新权证-PAR(DUT(PAR))(参照图 32)。数据更新权证-PAR(DUT(PAR))是更新分区管理器的管理数据的处理中应用的权证。

在此情况下，在步骤 S529 中，判定权证上记述的 Integrity Check Type(权证(Ticket)的完整性验证值的类型(公开密钥体制(Public)/对称密钥体制(Common)))的设定是否是公开密钥体制(Public)。

在完整性验证值的类型(Integrity Check Type)是公开密钥体制(Public)的情况下，进至步骤 S530，执行各种处理。步骤 S530 中执行的处理首先是使用分区管理器认证机构(CA(PAR))的公开密钥 PUB CA(PAR)的权证发行者(Ticket Issuer)公开密钥证书(CERT)验证处理。

在将数据更新权证-PAR (DUT(PAR))发行部件(DUT Issuer)发行的权证(Ticket)向权证用户发送时，在公开密钥体制的情况下，数据更新权证(DUT)发行部件(DUT Issuer)的公开密钥证书(CERT_DUTI)也一起被发送到设备。其中，DUT发行部件的公开密钥证书(CERT_DUTI)的属性(Attribute)与设备内的 PKDB(PUB)(Partition Key Definition Block)中记录的权证发行部件代码(DUTIC_PAR)一致。

在公开密钥证书(参照图 11)上附加有用分区管理器认证机构(CA(PAR))的私有密钥执行的签名，用分区管理器认证机构(CA(PAR))的公开密钥 PUB CA(PAR)来验证该签名。签名生成、验证例如作为根据先前说明过的图 12、图 13 的流程执行的处理来执行。通过该签名验证，来判定权证发行者的公开密钥证书(CERT)是否是未窜改过的合法的公开密钥证书(CERT)。

进而，在步骤 S530 中，判定通过签名验证确认了完整性的权证发行部件的公开密钥证书(CERT)的选项区域中记录的用户所属组代码、和设备内的 PKDB (Partition Key Definition Block)中记录的权证发行部件代码(DUTIC_PAR: DUT Issuer Category for Partition)是否一致。

如图 11 的公开密钥证书的说明一栏中记述的那样，在公开密钥证书中，记录有作为各权证(PRT、FRT、SPT、DUT)的发行部件的权证发行部件(Ticket Issuer)的所属组代码，在此情况下，为 DUTIC (DUT Issuer Code)。通过确认该选项区域的代码和设备内的

PKDB(PUB)(Partition Key Definition Block)中记录的权证发行部件代码(DUTIC: DUT Issuer Category)(参照图 21)一致，来确认所接收权证(DUT)是合法的权证发行部件发行的权证。

进而，设备参照设备的存储部内的设备密钥区域(参照图 18)中保存的 CRL_DEV(注册有作废设备(Device)、作废机器(作为设备存取机器的读写器、PC 等权证用户、权证发行部件)的公开密钥证书标识符(例如，序列号: SN)的作废表(Revocation List (Certificate)))，来验证权证发行部件(Ticket Issuer)是否未作废。

进而，执行所接收权证数据更新权证-PAR(DUT(PAR))(参照图 32)上记录的签名、即 Integrity Check Value(权证(Ticket)的完整性验证值(公开密钥体制: 签名(Signature)))的验证，确认权证是否未被窜改。签名验证与先前的公开密钥证书的签名验证同样，例如根据与图 13 的流程同样的序列来执行。

以上，(1)权证发行者(Ticket Issuer)的公开密钥证书(CERT)是未窜改过的合法的公开密钥证书(CERT)；(2)权证发行者(Ticket Issuer)的公开密钥证书(CERT)的选项区域中记录的代码、和设备内的 PKDB(PUB) (Partition Key Definition Block)中记录的权证发行部件代码(DUTIC_PAR: DUT Issuer Category for Partition)一致；(3)权证发行部件(Ticket Issuer)未作废；(4)通过验证所接收权证(DUT)的签名(Signature)而确认权证没有窜改。以以上全部被确认为条件而认为数据更新权证-PAR(DUT)的完整性验证成功。在上述(1)~(4)中某一个未被确认的情况下，判定为不能得到数据更新权证-PAR(DUT(PAR))的完整性的确认，中止利用数据更新权证-PAR(DUT(PAR))的处理。

此外，在步骤 S529 中，在判定为权证上记述的 Integrity Check Type(权证(Ticket)的完整性验证值的类型(公开密钥体制(Public)/对称密钥体制(Common)))的设定是对称密钥体制(Common)的情况下，在步骤 S531 中，判定数据更新权证-PAR(DUT(PAR))上记述的 Old Data Code(被更新的旧数据的代码)所示的数据是否是分区密钥区域(参照图 23)中保存的 Kdut_PAR1(数据更新权证(DUT)的 MAC 验证密钥)或 Kdut_PAR2(数据更新加密密钥)。

在数据更新权证-PAR(DUT(PAR))上记述的 Old Data Code(被更

新的旧数据的代码)所示的数据是分区密钥区域(参照图 23)中保存的 Kdut_PAR1(数据更新权证(DUT)的 MAC 验证密钥)或 Kdut_PAR2(数据更新加密密钥)的情况下,在步骤 S533 中,用分区密钥区域(参照图 23)中保存的 Kdut_PAR3(数据更新权证(DUT)的 MAC 验证密钥)来执行 MAC 验证处理;而在数据更新权证-PAR(DUT(PAR))上记述的 Old Data Code(被更新的旧数据的代码)所示的数据不是分区密钥区域(参照图 23)中保存的 Kdut_PAR1(数据更新权证(DUT)的 MAC 验证密钥)或 Kdut_PAR2(数据更新加密密钥)的情况下,在步骤 S532 中,用分区密钥区域(参照图 23)中保存的 Kdut_PAR1(数据更新权证(DUT)的 MAC 验证密钥)来执行 MAC 验证处理。

如上所述执行 MAC 验证密钥的区分使用是因为,在待更新的数据是 Kdut_PAR1(数据更新权证(DUT)的 MAC 验证密钥)或 Kdut_PAR2(数据更新加密密钥)的情况下,这些密钥数据是由于某些理由、例如密钥信息的泄漏等而预定要停止使用的信息,所以避免用这些待更新数据来进行 MAC 验证。MAC 验证处理根据先前说明过的使用图 59 的 DES 加密处理结构的 MAC 值生成处理来执行。

对已保证没有窜改的例如数据发送端在生成数据时生成的 ICV (Integrity Check Value)、和数据接收端根据接收数据而生成的 ICV 进行比较,如果得到同一 ICV,则保证数据没有窜改,而如果 ICV 不同,则判定为有窜改。已保证没有窜改的例如数据发送端在生成数据时生成的 ICV 如与图 32 的数据更新权证(DUT)的格式有关的记述中说明过的那样,被保存在数据更新权证(DUT)的 ICV (Integrity Check Value)字段中。

对设备生成的 ICV 和所接收权证---数据更新权证-PAR(DUT(PAR))中保存的 ICV 进行比较,如果一致,则判定为权证有完整性;而在不一致的情况下,判定为权证有窜改,中止利用数据更新权证-PAR(DUT(PAR))的处理。

通过上述处理来完成数据更新权证-PAR(DUT(PAR))上记述的 Integrity Check Type 是对称密钥体制的情况下的数据更新权证-PAR(DUT(PAR))验证处理。

在以上处理中确认了权证完整性后,进至图 58 的步骤 S541,以下,执行用户检查、即作为权证用户与设备正在执行通信的作为设备

存取机器的读写器(或PC等)的检查。

在步骤S541中，设备检查所接收权证(PRT、FRT、SPT、或DUT)的Authentication Flag(表示在权证(Ticket)利用处理中是否需要与设备(Device)进行相互鉴别的标志)。在标志表示无需鉴别的情况下，不执行处理就结束。

在步骤S541的标志检查处理中，在标志表示需要鉴别的情况下，进至步骤S542，以权证用户(想要应用权证对设备执行处理的作为设备存取机器的读写器、PC等)所属组为关键字来查阅鉴别表(参照图51)。

接着，在步骤S543中，检查所接收权证的Authentication Type(设备(Device)的相互鉴别类型(公开密钥鉴别、对称密钥鉴别、或任一种皆可(Any)))，在任一种皆可(Any)的情况下，进至步骤S544，判定步骤S542中检查过的组的相互鉴别数据是否被保存在鉴别表(参照图51)中。如果判定为在表中保存有对应组的相互鉴别信息，权证用户(想要应用权证对设备执行处理的作为设备存取机器的读写器、PC等)和设备间的相互鉴别已完毕，则认为权证用户(例如，作为设备存取机器的读写器)的完整性已被确认，判定为用户检查成功而结束处理。在鉴别表(参照图51)中未保存对应组的相互鉴别信息的情况下，判定为用户检查未完，出错结束。

在步骤S543中，在所接收权证的Authentication Type(记录有设备(Device)的相互鉴别类型(公开密钥鉴别、对称密钥鉴别、或任一种皆可(Any))的数据)不是任一种皆可(Any)的情况下，在步骤S545中，判定Authentication Type是否是公开密钥鉴别。

在Authentication Type是公开密钥鉴别的情况下，进至步骤S546，判定步骤S542中检查过的组的公开密钥相互鉴别数据是否被保存在鉴别表(参照图51)中。在判定为在表中保存有对应组的公开密钥相互鉴别信息、权证用户(想要应用权证对设备执行处理的作为设备存取机器的读写器、PC等)和设备间的相互鉴别作为公开密钥鉴别处理已成立的情况下，进至步骤S547，判定在待处理权证(PRT、FRT、SPT或DUT)中是否存在权证用户的标识符，在存在的情况下，在步骤S548中判定对方鉴别者(权证用户---作为设备存取机器的读写器等)的公开密钥证书中的作为标识数据(DN)而记录的标识符、范畴或序列号(SN)

和权证中保存的作为权证用户的标识数据而记录的标识符、范畴或序列号(SN)是否一致。在一致的情况下，认为用户确认成功而结束处理。

在步骤S546中，在判定为步骤S542中检查过的组的公开密钥相互鉴别数据未被保存在鉴别表(参照图51)中、权证用户(想要应用权证对设备执行处理的作为设备存取机器的读写器、PC等)和设备间的相互鉴别未作为公开密钥鉴别处理而成立的情况下，判定为用户检查未完，出错结束。

此外，在步骤S548中判定为对方鉴别者(权证用户---作为设备存取机器的读写器等)的公开密钥证书中的作为标识数据(DN)而记录的标识符、范畴或序列号(SN)和权证中保存的权证用户的标识符不一致的情况下，也判定为用户检查未完，出错结束。

其中，在权证中不存在权证用户的标识符的情况下，不执行步骤S548的处理，认为用户确认成功而结束处理。

在步骤S545中，在判定为所接收权证的Authentication Type(记录有设备(Device)的相互鉴别类型(公开密钥鉴别、对称密钥鉴别、或任一种皆可(Any))的数据)不是公开密钥鉴别的情况下，进至步骤S549，判定步骤S542中检查过的组的对称密钥相互鉴别数据是否被保存在鉴别表(参照图51)中。在判定为在表中保存有对应组的对称密钥相互鉴别信息、权证用户(想要应用权证对设备执行处理的作为设备存取机器的读写器、PC等)和设备间的相互鉴别作为对称密钥鉴别处理已成立的情况下，进至步骤S550，判定在待处理权证(PRT、FRT、SPT或DUT)中是否存在权证用户的标识符，在存在的情况下，在步骤S551中判定对方鉴别者(权证用户---作为设备存取机器的读写器等)的标识数据(IDrw)和权证中保存的权证用户的标识符是否一致。在一致的情况下，认为用户确认成功而结束处理。

在步骤S549中，在判定为步骤S542中检查过的组的对称密钥相互鉴别数据未被保存在鉴别表(参照图51)中、权证用户(想要应用权证对设备执行处理的作为设备存取机器的读写器、PC等)和设备间的相互鉴别未作为对称密钥鉴别处理而成立的情况下，判定为用户检查未完，出错结束。

此外，在步骤S551中判定为对方鉴别者(权证用户---作为设备存取机器的读写器等)的标识数据(IDrw)和权证中保存的权证用户的标

识符不一致的情况下，也判定为用户检查未完，出错结束。

其中，在权证中不存在权证用户的标识符、或所有权证用户可利用的情况下，不执行步骤S550的处理，认为用户确认成功而结束处理。

以上是图47的流程中的步骤S413中设备执行的权证完整性及用户检查处理。

(分区创建删除处理)

接着，用图60、图61的处理流程来详细说明图47的流程所示的步骤S415中执行的基于分区注册权证(PRT)的分区创建、删除处理。分区创建、删除处理是从权证用户(例如，作为设备存取机器的读写器、PC等)接收到分区注册权证(PRT)的设备根据分区注册权证(PRT)执行的处理。

在图60的步骤S601中，设备验证接收到的分区注册权证(PRT: Partition Registration ticket)上记录的处理类型、即Operation Type(分区(Partition)创建还是删除的指定(创建(Generate)/删除(Delete)))。在处理类型(Operation Type)是分区(Partition)创建的情况下，执行步骤S602以下的处理；而在处理类型(Operation Type)是分区(Partition)删除的情况下，执行步骤S621以下的处理。

首先，说明分区创建处理。设备在步骤S602中验证在设备的存储部中是否存在代码与分区注册权证(PRT)上记述的分区管理器代码(PMC)相同的分区。该判定可如下来进行：验证在设备的存储部的分区定义块(参照图19)中是否记述有与所接收权证(PRT)的记述代码相同的代码。

在设备中已经存在同一代码(PMC)的分区的情况下，不许存在具有同一代码的重复分区，不执行分区的创建，出错结束。而在设备中不存在同一代码的分区的情况下，在步骤S603中，对设备管理信息块(参照图15)的设备(Device)内的空闲块数(Free Block Number in Device)、和分区注册权证(PRT)上记述的分区长度(Partition Size)进行比较，判定在设备的存储部中是否存在权证(PRT)上记述的分区长度(Partition Size)以上的空闲块区域。在不存在的情况下，不能创建PRT上记述的长度的分区，所以出错结束。

在判定为在设备的存储部中存在权证(PRT)上记述的分区长度(Partition Size)以上的空闲块区域的情况下，进至步骤S604，参照

设备管理信息块的空闲区域指针(Pointer of Free Area)，在设备的空闲区域(Free Area in Device)的最高块中保留分区定义块(PDB)区域(参照图 19)。

接着，设备向保留的分区定义块(PDB)区域中拷贝分区注册权证(PRT)上记述的分区管理器代码(PMC)(S605)、PRT 上记述的 PMC 版本(S606)。

进而，向分区定义块(PDB)区域的分区起始位置(Partition Start Position)上，拷贝设备管理信息块(参照图 15)的空闲区域指针(Pointer of Free Area)(S607)，进而向分区定义块(PDB)区域的分区长度(Partition Size)中拷贝分区注册权证(PRT)上记述的分区长度(Partition Size)(S608)。

接着，将设备管理信息块(参照图 15)的空闲区域指针(Pointer of Free Area)加上分区定义块(PDB)区域的分区长度(Partition Size)中拷贝的值(S609)，将设备管理信息块(参照图 15)的设备(Device)内的空闲块数(Free Block Number in Device)减去分区长度(Partition Size)+1(S610)。其中，+1 表示分区定义块(PDB)用的块。

接着，将设备管理信息块(参照图 15)的分区数(Partition Number)加上 1，即加上创建的分区数(1)(S611)。

接着，在图 61 的步骤 S631 中，将创建的分区区域的最高块设定为分区管理信息块(PMIB: Partition Management Information Block)(参照图 20)，向设定的分区管理信息块(PMIB)的分区管理器代码(PMC)字段中拷贝分区注册权证(PRT)的 PMC(S632)，向分区管理信息块(PMIB)的 PMC 版本字段中拷贝分区注册权证(PRT)的 PMC 版本(S633)，向分区管理信息块(PMIB)的分区总块数(Total Block Number in Partition)字段中拷贝分区注册权证(PRT)的分区长度(Partition Size)(S634)。

进而，向分区管理信息块(PMIB)的分区空闲块数(Free Block Number in Partition)字段中记录分区注册权证(PRT)的分区长度(Partition Size)-3(S635)。其中，-3 表示减去已经预定使用的分区管理信息块(PMIB)、对称密钥类分区密钥定义块(PKDB(common))、公开密钥类分区密钥定义块(PKDB(PUB))这 3 个块。

进而，向分区管理信息块(PMIB)的文件数(File Number)中填写

0(S636)。此时在分区内未设定文件。文件可使用文件注册权证(FRT)来设定。使用该文件注册权证(FRT)的文件注册处理待后述。

进而，向分区管理信息块(PMIB)的空闲区域指针(Pointer of Free Area)中拷贝分区定义块(PDB)的起始位置(Start Position)，结束分区设定注册。

接着说明图 60 的步骤 S621～S628 的分区删除处理。在步骤 S621 中，验证在设备的存储部中是否存在代码与分区注册权证(PRT)上记述的分区管理器代码(PMC)相同的分区。该判定可如下来进行：验证在设备的存储部的分区定义块(参照图 19)中是否记述有与所接收权证(PRT)的记述代码相同的代码。

在设备中不存在同一代码(PMC)的分区的情况下，不能删除分区，所以出错结束。而在设备中存在同一代码的分区的情况下，在步骤 S622 中，判定在设备中是否存在待删除分区以后创建的分区。在不存在的情况下，待删除分区是最新的分区，在步骤 S629 中删除待删除分区的分区定义块(PDB)(参照图 19)。

在步骤 S622 中，在判定为在设备中存在待删除分区以后创建的分区的情况下，将后创建的分区(后分区)的数据向低处移动相当于待删除分区的长度(PS)的距离(S623)，进而将后分区的分区定义块(PDB)向高处移动 1 个块(S624)。此外，将后分区的分区定义块(PDB)上记录的分区起始位置(Partition Start Position)减去删除分区的长度(PS)(S625)。

在步骤 S625 或 S629 的处理后，在步骤 S626 中，将设备管理信息块(DMIB)(参照图 15)的设备(Device)内的空闲块数(Free Block Number in Device)加上删除分区的长度(PS)+1。+1 表示删除分区的分区定义块(DPB)用的块。

接着在步骤 S627 中，将设备管理信息块(参照图 15)的空闲区域指针(Pointer of Free Area)的值减去删除分区的长度(PS)。进而，在步骤 S628 中，将设备管理信息块(参照图 15)的分区数(Partition Number)减去 1，即减去已删除的分区数(1)，结束基于分区注册权证(PRT)的分区删除处理。

以上是图 47 的处理流程中的步骤 S415 的基于分区注册权证(PRT)的分区创建、删除处理。

(分区初始注册)

接着，用图 62 以下的流程来详细说明图 47 的处理流程中的步骤 S406、S419 的分区初始数据写入处理、即基于分区注册权证(PRT)的分区初始注册处理。

在图 62、图 63、图 64 所示的处理流程中，左侧示出分区管理器管辖的初始注册装置的处理，右侧示出设备(参照图 5)的处理。其中，分区管理器管辖的初始注册装置是可对设备进行读取写入处理的装置(例如，作为设备存取机器的读写器、PC)，具有与图 10 的作为设备存取机器的读写器相当的结构。如图 47 的处理流程所示，在图 62 的处理开始以前，在初始注册装置和设备间，相互鉴别成立，在权证完整性、用户检查中确认了权证及用户(权证用户---作为设备存取机器的读写器等)的完整性，并且基于分区注册权证(PRT)的分区创建处理已结束。此外，图 62、图 63、图 64 的初始注册装置和设备间的数据发送接收是发送接收用相互鉴别时生成的会话密钥 Kses 加密过的数据。

在图 62 的步骤 S641 中，初始注册装置判定在分区鉴别中是否使用对称密钥。该判定参照使用的分区注册权证(PRT)(参照图 26)的 Authentication Type(设备(Device)的相互鉴别类型(公开密钥鉴别、对称密钥鉴别、或任一种皆可(Any)))字段来进行。

如图 62 所示，在分区鉴别中使用对称密钥的情况下，执行步骤 S642～S643、S651～S654；而在分区鉴别中不使用对称密钥的情况下，省略这些步骤。

在分区鉴别中使用对称密钥的情况下，在步骤 S642 中，作为对称密钥鉴别数据写入命令，初始注册装置将 MKauth_PAR_A“双向个别密钥鉴别主密钥”、Kauth_PAR_B“双向个别密钥鉴别对称密钥”、IRL_PAR“注册有作废设备(Device)的设备标识符(ID)的作废表(Revocation List(Device ID))”、及其版本信息发送到设备。

在步骤 S651 中，设备接收上述写入命令，在步骤 S652 中，将所接收数据写入到分区密钥区域(参照图 23)中。接着，执行通过数据写入而产生的指针、长度、设备内的空闲块数的调整(S653)，将写入结束通知发送到注册装置(S654)。

接收到写入结束通知(S643)的注册装置在步骤 S644 中判定在分区鉴别中是否使用公开密钥。如图 62 所示，在分区鉴别中使用公开密

钥的情况下，执行步骤S645～S649、S656～S664；而在分区鉴别中不使用公开密钥的情况下，省略这些步骤。

在分区鉴别中使用公开密钥的情况下，在步骤S645中，作为公开密钥鉴别数据写入命令，注册装置将PUB_CA(PAR)“发行分区管理器公开密钥证书的认证机构CA(PAR)的公开密钥”、PARAM_PAR“分区(Partition)的公开密钥参数”、CRL_PAR“注册有作废设备(Device)的公开密钥证书标识符(例如，序列号：SN)的作废表(Revocation List(Certificate))、及它们的版本信息发送到设备。

在步骤S655中，设备接收上述写入命令，在步骤S656中，将所接收数据写入到分区密钥区域(参照图23)中。接着，执行通过数据写入而产生的指针、长度、设备内的空闲块数的调整(S657)，将写入结束通知发送到注册装置(S658)。

接收到写入结束通知(S646)的注册装置将公开密钥和私有密钥的密钥对生成命令发送到设备(S647)。其中，在本实施例中，密钥对的生成由设备执行，但是例如也可以由注册装置执行并提供给设备。

接收到密钥对生成命令(S659)的设备在设备内的加密处理部(参照图5)中生成公开密钥(PUB PAR)和私有密钥(PRI PAR)的密钥对，将生成的密钥写入到分区密钥区域(参照图23)中(S660)。接着执行通过数据写入而产生的指针、长度、设备内的空闲块数的调整(S661)，将生成保存的公开密钥发送到注册装置(S662)。

注册装置从设备接收公开密钥(PUB PAR)(S648)，与先前从设备接收到的设备的标识符IDm一起，保存到分区管理器内的数据库(DB(PAR))(参照图9)中。

接着，分区管理器的注册装置判定在文件注册权证(FRT：File Registration Ticket)验证处理中是否使用对称密钥(S671)。对于权证鉴别，如前所述，可应用基于MAC值验证等的对称密钥体制、和进行基于私有密钥的签名生成、基于公开密钥的签名验证的公开密钥体制中的某一种，分区管理器可以设定设备采用的验证处理体制。分区管理器按照设备采用的FRT权证验证方式将可执行对称密钥、公开密钥中的某一种、或两种体制的数据设定到设备中。

在设定为文件注册权证(FRT：File Registration Ticket)验证处理中执行对称密钥验证的情况下，分区管理器将对称密钥体制的FRT

验证所需的信息(例如，FRT验证对称密钥)设置到设备中；而如果设备是不执行对称密钥鉴别的设备，则不将这些信息保存到设备中。

如图63所示，在FRT验证中使用对称密钥体制的情况下，执行步骤S672～S673、S681～S684；而在FRT验证中不使用对称密钥的情况下，省略这些步骤。

在FRT验证中使用对称密钥的情况下，在步骤S672中，作为FRT验证对称密钥写入命令，注册装置将Kfrt“文件注册权证(FRT)的MAC验证密钥”、及版本信息发送到设备。

在步骤S681中，设备接收上述写入命令，在步骤S682中，将所接收数据写入到分区密钥区域(参照图23)中。接着，执行通过数据写入而产生的指针、长度、设备内的空闲块数的调整(S683)，将写入结束通知发送到注册装置(S684)。

接收到写入结束通知(S673)的注册装置在步骤S674中判定在FRT验证中是否使用公开密钥。如图63所示，在FRT验证中使用公开密钥的情况下，执行步骤S675～S676、S685～S690；而在FRT验证中不使用公开密钥的情况下，省略这些步骤。

在FRT验证中使用公开密钥的情况下，在步骤S675中，作为FRT验证数据写入命令，注册装置将FRTIC (FRT Issuer Category)“文件注册权证(FRT)发行者范畴”、PUB_CA(PAR)“发行分区管理器公开密钥证书的认证机构 CA(PAR)的公开密钥”、PARAM_PAR“分区(Partition)的公开密钥参数”、CRL_PAR“注册有作废设备(Device)的公开密钥证书标识符(例如，序列号：SN)的作废表(Revocation List(Certificate))”、及其版本信息发送到设备。

在步骤S685中，设备接收上述写入命令，在步骤S686中，将所接收数据中的 FRTIC (FRT Issuer Category)“文件注册权证(FRT)发行者范畴”写入到公开密钥类分区密钥定义块(PKDB: Partition Key Definition Block (PUB)(参照图22))中，将版本信息写入到同一块的版本区域中。

接着，设备在步骤S687中判定PUB_CA(PAR)“发行分区管理器公开密钥证书的认证机构 CA(PAR)的公开密钥数据”是否已写入，在未写入的情况下，在步骤S688中，将PUB_CA(PAR)、PARAM_PAR、CRL_PAR写入到分区密钥区域(参照图23)中。接着，执行通过数据写入而产生

的指针、长度、设备内的空闲块数的调整(S689)，将写入结束通知发送到注册装置(S690)。

接收到写入结束通知(S676)的注册装置接着在步骤 S701 中判定设备是否支持对称密钥数据的更新。设备中保存的某些数据可作为待更新数据用前述数据更新权证(DUT: Data Update Ticket)(参照图 32)来更新。待更新数据如先前用图 33 所述。在使用该数据更新权证(DUT: Data Update Ticket)的更新处理中，也可采用对称密钥体制、或公开密钥体制中的某一种体制，分区管理器按照设定的分区将可执行某一种体制或两种体制的数据设定到设备中。

在设定的分区执行基于对称密钥体制的数据更新的情况下，分区管理器将对称密钥体制的数据更新处理所需的信息(例如，数据更新权证(DUT)的 MAC 验证密钥等)设置到设备中；而如果设备是不执行对称密钥鉴别的设备，则不将这些信息保存到设备的分区密钥区域中。

如图 64 所示，在使用数据更新权证(DUT: Data Update Ticket)的数据更新处理中使用对称密钥体制的情况下，执行步骤 S702～S703、S711～S714；而在数据更新中不使用对称密钥体制的情况下，省略这些步骤。

在数据更新中使用对称密钥的情况下，在步骤 S702 中，作为数据更新权证(DUT: Data Update Ticket)验证对称密钥写入命令，注册装置将 Kdut_PAR1“数据更新权证(DUT)的 MAC 验证密钥”、Kdut_PAR2“数据更新加密密钥”、Kdut_PAR3“数据更新权证(DUT)的 MAC 验证密钥”、Kdut_PAR4“数据更新加密密钥”及它们的版本信息发送到设备。

在步骤 S711 中，设备接收上述写入命令，在步骤 S712 中，将所接收数据写入到分区密钥区域(参照图 23)中。接着，执行通过数据写入而产生的指针、长度、设备内的空闲块数的调整(S713)，将写入结束通知发送到注册装置(S714)。

接收到写入结束通知(S703)的注册装置在步骤 S704 中判定设备中设定的分区是否支持使用公开密钥体制的数据更新权证(DUT: Data Update Ticket)的数据更新处理。如图 64 所示，在支持公开密钥体制的情况下，执行步骤 S705～S706、S715～S718；而在不支持公开密钥体制的情况下，省略这些步骤。

在支持公开密钥体制的情况下，在步骤S705中，作为数据更新权证(DUT: Data Update Ticket)发行者代码写入命令，注册装置将DUTIC_PAR (DUT Issuer Category)“数据更新权证(DUT: Data Update Ticket)发行者范畴”、及版本信息发送到设备。

在步骤S715中，设备接收上述写入命令，在步骤S716中，将所接收数据写入到公开密钥类分区密钥定义块(PKDB(PUB): Partition Key Definition Block (PUB))中。接着，执行通过数据写入而产生的指针、长度、设备内的空闲块数的调整(S717)，将写入结束通知发送到注册装置(S718)。注册装置接收写入结束通知(S706)并结束处理。

分区管理器执行的初始注册处理(图62～图64的处理流程)完成的状态下设备的存储器内保存数据结构例示于图65。在上述流程(图62～图64)中，从注册装置发送并将下述数据写入到图65所示的分区(Partition)区域中的分区密钥区域中。

* IRL_PAR: 注册有分区存取作废设备(Device)、作废机器(作为设备存取机器的读写器、PC等权证用户、权证发行部件)的标识符(ID)的作废表(Revocation List (Device ID))

* CRL_PAR: 注册有分区存取作废设备(Device)、作废机器(作为设备存取机器的读写器、PC等权证用户、权证发行部件)的公开密钥证书标识符(例如，序列号: SN)的作废表(Revocation List (Certificate))

* Kauth_PAR_B: 双向个别密钥鉴别对称密钥

* MKauth_PAR_A: 双向个别密钥鉴别主密钥

* Kdut_PAR1: 数据更新权证(DUT)的MAC验证密钥

* Kdut_PAR2: 数据更新加密密钥

* Kdut_PAR3: 数据更新权证(DUT)的MAC验证密钥

* Kdut_PAR4: 数据更新加密密钥

* Kfrt: 文件注册权证(FRT)的MAC验证密钥

此外，在设备中生成并写入

* PUB_PAR: 分区(Partition)的公开密钥

* PRI_PAR: 分区(Partition)的私有密钥。

此外，

* PARAM_PAR：分区(Partition)的公开密钥参数

* PUB_CA(PAR)：认证机构CA(PAR)的公开密钥

对称密钥类分区密钥信息块(Partition Key Definition Block(Common))

公开密钥类分区密钥信息块(Partition Key Definition Block(PUB))

分区管理信息块(Partition Management Information Block)

各数据是在创建分区时(参照处理流程图60、图61)写入的数据。

[B4. 2. 分区管理器管理下的公开密钥证书发行处理]

接着用图66以下来说明分区管理器执行的分区公开密钥证书发行处理。在设备中，可保存整个设备的鉴别、以设备为单位的处理中可应用的设备公开密钥证书(CERT DEV)、以及对设备内的特定的分区进行处理时的鉴别及其他验证处理等可应用的分区公开密钥证书(CERT PAR)。可对设备中设定的每个分区设定保存分区公开密钥证书(CERT PAR)。

分区公开密钥证书(CERT PAR)通过经分区管理器管辖的注册机构将认证机构(CA for PM)(参照图2、图3)发行的公开密钥证书授予设备的过程来发行，对分区管理器管辖的注册机构发行的公开密钥证书(CERT PAR)执行管理(数据库332(参照图9))。

根据图66及图67，来说明分区管理器管辖的注册机构对设定分区执行的分区公开密钥证书(CERT PAR)发行处理的过程。在图66、图67中，左侧是分区管理器管辖的注册机构的CERT(公开密钥证书)发行装置、具体地说是图9所示的分区管理器的结构图中的控制部件331的处理，右侧是设备的处理。

首先在步骤S721中，CERT发行装置取得作为分区公开密钥证书(CERT PAR)的发行对象的设备的用户信息，进行证书发行的许可(判定)，确保与作为发行对象的设备的信道。作为分区公开密钥证书(CERT PAR)的发行对象的设备的用户信息例如可从设备的初始注册时生成的数据中取得。其中，用户信息也可以在设定与设备的信道后，从设备取得。信道不管有线、无线，只要作为可发送接收数据的信道来确保即可。

接着，CERT发行装置在步骤S722中，将包含随机数R的鉴别数

据生成命令发送到设备。接收到鉴别数据生成命令(S731)的设备对接收随机数 R、和设备标识符(IDm)的结合数据应用设备私有密钥(PRIDEV)来执行数字签名(S)生成处理(参照图 12)(S732)。设备将设备的设备数据(IDm)和签名(S)发送到 CERT 发行装置。

从设备接收到设备数据(IDm)和签名(S)(S723)的 CERT 发行装置以接收到的设备标识数据(IDm)为搜索关键字，从数据库 DB(PAR)332 中取得已保存的设备公开密钥(PUB PAR)。进而，应用取得的设备公开密钥(PUB PAR)来执行签名(S)的验证处理(参照图 13)(S725)。在验证未成功的情况下，判定为来自设备的发送数据是非法数据，结束处理。

在验证成功的情况下，请求认证机构(CA for PM)620 进行分区公开密钥证书(CERT PAR)发行处理(S727)。分区管理器接收认证机构 620 发行的分区公开密钥证书(CERT PAR)(S728)，发送到设备(S729)。

从分区管理器(注册机构)接收到分区公开密钥证书(CERT PAR)的设备用预先已保存在分区密钥区域(参照图 23)中的认证机构的公开密钥(PUB CA(PAR))来执行接收到的分区公开密钥证书(CERT PAR)的签名验证。即在公开密钥证书中有用认证机构的私有密钥执行的签名(参照图 11)，进行该签名验证(S736)。

在签名验证失败的情况下，判定为不是合法的公开密钥证书，将出错通知发送到 CERT 发行装置(S745)。

在签名验证成功的情况下，比较分区公开密钥证书(CERT PAR)中保存的设备公开密钥(PUB PAR)和本设备中保管的设备公开密钥(PUB PAR)(S741)，在不一致的情况下执行出错通知；而在一致的情况下，将接收到的分区公开密钥证书(CERT PAR)保存到分区密钥区域(参照图 23)中(S743)。其中，在发行分区公开密钥证书(CERT PAR)以前，在该区域中保存本设备生成的公开密钥(PUB PAR)，在发行合法的分区公开密钥证书(CERT PAR)时，由分区公开密钥证书(CERT PAR)来盖写。

在分区公开密钥证书(CERT PAR)的保存结束后，将保存处理结束通知发送到 CERT 发行装置(S744)。CERT 发行装置接收保存处理结束通知(S751)，确认保存成功(S752)，结束处理。在未得到保存成功的确认的情况下，作为出错而结束处理。

[B4. 3. 分区创建处理各方式中的处理过程]

如上所述，在分区设定注册处理中，在分区管理器管理的作为设备存取机器的读写器和设备间执行相互鉴别，根据分区注册权证(PRT)来设定分区。如上所述，相互鉴别处理的形态是公开密钥相互鉴别、对称密钥相互鉴别这2种中的某一种，并且权证(PRT)验证处理也执行公开密钥类签名验证、对称密钥类MAC验证这2种中的某一种。即处理形态大体分为下述4种形态。

(A)相互鉴别(公开密钥)，权证(PRT)验证(公开密钥)

(B)相互鉴别(公开密钥)，权证(PRT)验证(对称密钥)

(C)相互鉴别(对称密钥)，权证(PRT)验证(对称密钥)

(D)相互鉴别(对称密钥)，权证(PRT)验证(公开密钥)。

以认证机构(CA(DM))、设备管理器(DM)、分区管理器(PM)、设备、各实体间执行的数据传送处理为中心，用附图来简洁说明这4种形态的处理。

(A)相互鉴别(公开密钥)，权证(PRT)验证(公开密钥)

首先，用图68来说明在相互鉴别处理中应用公开密钥体制、在权证(PRT)验证中应用公开密钥体制的情况下各实体间的数据传送。其中，以下，为了简化说明，如图68所示，设认证机构(CA)为1个，在设备管理器内设定1个注册机构，经这些注册机构、认证机构来发行设备管理器公开密钥证书(Cert. DM)、分区管理器公开密钥证书(Cert. PM)两者。此外，分区注册权证(PRT)的发行部件是设备管理器(DM)，对分区注册权证(PRT)的签名用设备管理器的私有密钥来执行。

按图示的号码顺序在各实体间执行数据传送。以下，根据各号码来说明处理。

(1)发行设备管理器(DM)的公开密钥证书(Cert. DM)

公开密钥证书(Cert. DM)由认证机构(CA)根据设备管理器的发行请求通过经注册机构的证书发行过程向设备管理器发行。

(2)发行分区管理器(PM)的公开密钥证书(Cert. PM)

公开密钥证书(Cert. PM)由认证机构(CA)根据分区管理器的发行请求通过经注册机构的证书发行过程向分区管理器发行。

(3)发行分区注册权证(PRT)

分区注册权证(PRT)由设备管理器管理的分区注册权证发行部件(PRT Ticket Issuer)向分区管理器(PM)发行。在此情况下，执行公

开密钥体制的签名生成、验证，所以用设备管理器的私有密钥来生成签名(Signature)(参照图 12)并附加到 PRT 上.

(4)将 PRT 及 DM 公开密钥证书(Cert. DM)提供给 PM

将设备管理器管理的分区注册权证发行部件(PRT Ticket Issuer)发行的分区注册权证(PRT)与 DM 公开密钥证书(Cert. DM)一起发送到分区管理器.

(5)PM 和设备间的相互鉴别

想要根据发行的 PRT 来创建分区的设备、和分区管理器(具体地说是权证用户---作为设备存取机器的读写器)执行公开密钥体制的相互鉴别(参照图 50).

(6)将 PRT 及 DM 公开密钥证书(Cert. DM)提供给设备

PM 和设备间的相互鉴别成立后，分区管理器(PM)向设备发送分区注册权证(PRT)、及 DM 公开密钥证书(Cert. DM).

设备对接收到的分区注册权证(PRT)执行下述确认: (1)权证发行者(Ticket Issuer)=DM 的公开密钥证书(CERT)是未窜改过的合法的公开密钥证书(CERT); (2)权证发行者(Ticket Issuer)的公开密钥证书(CERT)的选项区域中记录的代码、和设备内的 DKDB (Device Key Definition Block)(PUB)中记录的权证发行部件代码(PRTIC: PRT Issuer Code)一致; (3)权证发行部件(Ticket Issuer)未作废; (4)通过验证所接收权证(PRT)的签名(Signature)而确认权证没有窜改.进而，确认 PRT 权证中保存的 PRT 用户(在此情况下是 PM: 权证用户---作为设备存取机器的读写器)和作为接收到的分区管理器的公开密钥证书的标识数据(DN)而记录的标识符、范畴或序列号(SN)一致，通过确认相互鉴别已完毕来执行 PRT 用户(PM: 作为设备存取机器的读写器)的验证(参照图 57、图 58).

(7)创建分区

分区注册权证(PRT)的验证、PRT 发行者(PRT Issuer)、PRT 用户的验证成功后，根据分区注册权证(PRT)上记述的规则在设备的存储部中创建分区(参照图 60、图 61).

(8)写入密钥数据

在设备的存储部中创建分区后，向创建的分区内保存各种密钥.

(9)读出公开密钥

(10)发行公开密钥证书

在对所创建分区进行各种服务时的鉴别处理(利用分区创建、文件创建、文件存取、数据更新等服务时的鉴别处理)时，在进行公开密钥鉴别的情况下，设备生成公开密钥、私有密钥的密钥对，将生成的公开密钥发送到分区管理器，经注册机构、认证机构进行公开密钥证书的发行处理，将发行的公开密钥证书保存到分区密钥区域(参照图 23)中。此时，向生成的公开密钥的保存区域中保存发行的公开密钥证书。其中，该(9)、(10)的处理在对创建分区进行各种服务时的鉴别处理(利用分区创建、文件创建、文件存取、数据更新等服务时的鉴别处理)时进行公开密钥鉴别的情况下执行即可。

通过以上处理，根据相互鉴别(公开密钥)、权证(PRT)验证(公开密钥)各方式来执行分区创建处理。

(B)相互鉴别(公开密钥)，权证(PRT)验证(对称密钥)

接着，用图 69 来说明在相互鉴别处理中应用公开密钥体制、在权证(PRT)验证中应用对称密钥体制的情况下各实体间的数据传送。按图示的号码顺序在各实体间执行数据传送。以下，根据各号码来说明处理。

(1)发行分区管理器(PM)的公开密钥证书(Cert. PM)

公开密钥证书(Cert. PM)由认证机构(CA)根据分区管理器的发行请求通过经注册机构的证书发行过程向设备管理器发行。

(2)发行分区注册权证(PRT)

分区注册权证(PRT)由设备管理器管理的分区注册权证发行部件(PRT Ticket Issuer)向分区管理器(PM)发行。在此情况下，将 MAC(Message Authentication Code)(参照图 59)作为对称密钥体制的验证值附加到 PRT 上。

(3)将 PRT 提供给 PM

将设备管理器管理的分区注册权证发行部件(PRT Ticket Issuer)发行的分区注册权证(PRT)发送到分区管理器。

(4)PM 和设备间的相互鉴别

想要根据发行的 PRT 来创建分区的设备、和分区管理器(具体地说是权证用户---作为设备存取机器的读写器)执行公开密钥体制的相互鉴别(参照图 50)。

(5)发送PRT

分区管理器将发行的分区注册权证(PRT)送至设备。设备对接收到的分区注册权证(PRT)执行MAC验证处理,执行PRT发行者(PRT Issuer)的验证,进而通过确认PRT权证中保存的PRT用户(在此情况下是PM:权证用户---作为设备存取机器的读写器)和作为接收到的分区管理器的公开密钥证书的标识数据(DN)而记录的标识符、范畴或序列号(SN)一致,来执行PRT用户(PM:作为设备存取机器的读写器)的验证(参照图57、图58)。

(6)创建分区

分区注册权证(PRT)的验证、PRT发行者(PRT Issuer)、PRT用户的验证成功后,根据分区注册权证(PRT)上记述的规则在设备的存储部中创建分区(参照图60、图61)。

(7)写入密钥数据

在设备的存储部中创建分区后,向创建的分区内保存各种密钥。

(8)读出公开密钥

(9)发行公开密钥证书

在对所创建分区进行各种服务时的鉴别处理(利用分区创建、文件创建、文件存取、数据更新等服务时的鉴别处理)时,在进行公开密钥鉴别的情况下,设备生成公开密钥、私有密钥的密钥对,将生成的公开密钥发送到分区管理器,经注册机构、认证机构进行公开密钥证书的发行处理,将发行的公开密钥证书保存到分区密钥区域(参照图23)中。此时,向生成的公开密钥的保存区域中保存发行的公开密钥证书。其中,该(8)、(9)的处理在对创建分区进行各种服务时的鉴别处理(利用分区创建、文件创建、文件存取、数据更新等服务时的鉴别处理)时进行公开密钥鉴别的情况下执行即可。

通过以上处理,根据相互鉴别(公开密钥)、权证(PRT)验证(对称密钥)各方式来执行分区创建处理。

(C)相互鉴别(对称密钥),权证(PRT)验证(对称密钥)

接着,用图70来说明在相互鉴别处理中应用对称密钥体制、在权证(PRT)验证中应用对称密钥体制的情况下各实体间的数据传送。按图示的号码顺序在各实体间执行数据传送。以下,根据各号码来说明处理。

(1)发行分区注册权证(PRT)

分区注册权证(PRT)由设备管理器管理的分区注册权证发行部件(PRT Ticket Issuer)向分区管理器(PM)发行。在此情况下，将MAC(参照图59)作为对称密钥体制的验证值附加到PRT上。

(2)将PRT提供给PM

将设备管理器管理的分区注册权证发行部件(PRT Ticket Issuer)发行的分区注册权证(PRT)发送到分区管理器。

(3)PM和设备间的相互鉴别

想要根据发行的PRT来创建分区的设备、和分区管理器(具体地说是权证用户---作为设备存取机器的读写器)执行对称密钥体制的相互鉴别(参照图53、图54)。

(4)发送PRT

分区管理器将发行的分区注册权证(PRT)送至设备。设备对接收到的分区注册权证(PRT)执行MAC验证处理，执行PRT发行者(PRT Issuer)的验证，进而确认PRT权证中保存的PRT用户(在此情况下是PM: 权证用户---作为设备存取机器的读写器)和分区管理器的标识符一致，通过确认相互鉴别已完毕来执行PRT用户(PM: 作为设备存取机器的读写器)的验证(参照图57、图58)。

(5)创建分区

分区注册权证(PRT)的验证、PRT发行者(PRT Issuer)、PRT用户的验证成功后，根据分区注册权证(PRT)上记述的规则在设备的存储部中创建分区(参照图60、图61)。

(6)写入密钥数据

在设备的存储部中创建分区后，向创建的分区内保存各种密钥。

(7)读出公开密钥

(8)发行公开密钥证书

在对所创建分区进行各种服务时的鉴别处理(利用分区创建、文件创建、文件存取、数据更新等服务时的鉴别处理)时，在进行公开密钥鉴别的情况下，设备生成公开密钥、私有密钥的密钥对，将生成的公开密钥发送到分区管理器，经注册机构、认证机构进行公开密钥证书的发行处理，将发行的公开密钥证书保存到分区密钥区域(参照图23)中。此时，向生成的公开密钥的保存区域中保存发行的公开密钥证书。

其中，该(7)、(8)的处理在对创建分区进行各种服务时的鉴别处理(利用分区创建、文件创建、文件存取、数据更新等服务时的鉴别处理)时进行公开密钥鉴别的情况下执行即可。

通过以上处理，根据相互鉴别(对称密钥)、权证(PRT)验证(公开密钥)各方式来执行分区创建处理。

(D)相互鉴别(对称密钥)，权证(PRT)验证(公开密钥)。

接着，用图 71 来说明在相互鉴别处理中应用对称密钥体制、在权证(PRT)验证中应用公开密钥体制的情况下各实体间的数据传送。按图示的号码顺序在各实体间执行数据传送。以下，根据各号码来说明处理。

(1)发行设备管理器(DM)的公开密钥证书(Cert. DM)

公开密钥证书(Cert. DM)由认证机构(CA)根据设备管理器的发行请求通过经注册机构的证书发行过程向设备管理器发行。

(2)发行分区注册权证(PRT)

分区注册权证(PRT)由设备管理器管理的分区注册权证发行部件(PRT Ticket Issuer)向分区管理器(PM)发行。在此情况下，执行公开密钥体制的签名生成、验证，所以用设备管理器的私有密钥来生成签名(Signature)(参照图 12)并附加到 PRT 上。

(3)将 PRT 及 DM 公开密钥证书(Cert. DM)提供给 PM

将设备管理器管理的分区注册权证发行部件(PRT Ticket Issuer)发行的分区注册权证(PRT)与 DM 公开密钥证书(Cert. DM)一起发送到分区管理器。

(4)PM 和设备间的相互鉴别

想要根据发行的 PRT 来创建分区的设备、和分区管理器(具体地说是权证用户---作为设备存取机器的读写器)执行对称密钥体制的相互鉴别(参照图 53、图 54)。

(6)将 PRT 及 DM 公开密钥证书(Cert. DM)提供给设备

PM 和设备间的相互鉴别成立后，分区管理器(PM)向设备发送分区注册权证(PRT)、及 DM 公开密钥证书(Cert. DM)。

设备对接收到的分区注册权证(PRT)执行下述确认：(1)权证发行者(Ticket Issuer)=DM 的公开密钥证书(CERT)是未窜改过的合法的公开密钥证书(CERT)；(2)权证发行者(Ticket Issuer)的公开密钥证

书(CERT)的选项区域中记录的代码、和设备内的 DKDB (Device Key Definition Block)(PUB)中记录的权证发行部件代码(PRTIC: PRT Issuer Code)一致; (3)权证发行部件(Ticket Issuer)未作废; (4)通过验证所接收权证(PRT)的签名(Signature)而确认权证没有窜改。进而,确认 PRT 权证中保存的 PRT 用户(在此情况下是 PM: 权证用户---作为设备存取机器的读写器)和作为分区管理器的公开密钥证书的标识数据(DN)而记录的标识符、范畴或序列号(SN)一致,通过确认相互鉴别已完毕来执行 PRT 用户(PM: 作为设备存取机器的读写器)的验证(参照图 57、图 58)。

(6)创建分区

分区注册权证(PRT)的验证、PRT 发行者(PRT Issuer)、PRT 用户的验证成功后,根据分区注册权证(PRT)上记述的规则在设备的存储部中创建分区(参照图 60、图 61)。

(7)写入密钥数据

在设备的存储部中创建分区后,向创建的分区内保存各种密钥。

(8)读出公开密钥

(9)发行公开密钥证书

在对所创建分区进行各种服务时的鉴别处理(利用分区创建、文件创建、文件存取、数据更新等服务时的鉴别处理)时,在进行公开密钥鉴别的情况下,设备生成公开密钥、私有密钥的密钥对,将生成的公开密钥发送到分区管理器,经注册机构、认证机构进行公开密钥证书的发行处理,将发行的公开密钥证书保存到分区密钥区域(参照图 23)中。此时,向生成的公开密钥的保存区域中保存发行的公开密钥证书。其中,该(8)、(9)的处理在对创建分区进行各种服务时的鉴别处理(利用分区创建、文件创建、文件存取、数据更新等服务时的鉴别处理)时进行公开密钥鉴别的情况下执行即可。

通过以上处理,根据相互鉴别(对称密钥)、权证(PRT)验证(公开密钥)各方式来执行分区创建处理。

[B4. 4. 利用文件注册权证(FRT)的文件创建、删除处理]

接着,说明在设备中创建的分区内应用文件注册权证(FRT)来创建或删除文件的处理。参照图 72 以下的流程等附图来进行说明。其中,在文件创建、删除处理中,包含设备和作为设备存取机器的读写器(分

区管理器)间的相互鉴别处理(设备鉴别或分区鉴别)、分区注册权证(FRT: File Registration Ticket)完整性验证处理。

下面说明图 72 所示的文件创建、删除处理。在图 72 中，左侧示出分区管理器的文件创建/删除装置的处理，右侧示出设备(参照图 5)的处理。其中，分区管理器的文件创建/删除装置是可对设备进行数据读取写入处理的装置(例如，作为设备存取机器的读写器、PC)，相当于图 10 的作为设备存取机器的读写器。首先，用图 72 来概要说明文件创建、删除处理，其后，用图 73 的流程来详细说明该处理中包含的文件创建、删除操作。

首先，在图 72 的步骤 S801 和 S810 中，执行文件创建/删除装置和设备间的相互鉴别处理。在执行数据发送接收的 2 个部件间，相互确认对方是否是合法的数据通信者，其后进行必要的数据传送。确认对方是否是合法的数据通信者的处理是相互鉴别处理。在相互鉴别处理时执行会话密钥的生成，将生成的会话密钥作为共享密钥来执行加密处理，进行数据发送。

相互鉴别处理是与先前的分区创建、删除处理一栏中说明过的同样的处理，执行分区鉴别。对它们分别应用对称密钥体制鉴别、或公开密钥体制鉴别处理中的某一种。该相互鉴别处理是与前述用图 48～图 56 说明过的同样的处理，所以省略其说明。

其中，作为相互鉴别处理应执行的处理由应用的文件注册权证(FRT)(参照图 27)的

* Authentication Flag: 表示在权证(Ticket)利用处理中是否需要与设备(Device)进行相互鉴别的标志

* Authentication Type: 设备(Device)的相互鉴别类型(公开密钥鉴别、对称密钥鉴别、或任一种皆可(Any))

来决定。

在鉴别处理失败的情况下(S802、S811 中为“否”)，表示不能确认相互是合法的机器、设备，不执行以下的处理，作为出错而结束处理。

如果鉴别处理成功，则文件创建/删除装置向设备发送文件注册权证(FRT: File Registration Ticket)。文件注册权证(FRT)是分区管理器管理下的文件注册权证(FRT)发行部件(FRT Issuer)发行的权

证。文件注册权证(FRT)是对设备的存取控制权证，是具有先前说明过的图 27 的数据格式结构的权证。

其中，在将文件注册权证(FRT)向权证用户发送时，在公开密钥体制的情况下，文件注册权证(FRT)发行部件(FRT Issuer)的公开密钥证书(CERT_FRTI)也一起发送。FRT 发行部件的公开密钥证书(CERT_FRTI)的属性(Attribute)与文件注册权证(FRT)发行部件(FRT Issuer)的标识符(FRTIC)一致。

接收到文件注册权证(FRT)(S812)的设备执行接收到的权证(FRT)的完整性和用户检查处理(S813)。权证完整性验证处理应用基于对称密钥体制的 MAC 验证、或基于公开密钥体制的签名验证处理中的某一种来执行。用户检查是检查发送来权证的机器(权证用户)的完整性的处理，在相互鉴别已成立时，作为验证对方鉴别者的标识数据、和权证中记录的权证用户标识符(参照图 27)是否一致等的处理来执行。这些处理是与先前的分区注册权证(FRT)应用处理的说明中用图 57～图 59 说明过的同样的处理，所以省略其说明。

在设备中，在所接收权证(FRT)的完整性和用户检查处理的结果是未能确认权证及用户合法的情况下(S814 中为“否”)，将文件注册权证(FRT)受理出错通知给文件创建/删除装置(S818)。在得以确认权证及用户合法的情况下(S814 中为“是”)，根据接收到的文件注册权证(FRT)上记述的规则来执行设备内的存储部中的文件创建、或删除处理。后面还将用别的流程来详述该处理。

根据文件注册权证(FRT)的记述，文件创建或删除处理成功(S816 中为“是”)后，将 FRT 受理成功通知给文件创建/删除装置(S817)。而在文件创建或删除处理失败(S816 中为“否”)的情况下，将 FRT 受理出错通知给文件创建/删除装置(S818)。

文件创建/删除装置接收 FRT 受理结果(S804)，判定 FRT 处理结果，在 FRT 受理结果是出错的情况下(S805 中为“否”)，作为出错而结束处理；而在 FRT 受理结果是成功(S805 中为“是”)的情况下，执行会话清除命令的发送接收(S806、S819)，抛弃设备一侧生成的鉴别表(S820)，结束处理。鉴别表是在步骤 S801、S810 的相互鉴别处理中生成的表，与前述分区注册权证(PRT)应用处理的项目中说明过的结构、即图 51 的结构相同。

这样利用文件注册权证(FRT)，在设备内设定的分区内执行新的文件的创建、或已创建的文件的删除。以下，用图 73 来说明该处理中包含的文件创建、删除处理(S815)。

(文件创建/删除处理)

接着,用图 73 的处理流程来详细说明图 72 的流程所示的步骤 S815 中执行的基于文件注册权证(FRT)的分区创建、删除处理。文件创建、删除处理是从权证用户(例如，作为设备存取机器的读写器、PC 等)接收到文件注册权证(FRT)的设备根据文件注册权证(FRT)执行的处理。

在图 73 的步骤 S821 中，设备验证接收到的文件注册权证(FRT: File Registration ticket)上记录的处理类型、即 Operation Type(分区(Partition)创建还是删除的指定(创建(Generate)/删除(Delete)))。在处理类型(Operation Type)是文件创建的情况下，执行步骤 S822 以下的处理；而在处理类型(Operation Type)是文件删除的情况下，执行步骤 S841 以下的处理。

首先，说明文件创建处理。设备在步骤 S822 中验证在设备的待处理分区内是否存在 ID 与文件注册权证(FRT)上记述的文件标识符(ID)相同的文件。该判定可如下来进行：验证在设备的存储部中设定的分区区域的文件定义块(参照图 24)中是否记述有与所接收权证(FRT)上记述的文件 ID 相同的文件 ID。

在设备中已经存在同一 ID 的文件的情况下，不许在同一分区内存在具有同一 ID 的重复文件，所以不执行文件的创建，出错结束。而在待处理分区内不存在同一 ID 的文件的情况下，在步骤 S823 中，对分区管理信息块(参照图 20)的分区内空闲块数(Free Block Number in Partition)、和文件注册权证(FRT)上记述的文件长度(File Size)进行比较，判定在设备的待处理分区内是否存在权证(FRT)上记述的文件长度(File Size)以上的空闲块区域。在不存在的情况下，不能创建 FRT 上记述的长度的文件，所以出错结束。

在判定为在设备的存储部的待处理分区内存在权证 (FRT)上记述的文件长度(File Size)以上的空闲块区域的情况下，进至步骤 S824，参照分区管理信息块的空闲区域指针(Pointer of Free Area)，在分区的空闲区域(Free Area in Partition)的最高块中保留文件定义块(FDB)区域(参照图 24)。

接着，设备向保留的文件定义块(FDB)区域中拷贝文件注册权证(FRT)上记述的文件 ID(PMC)(S825)，进而，向文件定义块(FDB)区域的文件起始位置(File Start Position)上，拷贝分区管理信息块(参照图 20)的空闲区域指针(Pointer of Free Area)(S826)。

进而，在步骤 S827 中，向文件定义块(FDB)的文件长度(File Size)、服务许可权证发行部件代码(SPTIC)、及版本(SPTIC Version)、文件结构类型(File Structure Type Code)、进行文件存取时指定的鉴别方式(Acceptable Authentication Type)、指定的验证方式(Acceptable Verification Type)中分别拷贝文件注册权证(FRT)上记述的各对应数据。

接着，在步骤 S828 中，用文件注册权证的 MAC 验证密钥 Kfrt 对文件注册权证(FRT)上保存的 Kspt_Encrypted(用该分区的文件注册权证的 MAC 验证密钥 Kfrt 对文件定义块(File Definition Block)中记载的服务许可权证(SPT)的 MAC 验证密钥 Kspt 进行加密所得的数据 Kfrt(Kspt))进行解密并保存到文件定义块(FDB)中。其中，文件注册权证的 MAC 验证密钥 Kfrt 已在创建分区时保存到分区密钥区域中。

接着，在 S829 中，将分区管理信息块(参照图 20)的分区内(Partition)内的空闲块数(Free Block Number in Partition)减去文件长度(File Size)+1。其中，+1 表示文件定义块(FDB)用的块。

接着，在步骤 S830 中，将分区管理信息块(参照图 20)的空闲区域指针(Pointer of Free Area)加上创建的文件长度(File Size)，在步骤 S831 中，将分区管理信息块的文件数(File Number)加上 1，即加上创建的文件数(1)。

接着，在步骤 S832 中，按照文件注册权证(FRT)上保存的 File Structure(创建的文件(File)的文件结构(Structure))来执行初始化处理。例如如果文件结构是随机(Random)，则复位到 0；而如果文件结构是循环(Cyclic)，则将指针、数据复位到 0 等。通过这些处理，在创建的分区内创建新的文件。

接着说明图 73 的步骤 S841～S848 的文件删除处理。在步骤 S841 中，验证在设备的存储部的待处理分区内是否存在 ID 与文件注册权证(FRT)上记述的文件 ID 相同的文件。该判定可如下来进行：验证在设备的存储部的分区定义块(参照图 24)中是否记述有与所接收权证(FRT)

上记述的文件 ID 相同的文件 ID。

在设备的待处理分区内不存在同一文件 ID 的文件的情况下，不能删除文件，所以出错结束。而在设备的待处理分区内存在同一 ID 的文件的情况下，在步骤 S842 中，判定在待处理分区内是否存在待删除文件以后创建的文件。在不存在的情况下，待删除文件是最新的文件，在步骤 S849 中删除待删除文件的文件定义块(FDB)(参照图 24)。

在步骤 S842 中，在判定为在待处理分区内存在待删除文件以后创建的文件的情况下，将后创建的文件(后文件)的数据向低处移动相当于待删除文件的长度(FS)的距离(S843)，进而将后文件的文件定义块(FDB)向高处移动 1 个块(S844)。此外，将后文件的文件定义块(FDB)上记录的文件起始位置(File Start Position)减去删除文件的长度(PFS)(S845)。

在步骤 S845 或 S849 的处理后，在步骤 S846 中，将分区管理信息块(PMIB)(参照图 20)的分区内空闲块数(Free Block Number in Partition)加上删除文件的长度(FS)+1。+1 表示删除文件的文件定义块(FPB)用的块。

接着在步骤 S847 中，将分区管理信息块(PMIB)(参照图 20)的空闲区域指针(Pointer of Free Area)的值减去删除文件的长度(FS)。进而，在步骤 S848 中，将分区管理信息块(PMIB)(参照图 20)的文件数(Partition Number)减去 1，即减去已删除的文件数(1)，结束基于文件注册权证(FRT)的文件删除处理。

以上是图 72 的处理流程中的步骤 S815 的基于文件注册权证(FRT)的文件创建、删除处理。

分区管理器执行的文件创建处理完成了的状态下设备的存储器内保存数据结构例示于图 74。在图 74 所示的分区(Partition)区域中，

文件定义块(1～N)(File Definition Block)

分区密钥区域(Partition Key Area)

对称密钥类分区密钥信息块(Partition Key Definition Block(Common))

公开密钥类分区密钥信息块(Partition Key Definition Block(PUB))

分区管理信息块(Partition Management Information Block)

的各数据是在创建文件时、或创建分区时写入的数据。文件区域(File Data Area 1～N)是通过文件创建处理在待处理分区内作为文件区域而保留的。

[B4. 5. 文件创建处理各方式中的处理过程]

在上述文件设定注册处理中，分区管理器进行管理，在文件注册权证用户---作为设备存取机器的读写器和设备间执行相互鉴别，根据文件注册权证(FRT)来设定文件。相互鉴别处理的形态是公开密钥相互鉴别、对称密钥相互鉴别这2种中的某一种，此外，权证(FRT)的验证处理也执行公开密钥类签名验证、对称密钥类MAC验证这2种中的某一种。即处理形态大体分为下述4种形态。

(A)相互鉴别(公开密钥)，权证(PRT)验证(公开密钥)

(B)相互鉴别(公开密钥)，权证(PRT)验证(对称密钥)

(C)相互鉴别(对称密钥)，权证(PRT)验证(对称密钥)

(D)相互鉴别(对称密钥)，权证(PRT)验证(公开密钥)。

以认证机构(CA(PM))、分区管理器(PM)、设备、各实体间执行的数据传送处理为中心，用附图来简洁说明这4种形态的处理。

(A)相互鉴别(公开密钥)，权证(FRT)验证(公开密钥)

首先，用图75来说明在相互鉴别处理中应用公开密钥体制、在权证(FRT)验证中应用公开密钥体制的情况下各实体间的数据传送。

按图示的号码顺序在各实体间执行数据传送。以下，根据各号码来说明处理。

(1)发行文件注册权证发行部件(FRT Issuer)的公开密钥证书(Cert. FRT Issuer)

文件注册权证发行部件(FRT Issuer)的公开密钥证书(Cert. FRT Issuer)根据来自文件注册权证发行部件(FRT Issuer)的发行请求通过经注册机构(RA)的证书发行过程从分区认证机构(CA(PAR))发行。其中，分区管理器也可兼作文件注册权证发行部件(FRT Issuer)，在此情况下，文件注册权证发行部件(FRT Issuer)的公开密钥证书可使用分区管理器(PM)的公开密钥证书。

(2)发行文件注册权证用户(FRT User)的公开密钥证书(Cert. FRT User)

文件注册权证用户(FRT User：具体地说，是向设备发送权证的作

为设备存取机器的读写器)的公开密钥证书(Cert. FRT User)根据来自文件注册权证用户(FRT User)的发行请求通过经注册机构(RA)的证书发行过程由分区认证机构(CA(PAR))发行。其中，分区管理器也可兼作文件注册权证用户(FRT User)，在此情况下，文件注册权证用户(FRT User)的公开密钥证书可使用分区管理器(PM)的公开密钥证书。

(3)文件注册权证(FRT)生成处理

文件注册权证(FRT)由分区管理器管理的文件注册权证发行部件(FRT Ticket Issuer)生成。在此情况下，执行公开密钥体制的签名生成、验证，所以用文件注册权证发行部件(FRT Ticket Issuer)的私有密钥来生成签名(Signature)(参照图 12)并附加到 FRT 上。

(4)将 FRT 及文件注册权证发行部件(FRT Ticket Issuer)公开密钥证书(Cert. FRT Issuer)提供给文件注册权证用户(FRT User)

将分区管理器管理的文件注册权证发行部件(FRT Ticket Issuer)发行的文件注册权证(FRT)与文件注册权证发行部件(FRT Ticket Issuer)公开密钥证书(Cert. FRT Issuer)一起发送到文件注册权证用户(FRT User)、即向设备发送权证的机器(例如，作为设备存取机器的读写器)。

(5)文件注册权证发行部件和设备间的相互鉴别

分区管理器(具体地说是文件注册权证用户(FRT User)---作为设备存取机器的读写器)向想要根据文件注册权证发行部件(FRT Ticket Issuer)发行的文件注册权证(FRT)来创建文件的设备发送权证用户(FRT User)的公开密钥证书(Cert. FRT User)，执行公开密钥体制的相互鉴别(参照图 50)。

(6)将 FRT 及文件注册权证发行部件(FRT Ticket Issuer)公开密钥证书(Cert. FRT Issuer)提供给设备

分区管理器(PM)和设备间的相互鉴别成立后，分区管理器(PM)(具体地说是文件注册权证用户(FRT User)---作为设备存取机器的读写器)向设备发送文件注册权证(FRT)、及文件注册权证发行部件(FRT Ticket Issuer)公开密钥证书(Cert. FRT Issuer)。

设备对接收到的文件注册权证(FRT)执行下述确认：(1)权证发行者(Ticket Issuer)的公开密钥证书(CERT FRT Issuer)是未窜改过的合法的公开密钥证书(CERT)；(2)权证发行者(Ticket Issuer)的公开

密钥证书(CERT FRT Issuer)的选项区域中记录的代码、和设备内的PKDB (Partition Key Definition Block)(PUB)中记录的权证发行部件代码(FRTIC: FRT Issuer Category)一致; (3)权证发行部件(Ticket Issuer)未作废; (4)通过验证所接收权证(FRT)的签名(Signature)而确认权证没有窜改,进而,确认FRT权证中保存的FRT用户(权证用户---作为设备存取机器的读写器)和作为权证用户(FRT User)的公开密钥证书(Cert. FRT User)的标识数据(DN)而记录的标识符、范畴或序列号(SN)一致,通过确认相互鉴别已完毕来执行 FRT用户(作为设备存取机器的读写器)的验证(参照图 57、图 58).

(7)向FDB中注册SPTIC及Kspt

设备向文件定义块(FDB: File Definition Block)中注册服务许可权证(SPT)用户(SPTIC)(例如,对设备的文件内的数据执行存取的作为设备存取机器的读写器)和 Kspt(服务许可权证(SPT)的 MAC 验证密钥(Kspt))(图 73 的流程中的步骤 S827、S828).

(8)保留文件数据区域

设备在待处理分区中保留具有文件注册权证(FRT)上记述的长度的文件区域.

通过以上处理,根据相互鉴别(公开密钥)、权证(FRT)验证(公开密钥)各方式来执行文件创建处理.

(B)相互鉴别(公开密钥),权证(FRT)验证(对称密钥)

接着,用图 76 来说明在相互鉴别处理中应用公开密钥体制、在权证(FRT)验证中应用对称密钥体制的情况下各实体间的数据传送.

按图示的号码顺序在各实体间执行数据传送.以下,根据各号码来说明处理.

(1)发行文件注册权证用户(FRT User)的公开密钥证书(Cert. FRT User)

文件注册权证用户(FRT User:具体地说,是向设备发送权证的作为设备存取机器的读写器)的公开密钥证书(Cert. FRT User)根据来自文件注册权证用户(FRT User)的发行请求通过经注册机构(RA)的证书发行过程由分区认证机构(CA(PAR))发行.其中,分区管理器也可兼作文件注册权证用户(FRT User),在此情况下,文件注册权证用户(FRT User)的公开密钥证书可使用分区管理器(PM)的公开密钥证书.

(2)文件注册权证(FRT)生成处理

文件注册权证(FRT)由分区管理器管理的文件注册权证发行部件(FRT Ticket Issuer)生成。在此情况下，作为对称密钥体制的验证值将 MAC (Message Authentication Code)(参照图 59)附加到 FRT 上。

(3)将 FRT 提供给文件注册权证用户(FRT User)

将分区管理器管理的文件注册权证发行部件(FRT Ticket Issuer)发行的文件注册权证(FRT)发送到文件注册权证用户(FRT User)、即向设备发送权证的机器(例如，作为设备存取机器的读写器)。

(4)文件注册权证发行部件和设备间的相互鉴别

分区管理器(具体地说是文件注册权证用户(FRT User)---作为设备存取机器的读写器)向想要根据文件注册权证发行部件(FRT Ticket Issuer)发行的文件注册权证(FRT)来创建文件的设备发送权证用户(FRT User)的公开密钥证书(Cert. FRT User)，执行公开密钥体制的相互鉴别(参照图 50)。

(5)将 FRT 提供给设备

分区管理器(PM)和设备间的相互鉴别成立后，分区管理器(PM)(具体地说是文件注册权证用户(FRT User)---作为设备存取机器的读写器)向设备发送文件注册权证(FRT)。设备对接收到的文件注册权证(FRT)执行 MAC 验证处理，执行 FRT 发行者(FRT Issuer)的验证，进而确认 FRT 权证中保存的 FRT 用户(权证用户---作为设备存取机器的读写器)和作为接收到的分区管理器的公开密钥证书的标识数据(DN)而记录的标识符、范畴或序列号(SN)一致，通过确认相互鉴别已完毕来执行 FRT 用户(PM: 作为设备存取机器的读写器)的验证(参照图 57、图 58)。

(6)向 FDB 中注册 SPTIC 及 Kspt

设备向文件定义块(FDB: File Definition Block)中注册服务许可权证(SPT)发行者范畴(SPTIC)(例如，对设备的文件内的数据执行存取的作为设备存取机器的读写器)和 Kspt(服务许可权证(SPT)的 MAC 验证密钥(Kspt))(图 73 的流程中的步骤 S827、S828)。

(7)保留文件数据区域

设备在待处理分区中保留具有文件注册权证(FRT)上记述的长度

的文件区域。

通过以上处理，根据相互鉴别(公开密钥)、权证(FRT)验证(对称密钥)各方式来执行文件创建处理。

(C)相互鉴别(对称密钥)，权证(FRT)验证(对称密钥)

接着，用图 77 来说明在相互鉴别处理中应用对称密钥体制、在权证(FRT)验证中应用对称密钥体制的情况下各实体间的数据传送。

按图示的号码顺序在各实体间执行数据传送。以下，根据各号码来说明处理。

(1)文件注册权证(FRT)生成处理

文件注册权证(FRT)由分区管理器管理的文件注册权证发行部件(FRT Ticket Issuer)生成。在此情况下，作为对称密钥体制的验证值将 MAC (Message Authentication Code)(参照图 59)附加到 FRT 上。

(2)将 FRT 提供给文件注册权证用户(FRT User)

将分区管理器管理的文件注册权证发行部件(FRT Ticket Issuer)发行的文件注册权证(FRT)发送到文件注册权证用户(FRT User)、即向设备发送权证的机器(例如，作为设备存取机器的读写器)。

(3)文件注册权证发行部件和设备间的相互鉴别

分区管理器(具体地说是文件注册权证用户(FRT User)---作为设备存取机器的读写器)与想要根据文件注册权证发行部件(FRT Ticket Issuer)发行的文件注册权证(FRT)来创建文件的设备执行对称密钥体制的相互鉴别(参照图 53、图 54)。

(4)将 FRT 提供给设备

分区管理器(PM)和设备间的相互鉴别成立后，分区管理器(PM)(具体地说是文件注册权证用户(FRT User)---作为设备存取机器的读写器)向设备发送文件注册权证(FRT)。设备对接收到的文件注册权证(FRT)执行 MAC 验证处理，执行 FRT 发行者(FRT Issuer)的验证，进而确认 FRT 权证中保存的 FRT 用户(权证用户---作为设备存取机器的读写器)和接收到的分区管理器的标识符一致，通过确认相互鉴别已完毕来执行 FRT 用户(PM: 作为设备存取机器的读写器)的验证(参照图 57、图 58)。

(5)向 FDB 中注册 SPTIC 及 Kspt

设备向文件定义块(FDB: File Definition Block)中注册服务许可权证(SPT)发行者范畴(SPTIC)(例如,对设备的文件内的数据执行存取的作为设备存取机器的读写器)和 Kspt(服务许可权证(SPT)的 MAC 验证密钥(Kspt))(图 73 的流程中的步骤 S827、S828)。

(6)保留文件数据区域

设备在待处理分区中保留具有文件注册权证(FRT)上记述的长度的文件区域。

通过以上处理,根据相互鉴别(对称密钥)、权证(FRT)验证(公开密钥)各方式来执行文件创建处理。

(D)相互鉴别(对称密钥),权证(FRT)验证(公开密钥)。

接着,用图 78 来说明在相互鉴别处理中应用对称密钥体制、在权证(FRT)验证中应用公开密钥体制的情况下各实体间的数据传送。

按图示的号码顺序在各实体间执行数据传送。以下,根据各号码来说明处理。

(1)发行文件注册权证发行部件(FRT Issuer)的公开密钥证书(Cert. FRT Issuer)

文件注册权证发行部件(FRT Issuer)的公开密钥证书(Cert. FRT Issuer)根据来自文件注册权证发行部件(FRT Issuer)的发行请求通过经注册机构(RA)的证书发行过程从分区认证机构(CA(PAR))发行。其中,分区管理器也可兼作文件注册权证发行部件(FRT Issuer),在此情况下,文件注册权证发行部件(FRT Issuer)的公开密钥证书可使用分区管理器(PM)的公开密钥证书。

(2)文件注册权证(FRT)生成处理

文件注册权证(FRT)由分区管理器管理的文件注册权证发行部件(FRT Ticket Issuer)生成。在此情况下,执行公开密钥体制的签名生成、验证,所以用文件注册权证发行部件(FRT Ticket Issuer)的私有密钥来生成签名(Signature)(参照图 12)并附加到 FRT 上。

(3)将 FRT 及文件注册权证发行部件(FRT Ticket Issuer)公开密钥证书(Cert. FRT Issuer)提供给文件注册权证用户(FRT User)

将分区管理器管理的文件注册权证发行部件(FRT Ticket Issuer)发行的文件注册权证(FRT)与文件注册权证发行部件(FRT Ticket Issuer)公开密钥证书(Cert. FRT Issuer)一起发送到文件注册权证

用户(FRT User)、即向设备发送权证的机器(例如，作为设备存取机器的读写器)。

(4)文件注册权证发行部件和设备间的相互鉴别

分区管理器(具体地说是文件注册权证用户(FRT User)---作为设备存取机器的读写器)与想要根据文件注册权证发行部件(FRT Ticket Issuer)发行的文件注册权证(FRT)来创建文件的设备执行对称密钥体制的相互鉴别(参照图53、图54)。

(5)将FRT及文件注册权证发行部件(FRT Ticket Issuer)公开密钥证书(Cert. FRT Issuer)提供给设备

分区管理器(PM)和设备间的相互鉴别成立后，分区管理器(PM)(具体地说是文件注册权证用户(FRT User)---作为设备存取机器的读写器)向设备发送文件注册权证(FRT)、及文件注册权证发行部件(FRT Ticket Issuer)公开密钥证书(Cert. FRT Issuer)。

设备对接收到的文件注册权证(FRT)执行下述确认：(1)权证发行者(Ticket Issuer)的公开密钥证书(CERT FRT Issuer)是未窜改过的合法的公开密钥证书(CERT)；(2)权证发行者(Ticket Issuer)的公开密钥证书(CERT FRT Issuer)的选项区域中记录的代码、和设备内的PKDB (Partition Key Definition Block)(PUB)中记录的权证发行部件代码(FRTIC: FRT Issuer Category)一致；(3)权证发行部件(Ticket Issuer)未作废；(4)通过验证所接收权证(FRT)的签名(Signature)而确认权证没有窜改，进而，确认FRT权证中保存的FRT用户(权证用户---作为设备存取机器的读写器)和权证用户(FRT User)的标识符一致，通过确认相互鉴别已完毕来执行FRT用户(作为设备存取机器的读写器)的验证(参照图57、图58)。

(6)向FDB中注册SPTIC及Kspt

设备向文件定义块(FDB: File Definition Block)中注册服务许可权证(SPT)发行者范畴(SPTIC)(例如，对设备的文件内的数据执行存取的作为设备存取机器的读写器)和Kspt(服务许可权证(SPT)的MAC验证密钥(Kspt))(图73的流程中的步骤S827、S828)。

(7)保留文件数据区域

设备在待处理分区中保留具有文件注册权证(FRT)上记述的长度的文件区域。

通过以上处理，根据相互鉴别(对称密钥)、权证(FRT)验证(公开密钥)各方式来执行文件创建处理。

[B4. 6. 利用服务许可权证(SPT)的服务(文件存取)处理]

接着，说明利用服务许可权证(SPT)(参照图 28、图 31)的文件存取处理。参照图 79 以下的流程等附图来进行说明。在文件存取处理中，包含设备和文件存取装置间的相互鉴别处理(设备鉴别或分区鉴别)、服务许可权证(SPT: Service Permission Ticket)完整性验证处理。

在图 79 的流程中，左侧示出文件存取装置的处理，右侧示出设备(参照图 5)的处理。其中，文件存取装置是分区管理器管理的装置，是可对设备进行数据读取写入处理的装置(例如，作为设备存取机器的读写器、PC)，相当于图 10 的作为设备存取机器的读写器。首先，用图 79 来概要说明文件存取装置执行的文件存取处理，其后，用图 80 以下的流程来依次详细说明该处理中包含的各处理。

首先，在图 79 的步骤 S851 和 S860 中，执行文件存取装置和设备间的相互鉴别处理。在执行数据发送接收的 2 个部件间，相互确认对方是否是合法的数据通信者，其后进行必要的数据传送。确认对方是否是合法的数据通信者的处理是相互鉴别处理。在相互鉴别处理时执行会话密钥的生成，将生成的会话密钥作为共享密钥来执行加密处理，进行数据发送。

相互鉴别处理是与先前的分区创建、删除处理一栏中说明过的同样的处理，执行分区鉴别。对它们分别应用对称密钥体制鉴别、或公开密钥体制鉴别处理中的某一种。该相互鉴别处理是与前述用图 48～图 56 说明过的同样的处理，所以省略其说明。

其中，作为相互鉴别处理应执行的处理由应用的服务许可权证(SPT)(参照图 28、图 31)的

* Authentication Flag: 表示在权证(Ticket)利用处理中是否需要与设备(Device)进行相互鉴别的标志

* Authentication Type: 设备(Device)的相互鉴别类型(公开密钥鉴别、对称密钥鉴别、或任一种皆可(Any))

来决定。

在鉴别处理失败的情况下(S852、S861 中为“否”)，表示不能确认相互是合法的机器、设备，不执行以下的处理，作为出错而结束处

理。

设备也可容许根据多个服务许可权证(SPT)来进行多个不同分区内的文件存取。例如，以设备鉴别成立为条件，可容许根据多个服务许可权证(SPT)来进行多个不同分区内的文件存取。每个分区的文件存取规则被记述在作为存取控制数据而构成的服务许可权证(SPT)上，设备从存取机器接收多个服务许可权证(SPT)，在各权证要求设备鉴别的情况下，根据记述以设备鉴别已成立为条件来容许各分区内的文件存取。

此外，在多个服务许可权证(SPT)分别确定了不同的鉴别条件的情况下，以各服务许可权证(SPT)上设定的分区鉴别的鉴别成立为条件，容许存取多个服务许可权证(SPT)的指定文件。

接着在步骤 S853 中，文件存取装置向设备发送服务许可权证(SPT: Service Permission Ticket)。服务许可权证(SPT)是分区管理器管理下的服务许可权证(SPT)发行部件(SPT Issuer)发行的权证。服务许可权证(SPT)是对设备的存取控制权证，是具有先前说明过的图 28、图 31 的数据格式结构的权证。

其中，在将服务许可权证(SPT)向权证用户发送时，在公开密钥体制的情况下，服务许可权证(SPT)发行部件(SPT Issuer)的公开密钥证书(CERT_SPTI)也一起发送。SPT 发行部件的公开密钥证书(CERT_SPTI)的属性(Attribute)与设备内的 FDB(File Definition Block)中记录的权证发行部件代码(SPTIC)一致。

接收到服务许可权证(SPT)(S862)的设备执行接收到的权证(SPT)的完整性和用户检查处理(S863)。权证完整性验证处理应用基于对称密钥体制的 MAC 验证、或基于公开密钥体制的签名验证处理中的某一种来执行。用户检查是检查发送来权证的机器(权证用户)的完整性的处理，在相互鉴别已成立时，作为验证对方鉴别者的标识数据、和权证中记录的权证用户标识符(参照图 28、图 31)是否一致等的处理来执行。这些处理是与先前的分区注册权证(PRT)应用处理的说明中用图 57～图 59 说明过的同样的处理，所以省略其说明。

在设备中，在所接收权证(SPT)的完整性和用户检查处理的结果是未能确认权证及用户合法的情况下(S864 中为“否”)，将服务许可权证(SPT)受理出错通知给文件存取装置(S868)。在得以确认权证及用户

合法的情况下(S864 中为“是”)，根据接收到的服务许可权证(SPT)上记述的规则来执行设备内的存储部中保存的文件的打开处理。后面还将用别的流程来详述该处理。

根据服务许可权证(SPT)的记述，文件打开处理成功(S866 中为“是”)后，将 SPT 受理成功通知给文件存取装置(S867)。而在文件打开处理失败(S866 中为“否”)的情况下，将 SPT 受理出错通知给文件存取装置(S868)。

文件存取装置接收 SPT 受理结果(S854)，判定 SPT 处理结果，在 SPT 受理结果是出错的情况下(S855 中为“否”)，作为出错而结束处理；而在 SPT 受理结果是成功(S855 中为“是”)的情况下，判定所有 SPT 发送是否都已结束(S856)，在有未发送 SPT 的情况下，重复执行步骤 S853 以下的处理。

在所有 SPT 发送都已结束的情况下，在步骤 S857、S869 中根据服务许可权证(SPT)来执行文件存取，在文件存取处理结束后，执行会话清除命令的发送接收(S858、S870)，抛弃设备一侧生成的鉴别表(S871)，结束处理。后面还将用别的流程来详述文件存取处理。其中，鉴别表是在步骤 S851、S860 的相互鉴别处理中生成的表，与前述分区注册权证(PRT)应用处理的项目中说明过的结构、即图 51 的结构相同。

这样利用服务许可权证(SPT)，对设备内设定的分区内的文件执行存取处理。以下，说明该处理中包含的文件打开处理(S865)、各种存取处理(S857、S869)。

(文件打开处理)

根据图 80 的流程，来说明文件打开处理(图 79，S865)。文件打开处理是根据设备接收到的服务许可权证(SPT)执行的处理。

在步骤 S881 中，设备判定在设备内是否创建了并存在接收到的服务许可权证(SPT)所指定的文件。在服务许可权证(SPT)中记录有待处理文件的文件 ID(参照图 28、图 31)，例如参照文件定义块(图 24)来判定有无具有同一 ID 的文件。在不存在 ID 与权证上记述的 ID 相同的文件的情况下，不能处理，所以出错结束。

在存在 ID 与权证上记述的 ID 相同的文件的情况下，在步骤 S882 中，向文件打开表中写入将服务许可权证(SPT)上记述的权证发行部件

(Ticket Issuer=PMC: Partition Manager Code)、和服务许可权证(SPT)上记述的文件 ID 相对应的项.

进而，在步骤 S883 中，向文件打开表中与生成的项相对应写入服务许可权证(SPT)上记述的文件存取模式[File Access Mode: 对允许存取的文件(File)的存取模式(Access Mode)]，在步骤 S884 中，写入服务许可权证(SPT)上记述的存取许可文件组[Group of Target File: 允许存取的文件(File)的组(Group)]，在步骤 S885 中，写入服务许可权证(SPT)上记述的存取许可文件标识符[Target File ID: 允许存取的文件(File)的标识符(ID)]，进而在步骤 S886 中，写入服务许可权证(SPT)上记述的对目标文件(Target File)的处理形态数据[Read/Write Permission: 对允许存取的文件(File)(目标文件(Target File))的处理形态(读出(Read)、写入(Write)的许可). 其中，作为对目标文件的处理，不限于读出(Read)、写入(Write)，可设定各种处理.

文件打开表的结构例示于图 81、图 82. 文件打开表是在设备中记录了处于存取处理状态的文件及存取模式等信息的表，记录设备接收到的服务许可权证(SPT)的记述信息并保存到设备的存储部件中.

在权证是只许可存取单个文件的形式的服务许可权证(参照图 28)的情况下，文件打开表保存

* Ticket Issuer: 权证发行部件(Ticket Issuer)的标识符

* File ID: 分区内的存取文件(File)的标识符(ID)

* File Access Mode: 对允许存取的文件(File)的存取模式(Access Mode)

的信息. 此情况下的文件打开表的结构例示于图 81.

如图 81 所示，在文件打开表中记述有分区管理器代码(PMC)作为组信息---Ticket Issuer“权证发行部件(Ticket Issuer)”的标识符，来判别分区，通过文件 ID 来标识文件，通过文件存取模式可判定可执行的存取形态(例如，读取(READ)、写入(Write)、加密解密(Enc、Dec)).

此外，在服务许可权证(SPT)是许可存取分区中设定的文件中的多个文件的形式的服务许可权证(参照图 31)的情况下，除了上述信息之外，还将

* Group of Target File: 允许存取的文件(File)的组(Group)

* Target File ID: 允许存取的文件(File)的标识符(ID)

* Read/Write Permission: 对允许存取的文件(File)(目标文件(Target File)) 的处理形态(读出(Read)、写入(Write))的许可

各信息写入表中。此情况下的文件打开表的结构例示于图 82。

如图 82 所示，在与许可存取多个文件的形式的服务许可权证对应而设定的文件打开表中，除了图 81 所示的数据之外，还保存有作为允许存取的目标文件(File)的组的作为分区标识数据的分区管理器代码(PMC)、作为允许存取的目标文件(File)的标识符(ID)的文件 ID、以及表示对目标文件(Target File)的处理形态的[Read/Write Permission]数据，可判定可对多个文件执行的处理。

对多个文件执行存取的处理例如是指用文件 A 中保存的密钥对文件 B 中保存的数据进行加密的情况等。为此，文件 B 需要对文件 A 的读出请求给予许可。在此情况下，将文件 B 称为源文件，将被给予许可的文件称为目标文件。

这样，设备根据与设备机器进行会话时接收到的服务许可权证(SPT)，来生成将作为权证发行部件(Ticket Issuer(PMC))的分区管理器代码(PMC)、作为执行文件打开处理的文件的标识数据的文件标识符、以及服务许可权证(SPT)上记述的存取模式相对应的文件打开表，可参照该文件打开表来判定可否执行从上述设备机器接收到的命令。

(文件存取处理)

接着，详细说明图 79 的步骤 S857、S869 中执行的文件存取处理。

首先，用图 83 来说明生成了图 81 所示的文件打开表的情况下的存取处理。图左侧示出 2 个文件存取装置(R/W: 作为设备存取机器的读写器)750、760，右侧示出创建了文件的设备 100 的分区部分。

文件存取装置(R/W: 读写器)750 与设备进行相互鉴别后，将

文件 ID: [0x0002]

文件存取模式: [读取: Read]

的存取许可权证发送到设备 100，认为权证完整性验证、权证发行者、用户验证成功。

此时，在设备中生成图 81 所示的文件打开表的第 2 行的项。该项表示可对用分区管理器代码(PMC1)标识的分区内的文件 ID[0x0002]

执行存取模式[读取: Read]的处理。

此时，文件存取装置(R/W: 读写器)750 生成命令并发送到设备。在设备例如接收到文件 ID[0x0002]的数据读取命令“Read Command(0x0002)”后，设备确认文件打开表的项，确认可对文件 ID[0x0002]执行存取模式[读取: Read]的处理，执行读取处理。

此外，在文件存取装置(R/W: 读写器)750 例如将文件 ID[0x0002]的数据写入命令“Write Command(0x0002)”、或文件 ID[0x0001]的数据加密处理命令“Encryption Command(0x0001)”发送到设备的情况下，接收到命令的设备确认文件打开表的项，确认从文件存取装置(R/W: 读写器)750 接收到的服务许可权证(SPT)未许可对文件 ID[0x0002]执行[写入: Write]处理、及文件 ID[0x0001]的[加密处理]，停止处理。

此外，文件存取装置(R/W: 读写器)760 与设备进行相互鉴别后，将

文件 ID: [0x0001]

文件存取模式: [加密解密处理: Enc&Dec]

的存取许可权证发送到设备 100，认为权证完整性验证、权证发行者、用户验证成功。

此时，在设备中生成图 81 所示的文件打开表的第 1 行的项。该项表示可对用分区管理器代码(PMC1)标识的分区内的文件 ID[0x0001]执行存取模式[加密解密处理: Enc&Dec]的处理。

此时，文件存取装置(R/W: 读写器)760 生成命令并发送到设备。在设备例如接收到文件 ID[0x0001]的加密命令[Encryption Command(0x0001)]后，设备确认文件打开表的项，确认可对文件 ID[0x0001]执行存取模式[加密解密处理: Enc&Dec]的处理，执行加密处理。

此外，在文件存取装置(R/W: 读写器)760 例如将文件 ID[0x0002]的数据读取命令“Read Command(0x0002)”发送到设备的情况下，接收到命令的设备确认文件打开表的项，确认从文件存取装置(R/W: 读写器)760 接收到的服务许可权证(SPT)未许可对文件 ID[0x0002]的[读取: Read]处理，停止处理。

这样，根据设备从服务许可权证用户---作为设备存取机器的读写

器接收到的服务许可权证(SPT)，设备根据前述的图 80 的处理流程来生成文件打开表，根据生成的文件打开表来决定可否执行来自文件存取装置---读写器的各命令，根据决定来执行处理。

接着，用图 84 来说明对 2 个文件执行处理的情况下的存取处理。图左侧示出 2 个文件存取装置(R/W: 作为设备存取机器的读写器)770、780，右侧示出创建了文件的设备 100 的分区部分。

首先,说明使用指定了目标文件的服务许可权证(SPT)(参照图 31)的处理的执行例。

文件存取装置(R/W: 读写器)770 与设备进行相互鉴别后，将

SPT 格式 1

文件 ID: [0x0001]

文件存取模式: [加密解密处理: Enc&Dec]

及 SPT 格式 2

文件 ID: [0x0002]

目标文件组: [PMC1]

目标文件 ID: [0x0001]

读写许可: [读取: Read]

这 2 个存取许可权证发送到设备 100，认为权证完整性验证、权证发行者、用户验证成功。

此时，在设备中生成图 82 所示的文件打开表的项。该项表示，用分区管理器代码(PMC1)标识的分区内的文件 ID[0x0001]是密钥文件，为了进行加密、解密而被打开。文件 ID[0x0002]是数据文件，文件存取模式(File Access Mode)一栏为空白，所以不能从外部读出，为了可对文件 ID[0x0001]执行[读取: Read]而被打开，被设定为文件打开表的项。

此时，文件存取装置(R/W: 读写器)770 生成命令并发送到设备。例如发送读取文件 ID[0x0002]、用文件 ID[0x0001]进行内部加密的命令“Internal Encryption Command(0x0001, 0x0002)”，在设备接收到命令后，设备确认文件打开表的项，判定可对文件 ID[0x0002]执行[读取: Read]，从而执行目标文件组[PMC1]、目标文件[0x0001]的[加密处理]，读取 ID[0x0002]的数据，用文件 ID[0x0001]的密钥(key)执行加密并将加密数据发送到存取装置。

根据使用指定了该目标文件的服务许可权证(SPT)(参照图 31)的处理，可取得将从某个文件读出的数据用另一个文件中保存的加密密钥进行加密所得数据，不用担心解密数据泄漏到外部。

接着，说明不是使用指定了目标文件的服务许可权证(SPT)(参照图 31)、而是使用多个指定了对单个文件的处理的服务许可权证(SPT)(参照图 28)的情况下的处理。

文件存取装置(R/W: 读写器)780 与设备进行相互鉴别后，将

作为 SPT 格式 1，

文件 ID: [0x0002]

文件存取模式: [读取: Read]

此外，作为 SPT 格式 2，

文件 ID: [0x0001]

文件存取模式: [加密解密处理: Enc&Dec]

这 2 个存取许可权证发送到设备 100，认为权证完整性验证、权证发行者、用户验证成功。

此时，在设备中生成图 81 所示的文件打开表的第 1 行及第 2 行的各项。该项表示可对用分区管理器代码(PMC1)标识的分区内的文件 ID[0x0001]执行存取模式[加密解密处理: Enc&Dec]的处理，可对用分区管理器代码(PMC1)标识的分区内的文件 ID[0x0002]执行存取模式[读取: Read]的处理。

此时，文件存取装置(R/W: 读写器)780 生成命令并发送到设备。首先，在设备接收到文件 ID“[0x0002]”的数据读取命令“Read Command(0x0002)”后，设备确认文件打开表的项，确认可对文件 ID“0x0002”执行存取模式[读取: Read]的处理，执行读取处理，向文件存取装置发送读取数据。

接着，文件存取装置(R/W: 读写器)780 又生成命令并发送到设备。在设备接收到用文件 ID[0x0001]对数据(Data)进行加密的命令[Encryption Command(0x0001, Data)]后，设备确认文件打开表的项，确认可对文件 ID[0x0001]执行存取模式[加密解密处理: Enc&Dec]的处理，执行加密处理，将加密数据[Encryption Data]发送到文件存取装置(R/W: 读写器)780。

这样，在不是使用指定了目标文件的服务许可权证(SPT)(参照图

31)、而是使用多个指定了对单个文件的处理的服务许可权证(SPT)(参照图 28)的情况下，执行待加密数据读出处理，文件存取装置 780 和设备间的数据传送处理的次数增加。此外，数据未被加密就被读出到设备外。

另一方面，使服务许可权证(SPT)(参照图 31)包含标识待存取的多个数据文件的多个文件标识符，在该多个文件标识符中，一个被设定为目标文件标识符，并且保存对目标文件的读取或写入许可数据，作为另一个数据文件的存取模式，设定使用该数据文件中保存的加密密钥的加密处理，如果采用上述结构，则搭载有存储器的设备从存取机器接收服务许可权证(SPT)，作为根据指定存取模式执行的处理，读取目标文件并用加密密钥进行加密处理，可执行搭载有存储器的设备内的内部加密处理。此外，能够防止数据未被加密就流出到设备外。

发行服务许可权证(SPT)的权证发行部件是管理搭载有存储器的设备的存储区域的实体---分区管理器管理下的权证发行部件，个别地发行按照各存取机器设定了各种存取模式的服务许可权证(SPT)，实现了可执行形态因各存取机器而异的存取的结构。

(会话密钥的使用形态)

其中，文件存取装置和设备间发送接收的数据多是设备的用户信息、或金额信息等应防止泄漏到外部的数据。因此，最好对文件存取装置和设备间发送接收的数据执行加密处理，并且在数据上附加作为窜改检查值的 MAC (Message Authentication Code)。

在数据的加密中可使用文件存取装置和设备间执行的相互鉴别处理中生成的会话密钥。如前所述，相互鉴别有对设备的设备鉴别、作为对各分区的鉴别的分区鉴别。在对分区中已创建的文件执行存取的情况下，传送数据时应用的加密密钥应用哪一种，有几个选择肢。

例如如图 85 所示，在设备 100 和存取装置 800 之间，有时有通过设备鉴别而生成的会话密钥 Kses1、和与分区管理器代码(PMC1)对应的分区进行分区鉴别而生成的会话密钥 Kses2、和与分区管理器代码(PMC2)对应的分区进行分区鉴别而生成的会话密钥 Kses3。

它们被保存在相互鉴别时生成的鉴别表(参照图 51、图 52)中，直至会话清除。

设备和与设备执行通信的作为设备存取机器的读写器(PC 等通信

装置)可将应用该多个会话密钥中的哪一个来执行加密通信作为规则来预先决定，根据决定的规则来应用会话密钥。

在以与多个不同分区分别对应而设定的鉴别条件---分区鉴别或设备鉴别的鉴别都成立为条件而容许多个不同分区内的文件存取的情况下，根据进行多个鉴别处理而取得的多个会话密钥来生成单个统一会话密钥，根据该统一会话密钥来执行与存取机器的通信数据的加密处理。

统一会话密钥生成手法中的一种方法是，在通过设备和与设备执行通信的作为设备存取机器的读写器(PC 等通信装置)间的相互鉴别处理而生成了多个会话密钥 Kses1～KsesN 的情况下，执行该多个会话密钥 Kses1～KsesN 的逻辑“异或”处理(例如，8 字节处理)，将运算结果作为通信数据的加密会话密钥。即，将通过

Kses=Kses1 XOR Kses2 XOR Kses3…

XOR：逻辑“异或”处理(例如，8 字节处理)

而算出的 Kses 用作会话密钥。

在设备和与设备执行通信的作为设备存取机器的读写器(PC 等通信装置)间，决定下述规则：对双方的鉴别表中保存的会话密钥进行逻辑“异或”运算，将其输出值用作会话密钥；根据该规则来计算会话密钥，用于通信数据的加密。此外，同样，可以用相互鉴别时同时共享的别的会话密钥、例如公开密钥鉴别时生成的会话密钥、或会话密钥生成数据、例如 Y 坐标的低 64 比特在会话中的通信数据上附加 MAC 值。将该 MAC 值与通信数据(有时是加密数据)一起发送，接收端进行 MAC 鉴别处理，从而可防止通信线路上的数据窜改。MAC 生成、验证处理请参照先前说明过的图 59。

或者，也可以设定下述规则：在通过设备和与设备执行通信的作为设备存取机器的读写器(PC 等通信装置)间的相互鉴别处理而取得的多个会话密钥 Kses1～KsesN 中，选择某 1 个密钥(例如，最新的会话密钥)，用作其后的通信处理中的数据加密密钥；根据该规则来选择会话密钥，用于通信数据的加密。

其中，上述通过多个会话密钥的运算执行的计算、或选择处理不仅可以应用于文件存取装置和设备间的加密通信，而且可以应用于所有权证(PRT、FRT、SPT、DUT)用户(作为设备存取机器的读写器等与设

备执行数据通信的机器)和设备间的加密通信处理中通过相互鉴别而生成了多个会话密钥的情况。可采用下述等措施：在各权证用户和设备之间，可将根据哪种规则由多个会话密钥来计算或选择要应用的会话密钥预先作为规则来决定，在相互确认了规则后执行，或在各权证上记录规则。

接着，用图 86、图 87 来说明文件存取装置对设备执行的存取处理(图 79 的处理流程中的步骤 S857、S869)过程的代表例。

用图 86 来说明只对 1 个文件执行存取的情况下的处理(Normal)，用图 87 来说明对多个文件进行存取的情况下的处理(Combination)。

首先，说明图 86 的只对 1 个文件执行存取的情况下的处理(Normal)。在图 86 的流程中，左侧示出文件存取装置，右侧示出设备(参照图 5)。其中，在文件存取处理中，在文件存取装置和设备间传送数据时，用相互鉴别处理中取得的会话密钥 Kses、或由多个会话密钥进行运算或选择所得的会话密钥来进行加密，并且执行窜改检查用的 MAC 的生成、验证处理。

文件存取装置在步骤 S891 中将存取命令发送到设备。该命令是指定待存取文件 ID、存取模式的命令，例如是先前用图 83 说明过的文件 ID[0x0002]的数据读取命令“Read Command(0x0002)”、或文件 ID[0x0001]的加密命令[Encryption Command(0x0001)]等。

设备接收到来自文件存取装置的命令(S901)后，判定命令中包含的文件 ID、存取模式在文件打开表是否已被记录为许可的项(S902)。在文件打开表中不存在与命令对应的文件 ID、存取模式的项的情况下，不根据命令来执行处理，将存取出错发送到文件存取装置(S908)。

在文件打开表中存在与命令对应的文件 ID、存取模式的项的情况下，在步骤 S903 中，参照设备的存储器内的对应分区的文件定义块(FDB)(参照图 24)中记录的文件存取鉴别方式(Acceptable Authentication Type：在进行特定的文件(File)存取时指定的鉴别方式)，来确认执行存取待存取文件的命令所需的鉴别级别(是否需要公开密钥鉴别)。

在步骤 S903 中，在文件定义块(FDB)的文件存取鉴别方式(Acceptable Authentication Type)被设定为需要公开密钥鉴别的情况下，在步骤 S904 中，判定存取命令所需的鉴别级别的鉴别---公开

密钥鉴别是否已完毕，在鉴别未完的情况下，不根据命令来执行处理，将存取出错发送到文件存取装置(S908)。鉴别结束或未完根据相互鉴别时设定的鉴别表(参照图 51)来判定。

在步骤 S903 中，在文件定义块(FDB)的文件存取鉴别方式(Acceptable Authentication Type)被设定为需要公开密钥鉴别、在步骤 S904 中判定为公开密钥鉴别已完毕的情况下，或者在文件定义块(FDB)的文件存取鉴别方式(Acceptable Authentication Type)被设定为不需要公开密钥鉴别的情况下，接着，在步骤 S905 中，参照设备的存储器内的对应分区的文件定义块(FDB)(参照图 24)中记录的文件存取鉴别方式(Acceptable Authentication Type: 在进行特定的文件(File)存取时指定的鉴别方式)，来确认执行存取待存取文件的命令所需的验证级别(是否需要公开密钥验证)。

在步骤 S905 中，在文件定义块(FDB)的文件存取验证方式(Acceptable Verification Type)被设定为需要公开密钥体制的权证验证的情况下，在步骤 S906 中，判定存取命令所需的验证级别的验证---公开密钥体制的权证验证是否已完毕，在验证未完的情况下，不根据命令来执行处理，将存取出错发送到文件存取装置(S908)。

在步骤 S905 中，在文件定义块(FDB)的文件存取验证方式(Acceptable Verification Type)被设定为需要公开密钥体制的权证验证、在步骤 S906 中判定为公开密钥体制的权证验证已完毕的情况下，或者在文件定义块(FDB)的文件存取验证方式(Acceptable Verification Type)被设定为不需要公开密钥体制的权证验证的情况下，在步骤 S907 中，执行从文件存取装置接收到的存取命令的处理，将结果发送到文件存取装置。

接收到存取命令结果(S892)的文件存取装置进而判定是否执行其他文件存取(S893)，在执行其他文件存取的情况下，重复执行步骤 S891 以下的处理；而在不执行其他文件存取的情况下结束处理。

接着，用图 87 来说明对多个文件进行存取的情况下的处理(Combination)。在图 87 的流程中，左侧示出文件存取装置，右侧示出设备(参照图 5)。其中，在文件存取处理中，在文件存取装置和设备间传送数据时，用相互鉴别处理中取得的会话密钥 Kses、或由多个会话密钥进行运算或选择所得的会话密钥来进行加密，并且执行窜改检

查用的MAC的生成、验证处理。

文件存取装置在步骤S911中将存取命令发送到设备。该命令是指定待存取文件ID(源)、目标文件ID、存取模式的命令，例如是先前用图84说明过的、指定用目标文件ID[0x0001]的密钥对源文件ID[0x0002]执行内部加密处理的命令[Internal Encryption Command(0x0001，0x0002)]等。

设备接收到来自文件存取装置的命令(S921)后，判定在文件打开表的目标文件ID的项中是否有存取命令的许可(S922)。在文件打开表的目标文件ID的项中不存在存取命令的许可的情况下，不根据命令来执行处理，将存取出错发送到文件存取装置(S934)。

而在文件打开表的目标文件ID的项中存在存取命令的许可的情况下，在步骤S923中，参照设备的存储器内的对应分区的文件定义块(FDB)(参照图24)中记录的文件存取鉴别方式(Acceptable Authentication Type：在进行特定的文件(File)存取时指定的鉴别方式)，来确认执行存取待存取目标文件的命令所需的鉴别级别(是否需要公开密钥鉴别)。

在步骤S923中，在为待存取目标文件设定的文件定义块(FDB)的文件存取鉴别方式(Acceptable Authentication Type)被设定为需要公开密钥鉴别的情况下，在步骤S924中，判定存取命令所需的鉴别级别的鉴别---公开密钥鉴别是否已完毕，在鉴别未完的情况下，不根据命令来执行处理，将存取出错发送到文件存取装置(S934)。鉴别结束或未完根据相互鉴别时设定的鉴别表(参照图51)来判定。

在步骤S923中，在为待存取目标文件设定的文件定义块(FDB)的文件存取鉴别方式(Acceptable Authentication Type)被设定为需要公开密钥鉴别、在步骤S924中判定为公开密钥鉴别已完毕的情况下，或者在文件定义块(FDB)的文件存取鉴别方式(Acceptable Authentication Type)被设定为不需要公开密钥鉴别的情况下，接着，在步骤S925中，参照设备的存储器内的对应分区的文件定义块(FDB)(参照图24)中记录的文件存取鉴别方式(Acceptable Authentication Type：在进行特定的文件(File)存取时指定的鉴别方式)，来确认执行存取待存取目标文件的命令所需的验证级别(是否需要公开密钥验证)。

在步骤 S925 中，在为待存取目标文件设定的文件定义块(FDB)的文件存取验证方式(Acceptable Verification Type)被设定为需要公开密钥体制的权证验证的情况下，在步骤 S926 中，判定存取命令所需的验证级别的验证---公开密钥体制的权证验证是否已完毕，在验证未完的情况下，不根据命令来执行处理，将存取出错发送到文件存取装置(S934)。

在步骤 S925 中，在为待存取目标文件设定的文件定义块(FDB)的文件存取验证方式(Acceptable Verification Type)被设定为需要公开密钥体制的权证验证、在步骤 S926 中判定为公开密钥体制的权证验证已完毕的情况下，或者在文件定义块(FDB)的文件存取验证方式(Acceptable Verification Type)被设定为不需要公开密钥体制的权证验证的情况下，接着，在步骤 S927 中，根据命令来确认存取命令中包含的目标文件 ID 所指定的文件的存取方法(Read/Write)。

设备判定来自文件存取装置的命令中包含的源文件 ID 所指定的文件对存取命令中包含的存取方法(Read/Write)是否已打开(S928)。在文件打开表中不存在用于执行命令的存取方法(Read/Write)的情况下，不根据命令来执行处理，将存取出错发送到文件存取装置(S934)。

在文件打开表中存在与命令对应的存取方法(Read/Write)的情况下，在步骤 S929 中，参照设备的存储器内的对应分区的文件定义块(FDB)(参照图 24)中记录的文件存取鉴别方式(Acceptable Authentication Type: 在进行特定的文件(File)存取时指定的鉴别方式)，来确认执行存取待存取源文件的命令所需的鉴别级别(是否需要公开密钥鉴别)。

在步骤 S929 中，在为待存取源文件设定的文件定义块(FDB)的文件存取鉴别方式(Acceptable Authentication Type)被设定为需要公开密钥鉴别的情况下，在步骤 S930 中，判定存取命令所需的鉴别级别的鉴别---公开密钥鉴别是否已完毕，在鉴别未完的情况下，不根据命令来执行处理，将存取出错发送到文件存取装置(S934)。鉴别结束或未完根据相互鉴别时设定的鉴别表(参照图 51)来判定。

在步骤 S929 中，在为待存取源文件设定的文件定义块(FDB)的文件存取鉴别方式(Acceptable Authentication Type)被设定为需要公开密钥鉴别、在步骤 S930 中判定为公开密钥鉴别已完毕的情况下，或

者在文件定义块(FDB)的文件存取鉴别方式(Acceptable Authentication Type)被设定为不需要公开密钥鉴别的情况下，接着，在步骤 S931 中，参照设备的存储器内的对应分区的文件定义块(FDB)(参照图 24)中记录的文件存取鉴别方式(Acceptable Authentication Type: 在进行特定的文件(File)存取时指定的鉴别方式)，来确认执行存取待存取源文件的命令所需的验证级别(是否需要公开密钥验证)。

在步骤 S931 中，在为待存取源文件设定的文件定义块(FDB)的文件存取验证方式(Acceptable Verification Type)被设定为需要公开密钥体制的权证验证的情况下，在步骤 S932 中，判定存取命令所需的验证级别的验证---公开密钥体制的权证验证是否已完毕，在验证未完的情况下，不根据命令来执行处理，将存取出错发送到文件存取装置(S934)。

在步骤 S931 中，在为待存取源文件设定的文件定义块(FDB)的文件存取验证方式(Acceptable Verification Type)被设定为需要公开密钥体制的权证验证、在步骤 S932 中判定为公开密钥体制的权证验证已完毕的情况下，或者在文件定义块(FDB)的文件存取验证方式(Acceptable Verification Type)被设定为不需要公开密钥体制的权证验证的情况下，在步骤 S933 中，执行从文件存取装置接收到的存取命令的处理，将结果发送到文件存取装置。

接收到存取命令结果(S912)的文件存取装置进而判定是否执行其他文件存取(S913)，在执行其他文件存取的情况下，重复执行步骤 S911 以下的处理；而在不执行其他文件存取的情况下结束处理。

假定在文件内保存有由某 1 个文件结构所指定的数据的情况而说明了上述文件存取处理，但是也可以将不同文件结构数据保存在 1 个文件内，通过对 1 个文件的 1 个命令，来执行与对上述多个文件的顺序处理同样的处理。

图 88 示出通过对 1 个文件的 1 个命令来对 1 个文件内的数据执行顺序处理的结构的说明图。

如图所示，文件是电子货币文件，由作为金额数据的[Purse]、作为利用日志数据的“Log”、作为对数据的加密或解密用的密钥数据的[Key]构成。

例如，如图 88(a)所示，规定存入命令(Deposit Command)，可以执行下述 2 个处理：将文件内的作为金额数据的[Purse]加上 X 日元(S941)，进而，向文件内的作为利用日志数据的“Log”中写入将[Purse]加上了 X 日元的记录(S942)。

作为与先前说明过的文件存取模式(参照图 29)的存款类对应的容许命令(参照图 30)，定义上述存入命令(Deposit Command)，在存取许可权证的文件存取模式(File Access Mode)中设定[存款类]，作为文件 ID(File ID)，生成指定构成电子货币的复合文件的存取许可权证(SPT)，从文件存取装置发送到设备后，通过与存入命令(Deposit Command)一起来发送存入金额数据，可如图 88(a)所示在设备中对 1 个文件内的数据执行顺序处理。

此外，如图 88(b)所示，规定收据生成命令[Make Receipt Command]，可以执行下述 3 个步骤的处理：将文件内的作为金额数据的[Purse]减去 X 日元(S945)，进而，向文件内的作为利用日志数据的“Log”中写入将[Purse]减去了 X 日元的记录(S946)，进而对“Log”应用作为数据的加密密钥数据的[Key]来附加签名并发送。

在此情况下，作为与文件存取模式(参照图 29)的取款类对应的容许命令(参照图 30)，定义上述收据生成命令[Make Receipt Command]，在存取许可权证的文件存取模式(File Access Mode)中设定[取款类]，作为文件 ID(File ID)，生成指定构成电子货币的复合文件的存取许可权证(SPT)，从文件存取装置发送到设备后，通过与收据生成命令[Make Receipt Command]一起来发送取出金额数据，可如图 88(b)所示在设备中对 1 个文件内的数据执行顺序处理。

这样，在服务许可权证(SPT)所指定的处理文件是复合文件的情况下，设备从复合文件内选择来自存取机器的接收命令的待处理文件并执行处理。在来自存取机器的数据处理命令是包含一系列多个处理的序列处理命令的情况下，设备从服务许可权证(SPT)所指定的复合文件内依次选择序列处理命令中包含的各命令的待处理文件并执行。

[B4. 7. 利用服务许可权证(SPT)的存取处理各方式中的处理过程]

在上述利用服务许可权证(SPT)的存取处理文件的设定注册处理中，在分区管理器管理的服务许可权证(SPT)用户---作为设备存取机

器的读写器和设备间执行相互鉴别，根据服务许可权证(SPT)来进行文件存取。相互鉴别处理的形态是公开密钥相互鉴别、对称密钥相互鉴别这2种中的某一种，并且权证(SPT)验证处理也执行公开密钥类签名验证、对称密钥类MAC验证这2种中的某一种。即处理形态大体分为下述4种形态。

(A)相互鉴别(公开密钥)，权证(SPT)验证(公开密钥)

(B)相互鉴别(公开密钥)，权证(SPT)验证(对称密钥)

(C)相互鉴别(对称密钥)，权证(SPT)验证(对称密钥)

(D)相互鉴别(对称密钥)，权证(SPT)验证(公开密钥)。

以认证机构(CA(PM))、分区管理器(PM)、SPT权证用户---作为设备存取机器的读写器、设备、各实体间执行的数据传送处理为中心，用附图来简洁说明这4种形态的处理。

(A)相互鉴别(公开密钥)，权证(SPT)验证(公开密钥)

首先，用图89来说明在相互鉴别处理中应用公开密钥体制、在权证(SPT)验证中应用公开密钥体制的情况下各实体间的数据传送。

按图示的号码顺序在各实体间执行数据传送。以下，根据各号码来说明处理。

(1)发行分区管理器(PM)的公开密钥证书(Cert. PM)

分区管理器(PM)的公开密钥证书(Cert. PM)根据来自分区管理器(PM)的发行请求通过经注册机构(RA)的证书发行过程从分区认证机构(CA(PAR))发行。其中，本结构是分区管理器兼作服务许可权证发行部件(SPT Issuer)的结构，是服务许可权证发行部件(SPT Issuer)的公开密钥证书使用分区管理器(PM)的公开密钥证书的结构。

(2)发行服务许可权证用户(SPT User)---作为设备存取机器的读写器(R/W)的公开密钥证书(Cert. RW)

服务许可权证用户(SPT User：具体地说，是向设备发送权证的作为设备存取机器的读写器(R/W))的公开密钥证书(Cert. R/W)根据来自服务许可权证用户(SPT User)---读写器(R/W)的发行请求通过经注册机构(RA)的证书发行过程由分区认证机构(CA(PAR))发行。其中，分区管理器也可兼作服务许可权证用户(SPT User)，在此情况下，服务许可权证用户(SPT User)的公开密钥证书可使用分区管理器(PM)的公开密钥证书。

(3)服务许可权证(SPT)生成处理

服务许可权证(SPT)由分区管理器管理的服务许可权证发行部件(SPT Ticket Issuer)生成。在此情况下，执行公开密钥体制的签名生成、验证，所以用服务许可权证发行部件(SPT Ticket Issuer)的私有密钥来生成签名(Signature)(参照图 12)并附加到 SPT 上。

(4)将 SPT 及作为服务许可权证发行部件(SPT Ticket Issuer)的分区管理器的公开密钥证书(Cert. PM)提供给服务许可权证用户(SPT User)---作为设备存取机器的读写器(R/W)

将分区管理器管理的服务许可权证发行部件(SPT Ticket Issuer)发行的服务许可权证(SPT)与作为服务许可权证发行部件(SPT Ticket Issuer)的分区管理器的公开密钥证书(Cert. PM)一起发送到服务许可权证用户(SPT User) ---作为设备存取机器的读写器(R/W)。

(5)作为设备存取机器的读写器(R/W)和设备间的相互鉴别

服务许可权证用户(SPT User)---读写器向想要根据服务许可权证发行部件(SPT Ticket Issuer)发行的服务许可权证(SPT)来执行文件存取的设备发送作为权证用户(SPT User)的读写器(R/W)的公开密钥证书(Cert. RW)，执行公开密钥体制的相互鉴别(参照图 50)。

(6)将 SPT 及作为服务许可权证发行部件(SPT Ticket Issuer)的分区管理器的公开密钥证书(Cert. PM)提供给设备

作为设备存取机器的读写器(R/W)和设备间的相互鉴别成立后，作为权证用户(SPT User)的读写器(R/W)向设备发送服务许可权证(SPT)、及作为服务许可权证发行部件(SPT Ticket Issuer)的分区管理器的公开密钥证书(Cert. PM)。

设备对接收到的服务许可权证(SPT)执行下述确认：(1)作为权证发行者(Ticket Issuer)的分区管理器的公开密钥证书(Cert. PM)是未窜改过的合法的公开密钥证书(CERT)；(2)权证发行者(Ticket Issuer)的公开密钥证书(CERT PM)的选项区域中记录的代码、和设备内的 FDB (File Definition Block)中记录的(SPTIC)一致；(3)权证发行部件(Ticket Issuer)未作废；(4)通过验证所接收权证(SPT)的签名(Signature)而确认权证没有窜改，进而，确认 SPT 权证中保存的 SPT 用户(作为权证用户的读写器)和作为权证用户(SPT User)的公开密钥证书(Cert. RW)的标识数据(DN)而记录的标识符、范畴或序列号

(SN)一致，通过确认相互鉴别已完毕来执行SPT用户(作为设备存取机器的读写器)的验证(参照图57、图58)。

(7)文件存取

设备根据服务许可权证(SPT)上记述的规则来存取待处理文件。

通过以上处理，根据相互鉴别(公开密钥)、权证(SPT)验证(公开密钥)各方式来执行文件存取处理。

(B)相互鉴别(公开密钥)，权证(SPT)验证(对称密钥)

接着，用图90来说明在相互鉴别处理中应用公开密钥体制、在权证(SPT)验证中应用对称密钥体制的情况下各实体间的数据传送。

按图示的号码顺序在各实体间执行数据传送。以下，根据各号码来说明处理。

(1)发行服务许可权证用户(SPT User)---作为设备存取机器的读写器(R/W)的公开密钥证书(Cert. RW)

服务许可权证用户(SPT User：具体地说，是向设备发送权证的作为设备存取机器的读写器(R/W))的公开密钥证书(Cert. R/W)根据来自服务许可权证用户(SPT User)---读写器(R/W)的发行请求通过经注册机构(RA)的证书发行过程由分区认证机构(CA(PAR))发行。其中，分区管理器也可兼作服务许可权证用户(SPT User)，在此情况下，服务许可权证用户(SPT User)的公开密钥证书可使用分区管理器(PM)的公开密钥证书。

(2)服务许可权证(SPT)生成处理

服务许可权证(SPT)由分区管理器管理的服务许可权证发行部件(SPT Ticket Issuer)生成。在此情况下，将MAC(Message Authentication Code)(参照图59)作为对称密钥体制的验证值附加到SPT上。

(3)将SPT提供给服务许可权证用户(SPT User)---作为设备存取机器的读写器(R/W)

将分区管理器管理的服务许可权证发行部件(SPT Ticket Issuer)发行的服务许可权证(SPT)发送到作为服务许可权证用户(SPT User)的读写器(R/W)。

(4)读写器(R/W)和设备间的相互鉴别

服务许可权证用户(SPT User)---作为设备存取机器的读写器向

想要根据服务许可权证发行部件(SPT Ticket Issuer)发行的服务许可权证(SPT)来执行文件存取的设备发送作为权证用户(SPT User)的读写器(R/W)的公开密钥证书(Cert. RW)，执行公开密钥体制的相互鉴别(参照图 50)。

(5)将 SPT 提供给设备

作为设备存取机器的读写器(R/W)和设备间的相互鉴别成立后，服务许可权证用户(SPT User)---读写器(R/W)向设备发送服务许可权证(SPT)。设备对接收到的服务许可权证(SPT)执行 MAC 验证处理，执行 SPT 发行者(SPT Issuer)的验证，进而确认 SPT 权证中保存的 SPT 用户(作为权证用户的读写器)和作为权证用户(SPT User)的公开密钥证书(Cert. RW)的标识数据(DN)而记录的标识符、范畴或序列号(SN)一致，通过确认相互鉴别已完毕来执行 SPT 用户(作为设备存取机器的读写器)的验证(参照图 57、图 58)。

(6)文件存取

设备根据服务许可权证(SPT)上记述的规则来存取待处理文件。

通过以上处理，根据相互鉴别(公开密钥)、权证(SPT)验证(对称密钥)各方式来执行文件存取处理。

(C)相互鉴别(对称密钥)，权证(SPT)验证(对称密钥)

接着，用图 91 来说明在相互鉴别处理中应用对称密钥体制、在权证(SPT)验证中应用对称密钥体制的情况下各实体间的数据传送。

按图示的号码顺序在各实体间执行数据传送。以下，根据各号码来说明处理。

(1)服务许可权证(SPT)生成处理

服务许可权证(SPT)由分区管理器管理的服务许可权证发行部件(SPT Ticket Issuer)生成。在此情况下，将 MAC(Message Authentication Code)(参照图 59)作为对称密钥体制的验证值附加到 SPT 上。

(2)将 SPT 提供给服务许可权证用户(SPT User)

将分区管理器管理的服务许可权证发行部件(SPT Ticket Issuer)发行的服务许可权证(SPT)发送到服务许可权证用户(SPT User)---作为设备存取机器的读写器。

(3)作为设备存取机器的读写器(R/W)和设备间的相互鉴别

服务许可权证用户(SPT User)---作为设备存取机器的读写器(R/W)与想要根据服务许可权证发行部件(SPT Ticket Issuer)发行的服务许可权证(SPT)来执行文件存取的设备执行对称密钥体制的相互鉴别(参照图 53、图 54)。

(4)将 SPT 提供给设备

作为设备存取机器的读写器(R/W)和设备间的相互鉴别成立后，服务许可权证用户(SPT User)---读写器向设备发送服务许可权证(SPT)。设备对接收到的服务许可权证(SPT)执行 MAC 验证处理，执行 SPT 发行者(SPT Issuer)的验证，进而确认 SPT 权证中保存的 SPT 用户(作为权证用户的读写器)和权证用户(SPT User)的标识符一致，通过确认相互鉴别已完毕来执行 SPT 用户(作为设备存取机器的读写器)的验证(参照图 57、图 58)。

(5)文件存取

设备根据服务许可权证(SPT)上记述的规则来存取待处理文件。

通过以上处理，根据相互鉴别(对称密钥)、权证(SPT)验证(对称密钥)各方式来执行文件存取处理。

(D)相互鉴别(对称密钥)，权证(SPT)验证(公开密钥)。

接着，用图 92 来说明在相互鉴别处理中应用对称密钥体制、在权证(SPT)验证中应用公开密钥体制的情况下各实体间的数据传送。

按图示的号码顺序在各实体间执行数据传送。以下，根据各号码来说明处理。

(1)发行分区管理器(PM)的公开密钥证书(Cert. PM)

分区管理器(PM)的公开密钥证书(Cert. PM)根据来自分区管理器(PM)的发行请求通过经注册机构(RA)的证书发行过程从分区认证机构(CA(PAR))发行。其中，本结构是分区管理器兼作服务许可权证发行部件(SPT Issuer)的结构，是服务许可权证发行部件(SPT Issuer)的公开密钥证书使用分区管理器(PM)的公开密钥证书的结构。

(2)服务许可权证(SPT)生成处理

服务许可权证(SPT)由分区管理器管理的服务许可权证发行部件(SPT Ticket Issuer)生成。在此情况下，执行公开密钥体制的签名生成、验证，所以用服务许可权证发行部件(SPT Ticket Issuer)的私有密钥来生成签名(Signature)(参照图 12)并附加到 SPT 上。

(3)将SPT及作为服务许可权证发行部件(SPT Ticket Issuer)的分区管理器的公开密钥证书(Cert. PM)提供给服务许可权证用户(SPT User)---作为设备存取机器的读写器(R/W)

将分区管理器管理的服务许可权证发行部件(SPT Ticket Issuer)发行的服务许可权证(SPT)与作为服务许可权证发行部件(SPT Ticket Issuer)的分区管理器的公开密钥证书(Cert. PM)一起发送到服务许可权证用户(SPT User)、即向设备发送权证的机器(例如,作为设备存取机器的读写器).

(4)作为设备存取机器的读写器(R/W)和设备间的相互鉴别

服务许可权证用户(SPT User)---作为设备存取机器的读写器与想要根据服务许可权证发行部件(SPT Ticket Issuer)发行的服务许可权证(SPT)来执行文件存取的设备执行对称密钥体制的相互鉴别(参照图53、图54).

(5)将SPT及作为服务许可权证发行部件(SPT Ticket Issuer)的分区管理器的公开密钥证书(Cert. PM)提供给设备

读写器(R/W)和设备间的相互鉴别成立后,服务许可权证用户(SPT User)---作为设备存取机器的读写器向设备发送服务许可权证(SPT)、及作为服务许可权证发行部件(SPT Ticket Issuer)的分区管理器的公开密钥证书(Cert. PM).

设备对接收到的服务许可权证(SPT)执行下述确认:(1)作为权证发行者(Ticket Issuer)的分区管理器的公开密钥证书(Cert. PM)是未窜改过的合法的公开密钥证书(CERT);(2)作为权证发行者(Ticket Issuer)的分区管理器的公开密钥证书(Cert. PM)的选项区域中记录的代码、和设备内的FDB (File Definition Block)中记录的权证发行部件代码(SPTIC)一致;(3)权证发行部件(Ticket Issuer)未作废;(4)通过验证所接收权证(SPT)的签名(Signature)而确认权证没有窜改,进而,确认SPT权证中保存的SPT用户(作为权证用户的读写器)和权证用户(SPT User)的标识符一致,通过确认相互鉴别已完毕来执行SPT用户(读写器)的验证(参照图57、图58).

(6)文件存取

设备根据服务许可权证(SPT)上记述的规则来存取待处理文件.

通过以上处理,根据相互鉴别(对称密钥)、权证(SPT)验证(公开

密钥)各方式来执行文件存取处理。

[B5. 利用数据更新权证(DUT)的设备数据更新处理]

接着，说明利用数据更新权证(DUT: Data Update Ticket)的设备数据更新处理。数据更新权证(DUT: Data Update Ticket)是执行设备中保存的各种数据的更新处理时应用的存取控制权证。权证用户(例如，作为设备存取机器的读写器)通过使用合法的数据更新权证(DUT)发行部件(Ticket Issuer)发行的 DUT，根据 DUT 上记录的过程来存取设备，能够执行 DUT 上记录的限制内的数据处理。

其中，如前所述，数据更新权证(DUT: Data Update Ticket)有为了执行设备管理器管理的数据项目的更新处理而应用的权证 DUT(DEV)、和为了执行分区管理器管理的分区内的数据项目的更新处理而应用的权证 DUT(PAR)(参照图 32)。

下面说明应用数据更新权证(DUT)对设备中保存的数据执行数据更新的处理。参照图 93 以下的流程等附图来进行说明。其中，在数据更新处理中，包含设备和执行数据更新的作为设备存取机器的读写器间的相互鉴别处理(设备鉴别或分区鉴别)、数据更新权证(DUT: Data Update Ticket)的完整性验证处理。

下面说明图 93 所示的数据更新处理流程。在图 93 中，左侧示出数据更新装置的处理，右侧示出设备(参照图 5)的处理。其中，数据更新装置是可对设备进行数据读取写入处理的装置(例如，作为设备存取机器的读写器、PC)，相当于图 10 的作为设备存取机器的读写器。首先，用图 93 来概要说明数据更新处理，其后，用图 94 的流程来详细说明该处理中包含的数据更新操作。

首先，在图 93 的步骤 S951 和 S960 中，执行数据更新装置和设备间的相互鉴别处理。在执行数据发送接收的 2 个部件间，相互确认对方是否是合法的数据通信者，其后进行所需的数据传送。确认对方是否是合法的数据通信者的处理是相互鉴别处理。在相互鉴别处理时执行会话密钥的生成，将生成的会话密钥作为对称密钥来执行加密处理并进行数据发送。

相互鉴别处理是与先前的分区创建、删除处理一栏中说明过的同样的处理，执行设备鉴别或分区鉴别中的某一种。对它们分别应用对称密钥体制鉴别、或公开密钥体制鉴别处理中的某一种。该相互鉴别

处理是与前述用图 48～图 56 说明过的同样的处理，所以省略其说明。

其中，作为相互鉴别处理应执行的处理由应用的数据更新权证(DUT)(参照图 32)的

* Authentication Type: 设备(Device)的相互鉴别类型(公开密钥鉴别、对称密钥鉴别、或任一种皆可(Any))

来决定。

在鉴别处理失败的情况下(S952、S961 中为“否(No)”)，表示不能确认相互是合法的机器、设备，不执行以下的处理，作为出错而结束处理。

如果鉴别处理成功，则数据更新装置向设备发送数据更新权证(DUT: Data Update Ticket)。数据更新权证(DUT)是由设备管理器或分区管理器管理下的数据更新权证(DUT)发行部件(DUT Issuer)发行的权证。数据更新权证(DUT)是对设备的存取控制权证，是具有先前说明过的图 32 的数据格式结构的权证。

其中，在将数据更新权证(DUT)向权证用户发送时，在公开密钥体制的情况下，数据更新权证(DUT)发行部件(DUT Issuer)的公开密钥证书(CERT_DUTI)也一起发送。DUT 发行部件的公开密钥证书(CERT_DUTI)的属性(Attribute)与设备内的 DKDB(PUB)(Device Key Definition Block)中记录的权证发行部件代码(DUTIC_DEV)或 PKDB(PUB)(Partition Key Definition block)中记录的权证发行部件代码(DUTIC_PAR)的标识符(DUTIC)一致。

接收到数据更新权证(DUT)(S962)的设备执行接收到的权证(DUT)的完整性和用户检查处理(S963)。权证完整性验证处理应用基于对称密钥体制的 MAC 验证、或基于公开密钥体制的签名验证处理中的某一种来执行。用户检查是检查发送来权证的机器(权证用户)的完整性的处理，在相互鉴别已成立时，作为验证对方鉴别者的标识数据、和权证中记录的权证用户标识符(参照图 32)是否一致等的处理来执行。这些处理是与先前的分区注册权证(PRT)应用处理的说明中用图 57～图 59 说明过的同样的处理，所以省略其说明。

在设备中，在所接收权证(DUT)的完整性和用户检查处理的结果是未能确认权证及用户合法的情况下(S964 中为“否”)，将数据更新权证(DUT)受理出错通知给数据更新装置(S968)。在得以确认权证及用户

合法的情况下(S964 中为“是(Yes)”)，根据接收到的数据更新权证(DUT)上记述的规则来执行设备内的存储部中的数据(参照图 33)的更新处理。后面将用别的流程来详述该处理。

根据数据更新权证(DUT)的记述，数据更新处理成功(S966 中为“是”)后，将 DUT 受理成功通知给数据更新装置(S967)。而在数据更新处理失败(S966 中为“否”)的情况下，将 DUT 受理出错通知给数据更新装置(S968)。

数据更新装置接收 DUT 受理结果(S954)，判定 DUT 处理结果，在 DUT 受理结果是出错的情况下(S955 中为“否”)，作为出错而结束处理；而在 DUT 受理结果是成功(S955 中为“是”)的情况下，执行会话清除命令的发送接收(S956、S969)，抛弃设备一侧生成的鉴别表(S970)，结束处理。鉴别表是在步骤 S951、S960 的相互鉴别处理中生成的表，与前述的分区注册权证(PRT)应用处理的项目中说明过的结构、即图 51 的结构相同。

这样利用数据更新权证(DUT)，执行设备内保存的数据的更新处理。以下，用图 94 来说明该处理中包含的数据更新操作(S965)。

图 94 的处理流程是在接收到数据更新权证(DUT)的设备中执行的处理，在与发送数据更新权证(DUT)的机器的相互鉴别成立、权证的验证也成功以后执行。

首先，在步骤 S971 中，设备由数据更新权证(DUT)的被更新的旧数据的代码(Old Data Code)来搜索待更新数据的版本。例如如果待更新的是设备管理器代码(DMC)，则在设备管理信息块(参照图 15)中记录有版本，而如果待更新的是分区管理器代码(PMC)，则在分区管理信息块(参照图 20)中记录有版本。此外，分区注册权证(PRT)发行部件(PRT Issuer)的版本被包含在设备定义块(参照图 16)中。再者，在作废表(IRL DEV、CRL DEV)等中包含版本信息。这样按照信息决定了版本信息的保存地址，设备由被更新的旧数据的代码(Old Data Code)来搜索待更新数据的版本。

接着，设备在步骤 S972 中参照数据更新权证(DUT)上记录的进行数据更新时的版本条件[Data Version Rule]，判定设定是否是[Any]。

如前所述，进行数据更新时的版本条件[Data Version Rule]有 3

种：Any、Exact、Older。Any 可与版本(Version)条件无关地进行数据更新；Exact 在与后续[Data Version Condition]指定的值相同的情况下可进行数据更新；Older 只在 New Data Version 更加新的情况下才可进行数据更新。其中，在版本条件[Data Version Rule]为 Any 或 Older 的情况下，不使用或忽略[Data Version Condition]。

在数据更新权证(DUT)的[Data Version Rule]的设定不是[Any]的情况下，根据版本条件[Data Version Rule]来执行处理。该步骤是 S973～S975。

在步骤 S973 中，参照数据更新权证(DUT)的[Data Version Rule]，来判定设定是否是[EXACT]。[EXACT]表示在与[Data Version Condition]指定的值相同的情况下可进行数据更新。在设定是[EXACT]的情况下，在步骤 S974 中，判定待更新数据[Old Data]的版本与数据更新权证(DUT)的[Data Version Condition]中记录的版本值是否一致。只在一致的情况下进至下一步骤；而在不一致的情况下，不执行更新处理，作为出错结束。

在步骤 S973 中判定为数据更新权证(DUT)的[Data Version Rule]不是[EXACT]的情况下，设定是[Older]。[Older]的设定是下述设定：只在数据更新权证(DUT)的表示新数据[New Data]的版本的[New Data Version]比待更新数据[Old Data]的版本新的情况下才进行更新。在该[Older]的设定的情况下，在步骤 S975 中，判定数据更新权证(DUT)的表示新数据[New Data]的版本的[New Data Version]是否比待更新数据[Old Data]的版本新，只在新的情况下进至下一步骤；而在不一致的情况下，不执行更新处理，作为出错结束。

接着，在步骤 S976 中，验证数据更新权证(DUT)的[Encryption Flag]。[Encryption Flag]是表示被更新的数据是否被加密过(加密过：Encrypted/未加密：none)的数据。在[Encryption Flag]表示待更新数据是未加密数据的情况下，在步骤 S977 中，将数据更新权证(DUT)的新数据[New Data]置换为设备的存储部中保存的待更新旧数据[Old Data]，处理结束。其中，在为待更新数据附加了版本的情况下，将数据更新权证(DUT：Data Update Ticket)中保存的要更新的数据的版本(New Data Version)保存到与设备内的更新数据对应而设定的版本保存区域中。

此外，在步骤S976中，在判定为[Encryption Flag]表示被更新的数据被加密过(加密过：Encrypted)的情况下，在步骤S978中，验证数据更新权证(DUT)的[Ticket Type]。[Ticket Type]是表示权证(Ticket)的类型(DUT(DEV)/DUT(PAR))的数据。DUT(DEV)表示是执行设备管理器管理的数据项目的更新处理的权证，而DUT(PAR)表示是为了执行分区管理器管理的分区内的数据项目的更新处理而应用的权证。

在权证类型[Ticket Type]表示DUT(DEV)的情况下，执行步骤S979～S982；而在表示DUT(PAR)的情况下，执行步骤S983～S986。

在权证类型[Ticket Type]表示DUT(DEV)的情况下，在步骤S979中，判定数据更新权证(DUT(DEV))上记述的Old Data Code(被更新的旧数据的代码)所示的数据是否是设备密钥区域(参照图 18)中保存的Kdut_DEV1(数据更新权证(DUT)的MAC验证密钥)、或Kdut_DEV2(数据更新加密密钥)。

在判定为数据更新权证(DUT(DEV))上记述的Old Data Code(被更新的旧数据的代码)所示的数据是设备密钥区域(参照图 18)中保存的Kdut_DEV1(数据更新权证(DUT)的MAC验证密钥)、Kdut_DEV2(数据更新加密密钥)的情况下，在步骤S980中，用设备的设备密钥区域(参照图 18)中保存的Kdut_DEV4(数据更新加密密钥)对数据更新权证(DUT(DEV))上保存的作为新数据[New Data]的Kdut_DEV1、Kdut_DEV2进行解密，盖写到设备的设备密钥区域中保存的Kdut_DEV1、Kdut_DEV2上。其中，一并将数据更新权证(DUT(DEV))上保存的要更新的数据的版本(New Data Version)保存到与设备内的更新数据对应而设定的版本保存区域、在此情况下是设备的设备密钥区域(参照图18)中。

接着，在步骤S981中，交换设备的设备密钥区域(参照图 18)中保存的Kdut_DEV1(数据更新权证(DUT)的MAC验证密钥)、和Kdut_DEV3(数据更新权证(DUT)的MAC验证密钥)，并且交换Kdut_DEV2(数据更新加密密钥)、和Kdut_DEV4(数据更新加密密钥)并结束处理。

其中，通过该交换Kdut_DEV1和Kdut_DEV3、及交换Kdut_DEV2和Kdut_DEV4，来始终维持Kdut_DEV3(数据更新权证(DUT)的MAC验

证密钥)、Kdut_DEV4(数据更新加密密钥)这一对儿比 Kdut_DEV1(数据更新权证(DUT)的 MAC 验证密钥)、Kdut_DEV2(数据更新加密密钥)这一对儿的版本新，可将改写对象始终设定为 Kdut_DEV1、Kdut_DEV2。

其中，在步骤 S979 中，在判定为数据更新权证(DUT(DEV))上记述的 Old Data Code(被更新的旧数据的代码)所示的数据不是设备密钥区域(参照图 18)中保存的 Kdut_DEV1(数据更新权证(DUT)的 MAC 验证密钥)、Kdut_DEV2(数据更新加密密钥)的情况下，在步骤 S982 中，用设备的设备密钥区域(参照图 18)中保存的 Kdut_DEV2(数据更新加密密钥)对数据更新权证(DUT(DEV))上保存的新数据[New Data]进行解密，盖写到数据更新权证(DUT(DEV))的 Old Data Code(被更新的旧数据的代码)所示的区域上。其中，在为待更新数据附加了版本的情况下，将数据更新权证(DUT(DEV))中保存的要更新的数据的版本(New Data Version)保存到与设备内的更新数据对应而设定的版本保存区域中。

另一方面，在步骤 S978 中，在权证类型[Ticket Type]表示 DUT(PAR)的情况下，执行步骤 S983～S986。

在权证类型[Ticket Type]表示 DUT(PAR)的情况下，在步骤 S983 中，判定数据更新权证(DUT(PAR))上记述的 Old Data Code(被更新的旧数据的代码)所示的数据是否是分区密钥区域(参照图 23)中保存的 Kdut_PAR1(数据更新权证(DUT)的 MAC 验证密钥)、或 Kdut_PAR2(数据更新加密密钥)。

在判定为数据更新权证(DUT(PAR))上记述的 Old Data Code(被更新的旧数据的代码)所示的数据是分区密钥区域(参照图 23)中保存的 Kdut_PAR1(数据更新权证(DUT)的 MAC 验证密钥)、Kdut_PAR2(数据更新加密密钥)的情况下，在步骤 S984 中，用设备的分区密钥区域(参照图 23)中保存的 Kdut_PAR4(数据更新加密密钥)对数据更新权证(DUT(PAR))上保存的作为新数据[New Data]的 Kdut_PAR1、Kdut_PAR2 进行解密，盖写到设备的分区密钥区域中保存的 Kdut_PAR1、Kdut_PAR2 上。其中，一并将数据更新权证(DUT(PAR))上保存的要更新的数据的版本(New Data Version)保存到与设备内的更新数据对应而设定的版本保存区域、在此情况下是设备的分区密钥区域(参照图 23)中。

接着，在步骤 S985 中，交换设备的分区密钥区域(参照图 23)中保存的 Kdut_PAR1(数据更新权证(DUT)的 MAC 验证密钥)、和 Kdut_PAR3(数据更新权证(DUT)的 MAC 验证密钥)，并且交换 Kdut_PAR2(数据更新加密密钥)、和 Kdut_PAR4(数据更新加密密钥)并结束处理。

其中，通过该交换 Kdut_PAR1 和 Kdut_PAR3、及交换 Kdut_PAR2 和 Kdut_PAR4，来始终维持 Kdut_PAR3(数据更新权证(DUT)的 MAC 验证密钥)、Kdut_PAR4(数据更新加密密钥)这一对儿比 Kdut_PAR1(数据更新权证(DUT)的 MAC 验证密钥)、Kdut_PAR2(数据更新加密密钥)这一对儿的版本新，可将改写对象始终设定为 Kdut_PAR1、Kdut_PAR2。

其中，在步骤 S983 中，在判定为数据更新权证(DUT(PAR))上记述的 Old Data Code(被更新的旧数据的代码)所示的数据不是分区密钥区域(参照图 23)中保存的 Kdut_PAR1(数据更新权证(DUT)的 MAC 验证密钥)、Kdut_PAR2(数据更新加密密钥)的情况下，在步骤 S986 中，用设备的分区密钥区域(参照图 23)中保存的 Kdut_PAR2(数据更新加密密钥)对数据更新权证(DUT(PAR))上保存的新数据[New Data]进行解密，盖写到数据更新权证(DUT(PAR))的 Old Data Code(被更新的旧数据的代码)所示的区域上。其中，在为待更新数据附加了版本的情况下，将数据更新权证(DUT(PAR))中保存的要更新的数据的版本(New Data Version)保存到与设备内的更新数据对应而设定的版本保存区域中。

以上处理是设备中执行的基于数据更新权证的数据更新操作。

从上述流程可以理解，在待更新数据是设备密钥区域中保存的

Kdut_DEV1(数据更新权证(DUT)的 MAC 验证密钥)

Kdut_DEV2(数据更新加密密钥)

或分区密钥区域中保存的

Kdut_PAR1(数据更新权证(DUT)的 MAC 验证密钥)

Kdut_PAR2(数据更新加密密钥)

的情况下，执行与其他更新处理不同的处理。

对这些 Kdut_DEV1(数据更新权证(DUT)的 MAC 验证密钥)、Kdut_DEV2(数据更新加密密钥)、Kdut_PAR1(数据更新权证(DUT)的 MAC 验证密钥)、Kdut_PAR2(数据更新加密密钥)的更新处理简洁地集

中示于图 95，下面说明该处理。按图 95 的(1)～(3)的顺序来进行说明。其中，Kdut_DEV1、2 和 Kdut_PAR1、2 的处理相同，所以说明更新 Kdut_DEV1、2 的情况。

(1)用设备的设备密钥区域(参照图 18)中保存的 Kdut_DEV4(数据更新加密密钥)对数据更新权证(DUT)中保存的作为新数据[New Data]的 Kdut_DEV1、Kdut_DEV2 进行加密后，保存到数据更新权证(DUT)中，将数据更新权证(DUT)发送到设备。此时，能够更新 Kdut_DEV1、Kdut_DEV2 的权证发行者必须知道 Kdut_DEV3、Kdut_DEV4。

(2)接收到数据更新权证(DUT)的设备用设备的设备密钥区域中保存的 Kdut_DEV4(数据更新加密密钥)对数据更新权证(DUT)保存的作为新数据[New Data]的 Kdut_DEV1、Kdut_DEV2 进行解密，盖写到设备的设备密钥区域中保存的 Kdut_DEV1、Kdut_DEV2 上。

(3)接着，设备交换设备的设备密钥区域(参照图 18)中新保存的 Kdut_DEV1(数据更新权证(DUT)的 MAC 验证密钥)、和以前已保存的 Kdut_DEV3(数据更新权证(DUT)的 MAC 验证密钥)。进而，交换新保存的 Kdut_DEV2(数据更新加密密钥)、和以前已保存的 Kdut_DEV4(数据更新加密密钥)。

通过该交换 Kdut_DEV1 和 Kdut_DEV3、及交换 Kdut_DEV2 和 Kdut_DEV4，来始终维持 Kdut_DEV3(数据更新权证(DUT)的 MAC 验证密钥)、Kdut_DEV4(数据更新加密密钥)这一对儿比 Kdut_DEV1(数据更新权证(DUT)的 MAC 验证密钥)、Kdut_DEV2(数据更新加密密钥)这一对儿的版本新。即，密钥 Kdut_DEV1 和 Kdut_DEV2 是经常使用的密钥，Kdut_DEV3 和 Kdut_DEV4 具有下述作为备份密钥的作用：在非常时更新 Kdut_DEV1 和 Kdut_DEV2，并且被置换为当前正在使用的密钥 Kdut_DEV1 和 Kdut_DEV2。

其中，Kdut_DEV1(数据更新权证(DUT)的 MAC 验证密钥)、Kdut_DEV2(数据更新加密密钥)被作为一对儿来使用，而 Kdut_DEV3(数据更新权证(DUT)的 MAC 验证密钥)、Kdut_DEV4(数据更新加密密钥)也被作为一对儿来使用。

以上，参照特定的实施例详解了本发明。然而，在不脱离本发明的精神的范围内，本领域的技术人员显然可修正或代用该实施例。即，以例示的形式公开了本发明，不应限定性地解释。为了判断本发明的

精神，应参考冒头记载的权利要求书一栏。

其中，说明书中说明过的一系列处理可通过硬件、软件、或两者的复合结构来执行。在通过软件来执行处理的情况下，可将记录有处理序列的程序安装到专用硬件中包含的计算机内的存储器中来执行，或者将程序安装到可执行各种处理的通用计算机中来执行。

例如，可以将程序预先记录在作为记录媒体的硬盘或 ROM (Read Only Memory，只读存储器)中。或者，可以将程序暂时或永久性地保存(记录)在软盘、CD-ROM (Compact Disc Read Only Memory，光盘只读存储器)、MO (Magneto optical，磁光)盘、DVD (Digital Versatile Disc，数字多功能盘)、磁盘、半导体存储器等可拆卸记录媒体上。可以将这种可拆卸记录媒体作为所谓的软件包来提供。

其中，程序除了从上述可拆卸记录媒体安装到计算机上之外，也可以从下载站点无线传送到计算机，或者经 LAN (Local Area Network，局域网)、因特网等网络有线传送到计算机，计算机接收这样传送来的程序，安装到内置的硬盘等记录媒体上。

其中，说明书中记载的各种处理不仅可根据记载按时间顺序来进行，也可以按照执行处理的装置的处理能力或需要来并行或个别地执行。此外，在本说明书中，系统是指多个装置的逻辑集合结构，各结构的装置不一定位于同一壳体内。

产业上的可利用性

如上所述，根据本发明的存储器存取控制系统、设备管理装置、分区管理装置、搭载有存储器的设备、及存储器存取控制方法、以及程序存储媒体，可在各设备或分区管理实体的管理下对分割为多个分区的存储区域的存取发行各种存取控制权证，在搭载有存储器的设备中根据各权证上记述的规则来执行处理，实现了各分区内数据的独立的管理结构。

再者，根据本发明的存储器存取控制系统、设备管理装置、分区管理装置、搭载有存储器的设备、及存储器存取控制方法、以及程序存储媒体，可使设备根据公开密钥、对称密钥中的某一种指定体制来执行分区鉴别、设备鉴别，在各种环境下执行设备及存取装置间的安全的数据通信。

再者，根据本发明的存储器存取控制系统、设备管理装置、分区管理装置、搭载有存储器的设备、及存储器存取控制方法、以及程序存储媒体，搭载有存储器的设备的存储部具有：1 个以上的分区区域，保存数据文件，作为由分区管理器管理的存储区域；设备管理器管理区域，由作为该搭载有存储器的设备的管理者的分区管理器管理；从存取机器接收设备管理器管理的存取控制权证、或分区管理器管理的存取控制权证作为对存储部的存取控制权证，按照所接收权证的记述来执行处理，指定应执行的相互鉴别形态、存取控制权证的验证形态，根据各形态来进行处理，所以可在各种环境下执行设备及存取装置间的安全的数据通信。

再者，根据本发明的存储器存取控制系统、设备管理装置、分区管理装置、搭载有存储器的设备、及存储器存取控制方法、以及程序存储媒体，在设备管理器、分区管理器的管理下，发行分区注册权证(PRT)、文件注册权证(FRT)、服务许可权证(SPT)、数据更新权证(DUT)，分别以鉴别、权证验证成立为条件来执行设备中的处理，所以可在各服务主体的管理下按照各种处理形态来提供服务、进行数据管理。

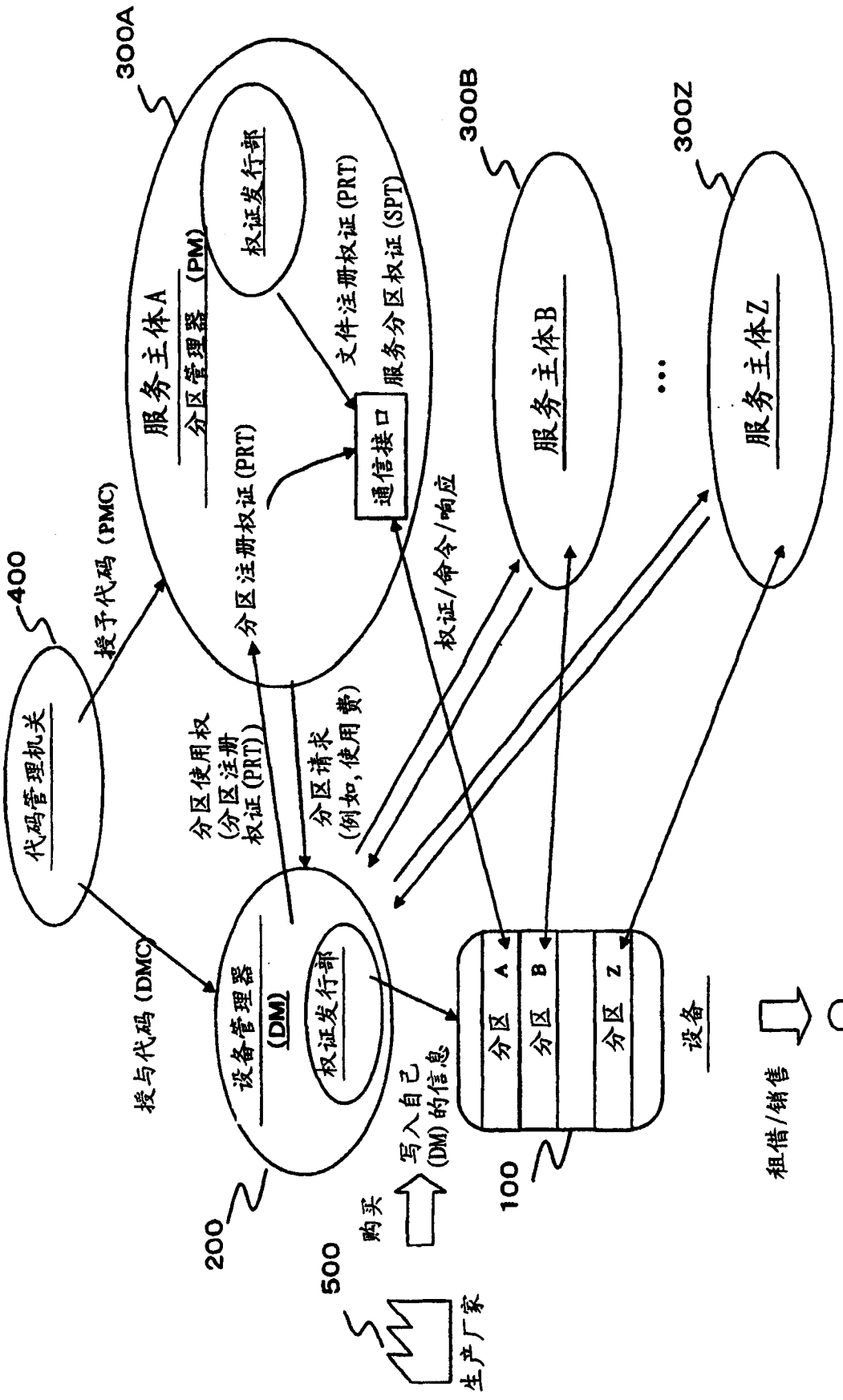

300A
服务主体A
分区管理器 (PM)
权证发行部
文件注册权证 (PRT)
服务分区权证 (SPT)
通信接口
分区注册权证 (PRT)
300B
服务主体B
300Z
服务主体Z
400
代码管理机关
授予代码 (PMC)
分区使用权
(分区注册
权证 (PRT))
分区请求
(例如,使用费)
权证/命令/响应
设备管理器
(DM)
权证发行部
授与代码 (DMC)
写入自己
(DM) 的信息
分区 A
分区 B
分区 Z
设备
200
100
500
购买
生产厂家
租借/销售
用户

图 1

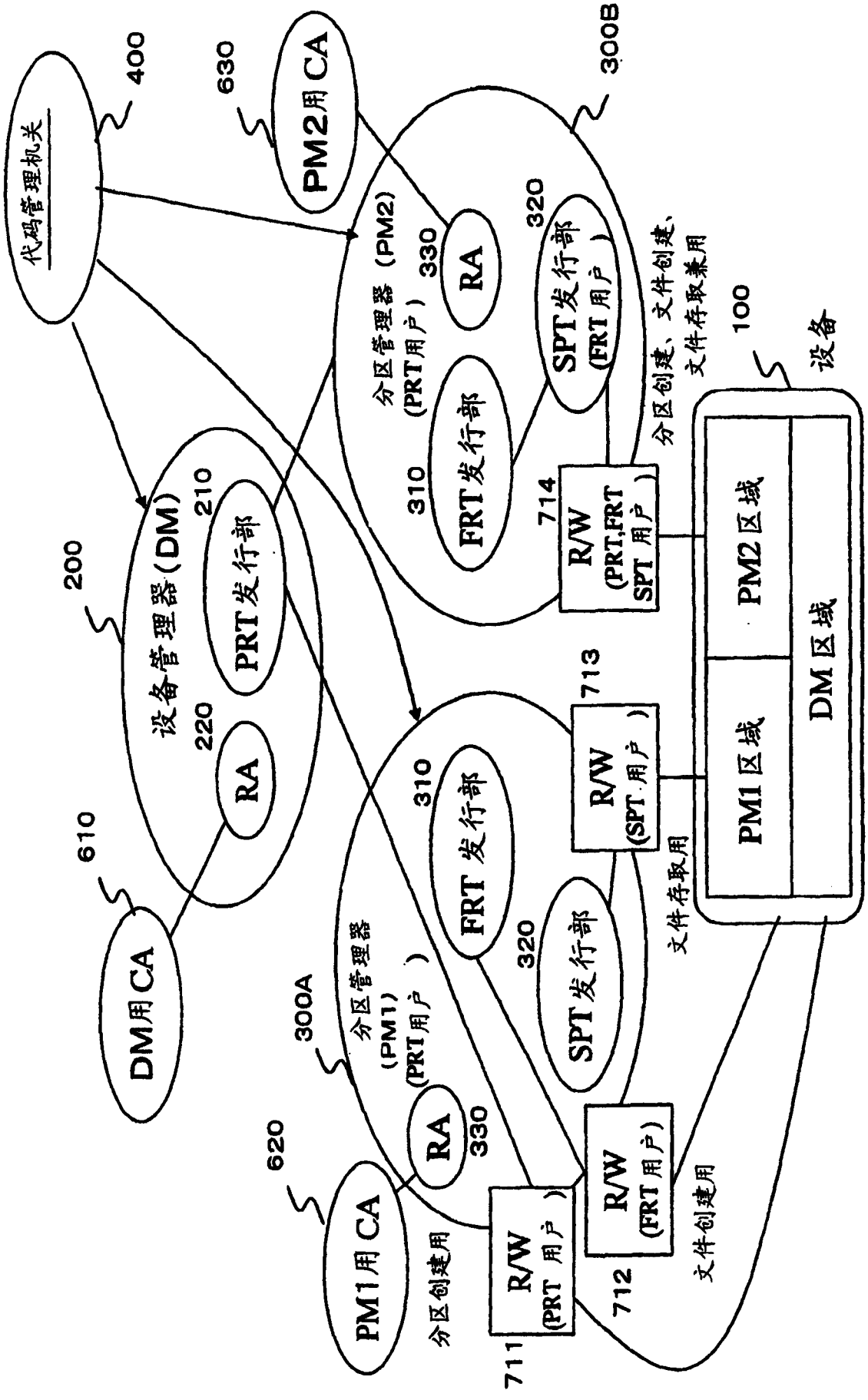

代码管理机关
400
PM2用CA
630
300B
分区管理器（PM2）
（PRT用户）
330
RA
320
SPT发行部
（FRT用户）
310
FRT发行部
714
R/W
（PRT,FRT
SPT用户）
分区创建、文件存取兼用
100
设备
PM2区域
DM区域
PM1区域
文件存取用
713
R/W
（SPT用户）
200
设备管理器（DM）
210
PRT发行部
220
RA
610
DM用CA
310
FRT发行部
320
SPT发行部
300A
分区管理器
（PM1）
（PRT用户）
620
PM1用CA
330
RA
分区创建用
711
R/W
（PRT用户）
712
R/W
（FRT用户）
文件创建用

图 2

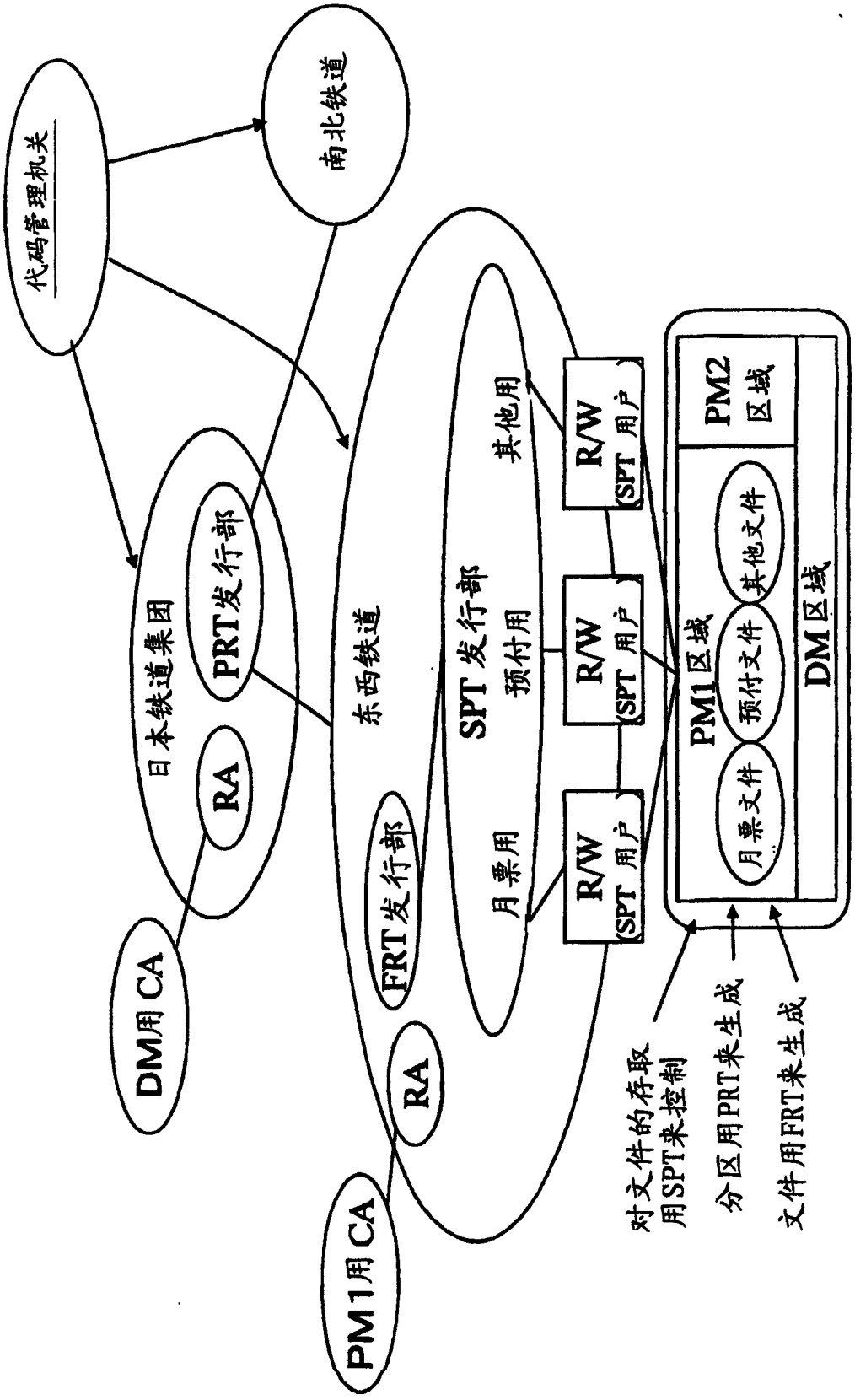

代码管理机关
南北铁道
日本铁道集团
PRT发行部
RA
DM用CA
东西铁道
FRT发行部
RA
PM1用CA
SPT发行部
月票用
预付用
其他用
R/W
(SPT 用户)
R/W
(SPT 用户)
R/W
(SPT 用户)
PM1区域
月票文件
预付文件
其他文件
DM区域
PM2
区域
对文件的存取
用SPT来控制
分区用PRT来生成
文件用FRT来生成

图 3

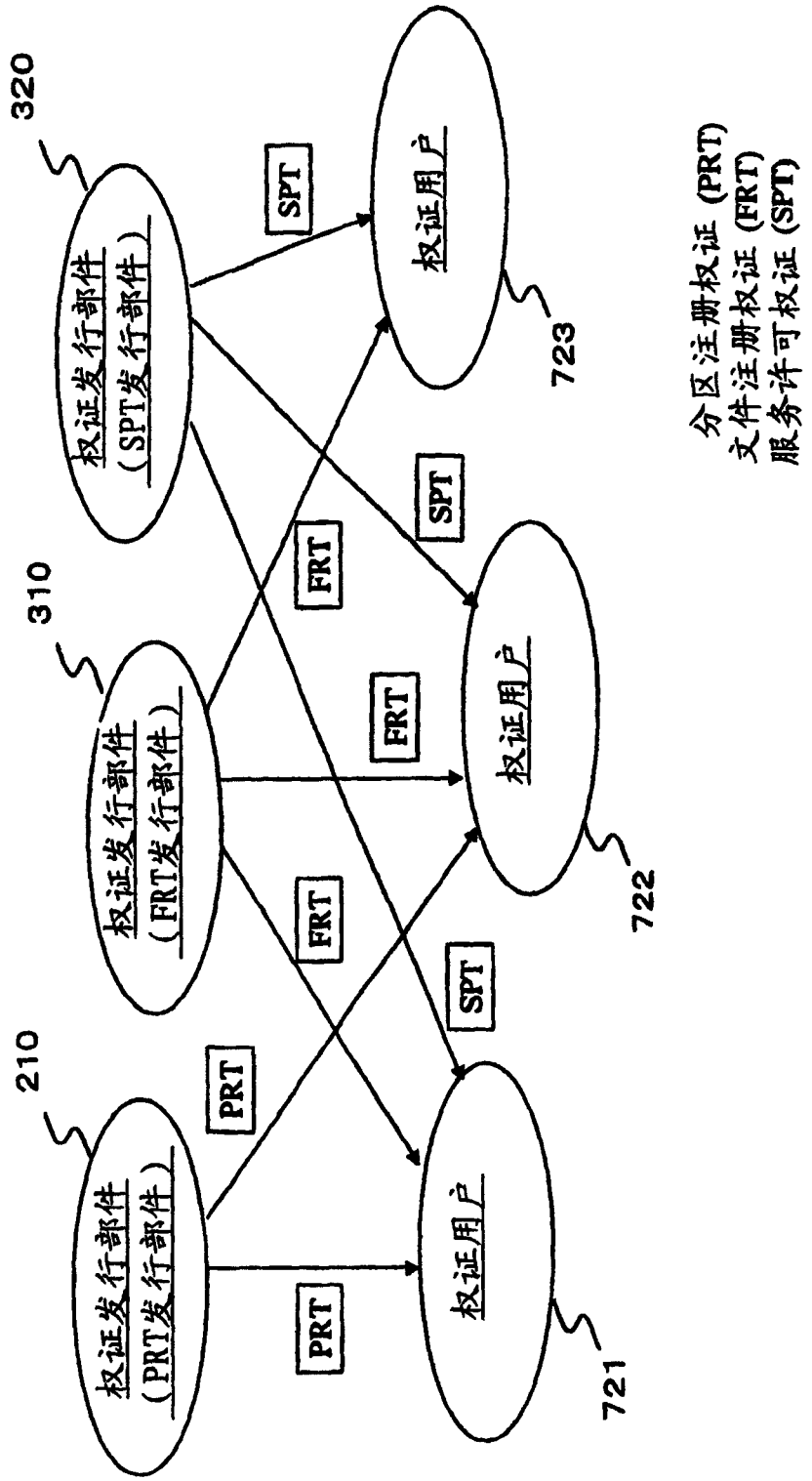

320
权证发行部件
(SPT发行部件)
310
权证发行部件
(FRT发行部件)
210
权证发行部件
(PRT发行部件)
PRT
FRT
SPT
权证用户
721
722
723
分区注册权证 (PRT)
文件注册权证 (FRT)
服务许可权证 (SPT)

图 4

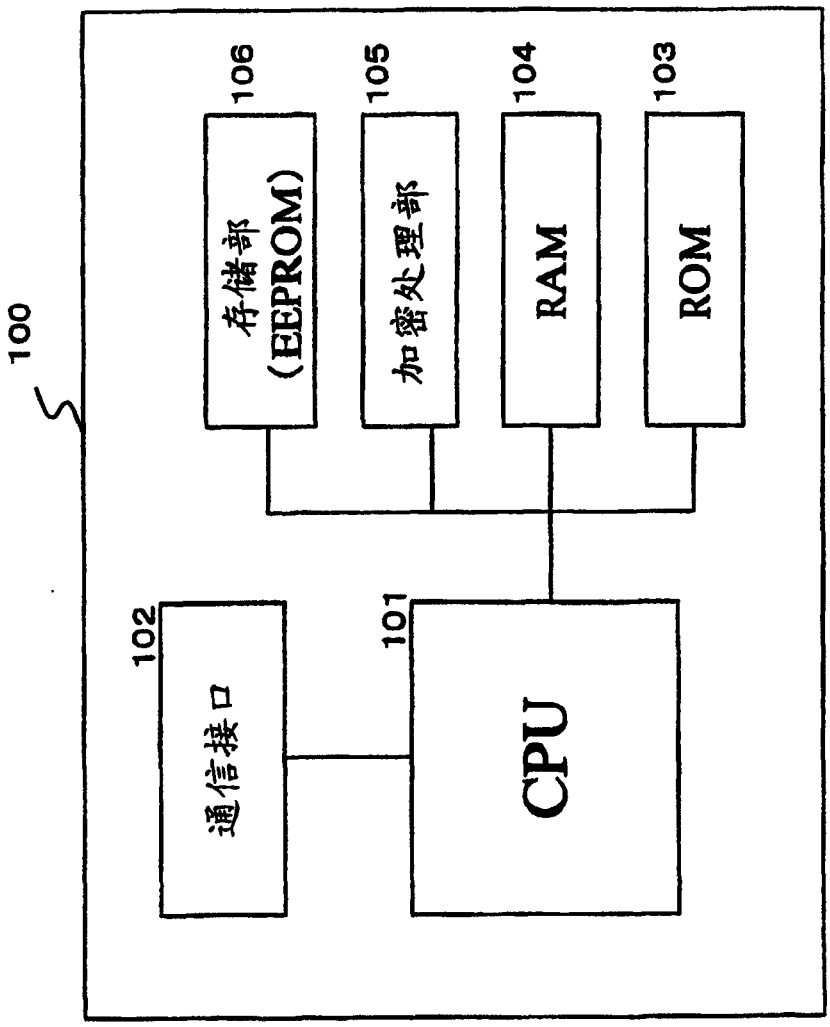


图 5

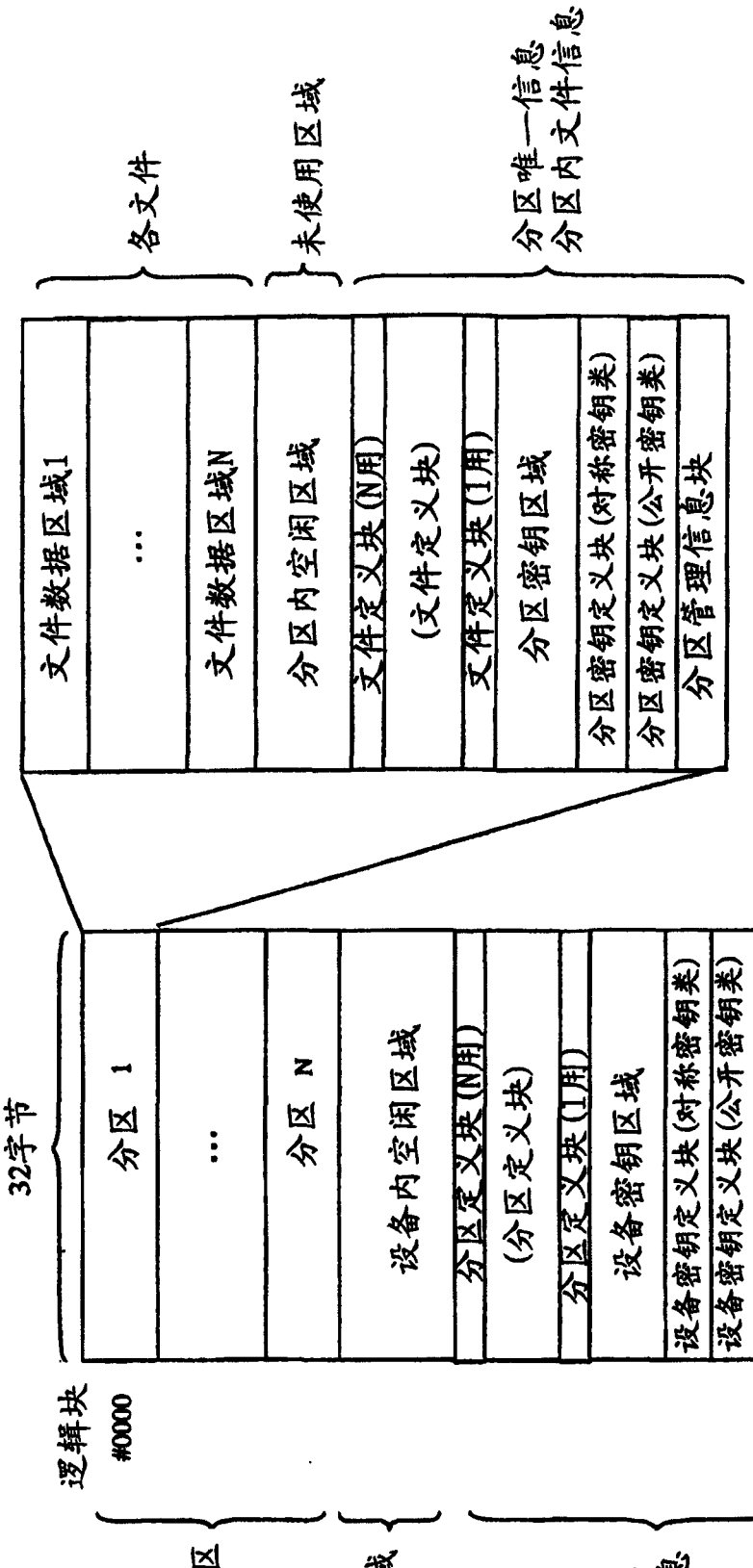

逻辑块
#0000
32字节
分区 1
…
分区 N
设备内空闲区域
分区定义块(N用)
(分区定义块)
分区定义块(1用)
设备密钥区域
设备密钥定义块(对称密钥类)
设备密钥定义块(公开密钥类)
设备管理信息块
生产信息块
#FFFFF
各分区
未使用区域
设备唯一信息
设备内分区信息
文件数据区域1
…
文件数据区域N
分区内空闲区域
文件定义块(N用)
(文件定义块)
文件定义块(1用)
分区密钥区域
分区密钥定义块(对称密钥类)
分区密钥定义块(公开密钥类)
分区管理信息块
各文件
未使用区域
分区唯一信息
分区内文件信息

图 6

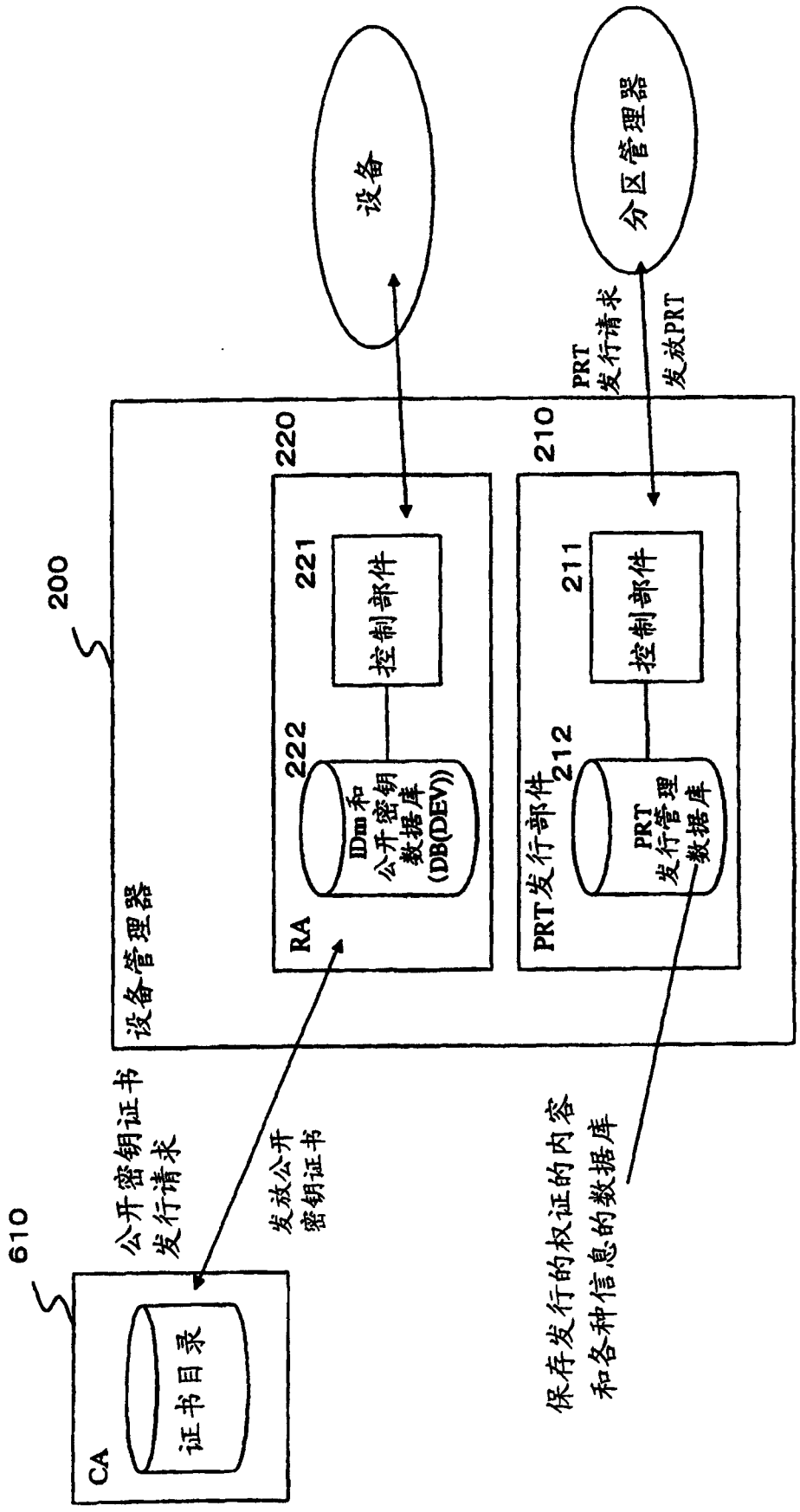

设备
分区管理器
PRT
发行请求
发放PRT
200
设备管理器
220
RA
221
控制部件
222
IDm和
公开密钥
数据库
(DB(DEV))
210
PRT发行部件
211
控制部件
212
PRT
发行管理
数据库
610
CA
证书目录
公开密钥证书
发行请求
发放公开
密钥证书
保存发行的权证的内容
和各种信息的数据库

图 7

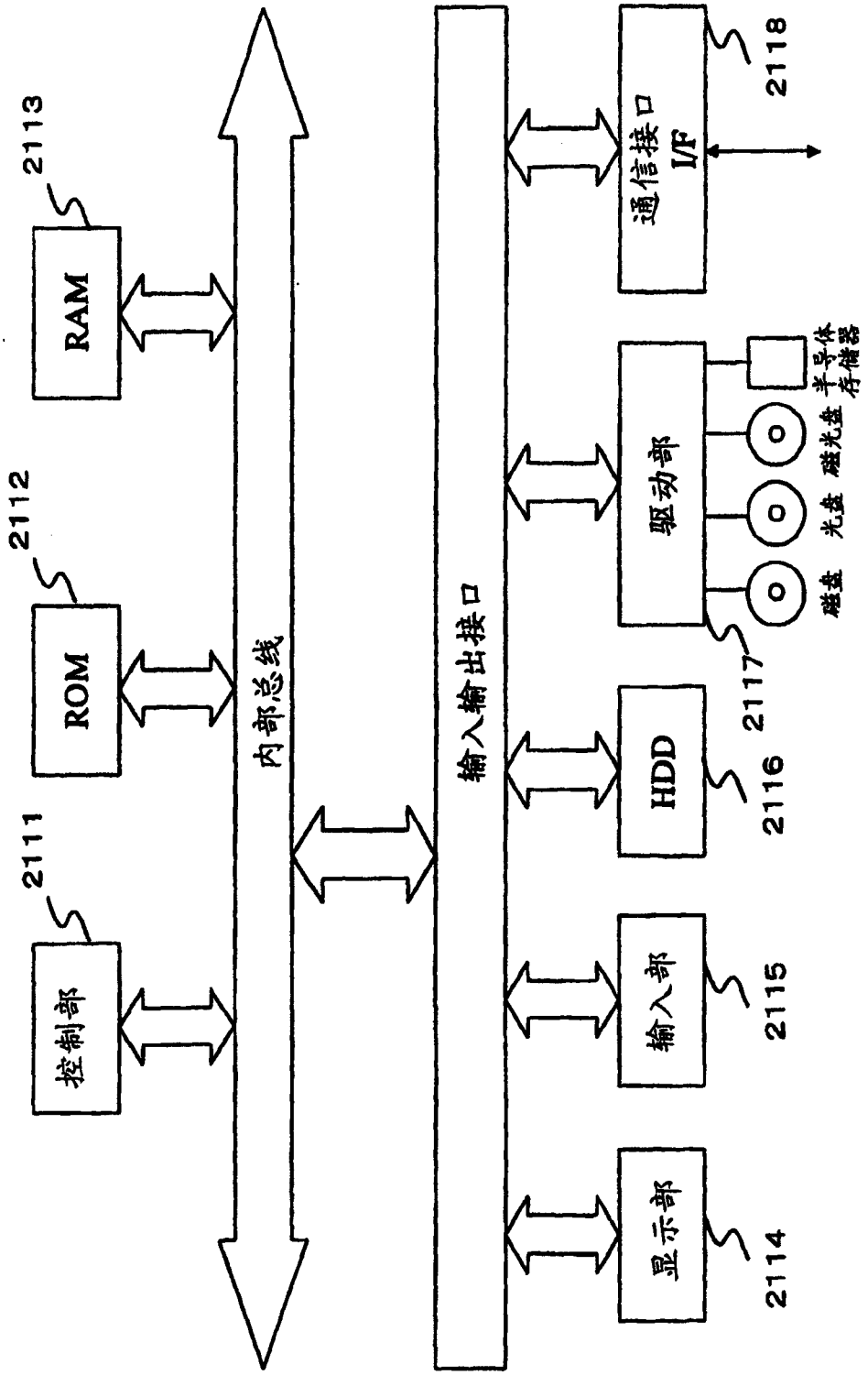

2113
RAM
2112
ROM
2111
控制部
内部总线
输入输出接口
通信接口
I/F
2118
驱动部
磁盘
光盘
磁光盘
半导体存储器
2117
HDD
2116
输入部
2115
显示部
2114

图 8

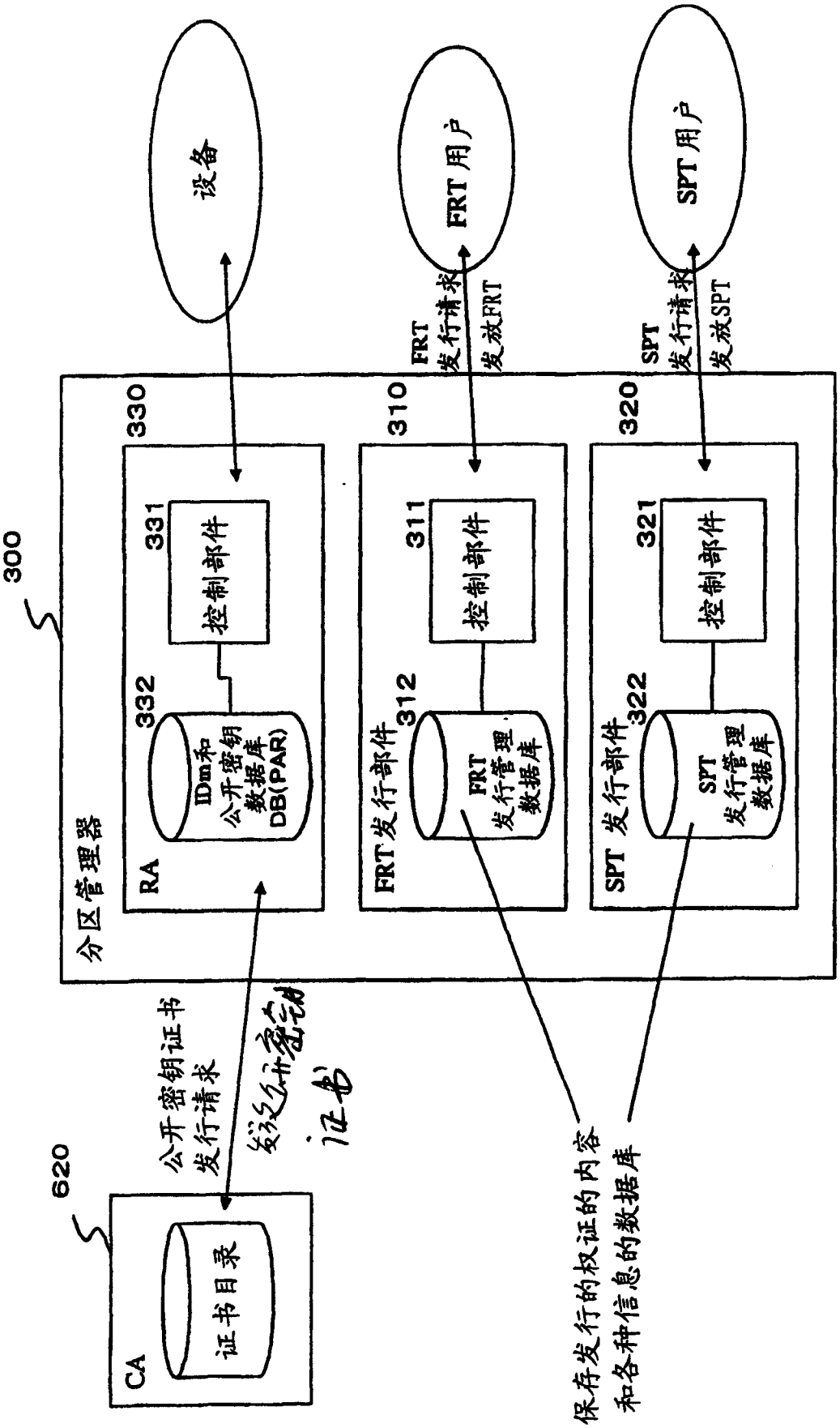

300
分区管理器
330
RA
331
控制部件
332
IDm和公开密钥数据库DB(PAR)
310
FRT发行部件
311
控制部件
312
FRT发行管理数据库
320
SPT 发行部件
321
控制部件
322
SPT发行管理数据库
设备
FRT 用户
SPT 用户
FRT发行请求
发放FRT
SPT发行请求
发放SPT
620
CA
证书目录
公开密钥证书发行请求
保存发行的权证的内容和各种信息的数据库

图 9

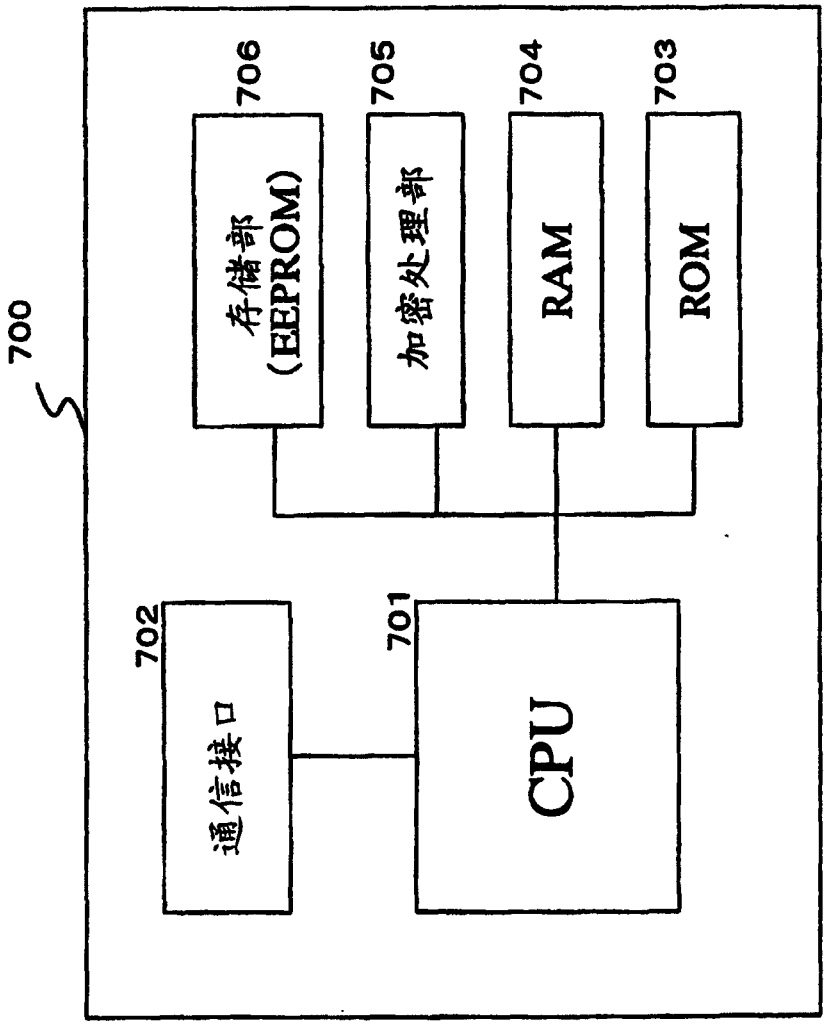


图 10

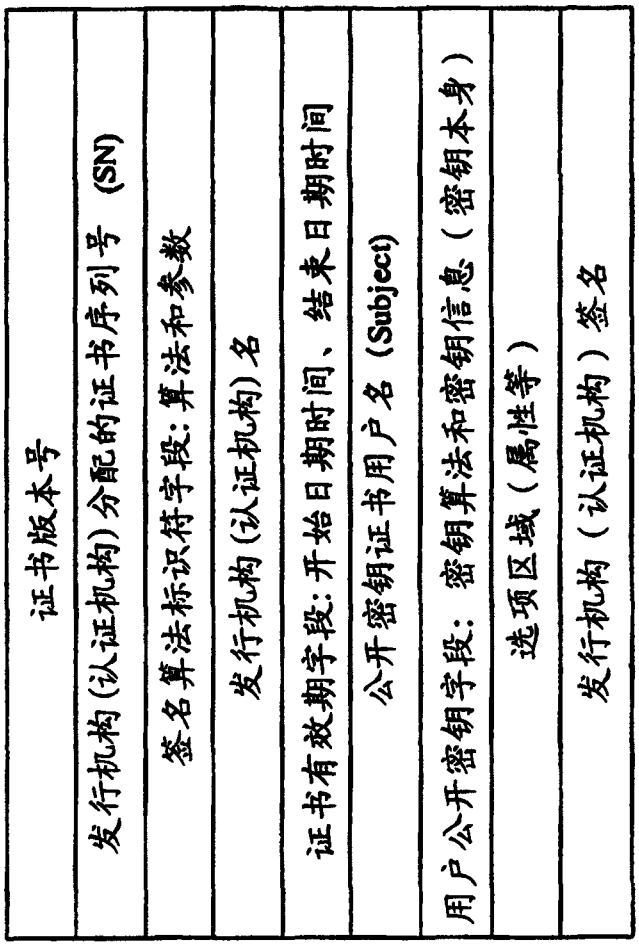

证书版本号
发行机构(认证机构)分配的证书序列号 (SN)
签名算法标识符字段:算法和参数
发行机构(认证机构)名
证书有效期字段:开始日期时间、结束日期时间
公开密钥证书用户名(Subject)
用户公开密钥字段:密钥算法和密钥信息(密钥本身)
选项区域(属性等)
发行机构(认证机构)签名

图 11

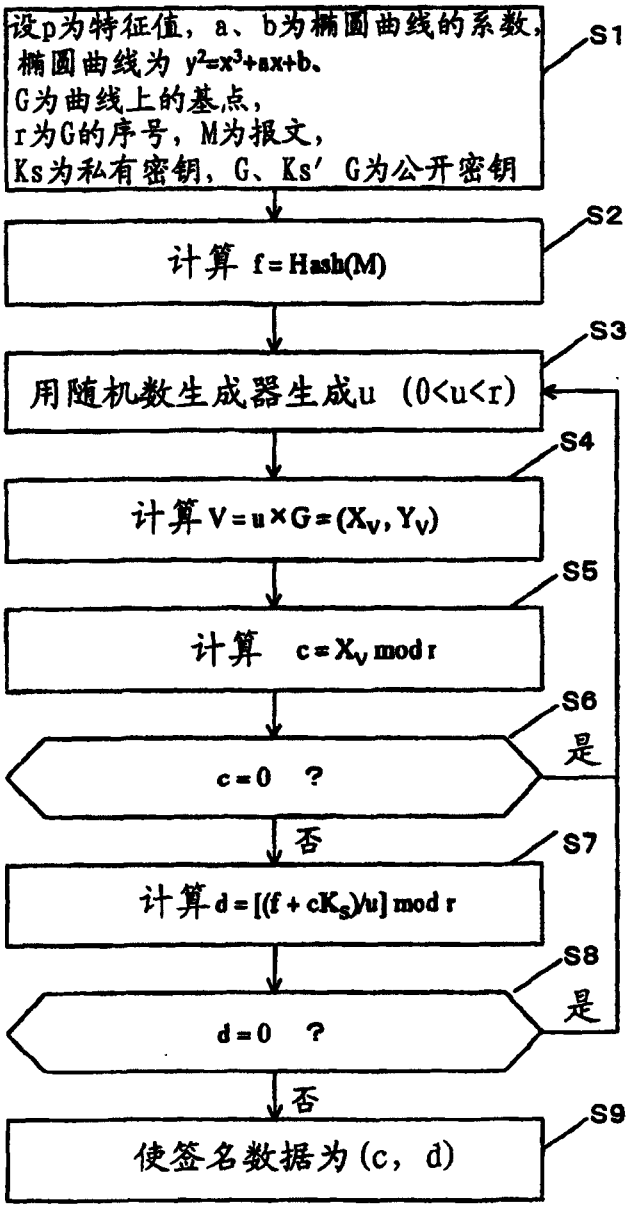

设p为特征值，a、b为椭圆曲线的系数，
椭圆曲线为 $y^2=x^3+ax+b$、
G为曲线上的基点，
r为G的序号，M为报文，
Ks为私有密钥，G、Ks′ G为公开密钥
S1
计算 f=Hash(M)
S2
用随机数生成器生成u (0<u<r)
S3
计算 $V=u\times G=(X_V, Y_V)$
S4
计算 $c=X_V \bmod r$
S5
c=0 ?
S6
是
否
计算 $d=[(f+cK_S)/u] \bmod r$
S7
d=0 ?
S8
是
否
使签名数据为(c，d)
S9

图 12

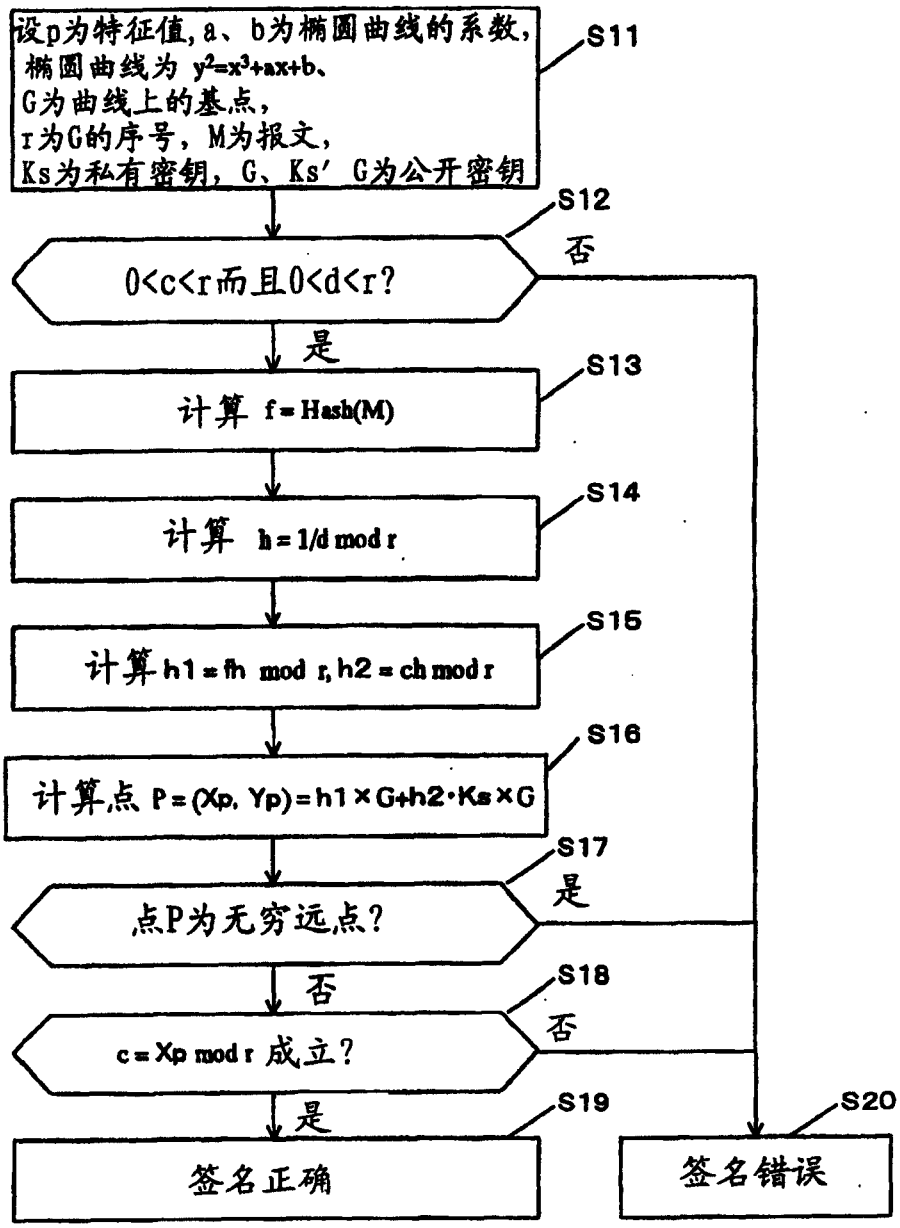

设p为特征值，a、b为椭圆曲线的系数，
椭圆曲线为 $y^2=x^3+ax+b$、
G为曲线上的基点，
r为G的序号，M为报文，
Ks为私有密钥，G、Ks′ G为公开密钥
S11
S12
0<c<r而且0<d<r?
否
是
S13
计算 f=Hash(M)
S14
计算 h=1/d mod r
S15
计算h1=fh mod r, h2=ch mod r
S16
计算点 P=(Xp, Yp)=h1×G+h2·Ks×G
S17
点P为无穷远点?
是
否
S18
c=Xp mod r 成立?
否
是
S19
签名正确
S20
签名错误

图 13

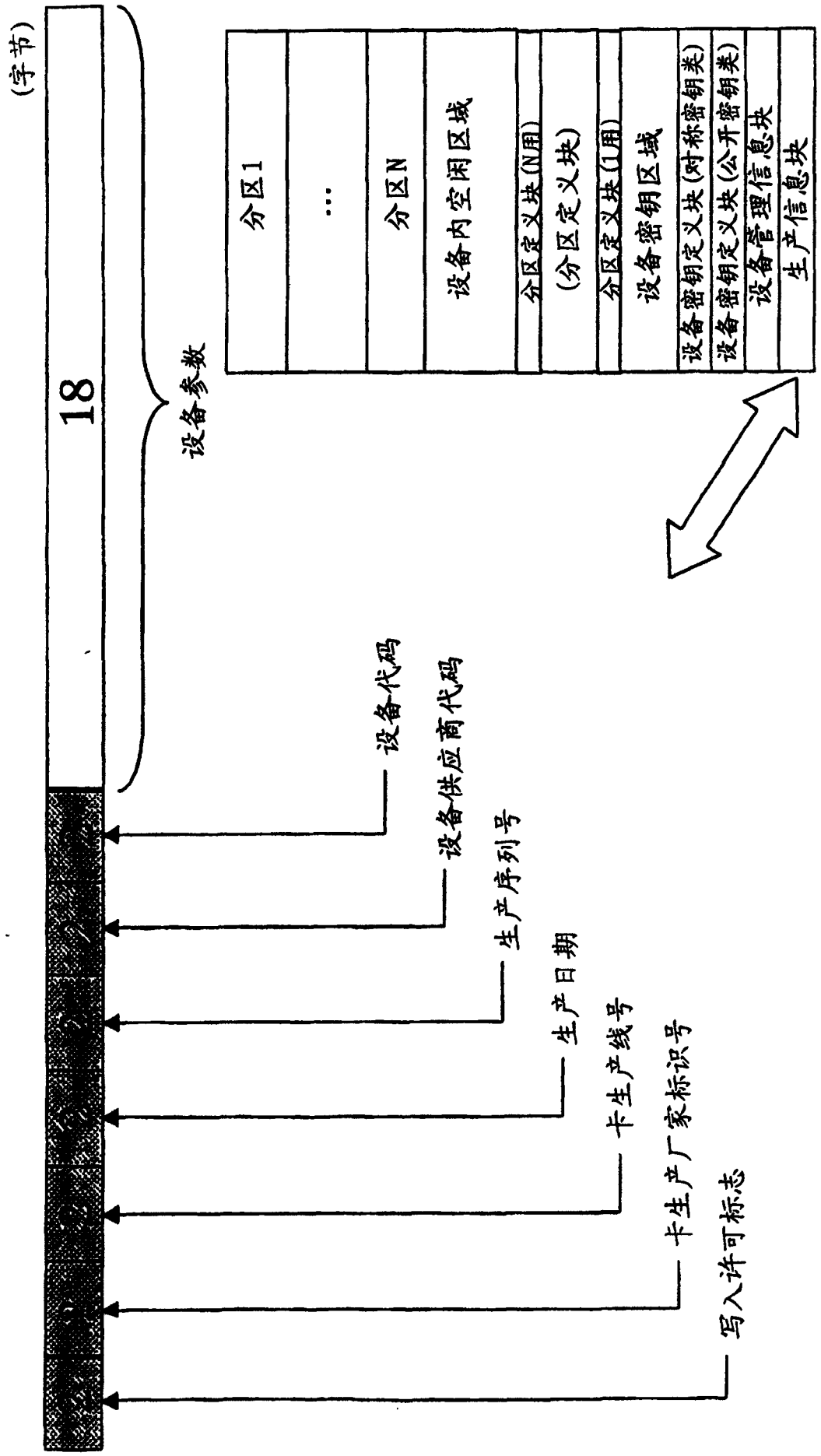

(字节)
18
设备参数
设备代码
设备供应商代码
生产序列号
生产日期
卡生产线号
卡生产厂家标识号
写入许可标志
分区1
...
分区N
设备内空闲区域
分区定义块(N用)
(分区定义块)
分区定义块(1用)
设备密钥区域
设备密钥定义块(对称密钥类)
设备密钥定义块(公开密钥类)
设备管理信息块
生产信息块

图 14

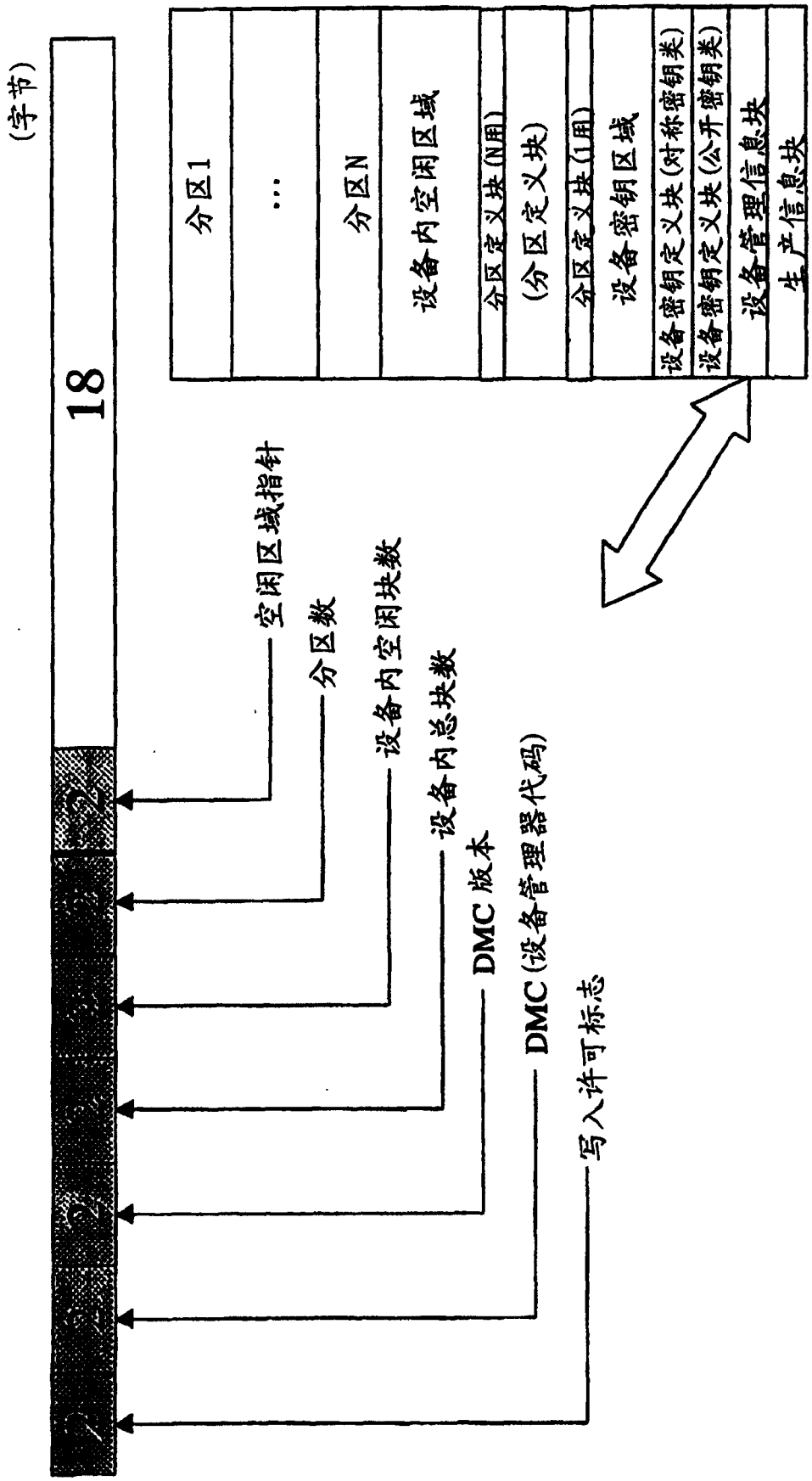

(字节)
18
空闲区域指针
分区数
设备内空闲块数
设备内总块数
DMC 版本
DMC (设备管理器代码)
写入许可标志
分区1
...
分区N
设备内空闲区域
分区定义块(N用)
(分区定义块)
分区定义块(1用)
设备密钥区域
设备密钥定义块(对称密钥类)
设备密钥定义块(公开密钥类)
设备管理信息块
生产信息块

图 15

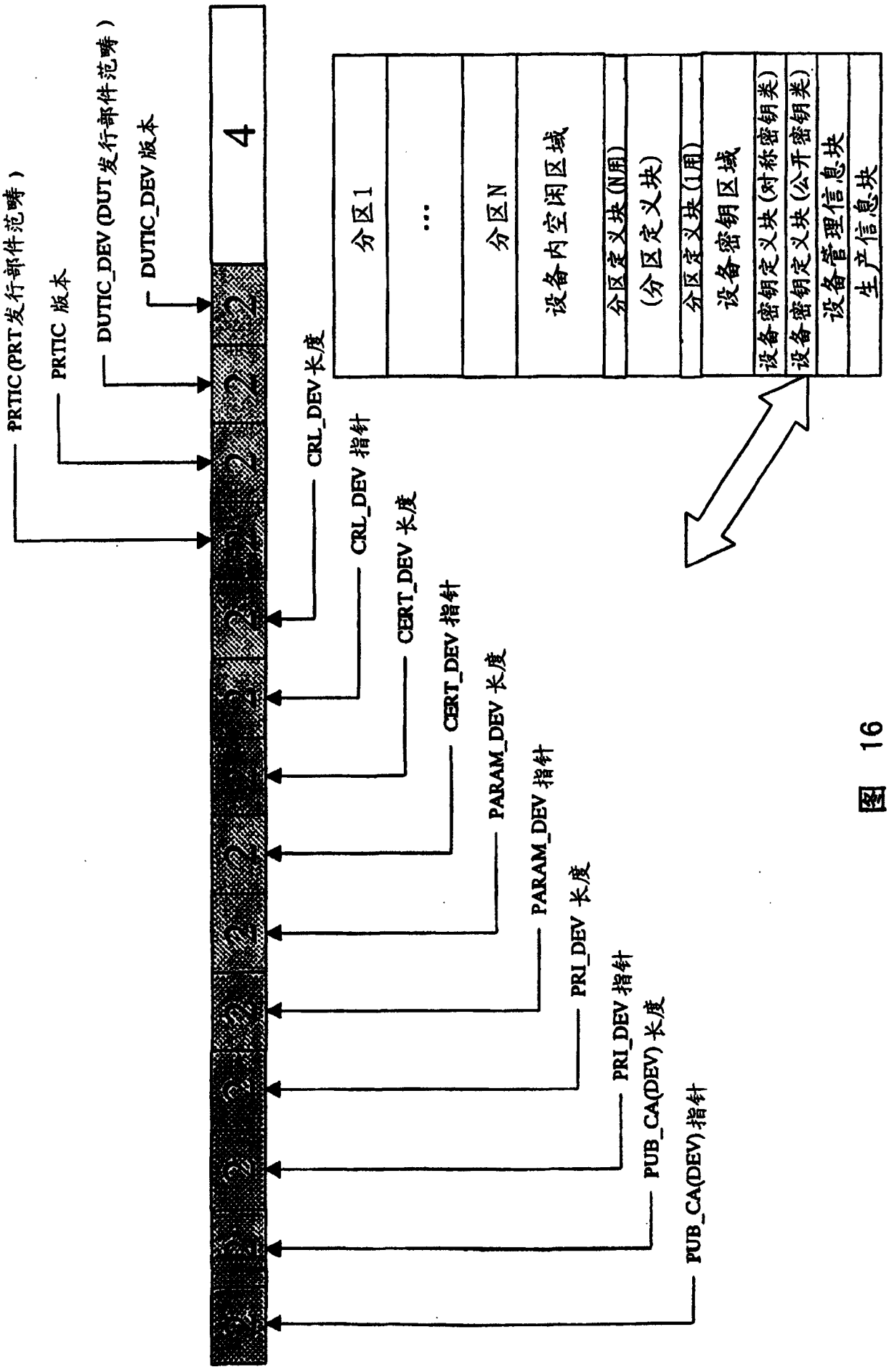

PRTIC(PRT发行部件范畴)
PRTIC 版本
DUTIC_DEV(DUT发行部件范畴)
DUTIC_DEV 版本
CRL_DEV 长度
CRL_DEV 指针
CERT_DEV 长度
CERT_DEV 指针
PARAM_DEV 长度
PARAM_DEV 指针
PRI_DEV 长度
PRI_DEV 指针
PUB_CA(DEV)长度
PUB_CA(DEV)指针
4
分区1
...
分区N
设备内空闲区域
分区定义块(N用)
(分区定义块)
分区定义块(1用)
设备密钥区域
设备密钥定义块(对称密钥类)
设备密钥定义块(公开密钥类)
设备管理信息块
生产信息块

图 16

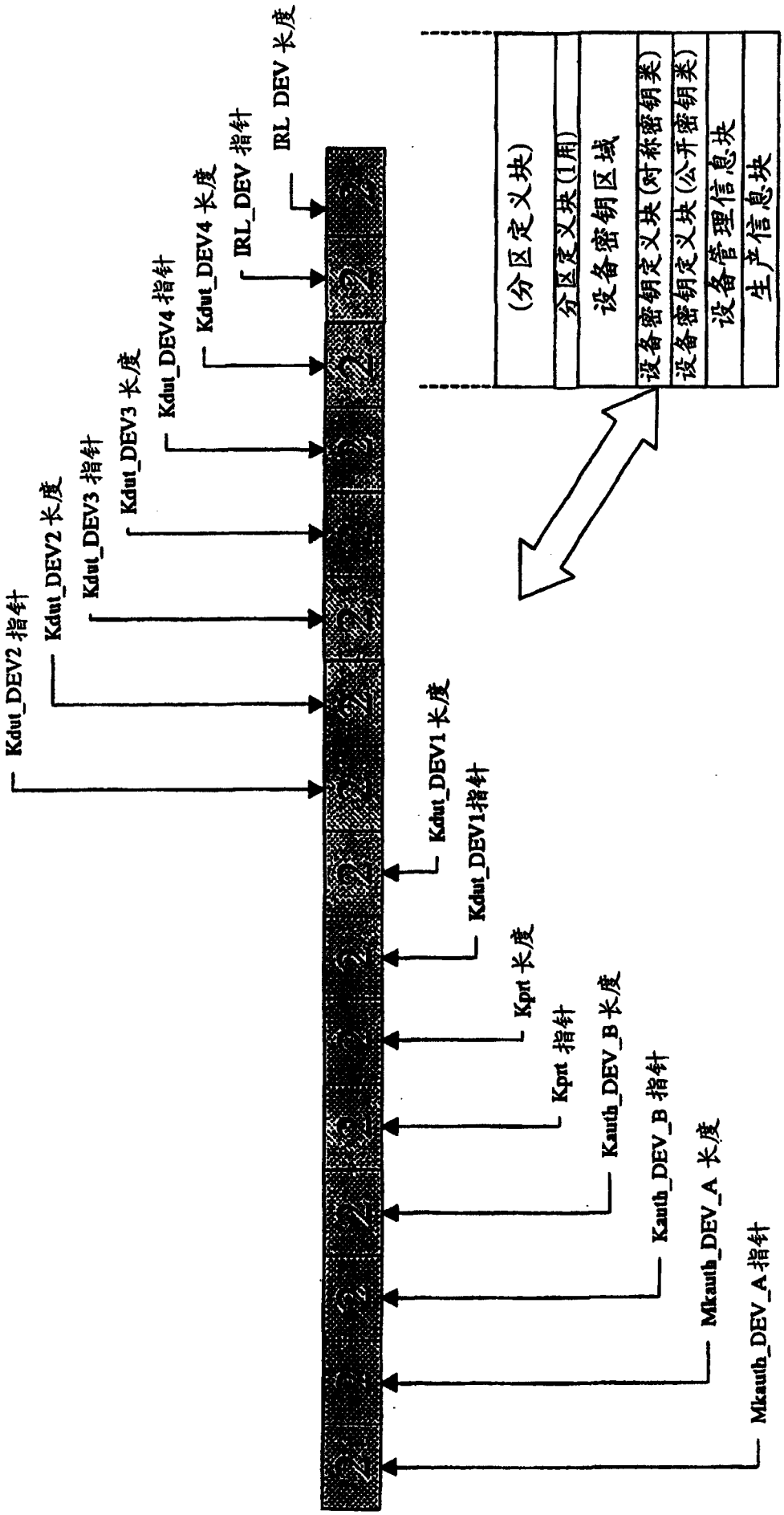

Kdut_DEV2 指针
Kdut_DEV2 长度
Kdut_DEV3 指针
Kdut_DEV3 长度
Kdut_DEV4 指针
Kdut_DEV4 长度
IRL_DEV 指针
IRL DEV 长度
Kdut_DEV1指针
Kdut_DEV1长度
Kprt 指针
Kprt 长度
Kauth_DEV_B 指针
Kauth_DEV_B 长度
Mkauth_DEV_A 指针
Mkauth_DEV_A 长度
(分区定义块)
分区定义块(1用)
设备密钥区域
设备密钥定义块(对称密钥类)
设备密钥定义块(公开密钥类)
设备管理信息块
生产信息块

图 17

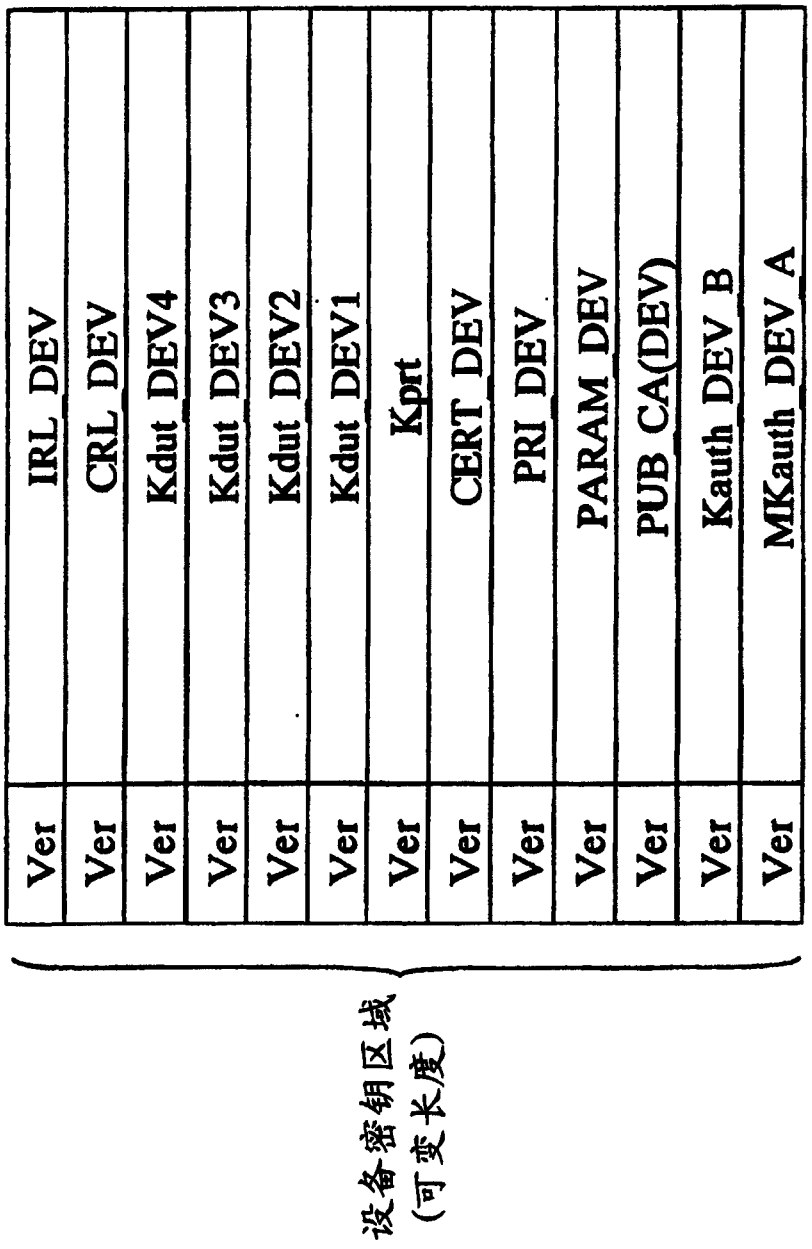

Ver IRL_DEV
Ver CRL_DEV
Ver Kdut_DEV4
Ver Kdut_DEV3
Ver Kdut_DEV2
Ver Kdut_DEV1
Ver Kprt
Ver CERT_DEV
Ver PRI_DEV
Ver PARAM_DEV
Ver PUB CA(DEV)
Ver Kauth_DEV_B
Ver MKauth_DEV_A
设备密钥区域
(可变长度)

图 18

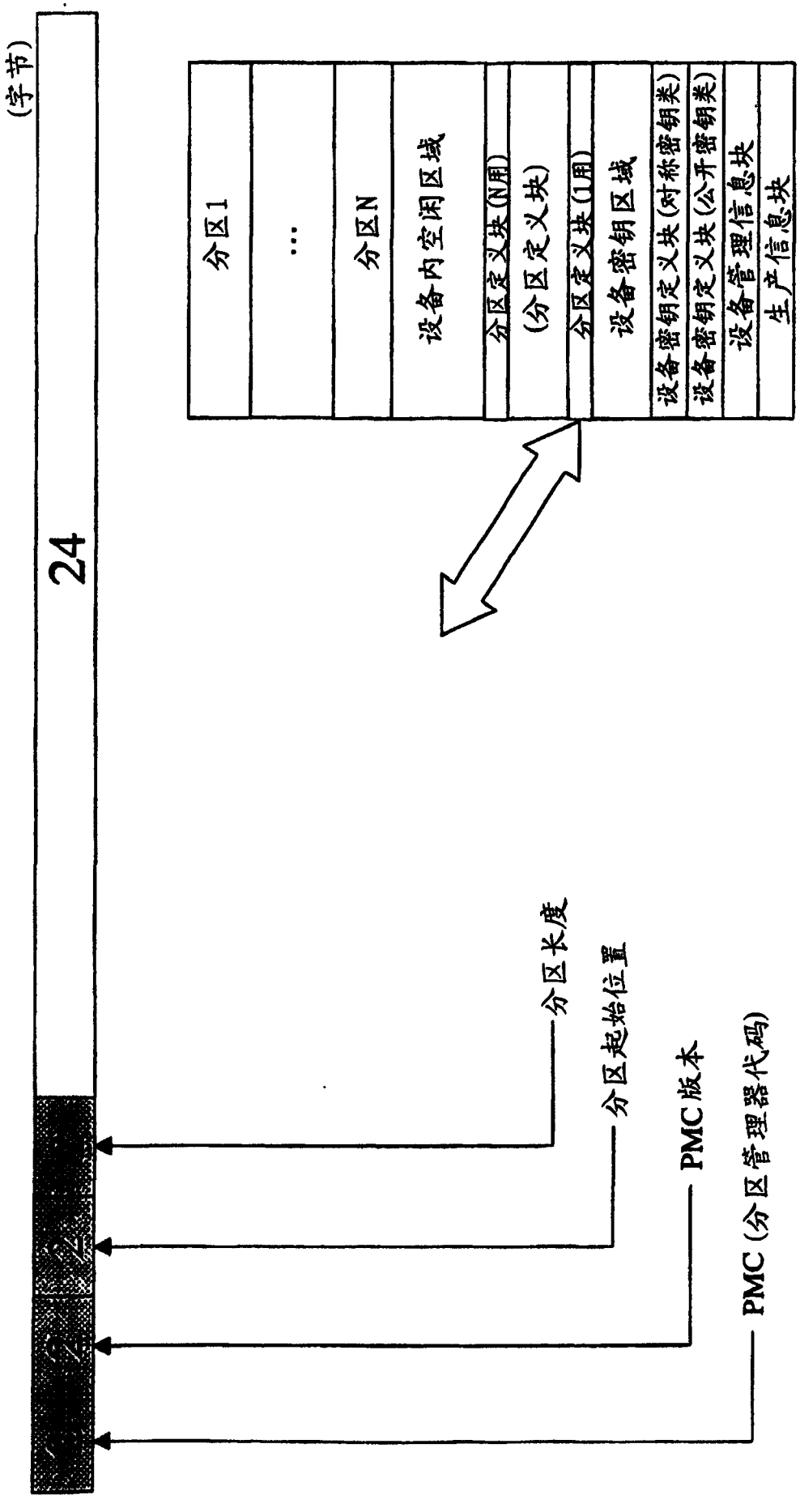

(字节)
24
分区长度
分区起始位置
PMC版本
PMC (分区管理器代码)
分区1
...
分区N
设备内空闲区域
分区定义块(N用)
(分区定义块)
分区定义块(1用)
设备密钥区域
设备密钥定义块(对称密钥类)
设备密钥定义块(公开密钥类)
设备管理信息块
生产信息块

图 19

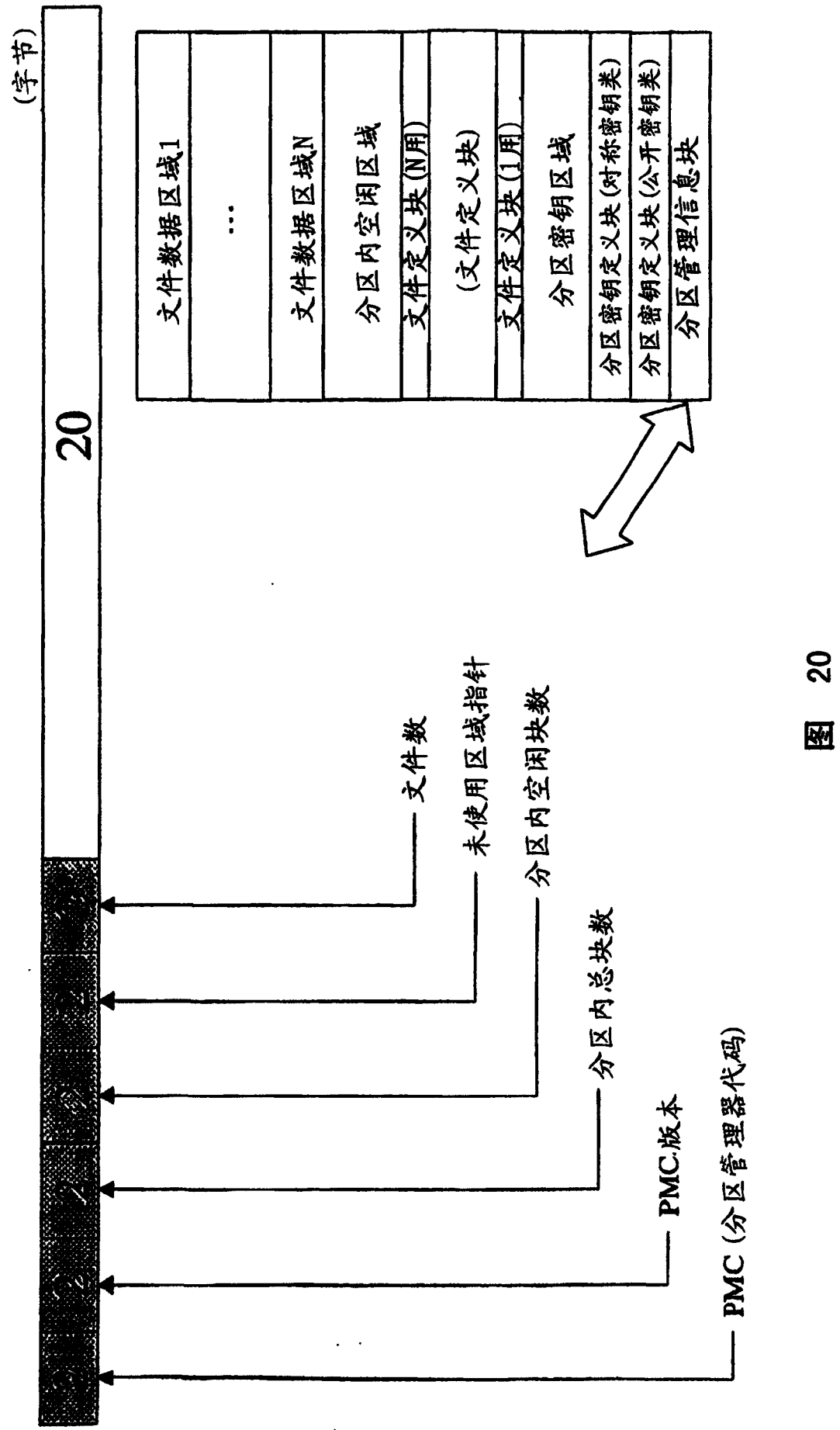

(字节)
20
文件数
未使用区域指针
分区内空闲块数
分区内总块数
PMC版本
PMC(分区管理器代码)
文件数据区域1
...
文件数据区域N
分区内空闲区域
文件定义块(N用)
(文件定义块)
文件定义块(1用)
分区密钥区域
分区密钥定义块(对称密钥类)
分区密钥定义块(公开密钥类)
分区管理信息块

图 20

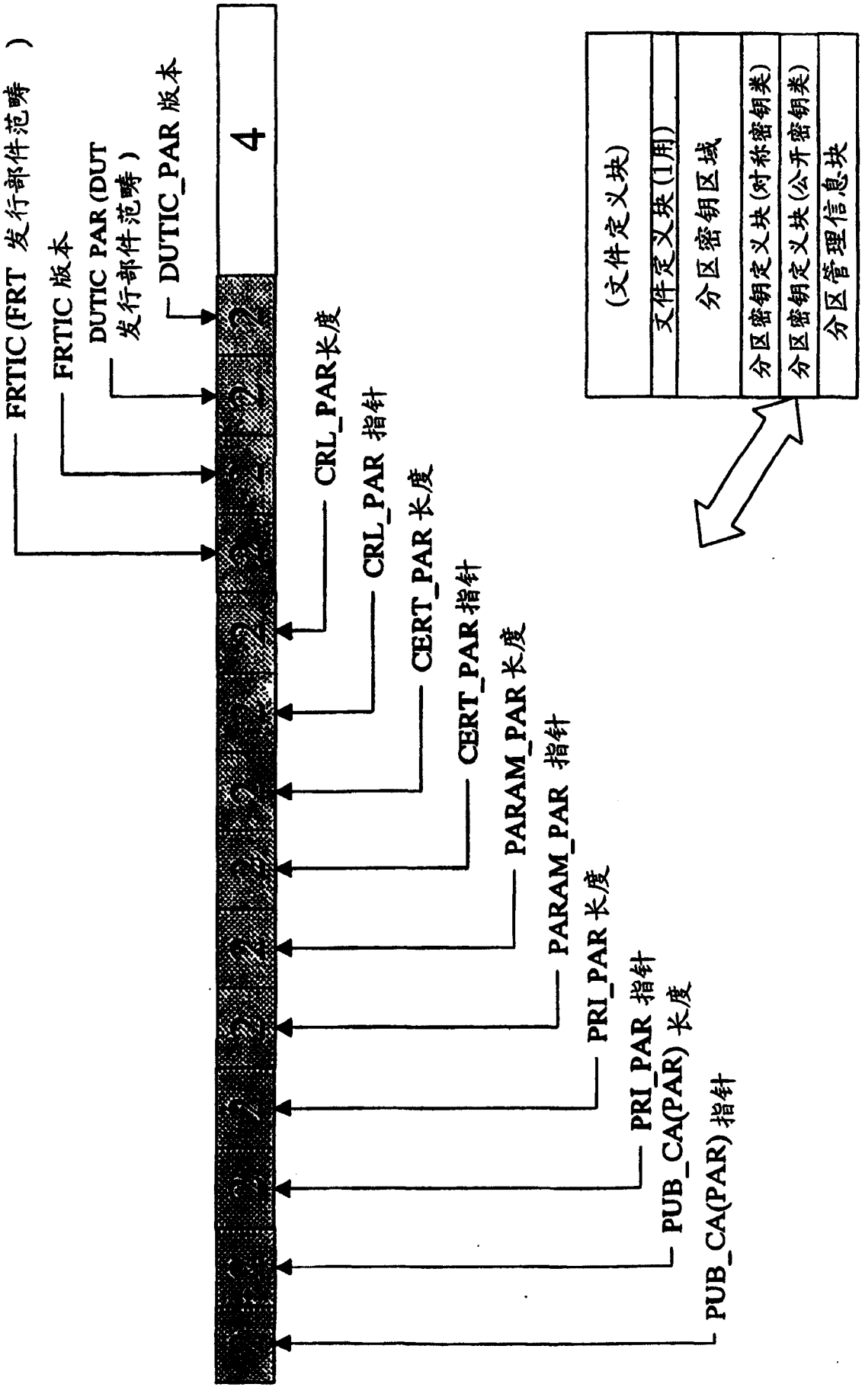

FRTIC (FRT 发行部件范畴)
FRTIC 版本
DUTIC PAR (DUT 发行部件范畴)
DUTIC_PAR 版本
4
CRL_PAR长度
CRL_PAR 指针
CERT_PAR 长度
CERT_PAR 指针
PARAM_PAR 长度
PARAM_PAR 指针
PRI_PAR 长度
PRI_PAR 指针
PUB_CA(PAR) 长度
PUB_CA(PAR) 指针
(文件定义块)
文件定义块(1用)
分区密钥区域
分区密钥定义块(对称密钥类)
分区密钥定义块(公开密钥类)
分区管理信息块

图 21

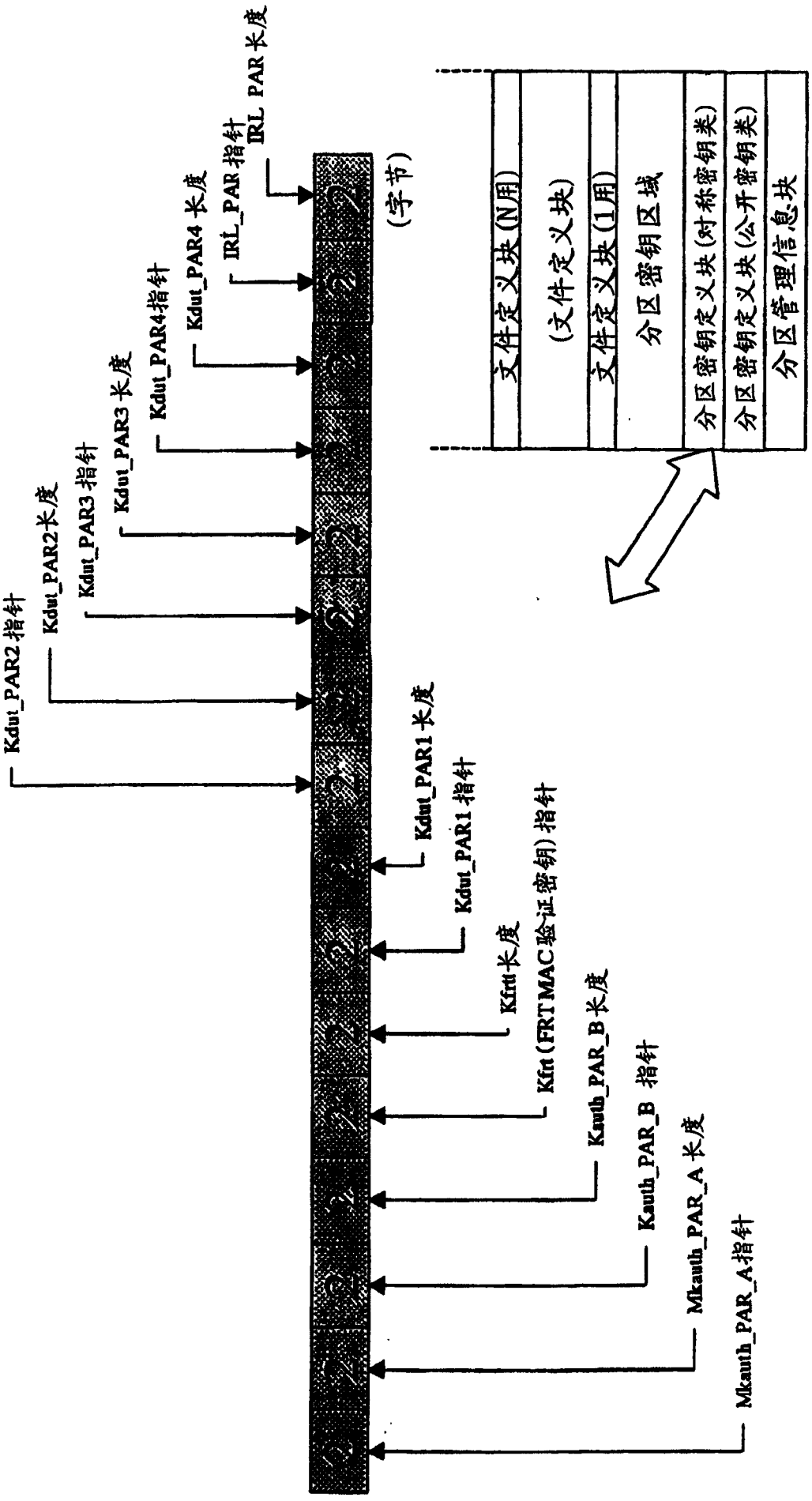

Kdut_PAR2指针
Kdut_PAR2长度
Kdut_PAR3指针
Kdut_PAR3长度
Kdut_PAR4指针
Kdut_PAR4长度
IRL_PAR指针
IRL PAR长度
(字节)
Kdut_PAR1长度
Kdut_PAR1指针
Kfrt长度
Kfrt(FRT MAC验证密钥)指针
Kauth_PAR_B长度
Kauth_PAR_B 指针
Mkauth_PAR_A长度
Mkauth_PAR_A指针
文件定义块(N用)
(文件定义块)
文件定义块(1用)
分区密钥区域
分区密钥定义块(对称密钥类)
分区密钥定义块(公开密钥类)
分区管理信息块

图 22

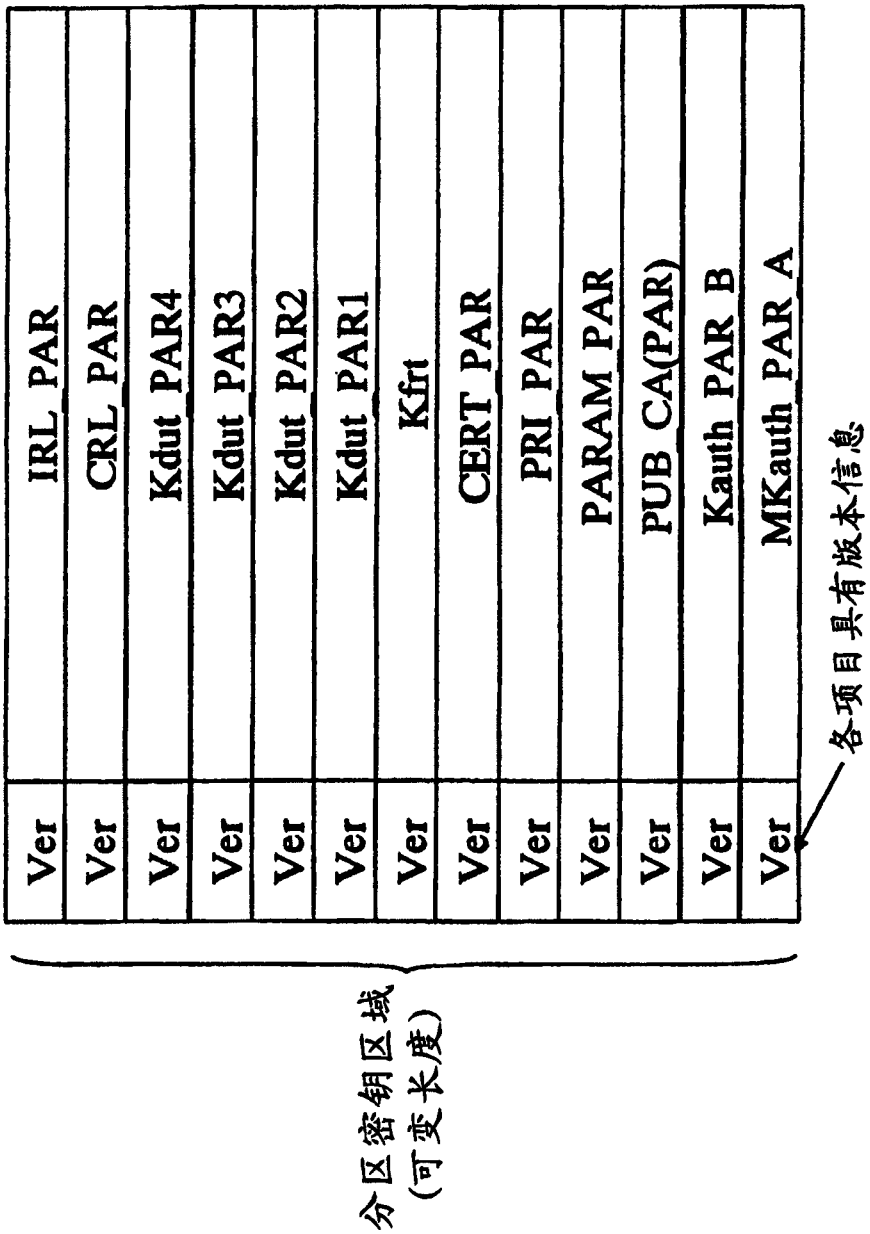

IRL_PAR
CRL_PAR
Kdut_PAR4
Kdut_PAR3
Kdut_PAR2
Kdut_PAR1
Kfrt
CERT_PAR
PRI_PAR
PARAM_PAR
PUB CA(PAR)
Kauth_PAR_B
MKauth_PAR_A
Ver
各项目具有版本信息
分区密钥区域
(可变长度)

图 23

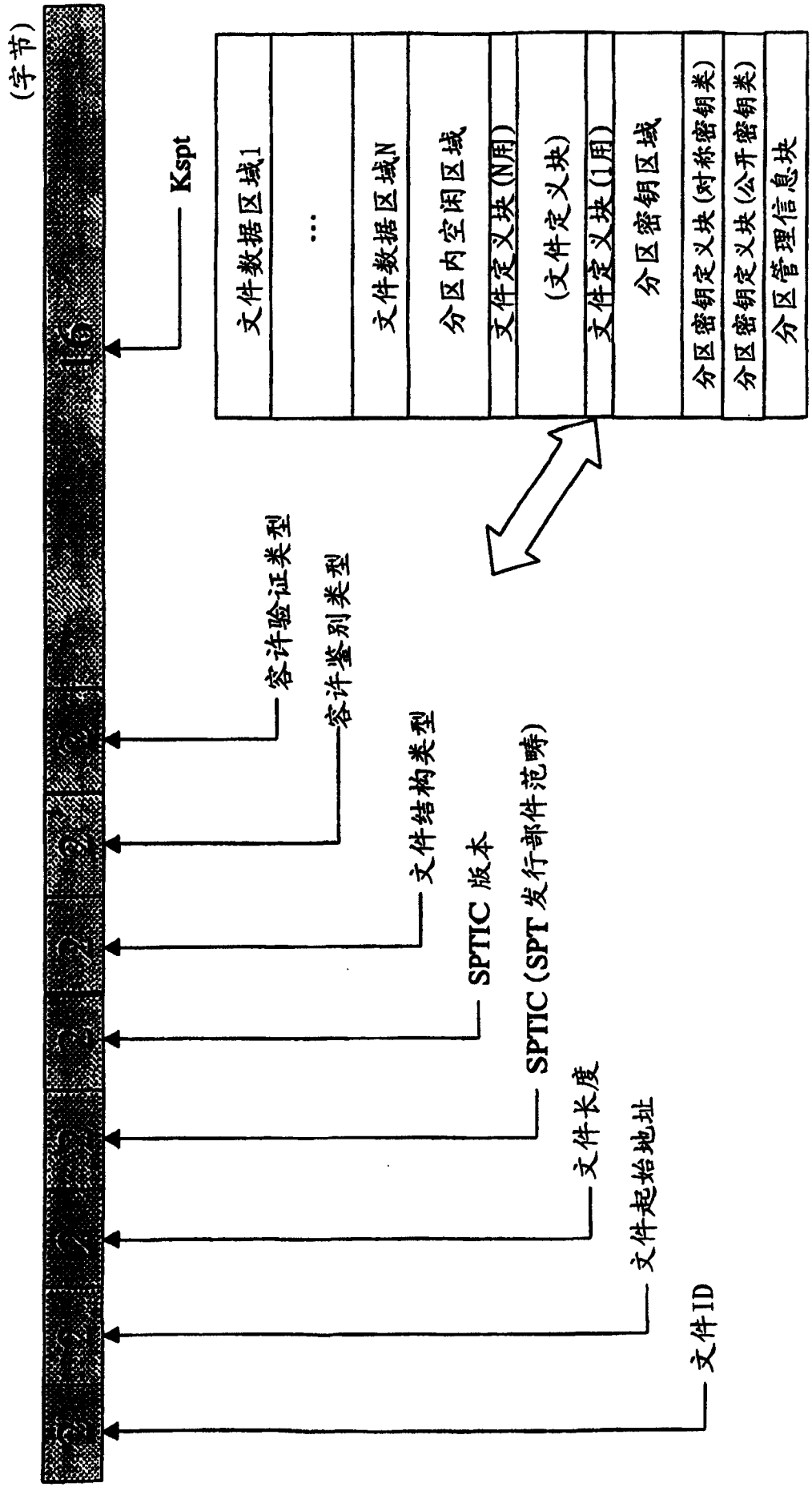
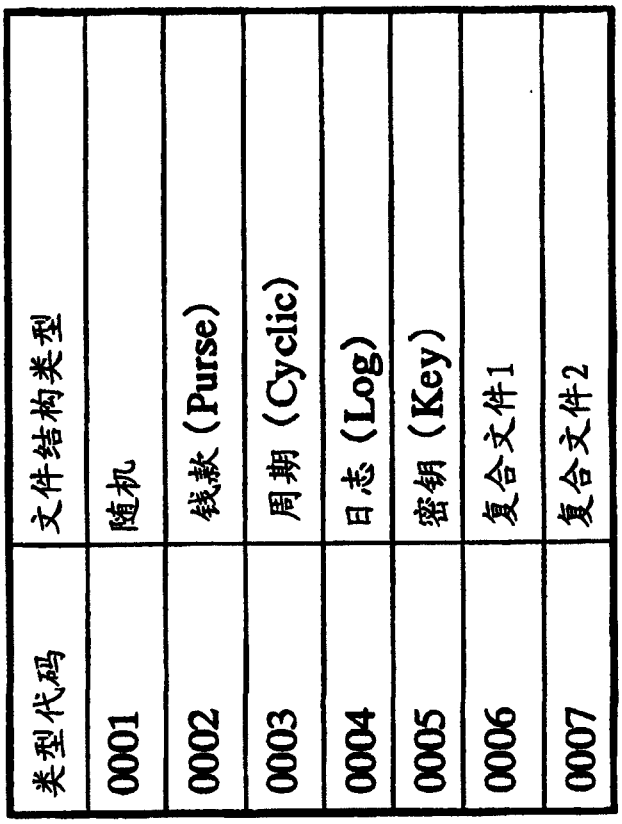

(字节)
Kspt
文件数据区域1
...
文件数据区域N
分区内空闲区域
文件定义块(N用)
(文件定义块)
文件定义块(1用)
分区密钥区域
分区密钥定义块(对称密钥类)
分区密钥定义块(公开密钥类)
分区管理信息块
容许验证类型
容许鉴别类型
文件结构类型
SPTIC 版本
SPTIC(SPT 发行部件范畴)
文件长度
文件起始地址
文件ID

图 24



类型代码	文件结构类型
0001	随机
0002	钱款（Purse）
0003	周期（Cyclic）
0004	日志（Log）
0005	密钥（Key）
0006	复合文件1
0007	复合文件2

图 25

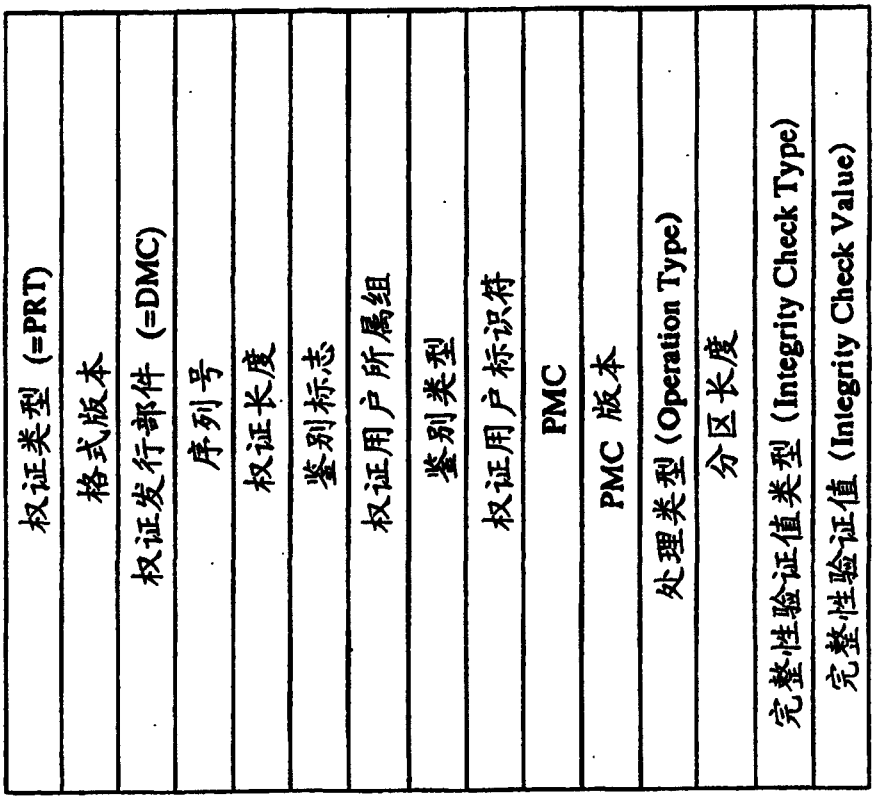

权证类型 (=PRT)
格式版本
权证发行部件 (=DMC)
序列号
权证长度
鉴别标志
权证用户所属组
鉴别类型
权证用户标识符
PMC
PMC 版本
处理类型 (Operation Type)
分区长度
完整性验证值类型 (Integrity Check Type)
完整性验证值 (Integrity Check Value)

图 26

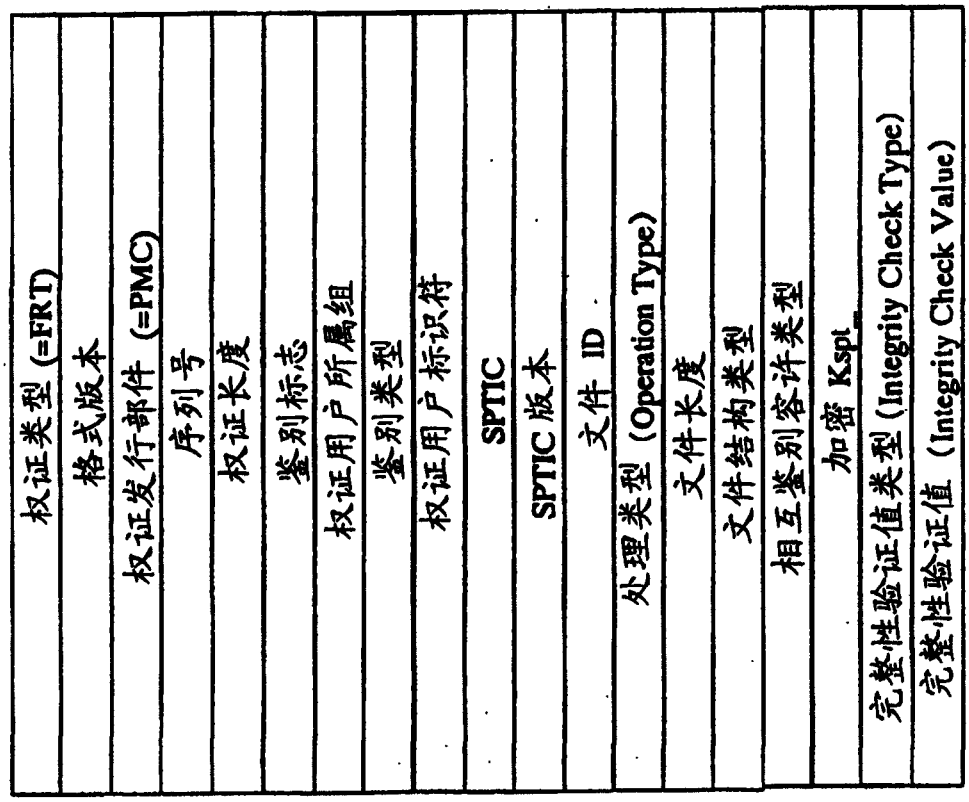

权证类型 (=FRT)
格式版本
权证发行部件 (=PMC)
序列号
权证长度
鉴别标志
权证用户所属组
鉴别类型
权证用户标识符
SPTIC
SPTIC 版本
文件 ID
处理类型 (Operation Type)
文件长度
文件结构类型
相互鉴别容许类型
加密 Kspt
完整性验证值类型 (Integrity Check Type)
完整性验证值 (Integrity Check Value)

图 27

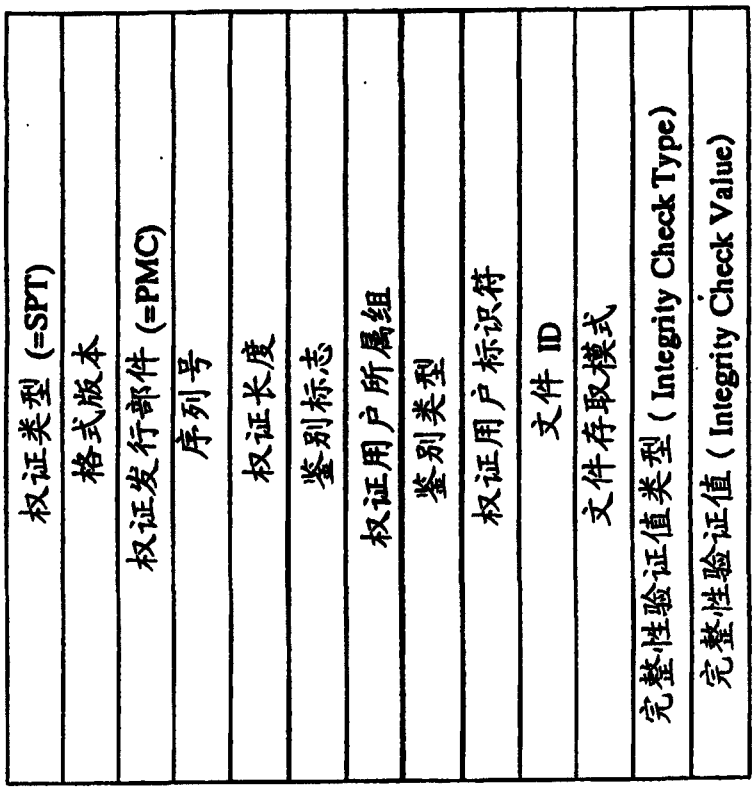

权证类型 (=SPT)
格式版本
权证发行部件 (=PMC)
序列号
权证长度
鉴别标志
权证用户所属组
鉴别类型
权证用户标识符
文件 ID
文件存取模式
完整性验证值类型 (Integrity Check Type)
完整性验证值 (Integrity Check Value)
一对儿

图 28

存取模式代码	存取模式	存取模式代码	存取模式	存取模式代码	存取模式
0001	读取 (Read)	0011	删除	0021	注册证书
0002	写入 (Write)	0012	增加(Add)	0022	验证证书
0003	加密 读取 (Read)	0013	减少(Sub)	0023	生成密钥对
0004	加密 写入 (Write)	0014	比较(Compare)	0024	验证密钥对
0005	带MAC读取 (Read)	0015	加密	0025	相互鉴别
0006	带MAC写入 (Write)	0016	解密	0026	存款类
0007	带MAC加密 Read	0017	生成签名	0027	取款类
0008	带MAC加密 Write	0018	验证签名	0028	
0009	只写	0019	生成MAC	0029	
0010	写入1次	0020	验证MAC	0030	

图 29

文件结构	存取模式	容许命令	文件结构	存取模式	容许命令
随机	Read	Read	复合文件 (电子货币)	存款类	Deposit (存入)
	Write	Write			
	加密 Read	EncRead		取款类	Withdraw (取出) Make Receipt (生成收据) Read Receipt (读出收据)
	加密 Write	EncWrite			
	带MAC Read	MacRead			
	带MAC Write	MacWrite			
	带MAC加密 Read	EncMacRead			
	带MAC加密 Write	EncMacWrite			

图 30

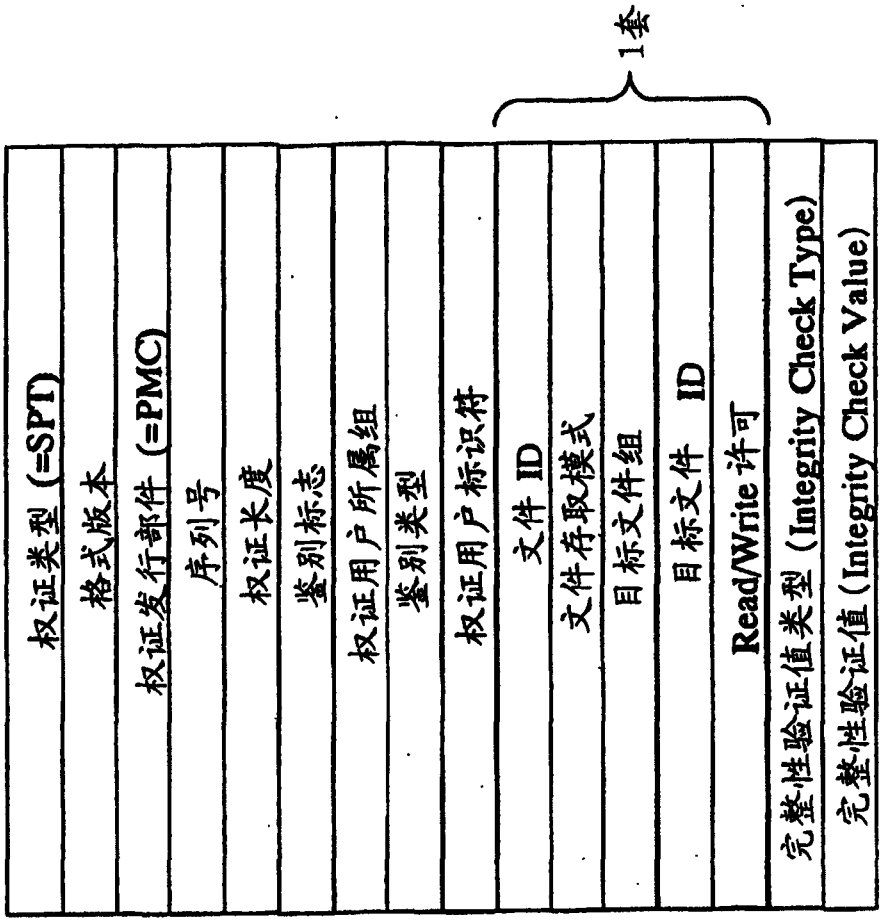

权证类型 (=SPT)
格式版本
权证发行部件 (=PMC)
序列号
权证长度
鉴别标志
权证用户所属组
鉴别类型
权证用户标识符
文件 ID
文件存取模式
目标文件组
目标文件 ID
Read/Write 许可
完整性验证值类型 (Integrity Check Type)
完整性验证值 (Integrity Check Value)
1套

图 31

权证类型 (=DUT(DEV))
格式版本
权证发行部件 (=DMC)
序列号
权证长度
权证用户所属组
权证用户标识符
鉴别类型
加密标志
旧数据代码
数据更新版本条件 (Data Version Rule)
数据更新版本值 (Data Version Condition)
更新数据长度 (Size of New Data)
更新数据 (New Data)
更新数据版本 (New Data Version)
完整性验证值类型 (Integrity Check Type)
完整性验证值 (Integrity Check Value)
DUT(DEV)
权证类型 (=DUT(PAR))
格式版本
权证发行部件 (=PMC)
序列号
权证长度
权证用户所属组
权证用户标识符
鉴别类型
加密标志
旧数据代码
数据更新版本条件 (Data Version Rule)
数据更新版本值 (Data Version Condition)
更新数据长度 (Size of New Data)
更新数据 (New Data)
更新数据版本 (New Data Version)
完整性验证值类型 (Integrity Check Type)
完整性验证值 (Integrity Check Value)
DUT(PAR)

图 32

代码	更新数据（New Data）	代码	更新数据（New Data）	代码	更新数据（New Data）
0001	DMC	0011	DUTIC_PAR	0021	MKauth_DEV_A
0002	DMC 版本	0012	DUTIC_PAR 版本	0022	Kauth_DEV_B
0003	PMC	0013	Kdut_DEV1,2	0023	IRL_DEV
0004	PMC 版本	0014	Kdut_PAR1,2	0024	IRL_PAR
0005	PRTIC	0015	Kprt	0025	
0006	PRTIC版本	0016	Kprt 版本	0026	
0007	FRTIC	0017	Kfrt	0027	
0008	FRTIC 版本	0018	Kfrt 版本	0028	
0009	DUTIC_DEV	0019	Kspt	0029	
0010	DUTIC_DEV 版本	0020	Kspt 版本	0030	

图 33

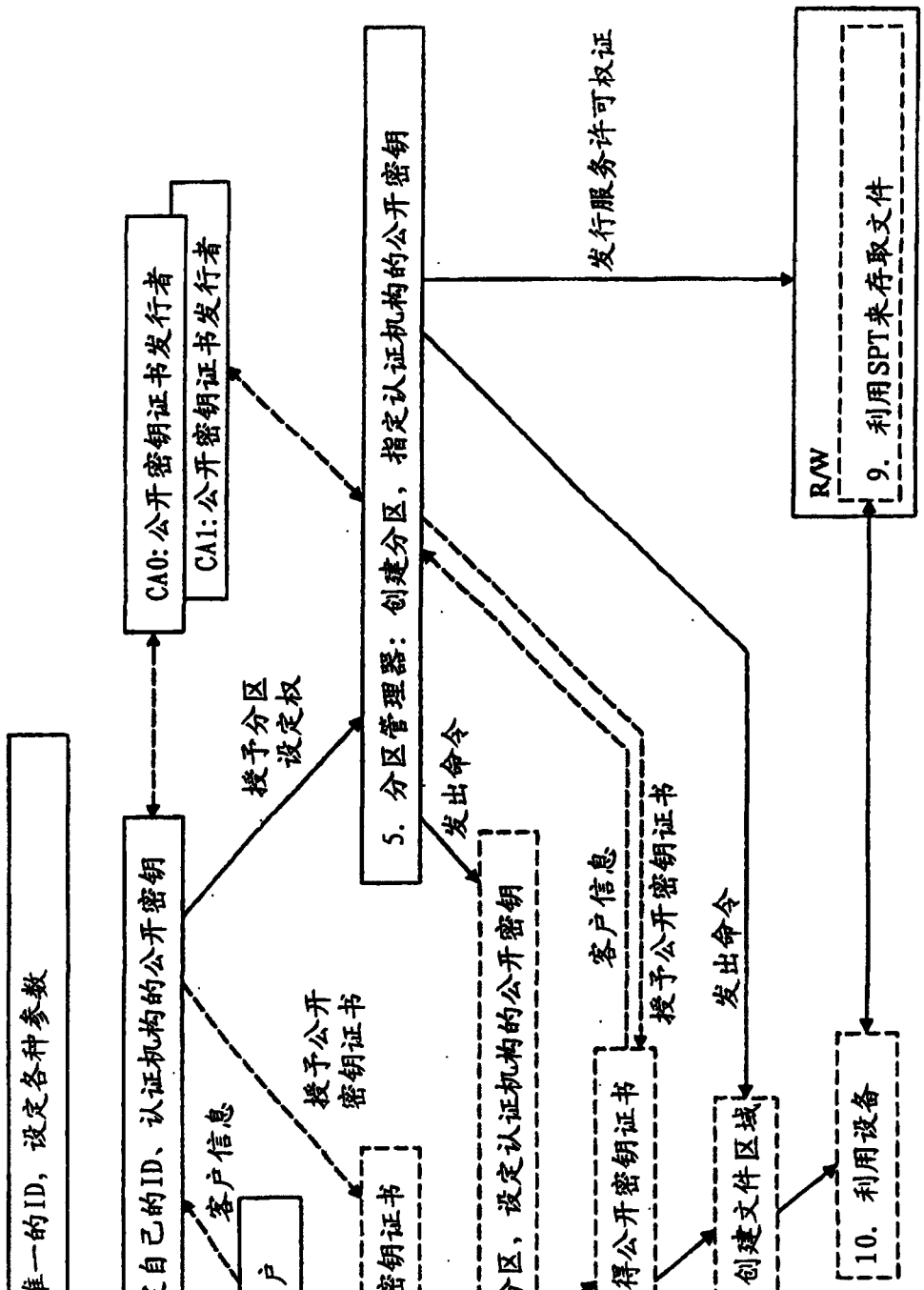

1. 生产者：授予设备唯一的ID，设定各种参数
销售
2. 设备管理器：设定自己的ID、认证机构的公开密钥
租借/销售
3. 客户：设备用户
客户信息
CA0:公开密钥证书发行者
CA1:公开密钥证书发行者
授予分区设定权
授予公开密钥证书
4. 取得公开密钥证书
5. 分区管理器：创建分区，指定认证机构的公开密钥
发出命令
6. 保留分区，设定认证机构的公开密钥
客户信息
7. 取得公开密钥证书
授予公开密钥证书
发出命令
8. 创建文件区域
10. 利用设备
发行服务许可权证
R/W
9. 利用SPT来存取文件

图 34

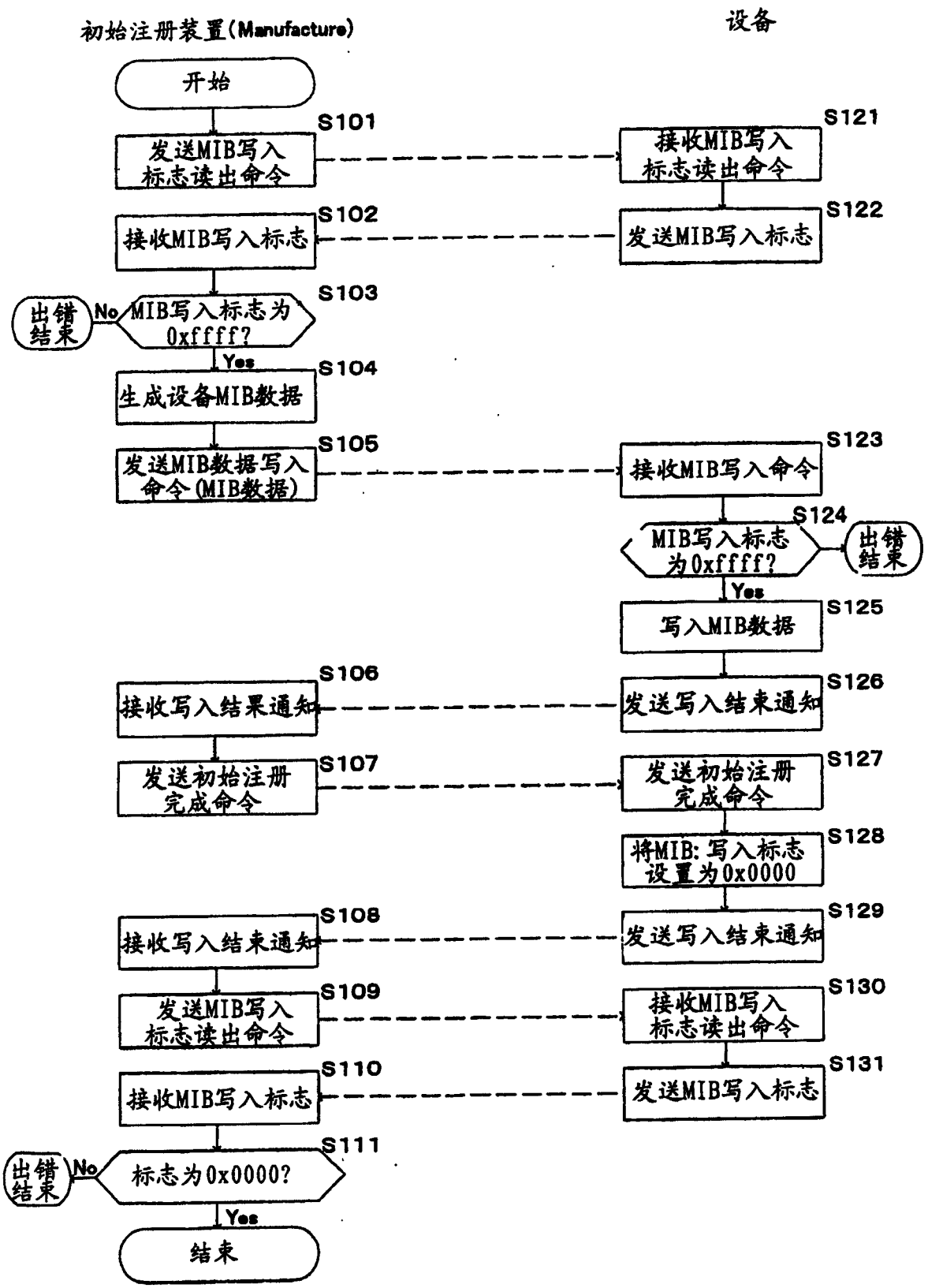

初始注册装置(Manufacture)
设备
开始
S101
发送MIB写入标志读出命令
S121
接收MIB写入标志读出命令
S102
接收MIB写入标志
S122
发送MIB写入标志
S103
出错结束
No
MIB写入标志为0xffff?
Yes
S104
生成设备MIB数据
S105
发送MIB数据写入命令(MIB数据)
S123
接收MIB写入命令
S124
MIB写入标志为0xffff?
出错结束
Yes
S125
写入MIB数据
S106
接收写入结果通知
S126
发送写入结束通知
S107
发送初始注册完成命令
S127
发送初始注册完成命令
S128
将MIB:写入标志设置为0x0000
S108
接收写入结束通知
S129
发送写入结束通知
S109
发送MIB写入标志读出命令
S130
接收MIB写入标志读出命令
S110
接收MIB写入标志
S131
发送MIB写入标志
S111
出错结束
No
标志为0x0000?
Yes
结束

图 35

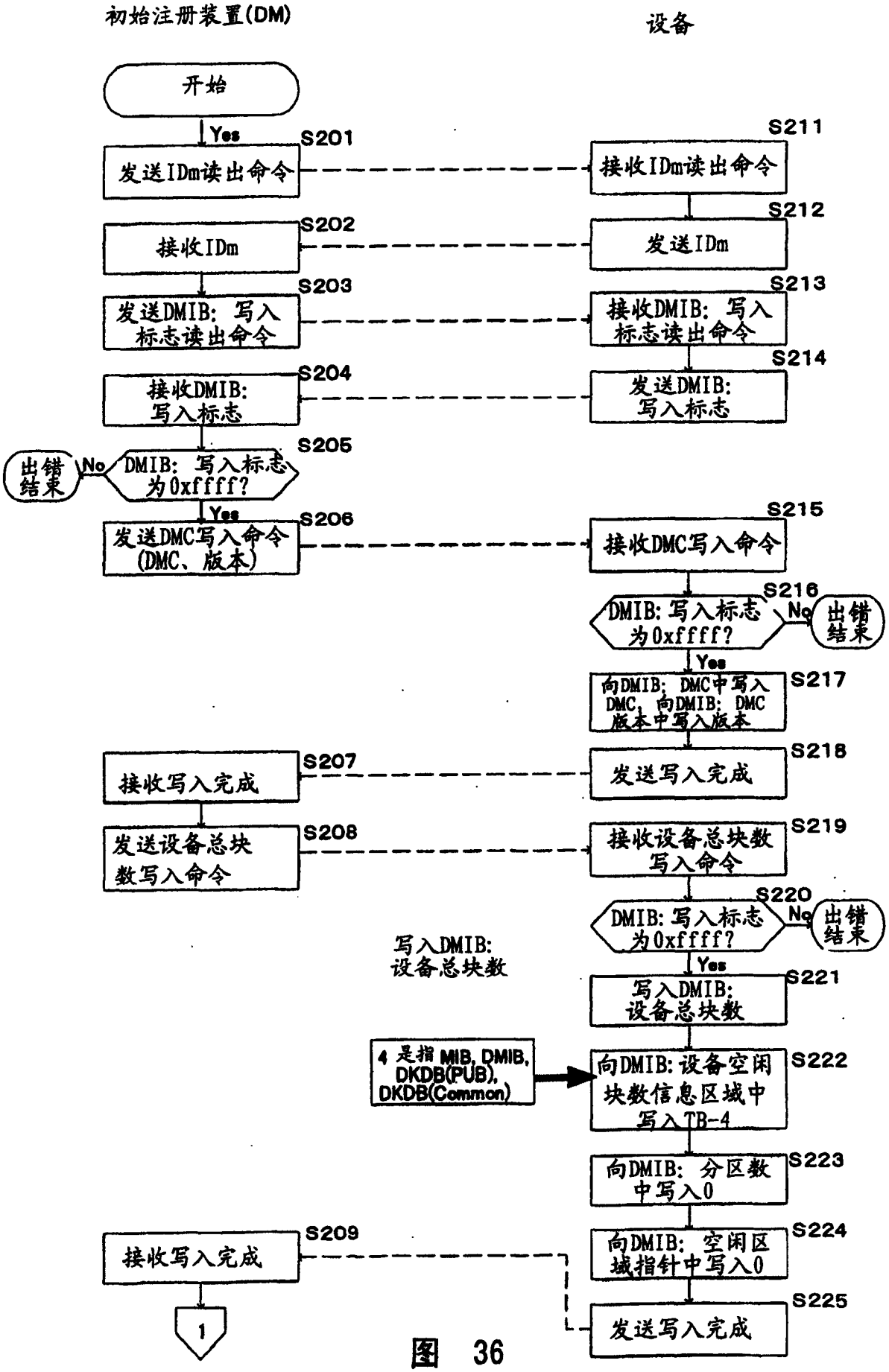

初始注册装置(DM)
设备
开始
Yes
S201
发送IDm读出命令
S211
接收IDm读出命令
S202
接收IDm
S212
发送IDm
S203
发送DMIB: 写入标志读出命令
S213
接收DMIB: 写入标志读出命令
S204
接收DMIB: 写入标志
S214
发送DMIB: 写入标志
S205
出错结束
No
DMIB: 写入标志为0xffff?
Yes
S206
发送DMC写入命令(DMC、版本)
S215
接收DMC写入命令
S216
DMIB: 写入标志为0xffff?
No
出错结束
Yes
向DMIB: DMC中写入DMC, 向DMIB: DMC版本中写入版本
S217
S207
接收写入完成
S218
发送写入完成
S208
发送设备总块数写入命令
S219
接收设备总块数写入命令
S220
DMIB: 写入标志为0xffff?
No
出错结束
写入DMIB: 设备总块数
Yes
S221
写入DMIB: 设备总块数
4 是指 MIB, DMIB, DKDB(PUB), DKDB(Common)
S222
向DMIB: 设备空闲块数信息区域中写入TB-4
S223
向DMIB: 分区数中写入0
S209
接收写入完成
S224
向DMIB: 空闲区域指针中写入0
S225
发送写入完成
1

图 36

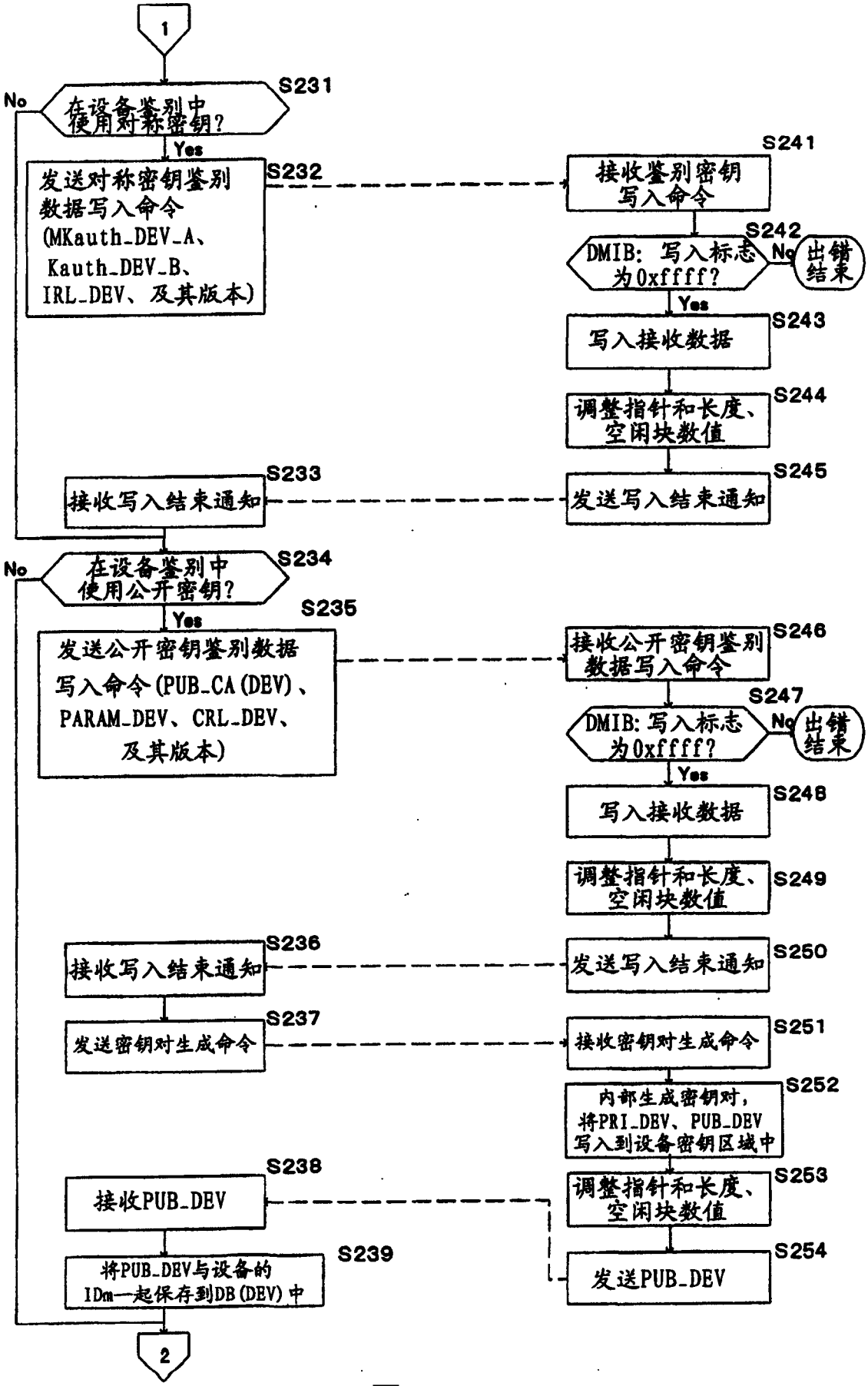

1
S231
No
在设备鉴别中使用对称密钥?
Yes
S232
发送对称密钥鉴别数据写入命令(MKauth_DEV_A、Kauth_DEV_B、IRL_DEV、及其版本)
S241
接收鉴别密钥写入命令
S242
DMIB: 写入标志为0xffff?
No
出错结束
Yes
S243
写入接收数据
S244
调整指针和长度、空闲块数值
S245
发送写入结束通知
S233
接收写入结束通知
S234
No
在设备鉴别中使用公开密钥?
Yes
S235
发送公开密钥鉴别数据写入命令(PUB_CA(DEV)、PARAM_DEV、CRL_DEV、及其版本)
S246
接收公开密钥鉴别数据写入命令
S247
DMIB: 写入标志为0xffff?
No
出错结束
Yes
S248
写入接收数据
S249
调整指针和长度、空闲块数值
S250
发送写入结束通知
S236
接收写入结束通知
S237
发送密钥对生成命令
S251
接收密钥对生成命令
S252
内部生成密钥对,将PRI_DEV、PUB_DEV写入到设备密钥区域中
S253
调整指针和长度、空闲块数值
S254
发送PUB_DEV
S238
接收PUB_DEV
S239
将PUB_DEV与设备的IDm一起保存到DB(DEV)中
2

图 37

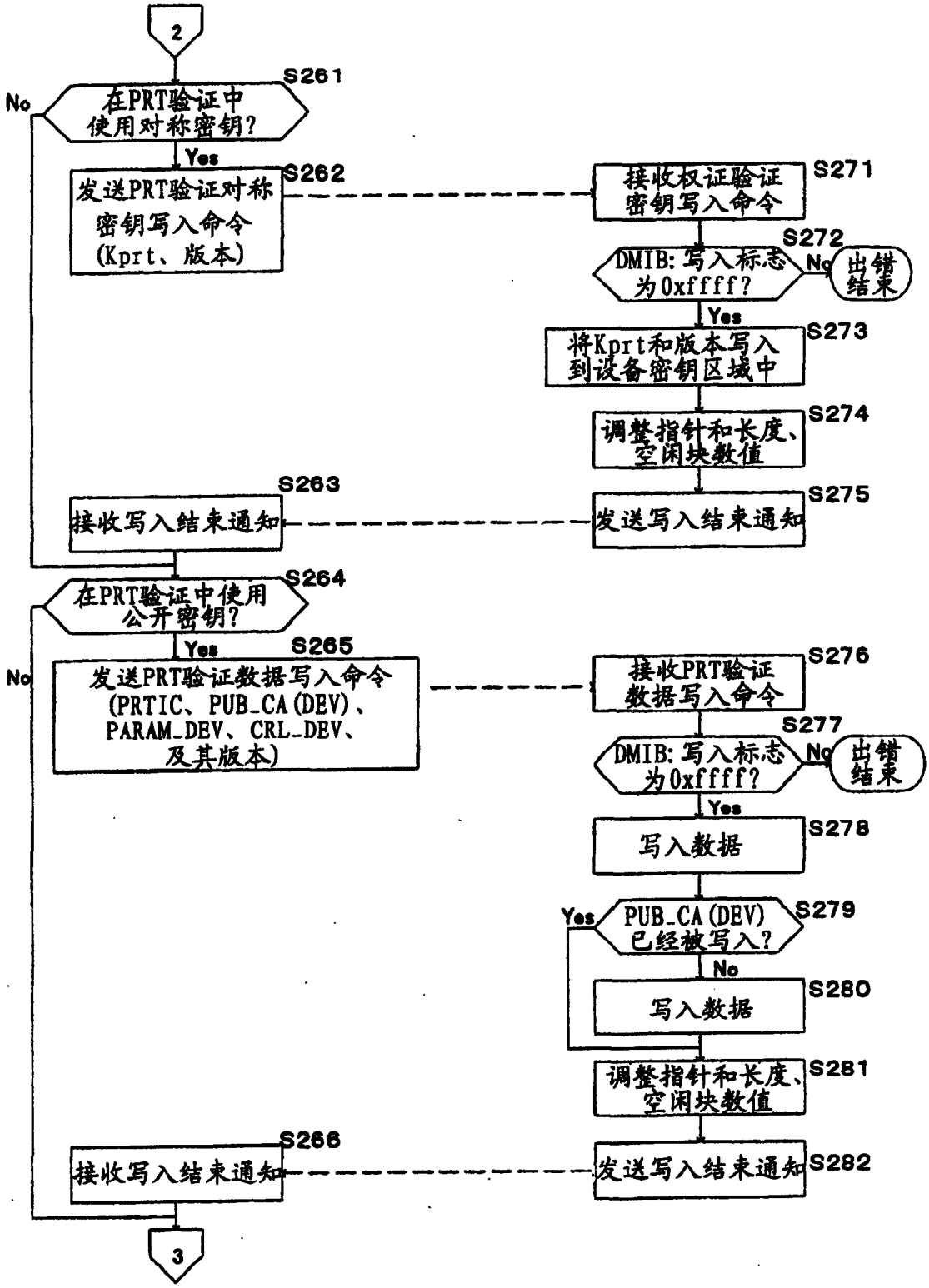

2
S261
No
在PRT验证中使用对称密钥?
Yes
S262
发送PRT验证对称密钥写入命令(Kprt、版本)
S271
接收权证验证密钥写入命令
S272
DMIB:写入标志为0xffff?
No
出错结束
Yes
S273
将Kprt和版本写入到设备密钥区域中
S274
调整指针和长度、空闲块数值
S275
发送写入结束通知
S263
接收写入结束通知
S264
在PRT验证中使用公开密钥?
Yes
S265
No
发送PRT验证数据写入命令(PRTIC、PUB_CA(DEV)、PARAM_DEV、CRL_DEV、及其版本)
S276
接收PRT验证数据写入命令
S277
DMIB:写入标志为0xffff?
No
出错结束
Yes
S278
写入数据
Yes
S279
PUB_CA(DEV)已经被写入?
No
S280
写入数据
S281
调整指针和长度、空闲块数值
S282
发送写入结束通知
S266
接收写入结束通知
3

图 38

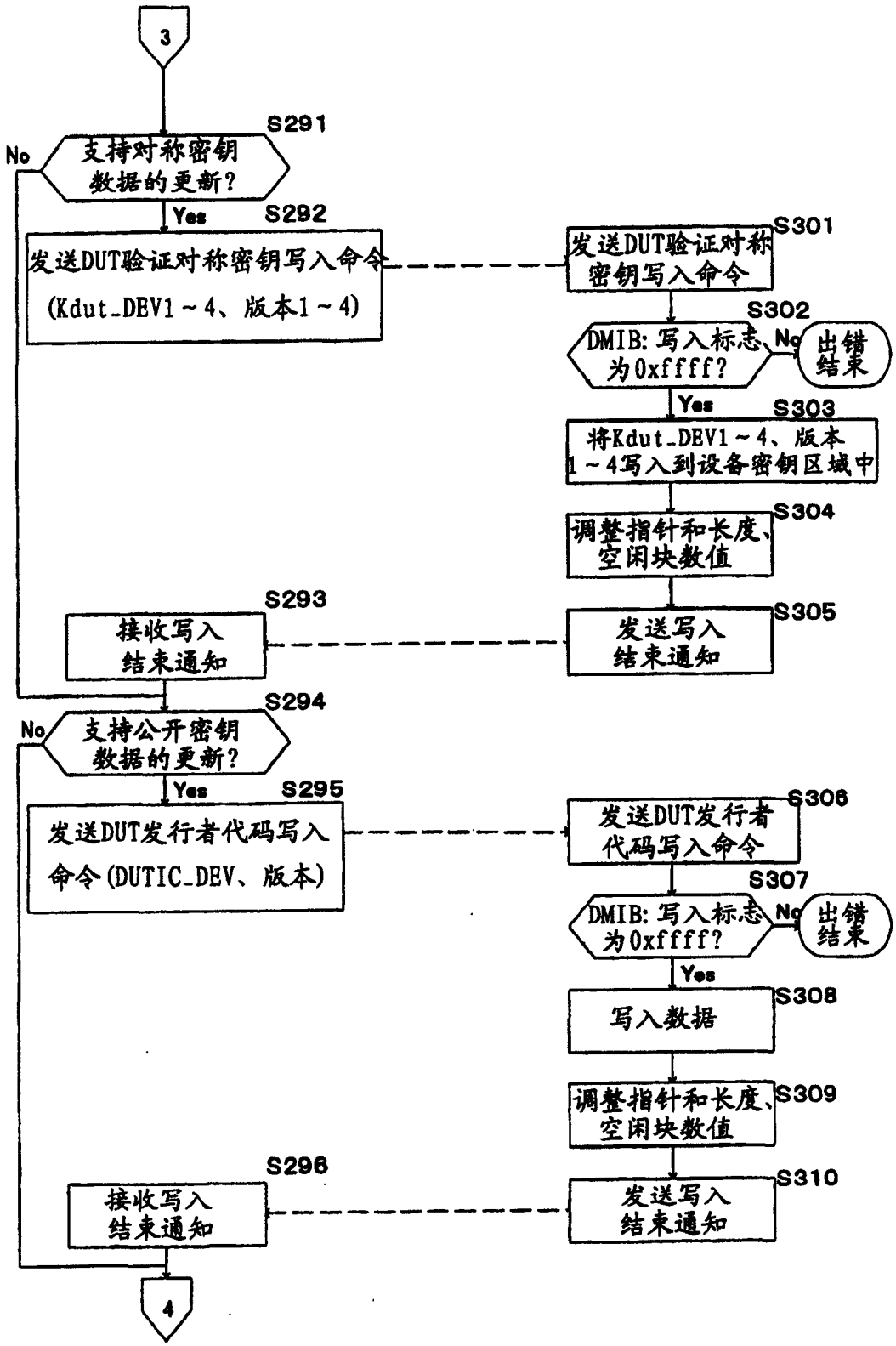

3
S291
No
支持对称密钥数据的更新?
Yes
S292
发送DUT验证对称密钥写入命令(Kdut_DEV1～4、版本1～4)
S301
发送DUT验证对称密钥写入命令
S302
DMIB: 写入标志为0xffff?
No
出错结束
Yes
S303
将Kdut_DEV1～4、版本1～4写入到设备密钥区域中
S304
调整指针和长度、空闲块数值
S305
发送写入结束通知
S293
接收写入结束通知
S294
No
支持公开密钥数据的更新?
Yes
S295
发送DUT发行者代码写入命令(DUTIC_DEV、版本)
S306
发送DUT发行者代码写入命令
S307
DMIB: 写入标志为0xffff?
No
出错结束
Yes
S308
写入数据
S309
调整指针和长度、空闲块数值
S310
发送写入结束通知
S296
接收写入结束通知
4

图 39

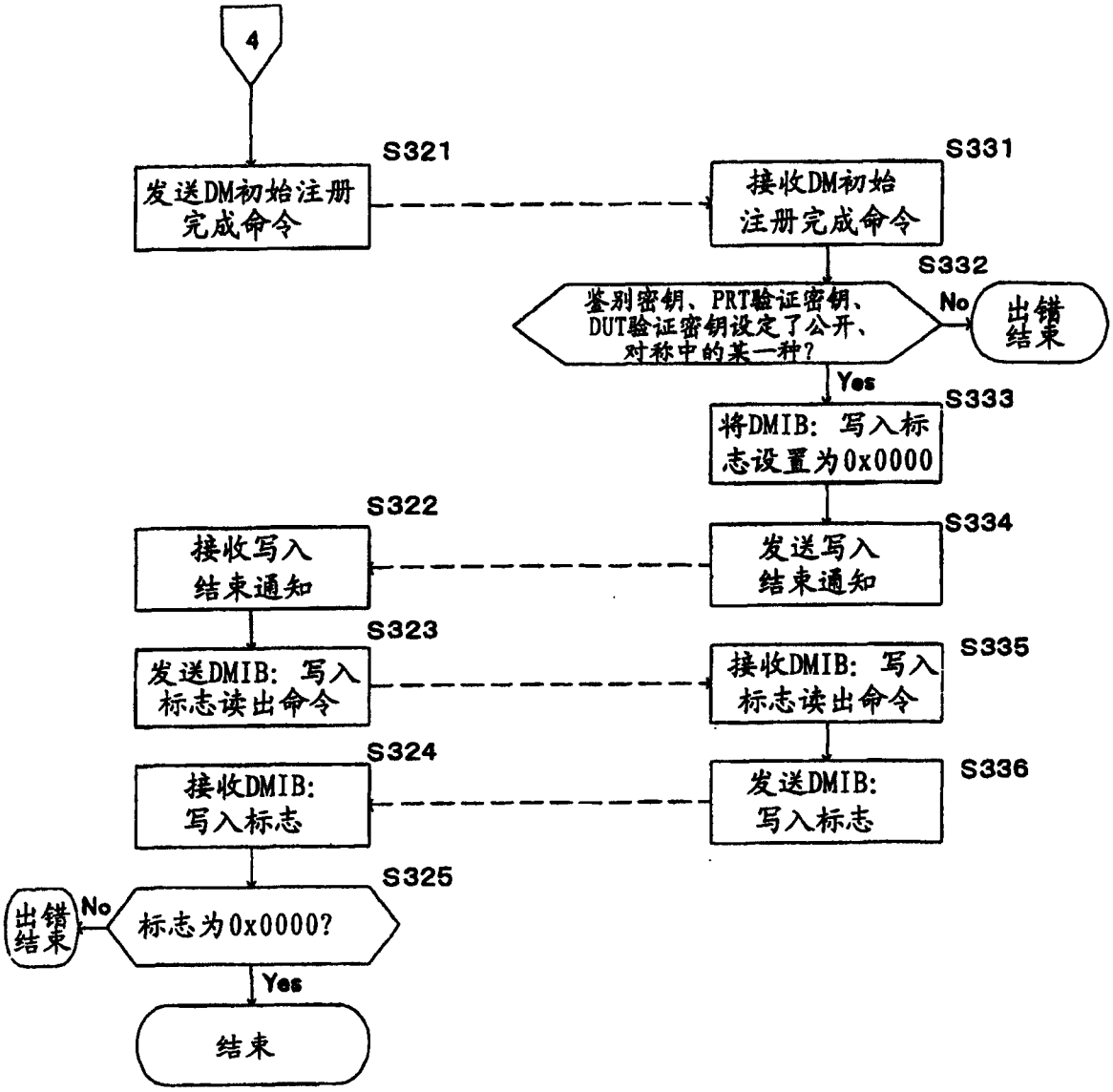

4
S321
发送DM初始注册完成命令
S331
接收DM初始注册完成命令
S332
鉴别密钥、PRT验证密钥、DUT验证密钥设定了公开、对称中的某一种?
No
出错结束
Yes
S333
将DMIB: 写入标志设置为0x0000
S322
接收写入结束通知
S334
发送写入结束通知
S323
发送DMIB: 写入标志读出命令
S335
接收DMIB: 写入标志读出命令
S324
接收DMIB: 写入标志
S336
发送DMIB: 写入标志
S325
出错结束
No
标志为0x0000?
Yes
结束

图 40

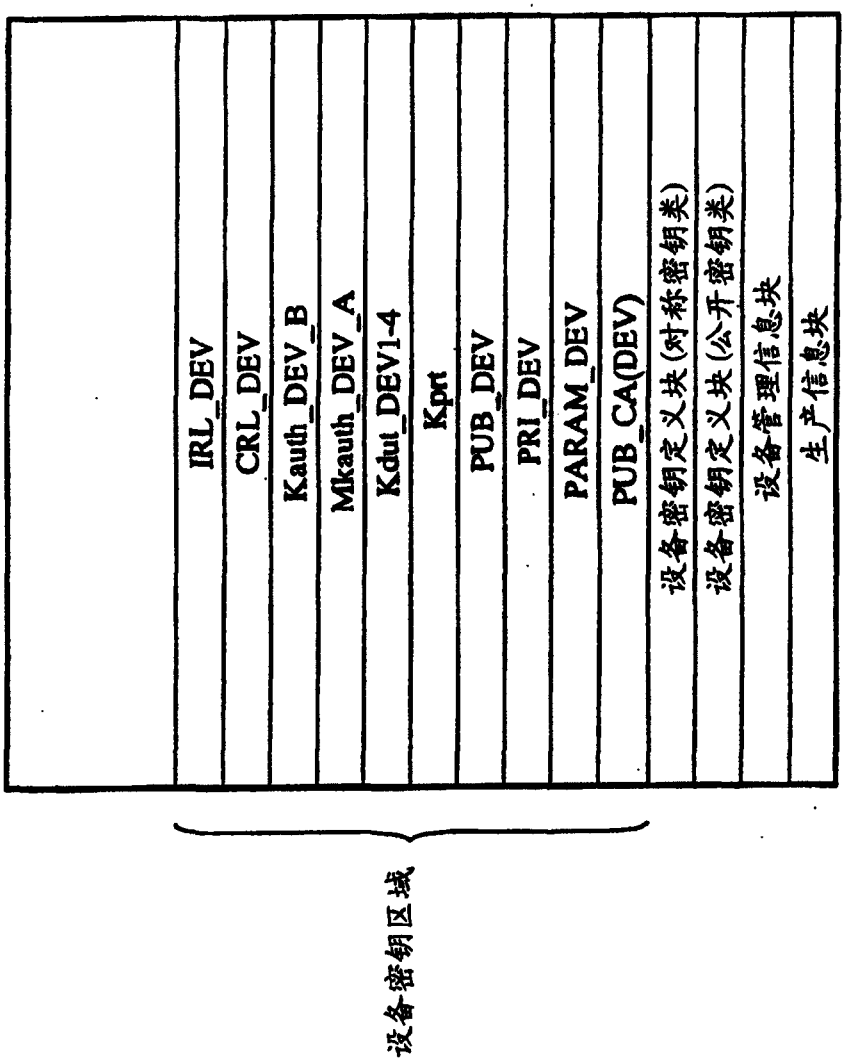

IRL_DEV
CRL_DEV
Kauth_DEV_B
Mkauth_DEV_A
Kdui_DEV1-4
Kprt
PUB_DEV
PRI_DEV
PARAM_DEV
PUB_CA(DEV)
设备密钥定义块(对称密钥类)
设备密钥定义块(公开密钥类)
设备管理信息块
生产信息块
设备密钥区域

图 41

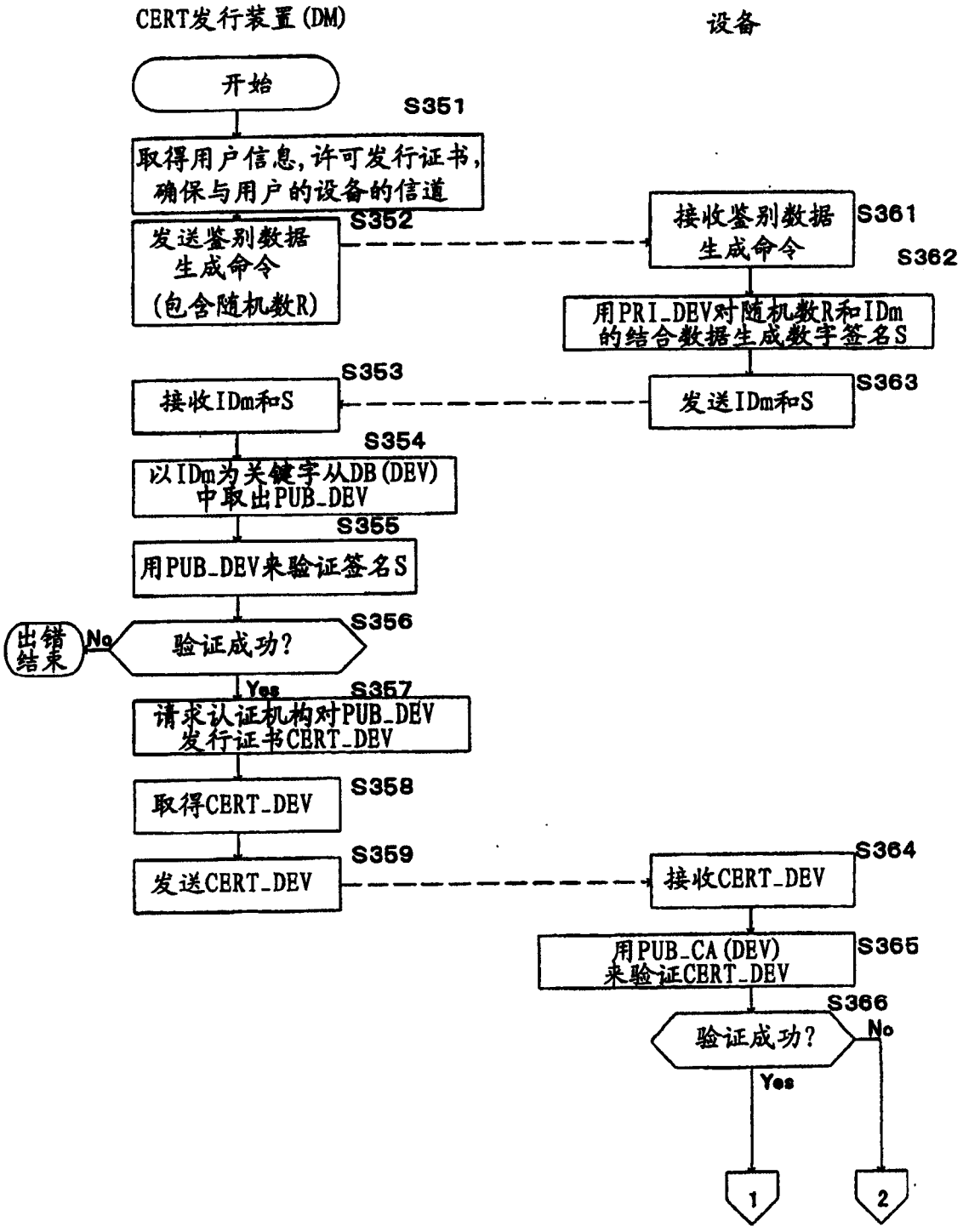

CERT发行装置(DM)
设备
开始
S351
取得用户信息,许可发行证书,
确保与用户的设备的信道
S352
发送鉴别数据
生成命令
(包含随机数R)
接收鉴别数据
生成命令
S361
S362
用PRI_DEV对随机数R和IDm
的结合数据生成数字签名S
S363
发送IDm和S
S353
接收IDm和S
S354
以IDm为关键字从DB(DEV)
中取出PUB_DEV
S355
用PUB_DEV来验证签名S
S356
验证成功?
No
出错
结束
Yes
S357
请求认证机构对PUB_DEV
发行证书CERT_DEV
S358
取得CERT_DEV
S359
发送CERT_DEV
S364
接收CERT_DEV
S365
用PUB_CA(DEV)
来验证CERT_DEV
S366
验证成功?
No
Yes
1
2

图 42

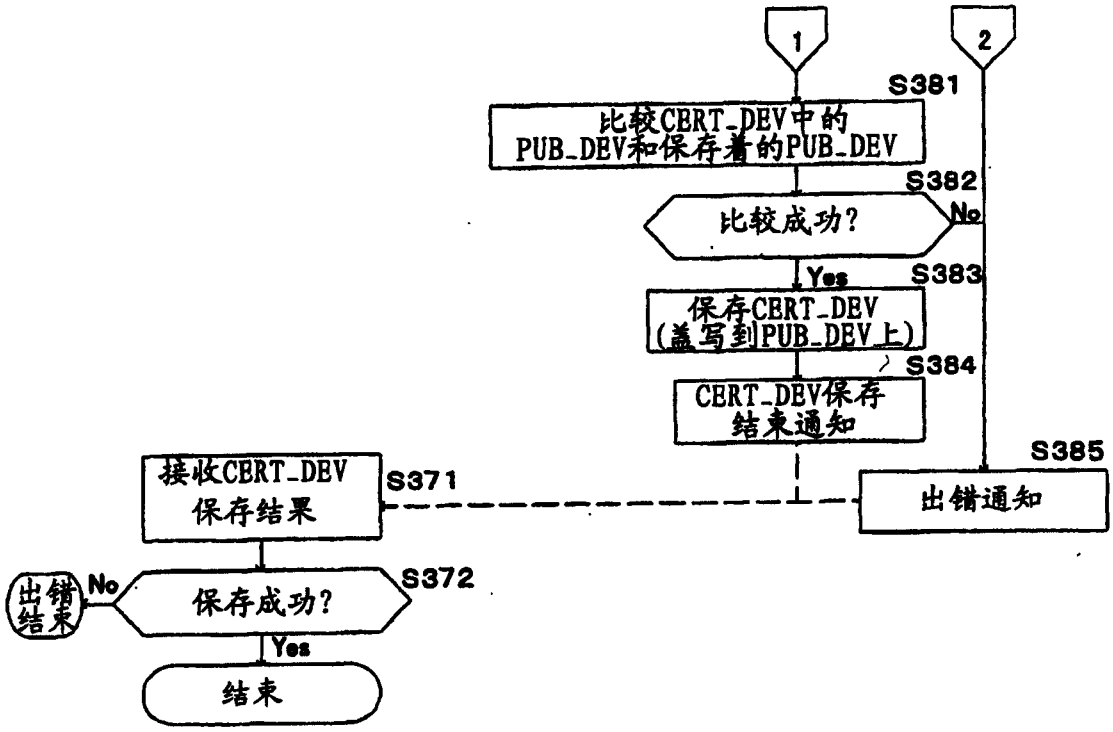

1
2
S381
比较CERT_DEV中的
PUB_DEV和保存着的PUB_DEV
S382
比较成功?
No
Yes
S383
保存CERT_DEV
(盖写到PUB_DEV上)
S384
CERT_DEV保存
结束通知
S385
出错通知
接收CERT_DEV
保存结果
S371
出错
结束
No
保存成功?
S372
Yes
结束

图 43

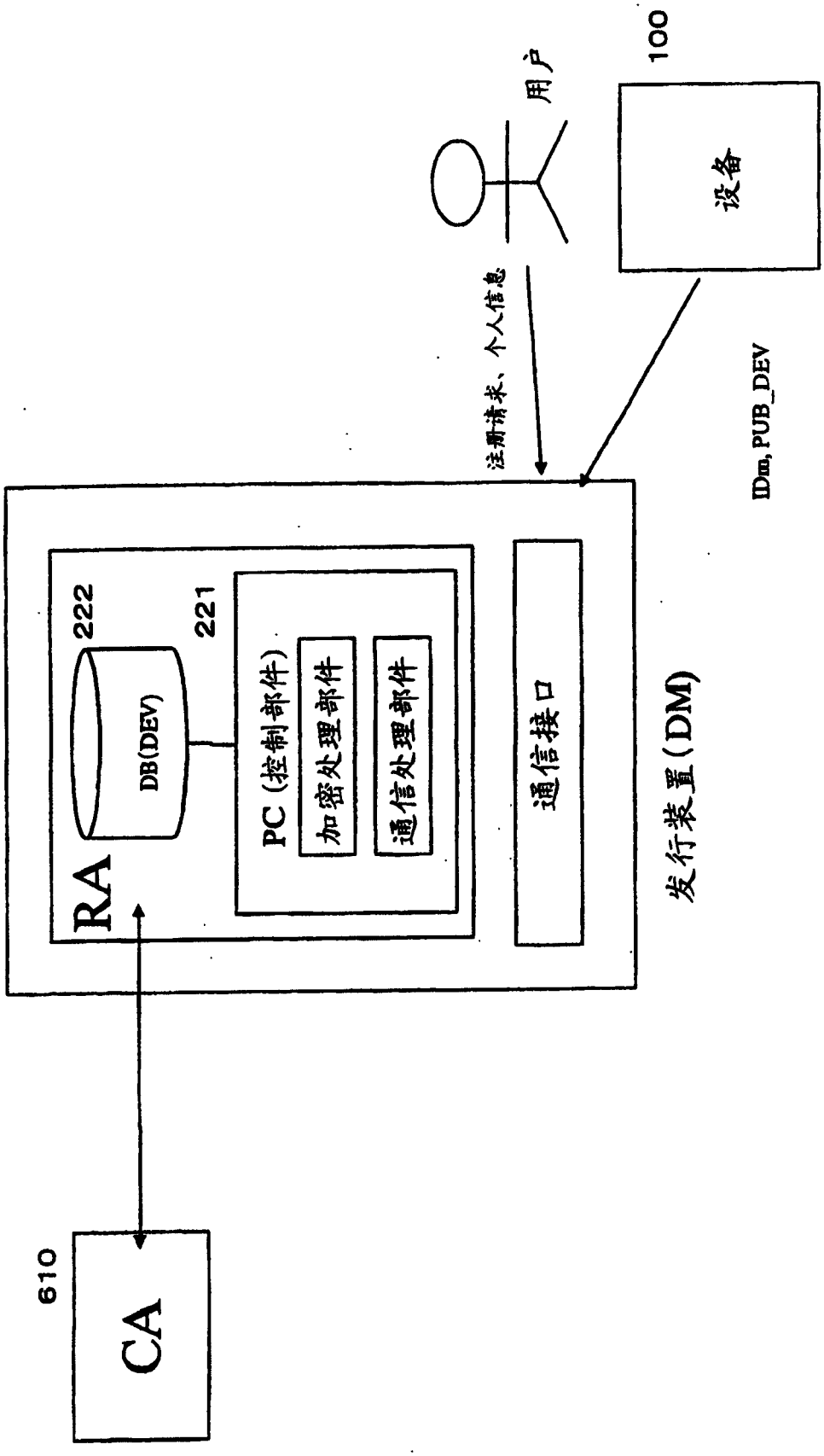

100
用户
设备
注册请求、个人信息
IDm, PUB_DEV
发行装置(DM)
RA
222
DB(DEV)
221
PC (控制部件)
加密处理部件
通信处理部件
通信接口
610
CA

图 44

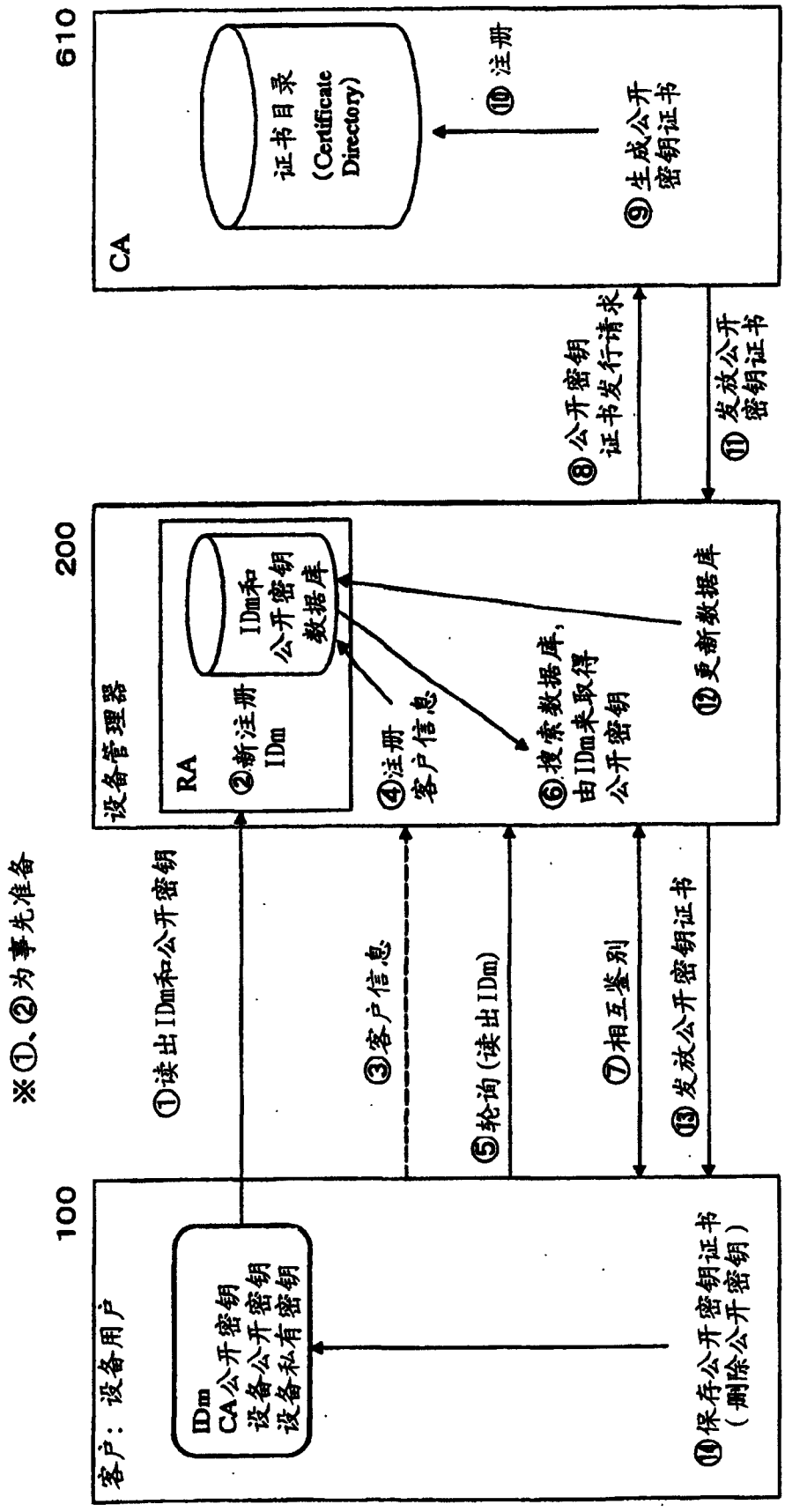

610
CA
证书目录
(Certificate Directory)
⑩注册
⑨生成公开密钥证书
⑧公开密钥证书发行请求
⑪发放公开密钥证书
200
设备管理器
RA
②新注册IDm
IDm和公开密钥数据库
④注册客户信息
⑥搜索数据库，由IDm来取得公开密钥
⑫更新数据库
※①、②为事先准备
①读出IDm和公开密钥
③客户信息
⑤轮询(读出IDm)
⑦相互鉴别
⑬发放公开密钥证书
100
客户：设备用户
IDm
CA公开密钥
设备公开密钥
设备私有密钥
⑭保存公开密钥证书
(删除公开密钥)

图 45

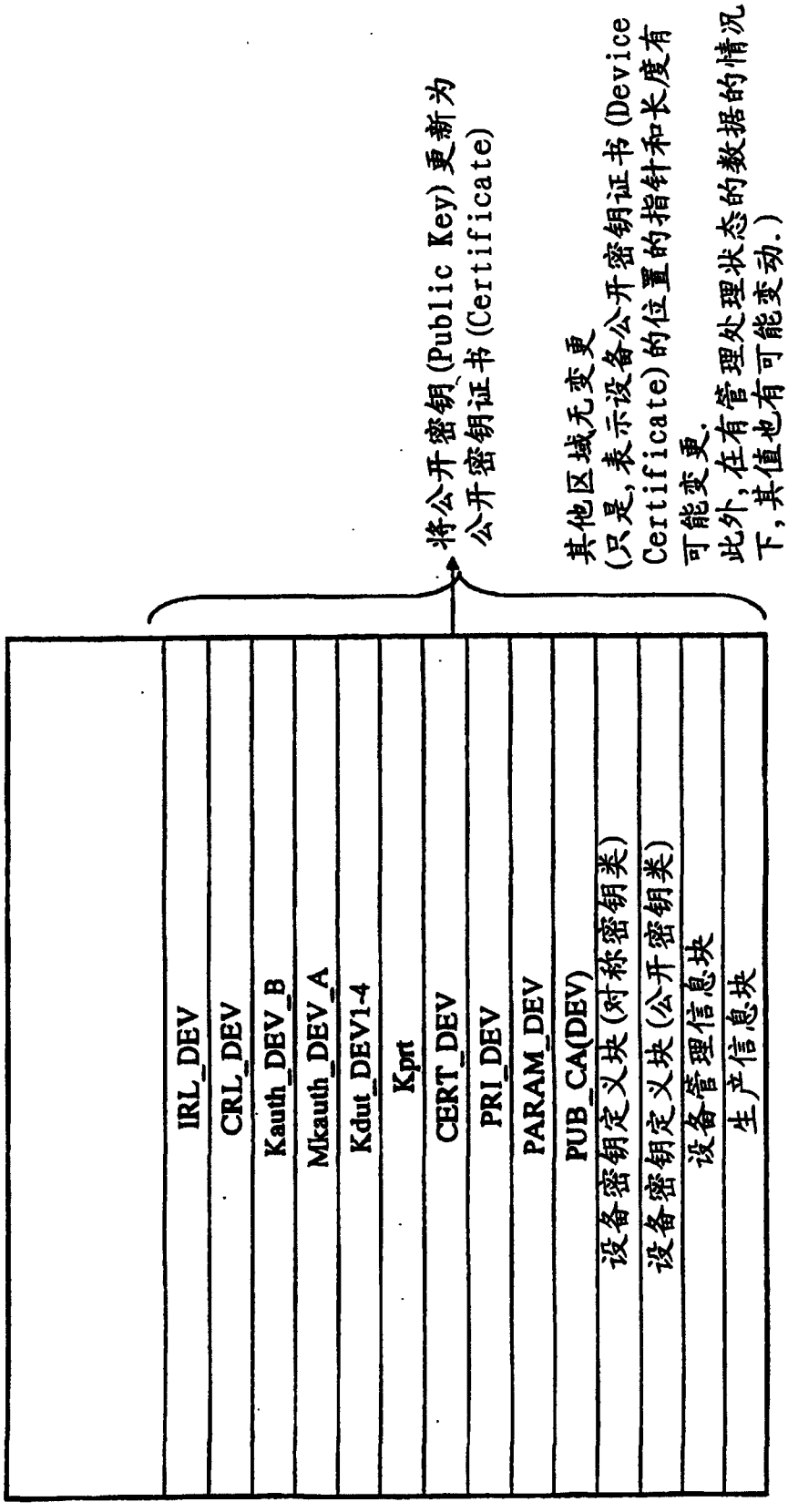

IRL_DEV
CRL_DEV
Kauth_DEV_B
Mkauth_DEV_A
Kdut_DEV1-4
Kprt
CERT_DEV
PRI_DEV
PARAM_DEV
PUB_CA(DEV)
设备密钥定义块(对称密钥类)
设备密钥定义块(公开密钥类)
设备管理信息块
生产信息块
将公开密钥(Public Key)更新为公开密钥证书(Certificate)
其他区域无变更
(只是,表示设备公开密钥证书(Device Certificate)的位置的指针和长度有可能变更.
此外,在有管理处理状态的数据的情况下,其值也有可能变动.)

图 46

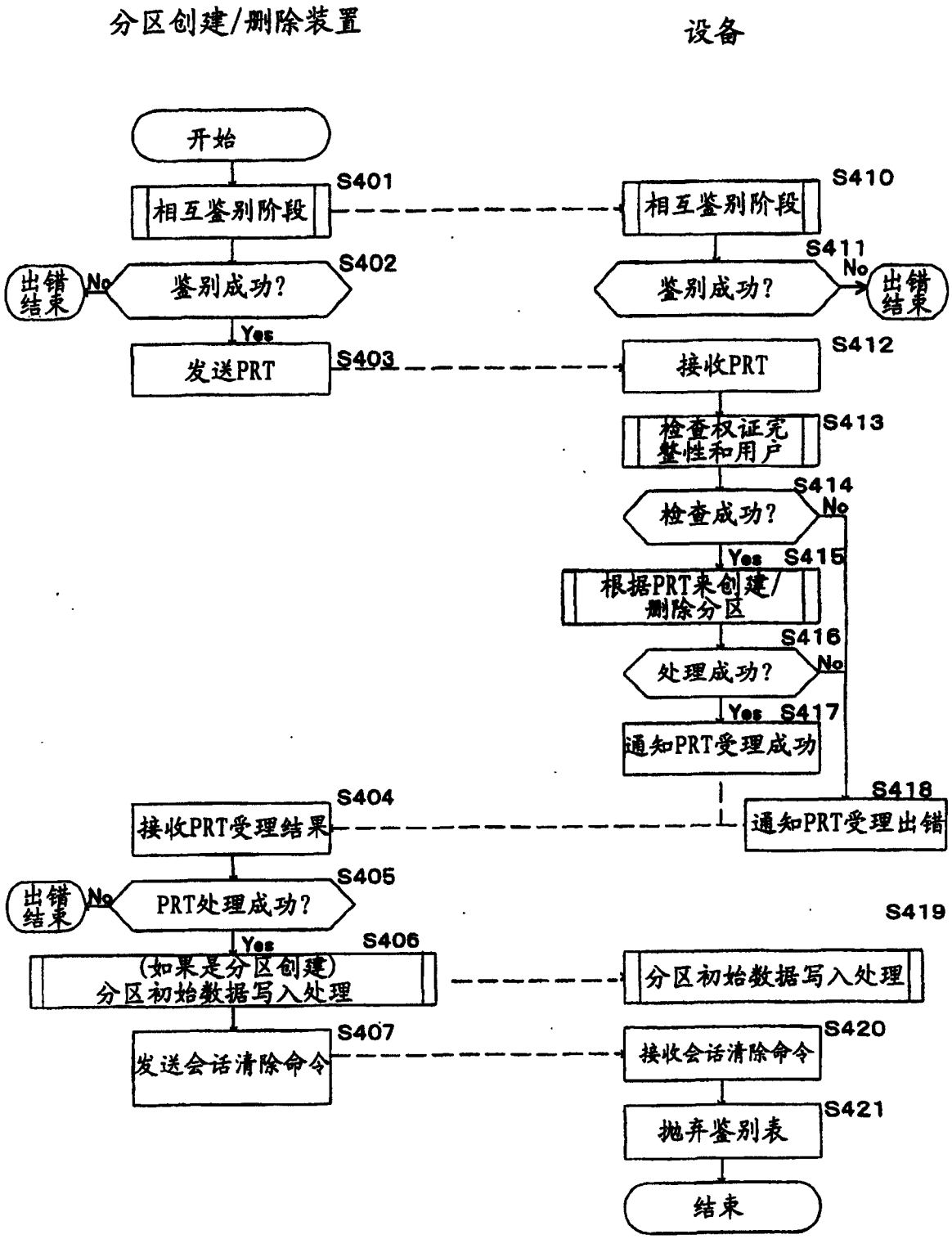

分区创建/删除装置
设备
开始
S401
相互鉴别阶段
S410
相互鉴别阶段
S411
S402
出错结束
No
鉴别成功?
鉴别成功?
No
出错结束
Yes
S403
发送PRT
S412
接收PRT
S413
检查权证完整性和用户
S414
检查成功?
No
Yes
S415
根据PRT来创建/删除分区
S416
处理成功?
No
Yes
S417
通知PRT受理成功
S418
通知PRT受理出错
S404
接收PRT受理结果
S405
出错结束
No
PRT处理成功?
S419
Yes
S406
(如果是分区创建)分区初始数据写入处理
分区初始数据写入处理
S407
发送会话清除命令
S420
接收会话清除命令
S421
抛弃鉴别表
结束

图 47

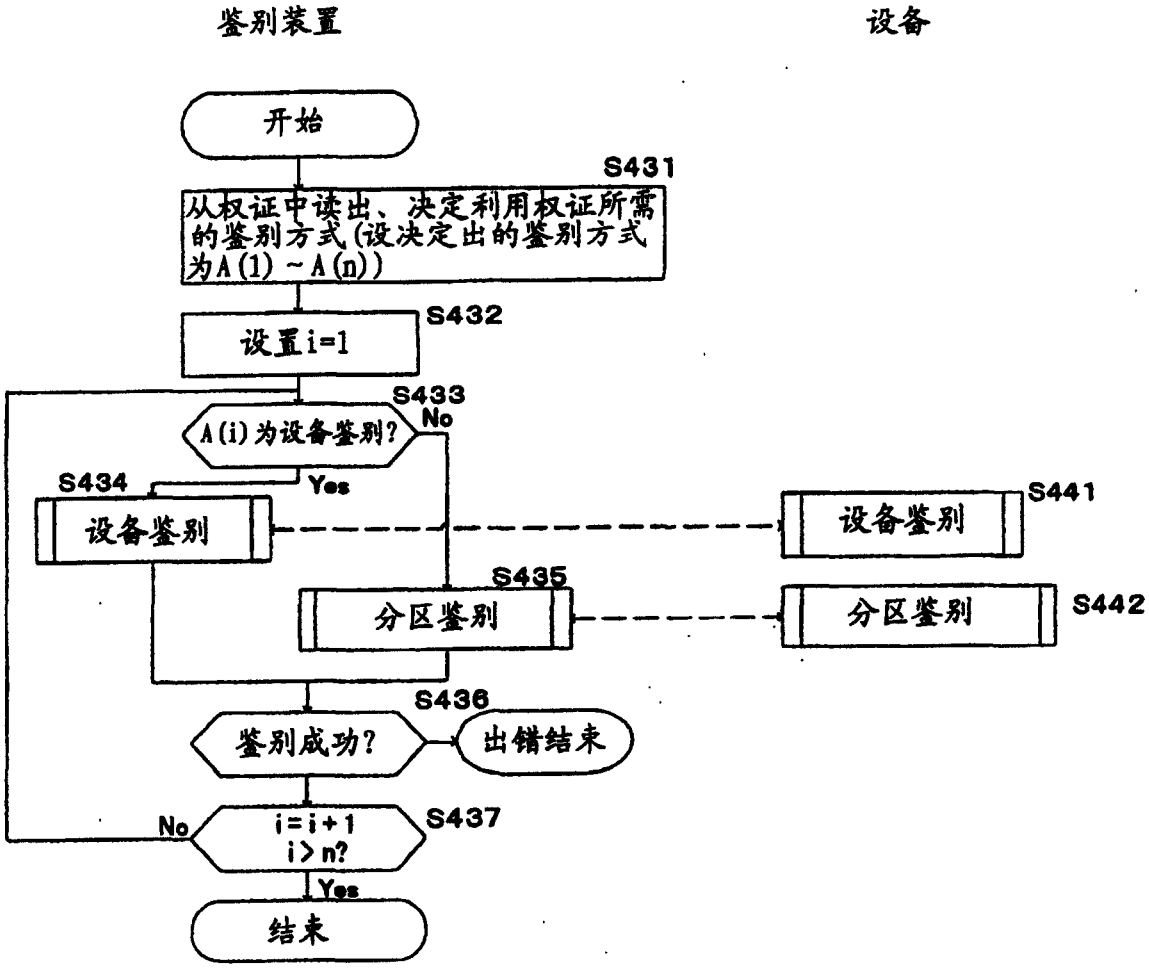

鉴别装置
设备
开始
S431
从权证中读出、决定利用权证所需的鉴别方式(设决定出的鉴别方式为A(1)～A(n))
S432
设置i=1
S433
A(i)为设备鉴别?
No
Yes
S434
设备鉴别
S441
设备鉴别
S435
分区鉴别
分区鉴别
S442
S436
鉴别成功?
出错结束
No
i=i+1
i>n?
S437
Yes
结束

图 48

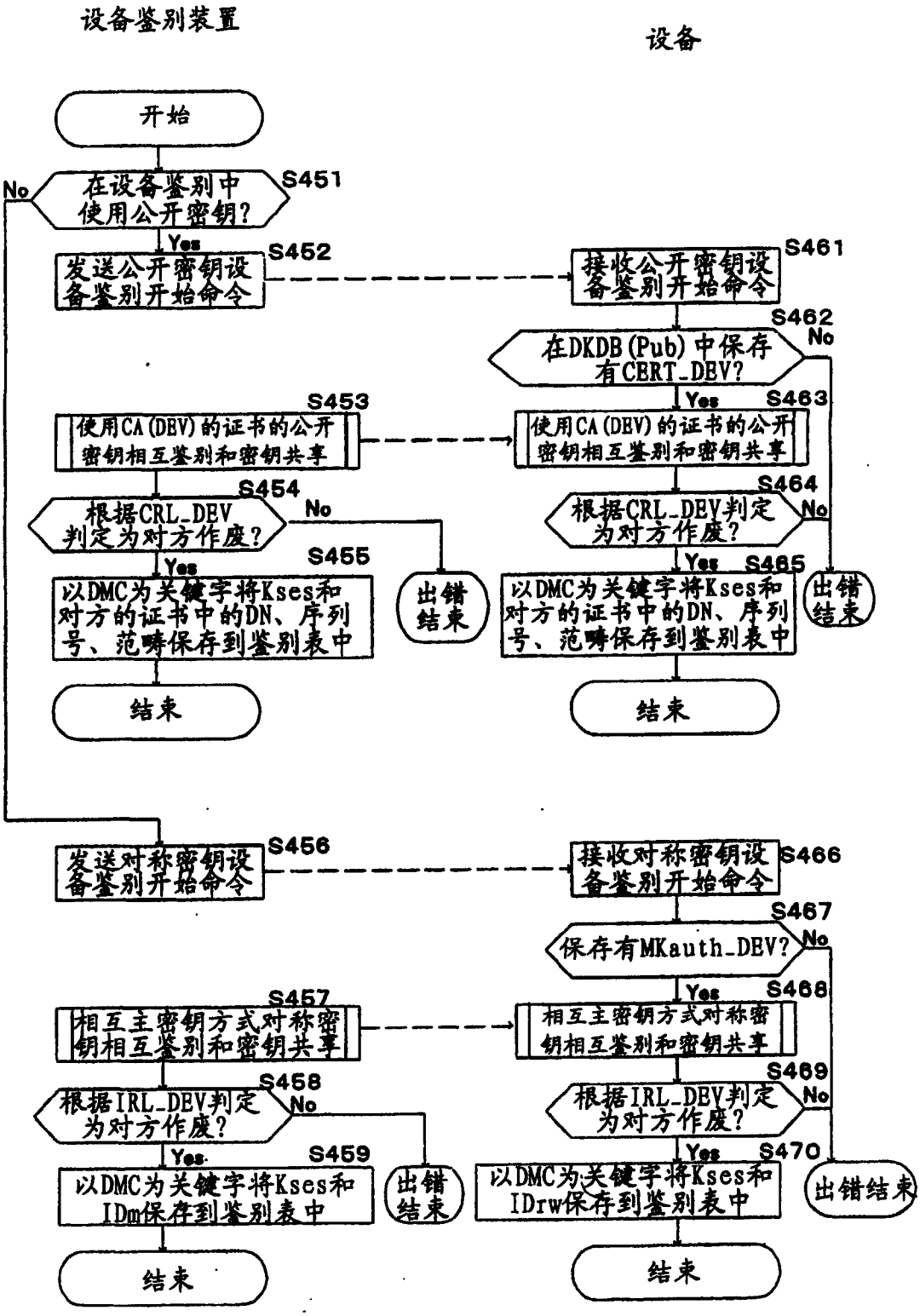

设备鉴别装置
设备
开始
在设备鉴别中使用公开密钥?
S451
No
Yes
发送公开密钥设备鉴别开始命令
S452
接收公开密钥设备鉴别开始命令
S461
在DKDB(Pub)中保存有CERT_DEV?
S462
No
Yes
使用CA(DEV)的证书的公开密钥相互鉴别和密钥共享
S453
使用CA(DEV)的证书的公开密钥相互鉴别和密钥共享
S463
根据CRL_DEV判定为对方作废?
S454
No
Yes
以DMC为关键字将Kses和对方的证书中的DN、序列号、范畴保存到鉴别表中
S455
出错结束
根据CRL_DEV判定为对方作废?
S464
No
Yes
以DMC为关键字将Kses和对方的证书中的DN、序列号、范畴保存到鉴别表中
S465
出错结束
结束
结束
发送对称密钥设备鉴别开始命令
S456
接收对称密钥设备鉴别开始命令
S466
保存有MKauth_DEV?
S467
No
Yes
相互主密钥方式对称密钥相互鉴别和密钥共享
S457
相互主密钥方式对称密钥相互鉴别和密钥共享
S468
根据IRL_DEV判定为对方作废?
S458
No
Yes
以DMC为关键字将Kses和IDm保存到鉴别表中
S459
出错结束
根据IRL_DEV判定为对方作废?
S469
No
Yes
以DMC为关键字将Kses和IDrw保存到鉴别表中
S470
出错结束
结束
结束

图 49

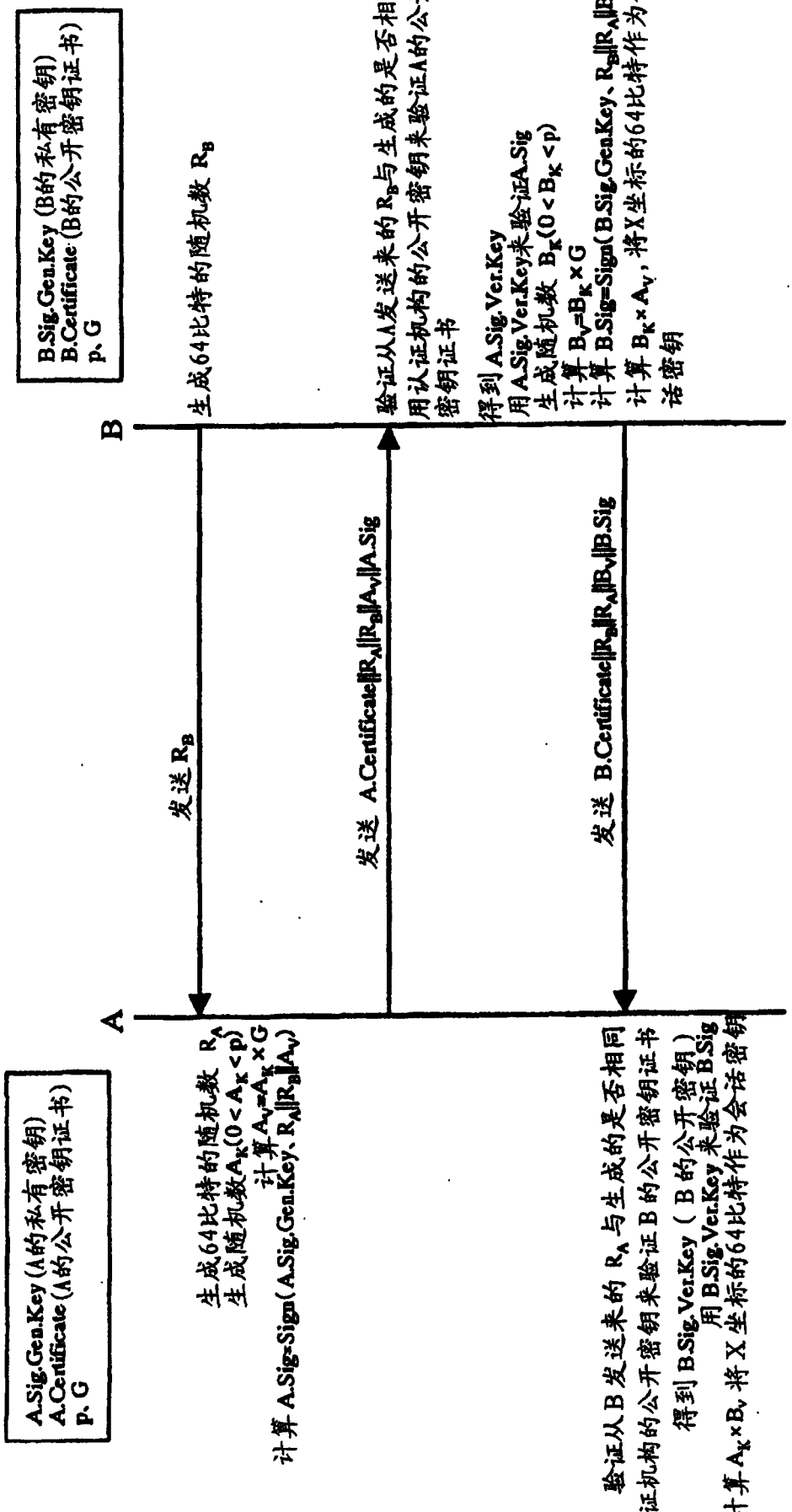

A.Sig.Gen.Key (A的私有密钥)
A.Certificate (A的公开密钥证书)
p、G
A
B
B.Sig.Gen.Key (B的私有密钥)
B.Certificate (B的公开密钥证书)
p、G
生成64比特的随机数 R_A
生成随机数 $A_K(0<A_K<p)$
计算 $A_V=A_K\times G$
计算 A.Sig=Sign(A.Sig.Gen.Key、R_A||R_B||A_V)
发送 R_B
生成64比特的随机数 R_B
发送 A.Certificate||R_A||R_B||A_V||A.Sig
验证从A发送来的 R_B 与生成的是否相同
用认证机构的公开密钥来验证A的公开密钥证书
得到 A.Sig.Ver.Key
用 A.Sig.Ver.Key 来验证 A.Sig
生成随机数 $B_K(0<B_K<p)$
计算 $B_V=B_K\times G$
计算 B.Sig=Sign(B.Sig.Gen.Key、R_B||R_A||B_V)
发送 B.Certificate||R_B||R_A||B_V||B.Sig
验证从B发送来的 R_A 与生成的是否相同
用认证机构的公开密钥来验证B的公开密钥证书
得到 B.Sig.Ver.Key（B的公开密钥）
用 B.Sig.Ver.Key 来验证 B.Sig
计算 $A_K\times B_V$，将X坐标的64比特作为会话密钥
计算 $B_K\times A_V$，将X坐标的64比特作为会话密钥

图 50

组	公开密钥体制鉴别者信息	会话密钥	对称密钥体制鉴别者信息	会话密钥
DMC	DN, 序列号, 范畴	Kses1	—	—
PMC1	—	—	ID_RW	Kses2
PMC 2	DN, 序列号, 范畴	Kses3	ID_RW	Kses4

图 51

组	公开密钥体制鉴别者信息	对称密钥体制鉴别者信息	会话密钥
DMC	DN, 序列号, 范畴	—	Kses1
PMC1	—	IDm	Kses2
PMC2	DN, 序列号, 范畴	IDm	Kses3

图 52

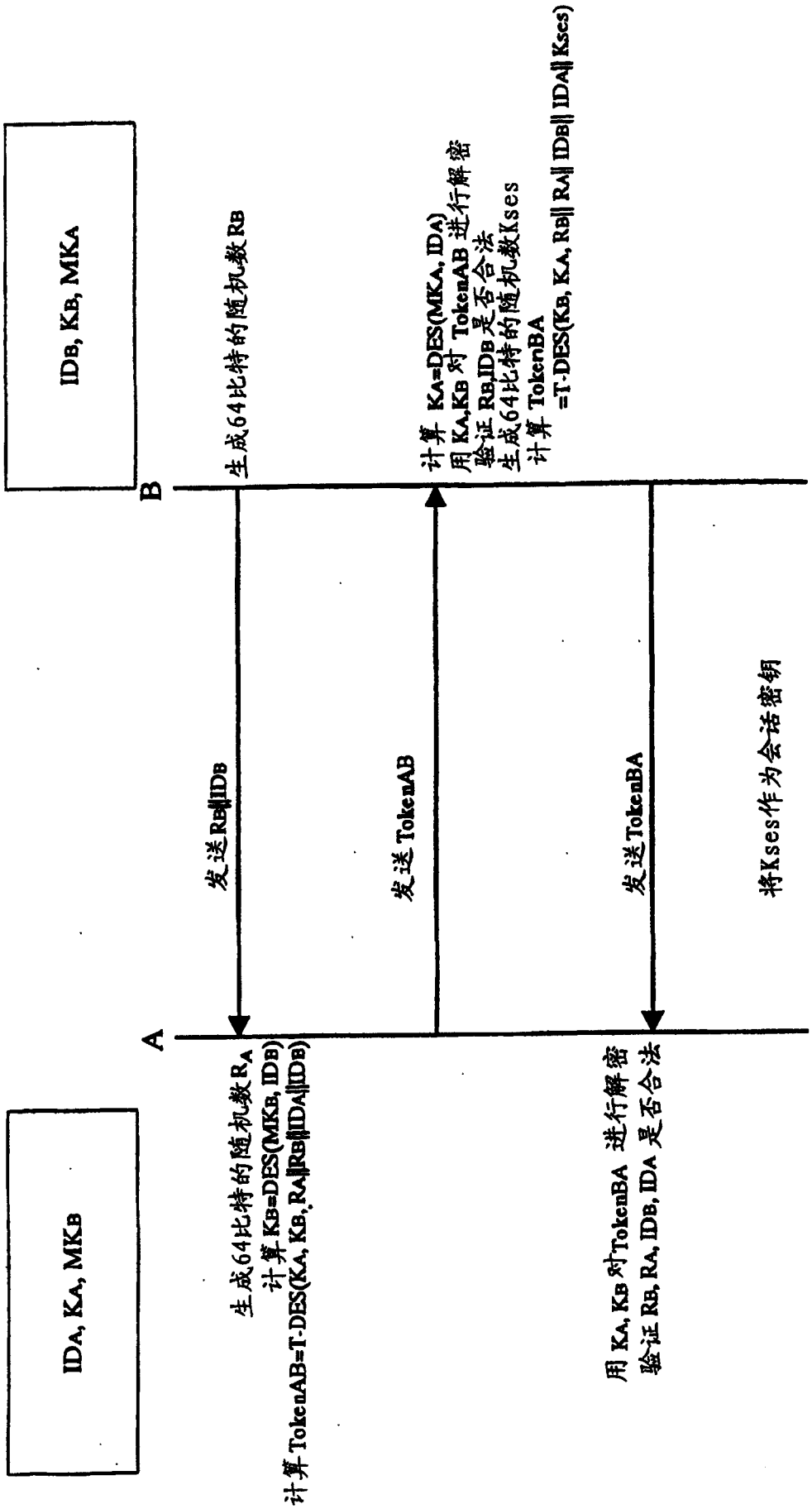

IDA, KA, MKB
IDB, KB, MKA
A
B
生成64比特的随机数RA
计算 KB=DES(MKB, IDB)
计算 TokenAB=T-DES(KA, KB, RA‖RB‖IDA‖IDB))
发送RB‖IDB
生成64比特的随机数RB
计算 KA=DES(MKA, IDA)
用KA,KB对 TokenAB 进行解密
验证 RB,IDB是否合法
生成64比特的随机数Kses
计算 TokenBA
=T-DES(KB, KA, RB‖ RA‖ IDB‖ IDA‖ Kses)
发送 TokenAB
发送 TokenBA
将Kses作为会话密钥
用 KA, KB对TokenBA 进行解密
验证 RB, RA, IDB, IDA 是否合法

图 53

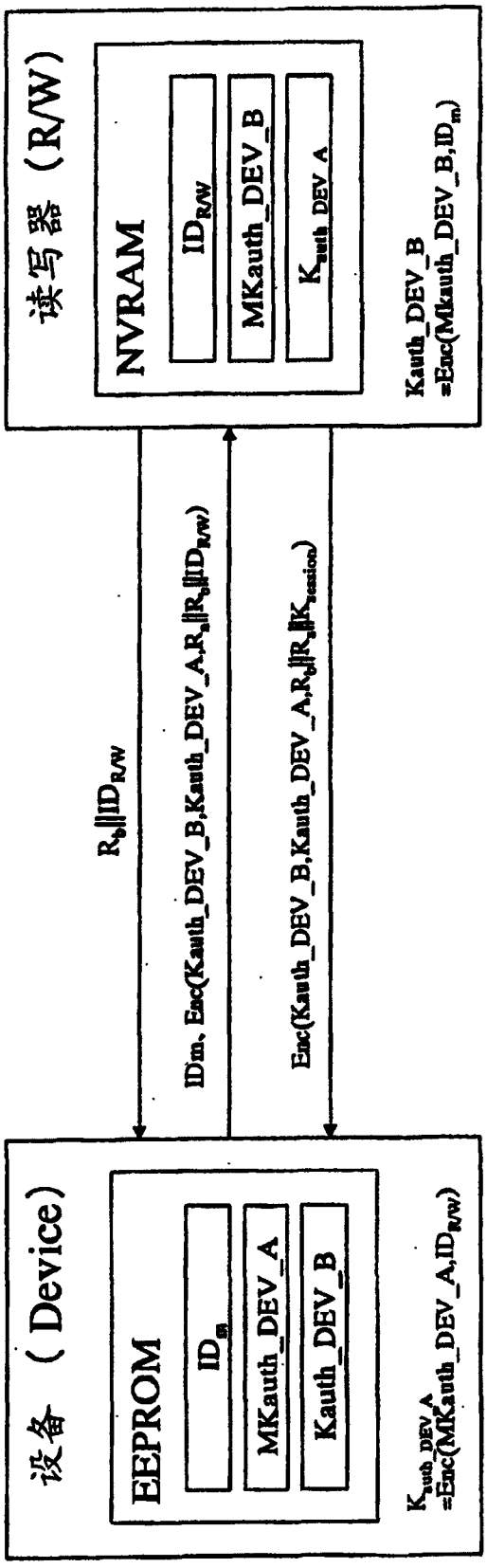

读写器（R/W）
NVRAM
$ID_{R/W}$
MKauth_DEV_B
$K_{auth\_DEV\_A}$
Kauth_DEV_B
=Enc(Mkauth_DEV_B,ID_m)
$R_b \| ID_{R/W}$
IDm, Enc(Kauth_DEV_B,Kauth_DEV_A,$R_a \| R_b \| ID_{R/W}$)
Enc(Kauth_DEV_B,Kauth_DEV_A,$R_a \| R_b \| K_{session}$)
设备（Device）
EEPROM
ID_m
MKauth_DEV_A
Kauth_DEV_B
$K_{auth\_DEV\_A}$
=Enc(MKauth_DEV_A,$ID_{R/W}$)

图 54

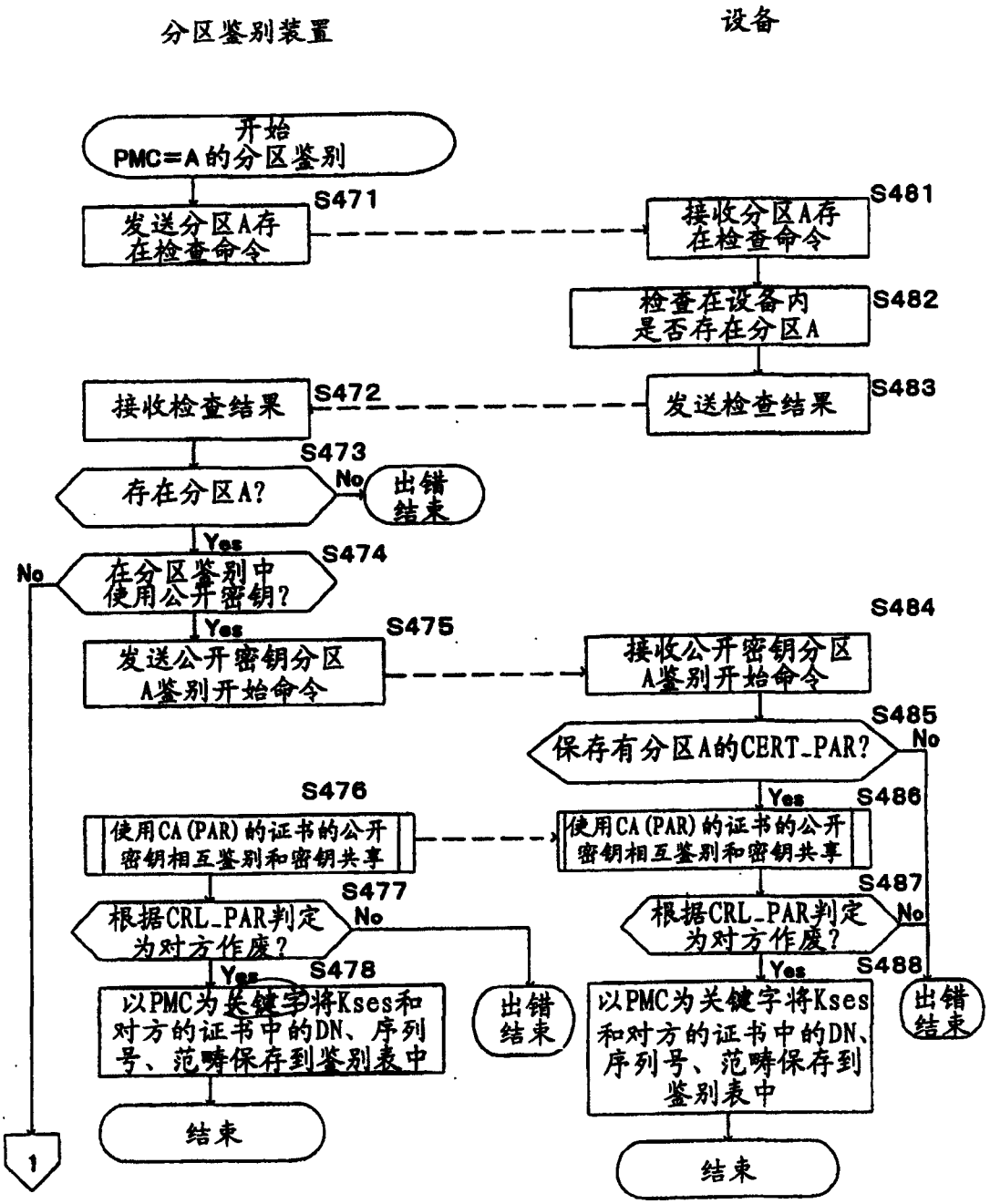

分区鉴别装置
设备
开始
PMC=A的分区鉴别
S471
发送分区A存在检查命令
S481
接收分区A存在检查命令
检查在设备内是否存在分区A
S482
接收检查结果
S472
发送检查结果
S483
S473
存在分区A?
No
出错结束
Yes
No
S474
在分区鉴别中使用公开密钥?
Yes
S475
发送公开密钥分区A鉴别开始命令
S484
接收公开密钥分区A鉴别开始命令
S485
保存有分区A的CERT_PAR?
No
S476
使用CA(PAR)的证书的公开密钥相互鉴别和密钥共享
Yes
S486
使用CA(PAR)的证书的公开密钥相互鉴别和密钥共享
S477
根据CRL_PAR判定为对方作废?
No
S487
根据CRL_PAR判定为对方作废?
No
Yes
S478
以PMC为关键字将Kses和对方的证书中的DN、序列号、范畴保存到鉴别表中
出错结束
Yes
S488
以PMC为关键字将Kses和对方的证书中的DN、序列号、范畴保存到鉴别表中
出错结束
结束
结束
1

图 55

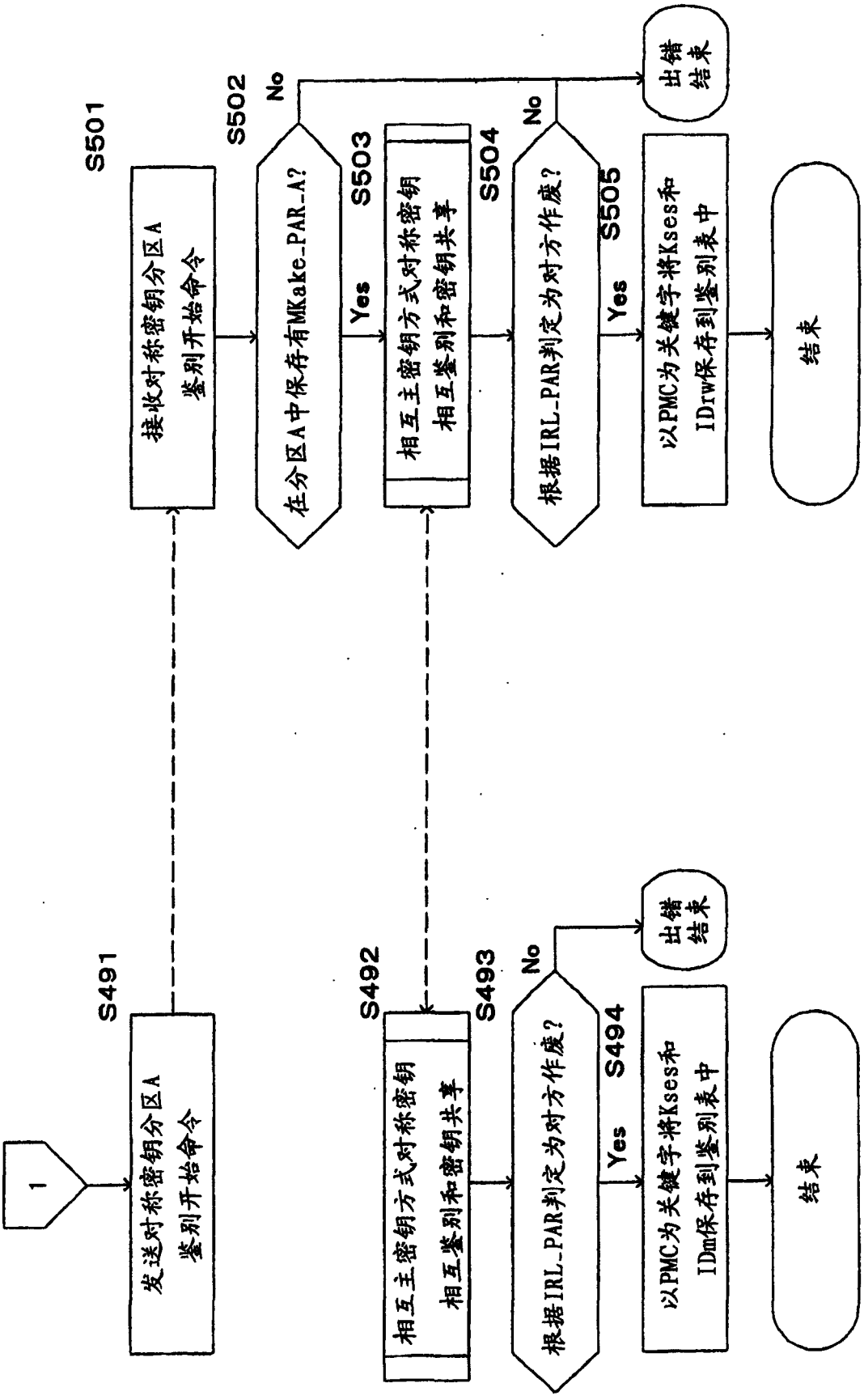

1
S491
发送对称密钥分区A
鉴别开始命令
S492
相互主密钥方式对称密钥
相互鉴别和密钥共享
S493
No
根据IRL_PAR判定为对方作废?
出错
结束
Yes
S494
以PMC为关键字将Kses和
IDm保存到鉴别表中
结束
S501
接收对称密钥分区A
鉴别开始命令
S502
No
在分区A中保存有MKake_PAR_A?
出错
结束
Yes
S503
相互主密钥方式对称密钥
相互鉴别和密钥共享
S504
No
根据IRL_PAR判定为对方作废?
出错
结束
Yes
S505
以PMC为关键字将Kses和
IDrw保存到鉴别表中
结束

图 56

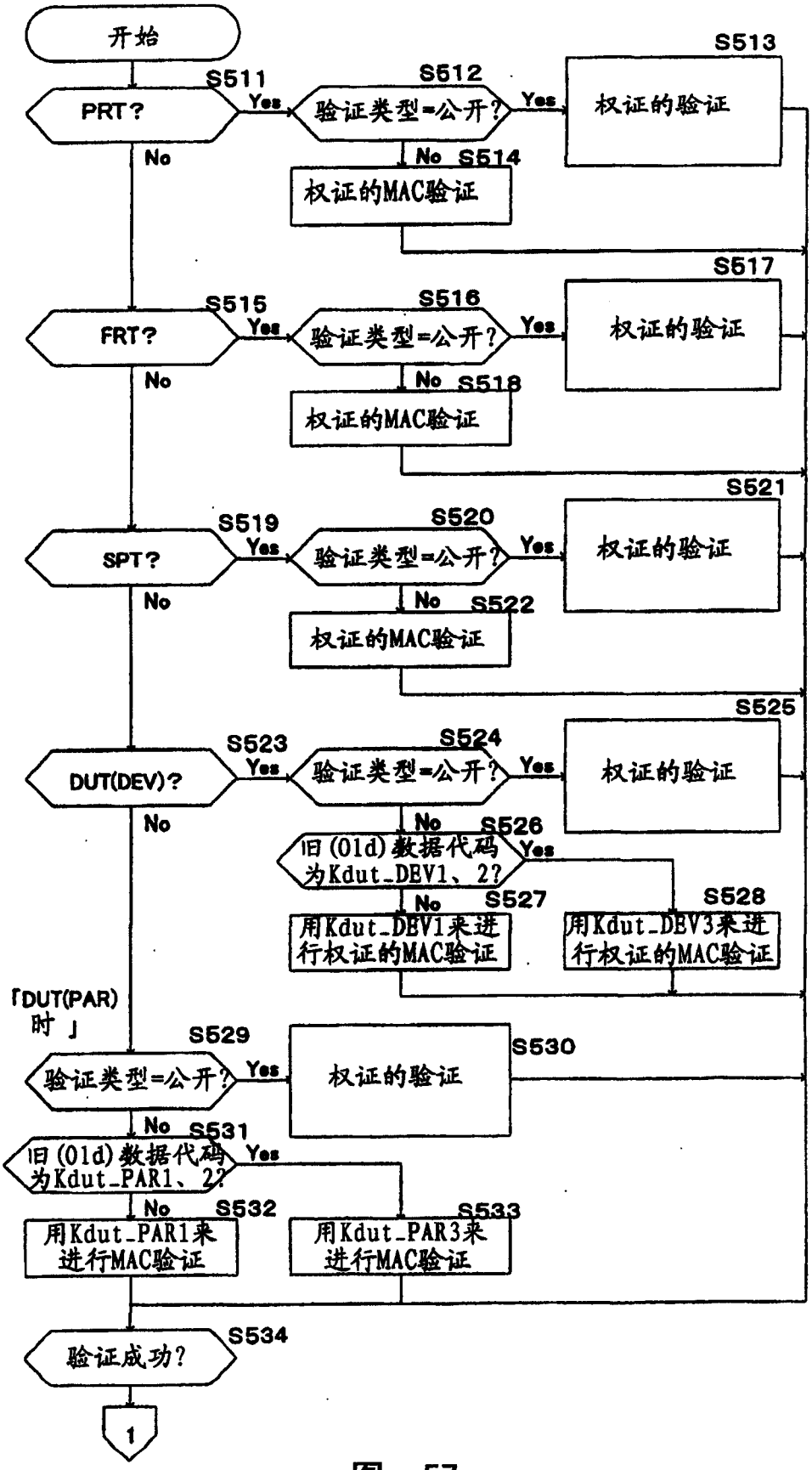

开始
PRT?
S511
Yes
No
验证类型=公开?
S512
Yes
No
S513
权证的验证
S514
权证的MAC验证
FRT?
S515
Yes
No
验证类型=公开?
S516
Yes
No
S517
权证的验证
S518
权证的MAC验证
SPT?
S519
Yes
No
验证类型=公开?
S520
Yes
No
S521
权证的验证
S522
权证的MAC验证
DUT(DEV)?
S523
Yes
No
验证类型=公开?
S524
Yes
No
S525
权证的验证
旧(Old)数据代码为Kdut_DEV1、2?
S526
Yes
No
S527
用Kdut_DEV1来进行权证的MAC验证
S528
用Kdut_DEV3来进行权证的MAC验证
「DUT(PAR)时」
验证类型=公开?
S529
Yes
No
S530
权证的验证
旧(Old)数据代码为Kdut_PAR1、2?
S531
Yes
No
用Kdut_PAR1来进行MAC验证
S532
用Kdut_PAR3来进行MAC验证
S533
验证成功?
S534
1

图 57

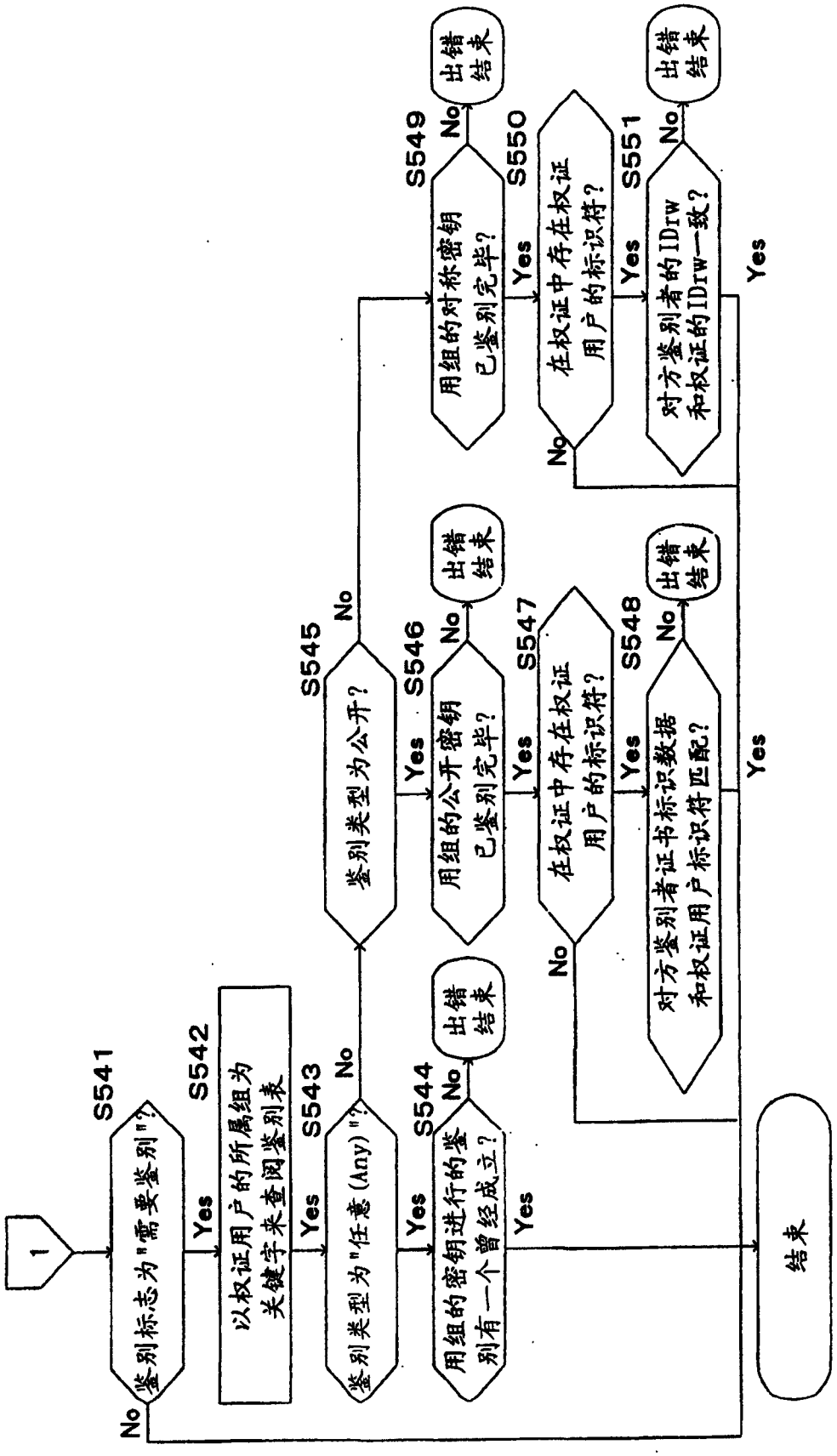

1
S541
鉴别标志为"需要鉴别"?
No
Yes
S542
以权证用户的所属组为关键字来查阅鉴别表
S543
鉴别类型为"任意(Any)"?
No
Yes
S544
用组的密钥进行的鉴别有一个曾经成立?
No
出错结束
Yes
S545
鉴别类型为公开?
No
Yes
S546
用组的公开密钥已鉴别完毕?
No
出错结束
Yes
S547
在权证中存在权证用户的标识符?
No
Yes
S548
对方鉴别者证书标识数据和权证用户标识符匹配?
No
出错结束
Yes
S549
用组的对称密钥已鉴别完毕?
No
出错结束
Yes
S550
在权证中存在权证用户的标识符?
No
Yes
S551
对方鉴别者的IDrw和权证的IDrw一致?
No
出错结束
Yes
结束

图 58

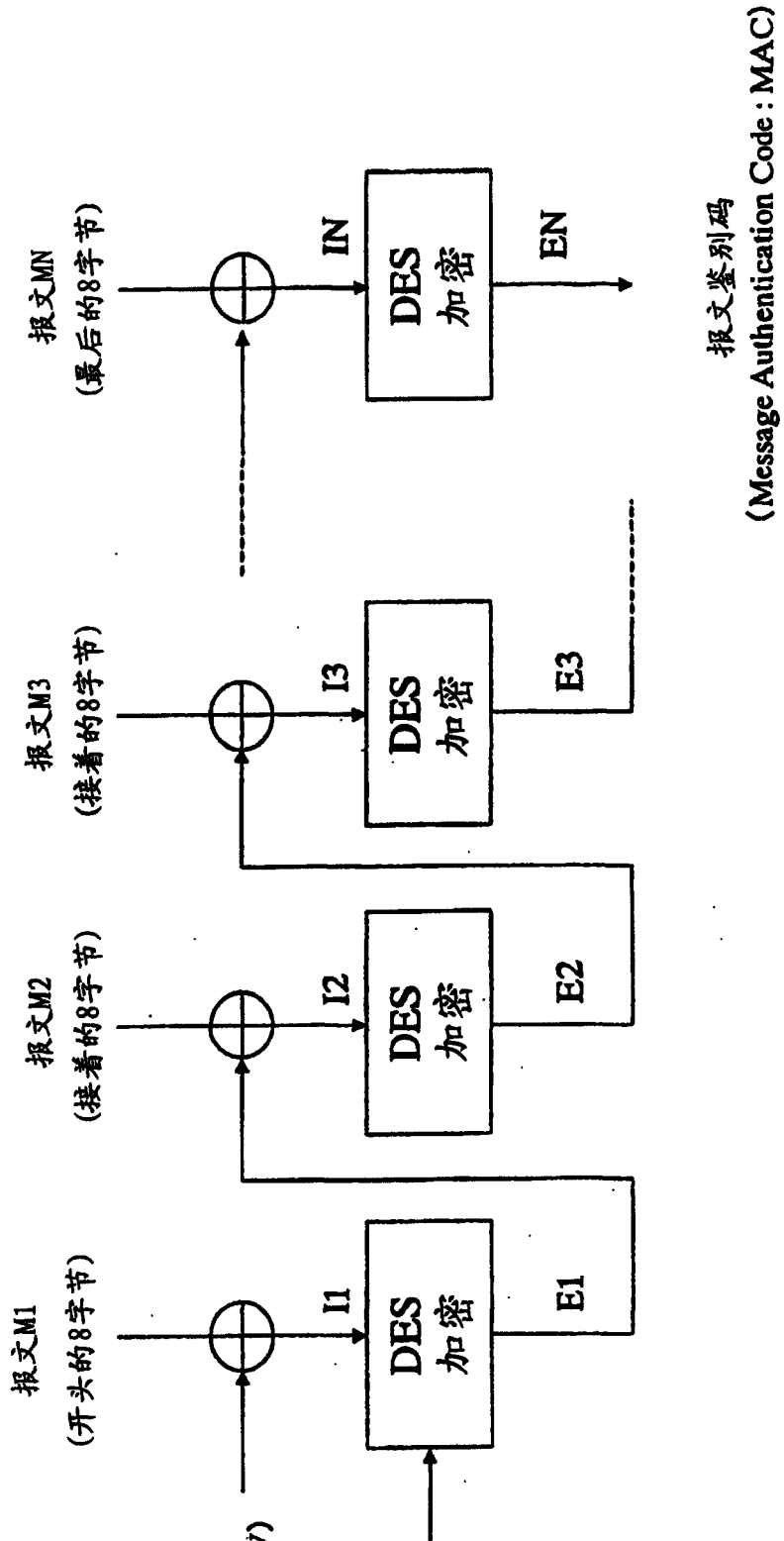

报文M1
(开头的8字节)
报文M2
(接着的8字节)
报文M3
(接着的8字节)
报文MN
(最后的8字节)
初值
(Initial Value: IV)
密钥
I1
I2
I3
IN
DES
加密
E1
E2
E3
EN
报文鉴别码
(Message Authentication Code : MAC)
：逻辑"异或"处理(以8字节为单位)

图 59

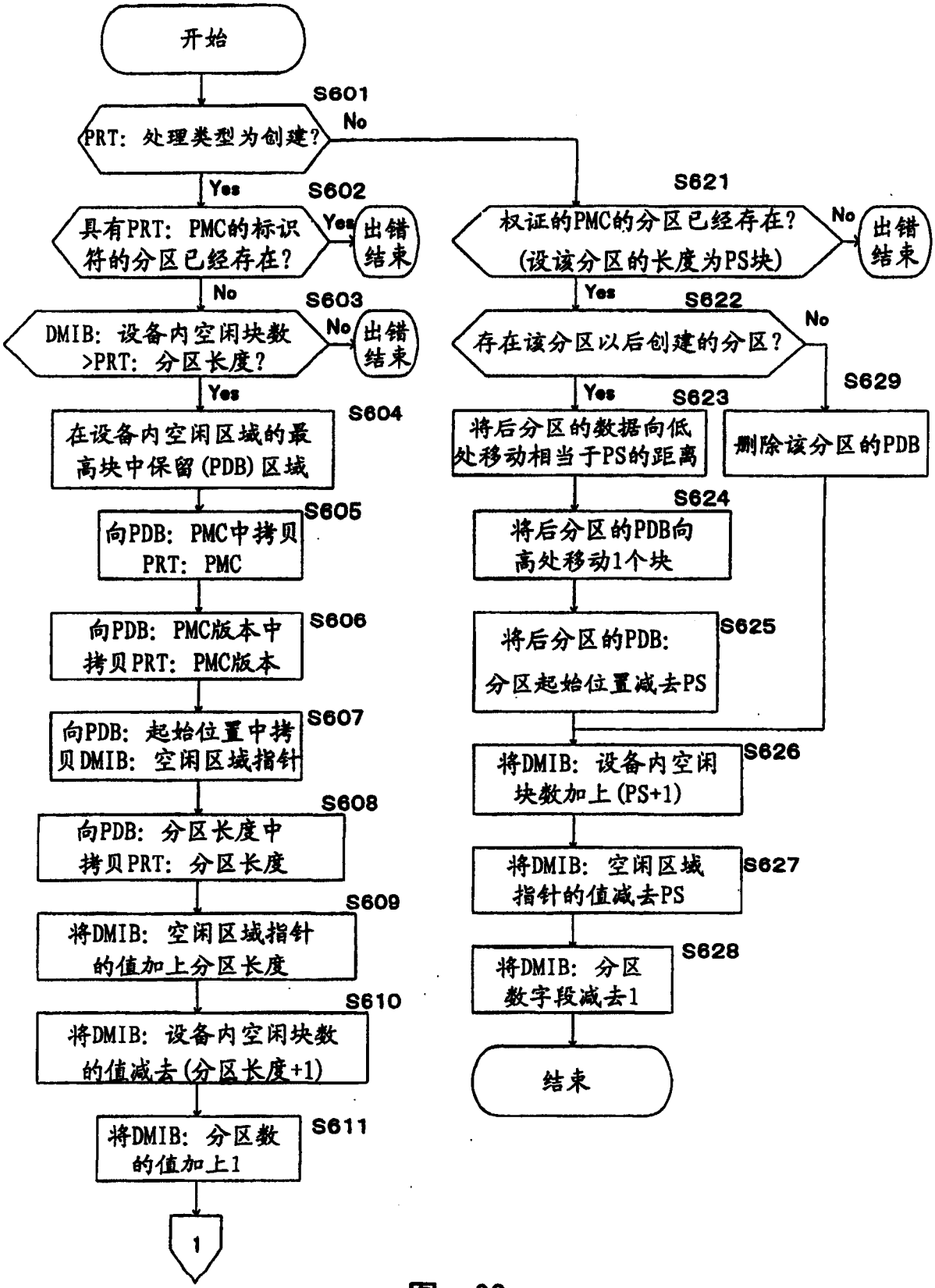

开始
S601
PRT：处理类型为创建？
No
Yes
S602
具有PRT：PMC的标识符的分区已经存在？
Yes
出错结束
No
S603
DMIB：设备内空闲块数>PRT：分区长度？
No
出错结束
Yes
S604
在设备内空闲区域的最高块中保留(PDB)区域
S605
向PDB：PMC中拷贝PRT：PMC
S606
向PDB：PMC版本中拷贝PRT：PMC版本
S607
向PDB：起始位置中拷贝DMIB：空闲区域指针
S608
向PDB：分区长度中拷贝PRT：分区长度
S609
将DMIB：空闲区域指针的值加上分区长度
S610
将DMIB：设备内空闲块数的值减去(分区长度+1)
S611
将DMIB：分区数的值加上1
1
S621
权证的PMC的分区已经存在？(设该分区的长度为PS块)
No
出错结束
Yes
S622
存在该分区以后创建的分区？
No
S629
删除该分区的PDB
Yes
S623
将后分区的数据向低处移动相当于PS的距离
S624
将后分区的PDB向高处移动1个块
S625
将后分区的PDB：分区起始位置减去PS
S626
将DMIB：设备内空闲块数加上(PS+1)
S627
将DMIB：空闲区域指针的值减去PS
S628
将DMIB：分区数字段减去1
结束

图 60

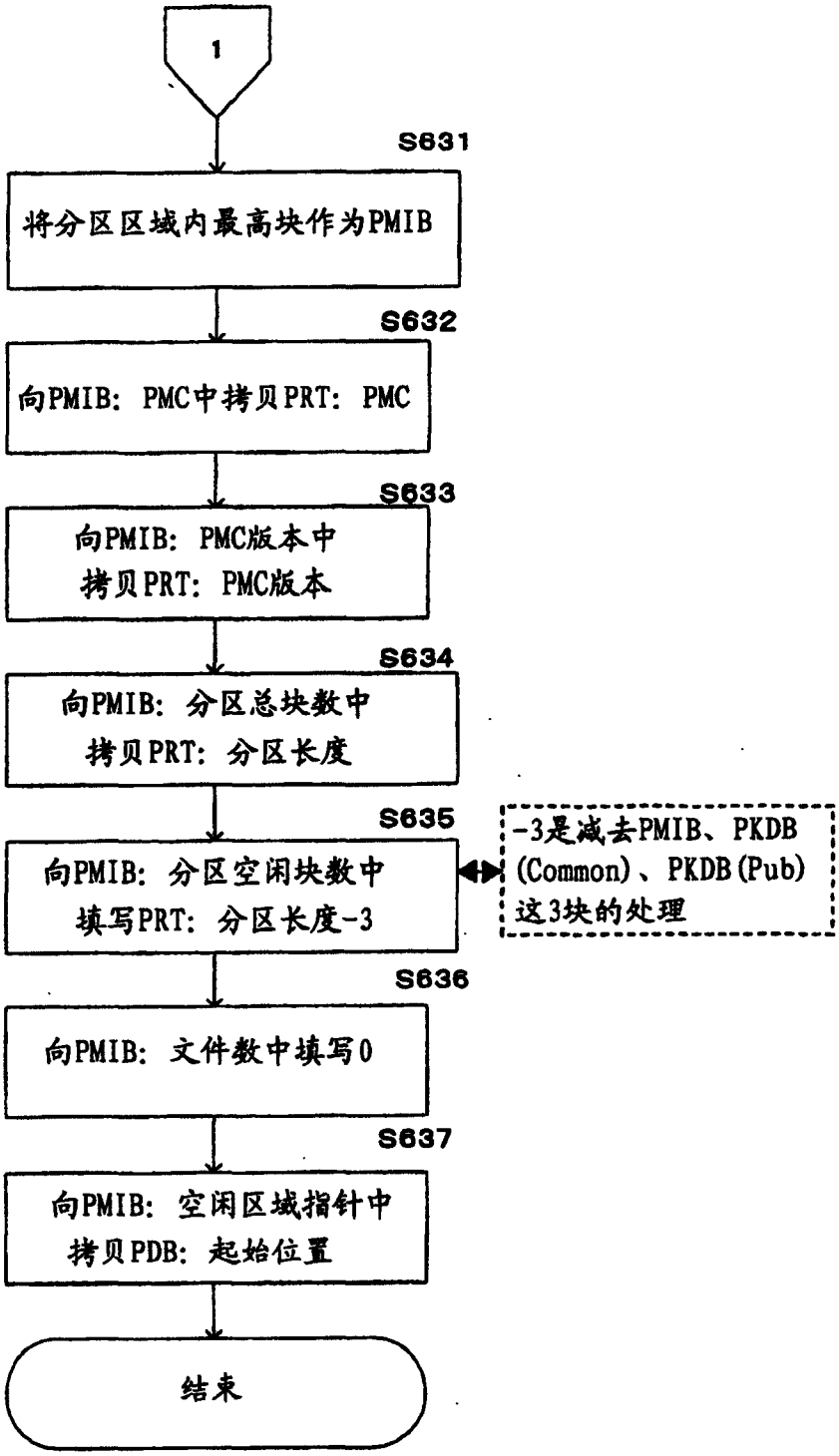

1
S631
将分区区域内最高块作为PMIB
S632
向PMIB：PMC中拷贝PRT：PMC
S633
向PMIB：PMC版本中
拷贝PRT：PMC版本
S634
向PMIB：分区总块数中
拷贝PRT：分区长度
S635
向PMIB：分区空闲块数中
填写PRT：分区长度-3
-3是减去PMIB、PKDB
(Common)、PKDB(Pub)
这3块的处理
S636
向PMIB：文件数中填写0
S637
向PMIB：空闲区域指针中
拷贝PDB：起始位置
结束

图 61

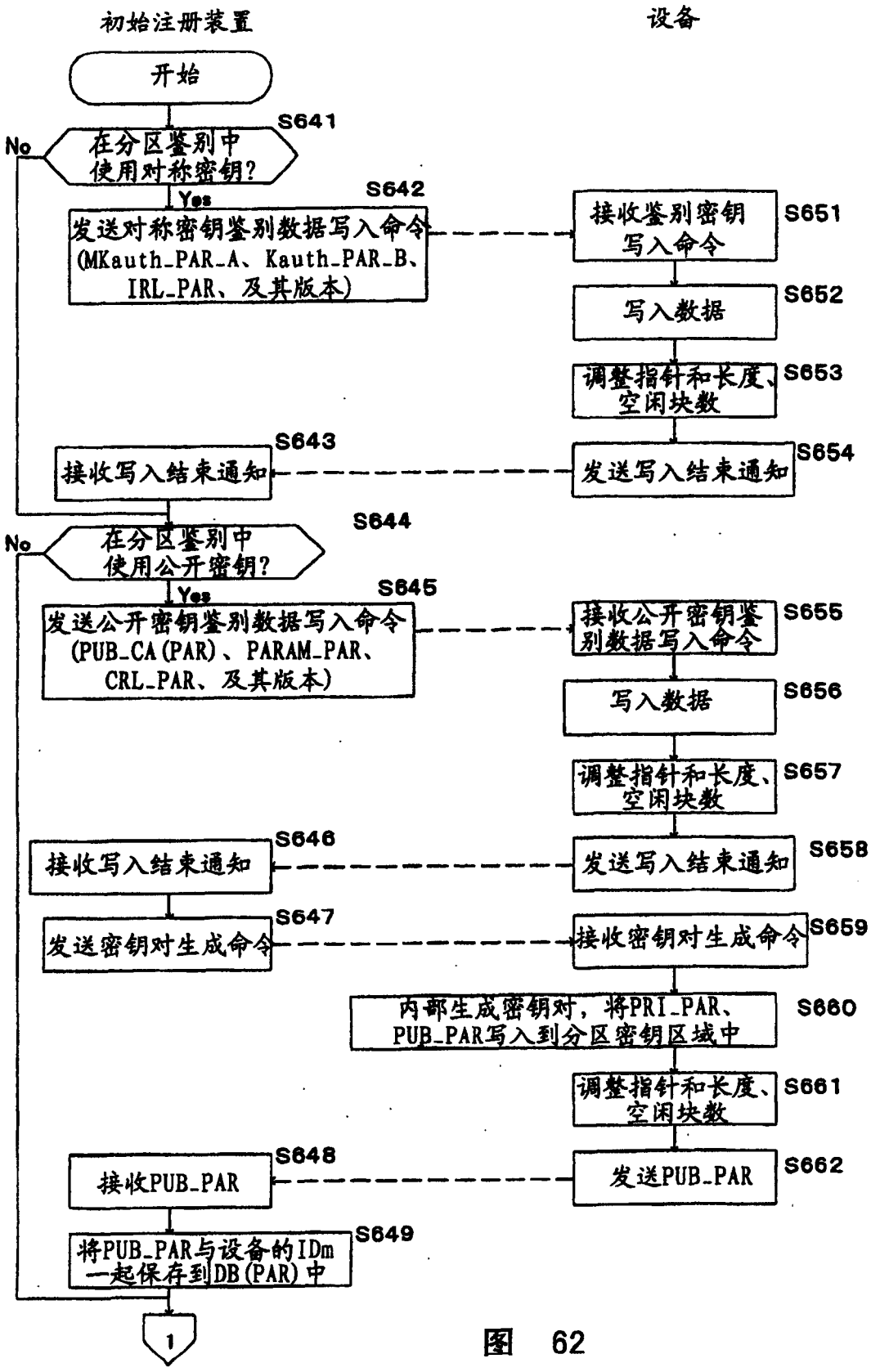

初始注册装置
设备
开始
S641
在分区鉴别中使用对称密钥?
No
Yes
S642
发送对称密钥鉴别数据写入命令(MKauth_PAR_A、Kauth_PAR_B、IRL_PAR、及其版本)
接收鉴别密钥写入命令
S651
写入数据
S652
调整指针和长度、空闲块数
S653
S643
接收写入结束通知
发送写入结束通知
S654
S644
在分区鉴别中使用公开密钥?
No
Yes
S645
发送公开密钥鉴别数据写入命令(PUB_CA(PAR)、PARAM_PAR、CRL_PAR、及其版本)
接收公开密钥鉴别数据写入命令
S655
写入数据
S656
调整指针和长度、空闲块数
S657
S646
接收写入结束通知
发送写入结束通知
S658
S647
发送密钥对生成命令
接收密钥对生成命令
S659
内部生成密钥对，将PRI_PAR、PUB_PAR写入到分区密钥区域中
S660
调整指针和长度、空闲块数
S661
S648
接收PUB_PAR
发送PUB_PAR
S662
S649
将PUB_PAR与设备的IDm一起保存到DB(PAR)中
1

图 62

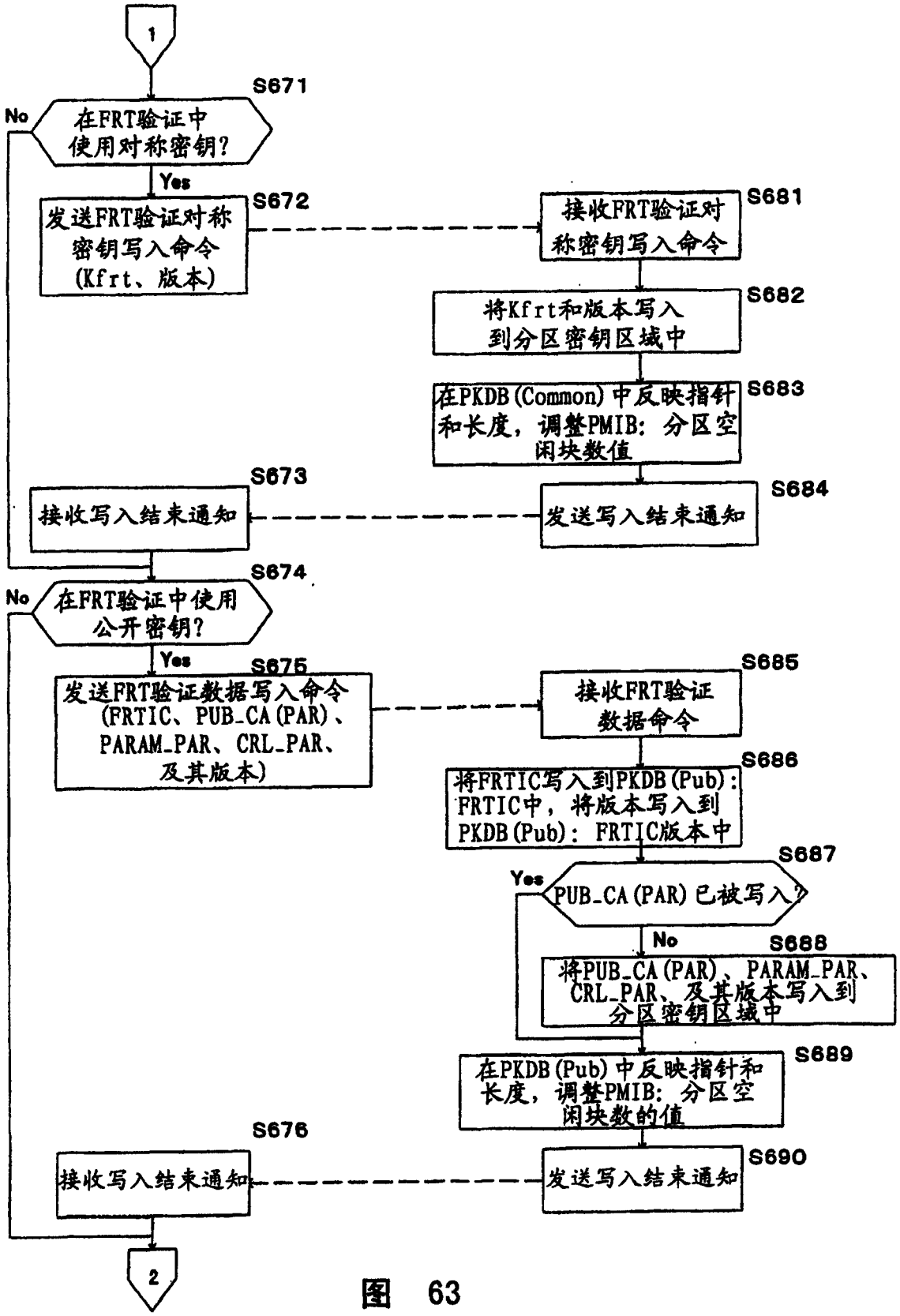

1
S671
No
在FRT验证中使用对称密钥?
Yes
S672
发送FRT验证对称密钥写入命令(Kfrt、版本)
S681
接收FRT验证对称密钥写入命令
S682
将Kfrt和版本写入到分区密钥区域中
S683
在PKDB(Common)中反映指针和长度，调整PMIB：分区空闲块数值
S673
接收写入结束通知
S684
发送写入结束通知
S674
No
在FRT验证中使用公开密钥?
Yes
S675
发送FRT验证数据写入命令(FRTIC、PUB_CA(PAR)、PARAM_PAR、CRL_PAR、及其版本)
S685
接收FRT验证数据命令
S686
将FRTIC写入到PKDB(Pub)：FRTIC中，将版本写入到PKDB(Pub)：FRTIC版本中
S687
Yes
PUB_CA(PAR)已被写入?
No
S688
将PUB_CA(PAR)、PARAM_PAR、CRL_PAR、及其版本写入到分区密钥区域中
S689
在PKDB(Pub)中反映指针和长度，调整PMIB：分区空闲块数的值
S676
接收写入结束通知
S690
发送写入结束通知
2

图 63

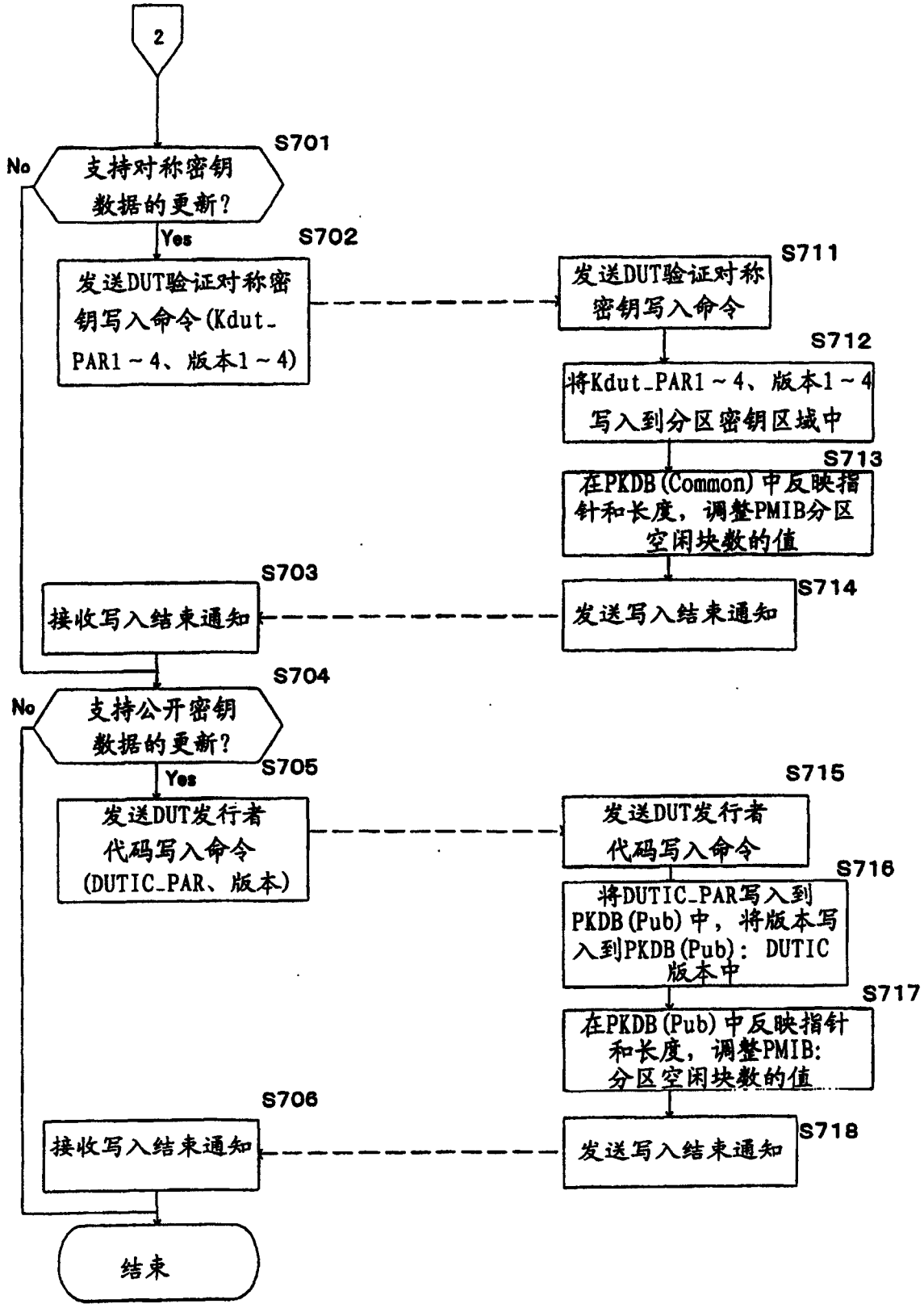

2
S701
No
支持对称密钥数据的更新?
Yes
S702
发送DUT验证对称密钥写入命令(Kdut_PAR1～4、版本1～4)
S711
发送DUT验证对称密钥写入命令
S712
将Kdut_PAR1～4、版本1～4写入到分区密钥区域中
S713
在PKDB(Common)中反映指针和长度，调整PMIB分区空闲块数的值
S714
发送写入结束通知
S703
接收写入结束通知
S704
No
支持公开密钥数据的更新?
Yes
S705
发送DUT发行者代码写入命令(DUTIC_PAR、版本)
S715
发送DUT发行者代码写入命令
S716
将DUTIC_PAR写入到PKDB(Pub)中，将版本写入到PKDB(Pub)：DUTIC版本中
S717
在PKDB(Pub)中反映指针和长度，调整PMIB:分区空闲块数的值
S718
发送写入结束通知
S706
接收写入结束通知
结束

图 64

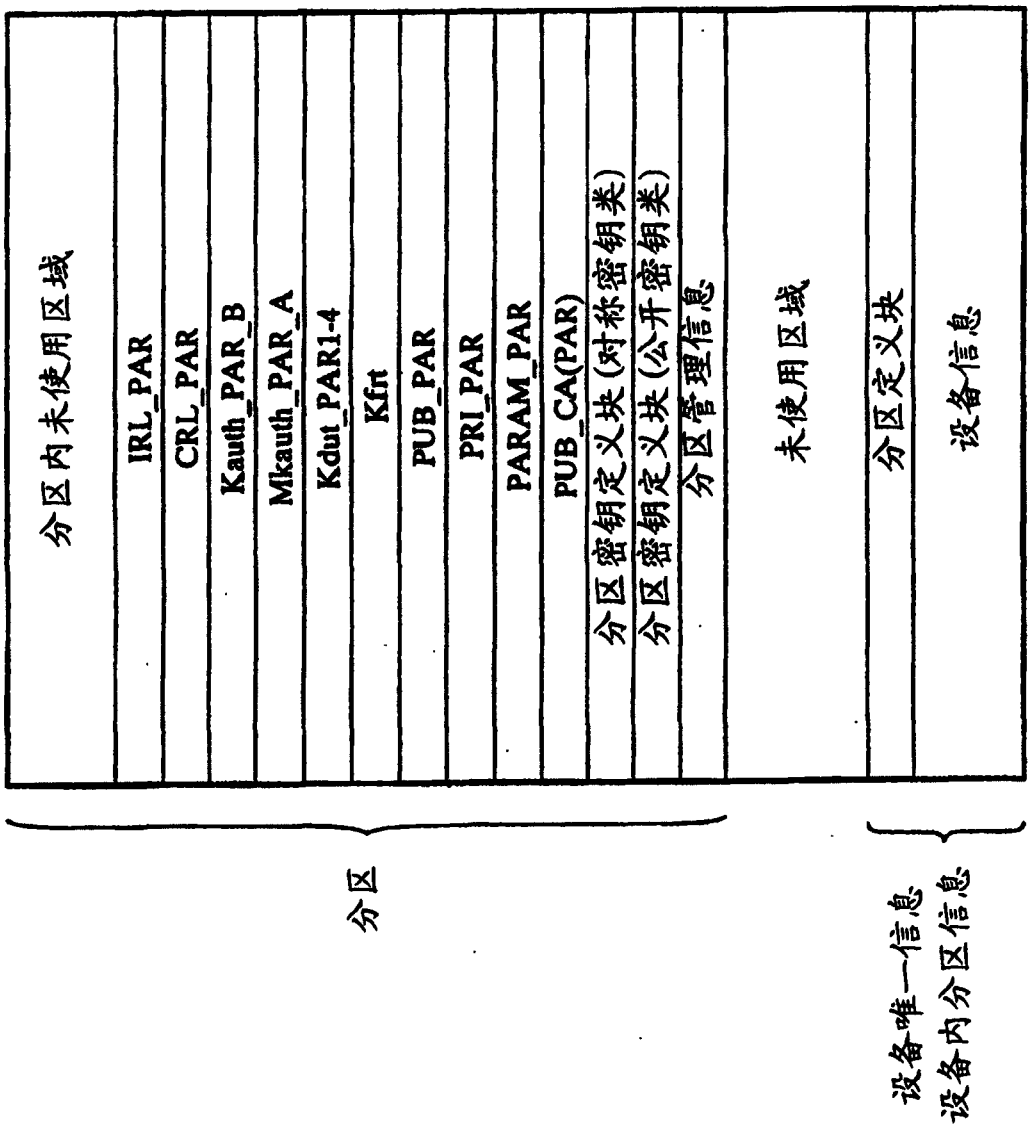

分区内未使用区域
IRL_PAR
CRL_PAR
Kauth_PAR_B
Mkauth_PAR_A
Kdut_PAR1-4
Kfrt
PUB_PAR
PRI_PAR
PARAM_PAR
PUB_CA(PAR)
分区密钥定义块(对称密钥类)
分区密钥定义块(公开密钥类)
分区管理信息
未使用区域
分区定义块
设备信息
分区
设备唯一信息
设备内分区信息

图 65

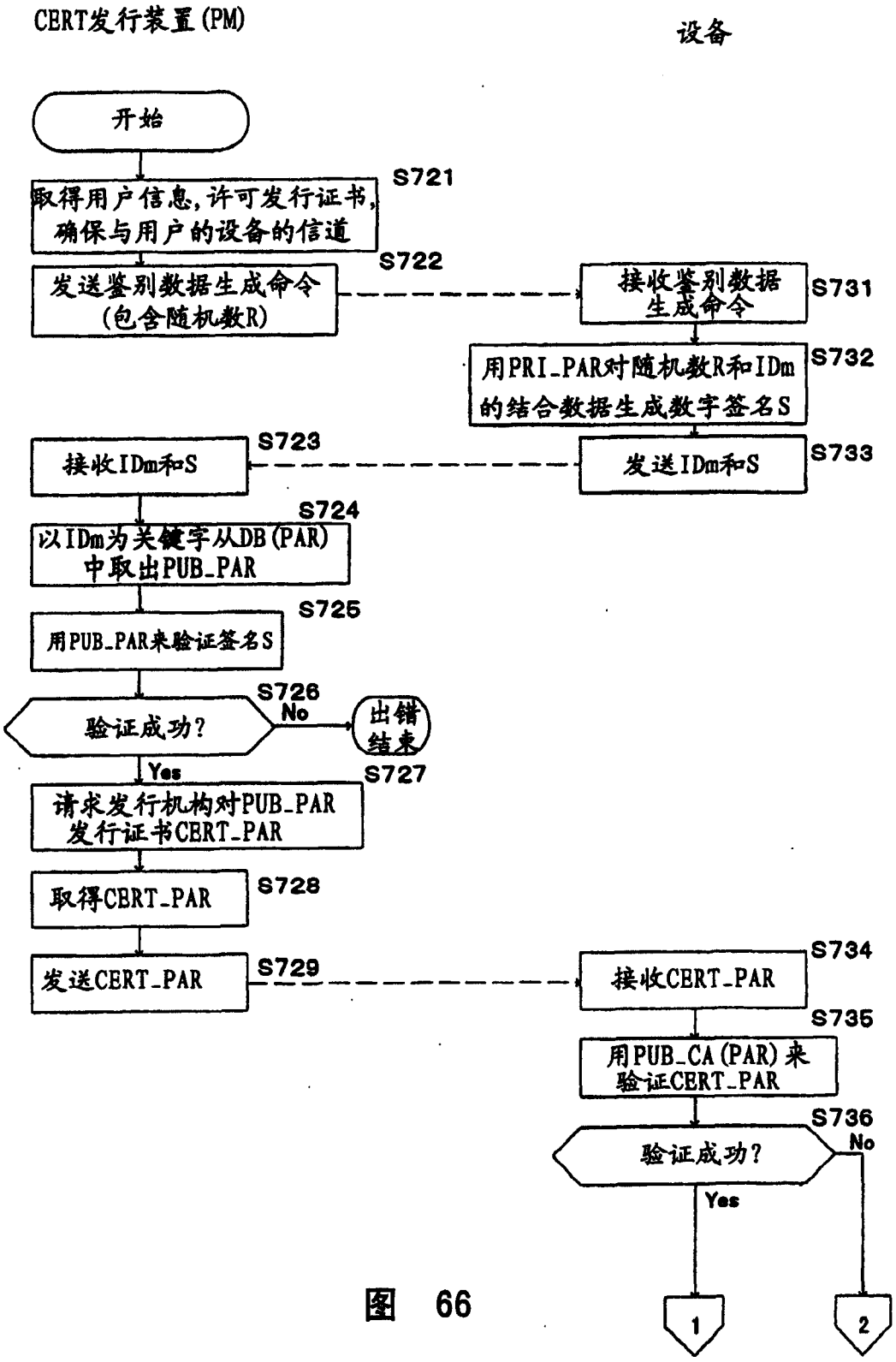

CERT发行装置(PM)
设备
开始
S721
取得用户信息,许可发行证书,确保与用户的设备的信道
S722
发送鉴别数据生成命令(包含随机数R)
接收鉴别数据生成命令
S731
用PRI_PAR对随机数R和IDm的结合数据生成数字签名S
S732
接收IDm和S
S723
发送IDm和S
S733
S724
以IDm为关键字从DB(PAR)中取出PUB_PAR
S725
用PUB_PAR来验证签名S
S726
No
验证成功?
出错结束
Yes
S727
请求发行机构对PUB_PAR发行证书CERT_PAR
S728
取得CERT_PAR
S729
发送CERT_PAR
接收CERT_PAR
S734
S735
用PUB_CA(PAR)来验证CERT_PAR
S736
No
验证成功?
Yes
1
2

图 66

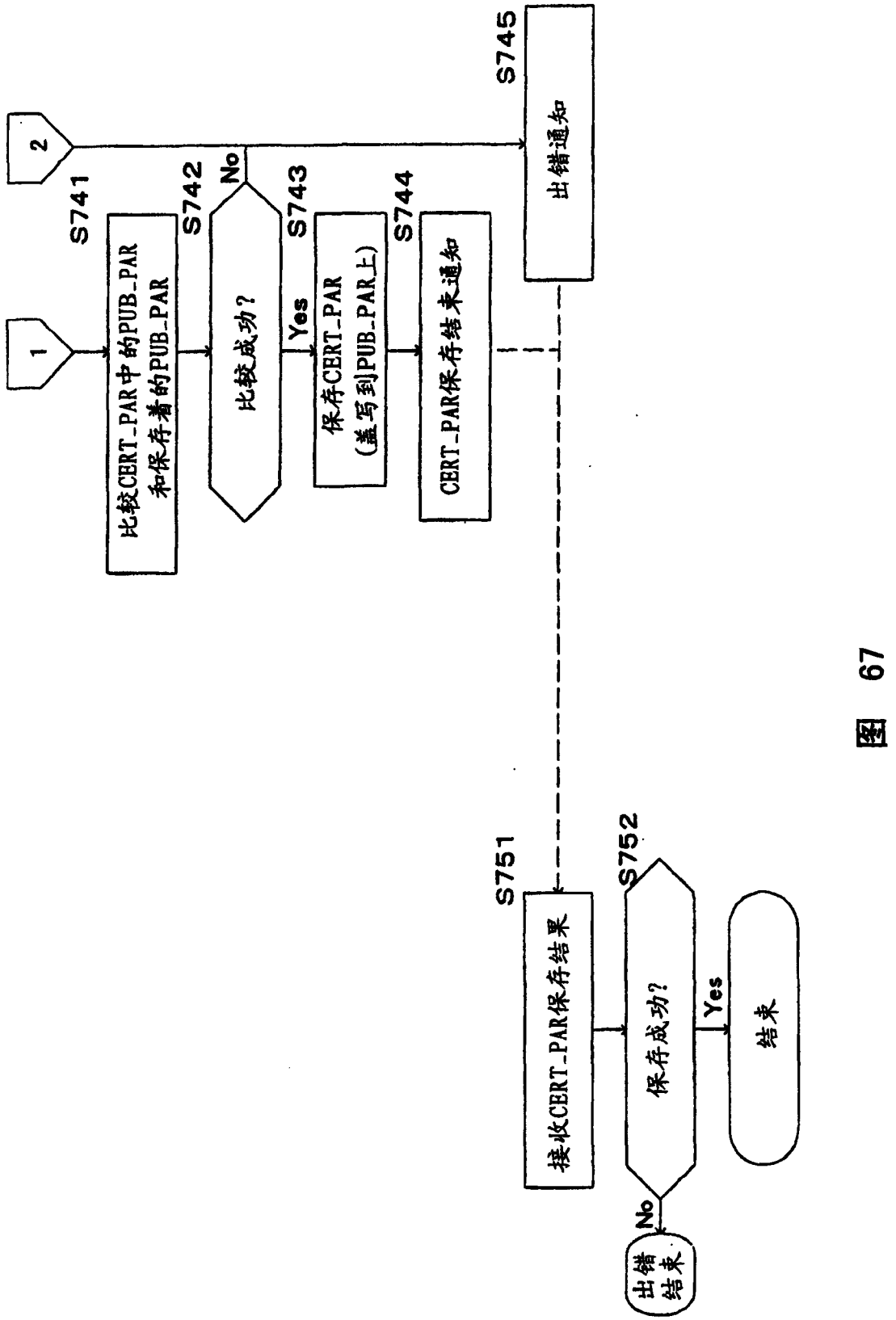

2
S741
比较CERT_PAR中的PUB_PAR和保存着的PUB_PAR
S742
比较成功?
No
Yes
S743
保存CERT_PAR(盖写到PUB_PAR上)
S744
CERT_PAR保存结束通知
S745
出错通知
1
S751
接收CERT_PAR保存结果
S752
保存成功?
No
Yes
出错结束
结束

图 67

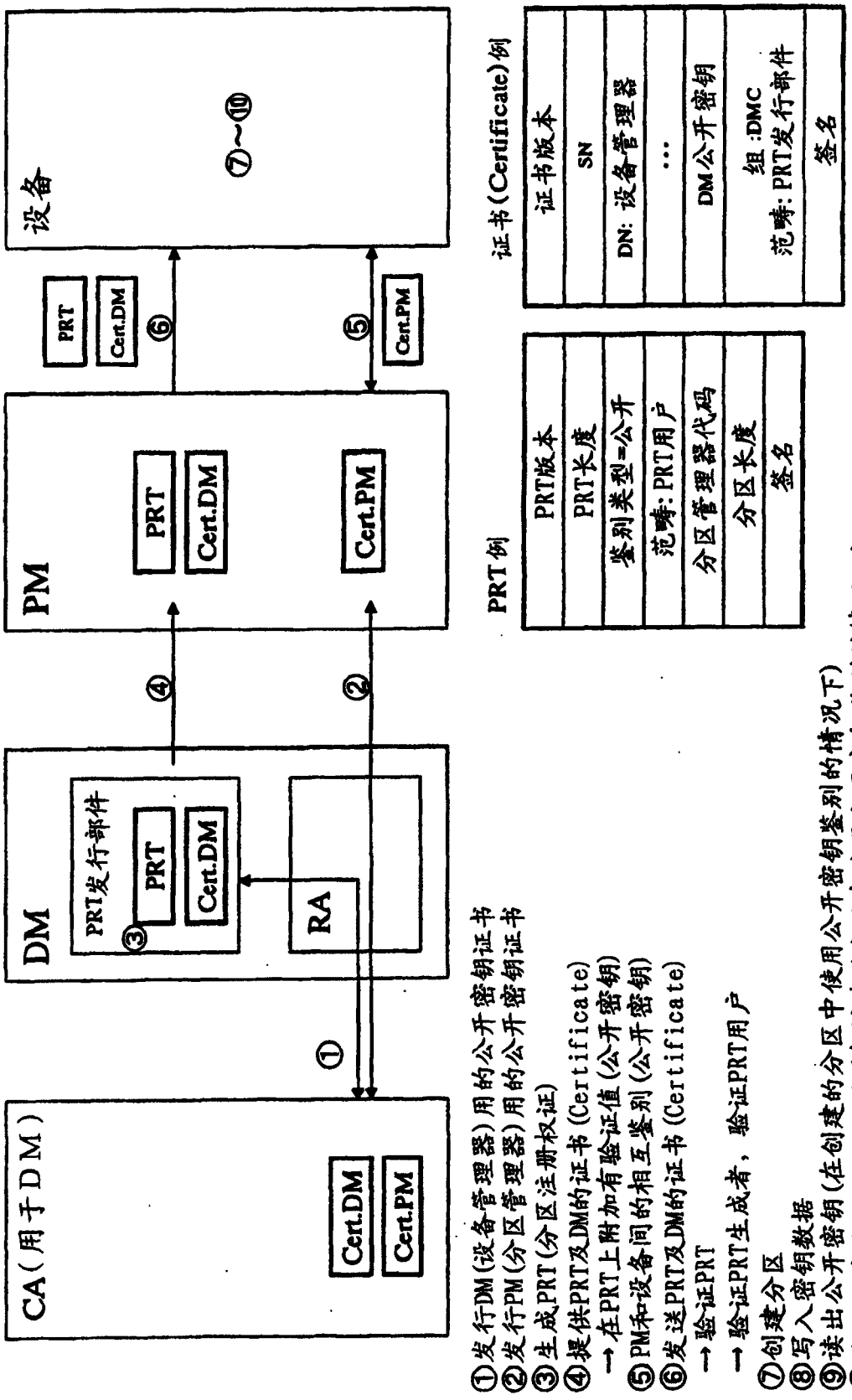

CA(用于DM)
Cert.DM
Cert.PM
DM
PRT发行部件
PRT
Cert.DM
RA
PM
PRT
Cert.DM
Cert.PM
设备
PRT
Cert.DM
Cert.PM
⑦~⑩
①
②
③
④
⑤
⑥
①发行DM(设备管理器)用的公开密钥证书
②发行PM(分区管理器)用的公开密钥证书
③生成PRT(分区注册权证)
④提供PRT及DM的证书(Certificate)
→在PRT上附加有验证值(公开密钥)
⑤PM和设备间的相互鉴别(公开密钥)
⑥发送PRT及DM的证书(Certificate)
→验证PRT
→验证PRT生成者，验证PRT用户
⑦创建分区
⑧写入密钥数据
⑨读出公开密钥(在创建的分区中使用公开密钥鉴别的情况下)
⑩发行证书(Certificate)(在创建的分区中使用公开密钥鉴别的情况下)
PRT例
PRT版本
PRT长度
鉴别类型=公开
范畴:PRT用户
分区管理器代码
分区长度
签名
证书(Certificate)例
证书版本
SN
DN:设备管理器
...
DM公开密钥
组:DMC
范畴:PRT发行部件
签名

图 68

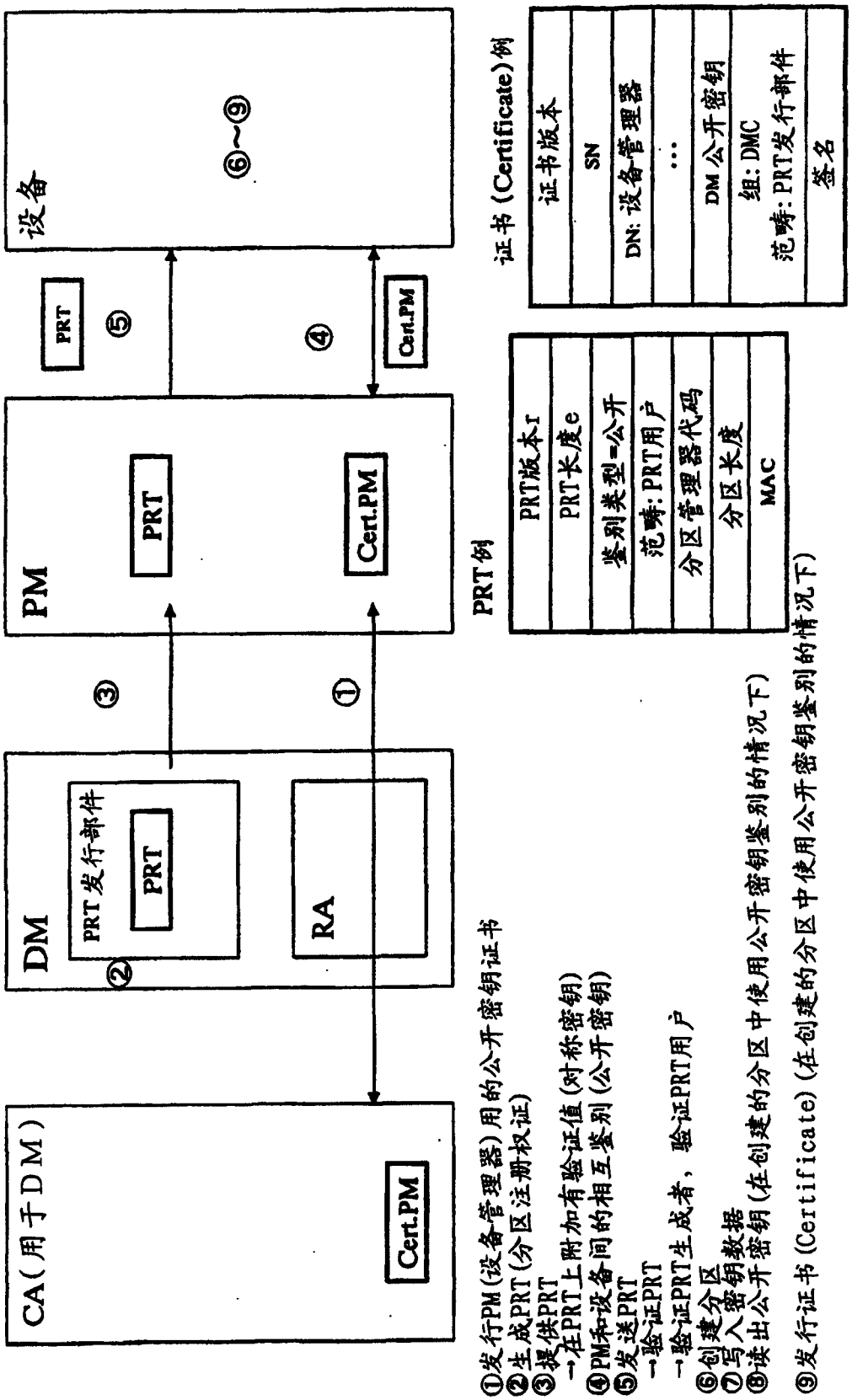

CA(用于DM)
Cert.PM
DM
PRT发行部件
PRT
RA
PM
PRT
Cert.PM
设备
PRT
Cert.PM
①
②
③
④
⑤
⑥~⑨
①发行PM(设备管理器)用的公开密钥证书
②生成PRT(分区注册权证)
③提供PRT
→在PRT上附加有验证值(对称密钥)
④PM和设备间的相互鉴别(公开密钥)
⑤发送PRT
→验证PRT
→验证PRT生成者，验证PRT用户
⑥创建分区
⑦写入密钥数据
⑧读出公开密钥(在创建的分区中使用公开密钥鉴别的情况下)
⑨发行证书(Certificate)(在创建的分区中使用公开密钥鉴别的情况下)
PRT例
PRT版本r
PRT长度e
鉴别类型=公开
范畴:PRT用户
分区管理器代码
分区长度
MAC
证书(Certificate)例
证书版本
SN
DN:设备管理器
...
DM公开密钥
组:DMC
范畴:PRT发行部件
签名

图 69

CA(用于DM)
DM
PRT发行部件
PRT
RA
①
②
PM
PRT
③
④
PRT
设备
⑤~⑧
PRT例
PRT版本
PRT长度
鉴别类型=对称
范畴:PRT用户
分区管理器代码
分区长度
MAC
证书(Certificate)例
证书版本
SN
DN:设备管理器
…
DM公开密钥
组:DMC
范畴:PRT发行部件
签名
①生成PRT(分区注册权证)
②提供PRT
→在PRT上附加有验证值(对称密钥)
③PM和设备间的相互鉴别(对称密钥)
④发送PRT
→验证PRT
→验证PRT生成者,验证PRT用户
⑤创建分区
⑥写入密钥数据
⑦读出公开密钥(在创建的分区中使用公开密钥鉴别的情况下)
⑧发行证书(Certificate)(在创建的分区中使用公开密钥鉴别的情况下)

图 70

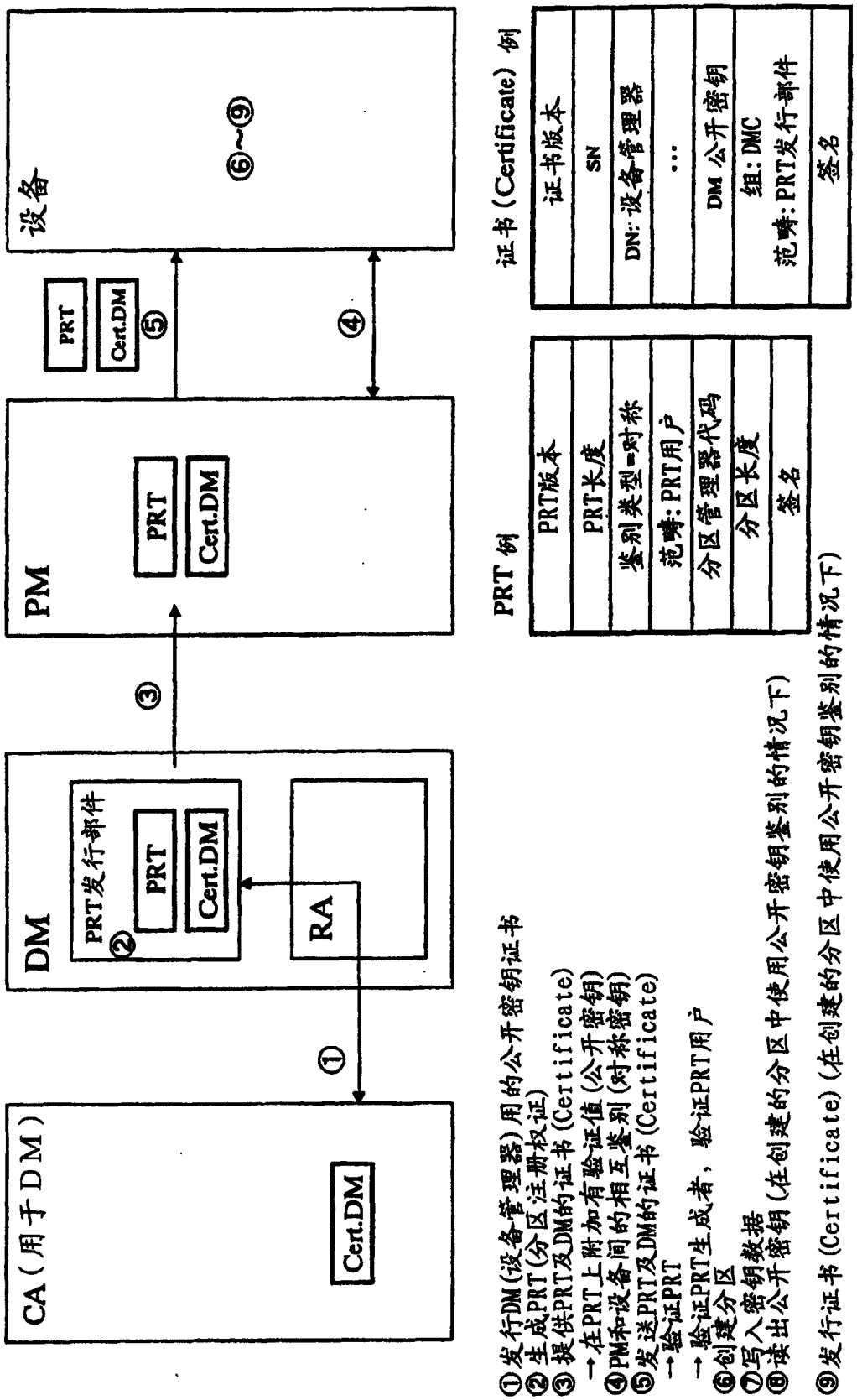

CA（用于DM）
Cert.DM
①
DM
PRT发行部件
②
PRT
Cert.DM
RA
③
PM
PRT
Cert.DM
④
PRT
Cert.DM
⑤
设备
⑥~⑨
①发行DM(设备管理器)用的公开密钥证书
②生成PRT(分区注册权证)
③提供PRT及DM的证书(Certificate)
→在PRT上附加有验证值(公开密钥)
④PM和设备间的相互鉴别(对称密钥)
⑤发送PRT及DM的证书(Certificate)
→验证PRT
→验证PRT生成者，验证PRT用户
⑥创建分区
⑦写入密钥数据
⑧读出公开密钥(在创建的分区中使用公开密钥鉴别的情况下)
⑨发行证书(Certificate)(在创建的分区中使用公开密钥鉴别的情况下)
PRT 例
PRT版本
PRT长度
鉴别类型=对称
范畴: PRT用户
分区管理器代码
分区长度
签名
证书（Certificate）例
证书版本
SN
DN: 设备管理器
...
DM 公开密钥
组: DMC
范畴: PRT发行部件
签名

图 71

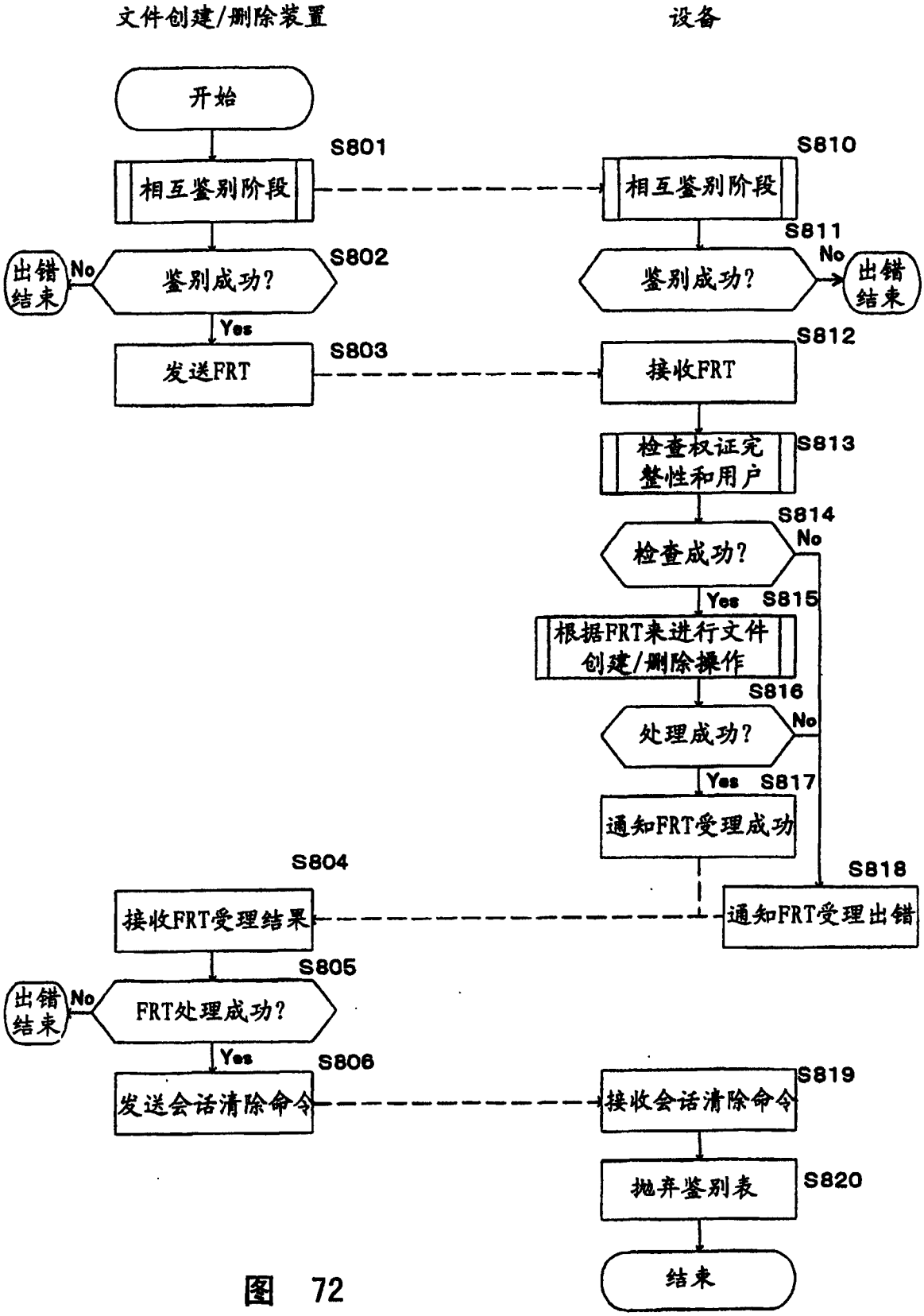

文件创建/删除装置
设备
开始
S801
相互鉴别阶段
S810
相互鉴别阶段
S811
鉴别成功?
No
出错结束
S802
鉴别成功?
No
出错结束
Yes
S803
发送FRT
S812
接收FRT
检查权证完整性和用户
S813
S814
检查成功?
No
Yes
S815
根据FRT来进行文件创建/删除操作
S816
处理成功?
No
Yes
S817
通知FRT受理成功
S804
S818
接收FRT受理结果
通知FRT受理出错
S805
出错结束
No
FRT处理成功?
Yes
S806
发送会话清除命令
S819
接收会话清除命令
抛弃鉴别表
S820
结束

图 72

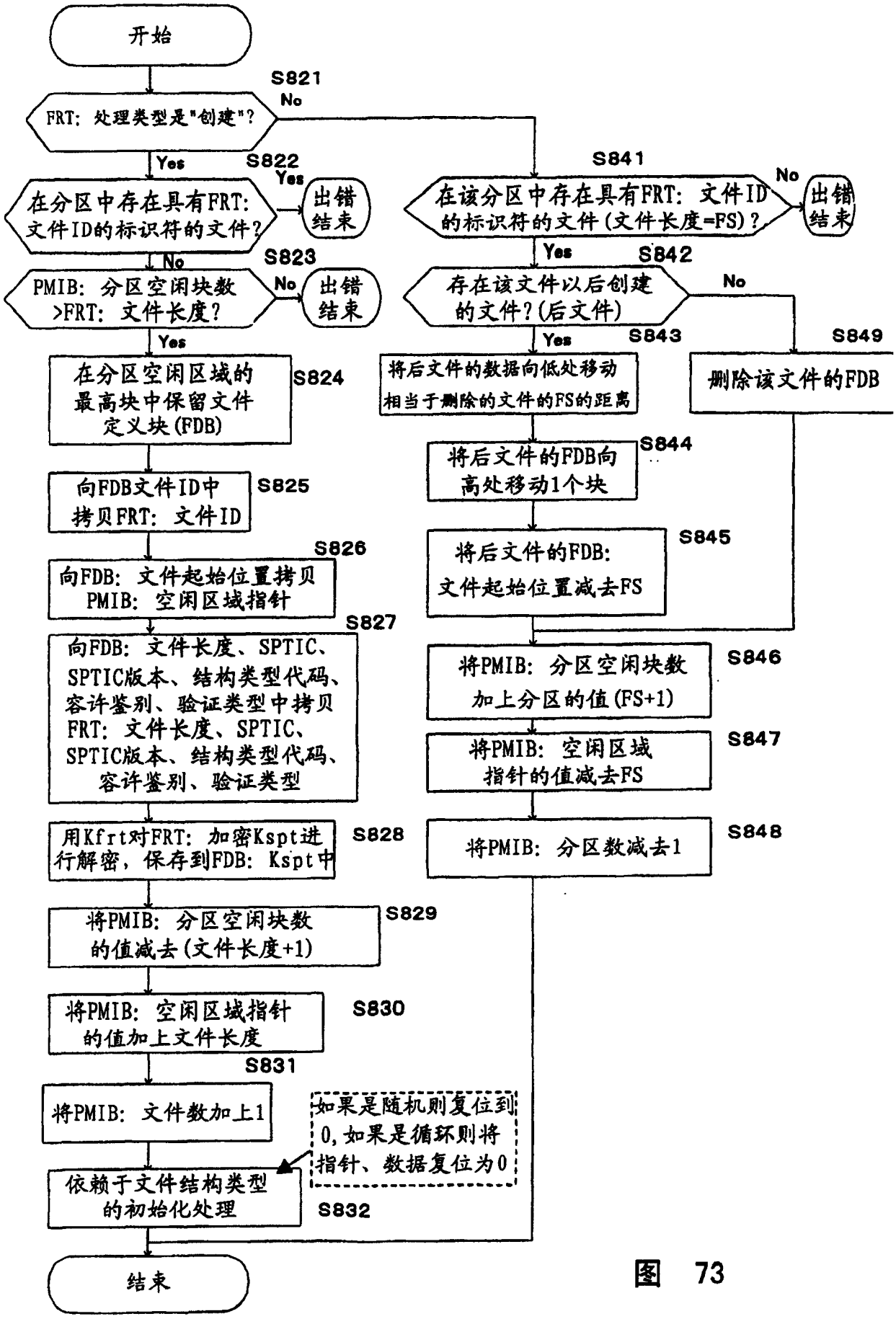

开始
S821
FRT: 处理类型是"创建"?
No
Yes
S822
在分区中存在具有FRT: 文件ID的标识符的文件?
Yes
出错结束
No
S823
PMIB: 分区空闲块数>FRT: 文件长度?
No
出错结束
Yes
在分区空闲区域的最高块中保留文件定义块(FDB)
S824
向FDB文件ID中拷贝FRT: 文件ID
S825
向FDB: 文件起始位置拷贝PMIB: 空闲区域指针
S826
向FDB: 文件长度、SPTIC、SPTIC版本、结构类型代码、容许鉴别、验证类型中拷贝FRT: 文件长度、SPTIC、SPTIC版本、结构类型代码、容许鉴别、验证类型
S827
用Kfrt对FRT: 加密Kspt进行解密，保存到FDB: Kspt中
S828
将PMIB: 分区空闲块数的值减去(文件长度+1)
S829
将PMIB: 空闲区域指针的值加上文件长度
S830
将PMIB: 文件数加上1
S831
依赖于文件结构类型的初始化处理
S832
如果是随机则复位到0,如果是循环则将指针、数据复位为0
结束
S841
在该分区中存在具有FRT: 文件ID的标识符的文件(文件长度=FS)?
No
出错结束
Yes
S842
存在该文件以后创建的文件?(后文件)
No
S849
删除该文件的FDB
Yes
S843
将后文件的数据向低处移动相当于删除的文件的FS的距离
S844
将后文件的FDB向高处移动1个块
S845
将后文件的FDB: 文件起始位置减去FS
S846
将PMIB: 分区空闲块数加上分区的值(FS+1)
S847
将PMIB: 空闲区域指针的值减去FS
S848
将PMIB: 分区数减去1

图 73

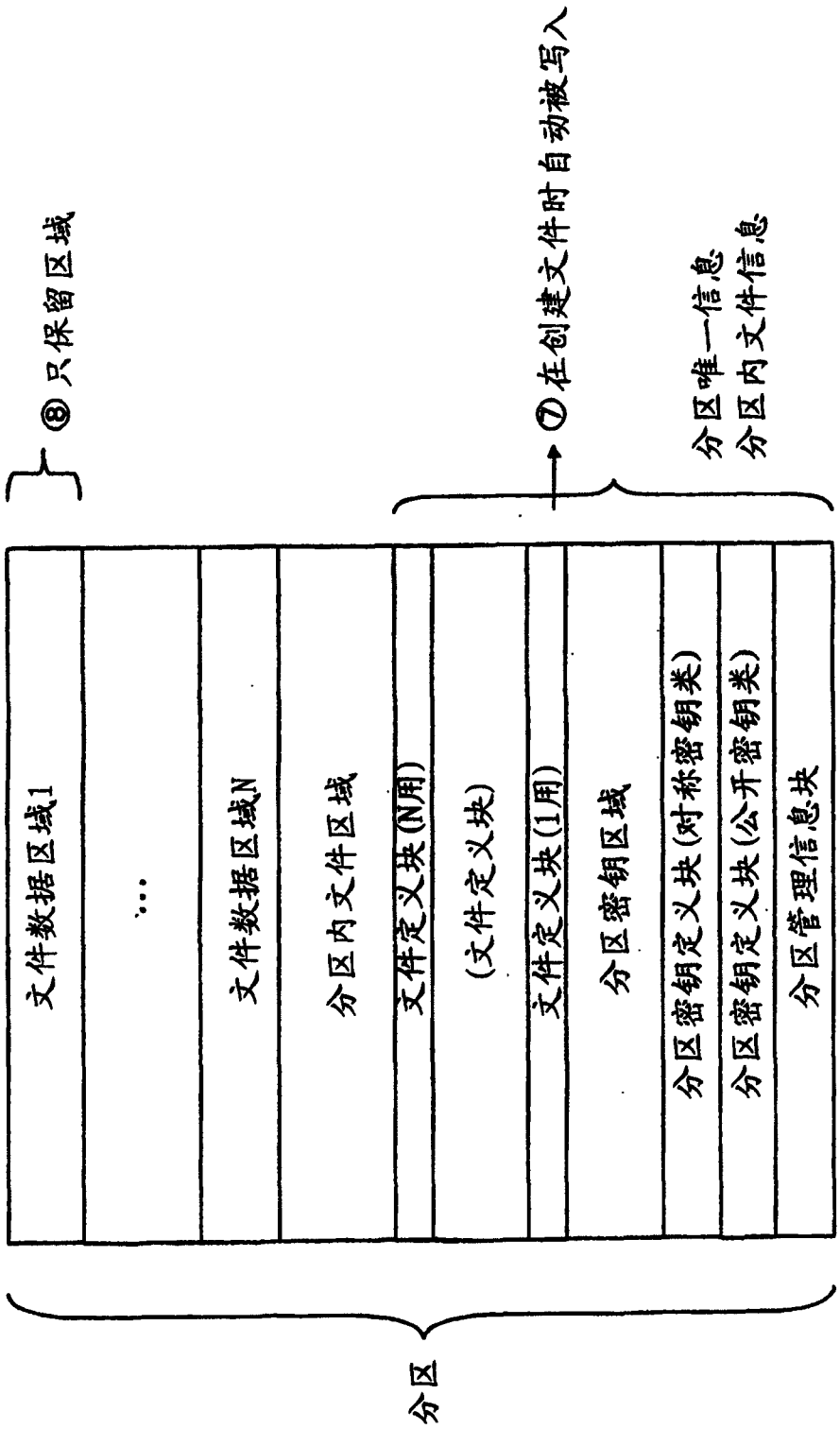
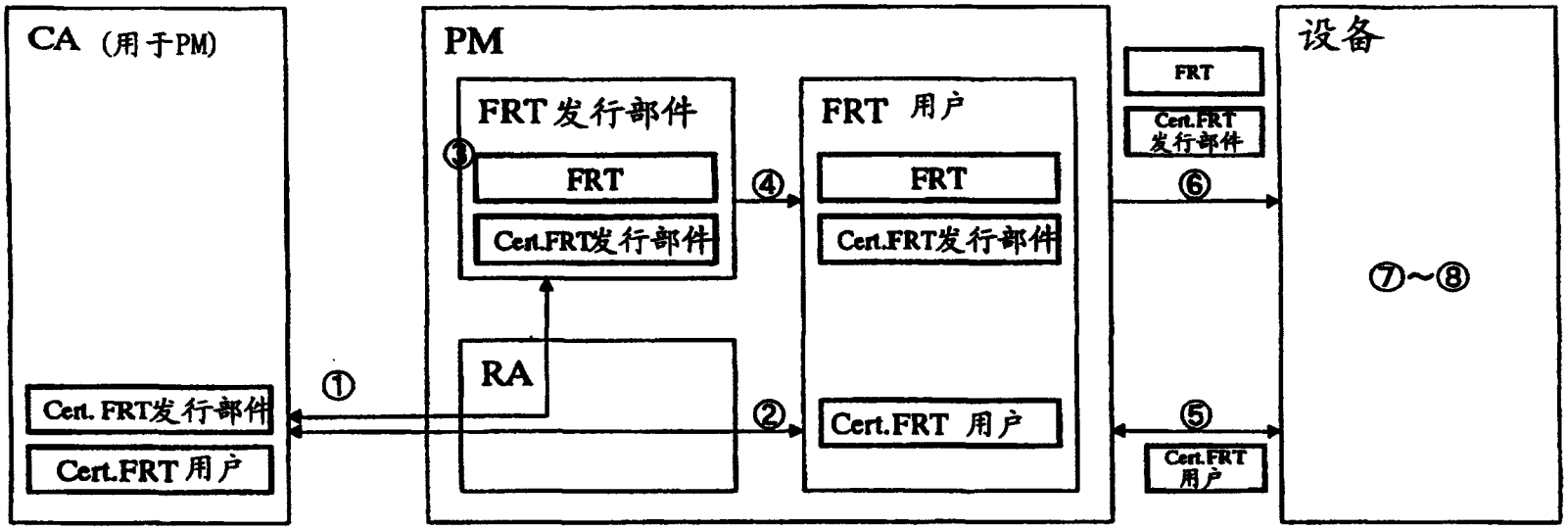

分区
文件数据区域1
...
文件数据区域N
分区内文件区域
文件定义块(N用)
(文件定义块)
文件定义块(1用)
分区密钥区域
分区密钥定义块(对称密钥类)
分区密钥定义块(公开密钥类)
分区管理信息块
⑦在创建文件时自动被写入
分区唯一信息
分区内文件信息
⑧只保留区域

图 74



①发行FRT发行部件用的公开密钥证书
②发行FRT用户用的公开密钥证书
③生成FRT(文件注册权证)
④提供FRT及FRT发行部件用的证书(Certificate)
→在FRT上附加有验证值(公开密钥)
⑤FRT发行部件和设备间的相互鉴别(公开密钥)
⑥发送FRT及FRT发行部件用的证书(Certificate)
→验证FRT
→验证FRT生成者，验证FRT用户
⑦向文件定义块中注册SPT发行部件范畴和Kspt
⑧保留文件数据区域

FRT 例

…
鉴别类型=公开
范畴: FRT用户
分区管理器代码
文件信息
范畴: SPT发行部件
Kfr(Kspt)
签名

证书 (Certificate) 例

证书版本
SN
DN: 设备管理器
…
PM公开密钥
组: DMC 范畴: FRT发行部件
签名

图 75

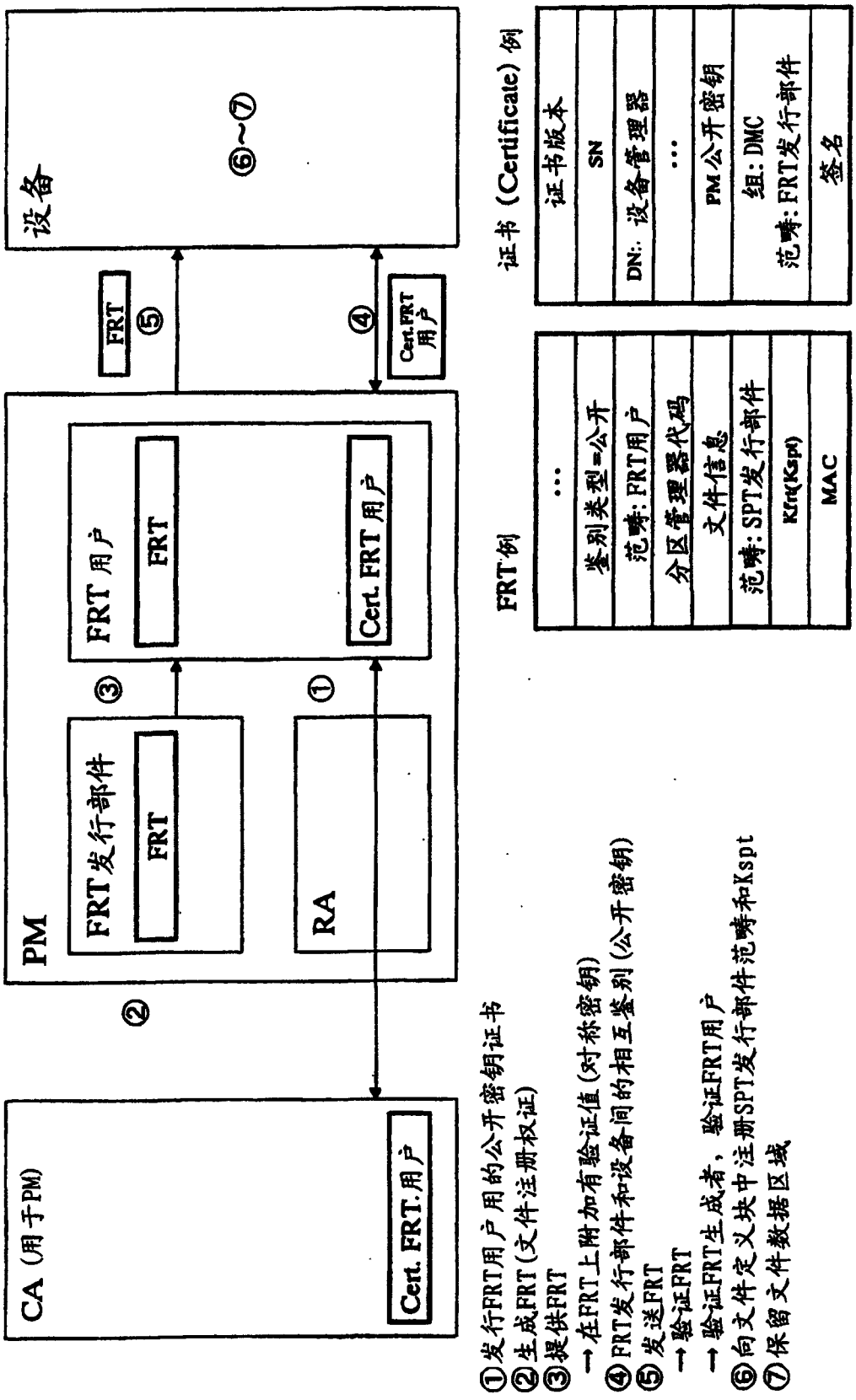

设备
⑥~⑦
FRT
⑤
④
Cert.FRT
用户
PM
FRT发行部件
FRT
③
FRT用户
FRT
Cert. FRT用户
②
①
RA
CA (用于PM)
Cert. FRT.用户
①发行FRT用户用的公开密钥证书
②生成FRT(文件注册权证)
③提供FRT
→在FRT上附加有验证值(对称密钥)
④FRT发行部件和设备间的相互鉴别(公开密钥)
⑤发送FRT
→验证FRT
→验证FRT生成者，验证FRT用户
⑥向文件定义块中注册SPT发行部件范畴和Kspt
⑦保留文件数据区域
FRT例
...
鉴别类型=公开
范畴: FRT用户
分区管理器代码
文件信息
范畴: SPT发行部件
Kfrt(Kspt)
MAC
证书 (Certificate) 例
证书版本
SN
DN: 设备管理器
...
PM公开密钥
组: DMC
范畴: FRT发行部件
签名

图 76

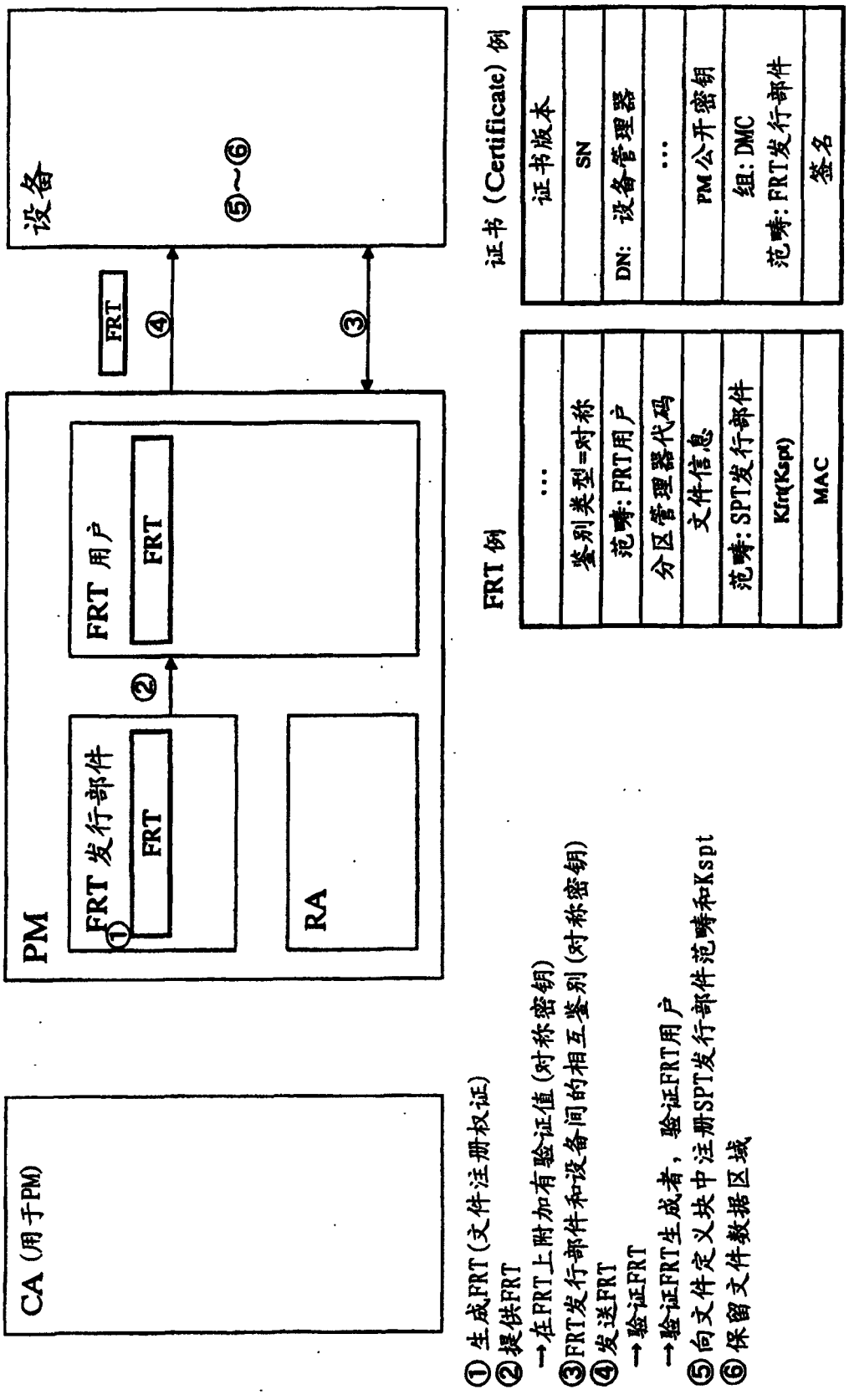
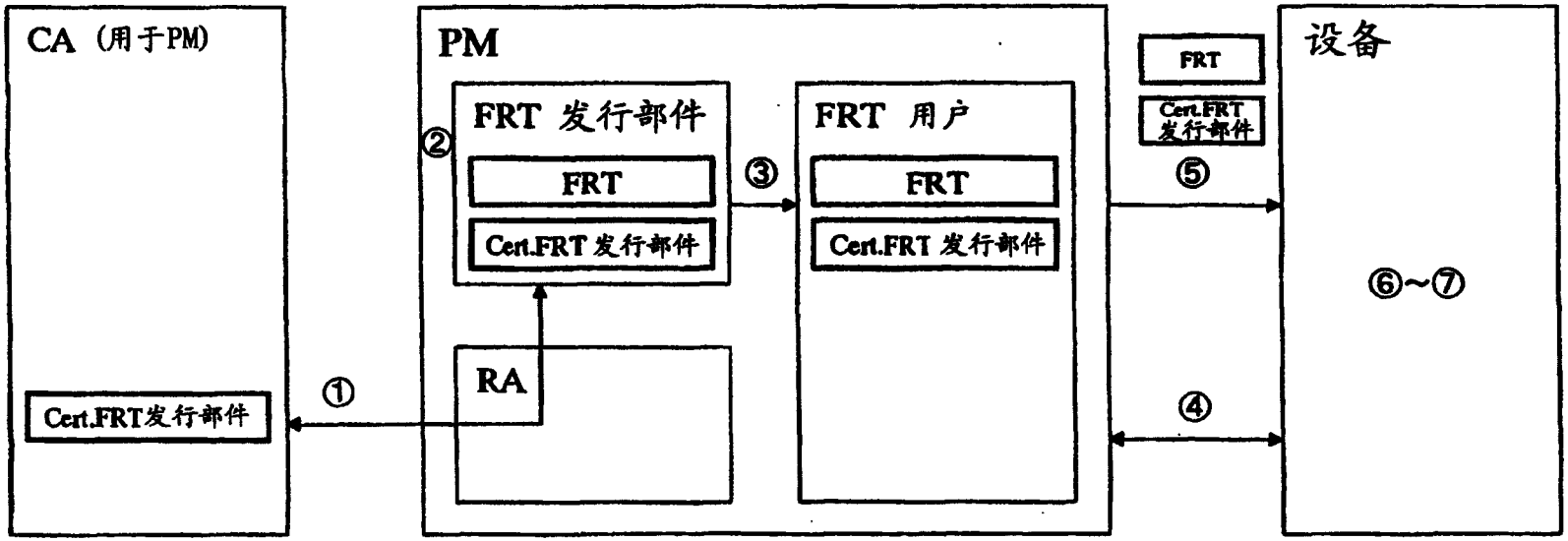

CA (用于PM)
PM
FRT 发行部件
FRT
RA
FRT 用户
FRT
①
②
③
④
⑤~⑥
FRT
设备
FRT 例
...
鉴别类型=对称
范畴: FRT用户
分区管理器代码
文件信息
范畴: SPT发行部件
Kirt(Kspt)
MAC
证书 (Certificate) 例
证书版本
SN
DN: 设备管理器
...
PM公开密钥
组: DMC
范畴: FRT发行部件
签名
①生成FRT(文件注册权证)
②提供FRT
→在FRT上附加有验证值(对称密钥)
③FRT发行部件和设备间的相互鉴别(对称密钥)
④发送FRT
→验证FRT
→验证FRT生成者，验证FRT用户
⑤向文件定义块中注册SPT发行部件范畴和Kspt
⑥保留文件数据区域

图 77



①发行FRT发行部件用的公开密钥证书
②生成FRT(文件注册权证)
③提供FRT及FRT发行部件用的证书(Certificate)
→在FRT上附加有验证值(公开密钥)
④FRT发行部件和设备间的相互鉴别(对称密钥)
⑤发送FRT及FRT发行部件用的证书(Certificate)
→验证FRT
→验证FRT生成者，验证FRT用户
⑥向文件定义块中注册SPT发行部件范畴和Kspt
⑦保留文件数据区域

FRT 例

…
鉴别类型=对称
范畴: FRT用户
分区管理器代码
文件信息
范畴: SPT发行部件
Kfr(Kspt)

证书 (Certificate) 例

证书版本
SN
DN: 设备管理器
…
PM 公开密钥
组: DMC 范畴: FRT发行部件
签名

图 78

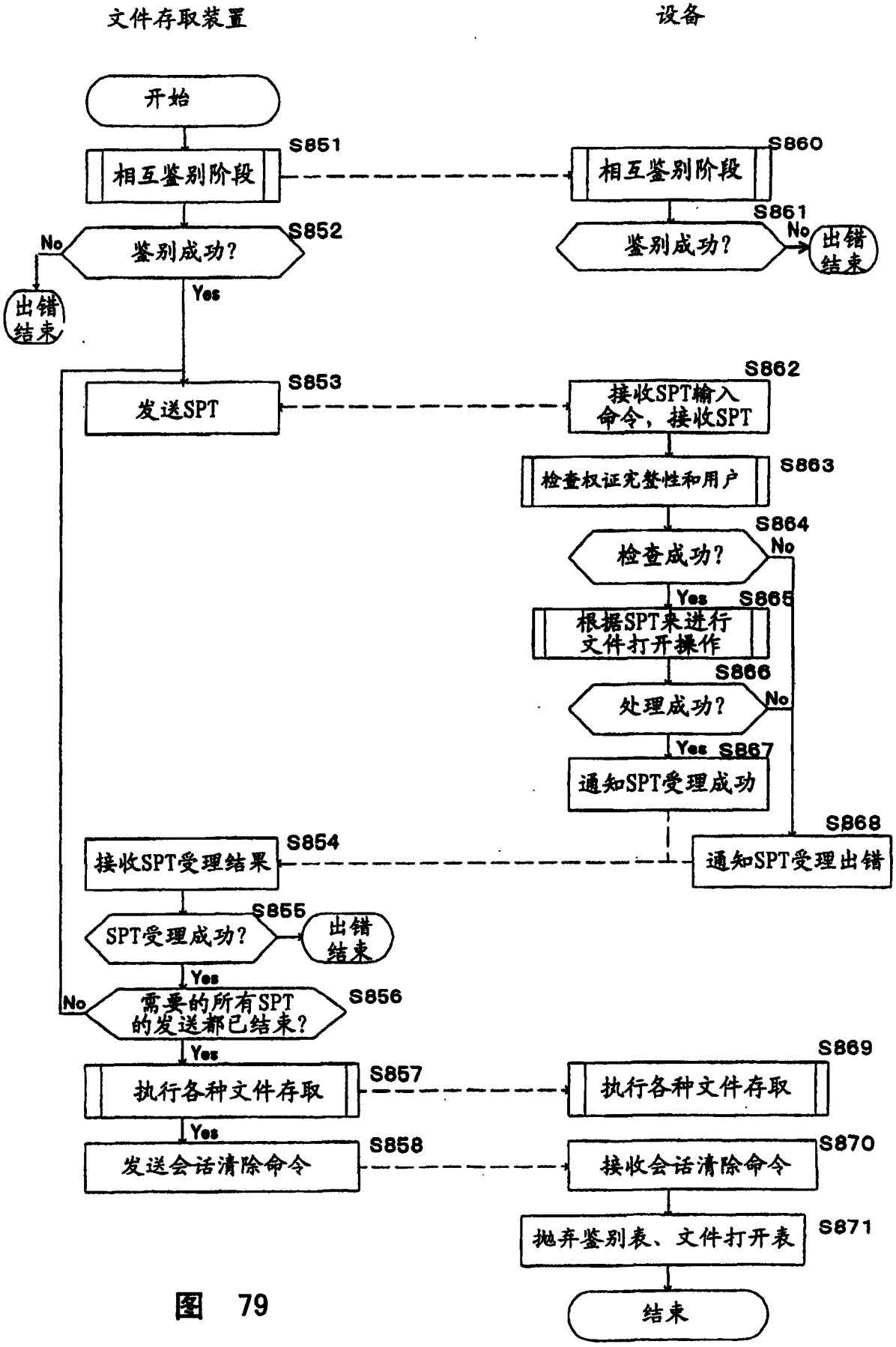

文件存取装置
设备
开始
S851
相互鉴别阶段
S860
相互鉴别阶段
S861
鉴别成功?
No
出错结束
S852
鉴别成功?
No
出错结束
Yes
S853
发送SPT
S862
接收SPT输入命令，接收SPT
S863
检查权证完整性和用户
S864
检查成功?
No
Yes
S865
根据SPT来进行文件打开操作
S866
处理成功?
No
Yes
S867
通知SPT受理成功
S868
通知SPT受理出错
S854
接收SPT受理结果
S855
SPT受理成功?
出错结束
Yes
No
需要的所有SPT的发送都已结束?
S856
Yes
S857
执行各种文件存取
S869
执行各种文件存取
Yes
S858
发送会话清除命令
S870
接收会话清除命令
S871
抛弃鉴别表、文件打开表
结束

图 79

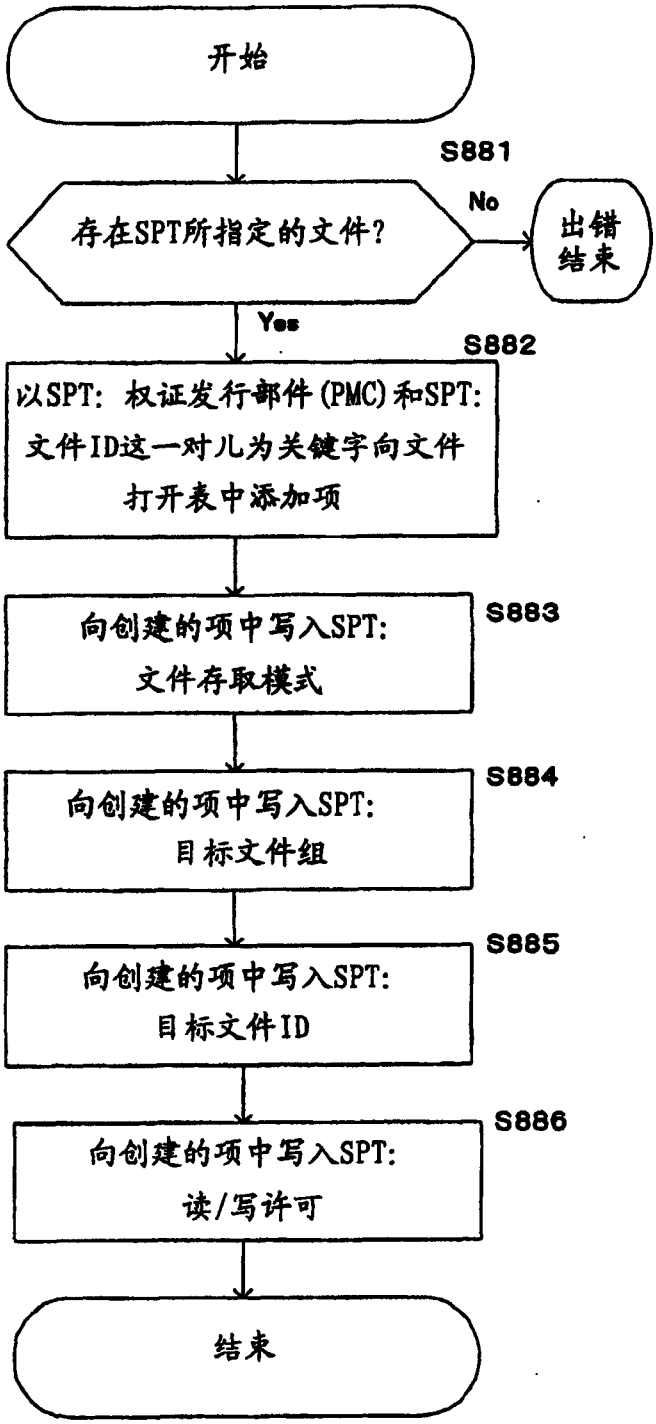
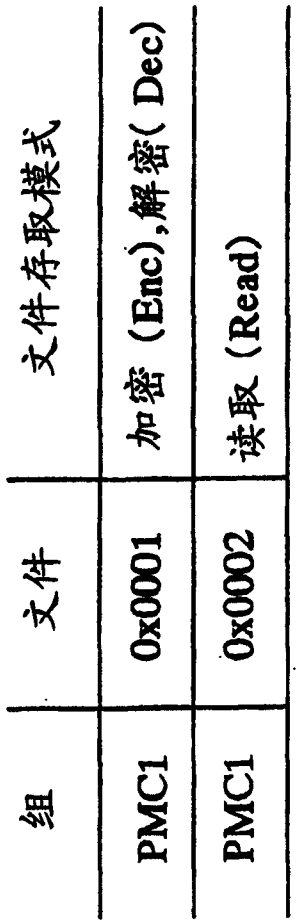
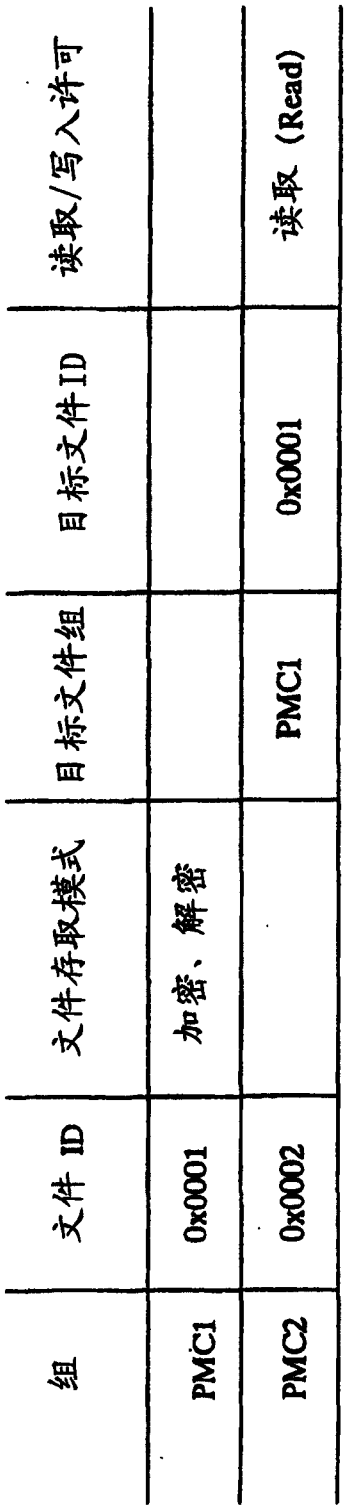

开始
S881
存在SPT所指定的文件?
No
出错结束
Yes
S882
以SPT: 权证发行部件(PMC)和SPT: 文件ID这一对儿为关键字向文件打开表中添加项
向创建的项中写入SPT: 文件存取模式
S883
向创建的项中写入SPT: 目标文件组
S884
向创建的项中写入SPT: 目标文件ID
S885
向创建的项中写入SPT: 读/写许可
S886
结束

图 80



组	文件	文件存取模式
PMC1	0x0001	加密 (Enc),解密(Dec)
PMC1	0x0002	读取 (Read)

图 81



组	文件 ID	文件存取模式	目标文件组	目标文件ID	读取/写入许可
PMC1	0x0001	加密、解密			
PMC2	0x0002		PMC1	0x0001	读取（Read）

图 82

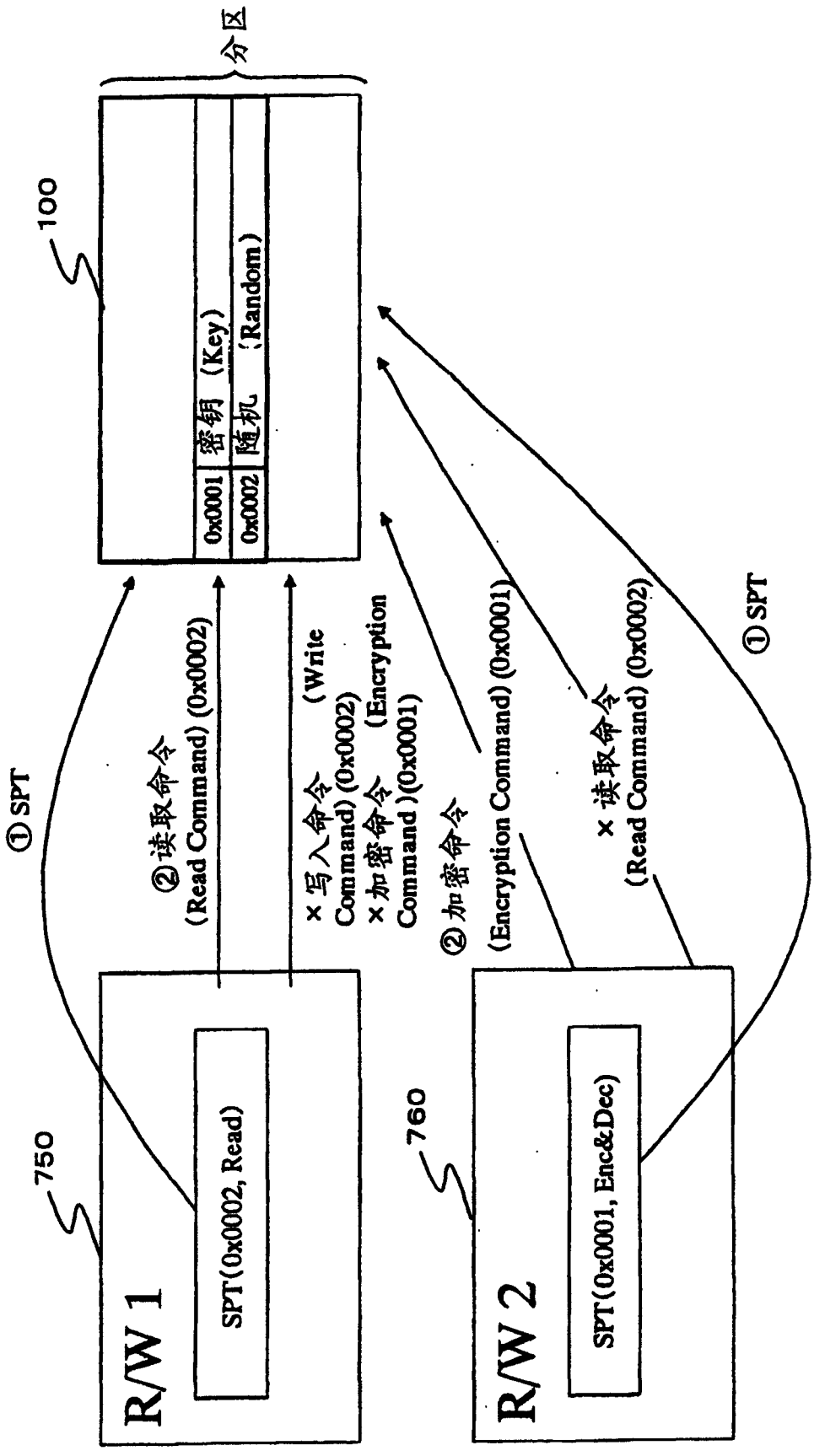

分区
100
0x0001
密钥 (Key)
0x0002
随机 (Random)
①SPT
②读取命令
(Read Command) (0x0002)
×写入命令
(Write
Command) (0x0002)
×加密命令
(Encryption
Command)(0x0001)
②加密命令
(Encryption Command) (0x0001)
×读取命令
(Read Command) (0x0002)
①SPT
750
R/W 1
SPT(0x0002, Read)
760
R/W 2
SPT(0x0001, Enc&Dec)

图 83

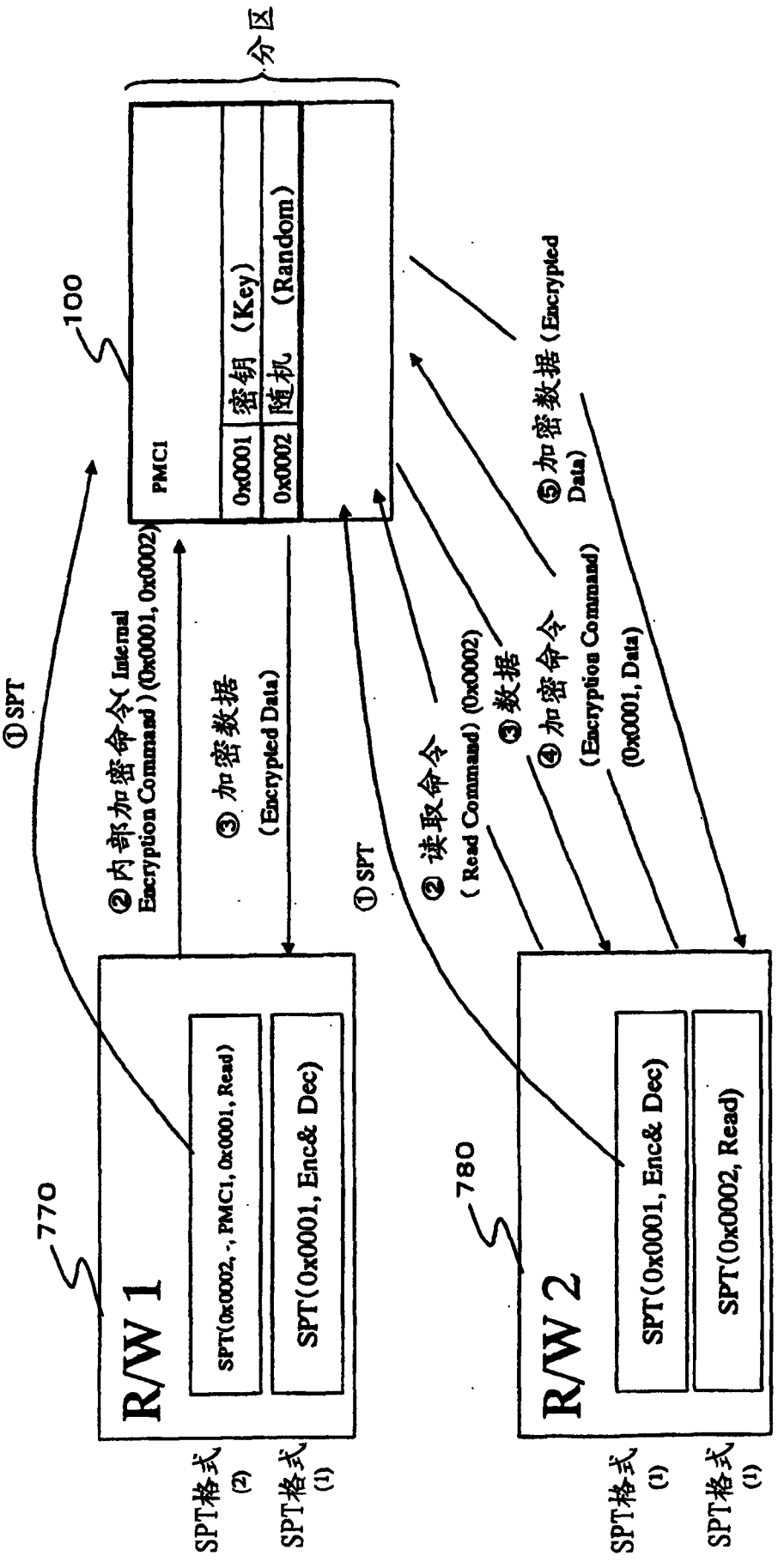
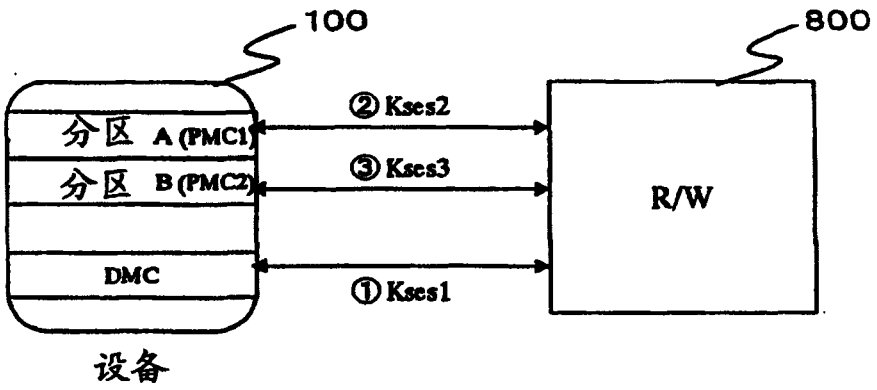

分区
100
PMC1
0x0001
密钥 (Key)
0x0002
随机 (Random)
①SPT
②内部加密命令(Internal Encryption Command)(0x0001, 0x0002)
③加密数据 (Encrypted Data)
①SPT
②读取命令 (Read Command)(0x0002)
③数据
④加密命令 (Encryption Command)(0x0001, Data)
⑤加密数据(Encrypted Data)
770
R/W 1
SPT(0x0002, -, PMC1, 0x0001, Read)
SPT(0x0001, Enc& Dec)
SPT格式 (2)
SPT格式 (1)
780
R/W 2
SPT(0x0001, Enc& Dec)
SPT(0x0002, Read)
SPT格式 (1)
SPT格式 (1)

图 84



1. 会话密钥 = Kses1 ⊕ Kses2 ⊕ Kses3

2. 会话密钥 = Kses3 ※·将最后一个会话密钥用作统一的会话密钥

⊕ ：逻辑“异或”处理（以8字节为单位）

图 85

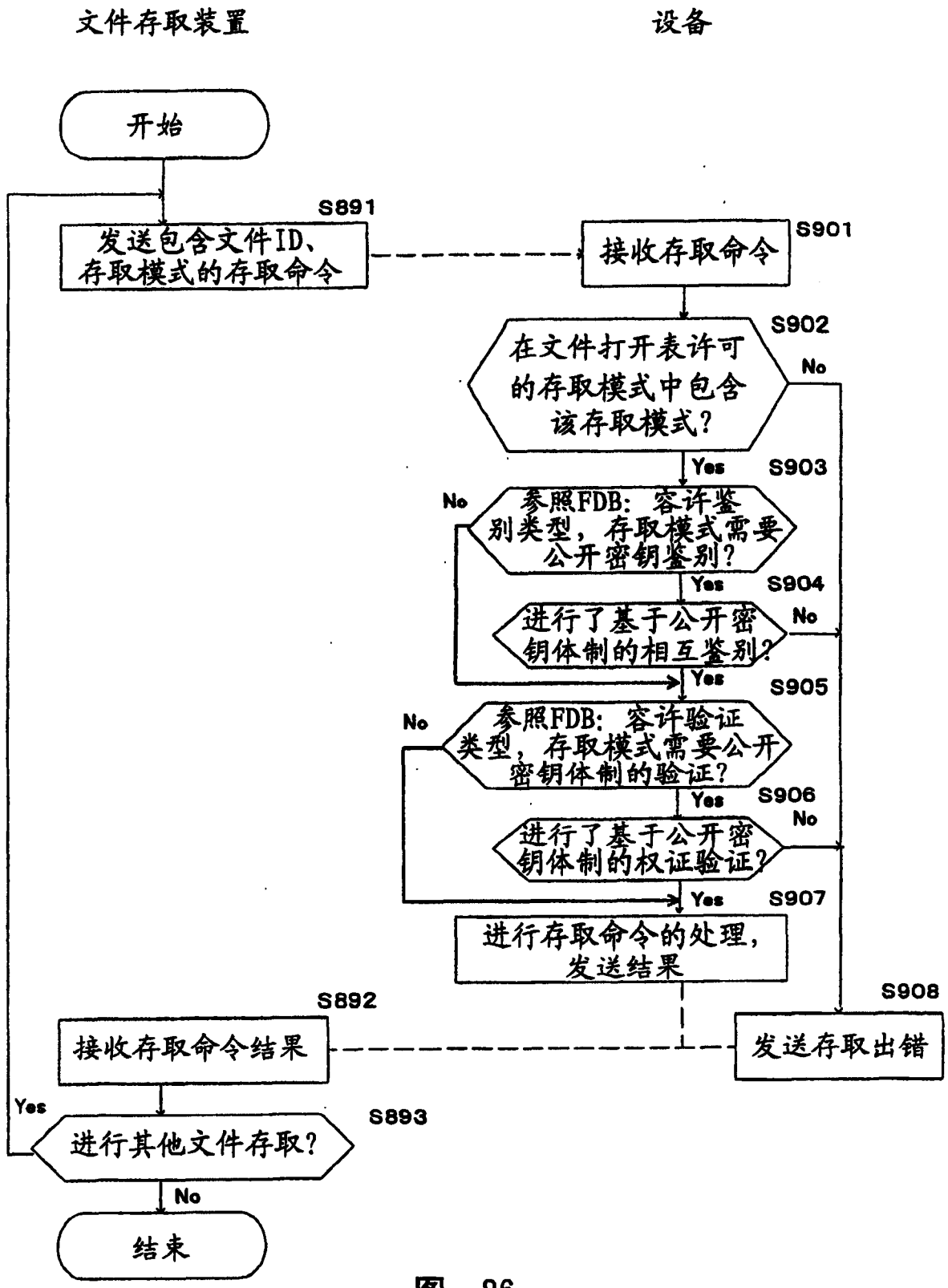

文件存取装置
设备
开始
S891
发送包含文件ID、存取模式的存取命令
S901
接收存取命令
S902
在文件打开表许可的存取模式中包含该存取模式?
No
Yes
S903
No
参照FDB: 容许鉴别类型，存取模式需要公开密钥鉴别?
Yes
S904
进行了基于公开密钥体制的相互鉴别?
No
Yes
S905
No
参照FDB: 容许验证类型，存取模式需要公开密钥体制的验证?
Yes
S906
进行了基于公开密钥体制的权证验证?
No
Yes
S907
进行存取命令的处理，发送结果
S908
发送存取出错
S892
接收存取命令结果
Yes
S893
进行其他文件存取?
No
结束

图 86

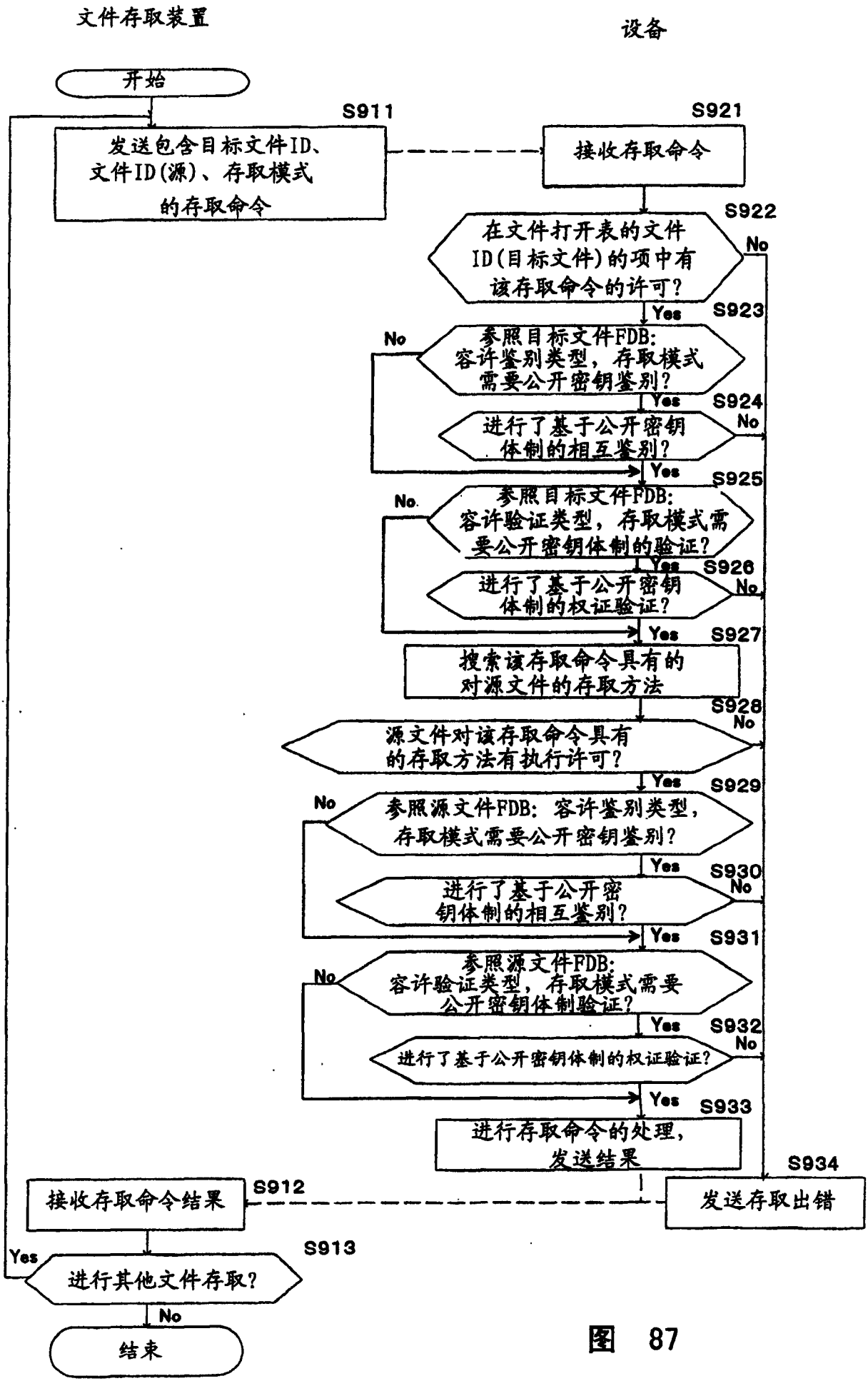

文件存取装置
设备
开始
S911
发送包含目标文件ID、文件ID(源)、存取模式的存取命令
S921
接收存取命令
S922
在文件打开表的文件ID(目标文件)的项中有该存取命令的许可?
No
Yes
S923
参照目标文件FDB:容许鉴别类型，存取模式需要公开密钥鉴别?
No
Yes
S924
进行了基于公开密钥体制的相互鉴别?
No
Yes
S925
参照目标文件FDB:容许验证类型，存取模式需要公开密钥体制的验证?
No
Yes
S926
进行了基于公开密钥体制的权证验证?
No
Yes
S927
搜索该存取命令具有的对源文件的存取方法
S928
源文件对该存取命令具有的存取方法有执行许可?
No
Yes
S929
参照源文件FDB:容许鉴别类型，存取模式需要公开密钥鉴别?
No
Yes
S930
进行了基于公开密钥体制的相互鉴别?
No
Yes
S931
参照源文件FDB:容许验证类型，存取模式需要公开密钥体制验证?
No
Yes
S932
进行了基于公开密钥体制的权证验证?
No
Yes
S933
进行存取命令的处理，发送结果
S934
发送存取出错
S912
接收存取命令结果
S913
Yes
进行其他文件存取?
No
结束

图 87

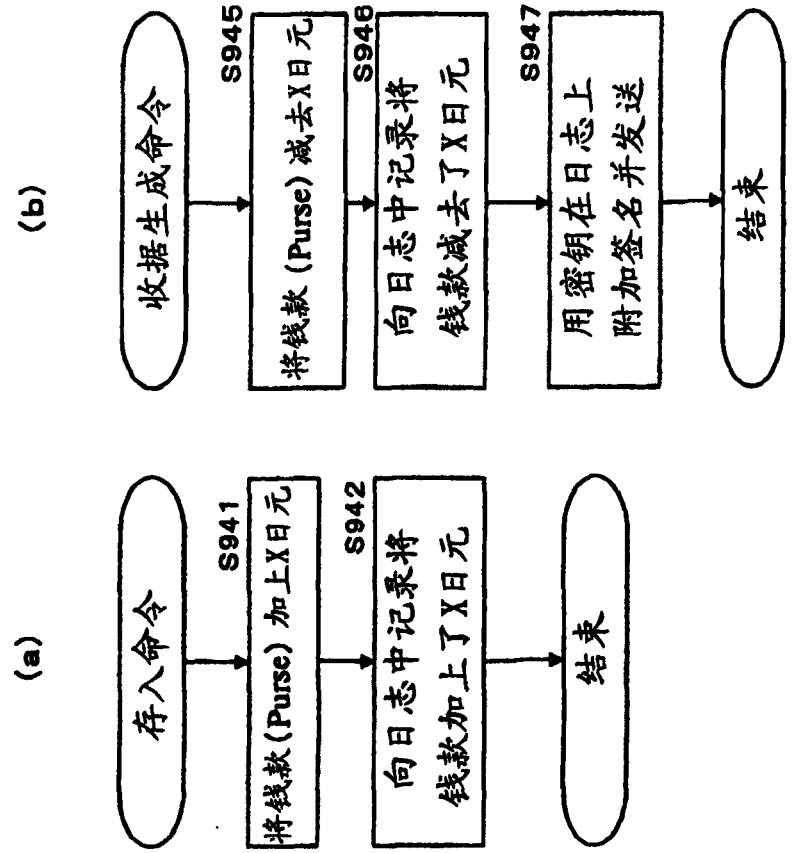
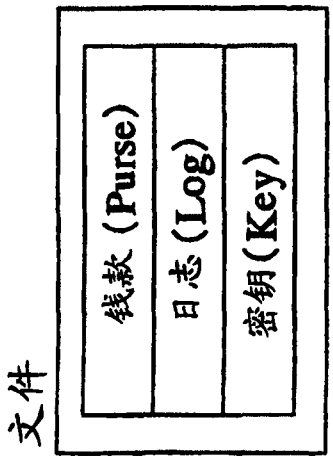

(a)
存入命令
S941
将钱款(Purse)加上X日元
S942
向日志中记录将
钱款加上了X日元
结束
(b)
收据生成命令
S945
将钱款(Purse)减去X日元
S946
向日志中记录将
钱款减去了X日元
S947
用密钥在日志上
附加签名并发送
结束

图 88


文件
钱款(Purse)
日志(Log)
密钥(Key)

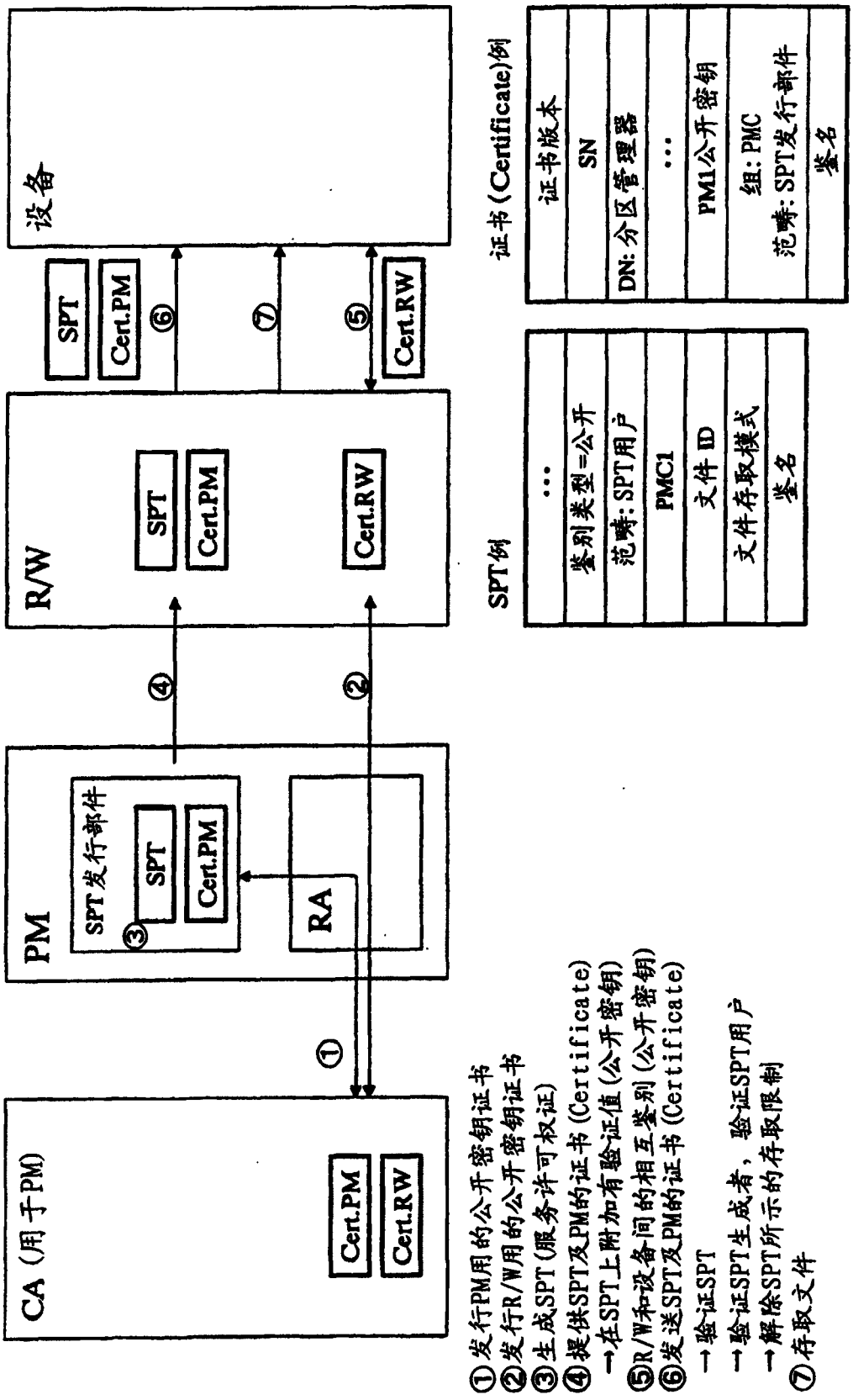

CA (用于PM)
Cert.PM
Cert.RW
PM
SPT发行部件
SPT
Cert.PM
RA
R/W
SPT
Cert.PM
Cert.RW
设备
SPT
Cert.PM
Cert.RW
①
②
③
④
⑤
⑥
⑦
①发行PM用的公开密钥证书
②发行R/W用的公开密钥证书
③生成SPT(服务许可权证)
④提供SPT及PM的证书(Certificate)
→在SPT上附加有验证值(公开密钥)
⑤R/W和设备间的相互鉴别(公开密钥)
⑥发送SPT及PM的证书(Certificate)
→验证SPT
→验证SPT生成者，验证SPT用户
→解除SPT所示的存取限制
⑦存取文件
SPT例
...
鉴别类型=公开
范畴: SPT用户
PMC1
文件 ID
文件存取模式
鉴名
证书(Certificate)例
证书版本
SN
DN: 分区管理器
...
PM1公开密钥
组: PMC
范畴: SPT发行部件
鉴名

图 89

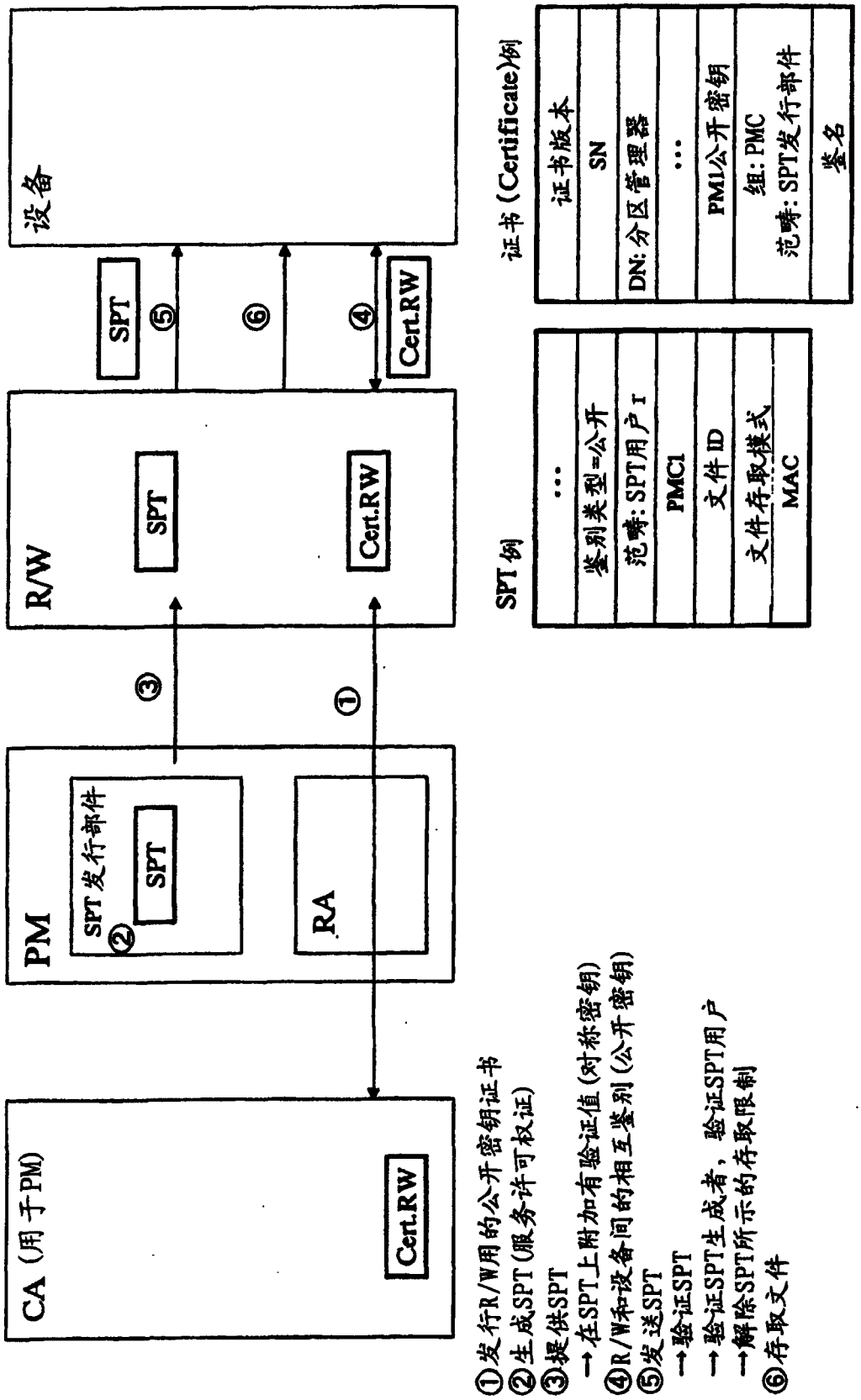

设备
R/W
SPT
Cert.RW
PM
SPT发行部件
SPT
RA
Cert.RW
CA (用于PM)
Cert.RW
①
②
③
④
⑤
⑥
①发行R/W用的公开密钥证书
②生成SPT(服务许可权证)
③提供SPT
→在SPT上附加有验证值(对称密钥)
④R/W和设备间的相互鉴别(公开密钥)
⑤发送SPT
→验证SPT
→验证SPT生成者，验证SPT用户
→解除SPT所示的存取限制
⑥存取文件
SPT 例
...
鉴别类型=公开
范畴: SPT用户ID
PMC1
文件ID
文件存取模式
MAC
证书 (Certificate)例
证书版本
SN
DN: 分区管理器
...
PML公开密钥
组: PMC
范畴: SPT发行部件
鉴名

图 90

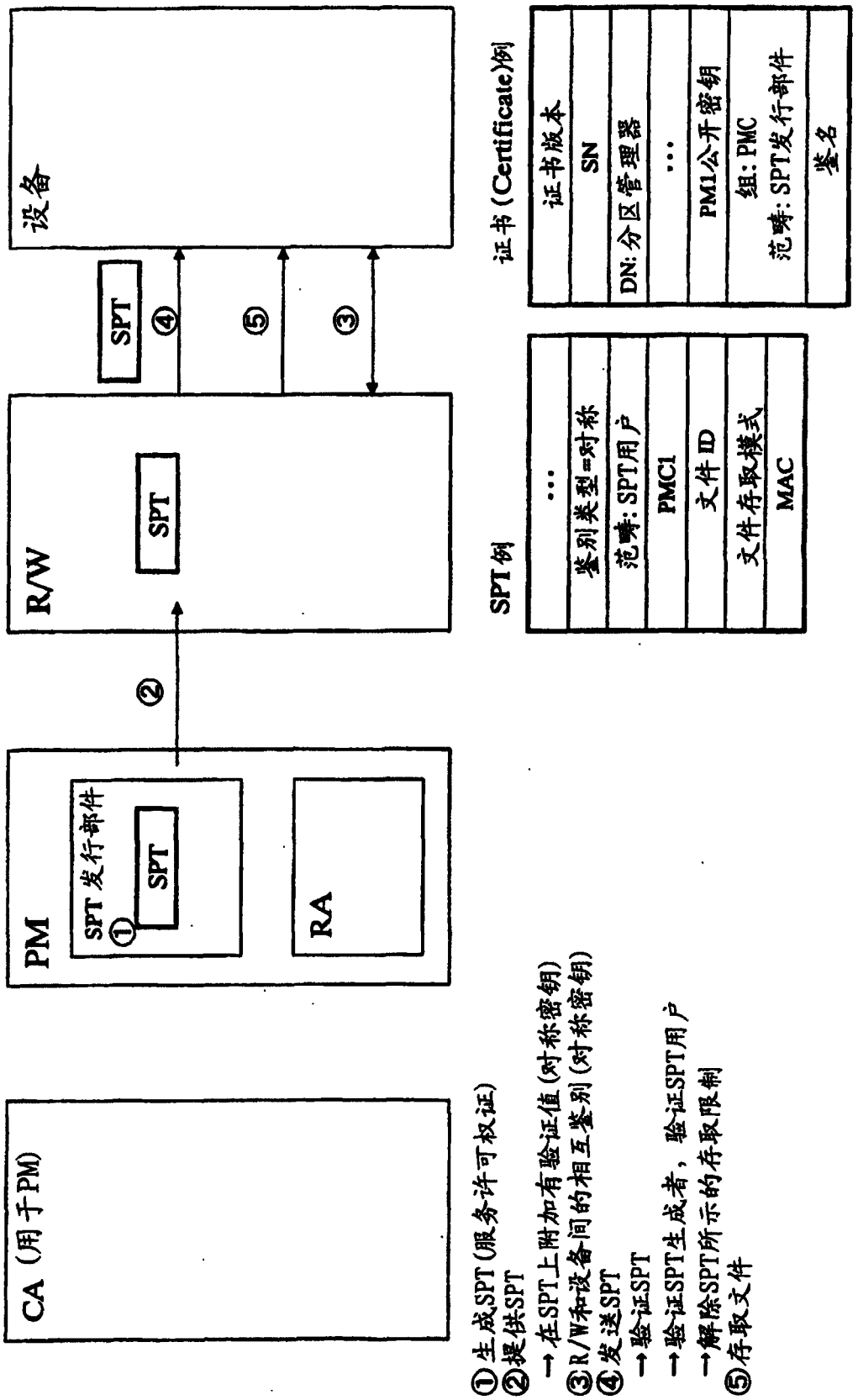

CA (用于PM)
PM
SPT 发行部件
①
SPT
RA
②
R/W
SPT
SPT
④
⑤
③
设备
①生成SPT(服务许可权证)
②提供SPT
→在SPT上附加有验证值(对称密钥)
③R/W和设备间的相互鉴别(对称密钥)
④发送SPT
→验证SPT
→验证SPT生成者，验证SPT用户
→解除SPT所示的存取限制
⑤存取文件
SPT 例
...
鉴别类型=对称
范畴: SPT用户
PMC1
文件 ID
文件存取模式
MAC
证书 (Certificate)例
证书版本
SN
DN: 分区管理器
...
PM1公开密钥
组: PMC
范畴: SPT发行部件
鉴名

图 91

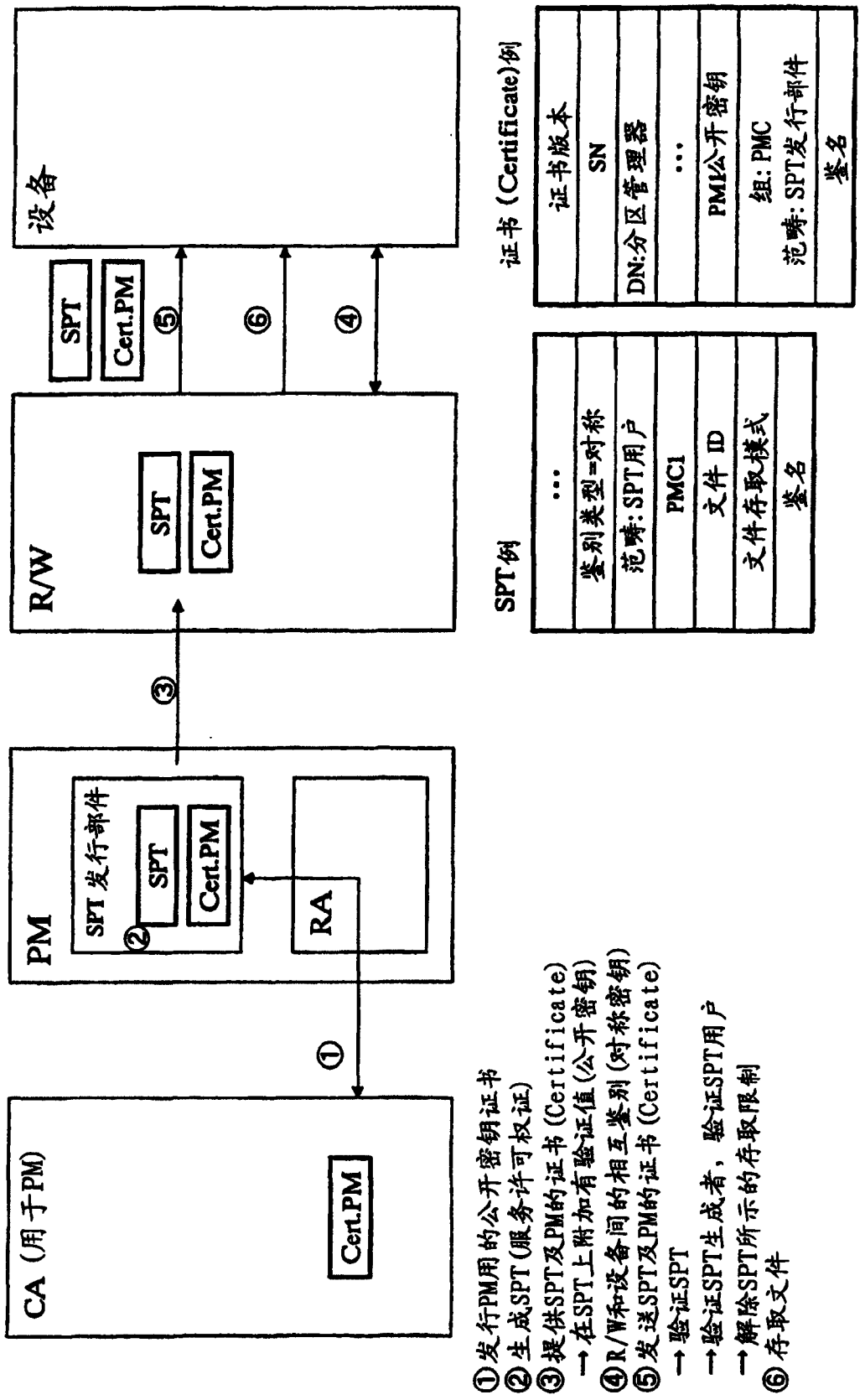

CA (用于PM)
Cert.PM
PM
SPT 发行部件
SPT
Cert.PM
RA
R/W
SPT
Cert.PM
设备
SPT
Cert.PM
①
②
③
④
⑤
⑥
①发行PM用的公开密钥证书
②生成SPT(服务许可权证)
③提供SPT及PM的证书(Certificate)
→在SPT上附加有验证值(公开密钥)
④R/W和设备间的相互鉴别(对称密钥)
⑤发送SPT及PM的证书(Certificate)
→验证SPT
→验证SPT生成者，验证SPT用户
→解除SPT所示的存取限制
⑥存取文件
SPT例
...
鉴别类型=对称
范畴:SPT用户
PMC1
文件 ID
文件存取模式
鉴名
证书(Certificate)例
证书版本
SN
DN:分区管理器
...
PM公开密钥
组: PMC
范畴: SPT发行部件
鉴名

图 92

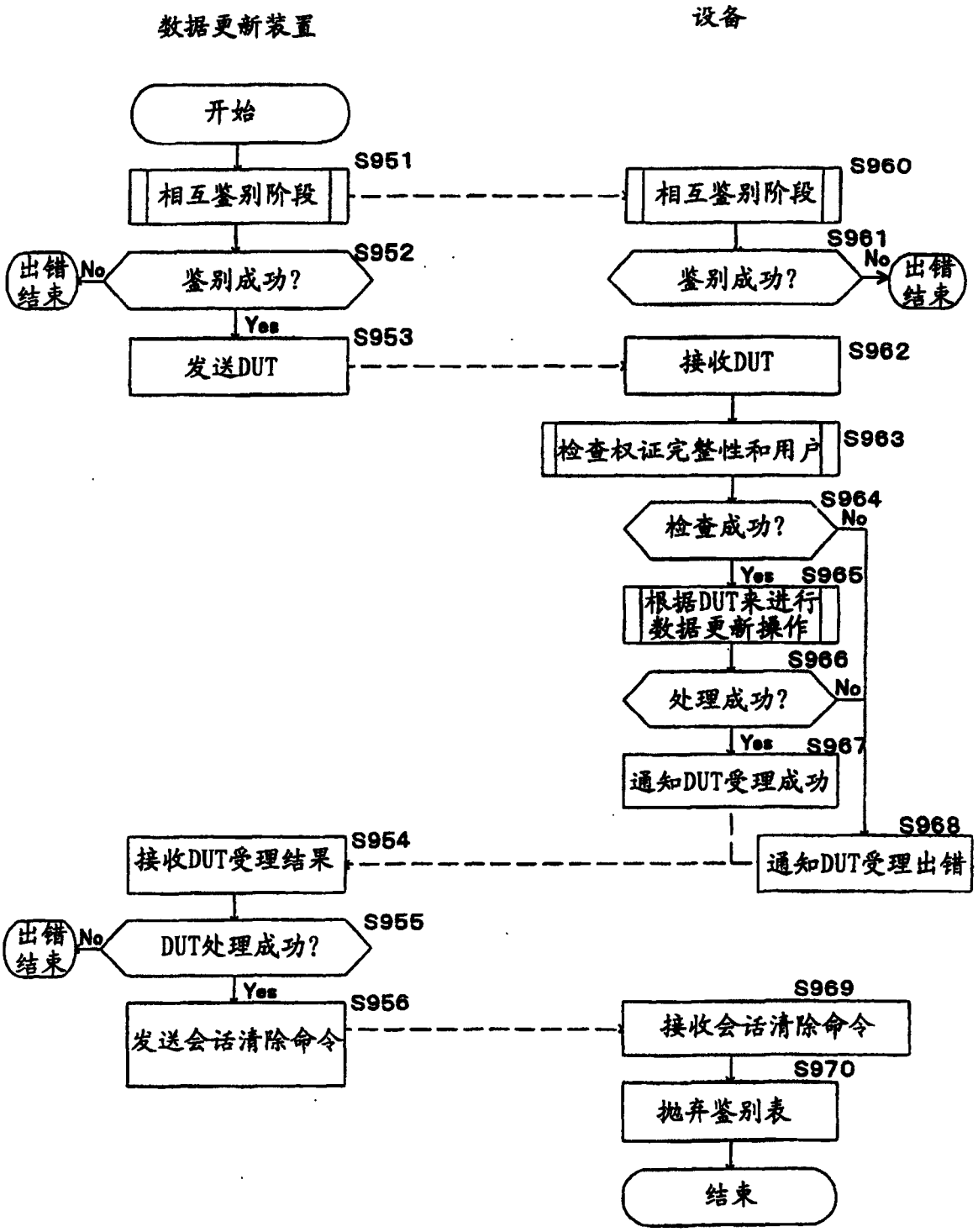

数据更新装置
设备
开始
S951
相互鉴别阶段
S960
相互鉴别阶段
S961
鉴别成功?
No
出错结束
S952
鉴别成功?
No
出错结束
Yes
S953
发送DUT
S962
接收DUT
S963
检查权证完整性和用户
S964
检查成功?
No
Yes
S965
根据DUT来进行数据更新操作
S966
处理成功?
No
Yes
S967
通知DUT受理成功
S968
通知DUT受理出错
S954
接收DUT受理结果
S955
DUT处理成功?
No
出错结束
Yes
S956
发送会话清除命令
S969
接收会话清除命令
S970
抛弃鉴别表
结束

图 93

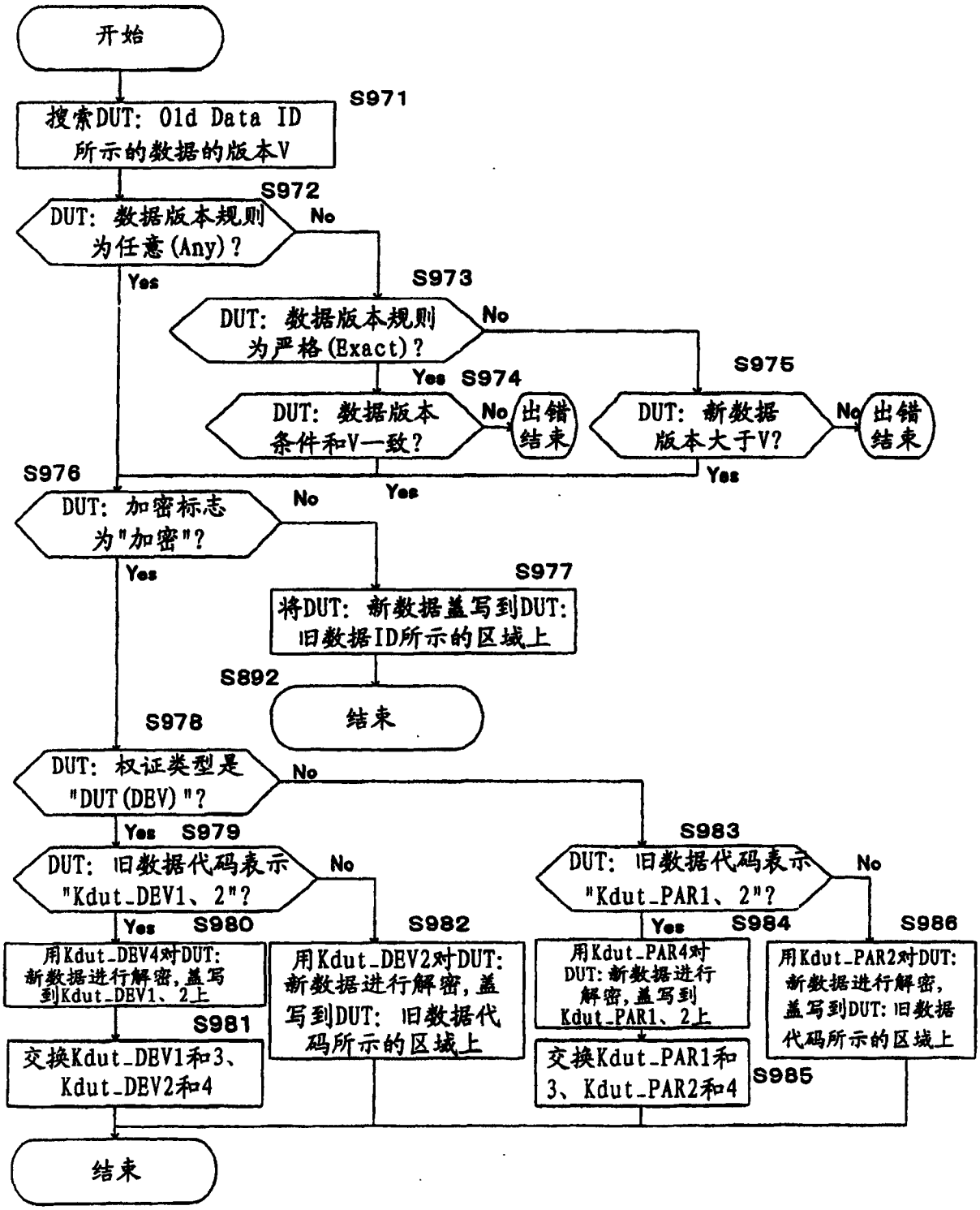
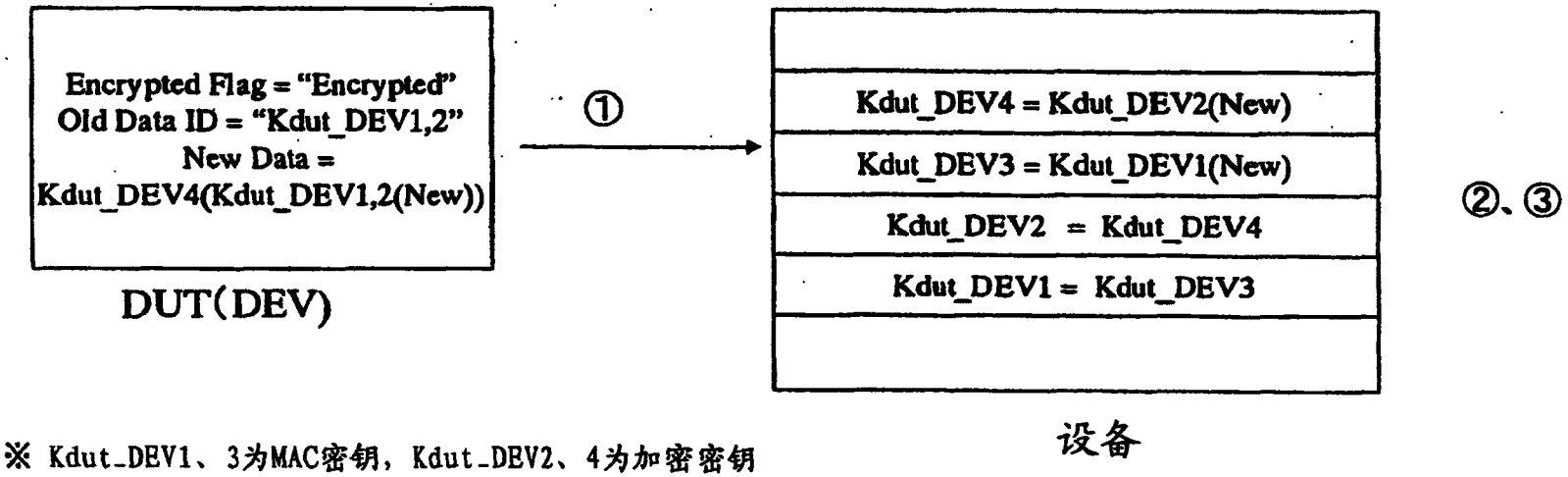

开始
S971
搜索DUT: Old Data ID 所示的数据的版本V
S972
DUT: 数据版本规则为任意(Any)?
No
Yes
S973
DUT: 数据版本规则为严格(Exact)?
No
Yes
S974
DUT: 数据版本条件和V一致?
No
出错结束
S975
DUT: 新数据版本大于V?
No
出错结束
Yes
Yes
S976
DUT: 加密标志为"加密"?
No
Yes
S977
将DUT: 新数据盖写到DUT: 旧数据ID所示的区域上
S892
结束
S978
DUT: 权证类型是"DUT(DEV)"?
No
Yes
S979
DUT: 旧数据代码表示"Kdut_DEV1、2"?
No
Yes
S980
用Kdut_DEV4对DUT: 新数据进行解密,盖写到Kdut_DEV1、2上
S981
交换Kdut_DEV1和3、Kdut_DEV2和4
S982
用Kdut_DEV2对DUT: 新数据进行解密,盖写到DUT: 旧数据代码所示的区域上
S983
DUT: 旧数据代码表示"Kdut_PAR1、2"?
No
Yes
S984
用Kdut_PAR4对DUT: 新数据进行解密,盖写到Kdut_PAR1、2上
S985
交换Kdut_PAR1和3、Kdut_PAR2和4
S986
用Kdut_PAR2对DUT: 新数据进行解密,盖写到DUT: 旧数据代码所示的区域上
结束

图 94

● 更新对象为Kdut_DEV1、2或Kdut_PAR1、2的情况(以Kdut_DEV1、2来说明)

① 用Kdut_DEV4对新的Kdut_DEV1、2(Kdut_DEV1(New)，Kdut_DEV2(New))进行加密.
Kdut_DEV4(Kdut_DEV1(New))，Kdut_DEV4(Kdut_DEV2(New))
记述到DUT中，发送到设备.
② 设备取出Kdut_DEV1(New)、Kdut_DEV2(New)，盖写旧的Kdut_DEV1、Kdut_DEV2
③ 交换Kdut_DEV1、3和Kdut_DEV2、4



※ Kdut_DEV1、3为MAC密钥，Kdut_DEV2、4为加密密钥
成对使用(Kdut_DEV1，Kdut_DEV2)、(Kdut_DEV3，Kdut_DEV4)

图 95

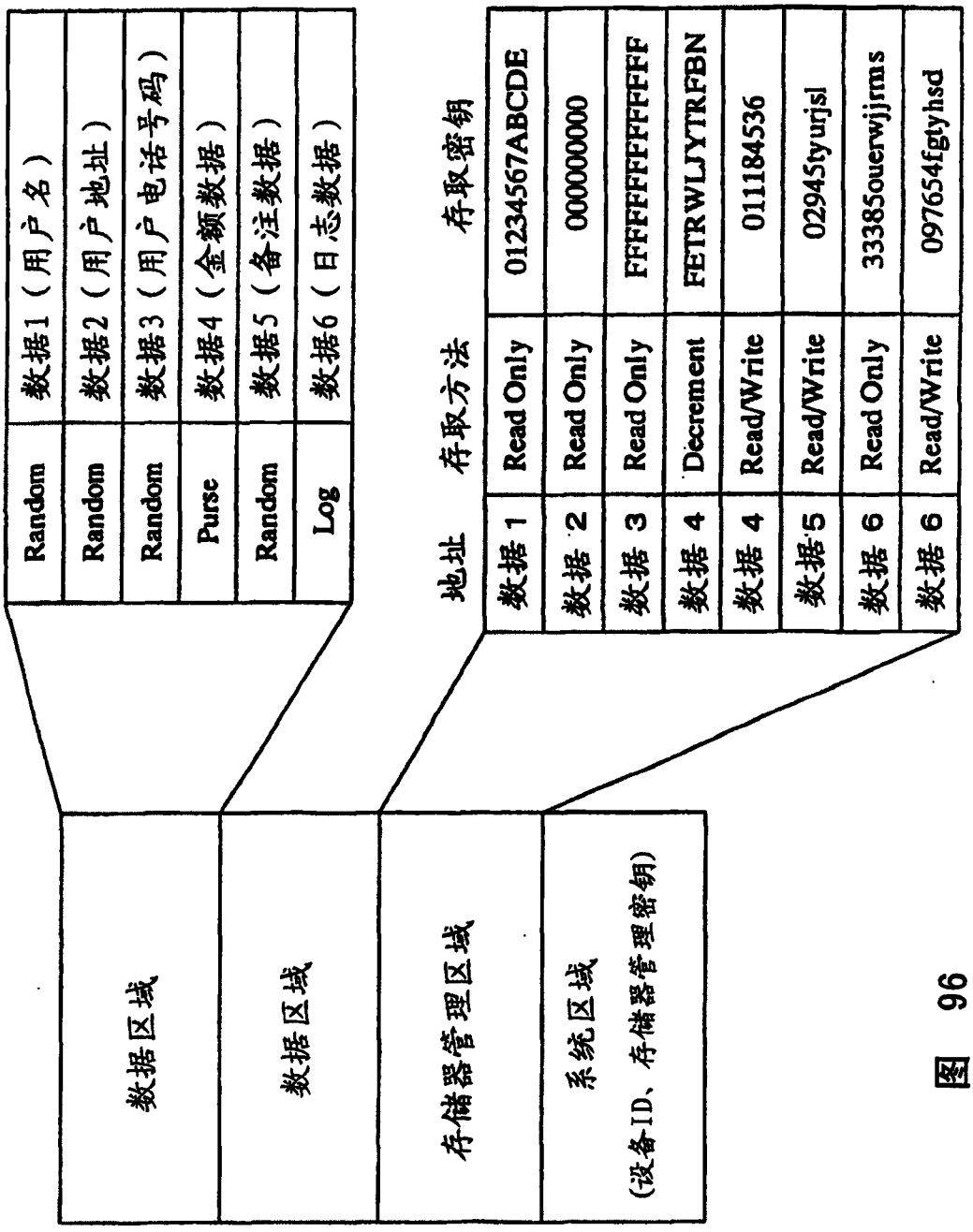

数据区域
数据区域
存储器管理区域
系统区域
(设备ID、存储器管理密钥)
Random 数据1（用户名）
Random 数据2（用户地址）
Random 数据3（用户电话号码）
Purse 数据4（金额数据）
Random 数据5（备注数据）
Log 数据6（日志数据）
地址 存取方法 存取密钥
数据 1 Read Only 01234567ABCDE
数据 2 Read Only 00000000
数据 3 Read Only FFFFFFFFFFFF
数据 4 Decrement FETRWLJYTRFBN
数据 4 Read/Write 011184536
数据 5 Read/Write 02945tyurjsl
数据 6 Read Only 33385ouerwjjrms
数据 6 Read/Write 097654fgtyhsd

图 96

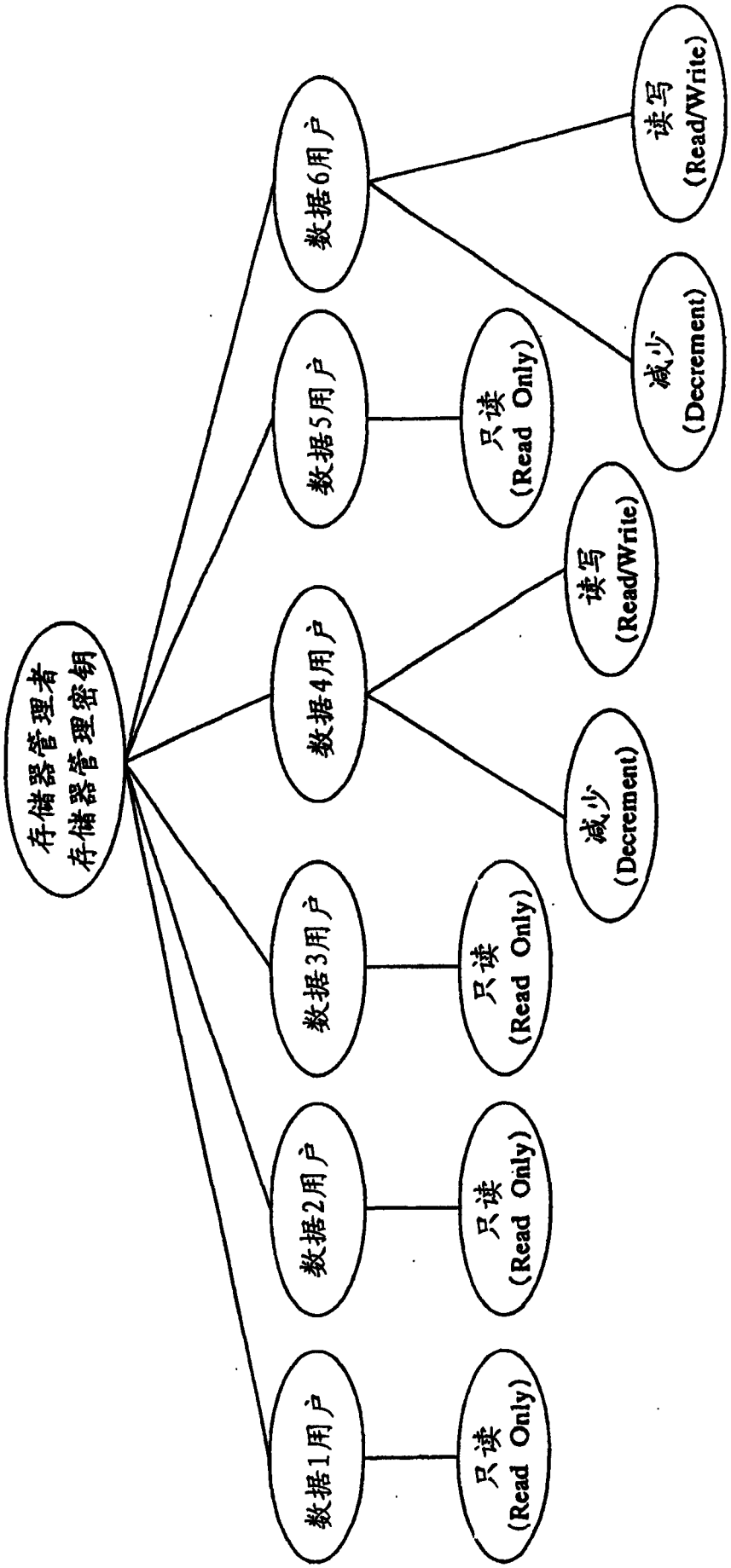

存储器管理者
存储器管理密钥
数据1用户
数据2用户
数据3用户
数据4用户
数据5用户
数据6用户
只读
(Read Only)
只读
(Read Only)
只读
(Read Only)
读写
(Read/Write)
减少
(Decrement)
读写
(Read/Write)
减少
(Decrement)

图 97

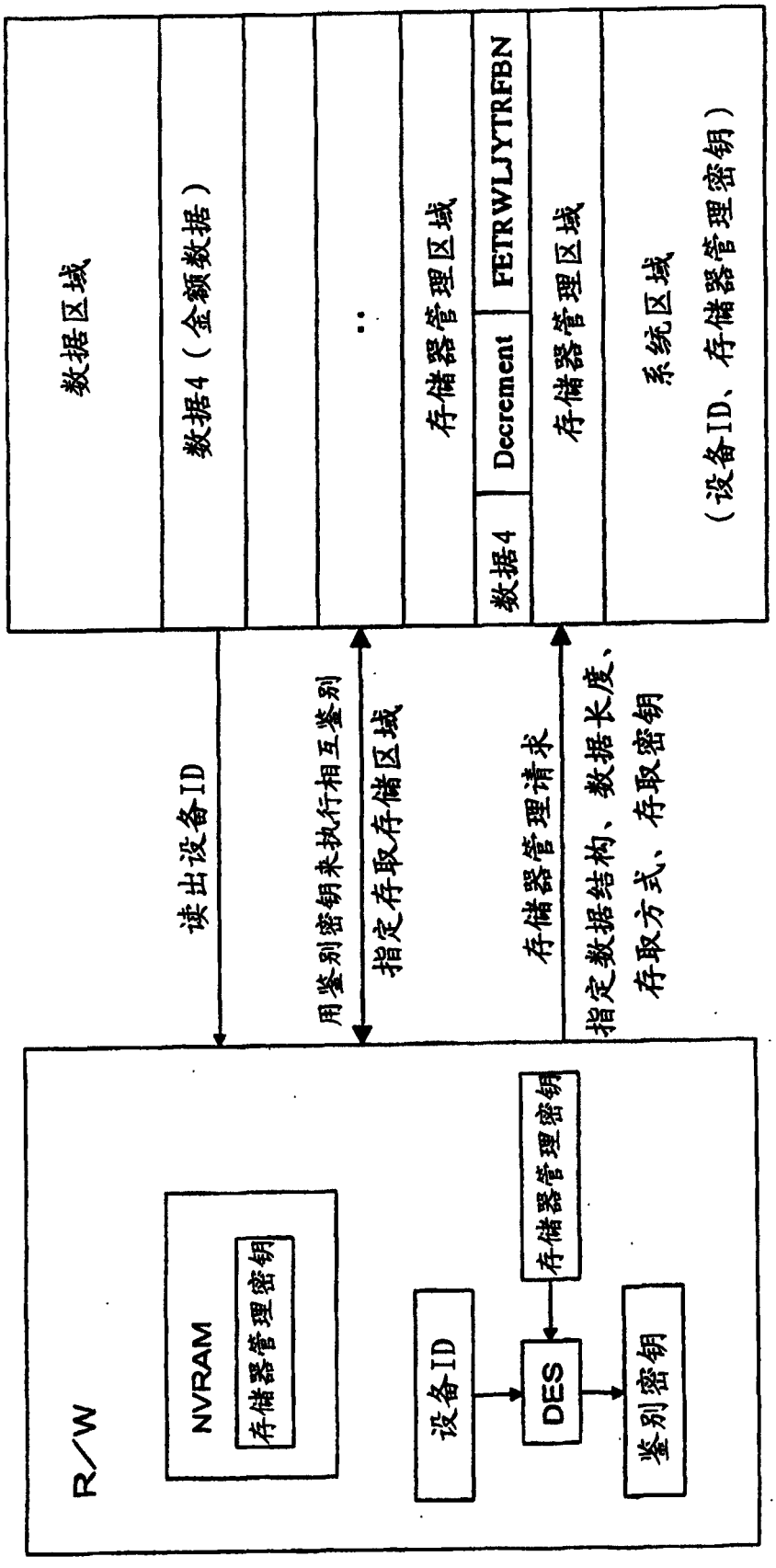

R/W
NVRAM
存储器管理密钥
设备ID
DES
存储器管理密钥
鉴别密钥
读出设备ID
用鉴别密钥来执行相互鉴别
指定存取存储区域
存储器管理请求
指定数据结构、数据长度、
存取方式、存取密钥
用会话密钥进行加密，或者
附加MAC值来防止窜改、泄漏。
数据区域
数据4（金额数据）
存储器管理区域
数据4
Decrement
FETRWLJYTRFBN
存储器管理区域
系统区域
（设备ID、存储器管理密钥）

图 98

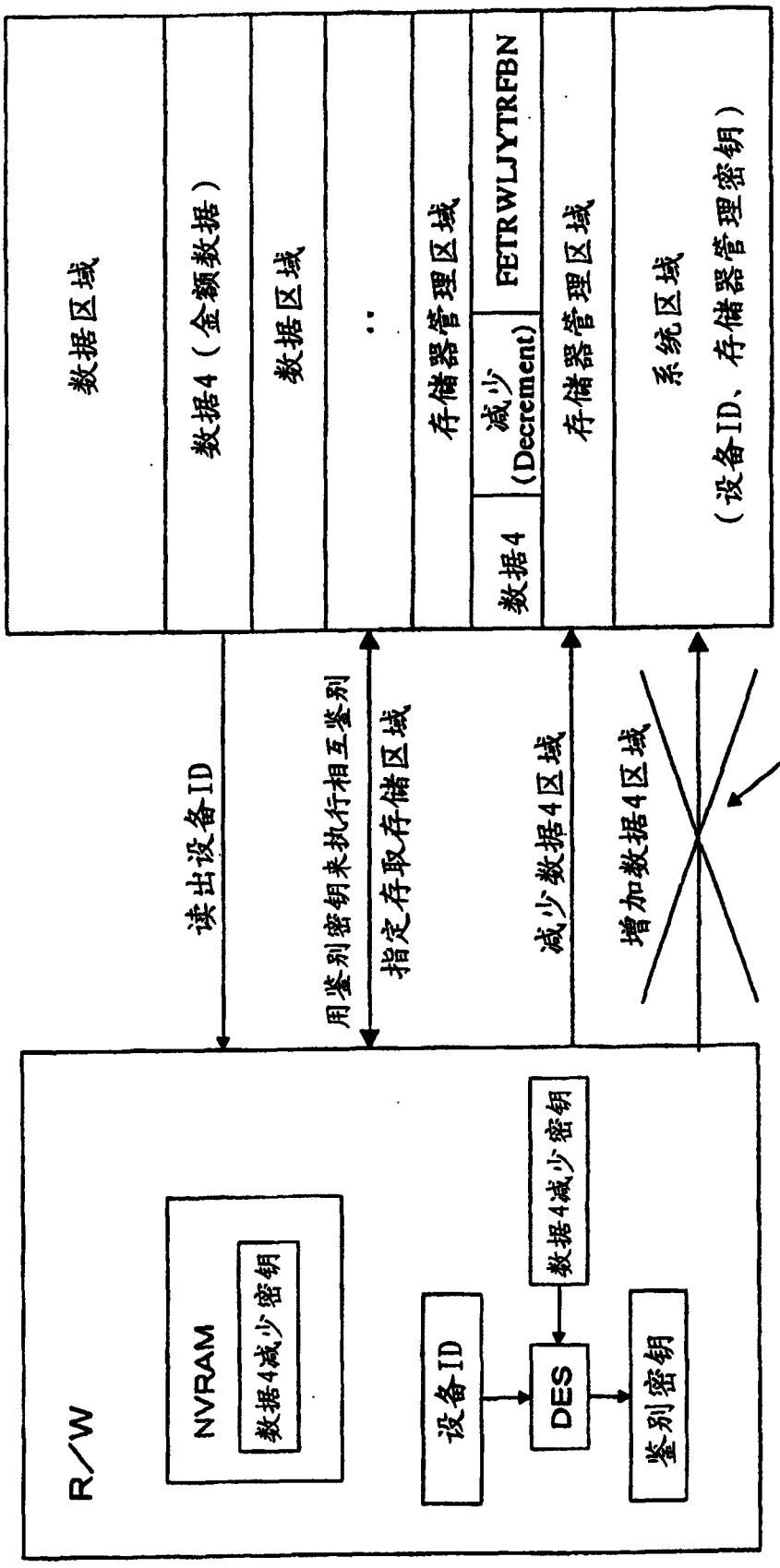

数据区域
数据4（金额数据）
数据区域
..
存储器管理区域
数据4
减少(Decrement)
FETRWLJYTRFBN
存储器管理区域
系统区域
（设备ID、存储器管理密钥）
读出设备ID
用鉴别密钥来执行相互鉴别
指定存取存储区域
减少数据4区域
增加数据4区域
能够通过相互鉴别来进行存取控制
R/W
NVRAM
数据4减少密钥
设备ID
DES
数据4减少密钥
鉴别密钥
即使R/W中的密钥泄漏，也只能进行规定的存取。（这里只能进行减少）

图 99