[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
G06K 19/07

[12] 发明专利申请公开说明书

[21] 申请号 03137974.5

[43] 公开日 2004 年 4 月 7 日

[11] 公开号 CN 1487468A

[22] 申请日 2003.6.10 [21] 申请号 03137974.5

[30] 优先权

[32] 2002. 6.10 [33] JP [31] 2002 - 169315

[71] 申请人 坂村健

地址 日本东京都

共同申请人 越塚登 株式会社 NTT 都科摩

[72] 发明人 坂村健 越塚登 石井一彦 森謙作 青野博 本郷節之

[74] 专利代理机构 北京银龙专利代理有限公司

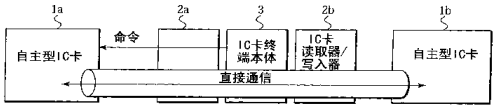
代理人 吴邦基

权利要求书 1 页 说明书 106 页 附图 7 页

[54] 发明名称 自主型集成电路卡

[57] 摘要

本发明涉及一种自主型集成电路卡，除通过物理层连接到集成卡读/写器的主机设备接口外还包括直接同通过网络连接到一集成电路卡终端本体的通信设备联系的一逻辑外部通信接口。一通信控制单元包含通过该外部通信接口直接同该通信设备通信的一软件模块。一个中央处理单元通过该通信控制单元进行认证并读取存储在非易失存储器中的值信息。另外，中央处理单元通过利用加密处理单元给该读值信息加密，并通过通信控制单元和该外部通信接口直接发送该加密的值信息给通信设备。



ISSN1008-4274

知识产权出版社出版

1．一种自主型集成电路卡包括：

一逻辑主机接口，该逻辑主体接口通过物理层与集成电路卡终端相连；

一逻辑外部通信接口，以及该逻辑外部通信接口与通过集成电路卡终端而物理的连接在其上的通信设备进行通信；以及

一集成电路芯片，该芯片用于识别通信设备的连接并且通过外部通信接口与通信设备直接进行通信。

2．根据权利要求 1 的自主型集成电路卡，

其中自主型集成电路芯片包括一个即就是一软件模块的通信控制单元，该通信控制单元通过逻辑外部通信接口可与通信设备直接进行通信。

3．根据权利要求 2 的自主型集成电路卡，

其中通信控制单元包括：

一个可为通信设备建立会话通信路经的会话管理命令组；以及

一个可为通信设备建立事务会话的事务管理命令组。

4．根据权利要求 1 的自主型集成电路卡，

其中集成电路芯片包括一加密处理单元，该单元执行与通信设备的相互认证处理，并且对与和通信设备进行通信有关的信息进行加密和解密。

5．根据权利要求 4 的自主型集成电路卡，

集成电路芯片包括一个分配的唯一标识符，且识别通信设备并根据该标识符来执行相互认证处理。

6．根据权利要求 4 的自主型集成电路卡，

其中加密处理单元根据所识别的通信设备的类型而在多个认证处理和多个加密处理当中分别选择适当的认证处理和适当的加密处理以执行这些处理。

7．根据权利要求 1 的自主型集成电路卡，

其中集成电路芯片包括一个用于存储值信息的存储器，并且该芯片与通信设备进行与值信息有关的通信。

自主型集成电路卡

技术领域

本发明涉及集成电路（IC）卡，更具体地说，涉及一种可直接地且可自主地与直接或通过网络而与IC卡终端相连的其他设备进行通信的自主型IC卡。

背景技术

图1给出了在IC卡和IC卡终端之间进行信号交换的传统系统示意图。

这里，IC卡终端包括IC卡读取器/写入器200以及IC卡终端本体300。尽管这里所示意性说明的IC卡100和IC卡读取器/写入器200是独立的，但是IC卡100和IC卡读取器/写入器200也可是那里所描述的无触点型，或者，IC卡100和IC卡读取器/写入器200可以是IC卡100典型的插入到IC卡读取器/写入器200中的有触点型。

当利用上述所描述的IC卡100来进行处理时，IC卡终端本体300用作主设备，IC卡100用作从设备。换句话说，当操作者意欲通过利用IC卡100来执行处理时，操作者首先将一指令输入到IC卡终端本体300的输入端（未给出）。为响应该指令，作为主设备的IC卡终端主体300通过IC卡读取器/写入器200将一命令传送到IC卡100。一旦接收到该命令，作为从设备的IC卡100执行与该命令相对应的处理，并且通过IC卡读取器/写入器200而将执行结果返回到IC卡终端主体300。

图2给出了在两个IC卡彼此进行通信的情况下传统的系统进行信号交换的示意图。

在附图中，除了如图1所示的结构外，IC卡读取器/写入器200b还与IC卡终端主体300相连。这里，IC卡100b以无触点方式或者有触点方式与IC卡读取器/写入器200b相连。

与如图1的情况相似，操作者首先将与IC卡间的通信有关的一指令输入到IC卡终端本体300的输入端（未给出）。为响应该指令，IC卡终

端主体 300 通过 IC 卡读取器/写入器 200a 将一命令传送到 IC 卡 100a。一旦接收到该命令，IC 卡 100a 执行与该命令相对应的处理，并且输出作为执行结果的响应 A。通过 IC 卡读取器/写入器 200a 将 IC 卡 100a 所输出的响应 A 输入到 IC 卡终端主体 300。就这种情况下的响应 A 而言，IC 卡终端主体 300 执行信息的中继操作。通过 IC 卡读取器/写入器 200b 将 IC 卡终端主体 300 所输出的响应 A 输入到 IC 卡 100b。IC 卡 100b 执行与所输入的响应 A 相对应的处理，并且输出作为执行结果的响应 B。通过 IC 卡读取器/写入器 200b 将 IC 卡 100b 所输出的响应 B 输入到 IC 卡终端主体 300。在这种情况下，IC 卡终端主体 300 执行与响应 B 有关的信息的中继操作。通过 IC 卡读取器/写入器 200a 将 IC 卡终端主体 300 所输出的响应 B 输入到 IC 卡 100a。

当在 IC 卡 100a 与 IC 卡 100b 间不断进行上述交换时，该操作基本上相同。换句话说，IC 卡终端主体 300 总是起这样的作用，即对 IC 卡 100a 与 IC 卡 100b 间所交换的信息执行中继操作。

顺便说一下，在 IC 卡间的传统通信过程中，IC 卡终端主体总是起这样的作用，即对如上所述的 IC 卡间的通信有关的信息执行中继操作。因此，传统地，由 IC 卡终端主体所表示的中间设备可篡改或窃取在 IC 卡间进行通信的信息。因此传统地存在这样的问题，即不能确保在 IC 卡间进行准确的信息通信。

发明内容

根据先前的问题而提出了本发明。本发明的一个目的就是提供了一种自主型 IC 卡，当该卡与直接或通过网络而与 IC 卡终端相连的其他通信设备进行通信时，通过将该卡直接与诸如 IC 卡终端这样的其他通信设备进行通信而确保在 IC 卡与其他的通信设备间进行准确信息的安全通信。

为了获得先前的目的，提出了一种自主型 IC 卡，该卡包括一逻辑主机接口、一逻辑外部通信接口、以及一 IC 芯片。该逻辑主机接口通过物理层与 IC 卡终端相连。该逻辑外部通信接口与通过物理的 IC 卡终端而连接在其上的通信设备进行通信。该 IC 芯片用于识别通信设备的连接并

且通过外部通信接口与通信设备直接进行通信。

根据本发明，自主型 IC 卡可自主地识别通过 IC 卡终端而连接在其上的通信设备，并且与通信设备直接进行通信。

在本发明的优选实施例中，IC 芯片包括即就是软件模块的一通信控制单元，该通信控制单元通过逻辑外部通信接口可与通信设备直接进行通信。

根据本发明，通信控制单元可利用软件模块来进行处理，该软件模块未意识到相连的 IC 卡终端。

在本发明的优选实施例中，通信控制单元包括一个可为通信设备建立会话通道路经的会话管理命令组以及一个可为通信设备建立事务会话的事务管理命令组。

在本发明的优选实施例中，IC 芯片包括一加密处理单元，该单元执行与通信设备的相互认证处理，并且对与和通信设备进行通信有关的信息进行加密和解密。

根据本发明，加密处理单元进行该通讯设备互相的认证处理过程，并且对有关通讯设备的通讯的信息进行加密和解密。

在本发明的优选实施例中，IC 芯片包括一个分配的唯一标识符。这里，IC 芯片识别通信设备并根据该标识符来执行相互认证处理。

在本发明的优选实施例中，加密处理单元根据所识别的通信设备的类型而在多个认证处理和多个加密处理当中分别选择适当的认证处理和适当的加密处理以执行这些处理。

在本发明的优选实施例中，IC 芯片包括一个用于存储值信息的存储器。这里，IC 芯片与通信设备进行与值信息有关的通信。

当结合随后附图来阅读本发明时，根据下述详细说明，本发明的特性、原理以及实用性则变得显而易见。

附图说明

图 1 给出了在 IC 卡和 IC 卡终端之间进行信号交换的传统系统示意图；

图 2 给出了在两个 IC 卡彼此进行通信的情况下传统的系统进行信号

交换的示意图；

图 3 给出了根据本发明的一个 IC 卡实施例的逻辑结构示意图；

图 4 给出了利用本发明的自主型 IC 卡来执行 IC 卡间的通信的系统结构示意图；

图 5 给出了利用本发明的自主型 IC 卡来执行通信的另一个系统结构示意图；

图 6 给出了通过公钥加密来进行发行者模式中的认证方法的示意图；

图 7 给出了依据私钥加密在主属模式认证的方法的示意图。

具体实施方式

现在，参考随后的附图对根据本发明的自主型 IC 卡实施例进行详细的说明。

图 3 给出了根据本发明的一个 IC 卡实施例的逻辑结构示意图。

在图 3 中，本发明的自主型 IC 卡包括一中央处理单元（CPU）10、一只读存储器（ROM）11、一随机访问存储器（RAM）12、一非易失存储器 13、一加密处理单元 14、一通信控制单元 15、一外部通信接口 16、以及一主机设备接口 17。

这里，上述结构是物理结构（硬件）和逻辑结构（软件）的混合体。下面将对这两个结构分别进行详细的说明。

主机设备接口 17 是传统的用于执行外部通信的物理接口。与现有技术相类似，主机设备接口 17 包括有触点型和无触点型。CPU10 是可执行除浮点操作之外的所有类型的操作这样的中央处理单元。值得注意的是，专门提供了一个浮点处理单元（未给出）以执行某些浮点操作。ROM11 是一只读存储器，只有随后所描述的本发明的自主型 IC 卡 1 和芯片 ID 才有的软件（例如操作系统（OS）或程序）存储在该 ROM11 中。RAM12 是一存储访问存储器，由在自主型 IC 卡内进行操作的软件所处理的数据被临时存储在该 RAM12 中。当电源停止向 IC 卡供电时，RAM12 中的内容被彻底删除。

非易失存储器 13 用于存储随后所描述的值信息。值信息即就是包含有诸如电子门票和电子货币这样的值的数据。值得注意的是，因为非易失存储器 13 是用于保存内容的存储器，因此即使电源停止了，其上的内容也可被保存。

以下是物理结构组件。

外部通信接口 16 是允许随后所描述的本发明的芯片可执行与外部的直接通信这样的逻辑接口，上述芯片包括 CPU10 等等。传统的 IC 卡只包括一逻辑接口以只与和其物理相连的设备进行通信，一般的，只与 IC 终端进行通信。然而，本发明的自主型 IC 卡 1 除了包括逻辑主机接口之外还包括外部通信接口 16，该外部通信接口用于执行与外部设备（具有体现在这里的本发明的芯片的服务客户机，这将在后面进行描述）的直接通信。值得注意的是，该外部通信接口是根据在随后详细描述的“本发明的 4.API 协议”中的一协议来进行通信的一接口。

通信控制单元 15 是一软件模块，该通信控制单元控制通过外部通信接口 16 而与外部设备的通信。通信控制单元 15 是可完全实现随后所详细描述的“本发明的 10.API 协议”的软件模块。该软件模块尤其与“10.1 会话管理命令组”以及“10.2 事务管理命令组”最相关。利用该模块可执行与外部设备的直接通信。

加密处理单元 14 是根据随后所描述的“3.4 认证、访问控制、以及加密”、“5.eTP 密钥证书”、以及“6.密钥实体”所定义的各种类型的相互认证和加密通信。加密处理单元 14 在多个可选的相互认证和多个可选的加密通信中分别选择适当的相互认证和适当的加密通信并对其进行切换。这里，通过利用加密处理单元 14 来执行相互认证和加密处理可进一步减小通过诸如 IC 卡终端这样的中间设备可窃取数据内容或篡改数据的风险。

值得注意的是，根据本发明的如图 3 所示结构单元最好是利用单片微型计算机（在下文中被称为“IC 芯片”）来实现的，其详细说明将列在实施例的说明的最后一部分。

现在，将对与包含在本发明自主型 IC 卡中的 IC 芯片有关的全部特性进行详细的说明。

根据本发明的 IC 芯片作为分布式环境中的一节点。因此，如果一设备可通过一网络而与诸如其他的 IC 卡进行物理通信，其中上述设备上的本发明 IC 卡以有触点方式或无触点方式与包括有 IC 卡读取器/写入器的

IC 卡终端主体相连，那么分别包括有根据本发明的 IC 芯片的这些 IC 卡均作为分布式环境中的一节点。此外，在这种情况下，IC 卡读取器/写入器和 IC 卡终端主体起网关（电桥）的作用，以使该网络的无触点通讯物理层上可传递。换句话说，上述外部通信接口 16 控制 IC 卡通信以作为分布式系统中的一节点，同时主机设备接口 17 起这样的作用，即作为无触点通信的可传递物理层。

因此，为了起分布式环境中的一节点，根据本发明的 IC 芯片具有整个分布式系统中的唯一标识符（ID），作为最简单的例子，该分布式系统由上述网络来表示。

此外，包含在本发明自主型 IC 卡中的 IC 芯片识别多个设备并且首先执行与这些设备的相互认证，这些设备通过网络相连并分别提供了相似的 IC 芯片。在确认这些设备是具有资格的对象之后，IC 卡于是执行与这些设备的通信。当进行认证时存在两种模式，即，（信息）发行者模式和所用者模式。在后面将对其进行详细的说明。

另外，根据本发明的 IC 卡可采用基于上述 ID 的存储控制列表以保护由 IC 芯片所拥有的资源。也就是说，将访问控制列表添加到每一个芯片和文件（以及记录）上。

根据本发明的 IC 芯片均等的处理“所用者”、“发行者”以及“其他的”服务客户机。IC 卡发布了一应用编程接口（API）以根据每个服务客户机所拥有的许可来改变访问控制列表。按照这种方式，IC 芯片可灵活的执行对诸如访问权的限制、免除、或传输这样的控制。

同时，根据本发明的 IC 芯片支持一混合方式以保留内容同时可将内容分布到多个信息容器中，例如，将内容分布到本发明的芯片与网络上的值信息存储服务器之间。根据本发明的 IC 芯片进一步提供了内容间的链接功能以作为用于支持混合模式的机制。

接下来，对有关利用本发明的自主型 IC 卡来进行通信的具体例子进行详细的描述。

图 4 给出了利用本发明的自主型 IC 卡来执行 IC 卡间的通信的系统结构示意图。

在该图中，自主型 IC 卡 1a 以有触点的方式或无触点的方式与位于 IC 卡终端主体 3 上的 IC 卡读取器/写入器 2a 相连。同时，自主型 IC 卡 1b 以有触点的方式或无触点的方式与位于 IC 卡终端主体 3 上的 IC 卡读取器/写入器 2b 相连。

在如上所述的结构中，当操作者想要使自主型 IC 卡 1a 与自主型 IC 卡 1b 进行通信时，操作者将一指令输入到位于 IC 卡终端主体 3 上的输入端（未给出）。为了响应该指令，IC 卡终端主体 3 通过 IC 卡读取器/写入器 2a 将一命令传送到自主型 IC 卡 1a。一旦接收到该命令，自主型 IC 卡 1a 以软件的方式通过如图 3 所述的外部通信接口与自主型 IC 卡 1a 进行直接且自主的通信。自主型 IC 卡 1a 可与自主型 IC 卡 1b 进行直接通信的原因就是自主型 IC 卡 1a 可起如上所述的分布式环境中一节点的作用。当这样进行通信时，如果如图 3 所示的加密处理单元 14 对通信数据进行加密和解密，则可必定防止在通信的过程中数据被篡改或窃取的风险。

图 5 给出了利用本发明的自主型 IC 卡来执行通信的另一个系统结构示意图。

当利用本发明的自主型 IC 卡进行通信时，与其进行通信的另外一方不必是 IC 卡，但是可以是其他的通信设备。在附图中，IC 卡终端主体 3 通过网络 4 与通信设备 5 相连。当自主型 IC 卡 1 通过网络 4 与通信设备 5 进行通信时，其接收到来自 IC 卡终端主体 3 的命令的自主型 IC 卡 1 以软件的方式与通信设备 5 进行直接且自主的通信。

现在，对有关 IC 芯片（在下文中被简单的称为“本芯片”）的详细规范进行描述，该芯片包含在本发明的自主型 IC 卡中。在下述说明中，信息容器（CHs）对应于上述自主型 IC 卡 1 和 1a，服务客户机（SCs）对应于与自主型 IC 卡 1 和 1a 进行通信的所有设备，即就是，对应于上述的 IC 卡终端主体 3、自主型 IC 卡 1b、以及通信设备 5。

这里，为了便于理解概述，在下面给出了表的内容，并且在下述说明中给出了 IC 芯片各个配置的索引。

表的内容

1．介绍

1.1 作为分布式环境中一节点的本芯片

1.2 本芯片 ID 所指定的相互认证方法

1.3 根据基于本芯片 ID 的访问控制列表方法的资源保护机制

1.4 实现芯片所用者、值信息发行者、以及值信息用户易理解的访问控制

1.5 具有重新运行性能的事务机制

1.6 具有链接功能的存储结构

1.7 相关芯片系统的相容性

1.8 与各种芯片相对应的适度标准化

2．系统规范说明

2.1 系统结构

2.2 本芯片的标识符（本芯片 ID）

3．本芯片的概述

3.1 本芯片

3.2 数据结构模型

3.3 本芯片 API

3.4 认证、访问控制以及加密

4．本芯片 API 协议

4.1 包类型

4.2 命令标识符列表（命令 ID）

4.3 错误代码列表

4.4 MAC，随附信息组

4.5 会话通信以及非会话通信

5．eTP 密钥证书

当在会话建立的期间进行认证时，要利用该证书

1.概述

2.详细说明

6．密钥实体

7．本发明的标准内容格式

8．利用密钥实体对本发明的操作标准内容进行解决处理

9．本芯片 API 的规范说明（数据类型定义）

10．本芯片 API 的规范说明（命令定义）

10.1 会话管理命令组

10.2 事务管理命令组

10.3 文件管理命令组

10.4 记录管理命令组

10.5 密钥实体管理命令组

10.6 认证辅助管理命令组

实现加密的详细说明

1．介绍

本芯片（添加有值信息的芯片）是一计算机系统（例如 IC 卡），该系统包括用于高功能分布式系统中的值信息的存储介质以及计算机系统的外部说明，其中与本发明有关的一方案针对的是上述分布式系统。假定将来技术进步了，本芯片可投入到一系列的 8 位 CPU 型、16 位 CPU 型、以及 32 位 CPU 型，并且本芯片可提供与通用操作有关的一通用命令和一通用消息格式。在本芯片的规范说明的各种系统当中，在采用 16 位 CPU 型的 IC 芯片作为目标硬件的情况下，该描述特别解释了该规范说明的概要。

在这章中，将对有关本芯片的特性进行详细的说明以与其他现有的抗篡改型芯片相比较。

1.1 作为分布式环境中一节点的本芯片

不像传统的 IC 芯片那样作为计算机外围设备单元以通过读取器/写入器来进行操作，本芯片作为分布式环境中的一节点。位于一网络和该芯片中的服务供应商模块、以及该芯片和一卡，在一个同等的基础上执行对等的通信。读取器/写入器设备构成了一网关（桥）以与本地网络（LAN）的可传递物理层进行无触点通信。

在本芯片的结构中，本芯片具有一标识符，该标识符是整个分布式

系统中唯一的。本芯片 ID 不仅被用于物理的识别芯片，而且还可控制分布式环境中的路径。这里，当进行许可的通信时，利用本芯片 ID 作为一标识符以与另一方进行通信。

因此，许可本芯片的目标不是读取器/写入器，而是网络中的运算实体（信息容器），该运算实体通过网络和读取器/写入器与芯片进行信息交换。

1.2 本芯片 ID 所指定的相互认证方法

在执行相互认证并因此确认与其进行通信的另一方是具有资格的对象之后本芯片执行通信。在本芯片结构中，包含有本芯片的信息容器（CHs）以及包含有本芯片的服务客户机（SCs）均包括兼容唯一标识符（本芯片 ID），并在相互认证之后确定与另一方进行通信的本芯片 ID。

1.3 根据基于本芯片 ID 的访问控制列表方法的资源保护机制

本芯片指定了一 SC，该 SC 通过相互认证而发布了本芯片的 API。因此，基于本芯片 ID 的访问控制列表被用于保护由本芯片所拥有的资源。本芯片此刻根据资源通过利用本芯片 ID 而表明了诸如“发行者”、“所用者”等等之类的属性。此外，可通过利用存储控制列表来指定由“发行者”、“所用者”以及“其他者”所发出的命令。

1.4 实现芯片所用者、值信息发行者、以及值信息用户易理解的访问控制

本芯片实现了可存储值信息的各种应用。值信息包括各种类型。例如，可以想到的下述类型信息可作为芯片中的信息：

－不能由芯片所用者来改变，但是仅可以由信息发行者来改变的信息（例如，电子门票的坐号）；

－即使不能对芯片所用者也不能公开的信息（例如，拥有改变电子门票的密钥）；

－仅由芯片所用者来完全控制的信息（例如，芯片所用者的个人信息）；以及

－可被任何人读取的信息。

本芯片同等的对待访问控制列表中的“发行者”、“所用者”以及“其

他者”SCs，并且发布了一 API 以根据每个 SC 所拥有的授权来改变访问控制列表。按照这种方式，本芯片可灵活的执行对诸如访问权的限制、免除、或传输这样的控制。

1.5 具有重新运行性能的事务机制

重要的是可安全地将值信息传送到本芯片。因此，尤其与值信息的创建和删除有关的是，本芯片提出了一事务机制以确保处理的可分性。在启动一事务之后，一个所发行的命令所导致的操作，可通过提交一个命令来终止该事务。若发出的是一异常终止命令，或若约定命令未在所定义的超时时间之内或之前到达，于是所发出的命令重新运行。类似的，如果本芯片在任何故障所造成的事务过程中断电了，那么当本芯片再次工作时重新运行处理发生在初始状态。

1.6 具有链接功能的存储结构

就信息处理能力以及诸如存储容量等等之类的资源而言，目前存在各种类型的防篡改芯片。在这些芯片中，存在这样一种情况，即具有较小资源的硬件不能将可想到的应用的所有值信息存储在芯片中。同时，考虑到处理效率，因此所希望的是为了保留而采用将值信息分布到芯片中的方法。因此，本发明支持混合方法以将内容分布到诸如本发明中的本芯片以及网络中的值信息存储服务器这样的多个信息容器中并对其进行保留。另外，本发明提供了内容间的链接功能以作为支持混合方法的机制。

1.7 相关芯片系统的相容性

就 API 命令协约、错误代码、静态的且动态的处理资源的方法等等而言，本芯片具有与和本发明有关的方案的其他结构的兼容性。例如，错误代码的一般部分为相关芯片和本芯片间共有。按照这种方式，本芯片可帮助具有开发相关芯片结构的经验的工程师来开发本芯片的应用。

1.8 与各种芯片相对应的适度标准化

本芯片提供了与各种 IC 卡相兼容的 API 系统，这些 IC 卡包括 8 位至 32 位 CPUs、接触型/不接触型/双重接触面、以及无线电频率识别（RF－IDs）的智能卡。

2．系统说明

2.1 系统结构

本芯片结构是分布式系统结构以可安全地存储防篡改值信息并且可安全地对计算机网络上的防篡改值信息进行交换。本芯片结构基本上包括下述组件。

（A）值信息网络基本结构（实体网络基本结构）

这是可对值信息进行交换的结构并且包括下述两个组件。

A－i）包含有本芯片的信息容器（CHs）

包含有本芯片的信息容器是分布式系统中的计算实体。包含有本芯片的信息容器存储值信息并提供了包含 API 的本芯片，因此允许对来自外部的值信息进行操作。例如，包含有本芯片的信息容器包括以下：

本芯片：可安全地存储值信息的防篡改 LSI 芯片。

本芯片盒：防篡改盒内的大容量电子安全罩，这可安全地存储值信息。

A－ii）包含有本芯片的服务客户机（SCs）

包含有本芯片的服务客户机是用于通过本芯片 API 来对存储在包含有本芯片的信息容器中的值信息进行访问的计算实体。例如，包含有本芯片的服务客户机包括下述类型。

本芯片：一个较完善的本芯片还具有作为客户机的功能。

发布服务器：可发布值信息并将值信息存储在包含有本芯片的信息容器中的服务器。

服务场系统：使用存储在本芯片中的值信息的应用系统。服务场系统例如包括电子门票检查门。

*注意：还存在这样一种计算实体，即可用作包含有本芯片的服务客户机以及芯片盒。

（B）认证/加密网络基础结构（AENI）

在本芯片可实现公钥加密系统的认证和加密的情况下，这是一认证系统。认证站起主要的作用。

（C）应用网络基础结构（ANI）

这相当于取决于应用的各种通信系统。在电子门票的情况下，例如，应用网络基础结构包括用于指示搜索或购买门票之类的网络协议。在ANI帧内执行直至购买的指令，并且在可提供本芯片功能的ENI内执行将门票作为值信息进行传送。

2.2 本芯片标识符（本芯片ID）

包含本芯片的信息容器以及服务客户机具有唯一的标识符，其每一个均是指本芯片标识符（本芯片IDs）。在网络中使用本芯片ID以对包含有本芯片的信息容器和服务客户机进行认证、对信息进行路径控制等等。本芯片由16个八位字节（128位）来表示。

*本芯片ID是在ENI通信协议中所使用的标识符的变换格式。本芯片ID未定义本芯片内的内部保留格式。

*考虑到便于实现路径控制，还可以想到的是采用可将ENI物理层上的一协议的标识符直接定义为本芯片ID的实现方法。例如，在无需修改的情况下，即可将网络地址的较低位用于本芯片ID。

*将所用者的本芯片ID定位为“0x00～00”（全“0”）。

例如，如果在以所用者的认证模式来对包含有本芯片的服务客户机进行认证之后，具有本芯片ID“X”的包含有本芯片的服务客户机建立了一会话，且在会话中创建了一文件、文件的发行者ID存储为“0”。

*将本芯片ID定义为“0xff～ff”（全“1”）。

3．关于本芯片的梗概

3.1 本芯片

本芯片是一较小尺寸的计算实体以使用户可携带值信息，利用该值信息上述用户可直接接收来自各种企业的服务。本芯片是这些规范主要处理的对象。通常，认为本芯片主要是由诸如IC卡、智能卡、或便携式终端这样的硬件来实现的。考虑到信息处理能力与这些硬件所提供的资源之间存在很大的差异，并且考虑到防篡改技术仍然处于发展阶段，因此本发明的规范提供了与各类硬件相对应的各种规范，同时只使接口标准化，由此制成了一系列芯片。

表1

本芯片/8	用于 8 位 CPU 芯片
本芯片/16	用于 16 位 CPU 芯片
本芯片/32	用于 32 位 CPU 芯片
本芯片/T	用于终端

表：一系列本芯片的规范

3.2 数据结构模型

从外面来看存储在芯片盒或本芯片内的值信息是可见的，因为它具有下述分层的数据结构。

● 文件夹（folder）

用于对多组文件进行编译的结构。

只有路由文件夹存在于本芯片/16 中。

● 文件

资源用于存储值信息。尽管一记录存储在该文件中，但是本芯片/16 采用这样一种结构，即在该结构中只有单一的记录存储在一文件中。

3.3 本芯片 API

本芯片是由包含有本芯片的服务客户机中的本芯片 API 来操作的。

基本上，本芯片 API 是基于利用下述过程的一会话的协议：

—在本芯片与包含有本芯片的服务客户机之间进行相互认证并建立加密通信路径，

→建立一会话（事务）；

—对命令消息进行传输；并且

—中止该会话（事务）

*一个会话/事务方法包括频繁的进行消息交换，因此这不适合需要诸如“&Go”这样的高应答的应用。因此，认证或加密被用于每个命令消息。就部分命令而言，也可采用非会话协议。

*以非会话的方式来处理所发出的“会话 ID=0”的命令。

■会话命令

eopn_ses 打开会话

ecfm_ses 确认会话

ecls_ses 关闭会话

■事务命令

eopn_tra 打开事务

ecfm_tra 确认事务

ecom_tra 关闭事务

eabo_tra 异常中止事务

■文件操作命令

ecre_fil 创建文件

edel_fil 删除文件

etra_fil 传输文件

eupd_fil 更新文件模式

eenc_fil 加密文件

edec_fil 解密文件

■记录操作命令

eupd_rec 更新纪录

erea_rec 读取纪录

■密钥实体操作命令

ecre_key 创建密钥

edel_key 删除密钥

eupd_key 更新密钥

■认证辅助命令

ecfm_cer 确认认证（certificate）

■控制命令

epol_chp 轮询芯片

eini_car 初始化卡

eupd_cer 更新 My 认证

eupd_cpk 更新 CA 公钥

3.4 认证、访问控制、以及加密

一旦建立了该会话，本芯片和包含有本芯片的服务客户机则执行相

互认证。由于用于认证的各类具体算法是专用于包含在本芯片中的硬件等等，因此可使用上述用于认证的各类具体算法。

*认证模式

认证包括一（信息）发行者模式和一所用者模式，这两种模式是在认证的过程中指定的。根据每个模式，认证算法（通常地）有极大的不同。

（信息）发行者模式:

该模式用于认证作为文件发行者的包含有本芯片的服务客户机。在由发行者模式进行认证之后，由包含有本芯片的服务客户机所创建的文件可访问发行者的授权，并且其他的资源可访问其他授权。

所用者模式：

该模式是用于认证作为芯片所用者的包含有本芯片的服务客户机。通常使用诸如口令这样的友好认证。由所用者模式所认证的包含有本芯片的服务客户机拥有所用者的授权。

• 在相互认证之后的本芯片的状况

在认证之后，本芯片保留了下述信息：

—已被正式认证的包含有本芯片的服务客户机的本芯片 ID;

—用于对消息进行加密以将其传送到包含有本芯片的服务客户机上的公钥;

—用于对来自包含有本芯片的服务客户机处的消息进行解密的公钥;

—会话 ID；以及

—会话模式（发行者模式/所有者模式）。

• 访问控制

将下述信息附着于芯片和文件（以及记录）上：

—表示发行者的本芯片 ID；以及

—访问控制列表（取决于授权所允许的操作）。

该访问受到由下述所定义的会话所拥有的授权的限制。

• 文件访问模式

可使具有本芯片 ID=eid 的 SC 可访问本芯片（其可创建文件 F=F.eid 的本芯片 SC 的芯片 ID）中的文件 F 的模式包括下面三种类型：

1．所有者访问

在通过了所有者认证之后，以会话方式进行访问

*在这种情况下，eid=0x00（全“0”）

2.（信息）发行者访问：

在通过了（信息）发行者认证之后，以会话方式进行访问，其中 eid=F.eid。

3．其他访问

在通过了创建者认证之后，访问会话或以非会话方式访问，其中 eid!=F.eid。

• 路由文件夹的访问控制列表

表 2

F	E	D	C	B	A	9	8
保留	保留	保留	保留	保留	保留	保留	保留
7	6	5	4	3	2	1	0
保留	保留	ACL5	ACL4	ACL3	ACL2	ACL1	ACL0

ACL0=0/1 ecre_fil 所有者访问不可/可以

*所有者的访问是否会允许在路由文件夹中创建一文件

ACL1=0/1 ecre_fil 其他访问 不可/可以

*除所有者访问之外的一访问是否会允许在路由文件夹中创建一文件

ACL2=0/1 edel_fil 所有者访问不可/可以

*所有者的访问是否会允许从路由文件夹中删除一文件

ACL3=0/1 edel_fil 其他访问 不可/可以

*除所有者访问之外的一访问是否会允许从路由文件夹中删除一文件

ACL4=0/1 etra_fil 所有者访问不可/可以

*所有者的访问是否会允许对来自路由文件夹的一文件进行传送

ACL5=0/1 etra_fi 其他访问 不可/可以

*除所有者访问之外的一访问是否会允许对来自路由文件夹的一文件进行传送

表 3

	Cre_fil	Del_fil	Tre_fil
所有者访问	ACL0	ACL2	ACL4
发行者访问	_	_	_
其他访问	ACL1	ACL3	ACL5

*路由文件夹的（信息）发行者作为所有者进行管理。

*换句话说，用于将值信息装入路由文件夹且用于创建密钥实体的通用值信息服务器根据路由文件夹而与“其他访问”相对应。

• 一文件的访问控制列表

表 4

F	E	D	C	B	A	9	8
ACLf	ACLe	ACLd	ACLc	ACLb	ACLa	ACL9	ACL8
7	6	5	4	3	2	1	0
ACL7	ACL6	ACL5	ACL4	ACL3	ACL2	ACL1	ACL0

ACL0=0/1 eupd_rec 所有者访问 不可/可以

ACL1=0/1 eupd_rec 其他访问 不可/可以

ACL2=0/1 erea_rec 所有者访问 不可/可以

ACL3=0/1 erea_rec 其他访问 不可/可以

ACL4=0/1 eupd_fim 所有者访问 不可/可以

ACL5=0/1 eupd_fim 其他访问 不可/可以

ACL6=0/1 elst_fil 所有者访问 不可/可以

ACL7=0/1 elst_fil 其他访问 不可/可以

ACL8=0/1 edel_fil 所有者访问 不可/可以
ACL9=0/1 edel_fil 其他访问 不可/可以

ACLa=0/1 etra_fil 所有者访问 不可/可以
ACLb=0/1 etra_fil 其他访问 不可/可以

ACLc=0/1 eenc_fil 所有者访问 不可/可以
ACLd=0/1 eenc_fil 其他访问 不可/可以

ACLe=0/1 edec_fil 所有者访问 不可/可以
ACLf=0/1 edec_fil 其他访问 不可/可以

表 5

	upd-rec	rea-rec	upd-fim	lst-fil	del-fil	tra-fil	enc-fil	dec-fil
所有者访问	ACL0	ACL2	ACL4	ACL6	ACL8	ACLa	ACLc	ACLe
发行者访问	总是可以	总是可以	总是可以	总是可以	总是可以	总是可以	总是可以	总是可以
其他访问	ACL1	ACL3	ACL5	ACL7	ACL9	ACLb	ACLd	ACLf

• 密钥实体的访问控制列表

表 6

F	E	D	C	B	A	9	8
ACLf	ACLe	ACLd	ACLc	ACLb	ACLa	ACL9	ACL8
7	6	5	4	3	2	1	0
ACL7	ACL6	ACL5	ACL4	ACL3	ACL2	ACL1	ACL0

ACL0=0/1 eupd_key 所有者访问 不可/可以

ACL1=0/1 eupd_key 其他访问 不可/可以

ACL2=0/1 erea_key 所有者访问 不可/可以

ACL3=0/1 erea_key 其他访问 不可/可以

ACL4=0/1 eupd _kym 所有者访问 不可/可以

ACL5=0/1 eupd _kym 其他访问 不可/可以

ACL8=0/1 edel_key 所有者访问 不可/可以

ACL9=0/1 edel_key 其他访问 不可/可以

ACLc=0/1 eenc_fil 所有者访问 不可/可以

ACLd=0/1 eenc_fil 其他访问 不可/可以

ACLe=0/1 edec_fil 所有者访问 不可/可以

ACLf=0/1 edec_fil 其他访问 不可/可以

表 7

	upd-key	rea-key	upd-kym	del-key	enc-fil	dec-fil
所有者访问	ACL0	ACL2	ACL4	ACL8	ACLc	ACLe
发行者访问	总是 可以	总是 可以	总是 可以	总是 可以	总是 可以	总是 可以
其他访问	ACL1	ACL3	ACL5	ACL9	ACLd	ACLf

4．本芯片 API 协议

4.1 包类型

*包类型右端的符号“□”表示代码化会话中的未加密部分，符号“■”表示代码化会话中的已加密部分。

*利用低端在前格式来存储等于或大于 2 个八位字节的数字数据。

路由报头

表 8

D0	D1	D2	D3	D4	D5	D6	D7	
本芯片 ID（目的 0）								□
本芯片 ID（目的 0）								□
本芯片 ID（源 0）								□
本芯片 ID（源 0）								□ □ □ □

本芯片会话单元（向前）

表 9

D0	D1	D2	D3	D4	D5	D6	D7	
SID	序列号	命令 ID		数据长度（字节数目）		保留		□ □
数据								■
……								■
数据								■ ■ ■

本芯片会话单元（向后）

表 10

D0	D1	D2	D3	D4	D5	D6	D7	
SID	序列号	错误代码		数据长度（字节数目）		保留		□ □
数据								■
……								■
数据								■ ■ ■

MAC，报尾

表 11

D0	D1	D2	D3	D4	D5	D6	D7	
MAC ID		MAC 长度		保留				□
MAC (1)								□
MAC (MD5 例如：2)								□ □ □

*在加密之后将 MAC 附着于数据上。

4.2 命令标识符的列表（命令 ID）

一个由 1 字节的正整数来表示本芯片 API 的命令标识符（命令 ID）。

通常按照一系列本芯片/8、16 和 32 来定义命令标识符，如下所示。

表 12

ID	助记符	8/16/32	含义
0x01	ecpn_ses	○◎◎	打开会话
0x02	ecfm_ses	○◎◎	确认会话
0x03	ecls_ses	○◎◎	关闭会话
0x04	eopn_ses	×○○	打开事务
0x05	ecfm_tra	×○○	确认事务
0x06	ecom_tra	×○○	提交事务
0x07	eabo_tra	×○○	异常中止事务
（0x11	ecre_fol	××○	创建文件夹）
（0x12	edel_fol	××○	删除文件夹）
（0x13	eupd_fom	××○	更新文件夹模式）
0x21	ecre_fil	○○○	创建文件
0x22	edel_fil	○○○	删除文件
0x23	etra_fil	×○○	传送文件
（0x24	erdm_fil	××○	修复文件）

0x25	eenc_fil	×○○	已加密文件
0x26	edec_fil	×○○	已解密文件
0x27	eupd_fim	○○○	更新文件模式
0x2F	elst_fid	○○○	列表文件 ID
（0x31	ecre_rec	××○	创建记录）
（0x32	edel_rec	××○	删除记录）
0x33	eupd_rec	○○○	更新记录
0x34	erea_rec	○○○	读取记录
（0x35	eupd_rcm	××○	更新记录模式）
0x41	epol_car	○○○	轮询卡
0x51	eini_car	○○○	初始化卡
0x52	eupd_cer	×○○	更新 My 证书
0x53	eupd_cpk	×○○	更新 CA 公钥
0x61	ecre_key	×○○	创建密钥
0x62	edel_key	×○○	删除密钥
0x63	eupd_key	×○○	更新密钥
0x71	ecfm_cer	×○○	确认证书

表：本芯片 API 的命令标识符列表

*未填入本芯片/16 第一版本

4.3 错误代码列表

由有符号整数来表示本芯片中的错误代码。当正常结束操作时，将返回 E_OK=0 或一个表示操作结果的正数。如果未正常结束操作或发生了一些错误，则返回一个负数。通常由下述来表示错误代码。

*“E_”表示在芯片中进行本地命令处理的过程中所出现的一般性错误（为相关芯片的错误代码所共有）

*“EN_”表示有通信有关的一般性错误（除会话错误之外，为相关芯片的错误代码所共有）

*“ES_”主要表示与安全相关的一般性错误（专用于本芯片）

表 13

代码	助记符	含义
-0x00	E_OK	正常结束
-0x05	E_SYS	系统错误
-0x0a	E_MOMEN	内存不足
-0x11	E_NOSPT	不支持的功能
-0x21	E-PAR	参数错误
-0x23	E_ID	无效的 ID 号
-0x34	E_MOEXS	不存在这样的目标
-0x3f	E_OBJ	无效的目标状态
-0x41	E_MACY	不可访问存储器，违规地访问存储器
-0x42	E_OACV	访问目标违规
-0x55	E_TMOUT	轮询失败或超时
-0x72	E_OBJNO	本芯片不可访问指定的目标号
-0x73	E_PROTO	本芯片不支持该协议
-0x74	E_RSFN	本芯片不支持该命令
-0x77	EN_PAR	参数不在本芯片所规定的范围内
-0x7a	EN_EXEC	由于本芯片端的资源不足而不可执行
-0x7f	EN_NOSES	没有指定的会话
-0x81	ES_AUTH	认证错误
-0x82	ES_EACL	访问控制违规

表：本芯片 API 的错误代码列表

4.4 MAC，报尾

表 14

MAC ID	长度（字节）	内容
0x01	16	校验总和
0x02	2	CRC-CCITT

0x03	4	保留
0x04	16	保留
0x05	16	具有 MD5 的 HMAC（RFC 2085）
0x06	16	保留

1 校验总和

表 15

D0	D1	D2	D3	D4	D5	D6	D7
MAC ID		MAC 长度		保留			
MAC(1)							
.....							
MAC(16)							

2 CRC-CCITT

CRC-CCITT 中的一阶多项式
G(X)=X^16+X^12+X^5+1

*当包号码是多位数时，在将后半字节填充为空值后，执行 CRC.

表 16

D0	D1	D2	D3	D4	D5	D6	D7
MAC ID		MAC 长度		保留			
MAC							

5 具有 MD5 的 HMAC[RFC 2085]

这在会话/事务加密通信路径中使用。

用于加密通信路径的密钥加密的密钥的最初 64 字节被作为密钥 K。这里，通过下述公式来计算 HMAC。

MDS(K xor opad | MDS (K xor ipad | TEXT))

opad+the byte 0x36 repeated 64 times

ipad+the byte 0x36 repeated 64 times

“ | ”表示简单位追加。

例如，(“0100” | “1001”) = (“01001001”)

*在建立会化之后执行加密通信的情况下，使 16 字节的加密密钥重复 4 次所获得的数目，换句话说，通过如下追加而使在 epon_ses( )的 RSA 认证情况下所交换的随机数被作为 64 字节的密钥。

K=Ra | Rb | Ra | Rb | Ra | Rb | Ra | Rb |

表 17

D0	D1	D2	D3	D4	D5	D6	D7
MAC ID		MAC 长度		保留			
MAC(1)							
MAC(2)							

4.5 会话通信和非会话通信

就与本芯片的通信而言，在“相互认证”之后建立了作为“加密”通信路径的会话/事务，并且此后执行“已加密”信息的通信。就“erea_rec command”而言（例如，读取记录的内容），支持无需通过“认证”阶段即可执行的非会话通信。在这种情况下，可通过检索具有“session ID=0”的命令来启动，以代替建立一会话。若相关的文件允许“其他访问”，则非会话通信的访问是成功的。就这一点而言，访问包是未加密的。这用在诸如门票控制这样的非常需要“touch&go 应用程序”的地方。

*利用该模式服务提供者则要承受很高的风险。下述声源卡也被采用。

5．eTP 密钥证书

在会话建立的过程中进行认证时，使用该证书。

1．概述

在利用本芯片/16 建立一会话的过程中所进行的相互认证是基于基本公钥的且存在 PKI 这样的前提。

—为包含有本芯片的 CHs 以及包含有本芯片的 SCs 的每一个均分配了本芯片 ID，并且在配给的过程中分配了成对的公钥和密钥以用于认证。

—总是将包含有本芯片的 CA 所签名的证书存储在包含有本芯片的 CH 或包含有本芯片的 SC 内，因此在建立会话/事务时，包含有本芯片的

CH 和包含有本芯片的 SC 可表明彼此拥有正式所配给的一对本芯片 ID 和公钥。

*尽管假定将来存在多个包含有本芯片的 CA，但是包含有本芯片的 CA 并不局限于该版本的这一个。

—在证书中，包含有本芯片的 CA 的签名（S_ca(id，PK_id)）证明了本芯片 ID、成对的公钥、以及(id，PK_id)是正式所配给的。

2．详细说明

由于在认证步骤之前存在一步骤，因此包含有本芯片的节点（共同涉及包含有本芯片的 SC 和包含有本芯片的 CH）彼此传送他们自己的本芯片的公钥以作为本芯片设备的发行者所签名的证书。*参见注释（1）

包含有本芯片的节点具有证书形式的由证书发行者所签发的一密钥的公钥。利用该密钥既可确认其合法性。

因此，对下述进行证实。*参见注释（2）

“从相对节点所传送来的公钥的正确性”=“发行着正式的签署”+“有效期内的证书”

当有效性未被确认，则中断通信而无需建立会话。*参见注释（3）

*注释（1）

在对基本公钥进行认证时，还应当考虑与这样一种情况有关，即当对此后会话中的高位值进行传输/交换时证书被“取消且无效”。本芯片/16 使用下述之一种：

1．CRL（无效列表），（然而，数据量变大）；或者

2．用于查询指定的证书是否被“取消且无效”的服务器访问。

*注释（2）

此外，还可设置如注释（1）所述的一查询以确定证书是否被“取消且无效”。

*注释（3）

在确认证书的有效性结束之前，行之有效的是实验性的启动认证步骤并采用由另外的任务来确认其有效性的实现方法，同时将认证所必需的信息包传送到对方，或等待来自对方的认证所必需的信息包。尽管这

样的实现需要中断和多任务 OS 以在芯片中实现，但是这仍是有效的装置以在短处理时期内即可完成整个认证。

前述说明是可减少在对诸如本芯片或卡这样的节点的通信进行初始化设置所需时间的设备。在服务机上所实现的节点总是首先脱机的确认签名，然后确认“取消性和无效性”。这防止了由于对来自具有无效证书设备的无效任何凭证进行传输所造成的浪费服务器查询处理的 DoS。

eTP 密钥证书的格式（版本 0.1）

表 18

D0	D1	D2	D3	D4	D5	D6	D7
ver	保留	证书号		签名法	保留	发行者 ID	
有效期开始				有效期结束			
本芯片 ID（1）							
本芯片 ID（2）							
CAKEY#	保留	数据长度		签名长度		保留	
公钥（1）							
……							
公钥（17）							
签名（1）							
……							
签名（16）							

• ver：证书格式的版本号，在 0.1 版本中将其设置为“1”。

• 证书号：本芯片 ID 的密钥的证书序列号

• 签名方法：签名算法指定符

0：直到证书的密钥字段的部分采用 SHA-1 杂散数，剩余部分采用 RSA 加密(实现)

1：保留

2：保留

3：保留

• 发行者 ID：证书的发行者（例如 CA 站）ID，在 0.1 版本中其固定为“0”

• 有效期开始，有效期结束：证书的有效期限

• 未签名的 4 字节整数（从 2001 年 1 月 1 日 0：00AM 开始对秒数进行累积）

• CAKEY#:当密钥签名使用多个密钥时，为区分起见的每个密钥的 ID 号

• 密钥长度：具有已添加报尾的密钥位长。

*当密钥不是证书中 8 字节的多位数时，将“0”（空值填充数）添加到用于存储密钥的密钥字段的结尾，并且因此将密钥作为 8 字节的多位数数据来存储。密钥字段中的“密钥长度”位是有效的密钥。

• 签名长度：将签名所使用的杂散数（例如 sha-1）的位长添加到报尾

• 杂散数最后在必要时利用空值填充数来获得 8 字节的多位数数据，然后通过利用 CA 密钥对数值进行加密来创建签名

当确认了签名，利用存储在包含有本芯片的节点中的 CA 服务器的公钥来对添加有报尾的 8 字节多位数数据进行解密。这些已解密值中的头部签名位长成为确认过程中所使用的哈西值。

• 长度的具体实施例（当使用 RSA/sha-1 签名时）

—RSA

利用 RSA 将本芯片认证的公钥设置为 1024 位

1024 位=128 字节=16*8 字节

“sha-1”作为签名过程中的杂散数函数

—sha-1

sha-1 的输出是 160 位

160 位=20 字节

该字节长是空值填充数的 8 倍，并且 24 字节数据是由 RSA 加密来签名的。

按照如下来计算证书的总长度：

证书的长度等于

• 固定长度部分（40 字节）；并且

密钥部分的长度等于

• 签名部分（128 字节）

因此，总长度等于这两个部分之和（168 字节）。

因为在加密之后签名部分需要 1024 位，因此可使用 128 字节的长度。

然而，应当注意的是密钥部分并不是简单的等于 128 字节。

这是因为，就 RSA 密钥而言，作为公钥的 1024 位的公共引用应当考虑 1024 位的成对 N 和指数 e（e，N）。因此，必须将两者（e，N）都输入到密钥部分。

在 RSA 的情况下，按照下述所描述的将 e 和 N 输入到密钥字段。

e 的字节长：（1 字节）

值 e 本身：按照需要将其被扩大为字节单元以被填埋为字节序列。随后，值 N 被填埋为字节序列。

考虑（e，N）中 e=3 的情况，例如，按照下述的将 130 字节的字节数据输入到密钥字段。

公钥=0x01|0x03 N 位（128 字节）的字节序列

因为 130 不是 8 的倍数，因此将该值填补为 136 字节。因而，密钥数据占用了 136 字节。

因此，按照如下来计算整个长度：

固定长度部分	（40 字节）
+必要部分所占用的长度	（136 字节）
+签名部分	（128 字节）
	=304 字节

按照这种方式，当 e=3 时总长等于 304 字节。

6．密钥实体

密钥实体是可编制加密密钥的系统数据格式和利用加密密钥来对值信息进行操作所必需的信息。本芯片将加密密钥和信息存储在普通的文件中。

表 19

D0	D1	D2	D3	D4	D5	D6	D7
密钥目标 ID		加密模式指定符		保留			
KEY（1）							
KEY（2）							
银行 ID		保留		支付差额			
LEN				保留			

加密模式指定符：说明了每一密钥使用了 KEY(1)和 KEY(2)中的哪一个加密算法。如下所述，由与说明“eopn_ses”中的加密通信路径的加密算法相同的方式来执行该说明。

表 20

加密方式指定符	加密类型
0x01	DES
0x02	3DES
0x03	Rijndel
0x04	Hierocrypt-3
0x05	camellia

7．本发明的标准内容格式

标准内容格式是可在本芯片内进行值信息操作的标准内容数据格式。通过密钥实体，可添加加密/解密操作并可在这种情况下自动改变密钥实体内部的描述信息。

基本上，这里假设可在本芯片内确定地执行这样一个销售模型，即该模型可对已加密内容进行解密而收费。

■整个内容格式

表 21

（A）内容头部
（B）账户头部
（C）加密内容本体

（D）签名报尾

（A）内容头部

表 22

D0	D1	D2	D3	D4	D5	D6	D7
内容 ID		片段 ID		出版者信息			

（B）账户头部

表 23

D0	D1	D2	D3	D4	D5	D6	D7
支付方案		表达式长度		参数（1）		参数（2）	
参数（3）		参数（4）		参数（5）		参数（6）	

（C）加密内容

表 24

D0	D1	D2	D3	D4	D5	D6	D7
加密方式指定符		数据长度		保留			
加密 KEY（1）							
加密 KEY（2）							
已加密内容本体（1）							
……							
已加密内容本体（m）							

*利用存储在密钥实体中的密钥来对加密 KEYs（1）和（2）进行加密，上述密钥实体是对内容进行解密所必需的。在本芯片/16 中，只有公用密钥加密被认为是密钥实体，这是由于文件存储能力所造成的，并且在这里不考虑公开密钥加密（因为在加密之后，其加密密钥的数据长度将会变长）

（D）签名报尾

表 25

D0	D1	D2	D3	D4	D5	D6	D7
签名 ID		签名长度		保留			

签名（1）
……
签名（n）

*将签名模式设置为与 eTP 包的签名报尾相同的模式

下述方法通过利用 K1 和 K2 来创建 K，上述 K1 和 K2 是载具有 MD5 的 HMAC 的情况下利用作为 64 字节密钥的密钥实体来包含在内容中的加密 Key（1）和加密 Key（2）进行解密所获得的

K=K1|K2| K1|K2| K1|K2 |K1|K2

■账户头部

表 26

支付方法	支付方案参数	参数 2	…
固定价格支付	0x01	价格（单位：sen(0.01yen)）	_

■已加密内容

表 27

加密方式指定符	加密类型
0x01	DES
0x02	3DES
0x03	Rijndel
0x04	Hierocrypt-3
0x05	camellia

■签名报尾

与本芯片的信息包的指定相类似

8．通过利用密钥实体来解决处理对本发明的标准内容所进行的结算过程

采用下述方法作为利用密钥实体的计算方法。

（1）对加密内容的账户头部的收费额进行计算（在本芯片/16 中这仅仅是固定额）。

（2）对密钥实体的 MONEY 字段的结算额进行扣除处理。

9．本芯片 API 的规范说明（数据类型定义）

B 已签名的 1 字节整数

UB 未签名的 1 字节整数

H 已签名的 2 字节整数

UH 未签名的 2 字节整数

本芯片 ID 16 字节整数

SID 未签名的 1 字节整数（会话 ID）

FID 未签名的 2 字节整数（文件/文件夹 ID）

RID 未签名的 2 字节整数（记录 ID）

CACL 未签名的 2 字节整数（芯片访问控制列表）

FACL 未签名的 2 字节整数（文件访问控制列表）

KEYACL 未签名的 2 字节整数（密钥目标访问控制列表）

TIME 未签名的 4 字节整数（从 2000 年 1 月 1 日 0：00AM 开始对秒数进行累积）

ERR 已签名的 2 字节整数

• 利用 C 语言表达定义的例子

```
#typedef unsigned char UB:
#typedef unsigned short UH:
#typedef unsigned long UW:

#typedef struct etronid{
        UB item[16];
}ETRONID;                    /*the present chip ID*/

#typedef   UB SID;           /*Session ID*/

#typedef   UB FID;           /*File ID*/
```

```
#typedef   UH RID;              /*Record ID*/

#typedef   UH CACL;             /*Chip Access Control List*/
#typedef   UH FACL;             /*File Access Control List*/
#typedef   UH KEYACL;           /*Key Entity Access Control List*/

#define   ISSUER_MODE1   /*Issuer Mode*/
#define   OWNER_MODE2    /*Owner Mode*/

#define   FILE_DYNAMIC 0x01     /*Dynamic Multi-record File*/
#define   FILE_STATIC 0x02      /*Static Single-record File*/

#typedef   UH TME;              /*Time*/
```

10．本芯片 API 的规范说明（命令的定义）

10.1 会话管理命令组

会话结构

打开/确认会话

[特征概要]

本芯片构建了安全的会话通信路径。当路径 1 符合要求时在使用认证的情况下可使用“eopn_ses”，并且在使用需要路径 2 的认证的情况下按照这个次序使用“ecnf_ses”。

通过这个会话结构，可共享临时的公用加密密钥，该临时的公用加密密钥只是在包含有本芯片的客户端、呼叫端、本芯片端、以及被呼叫端之间间进行会话期间有效。

根据包内的参数可选择认证和加密类型的方式。当指定了本芯片上的不可用的认证/加密时，则会出现错误。

[函数表达]

ERR eopn_ses (ETRONID destid, ETRONID srcId, TIME t,
UB authmode, UB authAlgorithm,
Ub sessionAlogrithm, UH len,
UB*authData, UH*rlen,
UB**rAuthData)

DestId	本芯片的芯片 ID，本芯片即就是发出命令的对象（目的本芯片 ID）
srcId	包含有本芯片的服务客户机的本芯片 ，服务客户机是调用命令的（源本芯片 ID）
t	时间
authMode	指定认证模式
authAlgorithm	认证所使用的算法指定符
sessionAlgorithm	在认证之后会话的加密算法的指定符
len	输出包长度（字节数）
authData	用于认证的通过包含有本芯片的服务客户机而到达本芯片的数据（len-16）
rlen	返回包长度（字节数）
rAuthData	用于认证的从本芯片返回至包含有本芯片的服务客户机的数据

[参数值]

□authMode

0x01 ISSUER	发行者模式
0x02 OWNER	所有者模式

□authAlgorithm

0x01 保留

0x02 RSA（利用证书进行离线认证）

0x03 RSA（在线认证以用于确认 CRL）

0x04 保留

□sessionAlgorithm

0x01 保留

0x02 保留

0x03 保留

0x04 保留

0x05 保留

[返回值]

在路径 1 的认证中

>0 会话 ID（当正常关闭时）

<0 错误代码

在路径 2 的认证中

<=0 错误代码

[输出包格式]

表 28

D0	D1	D2	D3	D4	D5	D6	D7
保留	序列号（0x01）	命令 ID (0x01)		Len (字节)		保留	

T	authMode	authAlgorithm	sesAlgorithm	authData
authData				
.....				
authData				

表 29

字段名	数据长度	含义
Command ID	2(=0x01)	命令代码
Len	2	包长度（字节数）
T	4	时间
Authmode	1	认证模式
AuthAlgorithm	1	认证算法指定符
sessionAlgorithm	1	会话加密算法指定符
AuthData	Len-16	用于认证的数据

[返回包格式]

表 30

D0	D1	D2	D3	D4	D5	D6	D7
保留	序列号（0x01）	错误代码		rlen (字节)		保留	
T				authMode	authAlgorithm	sesAlgorithm	authData
rAuthData							
.....							
rAuthData							

表 31

字段名	数据长度	含义
Error Code	2	错误代码（，会话 ID（在路径 1 的

		情况下))
Rlen	2	返回包长度（字节数）
RAuthData	Rlen-8	用于认证的数据

[函数表达式]

ERR eopn_ses (ETRONID destid, ETRONID srcId, TIME t,
UH len, UB*authData, UH*rlen ,
UB**rAuthData);

destId 本芯片的芯片 ID，本芯片即就是发出命令的对象（目标该芯片 ID）

srcId 包含有本芯片的服务客户机的本芯片 ，服务客户机是调用命令的（源芯片 ID）

t 时间

len 输出包长度（字节数）

authData 用于认证的通过包含有本芯片的服务客户机而到达本芯片的数据（len-12）

rlen 返回包长度（字节数）

rAuthData 用于认证的从本芯片返回至包含有本芯片的服务客户机的数据（rlen-8）

[返回值]

>0 会话 ID（当正常关闭时）

<0 错误代码

[输出包格式]

表 32

D0	D1	D2	D3	D4	D5	D6	D7
保留	序列号（0x02）	命令 ID (0x02)		Len (字节)		保留	
T				authData			
……							
authData							

表 33

字段名	数据长度	含义
Command ID	2(=0x01)	命令代码
Len	2	包长度（字节数）
T	4	时间
AuthData	Len-12	用于认证的数据

[返回包格式]

表 34

D0	D1	D2	D3	D4	D5	D6	D7
保留	序列号（0x02）	错误代码		rlen		保留	
rAuthData							
……							
rauthData							

表 35

字段名	数据长度	含义
Error Code	2	错误代码，会话 ID
rlen	2	返回包长度（字节数）
rAuthData	len-8	用于认证的数据

[详细说明]

在对包含有本芯片的服务客户机与本芯片之间的相互认证之后，“打开/确认命令”建立了一会话以作为安全的加密通信路径。具体的运作取决于认证和加密的方式，并且该方式通常取决于本发明 N 芯片的加密支持功能（实现的相关性）。会话建立协议包括发行者模式中的认证以及所有者模式中的认证，由“打开会话”命令中的 authmode 参数来区分这两者。

■通过公钥加密来进行发行者模式中的认证（authmode=ISSUER）

图 6 给出了通过公钥加密来进行发行者模式中的认证方法的示意图。以下参考图 6 对该认证方法进行详细的举例说明。

（1）eopn_ses（A→B）

《参数例》

- authMode 0x02(发行者)
- authAlgorithm 0x03(RSA:用于确认 CRL 的在线认证)
- sessionAlgorithm 0x02(3DES)
- authData 用于 A 的 eTP 公钥证书
- MAC 报尾 校验总和

《B 的作用》

• 确认 A 的 eTP 证书的有效性

—确认 eTP 证书的终止日期

—通过使用 CA 站→（2）的公钥来确认 eTP 证书上的签名

—确认 CRL 列表

• 取回 A 的公钥

（2）ecfm_cer(B→CRL Rep.Server)

《参数例》

• checkMode	0x00
• certificateId	**********
• MAC 报尾	校验总和

（3）ecfm_cer(B←CRL Rep.Server)

《参数例》

• Error Code	E_OK 或否
• MAC 报尾	校验总和

（4）eopn_ses(A←B)

《参数例》

• Error Code	E_OK 或否
• authMode	0x02(发行者)
• authAlgorithm	0x03(RSA:用于确认 CRL 的在线认证)
• sessionAlgorithm	0x02(3DES)
• authData	1.用于 B 的 eTP 公钥证书

2.采用A的公钥和64位随机数Rb进行RSA加密所获得的值

• MAC 报尾 校验总和

《A 端的作用》

• 确认 B 的 eTP 证书的有效性

—确认 eTP 证书的终止日期

—确认→到（5）的 CRL 列表

—通过使用 CA 站的公钥来确认 eTP 证书上的签名

• 从证书取回 B 的公钥

• 在采用 A 的密钥解密之后，取回 authdata2 中接收到的随机数 Rb

• 指定取回到的随机数 Rb 作为用于会话加密的 3DES 的密钥的一部分

• 利用 A 的密钥为取回到的随机数 Rb 创建数字签名

（5）ecfm_cer(A→CRL Rep.Server)

《参数例》

• checkMode 0x00

• certificateId **********

• MAC 报尾 校验总和

（6）ecfm_cer(A←CRL Rep.Server)

《参数例》

• Error Code E_OK 或否

• MAC 报尾 校验总和

（7）ecfm_ses（A→B）

《参数例》

•authData 1.采用 B 的公钥和 64 位随机数 Rb 进行 RSA 加密所获得的值，以及
2.附加并传送采用 A 的私钥进行 RSA 加密所获得的一个值，以及通过接受自 B 的随机数的 A 的私钥的数字签名

• MAC 报尾 校验总和

《B 方作用》

•取回 Ra 和采用用于 A 的公钥解密 authData 中接收到的数据从而获得的数字签名

• 指定取回到的 Ra 作为用于会话加密的 3DES 的私钥

• 为利用 B 的私钥取回到的 Ra 创建数字签名

• 确认取回到的带有用于 A 的公钥的签名

（8）ecfm_ses(呼叫者←被叫者)

《参数例》

• Error Code E_OK 或否

• authData 依据 B 的私钥为来自 A 的随机数 Ra 数字签名

• MAC 报尾 校验总和

《A 端作用》

• 利用 B 的公钥确认所接收到的签名

■认证算法（辅助）详述

■算法详述

·加密和认证的密钥

私钥 sk=d

公钥 pk=(e,n)

·用于普通句 m 的加密算法

E_pk(m)=m^e mod n

·已加密句 c 的解密算法

D_sk(c)=c^d mod n

·为消息 M 创建数字签名算法

f_sh(M)=(h(M))^d mod n

*此处，h（M）为诸如 MD5 或 SHA-1 的散列函数

·对消息 M 的数字签名 s 的检验算法

g_pk(M,s)=if(h(M)==s^e mod n)then 1 else 0 endif

■相应结点信息列表

·CA 站（C）

CA 公钥 pk_c(130B)

CA 密钥 sk__c(128B)

·呼叫方（A）

A 公钥 pk_a(130B)

A 密钥 sk_a(128B)

A 证书 cer_a(304B)

在 A 的证书上 CA 站的签名 s_(a,c)=(128B)

CA 站的公钥 pk_c(130B)

在相互认证过程中创建的随机数 Ra （8B）

• 被叫方（B）

B 公钥 pk_b(130B)

B 密钥 sk_b(128B)

B 证书 cer_b(304B)

在 B 的证书上 CA 站的签名 s_(b,c)=(128B)

CA 站的公钥 pk_c(130B)

在相互认证过程中创建的随机数 Rb（8B）

■基于 RSAd 的 eTP 的认证算法详述

eopn_ses：(A) →(B)

[所传送的信息]（cer_a） *总共 304 字节

A 的证书 cer_a:包括 pk_a 和 s_(a,c)

[B 的作用]

证书期满确认

证书签名确认 g_(pk_c)(cer_a,s_(a,c))

证书的 CRL 确认

eopn_ses:(A)←(B)

[所传送的信息] (cer_b|rb) *总共 432 字节

B 的证书 cer_b:包括 pk_b 和 s_(b,c)

B 创建的随机数（Rb） rb=E_(pk_a)(Rb)

[A 的作用]

证书期满确认

证书签名确认 g_(pk_c)(cer_b,s_(b,c))

证书的 CRL 确认

消息（rb）解密 Rb'=D_(sk_a)(rb)

为所接收到的随机数（Rb'）创建签名 srb=f_(sk_a)(Rb')

ecfm_ses: (A)→(B)

[所传送的信息] (x_a1|x_a2) *总共 256 字节

A 创建的随机数（Ra）和签名（srb） x_a1=E_(pk_b)(Ra)

x_a2=E_(pk_b)(srb)

[B 的作用]

消息（x_a）解密 Ra'=D_(sk_b)(x_a1)

srb'=D_(sk_b)(x_a2)

检验所接收到的签名（srb） g_(pk_a)(Rb,srb')

为所接收到的随机数（Ra'）创建签名 sra＝f_(sk_b)(Ra')

ecfm_ses:(A)←(B)

[所接收到的信息] （x_b）*总共 128 字节

B 创建的签名（sra） x_b=E_(pk_a)(sra)

[A 的作用]

消息（x_b）解密 sra'=D_(sk_a)(x_b)

检验所接收到的签名（sra） g_(pk_b)(Ra,sra')

*这样，A 和 B 均有一对（Ra, Rb）且采用特定算法从中创建会话公钥。

■在建立会话之后设置加密通信参数

例）3DES 情况

基于 E-D-E 模式采用 Ra-Rb-Ra 执行 3DES 加密。在此情况下，存储在“eopen_ses()”输出包的 sid 列中的值被用作会话 ID。也就是说，首先

被呼叫的呼叫者所指定的会话 ID 用得其所。

例）DES 情况

将 Rb 作为密钥进行 DES 加密。在此情况下，Ra 被用作会话 ID。

■依据私钥加密在主属模式（authmode＝OWNER）认证

图 7 给出了依据私钥加密在主属模式认证的方法的示意图。以下将参考图 7 描述依据采用“eopn_ses”和“ecfm_ses”的 2 路模式主属认证来建立会话的操作例。此模式基本上采用图 7 所示的方法。尤其，芯片内存储了主属密码，且利用密码执行认证。当开始认证时，当前芯片创建了一随机数并将随机数传送至 R/W。随后，R/W 方利用随机数加密密码，然后将密码传送至当前芯片。

（1）eopn_ses(A→ B)

创建随机数 Ra，然后

• ID_A；且

• Ra 被传送至 B

（2）eopn_ses(A←B)

创建随机数 Rb，然后，

• X1＝E1_Key(ID_A|Ra);且

• 计算 X2=E2_P(Pb)并将其返回 A

（3）ecfm_ses(A→B)

a.利用 D1_Key（X1）解密并确认 Ra。（执行认证以确保当前芯片质量）。

b.计算 R_B=D2_P(X2)并解密 X2 以取回 Rb。

c.通过部分位反相来修改 R_B 以创建 R'b，然后，

• 创建 Y＝E3_P(R'b)并传送至 B。

（4）ecfm_ses(A←B)

利用 D3_p（Y）执行解密并确认 R'b。（执行认证以确认有资格的主属正通过当前芯片服务客户机进行操作）

从 Ra 创建会话 ID（sid）并返回至 A

*此处，“|”为追加并加入位序列。例如，当 A＝“010010011”且 B=“00100101”，那么，（A|B）＝“01001001100100101”

ID_A 和 ID_B 分别表示当前芯片服务客户机（A）和当前芯片（B）的当前芯片 ID（128 位）。

*假设 E2 为 DES 操作，E3 为其中一变元为位反相并属于 DES 的操作。

会话的关闭

关闭会话

[特征概述]

会话被关闭。

[函数表达式]

ERR ecls_ses (SID sid, ETRONID destId, ETRONID srcID,

TIME t, UH len, UH*rlen);

destId 作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）

srcId 调用命令的当前芯片服务客户机的当前芯片 ID（当前源芯片 ID）

t 时间

len 一输出包长（一字节数）

rlen 一返回包长（一字节数）

[返回值]

<=0 错误代码

[输出包格式]

表 38

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	命令 ID（0x03）		Len（12）		保留	
T							

表 39

字段名	数据长	含义
命令 ID	2（=0x03）	命令代码
Len	2(=12)	包长（字节数）

t	4	时间

[返回包格式]

表 40

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		Len(8)		保留	

表 41

字段名	数据长	含义
错误代码	2	错误代码（、会话 ID）
rlen	2	返回包长（字节数）

[访问控制]

当前芯片服务客户机可利用此命令来构建会话。

[详细特征]

关闭所建立会话。

10.2 事务管理命令组

构建事务会话

打开/确认事务

[特征概述]

为当前芯片构建一事务会话。“eopn_tra”用在当满足 1 路时认证的情况，且“eopn_tra”和“ecnf_tra”以此顺序用于 2 路认证的情况

*基本操作类似于“eopn_ses”和“ecfm_ses”。然而，其不同在于对随后处理的返回能力。

[函数表达式]

ERR eopn_tra(ETRONID destID, ETRONID scrID, TIME t,
UB authMode, UB authAlgorithm,
UBsessionAlgorithm, UH len,
UB*authData, UH*rlen,
UB**aAuthData);

destID	作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）
scrId	调用命令的当前芯片服务客户机的当前芯片 ID（当前源芯片 ID）
t	时间
authMode	规定认证模式
authAlgorithm	用作认证的算法说明符
sessionAlgorithm	认证之后会话的加密算法说明符
len	一输出包长（一字节数）
authData	从当前芯片服务客户机传送到当前用于认证的芯片的数据（len-16）
rlen	一返回包长（一字节数）
rAuthData	从当前芯片返回到当前用于认证的芯片服务客户机的数据（len-8）

[参数值]

参照“eopn_ses”

[返回值]

在 1 路认证中

>0 会话 ID（普通关闭）

<0 错误代码

在 2 路认证

<=0 错误代码

[输出包格式]

表 42

D0	D1	D2	D3	D4	D5	D6	D7
保留	序列（0x01）	命令 ID（0x04）		Len		保留	
T				authMode	Auth Algorithm	ses Algorithm	auth Data
authData							
…							
authData							

表 43

字段名	数据长	含义
命令 ID	2（＝0x04）	命令代码
Len	2	包长（字节数）
T	4	时间
Authmode	1	认证模式
authAlgorithm	1	认证算法说明符
sessionAlgorithm	1	会话加密算法说明符
AuthData	len-16	认证数据

[返回包格式]

表 44

D0	D1	D2	D3	D4	D5	D6	D7
保留	序列（0x01）	错误代码		Rlen（字节）		保留	
rAuthData							
…							
rAuthData							

表 45

字段名	数据长	含义
错误代码	2	错误代码（、会话 ID(在 1 路情况下)）
rlen	2	返回包长（字节数）
rAuthData	rlen-8	认证数据

[函数表达式]

ERR ecnf_tra(ETRONID destId, ETRONID srcId, TIME t,
UH len, UB*authData,
UH*rlen, UB**rAuthData);

destId 作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）

srcId 调用命令的当前芯片服务客户机的当前芯片 ID（当前源芯片 ID）

t 时间

len 一输出包长（一字节数）

authData 从当前芯片服务客户机传送到当前用于认证的芯片的数据（len-12）

rlen 一返回包长（一字节数）

rAuthData	从当前芯片返回到当前用于认证的芯片服务客户机的数据（rlen-8）

[返回值]

>0	会话 ID（普通关闭）
<0	错误代码

[输出包格式]

表 46

D0	D1	D2	D3	D4	D5	D6	D7
保留	序列（0x00）	命令 ID（0x05）		len(字节)		保留	
t				authData			
...							
authData							

表 47

字段名	数据长	含义
命令 ID	2(=0x02)	命令代码
len	2	包长（字节数）
t	4	时间
authData	len-12	认证数据

[返回包格式]

表 48

D0	D1	D2	D3	D4	D5	D6	D7
保留	序列（0x02）	错误代码		rlen（字节）		保留	

rAuthData
...
rAuthData

表 49

字段名	数据长	含义
错误代码	2	错误代码（，会话 ID）
rlen	2	返回包长（字节数）
rAuthData	rlen-8	认证数据

[详细特征]

类似于“eopn_ses”和“ecfm_ses”涉及认证和加密。

然而，在建立事务之后只可发布有以下当前芯片 API。假如发布了其它 API 则将发生一错误。

●创建文件 ecre_fil

●删除文件 edel_fil

●创建记录 ecre_rec

●删除记录 edel_rec

●更新记录 eupd_rec

必须保证当发布“econ_tra”时反映了此处发布的命令序列，且在发布“eabo_tra”或一段特定时间不发布“ecom_tra”后超时的情况下，保证完全返回命令序列。

关闭事务

提交/终止事务

[特征概述]

关闭事务。“ecom_tra”用作关闭提交，且“eabo_tar”用作关闭终止。

[函数表达式]

ERR ecom_tra (SID sid, ETRONID destId,
ETRONID scrID, TIME t,UH len,
UH*rlen);

DestID 作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）

srcI 调用命令的当前芯片服务客户机的当前芯片 ID(当前源芯片 ID)

t 时间

len 一输出包长（一字节数）

rlen 一返回包长（一字节数）

[返回值]

<=0 错误代码

[输出包格式]

表 50

D0	D1	D2	D3	D4	D5	D6	D7
保留	序列号	命令 ID（0x06）		len(12)		保留	
t							

表 51

字段名	数据长	含义
命令 ID	2(=0x06)	命令代码
len	2(0x0C)	包长（字节数）

t	4	时间

[返回包格式]

表 52

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		rlen(8)		保留	

表 53

字段名	数据长	含义
错误代码	2	错误代码（、会话 ID）
Rlen	2(0x08)	返回包长（字节数）

[函数表达式]

ERR eabo_tra (SID sid, ETRONID destId,
ETRONID srcId，TIME t，UH len，
UH*rlen)；

destID 作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）

srcId 调用命令的当前芯片服务客户机的当前芯片 ID（当前源芯片 ID）

t 时间

len 一输出包长（一字节数）

rlen 一返回包长（一字节数）

[返回值]

<=0 错误代码

[输出包格式]

表 54

D0	D1	D2	D3	D4	D5	D6	D7
保留	序列号	命令 ID（0x07）		len(12)		保留	
t							

表 55

字段名	数据长	含义
命令 ID	2(=0x07)	命令代码
len	2(0x0C)	包长（字节数）
T	4	时间

[返回包格式]

表 56

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		rlen(8)		保留	

表 57

字段名	数据长	含义
错误代码	2	错误代码（、会话 ID）
Rlen	2(0x06)	返回包长（字节数）

[访问控制]

当前芯片服务客户机可利用此命令来构建会话。

[详细特征]

●所建立的事务被提交或终止。

●当事务被提交时，在“ecfm_tra”和“ecom_tra”之间发布的所有命令被映射到当前芯片的非易失性存储器。

●当事务被终止或等待特定时期之后未提交时，然后在“ecfm_tra”和“ecom_tra”之间发布的所有命令被映射到当前芯片的非易失性存储器。

10.3 文件管理命令组

[用于实现的受限项]

此系统中的文件具有以下限制。

●可用文件数上限 50

●文件 ID 0 至 49

●文件大小上限 256（字节）

文件的创建

创建文件

[特征概述]

创建一（空）文件。

[函数表达式]

```
ERR ecre_fil (SID sid, ETRONID destId, ETRONID srcId,
                    TIME t, UH len, FID fid, UH blk,
                    UH cnt, FACL facl, UH*rlen);
```

sid	会话 ID
destId	作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）

srcId	调用命令的当前芯片服务客户机的当前芯片 ID（当前源芯片 ID）
t	时间
fid	将被创建的文件 ID
facl	文件访问控制列表的一初始值
blk	文件的一起始地址（此处调整成“1”）
cnt	一文件长（一字节数）
len	一输出包长（一字节数）
rlen	一返回包长（一字节数）

[返回值]

>0	所创建的文件 ID（普通关闭）
<0	错误代码

[输出包格式]

表 58

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	命令 ID（0x21）		len(24)		保留	
t				Fid		保留	
facl		保留		blk		cnt	

表 59

字段名	数据长	含义
命令 ID	2（=0x21）	命令代码
len	2（=0x18）	包长（字节数）
t	4	时间
fid	2	文件 ID

facl	2	初始访问控制列表
blk	2	0x0000
cnt	文件 ID	文件长（字节）

[返回包格式]

表 60

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		rlen(8)		保留	

表 61

字段名	数据长	含义
错误代码	2	文件 ID、错误代码
Rlen	2(=0x08)	返回包长（字节数）

[访问控制]

●总可用于发行者模式下（一父系文件夹的发行者的当前芯片 ID 等于构建会话的 SC 的当前芯片 ID）。

●假如父系文件夹的访问控制列表允许，可用于主属模式以及其他模式。

[操作详述]

创建一文件 ID 为 fid 的文件。当已采用指定 fid，一错误发生。

文件的删除

删除文件

[特征概述]

删除文件。

[函数表达式]

ERR edel_fil (SID sid, ETRONID destId，ETRONID srcId,
TIME t, UH len, FID fid, UH*rlen);

sid 会话 ID

destId 作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）

srcId 调用命令的当前芯片服务客户机的当前芯片 ID（当前源芯片 ID）

t 时间

fid 将被删除的文件 ID

len 一输出包长（一字节数）

rlen 一返回包长（一字节数）

[返回值]

<=0 错误代码

[输出包格式]

表 62

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	命令 ID（0x22）		len(16)		保留	
t				fid		保留	

表 63

字段名	数据长	含义
命令 ID	2（=0x22）	命令代码
len	2（=0x10）	包长（字节数）
t	4	时间

fid	2	将被删除的文件 ID

[返回包格式]

表 64

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		rlen(8)		保留	

表 65

字段名	数据长	含义
错误代码	2	文件 ID、错误代码
rlen	2(＝0x08)	返回包长（字节数）

[访问控制]

●为运行“edel_fil”必须满足以下条件。此处须注意包括两种访问控制列表。

（情况 1）主属认证访问情况

1．必须允许命令“edel_fil”作为“主属访问”路由文件夹。也就是说，必须满足“路由文件夹的 ACL2＝＝1”。

2．必须允许命令“edel_fil”作为“主属访问”目标文件。也就是说，必须满足“文件的 ACL8＝＝1”。

（情况 2）发行者认证访问情况

1．必须允许命令“edel_fil”作为“其它访问”路由文件夹。也就是说，必须满足“路由文件夹的 ACL3＝＝1”。

2．目标文件与 ACL 无关（总 OK）。

（情况 3）其它访问情况

1．必须允许命令“edel_fil”作为“其它访问”路由文件夹。也就是

说，必须满足“路由文件夹的 ACL3＝＝1”。

2．必须允许命令“edel_fil”作为“其它访问”目标文件。也就是说，必须满足“文件的 ACL9＝＝1”。

[操作详述]

删除一文件 ID 为 fid 的文件。清除规定文件物理位置的一控制块且释放非易失性存储器，而且存储器中所有数据为设置为“0”或“1”

请求传送一文件

请求文件传送

[特征概述]

将文件内容传送至其它当前芯片信息容器。

[函数表达式]

ERR atra_fil (SID sid, ETRONID destId, ETRONID srcID,
TIME t, UH len, FID fid,
ETRONID targetID, Fid targetFid,
UH*rlen);

sid	会话 ID
destId	作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）
srcId	调用命令的当前芯片服务客户机的当前芯片 ID（当前源芯片 ID）
t	时间
fid	将被创建的文件 ID
targetId	文件的传送目的地的当前芯片 ID
targetFid	在传送目的地将被传送的文件的文件 ID

len 一输出包长（一字节数）

rlen 一返回包长（一字节数）

[返回值]

>0 实际被更新的数据长（普通关闭）

<0 错误代码

[包格式]

表 66

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	命令 ID（0x23）		len(36)		保留	
T				fid		保留	
Targeted							
Targeted							
targetFid	保留						

表 67

字段名	数据长	含义
命令 ID	2（=0x23）	命令代码
len	2（=0x24）	包长（字节数）
t	4	时间
fid	2	文件 ID
Targeted	16	文件将被传送到的目的地的当前芯片 ID
targetFidId	2	传送目的地的文件 ID

[返回包格式]

表 68

D0	D1	D2	D3	D4	D5	D6	D7

SID	序列号	错误代码	rlen(8)	保留

表 69

字段名	数据长	含义
错误代码	2	文件 ID、错误代码
Rlen	2(=0x08)	返回包长（字节数）

[操作详述]

发生系统调用的芯片执行以下一系列操作。

以下当前芯片 API 序列向传送目的芯片发布。

●建立事务会话 eopn_tra

ecfm_tra

●创建文件 ecre_fil

●创建记录 ecre_rec

. .

. .

●提交事务 ecom_tra

假如中期检测到任何异常则终止事物。其间，不映射会话命令除非提交命令正常。

文件数据的加密和解密

加密/解密文件

[特征概述]

存储已加密内容的文件被解密。

[函数表达式]

ERR eenc_fil (SID sid, ETRONID destId, ETRONID srcID,

TIME t, UB len, FID srcFid,

FID destFid, FID keyed, UB*rlen,
UW*currentMoney, UW*payedmoney);

ERR ednc_fil (SID sid, ETRONID destId, ETRONID srcID,
TIME t, UB len, FID srcFid,
FID destFid, FID keyed, UB*rlen,
UW*currentMoney, UW*payedmoney);

sid	会话 ID
destId	作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）
srcId	调用命令的当前芯片服务客户机的当前芯片 ID（当前源芯片 ID）
t	时间

srcFid	存储已加密内容的文件 ID
destFid	存储已解密内容的文件 ID
keyId	存储将被用于解密处理的密钥实体的文件 ID
currentmoney	留在密钥实体的钱款余额
payedmoney	当前结算金额

len	一输出包长（一字节数）
rlen	一返回包长（一字节数）

[返回值]

<=0	错误代码

[包格式]

表 70

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	命令 ID（0x34、35）		len(84＋签名长)		保留	
t				srcFid		保留	
destFid	保留			keyed		保留	

表 71

字段名	数据长	含义
命令 ID	2（0x34、35）	命令代码
len	2	包长（字节数）
t	4	时间
srcFid	1	源文件 ID
destFid1	1	输出目的文件 ID
keyed	1	密钥实体 ID

[返回包格式]

表 72

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		rlen(16)		保留	
currentMoney				PayedMoney			

表 73

字段名	数据长	含义
错误代码	2	文件 ID、错误代码
rlen	2	返回包长（字节数）
currentMoney	4	密钥实体的钱款余额
payedMoney	4	此次结算钱款量

[详细特征]

■edec_fil（）操作

表 74

●已加密内容（srcFid）		●已解密内容（destFid）
（A）内容头	→	（A）内容头
（B）记账头	→	（B）记账头
	→	
加密密钥（1）		00000000000000000000000000
加密密钥（2）		00000000000000000000000000
已加密内容体（1）	→	已解密内容体（1）
…	→	…
已加密内容体（m）	→	已解密内容体（m）
（D）签名尾	→	（D）签名尾

■edec_fil（）操作

表 75

●原始内容（srcFid）		●已加密内容（destFid）
（A）内容头	→	（A）内容头
（B）记账头	→	（B）记账头
	→	
加密密钥（1）	→	00000000000000000000000000
加密密钥（2）	→	00000000000000000000000000
“未加工”内容体（1）	→	已加密内容体（1）
…	→	…
“未加工”内容体（m）	→	已加密内容体（m）
		（D）签名尾

*利用“原始 KEY”进行内容加密。

*利用密钥实体所规定的密钥执行 KEY 加密。

*在完成加密之后对数据签名

*重复四次原始 KEY 所获得的值被用作 MD5 的 HMAC 中 64 位私钥 K

改变文件的访问控制列表

更新文件模式

[特征概述]

文件的访问控制列表被改变。

[函数表达式]

```
ERR eupd_fim (SID sid, ETRONID destId, ETRONID srcId,
                    TIME t, UH len, FID fid, FILACL filacl,
                    UH*rlen);
```

sid	会话 ID
destId	作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）
srcId	调用命令的当前芯片服务客户机的当前芯片 ID（当前源芯片 ID）
t	时间
fid	访问控制列表中所改变的对象的文件 ID
filacl	文件访问列表的更新数据
callerId	调用命令的当前芯片服务客户机的当前芯片 ID
len	一输出包长（一字节数）

rlen 一返回包长（一字节数）

[返回值]

<=0 错误代码

[输出包格式]

表 76

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	命令 ID（0x27）		len(20)		保留	
T				fid		保留	
filacl	保留						

表 77

字段名	数据长	含义
命令 ID	2（=0x27）	命令代码
len	2（=0x14）	包长（字节数）
t	4	时间
fid	2	文件 ID
filacl	2	文件访问列表

[返回包格式]

表 78

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		rlen(8)		保留	

表 79

字段名	数据长	含义
错误代码	2	文件 ID、错误代码
rlen	2（＝0x08）	返回包长（字节数）

[访问控制]

●在关于对象文件的发行者模式总是可用的（一父系文件夹的发行者的当前芯片 ID 等于构建会话的 SC 的当前芯片 ID）

●假如对象文件的访问控制列表允许，则在主属模式以及其它模式总是可用的

[特征概述]

文件的访问控制列表改变为一特定值。

所定义的文件列表的获取

列出文件 ID

[特征概述]

获取所定义的文件列表

[函数表达式]

```
ERR elst_fid (SID sid, ETRONID destId, ETRONID srcId,
                    TIME t, UH len, FID fid, UH*rlen,
                    UB**fileCtrlBlk);
```

sid 会话 ID

destId 作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）

srcId 调用命令的当前芯片服务客户机的当前芯片 ID

（当前源芯片 ID）

t 时间

fid 对象文件 ID

fileCtrlBlk 对象文件的控制块

len 一输出包长（一字节数）

rlen 一返回包长（一字节数）

[返回值]

<=0 错误代码

[输出包格式]

表 80

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号 NO.	命令 ID (0x2F)		len （16）		保留	
T				fid		保留	

表 81

字段名	数据长	含义
命令 ID	2（=0x2F）	命令代码
len	2（16）	包长（字节数）
t	4	时间
fid	2	文件 ID

[返回包格式]

表 82

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		rlen		保留	
FileCtrlBlk							
…							
FileCtrlBlk							

表 83

字段名	数据长	含义
错误代码	2	文件 ID、错误代码
rlen	2	返回包长（字节数）
fileCtrlBlk	*rlen-8	对象文件的控制块

[访问控制]

●在关于对象文件的发行者模式总是可用的（一父系文件夹的发行者的当前芯片 ID 等于构建会话的 SC 的当前芯片 ID）

●假如对象文件的访问控制列表允许，则在主属模式以及其它模式总是可用的

10.4 记录管理命令组

更新记录数据

更新记录

[特征概述]

一记录内容被改变。

[函数表达式]

ERR eupd_rec (SID sid, ETRONID destId, ETRONID srcId,
TIME t, UH len, FID fid, UH blk,
UH cnt, UB*data, UH*rlen);

sid	会话 ID
destId	作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）
srcId	调用命令的当前芯片服务客户机的当前芯片 ID（当前源芯片 ID）
t	时间
fid	将被改变的文件 ID
blk	将被改变的记录号（此处固定为 1）
cnt	将被改变的一记录数据长（一字节数）
data	将被改变的数据内容（cnt 字节）
len	一输出包长（一字节数）
rlen	一返回包长（一字节数）

[返回值]

>0	实际更新的数据长（普通关闭）
<0	错误代码

[包格式]

表 84

D0	D1	D2	D3	D4	D5	D6	D7

SID	序列号	命令 ID（0x33）	len	保留
T			fid	保留
blk	cnt		data	
Data				
…				
Data				

表 85

字段名	数据长	含义
命令 ID	2（0x33）	命令代码
len	2	包长（字节数）
t	4	时间
fid	2	文件 ID
blk	2	所改变的记录 ID（此处固定为 1）
cnt	2	数据长（字节数）
data	Cnt	将被写的数据

[返回包格式]

表 86

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		rlen(8)		保留	

表 87

字段名	数据长	含义
错误代码	2	文件 ID、错误代码
rlen	2	返回包长（字节数）

[访问控制]

●在关于对象文件的发行者模式总是可用的（一父系文件夹的发行者的当前芯片 ID 等于构建会话的 SC 的当前芯片 ID）

●假如对象记录的访问控制列表允许，则在主属模式以及其它模式总是可用的

[操作详述]

发布一用于写来自 clk 记录开始处的 cnt 字节数据的请求。依据硬件条件为 cnt 提供一上限。实际所写字节数被返回至 rCnt。

从记录中读数据

读记录

[特征概述]

记录内容被读出。

[函数表达式]

```
ERR erea_rec (SID sid, ETRONID destId, ETRONID srcId,
                    TIME t, UH len, FID fid, UH blk,
                    UH cnt, UH*rlen, UH*rCnt, UB*rData,);
```

sid	会话 ID
destId	作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）
srcId	调用命令的当前芯片服务客户机的当前芯片 ID（当前源芯片 ID）
t	时间
fid	将被读出的文件 ID

blk 将被读出的记录 ID

cnt 将被读出的一数据长（一字节数）

rCnt 实际读出的数据长（一字节数）

rData 读出的数据（rCnt 字节）

len 一输出包长（一字节数）

rlen 一返回包长（一字节数）

[返回值]

<=0 错误代码

[包格式]

表 88

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	命令 ID（0x33）		len		保留	
T				fid		保留	
blk		cnt					

表 89

字段名	数据长	含义
命令 ID	2（0x34）	命令代码
len	2	包长（字节数）
t	4	时间
fid	2	文件 ID
blk	2	所改变的记录 ID
cnt	2	数据长（字节数）

[返回包格式]

表 90

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		rlen(8)		保留	
rCnt		保留		rData			
rData							
...							
rData							

表 91

字段名	数据长	含义
错误代码	2	读出数据长（字节数）、错误代码
rCnt	2	实际读出的数据长（字节数）
rData	rCnt	读出数据

[访问控制]

●在关于对象文件的发行者模式总是可用的（一父系文件夹的发行者的当前芯片 ID 等于构建会话的 SC 的当前芯片 ID）

●假如对象记录的访问控制列表允许，则在主属模式以及其它模式总是可用的

[操作详述]

发布一用于读来自 blk 记录开始处的 cnt 字节数据的请求。依据硬件条件为 cnt 提供一上限。实际所读字节数被返回至 rCnt。

通常在建立会话时发布并运行此命令。然而，未建立会话的当前芯片服务客户机也可能通过专门的“touch&go”应用且 sid＝0 来调用此命

令。在此情况下，记录的访问控制列表允许其它权利，读出之后运行此命令。

10.5 密钥实体管理命令组

[用于实现的受限项]

此系统中的密钥具有以下限制。

●可用文件数上限 10
●密钥实体 ID 0 至 9
●密钥实体大小上限 256（字节）

一密钥实体的创建
创建密钥实体

[特征概述]

[函数表达式]

```
ERR ecre_key (SID sid, ETRONID destId, ETRONID srcId,
                    TIME t, UH len, UH kid,
                    UH keyAlgorithm, UB*key, UH bankId,
                    KEYACL acl, UL initMoney Val,
                    UH*rlen);
```

sid	会话 ID
destId	作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）
srcId	调用命令的当前芯片服务客户机的当前芯片 ID（当

前源芯片 ID）

t 时间

kid 密钥实体 ID

KeyAlgorithm 加密算法说明符

key 一加密密钥

bankID 一银行 ID

initMoney Val 一钱款初始量

acl 一初始访问控制列表值（未用于第一版）

len 一输出包长（一字节数）

rlen 一返回包长（一字节数）

[返回值]

<=0 错误代码

[包格式]

表 92

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	命令 ID（0x61）		len		保留	
T				kid		keyAlgorithm	
KEY(1)							
KEY(2)							
BankID		acl		initMoney Val			

表 93

字段名	数据长	含义

命令 ID	2（0x61）	命令代码
Len	2	包长（字节数）
T	4	时间
Kid	2	密钥实体 ID
keyAlgorithm	2	加密算法说明符
Key	16	加密密钥
BankID	2	银行 ID
Acl	2	初始访问控制列表值（未用于第一版）
initMoneyVal	4	初始钱款量

[返回包格式]

表 94

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		rlen(8)		保留	

表 95

字段名	数据长	含义
错误代码	2	错误代码
rlen	2	返回包长（字节数）

密钥实体的删除

删除密钥实体

[特征概述]

[函数表达式]

ERR edel_key (SID sid,ETRONID destId，ETRONID srcId,

TIME t, UH len, UH kid, UH*bankId,
UL*remainingMoney, UH*rsize,
UB**booklist, UH*rlen);

sid	会话 ID
destId	作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）
srcId	调用命令的当前芯片服务客户机的当前芯片 ID（当前源芯片 ID）
t	时间

kid	将被删除的密钥实体 ID
bankID	银行 ID
remainingMoney	钱款的剩余量
rsize	分配列表长
bookList	分配列表

len	一输出包长（一字节数）
rlen	一返回包长（一字节数）

[返回值]
<=0 错误代码

[包格式]
表 96

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	命令 ID（0x62）		Len		保留	
T				Kid		保留	

表 97

字段名	数据长	含义
命令 ID	2（0x62）	命令代码
len	2	包长（字节数）
t	4	时间
kid	2	密钥实体 ID

[返回包格式]

表 98

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		rlen(16)		保留	
银行 ID		保留		在消费后的余额 currenMoney			

表 99

字段名	数据长	含义
错误代码	2	文件 ID、错误代码
rlen	2	返回包长（字节数）
bankId	2	银行 ID
currentMoney	4	当前钱款余额

更新密钥实体

更新密钥实体

[特征概述]

更新指定密钥实体的钱款量。

[函数表达式]

ERR eupd_key (SID sid, ETRONID destId, ETRONID srcId,
TIME t, UH len, UH kid, UL addMoney,
UH*rlen, UL*currentMoney);

sid	会话 ID
destId	作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）
srcId	调用命令的当前芯片服务客户机的当前芯片 ID（当前源芯片 ID）
t	时间
kid	将被更新的密钥实体 ID
addMoney	将被增加的钱款量
currentMoney	增加后的钱款量
len	一输出包长（一字节数）
rlen	一返回包长（一字节数）

[返回值]

<=0 错误代码

[包格式]

表 100

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	命令 ID（0x63）		len		保留	
T				kid		保留	
AddMoney							

表 101

字段名	数据长	含义
命令 ID	2（0x63）	命令代码
Len	2	包长（字节数）
T	4	时间
Kid	2	将被更新的密钥实体 ID
AddMoney	4	将增加的钱款量

[返回包格式]

表 102

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		rlen(12)		保留	
增加后的 currentMoney 余额							

表 103

字段名	数据长	含义
错误代码	2	错误代码
Rlen	2（12）	返回包长（字节数）
currentMoney	2	增加后的钱款量

读密钥实体的信息

读密钥实体

[特征概述]

从特定密钥实体中读出钱款量信息。

[函数表达式]

ERR erea_key (SID sid, ETRONID destId, ETRONID srcId,
TIME t, UH len, UH kid, UH*rlen,
UL*currentMoney);

sid	会话 ID
destId	作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）
srcId	调用命令的当前芯片服务客户机的当前芯片 ID（当前源芯片 ID）
t	时间
kid	作为用于读钱款量信息的对象的密钥实体 ID
currentMoney	当前钱款量
len	一输出包长（一字节数）
rlen	一返回包长（一字节数）

[返回值]

<=0	错误代码

[包格式]

表 104

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	命令 ID（0x64）		len		保留	
T				kid		保留	

表 105

字段名	数据长	含义

命令 ID	2（0x63）	命令代码
len	2	包长（字节数）
t	4	时间
kid	2	将被读的密钥实体 ID

[返回包格式]

表 106

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		rlen(12)		保留	
currentMoney							

表 107

字段名	数据长	含义
错误代码	2	错误代码
Rlen	2(=12)	返回包长（字节数）
CurrentMoney	4	当前钱款量

更新密钥实体的访问控制列表

更新密钥模式

[特征概述]

密钥实体的访问控制列表被更新。

*第一版中未下载

[函数表达式]

```
ERR eupd_key (SID sid, ETRONID destId, ETRONID srcId,
                    TIME t, UH len, UH kid, KEYACL keyacl,
```

UH*rlen);

sid	会话 ID
destId	作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）
srcId	调用命令的当前芯片服务客户机的当前芯片 ID（当前源芯片 ID）
t	时间
kid	访问控制列表中所改变的对象的密钥实体 ID
keyacl	将被更新的密钥实体访问控制列表
len	一输出包长（一字节数）
rlen	一返回包长（一字节数）

[返回值]

<=0	错误代码

[输出包格式]

表 108

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	命令 ID（0x65）		len		保留	
T				kid		保留	

表 109

字段名	数据长	含义
命令 ID	2（0x65）	命令代码
Len	2	包长（字节数）

T	4	时间
Kid	2	将被更新的密钥实体 ID
keyacl	2	密钥实体访问列表

[返回包格式]

表 110

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		rlen(8)		保留	
currentMoney							

表 111

字段名	数据长	含义
错误代码	2	文件 ID、错误代码
Rlen	2（=8）	返回包长（字节数）
currentMoney	4	增加后的钱款量

所定义的密钥实体列表的获取

列出密钥实体 ID

[特征概述]

获取所定义的密钥实体列表

[函数表达式]

```
ERR elst_kid (SID sid, ETRONID destId, ETRONID srcId,
                    TIME t, UH len, KID kid, UH*rlen,
                    UB**keyCtrlBlk);
```

sid	会话 ID
destId	作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）
srcId	调用命令的当前芯片服务客户机的当前芯片 ID（当前源芯片 ID）
t	时间
kid	对象密钥实体 ID
keyCtrlBlk	对象密钥实体的控制块
len	一输出包长（一字节数）
rlen	一返回包长（一字节数）

[返回值]

<=0 错误代码

[输出包格式]

表 112

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号 NO.	命令 ID(0x6F)		len（16）		保留	
T				kid		保留	

表 113

字段名	数据长	含义
命令 ID	2（=0x6F）	命令代码
len	2（=16）	包长（字节数）
t	4	时间
kid	2	密钥实体 ID

[返回包格式]

表 114

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		rlen		保留	
keyCtrlBlk							
…							
keyCtrlBlk							

表 115

字段名	数据长	含义
错误代码	2	错误代码
rlen	2(=8)	返回包长（字节数）
keyCtrlBlk	*rlen-8	对象文件的控制块

[访问控制]

●在关于对象密钥实体的发行者模式总是可用的（一父系文件夹的发行者的当前芯片 ID 等于构建会话的 SC 的当前芯片 ID）

●假如对象密钥实体的访问控制列表允许，则在主属模式以及其它模式总是可用的

10.6 认证辅助管理命令组

检测 eTP 证书

确认证书

[函数表达式]

ERR ecfm_cer (SID sid, ETRONID destId，ETRONID srcId,

TIME t, UH len, UB checkMode,
UH serial, UH caId, UB*rlen, UB*crl);

sid	会话 ID
destId	作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）
srcId	调用命令的当前芯片服务客户机的当前芯片 ID（当前源芯片 ID）
t	时间
checkMode	将被无效的列表的检测模式
serial	将被确认的证书的证书号
caId	将被确认的证书的发行者 ID
crl	一无效列表（当前芯片格式）
len	一输出包长（一字节数）
rlen	一返回包长（一字节数）

[checkMode]

0x00	只返回检测
0x01	获取相应无效列表

[返回值]

>0	会话 ID（普通关闭）
<0	错误代码

[输出包格式]

表 116

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	命令 ID(0x71)		Len		保留	
T				checkMode		保留	
Serial	CA ID						

表 113

字段名	数据长	含义
命令 ID	2（=0x71）	命令代码
len	2	包长（字节数）
t	4	时间
checkMode	2	检测模式
certificateId	16	将被检测的证书 ID

[返回包格式]

表 118

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		rlen（8）		保留	
Crl							
…							
Crl							

表 119

字段名	数据长	含义
错误代码	2	错误代码
Rlen	2	无效列表长
Crl	rlen-8	无效列表

■5.3 准当前芯片 API 命令定义

如下所述的 API 命令具有类似于当前芯片 API 命令的格式。但是，以下命令不仅由当前芯片服务客户机发布还可由其它实体发布。这些命令主要以取决于硬件的管理和操作为中心，且为用于支持低层通信的接口。这些命令可由芯片读取/写入单元发布或若生产则由一计算机发布。

轮询一芯片
轮询芯片

[特征概述]
读出当前芯片状态（相关实现、相关操作）
[函数表达式]
ERR epol_cer (ETRONID* destId，ETRONID srcId,TIME t);

destId	轮询返回的芯片 ID
srcId	调用命令的当前芯片服务客户机的当前芯片 ID
t	时间

[返回值]

<=0	错误代码

[包格式]

表 120

D0	D1	D2	D3
保留（0x00）	序列（0x00）	Com. ID（0x41）	保留

表 121

字段名	数据长	含义
命令 ID	1（0x41）	命令代码

[返回包格式]

表 122

D0	D1	D2	D3	D4	D5	D6	D7
SID（0x00）	序列(0x00)	错误代码	rlen(0x08)	cardVersion		保留	

表 123

字段名	数据长	含义
错误代码	1	错误代码
CardVersion	2	卡版本
rlen	1	数据包长（=0x08）

*eared 和 clientId 均附于一芯片头并因此不包含在主体中。

[访问控制]

此 API 总是可用的。

在会话过程中不总存在。

[详细特征]

芯片的当前芯片 ID 被存储并返回到“epol_car”的返回包的路由头的 srcID 位置。因此，任何人可利用此用于在起始处获得当前芯片 ID 的 API。

任何人，即使不在该会话中，都能发出该命令。

芯片的初始化

初始化芯片

[特征概述]

创建芯片管理部分以初始化当前芯片。

[函数表达式]

```
ERR eini_car (ETRONID destId，ETRONID srcId, TIME t,
                    UB initKey1[8],UB initKey2[8],
                    UB initPasswd[8],UB fileNum,
                    UB keyObjLen,UB swapLen,CACL cacl,
                    UB rsasecretKey[128],UH len,
                    UH *rlen);
```

destId	作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）
srcId	全“1”
t	时间

initKey1	用于认证的一公钥（1）
initKey2	用于认证的一公钥（2）
initPasswd	一初始密码
fileNum	芯片上可创建的文件数上限
keyObjNum	密钥实体数上限
swapLen	非易失性存储器中交换区域大小
cacl	芯片访问控制列表
rsaSecretKey	用于固有卡认证（RSA）的私钥

len	一输出包长（一字节数）
rlen	一返回包长（一字节数）

[返回值]

<=0 错误代码

[包格式]

表 124

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	命令 ID(0x51)		len（176）		保留	
T				保留			
用于认证的公钥（1）							
用于认证的公钥（2）							
密码							
Cacl	FileNum	KeyobjNum	swapLen	保留	版本		
固有（RSA）私钥（1）							
…							
固有（RSA）私钥（16）							

表 125

字段名	数据长	含义
命令 ID	2（0x51）	命令代码
len	2（176）	包长（字节数）
t	4	时间
initKey1	8	用于认证的公钥（1）
initKey2	8	用于认证的公钥（2）
initPasswd	8	认证密码
fileNum	1	文件数上限（50）

keyObjNum	1	密钥实体数上限（10）
swapLen	1	交换区域
Cacl	1	初始卡访问控制列表
rsaAecretKey	128	固有认证 RSA 密钥（1024 位）

[返回包格式]

表 126

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		rlen（8）		保留	

表 127

字段名	数据长	含义
错误代码	2	错误代码
rlen	2	返回包长

[详细特征]

依据一给定变元初始化一系统控制块。

所有其它存储区域被清除并被设置成“0”。

[访问控制]

仅在当前芯片的当前芯片 ID＝0xff...ff(全“1”)时，此命令被接收

转换固有认证公钥证书

更新我的证书

[特征概述]

更新当前芯片的固有证书。

[函数表达式]

ERR eupd_cer (ETRONID destId，ETRONID srcId, TIME t,
UB*certificate, UH len, UH *rlen);

sid	会话 ID
destId	作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）
srcId	调用命令的包含当前芯片服务客户机的当前芯片 ID（当前源芯片 ID）
t	时间
certificate	证书
len	一输出包长（一字节数）
rlen	一返回包长（一字节数）

[返回值]

<=0	错误代码

[包格式]

表 128

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	命令 ID(0x52)		Len（320）		保留	
t				保留			

证书（1）
…
证书（38）

表 129

字段名	数据长	含义
命令 ID	2（0x52）	命令代码
Len	2（304）	包长
T	4	时间
Certificate	304	证书

[返回包格式]

表 130

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		rlen（8）		保留	

[返回包格式（例）]

表 131

字段名	数据长	含义
错误代码	2	错误代码、会话 ID
rlen	2	返回包长（字节数）

转换一 CA 站公钥

更新 CA 公钥

[特征概述]

交换 CA 站的公钥。

[函数表达式]

ERR eupd_cpk (ETRONID destId，ETRONID srcId, TIME t,
UB*caPublicKey, UH len, UH *rlen);

sid	会话 ID
destId	作为一命令对象的当前芯片的芯片 ID（当前目的芯片 ID）
srcId	调用命令的包含当前芯片服务客户机的当前芯片 ID（当前源芯片 ID）
t	时间
caPublicKey	CA 站的公钥
len	一输出包长（一字节数）
rlen	一返回包长（一字节数）

[返回值]

<=0	错误代码

[包格式]

表 132

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	命令 ID(0x53)		len（144）		保留	
t				保留			
CA 站公钥（1）							

…
CA 站公钥（16）

表 133

字段名	数据长	含义
命令 ID	2（0x53）	命令代码
len	2（144）	包长
t	4	时间
Certificate	128	RSA 证书（1024 位）

[返回包格式]

表 134

D0	D1	D2	D3	D4	D5	D6	D7
SID	序列号	错误代码		rlen（8）		保留	

[返回包格式（例）]

表 135

字段名	数据长	含义
错误代码	2	错误代码、会话 ID
Rlen	2	返回包长（字节数）

加密实现说明

■DES

从 64 位密钥创建 56 位 DES 密钥的方法

指定每字节的最低位为奇偶校验位，如下所示（一芯片仅忽略奇偶校验位而不检测它）。通过增加 8 字节的以上 7 位而获得的 56 位被用作 DES。

●密钥（1）

b7b6b5b4b3b2b1b0 b7b6b5b4b3b2b1 b0b7b6b5b4b3b2b1b0

□□□□□□□■ □□□□□□□ ■□□□□□□□■

b7b6b5b4b3b2b1b0

□□□□□□□■

●密钥（2）

b7b6b5b4b3b2b1b0 b7b6b5b4b3b2b1 b0b7b6b5b4b3b2b1b0

□□□□□□□■ □□□□□□□ ■□□□□□□□■

b7b6b5b4b3b2b1b0

□□□□□□□■

■RSA

一 RSA 密钥的数据格式

密钥如下所示。

d＝1024 位（128 字节）

n＝1024 位（128 字节）

e＝固定为 0x003（1 字节）

密钥 sk＝d（128 字节）

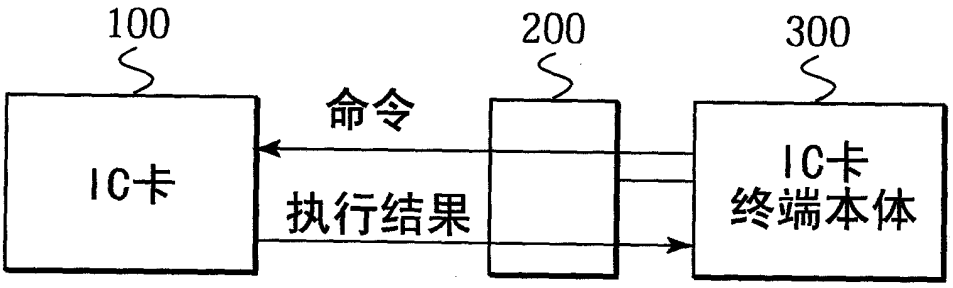
公钥 pk＝e（1 字节）|e（1 字节）|n（128 字节）的长度

如上所述，依据本发明，本发明的自主型 IC 卡自主识别将经由一 IC

卡终端连接的通信设备并直接与此通信设备通信。因此，有可能保证准确信息的安全通信。而且，当本发明的自主型 IC 卡包括一用于相互执行认证处理和加密解密包含与通信设备通信的信息的加密处理单元时，有可能判断适当的通信设备并有效减少经由中间设备盗窃数据内容或伪造数据的危险。当数据与诸如一电子票的值信息相关时以上减少危险尤其有意义。而且，在依据所标识的通信设备种类的大量认证处理和大量加密处理当中，加密处理单元选择并执行适当认证处理和加密处理，此时有可能执行适合于所标识的通信设备的处理。

可理解地，本发明的修改和适应性改变对于本领域普通技术人员来说是显而易见的且所附权利要求的范围包含了此明显的修改和改变。

图1



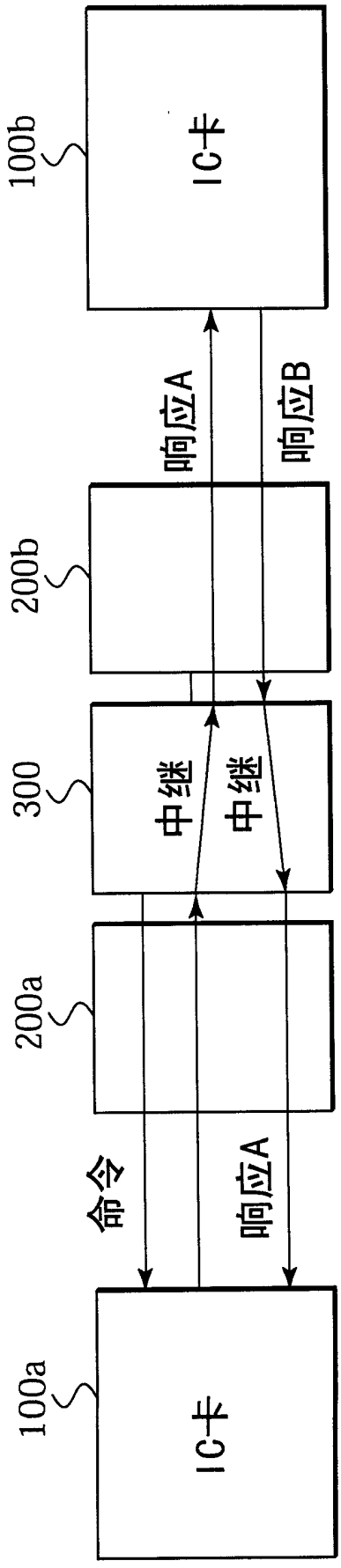


图2

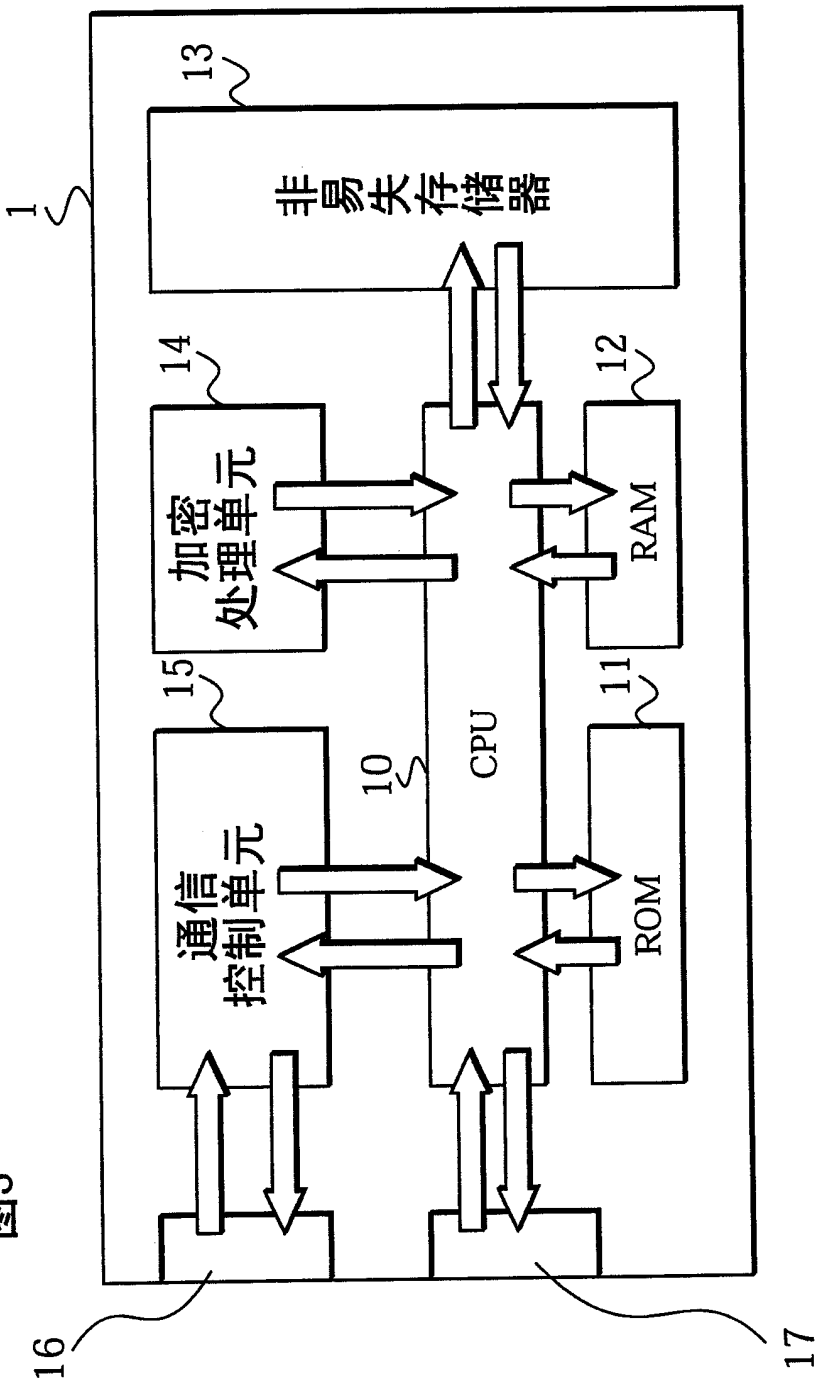


图3

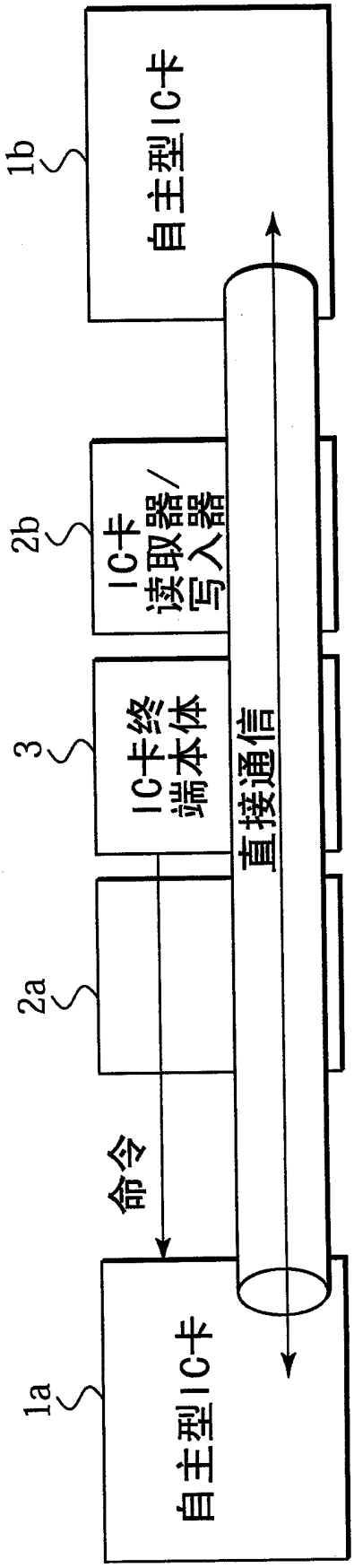


图4

图5

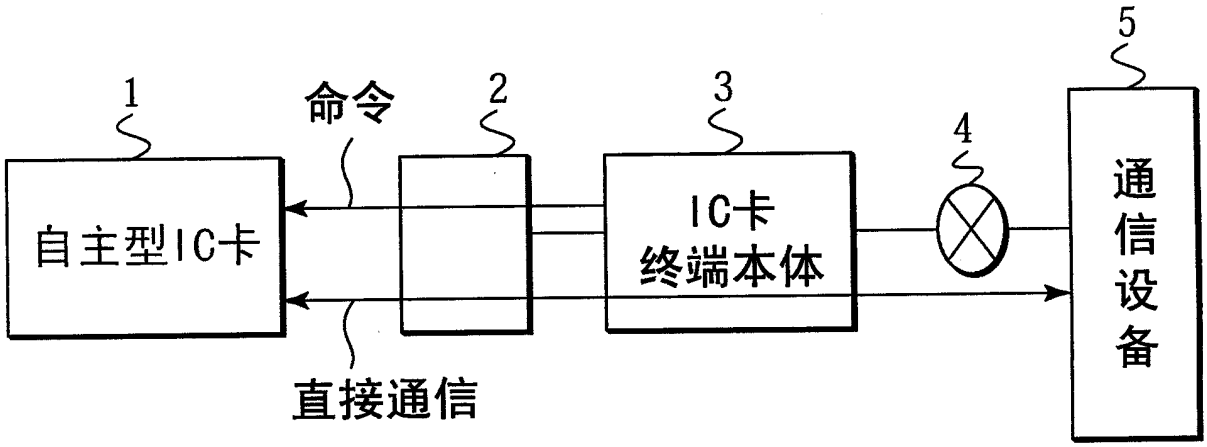


图 6

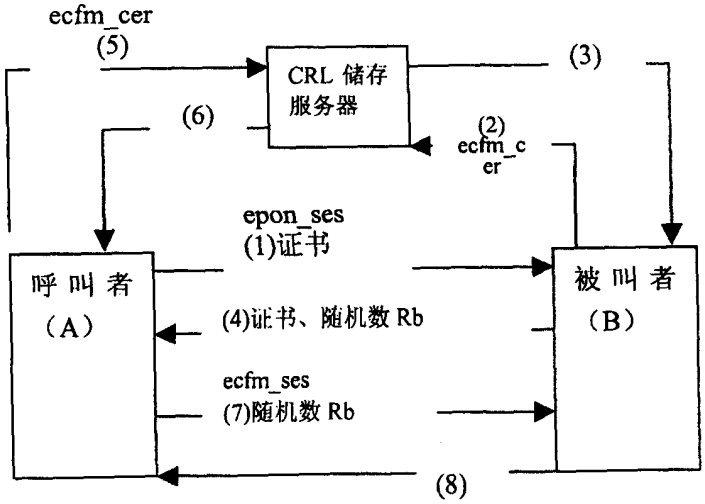
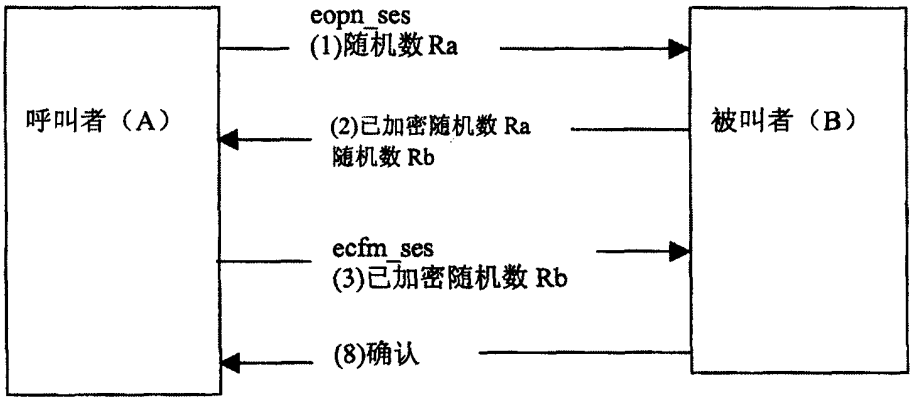

ecfm_cer
(5)
CRL 储存
服务器
(3)
(6)
(2)
ecfm_c
er
epon_ses
(1)证书
呼 叫 者
（A）
被 叫 者
（B）
(4)证书、随机数 Rb
ecfm_ses
(7)随机数 Rb
(8)

图 7


呼叫者（A）
eopn_ses
(1)随机数 Ra
(2)已加密随机数 Ra
随机数 Rb
ecfm_ses
(3)已加密随机数 Rb
(8)确认
被叫者（B）