



(12) 发明专利

(10) 授权公告号 CN 102663473 B

(45) 授权公告日 2014. 10. 08

(21) 申请号 201210123334. 9

(22) 申请日 2012. 04. 25

(73) 专利权人 山东神思电子技术股份有限公司
地址 250101 山东省济南市历下区(高新区)
舜华西路 699 号

(72) 发明人 郭延海 孙涛

(74) 专利代理机构 济南泉城专利商标事务所
37218

代理人 丁修亭

(51) Int. Cl.

G06K 19/073 (2006. 01)

(56) 对比文件

CN 101042737 A, 2007. 09. 26, 全文.

WO 2011/041447 A2, 2011. 04. 07, 全文.

CN 101950342 A, 2011. 01. 19, 说明书第
0029-0065 段, 附图 1-3.

CN 102073558 A, 2011. 05. 25, 全文.

邓赟. 智能卡操作系统(COS)安全管理研

究.《硅谷》. 中国科技新闻学会, 2010, (第2期),
第 50-51 页.

李蔚. 金融智能卡系统安全机制的研究.
《中国优秀硕士学位论文全文数据库信息科技辑》.
2009, (第4期), 第一张第 1.4.4 节、第三章.

审查员 杨越松

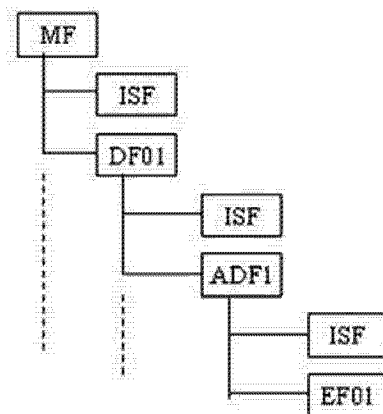
权利要求书1页 说明书6页 附图2页

(54) 发明名称

智能卡 COS 安全机制的实现方法

(57) 摘要

本发明公开了一种智能卡 COS 安全机制的实现方法, 智能卡被读写终端设备识别时, 通过外部认证与核实的结果对所述安全控制字的相应位按照预定的方法置位; 读取被置位的所述安全控制字, 依据所述预定的方法解析该安全控制字, 确定安全属性的使用逻辑; 验证因状态转移产生更新的安全状态是否满足上述逻辑条件下的安全属性, 并在满足对应的安全属性时, 执行被允许的操作; 以及在智能卡与读写终端设备预定的操作完成后, 所述安全控制字复位。依据本发明的智能卡 COS 安全机制的实现方法可以简化安全机制的实现。



1. 一种智能卡 COS 安全机制的实现方法,其特征在于,在安全属性中定义以表示安全属性使用逻辑的安全控制字初始化为相应用户的权限是对文件的访问被禁止;

从而,智能卡被读写终端设备识别时,通过外部认证与核实的结果对所述安全控制字的相应位按照预定的方法置位;

读取被置位的所述安全控制字,依据所述预定的方法解析该安全控制字,确定安全属性的使用逻辑;

验证因状态转移产生更新的安全状态是否满足上述逻辑条件下的安全属性,并在满足对应的安全属性时,执行被允许的操作;以及

在智能卡与读写终端设备预定的操作完成后,所述安全控制字复位。

2. 根据权利要求1所述的智能卡 COS 安全机制的实现方法,其特征在于,当前目录的安全属性及其父目录的安全属性被分组,一组表示当前目录的安全属性,另一组表示其父目录的安全属性,从而,安全机制在安全属性的管理时,安全控制字至少含有一位表示当前目录安全属性的屏蔽与否,并含有一位表示其父目录安全属性的屏蔽与否。

3. 根据权利要求2所述的智能卡 COS 安全机制的实现方法,其特征在于,安全控制字含有一表示当前目录安全属性组内安全属性使用逻辑的位,该位的两个状态对应为逻辑与和逻辑或;

相应地,安全控制字还含有一表示当前目录的父目录安全属性组内安全属性使用逻辑的位,该位的两个状态对应为逻辑与和逻辑或。

4. 根据权利要求3所述的智能卡 COS 安全机制的实现方法,其特征在于,安全控制字还含有一在当前目录安全属性及其父目录安全属性都没有被屏蔽的情况下表示两组内所有安全属性使用逻辑的位,该位的两个状态对应位逻辑与和逻辑或。

5. 根据权利要求2至4任一所述的智能卡 COS 安全机制的实现方法,其特征在于,使用位逻辑表示安全属性,对应为位逻辑的两个状态表示相应安全属性需要满足与否。

6. 根据权利要求5所述的智能卡 COS 安全机制的实现方法,其特征在于,当前目录的安全属性及其父目录的安全属性各有8个。

智能卡 COS 安全机制的实现方法

技术领域

[0001] 本发明涉及一种智能卡 COS 安全机制的实现方法,其中,COS 为 Chip Operating System 的缩略语,即片内操作系统。

背景技术

[0002] 智能卡技术广泛应用于政府办公、公共事业、通讯、金融、社保、交通、医疗、教育、娱乐、信息安全等社会各个行业。智能卡在极大的方便了老百姓生活的同时,安全机制的制约是需要考虑的问题。

[0003] 智能卡也叫 CPU 卡,是一种集成了 CPU、存储单元、加密协处理器的集成电路卡。智能卡 CPU 内需要加载 COS 使智能卡实现“智能”。COS 一般都有自己的安全体系,其安全性能通常是衡量 COS 的重要技术指标,为此需要进一步的了解智能卡 COS 的体系结构。

[0004] 智能卡 COS 用于实现控制智能卡与外部的信息交换,管理智能卡内部的存储器,并在智能卡内部进行各种命令的出理。依据这些应用,参见说明书附图 1,符合智能卡规范和国际标准的智能卡 COS 由传输管理、命令管理(又叫命令解释)、文件体系(又叫文件管理)、安全体系四个部分组成。

[0005] 其中,传输管理,即图 1 中的通信管理模块,负责智能卡和接口设备之间的数据通信,如对由终端读写设备发出的命令进行接收,同时,并对该命令的响应按照通信传输协议的格式发送出去,详见国际标准 ISO/IEC7816-4。

[0006] 命令管理,及图 1 中的命令解析模块,是对外部输入的每条命令做语法分析,分析和检查命令参数是否正确,然后根据命令参数的含义执行相应的功能模块,并返回响应,完成与读写终端设备的信息交换。

[0007] 文件管理,也就是文件体系,用于控制对文件的操作和访问。参见说明书附图 2,文件系统为树型结构,并在文件系统中定义了三种文件类型,作为根结点的主文件(MF, Master File)、作为中间节点的目录文件(DF, Deicated File)和作为页节点的基本文件(EF, Elementary File)。

[0008] 其中 MF 是整个文件系统的根目录,即上面所述的树型结构的根节点。DF 是一种类似于中间目录的专用文件,可以包含其他的 DF 和 EF。而 EF 则是集成电路卡文件系统的基本文件,是真正用来存储数据的文件。对于 EF 来说,其文件体包含用于存放用户数据的数据文件和用于设置访问权限的密钥文件(ISF, Internal Secret Files)。

[0009] 另外,在智能卡的文件结构中,MF 只能有一个,用户不能在文件的存取过程中越层存取。若想读写子 DF 下的 EF,必须经过高层的文件层次中的某一个 DF 的大小申请文件时预定并且不能修改。

[0010] 如前所述,安全体系是决定智能卡迅猛发展的核心要素,在于通过其能够为用户提供一个较高的安全保证。安全体系包括了文件安全属性、系统安全状态、安全机制、加解密算法。

[0011] 安全机制用于安全状态与安全属性的联系,可以进一步理解为实现安全状态的转

移所使用的方法或手段。从而,COS 安全体系的基本工作原理可以这样认为,一种安全状态经过安全机制的一些转移方法或手段就可以转移到另一种状态,进而将被转移到的另一种状态与相应的某个安全属性相比较,若一致,就执行该安全属性对应的命令,即安全机制是确定是否允许当前操作的机制。

[0012] 那么依据安全机制的描述,对应地,安全状态表示了智能卡在当前情况下所处的状态,这种状态会在智能卡进行完复位应答或者在他处理完某条上述命令之后得到,所以,安全状态表示了智能卡在整個过程中处于了某种状态。从而,利用智能卡在当前已满足条件的集合可以表示当前的安全状态,各种安全状态和他的转换条件组合在一起就构成了状态机,而状态机就构成了对安全机制的约束。

[0013] 对于安全属性,则是对智能卡的数据对象的访问控制,为关联文件结构与安全机制的要素,那么对数据对象的访问主要涉及两方面的内容,即文件的安全属性和操作命令。其中文件的安全属性包括了两方面的内容:允许进行操作的类型和进行操作需要满足的安全状态。文件的安全属性在文件创建时被定义,包含在文件描述块中,由文件系统管理和维护。操作对应的命令的安全属性是由具体的命令进行定义的,它又进一步包括两方面的内容,即命令全部报文的安全控制和命令数据的安全控制。

[0014] 承上所述,安全机制主要包括三个方面的功能,分别是数据加密与解密、鉴别与核实,以及文件访问的安全控制。其中,数据的加密/解密贯穿安全体系的整个过程;鉴别与核实就是对身份的认证;文件访问的安全控制与文件管理关联。

[0015] 传统智能卡的系统安全状态是定义的具体数值,文件安全属性则是以具体的数值定义的连续的区间。随着智能卡应用的广泛推广,社会对智能卡的要求越来越具体、越来越全面。随着国家金卡工程的深入发展,现在智能卡普遍实现了多应用设计。智能卡的多应用扩展在方便用户和满足社会需求的同时,这种线性、区间性安全机制越来越不能满足用户对智能卡个性化权限控制的需求。

[0016] 在多部门或多应用需要对于智能卡某些文件进行相应权限的操作时,各部门之间的密钥是独立的。比如一个应用下的几个文件对应几个部门,本部门具有本部门文件的读写擦除权限,具有其他部门文件的读权限,上级部门具有所有文件的读写修改权限。在这种情况下,需要定义的安全属性是离散的,传统智能卡的安全机制实现起来变得越来越困难。

发明内容

[0017] 为此,本发明的目的在于提供一种利用安全控制字控制各安全属性关系实现个性化文件访问权限的智能卡 COS 安全机制,简化安全机制的实现。

[0018] 本发明所采用的技术方案为:

[0019] 一种智能卡 COS 安全机制的实现方法,在安全属性中定义以表示安全属性使用逻辑的安全控制字初始化为相应用户的权限是对文件的访问被禁止;

[0020] 从而,智能卡被读写终端设备识别时,通过外部认证与核实的结果对所述安全控制字的相应位按照预定的方法置位;

[0021] 读取被置位的所述安全控制字,依据所述预定的方法解析该安全控制字,确定安全属性的使用逻辑;

[0022] 验证因状态转移产生更新的安全状态是否满足上述逻辑条件下的安全属性,并在

满足对应的安全属性时,执行被允许的操作;以及

[0023] 在智能卡与读写终端设备预定的操作完成后,所述安全控制字复位。

[0024] 依据本发明的上述方法,使用安全控制字描述安全属性,确定安全属性的使用方式,从而,安全属性的使用逻辑具有比较大的开放性,使得安全安全机制的实现变得非常容易。同时,控制字被定义和使用对资源的占用比较少,同时控制逻辑相对简单,可以简化安全机制的实现。

[0025] 在进一步提供的方案中以位逻辑实现的安全状态和安全属性,进一步简化安全机制的实现。

[0026] 因此,作为进一步的实现,在上述智能卡 COS 多维安全机制的实现方法的基础上,当前目录的安全属性及其父目录的安全属性被分组,一组表示当前目录的安全属性,另一组表示其父目录的安全属性,从而,安全机制在安全属性的管理时,安全控制字至少含有一位表示当前目录安全属性的屏蔽与否,并含有一位表示其父目录安全属性的屏蔽与否。

[0027] 上述智能卡 COS 安全机制的实现方法,安全控制字含有一表示当前目录安全属性组内安全属性使用逻辑的位,该位的两个状态对应为逻辑与和逻辑或;

[0028] 相应地,安全控制字还含有一表示当前目录的父目录安全属性组内安全属性使用逻辑的位,该位的两个状态对应为逻辑与和逻辑或。

[0029] 上述智能卡 COS 安全机制的实现方法,安全控制字还含有一在当前目录安全属性及其父目录安全属性都没有被屏蔽的情况下表示两组内所有安全属性使用逻辑的位,该位的两个状态对应位逻辑与和逻辑或。

[0030] 上述智能卡 COS 安全机制的实现方法,使用位逻辑表示安全属性,对应为位逻辑的两个状态表示相应安全属性需要满足与否。

[0031] 上述智能卡 COS 安全机制的实现方法,当前目录的安全属性及其父目录的安全属性各有 8 个。

附图说明

[0032] 图 1 为 COS 功能模块图。

[0033] 图 2 为文件系统结构图。

[0034] 图 3 为依据本发明设定的一个文件结构图。

具体实施方式

[0035] 依据本发明提出以位逻辑实现的安全状态和安全属性,并利用安全控制字控制各安全属性关系来实现个性化文件访问权限的智能卡 COS 多维安全机制。

[0036] 控制字被存放在寄存器中,能够被定义和修改,通过预定义控制字各位所对应的操作、状态等被逆解析,完成相关的控制、操作、状态传递等。

[0037] 对应地,安全控制字在于控制安全属性的使用,具体为访问权限的控制,在安全属性中对安全控制字进行定义,并且其初始化的内容应当表现为对访问权限的全面禁止,即没有获得任何安全属性条件下的授权。

[0038] 在一个实施例中,采用位逻辑实现安全状态,共 16 个安全状态,每个安全状态是一个位逻辑 0 或逻辑 1,其中逻辑 0 表示本安全状态不满足,逻辑 1 表示本安全状态满足。

[0039] 位逻辑的对象是二进制信号,在嵌入式系统中含有规范的位逻辑指令,并且可以进行简单的布尔运算。

[0040] 依据上述内容,16 个位逻辑单元组合成两个字节。其中前 8 个安全状态为当前目录的父目录安全状态(用 PDF_AC 表示),后 8 个为当前目录安全状态(用 CDF_AC 表示)。

[0041] 在 PIN(Personal Identification Number,个人识别密码)或 KEY 属性中定义了本 PIN 或 KEY 验证后得到的安全状态 KEY_SET_AC。那么针对获得的安全状态可以区分用户及其访问权限,从而可以规定的表现在安全控制字的定义和调用上。

[0042] 安全状态获得途径如下:

[0043] 1)CDF_AC 安全控制字在“Verify”和“External Authentication”指令执行后获得。

[0044] 其中 Verify 指核实,是指对智能卡的持有者的合法性进行验证,表现在通过由用户向智能卡出示仅有他本人才知道的通行字,并由智能卡对该通行字的正确性进行判断来达到验证的目的。通行字在相关用户被设立时确定,并可以进行修改,同时,用户被设立时,其访问权限即被设定,安全控制字产生的一个条件被确定。

[0045] 另外,在通行字传送过程中避免不被窃听,还可以对传送的通行字进行加密。

[0046] External Authentication 则指外部认证,又称外部鉴别,是指对智能卡(或者是终端读写设备)的合法性进行验证,在这个过程中,通过智能卡和读写设备双方同时对任意一个相同的随机数进行相同的加密运算,如 DES(Data Encryption Standard,数据加密标准)算法,然后判断双方运算结果是否一直来达到验证的目的,外部认证用来智能卡对终端读写设备的合法性进行验证。该读写合法性也被定义,从而安全控制字产生的另一个条件被确定。

[0047] 那么,在用户设立时,一个安全控制字名义上就被分配了。

[0048] 对应的语句为: CDF_AC=CDF_AC || KEY_SET_AC。

[0049] 2)复位操作: MF 将被自动选择为当前的应用文件。并且 CDF_AC=0x00, PDF_AC=0x00。

[0050] 3)从当前的应用文件查找它的子 DF 文件: PDF_AC=CDF_AC, CDF_AC=0x00。

[0051] 4)从当前的应用文件查找它的父 DF 文件: CDF_AC=PDF_AC, PDF_AC=0x00。如果当前应用文件的父 DF 文件时 MF 文件,那么: CDF_AC=PDF_AC。

[0052] 加以对应的,即安全状态与安全属性的对应,采用位逻辑集合实现安全属性,共 16 个安全属性,每个安全属性是一个位逻辑 0 或位逻辑 1,其中逻辑 0 表示本安全属性不需要满足,逻辑 1 表示本安全属性需要满足。

[0053] 其中前 8 个安全状态对应为当前目录的父目录安全状态的安全属性,后 8 个对应当前目录安全状态的安全属性。

[0054] 安全属性是在安全属性中定义。

[0055] 那么使用安全控制字来实现安全属性的个性化管理,安全控制字也是在安全属性中定义。

[0056] 再一个实施例中安全控制字由 5 个 bit 组成。实现功能如下:

[0057] bit1 为前 8 位安全属性屏蔽位,逻辑 0 表示屏蔽,前 8 位安全属性无效;逻辑 1 表示未屏蔽,前 8 位安全属性有效。

[0058] Bit2 为后 8 位安全属性屏蔽位, 逻辑 0 表示屏蔽, 后 8 位安全属性无效; 逻辑 1 表示未屏蔽, 后 8 位安全属性有效。

[0059] Bit3 为访问权限对前 8 位安全属性关系要求, 逻辑 0 表示与, 即安全状态必须满足所有需要满足的安全属性; 逻辑 1 表示或, 即安全状态至少满足需要满足的安全属性之一即可。

[0060] Bit4 为访问权限对后 8 位安全属性关系要求, 逻辑 0 表示与, 即安全状态必须满足所有需要满足的安全属性; 逻辑 1 表示或, 即安全状态至少满足需要满足的安全属性之一即可。

[0061] Bit5 为访问权限对前 8 位安全属性和后 8 位安全属性关系的要求, bit1 和 bit2 都是逻辑 1 时有效。逻辑 0 表示与, 必须同时满足按 bit3 逻辑状态安全状态满足前 8 位安全属性和按 bit4 逻辑状态安全状态满足后 8 位安全属性; 逻辑 1 表示或, 至少满足按 bit3 逻辑状态安全状态满足前 8 位安全属性和按 bit4 逻辑状态安全状态满足后 8 位安全属性其一即可。

[0062] 依据上述定义, 在处理安全属性的使用时, 即安全机制的实现中, 通过控制字某些位的置位与复位能够灵活的进行配置。从而, 可以个性化的对安全属性进行管理。

[0063] 由于控制字存放在寄存器中, 只对寄存器有一点消耗, 寄存器位宽如 16 位、32 位和 64 位, 在当前的系统配置条件下完全可以满足所述的安全控制字的需要。

[0064] 依据上述内容, 一个实施例描述如下:

[0065] 参见说明书附图 3 所示的文件结构中, 文件 EF01 读权限安全属性为 0000001100001001B。

[0066] 那么, 设定的 EF01 读安全控制字为 11111B, 则需要 $PDF_AC \& 00000011B \neq 0$ 或 $CDF_AC \& 00001001B \neq 0$ 之一满足, 即可获取读权限。

[0067] 若 EF01 读安全控制字为 01111B, 则需要 $PDF_AC \& 00000011B \neq 0$ 和 $CDF_AC \& 00001001B \neq 0$ 都满足, 才可获取读权限。

[0068] 若 EF01 读安全控制字为 10011B, 则需要 $PDF_AC \& 00000011B = 00000011B$ 或 $CDF_AC \& 00001001B = 00001001B$ 之一满足, 即可获取读权限。

[0069] 若 EF01 读安全控制字为 00011B, 则需要 $PDF_AC \& 00000011B = 00000011B$ 和 $CDF_AC \& 00001001B = 00001001B$ 都满足, 即可获取读权限。

[0070] 若 EF01 读安全控制字为 10111B, 则需要 $PDF_AC \& 00000011B \neq 0$ 或 $CDF_AC \& 00001001B = 00001001B$ 之一满足, 即可获取读权限。

[0071] 若 EF01 读安全控制字为 00111B, 则需要 $PDF_AC \& 00000011B \neq 0$ 和 $CDF_AC \& 00001001B = 00001001B$ 都满足, 即可获取读权限。

[0072] 若 EF01 读安全控制字为 11011B, 则需要 $PDF_AC \& 00000011B = 00000011B$ 或 $CDF_AC \& 00001001B \neq 0$ 之一满足, 即可获取读权限。

[0073] 若 EF01 读安全控制字为 01011B, 则需要 $PDF_AC \& 00000011B = 00000011B$ 和 $CDF_AC \& 00001001B \neq 0$ 都满足, 即可获取读权限。

[0074] 若 EF01 读安全控制字 $x1x10B$ (x 表示该位可以是任意值), 则需要 $CDF_AC \& 00001001B \neq 0$ 满足, 即可获取读权限。

[0075] 若 EF01 读安全控制字 $x0x10B$, 则需要 $CDF_AC \& 00001001B = 00001001B$ 满足, 即可

获取读权限。

[0076] 若 EF01 读安全控制字 xx101B, 则需要 PDF_AC&00000011B $\neq$ 0 满足, 即可获取读权限

[0077] 若 EF01 读安全控制字 xx001B, 则需要 PDF_AC&00000011B=00000011B 满足, 即可获取读权限。

[0078] 一种以位逻辑定义的安全状态和安全属性, 并利用安全控制字控制各安全属性关系来实现个性化文件访问权限的智能卡 COS 多维安全机制被进一步描述如下:

[0079] 设定一个文件结构如图 3 所示:

[0080] ADF1 下的 EF01 的读权限需要验证 DF01 下 ISF 文件中的 PIN1 和 KEY1, 同时需要验证 ADF1 下 ISF 文件中的 PIN2 或 KEY2。以下步骤实现 EF01 的安全机制的和访问权限:

[0081] 实现安全机制如下:

[0082] 1、定义 DF01 下 PIN1 验证后得到的安全状态是 01H, KEY1 的验证后得到的安全状态为 02H;

[0083] 2、定义 ADF1 下 PIN2 验证后得到的安全状态是 01H, KEY2 的验证后得到的安全状态为 04H;

[0084] 3、定义 EF01 的读安全属性为 00000110000101B, 即对应父目录的安全状态 01H、02H, 当前目录的安全状态 01H、04H。

[0085] 4、定义 EF01 的读安全控制字为 11011B, 即当前目录的父目录安全状态和当前目录的安全状态都未屏蔽, 当前目录的父目录安全状态 01H、02H 都要满足同时当前目录的安全状态 01H、04H 至少满足一个才可以获得 EF01 读权限。

[0086] 获取 EF01 读权限如下:

[0087] 5、先选择 DF01, 当前目录安全状态为 00H。

[0088] 6、验证 PIN1 获得安全状态 01H, 得到当前目录安全状态为 01H。

[0089] 7、“External Authentication” 指令验证 KEY1, 获得安全状态 02H; 得到当前目录安全状态为 03H。

[0090] 8、选择 ADF1, 父目录安全状态为 03H, 当前目录安全状态为 00H

[0091] 9.1 验证 PIN2, 获得安全状态 01H, 得到当前目录安全状态为 01H。转到步骤 10

[0092] 9.2 “External Authentication” 指令验证 KEY1, 获得安全状态 04H; 得到当前目录安全状态为 04H。转到步骤 10

[0093] 9.3 验证 PIN2, 获得安全状态 01H, 得到当前目录安全状态为 01H; “External Authentication” 指令验证 KEY1, 获得安全状态 04H; 得到当前目录安全状态为 05H。转到步骤 10

[0094] 10、满足 EF01 读权限, 执行 EF01 读操作。

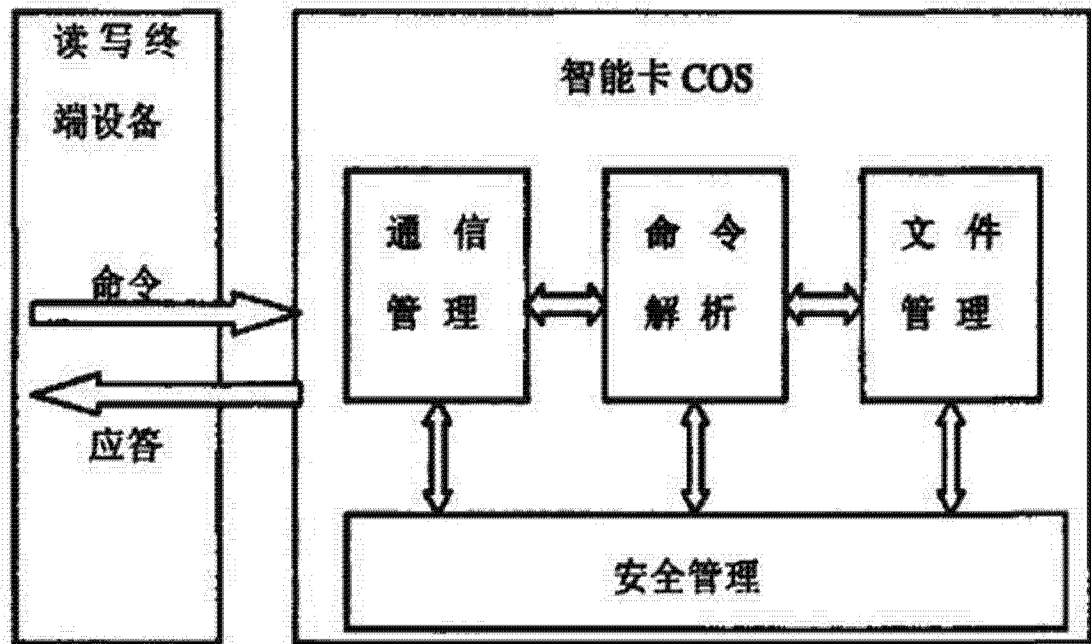


图 1

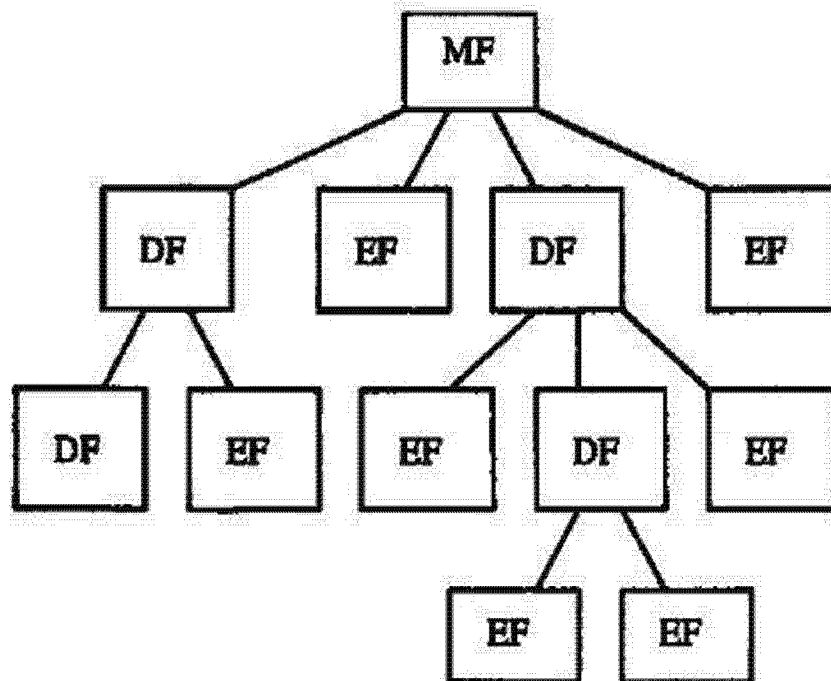


图 2

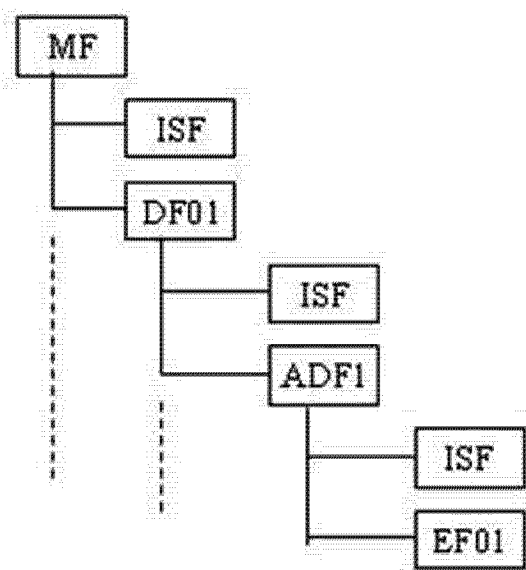


图 3