(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 1898956 B
(45)授权公告日 2012.02.22

(21)申请号 200480035774.2

(22)申请日 2004.12.15

(30)优先权数据
421616/2003 2003.12.18 JP
60/530,663 2003.12.19 US

(85)PCT申请进入国家阶段日
2006.06.02

(86)PCT申请的申请数据
PCT/JP2004/019126 2004.12.15

(87)PCT申请的公布数据
WO2005/060256 EN 2005.06.30

(73)专利权人 松下电器产业株式会社
地址 日本大阪府

(72)发明人 楠堂忠夫 盐见隆一

(74)专利代理机构 永新专利商标代理有限公司 72002
代理人 钟胜光

(51)Int.Cl.
H04N 21/443(2011.01)
G06F 9/445(2006.01)

(56)对比文件
US 2003/0217369 A1,2003.11.20, 全文.
CN 1367628 A,2002.09.04, 全文.
US 5625693 A,1997.04.29, 全文.
US 2003/0114144 A1,2003.06.19, 全文.
DVB.Digital Video Broadcasting (DVB) Multimedia Home Platform(MHP) Specification 1.1.1,ETSI TS 102 812 V1.2.1.2003, 第301-315,334-351页.
CABLELABS.OpenCable Application Platform Specification,OCAP 1.0 Profile, OC-SP-OCAP1.0-IF-I09-031121.2003, 第38-53,73-80,108-118页、图14-1.

审查员 夏刊

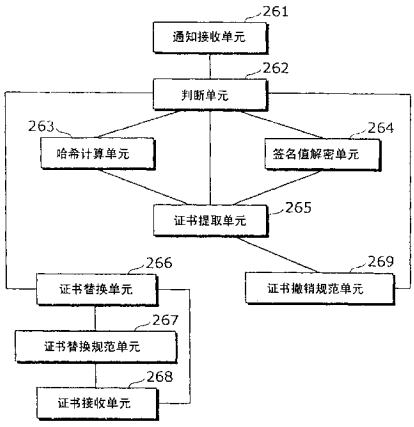
权利要求书 2 页 说明书 24 页 附图 43 页

(54)发明名称

验证和运行应用程序的方法

(57)摘要

根据常规技术,在一旦将程序保存到非易失性存储器中并随后被激活的情况下,紧接着在该激活之前执行该程序的验证。然而在开始激活该程序之前,需要诸如解密加密值这样的计算,这造成了程序响应度与所需的计算时间成比例地下降的问题。为了解决这个问题,紧接着在该程序被存储之前,执行程序的验证,以便在程序激活时不执行验证或仅进行一部分验证来检验证书的有效性。



CN 1898956 B

1. 一种已验证程序运行方法，包括：

验证和保存步骤，验证包括在被接收的传输流中的程序，并且根据与已验证程序的每一个数据文件的存储相关信息，将所述程序保存到广播接收机中；以及

运行步骤，运行所保存的已验证程序；

其中所述验证和保存步骤包括：

第一步骤，检验第一哈希值和第二哈希值是否一致，所述第一哈希值是根据在所述程序中包括的每一个数据文件计算得到的，所述第二哈希值保存在与所述每一个数据文件相对应的哈希文件中；

第二步骤，检验在所述程序中包括的证书文件是否有效；

第三步骤，检验解密值和第三哈希值是否一致，所述解密值是通过使用在所述程序的所述证书文件中包括的叶证书的公开密钥，来解密在所述程序中包括的签名文件的签名值而获得的，所述第三哈希值是通过根据位于所述程序的顶级目录中的哈希文件计算得到的；以及

第四步骤，在满足所有以下条件的情况下：在所述第一步骤中所述第一和第二哈希值被检验是一致的；在所述第二步骤中所述证书文件被检验是有效的；以及在所述第三步骤中所述解密值和第三哈希值被检验是一致的，验证所述程序并根据所述存储相关信息，在辅助存储单元中保存所述已验证程序的每一个数据文件，而不运行所述已验证程序，其中，该解密值是通过解密该签名值而获得的，该辅助存储单元是该广播接收机中包含的记录单元并且即使在电源断开时保存所存储的数据；以及

所述运行步骤包括：

第五步骤，对于在所述辅助存储单元中保存的程序，仅再执行在所述验证和保存步骤中已执行的第一至第三步骤中的第二步骤；以及

在所述运行步骤中，再次验证所述辅助存储单元中保存的程序，并且仅在所述第五步骤中在所述辅助存储单元中保存的程序中所包括的所述证书文件被检验为有效的情况下，运行所述保存的程序。

2. 一种已验证程序运行装置，包括：

验证和保存单元，用于验证包括在被接收的传输流中的程序，并且根据与已验证程序的每一个数据文件的存储相关信息来保存所述程序；以及

运行单元，用于运行所述保存的已验证程序；

其中所述验证和保存单元包括：

第一检验单元，用于检验第一哈希值和第二哈希值是否一致，所述第一哈希值是根据在所述程序中包括的每一个数据文件计算得到的，所述第二哈希值保存在与所述每一个数据文件相对应的哈希文件中；

第二检验单元，用于检验在所述程序中包括的证书文件是否有效；

第三检验单元，用于检验解密值和第三哈希值是否一致，所述解密值是通过使用在所述程序的所述证书文件中包括的叶证书的公开密钥，来解密在所述程序中包括的签名文件的签名值而获得的，所述第三哈希值是通过根据位于所述程序的顶级目录中的哈希文件计算得到的；以及

保存单元，用于在满足所有以下条件的情况下：所述第一和第二哈希值被所述第一检

验单元检验为是一致的；所述证书文件被所述第二检验单元检验为是有效的；以及所述解密值和第三哈希值被所述第三检验单元检验为是一致的，验证所述程序并根据所述存储相关信息，保存所述已验证程序的每一个数据文件，而不运行所述已验证程序；其中，该解密值是通过解密该签名值而获得的；以及

所述运行单元包括：

第四检验单元，用于对于在所述保存单元中保存的程序，再次执行检验操作，所述检验操作是与在所述验证和保存单元的所述第一、第二、第三检验单元已执行的检验中由所述第二检验单元执行的检验相同的操作；

其中，所述运行单元再次验证所述保存单元中保存的程序，并且仅在所述第四检验单元验证在所述保存单元中保存的程序中所包括的证书文件是有效的情况下，运行所述保存的程序。

验证和运行应用程序的方法

技术领域

[0001] 本发明涉及一种已验证程序运行方法，其检验所下载的程序的可靠性并运行已被检验为可靠的程序。

背景技术

[0002] 在数字电视中下载程序以及检查／保证该程序的可靠性的功能已在DVB-MHP规范“ETSI TS 101 812 V1.2.1 DVB-MHP Specification1.0.2”等中描述了。这个DVB-MHP规范规定了检验叠加到广播电波上的被接收的程序没有被篡改以及检验该程序是否由可信赖的机构发行的功能。这个功能使得防止被改写的程序和电子欺骗第三方的程序被激活成为可能，其中，该被改写的程序没有按照初始所要求的那样工作，由此可能对数字电视造成损害。

[0003] 此外，日本公开的专利申请No.2000-29833描述了一项技术，它由存储和发送数据的服务器装置和通过网络接收数据的终端装置组成，该技术可以防止利用在终端装置上存储接收到的数据来对存储的数据进行非法使用。日本公开的专利申请No.2000-29833的图1描绘了该技术，其中，响应于来自终端装置20的请求，服务器装置10将保存在存储单元15中的数据拷贝到存储单元23中，且当想使用保存在存储单元23中的数据时，查询单元26向服务器装置10进行查询，认证单元13对该数据的使用进行审核，如果没有问题，则终端装置20使用该数据。即使电源打开／关闭，上述装置也能够在对保存于非易失性存储器中的数据进行可靠性检查后下载该数据。检查程序和数据的可靠性在下文中称作验证。

[0004] 但是，根据常规技术，在一旦在把程序保存到非易失性存储器中以在该装置通电／断电后激活该程序的情况下，紧接着在程序被激活之前对其进行验证。在这种情况下，在开始激活所述程序之前， 执行诸如对已加密值进行解密之类的计算是有必要的，这造成了随着计算需要越长的时间响应度降低越多的问题。尤其是在程序被频繁地激活或是程序容量很大的情况下，响应度变得越来越低，因为计算量与激活频率和容量成正比地增加。

[0005] 鉴于上述问题，期望提供一种诸如具有高响应度的数字电视这样的程序验证装置，它能够在保证程序可靠性的同时，缩短程序被激活之前所需的时间。

发明内容

[0006] 本发明的目的在于提供一种已验证程序运行方法，其能够通过紧接着在程序被存储前执行验证以及在程序激活时不执行验证或仅执行一部分验证，来保证可靠性并且改善响应度。

[0007] 为了解决上述常规问题，根据本发明的已验证程序运行方法包括：验证和保存步骤，验证包括在传输流中的程序，并且根据与所述已验证程序的每一个数据文件的存储相关的信息，将所述程序保存到广播接收机中；以及运行步骤，运行所述保存的已验证程序；其中所述验证和保存步骤包括：第一步骤，检验两个哈希值是否一致，所述哈希值的其中一个是根据在所述程序中包括的每一个数据文件计算得到的，另一个哈希值保存在与所述每

一个数据文件相对应的哈希文件中；第二步骤，检验在所述程序中包括的证书文件是否有效；第三步骤，检验解密值和哈希值是否一致，所述解密值是通过使用在所述程序的所述证书文件中包括的叶证书的公开密钥，来解密在所述程序中包括的签名文件的签名值而获得的，所述哈希值是通过根据位于所述程序的顶级目录中的哈希文件计算得到的；以及第四步骤，在满足所有以下条件的情况下：在所述第一步骤中所述两个哈希值被检验是一致的；在所述第二步骤中所述证书文件被检验是有效的；以及在所述第三步骤中所述解密值和哈希值被检验是一致的，验证所述程序并根据所述与存储有关的信息来保存所述已验证程序的每一个数据文件，以及所述运行步骤包括第五步骤，检验在所述保存的程序中包括的所述证书文件是否有效，以及 在所述运行步骤中，再次验证所述保存的程序，并且仅在所述第五步骤中在所述保存的程序中所包括的所述证书文件被检验为有效的情况下，运行所述保存的程序。

[0008] 因此，缩短在程序被激活前的所需时间同时保证程序可靠性变得可能。

[0009] 此外，在所述程序具有目录结构的情况下，在每一个目录中所包括的每一个数据文件和与所述每一个数据文件相对应的所述哈希文件都位于相同的目录中，以及为在每一个目录中所包括的每一个数据文件运行所述第一步骤。

[0010] 因此，对于包含在每个目录中的每个数据文件，检验根据数据文件计算得出的哈希值和保存在与所述数据文件相对应的哈希文件中的哈希值是否一致变得可能。

[0011] 此外，所述第二步骤包括第六步骤，检验两个根证书是否匹配，所述根证书的其中一个位于在所述程序内所包括的所述证书文件中，另一个根证书安装在所述广播接收机中，以及在所述第二步骤中，在所述两个根证书匹配的情况下所述证书文件被检验为有效。

[0012] 在这里，第二步骤还包含第七步骤，检验在所述程序内包括的所述证书文件中的每一个证书的有效期，以及在所述第二步骤中，在满足以下两个条件的情况下：所述两个根证书匹配；以及执行所述验证的时间在所述证书文件中的每一个证书的有效期内，所述证书文件被检验为有效。

[0013] 因此，在根证书不匹配以及所述证书期满的情况下，防止程序被保存变得可能。

[0014] 此外，第五步骤包括第八步骤，检验两个根证书是否匹配，所述根证书的其中一个位于在所述保存的程序中包括的所述证书文件内，另一个根证书安装在所述广播接收机中，以及在所述第五步骤中，在所述两个根证书匹配的情况下，在所述保存的程序内包括的所述证书文件被检验为有效。

[0015] 在这里，第五步骤还包括第九步骤，检验在所述保存的程序内包括的所述证书文件中的每一个证书的有效期，以及在所述第五步骤中，在满足以下两个条件的情况下：所述两个根证书匹配；以及执行所述运行的时间位于在所述证书文件中的每一个证书的有效期内，在所述保存的程序中包括的所述证书文件被检验为有效。

[0016] 因此，在根证书不匹配以及所述证书期满的情况下，防止该程序被运行变得可能。

[0017] 注意，不仅可以将本发明实现为上述的已验证程序运行方法，而且可以将本发明实现为包括作为其单元的、所述已验证程序运行方法中包括的特征步骤的已验证程序运行装置，以及作为使得计算机运行这些步骤的程序。还应该注意，可以在诸如 CD-ROM 这样的记录介质上和经由诸如因特网这样的传输介质来分发该程序。

[0018] 从以上描述可以明显看出，根据本发明的已验证程序运行方法能够在保证程序可

靠性的同时缩短在程序被激活前的所需时间。

[0019] 在此，以参考的方式作为其整体插入在 2003 年 12 月 18 日提交的日本专利申请 No. 2003-421616 和在 2003 年 12 月 19 日提交的临时美国专利申请 No. 60/530663 的所有内容，包括说明书、附图和权利要求书。

附图说明

[0020] 根据下面结合举例说明本发明具体实施例的附图的描述，本发明的这些以及其它目的、优点和特点将变得明显。

[0021] 图 1 是示出了根据本发明第一实施例的有线电视系统的结构的框图；

[0022] 图 2 是示出了根据本发明的有线电视系统中使用将用于在首端和多个终端装置之间通信的频带的示例图；

[0023] 图 3 是示出了根据本发明的有线电视系统中使用将用于所述首端和所述终端装置之间通信的频带的示例图；

[0024] 图 4 是示出了根据本发明的有线电视系统中使用将用于所述首端和所述终端装置之间通信的频带的示例图；

[0025] 图 5 是示出了根据本发明的有线电视系统中的终端装置结构的示意图；

[0026] 图 6 是示出了根据本发明的有线电视系统中的终端装置示例外观图的示意图；

[0027] 图 7 是示出了根据本发明的 POD 504 硬件结构的示意图；

[0028] 图 8 是示出了根据本发明的 POD 504 中所存储的程序结构的示意图；

[0029] 图 9 是示出了 MPEG 标准中所定义的分组结构的示意图；

[0030] 图 10 是示出了 MPEG2 传输流示例的示意图；

[0031] 图 11 是示出了在以面板形式配置输入单元 513 的情况下输入单元 513 的示例外观图的示意图；

[0032] 图 12 是示出了根据本发明的存储在终端装置 500 中的程序结构的示意图；

[0033] 图 13A 是示出了根据本发明的由显示装置 509 所显示的一个显示屏示例的示意图，图 13B 是示出了根据本发明的由显示装置 509 所显示的一个显示屏示例的示意图；

[0034] 图 14 是示出了根据本发明的在辅助存储单元 510 中所存储的信息示例的示意图；

[0035] 图 15A、15B 和 15C 是每一个示出了根据本发明的在主存储单元 511 中所存储的信息示例的示意图；

[0036] 图 16 是示出了根据本发明的在 MEPG2 标准中所规定的 PAT 内容的示意图；

[0037] 图 17 是示出了根据本发明的在 MEPG2 标准中所规定的 PMT 内容的示意图；

[0038] 图 18 是示出了根据本发明的在 DVB-MHP 标准中所规定的 AIT 内容的示意图；

[0039] 图 19 是示出了根据本发明的将以 DSMCC 格式传输的文件系统的示意图；

[0040] 图 20 是示出了根据本发明的 XAIT 内容的示意图；

[0041] 图 21 是示出了根据本发明的在辅助存储单元 510 中所存储的信息示例的示意图；

[0042] 图 22A、22B 和 22C 是分别示出了根据本发明的包含文件或目录的哈希值的文件示例的示意图；

[0043] 图 23 是示出了根据本发明的证书链结构的示意图；

[0044] 图 24 是示出了根据本发明的 X.509 证书结构的示意图；

[0045] 图 25 是示出了根据本发明的签名文件结构的示意图；

[0046] 图 26 是示出了根据本发明的安全模块组件的示意图；

[0047] 图 27 是示出了根据本发明的在验证文件系统时所执行的操作的流程图；

[0048] 图 28 是根据本发明的在接收到程序预激活通知时不执行验证的情况下的流程图；

[0049] 图 29 是示出了根据本发明的在对文件系统执行篡改检查时的操作的流程图；

[0050] 图 30 是示出了根据本发明的在通过使用签名文件来执行篡改检查时的操作的流程图；

[0051] 图 31 是示出了根据本发明的在检查叶证书和中间证书之间的链关系时所执行的操作的流程图；

[0052] 图 32 是示出了根据本发明的在检查中间证书和根证书之间的链关系时所执行的操作的流程图；

[0053] 图 33 是示出了根据本发明的在检查根证书中的签名时所执行的操作的流程图；

[0054] 图 34 是示出了根据本发明的用于指定将被存储的文件的文件示例的示意图；

[0055] 图 35 是示出了根据本发明的在执行文件系统的验证时所执行的操作的流程图；

[0056] 图 36 是示出了根据本发明的当接收到预激活通知时在检查 X.509 证书的有效性的时候所执行的操作的流程图；

[0057] 图 37 是示出了根据本发明的将从下载模块中接收到的编码文件简化结构的示意图；

[0058] 图 38A、38B 和 38C 是每一个示出了根据本发明的由终端装置拥有的证书被替换的示意图；

[0059] 图 39 是示出了根据本发明的当执行证书替换时所执行的操作的流程图；

[0060] 图 40 是示出了根据本发明的当接收到预激活通知时在比较根证书的时候所执行的操作的流程图；

[0061] 图 41 是示出了根据本发明的 CRL 结构的示意图；

[0062] 图 42 是示出了根据本发明的 CRL 中已被撤销证书列表的示意图；

[0063] 图 43 是示出了根据本发明的包含 CRL 的文件系统示例的示意图；

[0064] 图 44 是示出了根据本发明的当基于哈希值和签名值检查 CRL 的有效性时所执行的操作的流程图；

[0065] 图 45 是示出了根据本发明的当基于证书间的关系和根证书间的比较结果来检查 CRL 的有效性时所执行的操作的流程图；

[0066] 图 46 是示出了根据本发明的包含有文件或目录的哈希值的文件示例的示意图；

[0067] 图 47 是示出了根据本发明的用于在程序保存时存在 CRL 的情况下执行验证的操作的流程图；

[0068] 图 48 是示出了用于在激活程序时存在 CRL 的情况下执行验证的操作的流程图；

[0069] 图 49 是示出了根据本发明的已撤销证书数据库的示意图；

[0070] 图 50 是示出了根据本发明的包含有用于指定将被存储的文件的文件的文件系统

示例的示意图；

[0071] 图 51 是示出了根据本发明的用于指定将被存储的文件的文件示例的示意图。

[0072] 最优实施例

[0073] 下面结合附图对本发明的实施例进行描述。

[0074] （第一实施例）

[0075] 下面将参照附图来说明根据本发明的有线电视系统的优选实施例。图 1 是示出了组成所述有线系统的装置之间的关系的方框图，这些装置是首端 101 和三个终端装置：终端装置 A111、终端装置 B112 以及终端装置 C113。在本实施例中，虽然三个终端装置连接到一个首端，但是如果任意数量的终端装置连接到该首端则实现本发明也是可以的。

[0076] 首端 101 向多个终端装置传送诸如视频、音频和数据这样的广播信号，并且接收从这些终端装置传送的数据。为了实现这个功能，对频带进行划分以用于在首端 101 和终端 A111、终端 B112 以及终端 C113 之间的数据传送。图 2 是示出了已划分频带示例的表。粗略地说有两种类型的频带：带外（缩写为 OOB）和带内。频带 5 ～ 130MHz 被分配给 OOB，主要用于首端 101 和终端装置 A111、终端装置 B112 以及终端装置 C113 之间进行数据交换。频带 130MHz ～ 864MHz 被分配给带内，主要用于包含视频和音频的广播频道。OOB 使用 QPSK，而带内使用 QAM64 作为调制技术。这里省略了调制技术的详细说明，因为它们是与本发明几乎不相关的公知技术。图 3 示出了如何使用 OOB 频带的更具体示例。频带 70MHz ～ 74MHz 用于从首端 101 传送数据。在这种情况下，所有终端装置 A111、终端装置 B112 以及终端装置 C113 从首端 101 接收相同的数据。同时，频带 10.0MHz ～ 10.1MHz 用于从终端装置 A111 向首端 101 传送数据，频带 10.1MHz ～ 10.2MHz 用于从终端装置 B112 向首端 101 传送数据，频带 10.2MHz ～ 10.3MHz 用于从终端装置 C113 向首端 101 传送数据。因此，从终端装置 A111、终端装置 B112 和终端装置 C113 向首端 101 传送对于每个终端装置唯一的数据是可能的。图 4 示出了使用带内频带的示例。频带 150 ～ 156MHz 和 156 ～ 162MHz 被分别分配给电视频道 1 和电视频道 2，且后面的频率以 6MHz 的间隔分配给电视频道。310MHz 和后面的频率以 1MHz 的间隔分配给无线电频道。上述频道的每一个可用于模拟广 播或数字广播。在数字广播的情况下，以遵循 MPEG2 规范的分组格式来传送数据，在这种情况下，除了音频和视频数据外，还能够传送打算用于各种数据广播系统的数据。

[0077] 首端 101 配置有 QPSK 调制单元、QAM 调制单元等以向各个频率范围传送合适的广播信号。此外，首端 101 配置有 QPSK 解调单元用于从终端装置接收数据。而且，还假定首端 101 配置有与上述调制单元和解调单元相关的各种设备。但是，这里省略它们的详细说明，因为本发明主要涉及终端装置。

[0078] 终端装置 A111、终端装置 B112 以及终端装置 C113 接收并再现从首端 101 传送的广播信号。此外，终端装置 A111、终端装置 B112 以及终端装置 C113 向首端 101 传送对于每个终端装置唯一的数据。在本实施例中，这三个终端装置可以有相同的结构。

[0079] 图 5 是示出了每个终端装置的硬件结构的框图。500 是一个终端装置，其由 QAM 解调单元 501、QPSK 解调单元 502、QPSK 调制单元 503、TS 解码器 505、音频解码器 506、扬声器 507、视频解码器 508、显示器 509、辅助存储单元 510、主存储单元 511、ROM512、输入单元 513 和 CPU514 组成。此外，POD504 可与终端装置 500 相连或分离。

[0080] 图 6 示出了一种薄形电视，其是终端装置 500 的示例外观图。虽然该终端装置能

够以各种结构来出现，但在本实施例中，将基于 OpenCable(TM) 和 OCAP 而配置的终端装置描述为一个示例。

[0081] 601 是薄形电视的铁盒，其中包含除 POD 504 外的终端装置 500 的所有组件。

[0082] 602 是显示器，其与图 5 中的显示器 509 相对应。

[0083] 603 是面板单元，其由多个按钮组成并与图 5 中的输入单元 513 相对应。

[0084] 604 是信号输入终端，电缆线与其连接用于向首端 101 传送信号和从首端 101 接收信号。该信号输入终端连接到图 5 中的 QAM 解调单元 501、QPSK 解调单元 502 以及 QPSK 调制单元 503。

[0085] 605 是与图 5 中的 POD 504 相对应的 POD 卡。如在图 6 中的 POD 卡 605 的情况下，POD 504 独立于终端装置 500 而实现，并且能够与终端装置 500 连接 / 分离。稍后将给出 POD 504 的详细说明。

[0086] 606 是插入 POD 卡 605 的插槽。

[0087] 参照图 5，根据包含由 CPU 514 指定的频率的调谐信息，QAM 解调单元 501 对在首端 101 中已被 QAM 调制并从首端 101 传送的信号进行解调，并将结果传递给 POD 504。

[0088] 根据包含由 CPU 514 指定的频率的调谐信息，QPSK 解调单元 502 对在首端 101 中已被 QPSK 调制并从首端 101 传送的信号进行解调，并将结果传递给 POD 504。

[0089] 根据包含由 CPU 514 指定的频率的解调信息，QPSK 调制单元 503 对从 POD 504 传递来的信号进行 QPSK 调制，并将结果传送给首端 101。

[0090] 如图 6 所示，POD 504 与终端装置 500 的主体相分离。终端 500 的主体与 POD 504 之间的连接接口的定义已在 OpenCable(TM) CableCARD(TM) Interface Specification (OC-SP-CC-IF-I15-031121) 中以及此规范所参考的规范中给出。注意此规范中的 CableCARD 指的是 POD。这里省略了详细描述，并仅对与本发明相关的组件加以说明。

[0091] 图 7 是示出 POD 504 内部结构的框图。POD 504 由第一解扰器单元 701、第二解扰器单元 702、扰码器单元 703、主存储单元 704、辅助存储单元 705 和 CPU 706。

[0092] 根据来自 CPU 706 的指令，第一解扰器单元 701 从终端装置 500 的 QAM 解调单元 501 中接收已加扰的信号，并对该信号进行解扰。然后，第一解扰器单元 701 向终端装置 500 的 TS 解码器 505 传送已解扰的信号。根据需要，由 CPU706 提供解扰器所需的诸如密钥这样的信息。更特别地，首端 101 广播若干个付费频道，且当用户购买了收看这些付费频道的权限时，第一解扰器单元 701 从 CPU706 接收诸如密钥这样的所需信息并执行解扰。因此，用户能够收看这些付费频道。当诸如密钥这样的所需信息不被提供时，第一解 扰器单元 701 没有执行解扰而直接将接收的信号传递给 TS 解码单元 505。

[0093] 根据来自 CPU 706 的指令，第二解扰器单元 702 从终端装置 500 的 QPSK 解调单元 502 中接收已加扰的信号，并且对该信号进行解扰。然后，第二解扰器单元 702 将该已解扰的数据传递给 CPU706。

[0094] 根据来自 CPU 706 的指令，扰码器单元 703 对从 CPU 706 接收到的数据进行加扰，并且将结果发送到终端装置 500 的 QPSK 调制单元 503 中。

[0095] 其具体组件是诸如 RAM 这样的主存储器的主存储单元 704，打算用于当 CPU 706 执行处理时临时存储数据。

[0096] 其具体组件是诸如快闪 ROM 这样的辅助存储器的辅助存储单元 705，用于存储 CPU

706 所运行的程序以及用于存储即使关闭电源也从不被删除的数据。

[0097] CPU 706 运行保存在辅助存储单元 705 中的程序。该程序由若干个子程序组成。图 8 示出了在辅助存储单元 705 中保存的程序示例。在图 8 中，程序 800 由若干子程序组成，包括主程序 801、初始化子程序 802、网络子程序 803、再现子程序 804 和 PPV 子程序 805。

[0098] 在这里，PPV 是每视付费的缩写，指的是允许用户在计费的基础上观看诸如电影这样的某种节目的服务。当用户输入他 / 她的身份识别码时，用户购买观看该节目的权限的事实被通知给首端 101，并且解扰该节目。因此，用户能够观看该节目。观看此节目需要用户在日后对该购买付款。

[0099] 主程序 801 是当开启电源时由 CPU 706 首先激活的子程序，其控制其它子程序。

[0100] 初始化子程序 802，在开启电源时由主程序 801 激活，其实施与终端装置 500 的信息交换等以执行初始化处理。该初始化处理已在 OpenCable(TM) CableCARD(TM) Interface Specification(OC-SP-CC-IF-I15-031121) 中以及该规范所参考的规范中详细定义。此外，初始化子程序 802 也执行在这些规范中没有定义的初始化处理。这里，引入该初始化处理的一部分。当开启电源时，初始化子程序 802 经由终端装置 500 的 CPU 514 向 QPSK 解调单元 502 通知存储在辅助存储单元 705 中的第一频率。QPSK 解调单元 502 使用提供的第一频率进行调谐，并将得到的信号传送给第二解扰器单元 702。此外，初始化子程序 802 向第二解扰器单元 702 提供诸如存储在辅助存储单元 705 中的第一密钥这样的解扰信息。因此，第二解扰器单元 702 执行解扰并将结果传递给运行子程序 802 的 CPU706。因此，初始化子程序能够接收该信息。在本实施例中，初始化子程序 802 经由网络子程序 803 接收信息。稍后将给出关于这的详细描述。

[0101] 此外，初始化子程序 802 经由终端装置 500 的 CPU 514 向 QPSK 调制单元 503 通知存储在辅助存储单元 705 中的第二频率。初始化子程序 802 向扰码器单元 703 提供存储在辅助存储单元 705 中的加扰信息。当初始化子程序 802 经由网络子程序 803 向扰码器单元 703 提供被发送的所需信息时，扰码器单元 703 使用所提供的加扰信息对数据进行加扰，并将该已加扰的数据提供给 QPSK 调制单元 503。QPSK 调制单元 503 对接收到的已加扰信息进行调制，并将该已调制信息发送到首端 101。

[0102] 因此，初始化子程序 802 经由终端装置 500、第二解扰器单元 702、扰码器单元 703 和网络子程序 803 实施与首端 101 的双向通信变成可能。

[0103] 由诸如主程序 801 和初始化子程序 802 这样的若干个子程序所使用的网络子程序 803，是用于实施与首端 101 的双向通信的子程序。更为特别地，网络子程序 803 表现为似乎使用网络子程序 803 的其它子程序正在实施与首端 101 的遵循 TCP/IP 的双向通信。这里省略了 TCP/IP 的详细说明，因为它是指定当在多个终端之间交换信息时所使用的协议的公知技术。当开启电源的时刻被初始化子程序 802 激活时，网络子程序 803 经由终端装置 500 向首端 101 通知一个 MAC 地址（媒体接入控制的缩写）以请求获得 IP 地址，所述 MAC 地址是标识 POD 504 的标识符并且预先保存在辅助存储单元 705 中。首端 101 经由终端装置 500 将所述 IP 地址通知给 POD 504，网络子程序 803 将该 IP 地址存储到主存储单元 704 中。此后，首端 101 和 POD 504 使用该 IP 地址作为 POD 504 的标识符来相互通信。

[0104] 再现子程序 804 向第一解扰器单元 701 提供存储在辅助存储单元 705 中的诸如第二密钥这样的解扰信息和由终端装置 500 提供的诸如第三密钥这样的解扰信息，以允许执

行解扰。此外，再现子程序 804 经由网络子程序 803 接收信息，该信息表明在第一解扰器单元 701 中输入的信号是一个 PPV 频道。根据该信号是 PPV 频道的通知，再现子程序 804 激活 PPV 子程序 805。

[0105] 当被激活以后，PPV 子程序 805 在终端装置 500 上显示提示用户购买该节目的消息，然后接收来自用户的输入。更特别地，当想要在屏幕上显示的信息被发送给终端装置 500 的 CPU 514 时，在终端装置 500 的 CPU 514 上运行的程序将该消息显示在终端装置 500 的显示器 509 上。然后，当用户经由终端装置 500 的输入单元 513 输入身份识别码时，终端装置 500 的 CPU 514 接收它并将它发送给在 POD 504 的 CPU 706 上运行的 PPV 子程序 805。PPV 子程序 805 经由网络子程序 803 将所接收的个人身份识别码发送给首端 101。当该个人身份识别码有效时，首端 101 经由网络子程序 803 向 PPV 子程序 805 通知诸如第四密钥这样的解扰所需的解扰信息。PPV 子程序 805 向第一解扰器单元 701 提供所接收的诸如第四密钥这样的解扰信息，然后第一解扰器单元 701 对输入信号进行解扰。

[0106] 参照图 5，TS 解码器 505 对从 POD 504 接收的信息执行过滤，并将必要的数据传递给音频解码器 506、视频解码器 508 和 CPU514。在这里，从 POD 504 发送的信号是 MPEG2 传输流。在 MPEG 规范 ISO/IEC138181-1 中已给出了关于 MPEG2 传输流的详细描述，因此本实施例不对它进行详细说明。MPEG2 传输流由多个长度固定的分组组成，并且每个分组都分配一个分组 ID。图 9 是示出了分组结构的示意图。900 是一个分组，其包括固定长度的 188 字节。头四个字节是头部 901，存储用于标识该分组的信息，其余的 184 字节 是载荷 902，存储想要承载的信息。903 示出了头部 901 的分解。分组 ID 包含于从 1^{st} 到 12^{th} ～ 24^{th} 比特的 13 个比特中。图 10 是举例说明将要传送的多个分组串的示意图。分组 1001 在它的头部中包含有分组 ID“1”，并在它的载荷中包含有视频 A 的第一信息。分组 1002 在它的头部中包含有分组 ID“2”，并在它的载荷中包含有音频 A 的第一信息。分组 1003 在它的头部中包含有分组 ID “3”，并在它的载荷中包含有音频 B 的第一信息。

[0107] 分组 1004 在它的头部中包含有分组 ID“1”，并在它的载荷中包含有视频 A 的第二信息，该第二信息是分组 1001 的后续信息。类似地，分组 1005、1026 和 1027 承载了其它分组的后续数据。通过以上述方式来连接具有相同分组 ID 的分组的载荷内容，以连续的次序来再现视频和音频是可能的。

[0108] 参照图 10，当 CPU 514 向 TS 解码器 505 指示分组 ID“1”和作为输出目标的“视频解码器 508”时，TS 解码器 505 从 POD 504 接收到的 MPEG2 传输流中提取出分组 ID 为“1”的分组，并将它们传递给视频解码器 508。因此，在图 10 中只有视频数据会被传递给视频解码器 508。同时，当 CPU 514 向 TS 解码器 505 指示分组 ID “2”和作为输出目标的“音频解码器 506”时，TS 解码器 505 从 POD 504 接收到的 MPEG2 传输流中提取出分组 ID 为“2”的分组，并将它们传递给音频解码器 506。在图 10 中，只有音频数据会被传递给音频解码器 506。

[0109] 该根据分组 ID 仅提取需要的分组的处理与将由 TS 解码器 505 执行的过滤相对应。TS 解码器 505 能够根据来自 CPU 514 的指令同时执行多于一个过滤处理。

[0110] 参照图 5，音频解码器 506 将嵌入在由 TS 解码器 505 提供的 MPEG2 传输流中的分组内的音频数据连接起来，对该已连接的数据执行数模转换，并将结果输出到扬声器 507。

[0111] 扬声器 507 将音频解码器 506 所提供的信号作为音频输出。

[0112] 视频解码器 508 将嵌入在由 TS 解码器 505 提供的 MPEG2 传输流中的分组内的视频数据连接起来，对该已连接的数据执行数模转换，并将结果输出到显示器 509 中。

[0113] 其具体组件为 CRT 或者液晶显示屏等的显示器 509 输出由视频解码器 508 所提供的视频信号，并且显示由 CPU 514 指定的信息等。

[0114] 其具体组件为快闪存储器、硬盘等的辅助存储单元 510，保存和删除由 CPU 514 指定的数据和程序。所保存的数据和程序由 CPU514 提出。即使当终端装置 500 被关闭电源时，所保存的数据和程序仍然保存在存储器中。

[0115] 其具体组件为 RAM 等的主存储器 511，临时保存由 CPU 514 指定的数据和程序，并删除它们。所保存的数据和程序由 CPU 514 提出。当终端装置 500 被关闭电源时，所保存的数据和程序被删除。

[0116] ROM 512 是只读存储设备，其具体组件为 ROM、CD-ROM 和 DVD 等。ROM 512 保存将由 CPU 514 运行的程序。

[0117] 其具体组件为面板或者远程控制器的输入单元 513，接收来自用户的输入。图 11 示出了在以面板形式配置输入单元 513 的情况下输入单元 513 的示例。1100 是面板，其与图 6 中示出的面板单元 603 相对应，该面板 1100 由七个按钮组成：向上光标按钮 1101、向下光标按钮 1102、向左光标按钮 1103、向右光标按钮 1104、OK 按钮 1105、取消按钮 1106 和 EPG 按钮 1107。当用户按下一个按钮时，该被按下的按钮的标识符就会通知给 CPU 514。

[0118] CPU 514 运行保存在 ROM 512 中的程序。根据来自该程序的将被运行的指令，CPU 514 控制 QAM 解调单元 501、QPSK 解调单元 502、QPSK 调制单元 503、POD 504、TS 解码器 505、显示器 509、辅助存储单元 510、主存储单元 511 以及 ROM 512。

[0119] 图 12 是示出了保存在 ROM 512 中并由 CPU 514 运行的程序的示例结构的示意图。

[0120] 程序 1200 由多个个子程序组成。更为具体地，程序 1200 由 OS1201、EPG 1202、JavaVM 1203、业务管理器 1204 和 Java 库 1205 组成。

[0121] OS 1201 是当终端装置 500 被开启电时将由 CPU 514 激活的子程序。OS 1201 是操作系统的缩写，操作系统的示例是 Linux 等。OS 1201 是由用于与另一程序并行地运行一个子程序的内核 1201a 和库 1201b 所组成的公知技术的通用名称，因此省略它的详细说明。在本实施例中，OS 1201 的内核 1201a 运行 EPG 1202 和 JavaVM1203 作为子程序。同时，库 1201b 向这些子程序提供控制终端装置 500 的组件所需的多个功能。

[0122] 在这里，引入调谐作为这类功能的一个例子。对于调谐功能，从另一子程序中接收包含频率的调谐信息且随后将其传递到 QAM 解调单元 501。因此，QAM 解调单元 501 基于所提供的调谐信息来执行解调并将该已解调的数据传递给 POD 504 是可能的。结果，其余子程序能够经由库 1201b 来控制该 QAM 解调单元。

[0123] EPG 1202 由用于向用户显示节目列表以及接收来自用户的输入的节目显示单元 1202a 和用于选择频道的再现单元 1102b 组成。这里，EPG 是电子节目向导的缩写。当终端装置 500 被通电时 EPG1202 被激活。在该已激活的 EPG 1202 中，节目显示单元 1202a 等候经由终端装置 500 的输入单元 513 来自用户的输入。这里，在输入单元 513 采用图 11 中举例说明的面板形式的情况下，当用户按下输入单元 513 上的 EPG 按钮 1107 时，CPU 514 被告知该 EPG 按钮的标识符。作为运行在 CPU 514 上的子程序的 EPG 1202 的节目显示单元 1202a，接收这个标识符，并在显示器 509 上显示节目信息。图 13A 和 13B 示出了在显示器

509 上显示的节目表的示例。参见图 13A，以表格的形式在显示器 509 上显示节目信息。列 1301 描述了时间信息。列 1302 描述了频道名称“频道 1”和在与列 1301 中所描述的各个时间相对应的时间段期间将要广播的节目。示出了在“频道 1”上从 9:00 到 10:30 广播节目“新闻 9”，以及从 10:30 到 12:00 广播“电影 AAA”。与列 1302 的情形一样，列 1303 描述了 频道名称“频道 2”和在与列 1301 中所描述的时间相对应的时间段期间将要广播的节目。从 9:00 到 11:00 广播节目“电影 BBB”，从 11:00 到 12:00 广播“新闻 11”。1330 是一个光标。当按下面板 1100 上的向左光标 1103 或向右光标 1104 时，光标 1330 将移动。当在图 13A 中举例说明的状态中按下向右光标 1104 时，光标 1330 如图 13B 中所示的向右移动。同时，当在图 13B 中举例说明的状态中按下向左光标 1103 时，光标 1330 如图 13A 所示的向左移动。

[0124] 当在图 13A 中所示的状态中按下面板 1100 上的 OK 按钮 1105 时，节目显示单元 1202a 向再现单元 1102b 通知“频道 1”的标识符。同时当在图 13B 中所示的状态中按下面板 1100 上的 OK 按钮 1105 时，节目显示单元 1202a 向再现单元 1102b 通知“频道 2”的标识符。

[0125] 此外，节目显示单元 1202a 周期性地经由 POD504 将来自首端 101 的将要显示的节目信息保存到主存储单元 511 中。通常，需要花费时间从首端中获取节目信息。然而，通过在按下输入单元 513 的 EPG 按钮 1107 时显示预先保存在主存储单元 511 中的节目信息来快速显示节目表变得可能。

[0126] 再现单元 1102b 使用接收的频道标识符来再现频道。频道标识符和频道之间的关系由辅助存储单元 510 预先保存为频道信息。图 14 示出了保存在辅助存储单元 510 中的频道信息的一个示例。以列表的形式保存频道信息。列 1401 描述频道的标识符。列 1402 描述频道名称。列 1403 描述调谐信息。在这里，调谐信息由提供给 QAM 解调单元 501 的诸如频率、传送速率和编码比率这样的值来表示。列 1404 描述节目编号。节目编号是用于标识由 MPEG2 标准所定义的 PMTs 的号码。关于 PMT 的描述将在稍后给出。行 1411 ～ 1414 的每一个指示每个频道的标识符、频道名称和调谐信息的集合。行 1411 描述了包括以“1”作为标识符、“频道 1”作为频道名称、频率“312MHz”作为调谐信息以及“101”作为节目编号的一个集合。再现单元 1102b 将接收到的频道的标识符直接传递给业务管理器以再现该频道。

[0127] 此外，当再现正在发生的同时用户按下面板 1100 上的向上光标 1101 和向下光标 1102 时，再现单元 1102b 经由 CPU 514 从输入单元 513 接收关于用户的该按压的通知，并将正被再现的频道切换到另一频道。首先，再现单元 1102b 在主存储单元 511 中保存当前被再现的频道的标识符。图 15A、B 和 C 示出了保存在主存储单元 511 中的频道的标识符的示例。图 15A 示出标识符“3”被存储，并且参照图 14 示出了正在再现频道名称为“TV3”的频道。当用户在图 15A 中所举例说明的状态中按下向上光标 1101 时，再现单元 1102b 查阅图 14 中所示的频道信息，并将频道名称为“频道 2”的频道的标识符“2”传递给服务管理器，以最新再现频道名称为“频道 2”的频道，该频道是表中的前一个频道。同时，再现单元 1102b 将主存储单元 511 中保存的标识改写为频道标识符“2”。图 15B 示出了该已改写的频道标识符。同时，当用户在图 15A 中所举例说明的状态中按下向下光标 1102 时，再现单元 1102b 查阅图 14 中所示的频道信息，并将频道名称为“TV Japan”的频道的标识符“4”传递给服

务管理器，以最新再现频道名称为“TV Japan”的频道，该频道是表中的下一个频道。同时，再现单元 1102b 将主存储单元 511 中的标识改写为频道标识符“4”。图 15C 示出了该已改写的频道标识符。

[0128] JavaVM 1203 是一种顺序地分析并运行以 Java(TM) 语言编写的程序的 Java 虚拟机。以 Java 语言编写的程序被编译成不依赖于硬件的被称为字节码的中间代码。Java 虚拟机是运行该字节码的解释器。某些 Java 虚拟机将字节码翻译成 CPU 514 能够解释的可运行形式，并将结果传递给运行它的 CPU 514。当内核 1201a 指定将被运行的 Java 程序时，JavaVM 1203 被激活。在本实施例中，内核 1201a 将业务管理器 1204 指定为将被运行的 Java 程序。包括“JavaLanguage Specification”(ISBN 0-201-6345 1-1) 在内的许多书中都给出了 Java 语言的详细解释。因此，这里省略它的详细描述。同时，包括“Java Virtual Machine Specification”(ISBN 0-201-63451-X) 在内的 许多书中都给出了关于 Java VM 本身的操作的详细解释。因此这里省略关于它的详细描述。

[0129] 作为以 Java 语言编写的 Java 程序的业务管理器 1204，由 JavaVM 1203 顺序地运行。通过 JNI(Java 本地接口）业务管理器 1204 调用另一不是以 Java 语言编写的子程序或被其调用是可能的。包括“Java Native Interface”(ISBN 0-201-63451-X) 在内的许多书中都给出了关于 JNI 的解释。因此这里省略关于它的详细描述。

[0130] 业务管理器 1204 通过 JNI 从再现单元 1102b 中接收频道的标识符。

[0131] 首先，业务管理器 1204 将频道的标识符传递给 Java 库 1205 中的调谐器 1205c，以请求调谐。调谐器 1205c 查阅保存在辅助存储单元 510 中的频道信息以获取调谐信息。假设业务管理器 1204 把频道的标识符“2”传递给调谐器 1205c，那么调谐器 1205c 就查阅图 14 中所示的列 1412，并且获取与该频道相对应的调谐信息“156MHz”。调谐器 1205c 经由 OS 1201 的库 1201b 将该调谐信息传递给 QAM 解调单元 501。QAM 解调单元 501 根据给它的调谐信息对从首端 101 发送来的信号进行解调，并将结果信号传递给 POD504。

[0132] 接下来，业务管理器 1204 请求 Java 库 1205 内部的 CA 1205d 执行解扰。CA 1205d 通过 OS 1201 中的库 1201b 向 POD 504 提供解扰所需的信息。基于该提供的信息，POD 504 对 QAM 解调单元 501 所提供的信号进行解扰，并将结果信号传递给 TS 解码器 505。

[0133] 接下来，业务管理器 1204 向 Java 库 1205 内部的 JMF 1205a 提供频道的标识符，以请求再现视频和音频。

[0134] 首先，JMF 1205a 从 PAT 和 PMT 中获取用于指定将要再现的视频和音频的分组 ID。PAT 和 PMT 是由 MPEG-2 标准所定义的表格，其示出了包含于 MPEG2 传输流中的节目队列。PAT 和 PMT 与音频和视频一起被承载于 MPEG2 传输流中所包含的分组的载荷中。请参考详细描述 PAT 和 PMT 的规范。这里，仅给出了 PAT 和 PMT 的概述。作为节目关联表的缩写的 PAT，承载于分组 ID 为 “0”的分组中。为了获取 PAT，JMF 1205a 通过 OS 1201 的库 1201b 向 TS 解码器 505 指示分组 ID “0”和 CPU 514。然后，TS 解码器 505 基于分组 ID “0”执行过滤，并将结果传递给 CPU 514。因此，JMF 1205a 能够收集 PAT 分组。图 16 举例说明了示意性地示出已收集的 PAT 信息的一个示例的表格。列 1601 描述了节目编号。列 1602 描述了分组 ID。列 1602 中所示的分组 ID 用来获取 PAT。行 1611 ～ 1613 的每一个是一对频道的节目编号和与其相应的分组 ID。这里，定义了三个频道。行 1611 定义了一对节目编号“101”和分组 ID “501”。假设提供给 JMF 1205a 的频道标识符是“2”，则 JMF 1205a 就

参阅图 14 中的列 1412 以获取与该频道标识符相应的节目编号“102”，然后查阅图 16 所示的 PAT 中的行 1612 以获取与节目编号“102”相应的分组 ID“502”。作为节目映射表的缩写的 PMT，承载于具有在 PAT 中所指定的分组 ID 的分组中。为了获取 PMT，JMF 1205a 通过 OS 1201 的库 1201b 向 TS 解码器 505 指示分组 ID 和 CPU 514。这里，所指定的分组 ID 为“502”。然后，TS 解码器 505 基于分组 ID“502”执行过滤，并将结果传递给 CPU 514。因此，JMF 1205a 能够收集 PMT 分组。图 17 举例说明了示意性地示出已收集的 PMT 信息的表格。列 1701 描述流类型。列 1702 描述分组 ID。在各个流类型中所指定的信息承载于具有在列 1702 中所指定的分组 ID 的分组的载荷中。列 1703 描述附加信息，行 1711 ～ 1714 的每一个是一对要传送的被称为基本流的信息的分组 ID 和类型。作为一对流类型“音频”和分组 ID“5011”的行 1711，指示该音频数据保存在分组 ID 为“5011”的分组的载荷中。JMF 1205a 从 PMT 中获取将要再现的视频和音频的分组 ID。参见图 17，JMF 1205a 从行 1711 中获取音频分组 ID“5011”，以及从行 1712 中获取视频分组 ID“5012”。

[0135] 然后，JMF 1205a 经由 OS 1201 的库 1201b 向 TS 解码器 505 提供多对获取的音频分组 ID 和作为输出目的地的音频解码器 506，以及视频分组 ID 和作为输出目的地的视频解码器 508。TS 解码器 505 基于该提供的分组 ID 和输出目的地执行过滤。这里，分组 ID 为“5011”的分组被传递给音频解码器 506，分组 ID 为“5012”的分组被传递给视频解码器 508。音频解码器 506 对所提供的分组执行数模转换以经由扬声器 507 再现音频。视频解码器 508 对所提供的分组执行数模转换以在显示器 509 上显示视频。

[0136] 最后，业务管理器 1204 向 Java 库 1205 中的 AM 1205b 提供频道标识符，以请求数据广播再现。这里，数据广播再现指的是提取包含于 MPEG2 传输流中的 Java 程序，并使 JavaVM 1203 运行它。作为将 Java 程序嵌入到 MPEG2 传输流中的技术，使用称为 DSMCC 的方法，其在 MPEG 规范 ISO/IEC138181-6 中描述了。这里省略了 DSMCC 的详细说明。DSMCC 规范定义了一种方法，用于在 MPEG2 传输流中以分组的形式对计算机所使用的包含目录和文件的文件系统进行编码。关于将要运行的 Java 程序的信息以 AIT 的格式承载于 MPEG2 传输流中的分组中。AIT 是应用信息表的缩写，其定义在 DVB-MVP 标准（正式被称为 ETSI TS 101 812 DVB-MHPspecification V1.0.2）的第十章中给出。

[0137] 首先，为了获取 AIT，与 JMF 1205a 的情形一样，AM 1205b 获取 PAT 和 PMT，以得到保存 AIT 的分组的分组 ID。假设“2”是所提供的频道标识符，且在图 16 中示出的 PAT 以及图 17 中示出的 PMT 将要被传送，那么 AM 1205b 根据 JMF 1205a 所遵循的相同步骤来获取图 17 中示出的 PMT。接下来，AM 1205b 从该 PMT 中提取其流类型为“数据”且具有作为附加信息的“AIT”的基本流的分组 ID。如图 17 所示，行 1713 中的基本流对应该基本流，因此，AM 1205b 从它那里获取分组 ID“5013”。

[0138] AM 1205b 将 AIT 的分组 ID 和作为输出目的地的 CPU 514 经由 OS 1201 的库 1201b 提供给 TS 解码器 505。然后，TS 解码器 505 基于该提供的分组 ID 执行过滤，并将结果传递给 CPU 514。因此，AM 1205b 能够收集 AIT 的分组。图 18 举例说明了示意性地示出已收集的 AIT 信息的表格。列 1801 描述 Java 程序的标识。根据 MHP 规范，这些标识定义为应用 ID，以标识 Java 程序是否应该被终端装置 500 的安全管理器 1205f 验证。当标识符值处于 0x0 到 0x3fff 的范 围内时，不需要验证，而当标识值处于 0x4000 到 0x7ffff 的范围内时，需要验证。其标识符值落入前者范围内的 Java 程序被称为“未签名程序”，而其标识符值落

入后者范围内的 Java 程序被称为“已签名程序”。列 1802 描述了用于控制 Java 程序的控制信息。该控制信息包括“autostart”、“present”和“kill”。“autostart”指的是终端装置 500 马上自动运行程序。“present”指的是该程序不被自动运行。“kill”指的是该程序将被终止。列 1803 描述了用于提取包含有 DSMCC 格式的 Java 程序的分组 ID 的 DSMCC 标识符。列 1804 描述了 Java 程序的程序名字。行 1811 和 1812 的每一个是关于 Java 程序的信息集合。行 1811 中所定义的 Java 程序是标识符“301”、控制信息“autostart”、DSMCC 标识符“1”以及程序名称“a/TopXlet”的集合。行 1812 所定义的 Java 程序是标识符“302”、控制信息“present”、DSMCC 标识符“1”以及程序名称“b/GameXlet”的集合。这里，这两个 Java 程序具有相同的 DSMCC 标识符。这表示两个 Java 程序包括在已经按照相同的 DSMCC 方法进行了编码的文件系统中。这里，仅为各个 Java 程序指定了四种信息，但实际上可以指定更多信息。详情请参见 DVB-MHP 规范。

[0139] AM 1205b 从 AIT 中找出“autostart”Java 程序，并提取相应的 DSMCC 标识符和 Java 程序名称。参见图 18，AM 1205b 提取行 1818 中的 Java 程序，并且获取 DSMCC 标识符“1”以及 Java 程序名称“a/TopXlet”。

[0140] 接下来，AM 1205b 使用从 AIT 获取的 DSMCC 标识符，从 PMT 中获取保存 DSMCC 格式的 Java 程序的分组的分组 ID。更具体而言，AM 1205b 从 PMT 中获取流类型为“数据”且附加信息中的 DSMCC 标识符相匹配的基本流中所包含的分组 ID。

[0141] 这里，假定该 DSMCC 标识符为“1”，且 PMT 为图 17 中所示，那么行 1714 中的基本流满足上述条件。因此，分组 ID “5014”将被提取。

[0142] AM 1205b 将嵌入有 DSMCC 格式的数据的分组的分组 ID 和作为输出目的地的 CPU 514，经由 OS 1201 的库 1201b 指示给 TS 解码器 505。这里，提供分组 ID“5014”。然后，TS 解码器 504 基于所提供的分组 ID 执行过滤，并将结果传递给 CPU 514。因此，AM1205b 能够收集所需的分组。AM 1205b 根据 DSMCC 方法从收集到的分组重建文件系统，并将该重建的文件系统保存到主存储单元 511 中。在下文中将用于从 MPEG2 传输的分组中提取诸如文件系统这样的数据并将该提取的数据存储到诸如主存储单元 511 这样的存储单元中的处理称为下载。

[0143] 图 19 示出了已下载的文件系统的示例。在该图中，圆圈代表目录，方框代表文件，其中 1901 是根目录、1902 是目录“a”、1903 是目录“b”、1904 是文件“TopXlet.class”，以及 1905 是文件“GameXlet.class”。

[0144] 接下来，AM 1205b 将已下载到主存储单元 511 的文件系统中的将要运行的 Java 程序传递给 JavaVM 1203。在这里，假定将要运行的 Java 程序名称为“a/TopXlet”，那么将“.class”附加到上述 Java 程序名称得到的文件“a/TopXlet.class”是将要运行的文件。“/”是目录和文件名称之间的分割符，并且如图 19 所示，文件 1904 是将要运行的 Java 程序。接下来，AM 1205b 将文件 1904 传递给 JavaVM 1203，由于描述该 Java 程序的标识符的列 1801 表示未签名程序，因此意味着不需要请求安全管理器 1205f 来对该 Java 程序执行验证。

[0145] JavaVM 1203 运行该接收的 Java 程序。

[0146] 在接收到另一频道的标识符时，业务管理器 1204 通过 Java 库 1205 中所包括的每个库来终止视频和音频的再现，以及终止正通过在该相同的 Java 库 1205 中所包括的每个

库而实现的 Java 程序的运行，然后基于该最新接收的频道标识符来执行视频和音频的再现以及 Java 程序的运行。

[0147] Java 库 1205 是保存在 ROM 512 中的多个 Java 库的集合。在本实施例中，Java 库 1205 包括 JMF 1205a、AM 1205b、调谐器 1205c、CA 1205d、POD 库 1205e、安全管理器 1205f 和下载的模块 1206 等。

[0148] 业务管理器 1204 和下载的模块 1206 经由 Java 库 1205 中所包含的 POD 库 1205e 与首端 101 进行双向通信。通过 POD 库 1205e 使用 QPSK 解调单元 502 和 QPSK 调制单元 503 经由 OS 1201 的库 1201b 和 POD 504 来实现这个双向通信。

[0149] 下载的模块 1206 能够通过这个通信从首端 101 中接收代码数据。代码数据指的是包含 X.509 证书和 / 或终端装置 500 的固件的二进制数据。图 37 是示出了代码数据的示意图，该代码数据仅描述了与本发明相关的部分。当接收到代码数据 37 时，如果包含有根证书 371，则下载的模块 1206 提取该证书，并将其传递给安全管理器 1205f。372 表示诸如固件这样的其它数据。

[0150] AM 1205b 从首端 101 接收与终端装置 500 应该保存在辅助存储单元 510 中的 Java 程序有关的信息。该信息被称作 XAIT 信息。该 XAIT 信息以任意形式在首端 101 和 POD 504 之间传送。只要包含有作为 XAIT 的所需信息，那么无论哪种传输格式，都可实现本发明。

[0151] 图 20 举例说明了示意性地示出从首端 101 获取的 XAIT 信息的示例的表格。列 2001 描述 Java 程序的标识符。列 2002 描述用于控制该 Java 程序的控制信息。该控制信息包括“autostart”和“present”。“autostart”表示当终端装置 500 被通电时自动运行程序，而“present”表示不自动运行程序。列 2003 描述用于提取包含有 DSMCC 格式的 Java 程序的分组 ID 的 DSMCC 标识符。列 2004 描述了 Java 程序的程序名称。列 2005 描述了 Java 程序的优先级。行 2011 和 2012 的每一个是关于各个 Java 程序的信息集合。在行 2011 中所定义的 Java 程序是标识符“0x7001”、控制信息“autostart”、DSMCC 标识符“1”以及程序名称“a/PPV1Xlet”的集合。从它的 Java 程序应用 ID 可以看出，这个 Java 程序是一个已签名程序。在这里，仅为各个 Java 程序指定了五种信息，但是即使定义了更多种信息，也能实现本发明。

[0152] 当接收到 XAIT 信息时，根据与用于从 AIT 信息中下载 Java 程序的步骤相同的步骤，AM 1205b 将来自 MPEG2 传输流的文件系统保存到主存储单元 511 中。在此之后，紧接着在 AM 1205b 将文件系统保存到辅助存储单元 510 之前，它向安全管理器 105f 发出一个预保存通知。此时，由根据本发明的安全管理器 1205f 发起验证操作，稍后描述其细节。一旦从安全管理器 1205f 获知激活已被启动，AM 1205b 将文件系统保存到辅助存储单元 510 中。接下来，AM 1205b 把 XAIT 信息和已下载文件系统的存储位置进行关联的结果保存到辅助存储单元 510 中。图 21 示出了相互关联地保存在辅助存储单元 510 中的 XAIT 信息和已下载文件系统的示例。这里，将 OSAP 规范中所定义的文件作为示例来描述。图 21 中与图 20 中的元件相同的元件相互是一样的，因此这里省略这些元件的说明。列 2101 保存已下载文件系统的存储位置。在图 21 中，该存储位置由箭头来表示。2110 是已下载文件系统，其中包含顶级目录 2111、目录“a”2112、目录“b”2113、文件“PPV1Xlet.calss”2114、文件“PPV2Xlet.class”2115、文件“ocap.hashfile”2116 ～ 2118、文件“ocap.certificate.1”2119 以及文件“ocap.signaturefile.1”2120。

[0153] 文件 2116 ～ 2118 是包含文件名称或目录名称以及相应哈希值的哈希文件。图 22A、22B 和 22C 是示出“ocap.hashfiles”的细节的示意图。图 22A 中的 221 示出了“ocap.hashfile”2116，图 22B 中的 222 示出了“ocap.hashfile”2117，以及图 22C 中的 223 示出了“ocap.hashfile”2118。存在于“/”目录 2111 中的“ocap.hashfile”221 在列 2211 中包括存在于相同目录 2111 中的“ocap.certificate.1”文件 2119、“ocap.signaturefile.1”文件 2120、“a”目录 2112 和“b”目录 2113。列 2212 表示了使用何种哈希算法来计算列 2213 中描述的每个值。与列 2211 中的文件或目录相关的列 2213 包含通过使用由列 2212 指定的算法计算得到的哈希值。目前，主要使用的哈希算法是 SHA1（安全哈希算法 1）和 MD5（消息摘要 5）。这些都是用于将具有任意长度的数据转换为固定长度字节值的公知算法，其具有如下特征：在原始分组被转换后预测该原始 分组是不可能的；并且他们用于检查文件是否已经被破坏或篡改。同时，哈希值是通过利用哈希算法而产生的伪随机数。当哈希算法是 SHA1 时，哈希值的长度是 20 字节，而当哈希算法是 MD5 时，哈希值的长度转换为 16 字节。对于有关 SHA1 和 MD5 的细节，请分别参考“FIPS-PUB 186-2 Secure Hash Standard”和“IETF RFC1321”。在这里，与列 2211 中所描述的各个目录“a”和“b”相对应的哈希值是已经分别为存在于“a”目录中的“ocap.hashfile”文件 2117 和存在于“b”目录中的“ocap.hashfile”文件 2118 计算的 SHA1 哈希值。

[0154] 与 221 中的“ocap.hashfile”情形一样，222 中的“ocap.hashfile”包括文件名、哈希算法以及存在于同一目录 2112 中的“PPV1Xlet.class”文件 2114 的哈希值。类似地，223 中所包括的是文件名、哈希算法和存在于同一目录 2113 下的“PPV2Xlet.class”文件 2115 的哈希值。

[0155] 在这里，只描述与本发明相关的属性，对于关于“ocap.hashfile”的详细信息，可参照 OCAP 规范“OpenCabel(TM)Application Platform Specifcation OCAP 1.0 Profile(OC-SP-OCAP 1.0-IF-I09-031121)”。

[0156] 文件 2119 是证书链。图 23 是示出了“ocap.certificate.1”文件 2119 的详细结构的示意图。描述“ocap.certificate.x”(x 是正整数）的典型结构的 231，包括根证书 2311、中间证书 2312 以及叶证书 2313。它们处于链状关系，其中例如根证书 2311 的持有者发布中间证书 2312，中间证书 2312 的持有者发布叶证书 2313。注意，根据 OCAP 规范，与签名文件“ocap.signaturefile.x”相关的证书链是具有相同值“x”的“ocap.certificate.x”。在图 21 的情况下，与“ocap.signaturefile.1”相对应的证书链是“ocap.certificate.1”。同样，根证书 2311、中间证书 2312 以及叶证书 2313 也以相同的 X.509 证书格式来配置。如 ITU-T 所推荐的，在信息和通信行业的各个领域中 X.509 证书被广泛地用作证书表现格式的实际标准。在图 23 中，虽然仅举例说明了三个证书，但存在多个中间证书的情况 也是有的。然而，在这种情况下，这些中间证书必须处于它们相互关联的链状状态中。

[0157] 图 24 是示出了 X.509 证书的结构的示意图。这里，仅举例说明描述本发明所需的属性。对于关于 X.509 证书的详细描述，请参见 IETF RFC3280 “Internet X.509 Public Key Infrastructure Certificateand CRLProfile”。241 表示 X.509 证书的属性区域，242 表示 X.509 证书的签名值。序列号 2411 表示了标识该证书的号码，签名算法 2412 表示用于确定签名值 242 的算法，本更新日期和时间 2413 表示 X.509 证书变得有效的日期和时间，下一更新日期和时间 2413 表示 X.509 证书期满的日期和时间，发布者名称 2415 表示发

布本 X.509 证书的管理机构名称，主体名称 2416 表示 X.509 证书的持有者，公开密钥 2417 表示主体名称 2416 的公开密钥，以及签名值 242 表示已经使用本 X.509 证书的发布者的私有密钥进行签名（加密）的值。在这个实施例中，虽然本更新日期和时间 2413 以及下一更新日期和时间 2414 需要日期和时间的信息，但是本更新日期和时间 2413 以及下一更新日期和时间 2414 并不总是需要时间信息。作为利用公开密钥和私有密钥的系统，公开密钥密码系统被广泛用于电子商务和其它领域。在公开密钥密码系统中，使用与用于加密明文的密钥不相同的密钥来解密已加密的文本。由于加密的密钥和解密的密钥是不同的，因此不可能根据解密的密钥来推测加密的密钥。这个加密的密钥对应于私有密钥，而这个解密的密钥对应于公开密钥。典型的公开密钥密码系统包括 RSA(Rivest-Shamir-Adleman) 和 DSA（数字签名标准）。

[0158] 文件 2120 是签名文件，图 25 是示出了“ocap.signaturefile.1”文件 2120 的示意图。251 表示标识相关的 X.509 证书的标识符，252 表示哈希签名算法，以及 253 表示签名值，该签名值已经使用 252 中所表示的哈希签名算法从“ocap.hashfile”2116 中计算得出。

[0159] 一旦 Java 程序被保存到辅助存储单元 510 中，即使在诸如频道改变和终端装置 500 的断电这样的原因导致该 Java 程序被从主存储单元 511 中删除的情况下，只要 AM 1205b 已经接收到图 20 中所示 的 XAIT 信息，那么不需要等待下载而激活该 Java 程序是可能的。换言之，在图 20 中，程序“/a/PPV1Xlet”的控制信息 2002 是“autostart”。因此，在图 21 的 2011 中，当搜索与“/a/PPV1Xlet”相对应的文件系统的存储位置 2101 并且然后文件 2114 被传递给 JavaVM 1203 时，激活保存在该文件系统中的 Java 程序“PPV1Xlet”。

[0160] 接下来，描述作为本发明的主要功能模块的安全管理器 1205f。

[0161] 安全管理器 1205f 从业务管理器 1204 中接收预保存通知，该预保存通知表示出图 20 的 2004 中所表示的“/a/PPV1Xlet”和“/b/PPVXlet2”将要被保存。当接收到该通知时，安全管理器 1205f 检查 Java 程序标识符 2001 的值以判定它是未签名程序还是已签名程序。在这里，因为该 Java 程序是一个已签名程序，因此安全管理器对低于“/”目录的文件系统进行验证。为了检验该文件系统，通过使用图 21 中所举例说明的 ocap.hashfile(2116 ～ 2118)、ocap.certificate.1(2119) 以及 ocap.signaturefile.1 来进行验证。

[0162] 图 26 示出了用于对文件系统进行验证的安全管理器 1205f 的组件。

[0163] 通知接收单元 261 用于紧接着 AM 1205b 将要保存之前接收预保存通知，以及用于将该事情通知给判断单元 262。

[0164] 判断单元 262 判断验证结果。它请求哈希计算单元 263 对文件系统进行哈希计算以接收哈希值。判断单元 262 从存在于“ocap.hashfile”文件内的哈希值 2213、2223 以及 2233 中提取出将要比较的值，并且检查它是否匹配该接收到的哈希值。如果它们不匹配，则判断单元 262 判定已被篡改，并且验证以失败结束。

[0165] 此外，判断单元 262 利用证书提取单元 265 来提取 X.509 证书的每一个，并且判断当前时间是否不在每一个 X.509 证书文件的本更新日期和时间 2313 之前，以及是否不在每一个 X.509 证书文件的下一更新日期和时间 2414 之后（即，当前时间在每一个 X.509 证书文件的本更新日期和时间 2313 与下一更新日期和时间 2414 之间）。当前日期和时间从 OS 1201 的库 1201b 中获得。如果有效期不满足“本更新日期和时间＜当前日期和时间＜下一

更新日期和时间”，则判断单元 262 判定验证失败。

[0166] 此外，为了验证证书链，判断单元 262 请求哈希计算单元 263 对每一个 X.509 证书的属性区域 241 进行哈希计算。然后，它请求签名值解密单元 264 进行计算用于对包含于每一个 X.509 证书中的签名值 242 进行解密，然后将得到的已解密值与由哈希值计算单元 263 所获取的哈希值进行比较，以检查证书链的状态。如果它们不匹配，则意味着证书不处于链状关系中，因此判定验证失败。同时，如果该值匹配且检验出证书处于链状关系中，则检查该证书链中的根证书是否被包含在终端装置 500 的辅助存储单元 510 中。如果不被包含，则判断单元 262 判定验证失败，认为执行比较是不可能的。

[0167] 当以下条件全都满足时，判断单元 262 判定验证成功：(1) 不存在篡改；(2) 在有效期内；(3) 证书处于链状关系中；以及 (4) 根证书匹配。

[0168] 当判断单元 262 请求计算每个文件的哈希值时，哈希计算单元 263 从 OS 1201 的库 1201b 中提取每个文件以对它们执行哈希计算，并将得到的值传递给判断单元 262。此外，哈希计算单元 263 从证书提取单元 265 中获取证书链 231 中的每个 X.509 证书，并对它们的每一个的属性区域 241 执行哈希计算。

[0169] 签名值解密单元 264 被判断单元 262 请求执行计算用于对每个 X.509 证书或“ocap.signaturefile.x”的签名值进行解密。当执行计算以对每个 X.509 证书的签名进行解密时，签名值解密单元 264 从证书提取单元 265 中获取证书链 231 中的每一个 X.509 证书，然后执行计算用于对它们的每一个的签名进行解密，并向判断单元 262 返回结果。

[0170] 证书提取单元 265 被判断单元 262、哈希计算单元 263 和签名值解密单元 264 请求以提取证书链 231 中的每一个 X.509 证书，并且提取和返回 X.509 证书。

[0171] 图 27 是一个概括当对文件系统执行验证时由安全管理器 1205f 执行的操作的流程图。基于这个流程图，描述在文件系统具有图 21 中所示的结构时所执行的操作。当从 AM1205b 中接收到文件系统的预保存通知时（步骤 S271），安全管理器 1205f 对比该文件系统的顶级“/”目录低的该文件系统进行篡改检查（步骤 S272）。在该篡改检查中，通过比较哈希值来检验出存在于该文件系统的每个目录中的文件没有损坏或改变。

[0172] 图 29 和图 30 是步骤 S272 的详细流程图。首先，如步骤 S291 中所示，为各个文件“ocap.certificate.1”和“ocap.signaturefile.1”以及存在于“/”目录中的各个目录“a”和“b”计算哈希值。注意，目录“a”和“b”的哈希值分别从“/a/ocap.hashfile”文件 222 和“/b/ocap.hashfile”文件 223 中计算得到。在步骤 S293 中，在步骤 292 中计算的哈希值与“/ocap.hashfile”中的 2213 中所描述的每个哈希值进行比较。在步骤 S294 中，如果所计算的哈希值的任何一个与 2213 中的哈希值都不相同，则判定已经被篡改（步骤 S297）。同时，当所有所计算的哈希值与 2213 中的哈希值都匹配时，则转到步骤 S295。在步骤 S295 中，检查是否存在未完成篡改检查的子目录。在当前阶段，目录“a”和“b”作为“/”目录中仍未执行篡改检查的子目录。因此，需要对这些目录“a”和“b”执行篡改检查。首先，在步骤 S296 中，焦点放在“a”目录上，其中执行与对“/”目录所执行的处理相同的处理。在完成“a”目录的篡改检查后，对“b”目录执行篡改检查。当对目录“a”和“b”的篡改检查都完成时，然后焦点放在“/”目录，并且执行图 30 中的步骤 S301 的处理。在步骤 S301 中，叶证书 2313 被从作为证书链 231 的“/ocap.certificate.1”文件 2119 中提取出来。然后，在步骤 S302 中，从所提取的叶证书 2313 中取得公开密钥 2417。接下来，在步骤 S303 中，

计算“/ocap.hashfile”文件 221 的哈希值。同时，在步骤 S304 中，使用存在于“/ocap.certificate.1”文件 2119 的叶证书 2313 中的公开密钥 2417 对“/ocap.signaturefile.1”文件 2120 中的签名值 242 进行解密。在步骤 S305 中，通过解密该签名值来检查在步 骤 S303 中所计算的哈希值是否等于在步骤 S304 中所取得的值。如果这些计算的值相匹配，那么可判定在“/”目录以下的文件系统未被篡改（步骤 S306）。同时，如果该计算的值不相匹配，则可判定该文件系统已被篡改（步骤 S307）。注意，虽然已经描述了以降序的方式从顶级“/”目录开始向着子目录的方向执行篡改检查，但是本发明并不局限于此。因此，可以以升序的方式从最低级目录开始向着顶级目录的方向来执行处理。通过上述处理，获取图 27 中步骤 S272 的结果。

[0173] 在步骤 S273 中，当步骤 S272 的结果为“已被篡改”时，判定验证已经失败且发出关于该事情的通告（步骤 S279），此后结束处理。当步骤 S272 的结果为“没有篡改”时，则执行步骤 S274 的处理。

[0174] 接下来，参见图 31 到图 33，给出了证书链验证的详细描述（步骤 S274）。假设首先检查中间证书 2312 和叶证书 2313，在图 31 中示出了其流程图。首先，从证书链 231 中提取出来中间证书 2313 和叶证书 2313（步骤 S311）。从该提取的叶证书 2313 中，提取本更新日期和时间 2413、下一更新日期和时间 2414 以及发布者名称 2415（步骤 S312）。其中，判断当前日期和时间是否在所述本更新日期和时间 2413 与下一个更新日期和时间 2414 之间，在这段时间期间证书保持有效（步骤 S313）。如果它超出了证书能够保持有效的期限，则证书链的验证以失败结束（步骤 S319）。同时，当判定它在该证书的有效期限内时，则提取中间证书 2312 中的主体名称 2416 和公开密钥 2417（步骤 S314），然后在中间证书 2312 的主体名称 2416 与叶证书 2313 的发布者名称 2415 之间进行比较，以判断中间证书 2312 与叶证书 2313 是否处于链状关系中（步骤 S315）。如果这些证书不处于链状关系中，那么证书链的验证失败。同时，当它们之间存在链状关系时，计算叶证书 2313 的属性区域 241 的哈希值（步骤 S316）。此外，使用中间证书 2312 的公开密钥 2417 对叶证书 2313 中的签名值 242 进行解密（步骤 S317）。当步骤 S316 和步骤 S317 都完成时，则检查各个步骤中得到的哈希值和已解密的签名值是否匹配（步骤 S318）。如果它们不相匹配，则证书链的验证以失败结束（步骤 S319）。

[0175] 接下来，在根证书 2311 与中间证书 2312 之间进行检查。图 32 是示出这个处理的流程图。从证书链 231 中提取出根证书 2311 和中间证书 2312（步骤 S321），然后对根证书 2311 和中间证书 2312 执行与对中间证书 2312 和叶证书 2313 所执行的检查相同的处理（步骤 S322 ～步骤 S328）。

[0176] 当在步骤 S328 中判定该值相匹配时，仅对根证书 2311 执行检查。图 33 是示出了仅对根证书 2311 执行检查的流程图。从在步骤 S321 中所提取出的根证书 2311 中，提取出本更新日期和时间 2413、下一更新日期和时间 2414，以及发布者名称 2415（步骤 S331）。其中，判断当前日期和时间是否在所述本更新日期和时间 2413 与下一更新日期和时间 2414 之间，在这段时间内证书保持有效（步骤 S332）。如果它超出了证书能够保持有效的期限，则证书链的验证以失败结束。同时，当判定它处于证书的有效期限内时，计算根证书 2311 的属性区域 241 的哈希值（步骤 S334）。此外，使用根证书 2311 的公开密钥 2417 对根证书 2311 的签名值 242 进行解密（步骤 S335）。当步骤 S334 和步骤 S335 都完成时，则检查在

各个步骤中得到的哈希值和已解密的签名值是否匹配（步骤S336）。假如它们相匹配，则证书链的验证成功（步骤S337），反之如果它们不相匹配，则证书链的验证以失败结束（步骤S338）。此时，步骤S274的处理结束。

[0177] 取决于步骤S274的结果，在步骤S275中执行不同的处理。当步骤S274的结果为“证书链的验证失败”时，判定验证已经失败，并且发出关于它的通知（步骤S279），随后，结束文件系统的验证。同时，在“证书链的验证成功”的情况下，则执行S276的处理。

[0178] 接下来，从终端装置500的辅助存储单元510中搜索与“/ocap.certificate.1”文件2119的根证书2311相同的证书（步骤S276）。当辅助存储单元510中不存在该相同证书时，在步骤S277中判定证书链231的验证失败，并且发出该验证失败的通知（步骤 S279），此后，该处理结束。同时，当包含有根证书2311时，则判定文件系统的验证成功，并且向AM 1205b发出关于验证成功的通知（步骤S278）。参见图28，即使在其后接收到Java程序的预激活通知（步骤S281），该处理仍什么都不执行而结束。

[0179] 在第一实施例中，当在某个时间之后将激活所保存的Java程序时，在那时不需要执行验证，因为紧接着在文件系统被保存之前已经验证了该文件系统。

[0180] 这里，将描述图34中所示的“应用描述文件”存在于文件系统中并且只有在其中所描述的文件将被保存的情形。例如，根据OCAP规范，以XML（可扩展标记语言）的格式来描述“应用描述文件”。图34示出了“应用描述文件”的一个示例。在图34中，没有图21中所示的“PPV2Xlet.class”2115的描述。因此，在这种情况下，“PPV2Xlet.class”2115没有作为存储目标被包括。在这种情况下，在步骤S292中没有为“PPV2Xlet.class”2115计算哈希值，因此在步骤S293中不会与在“ocap.hashfile”文件2118中所描述的2233内的哈希值进行比较。在步骤S294中，可以通过规定不被作为存储目标包括的文件在应用之外来转换到步骤S295的处理。

[0181] （第二实施例）

[0182] 当在文件系统被保存之后的某个时间将激活包含于该文件系统中的Java程序（PPV1Xlet.class 2114或者PPV2Xlet.class 2115）时，存在包括在“/ocap.certificate.1”文件2119中的X.509证书的其中一个的有效性期满的可能性（即，Java程序的激活日期和时间＞下一更新日期和时间2414）。然而，即使在证书链231中包含已经期满的X.509证书的情况下，第一实施例仍然允许Java程序被激活。

[0183] 因此，通过向第一实施例增加在激活Java程序时检验包括在证书链231中的每个证书2311、2312以及2313的有效性没有期满的功能，来实现本实施例。图26示出了本实施例中的组件。已经在第一个实施例中描述了本实施例必需的组件261-265，因此这里不给出其的描述。

[0184] 对于流程图，图27的流程图被图35的流程图替换并且增加了图36的流程图。

[0185] 参见图35，紧接着在文件被保存之前将要执行的处理（步骤S351到步骤S357）与在第一个实施例中所说明的处理（步骤S271到步骤S277）相同，因此省略其描述。如果验证没有失败，则该处理进行到图36中所示的流程图。当通知作为Java程序的PPV1Xlet.class2114在某个时间之后将被激活时（步骤S361），从“ocap.certificate.1”文件2119中提取出每一个X.509证书，即，根证书2311、中间证书2312以及叶证书2313（步骤S362）。然后，从叶证书开始到根证书逐一地选择该提取的X.509证书（步骤S363），并且检查当前

日期和时间是否处于每一个所选择的 X.509 证书的本更新日期和时间 2413 与下一更新日期和时间 2414 之间（步骤 S364）。如果当前日期和时间不处于本更新日期和时间 2413 与下一更新日期和时间 2414 之间，则判定验证失败，并且发出关于该事情的通知（步骤 S367）。在另一情况下，检查是否已经对所有的 X.509 证书执行了检查（步骤 S365）。如果对所有 X.509 证书还没有完成检查，那么处理返回到 S363，重复后续步骤。同时，当在步骤 S365 中已经检查所有 X.509 证书时，判定验证成功，并且发出关于验证成功的通知（步骤 S366），此后处理结束。通过增加图 36 的流程图所示的处理，向 AM 1205b 通知验证失败以便其有效期已经期满的 Java 程序不会被激活变得可能。当安全管理器 1205f 通知验证失败时，AM 1205b 在没有向 JavaVM1203 传递该 Java 程序的情况下中止所述激活。

[0186] （第三实施例）

[0187] 如第一实施例中所描述的，辅助存储单元 510 包括作为根证书的 X.509 证书，其与证书链 231 中的根证书 2311 进行比较。为了防备由剽窃等导致降低证书的可靠性的情形，保存在辅助存储单元 510 中的根证书被新的 X.509 证书替换（以下称作为证书替换）。新的 X.509 证书从首端 101 传送到终端装置 500，以经由下载的模块 106 传递给安全管理器 1205f。

[0188] 图 38A、38B 和 38C 的每一个是示出了辅助存储单元 510 中的根证书被安全管理器 1205f 替换（证书替换）的示意图。在这种情况下，证书 A381 是一个将被替换的旧证书，而证书 B382 是一个新证书。图 38A 中的 38-1 示出了在执行证书替换之前保存在辅助存储单元 510 中的证书，图 38B 的 38-2 示出了正在替换当中的证书，以及图 38C 的 38-3 示出了在执行证书替换后保存在辅助存储单元 510 中的证书。

[0189] 在第一和第二实施例中，即使在 Java 程序被保存之后执行证书替换，在 Java 程序的激活时间也没有考虑新证书。例如，考虑以下：当安全管理器 1205f 响应于它的预保存通知而正在验证 Java 程序时，证书链 231 中的根证书 2311 与证书 A3811 匹配；以及在证书 A381 被证书 B382 替换后，安全管理器 1205f 接收 Java 程序的预激活通知。在这个时候，辅助存储单元 510 不包括与证书链 231 中的根证书相匹配的任何证书，意味着该证书是不可靠的。然而，在第一和第二实施例中，由于紧接着在 Java 程序的激活之前没有在根证书之间进行比较（即，证书链 231 中的根证书 2311 没有与证书 B382 进行比较），因此，没有向 AM1205b 发出关于验证失败的通知。结果，AM1205b 使 Java 程序被激活。

[0190] 因此，本实施例增加在 Java 程序激活时由于证书替换而进行根证书比较的功能。

[0191] 图 26 示出了本实施例的组件。由于已经描述组件 261 ～ 265，因此省略其说明。增加了证书替换单元 266、证书替换规范单元 267 以及证书接收单元 268。

[0192] 当证书替换规范单元 267 判定比所接收到的证书旧的证书被保存在辅助存储单元 510 中时，证书替换单元 266 使用新的证书替换该旧的证书。同时，当证书替换规范单元 267 判定没有较旧的证书被保存时，证书替换单元 266 将新的证书保存到辅助存储单元 510 中。

[0193] 证书替换规范单元 267 接收由证书接收单元 268 所接收到的证书。然后，通过使用 OS 1201 的库 1201b，它检查保存在辅助存储 单元 510 中的证书，以查看是否存在其发布者相同并且比所接收到的证书旧的任何证书。

[0194] 当下载的模块 1206 从首端 101 接收新的证书时，证书接收单元 268 接收该新的证

书。当接收到该证书时，证书接收单元 268 将其传递给证书替换单元 266 和证书替换规范单元 267。

[0195] 此外，图 39 和图 40 随后被增加到图 35 的流程图。

[0196] 图 39 是执行证书替换时的流程图，而图 40 是执行证书替换之后激活 Java 程序时的流程图。参见图 39，当接收到证书替换的请求时（步骤 S391），提取出该接收的证书的发布者名称（步骤 S392）。检查在终端装置 500 的辅助存储单元 500 中是否存在将需要替换的旧证书（步骤 S393），且仅当存在旧证书时，删除该证书。然后，将该接收的证书保存到辅助存储单元 510 中（步骤 S395）。当在某个时间后接收到该 Java 程序的激活通知时（步骤 S401），在辅助存储单元 510 中搜索与证书链 231 中的根证书 2311 相匹配的证书（步骤 S402），如果存在任何证书（步骤 S403），则判定验证成功并且发出关于这个事情的通知（步骤 S404）。如果不存在相匹配的证书（步骤 S403），则判定验证失败，且发出关于这个事情的通知（步骤 S405）。注意，在步骤 S404 中判定验证成功之前，在检验证书链中的每个 X.509 证书满足“本更新日期和时间＜当前日期和时间＜下一更新日期和时间”后推断出验证是成功的也是可能的。

[0197] 此外，除了检查根证书是否匹配外，在 S402 之前执行图 31 ～图 33 所示的检查以查看证书链中的证书是否处于链状关系中之后来判定验证是成功的 / 不成功的也是可能的。

[0198] 此外，虽然对基于发布者名称来指定应当被替换的证书的情形给出了上述描述，但是，也可以基于诸如主体名称这样的另外属性值来指定证书。

[0199] （第四个实施例）

[0200] 当在文件系统被保存之后的某个时间将激活包括在该文件系统中的 Java 程序 (PPV1Xlet.class 2114 或者 PPV2Xlet.class 2115) 时，存 在一种情况，其中，由除了包括于“/ocap.certificate.1”文件 2119 中的任一 X.509 证书的有效期期满和根证书被替换之外的原因引起证书被撤销。这种配置即使存在被撤销的证书也允许 Java 程序被激活。

[0201] 这里，CRL（证书撤销列表）是一种广为人知的证书撤销者。图 41 是示出了 CRL 结构的示意图。这里，仅举例说明解释本发明所需的属性。对于有关 CRL 的更多详情，请参考 IETF RFC 3280“Internet X.509 Public Key Infrastructure Certificate and CRLProfile”。411 表示 CRL 的属性区域，412 表示签名值 413 的签名算法，以及 413 表示 CRL 的签名值。发布者名称 4111 表示这个 CRL 的发布者，本更新日期和时间 4112 表示该 CRL 生效的日期和时间，下一日期和时间 4113 表示该 CRL 的有效性期满的日期和时间，以及被撤销证书列表 4114 表示关于被撤销的 X.509 证书的信息。图 42 是示出了被撤销证书列表 4114 结构的示意图。这里，也仅举例说明解释本发明所需的属性。被撤销证书列表 4114 中存储了关于多个被撤销的 X.509 证书的信息。在图 42 的情况下，作为关于被撤销的“证书 A”421 的信息，包括了用于唯一标识了该证书的序列号 4211 和“证书 A”421 被撤销的日期和时间 4212。其它被撤销证书也与 421 相同。

[0202] 图 43 是包含有 CRL 的文件系统的一种示例配置。内部保存了“/”目录 432、“a”目录 432、“SimpleXlet.class”文件 433、“ocap.hashfile”文件 434 ～ 435、“ocap.certificate.1”文件 436、“ocap.signaturefile.1”文件 437、“ocap.crl.2”文件 438，以及“ocap.certificate.2”文件 439。对不包含 CRL 的文件系统的验证与第一实施例中所描

述的相同。因此，在本实施例中，焦点放在以 CRL 格式构造的“ocap.crl.2”文件 438 和作为该文件的证书链的“ocap.certificate.2”文件 439 上。注意，根据 OCAP 规范，“ocap.crl.x”的证书链是“ocap.certificate.x”。在图 43 的情况中，“ocap.crl.2”的证书链是“ocap.certificate.2”。

[0203] 图 46 是示出了“ocap.hashfile”文件 434 结构的示意图。461 示出了 ocap.hashfile 434 的详情。461 中存在于“/”目录 431 中的 ocap.hashfile，包含与存在于相同目录 431 中的每一个“ocap.certificate.1”文件 436、“ocap.signaturefile.1”文件 437、“a”目录 432、“ocap.crl.2”文件 438 以及“ocap.certificate.2”文件 439 相关的哈希值。

[0204] 图 44 是说明 CRL 的验证的流程图。下面将描述其中文件系统具有图 43 中所示的配置的示例。首先，从 CRL 中提取本更新日期和时间 4112 以及下一更新日期和时间 4113（步骤 S441），然后检查当前日期和时间是否在所述本更新日期和时间 4112 与下一更新日期和时间 4113 之间（步骤 S442）。如果不在，则判定该 CRL 无效（步骤 S447）。如果当前日期和时间在它们之间，则计算属性区域 411 的哈希值，以检验“ocap.crl.2”文件 438 的签名值（步骤 S443）。同时，从作为证书链的“ocap.certificate.2”文件 439 中提取叶证书 2313 的公开密钥 2417（步骤 S444），并且使用所提取的公开密钥 2417 对“ocap.crl.2”文件 438 的签名值 413 进行解密（步骤 S445）。然后，检查步骤 S443 中获取的哈希值是否等于步骤 S445 中获取的解密值（步骤 S446）。如果它们不相等，则判定该 CRL 无效（步骤 S447）。如果它们相等，参见图 45，那么对作为证书链的“ocap.certificate.2”文件 439 执行验证（步骤 S451）。验证证书链的方法与图 31 到图 33 中所示的方法相同，因此这里描述它。接下来，判断证书链的验证是否成功（步骤 S452），并且如果该验证失败，则判定该 CRL 无效（步骤 S456）。同时，如果该验证成功，则从辅助存储单元 510 中搜索与根证书相同的证书（步骤 S453）。在这里，如果没有匹配的根证书，则判定该验证失败以及该 CRL 无效（步骤 S456），反之如果存在相匹配的根证书，则判定该验证成功以及该 CRL 有效（步骤 S455）。

[0205] 下面描述了解决这一问题的方法，即尽管按照 CRL 证书被撤销了但 Java 程序仍被激活。为了支持这，本实施例增加以下功能，即当发出 Jave 程序的激活通知时，判断用于验证该 Jave 程序的证书是否为 CRL 中的被撤销证书。

[0206] 图 26 示出了本实施例的组件。除了对其做了一些增加的 262 和仍没有描述的 269 之外，对上面已经过描述的组件不再给出描述。

[0207] 还能验证 CRL 的判断单元 262，请求证书撤销规范单元 269 指定将由 CRL 撤销的证书。然后，当通知接收单元 261 接收到与证书撤销规范单元 269 所指定的被撤销证书相关的 Java 程序的预激活通知时，判断单元 262 判定验证失败。同时，当在判断单元 262 已经不能验证 CRL 且因此判定该 CRL 无效的状态下，通知接收单元 261 接收到 Java 程序的预激活通知时，判断单元 262 判定验证成功。

[0208] 当判断单元 262 认为该 CRL 的验证成功时，证书撤销规范单元 269 指定由证书提取单元 265 所提取的 X.509 证书中的哪一个证书是被撤销证书。

[0209] 对于流程图而言，增加图 47 和图 48。下面给出根据这些流程图的描述。假设现在发出图 21 中所示的文件系统的预保存通知，则启动图 35 的流程图中所示的处理，并且在

适当的时候完成步骤 S357 的处理。假设然后接收到图 43 中所示的另一文件系统的预保存通知，那么在完成图 44 的流程图中所示的处理后，执行步骤 S471 到步骤 S477。步骤 S471 到步骤 S477 的处理与步骤 S351 到 S357 的处理相同。当到达步骤 S478 并且如果“ocap.crl.2”文件 438 的验证（图 44 和 45 的流程图）成功时，关于该文件中所包含的被撤销证书的信息被写入到被撤销证书数据库中。图 49 是示出了该被撤销证书数据库的示意图。在列 491 中保存发布者名称，在列 492 中保存证书序列号，以及在列 493 中保存撤销的日期和时间。在这里，当接收到“PPV1Xlet.class”2114 的预激活通知时（步骤 S481），检查在“ocap.certificate.1”文件 2119 的证书链 231 中包含的 X.509 证书的任意一个，是否包括在被撤销证书数据库中（步骤 S483）。如果存在该证书的任意一个，则判定验证失败且发出关于此的通知（步骤 S486）。同时，当没有适用的证书时，对整个证书链进行检查（步骤 S484），且发出判定验证成功的通知（步骤 S485）。经由上述处理，通过对于其证书在检验时是有效的但在 Java 程序被激活时该证书被 CRL 撤销的文件系统，判断文件的验证失败，来解决该不应当被激活的 Java 程序被激活的问题。

[0210] 注意，在第一到第四个实施例中，当接收到 Jave 程序的预激活通知时，通过使用放置在每个目录中的“ocap.hashfile”来进一步进行检验以查看文件系统的树结构是否正确是可能的。

[0211] 此外，虽然为了简化的目的，在证书链中仅有一个中间证书，但是可以有多个中间证书。然而，当执行它的证书链的验证时，所有中间证书都必须处于链状关系中。

[0212] 此外，虽然已经以上述次序描述了以下处理，但是本发明并不限于该次序：检查存在 / 不存在篡改；验证证书链；检查以查看辅助存储单元是否包括与证书链中的根证书相同的根证书。

[0213] 此外，对于文件系统的保存，安全管理器 1205f 可以使用 OS 的库 1201b 来保存它。同样，第一到第四个实施例也可用于这样的情形，其中“应用描述文件”被放置在在文件系统的顶级目录“/”中，而就它的内容而言，仅将文件系统的一部分表示为将要保存的文件。因此，如果仅处理将要保存的文件，则不会出现问题。

[0214] 此外，虽然上面将程序描述为存储目标，但是除了程序之外的数据也可以是存储目标，意味着第一到第四个实施例也可应用到数据上。

[0215] 此外，存在多于一个“ocap.certificate.x”对应于“ocap.signaturefile.x”的可能性，在这种情况下，“ocap.certificate.x”文件的至少一个的验证需要是成功的。

[0216] 同样，虽然已经将“ocap.certifidcate.x”呈现为一个示例证书链，已经将“ocap.hashfile”呈现为具有哈希值的示例文件，以及已经将“ocap.signaturefile.x”呈现为一个示例文件用于检查在“/”目录中的“ocap.hashfile”是否已经被篡改，但是本发明并不局限于这些文件名称。

[0217] 而且，在认证失败的情况下，可以在重新下载之后再次进行认证。

[0218] 此外，在认证失败的情况下，所保存的程序连同已经用于验证的证书链、签名文件和哈希文件可以被删除，以保留足够的容量用于存储空间。

[0219] 在这里，描述了这样一种情形，其中组成程序的文件系统具有图 50 中所示的配置，但是如图 51 中所示的“应用描述文件”的情形那样，没有描述将用于验证的文件。图 50 中所示的 5011 到 5020 等同于图 21 中所示的 2111 到 2120。5021 表示“应用描述文

件”，其描述将要保存的文件。在图 51 的“应用描述文件”中，没有描述验证所需的“ocap.certificate.1”5019、“ocap.signaturefile.1”5020 和“ocap.hashfile”5017。在这种情况下，假如文件正如图 51 所示的那样被存储，那么执行验证所需的文件将不被保存。因此，在激活时不能执行在第二、三和四实施例中呈现的验证。当所保存的程序将要被激活，并且在示出该程序被保存之前的文件的图 50 中所示的文件已做好下载的准备，该保存的文件可以用作组成该程序的文件，以及用于验证的文件可以再次下载用于验证。

[0220] 然而，也可以存在这样的情况，其中示出程序被保存之前的文件的图 50 中所示的文件不能被下载。因此，即使在“应用描述文件”中没有描述它们，验证所需的文件也可以被保存以用于在程序激活时执行验证。

[0221] 尽管上面仅对本发明的一些示意性实施例进行了详细的描述，但是本领域的技术人员应当理解在没有本质上脱离本发明的新颖性教导和优点的情况下，在该示意性实施例中进行许多变形是可能的。因此，所有这样的变形都被认为是在本发明的范围之内。

[0222] 工业实用性

[0223] 根据本发明的能够保证程序可靠性和提高响应度的已验证程序的运行方法，对临时提高数字电视接收机的功能和将功能增加给它是很有用的。此外，本发明不仅可用于数字电视，而且也可用于诸如临时提高诸如个人电脑和移动电话这样的信息设备的功能和临时增加功能给该信息设备。

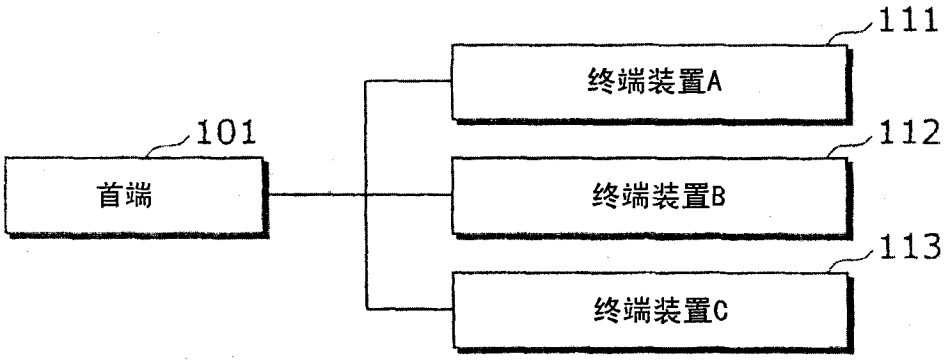


图1

频带	使用	调制技术
5～130MHz	带外（OOB） 首端和终端之间数据交换	QPSK
130～864MHz	带内 包含视频和音频的普通电视广播	QAM

图2

频带	使用
70～74MHz	从首端101到终端装置的数据传送
10.0～10.1MHz	从终端装置A111到首端101的数据传送
10.1～10.2MHz	从终端装置B112到首端101的数据传送
10.2～10.3MHz	从终端装置C113到首端101的数据传送

图3

频带	使用
150～156MHz	电视频道1
156～162MHz	电视频道2
⋮	⋮
310～311MHz	无线电频道1
⋮	⋮

图4

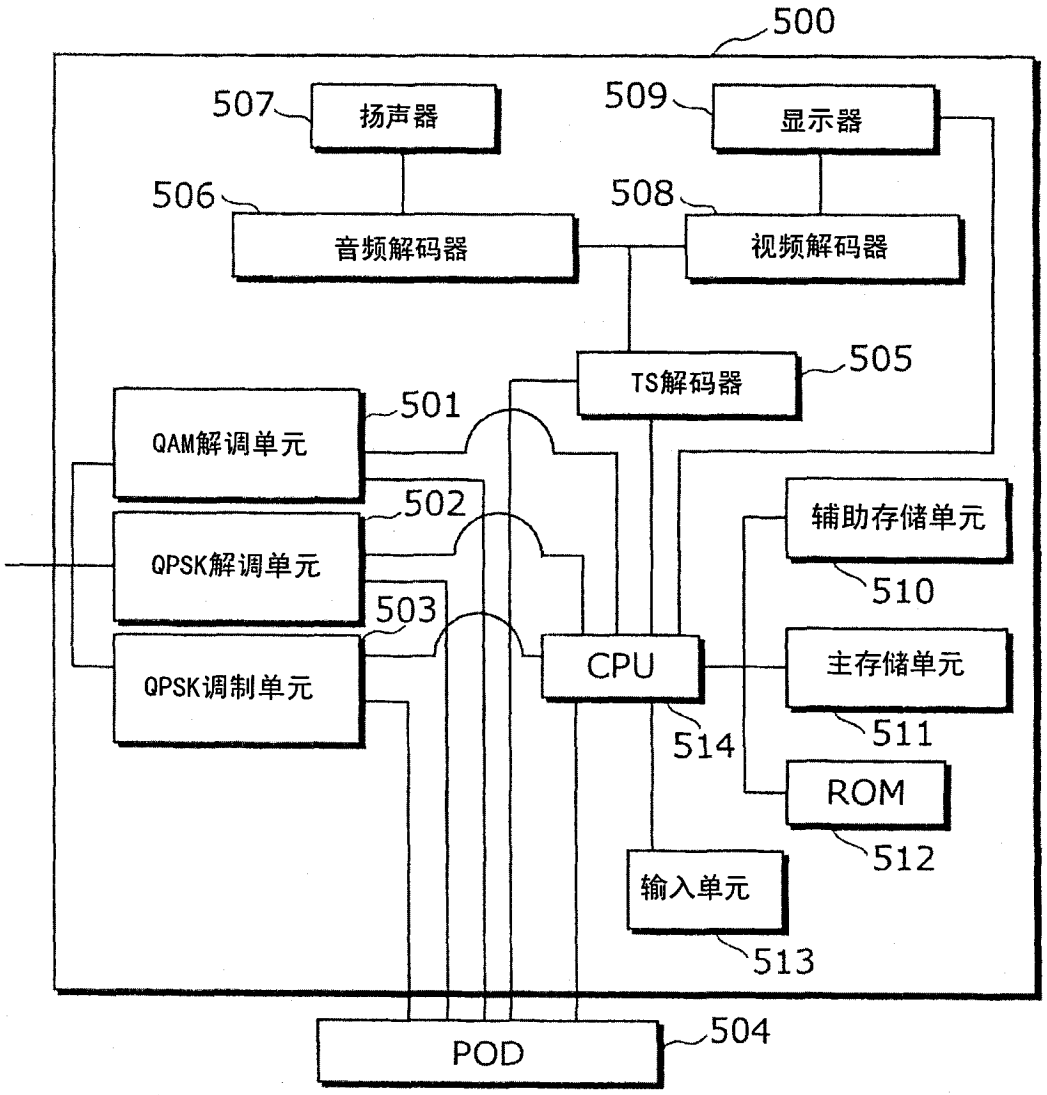

500
507
扬声器
509
显示器
506
音频解码器
508
视频解码器
505
TS解码器
501
QAM解调单元
502
QPSK解调单元
503
QPSK调制单元
辅助存储单元
510
CPU
主存储单元
511
514
ROM
512
输入单元
513
504
POD

图5

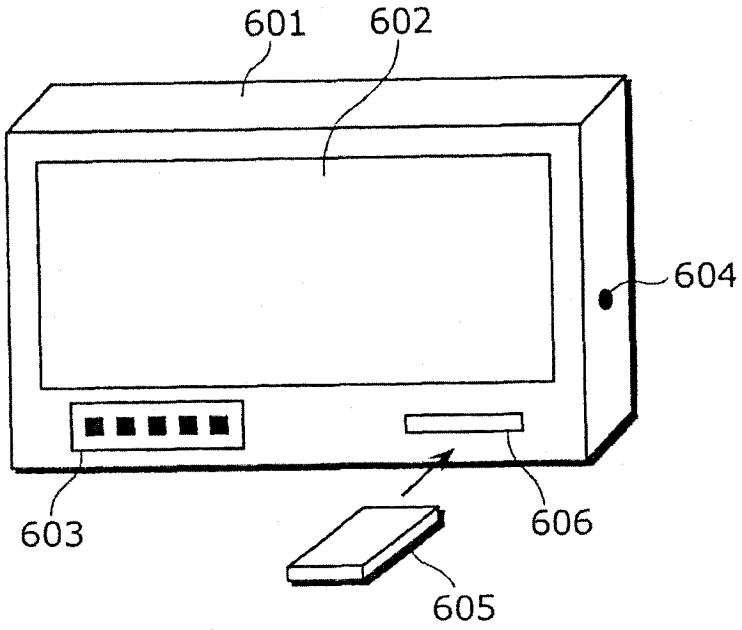

601
602
604
603
606
605

图6

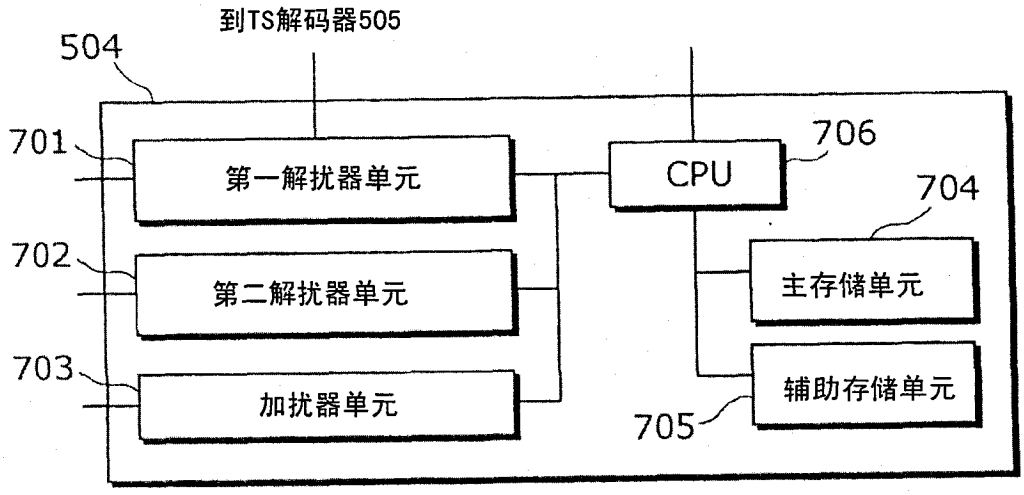

到TS解码器505
504
701
第一解扰器单元
702
第二解扰器单元
703
加扰器单元
CPU
706
704
主存储单元
辅助存储单元
705

图7

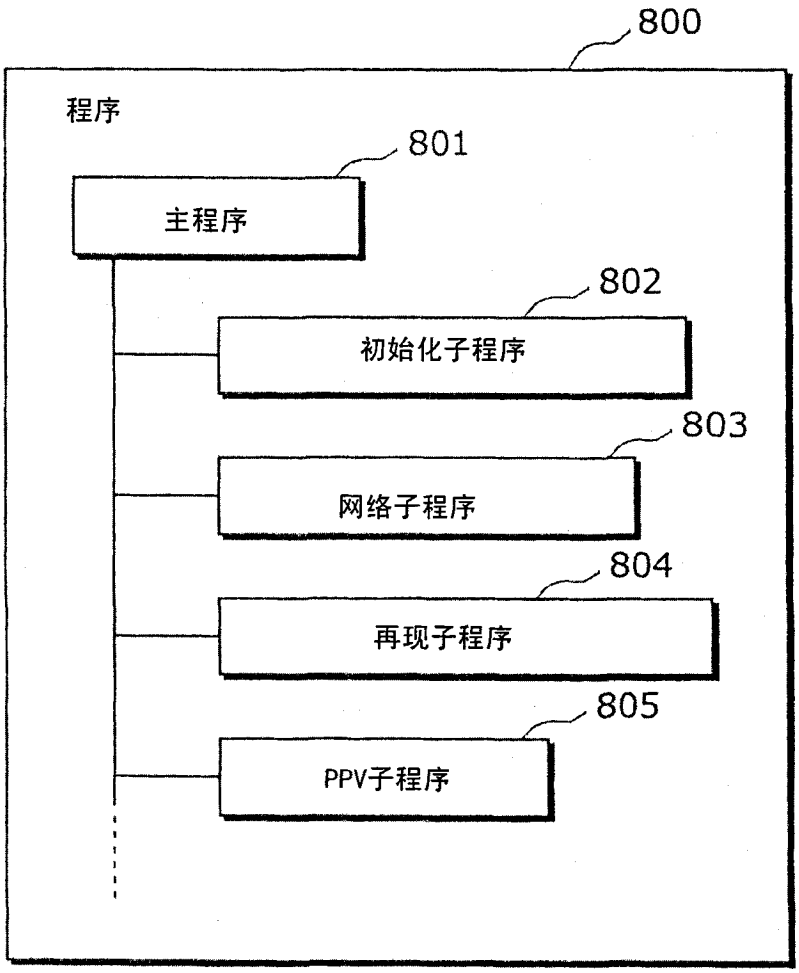


图8

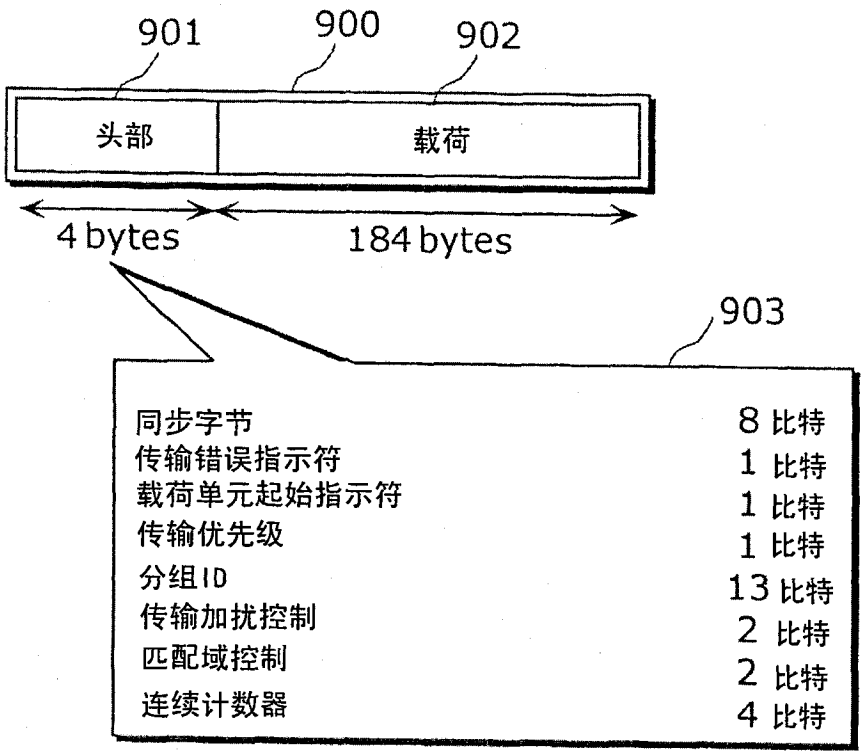

901
900
902
头部
载荷
4 bytes
184 bytes
903
同步字节 8 比特
传输错误指示符 1 比特
载荷单元起始指示符 1 比特
传输优先级 1 比特
分组ID 13 比特
传输加扰控制 2 比特
匹配域控制 2 比特
连续计数器 4 比特

图9

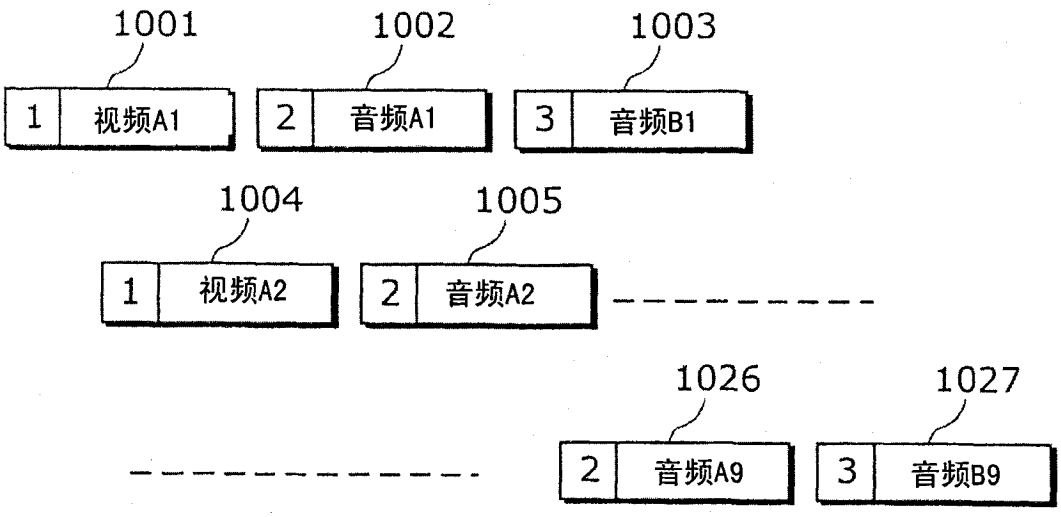

1001
1002
1003
1
视频A1
2
音频A1
3
音频B1
1004
1005
1
视频A2
2
音频A2
1026
1027
2
音频A9
3
音频B9

图10

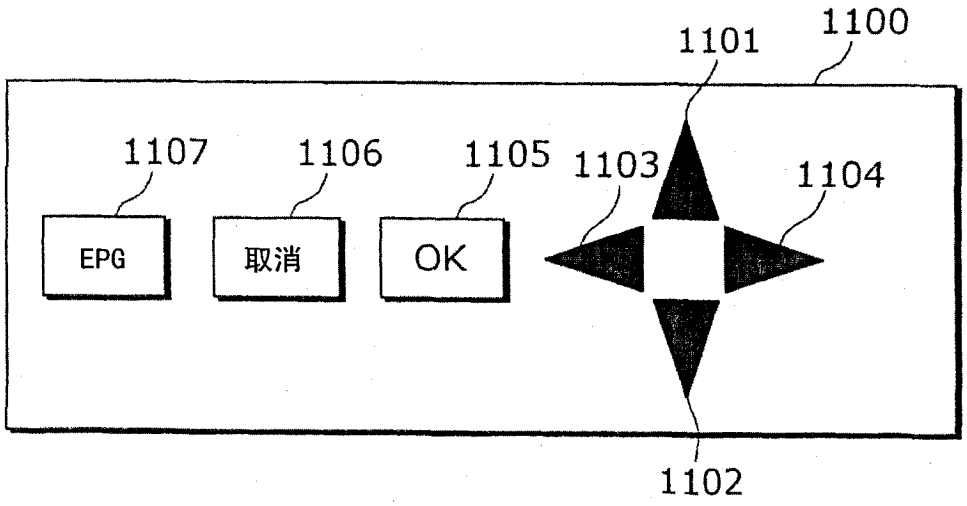

1101
1100
1107
1106
1105
1103
1104
EPG
取消
OK
1102

图11

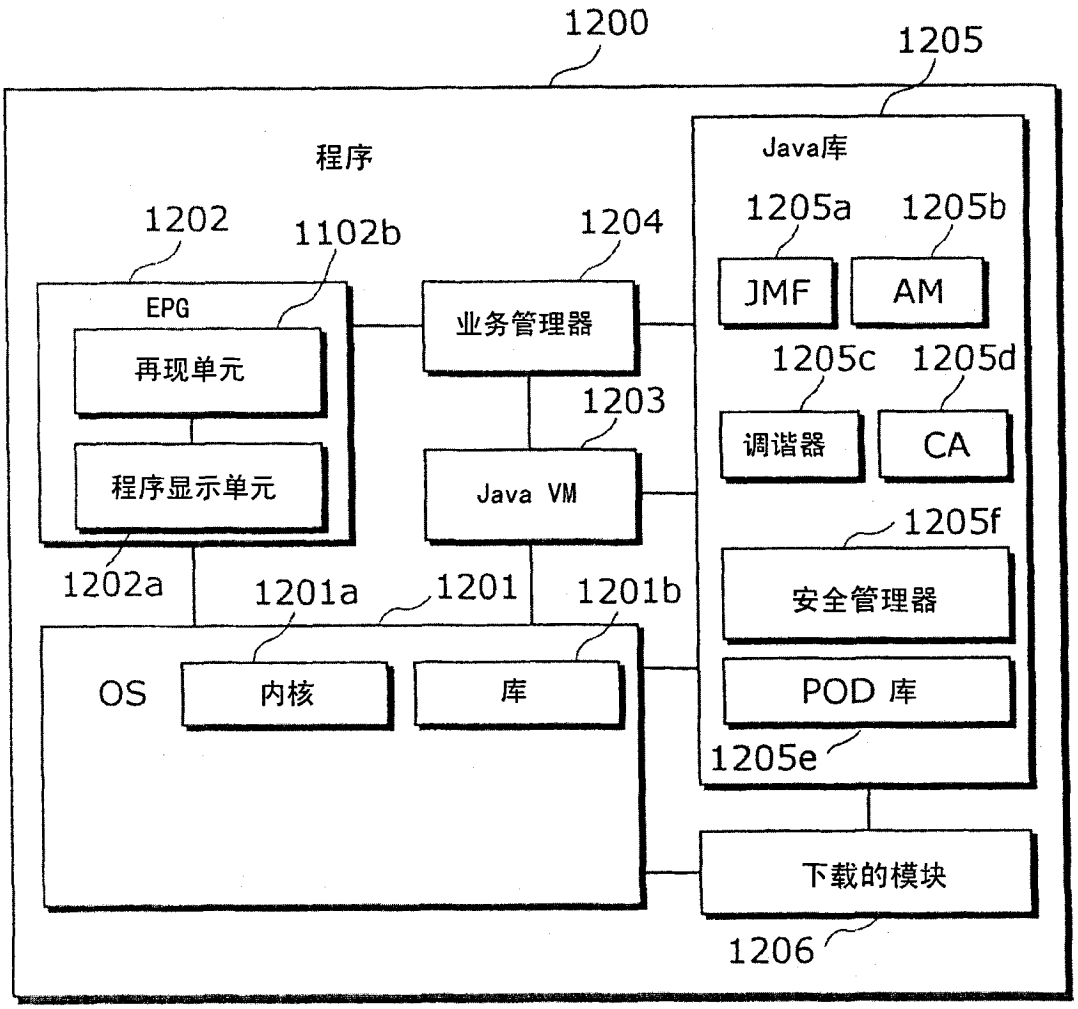

1200
1205
程序
Java库
1202
1102b
1204
1205a
1205b
EPG
再现单元
程序显示单元
业务管理器
JMF
AM
1205c
1205d
1203
调谐器
CA
Java VM
1205f
1202a
1201a
1201
1201b
安全管理器
OS
内核
库
POD 库
1205e
下载的模块
1206

图12

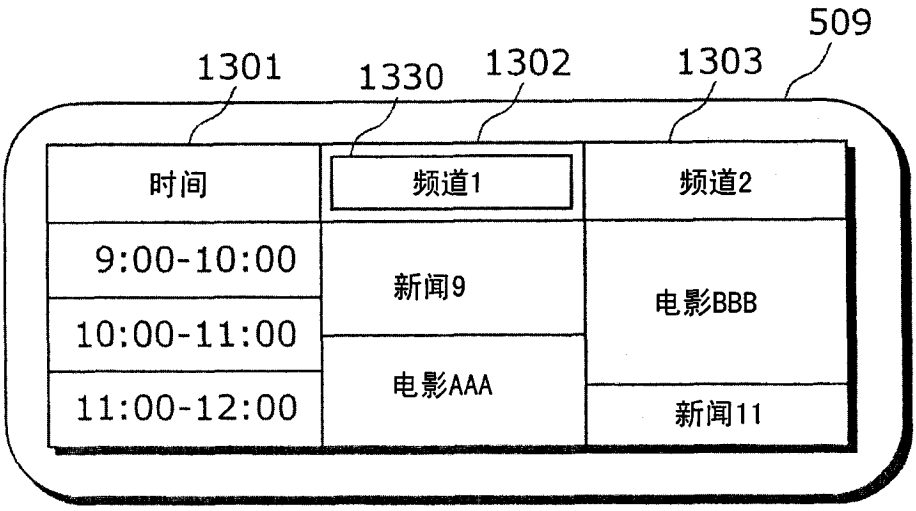

509
1301
1330
1302
1303
时间
频道1
频道2
9:00-10:00
新闻9
电影BBB
10:00-11:00
电影AAA
11:00-12:00
新闻11

图13A

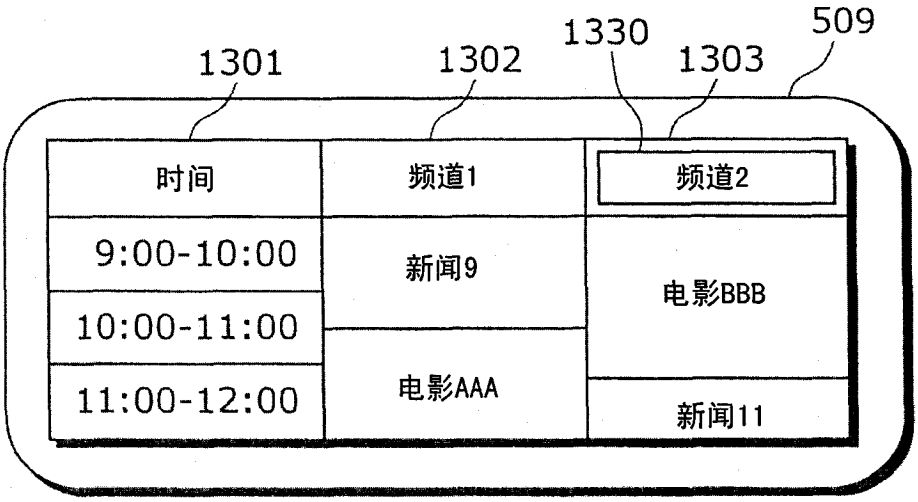

509
1301
1302
1330
1303
时间
频道1
频道2
9:00-10:00
新闻9
电影BBB
10:00-11:00
电影AAA
11:00-12:00
新闻11

图13B

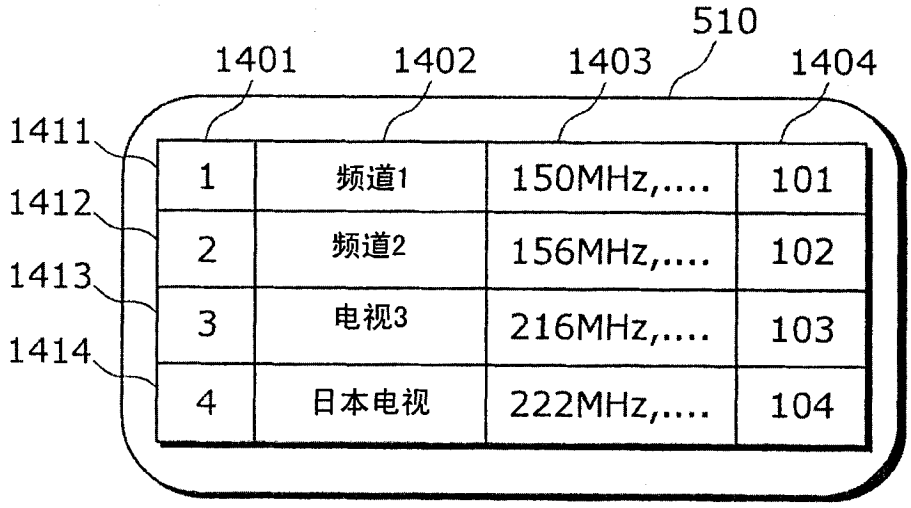


图14

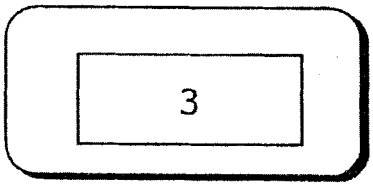


图15A

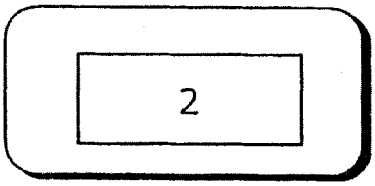


图15B

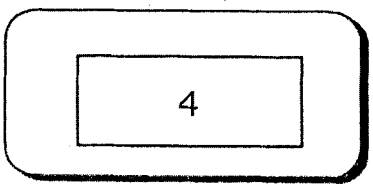


图15C

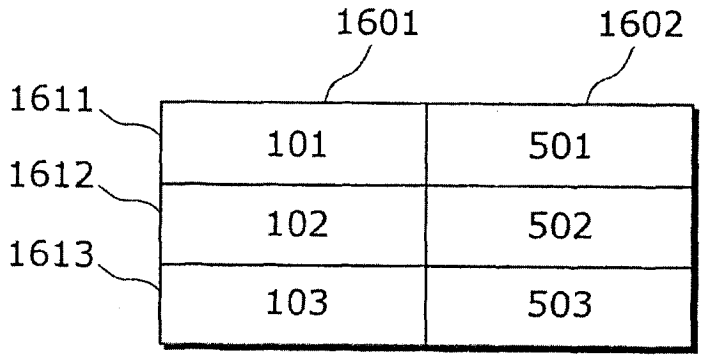

1601
1602
1611
1612
1613
101
501
102
502
103
503

图16

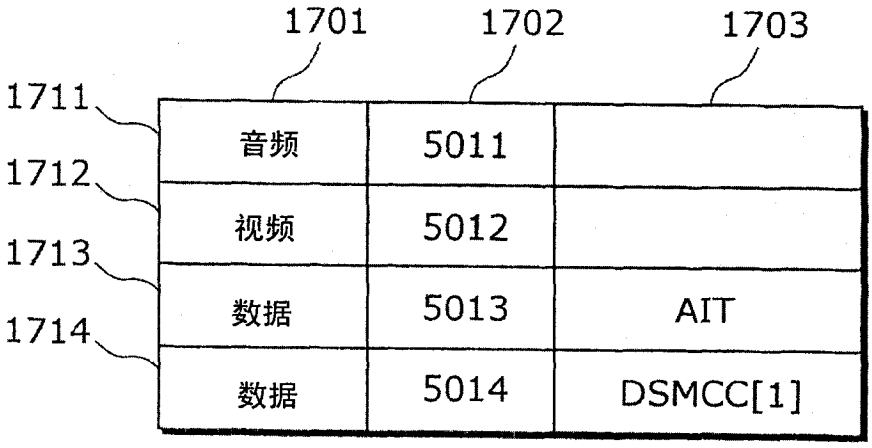

1701
1702
1703
1711
1712
1713
1714
音频
5011
视频
5012
数据
5013
AIT
数据
5014
DSMCC[1]

图17

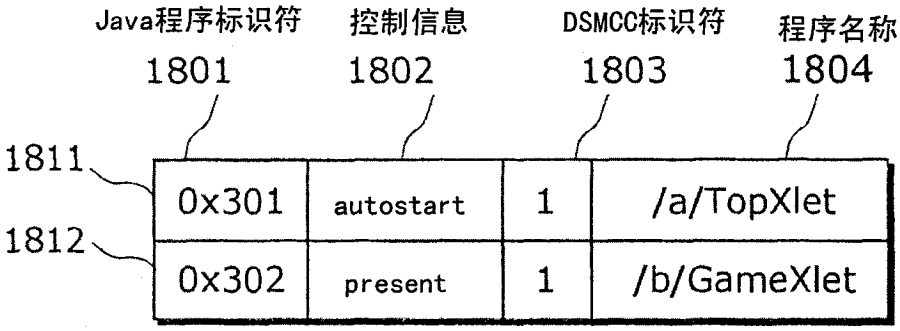

Java程序标识符
1801
控制信息
1802
DSMCC标识符
1803
程序名称
1804
1811
1812
0x301
autostart
1
/a/TopXlet
0x302
present
1
/b/GameXlet

图18

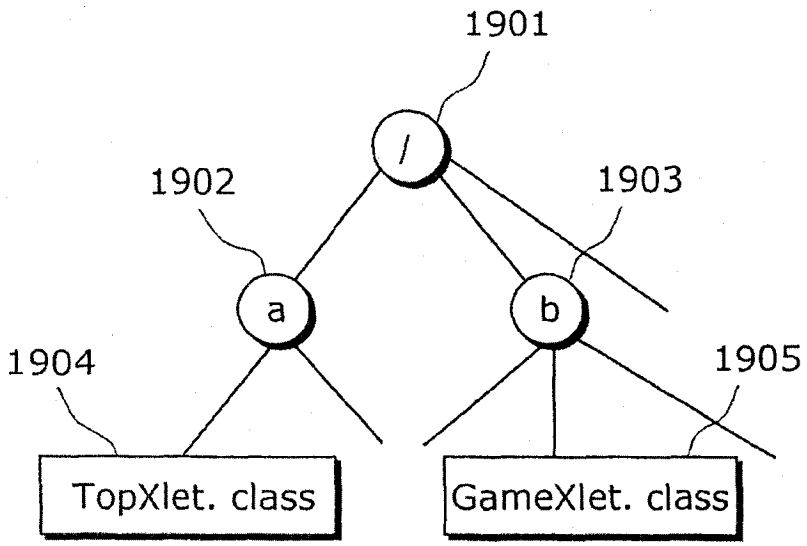

1901
/
1902
1903
a
b
1904
1905
TopXlet. class
GameXlet. class

图19

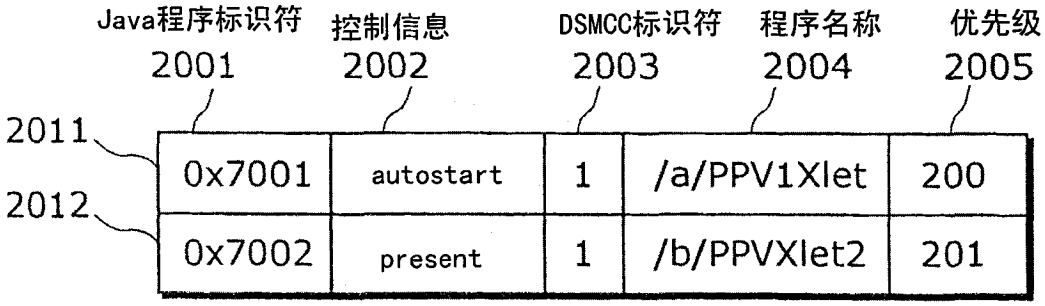

Java程序标识符
2001
控制信息
2002
DSMCC标识符
2003
程序名称
2004
优先级
2005
2011
2012
0x7001
autostart
1
/a/PPV1Xlet
200
0x7002
present
1
/b/PPVXlet2
201

图20

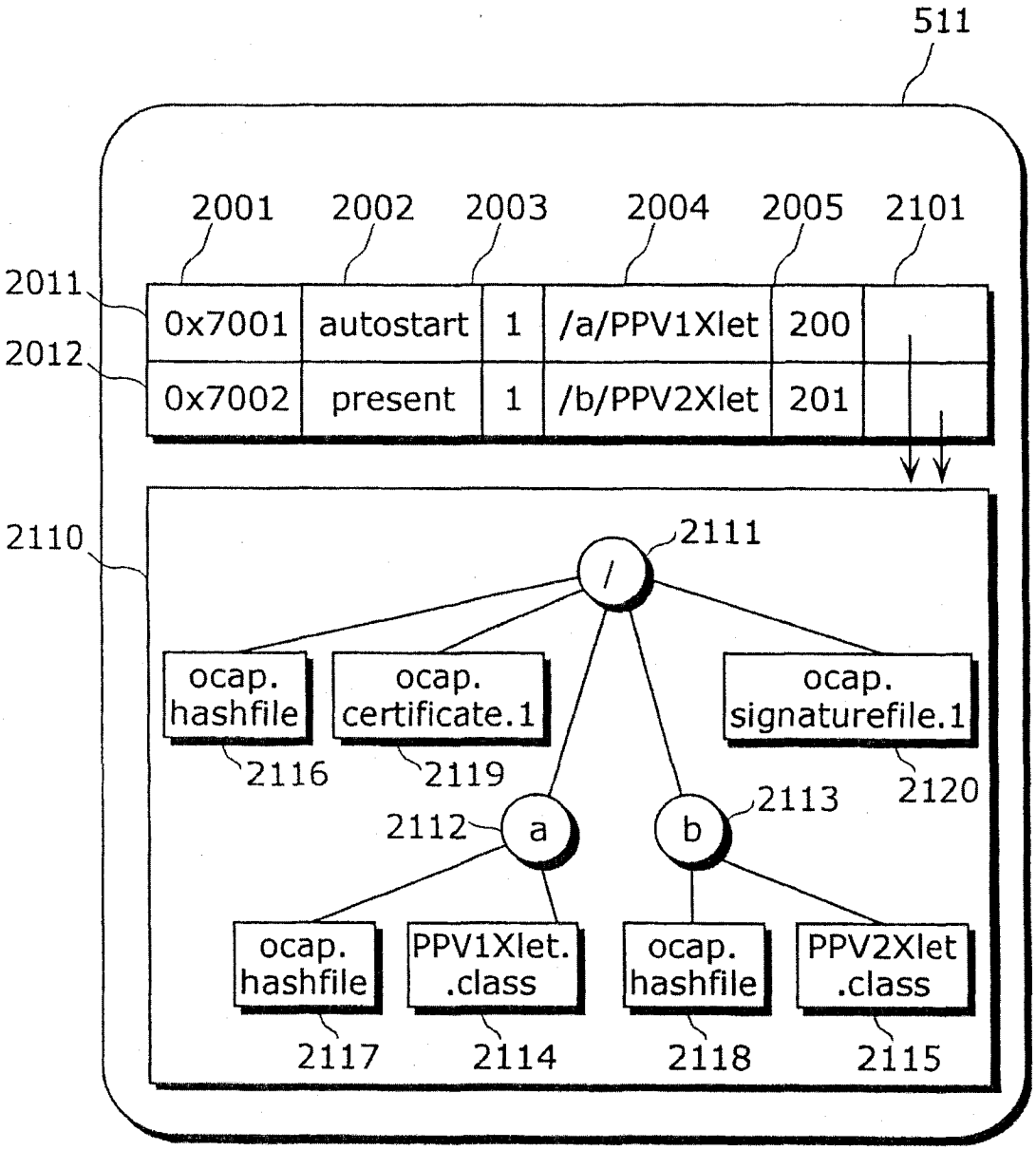

511
2001
2002
2003
2004
2005
2101
2011
2012
0x7001
autostart
1
/a/PPV1Xlet
200
0x7002
present
1
/b/PPV2Xlet
201
2110
2111
/
ocap.
hashfile
ocap.
certificate.1
ocap.
signaturefile.1
2116
2119
2120
2112
a
b
2113
ocap.
hashfile
PPV1Xlet.
.class
ocap.
hashfile
PPV2Xlet
.class
2117
2114
2118
2115

图21

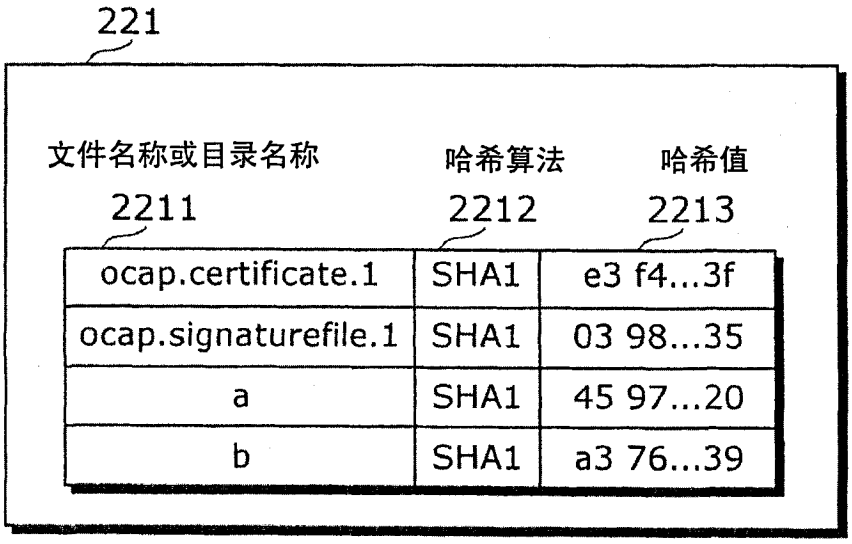


图22A

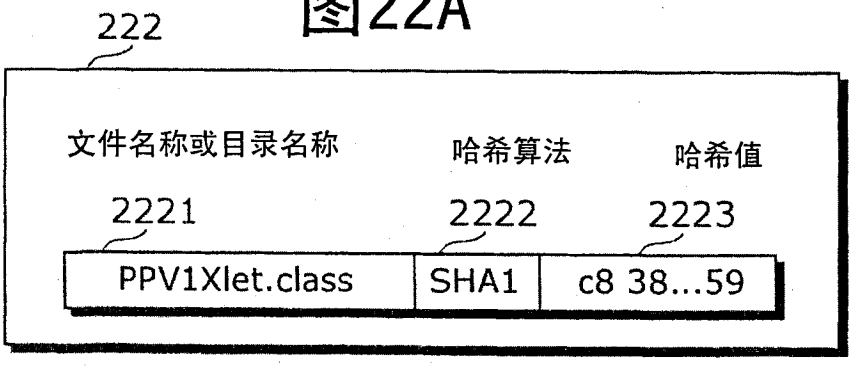


图22B

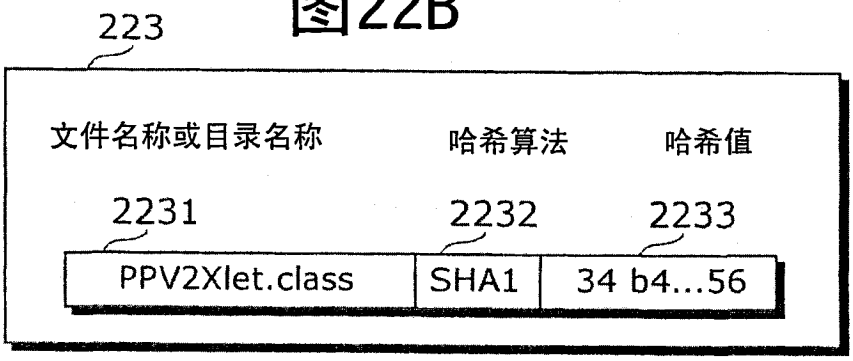


图22C

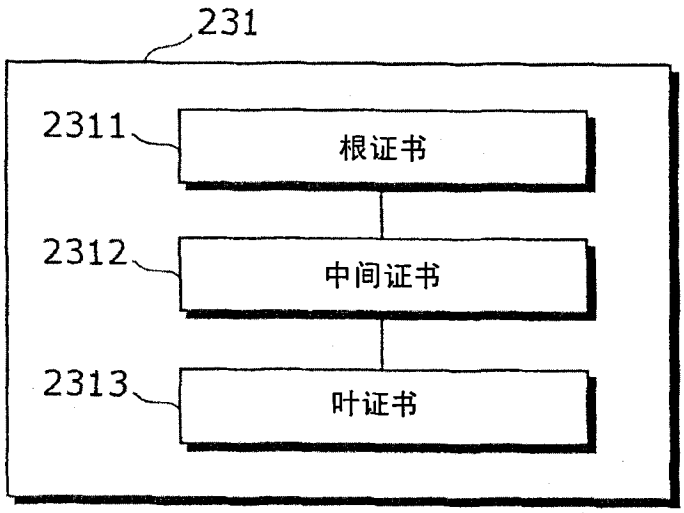


图23

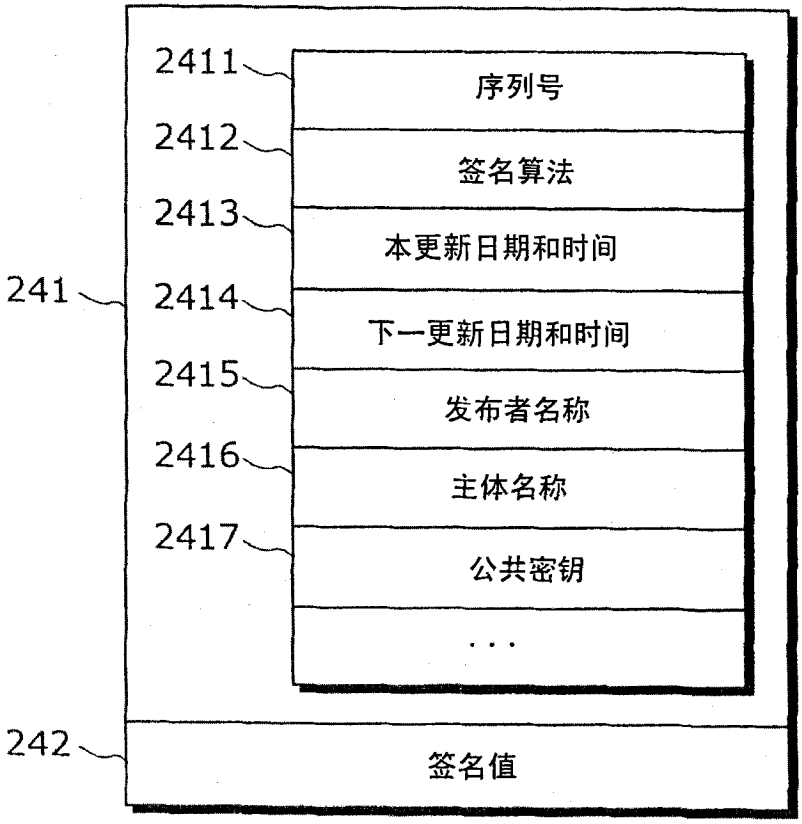

241
2411
序列号
2412
签名算法
2413
本更新日期和时间
2414
下一更新日期和时间
2415
发布者名称
2416
主体名称
2417
公共密钥
. . .
242
签名值

图24

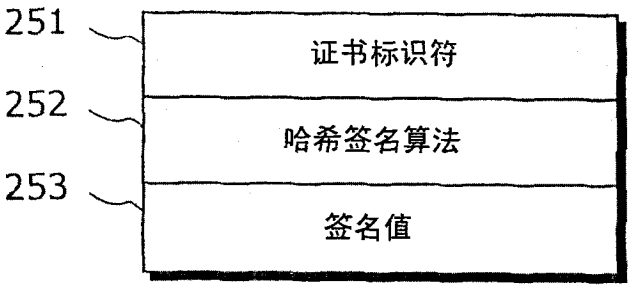


图25

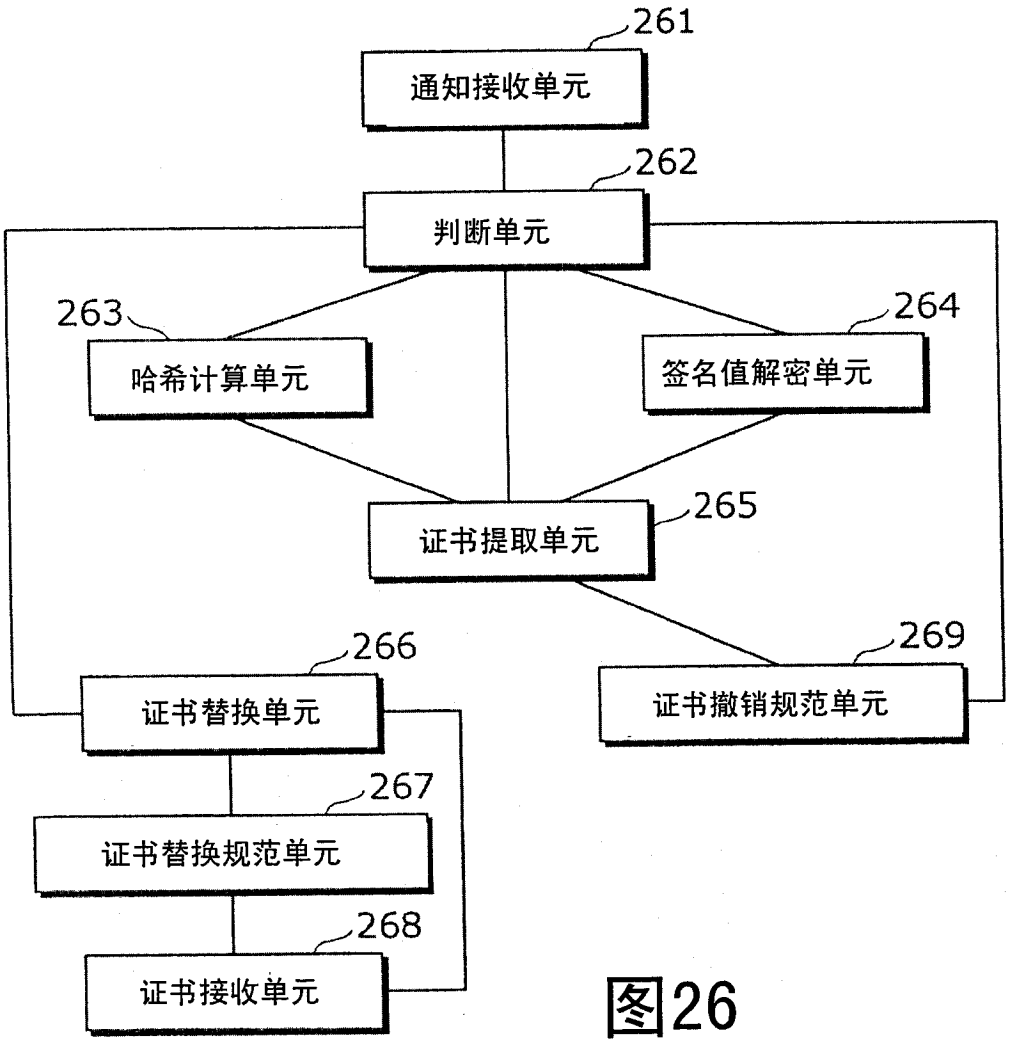

261
通知接收单元
262
判断单元
263
哈希计算单元
264
签名值解密单元
265
证书提取单元
266
证书替换单元
267
证书替换规范单元
268
证书接收单元
269
证书撤销规范单元

图26

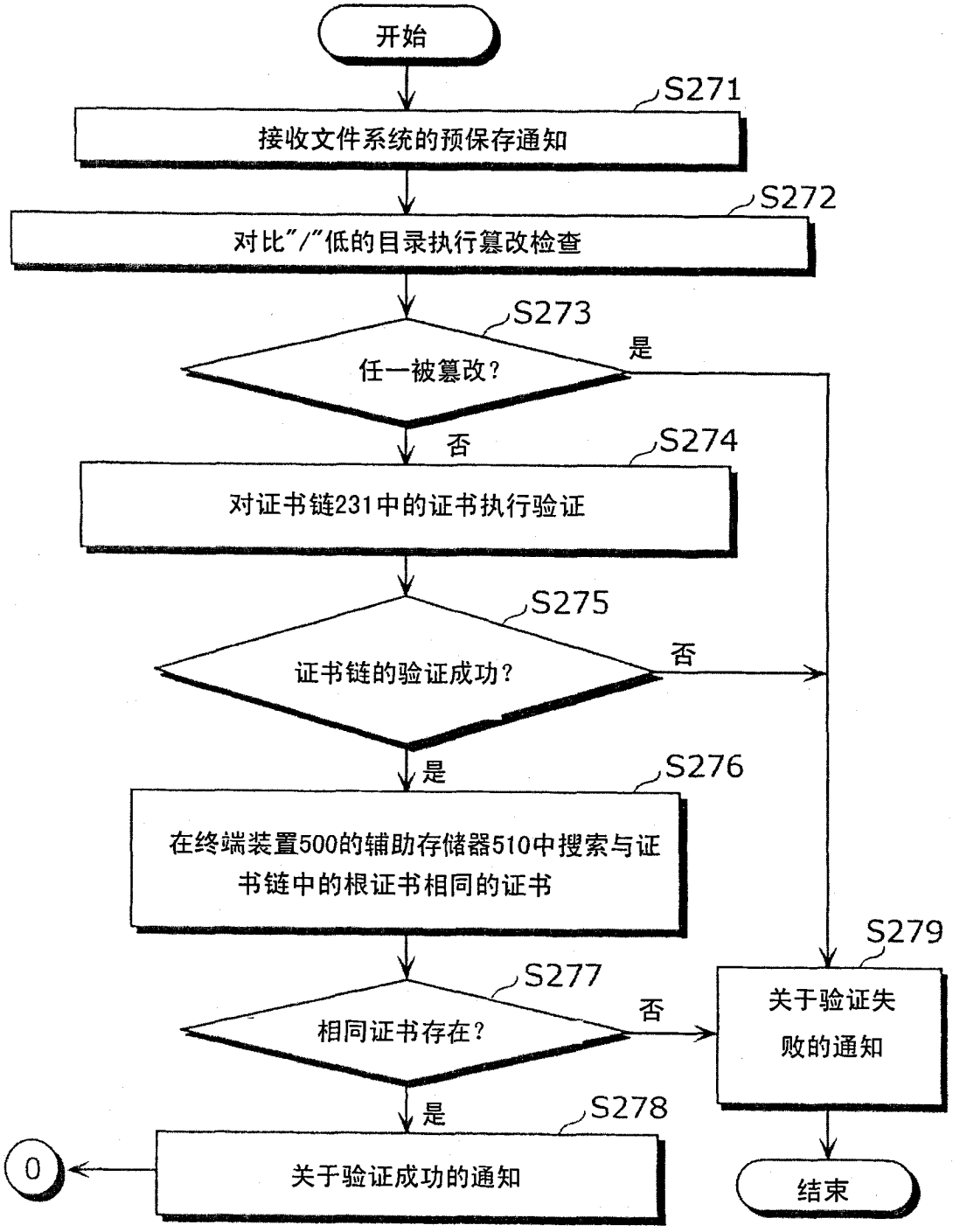

开始
S271
接收文件系统的预保存通知
S272
对比"/"低的目录执行篡改检查
S273
任一被篡改？
是
否
S274
对证书链231中的证书执行验证
S275
证书链的验证成功？
否
是
S276
在终端装置500的辅助存储器510中搜索与证书链中的根证书相同的证书
S277
相同证书存在？
否
是
S278
关于验证成功的通知
0
S279
关于验证失败的通知
结束

图27

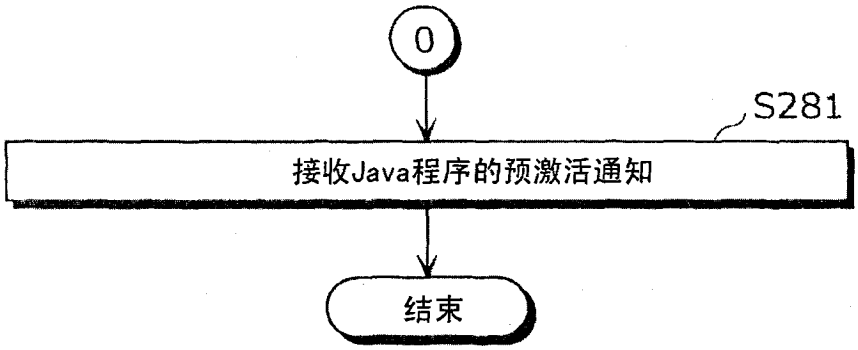


图28

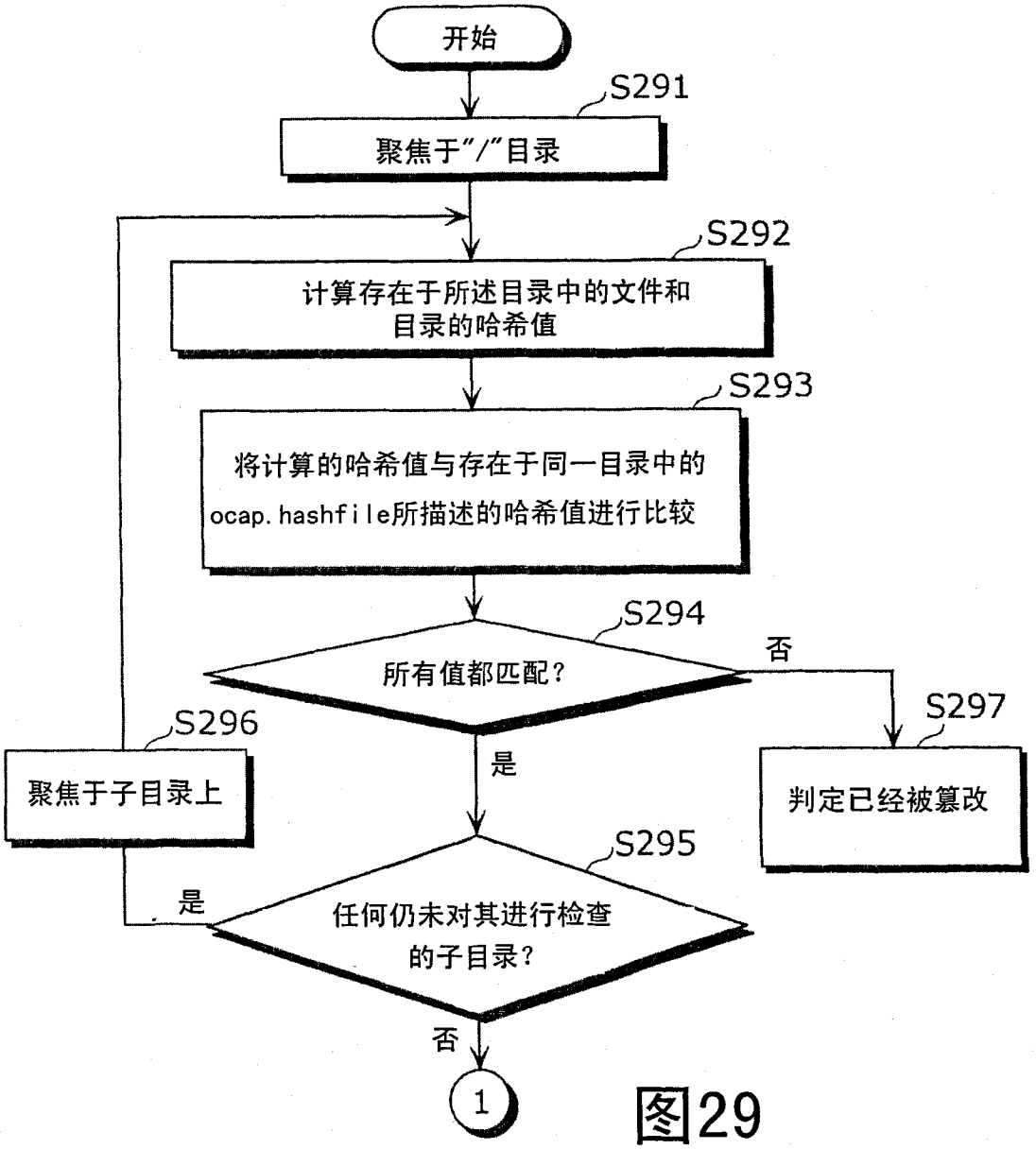

开始
S291
聚焦于"/"目录
S292
计算存在于所述目录中的文件和目录的哈希值
S293
将计算的哈希值与存在于同一目录中的ocap.hashfile所描述的哈希值进行比较
S294
所有值都匹配？
否
S297
判定已经被篡改
是
S296
聚焦于子目录上
S295
是
任何仍未对其进行检查的子目录？
否
1

图29

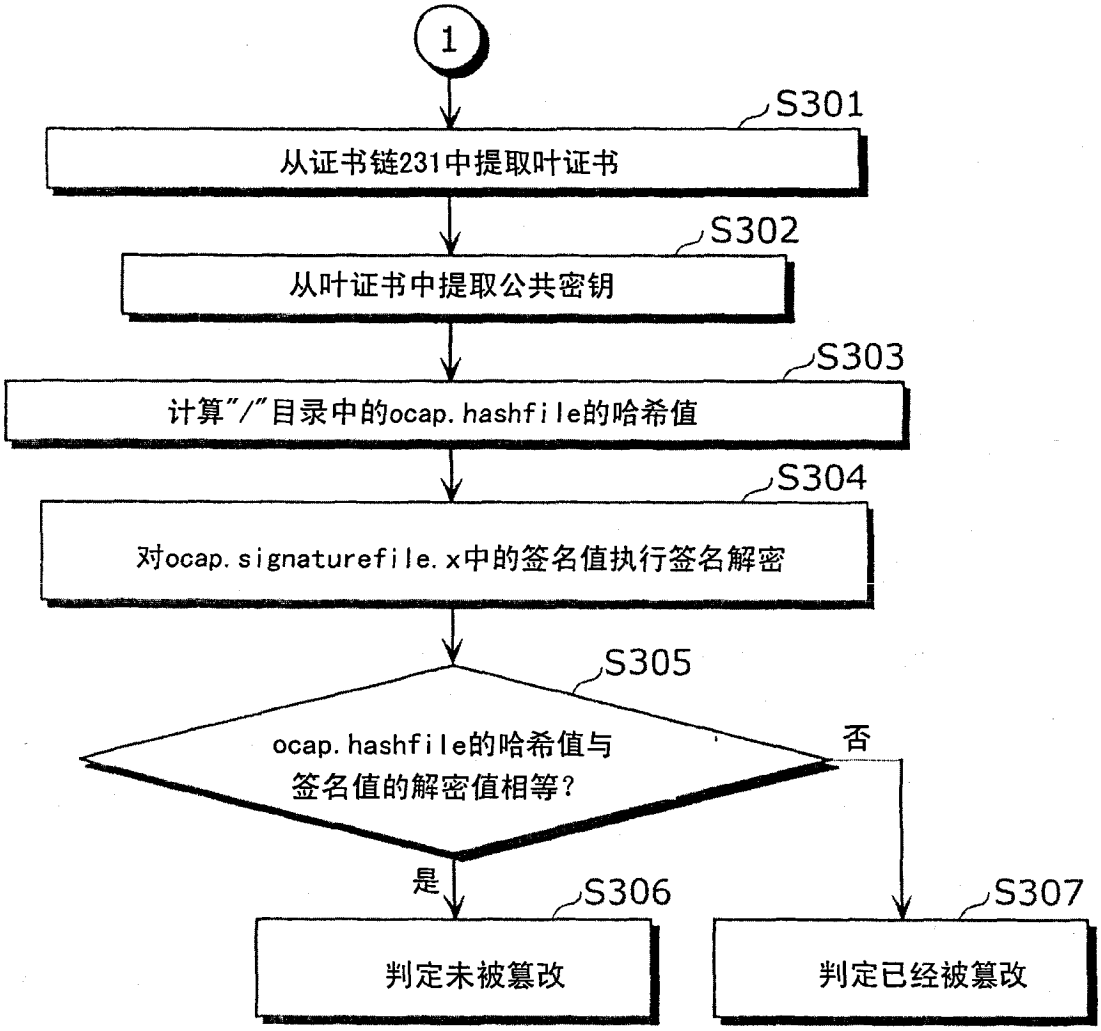

1
S301
从证书链231中提取叶证书
S302
从叶证书中提取公共密钥
S303
计算"/"目录中的ocap.hashfile的哈希值
S304
对ocap.signaturefile.x中的签名值执行签名解密
S305
ocap.hashfile的哈希值与
签名值的解密值相等?
否
是
S306
判定未被篡改
S307
判定已经被篡改

图30

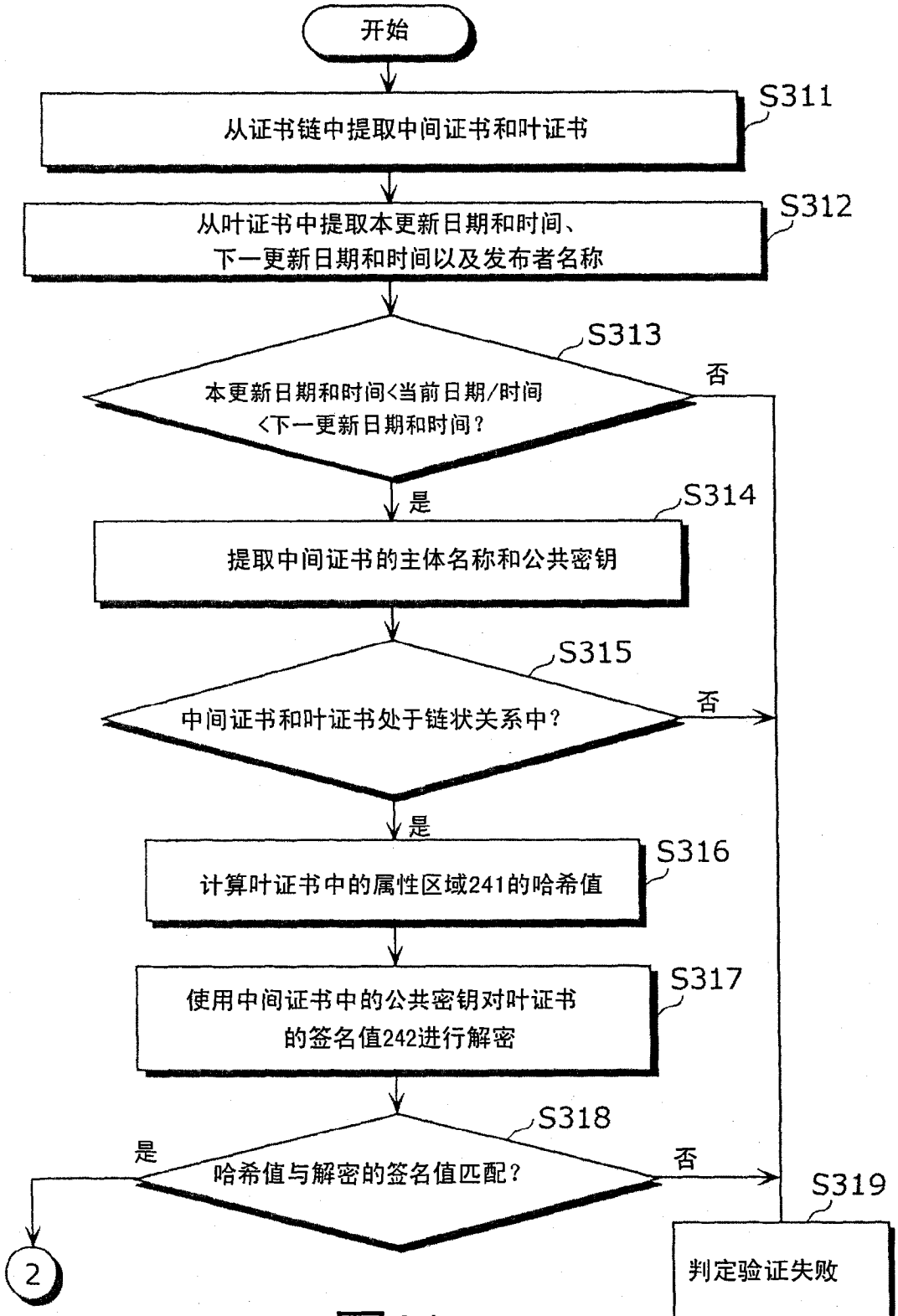

开始
S311
从证书链中提取中间证书和叶证书
S312
从叶证书中提取本更新日期和时间、
下一更新日期和时间以及发布者名称
S313
本更新日期和时间<当前日期/时间
<下一更新日期和时间？
否
是
S314
提取中间证书的主体名称和公共密钥
S315
中间证书和叶证书处于链状关系中？
否
是
S316
计算叶证书中的属性区域241的哈希值
S317
使用中间证书中的公共密钥对叶证书
的签名值242进行解密
S318
是
哈希值与解密的签名值匹配？
否
S319
判定验证失败
2

图31

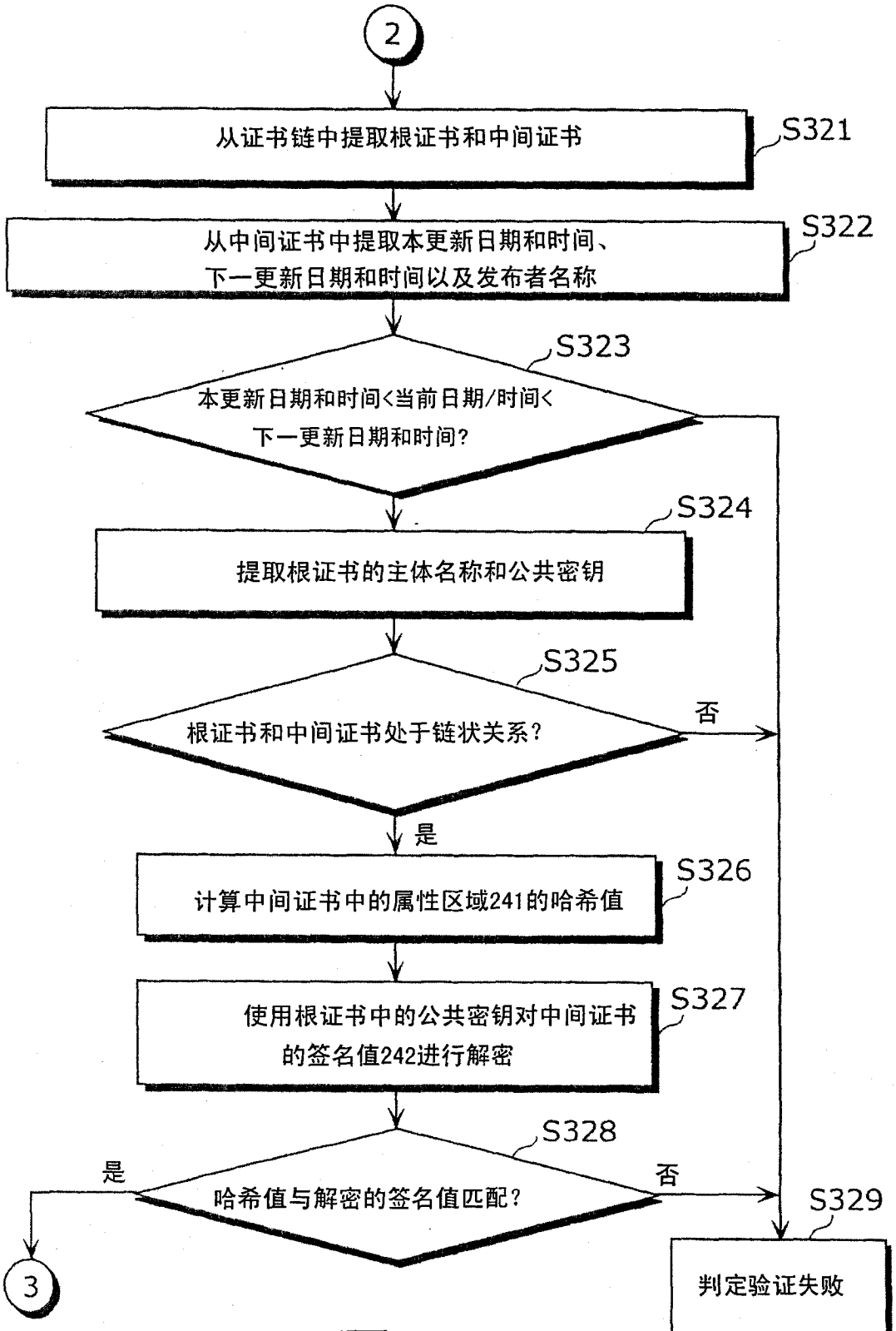

2
从证书链中提取根证书和中间证书
S321
从中间证书中提取本更新日期和时间、
下一更新日期和时间以及发布者名称
S322
S323
本更新日期和时间<当前日期/时间<
下一更新日期和时间?
S324
提取根证书的主体名称和公共密钥
S325
根证书和中间证书处于链状关系?
否
是
S326
计算中间证书中的属性区域241的哈希值
S327
使用根证书中的公共密钥对中间证书
的签名值242进行解密
S328
是
哈希值与解密的签名值匹配?
否
S329
判定验证失败
3

图32

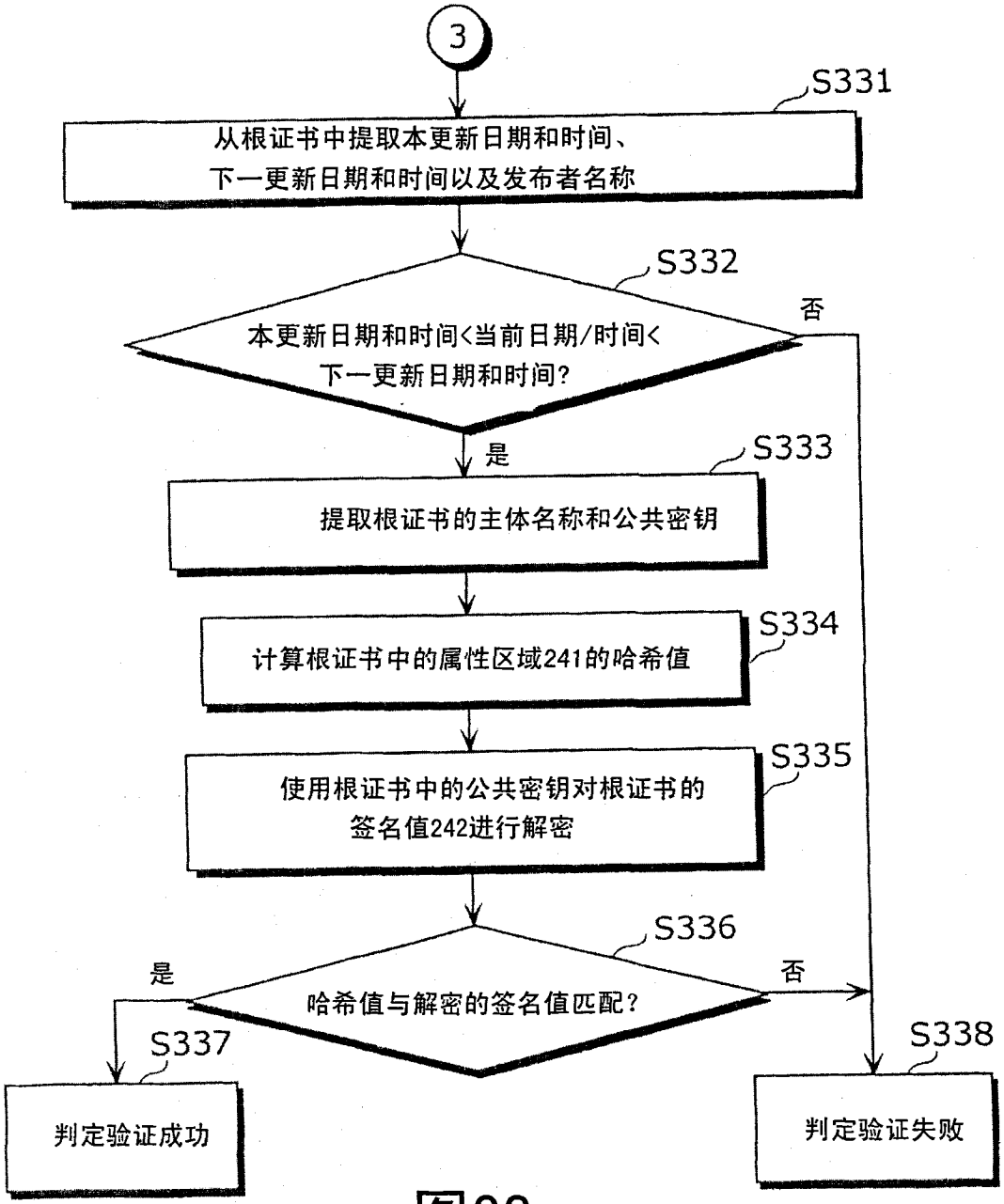

3
S331
从根证书中提取本更新日期和时间、
下一更新日期和时间以及发布者名称
S332
本更新日期和时间<当前日期/时间<
下一更新日期和时间?
否
是
S333
提取根证书的主体名称和公共密钥
S334
计算根证书中的属性区域241的哈希值
S335
使用根证书中的公共密钥对根证书的
签名值242进行解密
S336
哈希值与解密的签名值匹配?
是
否
S337
判定验证成功
S338
判定验证失败

图33

```
"-//OCAP//DTD Application Description File 1.0//EN"
"http://www.cablelabs.com/ocap/dtd/applicationdescriptionfile
-1-0.dtd"
<applicationdescription>
   <dir name="/">
      <file name="ocap.hashfile" size="25"/>
      <file name="ocap.certificate.1" size="100"/>
      <file name="ocap.signaturefile.1" size="30"/>
      <dir name="a">
         <file name="ocap.hashfile" size="15"/>
         <file name="PPV1Xlet.class" size="1000"/>
      </dir>
      <dir name="b">
         <file name="ocap.hashfile"/>
      </dir>
   </dir>
</applicationdescription>
```

图34

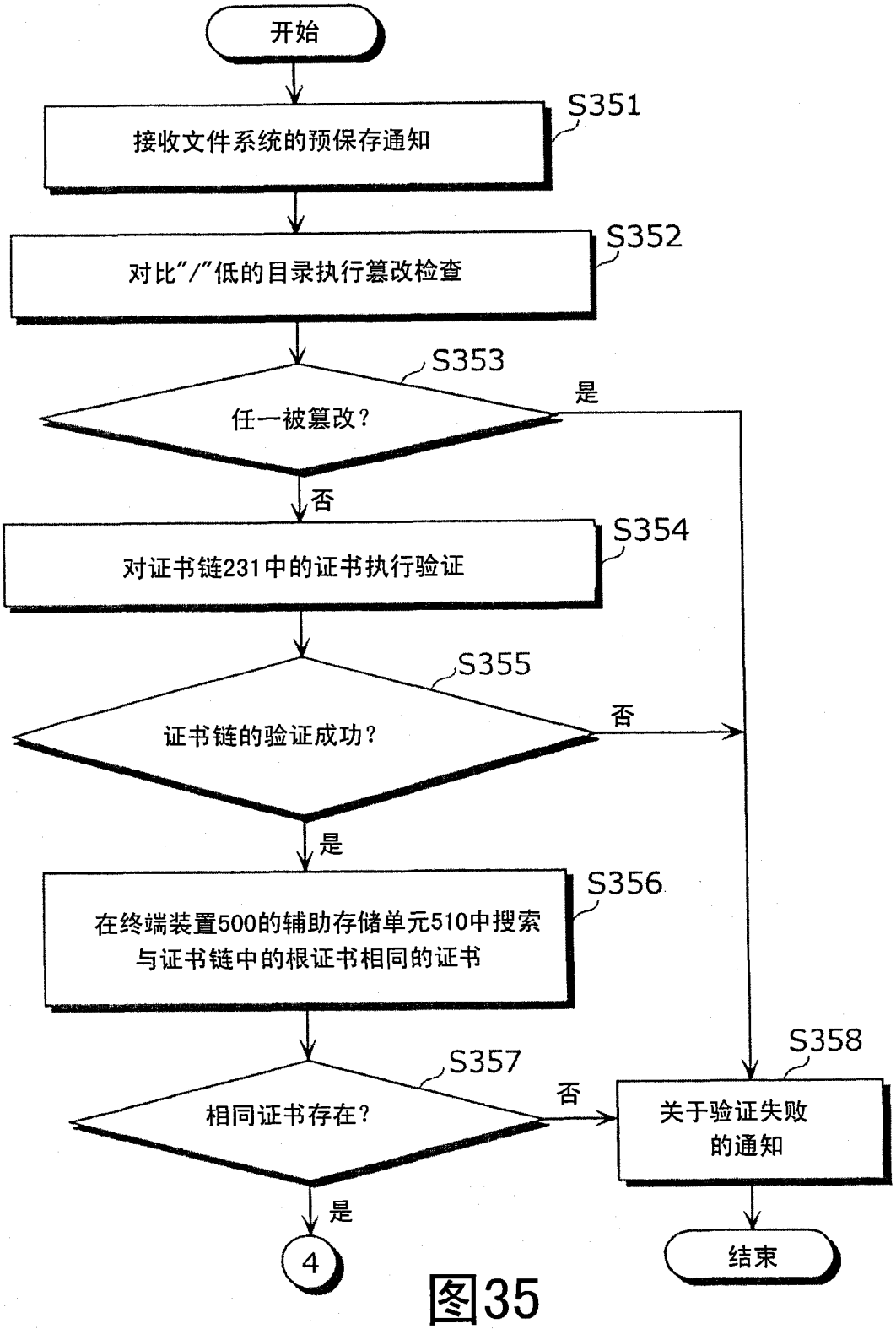

开始
接收文件系统的预保存通知
S351
对比"/"低的目录执行篡改检查
S352
任一被篡改？
S353
是
否
对证书链231中的证书执行验证
S354
证书链的验证成功？
S355
否
是
在终端装置500的辅助存储单元510中搜索
与证书链中的根证书相同的证书
S356
相同证书存在？
S357
否
是
4
关于验证失败
的通知
S358
结束

图35

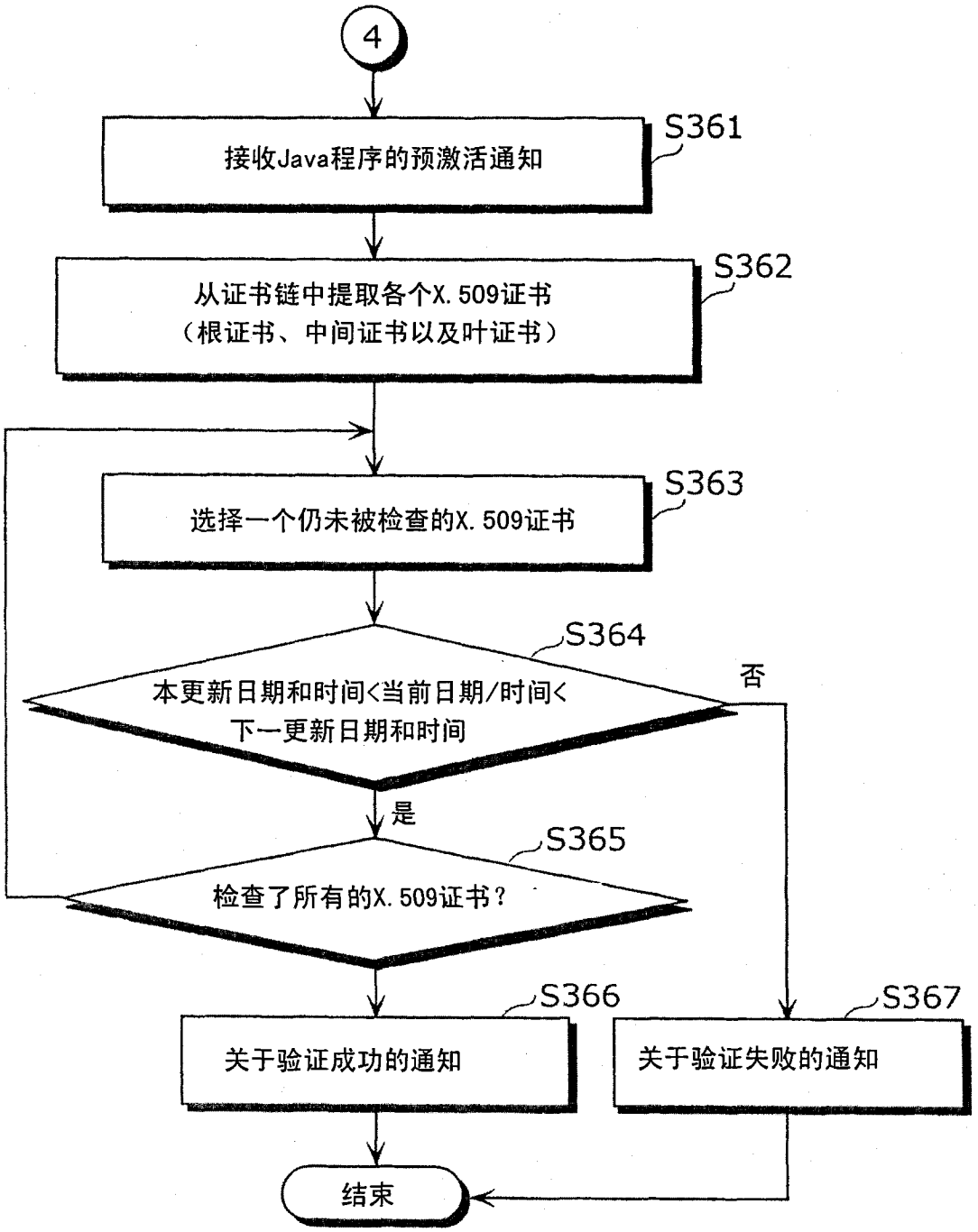

4
接收Java程序的预激活通知
S361
从证书链中提取各个X.509证书
（根证书、中间证书以及叶证书）
S362
选择一个仍未被检查的X.509证书
S363
本更新日期和时间<当前日期/时间<
下一更新日期和时间
S364
否
是
检查了所有的X.509证书？
S365
关于验证成功的通知
S366
关于验证失败的通知
S367
结束

图36

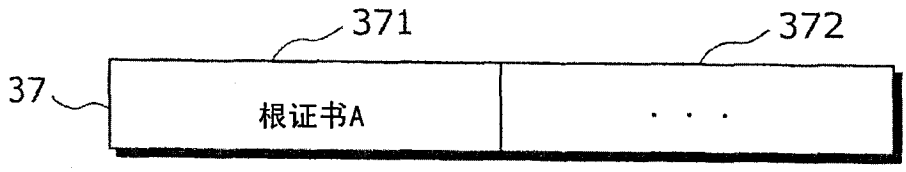


图37

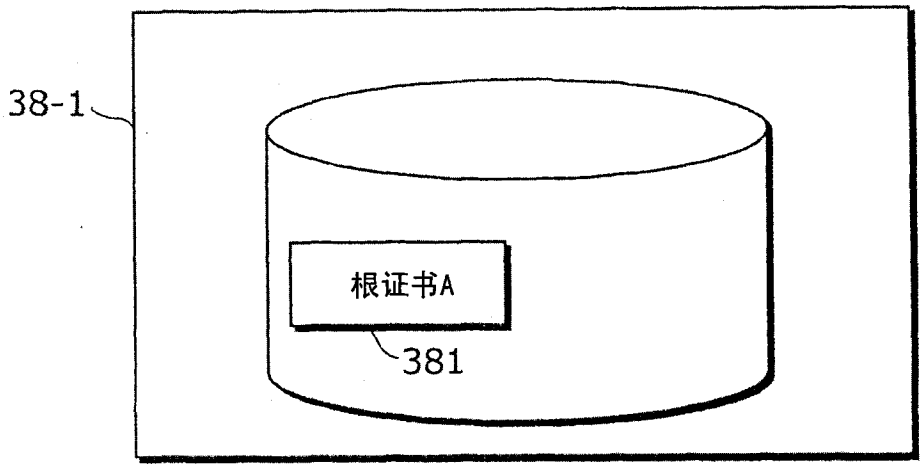


图38A

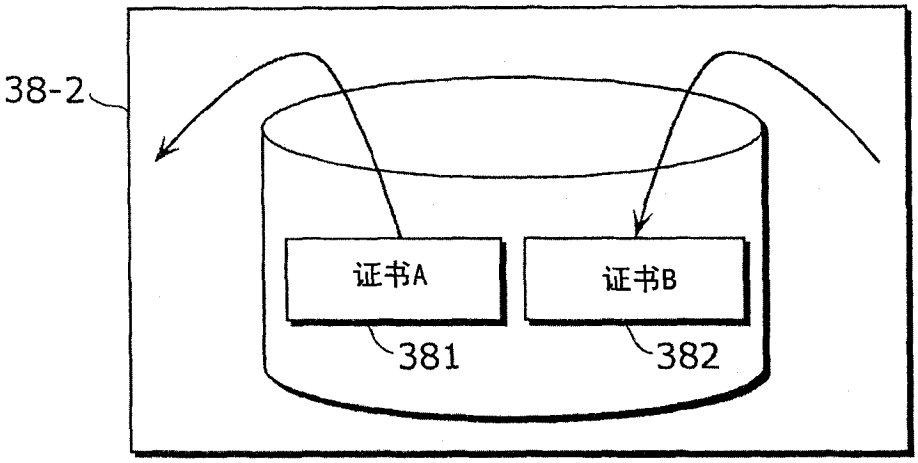


图38B

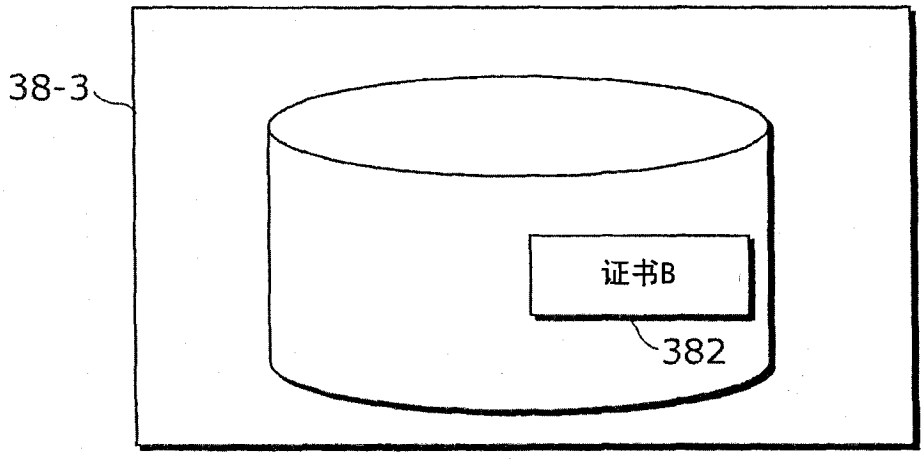


图38C

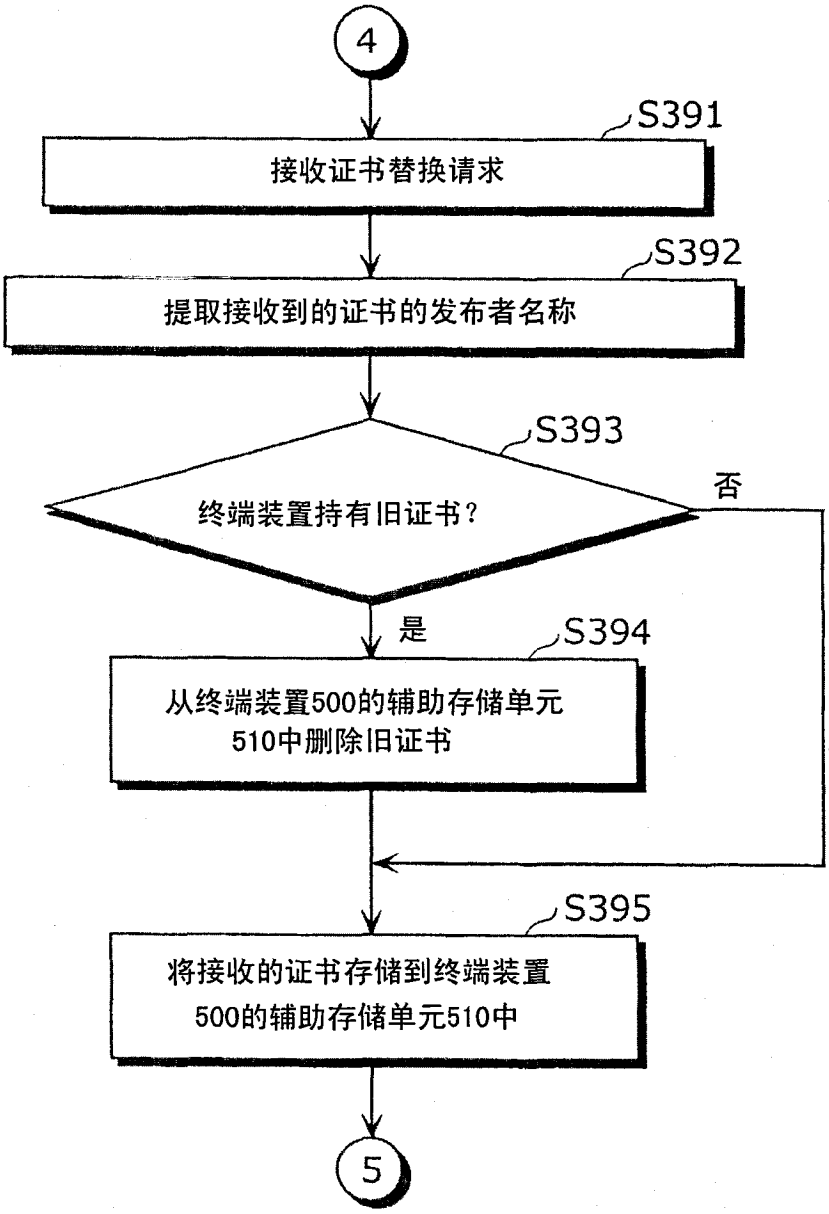

4
S391
接收证书替换请求
S392
提取接收到的证书的发布者名称
S393
终端装置持有旧证书？
否
是
S394
从终端装置500的辅助存储单元510中删除旧证书
S395
将接收的证书存储到终端装置500的辅助存储单元510中
5

图39

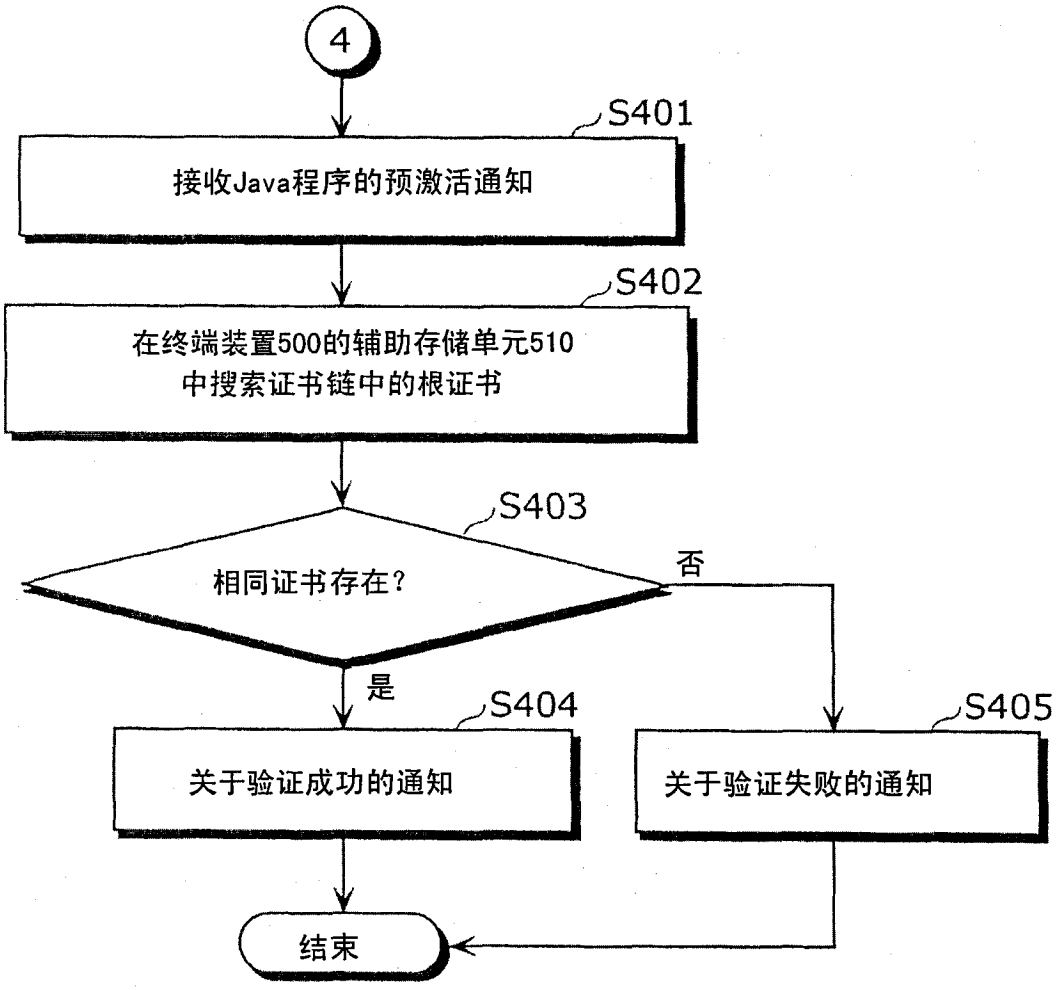

4
S401
接收Java程序的预激活通知
S402
在终端装置500的辅助存储单元510
中搜索证书链中的根证书
S403
相同证书存在？
否
是
S404
关于验证成功的通知
S405
关于验证失败的通知
结束

图40

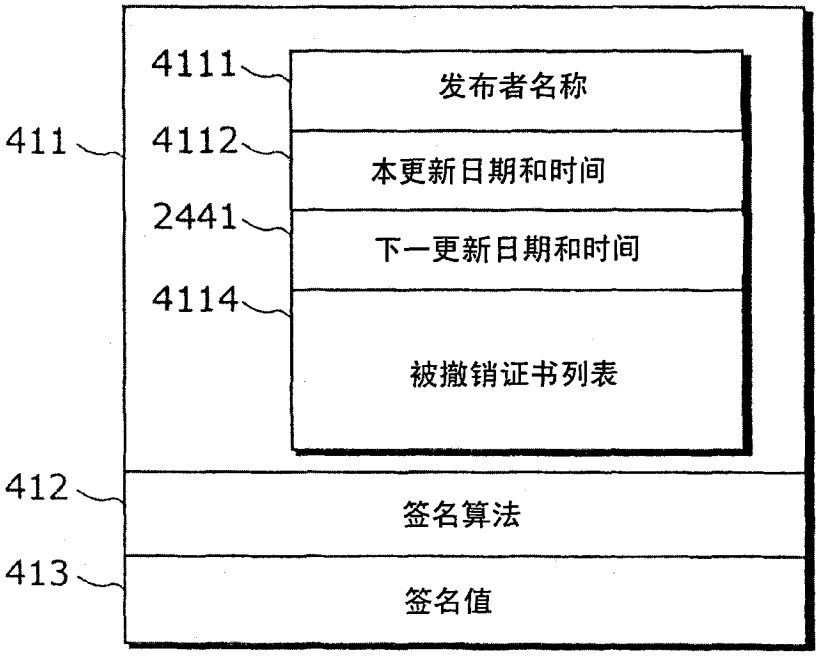

411
4111
发布者名称
4112
本更新日期和时间
2441
下一更新日期和时间
4114
被撤销证书列表
412
签名算法
413
签名值

图41

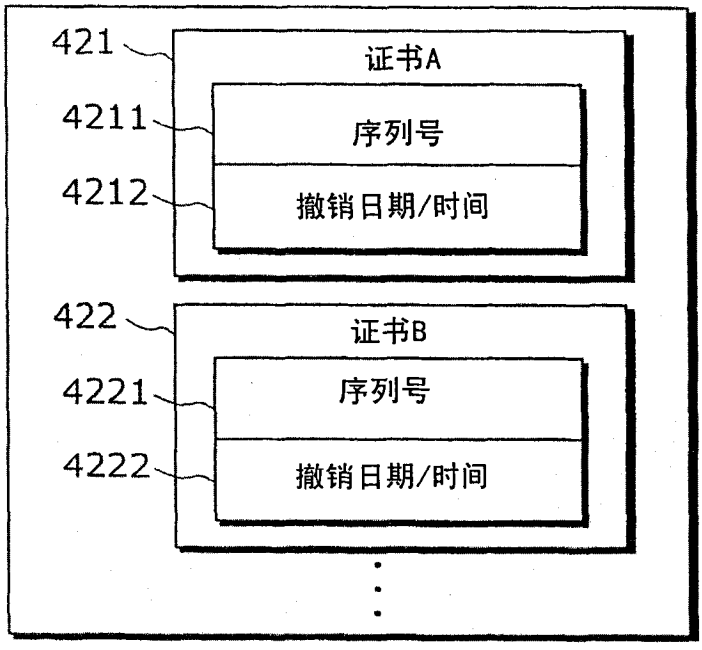

421
证书A
4211
序列号
4212
撤销日期/时间
422
证书B
4221
序列号
4222
撤销日期/时间

图42

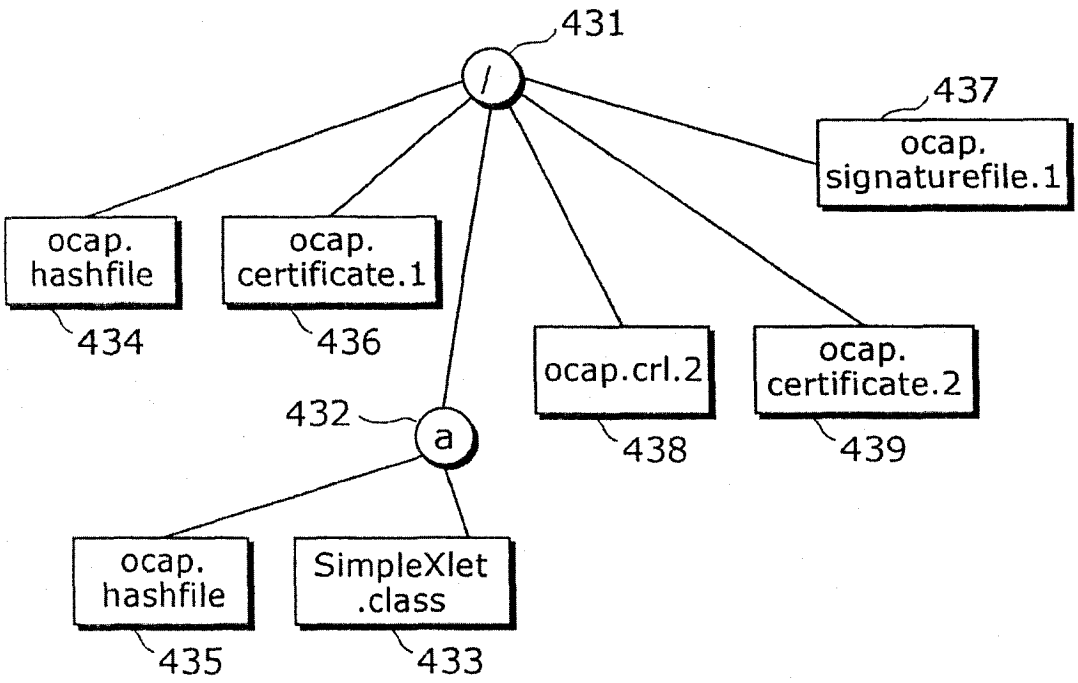

431
/
437
ocap.
signaturefile.1
ocap.
hashfile
434
ocap.
certificate.1
436
ocap.crl.2
438
ocap.
certificate.2
439
432
a
ocap.
hashfile
435
SimpleXlet
.class
433

图43

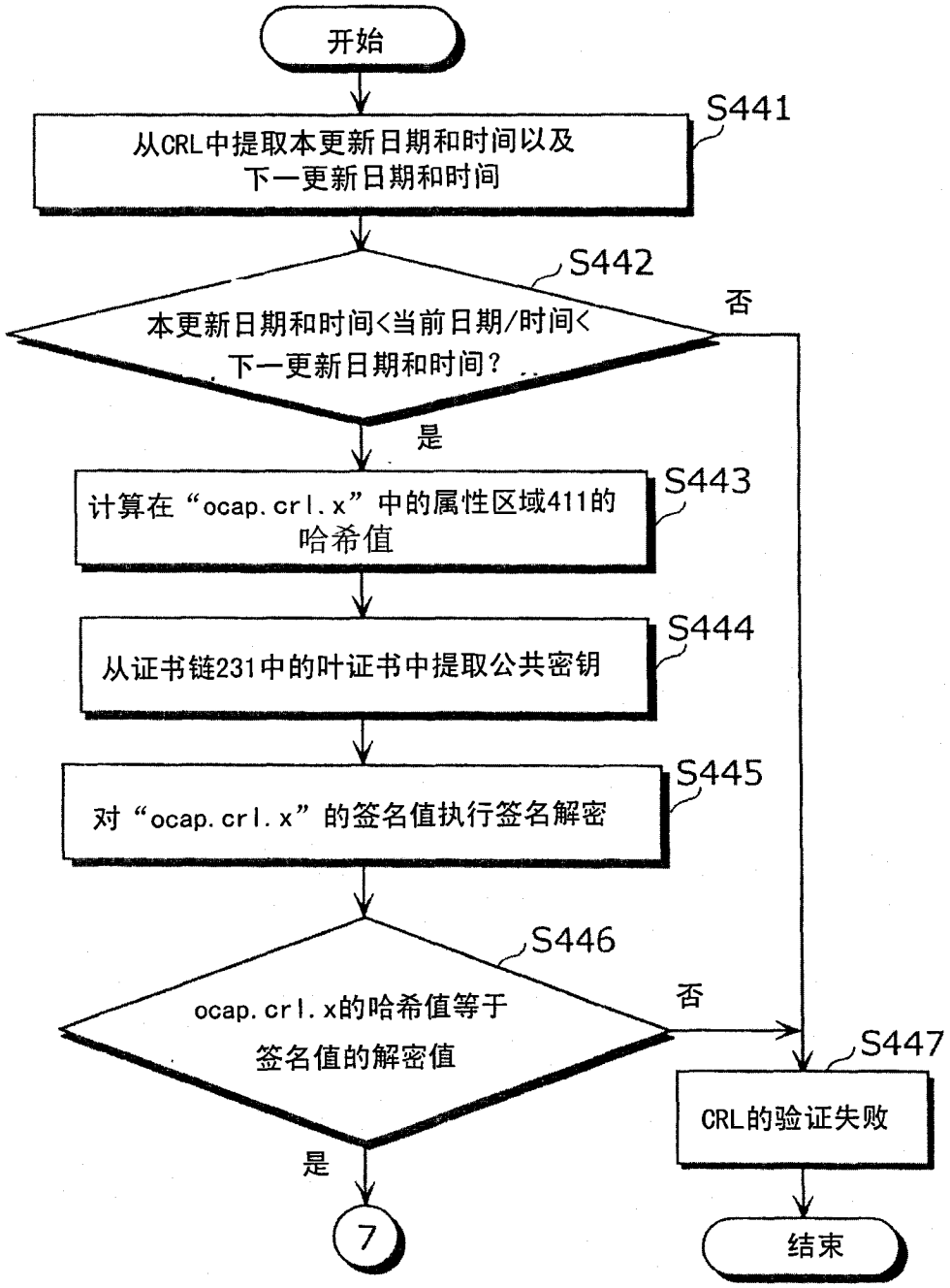

开始
S441
从CRL中提取本更新日期和时间以及
下一更新日期和时间
S442
本更新日期和时间<当前日期/时间<
下一更新日期和时间？
否
是
S443
计算在“ocap.crl.x”中的属性区域411的
哈希值
S444
从证书链231中的叶证书中提取公共密钥
S445
对“ocap.crl.x”的签名值执行签名解密
S446
ocap.crl.x的哈希值等于
签名值的解密值
否
是
S447
CRL的验证失败
7
结束

图44

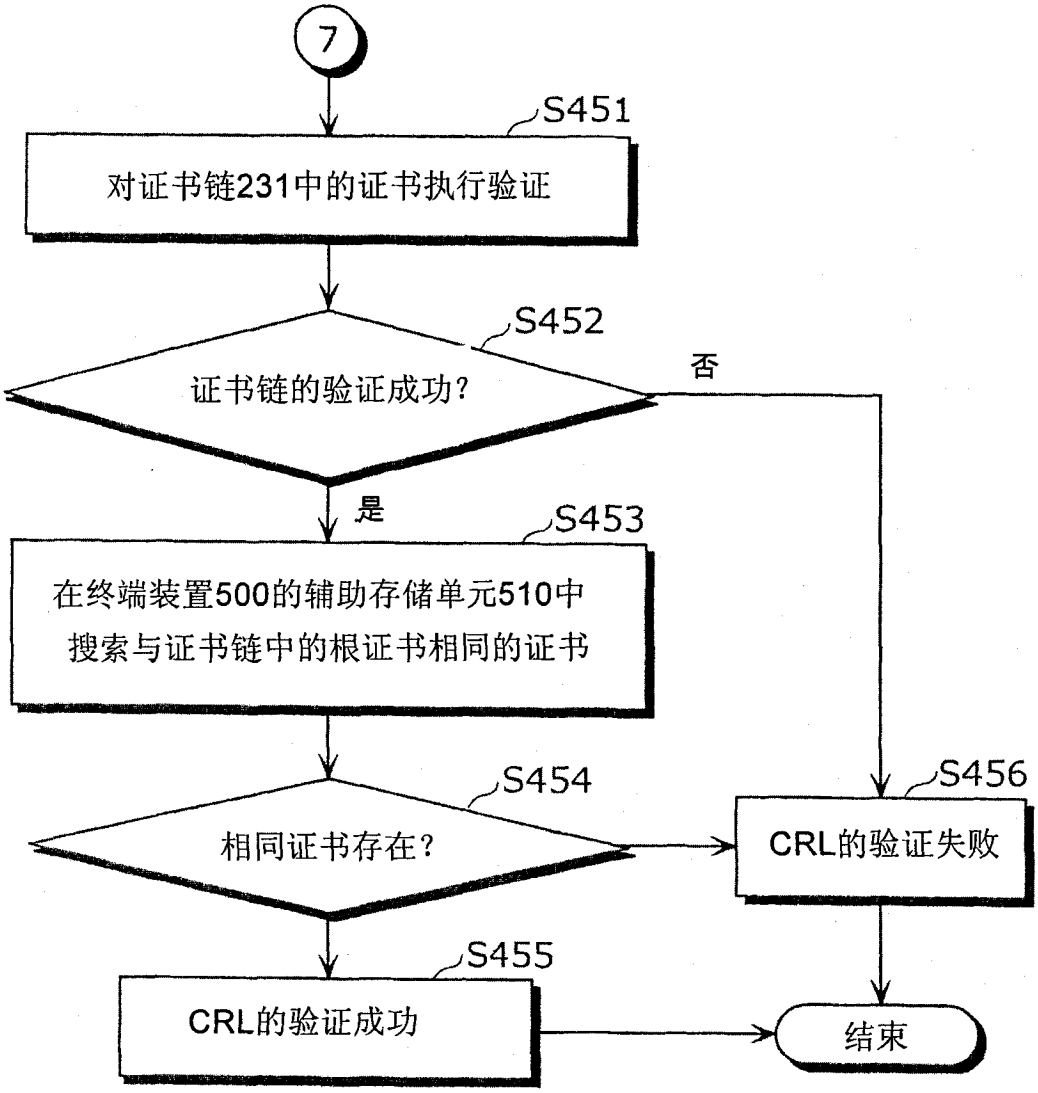

7
S451
对证书链231中的证书执行验证
S452
证书链的验证成功?
否
是
S453
在终端装置500的辅助存储单元510中
搜索与证书链中的根证书相同的证书
S454
相同证书存在?
S456
CRL的验证失败
S455
CRL的验证成功
结束

图45

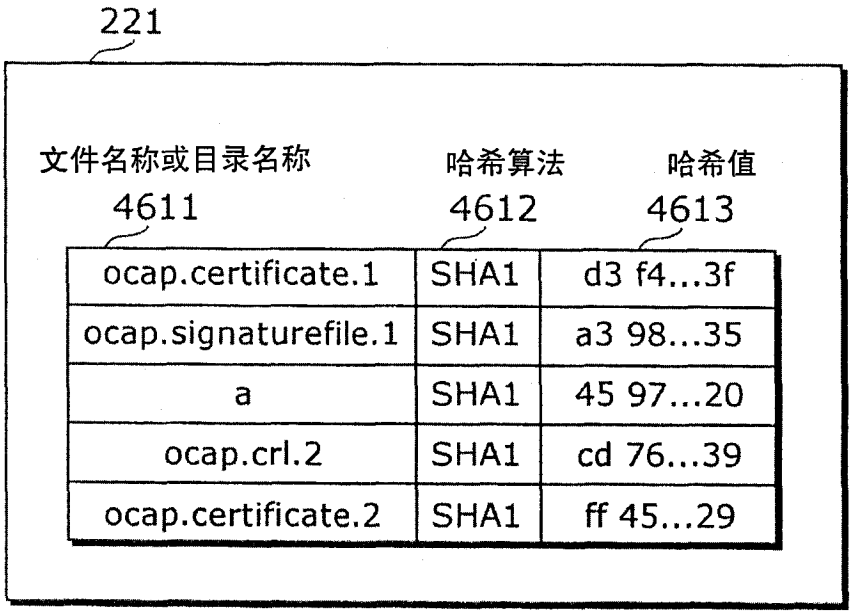

221
文件名称或目录名称
4611
哈希算法
4612
哈希值
4613
ocap.certificate.1
SHA1
d3 f4...3f
ocap.signaturefile.1
SHA1
a3 98...35
a
SHA1
45 97...20
ocap.crl.2
SHA1
cd 76...39
ocap.certificate.2
SHA1
ff 45...29

图46

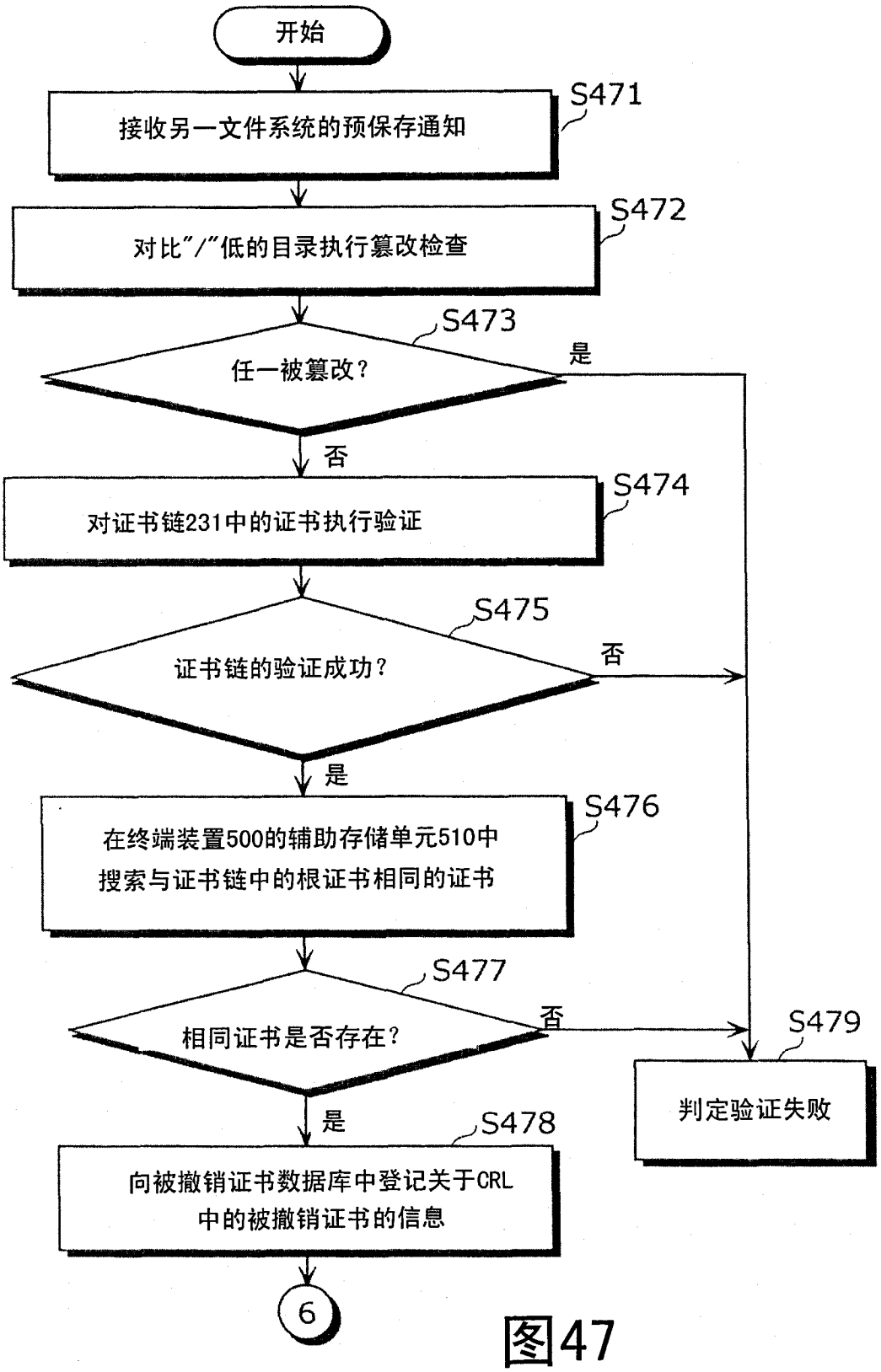

开始
S471
接收另一文件系统的预保存通知
S472
对比"/"低的目录执行篡改检查
S473
任一被篡改？
是
否
S474
对证书链231中的证书执行验证
S475
证书链的验证成功？
否
是
S476
在终端装置500的辅助存储单元510中
搜索与证书链中的根证书相同的证书
S477
相同证书是否存在？
否
S479
判定验证失败
是
S478
向被撤销证书数据库中登记关于CRL
中的被撤销证书的信息
6

图47

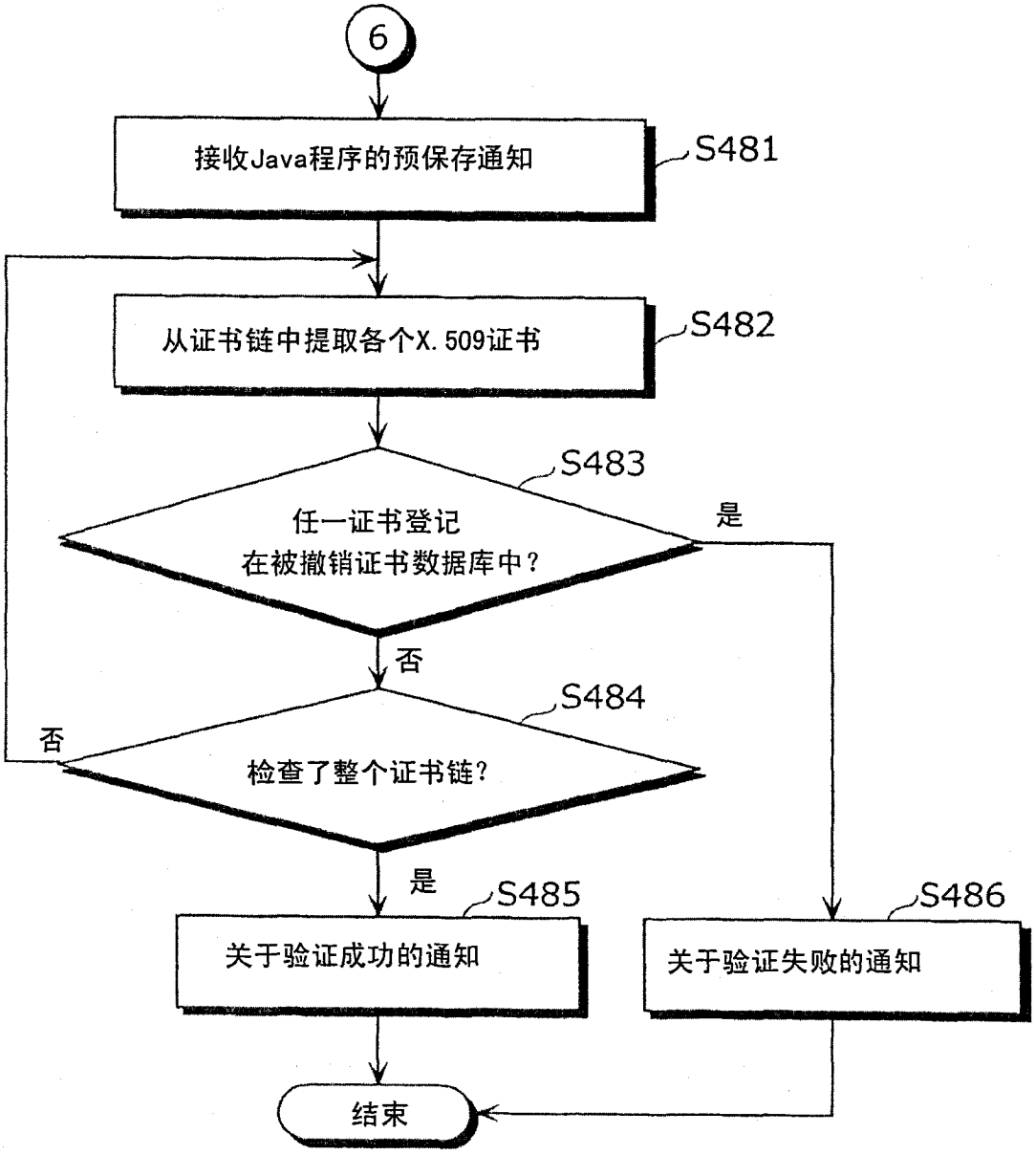

6
接收Java程序的预保存通知
S481
从证书链中提取各个X.509证书
S482
S483
任一证书登记
在被撤销证书数据库中？
是
否
S484
检查了整个证书链？
否
是
S485
关于验证成功的通知
S486
关于验证失败的通知
结束

图48

发布者名称 491	序列号 492	撤销日期/时间 493
P	3	2003-06-23 15:00 GMT
S	5	2003-04-12 23:00 GMT
D	1	2002-08-03 09:10 GMT
T	10	2003-12-02 05:00 GMT
K	13	2003-12-04 02:50 GMT

图49

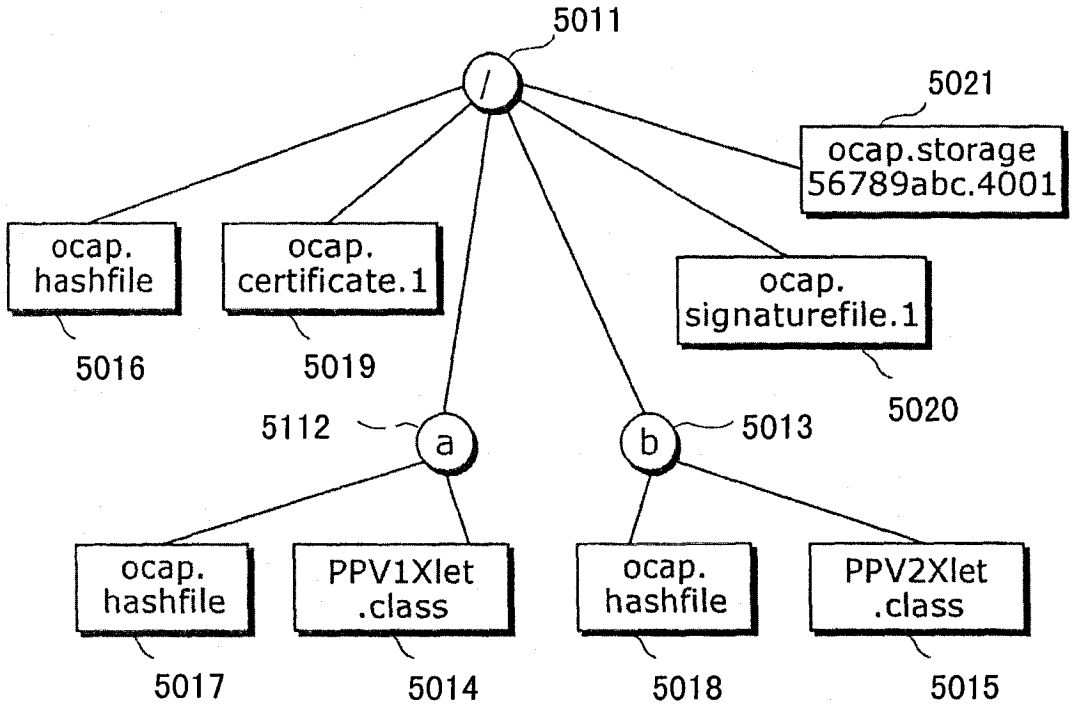


图 50

```
"-//OCAP//DTD Application Description File 1.0//EN"
"http://www.cablelabs.com/ocap/dtd/applicationdescriptionfile
-1-0.dtd"
<applicationdescription>
   <dir name="/">
      <file name="ocap.hashfile" size="25"/>
      <dir name="a">
         <file name="PPV1Xlet.class" size="1000"/>
      </dir>
      <dir name="b">
         <file name="ocap.hashfile" size="15"/>
         <file name="PPV2Xlet.class" size="1000"/>
      </dir>
   </dir>
</applicationdescription>
```

图51