(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 103202045 B
(45)授权公告日 2016.06.01

(21)申请号 201180053416.4
(22)申请日 2011.11.04
(30)优先权数据
61/410,781 2010.11.05 US
(85)PCT国际申请进入国家阶段日
2013.05.06
(86)PCT国际申请的申请数据
PCT/US2011/059274 2011.11.04
(87)PCT国际申请的公布数据
WO2012/061678 EN 2012.05.10
(73)专利权人 交互数字专利控股公司
地址 美国特拉华州
(72)发明人 Y·C·沙阿 L·凯斯 D·F·豪利 I·查 A·莱切尔 A·施米特
(74)专利代理机构 北京润平知识产权代理有限公司 11283
代理人 陈潇潇 刘国平
(51)Int.Cl.
H04W 12/10(2009.01)
G06F 21/10(2013.01)
G06F 21/57(2013.01)
(56)对比文件
US 2007113120 A1,2007.05.17,
WO 2010102259 A2,2010.09.10,
WO 02056133 A2,2002.07.18,
审查员 马洁

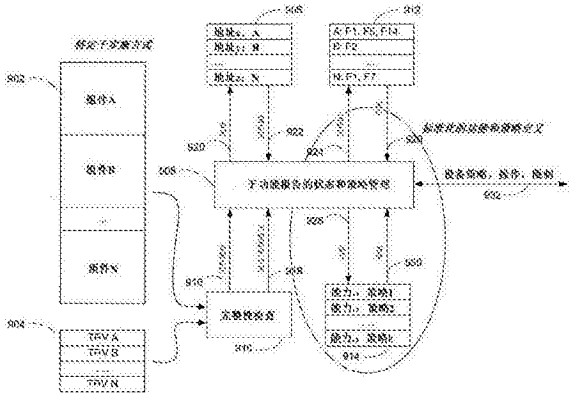
权利要求书3页 说明书36页 附图27页

(54)发明名称

设备检验、遇险指示和补救

(57)摘要

一种无线通信设备，其可以被配置成执行完整性检查和向网络实体的询问，来隔离无线通信设备上的发生故障的组件的部分来进行补救。一旦确定了设备的组件上的完整性故障，则设备可以识别与该组件相关联的功能，并且向网络实体指示该发生故障的功能。无线网络设备和网络实体都可以使用组件－功能映射来识别发生故障的功能和/或发生故障的组件。在接收到设备处的完整性故障的指示之后，网络实体可以确定在设备处执行一个或多个附加的完整性检查迭代，以便窄化发生故障的组件上的完整性故障的范围。一旦隔离了完整性故障，则网络实体可以修复无线通信设备上的发生故障的组件的部分。



CN 103202045 B

1.一种用于执行完整性检查的方法，该方法包括：

对关联于无线通信设备的组件执行第一完整性检查；

确定该组件未能通过所述第一完整性检查；

确定与发生故障的所述组件相对应的功能；

向网络实体发送对与所述发生故障的组件相对应的所述功能的指示；

从所述网络实体接收对所述发生故障的组件执行第二完整性检查以便确定所述发生故障的组件中导致该组件未能通过所述第一完整性检查的部分的请求；以及

对所述发生故障的组件执行所述第二完整性检查，以便隔离所述发生故障的组件的所述部分，从而由所述网络实体进行补救。

2.根据权利要求1所述的用于执行完整性检查的方法，其中所述接收步骤包括：从所述网络实体接收对所述发生故障的组件的所述部分执行多次所述第二完整性检查迭代的请求；以及

其中所述对所述发生故障的组件执行所述第二完整性检查的步骤包括：对所述发生故障的组件的所述部分执行多次所述第二完整性检查迭代，以便进一步隔离所述发生故障的组件的所述部分，以便由所述网络实体进行补救。

3.根据权利要求1所述的用于执行完整性检查的方法，其中所述指示包括警报，该警报被配置成触发与所述网络实体进行的远程更新过程，以便替换所述发生故障的组件的所述部分。

4.根据权利要求1所述的用于执行完整性检查的方法，该方法还包括：基于保存在所述无线通信设备处的组件-功能映射来确定与所述发生故障的组件相对应的功能。

5.根据权利要求1所述的用于执行完整性检查的方法，其中所述指示被安全地发送至所述网络实体。

6.根据权利要求1所述的用于执行完整性检查的方法，其中确定所述组件未能通过所述第一次完整性检查还包括：

确定与所述组件相关联的完整性量度；

将该完整性量度与关联于所述组件的可信基准值相比较；以及

确定所述完整性量度与所述可信基准值不匹配。

7.根据权利要求1所述的用于执行完整性检查的方法，该方法还包括：

接收与所述发生故障的组件的所述部分相关联的替换组件；以及

用所述替换组件来替换所述发生故障的组件的所述部分。

8.根据权利要求7所述的用于执行完整性检查的方法，其中所述发生故障的组件的所述部分是用使用了其他通过了所述第一完整性检查的与所述网络设备相关联的组件的替换组件来替换的。

9.根据权利要求1所述的用于执行完整性检查的方法，该方法还包括：在所述确定所述组件未能通过所述第一完整性检查时，阻止释放用于附着于所述网络实体的密钥。

10.根据权利要求1所述的用于执行完整性检查的方法，其中所述发生故障的组件是在多阶段的安全引导处理期间确定的。

11.根据权利要求1所述的用于执行完整性检查的方法，其中所述网络实体包括设备补救服务器，并且其中所述指示被直接地或者经由网络发送到所述设备补救服务器。

12.根据权利要求1所述的用于执行完整性检查的方法，其中所述第一完整性检查和所述第二完整性检查是在可信任环境中执行的。

13.根据权利要求1所述的用于执行完整性检查的方法，其中所述功能与将基于所述第一完整性检查的结果来应用的网络策略相关联。

14.根据权利要求1所述的用于执行完整性检查的方法，其中所述功能通过与至少一个其他无线通信设备上的至少一个其他组件相关联而在多个无线通信设备上被标准化。

15.一种用于执行完整性检查的系统，该系统包括：

加载器，被配置成：对与无线通信设备相关联的组件执行第一完整性检查；确定该组件未能通过所述第一完整性检查；以及对发生故障的所述组件的部分执行第二完整性检查，以便隔离所述发生故障的组件的部分，从而由网络实体进行补救；

策略管理实体，被配置成确定与所述发生故障的组件相对应的网络功能；以及

平台完整性策略引擎(PIPE)，被配置成：接收来自所述加载器的对所述发生故障的组件的指示；接收与所述发生故障的组件相对应的所述网络功能；向网络实体报告对与所述发生故障的组件相对应的所述网络功能的指示；以及从所述网络实体接收对所述发生故障的组件执行第二完整性检查以便确定所述发生故障的组件中导致该组件未能通过所述第一完整性检查的部分的请求。

16.根据权利要求15所述的用于执行完整性检查的系统，其中所述加载器还被配置成执行以下处理来确定所述组件未能通过所述第一完整性检查：

确定与所述组件相关联的完整性量度；

将所述完整性量度与关联于所述组件的可信基准值相比较；以及

确定所述完整性量度与所述可信基准值不匹配。

17.根据权利要求15所述的用于执行完整性检查的系统，其中所述PIPE还被配置成在从所述加载器接收到对所述发生故障的组件的指示时执行下列各项中的一项或多项：将所述系统断电、阻止加载所述发生故障的组件、阻止访问用以向所述网络实体验证所述无线通信设备的验证密钥、或者阻止访问安全功能。

18.根据权利要求15所述的用于执行完整性检查的系统，其中所述PIPE还被配置成核实设备管理功能，该设备管理功能被配置成：接收来自所述网络实体的替换组件；以及使用所述替换组件来替换所述发生故障的组件的所述部分。

19.根据权利要求15所述的用于执行完整性检查的系统，其中所述PIPE还被配置成接收来自外部实体的策略，并且将所述网络功能映射到所述策略中的一者或多者，其中所述策略被配置成基于所述第一完整性检查的结果来应用。

20.一种驻留在无线通信网络上的设备补救服务器，被配置成补救无线通信设备上的未能通过完整性检查的组件的部分，其中所述设备补救服务器被配置成：

从所述无线通信设备接收对与该无线通信设备上的未能通过所述完整性检查的组件相关联的网络功能的指示；

基于接收到的对所述网络功能的指示来确定发生故障的所述组件；

执行向所述无线通信设备的询问，以便隔离所述发生故障的组件的部分来进行补救；

基于一个或多个标准来确定用于所述发生故障的组件的所述部分的修复或替换；以及

将对所述发生故障的组件的所述部分的修复或替换发送到所述无线通信设备。

21.根据权利要求20所述的设备补救服务器，其中对所述发生故障的组件的所述部分的修复或替换包括软件代码。

22.根据权利要求20所述的设备补救服务器，其中该设备补救服务器还被配置成接收表明与所述完整性检查相关联的信息可靠的指示。

设备检验、遇险指示和补救

[0001] 相关申请的交叉引用

[0002] 本申请要求于2010年11月5日提交的美国临时专利申请61/410,781的权益,其中该申请的内容在这里全部引入作为参考。

背景技术

[0003] 无线通信系统可以使用多余的资源来执行完整性检查和/或修复该系统内部无线通信设备上的完整性故障。当前的完整性检查可以在大型单体码块上执行,以便确定无线通信设备的完整性是否受损。例如,通过完整性检查,可以在无线通信设备上检测到未经授权的修改、篡改和/或受损软件。一旦检测到很大的单体码块的完整性故障,则网络可以采用大型单个码块的形式来将更新下载至无线通信设备,以便修复故障。这样做有可能需要不必要的带宽,并且在系统上增加了不需要的计算负担。

[0004] 此外,多种类型和/或模型的无线通信设备可以用于与网络通信或是在网络上通信,并且其中每一个设备都具有不同的软件和硬件形式。由于硬件和软件开发实践有可能会随着公司而不同,因此,这些不同的设备可能会导致难以使无线通信设备上发生故障的组件的报告处理标准化。

发明内容

[0005] 本概述是为了以简化形式引入在以下的详细描述中被更进一步描述的不同概念而被提供的。

[0006] 在这里描述的是用于对无线通信设备的一个或多个组件执行完整性检查的系统、方法和设备实施例。这里描述的第一完整性检查可以在与无线通信设备相关联的组件上执行。该组件可被确定未能通过第一完整性检查。这时可以向网络实体发送一个关于与所述发生故障的组件相对应的功能的指示。从该网络实体可以接收一个在发生故障的组件上执行第二完整性检查以便确定发生故障的组件中导致该组件未通过第一完整性检查的部分的请求。所述第二完整性检查可以在发生故障的组件上执行,以便隔离所述发生故障的组件的部分,以通过网络实体来补救。

[0007] 根据一个例示实施例,加载器可以被配置成在与无线通信设备相关联的组件上执行第一完整性检查。该加载器还可以被配置成确定组件未通过第一完整性检查,并且在发生故障的组件上执行第二完整性检查,以便隔离发生故障的组件的部分,以通过网络实体来补救。

[0008] 根据一个例示实施例,平台完整性策略引擎(PIPE)可以被配置成接收来自加载器的关于发生故障的组件的指示,以及接收与发生故障的组件相对应的网络功能。该PIPE还可以被配置成向网络实体报告一个关于与发生故障的组件相对应的功能的指示,并且从网络实体接收在发生故障的组件上执行第二完整性检查以便确定发生故障的组件中导致该组件未通过第一完整性检查的部分的请求。

[0009] 根据一个例示实施例,设备补救服务器可以如这里所述。该设备补救服务器可以

驻留在无线通信网络上，并且被配置成补救无线通信设备上未能通过完整性检查的组件的部分。例如，设备补救服务器可以被配置成从无线通信设备接收一个关于与无线通信设备上未能通过完整性检查的组件相关联的网络功能的指示。所述发生故障的组件可以是基于接收到的关于网络功能的指示而被确定的。设备补救服务器还可以被配置成执行向无线通信设备的询问，以便通过隔离发生故障的组件的部分来进行补救。该设备补救服务器还可以被配置成基于一个或多个标准(criteria)来确定发送给无线通信设备的关于发生故障的组件的修复或替换。一旦基于该标准确定了修复或替换，则设备补救服务器可以将用于发生故障的组件的部分的修复或替换发送到无线通信设备。多种不同的标准可以用于确定发生故障的组件的修复或替换。例如，所述标准可以基于发生故障的组件的大小或其他某个因素。在一个例示实施例中，补救服务器可以基于OS软件版本来替换特定组件。其他例示标准可以包括但不局限于：版本号；每一个设备/组件/代码部分的最近更新或成功补救的日期/时间；代码或组件的所有权；代码许可的状况(例如数字权利管理)；发生故障的组件、比特或字节的数量；发生故障的代码部分的大小；以及所指定或计算的代码部分或组件的风险或损害影响值。

[0010] 本概述是为了以简化形式引入精选概念而被提供的，并且在以下的详细描述中将会进一步描述这些概念。本概述既不是为了确定所保护主题的关键特征或必要特征，也不是为了限定所保护的主题的范围。此外，所要保护的主题并不局限于解决在本公开的任何部分中指出的任何或所有缺陷的范围。

附图说明

[0011] 更详细的理解可以从以下结合附图举例给出的描述中得到，其中：

[0012] 图1A示出的是可以实施所公开的一个或多个实施例的例示通信系统；

[0013] 图1B示出的是可以实施所公开的一个或多个实施例的例示无线发射/接收单元；

[0014] 图1C示出的是可以实施所公开的一个或多个实施例的例示系统无线电接入网络；

[0015] 图2是示出了无线通信设备上的引导序列的例示实施例的图示；

[0016] 图3是示出了对象文件与可执行映像之间的联系的图示；

[0017] 图4是示出了对象文件与可执行映像之间的联系的另一个图示；

[0018] 图5示出的是文件的组件TRV区段的一个示例。

[0019] 图6是示出了组件-网络功能映射的图示；

[0020] 图7示出的是文件的组件-功能映射区段的示例；

[0021] 图8是示出了在引导序列中使用的平台能力策略初启(bring-up)的图示；

[0022] 图9是示出了这里描述的在完整性检查和/或报告期间使用表格或映射的图示；

[0023] 图10是示出了这里描述的报告和补救处理的例示综述的图示；

[0024] 图11示出的是通过在设备上收集信息来确定可以执行的操作的例示呼叫流程图；

[0025] 图12示出的是用于在设备与网络实体之间执行询问的例示呼叫流程图；

[0026] 图13示出的是在无线通信设备的组件上执行的询问的示例；

[0027] 图14示出的是在无线通信设备组件上执行的询问的另一个示例；

[0028] 图15示出的是与遇险警报以及单体代码替换相关的例示呼叫流程图；

[0029] 图16示出的是与远程软件遇险/补救相关联的例示呼叫流程图；

[0030] 图17示出的是与实施SeGW验证尝试的远程软件遇险/补救相关联的例示呼叫流程图；

[0031] 图18A示出的是与远程软件遇险/补救相关联的例示呼叫流程图，其中网络可以禁止经由SeGW的验证；

[0032] 图18B示出的是与实施即时限制访问和精确访问控制的远程软件遇险/补救相关联的例示呼叫流程图；

[0033] 图19示出的是将无线通信设备软件组件补救与SeGW访问相关联的例示呼叫流程图；

[0034] 图20示出的是与中继节点的能力自举(bootstrap)相关联的例示呼叫流程图；

[0035] 图21示出的是与具有经过验证的管理能力的中继节点补救相关联的例示呼叫流程图；

[0036] 图22示出的是具有功能组件和代码/数据存储组件的例示系统；

[0037] 图23示出的是引导序列的阶段以及在每一个阶段实施的不同实体间的交互的例示流程图；

[0038] 图24A示出的是被线性组合以创建TRV的量度序列的图示；以及

[0039] 图24B示出的是使用Merkle散列树来创建TRV的值的组合的图示。

具体实施方式

[0040] 图1A-24B涉及的是可以实施所公开的系统、方法和手段的例示实施例。这里描述的实施例是例示性而不是限制性的。虽然在这里示出并描述了协议流程，但是这些流程的顺序是可以改变的，和/或附加流程是可以添加的。

[0041] 图1A、1B和1C示出的是可以在这里描述的实施例中使用的例示通信系统和设备。图1A是可以实施所公开的一个或多个实施例的例示通信系统100的图示。通信系统100可以是为多个无线用户提供语音、数据、视频、消息传递、广播等内容的多址接入系统。该通信系统100通过共享包括无线带宽在内的系统资源来允许多个无线用户访问此类内容，举例来说，通信系统100可以使用一种或多种信道接入方法，如码分多址(CDMA)、时分多址(TDMA)、频分多址(FDMA)、正交FDMA(OFDMA)、单载波FDMA(SC-FDMA)等等。

[0042] 如图1A所示，通信系统100可以包括无线发射/接收单元(WTRU)102a、102b、102c、102d，无线电接入网络(RAN)104，核心网络106，公共交换电话网络(PSTN)108，因特网110以及其他网络112，但是应该了解，所公开的实施例设想了任意数量的WTRU、基站、网络和/或网络部件。WTRU102a、102b、102c、102d中的每一者可以是被配置成在无线环境中工作和/或通信的任何类型的设备。例如，WTRU102a、102b、102c、102d可以被配置成发射和/或接收无线信号，并且可以包括用户设备(UE)、移动站、固定或移动订户单元、寻呼机、蜂窝电话、个人数字助理(PDA)、智能电话、膝上型计算机、上网本、个人计算机、无线传感器、消费类电子设备等等。

[0043] 通信系统100还可以包括基站114a和基站114b。基站114a、114b中的每一者可以是被配置成通过与WTRU102a、102b、102c、102d中的至少一个无线对接来促成对一个或多个通信网络的访问，例如核心网络106、因特网110和/或网络112。例如，基站114a、114b可以是基地收发信台(BTS)、节点B、e节点B、家用节点B、家用e节点B、站点控制器、接入点(AP)、无线

路由器等等。虽然每一个基站114a、114b都被描述成是单个部件，但是应该了解，基站114a、114b可以包括任何数量的互连基站和/或网络部件。

[0044] 基站114a可以是RAN104的一部分，并且所述RAN还可以包括其他基站和/或网络部件(未显示)，例如基站控制器(BSC)、无线电网络控制器(RNC)、中继节点等等。基站114a和/或基站114b可以被配置成在被称之为为小区(未显示)的特定地理区域内部发射和/或接收无线信号。小区可以进一步分成小区扇区。例如，与基站114a相关联的小区可以分成三个扇区。因此，在一个实施例中，基站114a可以包括三个收发信机，也就是说，每一个收发信机对应于小区的一个扇区。在另一个实施例中，基站114a可以使用多输入多输出(MIMO)技术，由此可以为小区中的每个扇区使用多个收发信机。

[0045] 基站114a、114b可以经由空中接口116来与一个或多个WTRU102a、102b、102c、102d进行通信，其中该空中接口可以是任何适当的无线通信链路(例如射频(RF)、微波、红外线(IR)、紫外线(UV)、可见光等等)。空中接口216可以采用任何适当的无线电接入技术(RAT)来建立。

[0046] 更具体地说，如上所述，通信系统100可以是一个多址接入系统，并且可以使用一种或多种信道接入方案，如CDMA、TDMA、FDMA、OFDMA、SC-FDMA等等。举例来说，RAN104中的基站114a和WTRU102a、102b、102c可以实施诸如通用移动电信系统(UMTS)陆地无线电接入(UTRA)之类的无线电技术，其中该技术可以使用宽带CDMA(WCDMA)来建立空中接口116。WCDMA可以包括下列通信协议，如高速分组接入(HSPA)和/或演进型HSPA(HSPA+)。HSPA可以包括高速下行链路分组接入(HSDPA)和/或高速上行链路分组接入(HSUPA)。

[0047] 在另一个实施例中，基站114a和WTRU102a、102b、102c可以实施诸如演进型UMTS陆地无线电接入(E-UTRA)之类的无线电技术，该技术可以使用长期演进(LTE)和/或高级LTE(LTE-A)来建立空中接口116。

[0048] 在其他实施例中，基站114a与WTRU102a、102b、102c可以实施IEEE802.16(全球微波接入互操作性(WiMAX))、CDMA2000、CDMA20001X、CDMA2000EV-DO、临时标准2000(IS-2000)、临时标准95(IS-95)、临时标准856(IS-856)、全球移动通信系统(GSM)、用于GSM演进的增强数据速率(EDGE)、GSM EDGE(GERAN)等无线电接入技术。

[0049] 图1A中的基站114b可以是例如无线路由器、家用节点B、家用e节点B或接入点，并且可以使用任何适当的RAT来促成局部区域中的无线连接，例如营业场所、住宅、交通工具、校园等等。在一个实施例中，基站114b和WTRU102c、102d可以通过实施诸如IEEE802.11之类的无线电技术来建立无线局域网(WLAN)。在另一个实施例中，基站114b和WTRU102c、102d可以通过实施诸如IEEE802.15之类的无线电技术来建立无线个人局域网(WPAN)。在再一个实施例中，基站114b和WTRU102c、102d可以通过使用基于蜂窝的RAT(例如WCDMA、CDMA2000、GSM、LTE、LTE-A等等)来建立微微小区或毫微微小区。如图1A所示，基站114b可以直接连接到因特网110。由此，基站114b未必需要经由核心网络106来接入因特网110。

[0050] RAN104可以与核心网络106通信，所述核心网络可以是被配置成向一个或多个WTRU102a、102b、102c、102d提供语音、数据、应用和/或借助网际协议的语音(VoIP)服务的任何类型的网络。例如，核心网络106可以提供呼叫控制、记账服务、基于移动位置的服务、预付费呼叫、因特网连接、视频分发等等，和/或执行高级安全功能，例如用户验证。虽然在图1A中没有显示，但是应该了解，RAN104和/或核心网络106可以直接或间接地和其他那些

与RAN104使用相同RAT或不同RAT的RAN进行通信。例如，除了与可以使用E-UTRA无线电技术的RAN104相连之外，核心网络106还可以与另一个使用GSM无线电技术的RAN(未显示)通信。

[0051] 核心网络106还可以充当WTRU102a、102b、102c、102d接入PSTN108、因特网110和/或其他网络112的网关。PSTN108可以包括提供简易老式电话服务(POTS)的电路交换电话网络。因特网110可以包括使用公共通信协议的全球性互联计算机网络设备系统，所述协议可以是TCP/IP互连网协议族中的传输控制协议(TCP)、用户数据报协议(UDP)和网际协议(IP)。网络112可以包括由其他服务供应商拥有和/或运营的有线或无线通信网络。例如，网络112可以包括与一个或多个RAN相连的另一个核心网络，其中所述一个或多个RAN可以与RAN104使用相同RAT或不同的RAT。

[0052] 通信系统100中一些或所有WTRU102a、102b、102c、102d可以包括多模能力，换言之，WTRU102a、102b、102c、102d可以包括在不同无线链路上与不同无线网络通信的多个收发信机。例如，图1A所示的WTRU102c可以被配置成与使用基于蜂窝的无线电技术的基站114a通信，以及与可以使用IEEE802无线电技术的基站114b通信。

[0053] 图1B是例示WTRU102的系统图示。如图1B所示，WTRU102可以包括处理器118、收发信机120、发射/接收部件122、扬声器/麦克风124、键盘126、显示器/触摸板128、不可拆卸存储器130、可拆卸存储器132、电源134、全球定位系统(GPS)芯片组136以及其他外围设备138。应该了解的是，在保持符合实施例的同时，WTRU102可以包括前述部件的任何子组合。

[0054] 处理器118可以是通用处理器、专用处理器、常规处理器、数字信号处理器(DSP)、多个微处理器、与DSP核心关联的一个或多个微处理器、控制器、微控制器、专用集成电路(ASIC)、现场可编程门阵列(FPGA)电路、其他任何类型的集成电路(IC)、状态机等等。处理器118可以执行信号编码、数据处理、功率控制、输入/输出处理和/或其他任何能使WTRU102在无线环境中工作的功能。处理器118可以耦合至收发信机120，收发信机120可以耦合至发射/接收部件122。虽然图1B将处理器118和收发信机120描述成是独立组件，但是应该了解，处理器118和收发信机120可以一起集成在一个电子封装或芯片中。

[0055] 发射/接收部件122可以被配置成经由空中接口116来发射或接收去往或来自基站(例如基站114a)的信号。例如，在一个实施例中，发射/接收部件122可以是被配置成发射和/或接收RF信号的天线。在另一个实施例中，举例来说，发射/接收部件122可以是被配置成发射和/或接收IR、UV或可见光信号的发射器/检测器。在再一个实施例中，发射/接收部件122可以被配置成发射和接收RF和光信号。应该了解的是，发射/接收部件122可以被配置成发射和/或接收无线信号的任何组合。

[0056] 此外，虽然在图1B中将发射/接收部件122描述成是单个部件，但是WTRU102可以包括任何数量的发射/接收部件122。更具体地说，WTRU102可以使用MIMO技术。因此，在一个实施例中，WTRU102可以包括两个或更多个经由空中接口116来发射和接收无线电信号的发射/接收部件122(例如多个天线)。

[0057] 收发信机120可以被配置成对发射/接收部件122将要发射的信号进行调制，以及对发射/接收部件122接收的信号进行解调。如上所述，WTRU102可以具有多模能力。因此，收发信机120可以包括允许WTRU102借助UTRA和IEEE802.11之类的多种RAT来进行通信的多个收发信机。

[0058] WTRU102的处理器118可以耦合至扬声器/麦克风124、键盘126和/或显示器/触摸

板128(例如液晶显示器(LCD)显示单元或有机发光二极管(OLED)显示单元),并且可以接收来自这些部件的用户输入数据。处理器118还可以向扬声器/麦克风124、键盘126和/或显示器/触摸板128输出用户数据。此外,处理器118可以从任何适当的存储器(例如不可拆卸存储器130和/或可拆卸存储器132)中访问信息,以及将信息存入这些存储器。所述不可拆卸存储器130可以包括随机存取存储器(RAM)、只读存储器(ROM)、硬盘或是其他任何类型的记忆存储设备。可拆卸存储器132可以包括订户身份模块(SIM)卡、记忆棒、安全数字(SD)记忆卡等等。在其他实施例中,处理器118可以从那些并非实际位于WTRU102的存储器访问信息,以及将数据存入这些存储器,其中举例来说,所述存储器可以位于服务器或家用计算机(未显示)。

[0059] 处理器118可以接收来自电源134的电力,并且可以被配置分发和/或控制用于WTRU102中的其他组件的电力。电源134可以是为WTRU102供电的任何适当的设备。举例来说,电源134可以包括一个或多个干电池组(如镍镉(NiCd)、镍锌(NiZn)、镍氢(NiMH)、锂离子(Li-ion)等等)、太阳能电池、燃料电池等等。

[0060] 处理器118还可以与GPS芯片组136耦合,该芯片组可以被配置成提供与WTRU102的当前位置相关的位置信息(例如经度和纬度)。作为来自GPS芯片组136的信息的补充或替换,WTRU102可以经由空中接口116接收来自基站(例如基站114a、114b)的位置信息,和/或根据从两个或更多个附近基站接收的信号的定时来确定其位置。应该了解的是,在保持符合实施例的同时,WTRU102可以借助任何适当的定位方法来获取位置信息。

[0061] 处理器118还可以耦合到其他外围设备138,这其中可以包括提供附加特征、功能和/或有线或无线连接的一个或多个软件和/或硬件模块。例如,外围设备138可以包括加速度计、电子指南针、卫星收发信机、数码相机(用于照片或视频)、通用串行总线(USB)端口、振动设备、电视收发信机、免提耳机、蓝牙®模块、调频(FM)无线电单元、数字音乐播放器、媒体播放器、视频游戏播放器模块、因特网浏览器等等。

[0062] 图1C是根据一个实施例的RAN104和核心网络106的系统图示。如上所述,RAN104可以使用E-UTRA无线电技术以经由空中接口116来与WTRU102a、102b、102c进行通信。并且该RAN104还可以与核心网络106通信。

[0063] RAN104可以包括e节点B140a、140b、140c,但是应该理解,在保持与实施例相符的同时,RAN104可以包括任意数量的e节点B。节点B140a、140b、140c中的每一者都可以包括一个或多个收发信机,以便经由空中接口116来与WTRU102a、102b、102c进行通信。在一个实施例中,e节点B140a、140b、140c可以实施MIMO技术。因此,举例来说,e节点B140a可以使用多个天线来向WTRU102a发射无线信号以及接收来自WTRU102a的无线信号。

[0064] e节点B140a、140b、140c中的每一者都可以与特定的小区(未显示)相关联,并且可以被配置成处理无线电资源管理决策、切换决策、上行链路和/或下行链路中的用户调度等等。如图1C所示,e节点B140a、140b、140c彼此可以经由X2接口来进行通信。

[0065] 图1C所示的核心网络106可以包括移动性管理网关(MME)142、服务网关144以及分组数据网络(PDN)网关146。虽然在前的每一个部件都被描述成是核心网络106的一部分,但是应该了解,这其中的任一部件都可以被核心网络运营商以外的实体拥有和/或运营。

[0066] MME142可以经由S1接口来与RAN104中的每一个e节点B140a、140b、140c相连,并且可以充当控制节点。例如,MME142可以负责验证WTRU102a、102b、102c的用户,激活/去激活

承载，在WTRU102a、102b、102c的初始附着过程中选择特定服务网关等等。MME142还可以提供控制平面功能，以便在RAN104与使用了诸如GSM或WCDMA之类的其他无线电技术的其他RAN(未显示)之间进行切换。

[0067] 服务网关144可以经由S1接口而与RAN104中的每一个e节点B140a、140b、140c相连。该服务网关144通常可以路由和转发去往/来自WTRU102a、102b、102c的用户数据分组。该服务网关144还可以执行其他功能，例如在e节点B间的切换过程中锚定用户面，在下行链路数据可供WTRU102a、102b、102c使用时触发寻呼，管理和存储WTRU102a、102b、102c的上下文等等。

[0068] 服务网关144还可以连接到PDN网关146，该PDN网关可以为WTRU102a、102b、102c提供针对因特网之类的分组交换网络的接入，以便促成WTRU102a、102b、102c与启用IP的设备之间的通信。

[0069] 核心网络106可以促成与其他网络的通信。例如，核心网络106可以为WTRU102a、102b、102c提供针对PSTN108之类的电路交换网络的接入，以便促成WTRU102a、102b、102c与传统的陆线通信设备之间的通信。例如，核心网络106可以包括IP网关(例如IP多媒体子系统(IMS)服务器)或与之通信，其中该IP网关充当的是核心网络106与PSTN108之间的接口。此外，核心网络106可以为WTRU102a、102b、102c提供针对网络112的接入，该网络可以包括其他服务供应商拥有和/或运营的其他有线或无线网络。

[0070] 这里描述的通信系统和设备可以用于管理通信设备的软件，管理通信设备的配置，和/或提供软件和/或配置信息补救，以便将设备恢复到原始状态。此外，在这里还描述了使用软件开发和代码发布工具的实施方式，其中所述软件开发和代码发布工具可以使用软件工具、网络协议、设备策略以及软件管理和/或远程调试技术来自动生成和管理可信代码库(code base)的软件方面的基准，从而在设备中嵌入用于可证实报告和补救的机制和基准。更进一步，在这里描述的是可以确保设备实施的完整性检查的故障指示的技术，其中包括对表明可以信任报告故障的设备的指示所进行的描述。

[0071] 这里描述的无线通信设备可以被配置成在具有多个阶段的安全引导处理的每个阶段执行完整性检查。在多个阶段的安全引导处理过程中，每一个阶段可以核实后续阶段，由此创建一个信任链。在具有多个阶段的安全引导处理的每个阶段中可以确定是否可以信任与该阶段相关联的组件。与组件相关联的完整性量度可被确定。该组件可以具有相关联的可信基准值。组件的可信赖度可以通过将完整性量度与可信基准值相比较来确定(例如测试)。如果完整性量度与可信基准值相匹配，则可以认为该组件是可信赖的。如果完整性量度未能与可信基准值相匹配，则认为该组件未必是可信赖的。虽然在这里描述的是通过将完整性量度与可信基准值相比较来执行完整性检查，但是应该了解，完整性检查是可以采用其他那些用于确定组件可信赖度的方式执行的。

[0072] 由于外部实体未必会辨认出组件，因此，举例来说，在多个网络和/或设备上可以定义和/或标准化功能，从而与标准规范和/或取决于实施的方式的相一致。组件可以与功能相关联。组件与功能之间的关系可以在组件-功能映射中给出。功能故障可以通过将发生故障的组件与组件-功能映射相比较来识别。

[0073] 设备可以向外部实体发送与功能故障相关联的警报。外部实体可以确定与功能相关联且可以用于修复或替换发生故障的组件的替换代码。换言之，替换代码可以是一个替

换组件。设备可以接收该替换组件，并且使用该替换组件来替换发生故障的组件。

[0074] 安全引导处理可以提供检验过程的基础。由不可变硬件可信根(RoT)发起的信任链可以核实所加载的初始代码的有效性。如图2所示，举例来说，该引导处理可以在每一个阶段通过信任链核实后续阶段的时候继续进行。

[0075] 在通电以及硬件初始化过程之后，设备可以启动一个例如图2所示的安全引导处理。初始的RoT可以用驻留于ROM中的安全存储器的引导加载器202表示。在一开始，ROM引导加载器202可以执行一些初始化功能。该ROM引导加载器202可访问安全凭证(例如融合键控信息)，并且可以使用该信息来核实第二阶段的引导加载器204(例如驻留在外部存储器中)的完整性。第二阶段的加载器204可以由硬件或软件密码加密装置进行检查，以确保其完整性。计算得到的第二阶段的加载器204的散列(hash)量度可以与一个用安全凭证签名并保存在外部存储器中的可信基准值(TRV)量度相比较。如果计算得到的量度与带签名的TRV相匹配，则第二阶段的加载器204可被核实并加载到内部存储器(例如RAM)中。该引导ROM可以跳转到第二阶段的加载器204的开端，并且信任链可以继续。所述核实处理中的初始阶段的故障可以表明后续核实有可能受损(例如，初始阶段的故障可以表明信任链发生了重大故障)。如果存在在核实处理的初始阶段指示出的故障，则可以将设备关闭。

[0076] 第二阶段的引导加载器204可以包括可信执行环境(TrE)代码以及可以检查附加代码并且将其加载到内部存储器的代码。TrE可以建立一个能够计算和存储附加的完整性基准值的可信环境和安全存储区域。该TrE完整性可以检查、加载和/或启动操作系统以及通信代码。随着每个阶段被检查和加载(例如，图2所示的方框202、204、206和208中的每一者都可以表示一个阶段)，所述信任链可以延续。例如，第二阶段的加载器204可以核实OS/通信(Comm)206和/或设备软件208的完整性。作为替换或补充，OS/通信206可以核实设备软件208的完整性。所述OS/通信206可以包括一个OS加载器。根据一个实施例，图2所示的实体可以按照执行顺序而被核实和/或相互核实，以便保持信任链。

[0077] 信任链可以由一个能够安全核实后续处理的执行过程来保持。该核实过程可以使用密码加密计算和/或可信基准值。驻留在非安全存储器中的代码可能易受攻击，并且可以在加载之前对其进行检查。在没有细粒度检验的情况下，第二阶段的引导加载器204可以检验剩余代码，以此作为体积(bulk)量度。如果测量得到的值不与单体块的TRV匹配，那么可以确定所述代码未必可信以及不可以加载该代码。设备未必能够通信，并且一个结果有可能包括借助高成本的上门服务或是运回设备进行修复来补救整个代码库。

[0078] 这里描述的方法、系统和手段可以使得设备能够检验已存储代码的较小组件，以及提供关于哪些组件出现故障以及哪些组件可以或者不可以被远程补救的详细描述。这种细粒度信息可被提供给补救管理器。

[0079] 在这里可以建立策略来确定哪些发生故障的组件是可以远程补救的。举个例子，如果设备检查并加载了TrE以及最小的通信代码集合，那么该设备可以保持充当用于补救管理器的代理的功能，以便识别以及向补救管理器报告发生故障的特定组件的信息。如果补救能力存在且属于设备的代理功能的一部分，则可以发起后续的远程补救过程，这样做可以减少使用高成本的现场人员更新。

[0080] 根据一个实施例，可执行映像可被分区和检验，以便能够实施细粒度的报告处理。可执行映像的生成可以在几个阶段中进行。组件可以是指一个包含了子程序和/或参数/数

据的文件。开发人员可以编写一个在组件源和头文件中捕获的程序。从这些组件源文件中，编译器和汇编器可以产生同时包含了机器二进制码和程序数据的目标文件(例如*.o)。链接器可以将这些目标文件作为输入，并且产生一个能够用于与其他目标文件附加链接的可执行映像或目标文件。链接器命令文件可以指示链接器如何组合对象文件，以及将二进制码和数据放置在目标嵌入式系统中的什么位置。

[0081] 链接器的功能可以是将多个对象文件合并成较大的可重新定位的对象文件、共享的对象文件或是最终的可执行映像。全局变量和非静态函数可被称为全局符号。在最终的可执行二进制映像中，符号可以是指存储器中的地址位置。该存储器位置的内容可以是用于变量的数据或是用于函数的可执行代码。

[0082] 编译器可以创建一个具有符号名称-地址映射的符号表，以此作为其产生的对象文件的一部分。在创建可重新定位的输出的时候，编译器可以产生地址，其中对于每一个符号来说，所述地址与所编译的文件是关联的。链接器执行的链接处理可以包括符号解析和/或符号重新定位。符号解析可以是链接器检查每一个目标文件并且为目标文件确定在哪个(哪些)(其他)目标文件中定义外部符号。符号重新定位可以是链接器将符号引用映射到其定义的处理。该链接器可以修改所链接的对象文件的机器码，由此，对符号的代码引用反映了指定给这些符号的实际地址。

[0083] 对象和可执行文件可以采用若干种格式，例如ELF(可执行和链接格式)以及COFF(公共对象-文件格式)。对象文件可被划分成区域或区段。这些区段可以保持下列各项中的一项或多项：例如可执行代码、数据、动态链接信息、调试数据、符号表、重新定位信息、注释、字串表或注解。这些区段可以用于向二进制文件提供信息，以及允许检查。函数区段可以包括下列各项中的一项或多项：可以存储系统函数地址的全局偏移表(GOT)、可以存储至GOT的间接链接的过程链接表(PLT)、可以用于内部初始化的.init/fini、以及可以用于构造器和析构器的.shutdown或.ctors/.dtors。数据区段可以包括下列各项中的一项或多项：用于只读数据的.rodata，用于已初始化数据的.data，或是用于未初始化数据的.bss。ELF区段可以包括下列各项中的一项或多项：用于启动的.init、用于字串的.text、用于停机的.fini、用于只读数据的.rodata、用于已初始化数据的.data、用于已初始化的线程数据的.tdata、用于未初始化线程数据的.tbss、用于构造器的.ctors、用于析构器的.dtors、用于全局偏移表的.got、或用于未初始化数据的.bss。

[0084] 一些区段可被加载到进程映像中。一些区段可以提供在构建进程映像的过程中使用的信息。一些区段可被限制成在链接对象文件的过程中使用。一些对象文件可以包括对位于其他对象文件中的符号的引用。该链接器可以创建一个符号表，所述符号表可以包括符号名称的列表及其在用于每一个对象文件的文本和数据分段中的相应偏移。在将对象文件链接在一起之后，链接器可以使用重新定位记录来找出可能未被填充的未解析符号地址。在编译了多个源文件(例如C/C++和汇编文件)并将其汇编成ELF对象文件之后，链接器可以组合这些对象文件，并且将来自不同对象文件的区段并入如图3所示的程序分段。

[0085] 如图3所示，对象文件302、304和306的文本区段308、316和324分别可以并入可执行映像338的文本分段332。作为补充或替换，启动区段310、318和326可以并入可执行映像338的文本分段332。同样，对象文件302、304和306的数据区段312、320和328分别可以并入可执行映像338的数据分段334。最后，对象文件302、304和306的未初始化数据区段314、322

和330分别可以并入可执行映像338的未初始化数据分段336。虽然图3示出了可以并入可执行映像338的区段的对象文件302、304和306的多个区段，但是应该理解，任意数量和/或组合的区段都可以并入可执行映像的区段。

[0086] 如图3所示，区段可以提供一种将相关区段归组的方式。每一个分段都可以包括具有相同或不同类型(例如文本、数据或动态区段)的一个或多个区段。操作系统在逻辑上可以依照在程序头表中提供的信息来拷贝文件分段。所述可执行和只读数据区段可以组合成单个文本区段。数据和未初始化的数据区段可以组合成数据分段或组合成其自己的单独的分段。由于这些分段可以在创建进程的时候加载到存储器中，因此，这些分段可被称为加载分段。诸如符号信息和调试区段之类的其他区段可以并入到其他的非加载分段。文本区段可以包括源代码以及已初始化的静态变量。编译器定义的多个区段可被加载到链接器定义的单个组合分段中。

[0087] 最终的可执行映像可以是用那些控制链接器如何组合区段和/或将区段分配到目标系统存储器映射的链接器命令(它可以是被叫链接器指令)产生的。链接器指令可以保持在链接器命令文件中。嵌入式开发人员可以使用该链接器命令文件来将可执行映像映射到目标系统。

[0088] 对发生故障的组件执行检验和细粒度报告的能力可以使用可执行映像中提供的附加信息，此外，开发和代码发布工具链是可以修改的。在可执行映像内部，每一个组件的基准量度以及识别所述量度所关联的组件的信息元素都是可以识别的。

[0089] 对发生故障的组件所进行的补救可以取决于设备报告标准的功能故障的能力。功能以及基于故障所采取的操作可以由网络运营商定义。软件开发者可以确定如何能将运营商定义的功能映射到其系统中的软件组件。在可执行映像中，功能映射是可见的，并且可以使用工具链增强。

[0090] 通过修改开发阶段使用的软件工具链，可以适应下列各项中的一项或多项：组件区段生成、安全的TRV生成和嵌入、组件-功能定义的插入、或策略及策略配置文件的插入。

[0091] 通过修改加载器的功能，可以适应检验和/或补救。加载器可被修改成执行下列各项中的一项或多项：在加载组件时对组件执行完整性检查、隔离或卸载发生故障的组件、管理策略、或在存储器中保存发生故障的组件ID以供策略管理器报告故障。

[0092] 在这里描述了可被应用于这里描述的软件开发和代码发布工具链来例证预期功能的例示实施例。然而，这里的实施例并不局限于这里提供的示例。

[0093] 链接器的作用可以是获取输入对象，以及将不同的区段组合成分段。由于可以在链接过程期间组合对象代码，因此，在符号表信息中可以保持识别对象文件中的函数或变量的能力。最终的合并代码区段可不提供可以用于触发这里描述的检验过程的检验方案。

[0094] 为了便于识别单个组件，加载器可以具有一种识别代码起源的方式，其中可以包括定义组件或是生成可被加载器识别的单个组件的TRV。例如，通过增强用于生成可执行映像的工具链，能够识别出与加载器检验功能的特定对象文件组件相关联的代码，从而识别特定的组件以及执行完整性检查。代码映像的重新排列可以改变输入对象的TRV(例如，如果修改一个输入对象，那么有可能导致改变其他输入对象的TRV)。此外还可以阻止工具链使用优化方法，以便阻止TRV的变化。

[0095] 对于需要完整性检查的组件来说，在这里可以为其生成特定区段名称。例如，区段

名称可以出现在最终的ELF可执行文件包含的区段头表中。通过归组用户定义的区段，加载器可以识别组件以及执行完整性检查。所述加载器可以向策略管理器报告组件的通过与否的状态。通过隔离那些通过/未能通过完整性检查的特定功能，可以允许策略管理器以精细的粒度或详细的等级来报告那些可能受到发生故障的组件影响的功能。

[0096] 图4示出了两个可以链接在一起形成单个可执行映像406的对象文件402和404。对象文件402具有两个不同的代码分段，即组件文件区段408和文本区段410。对象文件404具有可以包括代码和/或恒定数据的单个代码分段，即组件文件区段416。对用户定义的代码区段、即组件文件区段408和组件文件区段416来说，这些区段可以连同与其映射的分段类型相关的大小和/或存储位置一起出现的区段报头中。在图4中用箭头描述了关于这种映射关系的一个示例。例如，对象文件402的组件文件区段408、文本区段410、数据区段412以及bss区段414分别可被映射到可执行映像406的组件文件区段420、文本区段424、数据区段426以及bss区段428。同样，对象文件404的组件文件区段416和bss区段418分别可以被映射到可执行映像406的组件文件区段422以及bss区段428。通过链接可执行映像406，可以在可执行映像406开端的预定位置包含用户定义的区段，由此可以按顺序对其进行检查。借助于检验的完整性检查可以仅限于代码库的一个子集。剩余那些被归组成较大文本分段的代码可以不被执行完整性检查，和/或是可以加载的。

[0097] 用户定义的区段的添加可以是可识别的，例如通过在组件代码中插入#_PRAGMA。仍旧参考图4的示例，在源文件的顶部可以插入指令#_PRAGMA区段（例如组件文件区段408）。通过增强编译器，可以借助编译器标记来包含用户定义的区段。如果设置了特定的标记，那么可以通过增强编译器来自动插入PRAGMA。此外，用户定义的区段的概念可以扩展，以便将用户定义的区段局限于那些可能被完整性检查的组件内部的功能。可用于设备的可信执行的功能可被分区或隔离成在启动过程中首先或者早期会被完整性检查的组件。

[0098] 区段可以借助链接器命令文件而被映射到存储器的特定分段。ELF对象可以依照源和目的地址来查看。在区段头表中标识的区段可以向加载器告知在哪儿发现所要移动的代码和/或数据的开端。在分段报头中标识的分段可以向加载器告知将组件拷贝至何处。在下文中示出了区段和程序报头的格式。

[0099]

区段报头

```
typedef struct {
Elf32_Word sh_name;
Elf32_Word sh_type;
Elf32_Word sh_flags;
Elf32_Addr sh_addr;
Elf32_Off sh_offset;
Elf32_Word sh_size;
Elf32_Word sh_link;
```

程序报头

```
typedef struct {
Elf32_Word p_type;
Elf32_Off p_offset;
Elf32_Addr p_vaddr;
Elf32_Addr p_paddr;
Elf32_Word p_filesz;
Elf32_Word p_memsz;
Elf32_Word p_flags;
```

区段报头

Elf32_Word sh_info;
Elf32_Word sh_addralign;
Elf32_Word sh_entsize;
} Elf32_Shdr;

程序报头

Elf32_Word p_align;
} Elf32_Phdr;

[0100]

[0101] sh_addr是程序区段可以在目标存储器中驻留的地址。p_addr可以是程序分段在目标存储器中驻留的地址。sh_addr和p_paddr字段可以是指加载地址。加载器可以使用来自区段报头的加载地址作为将映像从非易失存储器传送到RAM的起始地址。

[0102] 组件区段标识的概念可被扩展至文件内部包含的一个以上的特定组件。该扩展可以通过识别将被执行完整性检查的特定子例程而不是整个文件来允许策略管理器所实施的访问控制的进一步的粒度。

[0103] TRV可以是特定组件或对象文件的预期量度(例如采用安全方式计算的组件的散列)。出于完整性核实的目的,举例来说,检验处理可以依靠将被检查的存在于可执行映像中的或是分别以安全方式加载的每一个组件的TRV。该处理可以采用多种方式来实现。作为例证,通过修改构建可执行映像的工具链,可以安全地计算组件或对象文件的散列值,并且安全地将计算得到的TRV插入同一对象文件的恰当区段。该计算可以作为链接过程的一部分来执行。计算TRV散列的顺序可以与加载和量度组件的顺序匹配,否则有可能无法正确匹配量度值。该散列值可以被加载器得到或是包含在可执行映像内部。为了保持信任链,在这里可以安全地生成和/或存储TRV。举例来说,通过使用与诸如软件/固件制造商的公钥之类的公钥相对应的私钥来对TRV值进行签名,可以保护TRV。

[0104] 组件对象文件可以包括单独的用户定义区段,以此作为如图5所示的与之关联的TRV的占位符。举个例子,如图5所示,组件目标文件408的TRV可被计算并保存在组件TRV区段502中。区段报头可以识别该区段的起始地址和/或尺寸,其中所述区段包括这里描述的用户定义区段。在对象文件402上,链接器可以在符号解析阶段结束时计算组件对象文件408的散列值,以及定位相应的组件TRV区段502。该组件TRV可被插入到处于存储器中的指定位置的可执行映像(例如组件TRV区段)。区段报头可以允许加载器在检验处理过程中定位特定区段的TRV。

[0105] 这里描述的设备检验可以关注于组件的完整性状态。由于软件的开发实践有可能会随着公司的不同而不同,因此有可能很难使关于发生故障的软件的报告处理标准化。

[0106] 在这里公开了用于将对照软件执行的功能来归组组件的方式标准化的系统、方法和手段。通过将功能标准化,可以使其与标准规范相一致和/或采用与实施方式无关的方式。在这里可以为可被标准化的设备功能预先定义一个功能列表。通过使用功能与组件之间的关联,可以允许将完整性检查期间发现的故障映射到特定功能以进行报告,并且在图6中描述了它的一个示例。

[0107] 如图6所示,信任域602可以包括相互映射的组件和功能。例如,可信任域602可以包括高级功能604、中间功能606和/或可信环境(TrE)核心功能608。高级功能604可以包括功能610和功能612。中间功能606可以包括功能614和功能616。TrE核心功能608可以包括功能618,例如RoT。功能612可以映射到组件620和/或622,其中举例来说,该组件可以是软件组件。功能616可以映射到组件624,并且举例来说,该组件同样可以是软件组件。功能618可

以映射到组件626，其中举例来说，该组件可以是固件组件。在对可信任域602执行完整性检查期间，在组件624上可能会确定一个故障。由于组件624映射到了功能616，因此可以确定网络功能616同样存在故障。由此，设备可以向网络发送关于发生故障的功能616的指示，以便进行补救。

[0108] 图6的功能与组件之间的映射是作为执行层映射而被示出的。作为图6示出的实施例的替换或补充，组件与功能之间的映射可以具有一个以上的层。可以使用具有两个以上的层的数据结构(例如，其中一个层用于组件，另一个层用于功能)。在该结构中，组件可被映射到一组子功能，并且子功能可以被映射到一组更高级的子功能。中间映射可以持续进行，直至处于最终子功能层的子功能被映射至处于最终层的最终功能。树或类似于树的数据结构可以是能够获取这种多层组件-功能映射关系的结构的一个示例。

[0109] 开发者可以确定组件如何映射至标准化功能。根据一个实施例，可执行映像可以包括组件-功能映射。举例来说，这种映射可以在编译时通过一个图形工具来实现，其中该图形工具对编译代码或源代码进行解析，显示了单个文件布局、文件中的函数相互依存性中的一项或多项，以及允许将组件映射到功能。功能可以是文本字段和/或缩写ID号码，其中所述字段和/或号码可以由用户输入和/或手动映射到组件/文件。开发工具可以创建一个引用了组件-功能映射表的区段。组件名称、功能名称以及ID连同交叉引用的互连一起可以作为所述表的元素而被包含。

[0110] 图7示出的是包含组件-功能映射的区段的一个示例。如图7所示，功能-组件映射区段702可以作为对象文件402之类的对象文件的区段而被包含。此外，如这里所述，功能-组件映射区段702还可以链接到可执行映像中的相应分段。

[0111] 加载器的功能可以是将代码从外部存储器引入内部存储器。信任链可以依靠正由从RoT和引导加载器开始的先前阶段核实的每一个已加载阶段的代码。第二阶段的加载器可以核实下一个阶段，所述下一个阶段则可以包括可信环境核心以及OS加载器。一旦OS被初始化且正在运行，则剩余的完整性检查可以如这里所述由标准的OS加载器执行。可执行映像可以在没有缓存的情况下被加载到RAM。这里公开的这些概念可以扩展成包含更多具有附加修改且可执行映像大于可用ROM的受限实施方式，例如使用缓冲存储器以及直接从ROM执行代码。

[0112] 将代码从外部引入内部存储器的加载器可以包括执行密码加密的完整性检查的能力。所述完整性检查回过来可以引用安全地保持在可信环境中的密码加密功能。在正常的操作中，加载器可以将文本区段410之类的组合文本区段以及数据区段412之类的数据区段中的代码和数据拷贝至链接器命令脚本和分段报头信息定义的内部存储器。与批量加载文本区段410和数据区段412不同，加载器可以识别用户定义的代码和/或数据区段。这其中的一些补充区段信息可用于完整性检查。加载器可以计算组件的完整性量度，然后将组件的TRV定位在与之关联的区段。区段报头可以提供区段的起始地址和大小。量度值可以与同一个组件的已存储TRV相比较。如果值是匹配的，则可以将代码加载到内部存储器。如果完整性量度不匹配，则不能加载代码，并且可以为组件记录故障和/或将故障报告给策略管理器。

[0113] 每一个组件的加载器核实结果可以保存在表明该组件已被检查的比特字段以及通过与否的比特指示中。在将全部代码移至内部存储器时，策略管理器可以基于已完成的

完整性检查结果来确定授予设备怎样的访问。一种实现该处理的方式是规定加载器可以访问安全的存储器位置来追踪完整性结果。过程链接表(PLT)可以用附加信息元素增强，以便追踪是否检查并核实了组件完整性，并且在每次将经过检查的组件加载到RAM的时候可以更新完整性检查结果以及所述信息。

[0114] 在具有有限存储器的嵌入式系统中，可以使用代码调换。代码调换可以包括将那些可能用于执行的功能加载到RAM中。如果使用了子组件，那么可以在RAM中未提供所述子组件的情况下使用PLT和GOT表来定位其地址。子组件可以是具有相关联的TRV的较大组件中的很小的部分。在加载子组件时对其进行动态检查的系统中，在每次加载子组件的时候都可以使用关于整个组件的完整性检查。这种需求有可能在系统上添加不必要的计算负担。

[0115] 组件可以分成子组件，并且可以计算一个中间TRV，所述中间TRV可以用于检查每一个子组件的完整性。此外，通过实施最小块尺寸，可以计算出中间散列。这个生成子组件的TRV散列的过程可被称为TRV分解(digestion)。例如，很小的子组件散列可以是基于存储页面块大小计算的。举例来说，这种将组件拆分成子组件的处理可以在对子组件执行作为安装或启动过程的一部分的完整性检查的时候进行。此外，通过增强GOT，可以包含每一个子组件的中间TRV。

[0116] 平台完整性策略引擎(PIPE)可以是整个平台的信任系统架构的一部分。例如，PIPE可以防止下列各项中的一项或多项：网络受到攻击、误用设备、在平台上以未经授权的方式传递或操纵敏感数据。PIPE可以依靠诸如引导加载器、策略管理器和虚拟化组件之类的不同操作系统功能来控制和创建安全可信赖的平台。所述PIPE可以控制不同的功能，这其中包括安全引导处理流程、关于软件组件的完整性检查量度的处理、依照策略的后续强制操作、和/或后续的软件负载控制流程。这些策略可以由外部的利益相关者定义，例如制造商或运营商，并且是可以在设备上供应的。此外，这些策略可以通过远程更新过程而在字段中被更新。

[0117] PIPE可以通过下列各项中的一项或多项来减小加载受损软件功能的风险：受控软件数据检查和加载操作、渐进式地安装更多的功能能力、或者在运行时保持组件的动态加载。依照加载操作的进度阶段，所述操作可以包括下列各项中的一项或多项：将平台断电；阻止加载一个或多个受损组件或是隔离一个或多个组件；触发针对网络中的安全网关或补救管理器之类的外部实体的警报，以便通告低级故障或是受损功能；禁止访问平台上的安全信息，例如验证密钥等等；禁止访问平台上的安全功能，例如验证算法等等；执行批量代码或数据重载/更新过程；替换受损的软件组件和/或配置数据；或者进行更详细的调查，包括以更高的细节粒度来对疑似受损的组件执行完整性检查，以便隔离组件中的故障位置。

[0118] 在一些情况中，故障有可能非常严重，以至于可信任环境由于核心TrE功能受损而不能保证平台的信任度。低级故障有可能触发诸如产生用可信根签名的默认警报消息之类的基本操作，其中可以包括完整性和回放保护以及机密性保护。换言之，一旦发生了严重的低级安全故障，则可以通过一个或多个可用通信信道来向网络发布遇险消息。

[0119] 由于所加载的功能已被构建并且变得日益复杂，设备有可能执行更复杂的操作，例如充当代表网络实体的安全可信赖代理，这样做可以促成用于诊断、报告和/或替换受损软件或配置数据的询问过程。

[0120] 依照成功核实的功能等级，可以提供针对平台上的资源的不同访问（例如由PIPE）。如果组件的完整性检查失败，那么它可能不是可信的。检测到的这种故障可以被安全标记并指示给网络（显性或隐性），并且引导流程有可能因为这个发生故障的状况而出现分支。这种完整性检查故障可被称为执行流故障，由此，经过检查的组件可能不是可信赖的，并且启动该组件有可能会导致执行恶意的、受损的、有故障的或是错误配置的代码，而这有可能导致设备以非指定和/或非预期的方式执行网络功能。由此，新组件和可用功能的加载可能受到先前加载的组件的完整性的影响。

[0121] 结果，执行环境有可能根据控制执行过程和/或每一个引导阶段以及每一个运行时处的访问权限而改变。例如，在引导过程中的每一个阶段可以基于在该时间产生的完整性量度来做出决定。后续的阶段和策略可以使用通过任何超越了执行阶段的可用安全信息运送或存储手段而从先前阶段传递的信息（状态、变量、参数、寄存器、文件等等）来确定自己的操作。例如，上层应用验证功能可以使用关于先前加载的组件的完整性的信息来确定自己的操作，其中包括选通释放那些用于与例如外部实体进行成功验证的密钥。

[0122] 例示的PIPE功能流程可以采用这里描述的方式执行。例如，在这里可以对RoT执行完整性检查，和/或其完整性可被核实。RoT可以对基线TrE执行完整性检查，和/或所述基线TrE的完整性可被核实。如果在对基线TrE进行完整性检查和/或核实中发生故障，那么PIPE可以阻止用于附着于网络的密钥的发布，触发针对网络的警报，和/或将设备断电。如果PIPE触发针对网络的警报，则可以加载能向网络发送警报的回退代码。该警报可以触发远程批量更新过程，以便替换TrE。

[0123] 如果在对基线TrE的完整性检查和/或核实中没有发生故障，那么可以加载基本的通信连接代码。这其中可以包括执行完整性检查以及加载基线操作系统模块、执行完整性检查以及加载基线管理客户端、和/或执行完整性检查以及加载通信模块。如果在检查操作系统模块、基线管理客户端和/或通信模块的完整性的同时识别出故障，那么PIPE可以阻止用于附着到网络的密钥的发布，触发针对网络的警报，通过执行远程批量过程来替换组件，执行远程组件更新过程，和/或将设备断电。如果PIPE触发远程批量更新过程，那么可以加载一个能够向网络发送警报的回退代码。该警报可以触发远程批量更新过程，以便替换基本代码。

[0124] 如果在对基本通信连接代码的完整性检查和/或核实中没有出现故障，那么可以对剩余的操作系统及管理客户端组件执行完整性检查和/或加载这些组件。这其中可以包括执行完整性检查和/或加载可重新定位和/或重新加载的功能模块。如果在完整性检查期间识别出故障，那么PIPE可以阻止释放用于附着到网络的密钥，向网络发送依照协议的故障报告，触发警报和/或请求远程更新组件过程，和/或将设备断电。所述故障报告可以指示发生故障的组件，该组件例如可以由网络远程更新。

[0125] PIPE的操作可以根据成功核实的引导链而改变。在引导过程的每一个阶段都可以基于为在该时间（或是到该时间）被执行了完整性检查的部分或全部底层平台评定的完整性以及所应用的策略来做出决定。这些策略可以根据所实现的信任等级而适配或被其他策略替换。执行环境可以根据每一个引导阶段的控制策略而改变。后续的阶段和/或策略可以使用通过超越了执行阶段的可用安全信息运送或存储手段而从先前阶段传递的信息（例如状态、变量、参数、寄存器、文件等等）。

[0126] 举例来说，如这里所述，PIPE可以在平台上规定策略，其中所述策略可以根据如图8所示的所实现的平台初启和信任状态的等级而进行适配或改变。如图8所示，依照相应的策略812，可以对TrE802执行完整性检查、以及加载和/或执行所述TrE802。在将TrE802确定成是可信实体并且执行所述TrE802之后，设备可以移动到引导序列的下一个阶段。例如，在这里可以依照策略814来对能力804执行完整性检查、以及加载和/或执行所述能力804。关于能力804的完整性检查、加载或执行可以基于与TrE802的完整性检查、加载和/或执行相关联的信息(例如信任状态)。在引导序列先前阶段中实施的策略812可以向在关于能力904的完整性检查、加载和/或执行期间实施的策略814发出通知。在将能力804确定成是可信实体并且执行了所述能力804之后，该设备可以移动到引导序列的下一个阶段。该引导序列可以通过依照相应的策略816、818和820分别检查、加载和/或执行能力806、808和810而继续进行。如这里所述，引导序列的每一个阶段可以由引导序列中的一个或多个先前阶段通知。

[0127] 这里描述的策略可以由运营商之类的可信外部实体规定。所实现平台信任状态的结果可被传递到外部实体，其中举例来说，所述外部实体有合法的兴趣/权利来了解平台的信任状态，诸如敏感应用或服务的运营商或提供者。应该指出的是，对于可以在启动或平台工作循环中的不同阶段评定的有关平台可能的不同状态的信息来说，这些信息可被传递到一个以上的外部实体。

[0128] 多层完整性检查以及将这种完整性检查绑定到平台信任状态的处理可以采用这里描述的方式执行。例如，这种完整性检查和绑定可以使用多层完整性检查，以便通过策略和强制执行来确保设备上的密钥验证设备实现的能力(例如针对网络上的服务器之类的外部核实器且执行补救任务的设备补救功能)。如果某个用于实现预期能力的预先已知的软件和/或配置文件集合通过了完整性检查，那么使用这种密钥来进行验证的安全敏感功能对于设备而言将会是可用的。举例来说，这种软件和/或配置文件可以包括低级OS功能、驱动、配置文件和/或通信堆栈代码的某个子集。

[0129] 多层完整性检查以及完整性检查绑定还可以包括用于保护验证密钥的策略，由此可以将密钥限制成只供已授权功能或过程使用。如果密钥未受保护，那么软件有可能受损害以访问该密钥。通过提供可信功能，设备可以：对密钥进行保护，以使其仅限于供有限集合的能力、功能或单个功能使用；保护对密钥执行操作的功能/程序；和/或限制可以调用这其中的某个特许功能的对象(例如用户、系统、脚本等等)。

[0130] 这里描述的关于多层完整性检查和绑定的实施例可以包括一组预先已知的软件，这些软件可以用于允许可能受损的设备向外部实体验证其部分能力(例如其报告故障以及与补救服务器或是用于此类服务器的AAA实体一起执行补救操作的能力)。外部实体(例如补救服务器)可以是一个逻辑实体，并且可以由设备管理服务器托管，其中对于H(e)NB来说，所述外部实体是H(e)MS。

[0131] 为了提供用于规定与特定平台上的实际实施方式无关的策略的机制，在这里可以通过规定特定平台能力所需要的功能来定义这些策略。由此，与特定能力相对应的策略也可以采用与这里公开的组件-功能映射相似的方式映射到功能。外部利益相关者可以将关于特定能力的策略规定给单个功能或若干个功能。而PIPE则可以在将能力映射到特定策略以及映射到平台初启阶段的过程中负责解释策略需求。

[0132] 如果某个能力使用了某组功能，那么可以为指定给该能力的相应策略规定这些功

能。举个例子，如果希望将执行补救的能力映射到某个策略，那么可以将用于执行补救的功能映射到相应的策略，例如，修补能力可以被功能1、2和3覆盖，从而规定将相应的策略映射到功能1、2和3。

[0133] 对于一些实施方式来说，在分层的平台初启与能力和功能之间存在着合理的相关等级。这些能力可以是随着功能一层一层地渐进式初启的。例如，在平台初启的较早阶段即可初启补救功能和能力。

[0134] 多层完整性检查可不按顺序确定不同功能的已检查-未检查状态。这些功能可以是依照顺序方式检查的。然而，由于不同的功能可被映射到不同的组件(随后可以在不同的阶段加载这些组件)，因此，依照功能的完整性确定处理有可能会以非顺序方式执行，或者以一种可不与依照组件的完整性检查的原子处理序列同步(在时间长度或时间顺序方面)的顺序方式执行。

[0135] 序列化可被用于策略实施。策略实施可以为给定的平台能力确定相应的功能是否通过了完整性检查和/或是否应该应用策略。

[0136] 多范围验证可以允许平台基于所实现的平台信任等级来向一个或多个外部实体进行验证。该验证机制(例如验证询问响应机制)可被用作一种将最终得到的设备完整性范围传达给外部实体的手段。所述外部实体可以验证并且同时获取关于设备完整性范畴/范围的指示。

[0137] 在经历多层完整性检查的同时，设备的TrE可以生成用于验证询问响应的不同参数或参数值，以便用于不同的目的。根据成功的完整性检查的范围，这种生成参数的处理可以基于或使用与外部验证实体共享的相同的秘密证书。所使用的可以是常见的询问-响应计算算法。然而，根据完整性检查的范围和/或验证目的，至这种算法的输入可能是不同的。举个例子，如果TrE成功检查了整个代码库，那么可以使用诸如“已成功检查整个代码库(entire code basis successfully checked)”之类的文本串作为验证询问响应计算算法的输入。如果TrE成功检查了用于实现遇险补救功能的组件，但是未必检查了整个代码库，那么可以使用诸如“已成功检查用于遇险补救的代码库(code base for distress remediation successfully checked)”之类的另一个字串。

[0138] 一旦向设备发送了其验证询问请求，则外部验证器还可以计算两个版本的“预期”询问响应。由于其可能不知道在计算设备上计算询问响应的过程中使用的输入，因此，它有可能需要同时计算所有这两个版本的响应。通过将接收自设备的响应与预期响应集合相比较，外部验证器可以隐性地成功测试出设备完整性“范围”。通过归纳，实体可以验证和/或核实在设备端成功检查了设备完整性的某个“范围”。在以上的示例中可以使用相同或相似的实施方式来验证设备的另一个特定(例如部分)能力，这一点取决于对功能进而是用于实现该功能的组件所进行的完整性检查。

[0139] 举个例子，可被核实信任度的外部指示可以是在安全引导处理期间将代码的完整性量度绑定到特定的引导循环执行和配置签名的受保护的签名证书版本。在这里可以包括安全的时间戳、受保护的依照引导循环递增的单调计数器、或引导循环秘密(例如现时(nonce)之类的在每个引导循环中生成或引入一次的隐藏随机值)。当满足前提条件时，验证密钥将会变得可用，并且当前的可信赖处理将会设置逐次引导的受保护一次性可编程“通过”标记，其中该标记可被限制成在下一次复位之前恢复至发生故障的状况(例如检测

到初始量度之后的故障的时候)。在发布给外部实体之前,诸如TrE之类的永久性可信任环境(例如进入运行时而不仅仅是引导时)可以使用验证协议现时来对状态信息进行签名。所述报告的区段可以由通过永久性核实的可信赖处理使用临时引导密钥来签名,由此,以后的处理可以将该状态呈现给外部实体,以便进行检验。

[0140] 在这里可以使用在验证协议中交换的一个或多个经过修改的随机现时,并且这种经过修改的现时可被用作询问响应以及所预期的询问响应计算的输入。换言之,在设备本身,设备的TrE可以尝试以常规方式使用在响应计算输入(例如在通过了完整性检查的情况下)中从验证服务器接收的一个或多个随机现时来计算其验证响应。如果关于整个设备的完整性检查失败,但是设备部分功能的完整性检查成功,例如"遇险/补救功能"的完整性检查成功,那么TrE可以计算该响应的另一个值,此时使用的将会是所接收的现时的修改版本。验证服务器可以知道在哪里/如何执行这种修改。关于这种修改的示例可以包括:改变初始现时中的已知位置的一个或多个比特。这样一来,除了计算初始的"设备验证"响应之外,设备和验证服务器都可以使用经过修改的现时输入来计算响应。在接收机上(例如验证服务器),服务器可以检查它从设备接收的响应是否与"设备验证"响应相匹配。如果它们不匹配,则与声明验证彻底失败不同,验证服务器可以将接收到的响应与使用经过修改的现时计算的另一个响应值相比较。如果它们匹配,那么验证服务器可以确定虽然设备整体可能没有通过验证,但是某些功能是可以通过验证的,例如本示例中的"遇险/补救功能"。

[0141] 在安全引导处理期间,通过对照相应的TRV来比较每一个软件组件,设备可以获悉组件的完整性故障,这样做有助于确保组件的比特精度以及源的真实性。

[0142] 检测到的组件检查故障可以在完整性检查状态信息元素中获取。出于安全性目的以及出于对报告和诊断效率的考虑,该状态数据的结构可以采用多种形式。该信息可以被支持策略管理过程的当前和后续引导阶段以及运行时进程读取。组件-功能映射可以用于确定组件与网络功能以及设备的其他功能的依存关系。功能可以是网络运营商之类的连接实体或是移动支付之类的设备支持的增值应用所依靠的功能。运营商和/或应用服务提供者可以定义哪些功能可被用于针对特定任务的可信设备操作,例如承载业务量、补救、管理操作或增值应用特征,并且可以在设备上设置用于表明哪些功能可用于特定网络操作的策略。

[0143] 一组网络功能可以来自网络运营商和/或应用服务供应商所预期的标准能力集合。这些功能可以包括无线电技术、协议堆栈中的层、网络协议、管理能力、安全和信任机制、增值应用特征等等。

[0144] 设备的策略管理器可以使用显示了依照设备组件的网络功能依存关系的嵌入式组件-功能表或映射(如这里所述)。在图9中显示了在完整性检查和/或报告期间使用这种组件-功能表或映射的处理。如图9所示,在910,设备可以通过使用相应的TRV904检查组件902的完整性量度来执行完整性检查。在执行了完整性检查之后,在916,设备策略管理实体908可以接收到一个或多个已经经过了完整性检查的组件的组件标识符(例如组件地址),并且会在918接收到关于每一个组件是否通过完整性检查的指示。该策略管理实体908可以确定与组件相关联的相应功能,以便发送至网络实体。举例来说,发生故障且需要补救的功能可被发送,或者可以发送关于已被检验的功能的指示。例如,策略管理实体908可以使用处于920的组件地址来检索处于922且与所给出的组件相对应的组件标识符。如图9所示,组

件标识符可以是从表906中检索的，其中所述表包含了组件地址到组件标识符的映射。策略管理实体908可以使用处于924的组件标识符来检索处于926且与所给出的组件相对应的组件功能。这些组件功能可以是从表912中检索的，其中所述表包含了组件标识符-组件功能的映射。在932，策略管理实体908可以使用所确定的功能（例如通过或未能通过完整性检查）来与网络实体进行交互。例如，策略管理实体可以执行完整性报告，接收设备策略，或者接收与所确定的一个或多个功能相对应的操作和/或限制。每一功能可与一个或多个策略930相关联。所收集的一个或多个功能可以一起启用设备内部的特定功能。这些能力可以是分层次的。例如，基础层能力可以是下一个能力层的子集，该下一个能力层则是接下来的能力层的子集等等。将能力分层的处理可以反映典型的以层或信任链为基础的安全引导序列。层可以是能够实现特定能力的功能的任意组合，或者它也可以是分层的层次与特定功能组合的混合组合。策略管理实体908可以使用处于928的组件映射功能来检索处于930且与给定能力相对应的一个或多个策略。这些策略可以是从表914中检索的，其中所述表包括能力与不同策略的映射。如932所示，这些策略可以是从网络实体接收的。

[0145] 发生故障的组件可能导致出现一个或多个发生故障的功能。软件开发和构建工具嵌入的组件-功能映射信息可以在确定功能的完整性检查故障量度的过程中为策略管理器提供支持，并且由此支持以标准化的方式报告可能与设备和制造商方面的实施方式无关的故障，例如验证功能或基带无线电协议堆栈故障。在设备上，该信息可以可被保护免于修改，这一点是由不可变的可信根（RoT）保证的。

[0146] 策略管理进程可以在引导和运行时环境中使用该信息。通过使用完整性检查处理的结果，设备上的策略管理进程可以确定哪一个网络功能发生故障。在加载器检查代码的时候，组件可以用一个在引导时间期间可用的引用（例如符号表或存储地址）来识别。在核实了完整性和源的真实性之前，被检查的代码可被隔离，以免在可执行存储器中使用。同样，从加载、检查被检查的代码时起以及在执行过程中，所述被检查的代码可以受到保护（例如通过以受保护的方式执行和存储，硬件保护，密码加密保护，虚拟化等等）。通过修改存储器映射、访问权限等等，可以阻止执行过程。作为在图9的表906和912中示出的示例，处于地址位置x的代码可以对应于组件A，所述组件可以映射到网络功能F1、F5和F14。举例来说，这个代码可以是一个原始的散列函数，并且有若干个通信协议是依赖于该函数的。完整性报告和补救有可能依赖于发生故障的基元。因此，在网络标准化列表中可以包含这些完整性报告和补救功能，以此作为底层的支持功能。

[0147] 功能故障的标准化列表可以向网络提供关于设备能力的细粒度信息。对于远端设备来说，可靠地理解设备所能实现的功能有助于应用资源来补救问题。在遇险情况下，如果有能力以安全和有保证的方式指示故障，从而使得网络可以检验遇险指示本身（例如源、时间和精度）的完整性，那么可以防止在虚假警报的情况下不必要地使用昂贵的网络资源。

[0148] 图10提供的是包含了例示系统组件的报告和补救系统的例示综述。如图10所示，在1002，在设备1004的组件上检测到故障。该故障可被指示给网络。例如，在1008，所述故障可以包含在报告给网络的设备报告1006中。举例来说，该故障可被指示成是与发生故障的组件相关联的发生故障的网络功能。在向网络报告故障时，网络可以做出关于访问控制的细粒度决定。例如，网络平台检验实体1010可以基于所报告的功能检查报告1006来允许禁止访问、局部访问或是完全访问。在1016，设备1004可以接收访问决定。如果在设备报告

1006中指示了故障，那么网络平台检验实体可以将功能报告传送给设备管理服务器1012，所述服务器可以从功能-组件映射信息1014中确定一个或多个发生故障的组件。该信息可以用于补救设备1004，以及重新加载设备1004中发生故障的软件组件。例如，设备管理服务器1012可以在1018询问设备1004，以便窄化与发生故障的组件相关联的发生故障的功能的范围。在将完整性故障隔离到设备1004的某个区段以便实施有效补救之后，设备管理服务器1012可以在1020请求软件更新（例如替换组件），例如从代码构建发布实体1022请求。设备管理服务器1012可以在1024向设备1004提供软件更新，以便补救发生故障的组件。这种制造商设备上的特定故障代码的抽象形式将会减小运营商紧密了解设备（与之相连）的负担，但是与此同时还可以允许基于发生故障的功能的粒状访问控制及补救决定。

[0149] 执行流程故障有可能是在处理链中调用的组件的故障。组件的执行可以保持很高的确信等级，同时扩展设备的能力。在当前已被核实的处理检查下一个处理的代码（并且有可能检查设备配置）的完整性并且发现其出现故障的时候，这时有可能检测到执行流程故障。后续引导代码中的故障有可能意味着当前可信处理可以执行下列各项中的一项或多项：将当前状态锁定成最后一个已知的良好状态；指示故障；将对于处理的控制保持在等待模式中；将控制权传递到遇险处理；和/或将控制权传递到不可信赖的处理。用于发生故障的阶段的验证和身份密钥可被锁定来避免进一步处理，以使不可信赖的处理无法通过报告（例如带签名的状态）或是验证技术（例如自主检验）来向网络指示有效状态。

[0150] 出于对效率的考虑，从设备发送到网络的故障报告可以包括用于提示来自运营商网络检验功能的操作以执行细粒度网关访问控制以及向网络设备管理实体指示设备故障的最低限度的信息量。管理实体可以通过查看制造商专用信息来确定故障的根本原因，以及确定哪些代码可被补救。该管理实体可以向网络检验功能反向提供进一步的信息，以便允许实体做出关于访问控制和/或补救的进一步策略决定。

[0151] 在下表1中描述了可以包含在故障报告中的发生故障的功能的例示集合。

[0152]

发生故障的功能列表（代码，数据，参数）
LTE
基带域

[0153] 表1：用于发生故障的功能的例示完整性违例（由TrE签名的故障报告）

[0154] 在这里可以发送关于通过或未能通过这样的结果的功能报告。表2提供了关于这种列表的一个示例。

[0155]

功能（代码，数据，参数）	状态（未能通过或通过）
总的结果	
LTE	
UMTS	

[0156]

核心网关命令	
管理命令	
WiFi	
UICC/智能卡信道	
应用域	
基带域	

[0157] 表2:用于功能的例示完整性违例(由TrE签名的报告)

[0158] 在验证期间,所述功能报告可以备选信道上的净荷中传送,其中举例来说,所述信道可以是用于网关验证的信道,由此,即使网关验证失败,所述报告也可以作为净荷而在验证序列中被发送。该净荷可以由设备的可信执行环境签名。

[0159] 可信执行环境(TrE)可以具有自己的功能字段(例如为了方便起见或出于对冗余度的考虑)。然而,如果功能或遇险净荷列表是由TrE签名的,或者验证密钥是受TrE保护的,那么在与网络取得联系的时候,TrE的完整性是已知的。TrE的一个或多个签名密钥可以用故障防护封闭方法来保护。换言之,如果TrE受损,那么它的一个或多个密钥是无法得到的(例如既无法被设备或是其内部功能和接口得到,也不能被I/O和测试端口得到,并且不能被外部的网络实体得到)。

[0160] 如果发生故障或警报,那么服务网络实体可以核实遇险设备具有可靠报告故障的特征,并且遇险设备的机制并未发生故障。有两种保证形式可以向网络指示报告机制可靠。一种机制可以采用静态的可信第三方担保的形式,例如特定类型的设备与某一组可信赖的报告能力相符的证书。这种担保可以提供关于处理和发送故障状况的警报和消息的设备能力以及牢固等级(或可信赖度)的信息。网络运营商可以使用经过证实的能力来建立在发生故障警报的情况下恰当自动地做出响应的过程。

[0161] 另一种担保形式可以是通过与设备进行的在线事务。在发生故障时间的时候,设备内部的机制可以提供一个关于报告机制完整性的指示。这种内部能力有可能涉及静态的担保形式,这是因为它可以允许第三方提供符合证书。

[0162] 在这里,关于设备遇险和补救的不同协议流程可以在H(e)NB的上下文中描述。然而,这里描述的概念并不局限于这种实施例,而是可以应用于任何通信设备。根据图16描述的例示实施例,设备完整性信息可被从H(e)NB通过SeGW发送到管理服务器(H(e)MS)。根据图17描述的另一个实施例,H(e)NB可以直接连接到管理服务器(H(e)MS)。根据图18A和18B描述的另一个例示实施例,设备完整性检查可被加强,以便允许在出现局部故障的情况下执行细粒度的访问控制。这些过程可以允许在DRF的控制下执行代码或数据块补救,其中所述DRF可以使用从网络中的补救实体接收的补救数据来改变代码或数据块。如果系统架构允许软件以及软件或硬件配置数据的变化,那么这其中可以包括这种变化。所示出的流程可以是针对软件补救的,但是并不局限于此。

[0163] 在网络管理实体与设备自身之间可以执行询问,在此期间,关于设备的发生故障的完整性的详细信息可以以比初始报告更精细的粒度提取。该信息可以导致发现故障原因,指示组件的哪个部分发生故障,以及提供粒度更细的代码修复和/或配置,以通过减小

软件下载大小并且由此平衡网络带宽需求以及代码下载大小，实现有效设备管理。根据一个例示实施例，使用IP网际协议的受管理网络设备可以将TR-069(CPE WAN)用于设备管理目的。该协议可以提供针对“自动配置服务器”(ACS)的访问。ACS应用可以在表3所示的协议栈中使用若干种能力。

[0164]

ACS
RPC方法
SOAP
HTTP
SSL/TLS

[0165]

TCP/IP

[0166] 表3TR-069协议堆栈

[0167] 该堆栈可以被RoT处理访问，所述RoT处理可以向连接TR-069的管理服务器提供高保证的遇险指示。但是，在遇险状况中可不使用每一个层的整个特征列表，由此可以修整堆栈来执行这些用于安全地将设备反向自举至完整的管理和设备功能的过程。

[0168] 图11示出的是通过在设备上收集信息来确定可以执行的操作的例示呼叫流程图。当发生完整性故障时，这时可以在H(e)NB1102之类的设备(例如TR-069设备)与H(e)MS1104之类的网络实体之间执行图11所示的序列。在1106，在H(e)NB1102与H(e)MS1104之间可以建立连接(例如TR-069连接)。例如，在1106，H(e)NB1102可以使用其TrE或RoT来建立连接。在1108，H(e)NB1102可以将一个或多个完整性故障报告给H(e)MS1104，例如通过使用信息请求和/或警报来报告。在1108，H(e)MS1104可以接收警报，并且在1110，它可以使用信息响应消息(例如TR-069通知响应)来对H(e)NB1102做出响应。

[0169] 粗略地说，设备的每一个软件组件都可以具有相应的TRV。如这里所述，当组件无法匹配基准的时候，该组件有可能具有完整性故障。一些组件有可能很大。例如，操作系统有可能是作为具有单个TRV的单个单体码块递送的。

[0170] 图12示出的是通过询问来隔离组件中的一个或多个故障的一个或多个位置的例示呼叫流程图。在询问过程期间，诸如H(e)MS之类的网络管理实体1204可以与诸如H(e)NB之类的设备1202交互，以便通过收集附加信息来确定可被执行以补救检测到的故障的操作。根据一个例示实施例，设备1202可以使用TrE或RoT来与网络管理实体1204交互。

[0171] 网络管理实体1204可以决定执行整个代码映像的单体下载。该下载可以包括重新载入当前代码映像以及经过更新的代码映像。由于可以将整个映像下载至网络管理实体1204，因此可以在已下载的映像中包含TRV(初始或更新)。网络管理实体1204可以决定下载被报告成发生故障的组件，而不是单体下载整个代码映像。由于可以将包括更新时的TRV在内的整个组件下载到设备1202，因此，所述映像可以是当前组件或更新组件的重新加载。对于更新的组件来说，客户端管理实体可以管理现有代码映像，以便确保整个映像的完整性和结构保持不变。

[0172] 如图12所示，在1206可以检测到完整性故障。举个例子，在设备1202的组件中有可能检测到完整性故障，并且可以像这里描述的那样向网络管理实体1204发送一个完整性检

查报告。在1208，在设备1202与网络管理实体1204之间可以建立一个连接(例如TR-069连接)。在1210，网络管理实体1204可以向设备1202发送一个关于参数的请求，其中所述参数涉及的是与设备1202上的完整性故障相关联的警报。在1212，设备1202可以向网络管理实体1204发送一个包含警报细节的参数响应。在1214，网络管理实体1204可以决定窄化设备上的组件的完整性故障，以便进行补救。为了窄化完整性故障，在1216，网络管理实体1204可以向设备1202发送一个参数请求，以便获取更细粒度的量度。在1218，设备1202可以对设备1202上的组件进行量度。在1220，网络管理实体1204可以对基准进行量度。例如，由于组件制造者可能没有提供组件量度并且设备1202可以使用动态粒度来产生组件量度，因此，网络管理实体1204可以即时生成基准值。举例来说，网络实体1204产生的基准量度可以与设备1202产生组件量度相比较。设备1202可以在1222发送一个向网络管理实体1204指示设备1202已经或者将要采用更细粒度的完整性量度的参数响应。

[0173] 在1224，网络管理实体可以发送一个关于一个或多个第一节点(例如一个或多个组件)的完整性量度的参数请求。在1226，设备1202可以向网络管理实体1204发送一个指示了一个或多个第一节点的完整性量度的参数响应。网络管理实体1204可以在1228发送一个关于来自设备1202的接下来的一个或多个节点(例如一个或多个组件)的量度的参数请求。举例来说，关于接下来的一个或多个节点的量度可以是基于从设备1202接收的一个或多个节点的量度请求的。此外，所述接下来的一个或多个节点可以是一个或多个第一节点的部分或子组件。在1230，设备1202可以向网络管理实体1204发送一个包含了所请求的接下来的一个或多个节点的量度的参数响应。举例来说，如1232所示，来自网络管理实体1204的参数请求和来自设备1202的参数响应可以重复进行，直至完整性故障与设备1202的区段(例如组件、函数或是其部分)隔离，以便实施有效的补救。在识别并隔离了完整性故障之后，在1234，设备1202与网络管理实体1204之间的连接(例如TR-069连接)可以关闭。

[0174] 即使在检测到故障并且产生了单体块或组件的多个量度之后，网络管理实体1204也可以决定进行询问。设备1202可以执行完整性量度和/或将他们安排在二叉树之类的用于快速引用的数据结构中。网络管理实体1204可以执行相同的处理。这种结构可以揭示单个块中的多个故障，由此网络管理实体1204可以快速确定哪些单体块或组件未能通过完整性检查，这样做可以窄化影响范围，并且可以潜在地减小软件下载业务量。如图12和13所示，经过修改的块的细节可被检查，以便以比用于组件或单体块的TRV更精细的粒度来隔离其被损坏的位置。

[0175] 借助于窄化组件故障范围的细粒度，信息网络管理实体可以为代码分段而不是整个码块创建下载映像。举例来说，如图13所示，在支持若干种功能的大型单体组件1302中有可能发生故障。由于整个组件1302有可能无法通过完整性检查，因此，总的量度值1304可以与整个组件1302相关联。在1306，通过设备与网络管理实体之间的询问，可以产生精细粒度的量度。所述询问有可能导致产生更精细粒度的完整性量度，以便定位大型单体组件1302内部的分段的故障，例如分段1308。例如，在询问之后，从完整性检查中得出的总的量度值1310可以包括分段1308的量度值7，而不是与整个大型单体组件1302相关联的量度值1304。在识别了分段1308处的故障之后，这时可以将一个修复补丁定位到这个分段。该修复补丁可以包括使用大型单体组件1302内部通过完整性检查的其他分段来修复分段1308上的故障的指令。

[0176] 所生成的完整性量度的数量可以基于设备类型来限制。当子组件下载是当前组件的重新加载时，设备不会使用附加信息来执行重新加载。但是，如果子组件下载是对已有组件的更新时，那么用于所述子组件的更新TRV可以包含在该下载中。设备客户端和网络管理实体可以同步，以便产生相同的更新组件映像，以使更新的TRV与更新的组件映像的散列相匹配。设备管理客户端可以管理已有代码映像，以便确保整个映像的完整性不变。

[0177] 询问处理有可能会修改网络所具有的所报告的发生故障的功能列表的版本，而这可以被反馈给网络检验、验证和/或网关实体，以便修改设备上的网络访问控制。

[0178] 该询问处理可以使用迭代量度方法来隔离组件故障，由此网络管理实体和设备可以渐进式地生成量度，从而窄化指示故障的这些子区段上的映像的视野。该方法可以减小用于给定分辨率的量度值的数量，由此允许存储器受限设备的量度与分辨率之间的折衷。举例来说，在图14中示出了这种询问处理的一个例示实施例。如图14所示，在组件1402中有可能发现故障，并且总的量度值1404可以与整个组件1402相关联。在发现组件1402中的故障之后，设备和网络可以执行询问，并且基于所述询问，设备可以在组件1402上执行第一量度迭代。例如，在第一量度迭代中，设备可以在组件1402的子区段1406和子区段1408上执行独立的量度。由此，设备可以确定子区段1406通过了完整性检查，而子区段1408则包含完整性故障原因。这一点可以用关联于子区段1406的量度值1以及关联于子区段1408的量度值2来指示。由于子区段1408仍未通过完整性检查，因此，总的量度值1404仍旧可以指示组件1402包含完整性故障。

[0179] 虽然将完整性故障窄化到了子区段1408，但是设备和网络可以通过执行附加询问过程来渐进式地生成用于补救的粒度逐渐精细的量度。作为询问结果，设备可以在组件1402上执行第二次量度迭代。如图14所示，第二次量度迭代可以在子区段1408上执行，这是因为这个子区段是被确定成完整性故障原因的子区段。设备可以在第二次迭代中对子区段1410和1412执行独立量度，这样做分别可以产生量度值3和量度值4。子区段1410和1412可以是子区段1408的子区段。量度值3可以指示子区段1408的子区段1410通过了完整性检查。然而，量度值4可以指示子区段1412未能通过完整性检查，并且包含了组件1402的完整检查故障的原因。

[0180] 虽然将完整性故障窄化到了子区段1412，但是设备和网络可以通过执行附加询问过程来渐进式地生成用于补救的粒度逐渐精细的量度。例如，设备可以执行第三次量度迭代。第三次量度迭代可以在子区段1412上执行，这是因为子区段1412是在第二次量度迭代之后被确定成完整性故障原因的子区段。如图14所示，设备可以在子区段1414和1416上执行独立量度。该设备可以确定子区段1414包含了完整性故障的原因。其中举例来说，这一点可以用量度值5指示。在第三次量度迭代之后，由于与包含完整性故障原因的组件1402的细粒度子区段相对应，与子区段1414相关联的功能可被发送至网络。作为询问结果，网络可以确定能被定位至子区段1414的修复补丁(例如替换或修复子区段)，并且将所述修复补丁发送至设备，以便替换或修复子区段1414。虽然图14示出了由于网络与设备之间的询问结果而在设备上执行的三次量度迭代，但是任意数量的量度迭代都可以执行，以便隔离包含完整性故障的组件部分。在一个例示实施例中，迭代次数是以服务供应商为基础的。在另一个例示实施例中，设备管理服务器保持了一个与不同标准相对应的故障数据库，其中所述标准包括设备标识和软件版本号码。举例来说，故障信息可以允许设备管理服务器评定询问

策略。

[0181] 在对比树的完全生成之类的其他算法来确定这种渐进式方法的值的过程中，关于网络通信成本、量度计算以及用于量度计算的值的负载之间的比较是可以加权的。

[0182] 在渐进式方法中，量度可以是在窄化故障字段的时候获取的。代码/数据可以被重复量度。通过使用混合方法，可以减少有可能被再次量度的代码/数据总量，其中举例来说，所述代码/数据初始可以将数据拆分成某个最优数量的子区段。通过用某个因数对映像进行划分，可以以渐进地方式进一步解析发生故障的一个或多个区段。该因数可以是依照迭代确定的，并且是以其他时间延迟以及性能考虑因素为基础的，由此通过在网络上使用最小带宽和最小功耗，优化故障隔离以及确定的速率。为了加快故障隔离处理，网络可以在用于指定组件故障的散列树之类的数据结构中预先生成一组预期散列，以及向设备发送这个子TRV量度树。这样做可以允许设备在这种渐进方法中进行比较，直至达到树的最高解析度。网络可以确定是否需要更大的解析度来优化补救过程(例如基于需要被校正的设备的数量)。如果需要更大的解析度，那么网络可以生成和发送始于更精确的位置的新树，或者可以交互执行所述比较。

[0183] 对于软件或数据更新来说，软件管理服务器可以对文件中的简单的二进制差异执行代码/数据更新，以便执行更新。设备管理客户端可以接收二进制更新，并且通过修改完整的代码映像来确保其完整性。相同的二进制差异原理可以扩展至包含用于软件管理实体的询问技术，从而减少软件更新期间的业务量。换言之，软件更新过程可以包括这里描述的很多相同的补救技术，其中包括二叉树搜索。

[0184] 图15示出的是涉及遇险警报以及单体代码替换的例示呼叫流程图。虽然该呼叫流程是用H(e)NB描述的，但是其他网络设备也是可以使用的。该呼叫流程序列可以包括这里描述的一个或多个步骤。举例来说，如图15所示，在1512，H(e)NB1502可以执行一个安全引导序列。在完整性检查阶段1514，H(e)NB1502有可能在操作代码中发现一个完整性故障，由此，H(e)NB1502的TrE不会提供用于SeGW1504的验证的私钥。然而，TrE有可能认定H(e)NB1502通过了H(e)MS1506的操作。例如，H(e)NB1502能够向H(e)MS1506发送一个警报，和/或可以将用于H(e)MS1506的验证的私钥提供给H(e)NB1502能够实施的过程。在1516，H(e)MS1506可以与H(e)NB1502建立IP连接(例如SSL/TLS启动)。在1518，H(e)NB1502可以向H(e)MS1506警告完整性故障。例如，H(e)MS1506可以接收一个来自H(e)NB1502的遇险警报，并且在1520向软件补救实体1510发送一个代码补救请求(例如用于重新加载或更新)。在1522，软件补救实体1510可以执行软件构建处理。在1524，软件补救实体1510可以向H(e)MS1506提供一个单体组件替换。例如，软件补救实体1510可以向H(e)MS1506提供一个单体补救映像。H(e)MS1506可以与H(e)NB1502建立安全连接。例如，在1526，H(e)MS1506可以与H(e)NB1502建立安全的IP连接(例如借助SSL/TLS启动)。该H(e)NB1502和H(e)MS1506可以通过执行验证来允许用于修复H(e)NB1502的软件/固件下载。在1528，H(e)MS1506可以更新H(e)NB1502的软件/固件，并且可以重新引导H(e)NB1502。在1530，H(e)NB1502可以重新引导和/或重新启动完整性检查。

[0185] 图16示出的是涉及远程软件遇险/补救的例示呼叫流程，其中H(e)NB1602向H(e)MS1606告知遇险。该呼叫流程序列可以包括这里描述的一个或多个步骤。此外，虽然该呼叫流程是用H(e)NB描述的，但是其他网络设备也是可以使用的。如图16所示，在1612，H(e)

NB1602可以执行一个安全引导序列。在1614的完整性检查阶段期间，H(e)NB1602的TrE有可能在用于正常操作的代码中发现了一个故障，由此，H(e)NB1602的TrE不会提供用于SeGW1604的验证的私钥。然而，TrE有可能认定H(e)NB1602能够安全地连接到H(e)MS1606，因此，TrE可以将用于H(e)MS1606的验证的私钥提供给H(e)NB1602能够实施的过程。H(e)NB1602不能(或者在一些故障中不能)尝试与SeGW1604进行验证。在1616，H(e)NB1602可以与H(e)MS1606建立IP连接(例如安全的SSL/TLS会话)。在1618，H(e)NB1602可以向H(e)MS1606警告H(e)NB1602的完整性检查状态(例如完整性故障)。例如，H(e)MS1606可以通过验证或是通过对带有TrE签名的完整性状态净荷信息实施的签名核实来核实H(e)NB1602的完整性信息的真实性。如果完整性检查状态表明在H(e)NB1602上存在完整性故障，那么H(e)MS1606可以在1620确定用于所指示的故障的软件修复。该软件修复可以包括更进一步地询问完整性故障。例如，在1622，H(e)MS1606可以诊断和/或询问H(e)NB1602。该H(e)MS1606可以详细确定故障原因(例如位置)，并且在1624，H(e)MS1606可以向软件补救实体1610发送一个用于重新加载或更新的代码补救请求。在1626，软件补救实体1610可以执行软件构建处理，并且向H(e)MS1606发送软件修复信息。例如，所述软件修复信息可以包括补救映像。在1630，H(e)MS1606可以与H(e)NB1602建立安全连接(例如经由IP连接)。H(e)NB1602和H(e)MS1606可以通过验证来允许用于修复H(e)NB1602的软件/固件下载。在1632，H(e)MS1606可以下载H(e)NB1602的软件/固件，和/或重新启动H(e)NB1602。在1634，H(e)NB1602可以执行安全的引导序列和/或重新启动完整性检查处理。在1636的完整性检查阶段期间，H(e)NB1602的TrE有可能发现完整性检查通过了用于SeGW1604的操作，并且可以提供用于SeGW1604的验证的私钥。在1638，H(e)NB1602和SeGW1604可以相互验证，并且在1640，SeGW1604可以向H(e)NB1602指示验证成功。

[0186] 图17示出的是涉及结合SeGW验证尝试的远程软件遇险/补救的例示呼叫流程图。该呼叫流程序列可以包括这里描述的一个或多个步骤。此外，虽然该呼叫流程是用H(e)NB描述的，但是其他网络设备也是可以使用的。如图17所示，在1712，H(e)NB1702可以执行一个安全引导序列。在1714的完整性检查阶段期间，H(e)NB1702有可能在用于SeGW1704的操作的代码中发现一个完整性故障，由此，H(e)NB1702的TrE不会提供用于SeGW1704的验证的私钥。然而，TrE有可能认定H(e)NB1702能够安全连接到H(e)MS1706，并且可以将用于H(e)MS1706的验证的一个或多个私钥提供给H(e)NB1702能够实施的过程。例如，在1716，H(e)NB1702可以通过使用IKEv2协议来尝试向SeGW1704进行验证，但是有可能会失败(例如因为私有验证密钥不可用)。在1718，H(e)NB1702可被提供一个表明与SeGW1704的验证失败的指示。在1720，H(e)NB1702可以与H(e)MS1706建立安全的IP连接(例如经由SSL/TLS启动)。例如，所述安全的IP连接可以基于H(e)NB1702与SeGW1704之间的失败的验证尝试来建立。在1722，H(e)NB1702可以向H(e)MS1706警告H(e)NB1702的完整性检查状态(例如完整性成功或失败)。H(e)MS1706可以通过验证或者通过对带有TrE签名的完整性状态净荷信息执行签名核实来核实H(e)NB1702的完整性信息的真实性。如果完整性检查状态指示在H(e)NB1702上存在组件的完整性故障，那么在1724，H(e)MS1706可以确定用于所指示的故障的软件修复。该软件修复可以包括更进一步地询问完整性故障。例如，在1726，H(e)MS1706可以诊断和/或询问H(e)NB1702。该H(e)MS1706可以详细确定故障原因，并且在1728向软件补救实体1710发送一个代码补救请求(例如用于重新加载或更新)。软件补救实体1710可以在1730执

行软件构建处理，并且在1732向H(e)MS1706提供软件修复(例如补救映像)。在1734，H(e)MS1706可以与H(e)NB1702建立安全的IP连接(例如SSL/TLS启动)。H(e)NB1702和H(e)MS1706可以通过验证来允许用于修复H(e)NB1702的软件/固件下载。在1736，H(e)MS1706可以下载H(e)NB1702的软件/固件，并且重新启动H(e)NB1702。在1738，H(e)NB1702可以执行一个安全引导序列和/或重新启动完整性检查。在1740的完整性检查阶段期间，H(e)NB1702可以发现通过了完整性检查，并且提供用于SeGW1704的验证的私钥。在1742，H(e)NB1702和SeGW1704可以相互验证，并且在1744，SeGW1704可以指示验证成功。

[0187] 图18A示出的是涉及远程软件遇险/补救的例示呼叫流程图，其中网络可以禁止通过SeGW进行验证。该呼叫流程序列可以包括这里描述的一个或多个步骤。此外，虽然该呼叫流程是用H(e)NB描述的，但是其他网络设备也是可以使用的。如图18A所示，在1812，H(e)NB1802可以执行一个安全的引导序列。在1814的完整性检查阶段期间，H(e)NB1802可能发现操作代码中的故障，因此，H(e)NB1802的TrE不会提供用于SeGW1804的验证的私钥。然而，TrE有可能认定H(e)NB1802能够安全连接到H(e)MS1806，由此将用于H(e)MS1806的验证的一个或多个密钥提供给H(e)NB1802能够实施的过程。在1816，H(e)NB1802可以尝试向SeGW1804进行验证(例如使用IKEv2协议)，但是有可能失败(例如因为私钥不可用)。H(e)NB1802可以在验证尝试期间提供关于H(e)NB1802的完整性且带有TrE签名的信息。在1818，SeGW1804可以将完整性信息(例如故障报告)转发给网络检验实体1808。网络检验实体1808可以基于其接收的信息来确定禁止一些通过SeGW1804的H(e)NB1802的业务量，并且可以在1820发送一个表明将一些或所有H(e)NB1802的业务量限制到于H(e)MS1806的细粒度访问决定。在1822a，SeGW1804可以发送一个关于验证失败的指示。在1824，出于补救目的，网络检验实体1808可以向H(e)MS1806发送H(e)NB1802的完整性信息。在1826，H(e)MS1806可以与H(e)NB1802建立安全IP连接(例如经由SSL/TLS启动)。在1828，举例来说，H(e)MS1806可以基于验证失败而确定执行软件修复。在1830，H(e)NB1802和H(e)MS1806可以通过验证来允许H(e)MS1806实施的诊断和询问。在1832，H(e)MS1806可以详细确定故障原因，并且向软件补救实体1810发送一个代码补救请求(例如用于重新加载或更新)。在1834，软件补救实体810可以执行软件构建处理，以便为H(e)NB1802构建替换软件组件(或是其一部分)。在1836，软件补救实体1810可以向H(e)MS1806提供软件修复信息(例如补救映像)。在1838，H(e)MS1806可以与H(e)NB1802建立安全的IP连接(例如经由SSL/TLS启动)。H(e)NB1802和H(e)MS1806可以通过验证来允许用于修复H(e)NB1802的软件/固件下载。在1840，H(e)MS1806可以下载H(e)NB1802的软件/固件。在1842，H(e)NB1802可以重新引导和/或加载并运行所下载的代码。

[0188] 图18B示出的是涉及与即时受限访问及精细访问控制相结合的远程软件遇险/补救的例示呼叫流程图。虽然该呼叫流程是用H(e)NB描述的，但是其他网络设备也是可以使用的。图18B的呼叫流程序列可以包括许多步骤，这些步骤与图18A示出的呼叫流程是相同或相似的。然而如图18B所示，H(e)NB1802与SeGW1804的验证可以成功，但是SeGW1804有可能限制访问H(e)NB1802。例如，在1814的完整性检查阶段期间，H(e)NB1802可能发现用于操作的代码中的故障，因此，H(e)NB1802的TrE不会提供用于SeGW1804的验证的私钥。然而，TrE有可能认定H(e)NB1802能够安全连接到H(e)MS1806，由此将用于H(e)MS1806的验证的一个或多个私钥提供给H(e)NB1802能够实施的过程。在1816，H(e)NB1802可以尝试向

SeGW1804进行验证(例如使用IKEv2协议)。H(e)NB1802可以在验证尝试期间提供关于H(e)NB1802的完整性且带有TrE签名的信息。在1818,SeGW1804可以将完整性信息转发给网络检验实体1808。网络检验实体1808可以基于接收到的信息来决定验证H(e)NB1802的访问,并且可以将此发送给SeGW1804。SeGW1804可以以来自网络检验实体1808的信息为基础验证H(e)NB1802,但是有可能限制H(e)NB1802的访问。在1822b,SeGW1804可以向H(e)NB1802发送一个表明验证成功但是SeGW1804访问受限的指示。在图18B的协议流程中示出的剩余步骤与图18A的协议中示出的步骤可以是相同或相似的。

[0189] 图19示出的是涉及与SeGW访问相结合的H(e)NB软件组件补救处理的例示呼叫流程图。该呼叫流程序列可以包括这里描述的一个或多个步骤。此外,虽然该呼叫流程是用H(e)NB描述的,但是其他网络设备也是可以使用的。如图19所示,在1912,H(e)NB1902可以执行一个安全引导序列。在1914的完整性检查阶段期间,H(e)NB1902未能发现设备代码中的故障,但是有可能在映射中发现了故障、遗漏组件或配置缺失。H(e)NB1902的TrE可以提供用于SeGW1904的验证的私钥。例如,在1916,H(e)NB1902可以向SeGW1904发送一个包含该私钥的状态报告(例如使用IKE)。在1918,SeGW1904可以将完整性信息转发给网络检验实体1908。网络检验实体1908可以基于接收到的信息来决定验证H(e)NB1902的访问,并且可以将此发送给SeGW1904。举例来说,在1920,网络检验实体1908可以根据运营商策略而向SeGW1904发送一个细粒度的访问决定。基于网络检验信息,SeGW1904可以验证H(e)NB1902,和/或根据运营商策略来设置访问许可。在1922,SeGW1904可以向H(e)NB1902发送一个关于验证成功的指示。在1924,网络检验实体1908可以向H(e)MS1906发送H(e)NB完整性信息。例如,所述完整性信息可以是出于重新配置的目的而被发送的。在1926,H(e)MS1906可以确定可用于补救故障的补救信息(例如软件/参数),并且可以在1928向软件补救实体1910发送一个补救请求(例如用于重新加载或更新)。在1930,软件补救实体1910可以执行软件构建处理,以便为H(e)NB1802构建替换软件组件(例如其一部分)。在1932,软件补救实体1910可以向H(e)MS1906提供补救信息(例如补救映像和/或软件/配置更新)。在1934,H(e)MS1906可以将补救信息转发给SeGW1904。在1936,SeGW1904和H(e)NB1902可以执行验证并建立安全的IP连接(例如使用IKEv2或IPsec)。在1938,SeGW1904可以将补救信息(例如补救映像或软件/固件更新)下载到H(e)NB1902。举例来说,SeGW1904可以发送一个关于发生故障的非重要软件/参数的更新(例如IKE和/或IPsec更新)。

[0190] 设备有可能同时具有旧有和先进能力。用于旧有能力的网络服务通常是可用的。用于先进能力的网络服务则有可能会很稀少和/或没有得到全面支持。为了减轻部署问题,运营商可以平衡旧有能力,以便缓慢移动到先进能力。这种平衡可以采用这里描述的方式使用。例如,设备可以在某个位置作为旧有设备附着于网络。网络可以验证设备,并且在其订阅中认定所述设备可能具有先进能力和/或能够安全地支持先进能力。如果用于该位置的网络支持先进能力,那么网络管理实体可以为该设备接入点提供针对更新服务器的细粒度访问。设备可以通过旧有接入点来访问更新服务器。在一些实施方式中,设备订阅可以限制针对高级特征而不是旧有服务的访问,因此,接入点通常有可能限制设备对于更新服务器而不是旧有网络的访问。所述更新服务器和设备则可以相互验证。更新服务器可以(例如通过验证隐性或显性地)检验设备的先进能力,和/或为设备提供访问证书、配置信息、接入点地址和/或代码,以便允许设备作为新型设备来直接重连。设备能够产生用于网络访问验

证的共享或非对称密钥对。网络可以使用新的设备证书来更新设备订阅数据库。设备则具有将用于先进网络位置的访问证书引入受保护区域的能力。在线证书生成处理可以允许未来针对先进网络且经过预先检验的连接。网络可以向先进网络的接入点(这些支持先进能力)告知具有某些证书的设备可附着在先进网络上。设备可以使用经过更新的证书来访问先进网络。先进网络实体则可以将设备作为先进设备来验证。

[0191] 用于管理验证的过程至少具有两个阶段。一个可以是用于配置的,另一个则是用于补救的。对于配置来说,网络管理实体可以验证中继节点作为确信安装用于与运营商的核心网络实施的后续验证过程的运营商证书的设备的能力。所述中继点可以借助于使用了平台验证的自发平台检验手段来向网络实体提供隐性证明。管理实体可以通过验证设备和/或核实RN制造商证书来了解RN完整性没有受损(因为自发检验可在关于设备完整性的内部核实成功的时候放出私有验证密钥)。

[0192] 对于补救来说,由于某些不重要的故障导致整个设备的完整性检查失败,因此,RN可以与管理实体执行远程修复过程。在这种情况下,如果RN的管理能力无损,那么RN可以向管理实体发送一个用于验证目的的管理能力证书,而不是整个设备的完整性证书。网络管理实体可以验证设备并核实证书。一旦成功,则网络管理实体可以通过执行远程过程来自举中继点的能力。

[0193] 在初始范围受限的情况下,可使用后一个过程来自举RN的能力。在这种情况下,第一个证书可以代表与管理实体一起执行基本操作的能力。第二个证书可以代表与管理实体一起执行更广泛的操作的能力。在这种情况下,中继点不能使用功能提升的范围来检测完整性故障。完整性检查范围还可以通过提供用以适应增强功能的附加更新策略来提升。这样一来,设备的能力可以与完整性检查的程度一起增长。

[0194] 该技术通常可以应用于安全网关验证。在针对核心网络的网关验证中可以使用代表设备受限范围的证书,以使网络能在验证期间确定验证权限。网关(例如H(e)NB验证中的SeGW或是用于RN验证的DeNB)可以验证设备并核实证书。基于证书信息以及对于特定设备身份的成功验证,网关可以限制那些出于补救或注册目的而对网络管理实体进行的访问。一旦成功配置、更新和/或补救了中继点,则可以将验证密钥提供给基于验证的增强。

[0195] 图20示出的是涉及中继节点能力自举的例示呼叫流程图。该呼叫流程序列可以包括这里描述的一个或多个步骤。举例来说,如图20所示,在2014,中继节点(RN)2002可以执行一个安全引导序列。在2016,RN2002可以建立一个安全环境和/或执行自发检验。在自发检验期间,RN2002没能发现故障。在2018,RN2002可以作为UE并通过eNB2004附着于网络,例如附着在MME/HSS2012上。在2020,MME2012可以将RN2002的受限访问指定给注册服务器2008。如图20所示,该受限访问可以是通过eNB2004指定的。RN2002和注册服务器2008可以相互验证。在2022,RN2002可以向注册服务器2008发送一个注册请求。在2024,注册服务器可以检验RN2002、配置RN2002和/或向RN2002发行一个证书。注册服务器2008可以通过对RN2002的私有注册服务器验证密钥的使用的RN2002安全环境发布来隐性检验RN2002,或者注册服务器2008也可以通过询问RN报告来检验RN2002。注册服务器2008可以配置用于DeNB2006附着的已验证RN2002,并且可以发行一个证书。注册服务器2008可以更新设备上的RN策略,以便绕过未来的重新引导处理中的注册步骤,以及将注册信息包含在自发检验处理中(例如添加包含了注册信息的TRV)。在RN2002上可以安装新的私钥,或者可以在该阶

段激活新的私钥。RN2002可以具有密钥生成能力，在这种情况下，在RN2002上可以产生密钥对。RN2002可以在其安全环境中安装私钥，并且可以将这个密钥版本绑定到注册配置以及低级安全引导阶段。在2026，注册服务器2008可以向中继节点2002指示注册结束。在2028，RN2002可以作为RN并通过DeNB2006附着于网络。在2030，MME2012可以通过DeNB2006来将RN2002的受限访问指定给配置服务器2010。在2032，RN2002可以向配置服务器2010发送一个RN配置请求。在2034，配置服务器2010可以检验RN2002、配置RN2002和/或向RN2002发行一个证书。例如，配置服务器2010可以配置用于操作的已检验RN2002，并且可以发行证书。配置服务器2010可以更新设备上的RN策略，以便绕过未来的重新引导处理的配置步骤，以及将配置信息包含在自发检验处理中（例如添加包含了配置信息的TRV）。在该阶段可以安装或者激活用于DeNB2006的验证的新私钥。在2036，配置服务器2010可以向RN2002指示配置结束。在2038，RN2002可以与DeNB2006发起S1建立处理。在2040，RN2002可以向DeNB2006发起X2建立处理。在2042，RN2002可以作为中继节点工作。

[0196] 在2044，在RN2002上有可能发生复位。例如，该复位可能是由网络、设备或停电引发的。在2046，RN2002可以执行安全的引导序列。在2048，在建立安全环境和/或自发检验期间，RN2002未能发现故障。由于注册服务器2008的信息和/或配置服务器2010的信息可以包含在2048的自发检验过程中，因此，RN2002可以继续在2050和2052分别建立S1和X2接口，而不需要与服务器接洽。在2054，RN2002也可以作为RN工作。如果注册服务器信息或配置服务器信息出现故障，那么策略可以或者不可以允许私有验证密钥发布或是后续网络处理的执行，直至重新配置。

[0197] 图21示出的是涉及使用了已验证管理能力的中继节点补救的例示呼叫流程图。该呼叫流程序列可以包括这里描述的一个或多个步骤。举例来说，如图21所示，在2114，中继节点（RN）2102可以执行一个安全的引导序列。在2116，RN2102可以建立一个安全环境和/或执行自发检验。在2116的自发检验期间，RN2102可以发现设备的非重要组件的故障。在2118，RN2102可以作为UE并通过eNB2104附着于网络（例如MME/HSS2112）。在2120，MME2112可以将RN2102的受限访问指定给注册服务器2106和/或补救服务器2110。例如，MME2112可以通过eNB2104来指定受限访问。RN2102和补救服务器2110可以相互验证。在2122，RN2102可以向补救服务器2110发送一个警报（例如补救请求）。在2124，补救服务器2110可以隐性地通过对RN2102的私有补救服务器验证密钥的使用的RN2102安全环境发布来检验RN管理能力，和/或补救服务器2110可以通过询问一个或多个RN2102的完整性状态报告来检验RN2102。在2126，RN2102和补救服务器2110可以（可选地）执行关于RN2102的询问。在2128，补救服务器2110可以补救（例如重新配置，修复或重新编程）RN2102，并且在2130向RN2102发送一个用于修复故障的补救信息。

[0198] 在2132，补救服务器2110可以远程命令RN2102安全地重新引导，或者由于RN2102已经执行了通电阶段中的第一阶段，并且用于平台验证而不仅仅例如是管理验证的设备完整性检查现在取得成功，RN2102可以直接前进至RN2102的通电序列中的下一个步骤。例如，在2134，RN2102可以建立一个安全环境和/或执行自发检验。在2136，RN2102可以作为UE并通过eNB2104附着于网络（例如MME/HSS2112）。在2138，MME2112可以将RN2102的受限访问指定给注册服务器2106和/或补救服务器2110。在2140，RN2102可以向注册服务器2106产生一个注册请求。在2142，注册服务器2106可以检验RN2102、配置RN2102和/或向RN2102发行一

个证书。在2144，该注册服务器2106可以向RN2102指示注册结束。

[0199] 这里描述的用于完整性检查、报告和补救的过程、系统和手段未必局限于这里描述的软件故障、TR-069，软件补救、H(e)NB等等。更进一步，所描述的操作是例示性的。应该理解的是，其他操作也是可以使用的，在不使用的情况下是可以省略的，和/或操作是可以添加的。

[0200] 在一些实施方式中，故障未必是针对软件的，而是针对设备上的配置数据或其他某些可量度组件的。在该实施方式中，举例来说，管理实体接收的更新未必来自软件补救实体，而是来自网络设备配置数据库。并且不是所有故障情景都可以使用询问过程。在这些情况中，作为来自设备的警报或报告发送的信息起初足以确定故障原因，或者至少足以确定可以执行的某种操作，其中管理实体可以借助该操作而在没有询问的情况下发起补救过程。举个例子，管理实体可以从类似设备的先前故障中识别出发生故障的组件量度，由此立即向设备提交更新。询问与更新之间的时间和业务量负载的折衷有可能是因为很小的组件造成的，并且这种折衷可以触发整个组件更新。

[0201] 在以下示例中描述了设备架构和补救更新过程。该例示设备架构和补救更新过程可以与复杂度和资源有限的小型设备一起使用。在这里描述的是与本示例相关联的限制。例如，就开发和/或部署生命周期以及操作而言，设备的应用代码库和可信代码库可以是独立的。这两个代码库可以是由独立的各方开发和/或部署的。可以提供设备的生产功能的应用代码库可以作为单个代码和数据块而被部署到设备的非易失存储器的某个部分。在补救过程中可以最低限度地涉及应用代码库，和/或在完整性检验过程中可不所述应用代码库。服务周期(例如设备代码库一部分的更新之间可供设备应用使用的可使用时间)有可能很长。举例来说，这意味着不能在任何时间不加区别地强制执行用于补救的重新引导。设备的正常操作不能因为补救或设备更新过程而被中断。用于完整性检验和补救的通信可以在设备启动的较早阶段进行，在此期间可以应用关于设备内部资源使用和通信带宽的严格限制。从可以加载并且随后逐一启动组件(例如程序和数据)的典型的引导循环的意义上讲，复杂的系统启动未必是存在的。

[0202] 例示的系统模型可以被描述成是在考虑了上述限制的情况下将系统拆分成TrE和正常组件的一般形式。系统架构的功能元件可以包括下列各项中的一项或多项：

[0203] RoT(可信根)，它可以是设备完整性检验处理在信任链中依靠的不变要素；

[0204] SEE(安全的执行环境)，它可以是对可执行代码和数据进行硬件和/或软件保护并且与系统的剩余部分隔离的特殊的执行环境。一种可能的实现方式可以是处理器或是处理器中的单个核心的安全执行模式。对于SEE的限制可以包括对运行时存储器以及可用的(专为SEE所有)非易失存储器(NVS)的限制。

[0205] 通信IF(通信接口)组件，它可以将基本通信能力暴露给SEE和/或NEE；和/或

[0206] NEE(正常执行环境)，它可以是设备中的应用代码的执行环境，其中举例来说，所述代码可以是不属于TrE的代码。NEE可以具有某些与SEE对接的接口，例如对接到通信IF以重新使用该通信能力的接口。

[0207] 这些功能元件可以供包含设备能力和组件的若干代码库使用。所述代码库可以分成TrECB(TrE代码库)和DACB(设备应用代码库)。TrECB可被分配给SEE，而DACB则可以被分配给NEE。如果DACB组件将要访问TrE的能力，那么该访问可以借助所述及的NEE与SEE之间

的接口执行。在诸如安全启动之后之类的从SEE到NEE的运行时间是没有执行控制的，反之亦然。TrECB和DACB可以保存在NVS的单独的部分。

[0208] 图22示出的是具有功能组件2202(左手侧)和代码/数据存储组件2204(右手侧)的例示系统。功能组件2202可以包括NEE2206和SEE2208。SEE2208可以包括通信接口2210和/或ROT2212。代码/数据存储组件2204可以包括NVS组件。NVS组件TRV_tmp2214可以包括与NEE2206相关联的TRV。NVS组件DACB2216可以包括DACB。NVS组件TRV NVS2218可以包括与SEE2208相关联的TRV。NVS组件TrECB NVS2220可以包括TrECB。不同的安全措施可以单独应用于NVS组件。这里的箭头可以指示图22所示实体之间的读/写访问。TrECB NVS2220可以提供不同的能力，例如完整性检验能力(IVC)、设备的常规通信能力(DGC)、回退/遇险能力(FB/DC)以及补救能力(RC)。在这里对照图22和图23论述了这些能力。

[0209] 如图22所示，NEE2206可以与SEE2208通信(例如从中读取和/或写入)。SEE2208可以使用通信接口2210来进行通信。所述SEE2208可以使用ROT2212来与代码/数据存储组件一起执行完整性检验。NEE2206可以写入NVS组件2214，以及读取NVS组件2216。SEE2208可以读取NVS组件2214。所述SEE2208还可以读取和写入NVS组件2216、2218和2220。

[0210] 图23示出的是引导序列的阶段以及在每个阶段实施的不同实体间的交互的例示流程图。如图23所示，在引导序列中可以实施NEE2206和SEE2208。SEE2208可用于引导序列的第一和第二阶段。第一阶段可以合并ROT2212。第二阶段可以合并TrECB NVS2220，所述TrECB NVS2220可以包括完整性检验能力(IVC)2318、设备的常规通信能力(DGC)2314、回退/遇险能力(FB/DC)2320、补救能力(RC)2324和/或TRV2326。IVC2318可被指定获取组件量度以及将其与TRV2326之类的TRV中包含的基准值相比较的任务。DGC2314可以向SEE2208提供基本的通信功能(所述功能转而可以暴露于NEE2206)。所述DGC2314可用于FB/DC2320和RC2324实施的补救和/或遇险指示。DGC2314同样可以经由接口暴露于DAC2322。FB/DC2320可以在满足某些条件的情况下被激活，以便执行相关功能，即分别使用回退代码库来替换DACB2216，由此向网络或设备用户指示遇险状态。RC2324可以是被指定了关于代码库的计划变更和/或校正的任务的机构。此外，RC2324还可以是TRV2326的管理器，并且RC2324能检查TRV2326的真实性。

[0211] NEE2206可用于引导序列的第三阶段。第三阶段可以合并DACB2216，其中所述DACB包括补救应用IF(RAIF)2316和/或设备应用能力(DAC)2322。RAIF2316可以是用于来自网络的到来的新TRV的传递接口。RAIF2316可以识别到来的通信中的TRV，并且将其保存在TRV临时非易失存储器中。DAC2322可以实现设备的应用专用功能。如果DAC希望使用来自通信IF的通信能力，那么对其进行的访问可以通过特殊的NEE-SEE IF来传达。

[0212] TRV2326可以保存在系统中的两个不同位置。TRV_tmp2214可以是用于RAIF2316接收的新TRV的临时存储器，例如，TRV_tmp2214可被从NEE2206写入。TRV_tmp可被从SEE2208读取，并且RC2324可以从中读取新的TRV，以及在对其进行了验证之后将其置入TRV NVS2218。

[0213] 图23所示的启动序列可以涉及简化系统架构中带有完整性检查的正常启动，例如，没有出现有可能需要遇险/回退或补救操作的故障状况的启动处理。在安全启动处理中的第一个阶段，RoT2212可被激活。与先前描述的安全启动的第二阶段不同，RoT2212可不核实并且激活随后可对已加载和启动的组件执行进一步的完整性检查的可信根加载器。这种

加载器/执行控制器可能不能在简化的系统架构中使用。取而代之的是，RoT2212可以在TrECB NVS2220中检查IVC2318的代码和数据块。在该结构中，由于RoT2212不能读取TRV2326，因此，IVC2318的代码有可能是不变的。由此，它可以对照固定的内置(在RoT2212中)基准值来检查IVC2318。

[0214] 在一个实施例中，RoT2212可以使用固定的根证书来检查IVC2318的代码和数据块。为此，TrECB NVS2220的IVC2318部分可以与处于TrECB NVS2220中的后一个IVC2318部分的签名一起保存。RoT2212用以检查签名的密钥可以处于所述提及的固定根证书中。通过下载新的IVC2318代码以及使用相同固定密钥的新代码的新签名，并且将后一个数据保存在TrECb NVS2220中，可以实施IVC2318的代码变化。IVC2318可以在SEE2208内部执行。IVC2318可以检查出TRV_tmp NVS2214为空(这可以是为简化系统架构中带有完整性检查的正常启动假设的)。IVC2318可以从TRV NVS2218中加载用于DGC2314的代码的指定TRV。

[0215] IVC2318可以通过核实TRV基准值以及任何附加数据的签名来检查TRV2326中的每一个被加载TRV的完整性。该签名可以处于TRV中，而IVC2318用以检查签名的密钥则处于IVC2318代码/数据块中。然后，IVC2318可以量度从TrECB NVS2220加载到SEE2208的DGC2314代码，并且将所述量度与后一个TRV中的基准值相比较。一旦成功，则可以激活DGC2314。激活意味着DGC2314可供执行，其中举例来说，所述DGC是在假设SEE2208的处理器和NEE2206的处理器在检查NEE2206代码时会遵守在DGC2314所处的SEE2304的运行时存储器部分上设置的标记“executable(可执行)”情况下通过设置所述标记来执行的。

[0216] IVC2318可以从TRV NVS2218中加载为RAIF2316的代码指定的TRV。IVC2318对从DACB NVS2216加载到NEE2206的RAIF2316的代码进行量度，并且将所述量度与后一个TRV中包含的基准值相比较。一旦成功，则可以激活RAIF2316。

[0217] IVC2318可以加载与预定序列中的DACB2216的某些部分相关联的每一个TRV。IVC2318可以量度由所加载的TRV指定并从DACB NVS2216加载到NEE2306的DAC2322的代码和数据的某些部分，并且将所述量度与后一个TRV中包含的基准值相比较。

[0218] 当执行DACB2216中的检查时(例如通过穷举可用于DACB2216的代码和数据的TRV序列)，IVC2318可以激活DAC2322，并且将执行过程传递到NEE2206的处理器(或者在SEE2208和NEE2206的处理器可以同时运行的时候启动该处理器)。如果在启动过程中没有出现特殊状况(例如完整性检查故障)，那么可以既不检查也不激活FB/DC2320和RC2324的代码和数据块。

[0219] 与诸如安全引导之类的更复杂系统中的安全启动的差别可以包括：IVC2318可以由TRV数据驱动。换言之，TRV可以具有关于不同代码库的哪些代码段和数据将被检查的信息。TRV2326可以由IVC2318顺序读取，其中所述IVC2318可以对其进行评估，以便发现应用了TRV基准完整性值的代码段和数据，以及读取和/或量度所述数据，并且将经过校对的量度与基准值相比较。

[0220] 由于单个TRV基准值可以对应于代码库中的多个代码段和数据，因此，TRV可以包括一个映射，例如关于代码库中的这些段的位置和长度的指示符，以及如何将这些段的量度合并成混合量度值的规定。图24A和24B示出了将TRV映射至代码库中的代码段和/或数据的例示实施方式。

[0221] 图24A是显示了可被线性组合以创建TRV的局部量度序列的图示。如图24A所示，

DACB的代码量度2402、2404、2406、2408和2410可被线性组合，以便创建TRV2412。根据一个例示实施例，代码量度2402、2404、2406、2408和2410可以通过应用散列链来组合，其中所述散列链是用TCG规定的可信平台模块的TPM扩展命令实现的。

[0222] 图24B示出的是使用Merkle散列树来创建TRV的值的组合的图示。如图24B所示，DACB的代码量度2416、2418、2420、2422、2424和2426可以用Merkle散列树组合，以便创建TRV2414。

[0223] 在执行补救和/或更新的设备与相应网络实体(例如H(e)MS)之间执行的交互询问过程可以使用这里描述的TRV-代码/数据段映射。这里描述的这些过程可以是这里描述的通用设备询问过程的特例。

[0224] 每一个代码段都可被量度，并且所述量度可被逐一发送到网络实体，在那里它会与代码段基准值的顺序列表中的相应代码段基准值比较，其中所述TRV基准值是先前通过诸如散列链之类的某种方法而从该列表中计算的。

[0225] 如果借助Merkle散列树检测到大量代码段，那么可以提高效率。该处理可以采用交互询问过程并通过递减树的等级来执行。在这里，TRV中包含的基准值以及从代码/数据中取出的量度值均可表示相同二叉树的根(在图表理论上)。如果它们不匹配，那么设备可以将这两个子节点值发送到网络实体，所述网络实体则可以确定哪一个存在故障，例如不与网络用以构件TRV基准值(它可为所述参考树的根)的参考树中的相同节点相匹配的子节点值。网络可以向设备发送一个可以声明哪个(些)分支失配的消息。该过程可以重复进行，直至确定了存在参考(叶片)值失配的代码段、构成量度树叶片的量度值。

[0226] 回过来参考图23，在这里可以执行那些以用于设备补救的TrECB2220的能力为基础的功能和过程。在一个实施例中，计划代码更新可以通过在操作期间更新一个或多个TRV2326以及在下一次启动时执行实际代码更新来执行。

[0227] 以下涉及的是计划进行的代码更新。可以假设设备已经执行了这里描述的启动处理。例如，RAIF2316可以经由通信IF2210和/或DGC2314接收来自H(e)MS之类的外部方的新TRV。RAIF2316可以将新接收的TRV保存在TRV_tmp NVS2214中。在以后的时间，设备可以重启。IVC2318可被执行完整性检查，并且在这里描述的SEE2208内部启动。IVC2318可以检查并发现TRV_tmp NVS2214非空，并且可以采用这里描述的方式继续前进。

[0228] TRV_tmp2214可以具有一个单独的新TRV。在第一个实施方式中，TRV_tmp中的新TRV可以是指DACB2216中的代码和/或数据。在第二个实施方式中，TRV_tmp中的新TRV可以是指TrECB2220中的代码和/或数据，例如DGC2314、FB/DC2320或RC2324的代码/数据。

[0229] 在第一实施方式中，IVC2318可以采用上述方式来核实TRV的真实性。一旦成功，则IVC2318可以将新TRV保存在TRV NVS2218中。IVC2318可以删除TRV NVS2218中的一个或多个旧TRV，其中所述旧TRV可以视为被新TRV取代。如何确定这些弃用TRV可以取决于实施方式。唯一的标识符可以作为TRV中的附加数据的一部分指定给TRV。举例来说，本段中描述的处理可被称为TRV摄入。

[0230] IVC2318可以量度从TrECB2220NVS加载到SEE2208的DGC2314的代码，并且将该量度与后一个TRV中包含的基准值相比较。一旦成功，则可以激活DGC2314。IVC2318可以加载和/或核实来自TRV NVS的TRV2326，并且可以为这其中的每一个指定的代码段或数据执行IV；其中举例来说，所述IV是以RAIF2316为开始并且前进至DACB2216的其他部分。当在IV序

列中遭遇到新摄入的TRV时，在DACB2216的代码和数据的指定部分上执行的IV必然失败(例如，假设新TRV具有与被弃用的一个或多个TRV不同的参考值)。

[0231] IVC2318可以从TRV NVS2218中加载用于RC代码的指定TRV。然后，IVC2318可以量度从TrECB2220NVS加载到SEE2208的DGC2314代码，并且可以将该量度与后一个TRV中包含的基准值相比较。一旦成功，则可以激活RC。

[0232] 通过与H(e)MS之类的相应网络实体一起实施的询问过程，RC2324可以确定诸如导致完整性量度故障的代码段和/或数据之类的需要更新的代码段和/或数据，以便重新产生新摄入的TRV的基准值。其中举例来说，该询问过程可以采用这里描述的方式执行。

[0233] 借助新的TRV，设备还可以接收哪些部分的代码和/或数据需要替换的详细资料。这样做可以消除RC2324与网络之间的询问过程。通过执行询问，可以将在设备的正常操作过程中下载且用于设备管理和/或补救的数据量减至最小。此外，通过执行询问，还可以允许设备“遗漏”某些(一个，新的)TRV和/或为其指定的代码段的某些中间更新的可能性。如果这种后续更新是累积的，那么它们往往会影响有可能在询问过程中发现的更大数量的代码段(但是不保证其处于仅限于最后一个TRV的更新所指定的代码段列表，其中在对同一个TRV进行一系列更新之后，其可能已经被设备遗漏)。

[0234] RC2324可以下载所确定的代码段和/或从相应网络实体确定的数据。根据一个示例，网络实体可以编译用TR-069签名的数据包和/或将其发送至设备。RC2324(或是IVC2318)可以检查接收数据的真实性，例如使用新摄入的TRV中的签名证书、用于检查TRV2326的根证书(例如由IVC2318)或是用于补救目的的专用证书(例如在RC2324的代码/数据库中)来核实数据包签名。

[0235] 在验证了已下载的代码段和/或数据之后，RC2324可以将其写入DACB2216NVS中的先前确定的片段位置。DGC2314可以将执行返还给IVC2318，所述IVC2318可以在同一个TRV上重新开始IV，其中举例来说，所述TRV可以是TRV序列中的新摄入的TRV。

[0236] 上述过程可以是循环的。这种情况会因为至少两个原因中的一个原因而出现。首先，攻击者可以将不与TRV基准值一致的自己的代码插入更新过程。此外，举例来说，由于网络侧的代码构建发生故障，TRV基准值和/或下载的代码有可能出现偶然的失配。在两种实施方式中，这些状态都是可以检测并用信号通告给网络的。该处理可以由IVC2318通过使用关于TRV使用率的重复计数器来实现。为了实现高保密性，这些计数器可以是通过针对TRV NVS的读取访问而递增的单调硬件计数器。如果检测到IV在单个TRV上重复次数过多(其数量可以取决于策略)，那么IVC2318可以检查并激活FB/DC2320，和/或将控制权传递给向网络发送一个相符信号的能力。

[0237] 与如上所述的第二实施方式(与参考TrECB2220中的代码和/或数据的TRV_tmp2214中的新TRV相关)相比，在这里未必使用第一实施方式(与参考DACB2216中的代码和/或数据的TRV_tmp2214中的新的TRV相关)，这是因为所述更新/补救有可能是为更新/补救过程中包含和活动的组件本身请求的。在这种情况下，以下过程中的一个或多个步骤可以被应用。IVC2318可以摄入新的TRV，但是可以将相应的旧TRV保持在TRV NVS2218中。新的TRV可以用诸如字串“NEW_DGC_TRV”之类的某个数据标志来进行标记，以便表明它是新的。IVC2318可以使用旧TRV检查和/或激活RC2324。RC2324则可以执行TrECB2220的某些部分的更新，其中所述更新可以以与第一实施方式中的描述相同的方式使用，但在第二实施方式

中，所述更新可被写入TrECB2220NVS。IVC2318可以使用新TRV检查TrECB2220的更新部分。一旦成功，则可以从TRV NVS2218中删除旧TRV，并且可以移除附着于新TRV的标记。其中举例来说，该处理可被延期至设备下一次重启之后。

[0238] 关于完整性检查的第一种故障状况有可能会在RoT2212检查IVC2318的代码的时候出现。如果未通过检查，则RoT2212可以停止系统，并且还可以向用户发送一个信号（例如光信号）。

[0239] 根据一个实施例，RoT2212能够检查FB/DC2320中的不变的部分（或是完整性受到如在用于IC代码的类似变体中描述的签名保护的可变部分），和/或调用遇险/回退过程的这些受限部分，其中所述部分可以借助这个受到自发检查的代码来得到，例如通过将所述代码加载到SEE2208并且在SEE2208中执行所述代码。

[0240] 接下来的可能失败的完整性检查可以是关于DGC2314和/或RAIF2316的完整性检查。前者可以是指设备没有可信赖的通信能力。在这种情况下，IVC2318可以尝试核实并激活FB/DC2320。FB/DC2320在某种程度上能够恢复可信赖的通信，并且能向网络发送遇险信号。如果不能的话，它可以用信号通告用户并停止系统。如果RC2324的IV在上述补救过程中发生故障，那么相同的过程也是可以应用的。

[0241] 在如上所述的第二实施方式中，如果RAIF2316的IV发生故障，则这可能意味着设备有可能丧失了接收TRV更新的能力。然后，设备首先可以尝试如上所述那样补救该状况。如果失败，则IVC2318可以核实和/或激活FB/DC2320，所述FB/DC2320转而可以采取特定操作，例如用某个默认代码替换RAIF2316。

[0242] 从以上描述中可以看出，作为在NEE2206中暴露的代码的一部分以及正常代码库的一部分，RAIF2316可能是设备补救中的最薄弱环节。由于在完整性检查中不会包含RAIF2316，因此，这种状况不会直接威胁设备完整性，但是，它有可能会禁用更新/补救并且由此将设备保持在弃用（例如故障）状态，从而为间接和拒绝服务攻击开启方便之门。根据一个实施例，RAIF2316实现的功能可以作为TrECB2220的一部分来提供，并且是在SEE2208中执行的。这样做可以对系统架构施加提前配置，因为这可能意味着SEE2208的一部分是活动的，并且预备接收新的TRV。例如，SEE的这种永久性活动可以在通信IF2210和DGC2314中实现。

[0243] 尽管以上以特定的组合描述了特征和元素，但是一个本领域普通技术人员将理解，每个特征或元素可以单独地或与其它的特征和元素任意组合地使用。此外，在此描述的方法可实施为整合在由计算机或处理器执行的计算机可读介质中的计算机程序、软件或固件。计算机可读介质的示例包括电子信号（通过有线或无线连接发送）和计算机可读存储介质。计算机可读存储介质的示例包括但不限制为只读存储器（ROM）、随机存取存储器（RAM）、寄存器、缓冲存储器、半导体存储器设备、诸如内部硬盘和可移除磁盘这样的磁性介质、磁光介质和诸如CD-ROM盘和数字通用盘（DVD）这样的光介质。与软件相关联的处理器可用来实施在WTRU、UE、终端、基站、RNC或任何主计算机中使用的射频收发信机。

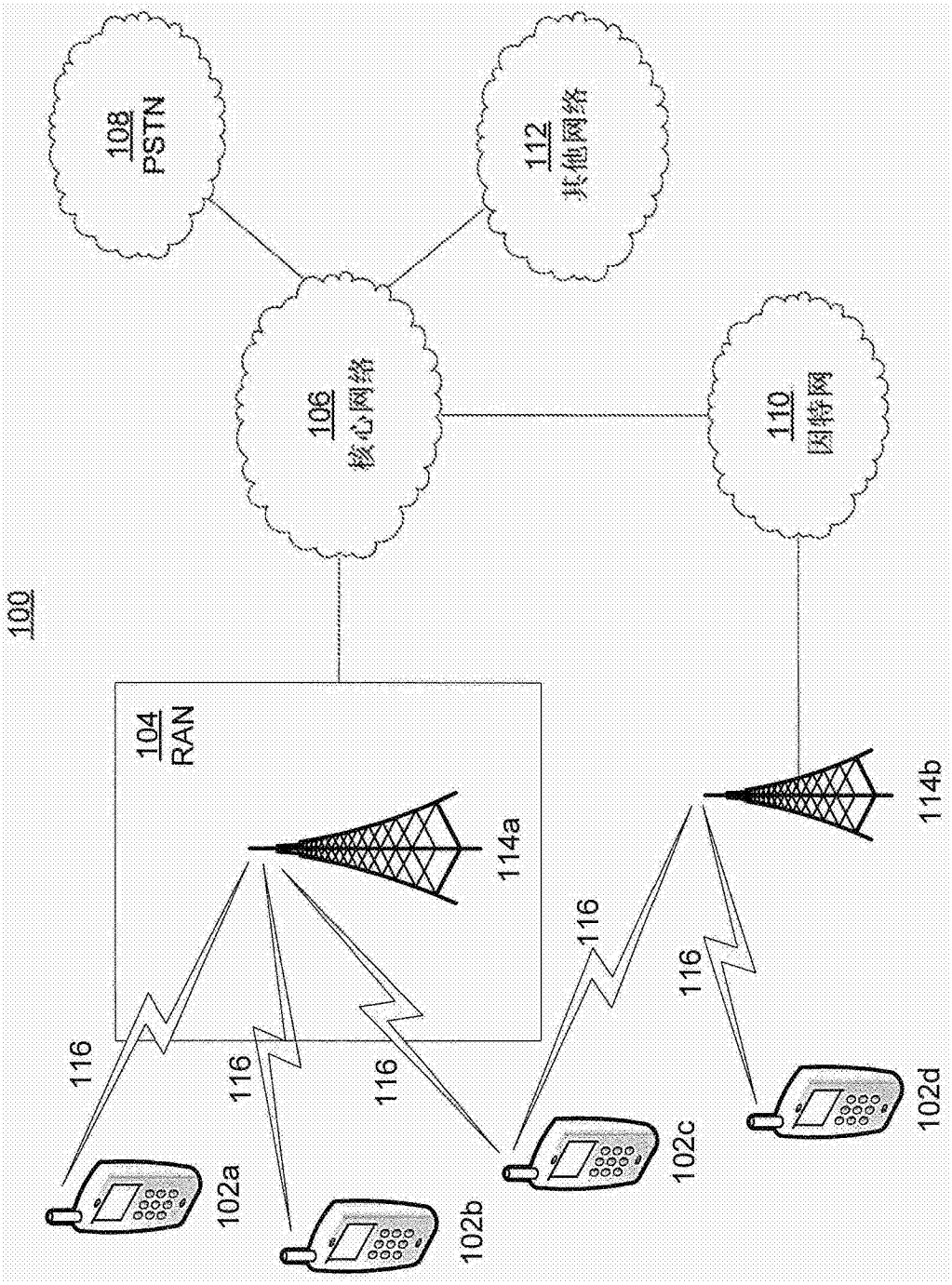

108
PSTN
112
其他网络
106
核心网络
110
因特网
100
104
RAN
114a
114b
116
116
116
116
116
102a
102b
102c
102d

图1A

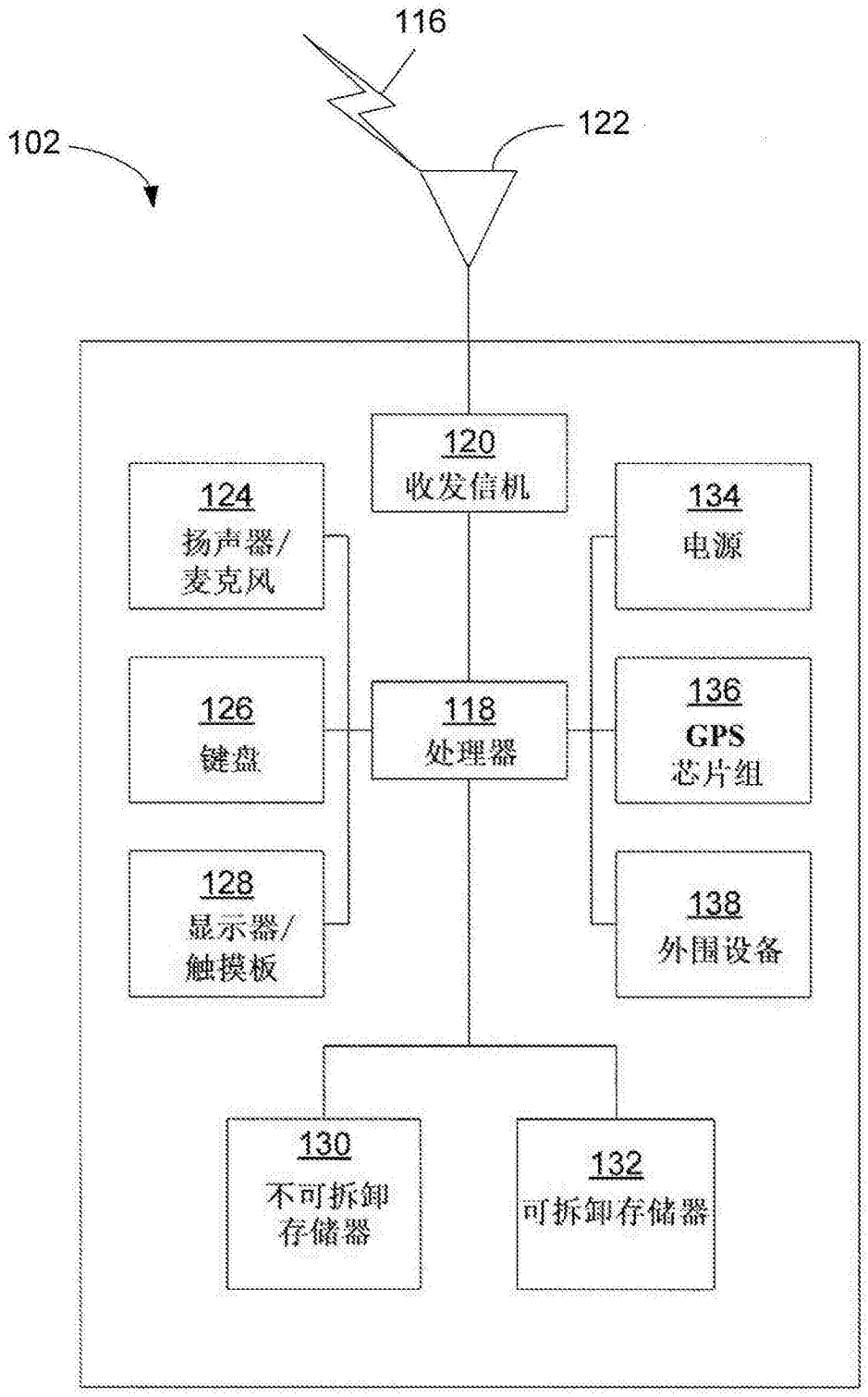

116
122
102
120
收发信机
124
扬声器/
麦克风
134
电源
126
键盘
118
处理器
136
GPS
芯片组
128
显示器/
触摸板
138
外围设备
130
不可拆卸
存储器
132
可拆卸存储器

图1B

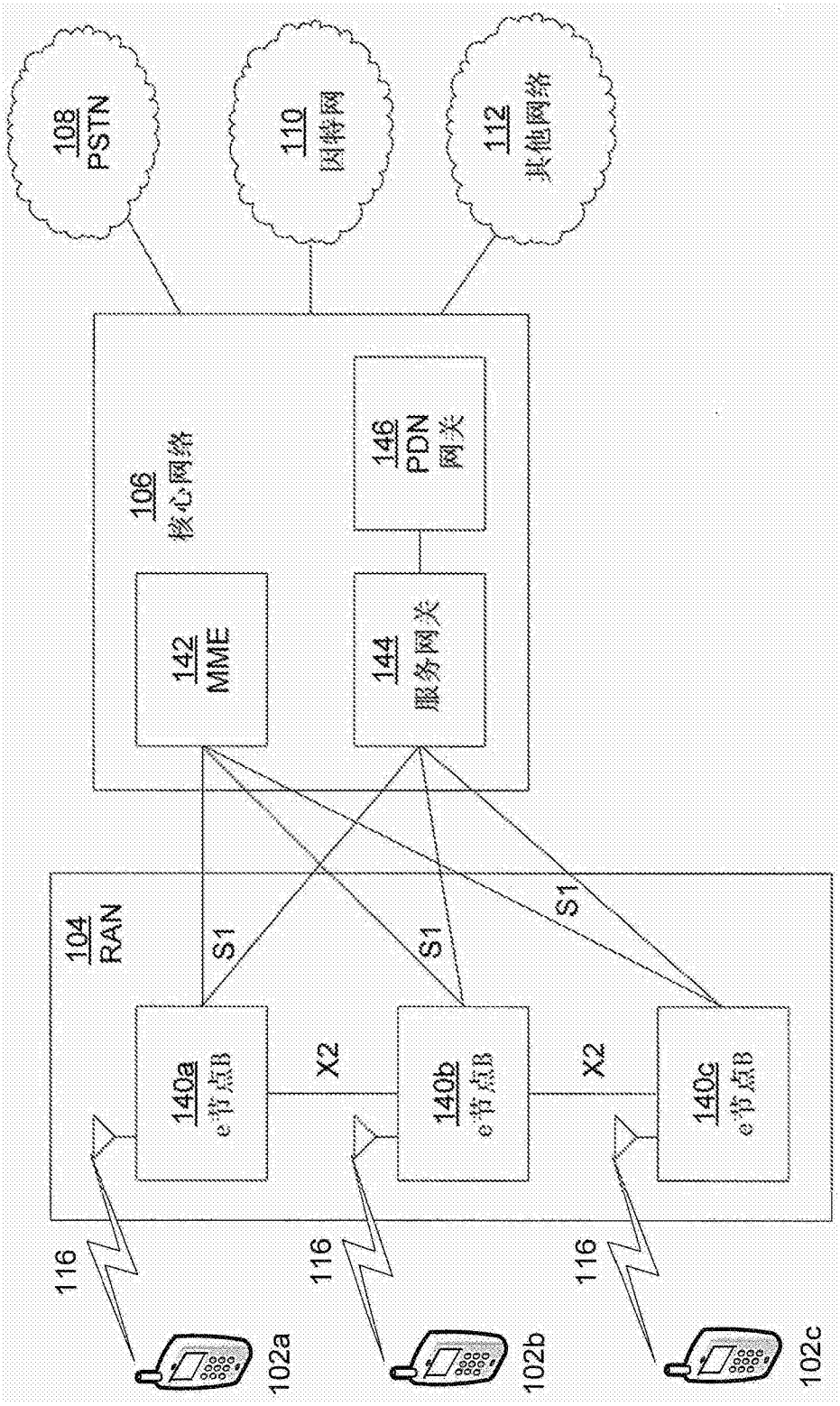

108 PSTN
110 因特网
112 其他网络
106 核心网络
142 MME
144 服务网关
146 PDN 网关
104 RAN
S1
S1
S1
140a e节点B
140b e节点B
140c e节点B
X2
X2
116
116
116
102a
102b
102c

图1C

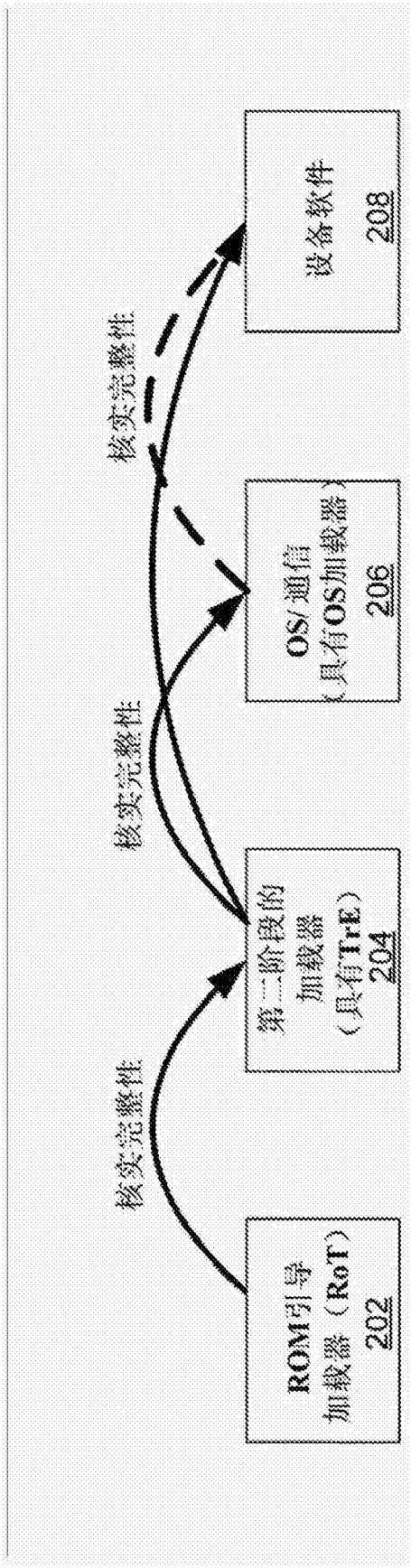

ROM引导
加载器（RoT）
202
第二阶段的
加载器
（具有TrE）
204
OS/通信
（具有OS加载器）
206
设备软件
208
核实完整性
核实完整性
核实完整性

图2

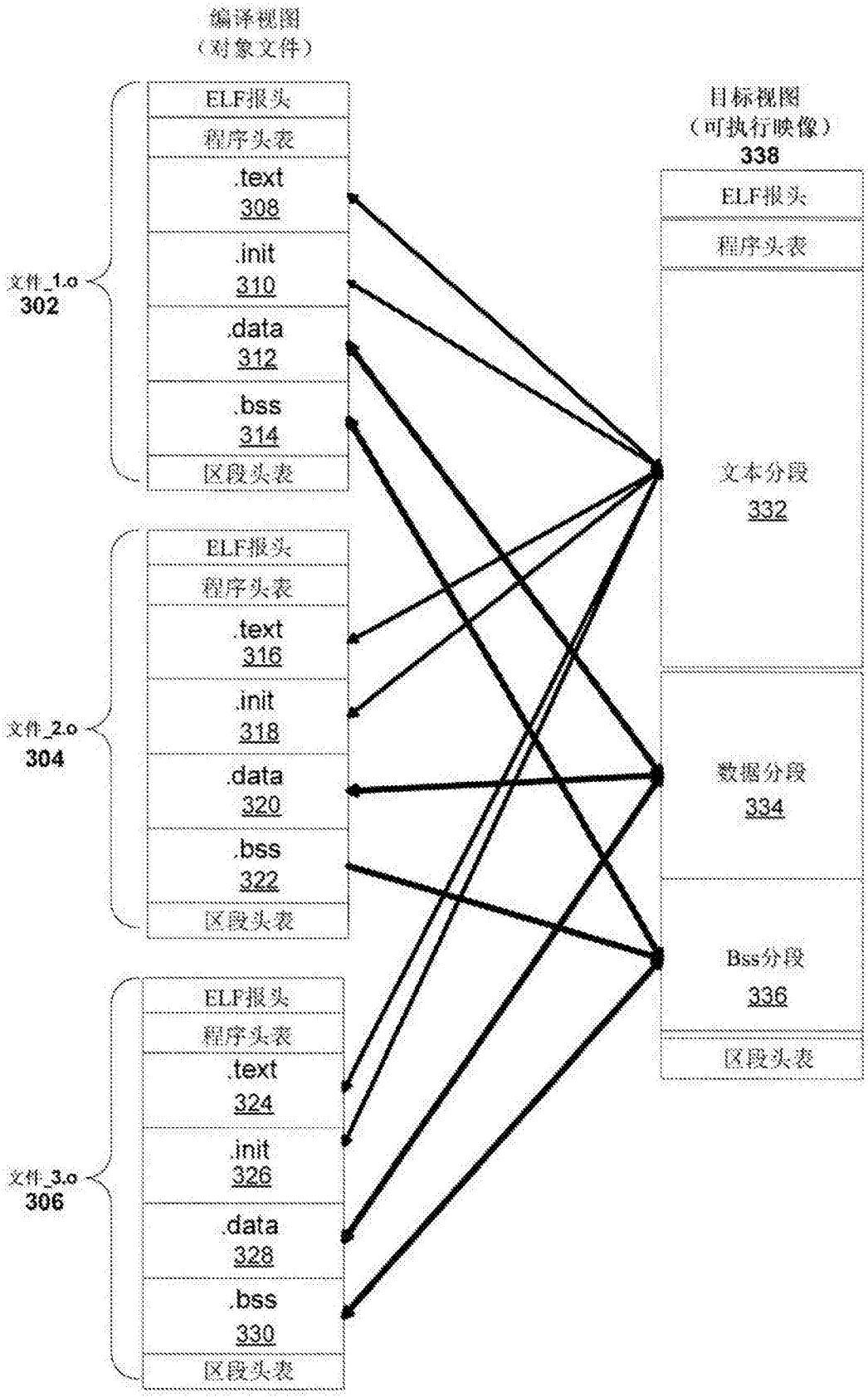

编译视图
（对象文件）
目标视图
（可执行映像）
338
文件_1.o
302
ELF报头
程序头表
.text
308
.init
310
.data
312
.bss
314
区段头表
文件_2.o
304
ELF报头
程序头表
.text
316
.init
318
.data
320
.bss
322
区段头表
文件_3.o
306
ELF报头
程序头表
.text
324
.init
326
.data
328
.bss
330
区段头表
ELF报头
程序头表
文本分段
332
数据分段
334
Bss分段
336
区段头表

图3

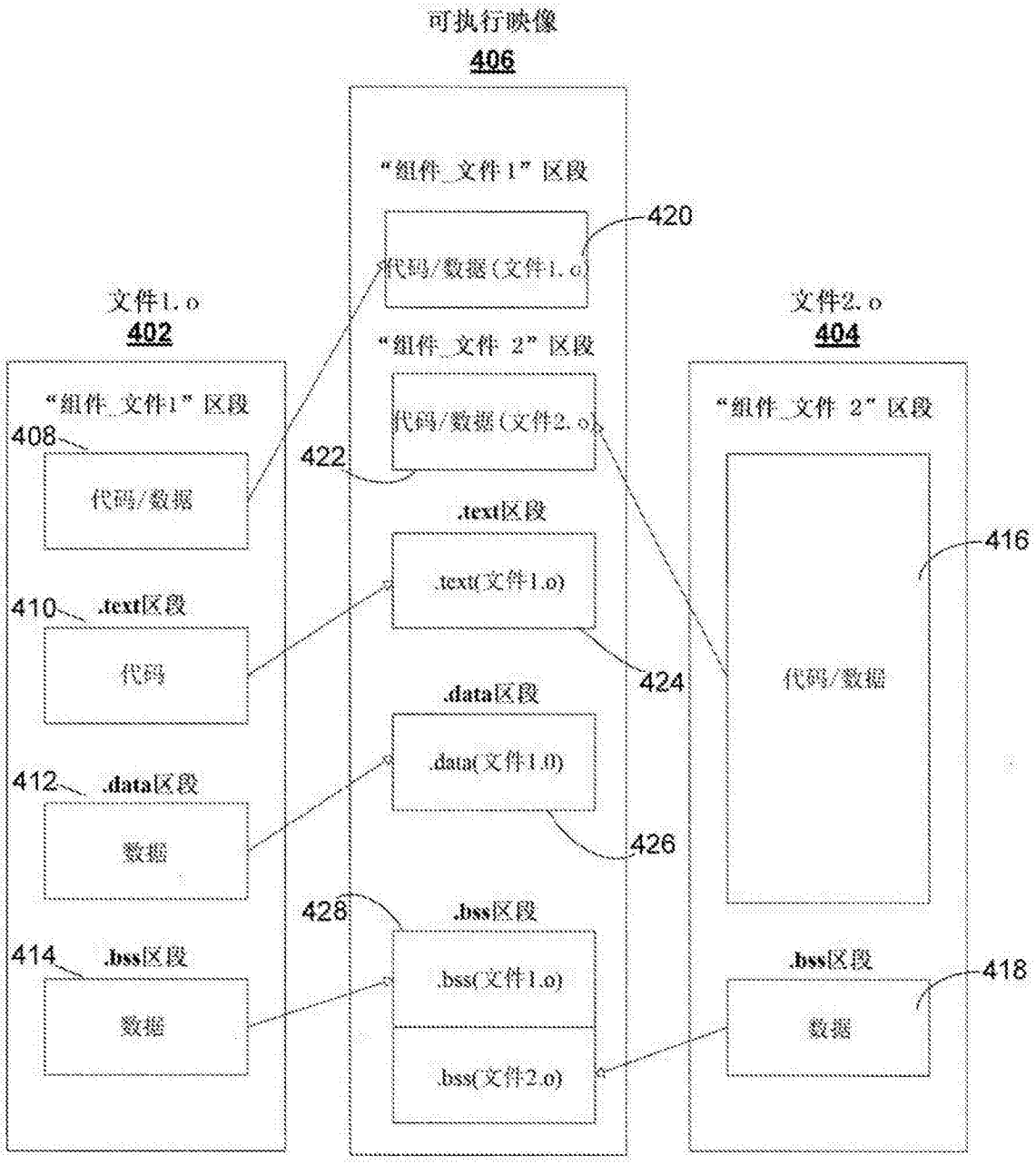

可执行映像
406
“组件_文件1”区段
420
代码/数据(文件1.o)
文件1.o
402
“组件_文件2”区段
代码/数据(文件2.o)
文件2.o
404
“组件_文件1”区段
408
代码/数据
422
.text区段
.text(文件1.o)
416
410
.text区段
代码
424
代码/数据
.data区段
.data(文件1.0)
412
.data区段
数据
426
428
.bss区段
.bss区段
414
.bss区段
.bss(文件1.o)
418
数据
数据
.bss(文件2.o)

图4

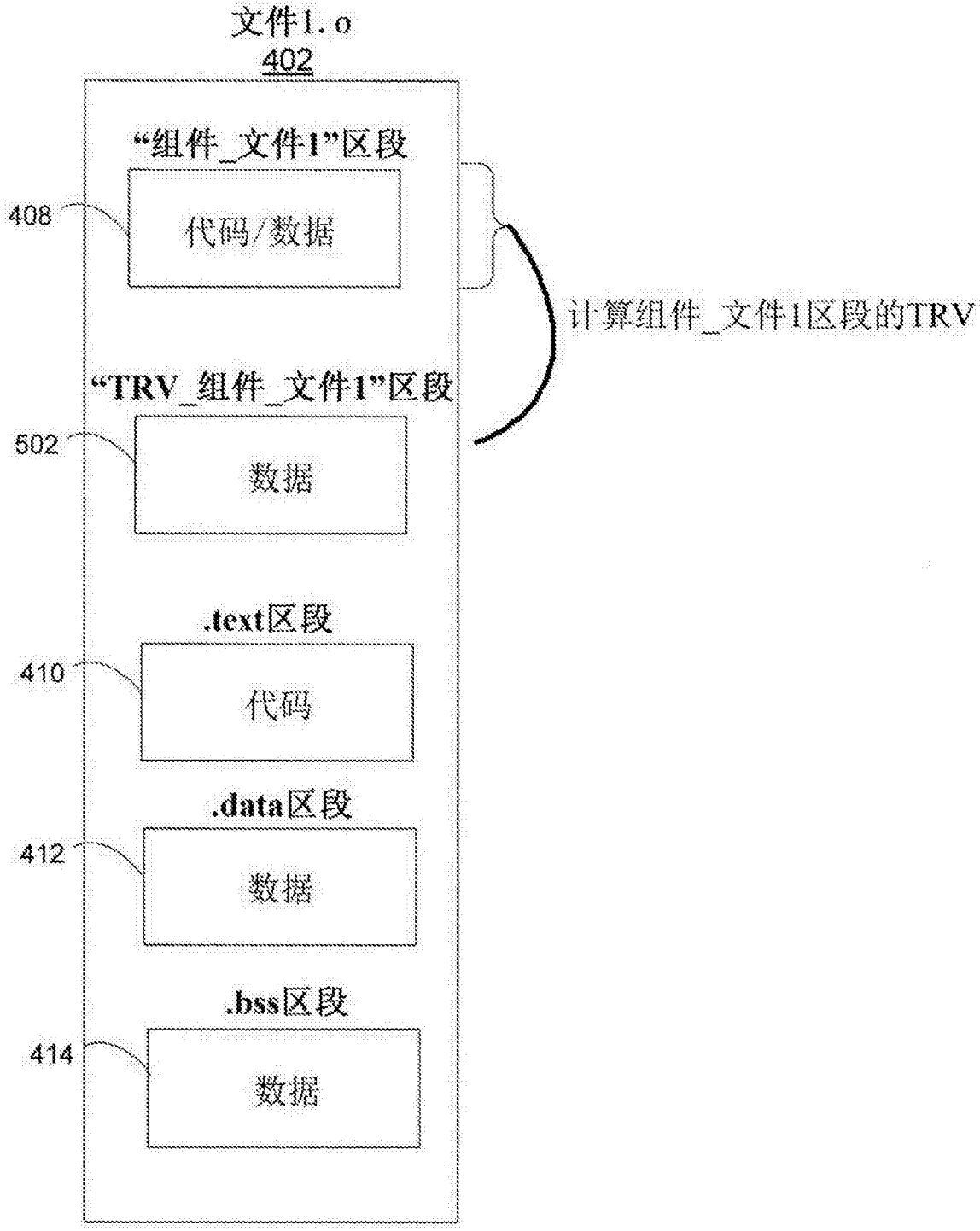

文件1.o
402
“组件_文件1”区段
408
代码/数据
计算组件_文件1区段的TRV
“TRV_组件_文件1”区段
502
数据
.text区段
410
代码
.data区段
412
数据
.bss区段
414
数据

图5

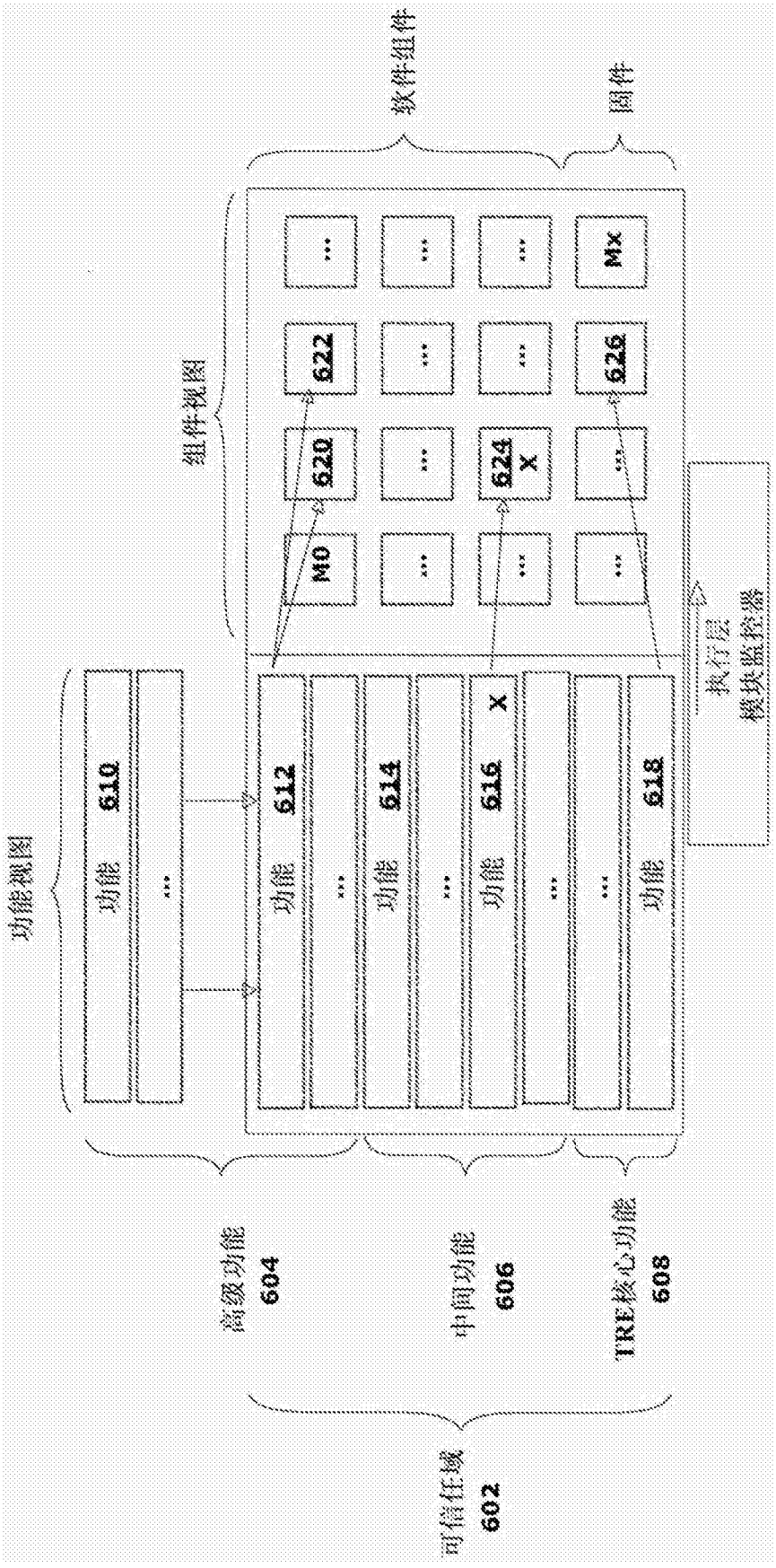

软件组件
固件
组件视图
M0
620
622
624
X
626
Mx
执行层
模块监控器
功能视图
功能 610
功能 612
功能 614
功能 616
X
功能 618
高级功能
604
中间功能
606
TRE核心功能
608
可信任域
602

图6

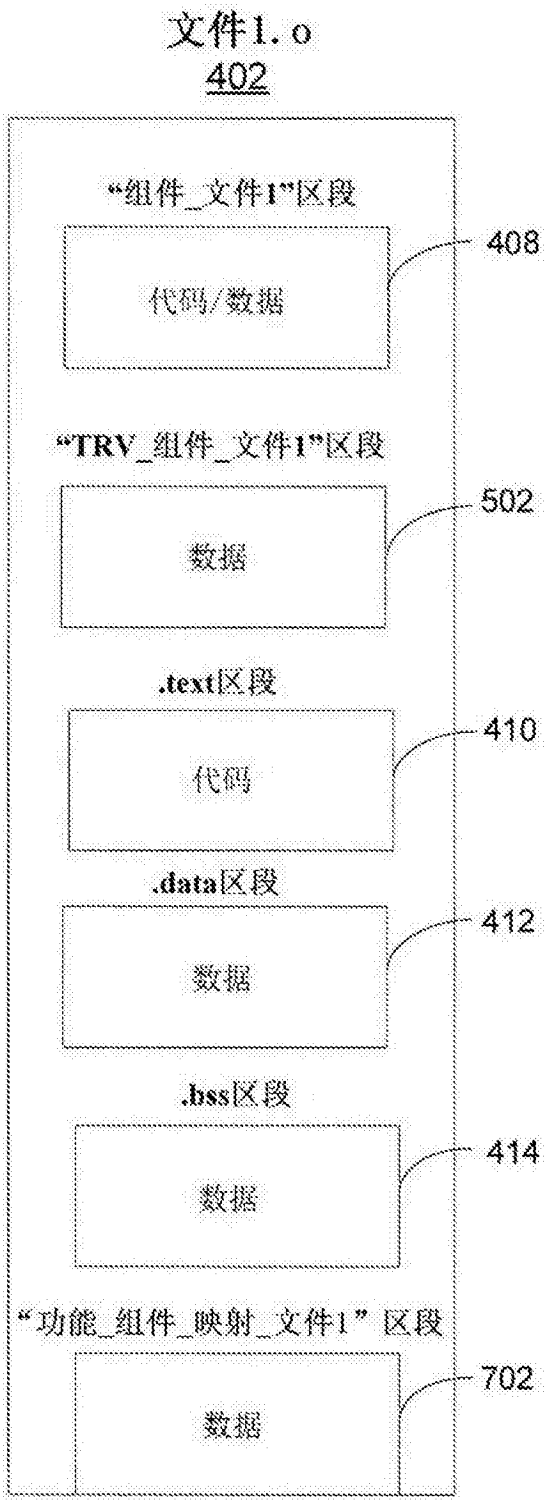

文件1.o
402
“组件_文件1”区段
代码/数据
408
“TRV_组件_文件1”区段
数据
502
.text区段
代码
410
.data区段
数据
412
.bss区段
数据
414
“功能_组件_映射_文件1”区段
数据
702

图7

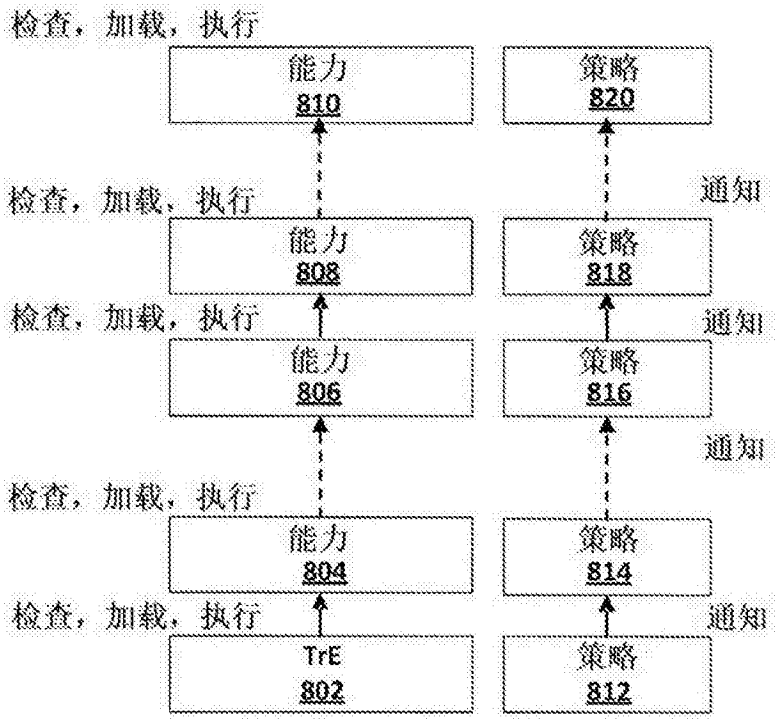

检查，加载，执行
能力
810
策略
820
通知
检查，加载，执行
能力
808
策略
818
通知
检查，加载，执行
能力
806
策略
816
通知
检查，加载，执行
能力
804
策略
814
通知
检查，加载，执行
TrE
802
策略
812

图8

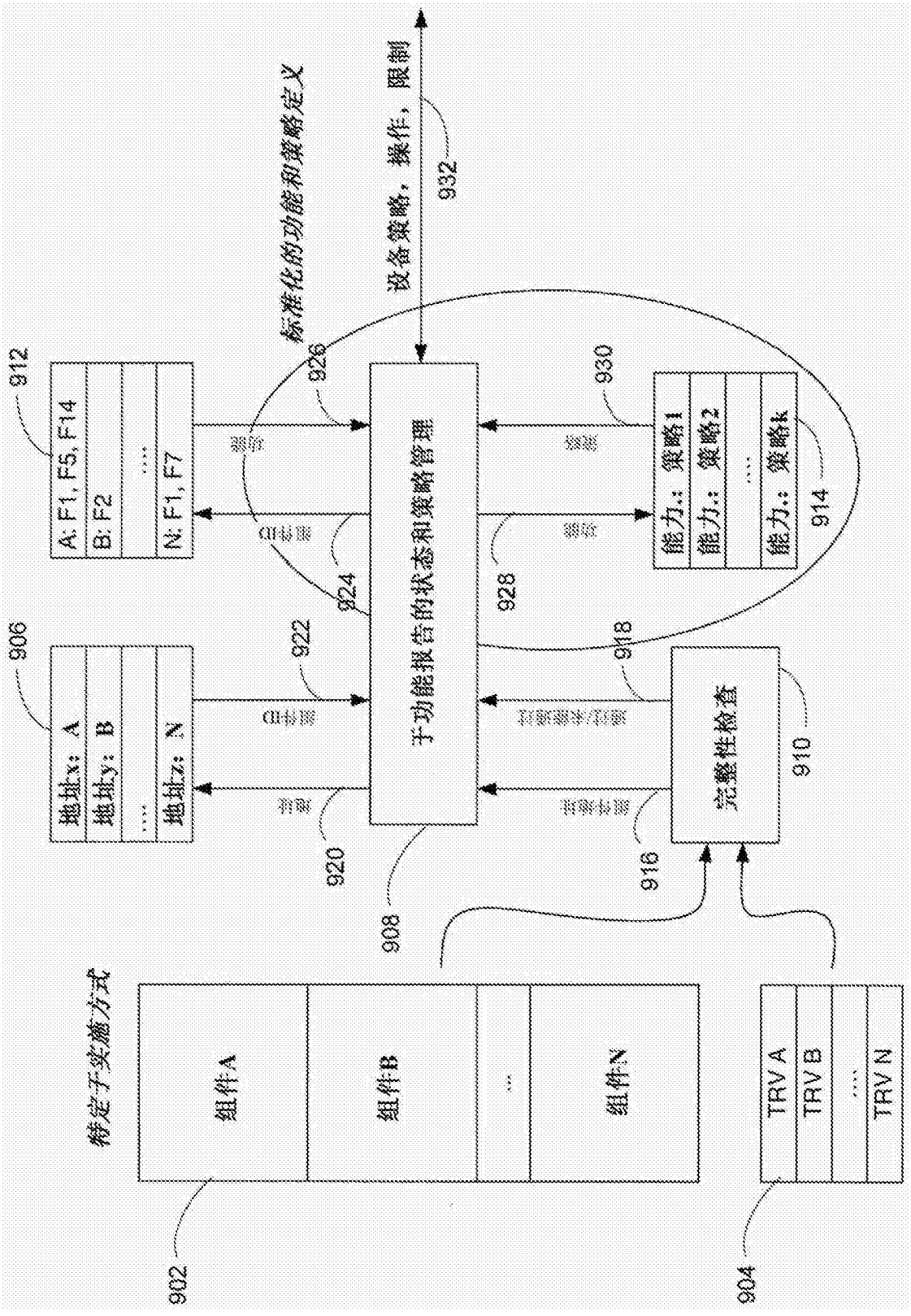

标准化的功能和策略定义
设备策略，操作，限制
932
912
A: F1, F5, F14
B: F2
....
N: F1, F7
926
930
手功能报告的状态和策略管理
能力.: 策略1
能力.: 策略2
....
能力.: 策略k
914
924
928
906
地址x: A
地址y: B
....
地址z: N
922
918
完整性检查
910
920
916
908
特定于实施方式
组件A
组件B
...
组件N
TRV A
TRV B
....
TRV N
902
904

图9

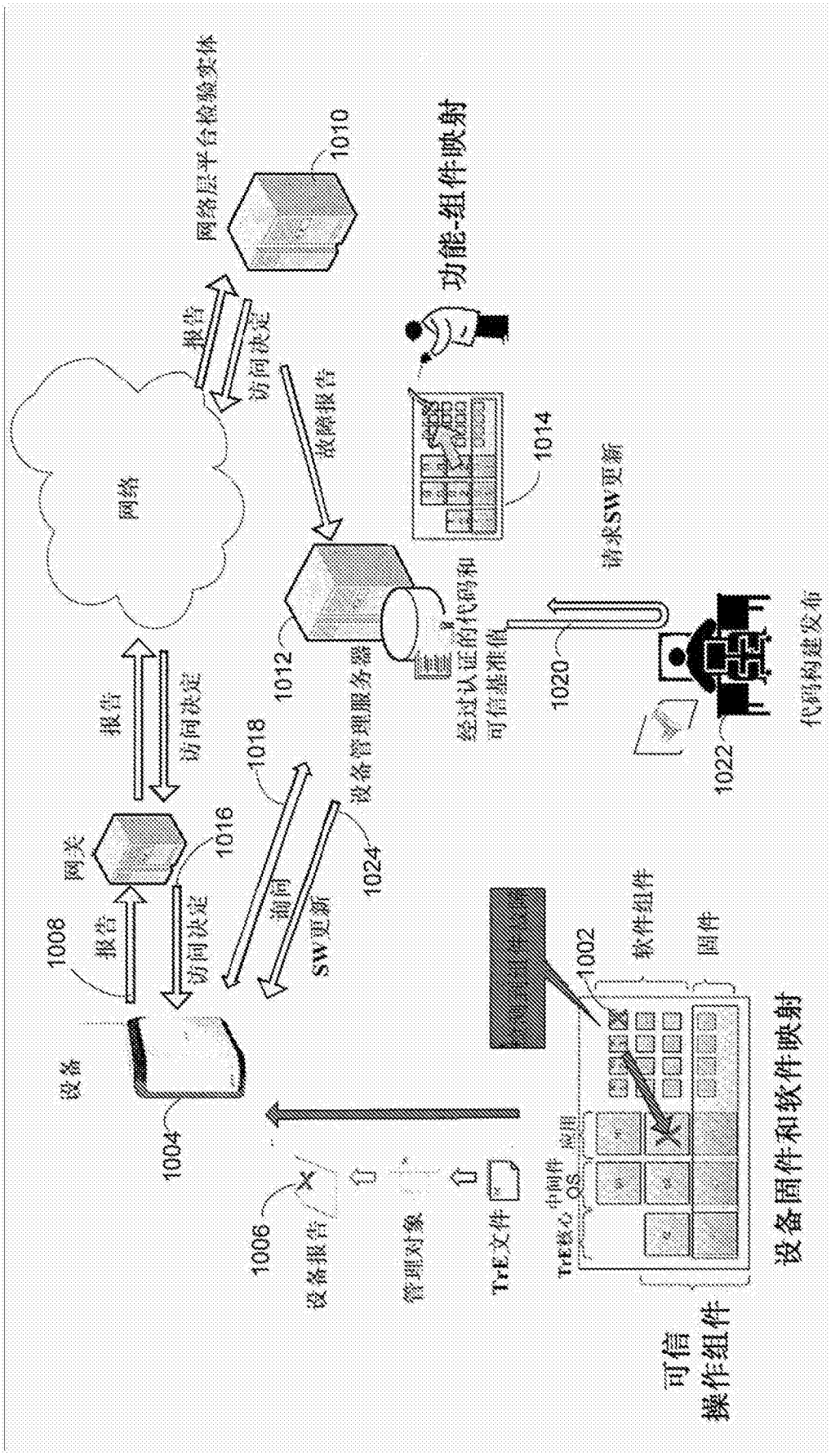

网络层平台检验实体
1010
功能-组件映射
报告
访问决定
故障报告
网络
1014
1012
设备管理服务器
经过认证的代码和
可信基准值
请求SW更新
1020
1022
代码构建发布
1018
趋向
SW更新
1024
网关
1016
1008
报告
访问决定
设备
1004
1002
软件组件
固件
1006
设备报告
管理对象
TrE文件
TrE核心OS
中间件
应用
可信
操作组件
设备固件和软件映射

图10

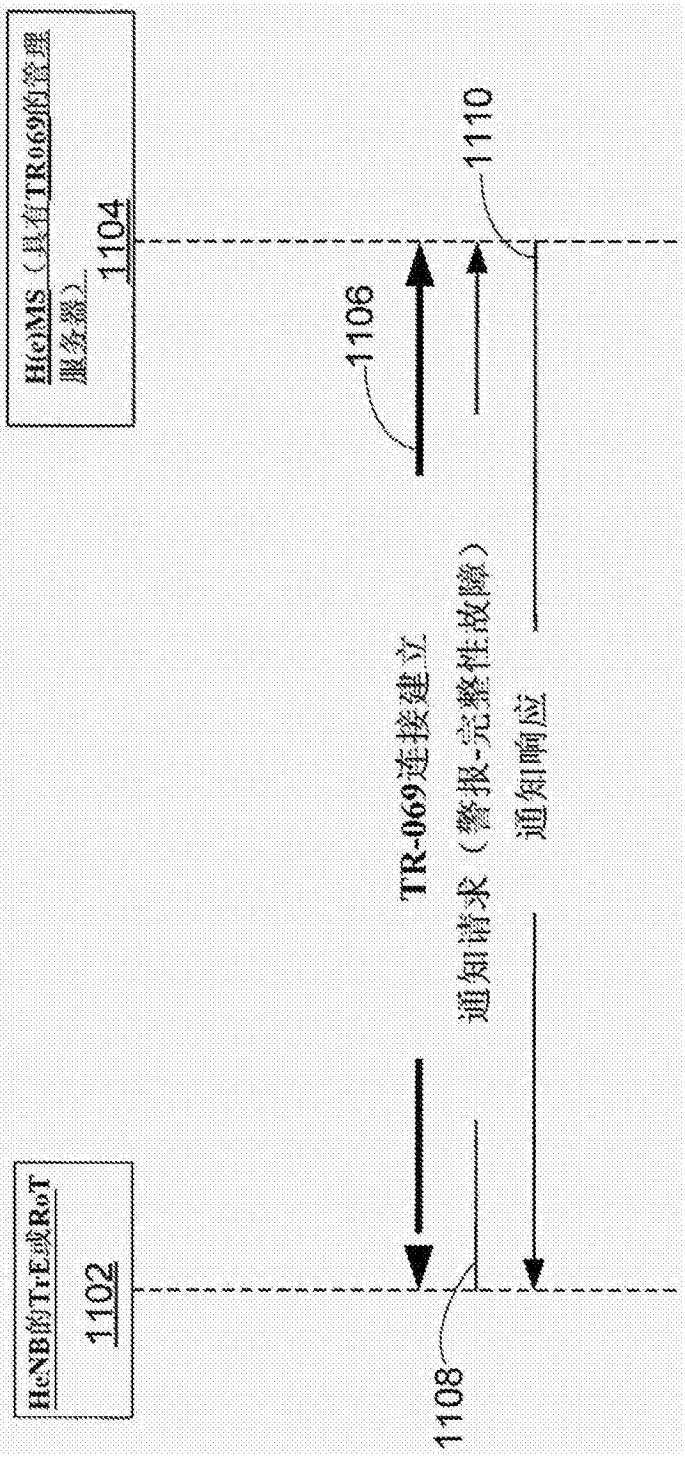

HeNB的TrE或RoT
1102
HeMS（具有TR069的管理服务器）
1104
1106
TR-069连接建立
通知请求（警报-完整性故障）
1108
1110
通知响应

图11

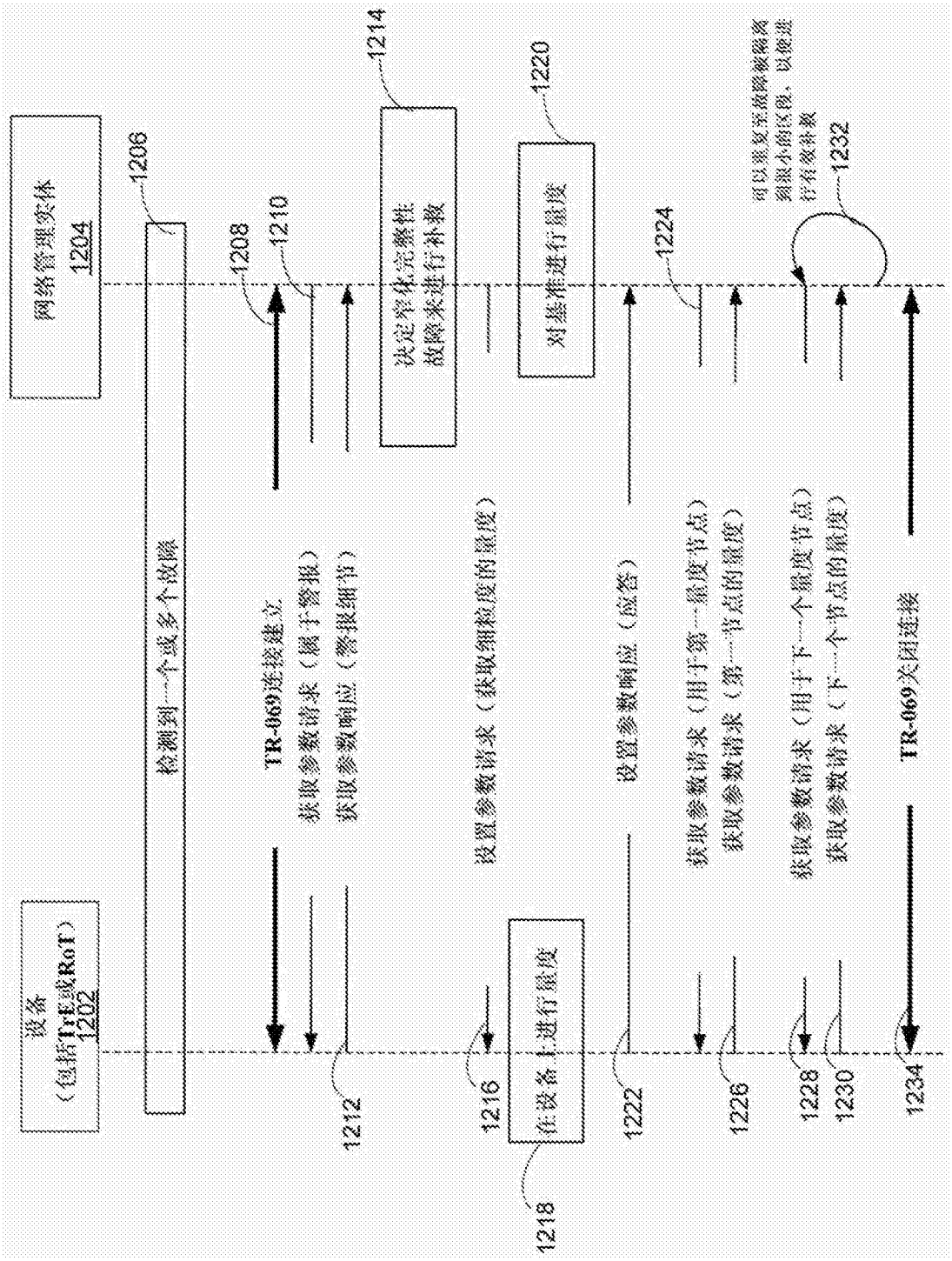

网络管理实体
1204
设备
（包括TrE或RoT）
1202
1206
检测到一个或多个故障
1208
1210
TR-069连接建立
获取参数请求（属于警报）
获取参数响应（警报细节）
1212
1214
决定优化完整性
故障来进行补救
1216
设置参数请求（获取细粒度的量度）
在设备上进行量度
1218
1220
对基准进行量度
1222
设置参数响应（应答）
1224
获取参数请求（用于第一量度节点）
获取参数请求（第一节点的量度）
1226
1228
获取参数请求（用于下一个量度节点）
获取参数请求（下一个节点的量度）
1230
可以重复至故障被隔离到最小的区段，以使进行有效补救
1232
TR-069关闭连接
1234

图12

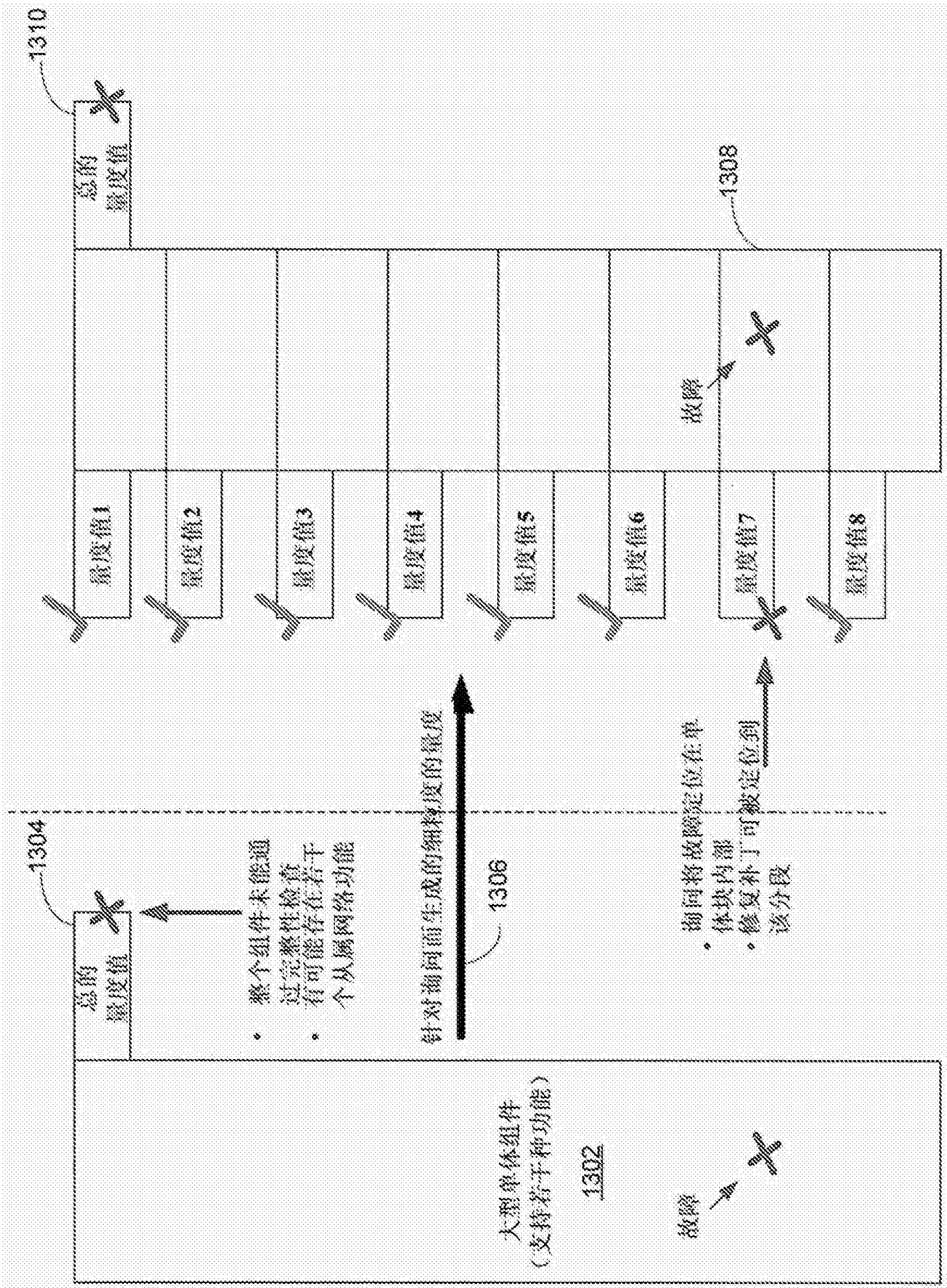

1310
总的量度值
1308
故障
量度值1
量度值2
量度值3
量度值4
量度值5
量度值6
量度值7
量度值8
1304
总的量度值
• 整个组件未能通过完整性检查
• 有可能存在若干个从属网络功能
针对询问而生成的细粒度的量度
1306
• 询问将故障定位在单体块内部
• 修复补丁可被定位到该分段
大型单体组件（支持若干种功能）
1302
故障

图13

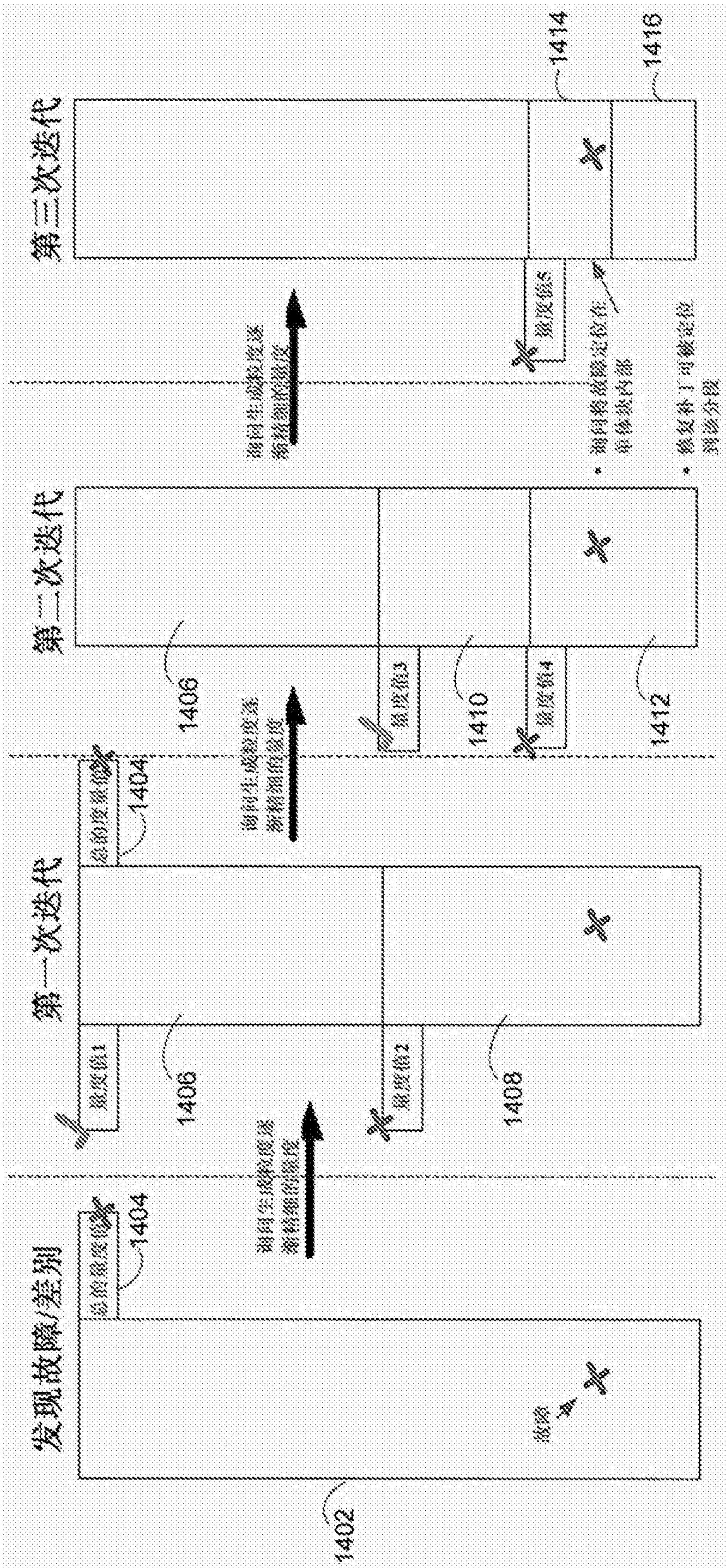

第三次迭代
第二次迭代
第一次迭代
发现故障/差别
1402
1404
1406
1408
1410
1412
1414
1416

图14

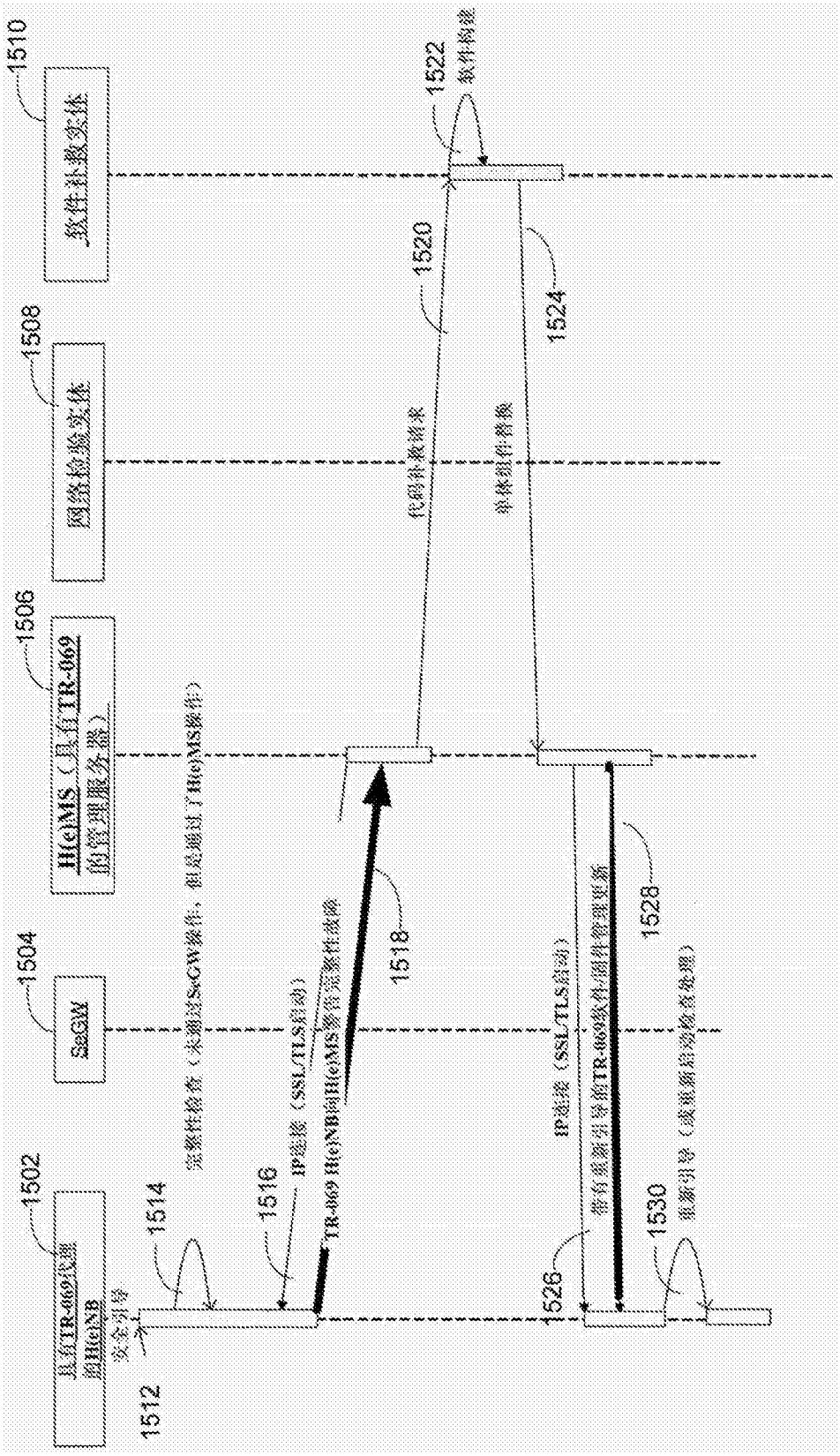

1510
软件补救实体
1522
软件构建
1508
网络检验实体
1520
代码补救请求
1524
单体组件替换
1506
HeNMS（具有TR-069的管理服务器）
1504
SeGW
完整性检查（未通过SeGW操作，但是通过了HeNMS操作）
1518
1502
具有TR-069代理的HeNB
1514
1516
IP连接（SSL/TLS启动）
TR-069 HeNB向HeNMS报告完整性故障
1528
IP连接（SSL/TLS启动）
带有重新导航TR-069软件/组件管理更新
1526
1530
重新引导（或重新启动检查处理）
安全引导
1512

图15

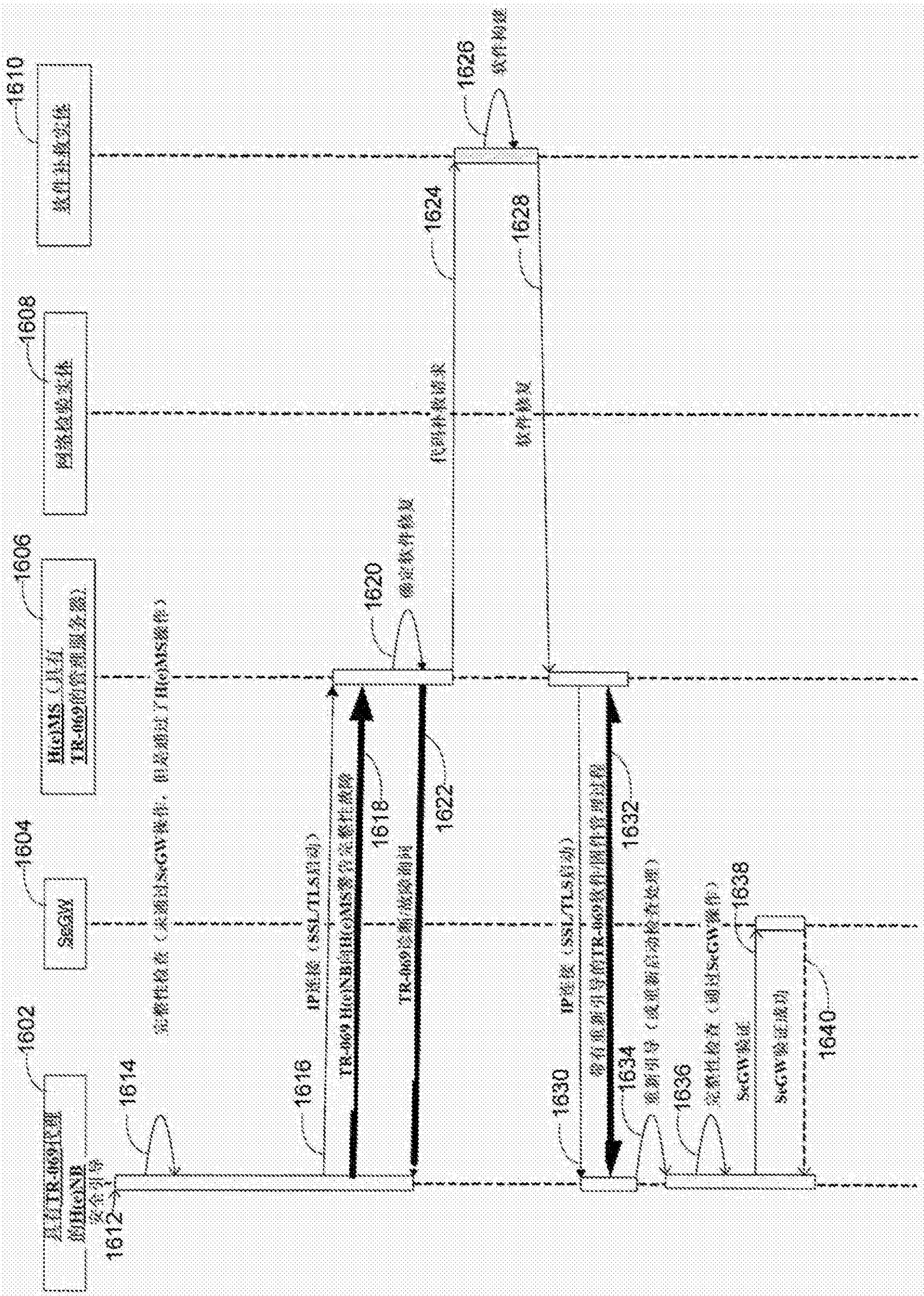

1602
1604
SeGW
1606
1608
1610
1612
1614
1616
1618
1620
1622
1624
1626
1628
1630
1632
1634
1636
1638
1640

图16

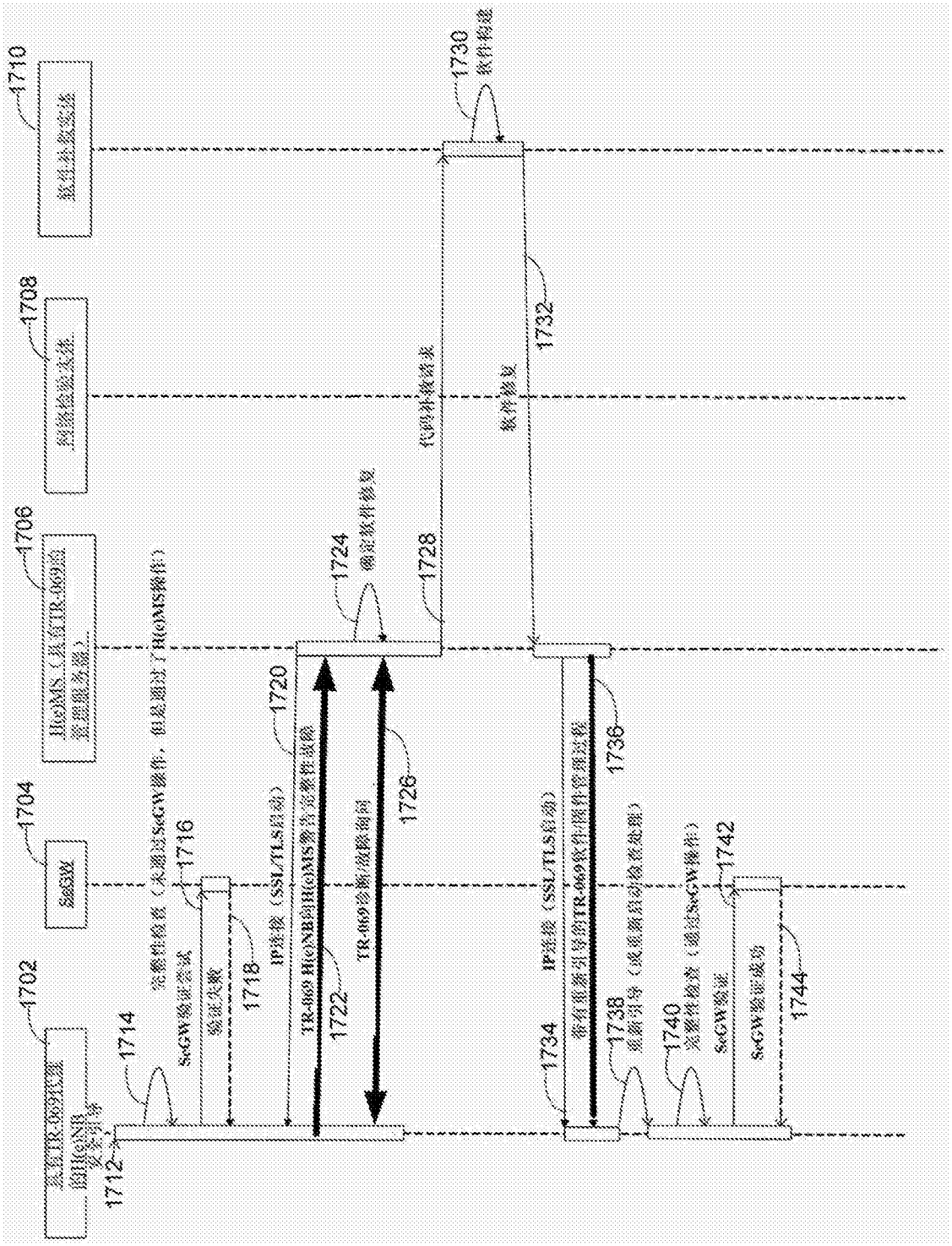

1710
1708
1706
1704
1702
1730
1732
1724
1728
1720
1726
1722
1736
1734
1738
1740
1742
1744
1714
1716
1718
1712

图17

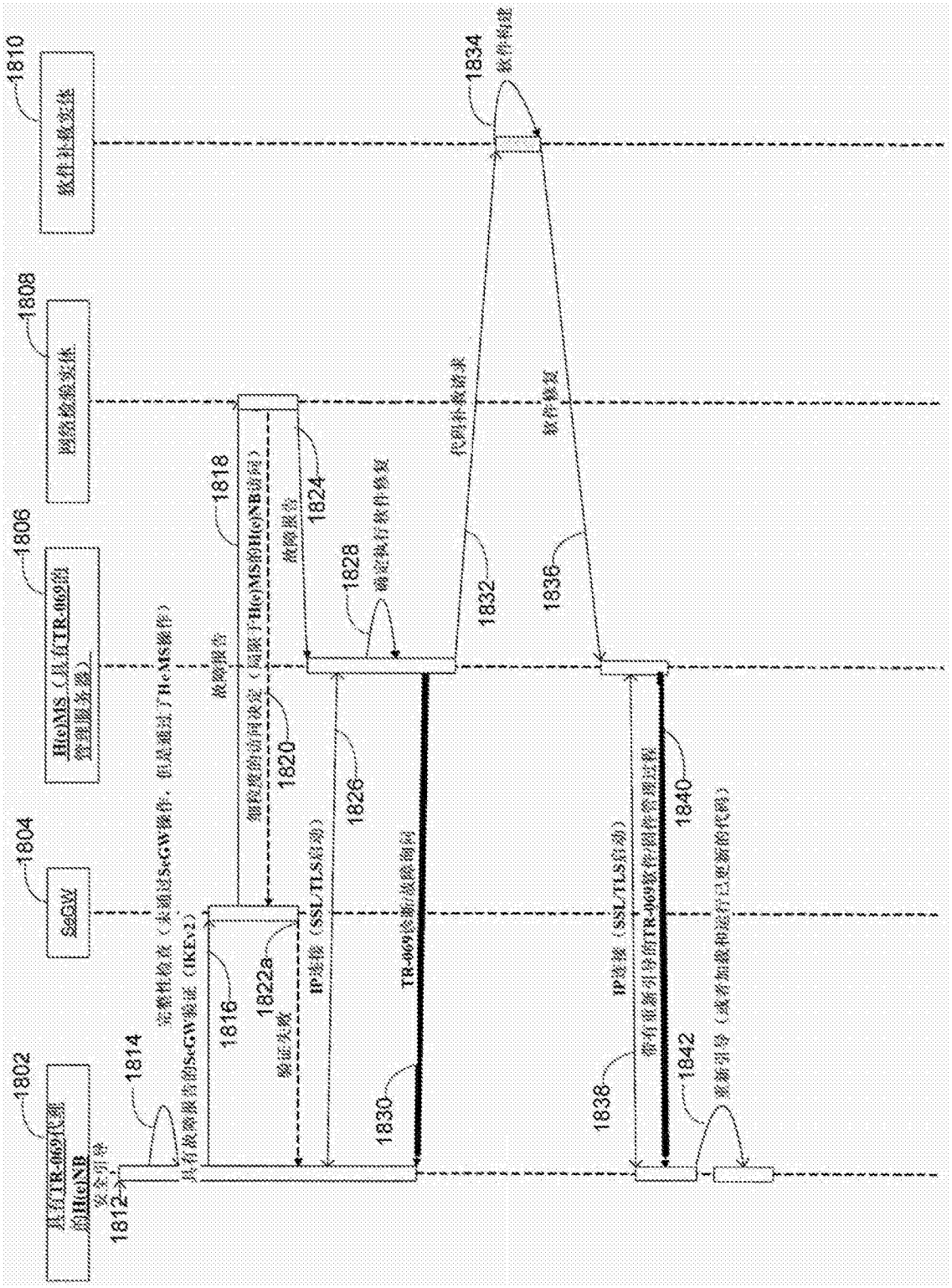

1802
具有TR-069代理的HeNB
1804
SeGW
1806
HeMS（具有TR-069的管理服务器）
1808
网络检测实体
1810
软件库实体
1812
发现引导
1814
具有故障报告的SeGW验证（IKEv2）
完整性检查（未通过SeGW操作，但是通过了HeMS操作）
1816
验证失败
1818
故障报告的请求（源于HeMS的HeNB请求）
1820
故障报告
1822a
IP连接（SSL/TLS启动）
1824
故障报告
1826
TR-069会话/故障报告
1828
确定执行软件修复
1830
1832
代码和数据请求
1834
软件传送
1836
软件修复
1838
IP连接（SSL/TLS启动）
1840
带有报告引导的TR-069会话管理会话
1842
更新引导（或者加载和运行已更新的代码）

图18A

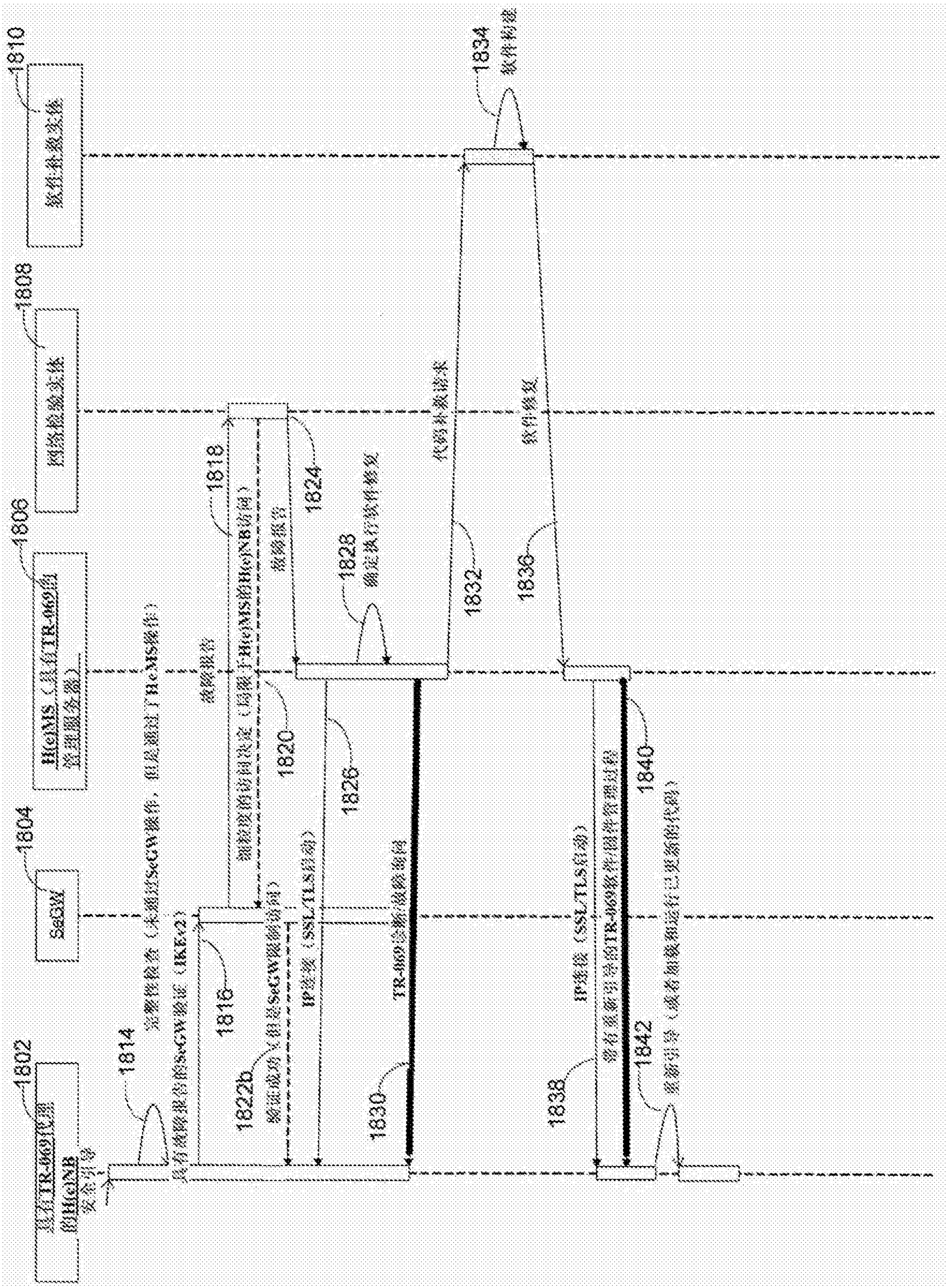

1802
具有TR-069代理的H(e)NB
1804
SeGW
1806
H(e)MS（具有TR-069的管理服务器）
1808
网络检验实体
1810
软件补救实体
安全引导
1814
完整性检查（未通过SeGW操作，但是通过了H(e)MS操作）
具有故障报告的SeGW验证（IKEv2）
1816
1818
细粒度的访问决定（局限于H(e)MS的H(e)NB访问）
故障报告
1820
1822b
验证成功，但是SeGW限制访问
1824
故障报告
IP连接（SSL/TLS启动）
1826
1828
确定执行软件修复
TR-069会话/故障报告
1830
1832
代码补救请求
1834
软件构建
软件修复
1836
IP连接（SSL/TLS启动）
1838
带有重新引导的TR-069软件/固件管理过程
1840
1842
重新引导（或者加载和运行已更新的代码）

图18B

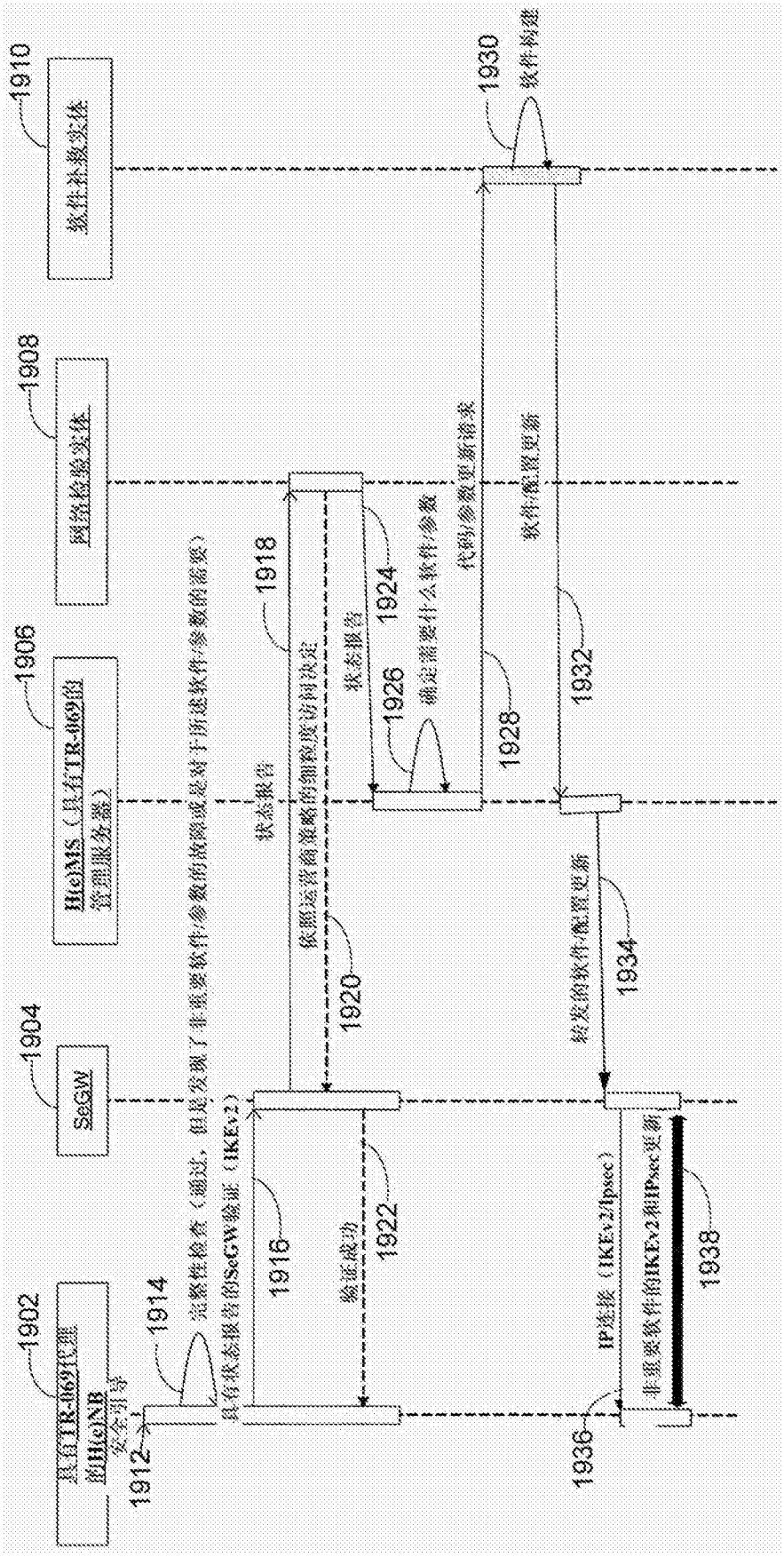

1910
软件决策实体
1930
软件构建
1908
网络检验实体
1906
HeMS（具有TR-069的管理服务器）
1904
SeGW
1902
具有TR-069C端的HeNB
1912
安全引导
1914
完整性检查（通过，但是发现了非重要软件/参数的故障或是对于所述软件/参数的需要）
具有状态报告的SeGW验证（IKEv2）
1916
验证成功
1922
状态报告
1918
依照运营商策略的细粒度访问决定
1920
状态报告
1924
确定需要什么软件/参数
1926
代码/参数更新请求
1928
软件/配置更新
1932
触发的软件/配置更新
1934
IP连接（IKEv2/Ipsec）
1936
非重要软件的IKEv2和IPsec更新
1938

图19

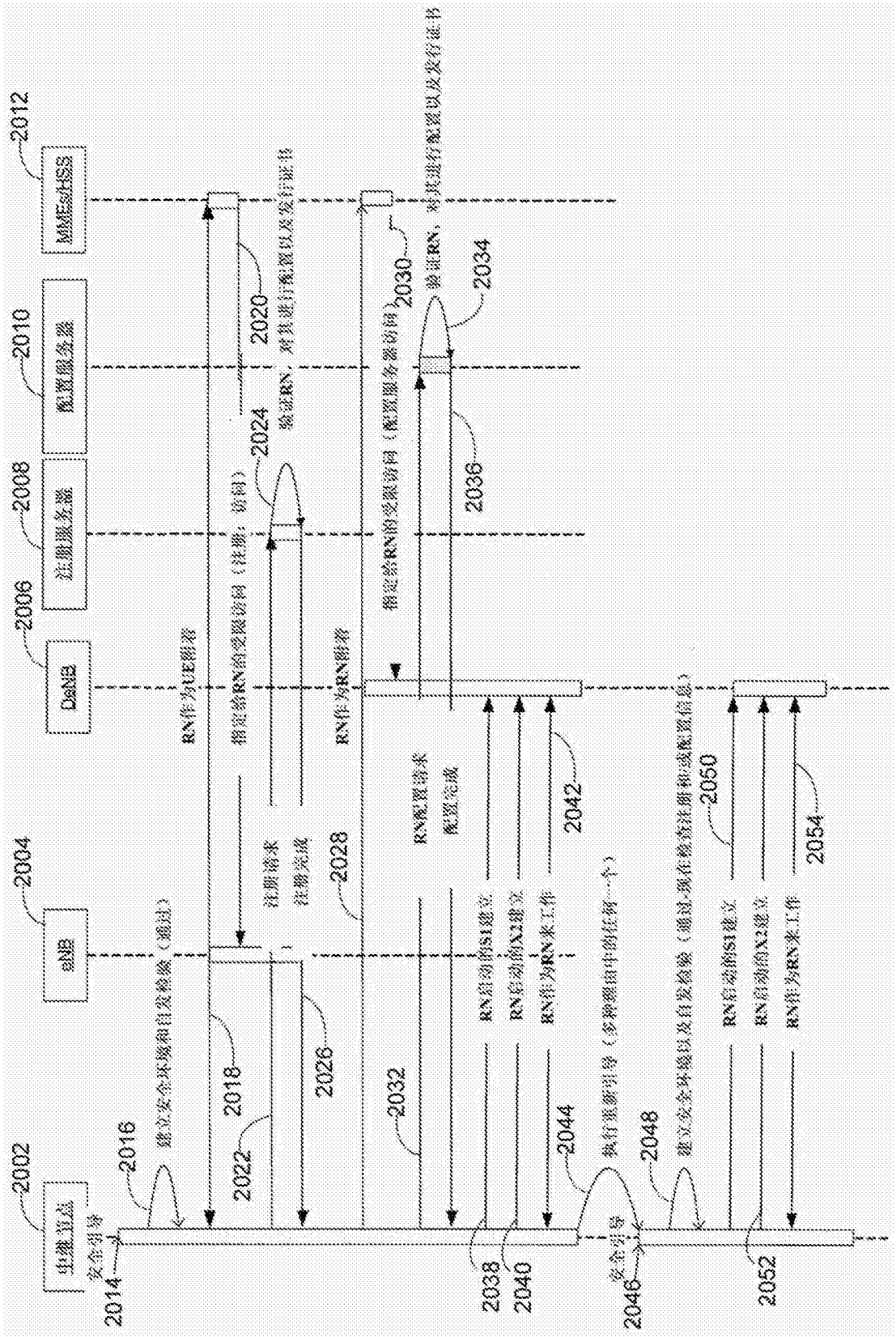

2002
中继节点
2004
eNB
2006
DeNB
2008
注册服务器
2010
配置服务器
2012
MME/HSS
RN作为UE附着
指定给RN的受限访问（注册，前向）
2020
验证RN，并且进行配置以及发行证书
2024
注册请求
注册完成
2028
RN作为RN附着
指定给RN的受限访问（配置服务器访问）
2030
验证RN，并且进行配置以及发行证书
2034
2036
RN配置请求
配置完成
2032
RN启动的S1建立
RN启动的X2建立
RN作为RN来工作
2042
2044
执行重新引导（多种模块中的任何一个）
2048
建立安全环境以及自发检验（确认现在检查注册和或配置信息）
2050
RN启动的S1建立
RN启动的X2建立
RN作为RN来工作
2054
2052
2038
2040
2046
建立安全环境和自发检验（激活）
2016
2018
2022
2026
安全引导
2014

图20

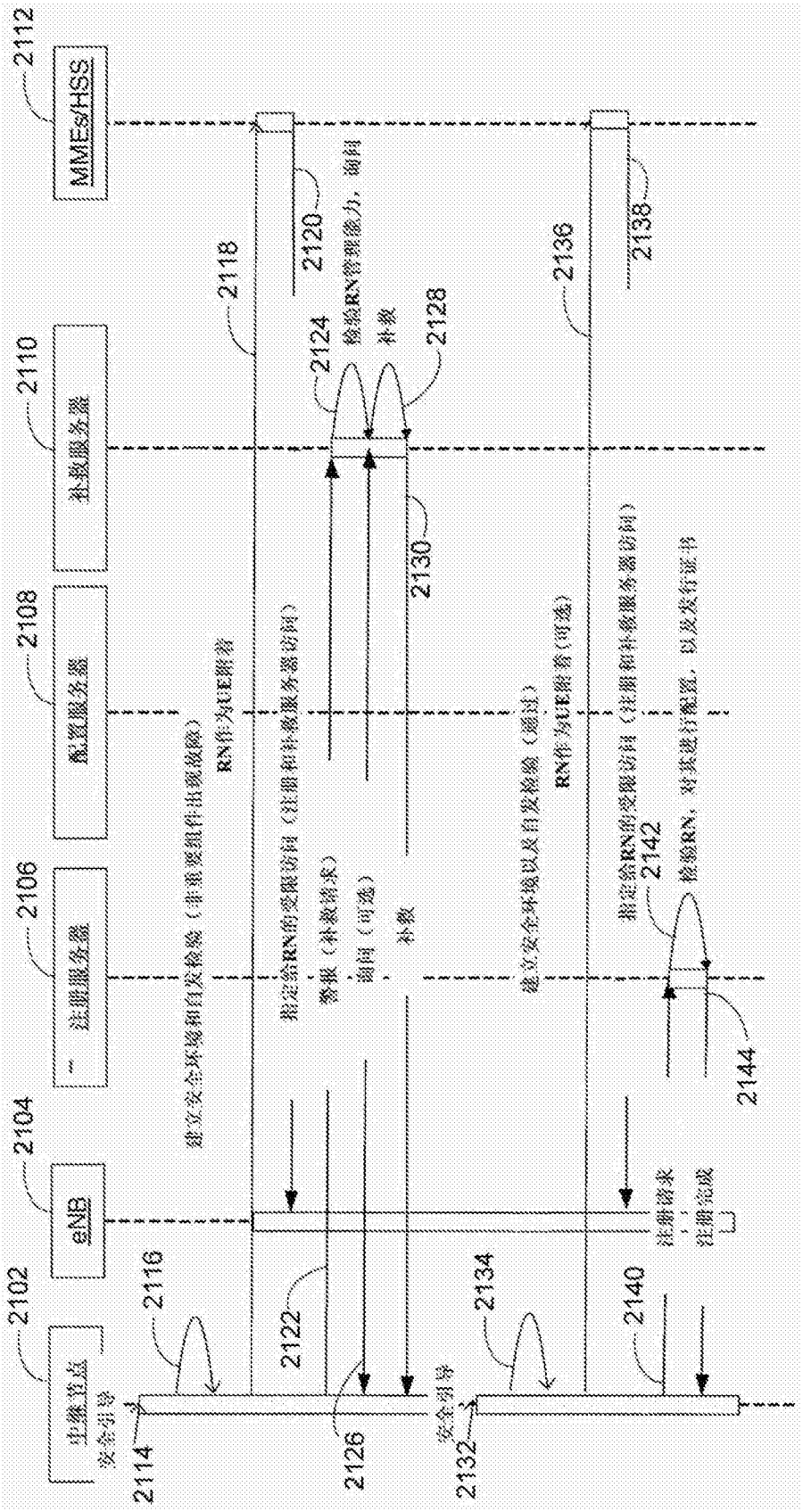

2112
MME/HSS
2110
补救服务器
2108
配置服务器
2106
注册服务器
2104
eNB
2102
中继节点
2118
2120
检验RN管理能力，询问
2124
补救
2128
2130
2136
2138
建立安全环境和自发检验（非重要组件出现检验）
RN作为UE附着
指定给RN的受限访问（注册和补救服务器访问）
警报（补救请求）
询问（可选）
补救
建立安全环境以及自发检验（通过）
RN作为UE附着（可选）
指定给RN的受限访问（注册和补救服务器访问）
2142
检验RN，对其进行配置，以及发行证书
2144
注册请求
注册完成
2116
2122
2126
2134
2140
安全引导
2114
安全引导
2132

图21

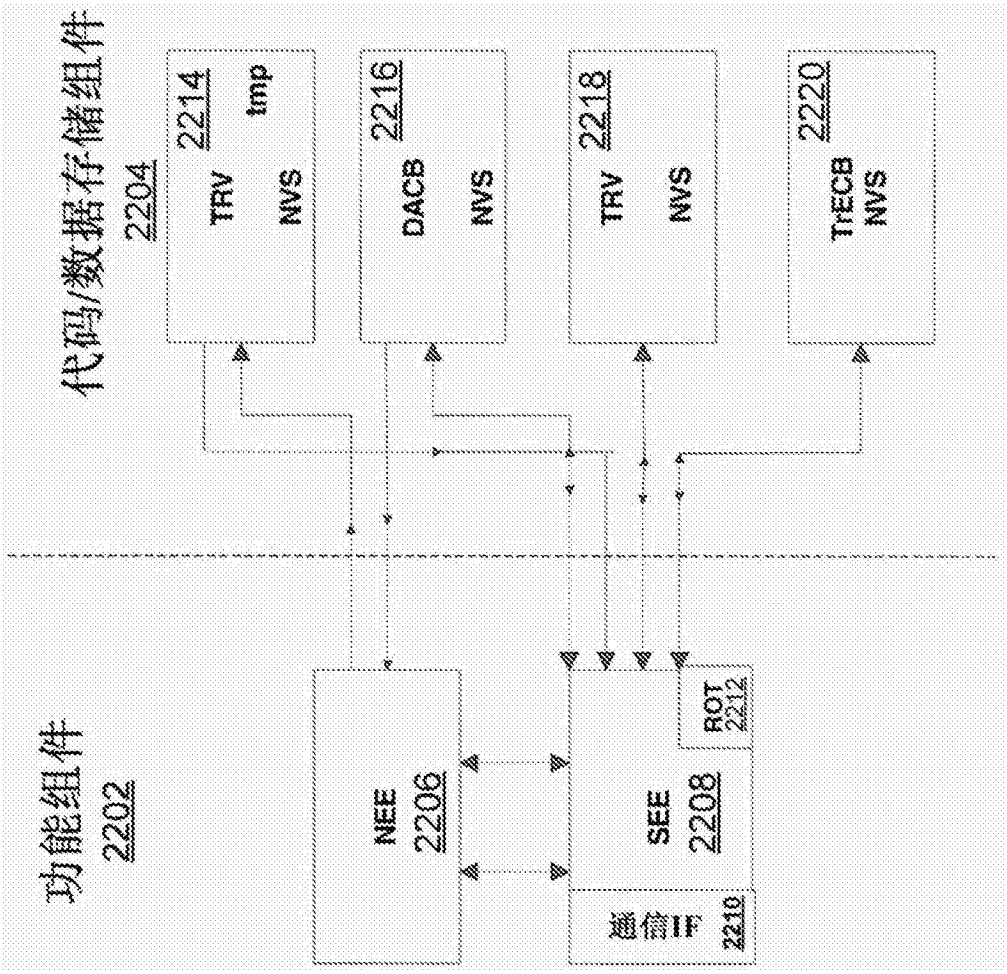

代码/数据存储组件
2204
2214
TRV
tmp
NVS
2216
DACB
NVS
2218
TRV
NVS
2220
TrECB
NVS
功能组件
2202
NEE
2206
SEE
2208
ROT
2212
通信IF
2210

图22

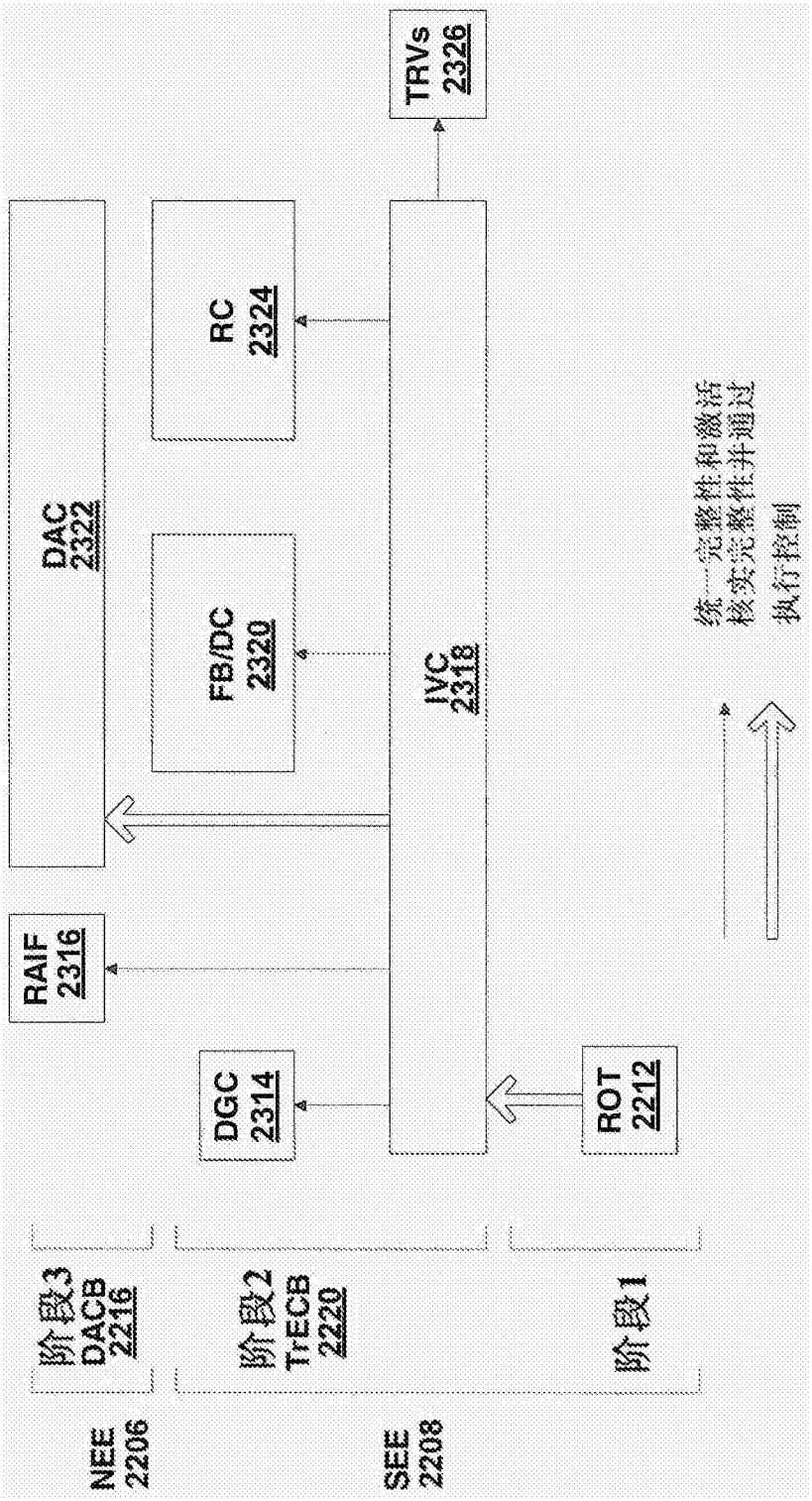

TRVs 2326
RC 2324
DAC 2322
FB/DC 2320
IVC 2318
统一完整性和激活
核实完整性并通过
执行控制
RAIF 2316
DGC 2314
ROT 2212
阶段3 DACB 2216
阶段2 TrECB 2220
阶段1
NEE 2206
SEE 2208

图23

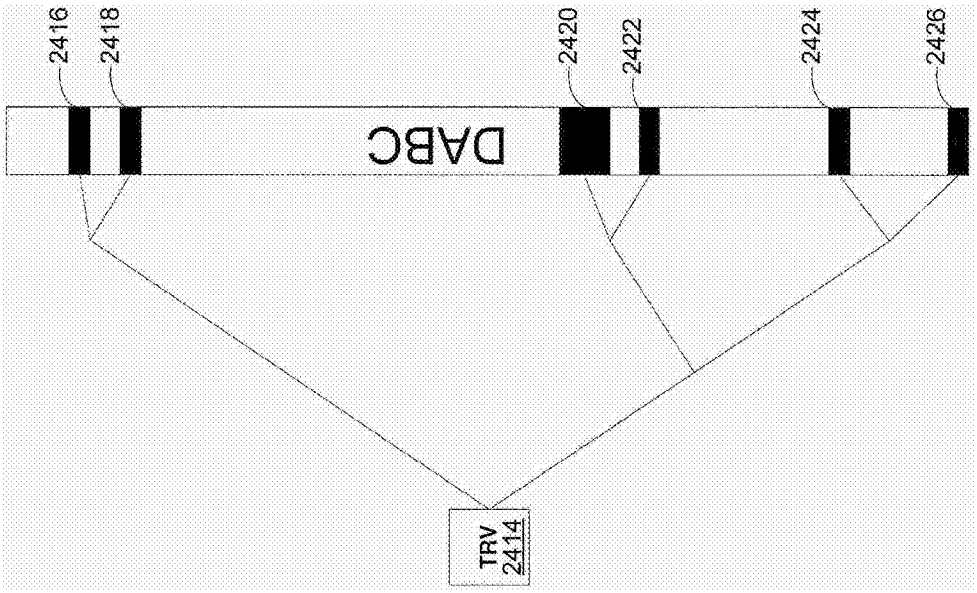

2416
2418
2420
2422
2424
2426
DABC
TRV
2414

图24B

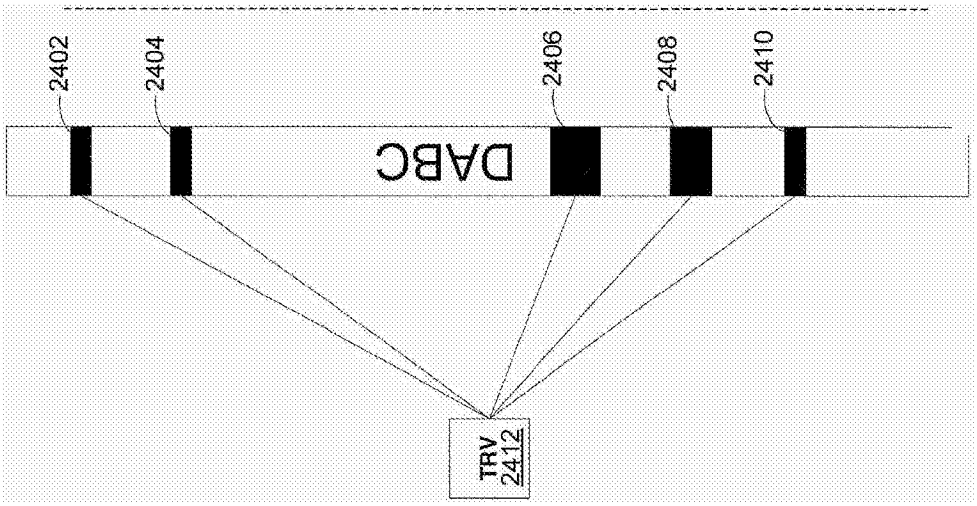

2402
2404
2406
2408
2410
DABC
TRV
2412

图24A