(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 103377337 A
(43)申请公布日 2013.10.30

(21)申请号 201310158165.7

(22)申请日 2013.05.02

(30)优先权数据
1207404.3 2012.04.27 GB

(71)申请人 通用电气航空系统有限公司
地址 英国格洛斯特郡

(72)发明人 C.J. 斯利菲尔德

(74)专利代理机构 中国专利代理(香港)有限公司 72001
代理人 姜甜 朱海煜

(51)Int.Cl.
G06F 21/50(2013.01)

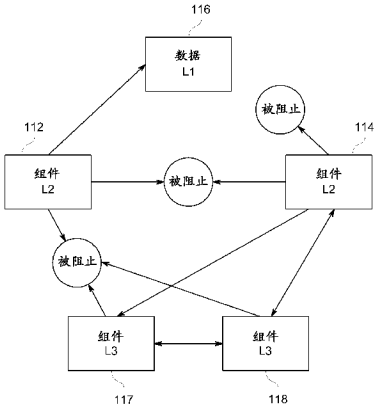
权利要求书2页 说明书9页 附图7页

(54)发明名称

用于控制计算机系统的组件之间的交互的安全性系统和方法

(57)摘要

本发明名称为“用于控制计算机系统的组件之间的交互的安全性系统和方法”。本发明涉及控制计算机系统的一个或多个组件之间的交互，其中对每个组件指配固定的安全性级别，并监视系统的组件之间所有当前活动的和新近请求的交互。本发明存在于一种控制计算机系统的一个或多个组件之间的交互的方法中，该系统包括调适成彼此交互以参与活动的多个组件，该方法包括对系统的每个组件指配固定的安全性级别，并监视系统的组件之间所有当前活动的和新近请求的交互，其中新近请求的交互包括源组件要与目的地组件交互的请求。



CN 103377337 A

1. 一种控制计算机系统(100)的一个或多个组件(110)之间的交互(140)的方法,所述系统(100)包括调适成彼此交互以参与活动(160)的多个组件(110),所述方法包括:

对所述系统的每个组件(110)指配固定的安全性级别;

监视所述系统的组件(110)之间的所有当前活动的和新近请求的交互(140),新近请求的交互包括源组件(112)要与目的地组件(114)交互的请求;

如果所述源(112)组件和目的地(114)组件的安全性级别之间的差超过一个级别,则禁止所述请求的交互;

如果组件(110)参与同具有比其自己低的指配的安全性级别的任何组件(110)的交互(140),且所述请求的交互(140)涉及具有更高指配的安全性级别的源组件或目的地组件(112、114),或所述源组件或目的地组件(112、114)当前参与同具有更高指配的安全性级别的任何组件(110)的交互,则禁止所述请求的交互;

如果组件(112)参与同具有比其自己高的指配的安全性级别的任何组件(110)的交互(140),且所述请求的交互(140)涉及具有更低指配的安全性级别的源组件或目的地组件(112、114),或所述源组件或目的地组件(112、114)当前参与同具有更低指配的安全性级别的任何组件(110)的交互,则禁止所述请求的交互;

如果组件包含具有比指配给所述组件的安全性级别高的安全性级别的数据,且所述请求的交互(140)涉及具有更低指配的安全性级别的源组件或目的地组件(112、114),或所述源组件或目的地组件(112、114)当前参与同具有更低指配的安全性级别的任何组件(110)的交互,则禁止所述请求;

如果组件包含具有比指配给所述组件的安全性级别低的安全性级别的数据,且所述请求的交互(140)涉及具有更高指配的安全性级别的源组件或目的地组件(112、114),或所述源组件或目的地组件(112、114)当前参与同具有更高指配的安全性级别的任何组件(110)的交互,则禁止所述请求;

否则,允许所述请求的交互(140)。

2. 如权利要求 1 所述的方法,其中监视所述系统的组件(110)之间的所有当前活动的和新近请求的交互(140),包括:

确定请求的交互(140)的所述源组件(112)和目的地组件(114)中每个组件的状态值,所述状态值是根据当前参与同所述请求(140)的交互(140)的所述源组件和目的地组件的交互的组件(100)的指配的安全性级别来确定的;

比较所述请求的交互(140)的所述源组件(112)和目的地组件(114)的所述状态值;

当所述请求的交互的所述源组件和目的地组件(112、114)的所述状态值之间存在大于单个的安全性级别的差时,实施状态阻止条件;以及

当状态阻止条件存在时,禁止所述请求的交互。

3. 如权利要求 2 所述的方法,还包括:

对交互(140)期间组件(110)可以参与的每个活动(160)指配优先级级别;

当已经实施状态阻止条件时,隔离已产生所述状态阻止条件的所述源组件和目的地组件(112、114)的现有交互(140);

将同所隔离的交互(140)中涉及的活动(160)关联的优先级级别与同所述源组件和目的地组件(112、114)之间的所述请求的交互(140)中涉及的活动(160)关联的优先级级别

比较；

当所述隔离的交互(140)的活动的优先级级别低于所述源组件与目的地组件(112、114)之间的所述请求的交互的活动的优先级级别时，提升所述状态阻止条件，并且允许所述请求的交互；

否则，保持所述状态阻止条件，并且禁止所述源组件与目的地组件(112、114)之间所述请求的交互。

4. 如权利要求 2 或 3 所述的方法，其中确定请求的交互(140)的所述源组件(112)和目的地组件(114)的状态值的步骤包括：

比较对所述源组件(112)和目的地组件(114)指配的安全性级别；

如果所述源组件(112)的指配的安全性级别低于所述目的地组件(114)的指配的安全性级别，则对所述源组件(112)指配对应于当前参与同所述源组件(112)交互的最低安全组件(110)的安全性级别的状态值，以及对所述目的地组件(114)指配对应于当前参与同所述目的地组件(114)交互(140)的最高安全组件(110)的安全性级别的状态值；

如果所述源组件(112)的指配的安全性级别高于所述目的地组件(114)的指配的安全性级别，则对所述源组件(112)指配对应于当前参与同所述源组件(112)交互的最高安全组件(110)的安全性级别的状态值，以及对所述目的地组件(114)指配对应于当前参与同所述目的地组件(114)交互(140)的最低安全组件(110)的安全性级别的状态值；

否则，对所述源组件(112)和所述目的地组件(114)指配对应于当前参与同相应的源组件(112)和目的地组件(114)交互(140)的最高安全组件(110)的安全性级别的状态值。

5. 一种包括计算机程序代码装置的计算机程序，所述计算机程序代码装置调适成在计算机上运行时执行如权利要求 1 至 4 中任一项所述的方法的步骤。

6. 如权利要求 5 所述的计算机程序，所述计算机程序包含在计算机可读介质上。

7. 一种控制计算机系统(100)的一个或多个组件(110)之间的交互(140)的安全性系统，所述计算机系统(100)包括调适成彼此交互以参与活动的多个组件(110)，所述系统包括：

包含如权利要求 6 所述的计算机程序的安全性模型强制实施机构 SMEM (130)。

8. 如权利要求 7 所述的系统，其中所述安全性模型强制实施机构(130)包含在所述计算机系统的操作系统(120)中。

9. 如权利要求 7 或 8 所述的系统，其中所述安全性模型强制实施机构(130)在安全环境中实现，所述系统(100)的任何组件(110)无法访问或规避所述安全环境。

用于控制计算机系统的组件之间的交互的安全性系统和方法

技术领域

[0001] 本发明涉及计算机系统，具体来说涉及计算机系统内的访问控制。

背景技术

[0002] 随着信息技术的持续迅速扩大，数量不断增长的数据采用数字形式以及实现此类数据的安全是目前大多数企业面对的主要挑战，从而需要通过部署更多安全但是同时部署更多可访问系统以在每个访问点处保护数据。商业公司将它们的Web站点托管在与其他组织的计算机资产联网的服务器上。许多商业和非商业(例如，政府、军队、健康和教育)组织通过网络通信并从与存储和处理敏感数据的系统联网的工作站访问Web。移动装置和关联的应用的广泛采用添加了又一个维度，其中此类装置正在渐增地被用于银行业务和消费交易。单个客户端或服务器的细分为攻击者提供了至整个组织的信息和计算资源的中间连接性，从而损害保密信息和潜在地造成组织的运行中的破坏。数据攻击的数量在过去五年已经增加至三倍多，这促成将安全性与渐增的访问需求(甚至更大优先级)进行平衡的需求。

[0003] 部署安全性模型中的典型要素是保密性、完整性、可达性和数据保证。数据保密性通过将公开限制于仅授权的访问来保证，而数据完整性确保保护数据免于修改，无论是蓄意的还是无意的。数据可达性意味着数据易于访问，而数据保证意味着特定实现提供有关预先建立的安全性目标的保密程度，例如其中保密性是在防御应用中是极为重要的，以及保密性和数据完整性在健康护理和金融应用中同样是重要的。

[0004] 多级安全性模型根据数据的敏感性来使用分级方法。具有不同的安全性分级的数据能够全部驻留在单个域中，并且能够被接收、处理、存储和传播，即便该域内并非所有用户具有访问域内所有数据的安全性许可(security clearance)。最为人熟知的多级安全性模型是Bell-LaPadula和Biba，其中系统包括主体和客体，其中读操作涉及从客体到主体的数据流动，以及写操作涉及从主体到客体的数据流动。Bell-LaPadula模型仅处理数据保密性，其中每个主体和客体具有由表示数据的保护级别的分级或许可(即，机密(SECRET)、分级保密(CLASSIFIED)等)组成的安全性级别。Bell-LaPadula模型强制实施两个属性。

[0005] (i)简单安全性属性：处于给定安全性的级别的主体不得读处于更高安全性级别的客体(不向上读)；以及

(ii)*-属性：处于给定安全性的级别的主体不得对处于更低安全性级别的客体写(不向下写)。

[0006] Biba模型单独处理完整性，完全低忽略保密性并且也强制实施与Bell-LaPadula的两个属性相反的两个属性：

(i)简单完整性属性：处于给定完整性的级别的主体不得读处于更低完整性级别的客体(不向下读)。

[0007] (ii)*完整性属性：处于给定完整性的级别的主体不得对处于更高完整性的级别的客体写(不向上写)。

[0008] 虽然 Bell-LaPadula 和 Biba 安全性模型均尝试处理跨多个安全性的级别的数据流，但是它们均众所周知地为约束性且欠灵活性的。两个模型实际上仅允许一个方向上的数据流，Bell-LaPadula 只允许向下读和向上写(相对于安全性级别)，从而确保数据保密性，以及 Biba 只允许向上读和向下写，从而确保数据完整性。但是，这两个模型均未同时确保数据完整性和保密性。如果严格地实施，则这两个模型存在固有的问题，因为实践中实现其中数据仅一个方向行进的系统是不可能的。

[0009] 发展了“变通方案”，以尝试在实际情况中同时实现两个模型，如允许禁止方向上的受限带宽流。但是，实际上，这是解密分级的形式将始终至少某种程度地损害系统的安全性。此外，这种解密分级往往涉及提高主体或客体的安全性或完整性级别以便尽可能降低风险，这样最终导致大多数主体 / 客体具有顶级安全性或完整性，实际上导致没有安全性或完整性级别划分的系统。为了确保系统的大多数敏感组件和数据的安全性，曾使用 Chinese wall 方法，其涉及围绕这些组件构建巨大的防御性机构，但是同样地，这导致系统的欠灵活性，并且未经济地利用资源。

[0010] 本发明的目的在于提供一种以使得系统或数据的安全性不受侵害的方式控制计算机系统的组件之间的交互的方法。

[0011] 本发明的另一个目的在于提供一种用于以使得系统或数据的安全性不受侵害的方式控制驻留在不同安全性级别上的组件之间的交互并且允许双向的数据流的方法。

发明内容

[0012] 本发明存在于一种控制计算机系统的一个或多个组件之间的交互的方法中，该系统包括调适成彼此交互以参与活动的多个组件，该方法包括对系统的每个组件指配固定的安全性级别，并监视系统的组件之间所有当前活动的和新近请求的交互，其中新近请求的交互包括源组件要与目的地组件交互的请求。首先，访问组件的指配的安全性级别，以及如果源组件和目的地组件的安全性级别之间的差超过一个级别，则禁止所请求的交互。如果存在组件的安全性级别之间的一个级别的差，则评估两个组件的交互。如果组件参与同具有比其自己低的指配的安全性级别的任何组件的交互，且所请求的交互涉及具有更高指配的安全性级别的源组件或目的地组件，或源组件或目的地组件当前参与同具有更高指配的安全性级别的任何组件交互，则禁止所请求的交互。但是，如果组件参与同具有比其自己高的指配的安全性级别的任何组件的交互，且所请求的交互涉及具有更低指配的安全性级别的源组件或目的地组件，或源组件或目的地组件当前参与同具有更低指配的安全性级别的任何组件交互，则禁止所请求的交互。再者，如果组件包含具有比指配给该组件的安全性级别高的安全性级别的数据，且所请求的交互涉及具有更低指配的安全性级别的源组件或目的地组件，或源组件或目的地组件当前参与同具有更低指配的安全性级别的任何组件交互，则禁止所请求的交互。但是，如果组件包含具有比指配给该组件的安全性级别低的安全性级别的数据，且所请求的交互涉及具有更高指配的安全性级别的源组件或目的地组件，或源组件或目的地组件当前参与同具有更高指配的安全性级别的任何组件交互，则禁止所请求的交互。在所有其他情况中，所请求的交互被允许。

[0013] 在监视系统的组件之间所有当前活动的和新近请求的交互时，为所请求的交互的每个源组件和目的地组件确定状态值，该状态值取决于同当前参与所请求的交互的源组件

和目的地组件的交互的组件的指配的安全性级别。

[0014] 比较所请求的交互的源组件和目的地组件的状态值，并在源组件与目的地组件的状态值之间存在多于单个安全性级别的差时，实施状态阻止条件。在状态阻止条件存在时，禁止所请求的交互。

[0015] 对交互期间组件可以参与的每个活动指配优先级级别。当实施了状态阻止条件时，隔离引起状态阻止条件的源组件和目的地组件的现有交互，并将隔离的交互中涉及的活动所关联的优先级级别与涉及源组件与目的地组件之间所请求的交互的活动所关联的优先级级别比较。当隔离的交互的活动的优先级级别低于源组件与目的地组件之间所请求的交互的活动的优先级级别时，提升状态阻止条件，并且允许所请求的交互。否则，保持状态阻止条件，并且源组件与目的地组件之间所请求的交互保持被禁止。

[0016] 本发明还存在于一种包括计算机可读代码部件的计算机程序中以及存在于计算机可读介质上包含的该计算机程序中，该计算机可读代码部件调适成执行上文描述的方法的所有步骤。

[0017] 在另一个方面中，本发明存在于用于控制计算机系统的一个或多个组件之间的交互的安全性系统中，该计算机系统包括调适成彼此交互以参与活动的多个组件，该系统包括包含上文的计算机程序的安全性模型强制实施机构 SMEM。

附图说明

[0018] 现在将仅参考附图描述本发明的实施例，其中：

图 1 图示其中可以实现本发明的计算机系统的系统框图；

图 2 至图 4 图示本发明实现的安全性模型的简单可行实现示例；

图 5 是图示新近请求的交易的两个组件的现有交互的框图；

图 6 是图示确定组件的状态如何以及是否存在状态阻止的流程图；以及

图 7 图示描述的公式 7。

具体实施方式

[0019] 正如本申请中所使用的，术语“组件”是指计算机相关的实体，硬件、硬件与软件的组合、软件或执行中的软件。例如，组件可以是但不限于，处理器上运行的进程、处理器、对象、可执行、执行的线程、程序和 / 或计算机。一个或多个组件可以驻留在进程和 / 或执行的线程内，并且可以将组件局部定位于一个计算机上和 / 或分布在两个或更多计算机之间。虽然本发明是依据软件组件来描述的，但是应该理解，本发明不限于此。

[0020] 参考图 1，示出计算机系统 100，其包括在操作系统 120 的控制下运行的多个组件 110，操作系统 120 包括安全性模型强制实施机构 SMEM（130）。安全性模型强制实施机构 SMEM(130)控制计算机系统 100 的所有组件 110 之间的所有交互 140，并在计算机操作系统 120 的具有与操作系统 120 相同的权限的内核级别上运行，由此能够监视和控制所有交互。交互 140 是从一个组件 110 到另一个组件以交互来执行一个或多个进程或访问的请求，并且可以包括交互期间在组件 110 之间传送的数据。安全性模型强制实施机构 SMEM 130 布置成实现安全性模型 150，安全性模型 150 评估系统的组件 110 之间的所有请求的交互，并基于评估允许或拒绝组件 110 之间所请求的交互。安全性模型强制实施机构 SMEM 130 在

安全环境中实现，系统 100 的任何组件 110 无法访问或规避该安全环境。

[0021] 应该理解，可以采用多种方式强制实施安全性模块 150，具体取决于所采用的体系结构。例如，SMEM 130 可以是具有监视和控制所有交互的权限的独立安全应用。

[0022] 系统 100 的每个组件 110 基于其功能性和 / 或所存储的数据的相对重要性或敏感性被指配安全性级别 q_1…..q_n，其中 q_1 表示最高级别的安全性，以及 q_n 表示最低级别的安全性。本发明的操作系统 120 运行多任务环境，其中每个组件 110 可以与系统 100 的一个或多个其他组件 110 交互以同时执行一个或多个活动或进程 160。为了描述本发明的目的，源组件 112 是请求交互 140 的组件，以及目的地组件 114 是期望与之交互的组件。根据本发明的安全性模型 150，SMEM 130 有关两个组件 110 之间的交互是否允许作出的评估不仅是基于这些组件的指配的安全性级别 q_1、…、q_n 来执行，而且取决于这两个组件 110 中每个组件正在执行的当前活动 160。

[0023] 由 SMEM 130 强制实施的本发明的安全性模型 150 的规则的简化摘要如下：

1. 如果组件 110 参与同更低安全性级别的组件 110 的交互 140，则它无法发起与作为比其本身安全性级别更高的组件或当前参与同比其本身安全性级别更高的组件交互的组件 110 的新交互 140 或接受从其发起的新交互请求。

[0024] 2. 如果组件 110 参与同更高安全性级别的组件 110 的交互 140，则它无法发起与作为比其本身安全性级别更低的组件 110 或当前参与同比其本身安全性级别更低的组件 110 的交互的组件 110 的交互 140 或接受从其发起的交互 140 请求。

[0025] 3. 如果组件 110 包含比其本身安全性级别更高的数据，则它无法发起与作为比其本身安全性级别更低的组件 110 或当前参与同比其本身安全性级别更低的组件 110 的交互 140 的组件 110 的交互 140 或接受从其发起的交互请求。

[0026] 4. 如果组件 110 包含比其本身安全性级别更低的数据，则它无法发起与作为比其本身安全性级别更高的组件 110 或当前参与同比其本身安全性级别更高的组件 110 的交互 140 的组件 110 的交互 140 或接受从其发起的交互请求。

[0027] 图 2 至图 4 图示本发明的安全性模型 150 的规则的三个不同简单可行实现示例。将范围从级别 1 到级别 4（L1-L4）的安全性级别 q 指配给每个组件 110，其中级别 1 表示最大安全，以及级别 4 表示最小安全。参考图 2，具有级别 2 的安全性级别的组件 112 正在从组件 116 访问数据，组件 116 由于其中存储的敏感数据而指配级别 1 的安全性级别，而也具有级别 2 的安全性级别的组件 114 正在与组件 117 和 118 交互，组件 117 和 118 均已指配级别 3 的安全性级别。在此情况中，组件 112 与 114 之间的交互被禁止，因为组件 112 正在与具有更高安全性级别的组件 116 交互，并且组件 114 当前参与同较低安全性级别的组件 117 和 118 交互。组件 114 与组件 116 之间的交互被禁止，因为组件 114 当前正在参与同较低安全性级别的组件 117 和 118 的交互。组件 117 和 118 可以独立于它们与较高安全性级别的组件 114 的交互彼此交互，因为它们具有相同的安全性级别，但是与较低安全性级别的任何组件 110 的通信将被禁止。组件 117 和 118 与组件 116 之间的通信被禁止，因为组件 112 当前正在访问组件 116 中的数据。

[0028] 参考图 3，组件 112 和 114 都正在访问组件 116 中的敏感数据。尽管这两个组件均参与组件 116 的交互，但是因为组件 112 和 114 具有相同的安全性级别，所以为共享数据而在组件 112 与组件 114 之间进行的交互被允许。但是，涉及组件 112 和 114 中任一个与组

件 117 或 118 中任一个的交互交易将被禁止，因为组件 112 和 114 当前正在访问组件 116 中的数据。组件 117 与 118 之间的交互被允许，因为它们具有相同的安全性级别，并且二者均未参与同更高或更低安全性级别的组件 100 的任何交互。

[0029] 在图 4 所示的情况中，组件 112 参与同组件 117 的交互，而组件 114 参与同组件 118 的交易。应用本发明的模型 150 的规则，组件 112 和 114 均被禁止访问组件 116 中的数据，因为二者参与同较低安全性级别的组件 117 和 118 的交互，但是可以彼此请求交互或彼此接受来自对方的交互请求。组件 112 还可以发起与组件 118 的通信或接受来自组件 118 的请求，而组件 114 可以请求与组件 117 的交互或接受来自组件 117 的交互请求。组件 117 与 118 也可以彼此通信，因为它们具有相同的安全性级别，但是，将被禁止组件 117 与 118 与更低安全性级别的组件 110 的任何通信。

[0030] 现在，将更详细地描述 SMEM 130 强制实施的本发明的安全性模型 150。正式地来表述，安全性模型 150 基于如下集合：

（i）系统组件 c：$c \in C(c_1 \ldots c_n)$ 标识系统的每个组件；

（ii）安全性状态 S，这是动态值，其根据组件在特定时间执行的特定活动而定。

[0031] （iii）安全性级别 q：$q = Q(q_1 \ldots\ldots q_n)$，这是基于每个组件的功能性或其中存储的任何数据的相对重要性或敏感性指配给每个组件的固定值，其中 q_1 表示最高安全性评级，以及 q_n 表示最低安全性评级。应该理解，q 的值越高，对其指配的安全性级别越低（即，指配为值 q_1 的组件具有比指配为值 q_3 的组件更高的安全性级别）。例如，在飞行器系统中，出于安全的原因，可以对与飞行器的控制相关的组件指配最高安全性级别 q_1，而监视和提供与飞行器的工作部分的移动相关的数据的飞行器系统内的传感器网络可以具有各指配较低安全性级别 q_3 的传感器节点，因为这些组件不如飞行器的控制那么关键。传感器与控制之间存在功能链接，但是它们被决策功能实体分开，决策功能实体被指配 q_2 的安全性级别。应该理解，虽然组件的安全性级别 q 在系统的运行时间是固定的，但是可以随着系统需求改变按需重新配置指配的安全性级别。

[0032] （iv）优先级级别 p：$p = P(p_1 \ldots\ldots p_n)$，这是对给组件的每个活动指配的固定值，其中 p_1 表示最高优先级评级，以及 p_n 表示最低优先级评级。与安全性级别 q 的情况一样，应该理解，p 的值越高，对其指配的优先级级别越低（即，指配为值 p_1 的活动将具有高于指配为值 p_3 的活动的优先级）。例如，在飞行器系统中，指配优先级级别 p_3 的例行功能性（如驱动作动器）可以被具有较高优先级级别的 p_2 的特殊状况（如传感器检测到的告警阈值）抢先（pre-empt）。

[0033] （v）组件的活动关联 $t \in T(t_1 \ldots\ldots t_l)^c$，其中 $t_l \subseteq c_k \times c_n$ 表示两个组件 c_k 与 c_n 之间的当前交互，以及 t_{l+1} 表示 SMEM 130 要评估的新近请求的交互。

[0034] 系统每个组件 c_k 110 依据其安全性状态 S_{c_k}、对该组件指配的固定的安全性级别 q_{c_k} 和组件的当前活动关联集合 T^{c_k}：

$c_k = S_{c_k} \times q_{c_k} \times T^{c_k}$ 表示为 $c_k = (c_1^k, c_2^k, c_3^k)$ 公式 1

每个活动 t_l 依据如下项来定义：参与该活动的两个组件 110（即，请求交互的源组件 k 112 和被请求与之交互的目的地组件 n 114）以及对活动 t 指配的固定的优先级级别 p_{t_l}：

$t_l = c_k \times c_n \times p_{t_l}$ 表示为 $t_l = (t_1^l, t_2^l, t_3^l)$ 公式 2

现有活动表示为 t_l，其表示涉及组件 k（t_1^l）和组件 n（t_2^l）的当前活动，其中组件 k 是发起

与目的地组件 n 的交互的源组件。新活动表示为(t_{i+1})，其表示涉及组件 k (t_1^{i+1})与组件 n (t_2^{i+1})之间的交互的新近请求的活动，由组件 k 发起且在交互被允许之前要由 SMEM 130 进行评估。

[0035] 状态值 S 是由 SMEM 130 为源 112 和目的地 114 组件就每个新交互请求确定的动态值，并且其反映每个组件的当前活动。所确定的状态值必须考虑到源组件 112 或目的地组件 114 当前活动地关联(即，参与同之交互)的所有组件 110 之间的安全性级别之差。例如，如图 5 所示，组件 k 当前正在与组件 d、e、f 和 g 交互，其中组件 d 已经被指配 q_2 的安全性级别，组件 e 已被指配 q_1 的安全性级别，以及组件 f 和 g 已被指配 q_3 的安全性级别(即，组件 e 是所有交互组件中的最高安全以及组件 f 和 g 是最低安全 $q_f > q_e$)。同时，组件 n 当前正在与组件 h 和 i 交互，其中对组件 i 指配安全性级别 q_2，以及对组件 h 指配安全性级别 q_3 (即，组件 i 比组件 h 更安全)。现在源组件 k 请求涉及组件 k 与组件 n 的又一个交互。

[0036] 首先，必须依据组件 k 和 n 的安全性级别确定请求的合法性。如果这些组件分隔多于一个安全性级别，则通信被禁止，并且无需任何进一步评估。

[0037] $$\varphi\left(\left(c_2^{t_1^{i+1}} \triangleq c_2^{t_2^{i+1}}\right) > 1\right) \rightarrow \psi(t_{i+1} = false) \qquad \text{公式 3}$$

现在将参考图 6 的流程图描述 SMEM 130 应用本发明的安全性模型 150 确定如图 5 所示的组件 k 和 n 的状态值。该过程开始于步骤 200，并且在步骤 202 中，读取组件 k 和 n 的指配的安全性级别 q_k 和 q_n。在步骤 204 中，SMEM 查看组件 k 和 n 中每个组件的所有当前活动关联(即，当前交互)，并读取涉及的组件的安全性级别。在步骤 206 中，SMEM 将组件 k 和 n 的安全性级别 q_k 和 q_n 比较。在步骤 208 中，如果确定组件 k 具有比组件 n 的安全性评级低的安全性评级(即，它更不安全)(即，$q_k > q_n$)，则必须将组件 k 的状态S_{c_k}指配为与它活动地关联的最低安全组件 110 的安全性级别对应的值(即，Q^{max})，同时必须对组件 n 的状态S_{c_n}指配为与它当前关联的最高安全组件 110 对应的值(即，Q^{min})。因此，在步骤 210 中，将为组件 k 分配与组件 f 和 g 的安全性级别对应的状态值S_{c_k}($S_{c_k} = q_{f,g} = q_3$)，并且将为组件 n 分配与组件 i 的安全性级别对应的状态值S_{c_n}($S_{c_n} = q_k = q_2$)。

[0038] 但是，如果在步骤 208 中，确定组件 k 没有比组件 n 的安全性评级更低的安全性级别(即，$q_k > q_n$)，则过程在步骤 212 中继续，其中确定组件 k 是否具有比组件 n 的安全性评级高的安全性评级(即，它更安全)(即，$q_k < q_n$)。如果情况如此，则必须将组件 k 的状态S_{c_k}指配为与它活动地关联的最高安全组件 110 的安全性级别对应的值(即，Q^{min})，同时必须将组件 n 的状态S_{c_n}指配为与它活动地关联的最低安全组件 110 的安全性级别对应的值(即，Q^{max})。因此，在步骤 214 中，将为组件 k 分配与组件 e 的安全性级别对应的状态值S_{c_k}($S_{c_k} = q_e = q_1$)，并且将为组件 n 分配与组件 i 的安全性级别对应的状态值 S ($S_{c_n} = q_k = q_3$)。

[0039] 但是，如果在步骤 212 中，确定组件 k 和 n 已指配相同的安全性级别，则对状态 S 指配与组件 k 或 n 当前活动地关联的最高安全组件 110 的安全性级别对应的值(即，Q^{min})。因此，在步骤 216 中，将为组件 k 分配与组件 e 的安全性级别对应的状态值S_{c_k}($S_{c_k} = q_e = q_1$)，并且将为组件 n 分配与组件 i 的安全性级别对应的状态值S_{c_n}($S_{c_n} = q_i = q_2$)。

[0040] 通过将组件 k 和 n 中每个组件的状态 S 设为与它们当前活动地关联的任何组件 110 中最低或最高安全性级别的组件对应的值，确保了任何时间处所有活动关联之间的最

大安全性许可。这样消除了不同安全性级别的组件之间禁止的交互的风险。

[0041] 正式地来表述，活动地关联的组件的安全性状态按如下公式指配：

$$对于c_k \rightarrow c_n \begin{cases} c_2^k > c_2^n, c_1^k = Q^{\max}(c_3^k) AND c_1^n = Q^{\min}(c_3^n) \\ c_2^k < c_2^n, c_1^k = Q^{\min}(c_3^k) AND c_1^n = Q^{\max}(c_3^n) \\ c_2^k = c_2^n, c_1^k = Q^{\min}(c_3^k) AND c_1^n = Q^{\min}(c_3^n) \end{cases} \quad 公式\ 4$$

如参考图 5 和图 6 描述的，SMEM 130 已确定源组件 112 和目的地组件 114 中每个组件的状态值S_{c_k}和S_{c_n}之后，基于组件 k 和 n 的状态值之差作出有关是否要允许所请求的交互的判断。如果组件 k 和 n 的确定的状态值 S 之差大于 1(即，组件 k 和 n 的最高 / 最低级别的当前关联的活动之间存在多于单个的安全性级别)，则安全性状态阻止发生并且这两个组件之间的交互被禁止。换言之，本发明的安全性模型 150 允许仅高一个安全性级别和仅低一个级别(即，它是可见到的一个级别)的组件之间的交互。因此，如果两个当前交互的任何组件的安全性级别之差超过单个安全性级别，则这两个组件之间所请求的交互将导致状态阻止条件。

[0042] 因此再次参考图 6，在步骤 210 中，其中由于在步骤 208 中组件 k 具有比组件 n 的安全性评级更低的安全性评级(即，$q_k > q_n$)，组件 k 的状态值已指配为$S_{c_k} = q_{f,g} = q_3$，以及组件 n 的状态值指配为$S_{c_n} = q_1 = q_2$，组件 k 和 n 的状态值之间(即，q_3 和 q_2 之间)存在一个安全性级别的差。因此，在步骤 218 中不存在安全性状态阻止情况，以及组件 k 和 n 之间所请求的新交互将被允许。

[0043] 但是，如果如图 6 的步骤 214 那样，由于在步骤 212 中组件 k 具有比组件 n 的安全性评级更高的安全性评级(即，$q_k > q_n$)，组件 k 的状态值已确定为$S_{c_k} = q_e = q_1$，以及组件 n 的状态值确定为$S_{c_n} = q_h = q_3$，组件 k 和 n 的状态值之间(即，q_1 和 q_3 之间)存在两个安全性级别的差。因此，在步骤 220 中存在状态阻止情况，以及组件 k 和 n 之间所请求的新交互将被禁止。对于图 6 的步骤 216，其中由于组件 k 和 n 具有相同安全性评级(即，步骤 208 或 212 均不为真)，组件 k 的状态值已确定为$S_{c_k} = q_e = q_1$，以及组件 n 的状态值确定为$S_{c_n} = q_1 = q_2$，确定组件 k 和 n 的状态值之间(即，q_1 和 q_2 之间)存在单个安全性级别的差。因此，在步骤 222 中，不存在状态阻止情况，并且新交互被允许。

[0044] 这可以正式地表示为：

$$(c_1^k = c_1^n) > 1 \begin{cases} 0, t_{i+1} = true \\ 1, t_{i+1} = false \end{cases} \quad 公式\ 5$$

简言之，在如下三个条件下将发生状态阻止：

(i)源组件 112 的安全性级别(q_{c_k})比目的地组件 114 的安全性级别(q_{c_n})低(即，较不安全)，并且同源组件 112 当前活动地关联的最低安全组件 110 的安全性级别($q(t_1^1)$)与同目的地组件 114 活动地关联的最高安全组件 110 的安全性级别($q(t_2^1)$)之间的差大于 1；

(ii)源组件 112 的安全性级别(q_{c_k})比目的地组件 114 的安全性级别(q_{c_n})高(即，更安全)，并且同源组件 112 活动地关联的最高安全组件 110 的安全性级别($(q(t_1^1))$)与同目的地组件 114 活动地关联的最低安全组件 100 的安全性级别($(q(t_2^1))$)之间的差大于 1。

[0045] (iii)源组件 112 的安全性级别(q_{c_k})等于目的地组件 114 的安全性级别(q_{c_n})，并且同源组件 112 当前活动地关联的最高安全组件 100 的安全性级别($q(t_1^1)$)与同目的地组件

114 当前活动地关联的最高安全组件 100 的安全性级别($q(t_2^i)$)之间的差大于 1。

[0046] 但是，即使状态阻止条件存在时，如果可以发生基于优先级的抢先，则源组件 112 和目的地组件 114 之间的交互可以仍获授权。如早前描述的，对所有活动指配优先级值，以及在状态阻止条件的情况中，如果所请求的交互涉及具有更高优先级值的活动，则可以抢先于更低优先级的活动。为了确定是否可以发生优先级抢先，隔离产生状态阻止的交互 160。

[0047] 对于状态阻止条件，如果产生状态阻止的源组件 112 的现有交互（t_I）具有比源组件 112 请求的组件 112 与 114 之间新近请求的交互（t_{I+1}）的优先级低的优先级（即，$t_3^i > t_3^{i+1}$），则 SMEM 130 确定应该发生优先级抢先，使得现有交互（t_1^i）被中断以及开始涉及组件 112 与 114 之间的交互的较高优先级的请求的新活动（t_{I+1}）。但是，如果现有交互（t_I）没有比新近请求的活动（t_{I+1}）的优先级更低的优先级（即，$t_3^i \le t_3^{i+1}$），则 SMEM 130 确定现有交互继续且新近请求的交互保持被禁止。

[0048] 简言之，对于产生状态阻止的所有活动，如果现有活动的优先级高于或等于所请求的新活动的优先级，则现有活动继续以及请求的交互保持被禁止。但是，如果请求的新活动的优先级大于产生状态阻止的现有活动的优先级，则开始优先级抢先，使得现有活动被中断，以及开始两个组件之间所请求的交互。

[0049] 这在公式 6 和公式 7 中予以正式地表示。

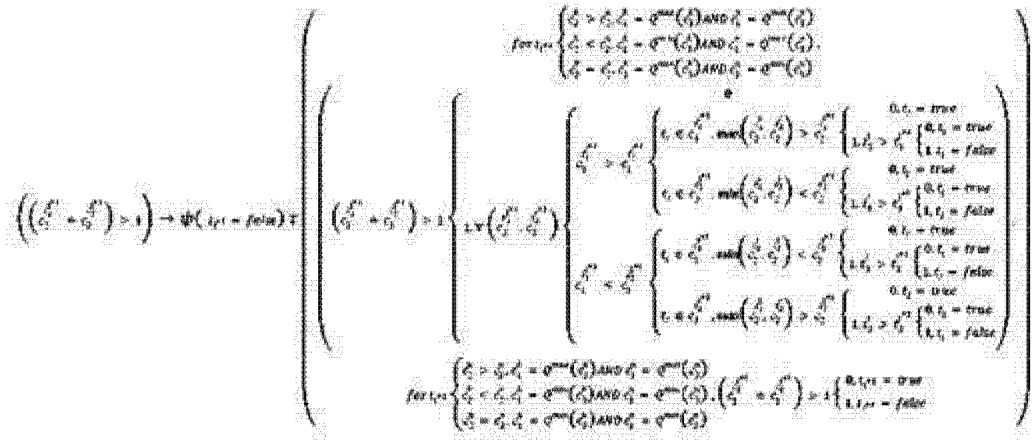
[0050]

a b c d e

$$\forall\left(c_3^{t_1^{i+1}}, c_3^{t_2^{i+1}}\right)\begin{cases} c_1^{t_1^{i+1}} > c_1^{t_2^{i+1}} \begin{cases} t_i \in c_3^{t_1^{i+1}}, \max\left(c_2^{t_1^i}, c_2^{t_2^i}\right) > c_2^{t_1^{i+1}} \begin{cases} 0, t_i = true \\ 1, t_3^i > t_3^{i+1} \begin{cases} 0, t_i = true \\ 1, t_i = false \end{cases} \end{cases} \\ t_i \in c_3^{t_2^{i+1}}, \min\left(c_2^{t_1^i}, c_2^{t_2^i}\right) < c_2^{t_2^{i+1}} \begin{cases} 0, t_i = true \\ 1, t_3^i > t_3^{i+1} \begin{cases} 0, t_i = true \\ 1, t_i = false \end{cases} \end{cases} \end{cases} \\ c_1^{t_1^{i+1}} < c_1^{t_2^{i+1}} \begin{cases} t_i \in c_3^{t_1^{i+1}}, \min\left(c_2^{t_1^i}, c_2^{t_2^i}\right) < c_2^{t_1^{i+1}} \begin{cases} 0, t_i = true \\ 1, t_3^i > t_3^{i+1} \begin{cases} 0, t_i = true \\ 1, t_i = false \end{cases} \end{cases} \\ t_i \in c_3^{t_2^{i+1}}, \max\left(c_2^{t_1^i}, c_2^{t_2^i}\right) > c_2^{t_2^{i+1}} \begin{cases} 0, t_i = true \\ 1, t_3^i > t_3^{i+1} \begin{cases} 0, t_i = true \\ 1, t_i = false \end{cases} \end{cases} \end{cases} \end{cases}$$

公式 6

或更详细地



公式 7

图 7 中也示出公式 7。

[0051] 简言之，对于提出新关联（$c_3^{t_1^{tH}}$）的源组件 112 的所有活动关联（即，与其他组件的所有当前交互）和提出新关联（$c_3^{t_2^{tH}}$）的目的地组件 114 的所有活动关联，首先需要确定提出的新关联（$c_3^{t_1^{tH}}$）的源组件 112 的状态值是否低于提出的新关联（$c_3^{t_2^{tH}}$）的目的地组件 114 的状态值或反之。重要的是注意因为已经确定源组件 112 和目的地组件 114 的安全性级别之间的差，所以没有状态值相等的选项。接下来，确定提出的新关联是活动关联集合中属于源组件 112 的成员（$t_1 \in c_3^{t_1^{tH}}$）还是属于目的地组件的成员（$t_1 \in c_3^{t_2^{tH}}$）。然后，必须评估现有关联（t_1）的组件 110 和 114 的安全性级别的差是否会导致状态阻止。例如，如果$t_1 \in c_3^{t_1^{tH}}$且$c_1^{t_1^{tH}} > c_1^{t_2^{tH}}$，则其中组件 110 之一具有大于$c_2^{t_1^{tH}}$的安全性级别情况下，任何 t_1 将产生状态阻止。如果不存在状态阻止，则现有关联（t_1）可以继续。如果存在状态阻止，则执行优先级评估。如果现有交互（t_1）具有高于或等于新近请求的关联的优先级，则它可以继续，否则现有交互必须终止，并且新近请求的交互被允许。

[0052] 上文概述的四个步骤被组合成单个评估，无论何时，只要源组件 110 请求与目的地组件 114 的活动，促成是否允许形成新关联的判断（即，t_{I+1}=*true*），则执行该评估。

[0053] （批注$\varphi(a) \rightarrow \psi(b) \tau(c)$读为如果 (a) 则 (b) 否则 (c)）

虽然本发明是结合目前视为最实用且优选的实施例来描述的，但是要理解本发明并不局限于所披露的实施例，并且在不背离基本教导的前提下可以使用多个变化。

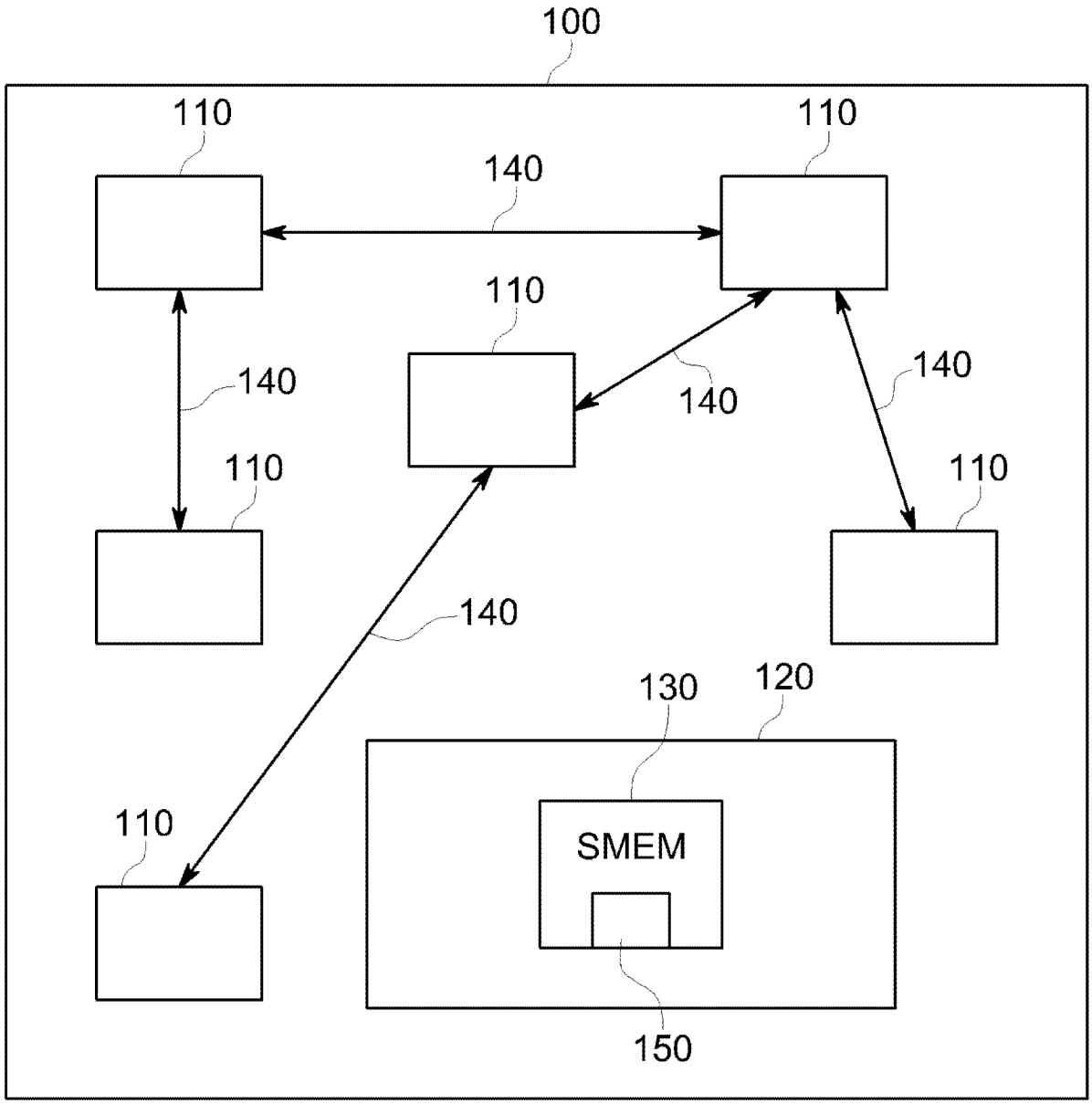


图 1

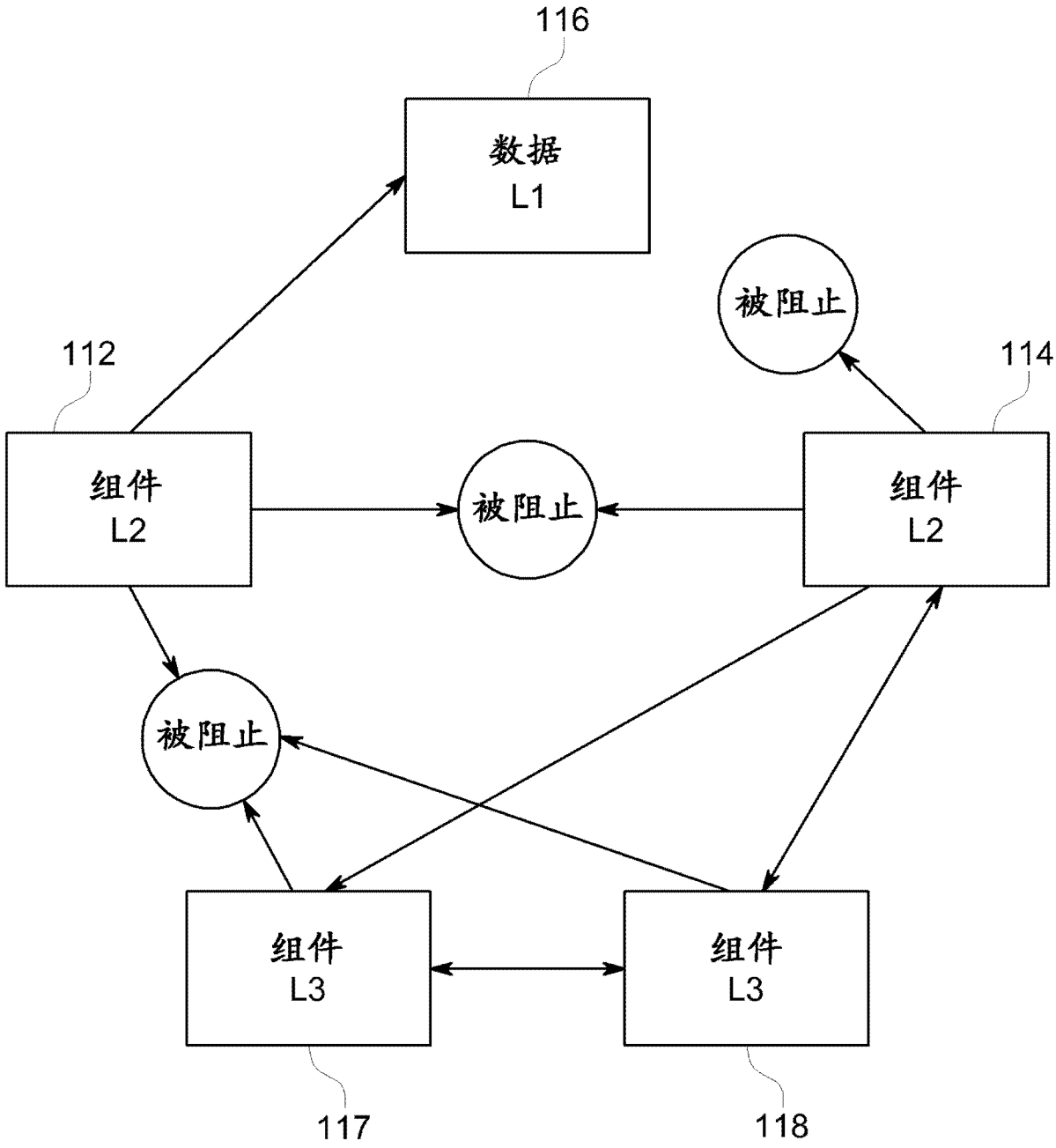


图 2

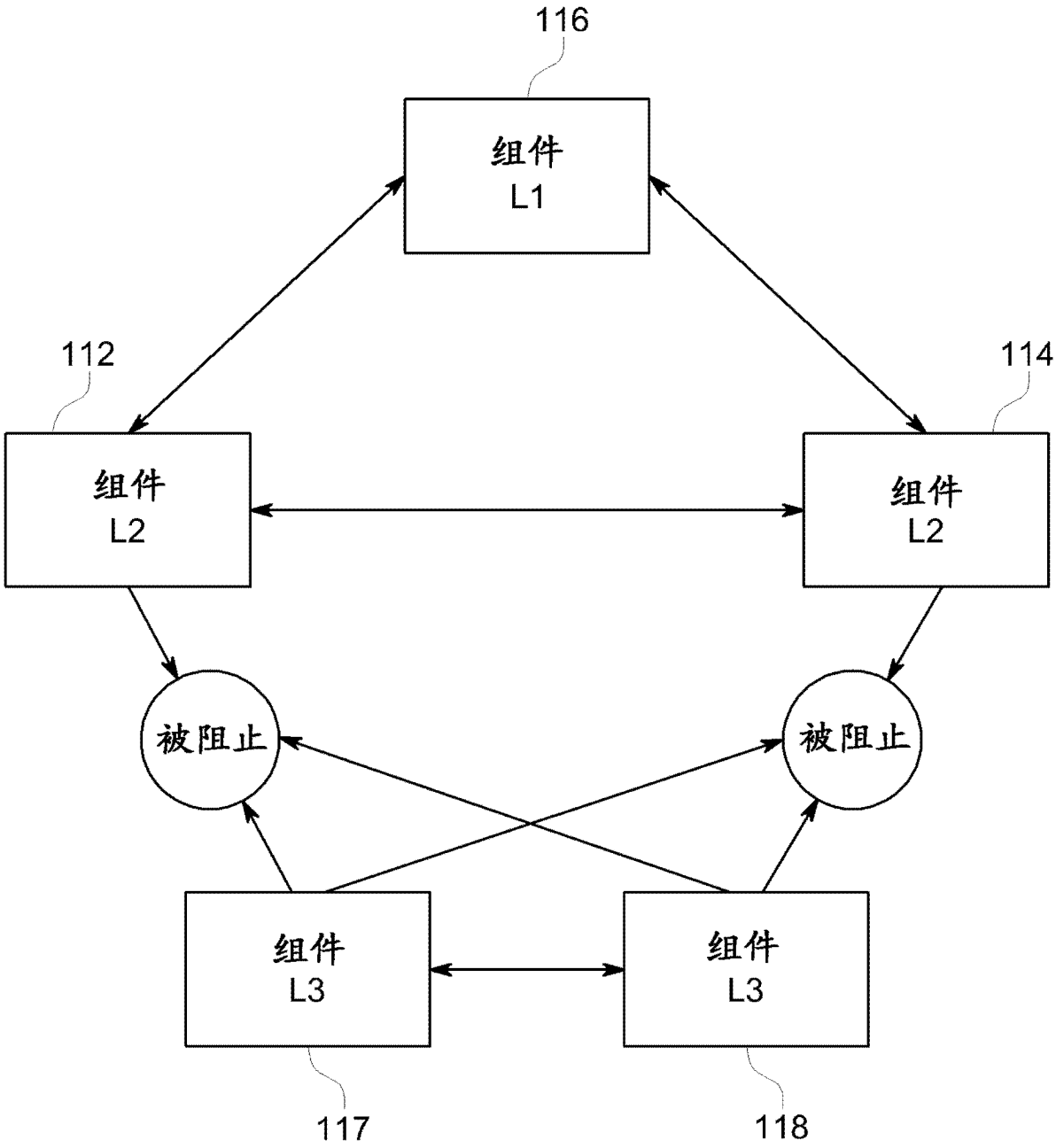


图 3

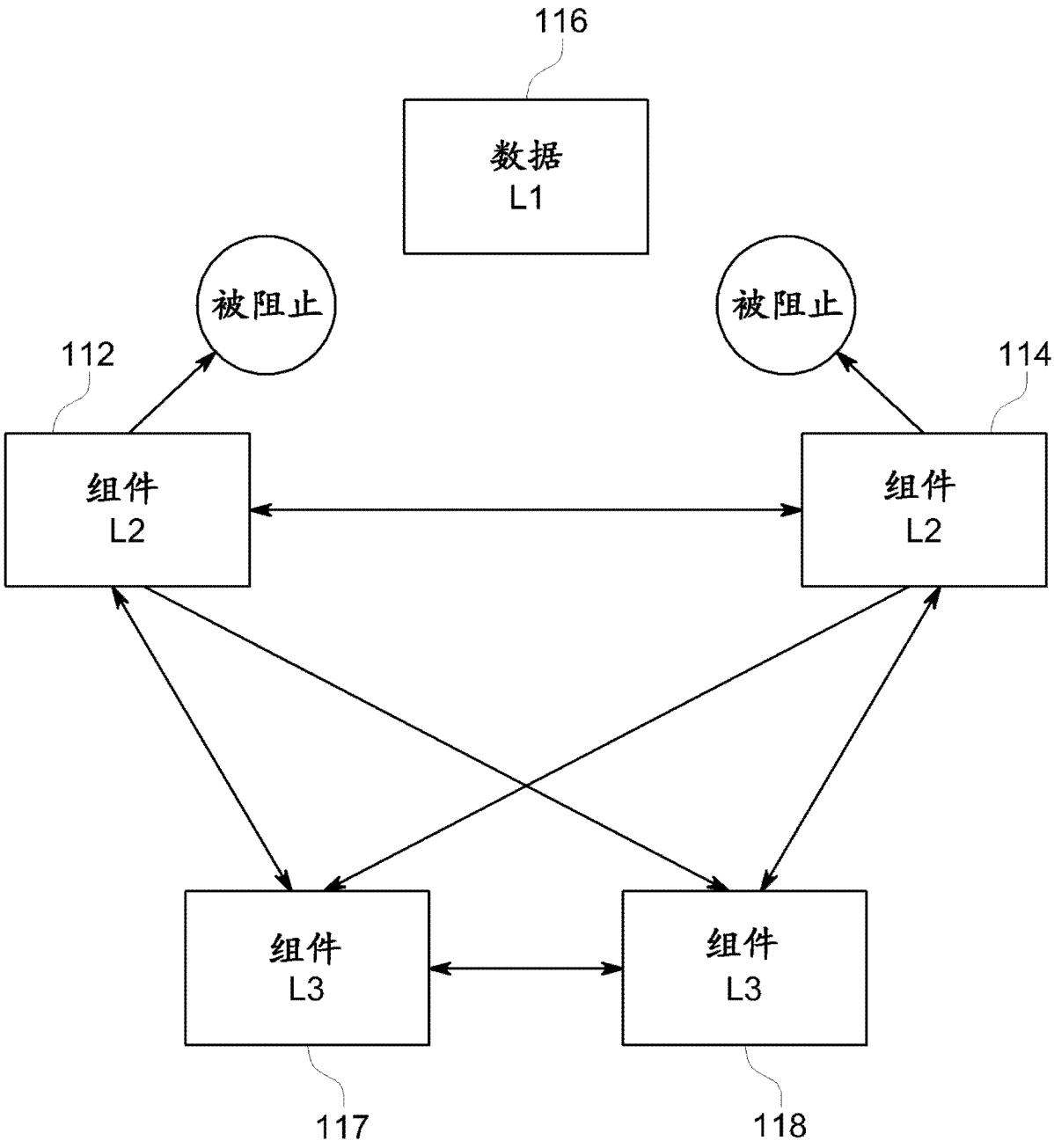


图 4

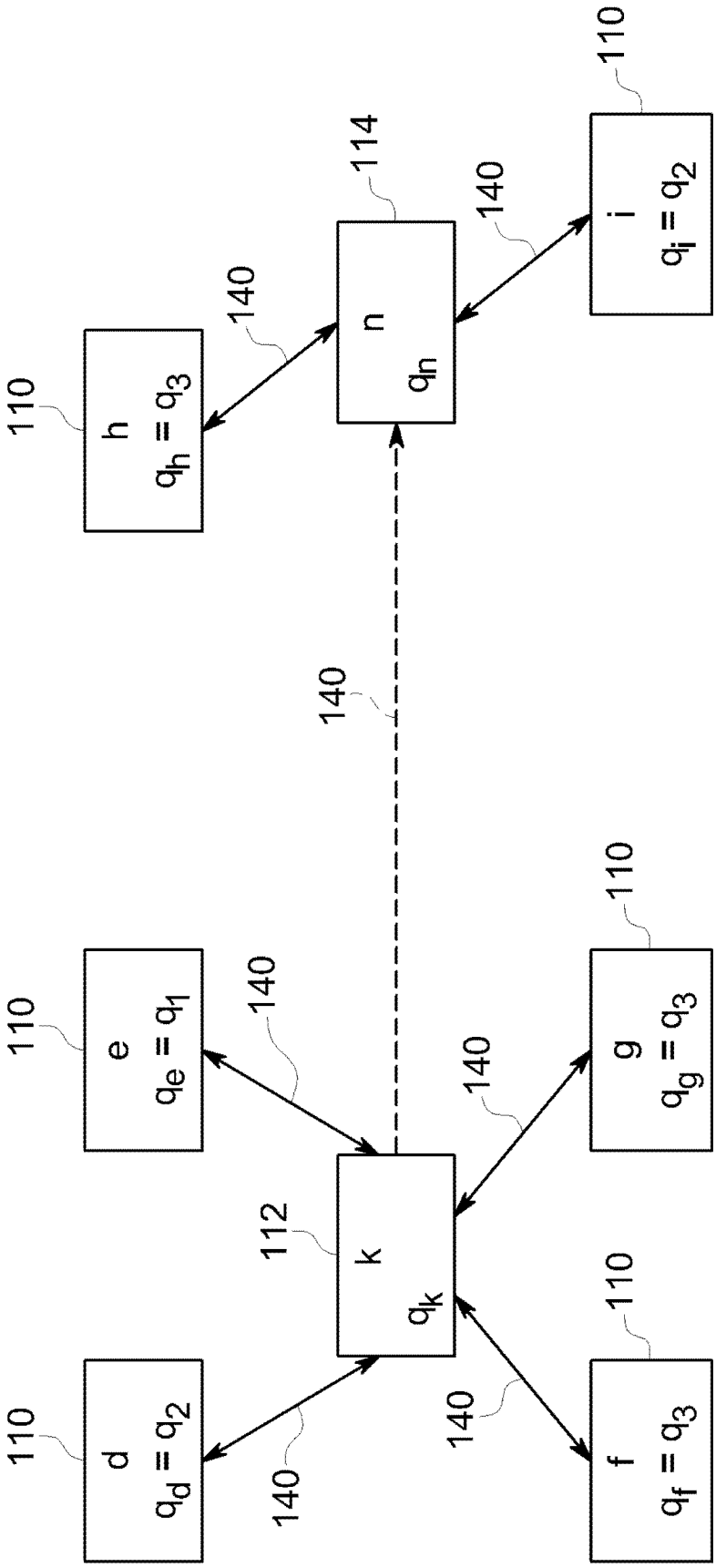

110
i
$q_i = q_2$
114
140
n
q_n
140
110
h
$q_h = q_3$
140
110
e
$q_e = q_1$
140
110
g
$q_g = q_3$
140
112
k
q_k
110
d
$q_d = q_2$
140
140
110
f
$q_f = q_3$

图 5

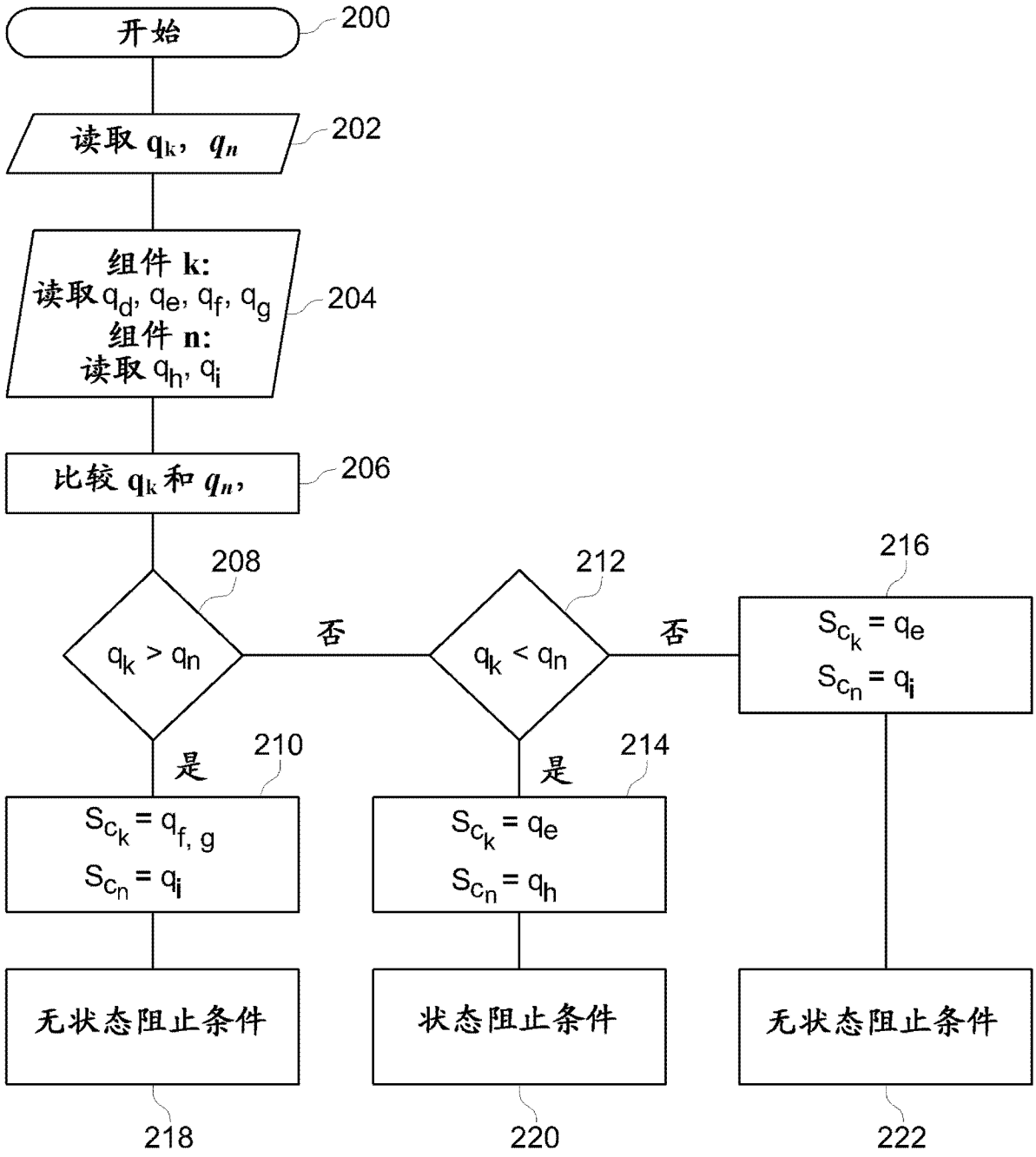

开始
200
读取 q_k, q_n
202
组件 k:
读取 q_d, q_e, q_f, q_g
组件 n:
读取 q_h, q_i
204
比较 q_k 和 q_n,
206
208
$q_k > q_n$
否
212
$q_k < q_n$
否
216
$S_{c_k} = q_e$
$S_{c_n} = q_i$
是
210
$S_{c_k} = q_{f,g}$
$S_{c_n} = q_i$
是
214
$S_{c_k} = q_e$
$S_{c_n} = q_h$
无状态阻止条件
状态阻止条件
无状态阻止条件
218
220
222

图 6

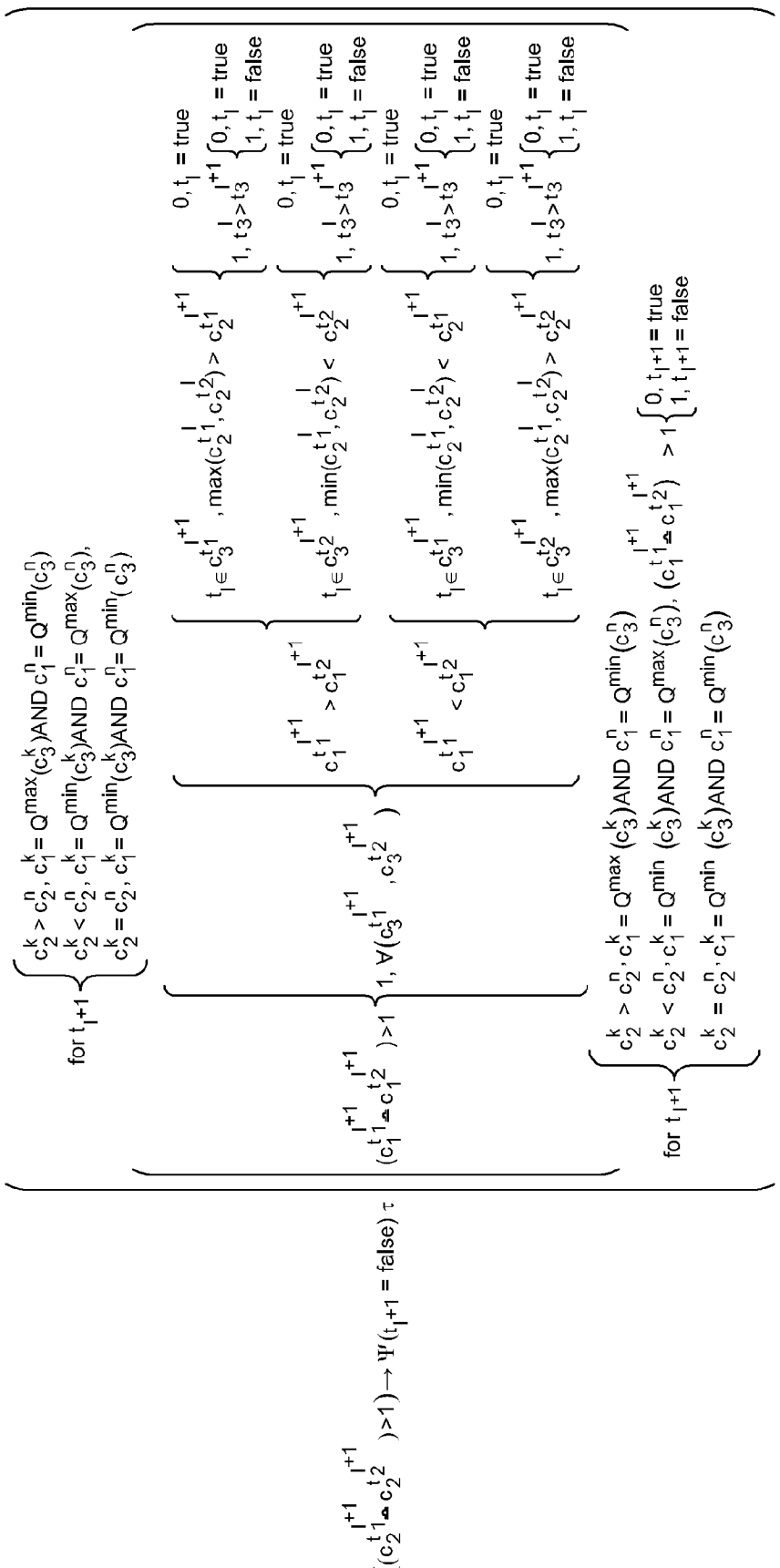


图 7