协作环境下的时空约束强制访问控制模型

范艳芳

(北京信息科技大学计算机学院 北京 100101) (网络文化与数字传播北京市重点实验室 北京 100101)

摘 要 安全的信息共享对信息系统而言至关重要。协作环境下的关键应用对信息共享和信息安全提出了更高的要求。已有的基于BLP模型的强制访问控制模型均无法满足协作环境下关键应用的访问控制需求。因此提出一种协作环境下的具有时空约束的强制访问控制模型,将任务、时间、空间等要素进行综合考虑,从而将逻辑安全和物理位置相结合,既增强了访问控制模型的安全性,又满足了协作环境下访问控制的灵活性。采用无干扰理论对所提模型的安全性进行了证明。

关键词 协作,强制访问控制,主动安全模型,时空约束,信息流,任务

中图法分类号 TP309 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.08.020

Temporal-Spatial-based Mandatory Access Control Model in Collaborative Environment

FAN Yan-fang

(Computer School, Beijing Information Science and Technology University, Beijing 100101, China)

(Beijing Key Laboratory of Internet Culture and Digital Dissemination Research, Beijing 100101, China)

Abstract Secure information sharing is a common goal for any information system. Critical applications in the collaborative environment put forward higher requirements for security and flexibility of information sharing. The existing mandatory access control model based on BLP model can't meet the requirements of access control for critical applications in collaborative environment. In this paper, a temporal-spatial-based mandatory access control model was proposed, which integrates task, time with space issues into access control model. Logic security is integrated with physical location in this model. So, it not only can enhance the security of access control, but also meets the flexibility of access control in collaborative environment. The security of the model was proved with non-interference theory.

Keywords Collaboration, Mandatory access control, Active security model, Temporal-spatial constraints, Information flow, Task

对访问控制的研究起源于20世纪60年代,其核心研究目标是解决安全数据共享问题。在过去几十年里,各种各样的访问控制模型被提出。最基础的访问控制模型有3类:自主访问控制模型、强制访问控制模型和基于角色的访问控制模型。其中,强制访问控制模型的安全性最高。

自我国推行信息安全等级保护制度以来,安全等级在3级及以上的信息系统都必须具备强制访问控制。这些被要求具备强制访问控制的信息系统被称为重要信息系统,其对安全提出了更高的要求。这类系统广泛支持BLP模型[1-4]的改进模型。BLP模型是由Bell和LaPadula于1973年首次提出,并得到广泛研究和应用的经典的强制访问控制模型。其核心是通过标记进行约束,实现信息从低向高流动,从而避免信息从高向低泄露。

BLP模型的不足在于其访问控制规则过于严格,从而导致灵活性不强,尤其是在协作环境下,BLP模型难以适用。

在协作环境下,经常要求多部门的多人配合来共同完成一项任务。不失一般性,以两方参与的协作环境为例,多方参与的协作环境与此类似。

一项任务需要从两个部门分别抽调一名员工参与。其中,来自部门1的甲是普通员工,按照他的级别,其安全标记为(秘密,{部门1}),即甲可以访问部门1所拥有的安全级别最高为秘密的文件。来自部门2的乙是部门2的负责人,按照他的级别,其安全标记为(机密,{部门2}),即乙可以访问部门2所拥有的安全级别最高为机密的文件。机密级的安全级高于秘密级。项目涉及到的文件为文件1(秘密,{部门1})和文件2(机密,{部门2})。

出于保密性需要,要求任务必须满足以下条件:1)所有参与者只能在上班时间(8点-17点)访问任务相关的材料;2)所有参与者只能在指定场所(如某个办公室或会议室)才能访问任务相关的材料;3)所有参与任务的人共享信息,即对任务相关的文件都有读权限,对任务形成的文件都有读写权限;4)参与任务期间,任务相关人员不能将任务相关内容外泄,即

到稿日期:2016-07-21 返修日期:2016-12-15 本文受国家自然科学基金面上项目(61672106),网络文化与数字传播北京市重点实验室开放课题(ICDD201609),北京市教委科研计划项目(KM201711232014),北京市自然科学基金项目(9021723401)资助。

范艳芳(1979-),女,博士,讲师,CCF会员,主要研究方向为安全模型、访问控制、云计算安全。

不能将任务相关内容写入非任务组人员可读的文件，非任务组人员对任务完成过程中形成的文件没有读写权限。

上述场景在协作环境下非常常见。BLP 模型要求：若用户 A 可以读文件 B，则 A 的安全级应大于或等于 B 的安全级，且 A 可以访问文件的范围包含 B 允许访问的范围；若用户 A 可以写文件 B，则 A 的安全级应小于或等于 B 的安全级，且 A 可以访问文件的范围包含于 B 允许访问的范围。遵循 BLP 模型，信息只能从低级别向高级别流动，双向的信息流仅能在同级别的主体间存在。如果将 BLP 模型应用于上述场景，则用户甲无权读文件 2 的内容，用户乙无权读文件 1 的内容。设项目执行过程中产生的文件为文件 3，则无论文件 3 的安全等级如何确定，都无法使甲、乙同时拥有对文件 3 的读、写权限。因此，无法实现信息在甲乙之间的双向流动，故 BLP 模型不适用于这类协作场景。文献[5]通过引入任务相关的访问控制属性，使得用户在参与任务时可以获得任务相关的访问权限，在不参与任务时，仅能依靠自身的安全等级进行访问，从而可以满足用户对文件的读写需求。但是一些关键应用[1)]，正如上述场景所述，其要求必须在指定的时间、地点进行访问，则文献[5]的模型无法满足此类需求。文献[6]在 BLP 模型的基础上扩展了时空约束，但是其与 BLP 模型一样，不适用于协作环境。文献[7-8]均对 BLP 模型的部分进行了改进，但是与协作无关，不能满足协作环境下的需求。文献[9-13]均考虑了协作环境下的某些场景，但是其不属于强制访问控制模型，安全性不足以满足要求。文献[14-15]考虑了多级安全系统中以组为中心的协作，但其更多地关注于组织内部与外部之间的协作，且缺乏时空约束。

综上，目前尚无访问控制模型能够满足上述协作环境下的所有需求。为此，本文在文献[1-4]的基础上，将时间和空间约束叠加在协作任务上，提出了一种协作环境下的时空约束强制访问控制模型(Collaboration supported Temporal-Spatial-based Mandatory Access Control Model，CTS-MAC)，其既满足协作环境下对信息共享的需求，又满足协作之外的信息流从低向高流动的需求，同时通过时间约束和空间约束，使模型的安全性得到进一步增强。

1 访问控制模型

参照文献[1-4]，定义如下模型元素。

1.1 模型元素

定义 1 一个访问控制模型可以定义为一个 8 元组$\langle S, O, C, K, D, P, T, A\rangle$，包括如下组成部分：$S$ 表示系统中主体的集合，$S_T \subset S$ 为可信主体集合，$S' = S - S_T$ 为非可信主体集合；O 表示系统中客体的集合；C 表示安全级别的集合；K 表示安全类别的集合，K^* 为其幂集；D 表示时间点的集合，集合元素用 d 表示，特别地，用 $d_c \in D$ 表示系统当前时间；P 表示位置的集合，集合元素用 p 表示，P^* 为其幂集；T 表示任务标识符的集合，T^* 为其幂集；A 表示访问方式的集合，$A = \{r, w\}$，其中 r 为读，w 为写。

文献[6]给出了主客体位置信息的表示和映射方法，该方法在本模型中同样适用。

在 BLP 模型中有 4 种访问方式：只读、追加、读写、执行。追加操作要求主体的安全等级支配客体的安全等级，读写操作要求主体的安全等级等于客体的安全等级。考虑到模型的主要目标是实现保密性，在本模型中仅定义一种写操作，要求执行写操作时主体的安全等级支配客体的安全等级。BLP 模型中的读写操作权限相当于同时拥有本模型中的读操作权限和写操作权限。BLP 模型中只读操作和执行操作对主客体安全等级之间的支配关系的要求是相同的，因此在本模型中仅考虑读操作。

定义 2 $L = C \times K^* \times T^*$ 表示安全等级的集合，集合元素 $l_i = (c_i, kt_i)$，其中 $c_i \in C, kt_i \subseteq K^* \times T^*$。$\propto$ 为定义在集合 L 上的偏序关系，读作“支配”，$l_i \propto l_j$ 当且仅当 $c_i \geqslant c_j \wedge kt_i \supseteq kt_j$。

在文献[1-3]的基础上将任务标识扩充到安全等级的集合中，其作用在于为主体的安全类别扩充任务标记，使得主体在参与任务期间对非任务相关的客体无写权限，从而避免了信息的泄露。

定义 3 PL 表示位置安全属性的集合。集合元素 $pl_i = (c_i, p_i)$，其中 $c_i \in C, p_i \subseteq P^*$。$\succ$ 为定义在集合 PL 上的偏序关系，$pl_i \succ pl_j$ 当且仅当 $c_i \geqslant c_j$ 且 $p_j \supseteq p_i$。

对于关键应用而言，必须对访问位置进行限制。然而，要求对客体的访问必须在客体的存储位置是不实际的。比如：若要求对某个信息系统的访问必须在存储相关数据的机房中进行，则会给实际工作带来诸多不便。为此，可以允许在某些符合安全策略的位置对客体进行访问。比如：用户可以在其办公室访问有权限的客体，而不是必须到机房进行。因此，在位置安全属性集合的元素中，位置表示为一个子集。在某一时间点，表示用户位置的子集中应只包含一个元素，即用户的当前位置。

定义 4 DD 表示时间区间的集合，对于 $\forall dd_i \in DD$，有 $dd_i = [d_1, d_2]$，其中 $d_1, d_2 \in D, d_1 \leqslant d_2$。

时间区间用于定义可以访问客体的时间区间以及允许主体访问的时间区间。

定义 5 B 表示当前访问方式的集合，集合元素 $b \in (S \times O \times A)^*$。

当前访问方式集用于定义当前主体对客体拥有的访问权限。

定义 6 M 表示系统访问控制矩阵的集合。用 $m \in M$ 表示当前的访问控制矩阵，m_{ij} 表示主体 s_i 对客体 o_j 的当前访问方式集。

定义 7 F 表示安全等级标记函数的集合，集合元素为 $\{subhighlev(s), subcurlev(s), objlev(o)\}$。其中，$subhighlev(s): S \rightarrow L$，表示主体 s 的最高安全等级标记函数；$subcurlev(s): S \rightarrow L$，表示主体 s 的当前安全等级标记函数，$subhighlev(s) \propto subcurlev(s)$；$objlev(o): O \rightarrow L$，表示客体 o 的安全等级标记函数。

1) 关键应用是指对系统的安全和运行起重要作用的应用，一旦遭到破坏，将对系统主要功能、安全性、可用性产生严重影响，甚至导致系统停止运行。

主体的最高安全等级标记函数的值为主体能够拥有的最高安全等级；主体的当前安全等级标记函数的值为主体当前的安全等级。比如，用户的最高安全级别为机密，但是他可以选择登录时的安全级别为秘密(机密的安全级高于秘密)。

定义 8 $V=B\times M\times F\times P\times D\times T$ 表示系统状态的集合，v 表示元素，v_0 表示初态。

本文的系统状态对文献[1-4]中的系统状态进行了扩充，包含了时间、空间、任务要素。

定义 9 PLF 表示位置安全属性标记函数的集合，集合元素为{$subposlev(s)$, $objposlev(o)$}。其中，$subposlev(s)$：$S\rightarrow PL$，表示主体 s 的位置安全属性标记函数；$objposlev(o)$：$O\rightarrow PL$，表示客体 o 的位置安全属性标记函数。

定义 10 PCF 表示位置约束检查函数的集合，集合元素为{$subposck$, $objposck$}。其中，$subposck$：$S\times F\times PLF\rightarrow${yes, no}，表示主体 s 的位置约束检查函数。假设 $subcurlev(s)=(c_1,k_1)$，$subposlev(s)=(c_2,p_1)$，判断规则为：如果 $c_2\leqslant c_1$，则 $subposck$=yes；否则，$subposck$=no。$objposck$：$O\times F\times PLF\rightarrow$ {yes, no}，表示客体 o 的位置约束检查函数。假设 $objlev(o)=(c_1,k_1)$，$objposlev(o)=(c_2,p_1)$，判断规则为：如果 $c_1\leqslant c_2$，则 $objposck$=yes；否则，$objposck$=no。

该约束检查函数表明主客体应遵循如下约束：主体所在位置的安全级不能高于主体的当前安全级；客体自身的安全级不应低于存储客体的位置的安全级。

定义 11 DCF 表示时间约束检查函数的集合，集合元素为{$subtimeck$, $objtimeck$}。其中，$subtimeck$：$S\times D\times DD\rightarrow$ {yes, no}，表示主体 s 的时间约束检查函数。假设 $d_c\in D$ 为系统当前时间，$[d_1,d_2]\in DD$ 为允许主体访问的时间区间，判断规则为：如果 $d_1\leqslant d_c\leqslant d_2$，则 $subtimeck$=yes；否则，$subtimeck$=no。$objtimeck$：$O\times D\times DD\rightarrow$ {yes, no}，表示客体 o 的时间约束检查函数。假设 $d_c\in D$ 为系统当前时间，$[d_1,d_2]\in DD$为允许客体被访问的时间区间，判断规则为：如果 $d_1\leqslant d_c\leqslant d_2$，则 $objtimeck$=yes；否则，$objtimeck$=no。

时间约束检查函数用于检查主客体是否满足时间约束。对于安全性要求较高的系统而言，主体只能在规定的时间内访问客体，而客体也只能在规定的时间内被访问。

定义 12 TIF 表示任务标记函数的集合，集合元素为{$subtaskid(s)$, $objtaskid(o)$}。其中，$subtaskid(s)$：$S\rightarrow T^*$，表示主体 s 的任务标识标记函数；$objtaskid(o)$：$O\rightarrow T^*$，表示客体 o 的任务标识标记函数。

一个主体同时只能参加一个任务，而一个客体可以同时属于多个任务。当主、客体不属于任何一个任务时，其任务标识为空集。

定义 13 $taskpos$：$T\rightarrow P^*$，表示任务执行位置映射函数。

任务执行位置映射函数将任务映射到一个位置集合的子集，其结果表示允许参与任务的位置。对于一些安全性要求较高的应用，常要求主体只能在指定位置参与任务。

定义 14 $tasktime$：$T\rightarrow DD$，表示任务执行时间映射函数。

任务执行时间映射函数将任务映射到一个时间区间，其结果表示允许参与任务的时间。

定义 15 TCF 表示任务检查函数集合，集合元素为{$taskposck$, $tasktimeck$}。其中，$taskposck$：$S\times PL\times T\rightarrow$ {yes, no}，表示任务域位置检查函数。设 $subposlev(s)=(c_1,p_1)$，判断规则为：如果 $p_1\subseteq taskpos(subtaskid(s))$，$taskposck$=yes；否则 $taskposck$=no。$tasktimeck$：$S\times D\times DD\rightarrow$ {yes, no} 表示任务域时间检查函数。设 $d_c,d_1,d_2\in D$，d_c 为系统当前时间，$tasktime(subtaskid(s))=[d_1,d_2]$，判断规则为：如果 $d_1\leqslant d_c\leqslant d_2$，则 $tasktimeck$=yes；否则，$tasktimeck$=no。

任务域时间/位置检查函数对主体的当前访问时间/位置是否满足任务的要求进行检查。若满足要求，则返回 yes；否则返回 no。

在实际工作中，多用户常聚集在同一个地方来协作完成某项任务，以便更安全、高效地进行沟通和交流。这种将任务限定在指定位置进行的方式能够提供更好的安全性。

定义 16 $objtype$：$O\rightarrow$ {release, draft} 表示客体类型映射函数。其中 release 表示客体为发布类型，即客体的内容已经确定，其拥有确定的安全属性；draft 表示客体的类型为草稿类型，即客体的内容尚未完全确定，其拥有的安全属性为临时指派的。

对客体进行分类的主要目的是对不同类型的客体提供不同的保护。协作过程中产生的一些中间文件为草稿类型的客体。

定义 17 $observe$：$S\rightarrow O^*$，表示主体有读权限的客体的集合。

定义 18 $alter$：$S\rightarrow O^*$，表示主体有写权限的客体的集合。

1.2 模型安全属性

下面对模型的安全属性进行定义。参考文献[5]的思路，将模型的安全属性分为两类。在进行访问控制时，满足其中任何一类，均允许访问。

(1)基于安全级的安全属性(Security Level based Security Property，SLSP)

1)读安全属性。状态 $v=(b,m,f,p,d,t)$ 满足读安全属性，当且仅当 $o\in observe(s)\Rightarrow$

$s\in S_T$

$subhighlev(s)\propto objlev(o)\wedge subposlev(s)\succ objposlev(o)\wedge subposck$=yes $\wedge$ $objposck$=yes $\wedge$ $subtimeck$=yes $\wedge$ $objtimeck$=yes

该属性说明可信主体在进行读操作时应同时满足如下条件：主体的最高安全等级支配客体的安全等级；主体的位置属性支配客体的位置属性；主体和客体满足位置约束；主体和客体满足时间约束。该属性保证主体不能在低安全级的环境中读高安全级的客体，从而避免了由于环境不安全可能导致的信息泄露。

$s\in S'$

$subcurlev(s)\propto objlev(o)\wedge subposlev(s)\succ objposlev(o)\wedge subposck$=yes $\wedge$ $objposck$=yes $\wedge$ $subtimeck$=yes $\wedge$ $objtimeck$=yes

该属性说明非可信主体在进行读操作时应同时满足如下条件：主体的当前安全等级支配客体的安全等级；主体的位置属性支配客体的位置属性；主体和客体满足位置约束；主体和客体满足时间约束。

可信主体和非可信主体的区别在于:对可信主体,仅要求其最高安全级满足对客体安全级的支配即可;而对非可信主体,还要求其当前安全级满足对客体安全级的支配。

2)写安全属性。状态 $v=(b,m,f,p,d,t)$ 满足写安全属性,当且仅当 $o\in alter(s)\Rightarrow$

$s\in S_T$

$subhighlev(s)\propto objlev(o)\wedge subposck=\text{yes}\wedge objposck=\text{yes}\wedge subtimeck=\text{yes}\wedge objtimeck=\text{yes}$

该属性说明可信主体在进行写操作时应同时满足如下条件:主体的最高安全等级支配客体的安全等级;主体和客体满足位置约束;主体和客体满足时间约束。

由于在安全等级比较低的环境中依然允许主体写高安全等级的客体,因此在本属性中不必满足主体的位置属性支配客体的位置属性。

$s\in S'$

$objlev(o)\propto subcurlev(s)\wedge subposck=\text{yes}\wedge objposck=\text{yes}\wedge subtimeck=\text{yes}\wedge objtimeck=\text{yes}$

该属性说明非可信主体在进行写操作时应同时满足如下条件:客体的安全等级支配主体的当前安全等级;主体和客体满足位置约束;主体和客体满足时间约束。

3)自主安全属性。状态 $v=(b,m,f,p,d,t)$ 满足自主安全属性,当且仅当对于每个 $s\in S,(s_i,o_j,x)\in b\Rightarrow x\in m_{ij}\wedge objtype(o)=\text{release}$。

该属性说明,只有发布类型的客体才能够依据基于安全等级的安全属性进行访问。

满足 SLSP 属性当且仅当同时满足上述 3 条安全属性。

(2)基于任务的安全属性(Task based Security Property,TSP)

$o\in observe(s)\Rightarrow r\in m_{ij}\wedge objposck=\text{yes}\wedge objtimeck=\text{yes}\wedge taskposck=\text{yes}\wedge tasktimeck=\text{yes}\wedge subtaskid(s)\subseteq objtaskid(o)$

$o\in alter(s)\Rightarrow w\in m_{ij}\wedge objtype(o_j)=\text{draft}\wedge objposck=\text{yes}\wedge objtimeck=\text{yes}\wedge taskposck=\text{yes}\wedge tasktimeck=\text{yes}\wedge subtaskid(s)\subseteq objtaskid(o)$

该属性说明在进行读操作时,应同时满足如下条件:在访问控制矩阵中,主体拥有对客体的读权限;客体的任务标识包含了主体的任务标识;主体和客体满足位置约束和时间约束。

在进行写操作时,应同时满足如下条件:在访问控制矩阵中,主体拥有对客体的写权限;客体的类型为草稿类型;客体的任务标识包含了主体的任务标识;主体和客体满足位置约束和时间约束。

写和读操作的区别主要在于对客体类型的约束。读操作不约束客体类型,即只要客体的任务标识包含了主体的任务标识,主体就可以对客体进行读访问。而写操作要求客体必须为草稿类型的客体。这个约束保证了参加任务的主体不能将读到的任务相关的信息写入任务之外的其他客体中,从而保证了任务相关信息不会外泄。此外,与 SLSP 属性不同,主体的时间约束和位置约束都是任务相关的。

2 模型安全性分析

无干扰理论[16]的基本思想可以理解为:如果用户甲对文件 1 的写操作没有影响到用户乙所能够读到的文件 1 的内容,则可认为用户甲不干扰用户乙。无干扰理论自提出后,常被用于从信息流的角度对访问控制模型的安全性进行证明。其主要思想可以理解为:如果能够证明满足访问控制安全属性的访问符合访问控制策略,则访问控制模型是安全的。下面沿用文献[16]的方法对本文所提出的模型进行安全性证明。

使用无干扰理论进行证明前,需要增加相关的定义和定理。

定义 19 Q 为系统动作的集合,用 q 表示动作,用希腊字符 α,β 等表示动作序列。

系统动作本质上是对访问方式的抽象。

定义 20 Z 为系统输出结果集合。

定义 21 J 为客体值的集合。

由于需要区分对客体进行访问操作前后客体值是否发生变化,因此需要将客体和客体值分开定义。

定义 22 $value:V\times O\rightarrow J$,表示状态 V 下客体 O 的值。

定义 23 $dom:O\rightarrow S$,表示发起动作的主体。

定义 24 $step:V\times O\rightarrow V$,表示系统执行完一个动作后,将从一个状态转换到另一个状态。$step(v_1,q)=v_2$ 表示执行动作 q 后,系统由状态 v_1 转换到状态 v_2。

定义 25 $output:V\times Q\rightarrow Z$。$output(v,q)$ 为系统在状态 v 下执行动作 q 的输出。

定义 26 $run:V\times Q^*\rightarrow V$,其中 Q^* 表示 Q 的幂集。$run(v,\odot)=v,\odot$ 表示系统没有任何动作,$run(v,q\circ\beta)=run(step(v,q),\beta)$,$\circ$ 表示连接。

该函数表示系统在某个状态下,执行一系列动作后,转入另一个状态。

定义 27(访问控制策略 $\triangleright$) 其是定义在主体集合 $S\times S$ 上的一个自反关系,表示主体间允许的干扰关系。$\not\triangleright=(S\times S)\setminus\triangleright$,$\setminus$ 表示差集。

$s_1,s_2\in S,s_1\triangleright s_2$ 表示访问控制策略允许主体 s_1 的执行干扰主体 s_2。$s_1\not\triangleright s_2$ 表示访问控制策略不允许主体 s_1 的执行干扰主体 s_2。

主体间的干扰关系可以通过客体来体现。例:若主体 s_1 有权读客体 o,主体 s_2 有权写客体 o,则通过客体 o,主体 s_2 干扰了主体 s_1。

定义 28 清除函数 $purge:Q^*\times S\rightarrow Q^*$,$purge(\odot,s)=\odot$,$purge(q\circ\beta,s)=\begin{cases}q\circ purge(\beta,s), & \text{如果}(dom(q)\triangleright s)\\ purge(\beta,s), & \text{否则}\end{cases}$。

函数 $purge(\beta,s)$ 的结果,是依据访问控制策略 $\triangleright$ 将不允许干扰 s 的动作从动作序列 β 中删除后得到的子序列。

对于执行给定动作序列 β 的主体 s 而言,若不使用清除函数和使用清除函数 $purge(\beta,s)$ 后得到的系统状态对 s 而言无法区分,则该系统是一个安全系统。为此,有如下定义:

定义 29 满足如下条件的系统关于 $\triangleright$ 是安全的。

$output(run(v_0,\beta),q)=output(run(v_0,purge(\beta,dom(q))),q)$

定义 30 若对每一个主体 $s\in S$,在状态集 V 上存在等价关系 $\overset{s}{\sim}$,则称系统是可分区观察的。

定义 30 将系统的状态划分为与主体相关的等价类,从而可通过观察等价类下客体的值来判断主体是否被干扰。

定理 1(展开定理)　如果系统是可分区观察的,满足如下条件的系统对 $\triangleright$ 是安全的[16]。

1) $v_1 \overset{dom(q)}{\sim} v_2 \rightarrow output(v_1,q)=output(v_2,q)$;

2) $v_1 \overset{s}{\sim} v_2 \rightarrow step(v_1,q) \overset{s}{\sim} step(v_2,q)$;

3) $dom(q)\ \&s \rightarrow v \overset{s}{\sim} step(v,q)$。

定理证明见文献[16]。展开定理的重要意义在于它提供了一种关联无干扰策略和访问控制机制的途径。根据定义,符号 $\triangleright$ 表示抽象的访问控制策略。下面首先给出满足引言中场景的访问控制策略,然后证明满足 CTS-MAC 模型的 TSP 属性或 SLSP 属性的访问请求符合上述访问控制策略。

定义 31　对主体 s 而言,如果主体在两个状态下读到的客体的值是相同的,则称主体 s 在两个状态下是等价的,即

$$v_1 \overset{s}{\sim} v_2 \leftrightarrow \forall o \in observe(s): value(v_1,o)=value(v_2,o)$$

访问控制机制的实施由引用监控器完成。对引用监控器做如下假设[16]:

1)主体执行动作 q 后的输出值仅依赖于 $dom(q)$ 可以读到的客体的值。

$$v_1 \overset{dom(q)}{\sim} v_2 \rightarrow output(v_1,q)=output(v_2,q) \tag{1}$$

2)如果主体执行动作 q 后改变了客体的值,则被改变的客体的值仅依赖于 $dom(q)$ 可以读到的客体的值。

$$v_1 \overset{dom(q)}{\sim} v_2 \wedge (value(step(v_1,q),o)\neq value(v_1,o) \vee value(step(v_2,q),o)\neq value(v_2,o)) \rightarrow value(step(v_1,q),o)=value(step(v_2,q),o) \tag{2}$$

3)如果主体执行动作 q 后改变了客体 o 的值,则 $dom(q)$ 一定有改变 o 的权限。

$$value(step(v_1,q),o)\neq value(v_1,o) \rightarrow o \in alter(dom(q)) \tag{3}$$

引用监控器假设表明,系统中所有对客体的操作都必须经过引用监控器。

定义 32　对于 $s_1,s_2 \in S, o \in observe(s_2), o \in alter(s_1)$,模型的访问控制策略 $\triangleright$ 定义为 $s_1 \triangleright s_2 \leftrightarrow$:

1) $s_1,s_2 \in S_T$

$$r \in m_{s_2 o} \wedge w \in m_{s_1 o} \wedge subhighlev(s_2) \propto objlev(o) \wedge subhighlev(s_1) \propto objlev(o) \wedge objtype(o)=\text{release} \wedge subposlev(s_2) \succ objposlev(o) \wedge subtimeck=\text{yes} \wedge objtimeck=\text{yes} \wedge subposck=\text{yes} \wedge objposck=\text{yes} \tag{4}$$

2) $s_1,s_2 \in S'$

$$r \in m_{s_2 o} \wedge w \in m_{s_1 o} \wedge subcurlev(s_2) \propto objlev(o) \propto subcurlev(s_1) \wedge objtype(o)=\text{release} \wedge subposlev(s_2) \succ objposlev(o) \wedge subtimeck=\text{yes} \wedge objtimeck=\text{yes} \wedge subposck=\text{yes} \wedge objposck=\text{yes} \tag{5}$$

3) $s_1 \in S' \wedge s_2 \in S_T$

$$r \in m_{s_2 o} \wedge w \in m_{s_1 o} \wedge subhighlev(s_2) \propto objlev(o) \propto subcurlev(s_1) \wedge objtype(o)=\text{release} \wedge subposlev(s_2) \succ objposlev(o) \wedge subtimeck=\text{yes} \wedge objtimeck=\text{yes} \wedge subposck=\text{yes} \wedge objposck=\text{yes} \tag{6}$$

4) $s_1 \in S_T \wedge s_2 \in S'$

$$r \in m_{s_2 o} \wedge w \in m_{s_1 o} \wedge subcurlev(s_2) \propto objlev(o) \wedge subhighlev(s_1) \propto objlev(o) \wedge objtype(o)=\text{release} \wedge subposlev(s_2) \succ objposlev(o) \wedge subtimeck=\text{yes} \wedge objtimeck=\text{yes} \wedge subposck=\text{yes} \wedge objposck=\text{yes} \tag{7}$$

5) $r \in m_{s_2 o} \wedge w \in m_{s_1 o} \wedge objtype(o)=\text{draft} \wedge taskposck=\text{yes} \wedge tasktimeck=\text{yes} \wedge objtimeck=\text{yes} \wedge objposck=\text{yes} \wedge subtaskid(s_2) \subseteq objtaskid(o) \wedge subtaskid(s_1) \subseteq objtaskid(o)$ (8)

定理 2　满足引用监控器假设和以下任意属性之一的系统对于定义 32 中的访问控制策略 $\triangleright$ 是安全的。

SLSP 属性:

$$o \in observe(s) \rightarrow (s \in S_T \wedge r \in m_{so} \wedge subhighlev(s) \propto objlev(o) \wedge objtype(o)=\text{release} \wedge subposlev(s) \succ objposlev(o) \wedge subposck=\text{yes} \wedge objposck=\text{yes} \wedge subtimeck=\text{yes} \wedge objtimeck=\text{yes}) \tag{9}$$

$$\vee (s \in S' \wedge r \in m_{so} \wedge subcurlev(s) \propto objlev(o) \wedge objtype(o)=\text{release} \wedge subposlev(s) \succ objposlev(o) \wedge subposck=\text{yes} \wedge objposck=\text{yes} \wedge subtimeck=\text{yes} \wedge objtimeck=\text{yes}) \tag{10}$$

$$o \in alter(s) \rightarrow (s \in S_T \wedge w \in m_{so} \wedge subhighlev(s) \propto objlev(o) \wedge objtype(o)=\text{release} \wedge subposck=\text{yes} \wedge objposck=\text{yes} \wedge subtimeck=\text{yes} \wedge objtimeck=\text{yes}) \tag{11}$$

$$\vee (s \in S' \wedge w \in m_{so} \wedge objlev(o) \propto subcurlev(s) \wedge objtype(o)=\text{release} \wedge subposck=\text{yes} \wedge objposck=\text{yes} \wedge subtimeck=\text{yes} \wedge objtimeck=\text{yes}) \tag{12}$$

TSP 属性:

$$o \in observe(s) \rightarrow r \in m_{so} \wedge taskposck=\text{yes} \wedge tasktimeck=\text{yes} \wedge objposck=\text{yes} \wedge objtimeck=\text{yes} \wedge subtaskid(s) \subseteq objtaskid(o) \tag{13}$$

$$o \in alter(s) \rightarrow w \in m_{so} \wedge taskposck=\text{yes} \wedge tasktimeck=\text{yes} \wedge objposck=\text{yes} \wedge objtimeck=\text{yes} \wedge objtype(o)=draft \wedge subtaskid(s) \subseteq objtaskid(o) \tag{14}$$

证明:要证明定理 2 成立,可以证明满足定理 2 的条件时定理 1 成立。要使定理 1 成立,需要证明定理 1 的如下 3 个条件成立:

$$1)\ v_1 \overset{dom(q)}{\sim} v_2 \rightarrow output(v_1,q)=output(v_2,q) \tag{15}$$

$$2)\ v_1 \overset{s}{\sim} v_2 \rightarrow step(v_1,q) \overset{s}{\sim} step(v_2,q) \tag{16}$$

$$3)\ dom(q)\ \&s \rightarrow v \overset{s}{\sim} step(v,q) \tag{17}$$

下面首先证明满足 SLSP 属性的访问符合访问控制策略,然后对 TSP 属性进行证明。

(1)SLSP 属性的证明

根据 s 和 $dom(q)$ 分别为可信主体和非可信主体,共有 4 种情况,其证明思路相同。下面仅以 s 和 $dom(q)$ 均为非可信主体为例进行证明。

式(15)可直接由式(1)得到。下面证明满足 SLSP 属性，式(16)和式(17)成立。

首先证明式(16)成立。式(16)可以被重写为：对于 $o\in observe(s)$，以下式子成立：

$v_1 \overset{s}{\sim} v_2 \rightarrow value(step(v_1,q),o)=value(step(v_2,q),o)$

根据定义 31，可得：

$value(v_1,o)=value(v_2,o)\wedge o\in observe(s)$ (18)

下面分 3 种情况进行证明：

1) $value(v_1,o)\neq value(step(v_1,q),o)$

由式(3)得：$o\in alter(dom(q))$。

由于 $o\in observe(s)$，根据式(10)和式(12)，可得：

$objlev(o)\propto subcurlev(dom(q))\wedge objtype(o)=$ release $\wedge$ $subcurlev(s)\propto objlev(o)\wedge w\in m_{dom(q)o}\wedge r\in m_{so}\wedge subposlev(s)\succ objposlev(o)\wedge subposck=$ yes $\wedge objposck=$ yes $\wedge subtimeck=$ yes $\wedge objtimeck=$ yes

根据偏序关系的传递性，有：

$r\in m_{so}\wedge w\in m_{dom(q)o}\wedge subcurlev(s)\propto objlev(o)\propto subcurlev(dom(q))\wedge objtype(o)=$ release $\wedge subposlev(s)\succ objposlev(o)\wedge subposck=$ yes $\wedge objposck=$ yes $\wedge subtimeck=$ yes $\wedge objtimeck=$ yes

由式(5)可知 $dom(q)\triangleright s$，由 $v_1 \overset{s}{\sim} v_2$ 可得 $v_1 \overset{dom(q)}{\sim} v_2$。由式(2)可得：$value(step(v_1,q),o)=value(step(v_2,q),o)$。

2) $value(v_2,o)\neq value(step(v_2,q),o)$

同情况 1)，证明过程省略。

3) $value(v_1,o)=value(step(v_1,q),o)\wedge value(v_2,o)=value(step(v_2,q),o)$ (19)

由式(18)和式(19)，可得：

$value(step(v_1,q),o)=value(step(v_2,q),o)$

综上，式(16)成立。

下面证明式(17)成立。

使用反证法进行证明，需要证明：

$\exists o\in observe(s): value(v_1,o)\neq value(step(v_1,q),o)\rightarrow dom(q)\triangleright s$。

如果 $value(v_1,o)\neq value(step(v_1,q),o)$，根据式(3)，有 $o\in alter(dom(q))$，由前提可知 $o\in alter(dom(q))\wedge o\in observe(s)$，根据式(10)和式(12)，可得：

$objlev(o)\propto subcurlev(dom(q))\wedge objtype(o)=$ release $\wedge$ $subcurlev(s)\propto objlev(o)\wedge w\in m_{dom(q)o}\wedge r\in m_{so}\wedge subposlev(s)\succ objposlev(o)\wedge subposck=$ yes $\wedge objposck=$ yes $\wedge subtimeck=$ yes $\wedge objtimeck=$ yes

根据偏序关系的传递性，有：

$r\in m_{so}\wedge w\in m_{dom(q)o}\wedge subcurlev(s)\propto objlev(o)\propto subcurlev(dom(q))\wedge objtype(o)=$ release $\wedge subposlev(s)\succ objposlev(o)\wedge subposck=$ yes $\wedge objposck=$ yes $\wedge subtimeck=$ yes $\wedge objtimeck=$ yes

由式(5)可知 $dom(q)\triangleright s$。式(17)成立。

(2)TSP 属性的证明

首先证明式(16)成立。式(16)可以被重写为：对于 $o\in observe(s)$，式子：$v_1 \overset{s}{\sim} v_2\rightarrow value(step(v_1,q),o)=value(step(v_2,q),o)$ 成立。

根据定义 31，可得：$value(v_1,o)=value(v_2,o)\wedge o\in observe(s)$。

下面分 3 种情况进行证明：

1) $value(v_1,o)\neq value(step(v_1,q),o)$

由式(3)得：$o\in alter(dom(q))$。

由于 $o\in observe(s)$，根据式(13)和式(14)，可得：

$subtaskid(dom(q))\subseteq objtaskid(o)\wedge objtype(o)=$ draft $\wedge w\in m_{dom(q)o}\wedge r\in m_{so}\wedge subtaskid(s)\subseteq objtaskid(o)\wedge taskposck=$ yes $\wedge objtimeck=$ yes $\wedge tasktimeck=$ yes $\wedge objposck=$ yes

由式(8)可知 $dom(q)\triangleright s$，由 $v_1 \overset{s}{\sim} v_2$ 可得：$v_1 \overset{dom(q)}{\sim} v_2$，由式(2)可得：$value(step(v_1,q),o)=value(step(v_2,q),o)$。

2) $value(v_2,o)\neq value(step(v_2,q),o)$

同情况 1)，证明过程省略。

3) $value(v_1,o)=value(step(v_1,q),o)\wedge value(v_2,o)=value(step(v_2,q),o)$

由式(18)和式(19)，可得：

$value(step(v_1,q),o)=value(step(v_2,q),o)$

综上，式(16)成立。

下面证明式(17)成立。

使用反证法进行证明，需要证明：

$\exists o\in observe(s): value(v_1,o)\neq value(step(v_1,q),o)\rightarrow dom(q)\triangleright s$

如果 $value(v_1,o)\neq value(step(v_1,q),o)$，根据式(3)，有 $o\in alter(dom(q))$，由前提可知 $o\in alter(dom(q))\wedge o\in observe(s)$，根据式(13)和式(14)，得到：

$subtaskid(s)\subseteq objtaskid(o)\wedge objtype(o)=$ draft $\wedge w\in m_{dom(q)o}\wedge r\in m_{so}\wedge subtaskid(dom(q))\subseteq objtaskid(o)\wedge taskposck=$ yes $\wedge tasktimeck=$ yes $\wedge objposck=$ yes $\wedge objtimeck=$ yes

由式(8)可知 $dom(q)\triangleright s$。式(17)成立。

证毕。

定理 2 证明了满足 SLSP 和 TSP 属性的系统关于定义 32 中的访问控制策略是安全的。

3 应用实例

下面将本文提出的 CTS-MAC 模型应用于引言中的场景。设任务的标识为 T1，允许访问任务的位置为 302 室，项目执行过程中产生的文件为文件 3，主体和客体的安全标记分别为：

甲(秘密，{部门 1，T1}，{302 室}，[8，17]，{T1})；

乙(机密，{部门 2，T1}，{302 室}，[8，17]，{T1})；

文件 1(秘密，{部门 1}，release，{302 室，机房}，[8，17]，{T1，T2})；

文件 2(机密，{部门 2}，release，{302 室，机房}，[8，17]，{T1，T3})；

文件 3(秘密,{部门 2},draft,{302 室,机房},[8,17],{T1})。

其中,主体的安全标记分别表示:安全级别、安全类别、当前位置、允许访问的时间、任务标识;客体的安全标记分别表示:安全级别、安全类别、客体类型、允许访问的位置、允许访问的时间、任务标识。文件 3 为 draft 类型的客体,在任务结束时将被转换成 release 类型的客体。此外,每个文件可以同时属于多个任务,因此文件 1 和文件 2 具有多个任务标识。

由于甲的安全等级支配文件 1 的安全等级,且文件 1 为 release 类型,甲的当前位置符合文件 1 对位置的要求,若甲的当前访问时间符合文件 1 对时间的要求,且文件 1 的存放位置符合位置安全级要求,则甲对文件 1 有读权限(满足 SLSP 属性);由于甲的安全类别中扩充了任务标识 T1,使得甲对文件 1 无写权限(不满足 SLSP)。由于文件 1 为 release 类型,因此甲对文件 1 无写权限(不满足 TSP 属性)。同样,对任务 T1 之外的任何文件,甲都无写权限。

由于文件 2 的任务标识包含了甲的任务标识,甲的当前位置符合任务和文件 2 对位置的要求,若甲的当前访问时间符合文件 2 对时间的要求,且文件 2 的存放位置符合位置安全级要求,则甲对文件 2 有读权限(满足 TSP 属性);由于文件 2 为 release 类型,因此甲对文件 2 没有写权限(不满足 TSP 属性);由于文件 2 的安全等级不能支配甲的安全等级,因此甲对文件 2 没有写权限(不满足 SLSP)。

由于文件 3 的任务标识包含了甲的任务标识,甲的当前位置符合文件 3 对位置的要求,若甲的当前访问时间符合文件 3 对时间的要求,则甲对文件 3 有读权限(满足 TSP 属性);由于文件 3 的类型为 draft,因此甲对文件 3 有写权限(满足 TSP 属性)。

同理,可知乙对文件 1 和文件 2 有读权限,对文件 3 有读写权限。

综上,任务相关信息可在甲、乙之间(任务域范围内)流动,且不会被泄露到任务之外的其他文件中,满足引言中的访问控制需求。

4　与其他模型的比较

现有的支持协作的访问控制模型大多基于两种模型提出,一种是考虑了信息流控制的模型[5,14-15],另一种是未考虑信息流控制的模型[9-10]。

文献[5]提出的 CSMAC 是基于 BLP 模型的,在模型中通过引入任务为中心的访问控制,使得模型的灵活性得到了极大增强。但是,该模型中并未考虑时空约束,而在很多应用中需要同时从时空方面对访问控制进行进一步约束,以更好地保证安全性。本文的 CTS-MAC 模型在 CSMAC 模型的基础上引入时空约束,不仅更严格地约束了主、客体的访问关系,而且将物理位置和已有的逻辑控制相结合,使得模型的安全性得到进一步增强。

文献[9-10]是以组为中心的访问控制,在模型中都体现出访问控制的权限与任务相关的思想。但是这些模型均缺乏对信息流的约束,安全性不足;同时,这些模型也都未考虑时空约束。

文献[14]提出了多级安全系统中以组为中心的协作模型 GEI,该模型主要考虑了与外部进行协作时如何保持安全性。该模型在操作中考虑了主、客体参加或离开组的各种操作,但是其只给出了管理模型和操作模型的形式化描述,未对所提模型的安全性进行分析,模型中也缺乏时空约束。

在文献[14]的基础上,文献[15]提出了一个 LCC 模型。该模型在传统安全类别的基础上增加了新的协作类别。文中对提出的 LCC 模型进行了形式化的描述,对 LCC 和 GEI 的安全性进行了证明。LCC 与 CTS-MAC 的区别在于,LCC 模型侧重于组织内部和外部之间的协作,而 CTS-MAC 主要关注组织内部不同部门之间的协作。此外,LCC 并不考虑时空约束。

各模型的具体对比信息如表 1 所列。

表 1　模型对比表

评价指标	BLP[1-4]	CSMAC[5]	TMAC[9]	C-TMAC[10]	GEI[14]	LCC[15]	CTS-MAC
模型复杂性	低	低	中	中	低	低	低
易用性	低	高	高	高	高	高	高
应用性	中	高	中	高	高	高	高
时间约束	不支持	不支持	不支持	不支持	不支持	不支持	支持
空间约束	不支持	不支持	不支持	不支持	不支持	不支持	支持
上下文信息	低	中	中	中	中	中	中
主动/被动	被动	主动	主动	主动	主动	主动	主动
信息流控制	支持	支持	不支持	不支持	支持	支持	支持
灵活性	低	中	中	中	中	中	中
安全性	高	高	中	中	高	高	高

结束语　兼具灵活性和安全性的访问控制模型一直是访问控制模型设计的目标之一。然而现有的基于 BLP 模型的强制访问控制模型不够灵活,无法满足协作环境下关键应用对访问控制的要求。本文提出的访问控制模型不仅满足了协作环境下不参与任务的信息流仅能从低到高流动的要求,还满足了在协作时信息可以在指定的任务域内流动,且不会破坏原有文件的安全性目标。同时通过时间和空间约束,将物理位置结合进来,更进一步提高了模型的安全性。

泛在计算、移动计算、云计算等新型计算模式的不断出现和发展,对访问控制提出了新的要求。目前越来越多的政务系统被移植到云端,将关键应用部署到云端已成为一种趋势。如何满足这些计算模式下的访问控制新需求是需要进一步研究的内容。

参 考 文 献

[1] BELL D E, LAPADULA L J. Secure Computer Systems: Mathematical Foundations [R]. Electronic Systems Division, Air Force Systems Command, Hanscom Air Force Base, Bedford, MA, USA, 1973.

[2] BELL D E, LAPADULA L J. Secure Computer Systems: A Mathematical Model [R]. Electronic Systems Division, Air Force Systems Command, Hanscom Air Force Base, Bedford, MA, USA, 1973.

[3] BELL D E, LAPADULA L J. Secure Computer Systems: A Refinement of the Mathematical Model [R]. Electronic Systems Division, Air Force Systems Command, Hanscom Air Force Base, Bedford, MA, USA, 1974.

[4] BELL D E, LAPADULA L J. Secure Computer System: Unified Exposition and MULTICS Interpretation: MTR-2997 Rev. 1 [R]. The MITRE Corporation, Bedford, MA, USA, 1976.

[5] FAN Y F, CAI Y. Collaboration Supported Mandatory Access Control Model[J]. Journal of Computer Research and Development, 2015, 52(10): 2411-2421. (in Chinese)
范艳芳，蔡英. 支持协作的强制访问控制模型[J]. 计算机研究与发展，2015，52(10)：2411-2421.

[6] FAN Y F, CAI Y, GENG X H. A Mandatory Access Control Model with Temporal and Spatial Constraints [J]. Journal of Beijing University of Posts and Telecommunications, 2012, 35(5): 111-114. (in Chinese)
范艳芳，蔡英，耿秀华. 具有时空约束的强制访问控制模型[J]. 北京邮电大学学报，2012，35(5)：111-114.

[7] WU Y J, LIANG H L, ZHAO C. A Multi-Level Security Model with Least Privilege Support for Trusted Subject[J]. Journal of Software, 2007, 18(3): 730-738 (in Chinese)
武延军，梁洪亮，赵琛. 一个支持可信主体特权最小化的多级安全模型[J]. 软件学报，2007，18(3)：730-738.

[8] ZHANG X F, XU F, SHEN C X. Research on Multilevel Security Model Based on Trustworthy State and Its Application[J]. Acta Electronica Sinica, 2007, 35(8): 1511-1515. (in Chinese)
张晓菲，许访，沈昌祥. 基于可信状态的多级安全模型及其应用研究[J]. 电子学报，2007，35(8)：1511-1515.

[9] THOMAS R K. Team-based Access Control (TMAC): A Primitive for Applying Role-based Access Controls in Collaborative Environments [C]// Proc of the 2nd Workshop on Role-Based Access Control. ACM, Fairfax, VA, USA, 1997: 13-19.

[10] GEORGIADIS C K, MAVRIDIS I, PANGALOS G, et al. Flexible Team-based Access Control Using Contexts [C] // The ACM Symposium on Access Control Models and Technologies 2001. Chantilly, Vriginia, USA, 2001: 21-27.

[11] 翟治年. 企业级协作环境中访问控制模型研究[D]. 北京：华南理工大学，2012.

[12] BIJON K Z, SANDHU R S, KRISHNAN R. A Group-centric Model for Collaboration with Expedient Insiders in Multilevel Systems [C]// The 2012 International Conference on Collaboration Technologies and Systems. 2012: 419-426.

[13] YAN X X, GENG T. Fused access control scheme for sensitive data sharing[J]. Journal on Communications, 2014, 35(8): 71-77. (in Chinese)
闫玺玺，耿涛. 面向敏感数据共享环境下的融合访问控制机制[J]. 通信学报，2014，35(8)：71-77.

[14] BIJON K Z, SANDHU R, KRISHNAN R. A group-centric model for collaboration with expedient insiders in multilevel systems[C]// International Conference on Collaboration Technologies and Systems. IEEE, 2012: 419-426.

[15] BIJON K Z, SANDHU R, KRISHNAN R, et al. A lattice interpretation of group-centric collaboration with expedient insiders [C] // International Conference on Collaborative Computing: Networking, Applications and Worksharing. IEEE, 2012: 200-209.

[16] RUSHBY J. Noninterference, Transitivity and Channel-control Security Policies: Technical Report, CSL-92-02 [R]. Menlo Park: Stanford Research Institute, 1992.

(上接第 89 页)

[7] LIU X, WANG Q Y, JIN X L. An Energy-Aware Data Gathering and Routing Protocol for WSN[J]. Journal of Computer Research and Development, 2008, 45(1): 83-89. (in Chinese)
刘昕，王全玉，金旭亮. 基于能量感知的数据汇聚和路由协议[J]. 计算机研究与发展，2008，45(1)：83-89.

[8] YUE J, ZHANG W M, XIAO W D, et al. Structure-free and dynamic-adaptive data fusion algorithm for wireless sensor networks[J]. Journal of China Institute of Communications, 2012, 33(9): 53-65. (in Chinese)
乐俊，张维明，肖卫东，等. 无结构动态适应无线传感器网络数据融合算法[J]. 通信学报，2012，33(9)：53-65.

[9] YUE J, ZHANG W M, XIAO W D, et al. A Clustering Data Fusion Algorithm Based on Unequal Division for Wireless Sensor Networks[J]. Journal of Computer Research and Development, 2011, 48(1): 247-254. (in Chinese)
乐俊，张维明，肖卫东，等. 无线传感器网络中一种基于非均匀划分的分簇数据融合算法[J]. 计算机技术与发展，2011，48(1)：247-254.

[10] SUN Y Q, PENG J, LIU T, et al. Uneven clustering routing protocol based on dynamic partition for wireless sensor network [J]. Journal on Communications, 2014, 35(1): 198-206. (in Chinese)
孙彦清，彭舰，刘唐，等. 基于动态分区的无线传感器网络非均匀成簇路由协议[J]. 通信学报，2014，35(1)：198-206.

[11] LIU A F, YANG G J, CHEN Z G. Energy hole avoid by alternately working with different cluster-radius for wireless sensor networks[J]. Journal on Communications, 2010, 31(1): 1-8. (in Chinese)
刘安丰，阳国军，陈志刚. 基于不等簇半径轮换工作的传感器网络能量空洞避免研究[J]. 通信学报，2010，31(1)：1-8.