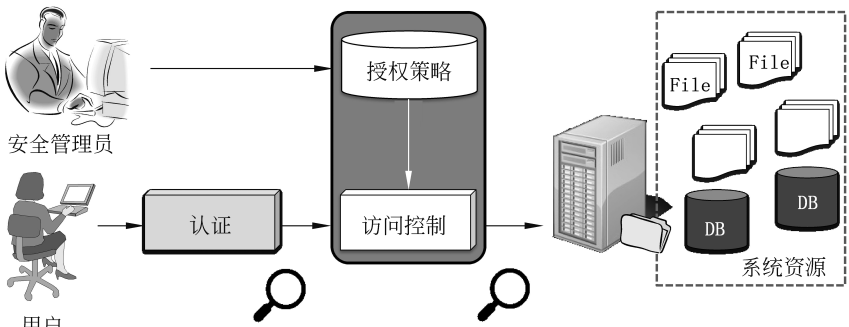
第5章 物联网安全基础——访问控制技术

访问控制的目的是确保系统资源访问的安全。在第4章中讨论了身份认证问题，即确认用户是否被允许接入系统、网络或服务。而访问控制机制明确了用户（主体）可以以何种方式访问何种系统资源（客体）。即，已经通过身份认证的用户能够以何种方式访问何种资源。本章主要涉及以下几方面：访问控制的基本概念、多级安全模型、物联网访问控制机制与其中的安全问题。

5.1 访问控制的基本概念

对于涉及系统资源访问的安全问题，往往使用术语“访问控制”表示。访问控制的主要目的是限制主体对客体的访问，从而保障数据资源在合法范围内得以有效使用和管理。为了达到上述目的，访问控制需要完成两个任务：一是识别和确认访问系统的用户，二是决定该用户可以对某一系统资源进行何种类型的访问。

如图5-1所示，在广义的访问控制框架下，存在着两个基本的研究领域，即身份认证与授权。如前所述，身份认证机制用于确认用户是否应该被允许对系统进行访问。授权则决定用户可以对某一系统资源进行何种类型的访问，即对已获得认证的用户的行为进行约束和限制[123]。例如，在系统中，仅允许类似管理员（administrator）等特权用户在系



认证：用户是否可以进入系统　授权：用户是否被允许进行该操作

图5-1　广义的访问控制框架

统中安装软件。身份认证是一个二值化的过程，即认证通过或拒绝；而授权则是更为细粒度的访问控制过程。在某些情景下，会把访问控制狭义地理解为授权过程。

5.1.1 访问控制的概念及要素

访问控制(access control)指系统对用户身份及其所属的预先定义的策略组限制使用数据资源能力的手段，通常用于系统管理员控制用户对服务器、目录、文件等网络资源的访问。访问控制是确保系统保密性、完整性、可用性的重要机制，是网络安全防范和资源保护的关键。

访问控制模型是从访问控制的角度出发，描述安全系统和建立安全模型的方法，需要使主体依据某些控制策略或权限对客体本身或其资源进行不同的授权访问。因此，访问控制模型通常包括主体、客体、访问控制策略 3 个要素[123]。

(1) 主体(Subject，S)。发起访问资源具体请求(如操作、存取)的实体，即活动对象，通常指某一用户，也可以是用户启动的进程、服务和设备等。

(2) 客体(Object，O)。被访问资源的实体，即必须进行控制的资源目标。所有可以被操作的信息、资源、对象都可以是客体。客体可以是信息、文件、记录、进程等，也可以是网络上的硬件设施、无线通信中的终端。

(3) 访问控制策略(Access Control Policy，A)。主体对客体的相关访问规则集合，用以确定一个主体对客体拥有何种访问能力，定义了主体与客体的可能作用方式。

访问控制即主体依据访问控制策略对客体本身或其资源进行不同权限的授权访问。

5.1.2 访问控制矩阵

访问控制矩阵是保障系统实现访问控制策略的经典模型，即将所有主体对于客体的权限存储在矩阵中。访问控制矩阵模型最早由 Bulter Lampson 于 1971 年提出，Graham 和 Denning 对该模型进行了修改。设主体的集合为 S，客体的集合为 O，主体对客体的权限类型集合为 R，在访问控制矩阵模型中，主体集合 S 和客体集合 O 之间的关系则用带有权限的矩阵 A 描述，A 中的任意元素 $a[s,o]$ 满足 $s\in S$，$o\in O$，$a[s,o]\subseteq R$。矩阵元素 $a[s,o]$ 表示主体 s 对客体 o 具有的权限。即，系统每一个主体对应访问控制矩阵中的一行(行索引)，客体则对应访问控制矩阵的一列(列索引)，主体 s 对客体 o 访问的许可(权限)即存储于矩阵中以 s 为索引的行与以 o 为索引的列相交的位置。表 5-1 给出了一个访问控制矩阵的例子，其中，与 UNIX 操作系统类似，x、r 和 w 分别代表执行、读和写的权限。在表 5-1 中，记账程序既被当作一个客体，又被当作一个主体。

表 5-1 访问控制矩阵

主体	客体				
	OS	记账程序	财务数据	保险数据	工资数据
Bob	rx	rx	r	--	--
Alice	rx	rx	r	rw	rw

续表

主体	客　体				
	OS	记账程序	财务数据	保险数据	工资数据
Mike	rwx	rwx	r	rw	rw
记账程序	rx	rx	rw	rw	r

在实际系统中，基于访问控制矩阵的分割（存储）方式的不同，有两种授权实现方式，即访问控制列表（Access Control List，ACL）和能力列表（Capability list，C-list）。

- 访问控制列表。以列（客体）为索引存储访问控制矩阵，这些列构成访问控制列表。当客体被访问时，就会应用与之对应的列，进而检查确认是否允许主体的访问操作。例如，表 5-1 中与“保险数据”对应的访问控制列表为

  （Bob，--），（Alice，rw），（Mike，rw），（记账程序，rw）
- 能力列表：以行（主体）为索引存储访问控制矩阵，这些行构成能力列表。当主体尝试执行某个操作时，就会应用与其对应的行，进而检查是否允许对客体进行该访问操作。例如，表 5-1 中 Alice 的能力列表为

（OS，rx），（记账程序，rx），（财务数据，r），（保险数据，rw），（工资数据，rw）

5.1.3　访问控制的类型

在可信计算的评估准则中，访问控制被分为自主访问控制与强制访问控制两大类。自主访问控制取决于拥有者的判断力，基于用户身份，是应用最为广泛的一种类型。此外，近些年来，基于角色的访问控制得到了广泛的研究和应用。

目前，访问控制类型有 3 种：自主访问控制、强制访问控制和基于角色的访问控制。

1. 自主访问控制

自主访问控制（Discretionary Access Control，DAC）模型是根据自主访问控制策略建立的一种模型，允许合法用户以用户或用户组的身份访问策略规定的客体，同时阻止非授权用户访问客体，某些用户还可以自主地把自己可以访问的客体的访问权限授予其他用户。自主访问控制又称为任意访问控制。Linux、UNIX、Windows 操作系统都提供自主访问控制的功能。

【定义 5.1】 如果个人用户可以设置访问控制机制允许或拒绝对客体的访问，那么这样的机制就称为自主访问控制，或称为基于身份的访问控制（Identity-Based Access Control，IBAC）。

自主访问控制的访问权限基于主体和客体的身份。客体通过允许特定的主体进行访问以限制对客体的访问。客体拥有者根据主体的身份规定限制，或者根据主体的拥有者规定限制。例如，如图 5-2 所示，Tom 有一本日记，他控制着对该日记的访问，他可以决定/设置谁能阅读（授予“读”权限）、谁不能阅读（拒绝“读”访问）。Tom 允许妈妈读，但其他人（如教师）不可以读。这是自主访问控制，因为对该日记的访问基于请求对客体（日记）的“读”访问的主体的身份（妈妈）。

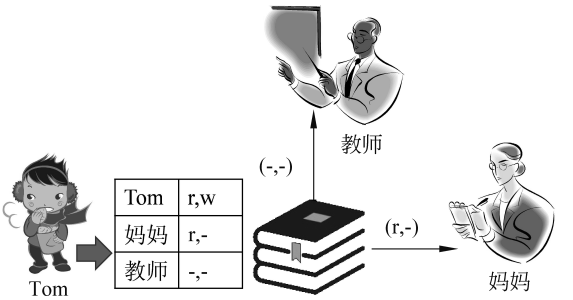


图 5-2　自主访问控制示例

在实现上，首先要对用户的身份进行鉴别，然后就可以按照访问控制列表赋予用户的权限允许或限制用户使用客体的资源。访问控制列表利用在客体上附加一个主体明细表以表示访问控制矩阵，主体明细表中的每一项包括主体的身份以及对客体的访问权限。主体访问权限的修改通常由特权用户（管理员）或特权用户组实现。然而，由于用户可以任意传递权限，那么没有访问文件 File1 权限的用户 A 就能够从有访问权限的用户 B 那里得到访问权限或是直接获得文件 File1。因此，DAC 模型提供的安全防护存在局限，不能给系统提供充分的数据保护。

2. 强制访问控制

强制访问控制（Mandatory Access Control，MAC）是比自主访问控制更为严格的访问控制策略，得到了广泛的商业关注和应用。在强制访问控制中，用户和客体都被赋予一定的安全级别，用户不能改变自身和客体的安全级别，只有管理员才能够确定用户和用户组的访问权限。

【定义 5.2】 通过系统机制控制对客体的访问，个人用户不能改变控制策略，这样的访问控制称为强制访问控制。

操作系统实施强制访问控制。主体和客体的拥有者都不能决定访问权限。通常，系统通过检查主体与客体的相关信息决定主体能否访问客体。规则描述允许访问的条件。强制访问控制有时也称为基于规则的访问控制。

例如，法律允许法庭在未经驾车人允许的情况下查看其驾驶记录。这是强制访问控制，因为驾驶记录的拥有者（驾车人）不能决定法庭对该信息的访问权限。

和自主访问控制不同的是，强制访问控制是一种多级访问控制策略（在 5.2 节会有详细讨论），它的主要特点是系统对访问主体和受控对象实行强制访问控制，系统事先给访问主体和受控对象分配不同的安全级别属性。在实施访问控制时，系统先对访问主体和受控对象的安全级别属性进行比较，再决定访问主体能否访问该受控对象。

3. 基于角色的访问控制

在现实中，访问信息的能力（权限）往往要根据一个人的工作性质确定。例如，Alice 是计算机系的会计，负责管理和记录系里的账目，因此她有权访问系里所有的账目。此后，Alice 调往学校财务处担任会计，因为她不再担任计算机系的会计，她也就不再拥有访问该系账目的权限。当系里委任 Bob 为新会计时，Bob 即获得了访问该系账目的权限。

即，访问账户的权限与会计的工作性质相关，而并非与特定个人（身份）相关。

【定义 5.3】 角色是指工作内容的集合。每个角色被授权执行一个或多个事务（即支持某项工作的活动）。角色 r 的授权事务集合记为 trans(r)。

角色(role)是指一个可以完成一定事务的命名组，不同的角色通过执行不同的事务完成各自的功能。事务(transaction)是指一个完成一定功能的过程，可以是一个程序或程序的一部分。角色是代表具有某种能力或某些属性的人的抽象概念，角色和组的主要区别在于：用户属于一个组是相对固定的，而用户能被指派为哪些角色则受时间、地点、事件等诸多因素影响。

【定义 5.4】 主体 s 的活动角色是指 s 当前所承担的角色，记为 actr(s)。

【定义 5.5】 主体 s 的授权角色是指 s 被授权承担的角色集合，记为 authr(s)。

【定义 5.6】 谓词断言 canexec(s,t)为真，当且仅当主体 s 在当前时间可以执行事务 t。

以下 3 条规则反映了主体执行事务的能力。

【规则 5.1】 设 S 为主体集合，T 为事务集合，角色指派规则是$(\forall s\in S)(\forall t\in T)[\text{canexec}(s,t)\rightarrow \text{actr}(s)\neq\varnothing]$。

该规则表示：如果一个主体可以执行某个事务，那么这个主体就有一个活动角色。该规则将事务执行与角色绑定，而不是与用户绑定。

【规则 5.2】 设 S 为主体集合，角色认证规则是$(\forall s\in S)[\text{actr}(s)\subseteq\text{authr}(s)]$。

该规则意味着主体必须得到授权以承担其活动角色。主体不能承担未经授权的角色。如果没有这个规则，则任意主体可以承担任意角色，从而就可以执行任意事务了。

【规则 5.3】 设 S 为主体集合，T 为事务集合，事务授权规则是$(\forall s\in S)(\forall t\in T)[\text{canexec}(s,t)\rightarrow t\in\text{trans}(\text{actr}(s))]$。

该规则的含义是一个主体不能执行当前角色没有被授权的事务。

基于角色的访问控制(Role-based Access Control，RBAC)的基本思想是：将访问许可权分配给一定的角色，用户通过承担不同的角色获得角色所拥有的访问许可权。这是因为在很多实际应用中，用户并不是可以访问的客体的拥有者。基于角色的访问控制从控制主体的角度出发，根据管理中相对稳定的职权和责任划分角色，将访问权限与角色相联系，这一点与传统的自主访问控制和强制访问控制将权限直接授予用户的方式不同，而是通过给用户分配合适的角色，让用户与访问权限相联系。角色成为访问控制中访问主体和受控对象之间的桥梁。系统管理员负责授予用户各种角色的成员资格或撤销某用户承担的某个角色。

基于角色的访问控制中通常定义不同的约束规则对模型中的各种关系进行限制，最基本的约束是“相互排斥”约束和“基本限制”约束，分别规定了模型中的互斥角色和一个角色可被分配的最大用户数。基于角色的访问控制中引进了角色的概念，用角色表示访问主体具有的职权和责任，灵活地表达和实现了安全策略，简化了访问权限的管理，解决了用户数量多、变动频繁的问题。

5.2 多级安全模型

本节将重点介绍多级安全模型的基本概念和最著名的两类多级安全模型——BLP模型和Biba模型。有关多级安全模型更为全面的介绍请参看相关文献。多级安全模型的目标在于建立带有安全等级的保护框架。

5.2.1 安全标记

在多级安全模型中，主体是用户，客体是被保护的数据。多级安全模型在将权限空间映射至主体的同时，也将分级机制映射于客体。通常使用四层分级机制(Security Classification，安全类别)和权限空间(Security Clearence，安全许可)：

Top Secret＞Secret＞Confidential＞ Unclassified

通过这种安全标记构成了偏序关系，可以进一步描述主体对客体的访问方式。设 o 是一个客体，s 是一个主体。那么 o 拥有一个安全类别，而 s 则会有一个安全许可。客体的安全等级标记为 $L(o)$，主体的安全等级标记为 $L(s)$。

当主体 s 的安全等级为 Top Secret，而客体 o 的安全等级为 Secret 时，用偏序关系可以表述为 $L(s) \geq L(o)$。考虑到偏序关系，主体对客体的访问主要有 4 种方式：

(1) 向下读(read down，rd)。主体的安全等级高于客体的安全等级时允许的读(r)操作。

(2) 向上读(read up，ru)。主体的安全等级低于客体的安全等级时允许的读(r)操作。

(3) 向下写(write down，wd)。主体的安全等级高于客体的安全等级时允许执行(e)的动作或写(w)操作。

(4) 向上写(write up，wu)。主体的安全等级低于客体的安全等级时允许执行(e)的动作或写(w)操作。

由于多级安全模型通过分级的安全标签实现了信息的单向流通，因此它一直被军方采用，其中最著名的是 Bell-LaPadula 模型(以下简称 BLP 模型)和 Biba 模型：BLP 模型具有只允许向下读、向上写的特点，可以有效地防止机密信息向下级泄露；Biba 模型则具有不允许向下读、向上写的特点，可以有效地保护数据的完整性。

5.2.2 机密性模型(BLP 模型)

BLP 模型是典型的机密性多级安全模型，主要应用于军事系统。BLP 模型通常是多级安全信息系统的设计基础，客体在处理绝密级数据和秘密级数据时，要防止处理绝密级数据的程序把信息泄露给处理秘密级数据的程序。BLP 模型的出发点是维护系统的机密性，有效地防止信息泄露。BLP 模型的目标是实现所有多级安全系统都必须满足的关于机密性的最低需求。表 5-2 是安全等级示例。

表 5-2　安全等级示例

安 全 等 级	安 全 许 可	安 全 类 别
Top Secret(TS)绝密	Elaine，Tomas	人事文件
Secret(S)机密	Alex，Samuel	电子邮件文件
Confidential(C)秘密	Lawrence，Clarence	活动日志文件
Unclassified(UC)公开	Mike，Sammy	电话清单文件

BLP 模型首先声明如下简单安全条件：

当且仅当 $L(s) \geqslant L(o)$ 时，主体 s 能够对客体 o 执行读操作

如表 5-2 所示，第一列为安全等级，是基本的机密性分类系统；第二列是主体根据安全许可分组的结果；第三列是文档根据安全类别分组的结果。依据表 5-1 的安全条件，Lawrence 不能读人事文件，但 Tomas 和 Alex 可以读活动日志文件。如果 Tomas 将人事文件的内容复制到活动日志文件中，则 Lawrence 即可读取到人事文件了。BLP 模型的第二条声明——* 特性(星特性)：

当且仅当 $L(s) \leqslant L(o)$ 时，主体 s 能够对客体 o 执行写操作

防止了这种情况的产生。

BLP 模型可以有效防止低级用户和进程访问安全等级更高的信息资源。此外，安全等级高的用户和进程也不能向安全等级低的用户和进程写入数据。上述访问控制原则可以表述为“无上读、无下写”。

BLP 模型的安全策略包括强制访问控制和自主访问控制两部分。强制访问控制中的安全特性要求：给定安全等级的主体仅被允许对同一安全等级和较低安全等级的客体进行读操作；给定安全等级的主体仅被允许向相同安全等级或较高安全等级的客体进行写操作。自主访问控制允许用户自行定义是否让其他用户或用户组访问数据。

5.2.3　完整性模型(Biba 模型)

与针对机密性的 BLP 模型不同，Biba 模型针对的是完整性。实际上，Biba 模型本质上是 BLP 模型在数学上的对偶。如果信任客体 o_1 的完整性，但不信任客体 o_2 的完整性，那么就不能信任由 o_1 和 o_2 组成的客体 o 的完整性。换言之，客体 o 的完整性等级是包含于 o 中的所有客体的完整性等级中最低的那个，即对完整性遵守低水印(low-watermark)原则。

与 BLP 模型的信息机密性相应，Biba 模型定义了信息完整性等级。设 $I(o)$ 表示客体 o 的完整性等级，$I(s)$ 表示主体 s 的完整性等级，Biba 模型由下面两个声明定义：

(1) 写访问规则。当且仅当 $I(o) \leqslant I(s)$ 时，主体 s 能够对客体 o 执行写操作。

(2) 读访问规则。当且仅当 $I(s) \leqslant I(o)$ 时，主体 s 能够对客体 o 执行读操作。

写访问规则表明，对主体 s 所写的内容的信任程度不应超过对 s 自身的信任程度。读访问规则表明，对主体 s 的信任程度不应超过 s 所读取的客体的最低完整性等级。从本质上说，为了避免 s 被完整性等级更低的客体“弄脏”，要禁止 s 查看这样的客体。

如图 5-3 所示，假设系统中的主体集合 S 包含 3 个进程，即 $S=\{\text{Process1},\text{Process2},\text{Process3}\}$；客体集合 O 包含 3 个文件，即 $O=\{$电子邮件文件(Email)，活动日志文件(Log)，电话清单文件(Tel)$\}$；系统的完整性等级 $I=\{$高，中，低$\}$；系统操作分别为读(r)、写(w)。可见，根据读访问规则，Process1 完整性等级($I(s)$)为高，该进程可以读取电子邮件文件($I(o)$为高)，但不可以读取活动日志文件($I(o)$为中)和电话清单文件($I(o)$为低)。根据写访问规则，Process1 可以写 Email、Log、Tel 这 3 个文件。Process2 可以读 Email、Log 文件，但不能读 Tel 文件；可以写 Log、Tel 文件，但不能写 Email 文件。

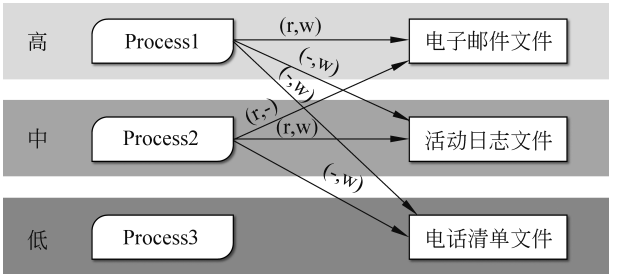


图 5-3　主客体完整性等级示例

在信息流向的定义方面，Biba 模型不允许信息从完整性等级低的进程流向完整性等级高的进程，也就是说，用户只能向比自己完整性等级低的客体写入信息，从而防止非法用户创建完整性等级高的客体信息，避免越权、篡改等行为的出现。Biba 模型可同时针对有层次的安全等级和无层次的安全类别。

Biba 模型的主要特征是“无下读、无上写”。这样就使得完整性等级高的文件一定是由完整性等级高的进程所产生的，从而保证了完整性等级高的文件不会被完整性等级低的文件或进程中的信息所覆盖。

Biba 模型在应用上限制性很强，因为其要求杜绝主体 s 查看完整性等级低的客体。在很多场景下，采用以下低水印原则有助于完整性策略的实施：

如果主体 s 能够对客体 o 执行读操作，那么 $I(s)=\min(I(s),I(o))$

在低水印原则下，主体 s 可以读取任何内容，但是在访问一个完整性等级较低的客体之后，主体 s 的完整性等级也会降低。由图 5-4 可见 BLP 模型和 Biba 模型之间的差异：BLP 模型用于机密性保护，隐含遵从高水印原则；而 Biba 模型用于完整性保护，隐含遵从低水印原则。

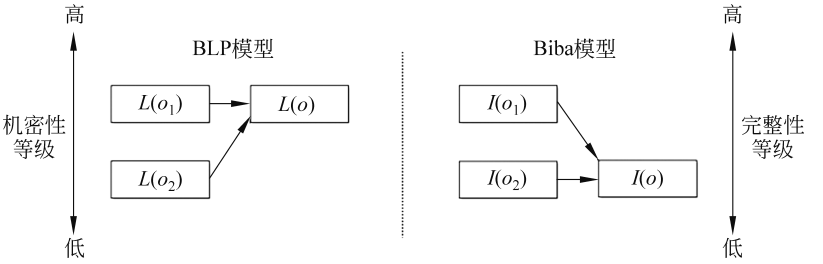


图 5-4　BLP 模型与 Biba 模型

5.3 物联网访问控制机制

5.3.1 物联网访问控制机制的实施

访问控制是主体向客体执行访问操作时的授权管理机制。在物联网场景中，访问控制的主体通常包括用户、物联网应用程序和第三方服务，客体主要是物联网设备。在物联网中，访问控制机制的实施常见于网络层与应用层。

1. 物联网网络层访问控制机制

在物联网网络层中，主要通过各类通信协议自身的访问控制机制实现对物联网设备信息（即物理感知状态）的访问控制。下面以 ZigBee 与 MQTT 协议为例，分别介绍物联网设备与网关之间以及网关与应用层之间的访问控制机制。

ZigBee 协议是一个基于 IEEE 802.15.4 标准的低功耗局域网协议，是一种短距离、低功耗的无线通信技术。该协议利用 2.4GHz 频段，最大传输速率为 250kb/s，传输距离为 10～100m。IEEE 802.15.4 协议有 3 种可选的安全特征：数据帧加密、完整性保护与访问控制。结合上述特征，ZigBee 提供了非安全模式、访问控制模式与安全模式[125]。在访问控制模式中，网络中的每一个节点均管理并维护一个访问控制列表（ACL），用于确定其余节点（即邻居节点）是否可信。IEEE 802.15.4 协议支持至多 255 个 ACL 条目，每个 ACL 条目包含邻居节点的地址、节点使用的安全策略、密钥以及最后使用的初始向量等信息。当某一节点接收到来自邻居节点的数据包时，该节点查询对应的 ACL 条目。当确定邻居节点可信时，就根据条目信息进行安全可信通信；否则通信被拒绝或者调用认证功能[126]。

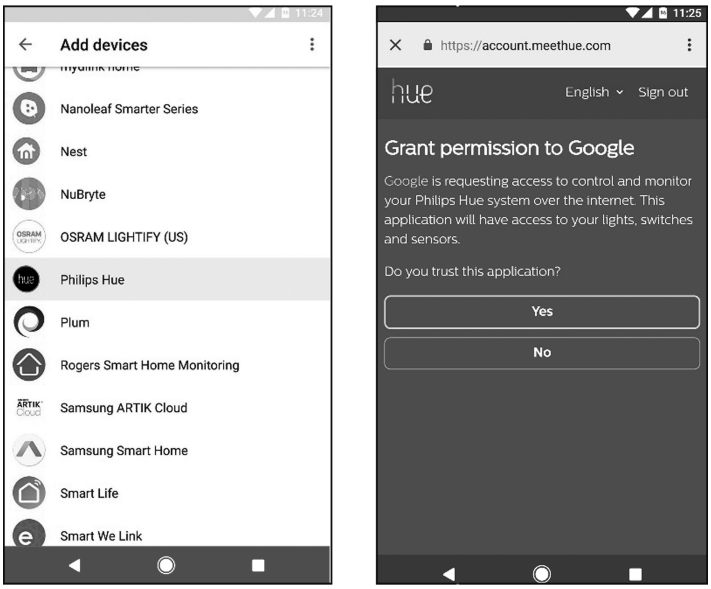
消息队列遥测传输（MQTT）是 ISO 标准（ISO/IEC PRF 20922）下基于发布/订阅模式的消息协议。它工作在 TCP/IP 协议族上，是为低带宽环境中的机器间（Machine-to-Machine，M2M）遥测而设计的。相较于 HTTP，MQTT 能节省更多的资源，带来较少的传输负担，因此在物联网设备中广泛应用。作为发布/订阅协议，MQTT 允许客户端写（即协议中的发布操作）和读（即协议中的订阅操作）主题。然而，并非所有的客户端都有权读、写所有主题。用户可以在 MQTT 代理上配置访问控制列表，限制不同客户端对主题的权限。在 MQTT 的访问控制列表文件中，用户可以分别为所有客户端、指定用户名的客户端以及指定 ID 的客户端配置对于不同主题的访问控制权限[127]。

2. 物联网应用层访问控制机制

在应用层中，现有物联网平台基本均允许用户自行配置不同用户对物联网设备的访问控制权限。例如，苹果公司的 HomeKit 平台允许用户在 Home App 上邀请其余 iCloud 账户成为家庭成员，并管理受邀用户对智能物联网设备的（远程）访问和编辑权限[128]。

当考虑到第三方服务（如 Amazon Alexa[129]、Google Assistant[130]、IFTTT[131]等）对物联网设备的访问控制时，通常由用户在第三方服务平台上向物联网平台发起授权请求，

并由物联网平台同意后获得相应物联网设备的访问权限。例如，当用户想要在 Google 平台上获取飞利浦 Hue 设备的访问权限时，就会在 Google 平台上向飞利浦 Hue 平台发起授权请求(图 5-5(a))，并由用户在飞利浦 Hue 平台上确认该请求(图 5-5(b))。



(a) (b)

图 5-5 物联网中第三方服务平台的请求授权操作示例

在现实中，物联网平台常使用单点登录标准协议 OAuth 2.0 进行授权。OAuth 意为 Open Authorization(开放授权)。OAuth 2.0 是一种应用层授权协议，旨在允许网站或其他应用程序代表用户访问由其他 Web 应用程序托管的资源。而当物联网设备通过网络层连接到物联网平台(通常在云上)时，这些设备就成为该平台拥有的资源，其访问权限可通过 OAuth 2.0 标准开放给其余服务平台。OAuth 2.0 标准中存在以下角色[132]：

(1) 资源所有者(resource owner)。可以授予受保护资源权限的实体，通常为终端用户。

(2) 资源服务器(resource server)。保存着受保护资源的服务器。

(3) 客户端(client)。代表资源所有者请求访问某一资源的应用程序。

(4) 授权服务器(authorization server)。在认证资源所有者身份并获取其同意授权后，颁发访问令牌给客户端。

在物联网场景中，资源所有者为用户，客户端为第三方服务平台，而资源服务器和授权服务器的功能通常均由物联网平台承担。因此，第三方服务平台通过 OAuth 2.0 标准获取物联网平台授权的过程可以表示为图 5-6。

具体来说，该过程包括用户对服务的授权及服务对服务的授权两个步骤[133]：

(1) 用户对服务的授权。用户在第三方服务平台中发起对物联网平台的单点登录流程(图 5-6 中的①和②)，并在物联网平台上同意对第三方服务平台的授权(图 5-6 中的③)。

(2) 服务对服务的授权。物联网平台向第三方服务平台颁发一个访问令牌(token)