第3章

访问控制和身份认证

访问控制和身份认证是保证物联网安全的两大重要技术。访问控制是授权用户访问物联网资源的过程，而身份认证是验证物联网用户证书真实性的过程。

本章从基本概念、实现原理和常用技术3方面分别介绍访问控制和身份认证相关机制。特别地，在访问控制方面，还介绍各种访问控制机制在物联网的实际应用及其优缺点；在身份认证方面，进一步介绍身份认证技术在物联网中的实际应用。

3.1 访问控制

访问控制是物联网的安全基础之一，它在物联网设备资源授权和信息保护方面尤为重要，用在某一方授权另一方（例如，用户）在何种条件下可以访问哪些资源或信息的场景中。传统的访问控制机制如访问控制列表（Access Control List，ACL）、基于角色的访问控制（Role-Based Access Control，RBAC）协议和基于属性的访问控制（Attribute-Based Access Control，ABAC）协议已经被广泛应用在物联网中。目前，为适应物联网环境下设备多源异构、数据体量大以及设备计算和存储能力有限的场景特性，许多新型、灵活、轻量级且扩展性强的访问控制机制相继被提出，如基于区块链技术的访问控制协议FairAccess、BlendCAC和ControlChain。

3.1.1 访问控制的基本概念

广义上讲，访问控制的定义包含认证（Authentication）、授权（Authorization）和审计（Auditing）3个属性（即AAA标准），工业上许多安全的访问控制协议都是基于AAA标准来设计的，如Diameter基础协议。首先，协议认证用户的身份，通常的方法有用户名和密码认证、智能卡认证和生物特征认证等；其次，对认证成功的用户，授权的一方决定哪些用户在什么条件下可以操作哪些资源对象，例如，授权用户名为U的用户只能通过Telnet访问服务器S；最后，用户的访问历史可以被审计，例如，用户名为U的用户在2019年3月11日通过Telnet访问了服务器S 10分钟。从狭义上讲，访问控制是决定授权某一方可以基于某些安

全模型和策略读取或修改某些资源对象的过程，即上述的第二个属性。通常，一个有效安全的访问控制机制需要满足 3 个特性：机密性(Confidentiality)、完整性(Integrity)和可用性(Availability)。机密性指资源对象不允许泄露给任何未被授权访问的一方；完整性指资源对象不可被任何未被授权访问的一方篡改；可用性是指任何被授权的一方可以正确地访问到相应的资源对象。然而，在物联网时代，新型的访问控制机制应该满足高延展性、灵活性和轻量级的需求。

3.1.2　访问控制的实现原理

总体来说，实现访问控制的过程包括定义访问控制策略(或规则)，基于定义的策略建立访问控制模型，基于建立的模型设计访问控制机制，基于设计的机制实现访问控制。例如，当对一个系统实现安全的访问控制机制时，可以先使用威胁诊断和漏洞评估方法制定系统的安全访问控制策略，然后运用基于角色的访问控制方法对制定好的策略进行访问控制建模，再基于访问控制标记语言 XACML 标准和 OAuth 2.0 框架(见 3.1.4 节)设计对应的访问控制机制，最后使用一些硬件或软件工具(如访问控制列表、加密算法、智能卡等)实现访问控制。

定义访问控制策略时需要明确被访问的对象(称为主体)、请求访问的对象(称为客体)以及主体和客体之间的交互活动。例如，在操作系统的保护中，主体可以是某些特定的进程，客体可以是这些进程的父进程，客体可以向操作系统发出请求中断哪些子进程。确定主体和客体之间的交互活动时，依据系统的威胁和漏洞评估结果，制定安全的针对对象之间活动的访问控制策略。常见的评估方法有 ISO/IEC 27002、ISO/IEC 27005 标准、OTACE EBIOS 规则，以及 MAHARI、CRAMM 和 OWASP 标准等。

在现实情况下，定义好的访问策略往往偏向于人类可读的语言。因此，计算机系统要识别策略存在语义的鸿沟，而建立合理的访问控制模型在这之间就起到了桥梁的作用。访问控制模型，也称授权模型，可以将访问策略形式化为机器可读的语言。目前，存在许多访问控制模型，如自主访问控制(Discretionary Access Control，DAC)模型、强制访问控制(Mandatory Access Control，MAC)模型、基于角色的访问控制(RBAC)模型、基于属性的访问控制(ABAC)模型和使用控制模型(Usage Control Model，UCM)。对一个系统的访问策略进行建模时，可以使用多种访问控制模型完成建模，这往往适用于异构的系统环境。

建立好合理的访问控制模型后，依据 ISO 的访问控制标准 ISO/IEC 10181-3，使用访问控制标记语言 XACML 和 OAuth 2.0 框架设计协议，实现基于对象的访问控制和审核策略，如引用监视器。在 ISO/IEC 10181-3 标准下，引用监视器除了主体和客体对象外，还包含另外两个实体：访问控制实施(Access Enforcement Facility，AEF)部件和访问决策实施(Access Decision Facility，ADF)部件。访问控制实施部件也称策略实施部件(Policy Enforcement Point，PEP)，访问决策实施部件也称策略决策部件(Policy Decision Point，PDP)。在协议设计中，请求者的访问请求先由 AEF 拦截解析，再转发给 ADF 根据制定的访问策略做出访问控制决策，选择接受或拒绝请求者的访问请求。

最后，应用软硬件结合的工具实现访问控制协议，例如，如何执行策略，如何依据访问

策略评估访问请求，检测策略执行时的异常情况以及验证策略的正确性等。除了上文提到的访问控制列表、加密算法、智能卡外，还有审计日志软件、防火墙和报警器。

3.1.3 访问控制的基本模型

3.1.2 节主要介绍了实现访问控制过程的基本原理，本节将具体解释常用于实现访问控制的技术，它们包括访问控制模型、标准协议和框架。

1. 自主访问控制模型

自主访问控制模型根据客体的身份和访问规则控制资源如何被访问，说明允许（或不允许）客体对哪些主体执行哪些操作。它通常包含了一个决定授权管理访问控制规则的管理员策略。自主访问控制模型之所以被称为是自主的，是因为用户（客体）可以将其权限传递给其他用户，但权限的授予和撤销由管理员策略决定。

访问矩阵（Access Matrix）是用于描述自主访问控制模型的常用方法，它将操作的授权表示为矩阵的形式，如表 3.1 所示。访问矩阵最早由 Lampson 提出，用于刻画对操作系统资源保护的策略，随后又被 Harrison 等人进一步形式化并对访问策略进行了复杂度分析。

表 3.1 访问矩阵

用　户	文　件		
	文 件 1	文 件 2	文 件 3
张三	读取，写入，执行	执行	NULL
李四	读取，写入	NULL	读取，执行

虽然访问矩阵能清晰地描述操作的授权情况，但由于它通常是一个庞大又稀疏的矩阵表格，应用在实际的系统中比较耗内存。因此，它衍生出了 3 种简洁实用的操作授权表示方式，分别为授权表（AL）、访问控制列表和基于权能的访问控制列表（CapBACL）。

如表 3.2 所示，授权表将访问矩阵中的非空元组转换为包含客体、授权操作和主体的表格。

表 3.2 授权表

客　体	授 权 操 作	主　体
张三	读取	文件 1
张三	写入	文件 1
张三	执行	文件 1
张三	执行	文件 2
李四	读取	文件 1
李四	写入	文件 1

续表

客　　体	授 权 操 作	主　　体
李四	读取	文件 3
李四	执行	文件 3

如图 3.1 所示，每个包含授权操作的元组以列的形式存储，并指定给客体，再将其关联到主体，来表达对于主体哪些客体具有哪些授权操作。

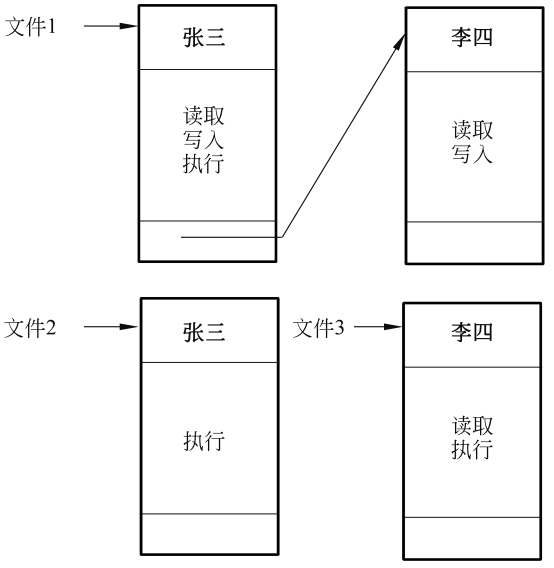

图 3.1　访问控制列表

如图 3.2 所示，每个针对特定主体的授权操作元组以行的形式存储，并被关联到客体，来表达哪些客体对哪些主体具有哪些授权操作。

总的来说，访问控制列表和基于权能的访问控制列表相比于访问矩阵和授权表能更灵活地对系统进行授权和管理。同时，它们也有各自的优势和缺点。具体来说，访问控制列表能支持迅速地查阅针对特定文件的授权情况，然而，当要查阅某一请求者对特定文件的操作授权情况时，则需要遍历所有的访问控制列表。对于基于权能的访问控制列表，它能迅速地定位特定请求者具有的所有操作授权情况，但当要查阅某一文件对于某一请求者的操作授权情况时，则需要遍历所有的基于权能的访问控制列表。

2. 强制访问控制模型

强制访问控制模型是一种基于中心化授权机构制定的访问控制规则强制执行访问策略的模型，最常见的模型为建立在对客体和主体分类上的多级安全策略模型，这种模型多应用在对数据库的多级别安全管理中。在客体的定义上，它与自主访问控制模型有点区别，在自主访问控制模型中将用户表示为客体，这里客体是指代表用户执行操作的进程，而用户就表示为用户，通过这样的表示方法，进程之间的访问和修改等授权操作也能被刻

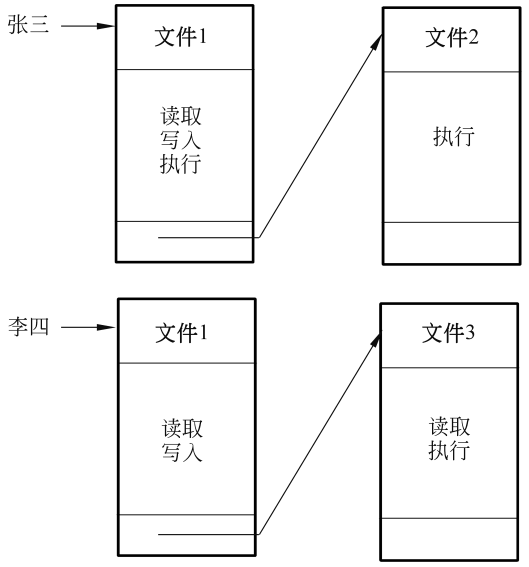

图 3.2　基于权能的访问控制列表

画出来。

在强制访问控制模型中，中心化的授权机构为每个主体和客体都赋值了一个访问类。该访问类是一个偏序集，偏序指定了集合之间的支配关系，用符号≥表示。访问类包含安全等级和类别两个元组。其中，安全等级包含具有层级关系的元素，从高到低排列包括绝密级别(T)、机密级别(S)、秘密级别(C)和公开级别(U)；类别则标识了功能和适用领域，如在军用系统中，类别可以为北约、核、军队等。

定义 3.1　给定一个安全等级全序集合 L 和一个类别集合 C，访问类定义为 L 和 $\Delta(C)$的笛卡儿积 AC$=L\times\Delta(C)$，其中 $\Delta(C)$为 C 的超集。$\forall c_1=(L_1,C_1),c_2=(L_2,C_2),c_1\geqslant c_2\rightarrow(L_1\geqslant L_2)\wedge(C_1\supseteq C_2)$。若 $c_1\geqslant c_2$ 不成立，$c_1\leqslant c_2$ 也不成立，则称 c_1 和 c_2 是不兼容的。

根据定义 3.1，在强制访问控制模型下，主体和客体之间的访问控制关系可以用格来描述。如图 3.3 表示了绝密等级和机密等级两种安全级别以及标明了核和军队两种类别的资源访问控制关系。如果一个用户拥有 TS 级别的安全许可和被允许访问标明“{军队}”类别的资源，则该用户可以访问 S 安全级别下且类别为“{军队}”的资源和 TS 安全级别下且无类别标签的资源。

3. 基于角色的访问控制模型

基于角色的访问控制模型是基于角色来控制用户访问资源的访问控制模型。根据定义 3.2，它包括用户、角色和权限。其中，权限表示对主体的操作。用户、角色和权限之间为多对多的关系，例如，一个用户可以有多个角色，一个角色可以有多种权限，同时，一个角色可以赋予多个用户，一种权限也可以赋予多个角色。如图 3.4 所示，它展示了医护系统中不同角色的用户对病历的操作权限。

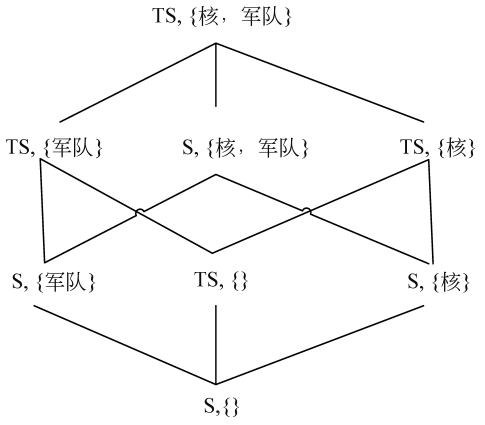

图 3.3　基于格表示的访问控制关系

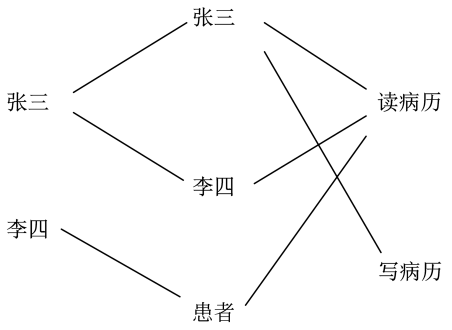

图 3.4　基于角色控制策略对病历的操作

定义 3.2　基于角色的访问控制模型由如下 6 部分组成。

(1) U,R,P,S 分别表示用户、角色、权限和会话集合。

(2) $P\times R\supseteq$PA,表示权限和角色的多对多关系。

(3) $U\times R\supseteq$UA,表示用户和角色的多对多关系。

(4) $f:s\to u$,$s\in S$,$u\in U$,实现将一个会话 s 映射到单个用户的函数 $f(s)$。

(5) $g:s\to 2^R$,实现将一个会话映射到一个角色集合的函数 $f(s)$。

(6) $\{r|(f(s),r)\in \text{UA},r\in R\}\supseteq g(s)$,且每个会话拥有的权限集合为 $\bigcup_{r\in g(s)}\{p|(p,r)\in \text{PA},p\in P\}$。

基于角色的访问控制模型可以简化对大量用户权限的管理,这是因为它依据用户之间需求的相似性将用户按角色分组,然后将权限分配给角色而不是用户。而且,通过对用户添加或撤销角色和对角色增添或撤销权限,大大降低对用户权限管理的开销。同时,它还具备其他优点:遵循“最小特权”原则实现对主体的管理,即将对某一主体的操作权限赋予属于相应角色的用户;实现了职责分离,通过分配给不同用户包含互斥权限的角色来避免对主体有冲突的操作。然而,基于角色的访问控制模型仅仅按角色这单一属性划分用户,现实世界中的用户还包含其他特征属性,因此,它不足以描述复杂的、细粒度的访问控制策略。下面介绍细粒度的基于属性的访问控制模型。

4. 基于属性的访问控制模型

基于属性的访问控制模型不同于基于角色的访问控制模型,考虑了主体(或资源)、客体和环境 3 方面中任何与安全访问相关的特征来定义权限,而不仅是角色这一特征,这些特征统一称为属性。下面从主体、客体和环境 3 方面描述各方的属性。

主体就是被客体访问的资源,可以是一个数据结构、系统模块或文档。文档的属性可以是标题、日期、作者等属性。客体可以指用户也可以是某个程序进程,是访问或操作主体的一方。而用户的属性可以是名字、所属机构、工作等,角色也可以当成一个属性。所以,基于属性的访问控制模型的功能其实涵盖了基于角色的访问控制模型的功能。除了

主体和客体的属性外，还有环境的属性，它指资源被访问的上下文信息，例如当前日期、当前时间、网络状况等。由于在语义上有如此丰富的属性描述，因此该模型能表达更复杂且更细粒度的访问控制策略。

基于属性的定义，该模型设计包含了策略模块和应用策略的访问控制架构模块。其中，策略模块如定义 3.3 所述，表示对一个客体赋值一种属性，例如，对客体赋予值为管理员的角色属性或表示对客体赋予值为 Alice 的名字属性。

定义 3.3 基于属性的访问控制模型的策略模块由以下 4 部分组成。

(1) s，r，e 分别表示客体、资源和环境。

(2) $SA_k(1\leqslant k\leqslant K)$表示客体的属性，$RA_m(1\leqslant m\leqslant M)$为资源的属性，$EA_n(1\leqslant n\leqslant N)$表示环境的属性，$K$，$M$，$N$ 都为正整数。

(3) ATTR(·)为属性赋值函数，如 ATTR(s)表示对客体进行 ATTR 属性的赋值，同时属性赋值函数存在以下关系：

$$\mathrm{ATTR}(s)\subseteq \mathrm{SA}_1\times \mathrm{SA}_2\times\cdots\times \mathrm{SA}_k$$

$$\mathrm{ATTR}(r)\subseteq \mathrm{SA}_1\times \mathrm{SA}_2\times\cdots\times \mathrm{SA}_m$$

$$\mathrm{ATTR}(e)\subseteq \mathrm{SA}_1\times \mathrm{SA}_2\times\cdots\times \mathrm{SA}_n$$

(4) $P(s,r,e)$表示客体在某特定环境下是否允许访问主体的策略；f 表示属性评估函数，返回值为真或假，$f(\mathrm{ATTR}(s),\mathrm{ATTR}(r),\mathrm{ATTR}(e))\rightarrow P(s,r,e)$。

应用定义好的访问控制策略需要属性机构、策略实施部件和策略决策部件的共同协作，这构成了该模型的架构模块。属性机构负责定义和管理属性，策略实施部件相当于 AEF，负责转发客体的访问请求给策略决策部件，同时策略决策部件相当于 ADF，它判断客体的请求是否被授权。

5. 使用控制模型

在基于属性的访问控制模型的基础上，使用控制模型支持动态更新控制策略中的属性。同时，它不但在用户访问资源之前制定访问策略，而且在用户访问资源的过程中也会根据更新的访问策略规定用户的访问权限。如果当前的访问策略不允许用户访问时，则立即终止用户的使用，这一特性称为访问策略的持续性。因此，使用控制模型更适合受保护的资源动态变化的应用环境。

通常，一个使用控制模型由 12 个原语来描述，除了前面提到的客体、客体属性、资源、资源属性、环境属性之外，还包括权限、基于属性构成的授权、职能及条件策略及用于描述策略动态更新的原语，即包括状态、状态转换操作和属性更新操作。

其中，基于属性构成的授权策略、职能策略和条件策略是一条访问策略的重要组成部分。授权策略是根据用户的属性和资源的属性制定的策略，例如名字属性为 Alice 的用户可以访问标题属性为 A 的文件资源。职能策略是用户在访问资源之前或过程中或之后一定要执行的操作，例如角色属性为“专家”的用户审阅申请文件之前一定要签署保密合约。条件策略是指强加在被访问资源之上的条件约束，例如一个文件资源只被公开访问 1 小时。

另一方面，状态、状态转换操作和属性更新操作这 3 个原语则被用来描述使用控制模

型的状态转换模型，体现访问策略的动态变化和持续性。如图 3.5 所示，状态可以包含“初始”请求中、访问中、已拒绝、已撤销和结束 6 个状态；状态转换操作使一个状态转换到下一个状态，如“尝试访问”操作将“初始”状态转换到“请求中”状态，“拒绝访问”操作将“请求中”状态转换到“已拒绝”状态；属性更新操作可以发生在每个状态，操作的结果返回真或假，例如准备更新操作可以发生在“请求中”状态，正在更新可以发生在“访问中”状态。

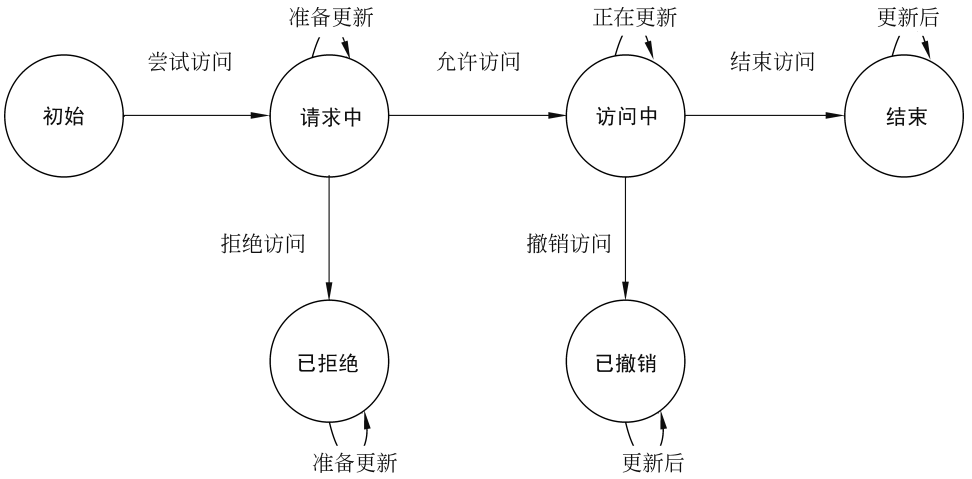

图 3.5　状态转换模型

基于上述 12 个原语，使用控制模型可以用定义 3.4 进行形式化地定义。根据定义 3.4，一条基于使用控制模型制定的访问策略是一个五元组，每个元素为每个组成部分的子集。

定义 3.4　使用控制模型包括 5 个组成部分 $M=(T,P_A,P_C,A_A,A_B)$。

(1) T 表示系统状态集合。

(2) P_A 表示基于客体属性和资源属性构成的授权有限集。

(3) P_C 表示基于环境属性构成的条件有限集。

(4) A_A 表示属性更新操作有限集。

(5) A_B 表示职能有限集。

每部分基于预定义的客体集合 S、资源集合 R、权限集合 P 和属性集合 A 构成。

3.1.4　访问控制的常用技术

1. XACML

XACML 是一种辅助实现访问控制的可扩展标记语言，在 2003 年被 OASIS(结构化信息标准促进组织)批准为标准通用的标记语言，已经被广泛地应用在分布式环境下的访问控制和授权过程。

图 3.6 展示了 XACML 在访问控制实现过程中的作用。XACML 不但定义了访问策略的语法结构和执行访问策略的语义结构，而且也规定了获取访问策略的访问请求格式

和访问响应格式。访问请求和访问响应的格式在策略实施部件和策略决策部件交互的消息中得以体现。一个 XACML 请求在语法结构上由任意大小的树结构组成，在语义上这些子树表达了访问策略的制定标准，子树的叶子规定了最小单元的策略规则，由于这些策略规则的存在，这些子树可以表达很丰富且很复杂的访问策略逻辑。XACML 请求通常包含了主体、与主体相关的属性、对主体的操作等。而 XACML 响应可以包含以下 4 种决策中的一个：允许、拒绝、请求无效和中断。通常情况下，返回“请求无效”或“中断”时，响应内容也包含了与决策相关的解释。同时，XACML 响应也会包含指示策略实施部件执行决策的指令。关于 XACML 的具体语法和语义结构可参见文献。

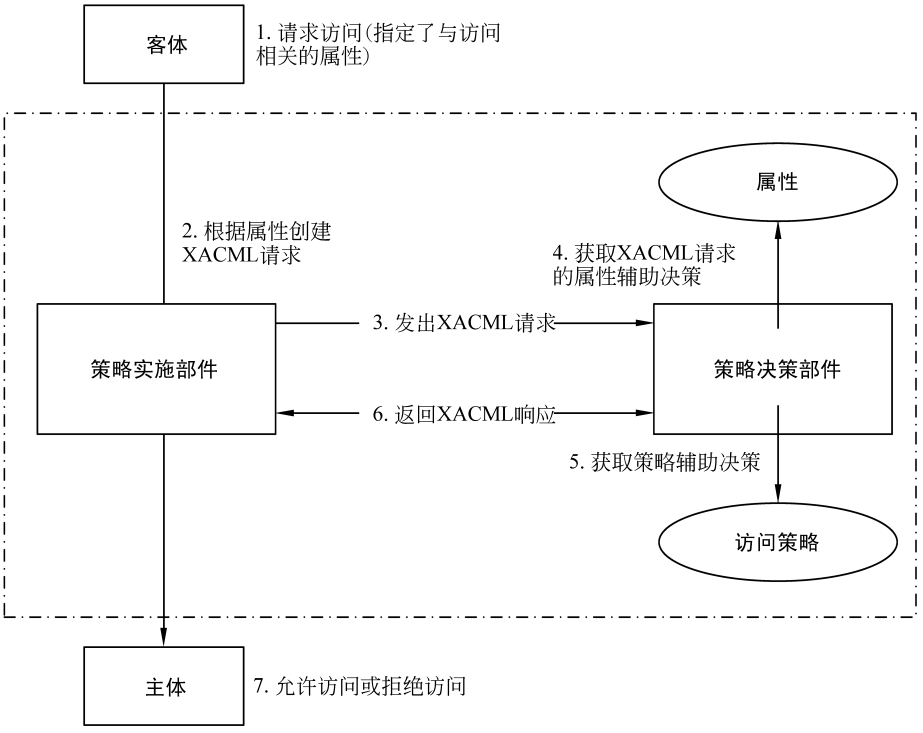

图 3.6　基于 XACML 访问控制概况图

2. OAuth 2.0 协议

OAuth 2.0 协议是一种开放授权标准协议，它允许用户在不需要提供账户和密码的情况下授权第三方应用获得访问服务器上用户资源的权限。由于 OAuth 1.0 协议为了保证安全性而在授权交互的过程中加入了签名和加密的步骤，并且要求交互双方管理状态和临时密钥，这使授权实现起来过于复杂，不能带来良好的用户体验。于是，OAuth 2.0 协议结合 HTTPS 来实现认证，替换了 OAuth 1.0 协议复杂的签名和加密步骤，从而更加轻便，带来更好的用户体验。目前，大多数互联网公司如谷歌、亚马逊、微软、腾讯等都是使用 OAuth 2.0 协议来实现应用的第三方授权。

协议定义了 4 个角色，分别为资源拥有者、资源服务器、客户端和授权服务器。资源

拥有者，也可以说是用户，负责协作授权服务器授予访问资源的权限（通常称为 Token）；资源服务器存有用户的资源，根据访问者的 Token 响应访问请求。客户端，通常称为第三方应用程序，可以代表资源拥有者，它在得到资源拥有者授权的情况下请求资源服务器。授权服务器负责认证客户端，验证客户端是否得到资源拥有者的授权，验证成功会授予客户端一个访问 Token。

图 3.7 展示了 OAuth 2.0 协议的基本工作流。①客户端向资源拥有者请求授权；②客户端得到资源拥有者授权的证书；③客户端向授权服务器请求访问 Token；④授权服务器对客户端的身份和证书认证成功后，发送 Token 给客户端；⑤客户端使用 Token 向资源服务器请求访问；⑥资源服务器验证 Token 的有效性之后接受客户端的请求。

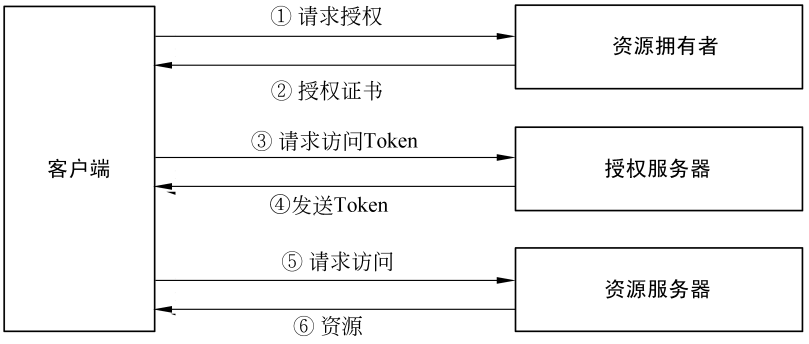

图 3.7　OAuth 2.0 协议的基本工作流

3.1.5　适用于物联网中的访问控制模型

在现实生活中，从小范围的可穿戴、家居设备到覆盖面更广的社区设备和工业设备，物联网技术服务于人们生活的方方面面。为了提高物联网技术的可靠性和安全性，针对不同场景下的物联网设备特点和安全需求，实现合适的安全访问控制技术已经成为重点研究话题之一。由于物联网中设备多种多样、设备通信架构和协议异构不统一及通信带宽资源有限等特性，3.1.4 节介绍的传统的访问控制技术并不能直接被应用到物联网环境中，例如，自主访问控制模型和强制访问控制模型适合用于保护封闭系统中的数据，不适合开放的物联网环境。面对物联网环境下的安全需求，有些研究者应用传统的访问控制模型，如 RBAC、ABAC 和 UCON 模型，提出管理物联网设备资源访问的方案，也存在一些不基于传统访问控制模型的访问控制方案。最近，随着区块链技术的高速发展，一些分布式的融合异构物联网环境的访问控制方案也被广泛讨论。

基于 RBAC、ABAC 或 UCON 模型的物联网访问控制方案：Barka 等人整合了基于角色的访问控制模型和 ISO 的访问控制标准 ISO/IEC 10181-3，实现对万维物联网（Web of Things，WoT）中物理设备之间的安全访问控制。他们的方案基本上遵循 3.1.2 节中介绍的访问控制实现过程，将模型中的各个组成部分和 WoT 中资源组件一一做了映射，构建由请求监听模块和策略更新模块组成的访问实施部件，以及包含授权模块的访问决策部件，支持对物理设备进行动态地访问控制，如图 3.8 所示。但是，他们的方案对物理

设备的访问控制是完全中心化的，终端物理设备不具备自主控制资源被访问的能力。

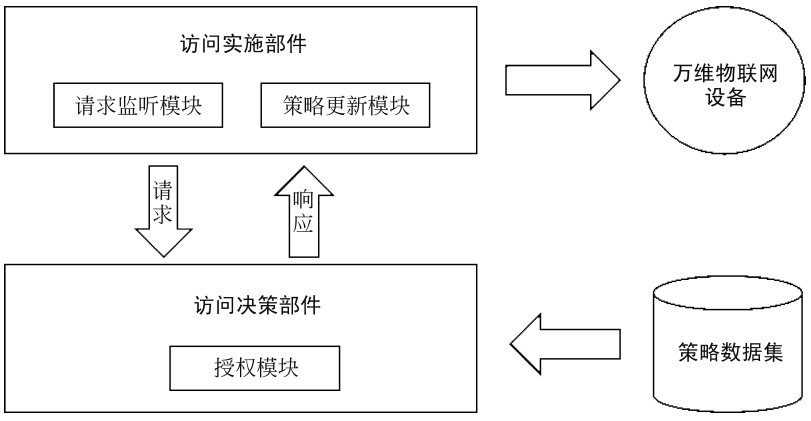

图 3.8 Barka 物联网访问控制方案的部件组成

不同于上述中心化的访问控制方案，R2BAC 机制建立在分布式信誉评估基于角色的访问控制模型，实现管理物联网系统中传感器节点之间访问资源的活动。其中，分布式的信誉管理模块由全网节点以服务质量（Quality of Service，QoS）为评估标准对传感器节点的网络行为的可信度进行评价，并以此建立节点之间的可信关系。在该方案中，传感器节点的信誉值和节点之间的可信程度决定了整个网络的连接拓扑，同时，一个节点的信誉值也决定了它访问其他节点资源时被赋予的角色。具体来说，在 R2BAC 机制中，用户指物联网系统中的传感器节点；赋予一个用户的角色由信誉管理模块决定，通过评估用户在网络中的信誉值决定该用户的角色对哪些资源有哪些操作权限；与传统的访问控制方案类似，权限就是一系列在物联网系统中对节点资源的操作。

由于 ABAC 模型相比于 RBAC 模型对用户和资源有更细粒度的刻画，例如 ABAC 模型中的属性不仅可以包含对 RBAC 模型中用户角色的刻画，还可以描述关于用户的权限属性和关于资源的特征属性等（见 3.1.4 节）。因此，Sciancalepore 等人基于 RBAC 模型和令牌的授权技术实现多个物联网平台资源的访问控制，并支持不同平台之间的数据共享。他们的方案利用 RBAC 模型完成关于用户的角色和权限属性以及物联网平台资源的特征属性的描述，并将属性封装成访问令牌集存储于认证授权管理（Authentication and Authorization Management，AAM）中心。如图 3.9 所示，用户要访问物联网平台资源时，先在认证授权管理中心完成身份认证，然后拿到允许访问指定物联网平台资源的令牌，接着用户凭借包含属性描述的令牌请求资源，被请求的物联网平台根据令牌中的属性描述和资源的访问策略的匹配结果响应用户的请求。

由于使用控制模型支持属性更新和访问策略持续性，因此，在理论上它相较于 RBAC 和 ABAC 更适合应用于动态开放的物联网环境。通过将使用控制模型抽象的组件一一映射到物联网的实体中，进而实现具体访问控制方法。具体来说，可以将客体映射为物联网设备，如车；客体属性描述设备的实际信息，如车的信誉值；主体映射为应用层的物联网服务，如路况服务；主体属性则为服务的实际信息，如路况服务提及的车载信息；条件则根据物联网环境决定，如建立在信誉值之上的设备之间连接的可信度；职能则依据实际需要

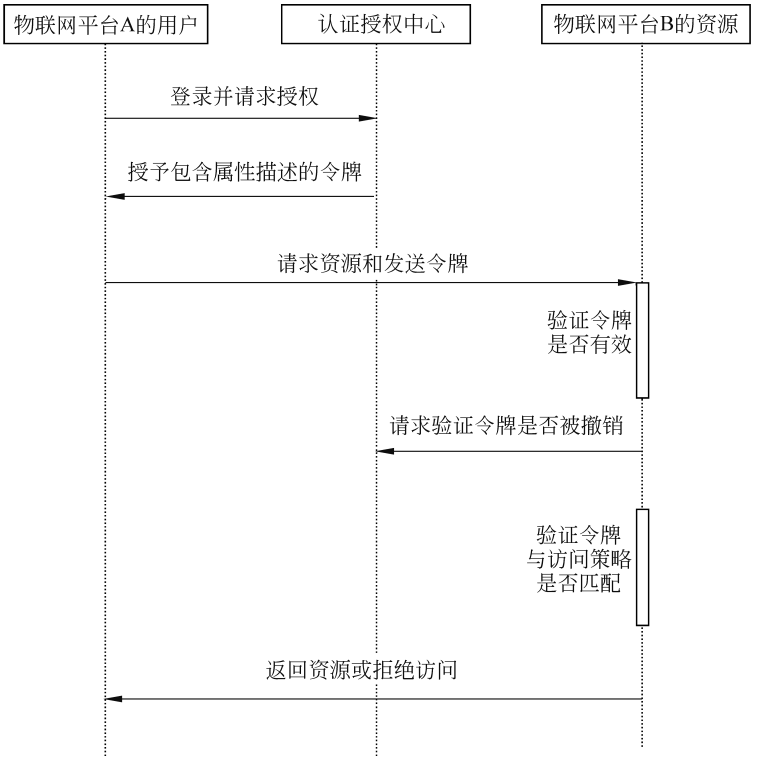

图 3.9　Sciancalepore 物联网访问控制方案交互图

来定，贯穿访问控制过程的进行前、进行中和进行后。然后，在设备和服务之间制定具体的控制策略，例如只有当设备的信誉值大于某个值时才能访问哪种可信程度的服务资源。目前，存在一些理论上将使用控制模型应用到物联网的方案，还没有在实际中部署案例。

在 3.1 节介绍自主访问控制模型时提到了基于权能的访问控制表示方法，由于它相较于访问控制列表更节省存储的开销，同时物联网设备的计算能力和存储空间有限，所以基于权能的访问控制表示方法被广泛地应用在物联网领域。ICAP 方案在传统的表示方法中增加了唯一标识设备身份的标识符使对资源访问的控制更加灵活。CCAAC 方案则在 ICAP 方案的基础上添加了一个属性用来表示被访问资源的上下文信息，使资源的访问控制更细粒度。不同于上述中心化的访问控制方案，DCapBAC 是一个可延展的分布式的访问控制方案，每个物联网设备有自主访问控制决策的权利，但它不考虑被访问资源的上下信息，没有描述访问策略如何制定，同时也不支持委托和撤销职能的功能，更多考虑访问实施部件的设计。

最近，区块链技术的兴起也为在物联网环境下部署分布式访问控制技术的研究带来了福音。ControlChain 就是一种基于区块链技术的访问控制框架，它将区块链作为一种分布式的数据库并分成 4 种区块链，包括关系区块链、上下文区块链、追责区块链和授权策略区块链。关系区块链用于存储所有设备的公开证书和维护设备之间的关系，上下文区块链记录了所有关于设备输入、处理和输出数据，追责区块链则存储所有对资源的访问

记录，授权策略区块链则维护了用户制定的设备访问控制策略。在访问控制策略的设计方面，ControlChain 沿用了访问控制列表、基于权能的访问控制列表和基于属性的访问控制模型 3 种传统的访问控制技术，不过，ControlChain 仅仅是理论上的框架。另一种同样基于区块链技术的访问控制方案 BleadCAC 进一步在理论仿真的层次上与传统的 RBAC 和 ABAC 方案做了处理延迟方面的对比，显示其优越性。它融合了传统的基于权能的访问控制列表和智能技术，实现了资源访问控制管理的可延展性以及策略的细粒度和动态管理。物联网中的用户对设备和设备数据有自主管理权力，制定和授予访问策略通过区块链的智能合约技术完成，由于智能合约被部署在分布式的区块链网络中，因此访问策略在分布式的环境被验证，增强了策略实施的安全性。目前，由于物联网环境设备和通信协议多源异构，传统的访问控制技术不能很好地解决其中的访问控制问题。因此，将区块链技术应用于增强物联网资源访问控制安全性的研究成了当前研究的热点。

3.2 身份认证

如今，互联网时代下在线服务涉及人们生活的方方面面。在人们使用在线服务之前，服务提供商往往需要对用户先进行身份认证。例如，账户密码认证，这里的账户就是指用户使用该在线服务的身份。

3.2.1 身份认证的基本概念

实际上，很难给身份下一个准确的定义。对身份的定义需要考虑具体的应用场景、语义上下文和身份的用途等。总体来说，身份是对一个实体在具体上下文环境中的表示，包含标志符和用户的证书，如用户密码、用户的配置文件等。在本书中，我们将身份定义为用户提供给应用和服务提供商的表示其身份的证书，从而应用和服务提供商可以使用用户证书来区别不同的用户并提供给用户不同的权限或服务。身份认证是应用和服务提供商验证用户证书真实性的过程，如用户登录在线服务系统时提供的用户名和密码是否与存储在服务端的用户证书相匹配。具体来说，这个过程往往由服务端的身份管理模块完成，身份管理模块定义了一系列身份管理策略和规则，管理用户身份的生命周期，包括身份创建、身份维护和身份删除。身份认证过程中包含 3 个角色：用户、服务提供商和身份提供商。使用服务的用户可以是一个真实的人也可以是虚拟的实体，但必须有唯一的身份。身份提供商除提供身份注册、身份认证和身份存储这些基本的功能外，也可以根据用户的身份提供不同级别的信用等级。基于身份提供商对用户的身份认证结果，服务提供商提供给用户相应的服务。

3.2.2 身份认证的实现原理

在介绍身份认证的基本概念时，提到了身份认证的基本过程，其中，身份提供商是整个认证过程的关键角色。在具体实现时，身份提供商可以为用户提供 4 种类型的身份：证书身份、标识身份、属性身份和指纹身份。证书身份是指基于证书来认证用户的身份，ITU-TX.509 是最常用的证书标准。标识身份通常通过用户名、邮箱或身份证等来唯一

标识身份。属性身份则指用来描述用户身份的属性，它可以是用户身份证的具体数据，例如用户名、地址或联系号码等，如果要提供这类身份，身份提供商往往需要和政府合作。最后一种指纹身份指用户在使用服务过程中独属于用户的一些用户数据记录，可以是信誉值或访问历史记录等，这种特殊的指纹身份可以用于计算机安全分析，例如，根据黑客对服务器的特殊访问行为来判别黑客对服务器实施的攻击类型。虽然上面提到了 4 种身份类型，但在常见的身份认证场景中，用户使用服务前的身份认证过程基本上使用前两种身份。

另一方面，根据服务提供商和身份提供商的关系，存在 3 种模式来实际部署身份管理模块。

(1) 单例模式。服务提供商和身份提供商两种角色是统一的整体，一个存储所有用户身份及相应证书的服务器同时充当这两种角色，负责对用户身份数据的所有操作，包括颁发、修改、认证、删除和授权等。很显然，这种身份管理部署模式容易使认证服务器因用户访问过量而负载过大。

(2) 中心化模式。不同于单例模式，它的服务提供商和身份提供商两个角色是职责分明的，服务提供商服务器负责响应用户的请求服务而身份提供商服务器则存储了用户的身份和证书，并负责用户的身份认证和授权等功能。以这种模式部署的身份管理系统常见有 PKI(Public Key Infrastructure)、Kerberos 和 CAS(Central Authentication Service)等。单例和中心化两种模式身份和证书都是中心化存储的，都存在单点故障的问题。

(3) 分布式模式。这种模式下存在多个服务提供商，每个服务提供商存储用户的部分身份信息，但只有一个身份提供商存储用户的证书信息，并负责用户身份认证和一些更复杂的功能，如基于身份的访问控制。分布式模式下的身份认证系统也有很多，如 OASIS、SAML 和 Shibboleth 框架等。

3.2.3　身份认证的常见技术

1. 基于用户名-密码的身份认证

基于用户名-密码的身份认证是最常见的，也是最基本的身份认证技术。通常是服务端先创建并存储用户的用户名和密码对，当用户使用服务之前，提交正确的用户名和密码给服务端，服务端通过匹配用户名和密码来完成对用户身份的认证。

这种认证技术虽然最简单，但安全性最弱。首先，它容易遭受攻击，暴力破解攻击是对用户密码最常见的攻击。黑客可以简单地写一个密码穷举脚本穷举用户密码字符的可能组合，进而找到匹配的密码。为抵抗暴力破解攻击，服务端可以限定用户请求身份认证的次数。例如，当用户身份认证请求失败的次数超过预设的门限值，服务端可以临时冻结用户的账户。然而，攻击者更有可能直接攻击服务端的用户密码数据库而不是直接发起对用户的攻击，从而一次性拿到所有的用户名和密码对。这就要求服务端对用户的实际密码进行加密后再进行存储，常见的方法有密码哈希和密码加盐。其次，这个认证方式的安全性很大程度上依赖用户本身设置的密码，有调查显示 90%的用户设置的密码都是弱

密码而且往往有规律可循，因此易遭黑客攻击。为了增强安全性，服务端可以在用户设置密码时根据一定的标准提示所设密码的安全强度，通常的设置标准是至少 8 个字符，并且是大小写字符、数字和特殊符号的组合。最后，用户很难对多个账户的密码进行管理，因此许多用户会对多个账户使用同一个密码，这同样带来了安全隐患。

2. 基于智能卡的身份认证

虽然基于用户名-密码的身份认证技术可以通过对数据库的用户密码进行哈希再存储来抵抗黑客的窃取，但是它无法抵抗攻击者对其进行篡改，从而导致用户认证失败。而基于智能卡的身份认证技术可以避免这样的攻击。它的身份认证组件包括两部分：智能卡硬件设备，它存储了用户的公钥证书信息；个人识别密码（Personal Identification Number，PIN），作为用户访问智能卡的认证密钥。

智能卡硬件设备既有完成简单认证过程的操作系统模块，也有安全存储个人信息的内存存储模块。内存存储模块提供了不可篡改的特性，只有用户提供正确的个人识别密码才能对内存访问相应的数据。操作系统模块则支持认证过程的密码操作，如数据签名和密钥交换等。

个人识别密码是智能卡认证用户的唯一密钥，它不会被传输到网上。当用户输入个人识别密码后，智能卡会检索对应的公钥并验证个人识别密码的有效性。

由于智能卡身份认证技术在一张智能卡上完成用户的整个认证过程，具有强认证性，既支持物理认证也允许在线逻辑认证，加强了认证过程的安全性和隐私性。因此，它也被广泛应用于许多场景，如密码管理、虚拟私有网认证、电子签名等。而且许多现实世界的机构组织也常用这种方式认证用户，如银行、公司、政府和学校等。

下面简单介绍用户基于智能卡身份认证技术登录 Windows Server 2003 的例子。首先，用户需要将其持有的智能卡插入读卡器，Windows 系统的登录模块检测到智能卡读取事件，将登录请求转发到身份识别和认证模块。其次，身份识别和认证模块提示用户输入个人识别密码，收到用户的个人识别密码后，转发给安全授权模块。再次，安全授权模块使用个人识别密码访问智能卡的内存并检索对应的公钥证书，利用公钥证书验证基于个人识别密码生成的签名。最后，如果签名验证成功，则生成一个登录会话密钥，该密钥用用户的公钥进行加密，加密好的登录会话密钥返回给用户。由于只有用户拥有个人识别密码，因此只有用户可以解密获得登录会话密钥，从而完成登录过程。

3. 基于公钥的身份认证

基于公钥的身份认证协议的构建主要以公钥密码学中的数据签名作为理论支撑（见 6.1 节），同时，通常会有一个可信的证书授权中心（或是密钥分发中心）负责分配给用户公钥证书。其实，有些研究者也提出了基于密钥的身份认证方案，但它要求客户端和服务器提前共享一个相同的密钥，这种方法在客户端（或用户）暴增的情况下存在密钥难以分配和管理的问题。本节只讨论基于公钥的身份认证协议，具体地，以 Kerberos 网络认证协议为例展开介绍。为了方便描述，先声明协议中的参与实体：客户端、认证服务器、密钥分发中心、授权过的票据、票据授权服务器和服务器。粗略地讲，用户登录客户端时需

要提供密码和用户名给认证服务器，认证服务器会把用户名转发给密钥分发中心，然后密钥分发中心会生成一个由票据授权服务器授权过的票据，这个票据将返回给合法的登录用户，之后用户需要提供有效的票据才可以访问服务器的服务。接下来，具体从以下 4 方面来介绍 Kerberos 协议，如图 3.10 所示。

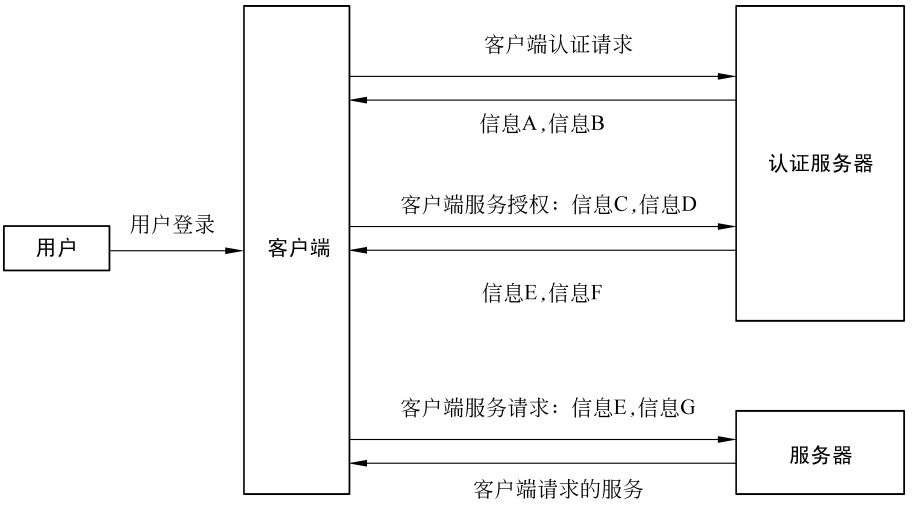

图 3.10　Kerberos 协议

(1) 用户登录。用户在客户端输入用户名和密码，客户端使用单向哈希函数以用户密码为输入并生成一个密钥k_u。

(2) 客户端认证。客户端代表用户请求服务，于是发送用户名给认证服务器，注意此时并没有将用户的密码和密钥发送给认证服务器。认证服务器检查用户是否存在数据库中，如果存在，则找到用户名所对应的密码，然后生成密码的哈希值作为密钥k_u。同时，密钥分发中心生成一个由k_u加密的票据授权服务器产生的会话密钥，记为信息 A，以及由票据授权服务器签过名的票据，记为信息 B。这个票据包含了客户端的 ID、IP 地址、票据有限时长和票据授权服务器产生的会话密钥k_{ct}。认证服务器将信息 A 和信息 B 返回给客户端，客户端收到后，用k_u解密消息 A 获得会话密钥。如果用户输入的密码是错误的，则k_u和认证服务器生成的密钥是不匹配的，这时不能获得会话密钥。客户端获得会话密钥之后就可以使用它和信息 B 与票据授权服务器进行安全通信，同时，这也表明用户已经向票据授权服务器证明了自己的身份。

(3) 客户端服务授权。当客户端要访问服务时，它需要发送两个信息：信息 C 和信息 D。信息 C 包含信息 B(票据)和所请求服务的 ID；信息 D 包含用会话密钥加密的客户端的 ID 和请求时间戳，称为认证器。票据授权服务器收到信息 C 和 D 之后，先从信息 C 中提取出信息 B，然后用会话密钥解密信息 B 并解密信息 D。接着，验证信息 D 中的客户端 ID 和信息 B 中的客户端 ID 是否匹配，如果匹配，票据授权服务器返回两个消息(信息 E 和信息 F)给客户端。其中信息 E 包含用服务器的密钥加密的票据，加密的内容有客户端 ID、IP 地址、票据有效时长和客户端与服务器之间的会话密钥 k_{cs}。信息 F 则是用 k_{ct}

加密后的 k_{cs}。

(4) 客户端服务请求。客户端收到票据授权服务器的信息 E 和信息 F 后就可以用它们向服务器认证它的身份。首先，客户端发送给服务器信息 E 和信息 G，信息 G 是用k_{cs}加密客户端 ID、时间戳生成的认证器。然后，服务器收到后解密信息 E 得到k_{cs}，接着使用k_{cs}解密信息 G，通过对比两个信息中的客户端 ID 是否相等，如果相等，则客户端验证通过，服务器可以提供所请求的服务给客户端。

4. 基于生物特征的身份认证

一方面，传统的基于用户名-密码的身份认证方法易遭受字典攻击；另一方面，基于公钥的身份认证方法用户的密钥难以维护。另外，由于密码和密钥可能丢失或者共享，这些方法并没有保证认证用户的真实性。因此，研究者提出了基于生物特征的身份认证方法，它利用用户的生理特征或行为特征作为用户的身份，常见的特征包括指纹、人脸、掌纹和虹膜等。

一个基于生物特征的系统其实是一种模式识别系统。一般来说，系统先收集每个个体的生理数据，然后提取相应的生理数据特征，最后再将特征数据和数据库中构建好的生物特征模板数据匹配。因此，基于生物特征的认证是通过个体的生理特征数据来构建和确认该个体的身份。相比于其他身份认证方法，基于生物特征的身份认证有许多优点。例如，特征难以猜测、不易丢失或遗忘、很难被复制和共享且不能被伪造。所以，一个安全的基于生物特征的身份认证方案可以抵抗常见的暴力破解攻击、重放攻击和篡改攻击。

下面介绍一个经典的结合了智能卡和生物特征进行两方相互身份认证的方案，如图 3.11 所示。从 3 个步骤介绍该方案：注册阶段、登录阶段和认证阶段。在注册阶段，用户首先在服务器一方提供的生物特征采集器录入生理特征 B，如指纹，并将密码 PW 和身份 ID 安全地发给服务器一方。其次，服务器计算用户数据的哈希值 $R=\text{hash}(\text{PW}\parallel\text{hash}(B))$和 $E=\text{hash}(\text{ID}\parallel S)\oplus R$，其中 S 是由服务器选择的随机数。最后，服务器将存有用户(ID，hash(·)，hash(B)，E)信息的智能卡安全地发配给相应用户。在登录阶段，用户将智能卡插入读卡器并向指纹验证器录入指纹B'(这里读卡器和指纹验证器充当客户端)，如果指纹匹配成功，用户继续输入密码 PW′，智能卡计算 $R'=\text{hash}(\text{PW}'\parallel\text{hash}(B'))$，$\text{M1}=E\oplus R'=\text{hash}(\text{ID}\parallel S)$ 以及 $\text{M2}=\text{M1}\oplus R_c$，其中 R_c 由智能卡随机选择的随机数，接着信息(ID，M2)被发送到服务器。服务器执行认证阶段的计算，首先它检验用户 ID 是否有效，然后计算 $\text{M3}=\text{hash}(\text{ID}\parallel S)$，$\text{M4}=\text{M2}\oplus\text{M3}=R_c$，$\text{M5}=\text{M3}\oplus R_s$ 和 $\text{M6}=\text{hash}(\text{M2}\parallel\text{M4})$，其中 R_s 是服务器选择的随机数，M5 和 M6 发送给客户端，客户端收到消息后验证 M6 是否与 $\text{hash}(\text{M2}\parallel R_c)$相等。如果相等则表明服务器是通过了客户端的验证，于是计算 $\text{M7}=\text{M5}\oplus\text{M1}=R_s$ 和 $\text{M8}=\text{hash}(\text{M5}\parallel\text{M7})$，并将 M8 发送给服务器。服务器收到 M8 后验证 M8 是否与 $\text{hash}(\text{M5}\parallel R_s)$相等，如果相等，服务器通过用户在客户端的登录请求，否则拒绝用户登录。

通过上面的介绍，不难发现这个方案加强了原本基于智能卡身份认证方法的安全性，能够实现客户端和服务器两方身份都不可被假冒，而且即使当智能卡丢失，攻击者也无法从智能卡中提取用户的秘密信息，这是由哈希函数的不可逆的特征保证的。

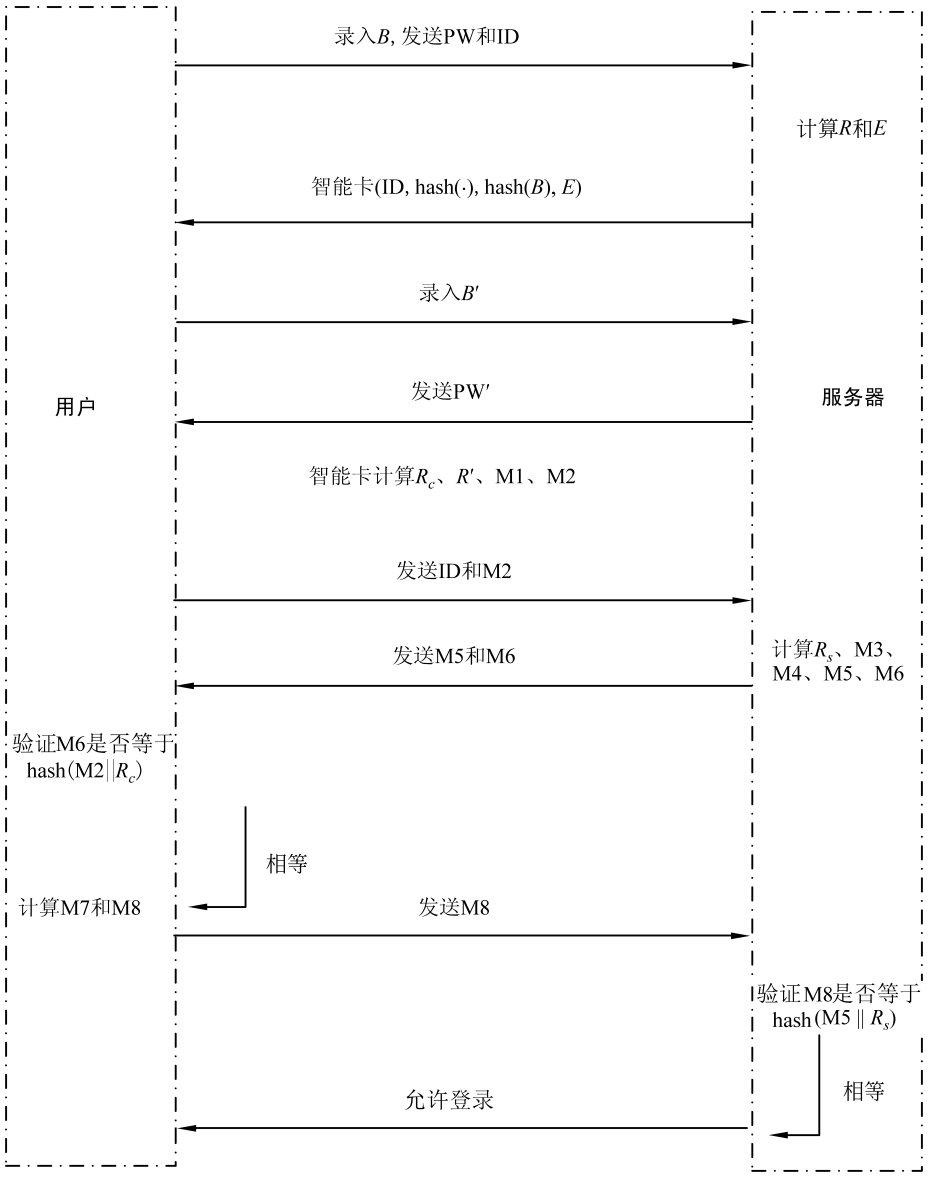

图 3.11　基于智能卡和生物特征的身份认证过程

3.2.4　身份认证技术在物联网中的应用

在物联网中，许多物联网设备通过相互通信并交换数据实现了多种多样的智能服务，如路径规划、交通导航和健康监测等。为了实现物联网设备之间可信地通信，防止和杜绝虚假设备，身份认证是必不可少的技术。本节从资源受限设备之间的身份认证以及云服务器之间的身份认证两个场景介绍身份认证技术在物联网中的应用。

一方面，物联网设备自身的计算资源和存储资源都有限，因此部署在设备之间的身份认证技术必须是轻量级的，例如基于生物特征的身份认证技术相比于其他身份认证技术

更易部署，它会被优先考虑。实际上，在无线传感器网络中，结合智能卡的基于指纹的身份认证技术被广泛应用。一般来说，应用智能卡和基于指纹的身份认证技术来实现网络中节点的认证需要基站的参与，它具有如下 4 个步骤：在预部署阶段，基站给网络中的每个传感器节点分配一个唯一 ID。用户为了访问传感器的数据需要向基站发送指纹信息、用户 ID 和密码完成注册，随后基站生成一个含有用户指纹信息的哈希值的智能卡给用户，并在数据库存储用户的指纹信息的哈希值。当用户登录时，用户把智能卡插入读卡器并输入指纹信息，如果指纹验证成功，用户输入密码，智能卡读取密码之后验证用户 ID 和密码是否匹配，如果匹配则登录成功。当用户访问指定 ID 的传感器节点的数据时，基站计算用户输入的指纹信息的哈希值，并将该哈希值与数据库中存储的指纹模板信息进行比对，若匹配正确，则允许用户访问。另一方面，由于云服务器拥有丰富的计算机资源，可以帮助资源受限的设备处理一些复杂的计算任务。因此，设备往往需要和云服务器进行安全认证前提下的通信。根据 3.2.2 节关于身份认证的实现原理，可以直接应用一些常见的身份认证技术实现设备与云服务器的认证通信。但有时候需要考虑场景的具体需求，基于已有的身份认证技术实现在功能上、性能上和安全性上更优的设备与云服务器身份认证方案。例如，通过引入职能分明的多代理合作完成认证过程来减低云服务器的认证负载，实现高扩展性的设备身份认证方案；通过设计双层认证过程实现安全程度可选择的身份认证过程；基于智能卡的身份认证技术实现用户只需进行一次身份认证就可以访问多个云服务器资源（互联云）的身份认证方案，这种情况下，用户的互联云账户就是用户的身份。为了降低通信开销和提高延展性，有研究者使用层级的架构实现互联云下的身份认证方案。

3.3 本章小结

如果没有访问控制和身份认证机制，物联网将面临很多安全威胁。访问控制机制允许定义访问控制策略，基于定义的策略建立访问控制模型、基于建立的模型设计访问控制协议，最后基于设计的协议实现访问控制的过程，可以有效保护物联网资源的访问安全。身份认证是应用和服务提供商验证用户证明或证书真实性的过程，确保操作物联网资源用户身份的唯一性。这两项技术在物联网中应用广泛，基于不同访问控制模型的访问技术和基于不同介质实现的身份认证技术在特定的物联网应用场景中，实现访问控制和身份认证发挥着关键性作用。

3.4 练习

一、填空题

1. 根据 AAA 标准，访问控制的定义包含_____、_____和_____ 3 个属性。

2. 通常来说，一个有效安全的访问控制协议需要满足 3 个特性：_____、_____、_____。

3. 定义访问控制策略时需要明确 3 方面，分别是_____、_____、_____。

4. _____支持动态更新控制策略中的属性。

5. OAuth 2.0 协议定义了 4 个角色，包括_____、_____、_____、_____。

6. 基于公钥的身份认证的协议有_____。

二、选择题

1. XACML 标记语言的响应可以包含(　　)决策。

A. 允许　　B. 拒绝　　C. 请求无效　　D. 中断

2. OAuth 2.0 协议结合 HTTPS 来实现认证，替换了 OAuth 1.0 中(　　)步骤，从而更加轻便，带来更好的用户体验。

A. 签名　　B. 加密　　C. 验证　　D. 解密

3. 根据服务提供商和身份提供商的关系，存在(　　)模式来实际部署身份管理模块。

A. 单例模式　　B. 多例模式　　C. 中心化模式　　D. 分布式模式

三、问答题

1. 描述现实世界中一个应用了基于角色访问控制模型的访问控制实例。

2. 简要描述 OAuth 2.0 协议的基本原理并举例现实世界中哪些物联网产品使用了该协议。

3. 传统的访问控制模型 DAC、MAC、RBAC、ABAC 和 UCON 模型的区别是什么？

4. 为什么区块链技术能被应用于增强物联网资源访问控制安全性？

5. 身份认证机制中的“身份”指什么？举一个现实世界中的例子。

6. 基于用户名-密码的身份认证机制最普遍，它的缺点是什么？在现实世界中使用时是如何克服这些缺点的？

7. 有哪些基于生物特征的身份认证例子？

8. 你觉得当前发展得如火如荼的机器学习技术能从哪些方面改进基于生物特征的身份认证技术？

9. 列出基于生物特征的和非基于生物特征的身份认证技术的优缺点。

3.5　实践：编程实现一个口令认证系统

基于口令的身份认证系统是由人和计算机组成的系统，该系统提供信息以支持组织内的操作、认证、管理、分析和决策功能。数据库作为访问和处理数据的工具，在系统中至关重要。我们需要在此基础上建立数据库及其应用程序系统，以便它可以有效地存储和管理满足应用程序需求的数据，包括信息管理和认证。信息管理要求是指应在数据库中存储和管理的数据对象；数据操作要求是指对数据对象的操作，例如查询、添加、删除、更改和其他操作。