数据库系统安全机制

数据库安全机制是用于实现数据库的各种安全策略的功能集合，正是由这些安全机制来实现安全模型，进而实现保护数据库系统安全的目标。近年来，对用户的认证与鉴别、存取控制、数据库加密及推理控制等安全机制的研究取得了不少新的进展。

1. 用户标识与鉴别

用户标识是指用户向系统出示自己的身份证明，最简单的方法是输入用户 ID 和密码。标识机制用于惟一标志进入系统的每个用户的身份，因此必须保证标识的惟一性。鉴别是指系统检查验证用户的身份证明，用于检验用户身份的合法性。标识和鉴别功能保证了只有合法的用户才能存取系统中的资源。

由于数据库用户的安全等级是不同的，因此分配给他们的权限也是不一样的，数据库系统必须建立严格的用户认证机制。身份的标识和鉴别是 DBMS 对访问者授权的前提，并且通过审计机制使 DBMS 保留追究用户行为责任的能力。功能完善的标识与鉴别机制也是访问控制机制有效实施的基础，特别是在一个开放的多用户系统的网络环境中，识别与鉴别用户是构筑 DBMS 安全防线的第 1 个重要环节。

近年来标识与鉴别技术发展迅速，一些实体认证的新技术在数据库系统集成中得到应用。目前，常用的方法有通行字认证、数字证书认证、智能卡认证和个人特征识别等。

通行字也称为“口令”或“密码”，它是一种根据已知事物验证身份的方法，也是一种最广泛研究和使用的身份验证法。在数据库系统中往往对通行字采取一

些控制措施，常见的有最小长度限制、次数限定、选择字符、有效期、双通行字和封锁用户系统等。一般还需考虑通行字的分配和管理，以及在计算机中的安全存储。通行字多以加密形式存储，攻击者要得到通行字，必须知道加密算法和密钥。算法可能是公开的，但密钥应该是秘密的。也有的系统存储通行字的单向 Hash 值，攻击者即使得到密文也难以推出通行字的明文。

数字证书是认证中心颁发并进行数字签名的数字凭证，它实现实体身份的鉴别与认证、信息完整性验证、机密性和不可否认性等安全服务。数字证书可用来证明实体所宣称的身份与其持有的公钥的匹配关系，使得实体的身份与证书中的公钥相互绑定。

智能卡（有源卡、IC 卡或 Smart 卡）作为个人所有物，可以用来验证个人身份，典型智能卡主要由微处理器、存储器、输入输出接口、安全逻辑及运算处理器等组成。在智能卡中引入了认证的概念，认证是智能卡和应用终端之间通过相应的认证过程来相互确认合法性。在卡和接口设备之间只有相互认证之后才能进行数据的读写操作，目的在于防止伪造应用终端及相应的智能卡。

根据被授权用户的个人特征来进行确证是一种可信度更高的验证方法，个人特征识别应用了生物统计学（Biometrics）的研究成果，即利用个人具有惟一性的生理特征来实现。个人特征都具有因人而异和随身携带的特点，不会丢失并且难以伪造，非常适合于个人身份认证。目前已得到应用的个人生理特征包括指纹、语音声纹（voice- print）、DNA、视网膜、虹膜、脸型和手型等。一些学者已开始研究基于用户个人行为方式的身份识别技术，如用户写签名和敲击键盘的方式等。

个人特征一般需要应用多媒体数据存储技术来建立档案，相应地需要基于多

媒体数据的压缩、存储和检索等技术作为支撑。目前已有不少基于个人特征识别的身份认证系统成功地投入应用。如美国联邦调查局（FBI）成功地将小波理论应用于压缩和识别指纹图样，从而可以将一个 10 MB 的指纹图样压缩成 500 KB，从而大大减少了数百万指纹档案的存储空间和检索时间。

2. 存取控制

访问控制的目的是确保用户对数据库只能进行经过授权的有关操作。在存取控制机制中，一般把被访问的资源称为“客体”，把以用户名义进行资源访问的进程、事务等实体称为“主体”。

传统的存取控制机制有两种，即 DAC（Discretionary Access Control，自主存取控制）和 MAC（Mandatory Access Control，强制存取控制）。在 DAC 机制中，用户对不同的数据对象有不同的存取权限，而且还可以将其拥有的存取权限转授给其他用户。DAC 访问控制完全基于访问者和对象的身份；MAC 机制对于不同类型的信息采取不同层次的安全策略，对不同类型的数据来进行访问授权。在 MAC 机制中，存取权限不可以转授，所有用户必须遵守由数据库管理员建立的安全规则，其中最基本的规则为“向下读取，向上写入”。显然，与 DAC 相比，MAC 机制比较严格。

近年来，RBAC（Role-based Access Control，基于角色的存取控制）得到了广泛的关注。RBAC 在主体和权限之间增加了一个中间桥梁——角色。权限被授予角色，而管理员通过指定用户为特定角色来为用户授权。从而大大简化了授权管理，具有强大的可操作性和可管理性。角色可以根据组织中的不同工作创建，然后根据用户的责任和资格分配角色，用户可以轻松地进行角色转换。而随着新应用和新系统的增加，角色可以分配更多的权限，也可以根据需要撤销相应的权限。

RBAC 核心模型包含了 5 个基本的静态集合,即用户集(users)、角色集(roles)、特权集 （perms）（包括对象集（objects）和操作集（operators）），以及一个运行过程中动态维护的集合，即会话集（sessions），如图 1-1 所示。

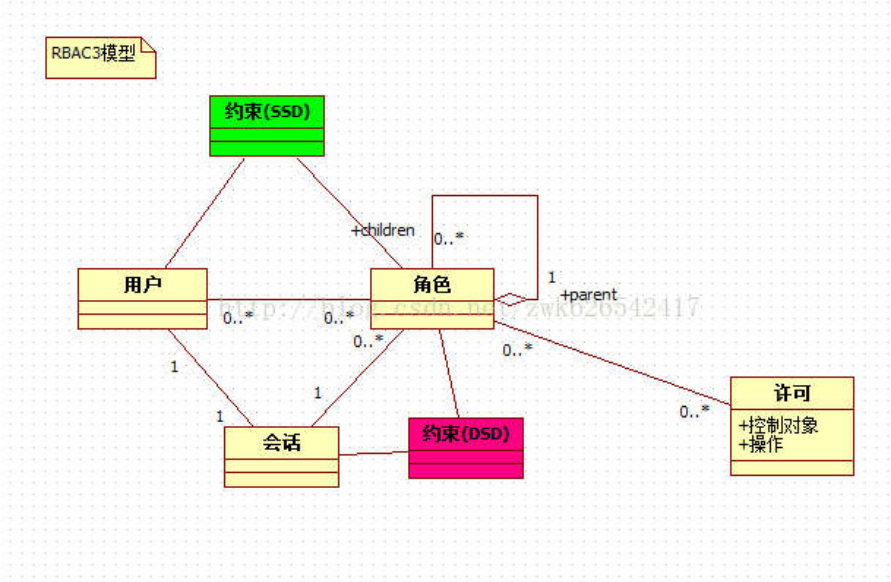

图 1-1　RBAC 核心模型

用户集包括系统中可以执行操作的用户，是主动的实体；对象集是系统中被动的实体，包含系统需要保护的信息；操作集是定义在对象上的一组操作，对象上的一组操作构成了一个特权；角色则是 RBAC 模型的核心，通过用户分配（UA）和特权分配（PA）使用户与特权关联起来。

RBAC 属于策略中立型的存取控制模型，既可以实现自主存取控制策略，又可以实现强制存取控制策略。它可以有效缓解传统安全管理处理瓶颈问题，被认为是一种普遍适用的访问控制模型，尤其适用于大型组织的有效的访问控制机制。

2002 年，Park. J 和 Sundhu. R 首次提出了 UCON（Usage Control，使用控制）的概念。UCON 对传统的存取控制进行了扩展，定义了授权（Authorization）、职责（Obligation）和条件（Condition）3 个决定性因素，同时提出了存取控制的连续性（Continuity）和易变性（Mutability）两个重要属性。UCON 集合了传统

的访问控制、可信管理，以及数字权力管理，从而用系统方式提供了一个保护数字资源的统一标准的框架，为下一代存取控制机制提供了新思路。

3.数据库加密

由于数据库在操作系统中以文件形式管理，所以入侵者可以直接利用操作系统的漏洞窃取数据库文件，或者篡改数据库文件内容。另一方面，数据库管理员（DBA）可以任意访问所有数据，往往超出了其职责范围，同样造成安全隐患。因此，数据库的保密问题不仅包括在传输过程中采用加密保护和控制非法访问，还包括对存储的敏感数据进行加密保护，使得即使数据不幸泄露或者丢失，也难以造成泄密。同时，数据库加密可以由用户用自己的密钥加密自己的敏感信息，而不需要了解数据内容的数据库管理员无法进行正常解密，从而可以实现个性化的用户隐私保护。

对数据库加密必然会带来数据存储与索引、密钥分配和管理等一系列问题，同时加密也会显著地降低数据库的访问与运行效率。保密性与可用性之间不可避免地存在冲突，需要妥善解决二者之间的矛盾。

数据库中存储密文数据后，如何进行高效查询成为一个重要的问题。查询语句一般不可以直接运用到密文数据库的查询过程中，一般的方法是首先解密加密数据，然后查询解密数据。但由于要对整个数据库或数据表进行解密操作，因此开销巨大。在实际操作中需要通过有效的查询策略来直接执行密文查询或较小粒度的快速解密。

一般来说，一个好的数据库加密系统应该满足以下几个方面的要求。

① 足够的加密强度，保证长时间且大量数据不被破译。

② 加密后的数据库存储量没有明显的增加。

③ 加解密速度足够快，影响数据操作响应时间尽量短。

④ 加解密对数据库的合法用户操作（如数据的增、删、改等）是透明的。

⑤ 灵活的密钥管理机制，加解密密钥存储安全，使用方便可靠。

3.1．数据库加密的实现机制

数据库加密的实现机制主要研究执行加密部件在数据库系统中所处的层次和位置，通过对比各种体系结构的运行效率、可扩展性和安全性，以求得最佳的系统结构。

按照加密部件与数据库系统的不同关系，数据库加密机制可以从大的方面分为库内加密和库外加密。

3.1.1 库内加密

库内加密在 DBMS 内核层实现加密，加 / 解密过程对用户与应用透明，数据在物理存取之前完成加 / 解密工作。

这种方式的优点是加密功能强，并且加密功能集成为 DBMS 的功能，可以实现加密功能与 DBMS 之间的无缝耦合。对于数据库应用来说，库内加密方式是完全透明的。

库内加密方式的主要缺点如下。

— 对系统性能影响比较大，BMS 除了完成正常的功能外，还要进行加 / 解密运算，从而加重了数据库服务器的负载。

— 密钥管理风险大，加密密钥与库数据保存在服务器中，其安全性依赖于 DBMS 的访问控制机制。

— 加密功能依赖于数据库厂商的支持，DBMS 一般只提供有限的加密算法与强度可供选择，自主性受限。

3.1.2 库外加密

在库外加密方式中,加/解密过程发生在 DBMS 之外,DBMS 管理的是密文。加/解密过程大多在客户端实现，也有的由专门的加密服务器或硬件完成。

与库内加密方式相比，库外加密的明显优点如下。

— 由于加/解密过程在客户端或专门的加密服务器实现，所以减少了数据库服务器与 DBMS 的运行负担。

— 可以将加密密钥与所加密的数据分开保存，提高了安全性。

— 由客户端与服务器的配合，可以实现端到端的网上密文传输。

库外加密的主要缺点是加密后的数据库功能受到一些限制,例如加密后的数据无法正常索引。同时数据加密后也会破坏原有的关系数据的完整性与一致性，这些都会给数据库应用带来影响。

在目前新兴的外包数据库服务模式中,数据库服务器由非可信的第三方提供,仅用来运行标准的 DBMS，要求加密解密都在客户端完成。因此，库外加密方式受到越来越多研究者的关注。

3.2 数据库加密的粒度

一般来说，数据库加密的粒度可以有 4 种，即表、属性、记录和数据元素。不同加密粒度的特点不同，总的来说，加密粒度越小，则灵活性越好且安全性越高，但实现技术也更为复杂，对系统的运行效率影响也越大。

1.表加密

表级加密的对象是整个表,这种加密方法类似于操作系统中文件加密的方法。即每个表与不同的表密钥运算，形成密文后存储。这种方式最为简单，但因为对表中任何记录或数据项的访问都需要将其所在表的所有数据快速解密,因而执行

效率很低，浪费了大量的系统资源。在目前的实际应用中，这种方法基本已被放弃。

2.属性加密

属性加密又称为“域加密”或“字段加密”，即以表中的列为单位进行加密。一般而言，属性的个数少于记录的条数，需要的密钥数相对较少。如果只有少数属性需要加密，属性加密是可选的方法。

3.记录加密

记录加密是把表中的一条记录作为加密的单位，当数据库中需要加密的记录数比较少时，采用这种方法是比较好的。

4.数据元素加密

数据元素加密是以记录中每个字段的值为单位进行加密，数据元素是数据库中最小的加密粒度。采用这种加密粒度，系统的安全性与灵活性最高，同时实现技术也最为复杂。不同的数据项使用不同的密钥，相同的明文形成不同的密文，抗攻击能力得到提高。不利的方面是，该方法需要引入大量的密钥。一般要周密设计自动生成密钥的算法，密钥管理的复杂度大大增加，同时系统效率也受到影响。

在目前条件下，为了得到较高的安全性和灵活性，采用最多的加密粒度是数据元素。为了使数据库中的数据能够充分而灵活地共享，加密后还应当允许用户以不同的粒度进行访问。

3.3 加密算法

加密算法是数据加密的核心，一个好的加密算法产生的密文应该频率平衡，随机无重码，周期很长而又不可能产生重复现象。窃密者很难通过对密文频率，

或者重码等特征的分析获得成功。同时，算法必须适应数据库系统的特性，加/解密，尤其是解密响应迅速。

常用的加密算法包括对称密钥算法和非对称密钥算法。

对称密钥算法的特点是解密密钥和加密密钥相同，或解密密钥由加密密钥推出。这种算法一般又可分为两类，即序列算法和分组算法。序列算法一次只对明文中的单个位或字节运算；分组算法是对明文分组后以组为单位进行运算，常用有 DES 等。

非对称密钥算法也称为“公开密钥算法”，其特点是解密密钥不同于加密密钥，并且从解密密钥推出加密密钥在计算上是不可行的。其中加密密钥公开，解密密钥则是由用户秘密保管的私有密钥。常用的公开密钥算法有 RSA 等。

目前还没有公认的专门针对数据库加密的加密算法，因此一般根据数据库特点选择现有的加密算法来进行数据库加密。一方面，对称密钥算法的运算速度比非对称密钥算法快很多，二者相差大约 2~3 个数量级；另一方面，在公开密钥算法中，每个用户有自己的密钥对。而作为数据库加密的密钥如果因人而异，将产生异常庞大的数据存储量。因此，在数据库加密中一般采取对称密钥的分组加密算法。

3.4 密钥管理

对数据库进行加密，一般对不同的加密单元采用不同的密钥。以加密粒度为数据元素为例，如果不同的数据元素采用同一个密钥，由于同一属性中数据项的取值在一定范围之内，且往往呈现一定的概率分布，因此攻击者可以不用求原文，而直接通过统计方法即可得到有关的原文信息，这就是所谓的统计攻击。

大量的密钥自然会带来密钥管理的问题。根据加密粒度的不同，系统所产生

的密钥数量也不同。越是细小的加密粒度，所产生的密钥数量越多，密钥管理也就越复杂。良好的密钥管理机制既可以保证数据库信息的安全性，又可以进行快速的密钥交换，以便进行数据解密。

对数据库密钥的管理一般有集中密钥管理和多级密钥管理两种体制，集中密钥管理方法是设立密钥管理中心。在建立数据库时，密钥管理中心负责产生密钥并对数据加密，形成一张密钥表。当用户访问数据库时，密钥管理机构核对用户识别符和用户密钥。通过审核后，由密钥管理机构找到或计算出相应的数据密钥。这种密钥管理方式方便用户使用和管理，但由于这些密钥一般由数据库管理人员控制，因而权限过于集中。

目前研究和应用比较多的是多级密钥管理体制，以加密粒度为数据元素的三级密钥管理体制为例，整个系统的密钥由一个主密钥、每个表上的表密钥，以及各个数据元素密钥组成。表密钥被主密钥加密后以密文形式保存在数据字典中，数据元素密钥由主密钥及数据元素所在行、列通过某种函数自动生成，一般不需要保存。在多级密钥体制中，主密钥是加密子系统的关键，系统的安全性在很大程度上依赖于主密钥的安全性。

3.5 数据库加密的局限性

数据库加密技术在保证安全性的同时，也给数据库系统的可用性带来一些影响。

1.系统运行效率受到影响

数据库加密技术带来的主要问题之一是影响效率。为了减少这种影响，一般对加密的范围做一些约束，如不加密索引字段和关系运算的比较字段等。

2.难以实现对数据完整性约束的定义

数据库一般都定义了关系数据之间的完整性约束，如主／外键约束及值域的定义等。数据一旦加密，DBMS 将难以实现这些约束。

3.对数据的 SQL 语言及 SQL 函数受到制约

SQL 语言中的 Group by、Order by 及 Having 子句分别完成分组和排序等操作，如果这些子句的操作对象是加密数据，那么解密后的明文数据将失去原语句的分组和排序作用。另外，DBMS 扩展的 SQL 内部函数一般也不能直接作用于密文数据。

4.密文数据容易成为攻击目标

加密技术把有意义的明文转换为看上去没有实际意义的密文信息，但密文的随机性同时也暴露了消息的重要性，容易引起攻击者的注意和破坏，从而造成了一种新的不安全性。加密技术往往需要和其他非加密安全机制相结合，以提高数据库系统的整体安全性。

数据库加密作为一种对敏感数据进行安全保护的有效手段，将得到越来越多的重视。总体来说，目前数据库加密技术还面临许多挑战，其中解决保密性与可用性之间的矛盾是关键。

4. 数据库审计

数据库审计是指监视和记录用户对数据库所施加的各种操作的机制。按照美国国防部 TCSEC/TDI 标准中关于安全策略的要求，审计功能是数据库系统达到 C2 以上安全级别必不可少的一项指标。

审计功能自动记录用户对数据库的所有操作，并且存入审计日志。事后可以利用这些信息重现导致数据库现有状况的一系列事件，提供分析攻击者线索的依据。

数据库管理系统的审计主要分为语句审计、特权审计、模式对象审计和资源审计，语句审计是指监视一个或者多个特定用户或者所有用户提交的 SQL 语句；特权审计是指监视一个或者多个特定用户或者所有用户使用的系统特权；模式对象审计是指监视一个模式中在一个或者多个对象上发生的行为；资源审计是指监视分配给每个用户的系统资源。

审计机制应该至少记录用户标识和认证、客体访问、授权用户进行并会影响系统安全的操作，以及其他安全相关事件。对于每个记录的事件，审计记录中需要包括事件时间、用户、时间类型、事件数据和事件的成功 / 失败情况。对于标识和认证事件，必须记录事件源的终端 ID 和源地址等；对于访问和删除对象的事件，则需要记录对象的名称。

审计的策略库一般由两个方面因素构成，即数据库本身可选的审计规则和管理员设计的触发策略机制。当这些审计规则或策略机制一旦被触发，则将引起相关的表操作。这些表可能是数据库自定义的，也可能是管理员另外定义的，最终这些审计的操作都将被记录在特定的表中以备查证。一般地，将审计跟踪和数据库日志记录结合起来，会达到更好的安全审计效果。

对于审计粒度与审计对象的选择，需要考虑系统运行效率与存储空间消耗的问题。为了达到审计目的，一般必须审计到对数据库记录与字段一级的访问。但这种小粒度的审计需要消耗大量的存储空间，同时使系统的响应速度降低，给系统运行效率带来影响。

5. 备份与恢复

一个数据库系统总是避免不了故障的发生。安全的数据库系统必须能在系统发生故障后利用已有的数据备份，恢复数据库到原来的状态，并保持数据的完整

性和一致性。数据库系统所采用的备份与恢复技术，对系统的安全性与可靠性起着重要作用，也对系统的运行效率有着重大影响。

5.1 数据库备份

常用的数据库备份的方法有如下 3 种。

（1）冷备份

冷备份是在没有终端用户访问数据库的情况下关闭数据库并将其备份，又称为“脱机备份”。这种方法在保持数据完整性方面显然最有保障，但是对于那些必须保持每天 24 小时、每周 7 天全天候运行的数据库服务器来说，较长时间地关闭数据库进行备份是不现实的。

（2）热备份

热备份是指当数据库正在运行时进行的备份，又称为“联机备份”。因为数据备份需要一段时间，而且备份大容量的数据库还需要较长的时间，那么在此期间发生的数据更新就有可能使备份的数据不能保持完整性，这个问题的解决依赖于数据库日志文件。在备份时，日志文件将需要进行数据更新的指令“堆起来”，并不进行真正的物理更新，因此数据库能被完整地备份。备份结束后，系统再按照被日志文件“堆起来”的指令对数据库进行真正的物理更新。可见，被备份的数据保持了备份开始时刻前的数据一致性状态。

热备份操作存在如下不利因素。

— 如果系统在进行备份时崩溃，则堆在日志文件中的所有事务都会被丢失，即造成数据的丢失。

— 在进行热备份的过程中，如果日志文件占用系统资源过大，如将系统存储空间占用完，会造成系统不能接受业务请求的局面，对系统运行产生影响。

— 热备份本身要占用相当一部分系统资源，使系统运行效率下降。

（3）逻辑备份

逻辑备份是指使用软件技术从数据库中导出数据并写入一个输出文件，该文件的格式一般与原数据库的文件格式不同，而是原数据库中数据内容的一个映像。因此逻辑备份文件只能用来对数据库进行逻辑恢复，即数据导入，而不能按数据库原来的存储特征进行物理恢复。逻辑备份一般用于增量备份，即备份那些在上次备份以后改变的数据。

5.2 数据库恢复

在系统发生故障后，把数据库恢复到原来的某种一致性状态的技术称为“恢复”，其基本原理是利用“冗余”进行数据库恢复。问题的关键是如何建立“冗余”并利用“冗余”实施数据库恢复，即恢复策略。

数据库恢复技术一般有 3 种策略，即基于备份的恢复、基于运行时日志的恢复和基于镜像数据库的恢复。

（1）基于备份的恢复

基于备份的恢复是指周期性地备份数据库。当数据库失效时，可取最近一次的数据库备份来恢复数据库，即把备份的数据拷贝到原数据库所在的位置上。用这种方法，数据库只能恢复到最近一次备份的状态，而从最近备份到故障发生期间的所有数据库更新将会丢失。备份的周期越长，丢失的更新数据越多。

（2）基于运行时日志的恢复

运行时日志文件是用来记录对数据库每一次更新的文件。对日志的操作优先于对数据库的操作，以确保记录数据库的更改。当系统突然失效而导致事务中断时，可重新装入数据库的副本，把数据库恢复到上一次备份时的状态。然后系统

自动正向扫描日志文件，将故障发生前所有提交的事务放到重做队列，将未提交的事务放到撤销队列执行，这样就可把数据库恢复到故障前某一时刻的数据一致性状态。

（3）基于镜像数据库的恢复

数据库镜像就是在另一个磁盘上复制数据库作为实时副本。当主数据库更新时，DBMS 自动把更新后的数据复制到镜像数据，始终使镜像数据和主数据保持一致性。当主库出现故障时，可由镜像磁盘继续提供使用，同时 DBMS 自动利用镜像磁盘数据进行数据库恢复。镜像策略可以使数据库的可靠性大为提高，但由于数据镜像通过复制数据实现，频繁的复制会降低系统运行效率，因此一般在对效率要求满足的情况下可以使用。为兼顾可靠性和可用性，可有选择性地镜像关键数据。

数据库的备份和恢复是一个完善的数据库系统必不可少的一部分，目前这种技术已经广泛应用于数据库产品中，如 Oracle 数据库提供对联机备份、脱机备份、逻辑备份、完全数据恢复及不完全数据恢复的全面支持。据预测，以“数据”为核心的计算（Data Centric Computing）将逐渐取代以“应用”为核心的计算。在一些大型的分布式数据库应用中，多备份恢复和基于数据中心的异地容灾备份恢复等技术正在得到越来越多的应用。

6. 推理控制与隐私保护

数据库安全中的推理是指用户根据低密级的数据和模式的完整性约束推导出高密级的数据，造成未经授权的信息泄露，这种推理的路径称为“推理通道”（Inference Channel）。近年来随着外包数据库模式及数据挖掘技术的发展，对数据库推理控制（Inference Control）和隐私保护（Privacy Protection）的要求也

越来越高。

6.1 推理通道

常见的推理通道有以下 4 种。

① 执行多次查询，利用查询结果之间的逻辑联系进行推理。用户一般先向数据库发出多个查询请求，这些查询大多包含一些聚集类型的函数（如合计和平均值等）。然后利用返回的查询结果，在综合分析的基础上推断出高级数据信息。

② 利用不同级别数据之间的函数依赖进行推理分析，数据表的属性之间常见的一种关系是“函数依赖”和“多值依赖”。这些依赖关系有可能产生推理通道，如同一病房的病人患的是同一种病，以及由参加会议的人员可以推得参与会议的公司等。

③ 利用数据完整性约束进行推理，例如关系数据库的实体完整性要求每一个元组必须有一个惟一的键。当一个低安全级的用户想在一个关系中插入一个元组，并且这个关系中已经存在一个具有相同键值的高安全级元组，那么为了维护实体的完整性，DBMS 会采取相应的限制措施。低级用户由此可以推出高级数据的存在，这就产生了一条推理通道。

④ 利用分级约束进行推理。一条分级约束是一条规则，它描述了对数据进行分级的标准。如果这些分级标准被用户获知的话，用户有可能从这些约束自身推导出敏感数据。

6.2 推理控制

迄今为止，推理通道问题仍处于理论探索阶段，没有一个一劳永逸的解决方法，这是由推理通道问题本身的多样性与不确定性所决定的。目前常用的推理控制方法可以分为两类，第 1 类是在数据库设计时找出推理通道，主要包括利用语

义数据模型的方法和形式化的方法。这类方法都是分析数据库的模式，然后修改数据库设计或者提高一些数据项的安全级别来消除推理通道；第 2 类方法是在数据库运行时找出推理通道，主要包括多实例方法和查询修改方法。

IBM Almaden 研究中心的 Kristen LeFevre 等基于推理控制方法实现了一个隐私保护数据库原型系统，该模型应用于 Hippocratic 数据库取得了较好的隐私保护效果，是目前所知的最为典型和最为成功的隐私保护数据库系统。系统建立了信息泄露的表语义与查询语义模型，通过修改 SQL 语言查询条件的方法来进行查询预处理，实现了数据元素粒度的推理控制。

这个模型主要通过对 SQL 查询语句的扩展，用 Case 和 Join 语句替换查询来实现推理控制。经过对隐私策略规则的定义和执行，用户可以自己决定涉及自身隐私数据的访问策略。而数据库可以控制未经授权用户对敏感数据的访问，这样有效地实现了隐私保护。

以上探讨了数据库的多种安全机制，有必要说明的是这些安全技术不是相互独立的，而是彼此依赖并相互支持的。存取控制的正确性依赖于安全的用户标识和鉴别机制，用户标识和鉴别机制也是入侵检测和审计的基础。存取控制是数据库安全最基本，也是最核心的措施，数据库加密在带来更高安全性的同时必然带来运行效率和可用性的降低。折中的结果是部分敏感信息加密，为此需要推理控制和隐私保护手段的有效配合。所谓未雨绸缪，有备无患，备份是几乎所有数据库必需的日常维护工作，是数据库恢复的前提；恢复则是数据库安全的最后一道屏障，亡羊补牢，为时未晚