成熟的数据库加密软件应该解决两大重要难题，一是应用透明，二是密文索引。

对数据进行加密，主要有三种方式：系统中加密、客户端(DBMS外层)加密、服务器端(DBMS内核层)加密。

## 数据库加密技术综述

目前，计算机大批量数据存储的安全问题、敏感数据的防窃取和防篡改问题越来越引起人们的重视。数据库系统作为计算机信息系统的核心部件，数据库文件作为信息的聚集体，其安全性将是信息产业的重中之重。

**数据库加密的必要性**

----大型数据库管理系统的运行平台一般是Windows NT和 Unix，这些操作系统的安全级别通常为C1、C2级。它们具有用户注册、识别用户、任意存取控制（DAC）、审计等安全功能。虽然DBMS在OS的基础上增加了不少安全措施，例如基于权限的访问控制等，但OS和DBMS对数据库文件本身仍然缺乏有效的保护措施，有经验的网上黑客会“绕道而行”，直接利用OS工具窃取或篡改数据库文件内容。这种隐患被称为通向DBMS的“隐秘通道”，它所带来的危害一般数据库用户难以觉察。分析和堵塞“隐秘通道”被认为是B2级的安全技术措施。对数据库中的敏感数据进行加密处理，是堵塞这一“隐秘通道”的有效手段。

----据有关资料报道，80％的计算机犯罪来自系统内部。在传统的数据库系统中，数据库管理员的权力至高无上，他既负责各项系统管理工作，例如资源分配、用户授权、系统审计等，又可以查询数据库中的一切信息。为此，不少系统以种种手段来削弱系统管理员的权力。实现数据库加密以后，各用户（或用户组）的数据由用户用自己的密钥加密，数据库管理员获得的信息无法进行正常脱密，从而保证了用户信息的安全。另外，通过加密，数据库的备份内容成为密文，从而能减少因备份介质失窃或丢失而造成的损失。由此可见，数据库加密对于企业内部安全管理，也是不可或缺的。

----也许有人认为，对数据库加密以后会严重影响数据库系统的效率，使系统不堪重负。事实并非如此。如果在数据库客户端进行数据加/脱密运算，对数据库服务器的负载及系统运行几乎没有影响。在普通PC机上,用纯软件实现DES加密算法的速度超过200K字节/秒，如果对一篇一万汉字的文章进行加密，其加/脱密时间仅需1/10秒，这种时间延迟用户几乎无感觉。目前，加密卡的加/脱密速度一般为1M位/ 秒，对中小型数据库系统来说，这个速度即使在服务器端进行数据的加/脱密运算也是可行的，因为一般的关系数据项都不会太长（多媒体数据另当别论）。例如，在同一时间里有10个用户并发查询，每个用户平均查找1000个汉字的数据，最先得到结果的用户延迟时间小于0.02秒，最后得到结果的用户也仅需等待约0.16秒。

**基本要求**

----根据我们的研究，一个良好的数据库加密系统应该满足以下基本要求：

----1．字段加密

----在目前条件下，加/脱密的粒度是每个记录的字段数据。如果以文件或列为单位进行加密，必然会形成密钥的反复使用，从而降低加密系统的可靠性或者因加脱密时间过长而无法使用。只有以记录的字段数据为单位进行加/脱密，才能适应数据库操作，同时进行有效的密钥管理并完成“一次一密”的密码操作。

----2．密钥动态管理

----数据库客体之间隐含着复杂的逻辑关系，一个逻辑结构可能对应着多个数据库物理客体，所以数据库加密不仅密钥量大，而且组织和存储工作比较复杂，需要对密钥实现动态管理。

----3．合理处理数据

----这包括几方面的内容。首先要恰当地处理数据类型，否则 DBMS将会因加密后的数据不符合定义的数据类型而拒绝加载；其次，需要处理数据的存储问题，实现数据库加密后，应基本上不增加空间开销。在目前条件下，数据库关系运算中的匹配字段，如表间连接码、索引字段等数据不宜加密。文献字段虽然是检索字段，但也应该允许加密，因为文献字段的检索处理采用了有别于关系数据库索引的正文索引技术。

----4．不影响合法用户的操作

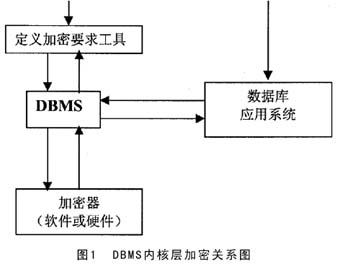
----加密系统影响数据操作响应时间应尽量短，在现阶段，平均延迟时间不应超过1/10秒。此外，对数据库的合法用户来说,数据的录入、修改和检索操作应该是透明的,不需要考虑数据的加/脱密问题。

**不同层次实现数据库加密**

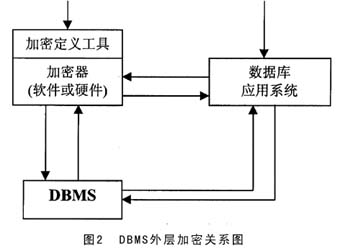
----我们可以考虑在三个不同层次实现对数据库数据的加密，这三个层次分别是OS、DBMS内核层和DBMS外层。

----在OS层，无法辨认数据库文件中的数据关系，从而无法产生合理的密钥，也无法进行合理的密钥管理和使用。所以，在OS层对数据库文件进行加密，对于大型数据库来说，目前还难以实现。

----在DBMS内核层实现加密，是指数据在物理存取之前完成加/脱密工作。这种方式势必造成DBMS和加密器(硬件或软件)之间的接口需要DBMS 开发商的支持。这种加密方式的优点是加密功能强，并且加密功能几乎不会影响DBMS 的功能。其缺点是在服务器端进行加/脱密运算，加重了数据库服务器的负载。这种加密方式如图1所示。



----比较实际的做法是将数据库加密系统做成DBMS的一个外层工具(如图2所示)。采用这种加密方式时，加/脱密运算可以放在客户端进行，其优点是不会加重数据库服务器的负载并可实现网上传输加密，缺点是加密功能会受一些限制。图中，“定义加密要求工具”模块的主要功能是定义如何对每个数据库表数据进行加密。在创建了一个数据库表后，通过这一工具对该表进行定义；“数据库应用系统”的功能是完成数据库定义和操作。数据库加密系统将根据加密要求自动完成对数据库数据的加/ 脱密。



**加密系统的有关问题**

----数据库加密系统首先要解决系统本身的安全性和可靠性问题，在这方面，可以采用以下几项安全措施：

----1． 在用户进入系统时进行两级安全控制

----这种控制可以采用多种方式，包括设置数据库用户名和口令，或者利用IC卡读写器/指纹识别器进行用户身份认证。

----2． 防止非法拷贝

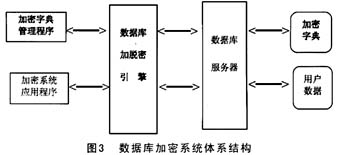
----对于纯软件系统，可以采用软指纹技术防止非法拷贝，当然，如果每台客户机上都安装加密卡等硬部件，安全性会更好。此外，还应该保留数据库原有的安全措施，如权限控制、备份/恢复和审计控制等。

----3． 安全的数据抽取方式

----提供两种卸出和装入数据库中加密数据的方式：其一是密文方式卸出，这种卸出方式不脱密，卸出的数据还是密文，在这种模式下，可直接使用 DBMS提供的卸出/装入工具；其二是明文方式卸出，这种卸出方式需要脱密，卸出的数据是明文，在这种模式下，可利用系统专用工具先进行数据转换，再使用DBMS提供的卸出/装入工具完成。

**系统结构**

----数据库加密系统分成两个功能独立的主要部件：一个是加密字典管理程序，另一个是数据库加/脱密引擎，体系结构如图3所示。



----数据库加密系统将用户对数据库信息具体的加密要求记载在加密字典中，加密字典是数据库加密系统的基础信息。

----加密字典管理程序，是管理加密字典的实用程序，是数据库管理员变更加密要求的工具。

----加密字典管理程序通过数据库加/脱密引擎实现对数据库表的加密、脱密及数据转换等功能，此时，它作为一个特殊客户来使用数据库加/脱密引擎。

----数据库加/脱密引擎是数据库加密系统的核心部件，负责在后台完成数据库信息的加/脱密处理，对应用开发人员和操作人员是透明的。

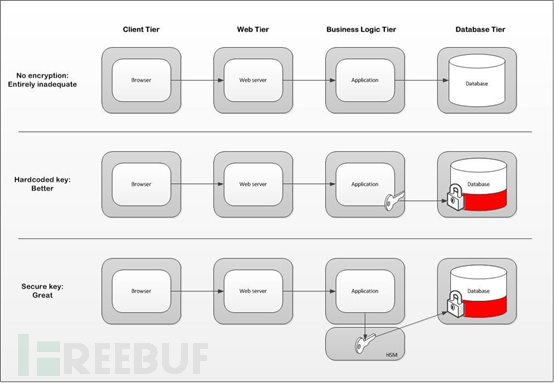
----按以上方式实现的数据库加密系统具有很多优点。首先，系统对数据库的最终用户完全透明，数据管理员可以指定需要加密的数据并根据需要进行明文/密文的转换工作；其次，系统完全独立于数据库应用系统，不需要改动数据库应用系统就能实现加密功能，同时系统采用了分组加密法和二级密钥管理，实现了“一次一密”；其三，系统在客户端进行数据加/脱密运算，不会影响数据库服务器的系统效率，数据加/脱密运算基本无延迟感觉。

----数据库加密系统能够有效地保证数据的安全，即使黑客窃取了关键数据，他仍然难以得到所需的信息，因为所有的数据都经过了加密。另外，数据库加密以后，可以设定不需要了解数据内容的系统管理员不能见到明文，大大提高了关键数据的安全性。

# 如何大幅提高加密数据库的安全性

**Web应用程序框架**

让我们先来恶补一下常见的Web应用程序框架。在一个典型的数据库加密Web应用程序中，应用程序会直接将密钥存储在服务器的某个角落。这样就被置于了逆向工程攻击之列，并且还会造成一些操作性的安全问题。这就是不使用HSM的后果，所以各位程序猿们赶紧使用HSM吧。

[](http://s6.51cto.com/wyfs02/M01/6C/62/wKioL1VIeOuSLNjMAACw-ASjZqQ668.jpg)

**科普：什么是HSM?**

一个HSM本质上就是一个协同处理器。就计算机架构而言，HSM可能含有冯诺依曼机的所有组件——包括储存、内存以及处理能力。计算机采用二进制算法和内存贮器后，指令和数据便可以一起存放在存贮器中，并可作同样处理，这样，不仅可以使计算机的结构大大简化，而且为实现运算控制自动化和提高运算速度提供了良好的条件。

HSM为主机本身提供专用离线加密服务。

[](http://s6.51cto.com/wyfs02/M02/6C/62/wKioL1VIeP6TMKxHAACFDr0EoEM991.jpg)

HSM致力于处理加密和保护加密进程，服务器内存不能访问重要数据，用户也无法看到明文密钥，因此可以保证应用程序和加密数据中间存在一个可信任的路径。这个时候你可能会想到还有可能被物理入侵呢，其实不用担心，HSM拥有一个防干扰的密封圈，可阻止攻击者的电子窃听和无线电监测。

**HSM可以用来防范什么?**

HSM可以防止攻击者从一个敏感数据库中盗取信息。如果攻击者获得了应用服务器(储存了明文密钥)的访问特权，密码可以重新找回。无论密码有多复杂，哪怕是被编译、被打包或者被更改，它也能给逆向工程了。从安全工程的角度来看，这不是一种最佳的方式。

从操作安全的角度看，让别人看到敏感、加密数据都是不明智的。其中包括终端用户，还有开发团队、系统管理员以及数据库管理员。从系统角度看，你同样不希望看到敏感数据从产品服务器传播到工作服务器或开发服务器上。如果你的数据库含有敏感数据，最好不要让任何人访问它。

**HSM是如何实施的?**

因为HSM是由一个应用程序开发团队部署，它是属于程序安全领域，而不是基础设施安全。硬件安全模块能够被应用程序用于建立一个安全、可靠的通路。应用程序必须使用HSM供应商提供的应用程序编程接口(API)建立专门执行加密的操作。这些API包括常见的加密功能，如对称算法、非对称算法和解密操作，散列消息身份验证代码，密码信息身份验证代码，RSA，数字签名算法，Diffie-Hellman密钥交换，随机数生成，素数生成以及格式保存生成。HSM程序编程接口可以由特定的供应商提供，比如IBM发布的常见加密体系结构，或遵循公共标准，如PKCS 11。显然，有人会需要做一些谨慎的购物来确保适当的功能可用。

**遵从FIPS-140-2标准**

然而HSM不受PCI合规的托管，它们为大多数支付方案使用。当前HSM的管理标准为来自美国国家标准和技术委员会，称为美国联邦信息处理标准(缩写：FIPS-140-2)。事实上，若你需要储存信用卡数据，你非常应该使用HSM。2009年PCI安全需求查看请点这里，2012年更新内容可以在这里查看。

当遵守FIPS-140-2标准的HSM被广泛运用于支付行业中，所有行业数据库的安全系数都得到了大幅提高。而需要牢记的是你的医疗记录对网络犯罪而言比你的信用卡更具有价值，所以请保持警惕。

**怎么开始使用加密?**

幸运的是，HSM认证产品拥有一个“票据交换所”，同时还有很多关于硬件安全模型的讨论，包括专为基于网络模块储存系统及其他定制的设计。然而，本文的目的是为HSM做一个高级的介绍，同时鼓励用户去SANS研究所和Stack Exchange(一系列问答网站，每一个网站包含不同领域的问题)做一些深入的阅读。

### 使用透明数据库加密 (Oracle)

### <http://www.oracle.com/ocom/groups/public/@otn/documents/webcontent/228942_zhs.htm>

当前主流Oracle安全增强方案包括前置代理、应用加密和Oracle自带加密选件TDE等。

# 如何在Oracle SecureFiles中加密数据

<http://netsecurity.51cto.com/art/201311/416979.htm>

# 加密你的SQLite

### <http://foggry.com/blog/2014/05/19/jia-mi-ni-de-sqlite/>

# [细说SQL Server中的加密](http://www.cnblogs.com/CareySon/archive/2012/04/01/SQL-SERVER-Encryption.html)

### <http://www.cnblogs.com/careyson/archive/2012/04/01/sql-server-encryption.html>

# MySQL数据库加密与解密

### <http://www.csdn.net/article/1970-01-01/312567>

# 数据库加密实现数据安全

**数据库加密**让**安全**技术又上了一个新台阶，确保了数据库系统中数据的安全。一位着名的信息安全专家曾经说过，信息安全无外乎三个方面。一是数据安全，二是系统安全，也就是时下较为热门的防火墙、入侵检测及VPN等边界安全，三是电子商务的安全。

数据加密、数据库加密的技术现状和未来的走向如何？这一领域的产品化解决方案是否可以满足行业和企业用户的需求？

**防火墙不是问题的全部**

以防火墙为代表的反入侵网络安全技术不等于信息安全的全部。在绝大多数信息系统中，核心数据和资料是以数据库的方式存储，没有加密的数据库就如同没有上锁的文件柜，对别有用心的人而言，剽窃、篡改易如反掌。因此，数据库的安全问题不容忽视。

数据库加密系统是为增强普通关系数据库管理系统的安全性而设计开发的。旨在提供一个安全适用的数据库加密平台，对通信和数据库存储的内容实施有效保护。它通过通信加密、数据库存储加密等安全方法实现了数据库数据存储和通信的保密和完整性要求，使得数据库以密文方式存储并在密态方式下工作，确保了数据安全。

**加密数据库迫在眉睫**

经过近几年的研究，我国数据库加密技术已经比较成熟。一些公司的数据库安全中间件技术，在保护用户原有的软硬件投资的前提下，可以有效实现数据库密态存储和查询。这一技术已经在实践中得到有效应用。

那么，为什么说数据库安全的问题是当前行业和企业用户迫在眉睫的安全问题呢？

首先，反拷贝的信息安全技术是真正可靠的技术，而数据库加密技术就是此类技术之一。敌对机构之间的情报战采用的手段之一往往是最直接的收买、拷贝方式。如果敌对方买通一名通常情况下不被人们注意的清洁工，即便这位清洁工不懂任何技术，但他只需要用被情报、特工人员经常使用的硬盘拷贝机，轻轻按一个按钮就可以在几分钟之内拷走全部数据！

而此时，防火墙、入侵检测等防护系统是起不到安全保卫作用的。而数据库加密后以密文方式存储，即使被窃取、被拷贝，机密数据也不会被敌对方获取，因为这时他们获取的，只不过是一堆几乎无法破解的密码。

其次，互联网的普及，移动通信、笔记本电脑的广泛使用对数据库安全构成了更大的威胁。无线通讯中随时有可能被截取、被仿冒、被侦听。无线上网、移动通讯在给人们带来方便和高效率的同时也带来了信息安全的重大隐患。

再次，数据库安全应该与操作系统、网络安全、CPU并重，共同组成信息安全战略的重心。只有制定标准，将数据库安全作为信息安全管理的一项重要内容进行实施和有效监控，才能使信息安全得到更进一步保障。

**7大特征实现数据库加密**

一般而言，一个行之有效的数据库加密系统主要有以下7个方面的功能和特性。

**1、身份认证：**用户除提供用户名、口令外，还必须按照系统安全要求提供其它相关安全凭证。系统可以选择使用终端密钥、用户USB Key等来增强身份认证的安全性。

**2、通信加密与完整性保护：**有关数据库的访问在网络传输中都被加密，通信的目的地还可以校验通信的完整性；通信一次一密的意义在于防重放、防篡改。

**3、数据库数据存储加密与完整性保护：**系统采用数据项级存储加密，即数据库中不同的记录、每条记录的不同字段都采用不同的密钥加密，辅以校验措施来保证数据库数据存储的保密性和完整性，防止数据的非授权访问和修改。

**4、数据库加密设置：**系统中可以选择需要加密的数据库列，以便于用户选择那些敏感信息进行加密而不是全部数据都加密。只对用户的敏感数据加密，可以提高数据库访问速度。这样有利于用户在效率与安全性之间进行自主平衡。

**5、多级密钥管理模式：**主密钥和主密钥变量保存在安全区域，二级密钥受主密钥变量加密保护，数据加密的密钥存储或传输时利用二级密钥加密保护，使用时受主密钥保护。

**6、安全备份：**系统提供数据库明文备份功能（为防止灾难发生，系统提供明文形式的数据库内容备份功能，以防止丢失密钥或数据造成灾难性后果）和密钥备份功能（用户可以同时使用数据库管理系统的备份功能和数据库加密系统的密钥备份功能来同时备份密文和密钥，在需要时进行恢复）。

**7、通用接口和广泛的平台支持：**系统采用开放的体系结构，支持标准SQL语句。

**1.2 对数据库加密系统基本要求**

(1) 字段加密；

(2) 密钥动态管理；

(3) 合理处理数据；

(4) 不影响合法用户的操作；

(5) 防止非法拷贝；

**相关知识库**

**数据加密的几种算法**

加密算法是一些公式和法则，它规定了明文和密文之间的变换方法。密钥是控制加密算法和解密算法的关键信息，它的产生、传输、存储等工作是十分重要的。

数据加密的基本过程包括对明文（即可读信息）进行翻译，译成密文或密码的代码形式。该过程的逆过程为解密，即将该编码信息转化为其原来的形式的过程。

**DES算法**DES（Data Encryption Standard）是由IBM公司在1970年以后发展起来的，于1976年11月被美国政府采用，DES随后被美国国家标准局和美国国家标准协会（American National Standard Institute,ANSI）承认。

**三重DES**DES的密码学缺点是密钥长度相对比较短，因此，人们又想出了一个解决其长度的方法，即采用三重DES。

**RSA算法**它是第一个既能用于数据加密也能用于数字签名的算法。它易于理解和操作，也很流行。算法的名字以发明者的名字命名：Ron Rivest, AdiShamir 和Leonard Adleman。

RSA算法它是第一个既能用于数据加密也能用于数字签名的算法。它易于理解和操作，也很流行。算法的名字就是发明者的名字：Ron Rivest, AdiShamir 和Leonard Adleman, 但RSA的安全性一直未能得到理论上的证明， RSA的安全性依赖于大数的因子分解，但并没有从理论上证明破译RSA的难度与大数分解难度等价。即RSA的重大缺陷是无法从理论上把握它的保密性能如何，而且密码学界多数人士倾向于因子分解不是NPC问题， RSA算法是第一个能同时用于加密和数字签名的算法，也易于理解和操作。RSA是被研究得最广泛的公钥算法，从提出到现在已近二十年，经历了各种攻击的考验，逐渐为人们接受，普遍认为是目前最优秀的公钥方案之一。

**AES算法**它是21世纪最新的取代DES算法的商用加密标准，在理论上，此加密方法需要国家军事量级的破解设备运算10年以上时间才可能破译。

AES是美国高级加密标准算法,将在未来几十年里代替DES在各个领域中得到广泛应用，尽管人们对AES还有不同的看法,但总体来说,AES作为新一代的数据加密标准汇聚了强安全性、高性能、高效率、易用和灵活等优点。AES设计有三个密钥长度:128,192,256位，相对而言，AES的128密钥比DES的56密钥强1021倍。AES算法主要包括三个方面：轮变化、圈数和密钥扩展。在理论上，此加密方法需要国家军事量级的破解设备运算10年以上时间才可能破译。

**1.4 数据库数据加密的实现**

使用数据库安全保密中间件对数据库进行加密是最简便直接的方法。主要是通过系统中加密、DBMS内核层(服务器端)加密和DBMS外层(客户端)加密。

在系统中加密，在系统中无法辨认数据库文件中的数据关系，将数据先在内存中进行加密，然后文件系统把每次加密后的内存数据写入到数据库文件中去，读入时再逆方面进行解密就，这种加密方法相对简单，只要妥善管理密钥就可以了。缺点对数据库的读写都比较麻烦，每次都要进行加解密的工作，对程序的编写和读写数据库的速度都会有影响。

在DBMS内核层实现加密需要对数据库管理系统本身进行操作。这种加密是指数据在物理存取之前完成加解密工作。这种加密方式的优点是加密功能强，并且加密功能几乎不会影响DBMS的功能，可以实现加密功能与数据库管理系统之间的无缝耦合。其缺点是加密运算在服务器端进行，加重了服务器的负载，而且DBMS和加密器之间的接口需要DBMS开发商的支持。

在DBMS外层实现加密的好处是不会加重数据库服务器的负载，并且可实现网上的传输，加密比较实际的做法是将数据库加密系统做成DBMS的一个外层工具，根据加密要求自动完成对数据库数据的加解密处理。

采用这种加密方式进行加密，加解密运算可在客户端进行，它的优点是不会加重数据库服务器的负载并且可以实现网上传输的加密，缺点是加密功能会受到一些限制，与数据库管理系统之间的耦合性稍差。

数据库加密系统分成两个功能独立的主要部件：一个是加密字典管理程序，另一个是数据库加解密引擎。数据库加密系统将用户对数据库信息具体的加密要求以及基础信息保存在加密字典中，通过调用数据加解密引擎实现对数据库表的加密、解密及数据转换等功能。数据库信息的加解密处理是在后台完成的，对数据库服务器是透明的。

按以上方式实现的数据库加密系统具有很多优点：首先，系统对数据库的最终用户是完全透明的，管理员可以根据需要进行明文和密文的转换工作；其次，加密系统完全独立于数据库应用系统，无须改动数据库应用系统就能实现数据加密功能；第三，加解密处理在客户端进行，不会影响数据库服务器的效率。

数据库加解密引擎是数据库加密系统的核心部件，它位于应用程序与数据库服务器之间，负责在后台完成数据库信息的加解密处理，对应用开发人员和操作人员来说是透明的。数据加解密引擎没有操作界面，在需要时由操作系统自动加载并驻留在内存中，通过内部接口与加密字典管理程序和用户应用程序通讯。数据库加解密引擎由三大模块组成：加解密处理模块、用户接口模块和数据库接口模块。

根据文件型数据库的特点，数据库的加密一般可以有三种方式：

密码等办法对数据库的增加、删除这些数据访问加以控制，防止不合法用户读写数据。但由于DBF文件采用的是ASCII明文存储，如果非法用户不使用程序读取数据，而是直接使用WINHEX、UltreaEdit等工具--甚至是DOS下的DEBUG来打开数据库文件，都可以直接阅读到数据库中的内容。因此，为了使得DBF文件可以更加安全，必须对数据的载体DBF 文件进行加密保护，主要的目的还是把ASCII码的信息变成不可直接阅读的数据。在文件型数据库中，记录的长度一般比较短，数据存储的时间长，相应的密钥保存时间也随数据生命周期而定。如果在库内使用同一密钥，则保密性差；如果不同记录使用不同的密钥，则密钥太多，管理相当复杂。因此，应该针对数据库的特点，研究相应的加密方法和密钥管理方法。  
根据文件型数据库的特点，数据库的加密一般可以有三种方式：  
**库外加密**  
考虑到文件型数据库系统是基于文件系统的，因而库外加密的办法，应该针对文件IO操作或操作系统而言的，因为数据库管理系统与操作系统的接口方式有三种：一是直接利用文件系统的功能；二是利用操作系统的I/O模块；三是直接调用存储管理。所以在采用库外加密的方法时，可以将数据先在内存中使用DES、RSA等方法进行加密，然后文件系统把每次加密后的内存数据写入到数据库文件中去（注意是把整个数据库普通的文件看待，而不是按数据关系写入），读入时再逆方面进行解密就可以正常使用了。这种加密方法相对简单，只要妥善管理密钥就可以了。缺点对数据库的读写都比较麻烦，每次都要进行加解密的工作，对程序的编写和读写数据库的速度都会有影响。  
**库内加密**  
如果从关系型数据库的各个方面出发，很容易形成库内加密的思想。关系型数据库的关键术语有：表、字段、行和数据元素。基本上可以针对这几方面形成一种加密的方法。   
（1）以表为单位：对于文件型数据库来说，一个文件只有一张表，因而对表的加密可以说是对文件的加密了。对过更改文件分配表(FAT)中的说明等手段可以实现对文件的简单加密，但这种加密方式涉及到文件系统底层，误操作容易造成FAT混乱，而且与文件系统格式有关，因而通常不宜采用。  
（2）以记录或字段（即二维表的行或列）为单位加密：通常情况下，我们访问数据库时都是以二维表方式进行的，二维表的每一行就是数据库的一条记录，二维表的每一列就是数据库的一个字段。如果以记录为单位进行加密，那么每读写一条记录只需进行一次加解密的操作，对于不需要访问到的记录，完全不需要进行任何操作，所以使用起来效率会高一些。但是由于每一个记录都必须有一个密钥与之匹配，因此产生和管理记录密钥比较复杂。以字段为单位的加密分析与以记录为单位的加密情况相似。 (3)以数据元素为单位加密：由于数据元素是数据库库内加密的最小单位，因而这种加密方式最彻底的但也是效率最低的。每个被加密的元素会有一个相应的密钥，所以密钥的产生和管理比记录加密方式还要复杂。  
**硬件加密**硬件加密主要是相对于软件加密的，是指在物理存储器与数据库系统之间加上一层硬件作为中间层，加密和解密的工作都由添加的硬件完成。不过由于添加的硬件与原计算机硬件之间可能存在着兼容问题，还有在进行控制读写的时候存在着繁琐的设置，所以这种应用起来也不会太广泛。