Oracle 白皮书
2012 年 7 月

Oracle Advanced Security
透明数据加密最佳实践

引言

本文介绍使用 Oracle Advanced Security 透明数据加密 (TDE) 的最佳实践。Oracle Advanced Security TDE 能够对存储介质上的敏感应用程序数据进行加密，且该操作对应用程序本身是完全透明的。TDE 负责处理与私有及公有数据的隐私和安全要求（如 PCI 和 California SB1386）有关的加密需求。Oracle Advanced Security TDE 列加密是在 Oracle Database 10g 第 2 版中引入的，支持对包含信用卡号或社会保险号的应用程序表列进行加密。Oracle Advanced Security TDE 表空间加密是在 Oracle Database 11gR1 中引入的。

重要概念

主加密密钥 — 对用于列加密和表空间加密的辅助数据加密密钥进行加密的加密密钥。主加密密钥是 Oracle Advanced Security 双层密钥架构的一部分。

统一的主密钥 — 在 Oracle Database 11g 第 2 版中，首次执行重新设定密钥操作后会生成统一的主加密密钥。统一的主密钥可轻松重新设定（轮换）。

表密钥 — 有时称为列密钥，该密钥用于对指定表中的一个或多个特定列进行加密。表密钥是在 Oracle Database 10g 第 2 版中引入的。这些密钥存储在 Oracle 数据字典中，并使用主加密密钥进行加密。

表空间密钥 — 用于对表空间加密的密钥。这些密钥使用主密钥加密，并存储在加密的表空间的表空间头中，以及每个操作系统的头（属于加密的表空间的文件）中。

Wallet — 位于数据库之外的 PKCS#12 格式文件，根据 PKCS#5 中定义的口令式加密方法进行加密。用于存储 TDE 主密钥。

高级加密标准 (AES) — 联邦信息处理标准 (FIPS) 197 中定义的对称式密码算法。AES 提供 3 种经过批准的密钥长度：256、192 和 128 位。

SPARC T4 和 Intel 提供的硬件加密加速

支持 AES-NI（一组高级加密标准新指令）的 Intel® CPU 提供用于 TDE 表空间加密的硬件加密加速。Oracle Database 11gR2 (11.2.0.2) TDE 表空间加密自动检测基于硬件的加密加速并将其用于数据**解密**；要实现硬件加速**加密**，需要安装补丁 10296641。使用 Oracle Database 11.2.0.3，硬件加密加速支持可扩展至支持 AES-NI 的 Intel CPU 上的 Solaris 11 x64，以及 T4 上的 Solaris 11 SPARC。不支持将硬件加密加速用于 TDE 列加密。

CPU 类型	ORACLE DATABASE 11.2.0.2 （需要安装补丁 10296641）	ORACLE DATABASE 11.2.0.3
支持 AES-NI 的 Intel® （所有支持 AES-NI 的 CPU）	Oracle Linux • Exadata X2 • 数据库机(*) • 11.2.0.2 在 Linux 和支持 AES-NI 的 Intel 上的任何其他部署 仅在 Exadata X2 (11.2.2.4) 上包含补丁包 11 的 Oracle Solaris 11 ***Express*** 无其他部署，因为 11.2.0.2 未针对 Solaris 11 ***Express*** 进行认证	添加 Solaris 11 x64
SPARC T4		Solaris 11 SPARC： • SPARC SuperCluster • 11.2.0.3 在 Solaris 11 SPARC 和 T4 上的任何其他部署

(*)：Oracle Database 11.2.0.2.4 中包含补丁 10296641，它是 Oracle 数据库机中使用的补丁级别。

在 Oracle Enterprise Manager 中管理透明数据加密

大多数数据库维护和配置工具都集成在一个称为 Enterprise Manager 的便捷、易用的 Web 界面中。默认安装 Oracle Enterprise Manager **Database** Control，并将其用于管理单个、本地数据库实例。Oracle Enterprise Manager **Grid** Control 是一个基础架构，允许从一个中央控制台监视、配置和维护许多分布式数据库。

以下命令将启动 Oracle Enterprise Manager Database Control：

```
$ emctl start dbconsole
```

使用浏览器访问 https://<hostname>:<port>/em 并提供具有足够权限管理数据库的用户（如“SYSTEM”）的用户名和口令。

在 Oracle Enterprise Manager Database Control 的主页面上，单击“Server”选项卡，然后在接下来的页面上，单击“Security”组内的“Transparent Data Encryption”，以到达透明数据加密的主页。

透明数据加密和 Oracle Database Vault

如果您的数据库使用 Oracle Database Vault 进行保护，则实施职责分离，其中包含在 Enterprise Manager 中控制用户的授权。为了使“SYSTEM”能够管理透明数据加密，“SYSTEM”必须是 Oracle Database Vault 中“Data Dictionary Realm”的“Participant”或“Owner”。该更改需要由具有 Database Vault“owner”角色的用户来完成。Oracle 建议对满足安全要求的各个用户使用自定义的 DBA 角色，而不是功能强大的“SYSTEM”用户。这些自定义的 DBA 角色可能需要是“Data Dictionary Realm”的“Participant”或“Owner”，也可能不需要，具体取决于其权限。

透明数据加密密钥架构

加密密钥与加密算法结合使用来对数据加密。Oracle Advanced Security TDE 使用双层加密密钥架构，该架构由一个主密钥和一个或多个表密钥和/或表空间密钥构成。表密钥和表空间密钥使用主密钥加密。主密钥存储在 Oracle Wallet 中。

密钥生成和备份

存储在 Oracle Wallet 中的 TDE 主密钥，由 Oracle 在初始配置 TDE 期间生成。主密钥由 Oracle 数据库内的伪随机数生成器生成。

始终备份与主密钥相关的钱包

- 初始创建主密钥后**立即**备份
- **只要**主密钥发生更改就备份
- 更改钱包口令**之前**备份

钱包是一个关键组件，应在安全的位置、本地**和**异地备份。

密钥交换/轮换

在 TDE 列加密中，主密钥和表密钥都可单独重新设定，从而能够细粒度实施各种安全策略。重新设定主密钥不影响应用程序的性能或可用性，因为它仅需要解密和重新加密表密钥，与相关的加密应用程序数据无关。而重新设定表密钥则需要认真规划，因为相关的应用程序数据必须首先解密，随后再使用新的表加密密钥重新加密。更改表密钥相当于执行全面的表更新。**升级到 Oracle Database 11gR1** 之后，执行 TDE 主密钥重新设定操作将在 Oracle Wallet 中透明地创建单独的 TDE 表空间加密主密钥。使用 ENCRYPT 语法创建的表空间将使用在每个表空间头中存储的表空间密钥对任何关联的数据文件加密。表空间密钥本身将使用新的表空间主密钥加密。**升级到 Oracle Database 11g 第 2 版**之后，执行 TDE 主密钥重新设定操作会将两个现有的主密钥合并成一个统一的主加密密钥，或创建一个新的统一主加密密钥。统一的主加密密钥可用于 TDE 列加密和 TDE 表空间加密，且可轻松重新设定。

请注意 Oracle Database 11gR1 中的限制，即无法重新设定表空间主密钥和表空间密钥。如果需要为指定的加密表空间重新设定密钥，Oracle 建议将数据移至一个新的加密表空间。请参见第 9 页中有关表空间重新设定密钥限制部分，了解有关将数据移至新的表空间以实现重新设定密钥操作的建议。Oracle Database 11g 第 2 版允许轮换统一的主加密密钥。

重新设定密钥支持				
	TDE 列加密		TDE 表空间加密	
	主密钥	表密钥	主密钥	表空间密钥
Oracle Database 10gR2	是	是	n/a	n/a
Oracle Database 11gR1	是	是	否	否
Oracle Database 11g 第 2 版	是 (*)	是	是 (*)	否

(*)：统一的主加密密钥可用于 TDE 列加密和 TDE 表空间加密

TDE 钱包管理

目录和文件权限

使用 Oracle Wallet 时，Oracle 建议限制关联的文件和目录权限。此外，设置钱包时还应设置强口令。Oracle Wallet 是用于存储（统一）TDE 主加密密钥的默认外部安全模块。

Oracle 建议将 Oracle Wallet 放置在 $ORACLE_BASE 目录树之外，以避免意外将钱包与加密数据一起存储在备份磁带上，例如：

```
/etc/ORACLE/WALLETS/<$ORACLE_UNQNAME>
```

由于 /etc 由“root”所有，因此由“root”创建这些目录；当创建完成后，将所有权更改为“oracle:oinstall”并将权限设置为仅“oracle”：

```
# cd /etc
# mkdir -pv ORACLE/WALLETS/DB01
# chown -R oracle:oinstall ORACLE
# chmod -R 700 ORACLE
```

将 sqlnet.ora 中的 ENCRYPTION_WALLET_LOCATION参数设置为新创建的目录：

```
ENCRYPTION_WALLET_LOCATION = (SOURCE = (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = /etc/ORACLE/WALLETS/$ORACLE_UNQNAME/)))
```

初始化钱包，然后使用 Enterprise Manager 或 SQL*Plus命令行界面添加主加密密钥：

```
SQL> alter system set encryption key identified by “password”;
```

成功创建钱包和主密钥后，将钱包文件的权限从初始值（由“oracle”用户的“umask”确定）降低为：

```
$ cd /etc/ORACLE/WALLETS/<$ORACLE_UNQNAME>
$ chmod 600 ewallet.p12
```

强烈建议备份数据库的同时始终备份钱包，但不将钱包与数据库备份包含在同一个介质中。此外，还建议对其内容执行任何操作（无论是重新设定主密钥，还是更改钱包口令）之前都备份钱包。

避免意外删除 TDE 钱包

为防止 Oracle TDE 钱包被意外删除，请将其设置为“immutable”（ext2、ext3 和 ext4 文件系统上的 Linux；OCFS）。

初次创建加密钱包（也可以是（本地）自动打开式钱包）后，导航至存储 Oracle Wallet 的目录，然后使用以下命令设置“不可变”位：

```
# chattr +i ewallet.p12
# chattr +i cwallet.sso
```

任何试图删除钱包的操作（由 root 用户或任何其他用户执行）将失败；写入钱包的重新设定密钥操作也将失败，因此对于重新设定密钥操作，必须取消“不可变”位的设置：

```
# chattr -i ewallet.p12
# chattr -i cwallet.sso
```

成功执行重新设定密钥操作后，重新启用“不可变”位。

在 Solaris 10 和 Solaris 11 中，在 ZFS 上设置或取消设置不可变位的命令为：

```
# chmod S+ci ewallet.p12                # chmod S-ci ewallet.p12
# chmod S+ci cwallet.sso                # chmod S-ci cwallet.sso
```

（本地）自动打开式钱包与加密钱包

加密钱包（“ewallet.p12”）使用钱包口令为钱包加密，从而为主密钥提供强大保护，钱包口令遵循口令式加密的 PKCS#5 标准。打开钱包是一个手动操作，且必须执行才可使主加密密钥应用于数据库。另外，还可以选择将主密钥复制到“自动打开式”钱包。该操作可以使用以下三种方式完成：Oracle Enterprise Manager、Oracle Wallet Manager 或“orapki”实用程序：

```
$ orapki wallet create –wallet <wallet_location> -auto_login
```

该命令创建自动打开式钱包（“cwallet.sso”）。为了在使用自动打开式钱包时极大增强安全性，从 Oracle Database 11.1.0.7 开始，可创建**本地**自动打开式钱包；除了在其上创建该钱包的计算机之外，无法在任何其他计算机上打开：

```
$ orapki wallet create –wallet <wallet_location> -auto_login_local
```

如果要将钱包集中存储在 ACFS 中，则无法在 Oracle RAC 中使用**本地**自动打开式钱包。

重要信息 — 请勿删除原始加密钱包。重新设定主密钥需要提供原始加密钱包。重新设定主密钥时，相应的（本地）自动打开式钱包会自动更新。

在与加密数据存储位置不同的位置备份自动打开式钱包。将自动打开式钱包与加密数据存储在一起缺少安全性，因为被盗或放错位置的磁带或磁盘上的钱包和数据没有保护。

强钱包口令

钱包口令对于提供强大的安全性至关重要。如果钱包口令受到损害，能够访问操作系统的用户只需复制数据库文件和钱包并使用钱包口令，即可将主密钥用于数据库并对加密的应用程序数据解密。不难看出，口令不仅需要强大，而且还要好记，因为如果忘记口令，则无法恢复。

形成强大且好记口令的一种方法是采用易记句子中每个单词的首个字符：“*I work from 9 to 5 almost every day of the week*”得出“Iwf9t5aedotw”，它满足好口令的普遍需求：它包含数字以及大写字符和小写字符，且长于建议的最小长度 10 个字符。这个句子非常好记，但您根本不需要记住这个复杂的口令。

了解有关钱包口令的拆分

使用 Enterprise Manager 时，始终屏蔽钱包口令，因此不仅易于向 DBA 隐藏钱包口令，而且可以在不同的管理人之间轻松拆分：管理人 A 输入口令的前半部分，然后管理人 B 输入口令的后半部分，且管理人 B 看不到管理人 A 在口令域中键入的内容。

在 Oracle Database 10gR2 中，SQL*Plus 命令行上的钱包口令明码显示，因此不可能进行口令拆分。如果客户要将“了解有关加密密钥的拆分”需求转化为“了解有关钱包口令的拆分”，以下脚本可能提供了一个解决方法：

在数据库中创建用户“Sentinel”，且仅为其授予“create session”和“alter system”权限

按照该文档中的说明创建安全的外部口令存储 (SEPS)。如该文档所述，在 tnsnames.ora 中创建一个条目，将其命名为“keyholder”；使用“$ tnsping keyholder”确认该条目正确无误。将用户“sentinel”的凭证添加到 SEPS：$ mkstore -wrl .-createCredential keyholder sentinel <password> 尝试使用“$ sqlplus /@keyholder”连接到数据库

该脚本（“set_key.sh”）在定义的位置（如果不存在）创建新钱包，然后将新 TDE 主加密密钥添加到钱包，或重新设定主加密密钥：

```
#!/bin/bash
#
get_pwd1(){read -s -p "1st half of password:" pwd1}
get_pwd2(){read -s -p "2nd half of password:" pwd2}
set_key(){sqlplus /@keyholder @set_key.sql $pwd1$pwd2}

get_pwd1
get_pwd2
set_key
```

SQL 脚本“set_key.sql”：

```
set termout off;
alter system set encryption key identified by "&1";
```

```
set termout on;
exit
```

可以编写类似的脚本来打开钱包，或使用 11.1.0.7 或更高版本中的“orapki”命令行工具更改钱包口令（请参见下一段）。

Oracle 建议在数据库服务器上创建钱包和初始化主密钥。如果计划使用远程计算机执行该操作，请在客户端与数据库服务器之间启用网络加密，以便两台计算机之间的通信是安全的。

更改钱包口令

更改现有钱包的口令之前，请确保已备份现有钱包。更改口令之后，请验证是否可使用新口令打开钱包。

更改口令与更改主密钥无关。伪随机数生成器生成主密钥，而钱包口令用作对钱包加密的密钥。在 Oracle Database 第 11.1.0.7 版之前，更改钱包口令需要使用 Oracle Wallet Manager。更改现有钱包的口令之前，请确保已备份该钱包。更改口令之后，请验证是否可使用新口令打开钱包。如果使用新口令无法打开钱包，只需还原钱包的备份副本，然后重新尝试更改口令。自 Oracle Database 第 11.1.0.7 版起，“orapki”实用程序得到了增强，支持从命令行进行钱包口令更改：

```
$ orapki wallet change_pwd -wallet <wallet_location>
```

同一个主机上的多个数据库

如果同一台服务器上安装了多个 Oracle 数据库，它们必须访问各自的 TDE 钱包。不支持在独立的实例之间共享同一个钱包，这有可能导致加密数据丢失。

如果多个数据库共享同一个 ORACLE_HOME，它们也可以在 $TNS_ADMIN 中共享同一个 sqlnet.ora 文件。为了访问各自的钱包，ENCRYPTION_WALLET_LOCATION 的 DIRECTORY 条目需要将每个数据库指向自己的钱包位置：

```
DIRECTORY = /etc/ORACLE/WALLETS/$ORACLE_UNQNAME
```

/etc/ORACLE/WALLETS/ 下的子目录名称反映了各个数据库的 ORACLE_UNQNAME 名称。

如果数据库不共享同一个 ORACLE_HOME，它们也将具有各自的 sqlnet.ora 文件，且这些文件必须指向各自的子目录。

TDE 表空间加密

一个 Oracle 数据库至少包含两个逻辑存储单元（我们将其称为表空间），它们共同存储数据库的所有数据。Oracle 数据库中的每个表空间均包含一个或多个文件（我们将其称为数据文件），这些文件的物理结构与运行 Oracle 数据库的操作系统保持一致。几乎所有数据库都包含多个附加表空间来存储应用程序特定的数据。

在 Oracle Database 11g 中，可将新的表空间定义为加密。将表空间定义为加密意味着，在操作系统上创建的物理数据文件将加密。默认情况下，新表空间中定义的任何表、索引和其他对象将加密，且无额外的存储空间需求。在数据读取过程中，Oracle 数据库将在数据到达数据库内存 (SGA) 之前自动将其解密。移出 SGA 然后写入文件系统的数据将被加密。TDE 表空间加密支持现有索引和外键按照加密前的方式继续运行，从而提供最佳性能。执行计划保持不变，完全消除了识别要加密的个别列的需求。

经过 TDE 表空间加密认证的管理软件

在 Oracle Database 11.1.0.7 和 Oracle Database 11g 第 2 版中，以下 Oracle 管理软件已经过 TDE 表空间加密认证：

- Oracle E-Business Suite（参见 http://blogs.oracle.com/stevenChan/certifications.html，了解最新更新）
- Oracle PeopleSoft Enterprise 8.48 和更高版本（迁移脚本和详细的实施指南）
- Oracle Siebel CRM 8.0 和更高版本
- Oracle JD Edwards EnterpriseOne
- SAP 6.40_EX2 和更高版本（仅限 Oracle Database 11g 第 2 版，请参见 SAP 说明 974876）
- RETEK Retail Sales Audit 13.1.5
- Primavera P6

经过执行内部基准测试以及客户反映，最终用户响应时间的性能影响为 4% 至 8%，CPU 利用率提高了 1% 至 5%。

将应用程序数据移至加密表空间

Oracle Database 11g 仅支持对新表空间加密。可以使用以下步骤将应用程序数据从现有的非加密表空间迁移至新的加密表空间：

1) 使用标准的备份步骤备份数据库

2) 使用 Oracle 数据泵导出（“expdp”）导出所有应用程序表空间，也可以选择压缩转储文件以进行快速处理以及减少存储空间

3) 创建明文表空间的加密版本：

   a. 使用“dbms_metadata.get_ddl”，提取用于创建应用程序表空间的原始 DDL（数据定义语言），然后使用 spool 将其生成 SQL 脚本

   b. 将“ENCRYPTION [using '<algorithm>'] DEFAULT STORAGE(ENCRYPT)”附加到每个“CREATE TABLESPACE”命令，无需更改任何其他参数。

   c. 删除原始的未加密表空间，可以使用命令：

      ```
      SQL> drop tablespace <name> including contents and datafiles;
      ```

      或：

      ```
      SQL> drop tablespace <name> including contents keep datafiles;
      ```

      （如果要使用“sdelete”或“shred”等操作系统级命令安全删除旧的明文数据文件）。

   d. 通过运行编辑的脚本来创建加密表空间

4) 使用 Oracle 数据泵导入（“impdp”）导入所有应用程序表空间

5) 验证应用程序正常工作

如上过程需要停机，但这并不总是可行的。要在迁移到加密表空间的同时使应用程序仍保持完全可用，Oracle 建议使用联机表重定义，它是 Oracle 企业版的一项成熟的高可用性特性。提供了一个随时可运行的脚本（附带详细的实施指南）。该脚本可轻松修改以反映您自己的迁移需求。

Oracle Database 11gR1 中的表空间重新设定密钥限制

在 Oracle Database 11.1.0.7（无论是全新安装还是从旧版本升级，无论是否存在存储用于 TDE 列加密的主加密密钥的 Oracle Wallet）中，首次执行“set key”或重新设定密钥操作都会创建一个用于 TDE 表空间加密的额外主加密密钥。用于 TDE 表空间加密的主加密密钥在 sqlnet.ora 所指定位置的 Oracle Wallet 中创建。Oracle Database 11gR1 不支持重新设定 TDE 表空间主密钥。如果有必要更改与加密表空间关联的主密钥，可以将上述脚本略微修改后应用于单个或所有应用程序表空间。或者，先使用数据泵实用程序从加密表空间提取应用程序数据，再创建新的加密表空间，然后使用 Oracle 数据泵导入 (impdp) 将数据导入到这个新的加密表空间。

1) 使用标准的备份步骤备份数据库。
2) 使用 Oracle 数据泵“expdp”导出应用程序表空间，可以选择压缩转储文件以及使用口令对其加密（请勿使用当前的主密钥加密转储文件）。
3) 提取用于构建加密表空间的 DDL（使用“dbms_metadata.get_ddl”），并使用 spool 生成 SQL 文件。
4) 删除原始的加密应用程序表空间，采用“including contents and datafiles”这个命令。
5) 使用第 2 步中创建的脚本构建新的加密表空间，该表空间现在使用新的表空间密钥加密。
6) 使用 Oracle 数据泵“impdp”导入应用程序表空间。
7) 验证应用程序正常工作。

TDE 列加密

TDE 列加密透明地对写入应用程序表列的敏感数据加密。此操作可通过以下两种方式完成：在 Enterprise Manager Database Control 中将敏感列标记为“encrypted”，或将“encrypt”关键字附加到 SQL DDL 语句。现有数据类型保持不变，因此加密对现有应用程序是完全透明的。每个包含一个或多个加密列的表都有自己的表加密密钥；这些密钥存储在数据字典中，并使用主密钥加密。

将数据写入加密列后，在将敏感值写入磁盘之前即刻将其加密。当授权的用户从数据库选择数据时，自动解密数据并以明文呈现。与 TDE 表空间加密一样，TDE 列加密能防止特权操作系统用户直接访问介质，以及丢失或放错位置的磁带和磁盘驱动器。

为在处理加密数据时提高性能，每个表均有自己的表密钥，该密钥用于相应特定表的所有加密列。在处理加密数据之前使用主密钥对这些表密钥解密，且这些表密钥在整个事务期间保持解密状态。

经过 TDE 列加密认证的 Oracle 管理软件

在 Oracle Database 10gR2 和 11g（建议 10.2.0.5、11.1.0.7 或 11.2.0.2/3）中，以下 Oracle 管理软件已经过 TDE 列加密认证：

- Oracle E-Business Suite
  （请参见 http://blogs.oracle.com/stevenChan/certifications.html，了解最新更新）
- Oracle PeopleSoft Enterprise 8.46 和更高版本
- Oracle Siebel CRM 7.7+
- Oracle Financial Services (iFlex)：FlexCube 10.0
- Oracle Retail Applications (Retek)：Retail Sales Audit (ReSA)：
  - ReSA 12.0 和 13.0（在 Oracle Database 10gR2 10.2.0.4 和更高版本中）
  - ReSA 13.1（在 Oracle Database 11gR1 11.1.0.7 中）
- Oracle Internet Directory 10.1.4.2
- SAP 6.40 和更高版本（SAP 说明 974876）

识别敏感列

识别包含社会保险号和信用卡号等敏感数据的表和列可能非常困难，在大型应用程序中尤其如此。一个可能有用的技巧是在 Oracle 数据字典中搜索常用于存储这类信息的列名和数据类型。以下示例使用 SQL 识别可能包含社会保险号的列：

```
SQL> select column_name, table_name, data_type from user_tab_cols where
column_name like '%SSN%' or
column_name like '%SECNUM%' or
column_name like '%SOCIAL%';
```

执行该语句的用户是拥有应用程序表的主用户或模式。该技巧也可用于其他类型的信息，只需在 SQL 文本中替换为其他字符串（如“PIN”）即可。

加密索引列

如果某索引列用于等值搜索且索引类型为 B 树索引（普通索引，不是降序索引），可以对该列加密。仅当使用“`no salt`”语法对一个或多个列加密时，才可在这些加密列上构建索引。如果使用加密列构建索引，索引中的相应列也将被加密。处理 SQL 语句之前，SQL 文本中指向加密列的值将使用目标表的表密钥进行加密；数据库检查索引，与加密值进行匹配，在索引中查找 rowid，显示基表中的相应行。遵循此步骤后，等值搜索对性能的影响可能会大幅降低。

请注意，范围扫描（“between”子句）无法使用加密索引。但是，个人身份信息 (PII) 极少在范围扫描操作中使用。根据表定义加密算法（默认为 AES192），即使可以在“create table”语句中的任何列定义生成语句也是如此。

减少存储开销

TDE 列加密在存储要求方面不同于 TDE 表空间加密；加密后，加密值可以比明文值长 1 到 52 个字节。默认情况下，TDE 列加密将“salt”和完整性检查（消息验证代码，简称 MAC）添加到每个加密值。我们在这里探讨的是何时可以不用使用这些安全特性以节省存储开销。

基本加密是确定性的，这意味着给定的明文将始终加密成相同的密文。当列中的值不唯一时，这种属性将缺乏安全性。例如，当某列包含有关罕见疾病的敏感患者信息时，大多数患者将输入“No”，仅少数输入“Yes”。“Yes”的所有密文将相同，否定答案的密文也如此。即使将答案加密，患有这种罕见疾病的人也很容易被识别。要超越这一限制，请使用“salt”（随机的16 字节字符串）修改每个明文值。相同明文输入得到的输出会生成完全不同的密文。但是，如果您可以保证列中的所有值都是独一无二的（例如，已将“UNIQUE INDEX”应用于该列的情况），可以在对列加密时设置“NO SALT”参数，以将存储开销减少到每个单元 16 个字节。加密盐是按列定义的；一个表可以包含同时使用或不使用加密盐加密的多列。

为进一步提供保护，还可以将具有 20 个字节的完整性检查附加到每个加密值，以检测密文是否被篡改。自 Oracle Database 10.2.0.4 和 11.1.0.7 起，可以将“NOMAC”参数用于 TDE 列加密，以便不再生成和存储这些额外的 20 个字节。“NOMAC”是基于表定义的；即使可以对一个或多个列指定“NOMAC”，它也应用于表中的所有加密列。

最后一种与列加密相关的存储开销是不可避免的。如果使用 AES 加密，每个明文值补全为 16 个字节；如果使用 3DES168 加密，则补全为 8 个字节：如果某个明文值需要 9 个字节的存储空间，它将扩展到 16 个字节；如果需要 16 个字节，则扩展到 32 个字节（使用 3DES168 时为 24 个字节），依此类推。

强烈建议在 11.1.0.7 中针对 TDE 列加密安装补丁 8421211，以便显著提高特定查询类型的性能，以及将 TDE 列加密应用于属于组合索引（其中，除待加密列之外的其他列用于函数索引）的某列时更正其行为。如果补丁不适用于您的版本/平台组合，请联系 Oracle 支持部门。

加密 GB 级表和 TB 级表中的列

有时，表包含许多行但应用程序停机时间很短，因此对一列或多列加密所需的“UPDATE”操作将需要很长时间，即使对表的“READ”访问在表的更新过程中仍然可行也是如此。

通常，这些表是您业务的基础，包含需要通过加密进行保护的个人身份信息 (PII)，但这些表不断更新，这些更新在不中断业务的情况下无法进行。

对于这些表，Oracle 提供了联机表重定义，它提供了一种透明方法，更改表特征的同时源表完全可用。有关详细步骤，请阅读相关文档，但以下给出了简单介绍：

1) 创建一个临时表，该表包含希望源表在执行相应过程后所具有的所需特征，例如：一个列加密，而其他所有列保持不变。
2) 启动重定义进程，同时源表完全可用。
3) 执行最后一步（源表脱机片刻；对用户和应用程序是透明的，无数据丢失！）后，删除临时表。

禁用并重新启用透明数据加密

通常，删除钱包（无论数据是否已加密），然后要创建新钱包和主加密密钥时，会显示错误消息“ORA-28374:typed master key not found in wallet”。

接下来的步骤**无法**用于恢复或替换丢失的钱包、钱包口令或 TDE 主加密密钥；它是为满足以下条件的管理员准备的：在某一时刻使用 TDE 主加密密钥创建钱包，从未对数据加密，决定不使用 TDE 并删除钱包，然后决定最终使用 TDE。

安装补丁 8682102 后，执行日志切换以在所有重做日志之间循环，然后创建新钱包和 TDE 主加密密钥。

TDE 表空间加密还是 TDE 列加密？

在 Oracle Database 11gR1 中，安全管理员或 DBA 可以在 TDE 表空间加密和 TDE 列加密之间进行选择；以下提供了一些指导：

选择 TDE 列加密的条件：	选择 TDE 表空间加密的条件：
敏感信息的位置已知	敏感信息的位置未知
所有应用程序列中需要加密的列不到 5%。	大多数应用程序数据被认为是敏感数据，或 您的行业需要满足**多个**国内和国际安全和隐私要求
数据类型和长度受 TDE 列加密的支持	并非所有包含敏感信息的数据类型都 受 TDE 列加密的支持
需要加密的列不是外键列	需要加密的列是外键列
需要加密的列上的索引是普通的 B 树索引	需要加密的列上的索引是函数索引
应用程序不对加密数据执行范围扫描	应用程序搜索一定范围的敏感数据
每个加密值将存储空间增加 1 至 52 个字节	不接受存储空间增加
性能影响取决于加密列所占的比例、选择或更新加密值的频率、 加密数据的大小以及其他变量。	恒定性能影响低于 10%
	如果您希望从硬件加密加速获益
	如果您希望同时享受加密 和压缩带来的好处。

加密数据的明文副本

在表的使用期中，数据可能分段、重新排列、排序、复制以及在表空间内移动；这会在数据库文件内留下数据的“幽灵副本”。对现有列加密时，仅对最新的“有效”副本加密，而在幽灵副本中留下较旧的明文版本。如果绕过数据库的访问控制而直接访问包含表空间的数据文件（如使用十六进制编辑器），旧的明文值可能有一段时间是可见的，直到这些块被数据库覆盖。要将该风险降至最低，请遵循以下建议：

1) 使用标准的备份步骤备份数据库。
2) 在新的数据文件中创建新的表空间 (`CREATE TABLESPACE... `)
3) 对原始表中的敏感数据加密 (`ALTER TABLE ... ENCRYPT`)
4) 将原始表空间中的所有表（包含或不包含加密列）移至新的数据文件 (`ALTER TABLE ... MOVE ...`)
5) 验证应用程序正常工作。

6) 删除原始表空间 (DROP TABLESPACE)。请勿使用“and datafiles”参数；Oracle 建议对操作系统级别的操作使用更有力的方法。

7) 使用针对您平台的“shred”或其他命令来删除操作系统级别的旧数据文件。

建议执行最后一步，以便降低能够发现数据库文件幽灵副本（操作系统或存储固件生成的）的概率。

查证

为应审计人员请求等情况提供加密证明，Oracle 提供了记录数据库加密状态的视图。对于 TDE 列加密，请使用视图“dba_encrypted_columns”，其中列出了所有加密列的所有者、表名、列名、加密算法和加密盐。对于 TDE 表空间加密，以下 SQL 语句列出了所有加密表空间及其加密算法和相应的加密数据文件：

```
SQL> SELECT t.name “TSName”, e.encryptionalg “Algorithm”, d.file_name
“File Name” FROM
v$tablespace t, v$encrypted_tablespaces e, dba_data_files d WHERE
t.ts# = e.ts# and t.name = d.tablespace_name;
```

以下 SQL 语句列出了表所有者、加密表空间中的表和加密算法：

```
SQL> SELECT a.owner “Owner”, a.table_name “Table Name”, e.encryptionalg
“Algorithm”, FROM
dba_tables a, v$encrypted_tablespaces e WHERE a.tablespace_name in
(select t.name from v$tablespace t, v$encrypted_tablespaces e where t.ts#
= e.ts#);
```

Oracle Data Guard

物理备用数据库

自 Oracle Database 10g 第 2 版的首次发行起，Oracle Data Guard 物理备用数据库可与透明数据加密配合使用。将重做日志文件从主数据库传输到备用数据库时，加密的应用程序数据保持加密状态。然而，仅当备用站点为只读模式或故障切换之后，才需要在辅助站点上显示主数据库的主密钥，但应用重做日志除外。建议将主钱包复制到辅助站点，以便发生故障切换时所有数据快速可用。此外，还可以选择将 Oracle Wallet 转换为自动打开式钱包，从而当数据库联机时自动将主密钥用于备用数据库。

将日志文件传送到备用数据库时，加密数据仍在日志文件中以及传输过程中保持加密状态。

当重新设定主数据库的主密钥时，需要将钱包重新复制到所有辅助站点。如果钱包在辅助站点上处于打开状态，则需要将其关闭（将从内存中删除旧的主密钥），然后新钱包可打开，以便将新的主密钥加载到数据库内存（新的主密钥在其中模糊存储）。

对用于在备用站点上访问敏感数据的钱包关注的客户可在辅助站点上更改钱包口令。但是，这样可能会增加口令管理的复杂性，且如果忘记口令，还有可能延迟备用站点的使用。

逻辑备用数据库

自 Oracle Database 11gR1 起，Oracle Data Guard 逻辑备用数据库可与透明数据加密配合使用。需要在辅助站点上显示并打开主密钥，以便从加密的日志文件读取数据时通过 SQL Apply 对数据解密。此外，在将数据写入逻辑备用数据库时，还可以选择使用相同的主密钥对传入的数据加密。

Oracle Streams

自 Oracle Database 11g 起，Oracle Streams 可与 TDE 配合使用。加密数据由 Streams 引擎解密以便传输数据（字符集、数据库版本、平台等），且在传输到其他数据库的过程中不被加密，因此建议使用 Oracle Advanced Security 网络加密对流量加密。当无法到达接收端且需要暂时存储数据时，将最初加密的数据以加密方式存储在磁盘上。在 Oracle Database 11g 之前，Oracle Streams 将加密列视为“不受支持的数据类型”，因此跳过关联的表。对于接收端数据库，本地钱包不需要与源钱包和主密钥相同，因为敏感内容以明文形式到达。

Active Data Guard

在主数据库上重新设定主密钥的操作中，会生成一个带有新主密钥 ID 的重做标记，该新主密钥 ID 将传输到备用数据库。假若备用数据库也可以访问钱包副本，它将使用该主密钥 ID 执行相同的主密钥重新设定操作。

Oracle 透明数据加密和 Oracle GoldenGate

Oracle Databases 10.2.0.5 和 11.2.0.2/3 提供对 Oracle GoldenGate 11.1.1.1 的内置支持；要将其启用，只需执行

```
SYSTEM> @$ORACLE_HOME/rdbms/admin/prvtclkm.plb;
```

在 Oracle 11.1.0.7 中，应用补丁 9409423 后，执行相同的步骤以启用 Oracle GoldenGate 11.1.1.1，以便从 Oracle 数据库中提取加密数据。

Oracle GoldenGate 和具有（本地）自动打开式钱包的 TDE

当源数据库中的 TDE 配置有（本地）自动打开式钱包时，需要将补丁 10395645 应用于 10.2.0.5、11.1.0.7 和 11.2.0.2。

要在将明文数据从 Oracle 数据库传输到其目标位置的过程中施加保护，Oracle GoldenGate 支持通过 BlowFish 或 SSH 对网络流量加密。

ORACLE 透明数据加密和 ORACLE GOLDEN GATE

ORACLE GOLDEN GATE 版本	TDE 列加密	TDE 表空间加密
11.1.1.1 之前的版本	对所有 DB 版本提供部分支持，但前提是**表** • 具有主键或唯一索引，且**加密列** • 为 CHAR 和 VARCHAR2 数据类型，而且 • 不是主键或唯一索引	不支持
11.1.1.1	Oracle 10.2.0.5 和 11.2.0.2/3 中提供内置支持，在 11.1.0.7 中需要安装补丁 9409423	

Oracle 透明数据加密和 Oracle RMAN

通过一个简单命令，即可对 RMAN 磁盘备份加密**并**压缩：

```
RMAN> connect target <ORACLE_SID>/<SYS password>;
RMAN> set encryption on;
RMAN> backup [as compressed backupset] database;
```

ORACLE RMAN 加密和压缩通过 TDE 加密的数据

数据	备份时具有的特性…		
	压缩	加密	压缩和加密
未加密	压缩数据	加密数据	数据先压缩再加密

ORACLE RMAN 加密和压缩通过 TDE 加密的数据			
使用 TDE 列加密进行加密	压缩数据；将加密列视为未加密，从而降低加密列的压缩率	加密数据；对加密列双重加密。	数据先压缩再加密；将加密列视为未加密；对加密列双重加密
使用 TDE 表空间加密对数据加密	对加密表空间解密、压缩和重新加密；对明文表空间压缩和加密（压缩后，它们无法与加密数据区分开）。	加密块无需更改即传给备份；对明文块加密以便备份。	对加密块解密、压缩和重新加密；对明文块压缩和加密。

Oracle RMAN 可对磁盘备份进行压缩和加密。可以使用源数据库的主密钥，或者，如果要将数据还原到其他数据库，也可以从口令生成加密密钥，从而无需共享主加密密钥。它还允许使用“双”模式，在该模式中，可以通过提供正确的主密钥或提供正确的口令解密备份文件。

Real Application Clusters (RAC)

Oracle RAC 中的 Oracle Wallet 管理

当将 Oracle Advanced Security TDE 配置为使用 Oracle Wallet 存储主密钥时，钱包/主密钥与数据库之间存在 1:1 的关系；对于 Oracle Database **11gR1** 中的 Real Application Clusters (RAC) 配置来说也是如此。在第一个实例上启用 TDE 后，需要将钱包和本地 `sqlnet.ora` 文件复制到其他所有实例，然后手动打开，才可将主密钥加载到每个实例的内存中。同样，在一个实例上重新设定主加密密钥时，需要将钱包复制到该集群中的其他所有实例；关闭钱包以从内存中删除旧主密钥，然后重新将其打开以加载新主密钥。Oracle **不**支持在 RAC 实例之间共享同一个 Oracle Wallet，因为在一个实例上重新设定主密钥，但未适当更新其他实例时，钱包可能会损坏。

在 Oracle Database **11g 第 2 版**中，Oracle 建议将 Oracle Wallet 存储在一个集中的位置：使用“`asmca`”(ASM Configuration Assistant) 在 ASM 之上创建 ACFS 文件系统；在其中存储钱包；所有实例中 `sqlnet.ora` 的条目如下所示：

```
ENCRYPTION_WALLET_LOCATION=
 (SOURCE = (METHOD = FILE)
  (METHOD_DATA =
   (DIRECTORY  =  /opt/oracle/acfsmounts/data_tdevolume/$ORACLE_UNQNAME/)))
```

当实例启动时，该文件系统自动挂载。打开和关闭钱包以及设置或重新设定/轮换 TDE 主加密密钥在所有节点之间同步。

当设置环境变量“ORACLE_UNQNAME”时，多个支持 RAC 的数据库可运行相同的 Grid Infrastructure，从而在 ACFS 中存储各自的 TDE 主加密密钥。例如，有两个支持 RAC 的数据库，分别将“ORACLE_UNQNAME”设置为“RAC-HR”和“RAC-FIN”，各自的钱包将存储在 ACFS 上的单独子目录下：

/opt/oracle/acfsmounts/data_tdevolume/**RAC-HR**和
/opt/oracle/acfsmounts/data_tdevolume/**RAC-FIN**

在正常的操作系统环境变量中必须设置“ORACLE_UNQNAME”，以及对数据库使用“srvctl”：

```
$ srvctl setenv database -d RAC-HR -T “ORACLE_UNQNAME=RAC-HR”
$ srvctl setenv database -d RAC-FIN -T “ORACLE_UNQNAME=RAC-FIN”
```

如果 Oracle Wallet 无法存储在 ACFS 文件系统中，则需要将其复制到所有实例：首先在第一个实例上创建钱包和主密钥，然后将钱包复制到其他所有实例。即使钱包未存储在中心位置，不更新钱包内容的钱包操作（钱包打开/关闭命令）也会在 RAC 实例之间同步，而主密钥设置或重新设定操作仅与集中、共享的钱包同步。

在涉及 RAC 的所有情况下，都应该应用自动打开功能，以便 TDE 完全支持 Oracle Restart（自动启动运行在 ASM 之上的实例）。当多个实例共享一个集中存储的 Oracle Wallet 时，无法使用**本地**自动打开式钱包。

使用 ACFS 访问控制保护 Oracle Wallet

当 TDE Wallet 存储在 ACFS 文件系统中时，自 Linux 上的 Oracle Database 11.2.0.2 和 Windows 上的 11.2.0.3 起，可以使用 ACFS 安全特性实施额外的访问控制和职责分离。要使用 ACFS 安全特性，必须首先针对集群对该特性初始化，**然后**将在其中使用 ACFS 安全特性的每个 ACFS 文件系统必须为应用 ACFS 安全特性做好准备，如 Oracle® 自动存储管理管理员指南 11g 第 2 版中所述。建议使用“asmca”执行这些操作。也可以通过命令行界面执行这些步骤，如下所示：

该示例假设已在以下目录中创建 TDE 钱包
/u01/opt/oracle/acfsmounts/data_tdevolume/wallet_dir/ewallet.p12和
/u01/opt/oracle/acfsmounts/data_tdevolume/wallet_dir/cwallet.sso

要针对集群初始化 ACFS 安全特性，请作为“root”用户执行：

```
# /sbin/acfsutil sec init -u secadmin -g secadmingrp
```

其中，“secadmin”是指定为（首位）安全管理员的现有操作系统用户，“secadmingrp”是指定为安全管理员组的现有操作系统组。所有安全管理员都必须属于这个安全管理员组。该命令仅可由“root”用户运行，且会提示输入口令。这个是新安全管理员需要输入的口令。建议“secadmin”用户在初始化 ACFS 安全特性之后立即更改口令：

```
$ /sbin/acfsutil sec admin password
```

一旦 ACFS 安全特性的口令由“secadmin”用户更改，“root”用户则无法对其进行任何更改。一旦针对集群初始化安全特性后，将在其中使用安全特性的每个 ACFS 文件系统必须为应用该安全特性做好准备：
作为拥有“SYSASM”权限的用户，**登录到 ASM 实例**，然后在 Oracle Database 11.2.0.2 中执行：

```
SYSASM> alter diskgroup DATA set attribute 'compatibility.asm' = '11.2.0.2';
SYSASM> alter diskgroup DATA set attribute 'compatibility.advm' = '11.2.0.2';
```

… 在 Oracle Database 11.2.0.3 中执行：

```
SYSASM> alter diskgroup DATA set attribute 'compatibility.asm' = '11.2.0.3';
SYSASM> alter diskgroup DATA set attribute 'compatibility.advm' = '11.2.0.3';
```

```
$ /sbin/acfsutil sec prepare -m /u01/opt/oracle/acfsmounts/data_tdevolume
```

其中，“-m”指定挂载 ACFS 文件系统的挂载点。

要保护 TDE 钱包，首先，必须在 ACFS 文件系统上创建一个领域：

```
$ /sbin/acfsutil sec realm create TDEWalletRealm
  -m /u01/opt/oracle/acfsmounts/data_tdevolume
  -d "Realm to protect the TDE Wallet"
  -e off
```

该命令在 ACFS 文件系统
（在“/u01/opt/oracle/acfsmounts/data_tdevolume”上挂载）上创建了一个名为“TDEWalletRealm”的领域。“-e off”选项指定已针对该领域禁用加密。

为了只允许 Oracle 可执行文件访问 TDE 钱包，可以创建应用程序规则，如下所示：

```
$ /sbin/acfsutil sec rule create allowOracleDBRule
  -m /u01/opt/oracle/acfsmounts/data_tdevolume
  -t application $ORACLE_HOME/bin/oracle
  -o ALLOW
```

“-t”指定规则的类型。在该示例中，它是应用程序规则。还支持基于用户名、时间或主机名的其他规则类型。创建规则后，即可将其添加到规则集，如下所示：

创建新规则集：

```
$ /sbin/acfsutil sec ruleset create TDEWalletRuleSet
  -m /u01/opt/oracle/acfsmounts/data_tdevolume
```

将规则添加到规则集：

```
$ /sbin/acfsutil sec ruleset edit TDEWalletRuleSet
  -m /u01/opt/oracle/acfsmounts/data_tdevolume
  -a allowOracleDBRule -o ANY_TRUE
```

（可选）要允许“orapki”等 Java 应用程序和 Oracle Wallet Manager（“owm”）访问 Oracle TDE 钱包，请执行以下命令：

```
$ /sbin/acfsutil sec rule create allowJavaAppRule
  -m /u01/opt/oracle/acfsmounts/data_tdevolume
  -t application $ORACLE_HOME/jdk/bin/java -o ALLOW
```

将规则添加到规则集：

```
$ /sbin/acfsutil sec ruleset edit TDEWalletRuleSet
  -m /u01/opt/oracle/acfsmounts/data_tdevolume
  -a allowJavaAppRule -o ANY_TRUE
```

然后 TDE 钱包可受该领域的保护：

```
$ /sbin/acfsutil sec realm add TDEWalletRealm
  -m /u01/opt/oracle/acfsmounts/data_tdevolume
  -u oracle -l ALL:TDEWalletRuleSet -f
  -r /u01/opt/oracle/acfsmounts/data_tdevolume/wallet_dir
```

实施这些保护后，“oracle”操作系统用户以及“secadmin”和“root”用户对 TDE 钱包既没有读取权限也没有写入权限；只有 Oracle 数据库可以打开和关闭钱包，以及重新设定 TDE 主加密密钥。

有关 ACFS 安全特性支持的所有命令和选项的更多详细信息，请参见 Oracle® 自动存储管理管理员指南 11g 第 2 版 (11.2)。

Oracle 数据库机

Oracle 数据库机是一个基于 Oracle Database 11.2.0.2 的预配置、双节点 RAC 系统。默认创建 ACFS 文件系统，从而允许集中存储 Oracle Wallet，包括应用前面讨论的强大访问控制。此外，数据库机中的 Intel® CPU 还基于 AES-NI 为 **TDE 表空间加密**提供硬件加密加速。Oracle Database 11.2.0.2.4 和 11.2.0.3 中包含补丁 10296641，因此加密和解密都获益于 Intel AES-NI 提供的硬件加密加速。

Exadata 数据库云服务器

基于 Oracle Database 11gR2 (11.2.0.2) 的 Exadata 数据库云服务器 X2 配有 Intel® Xeon® CPU，这些 CPU 为 **TDE 表空间加密**提供基于硬件的加密加速。X2-2 和 X2-8 的存储节点是相同的；它们使用支持 AES-NI 的 Intel® Xeon® L5640 CPU。如果 Oracle 优化器决定将查询下推至智能存储单元，则在存储节点上解密数据；否则，在计算节点上解密数据。下表说明了该事实：

EXADATA 型号	X2-2		X2-8	
节点	**加密**	**解密**	**加密**	**解密**
计算	通过以下补丁启用 Intel® Xeon® X5670 中的硬件加速（快 6 倍）：10296641	默认启用 Intel® Xeon® X5670 中的硬件加速（快 8 倍）	通过以下补丁启用 Intel® Xeon® E7-8870 中的硬件加速（快 6 倍）：10296641	默认启用 Intel® Xeon® E7-8870 中的硬件加速（快 8 倍）
存储	n/a	默认启用 Intel® Xeon® L5640 中的硬件加速	n/a	默认启用 Intel® Xeon® L5640 中的硬件加速

在 2011 年 12 月 6 日之前订购的 Exadata X2-8 中，计算节点配有 Intel X7560，它基于 Nehalem 技术提供硬件加密加速，速度将提高约 2 倍。如果 Exadata 已升级到 11.2.0.3，则不需要补丁 10296641。

当使用 TDE **表空间**加密时，发送到 Exadata 的数据（可选）首先使用 Exadata 混合列压缩 (EHCC) 进行压缩，然后加密，最后写入磁盘。

当选择数据时，首先将其解密，然后应用“智能扫描”从结果集中删除不需要的数据，最后对结果集解压缩，并返回至数据库。

Exadata 混合列压缩与 TDE **表空间**加密结合使用将大大补偿 TDE 表空间加密带来的较小性能开销，因为是对*压缩*数据加密：当压缩数据集的大小比未压缩的数据集小 35% 时，使用 TDE **表空间**加密进行加密和解密的数据会减少 35%。

甲骨文（中国）软件系统有限公司

北京远洋光华中心办公室

地址：北京市朝阳区景华南街5号远洋光华中心C座21层
邮编：100020
电话：(86.10) 6535-6688
传真：(86.10) 6515-1015

北京汉威办公室

地址：北京市朝阳区光华路7号汉威大厦10层1003-1005单元
邮编：100004
电话：(86.10) 6535-6688
传真：(86.10) 6561-3235

北京甲骨文大厦

地址：北京市海淀区中关村软件园24号楼甲骨文大厦
邮编：100193
电话：(86.10) 6106-6000
传真：(86.10) 6106-5000

北京国际软件大厦办公室

地址：北京市海淀区中关村软件园9号楼国际软件大厦二区308单元
邮编：100193
电话：(86.10) 8279-8400
传真：(86.10) 8279-8686

北京孵化器办公室

地址：北京市海淀区中关村软件园孵化器2号楼A座一层
邮编：100193
电话：(86.10) 8278-6000
传真：(86.10) 8282-6401

上海名人商业大厦办公室

地址：上海市黄浦区天津路155号名人商业大厦12层
邮编：200001
电话：(86.21) 2302-3000
传真：(86.21) 6340-6055

上海腾飞浦汇大厦办公室

地址：上海市黄浦区福州路318号腾飞浦汇大厦508-509室
邮编：200001
电话：(86.21) 2302-3000
传真：(86.21) 6391-2366

上海创智天地10号楼办公室

地址：上海市杨浦区淞沪路290号创智天地10号楼512-516单元
邮编：200433
电话：(86.21) 6095-2500
传真：(86.21) 6107-5108

上海创智天地11号楼办公室

地址：上海市杨浦区淞沪路303号创智天地科教广场3期11号楼7楼
邮编：200433
电话：(86.21) 6072-6200
传真：(86.21) 6082-1960

上海新思大厦办公室

地址：上海市漕河泾开发区宜山路926号新思大厦11层
邮编：200233
电话：(86.21) 6057-9100
传真：(86.21) 6083-5350

广州国际金融广场办公室

地址：广州市天河区珠江新城华夏路8号合景国际金融广场18楼
邮编：510623
电话：(86.20) 8513-2000
传真：(86.20) 8513-2380

成都中海国际中心办公室

地址：成都市高新区交子大道177号中海国际中心7楼B座02-06单元
邮编：610041
电话：(86.28) 8530-8600
传真：(86.28) 8530-8699

深圳飞亚达科技大厦办公室

地址：深圳市南山区高新南一道飞亚达科技大厦16层
邮编：518057
电话：(86.755) 8396-5000
传真：(86.591) 8601-3837

深圳德赛科技大厦办公室

地址：深圳市南山区高新南一道德赛科技大厦8层0801-0803单元
邮编：518057
电话：(86.755) 8660-7100
传真：(86.755) 2167-1299

大连办公室

地址：大连软件园东路23号大连软件园15号楼502
邮编：116023
电话：(86.411) 8465-6000
传真：(86.755) 8465-6499

苏州办公室

地址：苏州工业园区星湖街328号苏州国际科技园5期11幢1001室
邮编：215123
电话：(86.512) 8666-5000
传真：(86.512) 8187-7838

沈阳办公室

地址：沈阳市和平区青年大街390号皇朝万鑫国际大厦A座39层3901&3911室
邮编：110003
电话：(86.24) 8393-8700
传真：(86.24) 2353-0585

济南办公室

地址：济南市泺源大街150号中信广场11层1113单元
邮编：250011
电话：(86.531) 6861-1900
传真：(86.531) 8518-1133

南京办公室

地址：南京市玄武区洪武北路55号置地广场19层1911室
邮编：210018
电话：(86.25) 8579-7500
传真：(86.25) 8476-5226

西安办公室

地址：西安市高新区科技二路72号西安软件园零壹广场主楼1401室
邮编：710075
电话：(86.29) 8834-3400
传真：(86.25) 8833-9829

重庆办公室

地址：重庆市渝中区邹容路68号大都会商厦1611室
邮编：400010
电话：(86.23) 6037-5600
传真：(86.23) 6370-8700

杭州办公室

地址：杭州市西湖区杭大路15号嘉华国际商务中心810&811室
邮编：310007
电话：(86.571) 8168-3600
传真：(86.571) 8717-5299

福州办公室

地址：福州市五四路158号环球广场1601室
邮编：350003
电话：(86.591) 8621-5050
传真：(86.591) 8801-0330

青岛办公室

地址：青岛市香港中路76号颐中皇冠假日酒店909室
邮编：266071
电话：(86.532) 8571-8888
传真：(86.591) 8571-6666

南昌办公室

地址：江西省南昌市西湖区沿江中大道258号
皇冠商务广场10楼1009室
邮编：330025
电话：(86.791) 8612-1000
传真：(86.791) 8657-7693

呼和浩特办公室

地址：内蒙古自治区呼和浩特市新城区迎宾北路7号
大唐金座19层北侧1902-1904室
邮编：010051
电话：(86.471) 3941-600
传真：(86.471) 5100-535

郑州办公室

地址：河南省郑州市中原区中原中路220号
裕达国际贸易中心A座2015室
邮编：450007
电话：(86.371) 6755-9500
传真：(86.371) 6797-2085

武汉办公室

地址：武汉市江岸区中山大道1628号
武汉天地企业中心5号大厦23层2301单元
邮编：430010
电话：(86.27) 8221-2168
传真：(86.27) 8221-2168

长沙办公室

地址：长沙市芙蓉区韶山北路159号通程国际大酒店1311-1313室
邮编：410011
电话：(86.731) 8977-4100
传真：(86.731) 8425-9601

石家庄办公室

地址：石家庄市中山东路303号石家庄世贸广场酒店14层1402室
邮编：050011
电话：(86.311) 6670-8080
传真：(86.311) 8667-0618

昆明办公室

地址：昆明市三市街六号柏联广场写字楼11层1103A室
邮编：650021
电话：(86.871) 6402-4600
传真：(86.871) 6361-4946

合肥办公室

地址：安徽省合肥市蜀山区政务新区怀宁路1639号平安大厦18层1801室
邮编：230022
电话：(86.551) 6595-8200
传真：(86.551) 6371-3182

南宁办公室

地址：广西省南宁市青秀区民族大道136-2号华润大厦B座2302室
邮编：530028
电话：(86.771) 391-8400
传真：(86.771) 577-5500

Oracle Advanced Security 透明数据加密最佳实践
2012 年 7 月
作者：Peter Wahl

公司网址：http://www.oracle.com（英文）
中文网址：http://www.oracle.com/cn（简体中文）
销售中心：800-810-0161
售后服务热线：800-810-0366
培训服务热线：800-810-9931

欢迎访问：
http://www.oracle.com（英文）
http://www.oracle.com/cn（简体中文）