**加密 ZFS 文件系统**

加密是对数据进行编码以实现保密性的过程，数据所有者需要使用密钥才能访问已编码的数据。使用 ZFS 加密的优点如下所述：

* ZFS 加密与 ZFS 命令集相集成。与其他 ZFS 操作一样，加密操作（例如密钥更改和重建密钥）是联机执行的。
* 您可以使用现有的存储池，只要对它们进行了升级。可以灵活地加密特定的文件系统。
* ZFS 加密可由后代文件系统继承。可以通过 ZFS 委托管理来委托密钥管理。
* 在 CCM 和 GCM 操作模式下，使用密钥长度为 128、192 和 256 的 AES（Advanced Encryption Standard，高级加密标准）对数据进行加密。
* ZFS 加密使用 Oracle Solaris 加密框架，该框架自动允许它访问加密算法的任何可用硬件加速或优化的软件实现。
* 当前，您无法对 ZFS 根文件系统或其他 OS 组件（例如 /var 目录）进行加密，即使它是单独的文件系统也是如此。

您可以在创建 ZFS 文件系统时设置加密策略，但是该策略不可更改。例如，在启用了加密属性的情况下创建了 tank/home/darren 文件系统。缺省的加密策略是提示您输入口令短语，口令短语的长度必须至少为 8 个字符。

# **zfs create -o encryption=on tank/home/darren**

Enter passphrase for 'tank/home/darren': **xxxxxxx**

Enter again: **xxxxxxxx**

确认文件系统已启用加密。例如：

# **zfs get encryption tank/home/darren**

NAME PROPERTY VALUE SOURCE

tank/home/darren encryption on local

当文件系统的加密值为 on 时，缺省的加密算法为 aes-128-ccm。

**包装密钥**用于加密实际的数据加密密钥。如上例所示，在创建加密的文件系统时，会将包装密钥从 zfs 命令传递到内核。包装密钥位于一个文件（采用 raw 或 hex 格式）中，或者派生自口令短语。

包装密钥的格式和位置是在 keysource 属性中指定的，如下所示：

keysource=format,location

* 格式为下列值之一：
  + raw－原始的密钥字节
  + hex－十六进制密钥字符串
  + passphrase－用以生成密钥的字符串
* 位置为下列值之一：
  + prompt－创建或挂载文件系统时提示您输入密钥或口令短语
  + file:///*filename*－密钥文件或口令短语文件在文件系统中的位置
  + pkcs11－描述 PKCS#11 令牌中密钥或口令短语位置的 URI
  + https://*location*－密钥文件或口令短语文件在安全服务器上的位置。不建议使用此方法以明文形式传输密钥信息。根据在 keysource 属性的格式部分中请求的内容，对 URL 执行 GET 可仅返回密钥值或口令短语。

对 keysource 使用 https:// 定位器时，服务器提供的证书必须是 libcurl 和 OpenSSL 信任的证书。将自己的信任锚点证书或自签名证书添加到 /etc/openssl/certs 下的证书库中。将 PEM 格式证书放在 /etc/certs/CA 目录下并运行以下命令：

# **svcadm refresh ca-certificates**

如果 keysource 格式为 **passphrase**，则包装密钥派生自口令短语。否则，keysource 属性值指向实际的包装密钥，为原始字节或十六进制格式。您可以指定口令短语存储在文件中，或者存储在提示输入的原始字节流中，这可能仅适合于编写脚本。

如果文件系统的 keysource 属性值标识 passphrase，则包装密钥派生自口令短语（使用 PKCS#5 PBKD2 和按文件系统随机生成的 salt）。这意味着，相同的口令短语在后代文件系统上使用时可生成不同的包装密钥。

文件系统的加密策略由后代文件系统继承，且不可删除。例如：

# **zfs snapshot tank/home/darren@now**

# **zfs clone tank/home/darren@now tank/home/darren-new**

Enter passphrase for 'tank/home/darren-new': **xxxxxxx**

Enter again: **xxxxxxxx**

# **zfs set encryption=off tank/home/darren-new**

cannot set property for 'tank/home/darren-new': 'encryption' is readonly

如果需要复制或迁移加密或未加密的 ZFS 文件系统，请注意以下要点：

* 当前，不能将以未加密形式发送的数据集流接收为加密的流，即使接收池的数据集已启用加密。
* 可以使用以下命令将未加密的数据迁移到已启用加密的池/文件系统：
  + cp -r
  + find | cpio
  + tar
  + rsync
* 可以将复制的加密文件系统流接收到加密文件系统中，且数据保持加密状态。有关更多信息，请参见[示例 5-4](http://docs.oracle.com/cd/E37934_01/html/E36658/gkkih.html" \l "gkkof)。

**更改加密 ZFS 文件系统的密钥**

通过使用 zfs key -c 命令，可以更改加密文件系统的包装密钥。必须首先已通过以下方式装入了现有的包装密钥：在引导时、通过显式装入文件系统密钥 (zfs key -l) 或者通过挂载文件系统 (zfs mount *filesystem*)。例如：

# **zfs key -c tank/home/darren**

Enter new passphrase for 'tank/home/darren': **xxxxxxxx**

Enter again: **xxxxxxxx**

在以下示例中，更改了包装密钥，并更改了 keysource 属性值以指定包装密钥来自一个文件。

# **zfs key -c -o keysource=raw,file:///media/stick/key** **tank/home/darren**

通过使用 zfs key -K 命令，可以更改加密文件系统的数据加密密钥，但是新的加密密钥仅用于新写入的数据。此功能可用来满足针对数据加密密钥的时限的 NIST 800-57 准则。例如：

# **zfs key -K tank/home/darren**

在上面的示例中，数据加密密钥不可见，也不是由您直接管理的。此外，您需要具有 keychange 委托才能执行密钥更改操作。

有以下加密算法可用：

* aes-128-ccm、aes-192-ccm、aes-256-ccm
* aes-128-gcm、aes-192-gcm、aes-256-gcm

ZFS keysource 属性标识用来包装文件系统数据加密密钥的密钥的格式和位置。例如：

# **zfs get keysource tank/home/darren**

NAME PROPERTY VALUE SOURCE

tank/home/darren keysource passphrase,prompt local

ZFS rekeydate 属性标识上次执行 zfs key -K 操作的日期。例如：

# **zfs get rekeydate tank/home/darren**

NAME PROPERTY VALUE SOURCE

tank/home/darren rekeydate Wed Jul 25 16:54 2012 local

如果加密文件系统的 creation 和 rekeydate 属性具有相同的值，则表明从未通过 zfs key -K 操作为该文件系统重建密钥。

**管理 ZFS 加密密钥**

您可以采用不同方式管理 ZFS 加密密钥，具体取决于您的需求，您既可以在本地系统上进行管理，也可以远程管理（如果需要一个中央位置）。

* **本地**－上述示例说明了包装密钥既可以是口令短语提示符，也可以是本地系统上文件中的存储的原始密钥。
* **远程**－可以使用密钥集中管理系统（如 Oracle 密钥管理程序）或者使用支持对 http 或 https URI 的简单 GET 请求的 Web 服务远程存储密钥信息。通过使用 PKCS#11 令牌，Oracle Solaris 系统可以访问 Oracle 密钥管理程序的密钥信息。

有关管理 ZFS 加密密钥的更多信息，请参见：

<http://www.oracle.com/technetwork/articles/servers-storage-admin/manage-zfs-encryption-1715034.html>

有关使用 Oracle 密钥管理程序管理密钥信息的信息，请参见：

<http://docs.oracle.com/cd/E24472_02/>

**委托 ZFS 密钥操作权限**

查看关于委托密钥操作的以下权限描述：

* 通过使用 zfs key -l 和 zfs key -u 命令装入或卸载文件系统密钥需要 key 权限。大多数情况下，还将需要 mount 权限。
* 通过使用 zfs key -c 和 zfs key -K 命令更改文件系统密钥需要 keychange 权限。

请考虑针对密钥使用（装入或卸载）和密钥更改授予单独的权限，这样您可以采用“两人”密钥操作模型。例如，确定哪些用户可以使用密钥，哪些用户可以更改这些密钥。或者，两种用户需要同时在场才能进行密钥更改。此模型还允许您构建密钥契约系统。

**挂载加密的 ZFS 文件系统**

在尝试挂载加密的 ZFS 文件系统时，请检查以下注意事项：

* 如果加密文件系统的密钥在引导时不可用，则不会自动挂载该文件系统。例如，加密策略设置为 passphrase,prompt 的文件系统在引导时不会挂载，因为引导过程不会中断以提示您输入口令短语。
* 如果要在引导时挂载加密策略设置为 passphrase,prompt 的文件系统，将需要使用 zfs mount 命令显式挂载它并指定口令短语，或者使用 zfs key -l 命令在引导系统之后提示您输入密钥。

例如：

# **zfs mount -a**

Enter passphrase for 'tank/home/darren': **xxxxxxxx**

Enter passphrase for 'tank/home/ws': **xxxxxxxx**

Enter passphrase for 'tank/home/mark': **xxxxxxxx**

* 如果加密文件系统的 keysource 属性指向另一个文件系统中的文件，则文件系统的挂载顺序可能影响在引导时是否挂载加密的文件系统，尤其是文件在可移除介质上时。

**升级加密的 ZFS 文件系统**

将 Solaris 11 系统升级到 Solaris 11.1 之前，请确保加密的文件系统已挂载。挂载加密的文件系统并提供口令短语（如果提示）。

# **zfs mount -a**

Enter passphrase for 'pond/amy': **xxxxxxxx**

Enter passphrase for 'pond/rory': **xxxxxxxx**

# **zfs mount | grep pond**

pond /pond

pond/amy /pond/amy

pond/rory /pond/rory

然后，升级加密的文件系统。

# **zfs upgrade -a**

如果尝试升级的加密 ZFS 文件系统未挂载，将显示类似于以下内容的消息：

# **zfs upgrade -a**

cannot set property for 'pond/amy': key not present

此外，zpool status 输出可能显示损坏的数据。

# **zpool status -v pond**

.

.

.

pond/amy:<0x1>

pond/rory:<0x1>

如果出现上述错误，请按上文所述重新挂载加密的文件系统。然后，清理并清除池错误。

# **zpool scrub pond**

# **zpool clear pond**

有关升级文件系统的更多信息，请参见[升级 ZFS 文件系统](http://docs.oracle.com/cd/E37934_01/html/E36658/gkkog.html" \l "scrolltoc)。

**ZFS 压缩、重复数据删除和加密属性之间的交互**

在使用 ZFS 压缩、重复数据删除和加密属性时，请检查以下注意事项：

* 在写入文件时，会对数据进行压缩和加密，并验证校验和。然后，会尽可能对数据进行重复删除。
* 在读取文件时，会验证校验和，并对数据进行解密。然后，根据需要对数据进行解压缩。
* 如果在克隆的加密文件系统上启用了 dedup 属性，且未对克隆使用 zfs key -K 或 zfs clone -K 命令，则会尽可能对来自所有克隆的数据进行重复数据删除。

**加密 ZFS 文件系统的示例**

**示例 5-1 通过使用原始密钥加密 ZFS 文件系统**

在以下示例中，通过使用 pktool 命令生成了一个 aes-256-ccm 加密密钥，并将其写入了文件 /cindykey.file 中。

# **pktool genkey keystore=file outkey=/cindykey.file keytype=aes keylen=256**

然后，在创建 tank/home/cindy 文件系统时指定了 /cindykey.file。

# **zfs create -o encryption=aes-256-ccm -o keysource=raw,file:///cindykey.file**

**tank/home/cindy**

**示例 5-2 使用不同的加密算法加密 ZFS 文件系统**

您可以创建一个 ZFS 存储池，并让该存储池中的所有文件系统都继承加密算法。在此示例中，创建了 users 池，创建了 users/home 文件系统，并通过使用口令短语对该文件系统进行了加密。缺省的加密算法是 aes-128-ccm。

然后，创建了 users/home/mark 文件系统，并通过使用 aes-256-ccm 加密算法对其进行了加密。

# **zpool create -O encryption=on users mirror c0t1d0 c1t1d0 mirror c2t1d0 c3t1d0**

Enter passphrase for 'users': **xxxxxxxx**

Enter again: **xxxxxxxx**

# **zfs create users/home**

# **zfs get encryption users/home**

NAME PROPERTY VALUE SOURCE

users/home encryption on inherited from users

# **zfs create -o encryption=aes-256-ccm users/home/mark**

# **zfs get encryption users/home/mark**

NAME PROPERTY VALUE SOURCE

users/home/mark encryption aes-256-ccm local

**示例 5-3 克隆加密的 ZFS 文件系统**

如果克隆文件系统从与其源快照相同的文件系统继承了 keysource 属性，则新的 keysource 是不必要的，并且如果 keysource=passphra e,prompt，系统也不会提示您输入新的口令短语。将为克隆使用相同的 keysource。例如：

缺省情况下，在克隆加密文件系统的后代时，系统不会提示您输入密钥。

# **zfs create -o encryption=on tank/ws**

Enter passphrase for 'tank/ws': **xxxxxxxx**

Enter again: **xxxxxxxx**

# **zfs create tank/ws/fs1**

# **zfs snapshot tank/ws/fs1@snap1**

# **zfs clone tank/ws/fs1@snap1 tank/ws/fs1clone**

如果要为克隆文件系统创建新密钥，请使用 zfs clone -K 命令。

如果是克隆加密的文件系统而不是加密的后代文件系统，则系统将提示您提供新密钥。例如：

# **zfs create -o encryption=on tank/ws**

Enter passphrase for 'tank/ws': **xxxxxxxx**

Enter again: **xxxxxxxx**

# **zfs snapshot tank/ws@1**

# **zfs clone tank/ws@1 tank/ws1clone**

Enter passphrase for 'tank/ws1clone': **xxxxxxxx**

Enter again: **xxxxxxxx**

**示例 5-4 发送和接收加密的 ZFS 文件系统**

在下面的示例中，根据加密的 /tank/home/darren 文件系统创建了 tank/home/darren@snap1 快照。然后，将快照发送到 bpool/snaps，由于已启用了加密属性，因此会加密生成的接收数据。但是，在发送过程中，tank/home/darren@snap1 流未加密。

# **zfs get encryption tank/home/darren**

NAME PROPERTY VALUE SOURCE

tank/home/darren encryption on local

# **zfs snapshot tank/home/darren@snap1**

# **zfs get encryption bpool/snaps**

NAME PROPERTY VALUE SOURCE

bpool/snaps encryption on inherited from bpool

# **zfs send tank/home/darren@snap1 | zfs receive bpool/snaps/darren1012**

# **zfs get encryption bpool/snaps/darren1012**

NAME PROPERTY VALUE SOURCE

bpool/snaps/darren1012 encryption on inherited from bpool

在这种情况下，将为接收的加密文件系统自动生成新密钥。