ORACLE
ZFS STORAGE
APPLIANCE

Best Practices for Deploying Encryption and Managing Its Keys on Oracle ZFS Storage Appliance

ORACLE WHITE PAPER | SEPTEMBER 2018

ORACLE

Introduction

When considering data encryption, it is important to realize that data can be either at rest or in transit. Each of these two states of data requires a different encryption technology. Data at rest is stored encrypted on disk or tape. For data in transit, encryption is handled by the transport layers used to transmit data between source and destination. This white paper focuses on encryption for data at rest on Oracle ZFS Storage Appliance.

Using encryption on data at rest provides protection of personal and business confidential information against unauthorized access due to data carriers falling into the wrong hands. It also avoids the costs of safe destruction of data carriers when they are swapped out due to hardware failure or hardware upgrades. By destroying the encryption keys used to encrypt the data, the data is considered to be safely erased.

This paper describes how to deploy the encryption functionality of Oracle ZFS Storage Appliance, its related key management, and best practices for implementation of these functions.

Why Employing Encryption Is Critical

Encryption is an important element in safeguarding a variety of data, including personal or confidential business data, against unauthorized access. Various government agencies are now requiring the use of encryption on data storage devices in their IT environments.

Data confidentiality security breaches can have serious effects for an organization, such as:

- Damaged reputation
- Financial impact due to regulatory fines and loss of business
- Risk of legal lawsuits or civil actions

Encrypting data at rest is a relatively simple and effective solution to mitigate these risks.

Key Concepts of Encryption

Encryption of data is one of numerous cryptographic methods, and is used to protect data against unauthorized access. There are several cryptographic technologies that use different algorithms. The different technologies are used for different types of cryptographic services. All technologies have the same purpose—protecting information by encoding it into an unreadable format and using a method of storing and/or transmitting data in a form that only its intended user or service can read and process. The following key cryptography security services can be distinguished:

Confidentiality — This cryptographic service provides confidentiality functionality by using encrypting technology. Most often data is encrypted using symmetric keys.

Authentication — This service provides identity verification of an entity that is requesting access to a resource by providing a cryptographic authentication item. Identities are verified using digital certificates.

Integrity — Integrity services are used to verify that data has not been modified or tampered with. Cryptographic hashing algorithms are used to create a message fingerprint to determine the originality of a piece of data.

Nonrepudiation — This service proves the origin of a message. When a message is sent, it is tagged with a digital fingerprint identifying the sender.

A proper business security plan encompasses all four security services.

Data encryption services are applicable to two types of data: data in flight and data at rest. Both need to be addressed to provide end-to-end data security.

Encryption of data in transit is often combined with authentication services to determine the identities of the sender and receiver and their permission to exchange data with each other.

Encryption of data at rest provides protection against unauthorized access of data when storage carriers fall into the wrong hands. To protect data from unauthorized access in an operating storage subsystem, you still need proper authentication systems, like LDAP or Active Directory set up within the computing infrastructure.

Data storage encryption is performed on data at rest; that is, on disk or tape. Access to the data is controlled by means of encryption keys. Storage subsystems use the encryption keys to access data (decrypt) for certain shares that have been used to encrypt the data on that share.

Data storage encryption can be done in several ways:

- Using encryption embedded in disk or tape hardware.
  This option requires costly storage devices, makes key management complex, and limits the granularity with which encryption can be applied.

- Using an encryption appliance in between disk storage subsystem and applications.
  This option is worth considering when encryption needs to be implemented in an existing infrastructure. Points to take into consideration are a possible data performance bottleneck and the addition of an extra device and vendor into the key management mix.
- Using encryption software on server or desktop.
  This option is commonly used by mobile devices or servers to encrypt locally stored data or backup software to safeguard against unauthorized access of data due to tape loss or theft. Separate software must be installed on the server or client to protect against the risk of unauthorized data access when exchanging disk drives or lost or stolen hardware. Encryption software is needed on each server or desktop, making it costly and difficult to manage.
- Embedded encryption in storage subsystems.
  Encryption of data and key management is handled by the storage subsystem. No extra hardware or software components are needed. Encryption is done at the storage subsystem controller level. While it might have some performance impact with certain sequential type workloads, the lower cost, flexibility, and easier key management outweigh the performance implications.
- Embedded encryption in the database layer.
  Using encryption in the database layer enables the use of encryption on specific database columns or the entire application database and a tight integration with related database management tools and user authentication functions.

Modern storage subsystems like Oracle ZFS Storage Appliance have enough CPU power to handle the encryption and key management functionality. This means you can avoid the use of expensive dedicated encryption devices or extra encryption appliances.

For external key management, the Oracle Key Manager solution offers central key management for all data storage encryption components. A clustered Oracle Key Manager provides protection against encryption key loss and loss of access to data due to an Oracle Key Manager server failure.

Key management rights are defined by key management policy roles. Certain roles can be assigned to users that match their operational responsibilities.

Oracle Transparent Data Encryption

Oracle Database uses authentication, authorization, and auditing mechanisms to secure data in the database, but not in the operating system data files where data is stored. To protect these data files, Oracle Database provides Transparent Data Encryption (TDE). TDE encrypts sensitive data stored in data files. To prevent unauthorized decryption, TDE stores the encryption keys in a security module external to the database.

Database users and applications do not need to manage key storage or create auxiliary tables, views, and triggers. An application that processes sensitive data can use TDE to provide strong data encryption with little or no change to the application.

Use TDE to protect confidential data stored in table columns. You can also use TDE to encrypt entire tablespaces.

Transparent Data Encryption stops attackers from bypassing the database and reading sensitive information from storage by enforcing data-at-rest encryption in the database layer. Applications and users authenticated to the database continue to have access to application data transparently (no application code or configuration changes are required), while attacks from OS users attempting to read sensitive data from table space files and attacks from thieves attempting to read information from acquired disks or backups are denied access to the clear text data. Transparent Data Encryption integrates directly with frequently used Oracle Database tools and technologies including Oracle Advanced Compression, Oracle Automatic Storage Management, a feature of Oracle Database, Oracle Recovery Manager, a feature of Oracle Database; (Oracle RMAN), Data Pump, a feature of Oracle Database, Oracle GoldenGate and more.

ZFS Encryption

Oracle ZFS Storage Appliance uses Oracle's ZFS file system features to provide its data storage encryption functionality. It uses a strong Advanced Encryption Standard (AES) 128,192, 256 bit or a two-tier security key architecture in which the ZFS encryption keys are further wrapped in a second layer of 256-bit encryption for wrapping keys. A single key can be used for the whole system, or unique keys can be used for individual projects and shares. Oracle ZFS Storage Appliance software version OS 8.8 introduces the ability to use encryption keys at the pool level, too.

When encryption is enabled, on either the entire pool, project or share level, all data on the respective source and its related metadata, like access control lists (ACLs) and quota information, are stored encrypted on the disks that are allocated to that resource. ZFS Storage Appliances can contain both, encrypted and non-encrypted pools, projects and shares. The encryption service is completely transparent to other file system services (like compression and deduplication) and protocols (like CIFS and NFS).

The encryption and key management of an Oracle ZFS Storage Appliance is controlled through the ZFS properties structure. This means that the normal ZFS properties inheritance rules apply here, too. The encryption will be hierarchical from pool level down to share/LUN level. That means, if the pool level encryption is active, all data in projects and shares/LUN´s belonging to that pool will be encrypted, and it is no longer possible to exclude data at the project or share/LUN level from the encryption.

A data encryption key is used to perform the encryption and decryption of the share data. The encryption key is generated by ZFS during the creation of the share and is stored in an encrypted state within the share. A wrapping key is used to encrypt and decrypt the data encryption key. It is this wrapping key that is used to gain access to the share, when ZFS mounts the share. Once the data key is decrypted, ZFS keeps this key in its cache until the share is unmounted or when the wrapping key is deleted.

When using data encryption on Oracle ZFS Storage Appliance, the key administrator has no need to manage the ZFS keys used to encrypt/decrypt the data stored on the Oracle ZFS Storage Appliance. The key administrator manages the wrapping keys, which are stored in the used keystore.

Both, a local keystore on Oracle ZFS Storage Appliance, and a keystore managed by Oracle Key Manager are supported by the Oracle ZFS Storage Appliance.

For an active encrypted pool or share/LUN, you can change its wrapping key value. The data encryption key is simply re-encrypted and the new encrypted value of the data encryption key will be stored within the ZFS metadata of the pool or share/LUN on disk. The value of the data encryption key itself can never be modified.

When a wrapping key is deleted, the related share or pool using that wrapping key is automatically taken offline. The pool or share is automatically brought back online if the wrapping key is restored from a backup (in case the wrapping key was accidentally deleted). If a wrapping key is permanently deleted, access to data on its related pool or share/LUN is lost and is considered securely erased.

In a clustered Oracle ZFS Storage Appliance configuration, the keystore is automatically synchronized between the two nodes. When a wrapping key is added, removed, or updated in the keystore, the entire keystore is updated on both nodes of the cluster.

Note: The wrapping key is frequently referred to as the *encryption key* in the Oracle ZFS Storage Appliance documentation and BUI/CLI.

Managing Keys in Oracle ZFS Storage Appliance

The wrapping keys used to enable access to encrypted pools or shares are all kept in the Oracle ZFS Storage Appliance keystore. Consequently, key management is about managing keys in the keystore. When an Oracle Key Manager keystore is used, the keys in the Oracle Key Manager keygroup used by Oracle ZFS Storage Appliance need to be administrated, too. Key management policies and administrative roles associated with these policies should be part of a wider organizational key management policy (KMP) and a related key management practices statement (KMPS). Such documents should include authorization and protection objectives and constraints that apply to the generation, distribution, accounting, storage, use, and destruction of cryptographic material.

The U.S. Department of Commerce's National Institute of Standards and Technology (NIST) has published a document, “Recommendation for Key Management,” Draft NIST Special Publication 800-57), which provides detailed guidance on the use of cryptographic key management and requirements for KMP and KMPS documents.

Defining Key Management Roles

In order to address the NIST requirements to separate audit, administration, and security functions, roles are created that can be assigned to different users.

Defining Role Authorizations for Local Key Management

For administrating the local and Oracle Key Manager keystore on Oracle ZFS Storage Appliance, the following role and role authorizations are defined:

ROLE ID	DESCRIPTION	
Keystore	Contain all the authorization options related to managing a keystore	
	Type of authorizations	Description
	listKeystore	List keys present in a keystore
	modifyKeystore	Permit keystore modifications
	readKeystore	Permit read access to sensitive values in a keystore

Managing Keys with Oracle Key Manager

Because Oracle Key Manager does centrally manage keys for multiple clients, more roles are present on the Oracle Key Manager node. Take these roles into account when you set up an agent for Oracle ZFS Storage Appliance on Oracle Key Manager. The following shows various roles and their functions. Examples follow in the next section.

ROLE ID	DESCRIPTION
Auditor	Views information about the KMA cluster
Backup Operator	Performs backups
Compliance Officer	Manages key policies and key groups
Operator	Manages agents, data units, and keys
Quorum Member	Views and adds credentials to operations pending quorum approval
Security Officer	Manages security settings, users, and sites, and transfers partners

Setting Up Encryption on Oracle ZFS Storage Appliance

Consider the following configuration requirements and related options for setting up encryption on Oracle ZFS Storage Appliance.

Setting Up Keystores

In order to use encrypted shares on Oracle ZFS Storage Appliance, a keystore must be configured. A keystore holds the wrapping keys used to access the data encryption keys used by ZFS. Each share on Oracle ZFS Storage Appliance holds its own encryption key. This encryption key is stored encrypted in the share's metadata. The wrapping key is used to encrypt and decrypt this key.

There are two type of keystores: a local keystore and an Oracle Key Manager keystore. The local keystore is used to manage and store wrapping keys on the appliance itself. When using the Oracle Key Manager keystore option, the wrapping keys are generated and managed on Oracle Key Manager.

The benefit of using a local keystore is that no extra hardware is needed to use the encryption functionality. Key management functions like secure backups of wrapping keys or recycling of wrapping keys need to be carefully managed for each Oracle ZFS Storage Appliance on which the encryption functionality is used.

The benefit of using Oracle Key Manager keystore is that it enables centralized managed wrapping keys and automated policies for key aging. Using a clustered Oracle Key Manager configuration protects against loss of wrapping keys due to loss of a node on which wrapping keys are stored. Remember that once wrapping keys are lost, data on shares or pools that were using those wrapping keys is lost, too, and the shares are considered securely erased.

Before setting up keystores and encrypted pools or shares, you should consider the requirements on key management and the backup strategy of encrypted shares.

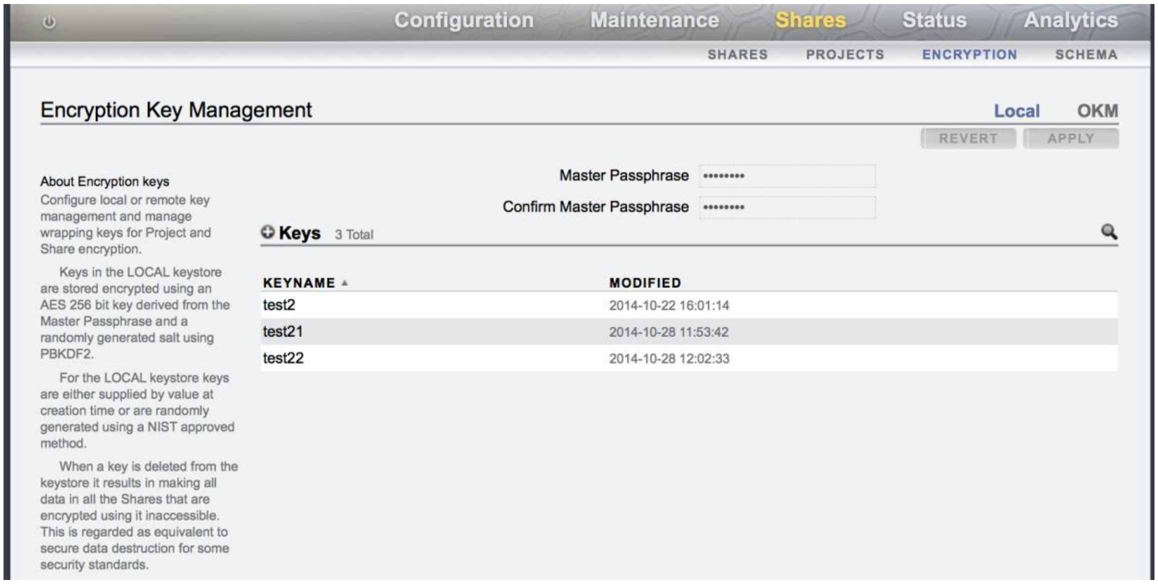
What business policies are used to store and manage encryption keys? Does the policy allow for keys to be stored and managed within the storage subsystem or should they be managed centrally and/or from a different location? The answers to these questions determine if a local or a remote keystore can be used.

A second factor to determine the use of local or remote keystores is the requirement for encrypted backups. When using data encryption for tape backups, Oracle's StorageTek T10000 tape cartridge should be used. Depending on customer key management policies, the tape backups could use the same Oracle Key Manager key lifecycle policies as the data on Oracle ZFS Storage Appliance.

The Best Practices section provides more details on these topics.

Setting up a local keystore

Wrapping keys are stored encrypted in the local keystore using an AES 256 bit key derived from the master passphrase. The master passphrase must be specified only once, when creating the keystore as shown in the following BUI screenshot.



Encryption Key Management

Master Passphrase ••••••••

Confirm Master Passphrase ••••••••

About Encryption keys

Configure local or remote key management and manage wrapping keys for Project and Share encryption.

Keys in the LOCAL keystore are stored encrypted using an AES 256 bit key derived from the Master Passphrase and a randomly generated salt using PBKDF2.

For the LOCAL keystore keys are either supplied by value at creation time or are randomly generated using a NIST approved method.

When a key is deleted from the keystore it results in making all data in all the Shares that are encrypted using it inaccessible. This is regarded as equivalent to secure data destruction for some security standards.

Keys 3 Total

KEYNAME ▲	MODIFIED
test2	2014-10-22 16:01:14
test21	2014-10-28 11:53:42
test22	2014-10-28 12:02:33

Figure 1. Configuring a local keystore in the Oracle ZFS Storage Appliance BUI

Once the keystore has been created, keys can be added to the keystore either by generating a random key or by providing a value during creation time.

Setting Up an Oracle Key Manager Keystore

Oracle Key Manager (referred to as OKM in the Oracle ZFS storage BUI) can be used to centrally provision and manage the distribution of keys and setup of retention policies for keys.

Before registering Oracle ZFS Storage Appliance as an agent within Oracle Key Manager, you must verify that Oracle Key Manager is configured properly and you have access to an Oracle Key Manager GUI. Using the Oracle Key Manager GUI, the key management appliance (KMA) can be monitored and configured. The KMA needs two network connections: one connected to the administrative network and the other to the so-called service network. It is a best practice to use a different subnet for each, as shown in the following diagram.

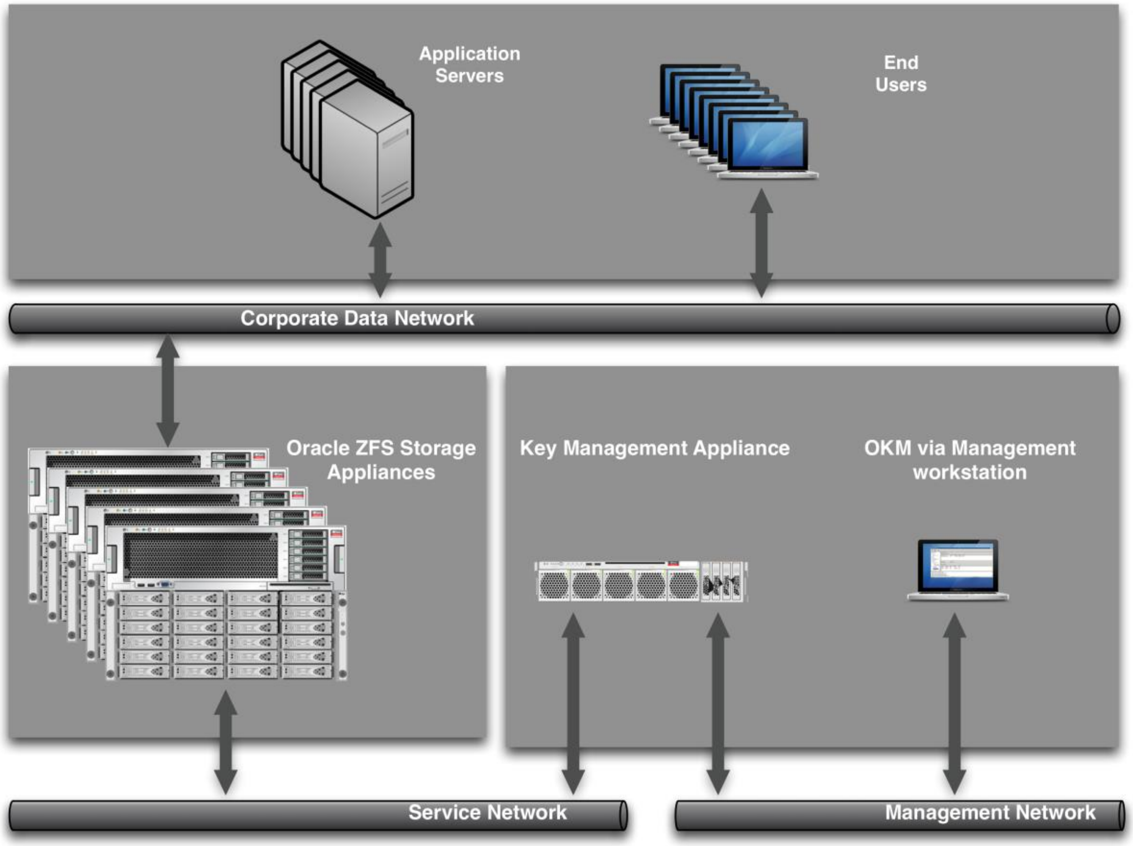


Figure 2. Schematic for properly configured key management system components

The GUI makes a connection to the KMA through the administrative network. The service network is used for key exchange traffic between the KMA and its registered agents—in this case, Oracle ZFS Storage Appliance.

The following configuration properties must be set with these steps:

1. Establish users with roles that allows them to define key policies, set up a key group, set up a site location, and register an agent.
2. Set up a site location to be used when registering Oracle ZFS Storage Appliance with Oracle Key Manager (optional). Requires an Oracle Key Manager security officer's role credentials.
3. Set up key policies to be used when creating a key group. Requires a compliance officer's role credentials.

4. Set up a key group, to be used when registering Oracle ZFS Storage Appliance with Oracle Key Manager. Requires an Oracle Key Manager compliance officer’s role credentials.
5. When using a single key management appliance (KMA) node, back up the Oracle Key Manager keys in order to enable the keys to be released for use by the KMA. After creating a backup, the KMA moves the keys from a *generated state* to the ready state.
   This is an important step as no key requests can be made on Oracle ZFS Storage Appliance as long as the keys on the KMA have not been transitioned to the *ready state*. Requires an Oracle Key Manager backup operator’s role credentials.
6. Register Oracle ZFS Storage Appliance. Requires an Oracle Key Manager operator’s role credentials.
7. Request wrapping keys from Oracle Key Manager on Oracle ZFS Storage Appliance. Requires either a user root credentials or a user with a role that contains *permit keystore modification* authorization on Oracle ZFS Storage Appliance.

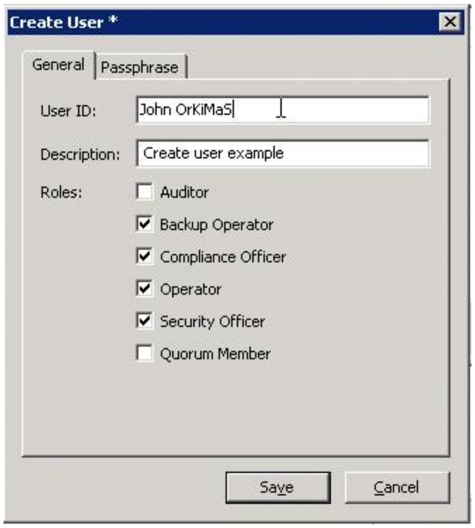
Performing Initial Oracle Key Manager Configuration Setup Steps

In the following examples, a single user is created that has multiple roles assigned to it to enable that user to perform all the mentioned configuration steps. Your organization might not allow a single user to have multiple authority roles.

Connect the Oracle ZFS Storage Appliance that is going to be registered with Oracle Key Manager to the service network subnet.

If Oracle Key Manager has already been set up and users, key policies, and key groups are already configured, you can skip the first few steps and start at step 6, registering Oracle ZFS Storage Appliance as Oracle Key Manager agent.

1. Create an admin user on Oracle Key Manager.
   Verify that there is no user already present on the KMA with the required roles. If not present, create a new admin user with the roles enabled as shown in the following screenshot.



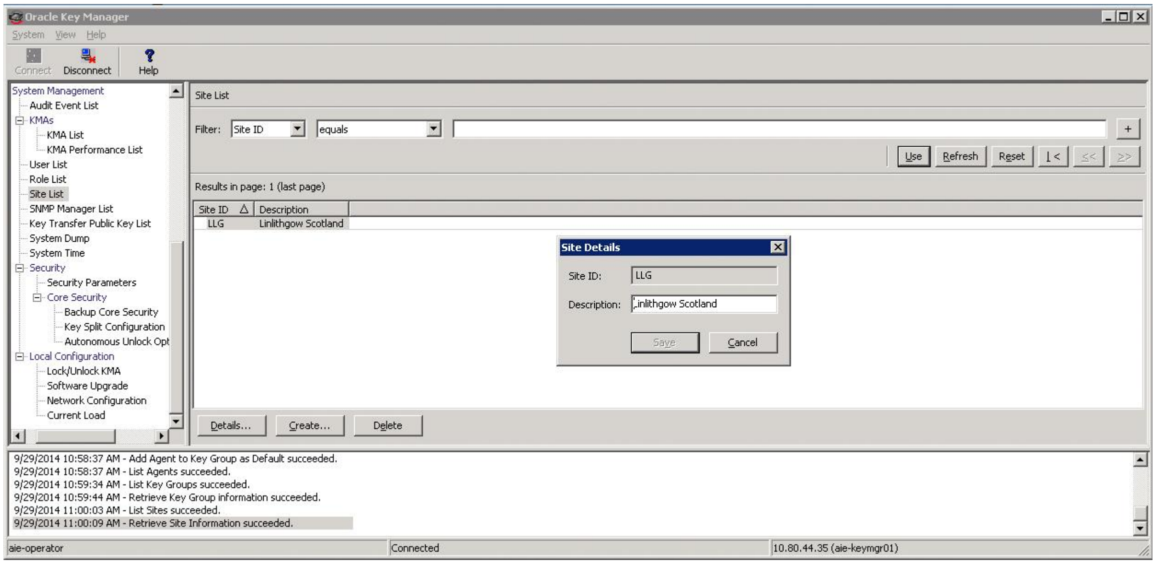
User ID: John OrKiMaS

Description: Create user example

Roles:
□ Auditor
☑ Backup Operator
☑ Compliance Officer
☑ Operator
☑ Security Officer
□ Quorum Member

Figure 3. Creating a user with the required roles in Oracle Key Manager

2. Create a site location (optional).
   When Oracle ZFS Storage Appliance is located on a different site or building, it is good practice to use site locations.



Filter: Site ID equals

Results in page: 1 (last page)

Site ID	Description
LLG	Linlithgow Scotland

Site ID: LLG

Description: Linlithgow Scotland

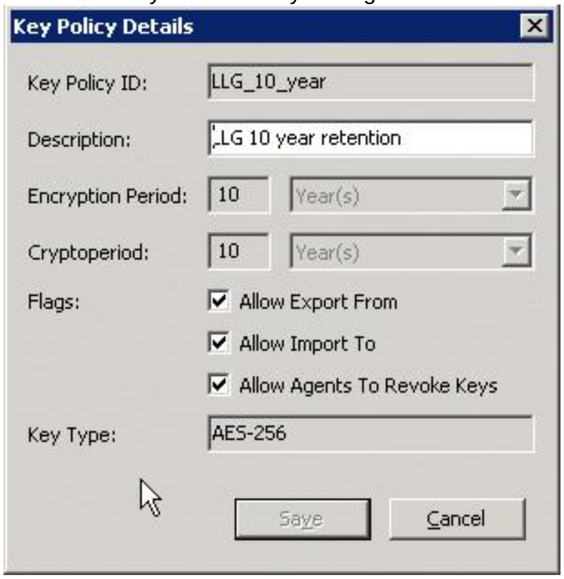
```
9/29/2014 10:58:37 AM - Add Agent to Key Group as Default succeeded.
9/29/2014 10:58:37 AM - List Agents succeeded.
9/29/2014 10:59:34 AM - List Key Groups succeeded.
9/29/2014 10:59:44 AM - Retrieve Key Group information succeeded.
9/29/2014 11:00:03 AM - List Sites succeeded.
9/29/2014 11:00:09 AM - Retrieve Site Information succeeded.
```

aie-operator | Connected | 10.80.44.35 (aie-keymgr01)

Figure 4. Providing site details for site locations in Oracle Key Manager

3. Create a key policy.
   Set up a key policy, select a retention policy that fits your requirements, and, if required, enable the *Allow Import, Allow Export* options.
   When setting the option *Allow Agents To Revoke Keys*, a deletion of a wrapping key on Oracle ZFS Storage Appliance results in the related key also being revoked in Oracle Key Manager. When the *Allow Agents to Revoke Keys* is not set, a key in Oracle Key Manager is no longer used by Oracle ZFS Storage Appliance when the related wrapping key is deleted from the Oracle ZFS Storage Appliance keystore. The related key will still exist in Oracle Key Manager until the key lifecycle completes or until the security officer

revokes the key in Oracle Key Manager.



Key Policy Details

Key Policy ID: LLG_10_year

Description: LLG 10 year retention

Encryption Period: 10 Year(s)

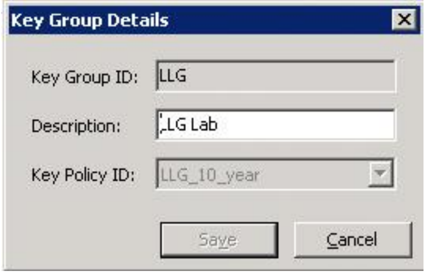
Cryptoperiod: 10 Year(s)

Flags:
☑ Allow Export From
☑ Allow Import To
☑ Allow Agents To Revoke Keys

Key Type: AES-256

Figure 5. Providing key policy details in Oracle Key Manager

4. Create a key group.
   A key group is a pool of keys to be used by the agent. Oracle ZFS Storage Appliance always requests keys from the Oracle Key Manager key group that is assigned as the default key group for the agent.



Key Group Details

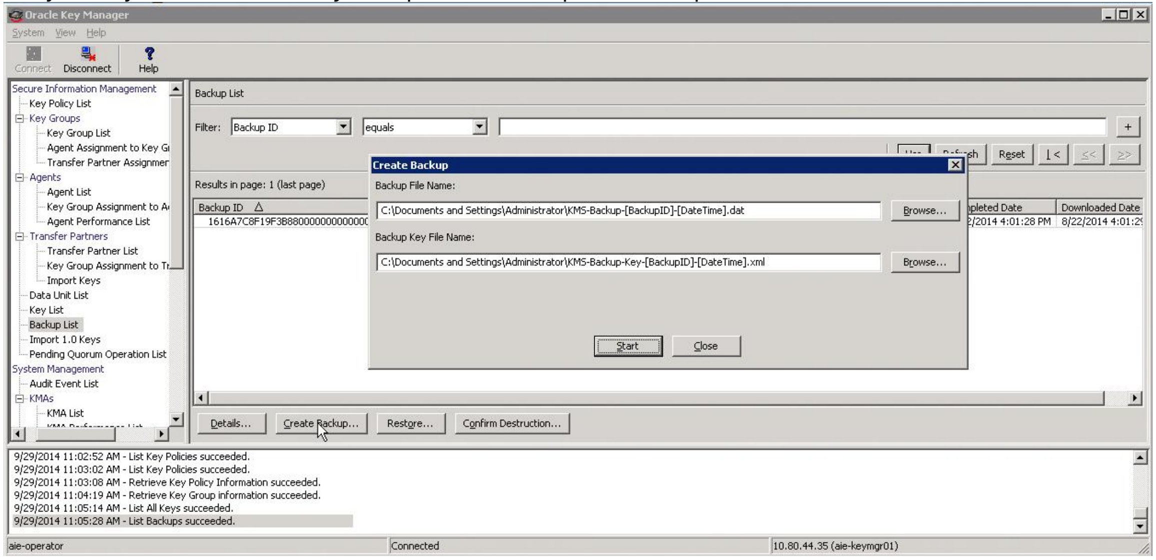
Key Group ID: LLG

Description: LLG Lab

Key Policy ID: LLG_10_year

Figure 6. Providing key group details in Oracle Key Manager

5. Create a backup of keys.
When a single node KMA is used (not a KMA cluster configuration), a backup of the Oracle Key Manager keys must be made before the agent is allowed to request keys from the default key group assigned to the agent.
Once a backup of the keys is made the key state will change from *Generated* to the *Ready* state. You can verify the key status under the *Key List* option from the operation tree pane in the GUI.



Filter: Backup ID equals

Results in page: 1 (last page)

1616A7C8F19F3B88000000000000000 | 2/2014 4:01:28 PM | 8/22/2014 4:01:2

Backup File Name: C:\Documents and Settings\Administrator\KMS-Backup-[BackupID]-[DateTime].dat

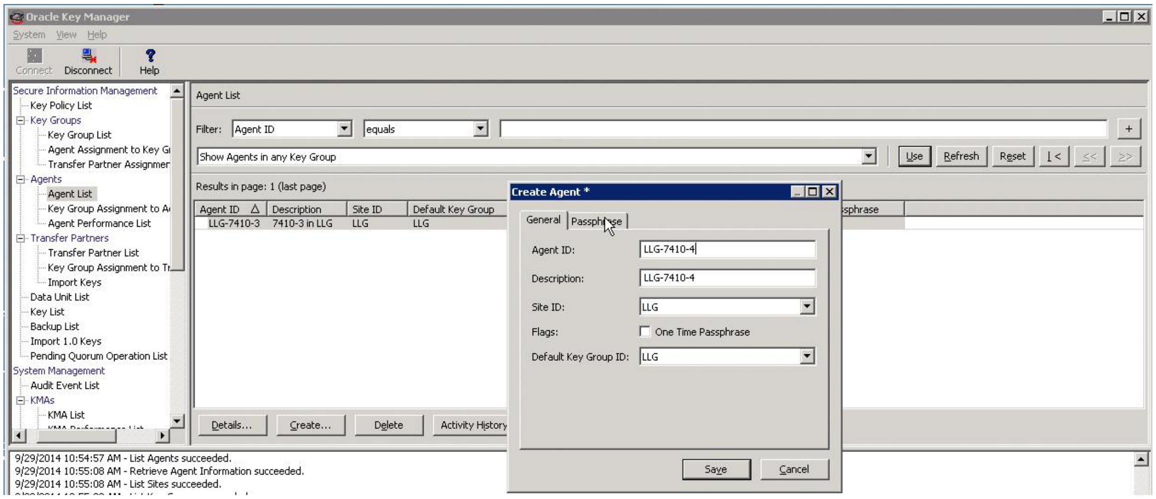
Backup Key File Name: C:\Documents and Settings\Administrator\KMS-Backup-Key-[BackupID]-[DateTime].xml

```
9/29/2014 11:02:52 AM - List Key Policies succeeded.
9/29/2014 11:03:02 AM - List Key Policies succeeded.
9/29/2014 11:03:08 AM - Retrieve Key Policy Information succeeded.
9/29/2014 11:04:19 AM - Retrieve Key Group information succeeded.
9/29/2014 11:05:14 AM - List All Keys succeeded.
9/29/2014 11:05:28 AM - List Backups succeeded.
```

aie-operator | Connected | 10.80.44.35 (aie-keymgr01)

Figure 7. Creating a backup of the keys

6. Registering Oracle ZFS Storage Appliance as agent in Oracle Key Manager.



Filter: Agent ID equals

Show Agents in any Key Group

Results in page: 1 (last page)

LLG-7410-3 | 7410-3 in LLG | LLG | LLG

Agent ID: LLG-7410-4

Description: LLG-7410-4

Site ID: LLG

Flags: ☐ One Time Passphrase

Default Key Group ID: LLG

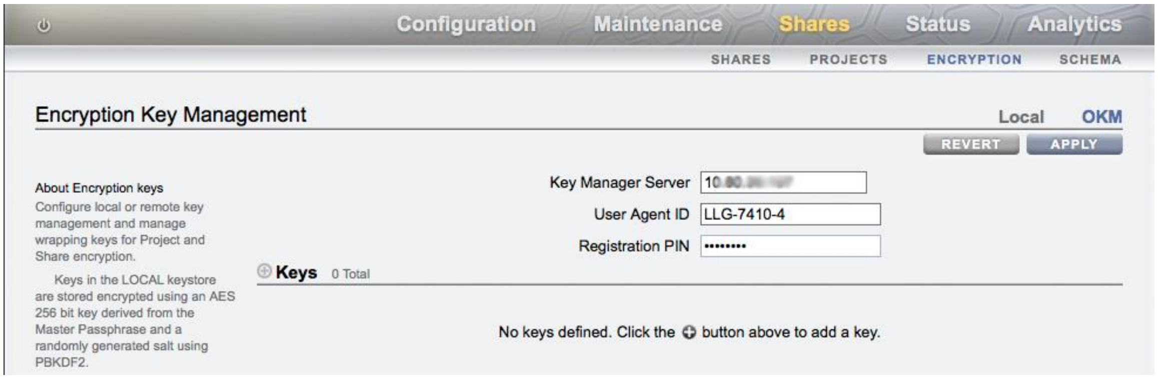
```
9/29/2014 10:54:57 AM - List Agents succeeded.
9/29/2014 10:55:08 AM - Retrieve Agent Information succeeded.
9/29/2014 10:55:08 AM - List Sites succeeded.
```

Figure 8. Setting a passphrase for a new agent in Oracle Key Manager

Before Oracle ZFS Storage Appliance can be registered with Oracle Key Manager, an agent must be created on Oracle Key Manager. Select the key group to be used by Oracle ZFS Storage Appliance and define a passphrase. This passphrase is used when issuing the registration request from Oracle ZFS Storage Appliance to Oracle Key Manager.

Important: The *One Time Passphrase* option (seen in the figure as a checkbox next to *Flags*) must not be used, because in an Oracle ZFS Storage Appliance cluster configuration, two registration requests are made to Oracle Key Manager, one for each appliance node. Also, using this option makes it difficult to reregister an appliance node, which must occur with an appliance hardware swap or a reconfiguration following a reset to factory defaults procedure. Only one agent needs to be created on the KMA when registering a clustered Oracle ZFS Storage Appliance.

After the agent is created on Oracle Key Manager, a registration request has to be issued from Oracle ZFS Storage Appliance, as illustrated in the following screenshot.



Encryption Key Management

About Encryption keys

Configure local or remote key management and manage wrapping keys for Project and Share encryption.

Keys in the LOCAL keystore are stored encrypted using an AES 256 bit key derived from the Master Passphrase and a randomly generated salt using PBKDF2.

Key Manager Server: 10 [redacted]

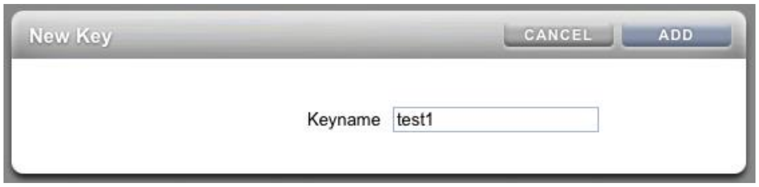
User Agent ID: LLG-7410-4

Registration PIN: ••••••••

No keys defined. Click the ⊕ button above to add a key.

Figure 8. Initiating a key registration request in the Oracle ZFS Storage Appliance BUI

7. Request one or more wrapping keys.
   Now that the communication between Oracle ZFS Storage Appliance and Oracle Key Manager is established, wrapping keys can be requested from Oracle Key Manager.



Keyname: test1

Figure 9. Requesting a new wrapping key in Oracle Key Manager

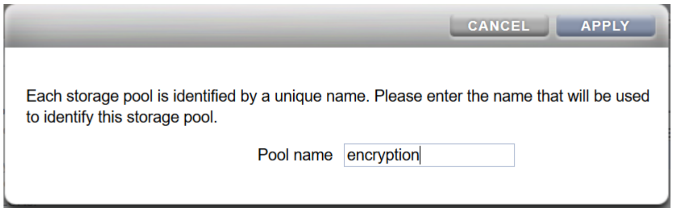
Creating Encrypted Pools

At this point, the Oracle ZFS Storage Appliance is ready to use encryption for pools, file systems and LUNs.

When creating a storage pool, there is an option to enable encryption for the whole pool. Once enabled, all projects and shares defined in that pool will be encrypted. Only encrypted projects or shares will be created on a pool with encryption enabled. A different pool that has encryption disabled is required to host non-encrypted projects and shares if needed.

A storage pool with encryption enabled is configured in the same way as any other pool except the additional options regarding encryption. This is shown by the following steps:

Start the configuration by selecting a name for the new pool.

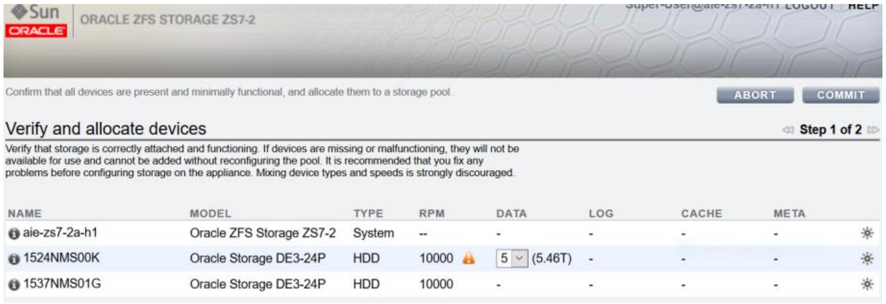


Each storage pool is identified by a unique name. Please enter the name that will be used to identify this storage pool.

Pool name: encryption

Figure 10 Configuring a new pool

Select the devices to be included.



Confirm that all devices are present and minimally functional, and allocate them to a storage pool.

Verify and allocate devices

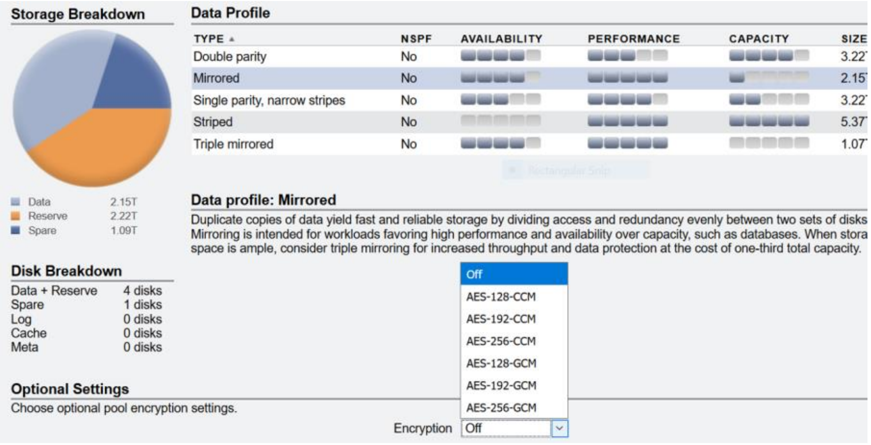
Step 1 of 2

Verify that storage is correctly attached and functioning. If devices are missing or malfunctioning, they will not be available for use and cannot be added without reconfiguring the pool. It is recommended that you fix any problems before configuring storage on the appliance. Mixing device types and speeds is strongly discouraged.

NAME	MODEL	TYPE	RPM	DATA	LOG	CACHE	META
aie-zs7-2a-h1	Oracle ZFS Storage ZS7-2	System	--	-	-	-	-
1524NMS00K	Oracle Storage DE3-24P	HDD	10000 ⚠	5 (5.46T)	-	-	-
1537NMS01G	Oracle Storage DE3-24P	HDD	10000	-	-	-	-

Figure 11 Verify and Allocate Devices

Select the profile of the new pool and the encryption key algorithm option.



Storage Breakdown

■	Data	2.15T
■	Reserve	2.22T
■	Spare	1.09T

Data Profile

TYPE ▲	NSPF	AVAILABILITY	PERFORMANCE	CAPACITY	SIZE
Double parity	No	■■■■□	■■■□□	■■■■□	3.22
Mirrored	No	■■■■□	■■■■■	■□□□□	2.15
Single parity, narrow stripes	No	■■■□□	■■■■□	■■□□□	3.22
Striped	No	□□□□□	■■■■■	■■■■■	5.37
Triple mirrored	No	■■■■□	■■■■■	□□□□□	1.07

Data profile: Mirrored

Duplicate copies of data yield fast and reliable storage by dividing access and redundancy evenly between two sets of disks Mirroring is intended for workloads favoring high performance and availability over capacity, such as databases. When stora space is ample, consider triple mirroring for increased throughput and data protection at the cost of one-third total capacity.

Disk Breakdown

Data + Reserve	4 disks
Spare	1 disks
Log	0 disks
Cache	0 disks
Meta	0 disks

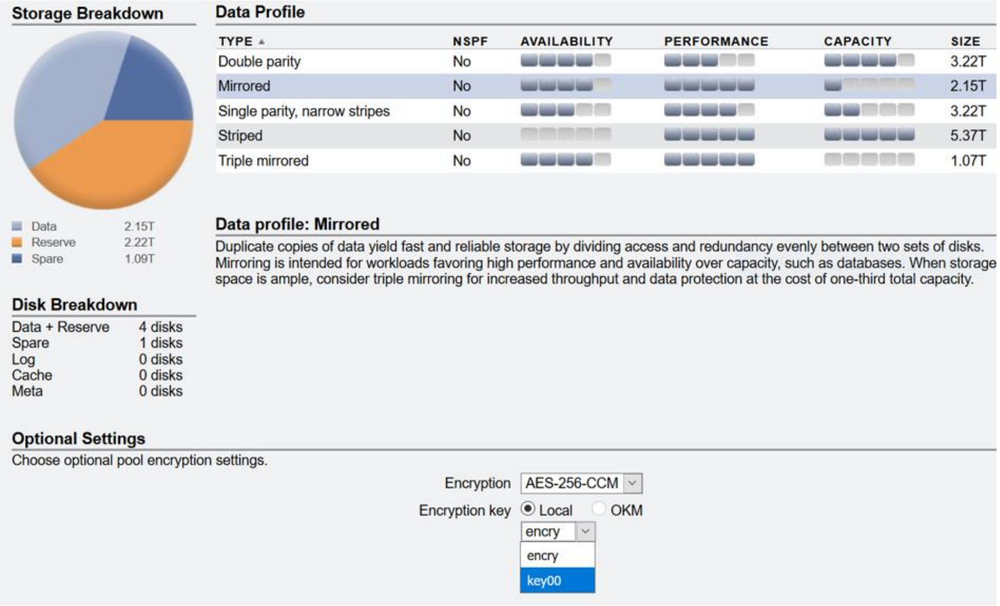
Optional Settings

Choose optional pool encryption settings.

Encryption: Off

Figure 12 Selection of Encryption Algorithm

Select the key to be used as the default encryption key for the projects and shares that will be defined on the new pool. Similar to project and share encryption, the key can either be locally stored or managed by Oracle Key Manager.



Storage Breakdown

Data	2.15T
Reserve	2.22T
Spare	1.09T

Disk Breakdown

Data + Reserve	4 disks
Spare	1 disks
Log	0 disks
Cache	0 disks
Meta	0 disks

Data Profile

TYPE ▲	NSPF	AVAILABILITY	PERFORMANCE	CAPACITY	SIZE
Double parity	No	■■■■□	■■■□□	■■■■□	3.22T
Mirrored	No	■■■■□	■■■■■	■□□□□	2.15T
Single parity, narrow stripes	No	■■■□□	■■■■□	■■□□□	3.22T
Striped	No	□□□□□	■■■■■	■■■■■	5.37T
Triple mirrored	No	■■■■□	■■■■■	□□□□□	1.07T

Data profile: Mirrored

Duplicate copies of data yield fast and reliable storage by dividing access and redundancy evenly between two sets of disks. Mirroring is intended for workloads favoring high performance and availability over capacity, such as databases. When storage space is ample, consider triple mirroring for increased throughput and data protection at the cost of one-third total capacity.

Optional Settings

Choose optional pool encryption settings.

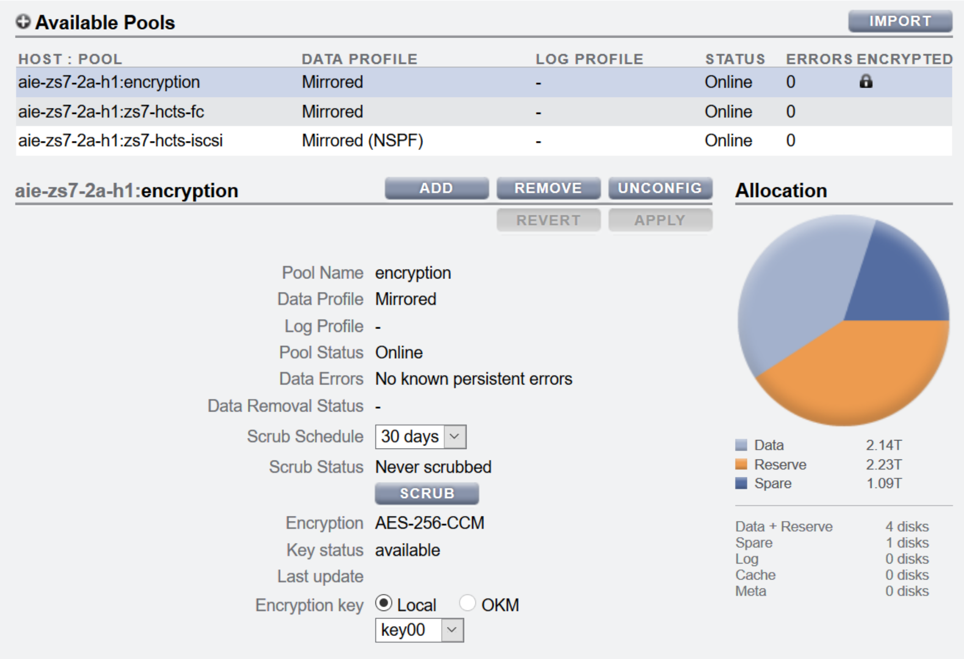
Encryption: AES-256-CCM

Encryption key: ● Local ○ OKM

encry

Figure 13 Selection of encryption key

The encrypted pool in the “Available Pools” list will be identified with a lock symbol below the “Encrypted” column.



Available Pools

HOST : POOL	DATA PROFILE	LOG PROFILE	STATUS	ERRORS	ENCRYPTED
aie-zs7-2a-h1:encryption	Mirrored	-	Online	0	🔒
aie-zs7-2a-h1:zs7-hcts-fc	Mirrored	-	Online	0	
aie-zs7-2a-h1:zs7-hcts-iscsi	Mirrored (NSPF)	-	Online	0	

aie-zs7-2a-h1:**encryption**

Pool Name	encryption
Data Profile	Mirrored
Log Profile	-
Pool Status	Online
Data Errors	No known persistent errors
Data Removal Status	-
Scrub Schedule	30 days
Scrub Status	Never scrubbed
Encryption	AES-256-CCM
Key status	available
Last update	
Encryption key	● Local ○ OKM
	key00

Allocation

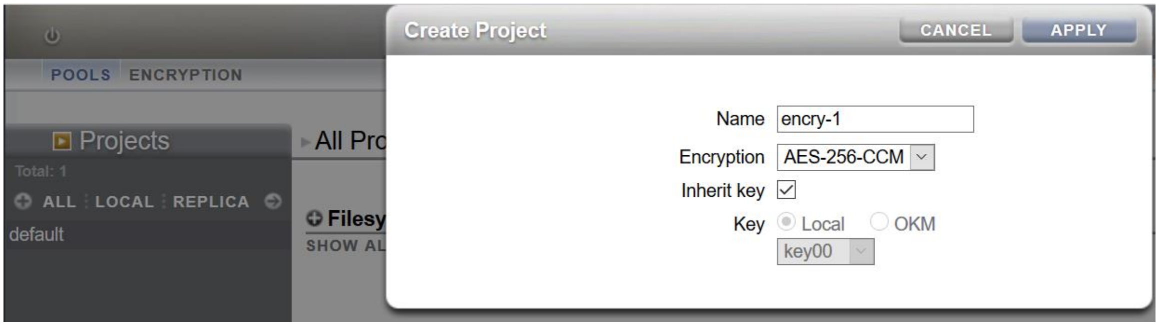
Data	2.14T
Reserve	2.23T
Spare	1.09T

Data + Reserve	4 disks
Spare	1 disks
Log	0 disks
Cache	0 disks
Meta	0 disks

Figure 14 Overview of available pools

Creating a Project on a pool with Encryption enabled

When creating a project on a pool that has encryption enabled, encryption cannot be disabled for any individual projects defined in that pool. Leave the “Inherit Key” box checked to use the default encryption key defined by the pool. Uncheck the “Inherit Key” box to use an alternate key (if available).



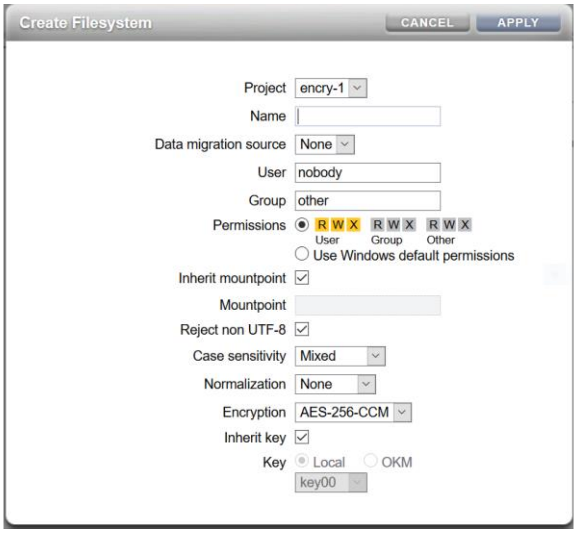
Create Project

Name	encry-1
Encryption	AES-256-CCM
Inherit key	☑
Key	○ Local ○ OKM
	key00

Figure 15 Creating a Project on an encrypted Pool

Creating Encrypted Shares or a LUN on a pool with Encryption enabled

When creating a share or LUN within a project on a pool that has encryption enabled, encryption cannot be disabled. Leave the “Inherit Key” box checked to use the default encryption key defined by the pool. Uncheck the “Inherit Key” box to use an alternate key (if available).



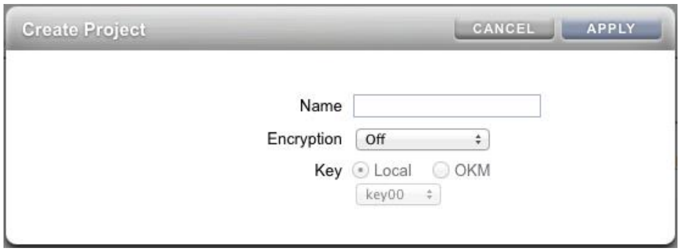
Create Filesystem

Project	encry-1
Name	
Data migration source	None
User	nobody
Group	other
Permissions	● R W X User R W X Group R W X Other ○ Use Windows default permissions
Inherit mountpoint	☑
Mountpoint	
Reject non UTF-8	☑
Case sensitivity	Mixed
Normalization	None
Encryption	AES-256-CCM
Inherit key	☑
Key	● Local ○ OKM key00

Figure 16 Creating a share on an encrypted pool

Creating Encrypted Projects on a pool without Encryption enabled

When creating a project on a pool that does not have encryption enabled, you have the option to enable encryption for all the shares in the project. Once encryption is selected, it cannot be switched off for individual shares in that project. The level of encryption can be selected per share when creating a share in an “encrypted” project.

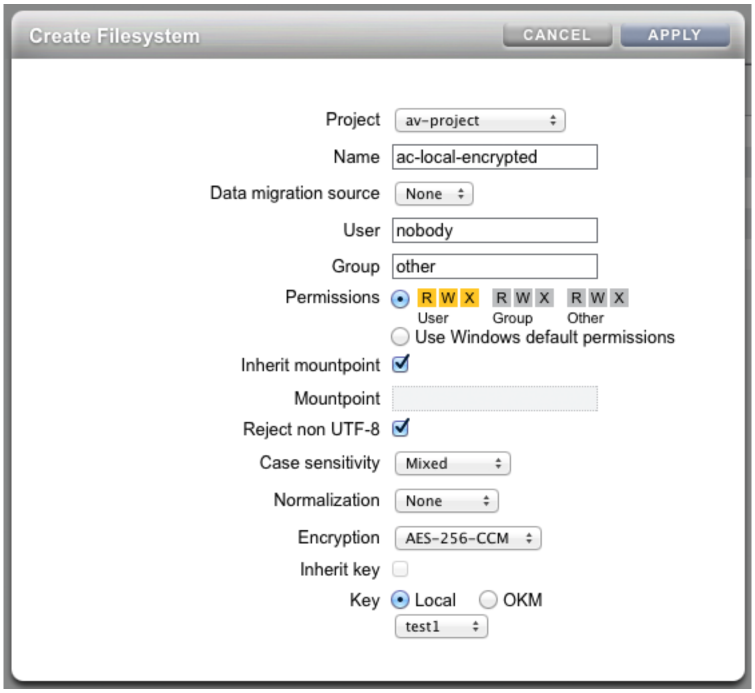


Create Project

Name	
Encryption	Off
Key	● Local ○ OKM key00

Figure 17. Encryption toggle in the Create Project dialog window of Oracle ZFS Storage Appliance

Encrypted shares can be created in existing projects or in newly created encrypted projects. When creating a share, select the level of encryption used for the data encryption in the file system. Select either the local or Oracle Key Manager keystore and the wrapping key to be used.

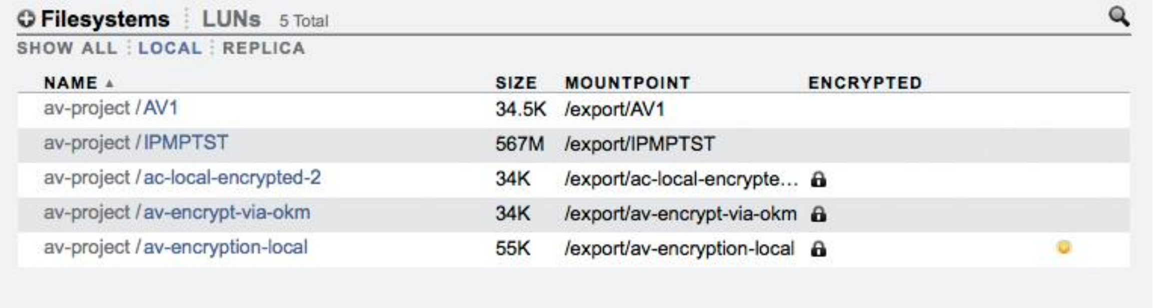


Project	av-project
Name	ac-local-encrypted
Data migration source	None
User	nobody
Group	other
Permissions	● R W X (User) R W X (Group) R W X (Other) ○ Use Windows default permissions
Inherit mountpoint	☑
Mountpoint	
Reject non UTF-8	☑
Case sensitivity	Mixed
Normalization	None
Encryption	AES-256-CCM
Inherit key	□
Key	● Local ○ OKM test1

Figure 18. Creating a share with level of encryption, keystore, and wrapping key designated

The *Inherit key* option is available if the project selected for the share has encryption set up in it. By selecting the inherit option, the wrapping key selected in the project is inherited by the new share. Note that this cannot be changed later.

In the Oracle ZFS Storage Appliance BUI, the shares using encryption are marked with a lock symbol. Shares that are unavailable are marked with a yellow LED status. One reason a share may be unavailable is that the wrapping key for the share is deleted.



NAME ▲	SIZE	MOUNTPOINT	ENCRYPTED	
av-project / AV1	34.5K	/export/AV1		
av-project / IPMPTST	567M	/export/IPMPTST		
av-project / ac-local-encrypted-2	34K	/export/ac-local-encrypte…	🔒	
av-project / av-encrypt-via-okm	34K	/export/av-encrypt-via-okm	🔒	
av-project / av-encryption-local	55K	/export/av-encryption-local	🔒	●

Figure 13. Displaying shares with status icons for encryption and unavailability

Shares created with the *Inherit key* option cannot be moved outside the project that owns the original key.

Protecting a Keystore with Backups

It is important to realize that when wrapping keys in a keystore are lost, all data on pools or shares that use wrapping keys from that keystore are now inaccessible—in other words, that data is considered erased. So, it is important to protect the loss of wrapping keys in a keystore due to hardware failures or other types of system failures. When using Oracle Key Manager cluster configurations, the keystore is automatically replicated between the KMA nodes in the Oracle Key Manager cluster, thus providing protection against loss of wrapping keys or loss of access to the keystore. In all other cases, the administrator needs to back up the keystore on a regular basis.

Creating a backup of a local keystore

A backup of the keystore on Oracle ZFS Storage Appliance can be created by creating a backup of the Oracle ZFS Storage Appliance configuration. This backup contains all the local encryption configuration of Oracle ZFS Storage Appliance and the contents of the local keystore. The backup can be used to restore all the configuration information of Oracle ZFS Storage Appliance. It is important to realize that the backup is made on an appliance itself, so to protect against a scenario in which the Oracle ZFS Storage Appliance controller hardware must be replaced, the backup file must be saved somewhere where it is considered to be secure.

Use the following commands in the CLI to back up a value of an individual key:

```
shares encryption local keys
select keyname=1
get key
```

Creating a backup of an Oracle Key Manager keystore

When using a keystore from an Oracle Key Manager configuration, you must account for the following:

- The configuration information on Oracle ZFS Storage Appliance used to register an appliance with the Oracle Key Manager agent setup.
- The Oracle Key Manager configuration information used to receive Oracle ZFS Storage Appliance and the keystore on Oracle Key Manager.

To protect the loss of configuration information on Oracle ZFS Storage Appliance, a backup of its configuration must be made and stored in a secure place.

When Oracle Key Manager is used in a cluster configuration, all Oracle Key Manager information is stored on every KMA node in the Oracle Key Manager cluster, giving protection against loss of (access to) a KMA node. However, when using a single KMA node configuration, a backup of the KMA keystore must be made and stored in a safe place.

Creating Snapshots and Clones of Encrypted Shares or Shares on Encrypted Pools

Snapshots of shares always inherit the wrapping key of the original share. When creating a clone of a snapshot, this dependency remains intact. Depending on your company’s security requirements, you might want to consider changing the wrapping key on a clone.

Replicating encrypted shares

Encrypted shares on Oracle ZFS Storage Appliance can be replicated in the same way as any type of share. Data is retrieved from the share unencrypted, sent over to the target Oracle ZFS Storage Appliance, and stored there encrypted.

Important: The name of the wrapping key is stored in the metadata of the share at the source side. At the start of the initial replication, the replication process on the target Oracle ZFS Storage Appliance is expecting to see a wrapping key with the same name in the same keystore (LOCAL or OKM) at the target Oracle ZFS Storage Appliance. The wrapping key does not need to have the same value. So before starting the initial replication of a number of shares, it is important that the names of the wrapping keys used by those shares have been created in the keystore on the target machine.

When using Oracle Key Manager in a replication environment, both the Oracle Key Manager agent serving the Oracle Storage ZFS Appliance replication source node and the Oracle Key Manager agent serving the Oracle ZFS Storage Appliance replication target node need to use the same Oracle Key Manager keygroup as the default group.

Best Practices When Using Encryption

Consider the following recommendations, best practices, and principles when planning the application of encryption with Oracle ZFS Storage Appliance.

Data Security Architecture

Encryption must be part of a broader data security architecture. This architecture must outline data security classifications, identify data regulations that must be adhered to, and detail implementation of those requirements. Identify authentication roles and groups.

“The Twenty Critical Security Controls for Effective Cyber Defense,” (commonly called the Consensus Audit Guidelines or CAG) is a publication of best practice guidelines for computer security. The group’s recommendations focus almost exclusively on technology solutions, and therefore are not a substitute for the much more comprehensive NIST guidelines. The CAG is a starting point for organizations that are reviewing or implementing a new security plan.

For additional protection against unauthorized access of data, user authentication services (LDAP, Active Directory), ACLs, and data network zoning/segmentation (VLANs) technologies need to be deployed.

Network Architecture Design

It is recommended to reserve a network interface on Oracle ZFS Storage Appliance for administrative access and connect this to a separate IP subnet that is only used for administrative tasks. This avoids the risk of losing administrative access to an appliance in case of network infrastructure problems in the production or corporate data network. See more details in the white paper, "Networking Best Practices with Oracle ZFS Storage Appliance" listed in Appendix A: References.

Oracle Key Manager Architecture Design

When using an Oracle Key Manager configuration to manage and serve wrapping keys to Oracle ZFS Storage Appliance, a third IP subnet is needed in order to separate the key management traffic from the administrative and data IP networks. A KMA node contains two network interfaces and requires different subnets to be configured for each of them as shown in the diagram in the previous section, Setting Up an Oracle Key Manager Keystore.

Creating a Backup of Wrapping Keys

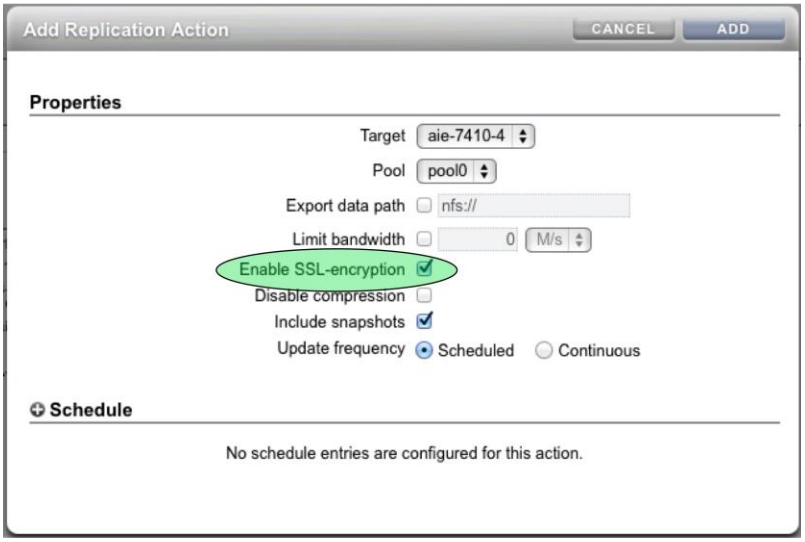
A wrapping key is the single entity that literally holds the access key to the share related to that wrapping key. Loss of the wrapping key results in loss of the share’s data.

Wrapping keys can become unavailable for the following reasons:

- Keys are deleted from the keystore; this could have been done by accidentally selecting a wrong key.
- A rollback is performed on Oracle ZFS Storage Appliance to a release that does not support encryption.
- A rollback is performed on Oracle ZFS Storage Appliance to a release where a keystore was not yet created.
- A rollback is performed on Oracle ZFS Storage Appliance to a release where no backup of an appliance configuration was created, no off-appliance backup of an appliance configuration file is present, and no (up-to-date) keystore is present in the current booted image, resulting in the loss of the keystore contents.
- Access to the KMA is lost.

Using Share Replication

When shares are replicated, data is read from the source like any application would, meaning it is unencrypted, decompressed and deduplicated. If replicating from a share where encryption is enabled, it is highly recommended that you use the SSL/TLS security option for the replication connection between the source and target Oracle ZFS Storage Appliance nodes when setting up a replication link to maintain data security.



Add Replication Action

Properties

Target: aie-7410-4

Pool: pool0

Export data path ☐ nfs://

Limit bandwidth ☐ 0 M/s

Enable SSL-encryption ☑

Disable compression ☐

Include snapshots ☑

Update frequency ● Scheduled ○ Continuous

Schedule

No schedule entries are configured for this action.

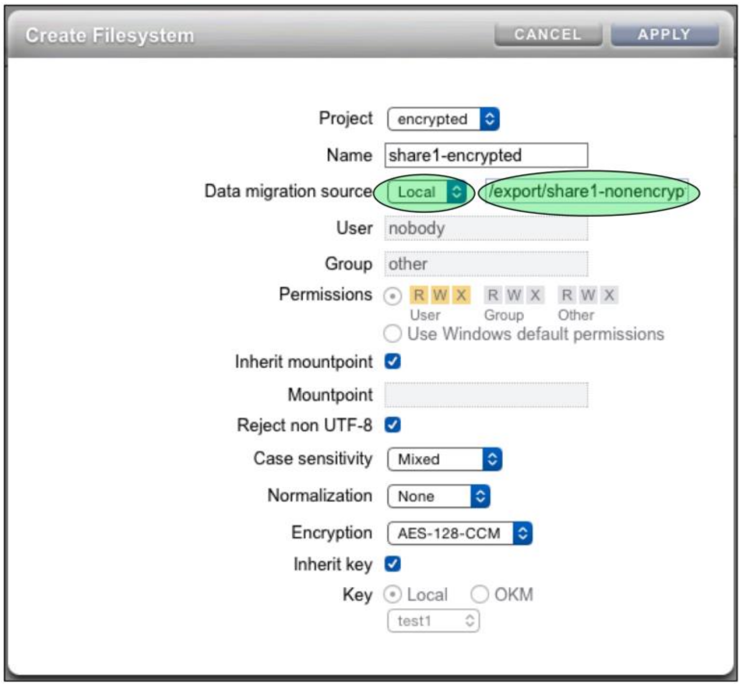
Figure 14. Using SSL/TLS for replicating an encryption-enabled share

Migrating Nonencrypted Shares to/from Encrypted Shares

The encryption feature must be specified for a share at the point of the share's creation, as it is not possible to engage the encryption feature after that point.

The Shadow Migration feature of Oracle ZFS Storage Appliance can be used for situations in which encryption is required for an existing dataset on an Oracle ZFS Storage Appliance. For each existing share, a new share must be created with the existing share specified as shadow in the *Data migration source* option in the Create Filesystem dialog window.

When considering a shadow migration task for multiple shares, check that the RAID option of the current pool still matches the availability requirements for the data set. If it does not match, this is the right moment to create a new pool and migrate all shares into the new pool using a new project definition that includes encryption-enabled.



Project: encrypted

Name: share1-encrypted

Data migration source: Local /export/share1-nonencryp

User: nobody

Group: other

Permissions: ● R W X (User) R W X (Group) R W X (Other)
○ Use Windows default permissions

Inherit mountpoint ☑

Mountpoint

Reject non UTF-8 ☑

Case sensitivity: Mixed

Normalization: None

Encryption: AES-128-CCM

Inherit key ☑

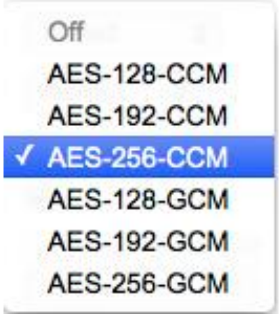
Key: ● Local ○ OKM
test1

Figure 15. Creating new shares through migration to change encryption status

You can also use Shadow Migration (using the same process just described) when there is a need to opt out of the use of encryption for shares.

Combining Deduplication and/or Compression on Encrypted Shares

All the options and protocols available for a share are available for shares enabled for encryption, with one exception. The nature of the AES–GCM encryption algorithm precludes any benefits gained from using deduplication. Therefore, it is highly recommended that you use any of the AES-XXX-CCM encryption type key options for shares that are used in combination with deduplication.



Off
AES-128-CCM
AES-192-CCM
✓ AES-256-CCM
AES-128-GCM
AES-192-GCM
AES-256-GCM

Figure 16. Encryption type option menu in Oracle ZFS Storage Appliance

Performance Considerations for Various Encryption Options and Configurations

When creating a project or a share you have the choice of what type of encryption to use. The choice depends on a few factors, regulatory requirements, performance impact, and use of deduplication (as mentioned in the previous paragraph). The following table summarizes the characteristics of each of the encryption options, listing them from low to high, in terms of Oracle ZFS Storage Appliance CPU performance impact.

TYPE OF ENCRYPTION	DESCRIPTION		
Off	Share or project not encrypted	Share or project is not encrypted	Dedupable
AES-128-CCM	Encryption mode with lowest CPU impact		Dedupable
AES-192-CCM			Dedupable
AES-256-CCM			Dedupable
AES-128-GCM		NIST SP800-38D recommended	Not dedupable
AES-192-GCM		NIST SP800-38D recommended	Not dedupable
AES-256-GCM	Encryption mode with highest CPU impact	NIST SP800-38D recommended	Not dedupable

In general, GCM type encryption uses roughly 20 to 50 percent more CPU resources than CCM type encryption. Select the type of encryption that meets the security requirements for the data sets used.

The performance impact of using encryption for applications that use small block size (8 Kb) random I/O workloads is far less than applications using a sequential I/O workload using large I/O block sizes (128 Kb to 1 MB). This is especially true for write operations as read operations can benefit from data being available directly from the data cache of Oracle ZFS Storage Appliance.

When using a large number of data volumes, it is worthwhile to investigate if there are parts in the data set that do not need to be encrypted and can be allocated on separate shares that do not have the encryption option enabled.

Keep in mind that when spreading data sets over different shares/projects it is recommended to keep them in the same project when using replication. A project acts as a consistency set, guaranteeing a constant data consistency over the multiple volumes/shares within a project.

When combining multiple data services, such as encryption, compression, and deduplication, more CPU resources are needed to serve the combination of these services for the selected shares/projects. When using multiple data services extensively, make sure Oracle ZFS Storage Appliance can satisfy the CPU demands of those data services.

Other general guidelines and performance considerations

- Encryption is a CPU-intensive process. Choose an Oracle ZFS Storage Appliance with a maximum number of CPU cores and a dual controller configuration for optimum performance.
- Large block sequential workloads (128 KB to 1 MB) pose the greatest encryption performance burden, especially with a 256-bit GCM type of encryption mode. If possible, use a lesser 128-bit CCM encryption mode for these types of workloads.
- Use of granular encryption (share/project) provides greater security and performance controls.
- Minimization of the use of other data services, such as replication, compression, and deduplication. These all compete for system CPU resources when encryption is enabled.

Recommendation for Applications Using Database Repositories

When using database type applications, the use of the Transparent Data Encryption feature is recommended. This approach provides an optimal integrated security solution, combining encryption and user authentication security functions. Transparent Data Encryption enables encryption of database columns or entire application table spaces. Its high-speed cryptographic operations make performance overhead negligible in most applications. Transparent Data Encryption integrates directly with frequently used Oracle Database tools and technologies including Oracle Advanced Compression, Oracle Automatic Storage Management, Oracle Recovery Manager (Oracle RMAN), Data Pump, Oracle GoldenGate, and more.

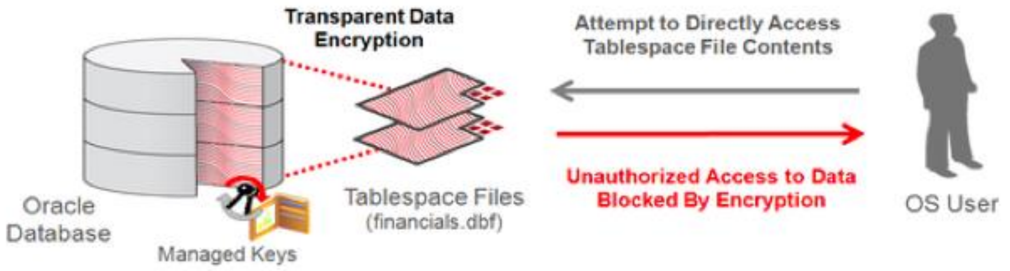

Transparent Data Encryption
Attempt to Directly Access Tablespace File Contents
Oracle Database
Managed Keys
Tablespace Files (financials.dbf)
Unauthorized Access to Data Blocked By Encryption
OS User

Figure 17. Applying Transparent Data Encryption for Oracle Database

Best Practices for Key Management

Key management should be focused around defining and executing strong policies and procedures that incorporate best practices. These policies and procedures depend on company/business security requirements, industry regulatory requirements, and government regulatory and security requirements. Requirements may differ per geographic location.

The following common-sense suggestions can be a helpful start.

Ensure keys are kept securely

This applies to both the keystore and any backup of the contents of the keystore. Make sure access to the keystore and managing of the contents of the keystore are controlled by setting the proper roles and access rights assigned to the users with those responsibilities. Access to the locations where any copies are kept must be tightly controlled as well. It makes no sense to properly secure access to the keystore but leave a backup copy of the keystore contents on a USB stick on a desk in the office.

Change the keys regularly

It is a best practice to have a policy in place for changing wrapping keys. Wrapping keys can always be changed on encrypted shares on Oracle ZFS Storage Appliance. This does not involve a re-encrypt cycle for the data in the share. A change of (the contents) of a wrapping key results in re-encryption of the encryption key used by ZFS on the encrypted share.

Manage how and to whom keys are assigned

Make sure that only users who are identified in the policies and security plan can access the keystore and manipulate the keys according to the security levels to which they have been assigned.

Decide on the granularity of keys

When encrypting shares, it has to be decided to do this on a per-Oracle ZFS Storage Appliance basis or per-project or per-share basis. Different projects/shares might be used by different areas in the business and have different requirements for type of encryption, key retention policies, and so on.

Appendix A: References

- Oracle ZFS Storage Appliance product information: http://www.oracle.com/us/products/servers-storage/storage/nas/overview/index.html
- Oracle ZFS Storage Appliance white papers and subject-specific resources: http://www.oracle.com/technetwork/server-storage/sun-unified-storage/documentation/index.html
- Recommended related white papers:
  - “Networking Best Practices with the Oracle ZFS Storage Appliance”
- Oracle ZFS Storage Appliance documentation library, including installation, analytics, customer service, and administration guides: http://docs.oracle.com/en/storage/
- Oracle Key Manager 3 Administration Guide (Appendix X, "Using OKM with Solaris ZFS Encryption") http://docs.oracle.com/cd/E50985_01/en/E41579/html/toc.htm
- "Critical Security Controls," SANS Institute. Consensus Audit Guidelines or CAG: http://www.sans.org/critical-security-controls
- Elaine Barker, William Barker, William Burr, William Polk, and Miles Smid. "Recommendation for Key Management—Part 2: Best Practices for Key Management Organization, Computer Security.” National Institute of Standards and Technology Special Publication 800-57, 2012. http://csrc.nist.gov/publications/nistpubs/800-57/SP800-57-Part2.pdf

Abbreviations and Terms Index

AES

Advanced Encryption Standard.

Agent

A storage device used to encrypt and decrypt data and using Oracle Key Manager to manage the encryption keys. An agent listener must be created in Oracle Key Manager for each storage device.

Encryption key

Data encryption key used to encode and decode data.

Keystore

A keystore is a repository of security certificates. In Oracle ZFS Storage Appliance, the keystore holds the wrapping keys used to encrypt and decrypt the actual encryption key that is used by ZFS. The ZFS share encryption key is used by ZFS to encrypt and decrypt the data in the ZFS share.

NIST

National Institute of Standards and Technology, Department of Commerce of the U.S. Government.

OKM

Oracle Key Manager, consisting of three main components:

(1) Key management appliance (KMA). A security-hardened box that delivers policy-based key management, authentication, access control, and key provisioning services. Multiple KMAs can be used to create a KMA cluster for extra reliability and redundancy for key storage.

(2) Key management software. Consists of two options: a GUI or CLI version to set up the KMA and perform key management functions. The software can be installed on an Oracle Solaris, Microsoft Windows, or Linux-based workstation.

(3) Hardware security module (HSM). A hardware cryptographic accelerator card.

Role

A set of permissions that is granted to a user on Oracle ZFS Storage Appliance and Oracle Key Manager to allow the performance of certain operations, like auditor, backup operator, compliance officer, operator, or security officer.

RSA

An algorithm for key encryption.

Wrapping key

A key used to decrypt and encrypt the ZFS encryption key. Wrapping keys are stored in the keystore in Oracle Storage Appliance.

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

blogs.oracle.com/oracle
facebook.com/oracle
twitter.com/oracle
oracle.com

Integrated Cloud Applications & Platform Services

Best Practices for Deploying Encryption and Managing Its Keys on Oracle ZFS Storage Appliance
September~~July~~ 2018
Author: Peter Brouwer, Ulrich Conrad, Application Integration Engineering