Oracle Key Manager Overview

WHITE PAPER / NOVEMBER 2018

DISCLAIMER

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Table of Contents

ORACLE KEY MANAGER OVERVIEW

INTRODUCTION

Oracle Key Manager is a comprehensive key management system designed to address the rapidly growing enterprise commitment to storage-based data encryption. Developed to comply with open standards, the application provides the capacity, scalability, and interoperability to manage encryption keys centrally over widely distributed and heterogeneous storage infrastructures.

Oracle Key Manager is specifically designed to meet the unique challenges of storage key management including:

- **Long-term key retention**: To ensure archive data is always available, Oracle Key Manager securely retains encryption keys for the full data lifecycle, which can exceed a decade in length.

- **Interoperability**: Oracle Key Manager provides the level of interoperability needed to support a diverse range of storage endpoints attached to open systems, cloud environments, or mainframe - all under a single storage key management service.

- **High availability**: With active N-node clustering, dynamic load balancing, and automated failover, Oracle Key Manager provides high availability whether the appliances are together in the same room or distributed around the world.

- **High capacity**: Oracle Key Manager manages large numbers of storage endpoints and even more storage keys. A single clustered appliance pair can provide key management services for thousands of storage devices and millions of storage keys.

Oracle Key Manager encryption supports the following encryption endpoints:

- Oracle Key Manager PKCS#11 Provider (pkcs11_kms):
    - Oracle Solaris 10, 11
    - Oracle Linux Server release 5.5+
- The Oracle Key Manager PKCS#11 Provider (pkcs11_kms) provides an interface for Transparent Data Encryption (TDE), a feature of Oracle Database since version 11gR2.

- Oracle ZFS Storage Appliance
- Solaris ZFS filesystems
- Oracle Key Manager Java Cryptography Extension (JCE) Provider
- Encryption-capable tape drives and libraries:
    - Oracle StorageTek SL150 Modular Tape Library
    - Oracle StorageTek T10000 family and 9840D
    - LTO4 and above

The Oracle Enterprise Manager System Monitoring Plug-in for Oracle Key Manager can be installed in an Oracle Enterprise Manager (OEM) environment in order to monitor OKM Key Management Appliances (KMAs) from an OEM Cloud Control Console.

This white paper addresses the following topics:

- Oracle Key Manager overview
- Basic operations
- Security features
- Key management practices
- Partner key transfer
- Backup and recovery practices
- Disaster recovery practices

ORACLE KEY MANAGER OVERVIEW

Oracle Key Manager consists of three main components:

- **Key Management Appliance (KMA)**: The KMA is a security-hardened box that delivers policy-based key management, authentication, access control, and key provisioning services. As a trust authority for storage networks, the Oracle Key Manager KMA ensures that all client endpoints (agents) are registered and authenticated and all encryption key creation, provisioning, and deletion are in performed in accordance with prescribed policies. Multiple KMAs connected via an IPv4 network form an Oracle Key Manager cluster. KMAs support IPv4 and IPv6 for management and agent communications.
- **KMA Management Software**
  - **GUI** - The KMA is a locked-down, security-hardened device. There are only very limited options available to a privileged security officer through the console or Service Processor of the appliance. Other than those specific operations, administration of Oracle Key Manager occurs through a GUI management program that is executed from a customer-provided workstation or server.
  - **CLIs** - To facilitate automation of various tasks the product includes two command line interface utilities that may be executed from a customer-provided workstation or server.
- **Hardware Security Module (HSM)**: The Key Management Appliance may be purchased with or without a FIPS 140-2 Level 3 certified hardware security module. A Thales PCIe-based HSM is available for shipment fully integrated with the KMA. A cluster may contain a mix of KMAs with and without HSMs. The HSM can also be added to a KMA after initial deployment. When an HSM is installed in a KMA, the appliance automatically configures the HSM to operate in FIPS mode.

The Oracle Key Manager cluster provides redundancy and increased bandwidth, and two networks (in addition to the KMA service processor’s network interface):

- Service network for communication between the encryption agents and the KMAs
- Management network for inter-KMA traffic and communication with remote management stations

This isolates the encryption endpoints from heavy corporate network traffic and improves response time for key requests. Oracle Key Manager interoperates with a variety of agents, including encryption-enabled tape drives and other encryption endpoints listed in the Introduction above. This white paper describes the general behavior of the Oracle Key Manager encryption solution with some endpoint examples using the Oracle StorageTek encrypting tape drives. Behavior specific to the LTO tape drive is included in Appendix B. Oracle Key Manager supports active N-node clustering with fully automated failover. Supported endpoints discover all KMAs in the system, and will interact with supported KMAs in the cluster (some agents will exclude down-level KMAs or KMAs not running in FIPS mode). Customer networks may be partitioned with KMAs and agents assigned to corresponding “sites”. Agents may prefer to use KMAs assigned to their site. Oracle recommends that each site have a pair of KMAs to provide continuous availability in the event that one KMA becomes busy or unavailable.

Any KMA is usable for administration functions, and changes made at any KMA asynchronously replicate to all other KMAs in the cluster. Keys generated at a site replicate to all other KMAs in the cluster to enable easy key sharing among sites and disaster recovery. Keys are pre-generated and not issued to endpoints until replicated. Administrative changes to a KMA propagate to all other KMAs in the cluster. The execution of all administration functions is through the Oracle Key Manager GUI, and any management endpoint can administer all KMAs in the cluster. The management CLI may also be used for various recurring tasks. See Figure 1 below.

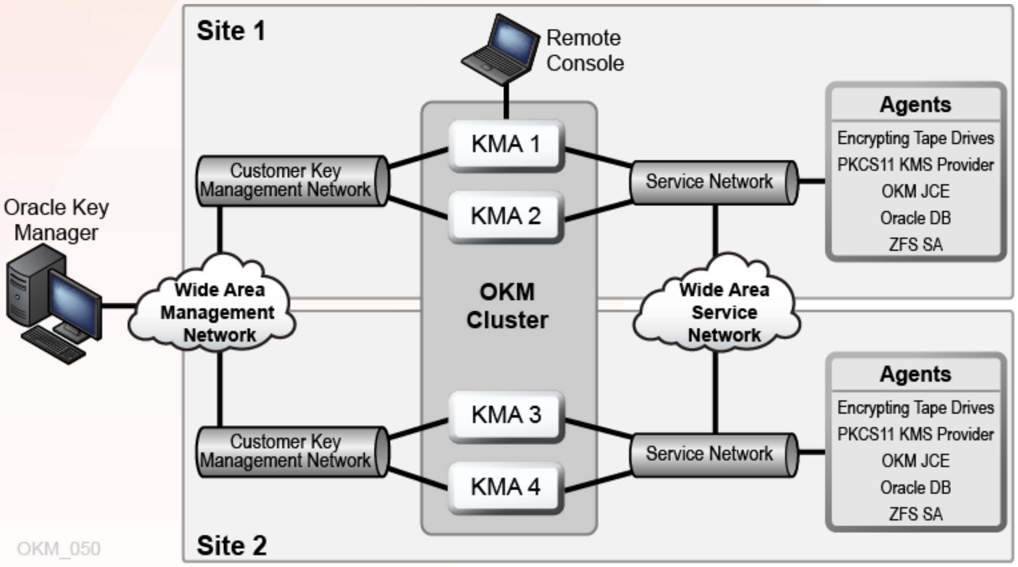

Figure 1: A typical multi-site configuration can be administered from a single administration workstation.

Note: Each KMA will have a second connection to the management network (not shown) for the Service Processor interface used to perform initial configuration tasks

Features

ORACLE KEY MANAGER FEATURES AND BENEFITS

FEATURES	BENEFITS
SECURITY	
Federal Information Processing Standard (FIPS)–certified cryptography	Uses FIPS 140-2 certified cryptography to provide Advanced Encryption Standard 256-bit encryption keys for application data
Role-based access control	Supports National Institute of Standards and Technology (NIST) SP800-60 operational roles to segregate operational functions
Quorum	Requires a minimum number of quorum members to activate a KMA, to create a user, to add roles to a user, and to restore Oracle Key Manager database backups; quorum parameters fully configurable
Hardened operating system	Provides additional security capabilities to prevent direct attacks on the Oracle Key Manager appliance.
Compliance	STIG reports may be generated and retrieved in addition to OKM auditing reports.
HIGH AVAILABILITY	
Active clustering and failover	Provides high availability through active N-node clustering with fully automated failover
Load balancing	Provides active load balancing for optimization
Near-synchronous replication	Provides near-synchronous, secure replication of transaction data among appliance nodes
ROLES AND ROLE SEGREGATION	
Granular role segregation	Provides operational segregation through six distinct role definitions—security officer, compliance officer, operator, backup operator, auditor, and quorum member—which are access controlled and functionally restricted, although users can have multiple roles if desired
Centralized role assignment	Provides centralized administration enabling the central assignment and administration of operational roles; automated data replication automatically distributes role data to all remote sites for disaster recovery purposes
Centralized policy management	Provides central management of data encryption policies through the compliance officer role; policy replication occurs automatically for disaster recovery purposes
OPERATIONS	
Robust API library	Provides a robust library of APIs to enable operational automation and third-party product integration; allows the rapid integration of third-party storage services, such as storage or backup application managers, to use Oracle Key Manager’s policy and key management functions
Audit logging	Maintains audit logs for all operational and key material events and transactions
Open standards	Supports open standards including standard certificate format X.509v3 certificates, Simple Object Access Protocol, and TLS
Management	
Secure management client	Provides a rich, cross-platform compatible client for local and remote management
Ease of use	Provides an easy-to-use and fully configurable management client
CERTIFICATION	
FIPS 140-2	Uses FIPS 140-2 certified cryptography for all key and cryptographic functions
NIST algorithm matching	Conforms to NIST guidelines for algorithm matching, including symmetric and asymmetric key pairing and hash and digital signatures

BASIC OPERATIONS

The Oracle Key Manager encryption system manages the following objects:

- Key policies and groups
- Agents
- Keys
- Data units

EXAMPLE 1

The following simplified example illustrates the basic working of the system using Oracle's StorageTek encrypting tape drives as an example.

- A backup or archive application makes a request to write data to a new tape volume.
- The volume is mounted in a drive with an encryption agent that has been enrolled in the cluster and given access to a key group (which is associated with a key policy that defines the characteristics of the keys in that group). Enrollment is a prerequisite step where an agent is securely authenticated with an OKM cluster. No key requests can occur until an agent has successfully enrolled.
- The tape drive reacts to the tape volume mount and uses its OKM cluster information to interact securely with a KMA. The embedded agent requests the creation of a data unit corresponding to the mounted volume and a new encryption key. The KMA pulls a key from its key pool, changes the key's state, associates the key with the agent's default key group, associates the key with the data unit, stores both in its database, provides the key to the agent, and replicates these changes across the network to all KMAs in the cluster.
- The encrypting tape drive accepts unencrypted data across the data path and encrypts that data with the new key. Volume dismounting occurs, and the encryption key is flushed from the drive's memory.
- The tape volume mounting occurs again for another write operation.
- The tape drive reads the data unit ID written on the media from a previous mount and has the embedded agent request keys associated with that data unit from a KMA in the cluster. Depending on the state of the key used on the previous write, which is determined by its key policy and the time elapsed since its activation, the agent either reuses that key to encrypt the new data to be appended or asks the KMA to create a new key to use on the current write operation.
- Data encryption encrypts the new data, dismounting of the volume occurs, and flushing of the key(s) from the drive's memory takes place.
- Sometime later, a request to read data written on the tape volume occurs.
- Mounting of the volume occurs again in an encrypting tape drive. The agent requests keys associated with that data unit from a KMA in the cluster. The agent selects the appropriate key, the tape drive decrypts the data, and returns it in unencrypted form across the data path to the application. Dismounting of the volume takes place, and the flushing of key(s) from the drive's memory occurs.

Key Policies and Groups

In Oracle Key Manager, keys are the primary managed object. Key groups are collections of keys used to encrypt data. Key groups may be used to implement multitenancy for separation of endpoints. Endpoints may not access, or create, keys within groups that they are not members. OKM uses groups for enforcing access control to keys. These groups are, in turn, associated with a key policy that defines the lifecycle of all keys in the group.

A key policy corresponds to NIST SP 800-57 cryptoperiod guidelines and specifies two important parameters:

- **Encryption period** - The length of time that a key should be used to encrypt data.
- **Cryptoperiod** - The length of time that a key should be used to decrypt data.

The encryption period and the cryptoperiod each start when the key is first issued to an agent. The cryptoperiod must be at least as long as the encryption period and is typically much longer. See Figure 2 below.

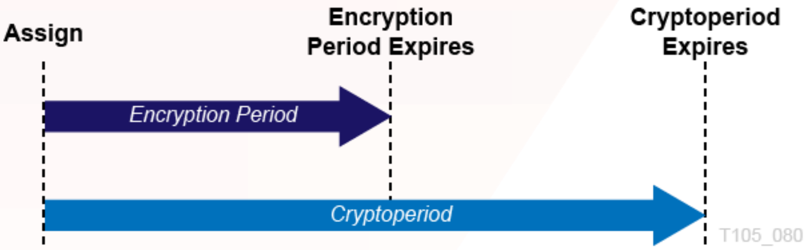

Figure 2: The relationship between the encryption period and cryptoperiod

A key may be used to encrypt data until its encryption period expires. Then that key should only decrypt data written with it. A key with an expired cryptoperiod may still be used to decrypt data if necessary, but is otherwise considered deactivated and can be destroyed. OKM issues an audit warning when expired keys are retrieved.

The following example demonstrates how the definition of a key policy can simplify key management and enable a key's lifecycle to closely mirror the retention period of the data it protects.

EXAMPLE 2

A backup application backs up 50 GB of data each day from each of 20 servers, uses a common drive pool with a different tape pool for each server's backups, and fills each tape in the pool completely before starting a new tape. Each data set must be retained for a period of one year. Data from one server fills a 500 GB tape volume every 10 days. A key policy is created with an encryption period of 10 days and a cryptoperiod of 54 weeks. A key group is created that uses this key policy and assigns this key group to be the default key group for every drive in the pool. With this configuration, the same key is used to encrypt all backup files written to one volume (regardless of what drive is used), and that key is deactivated a few days after the retention period for the last backup file written to the tape expires.

EXAMPLE 3

A backup application in a Multiple Virtual Storage (MVS) mainframe environment executes a daily backup. The application is assigned to a particular pool of tapes using drives that are not shared by other applications. Data retention is set for a one-year period managed by Oracle Key Manager, and the data is scattered upon an unknown number of tape volumes within that pool each day. A key policy is created with a one-year encryption period and a two-year cryptoperiod. A key group is created that uses this key policy and this key group is assigned as the default key group for every drive in the backup application's pool. All data written to a single tape volume by a drive in that pool will be encrypted with the same key for a period of one year. Call this key "Key 1." After the one-year encryption period for Key 1 expires, when the tape volume is next mounted, a request for a write operation results in Oracle Key Manager creating a new key: Key 2. Data written to this tape volume during the next year is encrypted with Key 2. The data written with Key 1 continues to be accessible, although Key 1 is deactivated two years after its activation date.

A key policy also indicates whether key groups associated with it enable exporting of keys from the group or importing of keys into the group. The section titled “Partner Key Transfer” discusses the use of these attributes in more detail.

Agents

An agent is an entity that implements the Oracle Key Manager agent protocol. For example, a peripheral storage device that performs encryption and decryption operations may embed an agent to obtains keys from OKM via the agent protocol. An agent must be enrolled in the Oracle Key Manager cluster before it is able to request keys from KMAs in this cluster.

TAPE DRIVE AGENTS

A tape drive ships in an unencrypted mode. Configuring the drive for encryption at a customer site entails going through an enrollment procedure.

Enrollment requires the following steps. Steps 1 and 2 are preparatory steps done through the Oracle Key Manager GUI by a user with operator privileges. Step 3, the actual agent enrollment, uses the Virtual Operator Panel (VOP) interface to the drive.

- **Step 1** Create an agent for the drive in a KMA, specifying an agent ID and passphrase.
- **Step 2** Assign one or more key groups to the agent and designate one group as its default key group. (When the agent requests the creation of a new write key, the agent’s default key group receives the key.)
- **Step 3** Use the VOP interface to set the drive offline. Navigate to Configure, Drive Data. Select the Encrypt tab. Enter the following information:
  - **Use tokens**: Select “No.” If “No” is selected, the VOP interface could be used to reset the drive and disable encryption at a later time. However, the drive will remain in encryption mode until the execution of this manual mode switch.
  - **Permanently encrypting**: Select “Yes” to place the agent permanently in encryption mode. Select “No” to allow disabling of encryption mode in the future.
  - **Agent ID**: Enter the agent ID specified in Step 1.
  - **Passphrase**: Enter the passphrase specified in Step 1. (This passphrase displays in plain text. A new passphrase is required to re-enroll the drive; therefore, securing this passphrase is unnecessary.)
  - **KMA IP address**: Enter the IP address of a KMA on the drive network. (The agent will use this IP address to contact the cluster and obtain the IP addresses of all KMAs in the cluster.)

When mounting a tape volume for the first time in a drive with an agent enrolled in an Oracle Key Manager cluster, the agent requests a KMA to create a new data unit and a new key in its default key group that it will use to encrypt data written to the volume. Keys transferred to a drive are stored in its memory only as long as the data unit with which they are associated is mounted. When dismounted, a tape volume has all keys associated with it flushed from its memory.

Typically, the agent passes to the KMA the barcode ID (VOLSER, or “volume serial number” plus media information) of the tape volume. The KMA populates the External Tag field of the data unit with this identifier, which appears on the barcode label on the exterior of the tape cartridge and is communicated to the drive by the library controller (in an automated library setting) or is read from an American National Standards Institute (ANSI) label on the tape. (A VOLSER is not available for data units associated with tape volumes without ANSI labels used in standalone drives.) However, neither the agent nor the cluster uses this tag for locating or processing data units. The External Tag field creates a convenient and easy way for the user to associate abstract data units displayed by the Oracle Key Manager GUI to physical tape volumes.

Key state changes that occur while a key is resident in the drive’s memory are unnoticed by the agent. Oracle Key Manager does not initiate communication with the agent, even when the objects it manages change state. When the drive has a mounted tape volume, the agent requests a KMA to send the keys associated with this data unit. The agent checks the state of each key as it comes in. Therefore, a state change for a key in use by the drive goes unnoticed by the agent until it receives that key from a KMA on a subsequent tape volume mount.

An agent should have access to all key groups associated with data units that it might process. When it requests the creation of a new key, that key will always be assigned to its default key group, but it can use a key in any other key group to which it has access to encrypt or decrypt data, as allowed by the key’s current state. For example, if a data unit with an active write key is mounted into a drive for a write operation, the agent will use this key if it has access to the key group to which the key belongs. If not, the application requesting the write might receive a returned write error. (The exception is a write from the beginning of tape, or BOT, when the drive does not have access to the active write key for that data unit. In this case, the KMA supplies a new key in the drive’s default key group, and the write proceeds.)

Keys

Each key used for encrypting data has a lifecycle determined by its key policy. It moves through a sequence of states that determine the operations the key can be used for.

KEY STATE TRANSITIONS

At startup, OKM generates a pool of pre-operational keys in the Generated state. Before becoming usable, a key must be protected against loss by automatic replication across a multi-node cluster or, in a single-node system, by manual creation of a system backup. Once protected the key can be used to encrypt data and is moved to the Ready state. (Because of the lack of automatic key protection in a single-node system, customers can only purchase multi-node systems.)

When first used to encrypt data, a key transitions to the Protect-and-Process state. Both its encryption period and its cryptoperiod begin at this time. In this state, the key can be used to encrypt and decrypt data. When its encryption period expires, it transitions to the Process-only state, at which time it can be used only to decrypt data. Eventually, the key’s cryptoperiod expires, and it moves to the Deactivated state. The expiration of the key’s cryptoperiod coincides with the end of the usefulness of the data that it protects, although the transition is purely logical. The key could still decrypt data if needed.

In normal operations, a key will transition from the Generated state to the Deactivated state as dictated by its key policy, and remain in that state indefinitely, enabling it to decrypt data as long as the data it protects exists. However, there is allowance made for events that would necessitate operator intervention into a key’s normal lifecycle.

At any point in a key’s lifecycle, if there is suspicion or knowledge that a key has been breached, a compliance officer can issue a manual declaration of compromise. A key in the Compromised state will no longer encrypt data but can decrypt data as required. Loss of an encrypted tape volume does not require declaring the keys associated with it compromised, because the keys used to encrypt the data are secure. However, if a Protect-and-Process key is exported inadvertently and shared with a key partner, marking the key as compromised might be appropriate to ensure that it does not encrypt any more data.

Customers might want to deny access to some data altogether to enforce retention policies. To allow this, a key that is either deactivated or compromised could be manually placed in the Destroyed state. Once a key is marked as destroyed, it will no longer be sent to an agent. If no backup containing the destroyed key exists, the key is marked as completely destroyed. Otherwise, it is marked as incompletely destroyed.

Because management of OKM backups is outside the control of Oracle Key Manager, the key state transition from Incompletely Destroyed to Completely Destroyed requires operator intervention. Once a backup file has been manually deleted, the backup operator can mark it as destroyed through the Oracle Key Manager GUI. When all backups containing a destroyed key have been marked as destroyed, the key's state will transition to Completely Destroyed. The system makes this key state transition automatically once all relevant backups have been marked as destroyed. However, it cannot verify the destruction of the backups containing the destroyed key.

For completeness, the system also allows a destroyed key in either sub-state to be compromised, moving it to the Destroyed Compromised state with the same sub-state. The Destroyed Compromised state is logically equivalent to the Destroyed state. See Figure 3 below.

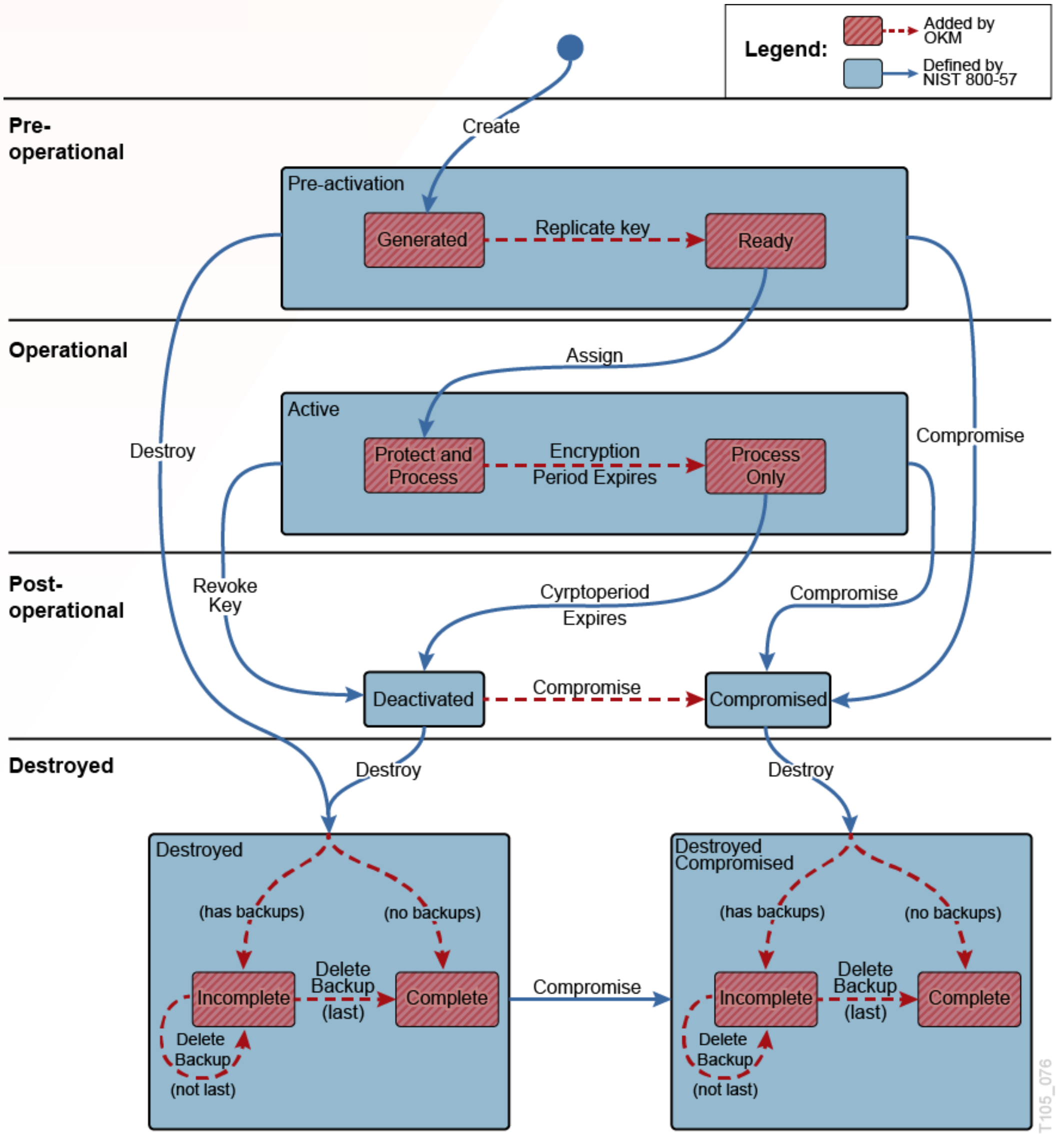

Figure 3: The states and transitions between states in the lifecycle of an encryption key

KEY DESTRUCTION

A destroyed key removed from Oracle Key Manager leaves only metadata attesting to its previous existence. It can no longer be used to decrypt data, and any data encrypted with it is effectively destroyed. Therefore, key destruction requires very careful consideration.

- WARNING: Key destruction can result in unintended loss of access to useful data. An Oracle Key Manager encryption agent cannot position past data for which it has no key. Therefore, data located past destroyed data on a tape is inaccessible, even if the key used to encrypt it still exists. In addition, imported StorageTek Crypto Key Management System 1.x data units can share a key across multiple data units. Destroying a key associated with one data unit might have the unforeseen side effect of destroying data on other data units.

The operator can select one or more data units and destroy post-activation keys associated with these data units. Options are available to destroy only deactivated keys, only compromised keys, or both. Because of the potential side effects, caution is necessary for this operation. Expiring the data through the backup or archive application has the same effect without the risk of unintended consequences.

If the destruction of keys is necessary, the safest policy is to wait until the deactivation of all keys associated with a data unit, and then destroy all keys associated with the data unit. Such a policy reduces the risk of unintended data loss. Having only one key associated with a data unit makes it much easier to adhere to this policy.

- WARNING: Any user with operator credentials can destroy deactivated or compromised keys.

Some protections are in place. It is not possible to destroy an active key. An active key must be marked as compromised before destruction. Marking a key as compromised requires compliance officer credentials--credentials not able to destroy a key. Thus, destroying an active key requires both compliance officer and operator credentials.

A single user with operator credentials can destroy a deactivated key. However, no user can deactivate a key. That transition happens only when the key's cryptoperiod expires, which should coincide with the last stage of the lifecycle of the data it protects. Therefore, if the cryptoperiod of a key is at least as long as the expected useful life of the data it protects, no single (rogue) user can destroy data encrypted with it until that data has outlived its usefulness.

Data Units

A data unit corresponds to endpoint items that hold encrypted data. For encrypting tape drives, a data unit is a tape volume; for ZFS it corresponds to a filesystem; for Oracle database it corresponds to the database. In Oracle Key Manager, once created, a key is associated with a data unit and remains logically associated with that data unit, even if the data it protects is overwritten or its key material is destroyed. The state of the data unit is a function of the state of the key(s) associated with it. For some endpoints, the agent will inform the Oracle Key Manager that the data has been overwritten. In this case, the Oracle Key Manager can change the in-use property of the Process-only keys associated with the data unit.

When created, the data unit is initially in No Key state (after the agent's request to create a data unit has completed and before its request to create a key has completed). As soon as a key is created for the data unit, the data unit moves to the Normal state. In this state, encrypted data can be written to the data unit. The data unit is readable, and new data can be written to the data unit. New data can either overwrite existing data or be written to a location that does not contain data. The details of this behavior are unknown to the Oracle Key Manager cluster and determined by the agent.

When the encryption period for the data unit's Protect-and-Process key expires, the data unit moves to the Needs Rekey state. This transition can occur while data is written to the data unit using the key whose encryption period has expired. The agent will continue to write data to the data unit using the expired key until the application managing the operation requests a dismount of the tape volume.

An agent may request all keys associated with a data unit that are still in use. If the data unit is in the Needs Rekey state, none of the keys transferred to the agent will be in the Protect-and-Process state. The agent may then request a KMA to create a new key for the data unit, and the data unit returns to Normal state. Over time, the data unit transitions back to the Needs Rekey state upon expiration of its Protect-and-Process key and typically remains in that state. However, if all of the keys associated with the data unit are destroyed, it moves to the Shredded state.

In that state, no data is readable from it, but new data can be written after requesting a new key, returning it to the Normal state. See Figure 4 below.

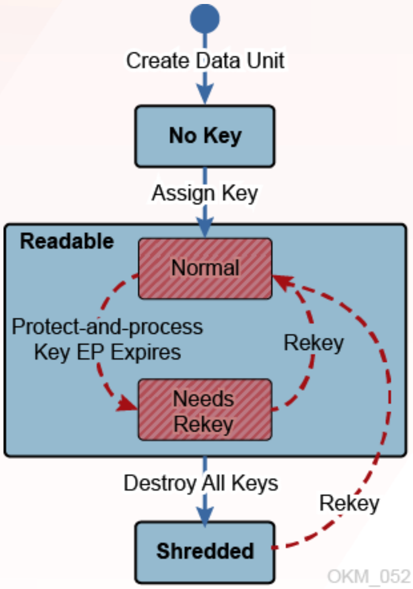

Figure 4: Data unit states and the allowed transitions for a data unit with a single key

Consider the following sequence of events using encrypting tape drives as an example:

- New media is mounted into an Oracle Key Manager encrypting drive as part of a backup operation. The agent requests the KMA to create a new data unit.
- The agent requests the KMA to create a key (Key 1) and encrypts and writes the backup data. The data unit moves to the Normal state.
- Sometime later, Key 1's encryption period expires, causing the data unit to change to the Needs Rekey state.
- The same media is again mounted for another backup operation. The agent requests the KMA to create a key (Key 2) and encrypts and writes the backup data. The data unit once again moves to the Normal state.
- Sometime later, Key 2's encryption period expires, causing the data unit to return to the Needs Rekey state.
- Eventually, Key 1's cryptoperiod expires, leaving it deactivated. Likewise, Key 2's cryptoperiod expires, leaving it deactivated.
- The operator destroys Key 1 and Key 2, causing the data unit to move to the Shredded state.

Figure 5 provides a visual representation of the states of the data unit and two keys involved in this sequence. The diagram consists of the following parts:

- State transitions for the data unit are in the center of the diagram.
- State transitions for the two keys are on either side.
- Numbered circles correspond to events in the sequence.

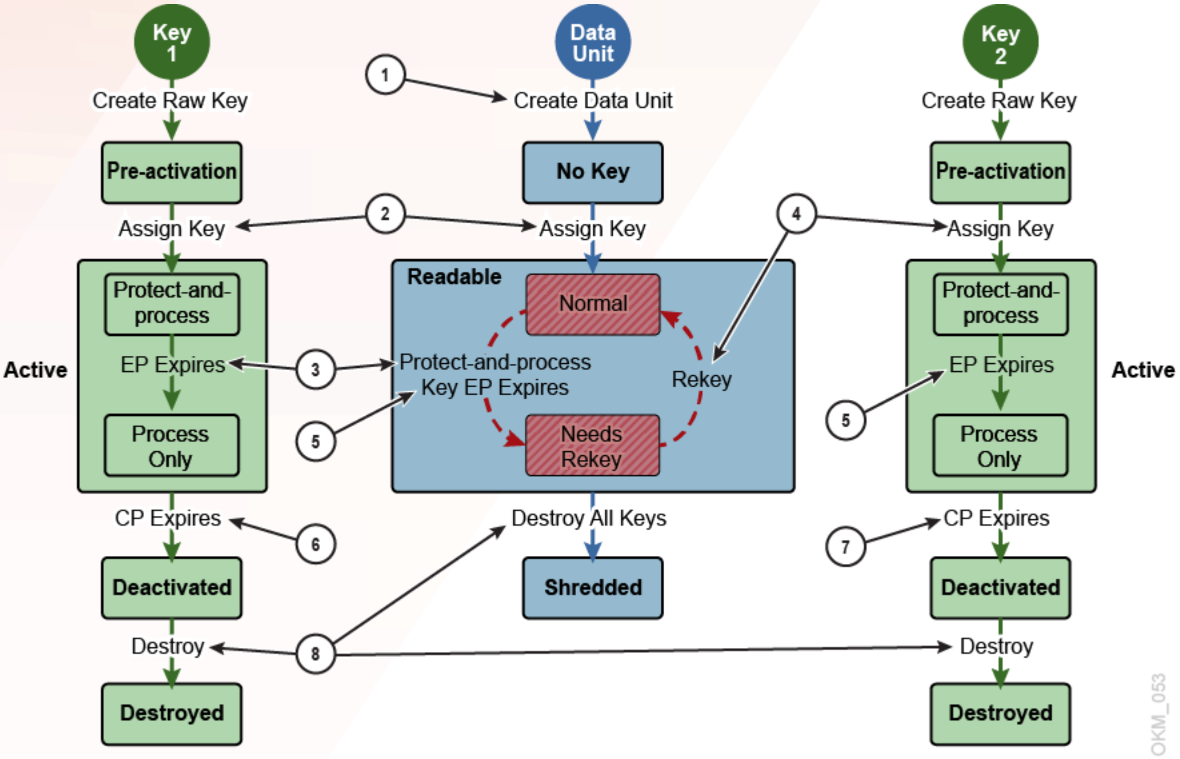

Figure 5: The states of the data unit and two keys involved.
Sequence:
1. Create Data Unit
2. Create Key 1
3. Key 1’s encryption period expires
4. Create Key 2
5. Key 2’s encryption period expires
6. Key 1’s cryptoperiod expires
7. Key 2’s cryptoperiod expires
8. Desrtroy Keys 1 and 2

Recycling/Scratching Data Units

In a tape environment, a tape volume can be recycled or set for scratch when the data on it is no longer necessary. When this happens, BOT writes new data, overwriting existing data on the media. The overwritten data is lost, and data not yet overwritten is only recoverable using special recovery tools. Therefore, the keys previously associated with the data unit become unnecessary. To reduce the number of keys that must be transmitted to the agent when the tape volume is mounted again, when the agent receives a request to write the media from BOT, it disassociates all keys previously associated with the data unit that are not in the Protect-and-Process state. (The agent will reuse an existing Protect-and-Process key to write new data until its encryption period expires.)

The term disassociates is not literal. Actually, a dissociated key continues to appear in the key list for the data unit, but two changes occur because of this action:

- The key’s In Use by Data Unit attribute changes from true to false.
- When the agent requests keys for this data unit, this key is no longer included in the response.

If a key is imported from a StorageTek Crypto Key Management System 1.x system, it might be associated with multiple data units. The disassociation operation applies only to the data unit currently mounted.

SECURITY FEATURES

An in-depth discussion of the various levels of security implemented in this solution is outside the scope of this white paper. The subsections below address only a few of these features.

Secure Communication

The communication protocol between an endpoint's agent and a KMA, between a new KMA added to the cluster and an existing KMA, or between a user and a KMA is the same. In each case, the system uses a registered passphrase for the entity initiating the communication to perform a challenge/response protocol. If successful, the entity is provided with a certificate and its corresponding private key. This certificate and private key are used with the TLS protocol. Mutual authentication is performed - both ends of any connection authenticate the other party.

For an encrypting tape drive, the authentication process of its encryption agent is performed using the VOP, or MD-VOP, tool. For other endpoint types there is typically a configuration tool to assist with enrollment; see the Oracle Key Manager 3 Administration Guide appendices. For a KMA joining an existing cluster, authentication is part of the QuickStart process, which is also described in the Oracle Key Manager 3 Administration Guide. For most agents, the certificate and private key are retained securely and used in subsequent OKM communication to bypass the enrollment process. For Oracle Key Manager GUI users, the enrollment process repeats every time the user logs in, because the assumption is that users might log in from different workstations.

FIPS 140-2 Level 3 Hardware Security Module

A KMA may be configured with a supported Hardware Security Module (HSM). This card is a PCI Express-based host bus adapter that installs into one of the slots of the KMA. Use of this card provides a FIPS 140-2 Level 3 certified HSM for advanced cryptographic security. The KMA automatically configures and manages the card to operate in FIPS 140-2 Level 3 mode.

AES Key Wrapping

AES Key Wrapping (RFC 3994) is used with 256-bit key encryption keys to protect symmetric keys as they are created, stored on the KMA, and transmitted to agents or within key transfer files. The only exception is for older agents that do not support the AES key wrap-specific calls within the agent protocol. For these older agents, the keys are unwrapped and transmitted in plain text within the protected TLS channel. When the cluster is enabled for FIPS mode, these older clients will not able able to retrieve keys from the cluster since AES Key Wrap will always be used.

Key Replication

When the first KMA of a cluster initializes, a large pool of raw keys is generated. When another KMA is added to the cluster, those raw keys are replicated to the new KMA and are then ready to be used to encrypt data. Each KMA that joins the cluster generates a pool of keys and replicates them to other cluster members. All KMAs will generate new keys as needed to maintain the key pool size so that ready keys are always available for agents. When an agent requests a new key for a data unit without a protect-and-process key, a raw key in the Ready state is drawn from the KMA's pool and assigned to the agent's default key group and to the data unit. The KMA database updates from this transaction are then replicated across the network to all KMAs in the cluster. At no time is any clear text key material transmitted across the network.

Role-Based Software Access

Access to Oracle Key Manager is limited to users who have been trusted with permission to perform certain roles. The following user roles are defined in the system:

- **Security Officer**: Manages security settings, users, sites, and transfer partners.
- **Compliance Officer**: Manages key policies and key groups and assigns key groups to agents and transfer partners.

- **Operator**: Manages agents, data units, and keys.
- **Backup Operator**: Performs Oracle Key Manager database backup and restore operations.
- **Auditor**: Views information about the Oracle Key Manager cluster.
- **Quorum Member**: Approves pending operations that require a quorum.

A single user account might have permission to perform multiple roles, and multiple user accounts might have permission to perform the same role. Some operations, such as sharing keys between trusted partners, might be multistep tasks that involve more than one user role. For maximum security, a different user should perform each role. For maximum convenience, one "superuser" might have all roles. A compromise that divides the roles among two or three users might provide an acceptable level of security and flexibility.

Quorum Protection

Some operations are critical enough to require an additional level of security. These operations include adding a KMA to a cluster, unlocking a KMA, creating users, adding roles to users, restoring Oracle Key Manager from a backup, and configuring key transfer partners. To implement this security, the system uses a set of key split credentials in addition to the role-based access described above.

Key split credentials consist of a set of user ID/passphrase pairs, together with the minimum number of these pairs necessary for the system to enable completion of certain operations. The key split credentials are also referred to as "the quorum" and the minimum number as "the quorum threshold". Operations that require that key split credentials be provided are referred to as "quorum operations."

Oracle Key Manager allows a maximum of 10 user ID/passphrase pairs. The quorum threshold can be set anywhere from one to the number of user ID/passphrase pairs defined. Setting it to one enables the completion of a quorum operation with only one quorum member present. Setting the threshold to the total number of pairs defined requires that all quorum members must approve these operations. The most common case would be to set this to a value greater than one, but less than the total, for example, three of five. This choice ensures that completion of these operations requires more than a single (possibly rogue) quorum member but allows for the situation where a member is unavailable. The user ID/passphrase pairs defined for the quorum are unrelated to the user roles described above. A user with quorum member credentials must first log into the cluster using the Oracle Key Manager GUI and then provide a user ID/passphrase pair to approve an operation.

KEY MANAGEMENT

Key Policies and Groups

An agent requests the KMA to create keys in its default key group as needed for data encryption. The key policy associated with this key group defines the lifecycle of these keys as well as whether keys can be exported from or imported into the group. As mentioned earlier, it is recommended that the key policy be defined so that the encryption period is sufficiently long to allow all data on one data unit to be encrypted with the same key. This reduces the total number of keys in the system and allows all data on one data unit to be managed as a whole, rather than on an individual file basis.

For tape encryption a good rule of thumb is to set the encryption period to at least three months, or longer if small amounts of data will be written at intervals until the tape is full. For example, a full year of biweekly payroll data might be stored on a single tape volume. To ensure that all data on this data unit is written with the same key would require a key policy with a much-longer encryption period.

Most open systems backup applications and some MVS mainframe applications will fill one data unit before writing to another, making the typical time required to fill a data unit fairly short. However, choosing a longer than needed encryption period is not a problem. Even if the encryption period is longer than the time required to fill a data unit, the key will not be reused, effectively terminating its encryption period when the tape is full. However, choosing a shorter than recommended encryption period can result in an agent rekeying a data unit several times before it is full. This will result in large numbers of keys being created, degrading Oracle Key Manager performance.

The Virtual Storage Manager (VSM) application of Virtual Tape Control System / Virtual Tape Subsystem (VTSS) and other MVS mainframe applications will not necessarily fill a tape data unit before writing to another data unit. Because of this, it makes sense to maintain longer encryption periods for the data units so the number of keys created for each data unit is kept to a minimum. The data written to Multiple Virtual Cartridges (MVCs) by the VSM in the form of a virtual tape volume is compressed by the VTSS then encrypted by the encryption agent.

Multiple key groups can use the same key policy. However, for many customers, a single key group per Oracle Key Manager cluster should suffice. This guarantees that any data unit known to the cluster can be accessed by any agent enrolled in the cluster. If data is to be shared with another cluster within the enterprise or with another enterprise partner, a new key group with the Export From attribute set could be created. Keys to be shared would be moved to this new key group, which would be assigned to the key transfer partner. The section titled “Partner Key Transfer” discusses the partner key transfer operation in much more detail.

Key Management System Clusters

Oracle Key Manager allows users the flexibility to structure their cluster to provide secure data protection while ensuring access to authorized users and ease of management. Clustering KMAs enables workload balancing and enables replication of key material as well as information about users, agents, key policies, and key groups. An enterprise can have agents (such as encrypting tape drives) in different locations within the same site or at multiple, geographically separated sites. Adherence to the following guidelines will ensure efficient and reliable performance:

- To reduce network latency, the KMAs servicing a pool of agents should be connected to the same network subnet as the agents.
- To spread the workload and avoid disruption during KMA outages (software upgrades or hardware failures), each pool of agents should have access to at least two KMAs.
- To prevent catastrophic loss of key data, KMAs located at two (or more) geographically separated sites should be clustered. This ensures business continuance in the event of a disaster that disables an entire site.

It is assumed that critical data generated at any site is replicated and vaulted offsite. If a site is lost, this backup data can be transferred to another operational site. Data units and keys associated with encrypted data (such as tape volumes) will be known to the KMAs at the sister site, and encrypted data required to continue business operations will be available. The damaged portion of the cluster can be restored easily at the same or a different location once site operations resume. The section titled “Key Management Appliance Recovery” describes how this restoration is accomplished.

PARTNER KEY TRANSFERS

KMAs at sites within an enterprise that routinely share data can be clustered so that all keys in use at any site are automatically available to all sites. In that situation, only tape media must be transported between sites to allow the desired sharing. However, the need might arise to share encrypted data between separate Oracle Key Manager clusters within the same enterprise or between enterprise partners. Oracle Key Manager provides a secure mechanism by which such key sharing can be accomplished. The key sharing mechanism is based on each partner having a public/private key pair generated by Oracle Key Manager at its site. A transfer key file containing a list of data units and keys to be shared is generated at the sender’s site. This file is encrypted using the receiver’s public key and signed with the sender’s private key. This ensures that only the receiver can access the shared keys and allows the receiver to verify that the expected sender is the source. The process is as follows:

- A security officer at each site obtains the public key generated by Oracle Key Manager, or creates a new public/private key pair, using the Oracle Key Manager GUI. (Only the public key information is displayed.) The partners exchange public keys. For each key, a fingerprint is generated at the sending site and computed at the receiving site. This fingerprint should be compared across sites to verify that no corruption has occurred during transfer of the key. For security reasons, this fingerprint is not sent along with the public key but verbally verified later.

- At each site, a security officer creates a key transfer partner through the Oracle Key Manager GUI, supplying the public key received from the partner site. Creating a key transfer partner is a quorum operation, so the threshold number of quorum members must be present to validate this operation. When the partner’s key is input, a fingerprint for that key is computed by Oracle Key Manager. If no corruption has occurred in transit, this fingerprint will match the one that was generated at the partner’s site at the time the public key was created.

- Each site assigns to its key transfer partner one or more key groups from which keys may be exported or to which keys may be imported. Key transfer can be two-way, so a site could be both an import site and an export site. These key groups must be associated with key policies that allow the desired export or import operation. This attribute of a key policy is modifiable if it was not set when the key policy was created.

- An operator at the sending site creates one or more key transfer files, which contain the data units and keys to be shared. A key can only be exported if it satisfies the following conditions:

- It belongs to a key group associated with the relevant transfer partner.

- Its key group is associated with a key policy that allows the desired export or import operation.

- It is active (in the Protect-and-Process or Process-Only state), deactivated, or compromised, with its In Use by Data Unit attribute set to true.

Each file is created by selecting the data units that are to be shared. Only data units with associated keys that meet the stated conditions can be included, and only the keys associated with those data units that meet the stated conditions will be included in the transfer file. To simplify the sharing process, write data to be shared to new or recycled tape volumes, change the key group for the keys associated with each of these data units to the export key group, and export keys for all of these data units. This process will ensure that all data intended to be shared will be accessible at the receiving site and only data intended to be shared is sent to the transfer partner.

- The sender's cluster encrypts the transfer key file using the receiving site's public key and signs the file using the sender's private key.
- The sender transfers the key file and the associated encrypted data to the receiver.
- The receiver enters the data tapes into the library and imports the tape volumes into the backup application.
- The receiver imports the keys and associated data units into the receiver's OKM cluster using the key transfer file, specifying a key group assigned to the key partner (sender) as the destination key group. Importing of keys does not require quorum validation.
- The receiver's KMA decrypts the transfer file and verifies the sender as its source. The KMA creates entries in its database for the shared keys and data units and associates each new key with the appropriate data unit.
- The keys and data units imported are replicated to all KMAs in the receiver's cluster. This process might take some time if a large number of items are shared. Decrypting data that require the imported keys might fail until sufficient time has elapsed for replication to complete. Data unit and key IDs contain the ID of a KMA in the creating cluster. Therefore, data units and keys imported into the receiver's cluster will have IDs distinct from those created by the receiver's cluster.

KEY MANAGEMENT APPLIANCE RECOVERY

Oracle Key Manager enables two types of recovery: recovery of individual KMAs, and recovery of the entire Oracle Key Manager cluster. Recovery of a single KMA can be accomplished with no impact to the rest of the cluster as long as one KMA for each drive pool remains operational. Furthermore, recovery of N-1 nodes of an N-node cluster can be accomplished without loss of any critical data. The following subsections address scenarios that require recovery of a single KMA. Recovery of an entire cluster is addressed in the section titled "Disaster Recovery."

Software Upgrade

Software upgrades can be applied to individual KMAs within a cluster while the cluster continues to provide key management services to agents thus providing continuous availability. When a KMA is unavailable during an update, agents will detect this and fail over to another available KMA in the cluster. Upgrading the software on a KMA requires two steps: uploading and applying an OKM software upgrade file, and activating the new software. Upgrade files can be quite large, therefore, the upgrade file should be uploaded and applied on one KMA at a time. Activation of the new software requires a reboot of the KMA server, therefore, KMA activations must be staggered so that at least one KMA connected to each agent pool is active at all times. As each KMA comes online, updates completed in the cluster while it was offline are replicated to it so that all KMAs in the cluster are resynchronized. Subsequent KMA upgrades should wait until resynchronization completes.

Network Disconnect

When a KMA is disconnected from the management network, the remaining KMAs in the cluster continue to attempt to contact it and will report communication errors in the audit event log. (The server reboot required to activate new software, as described above, produces this same behavior.) Agents communicate without interruption with the remaining operational KMAs attached to their service network. When the KMA is reconnected to the network, updates completed in the cluster while it was offline are replicated to it.

Hardware Failure

When a KMA server fails, the recovery procedure depends on the component that fails. If the component is one of the following components, then that component can be replaced.

- Fan modules
- Power supplies

- PCIe riser
- Power distribution boards (PDBs)
- Paddle card
- Connector breakout board
- HSM PCIe card
- Motherboard component
  - DIMMs
  - CPUs
  - Battery
  - Disk backplane

If some other component (such as the motherboard itself or a Network Interface Card) fails, then the entire server should be replaced. Network Interface Cards (NICs) are not replaceable independently of the motherboard, so they are not a Field Replacement Unit (FRU) for the KMA.

If possible, the KMA to be serviced should be logged out of the OKM cluster so that KMAs will refrain from attempting to replicate updates during the service outage. Once the service procedure is completed, the KMA can be rejoined to the cluster.

Replacing an Entire KMA

To replace an entire KMA, it should first be deleted from the cluster so that the remaining KMAs will no longer attempt to communicate with it. If the OKM console is still accessible, the option to reset the KMA to factory default settings could be executed. This reset operation will return the unit to its factory defaults. Disposition of the failed server is handled by the customer. The reset operation can also be used prior to moving a KMA from one cluster to another. When a KMA has been reset, it will appear exactly like a new KMA fresh from the factory except that the Service Processor network configuration persists. These network parameters can be reset during the QuickStart process. The optional disk scrub can be used to remove all traces of its former cluster.

A replacement KMA server is then configured and added to the cluster, as described in the Oracle Key Manager Installation and Service Manual and OKM Administration Guide. Once the new server is known to the cluster, cluster information is replicated to it, and it becomes an active member of the cluster.

Replacing a Disk Drive

To replace the disk drive in a KMA, it should first be deleted from the cluster so that the remaining KMAs will no longer attempt to communicate with it. If the OKM console is still accessible, the option to reset the KMA to factory default settings could be executed. This reset operation will return the unit to its factory defaults. Disposition of the failed disk drive is handled by the customer.

A new disk drive preloaded with OKM software is then installed in the KMA and the KMA added to the cluster, as described in the Oracle Key Manager Installation and Service Manual. Once the server is known (again) to the cluster, cluster information is replicated to it, and it becomes an active member of the cluster.

KEY MANAGEMENT SOLUTION BACKUP

Backing up Oracle Key Manager involves two types of backups: a core security backup and a database backup. The following subsections describe the purpose of each and when each type of backup should be taken.

Core Security Backup

The core security backup backs up the system master key and quorum information and is required if a system is to be restored from a database backup. This backup should be taken immediately after a cluster has been configured and again whenever a change is made to the quorum.

As described in the section titled "Security Features," the passphrases of the quorum members are used to protect the system master key. The master key is split into N pieces, where N is the size of the quorum, in such a way that the entire key can be reconstructed from M pieces, where M is the quorum threshold. The passphrase of each quorum member is used to generate a key that is used to encrypt one of the N pieces of the master key. When the user IDs and passphrases of any M of the N quorum members are provided, the system can reconstruct the master key. This master key is required to complete critical security operations.

The core security backup file contains the system master key, split and wrapped as described above. The file is an XML file in human-readable form. The user IDs of the quorum members are in plain text, which can be useful to verify the identities of the quorum members. Key and passphrase information in the file is securely encrypted. In particular, the master key is well protected. Reconstruction of the master key would require access to the algorithm used to split the master key, the user IDs and passphrases of the required subset of quorum members, the algorithm used to generate keys from these passwords, and the algorithm used to encrypt the key splits with these keys. Even so, each copy of the core security backup file should be kept in a very secure place. The quorum information should be modified in the event a quorum member is added or removed or a member's passphrase is compromised. A new core security backup should be taken after the quorum information is modified, and all copies of the previous core security backup file should be destroyed completely.

The core security backup operation is done using the Oracle Key Manager GUI or CLI. The file name for the core security backup file generated by this operation is stored in the database. However, the core security backup file itself is stored on the Oracle Key Manager administration workstation or some remote location accessible from it, not on any KMA server, and managed entirely outside the Oracle Key Manager system.

To provide the required level of protection, the core security backup file could be placed on a thumb drive and given to a highly-trusted employee. That employee would keep the file at all times. When a quorum change is necessary, the thumb drive containing the first file would be crushed to prevent any possibility of future access. A new core security backup would be taken, placed on a new thumb drive, and protected with the same level of security.

Oracle Key Manager Backup

In most business-critical systems, the security of the database is protected by regular backups with journaled updates, which can be applied to ensure that the most recent changes are included if a restore from backup is required. Backing up these systems is critical because only a single instance of the database can exist.

Each KMA stores OKM cluster information in an internal database. The security of this Oracle Key Manager database is provided by its replication across multiple clustered KMAs located at geographically separated sites. Restoration of the Oracle Key Manager database from backup will be necessary only if every KMA at every site is destroyed at the same time. The Oracle Key Manager GUI provides a database backup facility to create a point-in-time copy of a KMA's database for use in such a circumstance. The KMA chosen for the backup should be synchronized with the cluster as a best practice.

Backing up the database is done by a user with backup operator credentials who is connected to a single KMA in the cluster. The backup operation produces a copy of the instance of the database residing on the KMA to which the user is connected. The operation does not disrupt agent operations as long as every agent has access to at least two KMAs. Other KMAs in the cluster provide uninterrupted service while the backup operation is in progress. The instance of the database on the KMA creating the backup is resynchronized after the operation is complete. Backing up the database produces two files: a backup file and a backup key file. The backup file is a copy of the database encrypted using the key stored in the backup key file. The backup key in the key file is encrypted using the system master key, which is contained in the core security backup file.

Therefore, the core security backup file, the backup key file, and the presence of the required subset of quorum members are all required to restore an Oracle Key Manager cluster from a backup file. The passphrases of the quorum members are needed to decrypt the system master key, which is needed to decrypt the backup key□which, in turn, is needed to decrypt the backup file. The file names for the backup files produced are stored in the database. However, the backup files themselves are stored on the Oracle Key Manager administration workstation or some remote location accessible to it, not on any KMA server. These files are totally outside the control of the Oracle Key Manager and should be managed carefully. Backup files should be backed up and stored offsite, away from the cluster and separate from the core security backup file needed to unlock them. Although the system currently provides no utility to set up automated backup schedules, regular backups can be automated using the Oracle Key Manager CLI..

Updates made by another KMA and not yet replicated to the KMA executing the backup, or made after a backup is taken will not be included if that backup is used as the source for a cluster restore. However, the restored system might be able to recover key and data unit information not included in the backup.

EXAMPLE 4

The creation of a key and a new data unit occurs on one KMA at the same time that a database backup is initiated on another KMA in the cluster. The backup operation captures the state of the database on the KMA executing the backup and does not include the new key or the new data unit. Sometime later, a disaster causes the loss of the entire Oracle Key Manager cluster, and the cluster is restored using this backup. A request is made to the new cluster to restore the data written at the time the backup was in progress. The drive reads the data unit ID from the media and requests all keys associated with that data unit. Oracle Key Manager has no record of that data unit, so the drive requests the specific key associated with the requested data. Oracle Key Manager also has no record of this key being assigned. However, it does have a list of unassigned keys (pre-generated keys in the ready state) generated in the original cluster before the backup was taken. If it finds the ID of the required key in this list, it assumes that this is a recovery operation, creates a new data unit, associates the key with the new data unit, and provides the key to the drive.

Some scenarios exist in which an assigned key might not be recoverable from a backup copy of the database. Nevertheless, the existence of multiple active copies of the database at geographically separated locations provides an extremely high level of security. In addition, no key material is ever delivered to an agent until it has been replicated across the cluster. Therefore, the system ensures access to critical data with a very high level of confidence.

- WARNING: No restore from backup can be done without a core security file, but any core security backup file can be used to restore a cluster from backup as long as the requisite threshold of quorum user ID/passphrase pairs stored in that file is provided. Therefore, copies of outdated core security backup files should be destroyed, and copies of the current core security backup file should be stored securely in different locations from each other and from the database backup files.

DISASTER RECOVERY

An Oracle Key Manager clustered environment that spans multiple, geographically-separated sites dramatically reduces the risk of a disaster destroying the entire cluster. In the unlikely event that an entire cluster must be recreated, most key data can be recovered by recreating the Oracle Key Manager environment from a recent database backup.

Many companies employ the services of a third-party disaster recovery (DR) site to allow them to restart their business operations as quickly as possible after a catastrophe. Periodic unannounced DR tests demonstrate the company's degree of preparedness to recover from a cataclysmic event. A number of possible scenarios exist, three of which are discussed here:

- **Scenario 1**: The company maintains a KMA that is part of its Oracle Key Manager cluster at the DR site. The DR site provides a pool of drives with network access to the KMA.
- **Scenario 2**: All of the company's KMAs are destroyed. The DR site provides a pool of drives with network access to a standalone KMA that is used at different times by different client companies.
- **Scenario 3**: The company maintains a KMA that is part of its Oracle Key Manager cluster at DR Site 1, but the DR test is run at Site 2, which has a standalone KMA attached to a pool of drives, as described in Scenario 2.

Scenario 1

This scenario is optimal; the company's Oracle Key Manager database is intact even though all of their sites have been destroyed. The primary focus is restoration of the data center and critical business systems. Move vaulted tapes to the DR site. Connect a laptop running the Oracle Key Manager GUI to the KMA network, and enroll the drives provided by the DR site in the company's KMA. A user with operator credentials (but no quorum) is required. Connect a host running the required backup application to the drives at the DR site and use it to restore the company's data center.

The Oracle Key Manager database is intact, so restoration of the company's Oracle Key Manager cluster is simple. The KMA at the DR site serves as the first KMA in the cluster. Remove the agents for the drives at the DR site. Replace the destroyed KMAs and add them, one at a time, to the cluster (as described in the section titled "Hardware Failure"). The database is replicated across the cluster. Create and enroll agents as replacement drives are brought into service.

Scenario 2

This scenario requires restoration of the company's Oracle Key Manager from a backup file. The following bulleted list describes the process for restoring the cluster. The DR site provides:

- A KMA, either fresh from the factory or reset to factory defaults by the previous client.
  - Note: Resetting the KMA is a console operation that requires the security officer's login. Unless a new KMA is provided each time, the DR site must ensure that clients reset the KMA using the security officer login before that person leaves the facility. This protects the client's data and allows the KMA to be reused by the next client. The scrub option on the reset should always be used to wipe all trace of the client's data from the server hard disk.
- Endpoint applications (agents) attached to the KMA network
- An administration workstation on the KMA network running the Oracle Key Manager GUI
- The company supplies:
- A trusted operator

- An OKM core security backup file, the most recent database backup file, and the corresponding database backup key file
- The required threshold number of quorum members whose user ID/passphrase pairs are contained in the core security backup file
- The encrypted data needing to be decrypted with keys managed within OKM.

The operator follows the steps prescribed in the the Oracle Key Manager Version Administration Guide, using the option to restore an OKM cluster from backup. DR site personnel must provide the relevant network information required by the QuickStart process.

Once the QuickStart procedure is completed, the operator configures the cluster as follows:

- **Step 1** Load the core security backup file, the backup key file, and the backup file into the management workstation. For security, do not leave the backup files on the management workstation after this process is complete.
- **Step 2** Start the Oracle Key Manager GUI and connect to the KMA using the security officer login created during the QuickStart process.
- **Step 3** Navigate to the Backup List panel, click the Restore button, and supply the complete pathname for each of the three backup files.
- **Step 4** Allow each quorum member present to input a user ID and passphrase when prompted by the software.
- **Step 5** Click Start to initiate the restore. The time required to complete the restore operation can vary from an hour to several hours, depending on the size of the original cluster.
- **Step 6** Once the restore is complete, remove the media containing the OKM backup files from the management station.
- **Step 7** Create a new user, for example, OkmAdm, and assign to that user all defined roles. This creates a single user who can execute all of the operations required to complete the configuration of the KMA.
- **Step 8** Disconnect from the KMA, and reconnect using the new superuser login.
- **Step 9** Create agents for the applications provided by the DR site and assign key groups to these agents as needed.
- **Step 10** Provide the agent IDs and passphrases and the IP address of a KMA accessible to the application's network to a DR site operator whose job it is to enroll the application agents with the KMA.

Once the applications are enrolled, the application can be used for restoration of the company's data.

When the restoration is complete, log in to the KMA console, and select the option to reset the KMA to factory defaults, choosing the scrub option to remove all traces of the company's Oracle Key Manager database from the DR site KMA.

Scenario 3

This scenario is a hybrid of the prior two scenarios. A copy of the company's Oracle Key Manager database exists at DR Site 1, but a DR test is being run at Site 2, which provides the same environment as the DR site in Scenario 2. Several options exist for handling this scenario, depending in part on what network connections exist between the two DR sites.

- **Option 1** Connect the drives at Site 2 directly to the management network of the KMA at Site 1.

- **Option 2** Add the KMA at Site 2 to the company's (single-node) cluster at Site 1.
- **Option 3** Back up the KMA at Site 1 and restore it to the KMA at Site 2.
- **Option 4** Transfer keys from the KMA at Site 1 to the KMA at Site 2.

OPTION 1

This option is simplest and quickest, if the application network at Site 2 can be connected to a WAN that connects to the company's KMA at Site 1. Create agents for the applications at Site 2 in the KMA at Site 1 and assign key groups as needed. (Creating the agents requires operator credentials; assigning the key groups requires compliance officer credentials.) Provide the agent IDs and passphrases and the IP address of the management port of the KMA at Site 1 to a DR site operator at Site 2 whose job it is to enroll the application's agents in the KMA. Once the enrollment is complete, the agents at Site 2 can be used to process the company's encrypted data. Configure a host with the required backup application to use the agents at Site 2 and proceed with the restoration of the company's data center. This option enables the restoration of the company's data center from Site 2, and the required keys never go into the DR site KMA.

OPTION 2

This option requires the following resources at each site.

Site 1

- A WAN connection between the KMAs at Site 1 and Site 2
- Security officer
- A threshold number of quorum members

Site 2

- A KMA, either fresh from the factory or reset to factory defaults
- Encryption applications attached to the KMA, typically the service network

At Site 1, the security officer executes the following steps:

- **Step 1** Connect a laptop to the KMA network and start the Oracle Key Manager GUI.
- **Step 2** Create a new KMA, specifying a KMA name and passphrase.
- **Step 3** Use the Service Processor interface on the KMA at Site 2 to execute the QuickStart procedure, selecting the option to add a KMA to a cluster. Supply the name and passphrase of the new KMA created in Step 2.
- **Step 4** Allow each quorum member present to enter a user ID and passphrase to complete the QuickStart procedure.
- **Step 5** Connect to the new KMA through the Oracle Key manager GUI, and check the progress of the replication of the Oracle Key Manager database to the new KMA.
- **Step 6** When the replication is complete, execute the lock/unlock KMA function.
- **Step 7** Click Unlock, and allow each quorum member present to enter a user ID and passphrase to complete the unlock operation.
- **Step 8** Create a new user, for example, OkmAdm, and assign to that user all defined roles. This creates a single user who can execute all of the operations required to complete the configuration of the KMA.

- **Step 9** Disconnect from the KMA and reconnect using the new superuser login.
- **Step 10** Create agents for the applications at Site 2, and assign key groups to these agents as needed.
- **Step 11** Provide the agent IDs and passphrases to a DR site operator at Site 2 whose job it is to enroll the applications in the KMA.

Once the applications are enrolled, the applications at Site 2 can be used to process the company's encrypted data and proceed with the restoration of the datacenter.

When the restoration is complete, delete the KMA at Site 2 from the cluster and use the ILOM interface to the KMA's host console at Site 2 to reset the KMA to factory defaults. The superuser created in Step 8 can be deleted as well.

OPTION 3

This option requires creating a backup of the company's Oracle Key Manager database at Site 1 and using that backup to create a copy of the company's cluster on the KMA at Site 2. Backup operator and security officer credentials are required at Site 1 to create the database backup, and a core security backup is required to unlock the backup key file at Site 2. Site 2 requirements are the same as those for the DR site in Scenario 2, including the required number of quorum members to complete the restore operation.

Once the backup files are created at Site 1, the process defined in Scenario 2 is used to complete the restoration of the company's data center. The restore from backup operation is slow, but effective if no quicker option is available.

OPTION 4

This option entails setting up a key transfer partner at each site. It is attractive because importing keys is very quick, much quicker than restoring keys from a backup file.

Site 2 must have a fully functioning standalone KMA with the encryption application's agents enrolled. This configuration can be done ahead of time or as part of the DR test. Setting up the KMA at Site 2 is done by a trusted operator of the company performing the DR test. Site 1 operations require security officer credentials.

The steps to be taken at each site are as follows:

Site 2

- Use the ILOM interface on the KMA at Site 2 to access the host console and complete the QuickStart procedure, using the Install First KMA in Cluster option.
- Use the Oracle Key Manager GUI to connect to the KMA using the newly created security officer login.
- Create a key policy with Import Allowed and a key group that uses this key policy.
- Obtain the public key information from the KMA at Site 1.
- Create a key transfer partner using this public key information and verify that the key fingerprint matches that at Site 1.
- Create agents for the application(s) at Site 2 and assign the import key group to them.
- Provide the agent IDs and passphrases to a DR site operator at Site 2 whose job it is to enroll the application's agent in the KMA.

Once the agents are enrolled, a host running the required application can be configured to use these agents.

Site 1

- Obtain the public key information from the KMA at Site 2.
- Create a key transfer partner using this public key information and verify that the key fingerprint matches that at Site 2.
- If necessary, modify all key policies to allow export
- Create a single superuser as described in Step 7 of Scenario 2.
- Disconnect from the KMA and reconnect using the superuser login.
- Assign all key groups to the transfer partner.
- Export keys for all data units to be used at Site 2.
- Transfer the export file containing the data units and keys to be shared to Site 2.
- Delete the key transfer partner and superuser and undo the key policy changes, if desired.

Site 2

- Load the export file received from Site 1 onto the management station.
- Import the data units and keys from the export file into the KMA database.
- Verify that a data unit exists in the KMA database for each application's data that is required to be restored.
- Remove the export file from the management station.

Restoration of the company's data center can proceed from this point at Site 2. Once the restoration is complete, log in to the console of the KMA at Site 2, and select the option to reset the KMA to factory defaults, using the scrub option to remove all traces of the company's private information.

CONCLUSION

Oracle Key Manager is a comprehensive key management platform designed to address the rapidly growing enterprise commitment to storage-based data encryption. It provides:

- Security
- Performance
- High capacity
- High availability
- Scalability
- Interoperability
- Central management
- Long-term key retention
- Ease-of-use
- Configurability
- Compliance and auditing

Customers employing Oracle Key Manager can be confident that their data is secure from unauthorized access and at the same time highly available to authorized users.

APPENDIX A: GLOSSARY

AES Advanced Encryption Standard.

agent A storage device used to encrypt and decrypt data.

ANSI American National Standards Institute.

API Application-programming interface.

BOT Beginning of tape.

data unit Abstract entity that represents a physical storage object (tape volume).

FIPS Federal Information Processing Standard.

ILOM Integrated Lights Out Manager; a dedicated system of hardware and supporting software that enables management of an Oracle server independent of the operating system.

key Data encryption key used to encode and decode data.

key group Group of keys associated with a key policy, used to enforce access to key material by encryption agents.

key ID Public identifier used to reference an encryption key.

key policy Policy that defines the lifecycle of keys in key groups associated with it.

key split credentials A set of user ID/passphrase pairs that must be provided to the system to perform certain security-critical operations.

keystore Secure location used to store encryption keys.

KMA Key Management Appliance; contains the key management database, key manager, and key store.

KMS Key Management System; a clustered group of KMAs. Rebranded in version 2.3 as the Oracle Key Manager.

MVC Multiple Virtual Cartridge.

NIST National Institute of Standards and Technology.

QuickStart A configuration menu executed automatically when a KMA is first powered on that collects the configuration data required to initialize the KMA.

quorum Key split credentials.

role A set of permissions that is granted to an Oracle Key Manager user to allow the performance of certain operations: auditor, backup operator, compliance officer, operator, or security officer.

RSA An algorithm for public key encryption.

SOAP Simple Object Access Protocol.

TLS Transport Layer Security.

VOLSER A tape volume serial number.

VOP Virtual Operator Panel interface to StorageTek tape drives.

VSM Virtual Storage Manager.

APPENDIX B: ORACLE KEY MANAGER OPERATION WITH HP AND IBM LTO TAPE DRIVE

Oracle Key Manager encryption operates somewhat differently when the agent is an HP or IBM LTO4/5+ encryption-enabled tape drive. Listed below are some behavioral differences:

Enrollment of an LTO encryption-capable drive into the OKM cluster is a multi-step process that requires the use of the VOP-LTO software designed specifically to manage the LTO drive. The first two steps are preparatory steps that are done through the Oracle Key Manager GUI by a user with operator privileges. The final two steps, the actual agent enrollment, are done using the VOP-LTO software.

- **Step 1** Create an agent for the drive in a KMA, specifying an agent ID and passphrase.
- **Step 2** Assign one or more key groups to the agent and designate one group as its default key group. (When an agent request results in the Oracle Key Manager creating a new write key, the key will be assigned to the agent's default key group.)
- **Step 3** Connect to the drive using the VOP-LTO software, and click the Configure Drive tab. In the fields provided, enter the agent ID and passphrase specified in Step 1 and the IP address of a KMA port on the drive network, and click Commit.
- **Step 4** Click the Diagnose Drive tab, and verify from the log entries that the commit operation succeeded. Return to the Configure Drive tab, and click Enroll. Again, monitor the log entries on the Diagnose Drive tab to verify the success of the enrollment operation. The Encrypt button at the top of the VOP display should now be colored blue.

The LTO drive stores at most one encryption key in its memory when it is configured to obtain keys from the Oracle Key Manager. StorageTek drives can store up to 32 keys.

The LTO drive does not prefetch keys associated with a tape volume when that volume is mounted. Instead, it reads the barcode label and a media identifier from the cartridge memory, sends this information to the Oracle Key Manager, and waits for receipt of an I/O request.

When the application issues a request to read encrypted data from the tape, the drive requests the key needed to decrypt that data. As long as the tape volume remains mounted, subsequent requests to read data written with the same key will be completed with no further requests to the Oracle Key Manager.

Write keys are handled differently. The HP LTO drive requests a key from the Oracle Key Manager on any write operation after the tape is repositioned. In a Veritas NetBackup environment, this means that if multiple backup images are to be written to the same tape volume, the drive requests a key from the Oracle Key Manager at the start of each backup job. If the encryption key last used to write data to the tape volume is still in the Protect-and-Process state, Oracle Key Manager resends that key to the drive. Otherwise, Oracle Key Manager creates a new key, associates it with the data unit, and sends it to the drive.

A description of how IBM LTO drives handle write keys appears in the "LTO4 Differences" section of the IBM LTO Tech Brief.

StorageTek drives use the Protect-and-Process key acquired when the tape volume is mounted to encrypt data as long as the tape volume is mounted. This reduces the key fetch overhead when writing multiple backup jobs to one tape volume.

ORACLE CORPORATION

Worldwide Headquarters
500 Oracle Parkway, Redwood Shores, CA 94065 USA

Worldwide Inquiries
TELE + 1.650.506.7000 + 1.800.ORACLE1
FAX + 1.650.506.7200
oracle.com

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com. Outside North America, find your local office at oracle.com/contact.

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

Integrated Cloud Applications & Platform Services

White Paper Title
January 2017
Author: [OPTIONAL]
Contributing Authors: [OPTIONAL]