An Oracle White Paper
November 2010

Oracle Key Manager Version 2.x
Security and Authentication White Paper

Audience

Anyone who is interested in learning more about the Security and Authentication aspects of the Oracle Key Manager (OKM 2.X.). Intended audiences are those who are already familiar with the information contained within the systems assurance and installation guide.

Related Publications

The following publications provide additional information about specific topics relating to the use and security of the Oracle Key Manager (OKM.)

DESCRIPTION	LINK OR PART NUMBER
StorageTek T10000A Tape Drive FIPS 140-2 Security Policy	http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1157.pdf
StorageTek T10000B Tape Drive FIPS 140-2 Security Policy	http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1156.pdf
StorageTek T9840D Tape Drive FIPS 140-2 Security Policy	http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2010.htm#1288
Sun Crypto Accelerator 6000 FIPS Security Policy	http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1050.pdf
Oracle Key Manager Technical Documentation: Systems Assurance Guide, Administrative Guide, Best Practices, Management Practices and Open Systems Implementation Practices	http://download.oracle.com/docs/cd/E19540-01/
Sun System Administration Guide: Security Services	http://download-llnw.oracle.com/docs/cd/E18752_01/html/816-4557/scf-1.html

Scope

This document provides an overview of the Security and Authentication aspects of the Oracle Key Manager Version 2.x (OKM 2.x), formerly known as Oracle's StorageTek Crypto Key Management System 2.x (KMS).

The Oracle Key Manager 2.X describes the family name for the product. The Oracle Key Manager is architected to provide highly secure and automated key management services to generic encryption agents, such as encrypting tape drives supported in Oracle Automation and Library products.

This document does not cover the details of installation and operation of the OKM since these are fully described in the documents listed in the table above.

Overview

The Oracle Key Manager consists of:

- **Key Management Appliance** – a Sun server loaded with the Key Management Appliance (KMA) software. One or more of these appliances are required for the encryption solution.
- **Cluster** – the full set of KMAs in the system. All of the KMAs are aware of each other, and asynchronously replicate information to each other. When the OKM documentation refers to the "Cluster", it means the common, collective information held by all KMAs in the cluster.
- **Agent** – A device or software that performs encryption, using keys managed by the Cluster. For the Oracle Key Manager, these are the supported encrypting tape drives. Agents are clients of the OKM and communicate with KMAs using the agent protocol.
- **Oracle Key Manager GUI** – A software component that provides a management graphical user interface (GUI). The Oracle Key Manager incorporates the management APIand uses this API to communicate with the KMAs in the Cluster. The Oracle Key Manager GUI must be installed on a customer-provided platform running Windows or Solaris. In addition to the system components, there are the external "actors" that interact with the system. These are described below.
- **Oracle Key Manager CLI** – A command line interface (CLI) utility supporting a subset of the same functions as the Oracle Key Manager GUI. The CLI allows automation of various tasks such as backup, key export or audit reporting.

The following sections present some of the important system use cases and a description of the system actors.

Note: The term "actors" covers both human and hardware or software modules that interact with OKM.

Actors

User

Users are persons who have a userid and passphrase, or a valid certificate, to authenticate with the Cluster. Users can interact with OKM through the OKM Console, the Oracle Key Manager GUI or the OKM CLI.

A user must be assigned one or more roles. Each role can perform a subset of the use cases described here (RBAC). The roles are:

Security Officer

The security officer role allows management of the Cluster's security settings, users, sites and transfer partners.

Compliance Officer

The compliance officer role manages key policies and key groups and determines which agents and transfer partners can use which key groups.

Operator

The operator role manages agents, data units and keys.

Backup Operator

The backup operator role performs backups.

Auditor

The auditor role can view information about the Cluster.

Agent

Strictly speaking, the agent is considered part of the system. For the encryption agent use cases, however, it is useful to consider the agent as an actor acting on the Cluster.

NTP Server

This is a server outside the system which uses NTP protocol to control the system time. Configured with the time use cases.

Technical Support

Technical Support is a qualified service person (QSP) who can connect directly into a KMA using ssh.

SNMP Manager

SNMP Managers may be configured as INFORM destinations for the Cluster.

Agent Operator

The Agent Operator is a person or software that can interact with an agent to complete the encryption agent use cases.

Authentication

The OKM architecture provides for mutual authentication between all elements of the system: KMA to KMA, agent to KMA and the Oracle Key Manager GUI, or CLI, to KMA for user operations.

In simple terms, enrollment of each element of the system (for example, a new encryption agent) is accomplished by creating an ID and a passphrase in the OKM that is then entered into the element to be added. For example, when a new drive (encryption agent) is added to the system, the agent and KMA automatically run a challenge/response protocol based on the shared passphrase that results in the agent obtaining the Root CA certificate and a new key pair and signed certificate for the agent. With the Root CA, agent certificate, and key pair in place, the agent can run the TLS protocol for all subsequent communications. All certificates are X.509 certificates.

The OKM behaves as a root certificate authority (CA) to generate a root certificate that is used in turn to derive (self-sign) the certificates used by agents, users and new KMAs. Full Public Key Infrastructure (PKI) is not implemented.

Generation of Agent Certificates:

- The Issuer Common Name (CN) is always set to 'RootCA'
- The KMA generates a Serial Number based on a concatenation of the 64-bit KMA ID and 64-bit sequential counter from the OKM DB.
- The Subject Common Name is the text name given to the Agent. In the following example, the certificate is issued to 'MyAgent'
- The agent stores this certificate, along with the corresponding private key, in a file called "clientkey.pem"

Note: The OKM does not support external Certificate Authorities.

Specific Enrollment Procedures

Agent to OKM

When a new tape drive (agent) is to be enrolled in OKM, a user with the role of Operator uses the GUI to enter the following parameters:

- Agent ID – a value that uniquely defines the Agent, between 1 to 64 characters
- Agent Description (optional) - a value that describes the Agent, between 1 to 64 characters
- Site ID (optional) – a click-down entry that defines the site location of the agent
- Passphrase (enter and confirm) – the OKM ensures that the selected passphrase meets the requirements for passphrase strength
- Minimum value 8 characters, maximum value 64 characters
- Must contain 3 of the four character classes: upper case, lower case, numeric or special characters

- The following special characters are allowed:
- ~ ! @ # $ % ^ & * ( ) - _ = + [{ }] ; : ' “ , . / ?
- Control characters including tabs and linefeeds are not allowed

The Agent ID and Passphrase must then be entered into the new drive using the StorageTek Virtual Operator Panel (VOP) tool along with the IP address of a KMA in the cluster.

Adding a New KMA to an Existing Cluster

When a Key Management Appliance is added to an existing cluster, additional safeguards are in place due to the sensitivity of this operation.

The user in the role of Security Officer uses the GUI to first create a KMA and then subsequently uses the QuickStart program to initialize the KMA and then join it to an existing cluster.

Initial authentication of the new KMA is accomplished by the challenge/response between the new KMA and an existing KMA in the cluster based on the secret passphrase shared as described below.

Create a KMA

The KMA platform will have already been assigned an IP address and the Security Officer will assign parameters to the KMA as follows

- KMA Name – a value that uniquely defines the KMA, between 1 to 64 characters
- KMA Description (optional) - a value that uniquely describes the KMA, between 1 to 64 characters
- Site ID (optional) – a click-down entry that defines the site location of the KMA
- Passphrase (enter and confirm) – the KMS ensures that the selected passphrase meets the requirements for passphrase strength

Initialize the KMA Using the QuickStart Program

- The Security officer enters the KMA name that must match the value provided
- See the *Administration Guide* for details of this process and other parameters that must be entered

Join the KMA to an Existing Cluster Using the QuickStart Program

- The Security Officer will select Join Existing Cluster
- Enters the IP Address or Host Name of an existing KMA in the cluster
- Enters the Passphrase for the new KMA assigned above
- At this point, a screen is presented for the requisite number of Quorum Members to to enter their credentials (see description later)
- On completion, the new KMA is now known to the cluster but is in a Locked Condition, communicating with other KMAs but is locked and will not distribute keys. The OKM Manager GUI is used to unlock the new KMA and complete the process

- The data and time on the new KMA set to the date and time from the specified existing KMA.
- The OKM Cluster will begin propagating all information to the newly added KMA. This will cause the new KMA to be busy until it is caught up with the existing KMAs. Existing KMAs will also be busy propagating changes to the new KMA.
- Once the replication lag for the new KMA drops to a similar value to other KMAs in the cluster, preferably near zero, the KMA should be manually unlocked. Replication lag is a heuristic that approximates the number of database updates needed to synchronize the KMA with the rest of the cluster. The Unlock operation is initiated by a Security Officer and validated by the Quorum.

User to OKM

The Oracle Key Manager GUI, or CLI, runs on a customer-supplied platform and users log in to the OKM cluster with a User ID and Passphrase created by a Security Officer and shared individually with the User. For the CLI, the authentication can also be accomplished using the user's X.509 certificate in order to avoid writing automation scripts with plaintext passphrases embedded.

To create a new user, an existing user with a Security Officer role uses the GUI to enter the following parameters:

- User ID – a value that uniquely defines the User, between 1 to 64 characters
- Description - a value that describes the User, between 1 to 64 characters
- Role: check boxes allow the selection of one or more pre-defined roles: Security Officer, Backup Operator, Compliance Officer, Operator, Auditor, Quorum Member.
- Passphrase (enter and confirm) – the cluster ensures that the selected passphrase meets the requirements for passphrase strength

When a User with defined credentials uses the GUI to log into the OKM, he or she enters the assigned ID and Passphrase which the OKM uses to conduct a challenge/response protocol based on this input and the user credentials stored in the OKM database. A successful challenge/response interaction results in the KMA sending a certificate to the system running the GUI to authenticate the specific Session connectivity. The user can subsequently choose to save this certificate to a PKCS#12 password protected file for use with the CLI.

Role of the SCA 6000 Card

The fundamental security of the Oracle Key Manager Appliance (KMA) is assured by the FIPS 140-2 Security Level 3 SCA 6000 Cryptographic Accelerator Card (Certificate 1050) with the cryptographic boundary defined as the connector to the SCA 6000 card. When the OKM is operated in FIPS compliant mode, keys do not leave the cryptographic boundary of the SCA 6000 card in unwrapped form. Beginning with version 2.1, the OKM always configures the SCA 6000 card for use in FIPS mode.

Given the constraints of using a standard server platform with its limitations on providing physical security, it was decided that little advantage would be obtained by pursuing FIPS 140-2 validation for the KMA based on a cryptographic boundary defined as the external covers of the server. Beginning with version 2.1, the OKM has been released with the necessary features required for FIPS 140-2 certification for the encryption agents supported by OKM.

Beginning with OKM version 2.3 the SCA 6000, also known as the hardware security module (HSM), is now an optional component. When a KMA is configured without an SCA 6000 card, cryptography is performed using the Solaris Cryptographic Framework (SCF) PKCS#11 soft token. When the SCA 6000 card is present, the KMA utilizes it through the PKCS#11 hardware token configured into the Solaris Cryptographic Framework (SCF).

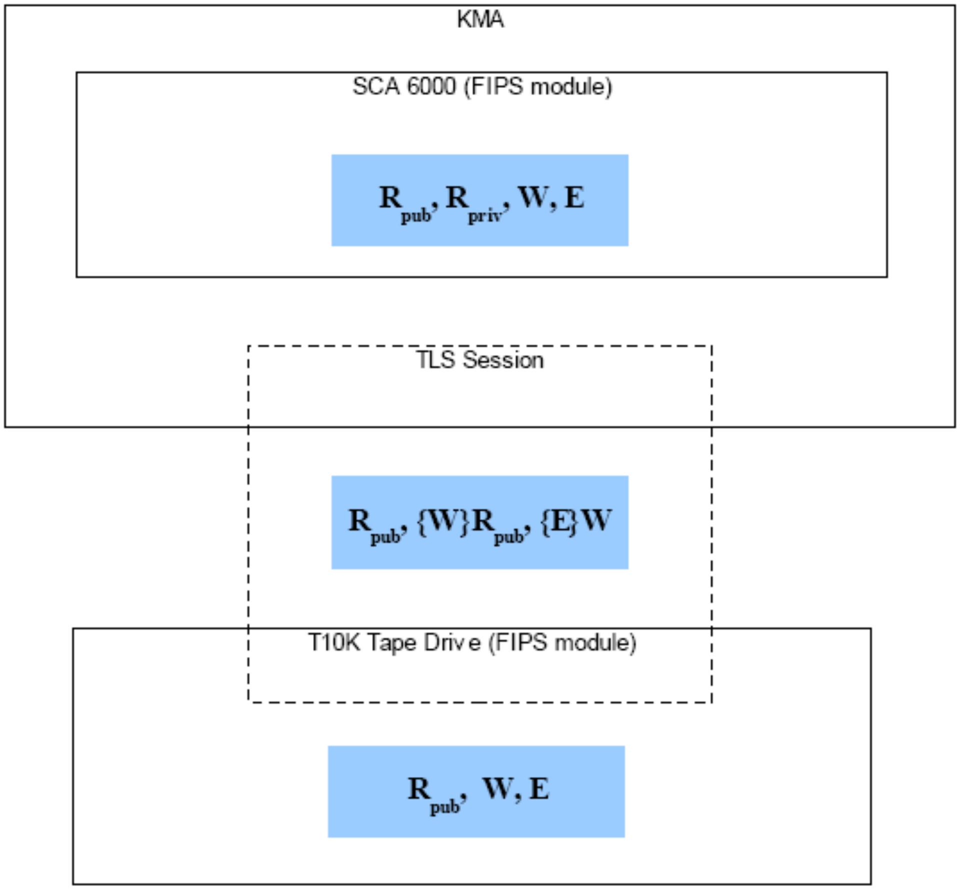
The SCA 6000 card uses the FIPS approved RNG specified in FIPS 186-2 DSA RNG using SHA-1 for generation of cryptographic keys. A non-approved hardware RNG is used for providing seeding material.

Earlier versions of the key management system, 2.0 thru 2.0.2 allowed the Solaris Cryptographic Framework (SCF) to fail-over to software cryptography if the SCA 6000 card was non-functional for any reason. This was accomplished through use of the SCF metaslot.

KMS Version 2.1, which provided support for the FIPS 140-2 validated versions of the tape drives. This did not allow this software fail-over and thus introduced a single point of failure in the KMA.

KMS version 2.2, and OKM version 2.3 plus future versions allow the system to operate with, or without, a functioning SCA 6000 card. This is the default mode and detects the absence, or failure of an SCA 6000 card. Without a functioning SCA 6000 card the hardware security module status is set and displayed in the Oracle Key Manager GUI indicating that cryptography is being performed in software. Fail-over to software cryptography is not allowed when the system is set in the FIPS mode. In FIPS mode, when an SCA 6000 card fails, the tape drive's requiring keys will receive an error and then fail-over to another KMA within the cluster.

Key Transmission



- Keys are transmitted from the KMA to encryption agents within the TLS channel and in wrapped form.
- Keys are generated and managed in KMA within SCA 6000 FIPS cryptographic boundary
- Keys are transported to the cryptographic boundary of the encrypting device (agent) on demand

Encryption key wrapping key transport

- Agent initiates by requesting public key from KMA
- RSA key pair $\mathbf{R_{pub}}$, $\mathbf{R_{priv}}$ generated in SCA 6000 by KMA
- $\mathbf{R_{pub}}$ exported from SCA and sent to Agent by KMA
- Agent generates and stores AES key, **W**, to be used as wrapping key for encryption keys
- $\mathbf{\{W\}R_{pub}}$ (RSA PKCS #1 V1.5) returned to KMA and stored associated with particular Agent

Encryption key transport

- Agent initiates by requesting tape drive encryption key
- KMA generates the AES encryption key E (used to encrypt user data) in SCA 6000
- KMA loads {W}Rpub into SCA 6000 and unwraps W with Rpriv
- E is wrapped with W in the SCA 6000 and {E}W is sent to the agent
- E is unwrapped with W in the agent

Notes:

- For KMS 2.0 and 2.0.2, wrapping of keys transmitted to the agent was provided by AES-256 CBC with HMAC Authentication
- KMS 2.1 and subsequent versions additionally support AES Keywrap to meet the latest NIST requirements; AES Keywrap is used by all supported agents.

OKM Replication

Each KMA in the cluster contains a replicated version of the entire OKM Database. Replication is a real-time process, transparent to the user.

The replication procedure for each KMA is executed when each Appliance starts. This procedure ensures that all peers have the Appliance’s last-known time-stamp vector and that the Appliance has a matrix of its peers’ last-known time-stamp vectors. The Appliance may also discover some new cluster members that it was unaware of previously.

Appliances exchange their peers' last-known time-stamp vectors on a request received from a peer Appliance. The Appliance updates its knowledge of the peer’s state, returns its time-stamp state to the peer, helps the peer discover any cluster members the peer is unaware of, and initiates a pull of any non-replicated peer updates.

Updates are integrated into the local KMA transaction processing sequence. It ensures that unique timestamps are assigned to new or updated records and that the replication log is updated. After the transaction is committed, a push procedure is triggered to immediately propagate the updates to peer Appliances.

A “pull” anti-entropy replication strategy ensures that all updates eventually propagate to all servers, even in the case of failures. In this strategy, each Appliance triggers this procedure at 60-second intervals to pull updates from a randomly selected peer. When a KMA receives such a PullReplication message, it triggers a procedure that will send the requesting peer any local updates with timestamps less than the last-known timestamps in the peer’s

All replication traffic is conducted over a TLS 1.0 channel with protection afforded by the cypher suite TLS_RSA_WITH_AES_256_CBC_SHA.

Master Key Functionality

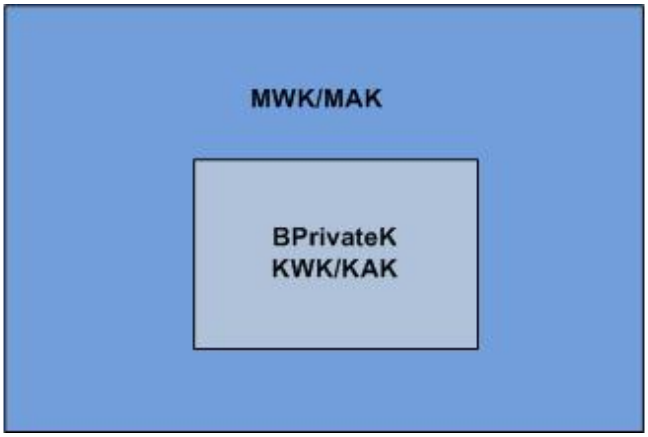
When a new OKM cluster is created, an AES-256 Master Key is automatically generated. This Master Key is then split into shares using the Shamir Shared Secret Algorithm applied to the Quorum.

During the setup operation, a Quorum is defined with each member of the Quorum having an individual ID and passphrase. The passphrases are then used are used to create keys that encrypt the shares created by the Shamir Shared Secret Algorithm. Neither the actual values for the passphrases nor the Master Key Value itself are stored in the system but the passphrases when entered by the Quorum unwrap the encrypted shares and regenerate the Master Key.

It should be noted that the Master Key has two components, the Master Wrapping Key (MWK) and the Master Authentication Key (MAK) but for brevity, we will refer only to the Master Wrapping Key.

At the same time, an AES-256 Key Wrapping Key (KWK) and Key Authentication Key (KAK) are created, along with a RSA 2048 private/public Key Pair, the Backup Public/Private keys.

This set of keys (KWK, BprivateK) only change if the Quorum is changed and are stored, wrapped by the MWK as the Core Security Backup.



Creating the Core Security Backup is a mandatory function when a new cluster is created and is performed by a Security Officer who specifies the location and name for the file.

OKM Backup

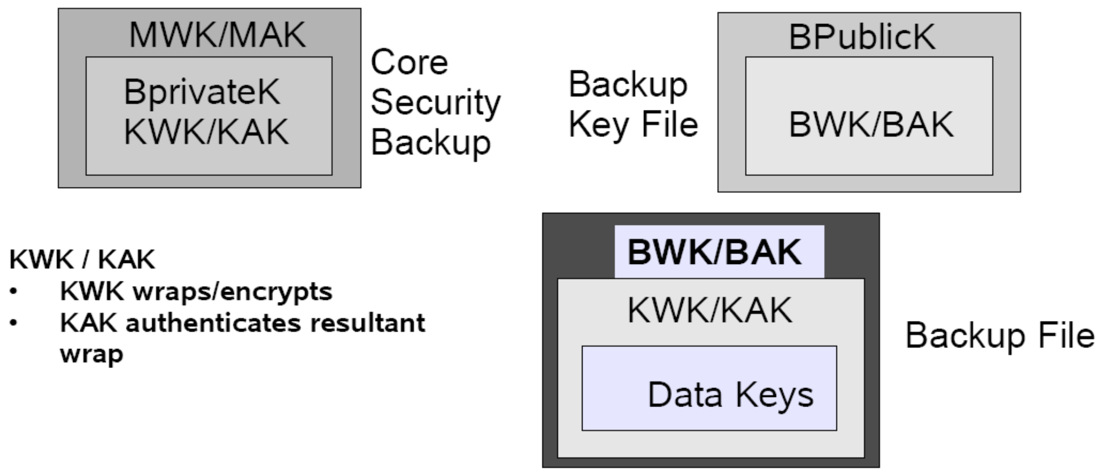
Backup of the OKM database is a function executed from the Oracle Key Manager GUI, or CLI, by a user in the role of a Backup Operator.

The Backup Operator is prompted to provide a file name (including location) for the Backup File itself and a separate file the Backup Key File. These files can be stored in any location accessible to the platform on which the GUI, or CLI, is being run.

Each time a Backup is created, an additional pair of AES-256 keys are created: Backup Wrapping Key (BWK) and Backup Authentication Key (BAK). These 2 keys are encrypted with the Backup Public Key and stored within the Backup Key File (XML) in the location specified by the Backup Operator.

The OKM database is then encrypted by the KMA. The Backup Operator is logged into using two layers of encryption (first the KWK/KAK and then the BWK/BAK) and the backup is stored in the specified location.

This provides highly robust security for the Key Database and allows it to be stored in an otherwise unprotected location.



Restore OKM Database

The initial step of the restore procedure is to

- Perform a QuickStart of the KMA in order to configure networking,
- Create the Security Officer user and
- Specify that the KMA will be restored from backup.

The Security Officer must then use the Oracle Key Manager GUI to login, initiate the restore procedure to browse for and enter the location of the three files involved (Core Security Backup, Key File Backup and Key Backup.)

The Quorum screen will then be displayed and the requisite number of Quorum members enter their ID and passphrase.

- The passphrase is used to generate the key and unwrap part of the shared secret Master Key
- The shares combine to recreate the master key (MWK/MAK)
- The master key unwraps the BprivateK and the Key Wrapping Key (KWK, KAK)
- The BprivateK unwraps the Backup Wrapping Key (BWK, BAK)
- The Backup Wrapping Key unwraps the Key Package encrypted with the Key Wrapping Key
- Finally the Key Wrapping Key unwraps the encrypted database and allows it to be downloaded into the KMA.

Note: As part of the Restore Operation, the pre-existing database for the entire OKM is reset and overwritten.

Key Transfers

Keys can be transferred from one cluster to another using the Key Transfer Process.

This is a two-stage process. The Transfer Partner relationship must be established in advance and is set up by users in the Security Officer Role at each party. Each party must create a Public/Private Key pair and transmit the value of the Public Key to the other. Using the GUI, the Security Officer must then enter the following information:

- Transfer Partner ID
- Transfer Partner Description
- Contact Information – this field describes how keys are to be transmitted to the partner – for example by email or by exchange of physical media
- Enabled – if the box is checked, the transfer partner can share keys with another partner.
- Allow Export To – if checked, this allows keys to be sent to the partner
- Allow Import From – if checked, this allows keys to be imported from the partner.

The next GUI Tab allows the Security Officer to enter the public key data received from the partner.

- New Public Key ID
- New Public Key Value
- New Public Key Fingerprint – this shows a hash value created from the Public Key value and allows verification that the key value has not been tampered with during transmission.

When the required information has been input, the GUI will prompt for a quorum to validate the operation.

Note that the Transfer Partner Relationship can be bi-directional or uni-directional depending on the Export/Import options selected.

When the Transfer Partnership set-up is complete, users in the Operator Role conduct the actual transfer of Keys.

Using the GUI or CLI, the Operator selects Data Units (and associated keys) to be exported, typically based on the VOLSER property of a data unit, that is, the data unit's external tag. All keys that are "In Use" on the tape defined by that VOLSER can be selected for Export from the Data Unit List panel of the GUI and from the CLI. The Operator specifies a destination transfer partner from the list and a file is created with the required data unit information and wrapped keys. The file is compressed and encrypted using the Public Key received from the Transfer Partner and signed with the Private Key form the "home" KMA. The file may then be safely transmitted to the destination transfer partner by email, etc.

Using the GUI, an Operator at the receiving partner then accesses the Import Keys function in the Transfer Partner menu, selects the Transfer Partner and Key Transfer File Name and defines the Key Group into which the Keys should be imported. An Operator at the receiving partner can also specify this information to import keys using the CLI. The Key Transfer Partner file is then decrypted using the receiving partner's Private Key and validated using the transmitting partner's Public Key.

Autonomous Unlock

The OKM offers the convenience option of Autonomous Unlock for each KMA in lieu of the Quorum. If Autonomous Unlock is enabled, the KMA will automatically unlock the OKM database and be ready for operation without requiring a minimal Quorum . With this option enabled, an attacker may be able to locate the master key and gain access to key material.

If Autonomous Unlock is disabled, when the KMA is power-up, Quorum intervention is required for the KMA to become operational and be capable of providing keys. Until the Quorum requirements are satisfied, all keys in the KMA are fully protected and a cryptographic attack capable of breaking multiple defenses, plus AES-256, is needed to access keys.

Toggling between Enabled/Disabled for Autonomous Unlock is a Security Officer function. Changing the state from Enabled to Disabled must be validated by the Quorum.

A KMA in the locked state is not able to unwrap the Master Key Material and thus is unable to access Data Unit Keys. As a result, the KMA is unable to service Agent requests to register new Data Units or to retrieve Data Unit Keys for existing Data Units.

It should be noted that certain functions can still be performed on a Locked KMA such as creating a Backup since the Backup File and the Backup Key File are both cryptographically protected.

Oracle is committed to developing practices and products that help protect the environment

Oracle Key Manager
Nov 2010

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

SOFTWARE. HARDWARE. COMPLETE.