Key Management Service (KMS) Integration on Cloud Platforms for 10.1.0

Created on: Feb 5, 2025

Table of Contents

Copyright

Chapter 1

Introduction

This user guide provides information for integrating the Key Management Service (KMS) on the cloud platforms, such as Amazon Web Services (AWS), Azure, and Google Cloud Platform (GCP) with the Protegrity Enterprise Security Administrator (ESA) appliance.

Table 1-1: KMS on Cloud Platforms

Cloud Platforms	Description
AWS	AWS KMS is an encryption and key management service which contains the AWS KMS keys and functionality to perform the AWS KMS operations. The KMS keys and functionality are used along with other AWS services to protect the data in AWS applications.
Azure	Azure Managed HSM, a part of the Azure Key Vault offering, enables you to safeguard cryptographic keys, and perform cryptographic operations.
GCP	GCP KMS is a cloud-based key management system which is used to perform the cryptographic operations to protect the data stored in the cloud or on-premise. Using the Cloud KMS, you can create, import, and manage the cryptographic keys.

This guide may also be used with the *Protegrity Key Management Guide*, which is intended to provide a general overview of the Key Management and explain its importance and impact on the Protegrity products.

1.1 Sections Contained in this Guide

The guide is broadly divided into the following sections.

- Section 1 *Introduction* defines the purpose and scope for this guide. In addition, it explains how information is organized in this guide.
- Section 2 *Support Matrix for the Cloud Platforms* describes the support matrix for the ESA appliance and the Google Cloud Platform (GCP) Key Store.
- Section 3 *Configuring the ESA with GCP Key Management* describes the steps to configure the ESA appliance to communicate with Google Cloud Platform (GCP) Key Store.
- Section 4 *Switching from Protegrity Soft HSM to GCP Key Store* provides information about switching between the Protegrity Soft HSM and the HSM client and Google Cloud Platform (GCP) Key Store.

Chapter 2

Support Matrix for the Cloud Platforms

The following table provides the compatibility information of the cloud platforms, such as Amazon Web Services (AWS), Azure, and Google Cloud Platform (GCP) with the Protegrity Enterprise Security Administrator (ESA) appliance.

Table 2-1: Support Matrix of the Cloud Platforms with the ESA

Cloud Platforms	Supported version
AWS	10.0.1 onwards **Note:** The AWS support is available with the additional patch ESA_PAP-ALL-64_x86-64_10.0.1.2443 on the ESA v10.0.1.
Azure	10.1.0 onwards **Note:** The Azure support is available with the additional patch ESA_PAP-ALL-64_x86-64_10.1.0.P.2467 on the ESA v10.1.0.
GCP	9.1.0.0 onwards **Note:** The GCP support is available with the additional patch ESA_PAP-ALL-64_x86-64_9.1.0.0.2156.FE-1 on the ESA v9.1.0.0.

Chapter 3

Configuring the ESA with AWS Key Management

This section provides information about configuring the ESA appliance with the AWS Key Store.

3.1 Prerequisites

Ensure that the following prerequisites are met before configuring the ESA with the AWS Key Store.

3.1.1 Authorization

The Amazon Web Services Key Management Service (AWS KMS) allows you to enable the creation of the data encryption key (DEK). Additionally, you can also encrypt and decrypt the data, and generate random bytes of data using the Key Management Gateway (KMGW) in the ESA.

To use the AWS KMS as a key store, the following permissions are required by the AWS user or role:

Table 3-1: Permissions for accessing AWS KMS user or role

Actions	Permissions	Description
Decrypt	kms:Decrypt	Enable decryption using a key
Encrypt	kms:Encrypt	Enable encryption using a key
TagResource	kms:TagResource	Enable the possibility to tag a resource
GenerateRandom	kms:GenerateRandom	Grant permission to generate random bytes
DescribeKey	kms:DescribeKey	Grant permission to view information about a key
CreateKey	kms:CreateKey	Grant permission to create a new key
List MasterKeys	tag:GetResources	List all the masterkeys

3.1.2 Keys in AWS KMS

The AWS KMS keys are tagged with the following two tags:

- **Owner**: Protegrity
- **Service**: KMGW

These tags are used to search or filter keys which are created by the KMGW.

3.1.3 Authentication

- **On AWS -**

  If the ESA EC2 instance is running on the AWS and an IAM role is setup, then the IAM role gets linked to the ESA EC2 instance. Further, the SDK automatically gets the credentials to perform authenticated calls to the AWS.

- **On Prem -**

  If the ESA is running on any other environment, then ensure to set up the environment by generating the long-term credentials on the AWS. The following environment variables are used to set the long-term credentials:

  Table 3-2: List of Environments Variables to set the long-term credentials

Environment	Values	Description
AWS_REGION*	us-east-1	The AWS region to use
AWS_ACCESS_KEY_ID*	AKI...	AWS Access key ID (long-term credentials)
AWS_SECRET_ACCESS_KEY*	wJalrXUt...CYEXAMPLEKEY	AWS Secret access key (long-term credentials)

> **Note:** Ensure that the environments marked with * are set when they are not running on the EC2 instance.

3.2 Configuring the External Key Store with the AWS Customer Managed Keys

The external key store with the AWS customer managed keys is configured by using one of the following methods:

- Using the roles which are set in the AWS KMS environment.
- Setting the environments (AWS_SECRET_KEY, AWS_ACCESS_KEY).

3.2.1 Configuring the AWS Key Store by using the Authorized IAM Role

Before you begin
Ensure that an ESA instance is created in the AWS.

➤ The following procedure provides the steps to configure the AWS Key Store by using the authorized IAM role:

1. In the AWS screen, navigate to **AWS** > **EC2**.
2. Select the required instance.
3. Navigate to **Actions** > **Security** > **Modify IAM**.
4. Select the role with the AWS KMS permissions.
   The IAM role is modified for the instance.
5. Update the environment variables for the AWS specification in the *keystore.env* config file at `opt/protegrity/keystore/external` directory.

   ```
   export PTY_KEYSTORE=AWS
   ```

   The Key Store is changed to the AWS in the configuration file.
6. Restart the KMGW service from the ESA Web UI.

7. In the ESA Web UI, navigate to **Policy Management** > **Key Stores** > **Key Store** to verify the AWS Key Store information.

```
Information
Manufacturer    Amazon
Description     AWS Key Management Service (KMS) API
Version         1.26.2
General
State           Inactive

Amazon Web Service (AWS) KMS details
Region          <region name>
Key             <tag name>
Service         <service name>
Owner           <owner name>
```

8. Click the **Test** button.
   The *Connection passed*, *Authentication passed*, and *Random passed* message appears.
9. Click the **Set as Active** button.
   The Key Store is switched successfully.

> **Note:** You must verify that the master key is rotated successfully in the AWS Key Store.

3.2.2 Configuring the AWS Key Store by using the long-term Variables

The following procedure provides the steps to configure the AWS Key Store by using the long-term variables:

1. Update the environment variables for the AWS specification in the *keystore.env* config file at the `opt/protegrity/keystore/external` directory.
   The following snippet displays the sample settings for the environment variables.

```
export PTY_KEYSTORE=AWS
export AWS_REGION=<region>
export AWS_ACCESS_KEY_ID=<access-key-id>
export AWS_SECRET_ACCESS_KEY=<secret-access-key>
```

2. On the ESA Web UI, navigate to **Key Management** > **Services** to restart the Key Management Gateway service.
   The **Key Management Gateway** service is restarted successfully.
3. Navigate to **Key Management** > **Key Stores** > **Key Store** to verify the AWS Key Store information.

```
Information
Manufacturer    Amazon
Description     AWS Key Management Service (KMS) API
Version         1.26.2
General
State           Inactive

Amazon Web Service (AWS) KMS details
Region          <region name>
Key             <tag name>
Service         <service name>
Owner           <owner name>
```

4. Click the **Test** button.
   The *Test is successful* message appears.
5. Click the **Set as Active** button.
   The AWS Key Store is set as active.

> **Note:** You must verify that the master key is generated and rotated in the AWS Key Store.

Chapter 4

Configuring the ESA with Azure Key Management

This section provides information about configuring the ESA appliance with Azure.

4.1 Prerequisites

Ensure that the following prerequisites are met before configuring the ESA with the Azure Key Store.

4.1.1 Configuring the Managed HSM

Protegrity supports the Managed HSM Key Vault type due to the presence of "Get Random Bytes" functionality which is not available in the standard Key Vault.

Ensure that the Azure Managed HSM is already set up and activated on your system.

For more information about setting up an Azure Managed HSM, refer to *https://learn.microsoft.com/en-us/azure/key-vault/managed-hsm/quick-create-cli*.

4.2 Configuring Azure in Key Management Gateway

This section provides information about configuring Azure in Key Management Gateway.

4.2.1 Authentication Components for Azure

Service Principal:

When you register a new application in Microsoft Entra ID, a Service Principal is automatically created. This Service Principal acts as the application identity within the Microsoft Entra tenant and controls access to resources based on the roles assigned to it.

Ensure that the Service Principal is created with access to the Key Vault.

For more information about Service Principal using the CLI, refer *https://learn.microsoft.com/en-us/cli/azure/azure-cli-sp-tutorial-1?tabs=bash*.

For more information about Service Principal using the Web UI, refer *https://learn.microsoft.com/en-us/entra/identity-platform/howto-create-service-principal-portal*.

System Managed Identity:

Some Azure resources, such as virtual machines, allow you to enable a System Managed Identity directly on the resource. The System Managed Identity exists only as long as the underlying resource remains active. The name of the System Managed Identity is the same as the Azure resource it's created for.

4.2.2 Azure Configuration

There are two ways to configure Azure in the Key Management Gateway (KMGW), depending on where the ESA is running.

- If the ESA is running On-Prem.
- If the ESA is running on the Azure cloud platform.
- If the ESA is running on any other cloud platform, other than Azure.

To enable an Azure Managed HSM as the Key Store backend, some changes are required in the "keystore.env" located at -

```
/opt/protegrity/keystore/external/keystore.env
```

Configuring Environments in KMGW:

The KMGW can be configured to authenticate against Azure Key Vault either by using a Service Principal or System Managed Identity.

Table 4-1: Environments in KMGW

Environments	Description	Service Principal	System Managed Identity*
PTY_AZURE_VAULT_URI	The name of the Key Vault. For example: https://example.managedhsm.azure.net.	Required	Required
PTY_KEYSTORE	The selected Key Store. It should be set to "Azure".	Required	Required
AZURE_CLIENT_ID	The ID for the Service Principal.	Required	Not required
AZURE_CLIENT_SECRET	The secret for the Service Principal.	Required	Not required
AZURE_SUBSCRIPTION_ID	The subscription ID where the Key Vault is present.	Required	Not required
AZURE_TENANT_ID	The Tenant ID where the Key Vault is present.	Required	Not required

Note: System Managed Identity* - It is only present when the ESA is running on an Azure VM instance.

4.3 Setting Up Azure in Key Management Gateway

This section provides information about setting up Azure in Key Management Gateway.

4.3.1 Setting Up the Service Principal

Before you begin
Ensure that the following prerequisites are met.

- Since the ESA is not running on Azure, it requires a Service Principal that performs the role of a user.

- The environment variables, such as, "PTY_AZURE_VAULT_URI", "AZURE_CLIENT_ID", "AZURE_CLIENT_SECRET", "AZURE_SUBSCRIPTION_ID", "AZURE_TENANT_ID", and "PTY_KEYSTORE" are set.

  For more information about the environment variables, refer to **Configuring Environments in KMGW** in *Azure Configuration*.

To configure the Azure Key Store by using the Service Principal:

1. Set up a Service Principal that performs the role of a user.
2. When you create the Service Principal, the "AZURE_CLIENT_SECRET" and "AZURE_CLIENT_ID" values are generated.
3. The Service Principal is added to the **Local RBAC** of the Azure Managed HSM. The Service Principal is assigned the "Managed HSM Crypto User" role.

   If you have the Azure CLI installed, then you can perform this operation with help of the following command.

   ```
   az keyvault role assignment create --hsm-name [NAME of key vault] --role  "Managed HSM
   Crypto User" --assignee [ID of service principal] --scope /
   ```

4. Update the environment variables for the Azure specification in the *keystore.env* config file at the `opt/protegrity/keystore/external` directory.

   The following snippet displays the sample settings for the environment variables.

   ```
   export PTY_AZURE_VAULT_URI=https://example.managedhsm.azure.net
   export PTY_KEYSTORE=Azure
   export AZURE_CLIENT_ID=<client-id>
   export AZURE_CLIENT_SECRET=<client-secret>
   export AZURE_SUBSCRIPTION_ID=<subscription-id>
   export AZURE_TENANT_ID=<tenant-id>
   ```

5. On the ESA Web UI, navigate to **Key Management** > **Services** to restart the Key Management Gateway service.

   The **Key Management Gateway** service is restarted successfully.

6. Navigate to **Key Management** > **Key Stores** > **Key Store** to verify the Azure Key Store information.

   ```
   Information
   Manufacturer    Microsoft
   Description     Azure Key Vault
   Version         1.1.0

   General
   State           Inactive

   Azure Managed HSM Key Vault details
   Vault URI       https://example.managedhsm.azure.net
   ```

7. Click the **Test** button.
   The *Test is successful* message appears.
8. Click the **Set as Active** button.

   The Azure Key Store is set as active.

   > **Note:** You must verify that the master key is generated and rotated in the Azure Key Store.

4.3.2 Setting Up the System Managed Identity

Before you begin
Ensure that the following prerequisites are met.

- The ESA is running on an Azure Virtual Machine (VM) instance.

- The VM should have the System Managed Identity enabled.

  For more information about enabling the system managed identity, refer *https://learn.microsoft.com/en-us/entra/identity/managed-identities-azure-resources/how-to-configure-managed-identities?pivots=qs-configure-portal-windows-vm#enable-system-assigned-managed-identity-during-creation-of-a-vm*.

- The environment variables, such as, "PTY_KEYSTORE" and "PTY_AZURE_VAULT_URI" are set.

  For more information about the environment variables, refer to **Configuring Environments in KMGW** in *Azure Configuration*.

To configure the Azure Key Store by using the System Managed Identity:

1. The System Managed Identity is added to the **Local RBAC** of the Azure Managed HSM. The System Managed Identity is assigned the "Managed HSM Crypto User" role.

   If you have the Azure CLI installed, then you can perform this operation with help of the following command.

   ```
   az keyvault role assignment create --hsm-name [NAME of key vault] --role "Managed HSM
   Crypto User" --assignee [ID of system managed identity] --scope /
   ```

2. Update the environment variables for the Azure specification in the *keystore.env* config file at `opt/protegrity/keystore/external` directory.

   ```
   export PTY_AZURE_VAULT_URI=https://example.managedhsm.azure.net
   export PTY_KEYSTORE=Azure
   ```

   The Key Store is changed to the Azure in the configuration file.
3. Restart the KMGW service from the ESA Web UI.
4. In the ESA Web UI, navigate to **Policy Management** > **Key Stores** > **Key Store** to verify the Azure Key Store information.

   ```
   Information
   Manufacturer    Microsoft
   Description     Azure Key Vault
   Version         1.1.0

   General
   State           Inactive

   Azure Managed HSM Key Vault details
   Vault URI       https://example.managedhsm.azure.net
   ```

5. Click the **Test** button.
   The *Connection passed*, *Authentication passed*, and *Random passed* message appears.
6. Click the **Set as Active** button.

   The Azure Key Store is switched successfully.

   > **Note:** You must verify that the master key is rotated successfully in the Azure Key Store.

Chapter 5

Configuring the ESA with GCP Key Management

This section provides information about configuring the ESA appliance with the GCP Key Store.

5.1 Prerequisites

Ensure that the following prerequisites are met before configuring the ESA with the GCP Key Store.

5.1.1 Authorization

The resources are organized into a hierarchy in the GCP Key Store. This hierarchy helps to manage and grant access to the resources at various levels of granularity. The scope of the role depends on the level of the resource hierarchy, where the role is granted to access the Google Cloud resources.

The user or service account attached to the ESA requires the following permissions:

Table 5-1: Permissions for accessing GCP KMS

Permissions	Description
cloudkms.cryptoKeyVersions.useToEncrypt	Enable encryption using a key
cloudkms.cryptoKeyVersions.useToDecrypt	Enable decryption using a key
cloudkms.locations.generateRandomBytes	Enable access to generate random bytes
cloudkms.cryptoKeys.create	Enable creation of keys in the Key ring
cloudkms.locations.get	Enable requesting information about the configured location
cloudkms.cryptoKeyVersions.destroy	Enable destruction of keys in the Key ring
cloudkms.cryptoKeys.get	Enable access for fetching info about a specific key
cloudkms.cryptoKeys.list	Enable access for fetching all keys in a Key ring
resourcemanager.projects.get	Enables the role access to handle a project

5.1.2 Creating a Key Ring

The Enterprise Security Administrator (ESA) appliances needs a Key ring to store Key Encryption Keys (KEK).

1. Navigate to the **Key Management** screen in the Google Cloud console.
2. Click **Create key ring**.

3. In the **Key ring name** field, enter the required name for the key ring.
4. From the **Key ring location** drop-down, select a location.

   Note:
   The Key ring must be created in a location where Hardware Security Module (HSM) support is enabled. For more information on supported locations refer to *https://cloud.google.com/kms/docs/locations*.

5. Click **Create**.

Key rings are created for the selected region.

5.2 Configuring a Key Store for the GCP

The following procedure provides the steps to configure the Key Store for the Google Cloud Platform (GCP).

1. Ensure that the service account has *Specific permissions* and *Key ring* created under GCP Hardware Security Module (HSM) supported location.
2. Update the environment variables for the GCP specification in the *keystore.env* config file at `opt/protegrity/keystore/external` directory.

   The following snippet displays the sample settings for the environment variables.

   Note:
   - If the ESA instance is created on the different platform, then add the application credentials file path to the *keystore.env* file.
   - For more information on exporting credentials file, refer to GCP documentation for Application Default Credentials (ADC) *https://cloud.google.com/docs/authentication/provide-credentials-adc*.

   ```
   export PTY_KEYSTORE=GCP
   export PTY_GCP_PROJECT=<project-id>
   export PTY_GCP_LOCATION=<keyring-location>
   export PTY_GCP_KEYRING=<keyring-name>
   ```

3. On the ESA Web UI, navigate to **Key Management** > **Services** to restart the Key Management Gateway (KMGW) service.

   The KMGW Service is restarted.
4. Click the **Key Management** > **Key Stores** > **Key Store** > **Test**.

   The *Test is successful* message appears.
5. Click the **Key Management** > **Key Stores** > **Key Store** > **Set as Active**.

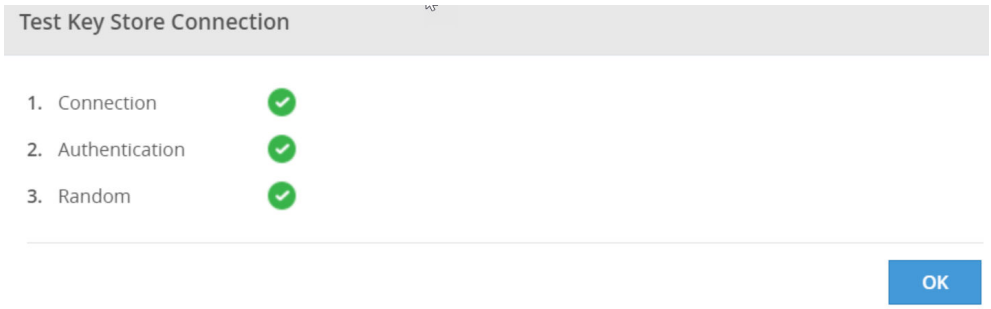
   The GCP Key Store is set as active.
6. On the ESA Web UI, navigate to **Key Management** > **Key Stores** > **Key Store** to verify the information for GCP. For more information, refer to *Viewing Key Store Information*.

   ```
   Information
   Manufacturer    Google
   Description     Cloud Key Management Service (KMS) API
   Version         1.4.0

   General
   State           Active

   Google Cloud KMS Details
   Project           <Project name>
   Key Ring          sample keyring
   Location          <location name>
   ```

5.3 Switching from Protegrity Soft HSM to GCP Key Store

The Protegrity Soft HSM stores the Master Key. If an organization uses an external Key Store as part of their IT infrastructure, then the Protegrity Key Management functionality can be configured to use the GCP Key Store.

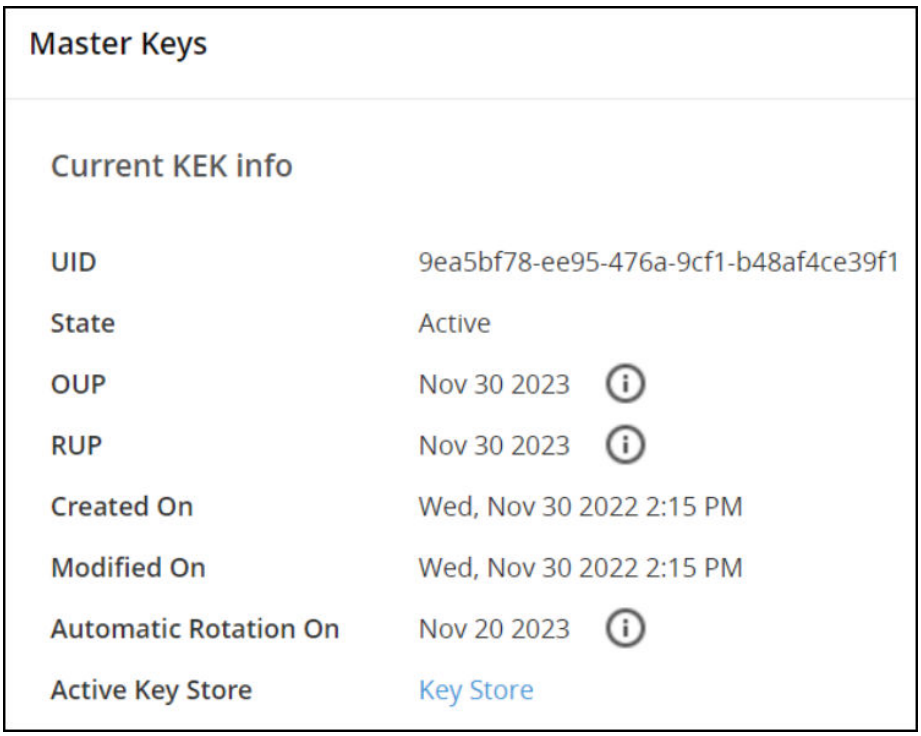
1. To switch from Protegrity Soft HSm to GCP key store, ensure that a *Key Store for GCP* is configured.
2. Test the Key Store connection for connectivity and authentication to the active Key Store. It also validates if the active Key Store generates random bytes to determine successful authentication and connection.
   a. On the ESA Web UI, navigate to **Key Management** > **Key Stores** > **Key Store**.
   b. Click **Test**.



   The **Test Key Store Connection** dialog box appears.
   c. Click **OK**.
   If the test fails due to any reason, then contact Protegrity Support for more information.
3. On the ESA Web UI, navigate to **Key Management** > **Key Stores** > **Key Store** to set the Key Store state to active.
   a. Click **Set As Active**.
   A confirmation message box appears.
   b. Click **OK**

> **Note:**
> If you are using the Key Store on an external network, then the policy management services might have a longer start-up time due to network latency

4. On the ESA Web UI, navigate to **Key Management** > **Master Keys** to verify the active Key Store.



5. On the GCP, navigate to **Key Management** > **Keyring-name** to verify the keys in the Key ring location.
   The keys are seen under the Key ring location.

   You must not make any modifications to the key version settings. For example, manually rotating them or enabling automatic rotation.

5.3.1 Viewing Key Store Information

The follwing section provides the general information related to the GCP can be viewed from the ESA Web UI.

➤ To view the Key Store information:

1. On the ESA Web UI, navigate to **Key Management** > **Key Stores**.
2. Click **Key Store**.
   The GCP-related information appears.

```
Information
Manufacturer    Google
Description     Cloud Key Management Service (KMS) API
Version         1.4.0

General
State           Active

Google Cloud KMS Details
Project           <Project name>
Key Ring          sample keyring
Location          <location name>
```

5.3.1.1 Viewing Keys under the Key Ring

This section provides steps to view the keys under the Key ring location.

To view keys under the Key ring:

1. After switching to GCP, a new KEK will have been created under the configured key ring.
2. To view the keys, navigate to the **GCP console** > **Key management** > **(Key ring name)** > **Keys**.
3. Verify that the master key uid and key under the Key ring location is the same.

> **Note:**
> The keys in the Key ring must not be rotated on GCP, as the ESA will not register it. Also, when a key is rotated on the ESA, a new key will be created in the Key ring and it will not create a new version of the key in GCP.

Chapter 6

Switching between External Key Stores

This section provides information about switching between external key stores.

Important: If you plan to switch external Key Stores due to change in vendor or an IP address change, then ensure that the external Key Store is first switched to the Protegrity Soft HSM. Then you can switch to another external Key Store after following the guidelines specified by the vendor of the other HSM.

For more information about switching from an external Key Store to the Protegrity Soft HSM, refer to the section **Switching from External Key Store to the Protegrity Soft HSM** in the *Protegrity Hardware Security Module (HSM) Integration Guide 9.1.0.1*.

Chapter 7

Troubleshooting

The following section provides information about errors that are related to Key Store integration with the ESA and the steps to resolve the errors.

Table 7-1: HSM Errors

Error/Problem	This may happen because…	Recovery
Failed to create a new Key Store. `Failed to activate new Key Store [Caused by: Failed to decrypt keys key not found: rpc error:code = NotFound desc = CryptoKey projects/project-id/locations/ keyring-location/ keyRings/example-keyring/cryptoKeys/ ca676ad1-702a-4cf0-8691-7e3896ea0432 not found.]`	This occurs when we configure GCP for Protegrity Soft HSM and try switching to GCP Key Store, it fails to create new Key Store.	The GCP Key Store must be active before configuring the GCP on the Protegrity Soft HSM.