

Data Encryption Workshop

User Guide

Issue 07
Date 2024-03-27



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: <https://www.huaweicloud.com/intl/en-us/>

Contents

| | |
|---|-----------|
| 1 Service Overview..... | 1 |
| 1.1 What Is DEW?..... | 1 |
| 1.2 Basic Concepts..... | 2 |
| 1.3 KMS..... | 3 |
| 1.3.1 Functions..... | 3 |
| 1.3.2 Advantages..... | 6 |
| 1.3.3 Application Scenarios..... | 6 |
| 1.3.4 Using KMS..... | 9 |
| 1.3.5 Cloud Services with KMS Integrated..... | 11 |
| 1.3.5.1 Encrypting Data in OBS..... | 11 |
| 1.3.5.2 Encrypting Data in EVS..... | 12 |
| 1.3.5.3 Encrypting Data in IMS..... | 12 |
| 1.3.5.4 Encrypting Data in SFS | 12 |
| 1.3.5.5 Encrypting Data in RDS..... | 12 |
| 1.3.5.6 Encrypting Data in DDS..... | 13 |
| 1.4 CSMS..... | 13 |
| 1.4.1 Functions..... | 13 |
| 1.4.2 Advantages..... | 14 |
| 1.4.3 Application Scenarios..... | 14 |
| 1.5 DEW Permission Management..... | 15 |
| 1.6 How to Access..... | 17 |
| 1.7 Related Services..... | 18 |
| 1.8 Personal Data Protection Mechanism..... | 21 |
| 2 User Guide..... | 22 |
| 2.1 Key Management Service..... | 22 |
| 2.1.1 Key Types..... | 22 |
| 2.1.2 Creating a Key..... | 23 |
| 2.1.3 Creating CMKs Using Imported Key Materials..... | 26 |
| 2.1.3.1 Overview..... | 26 |
| 2.1.3.2 Importing Key Materials..... | 27 |
| 2.1.3.3 Deleting Key Materials..... | 33 |
| 2.1.4 Managing CMKs..... | 34 |
| 2.1.4.1 Viewing a CMK..... | 34 |

| | |
|---|----|
| 2.1.4.2 Enabling One or More CMKs..... | 35 |
| 2.1.4.3 Disabling One or More CMKs..... | 36 |
| 2.1.4.4 Deleting One or More CMKs..... | 36 |
| 2.1.4.5 Canceling the Scheduled Deletion of One or More CMKs..... | 37 |
| 2.1.4.6 Adding a Key to a Project..... | 38 |
| 2.1.5 Searching for a Key..... | 39 |
| 2.1.6 Using the Online Tool to Encrypt and Decrypt Small-Size Data..... | 39 |
| 2.1.7 Managing Tags..... | 40 |
| 2.1.7.1 Adding a Tag..... | 40 |
| 2.1.7.2 Modifying Tag Values..... | 41 |
| 2.1.7.3 Deleting Tags..... | 42 |
| 2.1.8 Rotating CMKs..... | 42 |
| 2.1.8.1 About Key Rotation..... | 42 |
| 2.1.8.2 Enabling Key Rotation..... | 45 |
| 2.1.8.3 Disabling Key Rotation..... | 46 |
| 2.1.9 Managing a Grant..... | 47 |
| 2.1.9.1 Creating a Grant..... | 47 |
| 2.1.9.2 Querying a Grant..... | 49 |
| 2.1.9.3 Revoking a Grant..... | 50 |
| 2.2 Cloud Secret Management Service..... | 51 |
| 2.2.1 Creating a Secret..... | 51 |
| 2.2.1.1 Creating a Shared Secret..... | 51 |
| 2.2.2 Managing Secrets..... | 52 |
| 2.2.2.1 Viewing a Secret..... | 52 |
| 2.2.2.2 Deleting a Secret..... | 53 |
| 2.2.3 Managing Secret Versions..... | 53 |
| 2.2.3.1 Saving and Viewing Secret Values..... | 53 |
| 2.2.3.2 Managing Secret Version Statuses..... | 54 |
| 2.2.4 Managing Tags..... | 55 |
| 2.2.4.1 Adding a Tag..... | 55 |
| 2.2.4.2 Searching for a Secret by Tag..... | 56 |
| 2.2.4.3 Modifying a Tag Value..... | 57 |
| 2.2.4.4 Deleting a Tag..... | 57 |
| 2.2.5 Creating an Event..... | 57 |
| 2.2.6 Managing Events..... | 59 |
| 2.2.6.1 Viewing Events..... | 59 |
| 2.2.6.2 Editing an Event..... | 60 |
| 2.2.6.3 Enabling an Event..... | 61 |
| 2.2.6.4 Disabling an Event..... | 62 |
| 2.2.6.5 Deleting an Event..... | 63 |
| 2.2.7 Viewing Notifications..... | 64 |
| 2.3 Auditing Logs..... | 64 |

| | |
|---|-----------|
| 2.3.1 Operations supported by CTS..... | 65 |
| 2.3.2 Querying Real-Time Traces..... | 66 |
| 2.4 Permission Control..... | 68 |
| 2.4.1 Creating a User and Authorizing the User the Permission to Access DEW..... | 68 |
| 2.4.2 Creating a Custom DEW Policy..... | 69 |
| 3 FAQs..... | 72 |
| 3.1 KMS Related..... | 72 |
| 3.1.1 What Is Key Management Service?..... | 72 |
| 3.1.2 What Is a Customer Master Key?..... | 72 |
| 3.1.3 What Is a Default Key?..... | 73 |
| 3.1.4 What Are the Differences Between a Custom Key and a Default Key?..... | 73 |
| 3.1.5 What Is a Data Encryption Key?..... | 74 |
| 3.1.6 Why Cannot I Delete a CMK Immediately?..... | 74 |
| 3.1.7 Which Cloud Services Can Use KMS for Encryption?..... | 74 |
| 3.1.8 How Do Cloud Services Use KMS to Encrypt Data?..... | 76 |
| 3.1.9 What Are the Benefits of Envelope Encryption?..... | 76 |
| 3.1.10 Is There a Limit on the Number of Custom Keys That I Can Create on KMS?..... | 77 |
| 3.1.11 Can I Export a CMK from KMS?..... | 77 |
| 3.1.12 Can I Decrypt My Data if I Permanently Delete My Custom Key?..... | 77 |
| 3.1.13 How Do I Use the Online Tool to Encrypt or Decrypt Small Volumes of Data?..... | 77 |
| 3.1.14 Can I Update CMKs Created by KMS-Generated Key Materials?..... | 78 |
| 3.1.15 When Should I Use a CMK Created with Imported Key Materials?..... | 78 |
| 3.1.16 What Types of Keys Can I Import?..... | 78 |
| 3.1.17 What Should I Do When I Accidentally Delete Key Materials?..... | 79 |
| 3.1.18 How Are Default Keys Generated?..... | 79 |
| 3.1.19 What Should I Do If I Do Not Have the Permissions to Perform Operations on KMS?..... | 79 |
| 3.1.20 Why Can't I Wrap Asymmetric Keys by Using -id-aes256-wrap-pad in OpenSSL?..... | 80 |
| 3.1.21 Key Algorithms Supported by KMS..... | 81 |
| 3.1.22 What Should I Do If KMS Failed to Be Requested and Error Code 401 Is Displayed?..... | 82 |
| 3.1.23 What Is the Relationship Between the Ciphertext and Plaintext Returned by the encrypt-data API?..... | 83 |
| 3.1.24 How Does KMS Protect My Keys?..... | 83 |
| 3.2 Credential Related..... | 83 |
| 3.2.1 Why Cannot I Delete the Version Status of a Secret?..... | 83 |
| A Change History..... | 84 |

1 Service Overview

1.1 What Is DEW?

DEW

Data is the core asset of an enterprise. Each enterprise has its core sensitive data, which needs to be encrypted and protected from breach.

Data Encryption Workshop (DEW) is a cloud data encryption service. It provides Key Management Service (KMS), , and Cloud Secret Management Service (CSMS). DEW secures your data and keys, as well as simplifies key management. DEW uses hardware security modules (HSMs) to protect the security of your keys and can be integrated with multiple cloud services. Additionally, DEW enables you to develop customized encryption applications.

Table 1-1 Service overview

| Service | Description |
|--|--|
| Key Management Service (KMS) | KMS is a secure, reliable, and easy-to-use service for managing your keys on the cloud. It helps you easily create, manage, and protect keys. KMS uses hardware security modules (HSMs) to protect keys. HSM meets the FIPS 140-2 Level 3 security requirements. It helps you create and manage keys. All keys are protected by root keys in HSMs to avoid key leakage. |
| Cloud Secret Management Service (CSMS) | CSMS is a secure, reliable, and easy-to-use secret hosting service. Users or applications can use CSMS to create, retrieve, update, and delete credentials in a unified manner throughout the secret lifecycle. CSMS can help you eliminate risks incurred by hardcoding, plaintext configuration, and permission abuse. |

1.2 Basic Concepts

This section describes the basic DEW concepts for you better understand and use DEW.

Table 1-2 Common encryption terms

| Term | Definition |
|---------------------------|--|
| Symmetric key encryption | <p>Symmetric key encryption is also called dedicated key encryption. The sender and receiver use the same key to encrypt and decrypt data.</p> <p>Advantage: Encryption and decryption are fast.</p> <p>Disadvantage: Each pair of keys must be unique. Key management is difficult if there are a large number of users.</p> <p>Scenario: Encrypt a large amount of data.</p> |
| Asymmetric key encryption | <p>Asymmetric key encryption is also called public key encryption. A pair of keys are used for encryption and decryption. One is a public key, and the other is a private key.</p> <p>Advantage: Different keys are used for encryption and decryption, enhancing security.</p> <p>Disadvantage: Encryption and decryption are slow.</p> <p>Scenario: Encrypt sensitive information.</p> |

Table 1-3 KMS terms

| Item | Definition |
|--------------------------------|--|
| Hardware Security Module (HSM) | <p>An HSM is a type of computer hardware that protects and manages the keys used by strong authentication systems and provides related cryptographic operations.</p> |
| Customer Master Key (CMK) | <p>A CMK is a main encryption key created by a user or cloud service using KMS. It is used to encrypt and protect data encryption keys (DEKs). One CMK can be used to encrypt one or more DEKs.</p> <p>CMKs are categorized into custom keys and default keys.</p> |

| Item | Definition |
|---------------------------|--|
| Default key | A default key is automatically created by another cloud service using KMS, such as Object Storage Service (OBS). The alias of a default key ends with /default . |
| Key material | Key materials are important input for cryptographic operations. A CMK consists of a key ID, metadata, and a key material. |
| Envelope encryption | Envelope encryption is the practice of encrypting data with a DEK and then encrypting the DEK with a root key that you can fully manage. In this case, CMKs are not required for encryption or decryption. |
| Data Encryption Key (DEK) | A DEK is used to encrypt data. |

1.3 KMS

1.3.1 Functions

KMS is a secure, reliable, and easy-to-use cloud service that helps users create, manage, and protect keys in a centralized manner.

It uses Hardware Security Modules (HSMs) to protect keys. All keys are protected by root keys in HSMs to avoid key leakage. The HSM module meets the FIPS 140-2 Level 3 security requirements.

It also controls access to keys and records all operations on keys with traceable logs. In addition, it provides use records of all keys, meeting your audit and regulatory compliance requirements.

Functions

- On the KMS console, you can:
 - Create, query, enable, and disable CMKs, as well as schedule and cancel CMK deletion.
 - Modify the alias and description of CMKs.
 - Use the online tool to encrypt and decrypt small-size data.
 - Import keys and delete key material.
 - Add, search for, edit, and delete tags.
 - Create, cancel, and query grants.
- You can use the API to perform the following operations:
 - Create, encrypt, or decrypt DEKs.
 - Retire grants.

For details, see *Data Encryption Workshop API Reference*.

- Generate hardware true random numbers.

You can generate 512-bit hardware true random numbers using a KMS API. The numbers can be used as a basis for key materials or as encryption parameters. For details, see the *Data Encryption Workshop API Reference*.

Key Algorithms Supported by KMS

Symmetric keys created on the KMS console use the AES algorithm. Asymmetric keys created by KMS support the RSA and ECC algorithms.

Table 1-4 Key algorithms supported by KMS

| Key Type | Algorithm Type | Key Specifications | Description | Application Scenario |
|---------------|----------------|--------------------|-------------------|---|
| Symmetric key | AES | AES_256 | AES symmetric key | <ul style="list-style-type: none"> • Data encryption and decryption • DEKs encryption and decryption <p>NOTE You can encrypt and decrypt a small amount of data using the online tool on the console. You need to call APIs to encrypt and decrypt a large amount of data.</p> |

| Key Type | Algorithm Type | Key Specifications | Description | Application Scenario |
|----------------|----------------|--|------------------------------------|---|
| Asymmetric key | RSA | <ul style="list-style-type: none"> • RSA_2048 • RSA_3072 • RSA_4096 | RSA asymmetric password | <ul style="list-style-type: none"> • Digital signature and signature verification • Data encryption and decryption <p>NOTE Asymmetric keys are applicable to signature and signature verification scenarios. Asymmetric keys are not efficient enough for data encryption. Symmetric keys are suitable for encrypting and decrypting data.</p> |
| | ECC | <ul style="list-style-type: none"> • EC_P256 • EC_P384 | Elliptic curve recommended by NIST | Digital signature and signature verification |

Table 1-5 describes the encryption and decryption algorithms supported for user-imported keys.

Table 1-5 Key wrapping algorithms

| Algorithm | Description | Configuration |
|--------------------|---|---|
| RSAES_OAEP_SHA_256 | RSA algorithm that uses OAEP and has the SHA-256 hash function | Select an algorithm based on your HSM functions. If your HSM supports the RSAES_OAEP_SHA_256 algorithm, use RSAES_OAEP_SHA_256 to encrypt key materials. |

1.3.2 Advantages

Extensive Service Integration

- By integrating with OBS, EVS, and IMS, you can use KMS to manage the keys of the services or use KMS APIs to encrypt and decrypt local data.
- By integrating with Cloud Trace Service (CTS), you can use CTS to view recent KMS operation records.

Regulatory Compliance

Keys are generated by third-party validated HSMs. Access to keys is controlled and all operations involving keys are traceable by logs, compliant with international laws and regulations.

Easy to Use

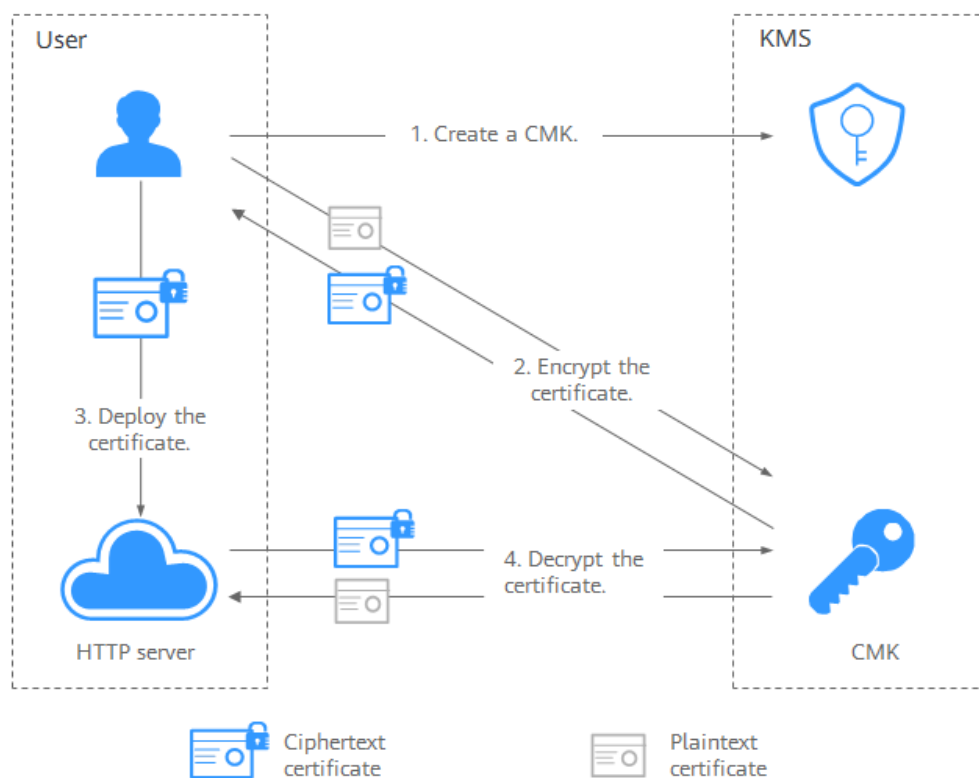
You can use and manage keys easily using the console or APIs, needless to purchase hardware encryption devices.

1.3.3 Application Scenarios

Small Data Encryption and Decryption

You can use the online tool on the KMS console or call KMS APIs to directly encrypt or decrypt a small amount of data, such as passwords, certificates, or phone numbers. Currently, a maximum of 4 KB of data can be encrypted or decrypted in this way.

Figure 1-1 shows an example about how to call the APIs to encrypt and decrypt an HTTPS certificate.

Figure 1-1 Encrypting and decrypting an HTTPS certificate

The procedure is as follows:

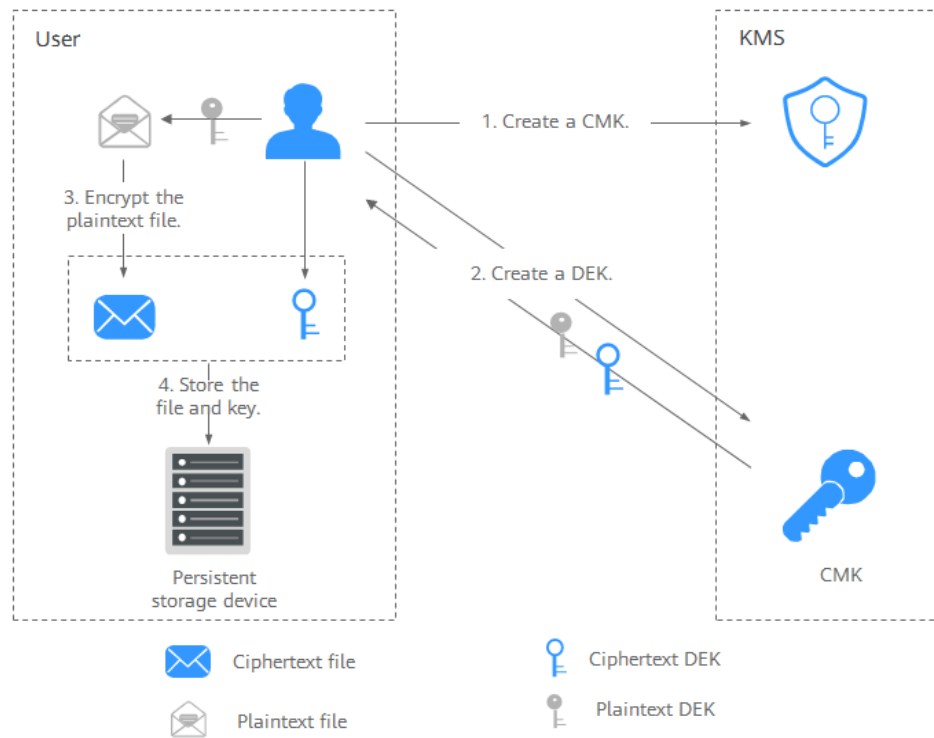
1. Create a CMK on KMS.
2. Call the **encrypt-data** API of KMS and use the CMK to encrypt the plaintext certificate.
3. Deploy the certificate onto a server.
4. The server calls the **decrypt-data** API of KMS to decrypt the ciphertext certificate.

Large Data Encryption and Decryption

If you want to encrypt or decrypt large volumes of data, such as pictures, videos, and database files, you can use the envelope encryption method, where the data does not need to be transferred over the network.

- **Figure 1-2** illustrates the process for encrypting a local file.

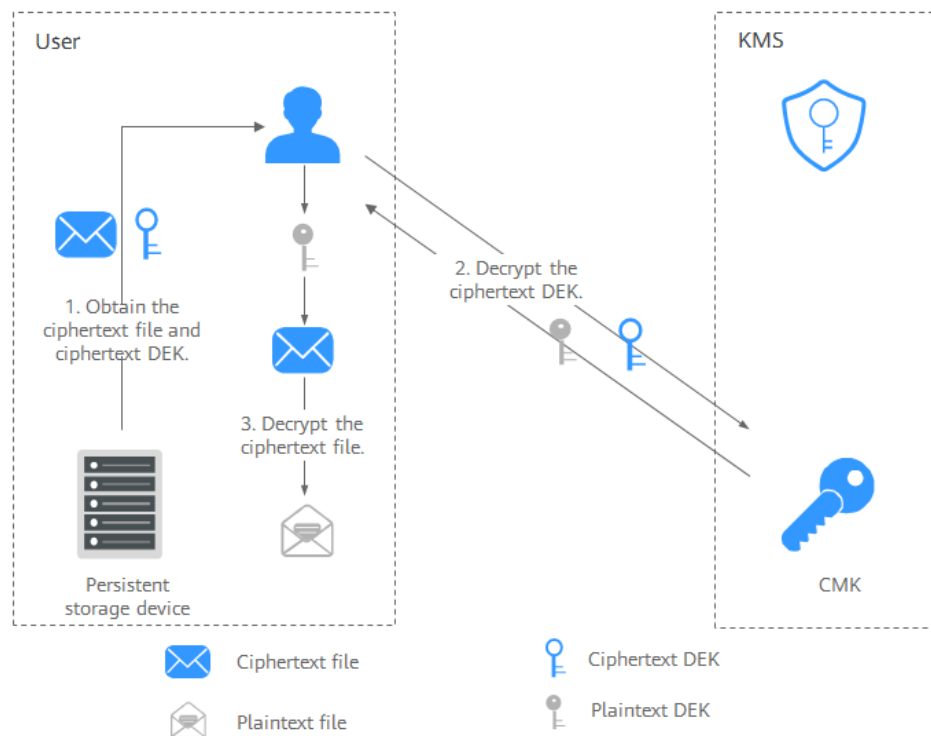
Figure 1-2 Encrypting a local file



The procedure is as follows:

- Create a CMK on KMS.
 - Call the **create-datakey** API of KMS to create a DEK. Then you get a plaintext DEK and a ciphertext DEK. The ciphertext DEK was generated by using a custom key to encrypt the plaintext DEK.
 - Use the plaintext DEK to encrypt the file. A ciphertext file is generated.
 - Save the ciphertext DEK and the ciphertext file together in a persistent storage device or a storage service.
- Figure 1-3** illustrates the process for decrypting a local file.

Figure 1-3 Decrypting a local file



The procedure is as follows:

- Obtain the ciphertext DEK and file from the persistent storage device or the storage service.
- Call the **decrypt-datakey** API of KMS and use the corresponding CMK (the one used for encrypting the DEK) to decrypt the ciphertext DEK. Then you get the plaintext DEK.
If the CMK is deleted, the decryption fails. Therefore, properly keep your CMKs.
- Use the plaintext DEK to decrypt the ciphertext file.

1.3.4 Using KMS

Interacting with Cloud Services

Cloud services use the envelope encryption technology and call KMS APIs to encrypt service resources. Your CMKs are under your own management. With your grant, cloud services use a specific custom key of yours to encrypt data.

The encryption process is as follows:

- Create a custom key on KMS.
- Cloud services call the **create-datakey** API of the KMS to create a DEK. Then you get a plaintext DEK and a ciphertext DEK.

NOTE

Ciphertext DEKs are generated when you use a CMK to encrypt the plaintext DEKs.

3. Cloud services use the plaintext DEK to encrypt a plaintext file, generating a ciphertext file.
4. Cloud services store the ciphertext DEK and ciphertext file in a persistent storage device or a storage service.

 **NOTE**

When users download the data from a cloud service, the service uses the custom key specified by KMS to decrypt the ciphertext DEK, uses the decrypted DEK to decrypt data, and then provides the decrypted data for users to download.

Table 1-6 List of cloud services that use KMS encryption

| Service Name | Description |
|-----------------------------------|---|
| Object Storage Service (OBS) | <p>You can upload objects to and download them from Object Storage Service (OBS) in common mode or server-side encryption mode. When you upload objects in encryption mode, data is encrypted at the server side and then securely stored on OBS in ciphertext. When you download encrypted objects, the data in ciphertext is decrypted at the server side and then provided to you in plaintext. OBS supports the server-side encryption with KMS-managed keys (SSE-KMS) mode. In SSE-KMS mode, OBS uses the keys provided by KMS for server-side encryption.</p> <p>For details about how to upload objects to OBS in SSE-KMS mode, see the <i>Object Storage Service Console Operation Guide</i>.</p> |
| Elastic Volume Service (EVS) | <p>If you enable the encryption function when creating an EVS disk, the disk will be encrypted with the DEK generated by using your CMK. Data stored in the EVS disk will be automatically encrypted.</p> <p>For details about how to use the encryption function of EVS, see <i>Elastic Volume Service User Guide</i>.</p> |
| Image Management Service (IMS) | <p>When creating a private image using an external image file, you can enable the private image encryption function and select a CMK provided by KMS to encrypt the image.</p> <p>For details about how to use the private image encryption function of Image Management Service (IMS), see <i>Image Management Service User Guide</i>.</p> |
| Scalable File Service (SFS) | <p>When creating a file system on SFS, the CMK provided by KMS can be selected to encrypt the file system, so that files stored in the file system are automatically encrypted.</p> <p>For details about how to use the file system encryption function of SFS, see <i>Scalable File Service User Guide</i>.</p> |
| Relational Database Service (RDS) | <p>When purchasing a database instance, you can enable the disk encryption function of the database instance and select a CMK created on KMS to encrypt the disk of the database instance. Enabling the disk encryption function will enhance data security.</p> <p>For details about how to use the disk encryption function of RDS, see <i>Relational Database Service User Guide</i>.</p> |

| Service Name | Description |
|---------------------------------|---|
| Document Database Service (DDS) | When purchasing a DDS instance, you can enable the disk encryption function of the instance and select a CMK created on KMS to encrypt the disk of the instance. Enabling the disk encryption function will enhance data security. For details about how to use the disk encryption function of DDS, see <i>Document Database Service User Guide</i> . |

Working with User Applications

To encrypt plaintext data, a user application can call the necessary KMS API to create a DEK. The DEK can then be used to encrypt the plaintext data. Then the application can store the encrypted data. In addition, the user application can call the KMS API to create CMKs. DEKs can be stored in ciphertext after being encrypted with the CMKs.

Envelope encryption is implemented, with CMKs stored in KMS and ciphertext DEKs in user applications. KMS is called to decrypt a ciphertext DEK only when necessary.

The encryption process is as follows:

1. The application calls the **create-key** API of KMS to create a custom key.
2. The application calls the **create-datakey** API of KMS to create a DEK. A plaintext DEK and a ciphertext DEK are generated.

NOTE

Ciphertext DEKs are generated when you use a CMK to encrypt the plaintext DEKs in [1](#).

3. The application uses the plaintext DEK to encrypt a plaintext file. A ciphertext file is generated.
4. The application saves the ciphertext DEK and the ciphertext file together in a persistent storage device or a storage service.

For details, see the *Data Encryption Workshop API Reference*.

1.3.5 Cloud Services with KMS Integrated

1.3.5.1 Encrypting Data in OBS

- When using Object Storage Service (OBS) to upload data with server-side encryption, you can select **SEE-KMS encryption** and use the key provided by KMS to encrypt the files to be uploaded. For more information, see *Object Storage Service User Guide*.

There are two types of CMKs that can be used:

- The default key **obs/default** created by KMS
- Custom keys that you created on the KMS console

- Alternatively, you can call OBS APIs to upload a file with server-side encryption using KMS-managed keys (SSE-KMS). For details, see the *Object Storage Service API Reference*.

1.3.5.2 Encrypting Data in EVS

- When purchasing a disk, you can choose **Advanced Settings > Encryption** to encrypt the disk using the key provided by KMS. For more information about EVS, see the *Elastic Volume Service User Guide*.

NOTE

Before you use the encryption function, EVS must be granted the permission to access KMS. If you have the right to grant the permission, you can grant the permission directly. If you do not have the permission, contact a user with the security administrator permissions to add the security administrator permission for you. Then, you can grant the permission. For more information about EVS, see the *Elastic Volume Service User Guide*.

There are two types of CMKs that can be used:

- The default key **evs/default** created by KMS
 - Custom keys that you create on the KMS console using KMS-generated key materials
- You can also call EVS APIs to create encrypted EVS disks. For details, see the *Elastic Volume Service API Reference*.

1.3.5.3 Encrypting Data in IMS

- When uploading an image file to Image Management Service (IMS), you can choose to encrypt the image file using a key provided by KMS to protect the file. For details, see the *Image Management Service User Guide*.

There are two types of CMKs that can be used:

- The default key **ims/default** created by KMS
 - Custom keys that you create on the KMS console using KMS-generated key materials
- You can also call IMS APIs to create encrypted image files. For details, see *Image Management Service API Reference*.

1.3.5.4 Encrypting Data in SFS

- When creating a file system using the Scalable File Service (SFS), you can select **KMS encryption** and use the key provided by the KMS to encrypt the file system. For more information, see the *Scalable File Service User Guide*.
You can use a custom key created on the KMS console for encryption.
- You can use the SFS API to create an encrypted file system. For details, see the *Scalable File Service API Reference*.

1.3.5.5 Encrypting Data in RDS

- When a user purchases a database instance from Relational Database Service (RDS), the user can select **Disk encryption** and use the key provided by KMS to encrypt the disk of the database instance. For more information, see the *Relational Database Service User Guide*.
You can use a custom key created on the KMS console for encryption.

- You can also call the RDS APIs to purchase encrypted database instances. For details, see the *Relational Database Service User Guide*.

1.3.5.6 Encrypting Data in DDS

- When a user creates a database instance from DDS, the user can select **Disk encryption** and use the key provided by KMS to encrypt the disk of the database instance. For more information, see the *Document Database Service User Guide*.

You can use a custom key created on the KMS console for encryption.

- You can also call the required API of DDS to purchase encrypted DB instances. For details, see *Document Database Service API Reference*.

1.4 CSMS

1.4.1 Functions

CSMS is a secure, reliable, and easy-to-use secret hosting service. Users or applications can use CSMS to create, retrieve, update, and delete credentials in a unified manner throughout the secret lifecycle. CSMS can help you eliminate risks incurred by hardcoding, plaintext configuration, and permission abuse.

Unified Secret Management

Applications and business systems have a large number of secrets and are difficult to manage.

CSMS can store, retrieve, and use secrets in a unified manner throughout their lifecycles.

Perform the following operations to manage secrets using CSMS:

1. Collect secrets.
2. Upload the secrets to CSMS.

Secure Secret Retrieval

Many applications store plaintext secrets, such as passwords, tokens, certificates, SSH keys, and API keys, in their configuration files to be used for authentication when they access databases or other services. Plaintext and hardcoded secrets are prone to breach and incur security risks.

CSMS allows users to dynamically query secrets via APIs instead of hardcoding the secrets, greatly reducing breach risks.

Perform the following operations to manage secrets using CSMS:

When an application reads its configurations, it calls CSMS APIs to retrieve secrets. Neither hardcoded nor plaintext secrets are required.

CSMS Basic Features

Table 1-7 CSMS basic features

| Function | Description |
|----------------------------------|---|
| Secret lifecycle management | <ul style="list-style-type: none"> • Create, view, and schedule and cancel the deletion of secrets. • Change the secret encryption key and description. |
| Secret version management | <ul style="list-style-type: none"> • Create and view secret versions. • View secret values. |
| Secret version status management | Update, query, and delete secret versions. |
| Secret tag management | Add, search for, edit, and delete tags. |

1.4.2 Advantages

Secret encryption

Secrets are encrypted by KMS before storage. Encryption keys are generated and protected by authenticated third-party HSM. When you retrieve secrets, they are transferred to local servers via TLS.

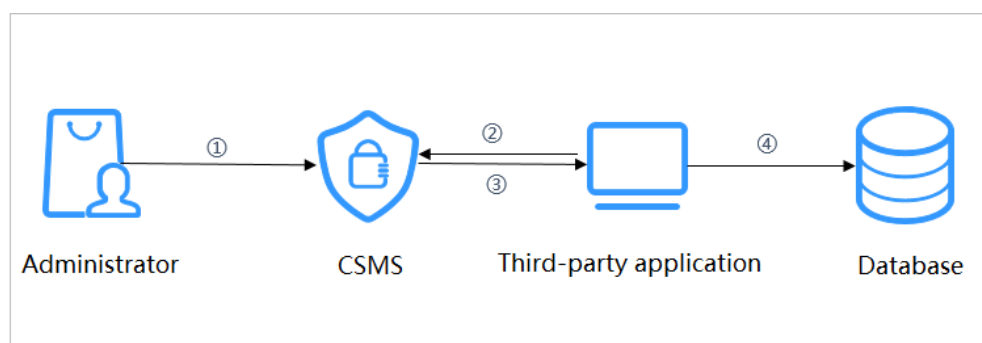
Secure secret retrieval

CSMS calls secret APIs instead of hard-coded secrets in applications. Secrets can be dynamically retrieved and managed. CSMS manages application secrets in a centralized manner to reduce breach risks.

1.4.3 Application Scenarios

This section uses a basic database username and its password as an example to describe how CSMS works.

Figure 1-4 Secret-based login process



The procedure is as follows:

- Step 1** Create a secret on the console or via an API to store database information (such as the database address, port, and password).
- Step 2** Use an application to access the database. CSMS will query the secret you created.
- Step 3** CSMS retrieves and decrypts the secret ciphertext, and securely returns the information stored in the secret to the application through the secret management API.
- Step 4** The application obtains the decrypted plaintext secret and uses it to access the database.

----End

1.5 DEW Permission Management

If you want to assign different access permissions to employees in an enterprise for the DEW resources purchased on the cloud platform, you can use Identity and Access Management (IAM) to perform refined permission management. IAM provides identity authentication, permissions management, and access control, helping you secure access to your resources.

With IAM, you can use your account to create IAM users for your employees, and grant permissions to control their access to specific resource types. For example, if you have software developers and you want to assign them the permission to access DEW but not to delete DEW or its resources, then you can create an IAM policy to assign the developers the permission to access DEW but prevent them from deleting DEW related data.

If the system account has met your requirements and you do not need to create an independent IAM user for permission control, then you can skip this section. This will not affect other functions of DEW.

DEW Permissions

By default, new IAM users do not have permissions assigned. You need to add a user to one or more groups, and attach permissions policies or roles to these groups. Users inherit permissions from their groups and can perform specified operations on cloud services based on the permissions.

DEW is a project-level service deployed and accessed in specific physical regions. To assign permissions to a user group, specify the scope as region-specific projects and select projects for the permissions to take effect. If **All projects** is selected, the permissions will take effect for the user group in all region-specific projects. Users need to switch to the authorized region when accessing KMS.

You can grant users permissions by using roles and policies.

- **Roles:** A type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. This mechanism provides only a limited number of service-level roles for authorization. When using roles to grant permissions, you must also assign other roles that the permissions depend on to take effect. However, roles are not an ideal choice for fine-grained authorization and secure access control.

- **Policies:** A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization, meeting requirements for secure access control. For example, you can grant KMS users only the permissions for managing a certain type of cloud servers. Most policies contain permissions for specific APIs, and permissions are defined using API actions.

For more information, see [Table 1-8](#).

Table 1-8 DEW permissions

| Role/Policy | Description | Type |
|-------------------|--|--------|
| KMS Administrator | Administrator permissions for the encryption key | Role |
| KMS CMK Admin | All permissions for the encryption keys | Policy |

[Table 1-9](#) lists the common operations supported by each system-defined permission of DEW. Select the permissions as needed.

Table 1-9 Common operations supported by each system-defined policy or role

| Operation | KMS Administrator |
|---------------------------------------|-------------------|
| Create a key | √ |
| Enable a key | √ |
| Disable a key | √ |
| Schedule key deletion | √ |
| Cancel scheduled key deletion | √ |
| Modify a key alias | √ |
| Modify key description | √ |
| Generate a random number | √ |
| Create a DEK | √ |
| Create a plaintext-free DEK | √ |
| Encrypt a DEK | √ |
| Decrypt a DEK | √ |
| Obtain parameters for importing a key | √ |
| Import key materials | √ |
| Delete key materials | √ |


| Operation | KMS Administrator |
|------------------------------|-------------------|
| Create a grant | √ |
| Revoke a grant | √ |
| Retire a grant | √ |
| Query the grant list | √ |
| Query retirable grants | √ |
| Encrypt data | √ |
| Decrypt data | √ |
| Enable key rotation | √ |
| Modify key rotation interval | √ |
| Disable key rotation | √ |
| Query key rotation status | √ |
| Query CMK instances | √ |
| Query key tags | √ |
| Query project tags | √ |
| Batch add or delete key tags | √ |
| Add tags to a key | √ |
| Delete key tags | √ |
| Query the key list | √ |
| Query key details | √ |
| Query instance quantity | √ |
| Query quotas | √ |

Related Links

- Two types of permission policies are provided by default: default policies and custom policies. Default policies are pre-defined by IAM and cannot be modified. If default policies do not meet your requirements, you can create custom policies for fine-grained permission control.
- Configure permission policies for a user group and add users to the group so that these users can obtain operation permissions defined in the policies.

1.6 How to Access

- Management console

Log in to the management console, click  in the upper left corner of the management console, and select a region or project. Then choose **Service List > Security > Data Encryption Workshop**.

- API

You can access DEW using the API. For details, see the *Data Encryption Workshop API Reference*.

1.7 Related Services

OBS

Object Storage Service (OBS) is a scalable service that provides secure, reliable, and cost-effective cloud storage for massive amounts of data. KMS provides central management and control capabilities of CMKs for OBS. It is used for server-side encryption with KMS-managed keys (SSE-KMS) on OBS.

EVS

Elastic Volume Service (EVS) offers scalable block storage for cloud servers. With high reliability, high performance, and rich specifications, EVS disks can be used for distributed file systems, development and test environments, data warehouse applications, and high-performance computing (HPC) scenarios to meet diverse service requirements. KMS provides central management and control capabilities of CMKs for EVS. It is used for encryption in EVS.

IMS

Image Management Service (IMS) allows you to manage the entire lifecycle of your images. KMS provides central management and control capabilities of CMKs for Image Management Service (IMS). It is used for private image encryption in IMS.

SFS

Scalable File Service (SFS) provides high-performance file storage (NAS) that can be expanded on demand. KMS provides central management and control capabilities of CMKs for SFS. It is used for file system encryption in SFS.

RDSRDS

Relational Database Service (RDS) Relational Database Service (RDS) is a cloud database relational database that is reliable, scalable, easy to manage, and immediately ready for use. KMS provides central management and control capabilities of CMKs for RDS. It is used for disk encryption in cloud databases relational databases.

ECS

An ECS is a basic computing component that consists of CPUs, memory, OS, and elastic volume service (EVS). After creating an ECS, you can use it like your local computer or physical server.

DDS

Document Database Service (DDS) is a MongoDB-compatible database service that is secure, highly available, reliable, scalable, and easy to use. It provides DB instance creation, scaling, redundancy, backup, restoration, monitoring, and alarm reporting functions with just a few clicks on the DDS console. KMS provides central management and control capabilities of CMKs for DDS. It is used for disk encryption in DDS.

CTS

Cloud Trace Service (CTS) provides you with a history of DEW operations. After the CTS service is enabled, you can view all generated traces to review and audit performed KMS operations. For details, see the *Cloud Trace Service User Guide*.

Table 1-10 DEW operations supported by CTS

| Operation | Resource Type | Trace Name |
|---------------------------------|---------------|-------------------------------|
| Create a key | cmk | createKey |
| Create a DEK | cmk | createDataKey |
| Create a plaintext-free DEK | cmk | createDataKeyWithoutPlaintext |
| Enable a key | cmk | enableKey |
| Disable a key | cmk | disableKey |
| Encrypt a DEK | cmk | encryptDatakey |
| Decrypt a DEK | cmk | decryptDatakey |
| Schedule key deletion | cmk | scheduleKeyDeletion |
| Cancel scheduled key deletion | cmk | cancelKeyDeletion |
| Generate random numbers | rng | genRandom |
| Modify a key alias | cmk | updateKeyAlias |
| Modify key description | cmk | updateKeyDescription |
| Prompt risks about CMK deletion | cmk | deleteKeyRiskTips |
| Import key materials | cmk | importKeyMaterial |
| Delete key materials | cmk | deleteImportedKeyMaterial |

| Operation | Resource Type | Trace Name |
|--------------------------------------|---------------|--------------------------------|
| Create a grant | cmk | createGrant |
| Retire a grant | cmk | retireGrant |
| Revoke a grant | cmk | revokeGrant |
| Encrypt data | cmk | encryptData |
| Decrypt data | cmk | decryptData |
| Add a tag | cmk | createKeyTag |
| Delete a tag | cmk | deleteKeyTag |
| Add tags in batches | cmk | batchCreateKeyTags |
| Delete tags in batches | cmk | batchDeleteKeyTags |
| Enable key rotation | cmk | enableKeyRotation |
| Modify key rotation interval | cmk | updateKeyRotationInterval |
| Disable key rotation | cmk | disableKeyRotation |
| Create a secret | csms | createSecret |
| Update a secret | csms | updateSecret |
| Delete a secret | csms | forceDeleteSecret |
| Schedule the deletion of a secret | csms | scheduleDelSecret |
| Cancel the scheduled secret deletion | csms | restoreSecretFromDeletedStatus |
| Create a secret status | csms | createSecretStage |
| Update a secret status | csms | updateSecretStage |
| Delete a secret status | csms | deleteSecretStage |
| Create a secret version | csms | createSecretVersion |
| Download a secret backup | csms | backupSecret |
| Restore a secret backup | csms | restoreSecretFromBackupBlob |

IAM

Identity and Access Management (IAM) provides the permission management function for DEW.

Only users who have KMS Administrator permissions can use DEW.

To apply for permissions, contact a user with Security Administrator permissions. For details, see the *Identity and Access Management User Guide*.

1.8 Personal Data Protection Mechanism

To ensure that your personal data, such as the username, password, and mobile phone number, will not be leaked or obtained by unauthorized or unauthenticated entities or people, DEW controls access to the data and records logs for operations performed on the data.

Personal Data to Be Collected

Table 1-11 lists the personal data generated or collected by DEW.

Table 1-11 Personal data

| Type | Source | Can Be Modified | Mandatory |
|-----------|--|-----------------|-----------|
| Tenant ID | <ul style="list-style-type: none">Tenant ID in the token when an operation is performed on the console.Tenant ID in the token when an API is invoked. | No | Yes |

Storage Mode

Tenant IDs are not sensitive data and are stored in plaintext.

Access Permission Control

Users can view only logs related to their own services.

Log Records

DEW records logs for all operations, such as editing, querying, and deleting, performed on personal data. The logs are uploaded to Cloud Trace Service (CTS). You can view only the logs generated for operations you performed.

2 User Guide

2.1 Key Management Service

2.1.1 Key Types

CMKs include custom keys and default keys. This section describes how to create, view, enable, disable, schedule the deletion, and cancel the deletion of custom keys.

Custom keys can be categorized into symmetric keys and asymmetric keys.

Symmetric keys are most commonly used for data encryption protection. Asymmetric keys are used for digital signature verification or sensitive information encryption in systems where the trust relationship is not mutual. An asymmetric key consists of a public key and a private key. The public key can be sent to anyone. The private key must be securely stored and only accessible to trusted users.

An asymmetric key can be used to generate and verify a signature. To securely transfer data, a signer sends the public key to a receiver, uses the private key to sign data, and then sends the data and signature to the receiver. The receiver can use the public key to verify the signature.

Table 2-1 Key algorithms supported by KMS

| Key Type | Algorithm Type | Key Specifications | Description | Usage |
|---------------|----------------|---|-------------------|--|
| Symmetric key | AES | <ul style="list-style-type: none">AES_256 | AES symmetric key | Encrypts and decrypts a small amount of data or data keys. |

| Key Type | Algorithm Type | Key Specifications | Description | Usage |
|----------------|----------------|--|------------------------------------|---|
| Asymmetric key | RSA | <ul style="list-style-type: none">• RSA_2048• RSA_3072• RSA_4096 | RSA asymmetric password | Encrypts and decrypts a small amount of data or creates digital signatures. |
| | ECC | <ul style="list-style-type: none">• EC_P256• EC_P384 | Elliptic curve recommended by NIST | Digital signature |

2.1.2 Creating a Key

This section describes how to create a custom key on the KMS console.

Custom keys can be categorized into symmetric keys and asymmetric keys.

Prerequisites

The account has KMS CMKFullAccess or higher permissions.

Constraints

- You can create up to 20 custom keys, excluding default keys.
- Symmetric keys are created using the AES key. The AES-256 key can be used to encrypt and decrypt a small amount of data or data keys.
- Asymmetric keys are created using RSA or ECC algorithms. RSA keys can be used for encryption, decryption, digital signature, and signature verification. ECC keys can be used only for digital signature and signature verification.
- Aliases of default keys end with **/default**. When choosing aliases for your custom keys, do not use aliases ending with **/default**.
- DEW does not limit the number of times that a key can be called.

Scenarios

- Encrypt data in OBS
- Encrypt data in EVS
- Encrypt data in IMS
- Encrypt an RDS DB instance
- Use custom keys to directly encrypt and decrypt small volumes of data.
- DEK encryption and decryption for user applications
- Message authentication code generation and verification
- Asymmetric keys can be used for digital signatures and signature verification.

Creating a Key

Step 1 Log in to the management console.

Step 2 Click  . . .

Step 3 Click **Create Bucket** in the upper right corner.

Step 4 Configure parameters in the **Create Key** dialog box.

- **Alias** is the alias of the key to be created.

 **NOTE**

- You can enter digits, letters, underscores (_), hyphens (-), colons (:), and slashes (/).
- You can enter up to 255 characters.
- **Key Algorithm:** Select a key algorithm. For more information, see [Table 2-2](#).

Table 2-2 Key algorithms supported by KMS

| Key Type | Algorithm Type | Key Specifications | Description | Usage |
|----------------|----------------|--|------------------------------------|---|
| Symmetric key | AES | - AES_256 | AES symmetric key | Encrypts and decrypts a small amount of data or data keys. |
| Asymmetric key | RSA | - RSA_2048 - RSA_3072 - RSA_4096 | RSA asymmetric password | Encrypts and decrypts a small amount of data or creates digital signatures. |
| | ECC | - EC_P256 - EC_P384 | Elliptic curve recommended by NIST | Digital signature |

- **Usage:** Select **SIGN_VERIFY**, **ENCRYPT_DECRYPT**, or **GENERATE_VERIFY_MAC**.
 - For an AES_256 symmetric key, the default value is **ENCRYPT_DECRYPT**.
 - For RSA asymmetric keys, select **ENCRYPT_DECRYPT** or **SIGN_VERIFY**. The default value is **SIGN_VERIFY**.
 - For an ECC asymmetric key, the default value is **SIGN_VERIFY**.

 **NOTE**

The key usage can only be configured during key creation and cannot be modified afterwards.

- (Optional) **Description** is the description of the custom key.

 **NOTE**

You can enter up to 255 characters.

- The **Enterprise Project** parameter needs to be set only for enterprise users. If you are an enterprise user and have created an enterprise project, select the required enterprise project from the drop-down list. The default project is **default**.

If there are no **Enterprise Management** options displayed, you do not need to configure it.

Step 5 (Optional) Add tags to the custom key as needed, and enter the tag key and tag value.

 **NOTE**

- After creating a CMK, you can click the alias of the CMK to go to the CMK details page and add a tag to the CMK.
- The same tag (including tag key and tag value) can be used for different custom keys. However, under the same custom key, one tag key can have only one tag value.
- A maximum of 20 tags can be added for one custom key.
- To delete a tag, click **Delete** next to it.

Step 6 Click **OK**. A message is displayed in the upper right corner of the page, indicating that the key is created successfully.

In the key list, you can view the created keys. The default status of a key is **Enabled**.

----End

Related Operations

- For details about how to upload objects with server-side encryption, see section "Uploading a File with Server-Side Encryption" in *Object Storage Service User Guide*.
- For details about how to encrypt data on EVS disks, see section "Creating an EVS Disk" in *Elastic Volume Service User Guide*.
- For details about how to encrypt private images, see section "Encrypting an Image" in *Image Management Service User Guide*.
- For details about how to encrypt disks for a database instance in RDS, see section "Purchasing an Instance" in the *Relational Database Service User Guide*.
- For details about how to create a DEK and a plaintext-free DEK, see sections "Creating a DEK" and "Creating a Plaintext-Free DEK" in *Data Encryption Workshop API Reference*.
- For details about how to encrypt and decrypt a DEK for a user application, see sections "Encrypting a DEK" and "Decrypting a DEK" in *Data Encryption Workshop API Reference*.

2.1.3 Creating CMKs Using Imported Key Materials

2.1.3.1 Overview

A custom key contains key metadata (key ID, key alias, description, key status, and creation date) and key materials used for encrypting and decrypting data.

- When a user uses the KMS console to create a custom key, the KMS automatically generates a key material for the custom key.
- If you want to use your own key material, you can use the key import function on the KMS console to create a custom key whose key material is empty, and import the key material to the custom key.

Important Notes

- **Security**
You need to ensure that random sources meet your security requirements when using them to generate key materials. When using the import key function, you need to be responsible for the security of your key materials. Save the original backup of the key material so that the backup key material can be imported to the KMS in time when the key material is deleted accidentally.
- **Availability and Durability**
Before importing the key material into KMS, you need to ensure the availability and durability of the key material.
Differences between the imported key material and the key material generated by KMS are shown in [Table 2-3](#).

Table 2-3 Differences between the imported key material and the key material generated by KMS

| Key Material Source | Difference |
|---------------------|--|
| Imported keys | <ul style="list-style-type: none">• You can delete the key material, but cannot delete the custom key and its metadata.• Such keys cannot be rotated.• When importing the key material, you can set the expiration time of the key material. After the key material expires, the KMS automatically deletes the key material within 24 hours, but does not delete the custom key and its metadata. It is recommended that you save a copy of the material on your local device because it may be used for re-import in cases of invalid key materials or key material mis-deletion. |
| Keys created in KMS | <ul style="list-style-type: none">• The key material cannot be manually deleted.• Symmetric keys can be rotated.• You cannot set the expiration time for key material. |

- **Association**
When a key material is imported to a custom key, the custom key is permanently associated with the key material. Other key materials cannot be imported into the custom key.
- **Uniqueness**
If you use the custom key created using the imported key material to encrypt data, the encrypted data can be decrypted only by the custom key that has been used to encrypt the data, because the metadata and key material of the custom key must be consistent.

2.1.3.2 Importing Key Materials

If you want to use your own key materials instead of the KMS-generated materials, you can use the console to import your key materials to KMS. CMKs created using imported materials and KMS-generated materials are managed together by KMS.

This section describes how to import key materials on the KMS console.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click **Import Key**. The **Import Key** dialog box is displayed.
- Step 3** Configure key parameters.
 - **Alias** is the alias of the key to be created.
 - 📖 **NOTE**
 - You can enter digits, letters, underscores (_), hyphens (-), colons (:), and slashes (/).
 - You can enter up to 255 characters.
 - **Key Algorithm**: Select a key algorithm. For more information, see [Table 2-4](#).

Table 2-4 Key algorithms supported by KMS

| Key Type | Algorithm Type | Key Specifications | Description | Usage |
|---------------|----------------|--------------------|-------------------|--|
| Symmetric key | AES | AES_256 | AES symmetric key | Encrypts and decrypts a small amount of data or data keys. |

| Key Type | Algorithm Type | Key Specifications | Description | Usage |
|----------------|----------------|--|------------------------------------|---|
| Asymmetric key | RSA | <ul style="list-style-type: none"> - RSA_2048 - RSA_3072 - RSA_4096 | RSA asymmetric password | Encrypts and decrypts a small amount of data or creates digital signatures. |
| | ECC | <ul style="list-style-type: none"> - EC_P256 - EC_P384 | Elliptic curve recommended by NIST | Digital signature |

- **Usage:** Select **SIGN_VERIFY**, **ENCRYPT_DECRYPT**, or **GENERATE_VERIFY_MAC**.
 - For an AES_256 symmetric key, the default value is **ENCRYPT_DECRYPT**.
 - For RSA asymmetric keys, select **ENCRYPT_DECRYPT** or **SIGN_VERIFY**. The default value is **SIGN_VERIFY**.
 - For an ECC asymmetric key, the default value is **SIGN_VERIFY**.

 **NOTE**

The key usage can only be configured during key creation and cannot be modified afterwards.

- (Optional) **Description** is the description of the custom key.

 **NOTE**

You can enter up to 255 characters.

Step 4 (Optional) Add tags to the custom key as needed, and enter the tag key and tag value.

 **NOTE**

- If a custom key has been created without any tag, you can add a tag to the custom key later if needed. Click the alias of the custom key, choose the **Tags** tab, and click **Add Tag**.
- The same tag (including tag key and tag value) can be used for different custom keys. However, under the same custom key, one tag key can have only one tag value.
- A maximum of 20 tags can be added for one custom key.
- If you want to delete a tag from the tag list when adding multiple tags, you can click **Delete** in the row where the tag to be added is located to delete the tag.

Step 5 Select **I understand the security and durability of using an imported key**, and create a custom key whose key material is empty.

Step 6 Click **Next** to go to the **Download the Import Items** step. Select a key wrapping algorithm based on .

Table 2-5 Key wrapping algorithms

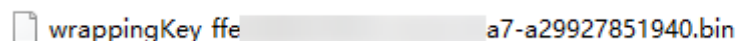
| Algorithm | Description | Configuration |
|--------------------|---|--|
| RSAES_OAEP_SHA_256 | RSA algorithm that uses OAEP and has the SHA-256 hash function | Select an algorithm based on your HSM functions. If the HSMs support the RSAES_OAEP_SHA_256 algorithm, use RSAES_OAEP_SHA_256 to encrypt key materials. |

NOTE

If you stop a key material import process and want to try again, click **Import Key Material** in the row of the required custom key, and import key material in the displayed dialog box.

Step 7 Obtain the wrapping key and import token. If you already have a key material, skip this step.

1. Obtain the wrapping key and import token.
 - Method 1: Click **Download and Continue** to download the wrapping key file, as shown in [Figure 2-1](#).

Figure 2-1 Downloaded fileA screenshot showing a file icon, the text "wrappingKey_ffe", a greyed-out field, and the file extension "a7-a29927851940.bin".

- **wrappingKey_KeyID** is the wrapping key. It is encoded in binary format and used to encrypt the wrapping key of the key material.
- Import token: You do not need to download it. The import wizard automatically transfers the import token. If you close the wizard before completing the import, the token will automatically become invalid.

NOTICE

The wrapping key expires in 24 hours. If the wrapping key is invalid, download it again.

The import wizard automatically transfers the import token. If you close the wizard before completing the import, the token will automatically become invalid. To retry import, open the import wizard again.

- Method 2: Obtain the wrapping key and import token by calling APIs.
 - i. Call the **get-parameters-for-import** API to obtain the wrapping key and import token.
 - o **public_key**: content of the wrapping key (Base-64 encoding) returned after the API call

- **import_token**: content of the import token (Base-64 encoding) returned after the API call

The following example describes how to obtain the wrapping key and import token of a CMK (ID:

43f1ffd7-18fb-4568-9575-602e009b7ee8; algorithm: **RSAES_OAEP_SHA_256**).

- Example request

```
{
  "key_id": "43f1ffd7-18fb-4568-9575-602e009b7ee8",
  "wrapping_algorithm": "RSAES_OAEP_SHA_256"
}
```

- Example response

```
{
  "key_id": "43f1ffd7-18fb-4568-9575-602e009b7ee8",
  "public_key": "public key base64 encoded data",
  "import_token": "import token base64 encoded data",
  "expiration_time": 1501578672
}
```

- ii. Save the wrapping key and convert its format. Only the key material encrypted using the converted wrapping key can be imported to the management console.
 - 1) Copy the content of the wrapping key **public_key**, paste it to a .txt file, and save the file as **PublicKey.b64**.
 - 2) Use OpenSSL to run the following command to perform Base-64 coding on the content of the **PublicKey.b64** file to generate binary data, and save the converted file as **PublicKey.bin**:
openssl enc -d -base64 -A -in PublicKey.b64 -out PublicKey.bin
 - iii. Save the import token, copy the content of the **import_token** token, paste it to a .txt file, and save the file as **ImportToken.b64**.
2. Use the wrapping key to encrypt the key material.

NOTE

After performing this step, you will obtain either of the following files:

Symmetric key scenario: **EncryptedKeyMaterial.bin** (key material)

Asymmetric key scenario: **EncryptedKeyMaterial.bin** (temporary key material) and **out_rsa_private_key.der** (private key ciphertext)

Method 1: Use the downloaded wrapping key to encrypt key materials on your HSM. For details, see the operation guide of your HSM.

Method 2: Use OpenSSL to generate a key material and use the downloaded wrapping key to encrypt the key material.

NOTE

If you need to run the **openssl pkeyutl** command, ensure your OpenSSL version is 1.0.2 or later.

- a. Generate a key material (256-bit symmetric key) and save it as **PlaintextKeyMaterial.bin**.
 - If the AES256 symmetric key algorithm is used, run the following command on the client where the OpenSSL tool has been installed:
openssl rand -out PlaintextKeyMaterial.bin 32

- 1) Generate a hexadecimal AES256 key.
openssl rand -out PlaintextKeyMaterial.bin -hex 32
 - 2) Convert the hexadecimal AES256 key to the binary format.
cat PlaintextKeyMaterial.bin | xxd -r -ps > PlaintextKeyMaterial.bin
 - b. Use the downloaded wrapping key to encrypt the key material and save the encrypted key material as **EncryptedKeyMaterial.bin**.
If the wrapping key was downloaded from the console, replace **PublicKey.bin** in the following command with the wrapping key name *wrappingKey_keyID*.

Table 2-6 Encrypting the generated key material using the downloaded wrapping key

| Wrapping Key Algorithm | Key Material Encryption |
|------------------------|--|
| RSAES_OAEP_SHA_256 | openssl pkeyutl -in PlaintextKeyMaterial.bin -inkey PublicKey.bin -out EncryptedKeyMaterial.bin -keyform der -pubin -encrypt -pkeyopt rsa_padding_mode:oaep -pkeyopt rsa_oaep_md:sha256 |

- 1) Generate a private key.
openssl genrsa -out pkcs1_rsa_private_key.pem 4096
 - 2) Convert the format to PKCS8.
openssl pkcs8 -topk8 -inform PEM -in pkcs1_rsa_private_key.pem -outform pem -nocrypt -out rsa_private_key.pem
 - 3) Convert the PKCS8 format to the DER format.
openssl pkcs8 -topk8 -inform PEM -outform DER -in rsa_private_key.pem -out rsa_private_key.der -nocrypt
 - 4) Use a temporary key material to encrypt the private key.

```
openssl enc -id-aes256-wrap-pad -K $(cat  
0xPlaintextKeyMaterial.bin) -iv A65959A6 -in  
rsa_private_key.der -out out_rsa_private_key.der
```

NOTE

By default, the `-id-aes256-wrap-pad` algorithm is not enabled in OpenSSL. To wrap a key, upgrade OpenSSL to the latest version and patch it first. For details, see FAQs.

Step 8 If you already have the key material, click **Existing Key Material**. The **Import Key Material** page is displayed.

Table 2-7 Parameters for importing key materials (for symmetric keys)

| Parameter | Description |
|--------------|---|
| Key ID | Random ID of a CMK generated during the CMK creation |
| Key material | Import a key material. For example, use the EncryptedKeyMaterial.bin file in Step 7.2.b . |

Table 2-8 Parameters for importing key materials (for asymmetric keys)

| Parameter | Description |
|------------------------|--|
| Key ID | Random ID of a CMK generated during the CMK creation |
| Temporary key material | Import a temporary key material. For example, select the EncryptedKeyMaterial.bin file in Step 7.2.b . |
| Private key ciphertext | Select private key ciphertext. For example, select the out_rsa_private_key.der file in Step 7.2.c . |

Step 9 Click **Next** to go to the **Import Key Token** step. Configure the parameters as described in [Table 2-9](#).

Table 2-9 Parameters for importing a key token

| Parameter | Description |
|------------------|--|
| Key ID | Random ID of a CMK generated during the CMK creation |
| Key import token | Select the import token obtained via API in 12.b . |

| Parameter | Description |
|------------------------------|--|
| Key material expiration mode | <ul style="list-style-type: none">• Key material will never expire: You use this option to specify that key materials will not expire after import.• Key material will expire: You use this option to specify the expiration time of the key materials. By default, key materials expire in 24 hours after import. After the key material expires, the system automatically deletes the key material within 24 hours. Once the key material is deleted, the key cannot be used and its status changes to Pending import. |

Step 10 Click **OK**. When the **Key imported successfully** message is displayed in the upper right corner, the materials are imported.

NOTICE

Key materials can be successfully imported when they match the corresponding CMK ID and token.

Your imported materials are displayed in the list of CMKs. The default status of an imported CMK is **Enabled**.

----End

2.1.3.3 Deleting Key Materials

When importing key materials, you can specify their expiration time. After the key material expires, KMS deletes it, and the status of the custom key changes to **Pending import**. You can manually delete the key materials as needed. The effect of expiration of the key material is the same as that of manual deletion of the key material.

This section describes how to delete imported key materials on the KMS console.

 **NOTE**

- To re-import a deleted key material, ensure the imported material is the same as the deleted one.
- Data encrypted using a CMK cannot be decrypted if the key material of the custom key was deleted. To decrypt the data, re-import the key material.

Prerequisites

- You have imported key materials for a CMK.
- The material source of the CMK is **External**.
- The CMK status is **Enabled** or **Disabled**.

Constraints

- To re-import a deleted key material, ensure the imported material is the same as the deleted one.

- Data encrypted using a CMK cannot be decrypted if the key material of the custom key was deleted. To decrypt the data, re-import the key material.
- After the deletion, the CMK will become unavailable and its status will change to **Pending import**.
- The key materials of asymmetric keys cannot be directly deleted. To delete them, perform the instructions in [Deleting One or More CMKs](#).

Procedure

Step 1 Log in to the management console.

Step 2 In the row containing the target CMK, click **Delete Key Material**.

Step 3 In the displayed dialog box, click **OK**. When **Key material deleted successfully** is displayed in the upper right corner, the key materials are successfully deleted.

After the deletion, the CMK will become unavailable and its status changes to **Pending import**.

----End

2.1.4 Managing CMKs

2.1.4.1 Viewing a CMK

This section describes how to view the information about the custom key on the KMS console, including the key alias, status, ID, and creation time. The status of a key can be **Enabled**, **Disabled**, **Scheduled deletion**, or **Pending import**.

Procedure

Step 1 Log in to the management console.

Step 2 Check the key list.


Table 2-10 Key list parameters

| Parameter | Description |
|-----------|---|
| Alias/ID | Alias of a key and the random ID of a key generated during its creation. NOTE Use this ID as the value of Path if you are creating a custom policy in IAM and have selected Specify resource path for KeyId . |

| Parameter | Description |
|-------------------------|---|
| Status | Status of a CMK, which can be one of the following: <ul style="list-style-type: none"> • Enabled The CMK is enabled. • Disabled The CMK is disabled. • Pending deletion The CMK is scheduled for deletion. • Pending import If your CMK does not have materials, its status is Pending import. |
| Key Algorithm and Usage | Key algorithm selected during key creation and its usage |
| Expiration Time | Expiration time of the key material. When the material expires, the CMK becomes an empty CMK. |
| Origin | Source of key material, which can be one of the following: <ul style="list-style-type: none"> • External The key is imported to the KMS from an external system. • Key Management Service The key is a default key or created in KMS. |
| Enterprise Project | Enterprise project the CMK is used for |
| Operation | Operations you can perform on the key, such as disable, delete, import key material, or cancel deletion. You can also assign keys to projects. |

Step 3 You can click the alias of a key to view its details.

 **NOTE**

To change the alias or description of the CMK, click  next to the value of **Alias** or **Description**.

- A default key (the alias suffix of which is **/default**) does not allow alias and description changes.
- The alias and description of a CMK cannot be changed if the CMK is in **Pending deletion** status.

----End

2.1.4.2 Enabling One or More CMKs

This section describes how to use the KMS console to enable one or more custom keys. Only enabled custom keys can be used to encrypt or decrypt data. A new custom key is in the **Enabled** state by default.

Prerequisites

The custom key you want to enable is in **Disabled** status.

Procedure

Step 1 Log in to the management console.

Step 2 In the row containing the target custom key, click **Enable**.

Step 3 In the displayed dialog box, click **OK** to enable the key.

NOTE

To enable multiple CMKs at a time, select them and click **Enable** in the upper left corner of the list.

----End

2.1.4.3 Disabling One or More CMKs

This section describes how to use the KMS console to disable one or more custom keys, thereby protecting data in urgent cases.

After being disabled, a custom key cannot be used to encrypt or decrypt any data. Before using a disabled CMK to encrypt or decrypt data, you must enable it by following instructions in [Enabling One or More CMKs](#).

Prerequisites

The CMK you want to disable is in **Enabled** status.

Constraints

- Default keys created by KMS cannot be disabled.
- A disabled CMK is still billable. It will stop incurring charges if it is deleted.

Procedure

Step 1 Log in to the management console.

Step 2 In the row containing the target CMK, click **Disable**.

Step 3 In the displayed dialog box, select **I understand the impact of disabling keys**, and click **OK**.

NOTE

To disable multiple CMKs at a time, select them and click **Disable** in the upper left corner of the list.

----End

2.1.4.4 Deleting One or More CMKs

Before deleting the CMK, confirm that it is not in use and will not be used. You can check the key usage in either of the following ways:

- Check the CMK permission to determine its possible usage scope. For details, see [Querying a Grant](#).
- Check audit logs to determine the actual usage.

Prerequisites

- The key to be deleted is in **Enabled**, **Disabled**, or **Pending import** status.

Constraints

- A key will not be deleted until its scheduled deletion period expires. You can set the period to a value within the range 7 to 1096 days.
Before the specified deletion date, you can cancel the deletion if you want to use the CMK. Once the scheduled deletion has taken effect, the CMK will be deleted permanently and you will not be able to decrypt data encrypted by the CMK. Exercise caution when performing this operation.
- Default keys created by KMS cannot be scheduled for deletion.
- A CMK in pending deletion status does not incur charges. If you cancel deletion, the charging resumes from the time when the CMK was scheduled to be deleted.

Procedure

Step 1 Log in to the management console.

Step 2 In the row containing the target CMK, click **Delete** in the **Operation** column.

Step 3 On the key deletion dialog box, enter the deletion delay time.

NOTE

- A key will not be deleted until its scheduled deletion period expires. You can set the period to a value within the range 7 to 1096 days. Before the specified deletion date, you can cancel the deletion if you want to use the CMK.
- A CMK in pending deletion status does not incur charges. If you cancel deletion, the charging resumes from the time when the CMK was scheduled to be deleted.

Step 4 Select **I understand the impact of deleting keys** and click **YesOK**.

----End

NOTE

To schedule the deletion of multiple CMKs at a time, select them and click **Delete** in the upper left corner of the list.

2.1.4.5 Canceling the Scheduled Deletion of One or More CMKs

This section describes how to use the KMS console to cancel the scheduled deletion of one or more custom keys prior to deletion execution. After the cancellation, the key is in **Disabled** status.

Prerequisites

The CMK for which you want to cancel the scheduled deletion is in **Pending deletion** status.

Procedure

Step 1 Log in to the management console.

Step 2 In the row containing the target CMK, click **Cancel Deletion**.

Step 3 In the displayed dialog box, click **OK** to cancel the scheduled deletion.

- If a key is created on the KMS console, the status of the key changes to **Disabled** after its scheduled deletion is canceled. For details about how to enable the key, see [Enabling One or More CMKs](#).
- If the CMK is created using imported materials, its status becomes **Disabled** after the cancellation. To enable the CMK, see [Enabling One or More CMKs](#).
- If the CMK is created using imported materials and no key materials have been imported for it, its status becomes **Pending import** after the cancellation. To use the CMK, perform [Creating CMKs Using Imported Key Materials](#).

NOTE

To cancel the deletion of multiple CMKs at a time, select them and click **Cancel Deletion** in the upper left corner of the list.

----End

2.1.4.6 Adding a Key to a Project

Enterprise Project is a cloud governance platform that matches the organizational structure and service management model of your company. It helps you manage enterprise projects, resources, personnel, finance, and applications in the cloud based on the hierarchical organization structure (companies, departments, and projects) and project service structure.

If you have enabled enterprise project management, you can add specified custom keys to enterprise projects on the KMS console.

Constraints

- The enterprise project management function has been enabled.
If you did not enable the enterprise project management function, the **Enterprise Project** option is not displayed on the console by default, and you cannot add keys to a project.
- The enterprise project of default keys cannot be changed.

Procedure

Step 1 Log in to the management console.

Step 2 In the row containing the target key, click **Add to Project**.

NOTE

If you are a non-enterprise user, the **Add to Project** option is not displayed in the operation column.

Step 3 Select a project.

Step 4 Click .

----End

2.1.5 Searching for a Key

This section describes how to search for a custom key by specifying attributes on the KMS page.

Procedure

Step 1 Log in to the management console.

Step 2 Click the search bar and select the criteria for filtering keys. Search for a key by specifying attributes.

NOTE

- You can search for keys by key alias, ID, status, creation time, usage, material source, and enterprise project.

----End

2.1.6 Using the Online Tool to Encrypt and Decrypt Small-Size Data

This section describes how to use the online tool to encrypt or decrypt small-size data (4 KB or smaller) on the KMS console.

Prerequisites

The custom key is in **Enabled** status.

Constraints

- Default keys cannot be used to encrypt or decrypt such data with the tool.
- Asymmetric keys cannot be used to encrypt or decrypt such data with the tool.
- You can call an API to use a default key to encrypt or decrypt small volumes of data. For details, see the *Data Encryption Workshop API Reference*.
- Use the current CMK to encrypt the data.
- Exercise caution when you delete a CMK. The online tool cannot decrypt data if the CMK used for encryption has been deleted.

Encrypting Data

Step 1 Log in to the management console.

Step 2 Click the alias of a custom key to view its details, and go to the online tool for data encryption and decryption.

Step 3 Click **Encrypt**. In the text box on the left, enter the data to be encrypted.

Step 4 Click **Execute**. Ciphertext of the data is displayed in the text box on the right.

 **NOTE**

- Use the current CMK to encrypt the data.
- You can click **Clear** to clear the entered data.
- You can click **Copy to Clipboard** to copy the ciphertext and save it in a local file.

----End

Decrypting Data

Step 1 Log in to the management console.

Step 2 You can click any non-default key in **Enabled** status to go to the encryption and decryption page of the online tool.

Step 3 Click **Decrypt**. In the text box on the left, enter the data to be decrypted.

 **NOTE**

- The tool will identify the original encryption CMK and use it to decrypt the data.
- If the key has been deleted, the decryption will fail.

Step 4 Click **Execute**. Plaintext of the data is displayed in the text box on the right.

 **NOTE**

- You can click **Copy to Clipboard** to copy the plaintext and save it in a local file.
- Enter the plaintext on the console, the text will be encoded to Base64 format before encryption.

The decryption result returned via API will be in Base64 format. Perform Base64 decoding to obtain the plaintext entered on the console.

----End

2.1.7 Managing Tags

2.1.7.1 Adding a Tag

Tags are used to identify keys. You can add tags to custom keys so that you can classify custom keys, trace them, and collect their usage status according to the tags.

Constraints

Tags cannot be added to default keys.

Procedure

Step 1 Log in to the management console.

Step 2 Click the alias of the target custom key to view its details.

Step 3 Click **Tags** to go to the tag management page.

Step 4 Click **Add Tag**. In the **Add Tag** dialog box, enter the tag key and tag value.

 **NOTE**

If you want to delete a tag to be added when adding multiple tags, you can click **Delete** in the row where the tag to be added is located to delete the tag.

Table 2-11 Tag parameters

| Parameter | Description | Value | Example Value |
|-----------|--|---|---------------|
| Tag key | <p>Name of a tag.</p> <p>The same tag (including tag key and tag value) can be used for different custom keys. However, under the same custom key, one tag key can have only one tag value.</p> <p>A maximum of 20 tags can be added for one custom key.</p> | <ul style="list-style-type: none"> • Mandatory. • The tag key must be unique for the same custom key. • 128 characters limit. • The value cannot start or end with a space. • The following character types are allowed: <ul style="list-style-type: none"> - English - Numbers - Space - Special characters: _-@ | cost |
| Tag value | Value of the tag | <ul style="list-style-type: none"> • This parameter can be empty. • 255 characters limit. • The following character types are allowed: <ul style="list-style-type: none"> - English - Numbers - Space - Special characters: _-@ | 100 |

Step 5 Click **OK** to complete.

----End

2.1.7.2 Modifying Tag Values

This section describes how to modify tag values on the KMS console.

Procedure

- Step 1** Log in to the management console.
 - Step 2** Click the alias of the target custom key to view its details.
 - Step 3** Click **Tags** to go to the tag management page.
 - Step 4** Click **Edit** of the target tag, and the **Edit Tag** dialog box is displayed.
 - Step 5** In the **Edit Tag** dialog box, enter a tag value, and click **OK** to complete the editing.
- End

2.1.7.3 Deleting Tags

This section describes how to delete tags on the KMS console.

Procedure

- Step 1** Log in to the management console.
 - Step 2** Click the alias of the target custom key to view its details.
 - Step 3** Click **Tags** to go to the tag management page.
 - Step 4** Click **Delete** of the target tag, and the **Delete Tag** dialog box is displayed.
 - Step 5** In the **Delete Tag** dialog box, click **Confirm**.
- End

2.1.8 Rotating CMKs

2.1.8.1 About Key Rotation

Purpose of Key Rotation

Keys that are widely or repeatedly used are insecure. To enhance the security of encryption keys, you are advised to periodically rotate keys and change their key materials.

The purposes of key rotation are:

- To reduce the amount of data encrypted by each key.
A key will be insecure if it is used to encrypt a huge number of data. The amount of data encrypted a key refers to the total number of bytes or messages encrypted using the key.
- To enhance the capability of responding to security events.
In your initial system security design, you shall design the key rotation function and use it for routine O&M, so that it will be at hand when an emergency occurs.
- To enhance the data isolation capability.

The ciphertext data generated before and after key rotation will be isolated. You can identify the impact scope of a security event based on the key involved and take actions accordingly.

Key Rotation Methods

You can use either of the following key rotation methods:

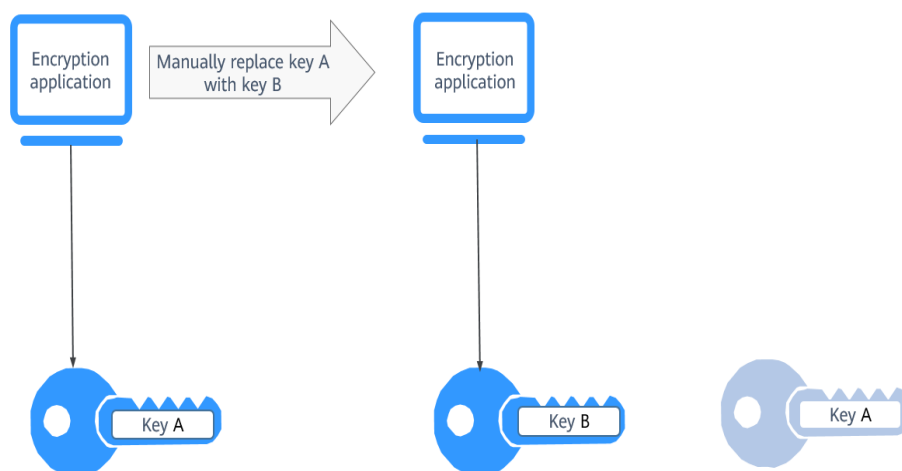
- Manual key rotation

Method 1: Create a key B to replace the currently used key A.

Method 2: Modify the key A and use it.

Take OBS as an example. To manually rotate a key, create a new custom key on the KMS console. Replace the old custom key with the new one on the OBS console.

Figure 2-2 Manual key rotation



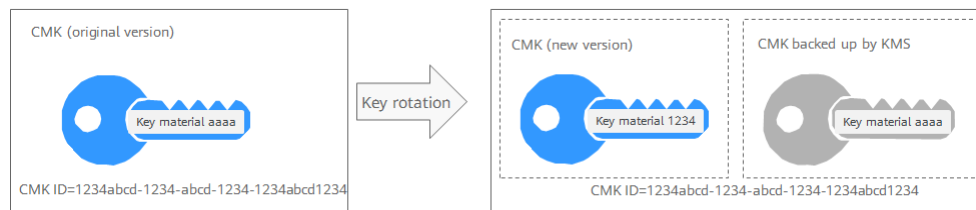
- Automatic key rotation

KMS automatically rotates keys based on the configured rotation period (365 days by default). The system automatically generates a new key to replace the key in use. Automatic key rotation only changes the key material of a CMK. The logical attributes of the key will not change, including its key ID, alias, description, and permissions.

Automatic key rotation has the following characteristics:

- a. Enable rotation for an existing custom key. KMS will automatically generate new key materials for the custom key.
- b. Data is not re-encrypted in an automatic key rotation. The DEK generated using the CMK is not automatically rotated, and data that has been encrypted using the CMK will not be encrypted again. If a DEK has been leaked, automatic rotation cannot contain the impact of the leakage.

Figure 2-3 Key rotation



NOTE

KMS retains all versions of a custom key, so that you can decrypt any ciphertext encrypted using the custom key.

- KMS uses the latest version of the custom key to encrypt data.
- When decrypting data, KMS uses the custom key version that was used to encrypt the data.

Rotation Modes

Table 2-12 Key rotation modes

| Key Type | Rotation Mode |
|--------------------------------|---|
| Default key | Cannot be rotated. |
| Custom key | Keys can be rotated automatically or manually, depending on the key algorithm type. <ul style="list-style-type: none"> • Symmetric key: Can be automatically or manually rotated. • Asymmetric key: Can only be manually rotated. |
| Disabled CMK | Disabled CMKs are not rotated. KMS keeps their rotation status unchanged. After a custom key is enabled, if it has been used for longer than the rotation period, KMS will immediately rotate keys. If the custom key has been used for shorter than the rotation period, KMS will implement the original rotation plan. For more information, see Disabling One or More CMKs . |
| CMKs in pending deletion state | KMS does not rotate CMKs in pending deletion status. After you cancel the deletion of a CMK, the previous key rotation status will be restored. If the custom key has been used for longer than the rotation period, KMS will immediately rotate keys. If the CMK has been used for shorter than the rotation period, KMS will implement the original rotation plan. For more information, see Scheduling the Deletion of One or More Keys . |

 **NOTE**

You can check the rotation details on the **Rotation Policy** page, including the last rotation time and number of rotations.

2.1.8.2 Enabling Key Rotation

This section describes how to enable rotation for a key on the KMS console.

By default, automatic key rotation is disabled for a custom key. Every time you enable key rotation, KMS automatically rotates custom keys based on the rotation period you set.

Prerequisites

- The key is enabled.
- The **Origin** of the key is **KMS**.
- Only symmetric keys can be rotated.

Constraints


- A disabled custom key is never rotated, even if rotation is enabled for it. KMS resumes rotation when this custom key is enabled. If you enable this custom key after one rotation period has passed, KMS will rotate it within 24 hours.
- Only CMKs can be rotated.

Procedure

Step 1 Log in to the management console.

Step 2 Click the alias of the target custom key to view its details.


Step 3 Click the **Rotation Policy** tab. The rotation switch is displayed.

Step 4 Click  to enable key rotation.

Step 5 In the **Enable Rotation Policy** dialog box, set the rotation period and click **OK**.

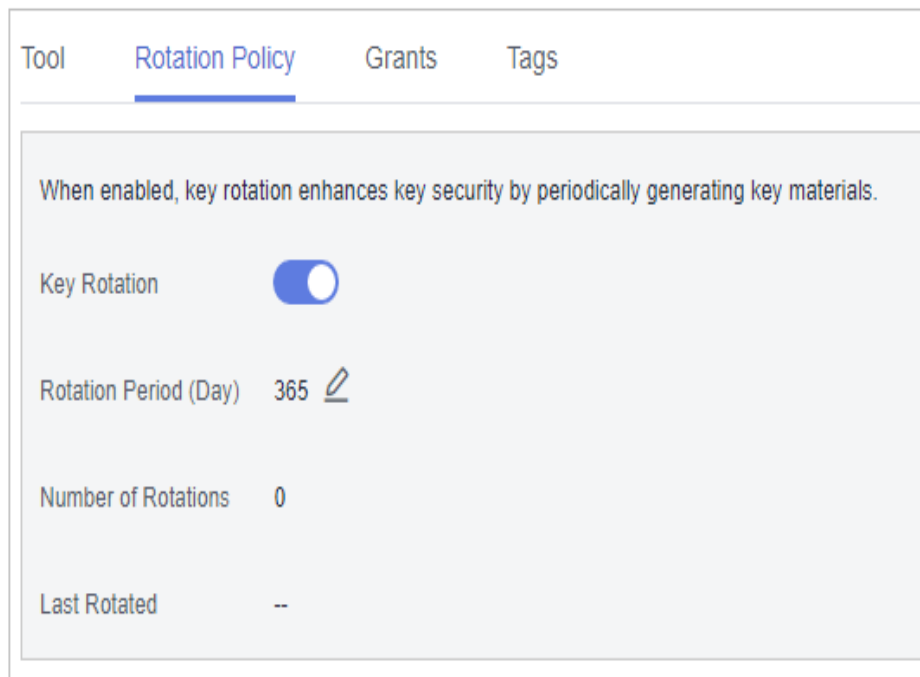
- Set the rotation period (unit: day) to an integer in the range 30 to 365. The default value is **365**.
- After the setting takes effect, the new rotation period starts.
- Configure the period based on how often a custom key is used. If it is frequently used, configure a short period. Otherwise, set a long one.

 **NOTE**

- A disabled custom key is never rotated, even if rotation is enabled for it.
- KMS resumes rotation when this custom key is enabled. If you enable this custom key after one rotation period has passed, KMS will rotate it within 24 hours.
- You can click  to change the rotation period. After the period is changed, KMS rotates the key by the new period.

Step 6 Check rotation details, as shown in the following figure.

Figure 2-4 Key rotation details



NOTE

You can click to change the rotation period. After the period is changed, KMS rotates the key by the new period.

----End

2.1.8.3 Disabling Key Rotation

This section describes how to disable rotation for a key on the KMS console.

Prerequisites

- The key is enabled.
- The **Origin** of the key is **KMS**.
- Key rotation has been enabled.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click the alias of a symmetric key.
- Step 3** Click **Rotation Policy** and the dialog box is displayed.
- Step 4** Click to disable key rotation.
- Step 5** In the displayed confirmation dialog box, click **OK**.

Step 6 Check the rotation status.

----End

2.1.9 Managing a Grant

2.1.9.1 Creating a Grant

You can create grants for other IAM users or accounts to use the custom key. You can create a maximum of 100 grants on a custom key.

Prerequisites

- You have obtained the ID of the grantee (user to whom permissions are to be authorized).
- The target custom key is in **Enabled** status.

Constraints

- The owner of a custom key can create a grant for the custom key on the KMS console or by calling APIs. The IAM users or accounts who have the grant creation permission assigned by the owner of the custom key can create grants for the custom key only by calling APIs.
- A maximum of 100 grants can be created for a custom key.
- Only users and accounts can be authorized. Agency authorization is not supported.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click the alias of the target custom key to go to its details page and create a grant on it.
- Step 3** Click the **Grants** tab.
- Step 4** Click **Create Grant**. The **Create Grant** dialog box is displayed.
- Step 5** In the dialog box that is displayed, enter the ID of the user to be authorized and select permissions to be granted. For more information, see [Table 2-13](#).

NOTICE

A grantee can perform the authorized operations only by calling the necessary APIs. For details, see the *Data Encryption Workshop API Reference*.

Table 2-13 Parameter description

| Parameter | Description | Example Value |
|----------------|--|--|
| User or Tenant | <p>Whether a user or an account is authorized.</p> <ul style="list-style-type: none"> • User User ID: Enter the IAM user ID. To obtain the ID, click the username in the upper right corner of the page, choose My Credentials. Choose API Credentials from the navigation pane, and copy the value of IAM User ID. After the authorization is complete, the IAM user can use the specified keys. • Account Account ID: Enter the IAM user ID. To obtain the ID, click the username in the upper right corner of the page, choose My Credentials. Choose API Credentials from the navigation pane and copy the value of Account ID. After the authorization is complete, all IAM users under the account can use the specified keys. | d9a6b2bdaedd 4ba586cabe63 72d1b312 |

| Parameter | Description | Example Value |
|------------|---|---------------|
| Operations | <p>The following permissions can be authorized:</p> <p>NOTE</p> <ul style="list-style-type: none"> • You can create multiple grants on a custom key to provide different permissions to the same user. The user's permissions on the custom key are the combination of all the grants. • This parameter cannot be left blank. • Selecting only Create Grant is not allowed. • Create Data Key Without Plaintext • Create Data Key • Encrypt Data Key • Decrypt Data Key • Query Key Information • Create Grant • Retire Grant <ul style="list-style-type: none"> – A grantee can retire a grant if the grantee does not need that permission. – If, before retiring a grant, the grantee has granted the permission to another user, that user's permission will not be affected by the grant retirement. • Encrypt Data • Decrypt Data | - |

Step 6 Click **OK**. When message **Grant created successfully** is displayed in the upper right corner, the grant has been created.

In the list of grants, you can view the grant name, grant type, grantee ID, granted operation, and creation time of the grant.

----End

2.1.9.2 Querying a Grant

You can view the details about a custom key grant on the KMS console, such as the grant ID, grantee user ID, granted operation, and creation time.

Prerequisites

You have created a grant.

Procedure

Step 1 Log in to the management console.

Step 2 Click the alias of the target custom key to view its details.

Step 3 Click **Grant** to view the grant information of the current custom key. [Table 2-14](#) describes the parameters.

Table 2-14 Parameter description

| Parameter | Description |
|--------------------|---|
| Grant ID | Randomly generated unique identification of a grant |
| Granted To | Whether permissions are granted to a user or account. |
| Granted Operations | Authorized operations (such as Create Data Key) on the custom key |
| Created | Time when the grant is created |
| Operation | Operations that can be performed on a grant. For example, you can revoke a grant. |

----End

2.1.9.3 Revoking a Grant

You can revoke a grant on the KMS console in either of the following scenarios:

- A grantee does not need the custom key grant. (The grantee can either tell the user who has created the grant to revoke the grant or call the necessary API to revoke the grant directly.)
- You do not want the grantee to have the grant.

When a grant is revoked, the grantee does not have the corresponding permission anymore. However, if the grantee has created the same grant to another user, permission of that user will not be affected.

This section describes how to revoke a grant on the KMS console.

Prerequisites

You have created a grant.

Procedure

Step 1 Log in to the management console.

Step 2 Click the alias of the target custom key to view its details.

Step 3 In the row of a grantee, click **Revoke Grant**.

Step 4 In the dialog box that is displayed, click **OK**. If **Grant *grant ID* revoked successfully** is displayed in the upper right corner, the grant has been revoked.

----End

2.2 Cloud Secret Management Service

2.2.1 Creating a Secret

2.2.1.1 Creating a Shared Secret

This section describes how to create a secret on the CSMS console.

You can create a secret and store its value in its initial version, which is marked as **SYSCURRENT**.

Constraints

- A user can create a maximum of 200 secrets.
- By default, the default key **csms/default** created by CSMS is used as the encryption key of the current secret. You can also create a user-defined symmetric key and use a user-defined encryption key on the KMS console.

Creating a Secret

Step 1 Log in to the management console.

Step 2 In the navigation pane, choose **Cloud Secret Management Service**.

Step 3 Click **Create Secret**. Configure parameters in the **Create Secret** dialog box,. For details, see [Table 2-15](#).

Table 2-15 Secret parameters

| Parameter | Description |
|--------------------|---|
| Type | Secret type. The default value is Shared secret . |
| Secret Name | Secret name NOTE Only letters, digits, hyphens (-), and underscores (_) are supported. |
| Enterprise Project | Enterprise project that the secret is to be bound to |
| Secret Value | Secret key/value pair and the plaintext secret to be encrypted |
| Description | Description of a secret |

| Parameter | Description |
|--------------------|--|
| KMS Encryption Key | <p>Select the default key csms/default or a custom key created on KMS.</p> <p>NOTE</p> <ul style="list-style-type: none"> CSMS encrypts private keys using the encryption key provided by KMS. When you use the KMS encryption function of the key pair, KMS creates a default key csms/default for you to use. For details about the custom keys created on KMS, see . |

Step 4 Click **Next** and set the rotation period.

Step 5 Click **Next** and confirm the creation information.

Step 6 Click **OK**.

In the secret list, you can view the created secrets. The default status of a secret is **Enabled**.

----End

2.2.2 Managing Secrets

2.2.2.1 Viewing a Secret

This section describes how to check secret names, statuses, and creation time on the CSMS console. The secret status can be **Enabled** or **Pending deletion**.

Procedure

Step 1 Log in to the management console.

Step 2 Check the secret list. For more information, see [Table 2-16](#).

Table 2-16 Secret list parameters

| Parameter | Description |
|--------------------|---|
| Secret Name/ID | Secret name |
| Status | Status of a secret. The value can be Enabled or Pending deletion . |
| Type | Secret type. The value can be Shared secret . |
| Created | Time when the secret is created |
| Enterprise Project | Enterprise project that the secret is to be bound to |
| Operation | You can manage secrets in the Operation column, for example, download secret backup, delete secrets, and cancel secret deletion. |

Step 3 Click a secret to view its details.

 **NOTE**

- You can click **Edit** to modify the encryption key and description of a secret.
- You can click **Refresh** to refresh secret information.

----End

2.2.2.2 Deleting a Secret

Before deleting a secret, confirm that it is not in use and will not be used.

Prerequisites

The secret to be deleted is in **Enabled** state.

Constraints

- A secret will not be deleted until its scheduled deletion period expires. You can set the period to a value within the range 7 to 30 days. Before the specified deletion date, you can cancel the deletion if you want to use the secret. If the scheduled deletion period of a secret expires, the secret will be deleted and cannot be restored.
- If you delete a secret immediately, you can restore it using the secret backup that you have downloaded in advance. Exercise caution when performing this operation.

Deleting a Secret

Step 1 Log in to the management console.

Step 2 In the row of a secret, click **Delete**.

Step 3 In the displayed dialog box, click **Schedule deletion** or **Delete now**. If you want to delete the secret in a specific time, set **Schedule deletion**.

Step 4 Click **OK** to delete the tag.

 **NOTE**

- A secret will not be deleted until its scheduled deletion period expires. You can set the period to a value within the range 7 to 30 days. Before the specified deletion date, you can cancel the deletion if you want to use the secret. If the scheduled deletion period of a secret expires, the secret will be deleted and cannot be restored.
- If you delete a secret immediately, you can restore it using the secret backup that you have downloaded in advance. Exercise caution when performing this operation.

----End

2.2.3 Managing Secret Versions

2.2.3.1 Saving and Viewing Secret Values

This section describes how to save and view secret values on the CSMS console.

You can create a new version of a secret to encrypt and keep a new secret value. By default, The latest secret version in **SYSCURRENT** state. The previous version is in the **SYSPREVIOUS** state.

Constraints

- A secret can have up to 20 versions.
- Secret versions are numbered v1, v2, v3, and so on based on their creation time.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click a secret name to go to the details page.
- Step 3** In the **Version List** area, click **Add Secret Version**
- Step 4** Click **OK**. A message is displayed in the upper right corner of the page, indicating that the value is added successfully.

View the latest secret value in the secret version list.
- Step 5** In the **Version List** area, locate the target secret version, click **View Secret** in the **Operation** column.
- Step 6** View the secret value and click **OK**.

----End

2.2.3.2 Managing Secret Version Statuses

This section describes how to add, change, and delete secret version statuses.

Secret values are encrypted and stored in secret versions. A version can have multiple statuses. Versions without any statuses are regarded as deprecated and can be automatically deleted by CSMS.

Constraints

- The initial version is marked by the **SYSCURRENT** status tag.
- You can mark a version with a tag created in the service or a custom tag. A version can have multiple status tags, but a status tag can be used for only one version. For example, if you add the status tag used by version A to version B, the tag will be moved from version A to version B.
- A secret can have up to 12 version statuses. A status can be used for only one version.
- **SYSCURRENT** and **SYSPREVIOUS** are preconfigured statuses and cannot be deleted.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click a secret name to go to the details page.

Step 3 In the **Version List** area, click **Manage Status** in the **Operation** column.

Step 4 In the **Manage Status** dialog box, add, change, or delete the status of a secret version.

- Adding a version status

In the **Manage Status** dialog box, click **Add** and enter a status name. Click **OK**.

 **NOTE**

A secret can have up to 12 version statuses. A status can be used for only one version.

- Updating the version status

In the **Manage Status** dialog box, click **Change** and select an existing version status. Click **OK**.

- Deleting the version status

In the **Manage Status** dialog box, click **Delete** and select a version status. Click **OK**.

 **NOTE**

SYSCURRENT and **SYSPREVIOUS** are preconfigured statuses and cannot be deleted.

----End

2.2.4 Managing Tags

2.2.4.1 Adding a Tag

Tags are used to identify secrets. You can easily classify and track secrets using tags.

Procedure

Step 1 Log in to the management console.

Step 2 Click a secret name to go to the details page.

Step 3 In the **Tags** area, click **Add Tag**. In the **Add Tag** dialog box, enter the tag key and tag value. [Table 2-17](#) describes the parameters.

 **NOTE**

- To delete a tag, click **Delete** next to it.

Table 2-17 Tag parameters

| Parameter | Description | Remarks |
|-----------|---|--|
| Tag key | Tag name. The tag keys of a secret cannot have duplicate values. A tag key can be used for multiple secrets. A secret can have up to 20 tags. | <ul style="list-style-type: none">• Mandatory.• The tag key must be unique for the same custom key.• characters limit.• The value cannot start or end with a space.• The following character types are allowed:<ul style="list-style-type: none">- Chinese- English- Numbers- Space- Special characters: _-@ |
| Tag value | Value of the tag | <ul style="list-style-type: none">• Optional• characters limit.• The following character types are allowed:<ul style="list-style-type: none">- Chinese- English- Numbers- Space- Special characters: _-@ |

Step 4 Click **OK**.

----End

2.2.4.2 Searching for a Secret by Tag

This section describes how to search for a secret by tag in a project on the CSMS console.

Prerequisites

Tags have been added.

Procedure


Step 1 Log in to the management console.

Step 2 Click **Search by Tag** to show the search box.

Step 3 In the search box, enter or select a tag key and a tag value.

Step 4

 **NOTE**

- Multiple tags can be added for one search. A maximum of 20 tags can be added for one search. Each search result meets all the search criteria.
- To delete a tag from the search criteria, click  next to the tag.
- You can click **Reset** to reset the search criteria.

----End

2.2.4.3 Modifying a Tag Value

This section describes how to modify tag values on the CSMS console.

Procedure

Step 1 Log in to the management console.

Step 2 Click a secret name to go to the details page.

Step 3 In the **Tags** area, click **Edit**.

Step 4 In the **Edit Tag** dialog box, enter a tag value and click **OK**.

----End

2.2.4.4 Deleting a Tag

This section describes how to delete tags on the CSMS console.

Procedure

Step 1 Log in to the management console.

Step 2 Click a secret name to go to the details page.

Step 3 In the **Tags** area, click **Delete**.

Step 4 In the **Delete Tag** dialog box, click **Confirm**.

----End

2.2.5 Creating an Event

This section describes how to create an event on the **Events** page.

When creating an event, you can set the event type to new **Version creation**, **Version expiry**, **Secret rotation**, and **Secret deletion**.

Constraints

You can create a maximum of 30 events.

Procedure

- Step 1** Log in to the management console.
- Step 2** In the navigation pane on the left, choose **Cloud Secret Management Service > Events**. The **Events** page is displayed.
- Step 3** Click **Create Event** in the upper right corner. The page for creating an event is displayed, as shown in [Creating an event](#).


Figure 2-5 Creating an event

Create Event

Event Name

Status Enabled Disabled

Topic Type/Name ▼

 What you use beyond the free quota given by SMN will be billed. [Pricing details](#)

Event Type





| <input type="checkbox"/> | Event Type | Level | Object | Description |
|--------------------------|-----------------|---|----------------|--|
| <input type="checkbox"/> | Version cre... |  Normal | Secret | Triggered when a version of a secret is created. |
| <input type="checkbox"/> | Version exp... |  Warning | Secret vers... | Triggered when a secret version expires. (Triggered only once for each expiry). |
| <input type="checkbox"/> | Secret rotat... |  Normal | Secret | Triggered when a secret is rotated. (Currently, only RDS secrets can be automatic... |
| <input type="checkbox"/> | Secret dele... |  Warning | Secret | Triggered when a secret is deleted. |

Table 2-18 Parameters for creating an event

| Parameter | Description |
|-----------------|--|
| Event Name | Name of the event to be created. NOTE Only letters, digits, hyphens (-), and underscores (_) are supported. |
| Status | The options are Enabled and Disabled . By default, Enabled is selected. |
| Topic Type/Name | Topic type: SMN is selected by default. Topic name: name of the topic created in SMN. NOTE For details about how to create a custom topic type or name, see . |
| Event Type | Supported event types, including Version creation , Version expiry , Secret rotation , and Secret deletion . |

Step 4 Click **OK**.

Step 5 View the created event in the event list, as shown in **Figure 2-6**. The default event status is **Enabled**.

Figure 2-6 Event list

| Event Name | Status | Subscription | Topic Type/Name | Created | Operation |
|------------|---------|---|-----------------|---------------------------------|---------------|
| Demo | Enabled | Version creation Secret rotation Secret ... | SMN | Jun 07, 2023 16:16:47 GMT+08:00 | Edit Delete |
| demo01 | Enabled | Version creation Version expiry Secret ... | SMN | Jun 12, 2023 11:37:29 GMT+08:00 | Edit Delete |
| demo010 | Enabled | Version creation Version expiry Secret ... | SMN | Jun 12, 2023 14:38:29 GMT+08:00 | Edit Delete |
| demo02 | Enabled | Version creation Version expiry Secret ... | SMN | Jun 12, 2023 11:38:19 GMT+08:00 | Edit Delete |
| demo03 | Enabled | Secret rotation | SMN | Jun 12, 2023 11:38:32 GMT+08:00 | Edit Delete |
| demo04 | Enabled | Secret deletion | SMN | Jun 12, 2023 11:38:50 GMT+08:00 | Edit Delete |
| demo05 | Enabled | Version creation | SMN | Jun 12, 2023 11:39:09 GMT+08:00 | Edit Delete |
| demo06 | Enabled | Secret deletion | SMN | Jun 12, 2023 11:39:20 GMT+08:00 | Edit Delete |
| demo09 | Enabled | Secret rotation | SMN | Jun 12, 2023 11:39:57 GMT+08:00 | Edit Delete |
| lylent | Enabled | Version creation Version expiry Secret ... | SMN | Jul 05, 2023 11:12:37 GMT+08:00 | Edit Delete |

----End

2.2.6 Managing Events

2.2.6.1 Viewing Events

This section describes how to view the information about the created events on the **Events** page, including the event name, status, subscription event type, topic type/name, and creation time.

Procedure

Step 1 Log in to the management console.

Step 2 In the navigation pane on the left, choose **Cloud Secret Management Service > Events**. The **Events** page is displayed.

Step 3 In the event list, view the event information. **Table 2-19** describes the parameters in the event list.

Figure 2-7 Event list

| Event Name | Status | Subscription | Topic Type/Name | Created | Operation |
|------------|---------|---|-----------------|---------------------------------|---------------|
| Demo | Enabled | Version creation Secret rotation Secret ... | SMN | Jun 07, 2023 16:16:47 GMT+08:00 | Edit Delete |
| demo01 | Enabled | Version creation Version expiry Secret ... | SMN | Jun 12, 2023 11:37:29 GMT+08:00 | Edit Delete |
| demo010 | Enabled | Version creation Version expiry Secret ... | SMN | Jun 12, 2023 14:38:29 GMT+08:00 | Edit Delete |
| demo02 | Enabled | Version creation Version expiry Secret ... | SMN | Jun 12, 2023 11:38:19 GMT+08:00 | Edit Delete |
| demo03 | Enabled | Secret rotation | SMN | Jun 12, 2023 11:38:32 GMT+08:00 | Edit Delete |
| demo04 | Enabled | Secret deletion | SMN | Jun 12, 2023 11:38:50 GMT+08:00 | Edit Delete |
| demo05 | Enabled | Version creation | SMN | Jun 12, 2023 11:39:09 GMT+08:00 | Edit Delete |
| demo06 | Enabled | Secret deletion | SMN | Jun 12, 2023 11:39:20 GMT+08:00 | Edit Delete |
| demo09 | Enabled | Secret rotation | SMN | Jun 12, 2023 11:39:57 GMT+08:00 | Edit Delete |
| lylent | Enabled | Version creation Version expiry Secret ... | SMN | Jul 05, 2023 11:12:37 GMT+08:00 | Edit Delete |

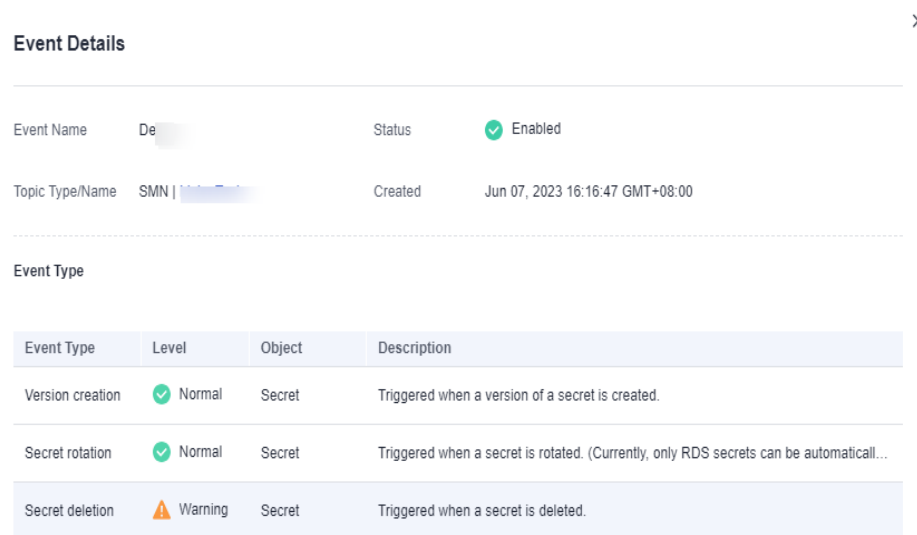
Table 2-19 Parameters in the event list

| Parameter | Description |
|------------|------------------|
| Event Name | Name of an event |

| Parameter | Description |
|-----------------|---|
| Status | Event status, including: <ul style="list-style-type: none"> • Enabled • Disabled |
| Subscription | Event type selected during event creation. The options are as follows: <ul style="list-style-type: none"> • Version creation • Version expiry • Secret rotation • Secret deletion |
| Topic Type/Name | Topic type: SMN is selected by default. Topic name: name of the topic created in SMN. |
| Created | Time when the event is created |
| Operation | You can edit or delete an event in the Operation column. |

Step 4 Click the name of an event name to view the event details, as shown in [Figure 2-8](#).

Figure 2-8 Event details



----End

2.2.6.2 Editing an Event

This section describes how to modify an event type on the **Events** page.

Procedure

- Step 1** Log in to the management console.
- Step 2** In the navigation pane on the left, choose **Cloud Secret Management Service > Events**. The **Events** page is displayed.
- Step 3** Click **Edit** in the **Operation** column of the target event. The **Edit Event** page is displayed.
- Step 4** Select the target event type, as shown in [Figure 2-9](#).

Figure 2-9 Editing an event

Event Type

| <input type="checkbox"/> | Event Type | Level | Object | Description |
|-------------------------------------|------------------|---|----------------|---|
| <input checked="" type="checkbox"/> | Version creation |  Normal | Secret | Triggered when a version of a secret is created. |
| <input type="checkbox"/> | Version expiry |  Warning | Secret version | Triggered when a secret version expires. (Triggered only once for each expiry). |
| <input checked="" type="checkbox"/> | Secret rotation |  Normal | Secret | Triggered when a secret is rotated. (Currently, only RDS secrets can be automatic |
| <input checked="" type="checkbox"/> | Secret deletion |  Warning | Secret | Triggered when a secret is deleted. |

- Step 5** Click **OK**.

----End

2.2.6.3 Enabling an Event

This section describes how to enable a disabled event on the **Events** page.

Prerequisites

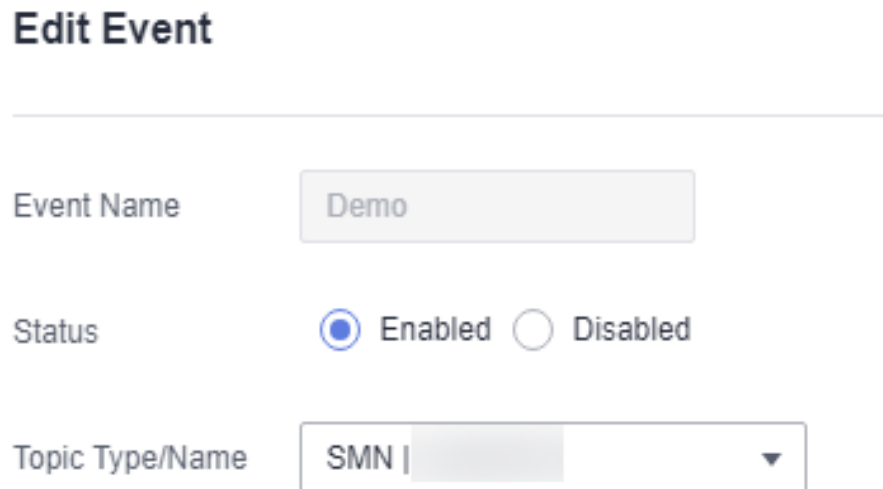
The event to be enabled must be in the **Disabled** state.

Procedure

- Step 1** Log in to the management console.
- Step 2** In the navigation pane on the left, choose **Cloud Secret Management Service > Events**. The **Events** page is displayed.
- Step 3** Click **Edit** in the **Operation** column of the target event. The **Edit Event** page is displayed.

Step 4 Select **Enabled** for **Status**.

Figure 2-10 Enabling an event



The screenshot shows a form titled "Edit Event". It contains three fields: "Event Name" with the value "Demo", "Status" with radio buttons for "Enabled" (selected) and "Disabled", and "Topic Type/Name" with a dropdown menu showing "SMN".

Step 5 Click **OK**. A message is displayed in the upper right corner of the page, indicating that the event status is updated successfully.

----End

2.2.6.4 Disabling an Event

This section describes how to disable an enabled event on the **Events** page.

Prerequisites

The event to be disabled must be in the **Enabled** state.

Procedure

- Step 1** Log in to the management console.
- Step 2** In the navigation pane on the left, choose **Cloud Secret Management Service > Events**. The **Events** page is displayed.
- Step 3** Click **Edit** in the **Operation** column of the target event. The **Edit Event** page is displayed.
- Step 4** Select **Disabled** for **Status**.

Figure 2-11 Disabling an event

Edit Event

| | |
|-----------------|---|
| Event Name | <input type="text" value="Demo"/> |
| Status | <input type="radio"/> Enabled <input checked="" type="radio"/> Disabled |
| Topic Type/Name | <input type="text" value="SMN "/> ▼ |

Step 5 Click **OK**. A message is displayed in the upper right corner of the page, indicating that the event is disabled successfully.

----End

2.2.6.5 Deleting an Event

This section describes how to delete a created event on the **Events** page. Before deleting an event, ensure that the event is no longer used.

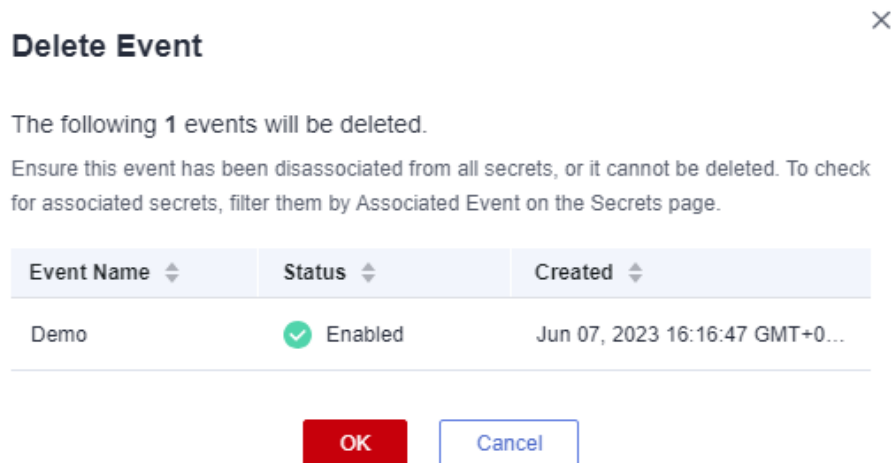
Constraints

Event notifications can be deleted only after all associated secrets have been canceled. If the associated secret is not canceled, the deletion will fail.

Procedure

- Step 1** Log in to the management console.
- Step 2** In the navigation pane on the left, choose **Cloud Secret Management Service > Events**. The **Events** page is displayed.
- Step 3** Click **Delete** in the **Operation** column of the target event. The **Delete Event** dialog box is displayed.

Figure 2-12 Deleting an event



Step 4 Click **OK**.

----End

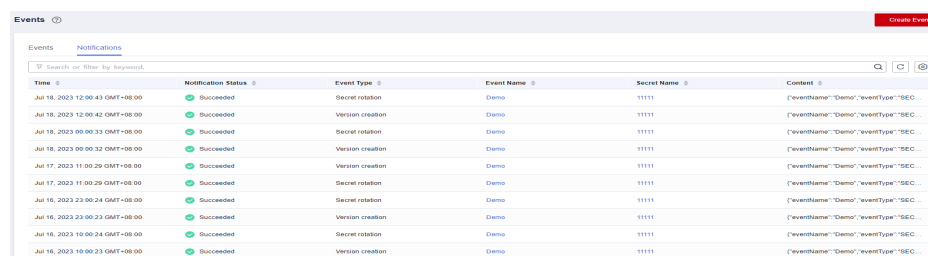
2.2.7 Viewing Notifications

This section describes how to view the event notifications.

Procedure

- Step 1** Log in to the management console.
- Step 2** In the navigation pane on the left, choose **Cloud Secret Management Service > Events**. The **Events** page is displayed.
- Step 3** Click the **Notifications** tab. The page for viewing notifications is displayed, as shown in [Figure 2-13](#).

Figure 2-13 Viewing notifications



Step 4 On the **Notifications** tab page, you can view the changes made to the secrets of the associated events.

----End

2.3 Auditing Logs

2.3.1 Operations supported by CTS

The tables in this section describe the DEW operations supported by CTS.

Table 2-20 DEW operations supported by CTS

| Operation | Resource Type | Trace Name |
|---------------------------------|---------------|-------------------------------|
| Create a key | cmk | createKey |
| Create a DEK | cmk | createDataKey |
| Create a plaintext-free DEK | cmk | createDataKeyWithoutPlaintext |
| Enable a key | cmk | enableKey |
| Disable a key | cmk | disableKey |
| Encrypt a DEK | cmk | encryptDatakey |
| Decrypt a DEK | cmk | decryptDatakey |
| Schedule key deletion | cmk | scheduleKeyDeletion |
| Cancel scheduled key deletion | cmk | cancelKeyDeletion |
| Generate random numbers | rng | genRandom |
| Modify a key alias | cmk | updateKeyAlias |
| Modify key description | cmk | updateKeyDescription |
| Prompt risks about CMK deletion | cmk | deleteKeyRiskTips |
| Import key materials | cmk | importKeyMaterial |
| Delete key materials | cmk | deleteImportedKeyMaterial |
| Create a grant | cmk | createGrant |
| Retire a grant | cmk | retireGrant |
| Revoke a grant | cmk | revokeGrant |
| Encrypt data | cmk | encryptData |
| Decrypt data | cmk | decryptData |
| Add a tag | cmk | createKeyTag |
| Delete a tag | cmk | deleteKeyTag |
| Add tags in batches | cmk | batchCreateKeyTags |
| Delete tags in batches | cmk | batchDeleteKeyTags |
| Enable key rotation | cmk | enableKeyRotation |

| Operation | Resource Type | Trace Name |
|--------------------------------------|---------------|--------------------------------|
| Modify key rotation interval | cmk | updateKeyRotationInterval |
| Disable key rotation | cmk | disableKeyRotation |
| Create a secret | csms | createSecret |
| Update a secret | csms | updateSecret |
| Delete a secret | csms | forceDeleteSecret |
| Schedule the deletion of a secret | csms | scheduleDelSecret |
| Cancel the scheduled secret deletion | csms | restoreSecretFromDeletedStatus |
| Create a secret status | csms | createSecretStage |
| Update a secret status | csms | updateSecretStage |
| Delete a secret status | csms | deleteSecretStage |
| Create a secret version | csms | createSecretVersion |
| Download a secret backup | csms | backupSecret |
| Restore a secret backup | csms | restoreSecretFromBackupBlob |

2.3.2 Querying Real-Time Traces

Scenarios

After you enable CTS and the management tracker is created, CTS starts recording operations on cloud resources. After a data tracker is created, the system starts recording operations on data in OBS buckets. CTS stores operation records generated in the last seven days.

This section describes how to query and export operation records of the last seven days on the CTS console.

- [Viewing Real-Time Traces in the Trace List](#)

Viewing Real-Time Traces in the Trace List


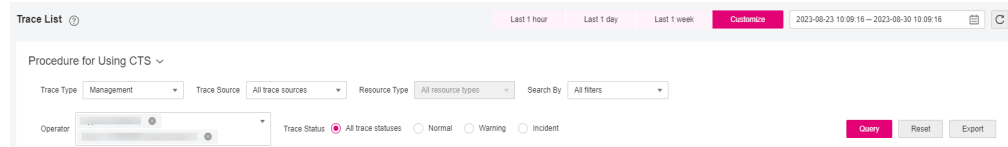


1. Log in to the management console.
2. Click  in the upper left corner and choose **Management & Deployment > Cloud Trace Service**. The CTS console is displayed.
3. Choose **Trace List** in the navigation pane on the left.
4. Set filters to search for your desired traces, as shown in [Figure 2-14](#). The following filters are available:


Figure 2-14 Filters



- **Trace Type, Trace Source, Resource Type, and Search By:** Select a filter from the drop-down list.
 - If you select **Resource ID** for **Search By**, specify a resource ID.
 - If you select **Trace name** for **Search By**, specify a trace name.
 - If you select **Resource name** for **Search By**, specify a resource name.
 - **Operator:** Select a user.
 - **Trace Status:** Select **All trace statuses, Normal, Warning, or Incident.**
 - Time range: You can query traces generated during any time range in the last seven days.
 - Click **Export** to export all traces in the query result as a CSV file. The file can contain up to 5000 records.
5. Click **Query**.
 6. On the **Trace List** page, you can also export and refresh the trace list.
 - Click **Export** to export all traces in the query result as a CSV file. The file can contain up to 5000 records.
 - Click  to view the latest information about traces.
 7. Click  on the left of a trace to expand its details.



8. Click **View Trace** in the **Operation** column. The trace details are displayed.



```
{
  "request": "",
  "trace_id": " ",
  "code": "200",
  "trace_name": "createDockerConfig",
  "resource_type": "dockerlogincmd",
  "trace_rating": "normal",
  "api_version": "",
  "message": "createDockerConfig, Method: POST Url=/v2/manage/utls/secret. Reason:",
  "source_ip": " ",
  "domain_id": " ",
  "trace_type": "ApiCall",
  "service_type": "SWR",
  "event_type": "system",
  "project_id": " ",
  "response": "",
  "resource_id": "",
  "tracker_name": "system",
  "time": "Nov 16, 2023 10:54:04 GMT+08:00",
  "resource_name": "dockerlogincmd",
  "user": {
    "domain": {
      "name": " ",
      "id": " "
    }
  }
}
```

9. For details about key fields in the trace structure, see section "Trace References" > "Trace Structure" and section "Trace References" > "Example Traces".

2.4 Permission Control

2.4.1 Creating a User and Authorizing the User the Permission to Access DEW

This section describes IAM's fine-grained permissions management for your DEW resources. With IAM, you can:

- Create IAM users for employees based on the organizational structure of your enterprise. Each IAM user has its own security credentials to access DEW resources.
- Grant users only the permissions required to perform a task.
- Entrust a cloud account or cloud service to perform professional, efficient O&M on your DEW resources.

If your account does not need individual IAM users, skip this chapter.

This section describes the procedure for granting permissions (see [Figure 2-15](#)).

Prerequisites

Before granting permissions to a user group, you need to understand the available DEW permissions, and grant permissions based on the real-life scenario. The following tables describe the permissions supported in DEW.

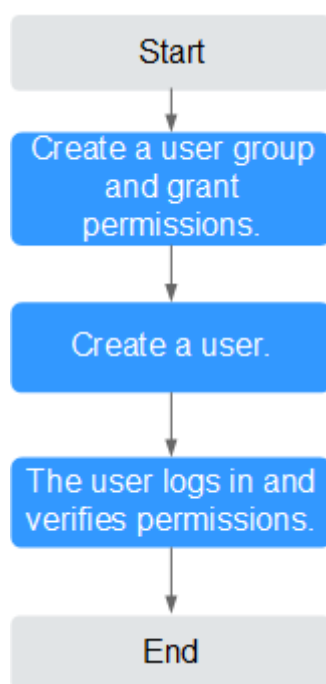
For the permissions of other services, see .

Table 2-21 DEW permissions

| Role/Policy | Description | Type |
|-------------------|--|--------|
| KMS Administrator | Administrator permissions for the encryption key | Role |
| KMS CMK Admin | All permissions for the encryption keys | Policy |

Authorization Process

Figure 2-15 Authorizing the DEW access permission to a user



1. Create a user group on the IAM console and grant the user group the **KMS CMK Admin** permission (indicating full permissions for keys).
2. Create a user on the IAM console and add the user to the user group created in [1](#).
3. and verify permissions.

Log in to the console as newly created user, and verify that the user only has read permissions for DEW.

2.4.2 Creating a Custom DEW Policy

Custom policies can be created as a supplement to the system policies of . For details about the actions supported by custom policies, see "Permissions Policies" in *Data Encryption Workshop API Reference*.

You can create custom policies in either of the following ways:

- Visual editor: You can select policy configurations without the need to know policy syntax.
Custom KMS policy parameters:
 - **Select service:** Select **Key Management Service**.
 - **Select action:** Set it as required.
 - **(Optional) Select resource:** Set **Resources** to **Specific** and **KeyId** to **Specify resource path**. In the dialog box that is displayed, set **Path** to the ID generated when the key was created. For details about how to obtain the ID, see "Viewing a CMK".
- JSON: Edit JSON policies from scratch or based on an existing policy. For details about how to create custom policies, see . This section describes typical DEW custom policies.

Example Custom Policies of DEW

- Example: authorizing users to create and import keys

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:cmk:create",
        "kms:cmk:getMaterial",
        "kms:cmkTag:create",
        "kms:cmkTag:batch",
        "kms:cmk:importMaterial"
      ]
    }
  ]
}
```

- Example: authorizing users to use keys

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:dek:crypto",
        "kms:cmk:get",
        "kms:cmk:crypto",
        "kms:cmk:generate",
        "kms:cmk:list"
      ]
    }
  ]
}
```

- Example: multi-action policy

A custom policy can contain actions of multiple services that are all of the global or project-level type. The following is a policy with multiple statements:

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "rds:task:list"
      ]
    }
  ],
}
```

```
{  
  "Effect": "Allow",  
  "Action": [  
    "kms:dek:crypto",  
    "kms:cmk:get",  
    "kms:cmk:crypto",  
    "kms:cmk:generate",  
    "kms:cmk:list"  
  ]  
}
```

3 FAQs

3.1 KMS Related

3.1.1 What Is Key Management Service?

KMS is a secure, reliable, and easy-to-use cloud service that helps users create, manage, and protect keys in a centralized manner.

It uses Hardware Security Modules (HSMs) to protect keys. All keys are protected by root keys in HSMs to avoid key leakage. The HSM module meets the FIPS 140-2 Level 3 security requirements.

It also controls access to keys and records all operations on keys with traceable logs. In addition, it provides use records of all keys, meeting your audit and regulatory compliance requirements.

3.1.2 What Is a Customer Master Key?

A Customer Master Key (CMK) is a Key Encryption Key (KEK) created by a user on KMS. It is used to encrypt and protect DEKs. One CMK can be used to encrypt one or more DEKs.

CMKs are categorized into custom keys and default keys.

- Custom keys
Keys created or imported by users on the KMS console.
- Default keys

When a user uses KMS for encryption in a cloud service for the first time, the cloud service automatically creates a key with the alias suffix **/default**.

You can use the management console to query but cannot disable or schedule the deletion of Default Master Keys.

Table 3-1 Default Master Keys

| Alias | Cloud Service |
|--------------|--|
| obs/default | Object Storage Service (OBS) |
| evs/default | Elastic Volume Service (EVS) |
| ims/default | Image Management Service (IMS) |
| csms/default | Cloud Secret Management Service (CSMS) |

3.1.3 What Is a Default Key?

A default key is automatically created by another cloud service using KMS, such as Object Storage Service (OBS). The alias of a default key ends with **/default**.

You can use the management console to query but cannot disable or schedule the deletion of default keys.

Default keys are hosted for free, and are charged based on the number of the API requests for them. If API requests exceed the free limit, the excess part will be charged.

Table 3-2 Default Master Keys

| Alias | Cloud Service |
|--------------|--|
| obs/default | Object Storage Service (OBS) |
| evs/default | Elastic Volume Service (EVS) |
| ims/default | Image Management Service (IMS) |
| vbs/default | Volume Backup Service (VBS) |
| kps/default | Key Pair Service (KPS) |
| csms/default | Cloud Secret Management Service (CSMS) |

NOTE

A default key is automatically created when a user employs the KMS encryption function for the first time in another cloud service.

3.1.4 What Are the Differences Between a Custom Key and a Default Key?

The following table describes the differences between a custom key and a default key.

Table 3-3 Differences between a custom key and a default key

| Item | Definition | Difference |
|-------------|--|--|
| Custom key | A Key Encryption Key (KEK) created using KMS. The key is used to encrypt and protect DEKs. A custom key can be used to encrypt multiple DEKs. | <ul style="list-style-type: none">• It can be disabled and scheduled for deletion.• It is billed per use after the being created or imported. |
| Default key | Automatically generated by the system when you use KMS to encrypt data in another cloud service for the first time. The suffix of the key is / default . Example: evs/default | <ul style="list-style-type: none">• It cannot be disabled or scheduled for deletion. |

3.1.5 What Is a Data Encryption Key?

A data encryption key (DEK) is used to encrypt data.

3.1.6 Why Cannot I Delete a CMK Immediately?

The decision to delete a CMK should be considered with great caution. Before deletion, confirm that the CMK's encrypted data has all been migrated. As soon as the CMK is deleted, you will not be able to decrypt data with it. Therefore, KMS offers a user-specified period of 7 to 1096 days for the deletion to finally take effect. On the scheduled day of deletion, the CMK will be permanently deleted. However, prior to the scheduled day, you can still cancel the pending deletion. This is a means of precaution within KMS.

3.1.7 Which Cloud Services Can Use KMS for Encryption?

Object Storage Service (OBS), Elastic Volume Service (EVS), and Image Management Service (IMS) can use KMS for encryption.

Table 3-4 List of cloud services that use KMS encryption

| Service Name | Description |
|-----------------------------------|---|
| Object Storage Service (OBS) | <p>You can upload objects to and download them from Object Storage Service (OBS) in common mode or server-side encryption mode. When you upload objects in encryption mode, data is encrypted at the server side and then securely stored on OBS in ciphertext. When you download encrypted objects, the data in ciphertext is decrypted at the server side and then provided to you in plaintext. OBS supports the server-side encryption with KMS-managed keys (SSE-KMS) mode. In SSE-KMS mode, OBS uses the keys provided by KMS for server-side encryption.</p> <p>For details about how to upload objects to OBS in SSE-KMS mode, see the <i>Object Storage Service Console Operation Guide</i>.</p> |
| Elastic Volume Service (EVS) | <p>If you enable the encryption function when creating an EVS disk, the disk will be encrypted with the DEK generated by using your CMK. Data stored in the EVS disk will be automatically encrypted.</p> <p>For details about how to use the encryption function of EVS, see <i>Elastic Volume Service User Guide</i>.</p> |
| Image Management Service (IMS) | <p>When creating a private image using an external image file, you can enable the private image encryption function and select a CMK provided by KMS to encrypt the image.</p> <p>For details about how to use the private image encryption function of Image Management Service (IMS), see <i>Image Management Service User Guide</i>.</p> |
| Scalable File Service (SFS) | <p>When creating a file system on SFS, the CMK provided by KMS can be selected to encrypt the file system, so that files stored in the file system are automatically encrypted.</p> <p>For details about how to use the file system encryption function of SFS, see <i>Scalable File Service User Guide</i>.</p> |
| Relational Database Service (RDS) | <p>When purchasing a database instance, you can enable the disk encryption function of the database instance and select a CMK created on KMS to encrypt the disk of the database instance. Enabling the disk encryption function will enhance data security.</p> <p>For details about how to use the disk encryption function of RDS, see <i>Relational Database Service User Guide</i>.</p> |
| Document Database Service (DDS) | <p>When purchasing a DDS instance, you can enable the disk encryption function of the instance and select a CMK created on KMS to encrypt the disk of the instance. Enabling the disk encryption function will enhance data security.</p> <p>For details about how to use the disk encryption function of DDS, see <i>Document Database Service User Guide</i>.</p> |

3.1.8 How Do Cloud Services Use KMS to Encrypt Data?

NOTE

Envelope encryption is an encryption method that enables DEKs to be stored, transmitted, and used in "envelopes" of CMKs. As a result, CMKs do not directly encrypt and decrypt data.

When users download the data from the cloud, the cloud service uses the CMK specified by KMS to decrypt the ciphertext DEK, use the decrypted DEK to decrypt data, and then provide the decrypted data for users to download.

3.1.9 What Are the Benefits of Envelope Encryption?

Envelope encryption is the practice of encrypting data with a DEK and then encrypting the DEK with a root key that you can fully manage. In this case, CMKs are not required for encryption or decryption.

Benefits:

- Advantages over CMK encryption in KMS
 - Users can use CMKs to encrypt and decrypt data on the KMS console or by calling KMS APIs.
 - A CMK can encrypt and decrypt data no more than 4 KB. An envelope can encrypt and decrypt larger volumes of data.
 - Data encrypted using envelopes does not need to be transferred. Only the DEKs need to be transferred to the KMS server.
- Advantages over encryption by using cloud services
 - Security
 - Data transferred to the cloud for encryption is exposed to risks such as interception and phishing.
 - During envelope encryption, KMS uses Hardware Security Modules (HSMs) to protect keys. All CMKs are protected by root keys in HSMs to avoid key leakage.
 - Trustworthiness
 - You will worry about data security on the cloud. It is also difficult for cloud services to prove that they never misuse or disclose such data.
 - If you choose envelope encryption, KMS will control access to keys and record all usages of and operations on keys with traceable logs, meeting your audit and regulatory compliance requirements.
 - Performance and cost
 - To encrypt or decrypt data using a cloud service, you have to send the data to the encryption server and receive the processed data. This process seriously affects your service performance and incurs high costs.
 - Envelope encryption allows you to generate DEKs online by calling KMS cryptographic algorithm APIs, and to encrypt a large amount of local data with the DEKs.

3.1.10 Is There a Limit on the Number of Custom Keys That I Can Create on KMS?

Yes.

You can create a maximum of 100 custom keys, including those in enabled, disabled, and pending deletion states. Default keys are not included.

3.1.11 Can I Export a CMK from KMS?

No.

To ensure CMK security, users can only create and use CMKs in KMS.

3.1.12 Can I Decrypt My Data if I Permanently Delete My Custom Key?

No.

If you have permanently deleted your custom key, the data encrypted using it cannot be decrypted. Before the scheduled deletion date of the custom key, you can cancel the scheduled deletion.

If the custom key is created using imported key material and only the key material is deleted, you can import the local backup of the key material to the custom key and reclaim the user data. If the key material is not backed up locally, user data cannot be reclaimed.

3.1.13 How Do I Use the Online Tool to Encrypt or Decrypt Small Volumes of Data?

You can use the online tool to encrypt or decrypt data in the following procedures:

Encrypting Data

Step 1 Log in to the management console.

Step 2 Click the alias of a custom key to view its details, and go to the online tool for data encryption and decryption.

Step 3 Click **Encrypt**. In the text box on the left, enter the data to be encrypted.

Step 4 Click **Execute**. Ciphertext of the data is displayed in the text box on the right.

NOTE

- Use the current CMK to encrypt the data.
- You can click **Clear** to clear the entered data.
- You can click **Copy to Clipboard** to copy the ciphertext and save it in a local file.

----End

 **NOTE**

Enter the plaintext on the console, the text will be encoded to Base64 format before encryption.

The decryption result returned via API will be in Base64 format. Perform Base64 decoding to obtain the plaintext entered on the console.

Decrypting Data

Step 1 Log in to the management console.

Step 2 You can click any non-default key in **Enabled** status to go to the encryption and decryption page of the online tool.

Step 3 Click **Decrypt**. In the text box on the left, enter the data to be decrypted.

 **NOTE**

- The tool will identify the original encryption CMK and use it to decrypt the data.
- If the key has been deleted, the decryption will fail.

Step 4 Click **Execute**. Plaintext of the data is displayed in the text box on the right.

 **NOTE**

- You can click **Copy to Clipboard** to copy the plaintext and save it in a local file.
- Enter the plaintext on the console, the text will be encoded to Base64 format before encryption.

The decryption result returned via API will be in Base64 format. Perform Base64 decoding to obtain the plaintext entered on the console.

----End

3.1.14 Can I Update CMKs Created by KMS-Generated Key Materials?

No.

Keys created using KMS-generated materials cannot be updated. You can only use KMS to create new CMKs to encrypt and decrypt data.

3.1.15 When Should I Use a CMK Created with Imported Key Materials?

- If you do not want to use KMS-generated key materials, you can import your own key materials to create a CMK. Such a CMK allows deletion of only the key materials when you do not need it. In addition, when you find that the key materials are mis-deleted, you can import the same materials to the CMK.
- You can also import off-cloud key materials to KMS when you want to use the same keys on and off the cloud. This practice has proved useful when users migrate local encrypted data onto cloud.

3.1.16 What Types of Keys Can I Import?

You can import 256-bit symmetric keys.

3.1.17 What Should I Do When I Accidentally Delete Key Materials?

You can import the backup key materials from your local device again.

NOTICE

Before importing key materials, you are advised to back up the materials. The materials to be re-imported must be consistent with the mis-deleted materials.

3.1.18 How Are Default Keys Generated?

Default keys are automatically generated.

When a user uses KMS for encryption in a cloud service for the first time, the cloud service automatically creates a key with the alias suffix **/default**.

You can use the management console to query but cannot disable or schedule the deletion of Default Master Keys.

Table 3-5 Default Master Keys

| Alias | Cloud Service |
|--------------|--|
| obs/default | Object Storage Service (OBS) |
| evs/default | Elastic Volume Service (EVS) |
| ims/default | Image Management Service (IMS) |
| csms/default | Cloud Secret Management Service (CSMS) |

3.1.19 What Should I Do If I Do Not Have the Permissions to Perform Operations on KMS?

Symptom

A message indicating lack of permissions is displayed when you attempt to perform operations on keys, such as view, create, or import keys.

Possible Causes

Your account is not associated with the required KMS system policies.

Solution

Step 1 Check whether your account has been associated with **KMS Administrator** and **KMS CMKFullAccess** policies.

For details about how to check your user groups and permissions, see .

If your account has been associated with required KMS system policies, go to [Step 2](#).

Step 2 Associate your account with required system policies.

- For details about how to add administrator permissions, see .
- For details about how to add a custom policy, see .

----End

3.1.20 Why Can't I Wrap Asymmetric Keys by Using `-id-aes256-wrap-pad` in OpenSSL?

Symptom

By default, the `-id-aes256-wrap-pad` algorithm is not enabled in OpenSSL. To wrap a key, upgrade OpenSSL to the latest version and patch it first.

Solution

Use bash commands to create a local copy of the existing OpenSSL. You do not need to delete or modify the default OpenSSL client installation configurations.

Step 1 Switch to the **root** user.

```
sudo su -
```

Step 2 Run the following command and record the OpenSSL version:

```
openssl version
```

Step 3 Run the following commands to create the **/root/build** directory. This directory will be used to store the latest OpenSSL binary file.

```
mkdir $HOME/build
```

```
mkdir -p $HOME/local/ssl
```

```
cd $HOME/build
```

Step 4 Download the latest OpenSSL version from <https://www.openssl.org/source/>.

Step 5 Download and decompress the binary file.

Step 6 Replace **openssl-1.1.1d.tar.gz** with the latest OpenSSL version downloaded in [step 4](#).

```
curl -O https://www.openssl.org/source/openssl-1.1.1d.tar.gz
```

```
tar -zxf openssl-1.1.1d.tar.gz
```

Step 7 Use the **gcc** tool to patch the version, and compile the downloaded binary file.

```
yum install patch make gcc -y
```

 **NOTE**

If you are using a version other than OpenSSL-1.1.1d, you may need to change the directory and commands used, or this patch may not work properly.

Step 8 Run the following commands:

```
sed -i "/BIO_get_cipher_ctx(benc, &ctx);/a\ EVP_CIPHER_CTX_set_flags(ctx,  
EVP_CIPHER_CTX_FLAG_WRAP_ALLOW);" $HOME/build/openssl-1.1.1d/apps/enc.c
```

Step 9 Run the following commands to compile the OpenSSL **enc.c** file:

```
cd $HOME/build/openssl-1.1.1d/  
./config --prefix=$HOME/local --openssldir=$HOME/local/ssl  
make -j$(grep -c ^processor /proc/cpuinfo)  
make install
```

Step 10 Configure the environment variable **LD_LIBRARY_PATH** to ensure that required libraries are available for OpenSSL. The latest version of OpenSSL has been dynamically linked to the binary file in the **\$HOME/local/ssl/lib/** directory, and cannot be directly executed in shell.

Step 11 Create a script named **openssl.sh** to load the **\$HOME/local/ssl/lib/** path before running the binary file.

```
cd $HOME/local/bin/  
echo -e '#!/bin/bash \nenv LD_LIBRARY_PATH=$HOME/local/lib/ $HOME/  
local/bin/openssl "$@"' > ./openssl.sh
```

Step 12 Run the following command to configure an execute bit on the script:

```
chmod 755 ./openssl.sh
```

Step 13 Run the following command to start the patched OpenSSL version:

```
$HOME/local/bin/openssl.sh  
----End
```

3.1.21 Key Algorithms Supported by KMS

Table 3-6 Key algorithms supported by KMS

| Key Type | Algorithm Type | Key Specifications | Description | Usage |
|----------------|----------------|--|-------------------------|---|
| Symmetric key | AES | <ul style="list-style-type: none">AES_256 | AES symmetric key | Encrypts and decrypts a small amount of data or data keys. |
| Asymmetric key | RSA | <ul style="list-style-type: none">RSA_2048RSA_3072RSA_4096 | RSA asymmetric password | Encrypts and decrypts a small amount of data or creates digital signatures. |

| Key Type | Algorithm Type | Key Specifications | Description | Usage |
|----------|----------------|---|------------------------------------|-------------------|
| | ECC | <ul style="list-style-type: none">EC_P256EC_P384 | Elliptic curve recommended by NIST | Digital signature |

3.1.22 What Should I Do If KMS Failed to Be Requested and Error Code 401 Is Displayed?

Symptom

An error is reported when KMS is requested or the cloud service encryption function is enabled.

Error information: **httpcode=401,code=APIGW.0301,Msg=Incorrect IAM authentication information: current ip:xx.xx.xx.xx refused**

Possible Causes

Access control is configured in IAM.


By default, IAM allows access from any IP addresses. If you configure ACL, the IP addresses and network segments out of the specified range cannot access KMS or use the cloud encryption feature.

Solution

- To access KMS through the cloud service console (for example, for OBS encryption purposes), allow access from network segments 10.0.0.0/8, 11.0.0.0/8, and 26.0.0.0/8.
- To call KMS via API, allow access from the source IP addresses.

Allowing Access from Specific IP Addresses

Step 1 Log in to the management console.

Step 2 Click  on the left of the page and choose **Management & Governance > Identity and Access Management**. The **Users** page is displayed.

Step 3 Choose **Security Settings** and click the **ACL** tab. Check whether **IP Address Ranges** and **IPv4 CIDR Blocks** are properly configured.

NOTE

The source IP address you use must be specified on both the **Console Access** and **API Access** tabs.

----End

3.1.23 What Is the Relationship Between the Ciphertext and Plaintext Returned by the encrypt-data API?

The basic length of the ciphertext returned by the encrypt-data API is 124 bytes. The ciphertext consists of multiple fields, including the key ID, encryption algorithm, key version, and ciphertext digest.

The plaintext has 16 bytes in each block. A block with fewer than 16 bytes will be padded. Ciphertext length = $124 + \text{Ceil}(\text{plaintext length}/16) \times 16$. The conversion result is encoded using Base64.

Take 4-byte plaintext input as an example. The calculation result is $124 + \text{Ceil}(4/16) \times 16 = 140$. The 140 bytes are converted into 188 bytes after Base64 encoding.

NOTE

Ceil is a round-up function. $\text{Ceil}(a) = 1$. The value range of a is $(0,1]$.

3.1.24 How Does KMS Protect My Keys?

The mechanism of KMS prevents anyone from accessing your keys in plaintext. KMS relies on hardware security modules (HSMs) that safeguard the confidentiality and integrity of your keys. Plaintext KMS keys are always encrypted by HSMs and are never stored on any disk. These keys are only utilized within the volatile memory of the HSMs for as long as necessary to perform the cryptographic operation you have requested.

3.2 Credential Related

3.2.1 Why Cannot I Delete the Version Status of a Secret?

SYSCURRENT and **SYSPREVIOUS** are preconfigured statuses and cannot be deleted.

A Change History

| Released On | Description |
|-------------|---|
| 2023-12-30 | This is the seventh official release. Added the secret management function. <ul style="list-style-type: none">• Added section "CSMS".• Added section "CSMS" in the user guide.• Added section "Why Cannot I Delete the Version Status of a Secret?". |
| 2023-06-30 | This is the sixth official release. <ul style="list-style-type: none">• Added section "Auditing Logs". |
| 2023-05-30 | This is the fifth official release. <ul style="list-style-type: none">• Added section "Personal Data Protection Mechanism".• Added descriptions about key algorithms and key wrapping algorithms in section "Functions".• Added section "Permission Control". |
| 2023-03-25 | This is the fourth official release. <ul style="list-style-type: none">• Added section "What Is DEW?".• The service name is changed to Data Encryption Workshop (DEW). |

| Released On | Description |
|-------------|--|
| 2022-10-25 | This is the third official release. <ul style="list-style-type: none">• Added descriptions about tags in section "Functions".• Added section "Advantages".• Added the description about Scalable File Service (SFS) in section "How to Use KMS".• Added descriptions about enterprise projects and the section "Adding a Key to a Project."• Added section "What Are the Differences Between a Custom Key and a Default Key?".• Added section "Can I Export a CMK from KMS?". |
| 2020-10-29 | This is the second official release. Modified the allowed maximum number of tag values in section "Creating a Key". |
| 2020-09-20 | This is the first official release. |