Key Management Service

Product Introduction

Product Introduction

What is KMS

Key Management Service (KMS) is a managed service for you to create and manage encryption keys (master keys) used to encrypt your data. Alibaba Cloud KMS enables you to maintain control over who can use your master keys and gain access to your encrypted data.

Scenarios

Role	Demand	Solution
Application/Website developer	I need encryption keys to protect my application data. I have secured and full access to the keys, but the plaintext encryption keys cannot be deployed to multiple servers where my applications are deployed.	With the envelope encryption of KMS, you can first create a master key, and use it to generate a data key, then use the data key to encrypt your application data. Because the encrypted data key is inherently protected by encryption, it can be deployed together with the application it encrypted, and the plaintext master key can be kept safely in KMS service.
Service developer	My customers manage their own keys, they can authorize me to use their keys to encrypt data when necessary.	The customers manage their master keys in KMS, and they can authorize you to call KMS APIs to encrypt data with the master keys.
Chief Security Officer	I need strict permission control over the encryption keys, and each authorization can be audited.	KMS can be integrated with RAM to achieve a fine-grained access control over keys. You can use CloudMonitor to audit key usage.

Benefits

Benefits

Advantage	Traditional key management solution	KMS
Cost effectiveness	Buying secure key management equipment to construct a secure physical environment results in high hardware costs. Designing and executing secure key management specifications involve high software costs.	With KMS, you pay only for what you need and the price is low.
Easy to use	Hardware equipment APIs lack standards, making them confusing and difficult to use. Solutions and configurations of communication channel security are cumbersome.	Unified and easy-to-use APIs. Standard HTTPS protocol.
Reliability	Generally, you must use an offline backup solution to guarantee high reliability.	KMS combines a distributed system and cryptographic hardware to achieve high reliability.

Scenarios

Common usage of KMS:

- Use CMK to encrypt and decrypt data
- Use envelope encryption to encrypt and decrypt data locally

Legend

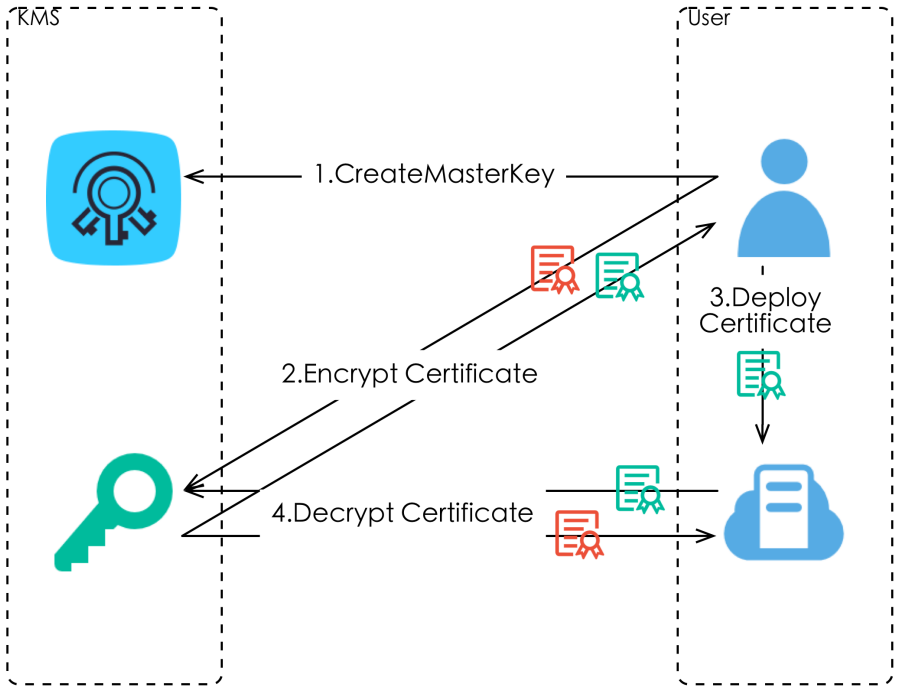
Symbol	Meaning	Symbol	Meaning
	CMK		Ciphertext key

	Plaintext certificate		Plaintext file
	Ciphertext certificate		Ciphertext file
	Plaintext key		

Use CMK to encrypt and decrypt data

You can use a CMK to encrypt and decrypt a small amount of data (less than 4 KB). KMS use secure channels for data transmission.

Scenario: protect the HTTPS certificate of a server



Procedure:

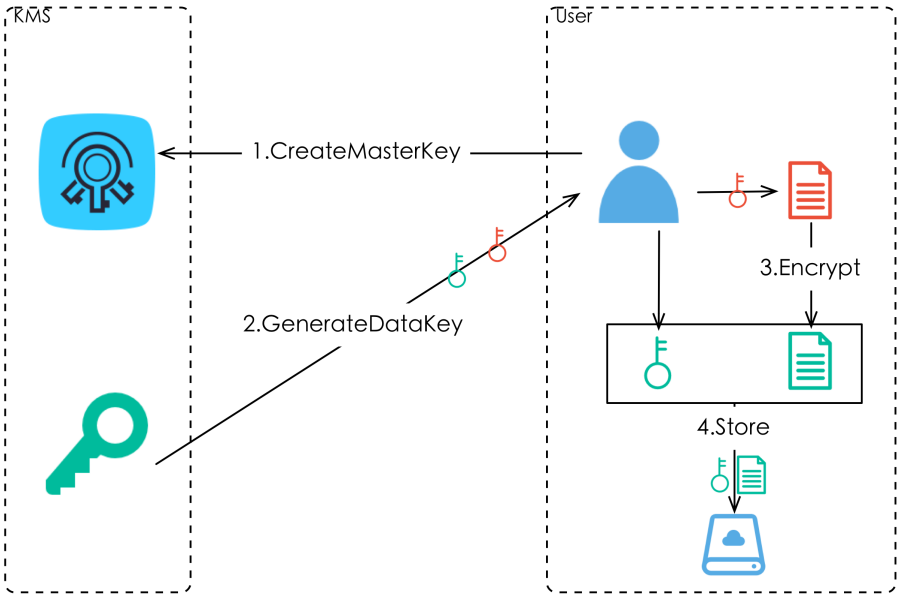
1. Create a CMK in the **KMS console** or by calling **CreateKey**.
2. Call **Encrypt** to encrypt the plaintext certificate.
3. Deploy the encrypted certificate on the server.

4. Call Decrypt to decrypt the encrypted certificate for authentication.

Use envelope encryption to encrypt and decrypt data locally

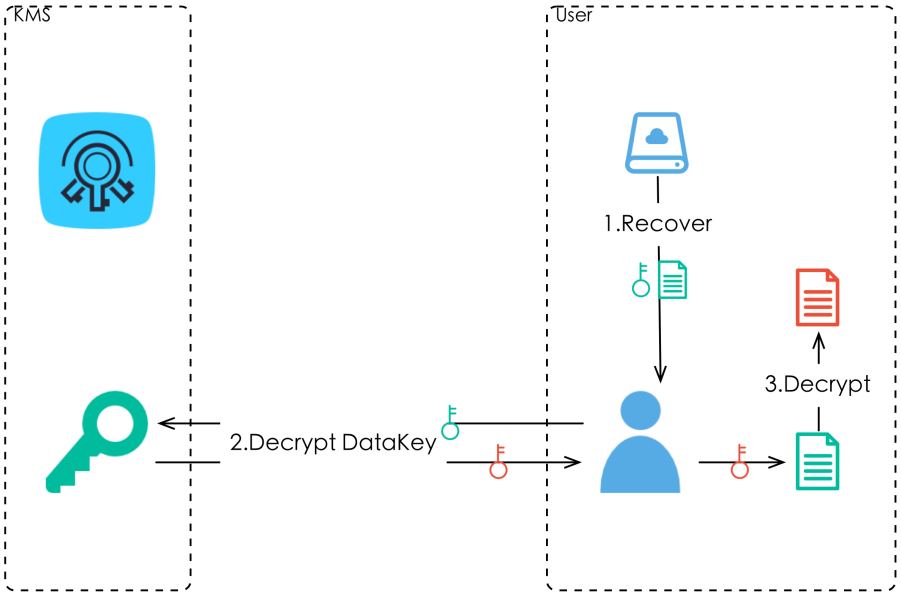
You can use KMS to create a CMK, and use the CMK to generate a data key, which is used as the encryption key to encrypt and decrypt large amounts of data locally. By this way, your cost of transmitting data through the network for encryption and decryption is saved.

Scenario: Encrypt a local file



Procedure:

1. Create a CMK in the KMS console or by calling CreateKey.
2. Call GenerateDataKey to generate a data key. It returns a plaintext data key and an encrypted data key.
3. Use the plaintext data key to encrypt the file locally, then erase the plaintext data key from memory.
4. Store the encrypted data key alongside the locally encrypted data.
5. To decrypt data locally:
    i. Call Decrypt to decrypt the encrypted data key into a plaintext data key.
    ii. Use the plaintext data key to decrypt data locally, then erase the plaintext data key from memory.



Note:

- We recommend that you use your RAM account to perform KMS operations for better permission control.

History

Release date	Version	Content
04/06/2016	1.0	KMS OBT
05/19/2016	1.1	3 more regions were supported: China North 2 (Beijing), China East 2 (Shanghai), and China South 1 (Shenzhen).
06/22/2016	1.2	Enabling and disabling keys were supported.
08/10/2016	1.3	Performance was optimized with some bugs fixed. Encryption and decryption APIs in EncryptionContext were supported.
09/20/2016	1.4	Performance was optimized with new keys added. 2 APIs to delete keys were added: ScheduleKeyDeletion and

		CancelKeyDeletion.
11/02/2016	1.5	Performance was optimized, and 4 more regions were supported: Asia Pacific NE 1 (Japan), Asia Pacific SE 2 (Sydney), Germany 1 (Frankfurt), and Middle East 1 (Dubai)。
01/22/2017	1.6	Performance was optimized, and 1 more region was supported: Hong Kong.
03/01/2017	1.7	Performance was optimized.
05/10/2017	1.8	Performance was optimized, and 1 more region was supported: China North 3 (Zhangjiakou).
06/05/2017	1.9	New API: DescribeRegions; SDK updated to 2.4.0.
11/15/2017	1.10	Performance was optimized, and 3 more regions were supported: China North 1 (Qingdao), China North 5 (Hohhot), and Asia Pacific SE (Malaysia).
03/30/2018	1.11	New API for BYOK(Bring Yourt Own Key), new API for alias, and 4 more regions were supported: US East 1 (Virginia), US West 1 (Silicon Valley), Asia Pacific SOU 1 (Mumbai), and Asia Pacific SE 5 (Jakarta).

Glossary

Terminology	Abbreviation	Definition
Key Management Service	KMS	Alibaba Cloud Key Management Service.
envelope encryption		Envelope encryption is the practice of encrypting plaintext data with a unique data key, and then encrypting the data key with a key encryption key (EDK). You may choose to encrypt the EDK with another EDK, and so on, but eventually you must have a master key. The master key is

		an unencrypted (plaintext) key with which you can decrypt one or more other keys.
Customer Master Key	CMK	CMK is the master key generated by using Alibaba Cloud KMS. It can encrypt data keys and generate envelope encryption. It can also encrypt a small amount of data.
EnvelopedDataKey / DataKey	EDK/DK	DK is the plaintext key to encrypt data, and EDK is the key to encrypt the DK by using envelope encryption.