AWS Key Management Service

API Reference

API Version 2014-11-01

AWS Key Management Service: API Reference

Table of Contents

Welcome

AWS Key Management Service (AWS KMS) is an encryption and key management web service. This guide describes the AWS KMS operations that you can call programmatically. For general information about AWS KMS, see the AWS Key Management Service Developer Guide.

> **Note**
>
> AWS provides SDKs that consist of libraries and sample code for various programming languages and platforms (Java, Ruby, .Net, iOS, Android, etc.). The SDKs provide a convenient way to create programmatic access to AWS KMS and other AWS services. For example, the SDKs take care of tasks such as signing requests (see below), managing errors, and retrying requests automatically. For more information about the AWS SDKs, including how to download and install them, see Tools for Amazon Web Services.

We recommend that you use the AWS SDKs to make programmatic API calls to AWS KMS.

Clients must support TLS (Transport Layer Security) 1.0. We recommend TLS 1.2. Clients must also support cipher suites with Perfect Forward Secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Signing Requests

Requests must be signed by using an access key ID and a secret access key. We strongly recommend that you *do not* use your AWS account access key ID and secret key for everyday work with AWS KMS. Instead, use the access key ID and secret access key for an IAM user, or you can use the AWS Security Token Service to generate temporary security credentials that you can use to sign requests.

All AWS KMS operations require Signature Version 4.

Logging API Requests

AWS KMS supports AWS CloudTrail, a service that logs AWS API calls and related events for your AWS account and delivers them to an Amazon S3 bucket that you specify. By using the information collected by CloudTrail, you can determine what requests were made to AWS KMS, who made the request, when it was made, and so on. To learn more about CloudTrail, including how to turn it on and find your log files, see the AWS CloudTrail User Guide.

Additional Resources

For more information about credentials and request signing, see the following:

- AWS Security Credentials - This topic provides general information about the types of credentials used for accessing AWS.
- AWS Security Token Service - This guide describes how to create and use temporary security credentials.
- Signing AWS API Requests - This set of topics walks you through the process of signing a request using an access key ID and a secret access key.

Commonly Used APIs

Of the APIs discussed in this guide, the following will prove the most useful for most applications. You will likely perform actions other than these, such as creating keys and assigning policies, by using the console.

- Encrypt (p. 27)
- Decrypt (p. 13)
- GenerateDataKey (p. 30)
- GenerateDataKeyWithoutPlaintext (p. 34)

This document was last updated on September 3, 2015.

Actions

The following actions are supported:

CreateAlias

Creates a display name for a customer master key. An alias can be used to identify a key and should be unique. The console enforces a one-to-one mapping between the alias and a key. An alias name can contain only alphanumeric characters, forward slashes (/), underscores (_), and dashes (-). An alias must start with the word "alias" followed by a forward slash (alias/). An alias that begins with "aws" after the forward slash (alias/aws...) is reserved by Amazon Web Services (AWS).

The alias and the key it is mapped to must be in the same AWS account and the same region.

To map an alias to a different key, call UpdateAlias (p. 62).

Request Syntax

```
{
    "AliasName": "string",
    "TargetKeyId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

AliasName

String that contains the display name. The name must start with the word "alias" followed by a forward slash (alias/). Aliases that begin with "alias/AWS" are reserved.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[a-zA-Z0-9:/_-]+$`

Required: Yes

TargetKeyId

An identifier of the key for which you are creating the alias. This value cannot be another alias but can be a globally unique identifier or a fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

AlreadyExistsException
The request was rejected because it attempted to create a resource that already exists.

HTTP Status Code: 409

DependencyTimeoutException
The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

InvalidAliasNameException
The request was rejected because the specified alias name is not valid.

HTTP Status Code: 400

KMSInternalException
The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

LimitExceededException
The request was rejected because a limit was exceeded. For more information, see Limits in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException
The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

CreateGrant

Adds a grant to a key to specify who can use the key and under what conditions. Grants are alternate permission mechanisms to key policies.

For more information about grants, see Grants in the *AWS Key Management Service Developer Guide*.

Request Syntax

```
{
    "Constraints": {
        "EncryptionContextEquals":
            {
                "string" :
                    "string"
            },
        "EncryptionContextSubset":
            {
                "string" :
                    "string"
            }
    },
    "GranteePrincipal": "string",
    "GrantTokens": [
        "string"
    ],
    "KeyId": "string",
    "Operations": [
        "string"
    ],
    "RetiringPrincipal": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

Constraints

The conditions under which the operations permitted by the grant are allowed.

You can use this value to allow the operations permitted by the grant only when a specified encryption context is present. For more information, see Encryption Context in the *AWS Key Management Service Developer Guide*.

Type: GrantConstraints (p. 67) object

Required: No

GranteePrincipal

The principal that is given permission to perform the operations that the grant permits.

To specify the principal, use the Amazon Resource Name (ARN) of an AWS principal. Valid AWS principals include AWS accounts (root), IAM users, federated users, and assumed role users. For examples of the ARN syntax to use for specifying a principal, see AWS Identity and Access Management (IAM) in the Example ARNs section of the *AWS General Reference*.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

GrantTokens

A list of grant tokens.

For more information, see Grant Tokens in the *AWS Key Management Service Developer Guide*.

Type: array of Strings

Length constraints: Minimum of 0 item(s) in the list. Maximum of 10 item(s) in the list.

Required: No

KeyId

The unique identifier for the customer master key (CMK) that the grant applies to.

To specify this value, use the globally unique key ID or the Amazon Resource Name (ARN) of the key. Examples:

- Globally unique key ID: 12345678-1234-1234-1234-123456789012
- Key ARN: arn:aws:kms:us-west-2:123456789012:key/12345678-1234-1234-1234-123456789012

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

Operations

A list of operations that the grant permits. The list can contain any combination of one or more of the following values:

- Decrypt
- Encrypt
- GenerateDataKey
- GenerateDataKeyWithoutPlaintext
- ReEncryptFrom
- ReEncryptTo
- CreateGrant
- RetireGrant

Type: array of Strings

Required: No

RetiringPrincipal

The principal that is given permission to retire the grant by using RetireGrant (p. 58) operation.

To specify the principal, use the Amazon Resource Name (ARN) of an AWS principal. Valid AWS principals include AWS accounts (root), IAM users, federated users, and assumed role users. For examples of the ARN syntax to use for specifying a principal, see AWS Identity and Access Management (IAM) in the Example ARNs section of the *AWS General Reference*.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: No

Response Syntax

```
{
    "GrantId": "string",
    "GrantToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

GrantId

The unique identifier for the grant.

You can use the `GrantId` in a subsequent RetireGrant (p. 58) or RevokeGrant (p. 60) operation.

Type: String

Length constraints: Minimum length of 1. Maximum length of 128.

GrantToken

The grant token.

For more information about using grant tokens, see Grant Tokens in the *AWS Key Management Service Developer Guide*.

Type: String

Length constraints: Minimum length of 1. Maximum length of 8192.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

DisabledException

The request was rejected because the specified key was marked as disabled.

HTTP Status Code: 409

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

InvalidGrantTokenException

The request was rejected because a grant token provided as part of the request is invalid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

LimitExceededException

The request was rejected because a limit was exceeded. For more information, see Limits in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

CreateKey

Creates a customer master key. Customer master keys can be used to encrypt small amounts of data (less than 4K) directly, but they are most commonly used to encrypt or envelope data keys that are then used to encrypt customer data. For more information about data keys, see GenerateDataKey (p. 30) and GenerateDataKeyWithoutPlaintext (p. 34).

Request Syntax

```
{
    "Description": "string",
    "KeyUsage": "string",
    "Policy": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

Description

Description of the key. We recommend that you choose a description that helps your customer decide whether the key is appropriate for a task.

Type: String

Length constraints: Minimum length of 0. Maximum length of 8192.

Required: No

KeyUsage

Specifies the intended use of the key. Currently this defaults to ENCRYPT/DECRYPT, and only symmetric encryption and decryption are supported.

Type: String

Valid Values: `ENCRYPT_DECRYPT`

Required: No

Policy

Policy to attach to the key. This is required and delegates back to the account. The key is the root of trust. The policy size limit is 32 KiB (32768 bytes).

Type: String

Length constraints: Minimum length of 1. Maximum length of 32768.

Pattern: `[\u0009\u000A\u000D\u0020-\u00FF]+`

Required: No

Response Syntax

```
{
    "KeyMetadata": {
        "AWSAccountId": "string",
        "Arn": "string",
        "CreationDate": number,
        "Description": "string",
        "Enabled": boolean,
        "KeyId": "string",
        "KeyUsage": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

KeyMetadata

Metadata associated with the key.

Type: KeyMetadata (p. 69) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

LimitExceededException

The request was rejected because a limit was exceeded. For more information, see Limits in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

MalformedPolicyDocumentException

The request was rejected because the specified policy is not syntactically or semantically correct.

HTTP Status Code: 400

UnsupportedOperationException

The request was rejected because a specified parameter is not supported.

HTTP Status Code: 400

Decrypt

Decrypts ciphertext. Ciphertext is plaintext that has been previously encrypted by using any of the following functions:

- GenerateDataKey (p. 30)
- GenerateDataKeyWithoutPlaintext (p. 34)
- Encrypt (p. 27)

Note that if a caller has been granted access permissions to all keys (through, for example, IAM user policies that grant `Decrypt` permission on all resources), then ciphertext encrypted by using keys in other accounts where the key grants access to the caller can be decrypted. To remedy this, we recommend that you do not grant `Decrypt` access in an IAM user policy. Instead grant `Decrypt` access only in key policies. If you must grant `Decrypt` access in an IAM user policy, you should scope the resource to specific keys or to specific trusted accounts.

Request Syntax

```
{
    "CiphertextBlob": blob,
    "EncryptionContext":
        {
            "string" :
                "string"
        },
    "GrantTokens": [
        "string"
    ]
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

CiphertextBlob
Ciphertext to be decrypted. The blob includes metadata.

Type: Blob

Length constraints: Minimum length of 1. Maximum length of 6144.

Required: Yes

EncryptionContext
The encryption context. If this was specified in the Encrypt (p. 27) function, it must be specified here or the decryption operation will fail. For more information, see Encryption Context.

Type: String to String map

Required: No

GrantTokens
For more information, see Grant Tokens.

Type: array of Strings

Length constraints: Minimum of 0 item(s) in the list. Maximum of 10 item(s) in the list.

Required: No

Response Syntax

```
{
    "KeyId": "string",
    "Plaintext": blob
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

KeyId

ARN of the key used to perform the decryption. This value is returned if no errors are encountered during the operation.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Plaintext

Decrypted plaintext data. This value may not be returned if the customer master key is not available or if you didn't have permission to use it.

Type: Blob

Length constraints: Minimum length of 1. Maximum length of 4096.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

DisabledException

The request was rejected because the specified key was marked as disabled.

HTTP Status Code: 409

InvalidCiphertextException

The request was rejected because the specified ciphertext has been corrupted or is otherwise invalid.

HTTP Status Code: 400

InvalidGrantTokenException

The request was rejected because a grant token provided as part of the request is invalid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

KeyUnavailableException

The request was rejected because the key was disabled, not found, or otherwise not available.

HTTP Status Code: 500

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

DeleteAlias

Deletes the specified alias. To map an alias to a different key, call UpdateAlias (p. 62).

Request Syntax

```
{
    "AliasName": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

AliasName

The alias to be deleted. The name must start with the word "alias" followed by a forward slash (alias/). Aliases that begin with "alias/AWS" are reserved.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[a-zA-Z0-9:/_-]+$`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

DescribeKey

Provides detailed information about the specified customer master key.

Request Syntax

```
{
    "KeyId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

KeyId

A unique identifier for the customer master key. This value can be a globally unique identifier, a fully specified ARN to either an alias or a key, or an alias name prefixed by "alias/".

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Alias ARN Example - arn:aws:kms:us-east-1:123456789012:alias/MyAliasName
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012
- Alias Name Example - alias/MyAliasName

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

Response Syntax

```
{
    "KeyMetadata": {
        "AWSAccountId": "string",
        "Arn": "string",
        "CreationDate": number,
        "Description": "string",
        "Enabled": boolean,
        "KeyId": "string",
        "KeyUsage": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

KeyMetadata

Metadata associated with the key.

Type: KeyMetadata (p. 69) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

DisableKey

Marks a key as disabled, thereby preventing its use. When a key is disabled, you can use the following operations to obtain information about or modify the key:

- CreateAlias (p. 4)
- DescribeKey (p. 17)
- EnableKey (p. 23)
- GetKeyPolicy (p. 39)
- GetKeyRotationStatus (p. 41)
- ListGrants (p. 45)
- ListKeyPolicies (p. 48)
- ListKeys (p. 51)
- PutKeyPolicy (p. 53)
- RetireGrant (p. 58)
- RevokeGrant (p. 60)
- UpdateAlias (p. 62)
- UpdateKeyDescription (p. 64)

All other operations that attempt to use a disabled key will result in an error.

Request Syntax

```
{
    "KeyId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

KeyId

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

DisableKeyRotation

Disables rotation of the specified key.

Request Syntax

```
{
    "KeyId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

KeyId

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

DisabledException

The request was rejected because the specified key was marked as disabled.

HTTP Status Code: 409

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

EnableKey

Marks a key as enabled, thereby permitting its use.

Request Syntax

```
{
    "KeyId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

KeyId

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

LimitExceededException

The request was rejected because a limit was exceeded. For more information, see Limits in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

EnableKeyRotation

Enables rotation of the specified customer master key.

Request Syntax

```
{
    "KeyId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

KeyId

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

DisabledException

The request was rejected because the specified key was marked as disabled.

HTTP Status Code: 409

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

Encrypt

Encrypts plaintext into ciphertext by using a customer master key. The `Encrypt` function has two primary use cases:

- You can encrypt up to 4 KB of arbitrary data such as an RSA key, a database password, or other sensitive customer information.
- If you are moving encrypted data from one region to another, you can use this API to encrypt in the new region the plaintext data key that was used to encrypt the data in the original region. This provides you with an encrypted copy of the data key that can be decrypted in the new region and used there to decrypt the encrypted data.

Unless you are moving encrypted data from one region to another, you don't use this function to encrypt a generated data key within a region. You retrieve data keys already encrypted by calling the GenerateDataKey (p. 30) or GenerateDataKeyWithoutPlaintext (p. 34) function. Data keys don't need to be encrypted again by calling `Encrypt`.

If you want to encrypt data locally in your application, you can use the `GenerateDataKey` function to return a plaintext data encryption key and a copy of the key encrypted under the customer master key (CMK) of your choosing.

Request Syntax

```
{
    "EncryptionContext":
        {
            "string" :
                "string"
        },
    "GrantTokens": [
        "string"
    ],
    "KeyId": "string",
    "Plaintext": blob
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

EncryptionContext

Name/value pair that specifies the encryption context to be used for authenticated encryption. If used here, the same value must be supplied to the `Decrypt` API or decryption will fail. For more information, see Encryption Context.

Type: String to String map

Required: No

GrantTokens

For more information, see Grant Tokens.

Type: array of Strings

Length constraints: Minimum of 0 item(s) in the list. Maximum of 10 item(s) in the list.

Required: No

KeyId

A unique identifier for the customer master key. This value can be a globally unique identifier, a fully specified ARN to either an alias or a key, or an alias name prefixed by "alias/".

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Alias ARN Example - arn:aws:kms:us-east-1:123456789012:alias/MyAliasName
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012
- Alias Name Example - alias/MyAliasName

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

Plaintext

Data to be encrypted.

Type: Blob

Length constraints: Minimum length of 1. Maximum length of 4096.

Required: Yes

Response Syntax

```
{
    "CiphertextBlob": blob,
    "KeyId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

CiphertextBlob

The encrypted plaintext. If you are using the CLI, the value is Base64 encoded. Otherwise, it is not encoded.

Type: Blob

Length constraints: Minimum length of 1. Maximum length of 6144.

KeyId

The ID of the key used during encryption.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

DisabledException

The request was rejected because the specified key was marked as disabled.

HTTP Status Code: 409

InvalidGrantTokenException

The request was rejected because a grant token provided as part of the request is invalid.

HTTP Status Code: 400

InvalidKeyUsageException

The request was rejected because the specified KeySpec parameter is not valid. The currently supported value is ENCRYPT/DECRYPT.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

KeyUnavailableException

The request was rejected because the key was disabled, not found, or otherwise not available.

HTTP Status Code: 500

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

GenerateDataKey

Generates a data key that you can use in your application to locally encrypt data. This call returns a plaintext version of the key in the `Plaintext` field of the response object and an encrypted copy of the key in the `CiphertextBlob` field. The key is encrypted by using the master key specified by the `KeyId` field. To decrypt the encrypted key, pass it to the `Decrypt` API.

We recommend that you use the following pattern to locally encrypt data: call the `GenerateDataKey` API, use the key returned in the `Plaintext` response field to locally encrypt data, and then erase the plaintext data key from memory. Store the encrypted data key (contained in the `CiphertextBlob` field) alongside of the locally encrypted data.

> **Note**
> You should not call the `Encrypt` function to re-encrypt your data keys within a region. `GenerateDataKey` always returns the data key encrypted and tied to the customer master key that will be used to decrypt it. There is no need to decrypt it twice.

If you decide to use the optional `EncryptionContext` parameter, you must also store the context in full or at least store enough information along with the encrypted data to be able to reconstruct the context when submitting the ciphertext to the `Decrypt` API. It is a good practice to choose a context that you can reconstruct on the fly to better secure the ciphertext. For more information about how this parameter is used, see Encryption Context.

To decrypt data, pass the encrypted data key to the `Decrypt` API. `Decrypt` uses the associated master key to decrypt the encrypted data key and returns it as plaintext. Use the plaintext data key to locally decrypt your data and then erase the key from memory. You must specify the encryption context, if any, that you specified when you generated the key. The encryption context is logged by CloudTrail, and you can use this log to help track the use of particular data.

Request Syntax

```
{
    "EncryptionContext":
        {
            "string" :
                "string"
        },
    "GrantTokens": [
        "string"
    ],
    "KeyId": "string",
    "KeySpec": "string",
    "NumberOfBytes": number
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

EncryptionContext

Name/value pair that contains additional data to be authenticated during the encryption and decryption processes that use the key. This value is logged by AWS CloudTrail to provide context around the data encrypted by the key.

Type: String to String map

Required: No

GrantTokens

For more information, see Grant Tokens.

Type: array of Strings

Length constraints: Minimum of 0 item(s) in the list. Maximum of 10 item(s) in the list.

Required: No

KeyId

A unique identifier for the customer master key. This value can be a globally unique identifier, a fully specified ARN to either an alias or a key, or an alias name prefixed by "alias/".

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Alias ARN Example - arn:aws:kms:us-east-1:123456789012:alias/MyAliasName
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012
- Alias Name Example - alias/MyAliasName

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

KeySpec

Value that identifies the encryption algorithm and key size to generate a data key for. Currently this can be AES_128 or AES_256.

Type: String

Valid Values: `AES_256 | AES_128`

Required: No

NumberOfBytes

Integer that contains the number of bytes to generate. Common values are 128, 256, 512, and 1024. 1024 is the current limit. We recommend that you use the `KeySpec` parameter instead.

Type: Number

Valid range: Minimum value of 1. Maximum value of 1024.

Required: No

Response Syntax

```
{
    "CiphertextBlob": blob,
    "KeyId": "string",
    "Plaintext": blob
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

CiphertextBlob

Ciphertext that contains the encrypted data key. You must store the blob and enough information to reconstruct the encryption context so that the data encrypted by using the key can later be decrypted. You must provide both the ciphertext blob and the encryption context to the Decrypt (p. 13) API to recover the plaintext data key and decrypt the object.

If you are using the CLI, the value is Base64 encoded. Otherwise, it is not encoded.

Type: Blob

Length constraints: Minimum length of 1. Maximum length of 6144.

KeyId

System generated unique identifier of the key to be used to decrypt the encrypted copy of the data key.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Plaintext

Plaintext that contains the data key. Use this for encryption and decryption and then remove it from memory as soon as possible.

Type: Blob

Length constraints: Minimum length of 1. Maximum length of 4096.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

DisabledException

The request was rejected because the specified key was marked as disabled.

HTTP Status Code: 409

InvalidGrantTokenException

The request was rejected because a grant token provided as part of the request is invalid.

HTTP Status Code: 400

InvalidKeyUsageException

The request was rejected because the specified KeySpec parameter is not valid. The currently supported value is ENCRYPT/DECRYPT.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

KeyUnavailableException

The request was rejected because the key was disabled, not found, or otherwise not available.

HTTP Status Code: 500

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

GenerateDataKeyWithoutPlaintext

Returns a data key encrypted by a customer master key without the plaintext copy of that key. Otherwise, this API functions exactly like GenerateDataKey (p. 30). You can use this API to, for example, satisfy an audit requirement that an encrypted key be made available without exposing the plaintext copy of that key.

Request Syntax

```
{
    "EncryptionContext":
        {
            "string" :
                "string"
        },
    "GrantTokens": [
        "string"
    ],
    "KeyId": "string",
    "KeySpec": "string",
    "NumberOfBytes": number
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

EncryptionContext

Name:value pair that contains additional data to be authenticated during the encryption and decryption processes.

Type: String to String map

Required: No

GrantTokens

For more information, see Grant Tokens.

Type: array of Strings

Length constraints: Minimum of 0 item(s) in the list. Maximum of 10 item(s) in the list.

Required: No

KeyId

A unique identifier for the customer master key. This value can be a globally unique identifier, a fully specified ARN to either an alias or a key, or an alias name prefixed by "alias/".

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Alias ARN Example - arn:aws:kms:us-east-1:123456789012:alias/MyAliasName
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012
- Alias Name Example - alias/MyAliasName

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

KeySpec

Value that identifies the encryption algorithm and key size. Currently this can be AES_128 or AES_256.

Type: String

Valid Values: `AES_256 | AES_128`

Required: No

NumberOfBytes

Integer that contains the number of bytes to generate. Common values are 128, 256, 512, 1024 and so on. We recommend that you use the `KeySpec` parameter instead.

Type: Number

Valid range: Minimum value of 1. Maximum value of 1024.

Required: No

Response Syntax

```
{
    "CiphertextBlob": blob,
    "KeyId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

CiphertextBlob

Ciphertext that contains the wrapped data key. You must store the blob and encryption context so that the key can be used in a future decrypt operation.

If you are using the CLI, the value is Base64 encoded. Otherwise, it is not encoded.

Type: Blob

Length constraints: Minimum length of 1. Maximum length of 6144.

KeyId

System generated unique identifier of the key to be used to decrypt the encrypted copy of the data key.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException
The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

DisabledException
The request was rejected because the specified key was marked as disabled.

HTTP Status Code: 409

InvalidGrantTokenException
The request was rejected because a grant token provided as part of the request is invalid.

HTTP Status Code: 400

InvalidKeyUsageException
The request was rejected because the specified KeySpec parameter is not valid. The currently supported value is ENCRYPT/DECRYPT.

HTTP Status Code: 400

KMSInternalException
The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

KeyUnavailableException
The request was rejected because the key was disabled, not found, or otherwise not available.

HTTP Status Code: 500

NotFoundException
The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

GenerateRandom

Generates an unpredictable byte string.

Request Syntax

```
{
    "NumberOfBytes": number
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

NumberOfBytes

Integer that contains the number of bytes to generate. Common values are 128, 256, 512, 1024 and so on. The current limit is 1024 bytes.

Type: Number

Valid range: Minimum value of 1. Maximum value of 1024.

Required: No

Response Syntax

```
{
    "Plaintext": blob
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Plaintext

Plaintext that contains the unpredictable byte string.

Type: Blob

Length constraints: Minimum length of 1. Maximum length of 4096.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

GetKeyPolicy

Retrieves a policy attached to the specified key.

Request Syntax

```
{
    "KeyId": "string",
    "PolicyName": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

KeyId

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

PolicyName

String that contains the name of the policy. Currently, this must be "default". Policy names can be discovered by calling ListKeyPolicies (p. 48).

Type: String

Length constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[\w]+`

Required: Yes

Response Syntax

```
{
    "Policy": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Policy

A policy document in JSON format.

Type: String

Length constraints: Minimum length of 1. Maximum length of 131072.

Pattern: `[\u0009\u000A\u000D\u0020-\u00FF]+`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

GetKeyRotationStatus

Retrieves a Boolean value that indicates whether key rotation is enabled for the specified key.

Request Syntax

```
{
    "KeyId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

KeyId

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

Response Syntax

```
{
    "KeyRotationEnabled": boolean
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

KeyRotationEnabled

A Boolean value that specifies whether key rotation is enabled.

Type: Boolean

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException
The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

InvalidArnException
The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException
The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

NotFoundException
The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

ListAliases

Lists all of the key aliases in the account.

Request Syntax

```
{
    "Limit": number,
    "Marker": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

Limit

When paginating results, specify the maximum number of items to return in the response. If additional items exist beyond the number you specify, the `Truncated` element in the response is set to true.

This value is optional. If you include a value, it must be between 1 and 100, inclusive. If you do not include a value, it defaults to 50.

Type: Number

Valid range: Minimum value of 1. Maximum value of 100.

Required: No

Marker

Use this parameter only when paginating results and only in a subsequent request after you've received a response with truncated results. Set it to the value of `NextMarker` from the response you just received.

Type: String

Length constraints: Minimum length of 1. Maximum length of 320.

Pattern: `[\u0020-\u00FF]*`

Required: No

Response Syntax

```
{
    "Aliases": [
        {
            "AliasArn": "string",
            "AliasName": "string",
            "TargetKeyId": "string"
        }
    ],
    "NextMarker": "string",
    "Truncated": boolean
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Aliases

A list of key aliases in the user's account.

Type: array of AliasListEntry (p. 66) objects

NextMarker

When `Truncated` is true, this value is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 320.

Pattern: `[\u0020-\u00FF]*`

Truncated

A flag that indicates whether there are more items in the list. If your results were truncated, you can use the `Marker` parameter to make a subsequent pagination request to retrieve more items in the list.

Type: Boolean

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

InvalidMarkerException

The request was rejected because the marker that specifies where pagination should next begin is not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

ListGrants

List the grants for a specified key.

Request Syntax

```
{
    "KeyId": "string",
    "Limit": number,
    "Marker": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

KeyId

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

Limit

When paginating results, specify the maximum number of items to return in the response. If additional items exist beyond the number you specify, the `Truncated` element in the response is set to true.

This value is optional. If you include a value, it must be between 1 and 100, inclusive. If you do not include a value, it defaults to 50.

Type: Number

Valid range: Minimum value of 1. Maximum value of 100.

Required: No

Marker

Use this parameter only when paginating results and only in a subsequent request after you've received a response with truncated results. Set it to the value of `NextMarker` from the response you just received.

Type: String

Length constraints: Minimum length of 1. Maximum length of 320.

Pattern: `[\u0020-\u00FF]*`

Required: No

Response Syntax

```
{
    "Grants": [
        {
            "Constraints": {
                "EncryptionContextEquals":
                    {
                        "string" :
                            "string"
                    },
                "EncryptionContextSubset":
                    {
                        "string" :
                            "string"
                    }
            },
            "GrantId": "string",
            "GranteePrincipal": "string",
            "IssuingAccount": "string",
            "Operations": [
                "string"
            ],
            "RetiringPrincipal": "string"
        }
    ],
    "NextMarker": "string",
    "Truncated": boolean
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Grants

A list of grants.

Type: array of GrantListEntry (p. 67) objects

NextMarker

When `Truncated` is true, this value is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 320.

Pattern: `[\u0020-\u00FF]*`

Truncated

A flag that indicates whether there are more items in the list. If your results were truncated, you can use the `Marker` parameter to make a subsequent pagination request to retrieve more items in the list.

Type: Boolean

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

InvalidMarkerException

The request was rejected because the marker that specifies where pagination should next begin is not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

ListKeyPolicies

Retrieves a list of policies attached to a key.

Request Syntax

```
{
    "KeyId": "string",
    "Limit": number,
    "Marker": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

KeyId

A unique identifier for the customer master key. This value can be a globally unique identifier, a fully specified ARN to either an alias or a key, or an alias name prefixed by "alias/".

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Alias ARN Example - arn:aws:kms:us-east-1:123456789012:alias/MyAliasName
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012
- Alias Name Example - alias/MyAliasName

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

Limit

When paginating results, specify the maximum number of items to return in the response. If additional items exist beyond the number you specify, the `Truncated` element in the response is set to true.

This value is optional. If you include a value, it must be between 1 and 1000, inclusive. If you do not include a value, it defaults to 100.

Currently only 1 policy can be attached to a key.

Type: Number

Valid range: Minimum value of 1. Maximum value of 1000.

Required: No

Marker

Use this parameter only when paginating results and only in a subsequent request after you've received a response with truncated results. Set it to the value of `NextMarker` from the response you just received.

Type: String

Length constraints: Minimum length of 1. Maximum length of 320.

Pattern: `[\u0020-\u00FF]*`

Required: No

Response Syntax

```
{
    "NextMarker": "string",
    "PolicyNames": [
        "string"
    ],
    "Truncated": boolean
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextMarker

When `Truncated` is true, this value is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 320.

Pattern: `[\u0020-\u00FF]*`

PolicyNames

A list of policy names. Currently, there is only one policy and it is named "Default".

Type: array of Strings

Truncated

A flag that indicates whether there are more items in the list. If your results were truncated, you can use the `Marker` parameter to make a subsequent pagination request to retrieve more items in the list.

Type: Boolean

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

ListKeys

Lists the customer master keys.

Request Syntax

```
{
    "Limit": number,
    "Marker": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

Limit

When paginating results, specify the maximum number of items to return in the response. If additional items exist beyond the number you specify, the `Truncated` element in the response is set to true.

This value is optional. If you include a value, it must be between 1 and 1000, inclusive. If you do not include a value, it defaults to 100.

Type: Number

Valid range: Minimum value of 1. Maximum value of 1000.

Required: No

Marker

Use this parameter only when paginating results and only in a subsequent request after you've received a response with truncated results. Set it to the value of `NextMarker` from the response you just received.

Type: String

Length constraints: Minimum length of 1. Maximum length of 320.

Pattern: `[\u0020-\u00FF]*`

Required: No

Response Syntax

```
{
    "Keys": [
        {
            "KeyArn": "string",
            "KeyId": "string"
        }
    ],
    "NextMarker": "string",
    "Truncated": boolean
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Keys
A list of keys.

Type: array of KeyListEntry (p. 68) objects

NextMarker
When `Truncated` is true, this value is present and contains the value to use for the `Marker` parameter in a subsequent pagination request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 320.

Pattern: `[\u0020-\u00FF]*`

Truncated
A flag that indicates whether there are more items in the list. If your results were truncated, you can use the `Marker` parameter to make a subsequent pagination request to retrieve more items in the list.

Type: Boolean

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException
The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

KMSInternalException
The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

PutKeyPolicy

Attaches a policy to the specified key.

Request Syntax

```
{
    "KeyId": "string",
    "Policy": "string",
    "PolicyName": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

KeyId

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

Policy

The policy to attach to the key. This is required and delegates back to the account. The key is the root of trust. The policy size limit is 32 KiB (32768 bytes).

Type: String

Length constraints: Minimum length of 1. Maximum length of 32768.

Pattern: `[\u0009\u000A\u000D\u0020-\u00FF]+`

Required: Yes

PolicyName

Name of the policy to be attached. Currently, the only supported name is "default".

Type: String

Length constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[\w]+`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

LimitExceededException

The request was rejected because a limit was exceeded. For more information, see Limits in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

MalformedPolicyDocumentException

The request was rejected because the specified policy is not syntactically or semantically correct.

HTTP Status Code: 400

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

UnsupportedOperationException

The request was rejected because a specified parameter is not supported.

HTTP Status Code: 400

ReEncrypt

Encrypts data on the server side with a new customer master key without exposing the plaintext of the data on the client side. The data is first decrypted and then encrypted. This operation can also be used to change the encryption context of a ciphertext.

Unlike other actions, `ReEncrypt` is authorized twice - once as `ReEncryptFrom` on the source key and once as `ReEncryptTo` on the destination key. We therefore recommend that you include the `"action":"kms:ReEncrypt*"` statement in your key policies to permit re-encryption from or to the key. The statement is included automatically when you authorize use of the key through the console but must be included manually when you set a policy by using the PutKeyPolicy (p. 53) function.

Request Syntax

```
{
    "CiphertextBlob": blob,
    "DestinationEncryptionContext":
        {
            "string" :
                "string"
        },
    "DestinationKeyId": "string",
    "GrantTokens": [
        "string"
    ],
    "SourceEncryptionContext":
        {
            "string" :
                "string"
        }
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

CiphertextBlob

Ciphertext of the data to re-encrypt.

Type: Blob

Length constraints: Minimum length of 1. Maximum length of 6144.

Required: Yes

DestinationEncryptionContext

Encryption context to be used when the data is re-encrypted.

Type: String to String map

Required: No

DestinationKeyId

A unique identifier for the customer master key used to re-encrypt the data. This value can be a globally unique identifier, a fully specified ARN to either an alias or a key, or an alias name prefixed by "alias/".

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Alias ARN Example - arn:aws:kms:us-east-1:123456789012:alias/MyAliasName
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012
- Alias Name Example - alias/MyAliasName

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

GrantTokens

For more information, see Grant Tokens.

Type: array of Strings

Length constraints: Minimum of 0 item(s) in the list. Maximum of 10 item(s) in the list.

Required: No

SourceEncryptionContext

Encryption context used to encrypt and decrypt the data specified in the `CiphertextBlob` parameter.

Type: String to String map

Required: No

Response Syntax

```
{
    "CiphertextBlob": blob,
    "KeyId": "string",
    "SourceKeyId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

CiphertextBlob

The re-encrypted data. If you are using the CLI, the value is Base64 encoded. Otherwise, it is not encoded.

Type: Blob

Length constraints: Minimum length of 1. Maximum length of 6144.

KeyId

Unique identifier of the key used to re-encrypt the data.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

SourceKeyId

Unique identifier of the key used to originally encrypt the data.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

DisabledException

The request was rejected because the specified key was marked as disabled.

HTTP Status Code: 409

InvalidCiphertextException

The request was rejected because the specified ciphertext has been corrupted or is otherwise invalid.

HTTP Status Code: 400

InvalidGrantTokenException

The request was rejected because a grant token provided as part of the request is invalid.

HTTP Status Code: 400

InvalidKeyUsageException

The request was rejected because the specified KeySpec parameter is not valid. The currently supported value is ENCRYPT/DECRYPT.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

KeyUnavailableException

The request was rejected because the key was disabled, not found, or otherwise not available.

HTTP Status Code: 500

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

RetireGrant

Retires a grant. You can retire a grant when you're done using it to clean up. You should revoke a grant when you intend to actively deny operations that depend on it. The following are permitted to call this API:

- The account that created the grant
- The `RetiringPrincipal`, if present
- The `GranteePrincipal`, if `RetireGrant` is a grantee operation

The grant to retire must be identified by its grant token or by a combination of the key ARN and the grant ID. A grant token is a unique variable-length base64-encoded string. A grant ID is a 64 character unique identifier of a grant. Both are returned by the `CreateGrant` function.

Request Syntax

```
{
    "GrantId": "string",
    "GrantToken": "string",
    "KeyId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

GrantId

Unique identifier of the grant to be retired. The grant ID is returned by the `CreateGrant` function.

- Grant ID Example - 0123456789012345678901234567890123456789012345678901234567890123

Type: String

Length constraints: Minimum length of 1. Maximum length of 128.

Required: No

GrantToken

Token that identifies the grant to be retired.

Type: String

Length constraints: Minimum length of 1. Maximum length of 8192.

Required: No

KeyId

A unique identifier for the customer master key associated with the grant. This value can be a globally unique identifier or a fully specified ARN of the key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: No

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException
The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

InvalidGrantTokenException
The request was rejected because a grant token provided as part of the request is invalid.

HTTP Status Code: 400

KMSInternalException
The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

NotFoundException
The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

RevokeGrant

Revokes a grant. You can revoke a grant to actively deny operations that depend on it.

Request Syntax

```
{
    "GrantId": "string",
    "KeyId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

GrantId
Identifier of the grant to be revoked.

Type: String

Length constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

KeyId
A unique identifier for the customer master key associated with the grant. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException
The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

InvalidArnException
The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

UpdateAlias

Updates an alias to map it to a different key.

An alias is not a property of a key. Therefore, an alias can be mapped to and unmapped from an existing key without changing the properties of the key.

An alias name can contain only alphanumeric characters, forward slashes (/), underscores (_), and dashes (-). An alias must start with the word "alias" followed by a forward slash (alias/). An alias that begins with "aws" after the forward slash (alias/aws...) is reserved by Amazon Web Services (AWS).

The alias and the key it is mapped to must be in the same AWS account and the same region.

Request Syntax

```
{
    "AliasName": "string",
    "TargetKeyId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

AliasName

String that contains the name of the alias to be modified. The name must start with the word "alias" followed by a forward slash (alias/). Aliases that begin with "alias/aws" are reserved.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[a-zA-Z0-9:/_-]+$`

Required: Yes

TargetKeyId

Unique identifier of the customer master key to be mapped to the alias. This value can be a globally unique identifier or the fully specified ARN of a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

You can call ListAliases (p. 43) to verify that the alias is mapped to the correct `TargetKeyId`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

UpdateKeyDescription

Updates the description of a key.

Request Syntax

```
{
    "Description": "string",
    "KeyId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 71).

The request requires the following data in JSON format.

Description

New description for the key.

Type: String

Length constraints: Minimum length of 0. Maximum length of 8192.

Required: Yes

KeyId

A unique identifier for the customer master key. This value can be a globally unique identifier or the fully specified ARN to a key.

- Key ARN Example - arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
- Globally Unique Key ID Example - 12345678-1234-1234-1234-123456789012

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 73).

DependencyTimeoutException

The system timed out while trying to fulfill the request. The request can be retried.

HTTP Status Code: 503

InvalidArnException

The request was rejected because a specified ARN was not valid.

HTTP Status Code: 400

KMSInternalException

The request was rejected because an internal exception occurred. The request can be retried.

HTTP Status Code: 500

NotFoundException

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 404

Data Types

The AWS Key Management Service API contains several data types that various actions use. This section describes each data type in detail.

> **Note**
> The order of each element in the response is not guaranteed. Applications should not assume a particular order.

The following data types are supported:

AliasListEntry

Description

Contains information about an alias.

Contents

AliasArn

String that contains the key ARN.

Type: String

Length constraints: Minimum length of 20. Maximum length of 2048.

Required: No

AliasName

String that contains the alias.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[a-zA-Z0-9:/_-]+$`

Required: No

TargetKeyId

String that contains the key identifier pointed to by the alias.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: No

GrantConstraints

Description

A structure for specifying the conditions under which the operations permitted by the grant are allowed.

You can use this structure to allow the operations permitted by the grant only when a specified encryption context is present. For more information about encryption context, see Encryption Context in the *AWS Key Management Service Developer Guide*.

Contents

EncryptionContextEquals

Contains a list of key-value pairs that must be present in the encryption context of a subsequent operation permitted by the grant. When a subsequent operation permitted by the grant includes an encryption context that matches this list, the grant allows the operation. Otherwise, the operation is not allowed.

Type: String to String map

Required: No

EncryptionContextSubset

Contains a list of key-value pairs, a subset of which must be present in the encryption context of a subsequent operation permitted by the grant. When a subsequent operation permitted by the grant includes an encryption context that matches this list or is a subset of this list, the grant allows the operation. Otherwise, the operation is not allowed.

Type: String to String map

Required: No

GrantListEntry

Description

Contains information about an entry in a list of grants.

Contents

Constraints

The conditions under which the grant's operations are allowed.

Type: GrantConstraints (p. 67) object

Required: No

GranteePrincipal

The principal that receives the grant's permissions.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: No

GrantId

The unique identifier for the grant.

Type: String

Length constraints: Minimum length of 1. Maximum length of 128.

Required: No

IssuingAccount

The AWS account under which the grant was issued.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: No

Operations

The list of operations permitted by the grant.

Type: array of Strings

Required: No

RetiringPrincipal

The principal that can retire the grant.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: No

KeyListEntry

Description

Contains information about each entry in the key list.

Contents

KeyArn

ARN of the key.

Type: String

Length constraints: Minimum length of 20. Maximum length of 2048.

Required: No

KeyId

Unique identifier of the key.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: No

KeyMetadata

Description

Contains metadata about a customer master key (CMK).

This data type is used as a response element for the CreateKey (p. 10) and DescribeKey (p. 17) operations.

Contents

Arn

The Amazon Resource Name (ARN) of the key. For examples, see AWS Key Management Service (AWS KMS) in the Example ARNs section of the *AWS General Reference*.

Type: String

Length constraints: Minimum length of 20. Maximum length of 2048.

Required: No

AWSAccountId

The twelve-digit account ID of the AWS account that owns the key.

Type: String

Required: No

CreationDate

The date and time when the key was created.

Type: DateTime

Required: No

Description

The friendly description of the key.

Type: String

Length constraints: Minimum length of 0. Maximum length of 8192.

Required: No

Enabled

Specifies whether the key is enabled.

Type: Boolean

Required: No

KeyId

The globally unique identifier for the key.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: Yes

KeyUsage

The cryptographic operations for which you can use the key. Currently the only allowed value is `ENCRYPT_DECRYPT`, which means you can use the key for the Encrypt (p. 27) and Decrypt (p. 13) operations.

Type: String

Valid Values: `ENCRYPT_DECRYPT`

Required: No

Common Parameters

This section lists the request parameters that all actions use. Any action-specific parameters are listed in the topic for the action.

Action

The action to be performed.

Default: None

Type: string

Required: Yes

AuthParams

The parameters that are required to authenticate a Conditional request. Contains:

- AWSAccessKeyID
- SignatureVersion
- Timestamp
- Signature

Default: None

Required: Conditional

AWSAccessKeyId

The access key ID that corresponds to the secret access key that you used to sign the request.

Default: None

Type: string

Required: Yes

Expires

The date and time when the request signature expires, expressed in the format YYYY-MM-DDThh:mm:ssZ, as specified in the ISO 8601 standard.

Condition: Requests must include either *Timestamp* or *Expires*, but not both.

Default: None

Type: string

Required: Conditional

SecurityToken

The temporary security token that was obtained through a call to AWS Security Token Service. For a list of services that support AWS Security Token Service, go to Using Temporary Security Credentials to Access AWS in **Using Temporary Security Credentials**.

Default: None

Type: string

Required: No

Signature

The digital signature that you created for the request. For information about generating a signature, go to the service's developer documentation.

Default: None

Type: string

Required: Yes

SignatureMethod

The hash algorithm that you used to create the request signature.

Default: None

Type: string

Valid Values: `HmacSHA256 | HmacSHA1`

Required: Yes

SignatureVersion

The signature version you use to sign the request. Set this to the value that is recommended for your service.

Default: None

Type: string

Required: Yes

Timestamp

The date and time when the request was signed, expressed in the format YYYY-MM-DDThh:mm:ssZ, as specified in the ISO 8601 standard.

Condition: Requests must include either *Timestamp* or *Expires*, but not both.

Default: None

Type: string

Required: Conditional

Version

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Default: None

Type: string

Required: Yes

Common Errors

This section lists the common errors that all actions return. Any action-specific errors are listed in the topic for the action.

IncompleteSignature
The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure
The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

InvalidAction
The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

InvalidClientTokenId
The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

InvalidParameterCombination
Parameters that must not be used together were used together.

HTTP Status Code: 400

InvalidParameterValue
An invalid or out-of-range value was supplied for the input parameter.

HTTP Status Code: 400

InvalidQueryParameter
The AWS query string is malformed or does not adhere to AWS standards.

HTTP Status Code: 400

MalformedQueryString
The query string contains a syntax error.

HTTP Status Code: 404

MissingAction
The request is missing an action or a required parameter.

HTTP Status Code: 400

MissingAuthenticationToken

The request must contain either a valid (registered) AWS access key ID or X.509 certificate.

HTTP Status Code: 403

MissingParameter

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

OptInRequired

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

RequestExpired

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

ServiceUnavailable

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

Throttling

The request was denied due to request throttling.

HTTP Status Code: 400

ValidationError

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400