AWS KMS 密码学细节

AWS Key Management Service

AWS Key Management Service: AWS KMS 密码学细节

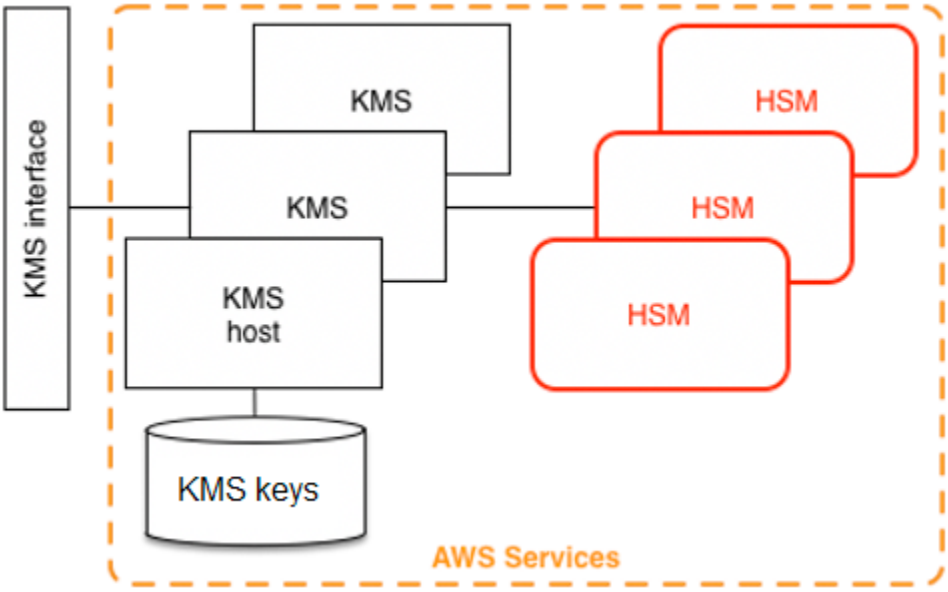
Table of Contents

密码学细节简介 AWS KMS

AWS Key Management Service (AWS KMS) 提供用于生成和管理加密密钥的 Web 界面，并充当保护数据的加密服务提供商。 AWS KMS 提供与服务集成的传统密钥管理 AWS 服务，通过集中管理和审计 AWS，提供一致的客户密钥视图。本白皮书详细描述了的加密操作 AWS KMS ，以帮助您评估该服务提供的功能。

AWS KMS 包括通过 AWS Management Console、命令行界面和 RESTful API 操作的 Web 界面，用于请求由经 FIPS 140-3 验证的硬件安全模块组成的分布式队列的加密操作 (HSMs) [1]。 AWS KMS HSM 是一款多芯片独立硬件加密设备，旨在提供专用的加密功能，以满足安全性和可扩展性要求。AWS KMS您可以在作为 AWS KMS keys管理的密钥下建立自己的基于 HSM 的加密层次结构。这些密钥仅在内存中可用， HSMs 并且仅在处理您的加密请求所需的必要时间内在内存中可用。您可以创建多个 KMS 密钥，每个密钥均由其密钥 ID 表示。只有在每个客户管理的 AWS IAM 角色和账户下，才能创建、删除客户的 KMS 密钥或用于加密、解密、签名或验证数据。您可以通过创建附加到密钥的策略来定义对谁可以管理 KMS 密钥的 and/or 使用进行访问控制。此类策略允许您为每个 API 操作定义特定于应用程序的用途。

此外，大多数 AWS 服务都支持使用 KMS 密钥对静态数据进行加密。此功能允许客户通过控制访问 KMS 密钥的方式和时间来控制 AWS 服务访问加密数据的方式和时间。



AWS KMS 是一项分层服务，由面向 Web AWS KMS 的主机和一层组成。 HSMs这些分层主机的分组构成了 AWS KMS 堆栈。对的所有请求都 AWS KMS 必须通过传输层安全协议 (TLS) 发出，并在 AWS KMS 主机上终止。 AWS KMS 主机仅允许使用提供完美前向保密性的密码套件进行 TLS。

AWS KMS 使用适用于所有其他 AWS API 操作的相同凭证和策略机制 AWS Identity and Access Management (IAM) 对您的请求进行身份验证和授权。

基本概念

学习一些基本的术语和概念将帮助您充分利用这些术语和概念 AWS Key Management Service。

AWS KMS key

> **Note**
>
> AWS KMS 正在将 “客户主密钥 (CMK)” 一词替换为 “AWS KMS key和 KMS 密钥”。这一概念并未改变。为了防止重大更改， AWS KMS 保留了该术语的一些变体。

表示密钥层次结构顶部的逻辑密钥。KMS 密钥将指定一个 Amazon Resource Name (ARN)，其中包含唯一密钥标识符或密钥 ID。 AWS KMS keys 有三种类型：

- 客户管理密钥 - 客户创建并控制客户管理密钥的生命周期和密钥策略。针对这些密钥发出的所有请求都被记录为 CloudTrail 事件。
- AWS 托管式密钥— AWS 创建和控制生命周期和关键策略 AWS 托管式密钥，这些策略是客户资源中的资源 AWS 账户。客户可以查看这些密钥的访问策略和 CloudTrail 事件 AWS 托管式密钥，但无法管理这些密钥的任何方面。针对这些密钥发出的所有请求都被记录为 CloudTrail 事件。
- AWS 拥有的密钥— 这些密钥由 AWS 创建并专门用于跨不同 AWS 服务的内部加密操作。客户无法查看中的关键策略或 AWS 拥有的密钥 使用情况 CloudTrail。

别名

与 KMS 密钥相关联的用户易记名称。在许多 AWS KMS API 操作中，别名可以与密钥 ID 互换使用。

权限

附加到 KMS 密钥的策略，用于定义密钥的权限。默认策略允许您定义的任何委托人，并 AWS 账户 允许添加引用该密钥的 IAM 策略。

授权

一开始不知道预期 IAM 委托人或使用持续时间，因此无法添加到密钥或 IAM 策略时，使用 KMS 密钥的委托权限。授权的用途之一是为 AWS 服务如何使用 KMS 密钥定义限定范围的权限。如果您没有直接签名的 API 调用，该服务可能需要使用您的密钥代表您对加密数据执行异步工作。

数据密钥

生成的加密密钥由 KMS 密钥保护。 HSMs AWS KMS 允许授权实体获取受 KMS 密钥保护的数据密钥。这些密钥可以作为明文（未加密）数据密钥和加密数据密钥返回。数据密钥可以是对称的，也可以是非对称的（返回公有部分和私有部分）。

密文

的加密输出 AWS KMS，有时也称为客户密文，以消除混淆。密文包含带有附加信息的加密数据，这些信息标识要在解密过程中使用的 KMS 密钥。加密数据密钥是使用 KMS 密钥时生成的密文的一个常见示例，但大小小于 4 KB 的任何数据都可以在 KMS 密钥下加密以生成密文。

加密上下文

与受保护信息关联的其他信息的键值对映射 AWS KMS。 AWS KMS 使用经过身份验证的加密来保护数据密钥。加密上下文已合并到加密密文中经过身份验证的加密的 AAD AWS KMS中。此上下文信息是可选的，在请求密钥（或加密操作）时不会返回。但如果使用，则需要此上下文值才能成功完成解密操作。加密上下文的预期用途是提供额外的经身份验证信息。这些信息可以帮助您执行策略并包含在 AWS CloudTrail 日志中。例如，您可以使用 {"key name":"satellite uplink key"} 键值对来命名数据密钥。随后使用该密钥会创建一个包含 “密钥名称” 的 AWS CloudTrail 条目：“卫星上行链路密钥”。此附加信息可提供有用的上下文，以了解使用指定 KMS 密钥的原因。

公有密钥

使用非对称密码（RSA 或椭圆曲线）时，公有密钥是公有-私有密钥对的“公有组成部分”。加密详细信息介绍公有密钥可以共享并分发给需要为公有-私有密钥对拥有者加密数据的实体。对于数字签名操作，公有密钥用于验证签名。

私有密钥

使用非对称密码（RSA 或椭圆曲线）时，私有密钥是公有-私有密钥对的“私有组成部分”。私有密钥用于解密数据或创建数字签名。与对称 KMS 密钥类似，私钥在中加密。 HSMs这些密钥仅在 HSM 的短期存储中解密，并且仅在处理加密请求所需的时间内解密。

AWS KMS 设计目标

AWS KMS 旨在满足以下要求。

持久性

加密密钥的耐久性旨在与中 AWS最高耐久性服务的耐久性相当。一个加密密钥可以加密长时间累积的大量数据。

值得信赖

密钥的使用受您定义和管理的访问控制策略的保护。没有导出明文 KMS 密钥的机制。加密密钥的机密性至关重要。多名具有特定角色访问权限的基于法定人数的访问控制的 Amazon 员工需要对这些权限执行管理操作。 HSMs

低延迟和高吞吐量

AWS KMS 提供延迟和吞吐量级别的加密操作，适合中 AWS其他服务使用。

独立区域

AWS 为需要限制不同区域数据访问的客户提供独立区域。可以在 AWS 区域内隔离密钥使用。

随机数的安全来源

由于强加密依赖于真正不可预测的随机数生成，因此 AWS KMS 提供优质且经过验证的随机数来源。

审核

AWS KMS 在 AWS CloudTrail 日志中记录加密密钥的使用和管理。您可以使用 AWS CloudTrail 日志来检查您的加密密钥的使用情况，包括 AWS 服务代表您使用密钥的情况。

为了实现这些目标，该 AWS KMS 系统包括一组管理 “域” 的 AWS KMS 运营商和服务主机运营商（统称为 “运营商”）。域是一组按区域定义的 AWS KMS 服务器 HSMs、和运营商。每个 AWS KMS 操作员都有一个硬件令牌，其中包含用于验证其操作的私钥和公钥对。它们 HSMs还有一个额外的私钥和公钥对，用于建立保护 HSM 状态同步的加密密钥。

此 paper 说明了如何 AWS KMS 保护您的密钥和其他要加密的数据。在本文档中，加密密钥或要加密的数据称为“机密”或“机密材料”。

AWS Key Management Service 基金会

本章中的主题描述了密码学原语 AWS Key Management Service 及其用途。他们还介绍了的基本要素 AWS KMS。

主题

加密基元

AWS KMS 使用可配置的加密算法，因此系统可以从一种已批准的算法或模式快速迁移到另一种已批准的算法或模式。初始原定设置加密算法集是从联邦信息处理标准（经 FIPS 批准）算法中选择的，用于确保其安全属性和性能。

熵和随机数生成

AWS KMS 密钥生成在 AWS KMS HSMs。 HSMs 实现使用 NIST SP800-90A Deterministic Random Bit Generator (DRBG) CTR_DRBG using AES-256 的混合随机数生成器。它采用 384 位熵的非确定性随机位生成器进行植入，并使用额外的熵进行更新，以便在每次调用加密材料时提供预测阻力。

对称密钥操作（仅加密）

其中使用的所有对称密钥加密命令都 HSMs 使用高级加密标准 (AES)，在 G alois 计数器模式 (GCM) 中使用 256 位密钥。解密的类似调用使用反函数。

AES-GCM 是一种经过身份验证的加密方案。除了对明文进行加密以生成密文外，它还计算密文上的身份验证标签和需要身份验证的任何其他数据（附加身份验证数据或 AAD）。身份验证标签有助于确保数据来自声称的来源，并且密文和 AAD 未被修改。

通常，我们的描述中会 AWS 忽略包含 AAD，尤其是在提及数据密钥的加密时。在这些情况下，周围的文本暗示要加密的结构在要加密的明文和要保护的明文 AAD 之间进行分区。

AWS KMS 提供了将密钥材料导入到 AWS KMS key 而不是依赖 AWS KMS 生成密钥材料的选项。可以使用 RSAES-OAEP 或 RSAES PKCS1- v1_5 对导入的密钥材料进行加密，以便在传输到 HS M 期间保护密钥。 AWS KMS RSA 密钥对是在上 AWS KMS HSMs生成的。导入的密钥材料在 AWS KMS HSM 上解密，然后在 AES-GCM 下重新加密，然后由服务存储。

非对称密钥操作（加密、数字签名和签名验证）

AWS KMS 支持对加密和数字签名操作使用非对称密钥操作。非对称密钥操作依赖于数学上相关的公有密钥和私有密钥对，可用于加密和解密或签名和签名验证，但不能同时用于二者。私钥永远不会处于 AWS KMS 未加密状态。您可以 AWS KMS 通过调用 AWS KMS API 操作在内部使用公钥，也可以下载公钥并在外部使用 AWS KMS。

AWS KMS 支持两种类型的非对称密码。

- RSA-OAEP（用于加密）与 RSA-PSS 和 RSA-PKCS-#1-v1_5（用于签名和验证）- 支持 RSA 密钥长度（以位为单位）：2048、3072 和 4096；从而满足不同的安全要求。
- 椭圆曲线 (ECC) - 专用于签名和验证。支持 ECC 曲线：NIST P256、P384、P521、SECP 256k1。

密钥派生函数

密钥派生函数用于从初始机密或密钥派生其他密钥。 AWS KMS 使用密钥派生函数 (KDF) 为 AWS KMS key下的每个加密派生每次调用的密钥。所有 KDF 操作在计数器模式下使用 KDF，使用带有 [0] 的 HMAC SHA256 [FIPS197FIPS18]。256 位派生密钥与 AES-GCM 一起使用，用于加密或解密客户数据和密钥。

AWS KMS 内部使用数字签名

数字签名还用于验证 AWS KMS 实体之间的命令和通信。所有服务实体都有一个椭圆曲线数字签名算法 (ECDSA) 密钥对。它们执行 ECDSA，如 Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS)（在加密消息语法 [CMS] 中使用椭圆曲线加密 [ECC]）和 X9.62-2005：Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)（金融服务行业的公有密钥加密：椭圆曲线数字签名算法 [ECDSA]）中所定义。这些实体使用联邦信息处理标准出版物 FIPS PUB 180-4 中定义的安全哈希算法，即。SHA384这些密钥在曲线 secp384r1 (NIST-P384) 上生成。

信封加密

许多加密系统中使用的基本结构是信封加密。信封加密使用两个或更多加密密钥来保护消息。通常，一个密钥派生自较长期间的静态密钥 k，另一个密钥是每条消息密钥 msgKey，该密钥生成以加密消息。信封通过加密以下消息形成：ciphertext = Encrypt(msgKey, message)。然后，消息密钥使用长期静态密钥进行加密：encKey = Encrypt(k, msgKey)。最后，这两个值 (encKey, ciphertext) 打包成一个结构，或信封加密的消息。

具有 k 访问权限的收件人可以打开信封加密的消息，方法是首先解密加密的密钥，然后解密消息。

AWS KMS 提供了管理这些长期静态密钥和自动执行数据信封加密过程的能力。

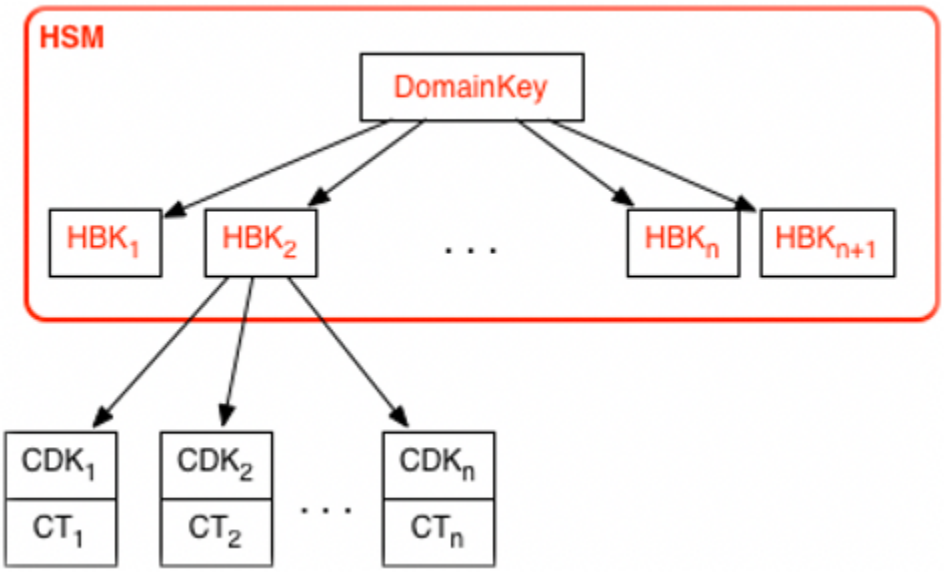
除了 AWS KMS 服务中提供的加密功能外，Encryption AWS SDK 还提供客户端信封加密库。您可以使用这些库来保护您的数据和用于加密该数据的加密密钥。

AWS KMS key 等级制度

您的密钥层次结构从顶级逻辑密钥开始，即 AWS KMS key。KMS 密钥表示顶级密钥材料的容器，在 AWS 服务命名空间中使用 Amazon Resource Name (ARN) 进行唯一定义。ARN 包含唯一生成的密钥标识符，即密钥 ID。KMS 密钥是根据用户通过 AWS KMS发起的请求创建的。收到后， AWS KMS 请求创建初始 HSM 支持密钥 (HBK)，将其放入 KMS 密钥容器中。HBK 在域中的 HSM 上生成，并且设计为永远不会以明文形式从 HSM 导出。相反地，HBK 在 HSM 管理的域密钥下以加密形式导出。HBKs这些导出的密钥令牌称为导出的密钥令牌 (EKTs)。

EKT 将导出到高持久性、低延迟的存储中。例如，假设您收到逻辑 KMS 密钥的 ARN。这表示您的密钥层次结构或加密上下文的顶部。您可以在自己的账户中创建多个 KMS 密钥，并像任何其他 AWS 命名资源一样对 KMS 密钥设置策略。

在特定 KMS 密钥的层次结构中，可以将 HBK 视为 KMS 密钥的一个版本。当您想要轮换 KMS 密钥时 AWS KMS，会创建一个新的 HBK，并将其与 KMS 密钥关联为 KMS 密钥的主动 HBK。较旧 HBKs 的数据会被保留，可用于解密和验证以前受保护的数据。但只有活动的加密密钥才能用于保护新信息。



您可以通过 AWS KMS 请求使用您的 KMS 密钥直接保护信息，或者请求受您的 KMS 密钥保护的其他由 HSM 生成的密钥。这些密钥称为客户数据密钥，或 CDKs。 CDKs 可以加密后返回为密文 (CT)、

纯文本或两者兼而有之。所有使用 KMS 密钥加密的对象（客户提供的数据或 HSM 生成的密钥）只能通过调用在 HSM 上进行解密。 AWS KMS

返回的密文或解密后的有效载荷永远不会存储在其中。 AWS KMS该信息通过与 AWS KMS的 TLS 连接返回给您。这也适用于 AWS 服务部门代表您拨打的电话。

密钥层次结构和特定密钥属性如下表中所示。

键	描述	生命周期
域密钥	仅在 HSM 内存中的 256 位 AES-GCM 密钥，用于包装 KMS 密钥（HSM 备用密钥）的版本。	每天轮换[1]
HSM 备用密钥	256 位对称密钥或者 RSA 或椭圆曲线私有密钥，用于保护客户数据和密钥，在域密钥下加密存储。。一个或多个 HSM 备用密钥组成 KMS 密钥（通过 keyId 表示）。	每年轮换[2]（可选配置）
派生加密密钥	仅在 HSM 内存中的 256 位 AES-GCM 密钥，用于加密客户数据和密钥。从每个加密的 HBK 派生。	每次加密时使用一次，并在解密时重新生成
客户数据密钥	以明文和密文形式从 HSM 导出的、用户定义的对称或非对称密钥。 在 HSM 备用密钥下加密，然后通过 TLS 通道返回给授权用户。	轮换和使用由应用程序控制

[1] AWS KMS 可能会不时地将域密钥轮换放宽到最多每周一次，以处理域管理和配置任务。

[2] 由您代您 AWS 托管式密钥 创建和管理 AWS KMS 的默认值每年自动轮换。

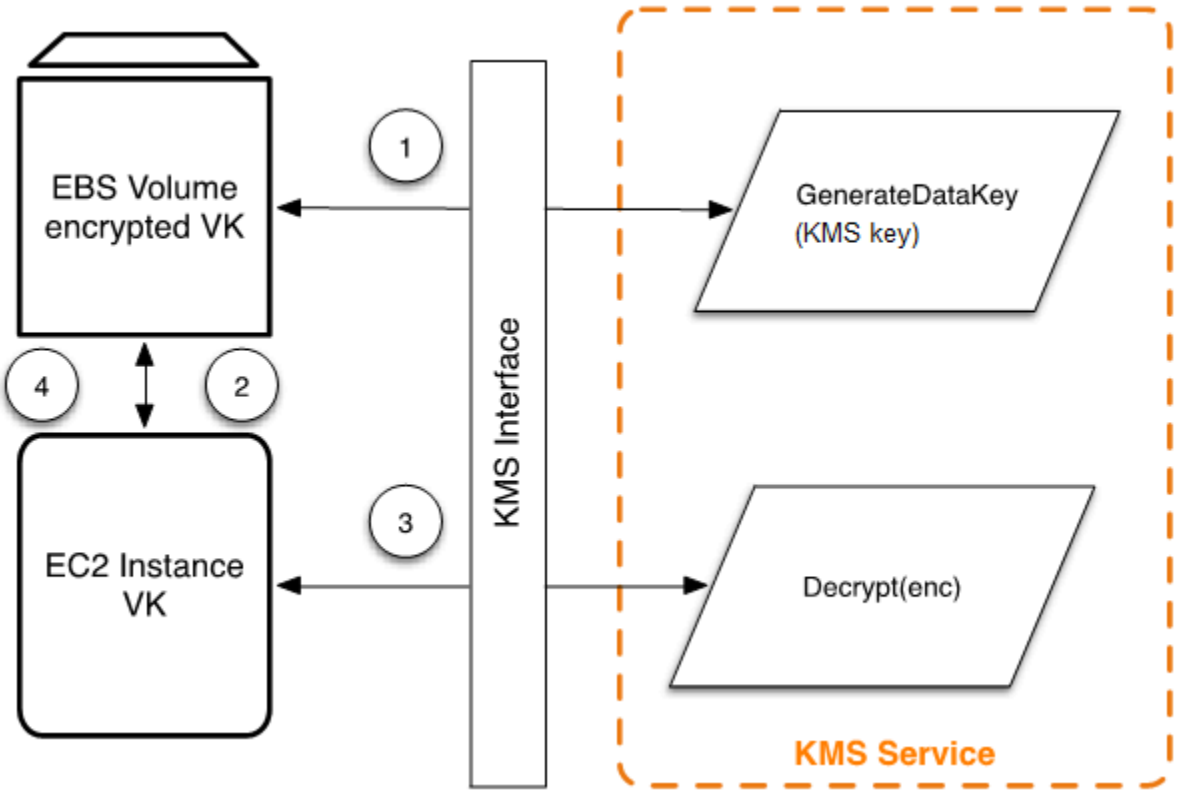
AWS KMS 用例

用例可以帮助您充分利用 AWS Key Management Service。第一个演示了如何在亚马逊弹性区块存储 (Amazon EBS) 卷 AWS KMS keys 上 AWS KMS 执行服务器端加密。第二个是客户端应用程序，它演示了如何使用信封加密来保护内容。 AWS KMS

主题

Amazon EBS 卷加密

Amazon EBS 提供卷加密功能。每个卷都使用 AES-256-XTS 进行加密。这需要两个 256 位的卷密钥，您可以将其视为一个 512 位的卷密钥。卷密钥在您账户中的 KMS 密钥下进行加密。要使 Amazon EBS 为您加密卷，必须具有账户中 KMS 密钥下生成卷密钥 (VK) 的访问权限。为此，您可以向 Amazon EBS 授权 KMS 密钥，从而生成数据密钥以及加密和解密这些卷密钥。现在，Amazon EBS AWS KMS 使用 KMS 密钥来生成 AWS KMS 加密的卷密钥。



以下工作流对写入 Amazon EBS 卷的数据进行加密：

1. Amazon EBS 通过 AWS KMS TLS 会话获取 KMS 密钥下的加密卷密钥，并将加密密钥与卷元数据一起存储。
2. 装入 Amazon EBS 卷后，将检索加密的卷密钥。
3. AWS KMS 通过 TLS 调用以解密加密的卷密钥。 AWS KMS 识别 KMS 密钥并向队列中的 HSM 发出内部请求以解密加密的卷密钥。 AWS KMS 然后通过 TLS 会话将卷密钥返回到包含您的实例的亚马逊弹性计算云 (Amazon EC2) 主机。
4. 卷密钥用于加密和解密传入和传出所连接 Amazon EBS 卷的所有数据。Amazon EBS 会保留加密的卷密钥，以备在日后内存中的卷密钥不再可用时使用。

有关使用 KMS 密钥加密亚马逊 EBS 卷的更多信息，请参阅AWS Key Management Service 开发者指南 AWS KMS中的亚马逊 Elastic Block Store 如何使用以及亚马逊用户指南和亚马逊 EC2 EC2 用户指南中的 Amazon EBS 加密。

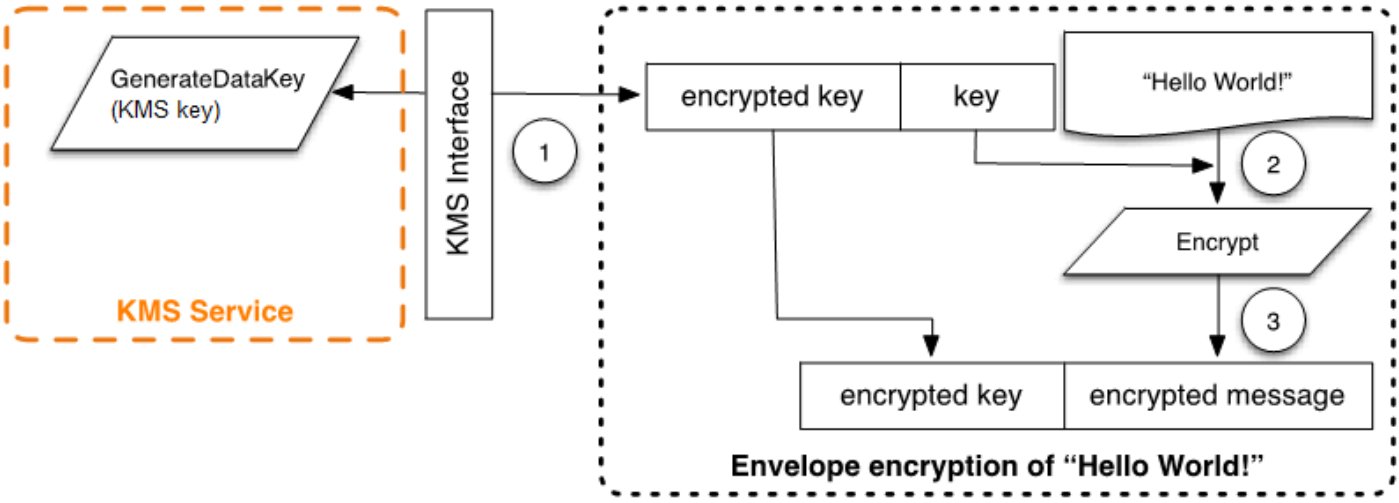
客户端加密

AWS Encryption SDK 包含一个 API 操作，用于使用 KMS 密钥执行信封加密。有关完整的建议和使用详细信息，请参阅相关文档。客户端应用程序可以使用 AWS Encryption SDK 来执行信封加密 AWS KMS。

```
// Instantiate the SDK
final AwsCrypto crypto = new AwsCrypto();
// Set up the KmsMasterKeyProvider backed by the default credentials
final KmsMasterKeyProvider prov = new KmsMasterKeyProvider(keyId);
// Do the encryption
final byte[] ciphertext = crypto.encryptData(prov, message);
```

客户端应用程序可以运行以下步骤：

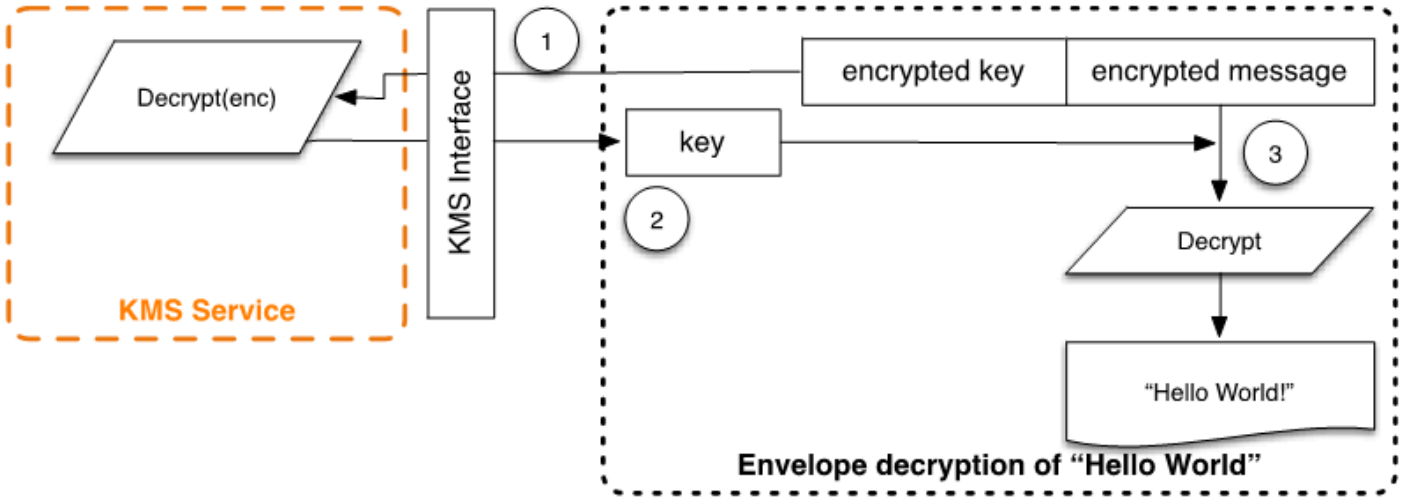
1. 在 KMS 密钥下请求新数据密钥。将返回加密的数据密钥和数据密钥的明文版本。
2. 在中 AWS Encryption SDK，使用纯文本数据密钥对消息进行加密。然后，明文数据密钥将从内存中删除。
3. 加密的数据密钥和加密的消息将组成一个密文字节数组。



可以使用解密功能对信封加密的消息进行解密，以获取最初加密的消息。

```
final AwsCrypto crypto = new AwsCrypto();
final KmsMasterKeyProvider prov = new KmsMasterKeyProvider(keyId);
// Decrypt the data
final CryptoResult<byte[], KmsMasterKey> res = crypto.decryptData(prov, ciphertext);
// We need to check the KMS key to ensure that the
// assumed key was used
if (!res.getMasterKeyIds().get(0).equals(keyId)) {
     throw new IllegalStateException("Wrong key id!");
}
byte[] plaintext = res.getResult();
```

1. 解 AWS Encryption SDK 析信封加密的消息以获取加密的数据密钥并 AWS KMS 向请求解密数据密钥。
2. AWS Encryption SDK 接收来自的纯文本数据密钥。 AWS KMS
3. 该数据密钥随后用于解密消息，从而返回初始明文。

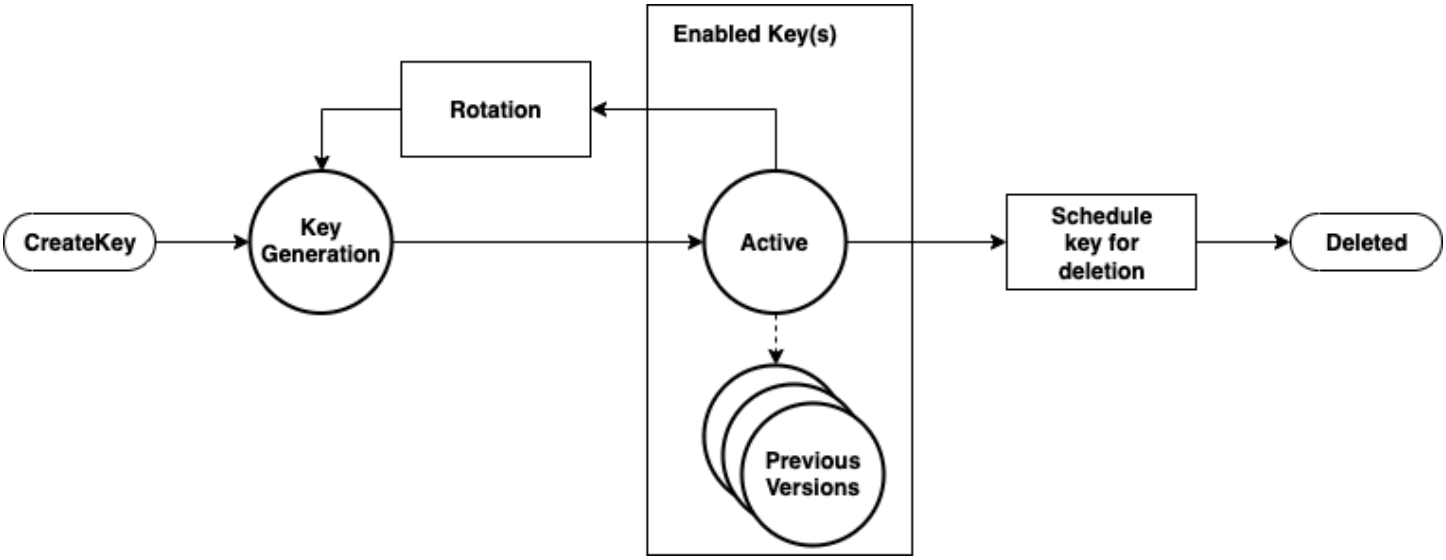

Decrypt(enc)
KMS Interface
KMS Service
1
encrypted key
encrypted message
key
2
3
Decrypt
"Hello World!"
Envelope decryption of "Hello World"

与 AWS KMS keys

AWS KMS key 是指可能引用一个或多个硬件安全模块 (HSM) 后备密钥 (HBKs) 的逻辑密钥。本主题说明了如何创建 KMS 密钥、导入密钥材料，以及如何启用、禁用、轮换和删除 KMS 密钥。

> **Note**
>
> AWS KMS 正在将 “客户主密钥 (CMK)” 一词替换为 “AWS KMS key和 KMS 密钥”。这一概念并未改变。为了防止重大更改， AWS KMS 保留了该术语的一些变体。

本章介绍了从创建到删除的 KMS 密钥生命周期，如下图所示。



主题

- 正在呼叫 CreateKey
- 导入密钥材料
- 启用和禁用密钥
- 删除 密钥
- 轮换密钥材料

正在呼叫 CreateKey

在 AWS KMS key 调用 `CreateKey`API 时会生成一个。

以下是 CreateKey 请求语法的子集。

```
{
  "Description": "string",
  "KeySpec": "string",
  "KeyUsage": "string",
  "Origin": "string";
  "Policy": "string"
}
```

请求接受采用 JSON 格式的以下数据。

描述

（可选）密钥的描述。我们建议您选择可帮助您决定密钥是否适合某项任务的描述。

KeySpec

指定要创建的 KMS 密钥的类型。默认值 SYMMETRIC_DEFAULT 会创建一个对称加密 KMS 密钥。对于对称加密密钥而言，此参数是可选的；对于所有其他密钥规范，则必须提供此参数。

KeyUsage

指定密钥的用途。有效值为 ENCRYPT_DECRYPT、SIGN_VERIFY 或 GENERATE_VERIFY_MAC。默认值为 ENCRYPT_DECRYPT。对于对称加密密钥而言，此参数是可选的；对于所有其他密钥规范，则必须提供此参数。

Origin

（可选）KMS 密钥的密钥材料来源。默认值为AWS_KMS，表示 AWS KMS 生成和管理 KMS 密钥的密钥材料。其他有效值包括EXTERNAL，它表示在没有密钥材料的情况下为导入的密钥材料创建AWS_CLOUDHSM的 KMS 密钥，以及在由您控制的 AWS CloudHSM 集群支持的自定义密钥存储中创建 KMS 密钥。

策略

（可选）要附加到密钥的策略。如果忽略该策略，将使用允许根账户和具有 AWS KMS 权限的 IAM 主体进行管理的原定设置策略（以下）创建该密钥。

有关此策略的详细信息，请参阅AWS Key Management Service 开发人员指南中的AWS KMS中的密钥策略和默认密钥策略。

CreateKey 请求会返回一个包含密钥 ARN 的响应。

```
arn:<partition>:kms:<region>:<account-id>:key/<key-id>
```

如果 `Origin` 为 `AWS_KMS`，则在创建 ARN 后，将利用已通过身份验证的会话发出对某个 AWS KMS HSM 的请求，以预调配硬件安全模块（HSM）备用密钥（HBK）。HBK 是与 KMS 密钥的这一密钥 ID 关联的 256 位密钥。它只能在 HSM 上生成，并且设计为永远不会以明文形式导出到 HSM 边界之外。HBK 将利用当前域密钥 DK_0 进行加密。 HBKs 这些经过加密的密钥令牌称为加密的密钥令牌 (EKTs)。尽管 HSMs 可以配置为使用各种密钥包装方法，但当前的实现在 Galois 计数器模式 (GCM) 中使用 AES-256，这是一种经过身份验证的加密方案。利用这种已通过身份验证的加密模式，可方便我们保护一些明文导出的密钥令牌元数据。

这在样式上表示为：

$$EKT = Encrypt(DK_0, HBK)$$

为您的 KMS 密钥和后续密钥提供了两种基本的保护形式 HBKs：在您的 KMS 密钥上设置的授权策略和对关联 HBKs的 KMS 密钥的加密保护。其余部分描述了中管理功能的加密保护和安全性。 AWS KMS

除了 ARN 以外，您还可以为密钥创建一个别名，从而常见一个用户友好的名称并将其与 KMS 密钥关联。将别名与 KMS 密钥关联后，就可以在加密操作中使用别名来标识 KMS 密钥。有关详细信息，请参阅《AWS Key Management Service 开发人员指南》中的使用别名。

KMS 密钥的使用有多个级别的授权。 AWS KMS 在加密内容和 KMS 密钥之间启用单独的授权策略。例如， AWS KMS 信封加密的 Amazon Simple Storage Service (Amazon S3) 对象会继承 Amazon S3 存储桶上的策略。但是，对必要加密密钥的访问权限由 KMS 密钥上的访问策略决定。有关 KMS 密钥授权的信息，请参阅《AWS Key Management Service 开发人员指南》中的身份验证和访问控制 AWS KMS。

导入密钥材料

AWS KMS 提供了一种用于导入用于 HBK 的加密材料的机制。如中所述正在呼叫 CreateKey，如果将CreateKey命令`Origin`设置为`EXTERNAL`，则会创建一个不包含底层 HBK 的逻辑 KMS 密钥。必须使用 `ImportKeyMaterial` API 调用导入加密材料。您可以使用此功能来控制密钥创建和加密材料的持久性。如果您使用此功能，我们建议您在环境中处理这些密钥和持久性时格外小心。有关导入密钥材料的完整详细信息和建议，请参阅 AWS Key Management Service 开发人员指南中的导入密钥材料。

正在呼叫 ImportKeyMaterial

`ImportKeyMaterial` 请求导入 HBK 的必要加密材料。加密材料必须是 256 位对称密钥。它必须使用 `WrappingAlgorithm` 中指定的算法在最近 `GetParametersForImport` 请求返回的公有密钥下进行加密。

`ImportKeyMaterial 请求`使用以下参数：

```
{
  "EncryptedKeyMaterial": blob,
  "ExpirationModel": "string",
  "ImportToken": blob,
  "KeyId": "string",
  "ValidTo": number
}
```

EncryptedKeyMaterial

导入的密钥材料使用 `GetParametersForImport` 请求中返回的公有密钥和该请求中指定的包装算法进行加密。

ExpirationModel

指定密钥材料是否过期。该值为 `KEY_MATERIAL_EXPIRES` 时，`ValidTo` 参数必须包含到期日期。该值为 `KEY_MATERIAL_DOES_NOT_EXPIRE` 时，不包含 `ValidTo` 参数。有效值为 `"KEY_MATERIAL_EXPIRES"` 和 `"KEY_MATERIAL_DOES_NOT_EXPIRE"`。

ImportToken

提供该公有密钥的同一 `GetParametersForImport` 请求返回的导入令牌。

KeyId

将与导入的密钥材料关联的 KMS 密钥。KMS 密钥的 `Origin` 必须为 `EXTERNAL`。

您可以删除并将同一导入的密钥材料重新导入指定的 KMS 密钥中，但无法导入任何其他密钥材料或将其与 KMS 密钥关联。

ValidTo

（可选）导入的密钥材料过期的时间。当密钥材料过期后， AWS KMS 将删除密钥材料，并且 KMS 密钥将变为不可用。在 `ExpirationModel` 的值为 `KEY_MATERIAL_EXPIRES` 时，则必须提供此参数，否则无效。

请求成功后，如果提供了 KMS 密钥，则可在指定的到期日期 AWS KMS 之前使用。导入的密钥材料过期后，EKT 将从 AWS KMS 存储层中删除。

启用和禁用密钥

禁用 KMS 密钥会阻止在加密操作中使用该密钥。它会暂停使用与 KMS 密钥关联的所有 HBKs 内容的权限。启用后可恢复对 HBKs 和 KMS 密钥的使用。启用和禁用是一种简单的请求，仅需 KMS 密钥的密钥 ID 或密钥 ARN 即可完成。

删除 密钥

授权用户可以使用 ScheduleKeyDeletionAPI 安排删除 KMS 密钥及所有关联密钥 HBKs。这本质上是一种破坏性的操作，从 AWS KMS中删除密钥时应谨慎行事。 AWS KMS 在删除 KMS 密钥时，强制要求最短等待时间为七天。在等待期间，密钥处于禁用状态，密钥状态为待删除。所有使用该密钥进行加密操作的调用都将失败。 ScheduleKeyDeletion 采用以下参数。

```
{
  "KeyId": "string",
  "PendingWindowInDays": number
}
```

KeyId

要删除的 KMS 密钥的唯一标识符。要指定该值，请使用 KMS 密钥的唯一密钥 ID 或密钥 ARN。

PendingWindowInDays

（可选）等待期（单位为天）。该值为可选项。范围为 7-30 天，默认值为 30 天。等待期结束后，AWS KMS 删除 KMS 密钥和所有关联的密钥 HBKs。

轮换密钥材料

授权用户可以为其客户管理型 KMS 密钥启用年度自动轮换。 AWS 托管式密钥 始终会每年轮换一次。

KMS 密钥轮换时，系统将创建一个新的 HBK，并将其标记为所有新加密请求所用密钥材料的当前版本。所有之前版本的 HBK 仍然会永久可供使用，以用于解密使用相应 HBK 版本加密的任何加密文字。由于 AWS KMS 不存储任何在 KMS 密钥下加密的密文，因此在较旧的、轮换的 HBK 下加密的密文需要 HBK 进行解密。您可以通过 ReEncrypt API 以使用 KMS 密钥的新 HBK 或其他 KMS 密钥重新加密任何加密文字，而不暴露明文。

有关启用和禁用密钥轮换的信息，请参阅《AWS Key Management Service 开发人员指南》中的轮换 Amazon KMS 密钥。

客户数据操作

建立 KMS 密钥后，可以使用该密钥执行加密操作。每当在 KMS 密钥下加密数据时，生成的对象就是客户密文。密文包含两个部分：未加密的标头（或明文）部分（作为附加的身份验证数据由经过身份验证的加密方案保护）以及加密部分。明文部分包括 HBK 标识符 (HBKID)。密文值的这两个不可变字段有助于确保将来 AWS KMS 可以解密对象。

主题

- 生成数据密钥
- Encrypt
- Decrypt
- 重新加密加密对象

生成数据密钥

授权用户可以使用 GenerateDataKey API（及相关 APIs）请求特定类型的数据密钥或任意长度的随机密钥。本主题提供此 API 操作的简化视图。有关详细信息，请参阅 AWS Key Management Service API 参考 GenerateDataKey APIs 中的。

- GenerateDataKey
- GenerateDataKeyWithoutPlaintext
- GenerateDataKeyPair
- GenerateDataKeyPairWithoutPlaintext

以下是 `GenerateDataKey` 请求语法。

```
{
 “EncryptionContext”: {“string” : “string”},
 “GrantTokens”: [“string”],
 “KeyId”: “string”,
 “NumberOfBytes”: “number”
}
```

请求接受采用 JSON 格式的以下数据。

KeyId

用于加密数据密钥的密钥的密钥标识符。此值必须标识对称加密 KMS 密钥。

此参数为必需参数。

NumberOfBytes

包含要生成的字节数的整数。此参数为必需参数。

调用者必须提供 `KeySpec` 或 `NumberOfBytes`，但不能同时提供两者。

EncryptionContext

（可选）包含在使用密钥的加密和解密过程中要进行身份验证的其他数据的名称-值对。

GrantTokens

（可选）表示提供生成或使用密钥的权限的授权令牌的列表。有关授权和授权令牌的更多信息，请参阅 AWS Key Management Service 开发人员指南中的 AWS KMS的身份验证和访问控制。

对命令进行身份验证后 AWS KMS，获取与 KMS 密钥关联的当前活动 EKT。它通过 AWS KMS 主机与域中 HSM 之间的受保护会话，将 EKT 以及您提供的请求和任何加密上下文传递给 HSM。

HSM 执行以下操作：

1. 生成请求的机密材料并将其保存在易失性存储器中。
2. 解密与请求中定义的 KMS 密钥的密钥 ID 匹配的 EKT 以获取主动 HBK = Decrypt(DK_i, EKT)。
3. 生成一次性的随机数 N。
4. 从 HBK 和 N 生成 256 位 AES-GCM 派生的加密密钥 K。
5. 加密机密材料 ciphertext = Encrypt(K, context, secret)。

`GenerateDataKey`通过 AWS KMS 主机和 HSM 之间的安全通道将纯文本秘密材料和密文返回给您。 AWS KMS 然后通过 TLS 会话将其发送给您。 AWS KMS 不保留明文或密文。如果没有密文、加密上下文和使用 KMS 密钥的授权，则无法返回底层机密。

以下是响应语法。

```
{
 "CiphertextBlob": "blob",
```

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

数据密钥的管理由您作为应用程序开发人员负责。要使用 AWS KMS 数据密钥（但不是数据密钥对）进行客户端加密的最佳实践，可以使用。AWS Encryption SDK

数据密钥可以以任何频率轮换。此外，数据密钥可以使用 `ReEncrypt` API 操作在不同的 KMS 密钥或轮换的 KMS 密钥下重新加密。有关详细信息，请参阅 AWS Key Management Service API 参考ReEncrypt中的。

Encrypt

的基本功能 AWS KMS 是对 KMS 密钥下的对象进行加密。在设计上， AWS KMS 提供低延迟的加密操作。 HSMs因此，直接调用加密函数时，可以加密的明文量限制为 4 KB。 AWS Encryption SDK 可用于加密较大的邮件。 AWS KMS，在对命令进行身份验证后，获取与 KMS 密钥相关的当前活动 EKT。它将 EKT 连同明文和加密上下文传递给区域中任何可用 HSM。它们通过 AWS KMS 主机与域中的 HSM 之间的经过身份验证的会话发送。

HSM 运行以下操作：

1. 解密 EKT 以获取 HBK = Decrypt(DK_i, EKT)。
2. 生成一次性的随机数 N。
3. 从 HBK 和 N 派生 256 位 AES-GCM 派生的加密密钥 K。
4. 加密明文 ciphertext = Encrypt(K, context, plaintext)。

密文值将返回给您，并且无论是纯文本数据还是密文都不会保留在基础架构中的任何地方。 AWS 如果没有密文、加密上下文和使用 KMS 密钥的授权，则无法返回底层明文。

Decrypt

对解密密文值 AWS KMS 的调用接受加密值 ciphertext 和加密上下文。 AWS KMS 使用AWS 签名版本 4 签名的请求对呼叫进行身份验证，并从密文中提取包装密钥的 HBKID。HBKID 用于获取解密密文、密钥 ID 和密钥 ID 的策略所需的 EKT。请求基于密钥策略、可能存在的授权以及引用密钥 ID 的任何关联 IAM 策略进行授权。`Decrypt` 功能与加密功能类似。

以下是 `Decrypt` 请求语法。

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

以下是请求参数。

CiphertextBlob

包括元数据的密文。

EncryptionContext

（可选）加密上下文。如果在 Encrypt 功能中指定了此参数，则此处也必须指定，否则解密操作将失败。有关更多信息，请参阅 AWS Key Management Service 开发人员指南中的加密内容。

GrantTokens

（可选）表示提供执行解密权限的授权的授权令牌列表。

密文和 EKT 连同加密上下文通过经身份验证的会话发送给 HSM 用于解密。

HSM 运行以下操作：

1. 解密 EKT 以获取 HBK = Decrypt(DK_i, EKT)。
2. 从密文结构中提取一次性的 N。
3. 从 HBK 和 N 重新生成 256 位 AES-GCM 派生的加密密钥 K。
4. 解密密文以获取 plaintext = Decrypt(K, context, ciphertext)。

生成的密钥 ID 和纯文本通过安全会话返回给 AWS KMS 主机，然后通过 TLS 连接返回给调用的客户应用程序。

以下是响应语法。

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

如果调用的应用程序希望确保明文的真实性，则必须验证返回的密钥 ID 是否为预期的密钥 ID。

重新加密加密对象

在一个 KMS 密钥下加密的现有客户密文可以通过重新加密命令重新加密为另一个 KMS 密钥。重新加密使用新的 KMS 密钥加密服务器端的数据，而不公开客户端密钥的明文。将先解密数据，然后再加密。

以下是请求语法。

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

请求接受采用 JSON 格式的以下数据。

CiphertextBlob

要重新加密的数据的密文。

DestinationEncryptionContext

（可选）重新加密数据时要使用的加密上下文。

DestinationKeyId

用于重新加密数据的密钥的密钥标识符。

GrantTokens

（可选）表示提供执行解密权限的授权的授权令牌列表。

SourceKeyId

（可选）用于解密数据的密钥的密钥标识符。

SourceEncryptionContext

（可选）用于加密和解密 `CiphertextBlob` 参数中指定的数据的加密上下文。

该过程将之前描述的解密和加密操作结合起来：客户密文在该客户密文引用的初始 HBK 下解密为预期 KMS 密钥下的当前 HBK。如果此命令中使用的 KMS 密钥相同，则此命令会将客户密文从旧版本的 HBK 移至最新版本的 HBK。

以下是响应语法。

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

如果调用应用程序想要确保底层明文的真实性，则必须验证 SourceKeyId 返回的内容是否符合预期。

AWS KMS 内部运营

AWS KMS 全球分布式密钥管理服务需要内部结构来扩展和确保其安全 HSMs 。

主题

域和域状态

内部可信内部 AWS KMS 实体的合作集合 AWS 区域 称为域。域包括一组受信任的实体、一组规则和一组机密密钥（称为域密钥）。域名密钥在属于 HSMs 该域的成员之间共享。域状态包括以下字段。

名称

用于标识此域的域名。

成员

HSMs 该域成员的列表，包括其公共签名密钥和公共协议密钥。

运算符

代表此服务运营商的实体、公共签名密钥和角色（AWS KMS 操作员或服务主机）的列表。

规则

在 HSM 上运行的每条命令必须满足的仲裁规则列表。

域密钥

域中当前正在使用的域密钥（对称密钥）列表。

完整域状态仅在 HSM 上可用。域状态作为导出的域令牌在 HSM 域成员之间同步。

域密钥

域 HSMs 中的所有人共享一组域密钥 $\{DK_r\}$。这些密钥通过域状态导出例程共享。导出的域状态可以导入作为域成员的任何 HSM 中。

域密钥 {DK_r} 组始终包含一个活动域密钥和多个停用的域密钥。域密钥每天轮换，以确保 AWS 符合密钥管理建议-第 1 部分。域密钥轮换期间，在传出域密钥下加密的所有现有 KMS 密钥将在新的活动域密钥下重新加密。活动域密钥用于加密任何新的域密钥 EKTs。过期的域密钥只能用于解密之前加密 EKTs 的天数，其天数相当于最近轮换的域密钥的数量。

导出的域令牌

我们经常需要在域参与者之间同步状态。这可以通过在对域进行更改时导出域状态来实现。域状态将导出为导出的域令牌。

名称

用于标识此域的域名。

成员

域名成员 HSMs 的列表，包括他们的签名和协议公钥。

运算符

实体、公有签名密钥和代表此服务运营商的角色的列表。

规则

在 HSM 域成员上运行的每条命令必须满足的仲裁规则列表。

加密的域密钥

信封加密的域密钥。域密钥通过上面列出的每个成员的签名成员进行加密，然后封装到其公有协议密钥中。

签名

由 HSM 生成的域状态签名，必须是导出域状态的域成员。

导出的域令牌构成域内运行的实体的基本信任源。

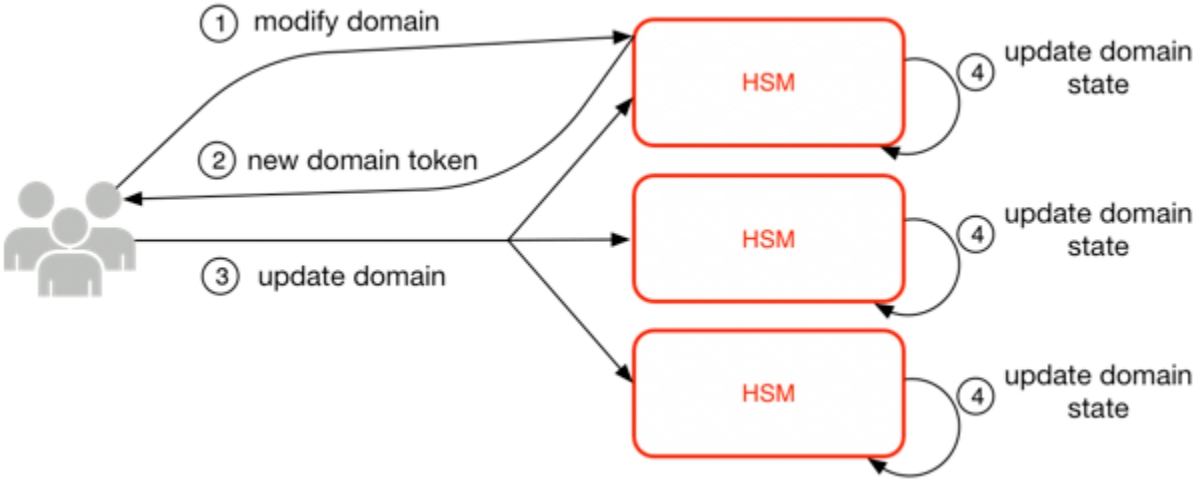
管理域状态

域状态通过经仲裁身份验证的命令进行管理。这些更改包括修改域中受信任参与者的列表、修改用于运行 HSM 命令的仲裁规则以及定期轮换域密钥。这些命令将按每条命令进行身份验证，而不是经过身份验证的会话操作，如下面的图像所示。

处于初始化和运行状态的 HSM 包含一组自行生成的非对称身份密钥、一个签名密钥对和一个密钥建立密钥对。通过手动流程， AWS KMS 操作员可以建立一个要在区域中的第一个 HSM 上创建的初

始域。此初始域包含完整的域状态，如本主题中之前所定义。它通过连接命令安装到域中定义的每个 HSM 成员。

HSM 加入初始域后，它将绑定到该域中定义的规则。这些规则管理使用客户加密密钥或者更改主机或域状态的命令。使用加密密钥的经过身份验证的会话 API 操作在之前已定义。



上图描述了如何修改域状态。该过程包括四个步骤：

1. 向 HSM 发送基于仲裁的命令以修改域。
2. 将生成新的域状态，并将其导出为新的导出域令牌。HSM 上的状态未修改，这意味着更改未在 HSM 上实施。
3. 向新导出的域令牌 HSMs 中的每个人发送第二条命令，以使用新的域令牌更新其域状态。
4. 新导 HSMs 出的域令牌中列出的可以对命令和域令牌进行身份验证。他们还可以解压域密钥以更新域 HSMs中所有人的域状态。

HSMs 不要彼此直接沟通。相反，一定数量的运营商会请求更改域状态，从而生成新的导出域令牌。域的服务主机成员用于将新的域状态分发给域中的每个 HSM。

域的离开和加入通过 HSM 管理功能完成。域状态的修改通过域管理功能完成。

离开域

使 HSM 离开域，从内存中删除该域的所有剩余部分和密钥。

加入域

使 HSM 加入新域或将其当前域状态更新为新域状态。现有域用作对此消息进行身份验证的初始规则集的来源。

创建域

导致在 HSM 上创建新域。返回可以分配给该域成员 HSMs 的第一个域令牌。

修改运营商

从域中授权运营商及其角色的列表中添加或删除运营商。

修改成员

在域中的授权 HSMs 列表中添加或删除 HSM。

修改规则

修改在 HSM 上运行命令所需的仲裁规则集。

轮换域密钥

导致创建新的域密钥并将其标记为活动域密钥。这会将现有活动密钥移动到已停用的密钥，并从域状态中删除最旧的已停用密钥。

内部通信安全

服务主机或 AWS KMS 操作员与之间的命令通过中描述的 HSMs 两种机制进行保护经身份验证的会话：法定签名的请求方法和使用 HSM-Service 主机协议的经过身份验证的会话。

经过法定签名的命令经过精心设计，任何操作员都无法修改其提供的关键安全保护。 HSMs 在经过身份验证的会话中运行的命令可帮助确保只有授权的服务运营商才能执行涉及 KMS 密钥的操作。所有与客户绑定的机密信息在整个 AWS 基础架构中都受到保护。

密钥建立

为了保护内部通信， AWS KMS 使用两种不同的密钥建立方法。第一种方法定义为 Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revision 2) 中的 C(1, 2, ECC DH)。此方案有一个带静态签名密钥的启动程序。启动程序会生成一个临时椭圆曲线 Diffie-Hellman (ECDH) 密钥并签名，旨在用于具有静态 ECDH 协议密钥的收件人。此方法使用一个临时密钥和两个使用 ECDH 的静态密钥。这是标签 C(1, 2, ECC DH) 的衍生。此方法有时称为一次性 ECDH。

第二种密钥建立方法是 C(2, 2, ECC, DH)。在此方案中，双方都有一个静态签名密钥，然后会生成、签名和交换临时 ECDH 密钥。此方法使用两个静态密钥和两个临时密钥（各自使用 ECDH）。这是标签 C(2, 2, ECC, DH) 的衍生。此方法有时称为临时 ECDH 或 ECDHE。所有 ECDH 密钥均在曲线 secp384r1 (NIST-P384) 上生成。

HSM 安全边界

的内部安全边界 AWS KMS 是 HSM。HSM 具有专有接口，在其处于运行状态时没有其他活动物理接口。运行的 HSM 在初始化期间使用必要的加密密钥进行预置，从而在域中建立其角色。HSM 的敏感加密材料仅存储在易失性存储器中，并在 HSM 退出运行状态（包括预期或非预期关机或重置）时擦除。

HSM API 操作可以通过单个命令进行身份验证，也可以通过服务主机建立的相互认证的机密会话进行身份验证。

仲裁签名命令

法定签名的命令由操作员向发出。 HSMs本节介绍如何创建、签名和验证基于仲裁的命令。这些规则相当简单。例如，命令 Foo 需要对角色 Bar 的两名成员进行身份验证。创建和验证基于仲裁的命令有三个步骤。第一步是初始命令创建；第二步是提交给要签名的其他运营商；第三步是验证和执行。

为介绍这些概念，假设有一组真实的运营商公有密钥和角色 $\{QOS_s\}$，以及一组仲裁规则 QR = $\{Command_i, Rule_{\{i,\ t\}}\}$，其中每个 Rule 均为一组角色，且最小数字为 N $\{Role_t, N_t\}$。为使命令满足仲裁规则，命令数据集必须由 $\{QOS_s\}$ 中列出的一组运营商进行签名，以使其满足该命令列出的规则之一。如前所述，仲裁规则和运营商组存储在域状态和导出的域令牌中。

实际上，初始签名者会将命令 $Sig_1 = Sign(dO_{p1}, Command)$ 签名。第二个运营商也会将命令 $Sig_2 = Sign(dO_{p2}, Command)$ 签名。双重签名的消息将发送给 HSM 执行。HSM 将执行以下操作：

1. 对于每个签名，它会从域状态中提取签名者的公有密钥，并验证命令中的签名。
2. 它会验证该组签名者是否满足命令规则。

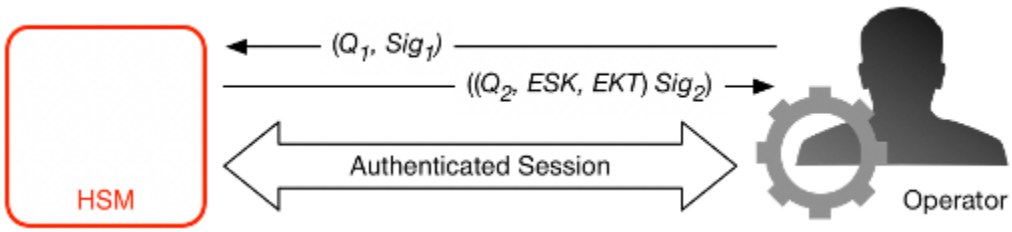
经身份验证的会话

您的密钥操作在面向外部 AWS KMS 的主机和. HSMs 这些命令与加密密钥的创建和使用以及安全随机数生成有关。这些命令在服务主机与之间经过会话验证的通道上运行。 HSMs除了需要真实性以外，这

些会话还需要机密性。这些会话上运行的命令包括返回明文数据密钥和为您解密的消息。为了确保这些会话不会被 man-in-the-middle攻击颠覆，会话需要经过身份验证。

此协议在 HSM 与服务主机之间执行经过相互身份验证的 ECDHE 密钥协议。交换由服务主机发起，并由 HSM 完成。HSM 还返回通过协商密钥加密的会话密钥 (SK) 和包含会话密钥的导出密钥令牌。导出的密钥令牌包含一个有效期，经过该期限后服务主机必须重新协商会话密钥。

服务主机是域的成员，拥有身份签名密钥对（DHO_i、$QHOS_i$）和 “身份 HSMs” 公钥的真实副本。它使用其身份签名密钥组来安全地协商会话密钥，该密钥可在服务主机与域中的任何 HSM 之间使用。导出的密钥令牌有一个与其关联的有效期，经过该期限后必须协商一个新密钥。



该过程从服务主机识别开始，它需要会话密钥才能在自身和域 HSM 成员之间发送和接收敏感通信流。

1. 服务主机会生成 ECDH 临时密钥对 (d_1, Q_1)，并使用其身份密钥 Sig1 = Sign(dOS,Q_1) 进行签名。
2. HSM 使用其当前域令牌验证收到的公有密钥的签名，然后创建 ECDH 临时密钥对 (d_2, Q_2)。然后，它 ECDH-key-exchange根据使用离散对数密码学的配对密钥建立方案建议（修订版）完成以形成协商的256位AES-GCM 密钥。HSM 会生成新的 256 位 AES-GCM 会话密钥。它使用协商密钥来加密会话密钥，从而形成加密的会话密钥 (ESK)。它还加密域密钥下的会话密钥作为导出的密钥令牌 EKT。最后，它使用其身份密钥对 Sig_2 = Sign(dHSK, (Q_2, ESK, EKT)) 将返回值签名。
3. 服务主机使用其当前域令牌验证收到的密钥的签名。然后，服务主机会根据 Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised) 完成 ECDH 密钥交换。接下来，它会解密 ESK 以获取会话密钥 SK。

在 EKT 的有效期内，服务主机可使用协商的会话密钥 SK 向 HSM 发送信封加密的命令。此经过身份验证的会话上的每个 service-host-initiated命令都包含 EKT。HSM 使用相同的协商会话密钥 SK 进行响应。

多区域密钥的复制过程

AWS KMS 使用跨区域复制机制将 KMS 密钥中的密钥材料从一个 HSM 复制 AWS 区域 到另一个密钥中的 HSM。 AWS 区域要使此机制工作，要复制的 KMS 密钥必须是多区域密钥。将 KMS 密钥从一个

区域复制到另一个区域时，区域 HSMs 中的无法直接通信，因为它们位于隔离的网络中。相反，在跨区域复制期间交换的消息将由代理服务传送。

在跨区域复制期间， AWS KMS HSM 生成的每条消息都使用复制签名密钥进行加密签名。复制签名密钥 (RSKs) 是 NIST P-384 曲线上的 ECDSA 密钥。每个区域至少拥有一个 RSK，并且每个 RSK 的公共组件与同一 AWS 分区中的所有其他区域共享。

将密钥材料从区域 A 复制到区域 B 的跨区域复制过程如下：

1. 区域 B 中的 HSM 在 NIST P-384 曲线上生成一个临时的 ECDH 密钥，即复制协议密钥 B (RAKB)。RAKB 的公有组件由代理服务发送到区域 A 中的 HSM。
2. 区域 A 中的 HSM 将接收 RAKB 的公有组件，然后在 NIST P-384 曲线上生成另一个临时的 ECDH 密钥，即复制协议密钥 A (RAKA)。HSM 将在 RAKA 和 RAKB 的公有组件上运行 ECDH 密钥建立方案，并从输出中派生一个对称密钥，即复制包装密钥 (RWK))。RWK 用于对即将复制的多区域 KMS 密钥的密钥材料进行加密。
3. RAKA 的公有组件和使用 RWK 加密的密钥材料将通过代理服务发送到区域 B 中的 HSM。
4. 区域 B 中的 HSM 将接收 RAKA 的公有组件和使用 RWK 加密的密钥材料。通过在 RAKB 和 RAKA 的公有组件上运行 ECDH 密钥建立方案，RWK 将派生 HSM。
5. 区域 B 中的 HSM 将使用 RWK 解密来自区域 A 的密钥材料。

持久性保护

通过使用导出的域令牌的离线 HSMs、多个非易失性存储以及加密的 KMS 密钥的冗余存储，为该服务生成的密钥提供了额外的服务持久性。离线 HSMs 用户是现有域的成员。除了不在线并参与常规域名操作外，脱机用户在域状态下 HSMs 显示的状态与现有 HSM 成员相同。

耐久性设计旨在保护一个区域中的所有 KMS 密钥，以 AWS 防在线密钥 HSMs 或存储在我们的主存储系统中的 KMS 密钥集大量丢失。 AWS KMS keys 导入的密钥材料不包括在其他 KMS 密钥提供的耐久性保护下。如果出现区域性故障 AWS KMS，则可能需要将导入的密钥材料重新导入 KMS 密钥中。

离线 HSMs数据和访问凭证存储在多个独立地理位置的受监控安全室内的保险箱中。每个保险箱都需要来自两个独立小组的至少一名 AWS 保安人员和一名 AWS KMS 操作员来获取这些材料。 AWS这些材料的使用受内部政策的约束，该政策要求有法定人数的 AWS KMS 操作员在场。

参考

使用以下参考资料获取有关本文档中引用的缩写、密钥、参与者和来源。

主题

缩写

以下列表说明了本文档中引用的缩写。

AES

高级加密标准

CDK

客户数据密钥

DK

域密钥

ECDH

椭圆曲线 Diffie-Hellman

ECDHE

临时椭圆曲线 Diffie-Hellman

ECDSA

椭圆曲线数字签名算法

EKT

导出密钥令牌

ESK

加密会话密钥

GCM

伽罗瓦计数器模式

HBK

HSM 备用密钥

HBKID

HSM 备用密钥标识符

HSM

硬件安全模块

RSA

Rivest、Shamir 和 Adleman（密码学）

secp384r1

高效密码学标准素数 384 位随机曲线 1

SHA256

摘要长度为 256 位的安全哈希算法

键

以下列表定义了本文档中引用的密钥。

HBK

HSM 备用密钥：HSM 备用密钥是 256 位根密钥，从中派生特定用途密钥。

DK

域密钥：域密钥是一个 256 位的 AES-GCM 密钥。它在所有域成员之间共享，用于保护 HSM 备用密钥材料和 HSM 服务主机会话密钥。

DKEK

域密钥加密密钥：域密钥加密密钥是在主机上生成的 AES-256-GCM 密钥，用于加密跨 HSM 主机同步域状态的当前域密钥组。

(dHAK,QHAK)

HSM 协议密钥对：每个启动的 HSM 在曲线 secp384r1 (NIST-P384) 上都有一个本地生成的椭圆曲线 Diffie-Hellman 协议密钥对。

(dE, QE)

临时协议密钥对：HSM 和服务主机会生成临时协议密钥。这些密钥是曲线 secp384r1 (NIST-P384) 上的椭圆曲线 Diffie-Hellman 密钥。它们是在两个用例中生成的：建立 host-to-host加密密钥以在域令牌中传输域密钥加密密钥，以及建立 HSM-Service 主机会话密钥以保护敏感通信。

(dHSK,QHSK)

HSM 签名密钥对：每个启动的 HSM 在曲线 secp384r1 (NIST-P384) 上都有一个本地生成的椭圆曲线数字签名密钥对。

(dOS,QOS)

Operator signature key pair：服务主机 AWS KMS 运营商和运营商都有一个身份签名密钥，用于向其他域参与者进行身份验证。

K

数据加密密钥：一个 256 位 AES-GCM 密钥，源自 HBK，在计数器模式下使用 NIST SP8 00-108 KDF，使用 HMAC 和。 SHA256

SK

会话密钥：创建会话密钥是服务主机运营商与 HSM 之间交换经身份验证的椭圆曲线 Diffie-Hellman 密钥的结果。交换的目的是保护服务主机与域成员之间的通信。

贡献者

以下个人和组织参与了本文档的编撰：

- Ken Beer，KMS AWS 密码学总经理
- Matthew Campagna，密码学首席安全工程师 AWS

参考书目

有关信息 AWS Key Management Service HSMs，请访问 NIST 计算机安全资源中心加密模块验证计划搜索页面并搜索 HS AWS Key Management Service M。

亚马逊 Web Services，一般参考（版本 1.0），“签署 AWS API 请求”，http://docs.aws.amazon。com/general/latest/gr/signing_aws_api_requests.html。

亚马逊 Web Services，“这是什么” AWS Encryption SDK，http://docs.aws.amazon。 com/encryption-sdk/latest/developer-guide/introduction.html。

美国联邦信息处理标准出版物，FIPS PUB 180-4。Secure Hash Standard，2012 年 8 月。可从 https://nvlpubs.nist 获得。 gov/nistpubs/FIPS/NIST.fips.180 -4.pdf。

美国联邦信息处理标准出版物 197，Announcing the Advanced Encryption Standard (AES)，2001 年 11 月。可从 http://csrc.nist 获得。 gov/publications/fips/fips197/fips-197.pdf。

美国联邦信息处理标准出版物 198-1，The Keyed-Hash Message Authentication Code (HMAC)，2008 年 7 月。可从 http://csrc.nist 获得。 gov/publications/fips/fips198-1/FIPS-198-1_final.pdf。

NIST 特别出版物 800-52 修订版 2，《传输层安全 (TLS) 实施选择、配置和使用指南》，2019 年 8 月。 https://nvlpubs.nist。 gov/nistpubs/SpecialPublications/NIST.sp.800-52r2.pdf。

PKCS #1 v2.2：RSA 密码学标准 (RFC 8017)，互联网工程任务组 (IETF)，2016 年 11 月。 https://tools.ietf.org/html/rfc8017。

Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC，NIST 特殊出版物 800-38D，2007 年 11 月。可从 http://csrc.nist 获得。 gov/publications/nistpubs/800-38D/SP-800-38D.pdf。

Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices，NIST 特殊出版物 800-38E，2010 年 1 月。可从 https://nvlpubs.nist 获得。 gov/nistpubs/Legacy/SP/nistspecialpublication800-38e.pdf。

关于@@ 使用伪随机函数推导密钥的建议，NIST 特别出版物 800-108，2009 年 10 月，可从 https://nvlpubs.nist 获得。 gov/nistpubs/legacy/sp/nistspecialpublication800-108.pdf。

Recommendation for Key Management - Part 1: General (Revision 5)，NIST 特殊出版物 800-57A，2020 年 5 月，可从以下网址获得：https://doi.org/10.6028/NIST.SP.800-57pt1r5。

Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised)，NIST 特殊出版物 800-56A 修订版 3，2018 年 4 月。可从 https://nvlpubs.nist 获得。 gov/nistpubs/SpecialPublications/NIST.sp.800-56ar3.pdf。

使用确定性随机位生成器生成随机数的建议，NIST 特别出版物 800-90A 修订版 1，2015 年 6 月，可从 https://nvlpubs.nist 获得。 gov/nistpubs/SpecialPublications/NIST.sp.800-90ar1.pdf。

SEC 2: Recommended Elliptic Curve Domain Parameters, Standards for Efficient Cryptography Group，版本 2.0，2010 年 1 月 27 日。

在@@ 加密消息语法 (CMS) 中使用椭圆曲线密码学 (ECC) 算法，Brown，D.，Turner，S.，互联网工程任务组，2010 年 7 月，http://tools.ietf。 org/html/rfc5753/。

X9.62-2005: Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)，美国国家标准协会，2005 年。

AWS KMS 加密详细信息的文档历史记录

下表介绍了对 AWS Key Management Service Cryptographic Details 文档的一些重要更改。我们还经常更新文档以处理您发送给我们的反馈意见。

变更	说明	日期
更新的内容	添加了有关 AWS KMS `ReplicateKey` 操作实现的详细信息。	2021 年 10 月 28 日
文档更改	将术语客户主密钥 (CMK) 替换为 AWS KMS key 和 KMS 密钥。	2021 年 8 月 30 日
初始版本	根据 KMS Cryptographic Details 技术白皮书创建了本指南	2020 年 12 月 30 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。