AWS KMS 密碼編譯詳細資訊

AWS Key Management Service

AWS Key Management Service: AWS KMS 密碼編譯詳細資訊

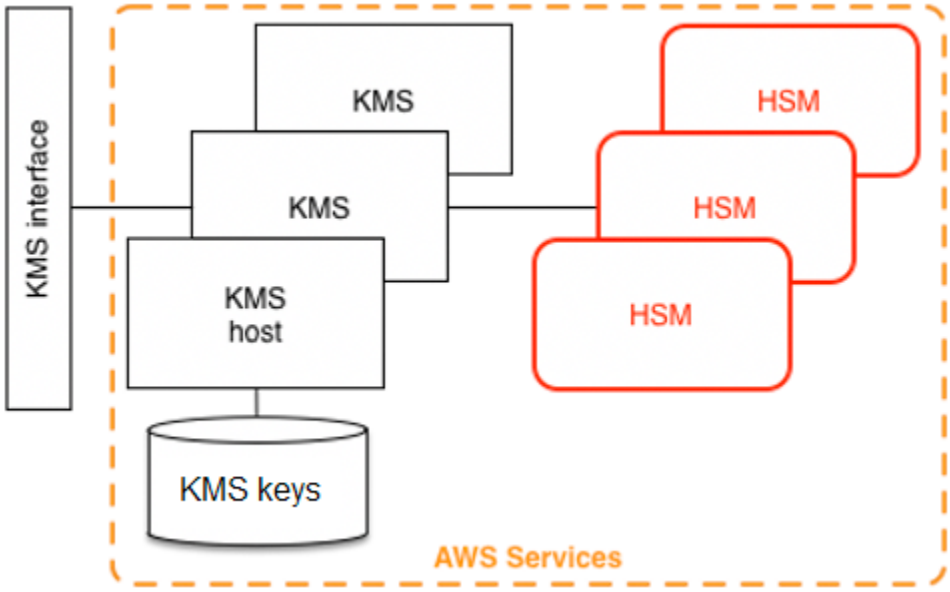
Table of Contents

AWS KMS 密碼編譯詳細資訊簡介

AWS Key Management Service(AWS KMS) 提供 Web 界面來產生和管理密碼編譯金鑰，並以密碼編譯服務供應商的身分運作，以保護資料。AWS KMS 提供傳統的金鑰管理服務，並與 AWS 服務整合，為提供跨 AWS 的一致客戶金鑰視圖，擁有集中式管理和稽核功能。本白皮書提供 AWS KMS 密碼編譯操作的詳細說明，以協助您評估服務所提供的功能。

AWS KMS 包含透過 AWS Management Console 的 Web 界面、命令列界面和 RESTful API 操作，以請求 FIPS 140-2 經驗證硬體安全模組 (HSM) 分散式機群的密碼編譯操作 [1]。AWS KMS HSM 是一款多晶片獨立硬體密碼編譯設備，旨在提供專用的密碼編譯功能，以滿足 AWS KMS 的安全和可擴展性需求。您可以在作為 AWS KMS keys 管理的金鑰下建立自己的 HSM 式密碼編譯階層。這些金鑰只能在處理密碼編譯請求所需的必要時間內在 HSM 上和記憶體內提供。您可以建立多個 KMS 金鑰，每個金鑰都以其金鑰 ID 表示。僅限於在由每個客戶管理的 AWS IAM 角色和帳戶下，可以建立、刪除客戶 KMS 金鑰，或用來加密、解密、簽署或驗證資料。您可以藉由建立連接至金鑰的政策，來定義誰可以管理和/或使用 KMS 金鑰的存取控制項。這些政策可讓您為每個 API 操作定義金鑰的應用程式特定用途。

除此之外，大多數 AWS 服務支援使用 KMS 金鑰對靜態資料加密。此功能可讓客戶控制 AWS 服務何時可以如何透過控制存取 KMS 金鑰的方式和時間來存取加密的資料。



AWS KMS 是一種分層服務，包含面向 Web 的 AWS KMS 主機和一層 HSM。這些分層主機的群組會形成 AWS KMS 堆疊。所有 AWS KMS 請求必須透過 Transport Layer Security 通訊協定 (TLS) 提出，並在 AWS KMS 主機終止。AWS KMS 主機只允許使用提供完整轉寄密碼的密碼套件。AWS KMS

會使用 AWS Identity and Access Management (IAM) 的相同憑證和政策機制 (可用於所有其他 AWS API 操作) 來驗證和授權您的請求。

基本概念

學習一些基本術語與概念將協助您充分利用 AWS Key Management Service。

AWS KMS key

> **Note**
>
> AWS KMS 正在使用 AWS KMS key 和 KMS 金鑰取代術語客戶主要金鑰 (CMK)。概念並沒有變更。為了防止重大變更，AWS KMS 會保留此術語的一些變化。

代表金鑰階層頂端的邏輯金鑰。賦予 KMS 金鑰一個 Amazon Resource Name (ARN)，其中包含唯一金鑰識別符或金鑰 ID。AWS KMS keys 有三種類型：

- 客戶管理的金鑰 – 客戶建立並控制客戶受管金鑰的生命週期和重要政策。針對這些金鑰發出的所有要求都會記錄為 CloudTrail 事件。
- AWS 受管金鑰 – AWS 會建立並控制 AWS 受管金鑰 的生命週期和金鑰政策，是客戶 AWS 帳戶中的資源。客戶可以檢視的存取政策和 CloudTrail 事件AWS 受管金鑰，但無法管理這些金鑰的任何層面。針對這些金鑰發出的所有要求都會記錄為 CloudTrail 事件。
- AWS 擁有的金鑰 – 這些金鑰是由 AWS 建立，並專門使用以在不同 AWS 服務之間進行內部加密操作。客戶無法掌握中的關鍵政策或AWS 擁有的金鑰使用情況 CloudTrail。

別名

與 KMS 金鑰相關聯的使用者易記名稱。別名可以在許多 AWS KMS API 操作中與金鑰 ID 交換搭配使用。

許可

連接至 KMS 金鑰的政策，用於定義金鑰的許可。預設政策允許您定義的任何主體，並允許 AWS 帳戶 新增參照金鑰的 IAM 政策。

授權

一開始預期的 IAM 主體或使用持續時間未知並因此新增至金鑰或 IAM 政策時使用 KMS 金鑰的委派許可。授權的用途之一是定義範圍下移的許可，以了解 AWS 服務可以如何使用 KMS 金鑰。在

沒有直接簽署之 API 呼叫的情況下，服務可能需要使用您的金鑰代表您對加密的資料執行非同步工作。

資料金鑰

在 HSM 上產生的加密金鑰，受 KMS 金鑰保護。AWS KMS 允許授權的實體取得受 KMS 金鑰保護的資料金鑰。它們可以同時以純文字 (未加密) 資料金鑰和加密資料金鑰傳回。資料金鑰可以是對稱的或非對稱的 (同時傳回公有和私有部分)。

加密文字

AWS KMS 的加密輸出，有時也稱為客戶加密文字，以消除混淆。加密文字包含加密的資料，其中包含識別要在解密程序中使用之 KMS 金鑰的其他資訊。加密的資料金鑰是使用 KMS 金鑰時產生之加密文字的一個常見範例，但任何大小小於 4 KB 的資料都可以在 KMS 金鑰下加密，以產生加密文字。

加密內容

與受 AWS KMS 保護資訊關聯之其他資訊的鍵值對映射。AWS KMS 使用經過驗證的加密來保護資料金鑰。加密內容會納入 AWS KMS 加密文字中已驗證加密的 AAD 中。此內容資訊是選用的，不會在請求金鑰 (或加密操作) 時傳回。但如果使用，則需要此內容值才能成功完成解密操作。加密內容預期用於提供其他經驗證的資訊。此資訊可協助您強制執行原則並包含在AWS CloudTrail 記錄檔中。例如，您可以使用 {"key name":"satellite uplink key"} 的鍵值對來命名資料金鑰。後續使用的金鑰會建立 AWS CloudTrail 項目，其中包含 "key name":"satellite uplink key"。此額外資訊可提供有用的內容，以了解為何使用指定的 KMS 金鑰。

公有金鑰

使用非對稱密碼 (RSA 或橢圓曲線) 時，公有金鑰是公有-私有金鑰對的「公有元件」。公有金鑰可以共用並分配至需要為公有-私有金鑰對擁有者加密資料的實體。對於數位簽章操作，公有金鑰用於驗證簽章。

私有金鑰

使用非對稱密碼 (RSA 或橢圓曲線) 時，私有金鑰是公有-私有金鑰對的「私有元件」。私有金鑰用於解密資料或建立數位簽章。與對稱 KMS 金鑰類似，私有金鑰會在 HSM 中加密。其只會解密到 HSM 的短期記憶體中，並且僅在處理您的密碼編譯請求所需的時間內。

AWS KMS 設計目標

AWS KMS 專為滿足以下需求而設計。

耐久性

密碼編譯金鑰的耐久性專門設計為等同於 AWS 中最具耐久性的服務。單一密碼編譯金鑰可以加密長時間累積的大量資料。

值得信賴

金鑰的使用受到您定義和管理之存取控制政策的保護。沒有任何機制可匯出純文字 KMS 金鑰。密碼編譯金鑰的機密性至關重要。若要對 HSM 執行管理動作，需要有多名 Amazon 員工擁有對以仲裁為基礎之存取控制項的角色特定存取權。

低延遲和高輸送量

AWS KMS 以延遲和適合 AWS 中其他服務使用的輸送量層級提供密碼編譯操作。

獨立區域

AWS 為需要限制不同區域資料存取的客戶提供獨立區域。可以在 AWS 區域 內隔離金鑰用量。

隨機數字的安全來源

由於強大的密碼編譯取決於真正不可預測的隨機數字產生，所以 AWS KMS 提供了高品質且經過驗證的隨機數字來源。

稽核

AWS KMS在記錄AWS CloudTrail 檔中記錄密碼編譯金鑰的使用和管理。您可以使用 AWS CloudTrail 日誌來檢查密碼編譯金鑰的使用情況，包括 AWS 服務代表您對金鑰的使用。

為了實現這些目標，AWS KMS 系統包含一組管理「網域」的 AWS KMS 電信業者和服務主機電信業者 (統稱「電信業者」)。網域是一組區域定義的 AWS KMS 伺服器、HSM 和電信業者。每位 AWS KMS 電信業者具有硬體字符，其中包含用來驗證其動作的私有和公有金鑰對。HSM 具有額外的私有和公有金鑰對，可建立保護 HSM 狀態同步的加密金鑰。

本白皮書說明了 AWS KMS 如何保護您要加密的金鑰和其他資料。在本文件中，您要加密的加密金鑰或資料稱為「機密」或「機密材料」。

AWS Key Management Service 基礎

本章中的主題描述了 AWS Key Management Service 的密碼編譯基本元素及其在何處使用。還介紹了 AWS KMS 的基本元素。

主題

密碼編譯基本元素

AWS KMS 使用可設定的密碼編譯演算法，讓系統可以快速地從一個核准的演算法或模式遷移到另一個演算法。針對安全屬性和效能，已從聯邦資訊處理標準 (經 FIPS 核准) 演算法中選取初始預設的密碼編譯演算法集。

熵和隨機數字產生

在 AWS KMS HSM 上執行 AWS KMS 金鑰產生動作。HSM 會實作混合式隨機數字產生器，其使用採用 AES-256 的 NIST SP800-90A 決定性隨機位元產生器。它植入了具有 384 位元熵的非決定性隨機位元產生器，並使用額外的熵進行更新，以便在每次呼叫加密材料時提供預測阻力。

對稱金鑰操作 (僅限加密)

HSM 中使用的所有對稱金鑰加密命令都會使用進階加密標準 (AES)，採用 256 位元金鑰的 Galois 計數器模式 (GCM)。要解密的類似呼叫使用反函數。

AES-GCM 是經過驗證的加密配置。除了加密純文字以產生加密文字，它還會針對加密文字和需要身分驗證的任何其他資料 (另外驗證的資料或 AAD) 運算身分驗證標籤。身分驗證標籤有助於確保資料來自所宣稱的來源，而且加密文字和 AAD 尚未修改。

通常，AWS 會忽略在說明中包含 AAD 內容，尤其是在提及資料金鑰的加密時。這些情況下的周圍文字會暗示，在要加密純文字和要保護純文字 AAD 之間會進行分區的要加密結構。

AWS KMS 提供了一個選項，讓您可以將金鑰材料匯入 AWS KMS key，而不是依賴 AWS KMS 來產生金鑰材質。此匯入的金鑰可以使用 RSAES-OAEP 或 RSAES-PKCS1-v1_5 來加密，以便於傳輸到 AWS KMS HSM 期間對金鑰提供保護。RSA 金鑰對會在 AWS KMS HSM 上產生。匯入的金鑰材料會在 AWS KMS HSM 上解密，並在由服務存放之前根據 AES-GCM 重新加密。

非對稱金鑰操作 (加密、數位簽章和簽章驗證)

AWS KMS 支援使用非對稱金鑰操作進行加密和數位簽章操作。非對稱金鑰操作依賴於數學相關的公有金鑰和私有金鑰對，可用於加密和解密或簽署和簽章驗證，但不能同時用於兩者。私有金鑰絕不會讓 AWS KMS 出現未加密的狀況。您可以呼叫 AWS KMS API 操作以使用 AWS KMS 內的公有金鑰，或者下載公有金鑰，在 AWS KMS 外面使用。

AWS KMS 支援兩種類型的非對稱加密。

- RSA-OAEP (用於加密) 與 RSA-PSS 和 RSA-PKCS-#1-v1_5 (用於簽署和驗證) – 針對不同的安全要求，支援 RSA 金鑰長度 (以位元為單位)：2048、3072 和 4096。
- 橢圓曲線 (ECC) – 專用於簽署和驗證。支援 ECC 曲線：NIST P256、P384、SECP 256k1。

金鑰衍生函數

密鑰衍生函數用於從初始機密或金鑰衍生額外的金鑰。AWS KMS 使用金鑰衍生函數 (KDF) 來衍生 AWS KMS key 下每次加密的每個呼叫金鑰。所有 KDF 操作均會使用 計數器模式中的 KDF (使用 HMAC [FIPS197] 與 SHA256 [FIPS180])。256 位元衍生金鑰與 AES-GCM 搭配使用，以加密或解密客戶資料和金鑰。

數位簽章的 AWS KMS 內部使用

數位簽章也可用來對命令和 AWS KMS 實體之間的通訊進行身分驗證。所有服務實體都有橢圓曲線數位簽章演算法 (ECDSA) 金鑰對。其執行密碼編譯訊息語法 (CMS) 中橢圓曲線密碼編譯 (ECC) 演算法的使用和 X9.62-2005：金融服務業的公有金鑰密碼編譯：橢圓曲線數位簽章演算法 (ECDSA)中所定義的 ECDSA。實體使用聯邦資訊處理標準出版物 FIPS PUB 180-4 (稱為 SHA384) 中所定義的安全雜湊演算法。金鑰於曲線 secp384r1 (NIST-P384) 產生。

封套加密

在許多密碼編譯系統中使用的基本結構是信封加密。信封加密使用兩個或多個密碼編譯金鑰來保護訊息。通常，一個金鑰是從較長期的靜態金鑰 k，另一個金鑰是每條訊息金鑰 msgKey (這是為了加密訊息而產生的)。信封是透過加密訊息：ciphertext = Encrypt(msgKey, message) 形成的。然後訊息金鑰使用長期靜態金鑰：encKey = Encrypt(k, msgKey) 進行加密。最後，將這兩個值 (encKey, ciphertext) 封裝成單一結構或信封加密訊息。

收件人 (可存取 k) 可以先解密加密的金鑰，然後解密訊息，以開啟封住的訊息。

AWS KMS 可讓您管理這些較長期的靜態金鑰，並自動化資料的信封加密程序。

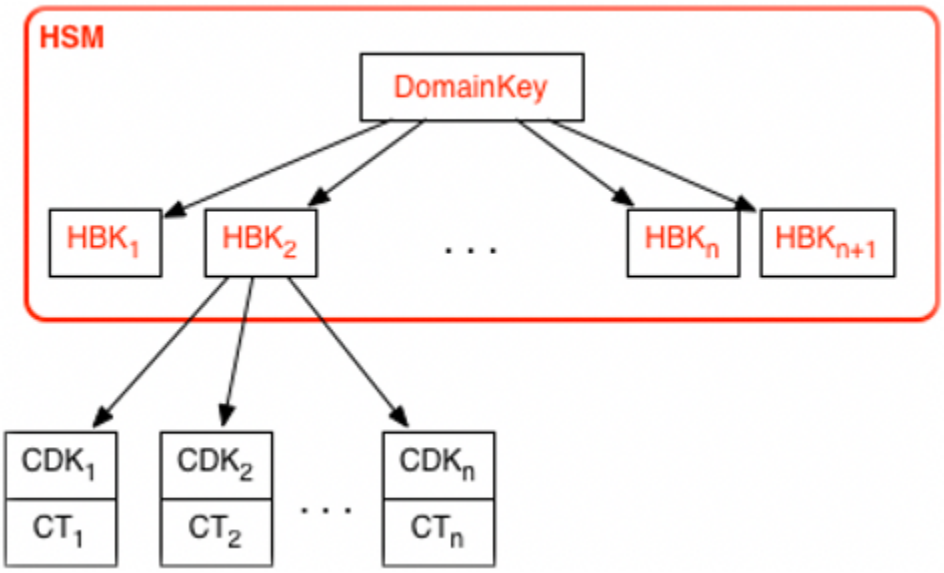
除了在 AWS KMS 服務內提供的加密功能，AWS 加密開發套件還提供用戶端側的信封加密程式庫。您可以使用這些程式庫來保護您的資料和用來加密該資料的加密金鑰。

AWS KMS key 階層

您的金鑰階層從頂層邏輯金鑰 AWS KMS key 開始。KMS 金鑰代表頂層金鑰材料的容器，並且在 AWS 服務命名空間內使用 Amazon Resource Name (ARN) 進行專屬定義。ARN 包括專屬產生的金鑰識別符，金鑰 ID。KMS 金鑰是根據使用者啟動的請求透過 AWS KMS 建立的。接收後，AWS KMS 請求建立初始 HSM 備份金鑰 (HBK)，以便放置在 KMS 金鑰容器中。HBK 是在網域中的 HSM 上產生的，並且設計為永遠不會以純文字形式從 HSM 匯出。相反地，HBK 會在受 HSM 管理的網域金鑰下加密匯出。這些匯出的 HBK 稱為匯出的金鑰字符 (EKT)。

EKT 會匯出至高度耐用、低延遲的儲存體。例如，假設您收到邏輯 KMS 金鑰的 ARN。這代表您的金鑰階層或密碼編譯內容的頂端。您可以在帳戶中建立多個 KMS 金鑰，並在 KMS 金鑰上設定政策，就像任何其他 AWS 命名資源一樣。

在特定 KMS 金鑰的階層中，HBK 可以被視為 KMS 金鑰的版本。當您想要透過 AWS KMS 輪換 KMS 金鑰，則會建立新的 HBK，並與 KMS 金鑰關聯，作為 KMS 金鑰的作用中 HBK。系統會保留較舊的 HBK，可用來解密和驗證先前受保護的資料。但是，只有作用中的密碼編譯金鑰可以用來保護新的資訊。



您可以透過 AWS KMS 提出請求，以使用 KMS 金鑰直接保護資訊，或請求其他受 KMS 金鑰保護之 HSM 產生的金鑰。這些金鑰稱為客戶資料金鑰 (CDK)。CDK 可以傳回加密為加密文字 (CT)，純文字，或兩者兼而有之。在 KMS 金鑰下加密的所有物件 (客戶提供的資料或 HSM 產生的金鑰) 只能透過 AWS KMS 呼叫在 HSM 上進行解密。

傳回的加密文字或解密的酬載永遠不會存放在 AWS KMS 內。資訊會透過至 AWS KMS 的 TLS 連線傳回給您。這也適用於 AWS 服務代表您進行的呼叫。

金鑰階層和特定金鑰屬性會出現在下表中。

金鑰	描述	生命週期
網域金鑰	僅在 HSM 記憶體中用來包裝 KMS 金鑰 (HSM 備份金鑰) 版本的 256 位元 AES-GCM 金鑰。	每日輪換 [1]
HSM 備份金鑰	256 位元對稱金鑰或 RSA 或橢圓曲線私有金鑰，用於保護客戶資料和金鑰，並以網域金鑰加密存放。一個或多個 HSM 備份金鑰由 KMS 金鑰組成，用 keyId 表示。	每年輪換 [2] (選用組態)
衍生的加密金鑰	僅在 HSM 記憶體中用來加密客戶資料和金鑰的 256 位元 AES-GCM 金鑰。從每個加密的 HBK 衍生。	每次加密使用一次，並在解密時重新產生
客戶資料金鑰	以純文字和加密文字從 HSM 匯出之使用者定義的對稱或非對稱金鑰。 在 HSM 備份金鑰下加密，並透過 TLS 通道傳回給授權的使用者。	輪換和使用由應用程式控制

[1] AWS KMS 可能會不時地將網域金鑰輪換放鬆到最多每週一次，以便進行網域管理和組態任務。

[2] 由 AWS KMS 代表您建立和管理的預設 AWS 受管金鑰 會每年自動輪換。

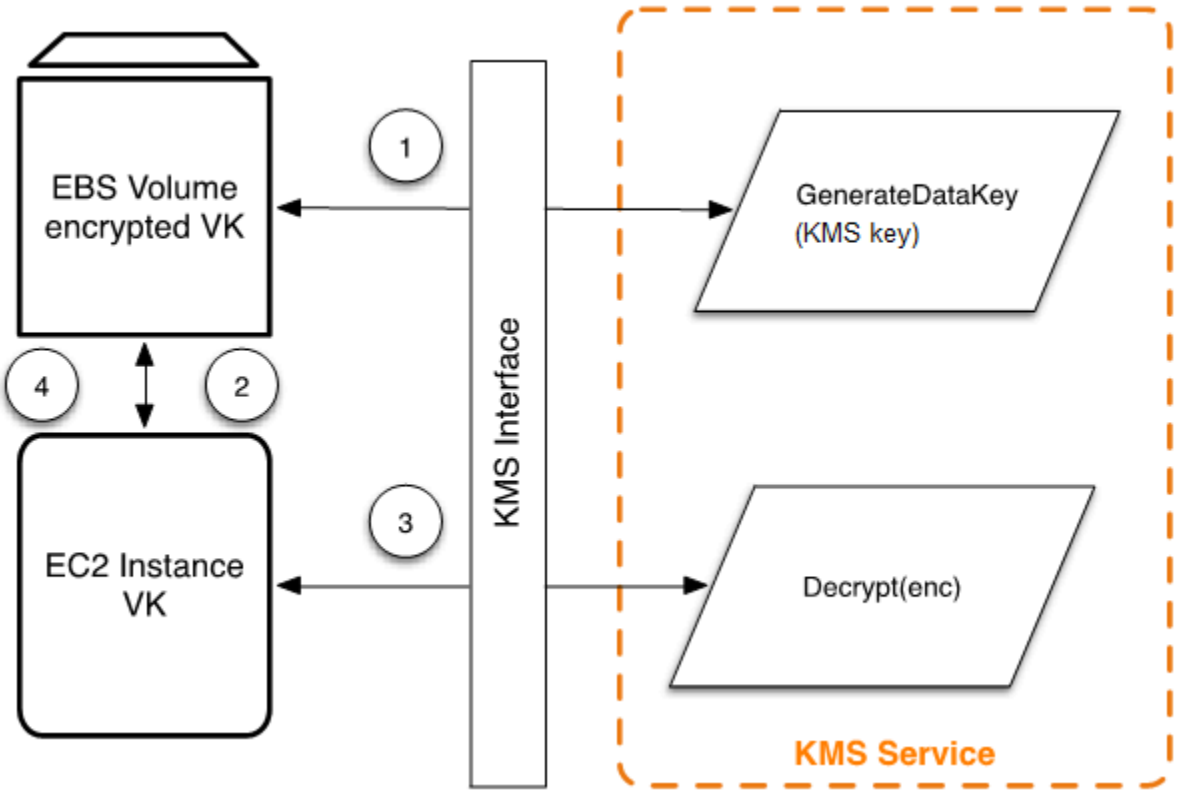
AWS KMS 使用案例

使用案例可協助您充分利用 AWS Key Management Service。第一個示範 AWS KMS 如何在 Amazon Elastic Block Store (Amazon EBS) 磁碟區使用 AWS KMS keys 執行伺服器端加密。第二個是用戶端應用程式，示範如何使用信封加密保護具有 AWS KMS 的內容。

主題

Amazon EBS 磁碟區加密

Amazon EBS 提供磁碟區加密功能。每個磁碟區都使用 AES-256-XTS 加密。這需要兩個 256 位元磁碟區金鑰，您可以將其視為一個 512 位元磁碟區金鑰。磁碟區金鑰會在您帳戶中的 KMS 金鑰下加密。為了讓 Amazon EBS 為您加密磁碟區，它必須具有存取權，才能在帳戶中的 KMS 金鑰下產生磁碟區金鑰 (VK)。您可以透過提供 Amazon EBS 授權給 KMS 金鑰，以建立資料金鑰，並加密和解密這些磁碟區金鑰。現在 Amazon EBS 將 AWS KMS 與 KMS 金鑰搭配使用來產生 AWS KMS 加密的磁碟區金鑰。



下列工作流程會對寫入 Amazon EBS 磁碟區的資料進行加密：

1. Amazon EBS 透過 TLS 工作階段的 AWS KMS 在 KMS 金鑰下取得加密的磁碟區金鑰，並存放具有磁碟區中繼資料的加密金鑰。
2. 掛載 Amazon EBS 磁碟區時，會擷取加密的磁碟區金鑰。
3. 透過 TLS 的 AWS KMS 呼叫來解密加密的磁碟區金鑰。AWS KMS 會識別 KMS 金鑰，並向機群中的 HSM 提出內部請求，以解密加密的磁碟區金鑰。然後，AWS KMS 將磁碟區金鑰傳回 Amazon Elastic Compute Cloud (Amazon EC2) 主機，該主機會包含 TLS 工作階段的執行個體。
4. 磁碟區金鑰用於加密和解密所有進出已連接 Amazon EBS 磁碟區的資料。Amazon EBS 會保留加密的磁碟區金鑰，以便稍後在記憶體中的磁碟區金鑰不再可用時使用。

如需使用 KMS 金鑰加密 Amazon EBS 磁碟區的詳細資訊，請參閱AWS Key Management Service開發人員指南中的 Amazon Elastic Block Store 如何使用AWS KMS，以及 Amazon EC2 Linux 執行個體使用者指南和 Amazon EC2 Windows 執行個體使用者指南中的 Amazon EBS 加密。

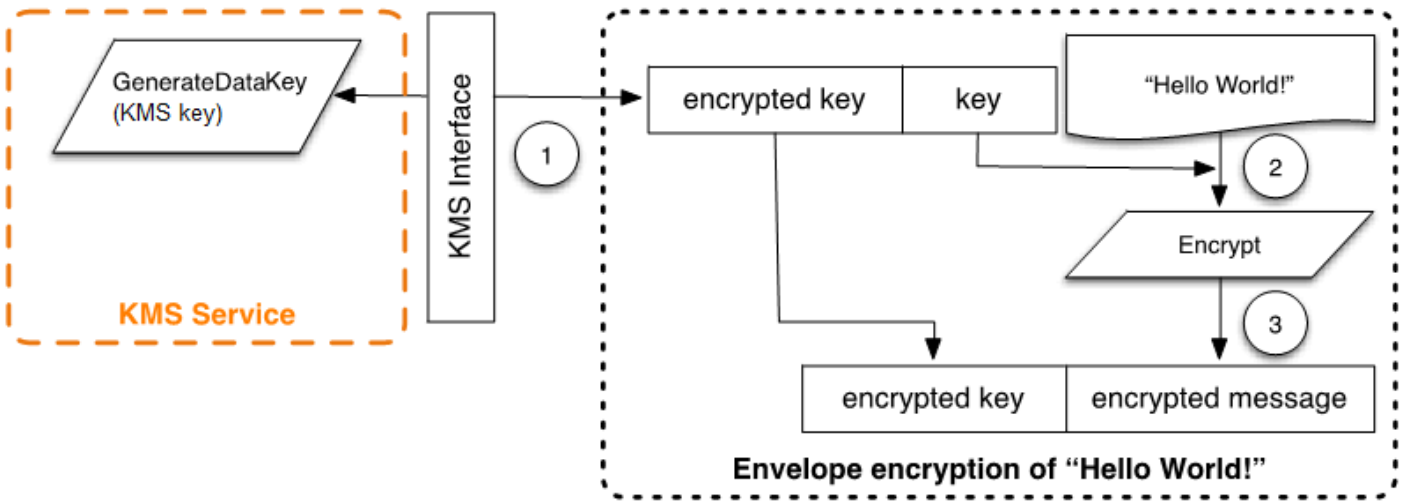
用戶端加密

AWS Encryption SDK 包含使用 KMS 金鑰執行信封加密的 API 操作。如需完整的建議和用量詳細資訊，請參閱相關文件。用戶端應用程式可以使用 AWS Encryption SDK 來執行使用 AWS KMS 的信封加密。

```
// 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);
```

用戶端應用程式可以執行以下步驟：

1. 在 KMS 金鑰下提出新資料金鑰的請求。系統會傳回加密的資料金鑰和資料金鑰的純文字版本。
2. 在 AWS Encryption SDK 內，純文字資料金鑰會用來加密訊息。然後，純文字資料金鑰會從記憶體中刪除。
3. 加密的資料金鑰和加密的訊息會結合成單一加密文字位元組陣列。

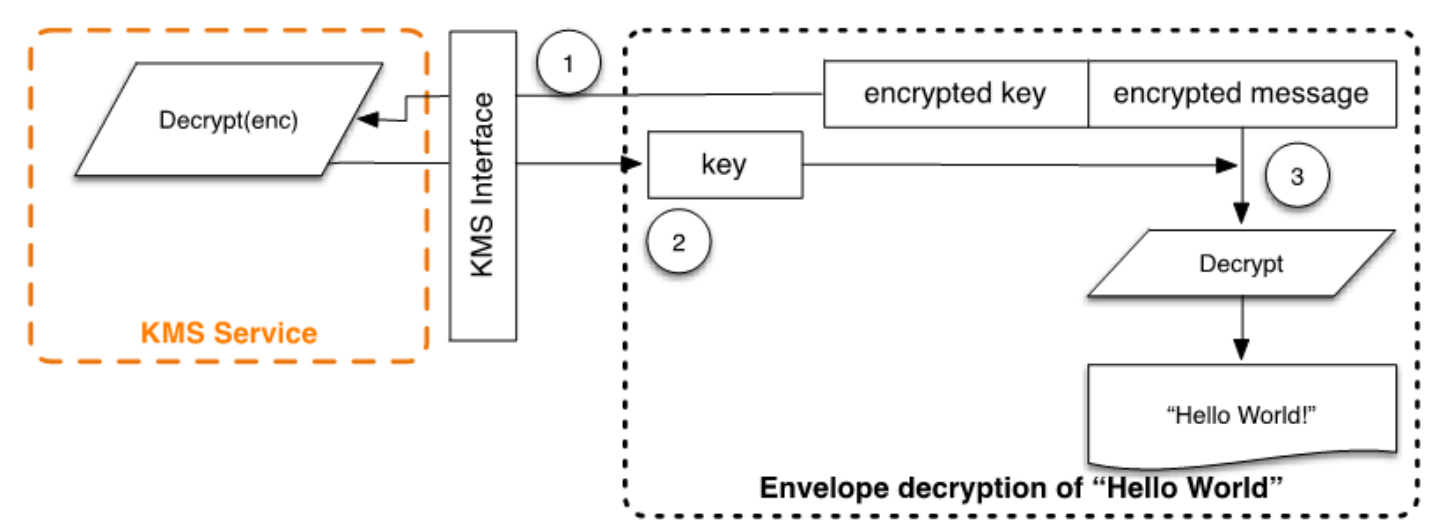


使用解密功能對信封加密的訊息進行解密，以取得原始加密的訊息。

```
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. AWS Encryption SDK 會剖析信封加密的訊息，以取得加密的資料金鑰，並向 AWS KMS 提出請求，以解密資料金鑰。
2. AWS Encryption SDK 會從 AWS KMS 接收純文字資料金鑰。
3. 然後，使用資料金鑰來解密訊息，傳回初始純文字。

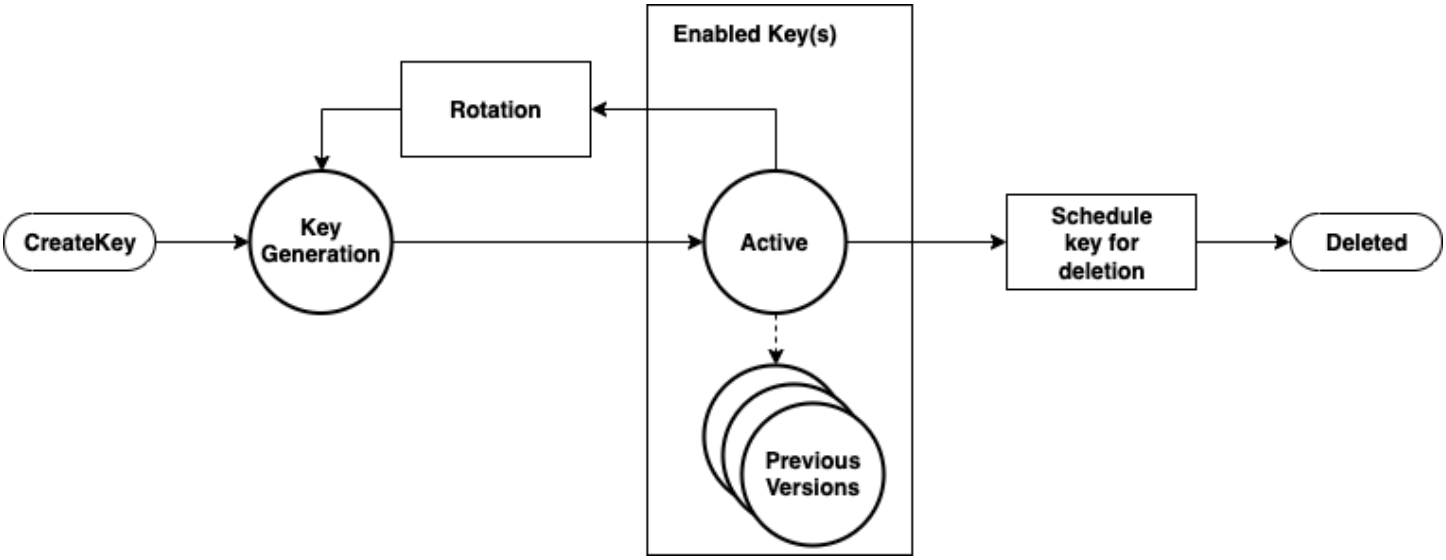


使用 AWS KMS keys

AWS KMS key 是指可能參照一或多個硬體安全模組 (HSM) 備份金鑰 (HBK) 的邏輯金鑰。本主題說明如何建立 KMS 金鑰、匯入金鑰材料，以及如何啟用、停用、輪換和刪除 KMS 金鑰。

> **Note**
>
> AWS KMS 正在將術語客戶主金鑰 (CMK) 取代為 AWS KMS key 和 KMS 金鑰。概念並沒有變更。為了防止重大變更，AWS KMS 會保留此術語的一些變化。

本章討論 KMS 金鑰從建立到刪除的生命週期，如下圖所示。



主題

- 呼叫 CreateKey
- 匯入金鑰材料
- 啟用和停用金鑰
- 刪除金鑰
- 輪換金鑰材料

呼叫 CreateKey

AWS KMS key 是因呼叫 `CreateKey` API 呼叫而產生的。

以下是 CreateKey 請求語法的子集。

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

請求接受採用 JSON 格式的下列資料。

描述

(選用) 金鑰的說明。建議您選擇描述以協助您決定金鑰是否適用於任務。

KeySpec

指定要建立的 KMS 金鑰類型。預設值 SYMMETRIC_DEFAULT 會建立對稱加密 KMS 金鑰。此參數對於對稱加密金鑰而言為選用，而對於所有其他金鑰規格而言則為必要。

KeyUsage

指定金鑰的使用。有效值為 ENCRYPT_DECRYPT、SIGN_VERIFY 或 GENERATE_VERIFY_MAC。預設值為 ENCRYPT_DECRYPT。此參數對於對稱加密金鑰而言為選用，而對於所有其他金鑰規格而言則為必要。

Origin

(選用) 指定 KMS 金鑰的金鑰素材來源。預設值為 AWS_KMS，這表示 AWS KMS 會產生與管理 KMS 金鑰的金鑰素材。其他有效值包含 EXTERNAL，代表不使用匯入金鑰素材的金鑰素材所建立的 KMS 金鑰，有效值也包含 AWS_CLOUDHSM，其會在受您控制 AWS CloudHSM 叢集所支援的自訂金鑰存放區中建立 KMS 金鑰。

政策

(選用) 要連接到金鑰的政策。如果省略政策，則會使用預設政策 (下列) 建立金鑰，以允許擁有 AWS KMS 許可的根帳戶和 IAM 主體對其進行管理。

如需有關政策的詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的 AWS KMS 中的金鑰政策和預設金鑰政策。

CreateKey 請求傳回包含金鑰 ARN 的回應。

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

如果 `Origin` 為 `AWS_KMS`，在建立 ARN 之後，會透過已驗證的工作階段提出對 AWS KMS HSM 的請求，以佈建硬體安全模組 (HSM) 備份金鑰 (HBK)。HBK 是 256 位元的金鑰，與 KMS 金鑰的此金鑰 ID 相關聯。它只能在 HSM 上產生，並且設計永遠不會以純文字格式匯出 HSM 邊界之外。HBK 在目前網域金鑰 DK_0 下加密。這些加密的 HBK 稱為加密金鑰字符 (EKT)。雖然 HSM 可以設定為使用各種金鑰包裝方法，但目前的實作會使用在已驗證加密配置 Galois 計數器模式 (GCM) 中的 AES-256。這種已驗證的加密模式允許我們保護部分純文字匯出金鑰字符中繼資料。

這在風格上表示為：

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

為您的 KMS 金鑰和後續 HBK 提供兩種基本保護形式：在 KMS 金鑰上設定的授權政策，以及關聯 HBK 上的密碼編譯保護。其餘的章節描述了 AWS KMS 中的密碼編譯保護和管理功能安全性。

除了 ARN 之外，您還可以透過建立金鑰的別名，建立讓使用者易記的名稱並將該名稱與 KMS 金鑰建立關聯。一旦別名已與 KMS 金鑰相關聯，就可以使用別名來在加密操作中識別 KMS 金鑰。如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的使用別名。

KMS 金鑰的使用範圍包括多個層級的授權。AWS KMS 會在加密內容與 KMS 金鑰之間啟用個別的授權政策。例如，AWS KMS 使用信封加密的 Amazon Simple Storage Service (Amazon S3) 物件會繼承 Amazon S3 儲存貯體上的政策。不過，對必要加密金鑰的存取權取決於 KMS 金鑰的存取政策。如需 KMS 金鑰授權的資訊，請參閱《AWS Key Management Service 開發人員指南》中的AWS KMS 的身分驗證與存取控制。

匯入金鑰材料

AWS KMS 提供匯入 HBK 所使用之密碼編譯材料的機制。如中所述呼叫 CreateKey，當 CreateKey 命令與`Origin`設定為搭配使用時`EXTERNAL`，會建立不包含基礎 HBK 的邏輯 KMS 金鑰。必須使用 `ImportKeyMaterial`API 呼叫匯入密碼編譯材料。您可以使用此功能來控制密碼編譯材料的金鑰建立和耐久性。如果您使用此功能，建議您在環境中對這些金鑰的處理和耐久性持謹慎態度。如需匯入金鑰材料的完整詳細資訊和建議，請參閱《AWS Key Management Service 開發人員指南》中的匯入金鑰材料。

呼叫 ImportKeyMaterial

`ImportKeyMaterial` 請求會匯入必要的 HBK 密碼編譯材料。密碼編譯材料必須是 256 位元對稱金鑰。它必須使用最近 `GetParametersForImport` 請求的傳回公有金鑰下 `WrappingAlgorithm` 中指定的演算法進行加密。

`ImportKeyMaterial` 請求會使用以下引數。

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

EncryptedKeyMaterial

使用公有金鑰加密的匯入金鑰材料會以使用該請求中指定包裝演算法的 `GetParametersForImport` 請求傳回。

ExpirationModel

指定金鑰材料是否過期。當此值為 `KEY_MATERIAL_EXPIRES` 時，`ValidTo` 參數必須包含過期日期。當此值為 `KEY_MATERIAL_DOES_NOT_EXPIRE` 時，請勿包含 `ValidTo` 參數。有效值為 `"KEY_MATERIAL_EXPIRES"` 和 `"KEY_MATERIAL_DOES_NOT_EXPIRE"`。

ImportToken

匯入字符會由提供公有金鑰的相同 `GetParametersForImport` 請求傳回。

KeyId

將與匯入金鑰材料相關聯的 KMS 金鑰。KMS 金鑰的 `Origin` 必須是 `EXTERNAL`。

您可以刪除和重新匯入相同的匯入金鑰素材至指定的 KMS 金鑰，但您無法匯入或關聯 KMS 金鑰任何其他金鑰素材。

ValidTo

(選用) 匯入的金鑰材料過期的時間。當金鑰材料過期時，AWS KMS 會刪除金鑰材料，讓 KMS 金鑰變成不可用。當 `ExpirationModel` 的值為 `KEY_MATERIAL_EXPIRES` 時，此參數為必要。否則為無效。

當請求成功時，KMS 金鑰可在 AWS KMS 中使用，直到指定的到期日期為止（如果提供到期日期）。匯入金鑰材料到期後，EKT 會從 AWS KMS 儲存層刪除。

啟用和停用金鑰

停用 KMS 金鑰可防止金鑰在密碼編譯操作中使用。此做法會暫停使用所有與 KMS 金鑰相關聯之 HBK 的功能。啟用還原 HBK 和 KMS 金鑰的使用。啟用和停用屬於簡單的請求，只需要使用金鑰 ID 或 KMS 金鑰的金鑰 ARN。

刪除金鑰

授權使用者可以使用 ScheduleKeyDeletion API 來安排刪除 KMS 金鑰和所有相關聯的 HBK。這是一項本質上具有破壞性的操作，從 AWS KMS 刪除金鑰時，應非常謹慎。刪除 KMS 金鑰時，AWS KMS 會強制執行最少七天的等待時間。在等待期間，金鑰會處於停用狀態，而金鑰狀態為 Pending Deletion (待刪除)。所有使用密鑰進行密碼編譯操作的調用都將失敗。 ScheduleKeyDeletion 採用下列引數。

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

KeyId

要刪除 KMS 金鑰的唯一識別符。若要指定該值，請使用唯一的金鑰 ID 或 KMS 金鑰的金鑰 ARN。

PendingWindowInDays

(選用) 等候期間，以天數指定。此值是選用的。範圍為 7 至 30 天，預設值為 30 天。等待期結束後，AWS KMS 會刪除 KMS 金鑰和所有相關聯的 HBK。

輪換金鑰材料

授權使用者可以啟用其客戶受管 KMS 金鑰的自動年度輪換。AWS 受管金鑰 一律每年輪換一次。

輪換 KMS 金鑰時，會建立新的 HBK，並標示為所有新加密請求的金鑰素材之目前版本。所有舊版的 HBK 仍然可以使用，以解密使用此 HBK 版本加密的任何密文。由於 AWS KMS 不會存放在 KMS 金鑰下加密的任何密文，因此在較舊的輪換 HBK 下加密的密文會要求 HBK 解密。您可以使用 ReEncrypt API 以在 KMS 金鑰的新 HBK 下或在不會公開純文字的其他 KMS 金鑰下重新加密任何密文。

如需啟用和停用金鑰輪換的資訊，請參閱《AWS Key Management Service 開發人員指南》中的輪換 AWS KMS key。

客戶資料操作

建立 KMS 金鑰之後，可以用於執行密碼編譯操作。每當使用 KMS 金鑰加密資料時，產生的物件就是客戶加密文字。加密文字包含兩個部分：未加密的標頭 (或純文字) 部分，由已驗證的加密配置作為其他已驗證資料進行保護，和一個加密的部分。純文字部分包括 HBK 識別符 (HBKID)。加密文字值的這兩個不可變欄位有助於確保 AWS KMS 在未來可以解密物件。

主題

產生資料金鑰

授權用戶可以使用 GenerateDataKey API（和相關 API）來請求特定類型的數據密鑰或任意長度的隨機密鑰。本主題提供此 API 操作的簡化檢視。如需詳細資訊，請參 GenerateDataKey 閱 AWS Key Management ServiceAPI 參考資料中的 API。

- GenerateDataKey
- GenerateDataKeyWithoutPlaintext
- GenerateDataKeyPair
- GenerateDataKeyPairWithoutPlaintext

以下是 `GenerateDataKey` 請求語法。

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

請求接受採用 JSON 格式的下列資料。

KeyId

用於加密資料金鑰之金鑰的金鑰識別符。此值必須能識別對稱加密 KMS 金鑰。

此為必要參數。

NumberOfBytes

包含要產生之位元組數量的整數。此為必要參數。

呼叫者必須提供 `KeySpec` 或 `NumberOfBytes`，但不能同時提供兩者。

EncryptionContext

(選用) Name-value (名称-值) 對，其中包含要在使用金鑰的加密和解密程序期間進行驗證的其他資料。

GrantTokens

(選用) 授予字符清單，代表授予提供產生或使用金鑰的許可。如需授予和授予字符的詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 AWS KMS 的身分驗證與存取控制。

在驗證命令之後，AWS KMS 會取得與 KMS 金鑰關聯的目前作用中的 EKT。它會將 EKT 連同您提供的請求和任何加密內容透過 AWS KMS 主機和網域中 HSM 之間的受保護工作階段傳遞給 HSM。

HSM 執行下列操作：

1. 產生請求的機密材料，並將其存放在揮發性記憶體中。
2. 解密符合請求中所定義 KMS 金鑰之金鑰 ID 的 EKT，以取得作用中 HBK = Decrypt(DK_j, EKT)。
3. 產生隨機 nonce N。
4. 從 HBK 和 N 產生 256 位元 AES-GCM 衍生的加密金鑰 K。
5. 加密機密材料 ciphertext = Encrypt(K, context, secret)。

`GenerateDataKey` 將純文字機密材料與密文透過 AWS KMS 主機和 HSM 之間的安全通道傳回給您。然後 AWS KMS 透過 TLS 工作階段將其傳送給您。AWS KMS 不會保留純文字或密文。如果沒有擁有加密文字、加密內容，以及使用 KMS 金鑰的授權，則無法傳回基礎機密。

以下是回應語法。

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

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

作為應用程式開發人員，資料金鑰的管理將交給您。如需用戶端加密搭配 AWS KMS 資料金鑰 (而不是資料金鑰對) 的最佳實務，您可以使用 AWS Encryption SDK。

資料金鑰可以依任何頻率輪換。此外，資料金鑰可以使用 `ReEncrypt` API 操作，根據不同的 KMS 金鑰或輪換的 KMS 金鑰重新加密資料金鑰。如需詳細資訊，請ReEncrypt參閱 AWS Key Management ServiceAPI 參考中的。

加密

AWS KMS 基本功能是在 KMS 金鑰下加密物件。根據設計，AWS KMS 在 HSM 上提供低延遲的密碼編譯操作。因此，在對函數進行加密的直接呼叫中，對可以加密的純文字量限制為 4 KB。AWS Encryption SDK 可用於加密較大的訊息。在驗證命令之後，AWS KMS 會取得與 KMS 金鑰相關的目前作用中 EKT。它會將 EKT 連同純文字和加密內容傳遞給區域中任何可用的 HSM。這些會透過 AWS KMS 主機和網域中 HSM 之間的已驗證工作階段傳送。

HSM 會執行下列項目：

1. 解密 EKT 以取得 HBK = Decrypt(DK_i, EKT)。
2. 產生隨機 nonce N。
3. 從 HBK 和 N 衍生 256 位元 AES-GCM 衍生的加密金鑰 K。
4. 加密純文字 ciphertext = Encrypt(K, context, plaintext)。

為您傳回加密文字值，並且不會在 AWS 基礎設施中的任何地方保留純文字資料或加密文字。如果沒有擁有加密文字和加密內容，以及使用 KMS 金鑰的授權，則無法傳回基礎純文字。

解密

解密加密文字值的 AWS KMS 呼叫會接受加密值加密文字和加密內容。AWS KMS 使用 AWS 簽章版本 4 已簽署的請求對呼叫進行身分驗證，並從加密文字中擷取包裝金鑰的 HBKID。HBKID 是用來取得解密加密文字、金鑰 ID 和金鑰 ID 政策所需的 EKT。請求是根據金鑰政策、可能存在的授予和參考金鑰 ID 的任何關聯 IAM 政策進行授權。`Decrypt` 函數類似於加密函數。

以下是 `Decrypt` 請求語法。

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

以下是請求參數。

CiphertextBlob

包括中繼資料的加密文字。

EncryptionContext

(選用) 加密內容。如果這是在 Encrypt 函數中指定的，則必須在此處指定，否則解密操作會失敗。如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的加密內容。

GrantTokens

(選用) 授予字符清單，代表授予提供執行解密的許可。

加密文字與 EKT 會連同加密內容，透過已驗證的工作階段傳送至 HSM 以進行解密。

HSM 會執行下列項目：

1. 解密 EKT，以取得 HBK = Decrypt(DK_i, EKT)。
2. 從加密文字結構擷取 nonce N。
3. 從 HBK 和 N 重新產生 256 位元 AES-GCM 衍生的加密金鑰 K。
4. 解密加密文字以取得 plaintext = Decrypt(K, context, ciphertext)。

產生的金鑰 ID 和純文字會透過安全的工作階段傳回 AWS KMS 主機，然後透過 TLS 連線回到呼叫客戶的應用程式。

以下是回應語法。

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

如果呼叫的應用程式想要確保純文字的真實性，則必須驗證傳回的金鑰 ID 是預期的。

重新加密已加密的物件

在一個 KMS 金鑰下加密的現有客戶加密文字可透過重新加密命令將其重新加密為另一個 KMS 金鑰。使用新的 KMS 金鑰在伺服器端重新加密資料，而不暴露在用戶端的金鑰純文字。資料會先解密，然後加密。

以下是請求語法。

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

請求接受採用 JSON 格式的下列資料。

CiphertextBlob

要重新加密的資料加密文字。

DestinationEncryptionContext

(選用) 重新加密資料時要使用的加密內容。

DestinationKeyId

用來重新加密資料之金鑰的金鑰 ID。

GrantTokens

(選用) 授予字符清單，代表授予提供執行解密的許可。

SourceKeyId

(選用) 用於解密資料之金鑰的金鑰識別符。

SourceEncryptionContext

(選用) 用於加密和解密 `CiphertextBlob` 參數中指定資料的加密內容。

該過程結合了先前描述的解密和加密操作：客戶加密文字在客戶加密文字所參考的初始 HBK 下解密至預期 KMS 金鑰下的當前 HBK。當此命令中使用的 KMS 金鑰相同時，此命令會將客戶加密文字從舊版的 HBK 移至最新版本的 HBK。

以下是回應語法。

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

如果呼叫應用程式想要確保基礎純文字的真實性，它必須驗證 SourceKeyId 傳回的是預期的。

AWS KMS 內部操作

針對全域分散式金鑰管理服務，需要 AWS KMS 內部元件才能擴展和保護 HSM。

主題

網域和網域狀態

AWS 區域 內可信任內部 AWS KMS 實體的合作集合稱為網域。網域包含一組可信任的實體、一組規則，以及一組稱為網域金鑰的機密金鑰。網域金鑰會在屬於網域成員的 HSM 之間共用。網域狀態由以下欄位組成。

名稱

用來識別此網域的網域名稱。

成員

屬於網域成員的 HSM 清單，包括其公有簽署金鑰和公有協定金鑰。

電信業者

實體清單、公有簽署金鑰，和角色 (AWS KMS 電信業者或服務主機），代表此服務的電信業者。

規則

必須滿足每個命令才能在 HSM 上執行命令的仲裁規則清單。

網域金鑰

目前在網域中使用的網域金鑰 (對稱金鑰) 清單。

完整網域狀態僅適用於 HSM。HSM 網域成員之間的網域狀態會同步為匯出的網域字符。

網域金鑰

網域中的所有 HSM 共用一組網域金鑰，$\{DK_r\}$。這些金鑰會透過網域狀態匯出常式共用。匯出的網域狀態可以匯入至屬於網域成員的任何 HSM。

一組網域金鑰 $\{DK_r\}$ 一律包含一個作用中的網域金鑰，以及數個已停用的網域金鑰。網域金鑰會每天輪換，以確保 AWS 符合金鑰管理的建議 - 第 1 部分。在網域金鑰輪換期間，所有在外寄網域金鑰下加密的現有 KMS 金鑰都會在新的作用中網域金鑰下重新加密。作用中的網域金鑰可用來加密任何新的 EKT。在等同於最近輪換網域金鑰次數的天數內，過期的網域金鑰只能用來解密先前加密的 EKT。

已匯出網域字符

需要定期同步處理網域參與者之間的狀態。這是透過每當變更網域時匯出網域狀態來完成的。網域狀態會匯出為匯出的網域字符。

名稱

用來識別此網域的網域名稱。

成員

屬於網域成員的 HSM 清單，包括其簽署和協定公有金鑰。

運算子

實體、公有簽署金鑰以及代表此服務電信業者的角色清單。

規則

必須滿足每個命令才能在 HSM 網域成員上執行命令的仲裁規則清單。

已加密的網域金鑰

信封加密的網域金鑰。網域金鑰會由上述每個成員的簽署成員進行加密，包裹在其公有協定金鑰中。

簽章

HSM 產生之網域狀態的簽章，必定是匯出網域狀態的網域成員。

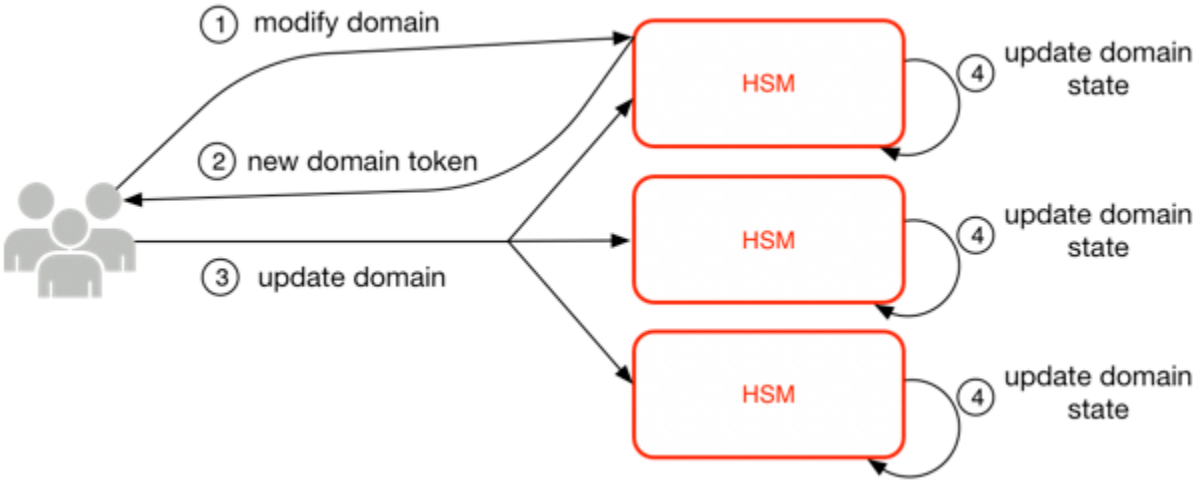
匯出的網域字符形成信任的基礎來源，用於網域內的實體操作。

管理網域狀態

網域狀態是透過仲裁驗證的命令進行管理。這些變更包括修改網域中可信任參與者的清單、修改執行 HSM 命令的仲裁規則，以及定期輪換網域金鑰。這些命令會以每個命令為基礎進行驗證，與已驗證工作階段操作完全不同 (如下圖所示)。

處於初始化和操作狀態的 HSM 包含一組自行產生的非對稱身分金鑰、簽署金鑰對，以及金鑰建立的金鑰對。透過手動程序，AWS KMS 電信業者可以建立要在區域中第一個 HSM 上建立的初始網域。此初始網域包含本主題先前定義的完整網域狀態。它是透過加入命令安裝至網域中每個已定義的 HSM 成員。

HSM 加入初始網域後，它會繫結至該網域中定義的規則。這些規則管控使用客戶密碼編譯金鑰或變更主機或網域狀態的命令。先前已定義之使用密碼編譯金鑰的已驗證工作階段 API 操作。



上述圖片描述了如何修改網域狀態。此程序包含四個步驟：

1. 以仲裁為基礎的命令會傳送至 HSM 以修改網域。
2. 系統會產生新的網域狀態，並匯出為新的匯出網域字符。系統不會修改 HSM 上的狀態，這表示該變更不會實際用於 HSM。
3. 第二個命令會傳送至新匯出的網域字符中的每個 HSM，以使用新的網域字符來更新其網域狀態。
4. 列在新匯出的網域字符中的 HSM 可以驗證命令和網域字符。其也可以解壓縮網域金鑰，以更新網域中所有 HSM 的網域狀態。

HSM 不會直接彼此通訊。相反地，電信業者的仲裁會請求變更網域狀態，從而產生新的匯出網域字符。網域的服務主機成員可用來將新網域狀態分配至網域中的每個 HSM。

離開和加入網域是透過 HSM 管理功能完成的。網域狀態的修改是透過網域管理功能完成的。

離開網域

讓 HSM 離開網域，從記憶體中刪除該網域的所有剩餘部分和金鑰。

加入網域

讓 HSM 加入新網域，或將其目前的網域狀態更新為新網域狀態。現有的網域會用作初始規則集的來源，以對此訊息進行驗證。

建立網域

在 HSM 上建立新網域。傳回可分配給網域成員 HSM 的第一個網域字符。

修改電信業者

在網域中新增或移除授權電信業者及其角色之清單中的電信業者。

修改成員

在網域中新增或移除已授權 HSM 清單中的 HSM。

修改規則

修改在 HSM 上執行命令所需的仲裁規則集。

輪換網域金鑰

建立新的網域金鑰，並將其標示為作用中的網域金鑰。這會將現有的作用中金鑰轉為已停用金鑰，並從網域狀態中移除最舊的已停用金鑰。

內部通訊安全

透過 已驗證的工作階段 中描述的兩種機制：仲裁簽署的請求方法和使用 HSM 服務主機通訊協定驗證的工作階段對服務主機或 AWS KMS 電信業者與 HSM 之間的命令進行保護。

仲裁簽署命令的設計是為了讓任何單一電信業者都無法修改 HSM 提供的重要安全保護。在已驗證工作階段上執行的命令有助於確保只有授權的服務電信業者可以執行涉及 KMS 金鑰的操作。所有客戶綁定的機密資訊都會在 AWS 基礎設施得到保護。

金鑰建立

為了保護內部通訊，AWS KMS 使用兩種不同的金鑰建立方法。第一種被定義為使用離散對數密碼編譯的成對金鑰建立配置的建議 (修訂版 2)中的 C(1, 2, ECC DH)。此配置擁有搭配靜態簽署金鑰的啟動

器。啟動器會產生並簽署暫時橢圓曲線 Diffie-Hellman (ECDH) 金鑰，用於具有靜態 ECDH 協定金鑰的收件人。此方法使用一個暫時金鑰和兩個的靜態金鑰 (使用 ECDH)。這是標籤 C(1, 2, ECC DH) 的衍生。此方法有時稱為單通 ECDH。

第二種金鑰建立方法是 C(2, 2, ECC, DH)。在這個配置中，雙方都有一個靜態簽署金鑰，其產生、簽署和交換暫時的 ECDH 金鑰。這個方法使用兩個靜態金鑰和兩個暫時金鑰，每個金鑰都使用 ECDH。這是標籤 C(2, 2, ECC, DH) 的衍生。這種方法有時被稱為 ECDH 暫時或 ECDHE。所有 ECDH 金鑰都會在曲線 secp384r1 (NIST-P384) 上產生。

HSM 安全邊界

AWS KMS 的內部安全邊界是 HSM。HSM 擁有專有界面，而且沒有其他處於操作狀態的作用中實體界面。在初始化期間，會使用必要的密碼編譯金鑰來佈建操作 HSM，以便在網域中建立其角色。HSM 的敏感密碼編譯材料只會存放在揮發性記憶體中，並在 HSM 移出操作狀態時清除，包括預期或非預期的關機或重設。

HSM API 操作會透過個別命令或透過服務主機建立的互相驗證機密工作階段進行驗證。

仲裁簽署的命令

仲裁簽署的命令是由電信業者核發給 HSM。本節描述了如何建立、簽署和驗證以仲裁為基礎的命令。這些規則相當簡單。例如，命令 Foo 需要角色 Bar 的兩個成員，以進行驗證。建立和驗證以仲裁為基礎的命令有三個步驟。第一步是初始命令建立；第二步是提交給其他電信業者進行簽署；第三步是驗證和執行。

為了引入這些概念，假設有一組真實的電信業者公有金鑰和角色 $\{QOS_s\}$，以及一組仲裁規則 QR = $\{Command_i, Rule_{\{i,\ t\}}\}$，其中每個規則是一組角色和最小數量 N $\{Role_t, N_t\}$。對於滿足仲裁規則的命令，命令資料集必須由列於 $\{QOS_s\}$ 中的一組電信業者簽署，使其符合為該命令列出的規則之一。如前所述，仲裁規則和電信業者集會以網域狀態和匯出的網域字符存放。

實際上，初始簽署者會簽署命令 $Sig_1 = Sign(dO_{p1}, Command)$。第二名電信業者也會簽署命令 $Sig_2 = Sign(dO_{p2}, Command)$。雙重簽署的訊息會傳送至 HSM 執行。HSM 將執行以下內容：

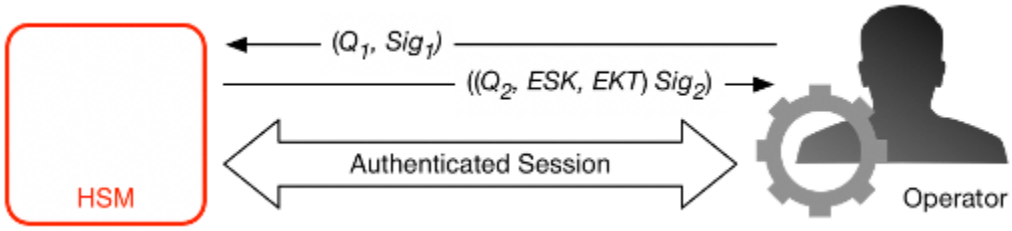
1. 對於每個簽章，它會從網域狀態擷取簽署者的公有金鑰，並驗證命令上的簽章。
2. 它會驗證該組簽署者是否滿足命令的規則。

已驗證的工作階段

您的金鑰操作會在面向外部的 AWS KMS 主機和 HSM 之間執行。這些命令與密碼編譯金鑰的建立和使用以及安全的隨機數字產生有關。命令會在服務主機和 HSM 之間的工作階段驗證通道上執行。除了需要真實性之外，這些工作階段還需要機密性。在這些工作階段上執行的命令包括傳回純文字資料金鑰，以及為您提供的解密訊息。為了確保這些工作階段無法透過 man-in-the-middle 攻擊顛覆，工作階段會經過驗證。

此通訊協定會在 HSM 與服務主機之間執行相互驗證的 ECDHE 金鑰協定。交換由服務主機啟動，並由 HSM 完成。HSM 也會傳回由交涉金鑰加密的工作階段金鑰 (SK)，以及包含工作階段金鑰的匯出金鑰字符。匯出的金鑰字符包含有效期間，之後服務主機必須重新交涉工作階段金鑰。

服務主機是網域的成員，且具有身分簽署金鑰對 ($dHOS_i$, $QHOS_i$) 和 HSM 身分公有金鑰的真實複本。它會使用其一組身分簽署金鑰來安全地交涉可在服務主機與網域中任何 HSM 之間使用的工作階段金鑰。匯出的金鑰字符具有與其相關聯的有效期間，之後必須交涉新的金鑰。



程序從服務主機辨識開始，它需要工作階段金鑰來傳送和接收本身與網域的 HSM 成員之間的敏感通訊流程。

1. 服務主機會產生 ECDH 暫時金鑰對 (d_1, Q_1) 並使用其身分金鑰 Sig1 = Sign(dOS,Q_1) 進行簽署。
2. HSM 會使用目前的網域字符來驗證已接收公有金鑰上的簽章，並建立 ECDH 暫時金鑰對 (d_2, Q_2)。然後，它會根據使用離散對數密碼編譯的成對金鑰建立配置的建議 (修訂版 2) 完成 ECDH-key-exchange，以形成交涉的 256 位元 AES-GCM 金鑰。HSM 會產生全新的 256 位元 AES-GCM 工作階段金鑰。它會使用交涉的金鑰加密工作階段金鑰，以形成加密的工作階段金鑰 (ESK)。它還會將網域金鑰下的工作階段金鑰加密為匯出的金鑰字符 EKT。最後，它會用其身分金鑰對 Sig_2 = Sign(dHSK, (Q_2, ESK, EKT)) 簽署傳回值。
3. 服務主機會使用其目前的網域字符來驗證已接收金鑰上的簽章。然後，服務主機會根據使用離散對數密碼編譯的成對金鑰建立配置的建議 (修訂版 2) 完成 EDCH 金鑰交換。接下來會解密 ESK 以取得工作階段金鑰 SK。

在 EKT 有效期間，服務主機可以使用交涉的工作階段金鑰 SK，將信封加密的命名傳送至 HSM。通過此身份驗證會話的每個 service-host-initiated 命令都包含 EKT。HSM 會使用相同交涉的工作階段金鑰 SK 進行回應。

多區域金鑰的複寫程序

AWS KMS 使用跨區域複寫機制，將 KMS 金鑰中的金鑰材料從一個 AWS 區域 中的 HSM 複製至不同 AWS 區域 中的 HSM。若要讓此機制能順利運作，正在複寫的 KMS 金鑰必須是多區域金鑰。將 KMS 金鑰從一個區域複寫到另一個區域時，區域中的 HSM 無法直接通訊，因為這些 HSM 均位於隔離的網路中。相反地，跨區域複寫期間交換的訊息是由 Proxy 服務負責傳遞。

在跨區域複寫期間，每則由 AWS KMS HSM 所產生的訊息均使用複寫簽署金鑰以密碼編譯方式進行簽署。複寫簽署金鑰 (RSK) 是 NIST P-384 曲線上的 ECDSA 金鑰。每個區域至少擁有一個 RSK，並且每個 RSK 的公有元件均與相同 AWS 分割區中的每個其他區域共用。

將金鑰材料從區域 A 複製到區域 B 的跨區域複寫處理的運作方式如下：

1. 區域 B 中的 HSM 會在 NIST P-384 曲線上產生一個短暫的 ECDH 金鑰，複寫協議金鑰 B (RAKB)。RAKB 的公有元件由代理服務傳送至區域 A 中的 HSM。
2. 區域 A 中的 HSM 接收 RAKB 的公有元件，然後在 NIST P-384 曲線上產生另一個暫時的 ECDH 金鑰，複寫協議金鑰 A (RAKA)。HSM 會在 RAKA 和 RAKB 的公有元件上執行 ECDH 金鑰建立配置，並從輸出衍生對稱金鑰複寫包裝金鑰 (RWK)。RWK 是用來加密正在複寫的多區域 KMS 金鑰的金鑰材料。
3. RAKA 的公有元件和使用 RWK 加密的金鑰材料會透過代理服務傳送至區域 B 的 HSM。
4. 區域 B 中的 HSM 會接收 RAKA 的公有元件，以及使用 RWK 加密的金鑰材料。HSM 由 RWK 在 RAKB 及 RAKA 的公有元件上執行 ECDH 金鑰建立配置而衍生。
5. 區域 B 中的 HSM 會使用 RWK 來解密區域 A 中的金鑰材料。

耐久性保護

服務所產生之金鑰的其他服務耐久性由離線 HSM 的使用、匯出網域字符的多個非揮發性儲存體以及加密 KMS 金鑰的備援儲存體提供。離線 HSM 是現有網域的成員。除了沒有上線和參與一般網域操作之外，離線 HSM 在網域狀態中顯示與現有 HSM 成員相同。

為防止 AWS 發生線上 HSM 或存放在主要儲存系統中之 KMS 金鑰集的大規模遺失，耐久性設計旨在保護區域中的所有 KMS 金鑰。具有匯入金鑰材料的 AWS KMS keys 不包含在其他 KMS 金鑰所提供

的耐久性保護之下。為防止 AWS KMS 的區域性故障，匯入的金鑰材料可能需要重新匯入至 KMS 金鑰。

離線 HSM 以及對其進行存取的憑證，都會存放在多個獨立地理位置上受監控之安全室內的保險箱內。每個保險箱至少需要有一名 AWS 資安管理人員及一名 AWS KMS 電信業者人員，來自兩個獨立的 AWS 團隊，以取得這些材料。這些資料的使用受內部政策規範，要求有一批 AWS KMS 電信業者。

參考資料

請使用下列參考材料來取得本文件中引用的縮寫、金鑰、參與者和來源相關資訊。

主題

縮寫

下列清單顯示本文件中參考的縮寫。

AES

進階加密標準

CDK

客戶資料金鑰

DK

網域金鑰

ECDH

橢圓曲線 Diffie-Hellman

ECDHE

橢圓曲線 Diffie-Hellman 暫時

ECDSA

橢圓曲線數位簽章演算法

EKT

匯出金鑰字符

ESK

加密工作階段金鑰

GCM

Galois 計數器模式

HBK

HSM 備份金鑰

HBKID

HSM 備份金鑰識別符

HSM

硬體安全模組

RSA

Rivest Shamir 和 Adleman (密碼邏輯)

secp384r1

高效密碼編譯主要 384 位元隨機曲線 1 的標準

SHA256

摘要長度 256 位元的安全雜湊演算法

鍵

下列清單定義了本文件中引用的金鑰。

HBK

HSM 備份金鑰：HSM 備份金鑰是 256 位元根金鑰，從中衍生特定使用金鑰。

DK

網域金鑰：網域金鑰是 256 位元 AES-GCM 金鑰。它會在網域的所有成員之間共用，並用來保護 HSM 備份金鑰材料和 HSM 服務主機工作階段金鑰。

DKEK

網域金鑰加密金鑰：網域金鑰加密金鑰是在主機上產生的 AES-256-GCM 金鑰，用於加密目前在 HSM 主機上同步網域狀態的一組網域金鑰。

(dHAK,QHAK)

HSM 協定金鑰對：每個起始的 HSM 在曲線 secp384r1 (NIST-P384) 上都有一個本機產生的橢圓曲線 Diffie-Hellman 協定金鑰對。

(dE, QE)

暫時協定金鑰對：HSM 和服務主機會產生暫時協定金鑰。這些是曲線 secp384r1 (NIST-P384) 上的橢圓曲線 Diffie-Hellman 金鑰。這些是在兩種使用案例中產生的：建立加 host-to-host 密金鑰以傳輸網域權杖中的網域金鑰加密金鑰，以及建立 HSM 服務主機工作階段金鑰以保護敏感通訊。

(dHSK,QHSK)

HSM 簽章金鑰對：每個起始的 HSM 在曲線 secp384r1 (NIST-P384) 上都有一個本機產生的橢圓曲線數位簽署金鑰對。

(dOS,QOS)

電信業者簽章金鑰對：服務主機電信業者和 AWS KMS 電信業者具有用來向其他網域參與者驗證自己的身分簽署金鑰。

K

資料加密金鑰：256 位元 AES-GCM 金鑰衍生自搭配 SHA256 使用 HMAC 之計數器模式 NIST SP800-108 KDF 的 HBK。

SK

工作階段金鑰：工作階段金鑰是由於服務主機電信業者和 HSM 之間交換的經驗證橢圓曲線 Diffie-Hellman 金鑰而建立的。交換的目的是保護服務主機與網域成員之間的通訊。

貢獻者

下列個人和組織為本文件作出了貢獻：

- 總經理 Ken Beer – KMS，AWS 密碼編譯
- 首席安全工程師 Matthew Campagna，AWS 密碼編譯

參考書目

如需 AWS Key Management Service HSM 的相關資訊，請前往 NIST 電腦安全資源中心密碼編譯模組驗證計劃搜尋頁面並搜尋 AWS Key Management Service HSM。

Amazon Web Services，一般參考 (版本 1.0)，「簽署 AWS API 請求」,http://docs.aws.amazon.com/general/latest/gr/signing_aws_api_requests.html。

Amazon Web Services，「什麼是 AWS Encryption SDK」，http://docs.aws.amazon.com/encryption-sdk/latest/developer-guide/introduction.html。

美國聯邦資訊處理標準出版物，FIPS PUB 180-4。安全雜湊標準，2012 年 8 月。可從 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf 取得。

美國聯邦資訊處理標準出版物 197，宣布進階加密標準 (AES)，2001 年 11 月。可從 http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf 取得。

美國聯邦資訊處理標準出版物 198-1,金鑰雜湊訊息身分驗證程式碼 (HMAC)，2008 年 7 月。可從 http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf 取得。

NIST 特別刊物 800-52 修訂版 2，《傳輸層安全性 (TLS) 實作的選擇、組態和使用準則》，2019 年 8 月。 https://nvlpubs.nist.gov/nistpubs/ SpecialPublications /奈特·斯普.

PKCS#1 v2.2：RSA 密碼編譯標準 (RFC 8017)，網際網路工程任務小組 (IETF)，2016 年 11 月。https://tools.ietf.org/html/rfc8017。

關於區塊密碼操作模式的建議：Galois 计數器模式 (GCM) 和 GMAC、NIST 特別出版物 800-38D，2007 年 11 月。可從 http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf 取得。

關於區塊密碼操作模式的建议：針對儲存裝置機密性的 XTS-AES 模式，NIST 特別出版物 800-38E，2010 年 1 月。可從 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38e.pdf 取得。

關於使用虛擬隨機函數之金鑰衍生的建議，NIST 特別出版物 800-108，2009 年 10 月，可從 https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-108.pdf 取得。

關於金鑰管理的建議 – 第一部分：基本資訊 (修訂版 5)，NIST 特別出版物 800-57A，2020 年 5 月，可從 https://doi.org/10.6028/NIST.SP.800-57pt1r5 取得。

使用離散對數密碼編譯的成對金鑰建立配置的建議 (修訂)，NIST 特別出版物 800-56A 修訂版 3，2018 年 4 月。可從以下網站SpecialPublications取得。https://nvlpubs.nist.gov/nistpubs/

針對使用確定性隨機位元產生器產生隨機數的建議，NIST 特別刊物 800-90A 修訂版本 1，2015 年 6 月，可從網站取得。SpecialPublications https://nvlpubs.nist.gov/nistpubs/

第 2 章：建議的橢圓曲線網域參數，高效密碼編譯群組的標準，版本 2.0，2010 年 1 月 27 日。

使用密碼編譯訊息語法 (CMS) 中的橢圓曲線密碼編譯 (ECC) 演算法，Brown，D., Turner，S.，網際網路工程任務小組，2010 年 7 月，http://tools.ietf.org/html/rfc5753/。

X9.62-2005：金融服務業的公有金鑰加密：橢圓曲線數位簽章演算法 (ECDSA)，美國國家標準協會，2005 年。

AWS KMS 密碼編譯詳細資訊的文件歷史記錄

下表說明了對「AWS Key Management Service 密碼編譯詳細資訊」文件所進行的重要變更。我們也會經常更新文件，以處理您傳送給我們的意見回饋。

變更	描述	日期
已更新內容	已新增有關 AWS KMS `ReplicateKey` 操作實作的詳細資訊。	2021 年 10 月 28 日
文件變更	使用 AWS KMS key 和 KMS 金鑰 取代術語客戶主要金鑰 (CMK)。	2021 年 8 月 30 日
初始版本	從「KMS 密碼編譯詳細資訊」技術論文建立本指南	2020 年 12 月 30 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。