AWS KMS Détails cryptographiques

AWS Key Management Service

AWS Key Management Service: AWS KMS Détails cryptographiques

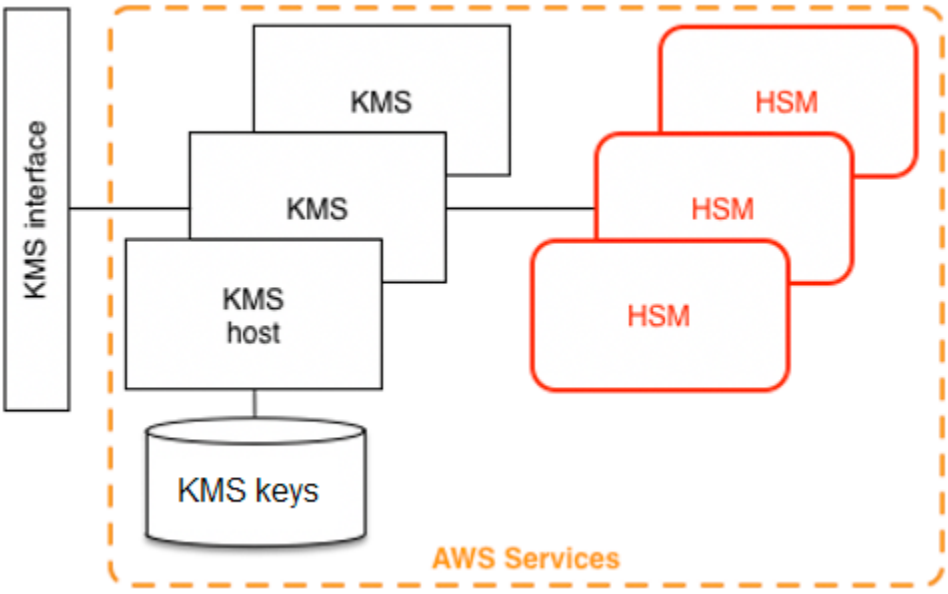
Table of Contents

Présentation des détails cryptographiques de AWS KMS

AWS Key Management Service (AWS KMS) fournit une interface Web aux fins de générer et gérer des clés cryptographiques et fonctionne comme un fournisseur de services cryptographiques pour la protection des données. AWS KMS offre des services traditionnels de gestion des clés intégrés aux AWS services afin de fournir une vue cohérente des clés des clients dans AWS, avec une gestion et un audit centralisés. Ce livre blanc fournit une description détaillée des opérations cryptographiques AWS KMS pour vous aider à évaluer les fonctions offertes par le service.

AWS KMS inclut une interface Web via le AWS Management Console, l'interface de ligne de commande et les opérations d'API RESTful pour demander des opérations cryptographiques d'une flotte distribuée de modules de sécurité matérielle (HSM) validés FIPS 140-2 [1]. Le AWS KMS HSM est un dispositif cryptographique matériel autonome multipuces conçu pour fournir des fonctions cryptographiques dédiées afin de répondre aux exigences de sécurité et d'évolutivité de AWS KMS. Vous pouvez établir votre propre hiérarchie cryptographique basée sur le HSM sous les clés que vous gérez en tant que AWS KMS keys. Ces clés sont disponibles uniquement sur les HSM et uniquement en mémoire pendant le temps nécessaire au traitement de votre demande cryptographique. Vous pouvez créer plusieurs clés KMS, chacune représentée par son ID de clé. Uniquement sous AWS les rôles et comptes IAM administrés par chaque client peuvent créer, supprimer ou utiliser des clés KMS client pour chiffrer, déchiffrer, signer ou vérifier des données. Vous pouvez définir des contrôles d'accès sur les personnes qui peuvent gérer et/ou utiliser les clés KMS en créant une politique associée à la clé. Ces politiques vous permettent de définir des utilisations propres à l'application de vos clés pour chaque opération d'API.

En outre, la plupart des AWS services prennent en charge le chiffrement des données au repos à l'aide des clés KMS. Cette possibilité permet aux clients de contrôler quand et comment AWS les services peuvent accéder aux données chiffrées en contrôlant quand et comment les clés KMS sont accessibles.



AWS KMS est un service à plusieurs niveaux, composé AWS KMS d'hôtes orientés vers le Web et d'un niveau de HSM. Le regroupement de ces hôtes à plusieurs niveaux forme la AWS KMS pile. Toutes les demandes adressées à AWS KMS doivent être effectuées via le protocole TLS (Transport Layer Security) et se terminer sur un AWS KMS hôte. AWS KMS hôtes autorisent uniquement TLS avec une suite de chiffrement qui fournit un secret de transmission parfait. AWS KMS authentifie et autorise vos demandes à l'aide des mêmes mécanismes d'identification et de politique AWS Identity and Access Management (IAM) qui sont disponibles pour toutes les autres AWS opérations d'API

Concepts de base

L'apprentissage de quelques termes et concepts de base vous aidera à tirer le meilleur parti de l'utilisation de AWS Key Management Service.

AWS KMS key

> **Note**
>
> AWS KMS remplace le terme clé principale client (CMK) par AWS KMS key et clé KMS. Le concept n'a pas changé. Pour éviter les changements de rupture, AWS KMS conserve quelques variations de ce terme.

Une clé logique qui représente le haut de votre hiérarchie de clés. Une clé KMS reçoit un ARN (Amazon Resource Name) qui inclut un identificateur de clé unique ou un ID de clé. AWS KMS keys ont trois types :

- Clé gérée par le client – Les clients créent et contrôlent le cycle de vie et les politiques de clé des clés gérées par le client. Toutes les demandes effectuées à l'aide de ces clés sont enregistrées en tant qu' CloudTrail événements.
- Clés gérées par AWS – AWScrée et contrôle le cycle de vie et les politiques de clé de Clés gérées par AWS, qui sont des ressources dans le Compte AWS d'un client. Les clients peuvent consulter les politiques d'accès et les CloudTrail événements relatifs à ces clésClés gérées par AWS, mais ne peuvent gérer aucun aspect de ces clés. Toutes les demandes effectuées à l'aide de ces clés sont enregistrées en tant qu' CloudTrail événements.
- Clés détenues par AWS – Ces clés sont créées et utilisées exclusivement par AWS pour des opérations de chiffrement interne dans différents AWS services. Les clients n'ont aucune visibilité sur les principales politiques ou sur Clé détenue par AWS l'utilisation de CloudTrail.

Alias

Nom convivial associé à une clé KMS. L'alias peut être utilisé de façon interchangeable avec l'ID de clé dans la plupart des AWS KMS opérations d'API

Autorisations

Politique associée à une clé KMS qui définit les autorisations sur la clé. La politique par défaut autorise toutes les entités que vous définissez, et autorise également Compte AWS à ajouter des politiques IAM qui font référence à la clé.

Octrois

L'autorisation déléguée d'utiliser une clé KMS lorsque les principales IAM prévues ou la durée d'utilisation est inconnue au début et ne peut donc pas être ajoutée à une clé ou à une politique IAM. L'une des utilisations des octrois consiste à définir les autorisations descendantes pour la façon dont un AWS service peut utiliser une clé KMS. Le service peut avoir besoin d'utiliser votre clé pour effectuer un travail asynchrone en votre nom sur des données chiffrées en l'absence d'un appel d'API signé directement de votre part.

Clés de données

Clés cryptographiques générées sur les clés HSM, protégées par une clé KMS. AWS KMS permet aux entités autorisées d'obtenir des clés de données protégées par une clé KMS. Elles peuvent être retournées à la fois sous forme de clés de données en texte brut (non chiffrées) et sous forme

de clés de données chiffrées. Les clés de données peuvent être symétriques ou asymétriques (avec les parties publiques et privées renvoyées).

Textes chiffrés

La sortie cryptée deAWS KMS, parfois appelé « texte chiffré du client » pour éviter toute confusion. Le texte chiffré contient des données chiffrées avec des informations supplémentaires qui identifient la clé KMS à utiliser dans le processus de déchiffrement. Les clés de données chiffrées sont un exemple courant de texte chiffré produit lors de l'utilisation d'une clé KMS, mais toutes les données de moins de 4 Ko peuvent être chiffrées sous une clé KMS pour générer un texte chiffré.

Contexte de chiffrement

Une carte de paires clé-valeur d'informations supplémentaires associées aux AWS KMS – informations protégées. AWS KMS utilise un chiffrement authentifié pour protéger les clés de données. Le contexte de chiffrement est incorporé dans l'AAD du chiffrement authentifié dans AWS KMS – textes chiffrés. Ces informations de contexte sont facultatives et ne sont pas renvoyées lors de la demande d'une clé (ou d'une opération de chiffrement). Mais si elles sont utilisées, cette valeur de contexte est nécessaire pour terminer avec succès une opération de déchiffrement. Une utilisation prévue du contexte de chiffrement est de fournir des informations authentifiées supplémentaires. Ces informations peuvent vous aider à appliquer les politiques et à être incluses dans les AWS CloudTrail journaux. Par exemple, vous pouvez utiliser une paire clé-valeur de {"key name":"satellite uplink key"} pour nommer la clé de données. L'utilisation ultérieure de la clé créera une AWS CloudTrail entrée qui comportera “key name”: “satellite uplink key.” Ces informations supplémentaires peuvent fournir un contexte utile pour comprendre pourquoi une clé KMS donnée a été utilisée.

Clé publique

Lors de l'utilisation de chiffrements asymétriques (RSA ou courbe elliptique), la clé publique est la « composante publique » d'une paire de clés publique-privée. La clé publique peut être partagée et distribuée aux entités qui ont besoin de chiffrer les données pour le propriétaire de la paire de clés publique-privée. Pour les opérations de signature numérique, la clé publique est utilisée pour vérifier la signature.

Clé privée

Lors de l'utilisation de chiffrements asymétriques (RSA ou courbe elliptique), la clé privée est la « composante privée » d'une paire de clés publique-privée. La clé privée est utilisée pour déchiffrer des données ou créer des signatures numériques. Similaire aux clés KMS symétriques,

les clés privées sont chiffrées dans des clés HSM. Elles sont uniquement déchiffrées dans la mémoire volatile de la clé HSM et seulement pour le temps nécessaire au traitement de votre demande cryptographique.

AWS KMS objectifs de conception

AWS KMS est conçu pour répondre aux critères suivants :

Durabilité

La durabilité des clés cryptographiques est conçue pour être égale à celle des services de durabilité les plus élevés dans AWS. Une seule clé cryptographique peut chiffrer de grands volumes de vos données qui se sont accumulés sur une longue période.

Digne de confiance

L'utilisation des clés est protégée par des politiques de contrôle d'accès que vous définissez et gérez. Il n'existe aucun mécanisme permettant d'exporter des clés KMS en texte brut. La confidentialité de vos clés cryptographiques est primordiale. Plusieurs employés d'Amazon disposant d'un accès spécifique aux contrôles d'accès basés sur le quorum doivent effectuer des actions administratives sur les HSM.

Faible latence et haut débit

AWS KMS fournit des opérations cryptographiques à des niveaux de latence et de débit adaptés à une utilisation par d'autres services dans AWS.

Régions indépendantes

AWS fournit des régions indépendantes pour les clients qui ont besoin de restreindre l'accès aux données dans différentes régions. L'utilisation de la clé peut être isolée dans un Région AWS.

Source sécurisée de nombres aléatoires

Étant donné que la cryptographie forte dépend de la génération de nombres aléatoires vraiment imprévisibles, AWS KMS fournit une source validée de nombres aléatoires de haute qualité.

Audit

AWS KMSenregistre l'utilisation et la gestion des clés cryptographiques dans des AWS CloudTrail journaux. Vous pouvez utiliser AWS CloudTrail les journaux pour inspecter l'utilisation de vos clés cryptographiques, notamment l'utilisation des clés par AWS les services en votre nom.

Pour atteindre ces objectifs, le AWS KMS système inclut un ensemble AWS KMS d'opérateurs et d'opérateurs d'hôtes de services (collectivement, les « opérateurs ») qui administrent des « domaines ». Un domaine est un ensemble de AWS KMS serveurs, HSM et opérateurs définis à l'échelle de la région. Chaque AWS KMS opérateur dispose d'un jeton matériel qui contient une paire de clés privées et publiques qui est utilisée pour authentifier ses actions. Les HSM disposent d'une paire de clés privées et publiques supplémentaires qu'ils utilisent pour établir des clés de chiffrement qui protègent la synchronisation d'état HSM.

Ce document illustre comment AWS KMS protège vos clés et autres données que vous souhaitez chiffrer. Tout au long de ce document, les clés de chiffrement ou les données que vous souhaitez chiffrer sont appelées « secrets » ou « éléments secrets ».

Principes de base de AWS Key Management Service

Les sujets de ce chapitre décrivent les primitives cryptographiques de AWS Key Management Service et où elles sont utilisées. Ils présentent également les éléments de base de AWS KMS.

Rubriques

Primitives cryptographiques

AWS KMS utilise des algorithmes cryptographiques configurables afin que le système puisse migrer rapidement d'un algorithme ou mode approuvé à un autre. L'ensemble initial d'algorithmes cryptographiques par défaut a été sélectionné dans les algorithmes Federal Information Processing Standard (FIPS Approved) pour leurs propriétés de sécurité et leurs performances.

Entropie et génération de nombres aléatoires

AWS KMS la génération de clés est effectuée sur les AWS KMS clés HSM. Les clés HSM mettent en œuvre un générateur de nombres aléatoires hybrides qui utilise le NIST SP800-90A Générateur de bits aléatoires déterministes (DRBG) CTR_DRBG utilisant l'algorithme de chiffrement AES-256. Il est alimenté par un générateur de bits aléatoires non déterministe avec 384 bits d'entropie et mis à jour avec de l'entropie supplémentaire aux fins de fournir une résistance à la prédiction à chaque appel d'élément cryptographique.

Opérations de clé symétrique (chiffrement uniquement)

Toutes les commandes de chiffrement par clé symétrique utilisées dans les clés HSM utilisent les Normes de chiffrement avancées (AES), dans Galois Counter Mode (GCM) à l'aide de clés de 256 bits. Les appels analogues pour déchiffrer utilisent la fonction inverse.

AES-GCM est un schéma de chiffrement authentifié. En plus de chiffrer le texte brut afin de produire du texte chiffré, il calcule une balise d'authentification sur le texte chiffré et toutes les données supplémentaires pour lesquelles une authentification est requise (données authentifiées supplémentaires, ou AAD). La balise d'authentification permet de s'assurer que les données proviennent de la source présumée et que le texte chiffré et l'AAD n'ont pas été modifiés.

Fréquemment, AWS omet l'inclusion de l'AAD dans nos descriptions, plus particulièrement en ce qui concerne le chiffrement des clés de données. Dans ces cas, le texte environnant laisse entendre que la structure à chiffrer est divisée entre le texte brut à chiffrer et l'AAD en texte clair à protéger

AWS KMS vous fourni une option pour importer des éléments de clé dans une AWS KMS key au lieu de vous fier à AWS KMS pour générer des éléments de clé. Cet élément de clé importé peut être chiffré à l'aide de RSAES-OAEP ou de RSAES-PKCS1-v1_5 pour protéger la clé pendant le transport vers le AWS KMS HSM. Les paires de clés RSA sont générées sur des clés AWS KMS HSM. L'élément de la clé importé est déchiffré sur une clé AWS KMS HSM, puis rechiffrée sous AES-GCM avant d'être stocké par le service.

Opérations de clés asymétriques (chiffrement, signature numérique et vérification de signature)

AWS KMS prend en charge l'utilisation d'opérations de clé asymétrique pour les opérations de chiffrement et de signature numérique. Les opérations de clé asymétrique reposent sur une paire de clés publique et privée mathématiquement liées que vous pouvez utiliser pour le chiffrement et le déchiffrement ou pour la signature et la vérification de la signature, mais pas les deux. La clé privée ne quitte jamais AWS KMS non chiffrée. Vous pouvez utiliser la clé publique dans AWS KMS en appelant les opérations d'API AWS KMS ou télécharger la clé publique et l'utiliser hors de AWS KMS.

AWS KMS prend en charge deux types de chiffrements asymétriques.

- RSA-OAEP (pour le chiffrement) et RSA-PSS et RSA-PKCS- #1 -v1_5 (pour la signature et la vérification) – Prend en charge les longueurs de clés RSA (en bits) : 2 048, 3 072 et 4 096 pour différentes exigences de sécurité.
- Courbe elliptique (ECC) – Utilisée exclusivement pour la signature et la vérification. Prend en charge les courbes ECC : NIST P256, P384, P521, SECP 256k1.

Fonctions de dérivation de clé

Une fonction de dérivation de clé est utilisée pour dériver des clés supplémentaires à partir d'un secret initial ou d'une clé. AWS KMS utilise une fonction de dérivation de clé (KDF) pour dériver des clés par appel pour chaque chiffrement sous un AWS KMS key. Toutes les opérations KDF utilisent la fonction KDF en mode compteur utilisant HMAC[FIPS197] avec SHA256 [FIPS180]. La clé dérivée de 256 bits est utilisée avec AES-GCM aux fins de chiffrer ou de déchiffrer les données et les clés des clients.

AWS KMS utilisation interne des signatures numériques

Les signatures numériques sont également utilisées pour authentifier des commandes et des communications entre AWS KMS entités. Toutes les entités de service disposent d'une paire de clés de l'algorithme de signature numérique à courbe elliptique (ECDSA). Elles exécutent l'ECDSA comme décrit dans Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS) and X9.62-2005: Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA). Les entités utilisent l'algorithme de hachage sécurisé défini dans Federal Information Processing Standards Publications, FIPS PUB 180-4, connu sous le nom de « SHA384 ». Les clés sont générées sur la courbe secp384r1 (NIST-P384).

Chiffrement d'enveloppe

Une construction de base utilisée dans de nombreux systèmes cryptographiques est le chiffrement d'enveloppe. Le chiffrement d'enveloppe utilise deux clés cryptographiques ou plus afin de sécuriser un message. Généralement, une clé est dérivée d'une clé statique à plus long terme k, et une autre clé est une clé par message, msgKey, qui est générée pour chiffrer le message. L'enveloppe est constituée en chiffrant le message :texte chiffré = Encrypt(msgKey, message). Ensuite, la clé de message est chiffrée à l'aide de la clé statique à long terme : encKey = Encrypt(k, msgKey). Enfin, les deux valeurs (encKey, texte chiffré) sont empaquetés dans une structure unique, ou dans un message chiffré par enveloppe.

Le destinataire, avec accès à k, peut ouvrir le message enveloppé en déchiffrant d'abord la clé chiffrée, puis en déchiffrant le message.

AWS KMS permet de gérer ces clés statiques à plus long terme et d'automatiser le processus de chiffrement d'enveloppe de vos données.

En plus des fonctionnalités de chiffrement fournies dans le AWS KMS service, le AWS kit SDK de chiffrement fournit des bibliothèques de chiffrement d'enveloppe côté client. Vous pouvez utiliser ces bibliothèques pour protéger vos données et les clés de chiffrement utilisées pour chiffrer ces données.

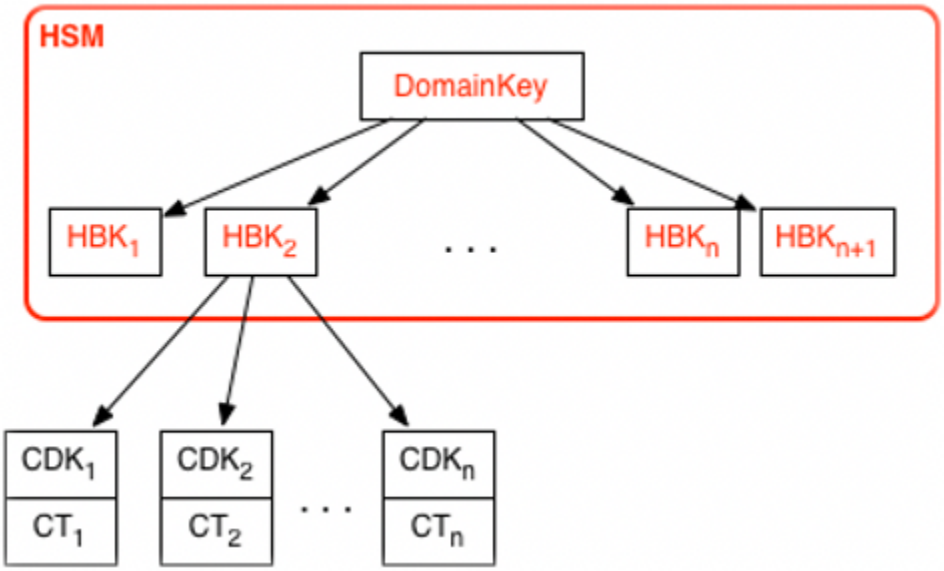
AWS KMS key HIÉRARCHIE

Votre hiérarchie de clés commence par une clé logique de niveau supérieur, une AWS KMS key. Une clé KMS représente un conteneur pour le matériel de clé de niveau supérieur et est définie de manière unique dans l'espace de noms service AWS avec un ARN (Amazon Resource Name). L'ARN comprend un identifiant de clé généré de manière unique, un ID de la clé. Une clé KMS

est créée à la suite d'une demande initiée par l'utilisateur via AWS KMS. À réception, AWS KMS demande la création d'une clé de sauvegarde HSM (HBK) initiale à placer dans le conteneur de clés KMS. La clé HBK est générée sur un HSM du domaine et conçue pour ne jamais être exporté depuis le HSM en texte brut. Au lieu de cela, la clé HBK est exportée chiffrée dans des clés de domaine gérées par HSM. Ces clés HBK exportées sont appelés « jetons de clés exportés » (EKT).

L'EKT est exporté vers un stockage hautement durable et à faible latence. Par exemple, supposons que vous receviez un ARN sur la clé KMS logique. Cela représente le haut d'une hiérarchie de clés, ou contexte cryptographique, pour vous. Vous pouvez créer plusieurs clés KMS dans votre compte et définir des politiques relatives à vos clés KMS, comme pour toute autre AWS ressource nommée.

Dans la hiérarchie d'une clé KMS spécifique, la clé HBK peut être considérée comme une version de la clé KMS. Lorsque vous souhaitez faire pivoter la clé KMS via AWS KMS, une nouvelle clé HBK est créée et associée à la clé KMS en tant que clé HBK active pour la clé KMS. Les anciennes clés HBK sont conservées et peuvent être utilisées pour déchiffrer et vérifier des données précédemment protégées. Mais seule la clé cryptographique active peut être utilisée pour protéger de nouvelles informations.



Vous pouvez faire des demandes via AWS KMS pour utiliser vos clés KMS aux fins de protéger directement les informations ou demander des clés supplémentaires générées par la clé HSM qui sont protégées sous votre clé KMS. Ces clés sont appelées « clés de données client », ou CDK. Les clés CDK peuvent être retournées chiffrées sous forme de texte chiffré (CT), en texte brut, ou les deux. Tous les objets chiffrés sous une clé KMS (que ce soient des données fournies par le client ou des clés générées par la clé HSM) peuvent uniquement être déchiffrés sur une clé HSM via un appel par AWS KMS.

Le texte chiffré retourné, ou la charge utile déchiffrée, n'est jamais stocké dans AWS KMS. Les informations vous sont retournées via votre connexion TLS à AWS KMS. Cela s'applique également aux appels effectués par AWS services en votre nom.

La hiérarchie des clés et les propriétés de ces clés spécifiques s’affichent dans le tableau suivant.

Clé	Description	Cycle de vie
Clé de domaine	Une clé AES-GCM 256 bits uniquement dans la mémoire d'une clé HSM utilisée pour envelopper les versions des clés KMS, les clés de sauvegarde HSM.	Rotation tous les jours[1]
Clé de sauvegarde HSM	Une clé symétrique 256 bits ou une clé privée RSA ou courbe elliptique, utilisée pour protéger les données et les clés du client stockées et chiffrées sous les clés de domaine. Une ou plusieurs clés de sauvegarde HSM comprennent la clé KMS, représentée par l'ID KeyID.	Rotation tous les ans[2] (configuration facultative)
Clé de chiffrement dérivée	Une clé AES-GCM 256 bits résidant uniquement dans la mémoire d'une clé HSM est utilisée pour chiffrer les données et les clés du client. Dérivée d'une clé HBK pour chaque chiffrement.	Utilisée une seule fois par chiffrement et régénérée au déchiffrement
Clé de données client	Clé symétrique ou asymétrique définie par l'utilisateur, exportée depuis une clé HSM en texte brut et en texte chiffré. Chiffrée sous une clé de sauvegarde HSM et renvoyée aux utilisateurs autorisés via le canal TLS.	Rotation et utilisation contrôlée par application

[1] AWS KMSpeut de temps à autre alléger la rotation des clés de domaine à une fréquence au plus hebdomadaire afin de tenir compte des tâches d'administration et de configuration du domaine.

[2] Par défaut, les Clés gérées par AWS créées et gérées par AWS KMS en votre nom sont automatiquement soumises à une rotation annuelle.

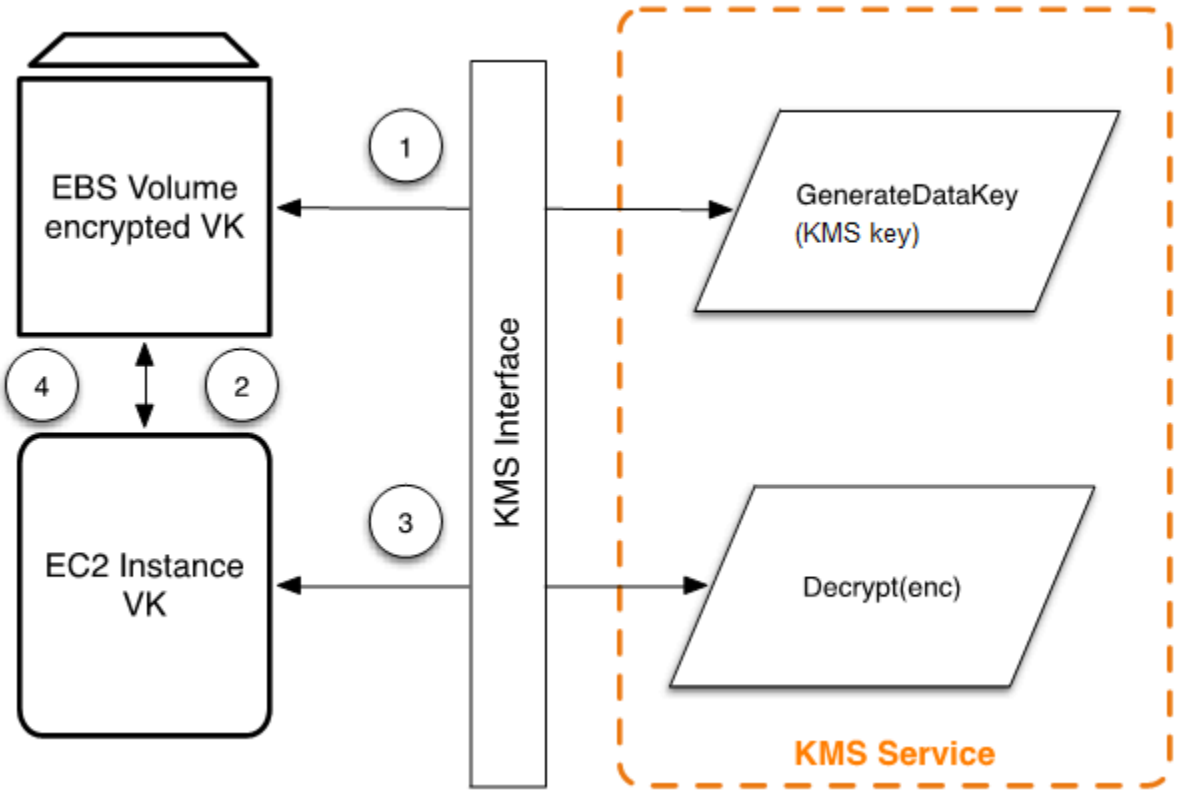
AWS KMS cas d'utilisation

Les cas d'utilisation peuvent vous aider à en tirer le meilleur parti AWS Key Management Service. La première montre comment s' AWS KMS effectue le chiffrement côté serveur AWS KMS keys sur un volume Amazon Elastic Block Store (Amazon EBS). La seconde est une application côté client qui montre comment utiliser le chiffrement d'enveloppe pour protéger le contenu avec. AWS KMS

Rubriques

Chiffrement de volume Amazon EBS

Amazon EBS offre une possibilité de chiffrement de volume. Chaque volume est chiffré à l'aide d'AES-256-XTS. Cela nécessite deux clés de volume de 256 bits, que vous pouvez considérer comme une seule clé de volume de 512 bits. La clé de volume est chiffrée sous une clé KMS de votre compte. Afin qu'Amazon EBS puisse chiffrer un volume pour vous, il doit disposer d'un accès pour générer une clé de volume (VK) sous une clé KMS dans le compte. Pour ce faire, vous autorisez à Amazon EBS un droit d'accès à la clé KMS aux fins de créer des clés de données, ainsi que pour chiffrer et déchiffrer ces clés de volume. Amazon EBS utilise AWS KMS désormais une clé KMS pour générer des clés de volume AWS KMS chiffrées.



Le flux de travail suivant chiffre les données en cours d'écriture sur un volume Amazon EBS :

1. Amazon EBS obtient une clé de volume chiffrée sous une clé KMS AWS KMS via une session TLS et stocke la clé chiffrée avec les métadonnées du volume.
2. Lorsque le volume Amazon EBS est monté, la clé de volume chiffrée est récupérée.
3. Un appel AWS KMS via TLS est effectué pour déchiffrer la clé de volume chiffrée. AWS KMS identifie la clé KMS et envoie une demande interne à un HSM de la flotte pour déchiffrer la clé de volume chiffrée. AWS KMS renvoie ensuite la clé de volume à l'hôte Amazon Elastic Compute Cloud (Amazon EC2) qui contient votre instance au cours de la session TLS.
4. La clé de volume est utilisée pour chiffrer et déchiffrer toutes les données en provenance et à destination du volume Amazon EBS associé. Amazon EBS conserve la clé de volume chiffrée pour utilisation ultérieure au cas où la clé de volume en mémoire ne serait plus disponible.

Pour plus d'informations sur le chiffrement des volumes Amazon EBS à l'aide de clés KMS, consultez la section Comment Amazon Elastic Block Store utilise AWS KMS le code du AWS Key Management Service développeur et le chiffrement Amazon EBS dans le guide de l'utilisateur Amazon EC2 et le guide de l'utilisateur Amazon EC2.

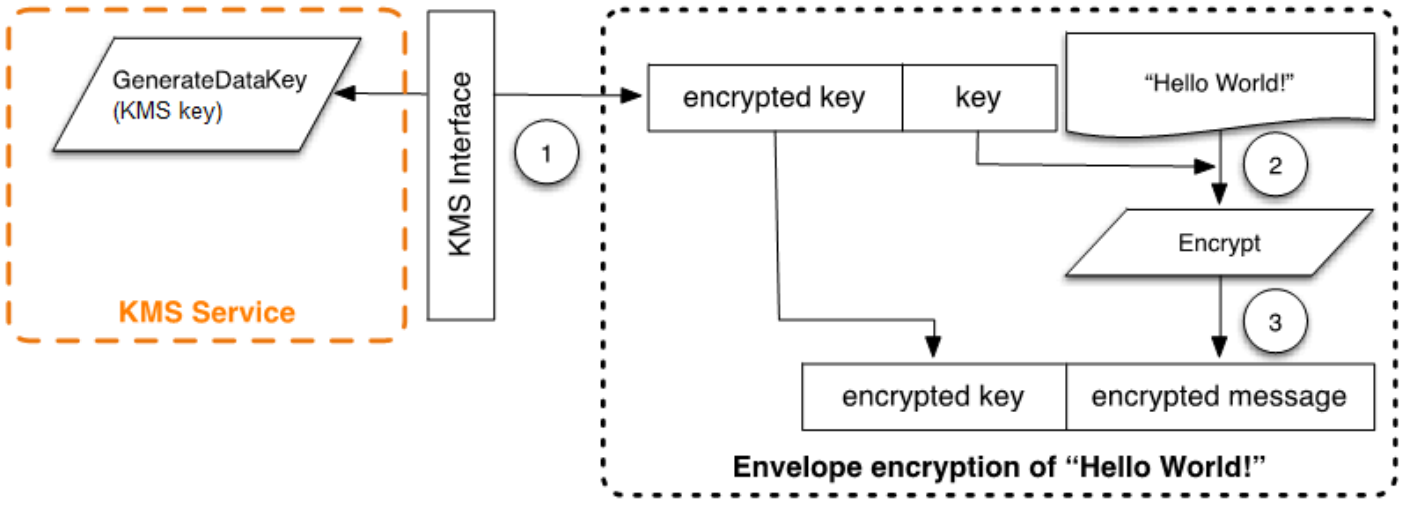
Chiffrement côté client

Le AWS Encryption SDK inclut une opération d'API permettant d'effectuer le chiffrement de l'enveloppe à l'aide d'une clé KMS. Pour obtenir des recommandations complètes et des détails sur l'utilisation, consultez la documentation associée. Les applications clientes peuvent utiliser le AWS Encryption SDK pour chiffrer les enveloppes à l'aide de 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);
```

L'application client peut exécuter les étapes suivantes :

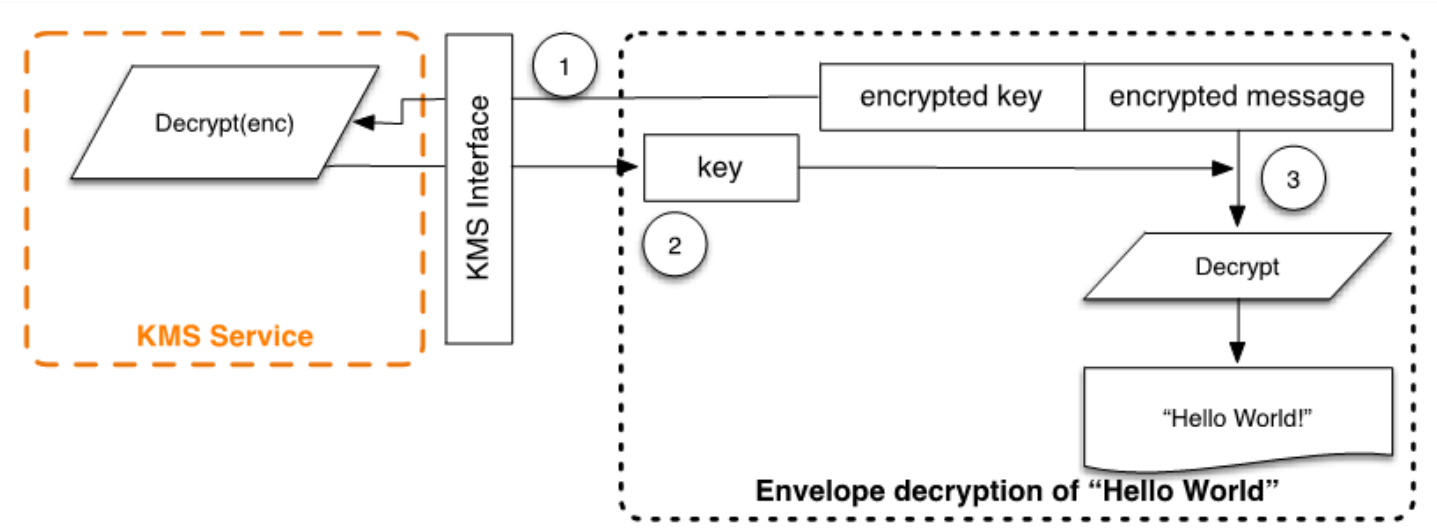
1. Une demande est faite sous une clé KMS pour une nouvelle clé de données. Une clé de données chiffrée et une version en texte brut de la clé de données sont renvoyées.
2. Dans le AWS Encryption SDK, la clé de données en texte brut est utilisée pour chiffrer le message. La clé de données en texte brut est alors supprimée de la mémoire.
3. La clé de données chiffrées et le message chiffré sont combinés en un seul tableau d'octets de texte chiffré.



Le message chiffré dans l'enveloppe peut être déchiffré à l'aide de la fonctionnalité de déchiffrement pour obtenir le message initialement chiffré.

```
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. Il AWS Encryption SDK analyse le message crypté par enveloppe pour obtenir la clé de données cryptée et fait une demande pour déchiffrer la clé AWS KMS de données.
2. AWS Encryption SDK reçoit la clé de données en texte brut de AWS KMS.
3. La clé de données est ensuite utilisée pour déchiffrer le message, en renvoyant le texte brut initial.

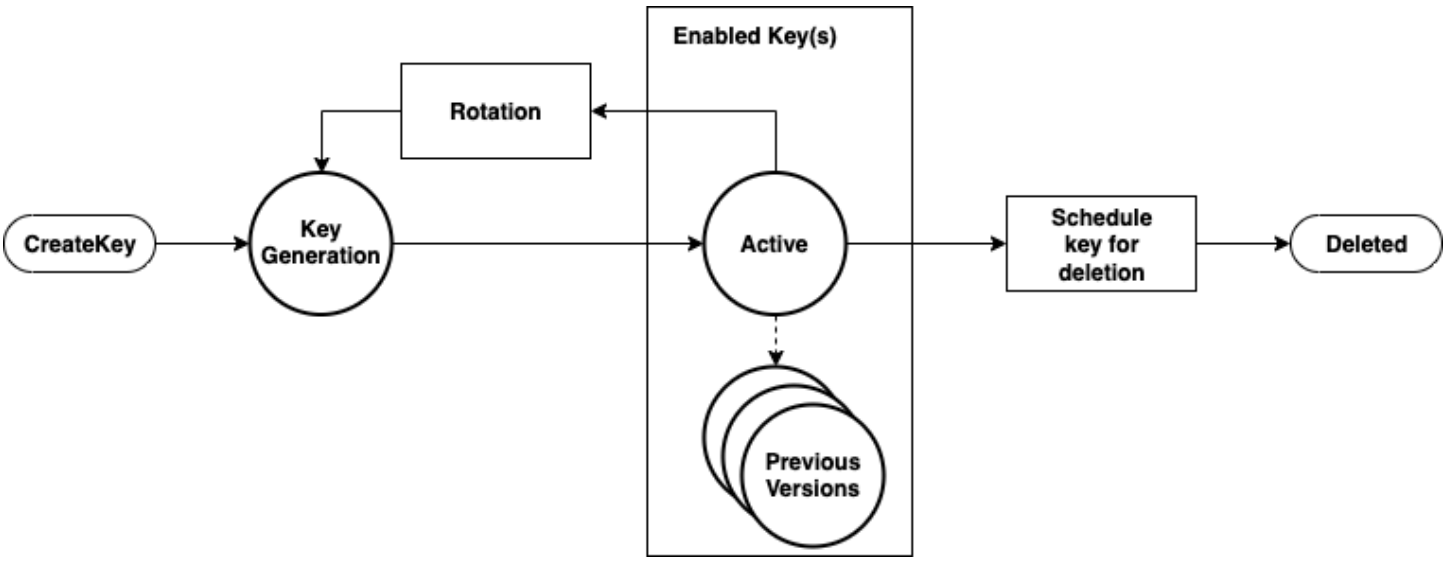


Utilisation de AWS KMS keys

Une clé AWS KMS key désigne une clé logique qui peut faire référence à une ou plusieurs clés de sauvegarde (HBK) du module de sécurité matériel (HSM). Cette rubrique explique comment créer une clé KMS, importer des éléments d'une clé, et comment activer, désactiver, effectuer une rotation et supprimer des clés KMS.

> **Note**
>
> AWS KMS remplace le terme clé principale client (CMK) par AWS KMS key et clé KMS. Le concept n'a pas changé. Pour éviter les changements de rupture, AWS KMS conserve quelques variations de ce terme.

Ce chapitre traite du cycle de vie d'une clé KMS, de sa création à sa suppression, comme illustré dans l'image suivante.



Rubriques

- Appel CreateKey
- Importation des éléments de clé
- Activation et désactivation des clés
- Suppression des clés
- Rotation des éléments de clé

Appel CreateKey

Une AWS KMS key est générée à la suite d'un appel à un `CreateKey` Appel d'API.

Les éléments suivants sont un sous-ensemble de la `CreateKey` syntaxe de demande.

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

Cette demande accepte les données suivantes au format JSON.

Description

(Facultatif) Description de la clé. Nous vous recommandons de choisir une description qui vous aide à déterminer si la clé est appropriée pour une tâche donnée.

KeySpec

Spécifie le type de clé KMS à créer. La valeur par défaut, SYMMETRIC_DEFAULT, crée une clé KMS de chiffrement symétrique. Ce paramètre est facultatif pour les clés de chiffrement symétriques et obligatoire pour toutes les autres spécifications de clé.

KeyUsage

Spécifie l'utilisation de la clé. Les valeurs valides sont `ENCRYPT_DECRYPT`, `SIGN_VERIFY` ou `GENERATE_VERIFY_MAC`. La valeur par défaut est `ENCRYPT_DECRYPT`. Ce paramètre est facultatif pour les clés de chiffrement symétriques et obligatoire pour toutes les autres spécifications de clé.

Origin

(Facultatif) Spécifie la source du matériel de clé pour la clé KMS. La valeur par défaut est `AWS_KMS`, ce qui indique que AWS KMS génère et gère le matériel de clé pour la clé KMS. Les autres valeurs valides comprennent `EXTERNAL`, qui représente une clé KMS créée sans matériel de clé pour le matériel de clé importé, et `AWS_CLOUDHSM` qui crée une clé KMS dans un magasin de clés personnalisé soutenu par un cluster AWS CloudHSM que vous contrôlez.

Politique

(Facultatif) Politique à associer à la clé. Si la politique est omise, la clé sera créée avec la politique par défaut (suivante) qui autorise le compte racine et les principaux IAM disposant des AWS KMS autorisations pour la gérer.

Pour de plus amples informations sur la stratégie, consultez Stratégies de clé dans AWS KMS et Stratégie de clé par défaut dans le Guide du développeur AWS Key Management Service.

La demande `CreateKey` renvoie une réponse qui inclut un ARN de clé.

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

Si l'`Origin` est `AWS_KMS`, après la création de l'ARN, une demande à une clé AWS KMS HSM sera effectuée sur une session authentifiée aux fins de provisionner une clé de sauvegarde (HBK) du module de sécurité matériel (HSM). La clé HBK est une clé de 256 bits associée à cet ID de clé de la clé KMS. Elle peut uniquement être générée sur une clé HSM et conçue pour ne jamais être exportée en dehors de la limite de la clé HSM en texte clair. La clé HBK est chiffrée sous la clé de domaine actuelle, DK_0. Ces clés HBK chiffrées sont appelées jetons de clé chiffrés (EKT). Bien qu'il soit possible de configurer les HSM pour utiliser une variété de méthodes d'enveloppement des clés, l'implémentation actuelle utilise AES-256 dans le Galois Counter Mode, un schéma de cryptage authentifié. Ce mode de chiffrement authentifié nous permet de protéger certaines métadonnées de jeton de clé exportées en texte clair.

Cette valeur est représentée comme :

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

Deux formes de protection fondamentales sont fournies à vos clés KMS et aux clés HBK ultérieures : les politiques d'autorisation définies sur vos clés KMS et les protections cryptographiques sur vos clés HBK associées. Les autres sections décrivent les protections cryptographiques et la sécurité des fonctions de gestion dans AWS KMS.

En plus de l'ARN, vous pouvez créer un nom convivial et l'associer à la clé KMS en créant un alias pour la clé. Après avoir associé un alias à une clé KMS, l'alias pourra être utilisé pour identifier la clé KMS dans les opérations cryptographiques. Pour plus d'informations, consultez la rubrique Utilisation d'alias dans le AWS Key Management Service Guide du développeur.

Plusieurs niveaux d'autorisations entourent l'utilisation des clés KMS. AWS KMS active des politiques d'autorisation distinctes entre le contenu chiffré et la clé KMS. Par exemple, un AWS KMS objet Amazon Simple Storage Service (Amazon S3) chiffré sous enveloppe hérite de la politique sur le compartiment Amazon S3. Toutefois, l'accès à la clé de chiffrement nécessaire est déterminé par la politique d'accès sur la clé KMS. Pour plus d'informations sur l'autorisation des clés KMS, consultez Authentification et contrôle d'accès pour AWS KMS dans le AWS Key Management ServiceGuide du développeur.

Importation des éléments de clé

AWS KMS fournit un mécanisme pour l'importation des éléments cryptographiques utilisés pour une clé HBK. Comme décrit dansAppel CreateKey, lorsque la CreateKey commande est utilisée avec `Origin` set to`EXTERNAL`, une clé KMS logique est créée qui ne contient aucun HBK sous-jacent. Les éléments cryptographiques doivent être importés à l'aide d'un `ImportKeyMaterial`Appel d'API. Vous pouvez utiliser cette fonction pour contrôler la création de clés et la durabilité des éléments cryptographiques. Si vous utilisez cette fonctionnalité, nous vous recommandons de faire preuve d'une grande prudence dans la manipulation et la durabilité de ces clés dans votre environnement. Pour obtenir des détails complets et des recommandations sur l'importation des éléments de clé, consultez Importation des éléments de clé dans le AWS Key Management Serviceguide du développeur.

Appel ImportKeyMaterial

La `ImportKeyMaterial` demande importe les éléments cryptographiques nécessaires pour la clé HBK. Les éléments cryptographiques doivent être une clé symétrique de 256 bits. Elle doit être chiffrée à l'aide de l'algorithme indiqué dans `WrappingAlgorithm` sous la clé publique retournée à partir d'un récente `GetParametersForImport` demande.

`Une ImportKeyMaterial demande` accepte les arguments suivants :

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

EncryptedKeyMaterial

Le matériel de clé importé chiffré avec la clé publique renvoyée dans une demande `GetParametersForImport` à l'aide de l'algorithme d'encapsulation spécifié dans cette demande.

ExpirationModel

Indique si les éléments de clé arrivent à expiration. Lorsque cette valeur est `KEY_MATERIAL_EXPIRES`, le `ValidTo` paramètre doit contenir une date d'expiration. Lorsque cette valeur est `KEY_MATERIAL_DOES_NOT_EXPIRE`, n'incluez pas les éléments du `ValidTo` paramètre. Les valeurs valides sont `"KEY_MATERIAL_EXPIRES"` et `"KEY_MATERIAL_DOES_NOT_EXPIRE"`.

ImportToken

Le jeton d'importation renvoyé par le même demande `GetParametersForImport` qui a fourni la clé publique.

KeyId

La clé KMS qui sera associée au matériau de clé importé. La `Origin` de la clé KMS doit être `EXTERNAL`.

Vous pouvez supprimer et réimporter le même matériel de clé importé dans la clé KMS spécifiée, mais vous ne pouvez pas importer ou associer la clé KMS à un autre matériel de clé.

ValidTo

(Facultatif) L'heure à laquelle les éléments de clé importés arrivent à expiration. Lorsque les éléments de clé expirent, AWS KMS supprime les éléments de clé et la clé KMS devient inutilisable. Ce paramètre est obligatoire lorsque la valeur de `ExpirationModel` est `KEY_MATERIAL_EXPIRES`. Sinon, elle n'est pas valide.

Lorsque la demande aboutit, la clé KMS peut être utilisée dans AWS KMS jusqu'à la date d'expiration spécifiée, le cas échéant. Après l'expiration des matériels de clé importés, la clé EKT est supprimée de la AWS KMS couche de stockage.

Activation et désactivation des clés

La désactivation d'une clé KMS empêche la clé d'être utilisée dans des opérations de chiffrement. Elle suspend la possibilité d'utiliser toutes les clés HBK associées à la clé KMS. L'activation restaure

l'utilisation des clé HBK et de la clé KMS. Activer et Désactiver sont de simples demandes qui acceptent uniquement l'ID de clé ou l'ARN de clé de la clé KMS.

Suppression des clés

Les utilisateurs autorisés peuvent utiliser l'API `ScheduleKeyDeletion` pour planifier la suppression d'une clé KMS et de toutes les clés HBK associées. Il s'agit d'une opération intrinsèquement destructrice, et vous devez faire preuve de prudence lorsque vous supprimez des clés de AWS KMS. AWS KMS applique un délai d'attente minimal de sept jours en cas de suppression de clés KMS. Pendant la période d'attente, la clé est placée dans un état désactivé avec un état de clé Suppression en attente. Tous les appels à utiliser la clé pour des opérations cryptographiques échoueront. ScheduleKeyDeletion prend les arguments suivants.

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

KeyId

L'identifiant unique de la clé KMS à supprimer. Pour préciser cette valeur, utilisez l'ID de clé unique ou l'ARN de clé de la clé KMS.

PendingWindowInDays

(Facultatif) La période d'attente en nombre de jours. Cette valeur est facultative. La plage est comprise entre 7 et 30 jours et la valeur par défaut est de 30 jours. À l'expiration de la période d'attente, AWS KMS supprime la clé KMS et toutes les cles HBK associées.

Rotation des éléments de clé

Les utilisateurs autorisés peuvent activer la rotation annuelle automatique de leurs clés KMS gérées par le client. les Clés gérées par AWS sont toujours soumises à rotation chaque année.

En cas de rotation d'une clé KMS, une nouvelle clé HBK sera créée et signalée comme la version actuelle du matériel de la clé pour toutes les nouvelles demandes de chiffrement. Toutes les versions précédentes de la clé HBK restent disponibles pour être utilisées à perpétuité pour déchiffrer les textes chiffrés qui ont été chiffrés à l'aide de cette version de clé HBK. Etant donné que AWS KMS ne stocke aucun texte chiffré chiffré sous une clé KMS, les textes chiffrés sous une ancienne clé HBK

ayant subi une rotation nécessitent que cette clé HBK pour être déchiffrés. Vous pouvez utiliser l'API `ReEncrypt` pour rechiffrer tout texte chiffré sous la nouvelle clé HBK pour la clé KMS ou sous une autre clé KMS sans exposer le texte en clair.

Pour plus d'informations sur l'activation et la désactivation de la rotation de clés, consultez la section Rotation des clés KMS AWS dans le AWS Key Management Service Guide du développeur.

Opérations sur les données client

Après avoir créé une clé KMS, il sera possible de l'utiliser pour effectuer des opérations cryptographiques. Chaque fois que des données sont chiffrées à l'aide d'une clé KMS, l'objet résultant est un texte chiffré client. Le texte chiffré comporte deux sections : une partie d'en-tête non chiffrée (ou texte clair), protégée par le schéma de chiffrement authentifié en tant que données authentifiées supplémentaires, et une partie chiffrée. La partie en texte clair comprend l'identificateur HBK (HBKID). Ces deux champs immuables de la valeur de texte chiffré permettent de s'assurer que AWS KMS pourra déchiffrer l'objet à l'avenir.

Rubriques

Génération des clés de données

Les utilisateurs autorisés peuvent utiliser l' GenerateDataKey API (et les API associées) pour demander un type spécifique de clé de données ou une clé aléatoire de longueur arbitraire. Cette rubrique fournit une vue simplifiée de cette opération API. Pour plus de détails, consultez les GenerateDataKey API dans la référence des AWS Key Management Service API.

- GenerateDataKey
- GenerateDataKeyWithoutPlaintext
- GenerateDataKeyPair
- GenerateDataKeyPairWithoutPlaintext

Voici la syntaxe de la demande `GenerateDataKey`.

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

```
}
```

La demande accepte les données suivantes au format JSON.

KeyId

Identifiant de la clé utilisée pour chiffrer la clé de données. Cette valeur doit identifier une clé KMS de chiffrement symétrique.

Ce paramètre est obligatoire.

NumberOfBytes

Un nombre entier qui contient le nombre d'octets à générer. Ce paramètre est obligatoire.

L'appelant doit fournir `KeySpec` ou `NumberOfBytes`, mais pas les deux.

EncryptionContext

(Facultatif) Nom : paire de valeur qui contient des données supplémentaires à authentifier lors des processus de chiffrement et de déchiffrement qui utilisent la clé.

GrantTokens

(Facultatif) Une liste de jetons d'octrois qui représentent les octrois qui fournissent des autorisations permettant de générer ou d'utiliser une clé. Pour en savoir plus sur les octrois et les jetons d'octrois, consultez Authentification et contrôle d'accès pour AWS KMS dans le AWS Key Management Service guide du développeur.

Après avoir authentifié la commande, AWS KMS, acquiert l'EKT actif actuel associé à la clé KMS. Il transmet l'EKT avec votre requête fournie et tout contexte de chiffrement à une clé HSM lors d'une session protégée entre le AWS KMS hôte et une clé HSM dans le domaine.

La clé HSM exécute les tâches suivantes :

1. Génère les éléments secrets demandés et les conserve dans la mémoire volatile.
2. Déchiffre l'EKT correspondant à l'ID de clé de la clé KMS définie dans la demande afin d'obtenir le HBK = Déchiffrer (DK_i, EKT).
3. Génère un nombre aléatoire à usage unique N.
4. Génère une clé de chiffrement K dérivée de l'AES-GCM 256 bits à partir d'une clé HBK et N.
5. Chiffre les éléments secrets texte chiffré = Chiffrer (K, contexte, secret).

La `GenerateDataKey` vous renvoie les éléments secrets en texte brut et le texte chiffré sur le canal sécurisé entre le AWS KMS hôte et la clé HSM, puis AWS KMS vous les envoie dans la session TLS. AWS KMS ne retient ni le texte brut ni le texte chiffré. Sans possession du texte chiffré, du contexte de chiffrement et de l'autorisation d'utiliser la clé KMS, le secret sous-jacent ne peut pas être renvoyé.

Voici la syntaxe de la réponse.

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

La gestion des clés de données vous est remise en tant que développeur de l'application. Pour les meilleures pratiques de chiffrement côté client avec des AWS KMS clés de données (mais pas de paires de clés de données), vous pouvez utiliser AWS Encryption SDK.

Leur rotation peut être effectuée à n'importe quelle fréquence. En outre, la clé de données elle-même peut être chiffrée à nouveau sur une autre clé KMS ou sur une clé KMS qui a subi une rotation à l'aide de l'outil `ReEncrypt` Opération d'API. Pour plus de détails, reportez-vous ReEncryptà la référence de AWS Key Management Service l'API.

Encrypt

Une fonction de base de AWS KMS consiste à chiffrer un objet sous une clé KMS. Par conception, AWS KMS fournit des opérations cryptographiques à faible latence sur les clés HSM. Par conséquent, il y a une limite de 4 Ko sur la quantité de texte brut qui peut être chiffré lors d'un appel direct à la fonction de chiffrement. La AWS Encryption SDK peut être utilisée pour chiffrer des messages plus volumineux.AWS KMS, après avoir authentifié la commande, acquiert l'EKT actif actuel relatif à la clé KMS. Il transmet l'EKT ainsi que le texte brut et le contexte de chiffrement à toute clé HSM disponible dans la région. Celles-ci sont envoyées sur une session authentifiée entre le AWS KMS hôte et une clé HSM dans le domaine.

La clé HSM exécute les opérations suivantes :

1. Déchiffre l'EKT pour obtenir la clé HBK = Déchiffrer (DK_i, EKT).
2. Génère un nombre aléatoire à usage unique N.
3. Dérive une clé de chiffrement K dérivée de l'AES-GCM 256 bits à partir d'une clé HBK et N.

4. Chiffre le texte brut ciphertext = Chiffrer (K, contexte, texte brut).

La valeur du texte chiffré vous est renvoyée, et ni les données en texte brut ni le texte chiffré ne sont conservés où que ce soit dans l'AWSinfrastructure. Sans possession du texte chiffré, du contexte de chiffrement et de l'autorisation d'utiliser la clé KMS, le texte brut sous-jacent ne peut pas être renvoyé.

Decrypt

Un appel à AWS KMS pour déchiffrer la valeur d'un texte chiffré accepte un texte chiffré de valeur et un contexte de chiffrement. AWS KMS authentifie l'appel en l'aide de laAWS signature version 4 demandes signées et extrait le HBKID pour la clé d'enveloppe à partir du texte chiffré. Le HBKID est utilisé pour obtenir l'EKT nécessaire pour déchiffrer le texte chiffré, l'ID de clé et la politique de l'ID de clé. La demande est autorisée en fonction de la politique de clé, des octrois qui peuvent être présents et des politiques IAM associées qui font référence à l'ID de clé. La fonction `Decrypt` est similaire à la fonction de chiffrement.

Voici la syntaxe de la demande `Decrypt`.

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

Voici les paramètres de la demande.

CiphertextBlob

Texte chiffré incluant les métadonnées.

EncryptionContext

(Facultatif) Le contexte de chiffrement. Si cela a été précisé dans la fonction `Encrypt`, cela doit être précisé ici sinon l'opération de déchiffrement échouera. Consultez Contexte de chiffrement dans le AWS Key Management Serviceguide du développeur pour en savoir plus.

GrantTokens

(Facultatif) Une liste de jetons d'octrois qui représentent les octrois qui fournissent des autorisations permettant d'effectuer le déchiffrement.

Le texte chiffré et l'EKT sont envoyés, avec le contexte de chiffrement, via une session authentifiée à une clé HSM pour déchiffrement.

La clé HSM exécute les opérations suivantes :

1. Déchiffre l'EKT pour obtenir la clé HBK = Déchiffrer (DK_i, EKT).
2. Extrait le nombre aléatoire à usage unique N à partir de la structure du texte chiffré.
3. Régénère une clé de chiffrement K dérivée de l'AES-GCM 256 bits à partir d'une clé HBK et N.
4. Déchiffre le texte chiffré pour obtenir le texte brut = Déchiffrer (K, contexte, texte chiffré).

L'ID de clé et le texte brut clair qui en résultent sont renvoyés au AWS KMS hôte sur la session sécurisée, puis reviennent à l'application client appelant via une connexion TLS.

Voici la syntaxe de la réponse.

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

Si l'application appelante veut s'assurer de l'authenticité du texte brut, elle doit vérifier que l'ID de clé retourné est celui attendu.

Rechiffrement d'un objet chiffré

Il est possible de rechiffrer un texte chiffré client existant chiffré sous une clé KMS sur une autre clé KMS à l'aide de la commande Rechiffrer. La commande Rechiffrer permet de rechiffrer les données côté serveur à l'aide d'une nouvelle clé KMS, sans exposer le texte brut des données côté client. Les données sont d'abord déchiffrées, puis rechiffrées.

Voici la syntaxe de la demande.

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

```
}
```

La demande accepte les données suivantes au format JSON.

CiphertextBlob

Texte chiffré des données à rechiffrer.

DestinationEncryptionContext

(Facultatif) Contexte de chiffrement à utiliser lorsque les données sont rechiffrées.

DestinationKeyId

Identificateur de clé de la clé utilisée pour rechiffrer les données.

GrantTokens

(Facultatif) Une liste de jetons d'octrois qui représentent les octrois qui fournissent des autorisations permettant d'effectuer le déchiffrement.

SourceKeyId

(Facultatif) Identificateur de clé de la clé utilisée pour déchiffrer les données.

SourceEncryptionContext

(Facultatif) Contexte de chiffrement utilisé pour chiffrer et déchiffrer les données spécifiées dans le `CiphertextBlob` paramètre.

Le processus combine les opérations de déchiffrement et de chiffrement des descriptions précédentes : le texte chiffré du client est déchiffré sous la clé HBK initiale référencée par le texte chiffré du client vers la clé HBK actuelle sous la clé KMS prévue. Lorsque les clés KMS utilisées dans cette commande sont identiques, cette commande déplace le texte chiffré du client d'une ancienne version d'une clé HBK vers la dernière version d'une clé HBK.

Voici la syntaxe de la réponse.

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

```
}
```

Si l'application appelante souhaite s'assurer de l'authenticité du texte en clair sous-jacent, elle doit vérifier que le texte SourceKeyId renvoyé est celui attendu.

AWS KMS opérations internes

AWS KMS internes sont nécessaires pour mettre à l'échelle et sécuriser les clés HSM pour un service de gestion des clés distribué à l'échelle mondiale.

Rubriques

Domaines et état du domaine

Une collection coopérative d'AWS KMSentités internes de confiance au sein d'un Région AWS est désigné comme un domaine. Un domaine comprend un ensemble d'entités de confiance, un ensemble de règles et un ensemble de clés secrètes, appelées « clés de domaine ». Les clés de domaine sont partagées entre les clés HSM membres du domaine. Un état de domaine se compose des champs suivants.

Nom

Un nom de domaine permettant d'identifier ce domaine.

Members

Une liste des clés HSM membres du domaine, notamment leur clé de signature publique et leurs clés d'accord public.

Opérateurs

Une liste d'entités, de clés de signature publiques et d'un rôle (AWS KMSopérateur ou hôte de service) qui représentent les opérateurs de ce service.

Règles

Une liste des règles de quorum pour chaque commande qui doit être satisfaite pour exécuter une commande sur la clé HSM.

Clés de domaine

Une liste des clés de domaine (clés symétriques) actuellement utilisées dans le domaine.

L'état complet du domaine est uniquement disponible sur la clé HSM. L'état du domaine est synchronisé entre les membres du domaine HSM en tant que jeton de domaine exporté.

Clés de domaine

Toutes les clés HSM d'un domaine partagent un ensemble de clés de domaine, $\{DK_r\}$. Ces clés sont partagées via une routine d'exportation d'état de domaine. L'état de domaine exporté peut être importé dans n'importe quelle clé HSM membre du domaine.

L'ensemble des clés de domaine, $\{DK_r\}$, inclut toujours une clé de domaine active et plusieurs clés de domaine désactivées. Une rotation quotidienne est effectuée sur les clés de domaine afin de s'assurer que AWS est conforme à la Recommandation pour la gestion des clés – Partie 1. Pendant la rotation de la clé de domaine, toutes les clés KMS existantes chiffrées sous la clé de domaine sortante sont à nouveau chiffrées sous la nouvelle clé de domaine active. La clé de domaine active est utilisée pour chiffrer les nouveaux EKT. Les clés de domaine expirées peuvent uniquement être utilisées pour déchiffrer les EKT précédemment chiffrés pendant un nombre de jours équivalent au nombre de clés de domaine qui ont récemment subi une rotation.

Jetons de domaine exportés

Il existe un besoin régulier de synchroniser l'état entre les participants du domaine. Ceci est effectué en exportant l'état du domaine chaque fois qu'une modification est apportée au domaine. L'état du domaine est exporté en tant que jeton de domaine exporté.

Nom

Un nom de domaine permettant d'identifier ce domaine.

Members

Une liste des clés HSM membres du domaine, notamment leurs clés publiques de signature et d'accord.

Opérateurs

Une liste d'entités, de clés de signature publiques et d'un rôle qui représentent les opérateurs de ce service.

Règles

Une liste des règles de quorum pour chaque commande qui doit être satisfaite pour exécuter une commande sur un membre de domaine HSM.

Clés de domaine chiffrées

Clés de domaine chiffrées par enveloppe. Les clés de domaine sont chiffrées par le membre de signature pour chacun des membres mentionnés ci-dessus, enveloppées dans leur clé d'accord public.

Signature

Une signature sur l'état du domaine générée par une clé HSM, nécessairement membre du domaine qui a exporté l'état du domaine.

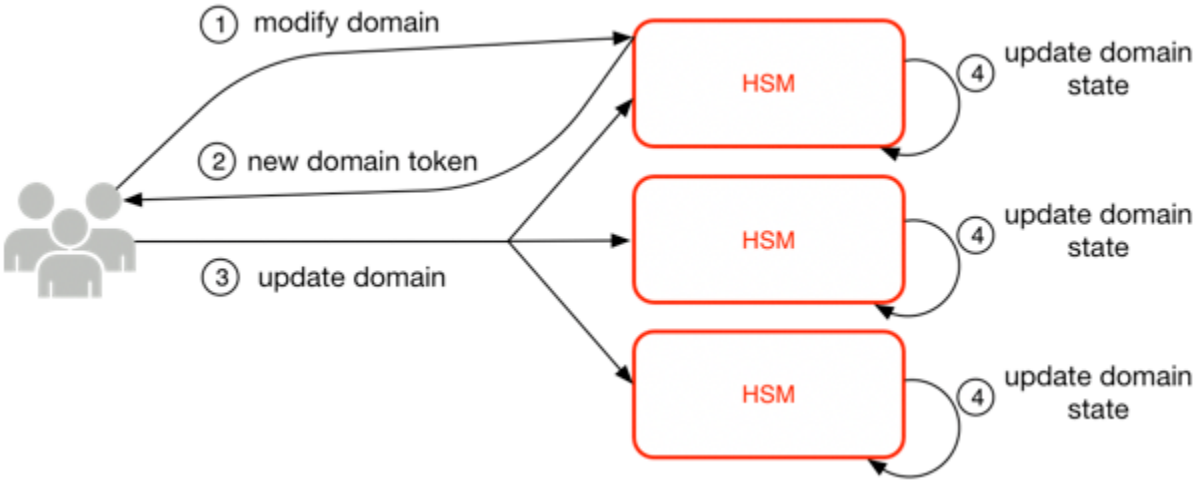
Le jeton de domaine exporté constitue la source essentielle de confiance pour les entités opérant dans le domaine.

Gestion des états de domaine

L'état du domaine est géré par des commandes authentifiées par quorum. Ces modifications comprennent la modification de la liste des participants de confiance dans le domaine, la modification des règles de quorum pour l'exécution des commandes HSM et la rotation régulière des clés de domaine. Ces commandes sont authentifiées sur une base individuelle, contrairement aux opérations de séance authentifiées ; comme illustré sur l’image suivante.

Dans son état initialisé et opérationnel, une HSM comporte un ensemble de clés d'identité asymétriques auto-générées, une paire de clés de signature et une paire de clés d'établissement de clé. Grâce à un processus manuel, un AWS KMS peut établir un domaine initial à créer sur une première clé HSM dans une région. Ce domaine initial se compose d'un état de domaine complet, tel que défini précédemment dans cette rubrique. Il est installé via une commande jointe à chacun des membres HSM définis dans le domaine.

Après qu'une clé HSM ait rejoint un domaine initial, elle est associée aux règles qui sont définies dans ce domaine. Ces règles régissent les commandes qui utilisent les clés cryptographiques du client ou qui modifient l'état de l'hôte ou du domaine. Les opérations d'API de session authentifiées qui utilisent vos clés cryptographiques ont été définies précédemment.



L'image ci-dessus illustre la manière de modifier un état de domaine. Le processus se compose de quatre étapes :

1. Une commande basée sur le quorum est envoyée à une clé HSM pour modifier le domaine.
2. Un nouvel état de domaine est généré et exporté en tant que nouveau jeton de domaine exporté. L'état de la clé HSM n'est pas modifié, ce qui signifie que le changement n'est pas appliqué à la clé HSM.
3. Une deuxième commande est envoyée à chacune des clés HSM dans le jeton de domaine récemment exporté aux fins de mettre à jour leur état de domaine avec le nouveau jeton de domaine.
4. Les clés HSM répertoriées dans le nouveau jeton de domaine exporté peuvent authentifier la commande et le jeton de domaine. Elles peuvent également décompresser les clés de domaine aux fins de mettre à jour l'état du domaine sur toutes les clés HSM du domaine.

Les clés HSM ne communiquent pas directement les unes avec les autres. Au lieu de cela, un quorum d'opérateurs demande une modification de l'état du domaine, ce qui se traduit par un nouveau jeton de domaine exporté. Un hôte de service membre du domaine est utilisé pour distribuer le nouvel état de domaine à chaque clé HSM du domaine.

La sortie d'un domaine et l'entrée dans un domaine sont réalisées à l'aide des fonctions de gestion des clés HSM. La modification de l'état du domaine est réalisée à l'aide des fonctions de gestion du domaine.

Quitter un domaine

Permet à une clé HSM de quitter un domaine, en supprimant tous les restes et les clés de ce domaine de la mémoire.

Entrer dans un domaine

Permet à une clé HSM d'entrer dans un nouveau domaine ou de mettre à jour son état actuel de domaine vers le nouvel état de domaine. Le domaine existant est utilisé comme source de l'ensemble initial de règles pour authentifier ce message.

Créer un domaine

Provoque la création d'un nouveau domaine sur une clé HSM. Renvoie un premier jeton de domaine qui peut être distribué aux clés HSM membres du domaine.

Modifier les opérateurs

Ajoute ou supprime des opérateurs de la liste des opérateurs autorisés et leurs rôles dans le domaine.

Modifier des membres

Ajoute ou supprime une clé HSM de la liste des clés HSM autorisées dans le domaine.

Modifier les règles

Modifie l'ensemble des règles de quorum requises pour exécuter des commandes sur une clé HSM.

Rotation des clés de domaine

Permet de créer une nouvelle clé de domaine et de la marquer comme clé de domaine active. Cela déplace la clé active existante vers une clé désactivée et supprime la clé désactivée la plus ancienne de l'état du domaine.

Sécurité des communications internes

Les commandes entre les hôtes de service ou AWS KMS les opérateurs et les clés HSM sont sécurisées à l'aide de deux mécanismes décrits dans Sessions authentifiées : une méthode de requête signée en quorum et une session authentifiée à l'aide d'un protocole hôte HSM Service.

Les commandes signées en quorum sont conçues de manière à ce qu'aucun opérateur ne puisse modifier les protections de sécurité essentielles fournies par les clés HSM. Les commandes qui s'exécutent sur les sessions authentifiées permettent de garantir que seuls les opérateurs de

service autorisés peuvent effectuer des opérations impliquant des clés KMS. Toutes les informations secrètes associées au client sont sécurisées dans l'AWS infrastructure.

Établissement de clé

Afin de sécuriser les communications internes, AWS KMS utilise deux méthodes d'établissement de clés différentes. La première est définie comme C (1, 2, ECC DH) dans la Recommandation pour les systèmes d'établissement de clés par paires utilisant la cryptographie par logarithme discret (révision 2). Ce système dispose d'un initiateur avec une clé de signature statique. L'initiateur génère et signe une clé Diffie-Hellman (ECDH) à courbe elliptique éphémère, conçue pour un destinataire disposant d'une clé d'accord ECDH statique. Cette méthode utilise une seule clé éphémère et deux clés statiques utilisant ECDH. Ceci est la dérivation de l'étiquette C (1, 2, ECC DH). Cette méthode est parfois appelée « ECDH à passage unique ».

La deuxième méthode d'établissement de clé est C (2, 2, ECC, DH). Dans ce système, les deux parties disposent d'une clé de signature statique, et elles génèrent, signent et échangent une clé ECDH éphémère. Cette méthode utilise deux clés statiques et deux clés éphémères, chacune utilisant ECDH. Ceci est la dérivation de l'étiquette C (2, 2, ECC, DH). Cette méthode est parfois appelée « ECDH éphémère » ou « ECDHE ». Toutes les clés ECDH sont générées sur la courbe secp384r1 (NIST-P384).

Limite de sécurité des clés HSM

La limite de sécurité interne de AWS KMS est la clé HSM. La clé HSM est dotée d'une interface propriétaire et ne possède aucune autre interface physique active dans son état opérationnel. Une clé HSM opérationnelle est provisionnée lors de son initialisation avec les clés cryptographiques nécessaires à l'établissement de son rôle dans le domaine. Les éléments cryptographiques sensibles de la clé HSM sont uniquement stockés dans une mémoire volatile et effacés que lorsque la clé HSM quitte l'état opérationnel, notamment lors des arrêts ou réinitialisations prévus ou non.

Les opérations de l'API de la clé HSM sont authentifiées soit par des commandes individuelles, soit par une session confidentielle mutuellement authentifiée établie par un hôte de service.

Commandes signées en quorum

Les commandes signées en quorum sont émises par les opérateurs aux clés HSM. Cette section décrit comment les commandes basées sur le quorum sont créées, signées et authentifiées. Ces règles sont assez simples. Par exemple, la commande Foo nécessite deux membres du rôle Bar pour être authentifiée. La création et la vérification d'une commande basée sur le quorum comporte trois étapes. La première étape est la création initiale de la commande ; la seconde est la soumission à d'autres opérateurs pour signature ; et la troisième est la vérification et l'exécution.

Aux fins de l'introduction des concepts, supposons qu'il existe un ensemble authentique de clés publiques et de rôles de l'opérateur $\{QOS_s\}$, et un ensemble de règles de quorum QR = $\{Command_i, Rule_{\{i, t\}}\}$ où chaque Règle est un ensemble de rôles et un nombre minimum N $\{Role_t, N_t\}$. Afin qu'une commande respecte la règle de quorum, le jeu de données de la commande doit être signé par un ensemble d'opérateurs répertoriés dans $\{QOS_s\}$ de façon à ce qu'ils répondent à l'une des règles répertoriées pour cette commande. Comme mentionné précédemment, l'ensemble des règles et des opérateurs du quorum sont stockés dans l'état du domaine et dans le jeton de domaine exporté.

Dans la pratique, un signataire initial signe la commande $Sig_1 = Sign(dO_{p1}, Command)$. Un second opérateur signe également la commande $Sig_2 = Sign(dO_{p2}, Command)$. Le message doublement signé est envoyé à une clé HSM pour exécution. La clé HSM effectue les tâches suivantes :

1. Pour chaque signature, elle extrait la clé publique du signataire de l'état du domaine et vérifie la signature sur la commande.
2. Elle vérifie que l'ensemble des signataires satisfait à une règle pour la commande.

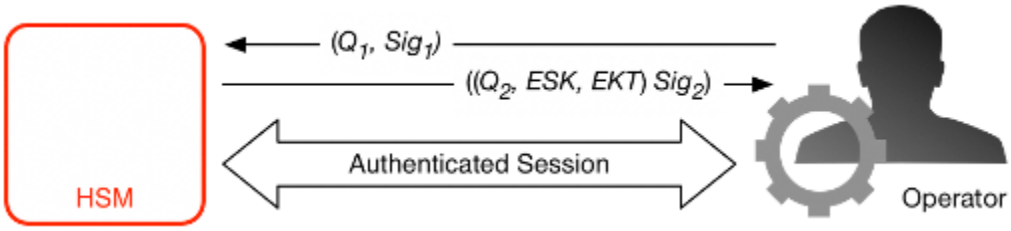
Sessions authentifiées

Vos opérations de clé s'exécutent entre les AWS KMS hôtes externes et les clés HSM. Ces commandes concernent la création et l'utilisation de clés cryptographiques et la génération sécurisée de nombres aléatoire. Les commandes s'exécutent sur un canal authentifié par une session entre les hôtes de service et les clés HSM. Outre le besoin d'authenticité, ces sessions exigent la confidentialité. Les commandes exécutées sur ces sessions comprennent le retour de clés de données en texte clair et de messages déchiffrés qui vous sont destinés. Pour s'assurer que ces sessions ne peuvent pas être subverties par man-in-the-middle des attaques, les sessions sont authentifiées.

Ce protocole exécute un accord de clé ECDHE mutuellement authentifié entre la clé HSM et l'hôte de service. L'échange est initié par l'hôte de service et achevé par la clé HSM. La clé HSM renvoie

également une clé de session (SK) chiffrée par la clé négociée et un jeton de clé exporté contenant la clé de session. Le jeton de clé exporté comporte une période de validité, après laquelle l'hôte de service devra renégocier une clé de session.

Un hôte de service est membre du domaine et possède une paire de clés identité-signature (DHO_i, $QHOS_i$) et une copie authentique des clés publiques d'identité des clés HSM. Il utilise son ensemble de clés identité-signature pour négocier en toute sécurité une clé de session qui peut être utilisée entre l'hôte de service et toute clé HSM du domaine. Les jetons de clé exportés ont une période de validité qui leur est associée, après laquelle une nouvelle clé devra être négociée.



Le processus commence par la reconnaissance de l'hôte de service qui a besoin d'une clé de session pour envoyer et recevoir des flux de communications sensibles entre lui-même et un membre de clé HSM du domaine.

1. Un hôte de service génère une paire de clés éphémères ECDH $(d)_1$, $Q._1$) et la signe avec sa clé d'identité Sig1 = Sig(dOS, Q_1).
2. La clé HSM vérifie la signature sur la clé publique reçue à l'aide de son jeton de domaine actuel et crée une paire de clés éphémères ECDH $(d)_2$, $Q._2$). Elle termine ensuite l'échange de clés ECDH conformément à la Recommandation pour les systèmes d'établissement de clés par paires utilisant la cryptographie par logarithme discret (révisée) afin de constituer une clé AES-GCM 256 bits négociée. La clé HSM génère une nouvelle clé de session AES-GCM 256 bits. Elle chiffre la clé de session à l'aide de la clé négociée afin de constituer la clé de session chiffrée (ESK). Elle chiffre également la clé de session sous la clé de domaine en tant que jeton de clé exporté EKT. Enfin, elle signe une valeur de retour à l'aide de sa paire de clés d'identité Sig_2 = Sign(dHSK, (Q_2, ESK, EKT)).
3. L'hôte de service vérifie la signature sur les clés reçues à l'aide de son jeton de domaine actuel. L'hôte de service effectue ensuite l'échange de clés ECDH conformément à la Recommandation pour les systèmes d'établissement de clés par paires utilisant la cryptographie par logarithme discret (révisée). Il déchiffre ensuite la clé ESK afin d'obtenir la clé de session SK.

Au cours de la période de validité dans l'EKT, l'hôte de service peut utiliser la clé de session négociée SK pour envoyer des commandes chiffrées par enveloppe à la clé HSM. Chaque service-host-initiated commande de cette session authentifiée inclut l'EKT. La clé HSM répond en utilisant la même clé de session SK négociée.

Processus de réplication pour clés multi-régions

AWS KMS utilise un mécanisme de réplication entre régions pour copier les éléments de clé d'une clé KMS d'un HSM d'une Région AWS à un HSM d'une autre Région AWS. Pour que ce mécanisme fonctionne, la clé KMS qui est répliquée doit être une clé multi-Régions. Lors de la réplication d'une clé KMS d'une région à une autre, les HSM des régions ne peuvent pas communiquer directement, car ils se trouvent dans des réseaux isolés. Au lieu de cela, les messages échangés pendant la réplication entre régions sont délivrés par un service proxy.

Pendant la réplication entre régions, chaque message généré par un HSM d'AWS KMS est signé de manière cryptographique à l'aide d'une clé de signature de réplication. Les clés de signature de réplication (RSK) sont des clés ECDSA sur la courbe NIST P-384. Chaque région possède au moins une clé RSK, et le composant publique de chaque clé RSK est partagé avec chaque autre région dans la même partition AWS.

Le processus de réplication entre régions pour copier les éléments de clé de la région A à la région B fonctionne comme suit :

1. Le HSM de la région B génère une clé ECDH éphémère sur la courbe NIST P-384, Replication Agreement Key B (RAKB). La composante publique de RAKB est envoyée à un HSM de la région A par le service proxy.
2. Le HSM de la région A reçoit la composante publique de RAKB, puis génère une autre clé ECDH éphémère sur la courbe NIST P-384, Replication Agreement Key A (RAKA). Le HSM exécute le schéma d'établissement de clé ECDH sur RAKA et la composante publique de RAKB, et dérive une clé symétrique de la sortie, la Replication Wrapping Key (RWK). La clé RWK est utilisée pour chiffrer les éléments de clé de la clé KMS multi-régions en cours de réplication.
3. La composante publique de RAKA et les éléments de clé chiffrés avec la clé RWK sont envoyés au HSM de la région B via le service proxy.
4. Le HSM de la région B reçoit la composante publique de RAKA et les éléments de clé chiffrés à l'aide de la clé RWK. Le HSM dérive la clé RWK en exécutant le schéma d'établissement de clé ECDH sur RAKB et la composante publique de RAKA.
5. Le HSM de la région B utilise la clé RWK pour déchiffrer la clé de la région A.

Protection de la durabilité

La durabilité supplémentaire du service pour les clés générées par le service est assurée par l'utilisation de HSM hors ligne, le stockage non volatile multiple des jetons de domaine exportés et le stockage redondant des clés KMS chiffrées. Les HSM hors connexion sont membres des domaines existants. À l'exception de ne pas être en ligne et de participer aux opérations de domaine régulières, les HSM hors ligne apparaissent de la même manière dans l'état du domaine que les membres HSM existants.

La conception de durabilité est destinée à protéger toutes les clés KMS d'une région si AWSdevait subir une perte à grande échelle des HSM en ligne ou de l'ensemble des clés KMS stockées dans notre système de stockage principal. AWS KMS keys avec des éléments de clé importés ne sont pas inclus dans les protections de durabilité accordées aux autres clés KMS. En cas de panne à l'échelle de la région en AWS KMS, les éléments de clé importé pourraient devoir être réimportés dans une clé KMS.

Les HSM hors ligne, ainsi que les informations d'identification pour y accéder, sont stockés dans des coffres-forts dans des salles de sécurité surveillées situées sur plusieurs sites géographiques indépendants. Chaque coffre-fort nécessite au moins un AWS agent de sécurité et un AWS KMS opérateur, issus de deux équipes indépendantes de AWS, pour obtenir ces éléments. L'utilisation de ces éléments est régie par une politique interne exigeant le quorum de AWS KMSopérateurs à être présents.

Référence

Utilisez les éléments de référence suivants pour obtenir des informations sur les abréviations, clés, contributeurs et sources cités dans le présent document.

Rubriques

Abréviations

La liste suivante illustre les abréviations citées dans le présent document.

AES

Norme de chiffrement avancée

CDK

clé de données client

DK

clé de domaine

ECDH

Diffie-Hellman

ECDHE

Courbe elliptique éphémère Diffie-Hellman

ECDSA

Algorithme de signature numérique à courbe elliptique

EKT

jeton de clé exporté

ESK

clé de session chiffrée

GCM

Galois Counter Mode

HBK

clé de sauvegarde HSM

HBKID

identifiant de clé de sauvegarde HSM

HSM

module de sécurité matérielle

RSA

Rivest Shamir et Adleman (cryptologique)

secp384r1

Standards for Efficient Cryptography prime 384-bit random curve 1

SHA256

Algorithme de hachage sécurisé de longueur du condensé de 256 bits

Clés

La liste suivante définit les clés référencées dans le présent document.

HBK

Clé de sauvegarde HSM : les clés de sauvegarde HSM sont des clés racine de 256 bits, à partir desquelles des clés d'utilisation spécifiques sont dérivées.

DK

Clé de domaine : une clé de domaine est une clé AES-GCM de 256 bits. Elle est partagée entre tous les membres d'un domaine et utilisée pour protéger les éléments des clés de sauvegarde HSM et les clés de session hôte de service HSM.

DKEK

Clé de chiffrement de clé de domaine : une clé de chiffrement de clé de domaine est une clé AES-256-GCM générée sur un hôte et utilisée pour chiffrer l'ensemble actuel des clés de domaine qui synchronisent l'état du domaine sur les hôtes HSM.

(dHAK,QHAK)

Paire de clés d'accord HSM : chaque clé HSM initiée dispose d'une paire de clés d'accord Diffie-Hellman à courbe elliptique générée localement sur la courbe secp384r1 (NIST-P384).

(dE, QE)

Paire de clés d'accord éphémère : les clés HSM et les hôtes de service génèrent des clés d'accord éphémères. Ce sont des clés Diffie-Hellman à courbe elliptique sur la courbe secp384r1 (NIST-P384). Elles sont générées dans deux cas d'utilisation : pour établir une clé de host-to-host chiffrement pour transporter les clés de chiffrement de clé de domaine sous forme de jetons de domaine et pour établir des clés de session hôte du service HSM afin de protéger les communications sensibles.

(dHSK,QHSK)

Paire de clés de signature HSM : chaque clé HSM initiée dispose d'une paire de clés de signature numérique à courbe elliptique générée localement sur la courbe secp384r1 (NIST-P384).

(dOS, QOS)

Paire de clés de signature de l'opérateur : les opérateurs de l'hôte de service et AWS KMS les opérateurs disposent d'une clé de signature d'identité utilisée pour s'authentifier auprès d'autres participants du domaine.

K

Clé de chiffrement des données : clé AES-GCM 256 bits dérivée d'une clé HBK utilisant la courbe NIST SP800-108 KDF en mode compteur utilisant HMAC avec SHA256.

SK

Clé de session : une clé de session est créée à la suite d'une clé Diffie-Hellman à courbe elliptique authentifiée échangée entre un opérateur hôte de service et une clé HSM. Le but de l'échange est de sécuriser les communicatiosn entre l'hôte de service et les membres du domaine.

Collaborateurs

Les personnes et organisations suivantes ont contribué à l'élaboration du présent document :

- Ken Beer, General Manager – KMS, AWS Cryptography
- Matthew Campagna, ingénieur principal de la sécurité, AWS Cryptography

Bibliographie

Pour obtenir des informations sur AWS Key Management Service les clés HSM, accédez au Centre de ressources de la sécurité informatique NIST , recherchez la page du programme de validation des modules cryptographiques, puis recherchez AWS Key Management Service Clés HSM.

Amazon Web Services, General Reference (Version 1.0), “Signing AWS API Request,” http://docs.aws.amazon.com/general/latest/gr/signing_aws_api_requests.html.

Amazon Web Services, “What is the AWS Encryption SDK,” http://docs.aws.amazon.com/encryption-sdk/latest/developer-guide/introduction.html.

Federal Information Processing Standards Publications, FIPS PUB 180-4. Secure Hash Standard, August 2012. Disponible sur https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf.

Federal Information Processing Standards Publication 197, Announcing the Advanced Encryption Standard (AES), November 2001. Disponible sur http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

Federal Information Processing Standards Publication 198-1, The Keyed-Hash Message Authentication Code (HMAC), July 2008. Disponible sur http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf.

Publication spéciale 800-52 du NIST, révision 2, Directives pour la sélection, la configuration et l'utilisation des implémentations du protocole TLS (Transport Layer Security), août 2019. https://nvlpubs.nist.gov/nistpubs/ SpecialPublications /NIST.SP.800-52R2.pdf.

PKCS#1 v2.2: RSA Cryptography Standard (RFC 8017), Internet Engineering Task Force (IETF), November 2016. https://tools.ietf.org/html/rfc8017.

Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, NIST Special Publication 800-38D, November 2007. Disponible sur http://csrc.nist.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 Special Publication 800-38E, January 2010. Disponible sur https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38e.pdf.

Recommendation for Key Derivation Using Pseudorandom Functions, NIST Special Publication 800-108, October 2009, disponible sur https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-108.pdf.

Recommendation for Key Management - Part 1: General (Revision 5), NIST Special Publication 800-57A, May 2020, disponible sur https://doi.org/10.6028/NIST.SP.800-57pt1r5.

Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised), NIST Special Publication 800-56A Revision 3, April 2018. Disponible à l'adresse https://nvlpubs.nist.gov/nistpubs/ SpecialPublications /NIST.SP.800-56AR3.pdf.

Recommandation pour la génération de nombres aléatoires à l'aide de générateurs de bits aléatoires déterministes, publication spéciale du NIST 800-90A, révision 1, juin 2015, disponible sur https://nvlpubs.nist.gov/nistpubs/ /NIST.sp.800-90AR1.pdf. SpecialPublications

SEC 2: Recommended Elliptic Curve Domain Parameters, Standards for Efficient Cryptography Group, Version 2.0, 27 January 2010.

Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS), Brown, D., Turner, S., Internet Engineering Task Force, July 2010, http://tools.ietf.org/html/rfc5753/.

X9.62-2005: Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), American National Standards Institute, 2005.

Historique du document pour AWS KMS Cryptographic Details

Le tableau suivant décrit les modifications importantes apportées à la documentation pour AWS Key Management Service Cryptographic Details. Nous mettons aussi la documentation à jour régulièrement pour prendre en compte les commentaires qui nous sont envoyés.

Modification	Description	Date
Contenu mis à jour	Ajout de détails sur l'implémentation de l'opération AWS KMS d'`ReplicateKey` .	28 octobre 2021
Modification de la documentation	Remplacez le terme clé principale client (CMK) par AWS KMS key et clé KMS.	30 août 2021
Première version	Création de ce guide à partir du document technique KMS Cryptographic Details	30 décembre 2020

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.