AWS KMS Détails cryptographiques

AWS Key Management Service

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Key Management Service: AWS KMS Détails cryptographiques

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques et la présentation commerciale d’Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n’est pas d’Amazon, d’une manière susceptible de créer une confusion parmi les clients, ou d’une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d’Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

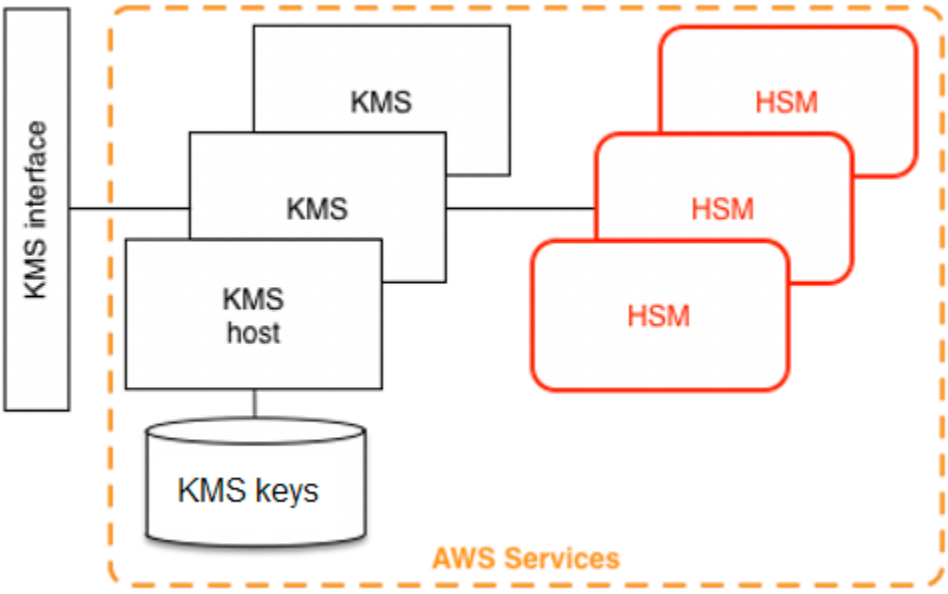
Table of Contents

Présentation des détails cryptographiques de AWS KMS

AWS Key Management Service (AWS KMS) fournit une interface Web pour générer et gérer des clés cryptographiques et fonctionne en tant que fournisseur de services cryptographiques pour protéger les données. AWS KMS propose des services traditionnels de gestion des clés intégrés à AWS des services visant à fournir une vue cohérente des clés des clients AWS, avec une gestion et un audit centralisés. Ce livre blanc fournit une description détaillée des opérations cryptographiques afin de vous aider AWS KMS à évaluer les fonctionnalités proposées par le service.

AWS KMS inclut une interface Web via l' AWS Management Console interface de ligne de commande et des opérations d' RESTfulAPI pour demander les opérations cryptographiques d'un parc distribué de modules de sécurité matériels validés par la norme FIPS 140-3 (HSMs) [1]. Le AWS KMS HSM est une appliance cryptographique matérielle autonome multipuce conçue pour fournir des fonctions cryptographiques dédiées répondant 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 ne sont disponibles que sur le HSMs 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. Ce n'est que dans le cadre des rôles et des comptes AWS IAM administrés par chaque client que les clés KMS du client peuvent être créées, supprimées ou utilisées pour chiffrer, déchiffrer, signer ou vérifier des données. Vous pouvez définir des contrôles d'accès pour déterminer qui peut gérer and/or l'utilisation des clés KMS en créant une politique attachée 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 de clés KMS. Cette fonctionnalité permet aux clients de contrôler comment et quand les AWS services peuvent accéder aux données chiffrées en contrôlant comment et quand les clés KMS sont accessibles.



AWS KMS est un service à plusieurs niveaux composé d' AWS KMS hôtes accessibles sur le Web et d'un niveau de. HSMs Le regroupement de ces hôtes hiérarchisés forme la AWS KMS pile. Toutes les demandes AWS KMS doivent être effectuées via le protocole TLS (Transport Layer Security) et se terminer sur un AWS KMS hôte. AWS KMS les hôtes n'autorisent le TLS qu'avec une suite de chiffrement qui assure une parfaite confidentialité des transmissions. AWS KMS authentifie et autorise vos demandes en utilisant les mêmes mécanismes d'identification et de politique Gestion des identités et des accès AWS (IAM) que ceux disponibles pour toutes les autres opérations d'API. AWS

Concepts de base

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

AWS KMS key

> **Note**
>
> AWS KMS remplace le terme clé principale du client (CMK) par AWS KMS keyclé KMS. Le concept n'a pas changé. Pour éviter des modifications AWS KMS intempestives, certaines variantes de ce terme sont conservées.

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— AWS crée et contrôle le cycle de vie et les politiques clés de Clés gérées par AWS, qui sont les ressources d'un client Compte AWS. Les clients peuvent consulter les politiques d'accès et les CloudTrail événements relatifs à ces clés Clés 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 AWS pour des opérations de chiffrement internes sur 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 manière interchangeable avec l'identifiant clé dans de nombreuses opérations d' AWS KMS API.

Autorisations

Politique associée à une clé KMS qui définit les autorisations sur la clé. La politique par défaut autorise tous les principes que vous définissez, ainsi que l'ajout de politiques IAM faisant référence Compte AWS 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 subventions consiste à définir des autorisations délimitées sur la manière 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 le HSMs, 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 AWS KMS, parfois appelée « 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 contenant des informations supplémentaires associées à des informations protégées AWS KMS. AWS KMS utilise un chiffrement authentifié pour protéger les clés de données. Le contexte de chiffrement est intégré à l'AAD du chiffrement authentifié dans les textes chiffrés chiffrés AWS KMS. 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 crée une AWS CloudTrail entrée qui inclut « nom de la clé » : « clé de liaison montante satellite ». 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. À l'instar des clés KMS symétriques, les clés privées sont cryptées HSMs. 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 exigences suivantes.

Durabilité

La durabilité des clés cryptographiques est conçue pour égaler celle des services les plus durables en 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 à un rôle aux contrôles d'accès basés sur le quorum sont tenus d'effectuer des actions administratives sur le. HSMs

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 aux 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 KMS enregistre l'utilisation et la gestion des clés cryptographiques dans des AWS CloudTrail journaux. Vous pouvez utiliser AWS CloudTrail les journaux pour contrôler l'utilisation de vos clés cryptographiques, y compris l'utilisation des clés par les AWS services en votre nom.

Pour atteindre ces objectifs, le AWS KMS système inclut un ensemble d' AWS KMS opérateurs et d'opérateurs hôtes de services (collectivement, les « opérateurs ») qui administrent les « domaines ». Un domaine est un ensemble de AWS KMS serveurs et d'opérateurs défini au niveau régional. HSMs Chaque AWS KMS opérateur dispose d'un jeton matériel qui contient une paire de clés privée et publique utilisée pour authentifier ses actions. Ils HSMs disposent d'une paire de clés privée et publique supplémentaire pour établir des clés de chiffrement qui protègent la synchronisation de l'état du HSM.

Ce paper explique comment AWS KMS protège vos clés et les 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 ».

AWS Key Management Service fondations

Les rubriques de ce chapitre décrivent les primitives cryptographiques AWS Key Management Service et l'endroit 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 rapidement passer d'un algorithme ou d'un 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 le AWS KMS HSMs. Ils HSMs implémentent un générateur de nombres aléatoires hybride qui utilise le NIST SP800-90A Deterministic Random Bit Generator (DRBG) CTR_DRBG using 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 ce cadre HSMs utilisent les normes de chiffrement avancées (AES), en mode compteur Galois (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.

AWS Omet souvent l'inclusion de l'AAD dans nos descriptions, en particulier lorsqu'il s'agit du 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 offre la possibilité d'importer du matériel clé dans un AWS KMS key au lieu de compter sur AWS KMS celui-ci pour générer le matériel clé. Ce matériel clé importé peut être chiffré à l'aide de RSAES-OAEP ou RSAES- PKCS1 -v1_5 pour protéger la clé pendant le transport vers le HSM. AWS KMS Les paires de clés RSA sont générées sur AWS KMS HSMs. Le matériel clé importé est déchiffré sur un AWS KMS HSM et rechiffré 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 à 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 sort jamais AWS KMS non chiffrée. Vous pouvez utiliser la clé publique interne AWS KMS en appelant les opérations de l' AWS KMS API, ou télécharger la clé publique et l'utiliser en dehors de AWS KMS.

AWS KMS supporte trois 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.
- Cryptographie post-quantique - Nouveaux algorithmes cryptographiques à clé publique résistants à l'informatique quantique. Supporte l'algorithme de signature numérique NIST FIPS 204 Module-Lattice (ML-DSA) avec des tailles de clé ML_DSA_44, ML_DSA_65 et ML_DSA_87.

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 le KDF en mode compteur en 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 de 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 les publications sur les normes fédérales de traitement de l'information, 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 long terme et d'automatiser le processus de chiffrement des enveloppes de vos données.

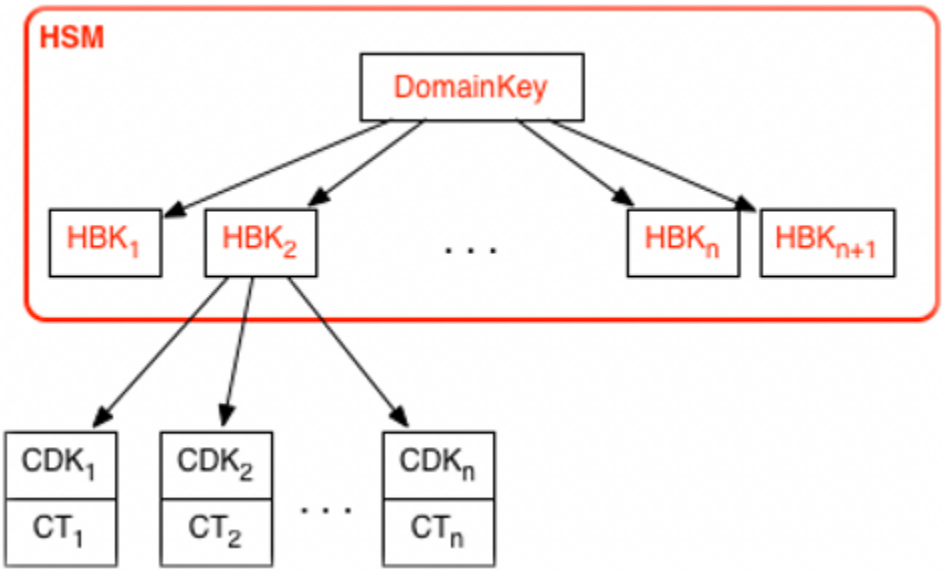
Outre les fonctionnalités de chiffrement fournies par le AWS KMS service, le SDK de AWS chiffrement fournit des bibliothèques de chiffrement d'enveloppes côté client. Vous pouvez utiliser ces bibliothèques pour protéger vos données et les clés de chiffrement utilisées pour chiffrer ces données.

AWS KMS key hiérarchie

Votre hiérarchie de clés commence par une clé logique de haut niveau, 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 sur la base d'une demande initiée par l'utilisateur via AWS KMS. À réception, AWS KMS demande la création d'une clé de sauvegarde HSM initiale (HBK) 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 jetons exportés HBKs sont appelés jetons clés exportés (EKTs).

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 sur vos clés KMS comme pour toute autre ressource AWS 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 AWS KMS, une nouvelle clé HBK est créée et associée à la clé KMS en tant que HBK active pour la clé KMS. HBKs Les plus anciens sont conservés et peuvent être utilisés pour déchiffrer et vérifier les données précédemment protégées. Mais seule la clé cryptographique active peut être utilisée pour protéger de nouvelles informations.



Vous pouvez demander AWS KMS à utiliser vos clés KMS pour protéger directement les informations ou demander des clés supplémentaires générées par HSM qui sont protégées par votre clé KMS.

Ces clés sont appelées clés de données client, ou CDKs. CDKs peuvent être renvoyés chiffrés sous forme de texte chiffré (CT), en texte brut ou les deux. Tous les objets chiffrés sous une clé KMS (qu'il s'agisse de données fournies par le client ou de clés générées par HSM) ne peuvent être déchiffrés que sur un HSM via un appel. AWS KMS

Le texte chiffré renvoyé, ou la charge utile déchiffrée, n'y est jamais stocké. AWS KMS Les informations vous sont retournées via votre connexion TLS à AWS KMS. Cela s'applique également aux appels effectués par AWS les 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	Rotation et utilisation

Clé	Description	Cycle de vie
	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.	contrôlée par application

[1] AWS KMS peut de temps à autre assouplir la rotation des clés de domaine à une rotation hebdomadaire au maximum pour tenir compte des tâches d'administration et de configuration du domaine.

[2] Les valeurs par défaut Clés gérées par AWS créées et gérées en votre AWS KMS nom font l'objet d'une rotation annuelle automatique.

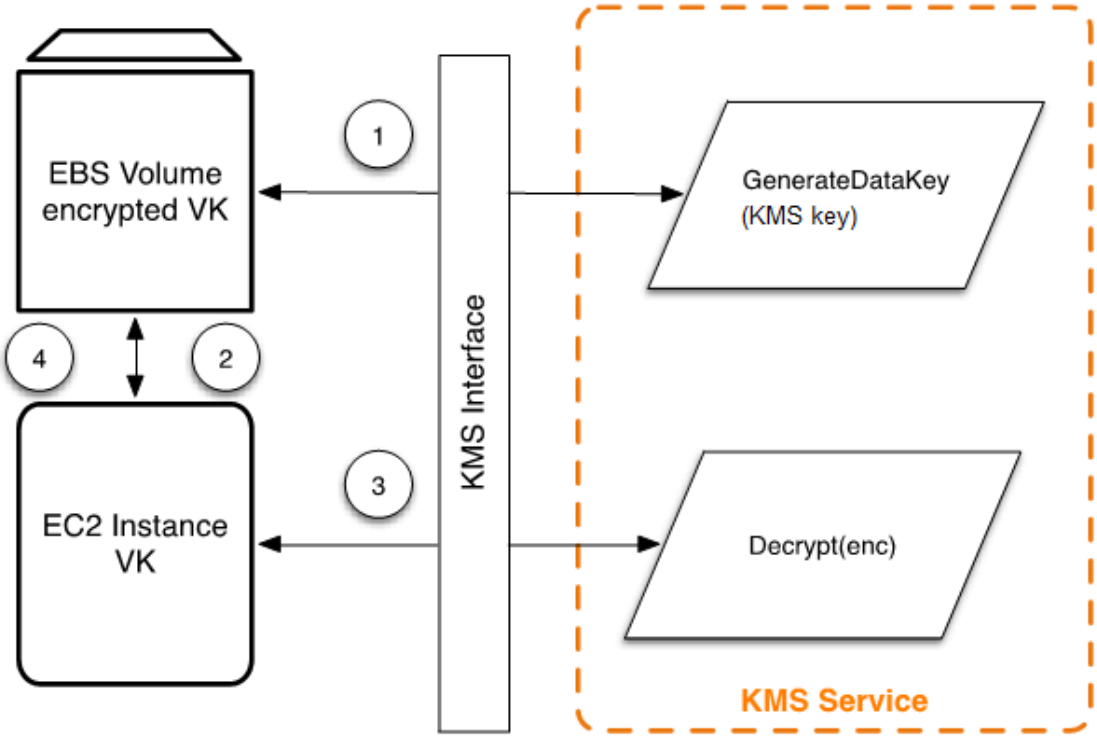
AWS KMS cas d'utilisation

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 dans le manuel du AWS Key Management Service développeur et le chiffrement Amazon EBS dans le guide de l' EC2 utilisateur Amazon et le guide de l'utilisateur Amazon EC2 .

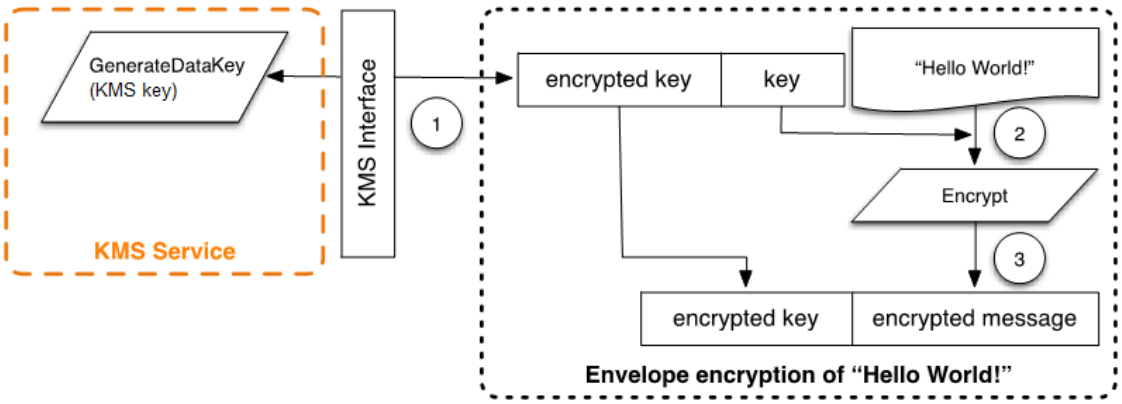
Chiffrement côté client

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.

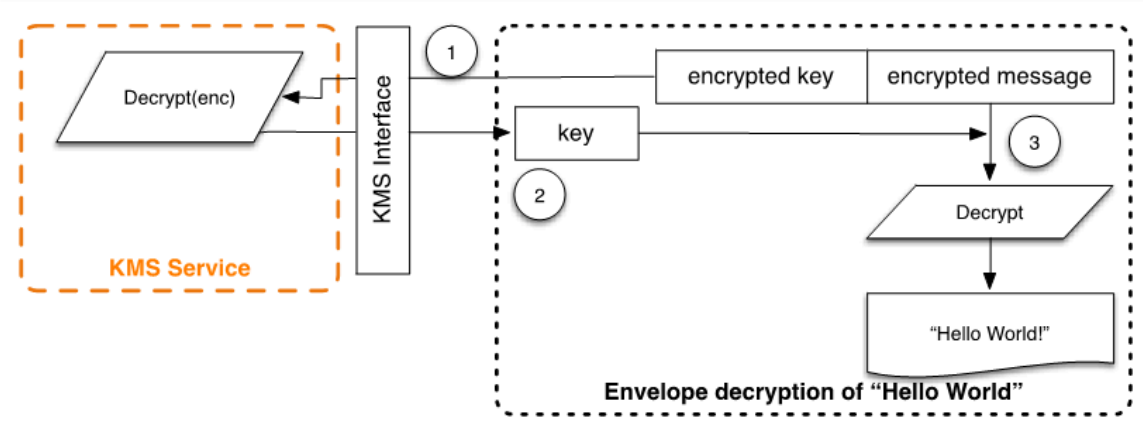


Le message chiffré dans l'enveloppe peut être déchiffré à l'aide de la fonctionnalité de déchiffrement pour obtenir le message initialement chiffré.

```
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.

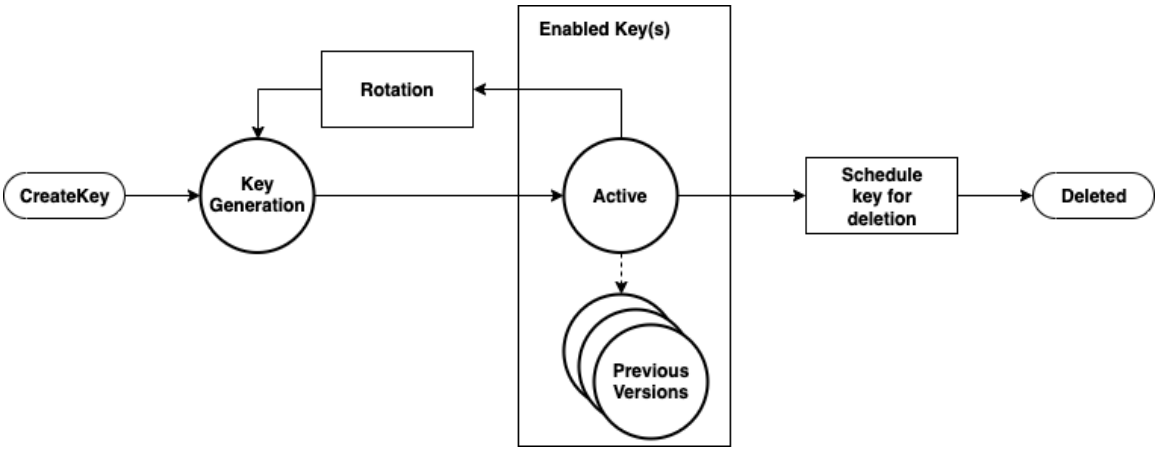


Travailler avec AWS KMS keys

An AWS KMS key fait référence à une clé logique qui peut faire référence à une ou plusieurs clés de sauvegarde du module de sécurité matériel (HSM) (HBKs). 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 du client (CMK) par AWS KMS keyclé KMS. Le concept n'a pas changé. Pour éviter des modifications AWS KMS intempestives, certaines variantes de ce terme sont conservées.

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 de clés
- Suppression de clés
- Rotation des éléments de clé

Appel CreateKey

Un AWS KMS key est généré à la suite d'un appel à l'appel `CreateKey`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 le matériel clé de la clé KMS est AWS KMS généré et géré. Les autres valeurs valides incluent`EXTERNAL`, qui représente une clé KMS créée sans matériel clé pour le matériel clé importé, et `AWS_CLOUDHSM` qui crée une clé KMS dans un magasin de clés personnalisé soutenu par un AWS CloudHSM cluster 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 jetons chiffrés HBKs sont appelés jetons clés chiffrés (EKTs). Bien qu'elle HSMs puisse être configurée pour utiliser diverses méthodes d'encapsulation de clés, l'implémentation actuelle utilise l'AES-256 en mode compteur Galois (GCM), 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(DK0, HBK)
```

Deux formes fondamentales de protection sont fournies à vos clés KMS et aux suivantes HBKs : les politiques d'autorisation définies sur vos clés KMS et les protections cryptographiques sur les clés associées HBKs. Les sections restantes 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'autorisation 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 Service Guide du développeur.

Importation des éléments de clé

AWS KMS fournit un mécanisme pour importer le matériel cryptographique utilisé pour un 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 Service guide 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 AWS KMS jusqu'à la date d'expiration spécifiée, si elle est fournie. Une fois le matériel clé importé expiré, l'EKT est supprimé de la couche de AWS KMS stockage.

Activation et désactivation de clés

La désactivation d'une clé KMS empêche la clé d'être utilisée dans des opérations de chiffrement. Cela suspend la possibilité d'utiliser tout HBKs ce qui est associé à la clé KMS. L'activation rétablit

l'utilisation de la clé HBKs 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 de clés

Les utilisateurs autorisés peuvent utiliser l'`ScheduleKeyDeletion`API pour planifier la suppression d'une clé KMS et de toutes les clés associées HBKs. 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 impose un délai d'attente minimal de sept jours lors de la suppression des 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. Une fois la période d'attente terminée, AWS KMS supprime la clé KMS et toutes les clés associées HBKs.

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. Étant donné qu'il AWS KMS ne stocke aucun texte chiffré sous une clé KMS, les textes chiffrés sous une ancienne clé HBK pivotée

nécessitent que celle-ci soit déchiffrée. 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 du texte chiffré permettent de garantir que l'objet AWS KMS pourra être déchiffré à l'avenir.

Rubriques

Génération des clés de données

Les utilisateurs autorisés peuvent utiliser l' GenerateDataKey API (et les APIs applications 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 le GenerateDataKey APIs document de référence de l'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 la demande que vous avez fournie et tout contexte de chiffrement à un HSM via une session protégée entre l' AWS KMS hôte et un HSM du 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).

`GenerateDataKey`vous renvoie le contenu secret en texte clair et le texte chiffré via le canal sécurisé entre l' AWS KMS hôte et le HSM. AWS KMS puis vous l'envoie via la session TLS. AWS KMS ne conserve 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 appliquer les meilleures pratiques de chiffrement côté client à l'aide de clés de AWS KMS données (mais pas de paires de clés de données), vous pouvez utiliser le. 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, consultez ReEncryptla référence de AWS Key Management Service l'API.

Encrypt

L'une des fonctions de base de AWS KMS est de chiffrer un objet sous une clé KMS. De par sa conception, AWS KMS fournit des opérations cryptographiques à faible latence sur HSMs. 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. Ils AWS Encryption SDK peuvent être utilisés 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. Ils sont envoyés via une session authentifiée entre l' AWS KMS hôte et un HSM du 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 dans l'infrastructure. AWS 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 AWS KMS à pour déchiffrer une valeur de texte chiffré accepte une valeur chiffrée (texte chiffré) et un contexte de chiffrement. AWS KMS authentifie l'appel à l'aide de requêtes signées version 4 de AWS signature et extrait le HBKID pour la clé d'encapsulation 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 Service guide 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'identifiant de clé et le texte en clair qui en résultent sont renvoyés à l' AWS KMS hôte via la session sécurisée, puis renvoyés à l'application client appelante 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 des composants internes sont nécessaires pour assurer l'évolutivité et la sécurité HSMs d'un service de gestion des clés distribué dans le monde entier.

Rubriques

Domaines et état du domaine

Un ensemble coopératif d' AWS KMS entités internes fiables au sein d'un Région AWS est appelé 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 HSMs les 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 de ceux HSMs qui sont membres du domaine, y compris leur clé de signature publique et leurs clés d'accord public.

Opérateurs

Liste d'entités, de clés de signature publiques et d'un rôle (AWS KMS opérateur ou hôte du service) représentant 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

Tous les HSMs membres 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. Les clés de domaine font l'objet d'une rotation quotidienne afin de garantir AWS leur conformité 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 toute nouvelle EKTs clé. Les clés de domaine expirées ne peuvent être utilisées que pour déchiffrer le chiffrement précédemment chiffré EKTs pendant un nombre de jours équivalent au nombre de clés de domaine récemment modifiées.

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 de ceux HSMs qui sont membres du domaine, y compris 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.

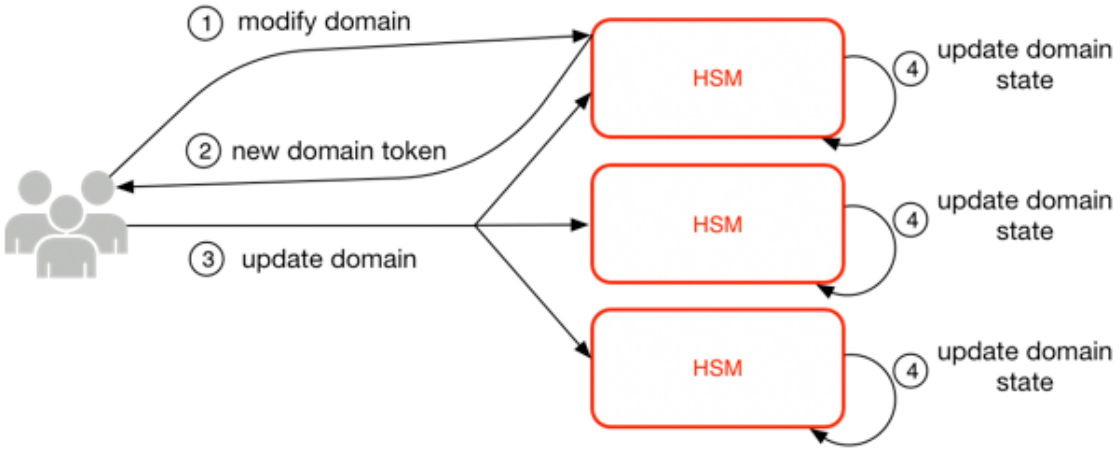
Le jeton de domaine exporté constitue la source essentielle de confiance pour les entités opérant dans le domaine.

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 opérateur peut établir un domaine initial à créer sur un premier 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 chacun des membres du jeton HSMs de domaine nouvellement exporté pour mettre à jour l'état de leur domaine avec le nouveau jeton de domaine.
4. Les éléments HSMs répertoriés dans le nouveau jeton de domaine exporté peuvent authentifier la commande et le jeton de domaine. Ils peuvent également déballer les clés de domaine pour mettre à jour l'état du domaine sur tous HSMs les éléments du domaine.

HSMs ne communiquez pas directement les uns 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 membres HSMs 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 un HSM de la liste des utilisateurs autorisés HSMs 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 ou AWS KMS opérateurs du service et eux HSMs sont sécurisées par le biais de deux mécanismes décrits dans Sessions authentifiées : une méthode de demande signée par quorum et une session authentifiée utilisant un protocole hôte de service HSM.

Les commandes signées par quorum sont conçues de telle sorte qu'aucun opérateur ne puisse modifier les protections de sécurité critiques qu'elles fournissent. HSMs 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 liées au client sont sécurisées dans l'ensemble de l' AWS infrastructure.

Établissement de clé

Pour sécuriser les communications internes, AWS KMS utilise deux méthodes d'établissement 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 intérieure de AWS KMS est le 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 par quorum sont émises par les opérateurs pour. HSMs 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.

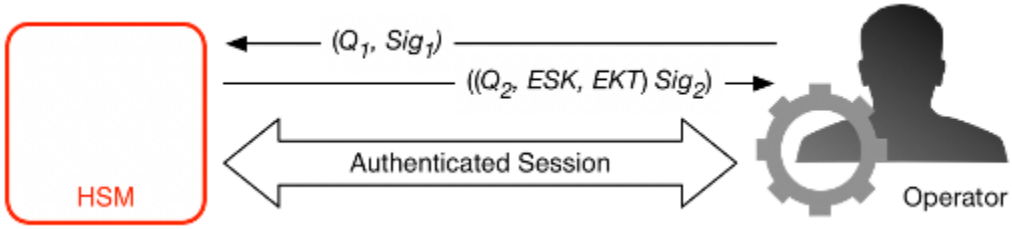
Sessions authentifiées

Vos principales opérations s'exécutent entre les AWS KMS hôtes orientés vers l'extérieur et le HSMs. 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 session entre les hôtes du service et le. HSMs 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 de signature d'identité ($DHos_i$, $QHOS_i$) et une copie authentique des « clés publiques d'identité HSMs ». 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 $(d)_2$, $Q._2$). Il complète ensuite la recommandation pour ECDH-key-exchange les schémas d'établissement de clés par paires utilisant la cryptographie à logarithme discret (révisée) pour former 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 le contenu clé d'une clé KMS d'un HSM d'un HSM d'un autre Région AWS vers un HSM d'un 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 l'autre, les HSMs régions ne peuvent pas communiquer directement, car elles se trouvent dans des réseaux isolés. Au lieu de cela, les messages échangés pendant la réplication entre régions sont délivrés par un service proxy.

Lors de la réplication entre régions, chaque message généré par un AWS KMS HSM est signé cryptographiquement à l'aide d'une clé de signature de réplication. Les clés de signature de réplication (RSKs) sont des clés ECDSA sur la courbe NIST P-384. Chaque région possède au moins une RSK, et la composante publique de chaque RSK est partagée avec toutes les autres régions de la même AWS partition.

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

La conception durable vise à protéger toutes les clés KMS d'une région en cas de perte à grande échelle des clés KMS en ligne HSMs ou de l'ensemble des clés KMS stockées dans notre système de stockage principal. AWS AWS KMS keys avec du matériel de clé importé ne sont pas inclus dans les protections de durabilité accordées aux autres clés KMS. En cas de panne à l'échelle de la région AWS KMS, le matériel clé importé devra peut-être être réimporté dans une clé KMS.

Les données hors ligne HSMs et les informations d'identification permettant d'y accéder sont stockées dans des coffres-forts situés dans des salles sécurisées surveillées situées dans plusieurs emplacements géographiques indépendants. Chaque coffre-fort nécessite au moins un agent AWS de sécurité et un AWS KMS opérateur, issus de deux équipes indépendantes AWS, pour obtenir ces matériaux. L'utilisation de ces matériaux est régie par une politique interne exigeant la présence d'un quorum d' AWS KMS opérateurs.

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

K

Clé de chiffrement des données : clé AES-GCM 256 bits dérivée d'un HBK utilisant le NIST SP800 -108 KDF en mode compteur à l'aide de 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, directeur général - KMS, AWS Cryptographie
- Matthew Campagna, ingénieur de sécurité principal, cryptographie AWS

Bibliographie

Pour plus d'informations à ce sujet AWS Key Management Service HSMs, rendez-vous sur la page de recherche du programme de validation des modules cryptographiques du NIST Computer Security Resource Center et recherchez AWS Key Management Service HSM.

Amazon Web Services, référence générale (version 1.0), « Demande d' AWS API de signature », http://docs.aws.amazon. com/general/latest/gr/signing_aws_api_requests.html.

Amazon Web Services, « Qu'est-ce que c'est » 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 à l'adresse 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 à l'adresse 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 à l'adresse 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 : Norme de cryptographie RSA (RFC 8017), Internet Engineering Task Force (IETF), novembre 2016. https://tools.ietf.org/html/rfc8017.

Recommandation pour les modes de fonctionnement du chiffrement par blocs : Galois/Counter mode (GCM) et GMAC, publication spéciale NIST 800-38D, novembre 2007. Disponible à l'adresse 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 à l'adresse https://nvlpubs.nist. gov/nistpubs/Legacy/SP/nistspecialpublication800-38e.pdf.

Recommandation pour la dérivation de clés à l'aide de fonctions pseudo-aléatoires, publication spéciale NIST 800-108, octobre 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 800-90A du NIST, révision 1, juin 2015, disponible sur https://nvlpubs.nist. gov/nistpubs/SpecialPublications/NIST.SP.800-90AR1.pdf.

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

Utilisation d'algorithmes de cryptographie à courbe elliptique (ECC) dans la syntaxe des messages cryptographiques (CMS), Brown, D., Turner, S., Internet Engineering Task Force, juillet 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 les AWS KMS détails cryptographiques

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 la mise en œuvre de l' AWS KMS `ReplicateKey` opération.	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.