AWS KMS Detail Kriptografi

AWS Key Management Service

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Key Management Service: AWS KMS Detail Kriptografi

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan antara para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan properti dari masing-masing pemilik, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau tidak.

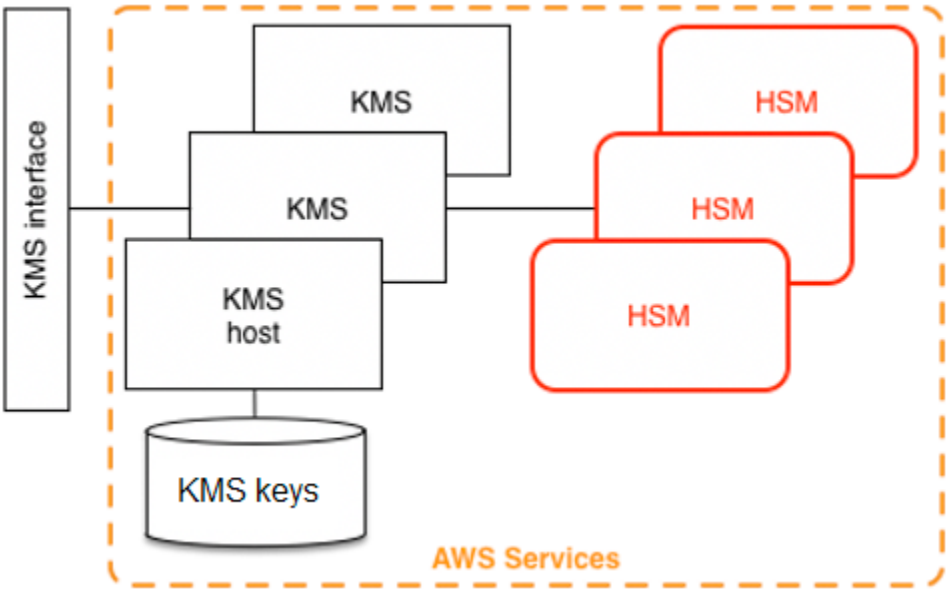
Table of Contents

Pengantar detail kriptografi AWS KMS

AWS Key Management Service (AWS KMS) menyediakan antarmuka web untuk menghasilkan dan mengelola kunci kriptografi dan beroperasi sebagai penyedia layanan kriptografi untuk melindungi data. AWS KMS menawarkan layanan manajemen kunci tradisional yang terintegrasi dengan layanan AWS untuk memberikan tampilan yang konsisten dari kunci pelanggan di AWS, dengan manajemen terpusat dan audit. Laporan resmi ini memberikan penjelasan mendetail tentang operasi kriptografi AWS KMS untuk membantu Anda dalam mengevaluasi fitur yang ditawarkan oleh layanan.

AWS KMS mencakup antarmuka web melalui AWS Management Console, antarmuka baris perintah, dan operasi API RESTful untuk meminta operasi kriptografi armada terdistribusi modul keamanan perangkat keras (HSM) yang divalidasi FIPS 140-2[1]. HSM AWS KMS adalah peralatan kriptografi perangkat keras mandiri multichip yang dirancang untuk menyediakan fungsi kriptografi khusus untuk memenuhi persyaratan keamanan dan skalabilitas AWS KMS. Anda dapat membuat hierarki kriptografi berbasis HSM Anda sendiri di bawah kunci yang Anda kelola sebagai. AWS KMS keys Kunci ini tersedia hanya di HSM dan hanya dalam memori untuk jangka waktu yang diperlukan untuk memproses permintaan kriptografi Anda. Anda dapat membuat beberapa kunci KMS, masing-masing diwakili oleh ID kuncinya. Hanya di bawah peran AWS IAM dan akun yang dikelola oleh setiap pelanggan, kunci KMS pelanggan dapat dibuat, dihapus, atau digunakan untuk mengenkripsi, mendekripsi, menandatangani, atau memverifikasi data. Anda dapat menentukan kontrol akses pada siapa yang dapat mengelola dan/atau menggunakan kunci KMS dengan membuat kebijakan yang dilampirkan ke kunci. Kebijakan tersebut memungkinkan Anda untuk menentukan penggunaan khusus aplikasi untuk kunci Anda bagi setiap operasi API.

Selain itu, sebagian besar AWS layanan mendukung enkripsi data saat istirahat menggunakan kunci KMS. Kemampuan ini memungkinkan pelanggan untuk mengontrol bagaimana dan kapan AWS layanan dapat mengakses data terenkripsi dengan mengontrol bagaimana dan kapan kunci KMS dapat diakses.



AWS KMS adalah layanan bertingkat yang terdiri atas host web AWS KMS dan tingkat HSM. Pengelompokan host bertingkat ini membentuk tumpukan AWS KMS. Semua permintaan untuk AWS KMS harus dibuat melalui protokol Keamanan Lapisan Pengangkutan (TLS) dan berakhir pada host AWS KMS. Host AWS KMS hanya mengizinkan TLS dengan ciphersuite yang menyediakan forward secrecy sempurna. AWS KMS mengautentikasi dan mengotorisasi permintaan Anda menggunakan mekanisme kredensial dan kebijakan yang sama dari AWS Identity and Access Management (IAM) yang tersedia untuk semua operasi API AWS lainnya.

Konsep Basic

Mempelajari beberapa istilah dan konsep dasar akan membantu Anda mendapatkan hasil maksimal dari AWS Key Management Service.

AWS KMS key

> **Note**
>
> AWS KMSmengganti istilah customer master key (CMK) dengan AWS KMS keydan kunci KMS. Konsepnya tidak berubah. Untuk mencegah perubahan yang melanggar, AWS KMS adalah menjaga beberapa variasi dari istilah ini.

Kunci logis yang mewakili bagian atas hierarki kunci Anda. Kunci KMS diberikan Amazon Resource Name (ARN) yang menyertakan pengenal kunci unik, atau ID kunci. AWS KMS keysmemiliki tiga jenis:

- Kunci terkelola pelanggan — Pelanggan membuat dan mengontrol siklus hidup dan kebijakan kunci kunci yang dikelola pelanggan. Semua permintaan yang dibuat terhadap kunci ini dicatat sebagai CloudTrail peristiwa.
- Kunci yang dikelola AWS— AWS membuat dan mengontrol siklus hidup dan kebijakan utamaKunci yang dikelola AWS, yang merupakan sumber daya dalam pelanggan. Akun AWS Pelanggan dapat melihat kebijakan dan CloudTrail acara akses untukKunci yang dikelola AWS, tetapi tidak dapat mengelola aspek apa pun dari kunci ini. Semua permintaan yang dibuat terhadap kunci ini dicatat sebagai CloudTrail peristiwa.
- Kunci milik AWS— Kunci ini dibuat dan secara eksklusif digunakan oleh AWS untuk operasi enkripsi internal di berbagai AWS layanan. Pelanggan tidak memiliki visibilitas ke dalam kebijakan utama atau Kunci milik AWS penggunaan di CloudTrail.

Alias

Nama yang ramah pengguna yang dikaitkan dengan kunci KMS. Alias dapat digunakan secara bergantian dengan ID kunci di banyak Operasi API AWS KMS.

Izin

Kebijakan yang dilampirkan pada kunci KMS yang menentukan izin pada kunci tersebut. Kebijakan default memungkinkan prinsip apa pun yang Anda tentukan, serta memungkinkan Akun AWS untuk menambahkan kebijakan IAM yang mereferensikan kunci.

Izin

Izin yang didelegasikan untuk menggunakan kunci KMS ketika prinsip IAM yang dimaksud atau durasi penggunaan tidak diketahui sejak awal dan oleh karena itu tidak dapat ditambahkan ke kunci atau kebijakan IAM. Salah satu penggunaan hibah adalah untuk menentukan izin tercakup bawah untuk bagaimana layanan dapat menggunakan kunci KMS. AWS Layanan mungkin perlu menggunakan kunci Anda untuk melakukan pekerjaan asinkron atas nama Anda pada data terenkripsi tanpa adanya panggilan API yang ditandatangani langsung dari Anda.

Kunci data

Kunci kriptografi yang dihasilkan pada HSM, dilindungi oleh kunci KMS. AWS KMSmemungkinkan entitas yang berwenang untuk mendapatkan kunci data yang dilindungi oleh kunci KMS. Kunci data tersebut dapat dikembalikan baik sebagai plaintext (tidak terenkripsi) kunci data dan sebagai

kunci data terenkripsi. Kunci data dapat simetris atau asimetris (dengan kedua bagian publik dan privat dikembalikan).

Ciphertext

Output terenkripsi dari AWS KMS, kadang-kadang disebut sebagai ciphertext pelanggan untuk menghilangkan kebingungan. Ciphertext berisi data terenkripsi dengan informasi tambahan yang mengidentifikasi kunci KMS untuk digunakan dalam proses dekripsi. Kunci data terenkripsi adalah salah satu contoh umum ciphertext yang dihasilkan saat menggunakan kunci KMS, tetapi data apa pun di bawah 4 KB dalam ukuran dapat dienkripsi di bawah kunci KMS untuk menghasilkan ciphertext.

Konteks enkripsi

Peta pasangan kunci-nilai informasi tambahan yang terkait dengan AWS KMS—informasi yang dilindungi. AWS KMS menggunakan enkripsi terautentikasi untuk melindungi kunci data. Konteks enkripsi dimasukkan ke dalam AAD enkripsi terautentikasi dalam ciphertext terenkripsi AWS KMS. Informasi konteks ini opsional dan tidak dikembalikan ketika meminta kunci (atau operasi enkripsi). Namun, jika digunakan, nilai konteks ini diperlukan untuk berhasil menyelesaikan operasi dekripsi. Penggunaan dimaksudkan konteks enkripsi adalah untuk memberikan informasi terautentikasi tambahan. Informasi ini dapat membantu Anda menegakkan kebijakan dan disertakan dalam AWS CloudTrail log. Misalnya, Anda dapat menggunakan sepasang kunci-nilai {"key name”:” satellite uplink key"} untuk memberikan nama kunci data. Penggunaan selanjutnya dari kunci menciptakan entri AWS CloudTrail yang menyertakan “nama kunci”: “kunci uplink satelit.” Informasi tambahan ini dapat memberikan konteks yang berguna untuk memahami mengapa kunci KMS yang diberikan digunakan.

Kunci publik

Bila menggunakan cipher asimetris (RSA atau kurva elips), kunci publik adalah “komponen publik” dari pasangan kunci publik-privat. Kunci publik dapat dibagi dan didistribusikan ke entitas yang perlu mengenkripsi data untuk pemilik pasangan kunci publik-privat. Untuk operasi tanda tangan digital, pasangan kunci publik digunakan untuk memverifikasi tanda tangan.

Kunci privat

Bila menggunakan cipher asimetris (RSA atau kurva elips), kunci publik adalah “komponen publik” dari pasangan kunci publik-privat. Kunci privat digunakan untuk mendekripsi data atau membuat tanda tangan digital. Mirip dengan kunci KMS simetris, kunci pribadi dienkripsi dalam HSM. Kunci tersebut didekripsi hanya ke dalam memori jangka pendek HSM dan hanya untuk waktu yang dibutuhkan untuk memproses permintaan kriptografi Anda.

Tujuan desain AWS KMS

AWS KMS dirancang untuk memenuhi persyaratan berikut.

Daya tahan

Daya tahan kunci kriptografi dirancang untuk menyamai ketahanan layanan daya tahan tertinggi di AWS. Kunci kriptografi tunggal dapat mengenkripsi volume besar data Anda yang telah terakumulasi dalam waktu lama.

Dapat Dipercaya

Penggunaan kunci dilindungi oleh kebijakan kontrol akses yang Anda tetapkan dan kelola. Tidak ada mekanisme untuk mengekspor kunci KMS plaintext. Kerahasiaan kunci kriptografi Anda sangat penting. Beberapa karyawan Amazon dengan akses khusus peran untuk kontrol akses berbasis kuorum diperlukan untuk melakukan tindakan administratif pada HSM.

Latensi rendah dan throughput yang tinggi

AWS KMS menyediakan operasi kriptografi pada tingkat latensi dan throughput yang cocok untuk digunakan oleh layanan lain di AWS.

Daerah Independen

AWSmenyediakan Wilayah independen untuk pelanggan yang perlu membatasi akses data di Wilayah yang berbeda. Penggunaan kunci dapat diisolasi dalam fileWilayah AWS.

Sumber nomor acak yang aman

Karena kriptografi yang kuat bergantung pada generasi angka acak yang benar-benar tidak dapat diprediksi, AWS KMS menyediakan sumber angka acak berkualitas tinggi dan tervalidasi.

Audit

AWS KMSmencatat penggunaan dan pengelolaan kunci kriptografi dalam AWS CloudTrail log. Anda dapat menggunakan log AWS CloudTrail untuk memeriksa penggunaan kunci kriptografi Anda, termasuk penggunaan kunci oleh layanan AWS atas nama Anda.

Untuk mencapai tujuan ini, sistem AWS KMS mencakup satu set operator AWS KMS dan operator host layanan (secara kolektif, “operator”) yang mengelola “domain”. Domain adalah satu set server, HSM, dan operator AWS KMS yang ditetapkan per Wilayah. Setiap operator AWS KMS memiliki token perangkat keras yang berisi pasangan kunci pribadi dan publik yang digunakan untuk

mengautentikasi tindakannya. HSM memiliki pasangan kunci pribadi dan publik tambahan untuk membuat kunci enkripsi yang melindungi sinkronisasi status HSM.

Makalah ini menggambarkan bagaimana AWS KMS melindungi kunci dan data lain yang ingin Anda enkripsi. Dalam seluruh dokumen ini, kunci enkripsi atau data yang ingin Anda enkripsi disebut sebagai “rahasia” atau “materi rahasia”.

Fondasi AWS Key Management Service

Topik dalam Bab ini menjelaskan primitif kriptografi AWS Key Management Service dan di mana penggunaannya. Topik tersebut juga memperkenalkan unsur-unsur dasar AWS KMS.

Topik

Primitif kriptografi

AWS KMS menggunakan algoritma kriptografi yang dapat dikonfigurasi sehingga sistem dapat dengan cepat bermigrasi dari satu algoritma atau mode yang disetujui ke algoritma atau mode yang lain. Pengaturan default awal algoritma kriptografi telah dipilih dari algoritma Federal Information Processing Standard (disetujui FIPS) untuk properti dan performa keamanannya.

Entropi dan pembangkitan bilangan acak

AWS KMSGenerasi kunci dilakukan pada AWS KMS HSM. HSMs menerapkan pembangkit bilangan acak hibrida yang menggunakan NIST SP800-90A Deterministic Random Bit Generator (DRBG) CTR_DRBG yang menggunakan AES-256. Pembangkit tersebut ditempatkan dengan generator bit acak nondeterministik dengan 384-bit entropi dan diperbarui dengan entropi tambahan untuk memberikan prediksi resistansi pada setiap panggilan untuk bahan kriptografi.

Operasi kunci simetris (enkripsi saja)

Semua perintah enkripsi kunci simetris yang digunakan dalam HSMs menggunakan Standar Enkripsi Tingkat Lanjut (AES), di Mode Penghitung Galois (GCM) menggunakan kunci 256-bit. Panggilan analog untuk mendekripsi menggunakan fungsi invers.

AES-GCM adalah skema enkripsi terautentikasi. Selain mengenkripsi plaintext untuk menghasilkan ciphertext, AES-CGM mengomputasi tag autentikasi atas ciphertext dan data tambahan yang memerlukan autentikasi diperlukan (data autentikasi tambahan, atau AAD). Tag autentikasi membantu memastikan bahwa data berasal dari sumber yang dimaksudkan dan bahwa ciphertext dan AAD belum dimodifikasi.

Seringkali AWS menghilangkan masuknya AAD dalam deskripsi kami, terutama ketika mengacu pada enkripsi kunci data. Penghilangan ini tersirat oleh teks sekitarnya dalam kasus ini bahwa

struktur yang akan dienkripsi dipartisi antara plaintext yang akan dienkripsi dan cleartext AAAD yang akan dilindungi.

AWS KMSmenyediakan opsi bagi Anda untuk mengimpor materi kunci ke dalam AWS KMS key alih-alih mengandalkan AWS KMS untuk menghasilkan materi utama. Bahan kunci yang diimpor ini dapat dienkripsi menggunakan RSAES-OAEP atau RSAES-PKCS1-v1_5 untuk melindungi kunci selama transportasi ke HSM. AWS KMS Pasangan kunci RSA dihasilkan pada AWS KMS HSM. Materi kunci yang diimpor didekripsi pada AWS KMS HSM dan dienkripsi ulang di bawah AES-GCM sebelum disimpan oleh layanan.

Operasi kunci asimetris (enkripsi, penandatanganan digital dan verifikasi tanda tangan)

AWS KMSmendukung penggunaan operasi kunci asimetris untuk operasi enkripsi dan tanda tangan digital. Operasi kunci asimetris bergantung pada pasangan kunci publik dan pasangan kunci privat yang terkait secara matematis yang dapat Anda gunakan untuk enkripsi dan dekripsi atau penandatanganan dan verifikasi tanda tangan, tetapi tidak keduanya. Kunci privat tidak pernah membiarkan AWS KMS tidak terenkripsi. Anda dapat menggunakan kunci publik dalam AWS KMS dengan memanggil operasi API AWS KMS, atau mengunduh kunci publik dan menggunakannya di luar AWS KMS.

AWS KMS mendukung dua jenis cipher asimetris.

- RSA-OAEP (untuk enkripsi) & RSA-PSS dan RSA-PKCS- #1 -v1_5 (untuk tanda tangan dan verifikasi) — Mendukung panjang kunci RSA (dalam bit): 2048, 3072, dan 4096 untuk persyaratan keamanan yang berbeda.
- Kurva Elips (ECC) — Digunakan secara eksklusif untuk penandatanganan dan verifikasi. Mendukung kurva ECC: NIST P256, P384, P521, SECP 256k1.

Fungsi derivasi kunci

Fungsi derivasi kunci digunakan untuk mendapatkan kunci tambahan dari rahasia awal atau kunci. AWS KMSmenggunakan fungsi derivasi kunci (KDF) untuk mendapatkan kunci per panggilan untuk setiap enkripsi di bawah file. AWS KMS key Semua operasi KDF menggunakan KDF dalam mode counter yang menggunakan HMAC [FIPS197] dengan SHA256 [FIPS180]. Kunci turunan 256-bit digunakan dengan AES-GCM untuk mengenkripsi atau mendekripsi data pelanggan dan kunci.

AWS KMSpenggunaan internal tanda tangan digital

Tanda tangan digital juga digunakan untuk mengautentikasi perintah dan komunikasi antara entitas AWS KMS. Semua entitas layanan memiliki pasangan kunci algoritma tanda tangan digital kurva elips (ECDSA). Entitas tersebut menjalankan ECDSA seperti yang ditentukan dalam Penggunaan Algoritma Elliptic Curve Cryptography (ECC) dalam sintaks pesan kriptografi (CMS) dan X9.62-2005: Kriptografi Kunci Publik untuk Industri Jasa Keuangan: Elliptic Curve Digital Signature Algorithm (ECDSA). Entitas menggunakan algoritma hash aman yang ditentukan dalam Publikasi Standar Pengolahan Informasi Federal, FIPS PUB 180-4, yang dikenal sebagai SHA384. Kunci dihasilkan pada secp384r1 kurva (NIST-P384).

Enkripsi amplop

Konstruksi basic yang digunakan dalam banyak sistem kriptografi adalah enkripsi amplop. Enkripsi amplop menggunakan dua kunci kriptografi atau lebih untuk mengamankan pesan. Biasanya, satu kunci berasal dari kunci statis jangka panjang k, dan kunci lainnya adalah kunci per-pesan, msgKey, yang dihasilkan untuk mengenkripsi pesan. Amplop dibentuk dengan mengenkripsi pesan: ciphertext = Encrypt(MsgKey, message) . Kemudian kunci pesan dienkripsi dengan kunci statis jangka panjang: encKey = Encrypt(k, MsgKey) . Akhirnya, dua nilai (EnCKey, ciphertext) dikemas ke dalam satu struktur, atau pesan terenkripsi amplop.

Penerima, dengan akses ke k, dapat membuka pesan yang diselimuti dengan terlebih dahulu mendekripsi kunci terenkripsi dan kemudian mendekripsi pesan.

AWS KMS menyediakan kemampuan untuk mengelola kunci statis jangka panjang ini dan mengotomatisasi proses enkripsi amplop data Anda.

Selain kemampuan enkripsi yang disediakan dalam AWS KMS layanan, AWSEncryption SDK menyediakan pustaka enkripsi amplop sisi klien. Anda dapat menggunakan perpustakaan ini untuk melindungi data Anda dan kunci enkripsi yang digunakan untuk mengenkripsi data tersebut.

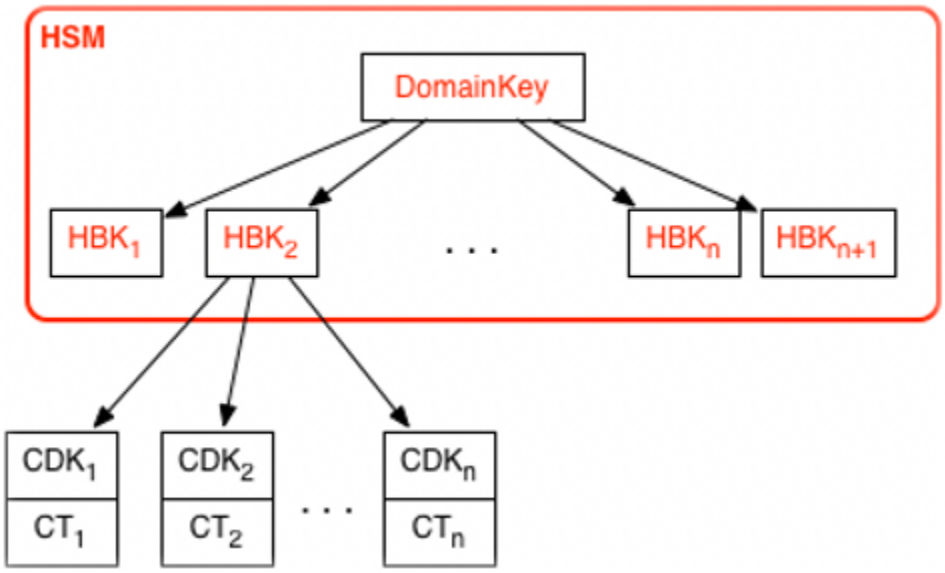
AWS KMS keyhierarki

Hirarki kunci Anda dimulai dengan kunci logis tingkat atas, dan. AWS KMS key Kunci KMS mewakili wadah untuk materi kunci tingkat atas dan didefinisikan secara unik dalam namespace AWS layanan dengan Amazon Resource Name (ARN). ARN menyertakan pengidentifikasi kunci yang dihasilkan secara unik, ID kunci. Kunci KMS dibuat berdasarkan permintaan yang diprakarsai pengguna melalui. AWS KMS Setelah penerimaan, AWS KMS meminta pembuatan kunci dukungan HSM awal (HBK)

untuk ditempatkan ke dalam wadah kunci KMS. HBK dihasilkan pada HSM di domain dan dirancang agar tidak pernah diekspor dari HSM di plaintext. Sebaliknya, HBK diekspor dalam kondisi dienkripsi berdasarkan kunci domain yang dikelola HSM. HBK yang diekspor ini disebut sebagai token kunci yang diekspor (EKT).

EKT diekspor ke penyimpanan latensi rendah yang sangat berdaya tahan. Misalnya, Anda menerima ARN ke kunci KMS logis. Ini mewakili bagian atas hierarki kunci, atau konteks kriptografi, untuk Anda. Anda dapat membuat beberapa kunci KMS dalam akun Anda dan menetapkan kebijakan pada kunci KMS Anda seperti sumber daya AWS bernama lainnya.

Dalam hierarki kunci KMS tertentu, HBK dapat dianggap sebagai versi kunci KMS. Saat Anda ingin memutar tombol KMSAWS KMS, HBK baru dibuat dan dikaitkan dengan kunci KMS sebagai HBK aktif untuk kunci KMS. HBK yang lebih lama disimpan dan dapat digunakan untuk mendekripsi dan memverifikasi data yang sebelumnya dilindungi. Tetapi hanya kunci kriptografi aktif yang bisa digunakan untuk melindungi informasi baru.



Anda dapat membuat permintaan melalui AWS KMS untuk menggunakan kunci KMS Anda untuk secara langsung melindungi informasi atau meminta kunci tambahan yang dihasilkan HSM yang dilindungi di bawah kunci KMS Anda. Kunci ini disebut kunci data pelanggan, atau CDK. CDK dapat dikembalikan dalam kondisi dienkripsi sebagai ciphertext (CT), di plaintext, atau keduanya. Semua objek yang dienkripsi di bawah kunci KMS (baik data yang disediakan pelanggan atau kunci yang dihasilkan HSM) dapat didekripsi hanya pada HSM melalui panggilan melalui. AWS KMS

Ciphertext yang dikembalikan, atau muatan yang didekripsi, tidak akan disimpan dalam AWS KMS. Informasi dikembalikan kepada Anda melalui koneksi TLS Anda ke AWS KMS. Ini juga berlaku untuk panggilan yang dibuat oleh AWS atas nama Anda.

Hierarki kunci dan properti kunci tertentu dicantumkan dalam tabel berikut.

Kunci	Deskripsi	Siklus hidup
Kunci domain	Kunci AES-GCM 256-bit hanya dalam memori HSM yang digunakan untuk membungkus versi kunci KMS, kunci pendukung HSM.	Dirotasi setiap hari[1]
Kunci dukungan HSM	Kunci simetris 256-bit atau RSA atau kunci pribadi kurva eliptik, digunakan untuk melindungi data dan kunci pelanggan serta disimpan dalam kondisi dienkripsi berdasarkan kunci domain. Satu atau lebih kunci dukungan HSM terdiri dari kunci KMS, diwakili oleh KeyID.	Dirotasi per tahun[2] (konfigurasi opsional)
Kunci enkripsi turunan	Kunci AES-GCM 256-bit hanya dalam memori HSM yang digunakan untuk mengenkripsi data dan kunci pelanggan. Berasal dari HBK untuk setiap enkripsi.	Digunakan sekali tiap enkripsi dan diregenerasi pada dekripsi
Kunci data pelanggan	Kunci simetris atau asimetris yang ditetapkan pelanggan diekspor dari HSM di plaintext dan ciphertext. Dienkripsi berdasarkan kunci cadangan HSM dan dikembalikan ke pengguna yang diotorisasi melalui saluran TLS.	Rotasi dan penggunaan dikendalikan oleh aplikasi

[1] AWS KMS mungkin dari waktu ke waktu melonggarkan rotasi kunci domain paling tidak ke skema waktu mingguan untuk memungkinkan administrasi domain dan tugas konfigurasi.

[2] Default yang Kunci yang dikelola AWS dibuat dan dikelola oleh AWS KMS atas nama Anda secara otomatis diputar setiap tahun.

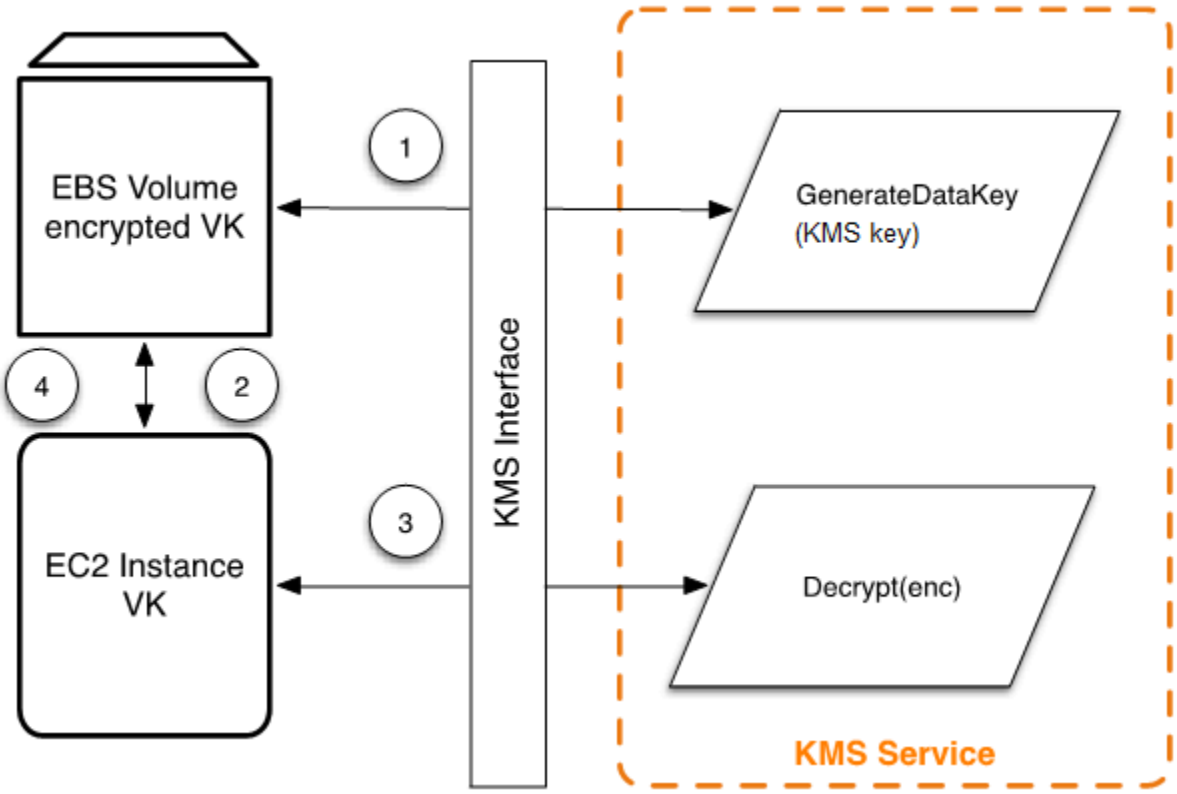
AWS KMS kasus penggunaan

Kasus penggunaan dapat membantu Anda mendapatkan hasil maksimal AWS Key Management Service. Yang pertama menunjukkan bagaimana AWS KMS melakukan enkripsi sisi server dengan volume Amazon AWS KMS keys Elastic Block Store (Amazon EBS). Yang kedua adalah aplikasi sisi klien yang menunjukkan bagaimana Anda dapat menggunakan enkripsi amplop untuk melindungi konten. AWS KMS

Topik

Enkripsi volume Amazon EBS

Amazon EBS menawarkan kemampuan enkripsi volume. Setiap volume dienkripsi menggunakan AES-256-XTS. Ini memerlukan dua kunci volume 256-bit, yang dapat Anda anggap sebagai satu kunci volume 512-bit. Tombol volume dienkripsi di bawah kunci KMS di akun Anda. Agar Amazon EBS mengenkripsi volume untuk Anda, ia harus memiliki akses untuk menghasilkan tombol volume (VK) di bawah kunci KMS di akun. Anda melakukan ini dengan memberikan hibah untuk Amazon EBS ke kunci KMS untuk membuat kunci data dan mengenkripsi dan mendekripsi kunci volume ini. Sekarang Amazon EBS menggunakan AWS KMS dengan kunci KMS untuk menghasilkan kunci volume AWS KMS terenkripsi.



Alur kerja berikut mengenkripsi data yang sedang ditulis ke volume Amazon EBS:

1. Amazon EBS memperoleh kunci volume terenkripsi di bawah kunci KMS AWS KMS melalui sesi TLS dan menyimpan kunci terenkripsi dengan metadata volume.
2. Ketika volume Amazon EBS terpasang, kunci volume terenkripsi akan diambil.
3. Panggilan ke AWS KMS over TLS dibuat untuk mendekripsi kunci volume terenkripsi. AWS KMS mengidentifikasi kunci KMS dan membuat permintaan internal ke HSM di armada untuk mendekripsi kunci volume terenkripsi. AWS KMS kemudian mengembalikan kunci volume kembali ke host Amazon Elastic Compute Cloud (Amazon EC2) yang berisi instance Anda selama sesi TLS.
4. Tombol volume digunakan untuk mengenkripsi dan mendekripsi semua data yang masuk ke dan dari volume Amazon EBS yang dipasang. Amazon EBS mempertahankan kunci volume terenkripsi untuk digunakan nanti jika tombol volume dalam memori tidak lagi tersedia.

Untuk informasi selengkapnya tentang mengenkripsi volume Amazon EBS dengan kunci KMS, lihat Cara Amazon Elastic Block Store menggunakan AWS KMS dalam AWS Key Management Service

Panduan Pengembang dan enkripsi Amazon EBS di Panduan Pengguna Amazon EC2 dan Panduan Pengguna Amazon EC2.

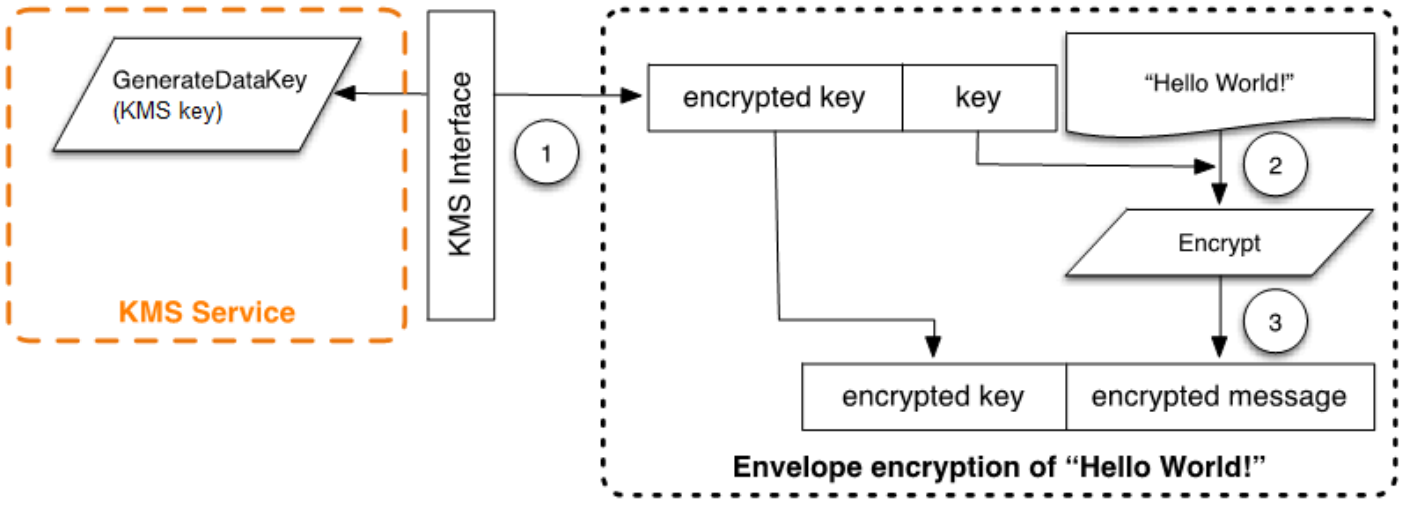
Enkripsi di sisi klien

AWS Encryption SDKTermasuk operasi API untuk melakukan enkripsi amplop menggunakan kunci KMS. Untuk rekomendasi lengkap dan detail penggunaan, lihat dokumentasi terkait. Aplikasi klien dapat menggunakan AWS Encryption SDK untuk melakukan enkripsi amplop menggunakan 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);
```

Aplikasi klien dapat menjalankan langkah-langkah berikut:

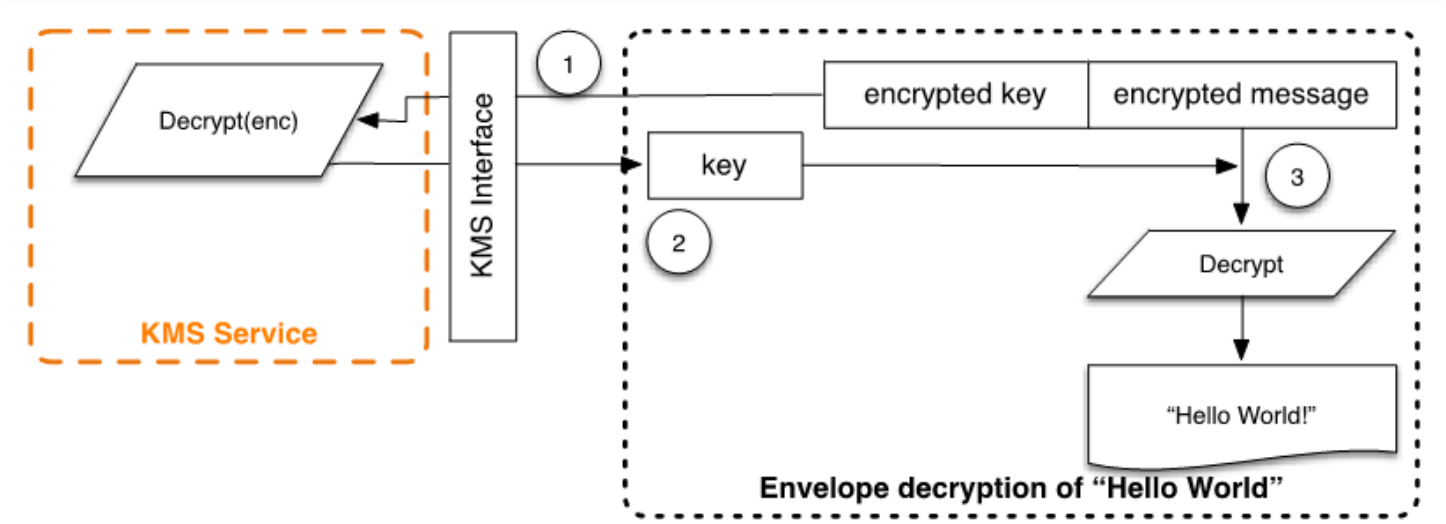
1. Permintaan dibuat di bawah kunci KMS untuk kunci data baru. Kunci data terenkripsi dan versi plaintext dari kunci data akan dikembalikan.
2. Di dalam AWS Encryption SDK, kunci data plaintext digunakan untuk mengenkripsi pesan. Kunci data plaintext kemudian dihapus dari memori.
3. Kunci data terenkripsi dan pesan terenkripsi digabungkan menjadi array byte ciphertext tunggal.



Pesan terenkripsi envelope dapat didekripsi menggunakan fungsionalitas dekripsi untuk mendapatkan pesan terenkripsi awal.

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

1. AWS Encryption SDK Mem-parsing pesan terenkripsi amplop untuk mendapatkan kunci data terenkripsi dan membuat permintaan untuk mendekripsi kunci data. AWS KMS
2. AWS Encryption SDK Menerima kunci data plaintext dari. AWS KMS
3. Kunci data kemudian digunakan untuk mendekripsi pesan, mengembalikan plaintext awal.

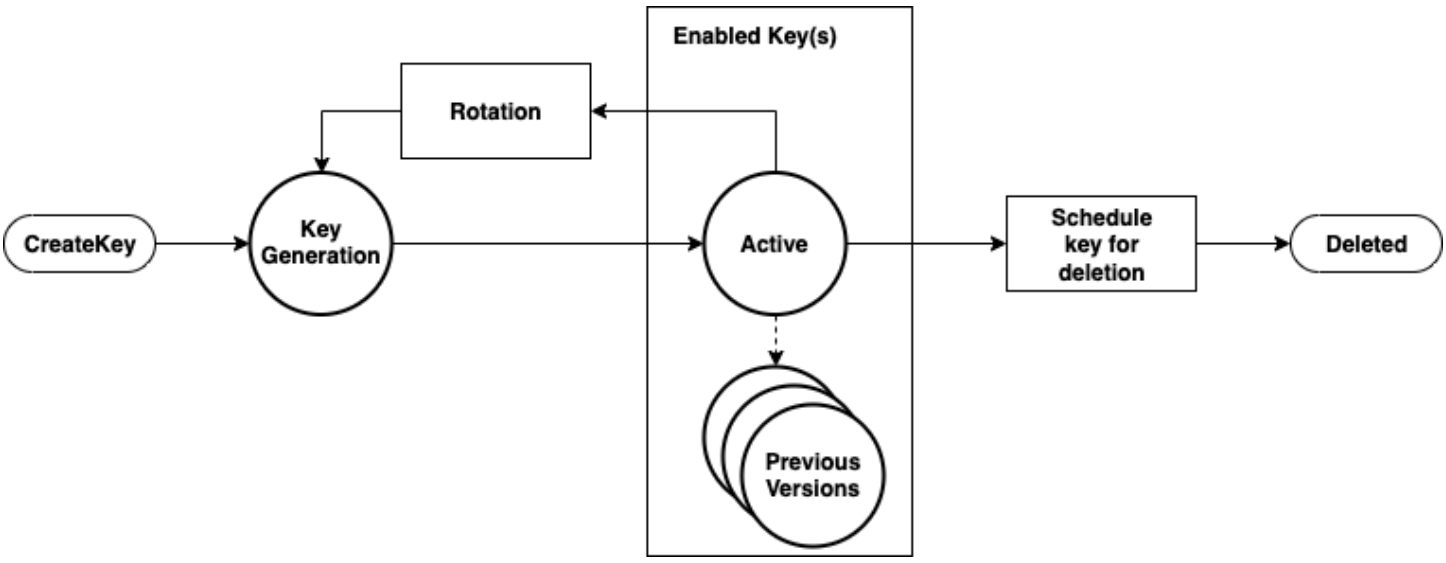


Bekerja dengan AWS KMS keys

An AWS KMS key mengacu pada kunci logis yang mungkin merujuk ke satu atau lebih kunci dukungan modul keamanan perangkat keras (HSM) (HBK). Topik ini menjelaskan cara membuat kunci KMS, mengimpor materi kunci, dan cara mengaktifkan, menonaktifkan, memutar, dan menghapus kunci KMS.

> **Note**
>
> AWS KMSmengganti istilah customer master key (CMK) dengan AWS KMS keydan kunci KMS. Konsepnya tidak berubah. Untuk mencegah perubahan yang melanggar, AWS KMS adalah menjaga beberapa variasi istilah ini.

Bab ini membahas siklus hidup kunci KMS dari pembuatan hingga penghapusan, seperti yang ditunjukkan pada gambar berikut.



Topik

- Memanggil CreateKey
- Mengimpor materi kunci
- Mengaktifkan dan menonaktifkan kunci
- Menghapus kunci
- Memutar bahan kunci

Memanggil CreateKey

An AWS KMS key dihasilkan sebagai hasil dari panggilan ke panggilan `CreateKey`API.

Berikut ini adalah bagian dari sintaks `CreateKey` permintaan.

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

Permintaan menerima data berikut dalam format JSON.

Deskripsi

Deskripsi (Opsional) kunci. Kami menyarankan Anda memilih deskripsi yang membantu Anda memutuskan apakah kunci sesuai untuk tugas.

KeySpec

Menentukan jenis kunci KMS untuk membuat. Nilai default, SYMMETRIC_DEFAULT, menciptakan kunci KMS enkripsi simetris. Parameter ini opsional untuk kunci enkripsi simetris, dan diperlukan untuk semua spesifikasi kunci lainnya.

KeyUsage

Menentukan penggunaan kunci. Nilai yang valid adalah `ENCRYPT_DECRYPT`, `SIGN_VERIFY`, atau `GENERATE_VERIFY_MAC`. Nilai default-nya adalah `ENCRYPT_DECRYPT`. Parameter ini opsional untuk kunci enkripsi simetris, dan diperlukan untuk semua spesifikasi kunci lainnya.

Asal

(Opsional) Menentukan sumber bahan kunci untuk kunci KMS. Nilai defaultnya adalah`AWS_KMS`, yang menunjukkan bahwa AWS KMS menghasilkan dan mengelola materi kunci untuk kunci KMS. Nilai valid lainnya termasuk`EXTERNAL`, yang mewakili kunci KMS yang dibuat tanpa bahan kunci untuk materi kunci yang diimpor, dan `AWS_CLOUDHSM` yang membuat kunci KMS di penyimpanan kunci khusus yang didukung oleh AWS CloudHSM cluster yang Anda kontrol.

Kebijakan

> Kebijakan (Opsional) untuk melampirkan ke kunci. Jika kebijakan dihilangkan, kunci dibuat dengan kebijakan default (berikut) yang memungkinkan akun root dan prinsipal IAM dengan izin untuk mengelolanya. AWS KMS

Untuk detail tentang kebijakan, lihat Kebijakan kunci di AWS KMS dan Kebijakan kunci default di Panduan AWS Key Management Service Pengembang.

`CreateKey`Permintaan mengembalikan respons yang menyertakan ARN kunci.

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

Jika ya`AWS_KMS`, setelah ARN `Origin` dibuat, permintaan ke AWS KMS HSM dibuat melalui sesi yang diautentikasi untuk menyediakan kunci dukungan modul keamanan perangkat keras (HSM) (HBK). HBK adalah kunci 256-bit yang dikaitkan dengan ID kunci ini dari kunci KMS. Hal ini dapat dihasilkan hanya pada HSM dan dirancang tidak pernah diekspor di luar batas HSM di cleartext. HBK dienkripsi di bawah kunci domain saat ini, DK_0. HBK terenkripsi ini disebut sebagai token kunci terenkripsi (EKT). Meskipun HSM dapat dikonfigurasi untuk menggunakan berbagai metode pembungkus kunci, implementasi saat ini menggunakan AES-256 dalam Galois Counter Mode (GCM), skema enkripsi yang diautentikasi. Mode enkripsi yang diautentikasi ini memungkinkan kami untuk melindungi beberapa metadata token kunci yang diekspor cleartext.

Ini secara gaya direpresentasikan sebagai:

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

Dua bentuk perlindungan mendasar disediakan untuk kunci KMS Anda dan HBK berikutnya: kebijakan otorisasi yang ditetapkan pada kunci KMS Anda dan perlindungan kriptografi pada HBK terkait Anda. Bagian yang tersisa menggambarkan perlindungan kriptografi dan keamanan fungsi manajemen di AWS KMS.

Selain ARN, Anda dapat membuat nama yang ramah pengguna dan mengaitkannya dengan kunci KMS dengan membuat alias untuk kunci tersebut. Setelah alias dikaitkan dengan kunci KMS, alias dapat digunakan untuk mengidentifikasi kunci KMS dalam operasi kriptografi. Untuk informasi selengkapnya, lihat Menggunakan alias di Panduan AWS Key Management Service Pengembang.

Beberapa tingkat otorisasi mengelilingi penggunaan kunci KMS. AWS KMSmemungkinkan kebijakan otorisasi terpisah antara konten terenkripsi dan kunci KMS. Sebagai contoh, sebuah Amazon

Simple Storage Service (Amazon S3) terenkripsi amplop AWS KMS mewarisi kebijakan pada bucket Amazon S3. Namun, akses ke kunci enkripsi yang diperlukan ditentukan oleh kebijakan akses pada kunci KMS. Untuk informasi tentang otorisasi kunci KMS, lihat Otentikasi dan kontrol akses untuk AWS KMS di Panduan PengembangAWS Key Management Service.

Mengimpor materi kunci

AWS KMS menyediakan mekanisme untuk mengimpor bahan kriptografi yang digunakan untuk HBK. Seperti dijelaskan dalamMemanggil CreateKey, ketika CreateKey perintah digunakan dengan `Origin` set to`EXTERNAL`, kunci KMS logis dibuat yang tidak mengandung HBK yang mendasarinya. Materi kriptografi harus diimpor menggunakan panggilan `ImportKeyMaterial` API. Anda dapat menggunakan fitur ini untuk mengontrol penciptaan kunci dan daya tahan bahan kriptografi. Jika Anda menggunakan fitur ini, kami sarankan Anda berhati-hati dalam penanganan dan daya tahan tombol ini di lingkungan Anda. Untuk detail lengkap dan rekomendasi untuk mengimpor materi utama, lihat Mengimpor materi utama di Panduan AWS Key Management Service Pengembang.

Memanggil ImportKeyMaterial

Parameter permintaan `ImportKeyMaterial` mengimpor bahan kriptografi yang diperlukan untuk HBK tersebut. Materi kriptografi harus menjadi kunci simetris 256-bit. Materi tersebut harus dienkripsi menggunakan algoritma yang ditentukan dalam `WrappingAlgorithm` berdasarkan kunci publik yang dikembalikan dari permintaan `GetParametersForImport` terbaru.

`ImportKeyMaterial`Permintaan mengambil argumen berikut.

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

EncryptedKeyMaterial

Materi kunci yang diimpor yang dienkripsi dengan kunci publik dikembalikan dalam `GetParametersForImport` permintaan menggunakan algoritma pembungkus yang ditentukan dalam permintaan tersebut.

ExpirationModel

Menentukan apakah bahan kunci kedaluwarsa. Ketika nilai ini adalah `KEY_MATERIAL_EXPIRES`, parameter `ValidTo` harus berisi tanggal kedaluwarsa. Ketika nilai ini adalah `KEY_MATERIAL_DOES_NOT_EXPIRE`, jangan masukkan parameter `ValidTo`. Nilai yang valid adalah `"KEY_MATERIAL_EXPIRES"` dan `"KEY_MATERIAL_DOES_NOT_EXPIRE"`.

ImportToken

Token impor dikembalikan oleh `GetParametersForImport` permintaan yang sama yang menyediakan kunci publik.

KeyId

Kunci KMS yang akan dikaitkan dengan bahan kunci impor. Kunci KMS harus`EXTERNAL`. `Origin`

Anda dapat menghapus dan mengimpor ulang materi kunci impor yang sama ke dalam kunci KMS yang ditentukan, tetapi Anda tidak dapat mengimpor atau mengaitkan kunci KMS materi kunci lainnya.

ValidTo

(Opsional) Waktu kedaluawarsa bahan kunci yang diimpor. Ketika materi kunci kedaluwarsa, AWS KMS menghapus materi kunci dan kunci KMS menjadi tidak dapat digunakan. Parameter ini diperlukan ketika nilai `ExpirationModel` adalah`KEY_MATERIAL_EXPIRES`. Kalau tidak, itu tidak valid.

Ketika permintaan berhasil, kunci KMS tersedia untuk digunakan AWS KMS sampai tanggal kedaluwarsa yang ditentukan, jika ada yang disediakan. Setelah bahan kunci yang diimpor kedaluwarsa, EKT dihapus dari lapisan AWS KMS penyimpanan.

Mengaktifkan dan menonaktifkan kunci

Menonaktifkan kunci KMS mencegah kunci digunakan dalam operasi kriptografi. Ini menangguhkan kemampuan untuk menggunakan semua HBK yang terkait dengan kunci KMS. Mengaktifkan mengembalikan penggunaan HBK dan kunci KMS. Aktifkan dan Nonaktifkan adalah permintaan sederhana yang hanya mengambil ID kunci atau kunci ARN dari kunci KMS.

Menghapus kunci

Pengguna yang berwenang dapat menggunakan `ScheduleKeyDeletion`API untuk menjadwalkan penghapusan kunci KMS dan semua HBK terkait. Ini adalah operasi yang merusak secara inheren, dan Anda harus berhati-hati saat menghapus kunci dari. AWS KMS AWS KMSmemberlakukan waktu tunggu minimal tujuh hari saat menghapus kunci KMS. Selama masa tunggu kunci ditempatkan dalam keadaan nonaktif dengan keadaan kunci Menunda Penghapusan. Semua panggilan untuk menggunakan kunci untuk operasi kriptografi akan gagal. ScheduleKeyDeletion mengambil argumen berikut.

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

KeyId

Pengidentifikasi unik untuk kunci KMS untuk dihapus. Untuk menentukan nilai ini, gunakan ID kunci unik atau ARN kunci dari kunci KMS.

PendingWindowInDays

(Opsional) Masa tunggu, dalam jumlah hari. Nilai ini bersifat opsional. Kisarannya adalah 7-30 hari dan nilai default adalah 30 hari. Setelah masa tunggu berakhir, AWS KMS hapus kunci KMS dan semua HBK terkait.

Memutar bahan kunci

Pengguna yang berwenang dapat mengaktifkan rotasi tahunan otomatis kunci KMS yang dikelola pelanggan mereka. Kunci yang dikelola AWSselalu diputar setiap tahun.

Ketika kunci KMS diputar, HBK baru dibuat dan ditandai sebagai versi bahan kunci saat ini untuk semua permintaan enkripsi baru. Semua versi HBK sebelumnya tetap tersedia untuk digunakan selamanya untuk mendekripsi ciphertext apa pun yang dienkripsi menggunakan versi HBK ini. Karena AWS KMS tidak menyimpan ciphertext apa pun yang dienkripsi di bawah kunci KMS, ciphertext yang dienkripsi di bawah HBK yang lebih lama dan diputar mengharuskan HBK untuk mendekripsi. Anda dapat menggunakan `ReEncrypt`API untuk mengenkripsi ulang ciphertext apa pun di bawah HBK baru untuk kunci KMS atau di bawah kunci KMS yang berbeda tanpa mengekspos plaintext.

Untuk informasi tentang mengaktifkan dan menonaktifkan rotasi kunci, lihat Memutar kunci AWS KMS di Panduan Pengembang. AWS Key Management Service

Operasi data pelanggan

Setelah Anda membuat kunci KMS, itu dapat digunakan untuk melakukan operasi kriptografi. Setiap kali data dienkripsi di bawah kunci KMS, objek yang dihasilkan adalah ciphertext pelanggan. Ciphertext berisi dua bagian: bagian header terenkripsi (atau cleartext), dilindungi oleh skema enkripsi yang dikonfirmasi sebagai data dikonfirmasi tambahan, dan bagian terenkripsi. Bagian cleartext termasuk pengenal HBK (HBKID). Bidang ini membantu memastikan bahwa AWS KMS dapat mendekripsi objek di masa depan.

Topik

Menghasilkan kunci data

Pengguna yang berwenang dapat menggunakan GenerateDataKey API (dan API terkait) untuk meminta jenis kunci data tertentu atau kunci acak dengan panjang arbitrer. Topik ini memberikan tampilan yang disederhanakan dari operasi API ini. Untuk detailnya, lihat GenerateDataKey API di Referensi AWS Key Management Service API.

- GenerateDataKey
- GenerateDataKeyWithoutPlaintext
- GenerateDataKeyPair
- GenerateDataKeyPairWithoutPlaintext

Berikut hasil sintaks permintaan `GenerateDataKey`.

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

```
}
```

Permintaan menerima data berikut dalam format JSON.

KeyId

Pengenal kunci kunci yang digunakan untuk mengenkripsi kunci data. Nilai ini harus mengidentifikasi kunci KMS enkripsi simetris.

Parameter ini diperlukan.

NumberOfBytes

Integer yang berisi jumlah byte yang dihasilkan. Parameter ini diperlukan.

Pemanggil harus menyediakan `KeySpec` atau `NumberOfBytes`, tapi tidak keduanya.

EncryptionContext

(Opsional) Pasangan nama-nilai yang berisi data tambahan untuk diautentikasi selama proses enkripsi dan dekripsi yang menggunakan kunci.

GrantTokens

(Opsional) Daftar token bantuan yang mewakili grants yang memberikan izin untuk menghasilkan atau menggunakan kunci. Untuk informasi selengkapnya tentang hibah dan token hibah, lihat Otentikasi dan kontrol akses untuk AWS KMS di Panduan AWS Key Management ServicePengembang.

Setelah mengautentikasi perintah,AWS KMS, memperoleh EKT aktif saat ini yang terkait dengan kunci KMS. Melewati EKT bersama dengan permintaan yang Anda berikan dan konteks enkripsi ke HSM selama sesi dilindungi antara host AWS KMS dan HSM di domain.

Program ini melakukan hal-hal berikut:

1. Menghasilkan materi rahasia yang diminta dan menahannya dalam memori yang mudah berubah.
2. Mendekripsi EKT yang cocok dengan ID kunci dari kunci KMS yang ditentukan dalam permintaan untuk mendapatkan HBK = Decrypt aktif (DK, EKT). ¡
3. Menghasilkan nonce acak N.
4. Membuat kunci enkripsi AES-GCM 256-bit K dari HBK dan N.
5. Mengenkripsi materi rahasia ciphertext = Enkripsi (K, konteks, rahasia).

`GenerateDataKey`mengembalikan materi rahasia plaintext dan ciphertext kepada Anda melalui saluran aman antara AWS KMS host dan HSM. AWS KMSkemudian mengirimkannya kepada Anda melalui sesi TLS. AWS KMStidak mempertahankan plaintext atau ciphertext. Tanpa memiliki ciphertext, konteks enkripsi, dan otorisasi untuk menggunakan kunci KMS, rahasia yang mendasarinya tidak dapat dikembalikan.

Berikut ini adalah sintaks responsnya.

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

Pengelolaan kunci data diserahkan kepada Anda sebagai developer aplikasi. Untuk enkripsi sisi klien praktik terbaik dengan kunci AWS KMS data (tetapi bukan pasangan kunci data), Anda dapat menggunakan file. AWS Encryption SDK

Kunci data dapat diputar pada frekuensi apa pun. Selanjutnya, kunci data dapat dienkripsi ulang di bawah kunci KMS yang berbeda atau kunci KMS yang diputar menggunakan operasi API. `ReEncrypt` Untuk detailnya, lihat ReEncryptdi Referensi AWS Key Management Service API.

Enkripsi

Fungsi dasar dari AWS KMS adalah untuk mengenkripsi objek di bawah kunci KMS. AWS KMS dirancang untuk menyediakan operasi kriptografi latensi rendah pada HSMS. Dengan demikian, ada batas 4 KB pada jumlah plaintext yang dapat dienkripsi dalam panggilan langsung ke fungsi mengenkripsi. AWS Encryption SDKDapat digunakan untuk mengenkripsi pesan yang lebih besar. AWS KMS, setelah mengotentikasi perintah, memperoleh EKT aktif saat ini yang berkaitan dengan kunci KMS. Ini melewati EKT, bersama dengan konteks plaintext dan enkripsi, ke HSM yang tersedia di Wilayah. Ini dikirim melalui sesi yang diautentikasi antara host AWS KMS dan HSM di domain.

HSM menjalankan berikut ini:

1. Mendekripsi EKT untuk mendapatkan HBK = Dekripsi (DK_i, EKT) .
2. Menghasilkan nonce acak N.
3. Membuat kunci enkripsi AES-GCM 256-bit K dari HBK dan N.
4. Mengenkripsi plaintext ciphertext = Enkripsi (K, konteks, plaintext).

Nilai ciphertext dikembalikan kepada Anda dan data plaintext atau ciphertext tidak dipertahankan di mana saja di infrastruktur AWS. Tanpa memiliki ciphertext dan konteks enkripsi, dan otorisasi untuk menggunakan kunci KMS, plaintext yang mendasarinya tidak dapat dikembalikan.

Dekripsi

Panggilan ke AWS KMS untuk mendekripsi nilai ciphertext menerima ciphertext nilai terenkripsi dan konteks enkripsi. AWS KMS mengautentikasi panggilan menggunakan permintaan yang ditandatangani dengan tanda tangan AWS versi 4 dan mengekstrak HBKID untuk kunci pembungkus dari ciphertext. HBKID digunakan untuk mendapatkan EKT yang diperlukan untuk mendekripsi ciphertext, ID kunci, dan kebijakan untuk ID kunci. Permintaan diizinkan berdasarkan kebijakan kunci, bantuan yang mungkin ada, dan kebijakan IAM terkait yang merujuk pada ID kunci. Parameter `Decrypt` berfungsi analog dengan fungsi enkripsi.

Berikut hasil sintaks permintaan `Decrypt`.

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

Berikut ini adalah parameter permintaan.

CiphertextBlob

Ciphertext termasuk metadata.

EncryptionContext

(Opsional) Konteks enkripsi. Jika ditentukan dalam fungsi `Encrypt`, hal ini harus ditentukan di sini atau operasi dekripsi gagal. Untuk informasi lebih lanjut, lihat Konteks enkripsi di Panduan PengembangAWS Key Management Service.

GrantTokens

(Opsional) Daftar token bantuan yang mewakili bantuan yang memberikan izin untuk melakukan dekripsi.

Parameter ciphertext dan EKT dikirim, bersama dengan konteks enkripsi, selama sesi autentikasi untuk HSM untuk dekripsi.

HSM menjalankan berikut ini:

1. Mendekripsi EKT untuk mendapatkan HBK = Dekripsi (DK_i, EKT) .
2. Mengekstrak nonce N dari struktur ciphertext.
3. Membuat kunci enkripsi AES-GCM 256-bit K dari HBK dan N.
4. Mendekripsi ciphertext untuk mendapatkan plaintext = Dekripsi (K, konteks, ciphertext).

ID kunci yang dihasilkan dan plaintext dikembalikan ke host AWS KMS selama sesi aman dan kemudian kembali ke aplikasi pelanggan panggilan melalui sambungan TLS.

Berikut ini adalah sintaks responsnya.

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

Jika aplikasi panggilan ingin memastikan keaslian plaintext, harus diverifikasi bahwa ID kunci yang dikembalikan adalah ID kunci yang diharapkan.

Mengenkripsi ulang objek terenkripsi

Ciphertext pelanggan yang sudah ada yang dienkripsi di bawah satu kunci KMS dapat dienkripsi ulang ke kunci KMS lain melalui perintah reencrypt. Enkripsi ulang data enkripsi di sisi server dengan kunci KMS baru tanpa mengekspos plaintext kunci di sisi klien. Data pertama kali didekripsi dan kemudian dienkripsi.

Berikut hasil sintaks permintaan.

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

Permintaan menerima data berikut dalam format JSON.

CiphertextBlob

Ciphertext data untuk dienkripsi ulang.

DestinationEncryptionContext

(Opsional) Konteks enkripsi yang akan digunakan ketika data dienkripsi ulang.

DestinationKeyId

Pengenal kunci kunci yang digunakan untuk mengenkripsi ulang data.

GrantTokens

(Opsional) Daftar token bantuan yang mewakili bantuan yang memberikan izin untuk melakukan dekripsi.

SourceKeyId

(Opsional) Pengidentifikasi kunci kunci yang digunakan untuk mendekripsi data.

SourceEncryptionContext

(Opsional) Konteks enkripsi yang digunakan untuk mengenkripsi dan mendekripsi data yang ditentukan dalam parameter `CiphertextBlob`.

Proses ini menggabungkan operasi dekripsi dan enkripsi dari deskripsi sebelumnya: Ciphertext pelanggan didekripsi di bawah HBK awal yang direferensikan oleh ciphertext pelanggan ke HBK saat ini di bawah kunci KMS yang dimaksud. Ketika kunci KMS yang digunakan dalam perintah ini sama, perintah ini memindahkan ciphertext pelanggan dari versi lama HBK ke versi terbaru HBK.

Berikut ini adalah sintaks responsnya.

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

Jika aplikasi panggilan ingin memastikan keaslian plaintext yang mendasarinya, itu harus memverifikasi yang SourceKeyId dikembalikan adalah yang diharapkan.

Operasi AWS KMS internal

AWS KMS internal diperlukan untuk menskalakan dan mengamankan HSM untuk layanan manajemen kunci yang didistribusikan secara global.

Topik

Domain dan status domain

Kumpulan kooperatif AWS KMS entitas internal tepercaya dalam suatu domain Wilayah AWS disebut sebagai domain. Domain mencakup seperangkat entitas tepercaya, seperangkat aturan, dan seperangkat kunci rahasia, yang disebut kunci domain. Kunci domain dibagi di antara HSMs yang merupakan anggota domain. Status domain terdiri atas bidang berikut.

Nama

Nama domain untuk mengidentifikasi domain ini.

Anggota

Daftar HSM yang merupakan anggota domain, termasuk kunci penandatanganan publik dan kunci perjanjian publik mereka.

Operator

Daftar entitas, kunci penandatanganan publik, dan peran (operator atau host layanan AWS KMS) yang mewakili operator layanan ini.

Aturan

Daftar aturan kuorum untuk setiap perintah yang harus dipenuhi untuk menjalankan perintah pada HSM.

Kunci domain

Daftar kunci domain (kunci simetris) saat ini digunakan dalam domain.

Status domain penuh tersedia hanya pada HSM. Status domain disinkronkan antara anggota domain HSM sebagai token domain yang diekspor.

Kunci domain

Semua HSM di domain berbagi satu set kunci domain, $\{DK_r\}$. Kunci ini dibagi melalui rutin ekspor status domain. Status domain yang diekspor dapat diimpor ke setiap HSM yang merupakan anggota domain.

Set kunci domain, $\{DK_r\}$, selalu menyertakan satu kunci domain aktif, dan beberapa kunci domain yang dinonaktifkan. Kunci domain dirotasi setiap hari untuk memastikan bahwa AWS mematuhi Rekomendasi untuk Manajemen Kunci - Bagian 1. Selama rotasi kunci domain, semua kunci KMS yang ada yang dienkripsi di bawah kunci domain keluar dienkripsi ulang di bawah kunci domain aktif yang baru. Kunci domain aktif digunakan untuk mengenkripsi EKT baru. Kunci domain yang kedaluwarsa dapat digunakan hanya untuk mendekripsi EKT yang dienkripsi sebelumnya selama beberapa hari yang setara dengan jumlah kunci domain yang baru dirotasi.

Token domain yang diekspor

Ada kebutuhan rutin untuk menyinkronkan status di antara peserta domain. Hal ini dilakukan melalui pengeksporan status domain setiap kali perubahan dibuat ke domain. Status domain diekspor sebagai token domain yang diekspor.

Nama

Nama domain untuk mengidentifikasi domain ini.

Anggota

Daftar HSM yang merupakan anggota domain, termasuk kunci publik penandatanganan dan perjanjian.

Operator

Daftar entitas, kunci penandatanganan publik, dan peran yang mewakili operator layanan ini.

Aturan

Daftar aturan kuorum untuk setiap perintah yang harus dipenuhi untuk menjalankan perintah pada anggota domain HSM.

Kunci domain terenkripsi

Kunci domain terenkripsi envelope. Kunci domain dienkripsi oleh anggota penandatanganan untuk tiap-tiap anggota yang tercantum di atas, disampul ke kunci perjanjian publik mereka.

Tanda tangan

Tanda tangan pada status domain yang dihasilkan oleh HSM, tentu anggota domain yang diekspor status domain.

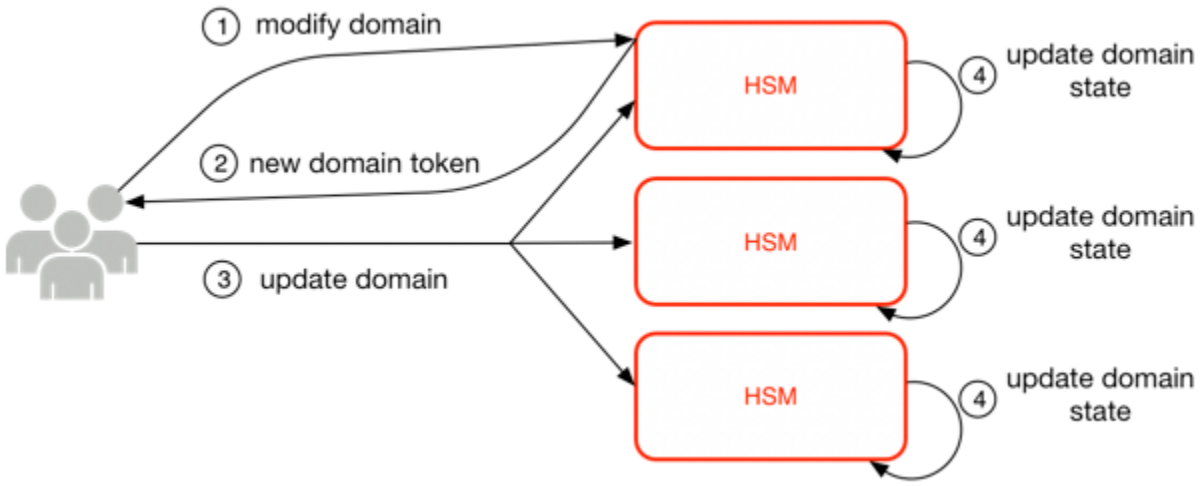
Token domain yang diekspor membentuk sumber kepercayaan mendasar untuk entitas yang beroperasi di dalam domain.

Mengelola status domain

Status domain dikelola melalui perintah terautentikasi kuorum. Perubahan ini termasuk modifikasi daftar peserta tepercaya di domain, modifikasi aturan kuorum untuk menjalankan perintah HSM, dan secara berkala memutar kunci domain. Perintah ini diautentikasi berdasarkan per-perintah sebagai lawan dari operasi sesi yang diautentikasi, seperti yang ditunjukkan pada gambar berikut.

Dalam keadaan inisialisasi dan operasionalnya, HSM berisi satu set kunci identitas asimetris yang dihasilkan sendiri, key pair penandatanganan, dan key pair key-establishment. Melalui proses manual, AWS KMS operator dapat membuat domain awal yang akan dibuat pada HSM pertama di suatu Wilayah. Domain awal ini terdiri atas status domain penuh seperti yang didefinisikan sebelumnya dalam topik ini. Hal ini diinstal melalui perintah penggabungan untuk tiap-tiap anggota HSM yang ditetapkan dalam domain.

Setelah HSM bergabung dengan domain awal, ia terikat dengan aturan yang didefinisikan dalam domain itu. Aturan ini mengatur perintah yang menggunakan kunci kriptografi pelanggan atau membuat perubahan pada status host atau domain. Operasi API sesi terautentikasi yang menggunakan kunci kriptografi Anda telah didefinisikan sebelumnya.



Gambar sebelumnya menggambarkan bagaimana status domain akan dimodifikasi. Prosesnya terdiri atas empat langkah:

1. Perintah berbasis kuorum dikirim ke HSM untuk memodifikasi domain.
2. Status domain baru dihasilkan dan diekspor sebagai token domain baru yang diekspor. Status pada HSM tidak dimodifikasi, yang berarti bahwa perubahan tidak diberlakukan pada HSM.
3. Perintah kedua dikirim ke tiap-tiap HSM di token domain yang baru diekspor untuk memperbarui status domain mereka dengan token domain baru.
4. HSM yang tercantum dalam token domain baru yang diekspor dapat mengautentikasi perintah dan token domain. Mereka juga dapat membongkar kunci domain untuk memperbarui status domain pada semua HSM di domain.

HSM tidak berkomunikasi secara langsung dengan satu sama lain. Sebaliknya, kuorum operator meminta perubahan ke status domain yang menghasilkan token domain baru yang diekspor. Anggota host layanan domain yang digunakan untuk mendistribusikan status domain baru untuk setiap HSM di domain.

Meninggalkan dan bergabung dari domain dilakukan melalui fungsi manajemen HSM. Modifikasi status domain dilakukan melalui fungsi manajemen domain.

Meninggalkan domain

> Penyebab HSM meninggalkan domain, menghapus semua sisa dan kunci dari domain tersebut dari memori.

Bergabung ke domain

Penyebab HSM bergabung dengan domain baru atau memperbarui status domain saat ini ke status domain baru. Domain yang ada digunakan sebagai sumber set awal aturan untuk mengautentikasi pesan ini.

Membuat domain

Penyebab domain baru yang akan dibuat pada HSM. Mengembalikan token domain pertama yang dapat didistribusikan ke HSM anggota domain.

Memodifikasi operator

Menambahkan atau menghapus operator dari daftar operator resmi dan peran mereka dalam domain.

Memodifikasi anggota

Menambahkan atau menghapus HSM dari daftar HSM resmi di domain.

Memodifikasi aturan

Memodifikasi set aturan kuorum yang diperlukan untuk menjalankan perintah pada HSM.

Merotasi kunci domain

Penyebab kunci domain baru yang akan dibuat dan ditandai sebagai kunci domain aktif. Ini memindahkan kunci aktif yang ada ke kunci yang dinonaktifkan dan menghapus kunci yang dinonaktifkan paling lama dari status domain.

Keamanan komunikasi internal

Perintah antara host layanan atau operator AWS KMS dan HSM dijamin melalui dua mekanisme yang ditampilkan dalam Sesi yang diautentikasi: metode permintaan yang ditandatangani kuorum dan sesi yang diautentikasi menggunakan protokol host layanan HSM.

Perintah yang ditandatangani kuorum dirancang sedemikian rupa sehingga tiap operator tidak akan dapat memodifikasi perlindungan keamanan penting yang diberikan HSM. Perintah yang berjalan melalui sesi yang diautentikasi membantu memastikan bahwa hanya operator layanan resmi yang dapat melakukan operasi yang melibatkan kunci KMS. Semua informasi rahasia yang terikat pelanggan diamankan di infrastruktur AWS.

Pembentukan kunci

Untuk mengamankan komunikasi internal, AWS KMS menggunakan dua metode pembentukan kunci yang berbeda. Metode pertama didefinisikan sebagai C (1, 2, ECC DH) di Rekomendasi untuk Skema Pembentukan Kunci Secara Berpasangan Menggunakan Kriptografi Logaritma Diskrit (Revisi 2). Skema ini memiliki inisiator dengan kunci penandatanganan statis. Inisiator menghasilkan dan menandatangani kunci kurva eliptik efemeral Diffie-Hellman (ECDH), yang ditujukan untuk penerima dengan kunci perjanjian ECDH statis. Metode ini menggunakan satu kunci efemeral dan dua kunci statis menggunakan ECDH. Itu adalah derivasi dari label C (1, 2, ECC DH). Metode ini kadang disebut ECDH pas tunggal.

Metode pembentukan kunci yang kedua adalah C (2, 2, ECC, DH). Dalam skema ini, kedua belah pihak memiliki kunci penandatanganan statis, dan mereka menghasilkan, menandatangani, dan saling menukarkan kunci ECDH efemeral. Metode ini menggunakan dua kunci statis dan dua kunci efemeral, masing-masing menggunakan ECDH. Itu adalah derivasi dari label C (2, 2, ECC, DH). Metode ini kadang-kadang disebut ECDH efemeral atau ECDHE. Semua kunci ECDH dihasilkan pada secp384r1 kurva (NIST-P384).

Batas keamanan HSM

Batas keamanan dalam dari AWS KMS adalah HSM. HSM memiliki antarmuka berpemilik dan antarmuka fisik aktif lainnya tidak berada dalam status operasionalnya. HSM operasional disediakan selama inisialisasi dengan kunci kriptografi yang diperlukan untuk menetapkan perannya dalam domain. Bahan kriptografi sensitif dari HSM hanya akan disimpan di dalam memori volatil dan terhapus saat HSM meninggalkan status operasional, termasuk shutdown atau reset yang disengaja maupun yang tidak.

Operasi HSM API diautentikasi baik oleh perintah individu maupun melalui sesi rahasia yang saling diautentikasi yang ditetapkan oleh oleh host layanan.

Perintah yang ditandatangani kuorum

Perintah yang ditandatangani kuorum diterbitkan oleh operator untuk HSM. Bagian ini menjelaskan bagaimana perintah berbasis kuorum dibuat, ditandatangani, dan diautentikasi. Aturan-aturan ini cukup sederhana. Sebagai contoh, perintah Foo membutuhkan dua anggota dari peran Bar untuk diautentikasi. Ada tiga langkah dalam pembuatan dan verifikasi perintah berbasis kuorum. Langkah yang pertama adalah pembuatan perintah awal; yang kedua adalah pengiriman ke operator tambahan untuk ditandatangani; dan yang ketiga adalah verifikasi dan eksekusi.

Untuk tujuan memperkenalkan konsep, anggap bahwa ada satu set autentik kunci publik dan peran operator $\{QOS_s\}$, dan satu set aturan kuorum QR = $\{Perintah_i, Aturan_{\{i, t\}}\}$ di mana masing-masing Aturan adalah seperangkat peran dan jumlah minimum N $\{Peran_t, N_t\}$. Agar perintah memenuhi aturan kuorum, set data perintah harus ditandatangani oleh satu set operator yang terdaftar di $\{QOS_s\}$ sehingga mereka memenuhi salah satu aturan yang tercantum untuk perintah itu. Seperti disebutkan sebelumnya, seperangkat aturan kuorum dan operator disimpan dalam status domain dan token domain yang diekspor.

Dalam praktiknya, penandatangan awal menandatangani perintah $Sig_1 = Sign(dO_{p1}, Perintah)$. Operator kedua juga menandatangani perintah $Sig_2 = Sign(dO_{p2}, Perintah)$. Pesan yang ditandatangani dua kali dikirim ke HSM untuk dilaksanakan. HSM melakukan hal berikut:

1. Untuk setiap tanda tangan, ia mengekstrak kunci publik penandatangan dari status domain dan memverifikasi tanda tangan pada perintah.
2. Ini memverifikasi bahwa himpunan penandatangan memenuhi aturan untuk perintah.

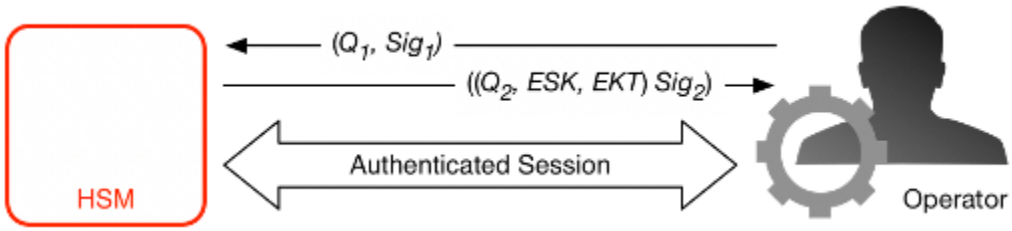
Sesi yang diautentikasi

Operasi kunci Anda berjalan di antara host AWS KMS yang berhadapan secara eksternal dan HSM. Perintah ini berkaitan dengan penciptaan dan penggunaan kunci kriptografi dan pembuatan nomor acak yang aman. Perintah berjalan melalui saluran yang diautentikasi sesi antara host layanan dan HSM. Selain kebutuhan akan autentikasi, sesi ini membutuhkan kerahasiaan. Perintah yang berjalan di sesi ini mencakup kembalinya kunci data cleartext dan pesan terdekripsi yang ditujukan untuk Anda. Untuk memastikan bahwa sesi ini tidak dapat ditumbangkan melalui man-in-the-middle serangan, sesi diautentikasi.

Protokol ini melakukan perjanjian kunci ECDHE yang saling diautentikasi antara HSM dan host layanan. Pertukaran diprakarsai oleh host layanan dan diselesaikan oleh HSM. HSM juga

mengembalikan kunci sesi (SK) yang dienkripsi oleh kunci negosiasi dan token kunci yang diekspor yang berisi kunci sesi. Token kunci yang diekspor berisi masa berlaku, sehingga host layanan harus menegosiasikan ulang kunci sesi setelah masa berlaku token tersebut habis.

Host layanan adalah anggota domain dan memiliki pasangan kunci penandatanganan identitas ($dHOS_i$, $QHOS_i$) dan salinan autentik kunci publik identitas HSM. Ia menggunakan set kunci penandatanganan identitas untuk secara aman menegosiasikan kunci sesi yang dapat digunakan antara host layanan dan setiap HSM di domain. Token kunci yang diekspor memiliki masa berlaku yang terkait dengannya, dan kunci baru harus dinegosiasikan setelah masa berlaku token habis.



Proses dimulai dengan pengenalan host layanan yang memerlukan kunci sesi untuk mengirimkan dan menerima komunikasi sensitif yang mengalir antara host ini dan anggota HSM domain.

1. Host layanan menghasilkan pasangan kunci efemeral ECDH (d_1, Q_1) dan menandatanganinya dengan kunci identitasnya Sig1 = Sign(dOS,Q_1).
2. HSM memverifikasi tanda tangan pada kunci publik yang diterima menggunakan token domain saat ini dan membuat pasangan kunci efemeral ECDH (d_2, Q_2). Ia kemudian melengkapi pertukaran kunci ECD sesuai dengan Rekomendasi untuk Skema Pembentukan Kunci secara Berpasangan Menggunakan Kriptografi Logaritma Diskrit (Revisi) untuk membentuk kunci AES-GCM 256-bit yang dinegosiasikan. HSM membuat kunci sesi AES-GCM 256-bit yang baru. Ia mengenkripsi kunci sesi dengan kunci negosiasi untuk membentuk kunci sesi terenkripsi (ESK). Hal ini juga mengenkripsi kunci sesi di bawah kunci domain sebagai token kunci yang diekspor EKT. Akhirnya, ia menandatangani nilai kembali dengan pasangan kunci identitasnya Sig_2 = Sign(dHSK, (Q_2, ESK, EKT)).
3. Host layanan memverifikasi tanda tangan pada kunci yang diterima menggunakan token domain saat ini. Host layanan kemudian melengkapi pertukaran kunci ECDH sesuai dengan Rekomendasi untuk Skema Pembentukan Kunci secara Berpasangan Menggunakan Kriptografi Logaritma Diskrit (Revisi). Berikutnya ia mendekripsi ESK untuk mendapatkan kunci sesi SK.

Selama masa berlaku di EKT, host layanan dapat menggunakan kunci sesi negosiasi SK untuk mengirim perintah terenkripsi envelope ke HSM. Setiap service-host-initiated perintah atas sesi yang

diautentikasi ini mencakup EKT. HSM merespons menggunakan kunci sesi SK ternegosiasi yang sama.

Proses replikasi untuk kunci Multi-wilayah

AWS KMSmenggunakan mekanisme replikasi lintas wilayah untuk menyalin materi kunci dalam kunci KMS dari HSM dalam satu Wilayah AWS ke HSM yang berbeda. Wilayah AWS Agar mekanisme ini berfungsi, kunci KMS yang sedang direplikasi harus berupa kunci Multi-wilayah. Saat mereplikasi kunci KMS dari satu Wilayah ke Wilayah lainnya, HSM di Wilayah tidak dapat berkomunikasi secara langsung, karena mereka berada di jaringan yang terisolasi. Sebagai gantinya, pesan yang dipertukarkan selama replikasi Lintas wilayah dikirimkan oleh layanan proxy.

Selama replikasi lintas wilayah, setiap pesan yang dihasilkan oleh AWS KMS HSM ditandatangani secara kriptografis menggunakan kunci penandatanganan replikasi. Kunci penandatanganan replikasi (RSK) adalah kunci ECDSA pada kurva NIST P-384. Setiap Wilayah memiliki setidaknya satu RSK, dan komponen publik dari setiap RSK dibagi dengan setiap Wilayah lain dalam partisi yang sama. AWS

Proses replikasi lintas wilayah untuk menyalin materi kunci dari Wilayah A ke Wilayah B berfungsi sebagai berikut:

1. HSM di Wilayah B menghasilkan kunci ECDH fana pada kurva NIST P-384, Kunci Perjanjian Replikasi B (RAKB). Komponen publik RAKB dikirim ke HSM di Wilayah A oleh layanan proxy.
2. HSM di Wilayah A menerima komponen publik RAKB dan kemudian menghasilkan kunci ECDH fana lainnya pada kurva NIST P-384, Kunci Perjanjian Replikasi A (RAKA). HSM menjalankan skema pembentukan kunci ECDH pada RAKA dan komponen publik RAKB, dan memperoleh kunci simetris dari output, Replication Wrapping Key (RWK). RWK digunakan untuk mengenkripsi materi kunci dari kunci KMS Multi-wilayah yang sedang direplikasi.
3. Komponen publik RAKA dan materi kunci yang dienkripsi dengan RWK dikirim ke HSM di Wilayah B melalui layanan proxy.
4. HSM di Wilayah B menerima komponen publik RAKA dan materi kunci yang dienkripsi menggunakan RWK. HSM berasal dari RWK dengan menjalankan skema pembentukan kunci ECDH pada RAKB dan komponen publik RAKA.
5. HSM di Wilayah B menggunakan RWK untuk mendekripsi materi kunci dari Wilayah A.

Perlindungan daya tahan

Daya tahan layanan tambahan untuk kunci yang dihasilkan oleh layanan disediakan oleh penggunaan HSM offline, beberapa penyimpanan token domain yang tidak mudah menguap, dan penyimpanan kunci KMS terenkripsi yang berlebihan. HSMs offline adalah anggota dari domain yang ada. Dengan pengecualian tidak sedang daring dan berpartisipasi dalam operasi domain biasa, HSM offline tampak identik dalam status domain sebagai anggota HSM yang ada.

Desain daya tahan dimaksudkan untuk melindungi semua kunci KMS di suatu Wilayah harus AWS mengalami kerugian skala besar baik HSM online atau set kunci KMS yang disimpan dalam sistem penyimpanan utama kami. AWS KMS keysdengan bahan kunci impor tidak termasuk dalam perlindungan daya tahan yang diberikan kunci KMS lainnya. Jika terjadi kegagalan di seluruh wilayahAWS KMS, bahan kunci yang diimpor mungkin perlu diimpor kembali ke kunci KMS.

HSM offline, dan kredensial untuk mengaksesnya, disimpan di brankas dalam ruang aman yang dipantau di beberapa lokasi geografis independen. Setiap brankas membutuhkan setidaknya satu petugas keamanan AWS dan satu operator AWS KMS, dari dua tim independen di AWS, untuk mendapatkan bahan-bahan ini. Penggunaan bahan-bahan ini diatur oleh kebijakan internal yang mewajibkan kehadiran kuorum operator AWS KMS.

Referensi

Gunakan bahan referensi berikut untuk mendapatkan informasi tentang singkatan, kunci, kontributor, dan sumber yang dikutip dalam dokumen ini.

Topik

Singkatan

Daftar berikut menjelaskan singkatan yang direferensikan dalam dokumen ini.

AES

Standar Enkripsi lanjutan

CDK

kunci data pelanggan

DK

Kunci domain

ECDH

Kurva Eliptik Diffie-Hellman

ECDHE

Kurva Eliptik Diffie-Hellman Efemeral

ECDSA

Algoritma Tanda Tangan Digital Kurva Eliptik

EKT

token kunci yang diekspor

ESK

kunci sesi terenkripsi

GCM

Mode Penghitung Galois

HBK

kunci cadangan HSM

HBKID

pengidentifikasi kunci cadangan HSM

HSM

modul keamanan perangkat keras

RSA

Rivest Shamir dan Adleman (kriptologi)

secp384r1

Standar untuk Kriptografi Efisien dengan kurva acak 384-bit primer 1

SHA256

Algoritma Hash Aman dari panjang intisari 256-bit

Kunci

Daftar berikut mendefinisikan kunci yang direferensikan dalam dokumen ini.

HBK

Kunci dukungan HSM: Kunci dukungan HSM adalah kunci root 256-bit, dari mana kunci penggunaan khusus diturunkan.

DK

Kunci domain: Kunci domain adalah kunci AES-GCM 256-bit. Hal ini dibagi di antara semua anggota domain dan digunakan untuk melindungi bahan kunci cadangan HSM dan kunci sesi host layanan HSM.

DKEK

Kunci enkripsi kunci domain: Kunci enkripsi kunci domain adalah kunci AES-256-GCM yang dihasilkan pada host dan digunakan untuk mengenkripsi rangkaian kunci domain saat ini yang menyinkronkan status domain di seluruh host HSM.

(dHAK, QHAK)

Pasangan kunci perjanjian HSM: Setiap HSM yang diinisiasi memiliki pasangan kunci perjanjian Kurva Eliptik Diffie-Hellman yang dihasilkan secara lokal pada secp384r1 kurva (NIST-P384).

(dE, QE)

Pasangan kunci perjanjian efemeral: HSM dan host layanan menghasilkan kunci perjanjian efemeral. Ini adalah kunci Kurva Eliptik Diffie-Hellman pada secp384r1 kurva (NIST-P384). Ini dihasilkan dalam dua kasus penggunaan: untuk membuat kunci host-to-host enkripsi untuk mengangkut kunci enkripsi kunci domain dalam token domain dan untuk membuat kunci sesi host layanan HSM untuk melindungi komunikasi sensitif.

(dHSK, QHSK)

Pasangan kunci tanda tangan HSM: Setiap HSM yang diinisiasi memiliki pasangan kunci Tanda Tangan Digital Kurva Eliptik yang dihasilkan secara lokal pada secp384r1 kurva (NIST-P384).

(dOS, QOS)

Operator signature key pair: Baik operator host layanan dan AWS KMS operator memiliki kunci penandatanganan identitas yang digunakan untuk mengautentikasi dirinya ke peserta domain lain.

K

Kunci enkripsi data: Kunci AES-GCM 256-bit yang berasal dari HBK menggunakan NIST SP800-108 KDF dalam mode penghitung menggunakan HMAC dengan SHA256.

SK

Kunci sesi: Kunci sesi dibuat sebagai hasil dari kunci Kurva Eliptik Diffie-Hellman yang dipertukarkan antara operator host layanan dan HSM. Tujuan pertukaran adalah untuk mengamankan komunikasi antara host layanan dan anggota domain.

Kontributor

Individu dan organisasi berikut berkontribusi terhadap dokumen ini:

- Ken Beer, General Manager - KMS, Kriptografi AWS
- Matthew Campagna, Kepala Insinyur Keamanan, Kriptografi AWS

daftar pustaka

Untuk informasi tentang HSM AWS Key Management Service, buka Pusat Sumber Daya Keamanan komputer NIST Halaman pencarian Program Validasi Modul Kriptografi dan cari HSM AWS Key Management Service.

Amazon Web Services, Referensi Umum (Versi 1.0), “Menandatangani Permintaan AWS API,” http://docs.aws.amazon.com/general/latest/gr/signing_aws_api_requests.html.

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

Publikasi Standar Pemrosesan Informasi Federal, FIPS PUB 180-4. Standar Hash Aman, Agustus 2012. Tersedia dari https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf.

Publikasi Standar Pemrosesan Informasi Federal 197, Mengumumkan Standar Enkripsi Lanjutan (AES), November 2001. Tersedia dari http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

Publikasi Standar Pemrosesan Informasi Federal 198-1, Kode Autentikasi Pesan Hash Berkunci (HMAC), Juli 2008. Tersedia dari http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf.

Publikasi Khusus NIST 800-52 Revisi 2, Pedoman Seleksi, Konfigurasi, dan Penggunaan Implementasi Transport Layer Security (TLS), Agustus 2019. https://nvlpubs.nist.gov/nistpubs/SpecialPublications /Nist.sp.800-52R2.pdf.

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

Rekomendasi untuk Mode Operasi Cipher Blok: Mode Galois/Penghitung (GCM) dan GMAC, Publikasi Khusus NIST 800-38D, November 2007. Tersedia dari http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf.

Rekomendasi untuk Mode Operasi Cipher Blok: Mode XTS-AES untuk Kerahasiaan pada Perangkat Penyimpanan, Publikasi Khusus NIST 800-38E, Januari 2010. Tersedia dari https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38e.pdf.

Rekomendasi untuk Derivasi Kunci Menggunakan Fungsi Pseudoacak, Publikasi Khusus NIST 800-108, Oktober 2009, Tersedia dari https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-108.pdf.

Rekomendasi untuk Manajemen Kunci - Bagian 1: Umum (Revisi 5), Publikasi Khusus NIST 800-57A, Mei 2020, Tersedia dari https://doi.org/10.6028/NIST.SP.800-57pt1r5.

Rekomendasi untuk Skema Pembentukan Kunci secara Berpasangan Menggunakan Kriptografi Logaritma Diskrit (Revisi), Publikasi Khusus NIST 800-56A Revisi 3, April 2018. Tersedia dari https://nvlpubs.nist.gov/nistpubs/ SpecialPublications /Nist.sp.800-56ar3.pdf.

Rekomendasi untuk Pembuatan Angka Acak Menggunakan Generator Bit Acak Deterministik, Publikasi Khusus NIST 800-90A Revisi 1, Juni 2015, Tersedia dari https://nvlpubs.nist.gov/nistpubs/ / Nist.sp.800-90AR1.pdf. SpecialPublications

SEC 2: Parameter Domain Kurva Eliptik yang Disarankan, Standar untuk Grup Kriptografi yang Efisien, Versi 2.0, 27 Januari 2010.

Penggunaan Algoritma Kriptografi Kurva Eliptik (ECC) dalam Sintaksis Pesan Kriptografi (CMS), Brown, D., Turner, S., Internet Engineering Task Force, Juli 2010, http://tools.ietf.org/html/rfc5753/.

X9.62-2005: Kriptografi Kunci Publik untuk Industri Jasa Keuangan: Algoritma Tanda Tangan Digital Kurva Eliptik (ECDSA), American National Standards Institute, 2005.

Sejarah Dokumen untuk Detail AWS KMS Kriptografi

Tabel berikut menjelaskan perubahan penting pada dokumentasi untuk Rincian AWS Key Management Service Kriptografi. Kami juga sering memperbarui dokumentasi untuk mengatasi umpan balik yang Anda kirimkan kepada kami.

Perubahan	Deskripsi	Tanggal
Konten yang diperbarui	Menambahkan rincian tentang implementasi AWS KMS `ReplicateKey` operasi.	28 Oktober 2021
Perubahan dokumentasi	Ganti istilah customer master key (CMK) dengan AWS KMS keydan kunci KMS.	Agustus 30, 2021
Rilis awal	Membuat panduan ini dari paper teknis KMS Cryptogra phic Details	30 Desember 2020

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.