Panduan Pengguna

AWS Kriptografi Pembayaran

AWS Kriptografi Pembayaran: Panduan Pengguna

Table of Contents

Apa ituAWSKriptografi Pembayaran?

AWSKriptografi Pembayaran dikelolaAWSlayanan yang menyediakan akses ke fungsi kriptografi dan manajemen kunci yang digunakan dalam pemrosesan pembayaran sesuai dengan standar industri kartu pembayaran (PCI) tanpa perlu bagi Anda untuk mendapatkan instans HSM pembayaran khusus.AWS Kriptografi Pembayaran memberi pelanggan yang melakukan fungsi pembayaran seperti acquirer, fasilitator pembayaran, jaringan, sakelar, prosesor, dan bank dengan kemampuan untuk memindahkan operasi kriptografi pembayaran mereka lebih dekat ke aplikasi di cloud dan meminimalkan dependensi pada pusat data tambahan atau fasilitas kolokasi yang berisi HSM pembayaran khusus.

Layanan ini dirancang untuk memenuhi aturan industri yang berlaku termasuk PIN PCI, PCI P2PE, dan PCI DSS, dan layanan memanfaatkan perangkat keras seperti ituPCI PTS HSM V3 dan FIPS 140-2 Level 3 bersertifikat. Ini dirancang untuk mendukung latensi rendah dantingkat up-time dan ketahanan yang tinggi.AWS Kriptografi Pembayaran sepenuhnya elastis dan menghilangkan banyak persyaratan operasional HSM di tempat, seperti kebutuhan untuk menyediakan perangkat keras, mengelola materi kunci dengan aman, dan untuk menjaga cadangan darurat di fasilitas yang aman.AWS Kriptografi Pembayaran juga memberi Anda opsi untuk berbagi kunci dengan mitra Anda secara elektronik, menghilangkan kebutuhan untuk berbagi komponen teks yang jelas.

Anda dapat menggunakanAWSAPI Pesawat Kontrol Kriptografi Pembayaranuntuk membuat dan mengelola kunci.

Anda dapat menggunakanAWSAPI Pesawat Data Kriptografi Pembayaranuntuk menggunakan kunci enkripsi untuk pemrosesan transaksi terkait pembayaran dan operasi kriptografi terkait.

AWSKriptografi Pembayaran menyediakan fitur penting yang dapat Anda gunakan untuk mengelola kunci Anda:

- Membuat dan mengelola simetris dan asimetrisAWSKunci Kriptografi Pembayaran, termasuk kunci TDES, AES, dan RSA dan menentukan tujuan yang dimaksudkan seperti untuk pembuatan CVV atau derivasi kunci DUKPT.
- Secara otomatis menyimpanAWSKunci Kriptografi Pembayaran dengan aman, dilindungi oleh modul keamanan perangkat keras (HSM) sambil menegakkan pemisahan kunci antara kasus penggunaan.
- Buat, hapus, daftar, dan perbarui alias, yang merupakan “nama ramah” yang dapat digunakan untuk mengakses atau mengontrol akses keAWSKunci Kriptografi Pembayaran.

- Tag AndaAWSKunci Kriptografi Pembayaran untuk identifikasi, pengelompokan, otomatisasi, kontrol akses, dan pelacakan biaya.
- Impor dan ekspor kunci simetris antaraAWSKriptografi Pembayaran dan HSM Anda (atau pihak ke-3) menggunakan Key Encryption Keys (KEK) mengikuti TR-31 (Interoperable Secure Key Exchange Key Block Specification).
- Impor dan ekspor simetris Key Encryption Keys (KEK) antaraAWSKriptografi Pembayaran dan sistem lain menggunakan pasangan kunci asimetris berikut dengan menggunakan sarana elektronik seperti TR-34 (Metode Untuk Distribusi Kunci Simetris Menggunakan Teknik Asimetris).

Anda dapat menggunakanAWSKunci Kriptografi Pembayaran dalam operasi kriptografi, seperti:

- Mengenkripsi, mendekripsi, dan mengenkripsi ulang data dengan simetris atau asimetrisAWSKunci Kriptografi Pembayaran.
- Menerjemahkan data sensitif dengan aman (seperti pin pemegang kartu) di antara kunci enkripsi tanpa mengekspos teks yang jelas sesuai dengan aturan PIN PCI.
- Menghasilkan atau memvalidasi data pemegang kartu seperti CVV, CVV2 atau ARQC.
- Buat dan validasi pin pemegang kartu.
- Menghasilkan atau memvalidasi tanda tangan MAC.

Konsep

Pelajari istilah dan konsep dasar yang digunakan dalam Kriptografi AWS Pembayaran dan bagaimana Anda dapat menggunakannya untuk membantu Anda melindungi data Anda.

Alias

Nama yang mudah digunakan yang dikaitkan dengan kunci Kriptografi AWS Pembayaran. Alias dapat digunakan secara bergantian dengan ARN kunci di banyak operasi API Kriptografi Pembayaran. AWS Alias memungkinkan kunci diputar atau diubah tanpa memengaruhi kode aplikasi Anda. Nama alias adalah satu string berisi hingga 256 karakter. Ini secara unik mengidentifikasi kunci Kriptografi AWS Pembayaran terkait dalam akun dan wilayah. Dalam Kriptografi AWS Pembayaran, nama alias selalu dimulai dengan. `alias/`

Format nama alias adalah sebagai berikut:

```
alias/<alias-name>
```

Sebagai contoh:

```
alias/sampleAlias2
```

ARN kunci

ARN kunci adalah Nama Sumber Daya Amazon (ARN) dari entri kunci dalam Kriptografi Pembayaran. AWS Ini adalah pengidentifikasi unik dan sepenuhnya memenuhi syarat untuk kunci Kriptografi AWS Pembayaran. ARN kunci mencakup Akun AWS, wilayah, dan ID yang dihasilkan secara acak. ARN tidak terkait atau berasal dari bahan kunci. Karena mereka secara otomatis ditetapkan selama operasi membuat atau mengimpor, nilai-nilai ini tidak idempoten. Mengimpor kunci yang sama beberapa kali akan menghasilkan beberapa ARN kunci dengan siklus hidupnya sendiri.

Format ARN kunci adalah sebagai berikut:

```
arn:<partition>:payment-cryptography:<region>:<account-id>:alias/<alias-name>
```

Berikut ini adalah contoh kunci ARN:

```
arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h
```

Pengenal Kunci

Pengenal Kunci adalah referensi ke kunci dan satu (atau lebih) dari mereka adalah input khas untuk operasi Kriptografi AWS Pembayaran. Pengidentifikasi kunci yang valid bisa berupa Key Arn a Key Alias.

AWS Kunci Kriptografi Pembayaran

AWS Kunci Kriptografi Pembayaran (kunci) digunakan untuk semua fungsi kriptografi. Kunci dihasilkan secara langsung oleh Anda menggunakan perintah create key atau ditambahkan ke sistem dengan memanggil key import. Asal kunci dapat ditentukan dengan meninjau atribut KeyOrigin. AWS Kriptografi Pembayaran juga mendukung kunci turunan atau perantara yang digunakan selama operasi kriptografi seperti yang digunakan oleh DUKPT.

Kunci-kunci ini memiliki atribut yang tidak dapat diubah dan dapat diubah yang ditentukan pada saat pembuatan. Atribut, seperti algoritma, panjang, dan penggunaan didefinisikan pada saat pembuatan dan tidak dapat diubah. Lainnya, seperti tanggal efektif atau tanggal kedaluwarsa, dapat dimodifikasi. Lihat Referensi API Kriptografi AWS Pembayaran untuk daftar lengkap atribut Kunci Kriptografi AWS Pembayaran.

AWS Kunci Kriptografi Pembayaran memiliki tipe kunci, terutama didefinisikan oleh ANSI X9 TR 31, yang membatasi penggunaannya untuk tujuan yang dimaksudkan sebagaimana ditentukan dalam PCI PIN v3.1 Persyaratan 19.

Atribut terikat ke kunci menggunakan blok kunci saat disimpan, dibagikan dengan akun lain, atau diekspor seperti yang ditentukan dalam PCI PIN v3.1 Persyaratan 18-3.

Kunci diidentifikasi dalam platform Kriptografi AWS Pembayaran menggunakan nilai unik yang dikenal sebagai nama sumber daya Amazon utama (ARN).

> **Note**
>
> Kunci ARN dihasilkan ketika kunci awalnya dibuat atau diimpor ke layanan Kriptografi AWS Pembayaran. Jadi, jika menambahkan materi kunci yang sama beberapa kali menggunakan fungsionalitas kunci impor, materi kunci yang sama akan ditempatkan di bawah beberapa kunci tetapi masing-masing dengan siklus hidup kunci yang berbeda.

Terminologi industri

Topik

Jenis kunci umum

AWK

Kunci kerja pengakuisisi (AWK) adalah kunci yang biasanya digunakan untuk bertukar data antara prosesor pengakuisisi dan jaringan (seperti Visa atau Mastercard). Secara historis AWK memanfaatkan 3DES untuk enkripsi dan akan direpresentasikan sebagai TR31_P0_PIN_ENCRYPTION_KEY.

BDK

Kunci derivasi dasar (BDK) adalah kunci kerja yang digunakan untuk menurunkan kunci berikutnya dan biasanya digunakan sebagai bagian dari proses PCI PIN dan PCI P2PE DUKPT. Hal ini dilambangkan sebagai TR31_B0_BASE_DERIVATION_KEY.

CMK

Kunci master kartu (CMK) adalah satu atau lebih kunci spesifik kartu yang biasanya berasal dari Kunci Master Penerbit, PAN dan PSN dan biasanya merupakan kunci 3DES. Kunci-kunci ini disimpan di EMV Chip selama personalisasi. Contoh CMK termasuk kunci AC, SMI dan SMC.

CMK-AC

Kunci kriptogram aplikasi (AC) digunakan sebagai bagian dari transaksi EMV untuk menghasilkan kriptogram transaksi dan merupakan jenis kunci master kartu.

CMK-SMI

Kunci integritas pesan aman (SMI) digunakan sebagai bagian dari EMV untuk memverifikasi integritas muatan yang dikirim ke kartu menggunakan MAC seperti skrip pembaruan pin. Ini adalah jenis kunci master kartu.

CMK-SMC

Kunci kerahasiaan pesan aman (SMC) digunakan sebagai bagian dari EMV untuk mengenkripsi data yang dikirim ke kartu seperti pembaruan pin. Ini adalah jenis kunci master kartu.

CVK

Kunci verifikasi kartu (CVK) adalah kunci yang digunakan untuk menghasilkan CVV, CVV2 dan nilai serupa menggunakan algoritma yang ditentukan serta memvalidasi input. Hal ini dilambangkan sebagai TR31_C0_CARD_VERIFICATION_KEY.

iCVV

iCVV adalah nilai seperti CVV2 tetapi disematkan dengan data setara track2 pada kartu EMV (Chip). Nilai ini dihitung menggunakan kode layanan 999 dan berbeda dari CVV1/CVV2 untuk mencegah informasi yang dicuri digunakan untuk membuat kredensyal pembayaran baru dari jenis yang berbeda. Misalnya, jika data transaksi chip diperoleh, tidak mungkin menggunakan data ini untuk menghasilkan strip magnetik (CVV1) atau untuk pembelian online (CVV2).

Ini menggunakan ??? kunci

IMK

Sebuah issuer master key (IMK) adalah kunci master yang digunakan sebagai bagian dari personalisasi kartu chip EMV. Biasanya akan ada 3 IMK - masing-masing untuk AC (kriptogram), SMI (kunci master skrip untuk integritas/tanda tangan), dan SMC (kunci master skrip untuk kerahasiaan/enkripsi) kunci.

IK

Kunci awal (IK) adalah kunci pertama yang digunakan dalam proses DUKPT dan berasal dari Kunci Derivasi Dasar (BDK). Tidak ada transaksi yang diproses pada kunci ini, tetapi digunakan untuk mendapatkan kunci future yang akan digunakan untuk transaksi. Metode derivasi untuk membuat IK didefinisikan dalam X9. 24-1:2017. Ketika TDES BDK digunakan, X9. 24-1:2009 adalah standar yang berlaku dan IK diganti dengan Initial Pin Encryption Key (IPEK).

IPEK

Kunci enkripsi PIN awal (IPEK) adalah kunci awal yang digunakan dalam proses DUKPT dan berasal dari Kunci Derivasi Dasar (BDK). Tidak ada transaksi yang diproses pada kunci ini, tetapi digunakan untuk mendapatkan kunci future yang akan digunakan untuk transaksi. IPEK adalah keliru karena kunci ini juga dapat digunakan untuk memperoleh enkripsi data dan kunci mac. Metode derivasi untuk membuat IPEK didefinisikan dalam X9. 24-1:2009. Ketika AES BDK digunakan, X9. 24-1:2017 adalah standar yang berlaku dan IPEK diganti dengan Initial Key (IK).

IWK

Kunci kerja penerbit (IWK) adalah kunci yang biasanya digunakan untuk bertukar data antara penerbit/penerbit prosesor dan jaringan (seperti Visa atau Mastercard). Secara historis IWK memanfaatkan 3DES untuk enkripsi dan direpresentasikan sebagai TR31_P0_PIN_ENCRYPTION_KEY.

KEK

Kunci enkripsi kunci (KEK) adalah kunci yang digunakan untuk mengenkripsi kunci lain baik untuk transmisi atau penyimpanan. Kunci yang dimaksudkan untuk melindungi kunci lain biasanya memiliki KeyUsage TR31_K0_KEY_ENCRYPTION_KEY sesuai dengan standar. TR-31

PEK

Kunci enkripsi PIN (PEK) adalah jenis kunci kerja yang digunakan untuk mengenkripsi PIN baik untuk penyimpanan atau transmisi antara dua pihak. IWK dan AWK adalah dua contoh penggunaan spesifik kunci enkripsi pin. Kunci ini direpresentasikan sebagai TR31_P0_PIN_ENCRYPTION_KEY.

PVK

Kunci verifikasi PIN (PVK) adalah jenis kunci kerja yang digunakan untuk menghasilkan nilai verifikasi PIN seperti PVV. Dua jenis yang paling umum adalah TR31_V1_IBM3624_PIN_VERIFICATION_KEY digunakan untuk menghasilkan nilai offset

IBM3624 dan TR31_V2_VISA_PIN_VERIFICATION_KEY digunakan untuk nilai verifikasi VISA/ABA.

Istilah lain

ARQC

Authorization Request Cryptogram (ARQC) adalah kriptogram yang dihasilkan pada waktu transaksi oleh kartu chip standar EMV (atau implementasi tanpa kontak yang setara). Biasanya, ARQC dihasilkan oleh kartu chip dan diteruskan ke penerbit atau agen mereka untuk memverifikasi pada waktu transaksi.

DUKPT

Derived Unique Key Per Transaction (DUKPT) adalah standar manajemen kunci yang biasanya digunakan untuk menentukan penggunaan kunci enkripsi sekali pakai pada POS/POI fisik. Secara historis DUKPT memanfaatkan 3DES untuk enkripsi. Standar industri untuk DUKPT didefinisikan dalam ANSI X9.24-3-2017.

EMV

EMV (awalnya Europay, Mastercard, Visa) adalah badan teknis yang bekerja dengan pemangku kepentingan pembayaran untuk menciptakan standar dan teknologi pembayaran yang dapat dioperasikan. Salah satu contoh standar adalah untuk kartu chip/contactless dan terminal pembayaran yang berinteraksi dengan mereka, termasuk kriptografi yang digunakan. Derivasi kunci EMV mengacu pada metode menghasilkan kunci unik untuk setiap kartu pembayaran berdasarkan set kunci awal seperti `IMK`

HSM

Modul Keamanan Perangkat Keras (HSM) adalah perangkat fisik yang melindungi operasi kriptografi (misalnya, enkripsi, dekripsi, dan tanda tangan digital) serta kunci yang mendasari yang digunakan untuk operasi ini.

KCV

Key Check Value (KCV) mengacu pada berbagai metode checksum primer yang digunakan untuk membandingkan kunci satu sama lain tanpa memiliki akses ke materi kunci yang sebenarnya. KCV juga telah digunakan untuk validasi integritas (terutama ketika bertukar kunci), meskipun peran ini sekarang disertakan sebagai bagian dari format blok kunci seperti. `TR-31` Untuk kunci TDES, KCV dihitung dengan mengenkripsi 8 byte, masing-masing dengan nilai nol, dengan kunci yang akan diperiksa dan mempertahankan 3 byte urutan tertinggi dari hasil terenkripsi. Untuk

kunci AES, KCV dihitung menggunakan algoritma CMAC di mana data input adalah 16 byte nol dan mempertahankan 3 byte urutan tertinggi dari hasil terenkripsi.

KDH

Key Distribution Host (KDH) adalah perangkat atau sistem yang mengirim kunci dalam proses pertukaran kunci seperti TR-34. Saat mengirim kunci dari Kriptografi AWS Pembayaran, itu dianggap sebagai KDH.

KIF

Fasilitas Injeksi Kunci (KIF) adalah fasilitas aman yang digunakan untuk menginisialisasi terminal pembayaran termasuk memuatnya dengan kunci enkripsi.

KRD

Perangkat Penerima Kunci (KRD) adalah perangkat yang menerima kunci dalam proses pertukaran kunci seperti TR-34. Saat mengirim kunci ke Kriptografi AWS Pembayaran, itu dianggap sebagai KRD.

KSN

Key Serial Number (KSN) adalah nilai yang digunakan sebagai masukan untuk enkripsi/dekripsi DUKPT untuk membuat kunci enkripsi unik per transaksi. KSN biasanya terdiri dari pengenal BDK, ID terminal semi-unik serta penghitung transaksi yang meningkat pada setiap transisi yang diproses pada terminal pembayaran tertentu.

PANCI

Nomor Akun Utama (PAN) adalah pengenal unik untuk akun seperti kartu kredit atau debit. Biasanya panjangnya 13-19 digit. 6-8 digit pertama mengidentifikasi jaringan dan bank penerbit.

Blok PIN

Sebuah blok data yang berisi PIN selama pemrosesan atau transmisi serta elemen data lainnya. Format blok PIN menstandarisasi konten blok PIN dan bagaimana hal itu dapat diproses untuk mengambil PIN. Sebagian besar blok PIN terdiri dari PIN, panjang PIN, dan sering berisi sebagian atau seluruh PAN. AWS Kriptografi Pembayaran mendukung format ISO 9564-1 0, 1, 3 dan 4. Format 4 diperlukan untuk kunci AES. Saat memverifikasi atau menerjemahkan PIN, ada kebutuhan untuk menentukan blok PIN dari data yang masuk atau keluar.

POI

Point of Interaction (POI), juga sering digunakan secara sinonim dengan Point of Sale (POS), adalah perangkat keras yang berinteraksi dengan pemegang kartu untuk menunjukkan kredensi

pembayaran mereka. Contoh POI adalah terminal fisik di lokasi pedagang. Untuk daftar terminal PCI PTS POI bersertifikat, lihat situs web PCI.

PSN

PAN Sequence Number (PSN) adalah nilai numerik yang digunakan untuk membedakan beberapa kartu yang dikeluarkan dengan PAN yang sama.

Kunci publik

Ketika menggunakan asymmetric ciphers (RSA), public key adalah komponen publik dari public-private key pair. 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 asymmetric ciphers (RSA), private key adalah komponen privat dari public-private key pair. Kunci privat digunakan untuk mendekripsi data atau membuat tanda tangan digital. Mirip dengan kunci Kriptografi AWS Pembayaran simetris, kunci pribadi dibuat dengan aman oleh HSM. Mereka didekripsi hanya ke dalam memori volatile HSM dan hanya untuk waktu yang diperlukan untuk memproses permintaan kriptografi Anda.

PVV

Nilai Verifikasi Pin (PVV) adalah nilai algoritmik yang diturunkan dari serangkaian input seperti nomor kartu dan PIN yang menghasilkan nilai yang dapat digunakan untuk validasi berikutnya. Salah satu skema tersebut dikenal sebagai Visa PVV (juga dikenal sebagai metode ABA) meskipun digunakan untuk PIN di jaringan apa pun.

Bungkus/Buka RSA

RSA wrap menggunakan kunci asimetris untuk membungkus kunci simetris (seperti kunci TDES) untuk transmisi ke sistem lain. Hanya sistem dengan kunci pribadi yang cocok yang dapat mendekripsi muatan dan memuat kunci simetris. Sebaliknya, RSA membuka, akan mendekripsi kunci yang dienkripsi dengan aman menggunakan RSA dan kemudian memuat kunci ke dalam Kriptografi Pembayaran. AWS RSA wrap adalah metode pertukaran kunci tingkat rendah dan tidak mengirimkan kunci dalam format blok kunci dan tidak menggunakan penandatanganan payload oleh pihak pengirim. Kontrol alternatif harus dipertimbangkan untuk memastikan pemeliharaan dan atribut kunci tidak bermutasi.

TR-34 juga menggunakan RSA secara internal, tetapi merupakan format terpisah dan tidak dapat dioperasikan.

TR-31

TR-31 (secara formal didefinisikan sebagai ANSI X9 TR 31) adalah format blok kunci yang didefinisikan oleh American National Standards Institute (ANSI) untuk mendukung mendefinisikan atribut kunci dalam struktur data yang sama dengan data kunci itu sendiri. Format blok kunci TR-31 mendefinisikan satu set atribut kunci yang terikat ke kunci sehingga mereka disatukan. AWS Kriptografi Pembayaran menggunakan persyaratan standar TR-31 bila memungkinkan untuk memastikan pemisahan kunci yang tepat dan tujuan utama. TR-31 telah digantikan oleh ANSI X9.143-2022.

TR-34

TR-34 adalah implementasi ANSI X9.24-2 yang menggambarkan protokol untuk mendistribusikan kunci simetris dengan aman (seperti 3DES dan AES) menggunakan teknik asimetris (seperti RSA). AWS Kriptografi Pembayaran menggunakan metode TR-34 untuk mengizinkan impor dan ekspor kunci yang aman.

Layanan terkait

AWS Key Management Service

AWSLayanan Manajemen Kunci (AWSKMS) adalah layanan terkelola yang memudahkan Anda membuat dan mengontrol kunci kriptografi yang digunakan untuk melindungi data Anda.AWS KMS menggunakan modul keamanan perangkat keras (HSM) untuk melindungi dan memvalidasiAWSKunci KMS.

AWS CloudHSM

AWS CloudHSMmenyediakan pelanggan dengan instans HSM tujuan umum khusus diAWSAwan.AWS CloudHSMdapat menyediakan berbagai fungsi kriptografi seperti membuat kunci, penandatanganan data atau mengenkripsi dan mendekripsi data.

Untuk informasi selengkapnya

- Untuk mempelajari tentang istilah dan konsep yang digunakanAWSKriptografi Pembayaran, lihatAWSKonsep Kriptografi Pembayaran.
- Untuk informasi tentangAWSPembayaran Kriptografi Kontrol Pesawat API, lihatAWSReferensi API Pesawat Kontrol Kriptografi Pembayaran.

- Untuk informasi tentangAWSPembayaran Kriptografi Data Pesawat API, lihatAWSReferensi API Pesawat Data Kriptografi Pembayaran.
- Untuk informasi teknis terperinci tentang caranyaAWSKriptografi Pembayaran menggunakan kriptografi dan mengamankanAWSKunci Kriptografi Pembayaran, lihatDetail kriptografi.

Titik akhir untuk AWS Payment Cryptography

Untuk terhubung secara terprogram ke AWS Payment Cryptography, Anda menggunakan titik akhir, URL titik masuk untuk layanan. AWS SDK dan alat baris perintah secara otomatis menggunakan titik akhir default untuk layanan Wilayah AWS berdasarkan konteks wilayah permintaan, jadi biasanya tidak perlu menetapkan nilai ini secara eksplisit. Bila diperlukan, Anda dapat menentukan titik akhir yang berbeda untuk permintaan API Anda.

Titik akhir bidang kendali

Nama wilayah	Wilayah	Titik Akhir	Protokol
AS Timur (Virginia Utara)	`us-east-1`	pesawat kontrol. payment-cryptography.us-east-1.amazonaws.com	HTTPS
AS Timur (Ohio)	`us-east-2`	pesawat kontrol. payment-cryptography.us-east-2.amazonaws.com	HTTPS
AS Barat (Oregon)	`us-west-2`	pesawat kontrol. payment-cryptography.us-west-2.amazonaws.com	HTTPS

Titik akhir bidang data

Nama wilayah	Wilayah	Titik Akhir	Protokol
AS Timur (Virginia Utara)	`us-east-1`	jalur data. payment-cryptography.us-east-1.amazonaws.com	HTTPS
AS Timur (Ohio)	`us-east-2`	jalur data. payment-cryptography.us-east-2.amazonaws.com	HTTPS

Nama wilayah	Wilayah	Titik Akhir	Protokol
AS Barat (Oregon)	`us-west-2`	jalur data. payment-cryptography.us-west-2.amazonaws.com	HTTPS

Memulai Kriptografi AWS Pembayaran

Untuk memulai dengan Kriptografi AWS Pembayaran, pertama-tama Anda ingin membuat kunci dan kemudian menggunakannya dalam berbagai operasi kriptografi. Tutorial di bawah ini menyediakan kasus penggunaan sederhana untuk menghasilkan kunci yang akan digunakan untuk menghasilkan/memverifikasi nilai CVV2. Untuk mencoba contoh lain dan menjelajahi pola penerapan dalam AWS, silakan coba Workshop Kriptografi AWS Pembayaran berikut atau jelajahi proyek sampel kami yang tersedia di Github

Tutorial ini memandu Anda melalui pembuatan satu kunci dan melakukan operasi kriptografi menggunakan kunci. Setelah itu, Anda menghapus kunci jika Anda tidak lagi menginginkannya, yang melengkapi siklus hidup kunci.

Topik

Prasyarat

Sebelum Anda mulai, pastikan bahwa:

- Anda memiliki izin untuk mengakses layanan. Untuk informasi selengkapnya, lihat kebijakan IAM.
- Anda telah AWS CLImenginstal. Anda juga dapat menggunakan AWSSDK atau AWSAPI untuk mengakses Kriptografi AWS Pembayaran, tetapi instruksi dalam tutorial ini menggunakan. AWS CLI

Langkah 1: Buat kunci

Langkah pertama adalah membuat kunci. Untuk tutorial ini, Anda membuat kunci 3DES (2KEY TDES) panjang ganda CVK untuk menghasilkan dan memverifikasi nilai CVV/CVV2.

```
$ aws payment-cryptography create-key \
    --exportable
    --key-attributes KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,\
        KeyClass=SYMMETRIC_KEY,\
        KeyModesOfUse='{Generate=true,Verify=true}'
```

Respons menggemakan kembali parameter permintaan, termasuk ARN untuk panggilan berikutnya serta Nilai Pemeriksaan Kunci (KCV).

```
{
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
        "KeyAttributes": {
            "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyAlgorithm": "TDES_2KEY",
            "KeyModesOfUse": {
                "Encrypt": false,
                "Decrypt": false,
                "Wrap": false,
                "Unwrap": false,
                "Generate": true,
                "Sign": false,
                "Verify": true,
                "DeriveKey": false,
                "NoRestrictions": false
            }
        },
        "KeyCheckValue": "CADDA1",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "Enabled": true,
        "Exportable": true,
        "KeyState": "CREATE_COMPLETE",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "CreateTimestamp": "2023-06-05T06:41:46.648000-07:00",
        "UsageStartTimestamp": "2023-06-05T06:41:46.626000-07:00"
    }
}
```

Perhatikan `KeyArn` yang mewakili kunci, misalnya arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi. Anda membutuhkannya di langkah berikutnya.

Langkah 2: Hasilkan nilai CVV2 menggunakan kunci

Pada langkah ini, Anda menghasilkan CVV2 untuk tanggal tertentu PAN dan kedaluwarsa menggunakan kunci dari langkah 1.

```
$  aws payment-cryptography-data generate-card-validation-data \
    --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi \
    --primary-account-number=171234567890123 \
    --generation-attributes CardVerificationValue2={CardExpiryDate=0123}
```

```
{
    "CardDataGenerationKeyCheckValue": "CADDA1",
    "CardDataGenerationKeyIdentifier": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/tqv5yij6wtxx64pi",
    "CardDataType": "CARD_VERIFICATION_VALUE_2",
    "CardDataValue": "144"
}
```

Perhatikan`cardDataValue`, dalam hal ini angka 3 digit 144. Anda membutuhkannya di langkah berikutnya.

Langkah 3: Verifikasi nilai yang dihasilkan pada langkah 2

Dalam contoh ini, Anda memvalidasi CVV2 dari langkah 2 menggunakan kunci yang Anda buat di langkah 1.

Jalankan perintah berikut untuk memvalidasi CVV2.

```
$  aws payment-cryptography-data  verify-card-validation-data \
    --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi \
    --primary-account-number=171234567890123 \
    --verification-attributes CardVerificationValue2={CardExpiryDate=0123} \
    --validation-data 144
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
```

```
    "KeyCheckValue": "CADDA1"
}
```

Layanan mengembalikan respons HTTP 200 untuk menunjukkan bahwa itu memvalidasi CVV2.

Langkah 4: Lakukan tes negatif

Pada langkah ini, Anda membuat tes negatif di mana CVV2 tidak benar dan tidak memvalidasi. Anda mencoba memvalidasi CVV2 yang salah menggunakan kunci yang Anda buat di langkah 1. Ini adalah operasi yang diharapkan misalnya jika pemegang kartu memasukkan CVV2 yang salah saat checkout.

```
$  aws payment-cryptography-data  verify-card-validation-data \
    --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi \
    --primary-account-number=171234567890123 \
    --verification-attributes CardVerificationValue2={CardExpiryDate=0123} \
    --validation-data 999
```

```
Card validation data verification failed.
```

Layanan mengembalikan respons HTTP 400 dengan pesan “Verifikasi data validasi kartu gagal” dan alasan INVALID_VALIDATION_DATA.

Langkah 5: (Opsional) Bersihkan

Sekarang Anda dapat menghapus kunci yang Anda buat di langkah 1. Untuk meminimalkan perubahan yang tidak dapat dipulihkan, periode penghapusan kunci default adalah tujuh hari.

```
$ aws payment-cryptography delete-key \
    --key-identifier=arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi
```

```
{
    "Key": {
        "CreateTimestamp": "2022-10-27T08:27:51.795000-07:00",
        "DeletePendingTimestamp": "2022-11-03T13:37:12.114000-07:00",
        "Enabled": true,
```

```
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_3KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": true,
                "DeriveKey": false,
                "Encrypt": true,
                "Generate": false,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": true,
                "Verify": false,
                "Wrap": true
            },
            "KeyUsage": "TR31_P0_PIN_ENCRYPTION_KEY"
        },
        "KeyCheckValue": "CADDA1",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "DELETE_PENDING",
        "UsageStartTimestamp": "2022-10-27T08:27:51.753000-07:00"
    }
}
```

Perhatikan dua bidang dalam output. `deletePendingTimestamp`Ini diatur ke tujuh hari di masa depan secara default. KeyState diatur ke`DELETE_PENDING`. Anda dapat membatalkan penghapusan ini kapan saja sebelum waktu penghapusan yang dijadwalkan dengan menelepon. `restore-key`

Mengelola kunci

Untuk memulai dengan Kriptografi AWS Pembayaran, Anda akan ingin membuat kunci Kriptografi AWS Pembayaran.

Topik di bagian ini menjelaskan cara membuat dan mengelola berbagai jenis kunci Kriptografi AWS Pembayaran, mulai dari pembuatan hingga penghapusan. Ini mencakup topik tentang membuat, mengedit dan melihat kunci, menandai kunci, membuat alias kunci, serta mengaktifkan dan menonaktifkan kunci.

Topik

Menghasilkan kunci

Anda dapat membuat kunci Kriptografi AWS Pembayaran dengan menggunakan operasi CreateKey API. Selama proses ini, Anda akan menentukan berbagai atribut kunci atau output yang dihasilkan seperti algoritma kunci (misalnya, TDES_3KEY), (misalnya TR31_P0_PIN_ENCRYPTION_KEY), operasi yang diizinkan KeyUsage (misalnya, mengenkripsi, menandatangani) dan apakah itu dapat diekspor. Anda tidak dapat mengubah properti ini setelah kunci Kriptografi AWS Pembayaran dibuat.

Contoh

- Menghasilkan Kunci Nilai Verifikasi PIN (PVV)

Menghasilkan kunci TDES 2KEY

Example

Perintah ini menghasilkan kunci TDES 2KEY untuk tujuan menghasilkan dan memverifikasi nilai CVV/CVV2. Respons menggemakan kembali parameter permintaan, termasuk ARN untuk panggilan berikutnya serta KCV (Key Check Value).

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,\
KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY, \
KeyModesOfUse='{Generate=true,Verify=true}'
```


```
                        {
    "Key": {
        "CreateTimestamp": "2022-10-26T16:04:11.642000-07:00",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
hjprdg5o4jtgs5tw",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_2KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": false,
                "DeriveKey": false,
                "Encrypt": false,
                "Generate": true,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": false,
                "Verify": true,
                "Wrap": false
            },
            "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"
        },
        "KeyCheckValue": "B72F",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
```

```
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "2022-10-26T16:04:11.559000-07:00"
    }
}
```

Menghasilkan Kunci Enkripsi Pin

Example Menghasilkan Kunci Enkripsi Pin (PEK)

Perintah ini menghasilkan kunci TDES 3KEY untuk tujuan mengenkripsi nilai PIN (dikenal sebagai Kunci Enkripsi Pin). Kunci ini dapat digunakan untuk mengamankan PIN penyimpanan atau untuk mendekripsi PIN yang disediakan selama upaya verifikasi, misalnya selama transaksi. Respons menggemakan kembali parameter permintaan, termasuk ARN untuk panggilan berikutnya serta KCV (Key Check Value).

```
$ aws payment-cryptography create-key --exportable --key-attributes \
                        KeyAlgorithm=TDES_3KEY,KeyUsage=TR31_P0_PIN_ENCRYPTION_KEY, \
                        KeyClass=SYMMETRIC_KEY,/

 KeyModesOfUse='{Encrypt=true,Decrypt=true,Wrap=true,Unwrap=true}'
```

```
{
    "Key": {
        "CreateTimestamp": "2022-10-27T08:27:51.795000-07:00",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_3KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": true,
                "DeriveKey": false,
                "Encrypt": true,
                "Generate": false,
                "NoRestrictions": false,
                "Sign": false,
```

```
                "Unwrap": true,
                "Verify": false,
                "Wrap": true
            },
            "KeyUsage": "TR31_P0_PIN_ENCRYPTION_KEY"
        },
        "KeyCheckValue": "9CA6",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "2022-10-27T08:27:51.753000-07:00"
    }
}
```

Buat kunci asimetris (RSA)

Example

Dalam contoh ini, kita akan menghasilkan asimetris RSA 2048 bit key pair baru. Kunci pribadi baru akan dihasilkan serta kunci publik yang cocok. Kunci publik dapat diambil menggunakan get PublicCertificate API.

```
$ aws payment-cryptography create-key --exportable \
--key-attributes
 KeyAlgorithm=RSA_2048,KeyUsage=TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION, \
KeyClass=ASYMMETRIC_KEY_PAIR,KeyModesOfUse='{Encrypt=true,
 Decrypt=True,Wrap=True,Unwrap=True}'
```

```
{
    "Key": {
        "CreateTimestamp": "2022-11-15T11:15:42.358000-08:00",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
nsq2i3mbg6sn775f",
        "KeyAttributes": {
            "KeyAlgorithm": "RSA_2048",
            "KeyClass": "ASYMMETRIC_KEY_PAIR",
            "KeyModesOfUse": {
                "Decrypt": true,
```

```
                "DeriveKey": false,
                "Encrypt": true,
                "Generate": false,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": true,
                "Verify": false,
                "Wrap": true
            },
            "KeyUsage": "TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION"
        },
        "KeyCheckValue": "40AD487F",
        "KeyCheckValueAlgorithm": "CMAC",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "2022-11-15T11:15:42.182000-08:00"
    }
}
```

Menghasilkan Kunci Nilai Verifikasi PIN (PVV)

Example

Perintah ini menghasilkan kunci TDES 3KEY untuk tujuan menghasilkan nilai PVV (dikenal sebagai Nilai Verifikasi Pin). Anda dapat menggunakan kunci ini untuk menghasilkan nilai PVV yang dapat dibandingkan dengan PVV terhitung berikutnya. Respons menggemakan kembali parameter permintaan, termasuk ARN untuk panggilan berikutnya serta KCV (Key Check Value).

```
$ aws payment-cryptography create-key --exportable/
--key-attributes KeyAlgorithm=TDES_3KEY,KeyUsage=TR31_V2_VISA_PIN_VERIFICATION_KEY,/
KeyClass=SYMMETRIC_KEY,KeyModesOfUse='{Generate=true,Verify=true}'
```

```
{
    "Key": {
        "CreateTimestamp": "2022-10-27T10:22:59.668000-07:00",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
j4u4cmnzkelhc6yb",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_3KEY",
```

```
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": false,
                "DeriveKey": false,
                "Encrypt": false,
                "Generate": true,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": false,
                "Verify": true,
                "Wrap": false
            },
            "KeyUsage": "TR31_V2_VISA_PIN_VERIFICATION_KEY"
        },
        "KeyCheckValue": "5132",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "2022-10-27T10:22:59.614000-07:00"
    }
}
```

Daftar kunci

List Keys menyajikan daftar kunci yang dapat diakses oleh pemanggil di akun ini dan Wilayah.

Example

```
$ aws payment-cryptography list-keys
```

```
 {"Keys": [
    {
        "CreateTimestamp": "2022-10-12T10:58:28.920000-07:00",
        "Enabled": false,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
alsuwfxug3pgy6xh",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_3KEY",
            "KeyClass": "SYMMETRIC_KEY",
```

```
            "KeyModesOfUse": {
                "Decrypt": true,
                "DeriveKey": false,
                "Encrypt": true,
                "Generate": false,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": true,
                "Verify": false,
                "Wrap": true
            },
            "KeyUsage": "TR31_P1_PIN_GENERATION_KEY"
        },
        "KeyCheckValue": "369D",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStopTimestamp": "2022-10-27T14:19:42.488000-07:00"
    }
    ]
}
```


Mengaktifkan dan menonaktifkan kunci

Anda dapat menonaktifkan dan mengaktifkan kembali kunci Kriptografi AWS Pembayaran. Saat Anda membuat kunci, itu diaktifkan secara default. Jika Anda menonaktifkan kunci, itu tidak dapat digunakan dalam operasi kriptografi apa pun sampai Anda mengaktifkannya kembali. Perintah penggunaan mulai/hentikan segera berlaku, jadi disarankan agar Anda meninjau penggunaan sebelum melakukan perubahan seperti itu. Anda juga dapat mengatur perubahan (mulai atau menghentikan penggunaan) agar berlaku di masa mendatang menggunakan `timestamp` parameter opsional.

Karena bersifat sementara dan mudah dibatalkan, menonaktifkan kunci Kriptografi AWS Pembayaran adalah alternatif yang lebih aman untuk menghapus kunci Kriptografi AWS Pembayaran, tindakan yang merusak dan tidak dapat diubah. Jika Anda mempertimbangkan untuk menghapus kunci Kriptografi AWS Pembayaran, nonaktifkan terlebih dahulu dan pastikan bahwa Anda tidak perlu menggunakan kunci untuk mengenkripsi atau mendekripsi data di masa mendatang.

Topik

- Mulai penggunaan kunci

- Hentikan penggunaan kunci

Mulai penggunaan kunci

Penggunaan kunci harus diaktifkan untuk menggunakan kunci untuk operasi kriptografi. Jika kunci tidak diaktifkan, Anda dapat menggunakan operasi ini untuk membuatnya dapat digunakan. Bidang `UsageStartTimestamp` akan mewakili kapan kunci menjadi/akan menjadi aktif. Ini akan menjadi masa lalu untuk token yang diaktifkan, dan di masa depan jika tertunda aktivasi.

Example

Dalam contoh ini, kunci diminta untuk diaktifkan untuk penggunaan kunci. Respons mencakup informasi kunci dan flag enable telah dialihkan ke true. Ini juga akan tercermin dalam objek respons list-keys.

```
$ aws payment-cryptography start-key-usage --key-identifier "arn:aws:payment-
cryptography:us-east-2:111122223333:key/alsuwfxug3pgy6xh"
```

```
{
      "Key": {
      "CreateTimestamp": "2022-10-12T10:58:28.920000-07:00",
      "Enabled": true,
      "Exportable": true,
      "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
alsuwfxug3pgy6xh",
      "KeyAttributes": {
          "KeyAlgorithm": "TDES_3KEY",
          "KeyClass": "SYMMETRIC_KEY",
          "KeyModesOfUse": {
              "Decrypt": true,
              "DeriveKey": false,
              "Encrypt": true,
              "Generate": false,
              "NoRestrictions": false,
              "Sign": false,
              "Unwrap": true,
              "Verify": false,
              "Wrap": true
          },
          "KeyUsage": "TR31_P1_PIN_GENERATION_KEY"
      },
```

```
      "KeyCheckValue": "369D",
      "KeyCheckValueAlgorithm": "ANSI_X9_24",
      "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
      "KeyState": "CREATE_COMPLETE",
      "UsageStartTimestamp": "2022-10-27T14:09:59.468000-07:00"
  }
}
```

Hentikan penggunaan kunci

Jika Anda tidak lagi berencana untuk menggunakan kunci, Anda dapat menghentikan penggunaan kunci untuk mencegah operasi kriptografi lebih lanjut. Operasi ini tidak permanen, sehingga Anda dapat membalikkannya menggunakan penggunaan kunci awal. Anda juga dapat mengatur kunci untuk dinonaktifkan di masa depan. Bidang `UsageStopTimestamp` akan mewakili kapan kunci menjadi/akan dinonaktifkan.

Example

Dalam contoh ini, diminta untuk menghentikan penggunaan kunci di masa mendatang. Setelah eksekusi, kunci ini tidak dapat digunakan untuk operasi kriptografi kecuali diaktifkan kembali melalui penggunaan kunci awal Respons mencakup informasi kunci dan tanda aktifkan telah dialihkan ke false. Ini juga akan tercermin dalam objek respons list-keys.

```
$ aws payment-cryptography stop-key-usage --key-identifier "arn:aws:payment-
cryptography:us-east-2:111122223333:key/alsuwfxug3pgy6xh"
```

```
{
  "Key": {
      "CreateTimestamp": "2022-10-12T10:58:28.920000-07:00",
      "Enabled": false,
      "Exportable": true,
      "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
alsuwfxug3pgy6xh",
      "KeyAttributes": {
          "KeyAlgorithm": "TDES_3KEY",
          "KeyClass": "SYMMETRIC_KEY",
          "KeyModesOfUse": {
              "Decrypt": true,
              "DeriveKey": false,
              "Encrypt": true,
              "Generate": false,
```

```
            "NoRestrictions": false,
            "Sign": false,
            "Unwrap": true,
            "Verify": false,
            "Wrap": true
        },
        "KeyUsage": "TR31_P1_PIN_GENERATION_KEY"
    },
    "KeyCheckValue": "369D",
    "KeyCheckValueAlgorithm": "ANSI_X9_24",
    "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
    "KeyState": "CREATE_COMPLETE",
    "UsageStopTimestamp": "2022-10-27T14:09:59.468000-07:00"
  }
}
```

Menghapus kunci

Menghapus kunci Kriptografi AWS Pembayaran menghapus materi kunci dan semua metadata yang terkait dengan kunci dan tidak dapat diubah kecuali salinan kunci tersedia di luar Kriptografi Pembayaran. AWS Setelah kunci dihapus, Anda tidak dapat lagi mendekripsi data yang dienkripsi di bawah kunci itu, yang berarti bahwa data mungkin menjadi tidak dapat dipulihkan. Anda harus menghapus kunci hanya ketika Anda yakin bahwa Anda tidak perlu menggunakannya lagi dan tidak ada pihak lain yang menggunakan kunci ini. Jika Anda tidak yakin, pertimbangkan untuk menonaktifkan kunci alih-alih menghapusnya. Anda dapat mengaktifkan kembali kunci yang dinonaktifkan jika Anda perlu menggunakannya lagi nanti, tetapi Anda tidak dapat memulihkan kunci Kriptografi AWS Pembayaran yang dihapus kecuali Anda dapat mengimpornya kembali dari sumber lain.

Sebelum menghapus kunci, Anda harus memastikan bahwa Anda tidak lagi membutuhkan kunci. AWS Kriptografi Pembayaran tidak menyimpan hasil operasi kriptografi seperti CVV2 dan tidak dapat menentukan apakah kunci diperlukan untuk materi kriptografi yang persisten.

AWS Kriptografi Pembayaran tidak pernah menghapus kunci milik AWS akun aktif kecuali Anda secara eksplisit menjadwalkannya untuk dihapus dan masa tunggu wajib berakhir.

Namun, Anda dapat memilih untuk menghapus kunci Kriptografi AWS Pembayaran karena satu atau beberapa alasan berikut:

- Untuk menyelesaikan siklus hidup kunci untuk kunci yang tidak lagi Anda perlukan

- Untuk menghindari overhead manajemen yang terkait dengan pemeliharaan kunci Kriptografi AWS Pembayaran yang tidak digunakan

> **Note**
>
> Jika Anda menutup atau menghapus Akun AWS, kunci Kriptografi AWS Pembayaran Anda menjadi tidak dapat diakses. Anda tidak perlu menjadwalkan penghapusan kunci Kriptografi AWS Pembayaran Anda terpisah dari penutupan akun.

AWS Kriptografi Pembayaran mencatat entri di AWS CloudTraillog Anda ketika Anda menjadwalkan penghapusan kunci Kriptografi AWS Pembayaran dan ketika kunci Kriptografi AWS Pembayaran benar-benar dihapus.

Tentang masa tunggu

Karena menghapus kunci tidak dapat diubah, Kriptografi AWS Pembayaran mengharuskan Anda untuk menetapkan masa tunggu antara 3-180 hari. Masa tunggu default adalah tujuh hari.

Namun, masa tunggu sebenarnya mungkin hingga 24 jam lebih lama dari yang Anda jadwalkan. Untuk mendapatkan tanggal dan waktu aktual ketika kunci Kriptografi AWS Pembayaran akan dihapus, gunakan GetKey operasi. Pastikan untuk mencatat zona waktu.

Selama masa tunggu, status kunci Kriptografi AWS Pembayaran dan status kunci adalah Penghapusan tertunda.

> **Note**
>
> Kunci Kriptografi AWS Pembayaran yang tertunda penghapusan tidak dapat digunakan dalam operasi kriptografi apa pun.

Setelah masa tunggu berakhir, Kriptografi AWS Pembayaran menghapus kunci Kriptografi AWS Pembayaran, aliasnya, dan semua metadata Kriptografi Pembayaran terkait AWS .

Gunakan masa tunggu untuk memastikan bahwa Anda tidak memerlukan kunci Kriptografi AWS Pembayaran sekarang atau di masa depan. Jika Anda menemukan bahwa Anda membutuhkan kunci selama masa tunggu, Anda dapat membatalkan penghapusan kunci sebelum masa tunggu berakhir.

Setelah masa tunggu berakhir, Anda tidak dapat membatalkan penghapusan kunci, dan layanan menghapus kunci.

Example

Dalam contoh ini, kunci diminta untuk dihapus. Selain informasi kunci dasar, dua bidang yang relevan adalah bahwa status kunci telah diubah menjadi DELETE_PENDING dan deletePendingTimestamp mewakili kapan kunci saat ini dijadwalkan untuk dihapus.

```
$ aws payment-cryptography delete-key \
                    --key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/kwapwa6qaifllw2h
```


```
  {
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyUsage": "TR31_V2_VISA_PIN_VERIFICATION_KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyAlgorithm": "TDES_3KEY",
            "KeyModesOfUse": {
                "Encrypt": false,
                "Decrypt": false,
                "Wrap": false,
                "Unwrap": false,
                "Generate": true,
                "Sign": false,
                "Verify": true,
                "DeriveKey": false,
                "NoRestrictions": false
            }
        },
        "KeyCheckValue": "",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "Enabled": false,
        "Exportable": true,
        "KeyState": "DELETE_PENDING",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "CreateTimestamp": "2023-06-05T12:01:29.969000-07:00",
```

```
        "UsageStopTimestamp": "2023-06-05T14:31:13.399000-07:00",
        "DeletePendingTimestamp": "2023-06-12T14:58:32.865000-07:00"
    }
}
```

Example

Dalam contoh ini, penghapusan yang tertunda dibatalkan. Setelah berhasil diselesaikan, kunci tidak akan lagi dihapus sesuai jadwal sebelumnya. Respons berisi informasi kunci dasar; selain itu, dua bidang yang relevan telah berubah - `KeyState` dan`deletePendingTimestamp`. `KeyState`dikembalikan ke nilai CREATE_COMPLETE, sementara `DeletePendingTimestamp` dihapus.

```
$ aws payment-cryptography restore-key --key-identifier arn:aws:payment-
cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h
```

```
{
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyUsage": "TR31_V2_VISA_PIN_VERIFICATION_KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyAlgorithm": "TDES_3KEY",
            "KeyModesOfUse": {
                "Encrypt": false,
                "Decrypt": false,
                "Wrap": false,
                "Unwrap": false,
                "Generate": true,
                "Sign": false,
                "Verify": true,
                "DeriveKey": false,
                "NoRestrictions": false
            }
        },
        "KeyCheckValue": "",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "Enabled": false,
        "Exportable": true,
        "KeyState": "CREATE_COMPLETE",
```

```
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "CreateTimestamp": "2023-06-08T12:01:29.969000-07:00",
        "UsageStopTimestamp": "2023-06-08T14:31:13.399000-07:00"
    }
}
```

Kunci impor dan ekspor

AWS Kunci Kriptografi Pembayaran dapat diimpor dari solusi lain atau diekspor ke solusi lain (seperti HSM lainnya). Ini adalah kasus penggunaan umum untuk bertukar kunci dengan penyedia layanan menggunakan fungsionalitas impor dan ekspor. Sebagai layanan cloud, AWS Payment Cryptography mengambil pendekatan elektronik modern untuk manajemen kunci sambil membantu Anda mempertahankan kepatuhan dan kontrol yang berlaku. Tujuan jangka panjangnya adalah untuk menjauh dari komponen kunci berbasis kertas menuju sarana pertukaran kunci elektronik berbasis standar.

Pertukaran Kunci Enkripsi Kunci (KEK)

AWS Kriptografi Pembayaran mendorong penggunaan kriptografi kunci publik (RSA) untuk pertukaran kunci awal menggunakan norma ANSI X9.24 TR-34 yang mapan. Nama umum untuk tipe kunci awal ini termasuk Key Encryption Key (KEK), Zone Master Key (ZMK) dan Zone Control Master Key (ZCMK). Jika sistem atau mitra Anda belum dapat mendukung TR-34, Anda juga dapat mempertimbangkan untuk menggunakan RSA Wrap/Unwrap.

Jika Anda memiliki kebutuhan untuk terus memproses komponen kunci paper sampai semua mitra mendukung pertukaran kunci elektronik, Anda dapat mempertimbangkan untuk mempertahankan HSM offline untuk tujuan ini.

> **Note**
>
> Jika Anda ingin mengimpor kunci pengujian Anda sendiri, silakan periksa proyek sampel di Github. Untuk petunjuk tentang cara mengimpor/mengekspor kunci dari platform lain, silakan baca panduan pengguna untuk platform tersebut.

Pertukaran Kunci Kerja (WK)

AWS Kriptografi Pembayaran menggunakan norma industri yang relevan (ANSI X9.24 TR 31-2018) untuk bertukar kunci kerja. TR-31 mengasumsikan bahwa KEK sebelumnya telah

dipertukarkan. Ini konsisten dengan persyaratan PIN PCI untuk mengikat materi kunci secara kriptografis ke jenis dan penggunaannya setiap saat. Kunci kerja memiliki berbagai nama termasuk kunci kerja pengakuisisi, kunci kerja penerbit, BDK, IPEK, dll.

Topik

Kunci impor

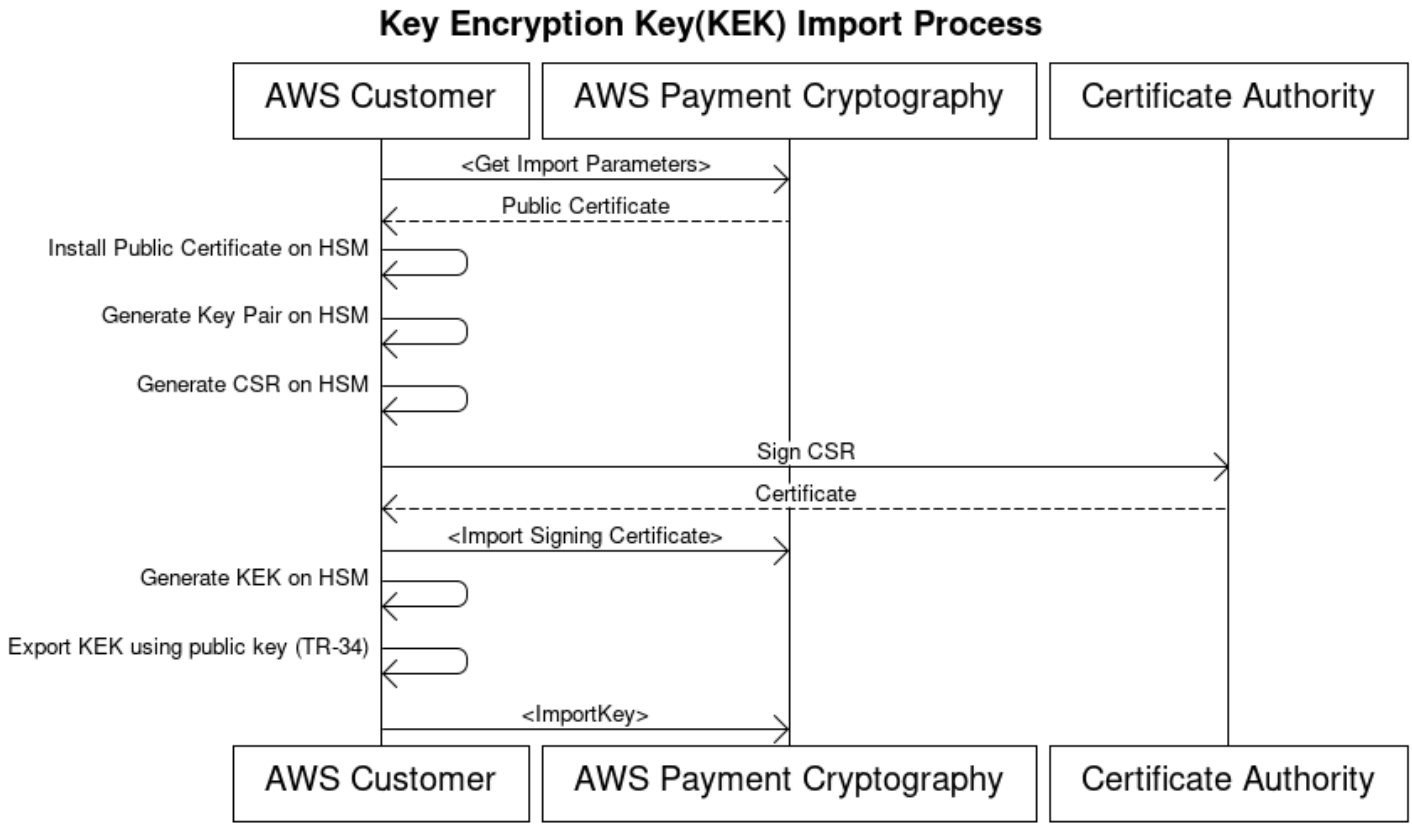
> **Important**
>
> Contoh mungkin memerlukan AWS CLI V2 versi terbaru. Sebelum memulai, pastikan Anda telah meningkatkan ke versi terbaru.

Topik

Mengimpor kunci simetris

Topik

Kunci impor menggunakan teknik asimetris (TR-34)



Tinjauan: TR-34 menggunakan kriptografi asimetris RSA untuk mengenkripsi kunci simetris untuk pertukaran serta memastikan sumber data (penandatanganan). Ini memastikan kerahasiaan (enkripsi) dan integritas (tanda tangan) dari kunci yang dibungkus.

Jika Anda ingin mengimpor kunci Anda sendiri, silakan periksa proyek sampel di Github. Untuk petunjuk tentang cara mengimpor/mengekspor kunci dari platform lain, silakan baca panduan pengguna untuk platform tersebut.

1. Panggil perintah inisialisasi impor

   Panggilan `get-parameters-for-import` untuk menginisialisasi proses impor. API ini akan menghasilkan keypair untuk tujuan impor kunci, menandatangani kunci dan mengembalikan sertifikat dan root sertifikat. Pada akhirnya, kunci yang akan diekspor harus dienkripsi menggunakan kunci ini. Dalam terminologi TR-34, ini dikenal sebagai Sertifikat KRD. Perhatikan bahwa sertifikat ini berumur pendek dan hanya ditujukan untuk tujuan ini.

2. Instal sertifikat publik pada sistem sumber utama

   Dengan banyak HSM, Anda mungkin perlu menginstal/memuat/mempercayai sertifikat publik yang dihasilkan pada langkah 1 untuk mengekspor kunci menggunakannya.

3. Hasilkan kunci publik dan berikan root sertifikat ke Kriptografi AWS Pembayaran

   Untuk memastikan integritas muatan yang ditransmisikan, itu ditandatangani oleh pihak pengirim (dikenal sebagai Host Distribusi Utama atau KDH). Pihak pengirim akan ingin menghasilkan kunci publik untuk tujuan ini dan kemudian membuat sertifikat kunci publik (X509) yang dapat diberikan kembali ke Kriptografi AWS Pembayaran. AWS Private CA adalah salah satu opsi untuk menghasilkan sertifikat, tetapi tidak ada batasan pada otoritas sertifikat yang digunakan.

   Setelah Anda memiliki sertifikat, Anda akan ingin memuat sertifikat root ke Kriptografi AWS Pembayaran menggunakan `importKey` perintah dan KeyMaterialType dari `ROOT_PUBLIC_KEY_CERTIFICATE` dan KeyUsageType dari. `TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE`

4. Ekspor kunci dari sistem sumber

   Banyak HSM dan sistem terkait mendukung kemampuan untuk mengekspor kunci menggunakan norma TR-34. Anda akan ingin menentukan kunci publik dari langkah 1 sebagai sertifikat KRD (enkripsi) dan kunci dari langkah 3 sebagai sertifikat KDH (penandatanganan). Untuk mengimpor ke Kriptografi AWS Pembayaran, Anda akan ingin menentukan format menjadi TR-34.2012 non-CMS dua format pass yang juga dapat disebut sebagai format TR-34 Diebold.

5. Panggil kunci impor

   Sebagai langkah terakhir, Anda akan memanggil ImportKey API dengan file KeyMaterialType . `TR34_KEY_BLOCK` `certificate-authority-public-key-identifier`Akan menjadi keYarn dari root CA yang diimpor pada langkah 3, `key-material` akan dibungkus bahan kunci dari langkah 4 dan `signing-key-certificate` merupakan sertifikat daun dari langkah 3. Anda juga perlu memberikan token impor dari langkah 1.

6. Gunakan kunci yang diimpor untuk operasi kriptografi atau impor berikutnya

   Jika yang diimpor KeyUsage adalah TR31_K0_KEY_ENCRYPTION_KEY, maka kunci ini dapat digunakan untuk impor kunci berikutnya menggunakan TR-31. Jika jenis kunci adalah jenis lain (seperti TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY), maka kunci tersebut dapat langsung digunakan untuk operasi kriptografi.

Kunci impor menggunakan teknik asimetris (RSA Unwrap)

Ikhtisar: Kriptografi AWS Pembayaran mendukung pembungkus/buka RSA untuk pertukaran kunci ketika TR-34 tidak layak. Mirip dengan TR-34, teknik ini menggunakan kriptografi asimetris RSA untuk mengenkripsi kunci simetris untuk pertukaran. Namun, tidak seperti TR-34, metode ini tidak memiliki muatan yang ditandatangani oleh pihak pengirim. Selain itu, teknik pembungkus RSA ini tidak menjaga integritas metadata kunci selama transfer karena tidak menyertakan blok kunci.

> **Note**
>
> Bungkus RSA dapat digunakan untuk mengimpor atau mengekspor kunci TDES dan AES-128.

1. Panggil perintah inisialisasi impor

   Panggilan `get-parameters-for-import` untuk menginisialisasi proses impor dengan tipe material kunci dari KEY_CRYPTOGRAM. WrappingKeyAlgorithm bisa RSA_2048 saat menukar kunci TDES. RSA_3072 atau RSA_4096 dapat digunakan saat menukar tombol TDES atau AES-128. API ini akan menghasilkan keypair untuk tujuan impor kunci, menandatangani kunci menggunakan root sertifikat dan mengembalikan sertifikat dan root sertifikat. Pada akhirnya, kunci yang akan diekspor harus dienkripsi menggunakan kunci ini. Perhatikan bahwa sertifikat ini berumur pendek dan hanya ditujukan untuk tujuan ini.

```
$ aws payment-cryptography get-parameters-for-import --key-material-type
 KEY_CRYPTOGRAM --wrapping-key-algorithm RSA_4096
```

```
{
    "ImportToken": "import-token-bwxli6ocftypneu5",
    "ParametersValidUntilTimestamp": 1698245002.065,
    "WrappingKeyCertificateChain": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0....",
    "WrappingKeyCertificate": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0....",
    "WrappingKeyAlgorithm": "RSA_4096"
}
```

2. Instal sertifikat publik pada sistem sumber utama

   Dengan banyak HSM, Anda mungkin perlu menginstal/memuat/mempercayai sertifikat publik (dan/atau akarnya) yang dihasilkan pada langkah 1 untuk mengekspor kunci menggunakannya.

3. Ekspor kunci dari sistem sumber

   Banyak HSM dan sistem terkait mendukung kemampuan untuk mengekspor kunci menggunakan bungkus RSA. Anda akan ingin menentukan kunci publik dari langkah 1 sebagai sertifikat (enkripsi) (WrappingKeySertifikat). Jika Anda membutuhkan rantai kepercayaan, ini terkandung dalam bidang respons WrappingKeyCertificateChain di langkah #1. Saat mengekspor kunci dari HSM Anda, Anda akan ingin menentukan formatnya menjadi RSA, Mode Padding = PKCS #1 v2.2 OAEP (dengan SHA 256 atau SHA 512).

4. Panggil kunci impor

   Sebagai langkah terakhir, Anda akan memanggil ImportKey API dengan file KeyMaterialType . `KeyMaterial` Anda akan memerlukan token impor dari langkah 1 dan `key-material` (bahan kunci yang dibungkus) dari langkah 3. Anda harus memberikan parameter kunci (seperti Key Usage) karena RSA wrap tidak menggunakan blok kunci.

```
$ cat import-key-cryptogram.json
              {
                  "KeyMaterial": {
                      "KeyCryptogram": {
                          "Exportable": true,
                          "ImportToken": "import-token-bwxli6ocftypneu5",
                          "KeyAttributes": {
                              "KeyAlgorithm": "AES_128",
                              "KeyClass": "SYMMETRIC_KEY",
                              "KeyModesOfUse": {
                                  "Decrypt": true,
                                  "DeriveKey": false,
                                  "Encrypt": true,
                                  "Generate": false,
                                  "NoRestrictions": false,
                                  "Sign": false,
                                  "Unwrap": true,
                                  "Verify": false,
                                  "Wrap": true
                              },
                              "KeyUsage": "TR31_K0_KEY_ENCRYPTION_KEY"
                          },
```

```
                        "WrappedKeyCryptogram": "18874746731....",
                        "WrappingSpec": "RSA_OAEP_SHA_256"
                    }
                }
            }
```

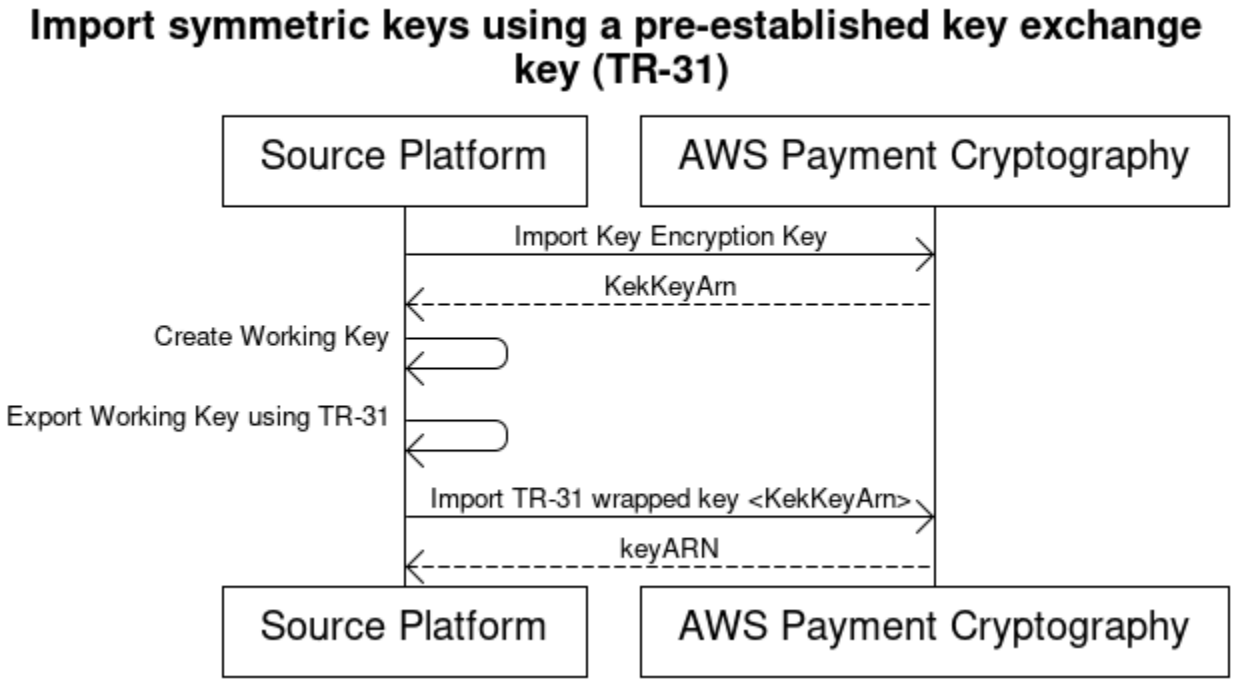
```
$ aws payment-cryptography import-key --cli-input-json file://import-key-
cryptogram.json
```

```
{
      "Key": {
          "KeyOrigin": "EXTERNAL",
          "Exportable": true,
          "KeyCheckValue": "DA1ACF",
          "UsageStartTimestamp": 1697643478.92,
          "Enabled": true,
          "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h",
          "CreateTimestamp": 1697643478.92,
          "KeyState": "CREATE_COMPLETE",
          "KeyAttributes": {
              "KeyAlgorithm": "AES_128",
              "KeyModesOfUse": {
                  "Encrypt": true,
                  "Unwrap": true,
                  "Verify": false,
                  "DeriveKey": false,
                  "Decrypt": true,
                  "NoRestrictions": false,
                  "Sign": false,
                  "Wrap": true,
                  "Generate": false
              },
              "KeyUsage": "TR31_K0_KEY_ENCRYPTION_KEY",
              "KeyClass": "SYMMETRIC_KEY"
          },
          "KeyCheckValueAlgorithm": "CMAC"
      }
  }
```

5. Gunakan kunci yang diimpor untuk operasi kriptografi atau impor berikutnya

   Jika yang diimpor KeyUsage adalah TR31_K0_KEY_ENCRYPTION_KEY, maka kunci ini dapat digunakan untuk impor kunci berikutnya menggunakan TR-31. Jika jenis kunci adalah jenis lain (seperti TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY), maka kunci tersebut dapat langsung digunakan untuk operasi kriptografi.

Impor kunci simetris menggunakan kunci pertukaran kunci yang telah ditetapkan sebelumnya (TR-31)



Ketika mitra bertukar beberapa kunci (atau untuk mendukung rotasi kunci), biasanya untuk pertama menukar kunci enkripsi kunci awal (KEK) menggunakan teknik seperti komponen kunci kertas atau dalam kasus Kriptografi AWS Pembayaran menggunakan TR-34.

Setelah KEK dibuat, Anda dapat menggunakan kunci ini untuk mengangkut kunci berikutnya (termasuk KEK lainnya). AWS Kriptografi Pembayaran mendukung pertukaran kunci semacam ini menggunakan ANSI TR-31 yang banyak digunakan dan didukung secara luas oleh vendor HSM.

1. Kunci Enkripsi Kunci Impor (KEK)

   Diasumsikan bahwa Anda telah mengimpor KEK Anda dan memiliki keYarn (atau KeyAlias) yang tersedia untuk Anda.

2. Buat kunci pada platform sumber

   Jika kunci belum ada, buat kunci di platform sumber. Sebaliknya, Anda dapat membuat kunci pada Kriptografi AWS Pembayaran dan menggunakan `export` perintah sebagai gantinya.

3. Ekspor kunci dari platform sumber

   Saat mengekspor, pastikan Anda menentukan format ekspor sebagai TR-31. Platform sumber juga akan meminta Anda untuk kunci yang akan diekspor dan kunci enkripsi kunci untuk digunakan.

4. Impor ke Kriptografi AWS Pembayaran

   Saat memanggil perintah ImportKey, WrappingKeyIdentifier harus menjadi keYarn (atau alias) dari kunci enkripsi kunci Anda dan WrappedKeyBlock merupakan output dari platform sumber.

   Example

```
$ aws payment-cryptography import-key \
            --key-material="Tr31KeyBlock={WrappingKeyIdentifier="arn:aws:payment-
cryptography:us-east-2:111122223333:key/ov6icy4ryas4zcza",\

 WrappedKeyBlock="D0112B0AX00E00002E0A3D58252CB67564853373D1EBCC1E23B2ADE7B15E967CC27B85D599
```

```
      {
        "Key": {
            "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h",
            "KeyAttributes": {
                "KeyUsage": "TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY",
                "KeyClass": "SYMMETRIC_KEY",
                "KeyAlgorithm": "AES_128",
                "KeyModesOfUse": {
                    "Encrypt": true,
                    "Decrypt": true,
                    "Wrap": true,
                    "Unwrap": true,
                    "Generate": false,
                    "Sign": false,
                    "Verify": false,
                    "DeriveKey": false,
                    "NoRestrictions": false
```

```
                }
            },
            "KeyCheckValue": "0A3674",
            "KeyCheckValueAlgorithm": "CMAC",
            "Enabled": true,
            "Exportable": true,
            "KeyState": "CREATE_COMPLETE",
            "KeyOrigin": "EXTERNAL",
            "CreateTimestamp": "2023-06-02T07:38:14.913000-07:00",
            "UsageStartTimestamp": "2023-06-02T07:38:14.857000-07:00"
        }
    }
```

Mengimpor kunci asimetris (RSA)

Mengimpor kunci publik RSA

AWS Kriptografi Pembayaran mendukung impor kunci RSA publik dalam bentuk sertifikat X.509. Untuk mengimpor sertifikat, Anda harus terlebih dahulu mengimpor sertifikat akarnya. Semua sertifikat harus tidak kedaluwarsa pada saat impor. Sertifikat harus dalam format PEM dan harus dikodekan base64.

1. Impor ke Sertifikat Root ke Kriptografi AWS Pembayaran

   Example

```
$ aws payment-cryptography import-key \
  --key-material='{"RootCertificatePublicKey":{"KeyAttributes":
{"KeyAlgorithm":"RSA_2048", \
  "KeyClass":"PUBLIC_KEY", "KeyModesOfUse":{"Verify":
 true},"KeyUsage":"TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE"}, \

 "PublicKeyCertificate":"LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURKVENDQWcyZ0F3SUJBZ0lCWkR
```

```
  {
    "Key": {
        "CreateTimestamp": "2023-08-08T18:52:01.023000+00:00",
        "Enabled": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
zabouwe3574jysdl",
```

```
        "KeyAttributes": {
            "KeyAlgorithm": "RSA_2048",
            "KeyClass": "PUBLIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": false,
                "DeriveKey": false,
                "Encrypt": false,
                "Generate": false,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": false,
                "Verify": true,
                "Wrap": false
            },
            "KeyUsage": "TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE"
        },
        "KeyOrigin": "EXTERNAL",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "2023-08-08T18:52:01.023000+00:00"
    }
}
```

2. Impor Sertifikat Kunci Publik ke Kriptografi AWS Pembayaran

   Anda sekarang dapat mengimpor kunci publik. Ada dua opsi untuk mengimpor kunci publik. `TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE`dapat digunakan jika tujuan kuncinya adalah untuk memverifikasi tanda tangan (misalnya saat mengimpor menggunakan TR-34). `TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION`dapat digunakan saat mengenkripsi data yang dimaksudkan untuk digunakan dengan sistem lain.

   Example

```
$ aws payment-cryptography import-key \
    --key-material='{"TrustedCertificatePublicKey":
{"CertificateAuthorityPublicKeyIdentifier":"arn:aws:payment-cryptography:us-
east-2:111122223333:key/zabouwe3574jysdl", \
    "KeyAttributes":
{"KeyAlgorithm":"RSA_2048","KeyClass":"PUBLIC_KEY","KeyModesOfUse":
{"Verify":true},"KeyUsage":"TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE"},\
    "PublicKeyCertificate":"LS0tLS1CRUdJTiB..."}}'
```

```
{
  "Key": {
      "CreateTimestamp": "2023-08-08T18:55:46.815000+00:00",
      "Enabled": true,
      "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/4kd6xud22e64wcbk",
      "KeyAttributes": {
          "KeyAlgorithm": "RSA_4096",
          "KeyClass": "PUBLIC_KEY",
          "KeyModesOfUse": {
              "Decrypt": false,
              "DeriveKey": false,
              "Encrypt": false,
              "Generate": false,
              "NoRestrictions": false,
              "Sign": false,
              "Unwrap": false,
              "Verify": true,
              "Wrap": false
          },
          "KeyUsage": "TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE"
      },
      "KeyOrigin": "EXTERNAL",
      "KeyState": "CREATE_COMPLETE",
      "UsageStartTimestamp": "2023-08-08T18:55:46.815000+00:00"
  }
}
```


Kunci ekspor

Topik

- Mengekspor kunci simetris
- Mengekspor kunci asimetris (RSA)

Mengekspor kunci simetris

> **Important**
>
> Contoh mungkin memerlukan AWS CLI V2 versi terbaru. Sebelum memulai, pastikan Anda telah meningkatkan ke versi terbaru.

Topik

Kunci ekspor menggunakan teknik asimetris (TR-34)

Tinjauan: TR-34 menggunakan kriptografi asimetris RSA untuk mengenkripsi kunci simetris untuk pertukaran serta memastikan sumber data (penandatanganan). Ini memastikan kerahasiaan (enkripsi) dan integritas (tanda tangan) dari kunci yang dibungkus. Saat mengekspor, Kriptografi AWS Pembayaran menjadi host distribusi utama (KDH) dan sistem target menjadi perangkat penerima kunci (KRD).

1. Panggil perintah inisialisasi ekspor

   Panggilan `get-parameters-for-export` untuk menginisialisasi proses ekspor. API ini akan menghasilkan keypair untuk tujuan ekspor kunci, menandatangani kunci dan mengembalikan sertifikat dan root sertifikat. Pada akhirnya, kunci pribadi yang dihasilkan oleh perintah ini digunakan untuk menandatangani payload ekspor. Dalam terminologi TR-34, ini dikenal sebagai sertifikat penandatanganan KDH. Perhatikan bahwa sertifikat ini berumur pendek dan hanya ditujukan untuk tujuan ini. Parameter ParametersValidUntilTimestamp menentukan durasinya.

   CATATAN: Semua sertifikat dikembalikan dalam format yang dikodekan base64

   Example

```
$ aws payment-cryptography get-parameters-for-export \
```

```
                    --signing-key-algorithm RSA_2048 --key-material-type
 TR34_KEY_BLOCK
```

```
{
              "SigningKeyCertificate":
 "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2RENDQXFTZ0F3SUJBZ0lRZFAzSzNHNEFKT0I4WTNpTmUvYl
              "SigningKeyCertificateChain":
 "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUY0VENDQThtZ0F3SUJBZ0lSQUtlN2piaHFKZjJPd3FGUWI5c3
              "SigningKeyAlgorithm": "RSA_2048",
              "ExportToken": "export-token-au7pvkbsq4mbup6i",
              "ParametersValidUntilTimestamp": "2023-06-13T15:40:24.036000-07:00"
}
```

2. Impor sertifikat Kriptografi AWS Pembayaran ke sistem penerima

   Impor rantai sertifikat yang disediakan pada langkah 1 ke sistem penerima Anda seperlunya.

3. Buat key pair, buat sertifikat publik dan berikan root sertifikat ke AWS Payment Cryptography

   Untuk memastikan kerahasiaan muatan yang ditransmisikan, itu dienkripsi oleh pihak pengirim (dikenal sebagai Host Distribusi Utama atau KDH). Pihak penerima (biasanya HSM Anda atau HSM mitra Anda) akan ingin menghasilkan kunci publik untuk tujuan ini dan kemudian membuat sertifikat kunci publik (x.509) yang dapat diberikan kembali ke Kriptografi Pembayaran. AWS AWS Private CA adalah salah satu opsi untuk menghasilkan sertifikat, tetapi tidak ada batasan pada otoritas sertifikat yang digunakan.

   Setelah Anda memiliki sertifikat, Anda akan ingin memuat sertifikat root ke Kriptografi AWS Pembayaran menggunakan `ImportKey` perintah dan KeyMaterialType dari `ROOT_PUBLIC_KEY_CERTIFICATE` dan KeyUsageType dari. `TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE`

   Sertifikat ini adalah TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE karena merupakan kunci root dan digunakan untuk menandatangani sertifikat daun. KeyUsageType Sertifikat daun untuk impor/ekspor tidak diimpor ke Kriptografi AWS Pembayaran tetapi diteruskan secara inline.

   > **Note**
   >
   > Jika sertifikat root sebelumnya diimpor, langkah ini dapat dilewati.

4. Panggil kunci Ekspor

   Sebagai langkah terakhir, Anda akan memanggil ExportKey API dengan KeyMaterialType aTR34_KEY_BLOCK. `certificate-authority-public-key-identifier`Akan menjadi keYarn dari impor CA root pada langkah 3, `WrappingKeyCertificate` akan menjadi sertifikat daun dari langkah 3 dan `export-key-identifier` merupakan keYarn (atau alias) yang akan diekspor. Anda juga perlu memberikan token ekspor dari langkah 1.

Kunci ekspor menggunakan teknik asimetris (RSA Wrap)

Ikhtisar: Kriptografi AWS Pembayaran mendukung pembungkus/buka RSA untuk pertukaran kunci ketika TR-34 bukan merupakan opsi yang tersedia oleh pihak lawan. Mirip dengan TR-34, teknik ini menggunakan kriptografi asimetris RSA untuk mengenkripsi kunci simetris untuk pertukaran. Namun, tidak seperti TR-34, metode ini tidak memiliki muatan yang ditandatangani oleh pihak pengirim. Juga, teknik pembungkus RSA ini tidak termasuk blok kunci yang digunakan untuk menjaga integritas metadata kunci selama transportasi.

> **Note**
>
> Bungkus RSA dapat digunakan untuk mengekspor kunci TDES dan AES-128.

1. Menghasilkan kunci RSA dan sertifikat pada sistem penerima

   Buat (atau identifikasi) kunci RSA yang akan digunakan untuk menerima kunci yang dibungkus. AWS Kriptografi Pembayaran mengharapkan kunci dalam format sertifikat X.509. Sertifikat harus ditandatangani oleh sertifikat root yang diimpor (atau dapat diimpor) ke dalam Kriptografi AWS Pembayaran.
2. Instal sertifikat publik root pada Kriptografi AWS Pembayaran

```
$ aws payment-cryptography  import-key --key-material='{"RootCertificatePublicKey":
{"KeyAttributes":{"KeyAlgorithm":"RSA_4096","KeyClass":"PUBLIC_KEY","KeyModesOfUse":
{"Verify":
 true},"KeyUsage":"TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE"},"PublicKeyCertificate":"LS
```

```
{
    "Key": {
    "CreateTimestamp": "2023-09-14T10:50:32.365000-07:00",
```

```
        "Enabled": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
nsq2i3mbg6sn775f",
        "KeyAttributes": {
        "KeyAlgorithm": "RSA_4096",
        "KeyClass": "PUBLIC_KEY",
        "KeyModesOfUse": {
        "Decrypt": false,
        "DeriveKey": false,
        "Encrypt": false,
        "Generate": false,
        "NoRestrictions": false,
        "Sign": false,
        "Unwrap": false,
        "Verify": true,
        "Wrap": false
        },
        "KeyUsage": "TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE"
        },
        "KeyOrigin": "EXTERNAL",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "2023-09-14T10:50:32.365000-07:00“
        }
       }
```

3. Panggil kunci ekspor

Selanjutnya Anda ingin menginstruksikan Kriptografi AWS Pembayaran untuk mengekspor kunci Anda menggunakan sertifikat daun Anda. Anda akan menentukan ARN untuk sertifikat root yang diimpor sebelumnya, sertifikat daun yang akan digunakan untuk ekspor dan kunci simetris untuk mengekspor. Outputnya akan menjadi versi hex yang dikodekan biner dibungkus (dienkripsi) dari kunci simetris Anda.

```
$ cat export-key.json
```

```
  {
      "ExportKeyIdentifier": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/tqv5yij6wtxx64pi",
      "KeyMaterial": {
          "KeyCryptogram": {
```

```
            "CertificateAuthorityPublicKeyIdentifier": "arn:aws:payment-
cryptography:us-east-2:111122223333:key/zabouwe3574jysdl",
            "WrappingKeyCertificate": "LS0tLS1CRUdJTiBD...",
            "WrappingSpec": "RSA_OAEP_SHA_256"
        }
    }
}
```

```
$ aws payment-cryptography export-key --cli-input-json file://export-key.json
```

```
{
    "WrappedKey": {
        "KeyMaterial":
"18874746731E9E1C4562E4116D1C2477063FCB08454D757D81854AEAEE0A52B1F9D303FA29C02DC82AE7785353
        "WrappedKeyMaterialFormat": "KEY_CRYPTOGRAM"
    }
}
```

4. Kunci impor ke sistem penerima

   Banyak HSM dan sistem terkait mendukung kemampuan untuk mengimpor kunci menggunakan RSA unwrap (termasuk Kriptografi AWS Pembayaran). Untuk melakukannya, tentukan kunci publik dari langkah 1 sebagai sertifikat (enkripsi). Dan formatnya harus ditentukan sebagai RSA, Mode Padding = PKCS #1 v2.2 OAEP (dengan SHA 256). Terminologi yang tepat dapat bervariasi menurut HSM.

   > **Note**
   >
   > AWS Kriptografi Pembayaran mengeluarkan kunci yang dibungkus di HexBinary. Anda mungkin perlu mengonversi format sebelum mengimpor jika sistem Anda memerlukan representasi biner yang berbeda seperti base64.

Ekspor kunci simetris menggunakan kunci pertukaran kunci yang telah ditetapkan sebelumnya (TR-31)

Ketika mitra bertukar beberapa kunci (atau untuk mendukung rotasi kunci), biasanya untuk pertama menukar kunci enkripsi kunci awal (KEK) menggunakan teknik seperti komponen kunci kertas atau dalam kasus Kriptografi AWS Pembayaran menggunakan TR-34. Setelah KEK dibuat, Anda dapat menggunakan kunci ini untuk mengangkut kunci berikutnya (termasuk KEK lainnya). AWS Kriptografi

Pembayaran mendukung pertukaran kunci semacam ini menggunakan ANSI TR-31 yang banyak digunakan dan didukung secara luas oleh vendor HSM.

1. Kunci Enkripsi Kunci Pertukaran (KEK)

   Diasumsikan bahwa Anda telah menukar KEK Anda dan memiliki keYarn (atau KeyAlias) yang tersedia untuk Anda.

2. Buat kunci pada Kriptografi AWS Pembayaran

   Jika kunci belum ada, buat kuncinya. Sebaliknya, Anda dapat membuat kunci pada sistem lain dan menggunakan perintah impor sebagai gantinya.

3. Ekspor kunci dari Kriptografi AWS Pembayaran

   Saat mengekspor, formatnya akan menjadi TR-31. Saat memanggil API, Anda akan menentukan kunci yang akan diekspor dan kunci pembungkus yang akan digunakan.

```
$ aws payment-cryptography export-key --key-material='{"Tr31KeyBlock":
 {"WrappingKeyIdentifier": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ov6icy4ryas4zcza"}}' --export-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/5rplquuwozodpwsp
```

```
{
             "WrappedKey": {
                 "KeyCheckValue": "73C263",
                 "KeyCheckValueAlgorithm": "ANSI_X9_24",
                 "KeyMaterial":
 "D0144K0AB00E0000A24D3ACF3005F30A6E31D533E07F2E1B17A2A003B338B1E79E5B3AD4FBF7850FACF9A37844
                 "WrappedKeyMaterialFormat": "TR31_KEY_BLOCK"
             }
     }
```

4. Impor ke sistem Anda

   Anda atau mitra Anda akan menggunakan implementasi kunci impor pada sistem Anda untuk mengimpor kunci.

Kunci Awal DUKPT Ekspor (IPEK/IK)

Saat menggunakan DUKPT, satu Kunci Derivasi Dasar (BDK) dapat dibuat untuk armada terminal. Terminal, bagaimanapun, tidak pernah memiliki akses ke BDK asli tetapi masing-masing disuntikkan

dengan kunci terminal awal yang unik yang dikenal sebagai IPEK atau Initial Key (IK). Setiap IPEK adalah kunci yang berasal dari BDK dan dimaksudkan untuk menjadi unik per terminal tetapi berasal dari BDK asli. Data derivasi untuk perhitungan ini dikenal sebagai Key Serial Number (KSN). Per X9.24, untuk TDES 10 byte KSN biasanya terdiri dari 24 bit untuk Key Set ID, 19 bit untuk terminal ID dan 21 bit untuk counter transaksi. Untuk AES, KSN 12 byte biasanya terdiri dari 32 bit untuk ID BDK, 32 bit untuk pengenal derivasi (ID) dan 32 bit untuk penghitung transaksi.

AWS Kriptografi Pembayaran menyediakan mekanisme untuk menghasilkan dan mengekspor kunci awal ini. Setelah dihasilkan, kunci ini dapat diekspor menggunakan metode bungkus TR-31, TR-34 dan RSA. Kunci IPEK tidak bertahan dan tidak dapat digunakan untuk operasi selanjutnya pada AWS Kriptografi Pembayaran

AWS Kriptografi Pembayaran tidak memberlakukan pemisahan antara dua bagian pertama dari KSN. Jika Anda ingin menyimpan pengenal derivasi bersama dengan BDK, Anda dapat menggunakan fitur AWS tag untuk tujuan ini.

> **Note**
>
> Bagian counter dari KSN (32 bit untuk AES DUKPT) tidak digunakan untuk derivasi IPEK/IK. Oleh karena itu, input 12345678901234560001 dan 12345678901234569999 akan menghasilkan IPEK yang sama.

```
$ aws payment-cryptography export-key --key-material='{"Tr31KeyBlock":
 {"WrappingKeyIdentifier": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ov6icy4ryas4zcza"}}' --export-key-identifier arn:aws:payment-
cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi --export-attributes
 'ExportDukptInitialKey={KeySerialNumber=12345678901234560001}'
```

```
{
    "WrappedKey": {
        "KeyCheckValue": "73C263",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyMaterial":
 "B0096B1TX00S000038A8A06588B9011F0D5EEF1CCAECFA6962647A89195B7A98BDA65DDE7C57FEA507559AF2A5D60
        "WrappedKeyMaterialFormat": "TR31_KEY_BLOCK"
    }
}
```

Mengekspor kunci asimetris (RSA)

Panggilan `get-public-key-certificate` untuk mengekspor kunci publik dalam bentuk sertifikat. API ini akan mengekspor sertifikat serta sertifikat root yang dikodekan dalam format base64.

CATATAN: API ini tidak idempoten - panggilan berikutnya dapat menghasilkan sertifikat yang berbeda meskipun kunci yang mendasarinya sama.

Example

```
$ aws payment-cryptography get-public-key-certificate \
        –key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/5dza7xqd6soanjtb
```

```
{
        "KeyCertificate": "LS0tLS1CRUdJTi...",
        "KeyCertificateChain": "LS0tLS1CRUdJT...“
}
```

Menggunakan alias

Alias adalah nama yang ramah untuk kunci Kriptografi AWS Pembayaran. Misalnya, alias memungkinkan Anda merujuk ke kunci sebagai `alias/test-key` ganti. `arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h`

Anda dapat menggunakan alias untuk mengidentifikasi kunci di sebagian besar operasi manajemen kunci (bidang kontrol), dan dalam operasi kriptografi (dataplane).

Anda juga dapat mengizinkan dan menolak akses ke kunci Kriptografi AWS Pembayaran berdasarkan alias mereka tanpa mengedit kebijakan atau mengelola hibah. Fitur ini merupakan bagian dari dukungan layanan untuk kontrol akses berbasis atribut (ABAC).

Sebagian besar kekuatan alias berasal dari kemampuan Anda untuk mengubah kunci yang terkait dengan alias kapan saja. Alias dapat membuat kode Anda lebih mudah ditulis dan dipelihara. Misalnya, Anda menggunakan alias untuk merujuk ke kunci Kriptografi AWS Pembayaran tertentu dan Anda ingin mengubah kunci Kriptografi AWS Pembayaran. Dalam hal ini, cukup kaitkan alias dengan kunci yang berbeda. Anda tidak perlu mengubah kode atau konfigurasi aplikasi Anda.

Alias juga mempermudah menggunakan kembali kode yang sama di Wilayah AWS berbeda. Buat alias dengan nama yang sama di beberapa Wilayah dan kaitkan setiap alias dengan kunci Kriptografi AWS Pembayaran di Wilayahnya. Ketika kode berjalan di setiap Wilayah, alias mengacu pada kunci Kriptografi AWS Pembayaran terkait di Wilayah tersebut.

Anda dapat membuat alias untuk kunci Kriptografi AWS Pembayaran dengan menggunakan API. `CreateAlias`

API Kriptografi AWS Pembayaran memberikan kontrol penuh atas alias di setiap akun dan Wilayah. API mencakup operasi untuk membuat alias (CreateAlias), melihat nama alias dan keYarn yang ditautkan (list-alias), mengubah kunci Kriptografi AWS Pembayaran yang terkait dengan alias (update-alias), dan menghapus alias (delete-alias).

Topik

Tentang alias

Pelajari cara kerja alias dalam Kriptografi AWS Pembayaran.

Alias adalah sumber daya independen AWS

> Alias bukan milik kunci Kriptografi AWS Pembayaran. Tindakan yang Anda lakukan pada alias tidak memengaruhi kunci terkaitnya. Anda dapat membuat alias untuk kunci Kriptografi AWS Pembayaran dan kemudian memperbarui alias sehingga dikaitkan dengan kunci Kriptografi AWS Pembayaran yang berbeda. Anda bahkan dapat menghapus alias tanpa efek apa pun pada kunci Kriptografi AWS Pembayaran terkait. Jika Anda menghapus kunci Kriptografi AWS Pembayaran, semua alias yang terkait dengan kunci tersebut akan menjadi tidak ditetapkan.
>
> Jika Anda menentukan alias sebagai sumber daya dalam kebijakan IAM, kebijakan mengacu pada alias, bukan ke kunci Kriptografi AWS Pembayaran terkait.

Setiap alias memiliki nama yang ramah

> Saat Anda membuat alias, Anda menentukan nama alias yang diawali oleh. `alias/` Misalnya `alias/test_1234`

Setiap alias dikaitkan dengan satu kunci Kriptografi AWS Pembayaran pada satu waktu

Alias dan kunci Kriptografi AWS Pembayarannya harus berada di akun dan Wilayah yang sama.

Kunci Kriptografi AWS Pembayaran dapat dikaitkan dengan lebih dari satu alias secara bersamaan, tetapi setiap alias hanya dapat dipetakan ke satu kunci

Misalnya, `list-aliases` output ini menunjukkan bahwa `alias/sampleAlias1` alias dikaitkan dengan tepat satu kunci Kriptografi AWS Pembayaran target, yang diwakili oleh properti. `KeyArn`

```
$ aws payment-cryptography list-aliases
```

```
  {
    "Aliases": [
        {
            "AliasName": "alias/sampleAlias1",
            "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h"
        }
    ]
}
```

Beberapa alias dapat dikaitkan dengan kunci Kriptografi AWS Pembayaran yang sama

Misalnya, Anda dapat mengaitkan `alias/sampleAlias1;` dan `alias/sampleAlias2` alias dengan kunci yang sama.

```
$ aws payment-cryptography list-aliases
```

```
{
        "Aliases": [
            {
                "AliasName": "alias/sampleAlias1",
                "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h"
```

```
            },
            {
                "AliasName": "alias/sampleAlias2",
                "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h"
            }
        ]
    }
```

Alias harus unik untuk akun dan Wilayah tertentu

Misalnya, Anda hanya dapat memiliki satu alias `alias/sampleAlias1` di setiap akun dan Wilayah. Alias peka huruf besar/kecil, tetapi kami merekomendasikan untuk tidak menggunakan alias yang hanya berbeda dalam kapitalisasi karena dapat rentan terhadap kesalahan. Anda tidak dapat mengubah nama alias. Namun, Anda dapat menghapus alias dan membuat alias baru dengan nama yang diinginkan.

Anda dapat membuat alias dengan nama yang sama di Wilayah yang berbeda

Misalnya, Anda dapat memiliki alias `alias/sampleAlias2` di AS Timur (Virginia N.) dan alias `alias/sampleAlias2` di AS Barat (Oregon). Setiap alias akan dikaitkan dengan kunci Kriptografi AWS Pembayaran di Wilayahnya. Jika kode Anda merujuk pada nama alias seperti `alias/finance-key`, Anda dapat menjalankannya di beberapa Wilayah. Di setiap Wilayah, ia menggunakan alias/SampleAlias2 yang berbeda. Lihat perinciannya di Menggunakan alias dalam aplikasi Anda.

Anda dapat mengubah kunci Kriptografi AWS Pembayaran yang terkait dengan alias

Anda dapat menggunakan `UpdateAlias` operasi untuk mengaitkan alias dengan kunci Kriptografi AWS Pembayaran yang berbeda. Misalnya, jika `alias/sampleAlias2` alias dikaitkan dengan kunci Kriptografi `arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h` AWS Pembayaran, Anda dapat memperbaruinya sehingga dikaitkan dengan kunci. `arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi`

> **Warning**
>
> AWS Kriptografi Pembayaran tidak memvalidasi bahwa kunci lama dan baru memiliki semua atribut yang sama seperti penggunaan kunci. Memperbarui dengan jenis kunci yang berbeda dapat mengakibatkan masalah dalam aplikasi Anda.

Beberapa kunci tidak memiliki alias

Alias adalah fitur opsional dan tidak semua kunci akan memiliki alias kecuali Anda memilih untuk mengoperasikan lingkungan Anda dengan cara ini. Kunci dapat dikaitkan dengan Alias menggunakan `create-alias` perintah. Selain itu, Anda dapat menggunakan operasi update-alias untuk mengubah kunci Kriptografi AWS Pembayaran yang terkait dengan alias dan operasi hapus-alias untuk menghapus alias. Akibatnya, beberapa kunci Kriptografi AWS Pembayaran mungkin memiliki beberapa alias, dan beberapa mungkin tidak memilikinya.

Memetakan kunci ke alias

Anda dapat memetakan kunci (diwakili oleh ARN) ke satu atau lebih alias menggunakan perintah. `create-alias` Perintah ini tidak idempoten - untuk memperbarui alias, gunakan perintah update-alias.

```
$ aws payment-cryptography create-alias --alias-name alias/sampleAlias1 \
             --key-arn arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h
```

```
{
    "Alias": {
        "AliasName": "alias/alias/sampleAlias1",
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h"
    }
}
```

Menggunakan alias dalam aplikasi Anda

Anda dapat menggunakan alias untuk mewakili kunci Kriptografi AWS Pembayaran dalam kode aplikasi Anda. `key-identifier`Parameter dalam operasi data Kriptografi AWS Pembayaran serta operasi lain seperti List Keys menerima nama alias atau alias ARN.

```
$ aws payment-cryptography-data generate-card-validation-data --key-identifier alias/
BIN_123456_CVK --primary-account-number=171234567890123 --generation-attributes
 CardVerificationValue2={CardExpiryDate=0123}
```

Saat menggunakan alias ARN, ingatlah bahwa alias pemetaan ke AWS kunci Kriptografi Pembayaran didefinisikan dalam akun yang memiliki kunci Kriptografi Pembayaran dan mungkin AWS berbeda di setiap Wilayah.

Salah satu penggunaan alias yang paling kuat adalah pada aplikasi yang berjalan dalam beberapa Wilayah AWS.

Anda dapat membuat versi aplikasi yang berbeda di setiap Wilayah atau menggunakan kamus, konfigurasi, atau pernyataan sakelar untuk memilih kunci Kriptografi AWS Pembayaran yang tepat untuk setiap Wilayah. Tetapi mungkin lebih mudah untuk membuat alias dengan nama alias yang sama di setiap Wilayah. Ingat bahwa nama alias peka terhadap huruf besar-kecil.

API Terkait

Tanda

Tag adalah pasangan kunci dan nilai yang bertindak sebagai metadata untuk mengatur kunci Kriptografi AWS Pembayaran Anda. Mereka dapat digunakan untuk mengidentifikasi kunci secara fleksibel atau mengelompokkan satu atau lebih kunci bersama-sama.

Dapatkan kunci

Kunci Kriptografi AWS Pembayaran mewakili satu unit bahan kriptografi dan hanya dapat digunakan untuk operasi kriptografi untuk layanan ini. GetKeys API mengambil input KeyIdentifier as dan mengembalikan atribut kunci yang tidak dapat diubah dan dapat diubah tetapi tidak mengandung materi kriptografi apa pun.

Example

```
$ aws payment-cryptography get-key --key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/kwapwa6qaifllw2h
```

```
{
  "Key": {
      "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h",
      "KeyAttributes": {
          "KeyUsage": "TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY",
          "KeyClass": "SYMMETRIC_KEY",
          "KeyAlgorithm": "AES_128",
```

```
            "KeyModesOfUse": {
                "Encrypt": true,
                "Decrypt": true,
                "Wrap": true,
                "Unwrap": true,
                "Generate": false,
                "Sign": false,
                "Verify": false,
                "DeriveKey": false,
                "NoRestrictions": false
            }
        },
        "KeyCheckValue": "0A3674",
        "KeyCheckValueAlgorithm": "CMAC",
        "Enabled": true,
        "Exportable": true,
        "KeyState": "CREATE_COMPLETE",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "CreateTimestamp": "2023-06-02T07:38:14.913000-07:00",
        "UsageStartTimestamp": "2023-06-02T07:38:14.857000-07:00"
    }
}
```

Dapatkan kunci publik/sertifikat yang terkait dengan key pair

Dapatkan Kunci Publik/Sertifikat mengembalikan kunci publik yang ditunjukkan oleh. `KeyArn` Ini bisa menjadi bagian kunci publik dari key pair yang dihasilkan pada AWS Payment Cryptography atau public key yang sebelumnya diimpor. Kasus penggunaan yang paling umum adalah menyediakan kunci publik ke layanan luar yang akan mengenkripsi data. Data tersebut kemudian dapat diteruskan ke aplikasi yang memanfaatkan Kriptografi AWS Pembayaran dan data dapat didekripsi menggunakan kunci pribadi yang diamankan dalam Kriptografi Pembayaran. AWS

Layanan mengembalikan kunci publik sebagai sertifikat publik. Hasil API berisi CA dan sertifikat kunci publik. Kedua elemen data dikodekan base64.

> **Note**
>
> Sertifikat publik yang dikembalikan dimaksudkan untuk berumur pendek dan tidak dimaksudkan untuk menjadi idempoten. Anda mungkin menerima sertifikat yang berbeda pada setiap panggilan API bahkan kunci publik itu sendiri tidak berubah.

Example

```
$ aws payment-cryptography get-public-key-certificate  --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/nsq2i3mbg6sn775f
```

```
{
  "KeyCertificate":
 "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2VENDQXFXZ0F3SUJBZ0lSQUo1OWd2VkpDd3dlYldMNldYZEpYY
  "KeyCertificateChain":
 "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUY0VENDQThtZ0F3SUJBZ0lSQUtlN2piaHFKZjJPd3FGUWI5c3VuC
}
```

Tombol penandaan

Dalam Kriptografi AWS Pembayaran, Anda dapat menambahkan tag ke kunci Kriptografi AWS Pembayaran saat Anda membuat kunci, dan menandai atau menghapus tag kunci yang ada kecuali mereka menunggu penghapusan. Tanda adalah opsional, tetapi tanda bisa sangat berguna.

Untuk informasi umum tentang tag, termasuk praktik terbaik, strategi penandaan, serta format dan sintaks tag, lihat Menandai AWS sumber daya di. Referensi Umum Amazon Web

Topik

- Tentang tag dalam Kriptografi AWS Pembayaran
- Melihat tag kunci di konsol
- Mengelola tag kunci dengan operasi API

- Pengontrolan akses ke tanda
- Menggunakan tag untuk mengontrol akses ke tombol

Tentang tag dalam Kriptografi AWS Pembayaran

Tag adalah label metadata opsional yang dapat Anda tetapkan (atau AWS dapat ditetapkan) ke sumber daya. AWS Setiap tanda terdiri dari kunci tanda dan nilai tanda, keduanya adalah string peka huruf besar/kecil. Nilai tanda bisa berupa string kosong (null). Setiap tag pada sumber daya harus memiliki kunci tag yang berbeda, tetapi Anda dapat menambahkan tag yang sama ke beberapa AWS sumber daya. Setiap sumber daya dapat memiliki hingga 50 tanda yang dibuat pengguna.

Jangan sertakan informasi rahasia atau sensitif dalam kunci tag atau nilai tag. Tag dapat diakses oleh banyak orang Layanan AWS, termasuk penagihan.

Dalam Kriptografi AWS Pembayaran, Anda dapat menambahkan tag ke kunci saat Anda membuat kunci, dan menandai atau menghapus tag kunci yang ada kecuali mereka menunggu penghapusan. Anda tidak dapat menandai alias. Tanda adalah opsional, tetapi tanda bisa sangat berguna.

Misalnya, Anda dapat menambahkan `"Project"="Alpha"` tag ke semua kunci Kriptografi AWS Pembayaran dan bucket Amazon S3 yang Anda gunakan untuk proyek Alpha. Contoh lain adalah menambahkan `"BIN"="20130622"` tag ke semua kunci yang terkait dengan nomor identifikasi bank tertentu (BIN).

```
  [
  {
    "Key": "Project",
    "Value": "Alpha"
  },
  {
    "Key": "BIN",
    "Value": "20130622"
  }
]
```

Untuk informasi umum tentang tag, termasuk format dan sintaks, lihat Menandai AWS sumber daya di. Referensi Umum Amazon Web

Tanda membantu Anda melakukan hal berikut:

- Identifikasi dan atur AWS sumber daya Anda. Banyak AWS layanan mendukung penandaan, sehingga Anda dapat menetapkan tag yang sama ke sumber daya dari layanan yang berbeda untuk menunjukkan bahwa sumber daya terkait. Misalnya, Anda dapat menetapkan tag yang sama ke kunci Kriptografi AWS Pembayaran dan volume atau rahasia Amazon Elastic Block Store (Amazon EBS) EBS). AWS Secrets Manager Anda juga dapat menggunakan tag untuk mengidentifikasi kunci untuk otomatisasi.
- Lacak AWS biaya Anda. Saat Anda menambahkan tag ke AWS sumber daya Anda, AWS buat laporan alokasi biaya dengan penggunaan dan biaya yang dikumpulkan berdasarkan tag. Anda dapat menggunakan fitur ini untuk melacak biaya Kriptografi AWS Pembayaran untuk proyek, aplikasi, atau pusat biaya.

  Untuk informasi selengkapnya tentang penggunaan tanda untuk alokasi biaya, lihat Menggunakan Tanda Alokasi Biaya dalam Panduan Pengguna AWS Billing . Untuk informasi tentang aturan untuk kunci tanda dan nilai tanda, lihat Pembatasan Tanda Ditetapkan Pengguna di AWS Billing Panduan Pengguna.
- Kontrol akses ke AWS sumber daya Anda. Mengizinkan dan menolak akses ke kunci berdasarkan tag mereka adalah bagian dari dukungan Kriptografi AWS Pembayaran untuk kontrol akses berbasis atribut (ABAC). Untuk informasi tentang mengendalikan akses ke Kriptografi AWS Pembayaran berdasarkan tag mereka, lihatOtorisasi berdasarkan tag Kriptografi AWS Pembayaran. Untuk informasi umum selengkapnya tentang penggunaan tag untuk mengontrol akses ke AWS sumber daya, lihat Mengontrol Akses ke AWS Sumber Daya Menggunakan Tag Sumber Daya di Panduan Pengguna IAM.

AWS Kriptografi Pembayaran menulis entri ke AWS CloudTrail log Anda saat Anda menggunakan TagResource, UntagResource, atau ListTagsForResource operasi.

Melihat tag kunci di konsol

Untuk melihat tag di konsol, Anda memerlukan izin penandaan pada kunci dari kebijakan IAM yang menyertakan kunci. Anda memerlukan izin ini selain izin untuk melihat kunci di konsol.

Mengelola tag kunci dengan operasi API

Anda dapat menggunakan AWS Payment Cryptography API untuk menambahkan, menghapus, dan mencantumkan tag untuk kunci yang Anda kelola. Contoh-contoh ini menggunakan AWS Command Line Interface (AWS CLI), tetapi Anda dapat menggunakan bahasa pemrograman yang didukung. Anda tidak dapat menandai Kunci yang dikelola AWS.

Untuk menambahkan, mengedit, melihat, dan menghapus tag untuk kunci, Anda harus memiliki izin yang diperlukan. Lihat perinciannya di Pengontrolan akses ke tanda.

Topik

- CreateKey: Tambahkan tag ke kunci baru
- TagResource: Menambahkan atau mengubah tag untuk kunci
- ListResourceTags: Dapatkan tag untuk kunci
- UntagResource: Hapus tag dari kunci

CreateKey: Tambahkan tag ke kunci baru

Anda dapat menambahkan tag saat membuat kunci. Untuk menentukan tag, gunakan `Tags` parameter CreateKeyoperasi.

Untuk menambahkan tag saat membuat kunci, pemanggil harus memiliki `payment-cryptography:TagResource` izin dalam kebijakan IAM. Minimal, izin harus mencakup semua kunci di akun dan Wilayah. Untuk rincian selengkapnya, lihat Pengontrolan akses ke tanda.

Nilai dari parameter `Tags` dari `CreateKey` adalah kumpulan kunci tanda peka huruf besar/kecil dan pasangan nilai kunci. Setiap tag pada kunci harus memiliki nama tag yang berbeda. Nilai tanda bisa berupa string kosong atau null.

Misalnya, AWS CLI perintah berikut membuat kunci enkripsi simetris dengan `Project:Alpha` tag. Saat menentukan lebih dari satu pasangan nilai kunci, gunakan spasi untuk memisahkan setiap pasangan.

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY, \
        KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY, \
        KeyModesOfUse='{Generate=true,Verify=true}' \
        --tags '[{"Key":"Project","Value":"Alpha"},{"Key":"BIN","Value":"123456"}]'
```

Ketika perintah ini berhasil, ia mengembalikan `Key` objek dengan informasi tentang kunci baru. Namun, `Key` tidak termasuk tanda. Untuk mendapatkan tag, gunakan operasi ListResourceTag.

TagResource: Menambahkan atau mengubah tag untuk kunci

TagResourceOperasi menambahkan satu atau lebih tag ke kunci. Anda tidak dapat menggunakan operasi ini untuk menambah atau mengedit tanda dalam Akun AWS berbeda.

Untuk menambahkan tanda, tentukan kunci tanda baru dan nilai tanda. Untuk mengedit tanda, tentukan kunci tanda yang sudah ada dan nilai tanda baru. Setiap tag pada kunci harus memiliki kunci tag yang berbeda. Nilai tanda bisa berupa string kosong atau null.

Misalnya, perintah berikut menambahkan **`UseCase`** dan **`BIN`** tag ke kunci contoh.

```
$ aws payment-cryptography tag-resource --resource-arn arn:aws:payment-
cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h --tags
 '[{"Key":"UseCase","Value":"Acquiring"},{"Key":"BIN","Value":"123456"}]'
```

Ketika perintah ini berhasil, maka tidak mengembalikan output apa pun. Untuk melihat tag pada kunci, gunakan operasi ListResourceTag.

Anda juga dapat menggunakan TagResource untuk mengubah nilai tanda dari tanda yang ada. Untuk mengganti nilai tanda, tentukan kunci tanda yang sama dengan nilai yang berbeda. Tag yang tidak tercantum dalam perintah modifikasi tidak diubah atau dihapus.

Sebagai contoh, perintah ini mengubah nilai tanda `Project` dari `Alpha` ke `Noe`.

Perintah akan mengembalikan http/200 tanpa konten. Untuk melihat perubahan Anda, gunakan `ListTagsForResource`

```
$ aws payment-cryptography tag-resource --resource-arn arn:aws:payment-cryptography:us-
east-2:111122223333:key/kwapwa6qaifllw2h \
        --tags '[{"Key":"Project","Value":"Noe"}]'
```

ListResourceTags: Dapatkan tag untuk kunci

Operasi ListResourceTag mendapatkan tag untuk kunci. Parameter `ResourceArn` (keYarn atau KeyAlias) diperlukan. Anda tidak dapat menggunakan operasi ini untuk melihat tag pada kunci yang berbeda Akun AWS.

Misalnya, perintah berikut mendapatkan tag untuk kunci contoh.

```
$ aws payment-cryptography list-tags-for-resource --resource-arn arn:aws:payment-
cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h

  {
    "Tags": [
```

```
        {
            "Key": "BIN",
            "Value": "20151120"
        },
        {
            "Key": "Project",
            "Value": "Production"
        }
    ]
}
```

UntagResource: Hapus tag dari kunci

UntagResourceOperasi menghapus tag dari kunci. Untuk mengidentifikasi tanda yang akan dihapus, tentukan kunci tanda. Anda tidak dapat menggunakan operasi ini untuk menghapus tag dari kunci yang berbeda Akun AWS.

Ketika berhasil, operasi `UntagResource` tidak mengembalikan output apa pun. Juga, jika kunci tag yang ditentukan tidak ditemukan pada kunci, itu tidak membuang pengecualian atau mengembalikan respons. Untuk mengonfirmasi bahwa operasi berhasil, gunakan operasi ListResourceTag.

Misalnya, perintah ini menghapus **`Purpose`** tag dan nilainya dari kunci yang ditentukan.

```
$ aws payment-cryptography untag-resource \
        --resource-arn arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h --tag-keys Project
```

Pengontrolan akses ke tanda

Untuk menambah, melihat, dan menghapus tag dengan menggunakan API, prinsipal memerlukan izin penandaan dalam kebijakan IAM.

Anda juga dapat membatasi izin ini dengan menggunakan kunci kondisi AWS global untuk tag. Dalam Kriptografi AWS Pembayaran, kondisi ini dapat mengontrol akses ke operasi penandaan, seperti TagResourcedan. UntagResource

Untuk kebijakan contoh dan informasi lebih lanjut, lihat Mengontrol Akses Berdasarkan Kunci Tanda di Panduan Pengguna IAM.

Izin untuk membuat dan mengelola tanda bekerja sebagai berikut.

pembayaran-kriptografi: TagResource

Memungkinkan prinsipal untuk menambah atau mengedit tanda. Untuk menambahkan tag saat membuat kunci, prinsipal harus memiliki izin dalam kebijakan IAM yang tidak terbatas pada kunci tertentu.

pembayaran-kriptografi: ListTags ForResource

Memungkinkan prinsipal untuk melihat tag pada kunci.

pembayaran-kriptografi: UntagResource

Memungkinkan prinsipal untuk menghapus tag dari kunci.

Izin tanda dalam kebijakan

Anda dapat memberikan izin penandaan dalam kebijakan kunci atau kebijakan IAM. Misalnya, kebijakan kunci contoh berikut memberikan izin penandaan pengguna tertentu pada kunci. Ini memberikan semua pengguna yang dapat mengasumsikan contoh peran Administrator atau Developer izin untuk melihat tanda.

```
{
  "Version": "2012-10-17",
  "Id": "example-key-policy",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {"AWS": "arn:aws:iam::111122223333:root"},
      "Action": "payment-cryptography:*",
      "Resource": "*"
    },
    {
      "Sid": "Allow all tagging permissions",
      "Effect": "Allow",
      "Principal": {"AWS": [
        "arn:aws:iam::111122223333:user/LeadAdmin",
        "arn:aws:iam::111122223333:user/SupportLead"
      ]},
      "Action": [
          "payment-cryptography:TagResource",
          "payment-cryptography:ListTagsForResource",
```

```
            "payment-cryptography:UntagResource"
        ],
        "Resource": "*"
      },
      {
        "Sid": "Allow roles to view tags",
        "Effect": "Allow",
        "Principal": {"AWS": [
          "arn:aws:iam::111122223333:role/Administrator",
          "arn:aws:iam::111122223333:role/Developer"
        ]},
        "Action": "payment-cryptography:ListResourceTags",
        "Resource": "*"
      }
    ]
}
```

Untuk memberikan izin penandaan prinsipal pada beberapa kunci, Anda dapat menggunakan kebijakan IAM. Agar kebijakan ini efektif, kebijakan kunci untuk setiap kunci harus mengizinkan akun menggunakan kebijakan IAM untuk mengontrol akses ke kunci.

Misalnya, kebijakan IAM berikut memungkinkan prinsipal untuk membuat kunci. Ini juga memungkinkan mereka untuk membuat dan mengelola tag pada semua kunci di akun yang ditentukan. Kombinasi ini memungkinkan prinsipal untuk menggunakan parameter tag CreateKeyoperasi untuk menambahkan tag ke kunci saat mereka membuatnya.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyCreateKeys",
      "Effect": "Allow",
      "Action": "payment-cryptography:CreateKey",
      "Resource": "*"
    },
    {
      "Sid": "IAMPolicyTags",
      "Effect": "Allow",
      "Action": [
        "payment-cryptography:TagResource",
        "payment-cryptography:UntagResource",
        "payment-cryptography:ListTagsForResource"
      ],
```

```
      "Resource": "arn:aws:payment-cryptography:*:111122223333:key/*"
    }
  ]
}
```

Membatasi izin tanda

Anda dapat membatasi izin penandaan dengan menggunakan ketentuan kebijakan. Kondisi kebijakan berikut dapat diterapkan ke izin `payment-cryptography:TagResource` dan `payment-cryptography:UntagResource`. Misalnya, Anda dapat menggunakan kondisi `aws:RequestTag/`*`tag-key`* mengizinkan prinsipal untuk menambahkan hanya tanda tertentu, atau mencegah prinsipal menambahkan tanda dengan kunci tanda tertentu.

- aws: RequestTag
- aws:ResourceTag/tag-key (hanya kebijakan IAM)
- aws: TagKeys

Sebagai praktik terbaik saat Anda menggunakan tag untuk mengontrol akses ke kunci, gunakan tombol `aws:RequestTag/`*`tag-key`* atau `aws:TagKeys` kondisi untuk menentukan tag (atau kunci tag) mana yang diizinkan.

Sebagai contoh, kebijakan IAM berikut ini mirip dengan yang sebelumnya. Namun, kebijakan ini memungkinkan prinsipal untuk membuat tanda (`TagResource`) dan menghapus tanda `UntagResource` hanya untuk tanda dengan kunci tanda `Project`.

Karena `TagResource` dan `UntagResource` permintaan dapat menyertakan beberapa tag, Anda harus menentukan operator `ForAllValues` atau `ForAnyValue` set dengan TagKeys kondisi aws:. Operator `ForAnyValue` mensyaratkan bahwa setidaknya salah satu kunci tanda dalam permintaan cocok dengan salah satu kunci tanda dalam kebijakan. Operator `ForAllValues` mensyaratkan bahwa semua kunci tanda dalam permintaan cocok dengan salah satu kunci tanda dalam kebijakan. `ForAllValues`Operator juga kembali `true` jika tidak ada tag dalam permintaan, tetapi TagResource dan UntagResource gagal ketika tidak ada tag yang ditentukan. Untuk detail tentang operator set, lihat Gunakan beberapa kunci dan nilai di Panduan Pengguna IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Sid": "IAMPolicyCreateKey",
      "Effect": "Allow",
      "Action": "payment-cryptography:CreateKey",
      "Resource": "*"
    },
    {
      "Sid": "IAMPolicyViewAllTags",
      "Effect": "Allow",
      "Action": "payment-cryptography:ListResourceTags",
      "Resource": "arn:aws:payment-cryptography:*:111122223333:key/*"
    },
    {
      "Sid": "IAMPolicyManageTags",
      "Effect": "Allow",
      "Action": [
        "payment-cryptography:TagResource",
        "payment-cryptography:UntagResource"
      ],
      "Resource": "arn:aws:payment-cryptography:*:111122223333:key/*",
      "Condition": {
          "ForAllValues:StringEquals": {"aws:TagKeys": "Project"}
      }
    }
  ]
}
```

Menggunakan tag untuk mengontrol akses ke tombol

Anda dapat mengontrol akses ke Kriptografi AWS Pembayaran berdasarkan tag pada kunci. Misalnya, Anda dapat menulis kebijakan IAM yang memungkinkan prinsipal mengaktifkan dan menonaktifkan hanya kunci yang memiliki tag tertentu. Atau Anda dapat menggunakan kebijakan IAM untuk mencegah prinsipal menggunakan kunci dalam operasi kriptografi kecuali kunci memiliki tag tertentu.

Fitur ini merupakan bagian dari dukungan Kriptografi AWS Pembayaran untuk kontrol akses berbasis atribut (ABAC). Untuk informasi tentang penggunaan tag untuk mengontrol akses ke AWS sumber daya, lihat Untuk apa ABAC? AWS dan Mengontrol Akses ke AWS Sumber Daya Menggunakan Tag Sumber Daya di Panduan Pengguna IAM.

> **Note**

AWS Kriptografi Pembayaran mendukung kunci konteks kondisi global aws:ResourceTag/tag-key, yang memungkinkan Anda mengontrol akses ke kunci berdasarkan tag pada kunci. Karena beberapa kunci dapat memiliki tag yang sama, fitur ini memungkinkan Anda menerapkan izin ke satu set kunci tertentu. Anda juga dapat dengan mudah mengubah kunci di set dengan mengubah tag mereka.

Dalam Kriptografi AWS Pembayaran, kunci `aws:ResourceTag/`*`tag-key`* kondisi hanya didukung dalam kebijakan IAM. Ini tidak didukung dalam kebijakan utama, yang hanya berlaku untuk satu kunci, atau pada operasi yang tidak menggunakan kunci tertentu, seperti ListKeysatau ListAliasesoperasi.

Mengontrol akses dengan tanda menyediakan cara sederhana, dapat diskalakan, dan fleksibel untuk mengelola izin. Namun, jika tidak dirancang dan dikelola dengan benar, itu dapat mengizinkan atau menolak akses ke kunci Anda secara tidak sengaja. Jika Anda menggunakan tanda untuk mengontrol akses, pertimbangkan praktik berikut.

- Gunakan tanda untuk memperkuat praktik terbaik dari akses dengan keistimewaan terkecil. Berikan kepala sekolah IAM hanya izin yang mereka butuhkan hanya pada kunci yang harus mereka gunakan atau kelola. Misalnya, gunakan tag untuk memberi label kunci yang digunakan untuk proyek. Kemudian berikan izin kepada tim proyek untuk hanya menggunakan kunci dengan tag proyek.
- Berhati-hatilah tentang memberikan prinsipal izin `payment-cryptography:TagResource` dan `payment-cryptography:UntagResource` yang memungkinkan mereka menambahkan, mengedit, dan menghapus tanda. Saat Anda menggunakan tag untuk mengontrol akses ke kunci, mengubah tag dapat memberikan izin kepada prinsipal untuk menggunakan kunci yang tidak diizinkan untuk digunakan. Itu juga dapat menolak akses ke kunci yang diperlukan oleh kepala sekolah lain untuk melakukan pekerjaan mereka. Administrator kunci yang tidak memiliki izin untuk mengubah kebijakan kunci atau membuat hibah dapat mengontrol akses ke kunci jika mereka memiliki izin untuk mengelola tag.

  Jika memungkinkan, gunakan kondisi kebijakan, seperti `aws:RequestTag/`*`tag-key`* atau `aws:TagKeys` untuk membatasi izin penandaan prinsipal untuk tag atau pola tag tertentu pada kunci tertentu.
- Tinjau prinsipal di Anda Akun AWS yang saat ini memiliki izin penandaan dan pembatalan tag dan sesuaikan, jika perlu. Kebijakan IAM mungkin mengizinkan izin tag dan untag pada semua kunci. Misalnya, kebijakan terkelola Admin memungkinkan prinsipal untuk menandai, menghapus tag, dan mencantumkan tag pada semua kunci.

- Sebelum menetapkan kebijakan yang bergantung pada tag, tinjau tag pada kunci di tag Anda Akun AWS. Pastikan bahwa kebijakan Anda hanya berlaku untuk tanda yang ingin Anda sertakan. Gunakan CloudTrail log dan CloudWatch alarm untuk mengingatkan Anda untuk menandai perubahan yang mungkin memengaruhi akses ke kunci Anda.
- Kondisi kebijakan berbasis tanda menggunakan pencocokan pola; mereka tidak terikat pada instans tertentu dari tanda. Kebijakan yang menggunakan kunci kondisi berbasis tanda memengaruhi semua tanda baru dan yang sudah ada yang cocok dengan pola. Jika Anda menghapus dan membuat ulang tanda yang cocok dengan kondisi kebijakan, kondisi berlaku untuk tanda baru, seperti halnya pada tanda lama.

Misalnya, pertimbangkan contoh kebijakan IAM berikut. Ini memungkinkan kepala sekolah untuk memanggil operasi Dekripsi hanya pada kunci di akun Anda yang merupakan Wilayah AS Timur (Virginia N.) dan memiliki tag. `"Project"="Alpha"` Anda mungkin melampirkan kebijakan ini ke peran dalam contoh proyek Alpha.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyWithResourceTag",
      "Effect": "Allow",
      "Action": [
        "payment-cryptography:DecryptData"
      ],
      "Resource": "arn:aws::us-east-1:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "Alpha"
        }
      }
    }
  ]
}
```

Contoh berikut kebijakan IAM memungkinkan prinsipal untuk menggunakan kunci apa pun dalam akun untuk operasi kriptografi tertentu. Tapi itu melarang prinsipal menggunakan operasi kriptografi ini pada kunci dengan tag atau tanpa tag. `"Type"="Reserved"` `"Type"`

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMAllowCryptographicOperations",
      "Effect": "Allow",
      "Action": [
        "payment-cryptography:EncryptData",
        "payment-cryptography:DecryptData",
        "payment-cryptography:ReEncrypt*"
      ],
      "Resource": "arn:aws:payment-cryptography:*:111122223333:key/*"
    },
    {
      "Sid": "IAMDenyOnTag",
      "Effect": "Deny",
      "Action": [
        "payment-cryptography:EncryptData",
        "payment-cryptography:DecryptData",
        "payment-cryptography:ReEncrypt*"
      ],
      "Resource": "arn:aws:payment-cryptography:*:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Type": "Reserved"
        }
      }
    },
    {
      "Sid": "IAMDenyNoTag",
      "Effect": "Deny",
      "Action": [
        "payment-cryptography:EncryptData",
        "payment-cryptography:DecryptData",
        "payment-cryptography:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "Null": {
          "aws:ResourceTag/Type": "true"
        }
      }
    }
  ]
```

```
}
```

Memahami atribut kunci untuk kunci Kriptografi AWS Pembayaran

Prinsip manajemen kunci yang tepat adalah bahwa kunci dicakup dengan tepat dan hanya dapat digunakan untuk operasi yang diizinkan. Dengan demikian, kunci tertentu hanya dapat dibuat dengan mode penggunaan kunci tertentu. Bila memungkinkan, ini sejalan dengan mode penggunaan yang tersedia seperti yang didefinisikan oleh TR-31.

Meskipun Kriptografi AWS Pembayaran akan mencegah Anda membuat kunci yang tidak valid, kombinasi yang valid disediakan di sini untuk kenyamanan Anda.

Tombol Simetris

- TR31_B0_BASE_DERIVATION_KEY
  - Algoritma Kunci yang Diizinkan: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Kombinasi mode penggunaan kunci yang diizinkan: { DeriveKey = true}, { NoRestrictions = true}
- TR31_C0_CARD_VERIFICATION_KEY
  - Algoritma Kunci yang Diizinkan: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Kombinasi mode penggunaan kunci yang diizinkan: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}
- TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY
  - Algoritma Kunci yang Diizinkan: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Kombinasi mode penggunaan kunci yang diizinkan: {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}, {= true} NoRestrictions
- TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS
  - Algoritma Kunci yang Diizinkan: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Kombinasi mode penggunaan kunci yang diizinkan: { DeriveKey = true}, { NoRestrictions = true}
- TR31_E1_EMV_MKEY_CONFIDENTIALITY
  - Algoritma Kunci yang Diizinkan: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Kombinasi mode penggunaan kunci yang diizinkan: { DeriveKey = true}, { NoRestrictions = true}
- TR31_E2_EMV_MKEY_INTEGRITY

  - Algoritma Kunci yang Diizinkan: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Kombinasi mode penggunaan kunci yang diizinkan: { DeriveKey = true}, { NoRestrictions = true}
- TR31_E4_EMV_MKEY_DYNAMIC_NUMBERS
  - Algoritma Kunci yang Diizinkan: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Kombinasi mode penggunaan kunci yang diizinkan: { DeriveKey = true}, { NoRestrictions = true}
- TR31_E5_EMV_MKEY_CARD_PERSONALISASI
  - Algoritma Kunci yang Diizinkan: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Kombinasi mode penggunaan kunci yang diizinkan: { DeriveKey = true}, { NoRestrictions = true}
- TR31_E6_EMV_MKEY_OTHER
  - Algoritma Kunci yang Diizinkan: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Kombinasi mode penggunaan kunci yang diizinkan: { DeriveKey = true}, { NoRestrictions = true}
- TR31_K0_KEY_ENCRYPTION_KEY
  - Algoritma Kunci yang Diizinkan: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Kombinasi mode penggunaan kunci yang diizinkan: {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}, {= true} NoRestrictions
- TR31_K1_KEY_BLOCK_PROTECTION_KEY
  - Algoritma Kunci yang Diizinkan: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Kombinasi mode penggunaan kunci yang diizinkan: {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}, {= true} NoRestrictions
- TR31_M1_ISO_9797_1_MAC_KEY
  - Algoritma Kunci yang Diizinkan: TDES_2KEY, TDES_3KEY
  - Kombinasi mode penggunaan kunci yang diizinkan: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}
- TR31_M3_ISO_9797_3_MAC_KEY
  - Algoritma Kunci yang Diizinkan: TDES_2KEY, TDES_3KEY
  - Kombinasi mode penggunaan kunci yang diizinkan: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}
- TR31_M6_ISO_9797_5_CMAC_KEY
  - Algoritma Kunci yang Diizinkan: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256

  - Kombinasi mode penggunaan kunci yang diizinkan: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}
- TR31_M7_HMAC_KEY
  - Algoritma Kunci yang Diizinkan: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Kombinasi mode penggunaan kunci yang diizinkan: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}
- TR31_P0_PIN_ENCRYPTION_KEY
  - Algoritma Kunci yang Diizinkan: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Kombinasi mode penggunaan kunci yang diizinkan: {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}, {= true} NoRestrictions
- TR31_V1_IBM3624_PIN_VERIFICATION_KEY
  - Algoritma Kunci yang Diizinkan: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Kombinasi mode penggunaan kunci yang diizinkan: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}
- TR31_V2_VISA_PIN_VERIFICATION_KEY
  - Algoritma Kunci yang Diizinkan: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Kombinasi mode penggunaan kunci yang diizinkan: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}

Tombol Asimetris

- TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION
  - Algoritma Kunci yang Diizinkan: RSA_2048, RSA_3072, RSA_4096
  - Kombinasi mode penggunaan kunci yang diizinkan: {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}
  - CATATAN:: {Encrypt = true, Wrap = true} adalah satu-satunya opsi yang valid saat mengimpor kunci publik yang dimaksudkan untuk mengenkripsi data atau membungkus kunci
- TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE
  - Algoritma Kunci yang Diizinkan: RSA_2048, RSA_3072, RSA_4096
  - Kombinasi mode penggunaan kunci yang diizinkan: {Sign = true}, {Verify = true}

- CATATAN:: {Verify = true} adalah satu-satunya opsi yang valid saat mengimpor kunci yang dimaksudkan untuk ditandatangani, seperti sertifikat root, sertifikat perantara, atau sertifikat penandatanganan untuk TR-34.

Operasi data

Setelah Anda membuat kunci Kriptografi AWS Pembayaran, itu dapat digunakan untuk melakukan operasi kriptografi. Operasi yang berbeda melakukan berbagai jenis aktivitas mulai dari enkripsi, hashing, serta algoritma spesifik domain seperti pembuatan CVV2.

Data terenkripsi tidak dapat didekripsi tanpa kunci dekripsi yang cocok (kunci simetris atau kunci pribadi tergantung pada jenis enkripsi). Algoritma hashing dan domain spesifik juga tidak dapat diverifikasi tanpa kunci simetris atau kunci publik.

Untuk informasi tentang jenis kunci yang valid untuk operasi tertentu, silakan lihat Kunci yang valid untuk operasi kriptografi

> **Note**
>
> Sebaiknya gunakan data uji saat berada di lingkungan non-produksi. Menggunakan kunci dan data produksi (PAN, ID BDK, dll.) di lingkungan non-produksi dapat memengaruhi cakupan kepatuhan Anda seperti untuk PCI DSS dan PCI P2PE.

Topik

Enkripsi, Dekripsi, dan Enkripsi Ulang Data

Metode enkripsi dan dekripsi dapat digunakan untuk mengenkripsi atau mendekripsi data menggunakan berbagai teknik simetris dan asimetris termasuk TDES, AES dan RSA. Metode ini juga mendukung kunci yang diturunkan menggunakan teknik DUKPT dan EMV. Untuk kasus penggunaan di mana Anda ingin mengamankan data di bawah kunci baru tanpa mengekspos data yang mendasarinya, ReEncrypt perintah juga dapat digunakan.

> **Note**
>
> Saat menggunakan fungsi enkripsi/dekripsi, semua input diasumsikan berada di HexBinary - misalnya nilai 1 akan dimasukkan sebagai 31 (hex) dan huruf kecil t direpresentasikan sebagai 74 (hex). Semua output ada di HexBinary juga.

Untuk detail tentang semua opsi yang tersedia, silakan baca Panduan API untuk Enkripsi, Dekripsi, dan Enkripsi Ulang.

Topik

Enkripsi data

`Encrypt Data`API digunakan untuk mengenkripsi data menggunakan kunci enkripsi data simetris dan asimetris serta kunci turunan DUKPT dan EMV. Berbagai algoritma dan variasi didukung termasuk`TDES`, `RSA` dan`AES`.

Input utama adalah kunci enkripsi yang digunakan untuk mengenkripsi data, data teks biasa dalam format HexBinary yang akan dienkripsi dan atribut enkripsi seperti vektor inisialisasi dan mode untuk sandi blok seperti TDES. Data plaintext harus dalam kelipatan 8 byte untuk`TDES`, 16 byte untuk `AES` dan panjang kunci dalam kasus. `RSA` Input kunci simetris (TDES, AES, DUKPT, EMV) harus empuk dalam kasus di mana data input tidak memenuhi persyaratan ini. Tabel berikut menunjukkan panjang maksimum plaintext untuk setiap jenis kunci dan jenis padding yang Anda tentukan `EncryptionAttributes` untuk kunci RSA.

Jenis bantalan	RSA_2048	RSA_3072	RSA_4096
`OAEP SHA1`	428	684	940
`OAEP SHA256`	380	636	892
`OAEP SHA512`	252	508	764
`PKCS1`	488	744	1000

Jenis bantalan	RSA_2048	RSA_3072	RSA_4096
None	488	744	1000

Output utama termasuk data terenkripsi sebagai ciphertext dalam format HexBinary dan nilai checksum untuk kunci enkripsi. Untuk detail tentang semua opsi yang tersedia, silakan baca Panduan API untuk Enkripsi.

Contoh-contoh

- Enkripsi data menggunakan kunci simetris AES
- Enkripsi data menggunakan kunci DUKPT
- Enkripsi data menggunakan kunci simetris turunan EMV
- Enkripsi data menggunakan kunci RSA

Enkripsi data menggunakan kunci simetris AES

> **Note**
>
> Semua contoh mengasumsikan kunci yang relevan sudah ada. Kunci dapat dibuat menggunakan CreateKeyoperasi atau diimpor menggunakan ImportKeyoperasi.

Example

Dalam contoh ini, kita akan mengenkripsi data plaintext menggunakan kunci simetris yang telah dibuat menggunakan CreateKeyOperasi atau diimpor menggunakan Operasi. ImportKey Untuk operasi ini, kunci harus KeyModesOfUse disetel ke `Encrypt` dan KeyUsage disetel ke`TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY`. Silakan lihat Kunci untuk Operasi Kriptografi untuk opsi lainnya.

```
$ aws payment-cryptography-data encrypt-data --key-identifier arn:aws:payment-
cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi --plain-text
 313233343132333431323334313233 34 --encryption-attributes 'Symmetric={Mode=CBC}'
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
    "KeyCheckValue": "71D7AE",
    "CipherText": "33612AB9D6929C3A828EB6030082B2BD"
}
```

Enkripsi data menggunakan kunci DUKPT

Example

Dalam contoh ini, kita akan mengenkripsi data plaintext menggunakan kunci DUKPT. AWS Dukungan Kriptografi Pembayaran `TDES` dan kunci `AES` DUKPT. Untuk operasi ini, kunci harus KeyModesOfUse disetel ke `DeriveKey` dan KeyUsage disetel ke`TR31_B0_BASE_DERIVATION_KEY`. Silakan lihat Kunci untuk Operasi Kriptografi untuk opsi lainnya.

```
$ aws payment-cryptography-data encrypt-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi
 --plain-text 31323334313233343132333431323334 --encryption-attributes
 'Dukpt={KeySerialNumber=FFFF9876543210E00001}'
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
    "KeyCheckValue": "71D7AE",
    "CipherText": "33612AB9D6929C3A828EB6030082B2BD"
}
```

Enkripsi data menggunakan kunci simetris turunan EMV

Example

Dalam contoh ini, kita akan mengenkripsi data teks yang jelas menggunakan kunci simetris turunan EMV yang telah dibuat. Anda dapat menggunakan perintah seperti ini untuk mengirim data ke kartu

EMV. Untuk operasi ini, kunci harus KeyModesOfUse disetel ke `Derive` dan KeyUsage disetel ke `TR31_E1_EMV_MKEY_CONFIDENTIALITY` atau`TR31_E6_EMV_MKEY_OTHER`. Silakan lihat Kunci untuk Operasi Kriptografi untuk lebih jelasnya.

```
$ aws payment-cryptography-data encrypt-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi
 --plain-text 33612AB9D6929C3A828EB6030082B2BD --encryption-attributes
 'Emv={MajorKeyDerivationMode=EMV_OPTION_A,PanSequenceNumber=27,PrimaryAccountNumber=1000000000
 InitializationVector=1500000000000999,Mode=CBC}'
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
    "KeyCheckValue": "71D7AE",
    "CipherText": "33612AB9D6929C3A828EB6030082B2BD"
}
```

Enkripsi data menggunakan kunci RSA

Example

Dalam contoh ini, kita akan mengenkripsi data plaintext menggunakan kunci publik RSA yang telah diimpor menggunakan operasi. ImportKey Untuk operasi ini, kunci harus KeyModesOfUse disetel ke `Encrypt` dan KeyUsage disetel ke`TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION`. Silakan lihat Kunci untuk Operasi Kriptografi untuk opsi lainnya.

Untuk PKCS #7 atau skema padding lainnya yang saat ini tidak didukung, mohon terapkan sebelum memanggil layanan dan pilih no padding dengan menghilangkan indikator padding 'Asymmetric= {} '

```
$ aws payment-cryptography-data encrypt-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/thfezpmsalcfwmsg
 --plain-text 313233343132333431323334313233343 --encryption-attributes
 'Asymmetric={PaddingType=OAEP_SHA256}'
```

```
{
 "CipherText":
 "12DF6A2F64CC566D124900D68E8AFEAA794CA819876E258564D525001D00AC93047A83FB13 \
```

```
 E73F06329A100704FA484A15A49F06A7A2E55A241D276491AA91F6D2D8590C60CDE57A642BC64A897F4832A3930
 \
 0FAEC7981102CA0F7370BFBF757F271EF0BB2516007AB111060A9633D1736A9158042D30C5AE11F8C5473EC70F067
 \
 72590DEA1638E2B41FAE6FB1662258596072B13F8E2F62F5D9FAF92C12BB70F42F2ECDCF56AADF0E311D4118FE3591
 \
 FB672998CCE9D00FFFE05D2CD154E3120C5443C8CF9131C7A6A6C05F5723B8F5C07A4003A5A6173E1B425E2B5E42AD
 \
 7A2966734309387C9938B029AFB20828ACFC6D00CD1539234A4A8D9B94CDD4F23A",
 "KeyArn": "arn:aws:payment-cryptography:us-east-1:529027455495:key/5dza7xqd6soanjtb",
 "KeyCheckValue": "FF9DE9CE"
}
```

Dekripsi data

`Decrypt Data`API digunakan untuk mendekripsi data menggunakan kunci enkripsi data simetris dan asimetris serta kunci turunan DUKPT dan EMV. Berbagai algoritma dan variasi didukung termasuk`TDES`, `RSA` dan`AES`.

Input utama adalah kunci dekripsi yang digunakan untuk mendekripsi data, data ciphertext dalam format HexBinary yang akan didekripsi dan atribut dekripsi seperti vektor inisialisasi, mode sebagai cipher blok dll. Output utama termasuk data yang didekripsi sebagai plaintext dalam format HexBinary dan nilai checksum untuk kunci dekripsi. Untuk detail tentang semua opsi yang tersedia, silakan baca Panduan API untuk Dekripsi.

Contoh-contoh

- Dekripsi data menggunakan kunci simetris AES
- Dekripsi data menggunakan kunci DUKPT
- Dekripsi data menggunakan kunci simetris turunan EMV
- Dekripsi data menggunakan kunci RSA

Dekripsi data menggunakan kunci simetris AES

Example

Dalam contoh ini, kita akan mendekripsi data ciphertext menggunakan kunci simetris. Contoh ini menunjukkan `AES` kunci tetapi `TDES_2KEY` dan `TDES_3KEY` juga didukung.

Untuk operasi ini, kunci harus KeyModesOfUse disetel ke `Decrypt` dan KeyUsage disetel ke`TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY`. Silakan lihat Kunci untuk Operasi Kriptografi untuk opsi lainnya.

```
$ aws payment-cryptography-data decrypt-data --key-identifier arn:aws:payment-
cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi --cipher-text
 33612AB9D6929C3A828EB6030082B2BD --decryption-attributes 'Symmetric={Mode=CBC}'
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
    "KeyCheckValue": "71D7AE",
    "PlainText": "31323334313233343132333431323334"
}
```

Dekripsi data menggunakan kunci DUKPT

> **Note**
>
> Menggunakan data dekripsi dengan DUKPT untuk transaksi P2PE dapat mengembalikan PAN kartu kredit dan data pemegang kartu lainnya ke aplikasi Anda yang perlu dipertanggungjawabkan saat menentukan cakupan PCI DSS-nya.

Example

Dalam contoh ini, kita akan mendekripsi data ciphertext menggunakan kunci DUKPT yang telah dibuat menggunakan CreateKeyOperasi atau diimpor menggunakan Operasi. ImportKey Untuk operasi ini, kunci harus KeyModesOfUse disetel ke `DeriveKey` dan KeyUsage disetel ke`TR31_B0_BASE_DERIVATION_KEY`. Silakan lihat Kunci untuk Operasi Kriptografi untuk opsi lainnya. Bila Anda menggunakanDUKPT, untuk `TDES` algoritma, panjang data ciphertext harus kelipatan 16 byte. Untuk `AES` algoritma, panjang data ciphertext harus kelipatan 32 byte.

```
$ aws payment-cryptography-data decrypt-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi
```

```
 --cipher-text 33612AB9D6929C3A828EB6030082B2BD --decryption-attributes
 'Dukpt={KeySerialNumber=FFFF9876543210E00001}'
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
    "KeyCheckValue": "71D7AE",
    "PlainText": "31323334313233343132333431323334"
}
```

Dekripsi data menggunakan kunci simetris turunan EMV

Example

Dalam contoh ini, kita akan mendekripsi data ciphertext menggunakan kunci simetris turunan EMV yang telah dibuat menggunakan operasi atau diimpor menggunakan operasi. CreateKeyImportKey Untuk operasi ini, kunci harus KeyModesOfUse disetel ke `Derive` dan KeyUsage disetel ke `TR31_E1_EMV_MKEY_CONFIDENTIALITY` atau`TR31_E6_EMV_MKEY_OTHER`. Silakan lihat Kunci untuk Operasi Kriptografi untuk lebih jelasnya.

```
$ aws payment-cryptography-data decrypt-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi
 --cipher-text 33612AB9D6929C3A828EB6030082B2BD --decryption-attributes
 'Emv={MajorKeyDerivationMode=EMV_OPTION_A,PanSequenceNumber=27,PrimaryAccountNumber=1000000000
 InitializationVector=1500000000000999,Mode=CBC}'
```

```
{
"KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi",
"KeyCheckValue": "71D7AE",
"PlainText": "31323334313233343132333431323334"
}
```

Dekripsi data menggunakan kunci RSA

Example

Dalam contoh ini, kita akan mendekripsi data ciphertext menggunakan key pair RSA yang telah dibuat menggunakan operasi. CreateKey Untuk operasi ini, kunci harus KeyModesOfUse disetel untuk mengaktifkan `Decrypt` dan KeyUsage mengatur ke`TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION`. Silakan lihat Kunci untuk Operasi Kriptografi untuk opsi lainnya.

Untuk PKCS #7 atau skema padding lainnya yang saat ini tidak didukung, pilih no padding dengan menghilangkan indikator padding 'Asymmetric= {} 'dan hapus padding setelah memanggil layanan.

```
$ aws payment-cryptography-data decrypt-data \
        --key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/5dza7xqd6soanjtb --cipher-text
 8F4C1CAFE7A5DEF9A40BEDE7F2A264635C... \
        --decryption-attributes 'Asymmetric={PaddingType=OAEP_SHA256}'
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-
east-1:529027455495:key/5dza7xqd6soanjtb",
    "KeyCheckValue": "FF9DE9CE",
    "PlainText": "313233343132333431323334313233343132333431323334"
}
```

Menghasilkan dan memverifikasi data kartu

Menghasilkan dan memverifikasi data kartu menggabungkan data yang berasal dari data kartu, misalnya CVV, CVV2, CVC dan DCVV.

Topik

- Hasilkan data kartu
- Verifikasi data kartu

Hasilkan data kartu

`Generate Card Data`API digunakan untuk menghasilkan data kartu menggunakan algoritma seperti CVV, CVV2 atau Dynamic CVV2. Untuk melihat kunci apa yang dapat digunakan untuk perintah ini, silakan lihat Kunci yang valid untuk operasi kriptografi bagian.

Banyak nilai kriptografi seperti CVV, CVV2, iCVV, CAVV V8 menggunakan algoritma kriptografi yang sama tetapi memvariasikan nilai input. Misalnya CardVerificationValue1 memiliki input ServiceCode, Nomor Kartu dan Tanggal Kedaluwarsa. Sementara CardVerificationValue2 hanya memiliki dua input ini, ini karena untuk CVV2/CVC2, ditetapkan pada 000. ServiceCode Demikian pula, untuk iCVV ServiceCode ditetapkan pada 999. Beberapa algoritma dapat menggunakan kembali bidang yang ada seperti CAVV V8 dalam hal ini Anda perlu berkonsultasi dengan manual penyedia Anda untuk nilai input yang benar.

> **Note**
>
> Tanggal kedaluwarsa harus dimasukkan dalam format yang sama (seperti MMYY vs YYMM) untuk pembuatan dan validasi untuk menghasilkan hasil yang benar.

Hasilkan CVV2

Example

Dalam contoh ini, kita akan menghasilkan CVV2 untuk PAN tertentu dengan input PAN dan tanggal kedaluwarsa kartu. Ini mengasumsikan bahwa Anda memiliki kunci verifikasi kartu yang dihasilkan.

```
$ aws payment-cryptography-data generate-card-validation-data --key-
identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi --primary-account-number=171234567890123 --generation-attributes
 CardVerificationValue2={CardExpiryDate=0123}
```

```
  {
      "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
      "KeyCheckValue": "CADDA1",
      "ValidationData": "801"
```

```
  }
```

Menghasilkan iCVV

Example

Dalam contoh ini, kami akan menghasilkan iCVV untuk PAN tertentu dengan inputPAN, kode layanan 999 dan tanggal kedaluwarsa kartu. Ini mengasumsikan bahwa Anda memiliki kunci verifikasi kartu yang dihasilkan.

Untuk semua parameter yang tersedia, lihat CardVerificationValue1 di panduan referensi API.

```
$ aws payment-cryptography-data generate-card-validation-data --key-
identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi --primary-account-number=171234567890123 --generation-attributes
 CardVerificationValue1='{CardExpiryDate=1127,ServiceCode=999}'
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
    "KeyCheckValue": "CADDA1",
    "ValidationData": "801"
}
```

Verifikasi data kartu

`Verify Card Data`digunakan untuk memverifikasi data yang telah dibuat menggunakan algoritma pembayaran yang mengandalkan prinsip enkripsi seperti. `DISCOVER_DYNAMIC_CARD_VERIFICATION_CODE`

Nilai input biasanya diberikan sebagai bagian dari transaksi masuk ke penerbit atau mitra platform pendukung. Untuk memverifikasi kriptogram ARQC (digunakan untuk kartu chip EMV), silakan lihat Verifikasi ARQC.

Untuk informasi selengkapnya, lihat VerifyCardValidationDatadi panduan API.

Jika nilainya diverifikasi, maka api akan mengembalikan http/200. Jika nilainya tidak diverifikasi, itu akan mengembalikan http/400.

Verifikasi CVV2

Example

Dalam contoh ini, kita akan memvalidasi CVV/CVV2 untuk PAN tertentu. CVV2 biasanya disediakan oleh pemegang kartu atau pengguna selama waktu transaksi untuk validasi. Untuk memvalidasi input mereka, nilai-nilai berikut akan diberikan saat runtime - Kunci untuk Digunakan untuk validasi (CVK),PAN, tanggal kedaluwarsa kartu dan CVV2 dimasukkan. Format kedaluwarsa kartu harus sesuai dengan yang digunakan dalam pembuatan nilai awal.

Untuk semua parameter yang tersedia, lihat CardVerificationValue2 di panduan referensi API.

```
$ aws payment-cryptography-data verify-card-validation-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi
 --primary-account-number=171234567890123 --verification-attributes
 CardVerificationValue2={CardExpiryDate=0123} --validation-data 801
```

```
  {
      "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
      "KeyCheckValue": "CADDA1"
  }
```

Verifikasi iCvv

Example

Dalam contoh ini, kami akan memverifikasi iCVV untuk PAN tertentu dengan input Key to Use for validation (CVK),, kode layanan 999PAN, tanggal kedaluwarsa kartu dan iCVV yang disediakan oleh transaksi untuk memvalidasi.

iCVV bukan nilai yang dimasukkan pengguna (seperti CVV2) tetapi disematkan pada kartu EMV. Pertimbangan harus diberikan apakah harus selalu memvalidasi saat disediakan.

Untuk semua parameter yang tersedia, lihat, CardVerificationValue1 dalam panduan referensi API.

```
$ aws payment-cryptography-data generate-card-validation-data --key-
identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
```

```
tqv5yij6wtxx64pi --primary-account-number=171234567890123 --generation-attributes
 CardVerificationValue1='{CardExpiryDate=1127,ServiceCode=999}'
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
    "KeyCheckValue": "CADDA1",
    "ValidationData": "801"
}
```

Menghasilkan, menerjemahkan, dan memverifikasi data PIN

Fungsi data PIN memungkinkan Anda untuk menghasilkan pin acak, nilai verifikasi pin (PVV) dan memvalidasi pin terenkripsi masuk terhadap PVV atau PIN Offset.

Terjemahan pin memungkinkan Anda menerjemahkan pin dari satu kunci kerja ke yang lain tanpa mengekspos pin dalam teks yang jelas seperti yang ditentukan oleh Persyaratan PIN PCI 1.

> **Note**
>
> Karena pembuatan dan validasi PIN biasanya merupakan fungsi penerbit dan terjemahan PIN adalah fungsi pengakuisisi yang khas, kami menyarankan Anda mempertimbangkan akses yang paling tidak dipriviledkan dan menetapkan kebijakan dengan tepat untuk kasus penggunaan sistem Anda.

Topik

Terjemahkan data PIN

Fungsi data PIN Translate digunakan untuk menerjemahkan data PIN terenkripsi dari satu set kunci ke yang lain tanpa data terenkripsi meninggalkan HSM. Ini digunakan untuk enkripsi P2PE

di mana kunci kerja harus berubah tetapi sistem pemrosesan tidak perlu, atau tidak diizinkan untuk, mendekripsi data. Input utama adalah data terenkripsi, kunci enkripsi yang digunakan untuk mengenkripsi data, parameter yang digunakan untuk menghasilkan nilai input. Kumpulan input lainnya adalah parameter output yang diminta seperti kunci yang akan digunakan untuk mengenkripsi output dan parameter yang digunakan untuk membuat output itu. Output utama adalah dataset yang baru dienkripsi serta parameter yang digunakan untuk menghasilkannya.

> **Note**
>
> Jenis kunci AES hanya mendukung blok ISO Format 4 pin.

Topik

- PIN dari PEK ke DUKPT
- PIN dari DUKPT ke AWK

PIN dari PEK ke DUKPT

Example

Dalam contoh ini, kami akan menerjemahkan PIN dari enkripsi PEK TDES menggunakan blok PIN ISO 0 ke Blok PIN AES ISO 4 menggunakan algoritma DUKPT. Biasanya ini mungkin dilakukan secara terbalik, di mana terminal pembayaran mengenkripsi pin dalam ISO 4 dan kemudian dapat diterjemahkan kembali ke TDES untuk pemrosesan hilir.

```
$ aws payment-cryptography-data translate-pin-data --encrypted-pin-block
 "AC17DC148BDA645E" --incoming-translation-
attributes=IsoFormat0='{PrimaryAccountNumber=171234567890123}' --incoming-
key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt  --outgoing-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/4pmyquwjs3yj4vwe --outgoing-translation-attributes
 IsoFormat4="{PrimaryAccountNumber=171234567890123}" --outgoing-dukpt-attributes
 KeySerialNumber="FFFF9876543210E00008"
```

```
{
        "PinBlock": "1F4209C670E49F83E75CC72E81B787D9",
```

```
            "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt",
            "KeyCheckValue": "7CC9E2"
        }
```

PIN dari DUKPT ke AWK

Example

Dalam contoh ini, kami akan menerjemahkan PIN dari PIN terenkripsi AES DUKPT ke pin yang dienkripsi di bawah AWK. Ini secara fungsional kebalikan dari contoh sebelumnya.

```
$ aws payment-cryptography-data translate-pin-data --encrypted-pin-
block "1F4209C670E49F83E75CC72E81B787D9" --outgoing-translation-
attributes=IsoFormat0='{PrimaryAccountNumber=171234567890123}' --outgoing-
key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt   --incoming-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/4pmyquwjs3yj4vwe --incoming-translation-attributes
 IsoFormat4="{PrimaryAccountNumber=171234567890123}" --incoming-dukpt-attributes
 KeySerialNumber="FFFF9876543210E00008"
```

```
        {
            "PinBlock": "AC17DC148BDA645E",
            "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt",
            "KeyCheckValue": "FE23D3"
        }
```

Hasilkan data PIN

Menghasilkan fungsi data PIN digunakan untuk menghasilkan nilai terkait PIN, seperti PVV dan offset blok pin yang digunakan untuk memvalidasi entri pin oleh pengguna selama waktu transaksi atau otorisasi. API ini juga dapat menghasilkan pin acak baru menggunakan berbagai algoritma.

Hasilkan Visa PVV untuk pin

Example

Dalam contoh ini, kami akan menghasilkan pin baru (acak) di mana output akan dienkripsi `PIN block` (. PinData PinBlock) dan a `PVV` (pindata.offset). Input kuncinya adalah`PAN`, the`Pin Verification Key`, the `Pin Encryption Key` and the. `PIN block format`

Perintah ini mengharuskan kuncinya bertipe`TR31_V2_VISA_PIN_VERIFICATION_KEY`.

```
$ aws payment-cryptography-data generate-pin-data --generation-key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/37y2tsl45p5zjbh2 --encryption-
key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/ivi5ksfsuplneuyt
 --primary-account-number 171234567890123 --pin-block-format ISO_FORMAT_0 --generation-
attributes VisaPin={PinVerificationKeyIndex=1}
```

```
{
            "GenerationKeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/37y2tsl45p5zjbh2",
            "GenerationKeyCheckValue": "7F2363",
            "EncryptionKeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/ivi5ksfsuplneuyt",
            "EncryptionKeyCheckValue": "7CC9E2",
            "EncryptedPinBlock": "AC17DC148BDA645E",
            "PinData": {
                "VerificationValue": "5507"
            }
        }
```

Hasilkan offset pin IBM3624 untuk pin

IBM 3624 PIN Offset juga kadang-kadang disebut metode IBM. Metode ini menghasilkan PIN alami/menengah menggunakan data validasi (biasanya PAN) dan Kunci PIN (PVK). Pin alami secara efektif merupakan nilai turunan dan deterministik sangat efisien untuk ditangani oleh penerbit karena tidak ada data pin yang perlu disimpan pada tingkat pemegang kartu. Kontra yang paling jelas adalah bahwa skema ini tidak cocok untuk pin yang dapat dipilih atau acak pemegang kartu. Untuk memungkinkan jenis pin tersebut, algoritma offset ditambahkan ke skema. Offset mewakili perbedaan antara pin yang dipilih pengguna (atau acak) dan kunci alami. Nilai offset disimpan oleh penerbit kartu atau prosesor kartu. Pada saat transaksi, layanan Kriptografi AWS Pembayaran secara internal menghitung ulang pin alami dan menerapkan offset untuk menemukan pin. Kemudian membandingkan ini dengan nilai yang diberikan oleh otorisasi transaksi.

Beberapa opsi ada untuk IBM3624:

- `Ibm3624NaturalPin`akan menampilkan pin alami dan blok pin terenkripsi
- `Ibm3624PinFromOffset`akan menghasilkan blok pin terenkripsi yang diberi offset
- `Ibm3624RandomPin`akan menghasilkan pin acak dan kemudian blok pin offset dan terenkripsi yang cocok.
- `Ibm3624PinOffset`menghasilkan offset pin yang diberikan pin yang dipilih pengguna.

Internal Kriptografi AWS Pembayaran, langkah-langkah berikut dilakukan:

- Pad panci yang disediakan hingga 16 karakter. Jika <16 disediakan, pad di sisi kanan menggunakan karakter padding yang disediakan.
- Mengenkripsi data validasi menggunakan kunci pembuatan PIN.
- Dekimalisasi data terenkripsi menggunakan tabel desimalisasi. Ini memetakan digit heksidesimal ke digit desimal misalnya 'A' dapat memetakan ke 9 dan 1 dapat memetakan ke 1.
- Dapatkan 4 digit pertama dari representasi heksidesimal output. Ini adalah pin alami.
- Jika pin yang dipilih pengguna atau acak dihasilkan, modulo kurangi pin alami dengan pin pelanggan. Hasilnya adalah offset pin.

Contoh

- Hasilkan offset pin IBM3624 untuk pin

Hasilkan offset pin IBM3624 untuk pin

Dalam contoh ini, kami akan menghasilkan pin baru (acak) di mana output akan dienkripsi `PIN block` (. PinData PinBlock) dan nilai `IBM3624` offset (pindata.offset). Inputnya adalah`PAN`, data validasi (biasanya pan), karakter padding,`Pin Verification Key`, dan. `Pin Encryption Key` `PIN block format`

Perintah ini mensyaratkan bahwa kunci pembuatan pin adalah tipe `TR31_V1_IBM3624_PIN_VERIFICATION_KEY` dan kunci enkripsi bertipe `TR31_P0_PIN_ENCRYPTION_KEY`

Example

Contoh berikut menunjukkan menghasilkan pin acak kemudian mengeluarkan blok pin terenkripsi dan nilai offset IBM3624 menggunakan Ibm3624 RandomPin

```
$ aws payment-cryptography-data generate-pin-data --generation-key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/37y2tsl45p5zjbh2
 --encryption-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/ivi5ksfsuplneuyt --primary-account-number
 171234567890123 --pin-block-format ISO_FORMAT_0 --generation-attributes
 Ibm3624RandomPin="{DecimalizationTable=9876543210654321,PinValidationDataPadCharacter=D,PinVal
```

```
{
             "GenerationKeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/37y2tsl45p5zjbh2",
             "GenerationKeyCheckValue": "7F2363",
             "EncryptionKeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/ivi5ksfsuplneuyt",
             "EncryptionKeyCheckValue": "7CC9E2",
             "EncryptedPinBlock": "AC17DC148BDA645E",
             "PinData": {
                 "PinOffset": "5507"
             }
         }
```

Verifikasi data PIN

Verifikasi fungsi data PIN digunakan untuk memverifikasi apakah pin sudah benar. Ini biasanya melibatkan membandingkan nilai pin yang sebelumnya disimpan dengan apa yang dimasukkan oleh pemegang kartu di POI. Fungsi-fungsi ini membandingkan dua nilai tanpa mengekspos nilai yang mendasari dari salah satu sumber.

Validasi PIN terenkripsi menggunakan metode PVV

Example

Dalam contoh ini, kita akan memvalidasi PIN untuk PAN tertentu. PIN biasanya disediakan oleh pemegang kartu atau pengguna selama waktu transaksi untuk validasi dan dibandingkan dengan nilai pada file (input dari pemegang kartu diberikan sebagai nilai terenkripsi dari terminal atau penyedia hulu lainnya). Untuk memvalidasi input ini, nilai-nilai berikut juga akan diberikan saat

runtime - Kunci yang digunakan untuk mengenkripsi pin input (ini seringIWK), PAN dan nilai untuk memverifikasi terhadap (baik a PVV atau). PIN offset

Jika Kriptografi AWS Pembayaran dapat memvalidasi pin, http/200 dikembalikan. Jika pin tidak divalidasi, itu akan mengembalikan http/400.

```
$ aws payment-cryptography-data verify-pin-data --verification-key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/37y2tsl45p5zjbh2 --encryption-
key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/ivi5ksfsuplneuyt
 --primary-account-number 171234567890123 --pin-block-format ISO_FORMAT_0  --
verification-attributes VisaPin="{PinVerificationKeyIndex=1,VerificationValue=5507}" --
encrypted-pin-block AC17DC148BDA645E
```

```
        {
            "VerificationKeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/37y2tsl45p5zjbh2",
            "VerificationKeyCheckValue": "7F2363",
            "EncryptionKeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/ivi5ksfsuplneuyt",
            "EncryptionKeyCheckValue": "7CC9E2",

        }
```

Validasi PIN terhadap offset pin IBM3624 yang disimpan sebelumnya

Dalam contoh ini, kami akan memvalidasi PIN yang diberikan pemegang kartu terhadap offset pin yang disimpan pada file dengan penerbit/prosesor kartu. Input serupa ??? dengan tambahan pin terenkripsi yang disediakan oleh terminal pembayaran (atau penyedia hulu lainnya seperti jaringan kartu). Jika pin cocok, api akan mengembalikan http 200. di mana output akan dienkripsi PIN block (. PinData PinBlock) dan nilai IBM3624 offset (pindata.offset).

Perintah ini mensyaratkan bahwa kunci pembuatan pin adalah tipe TR31_V1_IBM3624_PIN_VERIFICATION_KEY dan kunci enkripsi bertipe TR31_P0_PIN_ENCRYPTION_KEY

Example

```
$ aws payment-cryptography-data generate-pin-data --generation-key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/37y2tsl45p5zjbh2
 --encryption-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/ivi5ksfsuplneuyt --primary-account-number
 171234567890123 --pin-block-format ISO_FORMAT_0 --generation-attributes
 Ibm3624RandomPin="{DecimalizationTable=9876543210654321,PinValidationDataPadCharacter=D,PinVal
```

```
{
"GenerationKeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/37y2tsl45p5zjbh2",
"GenerationKeyCheckValue": "7F2363",
"EncryptionKeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt",
"EncryptionKeyCheckValue": "7CC9E2",
"EncryptedPinBlock": "AC17DC148BDA645E",
"PinData": {
  "PinOffset": "5507"
}
}
```

Verifikasi kriptogram permintaan autentikasi (ARQC)

API kriptogram permintaan autentikasi verifikasi digunakan untuk memverifikasi ARQC. Generasi ARQC berada di luar cakupan Kriptografi AWS Pembayaran dan biasanya dilakukan pada Kartu Chip EMV (atau setara digital seperti dompet seluler) selama waktu otorisasi transaksi. ARQC unik untuk setiap transaksi dan dimaksudkan untuk menunjukkan validitas kartu secara kriptografis serta untuk memastikan bahwa data transaksi sama persis dengan transaksi saat ini (yang diharapkan).

AWS Kriptografi Pembayaran menyediakan berbagai opsi untuk memvalidasi ARQC dan menghasilkan nilai ARPC opsional termasuk yang didefinisikan dalam EMV 4.4 Buku 2 dan skema lain yang digunakan oleh Visa dan Mastercard. Untuk daftar lengkap semua opsi yang tersedia, silakan lihat VerifyCardValidationData bagian di Panduan API.

Kriptogram ARQC biasanya memerlukan input berikut (meskipun ini mungkin berbeda berdasarkan implementasi):

- PAN - Ditentukan di PrimaryAccountNumber lapangan
- Nomor Urutan PAN (PSN) - ditentukan di lapangan PanSequenceNumber

- Metode Derivasi Kunci seperti Common Session Key (CSK) - Ditentukan dalam SessionKeyDerivationAttributes
- Master Key Derivation Mode (seperti EMV Option A) - Ditentukan dalam MajorKeyDerivationMode
- Data transaksi - serangkaian berbagai transaksi, terminal dan data kartu seperti Jumlah dan Tanggal - ditentukan dalam TransactionData bidang
- Penerbit Master Key - kunci utama yang digunakan untuk mendapatkan kunci kriptogram (AC) yang digunakan untuk melindungi transaksi individu dan ditentukan di lapangan KeyIdentifier

Topik

Membangun data transaksi

Konten (dan urutan) yang tepat dari bidang data transaksi bervariasi menurut implementasi dan skema jaringan tetapi bidang minimum yang direkomendasikan (dan urutan penggabungan) didefinisikan dalam EMV 4.4 Buku 2 Bagian 8.1.1 - Pemilihan Data. Jika tiga bidang pertama adalah jumlah (17.00), jumlah lain (0.00) dan negara pembelian, yang akan menghasilkan data transaksi dimulai sebagai berikut:

- 000000001700 - jumlah - 12 posisi tersirat dua digit desimal
- 000000000000 - jumlah lainnya - 12 posisi tersirat dua digit desimal
- 0124 - kode negara empat digit
- Data Transaksi Keluaran (sebagian) - 00000000170000000000000000000124

Padding data transaksi

Data transaksi harus empuk sebelum dikirim ke layanan. Sebagian besar skema menggunakan padding ISO 9797 Metode 2, di mana string hex ditambahkan oleh hex 80 diikuti oleh 00 hingga bidang adalah kelipatan dari ukuran blok enkripsi; 8 byte atau 16 karakter untuk TDES dan 16 byte atau 32 karakter untuk AES. Alternatif (metode 1) tidak umum tetapi hanya menggunakan 00 sebagai karakter padding.

ISO 9797 Metode 1 Padding

Tidak empuk:
0000000017000000000000000840008000800080000840160517000000000938000 00B03011203 (74 karakter atau 37 byte)

Empuk:
0000000017000000000000000840008000800080000840160517000000000938000 00B03011203 000000 (80 karakter atau 40 byte)

ISO 9797 Metode 2 Padding

Tidak empuk:
0000000017000000000000000840008000800080000840160517000000000938000 00B1F220103000000 (80 karakter atau 40 byte)

Empuk:
0000000017000000000000000840008400080000840160517000000000938000 00B1F220103000000 80000000000000 (88 karakter atau 44 byte)

Contoh

Visa CVN10

Example

Dalam contoh ini, kami akan memvalidasi ARQC yang dihasilkan menggunakan Visa CVN10.

Jika Kriptografi AWS Pembayaran dapat memvalidasi ARQC, http/200 dikembalikan. Jika arqc tidak divalidasi, itu akan mengembalikan respons http/400.

```
$ aws payment-cryptography-data verify-auth-request-cryptogram --auth-request-
cryptogram D791093C8A921769 \
--key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6nl62t5ushfk \
--major-key-derivation-mode EMV_OPTION_A \
--transaction-data
 0000000017000000000000000840008000800008401605170000000009380000B03011203000000 \
--session-key-derivation-attributes='{"Visa":{"PanSequenceNumber":"01" \
,"PrimaryAccountNumber":"9137631040001422"}}'
```

```
{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/pw3s6nl62t5ushfk",
  "KeyCheckValue": "08D7B4"
}
```

Visa CVN18 dan Visa CVN22

Example

Dalam contoh ini, kami akan memvalidasi ARQC yang dihasilkan menggunakan Visa CVN18 atau CVN22. Operasi kriptografi sama antara CVN18 dan CVN22 tetapi data yang terkandung dalam data transaksi bervariasi. Dibandingkan dengan CVN10, kriptogram yang sama sekali berbeda dihasilkan bahkan dengan input yang sama.

Jika Kriptografi AWS Pembayaran dapat memvalidasi ARQC, http/200 dikembalikan. Jika arqc tidak divalidasi, itu akan mengembalikan http/400.

```
$ aws payment-cryptography-data verify-auth-request-cryptogram \
--auth-request-cryptogram 61EDCC708B4C97B4
--key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6nl62t5ushfk \
--major-key-derivation-mode EMV_OPTION_A
--transaction-data
 00000000170000000000000008400080008000084016051700000000093800000B1F22010300000000000
 \
00000000000000000000000000000000000000000000800000000000000000
--session-key-derivation-attributes='{"EmvCommon":
{"ApplicationTransactionCounter":"000B", \
"PanSequenceNumber":"01","PrimaryAccountNumber":"9137631040001422"}}'
```

```
{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/pw3s6nl62t5ushfk",
  "KeyCheckValue": "08D7B4"
}
```

Hasilkan dan verifikasi MAC

Message Authentication Codes (MAC) biasanya digunakan untuk mengautentikasi integritas pesan (apakah sudah dimodifikasi). Hash kriptografi seperti HMAC (Hash Based Message Authentication

Code), CBC-MAC dan CMAC (Cipher-based Message Authentication Code) juga memberikan jaminan tambahan kepada pengirim MAC dengan menggunakan kriptografi. HMAC didasarkan pada fungsi hash sementara CMAC didasarkan pada blok cipher.

Semua algoritma MAC dari layanan ini menggabungkan fungsi hash kriptografi dan kunci rahasia bersama. Mereka mengambil pesan dan kunci rahasia, seperti materi kunci dalam kunci, dan mengembalikan tag atau mac unik. Jika bahkan satu karakter pesan berubah, atau jika kunci rahasia berubah, tag yang dihasilkan sama sekali berbeda. Dengan membutuhkan kunci rahasia, MAC kriptografi juga memberikan keaslian; tidak mungkin untuk menghasilkan mac identik tanpa kunci rahasia. MAC kriptografi kadang-kadang disebut tanda tangan simetris, karena mereka bekerja seperti tanda tangan digital, tetapi menggunakan satu kunci untuk penandatanganan dan verifikasi.

AWSKriptografi Pembayaran mendukung beberapa jenis MAC:

ISO9797 ALGORITMA 1

Dilambangkan dengan ISO9797_ALGORITHM1 `KeyUsage`

ISO9797 ALGORITMA 3 (MAC Eceran)

Dilambangkan dengan ISO9797_ALGORITHM3 `KeyUsage`

ISO9797 ALGORITMA 5 (CMAC)

Dilambangkan dengan dari `KeyUsage` TR31_M6_ISO_9797_5_CMAC_KEY

HMAC

Dilambangkan dengan `KeyUsage` TR31_M7_HMAC_KEY termasuk HMAC_SHA224, HMAC_SHA256, HMAC_SHA384 dan HMAC_SHA512

Topik

- Menghasilkan MAC
- Verifikasi MAC

Menghasilkan MAC

Generate MAC API digunakan untuk mengautentikasi data terkait kartu, seperti melacak data dari strip magnetik kartu, dengan menggunakan nilai data yang diketahui untuk menghasilkan MAC (Message Authentication Code) untuk validasi data antara pihak pengirim dan penerima. Data

yang digunakan untuk menghasilkan MAC termasuk data pesan, kunci enkripsi MAC rahasia dan algoritma MAC untuk menghasilkan nilai MAC yang unik untuk transmisi. Pihak penerima MAC akan menggunakan data pesan MAC yang sama, kunci enkripsi MAC, dan algoritma untuk mereproduksi nilai MAC lain untuk perbandingan dan otentikasi data. Bahkan jika satu karakter pesan berubah atau kunci MAC yang digunakan untuk verifikasi tidak identik, nilai MAC yang dihasilkan berbeda. API mendukung kunci enkripsi DUPKT MAC, HMAC dan EMV MAC untuk operasi ini.

Nilai masukan untuk `message-data` harus data HexBinary.

Dalam contoh ini, kita akan menghasilkan HMAC (Hash Based Message Authentication Code) untuk otentikasi data kartu menggunakan algoritma `HMAC_SHA256` HMAC dan kunci enkripsi HMAC. Kuncinya harus KeyUsage disetel ke `TR31_M7_HMAC_KEY` dan KeyModesOfUse ke`Generate`. Kunci MAC dapat dibuat dengan Kriptografi AWS Pembayaran dengan menelepon CreateKeyatau diimpor dengan menelepon ImportKey.

Example

```
$ aws payment-cryptography-data generate-mac \
    --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
qnobl5lghrzunce6 \
    --message-data
 "3b3130383834393030313037333934313533d32343038323236303030373030303f33" \
    --generation-attributes Algorithm=HMAC_SHA256
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
qnobl5lghrzunce6,
    "KeyCheckValue": "2976E7",
    "Mac": "ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C"
}
```

Verifikasi MAC

Verifikasi MAC API digunakan untuk memverifikasi MAC (Kode Otentikasi Pesan) untuk otentikasi data terkait kartu. Itu harus menggunakan kunci enkripsi yang sama yang digunakan selama menghasilkan MAC untuk menghasilkan kembali nilai MAC untuk otentikasi. Kunci enkripsi MAC dapat dibuat dengan Kriptografi AWS Pembayaran dengan menelepon CreateKeyatau diimpor dengan menelepon ImportKey. API mendukung kunci enkripsi DUPKT MAC, HMAC dan EMV MAC untuk operasi ini.

Jika nilai diverifikasi, maka parameter respons `MacDataVerificationSuccessful` akan kembali`Http/200`, jika tidak `Http/400` dengan pesan yang menunjukkan itu`Mac verification failed`.

Dalam contoh ini, kami akan memverifikasi HMAC (Hash Based Message Authentication Code) untuk otentikasi data kartu menggunakan algoritma `HMAC_SHA256` HMAC dan kunci enkripsi HMAC. Kuncinya harus KeyUsage disetel ke `TR31_M7_HMAC_KEY` dan KeyModesOfUse ke`Verify`.

Example

```
$ aws payment-cryptography-data verify-mac \
     --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
qnobl5lghrzunce6 \
     --message-data
 "3b343038383439303031303733393431353d323430383232363030303730303f33" \
     --verification-attributes='Algorithm=HMAC_SHA256' \
     --mac ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
qnobl5lghrzunce6,
    "KeyCheckValue": "2976E7",
}
```

Kunci yang valid untuk operasi kriptografi

Kunci tertentu hanya dapat digunakan untuk operasi tertentu. Selain itu, beberapa operasi dapat membatasi mode penggunaan kunci untuk kunci. Silakan lihat tabel berikut untuk kombinasi yang diizinkan.

> **Note**
>
> Kombinasi tertentu, meskipun diizinkan, dapat menciptakan situasi yang tidak dapat digunakan seperti menghasilkan kode CVV `(generate)` tetapi kemudian tidak dapat memverifikasinya. `(verify)`

Topik

- GenerateCardData
- VerifyCardData
- GeneratePinData (untuk skema VISA/ABA)
- GeneratePinData (untukIBM3624)
- VerifyPinData (untuk skema VISA/ABA)
- VerifyPinData (untukIBM3624)
- Dekripsi Data
- Enkripsi Data
- Terjemahkan Pin Data
- Hasilkan/Verifikasi MAC
- VerifyAuthRequestCryptogram
- Kunci Impor/Ekspor
- Jenis kunci yang tidak digunakan

GenerateCardData

Titik Akhir API	Operasi atau Algoritma Kriptografi	Penggunaan Kunci yang Diizinkan	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
GenerateCardData	• AMEX_CARD_SECURITY_CODE_VERSION_1 • AMEX_CARD_SECURITY_CODE_VERSION_2	TR31_C0_CARD_VERIFICATION_KEY	• TDES_2KUNCI • TDES_3KUNCI	{Hasilkan = benar}, {Hasilkan = benar, Verifikasi = benar}
GenerateCardData	• CARD_VERIFICATION_VALUE_1	TR31_C0_CARD_VERIFICATION_KEY	• TDES_2KUNCI	{Hasilkan = benar}, {Hasilkan =

Titik Akhir API	Operasi atau Algoritma Kriptografi	Penggunaan Kunci yang Diizinkan	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
	• CARD_VERIFICATION_VALUE_2			benar, Verifikasi = benar}
GenerateCardData	• CARDHOLDER_AUTHENTICATION_VERIFICATION_VALUE	TR31_E6_EMV_MKEY_OTHER	• TDES_2KUNCI	{ DeriveKey = benar}
GenerateCardData	• DYNAMIC_CARD_VERIFICATION_CODE	TR31_E4_EMV_MKEY_DYNAMIC_NUMBERS	• TDES_2KUNCI	{ DeriveKey = benar}
GenerateCardData	• DYNAMIC_CARD_VERIFICATION_VALUE	TR31_E6_EMV_MKEY_OTHER	• TDES_2KUNCI	{ DeriveKey = benar}

VerifyCardData

Operasi atau Algoritma Kriptografi	Penggunaan Kunci yang Diizinkan	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
• AMEX_CARD_SECURITY_CODE_VERSION_1	TR31_C0_CARD_VERIFICATION_KEY	• TDES_2KUNCI • TDES_3KUNCI	{Hasilkan = benar}, {Hasilkan = benar, Verifikasi = benar}

Operasi atau Algoritma Kriptografi	Penggunaan Kunci yang Diizinkan	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
• AMEX_CARD_SECURITY_CODE_VERSION_2			
• CARD_VERIFICATION_VALUE_1 • CARD_VERIFICATION_VALUE_2	TR31_C0_CARD_VERIFICATION_KEY	• TDES_2KUNCI	{Hasilkan = benar}, {Hasilkan = benar, Verifikasi = benar}
• CARDHOLDER_AUTHENTICATION_VERIFICATION_VALUE	TR31_E6_EMV_MKEY_OTHER	• TDES_2KUNCI	{ DeriveKey = benar}
• DYNAMIC_CARD_VERIFICATION_CODE	TR31_E4_EMV_MKEY_DYNAMIC_NUMBERS	• TDES_2KUNCI	{ DeriveKey = benar}
• DYNAMIC_CARD_VERIFICATION_VALUE	TR31_E6_EMV_MKEY_OTHER	• TDES_2KUNCI	{ DeriveKey = benar}

GeneratePinData (untuk skema VISA/ABA)

`VISA_PIN or VISA_PIN_VERIFICATION_VALUE`

Tipe Kunci	Penggunaan Kunci yang Diizinkan	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
Kunci Enkripsi PIN	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KUNCI • TDES_3KUNCI	• {Encrypt = true, Wrap = true} • {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true} • { NoRestrictions = benar}
Kunci Pembuatan PIN	TR31_V2_VISA_PIN_VERIFICATION_KEY	• TDES_3KUNCI	• {Menghasilkan = benar} • {Hasilkan = benar, Verifikasi = benar}

GeneratePinData (untuk**IBM3624**)

`IBM3624_PIN_OFFSET,IBM3624_NATURAL_PIN,IBM3624_RANDOM_PIN, IBM3624_PIN_FROM_OFFSET)`

Tipe Kunci	Penggunaan Kunci yang Diizinkan	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
Kunci Enkripsi PIN	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KUNCI • TDES_3KUNCI	Untuk IBM3624_NATURAL_PIN, IBM3624_RANDOM_PIN, IBM3624_PIN_FROM_OFFSET • {Encrypt = true, Wrap = true}

Tipe Kunci	Penggunaan Kunci yang Diizinkan	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
			• {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true} • { NoRestrictions = benar} Untuk IBM3624_PIN_OFFSET • {Encrypt = true, Unwrap = true} • {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true} • { NoRestrictions = benar}
Kunci Pembuatan PIN	TR31_V1_IBM3624_PIN_VERIFICATION_KEY	• TDES_3KUNCI	• {Menghasilkan = benar} • {Hasilkan = benar, Verifikasi = benar}

VerifyPinData (untuk skema VISA/ABA)

`VISA_PIN`

Tipe Kunci	Penggunaan Kunci yang Diizinkan	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
Kunci Enkripsi PIN	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KUNCI • TDES_3KUNCI	• {Dekripsi = benar, Buka bungkus = benar} • {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true} • { NoRestrictions = benar}
Kunci Pembuatan PIN	TR31_V2_VISA_PIN_VERIFICATION_KEY	• TDES_3KUNCI	• {Verifikasi = benar} • {Hasilkan = benar, Verifikasi = benar}

VerifyPinData (untuk**IBM3624**)

`IBM3624_PIN_OFFSET,IBM3624_NATURAL_PIN,IBM3624_RANDOM_PIN, IBM3624_PIN_FROM_OFFSET)`

Tipe Kunci	Penggunaan Kunci yang Diizinkan	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
Kunci Enkripsi PIN	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KUNCI • TDES_3KUNCI	Untuk IBM3624_NATURAL_PIN, IBM3624_RANDOM_PIN, IBM3624_PIN_FROM_OFFSET

Tipe Kunci	Penggunaan Kunci yang Diizinkan	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
			• {Dekripsi = benar, Buka bungkus = benar} • {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true} • { NoRestrictions = benar}
Kunci Verifikasi PIN	TR31_V1_IBM3624_PIN_VERIFICATION_KEY	• TDES_3KUNCI	• {Verifikasi = benar} • {Hasilkan = benar, Verifikasi = benar}

Dekripsi Data

Tipe Kunci	Penggunaan Kunci yang Diizinkan	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
DUKPT	TR31_B0_BASE_DERIVATION_KEY	• TDES_2KUNCI • AES_128 • AES_192 • AES_256	• { DeriveKey = benar} • { NoRestrictions = benar}
EMV	TR31_E1_EMV_MKEY_CONFIDENTIALITY TR31_E6_EMV_MKEY_OTHER	• TDES_2KUNCI	• { DeriveKey = benar}

Tipe Kunci	Penggunaan Kunci yang Diizinkan	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
RSA	TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION	• RSA_2048 • RSA_3072 • RSA_4096	• {Dekripsi = benar, buka bungkus=Benar} • {encrypt=True, wrap=True, Dekripsi = true, buka bungkus=Benar}
Tombol simetris	TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY	• TDES_2KUNCI • TDES_3KUNCI • AES_128 • AES_192 • AES_256	• {Dekripsi = benar, buka bungkus=Benar} • {encrypt=True, wrap=True, Dekripsi = true, buka bungkus=Benar} • { NoRestrictions = benar}

Enkripsi Data

Tipe Kunci	Penggunaan Kunci yang Diizinkan	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
DUKPT	TR31_B0_BASE_DERIVATION_KEY	• TDES_2KUNCI • AES_128 • AES_192 • AES_256	• { DeriveKey = benar} • { NoRestrictions = benar}

Tipe Kunci	Penggunaan Kunci yang Diizinkan	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
EMV	TR31_E1_EMV_MKEY_CONFIDENTIALITY TR31_E6_EMV_MKEY_OTHER	• TDES_2KUNCI	• { DeriveKey = benar}
RSA	TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION	• RSA_2048 • RSA_3072 • RSA_4096	• {Enkripsi = benar, bungkus = benar} • {encrypt=True, wrap=True, Dekripsi = true, buka bungkus=Benar}
Tombol simetris	TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY	• TDES_2KUNCI • TDES_3KUNCI • AES_128 • AES_192 • AES_256	• {Enkripsi = benar, bungkus = benar} • {encrypt=True, wrap=True, Dekripsi = true, buka bungkus=Benar} • { NoRestrictions = benar}

Terjemahkan Pin Data

Arahan	Tipe Kunci	Penggunaan Kunci yang Diizinkan	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
Sumber Data Masuk	DUKPT	TR31_B0_BASE_DERIVATION_KEY	• TDES_2KUNCI • AES_128 • AES_192 • AES_256	• { DeriveKey = benar} • { NoRestrictions = benar}
Sumber Data Masuk	Non-DUKPT (PEK, AWK, IWK, dll)	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KUNCI • TDES_3KUNCI • AES_128 • AES_192 • AES_256	• {Dekripsi = benar, Buka bungkus = benar} • {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true} • { NoRestrictions = benar}
Target Data Keluar	DUKPT	TR31_B0_BASE_DERIVATION_KEY	• TDES_2KUNCI • AES_128 • AES_192 • AES_256	• { DeriveKey = benar} • { NoRestrictions = benar}
Target Data Keluar	Non-DUKPT (PEK, IWK, AWK, dll)	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KUNCI	• {Encrypt = true, Wrap = true}

Arahan	Tipe Kunci	Penggunaan Kunci yang Diizinkan	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
			• TDES_3KUNCI • AES_128 • AES_192 • AES_256	• {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true} • { NoRestrictions = benar}

Hasilkan/Verifikasi MAC

Kunci MAC digunakan untuk membuat hash kriptografi dari pesan/badan data. Tidak disarankan untuk membuat kunci dengan mode penggunaan kunci terbatas karena Anda tidak akan dapat melakukan operasi pencocokan. Namun, Anda dapat mengimpor/mengekspor kunci hanya dengan satu operasi jika sistem lain dimaksudkan untuk melakukan separuh lainnya dari pasangan operasi.

Penggunaan Kunci yang Diizinkan	Penggunaan Kunci yang Diizinkan	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
Kunci MAC	TR31_M1_ISO_9797_1_MAC_KEY	• TDES_2KUNCI • TDES_3KUNCI	• {Menghasilkan = benar} • {Hasilkan = benar, Verifikasi = benar} • {Verifikasi = benar} • {Menghasilkan = benar}
Kunci MAC (MAC Ritel)	TR31_M1_ISO_9797_3_MAC_KEY	• TDES_2KUNCI • TDES_3KUNCI	• {Menghasilkan = benar}

Penggunaan Kunci yang Diizinkan	Penggunaan Kunci yang Diizinkan	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
			• {Hasilkan = benar, Verifikasi = benar} • {Verifikasi = benar} • {Menghasilkan = benar}
Kunci MAC (CMAC)	TR31_M6_ISO_9797_5_CMAC_KEY	• TDES_2KUNCI • TDES_3KUNCI • AES_128 • AES_192 • AES_256	• {Menghasilkan = benar} • {Hasilkan = benar, Verifikasi = benar} • {Verifikasi = benar} • {Menghasilkan = benar}
Kunci MAC (HMAC)	TR31_M7_HMAC_KEY	• TDES_2KUNCI • TDES_3KUNCI • AES_128 • AES_192 • AES_256	• {Menghasilkan = benar} • {Hasilkan = benar, Verifikasi = benar} • {Verifikasi = benar} • {Menghasilkan = benar}

VerifyAuthRequestCryptogram

Penggunaan Kunci yang Diizinkan	Opsi EMV	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
• OPSI A • OPSI B	TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS	• TDES_2KUNCI	• { DeriveKey = benar}

Kunci Impor/Ekspor

Tipe operasi	Penggunaan Kunci yang Diizinkan	Algoritma Kunci yang Diizinkan	Kombinasi yang diizinkan dari mode penggunaan utama
Kunci Pembungkus TR-31	TR31_K1_KEY_BLOCK_PROTECTION_KEY TR31_K0_KEY_ENCRYPTION_KEY	• TDES_2KUNCI • TDES_3KUNCI • AES_128	• {Encrypt = true, Wrap = true} (hanya ekspor) • {Decrypt = true, Unwrap = true} (hanya impor) • {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}
Impor CA tepercaya	TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE	• RSA_2048 • RSA_3072 • RSA_4096	• {Verifikasi = benar}
Impor sertifikat kunci publik untuk enkripsi asimetris	TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION	• RSA_2048 • RSA_3072 • RSA_4096	• {Encrypt=true, wrap=True}

Jenis kunci yang tidak digunakan

Jenis kunci berikut saat ini tidak digunakan oleh Kriptografi AWS Pembayaran

- TR31_P1_PIN_GENERATION_KEY
- TR31_K3_ASYMMETRIC_KEY_FOR_KEY_AGREEMENT

Keamanan dalam Kriptografi AWS Pembayaran

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan cloud dan keamanan dalam cloud:

- Keamanan cloud —AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program AWS Kepatuhan Program AWS Kepatuhan . Untuk mempelajari tentang program kepatuhan yang berlaku untuk Kriptografi AWS Pembayaran, lihat AWS Services in Scope by Compliance Program .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Topik ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Kriptografi AWS Pembayaran. Ini menunjukkan kepada Anda cara mengkonfigurasi Kriptografi AWS Pembayaran untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga belajar cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya Kriptografi AWS Pembayaran Anda.

Topik

Perlindungan data dalam Kriptografi AWS Pembayaran

Model tanggung jawab AWS bersama model berlaku untuk perlindungan data dalam Kriptografi AWS Pembayaran. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam Pertanyaan Umum Privasi Data. Lihat informasi tentang perlindungan data di Eropa di pos blog Model Tanggung Jawab Bersama dan GDPR AWS di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensyal dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan sumber daya. AWS Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi yang lebih lengkap tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-2.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan Kriptografi AWS Pembayaran atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami

sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

AWS Payment Cryptography menyimpan dan melindungi kunci enkripsi pembayaran Anda agar sangat tersedia sekaligus memberi Anda kontrol akses yang kuat dan fleksibel.

Topik

Melindungi bahan utama

Secara default, AWS Payment Cryptography melindungi materi kunci kriptografi untuk kunci pembayaran yang dikelola oleh layanan. Selain itu, AWS Payment Cryptography menawarkan opsi untuk mengimpor materi utama yang dibuat di luar layanan. Untuk detail teknis tentang kunci pembayaran dan materi utama, lihat Detail Kriptografi Kriptografi Pembayaran AWS.

Enkripsi data

Data dalam AWS Payment Cryptography terdiri dari kunci AWS Payment Cryptography, materi kunci enkripsi yang mereka wakili, dan atribut penggunaannya. Materi utama ada dalam teks biasa hanya dalam modul keamanan perangkat keras AWS Payment Cryptography (HSM) dan hanya saat digunakan. Jika tidak, bahan dan atribut utama dienkripsi dan disimpan dalam penyimpanan persisten yang tahan lama.

Materi utama yang dihasilkan atau dimuat oleh AWS Payment Cryptography untuk kunci pembayaran tidak pernah meninggalkan batas AWS Payment Cryptography HSM yang tidak terenkripsi. Ini dapat diekspor dienkripsi oleh operasi AWS Payment Cryptography API.

Enkripsi diam

AWS Payment Cryptography menghasilkan materi utama untuk kunci pembayaran di HSM yang terdaftar di PCI PTS HSM. Saat tidak digunakan, bahan kunci dienkripsi oleh kunci HSM dan ditulis

ke penyimpanan yang tahan lama dan persisten. Materi utama untuk kunci Kriptografi Pembayaran dan kunci enkripsi yang melindungi materi kunci tidak pernah meninggalkan HSM dalam bentuk teks biasa.

Enkripsi dan pengelolaan materi kunci untuk kunci Kriptografi Pembayaran ditangani sepenuhnya oleh layanan.

Untuk detail selengkapnya, lihat AWS Key Management Service Cryptographic Details.

Enkripsi bergerak

Materi utama yang dihasilkan atau dimuat oleh AWS Payment Cryptography untuk kunci pembayaran tidak pernah diekspor atau ditransmisikan dalam operasi AWS Payment Cryptography API dalam cleartext. AWS Payment Cryptography menggunakan pengidentifikasi kunci untuk mewakili kunci dalam operasi API.

Namun, beberapa operasi AWS Payment Cryptography API mengekspor kunci yang dienkripsi oleh kunci pertukaran kunci yang sebelumnya dibagikan atau asimetris. Selain itu, pelanggan dapat menggunakan operasi API untuk mengimpor materi kunci terenkripsi untuk kunci pembayaran.

Semua panggilan AWS Payment Cryptography API harus ditandatangani dan ditransmisikan menggunakan Transport Layer Security (TLS). AWS Payment Cryptography memerlukan versi TLS dan cipher suite yang didefinisikan oleh PCI sebagai “kriptografi kuat”. Semua titik akhir layanan mendukung TLS 1.0-1.3 dan TLS pasca-kuantum hibrida.

Untuk detail selengkapnya, lihat AWS Key Management Service Cryptographic Details.

Privasi lalu lintas antar jaringan

AWS Payment Cryptography mendukung AWS Management Console dan serangkaian operasi API yang memungkinkan Anda membuat dan mengelola kunci pembayaran dan menggunakannya dalam operasi kriptografi.

AWS Pament Cryptography mendukung dua opsi konektivitas jaringan dari jaringan pribadi Anda ke AWS.

- Koneksi VPN IPsec melalui internet.
- AWS Direct Connect, yang menautkan jaringan internal Anda ke lokasi AWS Direct Connect melalui kabel serat optik Ethernet standar.

Semua panggilan API Kriptografi Pembayaran harus ditandatangani dan ditransmisikan menggunakan Transport Layer Security (TLS). Panggilan juga memerlukan suite penyandian modern yang mendukung kerahasiaan penerusan sempurna. Lalu lintas ke modul keamanan perangkat keras (HSM) yang menyimpan materi kunci untuk kunci pembayaran hanya diizinkan dari host AWS Payment Cryptography API yang diketahui melalui jaringan internal AWS.

Untuk terhubung langsung ke AWS Payment Cryptography dari virtual private cloud (VPC) Anda tanpa mengirimkan lalu lintas melalui internet publik, gunakan titik akhir VPC, yang didukung oleh AWS. PrivateLink Untuk informasi selengkapnya, lihat Menghubungkan ke Kriptografi Pembayaran AWS melalui titik akhir VPC.

AWS Payment Cryptography juga mendukung opsi pertukaran kunci pasca-kuantum hybrid untuk protokol enkripsi jaringan Transport Layer Security (TLS). Anda dapat menggunakan opsi ini dengan TLS saat Anda terhubung ke titik akhir AWS Payment Cryptography API.

Ketahanan dalam AWS Kriptografi Pembayaran

AWS Infrastruktur global dibangun di sekitar AWS Wilayah dan Availability Zone. Wilayah memberikan beberapa Zona Ketersediaan yang terpisah dan terisolasi secara fisik, yang terkoneksi melalui jaringan latensi rendah, throughput tinggi, dan sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat Infrastruktur AWS Global.

Isolasi regional

AWS Payment Cryptography adalah layanan Regional yang tersedia di beberapa wilayah.

Desain Kriptografi Pembayaran AWS yang terisolasi secara regional memastikan bahwa masalah ketersediaan di satu Wilayah AWS tidak dapat memengaruhi operasi Kriptografi Pembayaran AWS di Wilayah lain mana pun. AWS Payment Cryptography dirancang untuk memastikan nol waktu henti yang direncanakan, dengan semua pembaruan perangkat lunak dan operasi penskalaan dilakukan dengan mulus dan tanpa terasa.

AWS Payment Cryptography Service Level Agreement (SLA) mencakup komitmen layanan sebesar 99,99% untuk semua API Kriptografi Pembayaran. Untuk memenuhi komitmen ini, AWS Payment

Cryptography memastikan bahwa semua data dan informasi otorisasi yang diperlukan untuk menjalankan permintaan API tersedia di semua host regional yang menerima permintaan tersebut.

Infrastruktur Kriptografi Pembayaran AWS direplikasi di setidaknya tiga Availability Zone (AZ) di setiap Wilayah. Untuk memastikan bahwa beberapa kegagalan host tidak memengaruhi kinerja Kriptografi Pembayaran AWS, Kriptografi Pembayaran AWS dirancang untuk melayani lalu lintas pelanggan dari AZ mana pun di Wilayah.

Perubahan yang Anda buat pada properti atau izin kunci pembayaran direplikasi ke semua host di Wilayah untuk memastikan bahwa permintaan berikutnya dapat diproses dengan benar oleh host mana pun di Wilayah. Permintaan untuk operasi kriptografi menggunakan kunci pembayaran Anda diteruskan ke armada modul keamanan perangkat keras AWS Payment Cryptography (HSM), yang mana pun dapat melakukan operasi dengan kunci pembayaran.

Desain multi-penyewa

Desain multi-tenant AWS Payment Cryptography memungkinkannya memenuhi ketersediaan SLA, dan mempertahankan tingkat permintaan yang tinggi, sekaligus melindungi kerahasiaan kunci dan data Anda.

Beberapa mekanisme penegakan integritas digunakan untuk memastikan bahwa kunci pembayaran yang Anda tentukan untuk operasi kriptografi selalu yang digunakan.

Materi kunci plaintext untuk kunci Kriptografi Pembayaran Anda dilindungi secara luas. Materi utama dienkripsi di HSM segera setelah dibuat, dan bahan kunci terenkripsi segera dipindahkan ke penyimpanan yang aman. Kunci terenkripsi diambil dan didekripsi dalam HSM tepat pada waktunya untuk digunakan. Kunci plaintext tetap dalam memori HSM hanya untuk waktu yang dibutuhkan untuk menyelesaikan operasi kriptografi. Materi kunci Plaintext tidak pernah meninggalkan HSM; itu tidak pernah ditulis ke penyimpanan persisten.

Untuk informasi selengkapnya tentang mekanisme yang digunakan AWS Payment Cryptography untuk mengamankan kunci Anda, lihat AWS Payment Cryptography Cryptography Details.

Keamanan infrastruktur di AWS Payment Cryptography

Sebagai layanan terkelola, AWS Payment Cryptography dilindungi oleh prosedur keamanan jaringan AWS global yang dijelaskan dalam whitepaper Amazon Web Services: Tinjauan Proses Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses AWS Payment Cryptography melalui jaringan. Klien harus mendukung Keamanan Lapisan Pengangkutan (TLS)

1.2 atau versi yang lebih baru. Klien juga harus mendukung cipher suite dengan perfect forward secrecy (PFS) seperti Ephemeral Diffie-Hellman (DHE) atau Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Sebagian besar sistem-sistem modern seperti Java 7 dan versi yang lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang dikaitkan dengan pengguna utama IAM. Atau Anda dapat menggunakan AWS Security Token Service (AWS STS) untuk membuat kredensial keamanan sementara untuk menandatangani permintaan.

Isolasi host fisik

Keamanan infrastruktur fisik yang digunakan AWS Payment Cryptography tunduk pada kontrol yang dijelaskan di bagian Keamanan Fisik dan Lingkungan Amazon Web Services: Tinjauan Proses Keamanan. Anda dapat menemukan lebih banyak detail dalam laporan kepatuhan dan temuan audit pihak ketiga yang tercantum di bagian sebelumnya.

AWS Payment Cryptography didukung oleh modul keamanan perangkat keras (HSM) khusus yang terdaftar di commercial-off-the-shelf PCI PTS HSM. Materi utama untuk kunci Kriptografi Pembayaran AWS disimpan hanya dalam memori volatile pada HSM, dan hanya saat kunci Kriptografi Pembayaran sedang digunakan. HSM berada di rak yang dikendalikan akses di dalam pusat data Amazon yang memberlakukan kontrol ganda untuk akses fisik apa pun. Untuk informasi terperinci tentang pengoperasian AWS Payment Cryptography HSM, lihat Detail Kriptografi Kriptografi Pembayaran AWS.

Menghubungkan ke Kriptografi AWS Pembayaran melalui titik akhir VPC

Anda dapat terhubung langsung ke Kriptografi AWS Pembayaran melalui titik akhir antarmuka pribadi di cloud pribadi virtual (VPC) Anda. Saat Anda menggunakan titik akhir VPC antarmuka, komunikasi antara VPC dan Kriptografi AWS Pembayaran dilakukan sepenuhnya di dalam jaringan. AWS

AWS Kriptografi Pembayaran mendukung titik akhir Amazon Virtual Private Cloud (Amazon VPC) yang didukung oleh. AWS PrivateLink Masing-masing VPC endpoint diwakili oleh satu atau lebih Antarmuka Jaringan Elastis (ENI) dengan alamat IP privat di subnet VPC Anda.

Titik akhir VPC antarmuka menghubungkan VPC Anda langsung ke Kriptografi AWS Pembayaran tanpa gateway internet, perangkat NAT, koneksi VPN, atau koneksi. AWS Direct Connect Instans

di VPC Anda tidak memerlukan alamat IP publik untuk berkomunikasi AWS dengan Kriptografi Pembayaran.

Wilayah

AWS Kriptografi Pembayaran mendukung kebijakan titik akhir VPC dan titik akhir VPC Wilayah AWS di mana Kriptografi Pembayaran didukung.AWS

Topik

Pertimbangan untuk titik akhir AWS VPC Kriptografi Pembayaran

Sebelum Anda menyiapkan titik akhir VPC antarmuka untuk Kriptografi AWS Pembayaran, tinjau topik properti dan batasan titik akhir Antarmuka di Panduan.AWS PrivateLink

AWS Dukungan Kriptografi Pembayaran untuk titik akhir VPC mencakup yang berikut ini.

- Anda dapat menggunakan titik akhir VPC Anda untuk memanggil semua operasi AWS Payment Cryptography Controlplane dan AWS operasi Payment Cryptography Dataplane dari VPC.
- Anda dapat membuat titik akhir VPC antarmuka yang terhubung ke titik akhir wilayah Kriptografi AWS Pembayaran.
- AWS Kriptografi Pembayaran terdiri dari bidang kontrol dan bidang data. Anda dapat memilih untuk mengatur satu atau kedua sub-layanan tetapi masing-masing dikonfigurasi secara terpisah.
- Anda dapat menggunakan AWS CloudTrail log untuk mengaudit penggunaan kunci Kriptografi AWS Pembayaran melalui titik akhir VPC. Lihat perinciannya di Mencatat VPC endpoint Anda.

Membuat titik akhir VPC untuk Kriptografi Pembayaran AWS

Anda dapat membuat titik akhir VPC untuk Kriptografi AWS Pembayaran dengan menggunakan konsol VPC Amazon atau API VPC Amazon. Untuk informasi selengkapnya, lihat Membuat titik akhir antarmuka di AWS PrivateLink Panduan.

- Untuk membuat titik akhir VPC untuk Kriptografi AWS Pembayaran, gunakan nama layanan berikut:

```
com.amazonaws.region.payment-cryptography.controlplane
```

```
com.amazonaws.region.payment-cryptography.dataplane
```

  Misalnya, di Wilayah AS Barat (Oregon) (`us-west-2`), nama layanannya adalah:

```
com.amazonaws.us-west-2.payment-cryptography.controlplane
```

```
com.amazonaws.us-west-2.payment-cryptography.dataplane
```

Untuk mempermudah penggunaan titik akhir VPC, Anda dapat mengaktifkan nama DNS pribadi untuk titik akhir VPC Anda. Jika Anda memilih opsi Aktifkan Nama DNS, nama host DNS Kriptografi AWS Pembayaran standar akan diselesaikan ke titik akhir VPC Anda. Misalnya, `https://controlplane.payment-cryptography.us-west-2.amazonaws.com` akan menyelesaikan ke titik akhir VPC yang terhubung ke nama layanan. `com.amazonaws.us-west-2.payment-cryptography.controlplane`

Opsi ini mempermudah untuk menggunakan VPC endpoint. AWS SDK dan AWS CLI menggunakan standar AWS Payment Cryptography DNS hostname secara default, sehingga Anda tidak perlu menentukan URL endpoint VPC dalam aplikasi dan perintah.

Untuk informasi selengkapnya, lihat Mengakses layanan melalui titik akhir antarmuka di Panduan.AWS PrivateLink

Menghubungkan ke titik akhir AWS VPC Kriptografi Pembayaran

Anda dapat terhubung ke Kriptografi AWS Pembayaran melalui titik akhir VPC dengan menggunakan SDK, AWS atau. AWS CLI AWS Tools for PowerShell Untuk menentukan VPC endpoint, gunakan nama DNS-nya.

Misalnya, perintah kunci-daftar ini menggunakan parameter `endpoint-url` untuk menentukan VPC endpoint. Untuk menggunakan perintah seperti ini, ganti contoh ID VPC endpoint dengan yang ada di akun Anda.

```
$ aws payment-cryptography list-keys --endpoint-url
```

Jika Anda mengaktifkan nama host privat ketika Anda membuat VPC endpoint Anda, Anda tidak perlu menentukan URL VPC endpoint di perintah CLI atau konfigurasi aplikasi. Nama host DNS Kriptografi AWS Pembayaran standar diselesaikan ke titik akhir VPC Anda. SDK AWS CLI dan SDK menggunakan nama host ini secara default, sehingga Anda dapat mulai menggunakan titik akhir VPC untuk terhubung ke titik akhir regional Kriptografi AWS Pembayaran tanpa mengubah apa pun dalam skrip dan aplikasi Anda.

Untuk menggunakan nama host pribadi, `enableDnsSupport` atribut `enableDnsHostnames` dan VPC Anda harus disetel ke. `true` Untuk mengatur atribut ini, gunakan operasi ModifyVpcAtribut. Untuk detailnya, lihat Melihat dan memperbarui atribut DNS untuk VPC Anda di Panduan Pengguna Amazon VPC.

Mengontrol akses ke VPC endpoint

Untuk mengontrol akses ke titik akhir VPC Anda untuk Kriptografi AWS Pembayaran, lampirkan kebijakan titik akhir VPC ke titik akhir VPC Anda. Kebijakan endpoint menentukan apakah prinsipal dapat menggunakan titik akhir VPC untuk memanggil operasi Kriptografi Pembayaran dengan sumber daya Kriptografi AWS Pembayaran tertentu. AWS

Anda dapat membuat kebijakan VPC endpoint ketika Anda membuat titik akhir Anda, dan Anda dapat mengubah kebijakan VPC endpoint setiap saat. Gunakan konsol manajemen VPC, atau operasi Endpoint atau CreateVpcEndpoint. ModifyVpc Anda juga dapat membuat dan mengubah kebijakan titik akhir VPC dengan menggunakan templat. AWS CloudFormation Untuk bantuan menggunakan konsol manajemen VPC, lihat Membuat titik akhir antarmuka dan Memodifikasi titik akhir antarmuka dalam Panduan.AWS PrivateLink

Topik

Tentang kebijakan VPC endpoint

Agar permintaan Kriptografi AWS Pembayaran yang menggunakan titik akhir VPC berhasil, prinsipal memerlukan izin dari dua sumber:

- Kebijakan berbasis identitas harus memberikan izin utama untuk memanggil operasi pada sumber daya (kunci Kriptografi AWS Pembayaran atau alias).
- Kebijakan VPC endpoint harus memberikan prinsipal izin untuk menggunakan titik akhir untuk membuat permintaan.

Misalnya, kebijakan kunci mungkin memberikan izin utama untuk memanggil Dekripsi pada kunci Kriptografi AWS Pembayaran tertentu. Namun, kebijakan titik akhir VPC mungkin tidak mengizinkan prinsipal tersebut untuk memanggil `Decrypt` kunci Kriptografi AWS Pembayaran tersebut dengan menggunakan titik akhir.

Atau kebijakan titik akhir VPC mungkin mengizinkan prinsipal untuk menggunakan titik akhir untuk memanggil StopKeyPenggunaan pada kunci Kriptografi Pembayaran tertentu AWS . Tetapi jika prinsipal tidak memiliki izin tersebut dari kebijakan IAM, permintaan gagal.

Kebijakan VPC endpoint default

Setiap VPC endpoint memiliki kebijakan VPC endpoint, tetapi Anda tidak diharuskan untuk menentukan kebijakan. Jika Anda tidak menentukan kebijakan, kebijakan titik akhir default memungkinkan semua operasi oleh semua prinsipal di semua sumber daya pada titik akhir.

Namun, untuk sumber daya Kriptografi AWS Pembayaran, kepala sekolah juga harus memiliki izin untuk memanggil operasi dari kebijakan IAM. Oleh karena itu, dalam praktik, kebijakan default mengatakan bahwa jika prinsipal memiliki izin untuk memanggil operasi pada sumber daya, mereka juga dapat memanggilnya dengan menggunakan titik akhir.

```
{
```

```
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Principal": "*",
      "Resource": "*"
    }
  ]
}
```

Untuk mengizinkan prinsipal menggunakan titik akhir VPC hanya untuk sebagian dari operasi yang diizinkan, buat atau perbarui kebijakan titik akhir VPC.

Membuat kebijakan VPC endpoint

Kebijakan VPC endpoint menentukan apakah prinsipal memiliki izin untuk menggunakan VPC endpoint untuk melakukan operasi pada sumber daya. Untuk sumber daya Kriptografi AWS Pembayaran, kepala sekolah juga harus memiliki izin untuk melakukan operasi dari kebijakan IAM.

Setiap pernyataan kebijakan VPC endpoint memerlukan unsur-unsur berikut:

- Prinsip-prinsip yang dapat melakukan tindakan
- Tindakan yang dapat dilakukan
- Sumber daya yang dapat digunakan untuk mengambil tindakan

Pernyataan kebijakan tidak menentukan VPC endpoint. Sebaliknya, berlaku untuk VPC endpoint di mana kebijakan tersebut terpasang. Untuk informasi selengkapnya, lihat Mengendalikan akses ke layanan dengan titik akhir VPC dalam Panduan Pengguna Amazon VPC.

Berikut ini adalah contoh kebijakan titik akhir VPC untuk AWS Kriptografi Pembayaran. Saat dilampirkan ke titik akhir VPC, kebijakan ini memungkinkan `ExampleUser` untuk menggunakan titik akhir VPC untuk memanggil operasi yang ditentukan pada kunci Kriptografi Pembayaran yang ditentukan. AWS Sebelum menggunakan kebijakan seperti ini, ganti contoh prinsipal dan pengidentifikasi kunci dengan nilai yang valid dari akun Anda.

```
{
   "Statement":[
      {
         "Sid": "AllowDecryptAndView",
         "Principal": {"AWS": "arn:aws:iam::111122223333:user/ExampleUser"},
```

```
         "Effect":"Allow",
         "Action": [
             "payment-cryptography:Decrypt",
             "payment-cryptography:GetKey",
             "payment-cryptography:ListAliases",
             "payment-cryptography:ListKeys",
             "payment-cryptography:GetAlias"
          ],
         "Resource": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h"
      }
   ]
}
```

AWS CloudTrail mencatat semua operasi yang menggunakan titik akhir VPC. Namun, CloudTrail log Anda tidak menyertakan operasi yang diminta oleh kepala sekolah di akun lain atau operasi untuk kunci Kriptografi AWS Pembayaran di akun lain.

Dengan demikian, Anda mungkin ingin membuat kebijakan titik akhir VPC yang mencegah prinsipal di akun eksternal menggunakan titik akhir VPC untuk memanggil operasi Kriptografi AWS Pembayaran apa pun pada kunci apa pun di akun lokal.

Contoh berikut menggunakan aws: PrincipalAccount global condition key untuk menolak akses ke semua prinsipal untuk semua operasi pada semua kunci Kriptografi AWS Pembayaran kecuali prinsipal ada di akun lokal. Sebelum menggunakan kebijakan seperti ini, ganti ID akun contoh dengan yang valid.

```
{
  "Statement": [
    {
      "Sid": "AccessForASpecificAccount",
      "Principal": {"AWS": "*"},
      "Action": "payment-cryptography:*",
      "Effect": "Deny",
      "Resource": "arn:aws:payment-cryptography:*:111122223333:key/*",
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalAccount": "111122223333"
        }
      }
    }
  ]
```

```
}
```

Melihat kebijakan VPC endpoint

Untuk melihat kebijakan titik akhir VPC untuk titik akhir, gunakan konsol manajemen VPC atau operasi Titik Akhir. DescribeVpc

AWS CLI Perintah berikut mendapatkan kebijakan untuk titik akhir dengan ID titik akhir VPC yang ditentukan.

Sebelum menggunakan perintah ini, ganti ID titik akhir contoh dengan yang valid dari akun Anda.

```
$ aws ec2 describe-vpc-endpoints \
--query 'VpcEndpoints[?VpcEndpointId==``].[PolicyDocument]'
--output text
```

Menggunakan VPC endpoint dalam pernyataan kebijakan

Anda dapat mengontrol akses ke sumber daya dan operasi Kriptografi AWS Pembayaran ketika permintaan berasal dari VPC atau menggunakan titik akhir VPC. Untuk melakukannya, gunakan salah satu kebijakan IAM

- Gunakan kunci kondisi `aws:sourceVpce` untuk memberikan atau membatasi akses berdasarkan VPC endpoint.
- Gunakan kunci kondisi `aws:sourceVpc` untuk memberikan atau membatasi akses berdasarkan VPC yang menjadi host endpoint privat.

> **Note**
>
> Kunci `aws:sourceIP` kondisi tidak efektif ketika permintaan berasal dari titik akhir VPC Amazon. Untuk membatasi permintaan ke VPC endpoint, gunakan kunci kondisi `aws:sourceVpce` atau `aws:sourceVpc`. Untuk informasi selengkapnya, lihat Identitas dan manajemen akses untuk titik akhir VPC dan layanan titik akhir VPC di Panduan.AWS PrivateLink

Anda dapat menggunakan kunci kondisi global ini untuk mengontrol akses ke kunci Kriptografi AWS Pembayaran, alias, dan operasi seperti CreateKeyitu tidak bergantung pada sumber daya tertentu.

Misalnya, kebijakan kunci sampel berikut memungkinkan pengguna untuk melakukan operasi kriptografi tertentu dengan kunci Kriptografi AWS Pembayaran hanya ketika permintaan menggunakan titik akhir VPC yang ditentukan, memblokir akses baik dari Internet dan koneksi (jika pengaturan). Ketika pengguna membuat permintaan ke Kriptografi AWS Pembayaran, ID titik akhir VPC dalam permintaan dibandingkan `aws:sourceVpce` dengan nilai kunci kondisi dalam kebijakan. Jika tidak cocok, permintaan ditolak.

Untuk menggunakan kebijakan seperti ini, ganti ID placeholder dan Akun AWS ID titik akhir VPC dengan nilai yang valid untuk akun Anda.

```
{
    "Id": "example-key-1",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable IAM policies",
            "Effect": "Allow",
            "Principal": {"AWS":["111122223333"]},
            "Action": ["payment-cryptography:*"],
            "Resource": "*"
        },
        {
            "Sid": "Restrict usage to my VPC endpoint",
            "Effect": "Deny",
            "Principal": "*",
            "Action": [
                "payment-cryptography:Encrypt",
                "payment-cryptography:Decrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:sourceVpce": ""
                }
            }
        }

    ]
}
```

Anda juga dapat menggunakan tombol `aws:sourceVpc` kondisi untuk membatasi akses ke kunci Kriptografi AWS Pembayaran Anda berdasarkan VPC tempat titik akhir VPC berada.

Kebijakan kunci sampel berikut memungkinkan perintah yang mengelola kunci Kriptografi AWS Pembayaran hanya ketika mereka berasal`vpc-12345678`. Selain itu, ini memungkinkan perintah yang menggunakan kunci Kriptografi AWS Pembayaran untuk operasi kriptografi hanya ketika mereka berasal. `vpc-2b2b2b2b` Anda mungkin menggunakan kebijakan seperti ini jika aplikasi berjalan dalam satu VPC, tetapi Anda menggunakan VPC terisolasi kedua untuk fungsi manajemen.

Untuk menggunakan kebijakan seperti ini, ganti ID placeholder dan Akun AWS ID titik akhir VPC dengan nilai yang valid untuk akun Anda.

```
{
    "Id": "example-key-2",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow administrative actions from vpc-12345678",
            "Effect": "Allow",
            "Principal": {"AWS": "111122223333"},
            "Action": [
                "payment-cryptography:Create*","payment-
cryptography:Encrypt*","payment-cryptography:ImportKey*","payment-
cryptography:GetParametersForImport*",
                "payment-cryptography:TagResource", "payment-
cryptography:UntagResource"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:sourceVpc": "vpc-12345678"
                }
            }
        },
        {
            "Sid": "Allow key usage from vpc-2b2b2b2b",
            "Effect": "Allow",
            "Principal": {"AWS": "111122223333"},
            "Action": [
                "payment-cryptography:Encrypt","payment-cryptography:Decrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:sourceVpc": "vpc-2b2b2b2b"
                }
```

```
            }
        },
        {
            "Sid": "Allow list/read actions from everywhere",
            "Effect": "Allow",
            "Principal": {"AWS": "111122223333"},
            "Action": [
                "payment-cryptography:List*","payment-cryptography:Get*"
            ],
            "Resource": "*",
        }
    ]
}
```

Mencatat VPC endpoint Anda

AWS CloudTrail mencatat semua operasi yang menggunakan titik akhir VPC. Ketika permintaan ke Kriptografi AWS Pembayaran menggunakan titik akhir VPC, ID titik akhir VPC muncul di entri log yang mencatat permintaan AWS CloudTrail tersebut. Anda dapat menggunakan ID titik akhir untuk mengaudit penggunaan titik akhir VPC Kriptografi AWS Pembayaran Anda.

Untuk melindungi VPC Anda, permintaan yang ditolak oleh kebijakan titik akhir VPC, tetapi sebaliknya diizinkan, tidak dicatat. AWS CloudTrail

Misalnya, entri log contoh ini mencatat GenerateMacpermintaan yang menggunakan titik akhir VPC. Bidang `vpcEndpointId` muncul di akhir entri log.

```
{
      "eventVersion": "1.08",
      "userIdentity": {
          "principalId": "TESTXECZ5U9M4LGF2N6Y5:",
          "arn": "arn:aws:sts::111122223333:assumed-role//",
          "accountId": "111122223333",
          "accessKeyId": "TESTXECZ5U2ZULLHHMJG",
          "sessionContext": {
              "sessionIssuer": {
                  "type": "Role",
                  "principalId": "TESTXECZ5U9M4LGF2N6Y5",
                  "arn": "arn:aws:iam::111122223333:role/",
                  "accountId": "111122223333",
                  "userName": ""
              },
```

```
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2024-05-27T19:34:10Z",
                "mfaAuthenticated": "false"
            },
            "ec2RoleDelivery": "2.0"
        }
    },
    "eventTime": "2024-05-27T19:49:54Z",
    "eventSource": "payment-cryptography.amazonaws.com",
    "eventName": "CreateKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "172.31.85.253",
    "userAgent": "aws-cli/2.14.5 Python/3.9.16 Linux/6.1.79-99.167.amzn2023.x86_64
 source/x86_64.amzn.2023 prompt/off command/payment-cryptography.create-key",
    "requestParameters": {
        "keyAttributes": {
            "keyUsage": "TR31_M1_ISO_9797_1_MAC_KEY",
            "keyClass": "SYMMETRIC_KEY",
            "keyAlgorithm": "TDES_2KEY",
            "keyModesOfUse": {
                "encrypt": false,
                "decrypt": false,
                "wrap": false,
                "unwrap": false,
                "generate": true,
                "sign": false,
                "verify": true,
                "deriveKey": false,
                "noRestrictions": false
            }
        },
        "exportable": true
    },
    "responseElements": {
        "key": {
            "keyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h",
            "keyAttributes": {
                "keyUsage": "TR31_M1_ISO_9797_1_MAC_KEY",
                "keyClass": "SYMMETRIC_KEY",
                "keyAlgorithm": "TDES_2KEY",
                "keyModesOfUse": {
                    "encrypt": false,
```

```
                    "decrypt": false,
                    "wrap": false,
                    "unwrap": false,
                    "generate": true,
                    "sign": false,
                    "verify": true,
                    "deriveKey": false,
                    "noRestrictions": false
                }
            },
            "keyCheckValue": "A486ED",
            "keyCheckValueAlgorithm": "ANSI_X9_24",
            "enabled": true,
            "exportable": true,
            "keyState": "CREATE_COMPLETE",
            "keyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
            "createTimestamp": "May 27, 2024, 7:49:54 PM",
            "usageStartTimestamp": "May 27, 2024, 7:49:54 PM"
        }
    },
    "requestID": "f3020b3c-4e86-47f5-808f-14c7a4a99161",
    "eventID": "b87c3d30-f3ab-4131-87e8-bc54cfef9d29",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "vpcEndpointId": "",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "-oo28vrvr.controlplane.payment-cryptography.us-
east-1.vpce.amazonaws.com"
    }
}
```


Praktik terbaik keamanan untuk Kriptografi AWS Pembayaran

AWS Kriptografi Pembayaran mendukung banyak fitur keamanan yang built-in atau yang dapat Anda terapkan secara opsional untuk meningkatkan perlindungan kunci enkripsi Anda dan memastikan bahwa mereka digunakan untuk tujuan yang dimaksudkan, termasuk kebijakan IAM, serangkaian

kunci kondisi kebijakan yang ekstensif untuk menyempurnakan kebijakan utama Anda dan kebijakan IAM dan penegakan aturan PIN PCI bawaan mengenai blok kunci.

> **Important**
>
> Pedoman umum yang diberikan tidak mewakili solusi keamanan yang lengkap. Karena tidak semua praktik terbaik sesuai untuk semua situasi, ini tidak dimaksudkan untuk menjadi preskriptif.

- Penggunaan Utama dan Mode Penggunaan: Kriptografi AWS Pembayaran mengikuti dan memberlakukan pembatasan penggunaan utama dan mode penggunaan seperti yang dijelaskan dalam ANSI X9 TR 31-2018 Spesifikasi Blok Kunci Pertukaran Kunci Aman yang Dapat Dioperasikan dan konsisten dengan Persyaratan Keamanan PIN PCI 18-3. Ini membatasi kemampuan untuk menggunakan satu kunci untuk berbagai tujuan dan secara kriptografis mengikat metadata kunci (seperti operasi yang diizinkan) ke materi kunci itu sendiri. AWS Kriptografi Pembayaran secara otomatis memberlakukan pembatasan ini seperti kunci enkripsi kunci (TR31_K0_KEY_ENCRYPTION_KEY) juga tidak dapat digunakan untuk dekripsi data. Lihat Memahami atribut kunci untuk kunci Kriptografi AWS Pembayaran untuk detail selengkapnya.
- Batasi pembagian materi kunci simetris: Hanya bagikan materi kunci simetris (seperti Kunci Enkripsi Pin atau Kunci Enkripsi Kunci) dengan paling banyak satu entitas lainnya. Jika ada kebutuhan untuk mentransmisikan materi sensitif ke lebih banyak entitas atau mitra, buat kunci tambahan. AWS Kriptografi Pembayaran tidak pernah mengekspos materi kunci simetris atau materi kunci pribadi asimetris secara jelas.
- Gunakan alias atau tag untuk mengaitkan kunci dengan kasus penggunaan atau mitra tertentu: Alias dapat digunakan untuk dengan mudah menunjukkan kasus penggunaan yang terkait dengan kunci seperti alias/BIN_12345_CVK untuk menunjukkan kunci verifikasi kartu yang terkait dengan BIN 12345. Untuk memberikan lebih banyak fleksibilitas, pertimbangkan untuk membuat tag seperti bin = 12345, use_case=acquiring, country=us, partner=foo. Alias dan tag juga dapat digunakan untuk membatasi akses seperti menegakkan kontrol akses antara mengeluarkan dan memperoleh kasus penggunaan.
- Praktekkan akses yang paling tidak istimewa: IAM dapat digunakan untuk membatasi akses produksi ke sistem daripada individu, seperti melarang pengguna individu membuat kunci atau menjalankan operasi kriptografi. IAM juga dapat digunakan untuk membatasi akses ke perintah dan kunci yang mungkin tidak berlaku untuk kasus penggunaan Anda, seperti membatasi kemampuan untuk menghasilkan atau memvalidasi pin untuk pengakuisisi. Cara lain untuk menggunakan

akses yang paling tidak memiliki hak istimewa adalah dengan membatasi operasi sensitif (seperti impor kunci) ke akun layanan tertentu. Lihat AWS Contoh kebijakan berbasis identitas Kriptografi Pembayaran sebagai contoh.

Lihat juga

- Manajemen identitas dan akses untuk Kriptografi AWS Pembayaran
- Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM

Validasi kepatuhan untukAWSKriptografi Pembayaran

Auditor pihak ketiga menilai keamanan dan kepatuhanAWSKriptografi Pembayaran sebagai bagian dari beberapaAWSprogram kepatuhan. Ini termasuk SOC, PCI, dan lain-lain.

AWSKriptografi Pembayaran telah dinilai untuk beberapa standar PCI selain PCI DSS. Ini termasuk PCI PIN Security (PCI PIN) dan PCI Point-to-Point (P2PE) Enkripsi. Silakan lihatAWS Artifactuntuk pengesahan yang tersedia dan panduan kepatuhan.

Untuk daftar layanan AWS dalam cakupan program kepatuhan tertentu, lihat Layanan AWS dalam Cakupan Program Kepatuhan. Untuk informasi umum, lihat Program Kepatuhan AWS.

Anda bisa mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact.

Tanggung jawab kepatuhan Anda saat menggunakanAWSKriptografi Pembayaran ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku.AWSmenyediakan sumber daya berikut untuk membantu dengan kepatuhan:

- Panduan Memulai Cepat Keamanan dan Kepatuhan—Panduan penyebaran ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar yang berfokus pada keamanan dan kepatuhanAWS.
- AWSSumber Daya Kepatuhan—Koleksi buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- Mengevaluasi Sumber Daya dengan Aturandi dalamAWS ConfigPanduan Pengembang—AWS Config; menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub—IniAWSlayanan memberikan pandangan komprehensif tentang status keamanan Anda di dalamnyaAWSyang membantu Anda memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik.

Manajemen identitas dan akses untuk Kriptografi AWS Pembayaran

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan AWS sumber daya Kriptografi Pembayaran. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan dalam Kriptografi AWS Pembayaran.

Pengguna layanan — Jika Anda menggunakan layanan Kriptografi AWS Pembayaran untuk melakukan pekerjaan Anda, maka administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak fitur Kriptografi AWS Pembayaran untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur dalam Kriptografi AWS Pembayaran, lihatPemecahan Masalah Identitas dan AWS akses Kriptografi Pembayaran.

Administrator layanan — Jika Anda bertanggung jawab atas sumber daya Kriptografi AWS Pembayaran di perusahaan Anda, Anda mungkin memiliki akses penuh ke Kriptografi AWS Pembayaran. Tugas Anda adalah menentukan fitur dan sumber daya Kriptografi AWS Pembayaran mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM untuk mengubah izin pengguna layanan Anda. Tinjau informasi di

halaman ini untuk memahami konsep Basic IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM dengan Kriptografi AWS Pembayaran, lihat. Bagaimana Kriptografi AWS Pembayaran bekerja dengan IAM

Administrator IAM — Jika Anda seorang administrator IAM, Anda mungkin ingin mempelajari detail tentang bagaimana Anda dapat menulis kebijakan untuk mengelola akses ke Kriptografi AWS Pembayaran. Untuk melihat contoh kebijakan berbasis identitas Kriptografi AWS Pembayaran yang dapat Anda gunakan di IAM, lihat. AWS Contoh kebijakan berbasis identitas Kriptografi Pembayaran

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensyal identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensyal yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas terfederasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS.

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensil Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat Menandatangani permintaan AWS API di Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, lihat Autentikasi multi-faktor dalam Panduan Pengguna AWS IAM Identity Center dan Menggunakan autentikasi multi-faktor (MFA) dalam AWS dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Pengguna dan grup IAM

Pengguna IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, kami merekomendasikan untuk mengandalkan kredensial sementara, bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan tertentu yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami merekomendasikan Anda merotasi kunci akses. Untuk informasi selengkapnya, lihat Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang dalam Panduan Pengguna IAM.

Grup IAM adalah identitas yang menentukan sekumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat memiliki grup yang bernama IAMAdmins dan memberikan izin ke grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, lihat Kapan harus membuat pengguna IAM (bukan peran) dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Anda dapat mengambil peran IAM untuk sementara AWS Management Console dengan beralih peran. Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk

informasi selengkapnya tentang cara menggunakan peran, lihat Menggunakan peran IAM dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat Membuat peran untuk Penyedia Identitas pihak ketiga dalam Panduan Pengguna IAM. Jika menggunakan Pusat Identitas IAM, Anda harus mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM akan mengorelasikan set izin ke peran dalam IAM. Untuk informasi tentang set izin, lihat Set izin dalam Panduan Pengguna AWS IAM Identity Center .
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (prinsipal tepercaya) di akun lain untuk mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya dalam Panduan Pengguna IAM.
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Sebagai contoh, ketika Anda memanggil suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.
  - Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses maju.

  - Peran layanan – Peran layanan adalah peran IAM yang dijalankan oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Membuat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.
  - Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensi sementara untuk aplikasi yang berjalan pada instans EC2 dan membuat atau permintaan API. AWS CLI AWS Cara ini lebih dianjurkan daripada menyimpan kunci akses dalam instans EC2. Untuk menetapkan AWS peran ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan dalam instans Amazon EC2 dalam Panduan Pengguna IAM.

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, lihat Kapan harus membuat peran IAM (bukan pengguna) dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat

kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasinya. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam. Akun AWS Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan yang dikelola atau kebijakan inline, lihat Memilih antara kebijakan yang dikelola dan kebijakan inline dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACL)

Daftar kontrol akses (ACL) mengendalikan prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, lihat Gambaran umum daftar kontrol akses (ACL) dalam Panduan Developer Amazon Simple Storage Service.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas ke entitas IAM (pengguna IAM atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batasan izin, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) — SCP adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di. AWS Organizations AWS Organizations adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur di organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS Untuk informasi selengkapnya tentang Organisasi dan SCP, lihat Cara kerja SCP dalam Panduan Pengguna AWS Organizations .
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Bagaimana Kriptografi AWS Pembayaran bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke Kriptografi AWS Pembayaran, Anda harus memahami fitur IAM apa yang tersedia untuk digunakan dengan AWS Kriptografi Pembayaran. Untuk mendapatkan pandangan tingkat tinggi tentang bagaimana Kriptografi AWS Pembayaran dan AWS layanan lainnya bekerja dengan IAM, lihat AWS Layanan yang Bekerja dengan IAM di Panduan Pengguna IAM.

Topik

- AWS Kebijakan berbasis identitas Kriptografi Pembayaran
- Otorisasi berdasarkan tag Kriptografi AWS Pembayaran

AWS Kebijakan berbasis identitas Kriptografi Pembayaran

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. AWS Kriptografi Pembayaran mendukung tindakan, sumber daya, dan kunci kondisi tertentu. Untuk mempelajari semua elemen yang Anda gunakan dalam kebijakan JSON, lihat Referensi Elemen Kebijakan JSON IAM dalam Panduan Pengguna IAM.

Tindakan

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Menyertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Tindakan kebijakan dalam Kriptografi AWS Pembayaran menggunakan awalan berikut sebelum tindakan:. `payment-cryptography:` Misalnya, untuk memberikan izin kepada seseorang untuk menjalankan operasi `VerifyCardData` API Kriptografi AWS Pembayaran, Anda menyertakan `payment-cryptography:VerifyCardData` tindakan tersebut dalam kebijakan mereka. Pernyataan kebijakan harus memuat elemen `Action` atau `NotAction`. AWS Kriptografi Pembayaran mendefinisikan serangkaian tindakannya sendiri yang menggambarkan tugas yang dapat Anda lakukan dengan layanan ini.

Untuk menetapkan beberapa tindakan dalam satu pernyataan, pisahkan dengan koma seperti berikut:

```
"Action": [
      "payment-cryptography:action1",
      "payment-cryptography:action2"
```

Anda dapat menentukan beberapa tindakan menggunakan wildcard (*). Misalnya, untuk menentukan semua tindakan yang dimulai dengan kata `List` (seperti `ListKeys` dan`ListAliases`), sertakan tindakan berikut:

```
"Action": "payment-cryptography:List*"
```

Untuk melihat daftar tindakan Kriptografi AWS Pembayaran, lihat Tindakan yang Ditentukan oleh Kriptografi AWS Pembayaran di Panduan Pengguna IAM.

Sumber daya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Sumber daya kunci kriptografi pembayaran memiliki ARN berikut:

```
arn:${Partition}:payment-cryptography:${Region}:${Account}:key/${keyARN}
```

Untuk informasi selengkapnya tentang format ARN, lihat Nama Sumber Daya Amazon (ARN) dan Ruang Nama AWS Layanan.

Misalnya, untuk menentukan instans `arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h` dalam pernyataan Anda, gunakan ARN berikut:

```
"Resource": "arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h"
```

Untuk menentukan semua kunci milik akun tertentu, gunakan wildcard (*):

```
"Resource": "arn:aws:payment-cryptography:us-east-2:111122223333:key/*"
```

Beberapa tindakan Kriptografi AWS Pembayaran, seperti untuk membuat kunci, tidak dapat dilakukan pada sumber daya tertentu. Dalam kasus tersebut, Anda harus menggunakan wildcard (*).

```
"Resource": "*"
```

Untuk menentukan beberapa sumber daya dalam satu pernyataan, gunakan koma seperti yang ditunjukkan di bawah ini:

```
"Resource": [
      "resource1",
      "resource2"
```

Contoh

Untuk melihat contoh kebijakan berbasis identitas Kriptografi AWS Pembayaran, lihat. AWS Contoh kebijakan berbasis identitas Kriptografi Pembayaran

Otorisasi berdasarkan tag Kriptografi AWS Pembayaran

AWS Contoh kebijakan berbasis identitas Kriptografi Pembayaran

Secara default, pengguna dan peran IAM tidak memiliki izin untuk membuat atau memodifikasi sumber daya Kriptografi AWS Pembayaran. Mereka juga tidak dapat melakukan tugas menggunakan AWS Management Console, AWS CLI, atau AWS API. Administrator IAM harus membuat kebijakan IAM yang memberikan izin kepada pengguna dan peran untuk melakukan operasi API tertentu pada sumber daya yang diperlukan. Administrator kemudian harus melampirkan kebijakan tersebut ke pengguna IAM atau grup yang memerlukan izin tersebut.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat Kebijakan pada Tab JSON dalam Panduan Pengguna IAM.

Topik

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Kriptografi AWS Pembayaran di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi

selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.

- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi akses API yang dilindungi MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan dalam IAM dalam Panduan Pengguna IAM.

Menggunakan konsol Kriptografi AWS Pembayaran

Untuk mengakses konsol Kriptografi AWS Pembayaran, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya Kriptografi AWS Pembayaran di akun Anda AWS . Jika Anda membuat kebijakan berbasis identitas

yang lebih ketat daripada izin minimum yang diperlukan, konsol tersebut tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna IAM atau peran) dengan kebijakan tersebut.

Untuk memastikan bahwa entitas tersebut masih dapat menggunakan konsol Kriptografi AWS Pembayaran, lampirkan juga kebijakan AWS terkelola berikut ke entitas. Untuk informasi selengkapnya, lihat Menambahkan Izin ke Pengguna dalam Panduan Pengguna IAM.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai alternatif, hanya izinkan akses ke tindakan yang cocok dengan operasi API yang sedang Anda coba lakukan.

Izinkan para pengguna untuk melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
```

```
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Kemampuan untuk mengakses semua aspek Kriptografi AWS Pembayaran

> **Warning**
>
> Contoh ini memberikan izin luas dan tidak disarankan. Pertimbangkan model akses yang paling tidak priviledged sebagai gantinya.

Dalam contoh ini, Anda ingin memberikan pengguna IAM di AWS akun Anda akses ke semua kunci Kriptografi AWS Pembayaran Anda dan kemampuan untuk memanggil semua API Kriptografi AWS Pembayaran termasuk keduanya ControlPlane dan operasi. DataPlane

```
{
   "Version": "2012-10-17",
   "Statement": [
         {
            "Effect": "Allow",
            "Action": [
               "payment-cryptography:*"
            ],
            "Resource": [
               "*"
            ]
         }
   ]
}
```

Kemampuan untuk memanggil API menggunakan kunci tertentu

Dalam contoh ini, Anda ingin memberikan pengguna IAM di AWS akun Anda akses ke salah satu kunci Kriptografi AWS Pembayaran Anda, `arn:aws:payment-cryptography:us-`

east-2:111122223333:key/kwapwa6qaifllw2h dan kemudian menggunakan sumber daya ini dalam dua API, GenerateCardData dan. VerifyCardData Sebaliknya, pengguna IAM tidak akan memiliki akses untuk menggunakan kunci ini pada operasi lain seperti atau DeleteKey ExportKey

Sumber daya dapat berupa kunci, diawali dengan key atau alias, diawali dengan. alias

```
   {
      "Version": "2012-10-17",
      "Statement": [
            {
               "Effect": "Allow",
               "Action": [
                  "payment-cryptography:VerifyCardData",
                  "payment-cryptography:GenerateCardData"
               ],
               "Resource": [
                  "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h"
               ]
            }
      ]
   }
```

Kemampuan untuk secara khusus menolak sumber daya

> **Warning**
>
> Pertimbangkan dengan cermat implikasi pemberian akses wildcard. Pertimbangkan model hak istimewa yang paling tidak.

Dalam contoh ini, Anda ingin mengizinkan pengguna IAM di AWS akun Anda mengakses salah satu kunci Kriptografi AWS Pembayaran Anda tetapi ingin menolak izin ke satu kunci tertentu. Pengguna akan memiliki akses ke VerifyCardData dan GenerateCardData dengan semua kunci dengan pengecualian yang ditentukan dalam pernyataan penolakan.

```
   {
      "Version": "2012-10-17",
      "Statement": [
```

```
            {
                "Effect": "Allow",
                "Action": [
                    "payment-cryptography:VerifyCardData",
                    "payment-cryptography:GenerateCardData"
                ],
                "Resource": [
                    "arn:aws:payment-cryptography:us-east-2:111122223333:key/*"
                ]
            },
            {
                "Effect": "Deny",
                "Action": [
                    "payment-cryptography:GenerateCardData"
                ],
                "Resource": [
                    "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h"
                ]
            }
        ]
    }
```

Pemecahan Masalah Identitas dan AWS akses Kriptografi Pembayaran

Topik akan ditambahkan ke bagian ini karena masalah terkait IAM yang khusus untuk Kriptografi AWS Pembayaran diidentifikasi. Untuk konten pemecahan masalah umum tentang topik IAM, lihat bagian pemecahan masalah pada Panduan Pengguna IAM.

Pemantauan Kriptografi AWS Pembayaran

Pemantauan adalah bagian penting dari pemeliharaan, ketersediaan, dan kinerja Kriptografi AWS Pembayaran dan solusi AWS lainnya. AWS menyediakan alat pemantauan berikut untuk mengawasi Kriptografi AWS Pembayaran, melaporkan saat terjadi kesalahan, dan mengambil tindakan otomatis jika diperlukan:

- Amazon CloudWatch memantau AWS sumber daya Anda dan aplikasi yang Anda jalankan AWS di secara langsung. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor yang disesuaikan, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik tertentu mencapai ambang batas yang ditentukan. Misalnya, Anda dapat membuat CloudWatch melacak penggunaan CPU atau metrik lain dari instans Amazon EC2 Anda dan secara otomatis meluncurkan instans baru ketika diperlukan. Untuk informasi lebih lanjut, lihat Panduan Pengguna Amazon CloudWatch.
- Amazon CloudWatch Logs membantu Anda memantau, menyimpan, dan mengakses berkas log dari instans Amazon EC2,,CloudTrail, dan sumber lainnya. Log CloudWatch dapat memantau informasi dalam file log dan memberi tahu Anda ketika ambang tertentu terpenuhi. Anda juga dapat mengarsipkan data log Anda dalam penyimpanan yang sangat tahan lama. Untuk informasi selengkapnya, lihat Panduan Pengguna Log CloudWatch Amazon.
- Amazon EventBridge dapat digunakan untuk mengotomatiskan AWS layanan Anda dan merespons peristiwa sistem secara otomatis seperti masalah ketersediaan aplikasi atau perubahan sumber daya. Peristiwa dari layanan AWS dikirimkan ke EventBridge dengan mendekati waktu nyata. Anda dapat menuliskan aturan sederhana untuk menunjukkan peristiwa mana yang sesuai kepentingan Anda, dan tindakan otomatis mana yang diambil ketika suatu peristiwa sesuai dengan suatu aturan. Untuk informasi selengkapnya, lihat Panduan EventBridge Pengguna Amazon.
- AWS CloudTrail merekam panggilan API dan peristiwa terkait yang dilakukan oleh atau atas nama akun AWS Anda dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang memanggil AWS, alamat IP sumber yang melakukan panggilan, dan kapan panggilan tersebut terjadi. Untuk mengetahui informasi selengkapnya, lihat Panduan Pengguna AWS CloudTrail.

> **Note**
>
> AWS CloudTraillog didukung untuk operasi Control Plane seperti CreateKey tetapi tidak untuk operasi Data Plane seperti Menghasilkan Data Kartu

Logging AWS Pembayaran Kriptografi API panggilan menggunakan AWS CloudTrail

AWSPembayaran Kriptografi terintegrasi denganAWS CloudTrail, sebuah layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di Kriptografi AWS Pembayaran. CloudTrailmerekam semua panggilan API untuk Kriptografi AWS Pembayaran sebagai peristiwa. Panggilan yang direkam mencakup panggilan dari konsol AWS Payment Cryptography dan panggilan kode ke operasi API AWS Payment Cryptography. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman berkelanjutan dari CloudTrail peristiwa S3 ke bucket Amazon S3, termasuk peristiwa untuk Kriptografi AWS Pembayaran. Jika Anda tidak membuat konfigurasi jejak, Anda masih dapat melihat kejadian terbaru dalam konsol CloudTrail di Riwayat peristiwa. Menggunakan informasi yang dikumpulkan olehCloudTrail, Anda dapat menentukan permintaan yang dibuat ke Kriptografi AWS Pembayaran, alamat IP asal permintaan tersebut dibuat, siapa yang membuat permintaan, kapan permintaan dibuat, dan detail lainnya.

Untuk mempelajari lebih lanjutCloudTrail, lihat Panduan AWS CloudTrail Pengguna.

> **Note**
>
> Integrasi Cloudtrail saat ini didukung untuk operasi bidang kontrol saja.

AWSInformasi Kriptografi Pembayaran di CloudTrail

CloudTrail diaktifkan pada akun AWS Anda saat Anda membuat akun tersebut. Ketika aktivitas terjadi di Kriptografi AWS Pembayaran, aktivitas tersebut dicatat dalam CloudTrail peristiwa AWS layanan lainnya di Riwayat peristiwa. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di akun AWS Anda. Untuk informasi lain, lihat Melihat Peristiwa dengan Riwayat Peristiwa CloudTrail.

Untuk catatan berkelanjutan tentang peristiwa di AWS akun Anda, termasuk peristiwa untuk Kriptografi AWS Pembayaran, buat jejak. Jejak memungkinkan CloudTrail untuk mengirim berkas log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di dalam konsol tersebut, jejak diterapkan ke semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di partisi AWS dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat membuat konfigurasi layanan AWS lainnya untuk menganalisis lebih lanjut dan bertindak berdasarkan data peristiwa yang dikumpulkan di log CloudTrail. Untuk informasi selengkapnya, lihat yang berikut:

- Gambaran umum untuk membuat jejak

- CloudTrailLayanan yang didukung dan integrasi
- Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail
- Menerima ima ima ima ima ima berkas CloudTrail log dari beberapa Wilayah
- Menerima ima ima ima ima berkas CloudTrail log dari beberapa akun

CloudTraillog operasi Kriptografi AWS Pembayaran, seperti CreateKey, ImportKey, DeleteKey, ListKeys, TagResource, dan semua operasi pesawat kontrol lainnya.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut:

- Bahwa permintaan dibuat dengan kredensial pengguna root atau pengguna AWS Identity and Access Management (IAM).
- Bahwa permintaan tersebut dibuat dengan kredensial keamanan sementara untuk peran atau pengguna gabungan.
- Bahwa permintaan dibuat oleh layanan AWS lain.

Untuk informasi selengkapnya, lihat Elemen userIdentity CloudTrail.

Memahami AWS entri berkas log Kriptografi Pembayaran

Jejak adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai berkas log ke bucket Amazon S3 yang telah Anda tentukan. Berkas log CloudTrail berisi satu atau beberapa entri log. Peristuwa mewakili satu permintaan dari sumber apa pun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. Berkas log CloudTrail bukan jejak tumpukan terurut dari panggilan API publik, sehingga berkas tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan tindakan AWS pembayaran Kriptografi`CreateKey`.

```
  {
CloudTrailEvent: {
  tlsDetails= {
    TlsDetails: {
      cipherSuite=TLS_AES_128_GCM_SHA256,
      tlsVersion=TLSv1.3,
```

```
          clientProvidedHostHeader=pdx80.controlplane.paymentcryptography.us-
west-2.amazonaws.com
        }
      },
    requestParameters=CreateKeyInput (
      keyAttributes=KeyAttributes(
        KeyUsage=TR31_B0_BASE_DERIVATION_KEY,
        keyClass=SYMMETRIC_KEY,
        keyAlgorithm=AES_128,
        keyModesOfUse=KeyModesOfUse(
          encrypt=false,
          decrypt=false,
          wrap=false
          unwrap=false,
          generate=false,
          sign=false,
          verify=false,
          deriveKey=true,
          noRestrictions=false)
        ),
      keyCheckValueAlgorithm=null,
      exportable=true,
      enabled=true,
      tags=null),
    eventName=CreateKey,
    userAgent=Coral/Apache-HttpClient5,
    responseElements=CreateKeyOutput(
      key=Key(
        keyArn=arn:aws:payment-cryptography:us-
east-2:111122223333:key/5rplquuwozodpwsp,
          keyAttributes=KeyAttributes(
            KeyUsage=TR31_B0_BASE_DERIVATION_KEY,
            keyClass=SYMMETRIC_KEY,
            keyAlgorithm=AES_128,
            keyModesOfUse=KeyModesOfUse(
              encrypt=false,
              decrypt=false,
              wrap=false,
              unwrap=false,
              generate=false,
              sign=false,
              verify=false,
              deriveKey=true,
              noRestrictions=false)
```

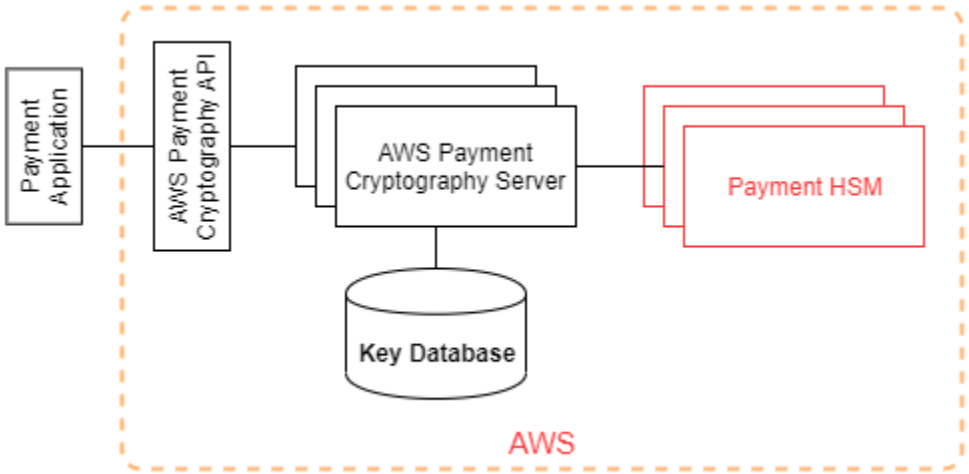
```
            ),
          keyCheckValue=FE23D3,
          keyCheckValueAlgorithm=ANSI_X9_24,
          enabled=true,
          exportable=true,
          keyState=CREATE_COMPLETE,
          keyOrigin=AWS_PAYMENT_CRYPTOGRAPHY,
          createTimestamp=Sun May 21 18:58:32 UTC 2023,
          usageStartTimestamp=Sun May 21 18:58:32 UTC 2023,
          usageStopTimestamp=null,
          deletePendingTimestamp=null,
          deleteTimestamp=null)
        ),
      sourceIPAddress=192.158.1.38,
        userIdentity={
          UserIdentity: {
            arn=arn:aws:sts::111122223333:assumed-role/TestAssumeRole-us-west-2-PDX80/
ControlPlane-IntegTest-68211a2a-3e9d-42b7-86ac-c682520e0410,
            invokedBy=null,
            accessKeyId=,
            type=AssumedRole,
            sessionContext={
              SessionContext: {
                sessionIssuer={
                  SessionIssuer: {arn=arn:aws:iam::111122223333:role/TestAssumeRole-us-
west-2-PDX80,
                  type=Role,
                  accountId=111122223333,
                  userName=TestAssumeRole-us-west-2-PDX80,
                  principalId=}
                },
                attributes={
                  SessionContextAttributes: {
                    creationDate=Sun May 21 18:58:31 UTC 2023,
                    mfaAuthenticated=false
                  }
                },
                webIdFederationData=null
              }
            },
          username=null,
          principalId=:ControlPlane-User,
          accountId=111122223333,
          identityProvider=null
```

```
        }
      },
      eventTime=Sun May 21 18:58:32 UTC 2023,
      managementEvent=true,
      recipientAccountId=111122223333,
      awsRegion=us-west-2,
      requestID=151cdd67-4321-1234-9999-dce10d45c92e,
      eventVersion=1.08, eventType=AwsApiCall,
      readOnly=false,
      eventID=c69e3101-eac2-1b4d-b942-019919ad2faf,
      eventSource=payment-cryptography.amazonaws.com,
      eventCategory=Management,
      additionalEventData={
    }
  }
}
```

Detail kriptografi

AWSKriptografi Pembayaran menyediakan antarmuka web untuk menghasilkan dan mengelola kunci kriptografi untuk transaksi pembayaran. AWS Kriptografi Pembayaran menawarkan layanan manajemen kunci standar dan kriptografi transaksi pembayaran dan alat yang dapat Anda gunakan untuk manajemen dan audit terpusat. Laporan resmi memberikan penjelasan mendetail tentang operasi kriptografi yang dapat Anda gunakan dalam kriptografi AWS pembayaran untuk membantu Anda dalam mengevaluasi fitur yang ditawarkan oleh layanan.

AWSKriptografi Pembayaran berisi beberapa antarmuka (termasuk RESTful API, melalui AWS CLI, AWS SDK, danAWS Management Console) untuk meminta operasi kriptografi dari armada terdistribusi modul keamanan perangkat keras yang divalidasi PCI PTS HSM.



AWSPembayaran kriptografi adalah layanan bertingkat yang terdiri atas host web AWS dan tingkat HSM. Pengelompokan host bertingkat ini membentuk tumpukan AWS Pembayaran. Semua permintaan ke kriptografi AWS pembayaran harus dilakukan melalui protokol Transport Layer Security (TLS) dan diakhiri pada host AWS Pembayaran. Host layanan hanya mengizinkan TLS dengan cipher suite yang memberikan kerahasiaan ke depan yang sempurna. Layanan mengautentikasi dan mengotorisasi permintaan Anda menggunakan kredensi dan mekanisme kebijakan IAM yang sama yang tersedia untuk semua operasi API lainnya. AWS

AWSServer Kriptografi Pembayaran terhubung ke HSM yang mendasarinya melalui jaringan pribadi dan non-virtual. Koneksi antara komponen layanan dan HSM diamankan dengan TLS bersama (MTL) untuk otentikasi dan enkripsi.

Tujuan desain

AWSKtografi Pembayaran dirancang untuk memenuhi persyaratan berikut:

- Dapat dipercaya - Penggunaan kunci dilindungi oleh kebijakan kontrol akses yang Anda tetapkan dan kelola. Tidak ada mekanisme untuk mengekspor kunci kriptografi AWS pembayaran 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. Tidak ada karyawan Amazon yang memiliki akses ke kunci atau cadangan utama (atau master) HSM. Kunci utama tidak dapat disinkronkan dengan HSM yang bukan bagian dari wilayah Kriptografi AWS Pembayaran. Semua kunci lainnya dilindungi oleh kunci utama HSM. Oleh karena itu, kunci Kriptografi AWS Pembayaran pelanggan tidak dapat digunakan di luar layanan Kriptografi AWS Pembayaran yang beroperasi di dalam akun pelanggan.
- Latensi rendah dan throughput tinggi - Kriptografi AWS Pembayaran menyediakan operasi kriptografi pada tingkat latensi dan throughput yang sesuai untuk mengelola kunci kriptografi pembayaran dan memproses transaksi pembayaran.
- Daya tahan — Daya tahan kunci kriptografi dirancang untuk menyamai ketahanan layanan daya tahan tertinggi di AWS. Kunci kriptografi tunggal dapat dibagikan dengan terminal pembayaran, kartu chip EMV, atau perangkat kriptografi aman lainnya (SCD) yang digunakan selama bertahun-tahun.
- Wilayah Independen — AWS menyediakan wilayah independen bagi pelanggan yang perlu membatasi akses data di berbagai wilayah atau harus mematuhi persyaratan residensi data. Penggunaan kunci dapat diisolasi dalam Wilayah AWS.
- Sumber nomor acak yang aman - Karena kriptografi yang kuat tergantung pada generasi nomor acak yang benar-benar tak terduga, Kriptografi AWS Pembayaran menyediakan sumber nomor acak yang berkualitas tinggi dan divalidasi. Semua generasi kunci untuk Kriptografi AWS Pembayaran menggunakan HSM terdaftar PCI PTS HSM, beroperasi dalam mode PCI.
- Audit — Kriptografi AWS Pembayaran mencatat penggunaan dan pengelolaan kunci kriptografi dalam CloudTrail log dan log layanan yang tersedia melalui Amazon. CloudWatch Anda dapat menggunakan CloudTrail log untuk memeriksa penggunaan kunci kriptografi Anda, termasuk penggunaan kunci oleh akun yang Anda gunakan kunci bersama. AWS Kriptografi Pembayaran diaudit oleh asesor pihak ketiga terhadap standar keamanan pembayaran PCI, merek kartu, dan regional yang berlaku. Panduan Pengesahan dan Tanggung Jawab Bersama tersedia di AWS Artifact.
- Elastis - AWS Pembayaran Kriptografi skala keluar dan sesuai dengan permintaan Anda. Alih-alih memprediksi dan memesan kapasitas HSM, AWS Payment Cryptography menyediakan kriptografi

pembayaran sesuai permintaan. AWS Kriptografi Pembayaran bertanggung jawab untuk menjaga keamanan dan kepatuhan HSM untuk menyediakan kapasitas yang cukup untuk memenuhi permintaan puncak pelanggan.

Yayasan

Topik dalam Bab ini menjelaskan primitif kriptografi Kriptografi AWS Pembayaran dan di mana mereka digunakan. Mereka juga memperkenalkan elemen dasar layanan.

Topik

Primitif kriptografi

AWS Kriptografi Pembayaran menggunakan algoritma kriptografi standar yang dapat parameter sehingga aplikasi dapat mengimplementasikan algoritma yang diperlukan untuk kasus penggunaannya. Himpunan algoritma kriptografi ditentukan oleh standar PCI, ANSI X9, EMVCo, dan ISO. Semua kriptografi dilakukan oleh HSM terdaftar standar PCI PTS HSM yang berjalan dalam mode PCI.

Entropi dan pembangkitan bilangan acak

AWS Pembangkitan kunci Kriptografi Pembayaran dilakukan pada HSM Kriptografi AWS Pembayaran. HSM menerapkan generator angka acak yang memenuhi persyaratan PCI PTS HSM untuk semua jenis dan parameter kunci yang didukung.

Operasi kunci simetris

Algoritma kunci simetris dan kekuatan kunci yang ditentukan dalam ANSI X9 TR 31, ANSI X9.24, dan PCI PIN Annex C didukung:

- Fungsi hash — Algoritma dari keluarga SHA2 dan SHA3 dengan ukuran output lebih besar dari 2551. Kecuali untuk kompatibilitas mundur dengan terminal pra-PCI PTS POI v3.
- Enkripsi dan dekripsi — AES dengan ukuran kunci lebih besar dari atau sama dengan 128 bit, atau TDEA dengan ukuran kunci lebih besar dari atau sama dengan 112 bit (2 kunci atau 3 kunci).
- Kode Otentikasi Pesan (MAC) CMAC atau GMAC dengan AES, serta HMAC dengan fungsi hash yang disetujui dan ukuran kunci lebih besar dari atau sama dengan 128.

AWS Kriptografi Pembayaran menggunakan AES 256 untuk kunci utama HSM, kunci perlindungan data, dan kunci sesi TLS.

Operasi kunci asimetris

Algoritma kunci asimetris dan kekuatan kunci yang ditentukan dalam ANSI X9 TR 31, ANSI X9.24, dan PCI PIN Annex C didukung:

- Skema pembentukan kunci yang disetujui - seperti yang dijelaskan dalam NIST SP800-56A (perjanjian kunci berbasis ECC/FCC2), NIST SP800-56B (perjanjian kunci berbasis IFC), dan NIST SP800-38F (enkripsi/pembungkus kunci berbasis AES).

AWS Host Payment Cryptography hanya mengizinkan koneksi ke layanan menggunakan TLS dengan cipher suite yang memberikan kerahasiaan forward yang sempurna.

Penyimpanan kunci

AWS Kunci Kriptografi Pembayaran dilindungi oleh kunci utama HSM AES 256 dan disimpan dalam blok kunci ANSI X9 TR 31 dalam database terenkripsi. Basis data direplikasi ke database dalam memori pada server Kriptografi AWS Pembayaran.

Menurut PCI PIN Security Normative Annex C, kunci AES 256 sama kuatnya dengan atau lebih kuat dari:

- TDEA 3-kunci

- RSA 15360 bit
- ECC 512 bit
- DSA, DH, dan MQV 15360/512

Impor kunci menggunakan tombol simetris

AWS Kriptografi Pembayaran mendukung impor kriptogram dan blok kunci dengan kunci simetris atau publik dengan kunci enkripsi kunci simetris (KEK) yang kuat atau lebih kuat dari kunci yang dilindungi untuk impor.

Impor kunci menggunakan tombol asimetris

AWS Kriptografi Pembayaran mendukung impor kriptogram dan blok kunci dengan kunci simetris atau publik yang dilindungi oleh kunci enkripsi kunci pribadi (KEK) yang sekuat atau lebih kuat dari kunci yang dilindungi untuk impor. Kunci publik yang disediakan untuk dekripsi harus memiliki keaslian dan integritasnya yang dijamin oleh sertifikat dari otoritas yang dipercaya oleh pelanggan.

KEK Publik yang disediakan oleh Kriptografi AWS Pembayaran memiliki otentikasi dan perlindungan integritas otoritas sertifikat (CA) dengan kepatuhan yang terbukti terhadap Keamanan PIN PCI dan Lampiran PCI P2PE A.

Ekspor kunci

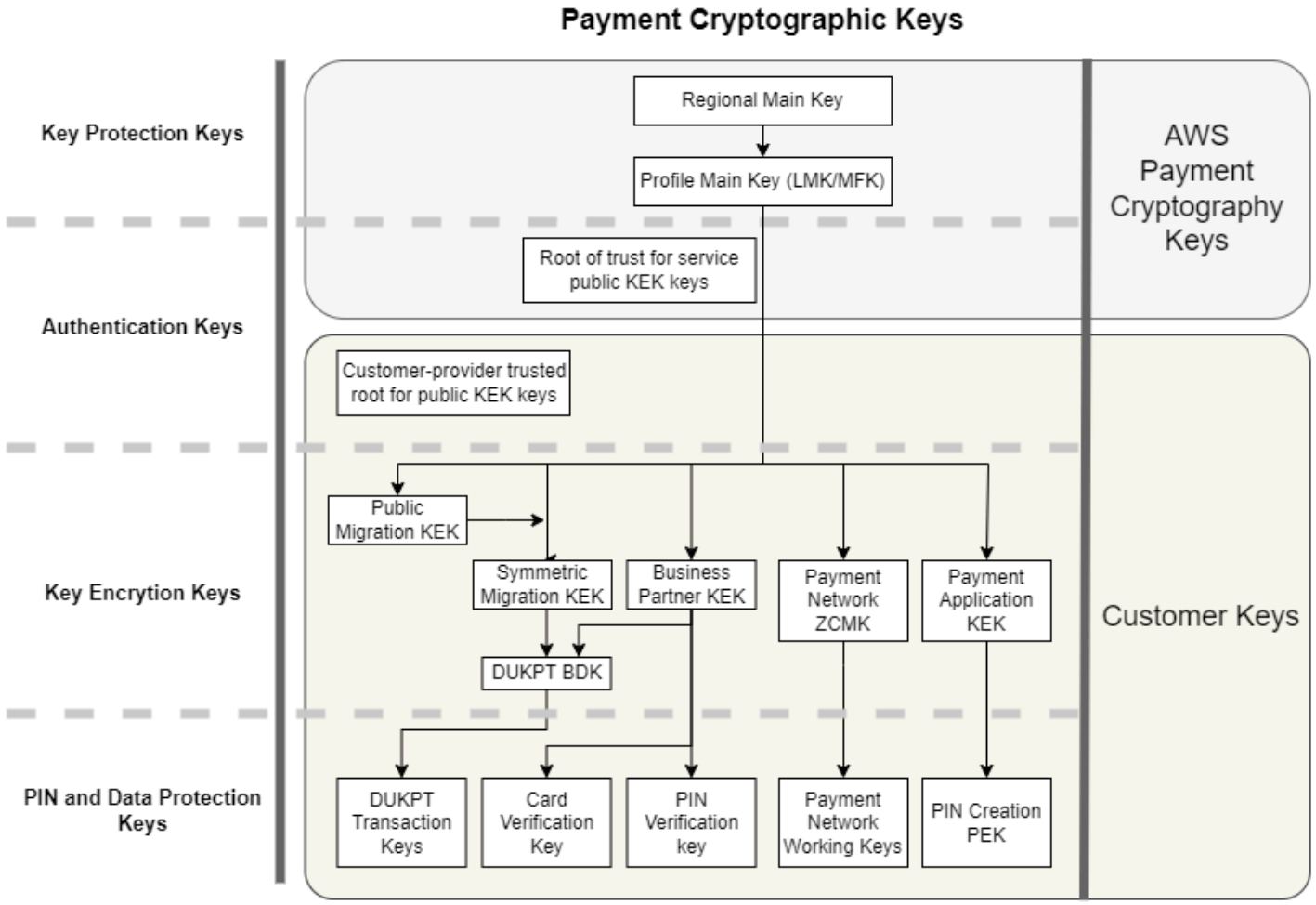
Kunci dapat diekspor dan dilindungi oleh kunci dengan yang sesuai KeyUsage dan yang sekuat atau lebih kuat dari kunci yang akan diekspor.

Protokol Kunci Per Transaksi Unik Berasal (DUKPT)

AWS Kriptografi Pembayaran mendukung dengan kunci derivasi dasar TDEA dan AES (BDK) seperti yang dijelaskan oleh ANSI X9.24-3.

Hirarki kunci

Hirarki kunci Kriptografi AWS Pembayaran memastikan bahwa kunci selalu dilindungi oleh kunci sekuat atau lebih kuat dari kunci yang mereka lindungi.



AWS Kunci Kriptografi Pembayaran digunakan untuk perlindungan kunci dalam layanan:

Kunci	Deskripsi
Kunci Utama Regional	Melindungi gambar HSM virtual, atau profil, yang digunakan untuk pemrosesan kriptografi. Kunci ini hanya ada di HSM dan backup aman.
Profil Kunci Utama	Kunci perlindungan kunci pelanggan tingkat atas, secara tradisional disebut Local Master Key (LMK) atau Master File Key (MFK) untuk kunci pelanggan. Kunci ini hanya ada di HSM dan backup aman. Profil mendefinisikan konfigurasi HSM yang berbeda seperti yang

Kunci	Deskripsi
	dipersyaratkan oleh standar keamanan untuk kasus penggunaan pembayaran.
Akar kepercayaan untuk kunci kunci enkripsi kunci publik (KEK) Kriptografi AWS Pembayaran	Kunci publik root tepercaya dan sertifikat untuk mengautentikasi dan memvalidasi kunci publik yang disediakan oleh Kriptografi AWS Pembayaran untuk impor dan ekspor kunci menggunakan kunci asimetris.

Kunci pelanggan dikelompokkan berdasarkan kunci yang digunakan untuk melindungi kunci dan kunci lain yang melindungi data terkait pembayaran. Ini adalah contoh kunci pelanggan dari kedua jenis:

Kunci	Deskripsi
Root tepercaya yang disediakan pelanggan untuk kunci KEK publik	Kunci publik dan sertifikat yang diberikan oleh Anda sebagai akar kepercayaan untuk mengautentikasi dan memvalidasi kunci publik yang Anda berikan untuk impor dan ekspor kunci menggunakan kunci asimetris.
Kunci Enkripsi Kunci (KEK)	KEK digunakan semata-mata untuk mengenkripsi kunci lain untuk pertukaran antara toko kunci eksternal dan Kriptografi AWS Pembayaran, mitra bisnis, jaringan pembayaran, atau aplikasi lain dalam organisasi Anda.
Kunci turunan Unik Per Transaksi (DUKPT) kunci derivasi dasar (BDK)	BDK digunakan untuk membuat kunci unik untuk setiap terminal pembayaran dan menerjemahkan transaksi dari beberapa terminal ke bank pengakuisisi tunggal, atau pengakuisisi, kunci kerja. Praktik terbaik, yang diperlukan oleh PCI Point-to-Point Encryption (P2PE), adalah bahwa BDK yang berbeda digunakan untuk model terminal yang berbeda,

Kunci	Deskripsi
	layanan injeksi atau inisialisasi kunci, atau segmentasi lain untuk membatasi dampak kompromi BDK.
Kunci master kontrol zona jaringan pembayaran (ZCMK)	ZCMK, juga disebut sebagai kunci zona atau kunci master zona, disediakan oleh jaringan pembayaran untuk membuat kunci kerja awal.
Kunci transaksi DUKPT	Terminal pembayaran yang dikonfigurasi untuk DUKPT memperoleh kunci unik untuk terminal dan transaksi. HSM yang menerima transaksi dapat menentukan kunci dari pengenal terminal dan nomor urut transaksi.
Kunci persiapan data kartu	Kunci master penerbit EMV, kunci kartu EMV dan nilai verifikasi, dan kunci perlindungan file data personalisasi kartu digunakan untuk membuat data untuk masing-masing kartu untuk digunakan oleh penyedia personalisasi kartu. Kunci dan data validasi kriptografi ini juga digunakan oleh bank penerbit, atau penerbit, untuk mengotentikasi data kartu sebagai bagian dari otorisasi transaksi.
Kunci persiapan data kartu	Kunci master penerbit EMV, kunci kartu EMV dan nilai verifikasi, dan kunci perlindungan file data personalisasi kartu digunakan untuk membuat data untuk masing-masing kartu untuk digunakan oleh penyedia personalisasi kartu. Kunci dan data validasi kriptografi ini juga digunakan oleh bank penerbit, atau penerbit, untuk mengotentikasi data kartu sebagai bagian dari otorisasi transaksi.

Kunci	Deskripsi
Kunci kerja jaringan pembayaran	Sering disebut sebagai kunci kerja penerbit atau kunci kerja pengakuisisi, ini adalah kunci yang mengenkripsi transaksi yang dikirim ke atau diterima dari jaringan pembayaran. Kunci-kunci ini sering diputar oleh jaringan, sering setiap hari atau setiap jam. Ini adalah kunci enkripsi PIN (PEK) untuk transaksi PIN/debit.
Kunci enkripsi Nomor Identifikasi Pribadi (PIN) (PEK)	Aplikasi yang membuat atau mendekripsi blok PIN menggunakan PEK untuk mencegah penyimpanan atau transmisi PIN teks yang jelas.

Operasi internal

Topik ini menjelaskan persyaratan internal yang diterapkan oleh layanan untuk mengamankan kunci pelanggan dan operasi kriptografi untuk kriptografi pembayaran yang didistribusikan secara global dan terukur dan layanan manajemen kunci.

Spesifikasi dan siklus hidup HSM

AWS Kriptografi Pembayaran menggunakan armada HSM yang tersedia secara komersial. HSM adalah FIPS 140-2 Level 3 yang divalidasi dan juga menggunakan versi firmware dan kebijakan keamanan yang tercantum pada daftar Perangkat PCI PCI PTS yang disetujui Dewan Standar Keamanan PCI sebagai keluhan PCI HSM v3. Standar PCI PTS HSM mencakup persyaratan tambahan untuk pembuatan, pengiriman, penyebaran, manajemen, dan penghancuran perangkat keras HSM yang penting untuk keamanan dan kepatuhan pembayaran tetapi tidak ditangani oleh FIPS 140.

Semua HSM dioperasikan dalam Mode PCI dan dikonfigurasi dengan kebijakan keamanan PCI PTS HSM. Hanya fungsi yang diperlukan untuk mendukung kasus penggunaan Kriptografi AWS Pembayaran yang diaktifkan. AWS Kriptografi Pembayaran tidak menyediakan pencetakan, tampilan, atau pengembalian PIN teks yang jelas.

Keamanan fisik perangkat HSM

Hanya HSM yang memiliki kunci perangkat yang ditandatangani oleh otoritas sertifikat Kriptografi AWS Pembayaran (CA) oleh produsen sebelum pengiriman yang dapat digunakan oleh layanan. Kriptografi AWS Pembayaran adalah sub-CA dari CA pabrikan yang merupakan akar kepercayaan untuk produsen HSM dan sertifikat perangkat. CA pabrikan mengimplementasikan ANSI TR 34 dan telah membuktikan kepatuhan dengan PCI PIN Security Annex A dan PCI P2PE Annex A. Pabrikan memverifikasi bahwa semua HSM dengan kunci perangkat yang ditandatangani oleh Payment Cryptography CA dikirim ke penerima yang ditunjuk AWS. AWS

Seperti yang dipersyaratkan oleh PCI PIN Security, pabrikan memasok daftar nomor seri melalui saluran komunikasi yang berbeda dari pengiriman HSM. Nomor seri ini diperiksa pada setiap langkah dalam proses instalasi HSM ke pusat data AWS. Akhirnya, operator Kriptografi AWS Pembayaran memvalidasi daftar HSM yang diinstal terhadap daftar pabrikan sebelum menambahkan nomor seri ke daftar HSM yang diizinkan untuk menerima AWS kunci Kriptografi Pembayaran.

HSM berada dalam penyimpanan aman atau di bawah kendali ganda setiap saat, yang meliputi:

- Pengiriman dari pabrikan ke fasilitas perakitan rak AWS.
- Selama perakitan rak.
- Pengiriman dari fasilitas perakitan rak ke pusat data.
- Tanda terima dan pemasangan ke ruang pemrosesan aman pusat data. Rak HSM memberlakukan kontrol ganda dengan kunci yang dikontrol akses kartu, sensor pintu alarm, dan kamera.
- Selama operasi.
- Selama dekomisioning dan penghancuran.

Lengkap chain-of-custody, dengan akuntabilitas individu, dipertahankan dan dipantau untuk setiap HSM.

Inisialisasi HSM

HSM hanya diinisialisasi sebagai bagian dari armada Kriptografi AWS Pembayaran setelah identitas dan integritasnya divalidasi oleh nomor seri, kunci perangkat yang diinstal pabrikan, dan checksum firmware. Setelah keaslian dan integritas HSM divalidasi, itu dikonfigurasi, termasuk mengaktifkan Mode PCI. Kemudian kunci utama wilayah Kriptografi AWS Pembayaran dan kunci utama profil ditetapkan dan HSM tersedia untuk layanan.

Layanan dan perbaikan HSM

HSM memiliki komponen yang dapat diservis yang tidak memerlukan pelanggaran batas kriptografi perangkat. Komponen-komponen ini termasuk kipas pendingin, catu daya, dan baterai. Jika HSM atau perangkat lain dalam rak HSM membutuhkan servis, kontrol ganda dipertahankan selama seluruh periode rak terbuka.

Penonaktifan HSM

Penonaktifan terjadi karena end-of-life atau kegagalan HSM. HSM secara logis di-zero sebelum dikeluarkan dari raknya, jika berfungsi, kemudian dihancurkan di dalam ruang pemrosesan yang aman dari pusat data AWS. Mereka tidak pernah dikembalikan ke pabrikan untuk diperbaiki, digunakan untuk tujuan lain, atau dikeluarkan dari ruang pemrosesan yang aman sebelum kehancuran.

Pembaruan firmware HSM

Pembaruan firmware HSM diterapkan bila diperlukan untuk menjaga keselarasan dengan versi terdaftar PCI PTS HSM dan FIPS 140-2 (atau FIPS 140-3), jika pembaruan terkait keamanan, atau ditentukan bahwa pelanggan dapat memperoleh manfaat dari fitur dalam versi baru. AWS Kriptografi Pembayaran HSM menjalankan off-the-shelf firmware, cocok dengan versi PCI PTS HSM yang terdaftar. Versi firmware baru divalidasi untuk integritas dengan versi firmware bersertifikat PCI atau FIPS kemudian diuji fungsionalitasnya sebelum diluncurkan ke semua HSM.

Akses operator

Operator dapat memiliki akses non-konsol ke HSM untuk pemecahan masalah dalam kasus yang jarang terjadi bahwa informasi yang dikumpulkan dari HSM selama operasi normal tidak cukup untuk mengidentifikasi masalah atau merencanakan perubahan. Langkah-langkah berikut dijalankan:

- Kegiatan pemecahan masalah dikembangkan dan disetujui dan sesi non-konsol dijadwalkan.
- HSM dihapus dari layanan pemrosesan pelanggan.
- Tombol utama dihapus, di bawah kendali ganda.
- Operator diizinkan akses non-konsol ke HSM untuk melakukan aktivitas pemecahan masalah yang disetujui, di bawah kendali ganda.
  - Setelah penghentian sesi non-konsol, proses penyediaan awal dilakukan pada HSM, mengembalikan firmware dan konfigurasi standar, kemudian menyinkronkan kunci utama, sebelum mengembalikan HSM untuk melayani pelanggan.

- Catatan sesi dicatat dalam pelacakan perubahan.
- Informasi yang diperoleh dari sesi digunakan untuk merencanakan perubahan masa depan.

Semua catatan akses non-konsol ditinjau untuk kepatuhan proses dan potensi perubahan pada pemantauan HSM, proses non-console-access manajemen, atau pelatihan operator.

Manajemen kunci

Semua HSM di suatu wilayah disinkronkan ke Kunci Utama Wilayah. Kunci Utama Wilayah melindungi setidaknya satu Kunci Utama Profil. Kunci Utama Profil melindungi kunci pelanggan.

Semua kunci utama dihasilkan oleh HSM dan didistribusikan dengan distribusi kunci simetris menggunakan teknik asimetris, selaras dengan ANSI X9 TR 34 dan PCI PIN Lampiran A.

Topik

Generasi

Kunci utama AES 256 bit dihasilkan pada salah satu HSM yang disediakan untuk armada layanan HSM, menggunakan generator nomor acak PCI PTS HSM.

Sinkronisasi kunci utama wilayah

Kunci utama wilayah HSM disinkronkan oleh layanan di seluruh armada regional dengan mekanisme yang ditentukan oleh ANSI X9 TR-34, yang meliputi:

- Otentikasi bersama menggunakan kunci dan sertifikat host distribusi kunci (KDH) dan perangkat penerima kunci (KRD) untuk memberikan otentikasi dan integritas untuk kunci publik.
- Sertifikat ditandatangani oleh otoritas sertifikat (CA) yang memenuhi persyaratan PCI PIN Annex A2, kecuali untuk algoritma asimetris dan kekuatan kunci yang sesuai untuk melindungi kunci AES 256 bit.
- Identifikasi dan perlindungan kunci untuk kunci simetris terdistribusi yang konsisten dengan ANSI X9 TR-34 dan PCI PIN Annex A1, kecuali untuk algoritme asimetris dan kekuatan kunci yang sesuai untuk melindungi kunci AES 256 bit.

Kunci utama wilayah dibuat untuk HSM yang telah diautentikasi dan disediakan untuk suatu wilayah dengan:

- Kunci utama dihasilkan pada HSM di wilayah tersebut. HSM itu ditetapkan sebagai host distribusi utama.
- Semua HSM yang disediakan di wilayah tersebut menghasilkan token otentikasi KRD, yang berisi kunci publik HSM dan informasi otentikasi yang tidak dapat diputar ulang.
- Token KRD ditambahkan ke daftar izin KDH setelah KDH memvalidasi identitas dan izin HSM untuk menerima kunci.
- KDH menghasilkan token kunci utama yang dapat diautentikasi untuk setiap HSM. Token berisi informasi otentikasi KDH dan kunci utama terenkripsi yang hanya dapat dimuat pada HSM yang telah dibuat untuknya.
- Setiap HSM dikirimkan token kunci utama yang dibuat untuk itu. Setelah memvalidasi informasi otentikasi HSM sendiri dan informasi otentikasi KDH, kunci utama didekripsi oleh kunci pribadi KRD dan dimuat ke kunci utama.

Jika satu HSM harus disinkronkan ulang dengan suatu wilayah:

- Ini divalidasi ulang dan disediakan dengan firmware dan konfigurasi.
- Jika baru di wilayah ini:
  - HSM menghasilkan token otentikasi KRD.
  - KDH menambahkan token ke daftar izinnya.
  - KDH menghasilkan token kunci utama untuk HSM.
  - HSM memuat kunci utama.
  - HSM tersedia untuk layanan ini.

Ini memastikan bahwa:

- Hanya HSM yang divalidasi untuk pemrosesan Kriptografi AWS Pembayaran dalam suatu wilayah yang dapat menerima kunci utama wilayah tersebut.
- Hanya kunci master dari Kriptografi AWS Pembayaran HSM yang dapat didistribusikan ke HSM di armada.

Rotasi kunci utama wilayah

Kunci utama wilayah diputar pada saat berakhirnya periode kripto, jika terjadi dugaan kompromi kunci, atau setelah perubahan pada layanan yang ditentukan untuk memengaruhi keamanan kunci.

Kunci utama wilayah baru dihasilkan dan didistribusikan seperti penyediaan awal. Kunci utama profil yang disimpan harus diterjemahkan ke kunci utama wilayah baru.

Rotasi kunci utama wilayah tidak memengaruhi pemrosesan pelanggan.

Sinkronisasi kunci utama profil

Kunci utama profil dilindungi oleh kunci utama wilayah. Ini membatasi profil ke wilayah tertentu.

Kunci utama profil disediakan sesuai:

- Kunci utama profil dihasilkan pada HSM yang memiliki kunci utama wilayah yang disinkronkan.
- Kunci utama profil disimpan dan dienkripsi dengan konfigurasi profil dan konteks lainnya.
- Profil ini digunakan untuk fungsi kriptografi pelanggan oleh HSM mana pun di wilayah dengan kunci utama wilayah.

Profil rotasi kunci utama

Kunci utama profil diputar pada saat berakhirnya periode kripto, setelah dugaan kompromi kunci, atau setelah perubahan pada layanan yang ditentukan untuk memengaruhi keamanan kunci.

Langkah-langkah rotasi:

- Kunci utama profil baru dihasilkan dan didistribusikan sebagai kunci utama yang tertunda seperti penyediaan awal.
- Proses latar belakang menerjemahkan materi kunci pelanggan dari kunci utama profil yang ditetapkan ke kunci utama yang tertunda.

- Ketika semua kunci pelanggan telah dienkripsi dengan kunci yang tertunda, kunci yang tertunda dipromosikan ke kunci utama profil.
- Proses latar belakang menghapus materi kunci pelanggan yang dilindungi oleh kunci kedaluwarsa.

Rotasi kunci utama profil tidak memengaruhi pemrosesan pelanggan.

Perlindungan

Kunci hanya bergantung pada hierarki kunci untuk perlindungan. Perlindungan kunci utama sangat penting untuk mencegah kehilangan atau membahayakan semua kunci pelanggan.

Kunci utama wilayah dapat dipulihkan dari cadangan hanya ke HSM yang diautentikasi dan disediakan untuk layanan. Kunci ini hanya dapat disimpan sebagai token kunci utama yang dapat diautentikasi dan dienkripsi dari KDH tertentu untuk HSM tertentu.

Kunci master profil disimpan dengan konfigurasi profil dan informasi konteks yang dienkripsi berdasarkan wilayah.

Kunci pelanggan disimpan dalam blok kunci, dilindungi oleh kunci master profil.

Semua kunci ada secara eksklusif dalam HSM atau disimpan dilindungi oleh kunci lain dengan kekuatan kriptografi yang sama atau lebih kuat.

Daya tahan

Kunci pelanggan untuk kriptografi transaksi dan fungsi bisnis harus tersedia bahkan dalam situasi ekstrem yang biasanya menyebabkan pemadaman. AWS Kriptografi Pembayaran menggunakan model redundansi tingkat ganda di seluruh zona dan wilayah ketersediaan. AWS Pelanggan yang membutuhkan ketersediaan dan daya tahan yang lebih tinggi untuk operasi kriptografi pembayaran daripada yang disediakan oleh layanan harus menerapkan arsitektur multi-wilayah.

Otentikasi HSM dan token kunci utama disimpan dan dapat digunakan untuk mengembalikan kunci utama atau menyinkronkan dengan kunci utama baru, jika HSM harus diatur ulang. Token diarsipkan dan digunakan hanya di bawah kontrol ganda bila diperlukan.

Keamanan komunikasi

Eksternal

AWS Titik akhir API Kriptografi Pembayaran memenuhi standar AWS keamanan termasuk TLS pada atau di atas 1.2 dan Signature Versi 4 untuk otentikasi dan integritas permintaan.

Koneksi TLS yang masuk dihentikan pada penyeimbang beban jaringan dan diteruskan ke penangan API melalui koneksi TLS internal.

Internal

Komunikasi internal antara komponen layanan dan antara komponen layanan dan layanan AWS lainnya dilindungi oleh TLS menggunakan kriptografi yang kuat.

HSM berada di jaringan pribadi non-virtual yang hanya dapat dijangkau dari komponen layanan. Semua koneksi antara HSM dan komponen layanan diamankan dengan TLS bersama (MTL), pada atau di atas TLS 1.2. Sertifikat internal untuk TLS dan mTL dikelola oleh Amazon Certificate Manager menggunakan AWS Private Certificate Authority. VPC internal dan jaringan HSM dipantau untuk aktivitas dan perubahan konfigurasi yang tidak terkecuali.

Manajemen kunci pelanggan

Pada AWS, kepercayaan pelanggan adalah prioritas utama kami. Anda mempertahankan kontrol penuh atas kunci yang Anda unggah atau buat di layanan di bawah akun AWS Anda dan bertanggung jawab untuk mengonfigurasi akses ke kunci.

AWS Kriptografi Pembayaran memiliki tanggung jawab penuh atas kepatuhan fisik HSM dan manajemen kunci untuk kunci yang dikelola oleh layanan. Ini membutuhkan kepemilikan dan pengelolaan kunci utama HSM dan penyimpanan kunci pelanggan yang dilindungi dalam basis data kunci Kriptografi AWS Pembayaran.

Pemisahan ruang kunci pelanggan

AWS Kriptografi Pembayaran memberlakukan kebijakan utama untuk semua penggunaan kunci, termasuk membatasi prinsipal ke akun yang memiliki kunci, kecuali kunci secara eksplisit dibagikan dengan akun lain.

Pencadangan dan pemulihan

Kunci dan informasi kunci untuk suatu wilayah dicadangkan ke arsip terenkripsi oleh. AWS Arsip membutuhkan kontrol ganda AWS untuk memulihkan.

Blok kunci

Semua kunci disimpan dalam blok kunci format ANSI X9 TR-31.

Kunci dapat diimpor ke layanan dari kriptogram atau format blok kunci lainnya yang didukung oleh ImportKey. Demikian pula, kunci dapat diekspor, jika dapat diekspor, ke format blok kunci lain atau kriptogram yang didukung oleh profil ekspor utama.

Penggunaan kunci

Penggunaan kunci dibatasi untuk yang dikonfigurasi KeyUsage oleh layanan. Layanan akan gagal setiap permintaan dengan penggunaan kunci yang tidak tepat, mode penggunaan, atau algoritma untuk operasi kriptografi yang diminta.

Hubungan pertukaran kunci

PCI PIN Security dan PCI P2PE mengharuskan organisasi yang berbagi kunci yang mengenkripsi PIN, termasuk KEK yang digunakan untuk berbagi kunci tersebut, tidak berbagi kunci tersebut dengan organisasi lain. Ini adalah praktik terbaik bahwa kunci simetris dibagi antara hanya 2 pihak, termasuk dalam organisasi yang sama. Ini meminimalkan dampak dari dugaan kompromi kunci yang memaksa penggantian kunci yang terkena dampak.

Bahkan kasus bisnis yang memerlukan kunci berbagi antara lebih dari 2 pihak, harus menjaga jumlah pihak ke jumlah minimum.

AWS Kriptografi Pembayaran menyediakan tag kunci yang dapat digunakan untuk melacak dan menegakkan penggunaan kunci dalam persyaratan tersebut.

Misalnya, KEK dan BDK untuk fasilitas injeksi kunci yang berbeda dapat diidentifikasi dengan menetapkan “KIF” = “POSStation” untuk semua kunci yang dibagikan dengan penyedia layanan tersebut. Contoh lain adalah menandai kunci yang dibagikan dengan jaringan pembayaran dengan “Jaringan” = “PayCard”. Penandaan memungkinkan Anda membuat kontrol akses dan membuat laporan audit untuk menegakkan dan mendemonstrasikan praktik manajemen utama Anda.

Penghapusan kunci

DeleteKey menandai kunci dalam database untuk dihapus setelah periode yang dapat dikonfigurasi pelanggan. Setelah periode ini kuncinya dihapus secara permanen. Ini adalah mekanisme keamanan untuk mencegah penghapusan kunci yang tidak disengaja atau berbahaya. Kunci yang ditandai untuk penghapusan tidak tersedia untuk tindakan apa pun kecuali. RestoreKey

Kunci yang dihapus tetap dalam cadangan layanan selama 7 hari setelah penghapusan. Mereka tidak dapat dipulihkan selama periode ini.

Kunci milik akun AWS tertutup ditandai untuk dihapus. Jika akun diaktifkan kembali sebelum periode penghapusan tercapai, kunci apa pun yang ditandai untuk penghapusan dipulihkan, tetapi dinonaktifkan. Mereka harus diaktifkan kembali oleh Anda untuk menggunakannya untuk operasi kriptografi.

Berbagi kunci

Kunci dapat dibagikan dengan akun lain di dalam atau di luar organisasi Anda menggunakan AWS Resource Access Manager (https://docs.aws.amazon.com/ARG/index.html). Kunci dapat dikelompokkan dalam pembagian sumber daya kemudian dibagikan dengan akun atau pengguna IAM tertentu dan peran dalam akun. Anda menentukan izin penggunaan untuk setiap pembagian sumber daya. Izin berbagi dibatasi oleh kebijakan sumber daya kunci. Kunci bersama tidak akan mengizinkan tindakan yang dibatasi oleh kebijakannya sendiri. Izin berbagi dapat ditarik kapan saja.

Pencatatan log dan pemantauan

Log layanan internal meliputi:

- CloudTrail log panggilan layanan AWS yang dilakukan oleh layanan
- CloudWatch log dari kedua peristiwa langsung masuk ke CloudWatch log atau peristiwa dari HSM
- File log dari HSM dan sistem layanan
- Arsip log

Semua sumber log memantau dan memfilter informasi sensitif, termasuk tentang kunci. Log ditinjau secara sistematis untuk memastikan bahwa mereka mengandung tidak mengandung informasi pelanggan yang sensitif.

Akses ke log dibatasi untuk individu yang dibutuhkan untuk menyelesaikan peran pekerjaan.

Semua log disimpan selaras dengan kebijakan penyimpanan log AWS.

Operasi pelanggan

AWS Kriptografi Pembayaran memiliki tanggung jawab penuh atas kepatuhan fisik HSM berdasarkan standar PCI. Layanan ini juga menyediakan penyimpanan kunci yang aman dan memastikan bahwa kunci hanya dapat digunakan untuk tujuan yang diizinkan oleh standar PCI dan ditentukan oleh Anda selama pembuatan atau impor. Anda bertanggung jawab untuk mengonfigurasi atribut utama dan akses untuk memanfaatkan kemampuan keamanan dan kepatuhan layanan.

Topik

Menghasilkan kunci

Saat membuat kunci, Anda menetapkan atribut yang digunakan layanan untuk menerapkan penggunaan kunci yang sesuai:

- Algoritma dan panjang kunci
- Penggunaan
- Ketersediaan dan kedaluwarsa

Tag yang digunakan untuk kontrol akses berbasis atribut (ABAC) digunakan untuk membatasi kunci untuk digunakan dengan mitra tertentu atau aplikasi juga harus ditetapkan selama pembuatan. Pastikan untuk menyertakan kebijakan untuk membatasi peran yang diizinkan untuk menghapus atau mengubah tag.

Anda harus memastikan bahwa kebijakan yang menentukan peran yang mungkin menggunakan dan mengelola kunci ditetapkan sebelum pembuatan kunci.

> **Note**
>
> Kebijakan IAM pada CreateKey perintah dapat digunakan untuk menegakkan dan mendemonstrasikan kontrol ganda untuk pembuatan kunci.

Mengimpor kunci

Saat mengimpor kunci, atribut untuk menerapkan penggunaan kunci yang sesuai ditetapkan oleh layanan menggunakan informasi yang terikat secara kriptografis di blok kunci. Mekanisme untuk mengatur konteks kunci fundamental adalah dengan menggunakan blok kunci yang dibuat dengan

sumber HSM dan dilindungi oleh KEK bersama atau asimetris. Ini sejalan dengan persyaratan PIN PCI dan mempertahankan penggunaan, algoritme, dan kekuatan kunci dari aplikasi sumber.

Atribut kunci penting, tag, dan kebijakan kontrol akses harus ditetapkan pada impor selain informasi di blok kunci.

Mengimpor kunci menggunakan kriptogram tidak mentransfer atribut kunci dari aplikasi sumber. Anda harus mengatur atribut dengan tepat dengan menggunakan mekanisme ini.

Seringkali kunci dipertukarkan menggunakan komponen teks yang jelas, ditransmisikan oleh penjaga kunci, kemudian dimuat dengan upacara yang menerapkan kontrol ganda di ruang aman. Ini tidak didukung secara langsung oleh Kriptografi AWS Pembayaran. API akan mengekspor kunci publik dengan sertifikat yang dapat diimpor oleh HSM Anda sendiri untuk mengekspor blok kunci yang dapat diimpor oleh layanan. Ini memungkinkan penggunaan HSM Anda sendiri untuk memuat komponen teks yang jelas.

Anda harus menggunakan Nilai cek kunci (KCV) untuk memverifikasi bahwa kunci yang diimpor cocok dengan kunci sumber.

Kebijakan IAM pada ImportKey API dapat digunakan untuk menegakkan dan menunjukkan kontrol ganda untuk impor kunci.

Mengekspor kunci

Berbagi kunci dengan mitra atau aplikasi lokal mungkin memerlukan kunci ekspor. Menggunakan blok kunci untuk ekspor mempertahankan konteks kunci fundamental dengan materi kunci terenkripsi.

Tag kunci dapat digunakan untuk membatasi ekspor kunci ke KEK yang berbagi tag dan nilai yang sama.

AWS Kriptografi Pembayaran tidak menyediakan atau menampilkan komponen kunci teks yang jelas. Ini memerlukan akses langsung oleh penjaga kunci ke PCI PTS HSM atau ISO 13491 perangkat kriptografi aman (SCD) yang diuji untuk tampilan atau pencetakan. Anda dapat membuat KEK asimetris atau KEK simetris dengan SCD Anda untuk melakukan upacara pembuatan komponen kunci teks yang jelas di bawah kendali ganda.

Nilai pemeriksaan kunci (KCV) harus digunakan untuk memverifikasi bahwa diimpor oleh kunci sumber pencocokan HSM tujuan.

Menghapus kunci

Anda dapat menggunakan API kunci hapus untuk menjadwalkan kunci untuk dihapus setelah periode waktu yang Anda konfigurasi. Sebelum itu kunci waktu dapat dipulihkan. Setelah kunci dihapus, kunci akan dihapus secara permanen dari layanan.

Kebijakan IAM pada DeleteKey API dapat digunakan untuk menegakkan dan menunjukkan kontrol ganda untuk penghapusan kunci.

Merotasi kunci

Efek rotasi kunci dapat diimplementasikan menggunakan alias kunci dengan membuat atau mengimpor kunci baru, kemudian memodifikasi alias kunci untuk merujuk ke kunci baru. Kunci lama akan dihapus atau dinonaktifkan, tergantung pada praktik manajemen Anda.

Kuota untuk AWS Payment Cryptography

Akun AWS Anda memiliki kuota default, yang sebelumnya disebut sebagai batas, untuk setiap layanan AWS. Kecuali dinyatakan lain, setiap kuota bersifat spesifik wilayah. Anda dapat meminta penambahan untuk beberapa kuota, sementara kuota lainnya tidak dapat ditambah.

Nama	Default	Dapat disesuaikan	Deskripsi
Alias	Setiap Wilayah yang didukung: 2.000	Ya	Jumlah maksimum alias yang dapat Anda miliki di akun ini di Wilayah saat ini.
Tingkat gabungan permintaan bidang kontrol	Setiap Wilayah yang didukung: 5 per detik	Ya	Jumlah maksimum permintaan pesawat kontrol per detik yang dapat Anda buat di akun ini di Wilayah saat ini. Kuota ini berlaku untuk semua operasi pesawat kontrol yang digabungkan.
Tingkat gabungan permintaan bidang data (asimetris)	Setiap Wilayah yang didukung: 20 per detik	Ya	Jumlah maksimum permintaan per detik untuk operasi bidang data dengan kunci asimetris yang dapat Anda buat di akun ini di Wilayah saat ini. Kuota ini berlaku untuk semua operasi pesawat data yang digabungkan.

Nama	Default	Dapat disesuaikan	Deskripsi
Tingkat gabungan permintaan bidang data (simetris)	Setiap Wilayah yang didukung: 500 per detik	Ya	Jumlah maksimum permintaan per detik untuk operasi bidang data dengan kunci simetris yang dapat Anda buat di akun ini di Wilayah saat ini. Kuota ini berlaku untuk semua operasi pesawat data yang digabungkan.
Kunci	Setiap Wilayah yang didukung: 2.000	Ya	Jumlah maksimum kunci yang dapat Anda miliki di akun ini di Wilayah saat ini, tidak termasuk kunci yang dihapus.

Riwayat dokumen untuk Panduan Pengguna Kriptografi AWS Pembayaran

Tabel berikut menjelaskan rilis dokumentasi untuk Kriptografi AWS Pembayaran.

Perubahan	Deskripsi	Tanggal
Rilis fitur	Menambahkan informasi tentang titik akhir VPC (PrivateLink) dan contoh iCVV.	30 Mei 2024
Rilis fitur	Informasi ditambahkan pada fitur baru seputar impor/ekspor kunci menggunakan RSA dan mengekspor kunci DUKPT IPEK/IK.	Januari 15, 2024
Rilis awal	Rilis awal Panduan Pengguna Kriptografi AWS Pembayaran	8 Juni 2023

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