Guía del usuario de

AWS Payment Cryptography



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

AWS Payment Cryptography: Guía del usuario de

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

Las marcas registradas y la imagen comercial de Amazon no se pueden utilizar en ningún producto o servicio que no sea de Amazon de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es la criptografía de AWS pagos? 1
Conceptos 2
Terminología del sector 4
Tipos de claves comunes 4
Otros términos 8
Servicios relacionados 13
Para obtener más información 14
Puntos de conexión 14
Puntos de conexión del plano de control 14
Puntos de conexión del plano de datos 17
Introducción 21
Requisitos previos 21
Paso 1: crear una clave 22
Paso 2: Genere un valor con la clave CVV2 23
Paso 3: verificar el valor generado en el paso 2 23
Paso 4: realizar una prueba negativa 24
Paso 5: eliminación (opcional) 24
Administración de claves 26
Crear claves 26
Creación de una clave de derivación base TDES de 2KEY 27
Crear una clave TDES de 2 CLAVES para CVV/CVV2 29
Crear una clave HMAC 30
Crear una clave AES-256 31
Creación de una clave de cifrado PIN (PEK) 32
Crear una clave asimétrica (RSA) 33
Creación de una clave de valor de verificación del PIN (PVV) 34
Creación de una clave ECC asimétrica 35
Enumerar las claves 36
Habilitación y desactivación de claves de 38
Iniciar el uso de claves 38
Detener el uso de claves 40
Claves de replicación 42
Ventajas de la replicación de claves Multi-Region 42
Cómo funciona la Multi-Region replicación clave 42

Limitaciones y consideraciones 42
Habilitar Multi-Region la replicación de claves 44
Deshabilitar Multi-Region la replicación de claves 46
Consideraciones de seguridad 47
Prácticas recomendadas 47
Precios 47
Eliminación de claves de 48
Acerca del período de espera 49
Importación y exportación de claves 52
Importar claves 54
Exportar claves 80
Temas avanzados 103
Uso de alias 122
Acerca de los alias 123
Usar alias en las aplicaciones 126
API relacionadas 127
Obtener claves 127
Consiga que el público se key/certificate asocie a un par de claves 129
Etiquetado de claves 130
Acerca de las etiquetas en AWS Payment Cryptography 130
Visualización de etiquetas clave en la consola 132
Administración de etiquetas de clave con operaciones de la API 132
Control del acceso a las etiquetas 135
Uso de etiquetas para controlar el acceso a las claves 139
Comprender los atributos de las claves 143
Claves simétricas 143
Claves asimétricas 145
Operaciones de datos 147
Cifrar, descifrar y volver a cifrar datos 147
Cifrar datos 148
Descifrado de datos 154
Generación y verificación de datos de tarjetas 158
Generar datos de tarjetas 159
Comprobación de datos de tarjetas 160
Generar, traducir y verificar los datos del PIN 162
Traducir datos PIN 163

Generar datos PIN 165
Comprobación de datos PIN 169
Generar y verificar el ARQC 173
Generar criptograma de solicitud de autenticación (ARQC) 174
Verificar el criptograma de solicitud de autenticación (ARQC) 176
Generar y verificar MAC 180
Generar MAC 182
Verificar MAC 186
Tipos de clave para operaciones de datos específicas 188
GenerateCardData 189
VerifyCardData 190
GeneratePinData (para VISA/ABA esquemas) 191
GeneratePinData (para IBM3624) 192
VerifyPinData (para VISA/ABA esquemas) 193
VerifyPinData (para IBM3624) 194
Descifrado de datos 195
Cifrado de datos 196
Traducir datos PIN 197
Generate/Verify MAC 198
GenerateMacEmvPinChange 200
VerifyAuthRequestCryptogram 201
Import/Export Clave 201
Tipos de claves sin utilizar 202
Casos de uso comunes 203
Emisores y procesadores de emisores 203
Funciones generales 203
Funciones específicas de la red 237
Facilitadores de adquisiciones y pagos 270
Uso de claves dinámicas 271
Características específicas de la región 273
AS2805 273
Intercambio de clave inicial (KEK) 275
Validación del KEK 276
Creación y transmisión de claves de trabajo 279
Exportación de claves de trabajo 281
Traducción de pines 282

Generación y validación de Mac 283
Seguridad 284
Protección de datos 284
Rotación del material de claves 286
Cifrado de datos 286
Cifrado en reposo 286
Cifrado en tránsito 287
Privacidad del tráfico entre redes 287
Resiliencia 288
Aislamiento regional 288
Multi-tenant diseño 289
Seguridad de la infraestructura 290
Aislamiento de hosts físicos 290
Utilice Amazon VPC y AWS PrivateLink 290
Consideraciones para AWS Puntos finales de VPC de criptografía de pagos 291
Crear un punto final de VPC para AWS Payment Cryptography 292
Conectar con un punto de conexión de VPC 293
Control del acceso a un punto de conexión de VPC 294
Utilizar un punto de conexión de VPC en una declaración de política 298
Registro de su punto de conexión de VPC 301
TLS híbrido postcuántico 304
Acerca del cifrado TLS postcuántico 305
Modo de uso 306
Acerca de PQC 306
Modo de uso 306
Prácticas recomendadas de seguridad 306
Validación de conformidad 308
Conformidad del servicio 308
Cumplimiento de los PIN 309
Temas comunes 309
Alcance de la evaluación 312
Operaciones de procesamiento de transacciones 314
Conformidad con P2PE 320
Identity and Access Management 321
Público 321
Autenticación con identidades 322

Cuenta de AWS usuario raíz 322
Usuarios y grupos de IAM 322
Roles de IAM 322
Administración del acceso con políticas 323
Identity-based políticas 323
Resource-based políticas 323
Listas de control de acceso (ACL) 324
Otros tipos de políticas 324
Varios tipos de políticas 324
Cómo AWS La criptografía de pagos funciona con IAM 325
AWS Políticas de criptografía de pagos Identity-based 325
Autorización basada en AWS Etiquetas de criptografía de pago 327
Identity-based ejemplos de políticas 327
Prácticas recomendadas relativas a políticas 328
Uso de la consola 329
Permitir a los usuarios consultar sus propios permisos 330
Posibilidad de acceder a todos los aspectos de AWS Payment Cryptography 331
Posibilidad de llamar a las API mediante claves específicas 331
Capacidad para denegar específicamente un recurso 332
Resource-based políticas 333
Consideraciones 334
Administrar políticas basadas en recursos 335
Resource-based ejemplos de políticas 337
Multi-party aprobación 338
Descripción general de 339
Operaciones protegidas 339
Requisitos previos 340
Activación y desactivación de la MPA 340
Introducción 341
Ejemplo: importe un certificado raíz con la MPA habilitada 341
AWS CloudTrail registro de eventos de MPA 343
Comprobar el estado de las solicitudes y gestionar los errores 345
Resolución de problemas 347
Monitorización 348
CloudTrail registros 349
AWS Información sobre criptografía de pagos en CloudTrail 349

Controle los eventos del plano en CloudTrail 350
Eventos de datos en CloudTrail 350
Comprensión AWS de las entradas de los archivos de registro del plano de control de criptografía de pagos 352
Descripción de las entradas de los archivos de registro del plano de datos de criptografía de AWS pagos 355
Detalles criptográficos 358
Objetivos de diseño 359
Principios básicos 360
Primitivas criptográficas 360
Entropía y generación de números aleatorios 361
Operaciones de clave simétrica 361
Operaciones con claves asimétricas 361
Almacenamiento de claves 362
Importación de claves simétricas 362
Importación de claves con claves asimétricas 362
Exportación de claves 363
Protocolo de clave única derivada por transacción (DUKPT) 363
Jerarquía de claves 363
Operaciones internas 367
Protección HSM 367
Administración general de claves 370
Gestión de las claves de los clientes 374
Seguridad de las comunicaciones 376
Registro y supervisión 377
Operaciones de clientes 377
Generación de claves 378
Importación de claves 378
Exportación de claves 379
Eliminación de claves de 380
Rotar claves de 380
Cuotas 381
Historial de revisión 383
..... ccclxxxv

¿Qué es la criptografía de AWS pagos?

AWS La criptografía de pagos es un AWS servicio gestionado que proporciona acceso a las funciones criptográficas y a la gestión de claves que se utilizan en el procesamiento de pagos de conformidad con los estándares del sector de las tarjetas de pago (PCI), sin necesidad de adquirir instancias de HSM de pago dedicadas. AWS La criptografía de pagos ofrece a los clientes que realizan funciones de pago, como los adquirentes, los facilitadores de pagos, las redes, los conmutadores, los procesadores y los bancos, la posibilidad de acercar sus operaciones criptográficas de pago a las aplicaciones en la nube y minimizar la dependencia de los centros de datos auxiliares o las instalaciones de colocación que contienen pagos dedicados HSMs.

El servicio está diseñado para cumplir con las normas aplicables de la industria, incluidas PCI PIN, PCI P2PE y PCI DSS, y aprovecha el hardware que cuenta con la certificación PCI PTS HSM V3 y FIPS 140-2 de nivel 3. Está diseñada para soportar una baja latencia y altos niveles de tiempo de actividad y resiliencia. AWS La criptografía de pagos es totalmente elástica y elimina muchos de los requisitos operativos de las instalaciones locales HSMs, como la necesidad de aprovisionar hardware, gestionar de forma segura el material clave y mantener las copias de seguridad de emergencia en instalaciones seguras. AWS La criptografía de pagos también le ofrece la opción de compartir las claves con sus socios de forma electrónica, lo que elimina la necesidad de compartir componentes de texto transparente en papel.

Puede usar la API del plano de control de AWS Payment Cryptography para crear y administrar claves.

Puede usar la API del plano de datos de AWS Payment Cryptography para usar claves de cifrado para el procesamiento de transacciones relacionadas con los pagos y las operaciones criptográficas asociadas.

AWS La criptografía de pagos ofrece funciones importantes que puede utilizar para gestionar sus claves:

- Cree y gestione claves de criptografía de AWS pagos simétricas y asimétricas, incluidas las claves TDES, AES y RSA, y especifique su propósito, por ejemplo, para la generación de CVV o la obtención de claves DUKPT.
- Guarde automáticamente sus claves de criptografía AWS de pagos de forma segura, protegidas por módulos de seguridad de hardware (), al tiempo que establece la separación de claves entre los casos de uso. HSMs

- Cree, elimine, enumere y actualice los alias, que son «nombres descriptivos» que se pueden usar para acceder o controlar el acceso a sus claves de criptografía de AWS pagos.
- Etiquete sus claves AWS de criptografía de pagos para identificarlas, agruparlas, automatizarlas, controlar el acceso y hacer un seguimiento de los costes.
- Importe y exporte claves simétricas entre AWS Payment Cryptography y su HSM (o terceros) mediante claves de cifrado de claves (KEK) según la norma TR-31 (especificación de bloques de claves de intercambio seguro de claves interoperable).
- Importe y exporte claves de cifrado de claves simétricas (KEK) entre AWS Payment Cryptography y otros sistemas mediante pares de claves asimétricas y, a continuación, utilice medios electrónicos como el TR-34 (método de distribución de claves simétricas mediante técnicas asimétricas).

Puede utilizar sus claves de criptografía de AWS pago en operaciones criptográficas, como:

- Cifre, descifre y vuelva a cifrar los datos con claves de criptografía de pagos simétricas o asimétricas. AWS
- Traducir de forma segura los datos sensibles (como los PIN de los titulares de tarjetas) entre claves de cifrado sin exponer el texto en claro, de acuerdo con las normas PCI sobre PIN.
- Genere o valide los datos del titular de la tarjeta, como el CVV o el ARQC. CVV2
- Generar y validar los pines del titular de la tarjeta.
- Generar o validar las firmas MAC.

Conceptos

Conozca los términos y conceptos básicos que se utilizan en la criptografía de AWS pagos y cómo puede utilizarlos para proteger sus datos.

Alias

> Un nombre fácil de usar que está asociado a una clave de criptografía AWS de pagos. El alias se puede usar indistintamente con la clave ARN en muchas de las operaciones de la API de criptografía de AWS pagos. Los alias permiten rotar o cambiar las claves sin que ello afecte al código de su aplicación. El nombre de alias es una cadena de hasta 256 caracteres. Identifica de forma exclusiva una clave de criptografía AWS de pagos asociada dentro de una cuenta y una región. En la criptografía de AWS pagos, los nombres de alias siempre comienzan por. `alias/`

El formato de un nombre de alias es el siguiente:

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

Por ejemplo:

```
alias/sampleAlias2
```

ARN de clave

El ARN de la clave es el nombre de recurso de Amazon (ARN) de una entrada de clave en AWS Payment Cryptography. Se trata de un identificador único y totalmente cualificado para la clave de criptografía AWS de pagos. Un ARN clave incluye una región Cuenta de AWS, y un ID generado aleatoriamente. El ARN no está relacionado ni se deriva del material de la clave. Como se asignan automáticamente durante las operaciones de creación o importación, estos valores no son idempotentes. Si se importa la misma clave varias veces, se obtendrá una clave múltiple ARNs con su propio ciclo de vida.

El formato de un ARN de clave es el siguiente:

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

A continuación, se muestra un ARN de clave de ejemplo.

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

Identificador de clave

Un identificador de clave es una referencia a una clave y una (o más) de ellas son entradas típicas de las operaciones de criptografía de AWS pagos. Los identificadores de clave válidos pueden ser una clave o un alias de clave.

AWS Claves de criptografía de pago

AWS Las claves de criptografía de pago (claves) se utilizan para todas las funciones criptográficas. Las claves se generan directamente mediante el comando crear clave o se añaden al sistema al llamar a la importación de claves. El origen de una clave se puede determinar revisando el atributo. KeyOrigin AWS La criptografía de pagos también admite claves derivadas o intermedias que se utilizan durante las operaciones criptográficas, como las que utiliza DUKPT.

Estas claves tienen atributos inmutables y mutables definidos en el momento de su creación. Los atributos, como el algoritmo, la longitud y el uso, se definen en el momento de la creación y no se pueden cambiar. Otros, como la fecha de entrada en vigor o la fecha de caducidad, se pueden modificar. Consulte la referencia de la API AWS de criptografía de pagos para obtener una lista completa de los atributos de las claves de criptografía de AWS pagos.

AWS Las claves de criptografía de pago tienen tipos de claves, definidos principalmente en la norma ANSI X9 TR 31, que restringen su uso a los fines previstos, tal como se especifica en el requisito 19 del PCI PIN v3.1.

Los atributos se vinculan a las claves mediante bloques de claves cuando se almacenan, se comparten con otras cuentas o se exportan, tal y como se especifica en el requisito 18-3 de PCI PIN v3.1.

Las claves se identifican en la plataforma de criptografía de AWS pagos mediante un valor único conocido como clave Amazon Resource Name (ARN).

> **Note**
>
> ARNLa clave se genera cuando se crea o importa inicialmente una clave al servicio de criptografía AWS de pagos. Por lo tanto, si añade el mismo material clave varias veces utilizando la funcionalidad de importación de claves, el mismo material clave se encontrará bajo varias claves ARNS pero cada una con un ciclo de vida clave diferente.

Terminología del sector

Temas

- Tipos de claves comunes
- Otros términos

Tipos de claves comunes

AWS Clave de criptografía de pago

Una clave AWS de criptografía de pago existe en una sola. Región de AWS Consiste en metadatos y material clave almacenados en el Servicio de Criptografía AWS de Pagos. Una clave

puede importarse de una fuente externa como un bloque de TR-31 claves o generarse mediante el Servicio de criptografía AWS de pagos.

AWK

Una clave de trabajo del adquirente (AWK) es una clave que se utiliza normalmente para intercambiar datos entre un acquirer/acquirer procesador y una red (como Visa o Mastercard). Históricamente, AWK utiliza el 3DES para el cifrado y se representa como TR31_P0_PIN_ENCRYPTION_KEY.

BDK

Una clave de derivación base (BDK) es una clave funcional que se utiliza para derivar claves posteriores y se utiliza normalmente como parte del proceso PCI PIN y PCI P2PE DUKPT. Se denomina TR31_B0_BASE_DERIVATION_KEY.

CMK

Una clave maestra de tarjeta (CMK) es una o más claves específicas de una tarjeta que normalmente se derivan de una clave maestra del emisor, un PAN y una PSN y, por lo general, son claves 3DES. Estas claves se almacenan en el chip EMV durante la personalización. Entre los ejemplos de CMK se incluyen las teclas AC, SMI y SMC.

CMK-AC

Una clave de criptograma de aplicación (AC) se utiliza como parte de las transacciones EMV para generar el criptograma de la transacción y es un tipo de clave maestra de tarjeta.

CMK-SMI

Una clave de integridad de mensajería segura (SMI) se utiliza como parte de EMV para verificar la integridad de las cargas útiles enviadas a la tarjeta mediante MAC, como los scripts de actualización de PIN. Es un tipo de clave maestra de la tarjeta.

CMK-SMC

Como parte del EMV, se utiliza una clave de confidencialidad segura de la mensajería (SMC) para cifrar los datos enviados a la tarjeta, como las actualizaciones del PIN. Es un tipo de clave maestra de la tarjeta.

CVK

Una clave de verificación de tarjeta (CVK) es una clave que se utiliza para generar valores CVV, CVV2 y similares mediante un algoritmo definido, así como para validar una entrada. Se denomina TR31_C0_CARD_VERIFICATION_KEY.

IMK

Una clave maestra del emisor (IMK) es una clave maestra que se utiliza como parte de la personalización de la tarjeta con chip EMV. Normalmente, habrá 3 IMK: una para las claves AC (criptograma), SMI (clave maestra de script para) y SMC (clave maestra de script para integrity/ signature). confidentiality/encryption

IK

Una clave inicial (IK) es la primera clave que se utiliza en el proceso DUKPT y se deriva de la clave de derivación básica (BDK). No se procesa ninguna transacción en esta clave, pero se usa para derivar claves futuras que se usarán para las transacciones. El método de derivación para crear una IK se definió en. X9.24-1:2017 Cuando se utiliza un BDK TDES, X9.24-1:2009 es el estándar aplicable y el IK se sustituye por la clave de cifrado con PIN inicial (IPEK).

IPEK

Una clave de cifrado de PIN inicial (IPEK) es la clave inicial que se utiliza en el proceso DUKPT y se deriva de la clave de derivación básica (BDK). No se procesa ninguna transacción en esta clave, pero se usa para derivar claves futuras que se usarán para las transacciones. IPEK es un nombre inapropiado, ya que esta clave también se puede utilizar para derivar claves de cifrado de datos y de Mac. El método de derivación para crear un IPEK se definió en. X9.24-1:2009 Cuando se utiliza un BDK de AES, X9.24-1:2017 es el estándar aplicable y el IPEK se sustituye por la clave inicial (IK).

IWK

Una clave de trabajo del emisor (IWK) es una clave que se utiliza normalmente para intercambiar datos entre un issuer/issuer procesador y una red (como Visa o Mastercard). Históricamente, IWK utiliza el 3DES para el cifrado y se representa como TR31_P0_PIN_ENCRYPTION_KEY.

KBPK

Una clave de protección de bloques de teclas (KBPK) es un tipo de clave simétrica que se utiliza para proteger los bloques de teclas y, por lo tanto, otras claves. wrap/encrypt Una KBPK es similar a una KEK, pero una KEK protege directamente el material de la clave, mientras que en TR-31 esquemas similares, la KBPK solo protege indirectamente la clave de trabajo. Cuando se utiliza TR-31, TR31_K1_KEY_BLOCK_PROTECTION_KEY es el tipo de clave correcto, aunque TR31_K0_KEY_ENCRYPTION_KEY se admite indistintamente con fines históricos.

KEK

Una clave de cifrado clave (KEK) es una clave que se utiliza para cifrar otras claves, ya sea para su transmisión o almacenamiento. Las claves destinadas a proteger otras claves suelen tener el valor TR31_K0_KEY_ENCRYPTION_KEY según el estándar. KeyUsage *TR-31*

PEK

Una clave de cifrado de PIN (PEK) es un tipo de clave funcional que se utiliza para cifrar los PIN, ya sea para su almacenamiento o transmisión entre dos partes. IWK y AWK son dos ejemplos de usos específicos de las claves de cifrado de PIN. Estas claves se representan como TR31_P0_PIN_ENCRYPTION_KEY.

PGK

PGK (clave de generación de PIN) es otro nombre para una clave de verificación de PIN. En realidad, no se usa para generar pines (que por defecto son números criptográficamente aleatorios), sino que se usa para generar valores de verificación como el PVV.

PRK

La clave de la región principal es la fuente de replicación autorizada de una clave de criptografía de pago determinada para la que se ha activado la replicación. PRK es una referencia a la función clave de la criptografía de pagos de origen en una Multi-Region configuración de replicación de claves. Cuando la replicación está habilitada en una clave de criptografía de pago, se denomina PRK para esa configuración de replicación de claves específica.

PVK

Una clave de verificación de PIN (PVK) es un tipo de clave de trabajo que se utiliza para generar valores de verificación de PIN, como la PVV. Los dos tipos más comunes son el TR31_V1_IBM3624_PIN_VERIFICATION_KEY, que se usa para generar valores de compensación del IBM3624, y el TR31_V2_VISA_PIN_VERIFICATION_KEY, que se usa para los valores de verificación. Visa/ABA También se conoce como clave de generación de pines.

KRK

Las claves de región de réplica son el material clave y los metadatos replicados que se copian de forma segura desde la PRK a una réplica configurada. Región de AWS Una RRK es una réplica de solo lectura de una clave de criptografía de pago. La RRK es una referencia, el papel que desempeña una clave específica en una configuración de replicación de Multi-Region claves. Todos los cambios clave en los metadatos, incluida la configuración de replicación, deben aplicarse a la PRK.

Otros términos

ARQC

El criptograma de solicitud de autorización (ARQC) es un criptograma generado en el momento de la transacción mediante una tarjeta con chip estándar EMV (o una implementación sin contacto equivalente). Por lo general, un ARQC se genera mediante una tarjeta con chip y se envía al emisor o a su agente para su verificación en el momento de la transacción.

CVV

El valor de verificación de una tarjeta es un valor secreto estático que, tradicionalmente, estaba incrustado en una banda magnética y se utilizaba para validar la autenticidad de una transacción. El algoritmo también se utiliza para otros fines, como iCVV, CAVV, CVV2. Es posible que no esté integrado de esta manera para otros casos de uso.

CVV2

El valor de verificación de una tarjeta 2 es un valor secreto estático que tradicionalmente se imprimía en el anverso (o reverso) de una tarjeta de pago y que se utiliza para verificar la autenticidad de los pagos con tarjetas no presentes (por ejemplo, por teléfono o en línea). Utiliza el mismo algoritmo que el CVV, pero el código de servicio está establecido en 000.

iCVV

iCVV es un CVV2-like valor pero está integrado con los datos equivalentes a track2 en una tarjeta EMV (chip). Este valor se calcula con un código de servicio 999 y es diferente al utilizado CVV1/CVV2 para evitar que la información robada se utilice para crear nuevas credenciales de pago de otro tipo. Por ejemplo, si se obtuvieron datos de transacciones con chips, no es posible utilizarlos para generar una banda magnética (CVV1) ni para realizar compras en línea (CVV2).

Utiliza una clave ???

DUKPT

La clave única derivada por transacción (DUKPT) es un estándar de administración de claves que se suele utilizar para definir el uso de claves de cifrado físicas de un solo uso. POS/POI Históricamente, DUKPT utiliza el 3DES para el cifrado. El estándar industrial para el DUKPT se define en el ANSI. X9.24-3-2017

ECC

El ECC (criptografía de curva elíptica) es un sistema de criptografía de clave pública que utiliza las matemáticas de las curvas elípticas para crear claves de cifrado. El ECC proporciona el

mismo nivel de seguridad que los métodos tradicionales, como el RSA, pero con longitudes de clave mucho más cortas, lo que proporciona una seguridad equivalente de una manera más eficiente. Esto es especialmente relevante para los casos de uso en los que RSA no es una solución práctica (longitud de clave RSA superior a 4096 bits). AWS La criptografía de pagos admite curvas definidas por el NIST para su uso en las operaciones del ECDH.

ECDH

El ECDH (curva elíptica Diffie-Hellman) es un protocolo de acuerdo clave que permite a dos partes establecer un secreto compartido (como un KEK o un PEK). En el ECDH, las Partes A y B tienen sus propios pares de claves público-privadas e intercambian claves públicas entre sí (en forma de certificados de criptografía de AWS pago), así como metadatos de derivación de claves (método de derivación, tipo de hash e información compartida). Ambas partes multiplican su clave privada por la clave pública de la otra y, gracias a las propiedades de la curva elíptica, ambas partes pueden derivar (generar) la clave resultante.

EMV

EMV (originalmente Europay, Mastercard y Visa) es un organismo técnico que trabaja con las partes interesadas en los pagos para crear estándares y tecnologías de pago interoperables. Un ejemplo de norma es el de chip/contactless las tarjetas y los terminales de pago con los que interactúan, incluida la criptografía utilizada. La derivación de claves EMV se refiere a los métodos que permiten generar claves únicas para cada tarjeta de pago a partir de un conjunto inicial de claves, como una `IMK`

HSM

Un módulo de seguridad de hardware (HSM) es un dispositivo físico que protege las operaciones criptográficas (por ejemplo, el cifrado, el descifrado y las firmas digitales), así como las claves subyacentes que se utilizan para estas operaciones.

KCAAS

Un custodio de claves como servicio (KCAAS) proporciona una variedad de servicios relacionados con la administración de claves. En el caso de las claves de pago, normalmente pueden convertir los componentes clave en papel en formularios electrónicos compatibles con la criptografía de AWS pago o convertir las claves protegidas electrónicamente en componentes en papel que podrían necesitar algunos proveedores. También pueden ofrecer servicios de custodia de llaves para las llaves cuya pérdida sería perjudicial para sus operaciones en curso. Los proveedores de KCAAS pueden ayudar a los clientes a reducir la carga operativa que supone gestionar el material clave fuera de un servicio seguro, como la criptografía de AWS pagos, de una forma que cumpla con los estándares PCI DSS, PCI PIN y PCI P2PE. AWS La criptografía

de pagos ofrece Intercambio de claves físicas una capacidad KCAAS integrada para convertir componentes clave en papel a formato electrónico.

KCV

El valor de comprobación de claves (KCV) se refiere a una variedad de métodos de suma de comprobación que se utilizan principalmente para comparar claves entre sí sin tener acceso al material de las claves propiamente dichas. Los KCV también se han utilizado para validar la integridad (especialmente cuando se intercambian claves), aunque esta función ahora se incluye como parte de los formatos de bloques de claves, como TR-31. En el caso de las claves TDES, el KCV se calcula cifrando 8 bytes, cada uno con un valor igual a cero, con la clave que hay que comprobar y reteniendo los 3 bytes más importantes del resultado cifrado. En el caso de las claves AES, el KCV se calcula mediante un algoritmo CMAC en el que los datos de entrada son 16 bytes de cero y se retienen los 3 bytes de orden superior del resultado cifrado. En el caso de las claves HMAC, el KCV se calcula utilizando el hash seleccionado al crear la clave en un mensaje de longitud cero, ocupando los 3 bytes más a la izquierda.

KDH

Un host de distribución de claves (KDH) es un dispositivo o sistema que envía claves en un proceso de intercambio de claves, como. TR-34 Cuando se envían claves desde AWS Payment Cryptography, se considera el KDH.

KIF

Un servicio de inyección de claves (KIF) es un servicio seguro que se utiliza para inicializar los terminales de pago e incluso cargarlos con claves de cifrado.

KRD

Un dispositivo receptor de claves (KRD) es un dispositivo que recibe claves en un proceso de intercambio de claves, como. TR-34 Al enviar claves a la criptografía de AWS pagos, se considera el KRD.

KSN

Un número de serie clave (KSN) es un valor que se utiliza como entrada en DUKPT encryption/decryption para crear claves de cifrado únicas por transacción. Por lo general, el KSN consta de un identificador BDK, un identificador de terminal semi-exclusivo y un contador de transacciones que se incrementa con cada transición procesada en un terminal de pago determinado. Por ejemplo X9.24, en el caso del TDES, el KSN de 10 bytes suele constar de 24 bits para el ID del conjunto de claves, 19 bits para el ID del terminal y 21 bits para el contador de transacciones, aunque el límite entre el ID del conjunto de claves y el ID del terminal no afecta a la función de

la criptografía de pagos. AWS En el caso del AES, el KSN de 12 bytes suele constar de 32 bits para el ID del BDK, 32 bits para el identificador de derivación (ID) y 32 bits para el contador de transacciones.

mPoC

El mPoC (punto de venta móvil con hardware comercial) es un estándar PCI que aborda los requisitos de seguridad de las soluciones que permiten a los comerciantes aceptar PIN de los titulares de tarjetas o pagos sin contacto mediante un teléfono inteligente u otros dispositivos móviles comerciales listos para usar (COTS).

PAN

El número de cuenta principal (PAN) es un identificador único para una cuenta, como una tarjeta de crédito o débito. Suele tener entre 13 y 19 dígitos. Los primeros 6 a 8 dígitos identifican la red y el banco emisor.

Bloqueo de PIN

Un bloque de datos que contiene un PIN durante el procesamiento o la transmisión, así como otros elementos de datos. Los formatos de bloque de PIN estandarizan el contenido del bloque de PIN y la forma en que se puede procesar para recuperar el PIN. La mayoría de los bloques de PIN están compuestos por el PIN, la longitud del PIN y, con frecuencia, contienen parte o todo el PAN. AWS La criptografía de pagos es compatible con los formatos ISO 9564-1 0, 1, 3 y 4. El formato 4 es obligatorio para las claves AES. Al verificar o traducir los PIN, es necesario especificar el bloque de PIN de los datos entrantes o salientes.

POI

El punto de interacción (POI), que también se utiliza con frecuencia de forma anónima con el punto de venta (POS), es el dispositivo de hardware con el que el titular de la tarjeta interactúa para presentar su credencial de pago. Un ejemplo de POI es la terminal física de un establecimiento comercial. Para ver la lista de terminales POI PCI PTS certificados, consulte el sitio web de PCI.

PSN

El número de secuencia PAN (PSN) es un valor numérico que se utiliza para diferenciar varias tarjetas emitidas con el mismo PAN.

Clave pública

Cuando se utilizan cifrados asimétricos (RSA, ECC), la clave pública es el componente público de un par de claves público-privadas. La clave pública se puede compartir y distribuir a entidades

que necesitan cifrar datos para el propietario del par de claves público-privado. Para las operaciones de firma digital, la clave pública se utiliza a fin de verificar la firma.

Clave privada

Cuando se utilizan cifrados asimétricos (RSA, ECC), la clave privada es el componente privado de un par de claves público-privadas. La clave privada se utiliza para descifrar los datos o crear firmas digitales. Al igual que las claves simétricas de criptografía AWS de pagos, los HSM crean las claves privadas de forma segura. Solo se descifran en la memoria volátil del HSM y únicamente durante el tiempo necesario para procesar su solicitud criptográfica.

PVV

Un valor de verificación de PIN (PVV) es un tipo de salida criptográfica que se puede utilizar para verificar un PIN sin almacenar el pin real. Aunque es un término genérico, en el contexto de la criptografía de AWS pagos, PVV se refiere al método PVV de Visa o ABA. Este PVV es un número de cuatro dígitos cuyas entradas son el número de la tarjeta, el número de secuencia panorámica, la propia bandeja y una clave de verificación del PIN. Durante la fase de validación, AWS Payment Cryptography recrea internamente el PVV utilizando los datos de la transacción y lo compara de nuevo con el valor almacenado por el AWS cliente de Payment Cryptography. En este sentido, es conceptualmente similar a un hash criptográfico o MAC.

RSA Wrap/Unwrap

La envoltura RSA utiliza una clave asimétrica para envolver una clave simétrica (como una clave TDES) para su transmisión a otro sistema. Solo el sistema con la clave privada coincidente puede descifrar la carga útil y cargar la clave simétrica. Por el contrario, RSA unwrap descifrará de forma segura una clave cifrada con RSA y, a continuación, la cargará en la criptografía de pagos. AWS El empaquetado RSA es un método de bajo nivel para intercambiar claves y no transmite las claves en formato de bloque de claves ni utiliza la firma de carga útil por parte de la parte que las envía. Se deben considerar controles alternativos para determinar la procedencia y comprobar que los atributos clave no están mutados.

TR-34 también utiliza RSA internamente, pero es un formato independiente y no es interoperable.

Código de servicio

Un código de servicio es un valor de tres dígitos codificado en la banda magnética y el chip de una tarjeta de pago que define los atributos de la tarjeta, como dónde se puede usar la tarjeta y si se requiere una autorización adicional. El código de servicio se utiliza como entrada al calcular los valores de verificación. Los diferentes valores de los códigos de servicio producen diferentes tipos

de valores de verificación, como el CVV (que utiliza el código de servicio de la banda magnética de la tarjeta), el iCVV (código de servicio 999 para las tarjetas con chip) y el CVV2 (código de servicio 000).

TR-31

TR-31 (definido formalmente como ANSI X9 TR 31) es un formato de bloques de teclas definido por el Instituto Nacional de Normalización de los Estados Unidos (ANSI) para permitir la definición de los atributos clave en la misma estructura de datos que los propios datos clave. El formato de bloque de TR-31 teclas define un conjunto de atributos clave que están vinculados a la clave para que se mantengan unidos. AWS La criptografía de pagos utiliza términos TR-31 estandarizados siempre que es posible para garantizar una separación y un propósito adecuados de las claves. TR-31 ha sido sustituida por el ANSI. X9.143-2022

TR-34

TR-34 es una implementación del ANSI X9.24-2 que describe un protocolo para distribuir de forma segura claves simétricas (como 3DES y AES) mediante técnicas asimétricas (como RSA). AWS La criptografía de pagos utiliza TR-34 métodos que permiten la importación y exportación seguras de claves.

X9.143

X9.143 es un formato de bloque de claves definido por el Instituto Nacional de Normalización de los Estados Unidos (ANSI) para proteger una clave y sus atributos en la misma estructura de datos. El formato de bloque de claves define un conjunto de atributos clave que están vinculados a la clave para que se mantengan unidos. AWS La criptografía de pagos utiliza términos X9.143 estandarizados siempre que es posible para garantizar una separación y un propósito adecuados de las claves. X9.143 sustituye a la TR-31propuesta anterior, aunque en la mayoría de los casos son compatibles con versiones anteriores y posteriores y los términos suelen utilizarse indistintamente.

Servicios relacionados

AWS Key Management Service

AWS El Servicio de administración de claves (AWS KMS) es un servicio administrado que le facilita la creación y el control de las claves criptográficas que se utilizan para proteger sus datos. AWS KMS utiliza módulos de seguridad de hardware (HSMs) para proteger y validar las claves de AWS KMS.

AWS CloudHSM

AWS CloudHSM proporciona a los clientes instancias de HSM dedicadas de uso general en la AWS nube. AWS CloudHSM puede proporcionar una variedad de funciones criptográficas, como la creación de claves, la firma de datos o el cifrado y descifrado de datos.

Para obtener más información

- Para obtener más información sobre los términos y conceptos utilizados en la criptografía de AWS pagos, consulte AWS Conceptos de criptografía de pagos.
- Para obtener información sobre la API del plano de control de criptografía de AWS pagos, consulte la referencia de la API del plano de control AWS de criptografía de pagos.
- Para obtener información sobre la API del plano de datos de criptografía de AWS pagos, consulte la referencia de la API del plano de datos AWS de criptografía de pagos.
- Para obtener información técnica detallada sobre cómo la criptografía de AWS pagos utiliza la criptografía y protege las claves de criptografía de AWS pagos, consulte Detalles criptográficos.

Puntos finales para AWS Payment Cryptography

Para conectarse mediante programación AWS Payment Cryptography, utilice un punto final, la URL del punto de entrada al servicio. AWS Los SDK y las herramientas de línea de comandos utilizan automáticamente el punto final predeterminado para el servicio en Región de AWS función del contexto regional de la solicitud, por lo que normalmente no es necesario establecer estos valores de forma explícita. Cuando sea necesario, puedes especificar un punto final diferente para tus solicitudes de API.

Puntos de conexión del plano de control

Nombre de la región	Región	Punto de conexión	Protocolo	
Este de EE. UU. (Ohio)	us-east-2	controlplane.payment-cryptography.us-east-2.amazonaws.com	HTTPS HTTPS	

Nombre de la región	Región	Punto de conexión	Protocolo
		controlplane.payment-cryptography.us-east-2.api.aws	
Este de EE. UU. (Norte de Virginia)	us-east-1	controlplane.payment-cryptography.us-east-1.amazonaws.com controlplane.payment-cryptography.us-east-1.api.aws	HTTPS HTTPS
Oeste de EE. UU. (Oregón)	us-west-2	controlplane.payment-cryptography.us-west-2.amazonaws.com controlplane.payment-cryptography.us-west-2.api.aws	HTTPS HTTPS
África (Ciudad del Cabo)	af-south-1	controlplane.payment-cryptography.af-south-1.amazonaws.com controlplane.payment-cryptography.af-south-1.api.aws	HTTPS HTTPS
Asia-Pacífico (Hyderabad)	ap-south-2	controlplane.payment-cryptography.ap-south-2.amazonaws.com controlplane.payment-cryptography.ap-south-2.api.aws	HTTPS HTTPS
Asia-Pacífico (Mumbai)	ap-south-1	controlplane.payment-cryptography.ap-south-1.amazonaws.com controlplane.payment-cryptography.ap-south-1.api.aws	HTTPS HTTPS

Nombre de la región	Región	Punto de conexión	Protocolo	
Asia-Pacífico (Osaka)	ap-northeast-3	controlplane.payment-cryptography.ap-northeast-3.amazonaws.com controlplane.payment-cryptography.ap-northeast-3.api.aws	HTTPS HTTPS	
Asia-Pacífico (Singapur)	ap-southeast-1	controlplane.payment-cryptography.ap-southeast-1.amazonaws.com controlplane.payment-cryptography.ap-southeast-1.api.aws	HTTPS HTTPS	
Asia-Pacífico (Sídney)	ap-southeast-2	controlplane.payment-cryptography.ap-southeast-2.amazonaws.com controlplane.payment-cryptography.ap-southeast-2.api.aws	HTTPS HTTPS	
Asia-Pacífico (Tokio)	ap-northeast-1	controlplane.payment-cryptography.ap-northeast-1.amazonaws.com controlplane.payment-cryptography.ap-northeast-1.api.aws	HTTPS HTTPS	
Canadá (centro)	ca-central-1	controlplane.payment-cryptography.ca-central-1.amazonaws.com controlplane.payment-cryptography.ca-central-1.api.aws	HTTPS HTTPS	
Europa (Fráncfort)	eu-central-1	controlplane.payment-cryptography.eu-central-1.amazonaws.com controlplane.payment-cryptography.eu-central-1.api.aws	HTTPS HTTPS	

Nombre de la región	Región	Punto de conexión	Protocolo	
Europa (Irlanda)	eu-west-1	controlplane.payment-cryptography.eu-west-1.amazonaws.com controlplane.payment-cryptography.eu-west-1.api.aws	HTTPS HTTPS	
Europa (Londres)	eu-west-2	controlplane.payment-cryptography.eu-west-2.amazonaws.com controlplane.payment-cryptography.eu-west-2.api.aws	HTTPS HTTPS	
Europa (París)	eu-west-3	controlplane.payment-cryptography.eu-west-3.amazonaws.com controlplane.payment-cryptography.eu-west-3.api.aws	HTTPS HTTPS	
América del Sur (São Paulo)	sa-east-1	controlplane.payment-cryptography.sa-east-1.amazonaws.com controlplane.payment-cryptography.sa-east-1.api.aws	HTTPS HTTPS	

Puntos de conexión del plano de datos

Nombre de la región	Región	Punto de conexión	Protocolo	
Este de EE. UU. (Ohio)	us-east-2	dataplane.payment-cryptography.us-east-2.amazonaws.com	HTTPS HTTPS	

Nombre de la región	Región	Punto de conexión	Protocolo
		dataplane.payment-cryptography.us-east-2.api.aws	
Este de EE. UU. (Norte de Virginia)	us-east-1	dataplane.payment-cryptography.us-east-1.amazonaws.com dataplane.payment-cryptography.us-east-1.api.aws	HTTPS HTTPS
Oeste de EE. UU. (Oregón)	us-west-2	dataplane.payment-cryptography.us-west-2.amazonaws.com dataplane.payment-cryptography.us-west-2.api.aws	HTTPS HTTPS
África (Ciudad del Cabo)	af-south-1	dataplane.payment-cryptography.af-south-1.amazonaws.com dataplane.payment-cryptography.af-south-1.api.aws	HTTPS HTTPS
Asia-Pacífico (Hyderabad)	ap-south-2	dataplane.payment-cryptography.ap-south-2.amazonaws.com dataplane.payment-cryptography.ap-south-2.api.aws	HTTPS HTTPS
Asia-Pacífico (Mumbai)	ap-south-1	dataplane.payment-cryptography.ap-south-1.amazonaws.com dataplane.payment-cryptography.ap-south-1.api.aws	HTTPS HTTPS

Nombre de la región	Región	Punto de conexión	Protocolo
Asia-Pacífico (Osaka)	ap-northeast-3	dataplane.payment-cryptography.ap-northeast-3.amazonaws.com dataplane.payment-cryptography.ap-northeast-3.api.aws	HTTPS HTTPS
Asia-Pacífico (Singapur)	ap-southeast-1	dataplane.payment-cryptography.ap-southeast-1.amazonaws.com dataplane.payment-cryptography.ap-southeast-1.api.aws	HTTPS HTTPS
Asia-Pacífico (Sídney)	ap-southeast-2	dataplane.payment-cryptography.ap-southeast-2.amazonaws.com dataplane.payment-cryptography.ap-southeast-2.api.aws	HTTPS HTTPS
Asia-Pacífico (Tokio)	ap-northeast-1	dataplane.payment-cryptography.ap-northeast-1.amazonaws.com dataplane.payment-cryptography.ap-northeast-1.api.aws	HTTPS HTTPS
Canadá (centro)	ca-central-1	dataplane.payment-cryptography.ca-central-1.amazonaws.com dataplane.payment-cryptography.ca-central-1.api.aws	HTTPS HTTPS
Europa (Fráncfort)	eu-central-1	dataplane.payment-cryptography.eu-central-1.amazonaws.com dataplane.payment-cryptography.eu-central-1.api.aws	HTTPS HTTPS

Nombre de la región	Región	Punto de conexión	Protocolo	
Europa (Irlanda)	eu-west-1	dataplane.payment-cryptography.eu-west-1.amazonaws.com dataplane.payment-cryptography.eu-west-1.api.aws	HTTPS HTTPS	
Europa (Londres)	eu-west-2	dataplane.payment-cryptography.eu-west-2.amazonaws.com dataplane.payment-cryptography.eu-west-2.api.aws	HTTPS HTTPS	
Europa (París)	eu-west-3	dataplane.payment-cryptography.eu-west-3.amazonaws.com dataplane.payment-cryptography.eu-west-3.api.aws	HTTPS HTTPS	
América del Sur (São Paulo)	sa-east-1	dataplane.payment-cryptography.sa-east-1.amazonaws.com dataplane.payment-cryptography.sa-east-1.api.aws	HTTPS HTTPS	

Cómo empezar con la criptografía AWS de pagos

Para empezar con la criptografía de AWS pagos, primero querrá crear claves y, después, utilizarlas en diversas operaciones criptográficas. El siguiente tutorial proporciona un caso de uso sencillo para generar una clave que se utilizará para generating/verifying CVV2 los valores. Para probar otros ejemplos y explorar los patrones de implementación en AWS, visite el siguiente taller de criptografía de AWS pagos o explore nuestro proyecto de muestra disponible en GitHub

En este tutorial, se explica cómo crear una clave única y cómo realizar operaciones criptográficas con ella. Después, borra la clave si ya no la desea, completando el ciclo de vida de la clave.

> **Warning**
>
> Los ejemplos de esta guía del usuario pueden utilizar valores de muestra. Recomendamos encarecidamente no utilizar valores de muestra en un entorno de producción, como los números de serie clave.

Temas

- Requisitos previos
- Paso 1: crear una clave
- Paso 2: Genere un valor con la clave CVV2
- Paso 3: verificar el valor generado en el paso 2
- Paso 4: realizar una prueba negativa
- Paso 5: eliminación (opcional)

Requisitos previos

Antes de comenzar, asegúrese de que:

- Tiene permiso para acceder al servicio. Para obtener más información, consulte Políticas de IAM.
- Ha instalado AWS CLI. También puede utilizar AWS SDKso acceder AWS APIsa la criptografía de AWS pagos, pero en las instrucciones de este tutorial se utiliza la AWS CLI.

Paso 1: crear una clave

El primer paso es crear una clave. Para este tutorial, debe crear una clave CVK 3DES de doble longitud (2KEY TDES) para generar y verificar los valores CVV/. CVV2

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY,KeyModesO
```

La respuesta refleja los parámetros de la solicitud, incluyendo un ARN para las llamadas posteriores y un valor de verificación clave (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"
    }
}
```

Tome nota de `KeyArn` que representa la clave, por ejemplo, arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi. Lo necesitará en el siguiente paso.

Paso 2: Genere un valor con la clave CVV2

En este paso, generas una CVV2 para una fecha de caducidad determinada PAN utilizando la clave del paso 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"
}
```

Tome nota de `cardDataValue`, en este caso el número de tres dígitos 144. Lo necesitará en el siguiente paso.

Paso 3: verificar el valor generado en el paso 2

En este ejemplo, validas el dato CVV2 del paso 2 con la clave que creaste en el paso 1.

Ejecute el siguiente comando para validar el 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"
}
```

El servicio devuelve una respuesta HTTP de 200 para indicar que ha validado la CVV2.

Paso 4: realizar una prueba negativa

En este paso, se crea una prueba negativa en la que no CVV2 es correcta ni se valida. Intenta validar un error CVV2 con la clave que creaste en el paso 1. Se trata de una operación esperada, por ejemplo, si el titular de la tarjeta ha introducido un error CVV2 al finalizar la compra.

```
$  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.
```

El servicio devuelve una respuesta HTTP de 400 con el mensaje “Fallo en la verificación de los datos de validación de la tarjeta” y el motivo INVALID_VALIDATION_DATA.

Paso 5: eliminación (opcional)

Ahora puede eliminar la clave que creó en el paso 1. Para minimizar los cambios irrecuperables, el periodo de eliminación de claves predeterminado es de siete de días.

```
$ 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_C0_CARD_VERIFICATION_KEY"
        },
        "KeyCheckValue": "CADDA1",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "DELETE_PENDING",
        "UsageStartTimestamp": "2022-10-27T08:27:51.753000-07:00"
    }
}
```

Tome nota de dos campos en la salida. El `deletePendingTimestamp` se fija por defecto para los siete días posteriores. El KeyState está establecido en `DELETE_PENDING`. Puede cancelar esta eliminación en cualquier momento antes de la hora de eliminación programada llamando al `restore-key`.

Administración de claves

Para empezar a usar la criptografía AWS de pagos, crea una clave de criptografía AWS de pagos.

En esta sección se explica cómo crear y administrar varios tipos de claves de criptografía AWS de pagos a lo largo de su ciclo de vida. Aprenderás a crear, ver y editar claves, así como a etiquetar claves, crear alias de claves y habilitar o deshabilitar claves.

Una clave AWS de criptografía de pagos es un recurso regional. Si tiene la intención de utilizar una clave determinada en varias Regiones de AWS, puede habilitar la replicación de Multi-Region claves, que copia de forma segura el material clave y los metadatos Regiones de AWS que especifique dentro de la misma AWS partición y cuenta. La clave de origen de la replicación de Multi-Region claves se conoce como clave de región principal (PRK) y sigue siendo la fuente autorizada para todas las actividades de administración de claves. La clave replicada se conoce como clave de región de réplica (RRK) y es una réplica de solo lectura de la PRK. Deberías considerar la posibilidad de usar Multi-Region claves junto con tus claves para cumplir los objetivos de diseño relacionados con la disponibilidad, la recuperación ante desastres y la baja latencia.

Temas

- Crear claves
- Enumerar las claves
- Habilitación y desactivación de claves de
- En proceso de replicación AWS Claves de criptografía de pago
- Eliminación de claves de
- Importación y exportación de claves
- Uso de alias
- Obtener claves
- Etiquetado de claves
- Descripción de los atributos clave de AWS Clave de criptografía de pagos

Crear claves

Puedes crear claves de criptografía de AWS pagos mediante la operación de la CreateKey API. Cuando creas una clave, especificas atributos como el algoritmo de la clave, el uso de la clave, las

operaciones permitidas y si es exportable. No puedes cambiar estas propiedades después de crear la clave de criptografía AWS de pagos.

> **Note**
>
> Si tienes habilitada la replicación de Multi-Region claves Cuenta de AWS y creas una clave de criptografía de pagos, esta clave se convertirá automáticamente en una clave de región principal (PRK). PRK se replica aunque no especifiques el `--replication-regions` parámetro en el comando. CreateKey Para obtener más información, consulte Cómo funciona la Multi-Region replicación clave.

Ejemplos

- Creación de una clave de derivación base TDES de 2KEY
- Crear una clave TDES de 2 CLAVES para CVV/CVV2
- Crear una clave HMAC
- Crear una clave AES-256
- Creación de una clave de cifrado PIN (PEK)
- Crear una clave asimétrica (RSA)
- Creación de una clave de valor de verificación del PIN (PVV)
- Creación de una clave ECC asimétrica

Creación de una clave de derivación base TDES de 2KEY

> **Note**
>
> Aunque 3KEY TDES es compatible con KeyUsage = TR31_B0_BASE_DERIVATION_KEY, actualmente ninguna operación admite este tipo de clave. El X9.24 estándar ANSI (DUKPT) define la compatibilidad únicamente con los tipos de claves TDES_2KEY y AES. ???

Example

Este comando crea una clave de derivación TDES 2KEY para DUKPT que se replicará desde las regiones del oeste de los Estados Unidos (Oregón) hasta las del este de los EE. UU. (Ohio). La

respuesta incluye los parámetros de la solicitud, un nombre de recurso de Amazon (ARN) para las llamadas posteriores y un valor de verificación de clave (KCV).

```
$ aws payment-cryptography create-key --exportable --key-attributes \
     "KeyUsage=TR31_B0_BASE_DERIVATION_KEY, \
     KeyClass=SYMMETRIC_KEY,KeyAlgorithm=TDES_2KEY, \
     KeyModesOfUse={DeriveKey=true}" \
     --replication-regions us-east-2 --region us-west-2
```

Ejemplo de código de salida:

```
{
    "Key": {
        "CreateTimestamp": "2022-10-26T16:04:11.642000-07:00",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/5rplquuwozodpwsp",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_2KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": false,
                "DeriveKey": true,
                "Encrypt": false,
                "Generate": false,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": false,
                "Verify": false,
                "Wrap": false
            },
            "KeyUsage": "TR31_B0_BASE_DERIVATION_KEY"
        },
        "KeyCheckValue": "FE23D3",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "2022-10-26T16:04:11.559000-07:00"
}
```

Crear una clave TDES de 2 CLAVES para CVV/CVV2

Example

Este comando crea una clave TDES de 2 teclas para generar y verificar valores. CVV/CVV2 La respuesta incluye los parámetros de la solicitud, un nombre de recurso de Amazon (ARN) para las llamadas posteriores y un valor de verificación de clave (KCV).

```
$ 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}'
```

Ejemplo de código de salida:

```
{
    "Key": {
        "CreateTimestamp": "2022-10-26T16:04:11.642000-07:00",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/7f7g4spf3xcklhzu",
        "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": "AEA5CD",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "2022-10-26T16:04:11.559000-07:00"
    }
}
```

Crear una clave HMAC

Example

Las claves HMAC se utilizan para generar o verificar los códigos de autenticación de mensajes hash (HMAC). En el caso de las claves HMAC, el tipo de hash se asigna en el momento de la creación de la clave (como HMAC_SHA224 y HMAC_SHA512) y no se puede modificar.

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=HMAC_SHA512,KeyUsage=TR31_M7_HMAC_KEY,KeyClass=SYMMETRIC_KEY,KeyModesOfUse='{Gene
 = true,Verify = true}'
```

Ejemplo de código de salida:

```
{
 "Key": {
 "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/qnobl5lghrzunce6",
 "KeyAttributes": {
 "KeyUsage": "TR31_M7_HMAC_KEY",
 "KeyClass": "SYMMETRIC_KEY",
 "KeyAlgorithm": "HMAC_SHA512",
 "KeyModesOfUse": {
 "Encrypt": false,
 "Decrypt": false,
 "Wrap": false,
 "Unwrap": false,
 "Generate": true,
 "Sign": false,
 "Verify": true,
 "DeriveKey": false,
 "NoRestrictions": false
 }
 },
 "KeyCheckValue": "2976E7",
 "KeyCheckValueAlgorithm": "HMAC",
 "Enabled": true,
 "Exportable": true,
 "KeyState": "CREATE_COMPLETE",
 "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
 "CreateTimestamp": "2025-07-30T10:06:12.142000-07:00",
 "UsageStartTimestamp": "2025-07-30T10:06:12.128000-07:00"
 }
}
```

Crear una clave AES-256

Example

Este comando crea una clave AES-256 simétrica para el cifrado y el descifrado de datos. Las claves AES proporcionan un cifrado seguro de los datos confidenciales y se suelen utilizar en el procesamiento de pagos para cifrar los datos de los titulares de tarjetas y otra información confidencial; sin embargo, las TDES se utilizan más comúnmente para casos de uso de emisores, como EMV.

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=AES_256,KeyUsage=TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY,KeyClass=SYMMETRIC_KEY,Key
```

Ejemplo de código de salida:

```
{
    "Key": {
        "CreateTimestamp": "2025-02-02T10:15:30.142000-08:00",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-1:111122223333:key/
kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyAlgorithm": "AES_256",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": true,
                "DeriveKey": false,
                "Encrypt": true,
                "Generate": false,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": true,
                "Verify": false,
                "Wrap": true
            },
            "KeyUsage": "TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY"
        },
        "KeyCheckValue": "2976F5",
        "KeyCheckValueAlgorithm": "CMAC",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "2025-02-02T10:15:30.128000-08:00"
    }
}
```

Creación de una clave de cifrado PIN (PEK)

Example

Este comando crea una clave TDES de 3 claves para cifrar los valores PIN, aunque las claves PIN también pueden ser AES en función de sus necesidades de interoperabilidad. Puede usar esta clave para almacenar los PIN de forma segura o descifrarlos durante la verificación, por ejemplo, en una transacción. La respuesta incluye los parámetros de la solicitud, un ARN para las llamadas posteriores y un KCV.

```
$ 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}'
```

Ejemplo de código de salida:

```
{
    "Key": {
        "CreateTimestamp": "2022-10-27T08:27:51.795000-07:00",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt",
        "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": "7CC9E2",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "2022-10-27T08:27:51.753000-07:00"
    }
}
```

Crear una clave asimétrica (RSA)

Example

Este comando genera un nuevo par de claves RSA asimétricas de 2048 bits. Crea una nueva clave privada y su clave pública correspondiente. Puedes recuperar la clave pública mediante la PublicCertificate API get.

```
$ 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}'
```

Ejemplo de código de salida:

```
{
    "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": "SHA-1",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "2022-11-15T11:15:42.182000-08:00"
    }
}
```

Creación de una clave de valor de verificación del PIN (PVV)

Example

Este comando crea una clave TDES de 3 teclas para generar valores PVV. Puede usar esta clave para generar un PVV que pueda compararse con un PVV calculado posteriormente. La respuesta incluye los parámetros de la solicitud, un ARN para las llamadas posteriores y un KCV.

```
$ 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}'
```

Ejemplo de código de salida:

```
{
    "Key": {
        "CreateTimestamp": "2022-10-27T10:22:59.668000-07:00",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/37y2tsl45p5zjbh2",
        "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": "7F2363",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "2022-10-27T10:22:59.614000-07:00"
    }
}
```

Creación de una clave ECC asimétrica

Example

Este comando genera un par de claves ECC para establecer un acuerdo de claves ECDH (curva elíptica Diffie-Hellman) entre dos partes. Con el ECDH, cada parte genera su propio par de claves ECC con el propósito clave K3 y el modo de uso X, e intercambian claves públicas. Luego, ambas partes usan su clave privada y la clave pública recibida para establecer una clave derivada compartida.

Para mantener el principio de uso único de las claves criptográficas en los pagos, recomendamos no reutilizar los pares de claves ECC para varios fines, como la derivación y la firma de claves ECDH.

```
$ aws payment-cryptography create-key --exportable \
    --key-attributes
 KeyAlgorithm=ECC_NIST_P256,KeyUsage=TR31_K3_ASYMMETRIC_KEY_FOR_KEY_AGREEMENT, \
    KeyClass=ASYMMETRIC_KEY_PAIR,KeyModesOfUse='{DeriveKey=true}'
```

Ejemplo de código de salida:

```
{
    "Key": {
        "CreateTimestamp": "2024-10-17T01:31:55.908000+00:00",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
wc3rjsssguhxtilv",
        "KeyAttributes": {
            "KeyAlgorithm": "ECC_NIST_P256",
            "KeyClass": "ASYMMETRIC_KEY_PAIR",
            "KeyModesOfUse": {
                "Decrypt": false,
                "DeriveKey": true,
                "Encrypt": false,
                "Generate": false,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": false,
                "Verify": false,
                "Wrap": false
            },
            "KeyUsage": "TR31_K3_ASYMMETRIC_KEY_FOR_KEY_AGREEMENT"
        },
        "KeyCheckValue": "7E34F19F",
        "KeyCheckValueAlgorithm": "SHA-1",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "2024-10-17T01:31:55.866000+00:00"
    }
}
```

Enumerar las claves

Usa la ListKeys operación para obtener una lista de las claves a las que puedes acceder en tu cuenta y región.

Example

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

Ejemplo de código de salida:

```
{
    "Keys": [
        {
            "CreateTimestamp": "2022-10-12T10:58:28.920000-07:00",
            "Enabled": false,
            "Exportable": true,
            "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/37y2tsl45p5zjbh2",
            "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": "7F2363",
            "KeyCheckValueAlgorithm": "ANSI_X9_24",
            "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
            "KeyState": "CREATE_COMPLETE",
            "UsageStopTimestamp": "2022-10-27T14:19:42.488000-07:00"
        }
    ]
}
```

Habilitación y desactivación de claves de

Puedes desactivar y volver a activar las claves de criptografía AWS de pagos. Al crear una clave, se habilita de forma predeterminada. Si deshabilitas una clave, no se podrá usar en ninguna operación criptográfica hasta que la vuelvas a habilitar. Start/stop los comandos de uso tienen efecto inmediato, por lo que se recomienda revisar su uso antes de realizar dicho cambio. También puede establecer un cambio (iniciar o detener el uso) para que surta efecto en el futuro utilizando el parámetro `timestamp` opcional.

Como es temporal y se deshace fácilmente, deshabilitar una clave de criptografía de AWS pagos es una alternativa más segura que eliminar una clave de criptografía de AWS pagos, una acción que es destructiva e irreversible. Si estás pensando en eliminar una clave de criptografía de AWS pagos, desactívala primero y asegúrate de que no necesitarás usarla para cifrar o descifrar datos en el futuro.

Temas

- Iniciar el uso de claves
- Detener el uso de claves

Iniciar el uso de claves

El uso de claves debe estar habilitado para poder utilizar una clave para operaciones criptográficas. Si una clave no está habilitada, puede utilizar esta operación para hacerla utilizable. El campo `UsageStartTimestamp` representará cuándo se activará la clave became/will . Esto será en el pasado para un token habilitado, y en el futuro si está pendiente de activación.

Example

En este ejemplo, se solicita la habilitación de una llave para su uso. La respuesta incluye la información de la llave y el indicador de habilitación ha pasado a verdadero. Esto también se reflejará en el objeto de respuesta lista-claves.

```
$ 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"
  }
}
```

Detener el uso de claves

Si ya no tiene previsto utilizar una clave, puede detener el uso de la clave para evitar que se realicen más operaciones criptográficas. Esta operación no es permanente, por lo que puede revertirla utilizando iniciar uso de clave. También puede configurar una clave para que se desactive en el futuro. El campo `UsageStopTimestamp` representará el momento en que la clave became/will se desactive.

Example

En este ejemplo, se solicita detener el uso de la llave en el futuro. Tras la ejecución, esta clave no podrá utilizarse para operaciones criptográficas a menos que se vuelva a habilitar mediante iniciar uso de clave. La respuesta incluye la información de la clave y que el indicador de habilitación haya pasado a falso. Esto también se reflejará en el objeto de respuesta lista-claves.

```
$ 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"
  }
}
```

En proceso de replicación AWS Claves de criptografía de pago

AWS La criptografía de pagos admite la replicación de Multi-Region claves, lo que te permite distribuir de forma segura el material clave y los metadatos de cualquier clave de criptografía de AWS pagos a una o más Regiones de AWS dentro de la misma AWS partición y cuenta.

La clave de origen se conoce como clave de región principal (PRK) y sigue siendo la fuente autorizada para todas las actividades de administración de claves, mientras que tanto la clave PRK como la clave de región de réplica (RRK) se pueden usar para operaciones criptográficas en sus respectivos países. Regiones de AWS

Ventajas de la replicación de claves Multi-Region

A continuación se describen algunos beneficios de la replicación Multi-Region clave.

- Configuración más sencilla para aplicaciones de alta disponibilidad: la criptografía de AWS pagos se encarga automáticamente de la distribución de las claves para que pueda utilizar una clave en varias Regiones de AWS sin necesidad de crear copias desacopladas de una clave determinada.
- Claves de alta disponibilidad y baja latencia: con Multi-Region la replicación de claves, puede acceder a sus claves de forma múltiple, Regiones de AWS lo que hace que tengan una alta disponibilidad, lo que reduce la latencia.
- Durabilidad del material clave: las claves de región de réplica son réplicas de claves completas y se pueden usar independientemente de su clave de región principal en las operaciones criptográficas. Una RRK proporciona una réplica duradera en caso de una pérdida catastrófica de datos de una PRK.

Cómo funciona la Multi-Region replicación clave

Cuando la replicación de Multi-Region claves está habilitada, el servicio de criptografía de AWS pagos utiliza mecanismos de distribución segura de claves para copiar el material clave y los metadatos a la réplica Regiones de AWS que usted especifique. Los cambios en los metadatos de una clave de región principal, como los atributos clave, el estado y la activación, se replican automáticamente en las claves de la región de réplica.

Limitaciones y consideraciones

Las siguientes son algunas de las Multi-Region principales limitaciones y consideraciones de la replicación.

- Las claves de criptografía de pago deben configurarse como exportables para poder utilizar la replicación de Multi-Region claves.
- Debe habilitar esta función para una Región de AWS o para claves de criptografía de pagos específicas.
  - Si esta función está habilitada para una Región de AWS, todas las claves de criptografía de AWS pagos creadas tras la activación se replicarán en la especificada. Región de AWS Las claves creadas en esta región se convertirán en claves de región principales. Las claves existentes en esta región no se replicarán automáticamente. Puede habilitar la replicación de Multi-Region claves para las claves existentes a nivel de clave y Región de AWS a nivel de clave.
  - Cada una Región de AWS puede tener una configuración única de replicación de Multi-Region claves.
  - La configuración de Multi-Region replicación de una clave tiene prioridad sobre la configuración de replicación de la Región de AWS Multi-Region clave.
- Una clave de región de réplica no se puede configurar para replicarse en otra Regiones de AWS.
- Multi-Region La replicación de claves está disponible para las claves de criptografía de pagos simétricas, como el estándar de cifrado triple de datos (3DES), el estándar de cifrado avanzado (AES) y el código de autenticación de Hash-based mensajes (HMAC), así como para las claves de criptografía de pagos asimétricas, como RSA y la criptografía de curva elíptica (ECC).
- Las claves de región de réplica son claves de solo lectura. Todos los cambios en la clave de la región principal se aplicarán a las claves de la región de la réplica.
- Los cambios en la clave de la región principal son, en última instancia, consistentes con las claves de la región de réplica.
- Las claves de criptografía de pago solo se pueden replicar con la misma AWS partición y cuenta.
- Las claves de región réplicas cuentan para tu Cuenta de AWS nivel límite de criptografía AWS de pagos.
- La clave de región principal y la clave de región de réplica utilizan el mismo identificador de clave, lo que permite hacer referencia a ambas claves con el mismo ARN en las políticas de IAM.
- Debe tener `CreateKey` permisos en la réplica Región de AWS para que la replicación se realice correctamente.

Habilitar Multi-Region la replicación de claves

Hay dos maneras de habilitar la replicación de Multi-Region claves para las claves de criptografía de AWS pagos.

1. Región de AWS: Multi-Region la replicación de claves se aplica a todas las claves nuevas que se creen en ella Región de AWS cuando está habilitada. Este método proporciona una replicación uniforme para todas las claves.
2. Claves AWS de criptografía de pago específicas: puede gestionar Multi-Region la replicación de claves individuales, lo que permite un nivel de control más detallado.

Una vez que se habilite la replicación de Multi-Region claves, tus claves de criptografía de pagos se replicarán según Regiones de AWS lo que especifiques.

> **Important**
>
> Multi-Region la replicación de claves no se puede pausar. Las claves se replican automáticamente a las que Regiones de AWS usted especifique una vez que se habilita la replicación. Multi-Region La replicación de claves se puede deshabilitar para una clave específica Región de AWS o para claves de criptografía de pago. Debe eliminarla Región de AWS como región de replicación de la clave de región principal para eliminar la clave de región de réplica.
>
> También puede llamar al comando `StopKeyUsage` API o stop-key-usage CLI de su PRK para detener el uso tanto del PRK como de todos los RRK asociados. No podrás usar estas claves en operaciones criptográficas. El uso del comando `StopKeyUsage` API o stop-key-usage CLI no detendrá la replicación de Multi-Region claves en curso habilitada para su PRK.

Puedes comprobar la configuración de replicación de Multi-Region claves para las claves de criptografía de AWS pagos de una determinada forma Región de AWS llamando al comando de la `GetDefaultKeyReplicationRegions` API o la get-default-key-replication-regions CLI. Las claves en las Región de AWS que se llame a esta acción o comando de la API se convertirán en su PRK.

Utilice los siguientes procedimientos para habilitar la replicación de Multi-Region claves.

For Región de AWS

- Utilice el siguiente comando para habilitar la replicación de Multi-Region claves para una Región de AWS que especifique. En este ejemplo, la replicación de Multi-Region claves está habilitada en EE. UU. Este (Ohio) y EE. UU. Oeste (Oregón). Para usar este comando, sustituya *italicized placeholder text* el comando del ejemplo por su propia información.

```
aws payment-cryptography enable-default-key-replication-regions \
    --replication-regions us-east-2 us-west-2
```

> **Note**
>
> Multi-Region La activación de la replicación de claves para un no Región de AWS cambiará la configuración de replicación de ninguna de las claves de criptografía de AWS pago existentes. Puede habilitar esta función para las claves existentes a nivel de clave. Solo las claves creadas después de Multi-Region habilitar la replicación de claves Región de AWS utilizarán la configuración de replicación de la región.

For specific AWS Payment Cryptography keys

- Utilice el siguiente comando para habilitar la replicación de Multi-Region claves de criptografía de pagos específicas. En este ejemplo, la replicación de Multi-Region claves está habilitada en el este de EE. UU. (Ohio). Para usar este comando, sustituya *italicized placeholder text* el comando del ejemplo por su propia información.

```
aws payment-cryptography add-key-replication-regions \
    --key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/kwapwa6qaifllw2h \
    --replication-regions us-east-2
```

Si lo prefiere, puede crear una nueva clave de criptografía de pagos con esta función habilitada; para ello, incluya la replicación Regiones de AWS en la solicitud de creación de la clave.

> **Note**
>
> La configuración de replicación clave tiene prioridad sobre la configuración de Región de AWS replicación.

Deshabilitar Multi-Region la replicación de claves

Si desea deshabilitar la replicación de Multi-Region claves, puede llamar a los comandos disable-default-key-replication o a la remove-key-replication-regions CLI, según cómo esté habilitada la replicación de Multi-Region claves. Tendrá que especificar el ARN de la clave y el Región de AWS para deshabilitar la replicación de Multi-Region claves.

Consideraciones

En última instancia, las eliminaciones de claves de la región de replicación son consistentes.

Puede comprobar la configuración de replicación de Multi-Region claves para las claves de criptografía de AWS pagos de una determinada criptomoneda Región de AWS mediante el comando de la `GetDefaultKeyReplicationRegions` API o la get-default-key-replication-regions CLI.

Utilice los siguientes procedimientos para deshabilitar la replicación de Multi-Region claves.

For Región de AWS

- Use el siguiente comando para deshabilitar la replicación de Multi-Region claves para una Región de AWS que usted especifique. En este ejemplo, la replicación de Multi-Region claves está deshabilitada en el este de EE. UU. (Ohio). Para usar este comando, sustituya *italicized placeholder text* el comando del ejemplo por su propia información.

```
aws payment-cryptography disable-default-key-replication-regions \
    --replication-regions us-east-2
```

For specific AWS Payment Cryptography keys

- Utilice el siguiente comando para deshabilitar la replicación de Multi-Region claves para una clave de criptografía de pagos específica. En este ejemplo, la replicación de Multi-Region claves está deshabilitada en el este de EE. UU. (Ohio). Para usar este comando, sustituya *italicized placeholder text* el comando del ejemplo por su propia información.

```
aws payment-cryptography remove-key-replication-regions \
    --key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/kwapwa6qaifllw2h \
    --replication-regions us-east-2
```

Consideraciones de seguridad

A continuación se indican las consideraciones de seguridad que se deben tener en cuenta al utilizar la replicación de Multi-Region claves para las claves de criptografía de pagos. Para obtener más información, consulte Prácticas recomendadas de seguridad para AWS Payment Cryptography.

- Limite el uso compartido de materiales clave.
- Siga el principio de mínimos privilegios y permisos al crear políticas de IAM.
- No puede realizar cambios en la clave de región de la réplica, ya que es una clave de solo lectura.

Prácticas recomendadas

Las siguientes son algunas de las prácticas recomendadas para usar la replicación de claves con Multi-Region claves de criptografía AWS de pagos.

- Asegúrese de que la aplicación siga funcionando aunque la replicación de Multi-Region claves a la especificada no Región de AWS sea inmediata. Si necesita saber cuándo se ha completado la replicación de Multi-Region claves, puede supervisarla con la acción de la GetKey API. Puede supervisar los eventos clave de replicación con AWS CloudTrail.
- Pruebe e implemente procesos de implementación automatizados en caso de que se produzca una conmutación por error de una región Región de AWS a otra.

Precios

Se le cobrará por las réplicas de claves de región que cree con AWS Payment Cryptography. Estas claves se cobran por. Región de AWS Para obtener la información más reciente sobre los precios de Payment Cryptography, consulta la página de precios AWS de Payment Cryptography.

Eliminación de claves de

La eliminación de una clave de criptografía de AWS pagos elimina el material clave y todos los metadatos asociados a la clave y es irreversible a menos que haya una copia de la clave disponible fuera de la criptografía de AWS pagos. Una vez que se elimina una clave, ya no pueden descifrar los datos que se habían cifrado con ella, lo que significa que los datos pueden volverse irrecuperables. Sólo debe eliminar una clave cuando esté seguro de que ya no necesita utilizarla y de que no hay terceros que la estén utilizando. Si no estás seguro, considera la posibilidad de detener el uso de la clave en lugar de eliminarla. Puedes volver a habilitar una clave desactivada si necesitas volver a utilizarla más adelante, pero no puedes recuperar una clave de criptografía de AWS pagos eliminada a menos que puedas volver a importarla desde otra fuente.

Antes de eliminar una clave, asegúrate de que ya no la necesitas. AWS La criptografía de pagos no almacena los resultados de operaciones criptográficas como la CVV2 y no puede determinar si se necesita una clave para cualquier material criptográfico persistente.

AWS La criptografía de pagos nunca elimina las claves que pertenecen a las AWS cuentas activas, a menos que programes explícitamente su eliminación y venza el período de espera obligatorio.

Sin embargo, puedes optar por eliminar una clave de criptografía de AWS pagos por uno o varios de los siguientes motivos:

- Para completar el ciclo de vida de una clave que ya no necesita
- Para evitar los gastos de administración asociados con el mantenimiento de las claves de criptografía AWS de pagos no utilizadas

> **Note**
>
> Si cierras o eliminas tu Cuenta de AWS clave de criptografía AWS de pagos se vuelve inaccesible. No necesitas programar la eliminación de tu clave de criptografía de AWS pagos sin tener que cerrar la cuenta.

AWS La criptografía de pagos registra una entrada en su AWS CloudTrail registro cuando programa la eliminación de la clave de criptografía de AWS pagos y cuando la clave de criptografía de AWS pagos se elimina realmente.

Al utilizar la replicación de claves y eliminar una Multi-Region clave de criptografía de pago que sea una clave de región principal (PRK), las claves de región de réplica (RRK) también se eliminarán automáticamente. Un RRK no se puede eliminar como un PRK. Si desea eliminar un RRK, tendrá que modificar las regiones de replicación de su PRK.

Acerca del período de espera

Como la eliminación de una clave es irreversible, la criptografía de AWS pagos requiere que establezcas un período de espera de entre 3 y 180 días. El periodo de espera predeterminado es de siete días.

Sin embargo, el período de espera real puede ser hasta 24 horas más largo que el programado. Para obtener la fecha y la hora reales en las que se eliminará la clave de criptografía de AWS pagos, utilice las operaciones. GetKey Asegúrese de anotar la zona horaria.

Durante el período de espera, el estado y el estado de la clave de criptografía de AWS pagos son Pendiente de eliminación.

> **Note**
>
> Una clave AWS de criptografía de pago pendiente de eliminación no se puede usar en ninguna operación criptográfica.

Una vez finalizado el período de espera, la criptografía de AWS pagos elimina la clave de criptografía de AWS pagos, sus alias y todos los metadatos relacionados con la criptografía de pagos. AWS

Usa el período de espera para asegurarte de que no necesitas la clave de criptografía de AWS pagos ahora o en el futuro. Si se da cuenta de que necesita la clave durante el periodo de espera, puede cancelar la eliminación de la clave antes de que finalice el periodo de espera. Una vez que finaliza el periodo de espera, no puede cancelar la eliminación de claves y el servicio elimina la clave.

Example

En este ejemplo, se solicita la eliminación de una clave. Además de la información básica sobre la clave, hay dos campos relevantes que indican que el estado de la clave se ha cambiado a DELETE_PENDING y delete PendingTimestamp representa el momento en el que está programada la eliminación de la clave.

```
$ 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": "0A3674",
        "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

En este ejemplo, se cancela un borrado pendiente. Una vez completada con éxito, la clave ya no se borrará según la programación anterior. La respuesta contiene la información básica de la clave; además, han cambiado dos campos relevantes: `KeyState` y `deletePendingTimestamp`. `KeyState` se devuelve a un valor de CREATE_COMPLETE, mientras que `DeletePendingTimestamp` se elimina.

```
$ 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": "0A3674",
        "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"
    }
}
```

Importación y exportación de claves

Puede importar claves de criptografía de AWS pagos de otras soluciones y exportarlas a otras soluciones, como los HSM. Muchos clientes intercambian claves con los proveedores de servicios mediante la función de importación y exportación. Diseñamos la criptografía de AWS pagos para utilizar un enfoque electrónico moderno para la administración de claves que le ayude a mantener el cumplimiento y los controles. Recomendamos utilizar un intercambio electrónico de claves basado en estándares en lugar de componentes clave en papel. Si necesita seguir procesando los componentes clave en papel hasta que todos los socios admitan el intercambio electrónico de claves, puede usarlo. Intercambio de claves físicas

Los puntos fuertes clave mínimos y el efecto en las funciones de importación y exportación

La PCI exige unos valores clave mínimos específicos para las operaciones criptográficas, el almacenamiento de claves y la transmisión de claves. Estos requisitos pueden cambiar cuando se revisen los estándares PCI. Las reglas especifican que las llaves de embalaje utilizadas para el almacenamiento o el transporte deben ser al menos tan resistentes como la clave que se está protegiendo. Aplicamos este requisito automáticamente durante la exportación y evitamos que las claves se protejan con claves más débiles, tal y como se muestra en la tabla siguiente.

En la tabla siguiente se muestran las combinaciones admitidas de claves para empaquetar, claves para proteger y métodos de protección.

	Clave de empaquetado											
Clave para proteger	TDES	TDES	AES_	AES_	AES_	RSA_	RSA_	RSA_	ECC_	ECC_	ECC_	Notas
TECLA TDES_2	TR-3	TR-3	TR-3	TR-3	TR-3	TR-3 RSA	TR-3 RSA	TR-3 RSA	ECDH	ECDH	ECDH	
CLAVE TDES_3	✗ No es comp e	TR-3	TR-3	TR-3	TR-3	TR-3 RSA	TR-3 RSA	TR-3 RSA	ECDH	ECDH	ECDH	

	Clave de empaquetado											
Clave para proteger	TDES	TDES	AES_	AES_	AES_	RSA_	RSA_	RSA_	ECC_	ECC_	ECC_	Notas
AES_128	✗ No es compe	✗ No es compe	TR-3	TR-3	TR-3	✗ No es compe	TR-3 RSA	TR-3 RSA	ECDI	ECDI	ECDI	
AES_192	✗ No es compe	✗ No es compe	✗ No es compe	TR-3	TR-3	✗ No es compe	✗ No es compe	✗ No es compe	✗ No es compe	ECDI	ECDI	
AES_256	✗ No es compe	✗ No es compe	✗ No es compe	✗ No es compe	TR-3	✗ No es compe	✗ No es compe	✗ No es compe	✗ No es compe	✗ No es compe	ECDI	

Para obtener más información, consulte el Apéndice D: Tamaños de clave y puntos fuertes mínimos y equivalentes para los algoritmos aprobados en los estándares PCI HSM.

Intercambio de claves de cifrado (KEK)

Se recomienda utilizar el estándar ANSI X9.24 TR-34. Este tipo de clave inicial se puede denominar clave de cifrado (KEK), clave maestra de zona (ZMK) o clave maestra de control de zona (ZCMK). Si sus sistemas o socios TR-34 aún no son compatibles, puede usar RSA. Wrap/Unwrap Si sus necesidades incluyen el intercambio de AES-256 claves, puede usar el ??? ECDH.

> **Note**
>
> Para importar tus propias claves de prueba o para sincronizarlas con tus HSM actuales, consulta el código de ejemplo de criptografía de AWS pagos que aparece en. GitHub

Intercambio de claves de trabajo (WK)

Utilizamos los estándares del sector (ANSI X9.24 TR 31-2018 y X9.143) para intercambiar las claves que funcionan. Esto requiere que ya hayas intercambiado una KEK mediante TR-34 esquemas como RSA Wrap, ECDH o similares. Este enfoque cumple con el requisito del PIN de la PCI para vincular criptográficamente el material clave según su tipo y uso en todo momento. Las claves de trabajo incluyen las claves de trabajo del adquirente, las claves de trabajo del emisor, BDK e IPEK.

Temas

- Importar claves
- Exportar claves
- Temas avanzados

Importar claves

> **Important**
>
> Los ejemplos requieren la versión más reciente de la CLI de AWS V2. Antes de empezar, asegúrese de haber actualizado a la versión más reciente.

Contenido

- Introducción a la importación de claves
- Importar claves simétricas
  - Importe claves mediante técnicas asimétricas () TR-34
  - Importe claves mediante técnicas asimétricas (ECDH)
  - Importe claves mediante técnicas asimétricas (RSA Unwrap)

- Importe claves simétricas mediante una clave de intercambio de claves preestablecida () TR-31
- Importación de claves públicas asimétricas (RSA, ECC)
  - Importar claves públicas RSA
  - Importación de claves públicas ECC

Introducción a la importación de claves

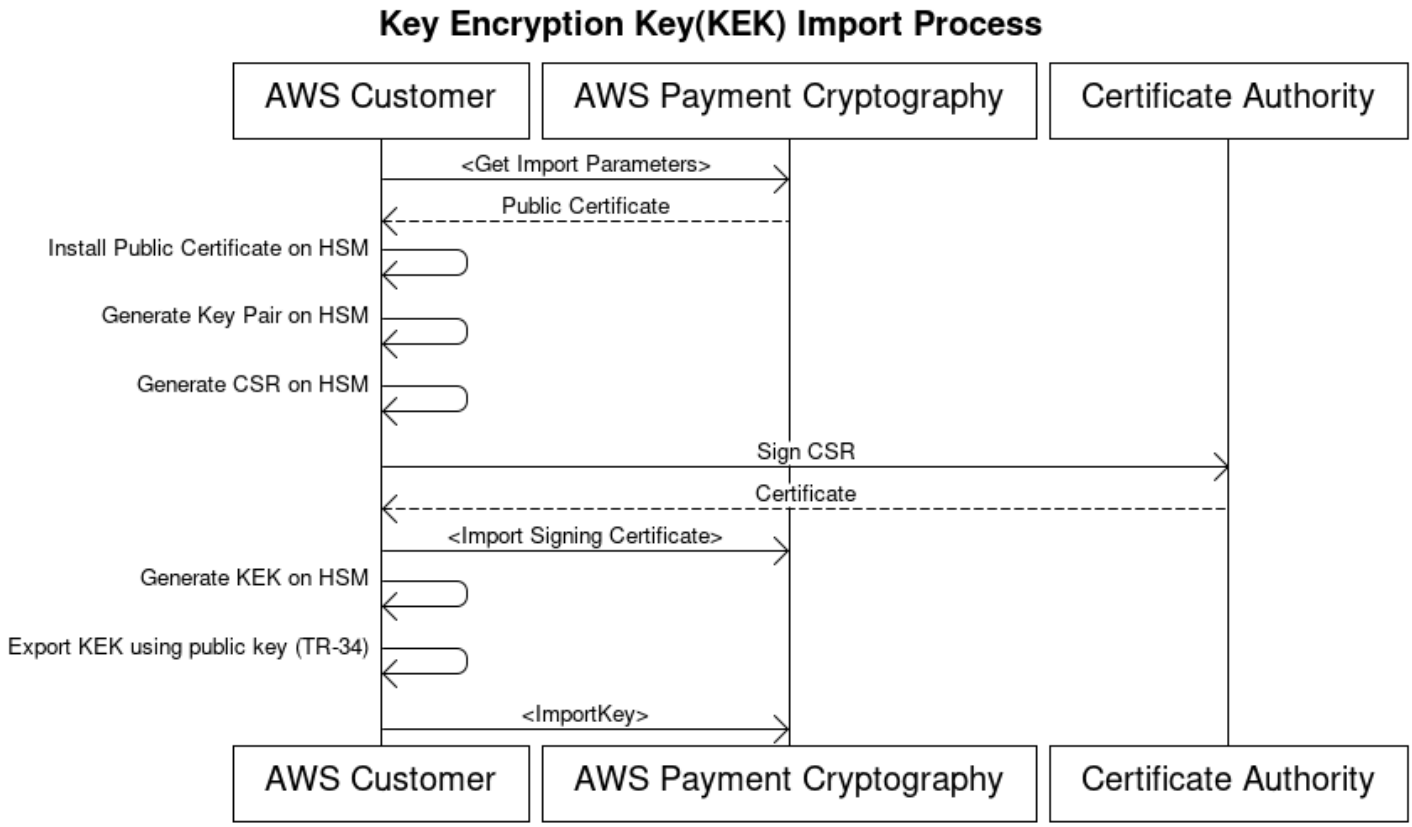
> **Note**
>
> Cuando se importan claves TR-31 o bloques de TR-34 claves, la criptografía de AWS pagos normalmente conserva (pero no utiliza) los encabezados opcionales. X9.143 El encabezado HM (tipo hash HMAC) se usa durante las operaciones criptográficas. El encabezado KP (KCV de la clave de empaquetado) es específico del proceso de importación y no se conserva.

Cuando se intercambian claves con una contraparte, normalmente se intercambia primero una clave de intercambio de claves (KEK). Esta clave se utilizará luego para proteger las claves subsiguientes. Si se utilizan formatos electrónicos, la KEK puede intercambiarse mediante técnicas asimétricas TR-34, como el ECDH o el RSA wrap. Las claves posteriores se intercambiarán mediante un intercambio de claves simétrico, como. TR-31 Esta KEK durará mucho tiempo y solo podrá actualizarse cada pocos años según la política y el período criptográfico definido.

Si solo se intercambian una o dos claves, también puede optar por utilizar técnicas asimétricas para intercambiar directamente esa clave, como un BDK. AWS La criptografía de pagos admite ambos métodos de intercambio de claves.

Importar claves simétricas

Importe claves mediante técnicas asimétricas () TR-34



TR-34 utiliza la criptografía asimétrica RSA para cifrar y firmar claves simétricas para su intercambio. Esto garantiza tanto la confidencialidad (cifrado) como la integridad (firma) de la clave empaquetada.

Para importar tus propias claves, consulta el ejemplo del proyecto AWS de criptografía de pagos publicado en GitHub. Para obtener instrucciones sobre cómo obtener import/export claves desde otras plataformas, puedes encontrar un ejemplo de código en la guía del usuario de esas plataformas GitHub o consultarla.

1. Ejecute el comando Initialize Import

   Llamar a `get-parameters-for-import` para inicializar el proceso de importación. Esta API genera un par de claves para la importación de claves, firma la clave y devuelve el certificado y la raíz del certificado. Cifre la clave que se va a exportar con esta clave. En TR-34 terminología, esto se conoce como certificado KRD. Estos certificados están codificados en base64, son de corta duración y están destinados únicamente a este propósito. Guarde el `ImportToken` valor.

```
$ aws payment-cryptography get-parameters-for-import \
    --key-material-type TR34_KEY_BLOCK \
    --wrapping-key-algorithm RSA_2048
```

```
{
    "ImportToken": "import-token-bwxli6ocftypneu5",
    "ParametersValidUntilTimestamp": 1698245002.065,
    "WrappingKeyCertificateChain": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0....",
    "WrappingKeyCertificate": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0....",
    "WrappingKeyAlgorithm": "RSA_2048"
}
```

2. Instalar el certificado público en el sistema fuente de claves

   Con la mayoría de los HSM, es necesario instalar, cargar o confiar en el certificado público generado en el paso 1 para exportar las claves con él. Esto puede incluir toda la cadena de certificados o solo el certificado raíz del paso 1, según el HSM.

3. Genere un par de claves en el sistema de origen y proporcione la cadena de certificados a AWS Payment Cryptography

   Para garantizar la integridad de la carga útil transmitida, la parte remitente (el host de distribución de claves o KDH) la firma. Genere una clave pública para este fin y cree un certificado de clave pública (X509) para contribuir a la criptografía de pagos. AWS

   Al transferir claves desde un HSM, cree un par de claves en ese HSM. El HSM, un tercero o un servicio similar AWS Privada CA pueden generar el certificado.

   Cargue el certificado raíz en AWS Payment Cryptography con los `importKey` comandos de o `RootCertificatePublicKey` y KeyMaterialType KeyUsageType de. `TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE`

   Para los certificados intermedios, utilice el `importKey` comando con KeyMaterialType de `TrustedCertificatePublicKey` y KeyUsageType de`TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE`. Repita este proceso para varios certificados intermedios. Utilice el último certificado importado `KeyArn` de la cadena como entrada para los siguientes comandos de importación.

> **Note**
>
> No importe el certificado hoja. Introdúzcalo directamente durante el comando de importación.

4. Exportar la clave desde el sistema fuente

   Muchos HSM y sistemas relacionados admiten la exportación de claves mediante la TR-34 norma. Especifique la clave pública del paso 1 como certificado KRD (cifrado) y la clave del paso 3 como certificado KDH (firma). Para importar a AWS Payment Cryptography, especifique el formato como formato de dos pasadas que TR-34.2012 no sea CMS, que también puede denominarse formato Diebold. TR-34

5. Clave de importación de llamadas

   Llama a la API ImportKey con un KeyMaterialType de`TR34_KEY_BLOCK`. Utilice el keyARN de la última CA importada en el paso 3 para`certificate-authority-public-key-identifier`, el material clave empaquetado del paso 4 para `key-material` y el certificado en hoja del paso 3 para. `signing-key-certificate` Incluya el token de importación del paso 1.

```
$ aws payment-cryptography import-key \
    --key-material='{"Tr34KeyBlock": { \
    "CertificateAuthorityPublicKeyIdentifier": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/zabouwe3574jysdl", \
    "ImportToken": "import-token-bwxli6ocftypneu5", \
    "KeyBlockFormat": "X9_TR34_2012", \
    "SigningKeyCertificate":
 "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2RENDQXFTZ0F3SUJ...", \
    "WrappedKeyBlock":
 "308205A106092A864886F70D010702A08205923082058E020101310D300B0609608648016503040201308203.
 \
    }'
```

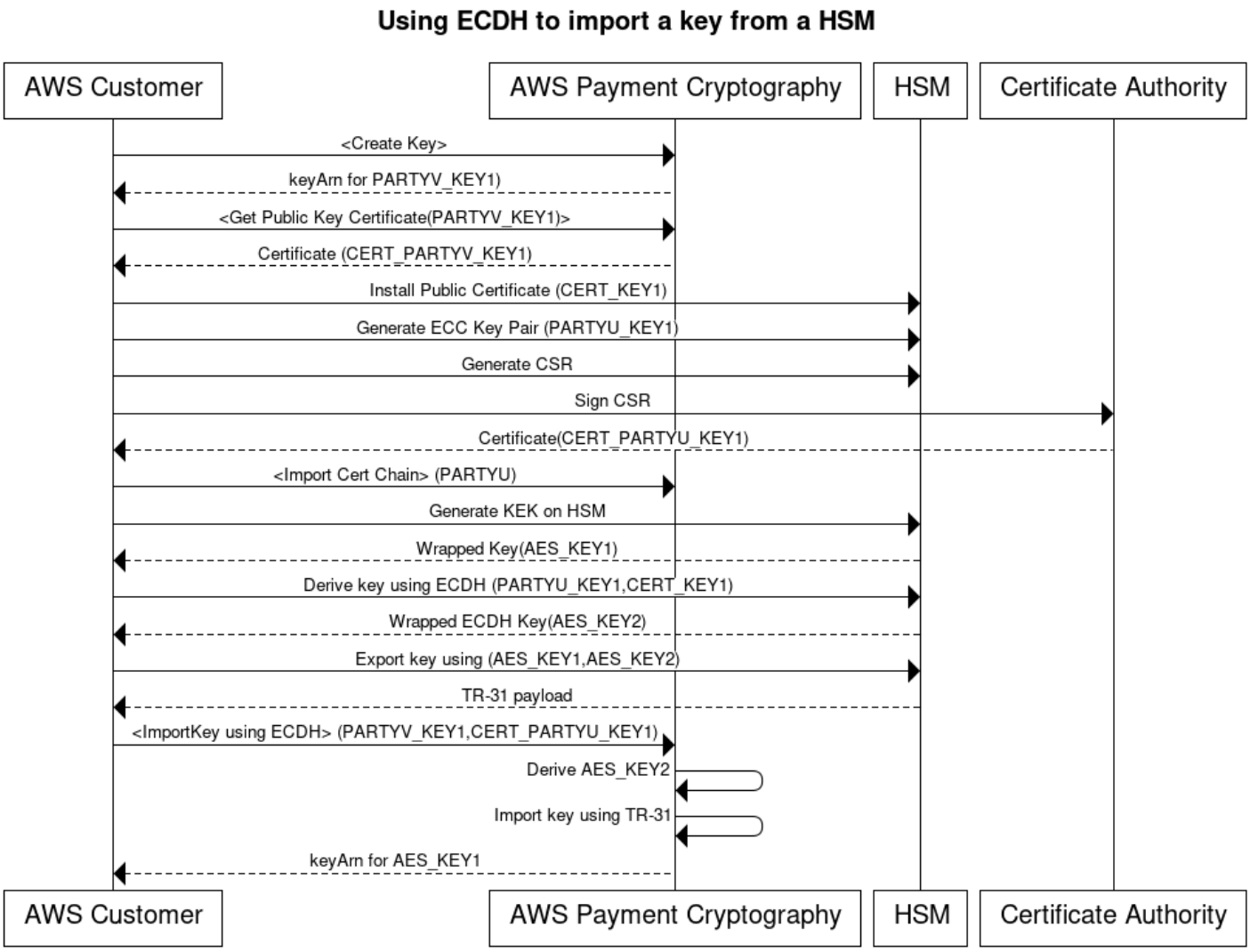
```
{
  "Key": {
    "CreateTimestamp": "2023-06-13T16:52:52.859000-04:00",
    "Enabled": true,
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ov6icy4ryas4zcza",
    "KeyAttributes": {
```

```
      "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_K1_KEY_ENCRYPTION_KEY"
    },
    "KeyCheckValue": "CB94A2",
    "KeyCheckValueAlgorithm": "ANSI_X9_24",
    "KeyOrigin": "EXTERNAL",
    "KeyState": "CREATE_COMPLETE",
    "UsageStartTimestamp": "2023-06-13T16:52:52.859000-04:00"
  }
}
```

6. Utilizar la clave importada para operaciones criptográficas o importaciones posteriores

   Si la clave importada KeyUsage fue TR31_K0_KEY_ENCRYPTION_KEY, puede usar esta clave para las importaciones de claves posteriores. TR-31 Para otros tipos de claves (como TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY), puede usar la clave directamente para las operaciones criptográficas.

Importe claves mediante técnicas asimétricas (ECDH)



Elliptic Curve Diffie-Hellman (ECDH) utiliza la criptografía asimétrica ECC para establecer una clave compartida entre dos partes sin necesidad de intercambiar claves previamente. Las claves ECDH son efímeras, por lo que Payment Cryptography no las almacena. AWS En este proceso, se obtiene una única vez mediante el KBPK/KEK ECDH. Esa clave derivada se usa inmediatamente para empaquetar la clave real que desea transferir, que puede ser otro KBPK, una clave IPEK u otros tipos de clave.

Al importar, el sistema de envío se conoce comúnmente como Parte U (iniciador) y la criptografía de AWS pagos se denomina Parte V (Respondedor).

> **Note**
>
> Si bien el ECDH se puede usar para intercambiar cualquier tipo de clave simétrica, es el único enfoque que puede transferir claves de forma segura. AES-256

1. Genere un par de claves ECC

   Llame `create-key` para crear un par de claves ECC para este proceso. Esta API genera un par de claves para la importación o exportación de claves. En el momento de la creación, especifique qué tipo de claves se pueden derivar con esta clave ECC. Cuando utilice el ECDH para intercambiar (empaquetar) otras claves, utilice un valor de. `TR31_K1_KEY_BLOCK_PROTECTION_KEY`

   > **Note**
   >
   > Si bien el ECDH de bajo nivel genera una clave derivada que se puede usar para cualquier propósito, la criptografía de AWS pagos limita la reutilización accidental de una clave para varios propósitos al permitir que una clave solo se use para un único tipo de clave derivada.

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=ECC_NIST_P256,KeyUsage=TR31_K3_ASYMMETRIC_KEY_FOR_KEY_AGREEMENT,KeyClass=ASYM
 --derive-key-usage "TR31_K1_KEY_BLOCK_PROTECTION_KEY"
```

```
{
                "Key": {
                    "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/wc3rjsssguhxtilv",
                    "KeyAttributes": {
                        "KeyUsage": "TR31_K3_ASYMMETRIC_KEY_FOR_KEY_AGREEMENT",
                        "KeyClass": "ASYMMETRIC_KEY_PAIR",
                        "KeyAlgorithm": "ECC_NIST_P256",
                        "KeyModesOfUse": {
                            "Encrypt": false,
                            "Decrypt": false,
                            "Wrap": false,
                            "Unwrap": false,
```

```
                            "Generate": false,
                            "Sign": false,
                            "Verify": false,
                            "DeriveKey": true,
                            "NoRestrictions": false
                        }
                    },
                    "KeyCheckValue": "2432827F",
                    "KeyCheckValueAlgorithm": "CMAC",
                    "Enabled": true,
                    "Exportable": true,
                    "KeyState": "CREATE_COMPLETE",
                    "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
                    "CreateTimestamp": "2025-03-28T22:03:41.087000-07:00",
                    "UsageStartTimestamp": "2025-03-28T22:03:41.068000-07:00"
                }
            }
```

2. Obtenga un certificado de clave pública

   Llame `get-public-key-certificate` para recibir la clave pública en forma de X.509 certificado firmado por la autoridad emisora de certificados de su cuenta que sea específico para la criptografía de AWS pagos en una región específica.

   Example

```
$ aws payment-cryptography get-public-key-certificate \
                --key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/wc3rjsssguhxtilv
```

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

3. Instala el certificado público en el sistema de contraparte (Party U)

   Con muchos HSM, es necesario instalar, cargar o confiar en el certificado público generado en el paso 1 para exportar las claves con él. Esto puede incluir toda la cadena de certificados o solo el certificado raíz del paso 1, según el HSM. Consulte la documentación del HSM para obtener más información.

4. Genere un par de claves ECC en el sistema fuente y proporcione una cadena de certificados a AWS Payment Cryptography

   En el ECDH, cada parte genera un par de claves y acuerda una clave común. Para que la criptografía de AWS pagos obtenga la clave, necesita la clave pública de la contraparte en formato de clave X.509 pública.

   Al transferir claves desde un HSM, cree un par de claves en ese HSM. En el caso de los HSM que admiten bloques de claves, el encabezado de la clave tendrá un aspecto similar al. `D0144K3EX00E0000` Al crear el certificado, normalmente se genera una CSR en el HSM y, a continuación, el HSM, un tercero o un servicio como este AWS Privada CA puede generar el certificado.

   Para cargar el certificado raíz en AWS Payment Cryptography, utilice el `importKey` comando «of» y «of». KeyMaterialType `RootCertificatePublicKey` KeyUsageType `TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE`

   Para los certificados intermedios, utilice el `importKey` comando con KeyMaterialType de `TrustedCertificatePublicKey` y KeyUsageType de`TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE`. Repita este proceso para varios certificados intermedios. Utilice el último certificado importado `KeyArn` de la cadena como entrada para los siguientes comandos de importación.

   > **Note**
   >
   > No importe el certificado hoja. Introdúzcalo directamente durante el comando de importación.

5. Obtenga una clave única mediante ECDH en el HSM de la Parte U

   Muchos HSM y sistemas relacionados admiten el establecimiento de claves mediante el ECDH. Especifique la clave pública del paso 1 como clave pública y la clave del paso 3 como clave privada. Para ver las opciones permitidas, como los métodos de derivación, consulta la guía de la API.

> **Note**
>
> Los parámetros de derivación, como el tipo de hash, deben coincidir exactamente en ambos lados. De lo contrario, generará una clave diferente.

6. Exportar la clave desde el sistema fuente

   Por último, exporte la clave que desea transportar a AWS Payment Cryptography mediante TR-31 comandos estándar. Especifique la clave derivada del ECDH como KBPK. La clave que se va a exportar puede ser cualquier clave TDES o AES sujeta a combinaciones TR-31 válidas, siempre que la clave de ajuste sea al menos tan fuerte como la clave que se va a exportar.

7. Clave de importación de llamadas

   Llama a la `import-key` API con un KeyMaterialType de`DiffieHellmanTr31KeyBlock`. Utilice el keyARN de la última CA importada en el paso 3 para`certificate-authority-public-key-identifier`, el material clave empaquetado del paso 4 para `key-material` y el certificado en hoja del paso 3 para. `public-key-certificate` Incluya el ARN de la clave privada del paso 1.

```
$ aws payment-cryptography import-key \
          --key-material='{
            "DiffieHellmanTr31KeyBlock": {
              "CertificateAuthorityPublicKeyIdentifier": "arn:aws:payment-
cryptography:us-east-2:111122223333:key/swseahwtq2oj6zi5",
              "DerivationData": {
                "SharedInformation": "1234567890"
              },
              "DeriveKeyAlgorithm": "AES_256",
              "KeyDerivationFunction": "NIST_SP800",
              "KeyDerivationHashAlgorithm": "SHA_256",
              "PrivateKeyIdentifier": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/wc3rjsssguhxtilv",
              "PublicKeyCertificate":
 "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUN....",
              "WrappedKeyBlock":
 "D0112K1TB00E0000D603CCA8ACB71517906600FF8F0F195A38776A7190A0EF0024F088A5342DB98E2735084A7
            }
          }'
```

```
{
        "Key": {
          "CreateTimestamp": "2025-03-13T16:52:52.859000-04:00",
          "Enabled": true,
          "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ov6icy4ryas4zcza",
          "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_K1_KEY_ENCRYPTION_KEY"
          },
          "KeyCheckValue": "CB94A2",
          "KeyCheckValueAlgorithm": "ANSI_X9_24",
          "KeyOrigin": "EXTERNAL",
          "KeyState": "CREATE_COMPLETE",
          "UsageStartTimestamp": "2025-03-13T16:52:52.859000-04:00"
        }
      }
```


8. Utilizar la clave importada para operaciones criptográficas o importaciones posteriores

   Si la clave importada KeyUsage fue TR31_K0_KEY_ENCRYPTION_KEY, puede usar esta clave para las siguientes importaciones de claves. TR-31 Para otros tipos de claves (como TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY), puede usar la clave directamente para las operaciones criptográficas.

Importe claves mediante técnicas asimétricas (RSA Unwrap)

Descripción general: la criptografía AWS de pagos admite RSA wrap/unwrap para el intercambio de claves cuando no es posible. TR-34 Del mismo modo TR-34, esta técnica utiliza la criptografía

asimétrica RSA para cifrar las claves simétricas para su intercambio. Sin embargo, a diferencia de esto TR-34, este método no hace que la parte remitente firme la carga útil. Además, esta técnica de empaquetado con RSA no mantiene la integridad de los metadatos clave durante la transferencia, ya que no incluye los bloques clave.

> **Note**
>
> Puedes usar el formato RSA para importar o exportar TDES y claves. AES-128

1. Llame al comando Initialize Import

   Llame get-parameters-for-import para inicializar el proceso de importación con un `KeyMaterialType` valor de. `KEY_CRYPTOGRAM` `RSA_2048`Utilícelo `WrappingKeyAlgorithm` cuando intercambie claves TDES. Úselo `RSA_3072` o `RSA_4096` cuando intercambie el TDES o AES-128 las claves. Esta API genera un par de claves para la importación de claves, firma la clave con una raíz de certificado y devuelve tanto el certificado como la raíz del certificado. Cifre la clave que se va a exportar con esta clave. Estos certificados son de corta duración y están destinados únicamente a este propósito.

   ```
   $ 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. Instalar el certificado público en el sistema fuente de claves

   Con muchos HSM, es necesario instalar, cargar o confiar en el certificado público (and/orsu raíz) generado en el paso 1 para exportar las claves que lo utilizan.

3. Exportar la clave desde el sistema fuente

Muchos HSM y sistemas relacionados admiten la exportación de claves mediante RSA Wrap. Especifique la clave pública del paso 1 como certificado de cifrado (). `WrappingKeyCertificate` Si necesita la cadena de confianza, utilice la `WrappingKeyCertificateChain` del paso 1. Al exportar la clave desde su HSM, especifique el formato como RSA, con el modo de relleno = PKCS #1 v2.2 OAEP (con SHA 256 o SHA 512).

4. Llame a clave de importación

   Llama a la import-key API con un de`KeyMaterialType`. `KeyMaterial` Necesitas el `ImportToken` del paso 1 y el `key-material` (material clave empaquetado) del paso 3. Proporcione los parámetros clave (como el uso de las claves), ya que RSA Wrap no utiliza bloques clave.

```
$ 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
```

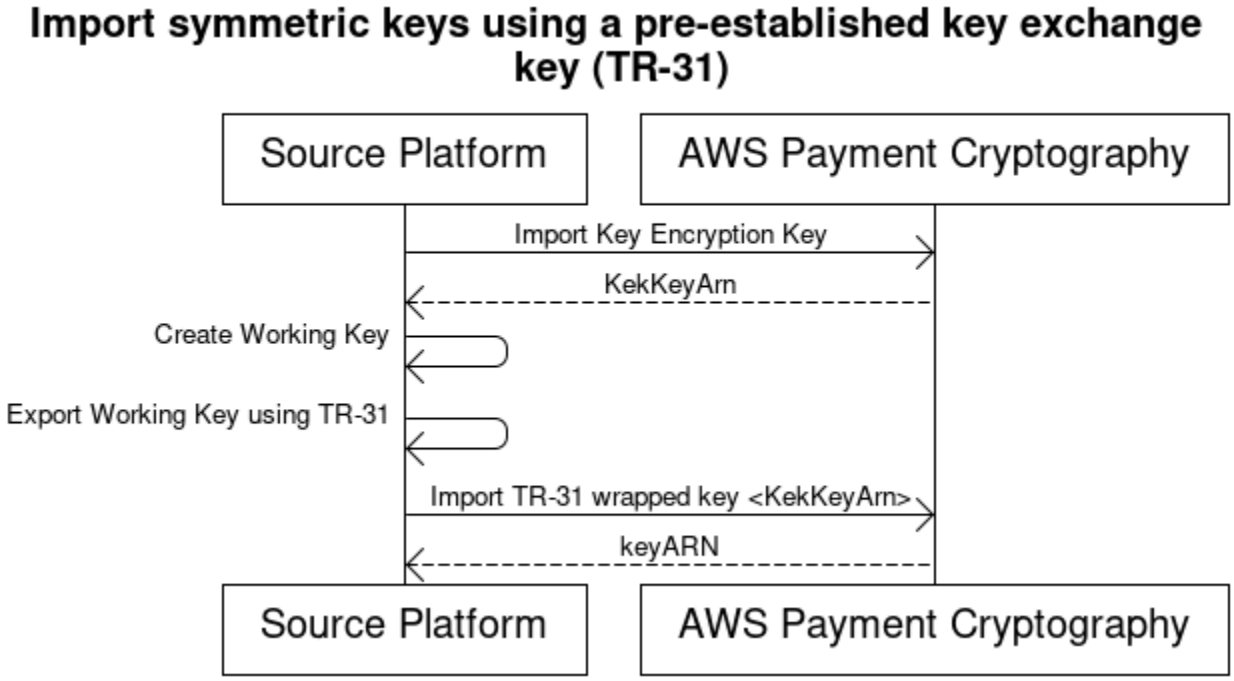
```
{
  "Key": {
    "KeyOrigin": "EXTERNAL",
    "Exportable": true,
    "KeyCheckValue": "DA1ACF",
    "UsageStartTimestamp": 1697643478.92,
    "Enabled": true,
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h",
    "CreateTimestamp": 1697643478.92,
    "KeyState": "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. Utilizar la clave importada para operaciones criptográficas o importaciones posteriores

   Si la clave importada `KeyUsage` fue `TR31_K0_KEY_ENCRYPTION_KEY` o`TR31_K1_KEY_BLOCK_PROTECTION_KEY`, puedes usar esta clave para las siguientes importaciones de claves. TR-31 Si el tipo de clave era de cualquier otro tipo (por

ejemploTR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY), puede usar la clave directamente para las operaciones criptográficas.

Importe claves simétricas mediante una clave de intercambio de claves preestablecida () TR-31



Cuando se intercambian varias claves o se admite la rotación de claves, los socios suelen intercambiar primero una clave de cifrado de clave inicial (KEK). Puedes intercambiar la KEK por la criptografía de AWS pagos, utilizando técnicas como TR-34 oIntercambio de claves físicas.

Tras establecer una KEK, puede utilizarla para transportar las claves siguientes (incluidas otras KEK). AWS La criptografía de pagos admite este intercambio de claves mediante ANSI TR-31, que los proveedores de HSM utilizan y admiten ampliamente.

1. Importar clave de cifrado (KEK)

   Asegúrate de que ya has importado tu KEK y de que tienes el KeyARN (o KeyAlias) disponible.

2. Crear la clave en la plataforma de origen

   Si la clave no existe, créala en la plataforma de origen. También puedes crear la clave en la criptografía AWS de pagos y usar el export comando.

3. Exportar la clave desde la plataforma de origen

Al exportar, especifique el formato de exportación como TR-31. La plataforma de origen solicitará la clave para exportar y la clave de cifrado que se utilizará.

4. Importar a AWS Payment Cryptography

   Al ejecutar el import-key comando, usa el keyARN (o alias) de tu clave: la clave de cifrado para. `WrappingKeyIdentifier` Usa la salida de la plataforma de origen para. `WrappedKeyBlock`

Example

```
$ aws payment-cryptography import-key \
    --key-material='{"Tr31KeyBlock": { \
    "WrappingKeyIdentifier": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/ov6icy4ryas4zcza", \
    "WrappedKeyBlock":
 "D0112B0AX00E00002E0A3D58252CB67564853373D1EBCC1E23B2ADE7B15E967CC27B85D5999EF58E11662991F
 \
    }'
```

```
{
  "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"
  }
}
```

Importación de claves públicas asimétricas (RSA, ECC)

Todos los certificados importados deben ser al menos tan seguros como el certificado emisor (predecesor) de la cadena. Esto significa que una CA RSA_2048 solo se puede usar para proteger un certificado modelo RSA_2048 y un certificado ECC debe estar protegido por otro certificado ECC de validez equivalente. Un certificado ECC P384 solo lo puede emitir una CA P384 o P521. Todos los certificados deben estar vigentes en el momento de la importación.

Importar claves públicas RSA

AWS La criptografía de pagos admite la importación de claves RSA públicas como certificados. X.509 Para importar un certificado, primero importe su certificado raíz. Todos los certificados deben estar vigentes en el momento de la importación. El certificado debe estar en formato PEM y codificado en base64.

1. Importe el certificado raíz a AWS Payment Cryptography

   Utilice el siguiente comando para importar el certificado raíz:

Example

2. Importe el certificado de clave pública a AWS Payment Cryptography

   Ahora puede importar una clave pública. Como TR-34 el ECDH depende de que se apruebe el certificado hoja en tiempo de ejecución, esta opción solo se usa para cifrar datos con una clave pública de otro sistema. KeyUsage se establecerá en TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION.

Example

```
$ aws payment-cryptography import-key \
    --key-material='{"Tr31KeyBlock": { \
    "WrappingKeyIdentifier": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/ov6icy4ryas4zcza", \
    "WrappedKeyBlock":
 "D0112B0AX00E00002E0A3D58252CB67564853373D1EBCC1E23B2ADE7B15E967CC27B85D5999EF58E11662991F
 \
    }'
```

```
{
  "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"
  }
}
```

Importación de claves públicas ECC

AWS La criptografía de pagos admite la importación de claves ECC públicas como certificados. X.509 Para importar un certificado, primero importe su certificado de CA raíz y cualquier certificado intermedio. Todos los certificados deben estar vigentes en el momento de la importación. El certificado debe estar en formato PEM y codificado en base64.

1. Importe el certificado raíz ECC a AWS Payment Cryptography

   Utilice el siguiente comando para importar el certificado raíz:

Example

2. Importe el certificado intermedio a AWS Payment Cryptography

   Utilice el siguiente comando para importar un certificado intermedio:

Example

3. Importe el certificado de clave pública (Leaf) a AWS Payment Cryptography

   Aunque puede importar un certificado ECC en formato Leaf, actualmente no hay ninguna función definida en la criptografía de AWS pagos para este certificado, aparte del almacenamiento. Esto se debe a que, cuando se utilizan las funciones del ECDH, el certificado modelo se aprueba en tiempo de ejecución.

Exportar claves

Contenido

- Exporte claves simétricas
  - Exporte claves mediante técnicas asimétricas () TR-34
  - Exporte las claves mediante técnicas asimétricas (ECDH)
  - Exporte las claves mediante técnicas asimétricas (RSA Wrap)
  - Exporte claves simétricas mediante una clave de intercambio de claves preestablecida () TR-31
- Exporte las claves iniciales de DUKPT () IPEK/IK
- Especifique los encabezados de los bloques clave para la exportación
  - Encabezados comunes
- Exporte claves asimétricas (RSA)

Exporte claves simétricas

> **Important**
>
> Asegúrese de tener la versión más reciente de AWS CLI antes de empezar. Para actualizar, consulte Instalación de AWS CLI.

Exporte claves mediante técnicas asimétricas () TR-34

TR-34 usa criptografía asimétrica RSA para cifrar y firmar claves simétricas para su intercambio. El cifrado protege la confidencialidad, mientras que la firma garantiza la integridad. Al exportar las claves, la criptografía de AWS pagos actúa como el host de distribución de claves (KDH) y el sistema de destino se convierte en el dispositivo receptor de claves (KRD).

> **Note**
>
> Si su HSM admite la TR-34 exportación pero no la TR-34 importación, le recomendamos que primero establezca una KEK compartida entre su HSM y Payment Cryptography. AWS TR-34 A continuación, puede utilizarla TR-31 para transferir las claves restantes.

1. Inicializa el proceso de exportación

   Ejecute get-parameters-for-export para generar un par de claves para la exportación de claves. Usamos este par de claves para firmar la TR-34 carga útil. En TR-34 terminología, este es el certificado de firma de KDH. Los certificados son de corta duración y solo son válidos durante el período especificado en. `ParametersValidUntilTimestamp`

   > **Note**
   >
   > Todos los certificados están codificados en base64.

   Example

   ```
   $ aws payment-cryptography get-parameters-for-export \
       --signing-key-algorithm RSA_2048 \
       --key-material-type TR34_KEY_BLOCK
   ```

   ```
   {
     "SigningKeyCertificate":
    "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2RENDQXFTZ0F3SUJ...",
     "SigningKeyCertificateChain": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS....",
     "SigningKeyAlgorithm": "RSA_2048",
     "ExportToken": "export-token-au7pvkbsq4mbup6i",
     "ParametersValidUntilTimestamp": "2023-06-13T15:40:24.036000-07:00"
   }
   ```

2. Importe el AWS Certificado de criptografía de pago a su sistema de recepción

   Importe la cadena de certificados del paso 1 a su sistema receptor.

3. Configure los certificados de su sistema receptor

Para proteger la carga útil transmitida, la parte emisora (KDH) la cifra. Su sistema receptor (normalmente su HSM o el HSM de su socio) necesita generar una clave pública y crear un certificado de clave pública. X.509 Puedes usarlo AWS Privada CA para generar certificados, pero puedes usar cualquier autoridad de certificación.

Una vez que tenga el certificado, importe el certificado raíz a AWS Payment Cryptography mediante el ImportKey comando. Establezca `KeyMaterialType` en `RootCertificatePublicKey` y `KeyUsageType` en `TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE`.

Lo utilizamos `TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE` `KeyUsageType` porque es la clave raíz que firma el certificado hoja. No necesitas importar los certificados modelo a AWS Payment Cryptography; puedes pasarlos en línea.

> **Note**
>
> Si ya importó el certificado raíz, omita este paso. Para certificados intermedios, utilice`TrustedCertificatePublicKey`.

4. Exporte su clave

   Llama a la ExportKey API con el `KeyMaterialType` valor establecido en`TR34_KEY_BLOCK`. Debes proporcionar:

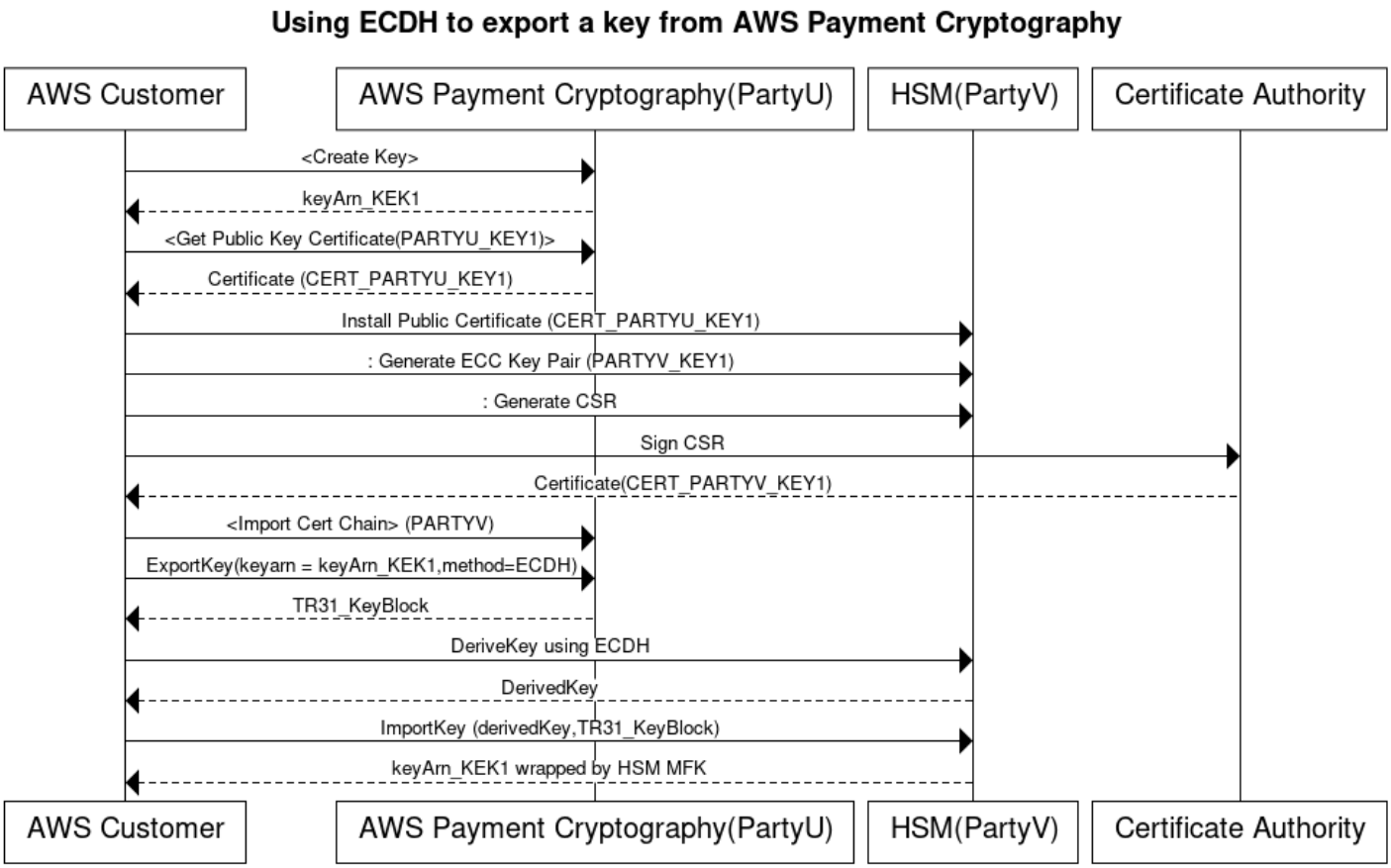
   - El keyARN de la CA raíz del paso 3 como `CertificateAuthorityPublicKeyIdentifier`
   - El certificado hoja del paso 3 como `WrappingKeyCertificate`
   - El keyARN (o alias) de la clave que desea exportar como `--export-key-identifier`
   - El token de exportación del paso 1

Example

```
$ aws payment-cryptography export-key \
    --export-key-identifier "example-export-key" \
    --key-material '{"Tr34KeyBlock": { \
    "CertificateAuthorityPublicKeyIdentifier": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/4kd6xud22e64wcbk", \
    "ExportToken": "export-token-au7pvkbsq4mbup6i", \
    "KeyBlockFormat": "X9_TR34_2012", \
    "WrappingKeyCertificate":
 "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2RENDQXFXZ0F3SUJBZ0lSQ..."} \
    }'
```

```
{
  "WrappedKey": {
    "KeyMaterial": "308205A106092A864886F70D010702A08205923082058...",
    "WrappedKeyMaterialFormat": "TR34_KEY_BLOCK"
  }
}
```

Exporte las claves mediante técnicas asimétricas (ECDH)



Elliptic Curve Diffie-Hellman (ECDH) utiliza la criptografía asimétrica ECC para establecer una clave compartida entre dos partes sin necesidad de intercambiar claves previamente. Las claves ECDH son efímeras, por lo que Payment Cryptography no las almacena. AWS En este proceso, se obtiene una única vez mediante el KBPK/KEK ECDH. Esa clave derivada se usa inmediatamente para empaquetar la clave que desea transferir, que puede ser otra clave KBPK, BDK, IPEK u otros tipos de clave.

Al exportar, la criptografía de AWS pagos se denomina Parte U (iniciador) y el sistema receptor se denomina Parte V (Respondedor).

> **Note**
>
> El ECDH se puede utilizar para intercambiar cualquier tipo de clave simétrica, pero es el único método que se puede utilizar para transferir AES-256 claves si aún no se ha establecido una KEK.

1. Genere un par de claves ECC

   Llame `create-key` para crear un par de claves ECC para este proceso. Esta API genera un par de claves para la importación o exportación de claves. En el momento de la creación, especifique qué tipo de claves se pueden derivar con esta clave ECC. Cuando utilice el ECDH para intercambiar (empaquetar) otras claves, utilice un valor de. `TR31_K1_KEY_BLOCK_PROTECTION_KEY`

   > **Note**
   >
   > Si bien el ECDH de bajo nivel genera una clave derivada que se puede usar para cualquier propósito, la criptografía de AWS pagos limita la reutilización accidental de una clave para varios propósitos al permitir que una clave solo se use para un único tipo de clave derivada.

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=ECC_NIST_P256,KeyUsage=TR31_K3_ASYMMETRIC_KEY_FOR_KEY_AGREEMENT,KeyClass=ASYM
 --derive-key-usage "TR31_K1_KEY_BLOCK_PROTECTION_KEY"
```

```
{
        "Key": {
            "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
wc3rjsssguhxtilv",
            "KeyAttributes": {
                "KeyUsage": "TR31_K3_ASYMMETRIC_KEY_FOR_KEY_AGREEMENT",
                "KeyClass": "ASYMMETRIC_KEY_PAIR",
                "KeyAlgorithm": "ECC_NIST_P256",
                "KeyModesOfUse": {
                    "Encrypt": false,
                    "Decrypt": false,
                    "Wrap": false,
                    "Unwrap": false,
                    "Generate": false,
                    "Sign": false,
                    "Verify": false,
                    "DeriveKey": true,
                    "NoRestrictions": false
                }
            },
```

```
            "KeyCheckValue": "2432827F",
            "KeyCheckValueAlgorithm": "CMAC",
            "Enabled": true,
            "Exportable": true,
            "KeyState": "CREATE_COMPLETE",
            "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
            "CreateTimestamp": "2025-03-28T22:03:41.087000-07:00",
            "UsageStartTimestamp": "2025-03-28T22:03:41.068000-07:00"
        }
    }
```

2. Obtenga un certificado de clave pública

   Llame `get-public-key-certificate` para recibir la clave pública en forma de X.509 certificado firmado por la autoridad emisora de certificados de su cuenta que sea específico para la criptografía de AWS pagos en una región específica.

   Example

```
$ aws payment-cryptography get-public-key-certificate \
          --key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/wc3rjsssguhxtilv
```

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

3. Instala el certificado público en el sistema de contraparte (Parte V)

   Con muchos HSM, es necesario instalar, cargar o confiar en el certificado público generado en el paso 1 para establecer las claves. Esto puede incluir toda la cadena de certificados o solo el certificado raíz, según el HSM. Consulte la documentación del HSM para obtener instrucciones específicas.

4. Genere un par de claves ECC en el sistema fuente y proporcione una cadena de certificados a AWS Payment Cryptography

   En el ECDH, cada parte genera un par de claves y acuerda una clave común. Para que la criptografía de AWS pagos obtenga la clave, necesita la clave pública de la contraparte en formato de clave X.509 pública.

Al transferir claves desde un HSM, cree un par de claves en ese HSM. En el caso de los HSM que admiten bloques de claves, el encabezado de la clave tendrá un aspecto similar al. `D0144K3EX00E0000` Al crear el certificado, normalmente se genera una CSR en el HSM y, a continuación, el HSM, un tercero o un servicio similar AWS Privada CA puede generar el certificado.

Para cargar el certificado raíz en AWS Payment Cryptography, utilice el `importKey` comando con KeyMaterialType de y de. `RootCertificatePublicKey` KeyUsageType `TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE`

Para los certificados intermedios, utilice el `importKey` comando con KeyMaterialType de `TrustedCertificatePublicKey` y KeyUsageType de`TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE`. Repita este proceso para varios certificados intermedios. Utilice el último certificado importado `KeyArn` de la cadena como entrada para los siguientes comandos de exportación.

> **Note**
>
> No importe el certificado hoja. Introdúzcalo directamente durante el comando de exportación.

5. Derive la clave y exporta la clave de AWS Payment Cryptography

Al exportar, el servicio obtiene una clave mediante el ECDH y, a continuación, la utiliza inmediatamente como KBPK para empaquetar la clave que se va a utilizar para la exportación. TR-31 La clave que se va a exportar puede ser cualquier clave TDES o AES sujeta a combinaciones TR-31 válidas, siempre que la clave de ajuste sea al menos tan segura como la clave que se va a exportar.

```
$ aws payment-cryptography export-key \
            --export-key-identifier arn:aws:payment-cryptography:us-
west-2:529027455495:key/e3a65davqhbpjm4h \
            --key-material='{
              "DiffieHellmanTr31KeyBlock": {
                "CertificateAuthorityPublicKeyIdentifier": "arn:aws:payment-
cryptography:us-east-2:111122223333:key/swseahwtq2oj6zi5",
                "DerivationData": {
                  "SharedInformation": "ADEF567890"
                },
```

```
                "DeriveKeyAlgorithm": "AES_256",
                "KeyDerivationFunction": "NIST_SP800",
                "KeyDerivationHashAlgorithm": "SHA_256",
                "PrivateKeyIdentifier": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/wc3rjsssguhxtilv",
                "PublicKeyCertificate": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FUR..."
              }
            }'
```

```
{
            "WrappedKey": {
                "WrappedKeyMaterialFormat": "TR31_KEY_BLOCK",
                "KeyMaterial":
 "D0112K1TB00E00007012724C0FAAF64DA50E2FF4F9A94DF50441143294E0E995DB2171554223EAA56D078C4CF
                "KeyCheckValue": "E421AD",
                "KeyCheckValueAlgorithm": "ANSI_X9_24"
            }
        }
```

6. Obtenga una clave única utilizando el ECDH en el HSM de la Parte V

   Muchos HSM y sistemas relacionados admiten el establecimiento de claves mediante el ECDH. Especifique la clave pública del paso 1 como clave pública y la clave del paso 3 como clave privada. Para conocer las opciones permitidas, como los métodos de derivación, consulta la guía de la API.

   > **Note**
   >
   > Los parámetros de derivación, como el tipo de hash, deben coincidir exactamente en ambos lados. De lo contrario, generará una clave diferente.

7. Importar la clave al sistema de destino

   Por último, importe la clave de AWS Payment Cryptography mediante TR-31 comandos estándar. Especifique la clave derivada del ECDH como KBPK y utilice el bloque de TR-31 claves que se exportó anteriormente desde AWS Payment Cryptography.

Exporte las claves mediante técnicas asimétricas (RSA Wrap)

Si TR-34 no está disponible, puedes usar RSA wrap/unwrap para el intercambio de claves. Del mismo modo TR-34, este método utiliza la criptografía asimétrica RSA para cifrar las claves simétricas. Sin embargo, RSA Wrap no incluye:

- Firma de payload por parte del remitente
- Bloques clave que mantienen la integridad de los metadatos clave durante el transporte

> **Note**
>
> Puede usar el formato RSA para exportar los TDES y AES-128 las claves.

1. Cree una clave y un certificado RSA en su sistema receptor

   Cree o identifique una clave RSA para recibir la clave empaquetada. Exigimos que las claves estén en formato X.509 de certificado. Asegúrese de que el certificado esté firmado por un certificado raíz que pueda importar a AWS Payment Cryptography.
2. Importe el certificado público raíz a AWS Payment Cryptography

   import-keyUtilícelo con la `--key-material` opción de importar el certificado

```
$ 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": "LS0tLS1CRUdJTiBDRV..."} \
    }'
```

```
{
  "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. Exporta tu clave

   Dígale a AWS Payment Cryptography que exporte su clave con su certificado modelo. Tienes que especificar:

   - El ARN del certificado raíz que importó en el paso 2
   - El certificado hoja para la exportación
   - La clave simétrica para exportar

   El resultado es una versión codificada en forma binaria (cifrada) de la clave simétrica envuelta (cifrada).

Example Ejemplo: exportar una clave

```
$ 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": "LS0tLS1CRUdJTiBDEXAMPLE...",
      "WrappingSpec": "RSA_OAEP_SHA_256"
    }
  }
}
```

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

```
{
  "WrappedKey": {
    "KeyMaterial":
 "18874746731E9E1C4562E4116D1C2477063FCB08454D757D81854AEAEE0A52B1F9D303FA29C02DC82AE778535
    "WrappedKeyMaterialFormat": "KEY_CRYPTOGRAM"
  }
}
```

4. Importe la clave a su sistema receptor

   Muchos HSM y sistemas relacionados admiten la importación de claves mediante RSA Unwrap (incluida la criptografía de AWS pagos). Al importar, especifique:

   - La clave pública del paso 1 como certificado de cifrado
   - El formato como RSA
   - Modo de relleno como PKCS #1 v2.2 OAEP (con SHA 256)

> **Note**
>
> Generamos la clave envuelta en formato HexBinary. Es posible que necesite convertir el formato si su sistema requiere una representación binaria diferente, como base64.

Exporte claves simétricas mediante una clave de intercambio de claves preestablecida () TR-31

Cuando se intercambian varias claves o se admite la rotación de claves, los socios suelen intercambiar primero una clave de cifrado de claves (KEK) inicial. Puedes intercambiar la KEK por la criptografía de AWS pagos, utilizando técnicas como TR-34 oIntercambio de claves físicas. Tras establecer una KEK, puede utilizarla para transportar las claves siguientes, incluidas otras KEK. Admitimos este intercambio de claves mediante ANSI TR-31, que es ampliamente compatible con los proveedores de HSM.

1. Configure su clave de cifrado de claves (KEK)

   Asegúrese de que ya ha intercambiado su KEK y de que tiene el KeyARN (o KeyAlias) disponible.

2. Crea tu clave en AWS Payment Cryptography

   Crea tu clave si aún no existe. Como alternativa, puede crear la clave en el otro sistema y usar el comando de importación.

3. Exporte su clave desde AWS Payment Cryptography

   Al exportar en TR-31 formato, especifique la clave que desea exportar y la clave de ajuste que va a utilizar.

Example Ejemplo: exportar una clave con el bloque de teclas TR31

```
$ 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":
 "D0144K0AB00E0000A24D3ACF3005F30A6E31D533E07F2E1B17A2A003B338B1E79E5B3AD4FBF7850FACF9A3784
    "WrappedKeyMaterialFormat": "TR31_KEY_BLOCK"
  }
}
```

4. Importe la clave a su sistema

   Utilice la implementación de la clave de importación de su sistema para importar la clave.

Exporte las claves iniciales de DUKPT () IPEK/IK

Al usar DUKPT, puede generar una única clave de derivación base (BDK) para una flota de terminales. Los terminales no tienen acceso directo al BDK. En su lugar, cada terminal recibe una clave de terminal inicial única, conocida como IPEK o clave inicial (IK). Cada IPEK se deriva del BDK mediante un número de serie clave (KSN) único.

La estructura de KSN varía según el tipo de cifrado:

- Para TDES: el KSN de 10 bytes incluye:
  - 24 bits para el identificador del conjunto de claves
  - 19 bits para el ID del terminal
  - 21 bits para el contador de transacciones
- Para AES: el KSN de 12 bytes incluye:
  - 32 bits para el ID de BDK

- 32 bits para el identificador de derivación (ID)
- 32 bits para el contador de transacciones

Proporcionamos un mecanismo para generar y exportar estas claves iniciales. Puede exportar las claves generadas mediante TR-31 métodos de TR-34 empaquetado RSA o mediante ellos. Ten en cuenta que las claves IPEK no se conservan y no se pueden usar para operaciones posteriores con la criptografía de pagos. AWS

No aplicamos la división entre las dos primeras partes de la KSN. Si quieres almacenar el identificador de derivación con el BDK, puedes usar etiquetas. AWS

> **Note**
>
> La parte del contador del KSN (32 bits para AES DUKPT) no se usa para la derivación. IPEK/IK Por ejemplo, las entradas 12345678901234560001 y 12345678901234569999 generarán el mismo IPEK.

```
$ 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"
}
}
```

Especifique los encabezados de los bloques clave para la exportación

Puede modificar o añadir la información de los bloques clave al exportar en formato ASC o en formato TR-31 . TR-34 En la siguiente tabla se describe el formato del bloque TR-31 clave y los elementos que se pueden modificar durante la exportación.

Atributo de bloque clave	Finalidad	¿Se puede modificar durante la exportación?	Notas
ID de versión.	Define el método utilizado para proteger el material clave. La norma incluye: • Versiones A y C (variante clave, obsoleta) • Versión B (derivación mediante TDES) • Versión D (derivación de claves mediante AES)	No	Usamos la versión B para las claves de empaquetado con TDES y la versión D para las claves de empaquetado con AES. Solo admitimos las versiones A y C para las operaciones de importación.
Longitud del bloque clave	Especifica la longitud del mensaje restante	No	Calculamos este valor automáticamente. Es posible que la longitud parezca incorrecta antes de descifrar la carga útil, ya que podemos añadir un relleno de teclas según lo exija la especificación.

Atributo de bloque clave	Finalidad	¿Se puede modificar durante la exportación?	Notas
Uso de claves	Define los propósitos permitidos para la clave, como: • C0 (verificación de la tarjeta) • B0 (clave de derivación básica)	No	
Algoritmo	Especifica el algoritmo de la clave subyacente. Apoyamos: • (MAREAS) • (HHMAC) • (UN AS)	No	Exportamos este valor tal cual.
Uso de claves	Define las operaciones permitidas, como: • Generar y verificar (C) • Encrypt/Decrypt/Wrap/Unwrap (B)	Sí*	
Versión clave	Indica el número de versión de la clave replacement/rotation. El valor predeterminado es 00 si no se especifica.	Sí, se puede añadir	

Atributo de bloque clave	Finalidad	¿Se puede modificar durante la exportación?	Notas
Exportabilidad clave	Controla si la clave se puede exportar: • N: sin capacidad de exportación • E - Exportar según X9.24 (bloques clave) • S: Exporta en formatos de bloque de teclas o sin bloques de teclas	Sí*	
Bloques de teclas opcionales	Sí, se puede añadir	Los bloques de claves opcionales son name/value pares enlazados criptográficamente a la clave. Por ejemplo, el KeySet ID de las claves DUKPT. Calculamos automáticamente el número de bloques, la longitud de cada bloque y el bloque de relleno (PB) en función de los datos name/value introducidos por el par.	

*Al modificar los valores, el nuevo valor debe ser más restrictivo que el valor actual en la criptografía de AWS pagos. Por ejemplo:

- Si el modo de uso clave actual es Generate=True, Verify=True, puedes cambiarlo a Generate=True, Verify=False
- Si la clave ya está configurada como no exportable, no puedes cambiarla a exportable

Al exportar claves, aplicamos automáticamente los valores actuales de la clave que se está exportando. Sin embargo, es posible que desees modificar o añadir esos valores antes de enviarlos al sistema receptor. Estos son algunos escenarios comunes:

- Al exportar una clave a un terminal de pago, defina su exportabilidad de tal `Not Exportable` manera que los terminales normalmente solo importan claves y no deberían exportarlas.
- Cuando necesites pasar los metadatos de la clave asociada al sistema receptor, usa encabezados TR-31 opcionales para vincular criptográficamente los metadatos a la clave en lugar de crear una carga personalizada.
- Defina la versión de la clave utilizando el `KeyVersion` campo para realizar un seguimiento de la rotación de las claves.

TR-31/X9.143 define encabezados comunes, pero puede usar otros encabezados siempre que cumplan con los parámetros de criptografía de AWS pagos y su sistema receptor pueda aceptarlos. Para obtener más información sobre los encabezados de los bloques clave durante la exportación, consulte los encabezados de los bloques clave en la guía de la API.

Este es un ejemplo de exportación de una clave BDK (por ejemplo, a un KIF) con estas especificaciones:

- Versión clave: 02
- KeyExportability: NO EXPORTABLE
- KeySetID: 00ABCDEFAB (00 indica la clave TDES, ABCDEFABCD es la clave inicial)

Como no especificamos los modos de uso clave, esta clave hereda el modo de uso de arn:aws:payment-cryptography:us-east- 2:111122223333: (= true). key/5rplquuwozodpwsp DeriveKey

> **Note**
>
> Incluso si estableces la exportabilidad como No exportable en este ejemplo, el KIF aún puede: ???
>
> - Derivar claves como las que se usan en IPEK/IK DUKPT
> - Exporte estas claves derivadas para instalarlas en los dispositivos
>
> Esto está permitido específicamente por las normas.

```
$ aws payment-cryptography export-key \
    --key-material='{"Tr31KeyBlock": { \
    "WrappingKeyIdentifier": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ov6icy4ryas4zcza", \
    "KeyBlockHeaders": { \
    "KeyModesOfUse": { \
    "Derive": true}, \
    "KeyExportability": "NON_EXPORTABLE", \
    "KeyVersion": "02", \
    "OptionalBlocks": { \
    "BI": "00ABCDEFABCD"}}} \
    }' \
    --export-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/5rplquuwozodpwsp
```

```
{
"WrappedKey": {
    "WrappedKeyMaterialFormat": "TR31_KEY_BLOCK",
    "KeyMaterial": "EXAMPLE_KEY_MATERIAL_TR31",
    "KeyCheckValue": "A4C9B3",
    "KeyCheckValueAlgorithm": "ANSI_X9_24"
    }
}
```

Encabezados comunes

X9.143 define ciertos encabezados para casos de uso comunes. Con la excepción del encabezado HM (HMAC Hash), AWS Payment Cryptography no analiza ni utiliza estos encabezados.

Nombre del encabezado	Finalidad	Validación típica	Notas
BI	Identificador de clave de derivación base para DUKPT	2 caracteres hexadecimales (00 para TDES, 11 para AES), 10 caracteres hexadecimales para TDES KSI u 8 caracteres hexadecimales para BDK ID (AES DUKPT).	Contiene el (ID de BDK, para AES DUKPT) o el identificador del conjunto de claves (KSI, para TDES DUKPT). Se puede usar para intercambiar el ID de BDK o el KSI, pero no es necesario intercambiar los demás datos contenidos en los bloques IK y KS. Normalmente, el BI se utiliza cuando se transmite a un KIF, mientras que el IK o el KS se utilizan cuando se inyectan en el propio terminal.
HM	Especifica el tipo de hash para las operaciones de HMAC	• 10 — SHA-1 • 20 — SHA-224 • 21 — SHA-256 • 22 — SHA-384 • 23 — SHA-512 • 24 — SHA-512/224 • 25 — SHA-512/256 • 30 — SHA3-224 • 31 — SHA3-256 • 32 — SHA3-384	El servicio rellena automáticamente este campo al exportar y lo analizará al importarlo. Los tipos de hash que no admite el servicio, como SHAKE128, se pueden importar, pero es posible que no se puedan utilizar para

Nombre del encabezado	Finalidad	Validación típica	Notas
		• 33 — SHA3-512 • 40 — SHAKE128 • 41 — SHAKE256	funciones criptográficas.
IK	Número de serie clave inicial para AES DUKPT	16 caracteres hexadecimales	Este valor se usa para crear una instancia del uso de la clave DUKPT inicial en el dispositivo receptor e identifica la clave inicial derivada de un BDK. Este campo normalmente contiene los datos de derivación, pero no el contador. Utilice KS para TDES DUKPT.
ES	Número de serie clave inicial para TDES DUKPT	20 caracteres hexadecimales	Este valor se usa para crear una instancia del uso de la clave DUKPT inicial en el dispositivo receptor e identifica la clave inicial derivada de un BDK. Este campo normalmente contiene los datos de derivación más un valor de contador reducido a cero. Utilice IK para AES DUKPT.

Nombre del encabezado	Finalidad	Validación típica	Notas
KP	Valor de verificación clave del KBPK ???	Dos caracteres hexadecimales representan el método KCV (00 para el X9.24 método y 01 para el método CMAC). Seguido del valor KCV, que normalmente es de 6 caracteres hexadecimales. Por ejemplo, 010FA329 representa el KCV de 0FA329 calculado con el método 01 (CMAC).	Calculado de acuerdo con el X9.24-1-2017 anexo A de la ANSI; el KCV se utiliza para la validación de claves, no como un mecanismo de integridad.
PB	Bloque de acolchado	caracteres ASCII imprimibles aleatorios	El servicio rellena automáticamente este campo al exportar para garantizar que los encabezados opcionales sean múltiplos de la longitud del bloque de cifrado

Exporte claves asimétricas (RSA)

Para exportar una clave pública en forma de certificado, utilice el get-public-key-certificate comando. Este comando devuelve:

- El certificado
- El certificado raíz

Ambos certificados están codificados en base64.

> **Note**
>
> Esta operación no es indempotente: las llamadas posteriores pueden generar certificados diferentes incluso cuando se utilice la misma clave subyacente.

Example

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

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

Temas avanzados

En esta sección se describen escenarios y configuraciones avanzados de intercambio de claves.

Temas

- Traiga su propia autoridad de certificación (BYOCA)
- Intercambio de claves físicas

Traiga su propia autoridad de certificación (BYOCA)

De forma predeterminada, cuando se necesita un certificado de clave pública para las claves asimétricas (RSA, ECC) creadas en el servicio, estos certificados son emitidos por una autoridad certificadora (CA) especializada en criptografía de AWS pagos y única para cada cuenta. El objetivo es facilitar su uso X.509 sin la carga de identificar o configurar una CA o administrar las solicitudes de firma de certificados (CSR).

AWS La criptografía de pagos también ofrece la posibilidad de utilizar su propia CA cuando sea necesario por motivos relacionados con las políticas o el cumplimiento de normas.

Temas

- Descripción general de
- Flujo de trabajo de exportación de BYOCA
- Flujo de trabajo de importación de BYOCA
- Notas adicionales

Descripción general de

Con BYOCA, puede usar su propia CA para TR-34 import/export las transferencias de claves RSA Unwrap y ECDH. Use BYOCA cuando necesite una cadena de certificados uniforme en toda su organización o cuando los socios requieran certificados de CA específicos. Los siguientes ejemplos muestran el flujo de trabajo de BYOCA para la exportación e importación de TR-34 claves. TR-34

Hay tres diferencias clave con respecto al flujo estándar TR-34 :

1. La clave RSA se crea con CreateKey. Antes, `GetParametersForExport` o la `GetParametersForImport` creó para usted.
2. La GetCertificateSigningRequest API crea una CSR. A continuación, su CA externa puede firmarla.
3. ImportKeyLas API ExportKey Y aceptan un certificado en el momento de la llamada. El token ahora es opcional.

⚠ Consideraciones importantes

- En estos ejemplos, se utilizan RSA-2048 claves y se envuelven una TDES-2KEY clave. Al exportar AES-128, asegúrese de que todas las claves sean RSA-3072 o RSA-4096.
- El error más común es que la clave representada por `SigningKeyIdentifier` y `SigningKeyCertificate` no coincide.

Flujo de trabajo de exportación de BYOCA

Los pasos siguientes muestran el flujo de trabajo completo de BYOCA para la exportación. TR-34

Steps

- Paso 1: Crear la clave RSA
- Paso 2: Generar la solicitud de firma del certificado
- Paso 3: Revise la CSR (opcional)
- Paso 4: Firme la CSR con una autoridad de certificación
- Paso 5: Importar el certificado de CA
- Paso 6: Obtenga el certificado de cifrado KRD
- Paso 7: Exportar la clave con BYOCA

Paso 1: Crear la clave RSA

En primer lugar, cree un par de claves RSA que, en última instancia, será el certificado de firma de KDH. Puede agregar etiquetas para identificar el propósito de la clave.

Example Cree una clave RSA para firmar

```
$ aws payment-cryptography create-key --exportable \
    --key-attributes
 KeyAlgorithm=RSA_2048,KeyUsage=TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE,KeyClass=ASYMMETRI
```

```
{
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-east-1:111122223333:key/
xgmq6fs6uow736uc",
        "KeyAttributes": {
            "KeyUsage": "TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE",
            "KeyClass": "ASYMMETRIC_KEY_PAIR",
            "KeyAlgorithm": "RSA_2048",
            "KeyModesOfUse": {
                "Sign": true
            }
        },
        "KeyCheckValue": "41E3723C",
        "KeyCheckValueAlgorithm": "SHA_1",
        "Enabled": true,
        "Exportable": true,
        "KeyState": "CREATE_COMPLETE",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY"
    }
}
```

Tome nota de ellaKeyArn, ya que la necesitará en el siguiente paso.

Paso 2: Generar la solicitud de firma del certificado

Genere una solicitud de firma de certificado (CSR) para que la firme su CA externa mediante la GetCertificateSigningRequest API. El resultado es un archivo PEM codificado en base64. Si decodifica el contenido en base64 y lo guarda, tendrá una CSR válida en formato PEM.

Example Genera CSR

```
$ aws payment-cryptography get-certificate-signing-request \
    --key-identifier arn:aws:payment-cryptography:us-east-1:111122223333:key/
xgmq6fs6uow736uc \
    --signing-algorithm SHA512 \
    --certificate-subject '{
        "CommonName": "MyCertificateAWSUSEAST",
        "Organization": "Amazon",
        "OrganizationUnit": "PaymentCryptography",
        "Country": "US",
        "StateOrProvince": "Virginia",
        "City": "Arlington"
    }'
```

```
{
    "CertificateSigningRequest": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBSRVFVRVNULS0tLS0..."
}
```

El CertificateSigningRequest campo contiene el archivo PEM completo, codificado en base64. El archivo PEM incluye los marcadores y. -----BEGIN CERTIFICATE REQUEST----- -----END CERTIFICATE REQUEST----- Envíe este valor a su CA para que lo firme.

Paso 3: Revise la CSR (opcional)

Si lo desea, puede usar OpenSSL para revisar el contenido de la CSR y asegurarse de que es válido y cumple con lo esperado.

Example Revisa la CSR con OpenSSL

```
$ echo "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBSRVFVRVNULS0tLS0..." | base64 -d | openssl req -
text
```

Paso 4: Firme la CSR con una autoridad de certificación

Tras generar la CSR, es necesario que la firme una autoridad de certificación (CA). En los entornos de producción, normalmente Autoridad de certificación privada de AWS utilizaría la infraestructura de CA establecida de su organización. Con fines de prueba, puede usar OpenSSL para crear un certificado autofirmado.

Utilización Autoridad de certificación privada de AWS

Para firmar la CSR Autoridad de certificación privada de AWS, primero decodifique la CSR codificada en base64, guárdela en un archivo y, a continuación, utilice la API. IssueCertificate

Example Firma la CSR con AWS Privada CA

```
$ echo "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBSRVFVRVNULS0tLS0..." | base64 -d > csr.pem

$ aws acm-pca issue-certificate \
    --certificate-authority-arn arn:aws:acm-pca:us-east-1:111122223333:certificate-
authority/12345678-1234-1234-1234-123456789012 \
    --csr fileb://csr.pem \
    --signing-algorithm SHA256WITHRSA \
    --validity Value=365,Type=DAYS
```

```
{
    "CertificateArn": "arn:aws:acm-pca:us-east-1:111122223333:certificate-
authority/12345678-1234-1234-1234-123456789012/certificate/abcdef1234567890"
}
```

A continuación, recupera el certificado firmado:

Example Recuperar el certificado firmado

```
$ aws acm-pca get-certificate \
    --certificate-authority-arn arn:aws:acm-pca:us-east-1:111122223333:certificate-
authority/12345678-1234-1234-1234-123456789012 \
    --certificate-arn arn:aws:acm-pca:us-east-1:111122223333:certificate-
authority/12345678-1234-1234-1234-123456789012/certificate/abcdef1234567890
```

```
{
    "Certificate": "-----BEGIN CERTIFICATE-----\nMIID...\n-----END CERTIFICATE-----",
    "CertificateChain": "-----BEGIN CERTIFICATE-----\nMIID...\n-----END
 CERTIFICATE-----"
```

```
}
```

Guarde el contenido del certificado para usarlo en el paso de exportación. Deberás codificarlo en base64 cuando lo proporciones a la API. `ExportKey`

Uso de OpenSSL para realizar pruebas

Con fines de prueba, puede usar OpenSSL para crear una CA autofirmada y firmar la CSR. En primer lugar, cree una clave privada de CA y un certificado autofirmado:

Example Cree una CA de prueba con OpenSSL

```
$ # Generate CA private key
openssl genrsa -out ca-key.pem 4096

$ # Create self-signed CA certificate
openssl req -new -x509 -days 3650 -key ca-key.pem -out ca-cert.pem \
    -subj "/C=US/ST=Virginia/L=Arlington/O=TestOrg/CN=Test CA"
```

A continuación, descodifique la CSR del paso anterior y fírmela con su CA de prueba:

Example Firme la CSR con OpenSSL

```
$ # Decode the base64-encoded CSR
echo "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBSRVFVRVNULS0tLS0..." | base64 -d > csr.pem

$ # Sign the CSR with the CA
openssl x509 -req -in csr.pem -CA ca-cert.pem -CAkey ca-key.pem \
    -CAcreateserial -out signed-cert.pem -days 365 -sha512
```

```
Certificate request self-signature ok
subject=C=US, ST=Virginia, L=Arlington, O=Amazon, OU=PaymentCryptography,
 CN=MyCertificateAWSUSEAST
```

El certificado firmado ya está aquí. `signed-cert.pem` Deberás codificar este certificado en base64 cuando lo proporciones a la API: `ExportKey`

Example Codifica en Base64 el certificado firmado

```
$ cat signed-cert.pem | base64 -w 0
```

Paso 5: Importar el certificado de CA

Es necesario confiar primero en cualquier CA que se utilice para evitar que se utilicen certificados arbitrarios. Importe el certificado raíz de su CA externa mediante la ImportKey API. Si utiliza una CA intermedia, `import-key` vuelva a llamar, pero especifique `TrustedPublicKey` en lugar de `RootCertificatePublicKey` especificar el ARN de la CA raíz.

Example Importe el certificado de CA raíz

```
$ 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": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0t..."
    }
}'
```

```
{
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-east-1:111122223333:key/
xivpaqy7qbbm7cdw",
        "KeyAttributes": {
            "KeyUsage": "TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE",
            "KeyClass": "PUBLIC_KEY",
            "KeyAlgorithm": "RSA_4096",
            "KeyModesOfUse": {
                "Verify": true
            }
        },
        "Enabled": true,
        "KeyState": "CREATE_COMPLETE",
        "KeyOrigin": "EXTERNAL"
    }
}
```

Tome nota de las CA `KeyArn` para utilizarlas en la etapa de exportación.

Paso 6: Obtenga el certificado de cifrado KRD

En este ejemplo, volvemos a realizar la importación a AWS Payment Cryptography, por lo que llamamos al servicio para recibir un certificado de clave pública de KRD mediante la API. GetParametersForImport En un escenario real, lo proporcionaría otro sistema, como un HSM, un cajero automático, un terminal de pago o un sistema de gestión de terminales de pago.

Example Obtener parámetros para importación

```
$ aws payment-cryptography get-parameters-for-import \
    --key-material-type "TR34_KEY_BLOCK" \
    --wrapping-key-algorithm RSA_2048
```

```
{
    "WrappingKeyCertificate": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0t...",
    "WrappingKeyCertificateChain": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0t...",
    "WrappingKeyAlgorithm": "RSA_2048",
    "ImportToken": "import-token-v2rxpl6drxeptn7w",
    "ParametersValidUntilTimestamp": "2025-11-01T18:45:31.271000-07:00"
}
```

Paso 7: Exportar la clave con BYOCA

Por último, exporte la clave TR-34 con su propio CA-signed certificado mediante la ExportKey API. Proporcione el certificado de firma firmado por su CA externa.

Example TR-34 Exporte con BYOCA

```
$ aws payment-cryptography export-key \
    --export-key-identifier arn:aws:payment-cryptography:us-east-1:111122223333:key/
iox73p5f4c4yjiod \
    --key-material '{
        "Tr34KeyBlock": {
            "CertificateAuthorityPublicKeyIdentifier": "arn:aws:payment-
cryptography:us-east-1:111122223333:key/j625deyfqlwctu57",
            "SigningKeyIdentifier": "arn:aws:payment-cryptography:us-
east-1:111122223333:key/xgmq6fs6uow736uc",
            "SigningKeyCertificate": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0t...",
            "KeyBlockFormat": "X9_TR34_2012",
            "WrappingKeyCertificate": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0t..."
        }
    }'
```

```
{
    "WrappedKey": {
        "WrappedKeyMaterialFormat": "TR34_KEY_BLOCK",
        "KeyMaterial": "3082055A06092A864886F70D010702A082054B30820547...",
        "KeyCheckValue": "3DCA31",
        "KeyCheckValueAlgorithm": "ANSI_X9_24"
    }
}
```

El sistema receptor ahora puede importar el bloque de claves exportado mediante el proceso de TR-34 importación estándar.

Flujo de trabajo de importación de BYOCA

Los siguientes pasos le guiarán por el flujo de trabajo de BYOCA para la importación de claves. TR-34 Utilice este flujo de trabajo para recibir una clave de criptografía de AWS pagos con su propio certificado de cifrado CA-signed KRD (dispositivo receptor de claves). Con la importación de BYOCA, usted controla la cadena de confianza mediante una CA que ya administra.

Steps

- Paso 1: Crear la clave RSA
- Paso 2: generar la solicitud de firma del certificado
- Paso 3: Revise la CSR (opcional)
- Paso 4: Firme la CSR con una autoridad de certificación
- Paso 5: Importar el certificado de CA
- Paso 6: Obtenga el certificado de firma de KDH del sistema de envío
- Paso 7: Importar la clave con BYOCA

Paso 1: Crear la clave RSA

Cree un par de claves RSA para usarlo como clave de cifrado KRD. Esta clave desenvuelve y descifra la clave entrante durante la importación. TR-34 La clave no se puede exportar porque permanece en el servicio como clave receptora.

Example Cree la clave de cifrado KRD

```
$ aws payment-cryptography create-key \
```

```
    --key-attributes
 KeyAlgorithm=RSA_2048,KeyUsage=TR31_K2_TR34_ASYMMETRIC_KEY,KeyClass=ASYMMETRIC_KEY_PAIR,KeyMod
```

```
{
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-east-1:111122223333:key/
bm7t4qv8hzk24jce",
        "KeyAttributes": {
            "KeyUsage": "TR31_K2_TR34_ASYMMETRIC_KEY",
            "KeyClass": "ASYMMETRIC_KEY_PAIR",
            "KeyAlgorithm": "RSA_2048",
            "KeyModesOfUse": {
                "Unwrap": true,
                "Decrypt": true
            }
        },
        "KeyCheckValue": "7FA29C1E",
        "KeyCheckValueAlgorithm": "SHA_1",
        "Enabled": true,
        "Exportable": false,
        "KeyState": "CREATE_COMPLETE",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY"
    }
}
```

Anote el valor `KeyArn`. Lo necesitará en el siguiente paso.

Paso 2: generar la solicitud de firma del certificado

Genere una solicitud de firma de certificado (CSR) para la clave de cifrado KRD mediante la GetCertificateSigningRequest API. El resultado es un archivo PEM codificado en base64. Si decodifica el contenido en base64 y lo guarda, tendrá una CSR válida en formato PEM.

Example Genere CSR para KRD Key

```
$ aws payment-cryptography get-certificate-signing-request \
    --key-identifier arn:aws:payment-cryptography:us-east-1:111122223333:key/
bm7t4qv8hzk24jce \
    --signing-algorithm SHA512 \
    --certificate-subject '{
        "CommonName": "MyImportKRDCertAWSUSEAST",
        "Organization": "Amazon",
        "OrganizationUnit": "PaymentCryptography",
```

```
        "Country": "US",
        "StateOrProvince": "Virginia",
        "City": "Arlington"
    }'
```

```
{
    "CertificateSigningRequest": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBSRVFVRVNULS0tLS0..."
}
```

El `CertificateSigningRequest` campo contiene el archivo PEM completo, codificado en base64. El archivo PEM incluye los marcadores y. `-----BEGIN CERTIFICATE REQUEST-----` `-----END CERTIFICATE REQUEST-----` Envíe este valor a su CA para que lo firme.

Paso 3: Revise la CSR (opcional)

Si lo desea, puede usar OpenSSL para revisar el contenido de la CSR y asegurarse de que es válido y cumple con lo esperado.

Example Revisa la CSR con OpenSSL

```
$ echo "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBSRVFVRVNULS0tLS0..." | base64 -d | openssl req -text
```

Paso 4: Firme la CSR con una autoridad de certificación

Después de generar la CSR, pídale a una autoridad de certificación (CA) que la firme. En los entornos de producción, normalmente Autoridad de certificación privada de AWS utilizaría la infraestructura de CA establecida de su organización. Con fines de prueba, puede usar OpenSSL para crear un certificado autofirmado.

Utilización Autoridad de certificación privada de AWS

Para firmar la CSR Autoridad de certificación privada de AWS, primero decodifique la CSR codificada en base64, guárdela en un archivo y, a continuación, utilice la API. IssueCertificate

Example Firma la CSR con AWS Privada CA

```
$ echo "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBSRVFVRVNULS0tLS0..." | base64 -d > krd-csr.pem

$ aws acm-pca issue-certificate \
    --certificate-authority-arn arn:aws:acm-pca:us-east-1:111122223333:certificate-authority/12345678-1234-1234-1234-123456789012 \
```

```
    --csr fileb://krd-csr.pem \
    --signing-algorithm SHA256WITHRSA \
    --validity Value=365,Type=DAYS
```

```
{
    "CertificateArn": "arn:aws:acm-pca:us-east-1:111122223333:certificate-
authority/12345678-1234-1234-1234-123456789012/certificate/fedcba0987654321"
}
```

A continuación, recupera el certificado firmado:

Example Recuperar el certificado firmado

```
$ aws acm-pca get-certificate \
    --certificate-authority-arn arn:aws:acm-pca:us-east-1:111122223333:certificate-
authority/12345678-1234-1234-1234-123456789012 \
    --certificate-arn arn:aws:acm-pca:us-east-1:111122223333:certificate-
authority/12345678-1234-1234-1234-123456789012/certificate/fedcba0987654321
```

```
{
    "Certificate": "-----BEGIN CERTIFICATE-----\nMIID...\n-----END CERTIFICATE-----",
    "CertificateChain": "-----BEGIN CERTIFICATE-----\nMIID...\n-----END
 CERTIFICATE-----"
}
```

Guarde el contenido del certificado para usarlo en el paso de importación. Base64-encode el certificado antes de proporcionarlo a la `ImportKey` API.

Uso de OpenSSL para realizar pruebas

Con fines de prueba, puede usar OpenSSL para crear una CA autofirmada y firmar la CSR. En primer lugar, cree una clave privada de CA y un certificado autofirmado:

Example Cree una CA de prueba con OpenSSL

```
$ # Generate CA private key
openssl genrsa -out ca-key.pem 4096

$ # Create self-signed CA certificate
openssl req -new -x509 -days 3650 -key ca-key.pem -out ca-cert.pem \
    -subj "/C=US/ST=Virginia/L=Arlington/O=TestOrg/CN=Test CA"
```

A continuación, descodifique la CSR del paso anterior y fírmela con su CA de prueba:

Example Firme la CSR con OpenSSL

```
$ # Decode the base64-encoded CSR
echo "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBSRVFVRVNULS0tLS0..." | base64 -d > krd-csr.pem

$ # Sign the CSR with the CA
openssl x509 -req -in krd-csr.pem -CA ca-cert.pem -CAkey ca-key.pem \
    -CAcreateserial -out krd-signed-cert.pem -days 365 -sha512
```

```
Certificate request self-signature ok
subject=C=US, ST=Virginia, L=Arlington, O=Amazon, OU=PaymentCryptography,
 CN=MyImportKRDCertAWSUSEAST
```

El certificado firmado ya está aquí. `krd-signed-cert.pem` Base64-encode este certificado antes de proporcionarlo a la `ImportKey` API:

Example Codifica en Base64 el certificado firmado

```
$ cat krd-signed-cert.pem | base64 -w 0
```

Paso 5: Importar el certificado de CA

Antes de usar una CA, primero debe confiar en ella. Importe el certificado raíz de su CA externa mediante la ImportKey API. Si utiliza una CA intermedia, `import-key` vuelva a llamar, pero especifique `TrustedPublicKey` en lugar de `RootCertificatePublicKey` especificar el ARN de la CA raíz.

Example Importe el certificado de CA raíz

```
$ 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": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0t..."
```

```
    }
}'
```

```
{
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-east-1:111122223333:key/
nqr3c5v2wp6yl8az",
        "KeyAttributes": {
            "KeyUsage": "TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE",
            "KeyClass": "PUBLIC_KEY",
            "KeyAlgorithm": "RSA_4096",
            "KeyModesOfUse": {
                "Verify": true
            }
        },
        "Enabled": true,
        "KeyState": "CREATE_COMPLETE",
        "KeyOrigin": "EXTERNAL"
    }
}
```

Tenga en cuenta las CA que `KeyArn` se utilizarán en el paso de importación.

Paso 6: Obtenga el certificado de firma de KDH del sistema de envío

En un intercambio de TR-34 claves, la parte remitente (el host de distribución de claves o KDH) proporciona su certificado de clave pública para la firma. Este certificado se usa para verificar la firma en el bloque de claves empaquetado durante la importación y garantizar que el material clave provenga de un remitente de confianza. En producción, este certificado normalmente proviene del sistema de envío (como un HSM, una terminal de pago o un sistema de administración de claves) a través de su propio proceso de distribución de certificados.

En este ejemplo, utilizas la criptografía de AWS pagos como remitente y destinatario para que los pasos sean breves. Llame GetParametersForExport para obtener el certificado de firma de KDH que normalmente proporcionaría el sistema de envío.

Example Obtenga los parámetros para la exportación

```
$ aws payment-cryptography get-parameters-for-export \
    --key-material-type "TR34_KEY_BLOCK" \
    --signing-key-algorithm RSA_2048
```

```
{
    "ExportToken": "export-token-k8sfhe2w9rjy4bx6",
    "SigningKeyCertificate": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0t...",
    "SigningKeyCertificateChain": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0t...",
    "SigningKeyAlgorithm": "RSA_2048",
    "ParametersValidUntilTimestamp": "2025-11-01T18:45:31.271000-07:00"
}
```

Tenga en cuenta la `SigningKeyCertificate` letra `ExportToken` y para usarla en el paso de importación.

Paso 7: Importar la clave con BYOCA

Por último, importe la clave TR-34 con su propio CA-signed certificado mediante la `ImportKey` API. Proporcione el certificado de cifrado KRD firmado por su CA externa.

El `RandomNonce` parámetro es obligatorio cuando se utiliza el método de 2 TR-34 pasos. El servicio no genera un nonce, sino un valor hexadecimal aleatorio que tú generas y debe ser el mismo tanto en el lado emisor como en el receptor. Cuando utilices el método 1-pass TR-34 (que se basa en una marca de tiempo en lugar de en una marca temporal), la marca de tiempo no debe estar en el futuro. El servicio no valida ninguna frescura en particular (por ejemplo, dentro de un número determinado de horas), pero puedes hacer que compruebes la frescura por tu cuenta si es necesario.

Example TR-34 Importe con BYOCA

```
$ aws payment-cryptography import-key --key-material='{
    "Tr34KeyBlock": {
        "CertificateAuthorityPublicKeyIdentifier": "arn:aws:payment-cryptography:us-
east-1:111122223333:key/nqr3c5v2wp6yl8az",
        "KeyBlockFormat": "X9_TR34_2012",
        "RandomNonce": "4997FBB4587D571F",
        "SigningKeyCertificate": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0t...",
        "WrappingKeyCertificate": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0t...",
        "WrappingKeyIdentifier": "arn:aws:payment-cryptography:us-
east-1:111122223333:key/bm7t4qv8hzk24jce",
        "WrappedKeyBlock": "D0112B0AX00E0000B82679114F470F540165EDFBF407..."
    }
}'
```

```
{
```

```
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-east-1:111122223333:key/
gwz7u3rfxd4o52sb",
        "KeyAttributes": {
            "KeyUsage": "TR31_K0_KEY_ENCRYPTION_KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyAlgorithm": "TDES_2KEY",
            "KeyModesOfUse": {
                "Encrypt": true,
                "Decrypt": true
            }
        },
        "KeyCheckValue": "3DCA31",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "Enabled": true,
        "Exportable": true,
        "KeyState": "CREATE_COMPLETE",
        "KeyOrigin": "EXTERNAL"
    }
}
```

La clave ya está importada a AWS Payment Cryptography y está lista para usarse. El `KeyOrigin` campo `EXTERNAL` indica que la clave proviene de fuera del servicio.

Notas adicionales

- Estos ejemplos se muestran con la CLI de AWS. La misma funcionalidad está disponible en todos los SDK de AWS, incluidos Java, Python, Go y Rust.
- Si está realizando pruebas con una CA autofirmada, puede usar OpenSSL para crear una CA de prueba y firmar la CSR. En producción, utilice la infraestructura de CA establecida de su organización.

Intercambio de claves físicas

Puede utilizar el intercambio de claves físicas para convertir de forma segura los componentes de claves criptográficas en papel a formato electrónico cuando sus socios o proveedores no admitan el intercambio electrónico de claves. Custodios de AWS claves capacitados llevan a cabo las ceremonias de entrega de llaves en instalaciones AWS-operated seguras con certificación PCI PIN y P2PE, y convierten los componentes de las claves en papel a formato electrónico mediante un HSM sin conexión. El servicio utiliza el intercambio de ECDH-based claves para entregar un bloque de

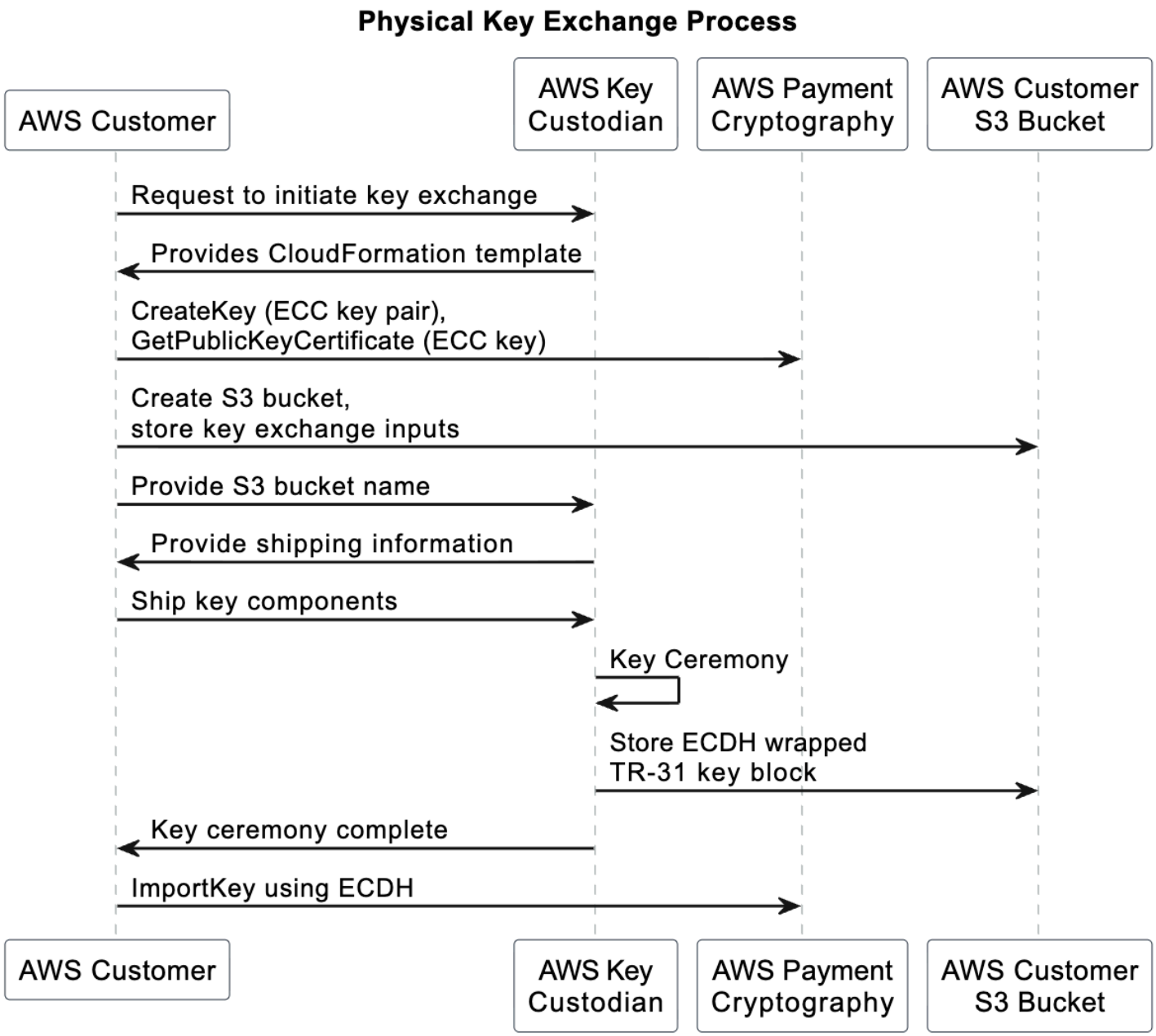
ECDH-wrapped TR-31 claves, que usted importa directamente a su AWS cuenta de criptografía de pagos.

> **Note**
>
> Siempre que sea posible, te recomendamos que utilices un sistema basado en estándaresImportación y exportación de claves. Utilice el intercambio de claves físicas solo cuando sus socios o proveedores no admitan los métodos de intercambio electrónico de claves, como ANSI X9.24 TR-34, RSA wrap/unwrap o ECDH. ???

Cómo funciona el intercambio físico de claves

Para iniciar el intercambio de claves en papel, una CloudFormation plantilla lo guiará a través de la configuración previa, incluida la creación de un par de claves ECC y un depósito S3 en su cuenta. A continuación, usted o su socio envían los componentes de las claves en papel a un centro AWS seguro, donde custodios de AWS claves capacitados llevan a cabo la ceremonia de entrega de las claves utilizando un HSM sin conexión a Internet. El resultado es un bloque de ECDH-wrapped TR-31 claves que se carga en tu bucket de S3 y que importas a tu cuenta mediante este método. Importe claves mediante técnicas asimétricas (ECDH) Physical Key Exchange admite la importación de claves KEK (uso de claves K1) o BDK (uso de claves B0) en los algoritmos de claves TDES y AES.

El siguiente diagrama muestra el proceso integral de intercambio de claves físicas.



1. Inicio: envías un ticket de soporte o trabajas con tu administrador de cuentas para enviar una solicitud.
2. Configuración del cliente: la criptografía de AWS pagos proporciona una CloudFormation plantilla para que complete los siguientes pasos previos:
   - Cree un par de claves ECC P521 en su cuenta de criptografía de AWS pagos y recupere el certificado de clave pública.

- Cree un bucket de Amazon S3 con una política que conceda acceso principal al servicio de criptografía AWS de pagos. read/write
- Guarde el certificado público ECC y la CA raíz de firma en el bucket de Amazon S3.
- Proporcione los atributos clave: el uso de las claves, los modos de uso clave y la cantidad de componentes clave en papel que se enviarán.

3. Nombre del bucket de S3: el cliente comparte el nombre del bucket de S3 creado por la CloudFormation pila, donde se almacenan el certificado de clave pública, la cadena de certificados y los atributos clave para que AWS Payment Cryptography inicie el intercambio de claves.
4. Coordinación de envíos: la criptografía de AWS pagos proporciona los detalles de envío para la instalación US-based segura. Usted o su socio envían los componentes clave en papel a los custodios AWS clave.
5. Recepción de componentes: los custodios AWS clave reciben cada componente en papel y envían un acuse de recibo por separado para cada componente.
6. Ceremonia de entrega de AWS llaves: los custodios clave realizan la ceremonia de entrega de llaves utilizando un HSM fuera de línea. El bloque de TR-31 claves resultante, empaquetado con una ECDH-derived AES-256 clave, el certificado público ECC del HSM sin conexión y su certificado de firma, se cargan en su bucket de Amazon S3.
7. Finalización: la criptografía del AWS pago envía una confirmación de que se ha completado la ceremonia de entrega de claves. A continuación, puede importar el bloque de TR-31 claves envuelto en ECDH a su cuenta de criptografía de AWS pagos utilizando este método. Importe claves mediante técnicas asimétricas (ECDH)
8. Facturación: se le facturará por cada clave intercambiada al completar con éxito la ceremonia de entrega de llaves.

Seguridad y conformidad

Physical Key Exchange funciona en instalaciones AWS seguras diseñadas para cumplir con los requisitos de seguridad física y lógica de PCI PIN y PCI P2PE. Existen los siguientes controles:

Doble control y separación de funciones

AWS Los custodios clave provienen de diferentes equipos con estructuras de información independientes. Existen procesos para garantizar que los pasos clave de las ceremonias se lleven a cabo bajo un doble control.

HSM offline

Las ceremonias clave se llevan a cabo utilizando módulos de seguridad de HSM-listed hardware PCI PTS certificados que funcionan sin conexión a la red. Su clave nunca existe en texto sin cifrar fuera del límite del HSM.

Entrega de claves criptográficas

El material clave se transfiere del HSM fuera de línea a su cuenta de criptografía de AWS pagos mediante el intercambio de ECDH-based claves, lo que garantiza una protección criptográfica de principio a fin.

Auditoría y cumplimiento

AWS cuenta con procesos para cumplir con los requisitos de cumplimiento aplicables, que se evalúan periódicamente para comprobar las certificaciones PCI PIN y P2PE. Revisa el paquete de cumplimiento de AWS Artifact para ver los informes a los que puedas hacer referencia en tus propias evaluaciones de PCI.

Uso de alias

Un alias es un nombre descriptivo para una clave de criptografía AWS de pagos. Por ejemplo, un alias le permite referirse a una clave como `alias/test-key` en lugar de `arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h`.

Puede usar un alias para identificar una clave en la mayoría de las operaciones de administración de claves (plano de control) y en las operaciones criptográficas (plano de datos).

También puedes permitir o denegar el acceso a la clave de criptografía de AWS pagos en función de sus alias sin editar las políticas ni gestionar las concesiones. Esta característica forma parte de la compatibilidad del servicio con el control de acceso basado en atributos (ABAC).

Los alias también facilitan la reutilización del mismo código en diferentes Regiones de AWS. Crea alias con el mismo nombre en varias regiones y asocia cada alias a una clave de criptografía de AWS pagos de su región. Cuando el código se ejecuta en cada región, el alias hace referencia a la clave de criptografía de AWS pagos asociada en esa región.

Puedes crear un alias para una clave de criptografía de AWS pagos mediante la `CreateAlias` API.

La API AWS de criptografía de pagos proporciona un control total de los alias de cada cuenta y región. La API incluye operaciones para crear un alias (CreateAlias), ver los nombres de los alias y

el KeyARN vinculado (alias de lista), cambiar la clave de criptografía de AWS pagos asociada a un alias (update-alias) y eliminar un alias (delete-alias).

Temas

- Acerca de los alias
- Usar alias en las aplicaciones
- API relacionadas

Acerca de los alias

AWS Descubre cómo funcionan los alias en la criptografía de pagos.

Un alias es un recurso independiente AWS

Un alias no es propiedad de una clave de criptografía de AWS pagos. Las acciones que realice en el alias no afectan a su clave asociada. Puede crear un alias para una clave de criptografía de AWS pagos y, a continuación, actualizar el alias para que se asocie a una clave de criptografía de AWS pagos diferente. Incluso puedes eliminar el alias sin que esto afecte a la clave de criptografía de AWS pagos asociada. Si elimina una clave de AWS Payment Cryptography, todos los alias asociados a esa clave quedarán sin asignar.

Si especificas un alias como recurso en una política de IAM, la política hace referencia al alias, no a la clave de criptografía de AWS pagos asociada.

Cada alias tiene un nombre coloquial

Cuando cree un alias, especifique el nombre del alias precedido por `alias/`. Por ejemplo, `alias/test_1234`

Cada alias está asociado a una clave de criptografía AWS de pagos a la vez

El alias y su clave AWS de criptografía de pagos deben estar en la misma cuenta y región.

Una clave de criptografía de AWS pagos se puede asociar a más de un alias al mismo tiempo, pero cada alias solo se puede asignar a una clave

Por ejemplo, esta salida `list-aliases` muestra que el alias `alias/sampleAlias1` está asociado con exactamente una clave de AWS Payment Cryptography de destino, que está representada por la propiedad de `KeyArn`.

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

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

Se pueden asociar varios alias a la misma clave de criptografía de pagos AWS

Por ejemplo, puede asociar los alias `alias/sampleAlias1;` y `alias/sampleAlias2` con la misma clave.

```
$ 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"
            }
        ]
    }
```

El alias debe ser único para una cuenta y región determinadas.

Por ejemplo, solo puede tener un alias `alias/sampleAlias1` en cada cuenta y región. Los alias distinguen entre mayúsculas y minúsculas, pero le recomendamos que no utilice alias que

sólo difieran en las mayúsculas, ya que pueden dar lugar a errores. No puede cambiar un nombre de alias. Sin embargo, puede eliminar el alias y crear un nuevo alias con el nombre deseado.

Puede crear alias con el mismo nombre en diferentes regiones

Por ejemplo, puede tener un alias `alias/sampleAlias2` en el Este de EE. UU. (Norte de Virginia) y un alias `alias/sampleAlias2` en el Oeste de EE. UU. (Oregón). Cada alias se asociaría a una clave de criptografía AWS de pagos en su región. Si su código se refiere a un nombre de alias como `alias/finance-key`, puede ejecutarlo en varias regiones. En cada región, usa una diferente alias/sampleAlias2. Para obtener más información, consulte Usar alias en las aplicaciones.

Puede cambiar la clave de criptografía de AWS pagos asociada a un alias

Puede usar la `UpdateAlias` operación para asociar un alias a una clave de criptografía AWS de pagos diferente. Por ejemplo, si el `alias/sampleAlias2` alias está asociado a la clave de criptografía de `arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h` AWS pagos, puede actualizarlo para que esté asociado a la `arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi` clave.

> **Warning**
>
> AWS La criptografía de pagos no valida que la clave antigua y la nueva tengan los mismos atributos, como el uso de la clave. La actualización con un tipo de clave diferente puede dar lugar a problemas en su aplicación.

> **Warning**
>
> Las actualizaciones de los alias están sujetas a la posible coherencia. Tras actualizar un alias, hay un breve período durante el cual el alias puede convertirse en la clave antigua o en la nueva. No actualice los alias si este comportamiento interrumpiría sus procesos.

Algunas claves no tienen alias

Un alias es una característica opcional y no todas las claves tendrán alias a menos que elija operar su entorno de esta manera. Las claves pueden asociarse con alias utilizando el comando `create-alias`. Además, puede usar la operación update-alias para cambiar la clave de AWS

Payment Cryptography asociada a un alias y la operación delete-alias para eliminar un alias. Como resultado, algunas claves de criptografía de AWS pagos pueden tener varios alias y otras puede que no tengan ninguno.

Asignación de una clave a un alias

Puede asignar una clave (representada por un ARN) a uno o más alias utilizando el comando `create-alias`. Este comando no es idempotente; para actualizar un alias, utilice el comando 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"
    }
}
```

Usar alias en las aplicaciones

Puedes usar un alias para representar una clave de criptografía AWS de pagos en el código de tu aplicación. El `key-identifier` parámetro de las operaciones de datos de criptografía de AWS pagos, así como de otras operaciones, como las claves de lista, acepta un nombre de alias o un ARN de alias.

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

Cuando utilices un ARN con alias, recuerda que el alias asignado a una clave de criptografía de AWS pagos se define en la cuenta a la que pertenece la clave de criptografía de AWS pagos y puede variar en cada región.

Uno de los usos más potentes de los alias es en aplicaciones que se ejecutan en múltiples Regiones de AWS.

Puedes crear una versión diferente de tu aplicación en cada región o usar un diccionario, una configuración o una sentencia switch para seleccionar la clave de criptografía de AWS pagos adecuada para cada región. Pero puede ser más fácil crear un alias con el mismo nombre de alias en cada región. El nombre del alias distingue entre mayúsculas y minúsculas.

API relacionadas

Etiquetas

Las etiquetas son pares de clave y valor que actúan como metadatos para organizar las claves de criptografía AWS de pagos. Pueden utilizarse para identificar claves de forma flexible o agrupar una o varias claves.

Obtener claves

Una clave AWS de criptografía de pagos representa una sola unidad de material criptográfico y solo se puede usar para las operaciones criptográficas de este servicio. La GetKeys API toma una KeyIdentifier como entrada y devuelve los metadatos clave, incluidos los atributos, el estado y las marcas de tiempo, pero no devuelve el material de la clave criptográfica real.

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"
  }
}
```

Consiga que el público se key/certificate asocie a un par de claves

Get Public Key/Certificate devuelve la clave pública indicada por`KeyArn`. Puede ser la parte de clave pública de un par de claves generado en la criptografía de AWS pagos o una clave pública importada anteriormente. El caso de uso más común es proporcionar la clave pública a un servicio externo que cifrará datos. Luego, esos datos se pueden transferir a una aplicación que utilice la criptografía de AWS pagos y los datos se pueden descifrar con la clave privada protegida en la criptografía de pagos. AWS

El servicio devuelve las claves públicas como un certificado público. El resultado de la API contiene la CA y el certificado de clave pública. Ambos elementos de datos están codificados en base64.

> **Note**
>
> El certificado público devuelto está pensado para ser de corta duración y no pretende ser idempotente. Es posible que reciba un certificado diferente en cada llamada a la API, aunque la clave pública en sí no cambie.

Example

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

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

Etiquetado de claves

En la criptografía de AWS pagos, puedes añadir etiquetas a una clave de criptografía de AWS pagos al crear una clave y etiquetar o desetiquetar las claves existentes, a menos que estén pendientes de ser eliminadas. Las etiquetas son opcionales, pero pueden ser muy útiles.

Para obtener información general sobre las etiquetas, incluidas las prácticas recomendadas, las estrategias de etiquetado y el formato y la sintaxis de las etiquetas, consulte los recursos para AWS etiquetar en. Referencia general de Amazon Web Services

Temas

- Acerca de las etiquetas en AWS Payment Cryptography
- Visualización de etiquetas clave en la consola
- Administración de etiquetas de clave con operaciones de la API
- Control del acceso a las etiquetas
- Uso de etiquetas para controlar el acceso a las claves

Acerca de las etiquetas en AWS Payment Cryptography

Una etiqueta es una etiqueta de metadatos opcional que puede asignar (o AWS puede asignar) a un AWS recurso. Cada etiqueta consta de una clave de etiqueta y a valor de etiqueta, que distinguen entre mayúsculas y minúsculas. El valor de la etiqueta puede ser una cadena vacía (nula). Cada etiqueta de un recurso debe tener una clave de etiqueta diferente, pero puedes agregar la misma etiqueta a varios AWS recursos. Cada recurso puede tener un máximo de 50 etiquetas creadas por el usuario.

No incluya información confidencial en la clave ni en el valor de la etiqueta. Muchas personas pueden acceder a las etiquetas Servicios de AWS, incluida la facturación.

En la criptografía de AWS pagos, puedes añadir etiquetas a una clave al crear la clave y etiquetar o desetiquetar las claves existentes, a menos que estén pendientes de ser eliminadas. No puede etiquetar alias. Las etiquetas son opcionales, pero pueden ser muy útiles.

Por ejemplo, puede añadir una `"Project"="Alpha"` etiqueta a todas las claves de criptografía de AWS pagos y a los buckets de Amazon S3 que utilice para el proyecto Alpha. Otro ejemplo es añadir una etiqueta `"BIN"="20130622"` a todas las claves asociadas a un número de identificación bancaria (BIN) específico.

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

Para obtener información general sobre las etiquetas, incluidos el formato y la sintaxis, consulte AWS los recursos de etiquetado en. Referencia general de Amazon Web Services

Las etiquetas lo ayudan a hacer lo siguiente:

- Identifique y organice sus AWS recursos. Muchos AWS servicios admiten el etiquetado, por lo que puede asignar la misma etiqueta a los recursos de diferentes servicios para indicar que los recursos están relacionados. Por ejemplo, puede asignar la misma etiqueta a una clave de criptografía de AWS pagos y a un volumen o secreto de Amazon Elastic Block Store (Amazon EBS). AWS Secrets Manager También puede utilizar etiquetas para identificar claves para la automatización.
- Realice un seguimiento de sus costos. AWS Cuando agregas etiquetas a tus AWS recursos, AWS genera un informe de asignación de costos con el uso y los costos agregados por etiquetas. Puede usar esta función para realizar un seguimiento de los costos de la criptografía de AWS pagos de un proyecto, una aplicación o un centro de costos.

  Para obtener más información sobre el uso de etiquetas para la asignación de costos, consulte Uso de etiquetas de asignación de costos en la Guía del usuario de AWS Billing . Para obtener información sobre las reglas para las claves y los valores de las etiquetas, consulte las restricciones de User-Defined etiquetas en la Guía del AWS Billing usuario.
- Controla el acceso a tus AWS recursos. Permitir y denegar el acceso a las claves en función de sus etiquetas forma parte del soporte de la criptografía de AWS pagos para el control de acceso basado en atributos (ABAC). Para obtener más información sobre el control de acceso a AWS Payment Cryptography basado en etiquetas, consulte Autorización basada en AWS Etiquetas de criptografía de pago. Para obtener más información general sobre el uso de etiquetas para

controlar el acceso a AWS los recursos, consulte Cómo controlar el acceso a los AWS recursos mediante etiquetas de recursos en la Guía del usuario de IAM.

AWS La criptografía de pagos escribe una entrada en su AWS CloudTrail registro cuando utiliza las operaciones TagResource UntagResource, o ListTagsForResource .

Visualización de etiquetas clave en la consola

Para ver las etiquetas en la consola, necesita permiso de etiquetado en la clave desde una política de IAM que incluya la clave. Necesita estos permisos además de los permisos para ver las claves en la consola.

Administración de etiquetas de clave con operaciones de la API

Puede utilizar la API de AWS Payment Cryptography para agregar, eliminar y enumerar etiquetas para las claves que administre. En estos ejemplos, se utiliza la AWS Command Line Interface (AWS CLI), pero puede usar cualquier lenguaje de programación admitido. No puedes etiquetar Claves administradas de AWS.

Para agregar, editar, ver y eliminar etiquetas de una clave, debe tener los permisos necesarios. Para obtener más información, consulte Control del acceso a las etiquetas.

Temas

- CreateKey: Agregar etiquetas a una clave nueva
- TagResource: Agregar o cambiar las etiquetas de una clave
- ListResourceTags: Obtenga las etiquetas de una clave
- UntagResource: Eliminar las etiquetas de una clave

CreateKey: Agregar etiquetas a una clave nueva

Puede añadir etiquetas cuando cree una llave. Para especificar las etiquetas, utilice el `Tags` parámetro de la CreateKey operación.

Para agregar etiquetas al crear una clave, la persona que llama debe tener el permiso `payment-cryptography:TagResource` en una política de IAM. Como mínimo, el permiso debe cubrir todas las claves de la cuenta y la región. Para obtener más información, consulte Control del acceso a las etiquetas.

El valor del parámetro `Tags` de `CreateKey` es una colección de pares de claves y valores de etiqueta que distinguen mayúsculas y minúsculas. Cada etiqueta de una clave debe tener un nombre de etiqueta diferente. El valor de etiqueta puede ser una cadena vacía o nula.

Por ejemplo, el siguiente AWS CLI comando crea una clave de cifrado simétrica con una `Project:Alpha` etiqueta. Cuando especifique más de un par de clave-valor, utilice un espacio para separar cada par.

```
$ 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"}]'
```

Cuando este comando se ejecuta correctamente, devuelve un objeto `Key` con información sobre la nueva clave. Sin embargo, `Key` no incluye etiquetas. Para obtener las etiquetas, utilice la ListResourceTags operación.

TagResource: Agregar o cambiar las etiquetas de una clave

La TagResource operación agrega una o más etiquetas a una clave. No puede usar esta operación para agregar o editar etiquetas en una Cuenta de AWS diferente.

Para agregar una etiqueta, especifique una clave de etiqueta nueva y un valor de la etiqueta. Para editar una etiqueta, especifique una clave de etiqueta existente y un nuevo valor de etiqueta. Cada etiqueta de una clave debe tener una clave de etiqueta distinta. El valor de etiqueta puede ser una cadena vacía o nula.

Por ejemplo, el siguiente comando agrega las etiquetas **`UseCase`** y **`BIN`** a una clave de ejemplo.

```
$ 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"}]'
```

Si este comando se realiza correctamente, no devuelve ningún resultado. Para ver las etiquetas de una clave, utilice la ListResourceTags operación.

También pueden utilizar TagResource para cambiar los valores de una etiqueta existente. Para sustituir los valores de etiqueta, especifique la misma clave de etiqueta con distintos valores. Las etiquetas no listadas en un comando de modificación no se cambian ni se eliminan.

Por ejemplo, este comando cambia el valor de la etiqueta `Project` de `Alpha` a `Noe`.

El comando volverá http/200 sin contenido. Para ver los cambios, utilice `ListTagsForResource`

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

ListResourceTags: Obtenga las etiquetas de una clave

La ListResourceTags operación obtiene las etiquetas de una clave. El parámetro `ResourceArn` (keyArn o keyAlias) es obligatorio. No puede usar esta operación para ver las etiquetas de claves en una Cuenta de AWS diferente.

Por ejemplo, el comando siguiente obtiene las etiquetas para una clave de ejemplo.

```
$ 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: Eliminar las etiquetas de una clave

La UntagResource operación elimina las etiquetas de una clave. Para identificar las etiquetas que desea eliminar, especifique las claves de etiqueta. No puede usar esta operación para eliminar etiquetas de claves una Cuenta de AWS diferente.

Cuando tiene éxito, la operación `UntagResource` no devuelve ningún resultado. Además, si la clave de etiqueta especificada no se encuentra en la clave, no arroja una excepción ni devuelve una respuesta. Para confirmar que la operación ha funcionado, utilícela ListResourceTags.

Por ejemplo, este comando elimina la etiqueta **`Purpose`** y todos sus valores de la clave especificada.

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

Control del acceso a las etiquetas

Para agregar, ver y eliminar etiquetas mediante el uso de la API, las entidades principales necesitan permisos de etiquetado en la política de IAM.

También puede limitar estos permisos mediante el uso de claves de condición AWS globales para las etiquetas. En la criptografía de AWS pagos, estas condiciones pueden controlar el acceso a las operaciones de etiquetado, como TagResource y. UntagResource

Para obtener más información y políticas de ejemplo, consulte Control del acceso en función de las claves de etiquetaen la Guía del usuario de IAM.

Los permisos para crear y administrar etiquetas funcionan de la siguiente manera.

criptografía de pagos: TagResource

Permite a las entidades principales agregar o editar etiquetas. Para agregar etiquetas al crear una clave, la entidad principal debe tener permiso en una política de IAM que no esté restringida a determinadas claves.

criptografía de pago: ListTagsForResource

Permite a las entidades principales ver etiquetas en claves.

criptografía de pago: UntagResource

Permite a las entidades principales eliminar etiquetas de las claves.

Permisos de etiquetas en políticas

Puede proporcionar permisos de etiquetas en una política de claves o una política de IAM. Por ejemplo, la siguiente política de claves de ejemplo ofrece permiso de etiquetar a los usuarios seleccionados en la clave. Da permiso a todos los usuarios que pueden asumir los roles de administrador o desarrollador de ejemplo para ver etiquetas.

JSON

```
{
  "Version":"2012-10-17",
  "Id": "example-key-policy",
  "Statement": [
    {
      "Sid": "EnableIAMUserPermissions",
      "Effect": "Allow",
      "Principal": {"AWS": "arn:aws:iam::111122223333:root"},
      "Action": "payment-cryptography:*",
      "Resource": "*"
    },
    {
      "Sid": "AllowAllTaggingPermissions",
      "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:ListTagsForResource",
      "Resource": "*"
    }
  ]
}
```

Para conceder permiso de etiquetado de entidades principales en varias claves, puede usar una política de IAM. Para que esta política sea efectiva, la política de claves de cada clave debe permitir a la cuenta utilizar políticas de IAM para controlar el acceso a clave.

Por ejemplo, la siguiente política de IAM permite a las entidades principales crear claves. También les permite crear y administrar etiquetas en todas las claves de la cuenta especificada. Esta combinación permite a los directores utilizar el parámetro tags de la CreateKey operación para añadir etiquetas a una clave mientras la crean.

JSON

```
{
  "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/*"
    }
  ]
}
```

Limitar los permisos de etiqueta

Puede limitar los permisos de etiquetado mediante condiciones de política. Las siguientes condiciones de política se pueden aplicar a los permisos `payment-cryptography:TagResource` y `payment-cryptography:UntagResource`. Por ejemplo, puede utilizar la condición `aws:RequestTag/`*`tag-key`* para permitir que una entidad principal agregue solo etiquetas particulares, o impedir que una entidad principal agregue etiquetas con claves de etiqueta concretas.

- leyes: RequestTag
- aws:ResourceTag/tag-key (solo políticas de IAM)
- leyes: TagKeys

Como práctica recomendada cuando utilice etiquetas para controlar el acceso a claves, utilice la clave de condición `aws:RequestTag/`*`tag-key`* o `aws:TagKeys` para determinar qué etiquetas (o claves de etiqueta) están permitidas.

Por ejemplo, la siguiente política IAM es similar a la anterior. Sin embargo, esta política permite a las entidades principales crear etiquetas (`TagResource`) y eliminar etiquetas `UntagResource` solo para etiquetas con una clave de etiqueta `Project`.

Dado que `TagResource` las `UntagResource` solicitudes pueden incluir varias etiquetas, debe especificar un operador `ForAllValues` o `ForAnyValue` establecer un operador con la TagKeys condición aws:. El operador `ForAnyValue` requiere que al menos una de las claves de etiqueta de la solicitud coincida con una de las claves de etiqueta de la política. El operador `ForAllValues` requiere que todas las claves de etiqueta de la solicitud coincidan con una de las claves de etiqueta de la política. El `ForAllValues` operador también devuelve `true` si no hay etiquetas en la solicitud, pero TagResource UntagResource falla cuando no se especifica ninguna etiqueta. Para obtener información detallada sobre los operadores de conjunto, consulte Usar varias claves y valores en la Guía del usuario de IAM.

JSON

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

Uso de etiquetas para controlar el acceso a las claves

Puedes controlar el acceso a la criptografía de AWS pagos en función de las etiquetas de la clave. Por ejemplo, puede escribir una política de IAM que permita a las entidades principales habilitar y desactivar solo las claves que tienen una etiqueta concreta. O bien, puede utilizar una política de IAM para evitar que las entidades principales utilicen claves en operaciones criptográficas, a menos que la clave tenga una etiqueta concreta.

Esta función forma parte del soporte de la criptografía de AWS pagos para el control de acceso basado en atributos (ABAC). Para obtener información sobre el uso de etiquetas para controlar el acceso a AWS los recursos, consulte ¿Para qué sirve ABAC? AWSy cómo controlar el acceso a AWS los recursos mediante etiquetas de recursos en la guía del usuario de IAM.

AWS La criptografía de pagos admite la clave de contexto de condición global aws:ResourceTag/tag-key, que permite controlar el acceso a las claves en función de las etiquetas de la clave. Dado que varias claves pueden tener la misma etiqueta, esta función le permite aplicar el permiso a un conjunto seleccionado de claves. También puede cambiar fácilmente las claves del conjunto cambiando sus etiquetas.

En la criptografía AWS de pagos, la clave de `aws:ResourceTag/`*`tag-key`* condición solo se admite en las políticas de IAM. No se admite en las políticas de claves, que se aplican solo a una clave, ni en las operaciones que no utilizan una clave determinada, como las operaciones ListKeys o ListAliases.

Controlar el acceso con etiquetas proporciona una forma sencilla, escalable y flexible de administrar los permisos. Sin embargo, si no está diseñado y administrado correctamente, puede permitir o denegar el acceso a sus claves inadvertidamente. Si utiliza etiquetas para controlar el acceso, tenga en cuenta las siguientes prácticas.

- Utilice etiquetas para reforzar la práctica recomendada de acceso menos privilegiado. Proporcione a las entidades principales de IAM solo los permisos que necesitan y únicamente en las claves de que deben usar o administrar. Por ejemplo, utilice etiquetas para etiquetar las claves utilizadas en un proyecto. A continuación, dé permiso al equipo del proyecto para usar solo claves con la etiqueta de proyecto.
- Tenga cuidado al dar a las entidades principales los permisos `payment-cryptography:TagResource` y `payment-cryptography:UntagResource` que les permiten agregar, editar y eliminar etiquetas. Cuando utiliza etiquetas para controlar el acceso a las claves, cambiar una etiqueta puede dar permiso a las entidades principales para usar claves que de otro modo no tenían permiso para usar. También puede denegar el acceso a las claves que otras entidades principales requieren para realizar sus trabajos. Los administradores de claves que no tienen permiso para cambiar políticas de claves o crear concesiones pueden controlar el acceso a claves si tienen permiso para administrar etiquetas.

  Siempre que sea posible, utilice una condición de política, como `aws:RequestTag/`*`tag-key`* o `aws:TagKeys` para limitar los permisos de etiquetado de una entidad principal a determinadas etiquetas o patrones de etiquetas en determinadas claves.
- Revisa las principales de tu cuenta Cuenta de AWS que actualmente tienen permisos para etiquetar y desetiquetar y ajústalas, si es necesario. Las políticas de IAM pueden habilitar permisos de etiqueta y desetiqueta en todas las claves. Por ejemplo, la política Admin permite a las entidades principales administradas etiquetar, desetiquetar y generar un lista de etiquetas en todas las claves.
- Antes de establecer una política que dependa de una etiqueta, revisa las etiquetas de las claves de tu. Cuenta de AWS Asegúrese de que su política solo se aplique a las etiquetas que desea incluir. Usa CloudTrail los registros y CloudWatch las alarmas para avisarte de los cambios en las etiquetas que puedan afectar al acceso a tus claves.
- Las condiciones de política basadas en etiquetas utilizan la coincidencia de patrones; no están vinculadas a una instancia concreta de una etiqueta. Una política que utiliza claves de condición basadas en etiquetas afecta a todas las etiquetas nuevas y existentes que coincidan con el patrón. Si elimina y vuelve a crear una etiqueta que coincida con una condición de política, la condición se aplica a la nueva etiqueta, igual que a la anterior.

Por ejemplo, tomemos el siguiente ejemplo de política de IAM. Permite a las entidades principales llamar a las operaciones de Descifrado sólo en claves de su cuenta que sean de la región Este de EE. UU. (Norte de Virginia) y tengan una etiqueta `"Project"="Alpha"`. Puede adjuntar esta política a roles del ejemplo de proyecto Alpha.

JSON

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

La siguiente política de IAM de ejemplo permite a la entidad principal utilizar la clave en la cuenta para operaciones criptográficas. Pero prohíbe a las entidades principales usar estas operaciones criptográficas en claves con una etiqueta `"Type"="Reserved"` o sin etiqueta `"Type"`.

JSON

```
{
  "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"
        }
      }
    }
  ]
}
```

Descripción de los atributos clave de AWS Clave de criptografía de pagos

Un principio de la gestión adecuada de claves es que éstas tengan un alcance apropiado y sólo puedan utilizarse para operaciones permitidas. Como tal, ciertas claves sólo pueden crearse con ciertos modos de uso. Siempre que sea posible, se alinea con los modos de uso disponibles según lo definido por. TR-31

Si bien la criptografía de AWS pagos evitará que cree claves no válidas, aquí se proporcionan combinaciones válidas para su comodidad.

Claves simétricas

- TR31_B0_BASE_DERIVATION_KEY
  - Algoritmos de clave permitidos: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Combinación permitida de modos de uso clave: { DeriveKey = true}, { NoRestrictions = true}
- TR31_C0_CARD_VERIFICATION_KEY
  - Algoritmos de claves permitidos: TDES_2KEY, TDES_3KEY, AES_128*, AES_192*, AES_256*
  - Combinación permitida de modos de uso clave: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, {= true} NoRestrictions
- TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY
  - Algoritmos de clave permitidos: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Combinación permitida de modos de uso clave: {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}, {= true} NoRestrictions
- TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS
  - Algoritmos clave permitidos: TDES_2KEY, TDES_3KEY*, AES_128*, AES_192*, AES_256*
  - Combinación permitida de modos de uso clave: {= true DeriveKey }, {= true} NoRestrictions
- TR31_E1_EMV_MKEY_CONFIDENTIALITY
  - Algoritmos de claves permitidos: TDES_2KEY, TDES_3KEY, AES_128*, AES_192*, AES_256*
  - Combinación permitida de modos de uso clave: {= DeriveKey true}, {= true} NoRestrictions
- TR31_E2_EMV_MKEY_INTEGRITY
  - Algoritmos de claves permitidos: TDES_2KEY, TDES_3KEY, AES_128*, AES_192*, AES_256*

  - Combinación permitida de modos de uso clave: {= DeriveKey true}, {= true} NoRestrictions
- TR31_E4_EMV_MKEY_DYNAMIC_NUMBERS
  - Algoritmos de claves permitidos: TDES_2KEY, TDES_3KEY, AES_128*, AES_192*, AES_256*
  - Combinación permitida de modos de uso clave: {= DeriveKey true}, {= true} NoRestrictions
- TR31_E5_EMV_MKEY_CARD_PERSONALIZATION
  - Algoritmos de claves permitidos: TDES_2KEY, TDES_3KEY, AES_128*, AES_192*, AES_256*
  - Combinación permitida de modos de uso clave: {= DeriveKey true}, {= true} NoRestrictions
- TR31_E6_EMV_MKEY_OTHER
  - Algoritmos de claves permitidos: TDES_2KEY, TDES_3KEY, AES_128*, AES_192*, AES_256*
  - Combinación permitida de modos de uso clave: {= DeriveKey true}, {= true} NoRestrictions
- TR31_K0_KEY_ENCRYPTION_KEY
  - Se recomienda utilizar TR31_K1_KEY_BLOCK_PROTECTION_KEY. Algoritmos de clave permitidos: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Combinación permitida de modos de uso clave: {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {= true} NoRestrictions
- TR31_K1_KEY_BLOCK_PROTECTION_KEY
  - Algoritmos de clave permitidos: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Combinación permitida de modos de uso clave: {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}, { NoRestrictions = true}
- TR31_M1_ISO_9797_1_MAC_KEY
  - Algoritmos clave permitidos: TDES_2KEY, TDES_3KEY
  - Combinación permitida de modos de uso clave: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, {= true} NoRestrictions
- TR31_M3_ISO_9797_3_MAC_KEY
  - Algoritmos clave permitidos: TDES_2KEY, TDES_3KEY
  - Combinación permitida de modos de uso clave: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, {= true} NoRestrictions
- TR31_M6_ISO_9797_5_CMAC_KEY
  - Algoritmos de clave permitidos: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256

- Combinación permitida de modos de uso clave: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, {= true} NoRestrictions

- TR31_M7_HMAC_KEY
  - Algoritmos de clave permitidos: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Combinación permitida de modos de uso clave: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, {= true} NoRestrictions
- TR31_P0_PIN_ENCRYPTION_KEY
  - Algoritmos de clave permitidos: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Combinación permitida de modos de uso clave: {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}, {= true} NoRestrictions
- TR31_V1_IBM3624_PIN_VERIFICATION_KEY
  - Algoritmos de clave permitidos: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Combinación permitida de modos de uso clave: {Generate = true}, {Verify = true}, {Generate = true, Verify= true} NoRestrictions
- TR31_V2_VISA_PIN_VERIFICATION_KEY
  - Algoritmos de clave permitidos: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
  - Combinación permitida de modos de uso clave: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, {= true} NoRestrictions

Claves asimétricas

- TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION
  - Algoritmos de clave permitidos: RSA_2048, RSA_3072, RSA_4096
  - Combinación permitida de modos de uso clave: {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}, {NoRestrictions = true}
  - NOTA: {Encrypt = true, Wrap = true} es la única opción válida al importar una clave pública destinada a cifrar datos o empaquetar una clave
- TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE
  - Algoritmos de clave permitidos: RSA_2048, RSA_3072, RSA_4096
  - Combinación permitida de modos de uso clave: {Sign = true}, {Verify = true}

  - NOTA: {Verify = true} es la única opción válida al importar una clave destinada a la firma, como un certificado raíz, un certificado intermedio o un certificado de firma para TR-34.
- TR31_K3_ASYMMETRIC_KEY_FOR_KEY_AGREEMENT
  - Se utiliza para algoritmos de acuerdo de claves, como el ECDH
  - Algoritmos de claves permitidos: ECC_NIST_P256, ECC_NIST_P384, ECC_NIST_P521
  - Combinación permitida de modos de uso clave: {= true}. DeriveKey
  - NOTA: DeriveKeyUsage se usa para especificar qué tipo de clave se derivará de esta clave base. Esto se corrige en la clave creation/import.
- TR31_K2_TR34_ASYMMETRIC_KEY
  - Clave asimétrica utilizada para mecanismos de intercambio de claves compatibles, como X9.24 TR-34
  - Algoritmos de claves permitidos: RSA_2048, RSA_3072, RSA_4096
  - Combinación permitida de modos de uso clave: {= true}. DeriveKey
  - Combinación permitida de modos de uso clave: {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}, {NoRestrictions = true}
  - NOTA: {Encrypt = true, Wrap = true} es la única opción válida al importar una clave pública destinada a cifrar datos o empaquetar una clave

* Actualmente, ninguna operación criptográfica admite esta combinación de algorithm/key tipos

Operaciones de datos

Una vez establecida una clave de criptografía de AWS pago, se puede utilizar para realizar operaciones criptográficas. Las diferentes operaciones realizan diferentes tipos de actividad, desde el cifrado y el hash hasta algoritmos específicos del dominio, como la generación de CVV2.

Los datos cifrados no se pueden descifrar sin la clave de descifrado correspondiente (la clave simétrica o la clave privada, según el tipo de cifrado). Del mismo modo, los algoritmos de hash y los algoritmos específicos del dominio no se pueden verificar sin la clave simétrica o la clave pública.

Para obtener información sobre los tipos de clave válidos para operaciones específicas, consulte Claves válidas para operaciones criptográficas

> **Note**
>
> Recomendamos utilizar datos de prueba en un entorno que no sea de producción. El uso de claves y datos de producción (PAN, ID de BDK, etc.) en un entorno ajeno a la producción puede afectar al ámbito de cumplimiento, por ejemplo, en el caso de PCI DSS y PCI P2PE.

Temas

- Cifrar, descifrar y volver a cifrar datos
- Generación y verificación de datos de tarjetas
- Generar, traducir y verificar los datos del PIN
- Generar y verificar el ARQC
- Generar y verificar MAC
- Claves válidas para las operaciones criptográficas

Cifrar, descifrar y volver a cifrar datos

Los métodos de cifrado y descifrado se pueden utilizar para cifrar o descifrar datos mediante una variedad de técnicas simétricas y asimétricas, incluidas TDES, AES y RSA. Estos métodos también admiten claves derivadas mediante las técnicas DUKPT y EMV. Para los casos de uso en los que desee proteger los datos con una clave nueva sin exponer los datos subyacentes, también se puede utilizar el ReEncrypt comando.

> **Note**
>
> Al utilizar las encrypt/decrypt funciones, se supone que todas las entradas están en HexBinary; por ejemplo, un valor de 1 se introducirá como 31 (hexadecimal) y una t minúscula se representará como 74 (hexadecimal). Todas las salidas también están en hexBinary.

Para obtener más información sobre todas las opciones disponibles, consulte la Guía de la API para cifrar, descifrar y volver a cifrar.

Temas

- Cifrar datos
- Descifrado de datos

Cifrar datos

La `Encrypt Data` API se utiliza para cifrar datos mediante claves de cifrado de datos simétricas y asimétricas, así como claves derivadas de DUKPT y EMV. Se admiten varios algoritmos y variaciones, incluidos `TDES`, `RSA` y `AES`.

Las entradas principales son la clave de cifrado utilizada para cifrar los datos, los datos de texto simple en formato HexBinary que se van a cifrar y los atributos de cifrado, como el vector de inicialización y el modo para los cifrados por bloques, como el TDES. Los datos en texto plano deben estar en múltiplos de 8 bytes para`TDES`, 16 bytes para `AES` y la longitud de la clave en el caso de. `RSA` Las entradas clave simétricas (TDES, AES, DUKPT, EMV) deben rellenarse en los casos en que los datos de entrada no cumplan estos requisitos. La siguiente tabla muestra la longitud máxima del texto sin formato para cada tipo de clave y el tipo de relleno que se define para las claves RSA. `EncryptionAttributes`

Tipo de relleno	RSA_2048	RSA_3072	RSA_4096
`OAEP SHA1`	4.28	684	940
`OAEP SHA256`	380	636	892
`OAEP SHA512`	252	508	764

Tipo de relleno	RSA_2048	RSA_3072	RSA_4096
PKCS1	488	744	1 000
None	488	744	1 000

Las salidas primarias incluyen los datos encriptados como texto cifrado en formato hexBinario y el valor de la suma de comprobación de la clave de encriptación. Para obtener más información sobre todas las opciones disponibles, consulte la Guía de API para Encrypt.

Ejemplos

- Cifrar datos utilizando la clave simétrica AES
- Cifrar los datos con la clave DUKPT
- Cifre los datos mediante una clave simétrica derivada de EMV
- Cifrado de los datos utilizando una clave de RSA

Cifrar datos utilizando la clave simétrica AES

> **Note**
>
> En todos los ejemplos se asume que la clave correspondiente ya existe. Las claves se pueden crear mediante la CreateKeyoperación o importar mediante la ImportKeyoperación.

Example

En este ejemplo, cifraremos los datos en texto plano mediante una clave simétrica que se creó mediante la CreateKeyoperación o se importó mediante la operación. ImportKey Para esta operación, la clave debe estar configurada en `Encrypt` y KeyModesOfUse KeyUsage establecida en. `TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY` Consulte Claves para operaciones criptográficas para ver más opciones.

```
$ aws payment-cryptography-data encrypt-data --key-identifier arn:aws:payment-
cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi --plain-text
 313233343132333431323334313233343 --encryption-attributes 'Symmetric={Mode=CBC}'
```

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

Cifrar los datos con la clave DUKPT

Example

En este ejemplo, cifraremos los datos en texto plano mediante una clave DUKPT. AWS Soportes de criptografía de pagos y claves DUKPT. `TDES AES` Para esta operación, la clave debe estar configurada en `DeriveKey` y KeyModesOfUse KeyUsage configurada en. `TR31_B0_BASE_DERIVATION_KEY` Consulte Claves para operaciones criptográficas para ver más opciones.

```
$ aws payment-cryptography-data encrypt-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi
 --plain-text 313233343132333431323334313233343 --encryption-attributes
 'Dukpt={KeySerialNumber=FFFF9876543210E00001}'
```

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

Cifre los datos mediante una clave simétrica derivada de EMV

Example

En este ejemplo, cifraremos los datos de texto no cifrado mediante una clave simétrica derivada de EMV que ya se ha creado. Puede utilizar un comando como este para enviar datos a una tarjeta EMV. Para esta operación, la clave debe estar KeyModesOfUse configurada en `Derive` y KeyUsage establecida en `TR31_E1_EMV_MKEY_CONFIDENTIALITY` o`TR31_E6_EMV_MKEY_OTHER`. Consulte Claves para operaciones criptográficas para obtener más información.

```
$ 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"
}
```

Cifrado de los datos utilizando una clave de RSA

Example

En este ejemplo, cifraremos los datos en texto plano mediante una clave pública RSA que se importó mediante la operación. ImportKey Para esta operación, la clave debe estar configurada en `Encrypt` y KeyModesOfUse KeyUsage establecida en. `TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION` Consulte Claves para operaciones criptográficas para ver más opciones.

Para PKCS #7 u otros esquemas de relleno no admitidos actualmente, solicítelos antes de llamar al servicio y seleccione sin relleno omitiendo el indicador de relleno 'Asymmetric={}'

```
$ aws payment-cryptography-data encrypt-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/thfezpmsalcfwmsg
 --plain-text 31323334313233343132333431323334 --encryption-attributes
 'Asymmetric={PaddingType=OAEP_SHA256}'
```

```
{
 "CipherText":
 "12DF6A2F64CC566D124900D68E8AFEAA794CA819876E258564D525001D00AC93047A83FB13 \
 E73F06329A100704FA484A15A49F06A7A2E55A241D276491AA91F6D2D8590C60CDE57A642BC64A897F4832A3930
 \
 0FAEC7981102CA0F7370BFBF757F271EF0BB2516007AB111060A9633D1736A9158042D30C5AE11F8C5473EC70F067
 \
 72590DEA1638E2B41FAE6FB1662258596072B13F8E2F62F5D9FAF92C12BB70F42F2ECDCF56AADF0E311D4118FE3591
 \
 FB672998CCE9D00FFFE05D2CD154E3120C5443C8CF9131C7A6A6C05F5723B8F5C07A4003A5A6173E1B425E2B5E42AD
 \
 7A2966734309387C9938B029AFB20828ACFC6D00CD1539234A4A8D9B94CDD4F23A",
 "KeyArn": "arn:aws:payment-cryptography:us-east-1:111122223333:key/5dza7xqd6soanjtb",
 "KeyCheckValue": "FF9DE9CE"
}
```

Descifrado de datos

La `Decrypt Data` API se utiliza para descifrar datos mediante claves de cifrado de datos simétricas y asimétricas, así como claves derivadas de DUKPT y EMV. Se admiten varios algoritmos y variaciones, incluidos `TDES`, `RSA` y `AES`.

Las entradas principales son la clave de descifrado utilizada para descifrar los datos, los datos del texto cifrado en formato hexBinario que deben descifrarse y los atributos de descifrado, como el vector de inicialización, el modo como los cifradores de bloques, etc. Las salidas principales incluyen los datos descifrados como texto plano en formato hexBinario y el valor de la suma de comprobación de la clave de descifrado. Para obtener más información sobre todas las opciones disponibles, consulte la Guía de API para descifrar.

Ejemplos

- Descifrar los datos mediante la clave simétrica AES
- Descifrar los datos con la clave DUKPT
- Descifre los datos mediante una clave simétrica derivada de EMV
- Descifrar datos con una clave de RSA

Descifrar los datos mediante la clave simétrica AES

Example

En este ejemplo, descifraremos los datos de texto cifrado mediante una clave simétrica. En este ejemplo se muestra una `AES` clave, pero también `TDES_2KEY` se admiten. `TDES_3KEY` Para esta operación, la clave debe estar KeyModesOfUse configurada en `Decrypt` y KeyUsage establecida en`TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY`. Consulte Claves para operaciones criptográficas para ver más opciones.

```
$ 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"
}
```

Descifrar los datos con la clave DUKPT

> **Note**
>
> El uso del descifrado de datos con DUKPT para las transacciones P2PE puede devolver a su aplicación el PAN de la tarjeta de crédito y otros datos del titular de la tarjeta que deberán tenerse en cuenta a la hora de determinar su alcance PCI DSS.

Example

En este ejemplo, descifraremos los datos de texto cifrado mediante una clave DUKPT que se creó mediante la CreateKeyoperación o se importó mediante la operación. ImportKey Para esta operación, la clave debe estar establecida en y KeyModesOfUse configurada en. `DeriveKey` KeyUsage `TR31_B0_BASE_DERIVATION_KEY` Consulte Claves para operaciones criptográficas para ver más opciones. Cuando se utiliza `DUKPT`, como algoritmo `TDES`, la longitud de los datos del texto cifrado debe ser un múltiplo de 16 bytes. Para el algoritmo `AES`, la longitud de los datos del texto cifrado debe ser un múltiplo de 32 bytes.

```
$ 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"
}
```

Descifre los datos mediante una clave simétrica derivada de EMV

Example

En este ejemplo, descifraremos los datos de texto cifrado mediante una clave simétrica derivada de EMV que se creó mediante la operación o se importó mediante la operación. CreateKeyImportKey Para esta operación, la clave debe estar establecida en y KeyModesOfUse establecida en o. `Derive` KeyUsage `TR31_E1_EMV_MKEY_CONFIDENTIALITY` `TR31_E6_EMV_MKEY_OTHER` Consulte Claves para operaciones criptográficas para obtener más información.

```
$ 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"
}
```

Descifrar datos con una clave de RSA

Example

En este ejemplo, descifraremos los datos de texto cifrado mediante un par de claves RSA que se creó mediante la operación. CreateKey Para esta operación, la clave debe estar configurada como `Decrypt` habilitada y KeyModesOfUse configurada como. KeyUsage `TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION` Consulte Claves para operaciones criptográficas para ver más opciones.

Para PKCS #7 u otros esquemas de relleno no admitidos actualmente, seleccione sin relleno omitiendo el indicador de relleno 'Asymmetric={}' y elimine el relleno después de llamar al servicio.

```
$ 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-1:111122223333:key/5dza7xqd6soanjtb",
    "KeyCheckValue": "FF9DE9CE",
    "PlainText": "313233343132333431323334313233343132333431323334"
}
```

Generación y verificación de datos de tarjetas

Generar y verificar los datos de la tarjeta incorpora datos derivados de los datos de la tarjeta, por ejemplo CVV CVV2, CVC y DCVV.

Temas

- Generar datos de tarjetas
- Comprobación de datos de tarjetas

Generar datos de tarjetas

La `Generate Card Data` API se utiliza para generar datos de tarjetas mediante algoritmos como CVV o Dynamic. CVV2 CVV2 Para ver qué claves se pueden usar para este comando, consulte la sección Claves válidas para operaciones criptográficas.

Muchos valores criptográficos, como el CVV, CVV2 el CVV o el CVV V7, utilizan el mismo algoritmo criptográfico, pero varían los valores de entrada. Por ejemplo, CardVerificationValue1 tiene entradas como el número de tarjeta y la fecha de caducidad ServiceCode. Si bien CardVerificationValue2 solo tiene dos de estas entradas, esto se debe a que para CVV2/CVC2, el ServiceCode está fijado en 000. Del mismo modo, para iCVV, el valor ServiceCode se fija en 999. Algunos algoritmos pueden reutilizar los campos existentes, como el CAVV V8, en cuyo caso tendrá que consultar el manual del proveedor para obtener los valores de entrada correctos.

> **Note**
>
> La fecha de caducidad debe introducirse en el mismo formato (por ejemplo, MMYY o YYMM) para que la generación y la validación arrojen resultados correctos.

Generar CVV2

Example

En este ejemplo, generaremos un CVV2 para un PAN determinado con entradas de la fecha de PAN caducidad de la tarjeta. Esto supone que se ha generado una clave de verificación de la tarjeta.

```
$ 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"
  }
```

Genera iCVV

Example

En este ejemplo, generaremos un iCVV para un PAN determinado con entradas dePAN, un código de servicio 999 y una fecha de caducidad de la tarjeta. Esto supone que se ha generado una clave de verificación de la tarjeta.

Para ver todos los parámetros disponibles, consulta el apartado CardVerificationValue1 de la guía de referencia de la 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"
}
```

Comprobación de datos de tarjetas

`Verify Card Data` se utiliza para verificar los datos que se han creado mediante algoritmos de pago que se basan en principios de cifrado, como `DISCOVER_DYNAMIC_CARD_VERIFICATION_CODE`.

Los valores de entrada suelen proporcionarse como parte de una transacción entrante a un emisor o a un socio de plataforma de apoyo. Para verificar un criptograma ARQC (utilizado para tarjetas con chips EMV), consulte Verificar el ARQC.

Para obtener más información, consulte VerifyCardValidationDatala guía de la API.

Si se verifica el valor, la API devolverá http/200. Si no se verifica el valor, devolverá http/400.

Verifica CVV2

Example

En este ejemplo, validaremos un CVV/ CVV2 para un PAN determinado. Por lo general, CVV2 lo proporciona el titular de la tarjeta o el usuario durante el momento de la transacción para su validación. Para validar su entrada, se proporcionarán los siguientes valores durante el tiempo de ejecución: la clave que se utilizará para la validación (CVK) y la fecha de caducidad de la tarjetaPAN, y CVV2 se ingresarán. El formato de caducidad de la tarjeta debe coincidir con el utilizado en la generación del valor inicial.

Para ver todos los parámetros disponibles, consulte el CardVerificationValueapartado 2 de la guía de referencia de la 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"
  }
```

Verifica el iCVV

Example

En este ejemplo, verificaremos un iCVV para un PAN determinado con las entradas de la clave que se utilizará para la validación (CVK), un código de servicio 999PAN, la fecha de caducidad de la tarjeta y el iCVV proporcionado por la transacción para la validación.

El iCVV no es un valor introducido por el usuario (por ejemplo CVV2), sino que está incrustado en una tarjeta EMV. Se debe tener en cuenta si siempre debe validarse cuando se proporciona.

Para ver todos los parámetros disponibles, consulte el apartado CardVerificationValue1 de la guía de referencia de la 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
 CardVerificationValue1='{CardExpiryDate=1127,ServiceCode=999} --validation-data 801
```

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

Generar, traducir y verificar los datos del PIN

Las funciones de datos PIN permiten generar códigos PIN al azar, valores de verificación de números PIN (PVV) y validar los pines entrantes cifrados comparándolos con valores PVV o compensaciones de PIN.

La traducción de pines permite convertir un pin de una clave funcional a otra sin exponer el pin en texto claro, tal y como se especifica en el requisito 1 del PIN PCI.

> **Note**
>
> Como la generación y la validación del PIN suelen ser funciones del emisor y la traducción del PIN es una función típica del adquirente, recomendamos que considere el acceso con menos privilegios y establezca las políticas adecuadas para el caso de uso de su sistema.

Temas

- Traducir datos PIN
- Generar datos PIN
- Comprobación de datos PIN

Traducir datos PIN

Las funciones de traducir datos PIN se utilizan para traducir los datos PIN cifrados de un conjunto de claves a otro sin que los datos cifrados salgan del HSM. Se utiliza para el cifrado P2PE, en el que las claves de trabajo deberían cambiar, pero el sistema de procesamiento no necesita descifrar los datos o no está autorizado a hacerlo. Las entradas principales son los datos cifrados, la clave de cifrado utilizada para cifrar los datos y los parámetros utilizados para generar los valores de entrada. El otro conjunto de entradas son los parámetros de salida solicitados, como la clave que se utilizará para cifrar la salida y los parámetros que se utilizarán para crear esa salida. Las salidas principales son un conjunto de datos recién cifrado, así como los parámetros utilizados para generarlo.

> **Note**
>
> Para cumplir con la normativa PCI, los PrimaryAccountNumber valores entrantes y salientes deben coincidir. No está permitido traducir un PIN de un PAN a otro.

Temas

- PIN de PEK a DUKPT
- PIN de PEK a PEK

PIN de PEK a DUKPT

Example

En este ejemplo, traduciremos un PIN de un bloque AES ISO de 4 PIN con el DUKPT al cifrado PEK TDES con un bloque de PIN ISO 0. Esto es habitual cuando un terminal de pago cifra un PIN en ISO 4 y, a continuación, puede volver a traducirlo a TDES para su procesamiento posterior si la siguiente conexión aún no admite el AES.

```
$ aws payment-cryptography-data translate-pin-data --encrypted-pin-block
 "AC17DC148BDA645E" --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": "1F4209C670E49F83E75CC72E81B787D9",
            "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt",
            "KeyCheckValue": "7CC9E2"
        }
```

PIN de PEK a PEK

Example

En este ejemplo, traducimos un PIN cifrado con un PEK (clave de cifrado PIN) a otro PEK. Esto se suele utilizar para enrutar transacciones entre diferentes sistemas o socios que utilizan diferentes claves de cifrado y, al mismo tiempo, mantener el PIN cifrado durante todo el proceso para cumplir con el PIN PCI. En este ejemplo, ambas claves utilizan el cifrado TDES 3KEY, pero hay varias opciones disponibles, como AES ISO-4 a TDES ISO-0, DUKPT a PEK o AS2805 a PEK.

```
$ 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-translation-attributes
 IsoFormat0='{PrimaryAccountNumber=171234567890123}' \
    --outgoing-key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
alsuwfxug3pgy6xh
```

```
{
    "PinBlock": "E8F2A6C4D1B93E7F",
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
alsuwfxug3pgy6xh",
    "KeyCheckValue": "9A325B"
}
```

El bloque PIN de salida ahora está cifrado con el segundo PEK y se puede transmitir de forma segura al sistema posterior que contiene la clave correspondiente.

Generar datos PIN

Las funciones de generación de datos PIN se utilizan para generar PIN-related valores, como el PVV y las compensaciones de bloques de pines que se utilizan para validar la entrada del PIN por parte de los usuarios durante el momento de la transacción o la autorización. Esta API también puede generar un nuevo pin al azar mediante varios algoritmos.

Genere un PIN aleatorio y un código Visa PVV coincidente

Example

En este ejemplo, generaremos un nuevo pin (aleatorio) en el que los resultados serán un () cifrado `PIN block` (PinData.PinBlock) y un PVV (pinData.Offset). Las entradas clave son PAN, `Pin Verification Key`, `Pin Encryption Key` y `PIN block format`.

Este comando requiere que la clave sea de tipo`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"
            }
        }
```

Genere un Visa PVV para un PIN conocido

Example

En este ejemplo, generaremos un PVV para un pin determinado (cifrado). Un PIN cifrado puede recibirse de forma inicial, por ejemplo, desde un terminal de pago o desde el titular de una tarjeta utilizando el flujo de pines que el usuario puede seleccionar. Las entradas clave sonPAN, la Pin Verification KeyPin Encryption Key, la y laEncrypted Pin Block. PIN block format

```
$ 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
 VisaPinVerificationValue={PinVerificationKeyIndex=1,EncryptedPinBlock=AA584CED31790F37}
```

```
{
            "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"
            }
        }
```

Genere el desfase del pin IBM3624 para un pin

El desfase de PIN del IBM 3624 también se denomina a veces método de IBM. Este método genera un natural/intermediate PIN utilizando los datos de validación (normalmente el PAN) y una clave PIN (PVK). Los pines naturales son, en efecto, un valor derivado y, al ser deterministas, son muy eficientes de manejar para un emisor, ya que no es necesario almacenar los datos del PIN a nivel del titular de la tarjeta. La desventaja más obvia es que este esquema no tiene en cuenta los pines aleatorios o seleccionables por el titular de la tarjeta. Para permitir esos tipos de pines, se agregó un algoritmo de compensación al esquema. El desfase representa la diferencia entre el pin seleccionado (o aleatorio) por el usuario y la clave natural. El emisor o procesador de la tarjeta almacena el valor

de compensación. En el momento de la transacción, el servicio de criptografía de AWS pagos recalcula internamente el pin natural y aplica la compensación para encontrar el pin. A continuación, lo compara con el valor proporcionado por la autorización de la transacción.

Existen varias opciones para el IBM3624:

- `Ibm3624NaturalPin`generará el pin natural y un bloque de pines cifrado
- `Ibm3624PinFromOffset`generará un bloque de pines cifrado si se obtiene una compensación
- `Ibm3624RandomPin`generará un pin aleatorio y, a continuación, el bloque de pines cifrado y desfasado correspondiente.
- `Ibm3624PinOffset`genera el desfase del pin dado el pin seleccionado por el usuario.

Como parte de la criptografía de AWS pagos, se llevan a cabo los siguientes pasos:

- Rellene el panel proporcionado con 16 caracteres. Si se proporcionan <16, rellena la parte derecha con el caracter de relleno proporcionado.
- Cifra los datos de validación mediante la clave de generación del PIN.
- Decimalice los datos cifrados mediante la tabla de decimalización. Esto asigna dígitos hexadecimales a dígitos decimales; por ejemplo, «A» puede asignarse a 9 y 1 puede asignarse a 1.
- Obtenga los primeros 4 dígitos de una representación hexadecimal de la salida. Este es el pin natural.
- Si un usuario seleccionó un pin o se generó un pin aleatorio, el módulo restará el pin natural del pin del cliente. El resultado es el desfase del pin.

Ejemplos

- Ejemplo: genere el desfase de polos del IBM3624 para un polo

Ejemplo: genere el desfase de polos del IBM3624 para un polo

En este ejemplo, generaremos un nuevo pin (aleatorio) en el que las salidas serán un valor cifrado `PIN block` (PinData.PinBlock) y un valor de `IBM3624` compensación (pin). Data.Offset Las entradas son los `PAN` datos de validación (normalmente la panorámica), el carácter de relleno, el`Pin Verification Key`, el `Pin Encryption Key` y el`PIN block format`.

Este comando requiere que la clave de generación del pin sea del tipo `TR31_V1_IBM3624_PIN_VERIFICATION_KEY` y que la clave de cifrado sea del tipo `TR31_P0_PIN_ENCRYPTION_KEY`

Example

En el siguiente ejemplo, se muestra cómo generar un pin aleatorio y, a continuación, generar el bloque de pines cifrado y el valor de compensación del IBM3624 mediante Ibm3624RandomPin

```
$ 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"
             }
         }
```

Comprobación de datos PIN

Las funciones de comprobación de datos PIN se utilizan para comprobar si un PIN es correcto. Por lo general, esto implica comparar el valor del PIN previamente almacenado con el que ingresó el titular de la tarjeta en un POI. Estas funciones comparan dos valores sin exponer el valor subyacente de ninguna de las fuentes.

Valide el PIN cifrado mediante el método PVV

Example

En este ejemplo, validaremos un PIN para un PAN determinado. Por lo general, el titular de la tarjeta o el usuario proporciona el PIN durante el momento de la transacción para su validación y se compara con el valor registrado (la entrada del titular de la tarjeta se proporciona como un valor cifrado del terminal u otro proveedor principal). Para validar esta entrada, también se proporcionarán los siguientes valores durante el tiempo de ejecución: la clave utilizada para cifrar el PIN de entrada (suele ser un`IWK`) PAN y el valor con el que se debe verificar (o un o). `PVV` `PIN offset`

Si la criptografía de AWS pagos puede validar el PIN, http/200 se devuelve un PIN. Si el PIN no está validado, devolverá un 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",

        }
```

Valide el PIN cifrado mediante el método PVV: error: PIN incorrecto

Example

En este ejemplo, intentaremos validar un PIN para un PAN determinado, pero fallará porque el PIN es incorrecto.

Al usar los SDK, aparece como {"Mensaje» :"Falló la verificación del bloqueo del PIN». , "Motivo» :"INVALID_PIN "}

```
$ 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=9999}" --
encrypted-pin-block AC17DC148BDA645E
```

```
       An error occurred (VerificationFailedException) when calling the VerifyPinData
 operation: Pin block verification failed.
```

Valide el PIN cifrado mediante el método PVV: error al introducir entradas incorrectas

Example

En este ejemplo, intentaremos validar un PIN para un PAN determinado, pero fallará debido a que las entradas son incorrectas y los datos entrantes no eran un PIN válido. Las causas más comunes son: la 1/wrong clave utilizada, 2/input parámetros como el formato de encuadre o bloque de pines son incorrectos, el 3/pin bloque está dañado.

```
$ aws payment-cryptography-data verify-pin-data --verification-key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/37y2tsl45p5zjbh2
 --encryption-key-identifier  --primary-account-number 171234567890123
 --pin-block-format ISO_FORMAT_0  --verification-attributes
 VisaPin="{PinVerificationKeyIndex=1,VerificationValue=9999}" --encrypted-pin-block
 AC17DC148BDA645E
```

```
       An error occurred (ValidationException) when calling the VerifyPinData
operation: Pin block provided is invalid. Please check your input to ensure all field
values are correct.
```

Valide un PIN comparándolo con el desfase de pines del IBM3624 previamente almacenado

En este ejemplo, valida un PIN proporcionado por el titular de la tarjeta comparándolo con el desfase de PIN almacenado en el archivo del emisor o procesador de la tarjeta. Las entradas incluyen la clave de verificación (PVK), la clave de cifrado (PEK), el número de cuenta principal, el bloqueo del PIN cifrado del terminal de pago (u otro proveedor principal, como una red de tarjetas) y el desfase de PIN almacenado anteriormente. Si el PIN coincide, la API devuelve una respuesta. HTTP/200

Este comando requiere que la clave de verificación del PIN sea de tipo `TR31_V1_IBM3624_PIN_VERIFICATION_KEY` y que la clave de cifrado sea de tipo`TR31_P0_PIN_ENCRYPTION_KEY`.

> ⚠ Validación de la longitud del PIN
>
> El `--pin-data-length` parámetro opcional especifica la longitud del PIN que se está verificando (de 4 a 12 dígitos). Si no especifica este parámetro, el servicio valida toda la

longitud del PIN que protege el valor de compensación. Si desea validar solo una parte del PIN, utilice el `--pin-data-length` parámetro. Por ejemplo, si `--pin-data-length` se establece en 4 pero el PIN real tiene 6 dígitos, el servicio valida solo los 4 dígitos situados más a la derecha. Los 2 dígitos restantes no se comprueban. Puedes hacerlo intencionadamente si el PIN del titular de la tarjeta tiene 6 dígitos, pero el PIN entrante solo tiene 4 dígitos, por ejemplo, cuando la red o el comercio entrantes solo admiten un PIN de 4 dígitos.

Example

```
$ 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 \
    --encrypted-pin-block AC17DC148BDA645E \
    --pin-data-length 4 \
    --verification-attributes
 Ibm3624PinOffset="{DecimalizationTable=9876543210654321,PinValidationDataPadCharacter=D,PinVal
```

```
{
    "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"
}
```

Generar y verificar el ARQC

Un criptograma de solicitud de autorización (ARQC) es un criptograma generado por una tarjeta con chip EMV (o su equivalente digital, como una cartera móvil) durante una transacción de pago. El ARQC valida criptográficamente la tarjeta y garantiza la integridad de los datos de la transacción. Incorpora los datos de la tarjeta, los datos del terminal y los datos específicos de la transacción para generar un valor único para cada transacción.

AWS La criptografía de pagos admite dos operaciones de ARQC. Puede generar un ARQC con fines de desarrollo y pruebas, y puede verificar un ARQC para los flujos de autorización de transacciones de producción.

Temas

- Generar criptograma de solicitud de autenticación (ARQC)
- Verificar el criptograma de solicitud de autenticación (ARQC)

Generar criptograma de solicitud de autenticación (ARQC)

La API de criptograma de generación de solicitud de autenticación se utiliza para generar el ARQC. Esta API le permite generar el ARQC mediante criptografía de AWS pagos con fines de desarrollo y pruebas.

> **Important**
>
> Esta operación está destinada únicamente a escenarios de desarrollo y prueba. No se recomienda utilizar esta operación como sustituto de la generación de criptogramas basada en tarjetas en los flujos de pago de producción.

Para ver todas las opciones disponibles, consulte la referencia de GenerateAuthRequestCryptogramla API.

Los criptogramas ARQC suelen requerir las siguientes entradas (aunque esto puede variar según la implementación):

- PAN: se especifica en el campo PrimaryAccountNumber
- Número de secuencia PAN (PSN): se especifica en el campo PanSequenceNumber
- Método de derivación de claves, como la clave de sesión común (CSK), especificado en el SessionKeyDerivationAttributes
- Modo de derivación de clave maestra (como la opción A de EMV): especificado en el MajorKeyDerivationMode
- Datos de la transacción: cadena de datos de la transacción, el terminal y la tarjeta, como el importe y la fecha. Se especifica en el TransactionData campo.
- Clave maestra del emisor: la clave maestra utilizada para derivar la clave del criptograma (AC). Esta clave protege las transacciones individuales y se especifica en el KeyIdentifier campo.

Para obtener más información sobre la creación y el relleno de los datos de las transacciones, consulte Verificar el criptograma de la solicitud de autenticación (ARQC).

Temas

- Ejemplos

Ejemplos

Visa CVN10

Example

El siguiente ejemplo genera un ARQC con Visa CVN10.

Si AWS Payment Cryptography genera el ARQC correctamente, se devuelve un http/200 ARN con el criptograma generado.

```
$ aws payment-cryptography-data generate-auth-request-cryptogram \
--key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6nl62t5ushfk \
--major-key-derivation-mode EMV_OPTION_A \
--transaction-data
 00000000170000000000000008400080008000084016051700000000093800000B03011203000000 \
--session-key-derivation-attributes='{"Visa":{"PanSequenceNumber":"01", \
"PrimaryAccountNumber":"9137631040001422"}}'
```

```
{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/pw3s6nl62t5ushfk",
  "KeyCheckValue": "08D7B4",
  "AuthRequestCryptogram": "D791093C8A921769"
}
```

Visa CVN18 y Visa CVN22

Example

El siguiente ejemplo genera un ARQC con Visa CVN18 o CVN22. Las operaciones criptográficas son las mismas entre el CVN18 y el CVN22, pero los datos contenidos en los datos de las transacciones varían. En comparación con el CVN10, se genera un criptograma completamente diferente incluso con las mismas entradas.

Si AWS Payment Cryptography genera el ARQC correctamente, se devuelve un ARN con el criptograma generado. http/200

```
$ aws payment-cryptography-data generate-auth-request-cryptogram \
--key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6nl62t5ushfk \
--major-key-derivation-mode EMV_OPTION_A \
--transaction-data
 00000000170000000000000084000800080000840160517000000000093800000B1F22010300000000000
 \
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",
  "AuthRequestCryptogram": "61EDCC708B4C97B4"
}
```

Verificar el criptograma de solicitud de autenticación (ARQC)

La API de verificación del criptograma de solicitud de autenticación se utiliza para verificar el ARQC. La generación del ARQC se realiza normalmente en una tarjeta con chip EMV (o un equivalente digital, como una cartera móvil) durante el tiempo de autorización de la transacción, aunque el servicio se puede generar con fines de desarrollo. Un ARQC es único para cada transacción y su objetivo es mostrar criptográficamente la validez de la tarjeta y garantizar que los datos de la transacción coincidan exactamente con los de la transacción actual (esperada).

AWS La criptografía de pagos ofrece una variedad de opciones para validar el ARQC y generar valores ARPC opcionales, incluidos los definidos en la EMV 4.4, libro 2, y otros esquemas utilizados

por Visa y Mastercard. Para obtener una lista completa de todas las opciones disponibles, consulte la sección de la VerifyCardValidationData Guía de API.

Los criptogramas ARQC suelen requerir las siguientes entradas (aunque esto puede variar según la implementación):

- PAN: se especifica en el campo PrimaryAccountNumber
- Número de secuencia PAN (PSN): especificado en el campo PanSequenceNumber
- Método de derivación de claves, como la clave de sesión común (CSK), especificado en el SessionKeyDerivationAttributes
- Modo de derivación de clave maestra (como la opción A de EMV): especificado en el MajorKeyDerivationMode
- Datos de la transacción: una cadena de varios datos de transacciones, terminales y tarjetas, como el importe y la fecha, especificados en el campo TransactionData
- Clave maestra del emisor: la clave maestra utilizada para derivar la clave de criptograma (AC) utilizada para proteger las transacciones individuales y especificada en el campo KeyIdentifier

Temas

- Creación de datos de transacciones
- Relleno de datos de transacciones
- Ejemplos

Creación de datos de transacciones

El contenido (y el orden) exactos del campo de datos de la transacción varían según la implementación y el esquema de red, pero los campos mínimos recomendados (y la secuencia de concatenación) se definen en la sección 8.1.1 del libro 2 de EMV 4.4: Selección de datos. Si los tres primeros campos son importe (17.00), otro importe (0,00) y país de compra, los datos de la transacción comenzarán de la siguiente manera:

- 000000001700 - cantidad: 12 posiciones implican un decimal de dos dígitos
- 000000000000 - otra cantidad: 12 posiciones implican un decimal de dos dígitos
- 0124: código de país de cuatro dígitos
- Datos de transacción de salida (parciales): 00000000170000000000000000124

Relleno de datos de transacciones

Los datos de las transacciones deben rellenarse antes de enviarlos al servicio. La mayoría de los esquemas utilizan el relleno ISO 9797 Método 2, en el que se añade una cadena hexadecimal 80 seguida de 00 hasta que el campo es un múltiplo del tamaño del bloque de cifrado; 8 bytes o 16 caracteres para TDES y 16 bytes o 32 caracteres para AES. La alternativa (método 1) no es tan común pero utiliza sólo 00 como caracteres de relleno.

Relleno ISO 9797 Método 1

Sin relleno:
0000000017000000000000000840008000800008401605170000000093800000B03011203
(74 caracteres o 37 bytes)

Con relleno:
0000000017000000000000000840008000800008401605170000000093800000B03011203000000
(80 caracteres o 40 bytes)

Relleno ISO 9797 Método 2

Sin relleno:
0000000017000000000000000840008000800008401605170000000093800000B1F220103000000
(80 caracteres o 40 bytes)

Sin relleno:
0000000017000000000000000840008000800008401605170000000093800000B1F220103000000800000
(88 caracteres o 44 bytes)

Ejemplos

Visa CVN10

Example

En este ejemplo, validaremos un ARQC generado con Visa CVN10.

Si la criptografía AWS de pagos es capaz de validar el ARQC, se devuelve un. http/200 Si, entonces, el ARCQ (criptograma de solicitud de autorización) no está validado, devolverá una respuesta. 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
 00000000170000000000000008400080008000084016051700000000093800000B03011203000000 \
--session-key-derivation-attributes='{"Visa":{"PanSequenceNumber":"01" \
,"PrimaryAccountNumber":"9137631040001422"}}'
```

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

Visa CVN18 y Visa CVN22

Example

En este ejemplo, validaremos un ARQC generado con Visa CVN18 o CVN22. Las operaciones criptográficas son las mismas entre el CVN18 y el CVN22, pero los datos contenidos en los datos de las transacciones varían. En comparación con el CVN10, se genera un criptograma completamente diferente incluso con las mismas entradas.

Si la criptografía de AWS pago puede validar el ARQC, se devuelve un. http/200 Si el ARCQ no está validado, devolverá un. 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
 00000000170000000000000084000800080000840160517000000000093800000B1F22010300000000000
 \
00000000000000000000000000000000000000000800000000000000
--session-key-derivation-attributes='{"EmvCommon":
{"ApplicationTransactionCounter":"000B", \
"PanSequenceNumber":"01","PrimaryAccountNumber":"9137631040001422"}}'
```

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

Generar y verificar MAC

Los códigos de autenticación de mensajes (MAC) se utilizan normalmente para autentificar la integridad de un mensaje (independientemente de si se ha modificado). Los hashes criptográficos, como el HMAC (código de autenticación de Hash-Based mensajes) CBC-MAC y el CMAC (código de autenticación de Cipher-based mensajes), proporcionan una seguridad adicional para el remitente del MAC mediante el uso de la criptografía. El HMAC se basa en funciones de hash, mientras que el CMAC se basa en cifrados por bloques. El servicio también es compatible con los algoritmos ISO9797 1 y 3, que son tipos de. CBC-MACs

Todos los algoritmos MAC de este servicio combinan una función hash criptográfica y una clave secreta compartida. Reciben un mensaje y una clave secreta, como el material clave de una llave, y devuelven una etiqueta o mac únicos. Si cambia incluso un carácter del mensaje, o si la clave secreta cambia, la etiqueta resultante es totalmente diferente. Al requerir una clave secreta, los MAC criptográficos también proporcionan autenticación; es imposible generar una mac idéntica sin la clave secreta. Los MAC criptográficos a veces se llaman firmas simétricas, porque funcionan como firmas digitales, pero utilizan una única clave para la firma y la verificación.

AWS La criptografía de pagos admite varios tipos de MAC:

ALGORITMO 1 ISO9797

Denotado por ISO9797_ALGORITHM1`KeyUsage`. Si el campo no es un múltiplo del tamaño de un bloque (8 caracteres bytes/16 hexadecimales para el TDES y 16 bytes/32 caracteres para el AES), Payment Cryptography aplicará automáticamente el método de relleno 1 según la norma ISO9797. AWS Si necesita otros métodos de relleno, puede aplicarlos antes de llamar al servicio.

ALGORITMO 3 ISO9797 (MAC minorista)

Denotado por `KeyUsage` ISO9797_ALGORITHM3. Se aplican las mismas reglas de relleno que en el Algoritmo 1

ALGORITMO 5 ISO9797 (CMAC)

Denotado por `KeyUsage` del TR31_M6_ISO_9797_5_CMAC_KEY

HMAC

Denotado por `KeyUsage` del TR31_M7_HMAC_KEY incluyendo HMAC_SHA224, HMAC_SHA256, HMAC_SHA384 y HMAC_SHA512

AS2805.4.1 MAC

Denotado por `KeyUsage` TR31_M0_ISO_16609_MAC_KEY. Para obtener más información sobre el AS2805, consulte ???

DUKPT MAC

El DUKPT MAC se utiliza normalmente para confirmar el origen y la carga útil de los mensajes en los terminales de pago. to/from Obtiene una clave mediante técnicas de derivación DUKPT y, a continuación, realiza el MAC. Las claves utilizadas con esta opción se indican con el símbolo TR31_B0_BASE_DERIVATION_KEY. `KeyUsage`

EMV MAC

En la documentación de EMV, la MAC suele denominarse clave de integridad. Deriva una clave mediante técnicas de derivación EMV y, a continuación, utiliza la norma ISO9797_ALGORITHM3 internamente. Por lo general, se utiliza para enviar los scripts del emisor a una tarjeta con chip para su reprogramación. Las claves utilizadas con esta opción se indican con el símbolo TR31_E2_EMV_MKEY_INTEGRITY`KeyUsage`. Si está enviando un script y actualizando un pin fuera de línea, asegúrese de que realice estas dos operaciones. GenerateMacEmvPinChange

Temas

- Generar MAC
- Verificar MAC

Generar MAC

La API Generate MAC se utiliza para autenticar los datos relacionados con las tarjetas, como rastrear los datos de una banda magnética de tarjetas, mediante el uso de claves criptográficas conocidas para generar un MAC (código de autenticación de mensajes) para la validación de los datos entre las partes remitentes y receptoras. Los datos utilizados para generar el MAC incluyen los datos de los mensajes, la clave de cifrado MAC secreta y el algoritmo MAC para generar un valor MAC único para la transmisión. La parte receptora del MAC utiliza los mismos datos del mensaje MAC, la misma clave de cifrado MAC y el mismo algoritmo para reproducir otro valor MAC con fines de comparación y autenticación de datos. Aunque cambie un carácter del mensaje o la clave MAC utilizada para la verificación no sea idéntica, el valor MAC resultante será diferente. La API admite las claves de cifrado MAC HMAC y EMV MAC (que utilizan una clave MAC estática y una clave DUKPT derivada) del algoritmo 1 de la ISO 9797-1 y el algoritmo 3 de la ISO 9797-1.

El valor de entrada para `message-data` debe ser un dato hexBinary.

Para obtener más información sobre todas las opciones de esta API, consulte y. GenerateMacVerifyMac

El parámetro opcional mac-length te permite truncar el valor de salida (aunque esto también se puede hacer dentro del código). Una longitud de 8 se refiere a 8 bytes o 16 caracteres hexadecimales.

Las claves MAC se pueden crear con criptografía AWS de pago mediante una llamada CreateKeyo importarse mediante una llamada ImportKey.

> **Note**
>
> Los algoritmos CMAC y HMAC no requieren relleno. Todos los demás requieren que los datos se rellenen hasta el tamaño de bloque del algoritmo, que es múltiplos de 8 bytes (16 caracteres hexadecimales) para el TDES y 16 bytes (32 caracteres hexadecimales) para el AES.

Ejemplos

- Genera HMAC
- Genere el MAC mediante el algoritmo ISO 9797-1 3
- Genere MAC mediante CMAC
- Genere MAC mediante DUKPT CMAC

Genera HMAC

En este ejemplo, generaremos un HMAC (código de autenticación de Hash-Based mensajes) para la autenticación de los datos de la tarjeta mediante el algoritmo HMAC `HMAC_SHA256` y la clave de cifrado HMAC. La clave debe estar KeyUsage configurada en y en. `TR31_M7_HMAC_KEY` KeyModesOfUse `Generate` La longitud del hash (por ejemplo, 256) se define cuando se crea la clave y no se puede modificar.

El parámetro opcional mac-length recortará el MAC de salida, aunque también se puede realizar fuera del servicio. Este valor está en bytes, por lo que un valor de 16 esperará una cadena hexadecimal de 32 de longitud.

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
```

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

Genere el MAC mediante el algoritmo ISO 9797-1 3

En este ejemplo, generaremos un MAC utilizando el algoritmo 3 de la norma ISO 9797-1 (MAC minorista) para la autenticación de los datos de las tarjetas. La clave debe estar KeyUsage configurada en `TR31_M3_ISO_9797_3_MAC_KEY` y KeyModesOfUse en. `Generate`

Example

```
$ aws payment-cryptography-data generate-mac \
    --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h \
    --message-data
 "3b3130383834393030313037333934313533d32343038323236303030373030303f33" \
    --generation-attributes="Algorithm=ISO9797_ALGORITHM3"
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h",
    "KeyCheckValue": "2976EA",
    "Mac": "A8F7A73DAF87B6D0"
}
```

Genere MAC mediante CMAC

El CMAC se usa más comúnmente cuando las claves son AES, pero también es compatible con el TDES. En este ejemplo, generaremos un MAC mediante el CMAC (algoritmo 5 de la norma ISO 9797-1) para la autenticación de los datos de la tarjeta con una clave AES. La clave debe estar KeyUsage configurada en y en. `TR31_M6_ISO_9797_5_CMAC_KEY` KeyModesOfUse `Generate`

Example

```
$ aws payment-cryptography-data generate-mac \
    --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi \
    --message-data
 "3b3130383834393030313037333934313533d32343038323236303030373030303f33" \
    --generation-attributes Algorithm="CMAC"
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
    "KeyCheckValue": "C1EB8F",
    "Mac": "1F8C36E63F91E4E93DF7842BF5E2E5F7"
}
```

Genere MAC mediante DUKPT CMAC

En este ejemplo, se genera una MAC mediante la DUKPT (clave única derivada por transacción) con la CMAC para la autenticación de los datos de la tarjeta. La clave debe estar KeyUsage establecida `TR31_B0_BASE_DERIVATION_KEY` y establecida en KeyModesOfUse `DeriveKey` true.

Las claves DUKPT obtienen una clave única para cada transacción mediante una clave de derivación básica (BDK) y un número de serie de la clave (KSN).

Example

```
$ aws payment-cryptography-data generate-mac --key-identifier arn:aws:payment-
cryptography:us-east-2:111122223333:key/qnobl5lghrzunce6 --message-data
 "3b3130383834393030313037333934313533d32343038323236303030373030303f33" --generation-
attributes="DukptCmac={KeySerialNumber="932A6E954ABB32DD00000001",DukptKeyVariant=BIDIRECTIONAL
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
qnobl5lghrzunce6",
    "KeyCheckValue": "C1EB8F",
    "Mac": "1F8C36E63F91E4E93DF7842BF5E2E5F7"
}
```

Verificar MAC

Verifique que la MAC API se utiliza para verificar MAC (código de autenticación de mensajes) para la autenticación de datos relacionados con tarjetas. Debe utilizar la misma clave de cifrado utilizada durante la generación del MAC para reproducir el valor MAC para la autenticación. La clave de cifrado MAC se puede crear con criptografía de AWS pago mediante una llamada CreateKeyo importarse mediante una llamada. ImportKey La API admite las claves de cifrado DUKPT MAC, HMAC y EMV MAC para esta operación.

Si el valor se verifica, el parámetro de respuesta `MacDataVerificationSuccessful` devolverá `Http/200`, en caso contrario `Http/400` con un mensaje indicando que `Mac verification failed`.

Ejemplos

- Verifica HMAC
- Verifique el MAC mediante DUKPT CMAC

Verifica HMAC

En este ejemplo, verificaremos un HMAC (código de autenticación de Hash-Based mensajes) para la autenticación de los datos de la tarjeta mediante el algoritmo HMAC `HMAC_SHA256` y la clave de cifrado HMAC. La clave debe estar KeyUsage establecida `TR31_M7_HMAC_KEY` y establecida en KeyModesOfUse `Verify` true.

Example

```
$ aws payment-cryptography-data verify-mac \
     --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
qnobl5lghrzunce6 \
     --message-data
 "3b343038383439303031303733393431353d323430383232363030303730303033f33" \
     --mac ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C \
     --verification-attributes Algorithm=HMAC_SHA256
```

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

Verifique el MAC mediante DUKPT CMAC

En este ejemplo, verificaremos un MAC mediante la DUKPT (clave única derivada por transacción) con la CMAC para la autenticación de los datos de la tarjeta. La clave debe estar KeyUsage establecida `TR31_B0_BASE_DERIVATION_KEY` y establecida en KeyModesOfUse `DeriveKey` true. Las claves DUKPT obtienen una clave única para cada transacción mediante una clave de derivación básica (BDK) y un número de serie de la clave (KSN). El valor de DukptKeyVariant debe coincidir entre el remitente y el destinatario. Por lo general, REQUEST se usa del terminal al servidor, VERIFY del servidor al terminal y BIDIRECTIONAL cuando se usa una sola clave en ambas direcciones.

Example

```
$ aws payment-cryptography-data verify-mac \
     --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi \
     --message-data
 "3b343038383439303031303733393431353d323430383232363030303730303033f33" \
     --mac D8E804EE74BF1D909A2C01C0BDE8EF34 \
     --verification-attributes
 DukptCmac='{"KeySerialNumber":"932A6E954ABB32DD00000001","DukptKeyVariant":"BIDIRECTIONAL"}'
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
    "KeyCheckValue": "C1EB8F"
}
```

Claves válidas para las operaciones criptográficas

Algunas claves solo se pueden utilizar para determinadas operaciones. Además, algunas operaciones pueden limitar los modos de uso de las claves. Consulte la tabla siguiente para las combinaciones permitidas.

> **Note**
>
> Ciertas combinaciones, si bien están permitidas, pueden crear situaciones inutilizables, como generar códigos CVV (`generate`) pero luego no pueden verificarlos (`verify`).

Temas

- GenerateCardData
- VerifyCardData
- GeneratePinData (para VISA/ABA esquemas)
- GeneratePinData (para IBM3624)
- VerifyPinData (para VISA/ABA esquemas)
- VerifyPinData (para IBM3624)

- Descifrado de datos
- Cifrado de datos
- Traducir datos PIN
- Generate/Verify MAC
- GenerateMacEmvPinChange
- VerifyAuthRequestCryptogram
- Import/Export Clave
- Tipos de claves sin utilizar

GenerateCardData

Punto de conexión de la API	Operación criptográfica o algoritmo	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
GenerateCardData	• AMEX_CARD_SECURITY_CODE_VERSION_1 • AMEX_CARD_SECURITY_CODE_VERSION_2	TR31_C0_CARD_VERIFICATION_KEY	• TDES_2KEY • TDES_3KEY	{ Generate = true },{ Generate = true, Verify = true }
GenerateCardData	• CARD_VERIFICATION_VALUE_1 • CARD_VERIFICATION_VALUE_2	TR31_C0_CARD_VERIFICATION_KEY	• TDES_2KEY	{ Generate = true },{ Generate = true, Verify = true }
GenerateCardData	• CARDHOLDER_AUTHENTICATION_V	TR31_E6_EMV_MKEY_OTHER	• TDES_2KEY	{ DeriveKey = verdadero}

Punto de conexión de la API	Operación criptográfica o algoritmo	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
	ERIFICATION_VALUE			
GenerateCardData	• DYNAMIC_CARD_VERIFICATION_CODE	TR31_E4_EMV_MKEY_DYNAMIC_NUMBERS	• TDES_2KEY	{ DeriveKey = verdadero}
GenerateCardData	• DYNAMIC_CARD_VERIFICATION_VALUE	TR31_E6_EMV_MKEY_OTHER	• TDES_2KEY	{ DeriveKey = verdadero}

VerifyCardData

Operación criptográfica o algoritmo	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
• AMEX_CARD_SECURITY_CODE_VERSION_1 • AMEX_CARD_SECURITY_CODE_VERSION_2	TR31_C0_CARD_VERIFICATION_KEY	• TDES_2KEY • TDES_3KEY	{ Generate = true }, { Generate = true, Verify = true }
• CARD_VERIFICATION_VALUE_1	TR31_C0_CARD_VERIFICATION_KEY	• TDES_2KEY	{ Generate = true }, { Generate = true, Verify = true }

Operación criptográfica o algoritmo	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
• CARD_VERIFICATION_VALUE_2			
• CARDHOLDER_AUTHENTICATION_VERIFICATION_VALUE	TR31_E6_EMV_MKEY_OTHER	• TDES_2KEY	{ DeriveKey = verdadero}
• DYNAMIC_CARD_VERIFICATION_CODE	TR31_E4_EMV_MKEY_DYNAMIC_NUMBERS	• TDES_2KEY	{ DeriveKey = verdadero}
• DYNAMIC_CARD_VERIFICATION_VALUE	TR31_E6_EMV_MKEY_OTHER	• TDES_2KEY	{ DeriveKey = verdadero}

GeneratePinData (para VISA/ABA esquemas)

`VISA_PIN or VISA_PIN_VERIFICATION_VALUE`

Tipo de clave	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
Clave de cifrado de PIN	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY	• { Encrypt = true, Wrap = true } • { Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true }

Tipo de clave	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
			• { NoRestrictions = verdadero}
Clave de generación de PIN	TR31_V2_VISA_PIN_VERIFICATION_KEY	• TDES_2KEY • TDES_3KEY	• { Generate = true } • { Generate = true, Verify = true }

GeneratePinData (para **IBM3624**)

`IBM3624_PIN_OFFSET,IBM3624_NATURAL_PIN,IBM3624_RANDOM_PIN, IBM3624_PIN_FROM_OFFSET)`

Tipo de clave	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
Clave de cifrado de PIN	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY	Para IBM3624_NATURAL_PIN, IBM3624_RANDOM_PIN, IBM3624_PIN_FROM_OFFSET • { Encrypt = true, Wrap = true } • { Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true } • {= verdadero} NoRestrictions

Tipo de clave	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
			Para IBM3624_PIN_OFFSET • { Encrypt = true, Unwrap = true } • { Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true } • { NoRestrictions = verdadero}
Clave de generación de PIN	TR31_V1_IBM3624_PIN_VERIFICATION_KEY	• TDES_2KEY • TDES_3KEY	• { Generate = true } • { Generate = true, Verify = true }

VerifyPinData (para VISA/ABA esquemas)

`VISA_PIN`

Tipo de clave	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
Clave de cifrado de PIN	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY	• { Decrypt = true, Unwrap = true } • { Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true }

Tipo de clave	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
			• { NoRestrictions = verdadero}
Clave de generación de PIN	TR31_V2_VISA_PIN_VERIFICATION_KEY	• TDES_2KEY • TDES_3KEY	• { Verify = true } • { Generate = true, Verify = true }

VerifyPinData (para **IBM3624**)

`IBM3624_PIN_OFFSET,IBM3624_NATURAL_PIN,IBM3624_RANDOM_PIN, IBM3624_PIN_FROM_OFFSET)`

Tipo de clave	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
Clave de cifrado de PIN	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY	Para IBM3624_NATURAL_PIN, IBM3624_RANDOM_PIN, IBM3624_PIN_FROM_OFFSET • { Decrypt = true, Unwrap = true } • { Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true } • {= verdadero} NoRestrictions

Tipo de clave	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
Clave de verificación de PIN	TR31_V1_IBM3624_PIN_VERIFICATION_KEY	• TDES_2KEY • TDES_3KEY	• { Verify = true } • { Generate = true, Verify = true }

Descifrado de datos

Tipo de clave	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
DUKPT	TR31_B0_BASE_DERIVATION_KEY	• TDES_2KEY • AES_128 • AES_192 • AES_256	• { DeriveKey = verdadero} • { NoRestrictions = verdadero}
EMV	TR31_E1_EMV_MKEY_CONFIDENTIALITY TR31_E6_EMV_MKEY_OTHER	• TDES_2KEY	• { DeriveKey = verdadero}
RSA	TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION	• RSA_2048 • RSA_3072 • RSA_4096	• { Decrypt = true, Unwrap=true} • {Encrypt=true, Wrap=true,Decrypt = true, Unwrap=true}
Claves simétricas	TR31_D0_SYMMETRIC_	• TDES_2KEY • TDES_3KEY	• {Decrypt = true, Unwrap=true}

Tipo de clave	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
	DATA_ENCRYPTION_KEY	• AES_128 • AES_192 • AES_256	• {Encrypt=true, Wrap=true,Decrypt = true, Unwrap=true} • { NoRestrictions = verdadero}

Cifrado de datos

Tipo de clave	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
DUKPT	TR31_B0_BASE_DERIVATION_KEY	• TDES_2KEY • AES_128 • AES_192 • AES_256	• { DeriveKey = verdadero} • { NoRestrictions = verdadero}
EMV	TR31_E1_EMV_MKEY_CONFIDENTIALITY TR31_E6_EMV_MKEY_OTHER	• TDES_2KEY	• { DeriveKey = verdadero}
RSA	TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION	• RSA_2048 • RSA_3072 • RSA_4096	• { Encrypt = true, Wrap=true} • {Encrypt=true, Wrap=true,Decrypt = true, Unwrap=true}

Tipo de clave	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
Claves simétricas	TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY • AES_128 • AES_192 • AES_256	• {Encrypt = true, Wrap=true} • {Encrypt=true, Wrap=true,Decrypt = true, Unwrap=true} • { NoRestrictions = verdadero}

Traducir datos PIN

Dirección	Tipo de clave	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
Origen de datos de entrada	DUKPT	TR31_B0_BASE_DERIVATION_KEY	• TDES_2KEY • AES_128 • AES_192 • AES_256	• { DeriveKey = verdadero} • { NoRestrictions = verdadero}
Origen de datos de entrada	no DUKPT (PEK, AWK, IWK, etc.)	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY • AES_128 • AES_192 • AES_256	• { Decrypt = true, Unwrap = true } • { Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true }

Dirección	Tipo de clave	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
				• { NoRestrictions = verdadero}
Objetivo de datos de salida	DUKPT	TR31_B0_BASE_DERIVATION_KEY	• TDES_2KEY • AES_128 • AES_192 • AES_256	• { DeriveKey = verdadero} • { NoRestrictions = verdadero}
Objetivo de datos de salida	no DUKPT (PEK, IWK, AWK, etc)	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY • AES_128 • AES_192 • AES_256	• { Encrypt = true, Wrap = true } • { Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true } • { NoRestrictions = verdadero}

Generate/Verify MAC

Las claves MAC se utilizan para crear hashes criptográficos message/body de un dato. No se recomienda crear una clave con modos de uso limitados, ya que no podrá realizar la operación de comparación. Sin embargo, puede import/export usar una clave con una sola operación si el otro sistema está diseñado para realizar la otra mitad del par de operaciones.

Uso permitido de claves	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
Clave MAC	TR31_M1_ISO_9797_1_MAC_KEY	• TDES_2KEY • TDES_3KEY	• { Generate = true } • { Generate = true, Verify = true } • { Verify = true } • { Generate = true }
MAC Key (MAC minorista)	TR31_M1_ISO_9797_3_MAC_KEY	• TDES_2KEY • TDES_3KEY	• { Generate = true } • { Generate = true, Verify = true } • { Verify = true } • { Generate = true }
Clave MAC (CMAC)	TR31_M6_ISO_9797_5_CMAC_KEY	• TDES_2KEY • TDES_3KEY • AES_128 • AES_192 • AES_256	• { Generate = true } • { Generate = true, Verify = true } • { Verify = true } • { Generate = true }
Clave MAC (HMAC)	TR31_M7_HMAC_KEY	• TDES_2KEY • TDES_3KEY • AES_128 • AES_192 • AES_256	• { Generate = true } • { Generate = true, Verify = true } • { Verify = true }
Clave MAC (AS2805)	TR31_M0_ISO_16609_MAC_KEY	• TDES_2KEY • TDES_3KEY	• { Generate = true } • { Generate = true, Verify = true } • { Verify = true }

GenerateMacEmvPinChange

GenerateMacEmvPinChange combina la generación de MAC y el cifrado de PIN para las operaciones de cambio de PIN fuera de línea de EMV. Esta operación requiere dos tipos de claves diferentes: una clave de integridad para la generación de MAC y una clave de confidencialidad para el cifrado del PIN.

Tipo de clave	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
Clave de integridad de mensajería segura	TR31_E2_EMV_MKEY_INTEGRITY	• TDES_2KEY	• { NoRestrictions = verdadero}
Clave de confidencialidad de mensajería segura	TR31_E1_EMV_MKEY_CONFIDENTIALITY	• TDES_2KEY	• { DeriveKey = verdadero}
PIN PEK (clave de cifrado PIN) actual	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY • AES_128 • AES_192 • AES_256	• { Decrypt = true, Unwrap = true } • { Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true } • { NoRestrictions = verdadero}
Nuevo PIN PEK (clave de cifrado PIN)	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY • AES_128 • AES_192 • AES_256	• { Decrypt = true, Unwrap = true } • { Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true } • { NoRestrictions = verdadero}

Tipo de clave	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
Clave ARQC **Note** Solo se aplica a los esquemas de derivación de Visa y Amex.	TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS	• TDES_2KEY	• { DeriveKey = verdadero}

VerifyAuthRequestCryptogram

Uso permitido de claves	Opción EMV	Algoritmo clave permitido	Combinación permitida de modos de uso clave
• OPCIÓN A • OOCIÓN B	TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS	• TDES_2KEY	• { DeriveKey = verdadero}

Import/Export Clave

Operation Type (Tipo de operación)	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
TR-31 Clave de embalaje	TR31_K1_KEY_BLOCK_PROTECTION_KEY	• TDES_2KEY • TDES_3KEY • AES_128 • AES_192	• {Encrypt = true, Wrap = true} (solo exportación)

Operation Type (Tipo de operación)	Uso permitido de claves	Algoritmo clave permitido	Combinación permitida de modos de uso clave
	TR31_K0_KEY_ENCRYPTION_KEY	• AES_256	• {Decrypt = true, Unwrap = true} (solo importación) • { Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true }
Importación de una CA de confianza	TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE	• RSA_2048 • RSA_3072 • RSA_4096	• { Verify = true }
Importación de un certificado de clave pública para un cifrado asimétrico	TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION	• RSA_2048 • RSA_3072 • RSA_4096	• {encrypt=true, wrap=true}
Clave utilizada para codificar algoritmos de acuerdo como el ECDH	TR31_K3_ASYMMETRIC_KEY_FOR_KEY_AGREEMENT	• ECC_NIST_P256 • ECC_NIST_P384 • ECC_NIST_P521	• { DeriveKey = verdadero}

Tipos de claves sin utilizar

La criptografía de AWS pagos no utiliza actualmente los siguientes tipos de claves

- TR31_P1_PIN_GENERATION_KEY

Casos de uso comunes

AWS La criptografía de pagos admite muchas operaciones criptográficas de pago típicas. Los siguientes temas sirven de guía sobre cómo usar estas operaciones en casos de uso comunes típicos. Para obtener una lista de todos los comandos, consulta la API de criptografía de AWS pagos.

Temas

- Emisores y procesadores de emisores
- Facilitadores de adquisiciones y pagos

Emisores y procesadores de emisores

Los casos de uso de los emisores suelen constar de unas pocas partes. Esta sección está organizada por función (por ejemplo, trabajar con pines). En un sistema de producción, las claves suelen estar distribuidas en una papelera determinada y se crean durante la configuración de la papelera, en lugar de colocarlas en línea, como se muestra aquí.

Temas

- Funciones generales
- Funciones específicas de la red

Funciones generales

Temas

- Genere un pin aleatorio y el PVV asociado y, a continuación, verifique el valor
- Genera o verifica un CVV para una tarjeta determinada
- Genere o verifique un CVV2 para una tarjeta específica
- Genere o verifique un iCVV para una tarjeta específica
- Verifique un ARQC EMV y genere un ARPC
- Genere y verifique un MAC EMV
- Genere un MAC EMV para cambiar el PIN

Genere un pin aleatorio y el PVV asociado y, a continuación, verifique el valor

Temas

- Cree la (s) clave (s)
- Genera un PIN aleatorio, genera PVV y devuelve el PIN y el PVV cifrados
- Valide el PIN cifrado mediante el método PVV
- PIN gestionados por el usuario

Cree la (s) clave (s)

Para generar un PIN aleatorio y el PVV, necesitarás dos claves: una clave de verificación del PIN (PVK) para generar el PVV y una clave de cifrado del PIN para cifrar el PIN. El PIN en sí mismo se genera aleatoriamente de forma segura dentro del servicio y no está relacionado criptográficamente con ninguna de las claves.

La PGK debe ser una clave del algoritmo TDES_2KEY basada en el propio algoritmo PVV. Un PEK puede ser TDES_2KEY, TDES_3KEY o AES_128. En este caso, dado que el PEK está diseñado para uso interno en su sistema, AES_128 sería una buena elección. Si un PEK se usa para intercambiarlo con otros sistemas (por ejemplo, redes de tarjetas, entidades adquirentes, cajeros automáticos) o se traslada como parte de una migración, TDES_2KEY puede ser la opción más adecuada por motivos de compatibilidad.

Crea el PEK

```
$ aws payment-cryptography create-key \
               --exportable
               --key-attributes
 KeyAlgorithm=AES_128,KeyUsage=TR31_P0_PIN_ENCRYPTION_KEY,\
                   KeyClass=SYMMETRIC_KEY,\
                   KeyModesOfUse='{Encrypt=true,Decrypt=true,Wrap=true,Unwrap=true}' --
tags='[{"Key":"CARD_BIN","Value":"12345678"}]'
```

La respuesta refleja los parámetros de la solicitud, incluyendo un ARN para las llamadas posteriores y un valor de verificación clave (KCV).

```
{
               "Key": {
                   "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt",
```

```
                "KeyAttributes": {
                    "KeyUsage": "TR31_P0_PIN_ENCRYPTION_KEY",
                    "KeyClass": "SYMMETRIC_KEY",
                    "KeyAlgorithm": "AES_128",
                    "KeyModesOfUse": {
                        "Encrypt": false,
                        "Decrypt": false,
                        "Wrap": false,
                        "Unwrap": false,
                        "Generate": true,
                        "Sign": false,
                        "Verify": true,
                        "DeriveKey": false,
                        "NoRestrictions": false
                    }
                },
                "KeyCheckValue": "7CC9E2",
                "KeyCheckValueAlgorithm": "CMAC",
                "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"
            }
        }
```

Toma nota de lo `KeyArn` que representa la clave, por ejemplo arn:aws:payment-cryptography:us-east- 2:111122223333:. key/ivi5ksfsuplneuyt Lo necesitará en el siguiente paso.

Crea el PVK

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_V2_VISA_PIN_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY,KeyMo
  --tags='[{"Key":"CARD_BIN","Value":"12345678"}]'
```

La respuesta refleja los parámetros de la solicitud, incluyendo un ARN para las llamadas posteriores y un valor de verificación clave (KCV).

```
{
                "Key": {
                    "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/ov6icy4ryas4zcza",
```

```
                "KeyAttributes": {
                    "KeyUsage": "TR31_V2_VISA_PIN_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": "51A200",
                "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"
            }
        }
```

Tome nota de lo `KeyArn` que representa la clave, por ejemplo arn:aws:payment-cryptography:us-east- 2:111122223333:. key/ov6icy4ryas4zcza Lo necesitará en el siguiente paso.

Genera un PIN aleatorio, genera PVV y devuelve el PIN y el PVV cifrados

Example

En este ejemplo, generaremos un nuevo PIN (aleatorio) de 4 dígitos en el que los resultados serán un () cifrado `PIN block` (PinData.PinBlock) y un `PVV` (pinData.VerificationValue). Las entradas clave son `PAN` el formato PIN `Pin Verification Key` (también conocido como clave de generación de pines) `Pin Encryption Key` y el `formato` PIN Block.

Este comando requiere que la clave sea de tipo`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"
            }
        }
```

Valide el PIN cifrado mediante el método PVV

Example

En este ejemplo, validaremos un PIN para un PAN determinado. Por lo general, el titular de la tarjeta o el usuario proporciona el PIN durante el momento de la transacción para su validación y se compara con el valor registrado (la entrada del titular de la tarjeta se proporciona como un valor cifrado del terminal u otro proveedor principal). Para validar esta entrada, también se proporcionarán los siguientes valores durante el tiempo de ejecución: el pin cifrado, la clave utilizada para cifrar el pin de entrada (a menudo denominado IWK) PAN y el valor con el que se realizará la verificación (ya sea un PVV o`PIN offset`).

Si la criptografía de AWS pagos puede validar el PIN, se devuelve un http/200 PIN. Si el PIN no está validado, devolverá un 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",
}
```

PIN gestionados por el usuario

Temas

- Descripción general de
- Crea las claves
- Establezca un PIN que el usuario pueda seleccionar
- Restablecer un PIN (PIN verde)
- Revela un PIN

Descripción general de

AWS La criptografía de pagos admite los flujos de PIN gestionados por los usuarios (gestionados por los clientes), en los que un PIN se intercambia de forma segura entre el dispositivo del titular de la tarjeta y el servicio a través del backend del cliente. El dispositivo y el servicio utilizan un acuerdo de claves de curva elíptica Diffie-Hellman (ECDH) para obtener una clave simétrica compartida en cada lado, y esa clave protege un bloque de 4 PIN en formato ISO. ??? Como la clave se obtiene y no se intercambia, el PIN transparente y el bloque PIN transparente nunca se transmiten más allá del dispositivo.

Estos flujos involucran a tres partes. El dispositivo del titular de la tarjeta genera su par de claves EC, obtiene la AES-128 clave ECDH, cifra o descifra el bloque de 4 PIN en formato ISO y contiene el PIN transparente; no tiene credenciales de AWS. El servidor del cliente (su servidor emisor o procesador) conserva las credenciales de AWS, llama a AWS Payment Cryptography, firma el certificado del dispositivo y almacena el PVV. ??? AWS Payment Cryptography realiza las operaciones criptográficas. El dispositivo solo se comunica con el servidor del cliente, que realiza todas las llamadas a Payment Cryptography. AWS

Este tema abarca tres flujos:

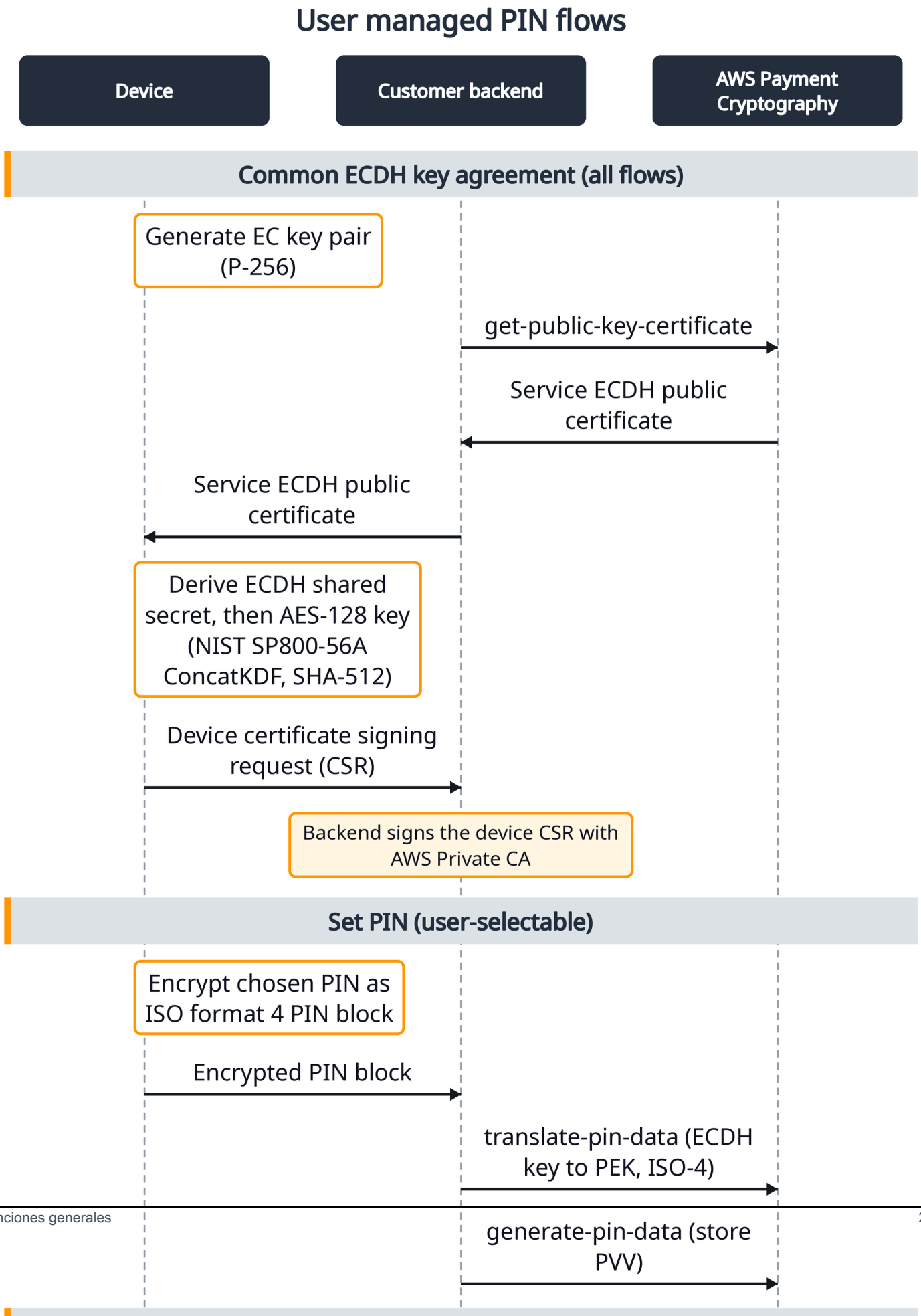
- User-selectable PIN (establecer PIN): el titular de la tarjeta elige su propio PIN en el dispositivo.

- PIN verde (PIN de restablecimiento): el servicio genera un PIN aleatorio que se entrega al titular de la tarjeta, por ejemplo, durante la primera emisión o al restablecer un PIN olvidado.
- Revelar el PIN: un bloque de PIN existente se devuelve a un dispositivo autorizado para casos de uso limitados.

Los tres flujos comparten la misma configuración de claves y la misma mecánica de asociación de dispositivos del ECDH, que se describen en Crear las claves. Para obtener más información sobre el acuerdo de claves en sí, consulte Importación de claves mediante técnicas asimétricas (ECDH).

> **Note**
>
> Estos flujos gestionan los PIN transparentes y los bloqueos de PIN únicamente en el dispositivo, nunca en transición a la criptografía de AWS pagos. Revisa cada flujo detenidamente con tu equipo de seguridad antes de implementarlo.

El siguiente diagrama muestra los flujos de PIN gestionados por los usuarios.


User managed PIN flows
Device
Customer backend
AWS Payment Cryptography
Common ECDH key agreement (all flows)
Generate EC key pair (P-256)
get-public-key-certificate
Service ECDH public certificate
Service ECDH public certificate
Derive ECDH shared secret, then AES-128 key (NIST SP800-56A ConcatKDF, SHA-512)
Device certificate signing request (CSR)
Backend signs the device CSR with AWS Private CA
Set PIN (user-selectable)
Encrypt chosen PIN as ISO format 4 PIN block
Encrypted PIN block
translate-pin-data (ECDH key to PEK, ISO-4)
generate-pin-data (store PVV)

Para ver un ejemplo completo y ejecutable de estos flujos con el AWS SDK para Python, consulte el ejemplo de conjuntos y flujos revelados del ECDH en el repositorio de ejemplos en. AWS GitHub

> **Note**
>
> El flujo típico incluye un dispositivo para el titular de la tarjeta que ejecuta el acuerdo de claves del ECDH y cifra o descifra el bloque de 4 PIN en formato ISO. Si ya proteges los PIN entre el dispositivo y el servidor del cliente por otros medios, puedes usar una versión modificada de estos flujos directamente entre el servidor del cliente y la criptografía de pagos. AWS En esa variante, el servidor del cliente ejecuta el acuerdo de claves del ECDH y las operaciones de bloqueo del PIN en lugar del dispositivo.

Crea las claves

Los tres flujos utilizan el mismo conjunto de claves: una clave de acuerdo ECDH para la clave de acuerdo con el dispositivo, una Generation/Verification clave PIN (PGK) para generar el PVV y una clave de cifrado PIN (PEK) a la que se traduce el bloque PIN para almacenarlo en el servidor. También importas la clave pública de la autoridad de certificación (CA) del dispositivo para que AWS Payment Cryptography confíe en los certificados del dispositivo presentados durante la traducción.

Crea la clave de acuerdo del ECDH. Esta clave se usa para derivar la clave simétrica compartida con el dispositivo.

```
$ aws payment-cryptography create-key \
    --exportable \
    --key-attributes KeyAlgorithm=ECC_NIST_P256,KeyClass=ASYMMETRIC_KEY_PAIR,\
        KeyUsage=TR31_K3_ASYMMETRIC_KEY_FOR_KEY_AGREEMENT,\
        KeyModesOfUse='{DeriveKey=true}'
```

La respuesta repite los parámetros de la solicitud e incluye el ARN de la clave ECDH.

```
{
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-east-1:111122223333:key/
xxxxxxxxxxxxxxx",
        "KeyAttributes": {
            "KeyUsage": "TR31_K3_ASYMMETRIC_KEY_FOR_KEY_AGREEMENT",
            "KeyClass": "ASYMMETRIC_KEY_PAIR",
```

```
            "KeyAlgorithm": "ECC_NIST_P256",
            "KeyModesOfUse": {
                "Encrypt": false,
                "Decrypt": false,
                "Wrap": false,
                "Unwrap": false,
                "Generate": false,
                "Sign": false,
                "Verify": false,
                "DeriveKey": true,
                "NoRestrictions": false
            }
        },
        "KeyCheckValue": "1A2B3C",
        "KeyCheckValueAlgorithm": "CMAC",
        "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"
    }
}
```

Cree la Generation/Verification clave PIN (PGK) utilizada para generar el PVV.

```
$ aws payment-cryptography create-key \
    --exportable \
    --key-attributes KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_V2_VISA_PIN_VERIFICATION_KEY,
\
        KeyClass=SYMMETRIC_KEY,\
        KeyModesOfUse='{Generate=true,Verify=true}'
```

```
{
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ov6icy4ryas4zcza",
        "KeyAttributes": {
            "KeyUsage": "TR31_V2_VISA_PIN_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": "51A200",
        "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"
    }
}
```

Cree la clave de cifrado PIN (PEK) a la que se traduce el bloque PIN del dispositivo para almacenarla en el backend.

```
$ aws payment-cryptography create-key \
    --exportable \
    --key-attributes KeyAlgorithm=AES_128,KeyUsage=TR31_P0_PIN_ENCRYPTION_KEY,\
        KeyClass=SYMMETRIC_KEY,\
        KeyModesOfUse='{Encrypt=true,Decrypt=true,Wrap=true,Unwrap=true}'
```

```
{
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt",
        "KeyAttributes": {
            "KeyUsage": "TR31_P0_PIN_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": "7CC9E2",
        "KeyCheckValueAlgorithm": "CMAC",
        "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"
    }
}
```

Importe la clave pública de la CA del dispositivo para que AWS Payment Cryptography pueda verificar los certificados del dispositivo presentados en llamadas de traducción posteriores. `<base64-CA-cert>`Sustitúyala por el certificado de CA codificado en base64.

```
$ aws payment-cryptography import-key \
    --key-material='{"RootCertificatePublicKey":{"KeyAttributes":
{"KeyAlgorithm":"ECC_NIST_P256","KeyClass":"PUBLIC_KEY","KeyModesOfUse":
{"Verify":true},"KeyUsage":"TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE"},"PublicKeyCertificat
CA-cert>"}}'
```

```
{
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-east-1:111122223333:key/
yyyyyyyyyyyyyyyy",
        "KeyAttributes": {
            "KeyUsage": "TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE",
            "KeyClass": "PUBLIC_KEY",
            "KeyAlgorithm": "ECC_NIST_P256",
            "KeyModesOfUse": {
                "Verify": true
            }
        },
        "KeyCheckValue": "4D5E6F",
        "KeyCheckValueAlgorithm": "CMAC",
```

```
        "Enabled": true,
        "Exportable": false,
        "KeyState": "CREATE_COMPLETE",
        "KeyOrigin": "EXTERNAL",
        "CreateTimestamp": "2023-06-05T06:41:46.648000-07:00",
        "UsageStartTimestamp": "2023-06-05T06:41:46.626000-07:00"
    }
}
```

Tome nota de cada una de las devueltas. `KeyArn` En los ejemplos siguientes se utilizan los siguientes marcadores de posición literales:

- Clave del acuerdo del ECDH: `arn:aws:payment-cryptography:us-east-1:111122223333:key/xxxxxxxxxxxxxxxx`
- Clave pública de CA importada: `arn:aws:payment-cryptography:us-east-1:111122223333:key/yyyyyyyyyyyyyyyy`
- PGK: `arn:aws:payment-cryptography:us-east-2:111122223333:key/ov6icy4ryas4zcza`
- PEK: `arn:aws:payment-cryptography:us-east-2:111122223333:key/ivi5ksfsuplneuyt`

En el dispositivo del titular de la tarjeta (normalmente un teléfono móvil o un dispositivo similar), los siguientes pasos se ejecutan en el código de aplicación del dispositivo para cada uno de los flujos Establecer PIN, Restablecer PIN y Revelar PIN: generar un par de claves EC en la curva P-256; obtener el certificado público ECDH del servicio del backend del cliente; derivar el secreto compartido del ECDH y, a continuación, una AES-128 clave de 16 bytes (utilizando el NIST SP800-56A ConcatKDF (SHA-5 12) con el `SharedInformation` valor; y cifrar o descifrar el bloque de 4 PIN en formato ISO con ese clave derivada. El dispositivo no tiene credenciales de AWS y no llama directamente a AWS Payment Cryptography; intercambia estos datos con el backend del cliente, que realiza las llamadas de servicio.

Como el dispositivo suele ser un teléfono, su par de claves EC y su certificado se aprovisionan la primera vez que se utilizan y se actualizan periódicamente a partir de entonces. El dispositivo genera el par de claves EC y una solicitud de firma de certificado (CSR), y el backend del cliente firma esa CSR mediante Autoridad de certificación privada de AWS una CA similar o una entidad emisora de certificados similar. El backend del cliente utiliza el certificado de dispositivo firmado resultante en sus llamadas de traducción de datos PIN, y la criptografía de AWS pagos lo compara con la clave pública de la CA que ha importado.

El servidor del cliente obtiene el certificado público ECDH del servicio y se lo entrega al dispositivo. El certificado se devuelve como un PEM codificado en base64 en el `KeyCertificate` campo, que se decodifica en base64 en un archivo PEM. El dispositivo usa este certificado como clave pública del servicio cuando obtiene el secreto compartido.

```
$ aws payment-cryptography get-public-key-certificate \
    --key-identifier arn:aws:payment-cryptography:us-east-1:111122223333:key/
xxxxxxxxxxxxxxx
```

> **Note**
>
> `SharedInformation`son 32 bytes aleatorios que se generan en el dispositivo y pasan del servidor del cliente a AWS Payment Cryptography en forma de cadena hexadecimal. Debe coincidir con el valor que el dispositivo usó en el KDF; de lo contrario, las claves derivadas no coincidirán.

Establezca un PIN que el usuario pueda seleccionar

En el flujo Establecer PIN, el titular de la tarjeta elige su propio PIN en el dispositivo (por ejemplo,). `1234` El dispositivo obtiene la clave compartida del ECDH, cifra el PIN elegido en un bloque de 4 PIN en formato ISO con esa clave y envía el bloque PIN cifrado, su solicitud de firma de certificado (CSR) y la información compartida al servidor del cliente. El servidor del cliente firma la CSR y, a continuación, realiza dos llamadas en el plano de datos. En este ejemplo se utiliza la muestra. PAN `171234567890123` Los valores del bloque PIN, el certificado y la información compartida que se muestran son marcadores de posición ficticios.

En primer lugar, traduzca el bloque ECDH-protected PIN al PEK del servidor. La clave entrante es la clave del acuerdo del ECDH, envuelta con Diffie-Hellman los parámetros y el certificado del dispositivo firmado; la clave saliente es el PEK.

```
$ aws payment-cryptography-data translate-pin-data \
    --encrypted-pin-block A1B2C3D4E5F60718 \
    --incoming-key-identifier arn:aws:payment-cryptography:us-east-1:111122223333:key/
xxxxxxxxxxxxxxx \
    --incoming-wrapped-key
 'WrappedKeyMaterial={DiffieHellmanSymmetricKey={CertificateAuthorityPublicKeyIdentifier=arn:aw
cryptography:us-east-1:111122223333:key/
yyyyyyyyyyyyyyyy,KeyAlgorithm=AES_128,KeyDerivationFunction=NIST_SP800,KeyDerivationHashAlgorit
signed-device-
```

```
cert>,SharedInformation=0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef}}'
 \
    --incoming-translation-attributes
 'IsoFormat4={PrimaryAccountNumber=171234567890123}' \
    --outgoing-key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt \
    --outgoing-translation-attributes
 'IsoFormat4={PrimaryAccountNumber=171234567890123}'
```

La respuesta contiene el bloque PIN que se ha vuelto a cifrar con el PEK en formato ISO 4.

```
{
    "PinBlock": "F7E6D5C4B3A29180",
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt",
    "KeyCheckValue": "7CC9E2"
}
```

A continuación, genere el PVV para el PEK-encrypted bloque PIN. El PGK genera el PVV y el PEK identifica la clave con la que se cifra el bloque PIN suministrado.

```
$ aws payment-cryptography-data generate-pin-data \
    --generation-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/ov6icy4ryas4zcza \
    --encryption-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/ivi5ksfsuplneuyt \
    --pin-block-format ISO_FORMAT_4 \
    --primary-account-number 171234567890123 \
    --generation-attributes
 'VisaPinVerificationValue={PinVerificationKeyIndex=1,EncryptedPinBlock=F7E6D5C4B3A29180}'
```

La respuesta contiene la entrada PVV `PinData.VerificationValue` y el bloque PIN cifrado. Guarde el PVV en el servidor del cliente para poder verificar el PIN más adelante. Para verificarlo, consulta Generar un PIN aleatorio y el PVV asociado y, a continuación, verifica el valor.

```
{
    "GenerationKeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ov6icy4ryas4zcza",
    "GenerationKeyCheckValue": "51A200",
    "EncryptionKeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt",
    "EncryptionKeyCheckValue": "7CC9E2",
```

```
    "EncryptedPinBlock": "F7E6D5C4B3A29180",
    "PinData": {
        "VerificationValue": "5507"
    }
}
```

Para obtener más información, consulta TranslatePinData y consulta la guía GeneratePinData de referencia de la API.

Restablecer un PIN (PIN verde)

En el proceso de restablecimiento del PIN (PIN verde), la criptografía de AWS pagos genera un PIN aleatorio que el servidor del cliente devuelve al dispositivo para mostrárselo al titular de la tarjeta, por ejemplo, durante la primera emisión o al restablecer un PIN olvidado. El dispositivo realiza la misma configuración de ECDH descrita en Crear las claves, y el servidor del cliente realiza las llamadas del plano de datos en su nombre. En este ejemplo se utiliza el ejemplo PAN171234567890123: el bloque PIN y los valores de la información compartida son marcadores de posición ficticios.

En primer lugar, genere un PIN aleatorio y su PVV, cifrando el bloque de PIN en el PEK.

```
$ aws payment-cryptography-data generate-pin-data \
    --generation-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/ov6icy4ryas4zcza \
    --encryption-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/ivi5ksfsuplneuyt \
    --pin-block-format ISO_FORMAT_4 \
    --primary-account-number 171234567890123 \
    --generation-attributes 'VisaPin={PinVerificationKeyIndex=1}'
```

La respuesta contiene la entrada PVV PinData.VerificationValue y el bloque PEK-encrypted PIN (formato ISO 4). Guarde el PVV en el servidor del cliente para verificarlo más adelante.

```
{
    "GenerationKeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ov6icy4ryas4zcza",
    "GenerationKeyCheckValue": "51A200",
    "EncryptionKeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt",
    "EncryptionKeyCheckValue": "7CC9E2",
    "EncryptedPinBlock": "F7E6D5C4B3A29180",
    "PinData": {
        "VerificationValue": "5507"
```

```
    }
}
```

A continuación, traduzca el bloque PEK-encrypted PIN (formato ISO 4) a la ECDH-derived clave del dispositivo (formato ISO 4) para que solo el dispositivo solicitante pueda leerlo.

```
$ aws payment-cryptography-data translate-pin-data \
    --encrypted-pin-block F7E6D5C4B3A29180 \
    --incoming-key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt \
    --incoming-translation-attributes
 'IsoFormat4={PrimaryAccountNumber=171234567890123}' \
    --outgoing-key-identifier arn:aws:payment-cryptography:us-east-1:111122223333:key/
xxxxxxxxxxxxxxx \
    --outgoing-wrapped-key
 'WrappedKeyMaterial={DiffieHellmanSymmetricKey={CertificateAuthorityPublicKeyIdentifier=arn:aw
cryptography:us-east-1:111122223333:key/
yyyyyyyyyyyyyyyy,KeyAlgorithm=AES_128,KeyDerivationFunction=NIST_SP800,KeyDerivationHashAlgorit
signed-device-
cert>,SharedInformation=0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef}}'
 \
    --outgoing-translation-attributes
 'IsoFormat4={PrimaryAccountNumber=171234567890123}'
```

La respuesta contiene el bloque PIN que se ha vuelto a cifrar con la ECDH-derived clave del dispositivo en formato ISO 4.

```
{
    "PinBlock": "A1B2C3D4E5F60718",
    "KeyArn": "arn:aws:payment-cryptography:us-east-1:111122223333:key/
xxxxxxxxxxxxxxx",
    "KeyCheckValue": "1A2B3C"
}
```

El servidor del cliente devuelve el bloque ECDH-encrypted PIN al dispositivo. El dispositivo descifra el bloque PIN devuelto con su ECDH-derived clave (en el código de la aplicación del dispositivo) y muestra el PIN al titular de la tarjeta. Como la clave solo se obtiene de ese dispositivo, ninguna otra persona puede descifrar el bloque PIN.

Para obtener más información, consulte GeneratePinData y TranslatePinData en la guía de referencia de la API.

Revela un PIN

El flujo Revelar PIN devuelve un bloque de PIN existente al dispositivo del titular de la tarjeta para que el dispositivo pueda mostrar o revelar el PIN de otro modo. El dispositivo realiza la misma configuración de ECDH descrita en Crear las claves, y el servidor del cliente convierte el bloque de PEK-encrypted PIN almacenado en la ECDH-derived clave y lo devuelve al dispositivo. En este ejemplo se utiliza el ejemplo PAN171234567890123; el bloque PIN y los valores de la información compartida son marcadores de posición ficticios.

> **Note**
>
> Considera la posibilidad de autenticar al titular de la tarjeta y dejar que establezca un nuevo PIN en lugar de revelar el PIN existente. Una excepción notable son los países que utilizan el PIN sin conexión a Internet, por ejemplo, en Europa, donde el PIN también se almacena en la tarjeta con chip. En esos casos, revelar el PIN existente puede ser más sencillo que reprogramar la tarjeta.

Traduce el bloque PEK-encrypted PIN almacenado (formato ISO 4) a la ECDH-derived clave del dispositivo (formato ISO 4).

```
$ aws payment-cryptography-data translate-pin-data \
    --encrypted-pin-block F7E6D5C4B3A29180 \
    --incoming-key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt \
    --incoming-translation-attributes
 'IsoFormat4={PrimaryAccountNumber=171234567890123}' \
    --outgoing-key-identifier arn:aws:payment-cryptography:us-east-1:111122223333:key/
xxxxxxxxxxxxxxx \
    --outgoing-wrapped-key
 'WrappedKeyMaterial={DiffieHellmanSymmetricKey={CertificateAuthorityPublicKeyIdentifier=arn:aw
cryptography:us-east-1:111122223333:key/
yyyyyyyyyyyyyyyy,KeyAlgorithm=AES_128,KeyDerivationFunction=NIST_SP800,KeyDerivationHashAlgorit
signed-device-
cert>,SharedInformation=0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef}}'
 \
    --outgoing-translation-attributes
 'IsoFormat4={PrimaryAccountNumber=171234567890123}'
```

La respuesta contiene el bloque PIN que se ha vuelto a cifrar con la ECDH-derived clave del dispositivo en formato ISO 4.

```
{
    "PinBlock": "A1B2C3D4E5F60718",
    "KeyArn": "arn:aws:payment-cryptography:us-east-1:111122223333:key/
xxxxxxxxxxxxxxx",
    "KeyCheckValue": "1A2B3C"
}
```

El servidor del cliente devuelve el bloque ECDH-encrypted PIN al dispositivo. El dispositivo descifra el bloque PIN devuelto con su ECDH-derived clave (en el código de la aplicación del dispositivo) para revelar el PIN. Como la clave solo se obtiene de ese dispositivo, ninguna otra persona puede descifrar el bloque PIN.

Para obtener más información, consulte la guía TranslatePinData de referencia de la API.

Genera o verifica un CVV para una tarjeta determinada

CVV o CVV1 es un valor que tradicionalmente está incrustado en la banda magnética de una tarjeta. No es lo mismo que el CVV2 (visible para el titular de la tarjeta y que se puede usar para compras en línea).

El primer paso es crear una clave. Para este tutorial, creará una clave CVK 3DES (TDES de 2 teclas) de doble longitud.

> **Note**
>
> CVV, CVV2 e iCVV utilizan algoritmos similares, si no idénticos, pero varían los datos de entrada. Todos usan el mismo tipo de clave TR31_C0_CARD_VERIFICATION_KEY, pero se recomienda usar claves distintas para cada propósito. Estas se pueden distinguir mediante etiquetas de alias, como en el ejemplo siguiente. and/or

Crea la clave

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY,KeyModesO
 --tags='[{"Key":"KEY_PURPOSE","Value":"CVV"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La respuesta refleja los parámetros de la solicitud, incluyendo un ARN para las llamadas posteriores y un valor de verificación clave (KCV).


```
{
            "Key": {
                "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
r52o3wbqxyf6qlqr",
                "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": "DE89F9",
                "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"
            }
        }
```


Tome nota de lo `KeyArn` que representa la clave, por ejemplo arn:aws:payment-cryptography:us-east- 2:111122223333:. key/r52o3wbqxyf6qlqr Lo necesitará en el siguiente paso.

Genera un CVV

Example

En este ejemplo, generaremos un CVV para un PAN determinado con las entradas de PAN 121 (tal como se define en ISO/IEC 7813) y la fecha de caducidad de la tarjeta.

Para ver todos los parámetros disponibles, consulte la CardVerificationValue1 guía de referencia de la API.

```
$ aws payment-cryptography-data generate-card-validation-data --key-
identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
r52o3wbqxyf6qlqr --primary-account-number=171234567890123 --generation-attributes
 CardVerificationValue1='{CardExpiryDate=1127,ServiceCode=121}'
```

```
                {
                    "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/r52o3wbqxyf6qlqr",
                    "KeyCheckValue": "DE89F9",
                    "ValidationData": "801"
                }
```

Valide el CVV

Example

En este ejemplo, verificaremos el CVV de un PAN determinado introduciendo un CVK, un código de servicio 121PAN, la fecha de caducidad de la tarjeta y el CVV proporcionado durante la transacción para su validación.

Para ver todos los parámetros disponibles, consulte la guía de CardVerificationValue1 referencia de la API.

> **Note**
>
> El CVV no es un valor introducido por el usuario (como el CVV2), pero normalmente está incrustado en una banda magnética. Se debe considerar si siempre debe validarse cuando se proporciona.

```
$ aws payment-cryptography-data verify-card-validation-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/r52o3wbqxyf6qlqr
 --primary-account-number=171234567890123 --verification-attributes
 CardVerificationValue1='{CardExpiryDate=1127,ServiceCode=121} --validation-data 801
```

```
{
                    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
r52o3wbqxyf6qlqr",
                    "KeyCheckValue": "DE89F9",
                    "ValidationData": "801"
}
```

Genere o verifique un CVV2 para una tarjeta específica

El CVV2 es un valor que tradicionalmente se indica en el reverso de una tarjeta y se usa para compras en línea. En el caso de las tarjetas virtuales, también puede mostrarse en una aplicación o en una pantalla. Desde el punto de vista criptográfico, es igual que el CVV1 pero con un valor de código de servicio diferente.

Crea la clave

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY,KeyModesO
 --tags='[{"Key":"KEY_PURPOSE","Value":"CVV2"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La respuesta refleja los parámetros de la solicitud, incluyendo un ARN para las llamadas posteriores y un valor de verificación clave (KCV).

```
{
               "Key": {
                   "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/7f7g4spf3xcklhzu",
                   "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": "AEA5CD",
                "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"
            }
        }
```

Tome nota de lo `KeyArn` que representa la clave, por ejemplo arn:aws:payment-cryptography:us-east- 2:111122223333:. key/7f7g4spf3xcklhzu Lo necesitará en el siguiente paso.

Genera un CVV2

Example

En este ejemplo, generaremos un CVV2 para un PAN determinado con las entradas PAN y la fecha de caducidad de la tarjeta.

Para ver todos los parámetros disponibles, consulte CardVerificationValue2 la guía de referencia de la API.

```
$ aws payment-cryptography-data generate-card-validation-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/7f7g4spf3xcklhzu
 --primary-account-number=171234567890123 --generation-attributes
 CardVerificationValue2='{CardExpiryDate=1127}'
```

```
                    {
                        "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/7f7g4spf3xcklhzu",
                        "KeyCheckValue": "AEA5CD",
                        "ValidationData": "321"
                    }
```

Valide un CVV2

Example

En este ejemplo, verificaremos el CVV2 de un PAN determinado introduciendo un CVK, la fecha de caducidad de la tarjeta PAN y el CVV proporcionado durante la transacción para su validación.

Para ver todos los parámetros disponibles, consulte la guía de CardVerificationValue2 referencia de la API.

> **Note**
>
> CVV2 y las demás entradas son valores introducidos por el usuario. Por lo tanto, el hecho de que esto no se valide periódicamente no es necesariamente una señal de un problema.

```
$ aws payment-cryptography-data verify-card-validation-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/7f7g4spf3xcklhzu
 --primary-account-number=171234567890123 --verification-attributes
 CardVerificationValue2='{CardExpiryDate=1127} --validation-data 321
```

```
{
                      "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/7f7g4spf3xcklhzu",
                      "KeyCheckValue": "AEA5CD",
                      "ValidationData": "801"
   }
```

Genere o verifique un iCVV para una tarjeta específica

iCVV usa el mismo algoritmo que CVV/CVV2 pero iCVV está integrado dentro de una tarjeta con chip. Su código de servicio es 999.

Crea la clave

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY,KeyModesO
 --tags='[{"Key":"KEY_PURPOSE","Value":"ICVV"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La respuesta refleja los parámetros de la solicitud, incluyendo un ARN para las llamadas posteriores y un valor de verificación clave (KCV).


```
{
               "Key": {
                   "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
c7dsi763r6s7lfp3",
                   "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": "1201FB",
                   "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"
               }
           }
```


Tome nota de lo `KeyArn` que representa la clave, por ejemplo arn:aws:payment-cryptography:us-east- 2:111122223333:. key/c7dsi763r6s7lfp3 Lo necesitará en el siguiente paso.

Genera un iCVV

Example

En este ejemplo, generaremos un iCVV para un PAN determinado con las entradas de 999PAN, un código de servicio (tal como se define en ISO/IEC 7813) y la fecha de caducidad de la tarjeta.

Para ver todos los parámetros disponibles, consulte la guía CardVerificationValue1 de referencia de la API.

```
$ aws payment-cryptography-data generate-card-validation-data --key-
identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
c7dsi763r6s7lfp3 --primary-account-number=171234567890123 --generation-attributes
 CardVerificationValue1='{CardExpiryDate=1127,ServiceCode=999}'
```

```
                    {
                        "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/c7dsi763r6s7lfp3",
                        "KeyCheckValue": "1201FB",
                        "ValidationData": "532"
                    }
```

Valide iCVV

Example

Para la validación, las entradas son CVK, un código de servicio 999PAN, la fecha de caducidad de la tarjeta y el iCVV proporcionado durante la transacción para validarla.

Para ver todos los parámetros disponibles, consulte la guía de CardVerificationValue1 referencia de la API.

> **Note**
>
> iCVV no es un valor introducido por el usuario (como CVV2), pero normalmente está incrustado en una tarjeta. EMV/chip Se debe considerar si siempre debe validarse cuando se proporciona.

```
$ aws payment-cryptography-data verify-card-validation-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/c7dsi763r6s7lfp3
 --primary-account-number=171234567890123 --verification-attributes
 CardVerificationValue1='{CardExpiryDate=1127,ServiceCode=999} --validation-data 532
```

```
{
                      "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/c7dsi763r6s7lfp3",
                      "KeyCheckValue": "1201FB",
                      "ValidationData": "532"
   }
```

Verifique un ARQC EMV y genere un ARPC

El ARQC (criptograma de solicitud de autorización) es un criptograma generado por una tarjeta EMV (con chip) y que se utiliza para validar los detalles de la transacción, así como el uso de una tarjeta autorizada. Incorpora datos de la tarjeta, del terminal y de la propia transacción.

En el momento de la validación en el backend, se proporcionan las mismas entradas a la criptografía de AWS pagos, el criptograma se recrea internamente y se compara con el valor proporcionado con la transacción. En este sentido, es similar a un MAC. El libro 2 de EMV 4.4 define tres aspectos de esta función: los métodos de derivación de claves (conocidos como clave de sesión común (CSK) para generar claves de transacción únicas, una carga útil mínima y los métodos para generar una respuesta (ARPC).

Los sistemas de tarjetas individuales pueden especificar campos transaccionales adicionales para incorporarlos o el orden en que aparecen esos campos. También existen otros esquemas de derivación específicos del esquema (generalmente obsoletos), que se tratan en otras partes de esta documentación.

Para obtener más información, consulte la guía VerifyCardValidationData de la API.

Crea la clave

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS,KeyClass=SYMMETRIC_KEY,KeyMod
 --tags='[{"Key":"KEY_PURPOSE","Value":"CVN18"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La respuesta refleja los parámetros de la solicitud, incluyendo un ARN para las llamadas posteriores y un valor de verificación clave (KCV).

```
{
                "Key": {
                    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6nl62t5ushfk",
```

```
                "KeyAttributes": {
                    "KeyUsage": "TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS",
                    "KeyClass": "SYMMETRIC_KEY",
                    "KeyAlgorithm": "TDES_2KEY",
                    "KeyModesOfUse": {
                        "Encrypt": false,
                        "Decrypt": false,
                        "Wrap": false,
                        "Unwrap": false,
                        "Generate": false,
                        "Sign": false,
                        "Verify": false,
                        "DeriveKey": true,
                        "NoRestrictions": false
                    }
                },
                "KeyCheckValue": "08D7B4",
                "KeyCheckValueAlgorithm": "ANSI_X9_24",
                "Enabled": true,
                "Exportable": true,
                "KeyState": "CREATE_COMPLETE",
                "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
                "CreateTimestamp": "2024-03-07T06:41:46.648000-07:00",
                "UsageStartTimestamp": "2024-03-07T06:41:46.626000-07:00"
            }
        }
```

Tome nota de lo `KeyArn` que representa la clave, por ejemplo arn:aws:payment-cryptography:us-east- 2:111122223333:. key/pw3s6nl62t5ushfk Lo necesitará en el siguiente paso.

Genera un ARQC

El ARQC se genera exclusivamente con una tarjeta EMV. Por lo tanto, la criptografía AWS de pagos no tiene ninguna facilidad para generar dicha carga útil. Para fines de prueba, hay varias bibliotecas disponibles en línea que pueden generar una carga útil adecuada, así como valores conocidos que generalmente proporcionan los distintos esquemas.

Valide un ARQC

Example

Si la criptografía de AWS pagos puede validar el ARQC, se devuelve un. http/200 Opcionalmente, se puede proporcionar un ARPC (respuesta) e incluirlo en la respuesta una vez validado el ARQC.

```
$ 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
 00000000170000000000000008400080008000084016051700000000093800000B1F220103000000000000000000000
 --session-key-derivation-attributes='{"EmvCommon":
{"ApplicationTransactionCounter":"000B",
 "PanSequenceNumber":"01","PrimaryAccountNumber":"9137631040001422"}}' --auth-response-
attributes='{"ArpcMethod2":{"CardStatusUpdate":"12345678"}}'
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6nl62t5ushfk",
    "KeyCheckValue": "08D7B4",
    "AuthResponseValue":"2263AC85"
}
```

Genere y verifique un MAC EMV

El MAC de EMV es un MAC que utiliza la entrada de una clave derivada de EMV y, a continuación, ejecuta un MAC ISO9797-3 (minorista) sobre los datos resultantes. El EMV MAC se utiliza normalmente para enviar comandos a una tarjeta EMV, como los scripts de desbloqueo.

> **Note**
>
> AWS La criptografía de pagos no valida el contenido del script. Consulte el manual de su esquema o tarjeta para obtener más información sobre los comandos específicos que debe incluir.

Para obtener más información, consulta MacAlgorithmEmv la guía de la API.

Temas

- Crea la clave
- Genere un MAC EMV

Crea la clave

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_E2_EMV_MKEY_INTEGRITY,KeyClass=SYMMETRIC_KEY,KeyModesOfUs
 --tags='[{"Key":"KEY_PURPOSE","Value":"CVN18"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La respuesta refleja los parámetros de la solicitud, incluyendo un ARN para las llamadas posteriores y un valor de verificación clave (KCV).

```
{
        "Key": {
            "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6nl62t5ushfk",
            "KeyAttributes": {
                "KeyUsage": "TR31_E2_EMV_MKEY_INTEGRITY",
                "KeyClass": "SYMMETRIC_KEY",
                "KeyAlgorithm": "TDES_2KEY",
                "KeyModesOfUse": {
                    "Encrypt": false,
                    "Decrypt": false,
                    "Wrap": false,
                    "Unwrap": false,
                    "Generate": false,
                    "Sign": false,
                    "Verify": false,
                    "DeriveKey": true,
                    "NoRestrictions": false
                }
            },
            "KeyCheckValue": "08D7B4",
            "KeyCheckValueAlgorithm": "ANSI_X9_24",
            "Enabled": true,
            "Exportable": true,
            "KeyState": "CREATE_COMPLETE",
            "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
            "CreateTimestamp": "2024-03-07T06:41:46.648000-07:00",
            "UsageStartTimestamp": "2024-03-07T06:41:46.626000-07:00"
        }
    }
```

Tome nota de lo `KeyArn` que representa la clave, por ejemplo arn:aws:payment-cryptography:us-east- 2:111122223333:. key/pw3s6nl62t5ushfk Lo necesitará en el siguiente paso.

Genere un MAC EMV

Lo habitual es que un proceso de backend genere un script EMV (como Card Unblock), lo firme con este comando (que obtiene una clave única específica para una tarjeta en particular) y, a continuación, devuelva el MAC. A continuación, se envía el comando + MAC a la tarjeta que se va a aplicar. Enviar el comando a la tarjeta está fuera del alcance de la criptografía de AWS pagos.

> **Note**
>
> Este comando está destinado a los comandos cuando no se envían datos cifrados (como el PIN). EMV Encrypt se puede combinar con este comando para agregar datos cifrados al script del emisor antes de ejecutar este comando

Datos del mensaje

Los datos del mensaje incluyen el encabezado y el comando de la APDU. Si bien esto puede variar según la implementación, este ejemplo es el encabezado APDU para el desbloqueo (84 24 00 00 08), seguido del ATC (0007) y, por último, del ARQC de la transacción anterior (999E57FD0F47CACE). El servicio no valida el contenido de este campo.

Modo de derivación de claves de sesión

Este campo define cómo se genera la clave de sesión. EMV_COMMON_SESSION_KEY se usa generalmente para las nuevas implementaciones, mientras que también se puede usar EMV2000 | AMEX | MASTERCARD_SESSION_KEY | VISA.

MajorKeyDerivationMode

EMV define los modos A, B o C. El modo A es el más común y la criptografía de pagos actualmente admite el modo A o el modo B. AWS

PAN

El número de cuenta suele estar disponible en el campo 5A del chip o en el campo 2 según la norma ISO8583, pero también se puede recuperar del sistema de tarjetas.

PSN

El número de secuencia de la tarjeta. Si no se usa, introduzca 00.

SessionKeyDerivationValue

Estos son los datos de derivación por sesión. Puede ser el último ARQC (ApplicationCryptogram) del campo 9F26 o el último ATC del 9F36, según el esquema de derivación.

Rellenado

El relleno se aplica automáticamente y utiliza el método de relleno 2 9797-1. ISO/IEC

Example

```
$ aws payment-cryptography-data generate-mac --message-data
 842400000800079 99E57FD0F47CACE --key-identifier arn:aws:payment-
cryptography:us-east-2:111122223333:key/pw3s6nl62t5ushfk --message-
data 8424000008999E57FD0F47CACE0007 --generation-attributes
 EmvMac="{MajorKeyDerivationMode=EMV_OPTION_A,PanSequenceNumber='00',PrimaryAccountNumber='2235
```

```
{
"KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/pw3s6nl62t5ushfk",
"KeyCheckValue": "08D7B4",
"Mac":"5652EEDF83EA0D84"
}
```

Genere un MAC EMV para cambiar el PIN

El cambio de PIN de EMV combina dos operaciones: generar un código MAC para un script del emisor y cifrar un nuevo PIN para cambiarlo sin conexión a Internet en una tarjeta con chip EMV. Este comando solo es necesario en algunos países en los que el PIN está almacenado en la tarjeta con chip (esto es habitual en los países europeos). Suele utilizarse cuando el titular de una tarjeta necesita cambiar su PIN y el nuevo PIN debe transmitirse de forma segura a la tarjeta junto con un MAC para verificar la autenticidad del comando.

> **Note**
>
> Si solo necesitas enviar comandos a la tarjeta pero no cambiar el PIN, considera usar en su lugar los comandos ARPC CSU o Generate EMV MAC.

Para obtener más información, consulta la guía de GenerateMacEmvPinChange la API.

Genere el MAC EMV y el PIN cifrado para cambiar el PIN

Esta operación requiere dos claves: una clave de integridad EMV (: TR31_E2_EMV_MKEY_INTEGRITY) para la generación de MAC y una clave de confidencialidad EMV (KeyUsage: TR31_E4_EMV_MKEY_CONFIDENCIALITY) para el cifrado del PIN. KeyUsage Lo habitual es que un proceso de backend genere un script de cambio de PIN de EMV, que incluye tanto el MAC del script del emisor como el nuevo PIN cifrado. A continuación, el comando y el PIN cifrado se envían a la tarjeta para actualizar el PIN sin conexión. Enviar el comando a la tarjeta está fuera del alcance de la criptografía de AWS pagos.

Datos del mensaje

Los datos del mensaje incluyen el comando APDU para el script del emisor. El servicio no valida el contenido de este campo.

Nuevo bloque de PIN cifrado

El nuevo bloque de PIN cifrado que se enviará a la tarjeta. Debe proporcionarse como un valor cifrado mediante una clave de cifrado PIN.

Nuevo identificador PIN PEK

La clave utilizada para cifrar el nuevo PIN antes de pasarlo a esta API.

Clave de integridad de mensajería segura

La clave de integridad EMV (KeyUsage: TR31_E2_EMV_MKEY_INTEGRITY) utilizada para la generación de MAC.

Clave de confidencialidad de mensajería segura

La clave de confidencialidad de EMV (KeyUsage: TR31_E4_EMV_MKEY_CONFIDENCIALITY) utilizada para el cifrado del PIN.

MajorKeyDerivationMode

EMV define los modos A, B o C. El modo A es el más común y la criptografía de pagos actualmente admite el modo A o el modo B. AWS

Mode

El modo de cifrado, normalmente CBC para las operaciones de cambio de PIN.

PAN

El número de cuenta, normalmente disponible en el campo 5A del chip o en el campo 2 según la norma ISO8583, pero también se puede recuperar del sistema de tarjetas.

PanSequenceNumber

El número de secuencia de la tarjeta. Si no se usa, introduzca 00.

ApplicationCryptogram

Estos son los datos de derivación por sesión, normalmente el último ARQC del campo 9F26.

PinBlockLengthPosition

Especifica dónde se codifica la longitud del bloque PIN. Normalmente se establece en NINGUNO. Consulta las especificaciones de tu esquema de tarjetas si no estás seguro.

PinBlockPaddingType

Especifica el tipo de relleno del bloque PIN. Normalmente se establece en NO_PADDING. Consulta las especificaciones de tu esquema de tarjetas si no estás seguro.

Example

```
$ aws payment-cryptography-data generate-mac-emv-pin-change \
    --message-data 00A4040008A000000004101080D80500000001010A04000000000000 \
    --new-encrypted-pin-block 67FB27C75580EFE7 \
    --new-pin-pek-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt \
    --pin-block-format ISO_FORMAT_0 \
    --secure-messaging-confidentiality-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/tqv5yij6wtxx64pi \
    --secure-messaging-integrity-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/pw3s6nl62t5ushfk \
    --derivation-method-attributes
 'EmvCommon={ApplicationCryptogram=1234567890123457,MajorKeyDerivationMode=EMV_OPTION_A,Mode=CB
```

```
{
    "SecureMessagingIntegrityKeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/pw3s6nl62t5ushfk",
    "SecureMessagingIntegrityKeyCheckValue": "08D7B4",
    "SecureMessagingConfidentialityKeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/tqv5yij6wtxx64pi",
    "SecureMessagingConfidentialityKeyCheckValue": "C1EB8F",
    "Mac": "5652EEDF83EA0D84",
    "EncryptedPinBlock": "F1A2B3C4D5E6F7A8"
}
```

Funciones específicas de la red

Temas

- Funciones específicas de visado
- Funciones específicas de Mastercard
- Funciones específicas de American Express
- Funciones específicas de JCB
- UnionPay funciones específicas

Funciones específicas de visado

Temas

- ARQC - CVN18/CVN22
- ARQC - CVN10
- 3DS CAV V7
- dCVV (valor dinámico de verificación de la tarjeta) - CVN17

ARQC - CVN18/CVN22

CVN18 y CVN22 utilizan el método CSK de derivación de claves. Los datos exactos de la transacción varían entre estos dos métodos. Consulte la documentación del esquema para obtener más información sobre cómo crear el campo de datos de la transacción.

ARQC - CVN10

El CVN10 es un método antiguo de Visa para las transacciones EMV, que utiliza la derivación por clave de tarjeta en lugar de la derivación por sesión (por transacción) y también utiliza una carga útil diferente. Para obtener información sobre el contenido de la carga útil, póngase en contacto con el esquema para obtener más información.

Crear clave

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS,KeyClass=SYMMETRIC_KEY,KeyMod
 --tags='[{"Key":"KEY_PURPOSE","Value":"CVN10"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La respuesta refleja los parámetros de la solicitud, incluyendo un ARN para las llamadas posteriores y un valor de verificación clave (KCV).


```
{
            "Key": {
                "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6nl62t5ushfk",
                "KeyAttributes": {
                    "KeyUsage": "TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS",
                    "KeyClass": "SYMMETRIC_KEY",
                    "KeyAlgorithm": "TDES_2KEY",
                    "KeyModesOfUse": {
                        "Encrypt": false,
                        "Decrypt": false,
                        "Wrap": false,
                        "Unwrap": false,
                        "Generate": false,
                        "Sign": false,
                        "Verify": false,
                        "DeriveKey": true,
                        "NoRestrictions": false
                    }
                },
                "KeyCheckValue": "08D7B4",
                "KeyCheckValueAlgorithm": "ANSI_X9_24",
                "Enabled": true,
                "Exportable": true,
                "KeyState": "CREATE_COMPLETE",
                "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
                "CreateTimestamp": "2024-03-07T06:41:46.648000-07:00",
                "UsageStartTimestamp": "2024-03-07T06:41:46.626000-07:00"
            }
        }
```


Tome nota de lo `KeyArn` que representa la clave, por ejemplo arn:aws:payment-cryptography:us-east- 2:111122223333:. key/pw3s6nl62t5ushfk Lo necesitará en el siguiente paso.

Valide el ARQC

Example

En este ejemplo, validaremos un ARQC generado con Visa CVN10.

Si la criptografía de AWS pagos puede validar el ARQC, se devuelve un. http/200 Si el arqc no está validado, devolverá una respuesta. 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
 00000000170000000000000008400080008000084016051700000000093800000B03011203000000 \
  --session-key-derivation-attributes='{"Visa":{"PanSequenceNumber":"01" \
  ,"PrimaryAccountNumber":"9137631040001422"}}'
```

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

3DS CAV V7

Para las transacciones con Visa Secure (3DS), el servidor de control de acceso (ACS) del emisor genera un CAVV (valor de verificación de la autenticación del titular de la tarjeta). El CAVV es una prueba de que se ha realizado la autenticación del titular de la tarjeta, es único para cada transacción de autenticación y lo proporciona la entidad adquirente en el mensaje de autorización. La versión 7 del CAVV vincula datos adicionales sobre la transacción a la aprobación, incluidos elementos como el nombre del comerciante, el importe de la compra y la fecha de compra. De este modo, se trata, en efecto, de un hash criptográfico de la carga útil de la transacción.

Criptográficamente, la CAVV V7 utiliza el algoritmo CVV, pero todas las entradas lo han hecho. changed/repurposed Consulte la tercera party/Visa documentación correspondiente sobre cómo producir las entradas para generar una carga útil del CAVV V7.

Crea la clave

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY,KeyModesO
 --tags='[{"Key":"KEY_PURPOSE","Value":"CAVV-V7"},
{"Key":"CARD_BIN","Value":"12345678"}]'
```

La respuesta refleja los parámetros de la solicitud, incluyendo un ARN para las llamadas posteriores y un valor de verificación clave (KCV).


```
{
            "Key": {
                "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
dnaeyrjgdjjtw6dk",
                "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": "F3FB13",
                "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"
            }
        }
```


Tome nota de lo `KeyArn` que representa la clave, por ejemplo arn:aws:payment-cryptography:us-east- 2:111122223333:. key/dnaeyrjgdjjtw6dk Lo necesitará en el siguiente paso.

Genere un CAVV V7

Example

En este ejemplo, generaremos un CAVV V7 para una transacción determinada con las entradas especificadas en las especificaciones. Tenga en cuenta que para este algoritmo, los campos pueden

reutilizarse, used/re por lo que no se debe suponer que las etiquetas de los campos coinciden con las entradas.

Para ver todos los parámetros disponibles, consulte la CardVerificationValue1 guía de referencia de la API.

```
$ aws payment-cryptography-data generate-card-validation-data --key-
identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
dnaeyrjgdjjtw6dk --primary-account-number=171234567890123 --generation-attributes
 CardVerificationValue1='{CardExpiryDate=9431,ServiceCode=431}'
```

```
            {
                "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
dnaeyrjgdjjtw6dk",
                "KeyCheckValue": "F3FB13",
                "ValidationData": "491"
            }
```

Valide CAVV V7

Example

Para la validación, las entradas son el CVK, los valores de entrada calculados y el CAVV proporcionado durante la transacción que se va a validar.

Para ver todos los parámetros disponibles, consulte la guía de CardVerificationValue1 referencia de la API.

> **Note**
>
> El CAVV no es un valor introducido por el usuario (como el CVV2), sino que lo calcula el ACS del emisor. Se debe considerar si siempre debe validarse cuando se proporciona.

```
$ aws payment-cryptography-data verify-card-validation-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/dnaeyrjgdjjtw6dk
```

```
 --primary-account-number=171234567890123 --verification-attributes
 CardVerificationValue1='{CardExpiryDate=9431,ServiceCode=431} --validation-data 491
```

```
{
                "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
dnaeyrjgdjjtw6dk",
                "KeyCheckValue": "F3FB13",
                "ValidationData": "491"
}
```

dCVV (valor dinámico de verificación de la tarjeta) - CVN17

El dCVV (valor dinámico de verificación de tarjeta) es un criptograma dinámico que se utiliza para transacciones EMV Visa-specific sin contacto. Se conoce como EMV primitivo y proporciona una seguridad mejorada al generar un valor de verificación único para cada transacción. El dCVV utiliza entradas como el número de cuenta principal (PAN), el número de secuencia PAN (PSN), el contador de transacciones de la aplicación (ATC), la fecha de caducidad de la tarjeta y el código de servicio. Todavía se usa en algunos lugares, pero en su mayoría ha sido reemplazado por otros algoritmos como el CVN18.

Para ver todos los parámetros disponibles, consulte la DynamicCardVerificationValue guía de referencia de la API.

Las entradas se `DynamicCardVerificationValue` asignan a los siguientes datos de autorización de Visa y transacciones EMV:

`ApplicationTransactionCounter`(ATC)

El contador de transacciones mantenido por el chip, incrementado para cada transacción. En los datos EMV, esto es una etiqueta. `9F36`

`PanSequenceNumber`

El número de secuencia PAN que distingue a las tarjetas que comparten el mismo PAN contenido. En los datos EMV, esto es una etiqueta`5F34`.

`CardExpiryDate`

La fecha de caducidad de la solicitud de la tarjeta (etiqueta EMV`5F24`), proporcionada en el mismo formato utilizado en las demás llamadas de verificación de la tarjeta.

`ServiceCode`

El código de servicio de tres dígitos incluido en los datos de seguimiento 2 (equivalentes) de la tarjeta.

Crear clave

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_E4_EMV_MKEY_DYNAMIC_NUMBERS,KeyClass=SYMMETRIC_KEY,KeyMod
 --tags='[{"Key":"KEY_PURPOSE","Value":"DCVV"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La respuesta refleja los parámetros de la solicitud, incluyendo un ARN para las llamadas posteriores y un valor de verificación clave (KCV).

```
{
            "Key": {
                "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
mw7dn3qxvkfh8ztc",
                "KeyAttributes": {
                    "KeyUsage": "TR31_E4_EMV_MKEY_DYNAMIC_NUMBERS",
                    "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": "A8E4D2",
                "KeyCheckValueAlgorithm": "ANSI_X9_24",
                "Enabled": true,
                "Exportable": true,
                "KeyState": "CREATE_COMPLETE",
                "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
                "CreateTimestamp": "2025-02-02T11:45:30.648000-08:00",
                "UsageStartTimestamp": "2025-02-02T11:45:30.626000-08:00"
```

```
            }
        }
```

Tome nota de lo `KeyArn` que representa la clave, por ejemplo arn:aws:payment-cryptography:us-east- 2:111122223333:. key/mw7dn3qxvkfh8ztc Lo necesitará en el siguiente paso.

Genera un dCVV

Example

En este ejemplo, generaremos un dCVV para una transacción EMV sin contacto. Las entradas incluyen el PAN, el número de secuencia PAN, el contador de transacciones de la aplicación, la fecha de caducidad de la tarjeta y el código de servicio.

```
$ aws payment-cryptography-data generate-card-validation-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/mw7dn3qxvkfh8ztc \
    --primary-account-number=4111112627662122 \
    --generation-attributes
 DynamicCardVerificationValue='{ApplicationTransactionCounter=01,PanSequenceNumber=00,CardExpir
 \
    --validation-data-length 5
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
mw7dn3qxvkfh8ztc",
    "KeyCheckValue": "A8E4D2",
    "ValidationData": "36667"
}
```

Valide el dCVV

Example

En este ejemplo, validaremos un dCVV proporcionado durante una transacción. Para la validación, se deben proporcionar las mismas entradas utilizadas para la generación.

Si la criptografía de AWS pagos se puede validar, http/200 se devuelve una. Si el valor no está validado, devolverá una http/400 respuesta.

```
$ aws payment-cryptography-data verify-card-validation-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/mw7dn3qxvkfh8ztc \
    --primary-account-number=4111112627662122 \
```

```
    --validation-data=36667 \
    --verification-attributes
 DynamicCardVerificationValue='{ApplicationTransactionCounter=01,PanSequenceNumber=00,CardExpir
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
mw7dn3qxvkfh8ztc",
    "KeyCheckValue": "A8E4D2"
}
```

Funciones específicas de Mastercard

Temas

- DCVC3
- ARQC -/ CVN14CVN15
- ARQC -/ CVN12CVN13
- SPA2 3DS AAV

DCVC3

DCVC3 es anterior a los CVN12 esquemas EMV, CSK y Mastercard y representa otro enfoque para utilizar claves dinámicas. A veces también se reutiliza para otros casos de uso. En este esquema, las entradas son datos PAN, PSN, Track1/Track2, un contador impredecible de números y transacciones (ATC).

Crear clave

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_E4_EMV_MKEY_DYNAMIC_NUMBERS,KeyClass=SYMMETRIC_KEY,KeyMod
 --tags='[{"Key":"KEY_PURPOSE","Value":"DCVC3"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La respuesta refleja los parámetros de la solicitud, incluyendo un ARN para las llamadas posteriores y un valor de verificación clave (KCV).

```
{
           "Key": {
               "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
hrh6qgbi3sk4y3wq",
               "KeyAttributes": {
```

```
                    "KeyUsage": "TR31_E4_EMV_MKEY_DYNAMIC_NUMBERS",
                    "KeyClass": "SYMMETRIC_KEY",
                    "KeyAlgorithm": "TDES_2KEY",
                    "KeyModesOfUse": {
                        "Encrypt": false,
                        "Decrypt": false,
                        "Wrap": false,
                        "Unwrap": false,
                        "Generate": false,
                        "Sign": false,
                        "Verify": false,
                        "DeriveKey": true,
                        "NoRestrictions": false
                    }
                },
                "KeyCheckValue": "08D7B4",
                "KeyCheckValueAlgorithm": "ANSI_X9_24",
                "Enabled": true,
                "Exportable": true,
                "KeyState": "CREATE_COMPLETE",
                "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
                "CreateTimestamp": "2024-03-07T06:41:46.648000-07:00",
                "UsageStartTimestamp": "2024-03-07T06:41:46.626000-07:00"
            }
        }
```

Tome nota de lo `KeyArn` que representa la clave, por ejemplo arn:aws:payment-cryptography:us-east- 2:111122223333:key/hrh6qgbi3sk4y3wq. Lo necesitará en el siguiente paso.

Genera un DCVC3

Example

Aunque normalmente DCVC3 se genera mediante una tarjeta con chip, también se puede generar manualmente, como en este ejemplo

```
$ aws payment-cryptography-data generate-card-validation-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/pw3s6nl62t5ushfk
 --primary-account-number=5413123456784808 --generation-attributes
 DynamicCardVerificationCode='{ApplicationTransactionCounter=0000,TrackData=524106000000069D13
```

```
{
```

```
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6nl62t5ushfk",
    "KeyCheckValue": "08D7B4",
    "ValidationData": "865"
  }
```

Valide el DCVC3

Example

En este ejemplo, validaremos un DCVC3. Tenga en cuenta que el ATC debe proporcionarse como un número hexadecimal, por ejemplo, un contador de 11 debe representarse como 000B. El servicio espera un valor de 3 dígitos DCVC3, por lo que si ha almacenado un valor de 4 (o 5) dígitos, simplemente trunque los caracteres de la izquierda hasta que tenga 3 dígitos (por ejemplo, 15321 debería dar como resultado un valor de 321 para los datos de validación).

Si la criptografía AWS de pagos puede validarse, se devolverá un http/200. Si el valor no se valida, devolverá una respuesta http/400.

```
$ aws payment-cryptography-data verify-card-validation-data  --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/pw3s6nl62t5ushfk
 --primary-account-number=5413123456784808 --verification-attributes
 DynamicCardVerificationCode='{ApplicationTransactionCounter=000B,TrackData=524106000000069D13
 --validation-data 398
```

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

ARQC -/ CVN14CVN15

CVN14 y CVN15 utilice el método EMV CSK de derivación de claves. Los datos exactos de las transacciones varían entre estos dos métodos; consulte la documentación del esquema para obtener más información sobre cómo crear el campo de datos de transacciones.

ARQC -/ CVN12CVN13

CVN12 y CVN13 son un método antiguo específico de MasterCard para transacciones EMV que incorpora un número impredecible en la derivación por transacción y también utiliza una carga útil

diferente. Para obtener información sobre el contenido de la carga útil, póngase en contacto con el programa.

Crear clave

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS,KeyClass=SYMMETRIC_KEY,KeyMod
 --tags='[{"Key":"KEY_PURPOSE","Value":"CVN12"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La respuesta refleja los parámetros de la solicitud, incluyendo un ARN para las llamadas posteriores y un valor de verificación clave (KCV).

```
{
            "Key": {
                "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6nl62t5ushfk",
                "KeyAttributes": {
                    "KeyUsage": "TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS",
                    "KeyClass": "SYMMETRIC_KEY",
                    "KeyAlgorithm": "TDES_2KEY",
                    "KeyModesOfUse": {
                        "Encrypt": false,
                        "Decrypt": false,
                        "Wrap": false,
                        "Unwrap": false,
                        "Generate": false,
                        "Sign": false,
                        "Verify": false,
                        "DeriveKey": true,
                        "NoRestrictions": false
                    }
                },
                "KeyCheckValue": "08D7B4",
                "KeyCheckValueAlgorithm": "ANSI_X9_24",
                "Enabled": true,
                "Exportable": true,
                "KeyState": "CREATE_COMPLETE",
                "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
                "CreateTimestamp": "2024-03-07T06:41:46.648000-07:00",
                "UsageStartTimestamp": "2024-03-07T06:41:46.626000-07:00"
            }
        }
```

Tome nota de lo `KeyArn` que representa la clave, por ejemplo arn:aws:payment-cryptography:us-east- 2:111122223333:key/pw3s6nl62t5ushfk. Lo necesitará en el siguiente paso.

Valide el ARQC

Example

En este ejemplo, validaremos un ARQC generado con Mastercard. CVN12

Si la criptografía de AWS pagos puede validar el ARQC, se devuelve un http/200. Si el arqc no se valida, devolverá una respuesta http/400.

```
$ aws payment-cryptography-data verify-auth-request-cryptogram --auth-request-
cryptogram 31BE5D49F14A5F01 \
  --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6nl62t5ushfk \
  --major-key-derivation-mode EMV_OPTION_A \
  --transaction-data
 00000000170000000000000008400080008000084016051700000000093800000B1F220103000000000000000000
 \
  --session-key-derivation-attributes='{"MastercardSessionKey":
{"ApplicationTransactionCounter":"000B","PanSequenceNumber":"01","PrimaryAccountNumber":"541312
```

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

SPA2 3DS AAV

SPA2(Aplicación de pago seguro) El AAV (valor de autenticación de la cuenta) se utiliza para las transacciones 3DS con Mastercard (también conocidas como Mastercard Idenity Check). Proporciona autenticación criptográfica para las transacciones de comercio electrónico mediante la generación de MAC basada en HMAC. El AAV se genera utilizando datos específicos de la transacción y una clave secreta compartida.

Crear clave

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=HMAC_SHA256,KeyUsage=TR31_M7_HMAC_KEY,KeyClass=SYMMETRIC_KEY,KeyModesOfUse='{Gene
```

```
 --tags='[{"Key":"KEY_PURPOSE","Value":"SPA2_AAV"},
{"Key":"CARD_BIN","Value":"12345678"}]'
```

La respuesta refleja los parámetros de la solicitud, incluyendo un ARN para las llamadas posteriores y un valor de verificación clave (KCV).

```
{
            "Key": {
                "KeyArn": "arn:aws:payment-cryptography:us-west-2:111122223333:key/
q5vjtshsg67cz5gn",
                "KeyAttributes": {
                    "KeyUsage": "TR31_M7_HMAC_KEY",
                    "KeyClass": "SYMMETRIC_KEY",
                    "KeyAlgorithm": "HMAC_SHA256",
                    "KeyModesOfUse": {
                        "Encrypt": false,
                        "Decrypt": false,
                        "Wrap": false,
                        "Unwrap": false,
                        "Generate": true,
                        "Sign": false,
                        "Verify": true,
                        "DeriveKey": false,
                        "NoRestrictions": false
                    }
                },
                "KeyCheckValue": "C661F9",
                "KeyCheckValueAlgorithm": "HMAC",
                "Enabled": true,
                "Exportable": true,
                "KeyState": "CREATE_COMPLETE",
                "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
                "CreateTimestamp": "2024-03-07T06:41:46.648000-07:00",
                "UsageStartTimestamp": "2024-03-07T06:41:46.626000-07:00"
            }
        }
```

Tome nota de lo `KeyArn` que representa la clave, por ejemplo arn:aws:payment-cryptography:us-west- 2:111122223333:key/q5vjtshsg67cz5gn. Lo necesitará en el siguiente paso.

Genera AAV SPA2

Example

En este ejemplo, generaremos el componente del valor de autenticación del emisor (IAV) del AAV mediante la generación de SPA2 HMAC MAC. Los datos del mensaje contienen la información específica de la transacción que se autenticará. El formato de los datos del mensaje debe seguir las SPA2 especificaciones de Mastercard y no se describe en este ejemplo.

> **Note**
>
> Revise las especificaciones de formato de su tarjeta Mastercard para insertar el IAV en el valor del AAV.

```
$ aws payment-cryptography-data generate-mac --key-identifier arn:aws:payment-
cryptography:us-west-2:111122223333:key/q5vjtshsg67cz5gn --message-data
 "2226400099919520FFFFd8b448be65694fe7b42f836bad396e9d" --generation-attributes
 Algorithm=HMAC --region us-west-2
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:111122223333:key/
q5vjtshsg67cz5gn",
    "KeyCheckValue": "C661F9",
    "Mac": "6FB2405E9D8A4C1F7B173F73ADD1A6DC358531CAB0E9994FC5B62012ADDE91FC"
}
```

Verifica el AAV SPA2

Example

En este ejemplo, verificaremos un SPA2 AAV. Para la verificación, se proporcionan los mismos datos del mensaje y el mismo valor MAC.

Si la criptografía de AWS pago puede validar el MAC, se devuelve un http/200. Si el MAC no está validado, devolverá una respuesta http/400.

```
$ aws payment-cryptography-data verify-mac --key-identifier arn:aws:payment-
cryptography:us-west-2:111122223333:key/q5vjtshsg67cz5gn --message-
data "2226400099919520FFFFd8b448be65694fe7b42f836bad396e9d" --mac
```

```
 "6FB2405E9D8A4C1F7B173F73ADD1A6DC358531CAB0E9994FC5B62012ADDE91FC" --verification-
attributes Algorithm=HMAC --region us-west-2
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:111122223333:key/
q5vjtshsg67cz5gn",
    "KeyCheckValue": "C661F9"
}
```

Funciones específicas de American Express

Temas

- CSC1
- CSC2
- iCSC
- 3D AEVV

CSC1

La versión 1 de CSC también se conoce como algoritmo CSC clásico. El servicio puede proporcionarlo como un número de 3,4 o 5 dígitos.

Para ver todos los parámetros disponibles, consulte el AmexCardSecurityCodeVersionpunto 1 en la guía de referencia de la API.

Crear clave

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY,KeyModesO
 --tags='[{"Key":"KEY_PURPOSE","Value":"CSC1"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La respuesta refleja los parámetros de la solicitud, incluyendo un ARN para las llamadas posteriores y un valor de verificación clave (KCV).

```
{
               "Key": {
                   "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
esh6hn7pxdtttzgq",
                   "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": "8B5077",
                "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"
            }
        }
```

Tome nota de lo `KeyArn` que representa la clave, por ejemplo arn:aws:payment-cryptography:us-east- 2:111122223333:key/esh6hn7pxdtttzgq. Lo necesitará en el siguiente paso.

Genera un CSC1

Example

```
$ aws payment-cryptography-data generate-card-validation-data --key-
identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
esh6hn7pxdtttzgq --primary-account-number=344131234567848 --generation-attributes
 AmexCardSecurityCodeVersion1='{CardExpiryDate=1224}' --validation-data-length 4
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
esh6hn7pxdtttzgq",
    "KeyCheckValue": "8B5077",
    "ValidationData": "3938"
```

```
  }
```

Valide el CSC1

Example

En este ejemplo, validaremos un CSC1.

Si la criptografía de AWS pago puede validarse, se devuelve un http/200. Si el valor no se valida, devolverá una respuesta http/400.

```
$ aws payment-cryptography-data verify-card-validation-data  --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/esh6hn7pxdtttzgq
 --primary-account-number=344131234567848 --verification-attributes
 AmexCardSecurityCodeVersion1='{CardExpiryDate=1224}''  --validation-data 3938
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
esh6hn7pxdtttzgq",
    "KeyCheckValue": "8B5077"
  }
```

CSC2

La versión 2 del CSC también se conoce como algoritmo CSC mejorado. El servicio puede proporcionarlo como un número de 3,4 o 5 dígitos. El código de servicio CSC2 suele ser 000.

Para ver todos los parámetros disponibles, consulte el AmexCardSecurityCodeVersionapartado 2 en la guía de referencia de la API.

Crear clave

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY,KeyModesO
 --tags='[{"Key":"KEY_PURPOSE","Value":"CSC2"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La respuesta refleja los parámetros de la solicitud, incluyendo un ARN para las llamadas posteriores y un valor de verificación clave (KCV).

```
{
            "Key": {
```

```
                "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
erlm445qvunmvoda",
                "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": "BF1077",
                "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"
            }
        }
```

Tome nota de lo `KeyArn` que representa la clave, por ejemplo arn:aws:payment-cryptography:us-east- 2:111122223333:key/erlm445qvunmvoda. Lo necesitará en el siguiente paso.

Genera un CSC2

En este ejemplo, generaremos a CSC2 con una longitud de 4. El CSC se puede generar con una longitud de 3,4 o 5. Para American Express, PANs debe tener 15 dígitos y empezar por 34 o 37. La fecha de caducidad suele tener el formato YYMM. El código de servicio puede variar; consulte el manual, pero los valores típicos son 000, 201 o 702

Example

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

```
erlm445qvunmvoda --primary-account-number=344131234567848 --generation-attributes
 AmexCardSecurityCodeVersion2='{CardExpiryDate=2412,ServiceCode=000}' --validation-
data-length 4
```

```
{
"KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/erlm445qvunmvoda",
"KeyCheckValue": "BF1077",
"ValidationData": "3982"
}
```

Valide el CSC2

Example

En este ejemplo, validaremos un CSC2.

Si la criptografía de AWS pago puede validarse, se devuelve un http/200. Si el valor no se valida, devolverá una respuesta http/400.

```
$ aws payment-cryptography-data verify-card-validation-data  --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/erlm445qvunmvoda
 --primary-account-number=344131234567848 --verification-attributes
 AmexCardSecurityCodeVersion2='{CardExpiryDate=2412,ServiceCode=000}' --validation-data
 3982
```

```
{
"KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/erlm445qvunmvoda",
"KeyCheckValue": "BF1077"
}
```

iCSC

El iCSC también se conoce como algoritmo CSC estático y se calcula con la versión 2 del CSC. El servicio puede proporcionarlo como un número de 3,4 o 5 dígitos.

Use el código de servicio 999 para calcular el iCSC de una tarjeta de contacto. Utilice el código de servicio 702 para calcular el iCSC de una tarjeta sin contacto.

Para ver todos los parámetros disponibles, consulte el apartado AmexCardSecurityCodeVersion2 de la guía de referencia de la API.

Crear clave

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY,KeyModesO
 --tags='[{"Key":"KEY_PURPOSE","Value":"CSC1"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La respuesta refleja los parámetros de la solicitud, incluyendo un ARN para las llamadas posteriores y un valor de verificación clave (KCV).

```
{
        "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-
east-1:111122223333:key/7vrybrbvjcvwtunv",
        "KeyAttributes": {
            "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"
            "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
            },
        },
        "KeyCheckValue": "7121C7",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "Enabled": true,
        "Exportable": true,
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "CreateTimestamp": "2025-01-29T09:19:21.209000-05:00",
        "UsageStartTimestamp": "2025-01-29T09:19:21.192000-05:00"
        }
    }
```

Tome nota de lo `KeyArn` que representa la clave, por ejemplo arn:aws:payment-cryptography:us-east- 1:111122223333:key/7vrybrbvjcvwtunv. Lo necesitará en el siguiente paso.

Genera un iCSC

En este ejemplo, generaremos un iCSC con una longitud de 4 para una tarjeta sin contacto utilizando el código de servicio 702. El CSC se puede generar con una longitud de 3,4 o 5. Para American Express, PANs debe tener 15 dígitos y empezar por 34 o 37.

Example

```
$ aws payment-cryptography-data generate-card-validation-data --key-identifier
 arn:aws:payment-cryptography:us-east-1:111122223333:key/7vrybrbvjcvwtunv
 --primary-account-number=344131234567848 --generation-attributes
 AmexCardSecurityCodeVersion2='{CardExpiryDate=1224,ServiceCode=702}' --validation-
data-length 4
```

```
{
    "KeyArn": arn:aws:payment-cryptography:us-east-1:111122223333:key/7vrybrbvjcvwtunv,
    "KeyCheckValue": 7121C7,
    "ValidationData": "2365"
}
```

Valide el iCSC

Example

En este ejemplo, validaremos un iCSC.

Si la criptografía de AWS pago puede validarse, se devuelve un http/200. Si el valor no se valida, devolverá una respuesta http/400.

```
$ aws payment-cryptography-data verify-card-validation-data  --key-identifier
 arn:aws:payment-cryptography:us-east-1:111122223333:key/7vrybrbvjcvwtunv
 --primary-account-number=344131234567848 --verification-attributes
 AmexCardSecurityCodeVersion2='{CardExpiryDate=1224,ServiceCode=702}' --validation-data
 2365
```

```
{
    "KeyArn": arn:aws:payment-cryptography:us-east-1:111122223333:key/7vrybrbvjcvwtunv,
    "KeyCheckValue": 7121C7
}
```

3D AEVV

El AEVV (valor de verificación de la cuenta 3-D Secure) de 3DS se utiliza para la autenticación 3-D Secure de American Express. Utiliza el mismo algoritmo CSC2 pero con parámetros de entrada diferentes. El campo de fecha de caducidad debe rellenarse con un número impredecible (aleatorio) y el código de servicio consta del código de resultados de la autenticación AEVV (1 dígito) más el código de autenticación de segundo factor (2 dígitos). La longitud de salida debe ser de 3 dígitos.

Para ver todos los parámetros disponibles, consulte el AmexCardSecurityCodeVersionapartado 2 de la guía de referencia de la API.

Crear clave

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY,KeyModesO
 --tags='[{"Key":"KEY_PURPOSE","Value":"3DS_AEVV"},
{"Key":"CARD_BIN","Value":"12345678"}]'
```

La respuesta refleja los parámetros de la solicitud, incluyendo un ARN para las llamadas posteriores y un valor de verificación clave (KCV).

```
{
        "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kw8djn5qxvfh3ztm",
        "KeyAttributes": {
            "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"
            "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
            },
        },
        "KeyCheckValue": "8F3A21",
```

```
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "Enabled": true,
        "Exportable": true,
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "CreateTimestamp": "2025-02-02T10:30:15.209000-05:00",
        "UsageStartTimestamp": "2025-02-02T10:30:15.192000-05:00"
        }
    }
```

Tome nota de lo `KeyArn` que representa la clave, por ejemplo arn:aws:payment-cryptography:us-east- 2:111122223333:key/kw8djn5qxvfh3ztm. Lo necesitará en el siguiente paso.

Genera un AEVV para 3DS

En este ejemplo, generaremos un AEVV de 3DS con una longitud de 3. El campo de fecha de caducidad contiene un número impredecible (aleatorio) (por ejemplo, 1234) y el código de servicio consta del código de resultados de autenticación AEVV (1 dígito) más el código de autenticación de segundo factor (2 dígitos), por ejemplo, 543, donde 5 es el código de resultados de autenticación y 43 es el código de autenticación de segundo factor. En el caso de American Express, PANs debe tener 15 dígitos y empezar por 34 o 37.

Example

```
$ aws payment-cryptography-data generate-card-validation-data --key-
identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
kw8djn5qxvfh3ztm --primary-account-number=344131234567848 --generation-attributes
 AmexCardSecurityCodeVersion2='{CardExpiryDate=1234,ServiceCode=543}' --validation-
data-length 3
```

```
{
    "KeyArn": arn:aws:payment-cryptography:us-east-2:111122223333:key/kw8djn5qxvfh3ztm,
    "KeyCheckValue": 8F3A21,
    "ValidationData": "921"
}
```

Valide el AEVV de la 3DS

Example

En este ejemplo, validaremos un AEVV de la 3DS.

Si la criptografía de AWS pago puede validarse, se devuelve un http/200. Si el valor no se valida, devolverá una respuesta http/400.

```
$ aws payment-cryptography-data verify-card-validation-data  --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/kw8djn5qxvfh3ztm
 --primary-account-number=344131234567848 --verification-attributes
 AmexCardSecurityCodeVersion2='{CardExpiryDate=1234,ServiceCode=543}' --validation-data
 921
```

```
{
    "KeyArn": arn:aws:payment-cryptography:us-east-2:111122223333:key/kw8djn5qxvfh3ztm,
    "KeyCheckValue": 8F3A21
}
```

Funciones específicas de JCB

Temas

- ARQC - CVN04
- ARQC - CVN01

ARQC - CVN04

El CVN04 de JCB utiliza el método CSK de derivación de claves. ??? Consulte la documentación del esquema para obtener más información sobre cómo crear el campo de datos de la transacción.

ARQC - CVN01

El CVN01 es un método antiguo de JCB para las transacciones EMV, que utiliza la derivación por clave de tarjeta en lugar de la derivación por sesión (por transacción) y también utiliza una carga útil diferente. Visa también usa este mensaje, por lo que el nombre del elemento tiene ese nombre, aunque también se usa para JCB. Para obtener información sobre el contenido de la carga útil, póngase en contacto con la documentación del esquema.

Cree la clave

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS,KeyClass=SYMMETRIC_KEY,KeyMod
 --tags='[{"Key":"KEY_PURPOSE","Value":"CVN10"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La respuesta refleja los parámetros de la solicitud, incluyendo un ARN para las llamadas posteriores y un valor de verificación clave (KCV).


```
{
                        "Key": {
                            "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/pw3s6nl62t5ushfk",
                            "KeyAttributes": {
                                "KeyUsage": "TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS",
                                "KeyClass": "SYMMETRIC_KEY",
                                "KeyAlgorithm": "TDES_2KEY",
                                "KeyModesOfUse": {
                                    "Encrypt": false,
                                    "Decrypt": false,
                                    "Wrap": false,
                                    "Unwrap": false,
                                    "Generate": false,
                                    "Sign": false,
                                    "Verify": false,
                                    "DeriveKey": true,
                                    "NoRestrictions": false
                                }
                            },
                            "KeyCheckValue": "08D7B4",
                            "KeyCheckValueAlgorithm": "ANSI_X9_24",
                            "Enabled": true,
                            "Exportable": true,
                            "KeyState": "CREATE_COMPLETE",
                            "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
                            "CreateTimestamp": "2024-03-07T06:41:46.648000-07:00",
                            "UsageStartTimestamp": "2024-03-07T06:41:46.626000-07:00"
                        }
                    }
```


Tome nota de lo `KeyArn` que representa la clave, por ejemplo arn:aws:payment-cryptography:us-east- 2:111122223333:. key/pw3s6nl62t5ushfk Lo necesitará en el siguiente paso.

Valide el ARQC

Example

En este ejemplo, validaremos un ARQC generado con el CVN01 de JCB. Utiliza las mismas opciones que el método Visa, de ahí el nombre del parámetro.

Si la criptografía de AWS pagos puede validar el ARQC, se devuelve un http/200 . Si el arqc no está validado, devolverá una respuesta. 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
 00000000170000000000000008400080008000084016051700000000093800000B03011203000000 \
             --session-key-derivation-attributes='{"Visa":{"PanSequenceNumber":"01" \
             ,"PrimaryAccountNumber":"9137631040001422"}}'
```

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

UnionPay funciones específicas

Los siguientes ejemplos se refieren a la modalidad UnionPay internacional (UPI), también denominada en ocasiones China UnionPay (CUP).

Temas

- Validación ARQC
- CVN2 (número de verificación de la tarjeta)
- AVN (número de verificación de autenticación)

Validación ARQC

UnionPay La validación ARQC utiliza el modo de derivación de claves EMV_OPTION_A. La clave de sesión se deriva del número de secuencia PAN y del contador de PrimaryAccountNumber transacciones de la aplicación mediante un esquema. UnionPay-specific Para obtener información sobre el formato de los datos de las transacciones y el orden de los campos, consulte la documentación del esquema.

Valide el ARQC

Example Valide un ARQC mediante la derivación de claves UnionPay

En este ejemplo se muestra cómo validar un ARQC generado mediante la derivación de claves. UnionPay

Si la criptografía de AWS pagos valida el ARQC, devuelve una respuesta HTTP 200. Si el ARQC no se valida, AWS Payment Cryptography devuelve una respuesta HTTP 400 con un código de motivo.

```
$ aws payment-cryptography-data verify-auth-request-cryptogram --auth-request-
cryptogram A12345B1A12345B1 \
  --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6nl62t5ushfk \
  --major-key-derivation-mode EMV_OPTION_A \
  --transaction-data 123456789ABCDEF80000000000000000 \
  --session-key-derivation-
attributes='UnionPay={PrimaryAccountNumber=4123412341234123,PanSequenceNumber=01,ApplicationTra
```

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

Si el ARQC no se puede validar, AWS Payment Cryptography devuelve una respuesta de error:

```
An error occurred (VerificationFailedException) when calling the
 VerifyAuthRequestCryptogram operation: Auth request cryptogram verification failed.

Reason: INVALID_AUTH_REQUEST_CRYPTOGRAM
```

CVN2 (número de verificación de la tarjeta)

CVN2 es el equivalente de CVV2. UnionPay Es un número de verificación de la tarjeta que se utiliza para transacciones en línea y en las que no se realiza la tarjeta. Criptográficamente, CVN2 usa el mismo algoritmo que CVV2 (). CardVerificationValue2 Las entradas son el número de cuenta principal (PAN) y la fecha de caducidad de la tarjeta (4 dígitos en formato MMYY).

Crea la clave

Para crear una clave CVN2, utilice el `create-key` comando con el `TR31_C0_CARD_VERIFICATION_KEY` uso y `TDES_2KEY` el algoritmo.

```
$ aws payment-cryptography create-key --exportable \
  --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY,KeyModesO
 \
  --tags='[{"Key":"KEY_PURPOSE","Value":"CVN2"},{"Key":"CARD_BIN","Value":"47613499"}]'
```

La respuesta repite los parámetros de la solicitud, incluido un ARN para las llamadas posteriores y un valor de verificación de clave (KCV).

```
{
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/7f7g4spf3xcklhzu",
        "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": "AEA5CD",
        "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"
    }
```

```
}
```

Toma nota del `KeyArn` valor, por ejemplo arn:aws:payment-cryptography:us-east- 2:111122223333:. key/7f7g4spf3xcklhzu Lo necesitará en los siguientes pasos.

Genere un CVN2

Example Genere un CVN2 para un PAN determinado

Este ejemplo genera un CVN2 para un PAN determinado con entradas de `PAN` y la fecha de caducidad de la tarjeta.

Para ver todos los parámetros disponibles, consulte la guía CardVerificationValue2 de referencia de la API.

```
$ aws payment-cryptography-data generate-card-validation-data \
  --key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/7f7g4spf3xcklhzu \
  --primary-account-number=4761349999991234 \
  --generation-attributes CardVerificationValue2='{CardExpiryDate=1226}'
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/7f7g4spf3xcklhzu",
    "KeyCheckValue": "AEA5CD",
    "ValidationData": "321"
}
```

Valide un CVN2

Example Valide un CVN2 para un PAN determinado

Este ejemplo valida un CVN2 para un PAN determinado con las entradas de`PAN`, la fecha de caducidad de la tarjeta y el CVN2 proporcionado durante la transacción.

Para ver todos los parámetros disponibles, consulta la guía de referencia de la CardVerificationValue2 API.

> **Note**
>
> CVN2 es un valor introducido por el usuario. Los errores de validación periódicos no indican necesariamente un problema.

```
$ aws payment-cryptography-data verify-card-validation-data \
  --key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/7f7g4spf3xcklhzu \
  --primary-account-number=4761349999991234 \
  --verification-attributes CardVerificationValue2='{CardExpiryDate=1226}' \
  --validation-data 321
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/7f7g4spf3xcklhzu",
    "KeyCheckValue": "AEA5CD"
}
```

AVN (número de verificación de autenticación)

El AVN (número de verificación de autenticación) es el criptograma de la UnionPay 3DS. Utiliza el algoritmo CVV (CardVerificationValue1) con campos de entrada reutilizados, como los que se muestran a continuación, pero consulte la documentación del esquema para obtener más información.

- PAN: número de cuenta principal (16-19 dígitos, del aReq)
- CardExpiryDate— Los cuatro dígitos menos significativos de un valor derivado del ID de transacción de DS (dsTransID)
- ServiceCode— 3 dígitos: resultado de la transacción de autenticación (1 dígito) + bit de llenado '0' (1 dígito) +indicador del generador de AV (1 dígito)

Crea la clave

Para crear una clave AVN, utilice el `create-key` comando con el `TR31_C0_CARD_VERIFICATION_KEY` uso y el `TDES_2KEY` algoritmo.

```
$ aws payment-cryptography create-key --exportable \
```

```
  --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY,KeyModesO
 \
  --tags='[{"Key":"KEY_PURPOSE","Value":"AVN"},{"Key":"CARD_BIN","Value":"47613499"}]'
```

La respuesta repite los parámetros de la solicitud, incluido un ARN para las llamadas posteriores y un valor de verificación de clave (KCV).

```
{
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
dnaeyrjgdjjtw6dk",
        "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": "F3FB13",
        "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"
    }
}
```

Toma nota del `KeyArn` valor, por ejemplo arn:aws:payment-cryptography:us-east- 2:111122223333:. key/dnaeyrjgdjjtw6dk Lo necesitará en los siguientes pasos.

Genera un AVN

Example Genera un AVN para un PAN determinado

Este ejemplo genera un AVN para un PAN determinado. El `CardExpiryDate` campo contiene los cuatro dígitos menos significativos derivados del dsTransID. El `ServiceCode` campo contiene el resultado de la autenticación, el bit de llenado y el indicador del generador.

Para ver todos los parámetros disponibles, consulte CardVerificationValue1 la guía de referencia de la API.

```
$ aws payment-cryptography-data generate-card-validation-data \
  --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
dnaeyrjgdjjtw6dk \
  --primary-account-number=4761349999991234 \
  --generation-attributes
 CardVerificationValue1='{CardExpiryDate=9431,ServiceCode=201}'
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
dnaeyrjgdjjtw6dk",
    "KeyCheckValue": "F3FB13",
    "ValidationData": "491"
}
```

Valida un AVN

Example Valide un AVN para un PAN determinado

En este ejemplo, se valida un AVN para un PAN determinado utilizando los mismos campos de entrada que lo generaron, junto con el valor del AVN que se debe verificar.

Para ver todos los parámetros disponibles, consulta la guía de CardVerificationValue1 referencia de la API.

```
$ aws payment-cryptography-data verify-card-validation-data \
  --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
dnaeyrjgdjjtw6dk \
  --primary-account-number=4761349999991234 \
  --verification-attributes
 CardVerificationValue1='{CardExpiryDate=9431,ServiceCode=201}' \
```

```
  --validation-data 491
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
dnaeyrjgdjjtw6dk",
    "KeyCheckValue": "F3FB13"
}
```

Facilitadores de adquisiciones y pagos

Los adquirentes PSPs y los facilitadores de pagos suelen tener un conjunto de requisitos criptográficos diferentes a los de los emisores. Los casos de uso comunes incluyen:

Descifrado de datos

Los datos (especialmente los datos panorámicos) pueden estar cifrados por un terminal de pago y deben ser descifrados por el servidor. Decrypt Data y Encrypt Data admiten una variedad de métodos, incluidas las técnicas de derivación TDES, AES y DUKPT. El propio servicio AWS de criptografía de pagos también es compatible con PCI P2PE y está registrado como un componente de descifrado PCI P2PE.

TranslatePin

Para garantizar la conformidad con el PIN PCI, los sistemas de adquisición no deberán tener los pines de los titulares de las tarjetas en blanco después de haberlos introducido en un dispositivo seguro. Por lo tanto, para pasar el pin del terminal a un sistema posterior (como una red de pago o un emisor), es necesario volver a cifrarlo con una clave diferente a la que utilizó el terminal de pago. Translate Pin lo logra al convertir un pin cifrado de una clave a otra de forma segura con el servicio.bbb. Con este comando, los pines se pueden convertir entre varios esquemas, como la derivación TDES, AES y DUKPT, y entre formatos de bloques de pines, como el ISO-0, el ISO-3 y el ISO-4.

VerifyMac

Los datos de un terminal de pago pueden estar machacados para garantizar que no se hayan modificado durante el tránsito. Verifica el Mac y GenerateMac admite una variedad de técnicas que utilizan claves simétricas, incluidas las técnicas de derivación TDES, AES y DUKPT para su uso con el algoritmo 1 de la ISO-9797-1, el algoritmo 3 de la ISO-9797-1 (MAC minorista) y las técnicas de CMAC.

Temas adicionales

- Uso de claves dinámicas

Uso de claves dinámicas

Las claves dinámicas permiten utilizar claves de un solo uso o de uso limitado para operaciones criptográficas como`EncryptData`. Este flujo se puede utilizar cuando el material clave cambia con frecuencia (por ejemplo, en todas las transacciones con tarjeta) y se desea evitar la importación del material clave al servicio. Las claves de corta duración se pueden utilizar como parte de SoftPOS/MPOC u otras soluciones.

> **Note**
>
> Esto se puede utilizar en lugar del flujo típico de criptografía de AWS pagos, en el que las claves criptográficas se crean o importan al servicio y las claves se especifican mediante un alias o un arn de clave.

Las siguientes operaciones admiten las claves dinámicas:

- EncryptData
- DecryptData
- ReEncryptData
- TranslatePin

Descifrado de datos

El siguiente ejemplo muestra el uso de claves dinámicas junto con el comando decrypt. En este caso, el identificador de clave es la clave de empaquetado (KEK) que protege la clave de descifrado (que se proporciona en el parámetro wrapped-key en formato TR-31). La clave empaquetada será el objetivo principal de D0 si se utiliza con el comando de descifrado junto con un modo de uso de B o D.

Example

```
$ aws payment-cryptography-data decrypt-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/ov6icy4ryas4zcza
```

```
 --cipher-text 12341234123412341234123412341234123A --decryption-attributes
 'Symmetric={Mode=CBC,InitializationVector=1234123412341234}'   --wrapped-key
 WrappedKeyMaterial={"Tr31KeyBlock"="D0112D0TN00E0000B05A6E82D7FC68B95C84306634B0000DA4701BE9BC
```

```
{
   "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ov6icy4ryas4zcza",
   "KeyCheckValue": "0A3674",
   "PlainText": "2E138A746A0032023BEF5B85BA5060BA"
}
```

Traducir un pin

En el siguiente ejemplo, se muestra el uso de teclas dinámicas junto con el comando translate pin para convertir una clave dinámica en una clave de trabajo de adquisición semiestática (AWK). En este caso, el identificador de clave entrante es la clave de empaquetado (KEK) que protege la clave de cifrado de pines dinámicos (PEK) que se proporciona en el formato TR-31. La clave empaquetada debe ser el objetivo principal, `P0` junto con un modo de uso de B o D. El identificador de clave saliente es una clave del tipo `TR31_P0_PIN_ENCRYPTION_KEY` y un modo de uso de Encrypt=True, wrap=TRUE

Example

```
$ aws payment-cryptography-data translate-pin-data --encrypted-pin-block
 "C7005A4C0FA23E02" --incoming-translation-
attributes=IsoFormat0='{PrimaryAccountNumber=171234567890123}'
 --incoming-key-identifier alias/PARTNER1_KEK  --outgoing-key-
identifier alias/ACQUIRER_AWK_PEK --outgoing-translation-attributes
 IsoFormat0="{PrimaryAccountNumber=171234567890123}"  --incoming-wrapped-key
 WrappedKeyMaterial={"Tr31KeyBlock"="D0112P0TB00S0000EB5D8E63076313162B04245C8CE351C956EA4A16CC
```

```
{
   "PinBlock": "2E66192BDA390C6F",
   "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ov6icy4ryas4zcza",
   "KeyCheckValue": "0A3674"
}
```

Características específicas de la región para AWS Payment Cryptography

Algunas funciones pueden ser específicas de una región y no usarse de otro modo. Estas funciones se describen con más detalle en esta sección.

AS2805

La norma australiana 2805 (AS2805) es una norma para las transferencias electrónicas de fondos que se utiliza principalmente para las transacciones de pago con tarjeta. Está mantenida por Standards Australia. https://www.standards.org.au/ El estándar consta de 6 libros que cubren numerosos temas, desde el formato de los mensajes hasta los estándares de cifrado.

La parte 6 proporciona orientación sobre la administración de claves, incluida la comunicación de host a host (nodo a nodo) y los requisitos criptográficos pertinentes, mientras que otros aspectos se tratan en otras partes. Toda la criptografía de este estándar se basa actualmente en el TDES.

El AS2805 presenta una serie de diferencias en comparación con otras implementaciones, que se resumen a continuación.

Protección de claves

Se basa en variantes de teclas en lugar de en bloques de teclas, como en TR-31/X9.143. AWS La criptografía de pagos almacena internamente todas las claves como bloques clave, pero permite la importación, la exportación y el cálculo mediante las variantes definidas por la norma AS2805.

Claves unidireccionales

La AS2805 exige el uso de teclas unidireccionales. Si ambos nodos necesitan generar códigos de autenticación de mensajes (MAC), utilizan dos claves.

Bloques de pines

La AS2805 define una técnica de derivación de claves para claves de cifrado PIN únicas por transacción. Esto se puede usar en lugar de DUKPT. El esquema AS2805 se basa en los datos de las transacciones (número de rastreo e importe de la transacción), en comparación con el uso del contador de transacciones por parte de DUKPT.

Validación del intercambio de claves

Define un proceso para validar la KEK antes de empezar a intercambiar claves de trabajo, como las claves PIN. En otros esquemas, las KEK se intercambian con poca frecuencia y se validan mediante el KCV.

El AS2805 utiliza el concepto de variantes de clave en lugar de bloques de claves para garantizar que las claves solo se utilicen para el propósito previsto (y único). A continuación se muestra cómo la criptografía de AWS pagos mapea las variantes y los bloques de teclas al importar, exportar o realizar otras funciones criptográficas con claves.

Tipo de clave AS2805	AWS Tipo de clave de criptografía de pagos
TERMINAL_MAJOR_KEY_VARIANT_00	TR31_K0_KEY_ENCRYPTION_KEY
VARIANT_CLAVE DE CIFRADO PIN_28	TR31_P0_PIN_ENCRYPTION_KEY
MESSAGE_AUTHENTICATION_KEY_ VARIANT_24	TR31_M0_ISO_16609_MAC_KEY
VARIANT_DE_CLAVE_CIFRADO DE DATOS_22	TR31_D0_SYMMETRIC_DATA_ENCR YPTION_KEY
VARIANT_MASK_82, VARIANT_MASK_82C0	Opciones disponibles como parte del proceso de validación de KEK. Estos tipos de claves son efímeros y el servicio no los almacena.

Dados dos nodos, node1 y node2, los siguientes ejemplos son desde la perspectiva del nodo1. AWS La criptografía de pagos admite las API de ambos lados del proceso.

Temas

- Intercambio de clave inicial (KEK)
- Validación del KEK
- Creación y transmisión de claves de trabajo
- Exportación de claves de trabajo
- Traducción de pines

- Generación y validación de Mac

Intercambio de clave inicial (KEK)

En el AS2805, cada lado tiene su propia KEK. Las KEK se refieren a la clave de la parte emisora que se utilizará siempre que la parte emisora necesite protect/wrap claves y enviarlas al nodo2. KEK (r) es la clave creada por el lado opuesto (node2).

> **Note**
>
> Estos términos son relativos: un lado crea una clave (el lado emisor) y el otro la recibe. Por lo tanto, dada la KEY1, en el nodo1 se la denomina KEK (s) y en el nodo2 como KEK (r).

Las KEK del AS2805 son siempre del tipo de clave = TR31_K0_KEY_ENCRYPTION_KEY, ya que se utilizan para proteger criptogramas y no bloques de claves. Esto se asigna a TERMINAL_MAJOR_KEY_VARIANT_00 tal como se define en el AS2805 6.1

Pasos:

1. Crea una clave

   Crea una clave con la CreateKey API. Creará una clave de tipo TR31_K0_KEY_ENCRYPTION_KEY

2. Determine el método para intercambiar claves con el node2

   Determine cómo intercambiar KEK con la contraparte. Para el AS2805, el método más común e interoperable es RSA Wrap.

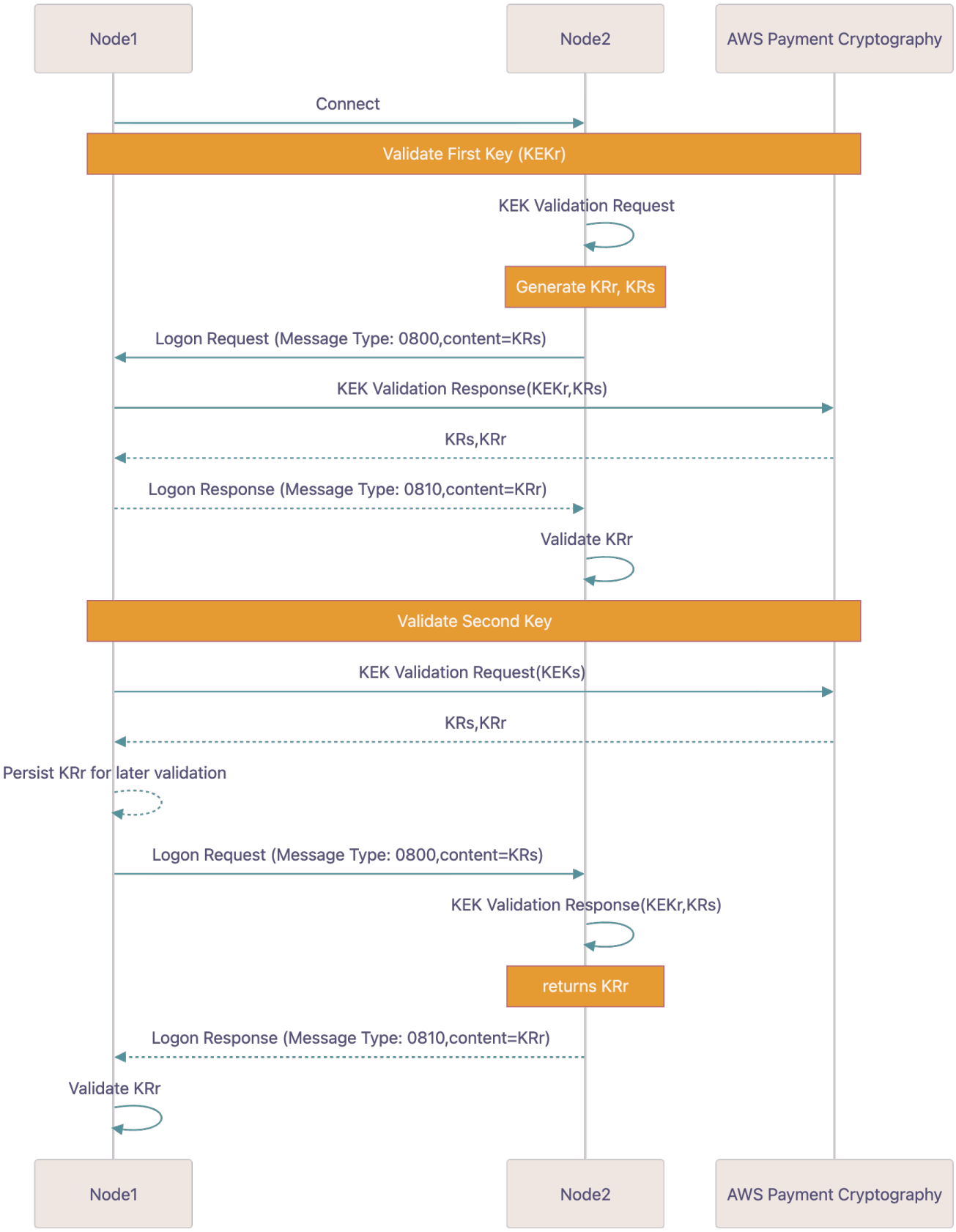
3. Exportar KEK

   Según su selección anterior, recibirá un certificado de clave pública del node2. Ejecutarás la exportación con ese certificado para proteger la clave (o derivarás una clave si utilizas el ECDH).

4. Importar KEKr

   Según su selección anterior, enviará un certificado de clave pública al node2. Ejecutarás la importación con ese certificado para cargar el KEKr del nodo 2 en el servicio.

Validación del KEK



Cuando su servicio (nodo1) se conecte al nodo2, cada una de las partes se asegurará de utilizar la misma KEK para las operaciones posteriores mediante un proceso denominado validación de KEK.

1. Pasos para validar la primera clave

1.1 Reciba los KRs

Node2 generará un KRs y se lo enviará como parte del proceso de inicio de sesión. Pueden usar la criptografía AWS de pagos para generar este valor u otra solución.

1.2 Generar una respuesta de validación de KEK

Su nodo generará una respuesta de validación de la KEK con entradas como la KEK (r) y los KRs proporcionados en el paso 1.

Example

```
cat >> generate-kek-validation-response.json
{
    "KekValidationType": {
        "KekValidationResponse": {
            "RandomKeySend": "9217DC67B8763BABCFDF3DADFCD0F84A"
        }
    },
    "RandomKeySendVariantMask": "VARIANT_MASK_82",
    "KeyIdentifier": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ov6icy4ryas4zcza"
}
```

```
$ aws payment-cryptography-data generate-as2805-kek-validation --cli-input-json
 file://generate-kek-validation-response.json
```

```
{
 "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ov6icy4ryas4zcza",
 "KeyCheckValue": "0A3674",
 "RandomKeyReceive": "A4B7E249C40C98178C1B856DB7FB76EB",
 "RandomKeySend": "9217DC67B8763BABCFDF3DADFCD0F84A"
}
```

1.3 La rentabilidad calculada (kRR)

Devuelve el kRR calculado al nodo2. Ese nodo lo comparará con el valor calculado en el paso 1.

2. Pasos para validar la segunda clave

2.1 Genere los KRr y los KRs

Tu nodo generará un valor aleatorio y una copia invertida (invertida) de este valor mediante AWS criptografía de pagos. El servicio generará ambos valores envueltos en las KEK. Estos se conocen como KR (s) y KR (r).

Example

```
cat >> generate-kek-validation-request.json
{
    "KekValidationType": {
        "KekValidationRequest": {
            "DeriveKeyAlgorithm": "TDES_2KEY"
        }
    },
"RandomKeySendVariantMask": "VARIANT_MASK_82",
    "KeyIdentifier": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
rhfm6tenpxapkmrv"
}
```

```
$ aws payment-cryptography-data generate-as2805-kek-validation --cli-input-json
 file://generate-kek-validation-request.json
```

```
{
 "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
rhfm6tenpxapkmrv",
 "KeyCheckValue": "DC1081",
 "RandomKeyReceive": "A4B7E249C40C98178C1B856DB7FB76EB",
 "RandomKeySend": "9217DC67B8763BABCFDF3DADFCD0F84A"
}
```

2.2 Enviar los KRs al nodo2

Envía los KRs al nodo2. Conserve el kRR para su posterior validación.

2.3 El nodo 2 genera la respuesta de validación de la KEK

Node2 usa el KEKr y los KRs, genera el KRr y lo envía de vuelta a su servicio.

2.4 Valida la respuesta

Compare el kRR del paso 1 con el valor devuelto en el paso 3. Si coinciden, continúe.

Creación y transmisión de claves de trabajo

Las claves de trabajo típicas utilizadas en el AS2805 incluyen dos conjuntos de claves:

Claves entre nodos, como la clave PIN de zona (ZPK), la clave de cifrado de zona (ZEK) y la clave de autenticación de zona (ZAK).

Claves entre terminales y nodos, como: la clave principal del terminal (TMK) y la clave PIN del terminal (TPK) si no se utiliza DUKPT.

> **Note**
>
> Recomendamos minimizar las claves clave por terminal y aprovechar técnicas como TR-34 el DUKPT siempre que sea posible, que utilizan un número menor de claves.

Example

En este ejemplo, hemos usado etiquetas opcionales para rastrear el propósito y el uso de esta clave. Las etiquetas no se utilizan como parte de la función criptográfica del sistema, pero se pueden utilizar para la categorización y el seguimiento financiero, así como para aplicar las políticas de IAM.

```
cat >> create-zone-pin-key.json
{
    "KeyAttributes": {
        "KeyUsage": "TR31_P0_PIN_ENCRYPTION_KEY",
        "KeyClass": "SYMMETRIC_KEY",
        "KeyAlgorithm": "TDES_2KEY",
        "KeyModesOfUse": {
            "Encrypt": true,
            "Decrypt": true,
            "Wrap": true,
            "Unwrap": true,
            "Generate": false,
            "Sign": false,
            "Verify": false,
            "DeriveKey": false,
            "NoRestrictions": false
        }
    },
    "KeyCheckValueAlgorithm": "ANSI_X9_24",
```

```
    "Exportable": true,
    "Enabled": true,
    "Tags": [
        {
            "Key": "AS2805_KEYTYPE",
            "Value": "ZONE_PIN_KEY_VARIANT28"
        }
    ]
}
```

```
$ aws payment-cryptography create-key --cli-input-json file://create-zone-pin-key.json
 --region ap-southeast-2
```

```
{
 "Key": {
 "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/alsuwfxug3pgy6xh",
 "KeyAttributes": {
 "KeyUsage": "TR31_P0_PIN_ENCRYPTION_KEY",
 "KeyClass": "SYMMETRIC_KEY",
 "KeyAlgorithm": "TDES_2KEY",
 "KeyModesOfUse": {
 "Encrypt": true,
 "Decrypt": true,
 "Wrap": true,
 "Unwrap": true,
 "Generate": false,
 "Sign": false,
 "Verify": false,
 "DeriveKey": false,
 "NoRestrictions": false
 }
 },
 "KeyCheckValue": "9A325B",
 "KeyCheckValueAlgorithm": "ANSI_X9_24",
 "Enabled": true,
 "Exportable": true,
 "KeyState": "CREATE_COMPLETE",
 "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
 "CreateTimestamp": "2025-12-17T09:05:27.586000-08:00",
 "UsageStartTimestamp": "2025-12-17T09:05:27.570000-08:00"
 }
}
```

Exportación de claves de trabajo

Para mantener la compatibilidad con terceros, la criptografía de AWS pagos admite las técnicas de empaquetado simétrico de claves AS2805, que utilizan variantes de clave en lugar de bloques de teclas, como los bloques de teclas. TR-31 Si las partes comparten varias claves, cada una debe exportarse de forma individual. Si los datos se envían de forma bidireccional, es posible que haya dos claves entre partes del mismo tipo, como ZAK (s) y ZAK (r), que cada parte utilice para generar códigos de autenticación de mensajes.

Los parámetros adicionales para importar y exportar en estos formatos se especifican en los comandos.

```
cat >> export-zone-pin-key.json
{
    "ExportKeyIdentifier": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
alsuwfxug3pgy6xh",
    "KeyMaterial": {
        "As2805KeyCryptogram": {
            "WrappingKeyIdentifier": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/rhfm6tenpxapkmrv",
            "As2805KeyVariant: "PIN_ENCRYPTION_KEY_VARIANT_28"
        }
    }
}
```

```
$ aws payment-cryptography export-key --cli-input-json file://export-zone-pin-key.json
 --region ap-southeast-2
```

```
{
    "WrappedKey": {
        "KeyCheckValue": "DC1081",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyMaterial": "HDC10AEF038E695DDD72AF08DC1BB422D",
        "WrappedKeyMaterialFormat": "KEY_CRYPTOGRAM",
        "WrappingKeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
rhfm6tenpxapkmrv"
    }
}
```

Traducción de pines

El AS2805 describe un modo de derivación de claves específico de la sesión en la sección 6.4. Tiene un propósito similar al del DUKPT y se puede usar cualquiera de los dos algoritmos, ya que el DUKPT se trata en la sección 6.7. En este esquema, una clave PIN de sesión (conocida como KPE) se deriva de la clave PIN del terminal utilizando SystemTraceAuditNumber (STAN) y como datos de derivación. TransactionAmount

El pin de traducción es una función común que puede traducir to/from una variedad de formatos. En este ejemplo, traducimos un PIN de un KPE a una clave de cifrado PIN (PEK), como cuando se envía un PIN a una red de pago.

```
cat >> translate-pin-as2805.json
{
    "EncryptedPinBlock": "B3B34B43BAB5F81A",
    "IncomingKeyIdentifier": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt",
    "IncomingTranslationAttributes": {
        "IsoFormat0": {
            "PrimaryAccountNumber": "9999179999900013"
        }
    },
      "IncomingAs2805Attributes": {
        "SystemTraceAuditNumber": "000348",
        "TransactionAmount": "000000000328"
    },
    "OutgoingKeyIdentifier": "",
    "OutgoingTranslationAttributes": {
        "IsoFormat0": {
            "PrimaryAccountNumber": "9999179999900013"
        }
    }
}
```

```
$ aws payment-cryptography-data translate-pin-data --cli-input-json file://translate-
pin-as2805.json  --region ap-southeast-2
```

```
{
    "WrappedKey": {
        "KeyCheckValue": "DC1081",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
```

```
        "KeyMaterial": "HDC10AEF038E695DDD72AF08DC1BB422D",
        "WrappedKeyMaterialFormat": "KEY_CRYPTOGRAM",
        "WrappingKeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
rhfm6tenpxapkmrv"
    }
}
```

Generación y validación de Mac

Los comandos de generación y verificación de MAC son compatibles con una variedad de MAC, incluidos HMAC, CMAC, EMV MAC, etc. Para el AS2805, existe una variación adicional definida en. AS2805.4.1 Por lo general, en el AS2805, los mensajes entrantes se verifican con este MAC y los mensajes salientes también incluyen un MAC.

```
cat verify-mac.json
{
    "KeyIdentifier": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
qnobl5lghrzunce6",
    "Mac": "86304058",
    "MessageData": "73D8BA54D3852951DAEA41",
    "VerificationAttributes": {
        "Algorithm": "AS2805_4_1"
    }
}
```

```
$ aws payment-cryptography-data verify-mac --cli-input-json file://verify-mac.json --
region ap-southeast-2
```

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

Seguridad en AWS Payment Cryptography

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, se beneficia de una arquitectura de red y centro de datos diseñada para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre usted AWS y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube:AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede usar de forma segura. Third-party los auditores prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWS programas de AWS Para obtener más información sobre los programas de cumplimiento que se aplican a la criptografía de AWS pagos, consulte el programa .
- Seguridad en la nube: su responsabilidad está determinada por el AWS servicio que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Este tema le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida al usar la criptografía AWS de pagos. Le muestra cómo configurar la criptografía AWS de pagos para cumplir sus objetivos de seguridad y cumplimiento. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus recursos de criptografía AWS de pagos.

Temas

- Protección de datos en AWS Payment Cryptography
- Resiliencia en AWS Payment Cryptography
- Seguridad de la infraestructura en AWS Payment Cryptography
- Conectarse a AWS Criptografía de pagos a través de un punto final de VPC
- Usar el cifrado TLS híbrido postcuántico
- Prácticas recomendadas de seguridad para AWS Payment Cryptography

Protección de datos en AWS Payment Cryptography

El modelo de responsabilidad AWS https://aws.amazon.com/compliance/shared-responsibility-model/ se aplica a la protección de datos en la criptografía AWS de pagos. Como se describe en este

modelo, AWS es responsable de proteger la infraestructura global que ejecuta todo el Nube de AWS. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las Preguntas frecuentes sobre la privacidad de los datos y las condiciones legales de la región de . Para obtener más información sobre la protección de datos en Europa, consulte el Centro del Reglamento General de Protección de Datos (RGPD).

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Úselo SSL/TLS para comunicarse con AWS los recursos. Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Configure el registro de actividad de los usuarios y de la API con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail rutas para capturar AWS actividades, consulte Cómo trabajar con CloudTrail rutas en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados que contiene Servicios de AWS.
- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger la información confidencial almacenada en Amazon S3.
- Si necesita módulos criptográficos validados por la FIPS 140-3 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-3.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabajas con criptografía de AWS pagos u otros métodos Servicios de AWS mediante la consola, la API o los SDK. AWS CLI AWS Cualquier dato que introduzca en etiquetas o campos de formato libre utilizados para los nombres se pueden emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor

externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

AWS Payment Cryptography almacena y protege sus claves de cifrado de pagos para que estén altamente disponibles al mismo tiempo que le proporciona un control de acceso sólido y flexible.

Temas

- Rotación del material de claves
- Cifrado de datos
- Cifrado en reposo
- Cifrado en tránsito
- Privacidad del tráfico entre redes

Rotación del material de claves

Por defecto, AWS Payment Cryptography protege el material criptográfico de las claves de pago gestionadas por el servicio. Además, AWS Payment Cryptography ofrece opciones para importar material de claves creado fuera del servicio. Para obtener más detalles sobre las claves de pago y el material de claves, consulte los detalles de criptografía de AWS Payment Cryptography.

Cifrado de datos

Los datos de AWS Payment Cryptography consisten en claves de AWS Payment Cryptography, el material de claves criptográficas que representan y sus atributos de uso. El material de claves solo existe en texto sin formato dentro de los módulos de seguridad de hardware (HSM) de AWS Payment Cryptography y solo cuando estén en uso. De lo contrario, el material de la clave y los atributos se cifran y se almacenan en almacenamiento persistente duradero.

El material de claves que AWS Payment Cryptography genera o carga para las claves de pago nunca abandona el límite de los HSM de AWS Payment Cryptography sin cifrar. Puede ser exportado cifrado por las operaciones API de AWS Payment Cryptography.

Cifrado en reposo

AWS Payment Cryptography genera material clave para las claves de pago en los HSM PCI PTS. HSM-listed Cuando no se utiliza, el material de claves se cifra mediante una clave de HSM y

se escribe en un almacenamiento duradero y persistente. El material de las claves de Payment Cryptography y las claves de cifrado que protegen el material de claves nunca dejan los HSM en forma de texto sin formato.

El cifrado y la administración del material de claves para las claves de Payment Cryptography está completamente a cargo del servicio.

Para obtener más información, consulte el documento técnico Detalles criptográficos de AWS Key Management Service.

Cifrado en tránsito

El material clave que la criptografía de AWS pagos genera o carga para las claves de pago nunca se exporta ni se transmite en las operaciones de la API de criptografía de AWS pagos en texto plano. AWS La criptografía de pagos utiliza identificadores clave para representar las claves de las operaciones de la API.

Sin embargo, algunas operaciones de la API exportan claves cifradas mediante una clave de intercambio de claves previamente compartida o asimétrica. Además, los clientes pueden usar las operaciones de la API para importar material de claves encriptadas para las claves de pago.

Todas las llamadas a la API de criptografía de AWS pagos deben firmarse y transmitirse mediante Transport Layer Security (TLS). AWS La criptografía de pagos requiere versiones de TLS y conjuntos de cifrado definidos por PCI como «criptografía sólida». Todos los terminales de servicio admiten el TLS 1.2—1.3 y el TLS poscuántico híbrido.

Para obtener más información, consulte el documento técnico Detalles criptográficos de AWS Key Management Service.

Privacidad del tráfico entre redes

AWS La criptografía de pagos admite una consola de administración de AWS y un conjunto de operaciones de API que le permiten crear y administrar claves de pago y utilizarlas en operaciones criptográficas.

AWS La criptografía de pagos admite dos opciones de conectividad de red desde su red privada a AWS.

- Una conexión de una conexión de VPN IPSec a través de Internet.

- AWS Direct Connect vincula su red interna con una ubicación de AWS Direct Connect a través de cable estándar Ethernet de fibra óptica.

Todas las llamadas de la API de Payment Cryptography deben firmarse y transmitirse mediante seguridad de la capa de transporte (TLS). Las llamadas también requieren un paquete de cifrado moderno que admita el secreto perfecto en el futuro. El tráfico a los módulos de seguridad de hardware (HSM) que almacenan material de claves para las claves de pago solo se permite desde hosts de la API de AWS Payment Cryptography a través de la red interna de AWS.

Para conectarse directamente a AWS Payment Cryptography desde su nube privada virtual (VPC) sin enviar tráfico a través de la Internet pública, utilice puntos de enlace de VPC con tecnología de AWS. PrivateLink Para obtener más información, consulte Conexión a AWS Payment Cryptography a través de un punto de conexión de VPC.

AWS Payment Cryptography admite también una opción de intercambio híbrido postcuántico de claves para el protocolo de cifrado de red de seguridad de la capa de transporte (TLS). Puede utilizar esta opción de TLS cuando se conecte a los puntos de conexión de la API de AWS Payment Cryptography.

Resiliencia en AWS Payment Cryptography

AWS La infraestructura global se basa en AWS regiones y zonas de disponibilidad. Las regiones proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja demora. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte Infraestructura AWS global.

Aislamiento regional

La criptografía de pagos de AWS es un servicio regional que está disponible en varias regiones.

El diseño regionalmente aislado de AWS Payment Cryptography garantiza que un problema de disponibilidad en una región de AWS no puede afectar al funcionamiento de AWS Payment

Cryptography en ninguna otra región. AWS Payment Cryptography está diseñado para garantizar un tiempo de inactividad planificado cero, con todas las actualizaciones de software y operaciones de escalado realizadas de manera imperceptible y sin problemas.

El Acuerdo de nivel de servicio (SLA) de AWS Payment Cryptography incluye un compromiso de servicio del 99,99 % para todas las API de AWS Payment Cryptography. Para cumplir este compromiso, AWS Payment Cryptography garantiza que todos los datos y la información de autorización necesarios para ejecutar una solicitud de la API estén disponibles en todos los hosts regionales que reciben la solicitud.

La infraestructura de AWS Payment Cryptography se replica en al menos tres zonas de disponibilidad (AZ) en cada región. Para garantizar que los errores de varios hosts no afecten al rendimiento, AWS Payment Cryptography está diseñado para atender el tráfico de clientes de cualquiera de las zonas de disponibilidad de una región.

Los cambios realizados en las propiedades o permisos de una clave de pagos se replican en todos los hosts de la región para garantizar que cualquier host de la región pueda procesar de manera correcta la solicitud posterior. Solicitudes de operaciones criptográficas mediante el uso de la clave de pagos se reenvían a una flota de módulos de seguridad de hardware (HSM) de AWS Payment Cryptography, cualquiera de los cuales puede realizar la operación con la clave de pagos.

Multi-tenant diseño

El diseño de varios inquilinos de AWS Payment Cryptography permite cumplir el SLA de disponibilidad y mantener altas tasas de solicitudes, al tiempo que protege la confidencialidad de las claves y los datos.

Se implementan varios mecanismos de cumplimiento de la integridad para garantizar que la clave de pagos especificada para la operación criptográfica sea siempre la que se utiliza.

El material de clave de texto sin formato para las claves de Payment Cryptography está ampliamente protegido. El material de clave se cifra en el HSM tan pronto como se crea y el material de clave cifrado se mueve de inmediato al almacenamiento seguro. La clave cifrada se recupera y se descifra dentro del HSM justo a tiempo para su uso. La clave de texto sin formato permanece en la memoria HSM solo durante el tiempo necesario para completar la operación criptográfica. El material de claves de texto sin formato nunca sale de los HSM; nunca se escribe en un almacenamiento persistente.

Para obtener más información acerca de los mecanismos que AWS Payment Cryptography utiliza para proteger sus claves, consulte Detalles criptográficos de AWS Payment Cryptography.

Seguridad de la infraestructura en AWS Payment Cryptography

Como servicio gestionado, AWS Payment Cryptography está protegido por los procedimientos de seguridad de red AWS global que se describen en el documento técnico Amazon Web Services: descripción general de los procesos de seguridad.

Para acceder a AWS Payment Cryptography través de la red, se utilizan las llamadas a la API AWS publicadas. Los clientes deben ser compatibles con seguridad de la capa de transporte (TLS) 1.2 o con una versión posterior. Los clientes también deben admitir conjuntos de cifrado con un secreto directo perfecto (PFS), como Ephemeral (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Diffie-Hellman La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Aislamiento de hosts físicos

La seguridad de la infraestructura física que AWS Payment Cryptography utiliza está sujeto a los controles que se describen en la sección Seguridad Física y Ambiental de Amazon Web Services: Información general de los procesos de seguridad. Puede encontrar más detalles en los informes de conformidad y los resultados de auditoría de terceros enumerados en la sección anterior.

La criptografía de pagos de AWS es compatible con módulos de seguridad de hardware (HSM) PCI PTS dedicados y listos para usar en el mercado. HSM-listed El material de claves para las claves de AWS Payment Cryptography se almacena solo en memoria volátil en los HSM, y solo mientras la clave de Payment Cryptography está en uso. Los HSM se encuentran en bastidores de acceso controlado dentro de los centros de datos de Amazon que aplican un doble control para cualquier acceso físico. Para obtener información detallada acerca de la operación de los HSM de AWS Payment Cryptography, consulte Detalles criptográficos de AWS Payment Cryptography.

Conectarse a AWS Criptografía de pagos a través de un punto final de VPC

Puedes conectarte directamente a la criptografía AWS de pagos a través de un punto final de interfaz privada en tu nube privada virtual (VPC). Cuando utilizas un punto final de interfaz de VPC, la

comunicación entre tu VPC y AWS Payment Cryptography se lleva a cabo completamente dentro de la red. AWS

AWS La criptografía de pagos es compatible con los puntos de enlace de Amazon Virtual Private Cloud (Amazon VPC) impulsados por. AWS PrivateLink Cada punto de conexión de VPC está representado por una o varias Interfaces de red elásticas (ENI) con direcciones IP privadas en las subredes de la VPC.

El punto final de la VPC de la interfaz conecta su VPC directamente a AWS Payment Cryptography sin necesidad de una pasarela de Internet, un dispositivo NAT, una conexión VPN o una conexión. AWS Direct Connect Las instancias de su VPC no necesitan direcciones IP públicas para comunicarse con la criptografía de pagos. AWS

Regions

AWS La criptografía de pagos admite los puntos de enlace de la VPC y las políticas de puntos finales de la VPC en todos los lugares Regiones de AWS en los que se admite la criptografía de pagos. AWS

Temas

- Consideraciones para AWS Puntos finales de VPC de criptografía de pagos
- Crear un punto final de VPC para AWS Payment Cryptography
- Conectarse a un AWS Punto final de VPC de criptografía de pagos
- Control del acceso a un punto de conexión de VPC
- Utilizar un punto de conexión de VPC en una declaración de política
- Registro de su punto de conexión de VPC

Consideraciones para AWS Puntos finales de VPC de criptografía de pagos

> **Note**
>
> Si bien los terminales de la VPC permiten conectarse al servicio en tan solo una zona de disponibilidad (AZ), recomendamos conectarse a tres zonas de disponibilidad por motivos de alta disponibilidad y redundancia.

Antes de configurar un punto final de la VPC de la interfaz para la criptografía de AWS pagos, consulte el tema sobre las propiedades y limitaciones del punto final de la interfaz de la guía. AWS PrivateLink

AWS La compatibilidad con la criptografía de pagos para un punto final de VPC incluye lo siguiente.

- Puede usar su punto de conexión de VPC para llamar a todas las operaciones del plano de control de criptografía de AWS pagos y a las operaciones del plano de datos de criptografía de AWS pagos desde una VPC.
- Puede crear un punto de enlace de VPC de interfaz que se conecte a un punto de enlace de la región de criptografía de pagos AWS .
- AWS La criptografía de pagos se compone de un plano de control y un plano de datos. Puede elegir configurar uno o ambos subservicios, AWS PrivateLink pero cada uno se configura por separado.
- Puedes usar AWS CloudTrail los registros para auditar el uso de las claves de criptografía de AWS pagos a través del punto de conexión de la VPC. Para obtener más información, consulte Registro de su punto de conexión de VPC.

Crear un punto final de VPC para AWS Payment Cryptography

Puede crear un punto de enlace de VPC para la criptografía de AWS pagos mediante la consola de Amazon VPC o la API de Amazon VPC. Para obtener más información, consulte Creación de un punto de conexión de interfaz en la Guía de AWS PrivateLink .

- Para crear un punto de enlace de VPC para la criptografía de AWS pagos, utilice los siguientes nombres de servicio:

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

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

  Por ejemplo, en la región Oeste de EE. UU. (Oregón) (`us-west-2`), los nombres de los servicios serían:

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

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

Para facilitar el uso del punto de conexión de VPC, puede habilitar un nombre de DNS privado para el punto de conexión de VPC. Si seleccionas la opción Habilitar el nombre DNS, el nombre de host DNS estándar de criptografía de AWS pagos se convierte en el punto final de la VPC. Por ejemplo, `https://controlplane.payment-cryptography.us-west-2.amazonaws.com` se resolvería en un punto de conexión de VPC conectado al nombre del servicio `com.amazonaws.us-west-2.payment-cryptography.controlplane`.

Esta opción facilita el uso del punto de conexión de VPC. Los AWS SDK AWS CLI utilizan el nombre de host DNS estándar de criptografía de AWS pagos de forma predeterminada, por lo que no es necesario especificar la URL del punto de enlace de la VPC en las aplicaciones y los comandos.

Para obtener más información, consulte Acceso a un servicio a través de un punto de conexión de interfaz en la Guía de AWS PrivateLink .

Conectarse a un AWS Punto final de VPC de criptografía de pagos

Puedes conectarte a la criptografía AWS de pagos a través del punto final de la VPC mediante un AWS SDK, el o. AWS CLI Herramientas de AWS para PowerShell Para especificar el punto de conexión de VPC, utilice su nombre de DNS.

Por ejemplo, este comando list-keys utiliza el parámetro `endpoint-url` para especificar el punto de conexión de VPC. Para utilizar un comando de este tipo, sustituya el ID del punto de conexión de VPC del ejemplo por uno de su cuenta.

```
$ aws payment-cryptography list-keys --endpoint-url https://
vpce-1234abcdf5678c90a-09p7654s-us-east-1a.ec2.us-east-1.vpce.amazonaws.com
```

Si ha habilitado los nombres de host privados al crear el punto de conexión de VPC, no es necesario que especifique la URL de este en los comandos de la CLI ni en la configuración de la aplicación. El nombre de host DNS estándar AWS de Payment Cryptography se resuelve en tu punto de enlace de la VPC. Los SDK AWS CLI y utilizan este nombre de host de forma predeterminada, por lo que puedes empezar a usar el punto de enlace de la VPC para conectarte a un punto de enlace regional de criptografía de AWS pagos sin cambiar nada en tus scripts y aplicaciones.

Para utilizar nombres de host privados, se deben establecer los atributos `enableDnsHostnames` y `enableDnsSupport` de la VPC en `true`. Para establecer estos atributos, usa la operación.

ModifyVpcAttribute Para obtener más información, consulte Ver y actualizar los atributos de DNS de su VPC en la Guía del usuario de Amazon VPC.

Control del acceso a un punto de conexión de VPC

Para controlar el acceso a su punto de enlace de VPC para la criptografía de AWS pagos, adjunte una política de punto de enlace de VPC a su punto de enlace de VPC. La política de puntos finales determina si los principales pueden usar el punto de enlace de la VPC para realizar operaciones de criptografía de pagos con recursos específicos AWS de criptografía de pagos. AWS

Puede crear una política de punto de conexión de VPC cuando cree el punto de conexión y puede cambiar la política de punto de conexión de VPC en cualquier momento. Utilice la consola de administración de la VPC o las operaciones o. CreateVpcEndpoint ModifyVpcEndpoint También puedes crear y cambiar una política de puntos finales de la VPC mediante una AWS CloudFormation plantilla. Para obtener ayuda sobre el uso de la consola de administración de la VPC, consulte Creación de un punto de conexión de interfaz y Modificación de un punto de conexión de interfaz en la Guía de AWS PrivateLink .

Para obtener ayuda sobre cómo escribir y dar formato a un documento de política JSON, consulte la Referencia de políticas JSON de IAM en la Guía del usuario de IAM.

Temas

- Uso de políticas de punto de conexión de VPC
- Política de puntos de conexión de VPC predeterminada
- Creación de una política de punto de conexión de VPC
- Visualización de una política de punto de conexión de VPC

Uso de políticas de punto de conexión de VPC

Para que una solicitud de criptografía de AWS pagos que utilice un punto final de la VPC tenga éxito, el principal necesita permisos de dos fuentes:

- Una política basada en la identidad debe conceder al principal permiso para invocar la operación en el recurso (claves o alias de criptografía AWS de pagos).
- Una política de extremo de VPC debe conceder a la entidad principal permiso para utilizar el punto de conexión para realizar la solicitud.

Por ejemplo, una política de claves puede conceder al principal permiso para ejecutar la función de desencriptación de una clave de criptografía de pago determinada AWS . Sin embargo, es posible que la política de terminales de la VPC no permita al principal invocar esas claves de criptografía de AWS pagos mediante `Decrypt` el punto de conexión.

O bien, una política de puntos de conexión de la VPC puede permitir que un principal utilice el punto de conexión para invocar determinadas claves de StopKeyUsage criptografía de AWS pagos. Sin embargo, si el principal no tiene esos permisos de una política de IAM, la solicitud no se realiza correctamente.

Política de puntos de conexión de VPC predeterminada

Cada punto de conexión de VPC tiene una política de punto de conexión de VPC, pero no es necesario que especifique la política. Si no especifica una política, la política de punto de conexión predeterminada permite todas las operaciones de todas las entidades principales en todos los recursos del punto de conexión.

Sin embargo, en el caso de los recursos de criptografía de AWS pagos, el principal también debe tener permiso para invocar la operación desde una política de IAM. Por lo tanto, en la práctica, la política predeterminada dice que si una entidad principal tiene permiso para llamar a una operación en un recurso, también puede llamarla mediante el punto de conexión.

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

Para permitir que las entidades principales utilicen el punto de conexión de VPC solo para un subconjunto de sus operaciones permitidas, cree o modifique la política de puntos de conexión de VPC.

Creación de una política de punto de conexión de VPC

Una política de punto de conexión de VPC determina si una entidad principal tiene permiso para utilizar el punto de conexión de VPC para realizar operaciones en un recurso. En el caso de los

recursos de criptografía de AWS pagos, el director también debe tener el permiso de una política de IAM para realizar las operaciones.

Cada declaración de política de puntos de conexión de VPC requiere los siguientes elementos:

- La entidad principal que puede realizar acciones
- Las acciones que se pueden realizar
- Los recursos en los que se pueden llevar a cabo las acciones

La declaración de política no especifica el punto de conexión de VPC. En cambio, se aplica a cualquier punto de conexión de VPC al que esté asociada dicha política. Para obtener más información, consulte Control del acceso a los servicios con puntos de conexión de VPC en la guía del usuario de Amazon VPC.

El siguiente es un ejemplo de una política de punto final de VPC para AWS la criptografía de pagos. Cuando se conecta a un punto final de la VPC, esta política permite usar el punto de enlace de la VPC `ExampleUser` para llamar a las operaciones especificadas en las claves de criptografía de pagos especificadas AWS . Antes de usar una política como esta, sustituya el identificador principal y clave del ejemplo por valores válidos de su cuenta.

```
{
   "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 registra todas las operaciones que utilizan el punto final de la VPC. Sin embargo, sus CloudTrail registros no incluyen las operaciones solicitadas por las personas principales de otras cuentas ni las operaciones relacionadas con las claves de criptografía de AWS pagos de otras cuentas.

Por lo tanto, tal vez te interese crear una política de puntos de conexión de la VPC que impida a los usuarios principales de cuentas externas utilizar el punto de conexión de la VPC para ejecutar cualquier operación de criptografía de AWS pagos en cualquier clave de la cuenta local.

En el siguiente ejemplo, se utiliza la clave de condición aws: PrincipalAccount global para denegar el acceso a todos los principales en todas las operaciones con todas las claves de criptografía de AWS pagos, a menos que el principal esté en la cuenta local. Antes de utilizar una política como esta, sustituya el ID de la cuenta de ejemplo por uno válido.

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

Visualización de una política de punto de conexión de VPC

Para ver la política de puntos finales de la VPC para un punto final, utilice la consola de administración de la VPC o la operación. DescribeVpcEndpoints

El siguiente AWS CLI comando obtiene la política del punto de enlace con el ID de punto de enlace de la VPC especificado.

Antes de ejecutar este comando, reemplace el ID de punto de conexión de ejemplo por uno válido de su cuenta.

```
$ aws ec2 describe-vpc-endpoints \
--query 'VpcEndpoints[?VpcEndpointId==`vpce-1234abcdf5678c90a`].[PolicyDocument]'
--output text
```

Utilizar un punto de conexión de VPC en una declaración de política

Puedes controlar el acceso a los recursos y las operaciones de criptografía de AWS pagos cuando la solicitud proviene de una VPC o utiliza un punto de conexión de la VPC. Para ello, utiliza una política de IAM ???

- Utilice la clave de condición `aws:sourceVpce` para conceder o restringir el acceso en función del punto de conexión de VPC.
- Utilice la clave de condición `aws:sourceVpc` para conceder o restringir el acceso en función de la VPC que aloja el punto de conexión privado.

> **Note**
>
> La clave de `aws:sourceIP` condición no entra en vigor cuando la solicitud proviene de un punto de enlace de Amazon VPC. Para restringir las solicitudes a un punto de conexión de VPC, utilice las claves de condición `aws:sourceVpce` o `aws:sourceVpc`. Para obtener más información, consulte Administración de identidades y accesos para puntos de conexión de VPC y servicios de puntos de conexión de VPC en la Guía de AWS PrivateLink .

Puede usar estas claves de condición globales para controlar el acceso a las claves de criptografía de AWS pagos y a los alias, así como a operaciones de CreateKey este tipo que no dependen de ningún recurso en particular.

Por ejemplo, el siguiente ejemplo de política de claves permite a un usuario realizar determinadas operaciones criptográficas con claves de criptografía de AWS pagos solo cuando la solicitud utiliza el punto final de la VPC especificado, lo que bloquea el acceso tanto desde Internet como desde las AWS PrivateLink conexiones (si está configurado). Cuando un usuario solicita la criptografía de AWS pagos, el ID de punto de enlace de la VPC de la solicitud se compara con el valor de la clave de `aws:sourceVpce` condición de la política. Si no coinciden, la solicitud se deniega.

Para usar una política como esta, reemplaza el ID del marcador de posición y los Cuenta de AWS ID de los puntos de enlace de la VPC por valores válidos para tu cuenta.

JSON

```
{
    "Id": "example-key-1",
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "EnableIAMPolicies",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:root"
                ]
            },
            "Action": [
                "payment-cryptography:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "RestrictUsageToMyVPCEndpoint",
            "Effect": "Deny",
            "Principal": "*",
            "Action": [
                "payment-cryptography:EncryptData",
                "payment-cryptography:DecryptData"
            ],
            "Resource": "arn:aws:payment-cryptography:us-east-1:111122223333:key/
*",
            "Condition": {
                "StringNotEquals": {
                    "aws:sourceVpce": "vpce-1234abcdf5678c90a"
                }
            }
        }

    ]
}
```

También puedes usar la clave de `aws:sourceVpc` condición para restringir el acceso a tus claves de criptografía de AWS pagos en función de la VPC en la que reside el punto de enlace de la VPC.

El siguiente ejemplo de política de claves permite que los comandos administren las claves de criptografía de AWS pagos solo cuando provienen de ellas. `vpc-12345678` Además, permite los comandos que utilizan las claves de criptografía de AWS pagos para las operaciones criptográficas solo cuando provienen de. `vpc-2b2b2b2b` Podría utilizar una política como esta en caso de que una aplicación se ejecute en una VPC, pero utiliza una segunda VPC aislada para funciones de administración.

Para usar una política como esta, sustituya el Cuenta de AWS identificador del marcador de posición y los identificadores de los puntos de conexión de la VPC por valores válidos para su cuenta.

JSON

```
{
    "Id": "example-key-2",
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAdminActionsFromVPC12345678",
            "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": "AllowKeyUsageFromVPC2b2b2b2b",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111122223333"
```

```
            },
            "Action": [
                "payment-cryptography:Encrypt*",
                "payment-cryptography:Decrypt*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:sourceVpc": "vpc-2b2b2b2b"
                }
            }
        },
        {
            "Sid": "AllowListReadActionsFromEverywhere",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111122223333"
            },
            "Action": [
                "payment-cryptography:List*",
                "payment-cryptography:Get*"
            ],
            "Resource": "*"
        }
    ]
}
```

Registro de su punto de conexión de VPC

AWS CloudTrail registra todas las operaciones que utilizan el punto final de la VPC. Cuando una solicitud de criptografía de AWS pagos utiliza un punto final de la VPC, el ID del punto de enlace de la VPC aparece en la entrada del AWS CloudTrail registro que registra la solicitud. Puedes usar el ID de punto de conexión para auditar el uso de tu punto de conexión de la VPC de criptografía de AWS pagos.

Para proteger tu VPC, las solicitudes denegadas por una política de puntos de conexión de la VPC, pero que de lo contrario se hubieran permitido, no se registran. AWS CloudTrail

Por ejemplo, esta entrada de log de ejemplo registra una solicitud GenerateMac que ha utilizado el punto de enlace de la VPC. El campo `vpcEndpointId` aparece al final de la entrada de log.

```
{
      "eventVersion": "1.08",
      "userIdentity": {
          "principalId": "TESTXECZ5U9M4LGF2N6Y5:i-98761b8890c09a34a",
          "arn": "arn:aws:sts::111122223333:assumed-role/samplerole/
i-98761b8890c09a34a",
          "accountId": "111122223333",
          "accessKeyId": "TESTXECZ5U2ZULLHHMJG",
          "sessionContext": {
              "sessionIssuer": {
                  "type": "Role",
                  "principalId": "TESTXECZ5U9M4LGF2N6Y5",
                  "arn": "arn:aws:iam::111122223333:role/samplerole",
                  "accountId": "111122223333",
                  "userName": "samplerole"
              },
              "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,
```

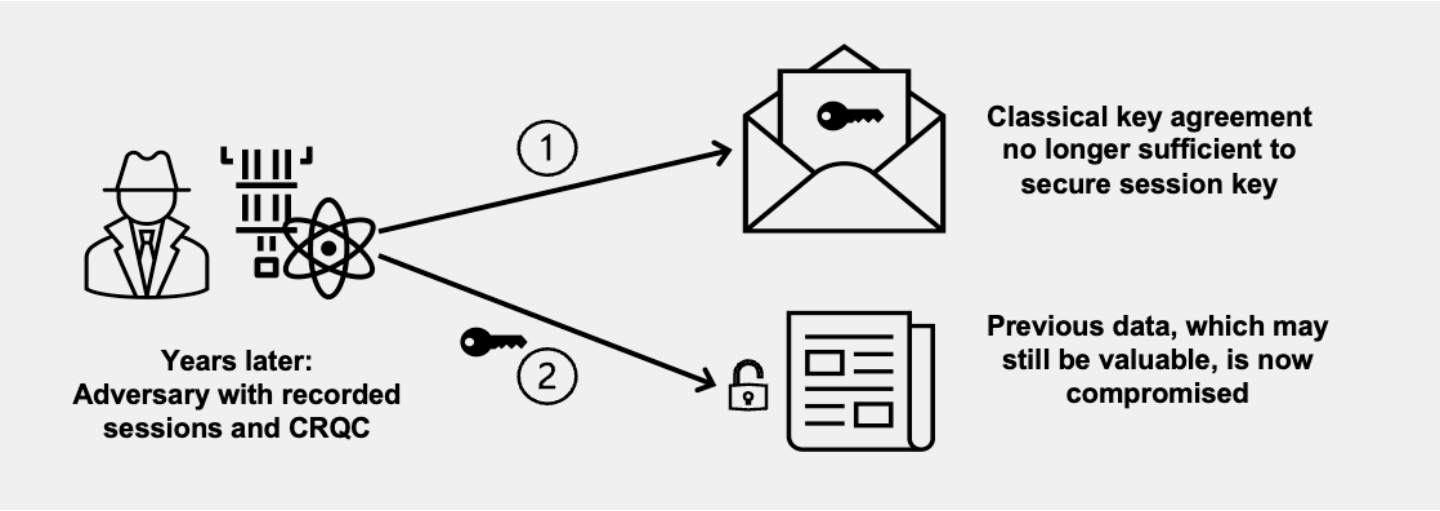
```
                    "verify": true,
                    "deriveKey": false,
                    "noRestrictions": false
                }
            },
            "exportable": true
        },
        "responseElements": {
            "key": {
                "keyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
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": "vpce-1234abcdf5678c90a",
```

```
      "eventCategory": "Management",
      "tlsDetails": {
          "tlsVersion": "TLSv1.3",
          "cipherSuite": "TLS_AES_128_GCM_SHA256",
          "clientProvidedHostHeader": "vpce-1234abcdf5678c90a-
oo28vrvr.controlplane.payment-cryptography.us-east-1.vpce.amazonaws.com"
      }
  }
```

Usar el cifrado TLS híbrido postcuántico

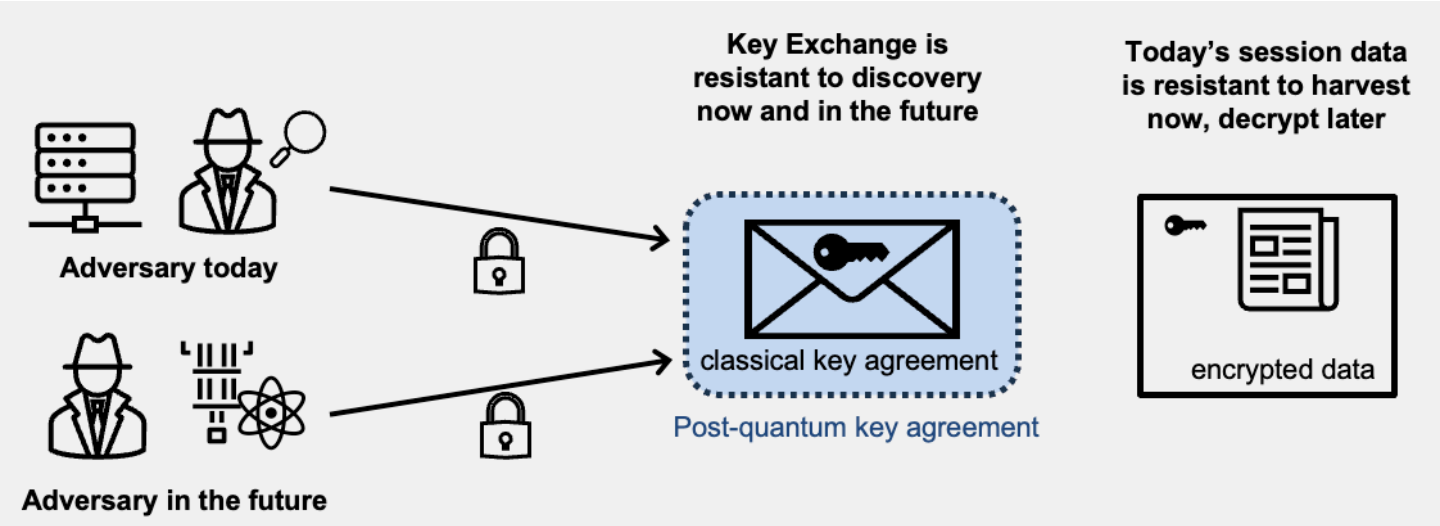
AWS La criptografía de pagos y muchos otros servicios admiten una opción híbrida de intercambio de claves poscuánticas para el protocolo de cifrado de red Transport Layer Security (TLS). Puede usar esta opción de TLS cuando se conecte a los puntos finales de la API o cuando utilice los SDK. AWS Estas características opcionales de intercambio híbrido postcuántico de claves son al menos tan seguras como el cifrado TLS que utilizamos hoy en día y es muy probable que aporten beneficios de seguridad adicionales.

Los datos que envías a los servicios habilitados están protegidos mientras están en tránsito mediante el cifrado que proporciona una conexión de seguridad de la capa de transporte (TLS). Los conjuntos de cifrado clásicos basados en RSA y ECC que AWS Payment Cryptography admite para las sesiones TLS hacen que los ataques por fuerza bruta contra los mecanismos de intercambio de claves sean inviables con la tecnología actual. Sin embargo, si en el futuro se utilizan ordenadores cuánticos (CRQC) a gran escala o relevantes desde el punto de vista criptográfico, los mecanismos de intercambio de claves TLS existentes serán susceptibles a estos ataques. Es posible que los adversarios empiecen a recopilar datos cifrados ahora con la esperanza de poder descifrarlos en el futuro (cosechar ahora y descifrarlos después). Si está desarrollando aplicaciones que se basan en la confidencialidad a largo plazo de los datos transmitidos a través de una conexión TLS, debería plantearse la posibilidad de migrar a la criptografía poscuántica antes de que los ordenadores cuánticos a gran escala estén disponibles para su uso. AWS está trabajando para prepararse para este futuro, y queremos que también estés bien preparado.



Para proteger los datos cifrados hoy contra posibles ataques futuros, AWS está colaborando con la comunidad criptográfica en el desarrollo de algoritmos de resistencia cuántica o poscuánticos. AWS ha implementado conjuntos híbridos de cifrado de intercambio de claves poscuánticos que combinan elementos clásicos y poscuánticos para garantizar que su conexión TLS sea al menos tan fuerte como lo sería con los conjuntos de cifrado clásicos.

Estos conjuntos de cifrado híbridos están disponibles para su uso en sus cargas de trabajo de producción cuando utilice versiones recientes de los SDK. AWS Para obtener más información sobre enable/disable este comportamiento, consulta Cómo habilitar el TLS poscuántico híbrido.



Acerca del intercambio de claves postcuántico híbrido en TLS

Los algoritmos que se AWS utilizan son un híbrido que combina la curva elíptica Diffie-Hellman (ECDH), un algoritmo clásico de intercambio de claves que se utiliza actualmente en TLS, con

Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM), un algoritmo de cifrado y establecimiento de claves de clave pública que el Instituto Nacional de Estándares y Tecnología (NIST) ha designado como su primer algoritmo estándar de acuerdo de claves poscuánticas. Este híbrido utiliza cada uno de los algoritmos de forma independiente para generar una clave. Luego combina las dos claves criptográficamente.

Modo de uso

Para obtener información sobre cómo habilitar el TLS poscuántico híbrido en los SDK, consulte Habilitar el TLS poscuántico híbrido. AWS https://docs.aws.amazon.com/sdkref/latest/guide/pqtls-details.html

Más información sobre PQC

Para obtener información sobre el proyecto de criptografía poscuántica del Instituto Nacional de Estándares y Tecnología (NIST), consulte Criptografía. Post-Quantum

Para obtener información sobre la estandarización de la criptografía poscuántica del NIST, consulte Estandarización de la criptografía. Post-Quantum

Habilitar el TLS poscuántico híbrido

Este tema se ha trasladado a la Guía de referencia sobre AWS los SDK y las herramientas.

Prácticas recomendadas de seguridad para AWS Payment Cryptography

AWS La criptografía de pagos admite muchas funciones de seguridad integradas o que puedes implementar de manera opcional para mejorar la protección de tus claves de cifrado y garantizar que se utilizan para los fines previstos, incluidas las políticas de IAM, un amplio conjunto de claves de condiciones de políticas para refinar tus políticas de claves y políticas de IAM, y la aplicación integrada de las reglas de PIN de PCI en relación con los bloques de claves.

> **Important**
>
> Las directrices generales proporcionadas no representan una solución de seguridad completa. Dado que no todas las mejores prácticas son adecuadas para todas las situaciones, no se pretende que sean prescriptivas.

- Uso de las claves y modos de uso: la criptografía de AWS pagos sigue y aplica las restricciones de uso y modo de uso de las claves, tal como se describe en la especificación de bloques de claves de intercambio seguro e interoperable de claves ANSI X9 TR 31-2018 y es coherente con el requisito de seguridad 18-3 del PIN de PCI. Esto limita la capacidad de usar una sola clave para varios propósitos y vincula criptográficamente los metadatos de la clave (como las operaciones permitidas) al propio material de la clave. AWS La criptografía de pagos aplica automáticamente estas restricciones, por ejemplo, que una clave de cifrado de claves (TR31_K0_KEY_ENCRYPTION_KEY) no se puede utilizar también para descifrar datos. Consulte Descripción de los atributos clave de AWS Clave de criptografía de pagos para obtener más detalles.
- Limitar el uso compartido de material de clave simétrica: sólo comparta material de claves simétricas (como claves de cifrado de pines o claves de cifrado de claves) con un máximo de otra entidad. Si es necesario enviar material confidencial a más entidades o socios, cree claves adicionales. AWS La criptografía de pagos nunca expone material de claves simétricas o asimétricas de forma transparente.
- Usa alias o etiquetas para asociar claves a determinados casos de uso o socios: los alias se pueden usar para indicar fácilmente el caso de uso asociado a una clave, por ejemplo, para indicar una clave de verificación de tarjeta asociada alias/BIN_12345_CVK al BIN 12345. Para ofrecer más flexibilidad, considere la posibilidad de crear etiquetas como bin=12345, use_case=acquiring,country=us,partner=foo. Los alias y las etiquetas también se pueden utilizar para limitar el acceso como, por ejemplo, para aplicar controles de acceso entre los casos de uso emisor y adquirente.
- Practicar el acceso con privilegio mínimo: IAM puede utilizarse para limitar el acceso de producción a los sistemas en lugar de a los individuos, como prohibir a los usuarios individuales la creación de claves o la ejecución de operaciones criptográficas. IAM también puede utilizarse para limitar el acceso tanto a comandos como a claves que pueden no ser aplicables para su caso de uso, como limitar la capacidad de generar o validar pines para un adquirente. Otra forma de utilizar el acceso con privilegio mínimo es restringir las operaciones sensibles (como la importación de claves) a cuentas de servicio específicas. Para ver ejemplos, consulte AWS Ejemplos de políticas de criptografía de pagos basadas en la identidad.

Véase también

- Gestión de identidad y acceso para AWS Payment Cryptography
- Consulte Prácticas recomendadas en IAM en la Guía del usuario de IAM.

Validación de conformidad para criptografía AWS de pagos

Al igual que con otros AWS servicios, los clientes requieren una comprensión clara del modelo de responsabilidad compartida en materia de seguridad y cumplimiento. Al tratarse de un servicio que admite pagos específicamente, es especialmente importante que los clientes de criptografía de AWS pagos comprendan el cumplimiento de las normas PCI aplicables. AWS Las evaluaciones de PCI DSS y PCI 3DS incluyen la criptografía de pagos. AWS Es posible que haya referencias al servicio en las guías de responsabilidad compartida, disponibles en, para estos informes. AWS Artifact Las evaluaciones de seguridad y Point-to-Point cifrado con PIN PCI (P2PE) son específicas de la criptografía de pagos. AWS

En esta sección se proporciona información sobre el estado y el alcance del cumplimiento del servicio, así como información que le será útil a la hora de planificar las evaluaciones de sus aplicaciones relacionadas con la seguridad de los códigos PCI PIN y PCI P2PE.

Temas

- Conformidad del servicio
- Planificación del cumplimiento de los PIN
- Uso del componente de descifrado de criptografía de AWS pagos en soluciones P2PE

Conformidad del servicio

Los auditores externos evalúan la seguridad y el cumplimiento de la criptografía de AWS pagos como parte de varios programas de AWS cumplimiento. Esto incluye SOC, PCI y otros.

AWS Además de los estándares PCI DSS y PCI 3DS, la criptografía de pagos se ha evaluado en relación con varios estándares PCI. Estos incluyen el cifrado PCI PIN Security (PCI PIN) y PCI (P2PE). Point-to-Point Consulte las certificaciones y las guías AWS Artifact de conformidad disponibles.

Para ver una lista de AWS los servicios incluidos en el ámbito de los programas de conformidad específicos, consulte Servicios de AWS incluidos . Para obtener información general, consulte Programas de conformidad de AWS.

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

Su responsabilidad en materia de cumplimiento al utilizar la criptografía de AWS pagos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar a garantizar el cumplimiento:

- Guías de inicio rápido de seguridad y conformidad: estas guías de implementación analizan consideraciones sobre arquitectura y proporcionan los pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad en AWS.
- AWS Recursos de https://aws.amazon.com/compliance/resources/ de cumplimiento: esta colección de libros de trabajo y guías puede aplicarse a su sector y ubicación.
- Evaluación de recursos con reglas en la Guía para desarrolladores de AWS Config :AWS Config evalúa en qué medida las configuraciones de sus recursos cumplen las prácticas internas, las directrices del sector y las normativas.
- AWS Security Hub CSPM—Este AWS servicio proporciona una visión integral del estado de su seguridad AWS que le ayuda a comprobar el cumplimiento de los estándares y las mejores prácticas del sector de la seguridad.

Planificación del cumplimiento de los PIN

Esta guía describe la documentación y las pruebas que necesitará para preparar una evaluación del PIN PCI de su aplicación de procesamiento de PIN que utilice criptografía de AWS pagos.

Al igual que ocurre con otras normas Servicios de AWS y normas de conformidad, es su responsabilidad utilizar el servicio de forma segura, configurando el control de acceso y utilizando los parámetros de seguridad de conformidad con los requisitos del PIN de PCI. En esta guía se analizarán esas configuraciones cuando sean adecuadas para cumplir un requisito.

Temas

- Temas comunes
- Alcance de la evaluación
- Operaciones de procesamiento de transacciones

Temas comunes

La migración de aplicaciones desde una conexión a HSM a un servicio gestionado, como la criptografía de AWS pagos, plantea problemas y conceptos comunes para los clientes y sus

asesores. En esta sección se proporciona información para aclarar cómo el uso seguro del servicio aborda estas situaciones.

Temas

- Responsabilidad compartida
- Configuración mínima de HSM
- Intercambio de claves entre el cliente y APC

Responsabilidad compartida

Los clientes que hayan asumido toda la responsabilidad en materia de seguridad y cumplimiento de las aplicaciones reestructurarán su conformidad para aprovechar la gestión de claves, los controles de seguridad y las capacidades gestionadas de HSM («el servicio») de AWS Payment Cryptography. Esto modificará por completo algunos requisitos AWS, como lo atestiguan las evaluaciones de terceros de AWS Payment Cryptography. Algunos requisitos se compartirán entre la solicitud del cliente y el servicio. Una aplicación es responsable de:

- Proporcionar información precisa al servicio
- Utilizar controles de seguridad de acuerdo con las recomendaciones del servicio y los requisitos de seguridad del PCI PIN
- Implementar los controles de seguridad necesarios mediante las herramientas proporcionadas por el servicio

Los clientes y sus asesores utilizarán las guías de responsabilidad compartida e implementación publicadas con las certificaciones de cumplimiento AWS Artifact para implementar los controles y la supervisión del control, y luego planificarán y completarán las evaluaciones.

Configuración mínima de HSM

El estándar de seguridad de datos PCI, el estándar fundamental de otros estándares PCI, exige que todos los sistemas estén configurados con la funcionalidad mínima necesaria para su funcionamiento. Los estándares PCI PIN, P2PE y otros estándares de soluciones aplican este requisito a la solución. HSMs HSMs debe habilitar únicamente las funciones necesarias para la solución.

AWS los servicios deben tratarse como sistemas y configurarse para ofrecer la funcionalidad mínima requerida. El estándar de seguridad de datos del sector de tarjetas de pago (PCI DSS) v4.0 de AWS

recomienda usar IAM para configurar la funcionalidad mínima de cada servicio de AWS que utilice la solución. Esto también se aplica a la criptografía de pagos. AWS Las políticas de IAM permiten permisos detallados para restringir las funciones criptográficas únicamente a los componentes de la aplicación que dependen de ellas.

Intercambio de claves entre el cliente y APC

PIN PIN Los requisitos de seguridad 8-4 y 15-2 exigen que el intercambio y la carga de las claves estén autenticadas y su integridad esté protegida. En el caso de la carga remota de los POI mediante claves, que se describe funcionalmente en el ANSI/ASC X9 TR-34 y se rige por el anexo A del PCI PIN, las claves públicas suelen figurar en certificados firmados por una autoridad de certificación que cumpla con el anexo A2. Para los intercambios entre organizaciones, las claves públicas utilizan otros mecanismos de autenticidad e integridad.

Todas las interacciones entre el cliente y AWS se realizan a través de AWS APIs, que autentica mutuamente cada llamada a la API y garantiza la integridad de las llamadas y las respuestas mediante TLS. AWS Identity and Access Management gestiona la autenticación de la aplicación del cliente con mecanismos como los tokens de seguridad y SigV4. El cliente autentica los puntos de enlace de la API de AWS mediante la autenticación del servidor TLS, que está integrada en AWS. SDKs Luego, TLS garantiza la confidencialidad e integridad de todos los datos que se transmiten entre el cliente y cada API de AWS.

APC APIs GetParametersForImport e ImportKey implementan una transferencia de claves del cliente al servicio. Si bien la autoridad de certificación (CA) proporcionada por GetParametersForImport ella no cumple con el anexo A2, es segura y exclusiva de la cuenta. Si bien no se puede confiar en que esta CA cumpla con los requisitos 8-4 y 15-2, sí que proporciona una verificación de integridad de la clave importada. También puede utilizar su propia CA aprovechando la API. GetCertificateSigningRequest

Los mecanismos que proporcionan autenticación de clave pública y garantía de integridad son:

- Autenticación proporcionada por la API de AWS
- La función MAC del certificado que proporciona proporciona la integridad de la clave GetParametersForImport, incluso si la información de identidad del certificado no es de confianza. La integridad de la clave también está asegurada por el MAC utilizado por TLS, protegiendo la sesión entre el cliente y AWS.

Los certificados y bloques de claves proporcionados por APC cumplen con el anexo A1, que especifica los requisitos para los certificados y la protección de claves mediante métodos asimétricos.

Alcance de la evaluación

El primer paso para planificar cualquier evaluación es documentar el alcance. En el caso del PCI PIN, el objetivo son los sistemas y procesos que protegen los PIN, incluida la protección de las claves criptográficas y los dispositivos que los protegen: terminales de pago, también denominados puntos de interacción (POI), HSM y otros dispositivos criptográficos seguros (SCD).

No abordaremos los requisitos sobre los que usted es plenamente responsable, ya que se refieren a áreas fuera del ámbito del servicio. Por ejemplo, la configuración y el aprovisionamiento de terminales de pago. Consulte la Guía de responsabilidad compartida sobre criptografía de AWS pagos para el PIN PCI, disponible en AWS Artifact

Temas

- Responsabilidad compartida
- High-Level Diagramas de red
- Tabla de claves
- Referencias de documentos

Responsabilidad compartida

AWS Payment Cryptography es una organización de cifrado y apoyo (ESO) y una PIN-Acquiring Third-Party entidad administradora (TPS), según se define en el Programa de seguridad PIN de Visa y figura en el Registro mundial de proveedores de servicios de Visa, bajo la denominación «Amazon Web Services, LLC». Esto significa que Visa permite que el PIN-Acquiring Third-Party VisaNet procesador (VNP), el VisaNet procesador PIN-Acquiring cliente que actúa como proveedor de servicios y otros proveedores de TPS y ESO utilicen el servicio sin necesidad de una evaluación adicional por parte de los evaluadores de PIN del cliente (evaluadores de PIN calificados por PCI o PCI QPA).

Otras marcas de tarjetas o proveedores de redes de pago pueden confiar en el programa de seguridad de PIN de Visa o tener sus propios programas. Póngase en contacto con nosotros AWS Support si tiene preguntas sobre el cumplimiento de los servicios de otros programas de redes de pago.

AWS proporciona la certificación de conformidad (AOC) de seguridad (AOC) del PCI con PIN y la Guía de responsabilidad compartida para AWS la criptografía de pagos en. AWS Artifact El uso de proveedores de servicios en el procesamiento de los PIN es algo habitual desde hace muchos años; sin embargo, el estándar de seguridad PCI PIN, hasta la versión 3.1, no aborda la gestión de proveedores de servicios de terceros. Tampoco lo hace el programa de seguridad de PIN de Visa. Los clientes de QPA han seguido el modelo establecido en el PCI, el DSS, el AOC y la Guía de Responsabilidad Compartida, según el cual la prueba de AWS cumplimiento de los requisitos aplicables se considera satisfactoria.

High-Level Diagramas de red

La plantilla de informes de PIN PCI requiere: «Para las entidades que se dedican al procesamiento de transacciones basadas en PIN, proporcione un esquema de red que describa los flujos de transacciones basadas en PIN con el uso del tipo de clave asociado. Además, los KIF y las entidades que se dedican a la distribución remota de claves mediante técnicas asimétricas deberían proporcionar flujos de material de codificación»

AWS Payment Cryptography ha presentado la estructura interna del servicio para nuestra evaluación del PIN PCI. Sus diagramas ilustrarán cómo llamar a las API del servicio para procesar los PIN.

Ejemplo de diagrama de red de alto nivel para aplicaciones de PIN que utilizan criptografía AWS de pagos:

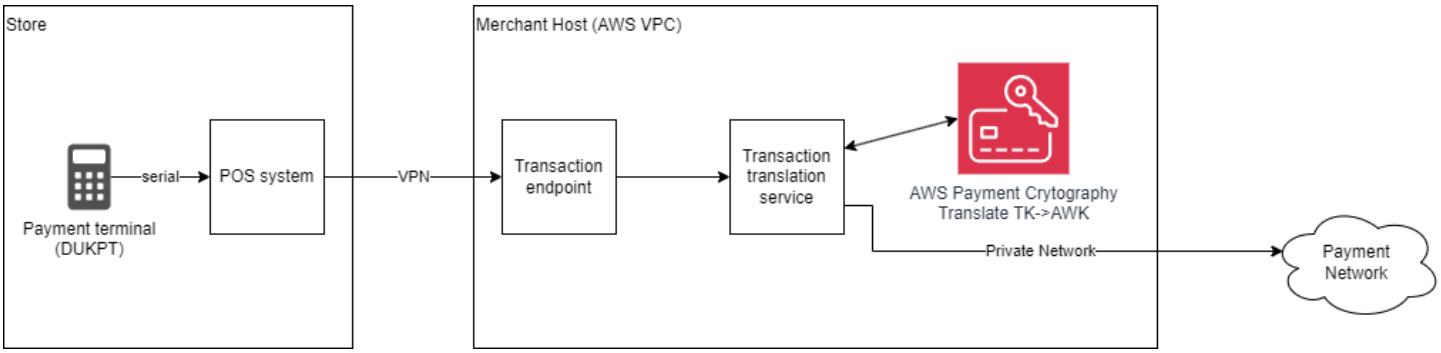


Tabla de claves

El informe exige que se enumeren todas las claves que protegen los PIN, directa o indirectamente. Todas las claves que existan en el servicio se pueden incluir en la ListKeysAPI.

Asegúrese de proporcionar la lista de claves de todas las regiones y cuentas que poseen las claves de su aplicación.

Referencias de documentos

La documentación y las recomendaciones del proveedor para el uso seguro de la criptografía de AWS pagos se encuentran en la Guía del usuario y en la Referencia de la API. Estos están enlazados, según corresponda, en esta guía.

Operaciones de procesamiento de transacciones

Los requisitos del PIN PCI se organizan en los objetivos de control. Cada objetivo de control agrupa los requisitos para proteger un aspecto de la seguridad de los PIN.

Temas

- Objetivo de control 1: Los PIN utilizados en las transacciones regidas por estos requisitos se procesan utilizando equipos y metodologías que garantizan su seguridad.
- Objetivo de control 2: Las claves criptográficas utilizadas para la gestión del PIN encryption/decryption y las claves relacionadas se crean mediante procesos que garantizan que no sea posible predecir ninguna clave ni determinar si ciertas claves son más probables que otras.
- Objetivo de control 3: Las claves se transportan o transmiten de forma segura.
- Objetivo de control 4: Key-loading los PIN-acceptance dispositivos HSM y POI se gestionan de forma segura.
- Objetivo de control 5: Las claves se utilizan de forma que se evite o detecte su uso no autorizado.
- Objetivo de control 6: Las claves se administran de forma segura.
- Objetivo de control 7: El equipo utilizado para procesar los PIN y las claves se gestiona de forma segura.

Objetivo de control 1: Los PIN utilizados en las transacciones regidas por estos requisitos se procesan utilizando equipos y metodologías que garantizan su seguridad.

Requisito 1: Los HSM utilizados por AWS Payment Cryptography se evaluaron como parte de nuestra evaluación de los PIN PCI. Para los clientes que utilizan el servicio, los requisitos 1 a 3 y 1 a 4 están «vigentes» en relación con el HSM gestionado por el servicio. Los resultados de HSM indicarán que las pruebas fueron certificadas por la QPA. AWS El certificado de conformidad con el PIN está disponible para consultarlo. AWS Artifact Será necesario inventariar y hacer referencia a otros SCD de su solución, como el POI.

Requisito 2: La documentación de sus procedimientos debe especificar cómo se protegen los PIN del titular de la tarjeta en lo que respecta a su divulgación a su personal, los protocolos de traducción

de los PIN implementados y la protección durante el procesamiento en línea y fuera de línea. Además, su documentación debe contener un resumen de los métodos de administración de claves criptográficas utilizados en cada zona.

Requisito 3: El POI debe configurarse para el cifrado y la transmisión seguros del PIN. AWS La criptografía de pagos solo admite las traducciones de bloques de PIN especificadas en el requisito 3-3.

Requisito 4: La aplicación no debe almacenar bloques de PIN. Los bloques de PIN, incluso cifrados, no deben conservarse en los diarios o registros de transacciones. El servicio no almacena los bloques de PIN y la evaluación de los PIN verifica que no estén en los registros.

Tenga en cuenta que el estándar de seguridad PCI PIN se aplica a la adquisición, «la gestión, el procesamiento y la transmisión seguros de los datos del número de identificación personal (PIN) durante el procesamiento de transacciones con tarjetas de pago en línea y fuera de línea en cajeros automáticos y terminales de punto de venta (POS)», como se indica en la norma. Sin embargo, el estándar se utiliza a menudo para evaluar la gestión de claves criptográficas para pagos que no están dentro del ámbito previsto. Esto puede incluir casos de uso del emisor en los que se almacenan los PIN. Las excepciones a los requisitos para estos casos deben acordarse con el público destinatario de la evaluación.

Objetivo de control 2: Las claves criptográficas utilizadas para la gestión del PIN encryption/decryption y las claves relacionadas se crean mediante procesos que garantizan que no sea posible predecir ninguna clave ni determinar si ciertas claves son más probables que otras.

Requisito 5: La generación de claves mediante criptografía de AWS pago se evaluó como parte de nuestra evaluación del PIN PCI. Esto se puede especificar en la columna «Generado por» de la tabla de claves.

Requisito 6: Los controles de seguridad de las claves contenidas en la criptografía de AWS pagos se evaluaron como parte de la evaluación del PIN PCI del servicio. Incluya descripciones de los controles de seguridad relacionados con la generación de claves en su aplicación y con cualquier otro proveedor de servicios.

Requisito 7: Debe tener una documentación sobre la política de generación de claves que especifique cómo se generan las claves y todas las partes afectadas deben conocerla. procedures/ policies Los procedimientos para la creación de claves mediante la API de APC deben incluir el uso

de funciones con permisos de creación de claves y aprobaciones para ejecutar scripts u otro código que cree claves. AWS CloudTrail los registros contienen todos los CreateKeyeventos con fecha y hora, el ARN clave y los identificadores de usuario. Los números de serie y los registros de los HSM para acceder a los medios físicos se evaluaron como parte de la evaluación del PIN del servicio.

Objetivo de control 3: Las claves se transportan o transmiten de forma segura.

Requisito 8: La transferencia de claves mediante criptografía de AWS pago se evaluó como parte de nuestra evaluación del PIN PCI. Deberás documentar los mecanismos de protección clave de las transferencias antes de importarlas y después de exportarlas desde Payment Cryptography. AWS El servicio proporciona valores de verificación clave para todas las claves a fin de validar su correcta transferencia.

El requisito 8-4 exige que las claves públicas se transmitan de manera que se proteja su integridad y autenticidad. El intercambio entre su aplicación y AWS se controla mediante la autenticación de la aplicación y AWS, mediante AWS Identity and Access Management métodos, la autenticación de punto final de AWS la API a la aplicación mediante certificados de servidor TLS. Además, las claves públicas exportadas o importadas a AWS Payment Cryptography tienen certificados firmados por entidades de certificación efímeras y específicas del cliente (consulte, y). GetPublicKeyCertificateGetParametersForImportGetParametersForExport Estas CA no se pueden utilizar como único método de autenticación porque no cumplen con el anexo A2 de seguridad del PCI PIN. Sin embargo, los certificados siguen garantizando la integridad de las claves públicas, y IAM proporciona la autenticación.

Al intercambiar claves públicas con sus socios comerciales mediante métodos asimétricos, debe garantizar la autenticación de la empresa a través del canal de comunicación, mediante un sitio web seguro de intercambio de archivos, por ejemplo.

Requisito 9: El servicio no utiliza componentes clave de texto claro ni los admite directamente.

Requisito 10: El servicio impone la resistencia relativa de las llaves de protección para su transporte. Usted es responsable de transferir las claves antes de importarlas y después de exportarlas desde AWS Payment Cryptography y de utilizar la API y TR-31 los parámetros que sean precisos para la importación, exportación y generación de las claves. Debe disponer de procedimientos documentados para describir los mecanismos de transferencia de claves y la lista de claves criptográficas utilizadas para la transferencia.

Requisito 11: La documentación de sus procedimientos debe especificar cómo se transportan las llaves. Los procedimientos para la transferencia de claves mediante la API de criptografía de AWS

pagos deben incluir el uso de funciones con permisos de importación y exportación de claves y la aprobación de autorizaciones para ejecutar scripts u otro código que cree claves. AWS CloudTrail los registros contienen todos los eventos y. ImportKeyExportKey

Objetivo de control 4: Key-loading los PIN-acceptance dispositivos HSM y POI se gestionan de forma segura.

Requisito 12: Usted es responsable de cargar las claves de los componentes o recursos compartidos. La gestión de las claves principales del HSM se evaluó como parte de la evaluación del PIN del servicio. AWS La criptografía de pagos no carga claves de recursos compartidos o componentes individuales. Consulte la sección Detalles criptográficos.

Requisitos 13 y 14: Deberá describir la protección clave para las transferencias antes y después de la exportación desde el servicio.

Requisito 15: La criptografía de AWS pagos proporciona valores de verificación de claves para todas las claves del servicio y garantiza la integridad de las claves públicas. Su aplicación es responsable de utilizar estas comprobaciones para validar las claves después de importarlas o exportarlas desde el servicio. Debe documentar los procedimientos para asegurarse de que existe un mecanismo de validación.

El requisito 15-2 exige que las claves públicas se carguen de manera que se proteja su integridad y autenticidad. ImportKey, junto con GetParametersForImport, prevé la validación de los certificados de firma proporcionados. Si los certificados proporcionados son autofirmados, la autenticación debe proporcionarse mediante un mecanismo independiente, por ejemplo, el intercambio seguro de archivos.

Requisito 16: La documentación de sus procedimientos debe especificar cómo se cargan las claves en el servicio. Los procedimientos para la importación de claves mediante la API deben incluir el uso de funciones con permisos de importación de claves y aprobaciones para ejecutar scripts u otro código que cargue claves. AWS CloudTrail los registros contienen todos los ImportKeyeventos. Debe incluir los mecanismos de registro en la documentación. El servicio proporciona valores de comprobación clave para todas las claves a fin de validar la carga correcta de las claves.

Objetivo de control 5: Las claves se utilizan de forma que se evite o detecte su uso no autorizado.

Requisito 17: El servicio proporciona mecanismos, como etiquetas y alias, para las claves que permiten rastrear las relaciones de intercambio de claves. Además, los valores de verificación clave

deben mantenerse separados para demostrar que los valores clave conocidos o predeterminados no se utilizan cuando se comparten las claves.

Requisito 18: El servicio proporciona comprobaciones de integridad de las claves mediante GetKeyy ListKeysmediante eventos de gestión de claves AWS CloudTrail, mediante los cuales se pueden detectar sustituciones no autorizadas o supervisar la sincronización de las claves entre las partes. El servicio almacena las claves exclusivamente en bloques clave. Usted es responsable del almacenamiento y uso de las claves antes de importarlas y después de exportarlas desde AWS Payment Cryptography.

Debe disponer de procedimientos para realizar una investigación inmediata en caso de que se produzca cualquier discrepancia durante el procesamiento de las transacciones basadas en un PIN o en caso de que se produzca algún imprevisto en la gestión de claves.

Requisito 19: El servicio utiliza las claves exclusivamente en los bloques clave, en la ejecución KeyUsage y en otros atributos clave para todas las operaciones. KeyModeOfUse Esto incluye la restricción de las operaciones con clave privada. Debe utilizar sus claves públicas para un único propósito, por ejemplo, el cifrado o la verificación de la firma digital, pero no para ambos. Debe utilizar cuentas separadas para la producción y test/development los sistemas.

Requisito 20: Usted es responsable de este requisito.

Objetivo de control 6: Las claves se administran de forma segura.

Requisito 21: El almacenamiento y el uso de las claves con criptografía de AWS pagos se evaluaron como parte de la evaluación del PIN PCI del servicio. En cuanto a los requisitos de almacenamiento relacionados con los componentes clave, usted es responsable de almacenarlos tal y como se describe en los puntos 21-2 y 21-3. Deberá describir los principales mecanismos de protección en la documentación de su política antes de la importación al servicio y después de su exportación desde el servicio.

Requisito 22: Los principales procedimientos de compromiso en materia de criptografía de AWS pagos se evaluaron como parte de la evaluación del PIN PCI del servicio. Deberá describir los principales procedimientos de detección y respuesta a las vulnerabilidades, incluidas la supervisión y la respuesta a las notificaciones emitidas por ellos. AWS

Requisito 23: La criptografía de AWS pagos no admite variantes ni otros métodos de cálculo de claves reversibles. Las claves principales de APC o las claves cifradas por ellas nunca están disponibles para los clientes. El uso del cálculo de claves reversibles se evaluó como parte de la evaluación del PIN PCI del servicio.

Requisito 24: Prácticas de destrucción de claves privadas y secretas internas La criptografía de AWS pagos se evaluó como parte de la evaluación del PIN PCI del servicio. Deberá describir el procedimiento de destrucción de claves antes de importarlas a APC y después de exportarlas desde APC. Los requisitos de destrucción relacionados con los componentes clave (24 y 24 horas y 23 horas) siguen siendo tu responsabilidad.

Requisito 25: El acceso a las claves secretas y privadas de la criptografía de AWS pagos se evaluó como parte de la evaluación del PIN PCI del servicio. Deberá disponer de un proceso y una documentación para controlar el acceso a las claves antes de importarlas y después de exportarlas desde AWS Payment Cryptography.

Requisito 26: Deberá describir el registro para acceder a las claves, los componentes clave o el material relacionado que se utilice fuera del servicio. Los registros de todas las actividades de administración clave que su aplicación realiza con el servicio están disponibles en AWS CloudTrail.

Requisito 27: Deberá describir los procedimientos de respaldo de las llaves, los componentes clave o el material relacionado que se utilice fuera del servicio.

Requisito 28: Los procedimientos para toda la administración de claves mediante la API deben incluir el uso de funciones con permisos de administración de claves y aprobaciones para ejecutar scripts u otro código que gestione las claves. AWS CloudTrail los registros contienen todos los eventos de administración clave

Objetivo de control 7: El equipo utilizado para procesar los PIN y las claves se gestiona de forma segura.

Requisito 29: Sus requisitos de protección física y lógica para los HSM se cumplen mediante el uso de la criptografía de AWS pagos.

Requisito 30: Su aplicación será responsable de todos los requisitos de protección física y lógica de los dispositivos POI.

Requisito 31: La protección de los dispositivos criptográficos seguros (SCD) utilizados por AWS Payment Cryptography se evaluó como parte de la evaluación del PIN PCI del servicio. Deberá demostrar la protección de cualquier otro SCD que utilice su aplicación.

Requisito 32: El uso de los SCD utilizados por AWS Payment Cryptography se evaluó como parte de la evaluación del PIN PCI del servicio. Deberá demostrar el control de acceso y la protección de cualquier otro SCD que utilice su aplicación.

Requisito 33: Deberá describir las protecciones de cualquier PIN-processing equipo que esté bajo su control.

Uso del componente de descifrado de criptografía de AWS pagos en soluciones P2PE

Las soluciones PCI P2PE pueden utilizar el componente de descifrado de criptografía de pagos.AWS Esto se documenta en la sección Point-to-Point Cifrado PCI: requisitos de seguridad y procedimientos de prueba, soluciones P2PE y uso de terceros proveedores de componentes and/or P2PE: «Un proveedor de soluciones (o un comerciante como proveedor de soluciones) puede subcontratar ciertas funciones P2PE a proveedores de componentes P2PE que cotizan en PCI e informar el uso de los componentes P2PE listados en PCI en su informe de validación de P2PE (P-ROV)», que está disponible en el sitio web de PCI.

Al igual que con otros servicios y normas de conformidad de AWS, es su responsabilidad utilizar el servicio de forma segura, configurar el control de acceso y utilizar los parámetros de seguridad de acuerdo con los requisitos de PCI P2PE. La guía del usuario del componente de descifrado P2PE de criptografía de pagos de AWS, que está disponible en AWS Artifact, contiene instrucciones detalladas para integrar la criptografía de AWS pagos con su solución PCI P2PE y el informe anual del componente de descifrado, que es obligatorio para los informes de conformidad.

Gestión de identidad y acceso para AWS Payment Cryptography

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los recursos. AWS Los administradores de IAM controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar los recursos de criptografía de AWS pagos. La IAM es una herramienta Servicio de AWS que puede utilizar sin coste adicional.

Temas

- Público
- Autenticación con identidades
- Administración del acceso con políticas
- Cómo AWS La criptografía de pagos funciona con IAM
- AWS Ejemplos de políticas de criptografía de pagos basadas en la identidad
- Resource-based políticas para AWS Payment Cryptography
- Multi-party aprobación de criptografía AWS de pagos
- Resolución de problemas AWS Criptografía de pago: identidad y acceso

Público

La forma de usar AWS Identity and Access Management (IAM) varía según la función que desempeñes:

- Usuario del servicio: solicite permisos al administrador si no puede acceder a las características (consulte Resolución de problemas AWS Criptografía de pago: identidad y acceso).
- Administrador del servicio: determine el acceso de los usuarios y envíe las solicitudes de permiso (consulte Cómo AWS La criptografía de pagos funciona con IAM).
- Administrador de IAM: escribe las políticas para administrar el acceso (consulte AWS Ejemplos de políticas de criptografía de pagos basadas en la identidad).

Autenticación con identidades

La autenticación es la forma en que inicias sesión AWS con tus credenciales de identidad. Debe autenticarse como usuario de Usuario raíz de la cuenta de AWS IAM o asumir una función de IAM.

Puede iniciar sesión como una identidad federada con las credenciales de una fuente de identidad, como AWS IAM Identity Center (IAM Identity Center), la autenticación de inicio de sesión único o las credenciales. Google/Facebook Para obtener más información sobre el inicio de sesión, consulte Cómo iniciar sesión en la Cuenta de AWS en la Guía del usuario de AWS Sign-In .

Para el acceso programático, AWS proporciona un SDK y una CLI para firmar criptográficamente las solicitudes. Para obtener más información, consulte AWS Signature Version 4 para solicitudes de API en la Guía del usuario de IAM.

Cuenta de AWS usuario raíz

Al crear una Cuenta de AWS, se comienza con una identidad de inicio de sesión denominada usuario Cuenta de AWS raíz, que tiene acceso completo a todos los Servicios de AWS recursos. Se recomiendaencarecidamente que no utilice el usuario raíz para las tareas diarias. Para ver la lista completa de las tareas que requieren credenciales de usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Usuarios y grupos de IAM

Un usuario de IAM es una identidad con permisos específicos para una sola persona o aplicación. Recomendamos el uso de credenciales temporales en lugar de usuarios de IAM con credenciales de larga duración. Para obtener más información, consulte Exigir a los usuarios humanos que utilicen la federación con un proveedor de identidad para acceder AWS mediante credenciales temporales en la Guía del usuario de IAM.

Un grupo de IAM especifica un conjunto de usuarios de IAM y facilita la administración de los permisos para grupos grandes de usuarios. Para obtener más información, consulte Casos de uso para usuarios de IAM en la Guía del usuario de IAM.

Roles de IAM

Un Rol de IAM es una identidad con permisos específicos que proporciona credenciales temporales. Puede asumir un rol cambiando de un rol de usuario a uno de IAM (consola) o llamando a una AWS CLI operación de AWS API. Para obtener más información, consulte Métodos para asumir un rol en la Guía del usuario de IAM.

Los roles de IAM son útiles para el acceso de usuario federado, los permisos de usuario de IAM temporales, el acceso entre cuentas, el acceso entre servicios y las aplicaciones que se ejecutan en Amazon EC2. Para obtener más información, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Administración del acceso con políticas

AWS Para controlar el acceso, puede crear políticas y adjuntarlas a AWS identidades o recursos. Una política define los permisos cuando están asociados a una identidad o un recurso. AWS evalúa estas políticas cuando un director hace una solicitud. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre los documentos de políticas de JSON, consulte Información general de políticas de JSON en la Guía del usuario de IAM.

Mediante las políticas, los administradores especifican quién tiene acceso a qué, definiendo qué entidad principal puede realizar acciones sobre qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM crea políticas de IAM y las agrega a roles, que los usuarios pueden asumir posteriormente. Las políticas de IAM definen permisos independientemente del método que se utilice para realizar la operación.

Identity-based políticas

Identity-based las políticas son documentos de política de permisos de JSON que se adjuntan a una identidad (usuario, grupo o rol). Estas políticas controlan qué acciones pueden realizar las identidades, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en la identidad, consulte Definición de permisos de IAM personalizados con políticas administradas por el cliente en la Guía del usuario de IAM.

Identity-based las políticas pueden ser políticas integradas (integradas directamente en una sola identidad) o políticas administradas (políticas independientes asociadas a varias identidades). Para obtener información sobre cómo elegir entre políticas administradas e insertadas, consulte Selección entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Resource-based políticas

Resource-based las políticas son documentos de políticas de JSON que se adjuntan a un recurso. Los ejemplos incluyen las Políticas de confianza de roles de IAM y las Políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios

pueden utilizarlos para controlar el acceso a un recurso específico. Debe especificar una entidad principal en una política basada en recursos.

Resource-based las políticas son políticas en línea que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso (ACL)

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios que admiten las ACL. AWS WAF Para obtener más información sobre las ACL, consulte Información general de Lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales que pueden establecer los permisos máximos que conceden los tipos de políticas más comunes:

- Límites de permisos: establecen los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM. Para obtener más información, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicios (SCP): especifican los permisos máximos para una organización o unidad organizativa en AWS Organizations. Para obtener más información, consulte Políticas de control de servicios en la Guía del usuario de AWS Organizations .
- Políticas de control de recursos (RCP): definen los permisos máximos disponibles para los recursos de las cuentas. Para obtener más información, consulte Políticas de control de recursos (RCP) en la Guía del usuario de AWS Organizations .
- Políticas de sesión: políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal para un rol o un usuario federado. Para obtener más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud

cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del usuario de IAM.

Cómo AWS La criptografía de pagos funciona con IAM

Antes de utilizar IAM para gestionar el acceso a la criptografía de AWS pagos, debe saber qué funciones de IAM están disponibles para su uso con la criptografía de pagos. AWS Para obtener una visión general de cómo funcionan la criptografía de AWS pagos y otros AWS servicios con IAM, consulte AWS Servicios que funcionan con IAM en la Guía del usuario de IAM.

Temas

- AWS Políticas de criptografía de pagos Identity-based
- Autorización basada en AWS Etiquetas de criptografía de pago

AWS Políticas de criptografía de pagos Identity-based

Con las políticas de IAM basadas en la identidad, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. AWS La criptografía de pagos admite claves de condiciones, recursos y acciones específicas. Para obtener información sobre todos los elementos que utiliza en una política JSON, consulte Referencia de los elementos de las políticas JSON de IAM en la Guía del usuario de IAM.

Acciones

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Las acciones políticas en criptografía de AWS pagos utilizan el siguiente prefijo antes de la acción:. `payment-cryptography:` Por ejemplo, para conceder a alguien permiso para ejecutar una operación de la API de `VerifyCardData` de AWS Payment Cryptography, incluya la acción `payment-cryptography:VerifyCardData` en su política. Las instrucciones de la política deben incluir un elemento `Action` o un elemento `NotAction`. AWS La criptografía de pagos define su propio conjunto de acciones que describen las tareas que puede realizar con este servicio.

Para especificar varias acciones en una única instrucción, sepárelas con comas del siguiente modo:

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

Puede utilizar caracteres comodín para especificar varias acciones (*). Por ejemplo, para especificar todas las acciones que comiencen con la palabra `List` (como, por ejemplo, `ListKeys` y `ListAliases`), incluya la siguiente acción:

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

Para ver una lista de las acciones de criptografía de AWS pagos, consulte las acciones definidas por la criptografía de AWS pagos en la Guía del usuario de IAM.

Recursos

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). En el caso de las acciones que no admiten permisos por recurso, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

El recurso de clave de criptografía de pagos tiene el siguiente ARN:

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

Para obtener más información sobre el formato de los ARN, consulte Nombres de recursos de Amazon (ARN) y espacios de nombres de AWS servicio.

Por ejemplo, para especificar la instancia de `arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h` en su instrucción, utilice el siguiente ARN:

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

Para especificar todas las claves que pertenecen a una cuenta específica, utilice el carácter comodín (*):

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

Algunas acciones AWS de criptografía de pagos, como las de creación de claves, no se pueden realizar en un recurso específico. En dichos casos, debe utilizar el carácter comodín (*).

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

Para especificar varios recursos en una única instrucción, utilice una coma, tal y como se indica a continuación:

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

Ejemplos

Para ver ejemplos de políticas de criptografía de AWS pagos basadas en la identidad, consulte. AWS Ejemplos de políticas de criptografía de pagos basadas en la identidad

Autorización basada en AWS Etiquetas de criptografía de pago

Puede adjuntar etiquetas a los recursos AWS de criptografía de pagos o pasarlas en una solicitud a criptografía de AWS pagos. Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `payment-cryptography:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

AWS Ejemplos de políticas de criptografía de pagos basadas en la identidad

De forma predeterminada, los usuarios y roles de IAM no tienen permiso para crear o modificar recursos de criptografía AWS de pagos. Tampoco pueden realizar tareas con la API Consola de

administración de AWS AWS CLI, o AWS . Un administrador de IAM debe crear políticas de IAM que concedan permisos a los usuarios y a los roles para realizar operaciones de la API concretas en los recursos especificados que necesiten. El administrador debe adjuntar esas políticas a los usuarios o grupos de IAM que necesiten esos permisos.

Para obtener información acerca de cómo crear una política basada en identidad de IAM con estos documentos de políticas JSON de ejemplo, consulte Creación de políticas en la pestaña JSON en la Guía del usuario de IAM.

Temas

- Prácticas recomendadas relativas a políticas
- Uso de AWS Consola de criptografía de pagos
- Permitir a los usuarios consultar sus propios permisos
- Posibilidad de acceder a todos los aspectos de AWS Payment Cryptography
- Posibilidad de llamar a las API mediante claves específicas
- Capacidad para denegar específicamente un recurso

Prácticas recomendadas relativas a políticas

Identity-based las políticas determinan si alguien puede crear recursos de criptografía de AWS pagos de tu cuenta, acceder a ellos o eliminarlos. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de tarea en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte Políticas y permisos en IAM en la Guía del usuario de IAM.

- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo CloudFormation. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utiliza el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Validación de políticas con el Analizador de acceso de IAM en la Guía del usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte Acceso seguro a la API con MFA en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Uso de AWS Consola de criptografía de pagos

Para acceder a la consola de criptografía de AWS pagos, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los recursos de criptografía de AWS pagos de su AWS cuenta. Si crea una política basada en identidad que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles de IAM) que tengan esa política.

Para garantizar que esas entidades puedan seguir utilizando la consola de criptografía de AWS pagos, adjunte también la siguiente política de AWS gestión a las entidades. Para obtener más información, consulte Adición de permisos a un usuario en la Guía del usuario de IAM.

No es necesario que concedas permisos mínimos de consola a los usuarios que solo realicen llamadas a la API AWS CLI o a la AWS API. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que intenta realizar.

Permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API AWS CLI o 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": "*"
        }
    ]
}
```

Posibilidad de acceder a todos los aspectos de AWS Payment Cryptography

> **Warning**
>
> Este ejemplo proporciona permisos amplios y no se recomienda. En su lugar, considere los modelos de acceso con menos privilegios.

En este ejemplo, desea conceder a un usuario de IAM de su AWS cuenta acceso a todas sus claves de criptografía de AWS pagos y la posibilidad de llamar a todas las API de criptografía de AWS pagos, incluidas ambas `ControlPlane` y las operaciones. `DataPlane`

JSON

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

Posibilidad de llamar a las API mediante claves específicas

En este ejemplo, quiere conceder a un usuario de IAM de su AWS cuenta acceso a una de sus claves de criptografía de AWS pagos y, a continuación, utilizar este recurso en dos API, `arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h` y. `GenerateCardValidationData` `VerifyCardValidationData` Por el contrario, el usuario de IAM no tendrá acceso para usar esta clave en otras operaciones, como `DeleteKey` o `ExportKey`

Los recursos pueden ser claves con el prefijo `key` o alias con el prefijo `alias`.

JSON

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

Capacidad para denegar específicamente un recurso

> **Warning**
>
> Considere detenidamente las implicaciones de conceder un acceso comodín. En su lugar, considere un modelo de privilegio mínimo.

En este ejemplo, quiere permitir que un usuario de IAM de su AWS cuenta acceda a cualquiera de sus claves de criptografía de AWS pagos, pero quiere denegar los permisos a una clave específica. El usuario tendrá acceso a `VerifyCardData` y `GenerateCardData` con todas las claves a excepción de la especificada en la instrucción de denegación.

JSON

```
{
```

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

Resource-based políticas para AWS Payment Cryptography

Resource-based las políticas son documentos de política de JSON que se adjuntan a un recurso, como una clave de criptografía de AWS pagos. En una política basada en recursos, se especifica quién puede acceder a la clave y las acciones que puede realizar en ella. Puede utilizar políticas basadas en recursos para:

- Conceda acceso a una sola clave a varios usuarios y roles.
- Conceda acceso a los usuarios o roles de otras AWS cuentas.

Temas

- Consideraciones
- Administrar políticas basadas en recursos
- Resource-based ejemplos de políticas

Al adjuntar una política basada en recursos a una clave de criptografía de AWS pagos, la criptografía de AWS pagos utiliza la lógica de evaluación de políticas de IAM para determinar si un mandante determinado está autorizado a realizar la acción solicitada. Para habilitar el acceso entre cuentas, puede especificar una cuenta completa o entidades de IAM de otra cuenta como principal en una política basada en recursos. Cross-account el acceso requiere dos políticas:

1. Resource-based política (cuenta del propietario de la clave): el propietario de la clave utiliza `PutResourcePolicy` para conceder el acceso a la cuenta de la persona que llama o al director de IAM.
2. Identity-based política (cuenta de la persona que llama): el administrador de IAM de la persona que llama también debe permitir la acción de criptografía de AWS pagos (por ejemplo`payment-cryptography:EncryptData`) en la política de IAM de la persona que llama.

Ambas políticas deben permitir la acción. Si falta alguna de ellas, se deniega la solicitud entre cuentas. `AccessDeniedException`

Si una política basada en los recursos permite el acceso a un principal de la misma cuenta, no se requiere ninguna política adicional basada en la identidad. Para obtener más información, consulte en qué se diferencian las funciones de IAM de Resource-based las políticas en la Guía del usuario de IAM.

⚠ Política de recursos: operaciones del plano de control

Resource-based las políticas no se aplican a las operaciones del plano de control de la política de recursos `PutResourcePolicy`, como `GetResourcePolicy`, y `DeleteResourcePolicy`. Esto evita posibles escenarios de bloqueo en los que una política de recursos pueda denegar la posibilidad de modificar o eliminar la propia política. El acceso a estas operaciones del plano de control se rige únicamente por las políticas de IAM basadas en la identidad.

Consideraciones

Tenga en cuenta lo siguiente cuando utilice políticas basadas en recursos con la criptografía de pagos. AWS

- AWS La criptografía de pagos impide automáticamente el acceso público a las claves. No puede crear una política basada en recursos que conceda acceso a directores públicos o anónimos. Todo

acceso a las claves AWS de criptografía de pagos requiere AWS directores autenticados, y el acceso público siempre está bloqueado.

- Resource-based las políticas se aplican por clave. Cada clave AWS de criptografía de pagos puede tener como máximo una política basada en recursos adjunta.
- Resource-based las políticas no se aplican a los alias. Cuando se hace referencia a una clave por su alias, se evalúa la política de recursos adjunta a la clave subyacente.
- Resource-based Las políticas no se aplican a las claves de región de réplica de solo lectura creadas mediante Multi-Region la replicación de claves en este momento. Las políticas de recursos solo se pueden adjuntar a la clave de región principal.
- El `Resource` elemento de una política basada en recursos debe ser `"*"` o coincidir exactamente con el ARN de la clave a la que se adjunta la política. Se `"*"` recomienda su uso porque permite reutilizar el mismo documento de política en varias claves.
- Las API de administración de políticas de recursos (`PutResourcePolicy``GetResourcePolicy`, y`DeleteResourcePolicy`) están restringidas al Cuenta de AWS propietario de la clave. Solo los directores de la cuenta del propietario de la clave pueden administrar las políticas de recursos.

Administrar políticas basadas en recursos

Puede administrar las políticas basadas en recursos para las claves de criptografía AWS de pagos mediante la API o. AWS CLI AWS Para usar este comando, sustituya el *`italicized placeholder text`* comando del ejemplo por su propia información.

Adjunte una política basada en recursos

Utilice la acción de la `PutResourcePolicy`API o el comando put-resource-policyCLI para adjuntar una política basada en recursos a una clave. Si ya existe una política, el comando la reemplaza.

En el siguiente ejemplo, se adjunta una política basada en recursos de un archivo JSON a una clave.

```
aws payment-cryptography put-resource-policy \
    --resource-arn arn:aws:payment-cryptography:us-
east-2:111122223333:key/kwapwa6qaifllw2h \
    --policy file://policy.json
```

Recupera una política basada en recursos

Utilice la acción de la `GetResourcePolicy`API o el comando get-resource-policyCLI para recuperar la política basada en recursos adjunta a una clave.

En el siguiente ejemplo, se recupera la política basada en recursos adjunta a una clave.

```
aws payment-cryptography get-resource-policy \
    --resource-arn arn:aws:payment-cryptography:us-
east-2:111122223333:key/kwapwa6qaifllw2h
```

La respuesta devuelve el documento de política:

```
{
    "Policy": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {
                    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
                },
                "Action": [
                    "payment-cryptography:EncryptData",
                    "payment-cryptography:DecryptData"
                ],
                "Resource": "*"
            }
        ]
    }
}
```

Eliminar una política basada en recursos

Utilice la acción de la DeleteResourcePolicyAPI o el comando delete-resource-policyCLI para eliminar la política basada en recursos de una clave.

En el siguiente ejemplo, se elimina la política basada en recursos adjunta a una clave.

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

Resource-based ejemplos de políticas

Otorgue acceso multicuenta a una clave

La siguiente política basada en los recursos concede a un usuario de otra AWS cuenta el permiso para utilizar una clave de criptografía de AWS pagos en operaciones criptográficas.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
            },
            "Action": [
                "payment-cryptography:GenerateCardValidationData",
                "payment-cryptography:VerifyCardValidationData"
            ],
            "Resource": "*"
        }
    ]
}
```

Otorga diferentes permisos a diferentes cuentas

La siguiente política basada en los recursos demuestra cómo conceder distintos permisos a los directores de cuentas distintas. En este ejemplo, un servidor de control de acceso (ACS) de una cuenta puede generar datos de validación de tarjetas, mientras que un servicio de autorización de pago de una cuenta diferente solo puede validar criptogramas de 3DS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow3DSACSToGenerate",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/3dsAcsRole"
            },
            "Action": [
```

```
                "payment-cryptography:GenerateCardValidationData"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowPaymentAuthToVerify",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::444455556666:role/PaymentAuthRole"
            },
            "Action": [
                "payment-cryptography:VerifyAuthRequestCryptogram"
            ],
            "Resource": "*"
        }
    ]
}
```

Multi-party aprobación de criptografía AWS de pagos

AWS La criptografía de pagos se integra con Multi-party la aprobación (MPA), una capacidad de Amazon Web Services Organizations, para ayudar a proteger las operaciones críticas mediante un proceso de aprobación distribuido. Con la MPA, puede exigir que varias personas de confianza aprueben determinadas operaciones de criptografía de AWS pagos antes de llevarlas a cabo.

Temas

- Descripción general de
- Operaciones protegidas
- Requisitos previos
- Activación y desactivación de la MPA
- Introducción
- Ejemplo: importe un certificado raíz con la MPA habilitada
- AWS CloudTrail registro de eventos de MPA
- Comprobar el estado de las solicitudes y gestionar los errores

Descripción general de

Multi-party La aprobación añade un nivel adicional de seguridad a las operaciones confidenciales de criptografía de AWS pagos, ya que requiere la aprobación de un grupo de personas de confianza antes de que la operación pueda continuar. Esto ayuda a proteger contra los cambios no autorizados en caso de que un único conjunto de credenciales se vea comprometido e impide que una sola persona realice un cambio unilateral.

Un equipo de aprobación es un grupo de aprobadores de su organización que usted designa para aprobar o denegar las solicitudes de operaciones protegidas. Los aprobadores de su organización gestionan en su totalidad el proceso de aprobación. Ningún miembro AWS del personal participa en la aprobación o denegación de las solicitudes.

Cuando el MPA está activado para una operación protegida, ocurre lo siguiente:

1. Un solicitante inicia la operación protegida.
2. La MPA crea una sesión de aprobación y notifica a los miembros del equipo de aprobación.
3. Los miembros del equipo de aprobación revisan la solicitud y la aprueban o rechazan a través del portal de la MPA.
4. Una vez que se alcanza el umbral mínimo de aprobaciones requerido, la operación continúa. Si el equipo de aprobación rechaza la solicitud o si el tiempo de sesión permitido expira antes de alcanzar el umbral de aprobación, la operación no se lleva a cabo. En ambos casos, el solicitante debe enviar una nueva solicitud para volver a intentar la operación.

> **Note**
>
> Al importar un certificado de CA raíz con el MPA activado, el `RequesterComment` parámetro es obligatorio. Este comentario se incluye en la notificación de aprobación que se envía al equipo de aprobación y proporciona el contexto de la solicitud.

Operaciones protegidas

AWS La criptografía de pagos admite la MPA para las siguientes operaciones:

- `ImportKey`con material `RootCertificatePublicKey` clave: la importación de un certificado de clave pública raíz es una operación fundamental, ya que los certificados raíz establecen la

base de confianza para todas las importaciones y exportaciones de claves posteriores mediante un intercambio de claves asimétrico, como. TR-34 Exigir la aprobación de varias partes para esta operación ayuda a garantizar que ninguna persona pueda establecer o cambiar unilateralmente la raíz de confianza de sus claves de criptografía de AWS pagos.

Requisitos previos

Antes de poder utilizar la MPA con la criptografía AWS de pagos, debe cumplir los siguientes requisitos previos:

- Configure MPA en su entorno de Amazon Web Services Organizations. Para obtener instrucciones, consulte ¿Qué es Multi-party la aprobación? en la Guía del usuario de Multi-party aprobación.
- Cree al menos un equipo de aprobación con los aprobadores necesarios.
- Comparta el equipo de aprobación con el equipo Cuenta de AWS que contiene sus claves AWS de criptografía de pago. AWS Resource Access Manager
- La cuenta de administración de su organización debe estar habilitada para su Multi-party aprobación.

Activación y desactivación de la MPA

Una vez que haya creado un equipo de aprobación, podrá habilitar la MPA para la criptografía de AWS pagos asociando el equipo a su cuenta. También puede deshabilitar el MPA desasociando el equipo, aunque la desasociación requiere la aprobación del equipo de aprobación actualmente asociado.

Habilite el MPA

Utilice la acción de la `AssociateMpaTeam` API o el comando associate-mpa-team CLI para asociar un equipo de aprobación a su cuenta de criptografía de AWS pagos. Una vez asociadas, las operaciones protegidas requieren la aprobación del equipo para poder continuar.

```
aws payment-cryptography associate-mpa-team \
    --team-arn arn:aws:mpa:us-east-1:111122223333:team/my-approval-team
```

Desactivar el MPA

Utilice la acción de la `DisassociateMpaTeam` API o el comando disassociate-mpa-team CLI para eliminar la asociación del equipo de aprobación. La disociación de un equipo es en sí misma una operación protegida que requiere la aprobación del equipo de aprobación actualmente asociado.

```
aws payment-cryptography disassociate-mpa-team \
    --team-arn arn:aws:mpa:us-east-1:111122223333:team/my-approval-team
```

Important

La desactivación del MPA requiere la aprobación del equipo de aprobación actualmente asociado. Esto garantiza que ninguna persona pueda eliminar unilateralmente la protección de aprobación multipartidista.

Note

El `--requester-comment` parámetro es opcional para associate-mpa-team y. disassociate-mpa-team

Introducción

Para empezar a utilizar la MPA para la criptografía de AWS pagos, consulte la Guía del usuario de Multi-party aprobación para obtener instrucciones de configuración detalladas, que incluyen cómo crear equipos de aprobación, configurar las políticas de aprobación y gestionar las sesiones de aprobación.

Ejemplo: importe un certificado raíz con la MPA habilitada

Una vez que se habilita el MPA y se asocia un equipo de aprobación a la `ImportKey` operación`RootCertificatePublicKey`, la solicitud de importación debe aprobarse antes de continuar.

1. Un solicitante llama `import-key` para importar un certificado de clave pública raíz. Para usar este comando, sustituya *`italicized placeholder text`* el comando del ejemplo por su propia información.

   ```
   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": "LS0tLS1CRUdJTi..."}}' \
    --requester-comment "Importing new root CA certificate for TR-34 key exchange
 with partner XYZ"
```

La respuesta devuelve una clave `KeyState` establecida en`CREATE_IN_PROGRESS`, lo que indica que la solicitud está pendiente de aprobación. La respuesta también incluye `MpaStatus` detalles sobre la sesión de aprobación:

```
{
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyUsage": "TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE",
            "KeyClass": "PUBLIC_KEY",
            "KeyAlgorithm": "RSA_4096"
        },
        "Enabled": true,
        "KeyState": "CREATE_IN_PROGRESS",
        "KeyOrigin": "EXTERNAL",
        "CreateTimestamp": "2026-04-27T10:15:30.000000+00:00",
        "UsageStartTimestamp": "2026-04-27T10:15:29.926000+00:00",
        "MpaStatus": {
            "MpaSessionArn": "arn:aws:mpa:us-
east-1:111122223333:session/abc123def456",
            "Status": "PENDING",
            "InitiationDate": "2026-04-27T10:15:30.000000+00:00"
        }
    }
}
```

2. Como la MPA está habilitada, la solicitud no se completa de forma inmediata. En su lugar, AWS Payment Cryptography crea una sesión de aprobación y devuelve una respuesta que indica que la aprobación está pendiente.

3. Los miembros del equipo de aprobación reciben una notificación y revisan la solicitud a través del portal de la MPA. Una vez que el número requerido de aprobadores apruebe la solicitud, la operación de importación continúa y se importa el certificado raíz.

AWS CloudTrail registro de eventos de MPA

Cuando la MPA está habilitada, la criptografía de AWS pagos registra los eventos del servicio hasta que AWS CloudTrailfinaliza una sesión de aprobación. Estos eventos registran el resultado del proceso de aprobación, incluso si la solicitud se aprobó o no. Puede utilizar estos registros para auditar la actividad de la MPA y realizar un seguimiento del estado de las operaciones protegidas.

MPA-related CloudTrail los eventos incluyen los siguientes campos en`serviceEventDetails`:

- `keyArn`— El ARN de la clave afectada por la operación.
- `operation`— La operación protegida que se solicitó.
- `mpaSessionArn`— El ARN de la sesión de aprobación de la MPA.
- `sessionStatus`— El resultado de la sesión de aprobación (`APPROVEDoFAILED`).

Solicitud aprobada

El siguiente ejemplo muestra un CloudTrail evento para una `ImportKey` solicitud que fue aprobada por el equipo de la MPA:


```
{
    "eventVersion": "1.11",
    "eventTime": "2026-04-28T18:49:51Z",
    "eventName": "ImportKey",
    "eventSource": "payment-cryptography.amazonaws.com",
    "eventType": "AwsServiceEvent",
    "eventCategory": "Management",
    "awsRegion": "us-east-1",
    "readOnly": false,
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "payment-cryptography.amazonaws.com"
    },
    "resources": [
```

```
        {
            "ARN": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/spa2dclzmsihlj4o",
            "accountId": "111122223333",
            "type": "AWS::PaymentCryptography::Key"
        }
    ],
    "serviceEventDetails": {
        "keyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/spa2dclzmsihlj4o",
        "operation": "ImportKey",
        "mpaSessionArn": "arn:aws:mpa:us-east-1:111122223333:session/my-approval-
team/44c76e07-8937-4d7d-bb9a-a646322e2a1e",
        "sessionStatus": "APPROVED"
    }
}
```


Solicitud fallida

En el siguiente ejemplo, se muestra un CloudTrail evento de una `ImportKey` solicitud que se denegó o se agotó el tiempo de espera:


```
{
    "eventVersion": "1.11",
    "eventTime": "2026-04-28T18:50:35Z",
    "eventName": "ImportKey",
    "eventSource": "payment-cryptography.amazonaws.com",
    "eventType": "AwsServiceEvent",
    "eventCategory": "Management",
    "awsRegion": "us-east-1",
    "readOnly": false,
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "payment-cryptography.amazonaws.com"
    },
    "resources": [
        {
            "ARN": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/qj46ku4qimypxdo7",
            "accountId": "111122223333",
            "type": "AWS::PaymentCryptography::Key"
```

```
        }
    ],
    "serviceEventDetails": {
        "keyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/qj46ku4qimypxdo7",
        "operation": "ImportKey",
        "mpaSessionArn": "arn:aws:mpa:us-east-1:111122223333:session/my-approval-team/
b0ac1994-14e1-47a6-bf1a-0b6fc0b845f2",
        "sessionStatus": "FAILED"
    }
}
```

Para obtener más información AWS CloudTrail, consulte la Guía del AWS CloudTrail usuario.

Comprobar el estado de las solicitudes y gestionar los errores

Puede comprobar el estado de una solicitud de MPA pendiente llamando `GetKey`. La respuesta incluye el `MpaStatus` campo con los detalles de la sesión de aprobación actual. Para usar este comando, sustituya *`italicized placeholder text`* el comando del ejemplo por su propia información.

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

Mientras la solicitud esté pendiente de aprobación, la respuesta se mostrará `KeyState` como `CREATE_IN_PROGRESS` y `MpaStatus.Status` como`PENDING`:

```
{
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyUsage": "TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE",
            "KeyClass": "PUBLIC_KEY",
            "KeyAlgorithm": "RSA_4096"
        },
        "Enabled": true,
        "KeyState": "CREATE_IN_PROGRESS",
        "KeyOrigin": "EXTERNAL",
        "CreateTimestamp": "2026-04-27T10:15:30.000000+00:00",
        "UsageStartTimestamp": "2026-04-27T10:15:29.926000+00:00",
```

```
        "MpaStatus": {
            "MpaSessionArn": "arn:aws:mpa:us-east-1:111122223333:session/abc123def456",
            "Status": "PENDING",
            "InitiationDate": "2026-04-27T10:15:30.000000+00:00"
        }
    }
}
```

Una vez que el número requerido de aprobadores apruebe la solicitud, `KeyState` se mueve a `CREATE_COMPLETE` y `MpaStatus.Status` a`APPROVED`. La clave ya está lista para su uso:

```
{
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyUsage": "TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE",
            "KeyClass": "PUBLIC_KEY",
            "KeyAlgorithm": "RSA_4096"
        },
        "Enabled": true,
        "KeyState": "CREATE_COMPLETE",
        "KeyOrigin": "EXTERNAL",
        "CreateTimestamp": "2026-04-27T10:15:30.000000+00:00",
        "UsageStartTimestamp": "2026-04-27T10:15:29.926000+00:00",
        "MpaStatus": {
            "MpaSessionArn": "arn:aws:mpa:us-east-1:111122223333:session/abc123def456",
            "Status": "APPROVED",
            "InitiationDate": "2026-04-27T10:15:30.000000+00:00"
        }
    }
}
```

Si el equipo de aprobación rechaza la solicitud o la sesión caduca antes de que se alcance el umbral de aprobación, `KeyState` cambia `CREATE_FAILED` y `MpaStatus.Status` cambia a`FAILED`:

```
{
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyUsage": "TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE",
```

```
            "KeyClass": "PUBLIC_KEY",
            "KeyAlgorithm": "RSA_4096"
        },
        "Enabled": true,
        "KeyState": "CREATE_FAILED",
        "KeyOrigin": "EXTERNAL",
        "CreateTimestamp": "2026-04-27T10:15:30.000000+00:00",
        "UsageStartTimestamp": "2026-04-27T10:15:29.926000+00:00",
        "MpaStatus": {
            "MpaSessionArn": "arn:aws:mpa:us-east-1:111122223333:session/abc123def456",
            "Status": "FAILED",
            "InitiationDate": "2026-04-27T10:15:30.000000+00:00",
            "StatusMessage": "Approval session expired or was denied"
        }
    }
}
```


No se puede usar una clave en `CREATE_FAILED` estado para operaciones criptográficas. Para volver a intentar la importación, debe enviar una nueva `ImportKey` solicitud, que creará una nueva sesión de aprobación.

Resolución de problemas AWS Criptografía de pago: identidad y acceso

Se agregarán temas a esta sección a medida que se IAM-related identifiquen problemas específicos de la criptografía de AWS pagos. Para obtener información general sobre la solución de problemas relacionados con la IAM, consulte la sección de solución de problemas de la Guía del usuario de IAM.

Supervisión de la criptografía AWS de pagos

La supervisión es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de la criptografía de AWS pagos y del resto de las soluciones de AWS. AWS proporciona las siguientes herramientas de supervisión para controlar la criptografía de AWS pagos, informar cuando algo va mal y tomar medidas automáticas cuando sea necesario:

- Amazon CloudWatch monitorea tus AWS recursos y las aplicaciones en las que AWS ejecutas en tiempo real. Puede recopilar métricas y realizar un seguimiento de las métricas, crear paneles personalizados y definir alarmas que le advierten o que toman medidas cuando una métrica determinada alcanza el umbral que se especifique. Por ejemplo, puedes CloudWatch hacer un seguimiento del uso de algunos APIs o notificarte si te estás acercando a tus cuotas de criptografía de AWS pagos. Para obtener más información, consulta la Guía del CloudWatch usuario de Amazon.
- Amazon CloudWatch Logs le permite supervisar, almacenar y acceder a sus archivos de registro desde EC2 instancias de Amazon y otras fuentes. CloudTrail CloudWatch Los registros pueden monitorear la información de los archivos de registro y notificarle cuando se alcanzan ciertos umbrales. También se pueden archivar los datos del registro en un almacenamiento de larga duración. Para obtener más información, consulta la Guía del usuario CloudWatch de Amazon Logs.
- AWS CloudTrailcaptura las llamadas a la API y los eventos relacionados realizados por su AWS cuenta o en su nombre y entrega los archivos de registro a un bucket de Amazon S3 que especifique. Puede identificar qué usuarios y cuentas llamaron AWS, el punto final al que se llamó, los recursos (claves) utilizados, la dirección IP de origen desde la que se realizaron las llamadas y cuándo se produjeron las llamadas. Para obtener más información, consulte la AWS CloudTrail Guía del usuario de .

Temas

- Registro de llamadas a la API de criptografía de AWS pagos mediante AWS CloudTrail

Registro de llamadas a la API de criptografía de AWS pagos mediante AWS CloudTrail

AWS La criptografía de pagos está integrada con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en criptografía de AWS pagos. CloudTrail captura todas las llamadas a la API para la criptografía AWS de pagos como eventos. Las llamadas capturadas incluyen las llamadas desde la consola de y las llamadas desde el código a las operaciones de la API de . Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos de criptografía de AWS pagos. Si no configura un registro, podrá ver los eventos de administración (plano de control) más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada por usted CloudTrail, puede determinar la solicitud que se realizó a AWS Payment Cryptography, la dirección IP desde la que se realizó la solicitud, quién la hizo, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, consulte la Guía del AWS CloudTrail usuario.

Temas

- AWS Información sobre criptografía de pagos en CloudTrail
- Controle los eventos del plano en CloudTrail
- Eventos de datos en CloudTrail
- Comprensión AWS de las entradas de los archivos de registro del plano de control de criptografía de pagos
- Descripción de las entradas de los archivos de registro del plano de datos de criptografía de AWS pagos

AWS Información sobre criptografía de pagos en CloudTrail

CloudTrail está habilitada en su AWS cuenta al crear la cuenta. Cuando se produce una actividad en la criptografía de AWS pagos, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Se puede ver, buscar y descargar los últimos eventos de la cuenta de AWS . Para obtener más información, consulte Visualización de eventos con el historial de CloudTrail eventos.

Para tener un registro continuo de los eventos de tu AWS cuenta, incluidos los relacionados con la criptografía de AWS pagos, crea un registro. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De manera predeterminada, cuando se crea un registro de

seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones de AWS . La ruta registra los eventos de todas las regiones de la AWS partición y envía los archivos de registro al bucket de Amazon S3 que especifique. Además, puede configurar otros AWS servicios para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento
- CloudTrail servicios e integraciones compatibles
- Configuración de las notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de CloudTrail registro de varias regiones
- Recibir archivos de CloudTrail registro de varias cuentas

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario le ayuda a determinar lo siguiente:

- Si la solicitud se realizó con credenciales de usuario root o AWS Identity and Access Management (IAM).
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro AWS servicio.

Para obtener más información, consulte el elemento userIdentity de CloudTrail .

Controle los eventos del plano en CloudTrail

CloudTrail registra las operaciones de criptografía de AWS pagos, como CreateKey,, ImportKeyDeleteKeyListKeysTagResource, y todas las demás operaciones del plano de control.

Eventos de datos en CloudTrail

Los eventos de datos proporcionan información sobre las operaciones de recursos que se realizan en un recurso o dentro de él, como el cifrado de una carga útil o la traducción de un PIN. Los eventos de datos son actividades de gran volumen que CloudTrail no se registran de forma predeterminada. Puede habilitar el registro de acciones de la API de eventos de datos para los eventos del plano de datos de criptografía de AWS pagos mediante nuestra consola CloudTrail APIs . Para obtener más información, consulte Registro de eventos de datos en la Guía del usuario de AWS CloudTrail .

Con CloudTrail, debe utilizar selectores de eventos avanzados para decidir qué actividades de la API de criptografía de AWS pagos se registran y registran. Para registrar los eventos del plano de datos de criptografía de AWS pagos, debes incluir el tipo de recurso y. `AWS Payment Cryptography key` `AWS Payment Cryptography alias` Una vez establecido esto, puede ajustar aún más las preferencias de registro seleccionando eventos de datos específicos para registrarlos, por ejemplo, mediante el uso del filtro `eventName` para realizar un seguimiento de los eventos de `EncryptData`. Para obtener más información, consulta `AdvancedEventSelector` en la AWS CloudTrail Referencia de la API de .

> **Note**
>
> Para suscribirse a los eventos de datos de criptografía de AWS pagos, debe utilizar selectores de eventos avanzados. Recomendamos suscribirse a los eventos clave y de alias para asegurarse de recibir todos los eventos.

AWS Eventos de datos de criptografía de pagos:

- `DecryptData`
- `EncryptData`
- `GenerateCardValidationData`
- `GenerateMac`
- `GeneratePinData`
- `ReEncryptData`
- `TranslatePinData`
- `VerifyAuthRequestCryptogram`
- `VerifyCardValidationData`
- `VerifyMac`
- `VerifyPinData`

Se aplican cargos adicionales a los eventos de datos. Para obtener más información, consulte AWS CloudTrail Precios.

Comprensión AWS de las entradas de los archivos de registro del plano de control de criptografía de pagos

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que usted especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una solicitud única de cualquier fuente e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un registro ordenado de las llamadas a la API pública, por lo que no aparecen en ningún orden específico.

En el siguiente ejemplo, se muestra una entrada de CloudTrail registro que muestra la `CreateKey` acción AWS de criptografía de pagos.

```
      {
    CloudTrailEvent: {
      tlsDetails= {
        TlsDetails: {
          cipherSuite=TLS_AES_128_GCM_SHA256,
          tlsVersion=TLSv1.3,
          clientProvidedHostHeader=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/
ControlPlane-IntegTest-68211a2a-3e9d-42b7-86ac-c682520e0410,
            invokedBy=null,
            accessKeyId=TESTXECZ5U2ZULLHHMJG,
            type=AssumedRole,
            sessionContext={
```

```
              SessionContext: {
                sessionIssuer={
                  SessionIssuer: {arn=arn:aws:iam::111122223333:role/TestAssumeRole-us-
west-2,
                  type=Role,
                  accountId=111122223333,
                  userName=TestAssumeRole-us-west-2,
                  principalId=TESTXECZ5U9M4LGF2N6Y5}
                },
                attributes={
                  SessionContextAttributes: {
                    creationDate=Sun May 21 18:58:31 UTC 2023,
                    mfaAuthenticated=false
                  }
                },
                webIdFederationData=null
              }
            },
          username=null,
          principalId=TESTXECZ5U9M4LGF2N6Y5: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={
    }
  }
}
```

El siguiente ejemplo muestra una entrada de CloudTrail registro que demuestra que la criptografía de AWS pagos permite la replicación de claves multirregionales.

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "payment-cryptography.amazonaws.com"
    },
    "eventTime": "2025-08-15T17:50:41Z",
    "eventSource": "payment-cryptography.amazonaws.com",
    "eventName": "SynchronizeMultiRegionKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "payment-cryptography.amazonaws.com",
    "userAgent": "payment-cryptography.amazonaws.com",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "55c0fcbc-5b2e-4bd2-a976-99305be6e6fc",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "keyArn": "arn:aws:payment-cryptography:us-east-1:111122223333:key/key-id",
        "replicationRegion": "us-east-2"
    },
    "eventCategory": "Management"
}
```


Descripción de las entradas de los archivos de registro del plano de datos de criptografía de AWS pagos

Los eventos del plano de datos se pueden configurar opcionalmente y funcionan de manera similar a los registros del plano de control, pero suelen tener volúmenes mucho más altos. Dado el carácter confidencial de algunas entradas y salidas de las operaciones del plano de datos de criptografía de AWS pagos, es posible que en algunos campos aparezca el mensaje «*** Datos confidenciales redactados ***». Esto no se puede configurar y su objetivo es evitar que los datos confidenciales aparezcan en los registros o registros.

El siguiente ejemplo muestra una entrada de CloudTrail registro que muestra la `EncryptData` acción de criptografía de AWS pagos.


```
    {
```

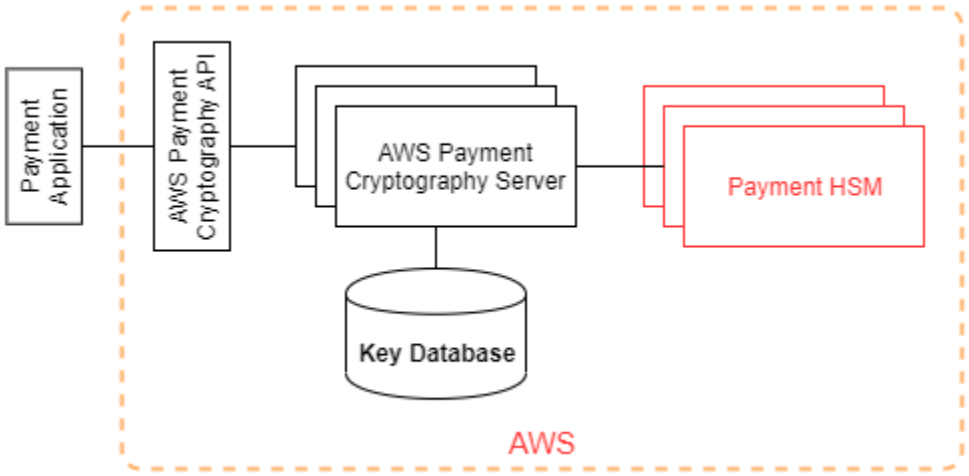
```
        "Records": [
            {
                "eventVersion": "1.09",
                "userIdentity": {
                    "type": "AssumedRole",
                    "principalId": "TESTXECZ5U2ZULLHHMJG:DataPlane-User",
                    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/DataPlane-
User",
                    "accountId": "111122223333",
                    "accessKeyId": "TESTXECZ5U2ZULLHHMJG",
                    "userName": "",
                    "sessionContext": {
                        "sessionIssuer": {
                            "type": "Role",
                            "principalId": "TESTXECZ5U9M4LGF2N6Y5",
                            "arn": "arn:aws:iam::111122223333:role/Admin",
                            "accountId": "111122223333",
                            "userName": "Admin"
                        },
                        "attributes": {
                            "creationDate": "2024-07-09T14:23:05Z",
                            "mfaAuthenticated": "false"
                        }
                    }
                },
                "eventTime": "2024-07-09T14:24:02Z",
                "eventSource": "payment-cryptography.amazonaws.com",
                "eventName": "GenerateCardValidationData",
                "awsRegion": "us-east-2",
                "sourceIPAddress": "192.158.1.38",
                "userAgent": "aws-cli/2.17.6 md/awscrt#0.20.11 ua/2.0 os/macos#23.4.0
 md/arch#x86_64 lang/python#3.11.8 md/pyimpl#CPython cfg/retry-mode#standard md/
installer#exe md/prompt#off md/command#payment-cryptography-data.generate-card-
validation-data",
                "requestParameters": {
                    "key_identifier": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/5rplquuwozodpwsp",
                    "primary_account_number": "*** Sensitive Data Redacted ***",
                    "generation_attributes": {
                        "CardVerificationValue2": {
                            "card_expiry_date": "*** Sensitive Data Redacted ***"
                        }
                    }
                },
```

```
                "responseElements": null,
                "requestID": "f2a99da8-91e2-47a9-b9d2-1706e733991e",
                "eventID": "e4eb3785-ac6a-4589-97a1-babdd3d4dd95",
                "readOnly": true,
                "resources": [
                    {
                        "accountId": "111122223333",
                        "type": "AWS::PaymentCryptography::Key",
                        "ARN": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/5rplquuwozodpwsp"
                    }
                ],
                "eventType": "AwsApiCall",
                "managementEvent": false,
                "recipientAccountId": "111122223333",
                "eventCategory": "Data",
                "tlsDetails": {
                    "tlsVersion": "TLSv1.3",
                    "cipherSuite": "TLS_AES_128_GCM_SHA256",
                    "clientProvidedHostHeader": "dataplane.payment-cryptography.us-
east-2.amazonaws.com"
                }
            }
        ]
    }
```

Detalles criptográficos

AWS La criptografía de pagos proporciona una interfaz web para generar y administrar claves criptográficas para las transacciones de pago. AWS La criptografía de pagos ofrece servicios estándar de administración de claves y herramientas y criptografía de transacciones de pago que puede utilizar para la administración y la auditoría centralizadas. Esta documentación proporciona una descripción detallada de las operaciones criptográficas que puede utilizar en la criptografía de AWS pagos para ayudarle a evaluar las funciones que ofrece el servicio.

AWS La criptografía de pagos contiene varias interfaces (incluida una API RESTful, a través de la CLI de AWS, el SDK de AWS y el Consola de administración de AWS) para solicitar operaciones criptográficas de una flota distribuida de módulos de seguridad de hardware PCI PTS HSM-validated.



AWS La criptografía de pagos es un servicio escalonado que consta de hosts de criptografía de AWS pagos con acceso a la web y un nivel de HSM. La agrupación de estos servidores escalonados forma la pila de criptografía de pagos. AWS Todas las solicitudes de criptografía de AWS pagos deben realizarse mediante el protocolo de seguridad de la capa de transporte (TLS) y finalizarse en un servidor de criptografía de pagos. AWS Los hosts del servicio solo permiten el uso de TLS con un conjunto de cifrado que proporciona una confidencialidad total. El servicio autentica y autoriza sus solicitudes mediante los mismos mecanismos de credenciales y políticas de IAM que están disponibles para todas las demás operaciones de la API. AWS

AWS Los servidores de criptografía de pagos se conectan al HSM subyacente a través de una red privada no virtual. Las conexiones entre los componentes del servicio y el HSM están protegidas con TLS mutuos (mTLS) para la autenticación y el cifrado.

Temas

- Objetivos de diseño
- Principios básicos
- Operaciones internas
- Operaciones de clientes

Objetivos de diseño

AWS La criptografía de pagos está diseñada para cumplir los siguientes requisitos:

- Confiable: el uso de las claves está protegido por las políticas de control de acceso que usted define y administra. No existe ningún mecanismo para exportar claves de criptografía de AWS pagos en texto plano. La confidencialidad de las claves criptográficas es crucial. Se requieren varios empleados de Amazon con acceso específico a los controles de acceso basados en quórums para realizar acciones administrativas en el. HSMs Ningún empleado de Amazon tiene acceso a las claves principales (o maestras) ni a las copias de seguridad de HSM. No se pueden sincronizar las claves principales HSMs que no formen parte de una región de criptografía de pagos. AWS Todas las demás claves están protegidas por las claves principales de HSM. Por lo tanto, las claves AWS de criptografía de pago de los clientes no se pueden utilizar fuera del servicio de criptografía de AWS pagos que opera en la cuenta del cliente.
- Baja latencia y alto rendimiento: la criptografía de AWS pagos proporciona operaciones criptográficas a un nivel de latencia y rendimiento adecuado para gestionar las claves criptográficas de pago y procesar las transacciones de pago.
- Durabilidad: la durabilidad de las claves criptográficas está diseñada para igualar a la de los servicios de mayor durabilidad en AWS. Una única clave criptográfica puede compartirse con un terminal de pago, una tarjeta con chip EMV u otro dispositivo criptográfico seguro (DSC) que se utilice durante muchos años.
- Regiones independientes: AWS proporciona regiones independientes para los clientes que necesiten restringir el acceso a los datos en diferentes regiones o necesiten cumplir requisitos de residencia de datos. El uso de claves se puede aislar dentro de una región de AWS.
- Fuente segura de números aleatorios: dado que una criptografía sólida depende de una generación de números aleatorios realmente impredecible, la criptografía de AWS pagos proporciona una fuente de números aleatorios validada y de alta calidad. Toda la generación de claves para la criptografía AWS de pagos utiliza un HSM PCI PTS que cotiza en el HSM y funciona en modo PCI.

- Auditoría: la criptografía de AWS pagos registra el uso y la gestión de las claves criptográficas en CloudTrail los registros y registros de servicio disponibles en Amazon. CloudWatch Puedes usar CloudTrail los registros para inspeccionar el uso de tus claves criptográficas, incluido el uso de las claves por parte de las cuentas con las que has compartido claves. AWS La criptografía de pagos es auditada por asesores externos según las normas PCI aplicables, la marca de la tarjeta y la seguridad de los pagos regionales. Las Certificaciones y las guías de Responsabilidad compartida están disponibles en AWS Artifact.
- Elastic: la criptografía de AWS pagos se amplía y amplía según su demanda. En lugar de predecir y reservar la capacidad de HSM, Payment Cryptography ofrece criptografía AWS de pagos bajo demanda. AWS La criptografía de pagos asume la responsabilidad de mantener la seguridad y la conformidad del HSM a fin de proporcionar la capacidad suficiente para satisfacer los picos de demanda de los clientes.

Principios básicos

Los temas de este capítulo describen las primitivas criptográficas de la criptografía de AWS pagos y dónde se utilizan. También presentan los elementos básicos del servicio.

Temas

- Primitivas criptográficas
- Entropía y generación de números aleatorios
- Operaciones de clave simétrica
- Operaciones con claves asimétricas
- Almacenamiento de claves
- Importación de claves simétricas
- Importación de claves con claves asimétricas
- Exportación de claves
- Protocolo de clave única derivada por transacción (DUKPT)
- Jerarquía de claves

Primitivas criptográficas

AWS La criptografía de pagos utiliza algoritmos criptográficos estándar parametrizables para que las aplicaciones puedan implementar los algoritmos necesarios para su caso de uso. El conjunto

de algoritmos criptográficos está definido por las normas PCI, ANSI X9, EMVco e ISO. Toda la criptografía se realiza mediante HSM normalizados de PCI PTS HSM que funcionan en modo PCI.

Entropía y generación de números aleatorios

AWS La generación de claves de criptografía de pagos se realiza en los HSM de criptografía de pagos. AWS Los HSM implementan un generador de números al azar que cumple con el requisito PCI PTS HSM para todos los tipos y parámetros de clave compatibles.

Operaciones de clave simétrica

Se admiten los algoritmos de clave simétrica y los puntos fuertes clave definidos en el ANSI X9 TR 31, el ANSI y el anexo C del PCI X9.24 PIN:

- Funciones hash: algoritmos de las familias SHA2 y SHA3 con un tamaño de salida superior a 2551. Excepto para la retrocompatibilidad con terminales POI v3 anteriores al PTS PCI.
- Cifrado y descifrado: AES con tamaño de clave mayor o igual a 128 bits, o TDEA con tamaño de clave mayor o igual a 112 bits (2 claves o 3 claves).
- Códigos de autenticación de mensajes (MAC) CMAC o GMAC con AES, así como HMAC con una función hash aprobada y un tamaño de clave mayor o igual a 128.

AWS La criptografía de pagos utiliza el AES 256 para las claves principales del HSM, las claves de protección de datos y las claves de sesión TLS.

Nota: Algunas de las funciones enumeradas se utilizan internamente para admitir protocolos y estructuras de datos estándar. Consulte la documentación de la API para ver los algoritmos compatibles con acciones específicas.

Operaciones con claves asimétricas

Se admiten los algoritmos clave asimétricos y los puntos fuertes clave definidos en el ANSI X9 TR 31, el ANSI X9.24 y el PCI PIN anexo C:

- Esquemas de establecimiento de claves aprobados, tal y como se describe en el NIST SP800-56A (acuerdo ECC/FCC2-based clave), el NIST SP800-56B (acuerdo clave) y el NIST (IFC-based clave). SP800-38F AES-based encryption/wrapping

AWS Los servidores de criptografía de pago solo permiten las conexiones al servicio mediante TLS con un conjunto de cifrado que proporciona un secreto total.

Nota: Algunas de las funciones enumeradas se utilizan internamente para admitir protocolos y estructuras de datos estándar. Consulte la documentación de la API para ver los algoritmos compatibles con acciones específicas.

Almacenamiento de claves

AWS Las claves de criptografía de pago están protegidas por las claves principales del HSM AES 256 y se almacenan en bloques de claves ANSI X9 TR 31 en una base de datos cifrada. La base de datos se replica en la base de datos en memoria de los servidores de criptografía de pagos. AWS

Según el Anexo C de la Normativa de seguridad PIN de la PCI, las claves AES 256 son igual o más fuertes que:

- TDEA de 3 claves
- RSA de 15360 bits
- ECC de 512 bits
- DSA, DH y MQV 15360/512

Importación de claves simétricas

AWS La criptografía de pagos permite la importación de criptogramas y bloques de claves con claves simétricas o públicas con una clave de cifrado de clave simétrica (KEK) igual o más segura que la clave protegida para la importación.

Importación de claves con claves asimétricas

AWS La criptografía de pagos permite la importación de criptogramas y bloques de claves con claves simétricas o públicas protegidas por una clave de cifrado de clave privada (KEK) igual o más segura que la clave protegida para la importación. La clave pública proporcionada para el descifrado debe tener su autenticidad e integridad garantizadas por un certificado de una autoridad de confianza del cliente.

Las KEK públicas que proporciona AWS Payment Cryptography cuentan con la protección de autenticación e integridad de una autoridad de certificación (CA) que certifica el cumplimiento de las normas PCI PIN Security y del anexo A de la PCI P2PE.

Exportación de claves

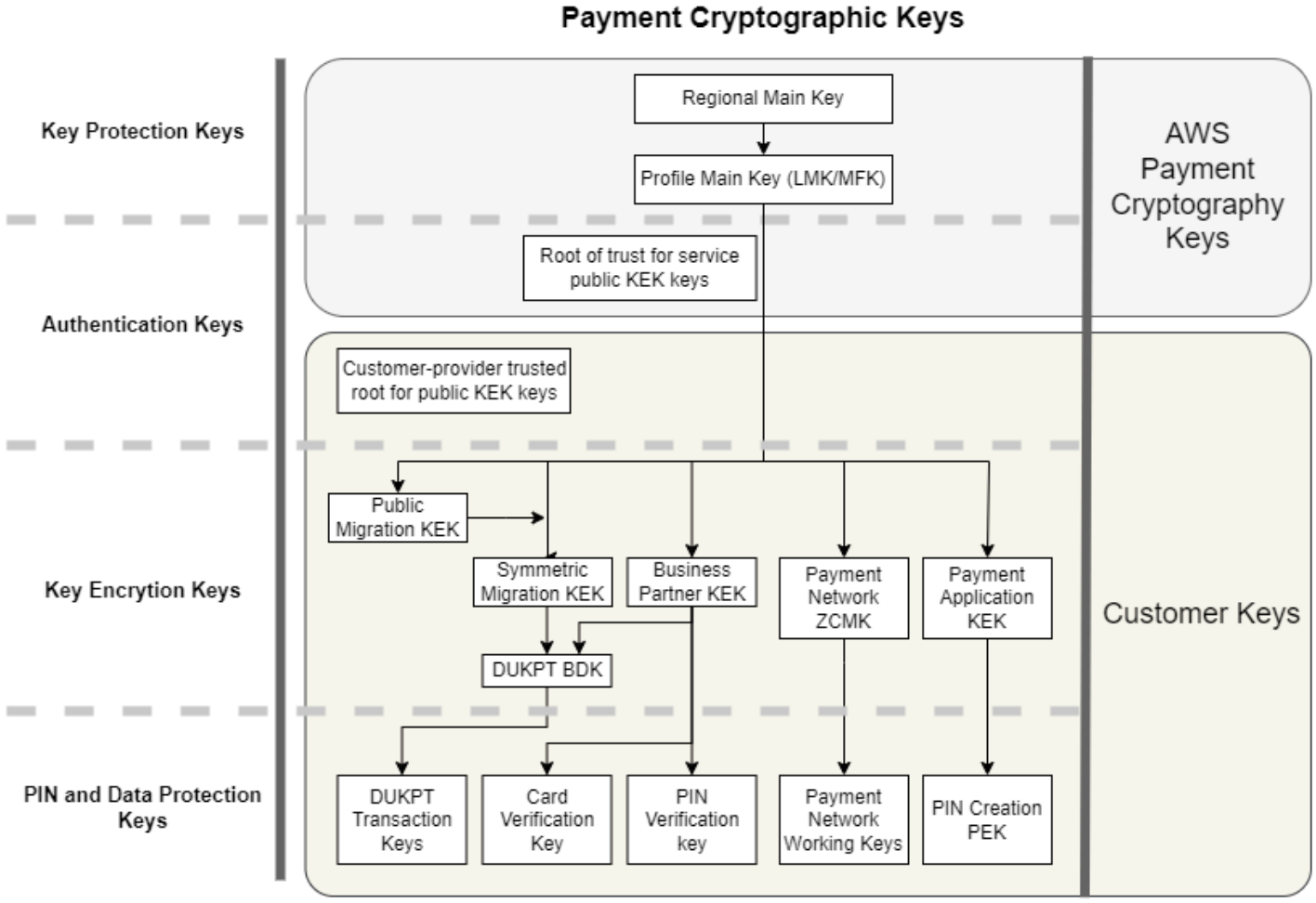
Las claves se pueden exportar y proteger con las claves adecuadas KeyUsage y que sean igual o más seguras que la clave que se va a exportar.

Protocolo de clave única derivada por transacción (DUKPT)

AWS La criptografía de pagos es compatible con las claves de derivación base (BDK) TDEA y AES, tal como se describe en la ANSI. X9.24-3

Jerarquía de claves

La jerarquía de claves de criptografía de AWS pagos garantiza que las claves estén siempre protegidas por claves tan sólidas o más seguras que las claves que protegen.



AWS Las claves de criptografía de pago se utilizan para la protección de las claves dentro del servicio:

Key	Description (Descripción)
Clave principal regional	Protege las imágenes virtuales del HSM, o perfiles, utilizadas para el procesamiento criptográfico. Esta clave sólo existe en el HSM y en las copias de seguridad.
Clave principal de perfil	Clave de protección de claves de cliente de nivel superior, tradicionalmente denominada clave maestra local (LMK) o clave maestra de archivo (MFK) para las claves de cliente. Esta clave sólo existe en el HSM y en las copias de seguridad. Los perfiles definen distintas configuraciones del HSM según lo requieran las normas de seguridad para los casos de uso de pagos.
Raíz de confianza para las claves de cifrado de clave pública (KEK) de criptografía de AWS pagos	La clave pública raíz de confianza y el certifica do para autenticar y validar las claves públicas proporcionados por AWS Payment Cryptogra phy para la importación y exportación de claves mediante claves asimétricas.

Las claves de cliente se agrupan por claves utilizadas para proteger otras claves y claves que protegen datos relacionados con el pago. Estos son ejemplos de claves de cliente de ambos tipos:

Key	Description (Descripción)
Customer-provided raíz de confianza para las claves KEK públicas	Clave pública y certificado proporcionados por usted como raíz de confianza para autenticar y validar las claves públicas que usted suministr a para la importación y exportación de claves utilizando claves asimétricas.
Claves de cifrado de claves (KEK)	Las KEK se utilizan únicamente para cifrar otras claves para su intercambio entre almacenes de claves externos y AWS Payment

Key	Description (Descripción)
	Cryptography, socios comerciales, redes de pago o diferentes aplicaciones de su organización.
Clave derivada única por transacción (DUKPT) clave derivada base (BDK)	Las BDK se utilizan para crear claves únicas para cada terminal de pago y traducir las transacciones de múltiples terminales a una única clave de trabajo del banco adquirente, o adquirente. La mejor práctica, que exige el Point-to-Point cifrado PCI (P2PE), es utilizar diferentes BDK para distintos modelos de terminales, servicios de inyección o inicialización de claves u otros tipos de segmentación para limitar el impacto de comprometer un BDK.
Clave maestra de control de zona de red de pagos (ZCMK)	Las ZCMK, también denominadas claves de zona o claves maestras de zona, son proporcionadas por las redes de pago para establecer claves de trabajo iniciales.
Claves de transacción DUKPT	Los terminales de pago configurados para DUKPT obtienen una clave única para el terminal y la transacción. El HSM que recibe la transacción puede determinar la clave a partir del identificador del terminal y del número de secuencia de la transacción.

Key	Description (Descripción)
Claves de preparación de datos de tarjetas	Las claves maestras del emisor EMV, las claves y valores de verificación de la tarjeta EMV y las claves de protección del archivo de datos de personalización de la tarjeta se utilizan para crear datos de tarjetas individua les para su uso por un proveedor de personali zación de tarjetas. Estas claves y los datos criptográficos de validación también son utilizados por los bancos emisores, o emisores, para autenticar los datos de las tarjetas como parte de la autorización de las transacciones.
Claves de preparación de datos de tarjetas	Las claves maestras del emisor EMV, las claves y valores de verificación de la tarjeta EMV y las claves de protección del archivo de datos de personalización de la tarjeta se utilizan para crear datos de tarjetas individua les para su uso por un proveedor de personali zación de tarjetas. Estas claves y los datos criptográficos de validación también son utilizados por los bancos emisores, o emisores, para autenticar los datos de las tarjetas como parte de la autorización de las transacciones.
Claves de trabajo de la red de pago	A menudo denominadas clave de trabajo del emisor o clave de trabajo del adquirente, son las claves que cifran las transacciones enviadas a las redes de pago o recibidas de ellas. Estas claves son rotadas con frecuencia por la red, a menudo diariamente o cada hora. Se trata de claves de cifrado con PIN (PEK) para las transacciones. PIN/Debit

Key	Description (Descripción)
Claves de cifrado (PEK) del número de identific ación personal (PIN)	Las aplicaciones que crean o descifran bloques de PIN utilizan PEK para evitar el almacenam iento o la transmisión del PIN en texto claro.

Operaciones internas

Este tema describe los requisitos internos implementados por el servicio para asegurar las claves de los clientes y las operaciones criptográficas para un servicio de criptografía de pagos y gestión de claves distribuido y escalable a nivel mundial.

Temas

- Protección HSM
- Administración general de claves
- Gestión de las claves de los clientes
- Seguridad de las comunicaciones
- Registro y supervisión

Protección HSM

Especificaciones y ciclo de vida del HSM{

AWS La criptografía de pagos utiliza una flota de productos disponibles en el mercado. HSMs HSMs Están validados por el FIPS 140-2 de nivel 3 y también utilizan versiones de firmware y la política de seguridad que figuran en la lista de dispositivos PCI PTS aprobada por el Consejo de Normas de Seguridad PCI como compatibles con PCI HSM v3. La norma PCI PTS HSM incluye requisitos adicionales para la fabricación, el envío, la implementación, la gestión y la destrucción del hardware HSM que son importantes para la seguridad y el cumplimiento de los pagos pero que no están contemplados en FIPS 140.

Los asesores externos comprueban la marca, el modelo, el firmware, la configuración, la gestión física del ciclo de vida, el control de cambios, los controles de acceso de los operadores, la gestión de las claves principales y todos los requisitos de PCI PIN y P2PE relacionados con las operaciones del HSM. HSMs

Todos funcionan en modo PCI y HSMs se configuran con la política de seguridad PCI PTS HSM. Solo están habilitadas las funciones necesarias para admitir los casos AWS de uso de la criptografía de pagos. AWS La criptografía de pagos no permite imprimir, mostrar ni devolver texto sin cifrar. PINs

Seguridad física del dispositivo HSM

El servicio solo podrá utilizar aquellos dispositivos HSMs que cuenten con las claves del dispositivo firmadas por una autoridad certificadora de criptografía de AWS pagos (CA) por el fabricante antes de la entrega. La criptografía de AWS pagos es una subentidad de certificación de la entidad emisora de certificados del fabricante y constituye la base de confianza de los certificados de fabricante y dispositivo de HSM. La CA del fabricante ha certificado el cumplimiento del anexo A de seguridad PCI PIN y el anexo A de PCI P2PE. El fabricante verifica que todos los HSM con claves de dispositivo firmadas por la CA de criptografía de AWS pagos se envíen al receptor designado de AWS.

Tal y como exige la seguridad PCI PIN, el fabricante suministra una lista de números de serie a través de un canal de comunicación distinto al del envío del HSM. Estos números de serie se comprueban en cada paso del proceso de instalación del HSM en los centros de datos de AWS. Por último, los operadores AWS de criptografía de pago validan la lista de HSM instalados comparándola con la lista del fabricante antes de añadir el número de serie a la lista de HSM autorizados a recibir claves de criptografía de pago. AWS

HSMs están guardadas de forma segura o bajo un doble control en todo momento, lo que incluye:

- El envío desde el fabricante a una instalación de montaje en bastidor de AWS.
- Durante el montaje en bastidor.
- Envío desde la instalación de montaje en bastidor a un centro de datos.
- Recepción e instalación en una sala de procesamiento segura de un centro de datos. Los bastidores HSM aplican un doble control con cerraduras de acceso controlado por tarjeta, sensores de puerta con alarma y cámaras.
- Durante las operaciones.
- Durante el desmantelamiento y la destrucción.

Se mantiene y supervisa un sistema completo chain-of-custody, con responsabilidad individual, para cada HSM.

Inicialización de HSM

Un HSM solo se inicializa como parte del conjunto de criptografía de AWS pagos después de validar su identidad e integridad mediante los números de serie, las claves de dispositivo instaladas por el fabricante y la suma de verificación del firmware. Una vez validada la autenticidad e integridad de un HSM, se configura, incluyendo la habilitación del Modo PCI. A continuación, se establecen las claves principales de la región de criptografía de AWS pagos y las claves principales del perfil, y el HSM queda a disposición del servicio.

Servicio y reparación del HSM

Los HSM tienen componentes reparables que no requieren la violación del límite criptográfico del dispositivo. Estos componentes incluyen ventiladores de refrigeración, fuentes de alimentación y baterías. Si un HSM u otro dispositivo dentro del bastidor de HSM necesita servicio, se mantiene el control dual durante todo el periodo en que el bastidor está abierto.

Desactivación de HSM

El desmantelamiento se produce debido a un HSM end-of-life o a un fallo de éste. Los HSM se ponen a cero de forma lógica antes de sacarlos de su rack, si están en funcionamiento, y luego se destruyen en las salas de procesamiento seguras de los centros de datos de AWS. Nunca se devuelven al fabricante para su reparación, ni se utilizan para otro fin, ni se sacan de otro modo de una sala de procesamiento segura antes de su destrucción.

Actualización del firmware del HSM

Las actualizaciones del firmware de HSM se aplican cuando es necesario mantener la alineación con las versiones incluidas en la lista PCI PTS HSM y FIPS 140-2 (o FIPS 140-3), si se trata de una actualización relacionada con la seguridad o si se determina que los clientes pueden beneficiarse de las funciones de una nueva versión. AWS Payment Cryptography HSMs ejecuta off-the-shelf el firmware que coincide con las versiones PCI PTS incluidas en la lista HSM. La integridad de las nuevas versiones de firmware se valida con las versiones de firmware certificadas por PCI o FIPS y, a continuación, se comprueba su funcionalidad antes de lanzarlas a todas. HSMs

Acceso del operador

Los operadores pueden tener acceso no consular a los HSM para la resolución de problemas en los raros casos en que la información recopilada de los HSM durante las operaciones normales sea insuficiente para identificar un problema o planificar un cambio. Se ejecutan los siguientes pasos:

- Se desarrollan y aprueban las actividades de resolución de problemas y se programa la sesión no consular.
- Se retira un HSM del servicio de procesamiento del cliente.
- Se eliminan las claves principales, bajo doble control.
- Se permite al operador el acceso no consular al HSM para realizar las actividades de resolución de problemas aprobadas, bajo control dual.
  - Tras la finalización de la sesión no consular, se realiza el proceso de aprovisionamiento inicial en el HSM, devolviendo el firmware y la configuración estándar y sincronizando la clave principal, antes de devolver el HSM al servicio de los clientes.
  - Los registros de la sesión se graban en el seguimiento de cambios.
  - La información obtenida de la sesión se utiliza para planificar futuros cambios.

Todos los registros de acceso ajenos a la consola se revisan para comprobar la conformidad con los procesos y los posibles cambios en la supervisión del HSM, el proceso de non-console-access gestión o la formación de los operadores.

Administración general de claves

Todos los HSM de una región se sincronizan con una clave principal de región. Una clave principal de región protege al menos una clave principal de perfil. Una clave principal de perfil protege claves de cliente.

Todas las claves principales son generadas por un HSM y distribuidas a mediante la distribución de claves simétricas utilizando técnicas asimétricas, alineadas con ANSI X9 TR 34 y PCI PIN Anexo A.

Generación

Las claves principales AES de 256 bits se generan en uno de los HSM aprovisionados para la flota de HSM de servicio, utilizando el generador de números al azar PCI PTS HSM.

Sincronización de la clave principal de región

Las claves principales de la región HSM son sincronizadas por el servicio en toda la flota regional con mecanismos definidos por ANSI X9 TR-34, que incluyen:

- Autenticación mutua mediante claves del host de distribución de claves (KDH) y del dispositivo receptor de claves (KRD) y certificados para proporcionar autenticación e integridad de las claves públicas.

- Los certificados están firmados por una autoridad de certificación (CA) que cumple los requisitos del Anexo A2 del PIN de la PCI, excepto para los algoritmos asimétricos y las intensidades de clave apropiadas para proteger las claves AES de 256 bits.
- La identificación y la protección de las claves simétricas distribuidas son compatibles con la norma ANSI X9 TR-34 y el PCI PIN anexo A1, excepto en lo que respecta a los algoritmos asimétricos y las intensidades de las teclas adecuadas para proteger las claves AES de 256 bits.

Las claves principales de la región se establecen para las HSMs que se han autenticado y aprovisionado para una región mediante:

- Se genera una clave principal en un HSM de la región. Ese HSM se designa como host de distribución de claves.
- Todas las provisiones HSMs en la región generan un token de autenticación KRD, que contiene la clave pública del HSM e información de autenticación que no se puede reproducir.
- Los tokens KRD se añaden a la lista de permitidos del KDH después de que éste valide la identidad y el permiso del HSM para recibir claves.
- El KDH produce un token de clave principal autenticable para cada HSM. Los tokens contienen la información de autenticación del KDH y la clave principal cifrada que sólo se puede cargar en el HSM para el que se ha creado.
- A cada HSM se le envía el token de clave principal creado para él. Tras validar la información de autenticación propia del HSM y la información de autenticación del KDH, la clave principal se descifra mediante la clave privada del KRD y se carga en la clave principal.

En caso de que un único HSM deba volver a sincronizarse con una región:

- Se vuelve a validar y se aprovisiona con firmware y configuración.
- Si es nuevo en la región:
  - El HSM genera un token de autenticación KRD.
  - El KDH añade el token a su lista de permitidos.
  - El KDH genera un token de clave principal para el HSM.
  - El HSM carga la clave principal.
  - El HSM se pone a disposición del servicio.

Esto garantiza que:

- Solo los HSM validados para el procesamiento de criptografía AWS de pagos en una región pueden recibir la clave maestra de esa región.
- Solo se puede distribuir una clave maestra de un HSM de criptografía de AWS pagos a un HSM de la flota.

Rotación de la clave principal de la región

Las claves principales de región se rotan al expirar el periodo de criptografía, en el improbable caso de que se sospeche que la clave está comprometida, o tras cambios en el servicio que se determine que afectan a la seguridad de la clave.

Se genera y distribuye una nueva clave principal de región como en la provisión inicial. Las claves principales de perfil guardadas deben traducirse a la nueva clave principal de región.

La rotación de la clave principal de región no afecta al procesamiento del cliente.

Sincronización de la clave principal de perfil

Las claves principales de perfil están protegidas por las claves principales de región. Esto restringe un perfil a una región específica.

Las claves principales de perfil se aprovisionan en consecuencia:

- Se genera una clave principal de perfil en un HSM que tenga sincronizada la clave principal de región.
- La clave principal del perfil se almacena y encripta con la configuración del perfil y otros contextos.
- El perfil es utilizado para las funciones criptográficas del cliente por cualquier HSM de la región con la clave principal de la región.

Rotación de la clave principal del perfil

Las claves principales de perfil se rotan al expirar el periodo criptográfico, tras sospecha de compromiso de la clave o tras cambios en el servicio que se determine que afectan a la seguridad de la clave.

Pasos de la rotación:

- Se genera una nueva clave principal de perfil y se distribuye como clave principal pendiente al igual que con la provisión inicial.

- Un proceso en segundo plano traduce el material de la clave de cliente de la clave principal de perfil establecida a la clave principal pendiente.
- Cuando todas las claves de cliente se han cifrado con la clave pendiente, ésta se promueve a clave principal de perfil.
- Un proceso en segundo plano borra el material de las claves de cliente protegido por la clave pendiente.

La rotación de la clave principal del perfil no afecta al procesamiento de los clientes.

Protección

Las claves sólo dependen de la jerarquía de claves para su protección. La protección de las claves principales es fundamental para evitar la pérdida o el compromiso de todas las claves de cliente.

Las claves principales de región son restaurables desde la copia de seguridad sólo para HSM autenticados y aprovisionados para el servicio. Estas claves sólo pueden almacenarse como tokens de clave principal mutuamente autenticables y cifrados de un KDH específico para un HSM específico.

Las claves principales de perfil se almacenan con la configuración del perfil y la información de contexto encriptada por región.

Las claves de cliente se almacenan en bloques de claves, protegidos por una clave maestra de perfil.

Todas las claves existen exclusivamente dentro de un HSM o se almacenan protegidas por otra clave de igual o mayor fuerza criptográfica.

Durabilidad

Las claves de cliente para la criptografía de las transacciones y las funciones empresariales deben estar disponibles incluso en situaciones extremas que suelen provocar interrupciones. AWS La criptografía de pagos utiliza un modelo de redundancia de varios niveles en todas las zonas y regiones de disponibilidad. AWS Los clientes que requieran una mayor disponibilidad y durabilidad para las operaciones criptográficas de pago que la proporcionada por el servicio deberán implementar arquitecturas multirregión.

Los tokens de autenticación del HSM y de la clave principal se guardan y pueden utilizarse para restaurar una clave principal o sincronizarse con una nueva clave principal, en caso de que deba

restablecerse un HSM. Los tokens se archivan y sólo se utilizan bajo doble control cuando es necesario.

El operador accede a las claves principales de HSM

Las claves principales solo existen en el HSM administrado por el servicio y protegido en instalaciones seguras de AWS. Las claves principales no se pueden exportar desde ningún HSM ni sincronizarse con un HSM que no esté inicializado por el fabricante para su uso en el servicio. Los operadores de AWS no pueden obtener claves principales de ninguna forma que puedan cargarse en un HSM no administrado por el servicio.

Gestión de las claves de los clientes

En AWS, la confianza del cliente es nuestra principal prioridad. Usted mantiene el control total de las claves que importe o cree en el servicio en su cuenta de AWS. Usted es responsable de configurar el acceso a las claves.

AWS Payment Cryptography es un proveedor de servicios que usa HSMs y administra las claves en nombre de los clientes, de forma similar a los proveedores de servicios de pago tradicionales. El servicio es totalmente responsable de la seguridad física y lógica de HSM. La responsabilidad de administrar las claves la comparten el servicio y los clientes, ya que el cliente debe proporcionar información precisa sobre las claves creadas por el servicio o importadas al mismo, que el servicio utiliza para garantizar el uso y la administración correctos de las claves. Las protecciones de segregación de datos de AWS se utilizan para garantizar que las claves que pertenecen a una cuenta de AWS no puedan comprometer las claves que pertenecen a otra.

AWS Payment Cryptography es plenamente responsable de la conformidad física del HSM y de la administración de claves de las claves gestionadas por el servicio. Esto requiere la propiedad y la administración de las claves principales del HSM y la protección de las claves de los clientes gestionadas mediante la AWS criptografía de pagos.

Separación del espacio de claves del cliente

AWS La criptografía de pagos aplica políticas clave para todos los usos de las claves, incluida la limitación del capital a la cuenta propietaria de la clave, a menos que una clave se comparta explícitamente con otra cuenta.

Las cuentas de AWS proporcionan una segregación completa del entorno entre clientes o aplicaciones de forma análoga a las implementaciones no basadas en la nube en diferentes centros de datos. Cada cuenta proporciona control de acceso aislado, redes, recursos informáticos,

almacenamiento de datos, claves criptográficas para la protección de datos y las transacciones de pago, y todos los recursos de AWS. Los servicios de AWS, como Organizations y Control Tower, permiten la administración empresarial de cuentas de aplicaciones independientes, de forma análoga a las jaulas o salas de un centro de datos empresarial.

El operador accede a las claves de los clientes

Las claves de cliente gestionadas por el servicio se almacenan protegidas por las claves principales de la partición y solo las puede utilizar la cuenta del cliente propietario o la cuenta que el propietario haya configurado específicamente para compartir las claves. Los operadores de AWS no pueden exportar ni realizar operaciones criptográficas o de administración de claves con las claves de los clientes mediante el acceso manual al servicio, que se administra mediante los mecanismos de acceso manual de los operadores de AWS.

El código de servicio que implementa la administración y el uso de las claves del cliente está sujeto a las prácticas de código seguro de AWS, según se evalúa en la evaluación PCI DSS de AWS.

Copia de seguridad y recuperación

Las claves y la información clave almacenadas internamente por el servicio para una región se guardan en archivos cifrados mediante. AWS Los archivos requieren un doble control AWS para restaurarlos.

Bloques de claves

Todas las claves se almacenan y procesan en bloques de claves con formato ANSI X9.143.

Las claves se pueden importar al servicio desde criptogramas u otros formatos de bloques de claves compatibles con. ImportKey Del mismo modo, las claves pueden exportarse, si son exportables, a otros formatos de bloques de claves o criptogramas soportados por perfiles de exportación de claves.

Uso de claves

El uso de claves está restringido a lo configurado KeyUsage por el servicio. El servicio fallará cualquier solicitud con un uso de clave, modo de uso o algoritmo inapropiados para la operación criptográfica solicitada.

Relaciones de intercambio de claves

PCI PIN Security y PCI P2PE exigen que las organizaciones que comparten claves que cifran PINs o archivan datos, incluidas las claves de intercambio de claves (KEK) utilizadas para compartir esas

claves, no compartan las mismas claves con ninguna otra organización. Se recomienda que las claves simétricas se compartan solo entre dos partes con un único propósito, incluso dentro de la misma organización. Esto minimiza el impacto de presuntos compromisos de claves que obliguen a reemplazar las claves afectadas.

Incluso los casos empresariales que requieren compartir claves entre más de 2 partes, deben mantener el número de partes al mínimo.

AWS La criptografía de pagos proporciona etiquetas clave que se pueden usar para rastrear y hacer cumplir el uso de las claves dentro de esos requisitos.

Por ejemplo, las claves KEK y BDK para diferentes instalaciones de inyección de claves se pueden identificar configurando «KIF» = «POSStation» para todas las claves compartidas con ese proveedor de servicios. Otro ejemplo sería etiquetar las claves compartidas con las redes de pago con «Network» = «PayCard». El etiquetado le permite crear controles de acceso y crear informes de auditoría para hacer cumplir y demostrar sus prácticas de gestión de claves.

Eliminación de claves

DeleteKey marca las claves de la base de datos para eliminarlas después de un período configurable por el cliente. Transcurrido este periodo, la llave se borra irremediablemente. Se trata de un mecanismo de seguridad para evitar el borrado accidental o malintencionado de una clave. Las claves marcadas para su eliminación no están disponibles para ninguna acción excepto: RestoreKey

Las llaves borradas permanecen en las copias de seguridad del servicio durante 7 días después de su borrado. No son restaurables durante este periodo.

Las claves pertenecientes a cuentas AWS cerradas se marcan para su borrado. Si la cuenta se reactiva antes de que se alcance el periodo de borrado, las claves marcadas para borrado se restauran, pero se desactivan. Deberá volver a habilitarlas para poder utilizarlas en operaciones criptográficas.

Seguridad de las comunicaciones

Externo

AWS Los terminales de la API de criptografía de pagos cumplen con los estándares de AWS seguridad, como el TLS 1.2 o superior y la versión 4 de Signature para la autenticación e integridad de las solicitudes.

Las conexiones TLS entrantes se terminan en equilibradores de carga de red y se reenvían a los gestores de API a través de conexiones TLS internas.

Interno

Las comunicaciones internas entre componentes de servicio y entre componentes de servicio y otro servicio de AWS están protegidas por TLS utilizando criptografía fuerte.

Los HSM se encuentran en una red privada no virtual a la que sólo se puede acceder desde los componentes de servicio. Todas las conexiones entre los HSM y los componentes de servicio están protegidas con TLS mutuo (mTLS), igual o superior a TLS 1.2. Los certificados internos para TLS y mTLS son administrados por el Gestor de certificación de Amazon mediante una Autoridad de certificación privada de AWS. La red interna VPCs y la red HSM se supervisan para detectar actividades inesperadas y cambios de configuración.

Registro y supervisión

Los registros de servicios internos incluyen:

- CloudTrail registros de las llamadas al servicio de AWS realizadas por el servicio
- CloudWatch registros de ambos eventos registrados directamente en los CloudWatch registros o eventos de HSM
- Archivos de registro de HSM y sistemas de servicio
- Archivos de registro

Todas las fuentes de registros controlan y filtran la información confidencial, incluida la relativa a las claves. Los registros se revisan sistemáticamente para garantizar que no contienen información sensible de los clientes.

El acceso a los registros está restringido a las personas necesarias para completar las funciones de su trabajo.

Todos los registros se retienen de acuerdo con las políticas de conservación de registros de AWS.

Operaciones de clientes

AWS La criptografía de pagos es totalmente responsable del cumplimiento físico del HSM según las normas PCI. El servicio también proporciona un almacén de claves seguro y garantiza que las claves solo se puedan utilizar para los fines permitidos por los estándares PCI y que usted especifique

durante la creación o la importación. Usted es responsable de configurar los atributos clave y el acceso para aprovechar las capacidades de seguridad y cumplimiento del servicio.

Temas

- Generación de claves
- Importación de claves
- Exportación de claves
- Eliminación de claves de
- Rotar claves de

Generación de claves

Al crear las claves, debe establecer los atributos que el servicio utiliza para hacer cumplir el uso de la clave:

- Longitud de algoritmo y clave
- De uso
- Disponibilidad y vencimiento

Las etiquetas que se utilizan para el control de acceso basado en atributos (ABAC) se utilizan para limitar las claves para su uso con socios o aplicaciones específicos. Asegúrese de incluir políticas para limitar los roles permitidos para borrar o cambiar etiquetas.

Debe asegurarse de que las políticas que determinan los roles que pueden utilizar y gestionar la clave se establecen antes de la creación de la clave.

> **Note**
>
> Las políticas de IAM relativas a los CreateKey comandos se pueden utilizar para imponer y demostrar un doble control en la generación de claves.

Importación de claves

Al importar claves, el servicio establece los atributos para imponer el uso conforme de la clave utilizando la información criptográficamente vinculada del bloque de claves. El mecanismo para

establecer el contexto de clave fundamental consiste en utilizar bloques de claves creados con el HSM de origen y protegidos por un KEK compartido o asimétrico. Esto se ajusta a los requisitos del PCI PIN y preserva el uso, el algoritmo y la solidez de las claves de la aplicación de origen.

En el momento de la importación, se deben establecer los atributos clave, las etiquetas y las políticas de control de acceso importantes, además de la información del bloque clave.

La importación de claves mediante criptogramas no transfiere los atributos clave de la aplicación de origen. Debe configurar los atributos de forma adecuada utilizando este mecanismo.

A menudo, las claves se intercambian utilizando componentes de texto claro, se transmiten por los custodios de las claves y, a continuación, se cargan con la ceremonia que implementa el doble control en una sala segura. Esto no es compatible directamente con la criptografía de AWS pagos. La API exportará una clave pública con un certificado que puede ser importado por su propio HSM para exportar un bloque de claves que sea importable por el servicio. Esto permite utilizar su propio HSM para cargar componentes de texto no cifrado.

Debe utilizar los valores de comprobación de claves (KCV) para verificar que las claves importadas coinciden con las claves de origen.

Las políticas de IAM de la ImportKey API se pueden utilizar para aplicar y demostrar un doble control en la importación de claves.

Exportación de claves

Para compartir claves con socios o aplicaciones en las instalaciones, es posible que sea necesario exportar las claves. El uso de bloques clave para las exportaciones mantiene el contexto fundamental de las claves con el material de las claves cifradas.

Las etiquetas de las claves pueden utilizarse para limitar la exportación de claves a KEK que compartan la misma etiqueta y el mismo valor.

AWS La criptografía de pagos no proporciona ni muestra un texto claro sobre los componentes clave. Esto requiere el acceso directo de los custodios de claves a los dispositivos criptográficos seguros (SCD) probados por la PCI PTS HSM o la ISO 13491 para su visualización o impresión. Puede establecer una KEK asimétrica o una KEK simétrica con su SCD para llevar a cabo la ceremonia de creación de componentes clave de texto claro bajo doble control.

Los valores de comprobación clave (KCV) se deben utilizar para comprobar que los valores importados por el HSM de destino coinciden con las claves de origen.

Eliminación de claves de

Puede usar la API de eliminación de claves para programar la eliminación de las claves tras el período de tiempo que usted configure. Antes de ese momento, las claves se pueden recuperar. Una vez que se eliminan las claves, se eliminan permanentemente del servicio.

Las políticas de IAM de la DeleteKey API se pueden utilizar para aplicar y demostrar un doble control a la hora de eliminar las claves.

Rotar claves de

El efecto de la rotación de claves puede implementarse utilizando alias de claves creando o importando una nueva clave y modificando después el alias de claves para que haga referencia a la clave nueva. La clave anterior se eliminaría o deshabilitaría, según sus prácticas de administración.

Cuotas para AWS Payment Cryptography

La cuenta de AWS tiene cuotas predeterminadas para cada servicios de AWS (estas cuotas se denominaban anteriormente “límites”). A menos que se indique otra cosa, cada cuota es específica de la región. Puede solicitar el aumento de algunas cuotas, pero otras no se pueden aumentar.

Name	Predeterminado	Ajusta	Description (Descripción)
Alias	Cada región admitida: 2000	Sí	El número máximo de alias que puede tener en esta cuenta en la región actual.
Tasa combinada de solicitudes de plano de control	Cada región admitida: 5 por segundo	Sí	El número máximo de solicitudes de plano de control por segundo que puede realizar en esta cuenta en la región actual. Esta cuota se aplica a todas las operaciones del plano de control combinadas.
Tasa combinada de solicitudes de planos de datos (asimétrica)	Cada región admitida: 20 por segundo	Sí	El número máximo de solicitudes por segundo para operaciones en el plano de datos con una clave asimétric a que puede realizar en esta cuenta en la región actual. Esta cuota se aplica a todas las operaciones del plano de datos combinadas.

Name	Predeterminado	Ajusta	Description (Descripción)
Tasa combinada de solicitudes de planos de datos (simétrica)	Cada región admitida: 500 por segundo	Sí	El número máximo de solicitudes por segundo para operaciones de plano de datos con una clave simétrica que puede realizar en esta cuenta en la región actual. Esta cuota se aplica a todas las operaciones del plano de datos combinadas.
Claves	Cada región admitida: 2 000	Sí	El número máximo de claves que puede tener en esta cuenta en la región actual, excluyendo las claves eliminadas.

Historial de documentos del AWS Guía del usuario de criptografía de pagos

En la siguiente tabla se describen las versiones de la documentación sobre criptografía de AWS pagos.

Cambio	Descripción	Fecha
Soporte asimétrico de claves para Multi-Region la replicación de claves	Multi-Region la replicación de claves ahora admite claves de criptografía de pagos asimétricas, como RSA y ECC, además de claves simétricas.	27 de julio de 2026
Nueva función: AS2805	Soporte para algoritmos y flujos para soportar el soporte regional del AS2805	17 de diciembre de 2025
Nueva función: replicación Multi-Region clave	Con Multi-Region la replicación de claves, puede replicar sus claves AWS de criptografía de pagos en varias Regiones de AWS.	10 de septiembre de 2025
Nueva función: ECDH	Con esta versión, el ECDH se puede usar para establecer una KEK compartida para un mayor intercambio de claves.	30 de marzo de 2025
Nueva guía de intercambio de claves	Se proporcionan nuevas directrices para los intercambios de claves. También se agregó información sobre los comandos comunes de JCB.	31 de enero de 2025
Lanzamiento de una nueva región	Se agregaron terminales para el lanzamiento de nuevas	31 de julio de 2024

	regiones en Europa (Fráncfor t), Europa (Irlanda), Asia Pacífico (Singapur) y Asia Pacífico (Tokio)	
CloudTrail para claves dinámicas y de plano de datos	Se agregó información sobre la utilización CloudTrail de operaciones (criptográficas) en el plano de datos, incluidos ejemplos. También se agregó información sobre el uso de claves dinámicas para ciertas funciones a fin de admitir mejor las claves únicas o de uso limitado que no deben importarse a la criptografía de AWS pagos	10 de julio de 2024
Ejemplos actualizados	Se agregaron nuevos ejemplos para la emisión de tarjetas	1 de julio de 2024
Lanzamiento de una función	Añadir información sobre los puntos de enlace de la VPC (PrivateLink) y ejemplos de iCVV.	30 de mayo de 2024
Lanzamiento de funciones	Se agregó información sobre las nuevas funciones relaciona das con el import/export uso de RSA y la exportación de claves DUKPT IPEK/IK .	15 de enero de 2024
Versión inicial	Publicación inicial de la guía del usuario de criptografía de AWS pagos	8 de junio de 2023

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.